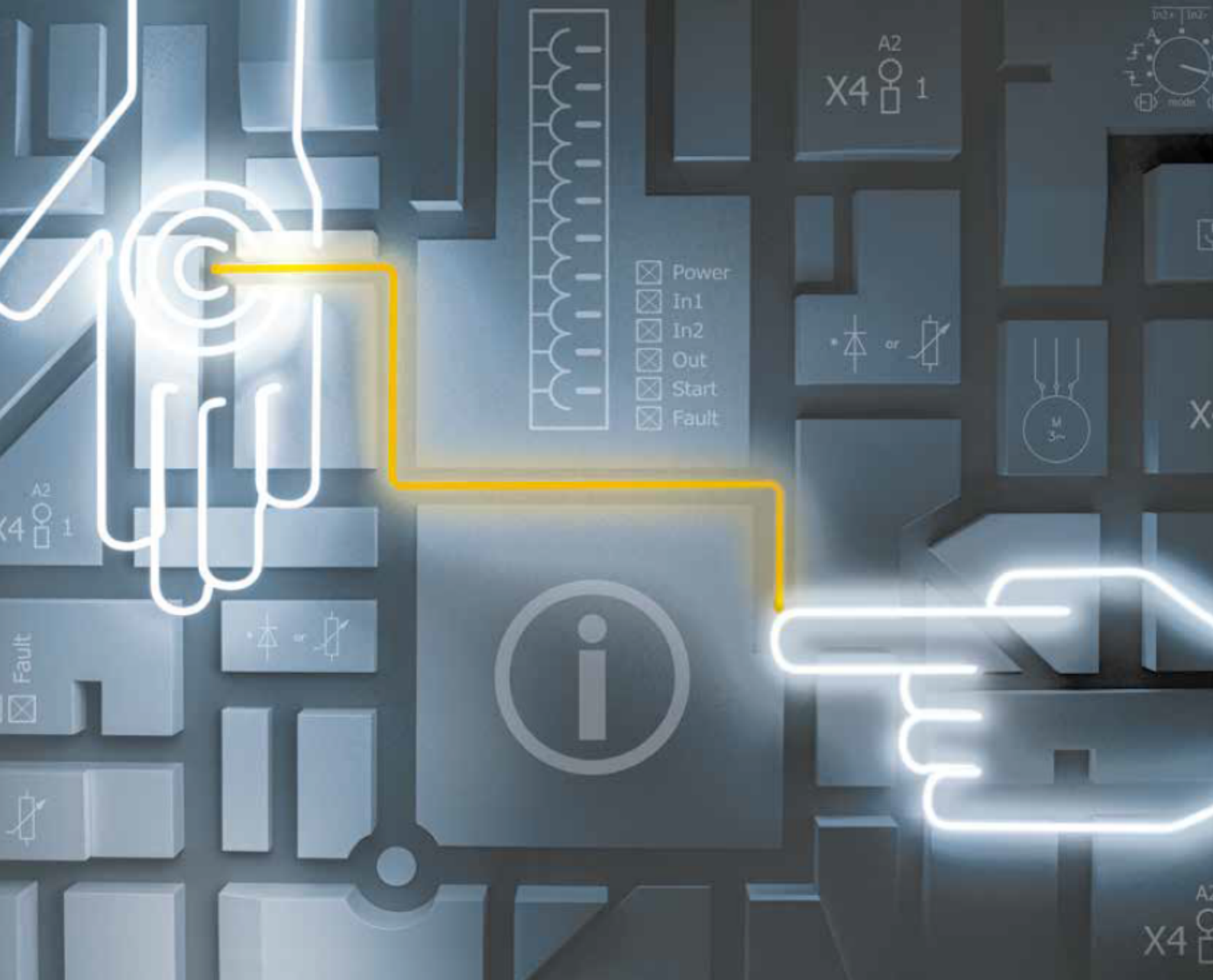




► Automation system PSS 4000 PSS 4000 Firmware 1.29.x

PILZ
THE SPIRIT OF SAFETY

System Description-1001467-EN-37



This document is the original document.

Where unavoidable, for reasons of readability, the masculine form has been selected when formulating this document. We do assure you that all persons are regarded without discrimination and on an equal basis.

All rights to this documentation are reserved by Pilz GmbH & Co. KG. Copies may be made for the user's internal purposes. Suggestions and comments for improving this documentation will be gratefully received.

CECE®, CHRE®, CMSE®, INDUSTRIAL PI®, Leansafe®, MYZEL®, PAS4000®, PAS-cal®, PASconfig®, Pilz®, PIT®, PMCprimo®, PMCprotego®, PM Ctendo®, PMD®, PMI®, PNOZ®, Primo®, PSEN®, PSS®, PVIS®, SafetyBUS p®, SafetyEYE®, SafetyNET p®, THE SPIRIT OF SAFETY® are registered and protected trademarks of Pilz GmbH & Co. KG in some countries.



SD means Secure Digital

1	Introduction	16
1.1	Validity of the documentation	16
1.2	Definition of symbols	17
2	Automation system PSS 4000	18
2.1	Introduction to PSS 4000	18
2.1.1	General	18
2.1.2	SafetyNET p.....	19
2.1.3	Devices in the automation system PSS 4000	20
2.1.3.1	System overview	20
2.1.3.2	System sections	22
2.1.3.3	PSS 4000-exclusive devices.....	26
2.1.3.4	PSS 4000-operable devices.....	28
2.1.3.5	PSS 4000-suitable devices	28
2.1.4	PAS4000	29
2.1.5	Diagnostics	30
2.1.6	OPC Server	31
2.2	Modularity with PSS 4000	32
2.2.1	Centralised view of a networked system	32
2.2.2	Scalability of the devices	33
2.2.3	Modularisation within programming	34
3	Safety	35
3.1	Additional documents that apply	35
3.2	Intended use	36
3.2.1	Safety Integrity Level (SIL) and Performance Level (PL).....	38
3.2.2	General safety assessments	39
3.2.3	Third-party manufacturer licence information.....	40
3.3	Use of qualified personnel.....	41
3.4	Failsafe inputs/outputs	42
4	Security	43
4.1	Generally required security measures	45
4.2	Security measures on PSS 4000-exclusive devices	45
4.3	Security measures on the tool PAS4000	46
4.4	Security measures for the project	47
4.5	Secure network	48
5	PSSu system basics	49
5.1	Structure.....	50
5.2	Module layout.....	53
5.3	Module names.....	55
5.4	Module bus PSSu	57
5.5	Base modules	58
5.6	Connections on the base modules	59
5.6.1	Connection levels on input/output modules	60
5.6.2	Colour marking on the connection levels	61
5.7	Connections on the compact modules	62

5.8	Maximum system expansion of a PSSu system	63
5.8.1	Number of modules and number of inputs/outputs on a PSSu system	64
5.8.2	Maximum current load.....	66
5.8.2.1	Maximum current load capacity of the supply voltage modules.....	66
5.8.2.2	Maximum current load capacity based on temperature	67
5.9	Slots of a PSSu system.....	68
5.9.1	Set/actual hardware registry on a PSSu system.....	70
5.10	Supply voltage and insulation	73
5.11	Supply groups	76
5.12	Maximum ambient temperature	78
5.13	Status LEDs on head modules.....	79
6	Basics about PSS67 devices	86
6.1	Structure.....	87
6.2	Module bus PSSu	88
6.3	Slots on a PSS67 device.....	88
6.4	Supply voltage.....	89
6.5	Status LEDs	90
7	Basics of PSS u2 systems	91
7.1	Structure of a PSS u2 system	92
7.2	Module layout.....	96
7.3	Module names.....	97
7.4	Module bus PSS u2	98
7.5	Backplanes.....	99
7.6	Maximum system expansion of a PSS u2 system	100
7.6.1	Maximum number of modules in a PSS u2 system	101
7.6.2	Output power of module supply	102
7.6.3	Current load capacity of periphery supply.....	104
7.7	Slots on a PSS u2 system.....	105
7.8	Set/actual hardware registry on a PSS u2 system.....	106
7.9	Validity of the output signals	107
7.10	Supply voltage.....	107
7.11	Supply groups	110
7.12	Maximum ambient temperature	112
7.13	Status LEDs on PSS u2 head modules	113
7.14	MDF files and module descriptions (MD).....	120
7.15	Error behaviour of the inputs and outputs.....	122
7.15.1	Behaviour of the ST outputs.....	122
7.15.2	Behaviour of the FS outputs.....	122
7.15.2.1	FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system"	122
7.15.2.2	FS shutdown behaviour "Shutdown of all FS outputs on the electronic module"	125
7.15.2.3	FS outputs as auxiliary outputs	129
7.15.3	Behaviour of the ST and FS inputs	129
8	Ethernet	130
8.1	Auto IP mechanism	130
8.2	Ethernet interface.....	131

8.2.1	RJ45 interface	132
8.2.2	RJ45 interface assignment.....	133
8.2.3	Connector and connection cable for RJ45	134
8.2.4	Status LEDs on the RJ45 interface	136
8.3	Extended network configuration.....	137
8.3.1	Ports' Ethernet mode	138
8.3.2	Contents of the network configuration file	140
8.3.2.1	DOCUMENT	142
8.3.2.2	HARDWARE	143
8.3.2.3	ETH_INTERFACE.....	144
8.3.2.4	DSCP_RULE.....	148
8.3.3	Validation of the network configuration file.....	151
8.3.4	Examples of network configuration files.....	158
8.3.4.1	Network configuration file with switched mode	158
8.3.4.2	Network configuration file with independent mode.....	161
8.3.5	Activation and deactivation of the network configuration	163
8.3.6	Requirements of receive data	165
9	SafetyNET p	166
9.1	Overview	166
9.1.1	SafetyNET p protocol versions.....	167
9.1.2	SafetyNET p in ISO/OSI's 7-layer reference model.....	168
9.1.3	RTFN (Real Time Frame Network)	169
9.1.4	Transfer of safety-related and non-safety-related data	170
9.2	Topology on SafetyNET p.....	171
9.2.1	Linear topology.....	171
9.2.2	Star topology	172
9.2.3	Dynamic structures	173
9.3	Selecting SafetyNET p components	174
9.4	SafetyNET p interface	175
9.5	Process data exchange via SafetyNET p.....	176
9.5.1	Cyclical processing of connections	178
9.5.2	SafetyNET p receive timeout	179
9.5.2.1	SNp protocol version 1: Communication errors and resumption of FS communication.....	180
9.5.2.2	SNp protocol version 2: Communication errors and resumption of FS communication.....	181
9.5.2.3	SNp protocol version 2: Configurable FS SafetyNET p wait time	182
9.5.3	Data transfer limits	184
9.6	Project-internal SafetyNET p communication	187
9.7	Cross-project SafetyNET p communication	188
9.8	FS heartbeat (FS-HBT).....	190
9.9	SafetyNET p scheduling mode.....	192
9.10	Network without Routers	193
9.11	Network with Routers	194
9.11.1	System variable PILZ_SNP_MULTICAST_TTL.....	196
9.12	SafetyNET p via WLAN.....	197
9.13	Availability optimised in SafetyNET p.....	199
9.14	Prevent data exchange with projects with identical project name.....	199

10	External communication	201
10.1	Communication modes for external communication	201
10.1.1	Fieldbuses	201
10.1.2	IP connections	201
10.2	Modbus/TCP	203
10.2.1	"IP connections" system section	204
10.2.2	Modbus/TCP connections	205
10.2.3	Modbus/TCP Server connection (server connections)	206
10.2.4	Function Codes (Client connections)	207
10.2.5	Relationship between function codes and Modbus/TCP data areas	208
10.2.6	Data exchange via Modbus/TCP	209
10.2.6.1	Send data and receive data	209
10.2.6.2	Server allocation table	210
10.2.7	Data transfer limits	212
10.2.8	Modbus/TCP I/O data	214
10.2.9	Address display of Modbus/TCP addresses	218
10.2.9.1	Address format of send data	218
10.2.9.2	Address format of receive data	218
10.2.9.3	Address display in the Server allocation table	218
10.2.10	Data consistency on Modbus/TCP	219
10.2.11	Special features when using the data areas 0x and 4x	220
10.3	Raw TCP	224
10.3.1	"IP connections" system section	225
10.3.2	Raw TCP connections	226
10.3.3	Data exchange via Raw TCP	227
10.3.4	FIFO memory for receive data	228
10.3.5	Processing receive data with Raw TCP	229
10.3.5.1	Typical application scenarios for Raw TCP	230
10.3.5.2	Basic workflow when processing receive data	232
10.3.5.3	Receive behaviour when FIFO memory is enabled	233
10.3.5.4	Receive behaviour when FIFO memory is disabled	235
10.3.6	Data transfer limits	237
10.3.7	Raw TCP I/O data	238
10.3.8	Address format of Raw TCP addresses	241
10.3.9	Data consistency on Raw TCP	242
10.4	Raw UDP	243
10.4.1	"IP connections" system section	244
10.4.2	Raw UDP connections	245
10.4.3	Data exchange via Raw UDP	246
10.4.4	FIFO memory for receive data	247
10.4.5	Data transfer limits	248
10.4.6	Raw UDP I/O data	249
10.4.7	Address format of Raw UDP addresses	252
10.4.7.1	Address format of send data	252
10.4.7.2	Address format of receive data	252
10.4.8	Data consistency on Raw UDP	253
10.5	UDP communication using blocks	254
10.5.1	"IP connections" system section	256
10.5.2	IP interfaces for UDP communication	257

10.5.3	UDP sockets	257
10.5.4	Data exchange via "UDP using blocks"	259
10.5.5	UDP send buffer and UDP receive buffer	260
10.5.6	Data transfer limits	261
10.5.7	Time behaviour when sending a UDP datagram	263
10.5.8	Time behaviour when receiving a UDP datagram	265
10.5.9	Time requirement during UDP communication using blocks	266
10.5.10	Maximum UDP send rate and UDP receive rate	269
10.5.11	Data consistency during UDP communication using blocks	270
10.5.12	Safe UDP communication using blocks	271
10.6	PROFIBUS-DP slave	273
10.6.1	"PROFIBUS-DP Slave" system section	274
10.6.2	Data exchange via PROFIBUS-DP	275
10.6.2.1	Send data and receive data	275
10.6.2.2	Virtual PROFIBUS modules for send data	277
10.6.2.3	Virtual PROFIBUS modules for receive data	279
10.6.3	Data transfer limits	281
10.6.4	I/O data of PROFIBUS-DP Slave	282
10.6.5	Address display of PROFIBUS-DP addresses	285
10.6.5.1	Address format of send data	285
10.6.5.2	Address format of receive data	286
10.6.6	PROFIBUS diagnostics	287
10.6.6.1	System-internal diagnostics	288
10.6.6.2	External diagnostics data	289
10.6.7	Data consistency on PROFIBUS-DP	292
10.7	PROFIBUS-DP Slave with PROFI-safe	293
10.7.1	"PROFIBUS-DP Slave with PROFI-safe" system section	294
10.7.2	PROFIBUS-DP in the "PROFIBUS-DP Slave with PROFI-safe" system section	295
10.7.3	PROFI-safe in the "PROFIBUS-DP Slave with PROFI-safe" system section	296
10.7.3.1	Pre-defined F-modules	297
10.7.3.2	Virtual PROFI-safe submodules for send data	300
10.7.3.3	Virtual PROFI-safe submodules for receive data	301
10.7.4	GSD file for virtual PROFIBUS modules	302
10.7.5	Data transfer limits	303
10.7.6	PROFI-safe I/O data	304
10.7.7	Address format of PROFI-safe addresses	307
10.7.7.1	Address format of send data	307
10.7.7.2	Address format of receive data	308
10.7.8	Deactivation and reactivation of an F-module	309
10.7.9	F-parameters	310
10.7.10	PROFI-safe diagnostics	311
10.7.10.1	Device-based diagnostics of an F-device	312
10.7.11	Data consistency on PROFI-safe	314
10.8	EtherNet/IP Adapter	315
10.8.1	"EtherNet/IP Adapter" system section	317
10.8.2	EtherNet/IP Adapter connections	318
10.8.3	Data exchange via EtherNet/IP	319
10.8.3.1	Send data and receive data	319
10.8.3.2	Virtual modules for send data	320

10.8.3.3	Virtual modules for receive data.....	321
10.8.4	Data transfer limits	322
10.8.5	EtherNet/IP Adapter I/O data	323
10.8.6	Address format of EtherNet/IP addresses.....	326
10.8.6.1	Address format of send data	326
10.8.6.2	Address format of receive data	327
10.8.7	EtherNet/IP diagnostics.....	328
10.8.8	Data consistency on EtherNet/IP	330
10.8.9	Supported EtherNet/IP objects and Common Services	331
10.8.9.1	Identity Object (Class ID 0x01)	332
10.8.9.2	Assembly Object (Class ID 0x04)	334
10.8.9.3	TCP/IP Interface Object (Class ID 0xF5)	335
10.8.9.4	Link Object (Class ID 0xF6)	336
10.8.9.5	Common Services.....	337
10.9	PROFINET IO DEVICE.....	338
10.9.1	"PROFINET IO DEVICE" system section	339
10.9.2	Incorporating and addressing a device in PROFINET	340
10.9.3	Identifying a PSS 4000 device in PROFINET	341
10.9.4	Data exchange via PROFINET	342
10.9.4.1	Sending and receiving data.....	342
10.9.4.2	Virtual PROFINET modules for send data	343
10.9.4.3	Virtual PROFINET modules for receive data	345
10.9.5	GSDML file.....	347
10.9.6	Data transfer limits	348
10.9.7	I/O data of PROFINET IO DEVICE	349
10.9.8	Address format of PROFINET IO addresses	352
10.9.8.1	Address format of send data	352
10.9.8.2	Address format of receive data	353
10.9.9	PROFINET IO diagnostics	354
10.9.10	Data consistency with PROFINET IO	355
11	Project	356
11.1	Project structure	356
11.2	Validity of process data	359
11.2.1	Substitute values.....	362
11.2.2	Evaluation of the valid bits in the user program	363
11.3	Task basics	364
11.3.1	Task properties	365
11.3.2	Task execution	366
11.3.3	Default configuration of tasks.....	369
11.4	I/O mapping rules within a PSS 4000 project.....	370
11.5	I/O mapping rules between PSS 4000 projects	373
11.6	Data consistency.....	377
11.6.1	Consistency ranges.....	378
11.6.1.1	Consistency ranges on 1st generation devices.....	378
11.6.1.2	Consistency ranges on 2nd generation devices	381
11.6.1.3	Example of consistency ranges of O-PI variables.....	384
11.6.2	Influence of scheduling on data consistency.....	385
11.7	Safety status of blocks	386

11.8	Device naming via PAS4000	388
11.8.1	Device identification data	389
11.8.2	Naming data	390
11.8.3	Device naming for PSS 4000-exclusive devices	392
11.8.3.1	Error behaviour during device naming	396
11.8.4	Device naming for PSS 4000-operable devices	397
11.9	Device naming when PSS 4000-exclusive devices are exchanged	400
11.10	Project download	401
11.10.1	Direct project download	401
11.10.2	Indirect project download	407
11.11	Sealed FS blocks	409
12	Mode of operation	410
12.1	Communication between PAS4000 and PSS 4000 devices	410
12.1.1	Establish a point-to-point connection	411
12.1.2	Establish a remote connection	412
12.2	Operating states and changes in operating state of a device	413
12.2.1	Flow chart	414
12.2.2	Operating states of a device	415
12.2.2.1	Device switched off	416
12.2.2.2	Operating status "Booting base system"	417
12.2.2.3	Operating status "Booting system sections"	418
12.2.2.4	Operating status "Device in RUN condition without error"	426
12.2.2.5	Operating status "Device in STOP condition without error"	427
12.2.2.6	Operating status "Major FS error"	428
12.2.3	Change in operating state of a device	430
12.3	Operating states and change in operating status of the FS resource	431
12.3.1	Flow chart	432
12.3.2	Operating states of the FS resource	433
12.3.2.1	FS resource in RUN condition without error	434
12.3.2.2	FS resource in STOP condition without error	435
12.3.2.3	FS resource in RUN condition with error: Task in TERMINATED condition	436
12.3.2.4	FS resource in RUN condition with error: Task in STOP condition	437
12.3.2.5	Major FS error	438
12.3.3	Change in operating status of the FS resource	439
12.4	Operating states and change in operating status of the ST resource	442
12.4.1	Flow chart	443
12.4.2	Operating states of the ST resource	444
12.4.2.1	ST resource in RUN condition without error	445
12.4.2.2	ST resource in STOP condition without error	446
12.4.2.3	ST resource in RUN condition with error: Task in TERMINATED condition	447
12.4.2.4	ST resource in RUN condition with error: Task in STOP condition	448
12.4.2.5	Minor ST error	449
12.4.3	Change in operating status of the ST resource	450
12.5	Operating states and changes in operating state of the FS module bus PSSu	453
12.5.1	Flow chart	454
12.5.2	Operating states of the FS module bus PSSu	455
12.5.2.1	FS module bus PSSu in RUN state without error	456
12.5.2.2	FS input module in RUN state without error	457

12.5.2.3	FS module in RUN condition with input error	458
12.5.2.4	FS input module in STOP state with module error	459
12.5.2.5	FS input modules of the device in STOP state with module error	460
12.5.2.6	FS output module in RUN state without error	461
12.5.2.7	Safe state of all FS outputs on the device	462
12.5.2.8	Major FS error	463
12.5.3	Change in operating state of the FS module bus PSSu	464
12.6	Operating states and change in operating state of the FS module bus PSS u2	466
12.6.1	Flowchart - Operating states and change in operating state of the FS module bus PSS u2	467
12.6.2	Sub-chart - FS input modules	469
12.6.3	Sub-chart - Shutdown of all FS outputs on the electronic module	470
12.6.4	Sub-chart - Shutdown of all FS outputs on the PSS u2 system	471
12.6.5	Sub-chart - FS output as auxiliary output	472
12.6.6	Operating states of the FS module bus PSS u2	473
12.6.6.1	FS module bus PSS u2 in RUN state without error	474
12.6.6.2	FS input module in RUN state without error	475
12.6.6.3	FS input module in RUN state with input error	476
12.6.6.4	FS input module in STOP state with module error	477
12.6.6.5	FS output module in RUN state without error	478
12.6.6.6	Safe state of all FS outputs on the module	479
12.6.6.7	Safe state of all FS outputs on the device	480
12.6.6.8	Auxiliary output without error	481
12.6.6.9	Safe state of an auxiliary output	481
12.6.6.10	Major FS error	481
12.6.7	Change in operating state of the FS module bus PSS u2	482
12.7	Operating states and changes in operating state of the ST module bus PSSu	486
12.7.1	Flow chart	487
12.7.2	Operating states of the ST module bus PSSu	488
12.7.2.1	ST module bus PSSu in RUN state without error	489
12.7.2.2	ST input module in RUN state without error	490
12.7.2.3	ST module in RUN condition with input error	491
12.7.2.4	ST input module in STOP state with module error	492
12.7.2.5	ST output module in RUN state without error	493
12.7.2.6	ST module in RUN condition with output error	494
12.7.2.7	ST output module in STOP state with module error	495
12.7.2.8	Minor ST error	496
12.7.3	Change in operating state of the ST module bus PSSu	497
12.8	Operating states and change in operating state of the ST module bus PSS u2	499
12.8.1	Flowchart	500
12.8.2	Operating states of the ST module bus PSS u2	501
12.8.2.1	ST module bus PSS u2 in RUN state without error	502
12.8.2.2	ST input module in RUN state without error	503
12.8.2.3	ST module in RUN condition with input error	504
12.8.2.4	ST input module in STOP state with module error	505
12.8.2.5	ST output module in RUN state without error	506
12.8.2.6	ST module in RUN condition with output error	507
12.8.2.7	ST output module in STOP state with module error	508
12.8.2.8	ST module bus PSS u2 in STOP state with error	509
12.8.3	Change in operating state of the ST module bus PSS u2	510

12.9	Operating states and changes in operating state of FS SafetyNET p	512
12.9.1	Flow chart.....	513
12.9.2	Operating states of FS SafetyNET p.....	514
12.9.2.1	FS SafetyNET p in RUN state without error.....	515
12.9.2.2	Major FS error	516
12.9.2.3	Major FS+ST error	517
12.9.2.4	SNp protocol version 1: FS SafetyNET p interface in RUN state with minor error.....	518
12.9.2.5	SNp protocol version 2: FS SafetyNET p interface in RUN state with minor error.....	519
12.9.3	Change in operating state of FS SafetyNET p	521
12.10	Operating states and changes in operating state of ST SafetyNET p	523
12.10.1	Flow chart.....	524
12.10.2	Operating states of ST SafetyNET p.....	525
12.10.2.1	ST SafetyNET p in RUN state without error.....	526
12.10.2.2	Major FS+ST error	527
12.10.2.3	ST SafetyNET p in RUN state with minor error.....	528
12.10.3	Change in operating state of ST SafetyNET p	529
12.11	Operating states and change in operating status with external communication	530
12.11.1	Flow chart.....	531
12.11.2	Operating states with external communication	532
12.11.2.1	System section in RUN condition without error.....	533
12.11.2.2	Minor error on the system section for external communication.....	534
12.11.2.3	System section in RUN condition with error.....	535
12.11.2.4	Major FS+ST error	536
12.11.2.5	Major FS error	537
12.11.3	Operating status change with external communication.....	538
12.12	Reset, restart, start and stop.....	540
12.12.1	Overview	541
12.12.1.1	Reset, restart, start and stop options	541
12.12.1.2	Impact on variables and application parameters.....	542
12.12.1.3	Impact on active "variable forcing"	543
12.12.1.4	Impact on "variable controlling"	544
12.12.1.5	Effects during active online changes.....	545
12.12.2	Cold start and Reboot command	546
12.12.3	Reset commands	547
12.12.4	Hot start commands	549
12.12.5	Restart commands	550
12.12.6	Stop commands	552
12.13	Functions of the reset button.....	554
12.13.1	Perform a warm reset for a device	555
12.13.2	Deliberate operator action to transfer the naming data and/or device project	556
12.13.3	Recovery mode	561
12.13.4	Manual start of OPC UA commissioning mode.....	562
12.14	Original reset.....	563
12.14.1	Effects of an original reset.....	564
12.14.1.1	Effects on a PSS 4000-exclusive device.....	565
12.14.1.2	Effects on a PSS 4000-operable device	566
12.14.2	Perform original reset.....	567
12.15	Enable principle.....	568
12.16	Reaction times	570

12.16.1	Reaction times for PSSu systems and PSS67 devices	571
12.16.1.1	Times that influence reaction times.....	572
12.16.1.2	Processing rules during data processing	578
12.16.1.3	Formation of data paths	579
12.16.1.4	Maximum reaction times for data subpaths	581
12.16.1.4.1	Maximum reaction time for data subpaths with module bus as data source.....	581
12.16.1.4.2	Data subpaths with SafetyNET p as data source.....	582
12.16.1.4.3	Data subpaths with external I-data as data source	583
12.16.1.4.4	Data subpaths with a task as data source	584
12.16.1.4.5	Example for calculating a data path on one device	585
12.16.1.4.6	Example for calculating a data path via three devices	586
12.16.2	Reaction times for PSS u2 systems.....	587
12.16.2.1	Times that influence reaction times.....	588
12.16.2.2	Processing rules during data processing	592
12.16.2.3	Formation of data paths	593
12.16.2.4	Principles for determining the maximum reaction time	595
12.16.2.5	Maximum reaction times for data subpaths	596
12.16.2.5.1	Data subpaths with module bus PSS u2 as data source	597
12.16.2.5.2	Data subpaths with SafetyNET p as data source.....	599
12.16.2.5.3	Data subpath with a task as data source	601
12.16.2.5.4	Example for calculating a data path on a PSS u2 system	603
12.16.2.5.5	Example for calculating a data path via three PSS u2 systems.....	605
12.17	System time synchronisation	607
12.17.1	System time synchronisation with SNTP	608
12.18	Memory	609
12.18.1	Memory types.....	609
12.18.2	Memory structure	610
12.18.3	Security SD card	613
12.18.4	Journaling file system.....	613
12.18.5	Bind SD card	613
13	Commissioning	614
13.1	General requirements	614
13.2	Initial commissioning	615
13.3	Recommissioning.....	619
14	Maintenance, repair, decommissioning	623
14.1	Maintenance and repair	623
14.2	Decommissioning.....	624
15	OPC Server (classic)	625
15.1	Overview	625
15.1.1	Design options	626
15.1.2	OPC properties	628
15.2	System requirements	629
15.2.1	PC with Microsoft Windows operating system	629
15.3	Licensing	630
15.4	Installation	631
15.4.1	Install PSS 4000 firmware on a PC.....	632

15.4.1.1	First-time installation of the PSS 4000 firmware	636
15.5	Communication and security settings for PCs	637
15.5.1	DCOM configuration.....	638
15.5.1.1	Start DCOM configuration.....	638
15.5.1.2	DCOM configuration for the OPC Server and OPC Clients	641
15.5.1.3	DCOM configuration for the OPCServer.....	647
15.5.1.4	OPCEnum configuration for the OPC Server.....	656
15.5.1.5	Special feature of the user "Administrators".....	664
15.5.1.6	Special feature of the user "NETWORK"	665
15.5.2	Firewall configuration	666
15.6	Commissioning the OPC Server	667
15.7	Data type assignment	668
15.8	Access to data in the PSS 4000 namespace.....	669
15.9	Network data (OPC Server (classic)).....	669
16	OPC Server UA (PC, PMI)	671
16.1	Overview	671
16.1.1	Design options	672
16.1.2	OPC UA properties	674
16.1.3	Limit values	676
16.1.4	Access rights of OPC UA Clients.....	678
16.1.5	Security settings for end points	678
16.2	System requirements	679
16.2.1	PC with Microsoft Windows operating system	679
16.3	Licensing	680
16.4	Installation	681
16.4.1	Install PSS 4000 firmware on a PC.....	681
16.5	Information security certificates.....	686
16.5.1	Create certificate during installation.....	687
16.5.2	Directory structure of the certificate store	688
16.5.3	Procedure for authentication	689
16.5.3.1	Authentication in the case of local installation.....	689
16.5.3.2	Authentication in the case of remote installation.....	689
16.5.3.3	Authentication in the case of OPC UA Clients of a third-party manufacturer.....	693
16.5.4	Generating and using custom certificates.....	694
16.5.5	Invalidating a certificate.....	696
16.5.6	Verifying a certificate.....	696
16.5.7	Security policies of OPC UA Servers.....	697
16.6	Firewall configuration	699
16.7	Data type assignment	700
16.7.1	Access to data in the PSS 4000 namespace.....	701
16.7.2	Special features when accessing arrays in the namespace	701
16.8	Network data (OPC Server UA).....	702
17	OPC Server UA (device-internal)	704
17.1	Overview	704
17.1.1	Design options	706
17.1.2	OPC UA properties	708

17.1.3	Limit values	710
17.1.4	Access rights of OPC UA Clients	711
17.1.5	Security settings for end points	711
17.1.6	Diagnostics	712
17.1.7	Audit log	712
17.2	System requirements	713
17.3	Activation/deactivation of the device-internal OPC UA Server	713
17.4	Ethernet mode	714
17.5	Information security certificates	715
17.5.1	OPC UA commissioning mode for certificate management	715
17.5.2	Security Policies of the device-internal PSS 4000 OPC Server UA	716
17.5.3	RSA key size	716
17.6	Firewall configuration	718
17.7	Data type assignment	719
17.7.1	Access to data in the PSS 4000 namespace	720
17.7.2	Special features when accessing arrays in the namespace	720
17.8	Network data (OPC Server UA)	721
18	Diagnostic basics	722
18.1	Diagnostic list	723
18.1.1	Diagnostic messages	724
18.1.1.1	State of diagnostic messages	726
18.1.1.2	Severity and priority	727
18.1.1.3	Responsibilities	728
18.1.1.4	Scopes	729
18.1.1.5	Location information	730
18.1.1.6	Diagnostic identifier	734
18.2	Diagnostic log	736
18.2.1	Log entries	737
18.3	Diagnostic server	739
19	PSS 4000 namespace	741
19.1	Basics	742
19.2	Structure of the namespace	743
19.3	Main area of the PSS 4000 namespace	744
19.4	General project data	745
19.4.1	Project information	746
19.4.2	user_prg	747
19.4.3	Diag	749
19.5	Device-based project data	754
19.5.1	Device	755
19.5.1.1	State	757
19.5.1.2	DeviceProject	770
19.5.1.3	Product	771
19.5.1.4	Network	774
19.5.1.5	FS resource	776
19.5.1.6	ST resource	778
19.5.1.7	Diag	780

19.5.1.8	OPCServer.....	787
19.6	Data types in the PSS 4000 namespace	788
20	Appendix	789
20.1	Factory default settings for port numbers for various services.....	789
20.2	Factory default settings for network parameters	790
20.3	Network data	791
	Glossary	792

1 Introduction

1.1 Validity of the documentation

This documentation is valid for the automation system PSS 4000. It is valid until new documentation is published.

The current documentation is available in the download area of the Pilz homepage (www.pilz.de -> Downloads).

This document often uses the term "PSS 4000" as an abbreviation of "Automation system PSS 4000".

1.2 Definition of symbols

Information that is particularly important is identified as follows:



DANGER!

This warning must be heeded! It warns of a hazardous situation that poses an immediate threat of serious injury and death and indicates preventive measures that can be taken.



WARNING!

This warning must be heeded! It warns of a hazardous situation that could lead to serious injury and death and indicates preventive measures that can be taken.



CAUTION!

This refers to a hazard that can lead to a less serious or minor injury plus material damage, and also provides information on preventive measures that can be taken.



NOTICE

This describes a situation in which the product or devices could be damaged and also provides information on preventive measures that can be taken. It also highlights areas within the text that are of particular importance.



INFORMATION

This gives advice on applications and provides information on special features.



In preparation

This function or description is not yet available.
It will not concern safety-related functions or descriptions.

2 Automation system PSS 4000

2.1 Introduction to PSS 4000

2.1.1 General

PSS 4000 is an automation system that can be used to tackle the widest range of automation functions. These include requirements from the fields of control technology (PLC), motion control and visualisation.

PSS 4000 can be used at every stage of automation, ranging from a small, decentralised automation terminal to a complex network at control level.

The automation system PSS 4000 handles non-safety-related functions (ST functions) as well as safety-related functions (FS functions) for control, motion control and visualisation tasks.

2.1.2 **SafetyNET p**

SafetyNET p is used for networked automation functions. SafetyNET p is an Ethernet-based multi-master bus system suitable for industrial use.

The SafetyNET p protocol contains a safe data channel for safety-related data. This channel is certified for data transfer in accordance with SIL 3 of IEC 61508. On SafetyNET p, safety and non-safety-related data is transferred via different data channels, but using the same connection cable.

In terms of the topology, SafetyNET p supports various familiar Ethernet architectures. So modular system architectures and individual subnetworks can be implemented. If necessary, this method can be used to divide automation functions into decentralised and largely independent function units. The main benefits of this type of architecture lie in the reusability, modification and availability of function units.

2.1.3 Devices in the automation system PSS 4000

2.1.3.1 System overview

Devices in the automation system PSS 4000 are called **PSS 4000 devices**. The diagram below provides an overview of how the devices are divided and subdivided:

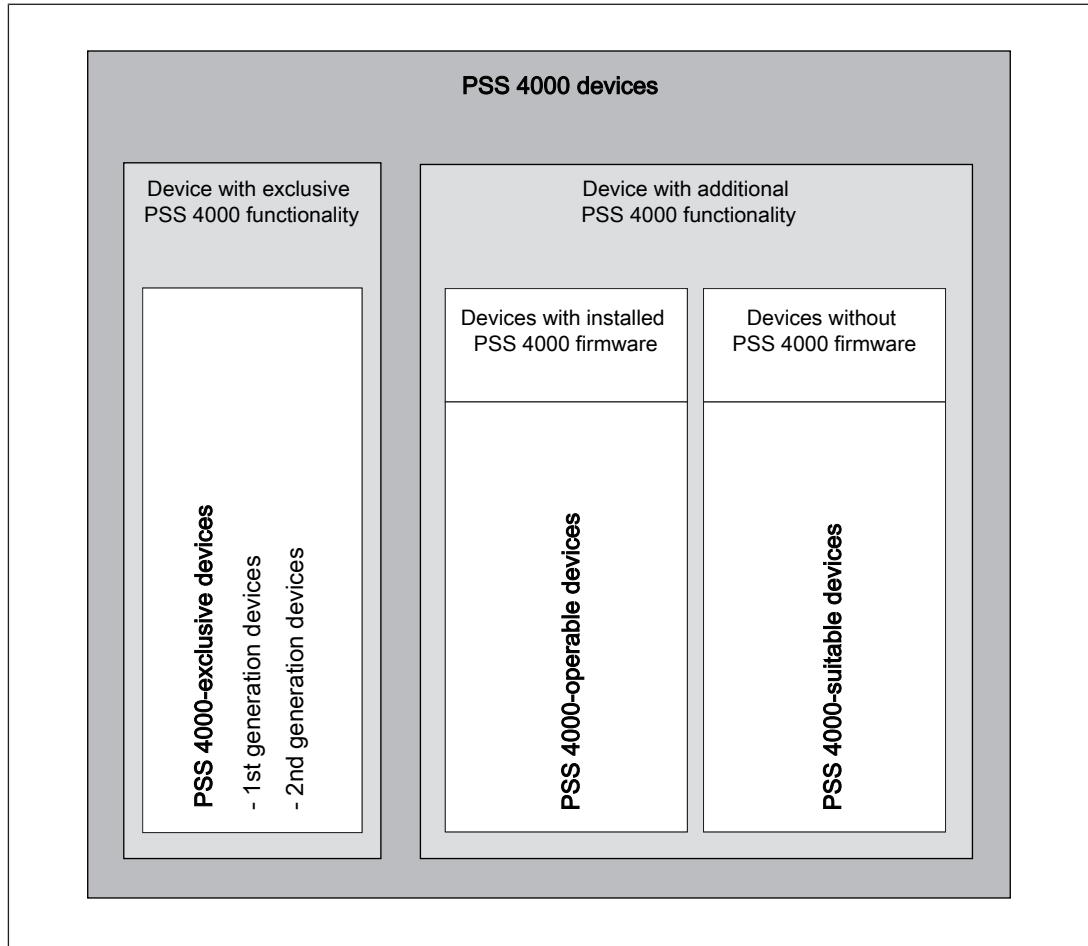


Fig.: Devices of the automation system PSS 4000

PSS 4000 devices are devices on which the PSS 4000 firmware is installed or on which the PSS 4000 firmware can be installed. As such, these devices have the basic properties of the automation system PSS 4000. The most important basic properties include:

- ▶ Ability to find the devices via PAS4000 using "Scan PSS 4000 Network"
- ▶ Incorporation into the PSS 4000 diagnostic system
- ▶ Respective configuration obtained via the PSS 4000 project download
- ▶ Respective device project obtained via the PSS 4000 project download

A further subdivision of the PSS 4000 devices is as follows:

- ▶ Devices that exclusively have PSS 4000 functionality (e.g. PSSu system head modules) are called **PSS 4000-exclusive devices**. PSS 4000-exclusive devices are divided into
 - 1st generation devices
 - 1st generation devices include PSSu systems and PSS67 devices.
 - 2nd generation devices

2nd generation devices include PSS u2 systems.

► Devices that also have PSS 4000 functionality are subdivided as follows:

- **PSS 4000-operable devices**

These are devices on which the PSS 4000 firmware is installed.

- **PSS 4000-suitable devices**

These are devices on which the PSS 4000 firmware is not installed, but on which the PSS 4000 firmware can be installed.

2.1.3.2 System sections

A system section is a defined unit of functionalities, which provides utilities on a device, which are visible to the user. To ensure that a system section can run on a PSS 4000-exclusive device or PSS 4000-operable device, the corresponding hardware requirements must be met.

The system sections that are available on a PSS 4000-exclusive device or PSS 4000-operable device depend on the device. Details of the PSS 4000-exclusive devices and PSS 4000-operable devices from Pilz are available in the operating manual for the relevant device.

Overview of available system sections

► Resources

- FS resource system section
Execution unit for safety-related control programs
- ST resource system section
Execution unit for non-safety-related control programs

► Module bus

- FS module bus system section PSSu
FS module bus of 1st generation devices; unit for data exchange between FS electronic modules/FS compact modules and the head module of a device
- ST module bus system section PSSu
ST module bus of 1st generation devices; unit for data exchange between ST electronic modules/ST compact modules and the head module of a device
- FS module bus system section PSS u2
FS module bus of 2nd generation devices; unit for data exchange between FS electronic modules and the head module of a device
- ST module bus system section PSS u2
ST module bus of 2nd generation devices; unit for data exchange between ST electronic modules and the head module of a device

► SafetyNET p

- FS SafetyNET p system section
Unit for transporting FS I/O data via SafetyNET p
- ST SafetyNET p system section
Unit for transporting ST I/O data via SafetyNET p

► External communication

- IP connections system section
Unit for transporting external I/O data via IP connections (e.g. Modbus/TCP, Raw TCP, Raw UDP)
- PROFIBUS-DP Slave system section
Unit for transporting external I/O data as PROFIBUS-DP Slave
- PROFIBUS-DP Slave with PROFIsafe system section

Unit for transporting non-safety-related external I/O data as PROFIBUS-DP Slave and safety-related external I/O data as PROFIsafe F-device

- EtherNet/IP Adapter system section

Unit for transporting non-safety-related external I/O data as EtherNet/IP Adapter

- PROFINET IO DEVICE system section

Unit for transporting external I/O data as PROFINET IO Device

► Server

- Diagnostic Server system section

Unit for providing diagnostic data

- System section OPC Server (classic)

Unit with standardised software interface in accordance with the standard "OPC Classic"; used to provide visualisation data

- System section OPC Server UA

Unit with standardised software interface in accordance with the standard "OPC UA"; used to provide visualisation data

Overview of the system sections of 1st generation devices

Product	FS resource	ST resource	FS module bus PSSu	ST module bus PSSu	FS SafetyNET p	ST SafetyNET p	IP connections	PROFIBUS-DP Slave	PROFIBUS-DP Slave with PROFIsafe	Ethernet/IP adapter	PROFINET IO DEVICE	Diagnostic Server	OPC Server UA
PSSu H PLC1 FS DP SN SD	◆	◆	◆	◆	◆	◆	◆	◆	---	---	---	---	---
PSSu H PLC1 FS SN SD (-T)(-R)	◆	◆	◆	◆	◆	◆	◆	---	---	◆ ¹⁾	◆ ¹⁾	---	---
PSSu H PLC2 FS SN SD (-T)	◆	◆	◆	◆	◆	◆	◆	---	---	◆ ¹⁾	◆ ¹⁾	---	◆ ²⁾
PSSu H PLC1 FS SN SD M12 (-T)(-R)	◆	◆	◆	◆	◆	◆	◆	---	---	◆ ¹⁾	◆ ¹⁾	---	---
PSSu H m F DP SN SD	◆	---	◆	◆	◆	◆	◆	◆	---	---	---	---	---
PSSu H m F DPsafe SN SD	◆	---	◆	◆	◆	◆	◆	---	◆	---	---	---	---
PSSu H m F DP ETH SD	◆	---	◆	◆	---	---	◆	◆	---	---	---	---	---
PSSu H FS SN SD(-T)(-R)	---	---	◆	◆	◆	◆	---	---	---	---	---	---	---
PSSu H FS2 SN SD(-T)	---	---	◆	◆	◆	◆	---	---	---	---	---	---	◆ ²⁾
PSSu H FS SN SD M12 (-T)(-R)	---	---	◆	◆	◆	◆	---	---	---	---	---	---	---
PSS67 PLC1 16FDI	◆	◆	◆	---	◆	◆	◆	---	---	◆ ¹⁾	◆ ¹⁾	---	---

Product	FS resource	ST resource	FS module bus PSSu	ST module bus PSSu	FS SafetyNET p	ST SafetyNET p	IP connections	PROFIBUS-DP Slave	PROFIBUS-DP Slave with PROFIsafe	Ethernet/IP adapter	PROFINET IO DEVICE	Diagnostic Server	OPC Server UA
PSS67 IO1 16FDI	---	---	◆	---	◆	---	---	---	---	---	---	---	---

¹⁾ Although both system sections for external communication are available on the device, only one system section can be used at a time.

²⁾ The system section OPC Server UA is only available on devices with a hardware version ≥ 2 .

Overview of the system sections of 2nd generation devices

Product	FS resource	ST resource	FS module bus PSS u2	ST module bus PSS u2	FS SafetyNET p	ST SafetyNET p	IP connections	Diagnostic Server	OPC Server UA
PSS u2 P42 F/S -R	◆	◆	◆	◆	◆	◆	◆	◆	◆
PSS u2 P12 F/S -R	◆	---	◆	◆	◆	◆	---	◆	◆

Overview of the system sections of PSS 4000-operable devices

Product	Diagnostic Server	OPC Server (classic)	OPC Server UA
PMI v8	◆	---	◆
PSS 4000 firmware for PC	◆	◆ ¹⁾	◆

¹⁾ A PSS 4000 project containing a 2nd generation device (PSS u2 system) may not contain a PSS 4000 OPC Server (classic).

For compatibility reasons, the system section PSS 4000 OPC Server (classic) is also included in the current PSS 4000 firmware versions, but use of the PSS 4000 OPC Server (classic) is not recommended in new projects.

To enable a PC, on which a PSS 4000 firmware version $\geq 1.25.0$ is installed, to execute a device project for a PSS 4000 OPC Server (classic), a firmware version $< 1.25.0$ must be configured in the project for the PC. This means that old projects can continue to be run on the PC. In new projects in which a firmware version $\geq 1.25.0$ is configured for the PC, a PSS 4000 OPC Server (classic) can no longer be configured on the PC.

2.1.3.3 PSS 4000-exclusive devices

PSS 4000-exclusive devices include 1st generation PSS 4000 devices (PSSu systems, PSS67 devices) and 2nd generation PSS 4000 devices (PSS u2 systems).

The system sections that are available on a device depend on that device (see [System sections](#) [22]). Details can also be found in the operating manual of the relevant device.

Performance classes

Available performance classes, depending on the design:

Design	Performance class			
	I/O	I/O+	multi	PLC
PSSu	x	---	x	x
PSS67	x	---	---	x
PSS u2	---	x	---	x

The performance class determines the basic functionality of a device. A distinction is made between the following performance classes:

► **PSS 4000 I/O (abbreviation: I/O)**

PSS 4000-exclusive device without control functionality for use as a decentralised I/O module. There are PSSu systems, for which the performance class "Decentralised system PSSu I/O" is documented. From the perspective of the PSS 4000, this corresponds to the performance class PSS 4000 I/O.

► **PSS 4000 I/O+ (abbreviation: I/O+):**

PSS 4000-exclusive device without control functionality for use as a decentralised I/O module with pre-processing of the I/O signals.

The performance class PSS 4000 I/O+ is only available on 2nd generation PSS 4000-exclusive devices (PSS u2 systems).

► **PSS 4000 multi (abbreviation: multi)**

PSS 4000-exclusive device with the functionality of a small safety controller

There are PSSu systems, for which the performance class "Control system PSSu multi" is documented. From the perspective of the PSS 4000, this corresponds to the performance class PSS 4000 multi.

► **PSS 4000 PLC (abbreviation: PLC)**

PSS 4000-exclusive device for controlling automation systems

There are PSSu systems, for which the performance class "Control system PSSu PLC" is documented. From the perspective of the PSS 4000, this corresponds to the performance class PSS 4000 PLC.

PSSu systems

A PSSu system is assembled from a variety of modules, to suit the functions required. The modular structure makes it easier to react to modifications and adjustments.

The scope of a PSSu system can also be adapted retrospectively, e.g. by

- Extending the number of I/Os
- Modifying the I/O technology
(e.g. by using relay outputs instead of semiconductor outputs)

- ▶ Adding I/O functions
(e.g. adding analogue evaluation, counter functions, temperature evaluation)
- ▶ By exchanging the head module, it is possible to convert a PSSu system with a lower performance class (e.g. PSS 4000 I/O) into a PSSu system with a higher performance class (e.g. PSS 4000 PLC).

PSSu systems can be assembled for

- ▶ Safety-related tasks
Module selection must be based on failsafe modules (FS modules).
- ▶ Non-safety-related tasks
Standard modules (ST modules) may be selected.
- ▶ Safety-related and non-safety-related tasks
The PSS u2 system can be assembled using a mixture of FS and ST modules.

A PSSu system is configured in the PAS4000's PSSu Module Editor.

PSS67 devices

In principle, a PSS67 device is a PSSu system in compact design for IP67 applications. As it does not have a modular design, it is not possible to adapt the function range retrospectively.

A PSS67 device is configured in the PSS67 editor of the PAS4000.

PSS u2 systems

A PSS u2 system is assembled from a variety of modules, to suit the functions required. The modular structure makes it easier to react to modifications and adjustments. With a PSS u2 system, it is also possible to adapt the function range retrospectively, for example by

- ▶ Extending the number of I/Os
- ▶ Modifying the I/O technology
(e.g. by using relay outputs instead of semiconductor outputs)
- ▶ By exchanging the head module, it is possible to convert a PSS u2 system with a lower performance class (e.g. PSS 4000 I/O+) into a PSS u2 system with a higher performance class (e.g. PSS 4000 PLC).

PSS u2 systems can be assembled for

- ▶ Safety-related tasks
Module selection must be based on failsafe modules (FS modules).
- ▶ Non-safety-related tasks
Standard modules (ST modules) may be selected.
- ▶ Safety-related and non-safety-related tasks
The PSS u2 system can be assembled using a mixture of FS and ST modules.

A PSS u2 system is configured in the PAS4000's PSS u2 Module Editor.

2.1.3.4 PSS 4000-operable devices

The PSS 4000 firmware is pre-installed on various operator terminals PMI from Pilz in their delivery state. Due to the pre-installed PSS 4000 firmware, these operator terminals are counted among the "PSS 4000-operable devices" from Pilz. As a result, these operator terminals have the system sections OPC Server UA and diagnostic server, for example.

A PC with the corresponding system requirements can be made into a "PSS 4000-operable device" from a third-party manufacturer by installing the PSS 4000 firmware. In this case the PC will also be known as a PSS 4000-PC. When the PSS 4000 firmware is installed, the PC obtains the system sections OPC Server UA and Diagnostic Server, for example (see [System sections available](#) [ 23]).

2.1.3.5 PSS 4000-suitable devices

A PSS 4000-suitable device must meet certain system requirements in order for the PSS 4000 firmware to be installed. These system requirements are defined through the PSS 4000 firmware that is to be installed on the PSS 4000-suitable device.

Example for PSS 4000 OPC Server:

- The PSS 4000 OPC Server can be installed on a PC with a Microsoft Windows operating system. You'll find the system requirements for installation on a PC under PC with Microsoft Windows operating system.

2.1.4 PAS4000

Design, configuration, programming, commissioning and diagnostics take place centrally via PAS4000 (Pilz Automation Suite 4000).

PAS4000 is a collection of various tools, which can be used to create and edit a project, e.g.:

- ▶ PSSu Module Editor

The required PSSu systems are assembled and configured in the PSSu Module Editor.

- ▶ PSS u2 Module Editor

The required PSS u2 systems are assembled and configured in the PSS u2 Module Editor.

- ▶ I/O Mapping Editor

The connection between the user program and hardware is established in the I/O Mapping Editor.

- ▶ IL Editor

In the IL Editor, the user program is programmed in the instruction list (IL) programming language.

2.1.5 Diagnostics

Effective diagnostics are required in order to keep machine downtimes to a minimum. Diagnostics must provide all the information necessary to recover errors quickly, prevent errors and analyse machine conditions.

The PSS 4000 distinguishes between system diagnostics and process diagnostics.

► System diagnostics

Diagnostics on hardware and firmware, including errors in the hardware configuration or user program.

The system diagnostics are provided by Pilz in full. Only a few adjustments can be made by the user, e.g. entering additional information for location information.

► Process diagnostics

Diagnostics on procedures and states. The process diagnostics are created by the user within the user program. Pilz blocks are supplied with process diagnostics, which users can adapt to suit their own requirements.

A diagnostic server is required if you wish to display a project's diagnostic data (see [Diagnostic Server](#)  739). A project may only contain one Diagnostic Server.

If the project contains an OPC Server (PSS 4000 OPC Server), the PSS 4000 device on which the OPC Server is installed will provide the Diagnostic Server. If there is no OPC Server, PAS4000 will start up its own Diagnostic Server if the diagnostic list or diagnostic log is to be displayed in PAS4000.

2.1.6 OPC Server

OPC is a standardised software interface, through which PC-based software from a range of manufacturers can exchange data. The OPC Foundation is responsible for maintaining and propagating the standard.

On networked plants, an OPC Server will typically collect diagnostic data, process data etc. and display it in the defined namespace. OPC Clients access and process the data provided by the OPC Server for process visualisation purposes and/or to record operating data.

Various OPC Servers can be used in the automation system PSS 4000.

- ▶ PC or PMI with PSS 4000 firmware

A PC or operator terminal PMI can obtain the OPC Server system section as part of the installation of the PSS 4000 firmware. By installing the PSS 4000 firmware, a PC becomes a "PSS 4000-operable device from a third-party manufacturer" and a PMI operator terminal becomes a "PSS 4000-operable device from Pilz" (see [System overview](#)  20]).

- ▶ PSS 4000-exclusive device with device-internal OPC UA Server

With a PSS 4000-exclusive device, a device-internal OPC UA Server is available if the device has the system section OPC UA Server (see [System sections](#)  22]).

In the automation system PSS 4000, a distinction is made between the two OPC standards "OPC Classic" and "OPC UA".

2.2 Modularity with PSS 4000

2.2.1 Centralised view of a networked system

PSS 4000 combines the benefits of a centralised controller at the highest automation level with the benefits of small, decentralised control units.

- ▶ Benefits at the highest automation level:
 - Software tool PAS4000, which can be used from a central location
 - Central configuration, including for decentralised, distributed PSS 4000-exclusive devices
 - A user program created centrally is distributed to several PSS 4000-exclusive devices
 - Rapid overview of the entire plant during troubleshooting, commissioning/re-commissioning and modification
- ▶ Benefits at the lowest automation level:
 - Decentralised function units can be commissioned independently
 - Increased availability (no "single point of failure")

2.2.2 Scalability of the devices

The term "scalability" is used for a device when it has a certain degree of adaptability to changing requirements.

A distinction can be made between "horizontal" and "vertical" scalability:

► Horizontal scaling

Functions are distributed to other devices or additional devices. On PSS 4000, for example, horizontal scaling can be used to:

- Distribute the control power to various PSS 4000-exclusive devices on a networked plant, to increase the processing power, for example
- Divide the user program into independent subfunctions, in order to distribute it to the project's PSS 4000-exclusive devices
- Divide the project into reusable parts

► Vertical scaling

A device or parts of a device are replaced by a more powerful alternative. On PSS 4000, for example, vertical scaling can be used to:

- Adapt the performance class of a PSSu system retrospectively by using a head module of a different performance class
- Retrospectively expand and adapt a modular PSS 4000-exclusive device with I/O modules

2.2.3 Modularisation within programming

On PSS 4000, control tasks can be distributed across an entire plant. As a result, the plant architecture can be modularised in line with the task assignment. This modularisation applies not only to the modular structure of PSS 4000-exclusive devices and the network but also to the user program. With component-oriented programming, the plant can be completely divided into functionalities. The aim is to achieve the highest possible level of standardisation and reusability while maintaining maximum flexibility and adaptability.

With a modularised user program you can achieve the following on PSS 4000, for example:

- ▶ Modifications will only impact locally
- ▶ Control tasks are standardised

A user program can be adopted for all functions with an identical structure; and once a modification has been made, it can be transferred to all relevant devices.

- ▶ Availability is increased, as error reactions are restricted locally

3 Safety

3.1 Additional documents that apply

Please refer to the safety guidelines in the operating manual for the respective products, and in the "Safety Manual PSS 4000". The "Safety Manual" also includes check lists designed to help you with the safety-related planning, construction and operation of a plant.

You should also refer to the information provided in the following documents:

- ▶ When using 1st generation devices (PSSu systems, PSS67 devices):
Installation Manual PSSuniversal
- ▶ When using 2nd generation devices (PSS u2 systems):
Installation Manual PSS u2 in the automation system PSS 4000
- ▶ PAS4000 Online Help

The standards current at the time of going to print shall apply.

3.2 Intended use

PSS 4000-exclusive devices of all performance classes are designed for use as follows:

- ▶ Safety-related applications via the FS resource of a PSS 4000-exclusive device, including connection to SafetyNET p for decentralised tasks
 - Performance class PSS 4000 I/O+
for use as a decentralised I/O module with pre-processing of the I/O signals
 - Performance class PSS 4000 multi and PSS 4000 PLC
with control functionality
- ▶ Non-safety-related applications via the ST resource of a PSS 4000-exclusive device of performance class PSS 4000 PLC, including connection to SafetyNET p and/or other standard bus systems for decentralised tasks.

The PSS 4000-exclusive devices of all performance classes are primarily suitable for use in machine safety circuits in which a safe condition is brought about by the removal of power.

Examples:

- ▶ Presses
- ▶ Transfer lines
- ▶ Tank storage facilities
- ▶ E-STOP functions
- ▶ Burner control systems
- ▶ Cable cars/traction operations
- ▶ Stage technology

The following is deemed improper use in particular:

- ▶ Any component, technical or electrical modification to a product
- ▶ Use of a product outside the areas described in the product documentation
- ▶ Any use that is not in accordance with the documented technical details.

Foreseeable misuse and abuse

- ▶ Without additional measures, the automation system PSS 4000 is not suitable for use in areas with increased environmental requirements (e.g. potentially explosive areas).
- ▶ The automation system PSS 4000 is not suitable for plants in which the removal of power does not lead to a safe condition.
- ▶ On passenger transportation systems, appropriate evacuation measures must be made or self-contained rescue equipment installed, should the plant come to a standstill due to a fault.
- ▶ Where locked zones have an electrical release, emergency release devices should be provided.
- ▶ Safety devices may not be overridden until appropriate measures of equal value are put in place (e.g. muting function). Measures of equal value may only be used if the preceding safety assessment permits it.

Examples:

- The safety gates may be open in setup mode provided there is a muting function, which is triggered by the operating mode selector switch.

- A light curtain can be interrupted by transported materials provided there is a muting function with the corresponding components (e.g. muting sensors including control and muting monitoring of the muting sensors via a muting block in the user program, for example).

3.2.1 Safety Integrity Level (SIL) and Performance Level (PL)

The PSS 4000-exclusive devices of all performance classes are safety components in accordance with Annex V of the Machinery Directive 2006/42/EC. Depending on the application area and its respective regulations, PSS 4000-exclusive devices of all performance classes can be used up to SIL 3 of EN 62061 and up to PL e (Cat. 4) of EN ISO 13849-1:2008. The relevant technical details for a risk assessment can be found in the respective operating manual.

SafetyNET p is used for networked automation functions. This safe communication was developed in accordance with relevant standards, such as EN IEC 61508. Independent certification bodies such as BG have confirmed the security of the concept and the suitability of SafetyNET p in safety-related applications up to PL e (Cat. 4) of EN ISO 13849-1 or SIL 3 of EN IEC 62061. The relevant technical details for the modules, as required for the risk assessment, can be found in the operating manual for the respective module.

3.2.2 General safety assessments

Before using a PSS 4000-exclusive device it is necessary to perform a safety assessment in accordance with the Machinery Directive. A PSS 4000-exclusive device as an individual component is a safety-related system for the purposes of EN/IEC 61508. It guarantees functional safety against hardware and firmware errors, for example. However, it does not guarantee the safety of the overall process and design, nor of the project.

The user is responsible for the safety of the project. Pay special attention when programming and observe local standards and regulations.

A faulty user program can jeopardize the safety of the entire process!

Define the safety requirements for the entire plant, for all phases of the service life and the entire safety lifecycle, and also define how these will be implemented from a technical and organisational standpoint.

3.2.3 Third-party manufacturer licence information

The products contain open source software, whose terms of use could further limit the products' application area. It is essential that you observe the third-party manufacturer licence information.

Information is available in the operating manual for the respective product.

3.3 Use of qualified personnel

The products may only be assembled, installed, programmed, commissioned, operated, de-commissioned and maintained by persons who are competent to do so.

A competent person is a qualified and knowledgeable person who, because of their training, experience and current professional activity, has the specialist knowledge required. In order to inspect, assess and handle products, devices, systems, plant and machinery, this person must be familiar with the state of the art and the applicable national, European and international laws, directives and standards.

It is the company's responsibility only to employ personnel who

- ▶ Are familiar with the basic regulations concerning health and safety / accident prevention,
- ▶ Have read and understood the information provided in the section entitled Safety
- ▶ Have a good knowledge of the generic and specialist standards applicable to the specific application.

3.4 Failsafe inputs/outputs

In FS applications, it is essential that short circuits and open circuits are unable to cause a hazardous condition within a plant/machine.

Various tests may be configured on FS input/output modules. These tests can be used to detect many short circuits and open circuits. Examples of such tests are:

- ▶ Off tests for outputs that are switched on
- ▶ On tests for outputs that are switched off
- ▶ Tests for shorts between the outputs
- ▶ Test pulses for detecting shorts between contacts on inputs

Wiring guidelines and connection examples are available in the operating manuals or data sheets for the input/output modules.

4 Security

The term "security" refers to information security in plants, systems, machines and networks. This includes protection from external attacks (remote access) as well as protection from attacks from insiders. Measures are taken that protect the integrity and confidentiality of sensitive data and guarantee plant/machine availability. So all measures that concern the integrity, availability, confidentiality and authenticity of data fall under the term "security" (see also the IEC 62443 series of standards).

Planning and implementation

Securing plants, systems, machines and networks against cyberthreats requires a holistic industrial security concept that is state of the art and is continuously developed further. All plants, systems, machinery and networks that require protecting must be assessed individually. The assessment must take into account organisational as well as technical measures. If necessary, seek advice from Technical Support.

Planning and implementation should be carried out in accordance with a defined security process. For example, your security process should take the following into account:

- ▶ Note the regulatory specifications and requirements from the locally applicable safety and application standards with regard to protection from manipulation.
- ▶ Perform a risk analysis (e.g. in accordance with IEC 62443-3-2) and then plan the resulting security measures carefully.
- ▶ Where possible, counter the security risks using a multi-layered concept (e.g. defence-in-depth concept).
- ▶ Name the necessary security measures and document them.
- ▶ Check the security measures after implementation.
- ▶ Carry out a new risk analysis after every change and adapt the security measures if necessary.
- ▶ It is advisable to review the security measures regularly, for example annually.

Defence-in-depth concept

In accordance with IEC 62443-4-1, the defence-in-depth concept is an approach in which multiple layers of independent security methods are used.

The defence-in-depth concept is based on the following rationale:

- ▶ Every protective measure can be overcome, and probably will be overcome.
- ▶ A multi-layer structure (e.g. onion model) implements several concerted and coordinated measures.
- ▶ Due to independent defence mechanisms in each layer, successful exploitation of a weakness in one layer is prevented, or made more difficult, by a measure in another layer.
- ▶ Generally with the defence-in-depth concept, an attack is made more difficult because the attacker has to circumvent different, independent security measures, one after the other.

The standard series IEC 62443 distinguishes between the following three areas of responsibility:

- ▶ Manufacturer
- ▶ System integrator

► Operator

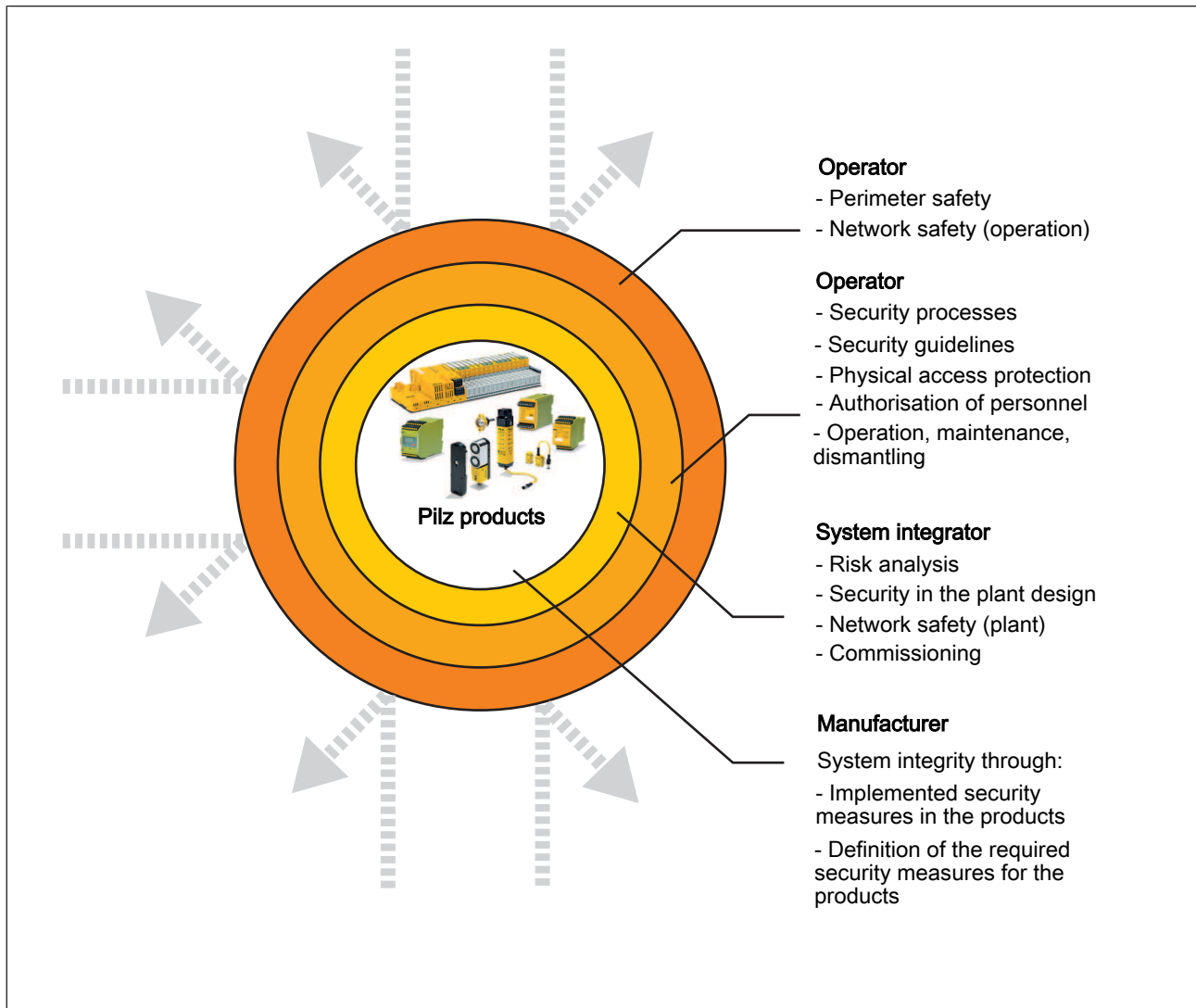


Fig.: Roles and responsibilities in the defence-in-depth onion model

Security incident management

Should the product have a security vulnerability, this will be published in a security advisory. You'll find the security advisories under www.pilz.com/security. You can also report any security problems there.

Implemented and required security measures

The section below describes the implemented and required security measures. To protect the devices in the PSS 4000 project, the tool PAS4000 and any data/projects created using the tool from cyber threats, it is essential that you use the implemented security measures and implement the required security measures.

4.1 Generally required security measures

- ▶ Ensure that all devices and software tools that exchange data with the devices in the PSS 4000 project are not compromised. These devices and software tools include, for example:
 - Controllers from third-party manufacturers, which exchange data with devices in the PSS 4000 project via system sections for external communication
 - Computers, including OPC Clients
 - Computers on which the PSS 4000 OPC Server UA is installed
 - Device that is used as the SNTP Server
- ▶ On devices and software tools without an automatic update function, you should check regularly to see if updates are available from the respective manufacturer.
- ▶ Install updates for all devices and software tools promptly. This applies in particular when security weaknesses have been rectified.
- ▶ Take measures to identify any non-authorised physical access to devices, e.g. mechanical interlocking of unassigned interfaces.
- ▶ Ensure computers are handled securely, e.g. screen locked when absent, use of passwords.
- ▶ Only assign strong passwords and handle the passwords carefully. Be guided by generally accepted guidelines such as NIST 800-63b for example.
- ▶ Only use adequately protected storage media to store and load data.
- ▶ If possible, HMIs should be set up so that each user must authenticate themselves.
- ▶ Systems (e.g. PCs, PMIs) should be set up so that the user cannot gain access to the operating system level (kiosk mode).

4.2 Security measures on PSS 4000-exclusive devices

Implemented security measures

- ▶ No firmware downgrades are allowed. This prevents outdated firmware with vulnerabilities from being loaded onto the product.
- ▶ The PSS 4000 firmware for the PSS u2 head modules, the PSSu head module PSSu H FS2 SN SD(-T) and the PSSu head module PSSu H PLC2 FS SN SD(-T) is digitally signed. The PSS 4000 firmware for all other PSSu head modules is not signed.
 - Significance for PSSu head modules PSSu H FS2 SN SD(-T) and PSSu H PLC2 FS SN SD(-T) with hardware version ≥ 2.0 and PSS u2 head modules:

During a firmware update it is established whether the firmware has been manipulated. If the firmware has been manipulated, it is seen as defective. The firmware update will not be performed and a corresponding diagnostic message will be issued. Also, the firmware is re-checked each time the system sections are booted. Manipulated firmware will not be executed.
 - Significance for PSSu head modules PSSu H FS2 SN SD(-T) and PSSu H PLC2 FS SN SD(-T) with hardware version < 2.0 :

During a firmware update it is established whether the firmware has been manipulated. If the firmware has been manipulated, it is seen as defective. The firmware update will not be performed and a corresponding diagnostic message will be issued.

Required security measures

- ▶ The product is not protected from physical manipulation or from reading of memory contents during physical access. Use appropriate measures to ensure that there is no physical access by unauthorised persons. You should also use security seals so that you can detect any manipulation of the product or interfaces. Installation inside a lockable control cabinet is recommended as a minimum measure.
- ▶ Ensure that the product cannot be compromised on any section of the route to the final point of use.
- ▶ With external communication (e.g. IP connections), data exchange on external devices is configured using third-party tools. Ensure that these tools can only be used by authenticated, authorised users.
- ▶ At www.pilz.com/security, you should check regularly to see if a new security advisory has been published for the product and take the required countermeasures.
- ▶ Any updates provided by Pilz should be installed promptly (e.g. firmware updates for head modules, electronic modules). This applies in particular when security weaknesses have been rectified.
- ▶ If you decommission a PSS 4000-exclusive device on which the integrated OPC UA Server was used, perform an original reset beforehand. This will ensure that all certificates and keys stored on the device are deleted and cannot be misused.

4.3 Security measures on the tool PAS4000

Implemented security measures

- ▶ The installer for the tool PAS4000 is digitally signed.
- ▶ If a password is changed, you will be prompted to enter the old password for authentication.
- ▶ Passwords are stored encrypted.

Required security measures

- ▶ The tool PAS4000 is a Windows tool. To protect the tool and any data/projects created using the tool from cyber threats, it is essential that you use the security measures provided by Windows.
 - Set up an account for each user on the Windows computer.
 - Install the tool with administrator rights, but use the tool as a standard user (without administrator rights).
 - Ensure that Windows updates are installed promptly, e.g. by activating the automatic update function.
 - Use a password to protect your user account.
 - By default, the tool saves data/projects in the Windows personal user directory, e.g. C:\User\<User name>. As a result, only this user has access to the data. Please note that saving to a different path may lead to security restrictions.
- ▶ At www.pilz.com/security, you should check regularly to see if a new security advisory has been published for the tool PAS4000 and take the required countermeasures.
- ▶ Any updates that Pilz provides for the tool PAS4000 should be installed promptly. This applies in particular when security vulnerabilities have been rectified.

4.4 Security measures for the project

Implemented security measures

- ▶ User management

In user management you can create multiple users and define which offline and online operations they may perform. The user must sign in to a project in order to access it.

- ▶ Protection of objects within the project

Blocks and derived data types can be write-protected, read/write protected and sealed.

- ▶ Check sums for objects within the project

Objects within the project are protected by check sums.

- ▶ Bind device projects to devices

You can use this function to ensure that projects can only be executed on prepared devices.

- ▶ Bind device projects to SD cards

You can use this function to ensure that a project cannot be copied from one SD card to another.

- ▶ Protection of files in the project directory

Files within the project directory are protected by check sums. Any changes to the files that are executed outside of PAS4000 will be detected. The project can no longer be activated.

Required security measures

- ▶ Only store projects and backup copies on an adequately protected storage medium. Only forward projects and backup copies to trustworthy people and use secure communication routes.

4.5 Secure network

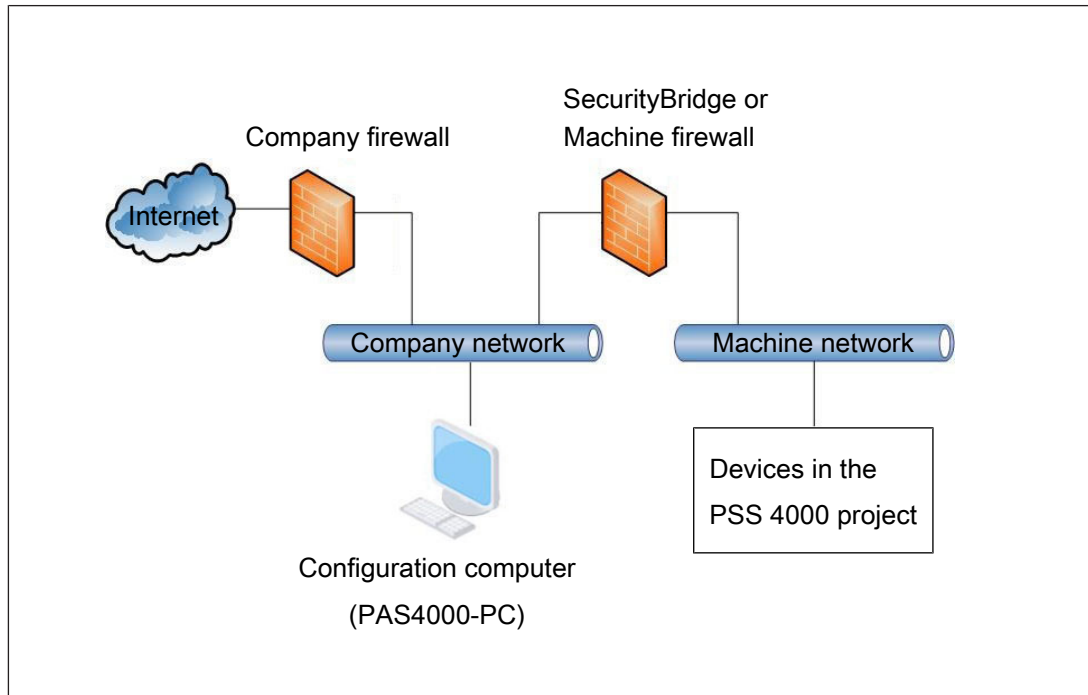


Fig.: Example of network overview

Required security measures

- ▶ With Ethernet-based networks, a logical and functional separation of the office environment (company network) and automation environment (machine network) is recommended. This can be achieved by a firewall or Security Bridge, for example.
- ▶ The computer on which the tool PAS4000 is installed and with which the devices are accessed must be protected from cyber attacks via a firewall or other appropriate measures. The installation of an appropriate solution to detect and protect against malware is recommended (keyword "endpoint protection"); this may be a virus scanner, for example.
- ▶ The installation of an appropriate solution to detect and protect against malware is also recommended on computers and PMIs on which a PSS 4000 OPC Server is installed (keyword "endpoint protection"); this may be a virus scanner, for example.
- ▶ For PCs and PMI operator terminals with PSS 4000 firmware, note the [Network data \(OPC Server \(classic\)\)](#) [669] and [Network data \(OPC Server UA\)](#) [702].
- ▶ For PSS 4000-exclusive devices, note the network data in the operating manuals for the head modules.
- ▶ Note the [network data](#) [669] in this document.
- ▶ Only allow the data exchange that's required for the application. Any data exchange that is not required for the application must be prevented by the relevant firewall.

5 PSSu system basics

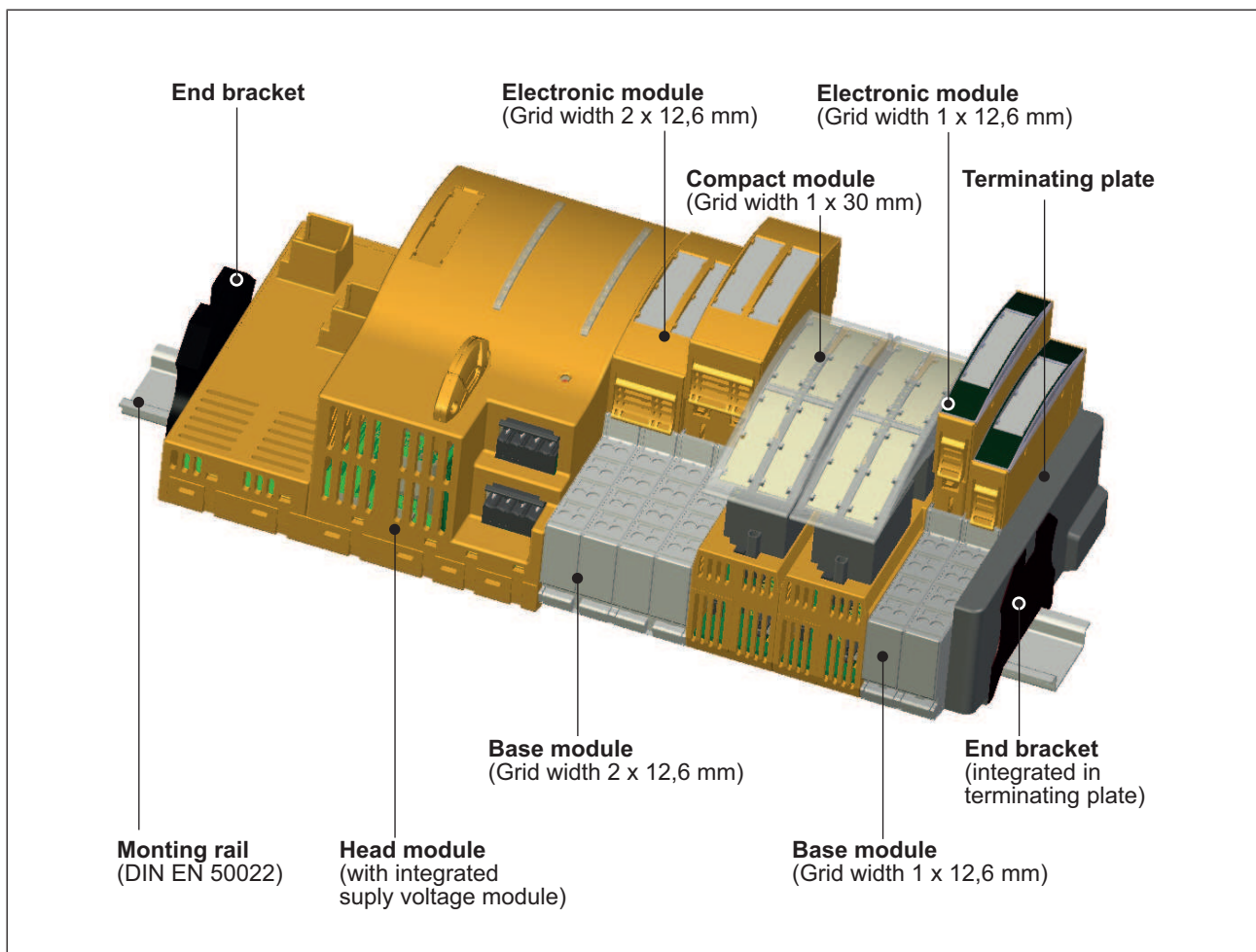
PSSu systems are PSS 4000-exclusive devices. A PSSu system is a PSS 4000-exclusive device in a modular design.

The system sections that are available on a device depend on that device (see [System sections](#)  22]). Details can also be found in the operating manual of the relevant device.

5.1 Structure

A PSSu system consists of:

- ▶ A head module with integrated supply voltage
- ▶ Input/output modules for standard and failsafe applications
- ▶ Supply voltage modules (only if necessary)
- ▶ End bracket to secure the system at the start
- ▶ Terminating plate with integrated bus terminating resistors and end bracket to secure the system at the end



Head module

The head module co-ordinates a PSSu system's entire data traffic and determines its performance class.

The performance class determines the basic functionality of a device. A distinction is made between the following performance classes:

- ▶ PSS 4000 I/O (abbreviation: I/O)

PSS 4000-exclusive device without control functionality for use as a decentralised I/O module. There are PSSu systems, for which the performance class "Decentralised system PSSu I/O" is documented. From the perspective of the PSS 4000, this corresponds to the performance class PSS 4000 I/O.

- ▶ PSS 4000 I/O+ (abbreviation: I/O+):
PSS 4000-exclusive device without control functionality for use as a decentralised I/O module with pre-processing of the I/O signals.
The performance class PSS 4000 I/O+ is only available on 2nd generation PSS 4000-exclusive devices (PSS u2 systems).
- ▶ PSS 4000 multi (abbreviation: multi)
PSS 4000-exclusive device with the functionality of a small safety controller
There are PSSu systems, for which the performance class "Control system PSSu multi" is documented. From the perspective of the PSS 4000, this corresponds to the performance class PSS 4000 multi.
- ▶ PSS 4000 PLC (abbreviation: PLC)
PSS 4000-exclusive device for controlling automation systems
There are PSSu systems, for which the performance class "Control system PSSu PLC" is documented. From the perspective of the PSS 4000, this corresponds to the performance class PSS 4000 PLC.

Features of a PSSu system, depending on the performance class:

- ▶ PSSu system of performance class PSS 4000 I/O
 - Consists of head module, electronic modules and/or compact modules
 - Control of I/Os via SafetyNET p through a control system of performance class PSS 4000 PLC or PSS 4000 multi.
- ▶ PSSu system of performance class PSS 4000 multi
 - Consists of head module, electronic modules and/or compact modules
 - FS resource available
 - One task only can be configured
 - Programmed via the PAS 4000 Multi editor
- ▶ PSSu system of performance class PSS 4000 PLC
 - Consists of head module, electronic modules and/or compact modules; however, the electronic modules/compact modules are not needed if the PSSu system is used exclusively to control decentralised I/Os with SafetyNET p
 - Control of I/Os from decentralised PSS 4000 systems of all performance classes when networking with SafetyNET p
 - ST resource and FS resource available
 - Max. of 9 tasks can be configured per resource
 - Programmed via the PAS4000 Multi editor or programmed in PAS4000 in accordance with IEC 61131

Input/output modules

Input/output modules are available in different designs:

- ▶ Electronic module and base module
- ▶ Compact module

The input/output modules are available for a wide range of input/output functions.

Supply voltage modules

The head module provides the supply voltage for the remaining modules. Additional power supplies may also be required (e.g. potential isolation, voltage refresh).

Supply voltage modules consist of an electronic module and base module.

Designs

Electronic modules and base modules:

The electronic modules determine the function of the supply voltage or input/output module.

► **Electronic modules**

- Are plugged into the base modules
- Communicate with the head module via the module bus

The base modules are the carrier units for the electronic modules.

► **Base modules**

- Are used to connect the field wiring
- Are available with the following connection types:
 - Cage clamp terminals or screw terminals

Compact modules:

The compact modules combine the function unit (inputs and/or outputs) and connection levels in one housing. Wiring is via multi-pin connectors with spring-loaded terminals, which are plugged into the connector strips on the module.

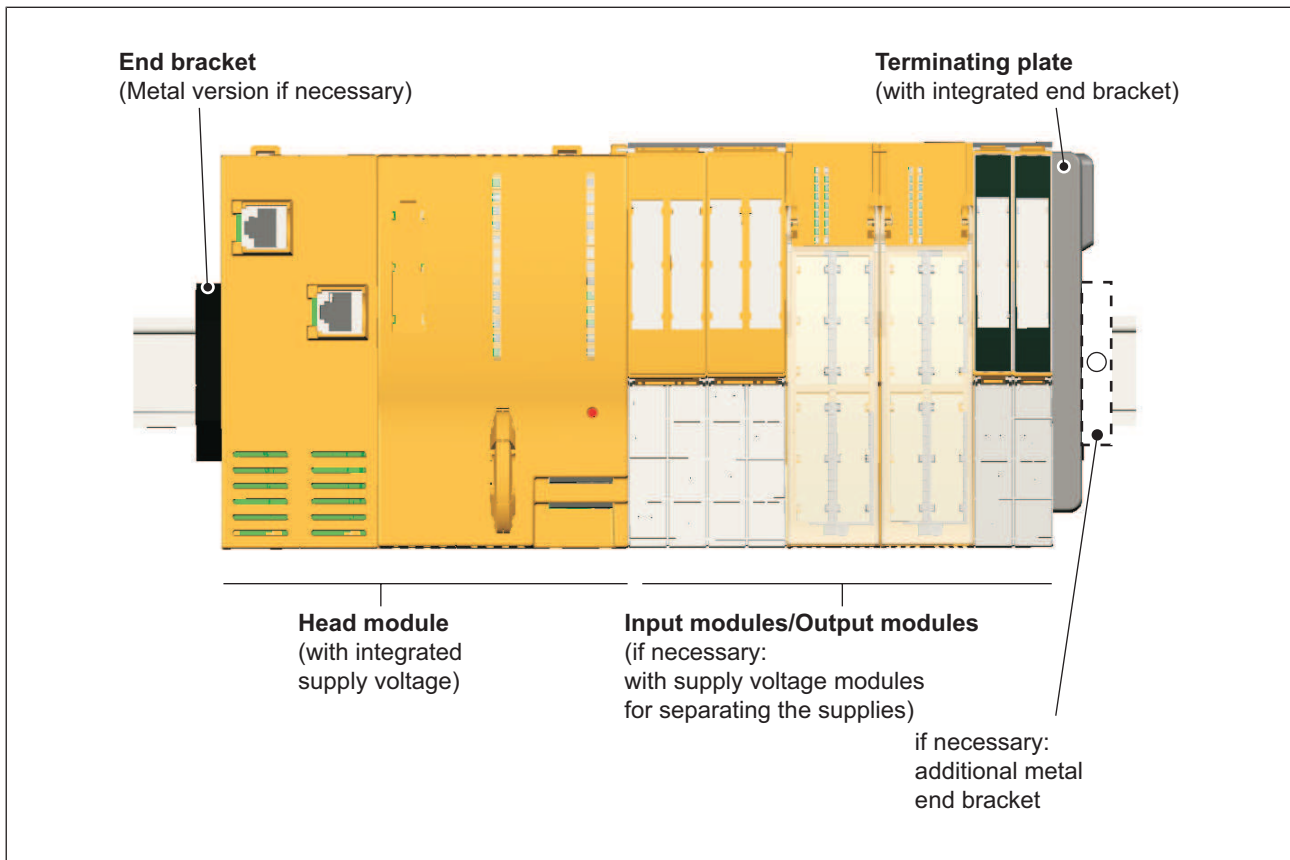
Compact modules

- **Do not need base modules**
- Are used to connect the field wiring
- Communicate with the head module via the module bus

5.2 Module layout

- ▶ The first module on a PSSu system is always a head module. The supply voltage is integrated within the head module:
 - Module supply (to supply the head module and input/output modules)
 - Periphery supply (to supply the sensors and actuators)
- ▶ Input/output modules for FS and ST applications may follow on to the right:
 - The input/output modules may be installed in any order.
 - Input/output modules for FS applications and ST applications may be combined as required.
 - If modules of the same function are combined into groups it makes things clearer and simplifies the wiring.
 - Base modules with screw terminals cannot be mixed in one PSSu system with base modules with cage clamp terminals.
 - A base module with screw terminals may not be installed to the left of a compact module.
 - The maximum number of input/output modules is determined by the system limits.
 - Input/output modules for FS applications may not be used with head modules that are only designed for ST applications.
- ▶ Additional supply voltage modules may be required in order to refresh the module supply or periphery supply.
- ▶ Supply groups require an additional voltage supply module at the start of each group. The modules belonging to the supply group will then follow to the right.
- ▶ The final element in a PSSu system is always a terminating plate containing the terminating resistors for the module bus.
- ▶ The system is attached to the mounting rail using fastening elements at the start and end of the system. The following end brackets are available, depending on vibration and shock stress:
 - Plastic version (standard)
 - Metal version (where there is increased stress)

Layout of the fastening elements



5.3 Module names

The module names provide information on their function. The names are made up of multi-stage combinations of letters and numbers. All module names begin with **PSSu**. This is followed by:

- ▶ **H** for head modules,
e.g.: PSSu **H** PLC1 FS DP SN SD
- ▶ **E** for electronic modules, then:
 - **S** for standard module,
e.g.: PSSu **E S** 4DI
 - **F** for failsafe module,
e.g.: PSSu **E F** 4DI
- ▶ **K** for compact modules, then:
 - **S** for standard module,
e.g.: PSSu **K S** 16DI
 - **F** for failsafe module,
e.g.: PSSu **K F** 16DI
- ▶ **B** for base modules,
e.g.: PSSu **BP** 1/8S

Base modules	Basic function	Size	Connection type	Additional function
PSSu B...	PSSu BP... – Base module for periphery (input/output) PSSu BP-C... – Base module for periphery with C-rail PSSu BS... – Base module for power supply PSSu BS-R... – Base module for refreshing the power supply	PSSu BP-C 1/8... – 1 x grid width, 8 connections PSSu BP-C 1/12... – 1 x grid width, 12 connections PSSu BP-C 2/16... – 2 x grid width, 16 connections PSSu BP-C 2/8... – 2 x grid width, 8 connections	PSSu BP-C 1/8C – Cage clamp terminals PSSu BP-C 1/8S – Screw terminals	PSSu BP 1/8S-J – Integrated cold junction compensation

Link modules have individual letter codes, depending on their application, e. g.:

- ▶ PSSu **XB** F-T, PSSu **XR** F-T
(for cable-based subdivision)

Accessories for the PSSu system are identified by **PSSu A...** (A for "Accessory").

PSSu modules are available as different product types:

- ▶ Base type
The basic versions of PSSu modules are called base type modules (e.g. PSSu E F DI OZ 2).

► T-type

PSSu modules that are suitable for use where there are increased environmental requirements on temperature and humidity are called T-type modules. Their functionality is no different from that of the base type modules. They are identified by a "**-T**" at the end of the product name, e.g.: PSSu E F DI OZ 2-**T**.

► R-type

PSSu modules that are specifically designed for railway applications and have railway approval are called R-type modules. They are identified by an "**-R**" in the product name (e.g. PSSu E F DI OZ 2-**R**).

5.4 Module bus PSSu

On the module bus, the data is transferred between head modules and electronic modules. The module bus is formed by arranging the base modules/compact modules together and connecting them via a mechanical latch.

The module bus consists of two separate bus systems:

- ▶ FS module bus
for transferring failsafe data
- ▶ ST module bus
for transferring standard data

5.5 Base modules

The base modules are the carrier units for the electronic modules. Base modules are structured as follows:

► **Module bus**

Two separate bus systems for failsafe data (FS module bus) and standard data (ST module bus).

► **Supply voltage for:**

– **Module supply**

The module supply is the internal supply voltage for the head module, compact modules and electronic modules.

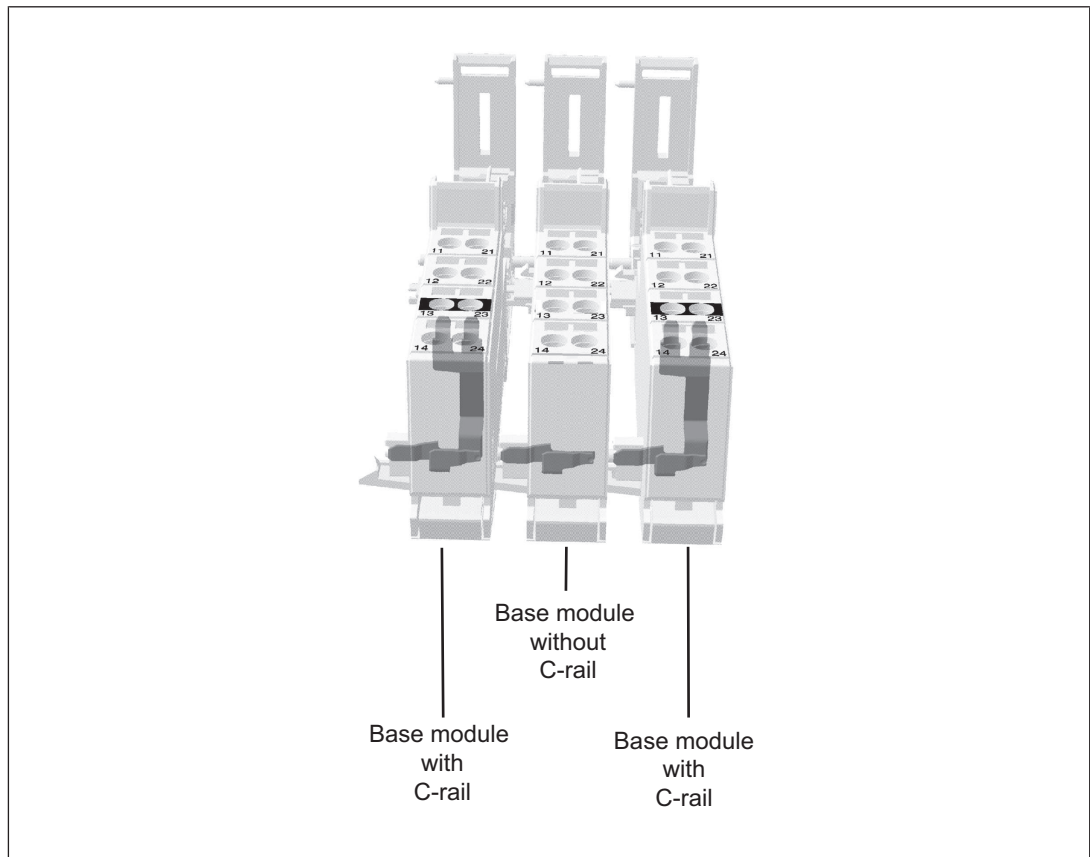
– **Periphery supply**

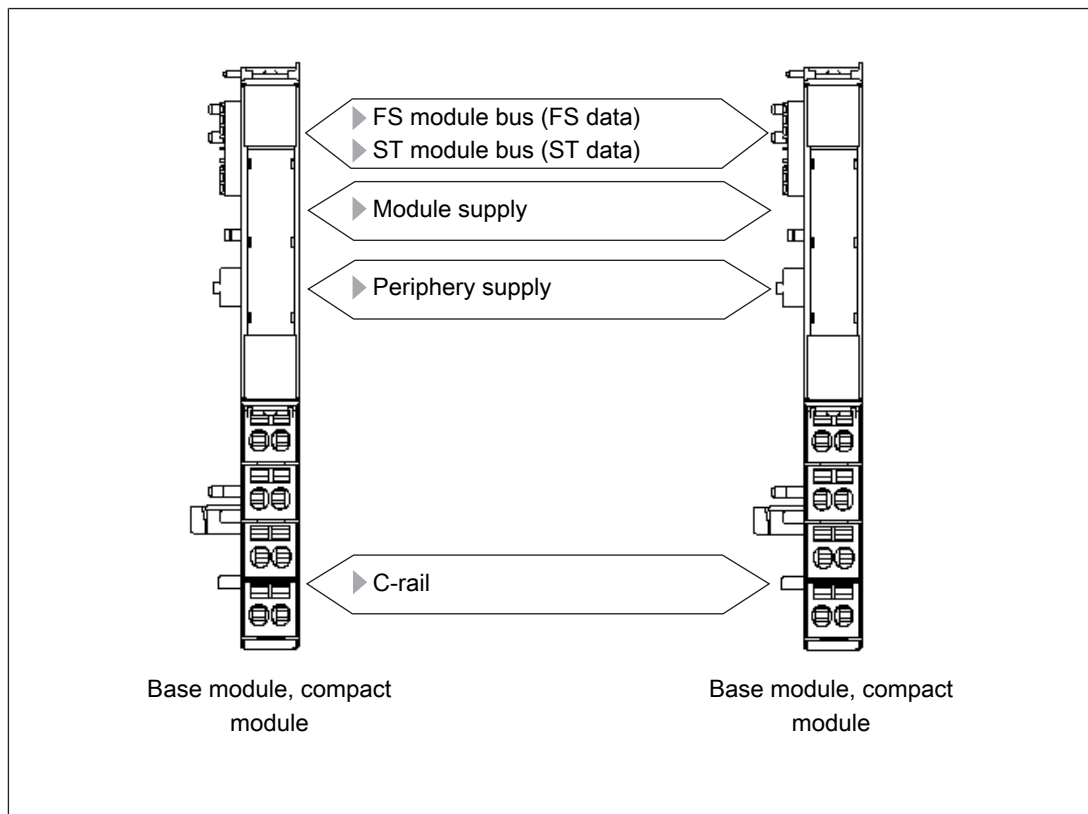
The periphery supply supplies 24 VDC to the sensors and actuators on digital input/outputs on electronic modules. On compact modules, the supply for the inputs is generated from the periphery supply; the supply for the outputs must be fed to the module terminals. The connection on the base module can be disconnected in order to form supply groups.

► **C-rail ("cross connection")**

The C-rail is an additional, free power rail. Access to the C-rail is available via base modules that contain **"-C"** in their description (e.g. PSSu BP-C 1/8 S, PSSu BP-C 1/8 C).

The C-rail supply must belong to the same circuit as the other base module connections. Connecting PE to the C-rail provides a simple form of shielding. On the compact modules, the C-rail is not looped through and is not available at the terminals.





5.6 Connections on the base modules

The connections on the base modules are divided into connection levels and connection columns. The connections have a two-digit number.

- ▶ The first digit denotes a base module's connection column (e.g.: connection **23** is in the second column).
- ▶ The second digit denotes a base module's connection level (e.g.: connection **23** is in the third level).

The function of the base module connections depends on the electronic module.

5.6.1 Connection levels on input/output modules

On input/output modules, the connection levels are typically arranged as follows:

- ▶ Connection level 1 and 4
Input and output connections
- ▶ Connection levels 2 and 3
Connections for the common supplies (periphery supply, analogue inputs/outputs or screening)

Many input/output modules may have two more connection levels:

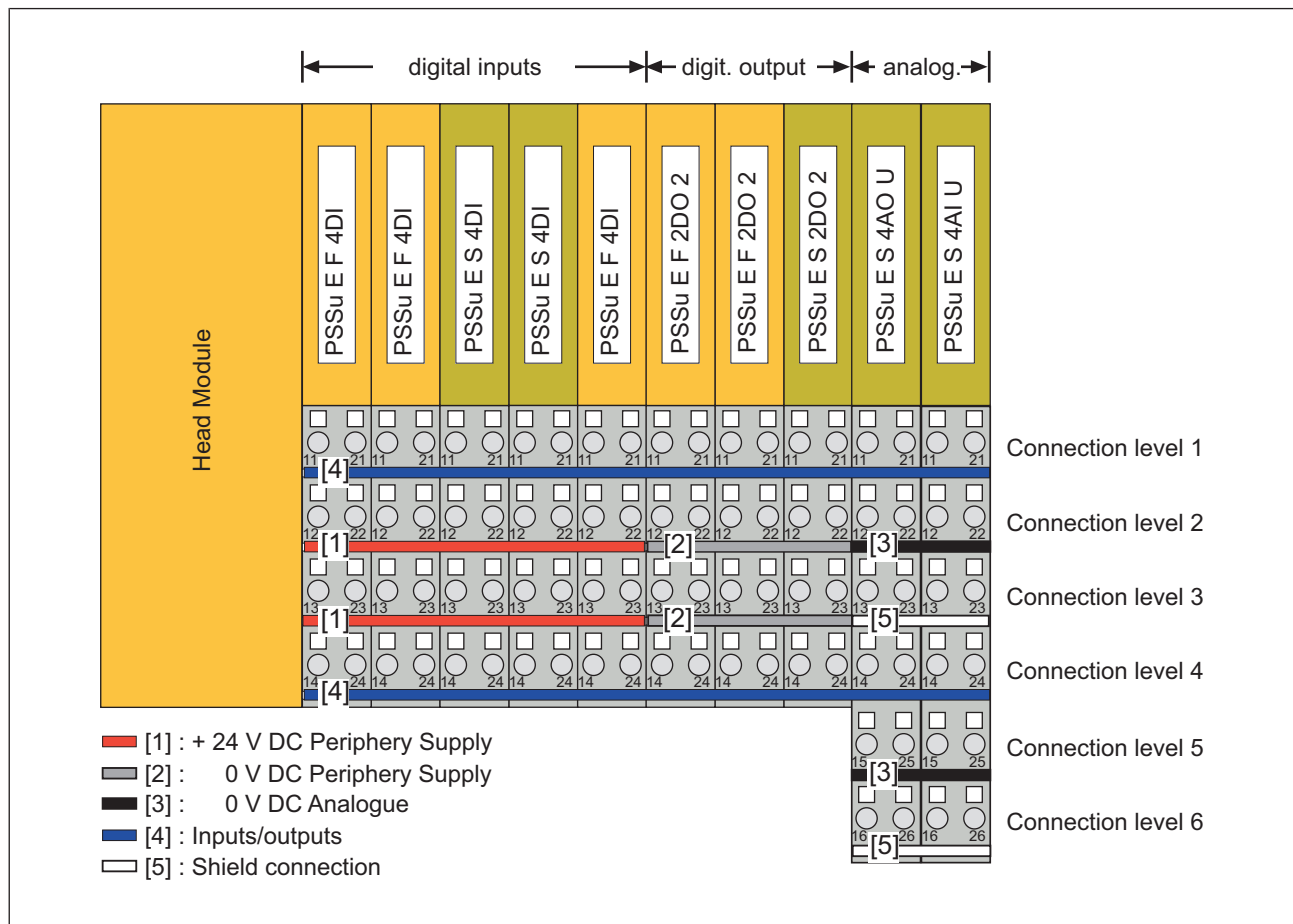
- ▶ Connection level 5 and 6
Connections for the common supplies (periphery supply, analogue inputs/outputs or screening)

Input/output modules can be installed in any order. However, for the connection diagram to be consistent it makes sense to arrange input/output modules of the same type into groups and not have them interrupted by compact modules (see example).

5.6.2 Colour marking on the connection levels

The colour marking on the connection levels is a wiring aid. Various colour markers are available for labelling. The colour markers can be used to label different supplies. This way the connection levels remain clearly visible, even after wiring.

Example:



5.7 Connections on the compact modules

The connections on the compact modules are divided into terminal rows. Single or three-row connectors with spring-loaded terminals can be plugged into the device's single-row 10-pin connector strip.

The function of the connections depends on the compact module.

► **Terminal row 1**

Connection of the inputs and outputs, periphery supply (24 V and 0 V).

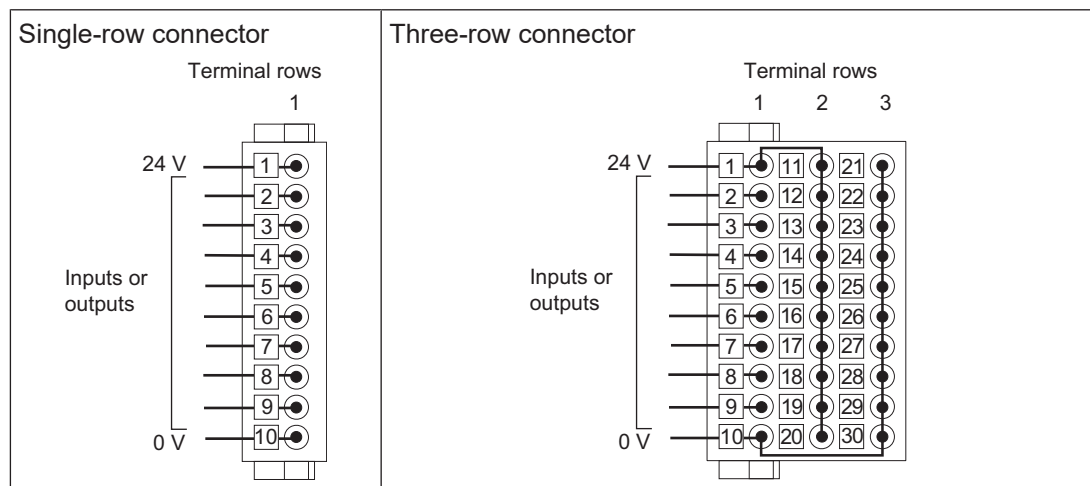
► **Terminal row 2**

All connections are linked; bridged to the 24 V periphery supply connection from the first connection level

► **Terminal row 3**

All connections are linked; bridged to the 0 V periphery supply connection from the first connection level

Input/output modules can be installed in any order. However, for the connection diagram to be consistent it makes sense to arrange input/output modules of the same type into groups.



5.8 Maximum system expansion of a PSSu system

The PSSu Module Editor on PAS4000 provides support in assembling a PSSu system. The tool checks the system limits as the data is entered.

The expansion of a PSSu system is limited by the following values

- ▶ Number of modules
- ▶ Number of FS modules
- ▶ Number of ST modules
- ▶ Number of failsafe inputs/outputs
- ▶ Number of standard inputs/outputs
- ▶ Maximum current load

The system limits are listed and explained below.

5.8.1 Number of modules and number of inputs/outputs on a PSSu system

The following system limits apply:

Module type	Max. number per PSSu system
- - -	Total of 64 modules
ST modules	64 modules
FS modules	64 modules
PSSu E F DIOZ 2	64 modules
ST bit modules	256 ST input bits
	240 ST output bits
FS bit modules	256 FS input bits
	256 FS output bits
ST byte modules	16 modules
FS byte modules	16 modules
PSSu E S RS232	16 modules
PSSu E S RS485	16 modules
PSSu K S RS232	16 modules
PSSu K S RS232 Modbus ASCII	16 modules
PSSu K F FAU	2 modules
PSSu K F EI	8 modules Please note: When used simultaneously with modules of type PSSu K F EI CV, a total of maximum 8 modules, comprising both module types, may be used. However, the upper limit of maximum 4 modules of type PSSu K F EI CV must be maintained. For example, the following combination is permitted: 6 modules of type PSSu K F EI and 2 modules of type PSSu K F EI CV.
PSSu K F EI CV	4 modules
PSSu K F FCU	12 modules

A PSSu system may have a max. 64 modules. When counting up the modules, please note the following peculiarities:

- ▶ The head module and passive junction modules (e.g. PSSu E PD, PSSu E PD1) are not counted.
- ▶ The following modules in a PSSu system are included in the count:
 - Supply voltage modules
(identifier: PSSu E F PS...)
 - Standard modules
(identifier: PSSu E S..., PSSu K S...)

- Failsafe modules
(identifier: PSSu E F..., PSSu K F...)

5.8.2 Maximum current load

5.8.2.1 Maximum current load capacity of the supply voltage modules

System supply	Max. current load
Module supply when supplied via - PSSu E F PS(-T)	1.5 A
Module supply when supplied via - PSSu E F PS1(-T)	2 A
Module supply when supplied via - PSSu E F PS2(-T -R)	1.0 A
Periphery supply	10 A
C-rail	10 A

► **Module supply**

The module supply's current load is the total current consumption resulting from the electronic modules and compact modules. If the total current consumption is higher than the supply module's current load capacity, the module supply must be refreshed with an additional supply module (see modules' technical details). The system software takes this into account.

► **Periphery supply**

Maximum current load: 10 A

The current load of the periphery supply is the sum of the current consumption of the sensors and actuators supplied via the input/output modules. If the current load is higher, the periphery supply must be refreshed with an additional supply module to prevent overload. Please refer to the derating diagram.

► **C-rail**

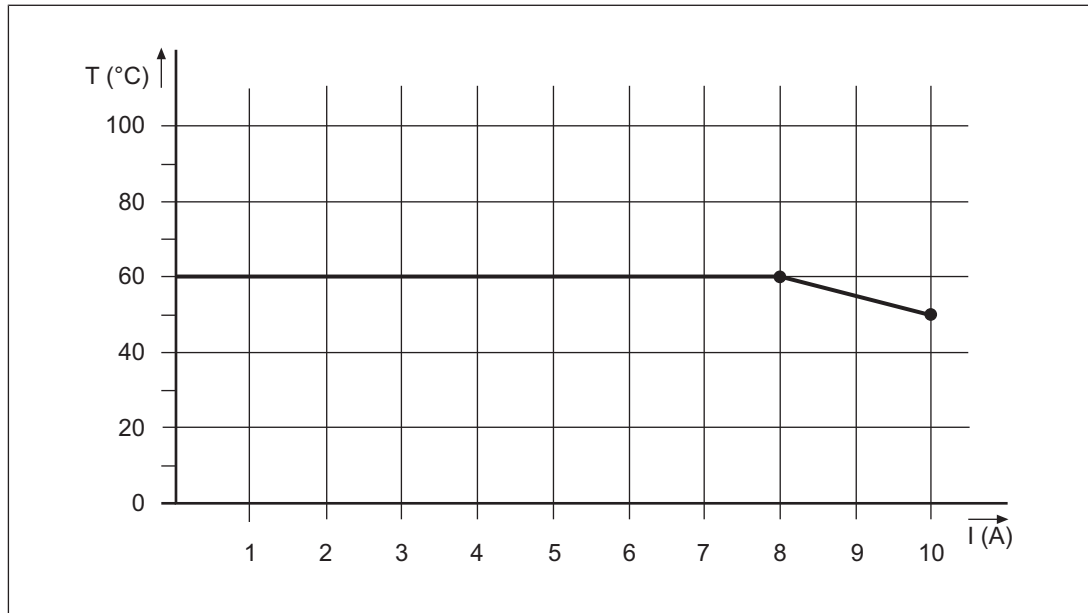
Maximum current load: 10 A

If the current load is higher, the C-rail must use a different supply to prevent overload. Please refer to the derating diagram.

5.8.2.2 Maximum current load capacity based on temperature

The maximum current load capacity with regard to the periphery supply and C-rail can only be achieved if the max. permitted ambient temperature is adhered to.

Derating diagram for periphery supply and C-rail: Temperature T dependent on load current I



T-type: Derating diagram for periphery supply and C-rail: Temperature T dependent on load current I



5.9 Slots of a PSSu system

The slot identifies the location of an electronic module, compact module or supply voltage module on a PSSu system.

There are different ways of looking at the structure of a PSSu system. A distinction is made between the physical structure and the logical structure. As a result, there are physical slots and logical slots.

Physical slots

The following applies for physical slot numbers on slots:

- ▶ The counting sequence starts with the 1st module after the head module.
- ▶ The counting sequence starts with slot number 0.
- ▶ For slot numbering the modules are counted from left to right in ascending order.

Example

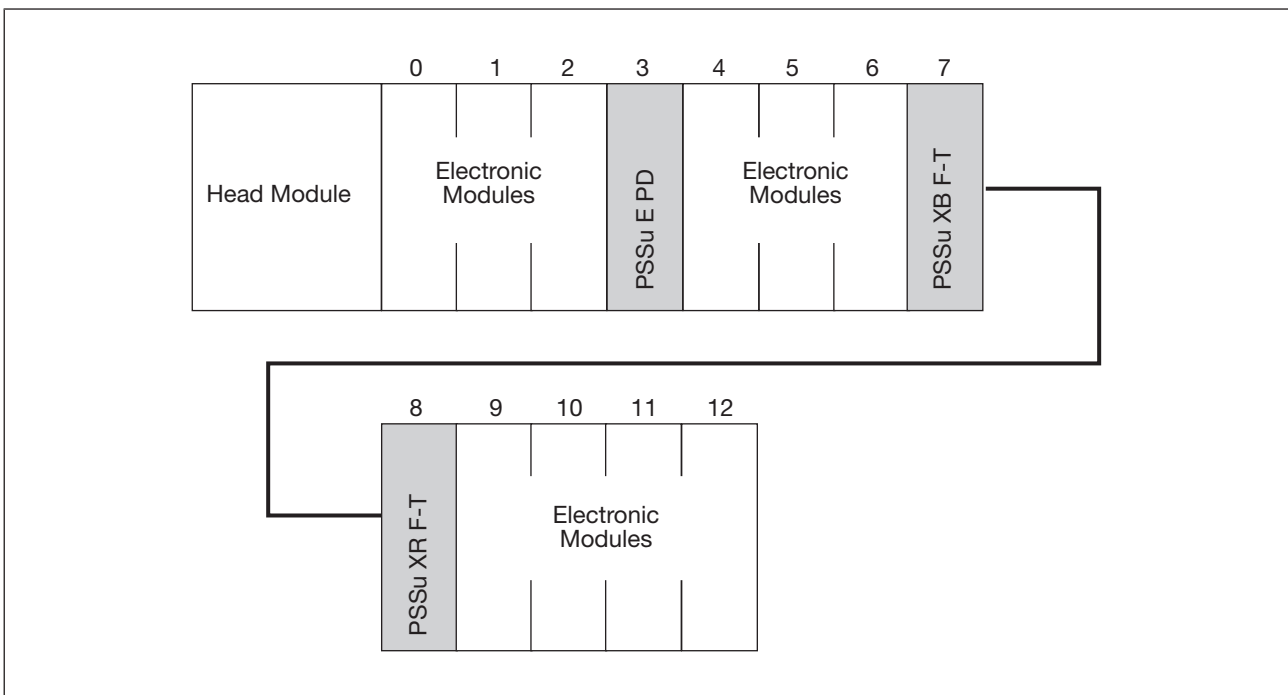


Fig.: Physical slot numbers (example)

Logical slots

The same rules apply for logical slot numbers as physical slot numbers, but link modules (e.g. PSSu XB F-T) and voltage distribution modules (e.g. PSSu E PD) are not counted.

Example

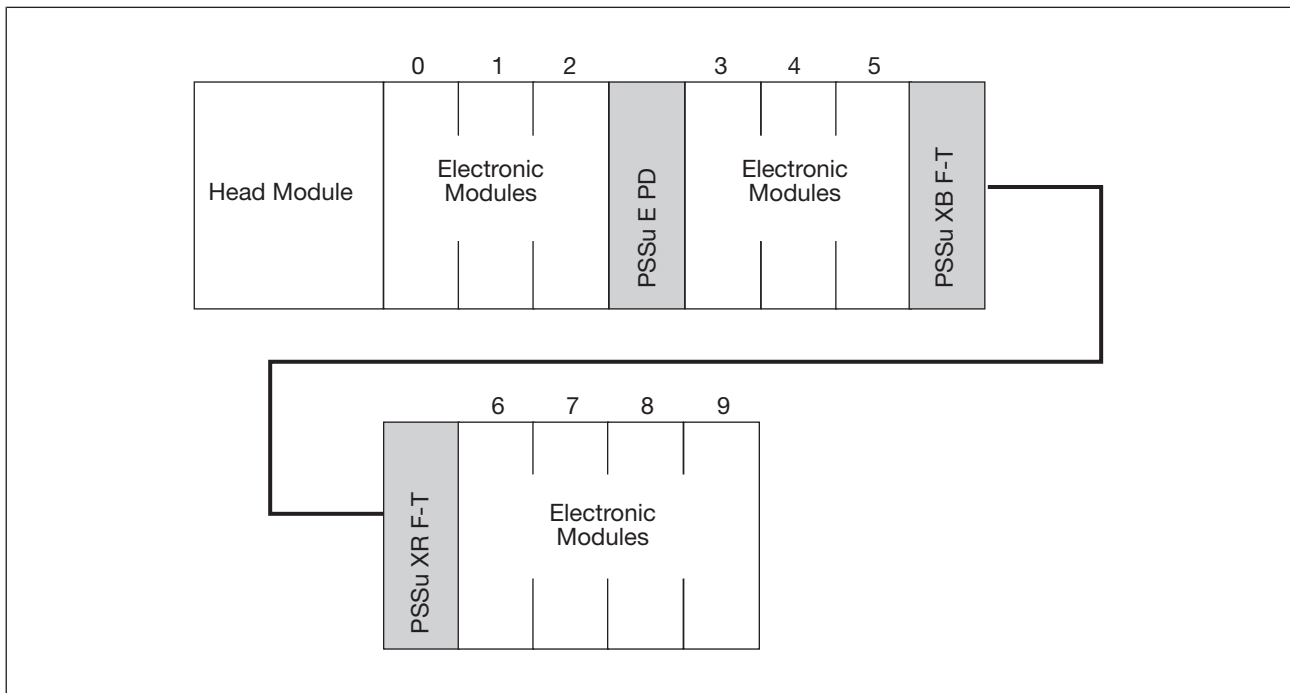


Fig.: Logical slot numbers (example)



INFORMATION

If the diagnostics provide information about a slot, that information refers to the logical slot number.

5.9.1 Set/actual hardware registry on a PSSu system

The device project in a PSSu system contains details of the sequence configured for the modules within their module group.

PSSu modules are divided into the following module groups:

- ▶ FS bit modules (e.g. PSSu E F 4DO 0.5)
- ▶ ST bit modules (e.g. PSSu E S 2DO 2)
- ▶ FS byte modules (e.g. PSSu E F AI U)
- ▶ ST byte modules (e.g. PSSu E S 2AI se)

If the actual hardware registry on the local PSSu system does not match the set hardware registry as regards the module sequence configured for the module group in PAS4000, a configuration error will occur.



INFORMATION

Electronic modules are supplied with a two-part coding element. When an electronic module is plugged into a base module for the first time, one part of the coding element remains on the electronic module, while its counterpart is fixed on to the base module. This is how the base module is coded. The coding element is designed to help prevent plugging errors (e.g. when exchanging an electronic module). An electronic module can only be inserted on a base module with the same mechanical coding. The coding is identified by a letter and a colour. Codings with the same letters are mechanically identical, but their colour may vary.

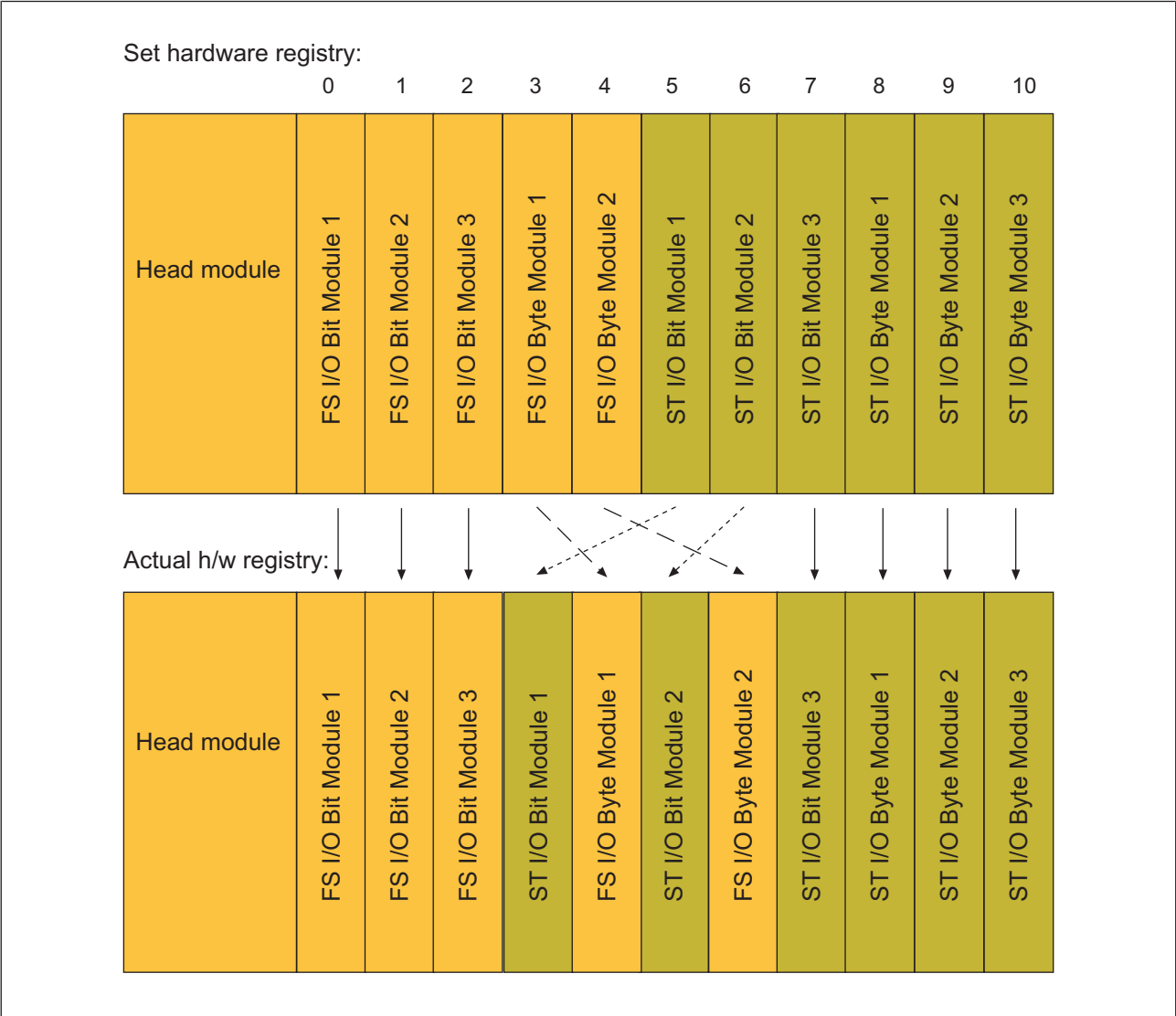


NOTICE

Please note:

When modules are exchanged, an ST electronic module must not be inserted on a base module that is coded with a coding element for an FS electronic module. You can recognise the coding elements for FS modules by the fact that they are yellow. If coding elements have been removed from a base module, you will need to recommission the system in accordance with the commissioning/recommissioning specifications when electronic modules are exchanged, see PSS 4000 Safety Manual.

Example 1: Variations in the set/actual hardware registry without configuration error



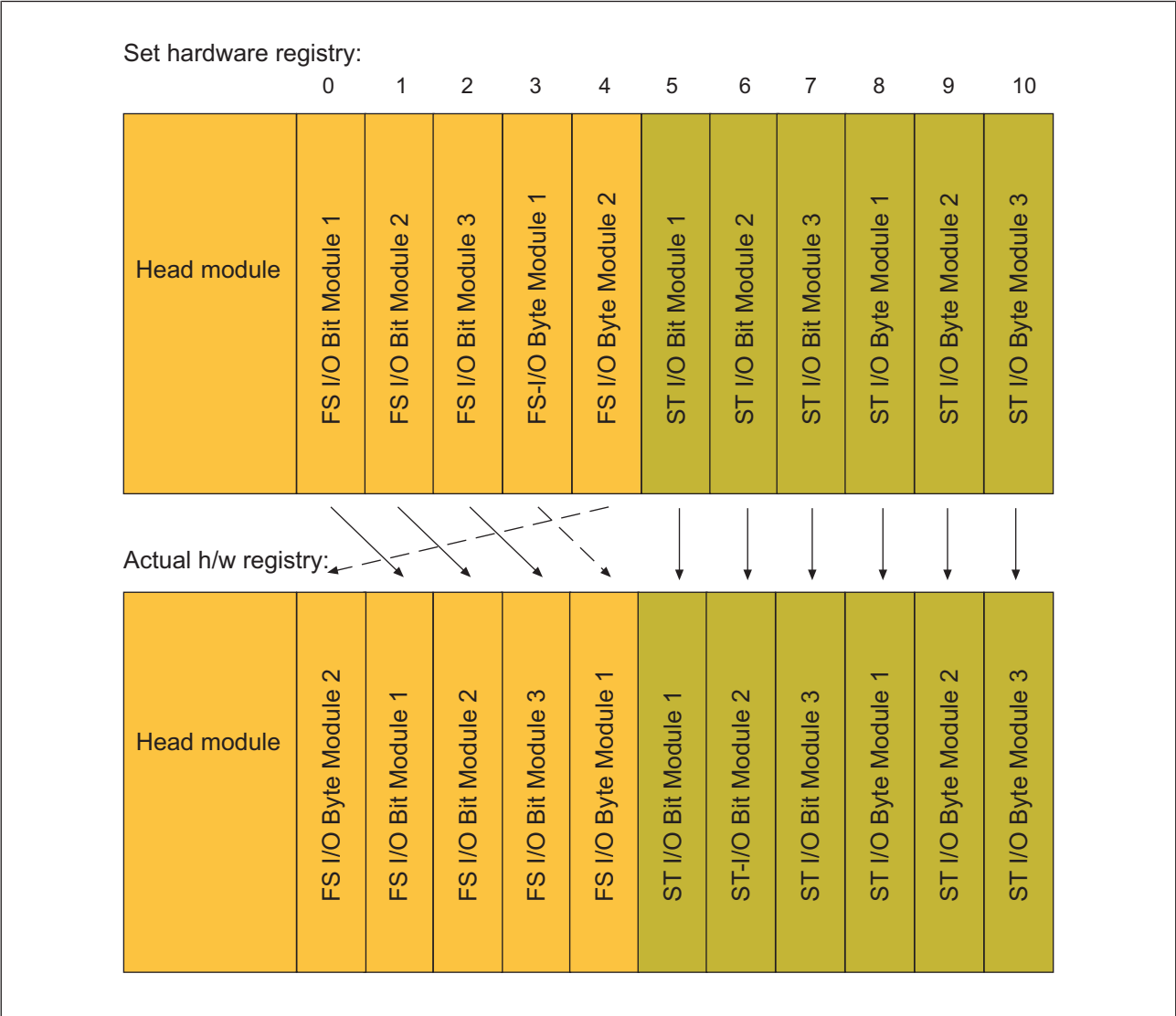
The set hardware registry differs from the actual hardware registry in terms of the slot, but the sequence within the module groups matches. This variation in the set/actual hardware registry **will not lead a configuration error**.



NOTICE

Please note that in this case, diagnostic messages will contain incorrect information about the location.

Example 2: Variations in the set/actual hardware registry with configuration error



The set hardware registry does not match the actual hardware registry either in terms of the slot or the sequence within the module groups. This variation in the set/actual hardware registry will lead to a **configuration error**.

5.10 Supply voltage and insulation

All voltages are supplied via the head module. If more voltages are needed (voltage refresh, potential isolation etc.), these will be supplied via additional supply voltage modules. The voltages are distributed to the modules via the base modules/compact modules.

► Module supply

The module supply is the internal supply voltage for the head module, the compact modules and the electronic modules.

► Periphery supply

The periphery supply supplies 24 VDC to the sensors and actuators on digital input/outputs on electronic modules. On compact modules, the supply for the inputs is generated from the periphery supply; the supply for the outputs must be fed to the module terminals.

The periphery supply can be disconnected in order to form supply groups. On digital failsafe modules, the periphery supply is used to provide test pulses. Failsafe input/output modules will not start until the periphery supply is connected correctly. On analogue input/output modules, the periphery supply is galvanically isolated from the inputs/outputs.

► C-rail

The C-rail is an additional, free power rail. Access to the C-rail is available via base modules that contain "-C" in their description (e.g. PSSu BP-C 1/8 S, PSSu BP-C 1/8 C).

The C-rail supply must belong to the same circuit as the other base module connections. Connecting PE to the C-rail provides a simple form of shielding. On the compact modules, the C-rail is not looped through and is not available at the terminals. Permitted in-feed at the C-rail:

- PE
- 0 V
- Shield
- - 30 VDC ... + 30 VDC
- - 48 VAC ... + 48 VAC

► Inputs/outputs on the analogue modules and counter modules

The inputs and outputs on the analogue modules and counter modules are galvanically isolated from the periphery supply. On the base modules, the supply for the outputs is generated from the periphery supply.

► Earth (ground)

The module supply and periphery supply have separate connections for the respective earth potential (Ground).

► Functional earth

A contact spring on the base modules and compact modules connects the modules' functional earth to the mounting rail.

The supply voltages for the module supply and periphery supply must be functional extra low voltages with safe electrical separation (PELV or SELV). The external power supplies must comply with EN 62368-1 and EN 61010-2-210. Further requirements of the power supplies can be found in the data sheets for the supply voltage modules or head modules.

When the voltages are fed separately using two power supplies, the module supply and periphery supply are galvanically isolated. If galvanic isolation is not required, a common power supply may be used for the periphery supply and module supply.

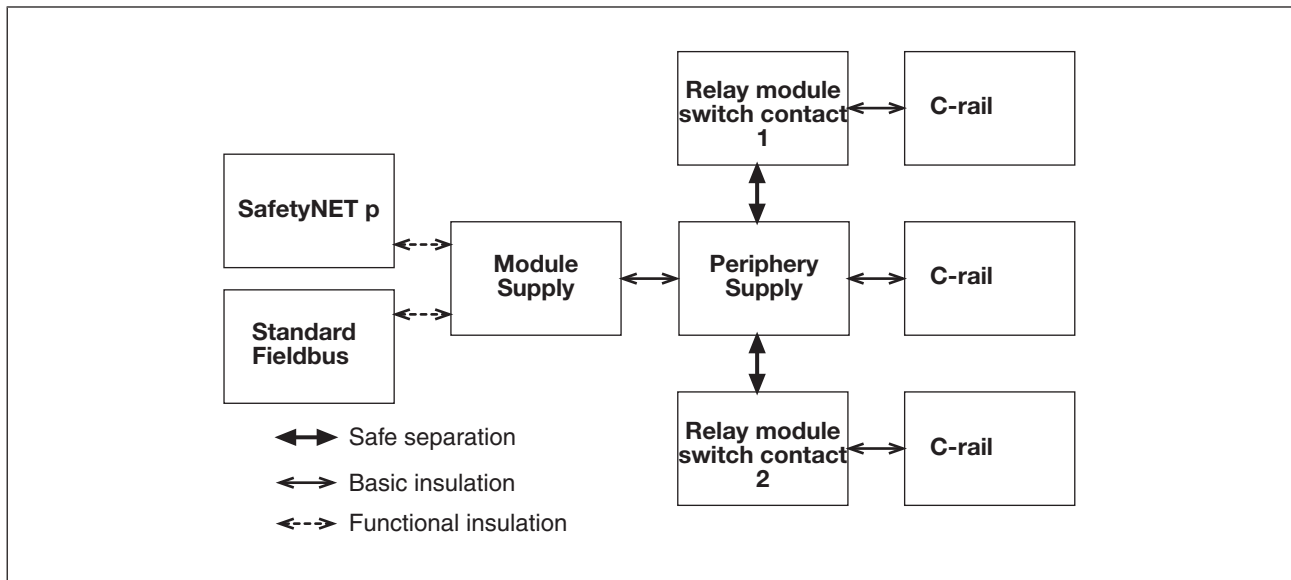
Separate power supplies for module supply and periphery supply increase availability: In the case of a short circuit in the periphery supply, the module supply is maintained and the head module can continue to exchange data with fieldbus systems, e.g. for diagnostic purposes.



INFORMATION

Please refer to the PSSuniversal Installation Manual.

Insulation of supplies in a PSSu system:



WARNING!

Risk of electrocution!

When voltage is applied, contact with live components could result in serious or even fatal injury from an electric shock.

If voltages higher than 50 VAC or 120 VDC are connected to a PSSu system, e.g. to switch contacts on relay modules, please note the following:

- Specific accident prevention regulations apply.
- The C-rail of the supply group should only be connected to the protective earth (PE).



INFORMATION

FS output circuits have been designed to guarantee maximum safety. Extensive tests are carried out internally for this purpose. If the periphery supply should dip during a test function, the relevant output will be deemed to be defective.

An error reaction is triggered: all safety-related system sections on the PSSu system will switch to a STOP condition and all FS outputs on the PSSu system will switch to a safe condition.

In the event of repeated voltage dips, power supplies with a buffered supply voltage should be used for greater availability.

5.11 Supply groups

When used with the relevant base module, any electronic supply voltage module is suitable to separate supplies and form supply groups.

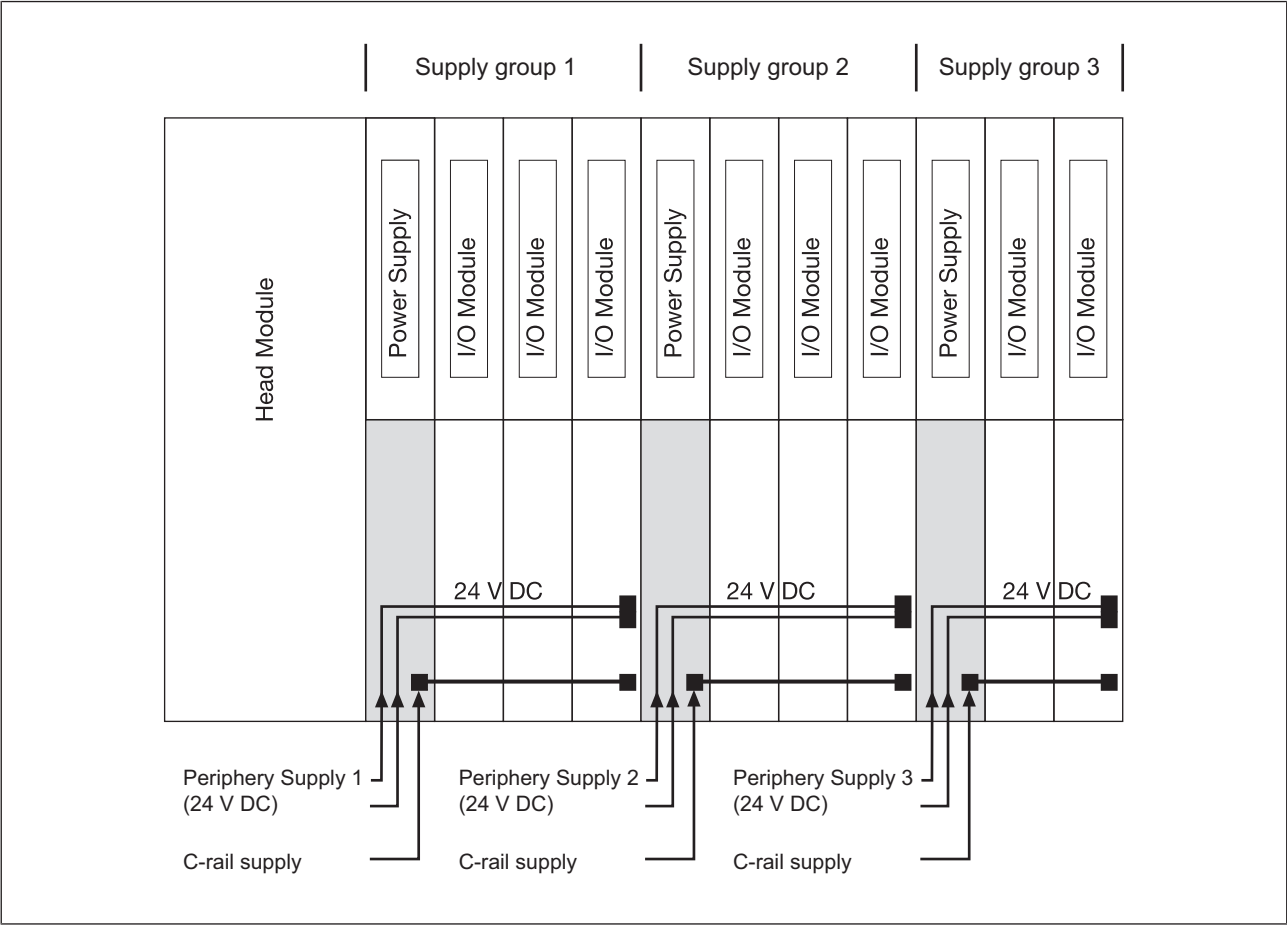
To separate supplies, the supply (periphery supply and C-rail supply) to the preceding (left-hand) modules is disconnected and a new supply provided for subsequent (right-hand) modules.

With a new supply group it is also possible to refresh the module supply by providing a new supply voltage. The 0 V supply on the module supply is never interrupted, but is always looped through across all base modules/compact modules.

You should always refer to the description of the relevant electronic supply voltage module.

The following modules are suitable for separating supplies, for example:

Electronic module	Base module	Module supply	Periphery supply and C-rail
PSSu E F PS	PSSu BS-R 1/8 S PSSu BS-R 1/8 C	Refreshing (new supply voltage is provided, 0 V supply is not interrupted).	Supply of the left-hand modules is interrupted and a new supply is provided for the right-hand modules.
PSSu E F PS1	PSSu BS-R 2/8 S PSSu BS-R 2/8 C		
PSSu E F PS-P	PSSu BS 1/8 S PSSu BS 1/8 C	No refreshing (5 V is not separated).	
PSSu E F BSW	PSSu BS 2/8 S PSSu BS 1/8 C		



The supply voltage for the first supply group can also be integrated within the head module.

5.12 Maximum ambient temperature

The technical details in a device's operating manual contain information about the maximum permitted ambient temperature. When a device is suitable for use where there are increased environmental requirements, the stated ambient temperature must be considered, based on the operating height.

The values stated in the technical details apply to the use of the device in operating heights up to max. 2000 m above SL. When used at higher levels, restrictions of the ambient temperature (standard IEC 61131-2) must be taken into account.

Operating height above SL [m]	Multiplication factors for the devices' ambient temperature
0 ... 2000	1.0
3000	0.9
4000	0.8
5000	0.7

Devices that are suitable for use where there are increased environmental requirements include devices with "-T" or "-R" in the product name.

Examples:

- ▶ PSSu H PLC1 FS SN SD-T
- ▶ PSSu H PLC1 FS SN SD-R
- ▶ PSSu H PLC2 FS SN SD-T
- ▶ PSSu H PLC1 FS SN SD M12-T
- ▶ PSSu H PLC1 FS SN SD M12-R
- ▶ PSSu H FS SN SD-T
- ▶ PSSu H FS SN SD-R
- ▶ PSSu H FS2 SN SD-T
- ▶ PSSu H FS SN SD M12-T
- ▶ PSSu H FS SN SD M12-R

Please refer to the information stated under "Intended use" in the operating manual of the relevant device.

5.13 Status LEDs on head modules

The head modules contain a number of status LEDs, which provide information on the status of various system sections. However, not every head module provides all of the status LEDs described below.

Depending on which head module is used, system sections may be available for external communication, which display the respective operating states via status LEDs (e.g. PROFIBUS-DP Slave, PROFINET IO DEVICE, PROFIBUS-DP Slave with PROFIsafe, EtherNet/IP Adapter). These status LEDs are described in detail in the operating manual for the relevant head module.

Please refer to the operating manual of the head module you are using.



INFORMATION

The diagnostic table and diagnostic log are available for diagnostic messages and remedies.



INFORMATION

Only with PSSu H FS2 SN SD(-T) and PSSu H PLC2 FS SN SD(-T) with hardware version ≥ 2.0 :

When the PSSu system is restarted, the LEDs in the head module's left-hand column light up green in succession (green chase light). This indicates that a released firmware version from Pilz is installed. If a red chase light is displayed, then a test version of the firmware is installed on the head module. If a test version is installed, this is also indicated during operation. In this case, the "DIAG" LED flashes red.

Legend



LED on



LED flashes



LED flashes



LED off

MBUS

The "MBUS" LED indicates the status of the FS and ST module bus.

Colour	Status	Meaning
---		No modules are configured and no modules are available.
Green		FS and ST module bus are operating without error
Red		Operating state "Safe state of all FS outputs on the PSSu system" or Unable to locate at least one module (e.g. a module has been removed during operation, set/actual hardware registry does not match)
		Operating state "FS module bus in STOP state with error: major FS error"

SD CARD or MEDIA

The LED indicates the status of the SD card and is used for device identification.

Colour	Status	Meaning
---		Supply voltage for module supply is missing
Red		▶ SD card is missing - or ▶ SD card not recognised - or ▶ SD card defective - or ▶ Recovery mode: For some reason, the file system of the SD card could not reconstruct a consistent state.
		▶ "Bind device projects to devices" function: – The device does not have the device key that matches the device project. - or – The device project does not have the project key that matches the device. - and/or ▶ "Bind device projects to SD cards" function: The device project is bound to an SD card, but this SD card is not inserted in the device.

Colour	Status	Meaning
Green		Naming data and device project on the device and SD card match
		▶ Product type on the SD card does not match the product type of the device/head module or ▶ No device project on the SD card or ▶ Recovery mode: The file system of the SD card is in a consistent state and the SD card can be removed.
Green-red		Naming data and device project on the device and SD card do not match
Orange		Device identification activated by user ▶ Identification as PSS 4000 device in SafetyNET p (PAS4000) or ▶ Identification as PROFINET IO Device in PROFINET (engineering tool for PROFINET)

FS RUN

The "FS RUN" LED indicates the status of the FS resource.

Colour	Status	Meaning
---		None of the tasks on the FS resource are started or no tasks are configured.
Green		Operating state "FS resource in RUN state without error": The FS resource tasks are running without error. The project is licensed.
		Operating state "FS resource in RUN state with error": - Task in TERMINATED state or - Task in STOP state At least one FS resource task is not running. The project is licensed.
Orange		Operating state "FS resource in RUN state without error": The FS resource tasks are running without error. The project is unlicensed.
		Operating state "FS resource in RUN state with error": - Task in TERMINATED state or - Task in STOP state At least one FS resource task is not running. The project is unlicensed.

ST RUN

The "ST RUN" LED indicates the status of the ST resource.

Colour	Status	Meaning
- - -		None of the tasks on the ST resource are started or no tasks are configured.
Green		Operating state "ST resource in RUN state without error": The ST resource tasks are running without error. The project is licensed.
		Operating state "ST resource in RUN state with error": - Task in TERMINATED state or - Task in STOP state At least one ST resource task is not running. The project is licensed.
Orange		Operating state "ST resource in RUN state without error": The ST resource tasks are running without error. The project is unlicensed.
		Operating state "ST resource in RUN state with error": - Task in TERMINATED state or - Task in STOP state At least one ST resource task is not running. The project is unlicensed.

DIAG

The "DIAG" LED indicates whether there is a fault in a system section. Precise evaluation can be made via the diagnostic list.

Colour	Status	Meaning
---		No system section is started, module supply is missing.
Green		No message of "Error" or "Warning" severity is present for the device.
		Device diagnostic list and device diagnostic log are being prepared
Red		A message of "Error" severity is present for at least one system section (see diagnostic table).
		▶ A major FS error is present for at least one FS system section (see diagnostic list). or ▶ The boot process was stopped because an internal error occurred or the firmware version has been manipulated. Diagnostics are no longer available. The reset button has no function.
		The installed PSS 4000 firmware has not been released. The firmware version is a test version, which may only be used for test purposes.
Orange		A message of at least "Warning" severity is present for the device (see diagnostic list).
Red - green		Start of "deliberate operator action" (function of reset pushbutton)

FS FORCE

The "FS FORCE" LED indicates the status of forcing and online changes on the FS resource.

Colour	Status	Meaning
---		On the FS resource, forcing is inactive and there is no on-line change active
Yellow		On the FS resource, forcing is active and/or there is at least one online change active

ST FORCE

The "ST FORCE" LED indicates the status of the online changes on the ST resource.

Colour	Status	Meaning
---		No online change is active on the ST resource. Note: Forcing on the ST resource is not supported.
Yellow		At least one online change is active on the ST resource.

FS SNp

The "FS SNp" LED indicates the status of the safety-related system section FS SafetyNET p.

Colour	Status	Meaning
---		System section FS SafetyNET p has not been started
Green		Operating state "FS SafetyNET p in RUN state without error"
		Operating state "FS SafetyNET p in RUN state with minor error"
Red		Operating state "FS SafetyNET p in STOP state with error: Major FS error"
		Operating state "FS SafetyNET p in STOP state with error: Major FS+ST error"

ST SNp

The "ST SNp" LED indicates the status of the non-safety-related system section ST SafetyNET p.

Colour	Status	Meaning
---		System section ST SafetyNET p has not been started
Green		Operating state "ST SafetyNET p in RUN state without error"
		Operating state "ST SafetyNET p in RUN state with minor error"
Red		Operating state "ST SafetyNET p in STOP state with error: Major FS+ST error"
		

5V, 24V

The "5 V" LED indicates the status of the module supply.

Colour	Status	Meaning
---		No supply voltage for module supply or supply voltage is faulty
Green		Module supply is available

The "24 V" LED indicates the status of the periphery supply.

Colour	Status	Meaning
---		No supply voltage for periphery supply or supply voltage is faulty
Green		Periphery supply is available

X3: LNK, X3: TRF, X4: LNK, X3: TRF

A PSSu system can have either one Ethernet interface (X3) or two Ethernet interfaces (X3 and X4) (see [Ethernet interface](#) [131]). An Ethernet interface is assigned two status LEDs on the head module as display elements. The status LEDs indicate various connection and communication states.

X3: LNK, X3: LNK

Colour	Status	Meaning
---		No network connection
Green		Network connection is error-free

The designation "LNK" stands for "LINK".

X3: TRF, X4: TRF

Colour	Status	Meaning
---		No data traffic
Yellow		Data traffic is error-free

The designation "TRF" stands for "TRAFFIC".

6 Basics about PSS67 devices

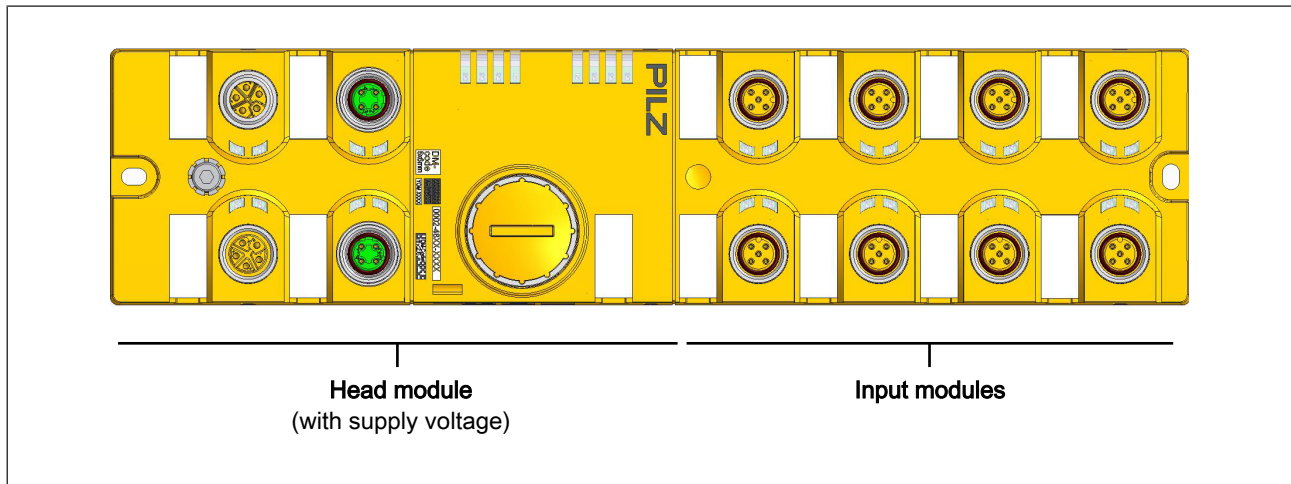
PSS67 devices are PSS 4000-exclusive devices. A PSS67 device is a PSS 4000-exclusive device in a compact design.

The system sections that are available on a device depend on that device (see [System sections](#)  22]). Details can also be found in the operating manual of the relevant device.

6.1 Structure

In principle, a PSS67 device is a PSSu system in compact design for IP67 applications. The IP67 housing contains:

- ▶ Head module
- ▶ Supply voltage
- ▶ Input modules for standard and failsafe applications



Head module

The head module co-ordinates the entire data traffic of a PSS67 device and determines its performance class.

The performance class determines the basic functionality of a device. A distinction is made between the following performance classes:

- ▶ PSS 4000 I/O (abbreviation: I/O)
PSS 4000-exclusive device without control functionality for use as a decentralised I/O module

- ▶ PSS 4000 I/O+ (abbreviation: I/O+)
PSS 4000-exclusive device without control functionality for use as a decentralised I/O module with pre-processing of the I/O signals

The performance class PSS 4000 I/O+ is only available on 2nd generation PSS 4000-exclusive devices (PSS u2 systems).

- ▶ PSS 4000 multi (abbreviation: multi)
PSS 4000-exclusive device with the functionality of a small safety controller
- ▶ PSS 4000 PLC (abbreviation: PLC)
PSS 4000-exclusive device to control automation plants

Features of a PSS67 device, depending on the performance class:

- ▶ PSS67 device of performance class PSS 4000 I/O
 - Control of I/Os via SafetyNET p through a control system of performance class PSS 4000 PLC or PSS 4000 multi.
- ▶ PSS67 device of performance class PSS 4000 PLC
 - Control of I/Os from decentralised PSS 4000 devices of all performance classes when networking with SafetyNET p

- ST resource and FS resource available
- Max. of 9 tasks per resource
- Programmed via the PAS4000 Multi editor or programmed in PAS4000 in accordance with IEC 61131

Input modules

PSS67 devices with digital inputs are available.

Supply voltage

The head module and input modules are supplied with the module supply (5 V) and periphery supply (24 V) via the supply voltage.

6.2 Module bus PSSu

The head module and input modules are connected via an internal module bus. The internal module bus transfers the data between head modules and input modules.

6.3 Slots on a PSS67 device

The slot indicates the location of an input/output module on a PSS67 device.

On PSS67 devices, the slots of the input/output modules are fixed as they are located in a single housing in an unchangeable, fixed sequence. The numbering starts at slot 0. Further information on the slots can be found in the operating manual of the relevant PSS67 device.

6.4 Supply voltage

Voltages on a PSS67 device:

- ▶ **Module supply**

The module supply is the internal supply voltage for the head module and input modules.

- ▶ **Periphery supply**

The periphery supply supplies 24 VDC to the sensors and test pulses of the digital input modules.

Connections for earth (Ground) and functional earth:

- ▶ **Earth (Ground)**

The module supply and periphery supply have separate connections for the respective earth potential (Ground).

- ▶ **Functional earth**

The PSS67 device contains a fixing screw for connection to functional earth.

The supply voltages for module supply and periphery supply must be extra low voltages with safe electrical separation (PELV or SELV) in accordance with VDE 0100, Part 410. Further requirements of the power supplies can be found in the operating manuals of the relevant PSS67 device.

When the voltages are fed separately using two power supplies, the module supply and periphery supply are galvanically isolated. If galvanic isolation is not required, a common power supply may be used for the periphery supply and module supply.

Separate power supplies for module supply and periphery supply increase availability: In the case of a short circuit in the periphery supply, the module supply is maintained and the head module can continue to exchange data with fieldbus systems, e.g. for diagnostic purposes.



NOTICE

Please refer to the information about the supply voltage connection and the maximum current load capacity in the operating manual of the relevant PSS67 device.

6.5 Status LEDs

A PSS67 device contains a number of status LEDs, which provide information on the status of various system sections. The meaning of these status LEDs is identical to the LEDs on the head module of a PSSu system (see [Status LEDs on head modules](#)  79).



INFORMATION

Please note the following differences between a PSS67 device and a PSSu system:

- On a PSS67 device, the Ethernet interfaces are not on X3 and X4, but on X21 and X22. However, the status LEDs are identical in terms of function and labelling.
- A PSS67 device does not contain the status LED "MBUS" as it has no meaning for PSS67 devices.

Further information is available in the operating manual of the relevant PSS67 device.

7 Basics of PSS u2 systems

PSS u2 systems are 2nd generation PSS 4000-exclusive devices. A PSS u2 system is a PSS 4000-exclusive device in a modular design.

The system sections that are available on a PSS u2 system depend on the device (see [System sections](#)  22]). Details can also be found in the operating manual of the relevant head module.

7.1 Structure of a PSS u2 system

A PSS u2 system that can be used in the automation system PSS 4000 consists of:

- ▶ A head module
- ▶ Electronic modules
 - A main supply voltage module
 - Input/output modules for standard and failsafe applications (the input/output modules are not needed if the PSS u2 system is used exclusively to control decentralised I/Os with SafetyNET p)
 - Periphery supply voltage modules (only if necessary)
- ▶ Backplanes
- ▶ Terminal blocks



NOTICE

All modules and backplanes that are used in a PSS u2 system in the automation system PSS 4000 must have "PSS u2" in the name and, in the products' operating manual, their use within the automation system PSS 4000 must be explicitly permitted in the intended use.

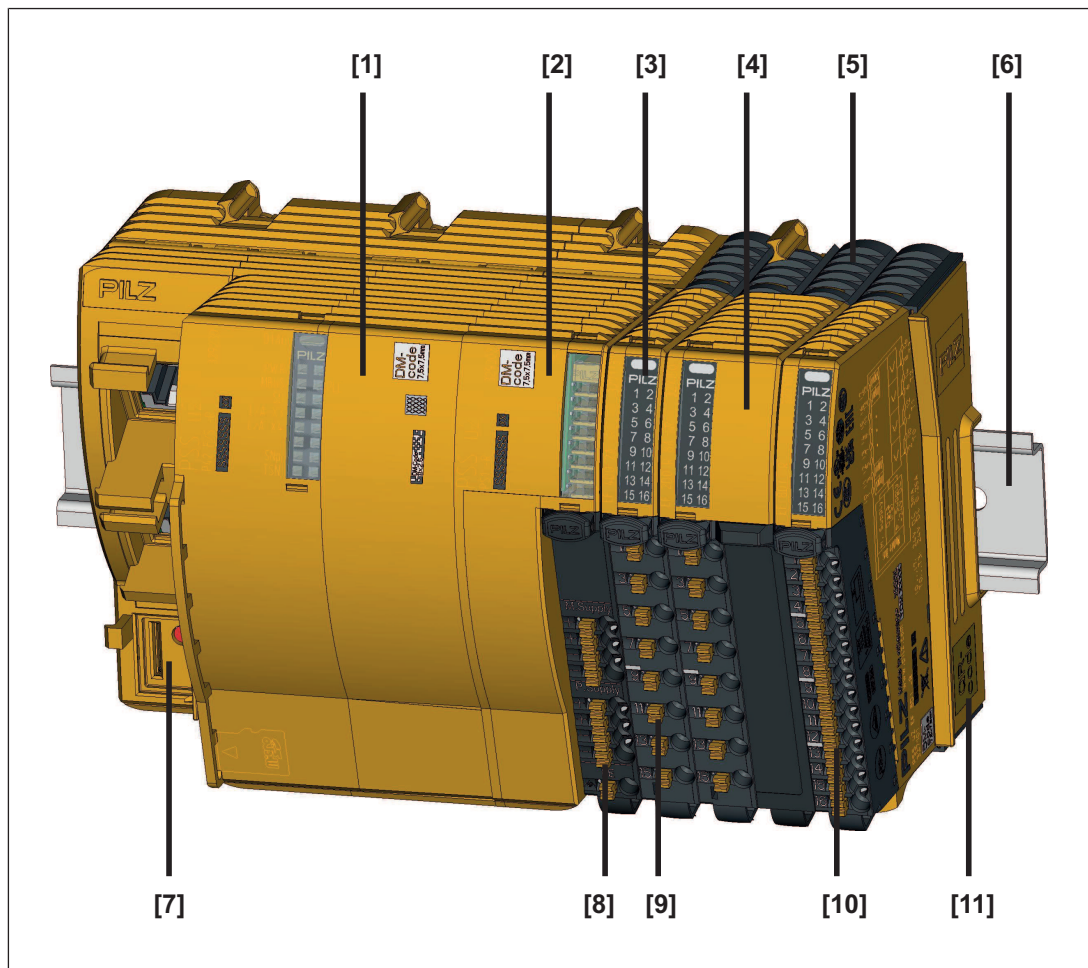


Fig.: Modular structure of PSS u2 in the automation system PSS 4000

Legend

- [1] Head module
- [2] Main supply voltage module
- [3] Electronic module, grid width 1 x 12.5 mm
- [4] Electronic module, grid width 2 x 12.5 mm
- [5] 4-port backplane
- [6] Mounting rail
- [7] Slot for microSD card
- [8] 9-pin terminal block
- [9] 8-pin terminal block
- [10] 16-pin terminal block
- [11] Protective plate (supplied with the main supply voltage module)

Head module

The head module co-ordinates the entire process data traffic of a PSS u2 system and determines its performance class.

The performance class determines the basic functionality of a device. A distinction is made between the following performance classes:

- ▶ PSS 4000 I/O+ (abbreviation: I/O+)
PSS 4000-exclusive device without control functionality for use as a decentralised I/O module with pre-processing of the I/O signals
- ▶ PSS 4000 PLC (abbreviation: PLC)
PSS 4000-exclusive device to control automation plants

Features of a PSS u2 system, depending on the performance class:

- ▶ PSS u2 system of performance class PSS 4000 I/O+
 - Consisting of head module, main voltage supply module and electronic modules
 - Control of I/Os via SafetyNET p through a control system of performance class PSS 4000 PLC
 - FS resource available for pre-processing of I/O signals
 - One task only can be configured
 - Programmed via the PAS4000 Multi Editor
The Multi programming function range is limited to the following blocks:
 - AND operation
 - OR operation
 - XOR - EXCLUSIVE OR operation
 - NOT - Negation
 - SEL - Binary selection
 - SR - Flip-Flop with overriding set
 - RS - Flip-Flop with overriding reset
 - R_TRIG - Detects a rising edge
 - F_TRIG - Detects a falling edge
 - TP - Pulse generator
 - TON - Switch-on delay
 - TOF - Switch-off delay
 - VALID - Evaluation of the valid bit
- ▶ PSS u2 system of performance class PSS 4000 PLC
 - Consists of head module, main supply voltage module, electronic modules; however, the electronic modules are not needed if the PSS u2 system is used exclusively to control decentralised I/Os with SafetyNET p
 - Control of I/Os from decentralised PSS 4000 systems of all performance classes when networking with SafetyNET p
 - ST resource and FS resource available
 - Max. of 9 tasks can be configured per resource
 - Programmed via the PAS4000 Multi Editor or programmed in PAS4000 in accordance with IEC 61131

The PSS 4000 firmware on the head module can be updated using PAS4000.

Electronic modules

The following electronic modules are available:

- ▶ Input/output modules

Input/output modules are available for input/output functions, e.g.

- Digital input/output

- ▶ Main supply voltage module

The main supply voltage module is inserted on the right, next to the head module. It provides the supply voltage:

- Module supply (to supply the head module, input/output modules and periphery supply voltage modules)
- Periphery supply (to supply the sensors and actuators)

- ▶ Periphery supply voltage modules

Modules to supply voltage to the sensors and actuators (periphery supply), e.g. for potential isolation or to refresh voltage.

Electronic modules

- ▶ Are plugged into the backplane.

Exception: the main supply voltage module does not need a backplane.

- ▶ Communicate with the head module via the module bus PSS u2.

The firmware of an electronic module can be updated using PAS4000.

Backplanes

- ▶ Are the carrier units for the electronic modules

Exception: the main supply voltage module does not need a backplane.

- ▶ Are available as 4-port backplanes

Terminal blocks

- ▶ Are used to connect the field wiring

- ▶ Are available with the following connection types:

- Push-in spring-loaded terminals

7.2 Module layout

- ▶ The first module on a PSS u2 system is always a head module.
- ▶ The main supply voltage module is always plugged into the right of the head module.
- ▶ Backplanes are installed to the right of it
- ▶ The input/output modules for FS and ST applications are plugged into the backplanes.
 - The input/output modules may be installed in any order.
 - Free slots may be left between the modules.
 - Input/output modules for FS applications and ST applications may be combined as required.
 - However, if modules of the same function are combined into groups it makes things clearer and simplifies the wiring.
 - The maximum number of input/output modules is determined through the defined system limits.
 - Input/output modules for FS applications cannot be used on head modules that do not have an FS resource.
- ▶ Additional periphery supply voltage modules may be required in order to refresh the periphery supply. The periphery supply voltage module may not be the last module in a PSS u2 system.
- ▶ Supply groups require an additional periphery supply voltage module at the start of each group. The modules belonging to the supply group will then follow to the right.

7.3 Module names

The module names provide information on their function. The names are made up of multi-stage combinations of letters and numbers. All module names begin with **PSS u2**. This is followed by:

- ▶ **P**<*Two-digit number*> for head modules, followed by:
F for FS resource and/or **S** for ST resource
e.g.: PSS u2 **P42** F/S
- ▶ **PS**<*Single-digit number*> for main supply voltage modules
e.g.: PSS u2 **PS1**
- ▶ **E** for electronic modules, followed by:
S for standard module,
e.g.: PSS u2 **ES** 4DI
F for failsafe module,
e.g.: PSS u2 **EF** 8DI
- ▶ **B** for backplane,
e.g.: PSS u2 **B** 4
- ▶ **T** for terminal block,
e.g.: PSS u2 **T** 8.

PSS u2 modules are available as different product types:

- ▶ Base type
The basic versions of PSS u2 modules are called base type modules (e.g. PSS u2 EF 8DI).
- ▶ R-type
PSS u2 modules that are suitable for use with increased environmental requirements and railway applications are called R-type modules. They are identified by an "**-R**" in the product name (e.g. PSS u2 EF 8DI **--R**).

7.4 **Module bus PSS u2**

On the module bus, the data is transferred between head modules and electronic modules. The module bus is formed by arranging the backplanes together and connecting them via a mechanical latch.

The module bus consists of two parts:

- ▶ FS module bus
for transferring failsafe data
- ▶ ST module bus
for transferring standard data

7.5 Backplanes

- ▶ The backplane is used for communication between the head module and the electronic modules and forms the carrier unit for the electronic modules. Communication also takes place when some slots on a backplane are unoccupied.
- ▶ The backplane can be used to interrupt the periphery supply, in order to form supply groups.

For potential isolation, the supply (periphery supply) of the upstream (left-hand) modules on the backplane is interrupted and is refreshed for the downstream (right-hand) modules at a periphery supply voltage module.

A shield connection element can be installed on the 4x backplane as an option. The shield connection element generates the direct contact to the mounting rail. The shield connection element is not supplied with the device.



NOTICE

R-type electronic modules may only be installed on R-type backplanes.

7.6 Maximum system expansion of a PSS u2 system

The PSS u2 Module Editor on PAS4000 provides support in assembling a PSS u2 system. The tool checks the system limits as the data is entered.

The following values limit the expansion of a PSS u2 system:

- ▶ Maximum number of modules
- ▶ Output power of module supply
- ▶ Current load capacity of periphery supply

The system limits are listed and explained below.

7.6.1 **Maximum number of modules in a PSS u2 system**

The number of modules is limited on the one hand by the number of slots, and on the other hand by the output power of the module supply.

Limited by the number of slots.

A maximum of 64 slots can be used. When counting the slots, please note the following:

- ▶ The head module is not counted.
- ▶ The main supply voltage module always occupies the first slot.
- ▶ There are electronic modules that occupy more than one slot.

Limited by the output power of the module supply

Further information is available under [Output power of module supply](#) [ 102].

7.6.2 Output power of module supply

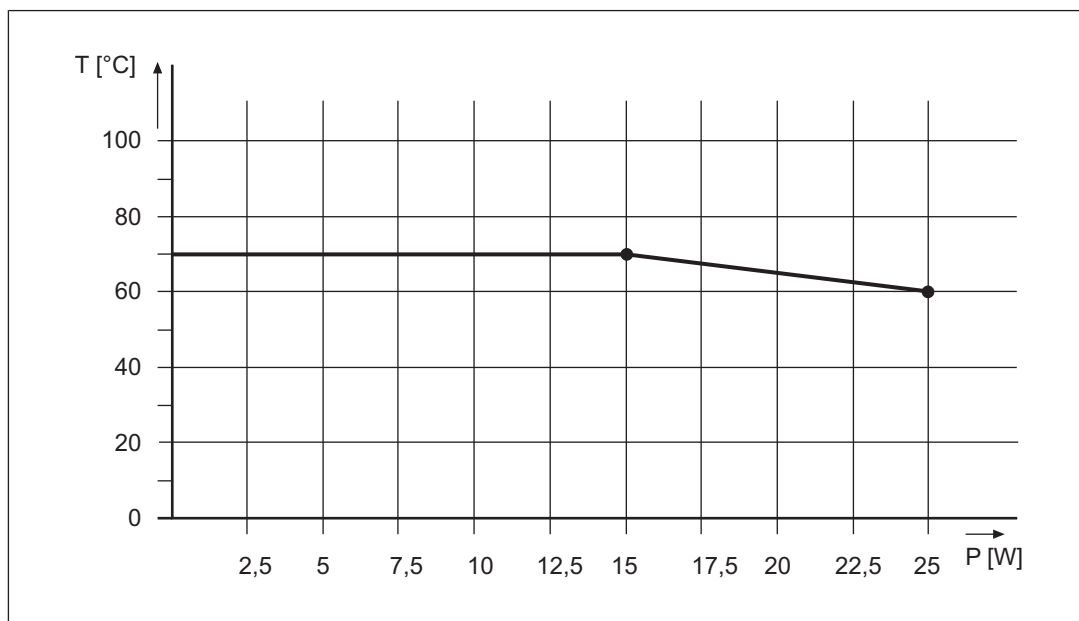
Module	Max. output power
Module supply from the main supply voltage module - PSS u2 PS1 -R	25 W

- ▶ The total power consumption of the electronic modules and head module must not be higher than the max. output power of the main supply voltage module (see Module's technical details).
- ▶ The module supply is provided exclusively by the main supply voltage module.
- ▶ The module supply cannot be refreshed.
- ▶ The module supply does not automatically switch off if the max. output power is exceeded.
- ▶ The module supply's consumption is displayed when a PSS u2 system is configured.

Please refer to the **module supply's derating diagrams**.

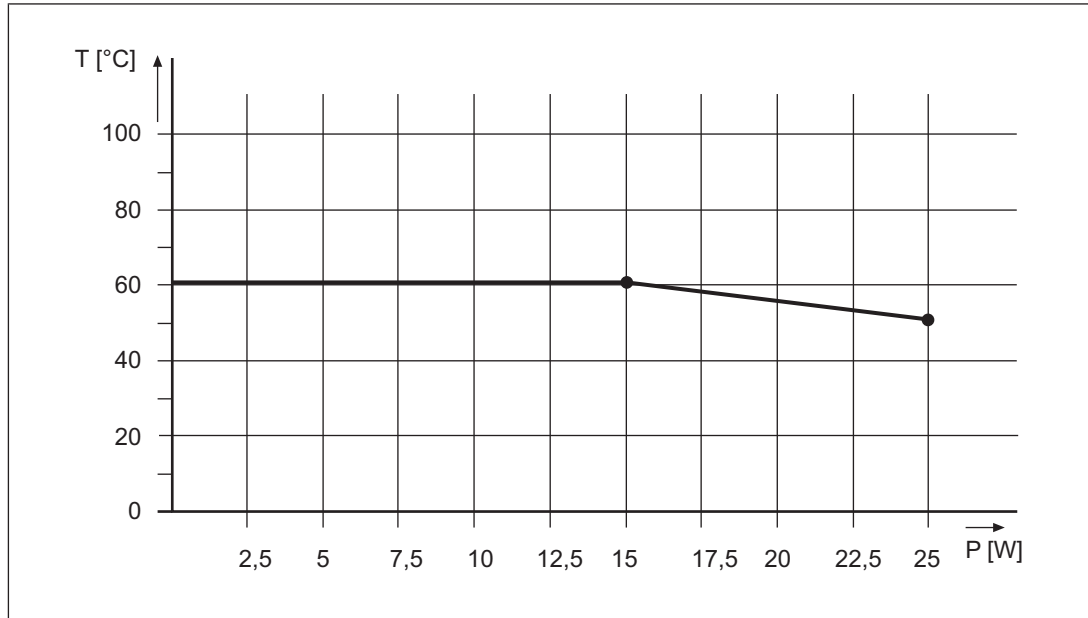
Derating diagram for infeed for module supply: Permitted ambient temperature T dependent on power P .

Installation: Mounting position on the *mounting rail is horizontal* (see Installation Manual PSS u2 in the automation system PSS 4000)



Derating diagram for infeed for module supply: Permitted ambient temperature T dependent on power P .

Installation: Mounting position on the *mounting rail is vertical*, head module is assembled on top (see Installation Manual PSS u2 in the automation system PSS 4000)



7.6.3 Current load capacity of periphery supply

Module	Max. current load capacity
Periphery supply from the main supply voltage module - PSS u2 PS1 -R	8 A
Periphery supply when supplied via - PSS u2 ES PSP -R	8 A

- ▶ The current load capacity of the periphery supply is the sum of the current consumption of the sensors and actuators that are supplied via the input/output modules.
- ▶ If the total current consumption is higher than the current load capacity of the main supply voltage module, then the periphery supply must be refreshed using a periphery supply voltage module, in order to avoid overload.
- ▶ The periphery supply does not switch off automatically if the maximum current load capacity is exceeded.
- ▶ The periphery supply's consumption is displayed when a PSS u2 system is configured.

7.7 Slots on a PSS u2 system

The slot indicates the physical location of a module on a PSS u2 system. The slot numbers are displayed in the diagnostic list and when configuring a PSS u2 system, for example.

The following applies for the slot numbers:

- ▶ The counting sequence starts with the 1st module after the head module.
- ▶ The counting sequence starts with slot number 1.
- ▶ There are slot numbers 1 ... 64.
- ▶ Free slots are also counted.
- ▶ If a module occupies more than one slot, then it is addressed via the lowest slot number.

The module numbers are also displayed when configuring a PSS u2 system.

The following applies for the module numbers:

- ▶ The modules are counted from left to right in ascending order.

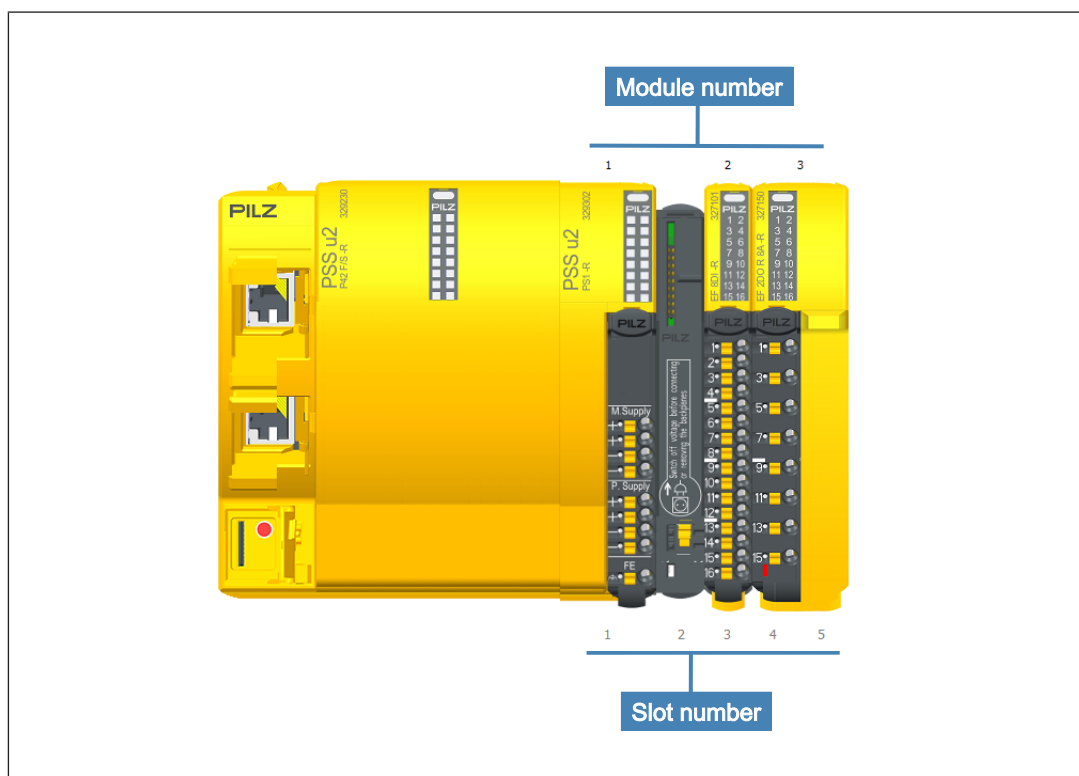


Fig.: Slot numbers and module numbers

7.8 Set/actual hardware registry on a PSS u2 system

The device project on a PSS u2 system contains information on the configured modules (hardware registry). During operation, for each slot, the set hardware registry is compared with what is physically present (actual hardware registry).

The following deviations lead to a configuration error:

- ▶ The product type of the configured module does not match the product type of the inserted module.
- ▶ The firmware version of the configured module is newer than the firmware version of the inserted module and the two versions are not compatible (the newer firmware version contains functions that the older firmware version does not support).
- ▶ A configured FS module with an FS output is not inserted and the FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system" is configured for the PSS u2 system.
- ▶ A configured supply voltage module is not inserted.
- ▶ A module is inserted, which may not be used in the automation system PSS 4000.

Examples of deviations that do not lead to a configuration error:

- ▶ ST input module or ST output module is not inserted.
- ▶ An FS output module is not inserted and the FS shutdown behaviour "Shutdown of all FS outputs on the electronic module" is configured.

7.9 Validity of the output signals

The *OutputValid* bit can be used to establish whether an output is issuing valid data. The bit is TRUE if the data source is supplying valid data, the output signal matches the signal of the data source and the output is error-free.

Availability of the *OutputValid* bit:

- ▶ The bit is available for FS output modules if the FS shutdown behaviour "Shutdown of all FS outputs on the electronic module" is configured for the PSS u2 system. In this case, the *OutputValid* bit applies to all outputs in the FS module output group. See [FS shutdown behaviour "Shutdown of all FS outputs on the electronic module" \[125\]](#).
- ▶ The bit is available for ST output modules with advanced diagnostic options. In this case, there is one bit per output. If an output module provides these outputs, it will be described in the operating manual under "I/O data available in I/O mapping".

The *OutputValid* bit is an I-datum of the output module. On an FS output module, the bit has the data type FS_I_DI. On an ST output module, the bit has the data type ST_I_DI. The bit can be found in I/O mapping.

7.10 Supply voltage

The supply voltage for the module supply and periphery supply is fed in via the main supply voltage module. The supply voltage for the periphery supply can be fed in again at a separate periphery supply voltage module. The voltages are distributed to the modules via the backplanes.

- ▶ Supply voltage for the module supply

The supply voltage for the module supply is fed in at the terminals labelled "M. Supply" on the main supply voltage module and is distributed, electrically isolated, to the head module and, via the backplanes, to the electronic modules.
- ▶ Supply voltage for the periphery supply

The supply voltage for the periphery supply is fed in at the main supply voltage module (or a periphery supply voltage module) at the terminals labelled "P. Supply".
- ▶ Module supply

The module supply is the internal supply voltage for main supply voltage module, head module and electronic modules. It is galvanically isolated from the supply voltage for the periphery supply.
- ▶ Periphery supply

The periphery supply supplies 24 VDC to the sensors and actuators on input/outputs on electronic modules. On analogue input/output modules, the periphery supply is galvanically isolated from the inputs/outputs.

The connection on the backplane can be disconnected in order to form supply groups. On digital failsafe modules, the periphery supply is used to provide test pulses. Failsafe input/output modules will not start until the periphery supply is connected correctly.

- ▶ Shield connection element

The shield connection element should be connected directly to the mounting rail. Permitted infeed at the shield connection element:

- Shield

► Earth (ground)

The supply voltage for the module supply and the supply voltage for the periphery supply have a common connection for the respective potential earth (ground).

► Functional earth

A contact spring on the backplanes/electronic modules/head modules connects the modules' functional earth to the mounting rail.

The supply voltages for the module supply and periphery supply must be functional extra low voltages with safe electrical separation (PELV or SELV). The external power supplies must comply with EN 62368-1 and EN 61010-2-210.

► Interference voltage on the supply lines

The user must employ additional measures on the supply voltage inputs of the main supply voltage module (PSS u2 PS1 -R) and the supply voltage module for the periphery supply (PSS u2 ES PSP -R), to ensure compliance with the limit values for the conducted noise emission in accordance with EN 50121-4:2016 + A1:2019. Compliance with the limit values is achieved by connecting a noise suppression filter between the external power supply and the supply voltage inputs of the supply voltage modules if, for example, a type FN2070-3-.. mains filter from Schaffner is used, or a noise suppression filter with comparable characteristics.

When the supply voltages are fed in separately using two power supplies, the supply voltage for the module supply and the supply voltage for the periphery supply are galvanically isolated. If galvanic isolation is not necessary, a common power supply may be used for the supply voltage for the periphery supply and the supply voltage for the module supply.

Separate power supplies for the supply voltage for the module supply and periphery supply increase availability: In the case of a short circuit in the supply voltage for the periphery supply, the supply voltage for the module supply is maintained. As a result, the user program continues to be executed on the head module and the head module can continue to exchange data with fieldbus systems, for diagnostic purposes for example.



WARNING!

Risk of electrocution

Serious or fatal injury could result, depending on the application.

- Always connect the mounting rail to the protective earth via an earthing terminal. This will be used to dissipate hazardous voltages in the case of a fault.
- Do not connect a base-insulated device to a supply voltage of over 50 VAC or 120 VDC parallel to the module supply.

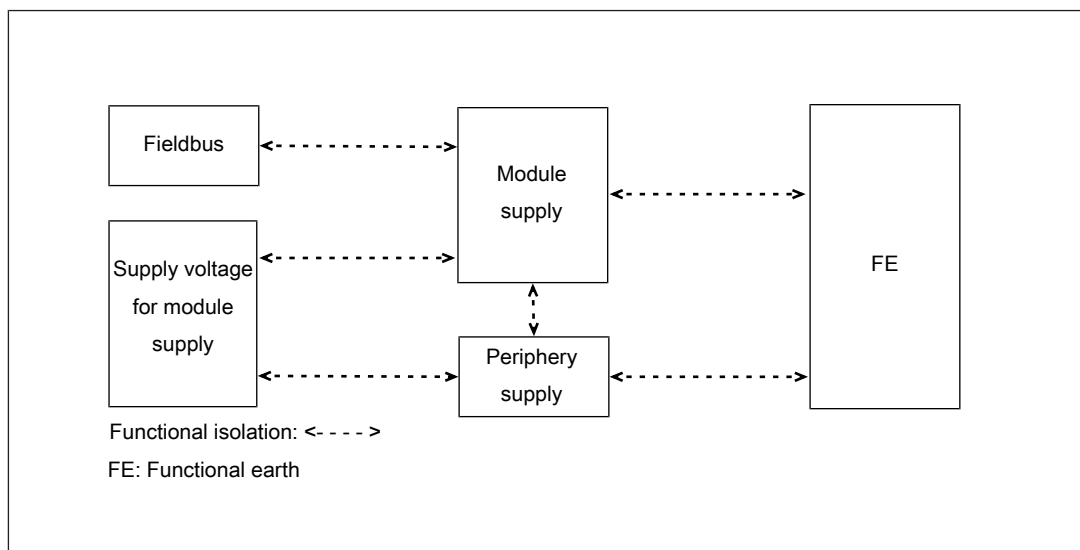


Fig.: Potential isolations



WARNING!

Risk of electrocution

When voltage is applied, contact with live components could result in serious or even fatal injury from an electric shock.

If voltages higher than 50 VAC or 120 VDC are connected to a PSS u2 system, to switch contacts on relay modules for example, please note:

- Specific accident prevention regulations apply.
- Only the functional earth (FE) may be connected to the shield connection element.
- When laying cables alongside or above modules, the dielectric strength of the cable insulation should be ≥ 1500 VAC.

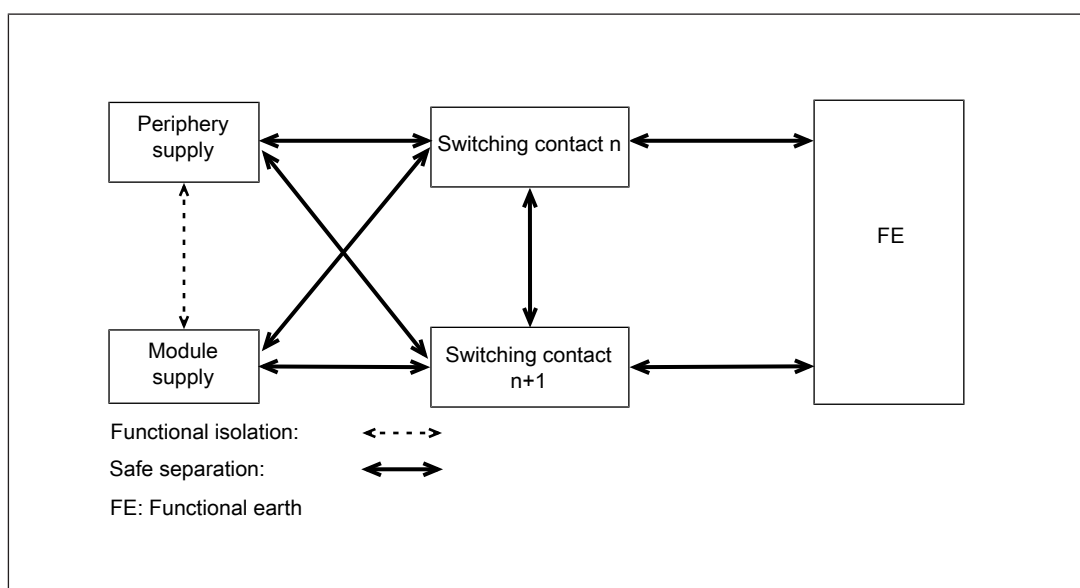


Fig.: Additional potential isolations in the case of modules for voltages above 50 V AC or 120 V DC

7.11 Supply groups

Supply groups can be formed in a number of ways, depending on the application.

Formation of supply groups and electrical isolation with a redundant structure

Depending on the requirements of an application's safety architecture, it may be necessary for a PSS u2 system to have a redundant structure. One of the potential measures is to form supply groups. For example, this procedure is used for SIL 4 railway applications (see System Description "Safety architectures for railway applications").

In this case, in order to form supply groups, the periphery supply to the upstream (left-hand) I/O modules is disconnected and a new supply is provided for downstream (right-hand) I/O modules via a separate periphery supply voltage module. The periphery supply voltage module does not necessarily have to be on the redundant backplane (see diagram).

The periphery supply is disconnected via the contact block on the corresponding slot. This is the exact slot in which the supply voltage module PSS u2 ES PSP -R must be inserted and the periphery supply fed in again. In the example (see diagram), this is slot 4 on backplane 1.

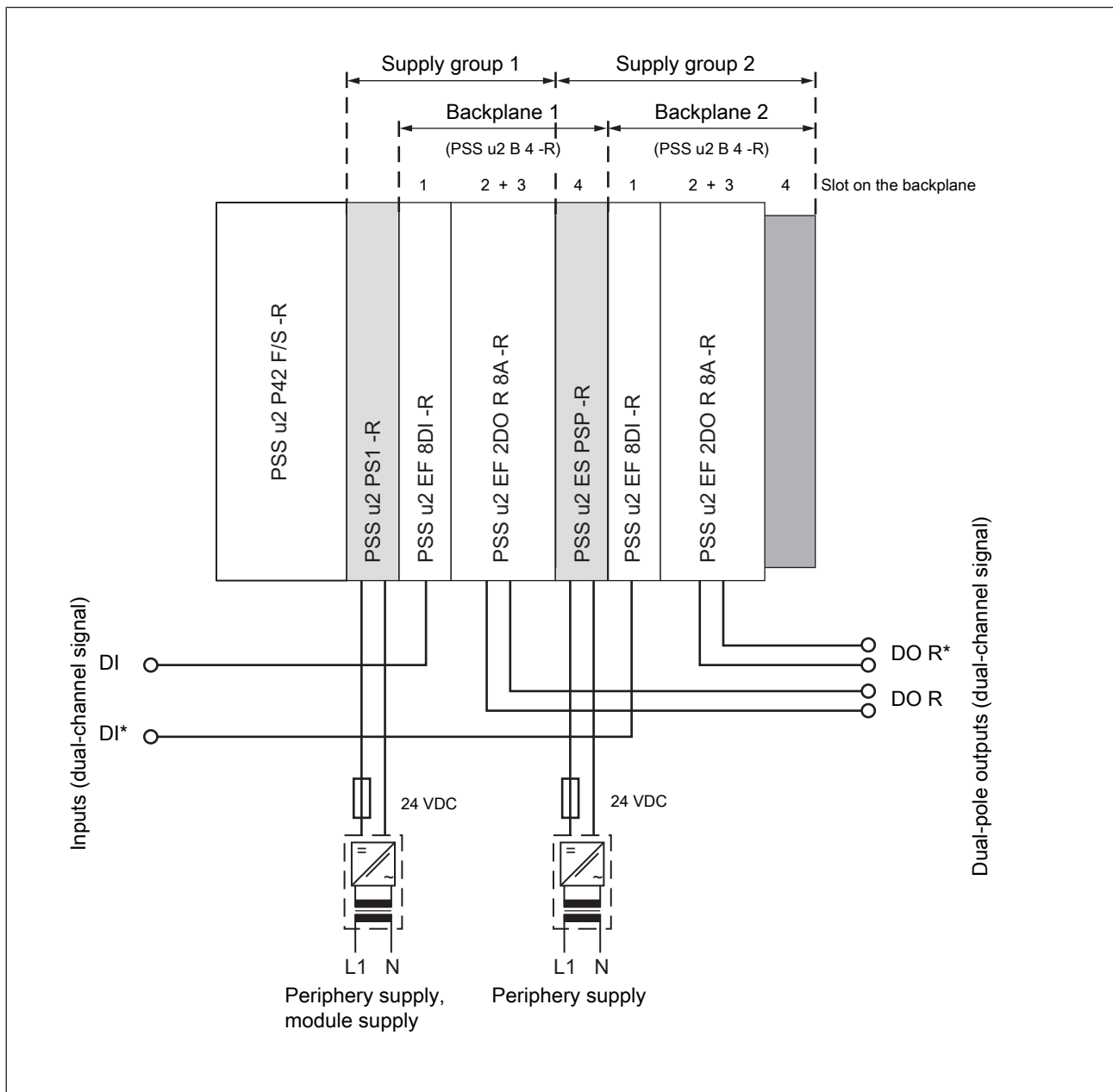


Fig.: PSS u2 supply groups (example)

Formation of supply groups to refresh the periphery supply

Due to the power consumption, it may be necessary to refresh the periphery supply. To do this, the periphery supply on the backplane for the upstream (left-hand) modules is disconnected and a new supply is provided for downstream (right-hand) modules via a periphery supply voltage module.

7.12 Maximum ambient temperature

The technical details in a device's operating manual contain information about the maximum permitted ambient temperature. When a device is suitable for use where there are increased environmental requirements, the stated ambient temperature must be considered, based on the operating height.

The values stated in the technical details apply to the use of the device in operating heights up to max. 2000 m above SL. When used at higher levels, restrictions of the ambient temperature (standard IEC 61131-2) must be taken into account.

Operating height above SL [m]	Multiplication factors for the devices' ambient temperature
0 ... 2000	1.0
3000	0.9
4000	0.8
5000	0.7

Devices that are suitable for use where there are increased environmental requirements include devices with "-R" in the product name.

Examples:

- ▶ PSS u2 P42 F/S -R
- ▶ PSS u2 P12 F/S -R
- ▶ PSS u2 PS1 -R
- ▶ PSS u2 B 4 -R
- ▶ PSS u2 ES PSP -R
- ▶ PSS u2 EF 4DO 2A -R
- ▶ PSS u2 EF 8DI -R
- ▶ PSS u2 EF 2DO R 8A -R

Refer to the device's operating manual, under "Intended use", for details of whether a device is suitable for use where there are increased environmental requirements.



CAUTION!

Risk of material damage

The incorrect ambient temperature during assembly/dismantling can lead to material damage.

- Comply with the permitted ambient temperature for assembly/dismantling: -20°C ... +45°C

7.13 Status LEDs on PSS u2 head modules

The head modules contain a number of status LEDs, which provide information about the status of the PSS u2 system and its system sections. Not every head module provides all of the status LEDs described below.



INFORMATION

When the PSS u2 system is restarted, the LEDs in the head module's left-hand column light up green in succession (green chase light). This indicates that a released firmware version from Pilz is installed. If a red chase light is displayed, then a test version of the firmware is installed on the head module. If a test version is installed, this is also indicated during operation. In this case, the "DIAG" LED flashes red.



INFORMATION

The diagnostic list and diagnostic log are available for diagnostic messages and remedies.

Legend



LED on



LED flashes



LED flashes quickly



LED flickers



LED off

"DIAG" LED

The "DIAG" LED indicates whether there is a fault or error on the device. Precise evaluation can be made via the diagnostic list.

Colour	Status	Description
Green		No message of "Error" or "Warning" severity is present for the device.
		▶ Device diagnostic list and device diagnostic log are being prepared. or ▶ When the DIAG, MBUS, PWR, SD CARD, SNp and TSN LEDs all flash green, the user has activated device identification.
Yellow		A message of at least "Warning" severity is present for the device (see diagnostic list).
Red		▶ A major FS error is present for at least one FS system section (see diagnostic list). or ▶ The firmware version has been manipulated.
		There is at least one message of "Error" severity present for the device, which can be recovered by the user (see diagnostic list).
		The installed PSS 4000 firmware has not been released. The firmware version is a test version, which may only be used for test purposes.

"PLC" LED

The "PLC" LED indicates the status of the FS and/or ST resource.

Colour	Status	Description
---		No tasks are configured.
Green		All configured FS tasks and ST tasks are running.
Yellow		The FS and/or ST resource is not in normal mode: ▶ At least one online change is active on the FS or ST resource. and/or ▶ Forcing is active on the FS resource.
Red		At least one task is not running.

"I/O+" LED

The "I/O+" LED indicates the status of the FS resource.

Colour	Status	Description
---		No task is configured.
Green		The FS task is being executed.
Yellow		The FS resource is not in normal mode: ▶ At least one online change is active on the FS resource. and/or ▶ Forcing is active on the FS resource.
Red		The task is not being executed.

"PWR" LED

The "PWR" LED indicates the status of the supply voltage for the module supply.

Colour	Status	Description
---		Supply voltage for the module supply is missing.
Green		Supply voltage for the module supply is OK.
		When die DIAG, MBUS, PWR, SD CARD, SNp and TSN LEDs all flash green, the user has activated device identification.
		Updating the firmware.

"MBUS" LED

The "MBUS" LED indicates the status of the PSS u2 module bus.

Colour	Status	Description
---		The module bus is just being started or an internal error occurred on the module bus.
Green		FS and ST module bus are operating without error.
		When die DIAG, MBUS, PWR, SD CARD, SNp and TSN LEDs all flash green, the user has activated device identification.
Red		An error that can be repaired by the user has occurred on the module bus. See diagnostic list.

"SD" LED

The "SD" LED indicates the status of the SD card.

Colour	Status	Description
Green		Naming data and device project on the device and SD card match.
		▶ There is no device project and no naming data on the SD card. - or ▶ Recovery mode: The file system of the SD card is in a consistent state and the SD card can be removed. - or ▶ When die DIAG, MBUS, PWR, SD CARD, SNp and TSN LEDs all flash green, the user has activated device identification.
Yellow		Naming data and device project on the device and SD card do not match. The "deliberate operator action" is required to load into the device.
		Device is ready for "deliberate operator action". Status: Start of "deliberate operator action" (function of reset pushbutton)

Colour	Status	Description
Red		▶ SD card is missing. or ▶ SD card was not detected. or ▶ SD card is defective. or ▶ Recovery mode: For some reason, the file system of the SD card could not reconstruct a consistent state.
		Naming data and/or device project from the SD card may not be loaded into the device: ▶ Product type on the SD card does not match the product type of the device/head module. or ▶ "Bind device projects to devices" function: – The device does not have the device key that matches the device project. or – The device project does not have the project key that matches the device. and/or ▶ "Bind device projects to SD cards" function: The device project is bound to an SD card, but this SD card is not inserted in the device.

"L/A" LED:

The "L/A" LED indicates the status of the Ethernet network connection.

Colour	Status	Description
---		Network connection is missing.
Green		Network connection is available.
		Data traffic is taking place.

"SNp" LED

The "SNp" LED indicates the status of SafetyNET p.

Colour	Status	Description
---		SafetyNET p is not in operation: FS and ST SafetyNET p are not started.
Green		SafetyNET p is in operation: ▶ All expected data is being received. and/or ▶ There is no data configured that is expected to be received.
		SafetyNET p is in operation: ▶ After the device is started, the device is waiting for the first time for receive data. or ▶ FS SafetyNET p is in the operating state "FS SafetyNET p in RUN state with minor error". or ▶ ST SafetyNET p is in the operating state "ST SafetyNET p in RUN state with minor error". or ▶ When die DIAG, MBUS, PWR, SD CARD, SNp and TSN LEDs all flash green, the user has activated device identification.

"TSN" LED

The "TSN" LED is on the device, but has no function. It flashes during device identification, but otherwise is off.

Colour	Status	Description
---		Irrelevant
Green		When the DIAG, MBUS, PWR, SD CARD, SNp and TSN LEDs all flash green, the user has activated device identification.

7.14 MDF files and module descriptions (MD)

There is an MDF file (module description file) for each head module, electronic module and also backplane. The MDF file contains a module description (MD) for each firmware version of the module. The module description includes all the properties and configuration options that are valid in the corresponding firmware version. If a new version of a firmware is released, then an additional module description is added to the MDF file.

When there are changes to an MDF file, the version of the MDF file is increased. The module descriptions also have version numbers. If a module description is affected by a change, then its version number is increased.

► MDF version

The MDF version consists of 3 digits:

<MDF major version>.<MDF minor version>.<MDF patch version>

Example: 01.02.01

► MD version

The MD version consists of 3 digits:

<MD major version>.<MD minor version>.<MD patch version>

Example: 03.04.01

The various digits of the MD version are changed, based on the severity of the change. The MD major version is incremented when there is a new firmware version. If only the MD minor and/or MD patch version changes, then the change has no effect on the module's FS data.

To enable a module to be used, the MDF file for that module must be present in the tool PAS4000. The MDF files available at the time the tool is released are automatically installed with the tool. Newer versions of the MDF files and MDF files for new modules can be imported into the tool. The current MDF files are available on the Internet at <https://www.pilz.com/support/downloads/>.



INFORMATION

The MDF files for the head modules are only installed together with the tool PAS4000 and cannot be imported retrospectively.

All the MDF files together form the MDF Library within the tool. The MDF Library contains exactly one MDF file for each module.

When a module is configured, the tool automatically uses the latest firmware version of the module. However, if required, the user can also select an older firmware version.

The tool always uses the module description that corresponds to the firmware. The MDF version and MD version that you are using is displayed in the module's properties.



NOTICE

It is important that the firmware version selected in the configuration corresponds to the module's actual firmware version. This is the only way to guarantee that all the properties configured on the inserted module (actual hardware registry) are also supported, and that a configuration error does not occur when starting the PSS u2 system.

7.15 Error behaviour of the inputs and outputs

7.15.1 Behaviour of the ST outputs

An ST output is shut down when there is an error at this exact output. If the output is error-free, it is reintegrated automatically. Reintegration is regardless of whether or not the output receives valid data. The affected ST output does not affect any other ST outputs.

With ST output modules with advanced diagnostic options, the *OutputValid* bit can be used to determine whether an output is providing valid data. See [Validity of the output signals](#) [107].

7.15.2 Behaviour of the FS outputs

In a PSS 4000 project, the shutdown behaviour of FS outputs can be configured for each individual PSS u2 system. The following options are available:

- ▶ [FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system"](#) [122]
- ▶ [FS shutdown behaviour "Shutdown of all FS outputs on the electronic module"](#) [125]

FS outputs can also be used as auxiliary outputs, see [FS outputs as auxiliary outputs](#) [129].

7.15.2.1 FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system"

In the event of an error on an FS output, used for an FS application, all FS outputs on the PSS u2 system that belong to the PSS u2 system's FS system output group are shut down. Following shutdown, the FS outputs provide substitute values.

When all errors at the FS outputs of the FS system output group have been recovered, the FS system output group can be reintegrated. In other words, the outputs are restored to a state in which they can output process data (whether they actually output process data or substitute values is unknown). Reintegration can occur by starting the module bus (e.g. by restarting the PSS u2 system or a warm reset) or by a falling edge at the *Reintegrate* bit.

The FS shutdown behaviour can be configured in the device properties for each individual PSS u2 system in a PSS 4000 project.



INFORMATION

The FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system" corresponds to the shutdown behaviour of PSS 4000-exclusive devices with firmware versions <1.25.0. The *Reintegrate* bit is a new addition, which does not have to be used however.

FS system output group

Each PSS u2 system forms its own FS system output group. Which outputs belong to the FS system output group depends on the I/O mapping.

The following outputs belong to the FS system output group:

- ▶ FS outputs to which an FS data source is mapped in I/O mapping

- FS outputs to which an FS data source and an ST data source are mapped in I/O mapping (enable principle)

The following outputs do not belong to the FS system output group:

- FS outputs to which no data source is mapped in I/O mapping
- FS outputs to which only an ST data source is mapped in I/O mapping (auxiliary outputs)

States of an FS system output group

An FS system output group switches between the states:

- NOT_READY_FOR_REINTEGRATION
- READY_FOR_REINTEGRATION
- OUTPUT_RUN

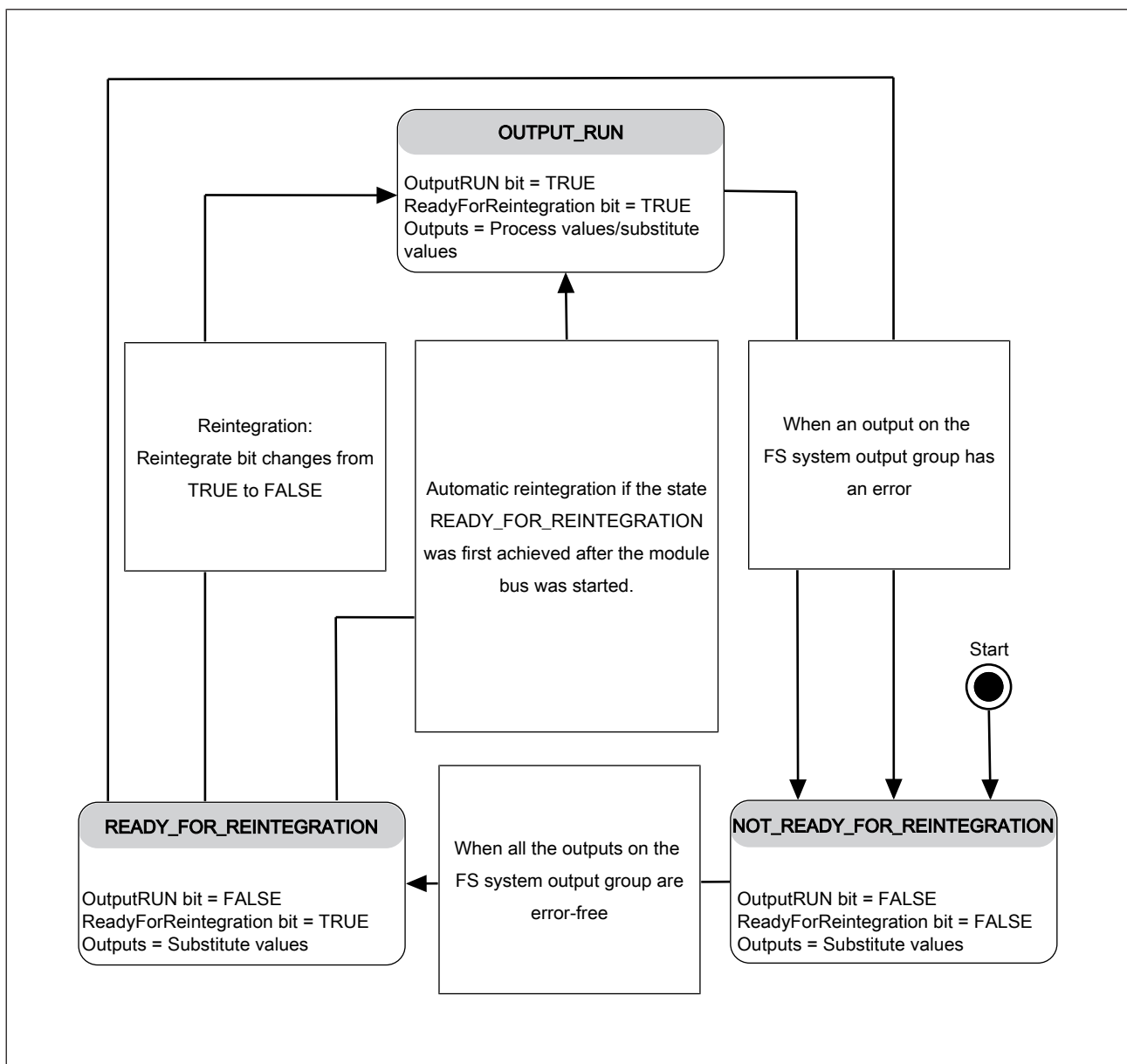


Fig.: States of an FS system output group

If the FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system" is configured for the project, the following information is available for the PSS u2 system in I/O mapping:

- ▶ *OutputRUN* bit (data type: FS_I_DI)
The *OutputRUN* bit is TRUE when no output in the FS system output group is registering an error. It is not possible to conclude from the *OutputRUN* bit whether the outputs are providing substitute values or process values, because only the outputs are tested and not the transmission path of the signal before the output. It may be that some outputs are providing process data and other outputs substitute values.
- ▶ *ReadyForReintegration* bit (data type: FS_I_DI)
The *ReadyForReintegration* bit is TRUE when it is possible to reintegrate the FS system output group.
- ▶ *Reintegrate* bit (data type: ST_O_DO)
The *Reintegrate* bit is used to trigger reintegration of the FS system output group. The *Reintegrate* bit should always be set to FALSE. Only in the state READY_FOR_REINTEGRATION should the bit change from TRUE to FALSE, to trigger reintegration. (Should the *Reintegrate* bit switch from TRUE to FALSE during the state OUTPUT_VALID or NOT_READY_FOR_REINTEGRATION, this will have no effect on the state of the FS system output group.)

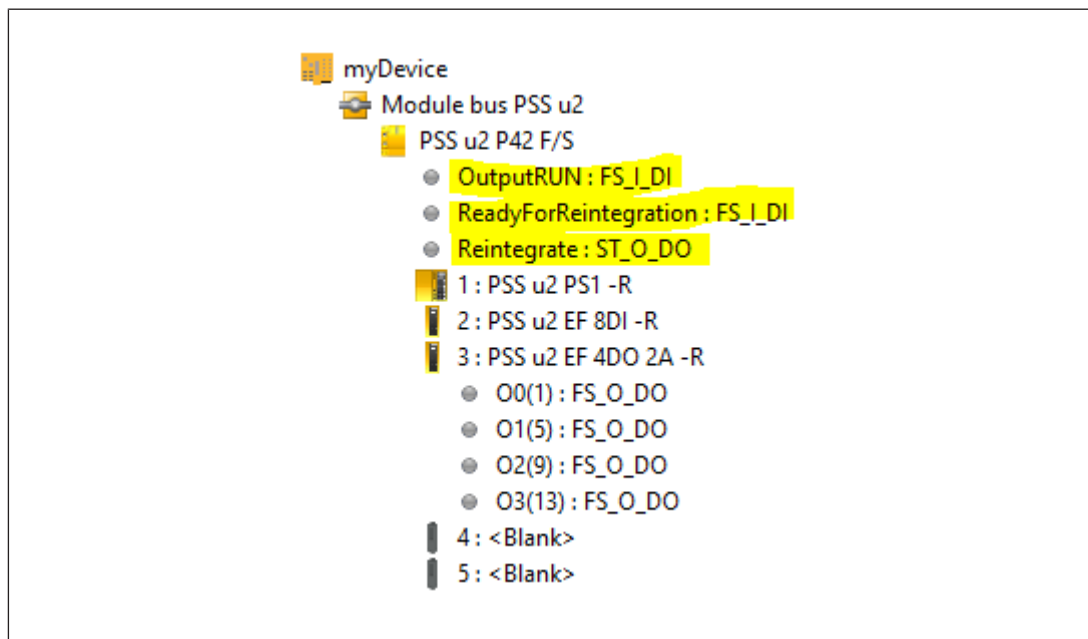


Fig.: Bits in I/O mapping

In addition to the bits available via I/O mapping, it is possible to access the following valid bits in the user program:

- ▶ Valid bit of the *OutputRUN* bit
- ▶ Valid bit of the *ReadyForReintegration* bit

Both valid bits always have the same state. It is sufficient to evaluate only one of the two valid bits. For further information on the valid bit, see [Validity of process data \[359\]](#).

Notes

- ▶ If the FS system output group is in the state NOT_READY_FOR_REINTEGRATION or READY_FOR_REINTEGRATION, the FS outputs provide substitute values.
- ▶ If the FS system output group is in the state OUTPUT_RUN, the FS outputs provide process values or substitute values. It may be that some outputs are providing process data and other outputs substitute values.
- ▶ In the case of an FS output to which an FS data source and an ST data source are mapped in I/O mapping (enable principle), only the FS data source influences the state of the FS system output group.
- ▶ If the FS system output group does not contain any FS outputs, it remains constantly in the state NOT_READY_FOR_REINTEGRATION.
- ▶ If an FS output module is configured on a slot, but the set actual hardware registry differs from the actual hardware registry, then the FS system output group remains in the state NOT_READY_FOR_REINTEGRATION.
- ▶ If the state READY_FOR_REINTEGRATION is achieved, a diagnostic message is issued, which prompts the user to perform reintegration (exception: if the state READY_FOR_REINTEGRATION was first achieved after the module bus was started, the diagnostic message does not appear because reintegration occurs automatically). If the *Reintegrate* bit is reset by the user program, the diagnostic message is superfluous. In this case, the diagnostic message can be suppressed by activating the option **Suppress diagnostic message when Reintegrate bit missing**. This option is individually configurable for each PSS u2 system in the project. The option can be found in the device properties of the PSS u2 system.

7.15.2.2

FS shutdown behaviour "Shutdown of all FS outputs on the electronic module"

In the event of an error on an FS output used for an FS application, only the FS outputs belonging to the same FS module output group are shut down. Following shutdown, the FS outputs provide substitute values.

When all errors on the FS outputs of the FS module output group have been recovered and the FS outputs receive valid data, the FS module output group can be reintegrated. In other words, the outputs are restored to the state in which they output process data.

Reintegration can be performed in 2 ways:

- ▶ Reintegration is triggered by a falling edge at *Reintegrate* Bit.

or

- ▶ Reintegration is triggered by starting the module bus (e.g. by restarting the PSS u2 system or a warm reset). It is a precondition for this way of reintegration that in the configuration of the PSS u2 system the **Automatic reintegration at startup** option is set to "Yes".

The FS shutdown behaviour can be configured in the device properties for each individual PSS u2 system in a PSS 4000 project.

The FS shutdown behaviour "Shutdown of all FS outputs on the electronic module" is suitable for controlling a cross-module shutdown behaviour within the user program. Also, the *OutputValid* bit can be evaluated in the user program, providing valuable information via the state of the outputs.

FS module output groups

Each FS output module forms its own FS module output group. Which outputs belong to the FS module output group depends on the I/O mapping.

The following outputs belong to the FS module output group of an FS output module:

- ▶ FS outputs to which an FS data source is mapped in I/O mapping
- ▶ FS outputs to which an FS data source and an ST data source are mapped in I/O mapping (enable principle)

The following outputs do not belong to the FS module output group:

- ▶ FS outputs to which no data source is mapped in I/O mapping
- ▶ FS outputs to which only an ST data source is mapped in I/O mapping (auxiliary outputs)

States of an FS module output group

An FS module output group switches between the states:

- ▶ NOT_READY_FOR_REINTEGRATION
- ▶ READY_FOR_REINTEGRATION
- ▶ OUTPUT_VALID

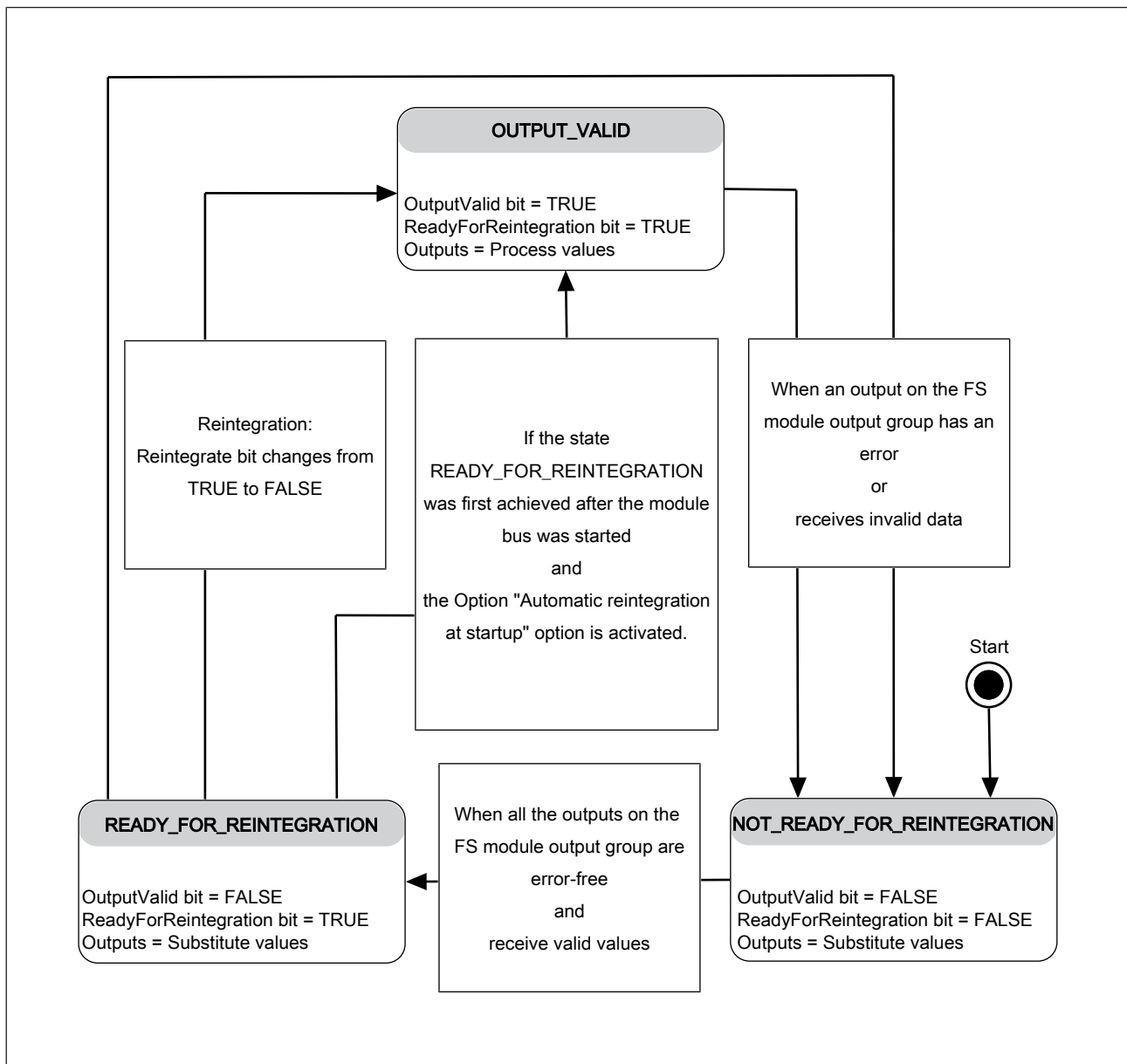


Fig.: States of FS module output groups

If the FS shutdown behaviour "Shutdown of all FS outputs on the electronic module" is configured for the project, the following information is available for each FS module output group in I/O mapping:

► **OutputValid** bit (data type: FS_I_DI)

The *OutputValid* bit is TRUE when all the outputs on the FS module output group are error-free and receive valid values. The outputs provide process values.

If the *OutputValid* bit is FALSE, the outputs provide substitute values. Should the valid bit of the *OutputValid* bit also be FALSE, then the process value in the data source is marked as invalid or access to the process value in the data source is not possible (potential reasons: input error, I/O module error, module bus error, module bus starting, network error).

► **ReadyForReintegration** bit (data type: FS_I_DI)

The *ReadyForReintegration* bit is TRUE when it is possible to reintegrate the FS module output group.

► *Reintegrate* bit (data type: ST_O_DO)

The *Reintegrate* bit is used to trigger reintegration of the FS module output group. The *Reintegrate* bit should always be set to FALSE. Only in the state READY_FOR_REINTEGRATION should the bit change from TRUE to FALSE, to trigger reintegration. (Should the *Reintegrate* bit switch from TRUE to FALSE during the state OUTPUT_VALID or NOT_READY_FOR_REINTEGRATION, this will have no effect on the state of the FS module output group.)

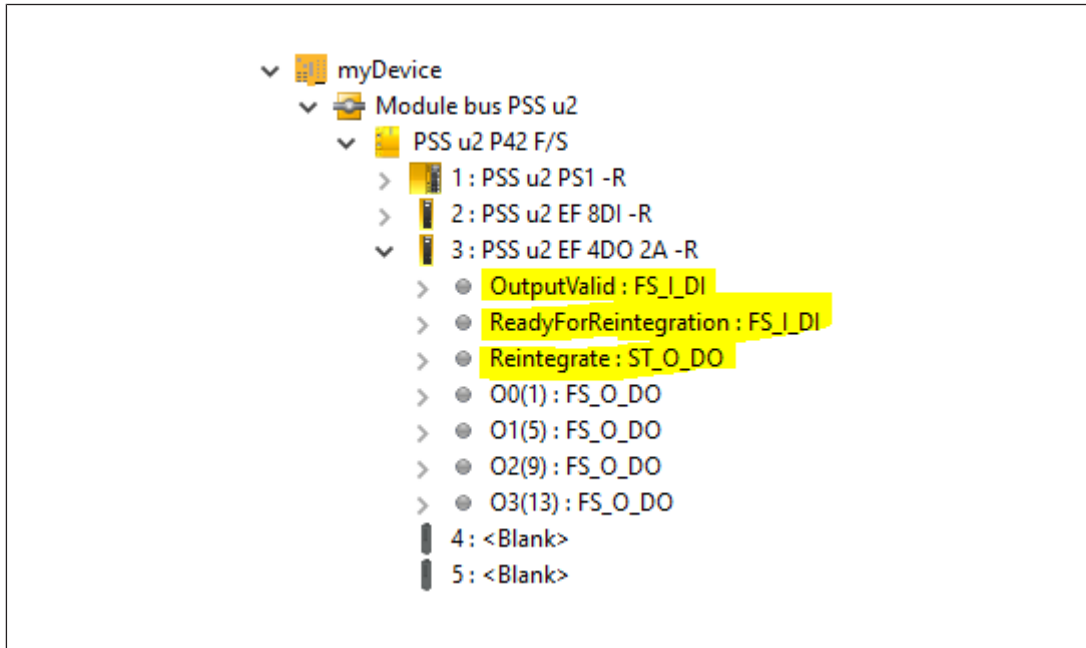


Fig.: Bits in I/O mapping

In addition to the bits available via I/O mapping, it is possible to access the following valid bits in the user program:

- Valid bit of the *OutputValid* bit
- Valid bit of the *ReadyForReintegration* bit

Both valid bits always have the same state. It is sufficient to evaluate only one of the two valid bits. For further information on the valid bit, see [Validity of process data \[359\]](#).

Notes

- The behaviour after the module bus is started depends on the configuration of the **Automatic reintegration at startup** option:
 - "Yes" (default setting)
As soon as the FS module output group reaches the state READY_FOR_REINTEGRATION, it automatically switches to the state OUTPUT_VALID.
 - "No"
When the FS module output group has reached the state READY_FOR_REINTEGRATION, the *Reintegrate* bit must first switch from TRUE to FALSE, to enable the FS module output group to switch to the state OUTPUT_VALID.

This option is individually configurable for each PSS u2 system in the project. The option can be found in the device properties of the PSS u2 system.

- ▶ In the case of an FS output to which an FS data source and an ST data source are mapped in I/O mapping (enable principle), only the FS data source influences the state of the FS module output group.
- ▶ If the FS module output group of an FS output module does not contain any FS outputs, it remains constantly in the state NOT_READY_FOR_REINTEGRATION.
- ▶ If an FS output module is configured on a slot, but the set actual hardware registry differs from the actual hardware registry, then the corresponding FS module output group remains in the state NOT_READY_FOR_REINTEGRATION.

7.15.2.3 FS outputs as auxiliary outputs

If an FS output is used exclusively for non-safety-related tasks, so it is only assigned ST data, then it is an auxiliary output. The behaviour of auxiliary outputs corresponds to that of ST outputs on ST electronic modules, see [Behaviour of the ST outputs](#) [ 122].

7.15.3 Behaviour of the ST and FS inputs

The behaviour of FS inputs and ST inputs is identical.

If there is an error at just one input, the process image of the input is set to the substitute value and the valid bit of the input is set to FALSE. The affected input does not influence any other inputs.

When the error has been recovered, the process image of the input is set to the process value and the valid bit of the input is set to TRUE.

Should an error relate to a whole input module, all the inputs on the module are handled as described.

8 Ethernet

8.1 Auto IP mechanism

The Auto-IP mechanism is used to assign a free IP address automatically and without manual configuration to a device in an Ethernet network. For this purpose, the IANA provides the address range 169.254.0.0/16.

PSS 4000-exclusive devices

The Auto-IP mechanism is activated for PSS 4000-exclusive devices in delivery status or after an original reset. Activation is performed when Booting system sections. During the boot process, the Auto-IP mechanism assigns these devices a free IP address from the range 169.254.1.0 ... 169.254.254.255.

The Auto-IP mechanism is deactivated during the device naming procedure (see [Device naming procedure for PSS 4000-exclusive devices](#) [ 392]).

PSS 4000-operable devices

PSS 4000-operable devices select a free IP address from the range 169.254.0.0/16.

8.2 Ethernet interface

PSS 4000 devices have an Ethernet interface for connecting to the Ethernet. The specification of the Ethernet interface (e.g. internal switch with 2 free switch ports), the connection technology (e.g. RJ45) and the naming of the Ethernet interface (e.g. ETHERNET, SafetyNET p) is dependent on the device. Please refer to the information stated in the operating manual of the device you are using.



NOTICE

Only use Ethernet subscribers (e.g. switches, hubs) that support the auto-sensing/autonegotiation function. Otherwise a duplex conflict (duplex mismatch) may arise, which can cause considerable delays or sporadic data transmission errors.

In the PSS 4000 automation system Ethernet interfaces are used as follows:

- ▶ Connection for data exchange via SafetyNET p
 - [SafetyNET p](#) [ 166]
- ▶ Connection for data exchange via external communication with IP connections such as Modbus/TCP, Raw UDP etc.
 - IP connections
- ▶ Connection of PAS4000 PC (= programming interface)
- ▶ Connection of an OPC Server
 - OPC Server (classic) - Design options
 - OPC Server (PC, PMI) - Design options
- ▶ Connection of an OPC UA Client
 - OPC Server (device-internal) - [Design options](#) [ 706]

8.2.1 RJ45 interface

PSS 4000 devices can have an Ethernet interface with RJ45 connection technology (RJ45 interface). The number of RJ45 interfaces depends on the device. Details of the PSS 4000-exclusive devices and PSS 4000-operable devices from Pilz are available in the operating manual for the relevant unit.

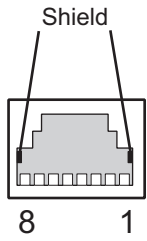
PSS 4000-exclusive devices with two RJ45 interfaces have an internal switch with two free switch ports. As a result, these devices do not need to use any additional switches or hubs.

The RJ45 interfaces on PSS 4000-exclusive devices have an autosensing and autocrossing function.

The autosensing function automatically detects the bit rate during data transmission. The autocrossing function means there is no need to distinguish between patch cable (uncrossed data line connection) and crossover cable (crossover data line connection) on the connection cable. The correct data line connection is automatically established internally. Patch cable can therefore be used as the connection cable for end devices as well as cascading.

8.2.2 RJ45 interface assignment

PSS 4000-exclusive devices with RJ45 interface have the following interface assignment:

RJ45 socket (8-pin)	PIN	Standard	Crossover
	1	TD+ (Transmit+)	RD+ (Receive+)
	2	TD- (Transmit-)	RD- (Receive-)
	3	RD+ (Receive+)	TD+ (Transmit+)
	4	n.c.	n.c.
	5	n.c.	n.c.
	6	RD- (Receive-)	TD- (Transmit-)
	7	n.c.	n.c.
	8	n.c.	n.c.
n.c. = not connected			

Socket for RJ45 connector (plan view)

8.2.3 Connector and connection cable for RJ45

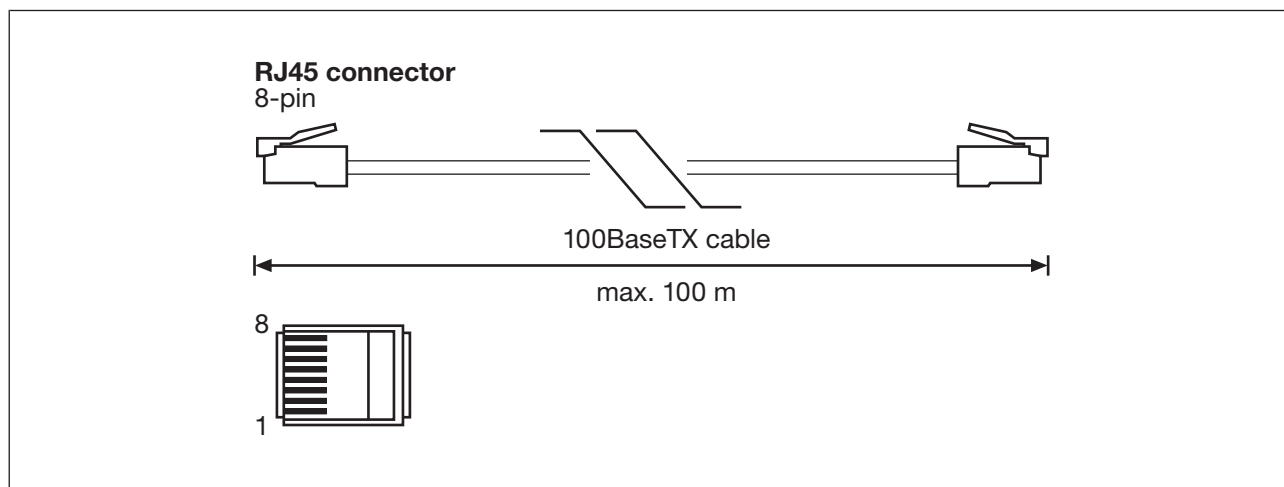


Fig.: Assignment of the RJ45 connector (viewed from below)

Please note the following minimum requirements of the connector and cable:

100BaseT cable

Bit rate	100 MBit/s
Max. segment length	100 m
Medium	STP, S/STP [1] (shielded/double-shielded twisted pair cable) 2 pairs Minimum Category 5 [2]
Connection	Shielded 8-pin RJ45 connector
Topology	Point-to-point/hub (hub, switch)

[1]

For Ethernet applications in an industrial environment, Pilz recommends double-shielded twisted pair cable (S/STP).

[2]

Appropriate TP cable:

Category 5	Up to 100 MHz (10Base-T, 100BaseT4, 100BaseTX)
Category 5e	Up to 100 MHz (1000Base-T)
Category 6	Up to 250 MHz
Category 6a	Up to 500 MHz (10GBase-T)
Category 7	Up to 600 MHz
Category 7a	Up to 1000 MHz

TP cables for Ethernet applications generally have a wave impedance of 100 Ohm.

The connection can be made via a 100BaseTX cable with uncrossed data line connection (patch cable).

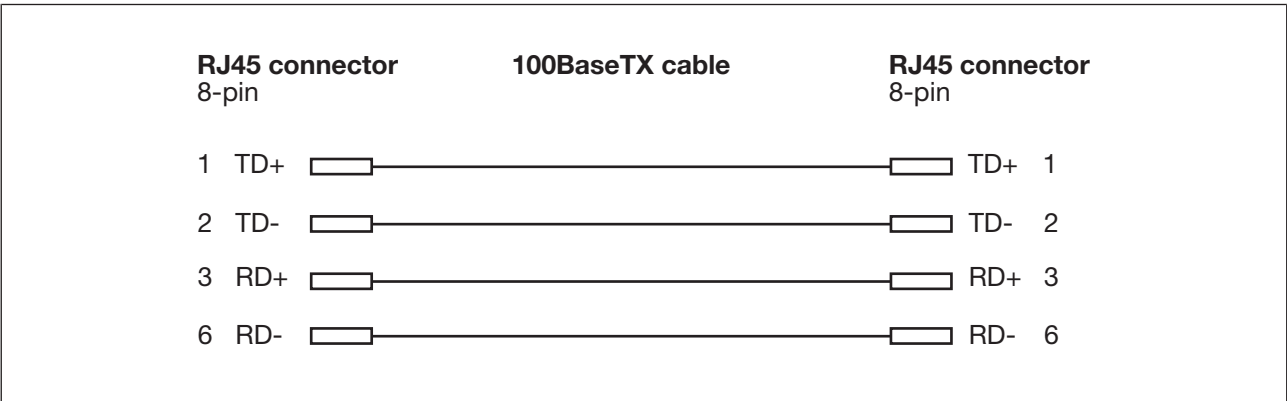


Fig.: 100BaseTX cable (standard)



INFORMATION

If you assemble the connection cable yourself we recommended that you use connector and cable made by the same manufacturer. This is because considerable variances may arise regarding connector dimensions and cable cross sections, due to the permitted tolerances.

8.2.4 Status LEDs on the RJ45 interface

An RJ45 version of the Ethernet interface generally has status LEDs to display various connection and communication states. Details can be found in the operating manual of the relevant device.

8.3 Extended network configuration



INFORMATION

The extended network configuration is only supported by PLC2 controllers and 2nd generation PSS 4000-exclusive devices with a firmware version $\geq$ 1.27.0.

Exactly one IP interface is configured for a PSS 4000-exclusive device during device naming. With the extended network configuration, additional IP interfaces with corresponding IP parameters can be configured if necessary, as well as the properties of MAC frames and the properties of Ethernet interfaces that are physically present (see Ethernet interface). These types of parameters and settings are combined under the term "ETH parameter". IP interfaces that are configured via the extended network configuration are called "ETH interfaces".

The user (e.g. programmer) must create a network configuration file for the extended network configuration. The network configuration file uses JSON as the data format. In this file the user configures the ETH parameters for one or up to 15 ETH interfaces, in accordance with the data model (see [Contents of the network configuration file](#) [140] and [Examples of network configuration files](#) [158]). The configurations in the network configuration file not only affect outgoing data packets (send data) but also incoming data packets (see [Requirements of receive data](#) [165]).

The extended network configuration is independent of the device project; i.e. the network configuration file is downloaded separately from the device project. The network configuration file is downloaded in the Online Network Editor of PAS4000. The user is free to select the name of the network configuration file. During download, PAS4000 renames the file "net.json". Once downloaded, the network configuration file can be found in the subdirectory /cfg of the SD card of the PSS 4000-exclusive device (see [Memory structure](#) [610]). To transfer data, a restart or reboot must be carried out (see [Activation and deactivation of the network configuration](#) [163]).

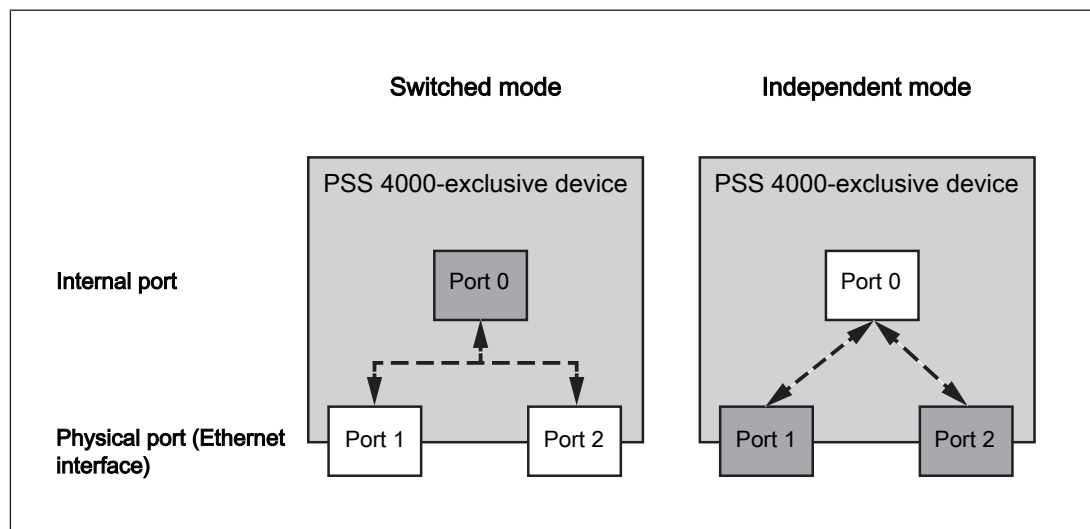
The configuration options are described separately. Further information about the configuration options is available under [Contents of the network configuration file](#) [140].

8.3.1 Ports' Ethernet mode

If a PSS 4000-exclusive device has several Ethernet interfaces (see Ethernet interface), on 2nd generation PSS 4000-exclusive devices, these interfaces can be operated either in the Ethernet mode "Switched Mode" or "Independent Mode". In this context, the name "Ethernet interface" is synonymous with "physical port". On a PSS u2 P42 F/S -R, the Ethernet interfaces are labelled/printed as "X3" and "X4", for example.

The Ethernet mode can be configured via a network configuration file. If Ethernet mode is configured via the network configuration file, then this mode applies to all interfaces that use the Ethernet interfaces or physical ports. These interfaces are:

- ▶ ETH interfaces that are configured in the network configuration file
- ▶ IP interfaces that are generated using the system control block pssCreateIPInterface
- ▶ The IP interface that is configured during device naming



Legend

Internal port	Hardware port for device-internal use, without interface socket
Physical port	Hardware port with interface socket
Switched mode	The physical ports (in this case: port 1 and port 2) operate as ports of a layer 2 switch. The switch is situated internally within the device. Within the device, the physical ports are connected to the internal port (in this case: port 0) of the layer 2 switch. Switched mode enables Ethernet frames to be exchanged directly between the physical ports, i.e. Ethernet frames are forwarded directly from one port to another port in the same network (cross traffic). The internal switch of the PSS 4000-exclusive device either selects the physical port for send data using its MAC table or sends the data via all physical ports, if the IP address cannot be resolved in the MAC table.

Independent mode (2nd generation PSS 4000-exclusive devices only)	The physical ports operate as separate network interfaces. Independent mode prevents Ethernet frames from being exchanged directly between the physical ports, i.e. "cross traffic" is prevented. The PSS 4000-exclusive device sends data via the configured physical port (see ETH parameter "<i>physical_port</i>" under ETH_INTERFACE [ 144]). Note: During device naming and when using the PSS 4000 system control block <i>pssCreateIPInterface()</i>, only the IP parameters can be configured for an IP interface; i.e. the user has no access to the configuration of a corresponding physical port. As independent mode also impacts these interfaces, the physical port with the least significant name is used as port in these cases (default setting). Example: on a PSS u2 P42 F/S -R with the names X3 and X4 for physical ports, this is physical port X3.



INFORMATION

Ethernet mode when using an device-internal OPC UA server

If the OPC UA server is used on a PSS 4000-exclusive device with system section OPC UA server, the device can only be used in switched mode.

8.3.2 Contents of the network configuration file

The network configuration file uses the JSON data model in accordance with ECMA-404, 2nd edition, published in December 2017.

The diagram below contains an overview of the layout and data structure of the network configuration file, for orientation. You'll find a sample file under [Network configuration file with switched mode](#) [158].

If you use the sample file as a template for your application-specific network configuration file, a deeper understanding of the data model is not strictly necessary, but merely serves as additional information.

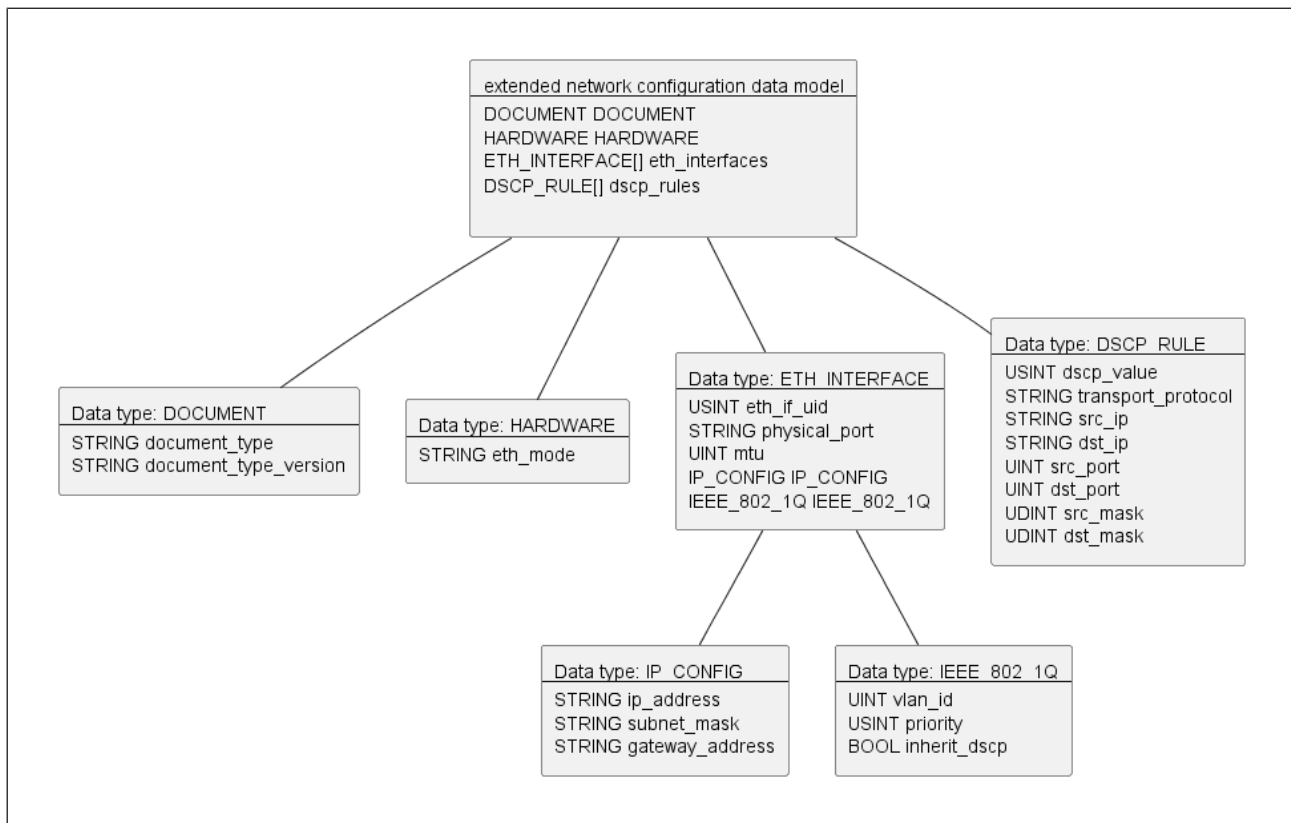


Fig.: Overview of the used data model

The following options are available with the extended network configuration:

- ▶ Reference document type and version of the network configuration file (see [DOCUMENT](#) [142])
- ▶ Define Ethernet mode for ports (see [Ports' Ethernet mode](#) [138] and [HARDWARE](#) [143])
- ▶ Define ETH parameters for IP interfaces (see [ETH_INTERFACE](#) [144])
- ▶ Define DSCP rule set for classification of network traffic (see [DSCP_RULE](#) [148])



INFORMATION

The following applies generally for creating the network configuration file:

- A parameter consists of the parameter name and parameter value (e.g. `"eth_mode": "switched"` or `"eth_if_uid": 2`).
- The parameter name for a parameter must always be configured with the parameter value; i.e. the network configuration file may not contain parameters without a parameter value.
- Parameters with default values do not need to be configured in the network configuration file. In this case, the default values are used.
- Optional parameters may be omitted.

8.3.2.1 DOCUMENT

DOCUMENT contains information about the document.

Example

```
"document": {  
  "type": "net_config",  
  "type_version": "V1.0.0"  
}
```

"document": {...}			
	Parameter	Value/value range	Meaning
	type	net_config	Pilz-defined type of document and therefore data model. The value may not be modified!
	type_version	V1.0.0	Pilz-defined version of document type and therefore version of the data model of the extended network configuration. The value may not be modified!

8.3.2.2 **HARDWARE**

HARDWARE defines the Ethernet mode of the ports for a PSS 4000-exclusive device with several physical ports.

Example

```
"hardware": {  
    "eth_mode": "switched"  
}
```

"hardware": {...}			
	Parameter	Value/value range	Meaning
	eth_mode	switched (Default value)	see Ports' Ethernet mode [ 138]
		independent	



NOTICE
Independent mode only for 2nd generation PSS 4000-exclusive devices

The configuration *"eth_mode": "independent"* exclusively affects the physical ports of 2nd generation PSS 4000-exclusive devices.

In contrast, if independent mode is configured on a PLC2 controller, the network configuration file is evaluated as faulty and transfer of the configuration is aborted, with an error message.

8.3.2.3

ETH_INTERFACE

ETH_INTERFACE can contain the ETH parameters of up to 15 ETH interfaces. The PSS 4000-exclusive device creates an ETH interface for each ETH interface configured in the network configuration file. For example, the IP addresses of the ETH interfaces can be used by PAS4000, the corresponding system sections (e.g. IP connections system section) and the corresponding system control blocks.

Example

```
"eth_interfaces": [
  {
    "eth_if_uid": 2,
    "physical_port": "X3",
    "mtu": 1500,
    "ip_config": {
      "ip_address": "192.168.2.210",
      "subnet_mask": "255.255.255.0",
      "gateway_address": "0.0.0.0"
    }
  },
  {
    "eth_if_uid": 3,
    "physical_port": "X3",
    "mtu": 576,
    "ieee_802_1q": {
      "vlan_id": 4,
      "priority": 3,
      "inherit_dscp": false
    },
    "ip_config": {
      "ip_address": "192.168.3.210",
      "subnet_mask": "255.255.255.0",
      "gateway_address": "0.0.0.0"
    }
  },
  (...)
]
```

"eth_interfaces": [{...}, ... {...}]			
ETH parameter	Value/value range	Meaning	
eth_if_uid	1 ...15	Unique identifier (UID) of the ETH interface created using the stated ETH parameters. For example, this parameter is referenced when using the system control block <code>pssUseETHInterface</code>.	
physical_port	<i><product-dependent name of port></i> (Default setting: <i><Port with the least significant name></i>) Example: on a PSS u2 P42 F/S -R with the names X3 and X4, X3 is the default setting.)	The Ethernet interfaces of a PSS 4000-exclusive device are printed with a name (e.g. X3, X4). This name must be stated here. Note: ▶ <i>"eth_mode": "switched"</i>: – <i>"physical_port"</i> is ignored and all Ethernet interfaces of the PSS 4000-exclusive device use the configured ETH parameters. ▶ <i>"eth_mode": "independent"</i>: – Data is sent and received via the port with the configured name. – Independent mode can only be configured for 2nd generation PSS 4000-exclusive devices. – If <i>"physical_port"</i> is not configured, then the default setting is used. Please note: The configuration of independent mode also applies for interfaces configured during device naming or when using the PSS 4000 system control block <code>pssCreateIPInterface()</code>. However, the physical port cannot be configured for these interfaces via <i>physical_port</i> (see Independent mode under Ports' Ethernet mode [138]). The default setting for <i>"physical_port"</i> is used for these interfaces.	
mtu	576 ... 1500 (Default value: 1500)	Maximum transmission unit (MTU) The MTU is the maximum payload of the Ethernet frame that can be transmitted via this ETH interface in a single transmission process, in Bytes.	

"eth_interfaces": [{...}, ... {...}]			
	ETH parameter	Value/value range	Meaning
	ieee_802_1q	Configuration in accordance with IEEE 802.1Q, to evaluate the VLAN identifier (VID) from the 802.1Q tag when receiving, and to set the VLAN identifier (VID) and Priority Code Point (PCP) in the 802.1Q tag when sending. The configuration of ieee_802_1q is optional. If the parameter is not configured, the tag in the Ethernet frame is not set.	
	vlan_id	1 ... 4093	VLAN identifier (VID) of the 802.1Q tag in the Ethernet frame For incoming data packets, please note the information under Requirements of receive data  165]
	priority	0 ... 7 (Default value: 0)	Priority Code Point (PCP) of the 802.1Q tag in the Ethernet Frame Note: ▶ If a value is configured for both "<i>priority</i>" and "<i>inherit_dscp</i>": <i>true</i>, then the 3 leading bits from the DSCP header are copied to the PCP of the 802.1Q tag. ▶ If no value is configured for "<i>priority</i>" and "<i>inherit_dscp</i>": <i>false</i>, then the default value of "<i>priority</i>" is used.
	inherit_dscp	false (Default value)	Configuration of the priority in the DSCP field of the IP header Note: ▶ "<i>inherit_dscp</i>": <i>true</i> The 3 priority bits (3 leading bits in the DSCP field) are copied into the PCP parameter (see "<i>priority</i>"). ▶ "<i>inherit_dscp</i>": <i>false</i> The value of "<i>priority</i>" is used.
		true	

"eth_interfaces": [{...}, ... {...}]			
	ETH parameter	Value/value range	Meaning
	ip_config	ETH parameter for configuration of the IP address, subnet mask and Gateway address for incoming and outgoing data (receive/send data) on this ETH interface	
	ip_address	<valid IPv4 address> (see [*1])	IP address, via which the ETH interface sends data and can receive data For incoming data packets, please note the information under Requirements of receive data  165)
	subnet_mask	<valid subnet mask> (see [*2])	Configuration of subnetwork ▶ Data that is sent to an IPv4 target address via this ETH interface and is in the same subnetwork is sent directly to this target address ▶ Data that is sent to an IPv4 target address via this ETH interface and is not in the same subnetwork is sent to the gateway address that is configured for the ETH interface. For incoming data packets, please note the information under Requirements of receive data  165).
	gateway_address	<valid gateway address> Default value: 0.0.0.0 (see [*3])	IPv4 address of the gateway to external subnetworks Configuration of "0.0.0.0": No gateway is used.

[*1] valid IPv4 address

Format:

- ▶ 4 integer values (0 ... 255)
- ▶ Integer values in decimal notation, separated by dots (e.g. 192.168.0.210)

[*2] valid subnet mask

Format:

- ▶ 4 integer values (0 ... 255)
- ▶ Integer values in decimal notation, separated by dots (e.g. 255.255.255.0)

[*3] valid gateway address

Format:

- ▶ 4 integer values (0 ... 255)
- ▶ Integer values in decimal notation, separated by dots (e.g. 0.0.0.0)

8.3.2.4 DSCP_RULE

DSCP rules by which the DSCP bits are set in the TOS field of the IP header of outgoing IP packets (send data).

The DSCP rules are evaluated from top to bottom according to their order in the network configuration file. The first appropriate rule is applied. All other rules are ignored.

A DSCP rule is appropriate if the following rules apply during evaluation:

- ▶ The transport protocol used for the send data corresponds to the transport protocol in the DSCP rule.
- ▶ The IPv4 source address corresponds to the source address in the DSCP rule.
The source address of the DSCP rule is evaluated by means of its mask. When Mask 0.0.0.0 is configured, this rule applies for any IPv4 source addresses.
- ▶ The IPv4 target address corresponds to the target address of the DSCP rule.
The target address of the DSCP rule is evaluated by means of its mask. When Mask 0.0.0.0 is configured, the rule applies for any IPv4 target addresses.
- ▶ When TCP or UDP is configured for the transport protocol:
 - The source port entered in the header of the transport protocol for send data corresponds to the source port in the DSCP rule.
 - The target port entered in the header of the transport protocol for send data corresponds to the target port in the DSCP rule.

If none of the configured DSCP rules apply, zeros are entered in the DSCP bits.

Example

```
"dscp_rules": [
  {
    "dscp_value": 4,
    "transport_protocol": "tcp",
    "dst_ip": "192.168.5.210",
    "dst_mask": "255.255.255.0",
    "dst_port": 0,
    "src_ip": "0.0.0.0",
    "src_mask": "255.255.255.0",
    "src_port": 0
  },
  {
    "dscp_value": 7,
    "transport_protocol": "tcp",
    "dst_ip": "192.168.5.210",
    "dst_mask": "255.255.255.0",
    "dst_port": 80,
    "src_ip": "192.168.5.170",
    "src_mask": "255.255.255.0",
    "src_port": 0
  },
]
```

(...)

]

"dscp_rules": [{...}, ... {...}]			
Parameter	Value/value range	Meaning	
dscp_value	0 ... 63	DSCP value The value is entered in the DSCP field of the IP header, provided the rule applies.	
transport_protocol	tcp	Transport protocol used for outgoing data (send data)	
	udp		
	any (Default value)	<i>"transport_protocol": any</i> The rule does not take into account the transport protocol or the configuration for the source port and target port.	
dst_ip	<valid IPv4 address> (Default value: 0.0.0.0) (see [*1])	IPv4 address of the data target When 0.0.0.0 is configured, this rule applies for any IPv4 target addresses.	
dst_mask	<valid subnet mask> (Default value: 255.255.255.255) (see [*2])	Mask for the IPv4 target address	
dst_port	0 ... 65535 (Default value: 0)	Number of the target port (TCP port/UDP port) in the header of the transport protocol. When 0 is configured, this rule applies for any target ports.	
src_ip	<valid IPv4 address> (Default value: 0.0.0.0) (see [*1])	IPv4 address of the data source When 0.0.0.0 is configured, this rule applies for any IPv4 source addresses.	
src_mask	<valid subnet mask> (Default value: 255.255.255.255) (see [*3])	Mask for the IPv4 source address	
src_port	0 ... 65535 (Default value: 0)	Number of the source port (TCP port/UDP port) in the header of the transport protocol. When 0 is configured, this rule applies for any source ports.	

[*1] valid IPv4 address

Format:

- ▶ 4 integer values (0 ... 255)
- ▶ Integer values in decimal notation, separated by dots (e.g. 192.168.0.210)

[*2] valid mask

Format:

- ▶ 4 integer values (0 ... 255)
- ▶ Integer values in decimal notation, separated by dots (e.g. 255.255.255.0)

[*3] valid gateway address

Format:

- ▶ 4 integer values (0 ... 255)
- ▶ Integer values in decimal notation, separated by dots (e.g. 0.0.0.0)

8.3.3 Validation of the network configuration file

If the network configuration file was created by the user, the file can be checked for syntax errors using an appropriate parser or a comparable tool. The following JSON schema is available for this purpose.

The JSON schema can be transferred to the tool you are using via the clipboard, for example. The tool can then check the network configuration file using the JSON schema.



NOTICE

In principle you can copy the following JSON schema to the tool you are using via the clipboard. This procedure may mean that the JSON schema needs editing. To avoid editing we recommend you copy the identical JSON schema from the PAS4000 Online Help.

If you wish to copy the JSON schema from this PDF file, please note the following in particular:

- Special characters are interpreted depending on the PDF Reader you are using. For example, leading zeros that are used for structuring can be removed.
- Paragraph format ("hard break") is used as the control character for line breaks. This can cause a malfunction at various points, depending on the tool you are using. Remove paragraph formats that are between the inverted commas in the stated values of parameters or descriptions, for example.

JSON schema

```
{
  "type": "object",
  "description": "This defines the structure of the network configuration file for the extended network configuration.",
  "title": "extended network configuration",
  "properties": {
    "document": {
      "type": "object",
      "description": "This defines basic information about the document.",
      "title": "document",
      "properties": {
        "type": {
          "title": "type",
          "description": "Document type defined by Pilz.",
          "type": "string",
          "enum": [
            "net_config"
          ]
        },
        "type_version": {
```

```
        "title": "type_version",
        "description": "Version of the data model for
the network configuration file, this version is defined by Pilz.",
        "type": "string",
        "enum": [
            "V1.0.0"
        ]
    },
    },
    "required": [ "type", "type_version" ]
},
"hardware": {
    "type": "object",
    "description": "This defines how the physical ports are
used.",
    "title": "hardware",
    "properties": {
        "eth_mode": {
            "title": "eth_mode",
            "description": "This defines the Ethernet mode
for devices with multiple physical ports. For 2nd generation
devices both modes are allowed, for 1st generation devices with
PLC2 only \"switched\" is allowed.",
            "type": "string",
            "default": "switched",
            "enum": [
                "independent",
                "switched"
            ]
        }
    }
},
},
"eth_interfaces": {
    "type": "array",
    "description": "This array contains configurations of 0
to 15 Ethernet interfaces.",
    "title": "eth_interfaces",
    "minItems": 0,
    "maxItems": 15,
    "items": {
        "type": "object",
        "description": "",
        "title": "eth_interfaces",
        "properties": {
            "eth_if_uid": {
                "title": "eth_if_uid",
                "description": "This defines a unique iden-
tifier for the Ethernet interface. This UID is used to address the
Ethernet interface in some system control blocks.",
                "type": "number",
```

```
        "minimum": 1,
        "maximum": 15
    },
    "physical_port": {
        "title": "physical_port",
        "description": "This defines the physical
port which shall provide the configured Ethernet interface. This
setting is only needed for eth_mode=independent. For
eth_mode=switched the setting of physical_port is ignored because
all physical ports provide the Ethernet interface.",
        "type": "string",
        "pattern": "^[X][0-9]{1})$"
    },
    "mtu": {
        "title": "mtu",
        "description": "This defines the maximum
transmission unit (MTU). The MTU is the maximum payload of the Eth-
ernet frame, which can be transferred in a single transaction. The
unit is Byte.",
        "type": "number",
        "default": 1500,
        "minimum": 576,
        "maximum": 1500
    },
    "ip_config": {
        "type": "object",
        "description": "This defines the IP para-
meters for the Ethernet interface.",
        "title": "ip_config",
        "properties": {
            "ip_address": {
                "title": "ip_address",
                "description": "This defines the IP
address in dot-decimal notation.",
                "type": "string",
                "format": "ipv4",
                "pattern": "^(?: (25[0-5] | 2[0-4]
[0-9] | 1[0-9] [0-9] | [1-9] [0-9] | [0-9]) (\\. (?!$) | $)) {4}$"
            },
            "subnet_mask": {
                "title": "subnet_mask",
                "description": "This defines the
subnet mask in dot-decimal notation.",
                "type": "string",
                "format": "ipv4",
                "pattern": "^(?: (25[0-5] | 2[0-4]
[0-9] | 1[0-9] [0-9] | [1-9] [0-9] | [0-9]) (\\. (?!$) | $)) {4}$"
            },
            "gateway_address": {
                "title": "gateway_address",
```

```

        "description": "This defines the
gateway address to external subnets. Setting is optional. If the
value is set to \"0.0.0.0\" or omitted no gateway_address is
used.",
        "type": "string",
        "format": "ipv4",
        "default": "0.0.0.0",
        "pattern": "^(?:25[0-5]|2[0-4]
[0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    },
    "required": [ "ip_address", "subnet_mask" ]
},
"ieee_802_1q": {
    "type": "object",
    "description": "This defines the 802.1Q
tag. The tag allows to set the VLAN-Identifier (VID) and the Prior-
ity Code Point (PCP). The parameter \"ieee_802_1q\" shall be omit-
ted when no 802.1Q field shall be added to the tag of the ethernet
frame.",
    "title": "ieee_802_1q",
    "properties": {
        "vlan_id": {
            "title": "vlan_id",
            "description": "This defines the
VLAN-Identifier (VID) of the 802.1Q tag in the ethernet frame.",
            "type": "number",
            "minimum": 1,
            "maximum": 4093
        },
        "priority": {
            "title": "priority",
            "description": "This defines the
Priority Code Point (PCP) of the 802.1Q tag in the ethernet
frame.",
            "type": "number",
            "default": 0,
            "minimum": 0,
            "maximum": 7
        },
        "inherit_dscp": {
            "title": "inherit_dscp",
            "description": "This defines the
priority in the DSCP field of the IP header. If the parameter is
set to \"true\", the precedence bits/class selector codepoints (3
leading bits) of DSCP field are copied to the parameter Priority
Code Point (PCP) of the 802.1Q tag in the ethernet frame. This
overrides possible settings of the parameter \"priority\". If para-
meter is set to \"false\", the value of the parameter \"priority\"
is used.",

```



```

        "type": "boolean",
        "default": false,
        "enum": [
            true,
            false
        ]
    }
}
},
"required": [ "eth_if_uid", "ip_config" ]
}
},
"dns_server": {
    "type": "array",
    "description": "Array of IP addresses of Domain Name
System servers that are used. None or several IP addresses may be
declared. The element \"dns_server\" shall be omitted when no Do-
main Name System server is defined.",
    "title": "dns_server",
    "minItems": 0,
    "maxItems": 2,
    "items": {
        "title": "dns_server",
        "description": "This defines the IP address of a
Domain Name System server. When used this value overrides declara-
tions in the naming data.",
        "type": "string",
        "format": "ipv4",
        "default": "0.0.0.0",
        "pattern": "^(?: (25[0-5]|2[0-4][0-9]|1[0-9][0-9]|
[1-9][0-9]|[0-9]) (\\.(?!$)|$)) {4}$"
    }
},
"dscp_rules": {
    "type": "array",
    "description": "This array defines the DSCP rules. The
DSCP bits in the TOS field of the IP header for outgoing IP pack-
ages (send data) are set according to these DSCP rules.",
    "title": "dscp_rules",
    "minItems": 0,
    "maxItems": 64,
    "items": {
        "type": "object",
        "description": "",
        "title": "dscp_rule",
        "properties": {
            "dscp_value": {
                "title": "dscp_value",

```

```

        "description": "This defines the DSCP value
that is written to the TOS field of the IP header in case this DSCP
rule matches. This value is mandatory for every DSCP rule.",
        "type": "number",
        "minimum": 0,
        "maximum": 63
    },
    "transport_protocol": {
        "title": "transport_protocol",
        "description": "This defines the used
transport protocol.",
        "type": "string",
        "default": "any",
        "enum": [
            "tcp",
            "udp",
            "any"
        ]
    },
    "src_ip": {
        "title": "src_ip",
        "description": "This defines the source IP
address in dot-decimal notation. When omitted this rule applies for
any source IP address.",
        "type": "string",
        "format": "ipv4",
        "default": "0.0.0.0",
        "pattern": "(?: (?: (?: 25[0-5] | 2[0-4][0-9] | 1[0-9]
[0-9] | [1-9][0-9] | [0-9]) (\\. (?!$) | $)) {4} $"
    },
    "dst_ip": {
        "title": "dst_ip",
        "description": "This defines the target IP
address in dot-decimal notation. When omitted this rule applies for
any target IP address.",
        "type": "string",
        "format": "ipv4",
        "default": "0.0.0.0",
        "pattern": "(?: (?: (?: 25[0-5] | 2[0-4][0-9] | 1[0-9]
[0-9] | [1-9][0-9] | [0-9]) (\\. (?!$) | $)) {4} $"
    },
    "src_mask": {
        "title": "src_mask",
        "description": "This defines the mask for
the source IP addresses in dot-decimal notation.",
        "type": "string",
        "format": "ipv4",
        "default": "255.255.255.255",
        "pattern": "(?: (?: (?: 25[0-5] | 2[0-4][0-9] | 1[0-9]
[0-9] | [1-9][0-9] | [0-9]) (\\. (?!$) | $)) {4} $"
    }

```

```
    },
    "dst_mask": {
      "title": "dst_mask",
      "description": "This defines the mask for
the target IP addresses in dot-decimal notation.",
      "type": "string",
      "format": "ipv4",
      "default": "255.255.255.255",
      "pattern": "^(?: (25[0-5] | 2[0-4] [0-9] | 1[0-9]
[0-9] | [1-9] [0-9] | [0-9]) (\\. (?!$) | $)) {4}$"
    },
    "src_port": {
      "title": "src_port",
      "description": "This defines the source
port in the header of the transport protocol (TCP, UDP). When omit-
ted or 0 this rule applies for any source port.",
      "type": "number",
      "default": 0,
      "minimum": 0,
      "maximum": 65535
    },
    "dst_port": {
      "title": "dst_port",
      "description": "This defines the target
port in the header of the transport protocol (TCP, UDP). When omit-
ted or 0 this rule applies for any target port.",
      "type": "number",
      "default": 0,
      "minimum": 0,
      "maximum": 65535
    }
  }
}
},
"required": [ "document", "hardware" ]
}
```

8.3.4 Examples of network configuration files

8.3.4.1 Network configuration file with switched mode



NOTICE

In principle you can copy the following listing to the tool you are using via the clipboard. This procedure may mean that the listing needs editing. To avoid editing we recommend you copy the identical listing from the PAS4000 Online Help.

If you wish to copy the listing from this PDF file, please note the following in particular:

- Special characters are interpreted depending on the PDF Reader you are using. For example, leading zeros that are used for structuring can be removed.
- Paragraph format ("hard break") is used as the control character for line breaks. This can cause a malfunction at various points, depending on the tool you are using. Remove paragraph formats that are between the inverted commas in the stated values of parameters or descriptions, for example.

```
{
  "document": {
    "type": "net_config",
    "type_version": "V1.0.0"
  },
  "hardware": {
    "eth_mode": "switched"
  },
  "eth_interfaces": [
    {
      "eth_if_uid": 1,
      "mtu": 1500,
      "ip_config": {
        "ip_address": "192.168.0.210",
        "subnet_mask": "255.255.255.0",
        "gateway_address": "0.0.0.0"
      }
    },
    {
      "eth_if_uid": 2,
      "mtu": 1500,
      "ip_config": {
        "ip_address": "192.168.2.210",
        "subnet_mask": "255.255.255.0",
        "gateway_address": "0.0.0.0"
      }
    }
  ]
}
```

```
,
{
  "eth_if_uid": 3,
  "mtu": 576,
  "ieee_802_1q": {
    "vlan_id": 4,
    "priority": 3,
    "inherit_dscp": false
  },
  "ip_config": {
    "ip_address": "192.168.3.210",
    "subnet_mask": "255.255.255.0",
    "gateway_address": "0.0.0.0"
  }
},
{
  "eth_if_uid": 14,
  "mtu": 1500,
  "ieee_802_1q": {
    "vlan_id": 10,
    "priority": 5,
    "inherit_dscp": true
  },
  "ip_config": {
    "ip_address": "192.168.10.210",
    "subnet_mask": "255.255.255.0",
    "gateway_address": "0.0.0.0"
  }
},
{
  "eth_if_uid": 5,
  "mtu": 1500,
  "ieee_802_1q": {
    "vlan_id": 11,
    "priority": 5,
    "inherit_dscp": true
  },
  "ip_config": {
    "ip_address": "192.168.11.210",
    "subnet_mask": "255.255.255.0",
    "gateway_address": "0.0.0.0"
  }
},
{
  "eth_if_uid": 12,
  "mtu": 1500,
  "ieee_802_1q": {
    "vlan_id": 12,
    "priority": 5,
    "inherit_dscp": true
  },
  "ip_config": {
    "ip_address": "192.168.12.210",
    "subnet_mask": "255.255.255.0",
    "gateway_address": "0.0.0.0"
  }
}
```

```
    },
    "ip_config": {
        "ip_address": "192.168.12.210",
        "subnet_mask": "255.255.255.0",
        "gateway_address": "0.0.0.0"
    }
},
{
    "eth_if_uid": 15,
    "mtu": 1500,
    "ieee_802_1q": {
        "vlan_id": 13,
        "priority": 5,
        "inherit_dscp": true
    },
    "ip_config": {
        "ip_address": "192.168.13.210",
        "subnet_mask": "255.255.255.0",
        "gateway_address": "0.0.0.0"
    }
}
],
"dscp_rules": [
    {
        "dscp_value": 4,
        "transport_protocol": "tcp",
        "dst_ip": "192.168.5.210",
        "dst_mask": "255.255.255.0",
        "dst_port": 0,
        "src_ip": "0.0.0.0",
        "src_mask": "255.255.255.0",
        "src_port": 0
    },
    {
        "dscp_value": 7,
        "transport_protocol": "tcp",
        "dst_ip": "192.168.5.210",
        "dst_mask": "255.255.255.0",
        "dst_port": 80,
        "src_ip": "192.168.5.170",
        "src_mask": "255.255.255.0",
        "src_port": 0
    }
]
}
```

8.3.4.2 Network configuration file with independent mode



NOTICE

In principle you can copy the following listing to the tool you are using via the clipboard. This procedure may mean that the listing needs editing. To avoid editing we recommend you copy the identical listing from the PAS4000 Online Help.

If you wish to copy the listing from this PDF file, please note the following in particular:

- Special characters are interpreted depending on the PDF Reader you are using. For example, leading zeros that are used for structuring can be removed.
- Paragraph format ("hard break") is used as the control character for line breaks. This can cause a malfunction at various points, depending on the tool you are using. Remove paragraph formats that are between the inverted commas in the stated values of parameters or descriptions, for example.

```
{
  "document": {
    "type": "net_config",
    "type_version": "V1.0.0"
  },
  "hardware": {
    "eth_mode": "independent"
  },
  "eth_interfaces": [
    {
      "eth_if_uid": 2,
      "physical_port": "X3",
      "mtu": 1500,
      "ip_config": {
        "ip_address": "192.168.2.210",
        "subnet_mask": "255.255.255.0",
        "gateway_address": "0.0.0.0"
      }
    },
    {
      "eth_if_uid": 3,
      "physical_port": "X3",
      "mtu": 1500,
      "ip_config": {
        "ip_address": "192.168.3.210",
        "subnet_mask": "255.255.255.0",
        "gateway_address": "0.0.0.0"
      }
    }
  ]
}
```

```
    "ieee_802_1q": {
      "vlan_id": 3,
      "priority": 4,
      "inherit_dscp": false
    }
  },
  {
    "eth_if_uid": 4,
    "physical_port": "X4",
    "mtu": 1500,
    "ip_config": {
      "ip_address": "192.168.4.210",
      "subnet_mask": "255.255.255.0",
      "gateway_address": "0.0.0.0"
    },
    "ieee_802_1q": {
      "vlan_id": 4,
      "priority": 5,
      "inherit_dscp": false
    }
  },
  {
    "eth_if_uid": 6,
    "physical_port": "X4",
    "mtu": 1500,
    "ip_config": {
      "ip_address": "192.168.6.210",
      "subnet_mask": "255.255.255.0",
      "gateway_address": "0.0.0.0"
    },
    "ieee_802_1q": {
      "vlan_id": 6,
      "priority": 7,
      "inherit_dscp": false
    }
  }
],
"dscp_rules": [

]
```


8.3.5 Activation and deactivation of the network configuration

After the download of the network configuration file in the Online Network Editor of PAS4000, the file can be found under the file name "net.json" in the subdirectory /cfg of the SD card of the PSS 4000-exclusive device.

Transfer and activation of the network configuration

After the download, a restart or reboot of the PSS 4000-exclusive device is required in order to transfer and activate the network configuration. After a restart or reboot, the configuration of DOCUMENT is validated in the network configuration file and the network configuration data (HARDWARE, ETH_INTERFACE and DSCP_RULE) is transferred and activated.



NOTICE

Network configuration data is not transferred

The device starts without the extended network configuration if one of the following cases is established after the restart or reboot of the PSS 4000-exclusive device:

- During the boot process of the PSS 4000-exclusive device, the network configuration file is not in the subdirectory /cfg of the SD card. The device starts with the configuration from device naming.
- After a restart or reboot, it is established that the network configuration file is faulty; i.e. the network configuration does not conform to the rules of the JSON schema and/or ETH parameters are violating a rule:

ETH parameter *eth_if_uid*:

The value for the unique identifier of an ETH interface has been assigned multiple times; i.e. on a PSS 4000-exclusive device, an identifier may only occur once.

ETH parameters *ip_address* and *vlan_id*:

Any combination of *ip_address* and *vlan_id* is present multiple times.

ETH parameter *vlan_id* with independent mode:

The parameter value for *vlan_id* is identical for several physical ports; i.e. on a PSS 4000-exclusive device, the parameter value of *vlan_id* may only occur once.

Please note that, during device naming or when using the PSS 4000 system control block `pssCreateIPInterface`, only the IP parameters of the IP interface are configured. If there is no extended network configuration on a PSS 4000-exclusive device, default settings are used for various parameters (see [Factory default settings for network parameters](#) [ 790]).

Deactivation of the network configuration

For deactivation, the network configuration file must first be deleted (Online Network Editor of PAS4000). Once deleted, a restart or reboot of the PSS 4000-exclusive device is required in order to ensure that the ETH interfaces that were configured in the network configuration file are no longer available on the device.

The PSS 4000-exclusive device starts without the data from the extended network configuration, with the configuration from device naming and various default settings (see [Factory default settings for network parameters](#) [ 790]).

**INFORMATION****Deletion of the network configuration file via an original reset**

If an original reset is carried out for the PSS 4000-exclusive device, then the network configuration file on the SD card is deleted. The device starts without the data from the extended network configuration, with the configuration from device naming and the default settings for Ethernet mode from the ports, MTU and DSCP.

8.3.6 Requirements of receive data

The configuration of the ETH parameter for the ETH interface of a PSS 4000-exclusive device applies primarily for outgoing data packets (send data). However, to ensure that data can also be received via an interface, an incoming data packet (receive data) must meet various requirements. The requirements will vary, depending on whether or not the IPv4 target address of the data packet is a broadcast address.

IPv4 address is not a broadcast address.

Incoming data packets that do not use a broadcast address as IPv4 target address are rejected if they do not meet **ALL** the following requirements:

- ▶ The incoming data packet is only accepted if it comes from the appropriate physical port (see [HARDWARE](#)  143]).
- ▶ In the incoming data packet, the IPv4 address that is configured for this ETH interface must be entered as the IPv4 target address (see *"ip_config" -> "ip_address"* under [ETH_INTERFACE](#)  144]); i.e. the data packet's IPv4 target address and the IPv4 address configured for the ETH interface must match
- ▶ 802.1Q tag is used
When using the 802.1Q tag, the VLAN identifier of the 802.1Q tag in the data packet header must match the VLAN identifier configured for the ETH interface (see *"ieee_802_1q" -> "vlan_id"* under [ETH_INTERFACE](#)  144])
- ▶ 802.1Q tag is not used
If the 802.1Q tag is not used in the data packet header, the VLAN identifier may not be configured for the ETH interface (see *"ieee_802_1q" -> "vlan_id"* under [ETH_INTERFACE](#)  144]).

IPv4 address is a broadcast address

Incoming data packets that use a broadcast address as IPv4 target address are rejected if they do not meet **ALL** the following requirements:

- ▶ The IPv4 target address of the incoming data packet must be a valid broadcast address. The broadcast address must address the subnetwork that was configured for the ETH interface (see *"ip_config" -> "subnet_mask"* under [ETH_INTERFACE](#)  144]).
- ▶ 802.1Q tag is used
When using the 802.1Q tag, the VLAN identifier of the 802.1Q tag in the data packet header must match the VLAN identifier configured for the ETH interface (see *"ieee_802_1q" -> "vlan_id"* under [ETH_INTERFACE](#)  144])
- ▶ 802.1Q tag is not used
If the 802.1Q tag is not used in the data packet header, the VLAN identifier may not be configured for the ETH interface (see *"ieee_802_1q" -> "vlan_id"* under [ETH_INTERFACE](#)  144]).

9 SafetyNET p

9.1 Overview

SafetyNET p is an Ethernet-based multi-master bus system suitable for industrial use. It is used as standard on PSS 4000 as a fieldbus system, to resolve decentralised tasks.

The SafetyNET p protocol contains a safe data channel, which is certified for data transfer in accordance with SIL 3 of IEC 61508. On SafetyNET p, both safety-related and non-safety-related data is transferred via the same bus cable.

SafetyNET p is available in "Real Time Frame Network" (RTFN) form.

The application layer on SafetyNET p (Layer 7 in ISO/OSI's 7-layer model) is based on CANopen. SafetyNET p combines the CAN Bus protocol with a standardised industrial application layer. This includes standardisation of the communication as well as the technical and functional performance data, which enables distributed automation units to be networked. Defined application objects are used in device profiles.

9.1.1 SafetyNET p protocol versions

The update of IEC 61784-3 involves additions to the SafetyNET p protocol. These additions are already considered in SafetyNET p protocol version 2.

The original protocol version will continue to be supported under SafetyNET p protocol version 1.



INFORMATION

Note:

- The SafetyNET p protocol version is configured in the project properties in the PSS 4000 project.
- It is not possible to mix devices with SafetyNET p protocol version 1 and 2 in one PSS 4000 project.



NOTICE

SafetyNET p protocol version on 2nd generation PSS 4000-exclusive devices

2nd generation PSS 4000-exclusive devices (PSS u2 systems) only use SafetyNET p protocol version 2.

If a PSS 4000 project contains both 1st generation PSS 4000-exclusive devices (PSSu systems, PSS67 devices) and 2nd generation PSS 4000-exclusive devices (PSS u2 systems), only SafetyNET p protocol version 2 can be used.

9.1.2 SafetyNET p in ISO/OSI's 7-layer reference model

The ISO/OSI (Open System Interconnection) reference model is a standardised format for describing all data communication functions for all data processing communication systems. Functionalities are divided to form a layer model, with 7 different functionalities. Field-bus communication is described by layers 1, 2 and 7.

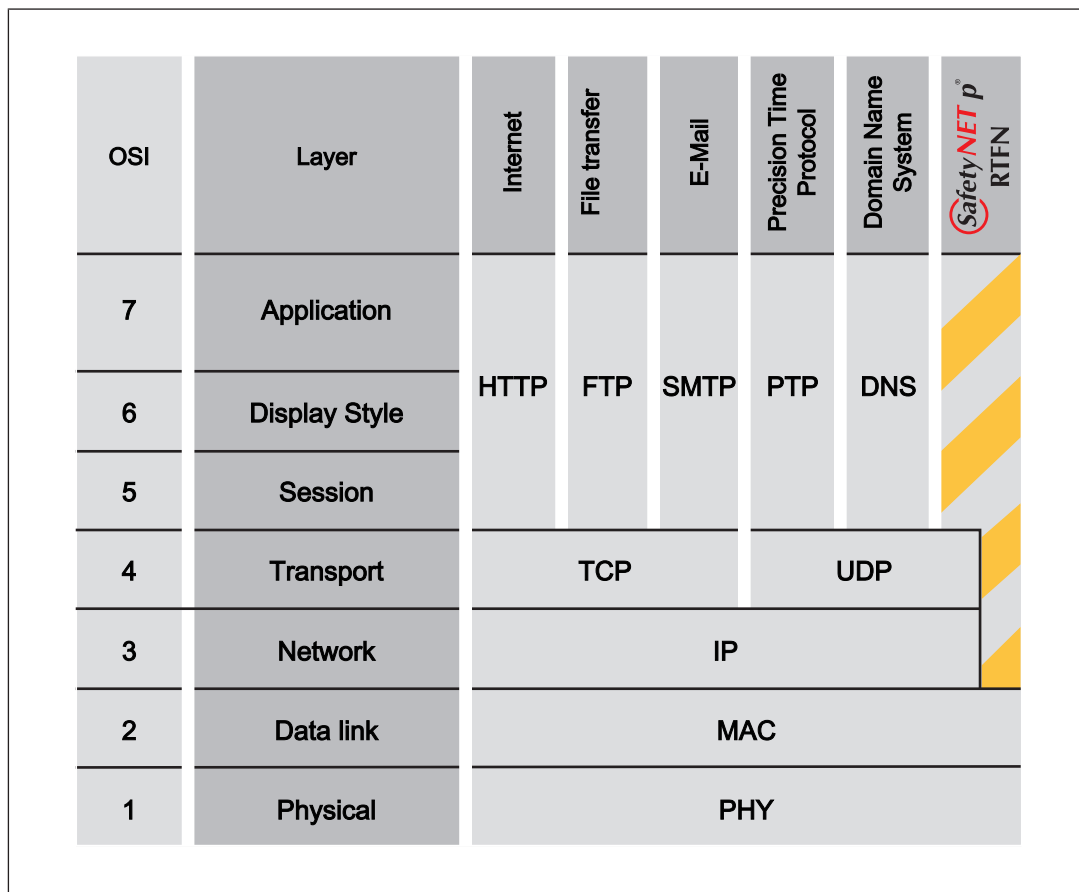


Fig.: SafetyNET p in ISO/OSI's 7-layer reference model

9.1.3 RTFN (Real Time Frame Network)

The RTFN transport layer is used at process control and cell level.

RTFN is fully compatible with the standard Ethernet familiar from office automation, as it is based on the same transport mechanisms. As RTFN is designed to be mixed with other utilities in an Ethernet network, with no specific restrictions, the RTFN subscribers communicate with other devices in the network (laptops, printers etc.) on an equal basis.



INFORMATION

If non-RTFN-enabled devices place a high load on the network, the availability of SafetyNET p can be adversely affected. Where the availability requirements are high, therefore, we recommend that you do not run any other communication in parallel with SafetyNET p, or that you use a switch with management functions.

9.1.4 Transfer of safety-related and non-safety-related data

SafetyNET p can be used to transfer safety-related data through an integrated safe communication layer. The security mechanisms are designed to meet up to SIL3 of IEC 61508. The safety-related data is sent encapsulated within SafetyNET p telegrams. As a result, all other network components such as switches or cable can be designed as standard Ethernet components. They do not affect the transfer of safety-related data.

Non-safety-related SafetyNET p subscribers such as PCs or standard control systems, for example, do not influence safety-related communication either. As a result it is possible to mix the operation of safety and non-safety-related subscribers within a network.

9.2 Topology on SafetyNET p

As SafetyNET p is an Ethernet-based bus system, all valid structures familiar to Ethernet, as line topology or star topology, are possible. You can also implement dynamic structures.

9.2.1 Linear topology

A linear topology is created by connecting several bus subscribers in a line, without branches. This topology is the familiar fieldbus structure. A linear topology is preferable on plants that extend over a wide area (e.g. conveyor systems, interconnecting manufacturing cells).

SafetyNET p subscribers often have two Ethernet interfaces (autosensing switch with two free ports) to enable a linear topology. That way there is no need for any additional network components, such as switches.

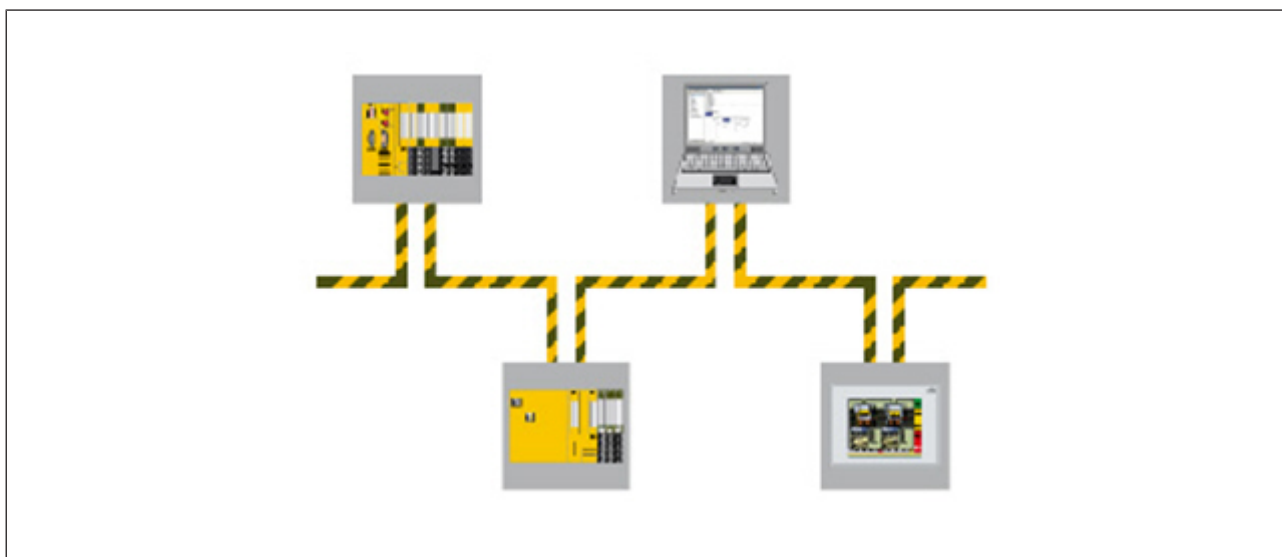


Fig.: Linear topology (principle)

9.2.2 Star topology

A star topology is formed when several subscribers are connected at a central point. A switch is generally used as the central signal distributor. The communication partners communicate via the switch in point-to-point connections. A star topology is preferable in areas with a high density of devices over a small physical area (e.g. small manufacturing cells, standalone production machines).

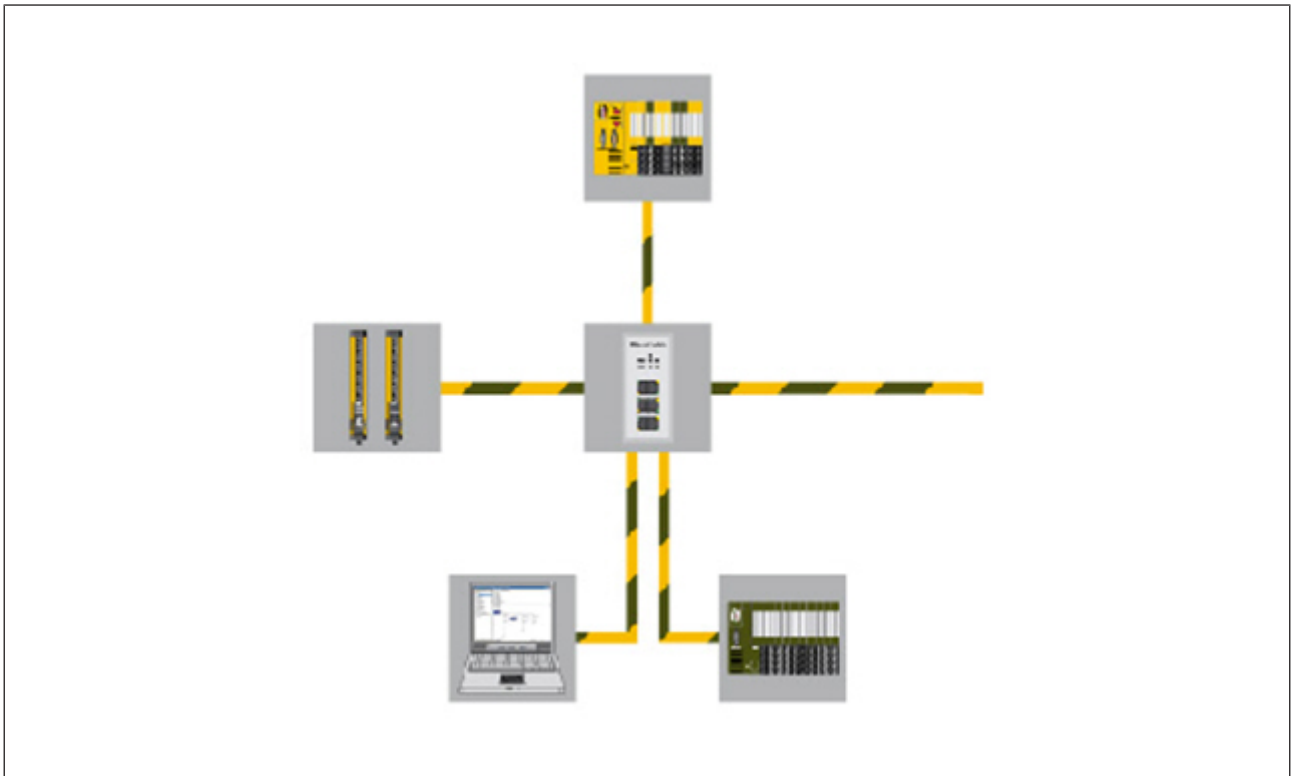


Fig.: Star topology (principle)

9.2.3 Dynamic structures

SafetyNET p supports the familiar Ethernet capacity to dynamically hot swap subscribers. This is particularly useful for service, maintenance and diagnostics.

9.3 Selecting SafetyNET p components

SafetyNET p supports the familiar, robust fieldbus connection technology with established plug-in connectors, as well as economical IP20 cabling in the form of RJ45 connectors in the control cabinet.

Conventional, standard Ethernet infrastructures can be used if the performance is satisfactory. This includes connectors, cables, routers, switches, gateways and communication channels.

We recommend you use components supplied by Pilz.

In principle it is possible to use other components, provided they are suitable for use in an industrial environment.



INFORMATION

If you assemble the connection cable yourself we recommended that you use connector and cable made by the same manufacturer. This is because considerable variances may arise regarding connector dimensions and cable cross sections, due to the permitted tolerances.

9.4 SafetyNET p interface

The physical connection for data exchange via SafetyNET p is provided via [Ethernet interface](#) [ 131].



NOTICE

Only use SafetyNET p subscribers (e.g. switches, hubs) that support the autosensing/autonegotiation function, otherwise a duplex conflict (duplex mismatch) may arise, which can cause considerable delays or sporadic data transmission errors.

9.5 Process data exchange via SafetyNET p

Process data exchange via SafetyNET p can occur with any PSS 4000 device that has the ST SafetyNET p and/or FS SafetyNET p system section.

The FS/ST SafetyNET p system section of a PSS 4000 device sends process data from its transmit data area to its communication partner via transmit connections and receives process data from its communication partner in its receive data area via receive connections.

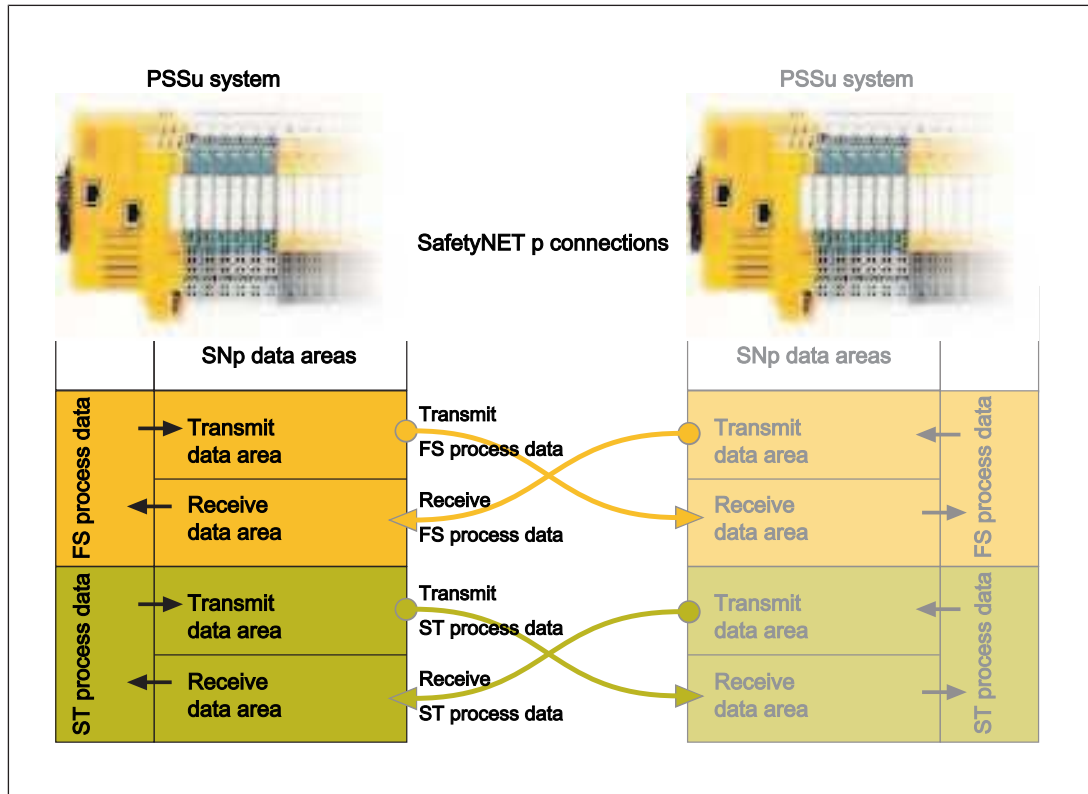


Fig.: Principle of process data exchange via SafetyNET p (e.g. PSSu systems)

Safety-related process data (FS process data) is sent via safe connections (FS connections) on SafetyNET p's safe data channel. Non-safety-related process data (ST process data) is sent via non-safety-related connections (ST connections) on SafetyNET p's non-safety-related data channel.

Each connection consists of a transmit part and a receive part. The transmit part includes an area in the transmit data area of the transmitting device. This area is called the TX data area. The receive part includes an area in the receive data area of the receiving device. This area is called the RX data area.

So each connection has its own TX data area on the transmitting device and its own RX data area on the receiving device.

► Process data exchange of FS process data

FS process data is exchanged via FS connections; i.e. FS process data is sent from the corresponding (FS-)Tx data area of a device via an FS-Tx connection and, from the perspective of the receiving device, is received in the corresponding (FS-)Rx data area via an FS-Rx connection.

► Process data exchange of ST process data

ST process data is exchanged via ST connections; i.e. ST process data is sent from the corresponding (ST-)Tx data area of a device via an ST-Tx connection and, from the perspective of the receiving device, is received in the corresponding (ST-)Rx data area via an ST-Rx connection.

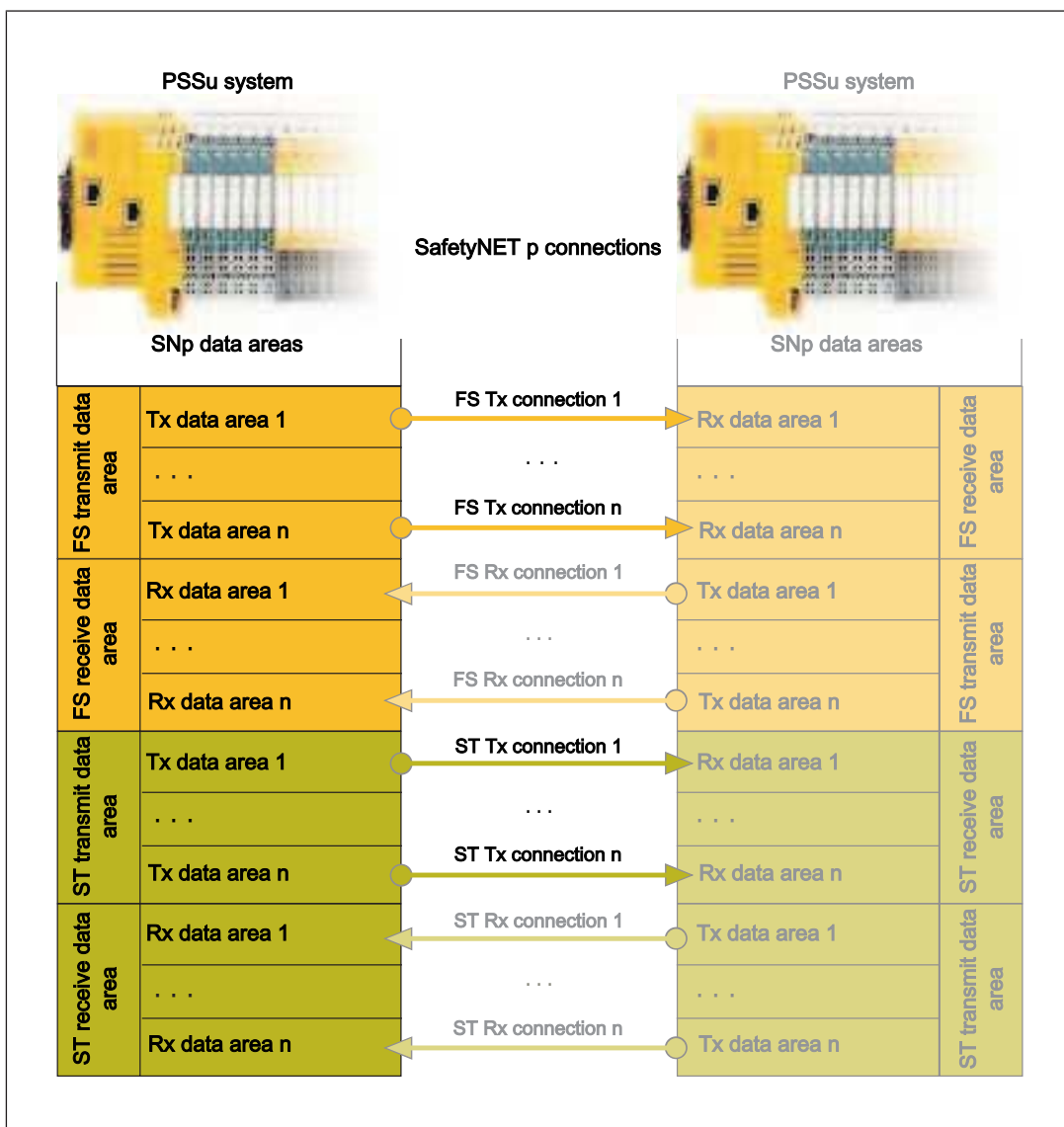


Fig.: Principle of process data exchange via connections (e.g. PSSu systems)

9.5.1 Cyclical processing of connections

The FS/ST SafetyNET p system section on the transmitting PSS 4000 device starts a Tx connection cyclically with the cycle time t_{SNp} and sends the process data via SafetyNET p (see [Reaction times](#)  570).

In the FS/ST SafetyNET p system section of the receiving device, event-driven tests check whether new process data is available for further processing in the Rx data area belonging to the Tx connection.

After the Tx connection is started, the process data must be available in the corresponding receive data area within the configured maximum transmission delay time t_{DelaySNp} . If the measured time (see [FS heartbeat \(FS-HBT\)](#)  190) exceeds the configured maximum transmission delay time t_{DelaySNp} , all FS receive connections (FS-Rx connections) received from the transmitting device are deactivated on the receiving device. The substitute values are used for the receive data, with valid bits = FALSE.

The system monitors this for all Tx connections within a project (see [Reaction times](#)  570).

9.5.2 SafetyNET p receive timeout

A PSS 4000 device uses the SafetyNET p receive timeout to monitor the data received for communication errors. This applies to both ST communication errors (system section ST SafetyNET p) and FS communication errors (system section FS SafetyNET p). To aid understanding, the information below considers SNp communication from the perspective of the receiving device, as monitoring occurs exclusively on the receive side:

- ▶ From the perspective of the device receiving the process data, the transmitting device can also be called the remote device.
- ▶ A connection through which a device receives process data from the remote device can also be called the receive connection (Rx connection).
 - The process data from an ST-Rx connection is in the corresponding ST-Rx data area.
 - The process data from an FS-Rx connection is located in the corresponding FS-Rx data area.
- ▶ An Rx connection is enabled when the relevant system section of a PSS 4000 device accepts the process data available for further processing in the corresponding Rx data area.
- ▶ An Rx connection is disabled when the relevant system section of a PSS 4000 device fails to accept or ignores the process data available for further processing in the corresponding Rx data area.
- ▶ In the Rx data area belonging to the Rx connection, new process data must be present before the SNp receive timeout expires, otherwise the relevant system section of the PSS 4000 device detects a lost connection and disables the Rx connection.
- ▶ The ST SafetyNET p system section of a PSS 4000 device can maintain ST-Rx connections to a single remote device as well as ST-Rx connections to several remote devices.
- ▶ The FS SafetyNET p system section of a PSS 4000 device can maintain FS-Rx connections to a single remote device as well as FS-Rx connections to several remote devices.

If a system section (ST SafetyNET p / FS SafetyNET p) detects a communication error, it disables the Rx connection(s), depending on the type of communication error. In the event of errors in FS communication, disabling also depends on the SafetyNET p protocol version.

- ▶ SafetyNET p protocol version of 1st generation PSS 4000 devices:
With 1st generation devices, either SNp protocol 1 or 2 can be used, depending on the device.
- ▶ SafetyNET p protocol version of 2nd generation PSS 4000 devices:
2nd generation devices always use SNp protocol version 2.

ST communication (= enabling of ST-Rx connections) is always resumed automatically after a communication error, provided the cause of the error is no longer present (see ST SafetyNET p RTFN in RUN condition with minor error). The resumption of FS communication (= enabling of FS-Rx connections) after a communication error depends on the SafetyNET p protocol version and the type of communication error.

- ▶ [SNp protocol version 1: Communication errors and resumption of FS communication](#)  180
- ▶ [SNp protocol version 2: Communication errors and resumption of FS communication](#)  181

9.5.2.1 SNp protocol version 1: Communication errors and resumption of FS communication

A distinction is made between the following communication errors on SNp protocol version 1:

▶ **Permanent error in FS communication**

An error is "permanent" if the receiving PSS 4000 device detects at least 2 connection errors within 30 s. FS communication with the remote device is disabled permanently.

▶ **One-off error in FS communication**

An error is "one-off" if the receiving PSS 4000 device detects a maximum of 1 connection error within 30 s. FS communication with the remote device is disabled temporarily.

The FS SafetyNET p system section of a PSS 4000 device detects a communication error when it fails to receive any new process data from the remote device within the SNp receive timeout. This is what happens:

- ▶ An FS SafetyNET p wait time of 1 s is started.
- ▶ The FS SafetyNET p system section of the PSS 4000 device switches to FS SafetyNET p in RUN state with minor error.
- ▶ At the very earliest the PSS 4000 device can re-enable FS communication with the remote device once the FS SNp wait time of 1 s has elapsed, provided the error has been recovered and/or the error is no longer present.
 - Permanent error
The user must execute one of the resets in order to resume FS communication.
 - One-off error
FS communication is resumed automatically.

9.5.2.2 SNp protocol version 2: Communication errors and resumption of FS communication

A distinction is made between the following communication errors on SNp protocol version 2:

- ▶ **Connection-based error in FS communication**
The error concerns FS communication via a single FS-Rx connection. The FS-Rx connection is temporarily disabled.
- ▶ **Device-based error in FS communication**
The error concerns FS communication between two PSS 4000 devices. All FS-Rx connections between the receiving device and the remote device are temporarily disabled.
- ▶ **Unspecific error in FS communication**
The error concerns FS communication between a PSS 4000 device and all its remote devices. All the PSS 4000 device's FS-Rx connections to all of its remote devices are temporarily disabled. An error is "unspecific" when the cause of the error cannot be clearly assigned to any of the remote devices.

The FS SafetyNET p system section of a PSS 4000 device detects a communication error when no new process data is available via an FS-Rx connection within the SafetyNET p receive timeout. Several FS-Rx connections may also be connected, depending on the type of communication error. This is what happens:

- ▶ An [FS SafetyNET p wait time](#)  182] is started.
- ▶ The FS SafetyNET p system section of the PSS 4000 device switches to FS SafetyNET p in RUN state with minor error.
- ▶ At the very earliest, the FS SafetyNET p system section of the PSS 4000 device can re-enable FS communication once the FS SafetyNET p wait time has elapsed, provided the error has been recovered and/or the error is no longer present.
FS communication is resumed (enabled) automatically after all communication errors.

9.5.2.3 SNp protocol version 2: Configurable FS SafetyNET p wait time

The FS SafetyNET p wait time for a PSS 4000 project can be configured, provided SNp protocol version 2 is configured for the PSS 4000 project.

The FS SafetyNET p wait time is the time a PSS 4000 project waits in order to detect all errors in FS communication in all FS SafetyNET p system sections; i.e. the FS SafetyNET p wait time guarantees that all communication errors on all FS-Rx connections in a PSS 4000 project are detected. During the SafetyNET p wait time, process data available for further processing in the affected FS-Rx data areas is ignored.

Calculate FS SafetyNET p wait time automatically

In the properties for the PSS 4000 project (PAS4000) it is possible to configure that the FS SafetyNET p wait time is calculated automatically (= default setting). With automatic calculation, the system determines the minimum FS SafetyNET p wait time for the project (= lowest permitted value). The minimum FS SafetyNET p wait time is displayed when building the project.

Configure FS SafetyNET p wait time via the user

In some circumstances it makes sense to configure the project with an FS SafetyNET p wait time that is greater than the minimum wait time. This is the case, for example, when an FS-Rx connection is disabled and re-enabled frequently. In the Project Manager for the PSS 4000 project (PAS4000), the FS SafetyNET p wait time can be configured in the following value range.

► Minimum FS SafetyNET p wait time

In the configuration the value must not drop below the lowest permitted value for the FS SafetyNET p wait time. The minimum FS SafetyNET p wait time $t_{\text{WaitTime\_min}}$ is calculated as follows:

$$t_{\text{WaitTime\_min}} = 2 [\max (t_{\text{Task}}; t_{\text{MBUS}}; t_{\text{ExtCo}}; t_{\text{SNp}})] + 1 t_{\text{BasicSNp}}$$

$t_{\text{WaitTime\_min}}$

Minimum FS SafetyNET p wait time for the PSS 4000 project

$\max (t_{\text{Task}}; t_{\text{MBUS}}; t_{\text{ExtCo}}; t_{\text{SNp}})$:

The largest of the values should be used for the calculation

t_{Task} :

Task cycle time of the FS task with the largest t_{Task} in the PSS 4000 project

t_{MBUS} :

Module bus cycle time of the FS module bus PSSu or FS module bus PSS u2 with the largest value for t_{MBUS} in the PSS 4000 project

t_{ExtCo} :

Only with PROFIsafe communication:
longest cycle time t_{ExtCo} in the PSS 4000 project for cyclical transmission of external O-data from the O-data area during PROFIsafe communication

t_{SNp} :

Longest cycle time for the cyclical start of an FS SNp-Tx connection (send FS data) in the PSS 4000 project during cross-project SafetyNET p communication

t_{BasicSNp} :

Basic cycle time for cyclical test for existing SNp send data in a project

► Maximum FS SafetyNET p wait time

When configuring the FS SNp wait time, the longest permitted value for the FS SafetyNET p wait time must not be exceeded.

Maximum FS SNp wait time: 60000 ms

9.5.3 Data transfer limits

The data transfer limits are determined by the following data:

► **Maximum number of ST connections per ST SafetyNET p system section**

The maximum number of ST-Tx connections is limited by the number of ST transmit data areas that an ST SafetyNET p system section can manage. The number of ST transmit data areas always corresponds to the number of ST receive data areas.

The number of transmit and receive data areas that an ST SafetyNET p system section can manage depends on the PSS 4000 device that is used (e.g. head module for control systems PSSu PLC, control systems PSSu multi, decentralised system PSSu I/O).

Please refer to the information stated in the operating manual of the device you are using.

The ST-Tx connections can either be established to a single PSS 4000 device or to the SafetyNET p system sections of various PSS 4000 devices.

Example: Control system PSSu PLC:

The ST SafetyNET p system section of a control system PSSu PLC can manage the transmit data areas for a max. 64 ST-Tx connections; consequently it can also manage the receive data areas for a max. 64 ST-Tx connections.

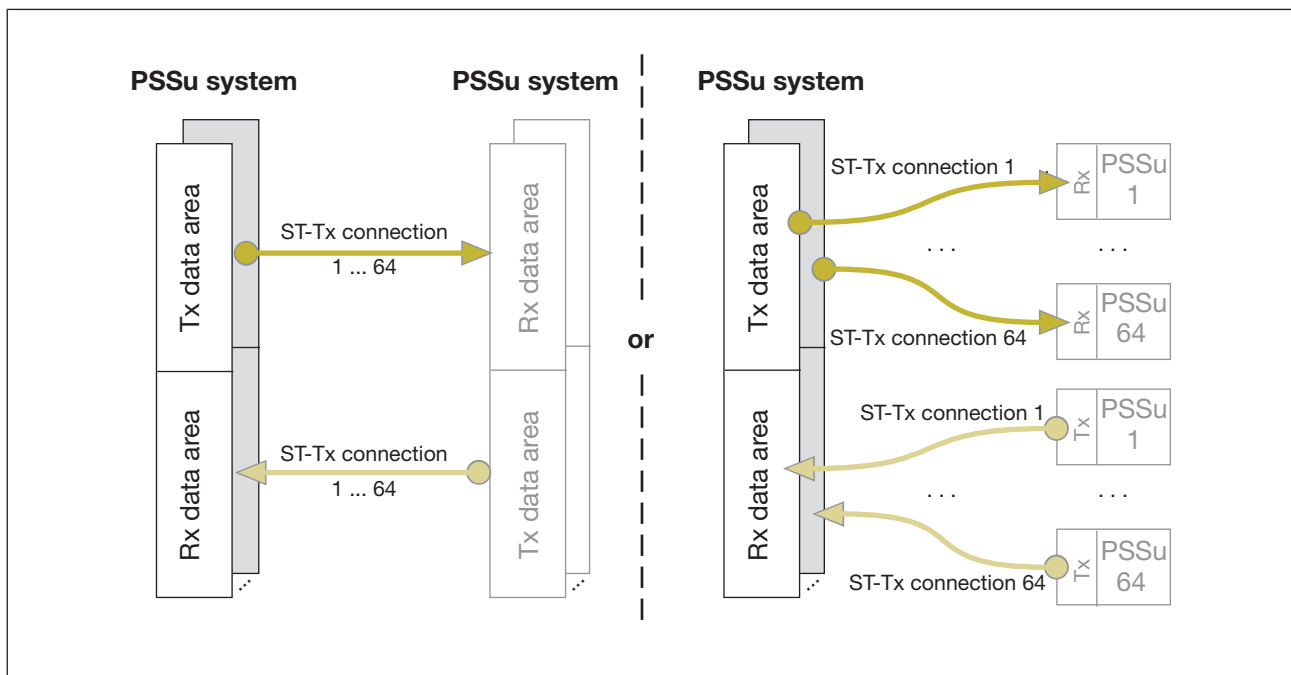


Fig.: Max. number of ST-Tx connections on control systems PSSu PLC (PSSu system as example)

► **Maximum number of FS connections per FS SafetyNET p system section**

The maximum number of FS-Tx connections is limited by the number of FS transmit data areas that an FS SafetyNET p system section can manage. The number of FS send data areas always corresponds to the number of FS receive data areas.

The number of transmit and receive data areas that an FS SafetyNET p system section can manage depends on the PSS 4000 device that is used (e.g. head module for control systems PSSu PLC, control systems PSSu multi, decentralised system PSSu I/O).

Please refer to the information stated in the operating manual of the device you are using.

The FS-Tx connections can either be established to a single PSS 4000 device or to the FS SafetyNET p system sections of various PSS 4000 devices.

Example: Control system PSSu PLC:

The FS SafetyNET p system section of a control system PSSu PLC can manage the transmit data areas for a max. 64 FS-Tx connections; consequently it can also manage the receive data areas for a max. 64 FS-Tx connections.

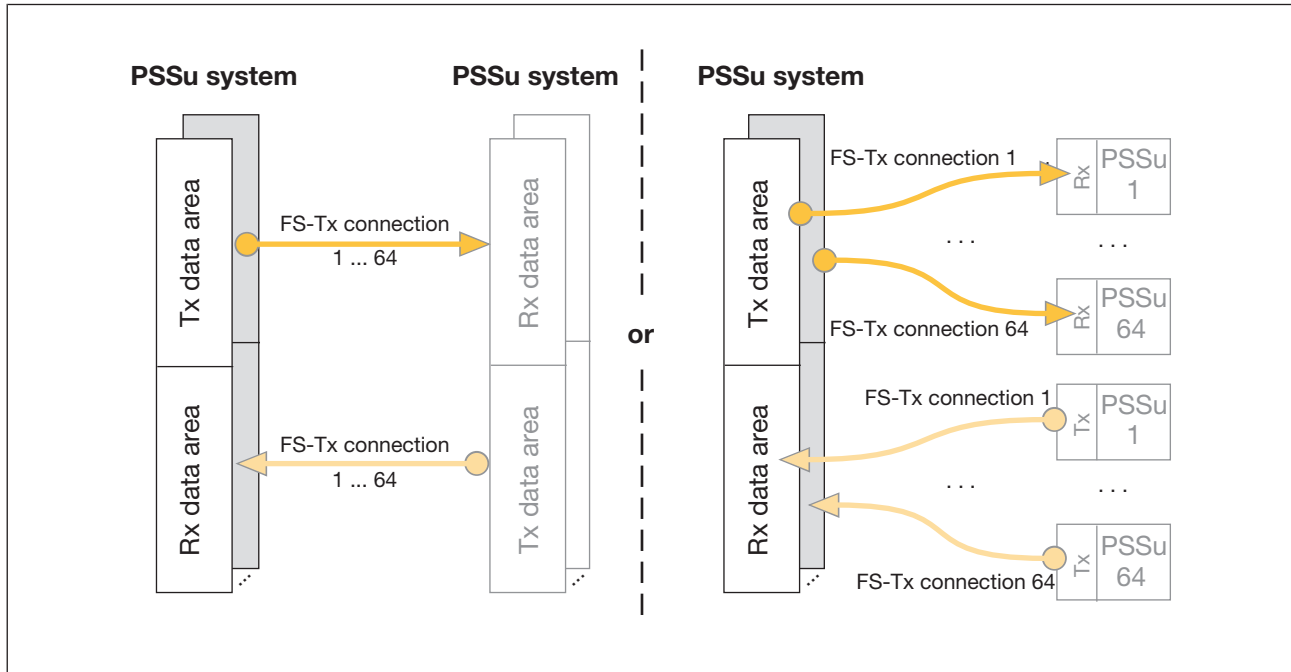


Fig.: Max. number of FS-Tx connections on control systems PSSu PLC (PSSu system as example)

► **Max. number of bytes per connection**

The number of bytes per connection depends on the PSS 4000 device you are using.

- Send ST process data from the Tx data area:
246 Bytes per ST-Tx connection
- Received ST process data in the Rx data area:
246 Bytes per ST-Tx connection
- FS process data sends from the Tx data area:
115 bytes per FS-Tx connection
- FS process data receives in the Rx data area:
115 bytes per FS-Tx connection

► **Max. number of variables per connection**

The number of variables that can be transmitted in a connection depends on their bit requirement. A variable's bit requirement depends on its data type and any possible expansion (valid bit).

If the variables to be transmitted exceed the max. number of bytes that can be transmitted with a connection, the variables are automatically split over additional connections.

Please note that this may affect data consistency (see [Data consistency](#) [377]).

– **Variables with elementary data type**

The bit requirement depends on the variables' data type (e.g. BOOL/SAFEBOOL: 1 Bit, WORD/SAFEWORD: 16 Bits).

– **Variables with directly derived data type**

The bit requirement corresponds to the bit requirement of the elementary data type from which the variable was derived.

– **Variables of ARRAY data type**

The bit requirement corresponds to the sum of the bits needed for each element. For example, the bit requirement of a one-dimensional array can be calculated by multiplying an element's bit requirement by the number of elements.

In terms of data consistency, the system ensures that the elements of an array are not split over different connections.

– **Variables of STRUCT data type**

The bit requirement corresponds to the sum of the bits needed for each element.

In terms of data consistency, the system ensures that the elements of a structure are not split over several connections.

– **Variables with VALID extension**

The variables' bit requirement is increased by 1 Bit. The valid bit is transmitted once only. This also applies for a variable of an I/O data type.



INFORMATION

The limits described here apply without restriction for **project-internal Snp communication**.

In contrast to some of the limits described here, the information stated under [I/O mapping rules between PSS 4000 projects](#) [ 373] applies for **cross-project Snp communication**.

9.6 Project-internal SafetyNET p communication

Project-internal SNp communication is data exchange between PSS 4000 devices that are all within the same project, via FS/ST SafetyNET p.

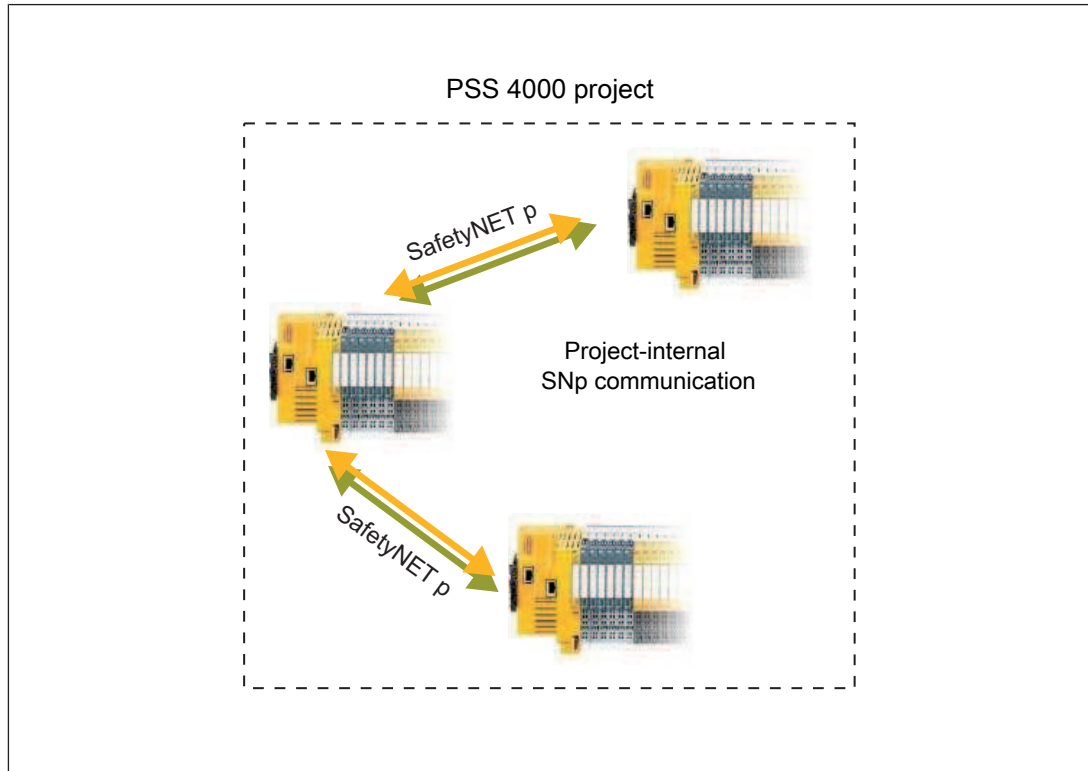


Fig.: Project-internal SafetyNET p communication (principle)

Further information on project-internal SNp communication can be found under [I/O mapping rules within a PSS 4000 project](#) [ 370].

9.7 Cross-project SafetyNET p communication

Cross-project Snp communication is data exchange via FS/ST SafetyNET p between PSS 4000-exclusive devices located in different PSS 4000 projects.

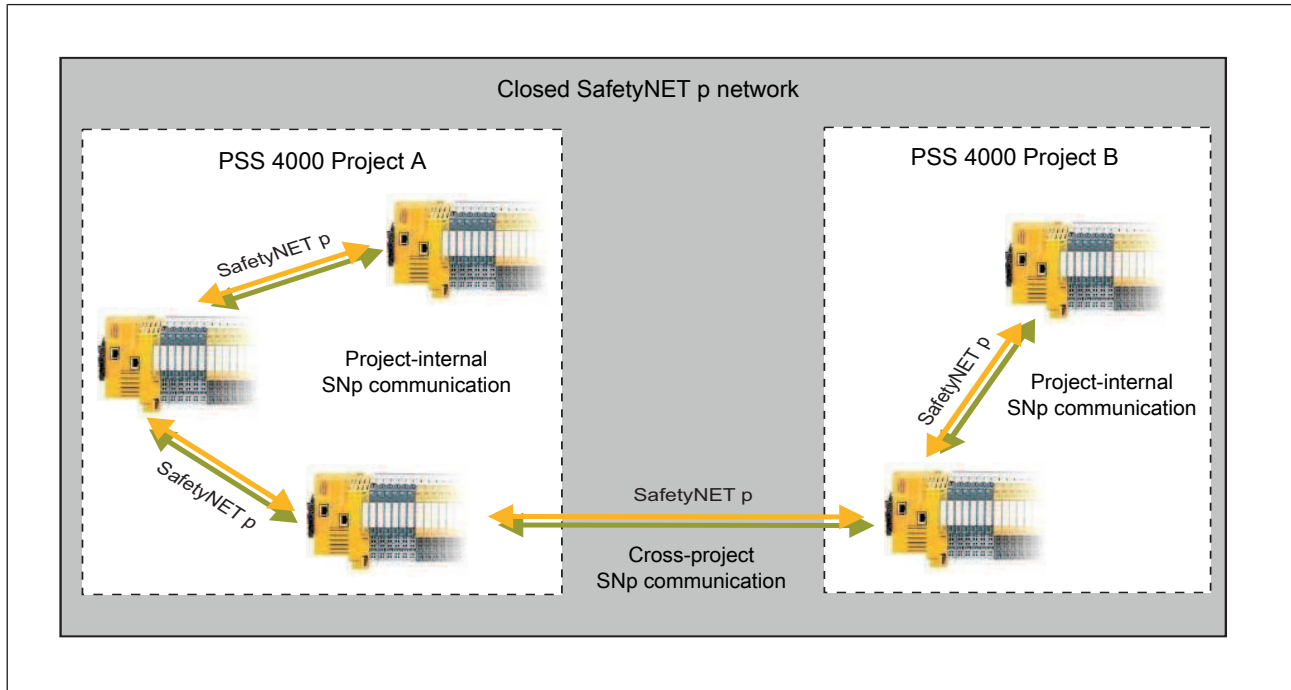


Fig.: Cross-project SafetyNET p communication (principle)

With cross-project Snp communication, each PSS 4000 project can read the O-PI variables of the other PSS 4000 projects and write to the I-PI variables of the other PSS 4000 projects. Each individual PSS 4000 project is created first in PAS4000 for this purpose.

In PAS4000, the PASconnect interface must be activated in all PSS 4000 projects that are to exchange data via SafetyNET p. As long as a project only contains 1st generation PSS 4000-exclusive devices (PSSu systems, P67 devices), PASconnect interface version 1.2.1.x can be activated. As soon as a project contains at least one 2nd generation PSS 4000-exclusive device (PSS u2 system), PASconnect interface version 1.3.x/1.4.x will need to be activated.

The I/O mapping between the PI variables of the PSS 4000 projects must then be carried out in the software tool PASconnect. Each individual PSS 4000 project must then be built and downloaded to the devices.

Further information on cross-project Snp communication can be found under [I/O mapping rules between PSS 4000 projects](#) [373].

With cross-project Snp communication there may only be one PASconnect project within a closed SafetyNET p network. All SafetyNET p-enabled devices within the same closed SafetyNET p network as the PASconnect project must belong to a subproject of the PASconnect project. This also applies if the device belongs to a control project that does not exchange data with any other control projects via SafetyNET p.

A SafetyNET p network is deemed to be "closed" when it is guaranteed that SafetyNET p data traffic does not extend beyond the network boundary.

Example: There are three control projects within a closed SafetyNET p network. Control project 1 exchanges data with control project 2 via SafetyNET p. Control project 3 contains SafetyNET p-enabled devices, but it does not exchange any data with the other control projects via SafetyNET p. In this case, all three control projects must be configured as subprojects of the same PASconnect project.

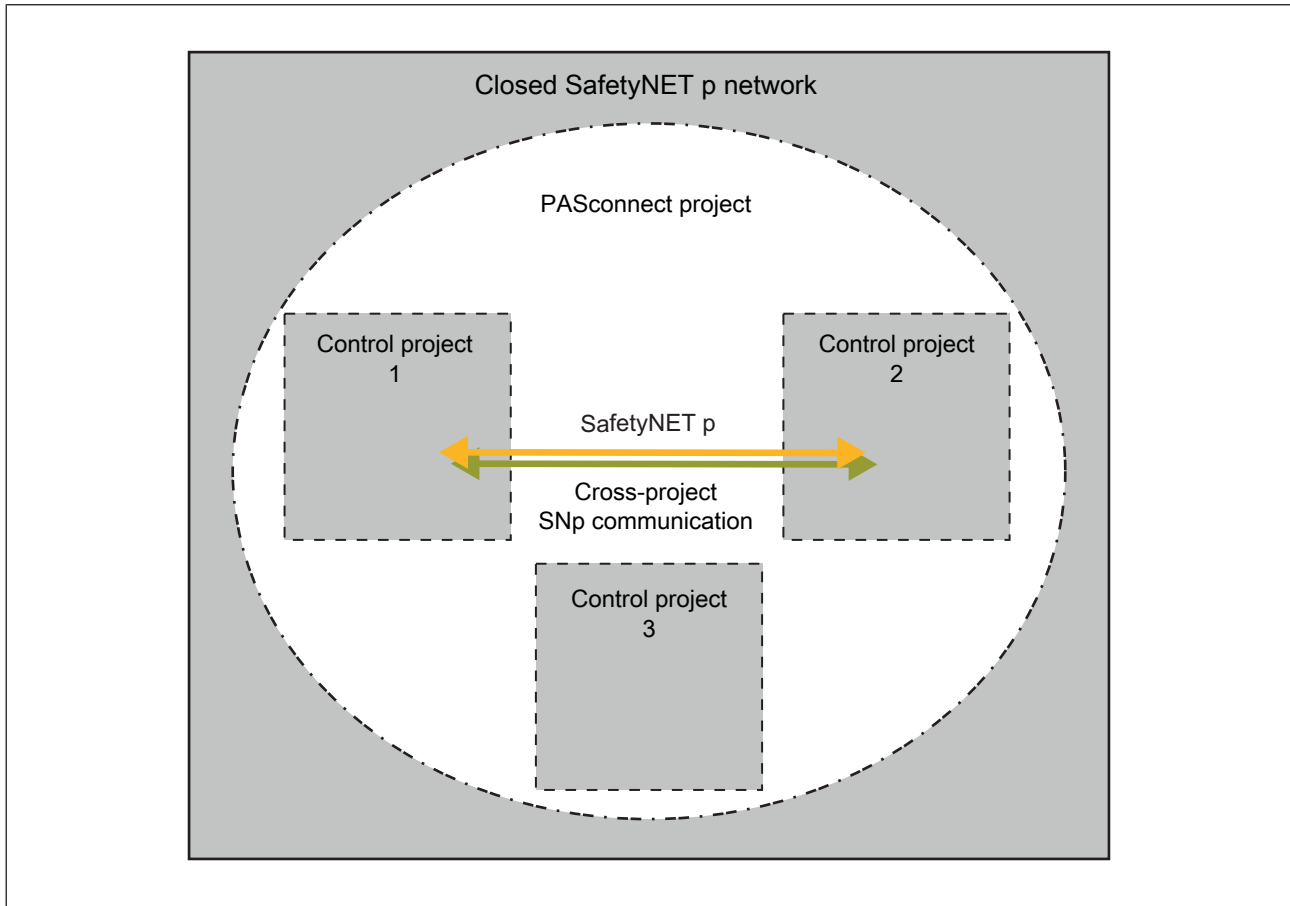


Fig.: Network requirements with cross-project Snp communication

9.8 FS heartbeat (FS-HBT)

The FS heartbeat (FS-HBT) is a control signal that is used to continuously check the accessibility of the communication partners (remote devices) during FS SNp communication. A PSS 4000 device sends an FS heartbeat to the respective remote device as part of each cycle and expects an FS-HBT response from the remote device within a certain time.



INFORMATION

For most projects it's not necessary to configure the FS heartbeat. The default configuration is sufficient. So it's not strictly necessary for you to deal with the issue of the "FS heartbeat". You should only modify the default configuration for the FS heartbeat if you have the relevant expert knowledge.

Please note the following relationships:

- ▶ The cycle time of the FS heartbeat determines the fault detection time; i.e. if the maximum transmission delay time t_{DelaySNp} is exceeded due to an error, then this will be recognised and detected based on the cycle time of the FS heartbeat. Consequently, a long cycle time for the FS heartbeat will lead to a long fault detection time.
- ▶ The FS heartbeat can generate an unwanted network load if a low value is configured for the cycle time of the FS heartbeat and/or a device is connected to lots of other devices via FS SNp.
- ▶ The cycle time of the FS heartbeat is **ALWAYS** an integer multiple of the basic cycle time t_{BasicSNp} .
 - Depending on the configuration, the value for the cycle time of the FS heartbeat changes when the basic cycle time t_{BasicSNp} changes. This repercussion is not normally desirable. On projects with cross-project FS SNp communication, the connected projects in PASconnect will also need to be rebuilt in this case.
 - The basic cycle time t_{BasicSNp} can also change when an ST task is added or modified.

Cycle time of the FS heartbeat

The cycle time with which a PSS 4000 device sends FS heartbeats via FS SNp connections is called the FS heartbeat cycle time ($t_{\text{SNp FS-HBT}}$). The FS heartbeat cycle time is identical for all PSS 4000 devices in a project.

▶ Automatic configuration of the FS heartbeat cycle time

- Projects without project linking (project-internal FS SNp communication)

When automatic configuration (default configuration) is used for the FS heartbeat cycle time on projects with project-internal FS SNp communication, PAS4000 uses the default value for the FS heartbeat cycle time. In this case, the user has nothing further to do.

For project-internal FS SNp communication, automatic configuration is recommended.

- Projects with cross-project FS SNp communication and PASconnect interface version 1.2.x

Note: PASconnect interface version 1.2.x is only available when a project only contains 1st generation PSS 4000-exclusive devices (PSSu systems, P67 devices).

With these projects, the FS heartbeat cycle time is always calculated automatically.

Note: PAS4000 always calculates the FS heartbeat cycle time as an integer multiple of the basic cycle time t_{BasicSNp} . The multiple is an internally defined value; i.e. a change to the basic cycle time inevitably leads to a change to the FS heartbeat cycle time.

► Manual configuration of the FS heartbeat cycle time

Value range: 6 ms ... 30000 ms

- Projects without project linking (project-internal FS SNp communication)

On projects with project-internal FS SNp communication, the FS heartbeat cycle time can also be configured manually. Although we recommend automatic configuration, in certain cases it may make sense to define the FS heartbeat cycle time manually. This is the case, for example, when the network load generated by the FS heartbeat is to be reduced.

Note: The manually configured value for the FS heartbeat cycle time is used internally to calculate the basic cycle time t_{BasicSNp} . This may mean that the basic cycle time t_{BasicSNp} changes.

- Projects with cross-project FS SNp communication and PASconnect interface version 1.3.x/1.4.x

Note: As soon as a project contains at least one 2nd generation PSS 4000-exclusive device (PSS u2 system), only PASconnect interface version 1.3.x/1.4.x is available.

With PASconnect interface version 1.3.x/1.4.x, the FS heartbeat cycle time can only be configured manually. With this procedure it is possible to make changes to a project with project linking as far as possible without repercussions; i.e. changes to a project should not have any repercussions on the projects linked to this project. In particular, changes should not modify the device-internal reaction and fault detection times of the linked projects.

Note: The manually configured value for the FS heartbeat cycle time is used internally to calculate the basic cycle time t_{BasicSNp} . This may mean that the basic cycle time t_{BasicSNp} changes.



NOTICE

For **manual configuration** of the FS heartbeat cycle time ($t_{\text{SNp FS-HBT}}$), please note:

- If the value configured for the FS heartbeat cycle time is greater than the default or recommended value, then the fault detection time is extended; i.e. if the maximum transmission delay time t_{DelaySNp} is exceeded due to a fault, this will not be recognised and detected until later.
- The manually configured FS heartbeat cycle time influences the internal calculation of the basic cycle time t_{BasicSNp} . If the manually configured FS heartbeat cycle time leads to a basic cycle time whose value is outside the permitted value range, then PAS4000 will suggest an appropriate value for the FS heartbeat cycle time during the build process.

9.9 SafetyNET p scheduling mode

SafetyNET p scheduling mode can be used to influence the cycle times of connections during project-internal communication. SafetyNET p scheduling mode is configured in PAS4000.

Scheduling mode is configured per device. The configuration of scheduling mode on a device impacts on all the device's project-internal Rx connections. The configuration applies to the device's project-internal FS connections and project-internal ST connections. If scheduling mode on a device is changed, the SafetyNET p configuration of the device itself is changed as a result, as well as the SafetyNET p configurations of all devices that send data to the device (project-internal Tx connections).

Changing scheduling mode on a device influences the check sum "FS device project" if at least one project-internal FS connection is affected by the change. It changes the check sum "FS device project" on the device on which scheduling mode was changed and on devices that send FS data to the device.



INFORMATION

SafetyNET p scheduling mode has no influence on the minimum FS SafetyNET p waiting time.

Scheduling mode "Reduce SafetyNET p response time"

"Reduce SafetyNET p response time" is the default configuration.

This scheduling mode corresponds to automatic configuration up to PSS 4000 Firmware Version 1.22.x.

To reduce the response time, data in this scheduling mode is sent more frequently than in scheduling mode "Reduce SafetyNET p receive load".

However, if there is a high data load, under some circumstances the data packets may not be processed quickly enough on the receive side. In this case, the affected system section on the receive side detects a communication error.

The scheduling mode "Reduce SafetyNET p response time" should only be deselected if the receiving device is overloaded.

Scheduling mode "Reduce SafetyNET p receive load"

To manage the data load on the receive side, data in this scheduling mode is sent less frequently than in scheduling mode "Reduce SafetyNET p response time". However, the response time is increased in this scheduling mode.

Note: In order to use this scheduling mode, there must be a maximum of 1 task each on a device's ST resource and FS resource.

9.10 Network without Routers

There are a number of rules to be observed when designing a SafetyNET p network for a project. If you are not using Routers, the following applies:

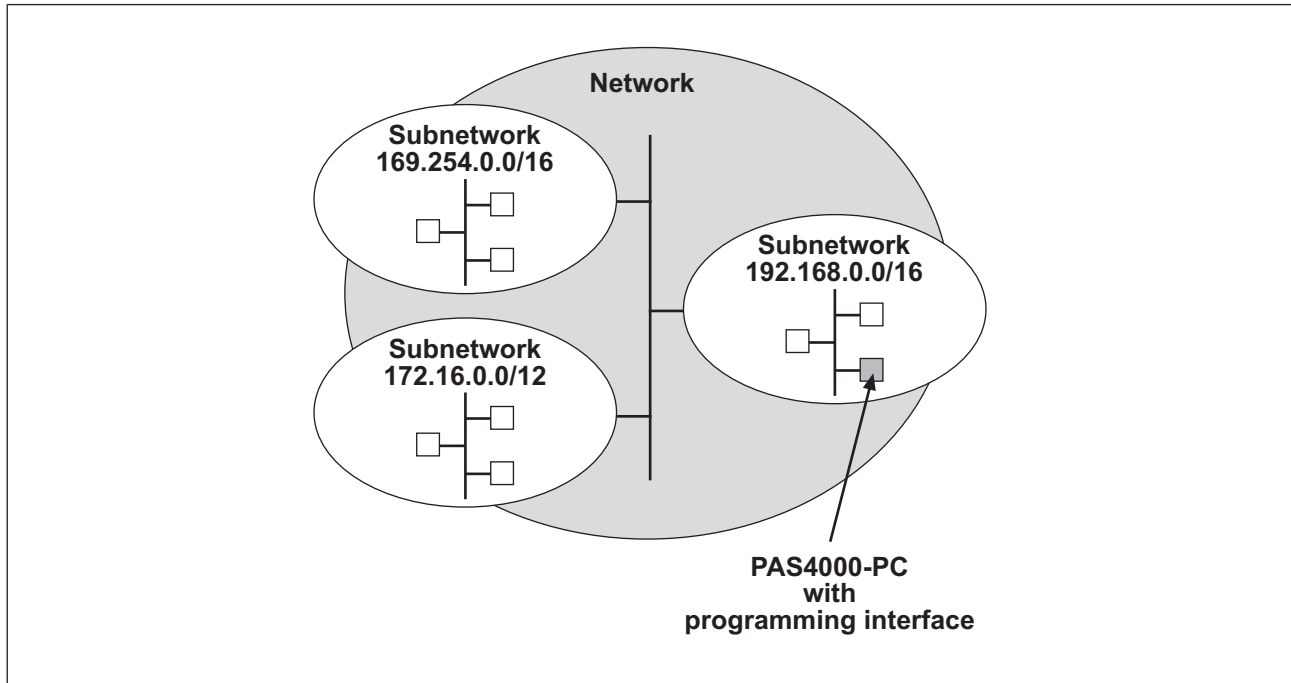


Fig.: Example for an Ethernet network without Router

► Scan

In PAS4000 there are various options for finding SafetyNET p subscribers (e.g. scan network, scan project).

- The "Scan network" process finds all the PSS 4000-exclusive/operable devices in a network connected to the same network adapter as the programming interface.
- The "Scan project" process finds all the PSS 4000-exclusive/operable devices that have been configured within the project and are located in a network connected to the same network adapter as the programming interface.

► Device naming via PAS4000

Device naming can be performed on devices found during the scanning process.

► Process data exchange

To enable process data exchange between the PSS 4000 devices within a project, the devices must be in the same subnetwork. Switches and hubs can be interconnected.

► Other online actions

Other online actions include all online actions with the exception of scanning and device naming, e.g.:

- Direct project download
- Diagnostics (diagnostic list, diagnostic log)
- Debugging (show, control, force variables; dynamic program display, etc.)

To enable these online actions to be performed, the programming interface on the PAS4000-PC must be in the same subnetwork as the project's PSS 4000 devices. Switches and hubs can be interconnected.

9.11 Network with Routers

On Ethernet, a Router connects two different Ethernet networks. As an Ethernet network is identified by the Net-ID (= part of the IP address), the Net-ID can be used to decide whether the subscriber is in the same network or whether data packets must be transferred to a Router to be forwarded.

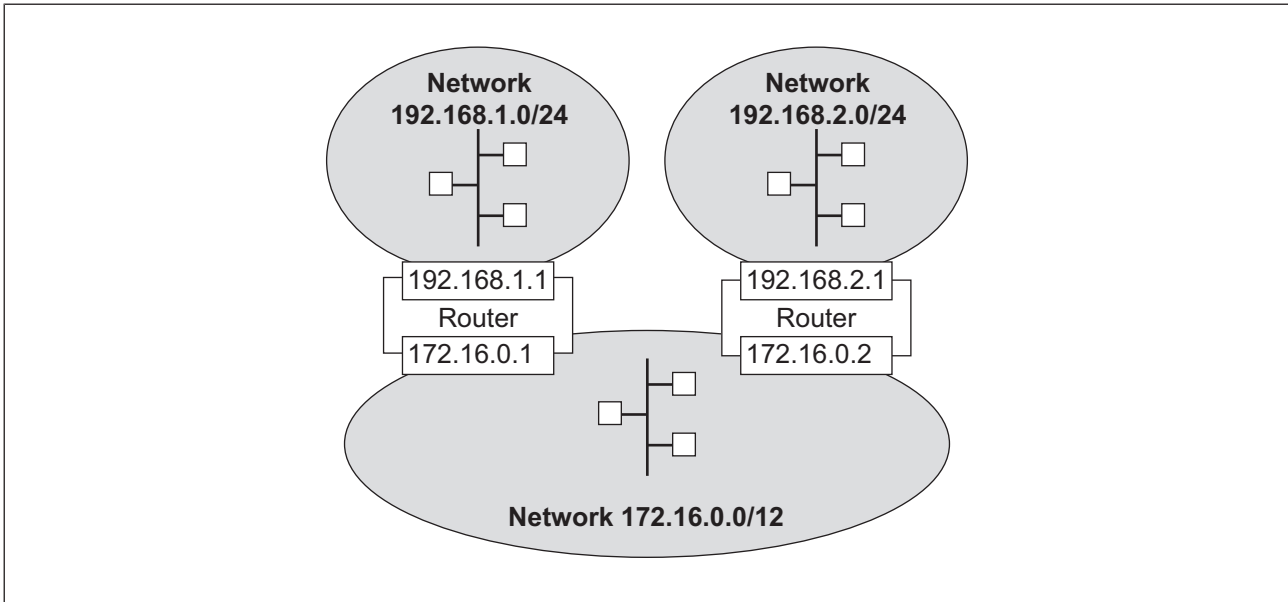


Fig.: Example for connecting Ethernet networks using Routers

If Routers are used in a SafetyNET p network, please note the following:

- Requirements of Routers

Only Routers that support a maximum transmission unit (MTU) of min. 302 Bytes can be used in a SafetyNET p network.

- Scan

In PAS4000 there are various options for finding SafetyNET p subscribers (e.g. scan network, scan project).

To prevent the system searching for all potential PSS 4000 devices worldwide, the scan function on PAS4000 works on the basis of an IP multicast (multicast address 224.0.23.61). The default setting for "Time To Live" (TTL = 1) prevents scanning across Routers with PAS4000 (see [System variable PILZ_SNP_MULTICAST_TTL \[196\]](#)).

- Device naming via PAS4000

If networks are connected via Routers, the default setting for "Time To Live" (TTL = 1) must have device naming performed directly from the respective network and not performed centrally (see [System variable PILZ_SNP_MULTICAST_TTL \[196\]](#)).

- Process data exchange

Provided a Router is configured correctly (e.g. Routing table), process data exchange will also work with PSS 4000-exclusive/operable devices which, from the perspective of the PAS4000-PC, are in a network behind a Router.

► Other online actions

Provided a Router is configured correctly (e.g. Routing table), the other online actions will also work for PSS 4000-exclusive/operable devices which, from the perspective of the PAS4000-PC, are in a network behind a Router. Other online actions include all online actions with the exception of scanning and device naming, e.g.:

- Direct project download
- Diagnostics (diagnostic list, diagnostic log)
- Debugging (show, control, force variables; dynamic program display, etc.)



NOTICE

Please note that MS Windows does not support routing to "Private IP addresses" (Private IPs) on the Internet. Private IP addresses are in private IP address ranges and have been defined by IANA as follows:

- 10.0.0.0 ... 10.255.255.255
10.0.0.0/8: Private network Class A with subnet mask 255.0.0.0
- 172.16.0.0 ... 172.31.255.255
172.16.0.0/12: Private network Class B with subnet mask 255.240.0.0
- 192.168.0.0 ... 192.168.255.255
192.168.0.0/16: Private network Class C with subnet mask 255.255.0.0

9.11.1 System variable PILZ_SNP_MULTICAST_TTL

Time To Live (TTL) is a parameter of the Internet Protocol (IP). TTL can be used to determine the operating range of IP packets.

Each router that an IP packet passes on the way from the transmitter to the target reduces the TTL value of the IP packet by 1. An IP packet whose TTL value falls to 0 is not transmitted by a router and is discarded.

Scanning and device naming are performed depending on the settings of the relevant TTL. The TTL value for these two actions can be configured on the PAS4000 PV via the system variable PILZ_SNP_MULTICAST_TTL.

Beside the correct configuration of a router, the setting for PILZ_SNP_MULTICAST_TTL is the key factor for reaching the PSS 4000 devices that are behind a router, as seen from the PAS4000 PC, when scanning and naming devices.



NOTICE

Please note that if PILZ_SNP_MULTICAST_TTL > 1 it is possible to intervene across all routers in other networks. This may jeopardise network safety.

The following applies for the configuration of the system variable PILZ_SNP_MULTICAST_TTL:

- ▶ Default TTL when system variable PILZ_SNP_MULTICAST_TTL is not available: 1
To prevent the system searching for all potential PSS 4000 devices worldwide, the default setting prevents scanning across routers and device naming across routers. It is not necessary to create and configure the system variable PILZ_SNP_MULTICAST_TTL if PAS4000 is to use the default setting. PAS4000 automatically uses the default setting when the system variable is not found at start-up.
- ▶ System variables are configured in the Windows 10 operating system under **Environment variables**.
Enter "Environment variable" in the Windows search field and select **Edit environment variables for your account**. In the dialogue box you can configure the variables for your own account and for all users. You will need administrator rights to configure the system variables for all users.
- ▶ Valid configuration for PILZ_SNP_MULTICAST_TTL: Integer value in the range of 1 ... 64
 - If a configuration is invalid (e. g. letters, value is not an integer) the default setting is used for PILZ_SNP_MULTICAST_TTL.
 - If a value > 64 is configured, the value 64 is used for PILZ_SNP_MULTICAST_TTL.
- ▶ The creation/change of the system variables PILZ_SNP_MULTICAST_TTL will not become effective until the PAS4000 PC is restarted.



NOTICE

If the OPC Server is installed on the same PC as PAS4000, the OPC server sets the system variable PILZ_SNP_MULTICAST_TTL to 64 during installation.

9.12 SafetyNET p via WLAN

In principle it is possible to use SafetyNET p in a wireless network in accordance with the WLAN standard. However, it is not advisable to replace a cable-based network with WLAN without first carrying out various analyses and taking additional supportive measures. Above all it is necessary to consider that if you are using a safe protocol, WLAN is not suitable for every application.

SafetyNET p has a safe data channel for safety-related data (FS communication). Transmission errors in FS communication will always be detected with SafetyNET p, with the system section switching to an appropriate operating state. If SafetyNET p is used in a WLAN that has low availability in terms of transmission, frequent errors in the WLAN communication can mean that FS communication is deactivated.

Please note the following:

- ▶ Before using WLAN, you should arrange for an expert to carry out a detailed on-site analysis. This analysis should clarify whether a stable, sufficiently error-free wireless transmission is possible. It's mainly the availability of the WLAN transmission that should be analysed here. In the analysis, particular attention should be paid to the availability in the reception areas.
- ▶ If possible, only devices that are suitable for transmitting safe protocols should be used as WLAN routers. For example, this is the case on devices that operate with a constantly high signal quality and seamless, redundant WLAN connections (e.g. PRP).
- ▶ If possible, use SafetyNET p protocol version 2. As communication errors can occur more frequently with WLAN connections, it is an advantage to use SafetyNET p protocol version 2. It supports the resumption of FS communication after a communication error (see [SNp protocol version 2: Communication errors and resumption of FS communication](#) [ 181]).



NOTICE

SafetyNET p protocol version on 2nd generation PSS 4000-exclusive devices

2nd generation PSS 4000-exclusive devices (PSS u2 systems) only use SafetyNET p protocol version 2.

If a PSS 4000 project contains both 1st generation PSS 4000-exclusive devices (PSSu systems, PSS67 devices) and 2nd generation PSS 4000-exclusive devices (PSS u2 systems), only SafetyNET p protocol version 2 can be used.

- ▶ Even if SafetyNET p protocol version 2 is used to resume FS communication once the cause of the error has been recovered, these errors can significantly affect the availability of a PSS 4000 device in some circumstances. Please note that the substitute values with valid bit = FALSE will be used for the affected process data while an error is present. As a result, the affected hardware outputs may switch to a safe state.
- ▶ Compared with a cable-based network, a WLAN generally has a lower bandwidth and higher latency, so the task cycle time for data transmission via WLAN may need to be configured higher than for a comparable cable-based network.

In an individual case this may mean the following:

To avoid interruptions to data transmission via WLAN, the cycle time of SafetyNET p (t_{SNp}) must be increased. However, the cycle time of SafetyNET p cannot be configured directly within a PSS 4000 project. It can only be influenced indirectly via task cycle times. Should the cycle time of SafetyNET p be greater than the cycle time calculated for the module bus (t_{MBUS}), then the I/O data of a device's module bus may not be mapped directly to the PI variables of blocks on another device. I/O mapping may only occur between PI variables on the two devices. The cycle time of SafetyNET p can then be increased indirectly via the blocks' task cycle times (t_{Task}).

Further information can be found under [Times that influence reaction times !\[\]\(dfbd6b3763a6d1d9afaa974f64e2e4b5_img.jpg\) 572](#) for 1st generation devices and under [Times that influence reaction times !\[\]\(b89ecf30df3dbaee65fa9f1829524a6e_img.jpg\) 588](#) for 2nd generation devices.

9.13 Availability optimised in SafetyNET p

SafetyNET p is one of the Ethernet-based bus systems. In principle, therefore, it is possible to operate the automation area (machine environment, field level) in the same network as an Ethernet-based company network (office area, building services, logistics). However, we would strongly advise against mixing the two areas, for several reasons.

Apart from the potential security problems, the vastly different communication requirements of the two areas should be taken into account. In an office, for example, typically there are large amounts of data to manage, while in the automation area it is mainly real-time behaviour that matters.

With Ethernet, the network load is increased with each subscriber and, what's more, office devices typically generate a higher network load than automation devices, so the two areas should be separated. Various known strategies are relevant (e.g. physical separation of both areas, network segmentation).

When building a network with SafetyNET p, a careful approach is needed even at the planning stage. To limit the network load and therefore increase availability, strategies should be developed for the following requirements in particular:

- ▶ Demarcation of the automation area (SafetyNET p), if the automation area has to be integrated into the higher level company network (office area).
- ▶ Removal of devices that permanently or sporadically generate a high network load from the automation area (SafetyNET p). If this is impossible, these devices should at least be demarcated from SafetyNET p, so that their network load has no effect in SafetyNET p. This may involve the following devices, for example:
 - Printers
 - Video cameras
 - Devices that frequently transmit broadcast or multicast telegrams



INFORMATION

To increase the availability of a control system we recommend that devices that permanently or sporadically generate a high network load should not be operated in the same network as the control system. In extreme cases, the network load of these devices can lead to a major FS error (see Operating status "Major FS error").

If you do not follow this recommendation it will not affect the safe function of the control system.

9.14 Prevent data exchange with projects with identical project name

Data exchange between PSS 4000 projects is only permitted when cross-project Snp communication is used. With cross-project Snp communication, PASconnect prevents PSS 4000 projects having identical project names.

However, where there is no actual data exchange between PSS 4000 projects in which FS data is processed, the situation can arise in which these projects have identical project names. If such projects are operated in the same SafetyNET p network, PSS 4000 cannot prevent cross-connections during data exchange unless additional measures are used. PSS 4000 projects that have identical project names and process FS data should be operated in closed SafetyNET p networks. A SafetyNET p network is deemed to be "closed" when it is guaranteed that SafetyNET p data traffic does not extend beyond the network boundary.

**INFORMATION**

If you **cannot** guarantee that the PSS 4000 projects are in closed SafetyNET p networks, then the projects may not have identical project names. Rename the projects.

10 External communication

10.1 Communication modes for external communication

External communication refers to data exchange in which output data (external O-data) leaves a PSS 4000 project and input data (external I-data) enters a PSS 4000 project. This is achieved with the help of various communication modes. The communication mode supported by a PSS 4000 device depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.

Communication modes for external communication are divided into "Fieldbuses" and "IP connections".

10.1.1 Fieldbuses

The term "Fieldbuses" embraces industrial fieldbuses with a standardised protocol (e.g. PROFIBUS-DP). For external communication via a fieldbus, a PSS 4000 device must have the corresponding system section (e.g. PROFIBUS-DP Slave system section). The physical connection is established via defined interfaces, depending on the fieldbus.

10.1.2 IP connections

The term "IP connections" embraces all protocols and communication links that are addressed via IP addresses (Modbus/TCP, Raw TCP, Raw UDP). For this type of external communication a PSS 4000 device must have the "IP connections" system section. The physical connection is established via an Ethernet interface.



NOTICE

Maximum number of IP connections

The "IP connections" system section can manage a maximum of 48 IP connections in total. This upper limit also applies if you are operating the "IP connections" system section with a variety of protocols or communication links.

SafetyNET p connections are **not** managed by the "IP connections" system section and so do not affect this upper limit.



NOTICE

If an IP connection cannot be established during the first initialisation attempt or if an IP connection is interrupted, reinitialisation will not take place immediately.

This behaviour results from the fact that IP connections are initialised successively. The time that elapses between the individual initialisations is a multiple of the highest common factor of the cycle time (t_{ExtCo}) of the individual IP connections. Depending on this time, it may take a considerable time for all IP connections to be restored under certain circumstances.

Special features with external communication via UDP

PSS 4000 provides two different methods of external communication via UDP:

- ▶ Raw UDP
- ▶ UDP communication using blocks

The IP connections system section is needed for both options. While communication via Raw UDP is governed by the rules for IP connections, UDP communication using blocks is subject to its own rules.

The following table compares the key features:

Raw UDP	UDP communication using blocks
Requires PSS 4000 device with IP connections system section	
Influences the upper limit of a max. total of 48 IP connections	- - -
Cyclical data transfer with t_{ExtCo}	Data transfer controlled via the user program
Configuration of connections in PAS4000	Configuration of PSS 4000 system control blocks
- - -	Use of the PSS 4000 system control blocks on the ST resource and/or FS resource
Data transfer via connections with configured data length (= fixed data length)	Data length configurable on PSS 4000 system control block; data transfer possible with variable data length
Possible UDP connections: ▶ Unicast (1:1 connection) ▶ Broadcast (1:"all" connection) ▶ Multicast (1:n connection)	Possible UDP connections: ▶ Unicast (1:1 connection; preferred connection type) ▶ Broadcast (1:"all" connection) ▶ Multicast (1:n connection; send only)
IO mapping required in PAS4000	Mapping of PSS 4000 system control blocks with I/O data

10.2 Modbus/TCP

Modbus is an open fieldbus standard published by the user organisation MODBUS-IDA.

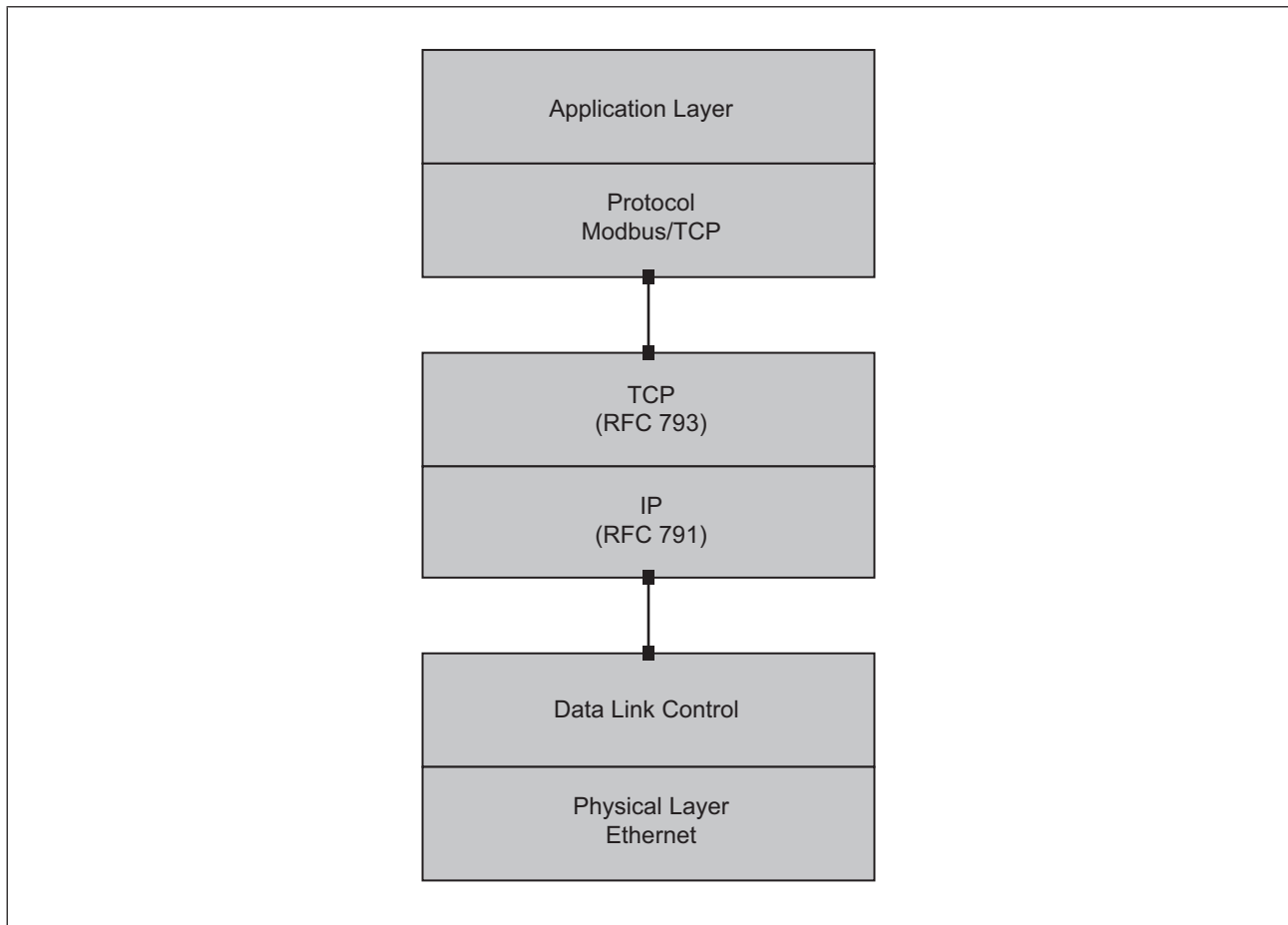


Fig.: Communication of a PSS 4000 device via Modbus/TCP

Modbus/TCP is a protocol based on Industrial Ethernet (TCP/IP over Ethernet). It is one of the protocols with Client/Server communication. Data is transferred via a request/response mechanism using function codes (FC).

Modbus/TCP is connection-oriented, i.e. before the payload is transferred via Modbus/TCP, a connection must first be established between two Modbus/TCP interfaces. The initiator of the connection is called the "Client". The communications partner with which the Client establishes the connection is called the "Server". Whether a connection on a device assumes the role of Client or Server is defined when a connection is configured. As a result, the Server/Client role only applies for that specific connection.

Data exchange via Modbus/TCP can be used on the automation system PSS 4000 for non-safety-related applications.

10.2.1 "IP connections" system section

In the automation system PSS 4000, external communication via Modbus/TCP is possible with all PSS 4000 devices that have the "IP connections" system section. In this case, the "IP connections" system section of a PSS 4000 device performs all the tasks required for communication. For example, these include:

- ▶ Management of the Modbus/TCP connections
(see [Modbus/TCP connections](#)  205)
- ▶ Management of the Modbus/TCP data areas
(see [Modbus/TCP data areas](#)  206)
- ▶ Support for function codes
(see [Function codes](#)  207)
- ▶ Data exchange
(see [Data exchange via Modbus/TCP](#)  209)

10.2.2 Modbus/TCP connections

The "IP connections" system section of a PSS 4000 device manages the Modbus/TCP connections that are configured for the device. The connection is configured in the PAS4000's IP Connections Editor.

Several Modbus/TCP connections can be configured for a PSS 4000 device (see [Data transfer limits](#) [212]). When a connection is configured you have the option to configure the connection for the role of Client or Server. If a Client role is configured for a connection, this is called a "Client connection" from the perspective of the PSS 4000 device, otherwise it is called a "Server connection" from the perspective of the PSS 4000 device.

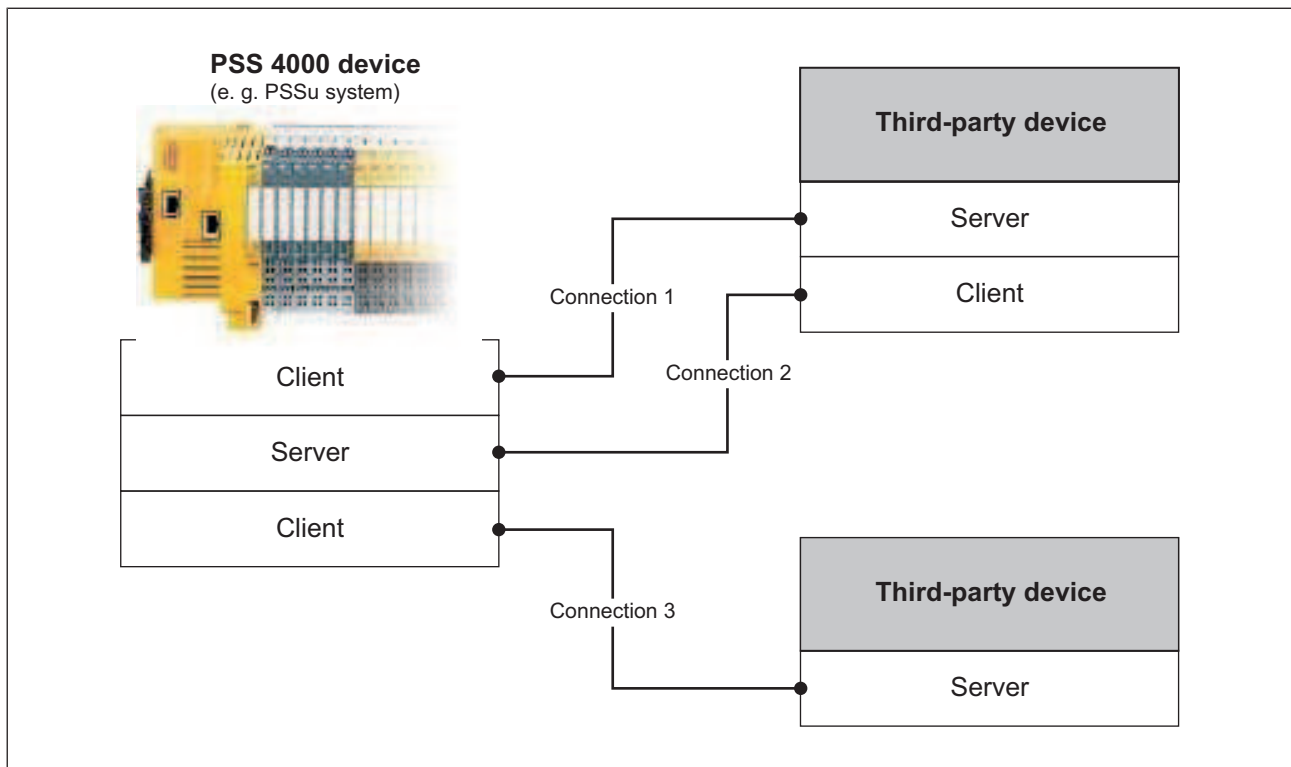


Fig.: Role assignment on Modbus/TCP connections (example)

Port numbers must be defined for data exchange via a Modbus/TCP connection. Server connections on a PSS 4000-exclusive device use Port 502 as a default setting. However, a different port number may be configured.



INFORMATION

Known applications (e.g. Telnet) are assigned fixed port numbers, to which each application can refer during data transfer. The range 0 ... 1023 contains reserved port numbers. Under no circumstances should these be used for your own applications. The complete list of reserved port numbers and their utilities are listed in RFC 1700.

Send and receive data is transferred cyclically on PSS 4000-exclusive devices. With cyclical transfer, a permanent connection is maintained between the connection Client and Server.

10.2.3 Modbus/TCP Server connection (server connections)

The "IP connections" system section of a PSS 4000 device supports the following Modbus/TCP data areas for connections with a Server role:

Data area	Modbus syntax	Example
Coils (Bit) 0x00000 ... 0x65535 [read/write]	0x[xxxxx]	0x05888
Discrete Inputs (Bit) 1x00000 ... 1x65535 [read only]	1x[xxxxx]	1x05888
Input Register (Word/16 Bits) 3x00000 ... 3x65535 [read only]	3x[xxxxx]	3x05888
Holding Register (Word/16 Bits) 4x00000 ... 4x65535 [read/write]	4x[xxxxx]	4x05888



INFORMATION

The "IP connections" system section of a PSS 4000 device starts the addressing for Modbus/TCP data areas at "0". On devices from other manufacturers, addressing may start at "1".

Please refer to the operating manual provided by the relevant manufacturer.

10.2.4 Function Codes (Client connections)

The "IP connections" system section of a PSS 4000 device supports the following function codes (FC) for connections with a Client role:

Function code	Function	
FC 01	Read Coils	The connection Client reads bit data from the connection Server, data length ≥ 1 Bit, e.g. digital inputs (receive data from 0x)
FC 02	Read Discrete Input	The connection Client reads bit data from the connection Server, data length ≥ 1 Bit, e.g. digital inputs (receive data from 1x)
FC 03	Read Holding Register	The connection Client reads word data from the connection Server, data length ≥ 1 Word, e.g. analogue inputs, counter inputs (receive data from 4x)
FC 04	Read Input Register	The connection Client reads word data from the connection Server, data length ≥ 1 Word, e.g. analogue inputs, counter inputs (receive data from 3x)
FC 05	Write Single Coil	The connection Client writes to one bit datum in the connection Server, data length = 1 Bit, e.g. digital input (send data to 0x)
FC 06	Write Single Register	The connection Client writes to one word datum in the connection Server, data length = 1 Word, e.g. analogue output, counter output (send data to 4x)
FC 15	Write Multiple Coils	The connection Client writes to multiple bit data in the connection Server, data length > 1 Bit, e.g. digital outputs (send data to 0x)
FC 16	Write Multiple Registers	The connection Client writes to multiple word data in the connection Server, data length > 1 Word, e.g. analogue inputs, counter inputs (send data to 4x)
FC 23	Read/Write Multiple Registers	The connection Client reads and writes multiple word data within a telegram (receive data from 4x and send data to 4x)



INFORMATION

A function code must be configured for each Client connection. This can be done in the PAS4000's IP Connections Editor.

10.2.5 Relationship between function codes and Modbus/TCP data areas

For Client connections, the following relationship exists between function codes and Modbus/TCP data areas:

Data area (Remote Address)	Send (Client)		Receive (Client)
0x Coils (Bit)	Data length	Function code	Function code
	= 1Bit	FC = 05 (Write Single Coil)	FC = 01 (Read Coils)
	> 1 Bit	FC = 15 (Write Multiple Coils)	
1x Discrete Inputs (Bit)	---		FC = 02 (Read Discrete Inputs)
3x Input Register (Wort)	---		FC = 04 (Read Input Register)
4x Holding Register (Word)	Data length	Function code	FC = 03 (Read Holding Registers)
	= 1 Word	FC = 06 (Write Single Register)	
	> 1 Word	FC = 16 (Write Multiple Registers)	
If there is an entry for both send and receive data, function code FC = 23 is used (Read/Write Multiple Registers).			

10.2.6 Data exchange via Modbus/TCP

It is always the connection Client that initiates data transfer. For a Client connection, the direction of the data flow (send/receive) and data area (Coils, Discrete Inputs, Input Registers, Holding Register) is established by defining the function code when the connection is configured. The data length is also defined when the connection is configured.

10.2.6.1 Send data and receive data

When sending, the connection Client sends data to a data area belonging to the connection Server. This is also known as "Writing data". For this to work, the target data destination must be configured on the device with the role of Client.

When receiving, the connection Client requests and receives data located in a data area belonging to the connection Server. This is also known as "Reading data". For this to work, the data source must be configured on the device with the role of Client.

From the perspective of the automation system PSS 4000, receive data is called external I-data, and send data is called external O-data (see [Validity of process data](#) [359]).

Please ensure that the relevant data area (data destination, data source) is available on the Server (see [Server allocation table](#) [210]).

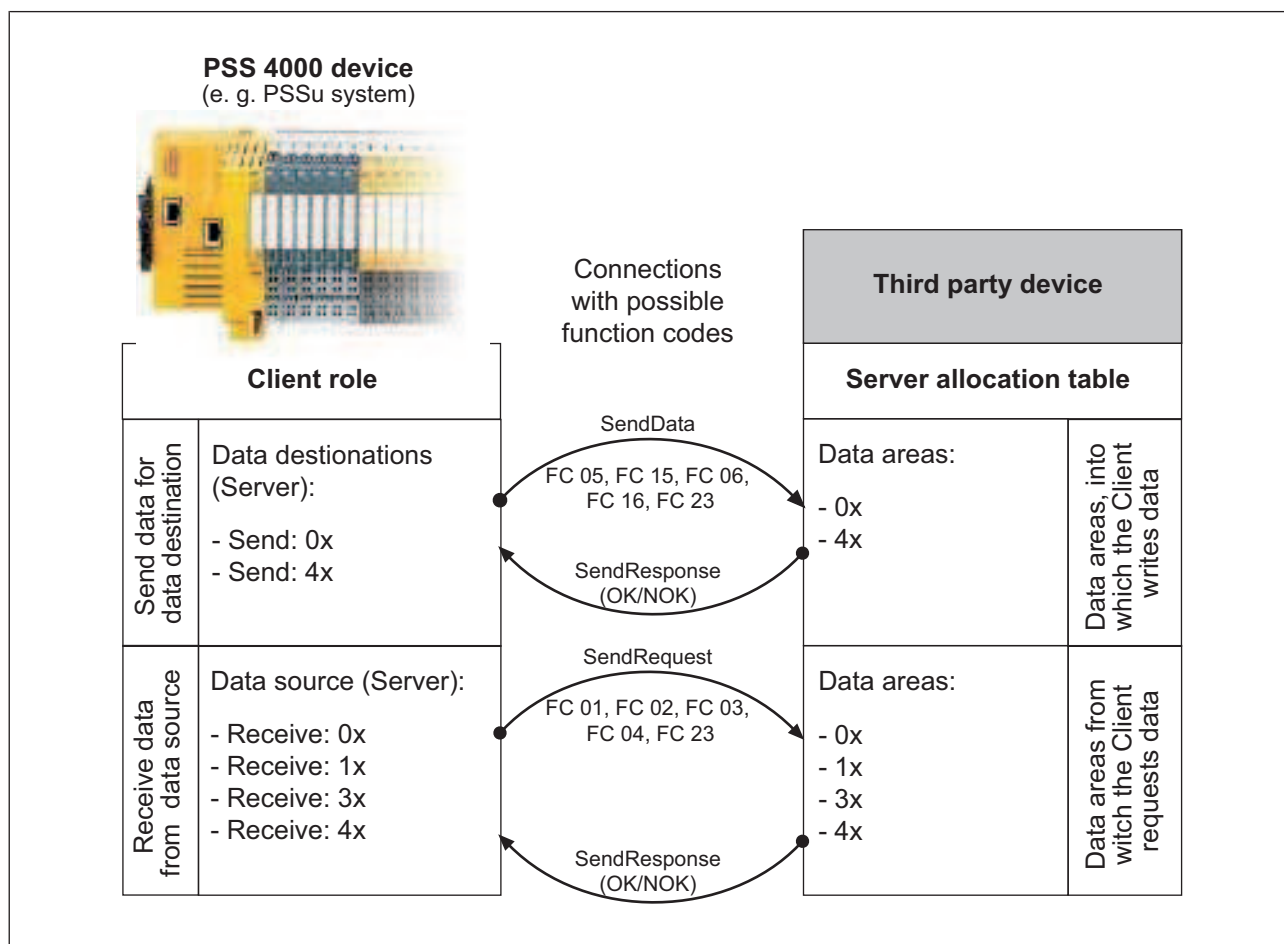


Fig.: PSS 4000 device with "IP connections" system section and Client role for connections (principle)

10.2.6.2 Server allocation table

To ensure that the connection Client can send data to the connection Server and receive data requested from the connection Server, the relevant data areas must be configured on the Server.

With Server connections this is achieved by configuring a Server allocation table when the connection is configured.

The following applies:

- ▶ The connection Client writes data to the Server allocation table's write data area (0xWrite, 4xWrite).
- ▶ The connection Client requests data from the Server allocation table's read data area (0xRead, 1x, 3x, 4xRead).

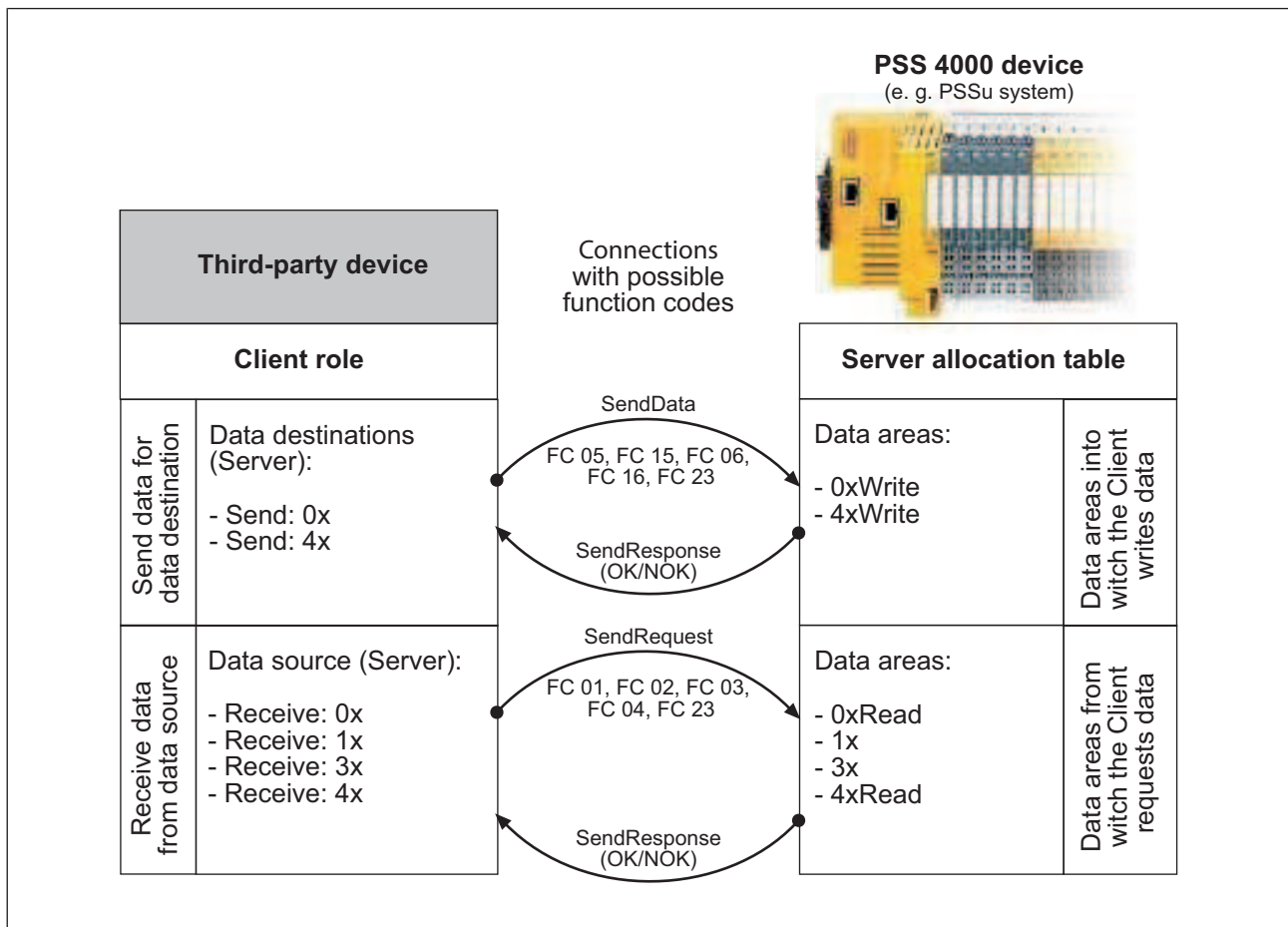


Fig.: PSS 4000 device with "IP connections" system section and Server role for connections (principle)

Limits when configuring a server allocation table in PAS4000

Each data area can have a start address in the range 0 ... (65536 – data length) and a maximum data length of 8000:

Data area	Start address	Data length
0xRead	0 ... (65536 – data length)	1 ... 8000
0xWrite		
1x		
3x		
4xRead		
4xWrite		

Example structure of a server allocation table in PAS4000:

Data area	Start address	Data length
0xRead	0	4
0xWrite	100	4
1x	0	0
3x	0	0
4xRead	0	2
4xWrite	100	2

10.2.7 Data transfer limits

Max. number of Modbus/TCP connections

The max. number of Modbus/TCP connections (Client connections and/or Server connections) depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.

Please also note the general upper limit of 48 IP connections for a PSS 4000 device.



NOTICE

Maximum number of IP connections

The "IP connections" system section can manage a maximum of 48 IP connections in total. This upper limit also applies if you are operating the "IP connections" system section with a variety of protocols or communication links.

SafetyNET p connections are **not** managed by the "IP connections" system section and so do not affect this upper limit.

Limits on Client connections

On Client connections, the maximum data length per telegram is restricted. The following table contains information on the maximum data length per telegram that is supported:

Data transfer		Data length per telegram
Read data (Bit)	FC 01 (Read Coils)	1 ... max. 2000
	FC 02 (Read Discrete Inputs)	
Read data (Bit)	FC 05 (Write Single Coil)	1 Bit
	FC 15 (Write Multiple Coils)	1 ... max. 1968
Read data (Word)	FC 03 (Read Holding Registers)	1 ... max. 125
	FC 04 (Read Input Register)	
Write data (Word)	FC 06 (Write Single Register)	1 Word
	FC 16 (Write Multiple Registers)	1 ... max. 123 Words
Read and write data (Word)	FC23 (Read/Write Multiple Registers)	Read 1 ... max. 125 Words Write 1 ... max. 121 Words

FC 03, FC 04 and FC 16 support data transfer with multiple telegrams. If the max. supported data length per telegram is exceeded when transferring data with FC 03, FC 04 or FC 16, the data will be spread over multiple telegrams. The following table contains details of the data length when transferring data with multiple telegrams:

Data transfer with multiple telegrams		Data length
Read data (Word)	FC 03 (Read Holding Registers)	From 126 ... max. 2000 Words
	FC 04 (Read Input Register)	
Write data (Word)	FC 16 (Write Multiple Registers)	From 124 ... max. 2000 Words

When a Client connection is configured in PAS4000 it is possible to influence the transfer of multiple telegrams. If multiple telegrams are to be transferred, you can configure whether one (sub)telegram is sent per connection cycle and whether it is necessary to wait for a response to each (sub)telegram before sending the next (sub)telegram. Otherwise, all (sub)telegrams in a connection cycle will be sent one after the other, without waiting for a response. The responses to the (sub)telegrams will only be collected once the transfer is complete, which may optimise data transfer.



INFORMATION

When devices from a third-party manufacturer are used as communication partners, limits regarding data lengths may vary. Please refer to the information stated in the operating manual of the relevant unit.

10.2.8 Modbus/TCP I/O data

The "IP connections" system section enables a PSS 4000 device to exchange I/O data with third-party devices via Modbus/TCP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding Modbus/TCP data areas.

I/O mapping of Modbus/TCP data areas

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O mapping for Modbus/TCP data areas:

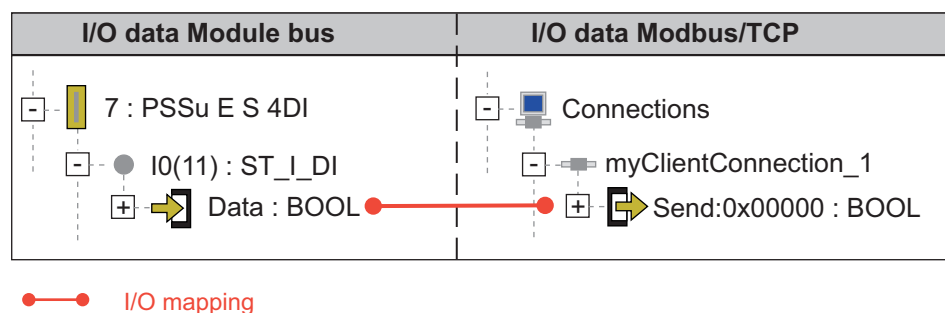
I/O mapping		on Modbus/TCP			
		Client role		Server role (Server allocation table)	
		Transmit data areas (0x, 4x)	Receive data areas (0x, 1x, 3x, 4x)	Read data areas (0x, 1x, 3x, 4x)	Write data areas (0x, 4x)
from module bus	I-datum (e.g. HW input)	◆	---	◆	---
	O-datum (e.g. HW output)	---	◆	---	◆
from PI variable	I-PI variable	---	◆	---	◆
	O-PI variable	◆	---	◆	---

◆ I/O mapping possible
 --- I/O mapping impossible

Examples

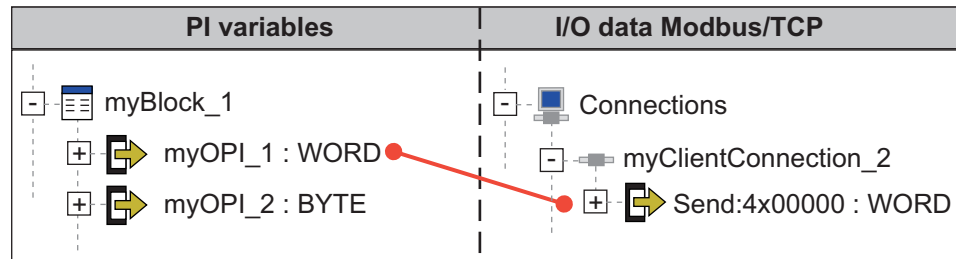
I/O mapping for Client roles

- ▶ The input signal at a hardware input is to be sent to a device from another manufacturer via a Modbus/TCP connection.
 - Direction of data flow: Hardware input -> Modbus/TCP
 - Used function code: FC05/FC15 (send)
 - Possible I/O mapping (example):



- ▶ The content of an O-PI variable from the user program is to be sent to a device from another manufacturer via a Modbus/TCP connection.
 - Direction of data flow: O-PI variable -> Modbus/TCP

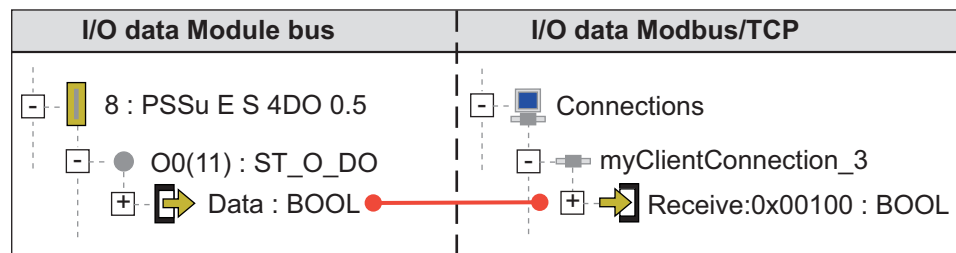
- Used function code: FC06/FC16/FC23 (send)
- Possible I/O mapping (example):



— I/O mapping

- The hardware output on a PSS 4000 device is to be controlled by a device from another manufacturer via a Modbus/TCP connection.

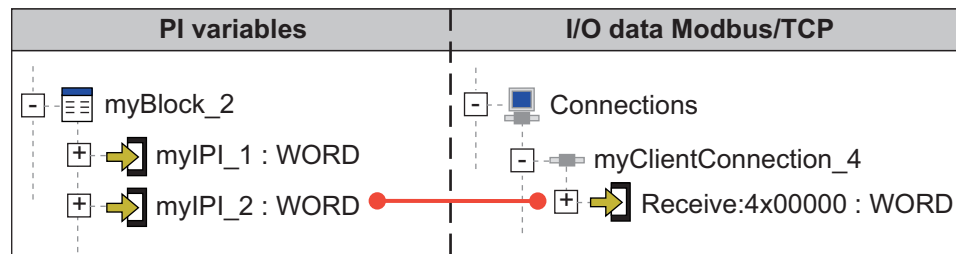
- Direction of data flow: Modbus/TCP -> Hardware output
- Used function code: FC01 (receive)
- Possible I/O mapping (example):



— I/O mapping

- A datum is to be requested by a device from another manufacturer via a Modbus/TCP connection. Once the data has been received it is to be made available for further processing within the user program.

- Direction of data flow: Modbus/TCP -> I-PI variable
- Used function code: FC03/FC23 (receive)
- Possible I/O mapping (example):

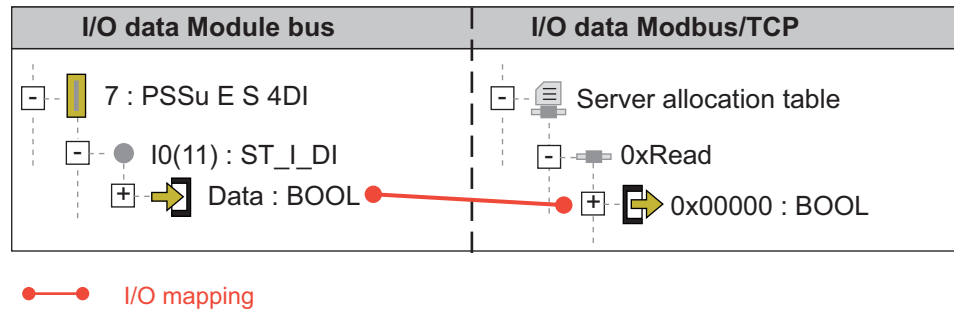


— I/O mapping

I/O mapping for Server roles (Server allocation table)

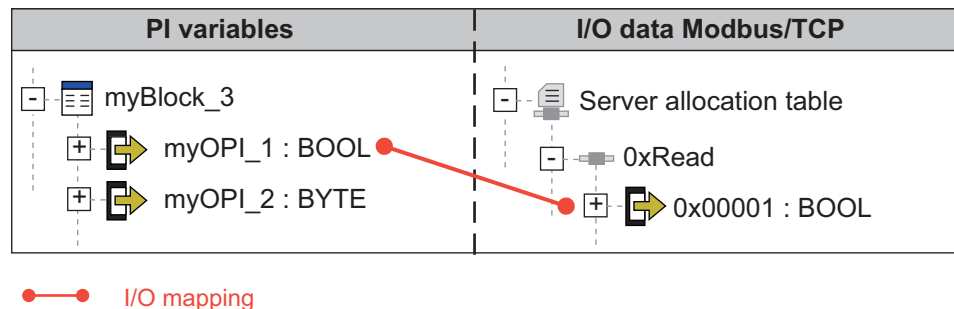
- ▶ The signal from a hardware input is made available in the Server allocation table's read data area. The piece of data can be read by a device from another manufacturer via a Modbus/TCP connection.

- Direction of data flow: Hardware input -> Modbus/TCP
- Possible I/O mapping:



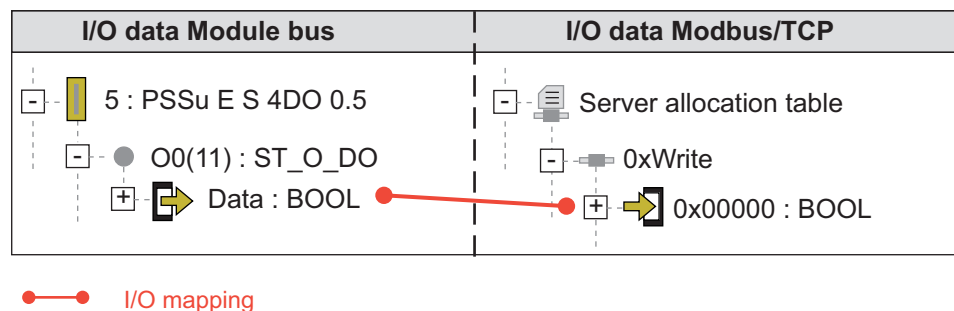
- ▶ The content of an O-PI variable is made available in the Server allocation table's read data area. The piece of data can be read by a device from another manufacturer via a Modbus/TCP connection.

- Direction of data flow: O-PI variable -> Modbus/TCP
- Possible I/O mapping:



- ▶ The hardware output on a PSS 4000 device is to be controlled by a device from another manufacturer via a Modbus/TCP connection. The device from another manufacturer writes to the Server allocation table's write data area.

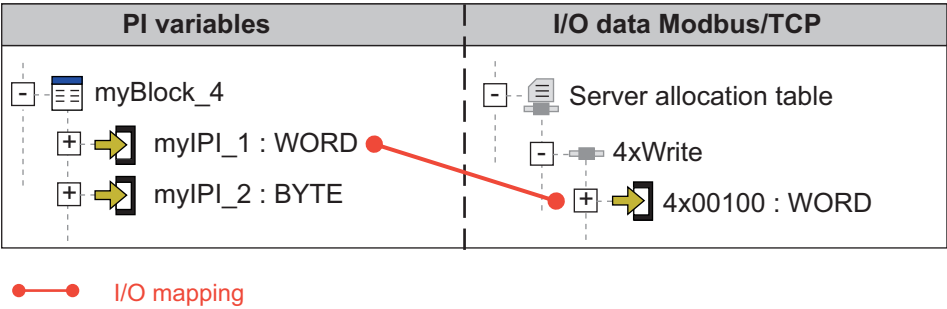
- Direction of data flow: Modbus/TCP -> Hardware output
- Possible I/O mapping:



- ▶ A piece of data is to be written to the Server allocation table's write data area by a device from another manufacturer via a Modbus/TCP connection. The data is to be processed further within the user program.

- Direction of data flow: Modbus/TCP -> I-PI variable

– Possible I/O mapping:



10.2.9 Address display of Modbus/TCP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

10.2.9.1 Address format of send data

The addresses of Modbus/TCP send data (Client role) are structured as follows:

<Device name>.Modbus.Client.<Connection name>.Send.<Start address>

Examples:

myDevice_1.Modbus.Client.myConnection1.Send.0x00000

myDevice_1.Modbus.Client.myConnection2.Send.4x00005

10.2.9.2 Address format of receive data

The addresses of Modbus/TCP receive data (Client role) are structured as follows:

<Device name>.Modbus.Client.<Connection name>.Receive.<Start address>

Examples:

myDevice_1.Modbus.Client.myConnection3.Receive.0x00000

myDevice_1.Modbus.Client.myConnection4.Receive.1x00010

myDevice_1.Modbus.Client.myConnection5.Receive.3x00000

myDevice_1.Modbus.Client.myConnection6.Receive.4x00005

10.2.9.3 Address display in the Server allocation table

The addresses of the Modbus/TCP Server allocation table (Server role) are structured as follows:

<Device name>.Modbus.Server.<Selectable data area in the Server allocation table>.<Start address>

Examples:

myDevice_2.Modbus.Server.0xRead.0x00000

myDevice_2.Modbus.Server.0xWrite.0x00100

myDevice_2.Modbus.Server.1x.1x00010

myDevice_2.Modbus.Server.3x.3x00000

myDevice_2.Modbus.Server.4xRead.4x00005

myDevice_2.Modbus.Server.4xWrite.4x00005

10.2.10 Data consistency on Modbus/TCP

The data consistency of data sent (Client role) or provided (Server role) by third party devices is the responsibility of those devices. From the PSS 4000's perspective, no statement can be made regarding the data consistency of those devices.

Data consistency on receive data from a Client connection

If a PSS 4000-exclusive/operable device is the connection Client, from the device's perspective, the receive data from this connection (FC 01, FC 02, FC 03, FC 04, FC 23) is external I-data.

The external I-data from a Client connection is treated as consistent data in PSS 4000 (see [Data consistency](#)  377), provided it is transferred in a single telegram (see [Data transfer limits](#)  212).

The external I-data from a Client connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a Modbus/TCP Server connection" (see [Consistency ranges](#)  378).

Data consistency on receive data from a Server connection

If a PSS 4000-exclusive/operable device is the connection Server, from the device's perspective, the receive data from this connection (data areas 0xWrite or 4xWrite) is external I-data.

The external I-data from a Server connection is treated as consistent data in PSS 4000 (see [Data consistency](#)  377), provided it is transferred in a single telegram (see [Data transfer limits](#)  212).

The external I-data from a Server connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a Modbus/TCP Server connection" (see [Consistency ranges](#)  378).

Data consistency on a Client connection's send data

If a PSS 4000-exclusive/operable device is the connection Client, from the device's perspective, the send data from this connection is external O-data.

The external O-data of a Client connection (FC 05, FC 15, FC 06, FC 16, FC 23) is consistent if it comes from the same consistency range on a data source (see [Data consistency](#)  377).

Data consistency on a Server connection's send data

If a PSS 4000-exclusive/operable device is the connection Server, from the device's perspective, the send data (data areas 0xRead, 1x, 3x, 4xRead) is external O-data. The external O-data of a Server connection is consistent if it comes from the same consistency range on a data source (see [Data consistency](#)  377).

10.2.11 Special features when using the data areas 0x and 4x

On some devices, Client connections are used to write data to the data areas (0x, 4x) of the Server allocation table and then automatically request the data back immediately, to show on the display for example.

If the connection's Server is a PSS 4000-exclusive/operable device, this "data mirroring" will only work if additional measures are taken for the device. There are minor differences in handling for IEC 61131 programming and Multi programming.

Procedure for IEC 61131 programming

- ▶ Create a block
e.g. program-type POU
- ▶ Declare the required I-PI variables in the block
Via the I-PI variables, I/O-mapping is used to access the write data areas (0xWrite, 4xWrite) of the Server allocation table.
e.g. myIPI_1 AT %I* : WORD
- ▶ Declare the corresponding O-PI variables in the block
The O-PI variables are needed in order to restore the I-PI variables in the block. This is a condition for "data mirroring" the I-data to the Read data areas (0xRead, 4xRead) of the Server allocation table.
e.g. myOPI_1 AT %Q* : WORD
- ▶ "Mirror data" by restoring the I-PI variables to the O-PI variables in the block

Example (programming in IL):

```
RESTORE:           LD           myIPI_1
                   ST           myOPI_1
```

- ▶ I/O-map the I-PI variables to the write data areas (0xWrite, 4xWrite) of the Server allocation table (I/O Mapping Editor of PAS4000)
- ▶ I/O-map the O-PI variables to the read data areas (0xRead, 4xRead) of the Server allocation table (I/O Mapping Editor of PAS4000)
- ▶ Assign the block to a resource from a PSS 4000-exclusive/operable device
e.g. ST resource from myDevice_1

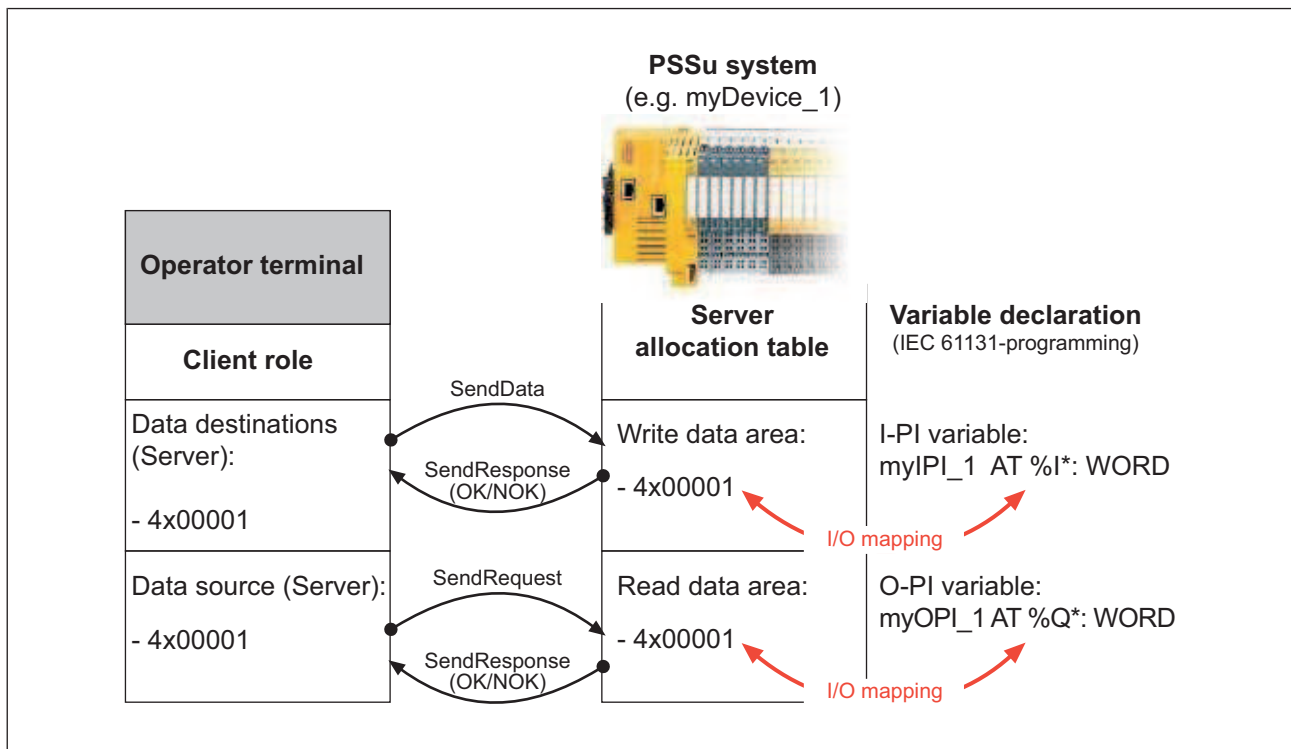


Fig.: Data mirroring in IEC 61131 programming (example)

Procedure for Multi programming

Option 1:

- ▶ Create an IEC 61131 basic block in the Multi program
- ▶ Declare the required PI variables in the IEC 61131 basic block and program "data mirroring"
This procedure is identical to the procedure for IEC-61131 programming (see above).
- ▶ I/O-map the I-PI variables to the write data areas (0xWrite, 4xWrite) of the Server allocation table
- ▶ I/O-map the O-PI variables to the read data areas (0xRead, 4xRead) of the Server allocation table
- ▶ Assign the block to a resource from a PSS 4000-exclusive/operable device
e.g. ST resource from myDevice_1

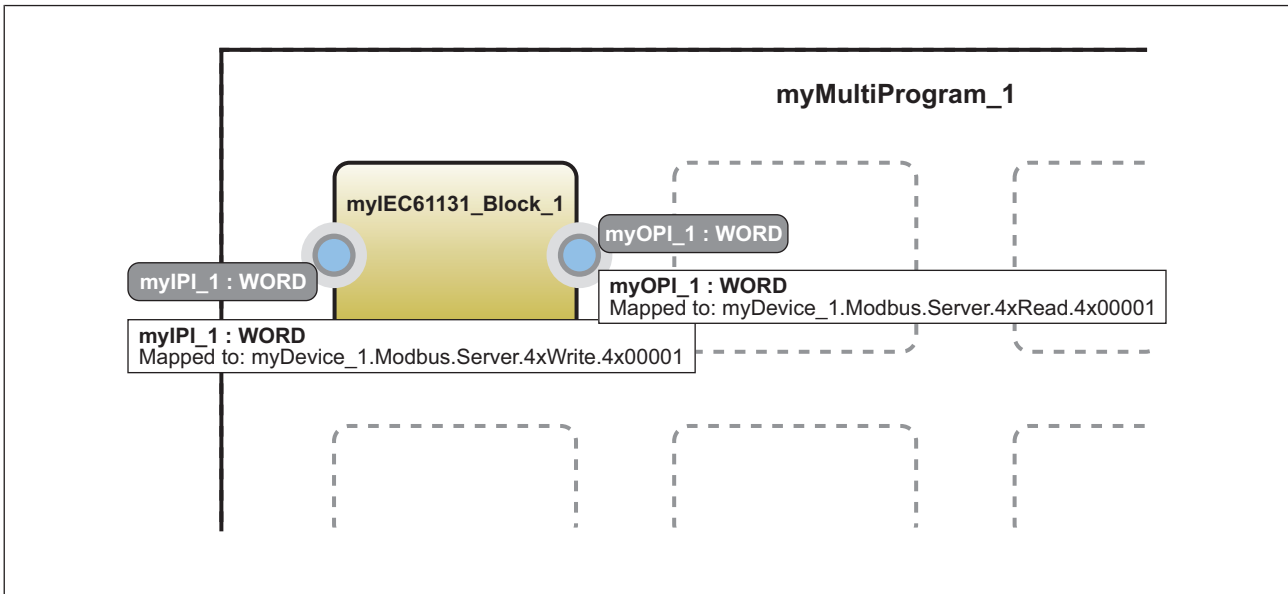
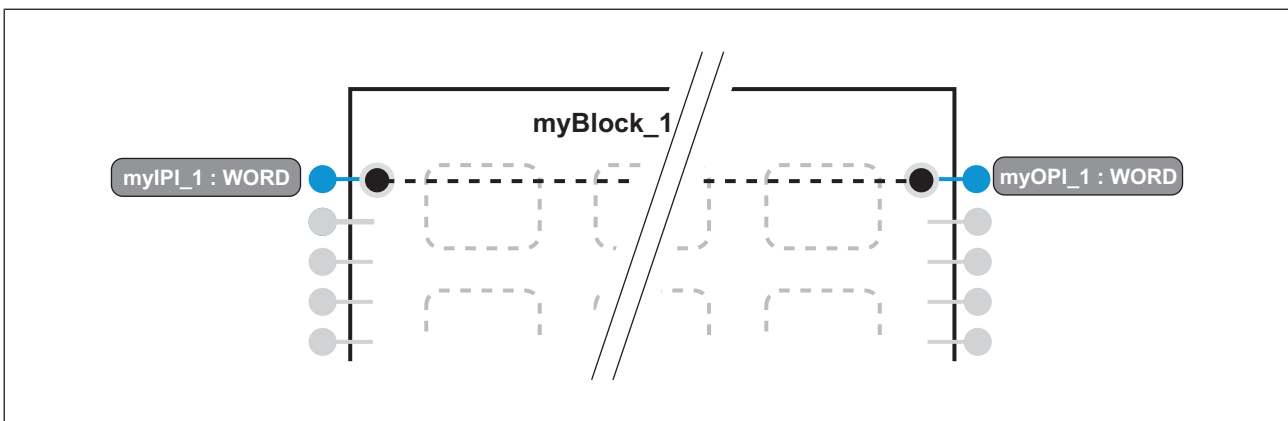


Fig.: I/O-mapping in Multi programming – IEC 61131 basic block (example)

Option 2:

- ▶ Create a component block (CB) in the Multi program
- ▶ Configure the block interface to
 - Configure the required I-PI points, which represent the I-PI variables
Via the I-PI points, I/O-mapping is used to access the write data areas (0xWrite, 4xWrite) of the Server allocation table.
e.g. myIPI_1 : WORD
 - Configure the corresponding O-PI points, which represent the O-PI variables
The O-PI points are needed in order to connect the I-PI points to the O-PI points within the block. This is a condition for "data mirroring" the I-data to the read data areas (0xRead, 4xRead) of the Server allocation table.
e.g. myOPI_1 : WORD
- ▶ "Mirror data" by connecting the I-PI points to the block's O-PI points

Example:



- ▶ Configure I/O mapping
 - I/O-map the I-PI points to the write data areas (0xWrite, 4xWrite) of the Server allocation table

- I/O-map the O-PI points to the read data areas (0xRead, 4xRead) of the Server allocation table

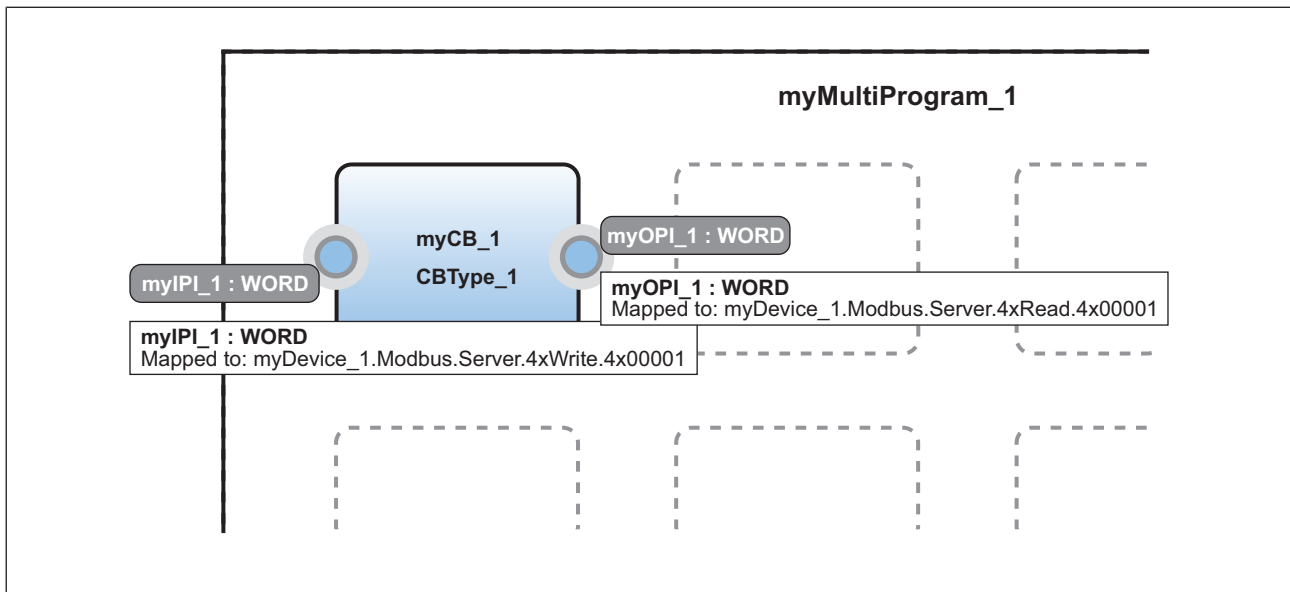


Fig.: I/O-mapping in Multi programming – Component block (example)

10.3 Raw TCP

The Transmission Control Protocol (TCP) is implemented in accordance with the following requirements:

- ▶ RFC793
- ▶ RFC1122
- ▶ RFC2988
- ▶ RFC5681

"Raw TCP" stands for data exchange via TCP, without an upper layer protocol.

TCP is a reliable, connection-oriented protocol. It is used for communication between two communication partners (end points). The communication service on a communication partner is identified via a port number. TCP uses the Internet Protocol (IP) for addressing. As such a TCP connection is uniquely identified by the following 4 values:

- ▶ IP address of source
- ▶ Port number of source
- ▶ IP address of destination
- ▶ Port number of destination

Data can be transmitted in both directions via a TCP connection specified in this way.

In the automation system PSS 4000, IP addresses, port numbers and transmission direction for Raw TCP connections are configured in PAS4000 when the connection is configured.

Data exchange via Raw TCP can be used in the automation system PSS 4000 for non-safety-related applications.

10.3.1 "IP connections" system section

In the automation system PSS 4000, external communication via Raw TCP is possible with all PSS 4000 devices that have the "IP connections" system section. In this case, the "IP connections" system section of a PSS 4000 device performs all the tasks required for communication. These include:

- ▶ Management of the Raw TCP connections
(see [Raw TCP connections](#)  226)
- ▶ Data exchange via Raw TCP
(see [Data exchange via Raw TCP](#)  227)

10.3.2 Raw TCP connections

The "IP connections" system section of a PSS 4000 device manages the Raw TCP connections that are configured for the device. The connection is configured in the PAS4000's IP Connections Editor.

The following Raw TCP connections can be configured for a PSS 4000 device:

- ▶ Send connections
- ▶ Receive connections

A FIFO memory can be enabled for receive data.

- ▶ Send and receive connections

A FIFO memory can be enabled for receive data.

Each Raw TCP connection is configured as to whether it assumes the role of Client or Server when the connection is established.

10.3.3 Data exchange via Raw TCP

Raw TCP is connection-oriented, i.e. before usable data is transferred via Raw TCP, a connection must first be established between two Raw TCP communication partners. The initiator of the connection is called the "Client". The communications partner with which the Client establishes the connection is called the "Server". The connection is configured in the PAS4000's IP Connections Editor.

If a Client role is configured for a connection in the IP Connections Editor, this is called a "Client connection" from the perspective of the PSS 4000 device, otherwise it is called a "Server connection" from the perspective of the PSS 4000 device. So the Server/Client role only applies for that specific connection and is only significant as communication is being established.

Multiple Raw TCP connections can be configured for a PSS 4000 device (see [Data transfer limits](#) [237]).

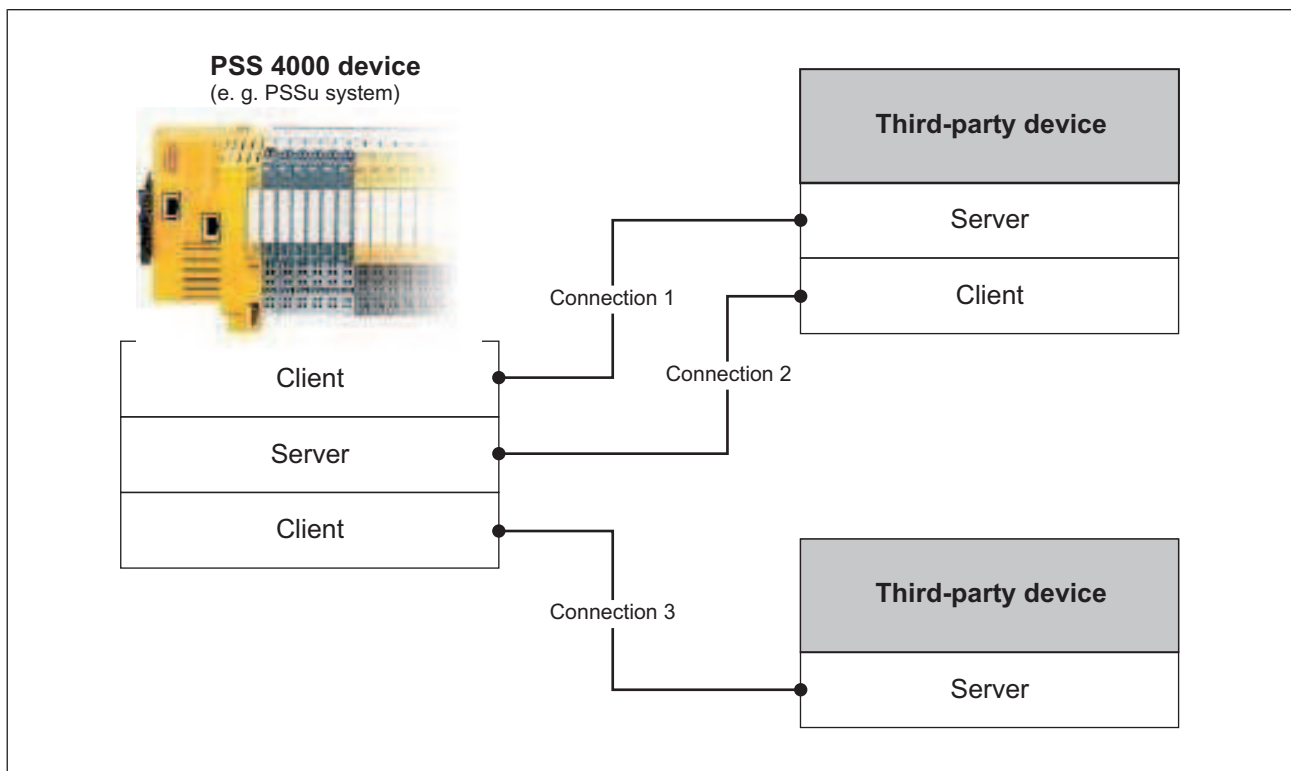


Fig.: Role assignment on Raw TCP connections (example)

Send and receive data is transferred cyclically on PSS 4000 devices. With cyclical transfer, a permanent connection is maintained between the connection Client and Server.

From the perspective of the automation system PSS 4000, receive data is called external I-data and send data is called external O-data.

10.3.4 FIFO memory for receive data

A FIFO memory can be activated for each Raw TCP connection with receive data during the configuration in PAS4000.

The size of the FIFO memory depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.

10.3.5 Processing receive data with Raw TCP

By its very design, TCP is intended for processing a continuous data stream. The data stream is divided into segments merely for transmission reasons.

The use of TCP to transmit cyclical process data initially conflicts with the design of TCP, as process data is typically sent continuously and cyclically in data packets of the same size.

Despite the original design of TCP, Raw TCP in the automation system PSS 4000 was designed so that process data could first and foremost be received and processed cyclically.

However, on PSS 4000 the receive behaviour of Raw TCP can be adapted to suit various application scenarios. This can be achieved through configuration in PAS4000.

Configuration options:

- ▶ Enable/disable FIFO memory
- ▶ Receive data may not exceed/fall below the configured data length

The table below provides an overview of how to sensibly combine the configuration options:

Configuration	Do not accept shorter data length	Accept shorter data length
Enable FIFO memory	Suitability for the scenario: "Transmit process data" (default setting) (see Typical application scenarios for Raw TCP [📖 230])	Suitability for the scenario: "Classic TCP data transmission", e.g. request-response protocols with variable data length (see Typical application scenarios for Raw TCP [📖 230])
Disable FIFO memory	Suitability for the scenario: "Transmit process data" when connection conditions are problematic, for example in the event of transmitter overclocking and frequent connection failure (see Typical application scenarios for Raw TCP [📖 230])	– No sensible scenario –

Information about the basic receive behaviour is available under [Basic workflow when processing receive data \[📖 232\]](#). Further information about the influence of the FIFO memory is available under [Receive behaviour when FIFO memory is enabled \[📖 233\]](#) and [Receive behaviour when FIFO memory is disabled \[📖 235\]](#).

10.3.5.1 Typical application scenarios for Raw TCP

Raw TCP can be configured in PAS4000 so that a variety of typical application scenarios are covered.

Scenario: "Transmit process data"

Aim of this type of data transmission:

The TCP connection receiver only extracts and processes the current receive data from the buffer if it corresponds to the correct data length.

Configuration for the receiver:

► Enable FIFO memory (default setting)

Under certain conditions it may be helpful to disable the FIFO memory. These include problematic connection conditions, which can occur in the event of transmitter overclocking and frequent connection failure for example. For more information on this subject please refer to "Behaviour of the receiver".

► Do **not** accept shorter data length

Expectations of receiver:

In this configuration the TCP connection receiver expects data packets that will always be of the same data length. To do this it is necessary to ensure that the transmitter application sends constant data units; i.e. the transmitter must send cyclical process data of the same data length. The data length must correspond to the receive data length that was configured for the receiver in PAS4000.

Behaviour of the receiver:

In each cycle (t_{ExtCo}), the TCP connection receiver extracts process data with the configured receive data length from the buffer.

► Behaviour when the data length is shorter:

If for some reason the receiver obtains data units with process data of a shorter data length, this will be regarded as an error. The connection will be terminated and re-established.

► Behaviour in the event of transmitter overclocking and frequent connection failure

In the event of transmitter overclocking, the transmitter's send rate is higher than the rate at which the receiver can accept process data. In the event of a frequent connection failure, the connection will need to be re-established each time.

In the default setting (FIFO memory enabled), both transmitter overclocking and frequent connection failure will lead to an increasing accumulation of old data in the buffer. In accordance with FIFO behaviour, data is extracted in the order in which it was received; i.e. old data will need to be processed before any new data that arrives.

In this situation it may be helpful to disable the FIFO memory. If FIFO mode is not configured, the process data from the most recent data packet is always used.

Scenario: "Classic TCP data transmission"

Aim of this type of data transmission:

The TCP connection receiver is supposed to extract and process all data from the buffer sequentially. The data length of the data packets in the buffer may be variable.

A special feature on the PSS 4000 is that the data extracted from the buffer always corresponds to the configured receive data length. So for this scenario the user will need to take additional measures in order to identify the data units. For example these could include protocol-based measures in the PSS 4000 user program, which can be used to control data transmission via request and response. Also, for example, the number of payload bytes transmitted can be sent with each data packet and evaluated accordingly in the receiver.

Configuration for the receiver:

- ▶ Enable FIFO memory
- ▶ Accept shorter data length

Expectations of receiver:

In this configuration the TCP connection receiver expects the transmitter to send data in accordance with the protocol-based measures defined in the application. The data units may have a different data length. So the receiver has no expectation with regard to the data length of a data packet.

Behaviour of the receiver:

In each cycle (t_{ExtCo}), the TCP connection receiver extracts data with the configured receive data length from the buffer. Data is extracted in the order in which it was received; i.e. starting with the oldest data. If the buffer contains less data than the configured receive data length, the remaining data will be filled with ZERO, up to the configured receive data length.

10.3.5.2 Basic workflow when processing receive data

TCP data is received asynchronously and stored in a buffer. The same principle is always applied when data is extracted from the buffer. In each cycle (t_{ExtCo}), the TCP connection receiver extracts the exact amount of data that was configured as the receive data length for the connection; i.e. in each cycle (t_{ExtCo}), data with a fixed data length is extracted from the buffer and transferred to the PSS 4000 array that was configured for the receive connection (see [Raw TCP I/O data](#) [238]). This data extraction procedure is used whether or not FIFO mode is configured. This procedure also applies for extracting data from a buffer containing data packets with a fixed data length as well as a buffer containing data packets with a variable data length.

Contents of the buffer when receiving data packets of an identical data length

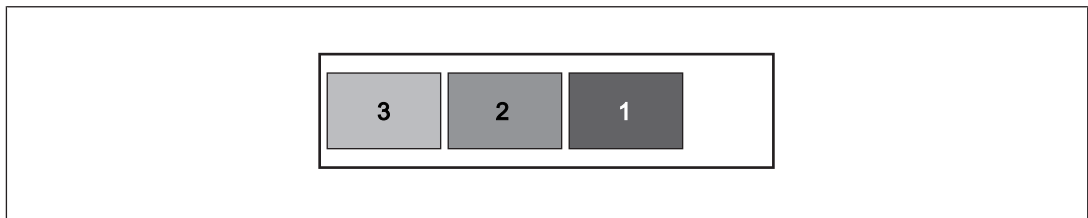


Fig.: Buffer with data packets of an identical data length

Contents of the buffer when receiving data packets of a variable data length:

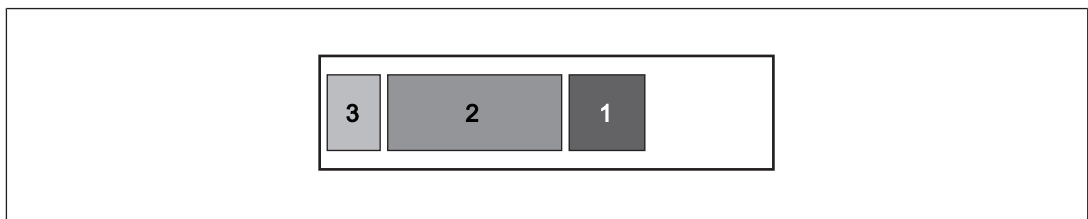


Fig.: Buffer with data packets of a variable data length

10.3.5.3 Receive behaviour when FIFO memory is enabled

This behaviour can be called "original TCP behaviour". The aim of this procedure is to prevent old data being rejected when new data arrives, even when it has not yet been processed.

With each cycle (t_{ExtCo}), the TCP connection receiver extracts data packets with the configured receive data length from the buffer and transfers them to the PSS 4000 array that was configured for the receive connection. In accordance with FIFO behaviour, data is extracted in the order in which it was received; i.e. when data is subsequently processed on the PSS 4000 device, processing starts with the data in the oldest data packet.

Should this procedure overload the receiver, in some circumstances the receiver may no longer be able to accept new data. Should this be the case, TCP has its own mechanisms for informing the transmitter that the receiver is overloaded. At this point the transmitter stops transmitting new data until the receiver signals its ability to receive again via the TCP mechanism.

The receive behaviour when the FIFO memory is enabled is an acceptable behaviour and leads to sensible results provided the TCP connection transmitter sends data units with a fixed data length. In this case the data packets in the buffer have an identical data length. If the transmitter sends data units with a variable data length, then "Accept shorter data length" should be configured for the receiver. In this case the buffer may potentially contain data packets with a different data length. However, as data is always extracted with the configured receive data length, data may potentially be extracted from several data packets. In this case additional application-related measures are required on the transmitter and receiver in order to achieve sensible results. If the buffer contains less data than the configured receive data length, with a variable data length the remaining data will be filled with ZERO, up to the configured receive data length.

Example of accepted behaviour with fixed data length:

The example below illustrates how data is extracted with the configured receive data length from a buffer containing data packets with identical or fixed data length.

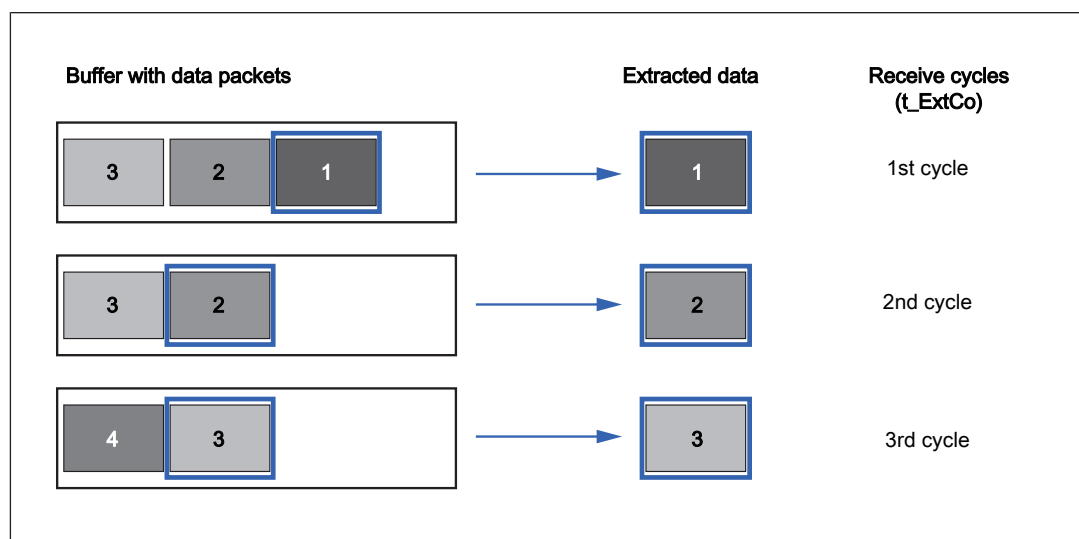


Fig.: FIFO behaviour used to extract data on data packets with fixed data length (example)

Example of behaviour with variable data length:

The example below illustrates how data is extracted with the configured receive data length from a buffer containing data packets with variable data length.

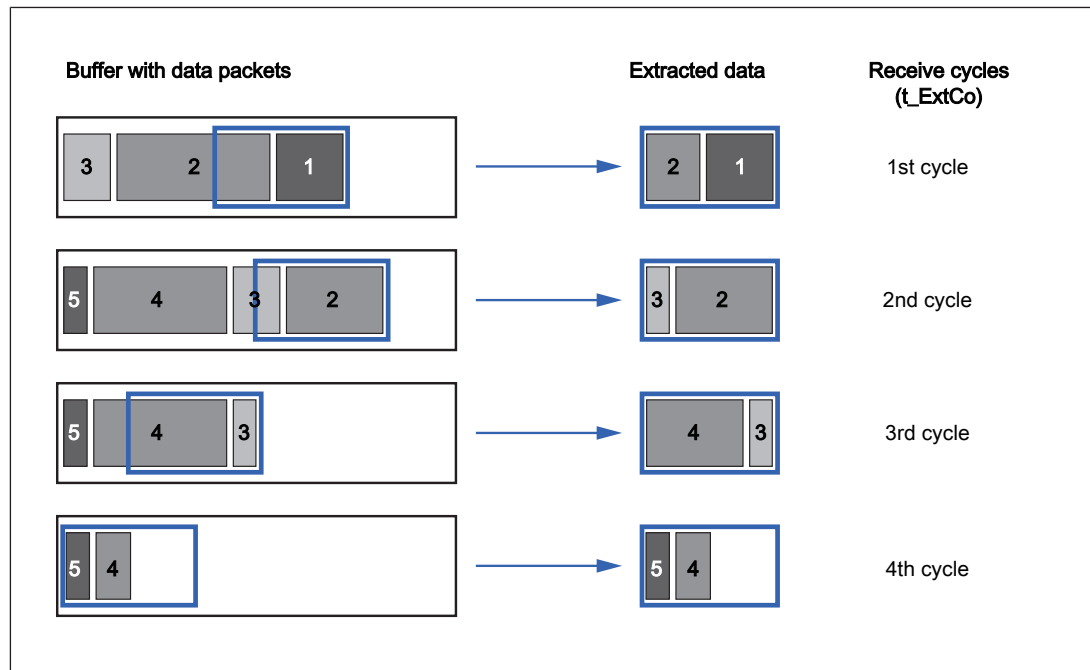


Fig.: FIFO behaviour used to extract data on data packets with variable data length (example)

Please note:

In this example, additional measures must be taken in the application to ensure that data that belongs together can be identified or reconstructed once the data has been extracted.

10.3.5.4 Receive behaviour when FIFO memory is disabled

This behaviour can be called "process data behaviour". The aim of this procedure is to process only the most recent receive data. For this purpose, in each cycle (t_{ExtCo}), the content of the buffer is examined and rejected as many times as necessary until there is no more receive data to be found. The data length during the examination corresponds to the configured receive data length. The data packet with the most recent receive data is extracted and is transferred to the array that was configured for the receive connection for subsequent processing.

The receive behaviour when the FIFO memory is disabled is an acceptable behaviour and leads to sensible results provided the TCP connection transmitter sends data units with a fixed data length. In this case the data packets in the buffer have an identical data length. If the transmitter sends data units with a variable data length, then "Accept shorter data length" should be configured for the receiver. In this case the buffer may potentially contain data packets with a different data length. However, as data is always examined with the configured receive data length, data is rejected without any consideration as to where the data belongs. If the buffer contains less data than the configured receive data length, with a variable data length the remaining data will be filled with ZERO, up to the configured receive data length. We are not aware of any sensible application for this scenario.

Example of accepted behaviour with fixed data length:

The example below illustrates how data is extracted with the configured receive data length from a buffer containing data packets with identical or fixed data length.

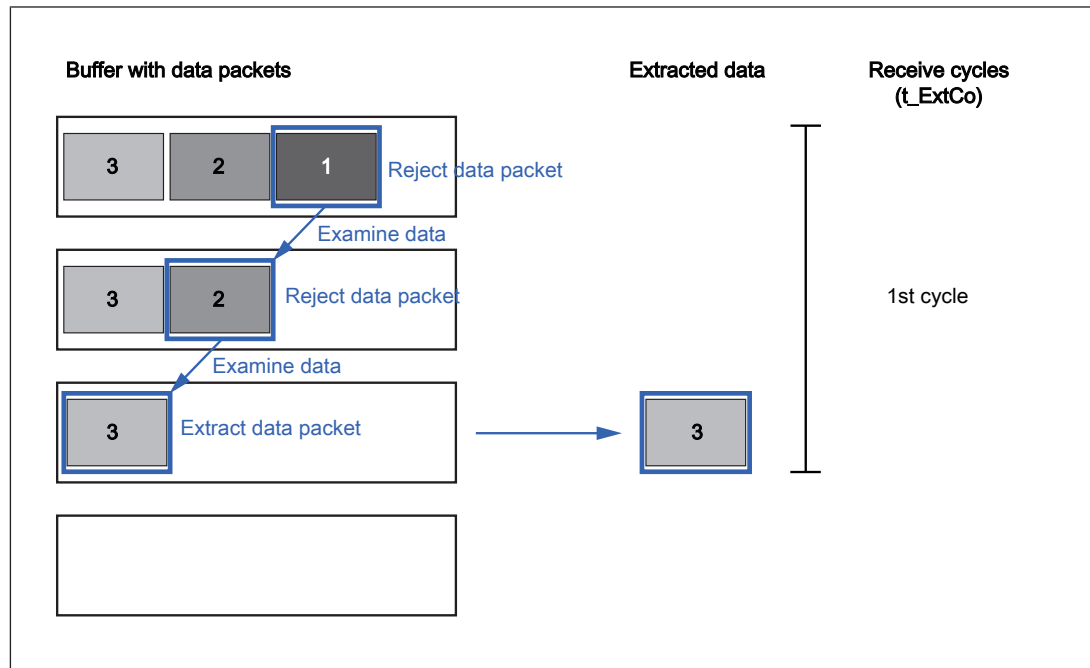


Fig.: FIFO behaviour not used to extract data on data packets with fixed data length (example)

Example of behaviour with variable data length:

The example below illustrates how data is extracted with the configured receive data length from a buffer containing data packets of variable data length.

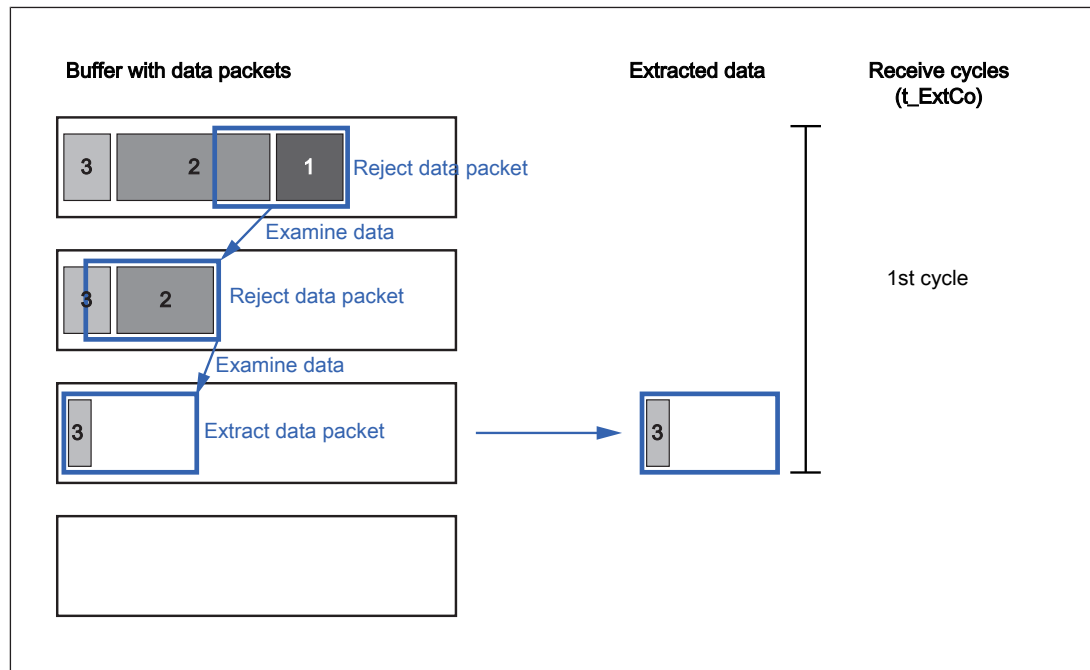


Fig.: FIFO behaviour not used to extract data on data packets with variable data length (example)

Please note that this example is **not** a sensible scenario.

10.3.6 Data transfer limits

When transmitting data via Raw TCP, the following limits apply for a PSS 4000 device:

- ▶ The max. number of Raw TCP connections depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device. Please also note the general upper limit of 48 IP connections for a PSS 4000 device.



NOTICE

Maximum number of IP connections

The "IP connections" system section can manage a maximum of 48 IP connections in total. This upper limit also applies if you are operating the "IP connections" system section with a variety of protocols or communication links.

SafetyNET p connections are **not** managed by the "IP connections" system section and so do not affect this upper limit.

- ▶ From the perspective of the PSS 4000 device, if one Server connection is configured, this Server may be used to connect a single Client. If communication is required with multiple Clients, from the perspective of the PSS 4000 device, a separate Server connection must be configured for each Client.
- ▶ A data length of 1 ... max. 1024 Bytes (external I-data) can be configured for a receive connection.
- ▶ A data length of 1 ... max. 1024 Bytes (external O-data) can be configured for a send connection.
- ▶ A data length of 1 ... max. 1024 Bytes for each transmission direction can be configured for a send and receive connection.
- ▶ With the default configuration (PAS4000), data packets must not be shorter than the configured data length. TCP segments that are shorter than the configured receive data length will be rejected and a diagnostic message will be displayed in the diagnostic list. If "Accept shorter data length when receiving" was configured in PAS4000, data packets with variable data length are accepted; i.e. TCP segments that are shorter than the configured receive data length are accepted.
- ▶ The size of the FIFO memory for a connection's receive data depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.

10.3.7 Raw TCP I/O data

The "IP connections" system section enables a PSS 4000 device to exchange I/O data with third-party devices via Raw TCP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding Raw TCP data areas.

I/O mapping of Raw TCP data areas

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O mapping for Raw TCP data areas:

I/O mapping		on Raw TCP	
		Receive data (= external I-data)	Send data (= external O-data)
of module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
of PI- variable	I-PI variable	◆	---
	O-PI variable	---	◆

- ◆ I/O mapping possible
 --- I/O mapping impossible

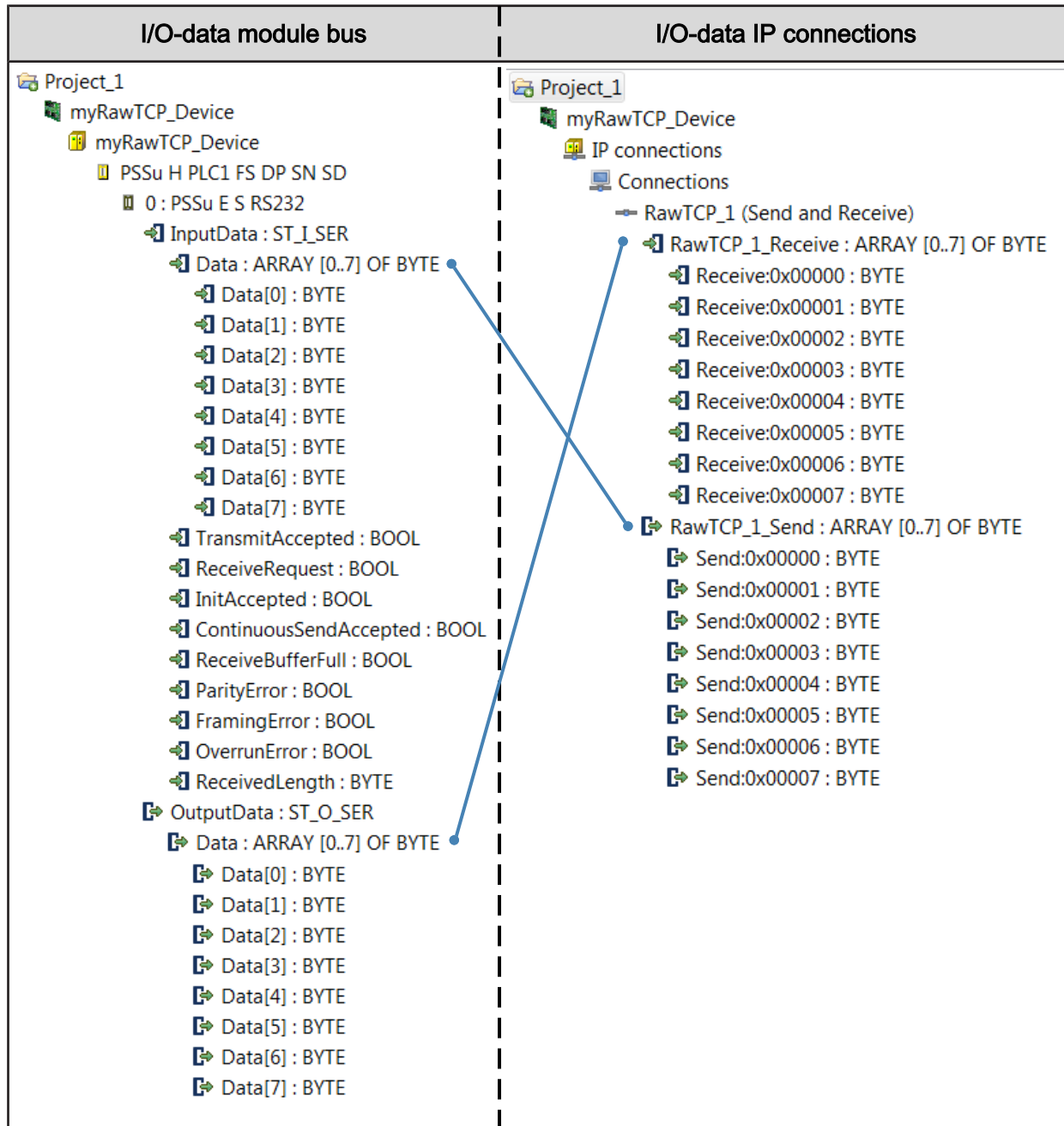
Examples

► I/O mapping of I/O data

The receive data of the electronic module PSSu K S RS232 is to be transmitted to the communication partner and data received by the communication partner is to be output on the PSSu K S RS232.

The send and receive connection "RawTCP_1" is configured for this purpose:

Possible I/O mapping (example):

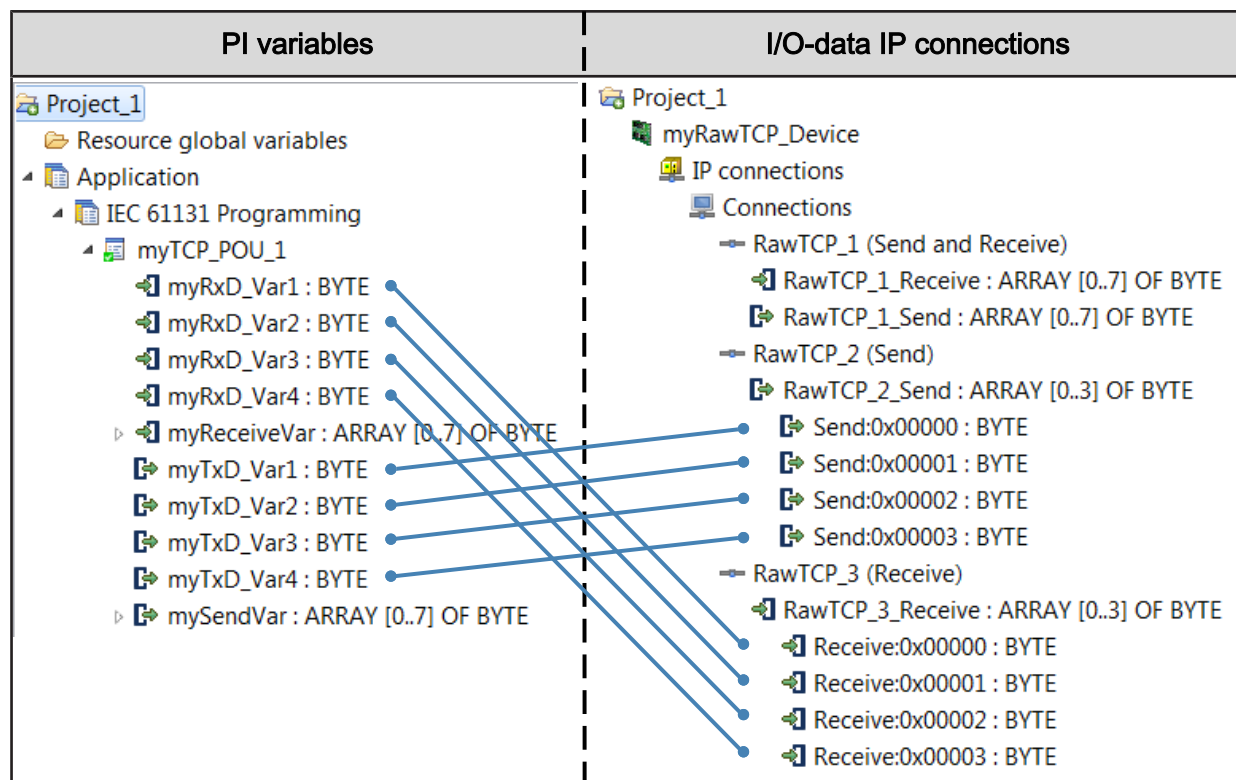


► I/O mapping of PI variables

The contents of a block's O-PI variables are to be transmitted to the communication partner and I-PI variables are to be assigned to the receive data. 2 connections are configured for this purpose:

- RawTCP_2 (send connection)
- RawTCP_3 (receive connection)

Possible I/O mapping (example):



— I/O mapping

10.3.8 Address format of Raw TCP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

Address format of send data

The addresses of Raw TCP send data are structured as follows:

<Device name>.TCP.<Connection name>.Send.Byte[<Element number>]

Example:

myDevice_1.TCP.myTxConnection_1.Send.Byte[0]

...

myDevice_1.TCP.myTxConnection_1.Send.Byte[1023]

Address format of receive data

The addresses of Raw UDP receive data are structured as follows:

<Device name>.TCP.<Connection name>.Receive.Byte.[<Element number>]

Example:

myDevice_1.TCP.myRxConnection_1.Receive.Byte[0]

...

myDevice_1.TCP.myRxConnection_1.Receive.Byte[1023]

10.3.9 Data consistency on Raw TCP

Where receive data is sent by third-party devices, these devices are responsible for the data consistency of this data. From the PSS 4000's perspective, no statement can be made regarding the data consistency of those devices.

Data consistency on receive data of a Raw TCP connection

If a PSS 4000-exclusive/operable device is the recipient of data from a Raw TCP connection, from the perspective of the PSS 4000 device the receive data is external I-data. External I-data from a Raw TCP connection is treated as consistent data in PSS 4000 (see [Data consistency](#)  377).

External I-data from a Raw TCP connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a Raw TCP connection" (see [Consistency ranges](#)  378).

Data consistency on send data of a Raw TCP connection

If a PSS 4000-exclusive/operable device in a Raw TCP connection is the transmitter of data, from the perspective of the device the send data is external O-data.

External O-data of a Raw TCP connection is consistent if it comes from the same consistency range on a data source (see [Data consistency](#)  377).

10.4 Raw UDP

The User Datagram Protocol (UDP) is implemented in accordance with RFC768. "Raw UDP" stands for data exchange via UDP, without an upper layer protocol.

UDP is used for communication between various communication partners. The communication service on a communication partner is identified via a port number. UDP uses the Internet Protocol (IP) for addressing, i.e. a UDP datagram is encapsulated in an IP frame and is sent to the communication partner as a data packet using the IP address.

UDP is a "non-connection-oriented" or "connectionless" protocol. This means:

- ▶ A connection to the communication partner is not established before starting transmission
- ▶ There is no guarantee that
 - Sent data packets arrive
 - Sent data packets arrive only once
 - Data packets arrive in the same order in which they were sent

As a result, UDP can only be used if the application is insensitive to lost and unsorted datagrams or if the application itself provides appropriate control and correction mechanisms on the communication partners.

Although UDP is a connectionless protocol, for simplicity's sake, the description below refers to the communication relationship between communication partners as a "connection".

Data exchange via Raw UDP can be used on the automation system PSS 4000 for non-safety-related applications.

10.4.1 "IP connections" system section

In the automation system PSS 4000, external communication via Raw UDP is possible with all PSS 4000 devices that have the "IP connections" system section. In this case, the "IP connections" system section of a PSS 4000 device performs all the tasks required for communication. These include:

- ▶ Managing the Raw UDP connections
(see [Raw UDP connections](#)  245)
- ▶ Data exchange
(see [Data exchange via Raw UDP](#)  246)

10.4.2 Raw UDP connections

The "IP connections" system section of a PSS 4000 device manages the Raw UDP connections that are configured for the device. The connection is configured in the PAS4000's IP Connections Editor.

The following Raw UDP connections can be configured for a PSS 4000 device:

► **Send connections (role = transmitter)**

- Unicast (1:1 connection)
- Broadcast (1:"all" connection)
Loopback is configurable
- Multicast (1:n connection) in accordance with RFC 1112, IGMVP V1

► **Receive connections (role = receiver)**

A FIFO memory can be enabled for receive data

- Unicast (1:1 connection)
- Broadcast (1:"all" connection)
- Multicast (1:n connection) in accordance with RFC 1112, IGMVP V1

► **Send and receive connections (role = transmitter/receiver)**

A FIFO memory can be enabled for receive data

- Unicast (1:1 connection)
- Broadcast (1:"all" connection)
Loopback when sending is configurable
- Multicast (1:n connection) in accordance with RFC 1112, IGMVP V1

10.4.3 Data exchange via Raw UDP

Raw UDP is not connection-oriented; i.e. Raw UDP does not have any mechanisms for establishing and ending a connection.

With connections that are configured in PAS4000 as send connections (role = transmitter), the PSS 4000 device sends data to one or more communication partners with a receiver role for this connection.

With connections that are configured in PAS4000 as receive connections (role = receiver), the connection waits for the communication partner with a transmitter role for this connection to send its data.

Connections that are configured in PAS4000 as send and receive connections (role = transmitter/receiver) are used to both send and receive data. A communication partner that opens communication, i.e. sends the first telegram via this connection, must be configured for send and receive connections. This communication partner is called the initiator. The communication partner without initiator functionality waits to receive the first telegram via this connection.

From the perspective of the automation system PSS 4000, receive data is called external I-data, and send data is called external O-data (see [Validity of process data](#)  359).

A send and receive connection with initiator functionality is configured in PAS4000 according to the following rule of thumb:

Role	Network settings	
Transmitter/receiver	Remote port number	Does not equal 0
	Remote IP address	Does not equal 0.0.0.0



INFORMATION

The initiator of a send and receive connection must meet the condition for both the remote port number and the remote IP address.

If just one of the conditions is **not** met, i.e. the remote port number is 0 and/or the remote IP address is 0.0.0.0, the send and receive connection will not have initiator functionality.

10.4.4 FIFO memory for receive data

A FIFO memory can be activated for each raw UDP connection with receive data during the configuration in PAS4000.

If the FIFO memory is enabled for a raw UDP connection, all incoming data will be collected in the FIFO memory and will be processed in the order in which it arrived. This procedure prevents the situation in which "old" data is rejected when "new" data arrives, even if it has not yet been processed.

The size of the FIFO memory depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device. If the memory overflows, incoming data will be rejected and a diagnostic message will be displayed in the diagnostic list.

10.4.5 Data transfer limits

When transferring data via Raw UDP, the following limits apply for a PSS 4000 device:

- ▶ The max. number of Raw UDP connections depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.
Please also note the general upper limit of 48 IP connections for a PSS 4000 device.



NOTICE

Maximum number of IP connections

The "IP connections" system section can manage a maximum of 48 IP connections in total. This upper limit also applies if you are operating the "IP connections" system section with a variety of protocols or communication links.

SafetyNET p connections are **not** managed by the "IP connections" system section and so do not affect this upper limit.

- ▶ A data length of 1 ... max. 1024 Bytes (external I-data) can be configured for a receive connection.
- ▶ A data length of 1 ... max. 1024 Bytes (external O-data) can be configured for a send connection.
- ▶ A data length of 1 ... max. 1024 Bytes for each transmission direction can be configured for a send and receive connection.
- ▶ The maximum configurable data length is supported for each telegram; i.e. data from send connections will not be spread over multiple telegrams.
- ▶ The number of data bytes transmitted in a telegram must exactly equal the number that was configured for the connection:
 - If a telegram received by a PSS 4000 device contains more data bytes than were configured for the connection, the surplus data bytes will be rejected. A diagnostic message will not be displayed in the diagnostic list, as this is the defined behaviour for the PSS 4000.
 - If a telegram received by a PSS 4000 device contains fewer data bytes than were configured for the connection, a diagnostic message will be displayed in the diagnostic list.
- ▶ The size of the FIFO memory for a connection's receive data depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.

10.4.6 Raw UDP I/O data

The "IP connections" system section enables a PSS 4000 device to exchange I/O data with third-party devices via Raw UDP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding Raw UDP data areas.

I/O mapping of Raw UDP data areas

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O mapping for Raw UDP data areas:

I/O mapping		on Raw UDP	
		Receive data (= external I-data)	Send data (= external O-data)
of module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
of PI- variable	I-PI variable	◆	---
	O-PI variable	---	◆

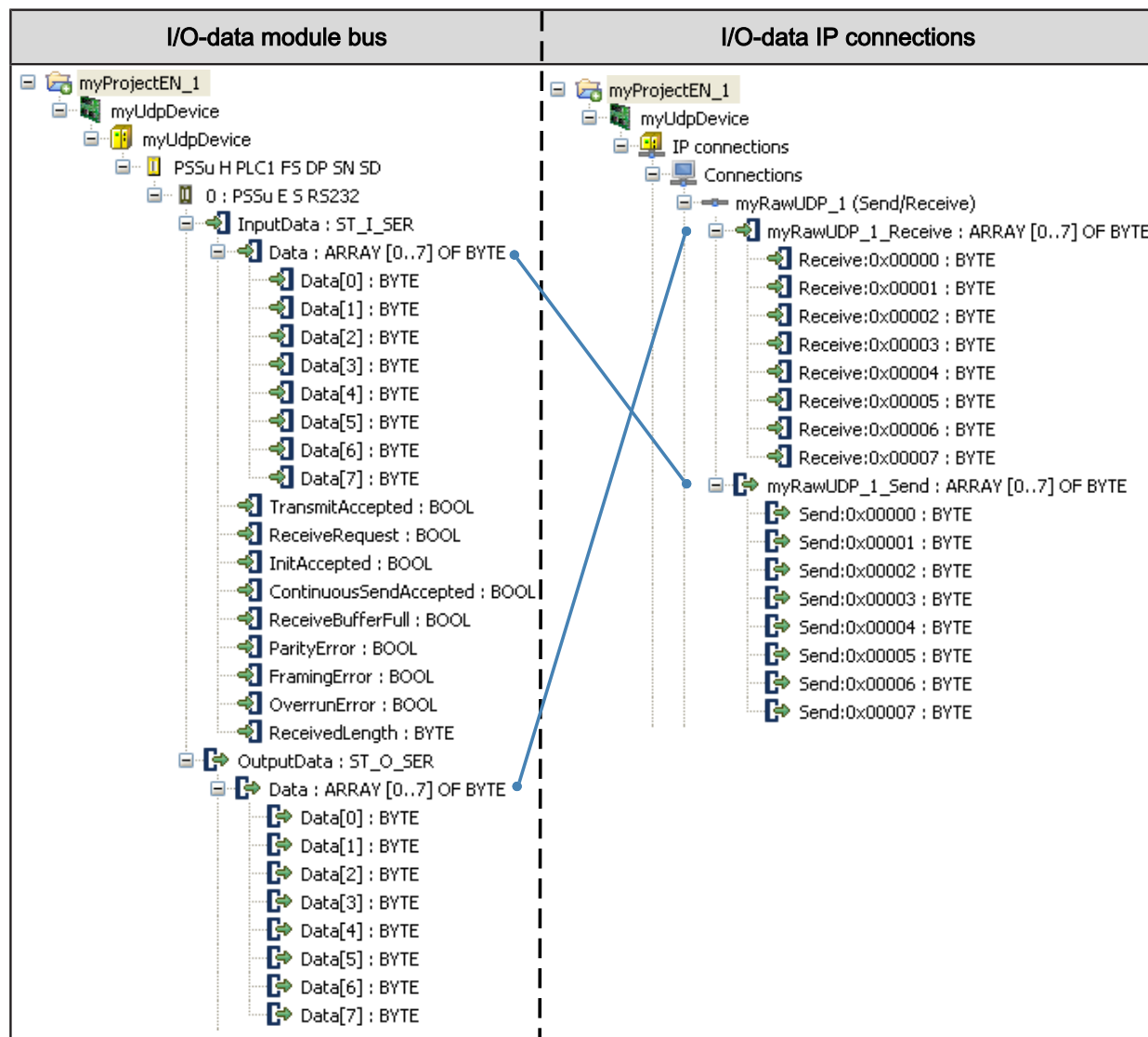
- ◆ I/O mapping possible
 --- I/O mapping impossible

Examples

► I/O mapping of I/O data

The receive data of the electronic module PSSu K S RS232 is to be transmitted to the communication partner and data received by the communication partner is to be output on the PSSu K S RS232. The send and receive connection "myRawUDP_1" is configured for this purpose:

Possible I/O mapping (example):



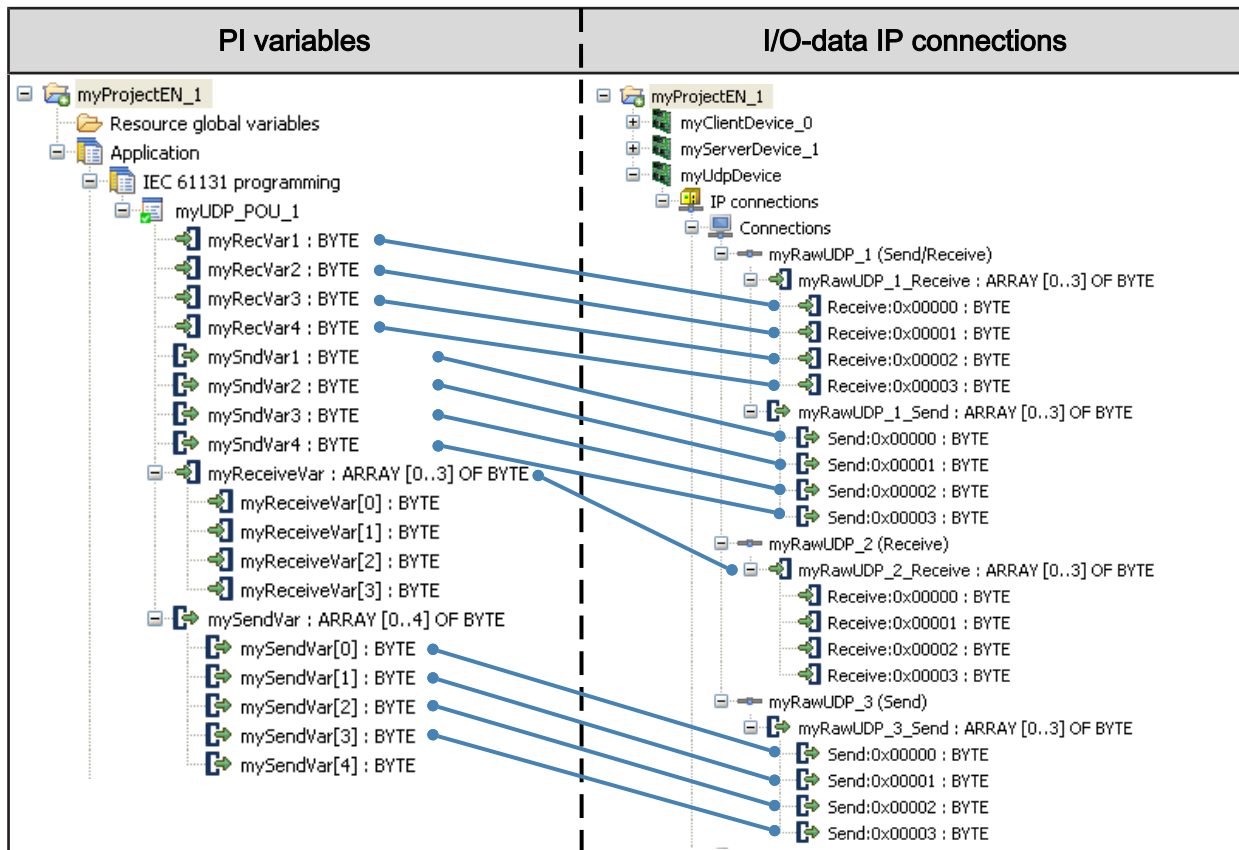
— I/O mapping

► I/O mapping of PI variables

The contents of a block's O-PI variables are to be transmitted to the communication partner and I-PI variables are to be assigned to the receive data. 3 connections are configured for this purpose:

- myRawUDP_1 (send and receive connection)
- myRawUDP_2 (send connection)
- myRawUDP_3 (receive connection)

Possible I/O mapping (example):



—●— I/O mapping

10.4.7 Address format of Raw UDP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

10.4.7.1 Address format of send data

The addresses of Raw UDP send data are structured as follows:

<Device name>.UDP.<Connection name>.Send.Byte[<Element number>]

Example:

myDevice_1.UDP.myTxConnection_1.Send.Byte[0]

...

myDevice_1.UDP.myTxConnection_1.Send.Byte[1023]

10.4.7.2 Address format of receive data

The addresses of Raw UDP receive data are structured as follows:

<Device name>.UDP.<Connection name>.Receive.Byte.[<Element number>]

Example:

myDevice_1.UDP.myRxConnection_1.Receive.Byte[0]

...

myDevice_1.UDP.myRxConnection_1.Receive.Byte[1023]

10.4.8 Data consistency on Raw UDP

Where receive data is sent by third-party devices, these devices are responsible for the data consistency of this data. From the PSS 4000's perspective, no statement can be made regarding the data consistency of those devices.

Data consistency on receive data of a Raw UDP connection

If a PSS 4000-exclusive/operable device is the recipient of data from a Raw UDP connection, from the perspective of the PSS 4000 device the receive data is external I-data.

External I-data from a Raw UDP connection is treated as consistent data in PSS 4000 (see [Data consistency](#)  377]).

The external I-data from a Raw UDP connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a Raw UDP connection" (see [Consistency ranges](#)  378]).

Data consistency on send data of a Raw UDP connection

If a PSS 4000-exclusive/operable device in a Raw UDP connection is the transmitter of data, from the perspective of the PSS 4000 device the send data is external O-data.

The external O-data of a Raw UDP connection is consistent if it comes from the same consistency range on a data source (see [Data consistency](#)  377]).

10.5 UDP communication using blocks

The User Datagram Protocol (UDP) is implemented in accordance with RFC768.

UDP is used for communication between various communication partners. The communication service on a communication partner is identified via a port number. UDP uses the Internet Protocol (IP) for addressing, i.e. a UDP datagram is encapsulated in an IP frame and is sent to the IP address of the communication partner as a data packet.

Three different IP interfaces can be used to send a UDP datagram, depending on the controller you are using:

► IP interface with the interface identifier 0

IP interface that was configured in the naming data of the PSS 4000 device. This interface is used as the default interface during UDP communication using blocks.

► IP interface with the interface identifier 1 and 2

IP interfaces that the user can create in the user program or with the network configuration file (extended network configuration).

Please note that these IP interfaces can only be used on PLC2 controllers (1st generation) and 2nd generation controllers.



NOTICE

IP interfaces with the interface identifier 1 and 2 can only be used on PLC2 controllers (1st generation) and 2nd generation controllers.

UDP is a "non-connection-oriented" or "connectionless" protocol. This means:

- A connection to the communication partner is not established before starting transmission
- There is no guarantee that
 - Sent data packets arrive
 - Sent data packets arrive only once
 - Data packets arrive in the same order in which they were sent

As a result, UDP can only be used if the application is insensitive to lost and unsorted datagrams or if the application itself provides appropriate control and correction mechanisms on the communication partners.

It makes sense to use UDP communication using blocks when data transfer is to be controlled through the PSS 4000 user program. It makes more sense to use Raw UDP for cyclical data transfer via UDP.

To send/receive a datagram via "UDP using blocks" on the automation system PSS 4000, the dedicated PSS 4000 system control blocks must be used in the user program:

► pssCreateIPInterface - Create IP interface

Only required when data is to be sent via the IP interface with the interface identifier 1 or 2 and the IP interface is to be created in the block.

► pssReleaseIPInterface - Release IP interface

Only required if an interface that was created using the block pssCreateIPInterface is to be released.

► pssUseETHInterface - Use ETH interface

Only required when data is to be sent via the IP interface with the interface identifier 1 or 2 and the IP interface is to be created using the network configuration file.

- ▶ pssReleaseETHInterface - Release ETH interface

Only required if an interface that was selected using the block pssUseETHInterface is to be released.

- ▶ pssCreateUDPSocket - Create UDP Socket

- ▶ pssCreateUDPSocketOnInterface - Create UDP Socket on IP interface

Only required when data is to be sent via the IP interface with the interface identifier 1 or 2.

- ▶ pssReleaseUDPSocket - Release UDP Socket

- ▶ pssUDPSend - Send UDP datagram

- ▶ pssUDPReceive - Receive UDP datagram

The blocks may be used on either the ST or FS resource.

UDP communication using blocks can be used on the automation system PSS 4000 for **non**-safety-related applications. Additional security measures are required to enable UDP communication using blocks to be used for safety-related applications. These security measures must be implemented through the application. The user is responsible for the correct implementation.

10.5.1 "IP connections" system section

In the automation system PSS 4000, external communication via UDP using blocks is possible with all PSS 4000 devices that have the "IP connections" system section. In this case, the "IP connections" system section of a PSS 4000 device performs all the tasks required for communication. These include:

- ▶ Management of UDP sockets
- ▶ Data exchange via "UDP using blocks"

10.5.2 IP interfaces for UDP communication

Three different IP interfaces can be used to send a UDP datagram, depending on the controller you are using:

- ▶ IP interface with the interface identifier 0

IP interface that was configured in the naming data of the PSS 4000 device. This interface is used as the default interface during UDP communication using blocks.

- ▶ IP interface with the interface identifier 1 and 2

IP interfaces that the user can create in the user program or with the network configuration file (extended network configuration).

Please note that these IP interfaces can only be used on PLC2 controllers (1st generation) and 2nd generation controllers.



NOTICE

IP interfaces with the interface identifier 1 and 2 can only be used on PLC2 controllers (1st generation) and 2nd generation controllers.

Using the IP interfaces it is possible to communicate with a maximum of 3 different networks simultaneously. In the blocks, interface identifiers help with easier referencing of the interfaces.

The default IP interface (IP interface with the interface identifier 0) is always available on a PSS 4000 device. There are two procedures for creating the IP interfaces with the interface identifier 1 and 2:

- ▶ In the user program

The IP interfaces with the interface identifier 1 and 2 can be created using the block `pssCreateIPInterface`. The IP interfaces that are created can only be used for UDP communication using blocks. If the IP interfaces with the interface identifier 1 and 2 are no longer needed, they can be released using the block `pssReleaseIPInterface`. After being released, the interfaces can be recreated with the same or an amended configuration.

- ▶ With the network configuration file

Up to 15 ETH interfaces can be created using the network configuration file, see [Extended network configuration \[137\]](#). The block `pssUseETHInterface` is used to select which of these ETH interfaces is to be used as the IP interface with the interface identifier 1 or 2. If the IP interfaces with the interface identifier 1 and 2 are no longer needed, they can be released using the block `pssReleaseETHInterface`.

You can mix both procedures for creating the IP interfaces, i.e. the IP interface with interface identifier 1 can be created with one procedure and IP interface with interface identifier 2 with the other.

10.5.3 UDP sockets

Before a PSS 4000 device can exchange data via UDP using the dedicated PSS 4000 system control blocks, its IP connections system section must have at least one open UDP Socket.

To create a UDP Socket, one of the following PSS 4000 system control blocks must be executed:

► **pssCreateUDPSocket**

Can be used if the UDP Socket is to be created on the default IP interface.

► **pssCreateUDPSocketOnInterface**

Must be used if the UDP Socket is to be created on the IP interface with interface identifier 1 or 2. With this block, UDP Sockets can also be created on the default IP interface (with interface identifier 0).

Datagrams can then be exchanged with other PSS 4000 system control blocks via the UDP Socket. Both unidirectional UDP communication (send only or receive only) and bidirectional UDP communication (send and receive) are possible via a UDP Socket.

UDP Sockets are managed by the IP connections system section of a PSS 4000 device. The system section can manage a maximum of 9 UDP Sockets; i.e. a maximum of 9 UDP Sockets can be open at the same time on a PSS 4000 device through multiple execution of `pssCreateUDPSocket`/`pssCreateUDPSocketOnInterface` (see [Data transfer limits](#)  261).

When a UDP Socket is created, the PSS 4000 device generates a unique reference value for the UDP Socket, called the socket handle. The socket handle that is generated is issued on the block `pssCreateUDPSocket`/`pssCreateUDPSocketOnInterface`. When executing the blocks `pssUDPSend`/`pssUDPReceive` (send/receive datagram) in the user program, the socket handle of the UDP Socket used to send/receive a datagram must be stated. The blocks `pssUDPSend`/`pssUDPReceive` must always be executed in the same task as the block `pssCreateUDPSocket`/`pssCreateUDPSocketOnInterface`.

A UDP Socket can be released again if necessary. This is done by executing the block `pssReleaseUDPSocket` in the user program. When executing `pssReleaseUDPSocket`, the socket handle of the UDP Socket to be released must be stated. `pssReleaseUDPSocket` must be executed in the same task as `pssCreateUDPSocket`/`pssCreateUDPSocketOnInterface`. Once a UDP Socket has been released, no more datagrams may be sent or received using the PSS 4000 system control blocks referenced for this UDP Socket.

10.5.4 Data exchange via "UDP using blocks"

Data exchange via UDP using blocks should be preferred for Unicast communication. Further information is available under [Special features with external communication via UDP](#)  [202]. A 1:1 communication relationship in which data is transferred between a single transmitter and a single receiver is called a Unicast communication.

With PSS 4000, the send data of a UDP datagram is called external O-data and the receive data of a UDP datagram is called external I-data.

► Send datagram with PSS 4000 system control block "pssUDPSend"

When pssUDPSend is executed, a UDP datagram is generated and is stored temporarily in the UDP send buffer associated with the task (see [UDP send buffer and UDP receive buffer](#)  [260]). The IP connections system section ensures that UDP datagrams are sent from the UDP send buffer. A UDP datagram contains the following data:

- Port number of the port on the remote device through which the UDP datagram is received
- IP address of the remote device for which the send data is destined.
- Payload
- Data length of payload

pssUDPSend can be executed several times during a task's execution time. The user program determines when the block is executed; i.e. the block does not need to be executed in each task cycle. pssUDPSend can be assigned different data each time it is executed (e.g. payload length is different, different remote device).

► Receive datagram with PSS 4000 system control block "pssUDPReceive"

The IP connections system section reads the UDP datagrams received via the network and ensures they are stored temporarily in the UDP receive buffer associated with the target task. Each time pssUDPReceive is executed, a UDP datagram is read from the UDP receive buffer (see [UDP send buffer and UDP receive buffer](#)  [260]). The user program determines when the block is executed; i.e. the block does not need to be executed in each task cycle. A UDP datagram contains the following data:

- Port number of the UDP socket on the PSS 4000 device
The port number is used to determine the UDP socket and therefore the target task of a UDP datagram.
- IP address of the remote device from which the receive data comes.
- Payload
- Data length of payload

10.5.5 UDP send buffer and UDP receive buffer

Each task in which PSS 4000 system control blocks are executed for data exchange via UDP has a UDP send buffer and a UDP receive buffer.

UDP send buffer

A task's UDP send buffer is used to temporarily store the UDP datagrams that are generated by executing the block pssUDPSend during a task's execution time. The IP connections system section fetches the UDP datagrams from a task's UDP send buffer and sends them via the network. The required size of the UDP send buffer is calculated automatically from the configured maximum UDP send rate.

Each time pssUDPSend is executed during a task's execution time, a UDP datagram is added to the task's UDP send buffer.

UDP receive buffer

The UDP datagrams received from the IP connections system section via the network are temporarily stored in a task's UDP receive buffer. The required size of the UDP receive buffer is calculated automatically from the configured maximum UDP receive rate.

Each time pssUDPReceive is executed during a task's execution time, a UDP datagram is removed from the task's UDP receive buffer. The UDP datagrams are removed in chronological order, starting with the oldest.

If the UDP datagrams are not retrieved from a task's UDP receive buffer in time, the memory capacity may not be sufficient. In this case, existing UDP datagrams will be replaced so that new UDP datagrams can be included in the UDP receive buffer. UDP datagrams will be replaced in chronological order, starting with the oldest UDP datagram.

10.5.6 Data transfer limits

When transferring data via "UDP using blocks", the following limits apply for a PSS 4000 device:

- ▶ The IP connections system section can manage a maximum of 2 further IP interfaces in addition to the default IP interface. The additional IP interfaces have the interface identifiers 1 and 2.
The IP interfaces with the interface identifiers 1 and 2 can be used in each task and on each resource. Their use does not depend on the task in which the IP interface was created (block `pssCreateIPInterface`) or selected (block `pssUseETHInterface`).
- ▶ If an IP interface with interface identifier 1 and an IP interface with interface identifier 2 already exist and an attempt is made to create another IP interface with interface identifier 1 or 2, then the "IP connections" system section will prevent it. The existing IP interfaces continue to be used unchanged.
- ▶ During a task cycle, one of the two blocks `pssCreateIPInterface` or `pssUseETHInterface` may be executed a maximum of once in a task for interface identifier 1. The same applies for interface identifier 2.
- ▶ During a task cycle, one of the two blocks `pssReleaseIPInterface` or `pssReleaseETHInterface` may be executed a maximum of once in a task for interface identifier 1. The same applies for interface identifier 2.
- ▶ The IP connections system section can manage a maximum of 9 UDP Sockets.
If 9 UDP Sockets have already been generated in the user program and an attempt is made to generate another UDP Socket, then the "IP connections" system section will prevent it; i.e. the UDP Socket will not be generated.
- ▶ The block `pssCreateUDPSocket/pssCreateUDPSocketOnInterface` can be executed several times in the same task. A maximum of 9 UDP Sockets can be generated within a task cycle.
Compliance with this limit is not monitored. If the limit is exceeded, however, the correct function cannot be guaranteed.
- ▶ If the block `pssCreateUDPSocketOnInterface` is executed with the interface identifier 1 or 2, although the IP interface has not yet been created, then the IP connections system section does not create the UDP Socket.
- ▶ A maximum of 9 UDP sockets can be released within a task cycle.
Compliance with this limit is not monitored. If the limit is exceeded, however, the correct function cannot be guaranteed.
- ▶ A maximum data length of 1464 Bytes can be configured for the payload of a UDP datagram.
As the system data type `UDP_DATAGRAM` must be used to send a UDP datagram, the system data type prevents this limit from being exceeded. If a UDP datagram containing more than the max. permitted number of payload bytes is received, the datagram will be rejected.
- ▶ A maximum of 50 UDP datagrams can be sent within a task cycle. The UDP datagrams may have a different payload length.
Compliance with the limit of maximum 50 UDP datagrams is not monitored. If the limit is exceeded, however, the correct function cannot be guaranteed.

- ▶ A maximum of 50 UDP datagrams can be received within a task cycle. The UDP datagrams may have a different payload length.
Compliance with the limit of maximum 50 UDP datagrams is not monitored. If the limit is exceeded, however, the correct function cannot be guaranteed.
- ▶ The maximum send data rate (= maximum UDP send rate) must be defined in PAS4000 for each task in which pssUDPSend is executed at least once.
A value in the range 128 ... 8192 may be configured for a task's maximum UDP send rate. The default setting is 1024 Bytes per task cycle
- ▶ The maximum data rate for receiving (= maximum UDP receive rate) must be defined in PAS4000 for each task in which pssUDPReceive is executed at least once.
A value in the range 128 ... 8192 may be configured for the maximum UDP receive rate. The default setting is 1024 Bytes per cycle of the UDP communication using blocks.

10.5.7 Time behaviour when sending a UDP datagram

You can start to send a UDP datagram by executing `pssUDPSend` during a task's execution time. The following applies when sending a UDP datagram:

- ▶ A maximum of 4 task cycles elapse from the moment `pssUDPSend` is executed (e.g. task cycle #m) until the IP connections system section sends the UDP datagram via the network (task cycle #m+3).
- ▶ The cycle time of UDP communication using blocks, which the IP connections system section uses to send a UDP datagram via the network, corresponds to the task cycle time t_{Task} of the task in which `pssUDPSend` was executed.

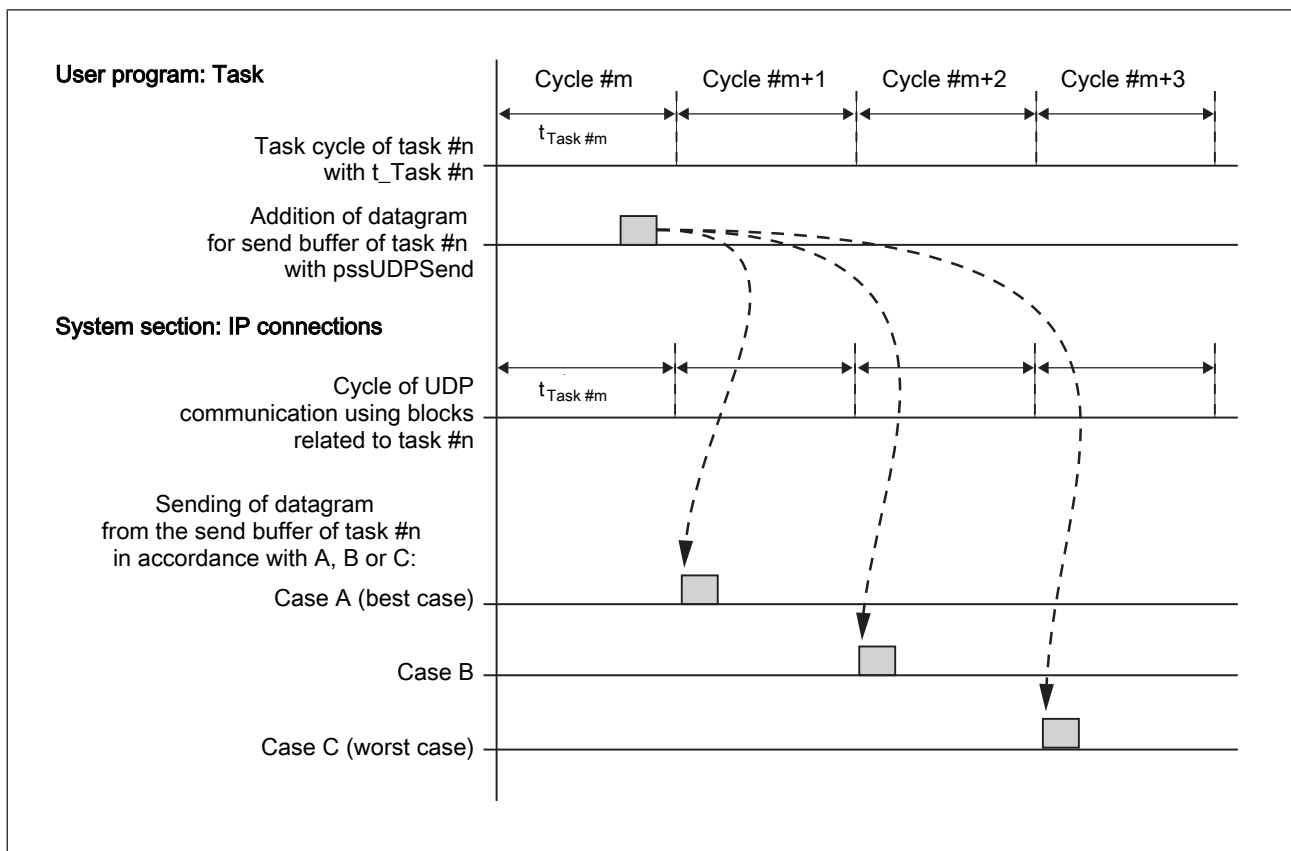


Fig.: Time behaviour when sending a UDP datagram with `pssUDPSend`



INFORMATION

In the same task cycle (e.g. task cycle #m) it is possible to execute both `pssCreateUDPSocket`, to create a UDP socket, and `pssUDPSend`, which uses this UDP socket to send a UDP datagram.

The UDP socket will be created in the very next task cycle (task cycle #m+1). At the very latest, the IP connections system section will send the UDP datagrams that use this socket in the third cycle of UDP communication using blocks (cycle #m+3) following task cycle #m.



NOTICE

The number of payload bytes on all UDP datagrams sent during a task cycle should be a maximum of half the maximum UDP send rate configured for the task (= max. number of payload bytes per task cycle); i.e. if possible, only half the configured maximum UDP send rate should be used. This way you can minimise delays between executing pssUDPSend (user program) and sending UDP datagrams (IP connections system section) via the network .

10.5.8 Time behaviour when receiving a UDP datagram

The following applies when receiving a UDP datagram:

- ▶ A maximum of 3 task cycles elapse from the moment a UDP datagram is received by the IP connections system section via the network until the UDP datagram is available in the target task's UDP receive buffer (see [Data exchange via "UDP using blocks" \[259\]](#)).
The cycle time of UDP communication using blocks, which the IP connections system section uses to receive a UDP datagram from the network, corresponds to the task cycle time t_{Task} of the target task's UDP datagram.
- ▶ The number of task cycles that elapse between the execution of pssUDPReceive and the UDP datagram being available in the user program depends on the design of the user program.



INFORMATION

In the same task cycle (e.g. task cycle #m) it is possible to execute both the block pssCreateUDPSocket, to create a UDP socket, and the block pssUDPReceive, which uses this UDP socket to receive UDP datagrams. The UDP socket will be created in the very next task cycle (task cycle #m+1). So the earliest that UDP datagrams that use this UDP socket will be available in the UDP receive buffer is the next but one task cycle (task cycle m+2).



NOTICE

The number of payload bytes on all UDP datagrams received during a cycle of UDP communication using blocks should be on average a maximum of half the maximum UDP receive rate configured for the task; i.e. if possible, only half the configured maximum UDP receive rate should be used. This way you can avoid delays when receiving UDP datagrams or avoid UDP datagrams being overwritten in the target task's UDP receive buffer.

10.5.9 Time requirement during UDP communication using blocks

UDP communication using blocks is a **non-cyclical** transmission procedure. For this reason, the time requirement during UDP communication using blocks can only be estimated approximately. In principle, the time requirement starts when the relevant UDP block is called in a user program task.



NOTICE

Although UDP communication using blocks counts as external communication and uses the IP connections system section, it is still a non-cyclical transmission procedure, unlike other PSS 4000 transmission procedures. As a result, the time requirement cannot be estimated using any of the formulas containing the cycle time for external communication t_{ExtCo} .

Key influences on time behaviour

- ▶ Time behaviour is mainly determined via the execution time of the task in which the UDP block is called. A task's execution time increases linearly depending on the maximum UDP send rate and UDP receive rate (see [Maximum UDP send rate and UDP receive rate](#) [📖 269]) plus the number of UDP datagrams sent and received.
- ▶ The size of a datagram has little effect on a task's execution time.
- ▶ If UDP blocks are executed in different tasks, this has no particular effect on the task's execution time. However you should note that tasks are executed according to their priority and the execution time may increase as a result (see Task properties and [Task execution](#) [📖 366]).

Influence of the maximum UDP send rate and UDP receive rate

The configured maximum UDP send rate and UDP receive rate increases a task's execution time. This influence is present irrespective of whether UDP datagrams are actually sent or received. The following relationship exists:

$$\text{Execution time}_{\text{Task (send and receive rate)}} = \text{Execution time}_{\text{Task}} + \text{Time}_{\text{UDP send rate}} + \text{Time}_{\text{UDP receive rate}}$$



INFORMATION

Please note that the execution time $\text{Execution time}_{\text{Task (send and receive rate)}}$ does not include the influence of the number of UDP datagrams sent and received.

The times of $\text{Time}_{\text{UDP send rate}}$ and $\text{Time}_{\text{UDP receive rate}}$ can only be estimated approximately. FS resource and ST resource must be considered separately.

Rule of thumb for estimating the time requirement on the FS resource

► Send:

$$\text{Time}_{\text{UDP send rate FS}} = (\text{Number of bytes}_{\text{Maximum UDP send rate}} * 0.9) \text{ microseconds}$$

► Receive:

$$\text{Time}_{\text{UDP receive rate FS}} = (\text{Number of bytes}_{\text{Maximum UDP receive rate}} * 1.3) \text{ microseconds}$$

Rule of thumb for estimating the time requirement on the ST resource

► Send:

$$\text{Time}_{\text{UDP send rate ST}} = (\text{Number of bytes}_{\text{Maximum UDP send rate}} * 0.3) \text{ microseconds}$$

► Receive:

$$\text{Time}_{\text{UDP receive rate ST}} = (\text{Number of bytes}_{\text{Maximum UDP receive rate}} * 0.24) \text{ microseconds}$$

Example

Configuration for an FS task:

► Maximum UDP send rate = 1500 Byte

► Maximum UDP receive rate = 2000 Byte

$$\text{Execution time}_{\text{Task (send and receive rate)}} = \text{Execution time}_{\text{Task}} + (1500 * 0.9) + (2000 * 1.3) \text{ microseconds}$$

Result:

The execution time of the FS task is increased by 3950 microseconds.

Influence of the number of UDP datagrams sent and received

The number of UDP datagrams sent and received increases a task's execution time; the size of a UDP datagram has little influence. The following relationship exists:

$$\text{Execution time}_{\text{Task (Number of datagrams)}} = \text{Execution time}_{\text{Task}} + \text{Time}_{\text{UDP send}} + \text{Time}_{\text{UDP receive}}$$



INFORMATION

Please note that the execution time_{Task (Number of datagrams)} does not include the influence of the maximum UDP send rate and UDP receive rate.

The times of Time_{UDP send} and Time_{UDP receive} can only be estimated approximately. FS resource and ST resource must be considered separately. The following rules of thumb apply for UDP datagrams with up to 100 Bytes payload.

Rule of thumb for estimating the time requirement on the FS resource

► Send:

$$(\text{Number}_{\text{Datagrams}} * 70) \text{ microseconds} \leq \text{Time}_{\text{UDP send FS}} \leq (\text{Number}_{\text{Datagrams}} * 80) \text{ microseconds}$$

► Receive:

$(\text{Number}_{\text{Datagrams}} * 100) \text{ microseconds} \leq \text{Time}_{\text{UDP receive FS}} \leq (\text{Number}_{\text{Datagrams}} * 120) \text{ microseconds}$

Using these rules of thumb it is possible to deduce that the execution time for a task increases by approx. 70 ... 80 microseconds for each UDP datagram sent in the task and by approx. 100 ... 120 microseconds for each UDP datagram received in the task.

Rule of thumb for estimating the time requirement on the ST resource

► Send:

$(\text{Number}_{\text{Datagrams}} * 60) \text{ microseconds} \leq \text{Time}_{\text{UDP send ST}} \leq (\text{Number}_{\text{Datagrams}} * 70) \text{ microseconds}$

► Receive:

$(\text{Number}_{\text{Datagrams}} * 80) \text{ microseconds} \leq \text{Time}_{\text{UDP receive ST}} \leq (\text{Number}_{\text{Datagrams}} * 100) \text{ microseconds}$

Using these rules of thumb it is possible to deduce that the execution time for a task increases by approx. 60 ... 70 microseconds for each UDP datagram sent in the task and by approx. 80 ... 100 microseconds for each UDP datagram received in the task.

Overall time requirement

As the influence of the maximum UDP send rate and UDP receive rate as well as the influence of the number of UDP datagrams sent and received must be taken into account, the following applies for calculating the overall time requirement:

$\text{Execution time}_{\text{Task overall}} = \text{Execution time}_{\text{Task}} + \text{Time}_{\text{UDP send rate}} + \text{Time}_{\text{UDP receive rate}} + \text{Time}_{\text{UDP send}} + \text{Time}_{\text{UDP Receive}}$

10.5.10 Maximum UDP send rate and UDP receive rate

In PAS4000, the maximum UDP send rate and maximum UDP receive rate must be configured for each task in which UDP communication is executed using the block pssUDPSend and/or pssUDPReceive.

Maximum UDP send rate

The maximum UDP send rate limits the number of payload bytes that may be sent within a task cycle when pssUDPSend is executed one or more times.

Rule of thumb for calculating the value of the maximum UDP send rate:

Based on a task's average send rate, double the value should be configured in PAS4000 for the task's maximum UDP send rate; i.e. the anticipated number of payload bytes sent within a task cycle using UDP datagrams should use half the configured maximum UDP send rate at most. The cycle time with which the IP connections system section sends UDP datagrams via the network is as large as the task cycle time t_{Task} .

Max. UDP send rate $\geq 2 \times$ max. number of send payload bytes per task cycle

If the value for the maximum UDP send rate is calculated using this rule of thumb and configured in PAS4000, then delays when sending UDP datagrams can be avoided.

Maximum UDP receive rate

The maximum UDP receive rate limits the number of payload bytes that may be received by the IP connections system section via UDP datagrams within a task cycle, with regard to the target task.

Rule of thumb for calculating the value of the maximum UDP receive rate:

Based on the target task's average receive rate, double the value should be configured in PAS4000 for the target task's maximum UDP receive rate; i.e. the anticipated number of payload bytes received for a target task within a task cycle using UDP datagrams should use half the configured maximum UDP receive rate at most. The cycle time with which the IP connections system section receives UDP datagrams for a target task is as large as the target task's cycle time t_{Task} .

Max. UDP receive rate $\geq 2 \times$ max. number of receive payload bytes per task cycle

If the value for the maximum UDP receive rate is calculated using this rule of thumb and configured in PAS4000, then delays when receiving UDP datagrams can be prevented.

10.5.11 Data consistency during UDP communication using blocks

This device is responsible for the data consistency of the receive data (payload) of a UDP datagram sent from the device of a third-party manufacturer. From the PSS 4000's perspective, no statement can be made regarding the data consistency of such a device.

Data consistency with receive data of a UDP datagram

If a PSS 4000-exclusive/operable device is the recipient of data from a UDP datagram, from the perspective of the device the receive data is external I-data.

External I-data from a UDP datagram is treated as consistent data in PSS 4000 (see [Data consistency](#)  377]).

The external I-data of a UDP datagram forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is the consistency range "ST-I data of a UDP communication using blocks" (see [Consistency ranges](#)  378]).

Data consistency with send data of a UDP datagram

If a PSS 4000-exclusive/operable device is the sender of a UDP datagram, from the perspective of the device the send data is external O-data.

The external O-data of a UDP datagram is consistent if it comes from the same consistency range on a data source (see [Data consistency](#)  377]).

10.5.12 Safe UDP communication using blocks

UDP communication using blocks is **non-safety-related** communication. This applies above all to the transmission route via UDP. Additional security measures are required to enable UDP communication using blocks to be used for safety-related applications. These security measures must be implemented through the application. The user is responsible for the correct implementation.

Please note the following:

► Block pssUDPSend

- If the block is executed on the FS resource, then the block processes the data from which the datagram is generated in a way that is safety-related.
- Should the PSS 4000 control system detect an error while generating the datagram, then the IP connections system section will not send the datagram. The PSS 4000 control system switches to the operating state "Major FS error".

► FS resource

Blocks:

- pssCreateUDPSocket
- pssReleaseUDPSocket
- pssUDPReceive
- pssCreateUDPSocketOnInterface
- pssCreateIPInterface
- pssReleaseIPInterface
- pssUseETHInterface
- pssReleaseETHInterface

Even if the blocks are executed on the FS resource, PSS 4000 cannot guarantee the safety-related execution of the blocks. Additional security measures must be implemented in the application.

► Transmission route via UDP

Additional security measures must be implemented in the application for the transmission route (e.g. black channel principle).



WARNING!

Hazard due to non-safety-related communication!

UDP communication using blocks is **non-safety-related** UDP communication. Data from a UDP communication using blocks may **not** be used for safety-related tasks without additional security measures, otherwise this could lead to serious injury or death, depending on the application.

For safe UDP communication you should ensure that

- Safety-related data is used in the application and
- Additional security measures are implemented in the application for the transmission route (e.g. black channel principle) and
- Appropriate procedures are used to verify safety for the application (e.g. through risk analysis and appropriate commissioning tests, plausibility tests)

10.6 PROFIBUS-DP slave

PROFIBUS is an open fieldbus standard whose communication is defined in the international standards IEC 61158 and IEC 61784. Further provisions have been defined in specifications published by the PROFIBUS User Group. These specifications are available from PROFIBUS International (see www.profibus.com).

For non-safety related applications, a PROFIBUS-DP enabled PSS 4000 device can be incorporated into PROFIBUS-DP as a Slave.

A PROFIBUS-DP enabled PSS 4000 device can be used with a Class 1 DP Master that supports the communication protocol PROFIBUS-DP with performance level DP-V0, DP-V1 or DP-V2.

As a PROFIBUS-DP Slave, a PROFIBUS-DP enabled PSS 4000 device has the following features:

- ▶ Uses the communication protocol PROFIBUS-DP with performance level DP-V0
- ▶ Devices addressed via DIP switches (e.g. on the head module of a PSSu system) or via the PROFIBUS-DP Master when using device address 126
- ▶ Selectable transmission rates (9.6 kBits/s ... 12 MBits/s)
- ▶ Supports the mode for automatic detection of the transmission rate (auto baud mode)
- ▶ Supports sync mode
- ▶ Supports freeze mode
- ▶ Supports the PROFIBUS-DP diagnostic telegram



INFORMATION

The description contains various technical terms, which are established and defined by the PROFIBUS standard. These include the terms "Master" and "Slave".

10.6.1 "PROFIBUS-DP Slave" system section

In the automation system PSS 4000, external communication via PROFIBUS-DP is possible with all PSS 4000 devices that have the "PROFIBUS-DP Slave" system section. In this case, the "PROFIBUS-DP Slave" system section performs all the tasks required for communication. For example these include managing the send and receive data areas and data exchange (see [Data exchange via PROFIBUS-DP](#)  275]).

The connection is configured in the PAS4000's PROFIBUS-DP Slave Editor.

10.6.2 Data exchange via PROFIBUS-DP

PSS 4000 devices with the "PROFIBUS-DP Slave" system section operate as PROFIBUS-DP Slaves when exchanging data via PROFIBUS-DP. It is always the PROFIBUS-DP Master that initiates data transfer. From the perspective of the automation system PSS 4000, the PROFIBUS-DP Master is outside the system.

10.6.2.1 Send data and receive data

The PROFIBUS-DP Master sends data from its send data area to the receive data area of the "PROFIBUS-DP Slave" system section of a PSS 4000 device and requests data from the "PROFIBUS-DP Slave" system section of a PSS 4000 device. The "PROFIBUS-DP Slave" system section of a PSS 4000 device sends the requested data from its send data area to the receive data area of the PROFIBUS-DP Master. In the automation system PSS 4000, the data areas are called "virtual PROFIBUS modules". Virtual PROFIBUS modules are available with a variety of data types.

For data exchange, corresponding virtual PROFIBUS modules must be configured for both the PROFIBUS-DP Master and "PROFIBUS-DP Slave" system section of a PSS 4000 device. The virtual PROFIBUS modules for send data are called O-modules. They have an "O" for "Output" in the module name (e.g. 8OX, OB, OW). The virtual PROFIBUS modules for receive data are called I-modules. They have an "I" for "Input" in the module name (e.g. 8IX, IB, IW).

The PROFIBUS DP Master obtains the required virtual PROFIBUS modules from the GSD file. This is supplied with PAS4000 and is also available in the download area of the Pilz homepage (www.pilz.de -> Downloads). "GSD" can be entered as the search term, for example. The software download is only available for registered users. Unregistered users can register free of charge.

The virtual PROFIBUS modules required for the "PROFIBUS-DP Slave" system section of a PSS 4000 device are configured during the PROFIBUS-DP Slave configuration in PAS4000. After configuration the virtual PROFIBUS modules are located on virtual slots (virtual slots 0 ... 79).

From the perspective of the automation system PSS 4000, the receive data is called external I-data, and the send data is called external O-data (see [Validity of process data](#)  359]).

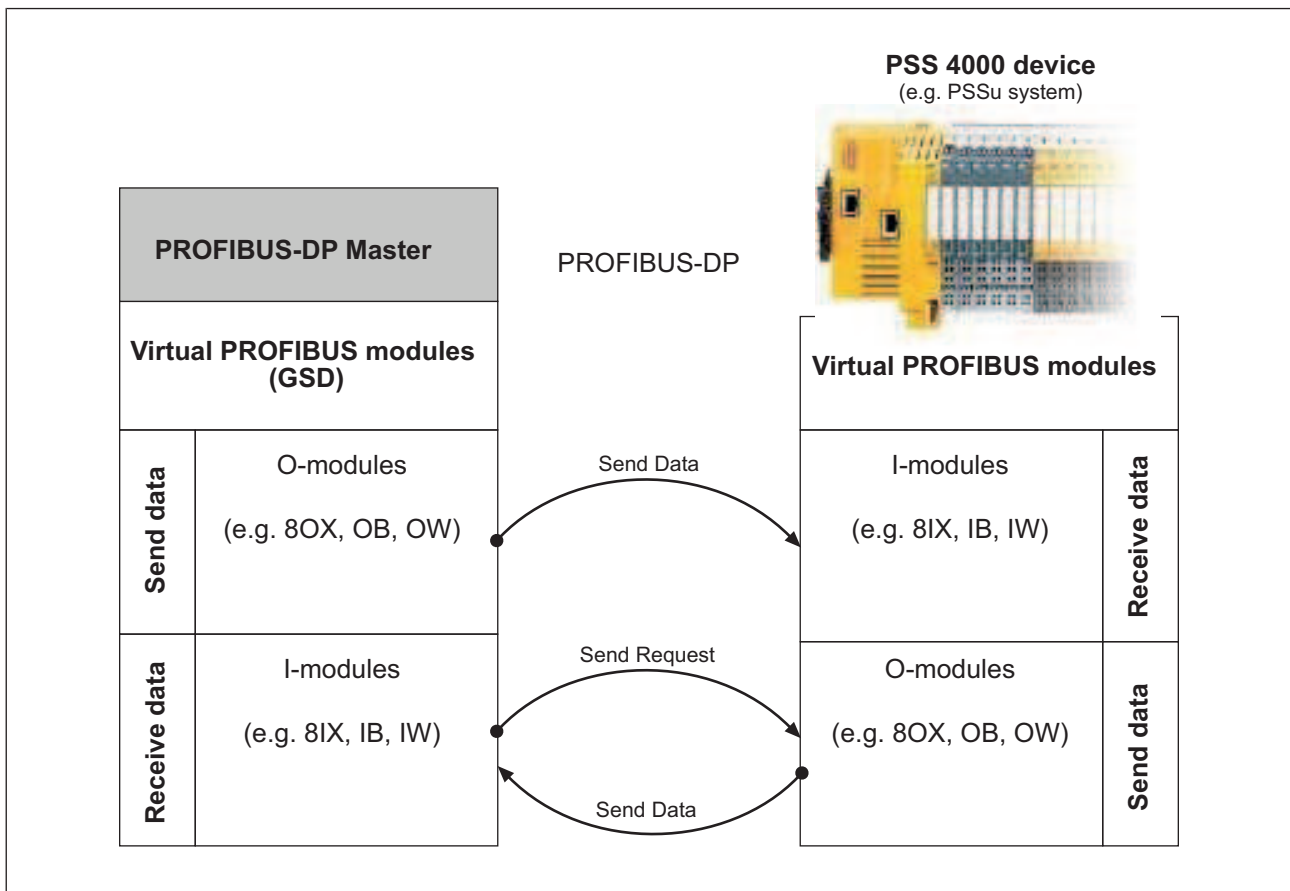


Fig.: Data exchange via virtual PROFIBUS modules (principle)

10.6.2.2 Virtual PROFIBUS modules for send data

The following virtual PROFIBUS modules (O-modules) can be configured in PAS4000 for the send data of a "PROFIBUS-DP Slave" system section:

Virtual PROFIBUS module	Data type	Application
8OX	ARRAY [0..7] OF BOOL	Virtual PROFIBUS bit module To send an array with 8 elements of data type BOOL
OB	BYTE	Virtual PROFIBUS byte module To send data of data type BYTE
OW	WORD	Virtual PROFIBUS byte module To send data of data type WORD
OD	DWORD	Virtual PROFIBUS byte module To send data of data type DWORD
OBAn	ARRAY [0..(n-1)] OF BYTE	Virtual PROFIBUS byte module To send an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
OWAn	ARRAY [0..(n-1)] OF WORD	Virtual PROFIBUS byte module To send an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64



INFORMATION

Note that only the virtual modules documented here with the specified data types are supported. The LWORD data type is not supported.

During data exchange, the PROFIBUS-DP Master requests data from O-modules on the automation system PSS 4000 and receives this data via its I-modules. Consequently, the O-modules on the automation system PSS 4000 must have corresponding I-modules in the PROFIBUS-DP Master. Also, the data type of an I-module in the PROFIBUS-DP Master must correspond to the data type of an O-module in the automation system PSS 4000.

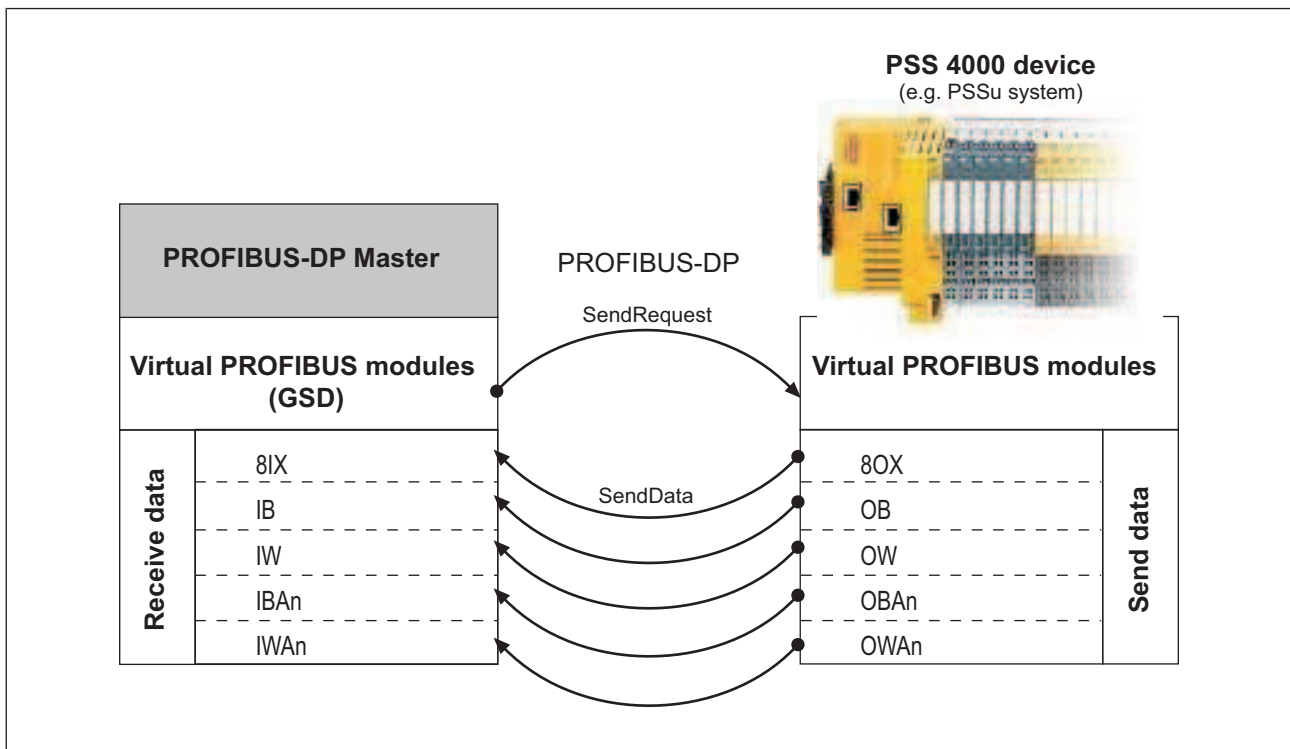


Fig.: Virtual PROFIBUS modules and their corresponding data types (send data of a PSS 4000 device)

10.6.2.3 Virtual PROFIBUS modules for receive data

The following virtual PROFIBUS modules (I-modules) can be configured in PAS4000 for the receive data of a "PROFIBUS-DP Slave" system section:

Virtual PROFIBUS module	Data type	Application
8IX	ARRAY [0..7] OF BOOL	Virtual PROFIBUS bit module To receive an array with 8 elements of data type BOOL
IB	BYTE	Virtual PROFIBUS byte module To receive data of data type BYTE
IW	WORD	Virtual PROFIBUS byte module To receive data of data type WORD
ID	DWORD	Virtual PROFIBUS byte module To receive data of data type DWORD
IBAn	ARRAY [0..(n-1)] OF BYTE	Virtual PROFIBUS byte module To receive an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
IWAn	ARRAY [0..(n-1)] OF WORD	Virtual PROFIBUS byte module To receive an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64



INFORMATION

Note that only the virtual modules documented here with the specified data types are supported. The LWORD data type is not supported.

During data exchange, the PROFIBUS-DP Master transmits its send data to the I-modules in the automation system PSS 4000 via O-modules. Consequently, the O-modules on the PROFIBUS-DP Master must have corresponding I-modules in the automation system PSS 4000. Also, the data type of an I-module must correspond to the data type of the O-module in the PROFIBUS-DP Master.

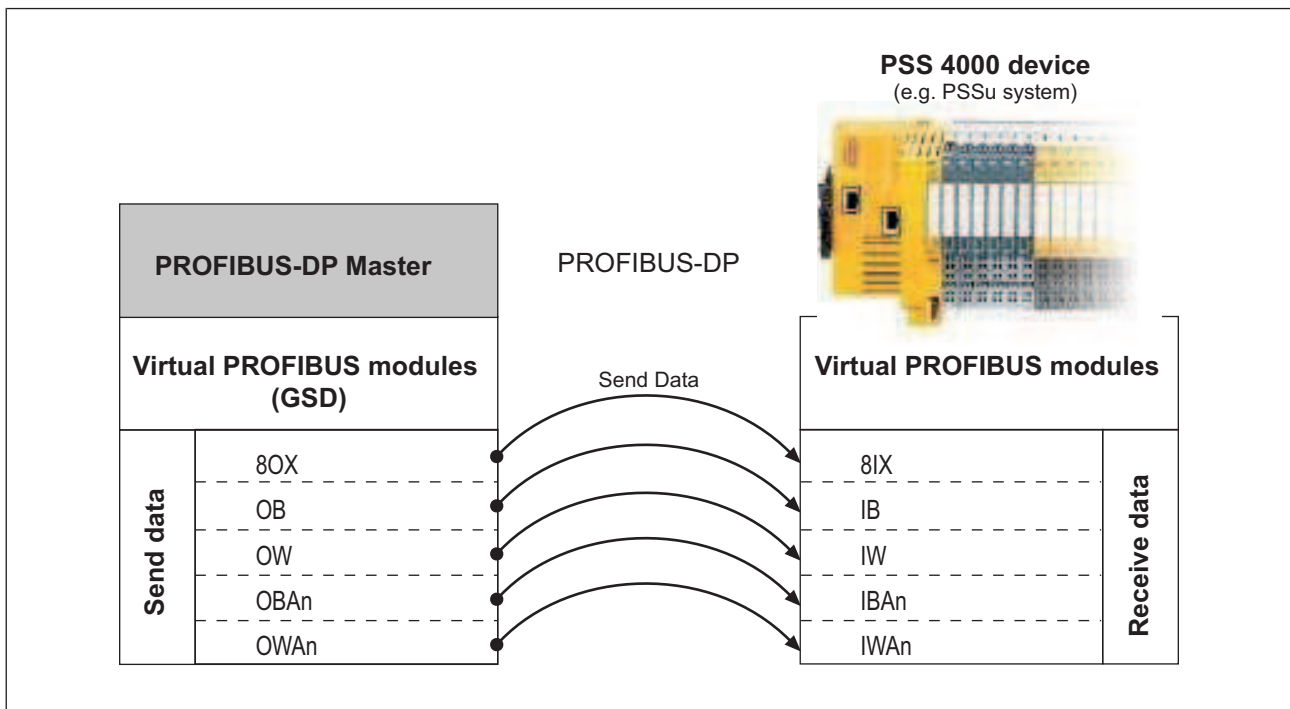


Fig.: Virtual PROFIBUS modules and their corresponding data types (receive data of a PSS 4000 device)

10.6.3 Data transfer limits

The following limits apply for data transfer via virtual PROFIBUS modules:

- ▶ A maximum of 80 virtual PROFIBUS modules with virtual slots 0 ... 79 can be configured for the "PROFIBUS-DP Slave" system section of a PSS 4000 device.
- ▶ A maximum of 244 Bytes of receive data (= external I-data) can be configured for the "PROFIBUS-DP Slave" system section of a PSS 4000 device.
- ▶ A maximum of 244 Bytes of send data (= external O-data) can be configured for the "PROFIBUS-DP Slave" system section of a PSS 4000 device.

10.6.4 I/O data of PROFIBUS-DP Slave

The "PROFIBUS-DP Slave" system section enables a PSS 4000 device to exchange I/O data with the PROFIBUS-DP Master via PROFIBUS-DP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding virtual PROFIBUS modules.

I/O-mapping of virtual PROFIBUS modules

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O-mapping for virtual PROFIBUS modules:

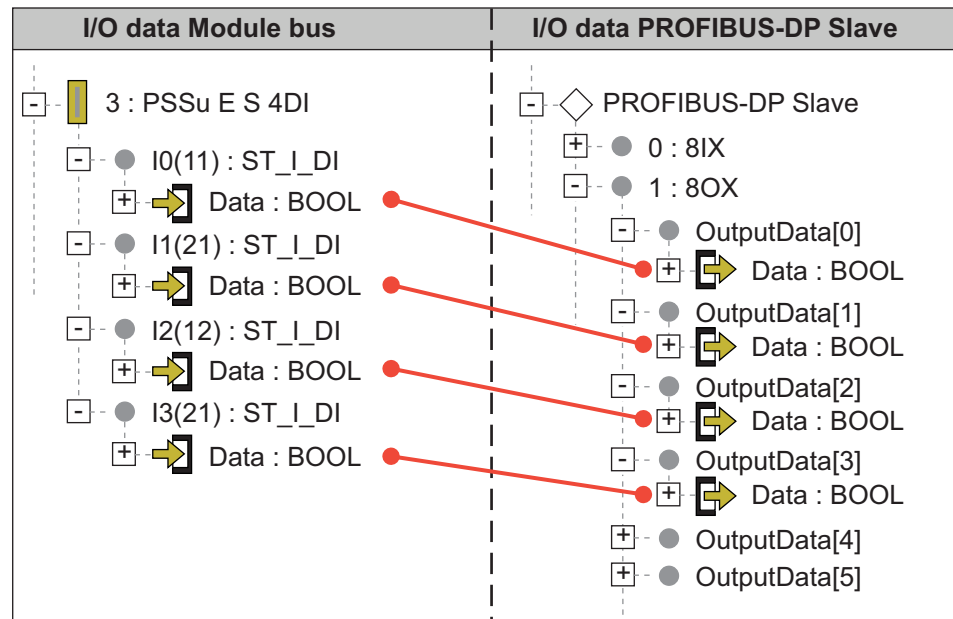
		Virtual PROFIBUS modules	
		Receive data (= external I-data of I modules such as 8IX, IB, IW, ID, IBAn, IWAn)	Transmit data (= external O-data of O-modules such as 8OX, OB, OW, OD, OBAn, OWAn)
Module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
PI variables	I-PI variable	◆	---
	O-PI variable	---	◆

- ◆ I/O mapping possible
- I/O mapping impossible

Examples

I/O-mapping for virtual PROFIBUS modules with send data

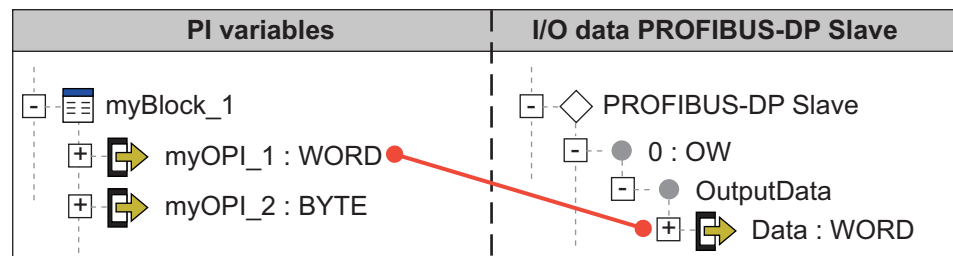
- ▶ The input signals of the hardware inputs of a PSSu E S 4DI are to be transferred to the PROFIBUS DP master.
 - Direction of data flow: Hardware input-> PROFIBUS DP
 - Virtual PROFIBUS module used: 8OX
 - Possible I/O mapping (example):



●—● I/O mapping

- The content of an O-PI variable of a block is to be transferred to the PROFIBUS DP master.

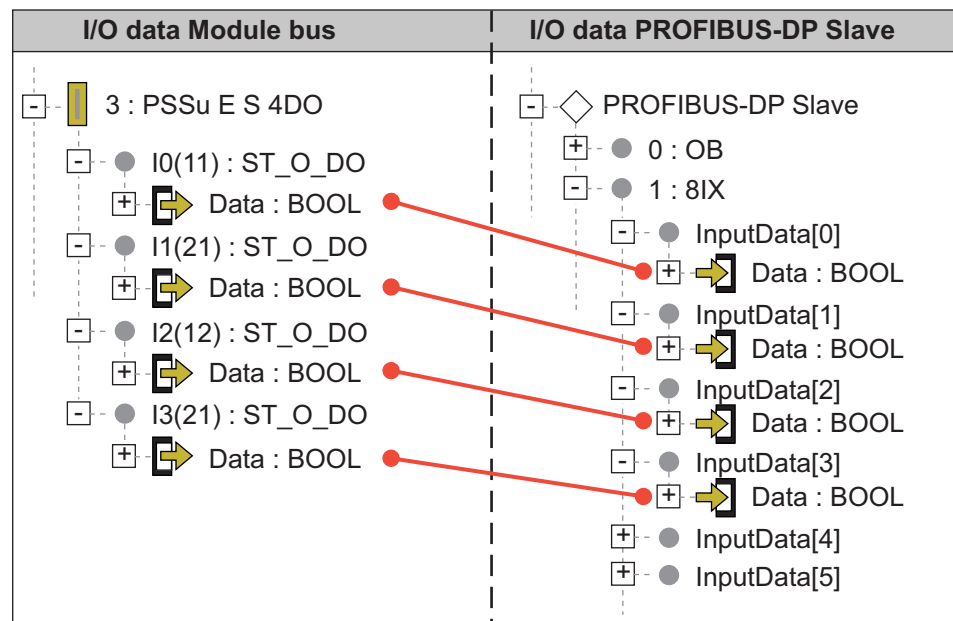
- Direction of data flow: O-PI variable -> PROFIBUS DP
- Virtual PROFIBUS module used: OW
- Possible I/O mapping (example):



●—● I/O mapping

I/O-mapping for virtual PROFIBUS modules with receive data

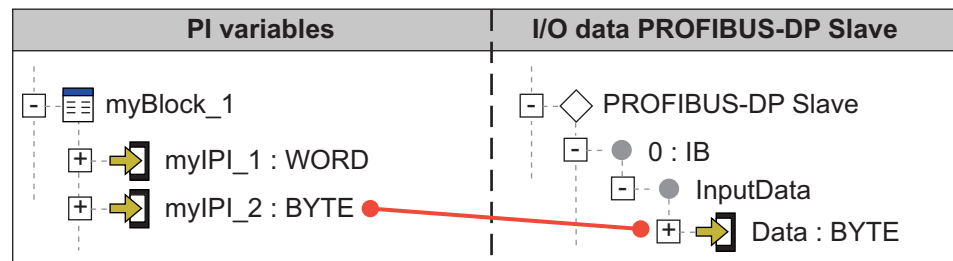
- PROFIBUS-DP master is to control the hardware outputs of a PSSu E S 4DO 0.5.
 - Direction of data flow: PROFIBUS-DP -> Hardware outputs
 - Virtual PROFIBUS module used: 8IX
 - Possible I/O mapping (example):



● I/O mapping

- ▶ A datum is to be transferred from the PROFIBUS-DP Master to the PSS 4000 device. Once the datum has been received it is to be made available for further processing within a block.

- Direction of data flow: PROFIBUS-DP -> I-PI variable
- Virtual PROFIBUS module used: IB
- Possible I/O mapping (example):



● I/O mapping

10.6.5 Address display of PROFIBUS-DP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

10.6.5.1 Address format of send data

The addresses of PROFIBUS-DP send data (O-data of virtual PROFIBUS modules) are structured as follows:

<Device name>.ProfibusDP.<Virtual slot>.OutputData[<Element number>]

Examples:

- ▶ Address format for an 8OX:

```
myDevice_1.ProfibusDP.2.OutputData[0]
myDevice_1.ProfibusDP.2.OutputData[1]
myDevice_1.ProfibusDP.2.OutputData[2]
myDevice_1.ProfibusDP.2.OutputData[3]
myDevice_1.ProfibusDP.2.OutputData[4]
myDevice_1.ProfibusDP.2.OutputData[5]
myDevice_1.ProfibusDP.2.OutputData[6]
myDevice_1.ProfibusDP.2.OutputData[7]
```

- ▶ Address format for an OB:

```
myDevice_1.ProfibusDP.3.OutputData
```

- ▶ Address format for an OW:

```
myDevice_1.ProfibusDP.4.OutputData
```

- ▶ Address format for an OBA2:

```
myDevice_1.ProfibusDP.5.OutputData[0]
myDevice_1.ProfibusDP.5.OutputData[1]
```

- ▶ Address format for an OWA4:

```
myDevice_1.ProfibusDP.6.OutputData[0]
myDevice_1.ProfibusDP.6.OutputData[1]
myDevice_1.ProfibusDP.6.OutputData[2]
myDevice_1.ProfibusDP.6.OutputData[3]
```

10.6.5.2 Address format of receive data

The addresses of PROFIBUS-DP receive data (I-data of virtual PROFIBUS modules) are structured as follows:

<Device name>.ProfibusDP.<Virtual slot>.InputData[<Element number>]

Examples:

- ▶ Address format for an 8IX:
 - myDevice_1.ProfibusDP.7.InputData[0]
 - myDevice_1.ProfibusDP.7.InputData[1]
 - myDevice_1.ProfibusDP.7.InputData[2]
 - myDevice_1.ProfibusDP.7.InputData[3]
 - myDevice_1.ProfibusDP.7.InputData[4]
 - myDevice_1.ProfibusDP.7.InputData[5]
 - myDevice_1.ProfibusDP.7.InputData[6]
 - myDevice_1.ProfibusDP.7.InputData[7]
- ▶ Address format for an IB:
 - myDevice_1.ProfibusDP.8.InputData
- ▶ Address format for an IW:
 - myDevice_1.ProfibusDP.9.InputData
- ▶ Address format for an IBA4:
 - myDevice_1.ProfibusDP.10.InputData[0]
 - myDevice_1.ProfibusDP.10.InputData[1]
 - myDevice_1.ProfibusDP.10.InputData[2]
 - myDevice_1.ProfibusDP.10.InputData[3]
- ▶ Address format for an IWA2:
 - myDevice_1.ProfibusDP.11.InputData[0]
 - myDevice_1.ProfibusDP.11.InputData[1]

10.6.6 PROFIBUS diagnostics

PROFIBUS-DP diagnostics can be performed via the PSS 4000 system diagnostics (see [Diagnostic basics](#) [ 722]).

However, the standard defines proprietary diagnostic options for PROFIBUS-DP. The current diagnostic information can be polled by a PROFIBUS-DP Master on the PROFIBUS-DP Slave via a diagnostic telegram.

The standard divides the diagnostic information in the diagnostic telegram into the following areas:

- ▶ [System-internal diagnostics](#) [ 288]
- ▶ [External diagnostics](#) [ 289]

The system-internal diagnostic data is located in the first 6 Bytes of the diagnostic telegram and its meaning is firmly specified in the standard (standard diagnostics). These obligatory 6 Bytes may be followed by one block or several blocks containing external diagnostic data.

On PSS 4000 devices with a "PROFIBUS-DP Slave" system section, a PROFIBUS DP diagnostic telegram consists of a min. 6 Bytes of diagnostic data (obligatory diagnostic data) and a max. of 12 Bytes of diagnostic data (6 Bytes of obligatory diagnostic data + 6 Bytes of external diagnostic data).

The diagnostic information is part of the GSD file.

10.6.6.1 System-internal diagnostics

In accordance with the standard, a diagnostic telegram always starts with 6 bytes of system-internal diagnostic data (standard diagnostics). The structure, content and meaning of the system-internal diagnostic data is firmly specified in the standard.

Structure and content of the system-internal diagnostic data

Byte No.	Bit No.	Meaning
1	0	Station_Non_Existent
	1	Station_Not_Ready
	2	Cfg_Fault
	3	Ext_Diag 0: No external diagnostic data present 1: External diagnostics data present
	4	Not_Supported
	5	Invalid_Slave_Response
	6	Prm_Fault
	7	Master_lock
2	0	Prm_req
	1	Stat_Diag
	2	Always "1"
	3	WD_on
	4	Freeze_Mode
	5	Sync_Mode
	6	Reserved
	7	Deactivated
3	6 ... 0	Reserved
	7	Ext_Diag_Overflow
4	7 ... 0	Address of the PROFIBUS-DP master
5	7 ... 0	Ident number of PROFIBUS-DP slave: High Byte PSSu systems: 0C
6	7 ... 0	Ident number of PROFIBUS-DP slave: Low Byte PSSu systems: B0



INFORMATION

The exact meaning of the Bytes 1 ... 6 can be found in the IEC 61158-6-3 (5.3.1) and IEC 61158-6-5 (6.2.3.3.1).

10.6.6.2 External diagnostics data

In accordance with the standard, a diagnostic block starts with a Header Byte. This Header Byte determines the type and length of the diagnostic block. The standard defines three formats:

- ▶ Device-based diagnostics
- ▶ ID-based diagnostics
- ▶ Channel-based diagnostics

PSS 4000 supports **device-based diagnostics**. 6 Bytes including the Header Byte are required for device-based diagnostic data on a PSS 4000 device with "PROFIBUS-DP Slave" system section.

Structure and content of device-based diagnostic data of PSS 4000 devices with "PROFIBUS-DP Slave" system section

Byte No.	Bit No.	Value	Meaning
n	0 ... 7	06h	Header
			Bit 7 and 6: 00 Device with "PROFIBUS-DP Slave" system section: Code for device-based diagnostics
			Bits 5 ... 0: 000110 Number of bytes in the device-based diagnostics (including Header Byte) Devices with "PROFIBUS-DP Slave" system section send 6 Bytes
n + 1	Reserved		
n + 2	Reserved		
n + 3	Reserved		

Byte No.	Bit No.	Value	Meaning
n + 4	0	0/1	Error message for the device
			0 No message
			1 A message of "Error" severity is present for at least one system section of the device or A major FS error is present for at least one FS system section of the device On PSS 4000-exclusive devices, this corresponds to the "DIAG" status LED: Status: "lights up red" or "flashes red".
	1	0/1	Error in the device's FS/ST module bus system sections
			0 No message
			1 Operating state "Safe state of all FS outputs on the device" or At least one module cannot be accessed (e.g. a module has been removed during operation, set/ actual hardware registry does not match) or Operating state "FS module bus in a STOP state with error: Major FS error" On PSS 4000-exclusive devices, this corresponds to the "MBUS" status LED: Status: "lights up red" or "flashes red".
	2	0/1	Error in the "FS SafetyNET p" system section of the PSS 4000 device
			0 No message
			1 Operating state "FS SafetyNET p in STOP state with error: Major FS error" or Operating state "FS SafetyNET p in STOP state with error: Major FS+ST error" On PSS 4000-exclusive devices, this corresponds to the "FS SNp" status LED: Status: "lights up red" or "flashes red".
	3	0/1	Error in the "ST SafetyNET p" system section of the device
			0 No message
			1 Operating state "ST SafetyNET p in STOP state with error: Major FS+ST error" On PSS 4000-exclusive devices, this corresponds to the "ST SNp" status LED: Status: "lights up red" or "flashes red".
	4 ... 7	0	Reserved

Byte No.	Bit No.	Value	Meaning
n + 5	0	0/1	Warning message for the device
			0 No message
			1 A message of "Warning" severity is present for the device On PSS 4000-exclusive devices, this corresponds to the "DIAG" status LED: Status: "lights up orange".
	1	0/1	Forcing on the FS resource
			0 No message
			1 Forcing on the FS resource is active and/or there is at least one online change active On PSS 4000-exclusive devices, this corresponds to the "FS FORCE" status LED: Status: "lights up yellow".
	2	0/1	Forcing on the ST resource
			0 No message
			1 Forcing on the ST resource is active and/or there is at least one online change active On PSS 4000-exclusive devices, this corresponds to the "ST FORCE" status LED: Status: "lights up yellow".
	3 ... 7	0	Reserved

10.6.7 Data consistency on PROFIBUS-DP

The data consistency of the data exchanged between the PROFIBUS-DP Master and PSS 4000 depends on the configuration of the PROFIBUS-DP Master.

PROFIBUS-DP supports three types of data consistency:

- ▶ Byte consistency
Bytes are transmitted as whole units.
- ▶ Word consistency
The two bytes in a word are always transmitted together.
- ▶ Buffer consistency
The whole data buffer is transmitted as one unit without a break.
From the perspective of the PSS 4000, the data buffer corresponds to the respective virtual PROFIBUS module for send data or receive data.

Data consistency of the receive data

From the perspective of the PSS 4000, the receive data of a virtual PROFIBUS module is external I-data.

As soon as external I-data is available in a virtual PROFIBUS module, it is treated as consistent data (see [Data consistency \[377\]](#)). The external I-data from a PROFIBUS module forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a PROFIBUS-DP connection" (see [Consistency ranges \[378\]](#)).

Data consistency of the send data

From the perspective of the PSS 4000, the send data of a virtual PROFIBUS module is external O-data.

The external O-data of a virtual PROFIBUS module is consistent if it comes from the same consistency range on a data source.

PSS 4000 always treats the external O data of a virtual PROFIBUS module as one unit. It is the task of the PROFIBUS-DP Master to guarantee the necessary data consistency.

10.7 PROFIBUS-DP Slave with PROFIsafe

PROFIsafe functionality can be added to the non-safety-related fieldbus system PROFIBUS-DP for safety-related communication. In this case, the fieldbus system is superimposed with the PROFIsafe protocol in accordance with the black channel principle. Communication via PROFIsafe is defined in the international standard IEC 61784-3-3.

▸ PROFIBUS-DP

A PSS 4000 device with the "PROFIBUS-DP Slave with PROFIsafe" system section can be incorporated into PROFIBUS-DP as a Slave for **non-safety-related** applications.

▸ PROFIsafe

A PSS 4000 device with the "PROFIBUS-DP Slave with PROFIsafe" system section can be incorporated into PROFIsafe as an F-device (F-slave) for **safety-related** applications.

A PSS 4000 device that operates as a PROFIBUS-DP Slave and/or F-device can be used with a Class 1 DP Master that supports the communication protocol PROFIBUS-DP with performance level DP-V0, DP-V1 or DP-V2.

Supported features for PROFIBUS-DP

- Uses the communication protocol PROFIBUS-DP with performance level DP-V0 for data exchange via PROFIBUS-DP
- Uses the communication protocol PROFIBUS-DP with performance level DP-V1 for data exchange via PROFIsafe
- Devices addressed via DIP switches (e.g. on the head module of a PSSu system) or via the PROFIBUS-DP Master when device address 126 is used on the DIP switch
- Selectable transmission rates (9.6 kBits/s ... 12 MBits/s)
- Supports the mode for automatic detection of the transmission rate (auto baud mode)
- Supports sync mode
- Supports freeze mode
- Supports the PROFIBUS-DP diagnostic telegram
- Data exchange via pre-defined virtual PROFIBUS modules (I-modules, O-modules)

Supported features for PROFIsafe

- Uses the communication protocol PROFIBUS-DP with performance level DP-V1 for data exchange via PROFIsafe
- PROFIsafe address is assigned via DIP switch (e.g. on the head module of a PSSu systems) or via PAS4000 when PROFIsafe address 0 is used on the DIP switch
- F-parameters are displayed and edited via the Master Tool (e.g. Siemens-Software HW Config)
- Data exchange via pre-defined virtual F-modules



INFORMATION

The description contains various technical terms, which are established and defined by the PROFIBUS standard. These include the terms "Master" and "Slave".

10.7.1 "PROFIBUS-DP Slave with PROFI-safe" system section

In the automation system PSS 4000, external communication via PROFIBUS-DP and/or PROFI-safe is possible with all PSS 4000 devices that have the "PROFIBUS-DP Slave with PROFI-safe" system section. In this case, the "PROFIBUS-DP Slave with PROFI-safe" system section performs all the tasks required for communication. For example, these include:

- ▶ Management of the send and receive data areas
- ▶ Data exchange via PROFIBUS-DP
- ▶ Data exchange via PROFI-safe

10.7.2 PROFIBUS-DP in the "PROFIBUS-DP Slave with PROFIsafe" system section

PSS 4000 devices with the "PROFIBUS-DP Slave with PROFIsafe" system section send **non-safety-related data** via PROFIBUS-DP. When exchanging data via PROFIBUS-DP they operate as PROFIBUS-DP Slaves. It is always the PROFIBUS-DP Master that initiates data transfer. From the perspective of the automation system PSS 4000, the PROFIBUS-DP Master is outside the system; i.e. the receive data is called external I-data and the send data is called external O-data (see [Validity of process data](#)  359).

Send data and receive data

The PROFIBUS-DP Master sends data from its send data area to the receive data area of the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device and requests data from the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device.

The "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device sends the requested data from its send data area to the receive data area of the PROFIBUS-DP Master.

In the automation system PSS 4000, these data areas are called "virtual PROFIBUS modules". The virtual PROFIBUS modules for receive data are called I-modules. They have an "I" for "Input" in the module name (e.g. 8IX, IB, IW). The virtual PROFIBUS modules for send data are called O-modules. They have an "O" for "Output" in the module name (e.g. 8OX, OB, OW).

Virtual PROFIBUS modules must be configured for both the PROFIBUS-DP Master and the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device. The PROFIBUS-DP Master contains all the necessary information on the potential virtual modules from the GSD file (see [GSD file for virtual PROFIBUS modules](#)  302).

The virtual PROFIBUS modules required for the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device are configured during the hardware configuration in PAS4000's PROFIBUS-DP Slave with PROFIsafe Editor. After configuration the virtual modules are located on virtual slots.

The following functionalities and procedures are described for the "PROFIBUS-DP Slave" system section, but also apply to the "PROFIBUS-DP-Slave with PROFIsafe" system section:

- ▶ Virtual PROFIBUS modules for receive data (I-modules)
(see [Virtual PROFIBUS modules for receive data](#)  279)
- ▶ Virtual PROFIBUS modules for send data (O-modules)
(see [Virtual PROFIBUS modules for send data](#)  277)
- ▶ PROFIBUS-DP Slave I/O data
(see [PROFIBUS-DP Slave I/O data](#)  282)
- ▶ Address format of PROFIBUS-DP addresses
(see Address format of PROFIBUS-DP addresses)
- ▶ PROFIBUS diagnostics
(see [PROFIBUS diagnostics](#)  287)
- ▶ Data consistency on PROFIBUS-DP
(see [Data consistency on PROFIBUS-DP](#)  292)

10.7.3 PROFIsafe in the "PROFIBUS-DP Slave with PROFIsafe" system section

PSS 4000 devices with the "PROFIBUS-DP Slave with PROFIsafe" system section send and receive **safety-related data** via PROFIsafe. When exchanging data via PROFIsafe they operate as F-devices. It is always the F-host (F-master) that initiates data transfer. From the perspective of the automation system PSS 4000, the F-host is outside the system; i.e. the receive data is called external I-data and the send data is called external O-data (see [Validity of process data](#)  359).

Send data and receive data

The F-host sends data from its send data area to the receive data area of the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device and requests data from the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device.

The "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device sends the requested data from its send data area to the receive data area of the F-host.

In the automation system PSS 4000, these data areas are called "virtual PROFIBUS modules". Various pre-defined virtual F-modules are available for data exchange via PROFIsafe. The F-modules consist of virtual PROFIsafe submodules for receive and send data. An F-module has an overall data length of 24 Bytes; 8 of these Bytes are assigned with virtual PROFIsafe submodules for receive data and 8 Bytes are assigned with virtual PROFIsafe submodules for send data.

The virtual PROFIsafe submodules for receive data are called I-submodules. They have an "I" for "Input" in the module name (e.g. 8I_SAFEBOOL, I_SAFEBYTE, I_SAFEWORD, I_SAFEDWORD, I_SAFEINT). The virtual PROFIsafe submodules for send data are called O-submodules. They have an "O" for "Output" in the module name (e.g. 8O_SAFEBOOL, O_SAFEBYTE, O_SAFEWORD, O_SAFEDWORD, O_SAFEINT).

Virtual PROFIBUS modules must be configured for both the F-host and the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device. The F-host obtains the required virtual PROFIBUS modules from the GSD file (see [GSD file for virtual PROFIBUS modules](#)  302).

The F-modules required for the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device are configured during the hardware configuration in PAS4000's PROFIBUS-DP Slave with PROFIsafe Editor. After configuration, an F-module is located on a virtual slot with subslots for the virtual PROFIsafe submodules.

10.7.3.1 Pre-defined F-modules

Pre-defined F-modules are available for a PSS 4000 device in PAS4000. These F-modules are pre-assigned with virtual PROFIsafe submodules (I/O submodules).

The F-module of a PSS 4000 device must have a corresponding F-module in the PROFIsafe F-host. The assignment of the F-modules is as follows:

F_32IX_32IX_32OX_32OX

The F-module F_32IX_32IX_32OX_32OX of a PSS 4000 device must have a corresponding F_32OX_32OX_32IX_32IX in the F-host.

F-host		PSS 4000 device		
F_32OX_32OX_32IX_32IX		F_32IX_32IX_32OX_32OX (default assignment)		
		Virtual subslot	I/O submodules	
Output	Unsigned32	0	8I_SAFEBOOL	Input
		1	8I_SAFEBOOL	
		2	8I_SAFEBOOL	
		3	8I_SAFEBOOL	
	Unsigned32	4	8I_SAFEBOOL	
		5	8I_SAFEBOOL	
		6	8I_SAFEBOOL	
		7	8I_SAFEBOOL	
Input	Unsigned32	8	8O_SAFEBOOL	Output
		9	8O_SAFEBOOL	
		10	8O_SAFEBOOL	
		11	8O_SAFEBOOL	
	Unsigned32	12	8O_SAFEBOOL	
		13	8O_SAFEBOOL	
		14	8O_SAFEBOOL	
		15	8O_SAFEBOOL	

F_32IX_IWA2_32OX_OWA2

The F-module F_32IX_IWA2_32OX_OWA2 of a PSS 4000 device must have a corresponding F_32OX_OWA2_32IX_IWA2 in the F-host.

F-host		PSS 4000 device		
F_32OX_OWA2_32IX_IWA2		F_32IX_IWA2_32OX_OWA2 (default assignment)		
		Virtual subslot	I/O submodules	
Output	Unsigned32	0	8I_SAFEBOOL	Input
		1	8I_SAFEBOOL	
		2	8I_SAFEBOOL	
		3	8I_SAFEBOOL	
	Integer16	4	I_SAFEINT	
	Integer16	5	I_SAFEINT	
Input	Unsigned32	6	8O_SAFEBOOL	Output
		7	8O_SAFEBOOL	
		8	8O_SAFEBOOL	
		9	8O_SAFEBOOL	
	Integer16	10	O_SAFEINT	
	Integer16	11	O_SAFEINT	

F_IWA4_OWA4

The F-module F_IWA4_OWA4 of a PSS 4000 device must have a corresponding F_OWA4_IWA4 in the F-host.

F-host		PSS 4000 device		
F_OWA4_IWA4		F_IWA4_OWA4		
		Virtual subslot	I/O submodules	
Output	Integer16	0	I_SAFEINT	Input
	Integer16	1	I_SAFEINT	
	Integer16	2	I_SAFEINT	
	Integer16	3	I_SAFEINT	

F-host		PSS 4000 device		
Input	Integer16	4	O_SAFEINT	Output
	Integer16	5	O_SAFEINT	
	Integer16	6	O_SAFEINT	
	Integer16	7	O_SAFEINT	

10.7.3.2 Virtual PROFIsafe submodules for send data

If an F-module contains O-submodules of SAFEBOOL data type, these can be assigned with O-submodules of another data type in PAS4000. This applies to the two F-modules F_32IX_32IX_32OX_32OX and F_32IX_IWA2_32OX_OWA2, for example. The following O-submodules are available for the configuration:

O-submodule	Data type	Usage
8O_SAFEBOOL	ARRAY [0..7] OF SAFE- BOOL	Virtual bit module Sends an array with 8 elements of SAFEBOOL data type
O_SAFEBYTE	SAFEBYTE	Virtual byte module Sends data of SAFEBYTE data type
O_SAFEWORD	SAFEWORD	Virtual WORD module Sends data of SAFEWORD data type
O_SAFEDWORD	SAFEDWORD	Virtual DWORD module Sends data of SAFEDWORD data type
O_SAFEINT	SAFEINT	Virtual INT module Sends data of SAFEINT data type



INFORMATION

Note that only the virtual modules documented here with the specified data types are supported. The SAFELWORD data type is not supported.

10.7.3.3

Virtual PROFIsafe submodules for receive data

If an F-module contains I-submodules of a SAFEBOOL data type, these can be assigned with I-submodules of another data type in PAS4000. This applies to the two F-modules F_32IX_32IX_32OX_32OX and F_32IX_IWA2_32OX_OWA2, for example. The following I-submodules are available for the configuration:

I-submodule	Data type	Usage
8I_SAFEBOOL	ARRAY [0..7] OF SAFE- BOOL	Virtual bit module Receives an array with 8 elements of BOOL data type
I_SAFEBYTE	SAFEBYTE	Virtual byte module Receives data of SAFEBYTE data type
I_SAFEWORD	SAFEWORD	Virtual WORD module Receives data of SAFEWORD data type
I_SAFEDWORD	SAFEDWORD	Virtual DWORD module Receives data of SAFEDWORD data type
I_SAFEINT	SAFEINT	Virtual INT module Receives data of SAFEINT data type

**INFORMATION**

Note that only the virtual modules documented here with the specified data types are supported. The SAFELWORD data type is not supported.

10.7.4 GSD file for virtual PROFIBUS modules

All virtual PROFIBUS modules available for the "PROFIBUS-DP-Slave with PROFIsafe" system section are contained in the corresponding GSD file.



INFORMATION

Only use the GSD file that belongs to the "PROFIBUS-DP-Slave with PROFIsafe" system section. It contains both the I-modules and O-modules for the PROFIBUS-DP Master, as well as the F-modules for the PROFIsafe F-host.

The GSD file is supplied with PAS4000 and is also available in the download area of the Pilz homepage (www.pilz.de -> Downloads). "GSD" can be entered as the search term, for example. The software download is only available for registered users. Unregistered users can register free of charge.

10.7.5 Data transfer limits

On a PSS 4000 device that has the "PROFIBUS-DP Slave with PROFIsafe" system section, data transfer is via virtual PROFIBUS modules (I-modules, O-modules, F-modules). The following limits apply for the "PROFIBUS-DP Slave with PROFIsafe" system section of a device:

- ▶ A maximum of 80 virtual PROFIBUS modules can be configured with the virtual slots 0 ... 79
- ▶ A maximum of one F-module can be configured
Please note: An F-module reduces the total number of virtual PROFIBUS modules available not by 1 but by 5.
- ▶ An F-module needs a total of 24 Bytes, of which
 - Exactly 8 Bytes are reserved for I-submodules (virtual PROFIsafe submodules for receive data)
 - Exactly 8 Bytes are reserved for O-submodules (virtual PROFIsafe submodules for send data)
- ▶ An F-module has a max. of 16 subslots for virtual PROFIsafe submodules
- ▶ A max. of 244 Bytes of receive data (= external I-data) can be configured
- ▶ A max. of 244 Bytes of send data (= external O-data) can be configured

10.7.6 PROFIsafe I/O data

The "PROFIBUS-DP Slave with PROFIsafe" system section enables a PSS 4000 device to exchange safety-related I/O data with the F-host via PROFIsafe. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the virtual PROFIsafe submodules of an F-module.

I/O mapping on F-modules

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O-mapping for F-modules:

I/O mapping		on F-modules	
		Receive data (= external I-data of virtual PROFIsafe submodules such as 8I_SAFEBOOL, I_SAFEBYTE, I_SAFEWORD, I_SAFEDWORD, I_SAFEINT for example)	Send data (= external O-data of virtual PROFIsafe submodules such as 8O_SAFEBOOL, O_SAFEBYTE, O_SAFEWORD, O_SAFEDWORD, O_SAFEINT for example)
of module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	----
of PI-variable	I-PI variable	◆	---
	O-PI variable	---	◆

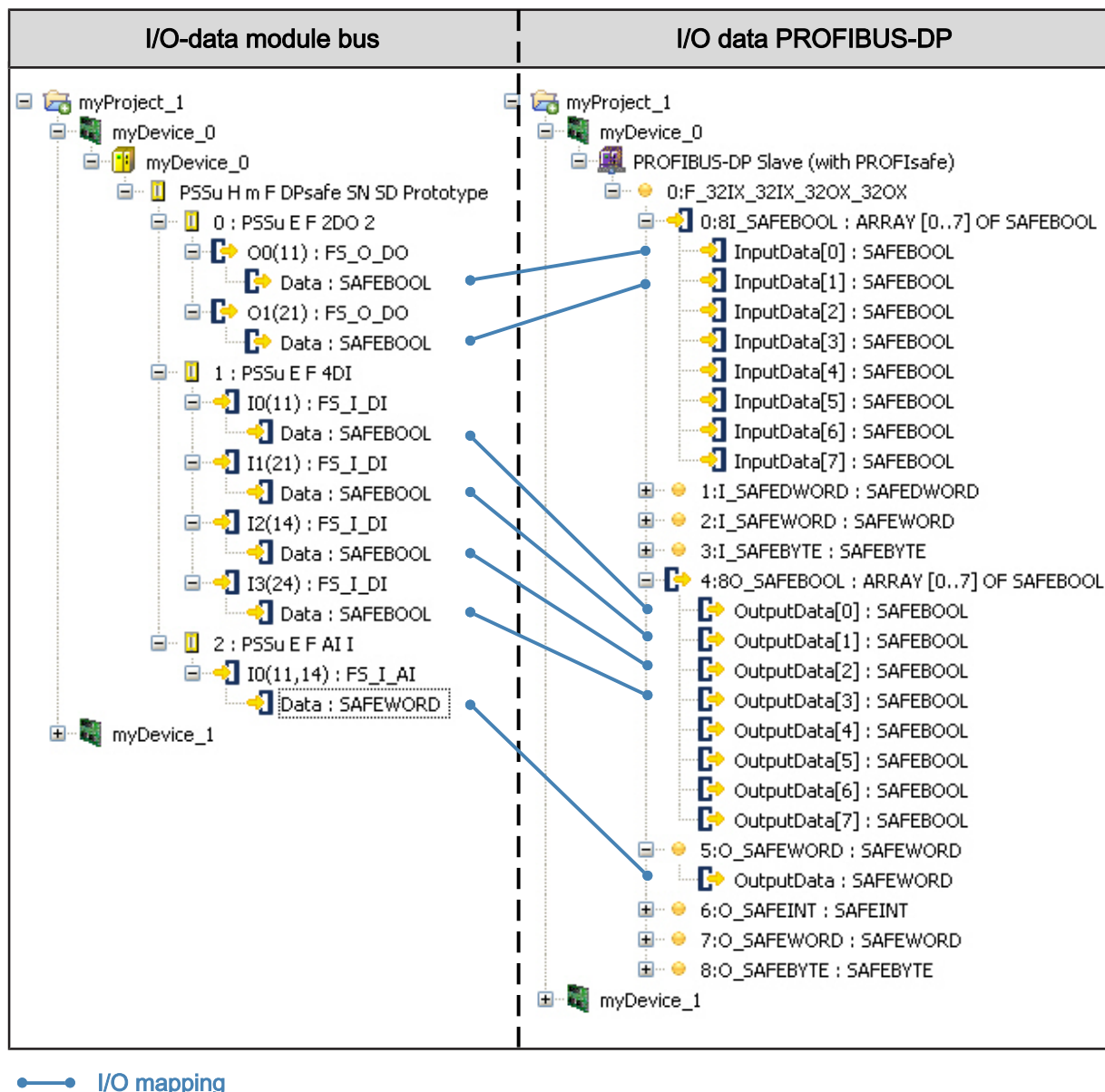
- ◆ I/O mapping possible
- I/O mapping impossible

Examples

► I/O mapping of I/O data

The F-host is to activate a PSSu E F 2DO 2 and the I-data of a PSSu E F 4DI and a PSSu E F AI I is to be transmitted to the F-host. The F-module F_32IX_32IX_32OX_32OX is configured with the necessary virtual I/O submodules for this purpose.

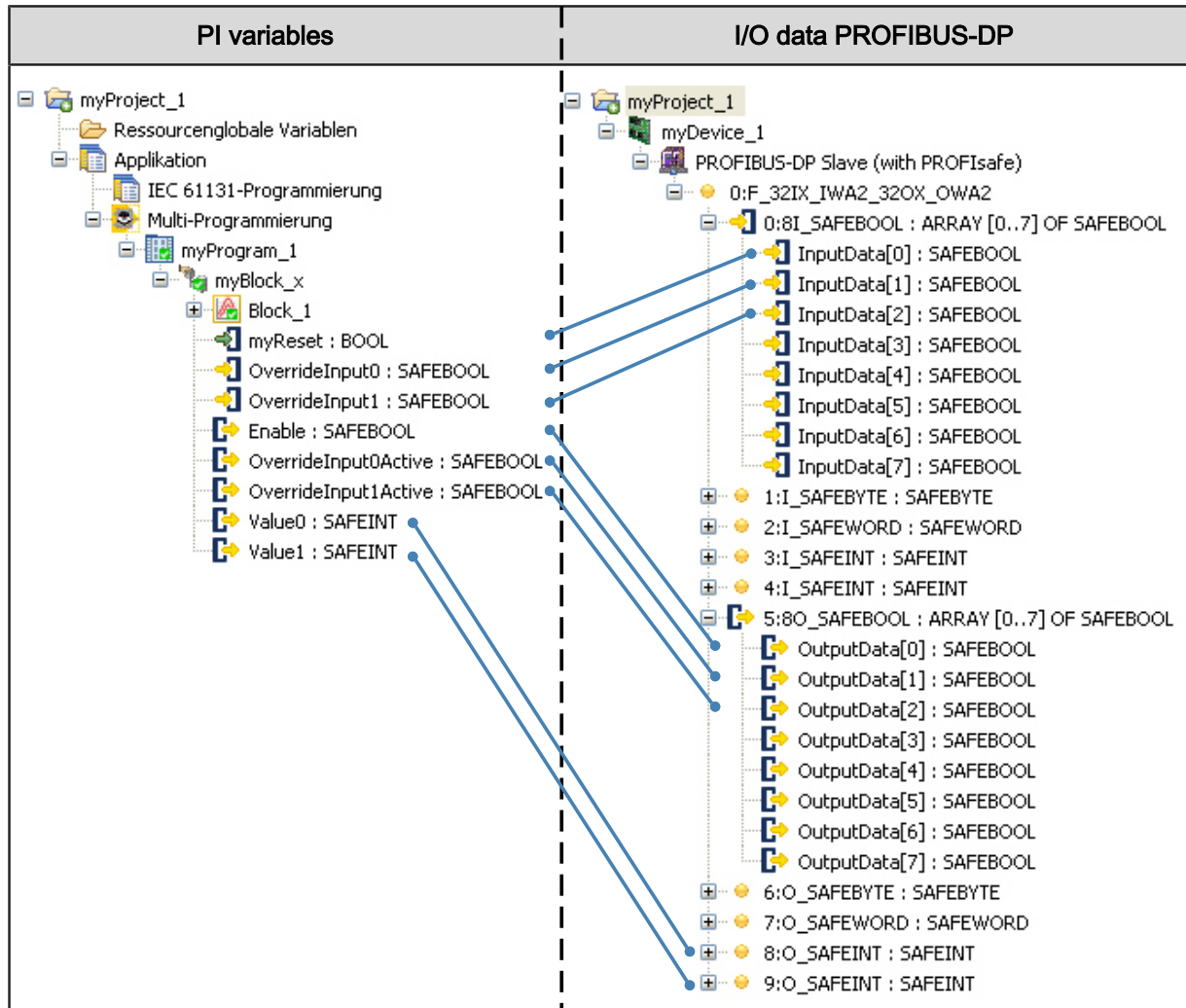
Possible I/O mapping (example):



► I/O mapping of PI variables

The contents of a block's O-PI variables are to be transmitted to the F-host; receive data arriving from the F-host is to be assigned to I-PI variables. The F-module F_32IX_IWA2_32OX_OWA2 is used for this purpose.

Possible I/O mapping (example):



—●— I/O mapping

10.7.7 Address format of PROFIsafe addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

10.7.7.1 Address format of send data

The addresses of PROFIsafe send data (O-data of virtual PROFIsafe submodules) are structured as follows:

<Device name>.ProfibusDP.<Virtual slot>.<Virtual subslot>.OutputData[<Element number>]

Examples:

► **Address format for an 8O_SAFEBOOL:**

```
myDevice_1.ProfibusDP.1.4.OutputData[0]
myDevice_1.ProfibusDP.1.4.OutputData[1]
myDevice_1.ProfibusDP.1.4.OutputData[2]
myDevice_1.ProfibusDP.1.4.OutputData[3]
myDevice_1.ProfibusDP.1.4.OutputData[4]
myDevice_1.ProfibusDP.1.4.OutputData[5]
myDevice_1.ProfibusDP.1.4.OutputData[6]
myDevice_1.ProfibusDP.1.4.OutputData[7]
```

► **Address format for an O_SAFEBYTE:**

```
myDevice_1.ProfibusDP.1.5.OutputData
```

► **Address format for an O_SAFEWORD:**

```
myDevice_1.ProfibusDP.1.6.OutputData
```

► **Address format for an O_SAFEDWORD:**

```
myDevice_1.ProfibusDP.4.3.OutputData
```

► **Address format for an O_SAFEINT:**

```
myDevice_1.ProfibusDP.4.4.OutputData
```

10.7.7.2 Address format of receive data

The addresses of PROFIsafe receive data (I-data of virtual PROFIsafe submodules) are structured as follows:

<Device name>.ProfibusDP.<Virtual slot>.<Virtual subslot>.InputData[<Element number>]

Examples:

► **Address format for an 8I_SAFEBOOL:**

```
myDevice_1.ProfibusDP.1.0.InputData[0]
myDevice_1.ProfibusDP.1.0.InputData[1]
myDevice_1.ProfibusDP.1.0.InputData[2]
myDevice_1.ProfibusDP.1.0.InputData[3]
myDevice_1.ProfibusDP.1.0.InputData[4]
myDevice_1.ProfibusDP.1.0.InputData[5]
myDevice_1.ProfibusDP.1.0.InputData[6]
myDevice_1.ProfibusDP.1.0.InputData[7]
```

► **Address format for an I_SAFEBYTE:**

```
myDevice_1.ProfibusDP.1.1.InputData
```

► **Address format for an I_SAFEWORD:**

```
myDevice_1.ProfibusDP.1.2.InputData
```

► **Address format for an I_SAFEDWORD:**

```
myDevice_1.ProfibusDP.4.0.InputData
```

► **Address format for an I_SAFEINT:**

```
myDevice_1.ProfibusDP.4.1.InputData
```

10.7.8 Deactivation and reactivation of an F-module

Send data that is transmitted to an F-module can be brought to a safe condition selectively via the F-host. This is called deactivation/reactivation of an F-module.

► Deactivation of an F-module

- The VALID bit becomes FALSE (see [Validity of process data](#)  359).
- The I-submodules of an F-module are set to their safe values (= zero).

► Reactivation of an F-module

- The VALID bit becomes TRUE, provided there is no PROFIsafe module error.
- The I-submodules of an F-module contain their process value.

10.7.9 F-parameters

F-parameters are available to adapt the PROFIsafe protocol to the respective F-device.

The F-parameters are part of the GSD file.

During the network configuration, the Master Tool (e.g. Siemens Software HW Config) can be used to display and configure the F-parameters and to download them to the F-device.

The F-host uses the F-parameters to check the connection with an F-device. For various F-parameters, indication will be provided if the content of the F-parameter on the F-host does not match that of the F-device (see [PROFIsafe diagnostics](#) [ 311]).

As an F-Device, a PSS 4000 device supports the following F-parameters:

F-parameters	Default configuration	Configuration in the Master Tool	Meaning
F_Check_SeqNr	1	Not configurable	The sequence number is included in the consistency check (CRC calculation).
F_Check_iPar	No Check	Not configurable	No iParameters
F_SIL	SIL3	Not configurable	SIL of F-device
F_CRC_Length	3 Octets	Not configurable	Length of the CRC2 key
F_Block_ID	No F_iPar_CRC	Not configurable	Type identification for parameters
F_Par_Version	V2 Mode	Not configurable	PROFIsafe version implemented within F-device
F_Source_Add	1	Value range: 1 ... 65534	F-address of F-host (automatically assigned by the Master Tool)
F_Dest_Add	1	Value range: 1 ... 65534	F-address of F-device
F_WD_Time	200 ms (because default t_{ExtCo} of PROFIBUS-DP Slave with PROFIsafe = 100 ms)	Value range: F_WD_Time_min ... 65535	Monitoring time in F-device; A valid, current safety telegram must arrive from the F-host within the monitoring time F_WD_Time. If the monitoring time is maintained, safe process data will be delivered to the other system sections via PROFIsafe. If the monitoring time is exceeded, data transfer via PROFIsafe is stopped and the substitute values are used for the affected process data with Valid Bits = FALSE. F_WD_Time must correspond to min. 2 x the cycle time (t_{ExtCo}) of PROFIBUS-DP Slave with PROFIsafe. F_WD_Time_min depends on the device.

10.7.10 PROFIsafe diagnostics

PROFIsafe diagnostics can be performed via the PSS 4000 system diagnostics (see [Diagnostic basics](#)  722]).

However, the standard defines proprietary diagnostic options for PROFIBUS-DP. The current diagnostic information can be polled by a PROFIBUS-DP Master on the PROFIBUS-DP Slave via a diagnostic telegram.

The diagnostic telegram of a PSS 4000 device with PROFIBUS-DP Slave with PROFIsafe system section contains the diagnostic data from the PROFIBUS diagnostics in the first 12 Bytes. It is divided into 2 blocks, each with 6 Bytes (see [PROFIBUS diagnostics](#)  287]). If diagnostic information is available from the PROFIsafe diagnostics, a block will follow with an additional 5 Bytes.

Structure of the diagnostic telegram:

Block 1	Byte 1 ... 6	System-internal diagnostic data see PROFIBUS-DP Slave – System-internal diagnostics  288]	PROFIBUS diagnostics
Block 2	Byte 7 ... 12	Device-based diagnostic data see PROFIBUS-DP Slave – External diagnostics  289]	
Block 3	Byte 13 ... 17	Device-based diagnostics of an F-device	PROFIsafe diagnostics

The diagnostic information is part of the GSD file.

10.7.10.1 Device-based diagnostics of an F-device

F-devices/F-modules can register diagnostic information via the diagnostic mechanisms defined by PROFIBUS-DP. As an F-device, a PSS 4000 device sends 5 Bytes with device-based diagnostic data in the diagnostic telegram.

Structure and content of device-based diagnostic data on an F-device

Byte No.	Bit No.	Value	Meaning
n	0 ... 7	0x05	Header
			Bit 7 and Bit 6: 00 PSS 4000 device with "PROFIBUS-DP Slave with PROFIsafe" system section: Code for device-based diagnostics
			Bit 5 = 0: 000101 Number of bytes in the device-based diagnostics (including Header Byte) PSS 4000 devices with "PROFIBUS-DP Slave with PROFIsafe" system section send 5 Bytes
n + 1	0 ... 7	0x81	Status_Type
			Bit 7: 1 Diagnostic identifier for "Status"
			Bit 6 = 0: 0000001 Code for "Status message" status type
n + 2	0 ... 7	0 ... 254	F-module slot from the perspective of the PROFIsafe F-host (see Pre-defined F-modules [297])
n + 3	0 ... 7	0x01 / 0x02	0x01 Status message present/active
			0x02 No status message present/active
n + 4	0 ... 7	0x40 ... 0x47	PROFIsafe alarms (see [1])

[1] One PSS 4000 device supports the following PROFIsafe alarms:

Hex	Number	Diagnostic information
0x40	64	Mismatch of safety destination address (F_Dest_Add) The value entered in the F-host for the parameter "F_Dest_Add" does not match the local PROFIsafe address.
0x41	65	Safety destination address not valid (F_Dest_Add) The value entered in the F-host for the parameter "F_Dest_Add" is invalid. Valid entry in the F-host: 1 ... 65534
0x42	66	Safety source address not valid (F_Source_Add) The value entered in the F-host for the parameter "F_Source_Add" is invalid. Valid entry in the F-host: 1 ... 65534

Hex	Number	Diagnostic information
0x43	67	Safety watchdog time value is 0 ms (F_WD_Time) The value entered in the F-host for the parameter "F_WD_Time" (0 ms) is invalid. Valid entry in the F-host [ms]: 1 ... 65534
0x44	68	Parameter "F_SIL" exceeds SIL from specific device application The value entered in the F-host for the parameter "F_SIL" is greater than the max. SIL on PSS 4000 (SIL 3). Valid entry in the F-host (max): "SIL3"
0x45	69	Parameter "F_CRC_Length" does not match the generated values The value entered in the F-host for the parameter "F_CRC_Length" is not supported. Valid entry in the F-host: "3 octet CRC"
0x46	70	Version of F-Parameter set incorrect The value entered in the F-host for the parameter "F_Par_Version" is not supported. Valid entry in the F-host: "V2 Mode"
0x47	71	CRC1-Fault The "CRC1" check sums in the F-host and virtual F-module do not match.

10.7.11 Data consistency on PROFI-safe

The F-host is responsible for the data consistency of the receive data sent by the F-host. From the PSS 4000's perspective, no statement can be made regarding the data consistency of the F-host.

Data consistency of the receive data

From the perspective of the PSS 4000, the receive data of an F-module is external I-data. As soon as external I-data is available in an F-module, it is treated as consistent data (see [Data consistency](#)  377). The external I-data of a virtual F-module forms a consistency range of the consistency range type "FS-I data of an external communication connection". The exact description is consistency range "FS-I data of a virtual PROFI-safe F-module" (see [Consistency ranges](#)  378).

Data consistency of the send data

From the perspective of the PSS 4000, the send data of an F-module is external O-data. The external O-data of an F-module is consistent if it comes from the same consistency range on a data source (see [Data consistency](#)  377). PSS 4000 always treats the external O-data of an F-module as one unit. It is the task of the F-host to guarantee the necessary data consistency.

10.8 EtherNet/IP Adapter

EtherNet/IP (Ethernet Industrial Protocol, abbreviation EIP) is an Ethernet-based fieldbus. With EtherNet/IP an open industrial standard has been defined, which expands the classic Ethernet with an industrial protocol (Common Industrial Protocol, abbreviation CIP), as an application layer for automation applications.

The open industrial standard EtherNet/IP was drawn up jointly by ControlNet International (CI) and the Open DeviceNet Vendor Association (ODVA), with help from the Industrial Ethernet Association (IEA). EtherNet/IP is standardised in the international IEC 61158 series of standards.

Common Industrial Protocol (CIP)

Implicit and explicit messages are used to transfer CIP message data in EtherNet/IP networks.

Implicit messages are typically smaller data packets containing I/O data for time-critical data transfers. Implicit messages are transferred via UDP.

Non-time-critical messages are transferred via explicit messages. Explicit messages contain configuration and information data, for example. They use the TCP/IP communication mechanism.

CIP is object-oriented. ODVA uses device profiles to define which objects and attributes must be supported by devices of a certain device class. Optional and manufacturer-defined objects and attributes are also possible. CIP uses an electronic data sheet, (abbreviation EDS) as a common basis for configuration.

Further information on the Common Industrial Protocol (CIP) or on EtherNet/IP is available from the ODVA (www.odva.org).

EtherNet/IP-enabled PSS 4000 devices from Pilz

For **non**-safety-related applications, an EtherNet/IP-enabled PSS 4000 device from Pilz can be incorporated as an adapter in EtherNet/IP.

An EtherNet/IP-enabled PSS 4000 device from Pilz supports the following as an EtherNet/IP Adapter:

- ▶ Communications Adapter device profile, device type 12₀ in accordance with the "Common Industrial Protocol Specification", Volume 1 of the ODVA.
Development and conformance test have been carried out in accordance with Edition 3.12, April 2012.
- ▶ Cyclical data exchange with an EtherNet/IP scanner
 - I/O Server functionality for I/O data from implicit messages
 - The I/O data is transferred via UDP and uses Port 2222.
- ▶ Acyclical data exchange as "Explicit Message Server"
 - Explicit Message Server functionality for explicit data
 - Support for acyclical data exchange with the following objects: Identity Object, TCP/IP Object and Link Object
- ▶ Identity Object
- ▶ TCP/IP Object
- ▶ Link Object

- ▶ Assembly Object with
 - An Output Assembly Object instance
 - An Input Assembly Object instance

10.8.1 "EtherNet/IP Adapter" system section

In the automation system PSS 4000, external communication via EtherNet/IP is possible with all PSS 4000 devices that have the "EtherNet/IP Adapter" system section. In this case, the EtherNet/IP Adapter system section of a PSS 4000 device performs all the tasks required for communication. These include:

- ▶ Administration of the supported objects
(see [Supported EtherNet/IP objects and Common Services](#)  331)
- ▶ Data exchange via EtherNet/IP
(see [Data exchange via EtherNet/IP](#)  319)

10.8.2 EtherNet/IP Adapter connections

Precisely one EtherNet/IP Adapter connection is currently supported per PSS 4000 device (Adapter). With this EtherNet/IP Adapter connection, data exchange can occur via precisely one Output Assembly Object instance and precisely one Input Assembly Object instance (see [Assembly Object](#)  334]).

10.8.3 Data exchange via EtherNet/IP

PSS 4000 devices with EtherNet/IP Adapter system section support data exchange as EtherNet/IP Adapters. It is always the EtherNet/IP scanner that initiates data transfer. From the perspective of the automation system PSS 4000, the EtherNet/IP scanner is outside the system.

10.8.3.1 Send data and receive data

With EtherNet/IP, the actual usable data (process data) is exchanged via the Assembly Object. A PSS 4000 device with EtherNet/IP Adapter system section supports the Assembly Object with one Output Assembly Object instance and one Input Assembly Object instance respectively. On the PSS 4000, the EtherNet/IP terms "Input" and "Output" are defined from the scanner's perspective. The advantage of this definition is that the views are identical in both the scanner tool and in PAS4000; i.e. the Output Assembly Object instance in the scanner tool is also the Output Assembly Object instance in PAS4000 and the Input Assembly Object instance in the scanner tool is also the Input Assembly Object instance in PAS4000.

To illustrate this point, on PSS 4000 the Output Assembly Object instance is called the "Scanner Output Assembly Object instance" and the Input Assembly Object instance is called the "Scanner Input Assembly Object instance".

Scanner Output Assembly Object instance

The Scanner Output Assembly Object instance contains the data that the scanner produces and places on the network; i.e. it contains the data received by a PSS 4000 device (Adapter). The data length of the data transferred using the Scanner Output Assembly Object instance must be configured equally in both the scanner and the PSS 4000 device. For the PSS 4000 device, this configuration is made in the EtherNet/IP Adapter Editor of PAS4000. As the Scanner Output Assembly Object instance contains receive data (= external I-data) from the perspective of the PSS 4000 device, virtual I-modules must be configured in the EtherNet/IP Adapter Editor. Virtual I-modules have an "I" for "Input" in the module name (e.g. 16IX, IB, IW, ID).

A PSS 4000 device expects the scanner to always send the run/idle header.

Scanner Input Assembly Object instance

The Scanner Input Assembly Object instance contains the data that the scanner consumes from the network; i.e. it contains the data sent by a PSS 4000 device (Adapter). The data length of the data transferred using the Scanner Input Assembly Object instance must be configured equally in both the scanner and the PSS 4000 device. For the PSS 4000 device, this configuration is made in the EtherNet/IP Adapter Editor of PAS4000.

As the Scanner Input Assembly Object instance contains send data (= external O-data) from the perspective of the PSS 4000 device, virtual O-modules must be configured in the EtherNet/IP Adapter Editor. Virtual O-modules have an "O" for "Output" in the module name (e.g. 16OX, OB, OW, OD).

A status word for sending PSS 4000 status information to the scanner can be configured as an option.

10.8.3.2 Virtual modules for send data

From the perspective of the PSS 4000 device (Adapter), the Scanner Input Assembly Object instance contains send data (= external O-data). The following virtual modules can be configured in PAS4000 for the send data of a PSS 4000 device:

Virtual module	Data type	Application
16OX	ARRAY [0..15] OF BOOL	Virtual EtherNet/IP bit module To send an array with 16 elements of data type BOOL
OBAn	ARRAY [0...(n-1)] OF BYTE	Virtual EtherNet/IP byte module To send an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
OW	WORD	Virtual EtherNet/IP byte module To send data of data type WORD
OWAn	ARRAY [0...(n-1)] OF WORD	Virtual EtherNet/IP byte module To send an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64
OD	DWORD	Virtual EtherNet/IP byte module To send data of data type DWORD



INFORMATION

Note that only the virtual modules documented here with the specified data types are supported. The LWORD data type is not supported.

The data length of the data transferred using the Scanner Input Assembly Object instance must be configured equally in both the scanner and the PSS 4000 device.

10.8.3.3 Virtual modules for receive data

From the perspective of the PSS 4000 device (Adapter), the Scanner Output Assembly Object instance contains receive data (= external I-data). The following virtual modules can be configured in PAS4000 for the receive data of a PSS 4000 device:

Virtual module	Data type	Application
16IX	ARRAY [0..15] OF BOOL	Virtual EtherNet/IP bit module To receive an array with 16 elements of data type BOOL
IBAn	ARRAY [0...(n-1)] OF BYTE	Virtual EtherNet/IP byte module To receive an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
IW	WORD	Virtual EtherNet/IP byte module To receive data of data type WORD
IWAn	ARRAY [0...(n-1)] OF WORD	Virtual EtherNet/IP byte module To receive an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64
ID	DWORD	Virtual EtherNet/IP byte module To receive data of data type DWORD



INFORMATION

Note that only the virtual modules documented here with the specified data types are supported. The LWORD data type is not supported.

The data length of the data transferred using the Scanner Output Assembly Object instance must be configured equally in both the scanner and the PSS 4000 device.

10.8.4 Data transfer limits

When transferring data via EtherNet/IP, the following limits apply for a PSS 4000 device (Adapter):

- ▶ A maximum of one connection (EtherNet/IP Adapter connection) to the scanner can be configured per PSS 4000 device.
- ▶ A PSS 4000 device supports a maximum of one Scanner Output Assembly Object instance and one Scanner Input Assembly Object instance.
- ▶ Virtual I-modules (external I-data) with a total data length of 0 ... 504 Bytes can be configured for a Scanner Output Assembly Object instance.

The total data length of a Scanner Output Assembly Object instance is a maximum of 510 Bytes (504 Bytes for virtual I-modules, 4 Bytes for the run/idle header and 2 Bytes for the Sequence Count).

- ▶ Virtual O-modules (external O-data) with a total data length of 0 ... 508 Bytes can be configured for a Scanner Input Assembly Object instance.

The total data length of a Scanner Input Assembly Object instance is a maximum of 510 Bytes (508 Bytes for virtual O-modules, 2 Bytes for the Sequence Count).

10.8.5 EtherNet/IP Adapter I/O data

The "EtherNet/IP Adapter" system section enables a PSS 4000 device to exchange I/O data with the EtherNet/IP scanner via EtherNet/IP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding virtual modules.

I/O-mapping of virtual modules

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O-mapping for virtual modules:

I/O mapping		on virtual modules	
		Data of the Scanner Output Assembly Object instance (= external I-data of I-modules such as 16IX, IBAn, IWAn, ID)	Data of the Scanner Input Assembly Object instance (= external O-data of O-modules such as 16OX, OBAn, OWAn, OD)
of module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
of PI-variable	I-PI variable	◆	---
	O-PI variable	---	◆

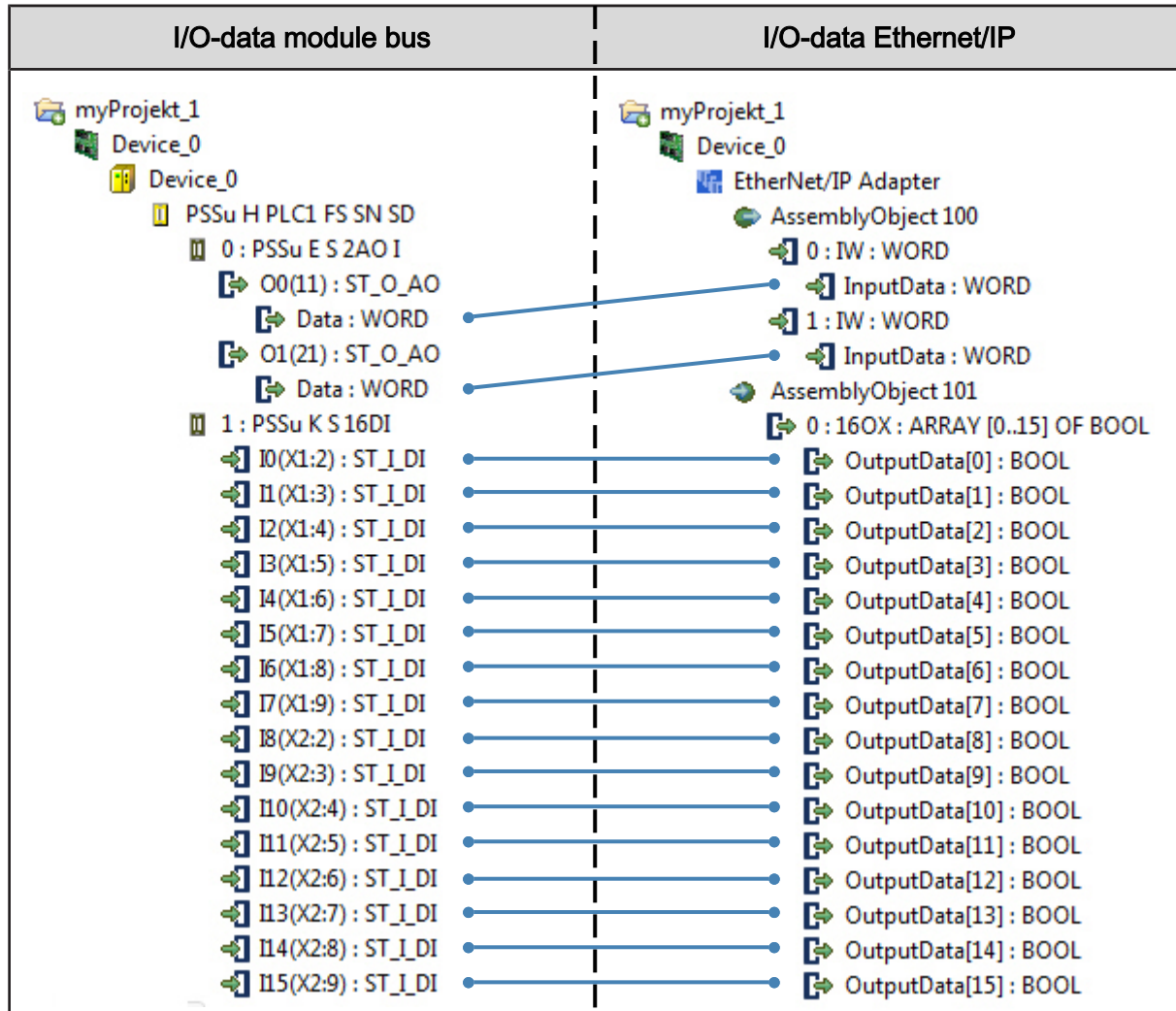
- ◆ I/O mapping possible
 --- I/O mapping impossible

Examples

► I/O mapping of I/O data

The input signals from hardware inputs on a PSSu K S 16DI are to be transmitted to the scanner and the scanner is to control the hardware outputs on the PSSu E S 2 AO I.

Possible I/O mapping (example):

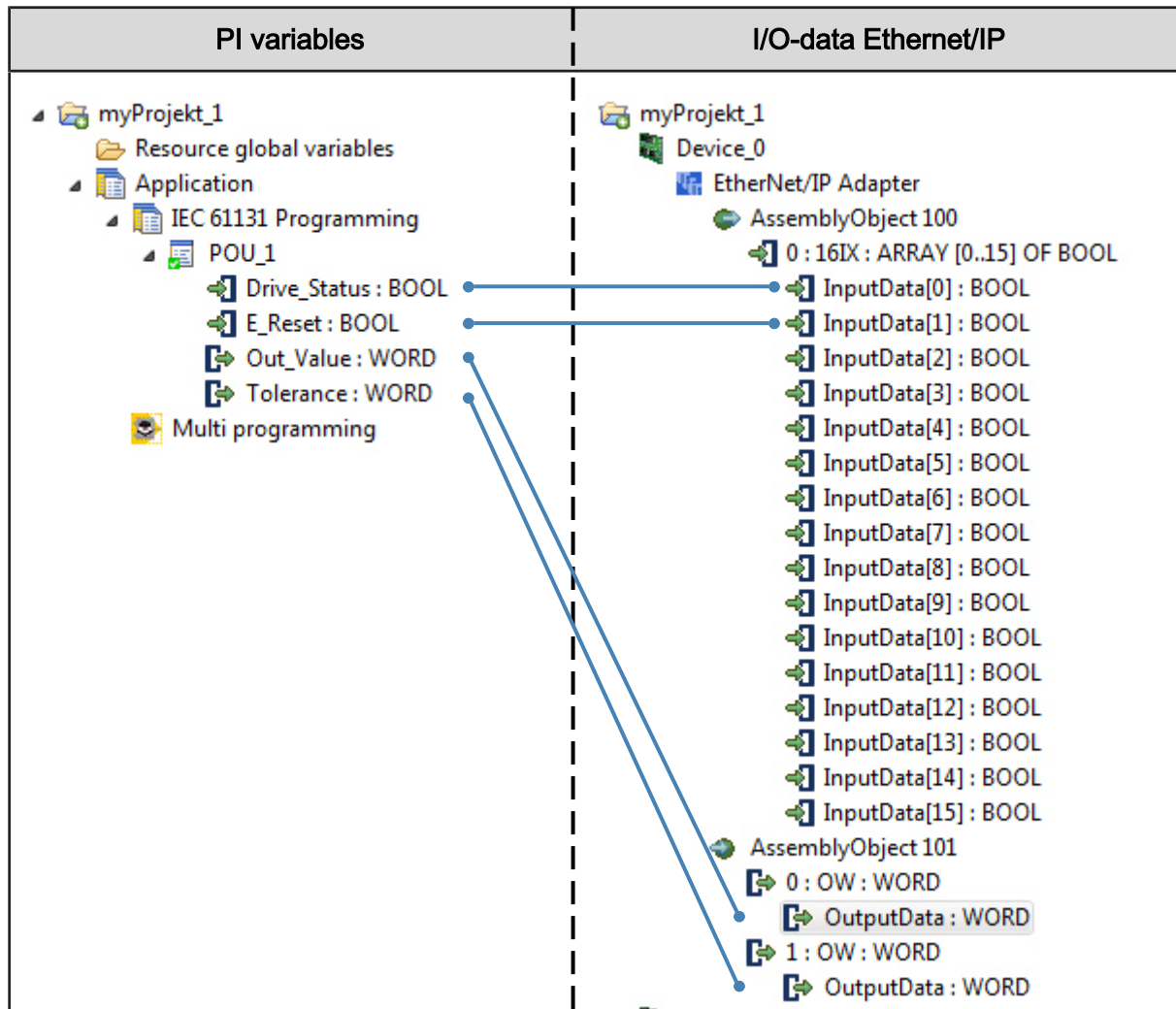


— I/O mapping

► I/O mapping of PI variables

The contents of a block's O-PI variables are to be transmitted to the scanner and I-PI variables are to be assigned to the receive data.

Possible I/O mapping (example):



●—● I/O mapping

10.8.6 Address format of EtherNet/IP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

10.8.6.1 Address format of send data

The addresses of EtherNet/IP send data (O-data of virtual modules) are structured as follows:

<Device Name>.EtherNetIP.AObj<Instance Number>.<Virtual Slot>.OutputData[<Element Number>]

Examples:

► Address format for a 16OX:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[0]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[1]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[2]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[3]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[4]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[5]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[6]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[7]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[8]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[9]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[10]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[11]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[12]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[13]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[14]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[15]
```

► Address format for an OBA2:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule4.OutputData[0]
myDevice_1.EtherNetIP.AObj101.VirtualModule4.OutputData[1]
```

► Address format for an OW:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule5.OutputData
```

► Address format for an OWA4:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule6.OutputData[0]
myDevice_1.EtherNetIP.AObj101.VirtualModule6.OutputData[1]
myDevice_1.EtherNetIP.AObj101.VirtualModule6.OutputData[2]
myDevice_1.EtherNetIP.AObj101.VirtualModule6.OutputData[3]
```

► Address format for an OD:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule7.OutputData
```

10.8.6.2 Address format of receive data

The addresses of EtherNet/IP receive data (I-data of virtual modules) are structured as follows:

<Device Name>.EtherNetIP.AObj<Instance Number>.<Virtual Slot>.InputData[<Element Number>]

Examples:

► Address format for a 16IX:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[0]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[1]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[2]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[3]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[4]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[5]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[6]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[7]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[8]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[9]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[10]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[11]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[12]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[13]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[14]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[15]
```

► Address format for an IBA2

```
myDevice_1.EtherNetIP.AObj100.VirtualModule9.InputData[0]
myDevice_1.EtherNetIP.AObj100.VirtualModule9.InputData[1]
```

► Address format for an IW:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule10.InputData
```

► Address format for an IWA4:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule11.InputData[0]
myDevice_1.EtherNetIP.AObj100.VirtualModule11.InputData[1]
myDevice_1.EtherNetIP.AObj100.VirtualModule11.InputData[2]
myDevice_1.EtherNetIP.AObj100.VirtualModule11.InputData[3]
```

► Address format for an ID:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule12.InputData
```

10.8.7 EtherNet/IP diagnostics

As an option, the transmission of a status word can be configured in PAS4000 for the Scanner Input Assembly Object instance. The status word contains device-based status information and is transmitted to the scanner for diagnostic purposes.

Structure and content of the status word

Octet 1 contains the error messages from the PSS 4000 device with EtherNet/IP Adapter system section:

Bit	Value	Meaning
0	0/1	Error message for the device
		0 No message
		1 ▶ A message of "Error" severity is present for at least one system section of the device - or ▶ A major FS error is present for at least one FS system section of the device On PSS 4000-exclusive devices, this corresponds to the "DIAG" status LED: Status: "lights up red" or "flashes red".
1	0/1	Error in the device's FS/ST module bus system sections
		0 No message
		1 ▶ Operating state "Safe state of all FS outputs on the device" - or ▶ At least one module cannot be accessed (e.g. a module has been removed during operation, actual/registered hardware does not match) or ▶ Operating state "FS module bus in a STOP state with error: Major FS error" On PSS 4000-exclusive devices, this corresponds to the "MBUS" status LED: Status: "lights up red" or "flashes red".
2	0/1	Error in the "FS SafetyNET p" system section of the device
		0 No message
		1 ▶ Operating state "FS SafetyNET p in STOP state with error: Major FS error" - or ▶ Operating state "FS SafetyNET p in STOP state with error: Major FS+ST error" On PSS 4000-exclusive devices, this corresponds to the "FS SNp" status LED: Status: "lights up red" or "flashes red".

Bit	Value	Meaning
3	0/1	Error in the "FS SafetyNET p" system section of the device
		0 No message
		1 Operating state "ST SafetyNET p in STOP state with error: Major FS+ST error" On PSS 4000-exclusive devices, this corresponds to the "ST Snp" status LED: Status: "lights up red" or "flashes red".
4	0	Reserved
5	0	Reserved
6	0	Reserved
7	0	Reserved

Octet 2 contains additional status information from the PSS 4000 device with EtherNet/IP Adapter system section:

Bit	Value	Meaning
0	0/1	Warning message for the device
		0 No message
		1 A message of "Warning" severity is present for the device On PSS 4000-exclusive devices, this corresponds to the "DIAG" status LED: Status: "lights up orange".
1	0/1	Forcing on the FS resource
		0 No message
		1 Forcing on the FS resource is active On PSS 4000-exclusive devices, this corresponds to the "FS FORCE" status LED: Status: "lights up yellow".
2	0/1	Forcing on the ST resource
		0 No message
		1 Forcing on the ST resource is active On PSS 4000-exclusive devices, this corresponds to the "ST FORCE" status LED: Status: "lights up yellow".
4	0	Reserved
5	0	Reserved
6	0	Reserved
7	0	Reserved

10.8.8 Data consistency on EtherNet/IP

The scanner is responsible for the data consistency of the receive data (Scanner Output instance of the Assembly Object), which is sent by the scanner via an EtherNet/IP connection. From the PSS 4000's perspective, no statement can be made regarding the data consistency of the scanner.

Data consistency on receive data via the Scanner Output Assembly Object instance

If a PSS 4000-exclusive/operable device is the recipient of data from an EtherNet/IP connection, from the perspective of the device the receive data is external I-data.

External I-data from an EtherNet/IP connection is treated as consistent data in PSS 4000. The external I-data from an EtherNet/IP connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of an EtherNet/IP Assembly Object instance".

Data consistency on send data via the Scanner Input Assembly Object instance

If a PSS 4000-exclusive/operable device in an EtherNet/IP connection is the transmitter of data, from the perspective of the device the send data is external O-data.

The external O-data of an EtherNet/IP connection is consistent if it comes from the same consistency range on a data source.

10.8.9 Supported EtherNet/IP objects and Common Services

For data exchange via EtherNet/IP, an EtherNet/IP subscriber is composed of a collection of CIP objects. CIP objects are divided into classes, instances and attributes.

PSS 4000 devices with EtherNet/IP Adapter system section support the following object classes as Adapters.

Class ID	Name of Object Class
0x01	Identity Object
0x02	Message Router Object
0x04	Assembly Object
0x06	Connection Manager Object
0xF4	Port Object
0xF5	TCP/IP Interface Object
0xF6	(Network Specific) Link Object

Appropriate diagnostic tools (e.g. tools from Rockwell) can be used to display the content of attributes. Various [Common Services](#)  [337](#) are supported.

10.8.9.1 Identity Object (Class ID 0x01)

The Identity Object is available on each PSS 4000 device with EtherNet/IP Adapter system section. It contains the device identification and other general information about the device.

Instance attributes of the Identity Object instance

Attribute ID	Name	Value	Description
1	Vendor ID	181 (0x00B5)	Manufacturer identification from Pilz
2	Device Type	12 (0x000C)	General product type description (12 = Communications Adapter)
3	Product Code	Default: 1 (0x0001)	Product's EtherNet/IP product code (e.g. 2070 for PSSu H PLC1 FS SN SD)
4	Revision	- - -	Version of the product's PSS 4000 firmware <Major-No>.<Minor-No> (e.g. 1.8) The 3rd digit of the version number is not displayed.
	Major Revision	<Major-No>	
	Minor Revision	<Minor-No>	

Attribute ID	Name	Value		Description
5	Status	Bit	Contents	Current status of product
		0	0	Not used
		1	0	Reserved
		2	1	EtherNet/IP has been configured
		3	0	Reserved
		4 to 7	0	Not used
			1	
			2	At least one EtherNet/IP connection is faulty
			3	No EtherNet/IP connection has been established
			4	Not used
			5	A major FS error is present for at least one FS system section (see diagnostic list) or A message of "Error" severity is present for at least one system section (see diagnostic list)
			6	At least one EtherNet/IP connection is in a RUN state
			7	At least one EtherNet/IP connection has been established; all EtherNet/IP connections are idle
			8, 9	Reserved
			10 to 15	Not used
		8	0	No message of "Warning" severity is present for the device.
			1	A message of "Warning" severity is present for the device (see diagnostic list)
		9	0	Not used
		10	0	There is no major FS error present on any FS system section and there is no message of "Error" severity present on any system section.
			1	A major FS error is present for at least one FS system section (see diagnostic list) or A message of "Error" severity is present for at least one system section (see diagnostic list)
		11	0	Not used
		12 to 15	0	
6	Serial number	<Serial No.>		Product serial number (e.g. serial number of head module on PSSu systems)
7	Product Name	Default: PSS 4000		Product's EtherNet/IP product name (e.g. PSSu H PLC1 FS SN SD)

10.8.9.2 Assembly Object (Class ID 0x04)

Data exchange between the scanner and adapter occurs via the Scanner Output Assembly Object instance and the Scanner Input Assembly Object instance of the Assembly Object.

Attributes of the Scanner Output Assembly Object instance

Attribute ID	Name	Value	Description
1	Attributes are not supported		
2			
3	Data	<Data>	The attribute contains the data that the scanner produces and places on the network in accordance with the configuration. From the perspective of the PSS 4000 device, this is the external I-data.
4	Size	<Data Length>	Data length in accordance with the configuration

Attributes of the Scanner Input Assembly Object instance

Attribute ID	Name	Value	Description
1	Attributes are not supported		
2			
3	Data	<Data>	The attribute contains the data that the scanner consumes from the network in accordance with the configuration. From the perspective of the PSS 4000 device, this is the external O-data.
4	Size	<Data Length>	Data length in accordance with the configuration

10.8.9.3 TCP/IP Interface Object (Class ID 0xF5)

The TCP/IP Interface Object can be used to scan a device's TCP/IP properties (IP address, subnet mask, Gateway address etc.). On a PSS 4000 device with EtherNet/IP Adapter system section, all TCP/IP properties contained in the Naming data are supported. So the TCP/IP Interface Object of a PSS 4000 device with EtherNet/IP Adapter system section obtains this configuration via the naming data during device naming.

Attributes of the TCP/IP Interface Object

Attribute ID	Name	Value	Description
1	Status	0	No data exchange configured via EtherNet/IP
		1	Data exchange configured via EtherNet/IP
2	Configuration capability	32 (0x00000020)	Configuration of the TCP/IP properties via device naming
3	Configuration control	0	Configuration of the TCP/IP properties via device naming is static
4	Physical Link Object	<Path to Physical Link Object>	The device has exactly one instance of the Physical Link Object; the attribute contains the path to this instance of the Physical Link Object
		0	The device has more than one instance of the Physical Link Object
5	Interface configuration	<IP address, subnet mask, Gateway address etc.>	TCP/IP properties in accordance with the naming data The attribute is "read only"; i.e. the Set-Service is not supported.
6	Host name	<Device name>	Device name from device naming The attribute is "read only"; i.e. the Set-Service is not supported.
		0	No host is configured and no host can be configured
7	Attribute is not supported		
8	TTL Value	1	Time To Live (TTL) for EtherNet/IP packets via IP-Multicast
9	Mcast Config	- - -	Configuration of the IP-Multicast addresses that are used for EtherNet/IP Multicast packets; Overall data length of attribute: 8 bytes
	Alloc control (1 Byte)	0	The IP-Multicast addresses are assigned via the specific default allocation algorithm.
		1	The IP-Multicast addresses are assigned using "Num Mcast" and "Mcast Start Addr".
		2	Reserved
	Reserved (1 Byte)		
	Num Mcast (2 Bytes)	<Device-dependent>	Number of assigned IP-Multicast addresses; Start address see "Mcast Start Addr"
	Mcast Start Addr (4 Bytes)	<Device-dependent>	Start address for the number of IP-Multicast addresses defined under "Num Mcast"

10.8.9.4 Link Object (Class ID 0xF6)

The Link Object contains information about the status of the physical interface assigned to it.

Attribute of Link Object

Attribute ID	Name	Value		Description	
1	Interface Speed	0		Poor connection or no connection	
		<Transmission rate>		Current transmission rate (10, 100, 1000, 10000 MBit/s)	
2	Interface Status Flags	Bit	Content	- - -	
		0	0	No Ethernet connection available	
			1	Ethernet connection available	
		1	0	Half duplex mode	
			1	Full duplex mode	
		2 to 4	0	Performing autonegotiation	
			1	Autonegotiation failed	
			2	Autonegotiation failed but transmission rate was detected	
			3	Autonegotiation was successful	
		5 to 6	4	Autonegotiation not performed	
			5	0	Not supported
			6	0	No hardware error found
		1		Hardware error detected	
3	Physical Address	<MAC Address>		Device's MAC address	
4, 5, 6	Attributes are not supported				
7	Interface Type	0		Unknown	
		1		Internal (not supported)	
		2		Twisted Pair	
		3		Fibre-optic cable	
8	Interface State	0		Unknown	
		1		Interface is activated and ready to send/receive data	
		2		Deactivated	
		3		Test mode	
9	Attribute is not supported				
10	Interface Label	<Interface name>		Text that names the interface (e.g. "Port X3")	

10.8.9.5 Common Services

Common Services are supported for the class and instance attributes as follows:

► **Get Services**

The contents of attributes with the Get-Services **Get_Attribute_All** (Service Code 0x01) and **Get_Attribute_Single** (Service Code 0x0E) can be polled for diagnostic purposes. Tools from third-party manufacturers (e.g. tools from Rockwell) support these Services. If non-supported attributes are accessed, the error code for "Attribute not supported" (0x14) is issued.

► **Set Services**

Set Services are supported for the following attributes:

- Scanner Output Assembly Object instance - Attribute 3
Set Services are available for Attribute 3, provided it is not participating in a cyclical connection.
- Connection Manager Object – Attributes 1 ... 8

► **Reset Service**

If the Reset Service is called by the EtherNet/IP Scanner using Service Code 0x00 or 0x01, all the EtherNet/IP Adapter connections on the relevant PSS 4000 device will be re-set. An entry is added to the diagnostic log and diagnostic list.

10.9 PROFINET IO DEVICE

PROFINET is an open fieldbus standard based on Industrial Ethernet. PROFINET communication is defined in the international standards IEC 61158, IEC 61158-5-10 (Application Layer Service Definition), IEC 61158-6-10 (Application Layer Protocol Specification), IEC 61784-1 Type 10 (Communication Profiles) and IEC 61784-2 (PROFINET IO). Further provisions have been defined in specifications published by the user organisation PROFIBUS PROFINET International (PI).

PROFINET-enabled PSS 4000 devices from Pilz

For **non**-safety related applications, a PROFINET-enabled PSS 4000 device can be incorporated into a PROFINET project as an IO device. A PROFINET-enabled PSS 4000 device is an IO device of Conformance Class A (CC-A). A PSS 4000 device is incorporated into a PROFINET project via the corresponding GSDML file (XML-based General Station Description file).

In accordance with the specification, a PROFINET IO system consists of at least one IO controller, which controls at least one IO device. A PSS 4000 device can be incorporated in just one PROFINET IO controller; i.e. multiple access (shared device) is not supported.

As a PROFINET IO device, a PSS 4000 device supports the following:

- ▶ IO Data Communication Relations (IO Data CR) for cyclical process data exchange
- ▶ Alarm Communication Relations (Alarm CR) to signal alarms in real-time
- ▶ PROFINET IO diagnostic telegram
- ▶ Data exchange via pre-defined virtual PROFINET modules (I-modules, O-modules)

10.9.1 **"PROFINET IO DEVICE" system section**

In the automation system PSS 4000, external communication via PROFINET is possible with all PSS 4000 devices that have the "PROFINET IO DEVICE" system section. In this case, the PROFINET IO DEVICE system section of a PSS 4000 device performs all the tasks required for communication. These include:

- ▶ Management of the send and receive data areas
- ▶ Data exchange via PROFINET

10.9.2 Incorporating and addressing a device in PROFINET

A PSS 4000 device is incorporated into a PROFINET project via an engineering tool for PROFINET.

During plant configuration, the IO devices are addressed in the engineering tool of the PROFINET controller using MAC addresses, IP addresses and symbolic names (PROFINET device name). The address resolution is based on the symbolic name, to which a unique MAC address is assigned.

After plant configuration, the engineering tool of the PROFINET controller loads all the information required for data exchange into the IO controller; this includes the IP addresses for the connected IO devices. An IO controller can recognise the configured IO devices by the symbolic name and assign the defined IP addresses to them.



NOTICE

A deactivated PROFINET IO device prevents communication with the PROFINET IO controller

Communication between a PROFINET IO controller and a PSS 4000 device is only possible if the PROFINET IO DEVICE system section of the PSS 4000 device has been activated. The PROFINET IO DEVICE system section is activated on a PSS 4000 device by configuring at least one virtual PROFINET module on the device.

Make sure that at least one virtual PROFINET module is configured on the PSS 4000 device. If not, the PROFINET IO controller will be unable to find the PSS 4000 device.

Please note the following for PSS 4000 devices with PROFINET IO DEVICE system section:

► PROFINET device name

- When a PSS 4000 device is incorporated via the engineering tool of the PROFINET controller, no device name is displayed in accordance with PROFINET requirements (default setting).
- The PROFINET device name of a PSS 4000 device is assigned via the engineering tool of the PROFINET controller and is only significant in PROFINET; i.e. the PROFINET device name does not affect the device name configured for the PSS 4000 device during device naming in PAS4000.
- The PROFINET device name of a PSS 4000 device is retained when the device is exchanged, if the PSS 4000 device has an SD card and the SD card is reused on the replacement device.

► PROFINET IP configuration

- Each PSS 4000 device has a separate IP configuration in PROFINET. The PROFINET IP configuration can only be performed via the engineering tool of the PROFINET controller or a DHCP Server.

Default setting: 0.0.0.0 for IP address, subnet mask and (standard) gateway.

- The PROFINET-IP configuration is only valid in PROFINET; i.e. the PROFINET-IP configuration does not affect the IP configuration configured for the PSS 4000 device during device naming in PAS4000.
- The PROFINET-IP configuration can be identical with the IP configuration configured for the PSS 4000 device during device naming in PAS4000. The two IP configurations may also differ, however.

However, please note

If you configure the same IP address in the PROFINET IP configuration (engineering tool of the PROFINET controller) and in the IP configuration of the PSS 4000 device (PAS4000), the subnet mask must be identical in both IP configurations.

- The PROFINET IP configuration is retained when the device is exchanged, if the PSS 4000 device has an SD card and the SD card is reused on the replacement device.

► MAC addresses

For "neighbourhood discovery" via the Link Layer Discovery Protocol (LLDP), PROFINET does not need the device MAC address but the MAC address of the physical (Ethernet) port, through which the device is connected to PROFINET.

- A PSS 4000 device has an integrated switch with 2 physical (Ethernet) ports. Both ports use the two MAC addresses that follow the device MAC address; i.e. a PSS 4000 device has 3 consecutive MAC addresses.
- The device MAC address is on the PSS 4000 device.

10.9.3 Identifying a PSS 4000 device in PROFINET

It is possible to identify a PROFINET IO device in PROFINET. To do this, an enquiry is sent to the required PROFINET IO device via the engineering tool of the PROFINET controller; the device must react to this enquiry. This process is also known as "device identification". If (PROFINET) device identification is running for a PSS 4000 device, the status LED for the SD card is activated. The "SD CARD" status LED on the device flashes orange.



NOTICE

A deactivated PROFINET IO device prevents communication with the PROFINET IO controller

Communication between a PROFINET IO controller and a PSS 4000 device is only possible if the PROFINET IO DEVICE system section of the PSS 4000 device has been activated. The PROFINET IO DEVICE system section is activated on a PSS 4000 device by configuring at least one virtual PROFINET module on the device.

Make sure that at least one virtual PROFINET module is configured on the PSS 4000 device. If not, the PROFINET IO controller will be unable to find the PSS 4000 device.

10.9.4 Data exchange via PROFINET

A PSS 4000 device with PROFINET IO DEVICE system section operates as an IO device in PROFINET. It is always the PROFINET IO controller that initiates data transfer. From the perspective of the automation system PSS 4000, the PROFINET IO controller is outside the system.

10.9.4.1 Sending and receiving data

The PROFINET IO controller sends data from its send data area to the receive data area of the PROFINET IO DEVICE system section of a PSS 4000 device and requests data from the PROFINET IO DEVICE system section of a PSS 4000 device. The PROFINET IO DEVICE system section of a PSS 4000 device sends the requested data from its send data area to the receive data area of the PROFINET IO controller.

In the automation system PSS 4000, the data areas are called "virtual PROFINET modules". Virtual PROFINET modules are available with a variety of data types. Both the PROFINET IO controller and the PROFINET IO DEVICE system section of a PSS 4000 device require corresponding virtual PROFINET modules for data exchange. The virtual PROFINET modules for send data are called O-modules. They have an "O" for "Output" in the module name (e.g. 8OX, 16OX, 32OX, OB, OW, OD).

The virtual PROFINET modules for receive data are called I-modules. They have an "I" for "Input" in the module name (e.g. 8IX, 16IX, 32IX, IB, IW, ID).

A PSS 4000 device is incorporated into a PROFINET project only in the engineering tool of the PROFINET controller. This is where the PROFINET-IP address and PROFINET device name are assigned, for example. The necessary virtual PROFINET modules are made available via the GSDML file (see [GSDML file](#)  347).

In the automation system PSS 4000, the virtual PROFINET modules are configured in the PROFINET IO DEVICE Editor of PAS4000. After configuration, the virtual PROFINET modules are located on virtual slots (virtual slots 0 ... 64).

From the perspective of the automation system PSS 4000, communication via PROFINET is external communication. The receive data is called external I-data and the send data is called external O-data (see [Validity of process data](#)  359 and [Substitute values](#)  362).

10.9.4.2 Virtual PROFINET modules for send data

The following virtual PROFINET modules (O-modules) can be configured in PAS4000 for the send data of the PROFINET IO DEVICE system section:

Virtual PROFINET module	Data type	Application
8OX	ARRAY [0..7] OF BOOL	Virtual PROFINET bit module To send an array with 8 elements of data type BOOL
16OX	ARRAY [0..15] OF BOOL	Virtual PROFINET bit module To send an array with 16 elements of data type BOOL
32OX	ARRAY [0..31] OF BOOL	Virtual PROFINET bit module To send an array with 32 elements of data type BOOL
OB	BYTE	Virtual PROFINET byte module To send data of data type BYTE
OW	WORD	Virtual PROFINET byte module To send data of data type WORD
OD	DWORD	Virtual PROFINET byte module To send data of data type DWORD
OBAn	ARRAY [0..(n-1)] OF BYTE	Virtual PROFINET byte module To send an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
OWAn	ARRAY [0..(n-1)] OF WORD	Virtual PROFINET byte module To send an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64
ODAn	ARRAY [0..(n-1)] OF DWORD	Virtual PROFINET byte module To send an array with n elements of data type DWORD n = 2, 4, 8, 16, 32 or 64



INFORMATION

Note that only the virtual modules documented here with the specified data types are supported. The LWORD data type is not supported.

During data exchange, the PROFINET IO controller requests data from O-modules on the automation system PSS 4000 and receives this data via its I-modules. Consequently, the O-modules on the automation system PSS 4000 must have corresponding I-modules in the PROFINET IO controller. Also, the data type of an I-module in the PROFINET IO controller must correspond to the data type of an O-module in the automation system PSS 4000.

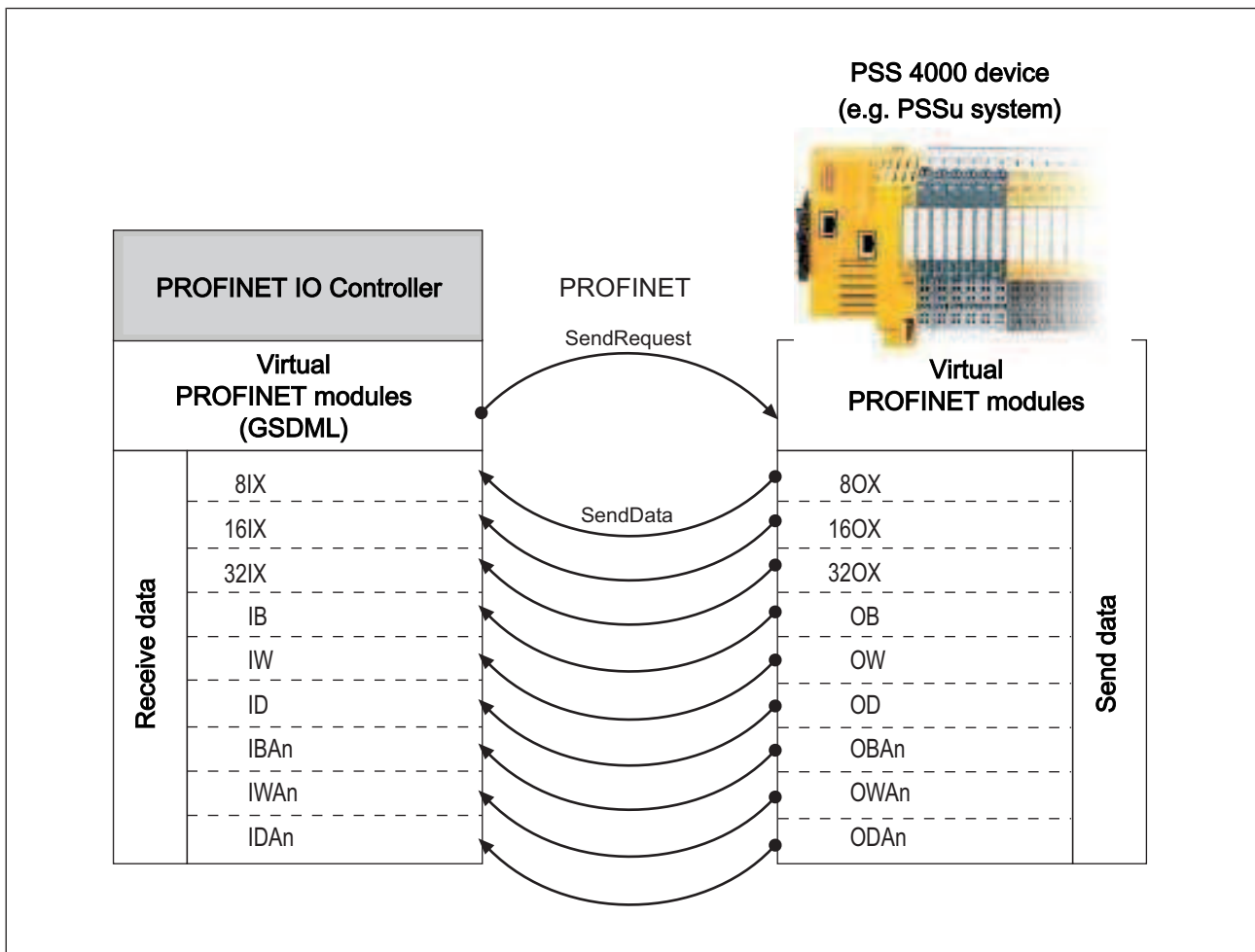


Fig.: Virtual PROFINET modules and their corresponding data types (send data of a PSS 4000 device)

10.9.4.3 Virtual PROFINET modules for receive data

The following virtual PROFINET modules (I-modules) can be configured in PAS4000 for the receive data of the PROFINET IO DEVICE system section:

Virtual PROFINET IO module	Data type	Application
8IX	ARRAY [0..7] OF BOOL	Virtual PROFINET bit module To receive an array with 8 elements of data type BOOL
16IX	ARRAY [0..15] OF BOOL	Virtual PROFINET bit module To receive an array with 16 elements of data type BOOL
32IX	ARRAY [0..31] OF BOOL	Virtual PROFINET bit module To receive an array with 32 elements of data type BOOL
IB	BYTE	Virtual PROFINET byte module To receive data of data type BYTE
IW	WORD	Virtual PROFINET byte module To receive data of data type WORD
ID	DWORD	Virtual PROFINET byte module To receive data of data type DWORD
IBAn	ARRAY [0..(n-1)] OF BYTE	Virtual PROFINET byte module To receive an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
IWAn	ARRAY [0..(n-1)] OF WORD	Virtual PROFINET byte module To receive an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64
IDAn	ARRAY [0..(n-1)] OF DWORD	Virtual PROFINET byte module To receive an array with n elements of data type DWORD n = 2, 4, 8, 16, 32 or 64



INFORMATION

Note that only the virtual modules documented here with the specified data types are supported. The LWORD data type is not supported.

During data exchange, the PROFINET IO controller transmits its send data to the I-modules in the automation system PSS 4000 via O-modules. Consequently, the O-modules on the PROFINET IO controller must have corresponding I-modules in the automation system PSS 4000. Also, the data type of an I-module must correspond to the data type of the O-module in the PROFINET IO controller.

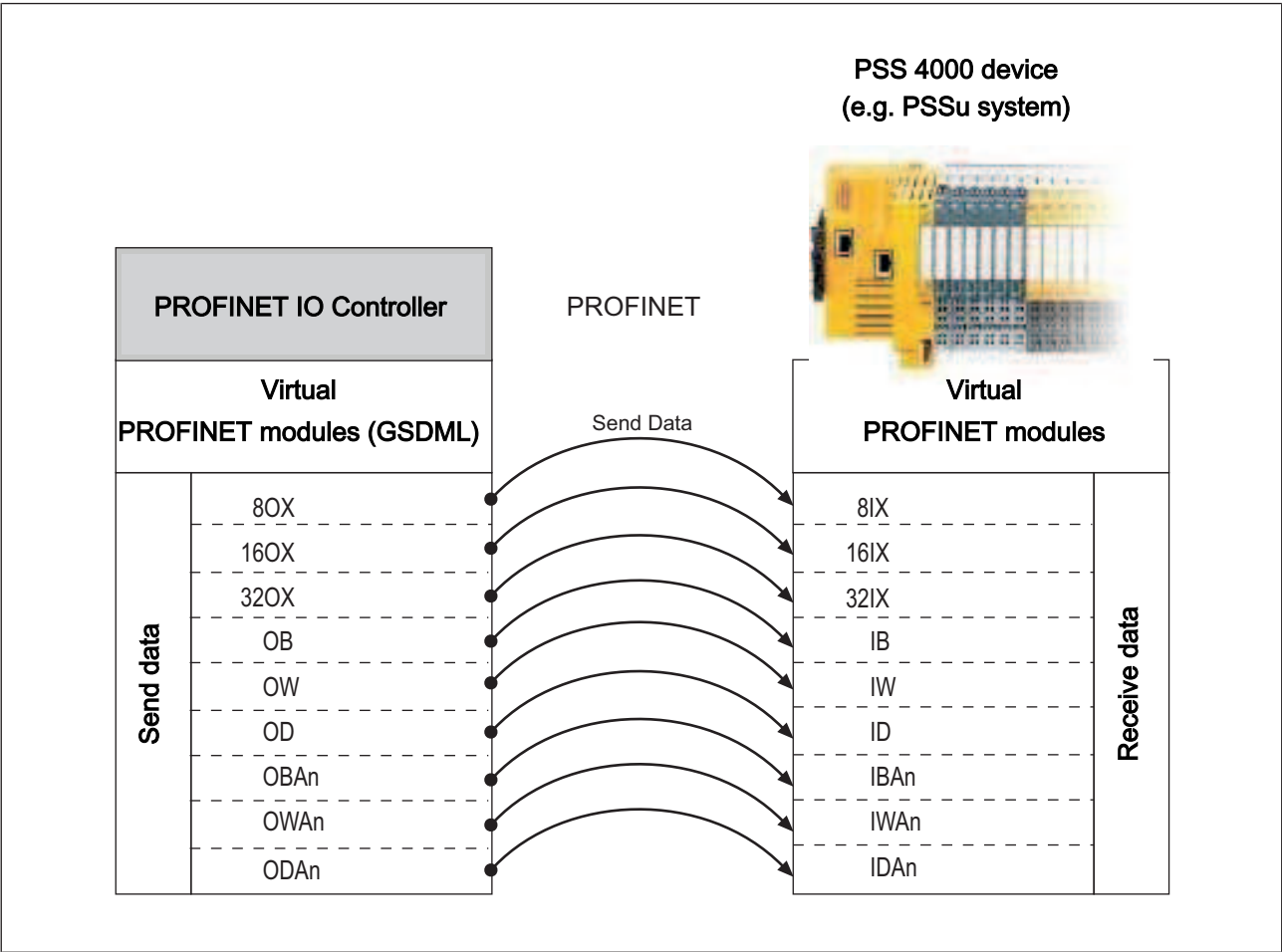


Fig.: Virtual PROFINET modules and their corresponding data types (receive data of a PSS 4000 device)

10.9.5 GSDML file

All the relevant data for a PROFINET device is contained in the corresponding GSD file (General Station Description). This file is XML-based and is also known as the GSDML file. It describes the properties and functions of a PROFINET device, which are significant for engineering as well as for data exchange.

The GSDML file of a PSS 4000 device with PROFINET IO DEVICE system section contains the following data, for example:

- ▶ All available virtual PROFINET modules
- ▶ Diagnostic messages with a remedy for various device-based PROFINET alarms

The GSDML file is supplied with PAS4000 and is available in the download area of the Pilz homepage (www.pilz.de -> Downloads).

10.9.6 Data transfer limits

The following limits apply for data transfer via virtual PROFINET modules:

- ▶ A maximum of 64 virtual modules can be configured for the PROFINET IO DEVICE system section of a PSS 4000 device.
- ▶ A maximum of 1440 Bytes of receive data (= external I-data) can be configured for the PROFINET IO DEVICE system section of a PSS 4000 device.
- ▶ A maximum of 1440 Bytes of send data (= external O-data) can be configured for the PROFINET IO DEVICE system section of a PSS 4000 device.
- ▶ Each configured virtual PROFINET module for external O-data requires an additional byte per data flow direction (send, receive) besides the pure payload.
- ▶ Each configured virtual PROFINET module for external I-data requires an additional byte per data flow direction (send, receive) besides the pure payload.
- ▶ The PROFINET IO DEVICE system section of a PSS 4000 device always requires 6 Bytes for each data flow direction (send, receive).

The byte requirement is calculated and displayed during the configuration in PAS4000.

Example

A virtual 32IX module has been configured for the PROFINET IO DEVICE system section of a PSS 4000 device.

	Data flow direction "Re- ceive" (external I-data)	Data flow direction "Send" (external O-data)
Virtual 32IX module		
Byte requirement for pure payload	4 Bytes	- - -
Byte requirement per virtual module	1 Byte	1 Byte
PROFINET IO Device system section		
Byte requirement per system section	6 Bytes	6 Bytes
Overall byte requirement for all configured modules	11 Bytes	7 Bytes

Due to the calculated byte requirement, the following number of bytes remain for the configuration of further virtual I/O modules:

- ▶ Virtual I-modules (external I-data, data flow direction "Receive"):
1440 Bytes - 11 Bytes = 1429 Bytes
- ▶ Virtual O-modules (external O-data, data flow direction "Send"):
1440 Bytes - 7 Bytes = 1433 Bytes

10.9.7 I/O data of PROFINET IO DEVICE

The PROFINET IO DEVICE system section enables a PSS 4000 device to exchange I/O data with the PROFINET IO controller via PROFINET. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding virtual PROFINET modules.

I/O-mapping of virtual PROFINET modules

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O-mapping for virtual PROFINET modules:

from/to		Virtual PROFINET modules	
		Receive data (= external I-data of I-modules such as 8IX, 16IX, 32IX, IB, IW, ID, IBAn, IWAn, IDAn)	Send data (= external O-data of O-modules such as 8OX, 16OX, 32OX, OB, OW, OD, OBAn, OWAn, ODAn)
Module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
PI variables	I-PI variable	◆	---
	O-PI variable	---	◆

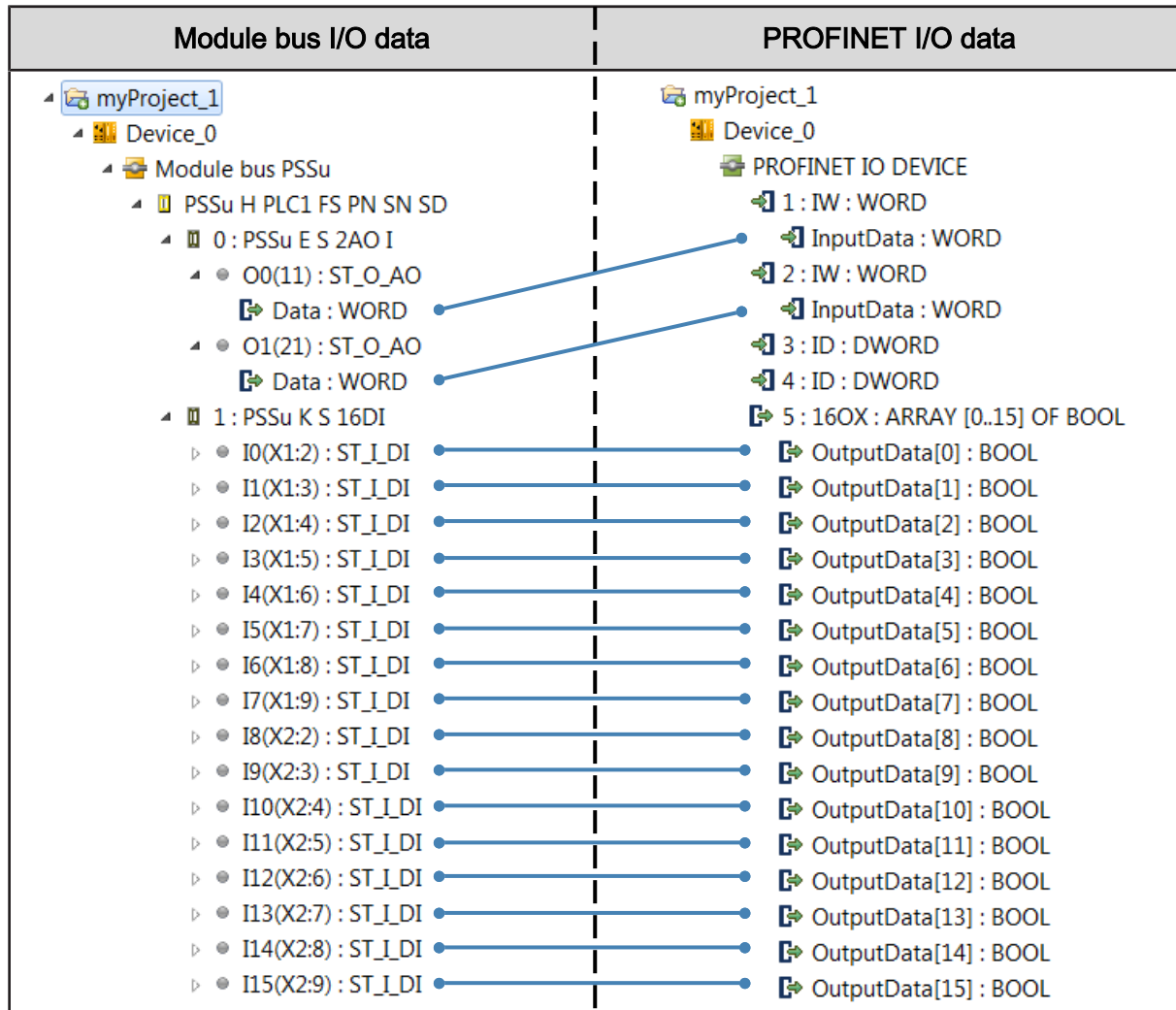
- ◆ I/O mapping possible
 --- I/O mapping impossible

Examples

► I/O-mapping of I/O data

The input signals from hardware inputs on a PSSu K S 16DI are to be transmitted to the PROFINET IO controller and the PROFINET IO controller is to control the hardware outputs on the PSSu E S 2 AO I.

Possible I/O mapping (example):

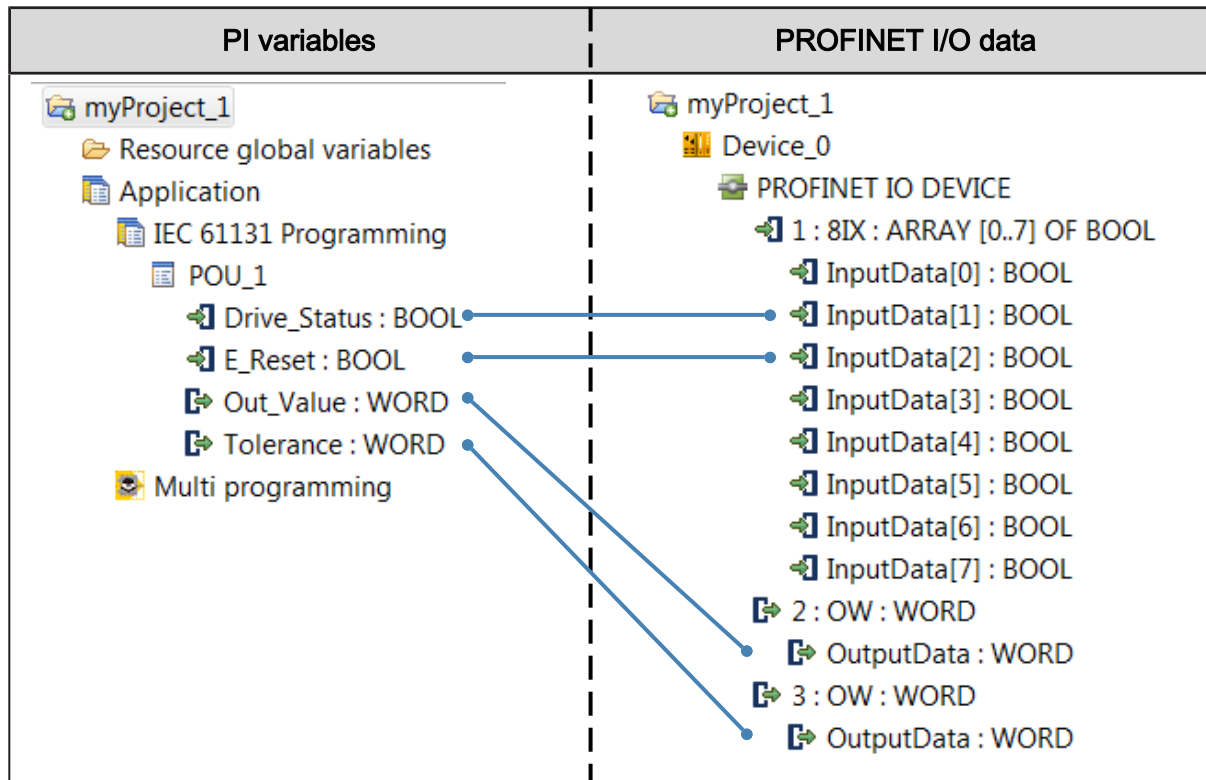


— I/O mapping

► I/O mapping of PI variables

The contents of a block's O-PI variables are to be transmitted to the PROFINET IO controller and I-PI variables are to be assigned to the receive data.

Possible I/O mapping (example):



—●— I/O mapping

10.9.8 Address format of PROFINET IO addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

10.9.8.1 Address format of send data

The addresses of PROFINET send data (O-data of virtual PROFINET modules) are structured as follows:

<Device name>.ProfinetIODev.<Virtual slot>.OutputData[<Element number>]

Examples:

- ▶ Address format for an 8OX:

```
myDevice_1.ProfinetIODev.1.OutputData[0]
myDevice_1.ProfinetIODev.1.OutputData[1]
myDevice_1.ProfinetIODev.1.OutputData[2]
myDevice_1.ProfinetIODev.1.OutputData[3]
myDevice_1.ProfinetIODev.1.OutputData[4]
myDevice_1.ProfinetIODev.1.OutputData[5]
myDevice_1.ProfinetIODev.1.OutputData[6]
myDevice_1.ProfinetIODev.1.OutputData[7]
```

- ▶ Address format for a 16OX:

```
myDevice_1.ProfinetIODev.2.OutputData[0]
...
myDevice_1.ProfinetIODev.2.OutputData[15]
```

- ▶ Address format for a 32OX:

```
myDevice_1.ProfinetIODev.3.OutputData[0]
...
myDevice_1.ProfinetIODev.3.OutputData[31]
```

- ▶ Address format for an OB:

```
myDevice_1.ProfinetIODev.4.OutputData
```

- ▶ Address format for an OBA4:

```
myDevice_1.ProfinetIODev.5.OutputData[0]
...
myDevice_1.ProfinetIODev.5.OutputData[3]
```

- ▶ Address format for an OW:

```
myDevice_1.ProfinetIODev.6.OutputData
```

- ▶ Address format for an OWA2:

```
myDevice_1.ProfinetIODev.7.OutputData[0]
myDevice_1.ProfinetIODev.7.OutputData[1]
```

- ▶ Address format for an OD:

```
myDevice_1.ProfinetIODev.8.OutputData
```

- ▶ Address format for an ODA4:

```
myDevice_1.ProfinetIODev.9.InputData[0]
...
myDevice_1.ProfinetIODev.9.InputData[3]
```

10.9.8.2 Address format of receive data

The addresses of PROFINET receive data (I-data of virtual PROFINET modules) are structured as follows:

<Device name>.ProfinetIODev.<Virtual slot>.InputData[<Element number>]

Examples:

► Address format for an 8IX:

```
myDevice_1.ProfinetIODev.10.InputData[0]
myDevice_1.ProfinetIODev.10.InputData[1]
myDevice_1.ProfinetIODev.10.InputData[2]
myDevice_1.ProfinetIODev.10.InputData[3]
myDevice_1.ProfinetIODev.10.InputData[4]
myDevice_1.ProfinetIODev.10.InputData[5]
myDevice_1.ProfinetIODev.10.InputData[6]
myDevice_1.ProfinetIODev.10.InputData[7]
```

► Address format for a 16IX:

```
myDevice_1.ProfinetIODev.11.InputData[0]
...
myDevice_1.ProfinetIODev.11.InputData[15]
```

► Address format for a 32IX:

```
myDevice_1.ProfinetIODev.12.InputData[0]
...
myDevice_1.ProfinetIODev.12.InputData[31]
```

► Address format for an IB:

```
myDevice_1.ProfinetIODev.13.InputData
```

► Address format for an IBA2:

```
myDevice_1.ProfinetIODev.14.InputData[0]
myDevice_1.ProfinetIODev.14.InputData[1]
```

► Address format for an IW:

```
myDevice_1.ProfinetIODev.15.InputData
```

► Address format for an IWA8:

```
myDevice_1.ProfinetIODev.16.InputData[0]
...
myDevice_1.ProfinetIODev.16.InputData[7]
```

► Address format for an ID:

```
myDevice_1.ProfinetIODev.17.InputData
```

► Address format for an IDA4:

```
myDevice_1.ProfinetIODev.18.InputData[0]
...
myDevice_1.ProfinetIODev.18.InputData[3]
```

10.9.9 PROFINET IO diagnostics

PROFINET IO diagnostics can be performed via the PSS 4000 system diagnostics (see [Diagnostic basics](#) [722]).

However, separate diagnostic mechanisms are defined by PROFINET. On PROFINET, IO devices can signal alarms and status messages to the IO controller.

PSS 4000 devices with PROFINET IO Device system section support device-based diagnostics in the form of PROFINET alarms. A diagnostic message is assigned to each PROFINET alarm type that can be signalled via a device. The diagnostic messages are stored in the device-specific GSDML file and include remedy information. Each diagnostic message also corresponds to a light signal from a status LED on the device.

With an appropriate PROFINET diagnostic tool, the corresponding diagnostic message and remedy information can be displayed for each PROFINET alarm.

If device-based diagnostics are activated during configuration in PAS4000 (PROFINET IO DEVICE Editor), as a PROFINET IO device a device can signal the following device states via PROFINET alarms:

Alarm type (activated in the PROFINET IO DEVICE Editor)	Message text (GSDML file)	Show LED (device)	
Error message for the device	A message of "Error" severity is present for at least one system section (see diagnostic list)	DIAG	Lights up red or flashes red
Error on module bus	Operating state "Safe state of all FS outputs on the device" or at least one module cannot be accessed (e.g. a module has been removed during operation, set/actual hardware registry does not match)	MBUS	Lights up red or flashes red
Error on FS SafetyNET p	Operating state "FS SafetyNET p in STOP state with error: Major FS error	FS SNp	Lights up red or flashes red
Error on ST SafetyNET p	Operating state "ST SafetyNET p in STOP state with error: Major FS+ST error	ST SNp	Lights up red or flashes red
Warning message for the device Note: If one of the error messages is present simultaneously for the device, the warning message will not be signalled.	A message of at least "Warning" severity is present for the device (see diagnostic list).	DIAG	LED orange
Forcing and online changes on the FS resource	On the FS resource, forcing is active and/or there is at least one online change active.	FS FORCE	Lights up yellow
Forcing and online changes on the ST resource	On the ST resource, forcing is active and/or there is at least one online change active.	ST FORCE	Lights up yellow

10.9.10 Data consistency with PROFINET IO

The PROFINET IO controller is responsible for the data consistency of the receive data sent by the PROFINET IO controller. From the PSS 4000's perspective, no statement can be made regarding the data consistency of the PROFINET IO controller.

Data consistency of the receive data

From the perspective of the PSS 4000, the receive data of a virtual PROFINET module is external I-data.

As soon as external I-data is available in a virtual PROFINET module, it is treated as consistent data (see [Data consistency](#)  377]. The external I-data of a virtual PROFINET module forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a PROFINET IO connection" (see [Consistency ranges](#)  378].

Data consistency of the send data

From the perspective of the PSS 4000, the send data of a virtual PROFINET module is external O-data.

The external O-data of a virtual PROFINET module is consistent if it comes from the same consistency range on a data source (see [Data consistency](#)  377]).

PSS 4000 always treats the external O data of a virtual PROFINET module as one unit. It is the task of the PROFINET IO controller to guarantee the necessary data consistency.

11 Project

11.1 Project structure

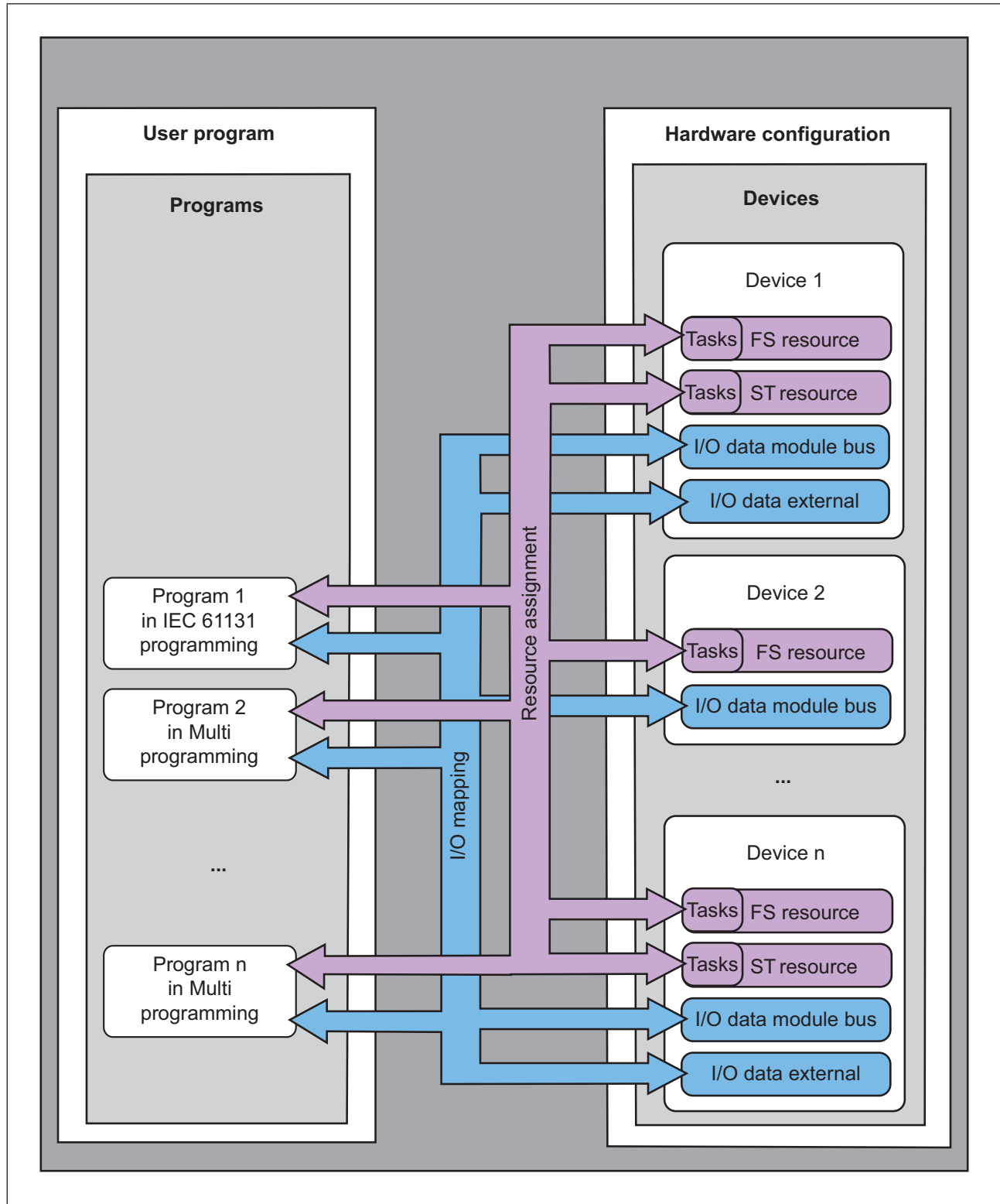


Fig.: Project structure

To operate a plant with the automation system PSS 4000, a project must be created in PAS4000.

A project consists of the hardware configuration and the user program.

The hardware configuration specifies which devices are used in the project and the settings used for the devices. The hardware registry is specified for each PSSu system or PSS u2 system, for example. The module bus I/O data, which is processed or provided by the device, is derived from the specification of the I/O modules. External communication can also be configured for devices with external communication (e.g. Ethernet/IP). This produces the external I/O data, which is processed or provided by the device.

The user program consists of one or more programs. Programs for control systems of performance class PSS 4000 PLC can be created in IEC 61131 programming and Multi programming. Programs for PSS 4000-exclusive devices of performance class PSS 4000 multi may only be created in Multi programming. Programs for PSS 4000-exclusive devices of performance class PSS 4000 I/O+ may only be created in Multi programming with limited function range.

► IEC 61131 programming

When programming in accordance with IEC 61131, three types of program organisation units (POUs) are available for structuring the user program:

- Programs (PRG)
- Function blocks (FB)
- Functions (FUN)

The program forms the higher structural level. Functions and function blocks can be called in a program.

Function blocks and functions perform specific individual tasks within the user program. The POUs can be formulated either in the IL (Instruction List) or STL (Structured Text) programming language.

► Multi programming

Multi programming is performed according to the component model in the graphics Multi Editor. Three types of blocks are available for structuring the user program:

- Multi program

A Multi program is used exclusively to structure the user program. Component blocks and basic blocks can be called up in a multi program.

- Component block (CB)

Component blocks are used to combine component blocks and/or basic blocks. Several separate sub-solutions for automation functions can be combined in this way.

Component blocks can be used to reproduce the plant and machine structure.

Component blocks are self-contained units, which can be easily reused.

- Basic block (BB)

Basic blocks are used to implement any complex automation solutions.



INFORMATION

The term "block" is used as a generic term for multi program, component block, basic block and program organisation units of the type program, function block and function.

The connection between the blocks and the I/O hardware data is made via the so-called PI variables. PI variables can also be used for data exchange between blocks.

If no resource global variables are used, the user program can be created independently of the used hardware. Information on which section of the user program is to be executed on which resource under which task does not need to be defined in the resource assignment until programming is finished. The tasks determine the run-time properties of sections of the user program, i.e. the priority with which they are executed.

Once programming is finished, the connection between the PI variables and the automation system I/O data must be established. This is done in the I/O mapping.

11.2 Validity of process data

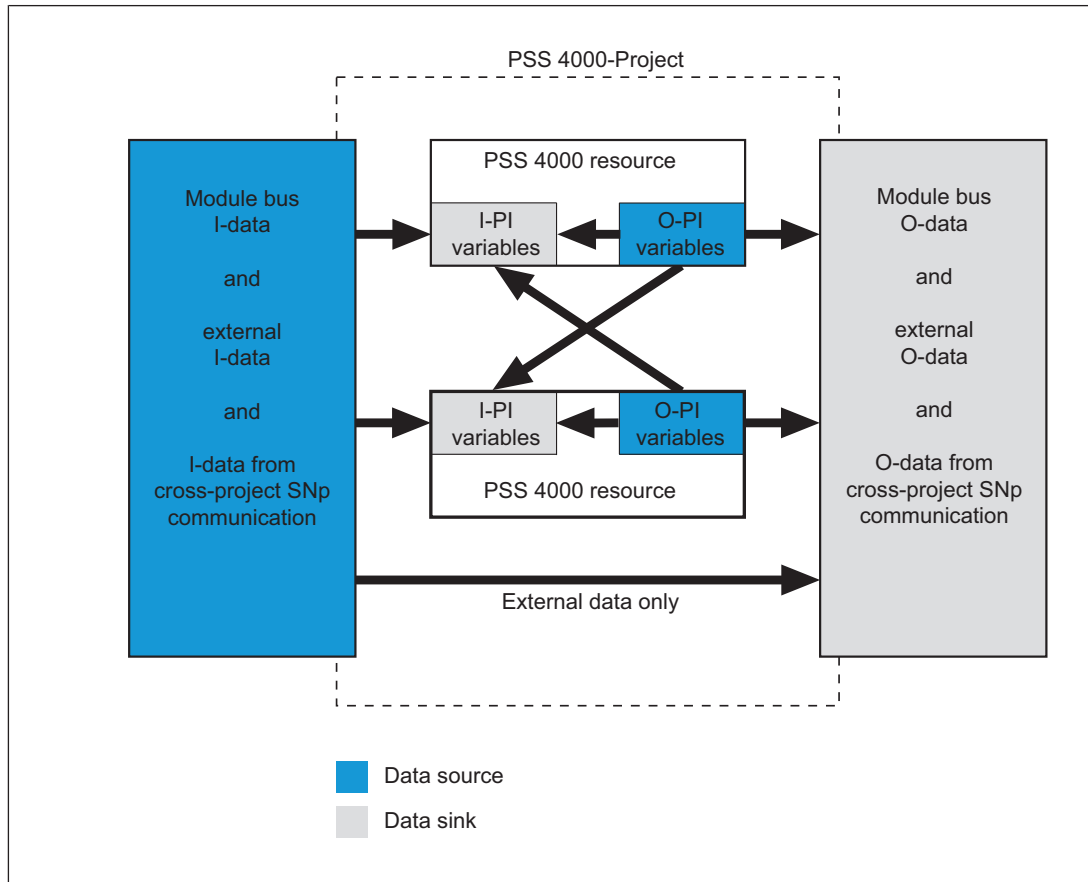


Fig.: Data flow within the project

The process data includes the I/O data and PI variables. Process data is exchanged between data sources and data sinks within a project.

Data sources are:

- ▶ **Module bus I-data**
Input data that comes from the FS/ST module bus system section, e.g. from modules of PSS 4000-exclusive devices
- ▶ **External I-data**
Input data that comes from the system sections for external communication, e.g. from EtherNet/IP
- ▶ **O-PI variables**
O-PI variables from PSS 4000 resources, e.g. from the FS resource or ST resource system sections; the resources may be on the same device or on different devices.
- ▶ **I-data from cross-project Snp communication**
Output data from other PSS 4000 projects, e.g. O-PI variables.

Data sinks are:

- ▶ **Module bus O-data**
Output data that is written to the FS/ST module bus system section, i.e. is downloaded to the modules of PSS 4000-exclusive devices

► External O-data

Output data that goes out externally via system sections for external communication, e.g. via EtherNet/IP.

► I-PI variables

I-PI variables from PSS 4000 resources, e.g. from the FS resource or ST resource system sections

► O-data from cross-project SNp communication

Input data from other PSS 4000 projects, e.g. I-PI variables.

The data flow is shown in the diagram using arrows. I/O mapping defines which data source supplies data to which data sink.

During the data flow, either the data source or the communication channel between the data source and data sink may malfunction, so that the data sink fails to receive a valid process value from the data source. To enable the data sink to establish whether the process value it has received is valid, almost all process values in the data sink have what's termed a "valid bit". Only the I-data for cross-project SNp communication has no valid bit.

The valid bit is FALSE if the process datum in the data source is marked invalid or if it's impossible to access the process datum in the data source.

If the valid bit is FALSE, the process value is invalid and may not be used. If this is the case, the data sink has a substitute value which is used instead of the process value.

Situations which set the valid bit to FALSE:

► On module bus I-data

- Error on an input (e.g. test pulse error)
- Error on an I/O module (e.g. module overheating)
- Error on the module bus
- Module bus starting

► On external I-data

- Communication is stopped
- Communication timeout

► On O-PI variables

- Task is in STOP state
- Resource is in STOP state

Note:

► Start-up behaviour

On start-up, situations may occur in which the user program in a device's resource is already being run, although data sources (e.g. the local module bus) have not yet supplied any valid data. In this case the user program will operate with the [Substitute values](#)  [362](#). As soon as the system section with the data source is up and running, the current data will be processed rather than the substitute values. The valid bits can be evaluated to ensure that this signal change does not provoke any unwanted reaction (see [Evaluation of the valid bits in the user program](#)  [363](#)).

A valid bit should be evaluated as a random sample from each system section from which data is processed in the user program (ST resource, FS resource, ST module bus, FS module bus, ST SafetyNET p, FS SafetyNET p, system section for external communica-

tion such as Ethernet/IP). For example, if an I-PI variable is assigned to an I-datum on the local FS module bus and the valid bit of that I-PI variable is TRUE, it's possible to conclude that the local FS module bus is running. At most, individual FS-I modules may be in an error state. If an I-PI variable is assigned to an I-datum on the ST module bus on another device and the valid bit of that I-PI variable is TRUE, it's possible to conclude that both the ST module bus on the other device and communication via ST SafetyNET p are running. At most, individual ST-I modules on the other device may be in an error state.

► Online action: "Forcing"

When forcing an I-PI variable for which a valid bit is declared, the valid bit is automatically forced with the value TRUE.

Exceptions:

- The user changes the force value for the valid bit to FALSE.
- The user deactivates forcing for the valid bit.
- The I-PI variable belongs to a write and read-protected block or sealed block. In this case, it will not be possible to force the valid bit.

11.2.1 Substitute values

The substitute values for process data are fixed.

Substitute values can be:

► Zero (= safe state)

For the USTRING data type this corresponds to the empty string; for the UCHAR data type it corresponds to the character U+000000.

► Last valid value

Process datum	Substitute value
I-PI variable on an FS resource	Zero
Output of an FS bit module	Zero
O-datum of an FS byte module	Zero
I-PI variable of an ST resource	Zero
Output from an ST bit module	Zero
O-datum from an ST byte module	Zero
ST-O datum PROFIBUS-DP Slave	Zero
ST-O datum Modbus/TCP Server connection	Last valid value
ST-O datum Modbus/TCP Client connection	Zero
ST-O datum Raw TCP	Zero
ST-O datum Raw UDP	Zero
ST-O datum EtherNet/IP Adapter	Zero
ST-O datum PROFINET IO DEVICE	Zero
FS-O datum PROFIBUS-DP Slave with PROFI-safe	Zero

11.2.2 Evaluation of the valid bits in the user program

If a data source fails to supply a valid process datum, the valid bit in the data sink is set to FALSE and the data sink is assigned the substitute value. This process can be monitored by evaluating the valid bit.

The valid bit from an I-PI variable can be evaluated in the user program. It's not yet possible to monitor the valid bits from other data sinks.

If the I-PI variable is a multiple-element variable (e.g. a structure of the I/O data type ST_I_AI), the valid bit for the overall structure is identical to the valid bits of the structure's individual elements.

Evaluation of the valid bit in Multi programming

The PSS 4000 system block "VALID" is used to evaluate the valid bit.

Evaluation of the valid bit in IEC 61131 programming

The VALID extension must be declared for the I-PI variable in order to evaluate the valid bit.

11.3 Task basics

Tasks determine the blocks' run-time properties (Multi programming: Multi program, component block, basic block; IEC 61131 programming: program-type POU), i.e. the conditions, the frequency and the priority with which they are executed.

One or more blocks are assigned to a task.

A resource may contain a maximum of 9 tasks. The number of tasks depends on the device. Details can be found in the device's operating manual.

The system organises task execution. The tasks' properties influence when the tasks are executed and in which order.



INFORMATION

For many projects it's not necessary to configure task properties. The default configuration is sufficient (see [Default configuration of tasks](#) [ 369]). So you don't necessarily have to deal with the issue of "Tasks".

11.3.1 Task properties

Task properties are defined in the task configuration:

► Name

The name must comply with the Rules for identifiers and must be unique on a resource.

► Type

The following types are available: "Periodic task" and "Periodic task, exclusive". Both task types behave the same, but the type "Periodic task, exclusive" may only be assigned to just one block (potentially with sub-blocks).

The periodic task is executed once only within a definable task cycle time. The intervals between executions may vary.

If the execution is not completed within the task cycle time, the operating state "Resource in RUN condition with task error" will result.

► Priority

The priority can be defined for each task. The priority determines the importance of a task in comparison with other tasks on a resource. As such it determines the order in which tasks on a resource are executed. The priorities are as follows: high, medium and low.

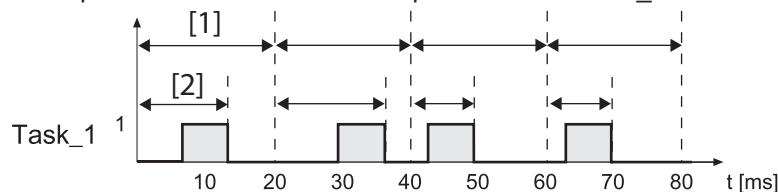
The number of times a priority may be assigned to a resource is specified. For example, a resource with 9 tasks may have 3 tasks with high priority, 3 with medium and 3 with low.

► Task cycle time (only on periodic tasks)

The task cycle time defines the period within which the task must be executed.

The task cycle time also serves as a subpriority. Where tasks have the same priority, the task with the shortest task cycle time is executed first.

Example: Possible execution of a periodic task "Task_1" with a task cycle time of 20 ms



[1] Task cycle time

[2] Execution time



INFORMATION

To increase the availability of a PSS 4000-exclusive device of performance class PSS 4000 PLC, PSS 4000 multi and PSS 4000 I/O+, we recommend that the value for the task cycle time is greater than or equal to the maximum execution time plus 20 %.

The maximum execution time should be determined over a longer period (e.g. more than 6 hours), during which the plant runs through every operating state.

If you do not follow this recommendation it will not affect the safe function of the device.

11.3.2 Task execution

A PSS 4000-exclusive device of performance class PSS 4000 PLC, PSS 4000 multi and PSS 4000 I/O+ is a multi-tasking system with pre-emptive scheduling. Up to 9 tasks may be available on one resource. The number of tasks depends on the device. Details can be found in the device's operating manual.

Execution of the tasks is organised by the system. The tasks' properties influence when the tasks are executed and in which order. The execution of lower priority tasks is interrupted if higher priority tasks are pending execution.

The following applies when executing tasks:

- ▶ Execution time

The execution time is the time from the moment the cycle starts to the moment the task execution is completed. If the task execution is interrupted by another task, the execution time is increased.

- ▶ Execution controlled per device

Each device within the project independently controls task execution. A device can have a maximum of one FS resource and one ST resource. A resource may contain a maximum of 9 tasks. The number of tasks depends on the device. Details can be found in the device's operating manual.

The number of times a priority may be assigned to a resource is specified. For example, a resource with 9 tasks may have 3 tasks with high priority, 3 with medium and 3 with low.

A resource may not have two tasks with the same priority and cycle time. Exception: A task with the task type "periodic task" and a task with the task type "periodic task, exclusive" may have the same priority and cycle time.

- ▶ Order by priority

The tasks are in accordance with their priority. High priority tasks are executed first, followed by medium and then low priority tasks. Where tasks have equal priority, the task cycle time decides. The task with the shortest task cycle time is executed first. If there are tasks with equal priority and task cycle time, the order is random.

- ▶ Interaction between FS and ST resource

If a device has an FS and an ST resource, both are executed by the same CPU. The following order applies:

1. FS tasks, high priority
2. ST tasks, high priority
3. FS tasks, medium priority
4. ST tasks, medium priority
5. FS tasks, low priority
6. ST tasks, low priority

Execution of the FS task with high priority always has precedence.

- ▶ Periodic check for pending tasks

With each system clock pulse the system checks whether any tasks are pending execution and redefines the order of execution. If a task is being executed, it will be interrupted if a higher priority task is pending.

- ▶ Execution order

If several programs have to be run within a task, the programs are executed in alphabetical order, based on the program names.

Multi programming: The execution order for the basic blocks within a Multi program is determined by PAS4000.

► Reading and outputting the process image and resource global variables

When task execution begins, a copy is created of the section of the process image (I-PI and O-PI) needed by the blocks in the task. The necessary resource global variables are also copied.

When the task is executed, only the values from the copies are read and written.

Not until the task is complete, in other words, when all the task's blocks have been executed, is the data from the copies transferred to the resource global variables and the process image of outputs (O-PI).

By using this procedure, data cannot be modified externally while the task is being executed. Even if a task is interrupted during execution, it will subsequently continue working with the data in its copies.

► Task interruption

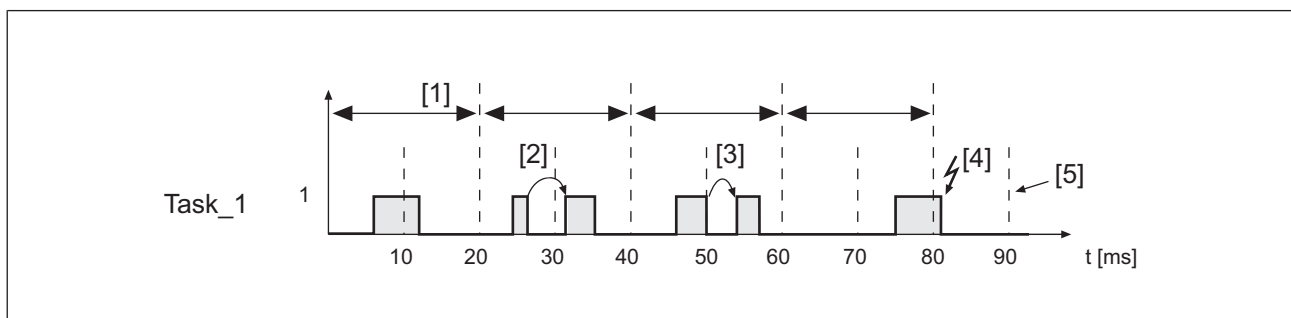
A higher priority task is not the only event that can interrupt a task. A task will also be interrupted, for example, if a block is waiting for a system function to be executed. As a result, when a block is executed, it is not certain whether a block has been executed previously in a higher priority task and whether its current data is available in the PI. If one task is to process data from another task, the user must ensure that the necessary data from the other task has already been processed, by evaluating a counter for example.

► Cycle time exceeded due to task configuration

To avoid the cycle time being exceeded unnecessarily on low priority tasks, the following rules should be observed when configuring multiple tasks on a resource: the task cycle time should become shorter with rising priority. See example 5.

Example 1

Execution of periodic task "Task_1", with a task cycle time of 20 ms and a system clock pulse of 10 ms



[1] Task cycle time

[2], [3] Interruption

[4] Error

[5] System clock pulse

With each system clock pulse the system checks whether any tasks are pending execution. The task "Task_1" is available for execution every 20 ms. However, it is not always executed immediately because higher priority tasks are executed first. The task may also be

interrupted because the program is waiting for a system function to be executed for example [2], or because a higher priority task is present on the system clock pulse [3]. If the task execution is not completed within the task cycle time, an error is triggered [4].

Example 2

A medium priority task is being executed. A high priority task is present on the system clock pulse. The first task is interrupted and the high priority task is executed. Execution of the first task is then continued.

Example 3

A high priority task with a task cycle time of 20 ms is being executed. A high priority task with a task cycle time of 10 ms is present on the system clock pulse. The first task is interrupted and the task with the shorter task cycle time is executed.

Example 4

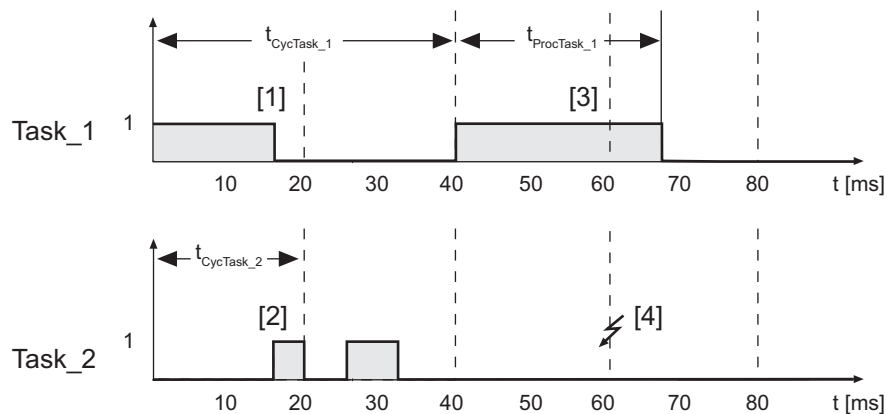
A medium priority ST task is being executed. A medium priority FS task is present on the system clock pulse. The ST task is interrupted and the FS task is executed first.

Example 5

Two tasks are executed on a resource: "Task_1" and "Task_2".

"Task_1" is configured with a task cycle time of 40 ms and "high" priority.

"Task_2" is configured with a task cycle time of 20 ms and "low" priority.



Due to its higher priority, "Task_1" is executed completely [1] before starting execution of "Task_2" [2]. Only when the execution time $t_{ProcTask_1}$ of "Task_1" is shorter than the cycle time $t_{CycTask_2}$ of "Task_2" is it possible for "Task_2" to be executed within its cycle time. If the execution time $t_{ProcTask_1}$ of "Task_1" is longer than the cycle time $t_{CycTask_2}$ of "Task_2" [3], "Task_2" cannot be executed in time and the cycle time will be exceeded as a result [4].

The following options are available to prevent the cycle time being exceeded:

- ▶ Configure a higher priority for "Task_2" than for "Task_1".
- ▶ Configure a cycle time for "Task_2" that is greater than or equal to that of "Task_1".
- ▶ Optimise the blocks that are executed in "Task_1" so that the execution time of "Task_1" is shorter than the cycle time of "Task_2".

11.3.3 Default configuration of tasks

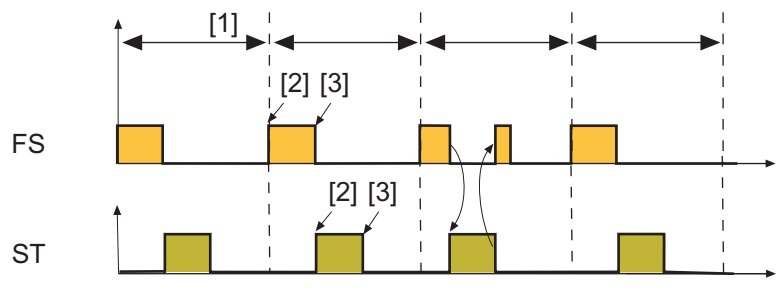
For many projects it's not necessary to configure task properties. The default configuration is sufficient.

The default configuration is designed so that the FS task is executed first on a device, followed by the ST task. The circumstance may arise where an FS block, for example, is waiting for a system function to be executed. In this case, execution of the FS task is interrupted and the ST task is executed.

If several programs have to be run within a task, the programs are executed in alphabetical order, based on the program names.

Multi programming: The execution order for the basic blocks within a Multi program is defined by PAS4000.

The FS task and ST task are executed once within the task cycle time of 100 ms.



[1] Task cycle time

[2] Resource global variables and process image of inputs (I-PI) and outputs (O-PI) are read in

[3] Output of resource global variables and process image of outputs (O-PI)

11.4 I/O mapping rules within a PSS 4000 project

When I/O mapping you must consider which data sources may be mapped to which data sinks. You must also bear in mind the data types of the data source and data sink.



INFORMATION

This is where the I/O mappings are described for the data flow within a project. These I/O mappings are made in PAS4000. I/O mappings for the data flow between projects (cross-project SNp communication) are made in PASconnect. Special rules apply to them, see [I/O mapping rules between PSS 4000 projects](#) [373].

Data sources and sinks

I/O mapping defines which data source supplies data to which data sink. The graphic uses arrows to show the data flow that's theoretically possible within the project.

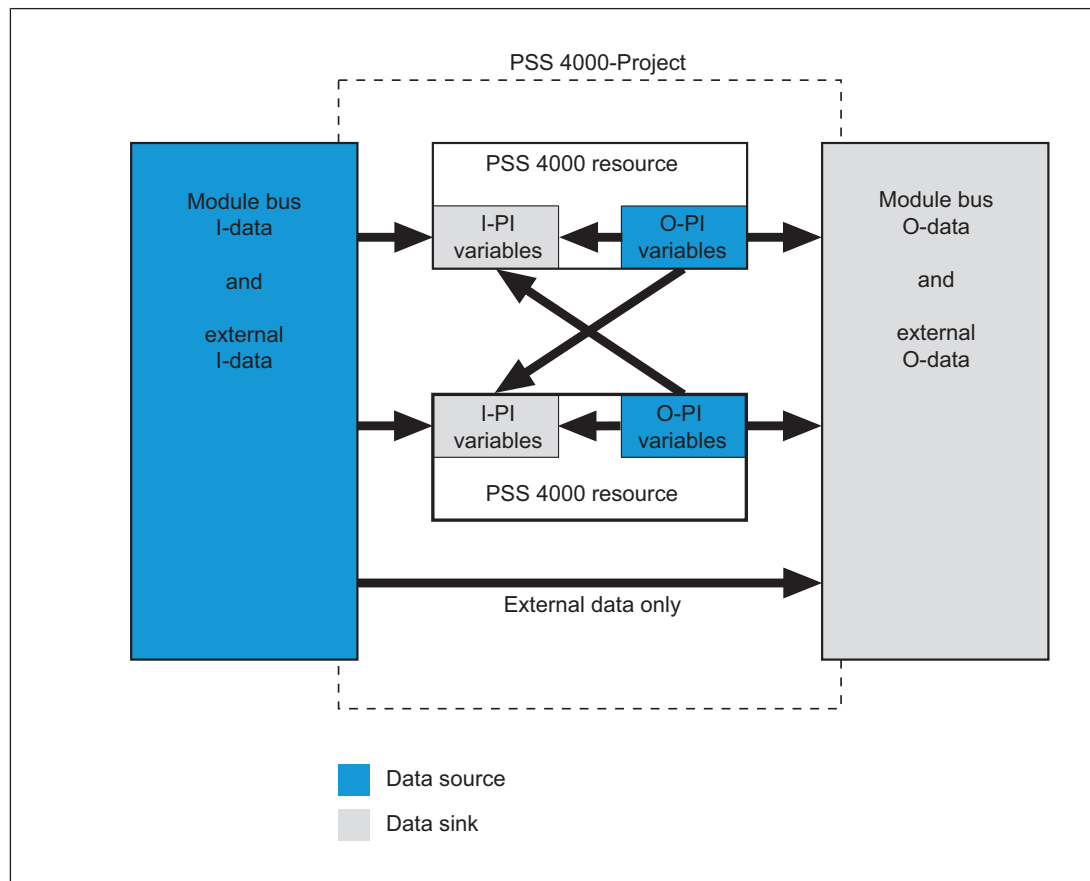


Fig.: Data flow in the project without cross-project SNp communication

The process data includes the I/O data and PI variables. Process data is exchanged between data sources and data sinks within a project.

Data sources are:

- Module bus I-data

Input data that comes from the FS/ST module bus system section, e.g. from modules of PSS 4000-exclusive devices

► External I-data

Input data that comes from the system sections for external communication, e.g. from EtherNet/IP

► O-PI variables

O-PI variables from PSS 4000 resources, e.g. from the FS resource or ST resource system sections; the resources may be on the same device or on different devices.

Data sinks are:

► Module bus O-data

Output data that is written to the FS/ST module bus system section, i.e. is downloaded to the modules of PSS 4000-exclusive devices

► External O-data

Output data that goes out externally via system sections for external communication, e.g. via EtherNet/IP.

► I-PI variables

I-PI variables from PSS 4000 resources, e.g. from the FS resource or ST resource system sections

The overview shows which I/O mappings are permitted.

I/O mapping		To (data sink)					
		Module bus ST-O datum	Module bus FS-O datum	ST-I-PI variable	FS-I-PI variable	External ST-O datum	External FS-O datum
From (data source)	Module bus ST-I datum	---	---	◆	---	◆	---
	Module bus FS-I datum	---	---	+	◆	+	◆
	ST-O-PI variable	◆	++	◆	---	◆	---
	FS-O-PI variable	+	◆	+	◆	+	◆
	External ST-I datum	◆	++	◆	---	---	---
	External FS-I datum	+	◆	+	◆	---	---

Legend:

- Not permitted (prevented by PAS4000)
- ◆ Permitted
- ++ Permitted, but the FS datum may exclusively be used for non-safety-related tasks.
- +
- Permitted, but the ST datum may still not be used for safety-related tasks.

Please note the following when I/O mapping

► I/O mappings between FS data source and ST data sink

If data from an FS data source is mapped to the data from an ST data sink, the data from the ST data sink may still not be used for safety-related tasks.

In other words, if ST output data (e.g. ST hardware outputs, ST-O data for external com-

munication) is controlled via the FS resource, this output data may only be used for **non-safety-related** tasks.

PAS4000 will warn you of such I/O mappings.

Should the application require the ST data sink to be used for safety-related tasks, then appropriate methods must be used to verify safety, for example via

- A separate risk analysis and
- Appropriate commissioning tests, plausibility tests

The user is responsible for the correct implementation.

► **I/O mappings between ST data source and FS data sink**

If data from an ST data source is mapped to the data from an FS data sink, the data from the FS data sink may not be used for safety-related tasks. PAS4000 will warn you of such I/O mappings. This will include, for example, the I/O mapping of:

- ST-O-PI variable to module bus FS-O datum
- ST-I datum externally to module bus FS-O datum

Should the application require the FS data sink to be used for safety-related tasks, then appropriate methods must be used to verify safety, for example via

- A separate risk analysis and
- Appropriate commissioning tests, plausibility tests

The user is responsible for the correct implementation.

► **I/O mapping between PI variable and PI variable in the same task**

After writing the O-PI variables, the value will not be available to read in to the mapped I-PI variable until the start of the next task cycle.

Data types

When I/O mapping for the data types of the data source and data sink, the following applies generally:

- If the data types of the data source and data sink are anonymous data types, then they must be structurally equivalent.
- If the data types of the data source and data sink are named data types, then they must be identical.
- I/O mappings between FS and ST are permitted in accordance with the above rules, i.e. including I/O mapping between BOOL and SAFEBOOL, for example.
Exception: If a data type is available exclusively on the ST resource and not on the FS resource, then I/O mapping is not possible. This applies for REAL and LREAL, for example.

The following additional rules apply for I/O data or PI variables of the ARRAY (Array) or STRUCT (Structure) data type:

- I-PI variable as data sink
If the I-PI variable is an ARRAY (Array) or STRUCT (Structure) data type, I/O mapping to its individual elements is **not** permitted. However, the data source may be an element of an array or structure.
- O-datum as data sink
If the O-datum is an ARRAY (Array) or STRUCT (Structure) data type, I/O mapping is possible to the whole array/structure or to the individual elements of the array/structure.

11.5 I/O mapping rules between PSS 4000 projects

Data exchange between PSS 4000 projects is configured in the software tool PASconnect with the help of I/O mapping. When I/O mapping you must consider which data sources may be mapped to which data sinks. You must also bear in mind the data types of the data source and data sink.

Data sources and data sinks

I/O mapping defines which data source supplies data to which data sink. The graphic uses arrows to show the data flow that's theoretically possible between projects.

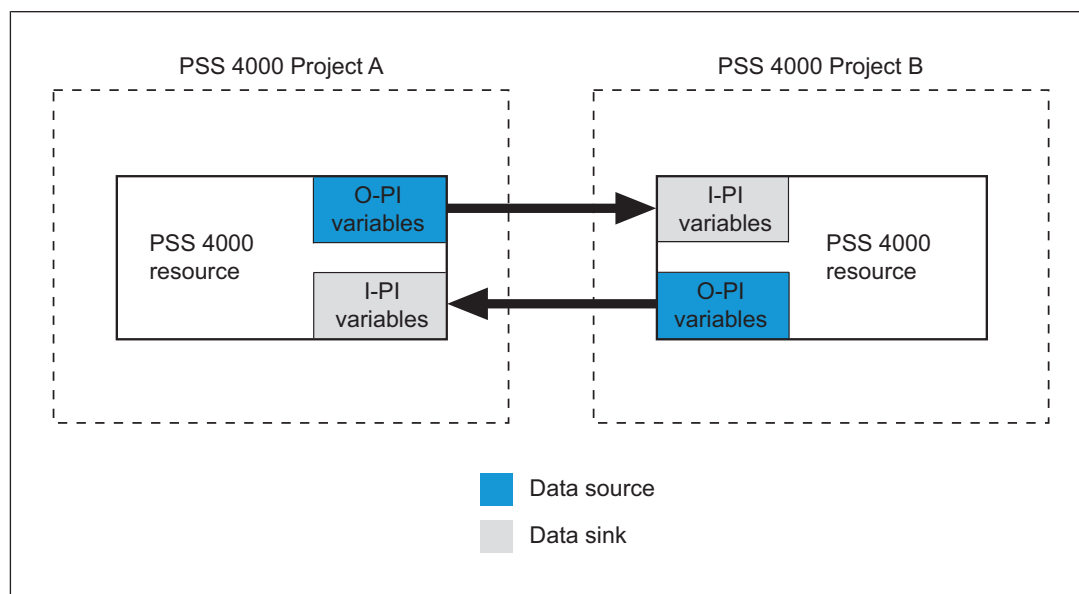


Fig.: Data flow between projects

Data sources are:

► O-PI variables

Any local O-PI variable from the user program on a project device's FS resource or ST resource. O-PI variables that are declared as resource global variables are not valid data sources for cross-project Snp communication.

Data sinks are:

► I-PI variables

Any local I-PI variable from the user program on a project device's FS resource or ST resource, when a VALID extension is not declared for the I-PI variable (valid bit) and I/O mapping has not yet been performed for the I-PI variable within the project. I-PI variables that are declared as resource global variables are not valid data sinks for cross-project Snp communication.

The overview shows which I/O mappings are permitted.

I/O mapping		To (data sink)	
		ST-I-PI variable	FS-I-PI variable
From (data source)	ST-O-PI variable	◆	---
	FS-O-PI variable	+	◆

Legend:

- Not permitted (prevented by PASconnect)
- ◆ Permitted
- + Permitted, but the ST datum may still not be used for safety-related tasks.

Please note the following when I/O mapping**► I/O mappings between FS data source and ST data sink**

If data from an FS data source is mapped to the data from an ST data sink, the data from the ST data sink may still not be used for safety-related tasks. PASconnect will warn you of such I/O mappings.

Should the application require the ST data sink to be used for safety-related tasks, then appropriate methods must be used to verify safety, for example via

- A separate risk analysis and
- Appropriate commissioning tests, plausibility tests

The user is responsible for the correct implementation.

► Data transfer limits

The Data transfer limits described in the System Description PSS 4000 for SafetyNET p apply for cross-project SNp communication between PSS 4000 projects.

In contrast to project-internal SNp communication, with cross-project communication between two devices there may be a maximum of one FS-Tx connection and one ST-Tx connection per task for each direction:

- 1 FS-Tx connection per task from device A to device B
- 1 FS-Tx connection per task from device B to device A
- 1 ST-Tx connection per task from device A to device B
- 1 ST-Tx connection per task from device B to device A

A maximum of 118 Bytes of FS process data (data with an FS data type) can be transferred with one FS-Tx connection.

A maximum of 251 Bytes of ST process data (data with an ST data type) can be transferred with one ST-Tx connection.

► Data types

With I/O mapping, the following applies for the data types of the data source and data sink:

- The data types of the data source and data sink must be elementary data types. I/O mapping between derived data types is not permitted, e.g. ARRAY data type (array) or STRUCT data type (structure).
- The name of the data type of the data source and data sink must match, but the SAFE prefix is not taken into account.
- I/O mappings between FS and ST are permitted in accordance with the above rules, i.e. I/O mapping also between BOOL and SAFEBOOL, for example.

► I/O mappings between PSS 4000 projects

If I/O mappings are to be performed between PSS 4000 projects, special care must be taken when mapping data sources to data sinks. In particular, to correctly identify the data sources and data sinks it is essential to note the project from which the data sources and data sinks are selected.

► **Change to I/O mappings between PSS 4000 projects**

If I/O mappings in the PASconnect project are changed, then the affected PSS 4000 projects must be rebuilt. The projects that are affected will depend on the PASconnect interface version (see below). If the projects are not rebuilt and downloaded, then the SafetyNET p configuration in the PSS 4000 projects will not be consistent. The data that is sent may not then correspond to the data anticipated by the receiver.

After the PSS 4000 projects have been built, commissioning (see PSS 4000 Safety Manual, section entitled "Comissioning and validity") or recommissioning (see PSS 4000 Safety Manual, section entitled "Changes" under "Change maintenance and decommissioning") must take place.

► **PASconnect interface version 1.2.x: PSS 4000 projects affected by changes to I/O mappings**

If I/O mappings in the PASconnect project are changed, then the PSS 4000 projects between which IO mappings have been changed must be rebuilt, as well as any projects that exchange data with these projects.

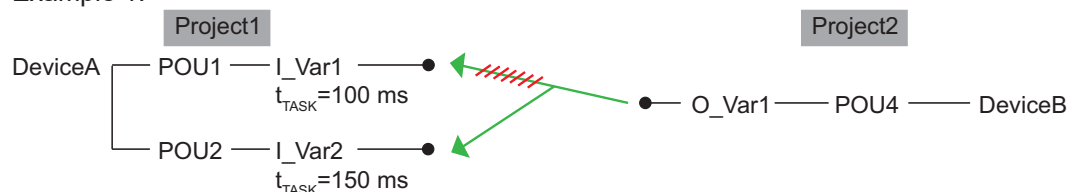
► **PASconnect interface version 1.3.x/1.4.x: PSS 4000 projects affected by changes to I/O mappings**

When you create a new I/O mapping or delete an existing I/O mapping, the PASconnect output data for the two PSS 4000 projects involved changes as a result and both projects must be rebuilt.

If an I/O mapping is changed, this does not necessarily mean that both PSS 4000 projects must be rebuilt. If the following requirements are met, one side of an I/O mapping can be changed without changing the PASconnect output data for the PSS 4000 project on the opposite side:

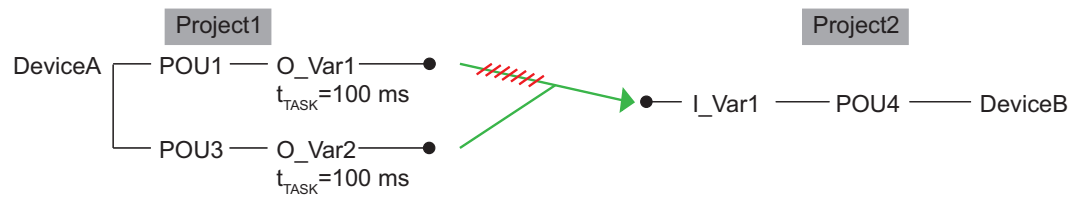
- The new variable must be on the same device as the old variable.
- In the case of O-PI variables, the task cycle time must not change.
- In the PASconnect project, the option **Include location information** must be deactivated.

Example 1:



After the change, *Project2* does not need to be rebuilt because the I-PI variables *I_Var1* and *I_Var2* from *Project1* are on the same device.

Example 2:



After the change, *Project2* does not need to be rebuilt because the O-PI variables *O_Var1* and *O_Var2* from *Project1* are on the same device and the task cycle times of the POUs are identical.

► **Exclusivity**

I-PI variables that are used for I/O mapping within a PSS 4000 project may no longer be used for cross-project Snp communication.

11.6 Data consistency

With PSS 4000, data consistency is generally understood as keeping together data that is recorded in the same task cycle and is considered current and related during a defined period.

Data consistency of a data source

In terms of data consistency, data sources (see [Validity of process data](#)  359) are divided into "consistency ranges". A data source's consistency range contains an amount of consistent data. This means that the process data on a data source is consistent if it comes from the same consistency range on the data source and is valid.

Data consistency with project-internal SafetyNET p communication

Process data in a consistency range that is transmitted within one PSS4000 project is consistent provided it is transmitted using a single SafetyNET p connection.

If the documented [SafetyNET p limits](#)  184 are exceeded during project-internal data transfer, the data from the consistency range is automatically split over several connections. In this case, data consistency is no longer guaranteed.

During the build process, PAS4000 issues a warning if the process data from a consistency range is split over several connections.

Data consistency with cross-project SafetyNET p communication

Process data in a consistency range that is transmitted across projects to other PSS 4000 projects is consistent provided it is transmitted using a single SafetyNET p connection (see "Data transfer limits" under [I/O mapping rules between PSS 4000 projects](#)  373).

During cross-project SNp communication, if process data cannot be transmitted using a single SNp connection, then PASconnect will prevent the SNp connection.

Data consistency of a data sink

A data sink's process data (see [Validity of process data](#)  359) is consistent if it comes from the same consistency range on a data source. No statement can be made regarding the relationship between a data sink's process data when it comes from various consistency ranges.

11.6.1 Consistency ranges

11.6.1.1 Consistency ranges on 1st generation devices

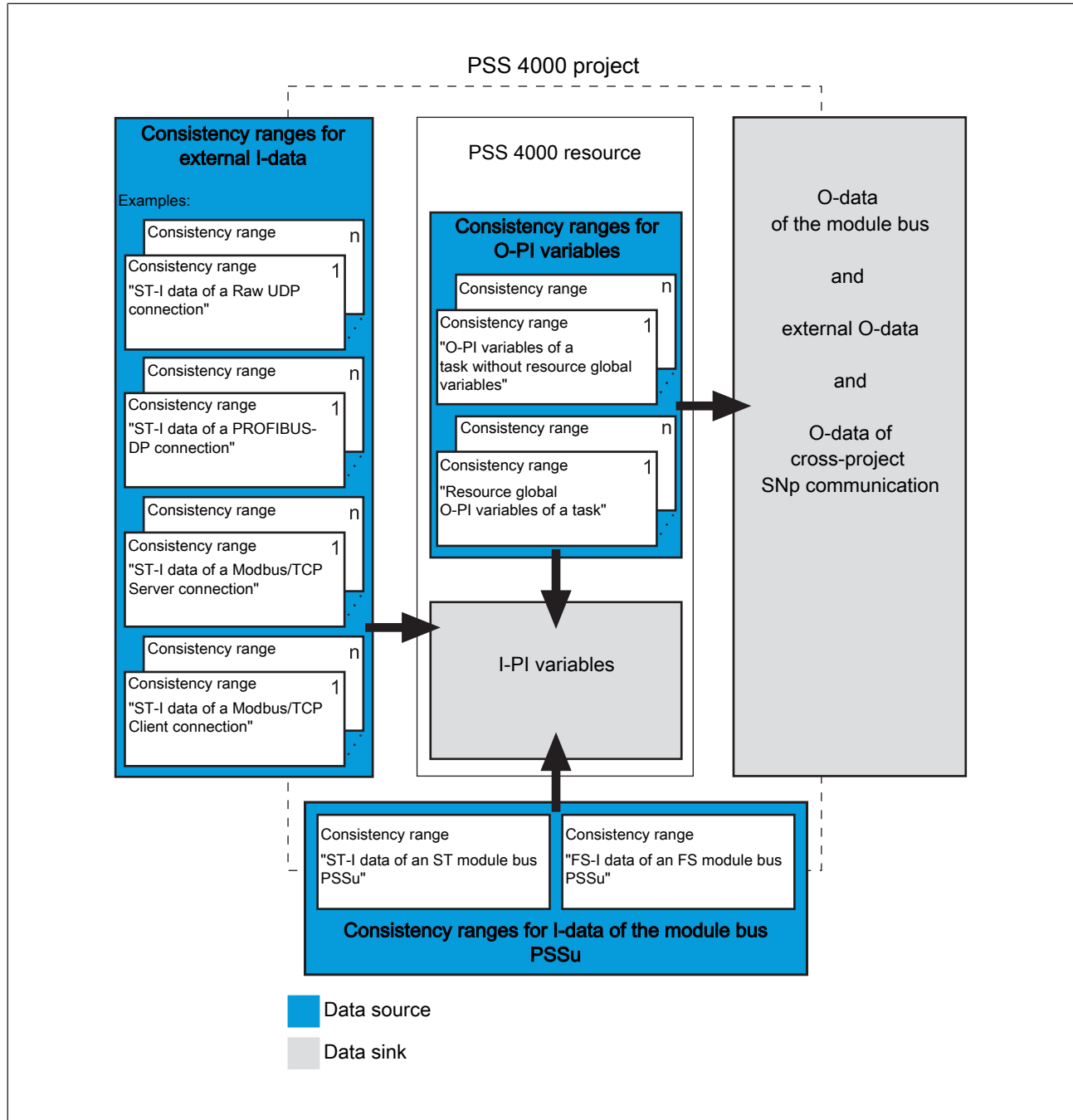


Fig.: Overview of a project's consistency ranges when using 1st generation devices

A project with 1st generation devices (e.g. PSSu systems) may have the following consistency ranges:

► **I-data of the module bus PSSu**

- Consistency range "FS-I data of an FS module bus"
- Consistency range "ST-I data of an ST module bus"

► External I-data

- Consistency range "ST-I data of an external communication connection"
- Consistency range "FS-I data of an external communication connection"

No statement can be made regarding the data consistency of data that is transmitted from external devices to a PSS 4000 device with corresponding system section via external communication. It depends on the communication mode (e.g. Modbus/TCP, PROFIBUS-DP, Raw UDP) and the options available with the communication mode. As soon as the data is available as external I-data in PSS 4000, it is treated as consistent data.

There are consistency ranges for the external I-data of any communication mode or external communication connection type. Each consistency range for external I-data behaves identically in terms of the data consistency of its data. In most cases, therefore, only the general consistency range for external I-data is stated (consistency range "ST-I data of an external communication connection" or consistency range "FS-I data of an external communication connection"). If necessary, however, the consistency range for each communication mode or each connection type can be stated separately.

Examples for the consistency range "ST-I data of an external communication connection":

- Consistency range "ST-I data of a Modbus/TCP Client connection"
- Consistency range "ST-I data of a Modbus/TCP Server connection"
- Consistency range "ST-I data of a PROFIBUS-DP connection"
- Consistency range "ST-I data of a Raw UDP connection"

Examples of the consistency range "FS-I data of an external communication connection":

- Consistency range "FS-I data of a virtual PROFIsafe F-module"

► O-PI variables

- Consistency range "O-PI variables of a task without resource global variables"

Such a consistency range includes all O-PI variables of blocks that are executed in the same task.

For this type of consistency range, no statement can be made regarding the relationship to resource global O-PI variables that are also executed in this task.

- Consistency range "Resource global O-PI variables of a task"

No statement can be made regarding the relationship of resource global variables to a consistency range "O-PI variables of a task without resource global variables". This is also the case if they are executed in the same task (see "Task 1" under [Example of consistency ranges of O-PI variables](#)  384).

The following applies for data consistency of resource global O-PI variables:

- Resource global O-PI variables that are accessed exclusively from blocks that are executed in the same task are consistent and form a consistency range (see "Task 1" and consistency ranges [1] and [2] under [Example of consistency ranges of O-PI variables](#)  384).
- The intersection of resource global O-PI variables accessed from blocks that are executed in different tasks is consistent and forms a consistency range (see "Task 2", "Task 3" and consistency range [3] under [Example of consistency ranges of O-PI variables](#)  384).

- No statement can be made regarding the data consistency of variables in a consistency range, which are mapped to an I-PI variable as well as a resource global I-PI variable, and are accessed from blocks that are executed in the same task.

11.6.1.2 Consistency ranges on 2nd generation devices

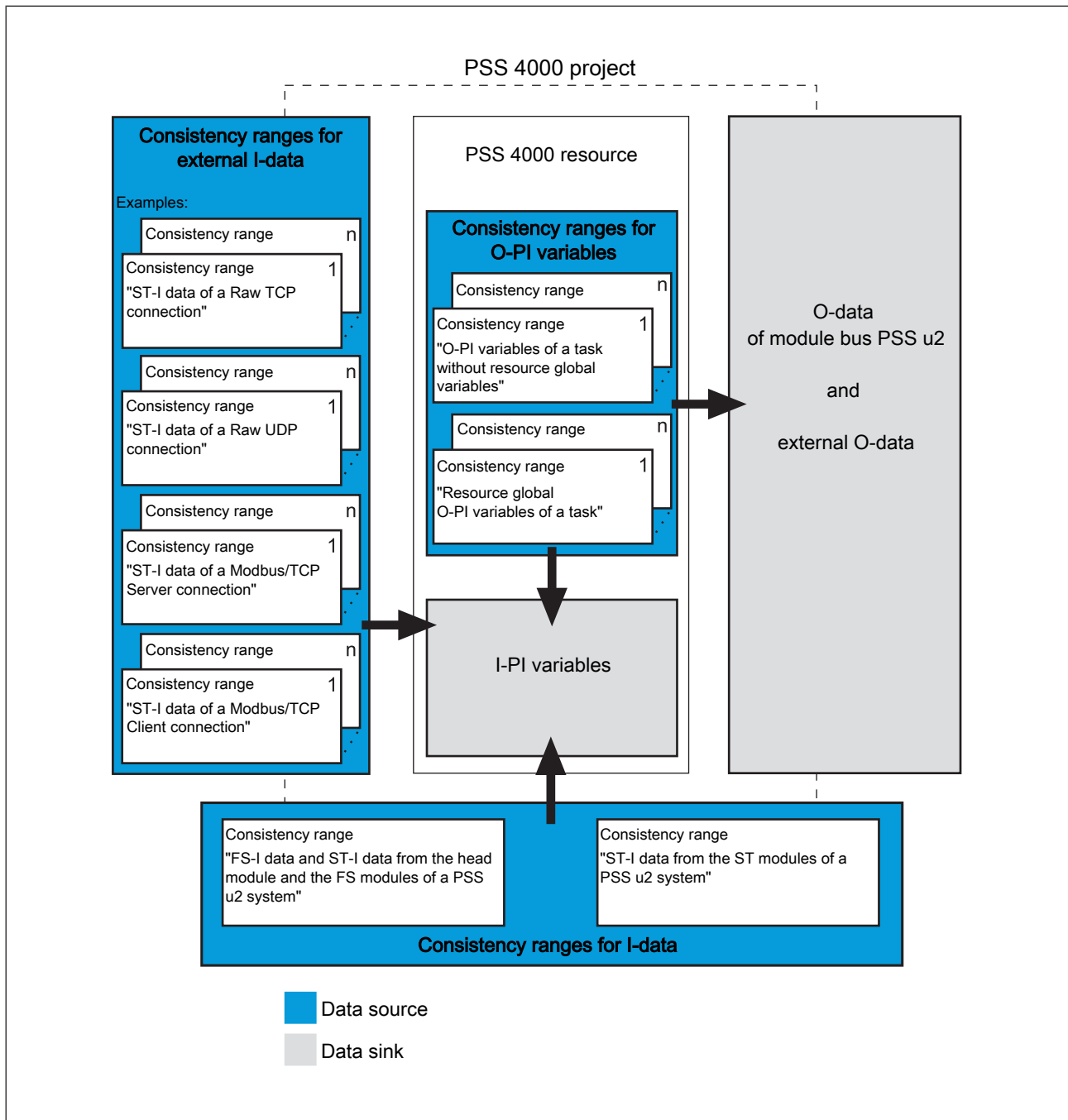


Fig.: Overview of a project's consistency ranges when using 2nd generation devices

A project with 2nd generation devices (PSS u2 systems) may have the following consistency ranges:

► **I-data**

- Consistency range "FS-I data and ST-I data from the head module and the FS modules of a PSS u2 system"

This I-data includes the I-data from the I/O data of the head module and the I-data that comes from FS modules and is transmitted via the FS section of the module bus PSS u2.

- Consistency range "ST-I data from the ST modules of a PSS u2 system"

This I-data includes the I-data that comes from ST modules and is transmitted via the ST section of the module bus PSS u2.

► **External I-data**

- Consistency range "ST-I data of an external communication connection"

No statement can be made regarding the data consistency of data that is transmitted from external devices to a 2nd generation device with corresponding system section via external communication. It depends on the communication mode (e.g. Modbus/TCP, Raw TCP, Raw UDP) and the options available with the communication mode.

As soon as the data is available as external I-data in PSS 4000, it is treated as consistent data.

There are consistency ranges for the external I-data of any communication mode or external communication connection type. Each consistency range for external I-data behaves identically in terms of the data consistency of its data. In most cases, therefore, only the general consistency range for external I-data is stated (consistency range "ST-I data of an external communication connection"). If necessary, however, the consistency range for each communication mode or each connection type can be stated separately.

Examples for the consistency range "ST-I data of an external communication connection":

- Consistency range "ST-I data of a Modbus/TCP Client connection"
- Consistency range "ST-I data of a Modbus/TCP Server connection"
- Consistency range "ST-I data of a Raw TCP connection"
- Consistency range "ST-I data of a Raw UDP connection"

► **O-PI variables**

- Consistency range "O-PI variables of a task without resource global variables"

Such a consistency range includes all O-PI variables of blocks that are executed in the same task.

For this type of consistency range, no statement can be made regarding the relationship to resource global O-PI variables that are also executed in this task.

- Consistency range "Resource global O-PI variables of a task"

No statement can be made regarding the relationship of resource global variables to a consistency range "O-PI variables of a task without resource global variables". This is also the case if they are executed in the same task (see "Task 1" under [Example of consistency ranges of O-PI variables](#)  384).

The following applies for data consistency of resource global O-PI variables:

- Resource global O-PI variables that are accessed exclusively from blocks that are executed in the same task are consistent and form a consistency range (see "Task 1" and consistency ranges [1] and [2] under [Example of consistency ranges of O-PI variables](#)  384).
- The intersection of resource global O-PI variables accessed from blocks that are executed in different tasks is consistent and forms a consistency range (see "Task 2", "Task 3" and consistency range [3] under [Example of consistency ranges of O-PI variables](#)  384).

- No statement can be made regarding the data consistency of variables in a consistency range, which are mapped to an I-PI variable as well as a resource global I-PI variable, and are accessed from blocks that are executed in the same task.

11.6.1.3 Example of consistency ranges of O-PI variables

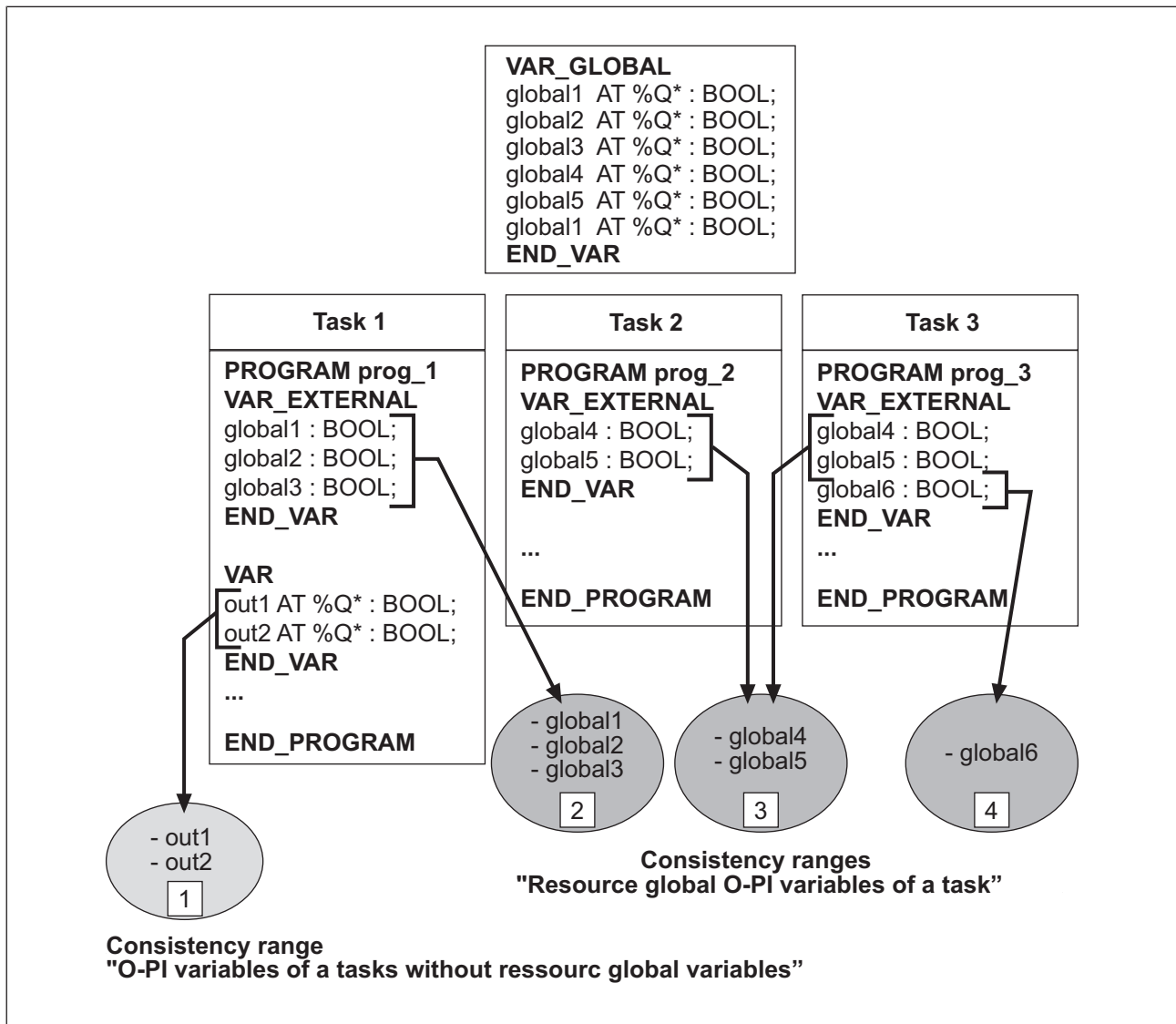


Fig.: Example of consistency ranges of O-PI variables

► Task 1

No statement can be made regarding the relationship of resource global variables to a consistency range "O-PI variables of a task without resource global variables". This also applies if they are executed in the same task.

► Task 1 and consistency ranges [1] and [2]

Resource global O-PI variables that are accessed exclusively from blocks that are executed in the same task are consistent and form a consistency range

► Task 2, task 3 and consistency range [3]

The intersection of resource global O-PI variables accessed from blocks that are executed in different tasks is consistent and forms a consistency range.

11.6.2 Influence of scheduling on data consistency

Scheduling regulates the order in which data in a project is processed and also ensures optimum data flow, taking into account the various cycle times, execution times and processing times (see [Reaction times](#)  570).

In PAS4000, if configurable cycle times (e.g. cycle time with external communication) are calculated automatically by the system, the system will determine the optimum conditions and reaction times for the project. Data consistency in the consistency ranges is guaranteed.

If configurable cycle times are configured by the user, the system will determine the optimum reaction times for the configured conditions. This may deteriorate in comparison with the reaction times that are determined automatically. Data consistency in the consistency ranges is guaranteed.

11.7 Safety status of blocks

In IEC 61131 programming it is possible to program function block-type and function-type POU's, which have a defined safety state. In Multi programming, blocks cannot be programmed with a safety state. However, in Multi programming the POU's programmed in IEC 61131 are used as basic blocks (IEC 61131 basic blocks). These basic blocks have a safety state.

Non-safety-related tasks may be programmed in ST blocks. ST blocks may be executed on either an ST or an FS resource. ST blocks are shown in green in PAS4000.

Safety-related tasks must be programmed in FS blocks. FS blocks may only be executed on an FS resource. FS blocks are shown in yellow in PAS4000.

In IEC 61131 programming, a POU becomes an FS-POU as soon as an FS data type variable is declared or an FS-POU is called/instantiated.

If it is possible to access non-safety-related data (ST data) within an FS block, the block becomes an "FS block with mixed input interface". They are given a special marking in PAS4000:

- ▶ Multi programming: half green and half yellow
- ▶ IEC 61131 programming: half white and half yellow

This special marking is intended to identify a block as potentially hazardous.

In IEC 61131 programming, an FS-POU will become an FS-POU with mixed input interface in the following circumstances:

- ▶ Declaration of an I-PI variable that is **not** an FS data type
- ▶ Declaration of a variable of type VAR_INPUT, VAR_IN_OUT or VAR_EXTERNAL, that is **not** an FS data type
- ▶ Call/instantiation of a POU in which an I-PI variable or a VAR_EXTERNAL type variable is declared, which is **not** an FS data type

When using FS blocks with mixed input interface, please note:

- ▶ Data on the ST resource is **not** safe. It may only be an **additional** criterion for a safety-related action (e.g. start/stop) and must **not** prevent safety functions. For example, if for whatever reason a stop signal has not been activated in the ST resource, or a start signal is continuously activated, this must not be able to compromise the safety of the machine or process.
- ▶ Appropriate procedures must be put in place to verify safety, e.g. by means of
 - A separate risk analysis and
 - Appropriate commissioning tests, plausibility tests
- ▶ We recommend that you test an FS-POU with mixed interface fully in the first instance, if necessary have it tested and certified by a notified body (e.g. BG, TÜV) and then use PAS4000 to seal it.
- ▶ The user is responsible for the correct implementation.



NOTICE

When sealed, FS-POUs with mixed input interface lose their label as potentially hazardous (half white and half yellow) and are shown in yellow. It is the responsibility of the manufacturer or notified body to ensure that access to non-safety-related data (ST data) within the block is not safety-critical.

11.8 Device naming via PAS4000

Device naming is performed by the user in PAS4000. The data required to address a device uniquely within the project is defined for each device in the project. This data is called **naming data**  390].

To enable the naming data to be transmitted to the devices within the project during device naming, they must be identifiable on the network. This is achieved using fixed data stored in each device as a factory setting. This data is called **device identification data**  389].

During device naming, a distinction is made between the following device types:

- ▶ **PSS 4000-exclusive devices**  392]
PSS 4000-exclusive devices support all device identification data and all naming data.
- ▶ **PSS 4000-operable devices**  397]
PSS 4000-operable devices support part of the device identification data and naming data.



INFORMATION

If networks are connected via routers, device naming can only be performed centrally if the system variable `PILZ_SNP_MULTICAST_TTL` is configured accordingly (see **Network with routers**  194]).

11.8.1 Device identification data

PSS 4000-exclusive devices: PSS u2 systems, PSSu systems, PSS67 devices

Device identification data	Meaning	Example
Product type	Kind of device (e.g. head module of a PSSu system)	PSSu H PLC1 FS SN SD
Order number	6-digit number by which the device can be ordered at Pilz (e.g. head module of a PSSu system)	312070
Serial number	6-digit number by which the device can be identified at Pilz (e.g. head module of a PSSu system)	100076

PSS 4000-operable devices from Pilz: Operator terminals PMI

Device identification data	Meaning	Example
Product type	Kind of device	PMI v812
Order number	6-digit number used to order the device at Pilz	266812
Serial number	6-digit number by which the device can be identified at Pilz	002020

PSS 4000-operable devices from a third-party manufacturer: PC

Device identification data	Meaning
Product type	Kind of device: PSS 4000-PC
Order number	"0" is used as the order number
Serial number	When the PSS 4000 firmware is first started, a random 6-digit number is generated, which is then used internally as a serial number for various identification processes

11.8.2 Naming data

PSS 4000-exclusive devices: PSS u2 systems, PSSu systems, PSS67 devices

Naming data	Default setting	Meaning
Device name	Combination of a prefix (e.g. PSS), 6-digit order number and 6-digit serial number Format: <Prefix>_<Order number>_<Serial number> Example: PSS_312070_100076	▶ Uniquely identifies a device within the project ▶ Used to identify the device on the network when the project is downloaded directly ▶ Forms part of an address (e.g. I/O mapping)
IP Address	IP address in accordance with the Ethernet convention (see Factory default settings for network parameters  790)	Used in conjunction with the serial number to identify the device on the network during device naming
Subnet mask	Settings in accordance with the Ethernet convention (see Factory default settings for network parameters  790)	In accordance with the Ethernet convention
Gateway address	Settings in accordance with the Ethernet convention (see Factory default settings for network parameters  790)	▶ In accordance with the Ethernet convention ▶ Available as an option
Address of preferred DNS server		
Address of alternative DNS server		

PSS 4000-operable devices from Pilz: Operator terminals PMI

Naming data	Default setting	Meaning
Device name	Combination of the prefix "PSS", the 6-digit order number and 6-digit serial number Format: <Prefix>_<Order number>_<Serial number>	▶ Uniquely identifies a device within the project ▶ Used to identify the device on the network when the project is downloaded directly
IP Address	IP address in accordance with the Ethernet convention	Used in conjunction with the serial number to identify the device on the network during device naming
Subnet mask	Settings in accordance with the Ethernet convention	In accordance with the Ethernet convention
Gateway address	Settings in accordance with the Ethernet convention	▶ In accordance with the Ethernet convention ▶ Available as an option
Address of preferred DNS server		
Address of alternative DNS server		

PSS 4000-operable devices: PC

Naming data	Meaning, content and default setting
Device name	▶ Uniquely identifies a device within the project ▶ Used to identify the device on the network when the project is downloaded directly The default device name corresponds to the computer or host name defined by the user when setting up the Windows operating system.
IP Address	During device naming, used in conjunction with the generated serial number to identify the device on the network IP address in accordance with the Ethernet convention
Subnet mask	Subnet mask in accordance with the Ethernet convention
Gateway address	Available as an option, depending on the device
Address of preferred DNS server	Available as an option, depending on the device
Address of alternative DNS server	

11.8.3 Device naming for PSS 4000-exclusive devices

When device naming for PSS 4000-exclusive devices, a distinction can be made between the following cases:

► **PSS 4000-exclusive device in its delivery state or after an original reset**

The factory-set defaults are stored on the device that is in its delivery state or has undergone an original reset (see [Naming data](#) [📖 390]). The Auto-IP mechanism is activated (see [Auto-IP mechanism](#) [📖 130]). No device project is present.

► **Changing the device name**

The device name of a PSS 4000-exclusive device which has already been through the naming process can be changed retrospectively. In this case, user-specific naming data is already present on the device. A device project may be present. However, it will be deleted during device naming.

Whether or not the IP parameters are also changed is irrelevant for the device naming procedure.

► **Changing the IP parameters**

The IP parameters of a PSS 4000-exclusive device which has already been through the naming process can be changed retrospectively. In this case, user-specific naming data is already present on the device. A device project may be present. It is retained during device naming.

Whether just one or several IP parameters are changed is irrelevant for the device naming procedure. It is usually only the IP address that is changed. However, if the device name is also changed, the procedure will be as described under "Changing the device name".



INFORMATION

The description below assumes that all preparatory measures have already been carried out. These include:

- An SD card is inserted on the PSS 4000-exclusive device.
- The relevant naming data has been edited in PAS4000.

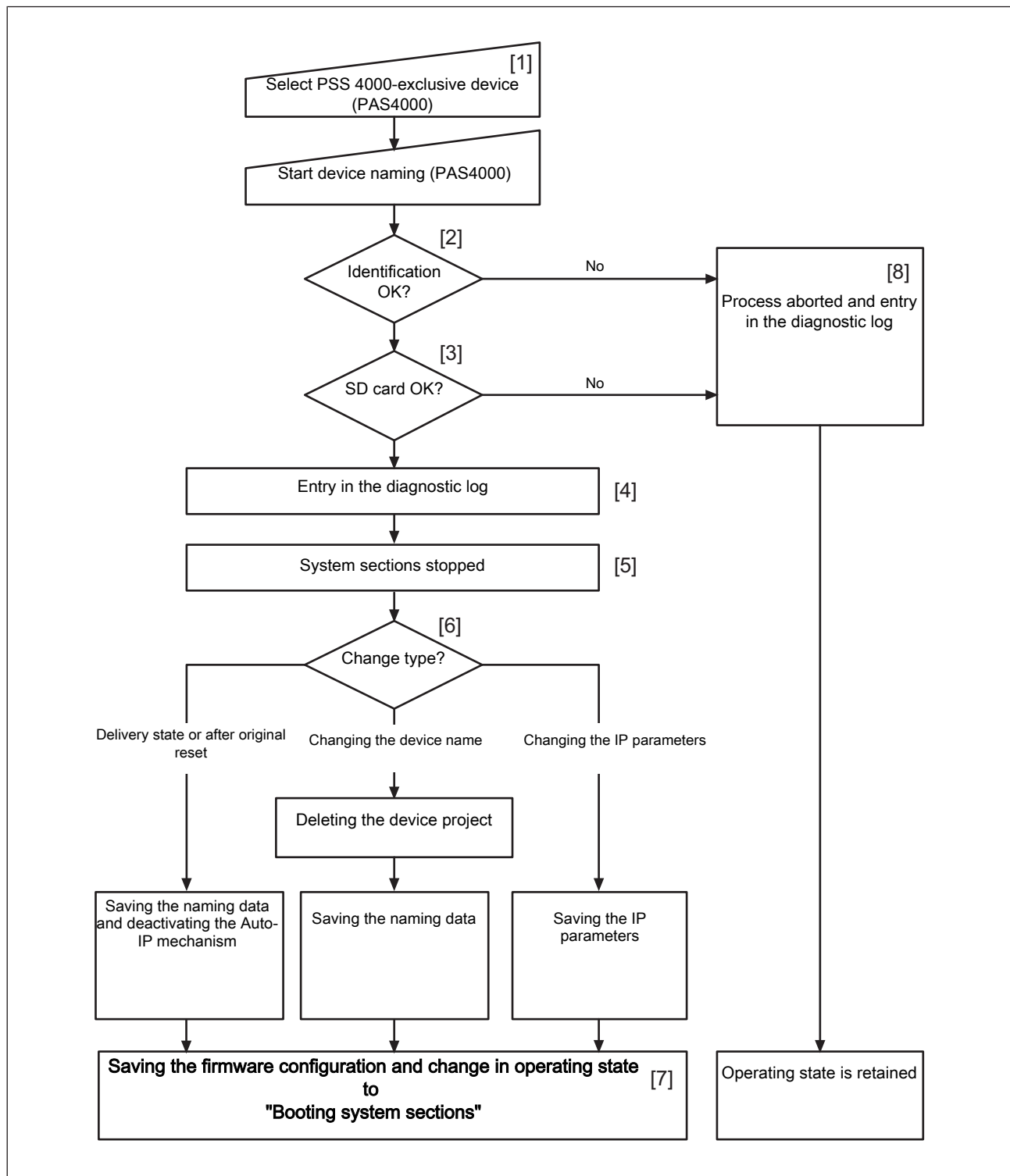


Fig.: Flowchart for device naming on PSS 4000-exclusive devices

Device naming procedure on a PSS 4000-exclusive device

The following description refers to the flowchart, Fig. "Flowchart for device naming of PSS 4000-exclusive devices".

[1]

The device for which device naming is to be performed is selected in PAS4000. A test can be carried out with PAS4000 to identify the selected device in the field. The test is called "Device identification".

1st generation PSS 4000 devices: During device identification, the LED for the SD card is activated on the selected device. The status LED (e.g. device's "SD CARD" LED) flashes orange.

2nd generation PSS 4000-exclusive devices: During device identification, the LEDs DIAG, MBUS, PWR, SD CARD, SNp and TSN are activated on the selected device. All LEDs flash green.

[2]

Once device naming has started, the device checks that the local serial number and IP address match the serial number and IP address stated as the destination for transferring naming data in PAS4000.

[3]

The device checks whether the SD card can be accessed, i.e. that the SD card is inserted and is not defective.

[4]

An entry in the diagnostic log shows that device naming has been carried out.

[5]

The device's system sections are stopped. On a PSS 4000-exclusive device this corresponds to the operating state "Device in STOP state without error".

[6]

The rest of the procedure depends on which naming data has been changed:

► **PSS 4000-exclusive device in its delivery state or after an original reset**

The factory-set default naming data in the device memory and on the SD card are replaced by the configured naming data. The Auto-IP mechanism is deactivated.

► **Changing the device name**

The existing naming data is deleted from the device memory and SD card. If the SD card already contains a device project, that device project is deleted from the SD card. On devices with an FS resource, the FS check sums of the device project and project name are also deleted from the device memory. The new naming data is then stored on the SD card.

► **Changing the IP parameters**

The existing naming data and any existing device project are retained. The IP address and all other IP parameters are deleted from the device memory and SD card. The new IP address and all other IP parameters are then stored on the SD card.

[7]

Before the device switches to the operating state "Bootling system sections", on PSS 4000-exclusive devices the firmware configuration is transferred to the device together with the device naming data. The firmware configuration includes the property **OPC Server port activated**, for example.

If device naming was performed with configured device data, then the firmware configuration is stored on both the device and the SD card. Otherwise, the firmware configuration stored on the device will also be saved on the SD card as a backup copy.

The device automatically switches to the Operating status "Bootling system sections".
The rest of the procedure depends on which naming data has been changed:

► **PSS 4000-exclusive device in its delivery state and changing the device name**

During the operating state "Bootling system sections", one of the effects is that the new naming data is loaded from the SD card into the device memory. The device starts up without a device project. The status LED of the SD card flashes green.

For the device to obtain a device project, a project must be downloaded (see [Direct project download](#)  401) and Indirect project download).

► **Changing the IP parameters**

During the operating state "Bootling system sections", one of the effects is that the new IP parameters are loaded from the SD card into the device memory. As soon as a device project is available, the device is started up with the device project.

If other devices in the network exchange data with the device, then a new project must be downloaded following the change to the IP address and/or other IP parameters. If not, the other devices will no longer be able to access the device.

[8]

If any of the checks are negative, the process is aborted. An entry will be added to the diagnostic list and the device's operating state remains unchanged.



NOTICE

No device naming in the operating state "Major FS error"

If a PSS 4000-exclusive device is in the operating state "Major FS error" and device naming is triggered in PAS4000, then device naming is not performed for this device.

11.8.3.1 Error behaviour during device naming

If any error should occur during device naming, the following error behaviour applies generally:

- ▶ Device naming process is aborted
- ▶ All system sections of the PSS 4000-exclusive device switch to a safe state
- ▶ All FS outputs on the PSS 4000-exclusive device switch to a safe state.
- ▶ An entry is added to the diagnostic list.

A cold start/reboot command must be executed for the PSS 4000-exclusive device. In this case, the PSS 4000-exclusive device will start up with the "old" naming data and, if available, the "old" device project.

11.8.4 Device naming for PSS 4000-operable devices

When device naming for PSS 4000-operable devices, please note the following:

► **Device name**

The device name edited during device naming in PAS4000 will appear only in PAS4000, in the diagnostic list and in the diagnostic log. A device name assigned at operating system level (computer name), will not be overwritten during device naming.

► **IP address**

The IP address at the device's operating system level is **not** overwritten by device naming. Please note the following for the IP address of PSS 4000-operable devices:

- The device's IP address must be defined during the hardware configuration and device naming in PAS4000, to enable the other devices in the PSS 4000 project to communicate with this device.
- The IP addresses that are defined in PAS4000 during the hardware configuration and device naming must match the IP address that has been configured at the device's operating system level.
- If the IP address is to be changed, the change must be made at the operating system level of the device (Microsoft Windows 10: Enter "Network connection" in the Windows search field and select **View network connections**. Right-click on the adapter that is used to establish the connection to the PSS 4000 project and select **Properties**. Select **Internet Protocol Version 4 (TCP/IPv4)** on the **Networking** tab and click on **Properties**.
- If it is necessary for a PSS 4000 project to change the IP address of a PSS 4000-operable device at operating system level, other devices that do not belong to the PSS 4000 project will no longer be able to access the PSS 4000-operable device via any existing Ethernet communication relationships. The IP address for these Ethernet communication relationships must be adapted in the relevant devices.



NOTICE

In a project, if a PC is both a PAS4000-PC (PC on which PAS4000 is installed) and a PSS 4000-PC (PC with OPC Server), the IP address of the Programming interface (PAS4000-PC) will also be used as the IP address for the PSS 4000-PC.

When device naming for PSS 4000-operable devices, a distinction can be made between the following cases:

► **PSS 4000-operable device during initial device naming**

After the PSS 4000 firmware has been installed, the factory set naming data is stored (see [Naming data](#) [390]). No device project is present.

► **Changing the device name**

When the device name is changed retrospectively, user-specific naming data will already be present on the PSS 4000-operable device. A device project may be present. However, it will be deleted during device naming.

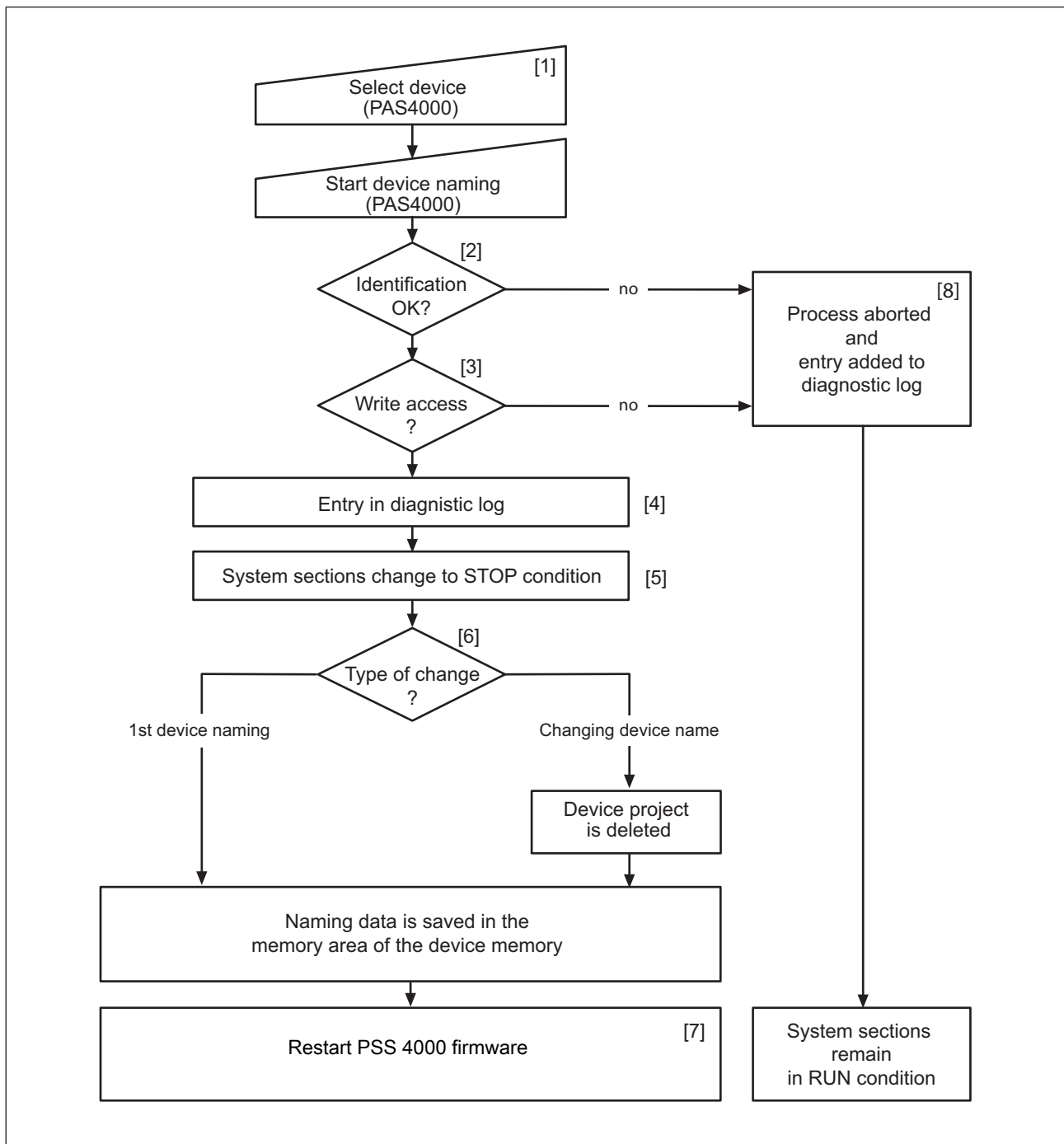


Fig.: Flowchart for device naming on PSS 4000-operable devices

Device naming procedure on PSS 4000-operable devices

The following description refers to the flowchart, Fig. "Flowchart for device naming on PSS 4000-operable devices".

[1]

The PSS 4000-operable device on which device naming is to be performed is selected in PAS4000. A test can be carried out with PAS4000 to identify the selected device. The test is called "Device identification". During device identification, a pop-up window with the fol-

lowing content is opened on the selected device:

PSS 4000 Device name: <Device name>.

The pop up window does not disappear until device identification is explicitly ended.

[2]

Once device naming has started, the device checks that the local serial number and IP address match the serial number and IP address stated as the destination for transferring naming data in PAS4000.

[3]

A test is carried out to check whether "write" privilege is available for the relevant memory area in the device's security settings.

[4]

An entry in the diagnostic log shows that device naming has been carried out.

[5]

The device stops the system sections of the PSS 4000 firmware (e.g. OPC Server, diagnostic server).

[6]

The rest of the procedure depends on which naming data has been changed:

► **PSS 4000-operable device during initial device naming**

The factory-set naming data is deleted from the relevant memory area of the device. The new naming data is then stored in the memory area. A project must then be downloaded, so that the device obtains a device project.

► **Changing the device name**

The existing naming data and any existing device project are deleted from the relevant memory area of the device. The new naming data is then stored in the memory area. A project must then be downloaded, so that the device obtains a device project.

[7]

After device naming, the PSS 4000 firmware on the PC or operator terminal PMI must be restarted manually in order for the naming data to be transferred.

[8]

If any of the checks are negative, the process is aborted. An entry will be added to the diagnostic list and the system sections of PSS 4000 firmware (e.g. OPC Server, diagnostic server) remain in a RUN state.

11.9 Device naming when PSS 4000-exclusive devices are exchanged

When a PSS 4000-exclusive device is exchanged it is not strictly necessary to undergo a new device naming procedure via PAS4000. The device naming data, the device project and the firmware configuration on the existing device can be transferred to the exchange device by re-using the existing SD card for the exchange device. If the OPC UA Server has been activated for the device, activation of the OPC UA Server is also transferred when the SD card is reused.



INFORMATION

If the PSS 4000-exclusive device is used for safety-related applications, when exchanging devices it is absolutely essential that you comply with the information regarding changes, as stated in the "PSS 4000 Safety Manual".

Prerequisites

The exchange device must still be in its delivery state or the device must undergo an original reset prior to the exchange.

Procedure

1. Switch the existing device to a STOP state and remove the supply voltage connectors.
2. Uninstall the existing device.
3. Remove the SD card from the existing device and insert it into the exchange device.
4. Install the exchange device.
5. Plug the voltage supply connectors into the exchange unit.
6. Follow the procedure below to transfer the device naming data and device project:
 - Press the reset button on the exchange device for longer than 5 s
 - Perform deliberate operator action for device naming:
Release the reset button - press the reset button - release the reset button
The deliberate operator action must be performed within 10 s, otherwise the transfer is aborted.

Once the naming data, device project, firmware configuration and activated OPC UA Server have been transferred, the exchange device is restarted; the behaviour is the same as for a cold start/reboot command.

Note: If the PSS 4000-exclusive device is a device on which the OPC UA Server is activated, additional measures are required. Use the Pilz NET Configurator to set up the certificates for the new device correctly.

11.10 Project download

11.10.1 Direct project download

During the direct project download the project is divided into device projects and downloaded to the PSS 4000-exclusive devices and, where available, to the project's PSS 4000-operable devices.

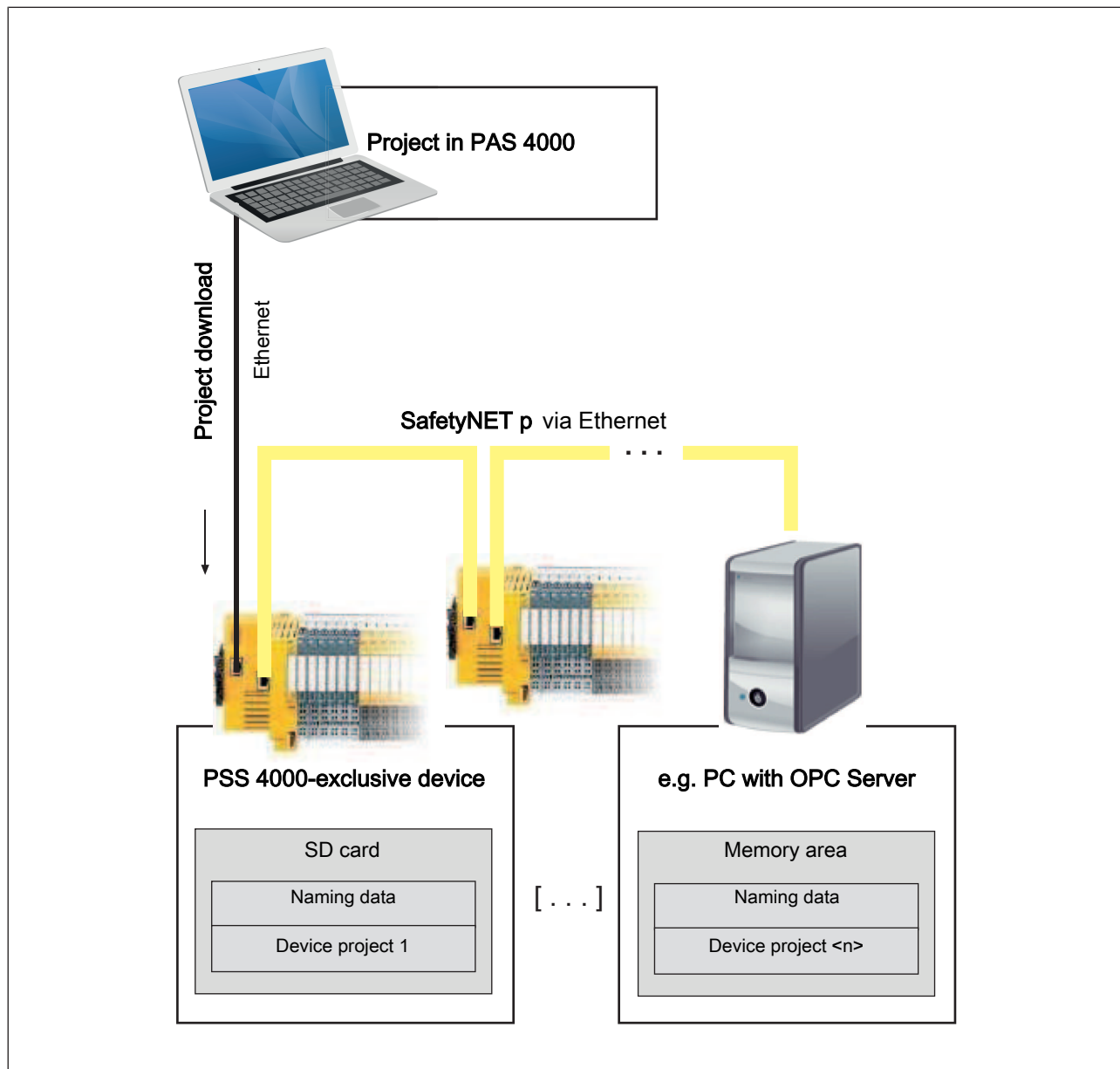


Fig.: Principle of direct project download (example)

During the project download, a backup copy of the project can also be downloaded onto a device in the project. The download of the backup copy does not influence the download of the project data. If the download of the backup copy fails, downloading the project is still executed.

Prerequisites

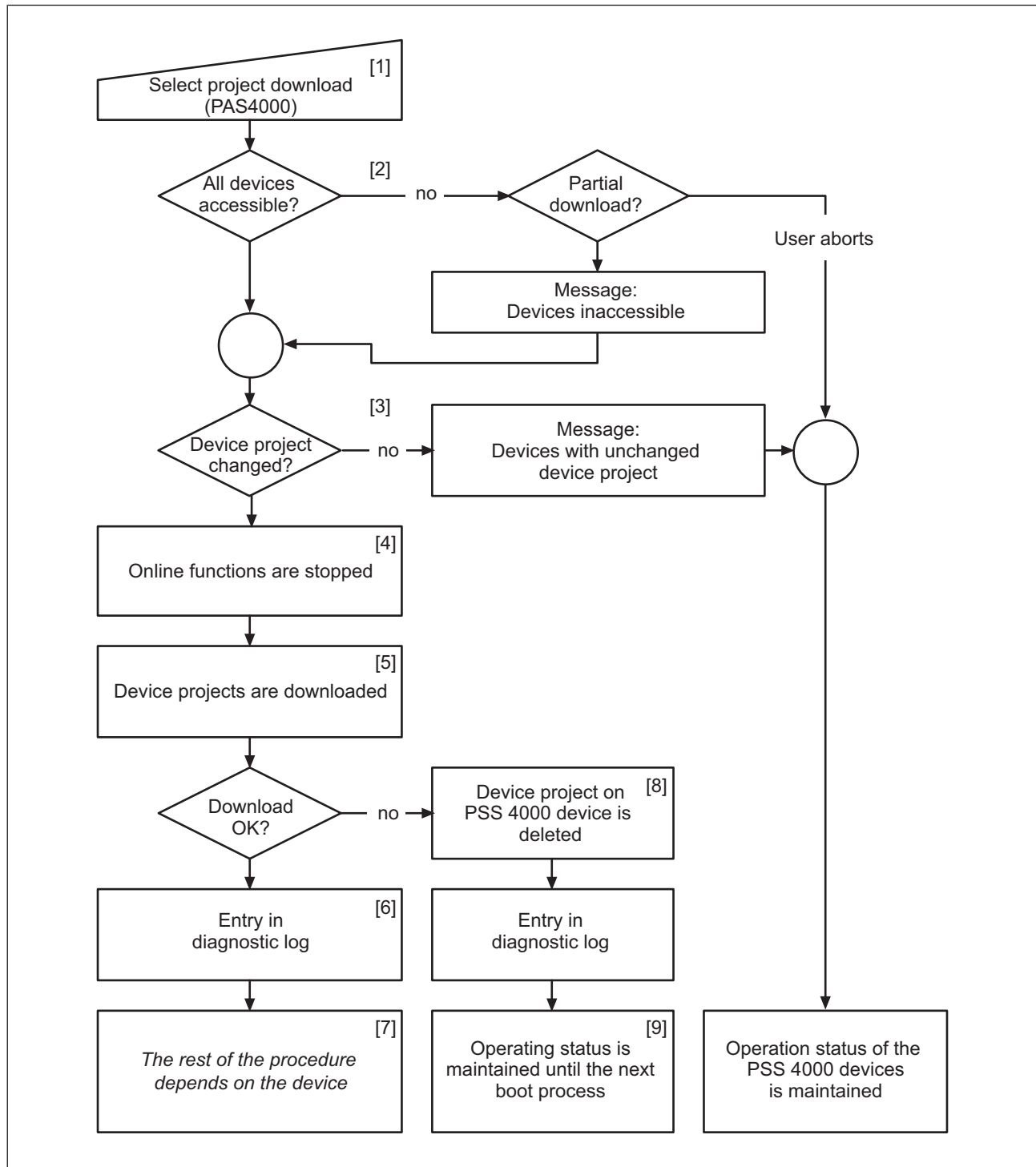
- ▶ There must be a physical connection (Ethernet cable) between the PAS4000-PC and the network containing the project's PSS 4000 devices (see Establish connection between PAS4000-PC and devices).
- ▶ The project must be activated in PAS4000.
- ▶ For each device in the project, the PSS 4000 firmware version that is configured in the hardware configuration must be lower than or equal to the actual PSS 4000 firmware version on the device.
- ▶ Device naming must have been performed (see Perform device naming in PAS4000).
- ▶ The project must have been built (see Build project).
- ▶ On project devices with SD card, the SD card is inserted.
- ▶ There is write access to the relevant memory areas;
e.g. the privilege for "Write" is set in the security settings for the relevant memory area of a PC.

**NOTICE**

Please note that the direct download for a PSS 4000 device will not be performed if an internal error is detected in any of its system sections. An internal error leads to a corresponding operating state on the affected system section(s):

- Operating state: Major FS error
Affected system sections: FS resource, FS module bus, FS SafetyNET p and safe communication channel of a system section for external communication (e.g. PROFIsafe from the PROFIBUS-DP Slave with PROFIsafe system section):
- Operating state with communication via SafetyNET p: Major FS+ST error
Affected system sections: FS SafetyNET p and ST SafetyNET p:
- Operating state with external communication: Major FS+ST error
(This operating state can only occur if the system section has a safety-related communication channel; e.g. PROFIsafe from the PROFIBUS-DP Slave with PROFIsafe system section)
Affected system section: System section for external communication (e.g. PROFIBUS-DP Slave with PROFIsafe).

Procedure for direct project download



[1]
Direct project download is selected in PAS4000.

[2]
PAS4000 checks that all the project's devices are accessible (e.g. devices are switched on, network connection cables to the devices are plugged in). If some of the project's devices are inaccessible, the following options are available:

- User aborts the direct project download
The current operating state of all devices is maintained.
- Direct project download is continued
All accessible devices are selected for the direct project download. The devices excluded from the download will be displayed in PAS4000.

[3]

PAS4000 checks each device to see if there is any difference between the existing device project and the new device project.

If a device project is unchanged, the device is excluded from the download. Devices on which the device project is unchanged are displayed in PAS4000. The current operating state of these devices is maintained.

[4]

The online functions (e.g. force variables, watch variables) are stopped. The device remains in its current operating state.

[5]

The device projects are downloaded for the **relevant** devices:

- PSS 4000-exclusive devices
If a device project already exists on the SD card of a PSS 4000-exclusive device, the device project is deleted from the SD card and the FS check sums from the device project and the project name are deleted from the device memory. The new device project is then stored on the SD card.
If online changes are active on a PSS 4000-exclusive device, these will be deactivated by the download.
- PSS 4000-operable devices
If available, on a PSS 4000-operable device the device project is deleted from the relevant memory area first and then the new device project is stored in the memory area.

[6]

Once a direct project download has been performed successfully, an entry is made in the diagnostic log.

[7]

Once a device project has been successfully downloaded, the rest of the procedure depends on the device:

- PSS 4000-exclusive devices
The device is restarted automatically.
- PSS 4000-operable devices
The PSS 4000 firmware on the device is restarted automatically.

Please note when starting a PSS 4000-exclusive device:

- The effect of start-up is that of a warm reset.
- PSS 4000-exclusive devices that have been stopped manually prior to the project download via one of the stop commands will not be started automatically. These devices will need to be restarted manually.
- If the user program from the old device project already contained non-volatile variables and/or volatile variables, the device will start up with the following values:

Case	Variable with the attribute RETAIN PERSISTENT	Volatile variable [1]	Application parameters
Variable is available in the user program unchanged (resource assignment, FQN and data type are the same)	Saved value	Variable is set to initial value	Value configured for the application parameter
Variable in the user program is new	Variable is set to initial value		
Variable is no longer available in the user program	Variable is deleted		Application parameter is deleted

[1] If a non-retentive variable is used as the application parameter, then the "Application parameter" column applies.

Note

When the project is downloaded to the devices, the content of a variable with the attribute RETAIN PERSISTENT is retained if

- ▶ the resource assignment of the block in which the variable is used,
- ▶ the FQN (full name in the PSS 4000 namespace) of the variable and
- ▶ its data type

are unchanged compared with the previous download.

[8]

Various errors may occur as the device project is downloaded to a PSS 4000 device.

Examples:

- ▶ User aborts
- ▶ Timeout
- ▶ Error occurs while saving the device project

Effect

On PSS 4000-exclusive devices, the device project is deleted from the SD card and the project name and FS check sums from the device project are deleted from the device memory. On PSS 4000-operable devices, the device project is deleted from the relevant memory area. An entry is added to the diagnostic log.

Remedy

Download project again

[9]

The device's operating state is maintained until the next boot process. The next time the system is booted (e.g. after a cold start/reboot command), the device will start up without a device project.



NOTICE

Guaranteeing the overall data consistency of a project after the direct project download

After a project download, the data on a project device is automatically consistent; i.e. on a device, the data in its device project is always consistent, even if the project download was aborted or failed.

If a project download was performed successfully and all project devices were accessible, then the device projects on all the devices of a PSS 4000 project will also be consistent between each other.

If the project download was aborted or failed, or not all the project devices were accessible, then it is the user's responsibility to guarantee that the device projects of all devices are consistent between each other.

In this case, proceed as follows:

- Check that the same project status has been committed on all the project's devices.

To do this, display the online information for each project device in the Online Network Editor of PAS4000. Check that the project information "Project name" and "Built on" matches on all project devices.

- If the project information for the devices does not match, download the project again. Make sure that all the project's devices are accessible and then download the project again.

11.10.2 Indirect project download

With an indirect project download, the project is divided into device projects. The device projects are stored on SD cards or in a directory, along with the naming data and firmware configurations. To complete the indirect download, the user must transfer the device projects to the devices manually, e.g. by inserting the SD card into a device and performing the deliberate operator action.

In this way the data can be prepared for operation without there being a connection between PAS4000 and the devices.



INFORMATION

Indirect download cannot be used for PMIs. PMIs can only obtain their device project via a direct project download.

There are two options:

► Perform indirect project download on SD cards

The device projects, the corresponding naming data and the firmware configurations are stored on the SD cards that will later be inserted into the device to allow the project to run. That way the SD cards can be prepared for operation without a connection between PAS4000 and the devices. This procedure is particularly useful for PSS 4000-exclusive devices (e.g. PSSu systems). SD cards can also be created for PCs, but these SD cards cannot be used directly (the data must be copied manually from the SD card to the PC).

► Perform indirect project download to a directory

All device projects, the corresponding naming data and the firmware configurations are stored in a directory. The directory may be on a local drive, on a network drive or on a mass storage device, such as a USB stick for example. From there the data can be copied to the SD cards, which are inserted into the device to allow the project to run. PC device projects can be copied to the PCs.

With the action "Indirect download to a directory", it is also possible to create an SD card, on which the device projects, the corresponding naming data and the firmware configurations for all the project's PSS 4000-exclusive devices (e.g. PSSu systems) are stored. If at any time an SD card is needed for a certain device, the relevant data must be moved to the SD card and then the suitable SD card is already available.



NOTICE

Guaranteeing the overall data consistency of a project after the indirect project download

After an indirect project download, the data on a project device is automatically consistent; i.e. on a device, the data in its device project is always consistent, even if the transfer of the device project from the SD card failed or files were damaged as they were copied.

It is the user's responsibility to ensure that the device projects of all the devices are consistent with each other.

In this case, proceed as follows:

- Check that the same project version has been transferred on all the project's devices.

To do this, display the online information for each project device in the Online Network Editor of PAS4000. Check that the "Project name" and "Built on" project information matches on all project devices.

As an alternative to PAS4000, "Project name" and "Built on" can be interrogated using the PSS 4000 OPC Server UA.

- If the project information on the devices does not match, perform the indirect project download again in PAS4000 and distribute the device projects on the devices again.

11.11 Sealed FS blocks

Sealed blocks are a programming aid. Sealed blocks cannot be modified and are safe from manipulation.

Sealed FS blocks process safety-related tasks that are required frequently (e.g. E-STOP).

They can be created by the user or are available from Pilz.

The use of certified, sealed FS blocks can speed up the certification process on projects, as the user program they contain will not require re-testing, even when used on machinery that is subject to mandatory inspection. There are

► Sealed user blocks

Additional information may be stored for sealed user blocks. This includes, for example:

- Name of the person who tested the sealed block
- Company name
- Name of the notified body
- Certification number

► Sealed Pilz blocks

Pilz provides various blocks which have been tested and certified by a notified body (e.g. BG, TÜV).

It is preferable to use blocks that have been certified by a notified body. You should check that the assumed operating conditions for these blocks match the conditions required by the user program.

12 Mode of operation

12.1 Communication between PAS4000 and PSS 4000 devices

In order to perform online actions, the computer on which PAS4000 is installed (PAS4000-PC) must be connected to the network containing the project's PSS 4000 devices (PSS 4000-exclusive devices and/or PSS 4000-operable devices).

Further information is available in PAS4000's online help.



NOTICE

Connecting the PAS4000-PC in Ethernet mode "Independent Mode"

If Ethernet mode "Independent Mode" has been configured for the physical ports (Ethernet interfaces) of a 2nd generation PSS 4000-exclusive device, the PAS4000-PC must be connected to the physical port with the lowest designation.

Further information is available under [Ports' Ethernet mode](#) [ 138].

12.1.1 Establish a point-to-point connection

With a point-to-point connection, PAS4000 can access an individual PSS 4000-exclusive/operable device as well as all PSS 4000-exclusive/operable devices on the network.

Principle

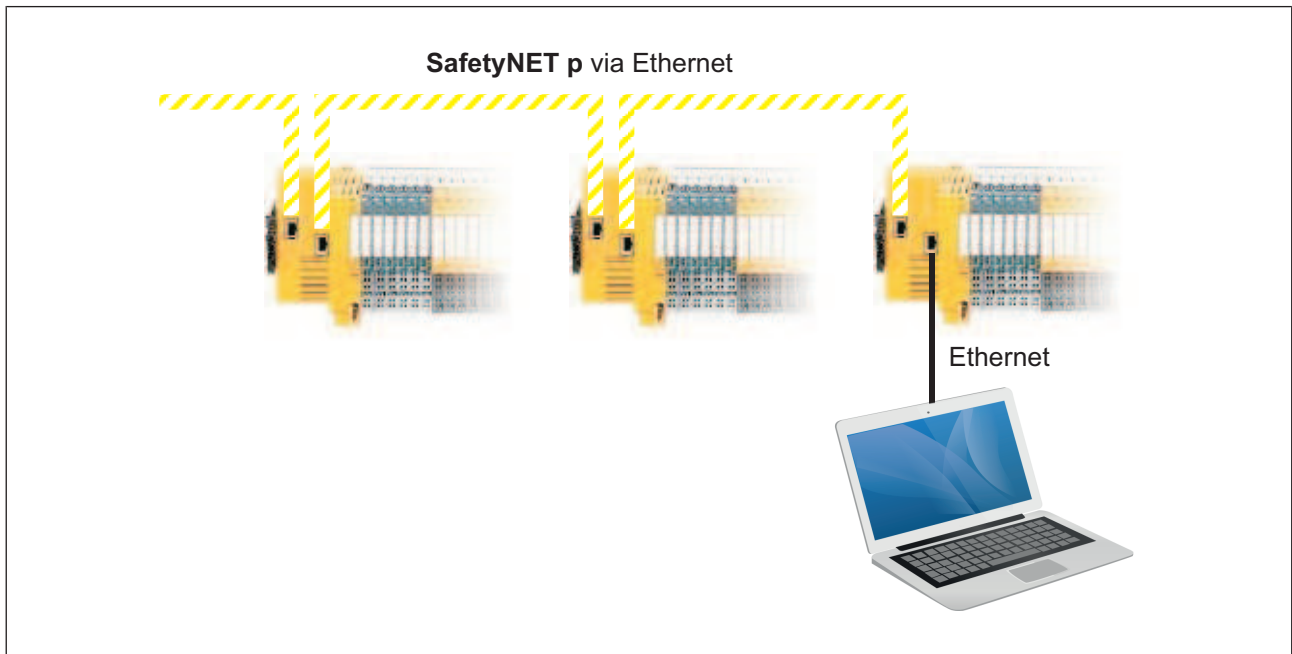


Fig.: Principle of a point-to-point connection with PAS4000

Please refer to the information on [Network without Routers](#) [ 193].

12.1.2 Establish a remote connection

With a remote connection, PAS4000 can access an individual PSS 4000-exclusive/operable device as well as all PSS 4000-exclusive/operable devices on the network.



NOTICE

Data security risk

Please note the special features regarding data security when communicating via remote connections (e.g. for diagnostics) (see "PSS 4000 Safety Manual").

Principle

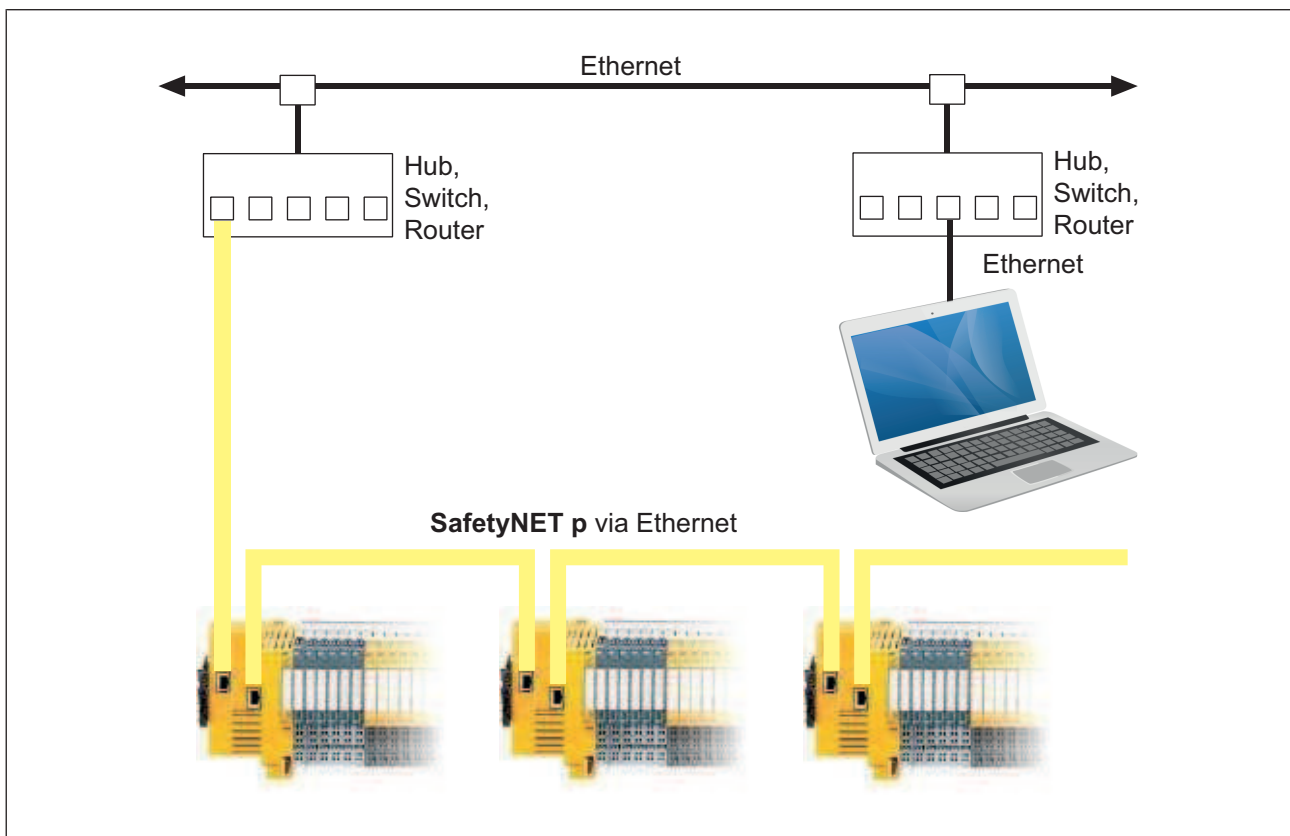


Fig.: Principle of a remote connection with PAS4000

Please note the restrictions when using Routers (see [Network with Routers](#) [194]).

12.2 Operating states and changes in operating state of a device

There are some operating states and changes in operating state that affect all system sections of a PSS 4000-exclusive device and others whose effects are restricted to one system section. A PSS 4000-exclusive device may contain all or just a subset of the system sections (see [System sections](#)  22]):

The description below deals exclusively with the operating states that affect all system sections of a PSS 4000-exclusive device.

The diagram "Operating states and changes in operating state of a PSS 4000-exclusive device" (see Flow chart) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The sections with Roman numerals I and II are connection points, which are referenced from other diagrams.
- ▶ The sections with Roman numerals III, IV, V a/b, VI a/b and VII refer to diagrams showing operating states and changes in operating state of system sections.

The following description assumes that there is a valid device project on the PSS 4000-exclusive device.

12.2.1 Flow chart

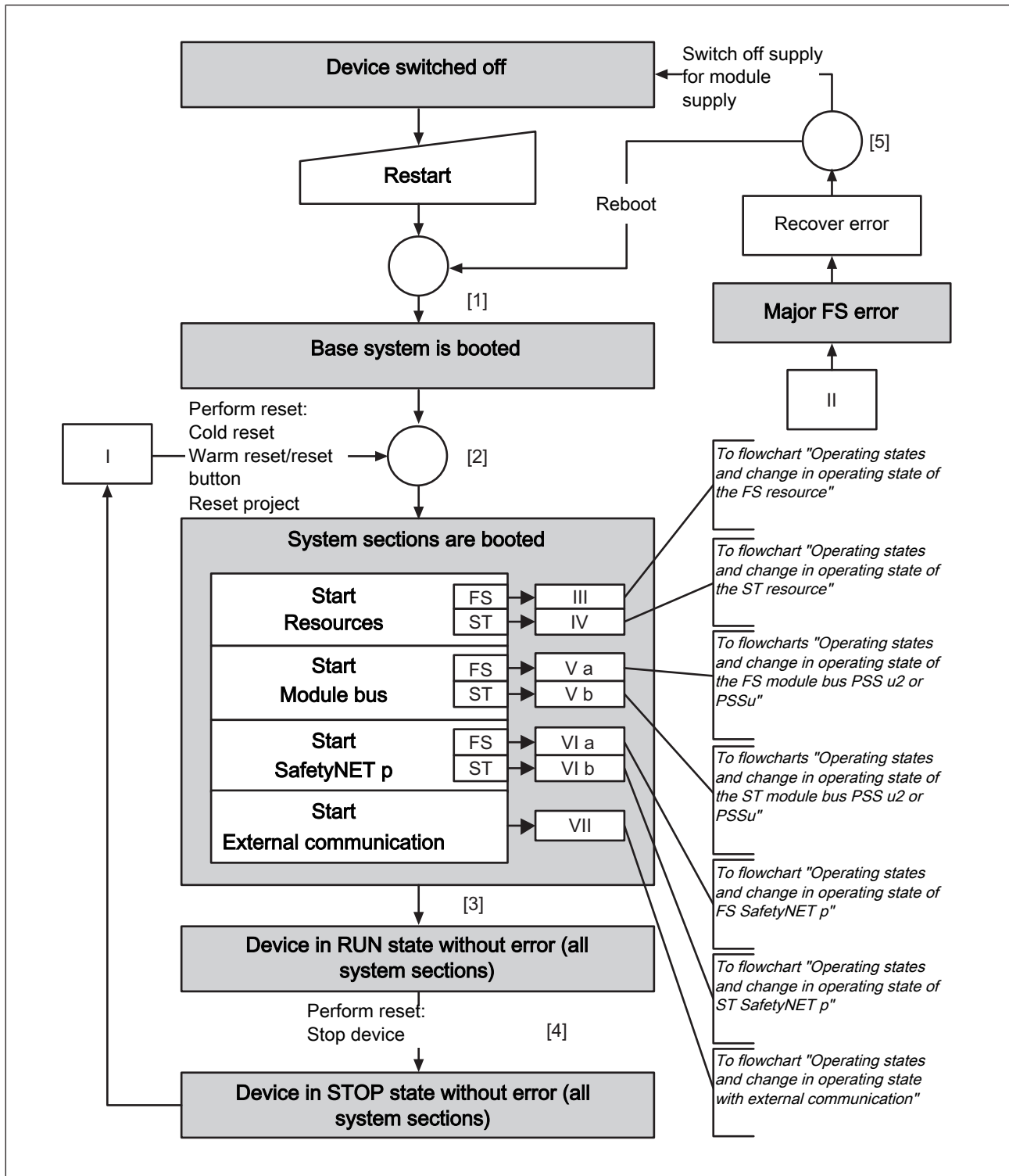


Fig.: Operating states and changes in operating state of a PSS 4000-exclusive device

12.2.2 Operating states of a device

A PSS 4000-exclusive device may assume the following operating states (see Flowchart):

- Operating state "Device switched off"
- Operating state "Bootling base system"
- Operating state "Bootling system sections"
- Operating state "Device in RUN state without error"
- Operating state "Device in STOP state without error"
- Operating state "Major FS error"



INFORMATION

The description of each operating state includes a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options](#) [ 540]).

12.2.2.1 Device switched off

"Device switched off" is the PSS 4000-exclusive device in a power-free state. Switching on the supply voltage for the module supply (cold start) switches the PSS 4000-exclusive device to the operating state "Booting base system".

12.2.2.2 Operating status "Booting base system"

Various start-up tests are performed (e.g. memory test). If there are any errors during the boot process, the PSS 4000-exclusive device will not start.

► Status LEDs on the device

- PSSu system: The "DIAG" LED on the device flashes red.
- PSS u2 system: The "DIAG" LED on the device lights up red.

Remedy

- Check the technical details (e.g. supply voltage for module supply, ambient temperature)
- It may be necessary to swap the device

12.2.2.3 Operating status "Bootling system sections"

"Bootling system sections" is either the phase

- ▶ After the base system is booted, until all system sections of a PSS 4000-exclusive device are in a RUN state or
- ▶ After a reset command is triggered (cold reset, reset button/warm reset), until all system sections of the PSS 4000-exclusive device are restored to a RUN state or
- ▶ After the base system is booted or a reset command is triggered, until a system section is in an error state

These error states are described in the respective system section.

Test phase

A test phase precedes the actual bootling of the system sections. In this test phase a hardware self test is carried out and the device firmware is verified. If an error is detected or the device firmware is corrupt, then the operating system stops the boot process.

In these cases, diagnostics are unavailable, the reset button has no function and all online actions are prevented.

- ▶ 1st generation PSS 4000-exclusive devices (e.g. PSSu systems):
In this state, the status LED "DIAG" flashes red.
- ▶ 2nd generation PSS 4000-exclusive devices (e.g. PSS u2 systems):
In this state, the status LED "DIAG" lights up red.

It is only possible to exit this state by switching off the voltage supply (restart the device).

If the various self tests have been performed without error, then the SD card is checked for consistency. If the journal of the journaling file system cannot be executed on the SD card, then the system sections are booted without the SD card (see [1] in the flowchart below).

Further information is available under [Journaling file system](#)  613] and [Bind SD card](#)  613].

If the reset button is pressed during the test phase, then the boot process stops after the tests and checks. The device switches to a recovery mode. Further information is available under Recovery mode.

Start the actual "Bootling system sections"

The SD card is used to decide the naming data, device project and firmware configuration with which bootling of the system sections is performed. Various tests are carried out for this purpose.

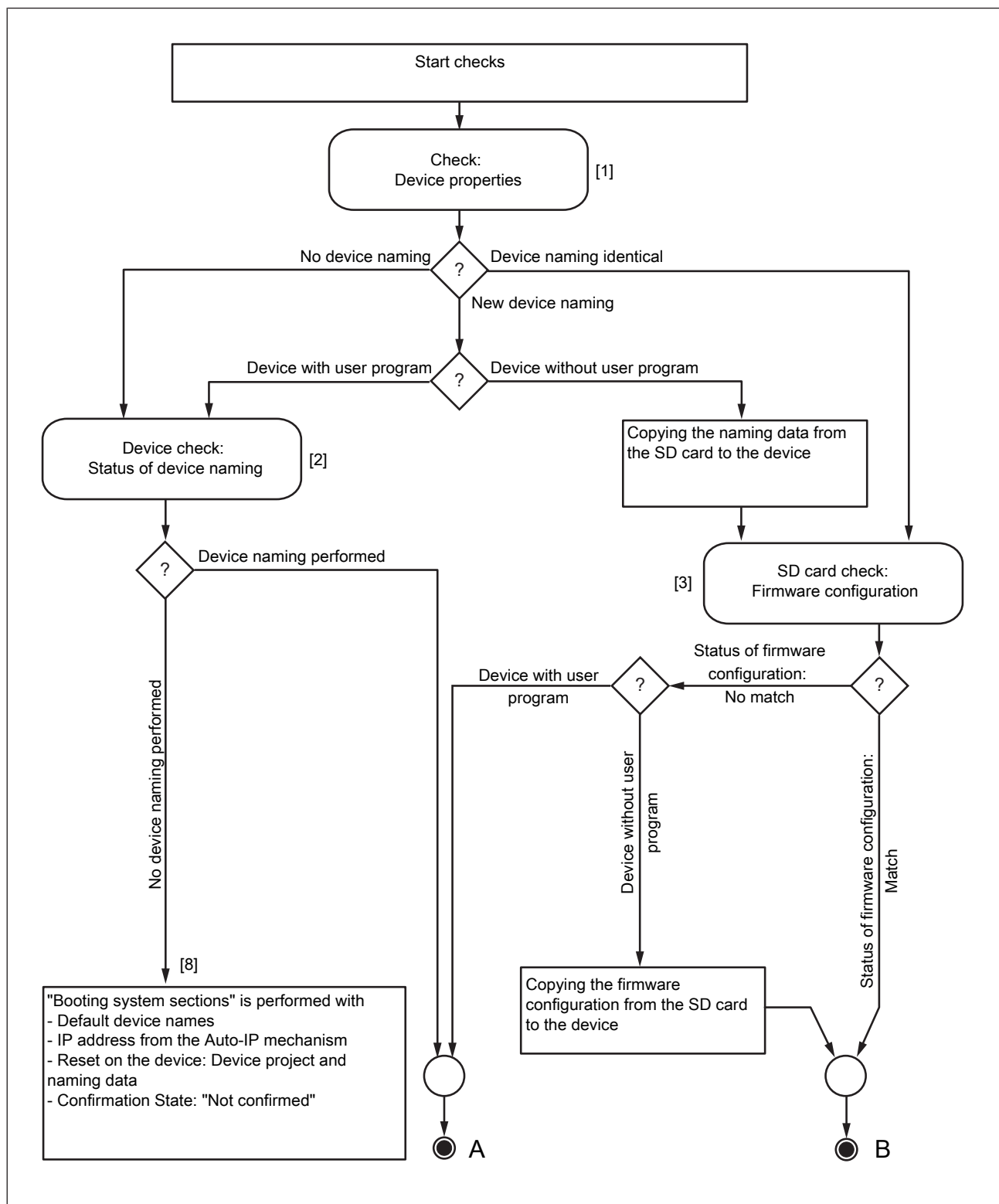


Fig.: Flowchart for checking the SD card (Part 1)

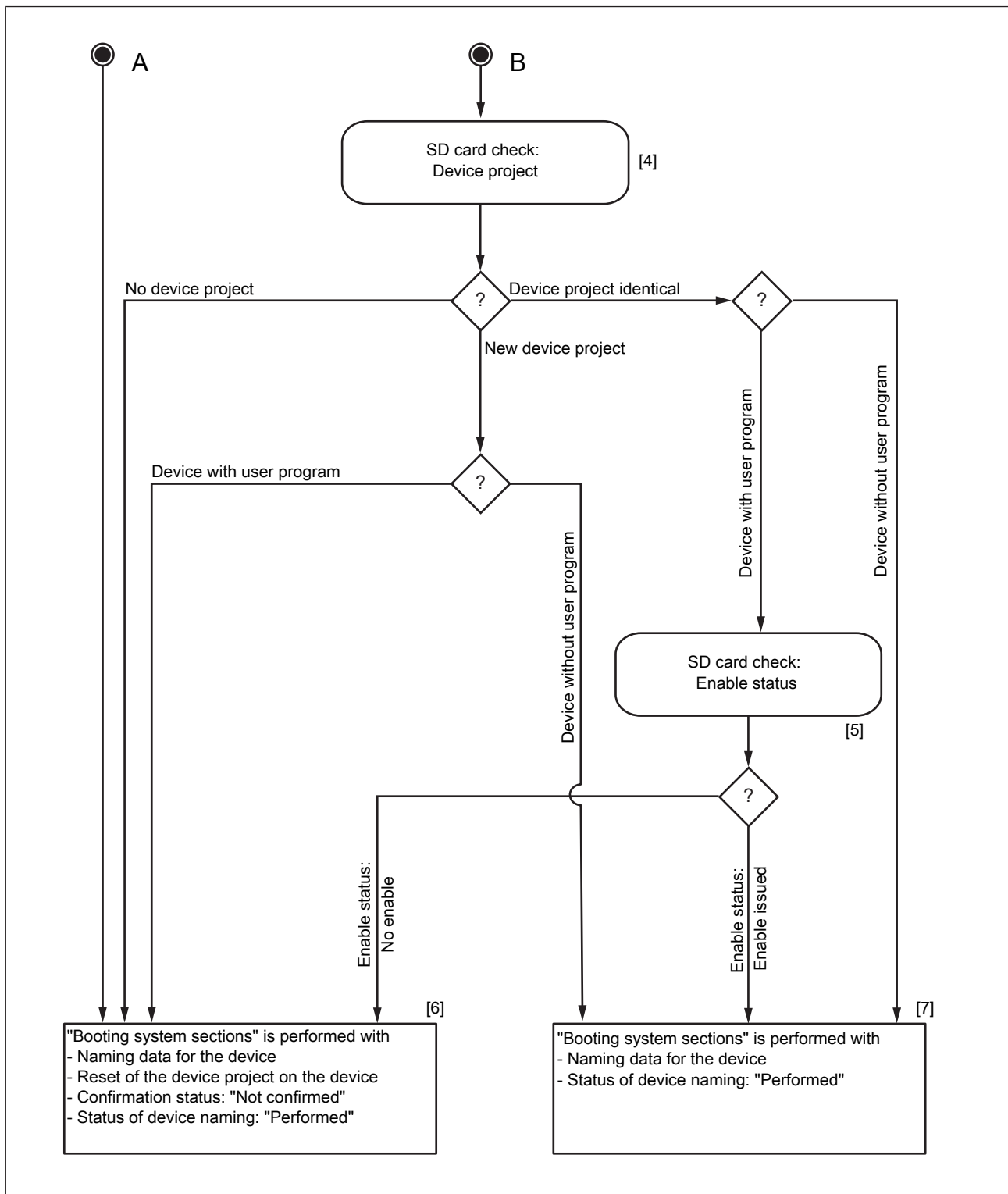


Fig.: Flowchart for checking the SD card (Part 2)

[1] Check: Device properties

A check determines whether the naming data on the SD card and device matches. For devices with a bound device project, a check is performed to see whether the project key matches the device key.

The check may produce one of the following test results:

► No device naming; i.e.

- SD card is not inserted or not intact
- SD card is empty or there is no naming data on the SD card
- The product type (see Device identification data) on the SD card does not match the device's product type

Further checks are carried out (see [2]).

► New device naming; i.e.

the naming data on the SD card does not match the naming data on the device.

A distinction is made between the following cases:

- It is a device on which a user program can be run.
Further checks are carried out (see [2]).
- It is a device on which no user program can be run or it is one of the following devices
 - PSSu H FS SN SD(-T)(-R)
 - PSSu H FS SN SD M12 (-T)(-R) or
 - PSS67 IO1 16FDI.

The naming data is copied from the SD card to the device.

Further checks are carried out (see [3]).

► Device naming identical; i.e.

- the naming data on the SD card matches the naming data on the device
- the product type (see Device identification data) on the SD card matches the device's product type

Further checks are carried out (see [3]).

[2] Device check: Status of device naming

A check determines whether device naming has already been performed for the device.

The check may produce one of the following test results:

► Device naming has not been performed for the device.

"Bootling of the system sections" occurs for the device with the settings in accordance with [8].

► Device naming has been performed for the device.

"Bootling of the system sections" occurs for the device with the settings in accordance with [6].

[3] SD card check: Firmware configuration

A check determines whether the firmware configuration on the SD card is identical to the firmware configuration on the device. Various settings are checked in the process. These settings include activation/deactivation of the OPC UA port, for example.

The firmware configuration is on the SD card in the directory /cfg (see [Memory structure](#)  610)).

The check may produce one of the following test results:

► Status of the firmware configuration: Match

All properties of the firmware configuration are identical. Further checks are carried out (see [4]).

► Status of the firmware configuration: No match

At least one of the firmware properties is not identical.

A distinction is made between the following cases:

- It is a device on which a user program can be run.

Bootling of the system sections occurs for the device with the settings in accordance with [6].

- It is a device on which no user program can be run or it is one of the following devices

- PSSu H FS SN SD(-T)(-R)
- PSSu H FS SN SD M12 (-T)(-R) or
- PSS67 IO1 16FDI

The firmware configuration is copied from the SD card to the device. Further checks are carried out (see [4]).

[4] SD card check: Device project

A check determines whether there is a device project on the SD card and whether the device project matches the device project on the device.

The check may produce one of the following test results:

► No device project; i.e.

- There is no device project on the SD card or
- The device project does not match the product type or
- The project's firmware version is too high or
- The device project is bound to another security SD card and not to the inserted security SD card or
- The project key of the device project on the SD card does not match the device key of the device or
- The option "Project requires suitable project key" is activated for the device but the device project on the SD card does not have a project key.

"Bootling of the system sections" occurs for the device with the settings in accordance with [6].

► New device project; i.e.

The project name or the check sum "FS device project" on the SD card is not identical to the project name or the check sum "FS device project" on the device.

A distinction is made between the following cases:

- It is a device on which a user program can be run.

Bootling of the system sections occurs for the device with the settings in accordance with [6].

- It is a device on which no user program can be run or it is one of the following devices

- PSSu H FS SN SD(-T)(-R)
- PSSu H FS SN SD M12 (-T)(-R) or
- PSS67 IO1 16FDI.

Bootling of the system sections occurs for the device with the settings in accordance with [7].

► Device project identical

The project name and the check sum "FS device project" on the SD card is identical to the project name and check sum "FS device project" on the device.

A distinction is made between the following cases:

- It is a device on which no user program can be run or it is one of the following devices
 - PSSu H FS SN SD(-T)(-R)
 - PSSu H FS SN SD M12 (-T)(-R) or
 - PSS67 IO1 16FDI.

Bootling of the system sections occurs for the device with the settings in accordance with [7].

- It is a device on which a user program can be run.
Further checks are carried out (see [5]).

[5] SD card check: Enable status

A check determines whether the user has issued the enable to transfer the device project from the SD card; i.e. the device project on the SD card is to be transferred and executed.

Note: The user can issue the enable as follows:

- [Direct project download !\[\]\(4695f05050b0d393767d0512587d4e50_img.jpg\) 401](#)
- Deliberate operator action to transfer the naming data and/or device project

A distinction is made between the following cases:

► Enable status: "Enable issued"

Bootling of the system sections occurs for the device with the settings in accordance with [7].

► Enable status: "No enable"

Bootling of the system sections occurs for the device with the settings in accordance with [6].

[6] The system sections are booted with the following settings:

- The device uses the device's naming data.
- The device performs a reset for the device project; i.e.
The device project on the device is deleted or the project name and the check sum "FS device project" on the device are deleted.
- The enable status is set to "No enable".

Note: As the device is starting without a device project and with the enable status "No enable", a project must either be downloaded directly (see [Direct project download](#)  401]) or the "Deliberate operator action to transfer the naming data and/or device project" must be performed (see Deliberate operator action to transfer the naming data and/or device project).

[7] The system sections are booted with the following settings:

- ▶ The device uses the device's naming data.
- ▶ The enable status is set to "Enable issued".
- ▶ The status of device naming is set to "Performed".
- ▶ The device uses the device's device project.

[8] The system sections are booted with the following settings:

- ▶ The naming data on the device is deleted; i.e.
The device activates the factory-set default naming data.
- ▶ The device activates its Auto-IP mechanism; i.e.
The device uses the IP address that was assigned via the Auto-IP mechanism.
- ▶ The device performs a reset for the device project; i.e.
If the device contains a device project, it is deleted or the project name and the check sum "FS device project" on the device are deleted.
- ▶ The enable status is set to "No enable"

Note: As the device is starting with the factory-set default settings, without a device project and with the enable status "No enable", device naming must be performed first (see Device naming via PAS4000) and then a project must either be downloaded directly (see [Direct project download](#)  401]) or the "Deliberate operator action to transfer the naming data and/or device project" must be performed (see Deliberate operator action to transfer the naming data and/or device project).



INFORMATION

When booting, all data and directories that are not relevant are deleted from the SD card, i.e.:

- data and directories that are not from the system and were stored by the user outside the /userfiles directory are deleted from the SD card.
- The following directories and files are retained on the SD card, along with the directories needed by functions of the PSS 4000 firmware:
 - /userfiles directory
 - CM-stick directory
 - "journal.jnl" file in the SD card's root directory
- Please note that only the directories of functions known to the device's PSS 4000 firmware remain on the SD card.
If the SD card of a device with a higher firmware version is used for a device, for example, the directories of all the functions available exclusively on the device with the higher firmware version will be deleted.

Behaviour while "Bootling system sections":

- ▶ If no faults are detected during the boot process, all system sections are started. The starts are executed in parallel (see Sections III ... VI Sections in the flowchart).
- ▶ If a fault is detected during the boot process, the affected system section switches to one of the error states. All unaffected system sections continue with the boot process and switch to a RUN state (see Sections III ... VII in the flowchart).
- ▶ The boot process has been completed when
 - All system sections have successfully loaded their configuration
"DIAG" LED: switches from "Flashing green" to "Lights up green"
 - A system section registers a fault and switches to an error state
"DIAG" LED on a PSSu system: switches from "Flashing green" to "Lights up orange", "Lights up red" or "Flashes red"
"DIAG" LED on a PSS u2 system: switches from "Flashing green" to "Lights up yellow", "Lights up red" or "Flashes red"
In these cases, unaffected system sections have still loaded their configuration successfully.
- ▶ Status LEDs
The status LEDs on the device provide information on the status of the system sections.

12.2.2.4 **Operating status "Device in RUN condition without error"**

This operating state is characterised by the fact that all system sections are in a RUN state and are operating without error (see Sections III ... VII in the flowchart).

► **Status LEDs**

The status LEDs on the device provide information on the status of the system sections.

12.2.2.5 Operating status "Device in STOP condition without error"

This operating state is achieved by triggering an explicit stop via PAS4000 (device is stopped).

Effects

- ▶ The system sections finish what they are currently processing. Any further processing is stopped. All system sections are in a STOP state. The behaviour during the stop state is described separately (see [Stop commands – Behaviour during the STOP state](#)  552).
- ▶ Status LEDs

The status LEDs on the device provide information on the status of the system sections.

Remedy

The following commands are available for starting the PSS 4000-exclusive device:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Start project (PAS4000)

12.2.2.6 Operating status "Major FS error"

If an error leads to the operating state "Major FS error", the affected sections of PSS 4000-exclusive device are ended as defined. A defined state is achieved.

Possible causes

- ▶ Plausibility error
e.g. irregularities between the microprocessors (different calculation results)
- ▶ Memory error
- ▶ Excessive network load, caused by devices that permanently or sporadically generate a high network load
e.g. printers, video cameras, devices that frequently transmit broadcast or multicast telegrams

Effects

- ▶ In the event of a major error in a safety-related system section, all other safety-related system sections are stopped and/or switched to a safe state:
 - The FS resource is in a STOP state and the FS resource tasks are no longer executed.
 - All hardware FS outputs are switched to a safe state.
 - FS process data communication via SafetyNET p is stopped.
- ▶ For process data whose values are determined via the FS resource and which is accessed via other resources, the substitute values are used with valid bits = FALSE.
- ▶ See PAS4000 online help for details of how to handle variables (volatile variables, non-volatile variables) whose values are determined via the FS resource
- ▶ Hardware ST outputs are unaffected
- ▶ Status LEDs on the device
The status LEDs on the device provide information on the status of the system sections.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the project diagnostic list and/or the diagnostic log
2. Recover error
3. Execute a restart/reboot command



NOTICE

No execution of commands in the operating state "Major FS error"

If a PSS 4000-exclusive device is in the operating state "Major FS error" and PAS4000 sends one of the following commands, then the device does not execute the command.

Affected commands:

- Device naming
- Direct project download
- Original reset
- Cold/warm reset
- Cold/warm restart
- One of the stop commands

In these cases, the device sends an error message to PAS4000.

12.2.3 Change in operating state of a device

This description of the change in operating state refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart).

[1] Change in operating state after a cold start/reboot command

When the module supply to a PSS 4000-exclusive device is switched on (cold start) or following a reboot command (PAS4000), the device initially switches to the state "Bootling base system", performs various start-up tests and then switches to the state "Bootling system sections".

Fundamentally, a change in operating state after "bootling the base system" can be enforced via a cold start/reboot command when the device is in any state (see Section II in the flowchart). For reasons of availability, we recommend that you only enforce a change in operating state when it is really necessary. For example, after recovering an error that has led to a major error.

[2] Change in operating state after a reset

Fundamentally, a reset (cold reset, reset button/warm reset or project reset) can be triggered when a PSS 4000-exclusive device is in any state (see Section I in the flowchart). The effect (e.g. how variables are handled) depends on which reset is used. After a reset, the device switches to the state "Bootling system sections".

For reasons of availability, we recommend that you only trigger a reset when it is really necessary.

The effects of the reset are described separately (see [\(Reset commands \[547\]\)](#)).

[3] Switching from "Bootling system sections" to "Device in RUN state without error"

This change of operating state takes place as soon as all system sections of the PSS 4000-exclusive device are in a RUN state and are operating without error.

Effects

- ▶ The millisecond timer is started.
- ▶ The content of non-volatile and volatile variables is documented under "Reset, restart, start and stop options" (see [\(Effects on local and resource global variables \[542\]\)](#)).
- ▶ All values in the PII/PIO are set to ZERO.

It's possible that the module bus may have already been started by the FS resource and/or ST resource. In this case, the PII may contain current input values.

- ▶ An entry is added to the diagnostic list.

[4] Switching from "Device in RUN state without error" to "Device in STOP state without error"

This change of operating state must be triggered explicitly via PAS4000 (e.g. stopping the device).

The effects of a stop are described separately (see [\(Stop commands \[552\]\)](#)).

[5] Switching from "Major FS error" to "Bootling system sections"

Once the error has been recovered, the PSS 4000-exclusive device can only be started via a cold start/reboot command. As a result of the cold start/reboot command the device switches to "Bootling base system", performs the start-up tests and then runs through the complete boot process for the state "Bootling system sections".

12.3 Operating states and change in operating status of the FS resource

The description below deals exclusively with the operating states of the FS resource of a PSS 4000-exclusive device, referring to the operating states of the device where necessary.

The diagram "Operating states and changes in operating state of the FS resource" (see [Flowchart !\[\]\(bd1a142de767a21e5362c595f844a4ff_img.jpg\) 432](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and changes in operating state of a device" (see [Flowchart !\[\]\(ce77bba2916ff045bdb9f4584b191293_img.jpg\) 432](#)), as the effects impact all system sections of the PSS 4000-exclusive device and not just the FS resource.
- ▶ The section marked with the Roman numeral III refers to the diagram "Operating states and changes in operating state of a device" (see [Flowchart](#)), as the FS resource is started when booting the system sections of a PSS 4000-exclusive device.

12.3.1 Flow chart

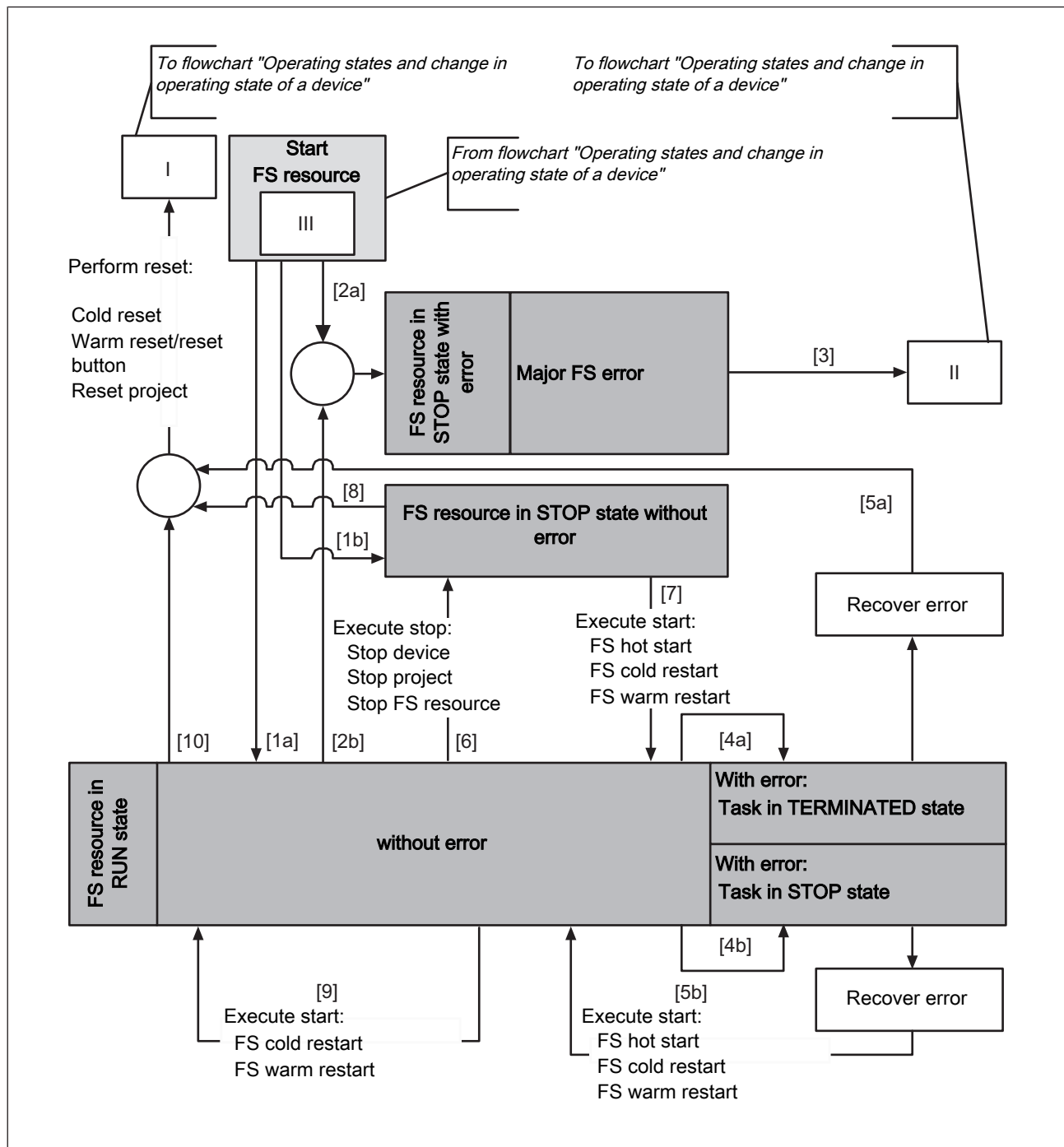


Fig.: Operating states and changes in operating state of the FS resource

12.3.2 Operating states of the FS resource

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The FS resource can adopt the following operating states (see [Flowchart \[📖 432\]](#)):

- ▶ Operating states "FS resource in RUN state":
 - FS resource in RUN state without error
 - FS resource in RUN state with error: Task in TERMINATED state
 - FS resource in RUN state with error: Task in a STOP state
- ▶ Operating state "FS resource in STOP state without error"
- ▶ Operating state "FS resource in STOP state with error":
 - Major FS error



INFORMATION

The descriptions of various operating states include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[📖 540\]](#)).

12.3.2.1 **FS resource in RUN condition without error**

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Restarting the device or the reboot command for the device via PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)
- ▶ After a start or restart via PAS4000
(FS cold restart, FS warm restart, FS hot start)

Effects

- ▶ In the operating state "FS resource in RUN state without error", the FS resource tasks are run without error.

12.3.2.2 FS resource in STOP condition without error

This operating state is achieved by triggering a stop via PAS4000 (device is stopped, FS resource is stopped, project is stopped).

Effects

- ▶ The behaviour of the hardware inputs, hardware outputs, process data and system sections while a resource is in a STOP state is described separately (see [Stop commands – Behaviour during the STOP state](#)  552]).
- ▶ The effects on local and resource global variables are described separately (see [Effects on local and resource global variables](#)  542]).

Remedy

- ▶ FS resource is (re)started:
 - FS hot start (PAS4000)
 - FS cold restart (PAS4000)
 - FS warm restart (PAS4000)
- ▶ Reset device:
 - Warm reset (PAS4000) or reset button (device)
 - Cold reset (PAS4000)
- ▶ Project is reset
 - Warm reset of project (PAS4000)
 - Cold reset of project (PAS4000)

12.3.2.3 FS resource in RUN condition with error: Task in TERMINATED condition

If an error leads to this operating state, the affected task is ended as defined. A defined state is achieved.

Possible causes

(see also IEC 61131-3, Table E1)

- ▶ Execution time error
e.g. timeout error because the execution time was exceeded

Effects

- ▶ The affected FS resource task is no longer executed, all other FS resource tasks continue to be executed.
- ▶ The FS resource continues to be in a RUN state
- ▶ The following applies for all O-PI variables that are declared as local variables or are referenced as VAR_EXTERNAL in the POU's of the task that has been stopped:
Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and valid bits = FALSE.
- ▶ The following applies for all O-PI variables that are created in Multi blocks as O-PI points or are referenced as VAR_EXTERNAL:
Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and valid bits = FALSE.
- ▶ The following applies for resource global variables (not O-PI variables) that are referenced via VAR_EXTERNAL in the blocks of the task that has been ended:
they retain the last valid, stored values and are not set to their initial values.
- ▶ If resource global variables are referenced via "VAR_EXTERNAL" in the blocks of a task that has not been ended, the value there can be overwritten. As a result of this overwrite, resource global O-PI variables will again be marked as valid.
- ▶ See PAS4000 online help for details of how to handle variables (volatile variables, non-volatile variables)

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or diagnostic log
2. Recover error
3. Perform reset:
Use warm reset or cold reset

12.3.2.4 FS resource in RUN condition with error: Task in STOP condition

If an error leads to this operating state, the affected task is stopped as defined. A defined state is achieved.

Possible causes

(see also IEC 61131-3, Table E1)

▶ **Runtime error**

e.g. a variable value exceeds the defined range, the numeric result of a function exceeds the value range for a data type, division by 0

Effects

- ▶ The affected FS resource task is no longer executed, all other FS resource tasks continue to be executed.
- ▶ The FS resource continues to be in a RUN state
- ▶ The following applies for all O-PI variables that are declared as local variables or are referenced as VAR_EXTERNAL in the POU's of the task that has been stopped:
Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and valid bits = FALSE.
- ▶ The following applies for all O-PI variables that are created in Multi blocks as O-PI points or are referenced as VAR_EXTERNAL:
Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and valid bits = FALSE.
- ▶ The following applies for resource global variables (not O-PI variables) that are referenced via "VAR_EXTERNAL" in the blocks of the task that has been stopped:
they retain the last valid, stored values and are not set to their initial values.
- ▶ If resource global variables are referenced via "VAR_EXTERNAL" in the blocks of a task that has not been stopped, the value there can be overwritten. As a result of this overwrite, resource global O-PI variables will again be marked as valid.
- ▶ See PAS4000 online help for details of how to handle variables (volatile variables, non-volatile variables)

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or diagnostic log
2. Recover error
3. Perform a reset or (re)start:
Use FS hot start, FS warm restart or FS cold restart
One of the reset commands can also be used. A reset command affects all the device's system sections and not just the FS resource.

12.3.2.5

Major FS error

If an error leads to this operating state, the affected sections of the FS resource are ended as defined. A defined state is achieved.

The effects of a major FS error impact all safety-related parts of a PSS 4000-exclusive device (see [Operating state "Major FS error" \[📖 428\]](#)).

12.3.3 Change in operating status of the FS resource

This description of the change in operating state assumes that the device contains a valid device project. The numbers refer to the change in operating state in the diagram "Operating states and change in operating state of the FS resource" (see [Flowchart \[📖 432\]](#)).



INFORMATION

The description of each change of operating state includes a list of the options for triggering the change in operating state. These options are described separately (see [Reset, restart, start and stop options \[📖 540\]](#)).

[1a] Switching from "Bootling system sections" to "FS resource in RUN state without error"

"Bootling system sections" is one of the operating states of a PSS 4000-exclusive device (see Flowchart). If no error is detected in a safety-related system section as the system sections are booted, the FS resource is started, switching to the state "FS resource in RUN state without error".

[1b] Switching from "Bootling system sections" to "FS resource in STOP state without error"

"Bootling system sections" is one of the operating states of a PSS 4000-exclusive device (see Flowchart). Under various circumstances the FS resource may not achieve the operating state "FS resource in RUN state without error", but changes instead to "FS resource in STOP state without error":

- ▶ As the result of a manual STOP command (see [6]), the FS resource is in the operating state "FS resource in STOP state without error". Instead of one of the start commands (see [7]), one of the reset commands is executed (see [8]). However, in this case, the FS resource cannot achieve a RUN state via any of the reset commands; instead it switches back to "FS resource in STOP state without error".
- ▶ The FS resource is in the operating state "FS resource in RUN state with error: task in STOP state". After the error has been recovered, one of the reset commands is executed instead of one of the start commands (see [5b]). However, in this case, the FS resource cannot achieve a RUN state via any of the reset commands; instead it switches to "FS resource in STOP state without error".

[2a] Switching from "Bootling system sections" to an error state

If an error is detected in a safety-related system section as the system sections are booted (see Flowchart), the FS resource switches to the operating state "Major FS error", depending on the error.

[2b] Switching from "FS resource in RUN state without error" to an error state

If an error is detected in a safety-related system section while the FS resource is in a RUN state, the FS resource switches to the operating state "Major FS error", depending on the error.

[3] Switching from "Major FS error" to "Bootling system sections"

Once the error has been recovered, the FS resource can only be started via a restart/reboot command. As a result of the cold start/reboot command, the PSS 4000-exclusive device switches to "Bootling base system", performs the start-up tests and then runs through the complete boot process for the state "Bootling system sections" (see Flowchart).

[4a] Switching from "FS resource in RUN state without error" to "FS resource in RUN state with error: task in TERMINATED state"

If an execution time error, for example, is found in an FS resource task, the task is aborted and is no longer executed. All other FS resource tasks continue running.

[4b] Switching from "FS resource in RUN state without error" to "FS resource in RUN state with error: task in STOP state"

If a runtime error, for example, is found in an FS resource task, the task is stopped and is no longer executed. All other FS resource tasks continue running.

[5a] Switching from "FS resource in RUN state with error: task in TERMINATED state" to "FS resource in RUN state without error"

Once the error has been recovered, the FS resource can be restarted via one of the reset commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[5b] Switching from "FS resource in RUN state with error: task in STOP state" to "FS resource in RUN state without error"

Once the error has been recovered, the affected task can be restarted via one of the (re)start commands:

- ▶ FS hot start
- ▶ FS warm restart
- ▶ FS warm start

[6] Switching from "FS resource in RUN state without error" to "FS resource in STOP state without error"

If the FS resource is in RUN state without error, the user can explicitly cause the FS resource to switch to a STOP state. The STOP state is maintained until the user explicitly restarts the FS resource.

It may be useful to trigger this change in operating state when commissioning plant sections, for example, or when making adjustments.

The STOP state of the FS resource can be triggered by one of the following options:

- ▶ Stopping the project (PAS4000)
- ▶ Stopping the device (PAS4000)
- ▶ Stopping the FS resource (PAS4000)

[7] Switching from "FS resource in STOP state without error" to "FS resource in RUN state without error"

The change in operating state can be brought about with one of the following commands:

- ▶ FS hot start (PAS4000)
- ▶ FS cold restart (PAS4000)
- ▶ FS warm restart (PAS4000)

[8] Reset in state "FS resource in STOP state without error"

If one of the reset commands is executed in this operating state (see [8]), then the FS resource switches back to "FS resource in STOP state without error" (see [1b]) as the system sections are booted. The change in operating state to "FS resource in RUN state without error" cannot be achieved with any of the reset commands.

[9] Reset in state "FS resource in RUN state without error"

If it is necessary to set variables to defined values, during commissioning for example, then one of the restart commands can be used (see [Effects on local variables and resource global variables](#) [ 542]).

This change in operating state can be triggered using one of the following commands:

- ▶ FS warm restart (PAS4000)
- ▶ FS cold restart (PAS4000)

[10] Switching from "FS resource in RUN state without error" to "Booting system sections"

This change in operating state affects all the device's system sections (see Flowchart).

The change in operating state can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

12.4 Operating states and change in operating status of the ST resource

The description below deals exclusively with the operating states of the ST resource of a PSS 4000-exclusive device, referring to the operating states of the device where necessary.

The diagram "Operating states and changes in operating state of the ST resource" (see [Flowchart \[443\]](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The section marked with the Roman numeral I refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the effects impact all system sections of the PSS 4000-exclusive device and not just the ST resource.
- ▶ The section marked with the Roman numeral IV refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the ST resource is started when booting the system sections of a PSS 4000-exclusive device.

12.4.1 Flow chart

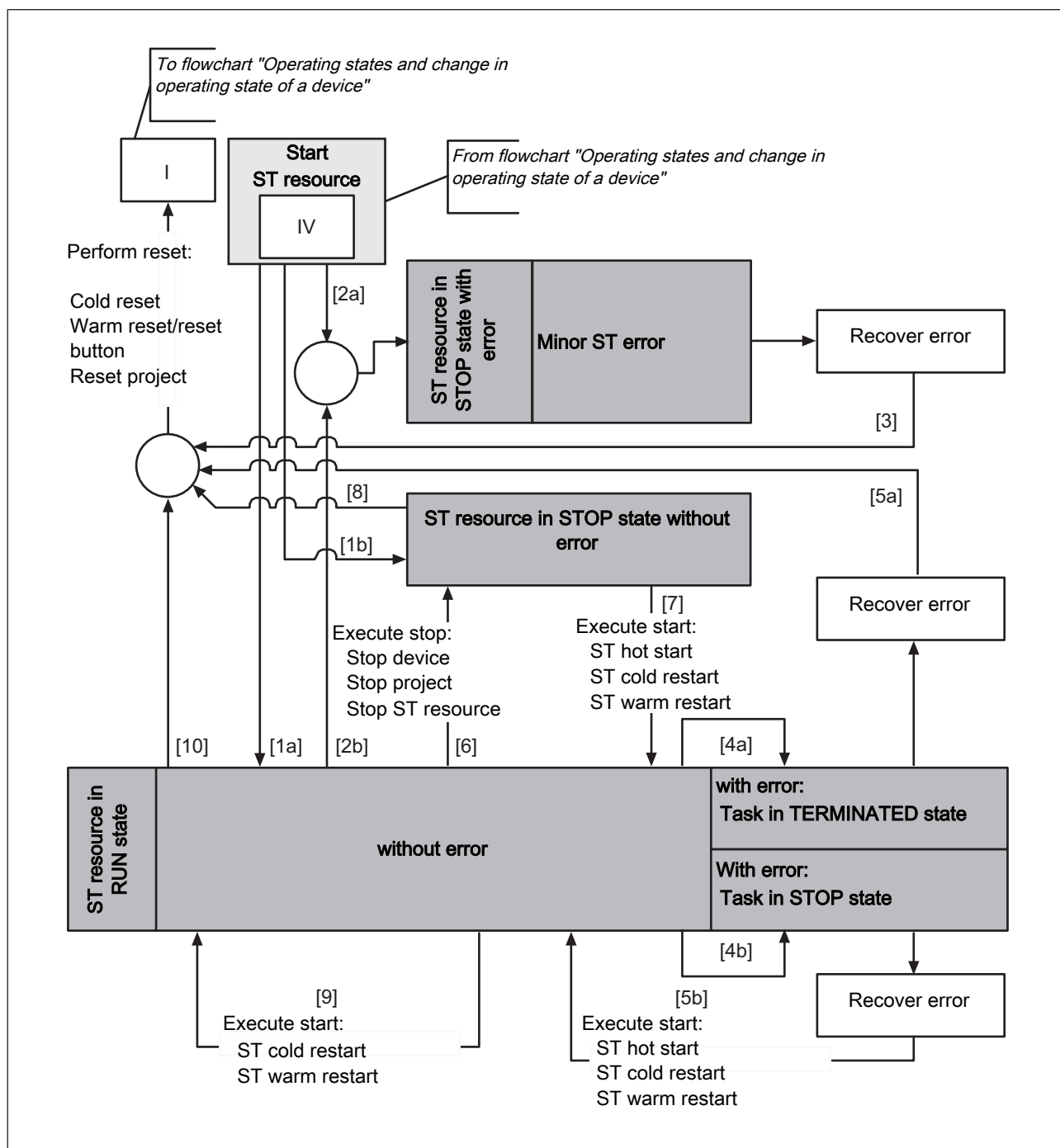


Fig.: Operating states and changes in operating state of the ST resource

12.4.2 Operating states of the ST resource

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The ST resource may assume the following operating states (see [Flowchart \[📖 443\]](#)):

- ▶ Operating states "ST resource in RUN state":
 - ST resource in RUN state without error
 - ST resource in RUN state with error: Task in TERMINATED state
 - ST resource in RUN state with error: Task in STOP state
- ▶ Operating state "ST resource in STOP state without error"
- ▶ Operating state "ST resource in STOP state with error":
 - Minor ST error



INFORMATION

The description of each operating state includes a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[📖 540\]](#)).

12.4.2.1 **ST resource in RUN condition without error**

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Restarting the device or the reboot command for the device via PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)
- ▶ After a start or restart via PAS4000
(ST cold restart, ST warm restart, ST hot start)

Effects

- ▶ In the operating state "ST resource in RUN state without error", the ST resource tasks are running without error.

12.4.2.2 ST resource in STOP condition without error

This operating state is achieved by triggering a stop via PAS4000 (device is stopped, ST resource is stopped, project is stopped).

Effects

- ▶ The behaviour of the hardware inputs, hardware outputs, process data and system sections while a resource is in a STOP state is described separately (see [Stop commands – Behaviour during the STOP state](#)  552]).
- ▶ The effects on local and resource global variables are described separately (see [Effects on local and resource global variables](#)  542]).

Remedy

- ▶ (Re)start the ST resource:
 - ST hot start (PAS4000)
 - ST cold restart (PAS4000)
 - ST warm restart (PAS4000)
- ▶ Reset device:
 - Warm reset (PAS4000) or reset button (device)
 - Cold reset (PAS4000)
- ▶ Project is reset
 - Warm reset of project (PAS4000)
 - Cold reset of project (PAS4000)

12.4.2.3 ST resource in RUN condition with error: Task in TERMINATED condition

If an error leads to this operating state, the affected task is ended as defined. A defined state is achieved.

Possible causes

(see also IEC 61131-3, Table E1)

- ▶ Execution time error
Task was not fully completed within the task cycle time

Effects

- ▶ The affected ST resource task is no longer executed, all other ST resource tasks continue to be executed.
- ▶ The ST resource continues to be in a RUN state.
- ▶ The following applies for all O-PI variables that are declared as local variables or are referenced as VAR_EXTERNAL in the POU of the task that has been stopped:
Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and valid bits = FALSE.
- ▶ The following applies for all O-PI variables that are created in Multi blocks as O-PI points or are referenced as VAR_EXTERNAL:
Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and valid bits = FALSE.
- ▶ For resource global variables (not O-PI variables) that are referenced via VAR_EXTERNAL in the blocks of the task that was stopped, the following applies:
They retain the last valid, stored values and are not set to their initial values.
- ▶ If resource global variables are referenced via VAR_EXTERNAL in the POU of a task that has not been stopped, the value there can be overwritten. As a result of this overwrite, resource global O-PI variables will again be marked as valid.
- ▶ See PAS4000 online help for details of how to handle variables (volatile variables, non-volatile variables)

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or diagnostic log
2. Recover error
3. Perform reset:
Use warm reset or cold reset

12.4.2.4 ST resource in RUN condition with error: Task in STOP condition

If an error leads to this operating state, the affected task is ended as defined. A defined state is achieved.

Possible causes

(see also IEC 61131-3, Table E1)

▶ **Runtime error**

e.g. a variable value exceeds the defined range, the numeric result of a function exceeds the value range for a data type, division by 0

Effects

- ▶ The affected ST resource task is no longer executed, all other ST resource tasks continue to be executed.
- ▶ The ST resource continues to be in a RUN state.
- ▶ The following applies for all O-PI variables that are declared as local variables or are referenced as VAR_EXTERNAL in the POU's of the task that has been stopped:
Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and valid bits = FALSE.
- ▶ The following applies for all O-PI variables that are created in Multi blocks as O-PI points or are referenced as VAR_EXTERNAL:
Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and valid bits = FALSE.
- ▶ For resource global variables (not O-PI variables) that are referenced via VAR_EXTERNAL in the blocks of the task that was stopped, the following applies:
They retain the last valid, stored values and are not set to their initial values.
- ▶ If resource global variables are referenced via VAR_EXTERNAL in the POU's of a task that has not been stopped, the value there can be overwritten. As a result of this overwrite, resource global O-PI variables will again be marked as valid.
- ▶ See PAS4000 online help for details of how to handle variables (volatile variables, non-volatile variables)

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or diagnostic log
2. Recover error
3. Execute (re)start
Use ST hot start, ST warm restart or ST cold restart
One of the reset commands can also be used. A reset command impacts all the device's system sections and not just the ST resource.

12.4.2.5

Minor ST error

If an error results in this operating state, the affected sections of the ST resource are ended as defined. A defined state is achieved.

Possible causes

- ▶ Error in the memory of the non-volatile variables

Effects

- ▶ The affected ST resource task is no longer executed, all other ST resource tasks are stopped.
- ▶ The ST resource is in a STOP state.
- ▶ For process data whose values are determined by this task, the substitute values are used; with PI variables, the corresponding valid bit also = FALSE.
- ▶ See PAS4000 Online Help for details of how to handle variables (volatile variables, non-volatile variables)

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Perform reset

12.4.3 Change in operating status of the ST resource

This description of the change in operating state assumes that the device contains a valid device project. The figures refer to the change in operating state in the diagram "Operating states and changes in operating state of the ST resource" (see [Flowchart \[443\]](#)).



INFORMATION

The descriptions of each change in operating state include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[540\]](#)).

[1a] Switching from "Bootling system sections" to "ST resource in RUN state without error"

"Bootling system sections" is one of the operating states of a PSS 4000-exclusive device (see [Flowchart](#)). If no error is detected in a safety-related system section as the system sections are booted, the ST resource is started, switching to the state "ST resource in RUN state without error".

[1b] Switching from "Bootling system sections" to "ST resource in STOP state without error"

"Bootling system sections" is one of the operating states of a PSS 4000-exclusive device (see [Flowchart](#)). Under various circumstances, as the system sections are booted, the ST resource may not achieve the operating state "ST resource in RUN state without error", but switches instead to "ST resource in STOP state without error":

- ▶ Due to a manually executed STOP command (see [6]), the ST resource is in the operating state "ST resource in STOP state without error". Instead of one of the start commands (see [7]), one of the reset commands is executed (see [8]). However, in this case, the ST resource cannot achieve a RUN state via any of the reset commands; instead it switches back to "ST resource in STOP state without error".
- ▶ The ST resource is in the operating state "ST resource in RUN state with error: Task in a STOP state". After the error has been recovered, one of the reset commands is executed instead of one of the start commands (see [5b]). However, in this case, the ST resource cannot achieve a RUN state via any of the reset commands; instead it switches to "ST resource in STOP state without error".

[2a] Switching from "Bootling system sections" to error state "Minor ST error"

If an error is detected as the system sections are booted (see [Flowchart](#)), the ST resource switches to the operating state "Minor ST error".

[2b] Switching from "ST resource in RUN state without error" to error state "Minor ST error"

If an error is detected while the ST resource is in a RUN state, the ST resource switches to the operating state "Minor ST error".

[3] Switching from "Minor ST error" to "Bootling system sections"

Once the error has been recovered, the ST resource can be restarted via one of the reset commands, provided an FS error is not present at the same time:

- ▶ Warm reset (PAS4000) or reset button (PSS 4000-exclusive device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)

- ▶ Cold reset of project (PAS4000)

[4a] Switching from "ST resource in RUN state without error" to "ST resource in RUN state with error: Task in TERMINATED state"

If an error is detected in an ST resource task, e.g. an execution time error is found, the task is aborted and is no longer executed. All other ST resource tasks continue running.

[4b] Switching from "ST resource in RUN state without error" to "ST resource in RUN state with error: Task in STOP state"

If an error is detected in an ST resource task, e.g. a runtime error is found, the task is stopped and is no longer executed. All other ST resource tasks continue running.

[5a] Switching from "ST resource in RUN state with error: Task in TERMINATED state after "ST resource in RUN state without error"

Once the error has been recovered, the ST resource can be restarted via one of the reset commands:

- ▶ Warm reset (PAS4000) or reset button (PSS 4000-exclusive device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[5b] Switching from "ST resource in RUN state with error: Task in STOP state" after "ST resource in RUN state without error"

Once the error has been recovered, the affected task can be restarted via one of the (re)start commands:

- ▶ ST hot start
- ▶ ST warm restart
- ▶ ST warm start

One of the reset commands can also be used. A reset command impacts the whole PSS 4000-exclusive device, not just the ST resource.

[6] Switching from "ST resource in RUN state without error" to "ST resource in STOP state without error"

If the ST resource is in a RUN state without error, the user can explicitly cause the ST resource to switch to a STOP state. The STOP state is maintained until the user explicitly re-starts the ST resource.

It may be useful to trigger this change in operating state when commissioning plant sections, for example, or when making adjustments.

The STOP state of the ST resource can be triggered by one of the following commands:

- ▶ Stopping the project (PAS4000)
- ▶ Stopping the device (PAS4000)
- ▶ Stopping the ST resource (PAS4000)

[7] Switching from "ST resource in STOP state without error" to "ST resource in RUN state without error"

This change in operating state can be brought about using one of the following commands:

- ▶ ST hot start (PAS4000)
- ▶ ST cold restart (PAS4000)
- ▶ ST warm restart (PAS4000)

[8] Reset in state "ST resource in STOP state without error"

If one of the reset commands is executed in this operating state (see [8]), as the system sections are booted, the ST resource switches back to "ST resource in STOP state without error" (see [1b]). A change in operating state to "ST resource in RUN state without error" cannot be achieved using any of the reset commands.

[9] Restart in state "ST resource in RUN state without error"

If it is necessary to set variables to defined values, during commissioning for example, then one of the restart commands can be used (see [Effects on local variables and resource global variables](#) [ 542]).

This change in operating state can be triggered using one of the following commands:

- ▶ ST warm restart (PAS4000)
- ▶ ST cold restart (PAS4000)

[10] Switching from "ST resource in RUN state without error" to "Bootling system sections"

This change in operating state affects all system sections of the PSS 4000-exclusive device (see Flowchart).

This change in operating state can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSS 4000-exclusive device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

12.5 Operating states and changes in operating state of the FS module bus PSSu

The description below deals exclusively with the operating states of the FS module bus PSSu of a PSS 4000-exclusive device, referring to the operating states of the device where necessary.

The diagram "Operating states and changes in operating state of the FS module bus PSSu" (see [Flowchart !\[\]\(4729e517bc6a7cd81c8025b9646574fb_img.jpg\) 454](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the effects impact all system sections of the PSS 4000-exclusive device and not just the FS module bus PSSu.
- ▶ The section marked with the Roman numeral Va refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the FS module bus PSSu is started when booting the system sections of a PSS 4000-exclusive device.

12.5.1 Flow chart

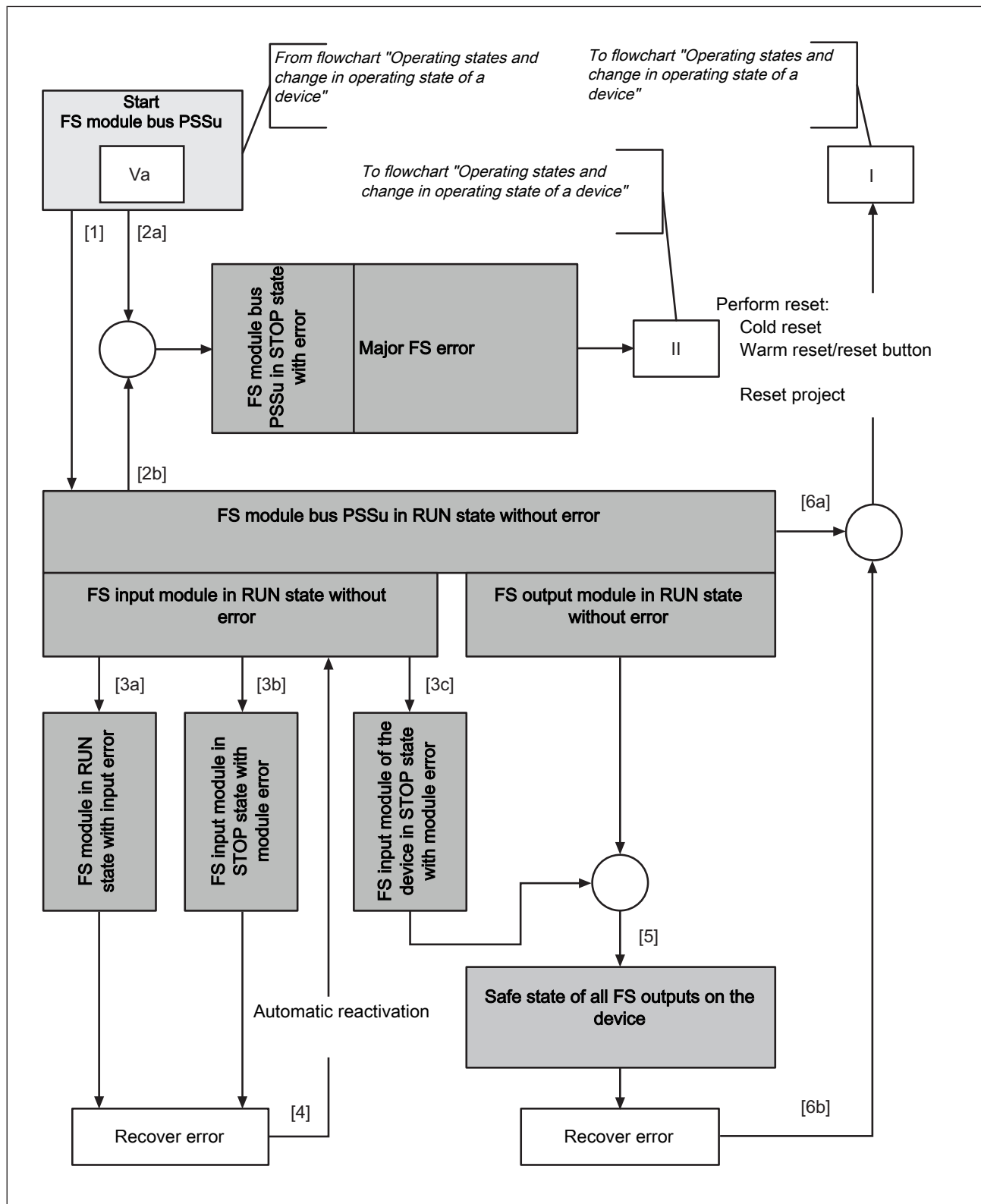


Fig.: Operating states and changes in operating state of the FS module bus PSSu

12.5.2 Operating states of the FS module bus PSSu

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The FS module bus PSSu can adopt the following operating states (see [Flowchart \[📖 454\]](#)):

- ▶ Operating states "FS module bus PSSu in RUN state without error" with
 - "FS input module in RUN state without error"
 - "FS input module in RUN state with input error"
 - "FS input module in STOP state with module error"
 - "FS input modules of the device in STOP state with module error"
 - "FS output module in RUN state without error"
 - "Safe state of all FS outputs on the device"
- ▶ Operating state "FS module bus PSSu in STOP state with error":
 - "Major FS error"



INFORMATION

The descriptions of various operating states include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[📖 540\]](#)).

12.5.2.1 FS module bus PSSu in RUN state without error

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Restarting the device or the reboot command for the device via PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ The FS module bus is in RUN state without error.
- ▶ The FS modules are operating without error, i.e.
 - the input signals/input values present at the inputs are recorded correctly and transmitted without error via the FS module bus PSSu
 - the outputs are activated correctly.

12.5.2.2 FS input module in RUN state without error

The operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status. The FS module bus PSSu is where the FS input module provides the input data for data transport to the head module.

12.5.2.3 FS module in RUN condition with input error

An FS input error only impacts the affected FS hardware input. The FS module bus remains in the operating state "FS module bus PSSu in RUN state without error".

Possible causes

- ▶ Test pulse error
- ▶ Input error

Effects

- ▶ The substitute value is used for the input signal/input value, with Valid Bit = FALSE.
- ▶ The FS input module remains in RUN state and the unaffected FS hardware inputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the FS hardware input is reactivated automatically.

12.5.2.4 FS input module in STOP state with module error

A module error only impacts the affected FS input module. The FS module bus remains in the operating state "FS module bus PSSu in RUN state without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation

Effects

- ▶ The substitute values are used for all the module's input signals/input values, with valid bits = FALSE.
- ▶ The unaffected FS input modules on the device continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the FS input module is reactivated automatically.

12.5.2.5 FS input modules of the device in STOP state with module error

The FS input error affects all FS input modules and FS outputs of the PSS 4000-exclusive device. The FS module bus remains in the operating state "FS module bus PSSu in RUN state without error".

Possible causes

- ▶ The device detects an FS input error where the error location cannot be uniquely identified (e. g. PSSu E F AI U).

Effects

- ▶ The substitute values are used for all the device's input signals/input values, with valid bits = FALSE
- ▶ All FS hardware outputs on the device switch to a safe state.
- ▶ The substitute values are used for the module's FS outputs, with valid bits = FALSE.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Perform reset
(warm reset/reset button, cold reset, project reset)

12.5.2.6 FS output module in RUN state without error

The operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status. The FS output module processes the output data provided by the FS module bus PSSu.

12.5.2.7 Safe state of all FS outputs on the device

If an error leads to this operating state, the effect of the error is limited to the FS hardware outputs of the PSS 4000-exclusive device on which it was detected. The FS module bus remains in the operating state "FS module bus PSSu in RUN state without error".

Possible causes

- ▶ Output error
e.g. shorts across contacts, short circuit, error in the output test
- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation
- ▶ Operating state "FS input modules of the device in STOP state with module error"

Effects

- ▶ All FS hardware outputs on the device switch to a safe state.
- ▶ The substitute values are used for the FS outputs of the modules.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Perform reset
(warm reset/reset button, cold reset, project reset)

12.5.2.8 Major FS error

If an error leads to this operating state, the affected sections of the FS module bus PSSu are ended as defined. A defined state is achieved.

The effects of a major FS error impact all safety-related parts of a PSS 4000-exclusive device (see [Operating state "Major FS error" \[📖 428\]](#)).

12.5.3 Change in operating state of the FS module bus PSSu

This description of the change in operating state assumes that the device contains a valid device project. The figures refer to the change in operating state in the diagram "Operating states and changes in operating state of the FS module bus PSSu" (see

[Flowchart \[📖 454\]](#)).



INFORMATION

The descriptions of each change in operating state include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[📖 540\]](#)).

[1] Switching from "Bootling system sections" to "FS module bus PSSu in RUN state without error"

"Bootling system sections" is one of the operating states of a PSS 4000-exclusive device (see Flowchart). If no error is detected in a safety-related system section as the system sections are booted, the FS module bus PSSu is started, switching to the state "FS module bus PSSu in RUN state without error".

[2a] Switching from "Bootling system sections" to "Major FS error"

If an error is detected in a safety-related system section as the system sections are booted, the FS module bus PSSu switches to the operating state "Major FS error".

The state "Bootling system sections" and the state "Major FS error" (see Flowchart) are both operating states that affect the whole PSS 4000-exclusive device.

[2b] Switching from "FS module bus PSSu in RUN state without error" to "Major FS error"

If an error is detected in a safety-related system section while the FS module bus PSSu is in a RUN state, the FS module bus PSSu switches to the operating state "Major FS error".

The operating state "Major FS error" (see Flowchart) is an operating state that affects the whole PSS 4000-exclusive device.

[3a] Switching from "FS input module in RUN state without error" to "FS module in RUN state with input error"

If an input error is detected on an FS hardware input, the FS input module switches to the operating state "FS module in RUN state with input error".

[3b] Switching from "FS input module in RUN state without error" to "FS input module in STOP state with module error"

If an error is detected on an FS input module, the relevant FS input module on the device switches to the operating state "FS input module in STOP state with module error".

[3c] Switching from "FS input module in RUN state without error" to "FS input modules on the device in STOP state with module error"

If an FS input error is detected and the PSSu-exclusive device cannot assign it uniquely to one of its FS hardware inputs, all the FS input modules on the device switch to the operating state "FS input module in STOP state with module error" and all the FS outputs on the device also switch to the operating state "Safe state of all FS outputs on the device" (see [5]).

[4] Switching from one of the operating states "FS input module in RUN state with input error" or

"FS input module in STOP state with module error" to "FS input module in RUN state without error"

If the relevant error (input error, module error) is recovered, the affected section on the FS input module is reactivated automatically and the module switches to the operating state "FS input module in RUN state without error"

[5] Switching from "FS output module in RUN state" to "Safe state of all FS outputs on the device"

If an error is detected on an FS output module or FS output, or if the FS input modules on the PSS 4000-exclusive device switch to the operating state "FS input modules on the device in STOP state with module error", all FS outputs on the device switch to a safe state.

[6a] Switching from "FS module bus PSSu in RUN state without error" to "Bootling system sections"

This change in operating state affects all system sections of the PSS 4000-exclusive device (see Flowchart).

This change in operating state can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[6b] Switching from "Safe state of all FS outputs on the device" to "Bootling system sections"

This change in operating state affects all system sections of the PSS 4000-exclusive device (see Flowchart).

Once the fault has been recovered, the change in operating state can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

12.6 Operating states and change in operating state of the FS module bus PSS u2

The description below deals exclusively with the operating states of the FS module bus PSS u2 and, where necessary, refers to the operating states of the device.

The diagram "Operating states and change in operating state of the FS module bus PSS u2" (see Flowchart - Operating states and change in operating state of the FS module bus PSS u2) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and change in operating state of a device" (see Flow chart), as the effects impact all of the device's system sections and not just the FS module bus PSS u2.
- ▶ The section marked with the Roman numeral III refers to the diagram "Operating states and change in operating state of a device" (see Flowchart), as the FS module bus PSS u2 is started when booting the system sections of a 2nd generation device.

12.6.1 Flowchart - Operating states and change in operating state of the FS module bus PSS u2

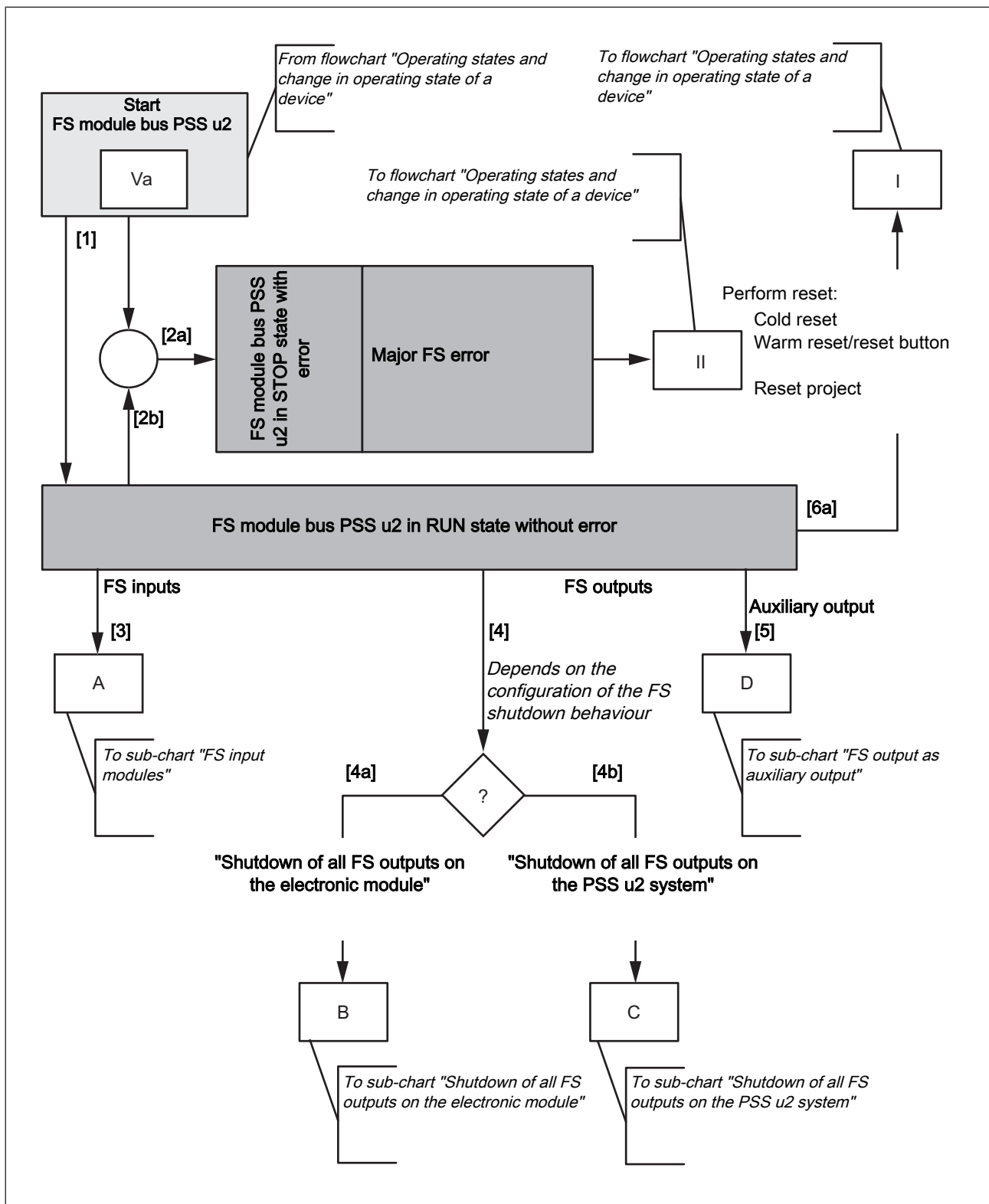


Fig.: Operating states and change in operating state of the FS module bus PSS u2

For the sake of clarity, the flowchart is divided into other sub-charts. You'll find the sub-charts under

- ▶ Sub-chart - FS input modules  469]
- ▶ Sub-chart - Shutdown of all FS outputs on the electronic module  470]
- ▶ Sub-chart - Shutdown of all FS outputs on the PSS u2 system  471]
- ▶ Sub-chart - FS output as auxiliary output  472]

12.6.2 Sub-chart - FS input modules

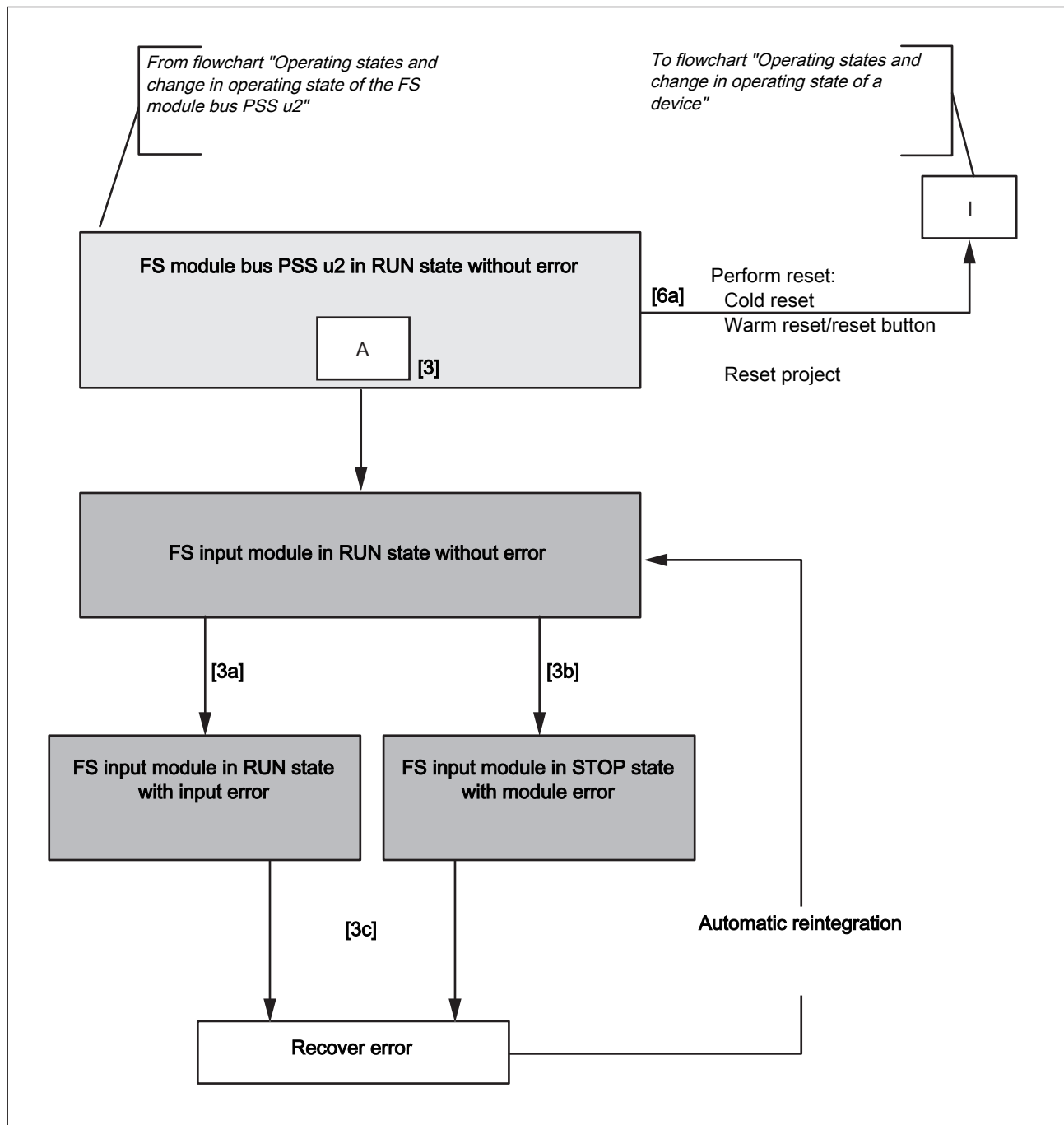


Fig.: Operating states and change in operating state of the FS module bus PSS u2: FS inputs

12.6.3 Sub-chart - Shutdown of all FS outputs on the electronic module

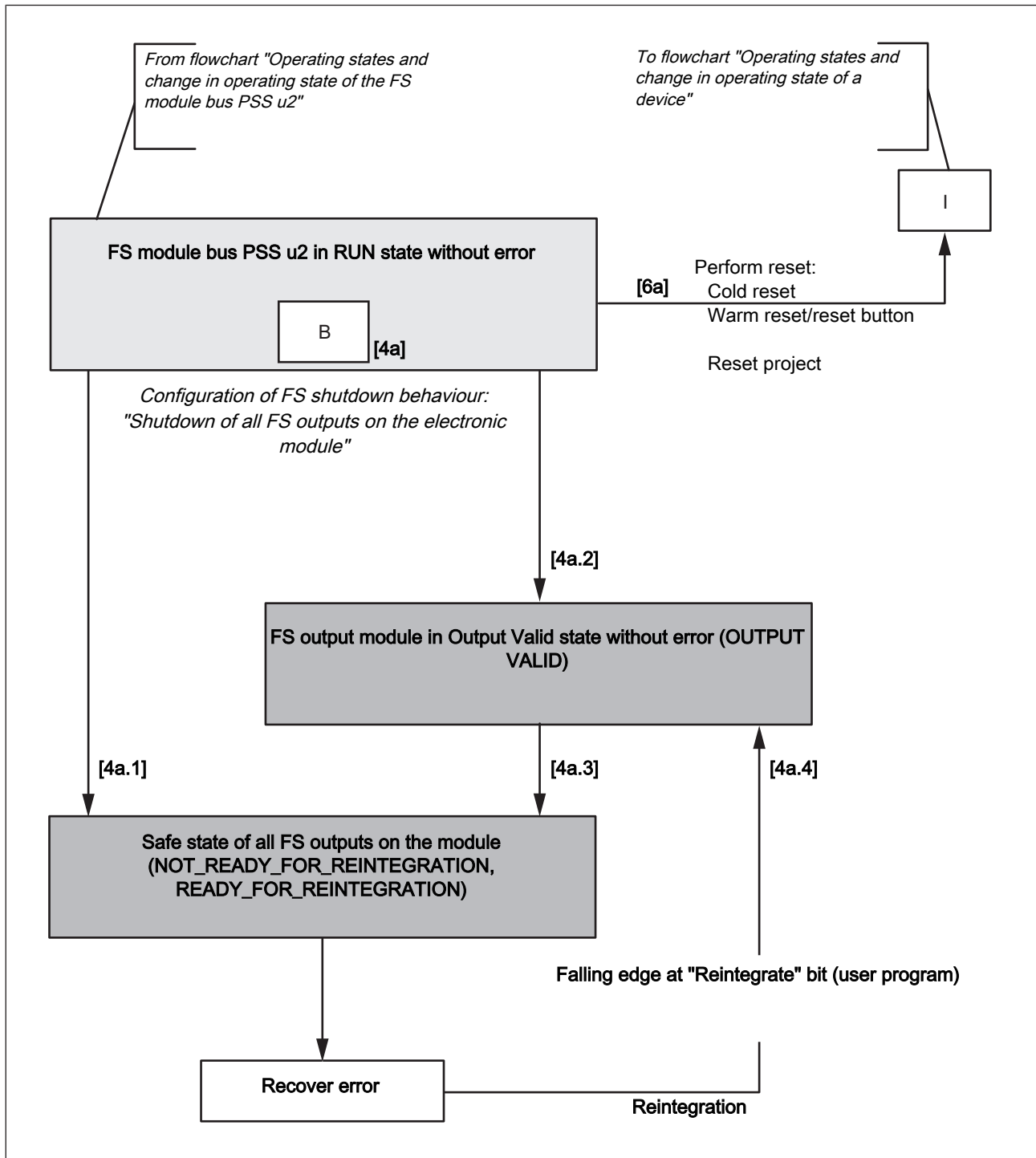


Fig.: Operating states and change in operating state of the FS module bus PSS u2: Shutdown of all FS outputs on the electronic module

12.6.4 Sub-chart - Shutdown of all FS outputs on the PSS u2 system

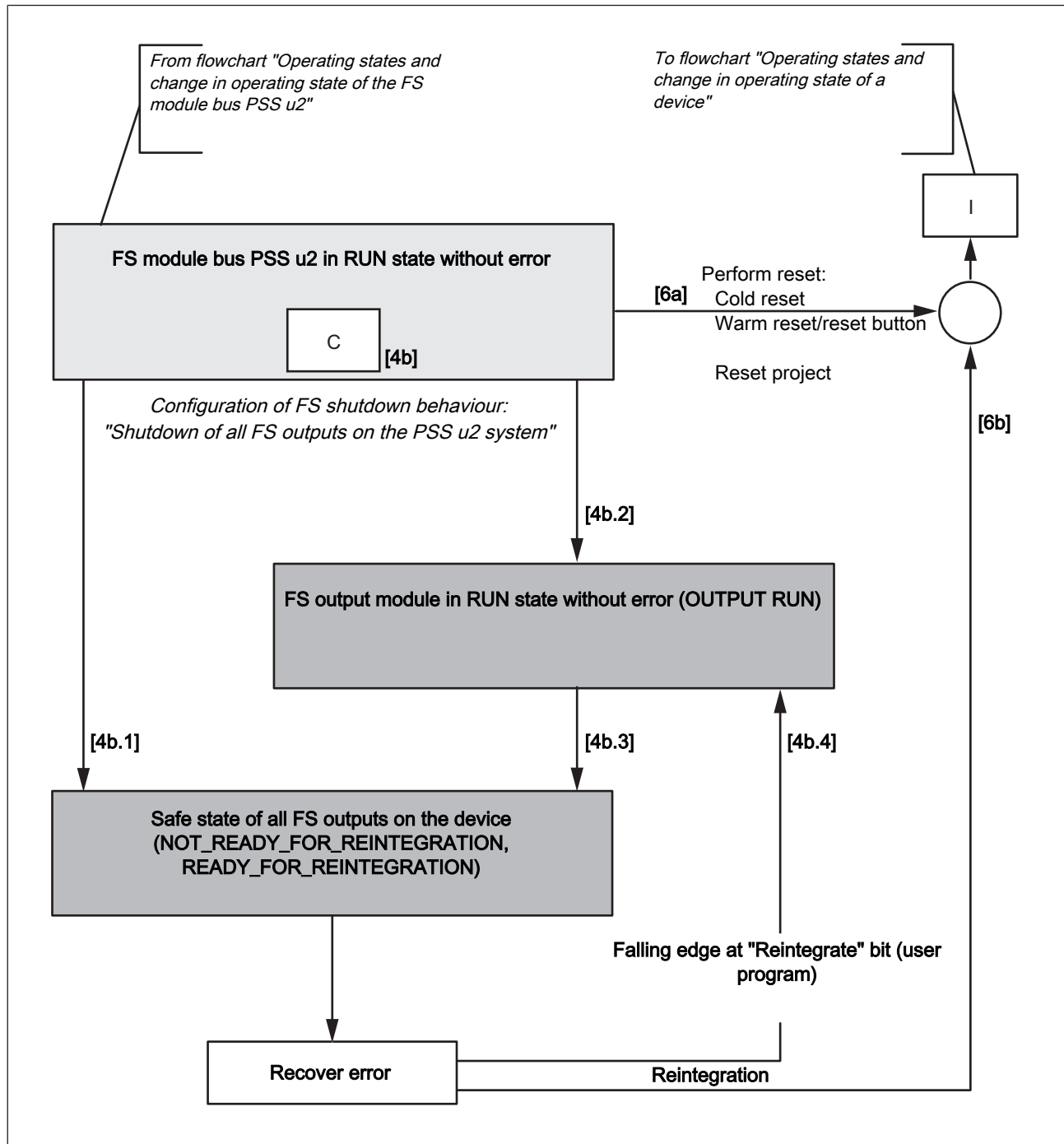


Fig.: Operating states and change in operating state of the FS module bus PSS u2: Shutdown of all FS outputs on the PSS u2 system

12.6.5 Sub-chart - FS output as auxiliary output

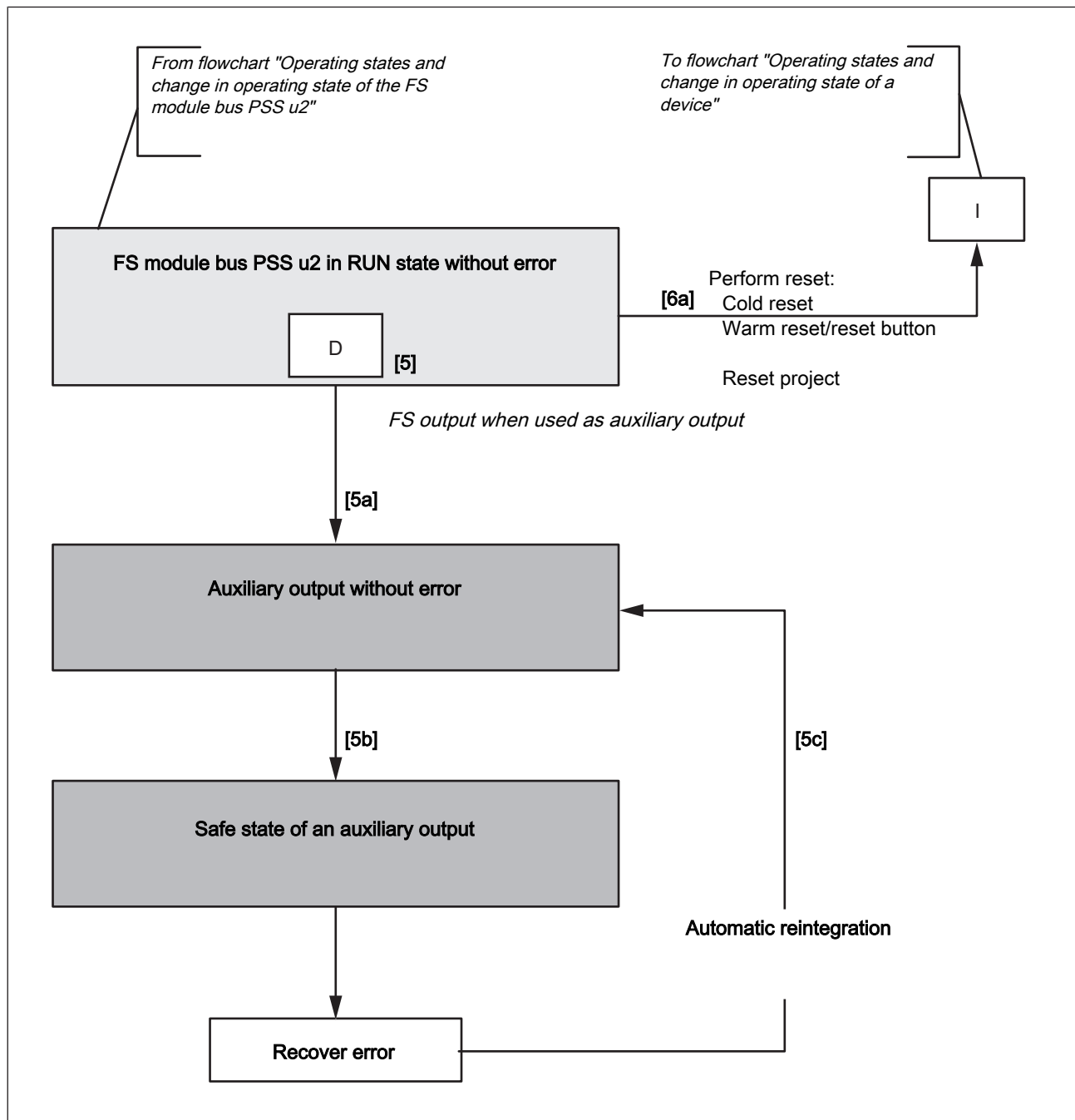


Fig.: Operating states and change in operating state of the FS module bus PSS u2: FS outputs as auxiliary outputs

12.6.6 Operating states of the FS module bus PSS u2

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The FS module bus PSS u2 may adopt the following operating states:

- ▶ Operating state "FS module bus PSS u2 in RUN state without error" with the following operating states (see Flowchart - Operating states and change in operating state of the FS module bus PSS u2):
 - FS inputs (see Sub-chart - FS input modules):
 - "FS input module in RUN state without error"
 - "FS input module in RUN state with input error"
 - "FS input module in STOP state with module error"
 - FS outputs with configuration of the shutdown behaviour "Shutdown of all FS outputs on the electronic module" and deactivation of the option "Automatic reintegration at startup" (see Sub-chart - Shutdown of all FS outputs on the electronic module):
 - "FS output module in Output Valid state without error (OUTPUT VALID)"
 - "Safe state of all FS outputs on the module"
 - FS outputs with configuration of the shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system" (see Sub-chart - Shutdown of all FS outputs on the PSS u2 system):
 - "FS output module in RUN state without error (OUTPUT RUN)"
 - "Safe state of all FS outputs on the device"
 - FS outputs when used as auxiliary outputs (see Sub-chart - FS output as auxiliary output):
 - "FS auxiliary output without error"
 - "FS auxiliary output with error"
- ▶ Operating state "FS module bus PSS u2 in STOP state with error" (see Flowchart - Operating states and change in operating state of the FS module bus PSS u2):
 - "Major FS error"



INFORMATION

The descriptions of various operating states include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options](#) [ 540]).

12.6.6.1 FS module bus PSS u2 in RUN state without error

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Restarting the device or the reboot command for the device via PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ The FS module bus PSS u2 is in RUN state without error.
- ▶ The FS modules are operating without error, i.e.
 - The input signals/input values present at the inputs are recorded correctly and transmitted without error via the FS module bus PSS u2
 - The outputs are activated correctly.

12.6.6.2 FS input module in RUN state without error

The operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status. The FS module bus PSS u2 is where the FS input module provides the input data for data transport to the head module.

12.6.6.3 FS input module in RUN state with input error

An FS input error only impacts the affected FS hardware input. The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Test pulse error
- ▶ Input error

Effects

- ▶ The substitute value is used for the input signal/input value, with valid bit = FALSE.
- ▶ The FS input module remains in RUN state and the unaffected FS hardware inputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the FS hardware input is reintegrated automatically.

12.6.6.4 FS input module in STOP state with module error

A module error only impacts the affected FS input module. The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation
- ▶ Configuration error (set/actual hardware registry does not match; i.e. wrong module or no module plugged in)

Effects

- ▶ The substitute values are used for all the module's input signals/input values, with valid bits = FALSE.
- ▶ The unaffected FS input modules on the device continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the FS input module is reintegrated automatically.

12.6.6.5 FS output module in RUN state without error

The operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status. The FS output module processes the output data provided by the FS module bus PSS u2.

12.6.6.6 Safe state of all FS outputs on the module

In a PSS 4000 project, the error behaviour of FS outputs can be configured for each individual PSS u2 system.

If the FS shutdown behaviour "Shutdown of all FS outputs on the electronic module" is configured for a PSS u2 system, then in the event of an error on an output used for an FS application, only the FS outputs belonging to the same FS module output group are shut down. The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Output error
e.g. shorts across contacts, short circuit, error in the output test
- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation
- ▶ Configuration error (set/actual hardware registry does not match; i.e. wrong module or no module plugged in)

Effects

- ▶ All FS hardware outputs on the electronic module switch to a safe state.
- ▶ The substitute values are used for the module's FS outputs.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Perform one of the following actions
 - Restart the PSS u2 system
It is a precondition for this way of reintegration that in the configuration of the PSS u2 system the "Automatic reintegration at startup" option is set to "Yes".
 - Perform reset
(Warm reset/reset pushbutton, cold reset of the project)
It is a precondition for this way of reintegration that in the configuration of the PSS u2 system the "Automatic reintegration at startup" option is set to "Yes".
 - Falling edge at "Reintegrate" bit

Further information is available under [FS shutdown behaviour "Shutdown of all FS outputs on the electronic module"](#) [ 125].

12.6.6.7 Safe state of all FS outputs on the device

In a PSS 4000 project, the error behaviour of FS outputs can be configured for each individual PSS u2 system.

If the FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system" is configured for a PSS u2 system, then in the event of an error on an output used for an FS application, all FS outputs on the PSS u2 system belonging to the PSS u2 system's FS system output group are shut down. The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Output error
e.g. shorts across contacts, short circuit, error in the output test
- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation
- ▶ Configuration error (set/actual hardware registry does not match; i.e. wrong module or no module plugged in)

Effects

- ▶ All FS hardware outputs on the electronic module switch to a safe state.
- ▶ The substitute values are used for the module's FS outputs.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Perform one of the following actions
 - Restart the PSS u2 system
 - Perform reset
(warm reset/reset button, cold reset, project reset)
 - Falling edge at "Reintegrate" bit

Further information is available under [FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system" \[!\[\]\(d5d7044e5caf6907399af2dced8d6ff8_img.jpg\) 122\]](#).

12.6.6.8 Auxiliary output without error

This operating state is achieved irrespective of the module bus status, as soon as the module containing the auxiliary output is ready for operation.

Note: If an FS output is used exclusively for non-safety-related tasks, so it is only assigned ST data, then it is an auxiliary output.

12.6.6.9 Safe state of an auxiliary output

An output error on an auxiliary output only impacts the affected FS hardware output. The impact is independent of the configuration of the FS shutdown behaviour. The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Output error
(e.g. overload, short between contacts, short circuit)
- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ Configuration error (set/actual hardware registry does not match; i.e. wrong module or no module plugged in)

Effects

- ▶ The auxiliary output switches to a safe state.
- ▶ The substitute value is used for the output signal.
- ▶ The FS output module remains in RUN state and the unaffected FS outputs or auxiliary outputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the auxiliary output is reintegrated automatically.

12.6.6.10 Major FS error

If an error results in this operating state, the affected sections of the FS module bus PSS u2 are ended as defined. A defined state is achieved.

The effects of a major FS error impact all safety-related parts of a PSS 4000-exclusive device (see [Operating state "Major FS error" \[428\]](#)).

12.6.7 Change in operating state of the FS module bus PSS u2

This description of the change in operating state assumes that the device contains a valid device project. The numbers refer to the change in operating state in the diagram "Operating states and change in operating state of the FS module bus PSS u2" and the corresponding subdiagrams.



INFORMATION

The descriptions of each change in operating state include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options](#)). [ 540]

[1] Switching from "Bootling system sections" to "FS module bus PSS u2 in RUN state without error"

Bootling system sections is one of the operating states of a PSS 4000-exclusive device (see [Flow chart \[!\[\]\(17acf1afa8cdf0b67c53d4865a5ed469_img.jpg\) 414\]](#) "Operating states and change in operating state of a PSS 4000-exclusive device"). If no fault is detected in a safety-related system section as the system sections are booted, the FS module bus PSS u2 is started, switching to the state "FS module bus PSS u2 in RUN state without error" (see [Flowchart - Operating states and change in operating state of the FS module bus PSS u2 \[!\[\]\(ece8cabb5adcd402275b8866019cc3b8_img.jpg\) 467\]](#)).

[2a] Switching from "Bootling system sections" to "Major FS error"

If a fault is detected in a safety-related system section as the system sections are booted, the FS module bus PSS u2 switches to the operating state "Major FS error".

The state "Bootling system sections" and the state "Major FS error" are both operating states that affect the whole PSS 4000-exclusive device (see [Flow chart \[!\[\]\(e1c624d4757f08486e89482c18364c17_img.jpg\) 414\]](#) "Operating states and change in operating state of a PSS 4000-exclusive device").

[2] Switching from "FS module bus PSS u2 in RUN state without error" to "Major FS error"

If a fault is detected in a safety-related system section while the FS module bus PSS u2 is in a RUN state, the FS module bus PSS u2 switches to the operating state "Major FS error".

The operating state "Major FS error" is an operating state that affects the whole PSS 4000-exclusive device (see [Flow chart \[!\[\]\(e9474ce1d70442456f8fe9c393ea149c_img.jpg\) 414\]](#) "Operating states and change in operating state of a PSS 4000-exclusive device").

[3] Switching to input module "FS input module in RUN state without error"

The operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status (see [Sub-chart - FS input modules \[!\[\]\(2b17f17ebbacc911bb0ff784ab641779_img.jpg\) 469\]](#)).

[3a] Switching from "FS input module in RUN state without error" to "FS input module in RUN state with input error"

If an input error is detected on an FS hardware input, the FS input module switches to the operating state "FS module in RUN state with input error". The affected input does not affect any other inputs.

The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

[3b] Switching from "FS input module in RUN state without error" to "FS input module in STOP state with module error"

If an error is detected on an FS input module, the relevant FS input module on the PSS u2 system switches to the operating state "FS input module in STOP state with module error". The affected module does not affect any other inputs or modules.

The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

[3c] Switching from one of the operating states "FS input module in RUN state with input error" or

"FS input module in STOP state with module error" to "FS input module in RUN state without error"

If the relevant error (input error, module error) has been recovered, then the affected FS inputs are reintegrated automatically and the module switches to the operating state "FS input module in RUN state without error".

[4] In a PSS 4000 project, the error behaviour of FS outputs can be configured for each individual PSS u2 system (see [Flowchart - Operating states and change in operating state of the FS module bus PSS u2](#)  467]).

Configuration options:

- ▶ FS shutdown behaviour "Shutdown of all FS outputs on the electronic module"
- ▶ FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system"

[4a] Behaviour of FS outputs with configuration of the FS shutdown behaviour "Shutdown of all FS outputs on the electronic module"

(see [Sub-chart - Shutdown of all FS outputs on the electronic module](#)  470))

▶ **[4a.1] Switch to "Safe state of all FS outputs on the module"**

The operating state is adopted irrespective of the module bus status, but dependent on the option "Automatic reintegration at startup":

- "Automatic reintegration at startup" option is deactivated:

When the option is deactivated, the operating state is also adopted when all outputs on the FS module output group are error-free and contain valid values.

- "Automatic reintegration at startup" option is activated:

When the option is activated, the operating state is only adopted if an output on the FS module output group has an error or contains invalid data.

▶ **[4a.2] Switching to "FS output module in Output Valid state without error"**

The operating state is adopted irrespective of the module bus status. The operating state is adopted when all the following conditions are met:

- The "Automatic reintegration at startup" option is activated.
- At start-up, all the outputs on the FS module output group are error-free and contain valid values.

▶ **[4a.3] Switching from "FS output module in Output Valid state without error" to "Safe state of all FS outputs on the module"**

The operating state is adopted if at least one of the following conditions is met:

- Error at an output on the FS module output group
- An output contains invalid data

Only the FS outputs belonging to the same FS module output group are shut down. The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

► **[4a.4] Switching from "Safe state of all FS outputs on the module" to "FS output module in Output Valid state without error"**

If all the outputs on the FS module output group are error-free and contain valid values, then the FS outputs on the FS module output group can be reintegrated by the user program triggering a falling edge at the *Reintegrate* bit.

Further information is available under [FS shutdown behaviour "Shutdown of all FS outputs on the electronic module"](#)  125].

[4b] Behaviour of FS outputs with configuration of the FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system"

(see [Sub-chart - Shutdown of all FS outputs on the PSS u2 system](#)  471])

► **[4b.1] Switching to "Safe state of all FS outputs on the device"**

The operating state is achieved as soon as a fault is detected when initialising a module on the FS system output group, irrespective of the module bus status.

► **[4b.2] Switching to "FS output module in RUN state without error"**

The operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status.

► **[4b.3] Switching from "FS output module in RUN state without error" to "Safe state of all FS outputs on the device"**

In the event of a fault on an output that's used for an FS application, all FS outputs on the PSS u2 system that are in the FS system output group of the PSS u2 system are shut down.

The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

► **[4b.4] Switching from "Safe state of all FS outputs on the device" to "FS output module in RUN state without error"**

If the fault has been recovered, then the FS outputs on the FS system output group can be reintegrated, either by the user program triggering a falling edge at the *Reintegrate* bit or by triggering one of the resets (see [6b]).

Further information is available under [FS shutdown behaviour "Shutdown of all FS outputs on the PSS u2 system"](#)  122].

[5] Behaviour of an FS output, if the FS output is used exclusively for non-safety-related tasks, so it is only assigned ST data (= auxiliary output).

► **[5a] Switching to "Auxiliary output without error"**

The operating state is achieved as soon as the module containing the auxiliary output is ready for operation, irrespective of the module bus status and irrespective of the configuration of the FS shutdown behaviour.

► **[5b] Switching from "Auxiliary output without error" to "Safe state of an auxiliary output"**

In the event of an output fault on an auxiliary output, the auxiliary output is switched to a safe state. The output fault only impacts the affected FS hardware output.

The FS module bus PSS u2 remains in the operating state "FS module bus PSS u2 in RUN state without error".

► **[5c] Switching from "Safe state of an auxiliary output" to "Auxiliary output without error"**

If the fault has been recovered, then the auxiliary output is reintegrated automatically.

[6a] Switching from "FS module bus PSS u2 in RUN state without error" to "Booting system sections"

This change in operating state impacts all system sections of the PSS 4000-exclusive device (see [Flow chart \[414\]](#) "Operating states and change in operating state of a PSS 4000-exclusive device").

The change in operating state can be triggered using one of the following commands:

- Warm reset (PAS4000) or reset button (device)
- Cold reset (PAS4000)
- Warm reset of project (PAS4000)
- Cold reset of project (PAS4000)

[6b] Switching from von "Safe state of all FS outputs on the device" to "Booting system sections"

This change in operating state impacts all system sections of the PSS 4000-exclusive device (see [Flow chart \[414\]](#) "Operating states and change in operating state of a PSS 4000-exclusive device").

When all faults at the FS outputs of the FS system output group have been recovered, the FS system output group can be reintegrated; i.e. the outputs are restored to the state in which they can output process data.

Reintegration can be achieved by starting the module bus PSS u2 using one of the following commands:

- Warm reset (PAS4000) or reset button (device)
- Cold reset (PAS4000)
- Warm reset of project (PAS4000)
- Cold reset of project (PAS4000)

12.7 Operating states and changes in operating state of the ST module bus PSSu

The description below deals exclusively with the operating states of the ST module bus PSSu of a PSS 4000-exclusive device, referring to the operating states of the device where necessary.

The diagram "Operating states and changes in operating state of the ST module bus PSSu" (see [Flowchart !\[\]\(dfbd6b3763a6d1d9afaa974f64e2e4b5_img.jpg\) 487](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the effects impact all system sections of the PSS 4000-exclusive device and not just the ST module bus PSSu.
- ▶ The section marked with the Roman numeral Vb refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the ST module bus PSSu is started when booting the system sections of a PSS 4000-exclusive device.

12.7.1 Flow chart

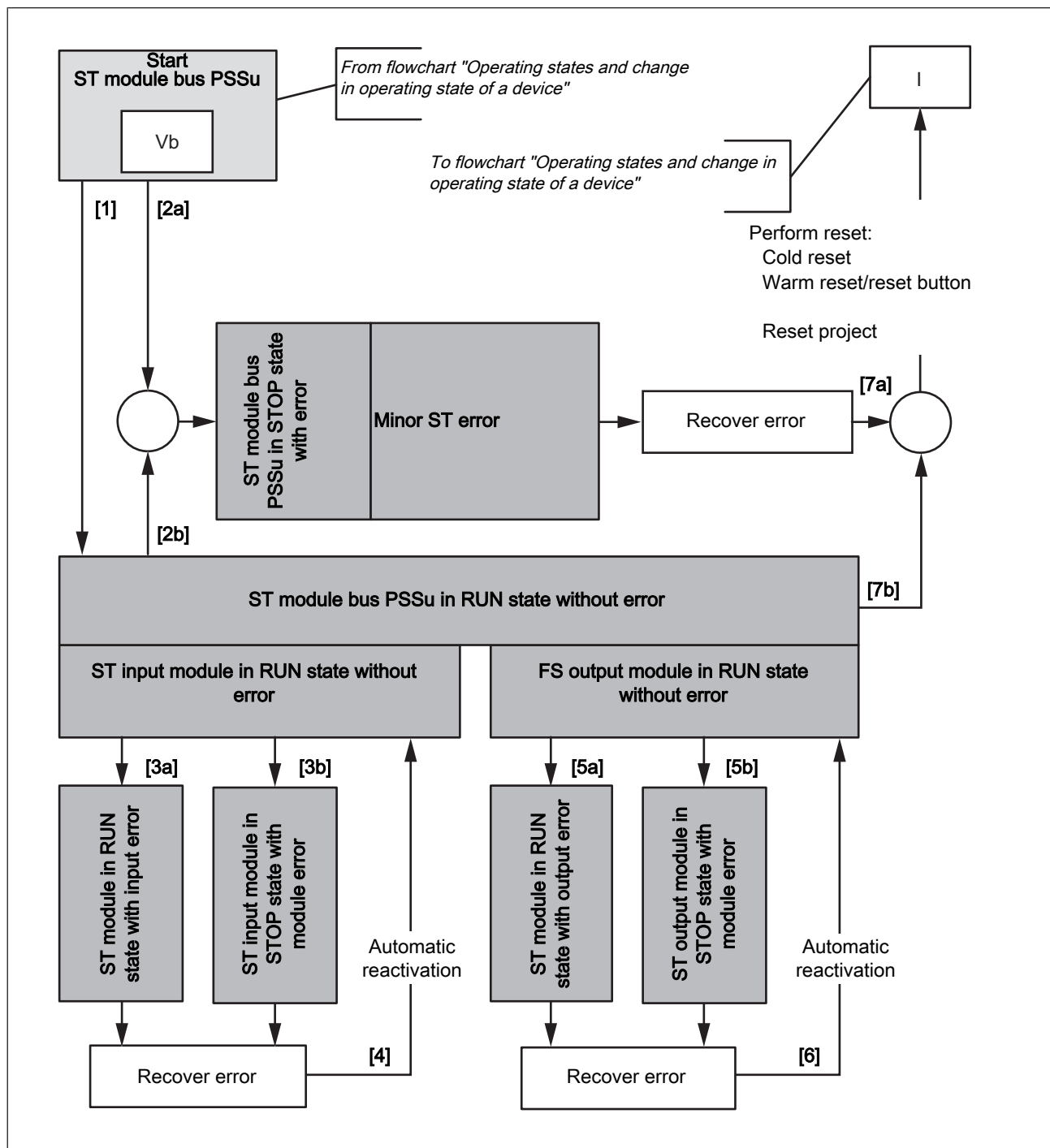


Fig.: Operating states and changes in operating state of the ST module bus

12.7.2 Operating states of the ST module bus PSSu

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The ST module bus can adopt the following operating states (see [Flowchart](#)  487):

- ▶ Operating states "ST module bus PSSu in RUN state without error" with
 - "ST input module in RUN state without error"
 - "ST module in RUN state with input error"
 - "ST input module in STOP state with module error"
 - "ST output module in RUN state without error"
 - "ST module in RUN state with output error"
 - "ST output module in STOP state with module error"
- ▶ Operating state "ST module bus PSSu in STOP state with error":
 - "Minor ST error"



INFORMATION

The descriptions of various operating states include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options](#)  540).

12.7.2.1 ST module bus PSSu in RUN state without error

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Restarting the device or the reboot command for the device via PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ The ST module bus PSSu is in RUN state without error.
- ▶ The ST modules are operating without error, i.e.
 - The input signals/input values present at the inputs are recorded correctly and transmitted without error via the ST module bus PSSu
 - The outputs are activated correctly.

12.7.2.2 ST input module in RUN state without error

The operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status. The ST module bus PSSu is where the ST input module provides the input data for data transport to the head module.

12.7.2.3 ST module in RUN condition with input error

An input error only influences the affected ST hardware input. The ST module bus remains in the operating state "ST module bus PSSu in RUN state without error".

Possible causes

- ▶ Input error

Effects

- ▶ The substitute value is used for the input signal/input value, with Valid Bit = FALSE.
- ▶ The ST input module remains in RUN state and the unaffected ST hardware inputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the ST hardware input is reactivated automatically.

12.7.2.4 ST input module in STOP state with module error

A module error only impacts the affected ST input module. The ST module bus remains in the operating state "ST module bus PSSu in RUN state without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation

Effects

- ▶ The substitute values are used for all the module's input signals/input values, with valid bits = FALSE.
- ▶ The unaffected ST input modules on the device continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the ST input module is reactivated automatically.

12.7.2.5 ST output module in RUN state without error

This operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status. The ST output module processes the output data provided by the ST module bus.

12.7.2.6 ST module in RUN condition with output error

An ST output error only impacts the affected ST hardware output. The ST module bus PSSu remains in the operating state "ST module bus PSSu in RUN state without error".

Possible causes

- ▶ Output error
(e.g. overload, short circuit)

Effects

- ▶ The substitute value is used for the output signal/output value.
- ▶ The ST output module remains in RUN state and the unaffected ST hardware outputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the ST hardware output is reactivated automatically.

12.7.2.7 ST output module in STOP state with module error

A module error in an ST output module only impacts the affected ST output module. The ST module bus PSSu remains in the operating state "ST module bus PSSu in RUN state without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation

Effects

- ▶ The substitute values are used for all the module's output signals/output values.
- ▶ The unaffected ST output modules on the device continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the ST output module is reactivated automatically.

12.7.2.8 Minor ST error

If an error leads to this operating state, the affected sections of the ST module bus PSSu are ended as defined. A defined state is achieved.

Possible causes

- ▶ Hardware registry error
(e.g. differences between the set/actual hardware registry)
- ▶ Defective ST module detected when starting the module bus PSSu

Effects

- ▶ If the error is detected when starting the ST module bus PSSu, the ST outputs are not activated and the ST inputs are not read.
- ▶ If other devices access the ST I/O data on this device, the substitute values are used for all signals/values, with valid bits = FALSE.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the device must be restarted via one of the reset commands.

12.7.3 Change in operating state of the ST module bus PSSu

This description of the change in operating state assumes that the device contains a valid device project. The numbers refer to the change in operating state in the diagram "Operating states and change in operating state of the ST module bus PSSu" (see Flow chart).



INFORMATION

The descriptions of each change in operating state include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options](#) [ 540]).

[1] Switching from "Bootling system sections" to "ST module bus PSSu in RUN state without error"

Bootling system sections is one of the operating states of a PSS 4000-exclusive device (see Flow chart of a device). If no error is detected as the system sections are booted, the ST module bus PSSu is started, switching to the state "ST module bus PSSu in RUN state without error".

[2a] Switching from "Bootling system sections" to "Minor ST error"

If an error is detected as the system sections are booted, the ST module bus PSSu switches to the operating state "Minor ST error".

The operating state "Minor ST error" is an operating state that only affects the ST module bus PSSu of a PSS 4000-exclusive device.

[2b] Switching from "ST module bus PSSu in RUN state without error" to "Minor ST error"

If an error is detected while the ST module bus PSSu is in RUN state, the ST module bus PSSu switches to the operating state "Minor ST error".

The operating state "Minor ST error" is an operating state that only affects the ST module bus PSSu of a PSS 4000-exclusive device.

[3a] Switching from "ST input module in RUN state without error" to "ST input module in RUN state with input error"

If an input error is detected on an ST hardware input, the ST input module switches to the operating state "ST module in RUN state with input error".

[3b] Switching from "ST input module in RUN state without error" to "ST input module in STOP state with module error"

If a module error is detected on an ST input module, the affected ST input module switches to the operating state "ST input module in STOP state with module error".

[4] Switching from one of the operating states "ST input module in RUN state with input error" or "ST input module in STOP state with module error" to "ST input module in RUN state without error"

If the relevant error (input error, module error) has been recovered, the affected section on the ST input module is reactivated automatically and the module switches to the operating state "ST input module in RUN state without error".

[5a] Switching from "ST output module in RUN state without error" to "ST output module in RUN state with output error"

If an output error is detected on an ST hardware output, the ST output module switches to the operating state "ST module in RUN state with output error".

[5b] Switching from "ST output module in RUN state without error" to "ST output module in STOP state with module error"

If a module error is detected on an ST output module, the affected ST output module switches to the operating state "ST output module in STOP state with module error"

[6] Switching from one of the operating states "ST output module in RUN state with output error" or "ST output module in STOP state with module error" to "ST output module in RUN state without error"

If the relevant error (output error, module error) has been recovered, the affected section on the ST output module is reactivated automatically and the module switches to the operating state "ST output module in RUN state without error".

[7a] Switching from "Minor ST error" to "Bootling system sections"

This change in operating state affects all the system sections of the PSS 4000-exclusive device (see Flow chart of a device. Once the error has been recovered, the ST module bus PSSu can be restarted via one of the reset commands:

- Warm reset (PAS4000) or reset button (device)
- Cold reset (PAS4000)
- Warm reset of project (PAS4000)
- Cold reset of project (PAS4000)

[7b] Switching from "ST module bus PSSu in RUN state without error" to "Bootling system sections"

This change in operating state affects all system sections of the PSS 4000-exclusive device (see Flow chart of a device).

The change in operating state can be triggered using one of the following commands:

- Warm reset (PAS4000) or reset button (device)
- Cold reset (PAS4000)
- Warm reset of project (PAS4000)
- Cold reset of project (PAS4000)

12.8 Operating states and change in operating state of the ST module bus PSS u2

The description below deals exclusively with the operating states of the ST module bus PSS u2 of a PSS 4000-exclusive device, referring to the operating states of the device where necessary.

The diagram "Operating states and change in operating state of the ST module bus PSS u2" (see [Flowchart !\[\]\(666e09182d4cd268646ea700ea60dcdf_img.jpg\) 500](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and change in operating state of a device" (see [Flow chart !\[\]\(f15d3c54be60b4fd0ce1da9fb3f67256_img.jpg\) 414](#)), as the effects impact all system sections of the PSS 4000-exclusive device and not just the ST module bus PSS u2.
- ▶ The section marked with the Roman numeral Vb refers to the diagram "Operating states and change in operating state of a device" (see [Flow chart !\[\]\(7bf135d42c40a6430c927b2fd03d7659_img.jpg\) 414](#)), as the ST module bus PSS u2 is started when booting the system sections of a PSS 4000-exclusive device.

12.8.1 Flowchart

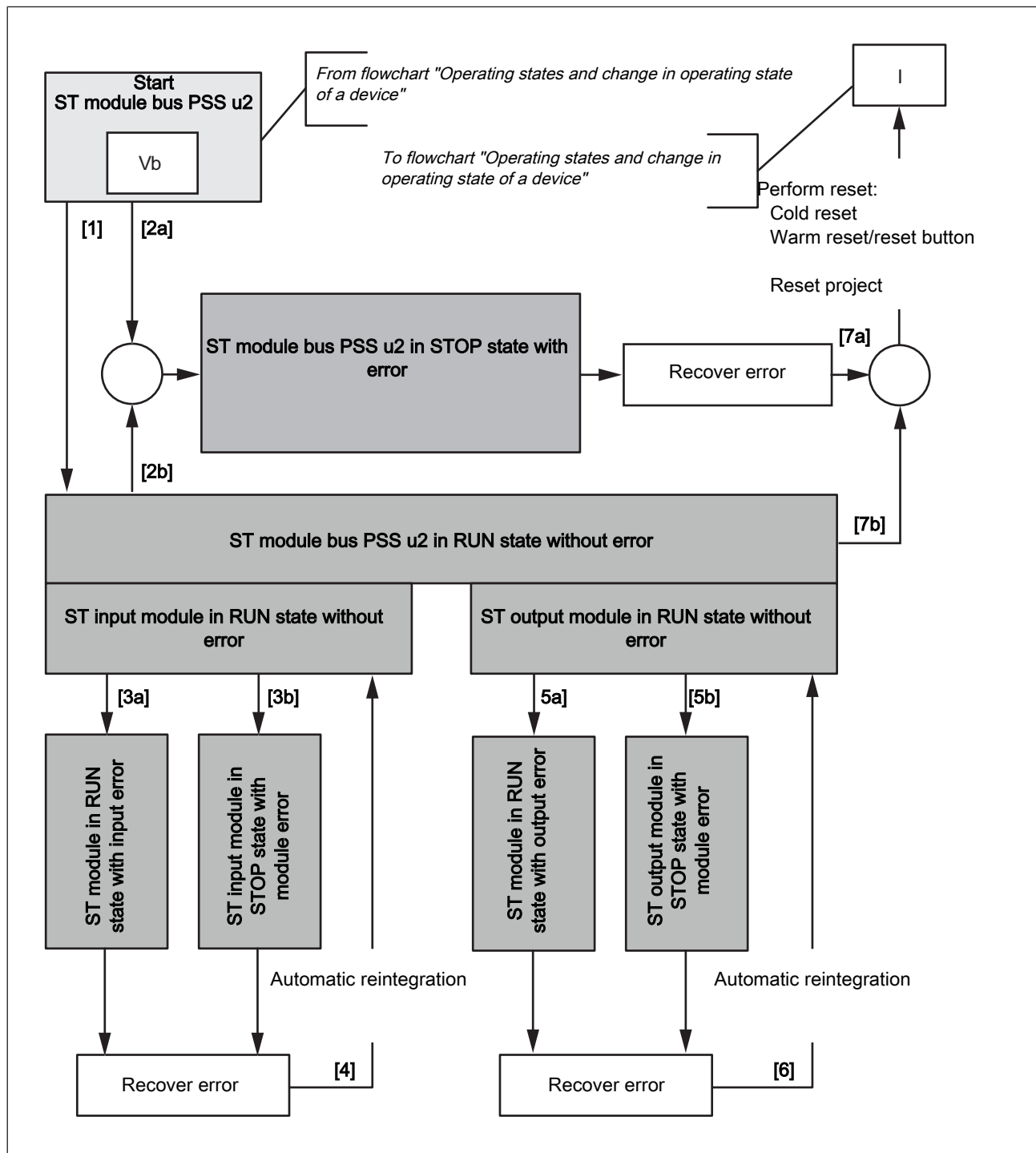


Fig.: Operating states and change in operating state of the ST module bus PSS 2

12.8.2 Operating states of the ST module bus PSS u2

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The ST module bus can adopt the following operating states (see [Flowchart \[500\]](#)):

- ▶ Operating states "ST module bus PSS u2 in RUN state without error" with
 - "ST input module in RUN state without error"
 - "ST module in RUN state with input error"
 - "ST input module in STOP state with module error"
 - "ST output module in RUN state without error"
 - "ST module in RUN state with output error"
 - "ST output module in STOP state with module error"
- ▶ Operating state "ST module bus PSS u2 in STOP state with error"



INFORMATION

The descriptions of various operating states include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[540\]](#)).

12.8.2.1 ST module bus PSS u2 in RUN state without error

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Restarting the device or the reboot command for the device via PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ The ST module bus PSS u2 is in RUN state without error.
- ▶ The ST modules are operating without error, i.e.
 - The input signals/input values present at the inputs are recorded correctly and transmitted without error via the ST module bus PSS u2
 - The outputs are activated correctly.

- ▶ Status LEDs on the device:

The "MBUS" LED lights up green, provided the FS module bus PSS u2 is also operating without error.

12.8.2.2 ST input module in RUN state without error

The operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status. The ST module bus PSS u2 is where the ST input module provides the input data for data transport to the head module.

12.8.2.3 ST module in RUN condition with input error

An input error only influences the affected ST hardware input. The ST module bus PSS u2 remains in the operating state "ST module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Input error

Effects

- ▶ The substitute value is used for the input signal/input value, with valid bit = FALSE.
- ▶ The ST input module remains in RUN state and the unaffected ST hardware inputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the ST hardware input is reactivated automatically.

12.8.2.4 ST input module in STOP state with module error

A module error only impacts the affected ST input module. The ST module bus PSS u2 remains in the operating state "ST module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation

Effects

- ▶ The substitute values are used for all the module's input signals/input values, with valid bits = FALSE.
- ▶ The unaffected ST input modules on the device continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the ST input module is reactivated automatically.

12.8.2.5 ST output module in RUN state without error

This operating state is achieved as soon as the module is ready for operation, irrespective of the module bus status. The ST output module processes the output data provided by the ST module bus PSS u2.

12.8.2.6 ST module in RUN condition with output error

An ST output error only impacts the affected ST hardware output. The ST module bus PSS u2 remains in the operating state "ST module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Output error
(e.g. overload, short circuit)

Effects

- ▶ The substitute value is used for the output signal/output value.
- ▶ The ST output module remains in RUN state and the unaffected ST hardware outputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the ST hardware output is reactivated automatically; i.e. the output is reintegrated automatically.

12.8.2.7 ST output module in STOP state with module error

A module error in an ST output module only impacts the affected ST output module. The ST module bus PSS u2 remains in the operating state "ST module bus PSS u2 in RUN state without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation

Effects

- ▶ The substitute values are used for all the module's output signals/output values.
- ▶ The unaffected ST output modules on the device continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the ST output module is reactivated automatically;
i.e. the outputs on the ST output module are reintegrated automatically.

12.8.2.8 ST module bus PSS u2 in STOP state with error

If an error results in this operating state, the affected sections of the ST module bus PSS u2 are ended as defined. A defined state is achieved.

Possible causes

- ▶ Error on the backplane
(e.g. contact interruption)

Effects

- ▶ If the error is detected when starting the ST module bus PSS u2, the ST outputs are not activated and the ST inputs are not read.
- ▶ If other devices access the ST I/O data on this device, the substitute values are used for all signals/values, with valid bits = FALSE.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Recover error
3. Once the error has been recovered, the device must be restarted via one of the reset commands.

12.8.3 Change in operating state of the ST module bus PSS u2

This description of the change in operating state assumes that the device contains a valid device project. The numbers refer to the change in operating state in the diagram [Flowchart \[500\]](#).



INFORMATION

The descriptions of each change in operating state include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[540\]](#)).

[1] Switching from "Bootling system sections" to "ST module bus PSS u2 in RUN state without error"

Bootling system sections is one of the operating states of a PSS 4000-exclusive device (see [Flow chart \[414\]](#) of a device). If no error is detected as the system sections are booted, the ST module bus PSS u2 is started and switches to the state "ST module bus PSS u2 in RUN state without error".

[2a] Switching from "Bootling system sections" to "ST module bus PSS u2 in STOP state with error"

If a fault is detected on the backplane as the system sections are booted, the ST module bus PSS u2 switches to the operating state "ST module bus PSS u2 in STOP state with error".

[2b] Switching from "ST module bus PSS u2 in RUN state without error" to "ST module bus PSS u2 in STOP state with error"

If a fault is detected on the backplane while the ST module bus PSS u2 is in RUN state, the ST module bus PSS u2 switches to the operating state "ST module bus PSS u2 in STOP state with error".

[3a] Switching from "ST input module in RUN state without error" to "ST input module in RUN state with input error"

If an input error is detected on an ST hardware input, the ST input module switches to the operating state "ST module in RUN state with input error".

[3b] Switching from "ST input module in RUN state without error" to "ST input module in STOP state with module error"

If a module error is detected on an ST input module, the affected ST input module switches to the operating state "ST input module in STOP state with module error".

[4] Switching from one of the operating states "ST input module in RUN state with input error" or "ST input module in STOP state with module error" to "ST input module in RUN state without error"

If the relevant error (input error, module error) has been recovered, the affected section on the ST input module is reactivated automatically and the module switches to the operating state "ST input module in RUN state without error".

[5a] Switching from "ST output module in RUN state without error" to "ST output module in RUN state with output error"

If an output error is detected on an ST hardware output, the ST output module switches to the operating state "ST module in RUN state with output error".

[5b] Switching from "ST output module in RUN state without error" to "ST output module in STOP state with module error"

If a module error is detected on an ST output module, the affected ST output module switches to the operating state "ST output module in STOP state with module error"

[6] Switching from one of the operating states "ST output module in RUN state with output error" or "ST output module in STOP state with module error" to "ST output module in RUN state without error"

If the relevant error (output error, module error) has been recovered, the affected section on the ST output module is reactivated automatically and the module switches to the operating state "ST output module in RUN state without error"; i.e. the outputs are reintegrated.

[7a] Switching from "ST module bus PSS u2 in STOP state with error" to "Bootling system sections"

This change in operating state affects all the system sections of the PSS 4000-exclusive device (see [Flow chart \[414\]](#) of a device). Once the error has been recovered, the ST module bus PSS u2 can be restarted via one of the reset commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[7b] Switching from "ST module bus PSS u2 in RUN state without error" to "Bootling system sections"

This change in operating state affects all system sections of the PSS 4000-exclusive device (see [Flow chart \[414\]](#) of a device).

The change in operating state can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

12.9 Operating states and changes in operating state of FS SafetyNET p

The description below deals exclusively with the operating states of the FS SafetyNET p system section of a PSS 4000-exclusive device, referring to the operating states of the device where necessary.

The diagram "Operating states and changes in operating state of FS SafetyNET p" (see [Flowchart \[513\]](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the effects impact all system sections of the PSS 4000-exclusive device and not just the system section FS SafetyNET p.
- ▶ The section marked with the Roman numeral VIa refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the system section FS SafetyNET p is started when booting the system sections of a PSS 4000-exclusive device.

12.9.1 Flow chart

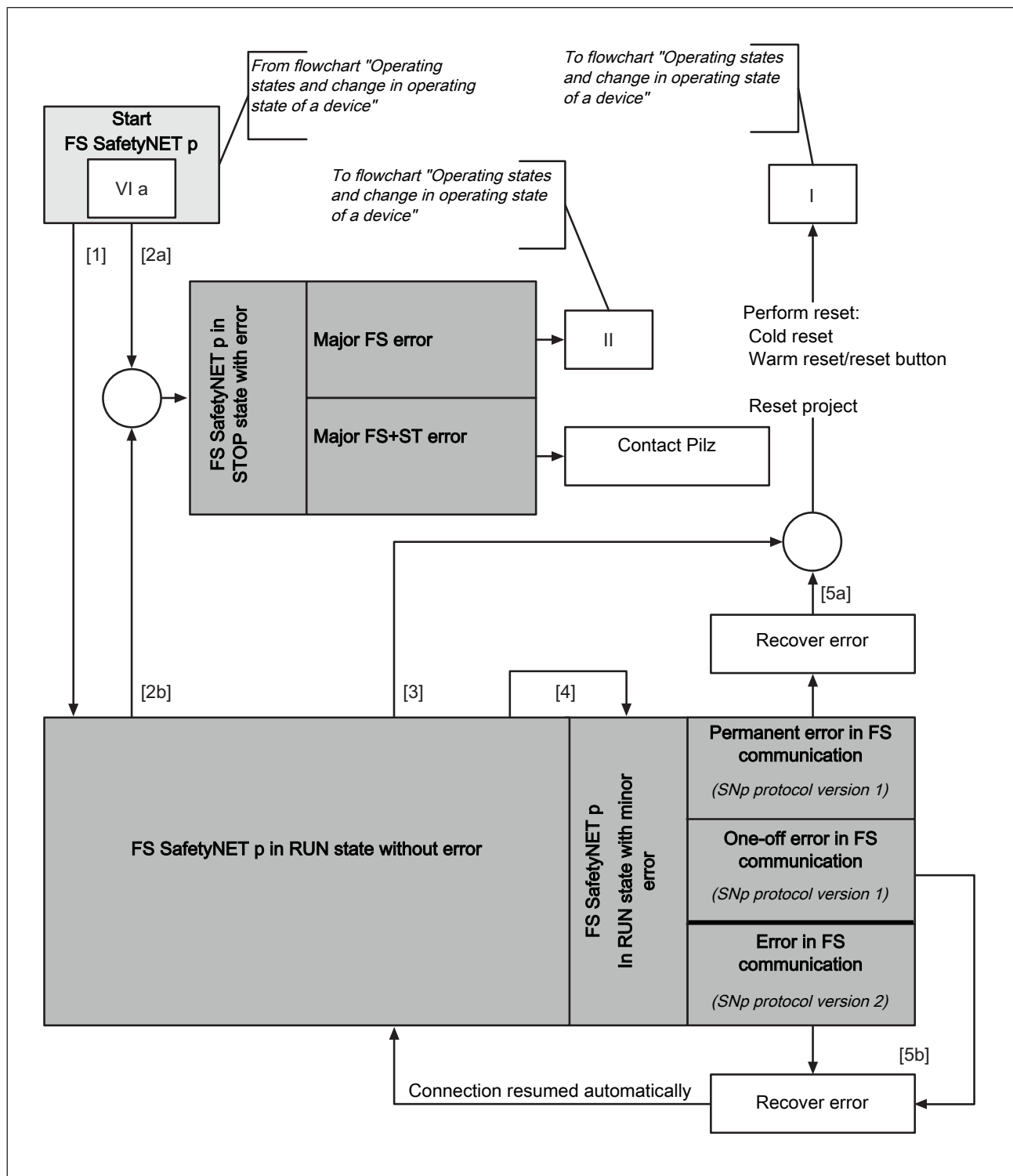


Fig.: Operating states and changes in operating state of FS SafetyNET p

12.9.2 Operating states of FS SafetyNET p

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The system section FS SafetyNET p may assume the following operating states (see [Flowchart \[513\]](#)):

- ▶ Operating state "FS SafetyNET p in RUN state without error"
- ▶ Operating state "FS SafetyNET p in STOP state with error":
 - "Major FS error"
 - "Major FS+ST error"
- ▶ Operating state "FS SafetyNET p in RUN state with minor error"
 - SafetyNET p protocol version 1:
 - "Permanent error in FS communication"
 - "One-off error in external communication"
 - SafetyNET p protocol version 2:
 - Connection-based error in FS communication
 - Device-based error in FS communication
 - Unspecific error in FS communication



INFORMATION

The description of each operating state includes a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[540\]](#)).

12.9.2.1 FS SafetyNET p in RUN state without error

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Restarting the device or the reboot command for the device via PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ In the operating state "FS SafetyNET p in RUN state without error", pending FS process data can be transmitted via SafetyNET p without error.

12.9.2.2 Major FS error

In the event of a error that leads to this operating state, the affected parts of the system section FS SafetyNET p are ended as defined. A defined state is achieved.

The effects of a major FS error impact all safety-related parts of a PSS 4000-exclusive device (see [Operating state "Major FS error" \[📖 428\]](#)).

12.9.2.3 Major FS+ST error

If an error leads to the operating state "Major FS+ST error", the affected parts of the system section FS SafetyNET p are ended as defined. A defined state is achieved.

Possible causes

- ▶ Internal error

Effects

- ▶ The system section FS SafetyNET p and the system section ST SafetyNET p both switch to a STOP state. No more process data can be exchanged with the device
- ▶ The substitute values are used for the affected process data, with valid bits = FALSE.

Remedy

Contact Pilz

12.9.2.4 SNp protocol version 1: FS SafetyNET p interface in RUN state with minor error

If an error leads to a minor error in FS communication, the affected section is stopped or ended as defined. A defined state is achieved.

Possible causes

► Permanent error in FS communication

An error is deemed to be "permanent" if at least 2 connection failures occur within 30 s.

Examples of permanent errors:

- A device cannot be reached temporarily because it has been removed from SafetyNET p.
- The bus connector on a device has been unplugged.

► One-off fault in FS communication

An error is deemed to be a "one-off" if a maximum of 1 connection failure occurs within 30 s.

Example of a one-off error:

- Transient, one-off error on the bus cable

Effects

- FS communication between the receiving device and the remote device is disabled; i.e. the process data available for further processing in the affected Rx data areas is not accepted by the device or is ignored.
- FS process data exchange with all other devices is continued.
- The system section FS SafetyNET p continues to be in a RUN state
- The substitute values are used for the affected process data, with valid bits = FALSE.

Remedy

1. Diagnostics

e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log.

2. Recover error

3. Reset or resume the connection automatically

- Permanent error in FS communication:

Perform reset (warm reset/reset button, cold reset, project reset), provided the FS SafetyNET p wait time of 1 s has elapsed and the cause of the error is no longer present.

- One-off error in FS communication:

FS communication resumes automatically provided the FS SafetyNET p wait time of 1 s has elapsed and the cause of the error is no longer present.

Once FS communication is resumed, the process values are output at the outputs.

12.9.2.5 Snp protocol version 2: FS SafetyNET p interface in RUN state with minor error

If an error leads to a minor error in FS communication, the affected section is stopped or ended as defined. A defined state is achieved.

Possible causes

- ▶ Connection-based error in FS communication
The error concerns FS communication via a single connection.
- ▶ Device-based error in FS communication
The error concerns FS communication between two devices.
- ▶ Non-specific error in FS communication
The error concerns FS communication between a device and all its remote devices. An error is "non-specific" when the cause of the error cannot be clearly assigned to any of the remote devices.

Examples:

- ▶ A device cannot be reached temporarily because it has been removed from SafetyNET p.
- ▶ The bus connector on a device has been unplugged.
- ▶ Fault in the SafetyBUS p infrastructure
- ▶ Errors on the bus cable
- ▶ Poor power supply
- ▶ Inconsistent device projects

Effects

- ▶ Type of error
 - Connection-based error in FS communication
The connection is temporarily disabled; i.e. the process data available for further processing in the corresponding Rx data area is not accepted by the device or is ignored. FS communication via other connections is unaffected.
 - Device-based error in FS communication
FS communication between the receiving device and the remote device is temporarily disabled; i.e. the process data available for further processing in the affected Rx data areas is not accepted by the device or is ignored. FS communication with other remote devices is unaffected.
 - Non-specific error in FS communication
FS communication with all remote devices is temporarily disabled; i.e. the device accepts no more process data from its remote devices.
- ▶ The system section FS SafetyNET p continues to be in a RUN state.
- ▶ The substitute values are used for the affected process data, with valid bits = FALSE.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or diagnostic log.
The diagnostic message can be used to identify which error is present.
2. Recover error

3. Automatically resume FS communication

FS communication resumes automatically provided the FS SafetyNET p wait time has elapsed and the cause of the error is no longer present. Once FS communication is resumed, the process values are output at the outputs.

12.9.3 Change in operating state of FS SafetyNET p

This description of the change in operating state assumes that the device contains a valid device project. The figures refer to the change in operating state in the diagram "Operating states and changes in operating state of FS SafetyNET p" (see [Flowchart \[513\]](#)).



INFORMATION

The descriptions of each change in operating state include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[540\]](#)).

[1] Switching from "Booting system sections" to "FS SafetyNET p in RUN state without error"

"Booting system sections" is one of the operating states of a PSS 4000-exclusive device (see Flowchart). If no error is detected in a safety-related system section as the system sections are booted, the system section FS SafetyNET p is started, switching to the state "FS SafetyNET p in RUN state without error".

[2a] Switching from "Booting system sections" to an error state

If an error is detected in a safety-related system section as the system sections are booted (see Flowchart), FS SafetyNET p switches to a STOP state, achieving the operating state "Major FS error" or "Major FS+ST error", depending on the error.

[2b] Switching from "FS SafetyNET p in RUN state without error" to an error state

If an error is detected in a safety-related system section while FS SafetyNET p is in a RUN state, FS SafetyNET p switches to a STOP state, achieving the operating state "Major FS error" or "Major FS+ST error", depending on the error.

[3] Switching from "FS SafetyNET p in RUN state without error" to "Booting system sections"

This change in operating state affects all system sections of the PSS 4000-exclusive device (see Flowchart).

This change in operating state can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[4] Switching from "FS SafetyNET p in RUN state without error" to "FS SafetyNET p in RUN state with minor error"

The change in operating state has resulted due to an FS communication error. Faults that result in this error are classified as follows:

- ▶ SafetyNET p protocol version 1:
 - Permanent error in FS communication
 - One-off fault in FS communication
- ▶ SafetyNET p protocol version 2:
 - Connection-based error in FS communication
 - Device-based error in FS communication

- Unspecific error in FS communication

[5a] Switching from "FS SafetyNET p in RUN state with error" to "FS SafetyNET p in RUN state without error"

SafetyNET p protocol version 1 only:

The operating state "FS SafetyNET p in RUN state with error" has resulted due to a permanent error in FS communication. Once the error has been recovered, the system section can be restarted via one of the reset commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[5b] Switching from "FS SafetyNET p in RUN state with error" to "FS SafetyNET p in RUN state without error"

- ▶ SafetyNET p protocol version 1:

The operating state "FS SafetyNET p in RUN state with error" has resulted due to a transient one-off error in FS communication. FS communication is resumed automatically once the FS SafetyNET p wait time has elapsed, provided the fault has been recovered and/or the error is no longer present.

- ▶ SafetyNET p protocol version 2:

The switch to "FS SafetyNET p in RUN state with error" has resulted due to a connection-based, device-based or unspecific error in FS communication. The affected connections are resumed automatically once the FS SafetyNET p wait time has elapsed, provided the fault has been recovered and/or the error is no longer present.

12.10 Operating states and changes in operating state of ST SafetyNET p

The description below deals exclusively with the operating states of the ST SafetyNET p system section of a PSS 4000-exclusive device, referring to the operating states of the device where necessary.

The diagram "Operating states and changes in operating state of ST SafetyNET p" (see [Flowchart \[524\]](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the effects impact all system sections of the PSS 4000-exclusive device and not just the system section ST SafetyNET p.
- ▶ The section marked with the Roman numeral VIb refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the system section ST SafetyNET p is started when booting the system sections of a PSS 4000-exclusive device.

12.10.1 Flow chart

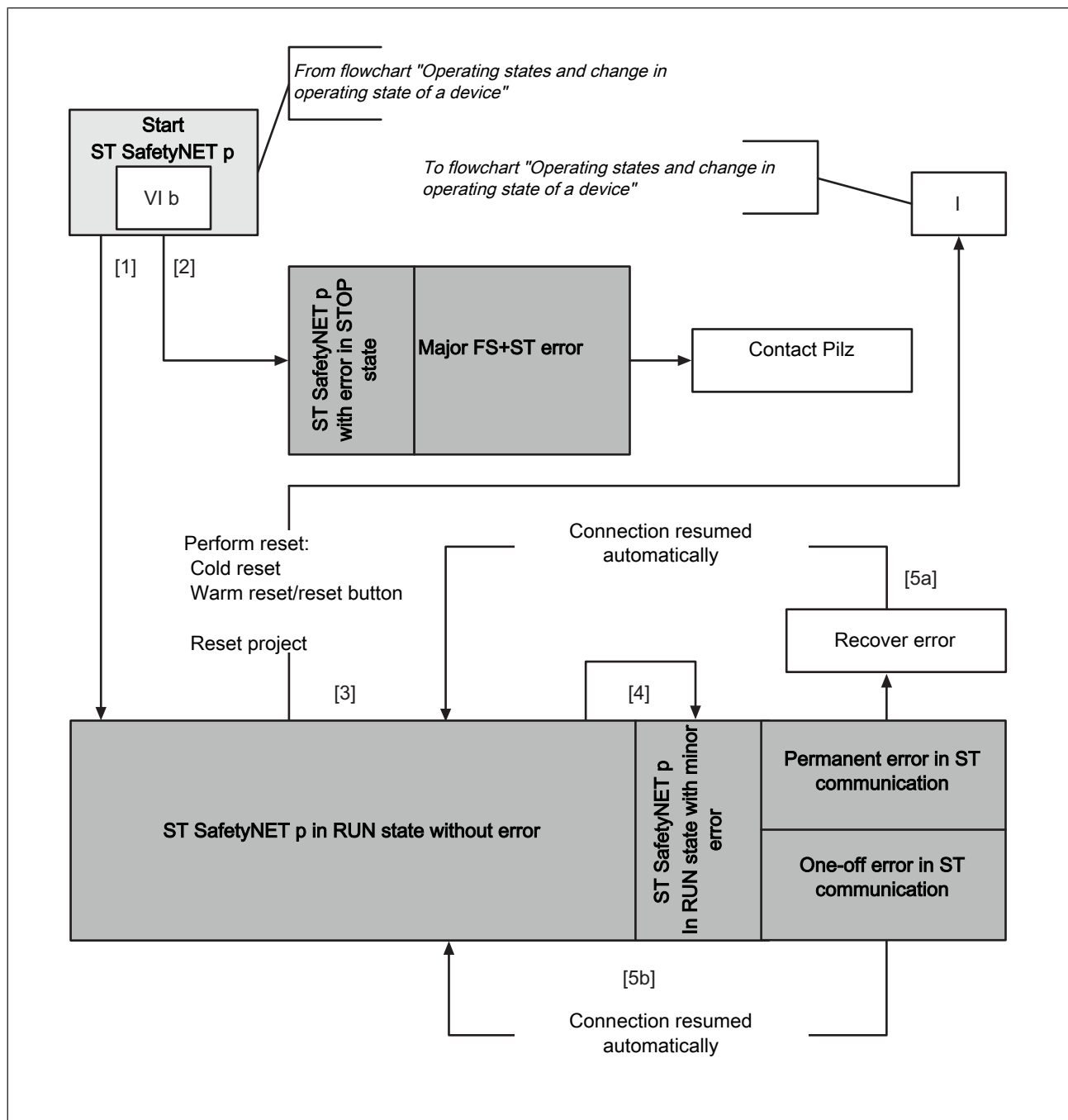


Fig.: Operating states and changes in operating state of ST SafetyNET p

12.10.2 Operating states of ST SafetyNET p

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The system section SafetyNET p may assume the following operating states (see [Flowchart \[524\]](#)):

- ▶ Operating state "ST SafetyNET p in RUN state without error"
- ▶ Operating state "ST SafetyNET p in STOP state with error":
 - "Major FS+ST error"
- ▶ Operating state "ST SafetyNET p in RUN state with minor error":
 - "Permanent error in ST communication"
 - "One-off error in ST communication"



INFORMATION

The description of each operating state includes a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[540\]](#)).

12.10.2.1 **ST SafetyNET p in RUN state without error**

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Restarting the device or the reboot command for the device via PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ In the operating state "ST SafetyNET p in RUN state without error", pending ST process data can be transmitted via SafetyNET p without error.

12.10.2.2 Major FS+ST error

If an error leads to the operating state "Major FS+ST error", the affected parts of the system section FS SafetyNET p are ended as defined. A defined state is achieved.

Possible causes

- ▶ Internal error

Effects

- ▶ The system section FS SafetyNET p and the system section ST SafetyNET p both switch to a STOP state. No more process data can be exchanged with the device
- ▶ The substitute values are used for the affected process data, with valid bits = FALSE.

Remedy

Contact Pilz

12.10.2.3 ST SafetyNET p in RUN state with minor error

In the event of a fault that leads to a minor error in ST communication, the affected section is stopped or ended as defined. A defined state is achieved.

Possible causes

► Permanent error in ST communication

Examples of permanent errors:

- A device cannot be reached temporarily because it has been removed from SafetyNET p.
- The bus connector on a device has been unplugged.

► One-off error in ST communication

Examples of a one-off error:

- Transient, one-off error on the bus cable

Effects

- ST process data is no longer exchanged with the affected device; ST process data continues to be exchanged with all other devices.
- The system section ST SafetyNET p continues to be in a RUN state.
- The substitute values are used for the affected process data, with valid bits = FALSE.

Remedy

1. Diagnostics

e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log

2. Recover error

3. Resume the connection

- Permanent error in ST communication:
The connection is resumed automatically.
- One-off error in ST communication:
The connection is resumed automatically.

12.10.3 Change in operating state of ST SafetyNET p

This description of the change in operating state assumes that the device contains a valid device project. The figures refer to the change in operating state in the diagram "Operating states and changes in operating state of ST SafetyNET p" (see [Flowchart \[524\]](#)).



INFORMATION

The descriptions of each change in operating state include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[540\]](#)).

[1] Switching from "Bootling system sections" to "ST SafetyNET p in RUN state without error"

"Bootling system sections" is one of the operating states of a PSS 4000-exclusive device (see Flowchart). If no error is detected as the system sections are booted, the system section ST SafetyNET p is started, switching to the state "ST SafetyNET p in RUN state without error".

[2] Switching from "Bootling system sections" to "Major FS+ST error"

If an error is detected as the system sections are booted (see Flowchart), the system section ST SafetyNET p switches to a STOP state, achieving the operating state "Major FS+ST error".

[3] Switching from "ST SafetyNET p in RUN state without error" to "Bootling system sections"

This change in operating state affects all system sections of the PSS 4000-exclusive device (see Flowchart).

This change in operating state can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[4] Switching from "ST SafetyNET p in RUN state without error" to "ST SafetyNET p in RUN state with minor error"

The change in operating state has resulted due to an ST communication error. Faults that result in this error are classified as follows:

- ▶ Permanent error in ST communication
- ▶ One-off error in ST communication

[5a] Switching from "ST SafetyNET p in RUN state with error" to "ST SafetyNET p in RUN state without error"

The operating state "ST SafetyNET p in RUN state with error" has resulted due to a permanent error in ST communication. Once the fault has been recovered, the connection is resumed automatically.

[5b] Switching from "ST SafetyNET p in RUN state with error" to "ST SafetyNET p in RUN state without error"

The operating state "ST SafetyNET p in RUN state with error" has resulted due to a transient one-off error in ST communication. The connection will be resumed automatically.

12.11 Operating states and change in operating status with external communication

The description below deals exclusively with the operating states with external communication (see Communication modes for external communication). The operating states are identical for the respective system section (e.g. system section "IP connections", system section "Ethernet/IP adapter"). Where necessary, reference is made to the operating states of the PSS 4000-exclusive device.

The diagram "Operating states and changes in operating state with external communication" (see [Flowchart \[531\]](#) ) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating state.
- ▶ The section marked with the Roman numeral I refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the effects impact all system sections of a PSS 4000-exclusive device and not just the system section for the respective external communication.
- ▶ The section marked with the Roman numeral VII refers to the diagram "Operating states and changes in operating state of a device" (see Flowchart), as the system section for the respective external communication is started when booting the system sections of a PSS 4000-exclusive device.

12.11.1 Flow chart

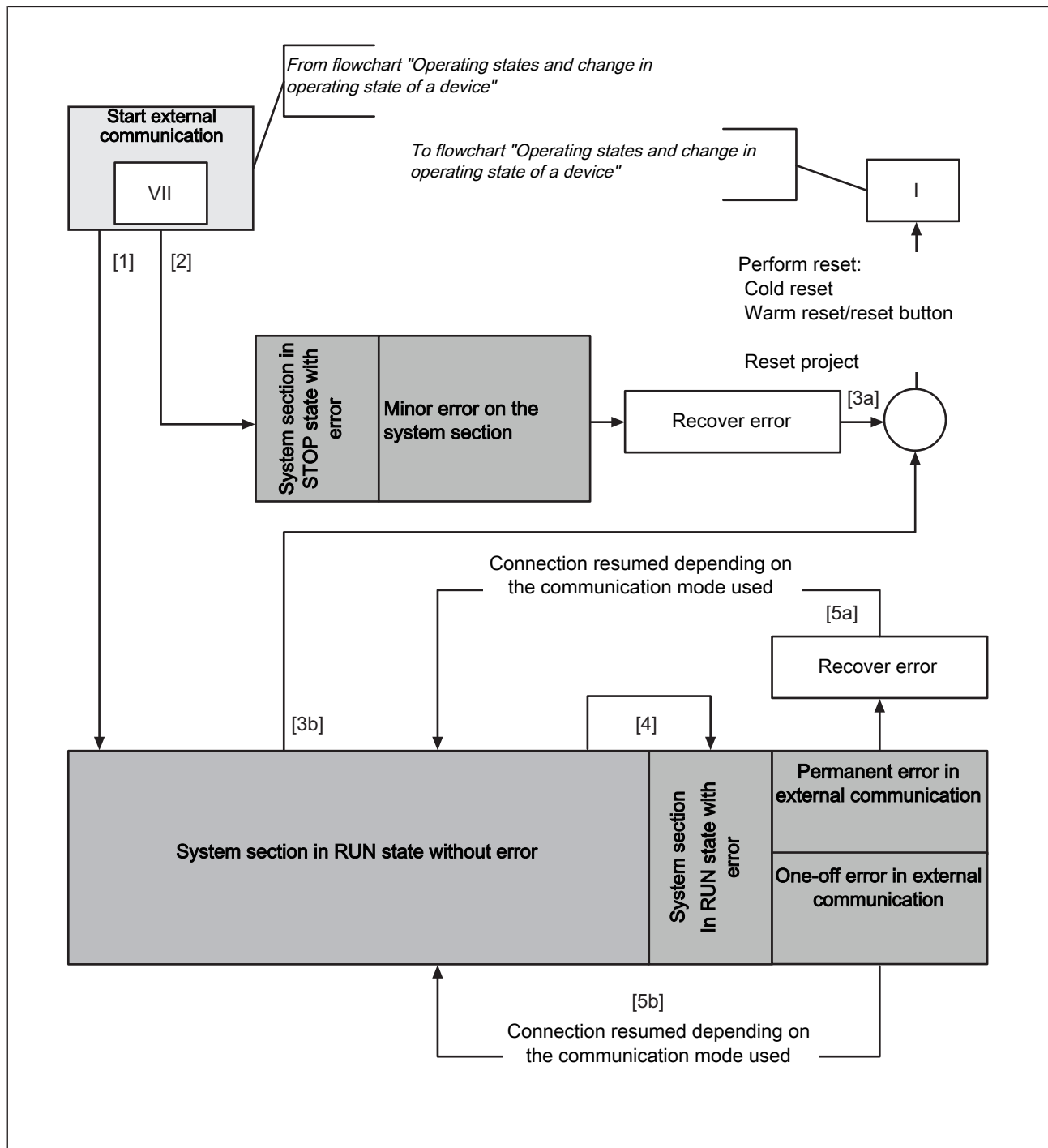


Fig.: Operating states and changes in operating state with external communication

12.11.2 Operating states with external communication

The description of the various operating states assumes that there is a valid device project on the PSS 4000-exclusive device.

The system section for the respective external communication (e.g. Modbus/TCP, Raw UDP, EtherNet/IP) may assume the following operating states (see [Flowchart \[📖 531\]](#)):

- ▶ Operating state "System section in RUN state without error"
- ▶ Operating state "System section in STOP state with error":
 - "Minor error on the system section for external communication"
- ▶ Operating state "System section in RUN state with error"
 - "Permanent error in external communication"
 - "One-off error in external communication"
- ▶ Major FS error
(only with communication via a safe communication medium such as PROFIsafe!)
- ▶ Major FS+ST error
(only with communication via a safe communication medium such as PROFIsafe!)



INFORMATION

- The descriptions of various operating states include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[📖 540\]](#)).
- In some communication modes, various operating states can be displayed via the device's status LEDs. The presence of status LEDs and their meaning depends on the communication mode used.
Examples:
Modbus/TCP, Raw TCP and Raw UDP are protocols that are based on Industrial Ethernet. They have no status LEDs to display operating states.
On PROFIBUS-DP Slave the two status LEDs "SF" and "BF" display various states.
On PROFINET IO DEVICE the status LED "BF" displays various states.
On PROFIsafe the "SO" status LED displays various states.
On EtherNet/IP adapter, the two status LEDs "MS" and "NS" display various states.

12.11.2.1 **System section in RUN condition without error**

This operating state is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Cold starting the device or the reboot command for the device in PAS4000
 - Pressing the reset button on the device
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ When the operating state of the respective external communication (e.g. Modbus/TCP, Raw UDP, EtherNET/IP) is "System section in RUN state without error", pending process data is transmitted without error using the respective communication mode.
- ▶ Status LEDs on the device:
Depend on the communication mode used

12.11.2.2 Minor error on the system section for external communication

In the event of an error that leads to this operating state, the affected parts of the system section are ended as defined. A defined state is achieved.

Possible causes

- ▶ Configuration error

Effects

- ▶ System section for the respective external communication switches to the STOP state. Process data cannot be exchanged with the device via the corresponding communication mode.
- ▶ The substitute values are used for the affected process data, with valid bits = FALSE.
- ▶ Status LEDs on the device:
Depend on the communication mode used

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log.
2. Recover error
e.g. synchronize the configuration of the communication partners
3. Perform reset

12.11.2.3 System section in RUN condition with error

If an error leads to an error in external communication, the affected system section is stopped or ended as defined. A defined state is achieved.

Possible causes

► Permanent error in external communication

Examples of permanent errors:

- The communication partner is not accessible because it was removed.
- The bus connector on the device has been unplugged.

► One-off error in external communication

Examples of a one-off error:

- Transient, one-off error on the bus cable, telegram loss

Effects

- Process data is no longer exchanged with the communication partner.
- The system section continues to be in a RUN state.
- The substitute values are used for the affected process data, with valid bits = FALSE.
- Status LEDs on the device:

Depend on the communication mode used

Remedy

1. Diagnostics

e.g. use PAS4000 to read the diagnostic list and/or diagnostic log

2. Recover error

3. Resume the connection

The options for resuming the connection depend on the communication mode that is used and in various cases also on the configuration options of the communication partners.

If this is supported by the communication mode, the corresponding system section automatically resumes the connection. Otherwise, the system section must be restarted via one of the reset commands:

- Warm reset (PAS4000) or reset button (device)
- Cold reset (PAS4000)
- Warm reset of project (PAS4000)
- Cold reset of project (PAS4000).



INFORMATION

Please note:

- Under some circumstances it is necessary to restart the communication partner
- The fault may have been caused by the communication partner. In this case the error on the communication partner must be recovered first, before the system section of the device can restore the connection.

12.11.2.4

Major FS+ST error

This operating state can only occur if the system section has a safety-related communication channel (e.g. PROFIsafe from the PROFIBUS-DP Slave with PROFIsafe system section). If an error leads to this operating state, the affected system section is ended as defined.

Possible causes

- ▶ Internal error

Effects

- ▶ The system section (e.g. PROFIBUS-DP Slave with PROFIsafe) switches to a STOP state.
- ▶ The substitute values are used for the affected process data, with valid bits = FALSE.

Remedy

Contact Pilz

12.11.2.5 Major FS error

This operating state can only occur if the system section has a safety-related communication channel (e.g. PROFIsafe from the PROFIBUS-DP Slave with PROFIsafe system section). In the event of a error that leads to this operating state, communication via the system section's safety-related communication channel is ended as defined (e.g. PROFIsafe from the system section PROFIBUS-DP Slave with PROFIsafe). A defined state is achieved.

The effects of a major FS error impact all safety-related parts of a PSS 4000-exclusive device (see [Operating state "Major FS error" !\[\]\(feabb98897b440bc8695a03336a6e2df_img.jpg\) 438](#)).

12.11.3 Operating status change with external communication

This description of the change in operating state assumes that the device contains a valid device project. The figures refer to the change in operating state in the diagram "Operating states and changes in operating state with external communication" (see

[Flowchart \[📖 531\]](#)).



INFORMATION

The descriptions of each change in operating state include a list of the options for triggering a change in operating state. These options are described separately (see [Reset, restart, start and stop options \[📖 540\]](#)).

[1] Switching from "Bootling system sections" to "System section in RUN state without error"

Bootling system sections is one of the operating states of a PSS 4000-exclusive device (see Flowchart). If no error is detected as the system sections are booted, the system section for the respective external communication is started, switching to the state "System section in RUN state without error".

[2] Switching from "Bootling system sections" to "Minor error on the system section for external communication"

If an error is detected as the system sections are booted (see Flowchart), the system section switches to a STOP state, achieving the operating state "Minor error on the system section for external communication".

[3a] Switching from "Minor error on the system section for external communication" to "Bootling system sections"

Once the error has been recovered, the system section for external communication can be restarted via one of the reset commands. This change in operating state affects all the device's system sections (see Flowchart).

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[3b] Switching from "System section in RUN state without error" to "Bootling system sections"

This change in operating state affects all the device's system sections (see Flowchart).

This change in operating state can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[4] Switching from "System section in RUN state without error" to "System section in RUN state with error"

The change in operating state has resulted due to a communication error. Faults that result in this error are classified as follows:

- ▶ Permanent error in external communication

- ▶ One-off error in external communication.

[5a] Switching from "System section in RUN state with error" to "System section in RUN state without error"

The operating state "System section in RUN state with error" has resulted due to a permanent error in external communication. Once the error has been recovered, the options for resuming the connection depend on the communication mode that is used and in various cases also on the configuration options of the communication partners.

If the actual cause of the error does not lie with the communication partner, the system section for external communication can be restarted with one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[5b] Switching from "System section in RUN state with error" to "System section in RUN state without error"

The operating state "System section in RUN state with error" has resulted due to a transient one-off error in external communication. The options for resuming the connection depend on the communication mode that is used and in various cases also on the configuration options of the communication partners.

If the actual cause of the error does not lie with the communication partner, the system section for external communication can be restarted with one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (device)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

12.12 Reset, restart, start and stop

PSS 4000 operates in accordance with the principle of "de-escalation". The aim of the principle of de-escalation is to limit the effects of intervening in the control process (e.g. when a fault occurs) to the affected areas if possible. All unaffected areas should continue operating without disruption if possible, and without repercussions on the safety function.

Various options are available to stop and start specific plant/machine sections, during commissioning for example. The implications of intervening in the control process depend on the specific command that is used.

12.12.1 Overview

12.12.1.1 Reset, restart, start and stop options

The following table provides an overview of the reset, restart, start and stop options and the corresponding commands in PAS4000:

Options	PAS4000
Cold start of a device (affects all of a device's system sections)	- - -
Reboot a project (affects all system sections on all the project's devices)	Reboot
Reboot a device (affects all of a device's system sections)	Reboot
Reset a project (affects all system sections on all the project's devices)	Cold reset
	Warm reset
Reset of device (affects all of a device's system sections)	Cold reset
	Warm reset
Reset via a device's reset button	- - -
Restart a project (affects all resources on all devices)	Cold restart
	Warm restart
Restart a resource (affects a device's resource)	FS cold restart
	ST cold restart
	FS warm restart
	ST warm restart
Start tasks on a resource (affects one or more tasks on a resource)	FS hot start
	ST hot start
Stop a project (affects all resources on all the project's devices)	Stop
Stop a device (affects all of a device's resources)	Stop
Stop a resource (affects a device's resource)	FS stop
	ST stop

12.12.1.2 Impact on variables and application parameters

The following table provides an overview of the impact of a reset, restart, start or stop on local variables (VAR), resource global variables (VAR_GLOBAL, not O-PI variables) and application parameters.

After one of the stated commands, the affected sections (e.g. tasks on a resource, system sections) are started up with the following variable values:

Action	Variable with attribute RETAIN PERSISTENT	Variable Non-retentive [1]	Application parameters
Cold start Reboot	Saved value	Variable is set to initial value	Value configured for the application parameter
Warm restart Warm reset			
Cold restart Cold reset	Variable is set to initial value		
Hot start	Saved value		
Stop	Saved value is retained		

[1] If a non-retentive variable is used as the application parameter, then the "Application parameter" column applies.

Notes

- ▶ The content of the variable with the attribute RETAIN PERSISTENT is saved after a task is executed and is retained even in the event of a power failure or restart/reboot.
- ▶ The saved content of the variable with the attribute RETAIN PERSISTENT may come from the penultimate task execution before the power failure.
If several variables are saved, some may come from the penultimate task execution and others from the last execution

12.12.1.3 Impact on active "variable forcing"

The following table provides an overview of the impact of a reset, restart, start or stop on active "variable forcing":

Action	Impact on "variable forcing"
Cold start Reboot	"Variable forcing" is stopped, "variable forcing" will then need to be started manually
Warm restart Warm reset	
Cold restart Cold reset	
Hot start	"Variable forcing" is stopped, "variable forcing" will then be started automatically through the system
Stop	"Variable forcing" is stopped, the way in which "variable forcing" is restarted depends on the subsequent (re-) start action/reset action: - Warm restart/warm reset - Cold restart/cold reset - Hot start

12.12.1.4 Impact on "variable controlling"

"Variable controlling" is executed once only. A reset, restart, start or stop command only has an impact if "variable controlling" has not yet been run.

The following table provides an overview of the impact of a reset, restart, start or stop on "variable controlling":

Action	Impact on "variable controlling"
Cold start Reboot	"Variable controlling" actions that have not yet been run are deleted
Warm restart Warm reset	
Cold restart Cold reset	
Hot start	"Variable controlling" actions that have not yet been run are run after a hot start
Stop	"Variable controlling" is not run, whether the "variable controlling" action is run depends on the subsequent (re-) start action/reset action: - Warm restart/warm reset - Cold restart/cold reset - Hot start

12.12.1.5 Effects during active online changes

The following table provides an overview of the effects of a reset, restart, start or stop when online changes are active:

Action	Effect on active online changes
Cold start Reboot	Online changes are disabled,
Warm restart Warm reset	
Cold restart Cold reset	
Hot start	Online changes remain active
Stop	Online changes remain active Further effects depend on the subsequent (re-)start action/reset action: - Warm restart/warm reset - Cold restart/cold reset - Hot start

12.12.2 Cold start and Reboot command

The effect of a restart (supply voltage for the module supply is switched off and on) and the reboot command is device-based, i.e. they affect all system sections on a PSS 4000-exclusive device (FS resource, ST resource etc.).

Procedure

Performing a restart/reboot command initiates the following procedure:

1. Processing is aborted
2. Device switches to the operating state "Bootng base system".
3. Operating state "Bootng system sections" is achieved and all system sections are started
4. Operational entry in the diagnostic log
5. Impact

The impact on variables and application parameters is described separately (see [Impact on variables and application parameters](#) [542]). Impact on "variable forcing" (see [Impact on active "variable forcing"](#) [543]) and on "Online changes" (see [Impact on active online changes](#) [545]) is also described separately.

Behaviour during a cold restart/reboot command

The table below describes the behaviour of the process data and system sections:

Process data, system sections		Behaviour	
Process data	I-data of the module bus with I/O mapping [1]	Bit modules	Inputs = current physical input signals
		Byte modules	Inputs = current physical input values
	O-data of module bus	Bit modules	Outputs = ZERO
		Byte modules	Outputs = switch-on values (see module's operating manual)
	O-PI variables with I/O mapping [1]	O-PI variables are marked as invalid	
System sections	FS resource	See "Procedure"	
	ST resource		
	FS module bus		
	ST module bus		
	FS SafetyNET p		
	ST SafetyNET p		
	External communication e.g. "IP connections" system section, "PROFIBUS-DP Slave" system section		

[1]

Process data that is accessed by other devices is set to substitute values in the other devices, valid bits = FALSE (see [Validity of process data](#) [359]).

12.12.3 Reset commands

The effect of a reset (cold reset, warm reset) is device-based, i.e. it affects all system sections (FS resource, ST resource etc.) of a PSS 4000-exclusive device. The main difference between a cold reset and a warm reset is the way in which the non-volatile variables are handled.

A cold reset can be used when a defined output state is to be brought about on a device in respect of the variables.

A warm reset can be used when the desired behaviour on a device is the same as that for a restart/reboot command in respect of the non-volatile variables.

The following resets are available:

- ▶ Cold resets
 - Cold reset of all a project's devices
 - Cold reset of a device
- ▶ Warm resets:
 - Warm reset of all a project's devices
 - Warm reset of a device
(PAS4000 or reset button)

Procedure

Performing a reset initiates the following procedure:

1. All system sections are stopped
 - Resetting a project:

The tasks on all resources and all other system sections on all the project's devices are run through to the end; any further execution is then stopped.
 - Resetting a device:

The tasks on all resources and all of the device's other system sections are run through to the end; any further execution is then stopped.
2. System sections are booted
 - Resetting a project:

All the project's devices switch to the operating state "Bootting system sections", including starting all system sections and all tasks on all resources that were in a RUN state or had a task error prior to the reset.
 - Resetting a device:

Selected device switches to the operating state "Bootting system sections", including starting all system sections and all tasks on all resources that were in a RUN state or had a task error prior to the reset.
3. Diagnostic log
 - Resetting a project:

Operational entry is made in the diagnostic log for each device in the project
 - Resetting a device:

Operational entry is made in the diagnostic log for the selected device

4. Diagnostic list

- Resetting a project:
Diagnostic list is deleted
- Resetting a device:
Messages for the selected device are deleted from the diagnostic list

5. Impact

The impact on variables and application parameters is described separately (see [Impact on variables and application parameters](#) [542]). Impact on "variable forcing" (see [Impact on active "variable forcing"](#) [543]) and on "Online changes" (see [Impact on active online changes](#) [545]) is also described separately.

Behaviour during a reset

The table below describes the behaviour of the process data and system sections during a reset:

Process data, system sections		Behaviour	
Process data	I-data of the module bus with I/O mapping [1]	Bit modules	Inputs = current physical input signals
		Byte modules	Inputs = current physical input values
	O-data of module bus	Bit modules	Outputs = ZERO
		Byte modules	Outputs = switch-on values (see module's operating manual)
	O-PI variables with I/O mapping [1]	O-PI variables are marked as invalid	
System sections	FS resource	See section entitled "Procedure"	
	ST resource		
	FS module bus		
	ST module bus		
	FS SafetyNET p		
	ST SafetyNET p		
	External communication e.g. "IP connections" system section, "PROFIBUS-DP Slave" system section		

[1]

Process data that is accessed by other devices is set to substitute values in the other devices, valid bits = FALSE (see [Validity of process data](#) [359]).

12.12.4 Hot start commands

A hot start (FS hot start, ST hot start) effects the following system sections of a PSS 4000-exclusive device:

- ▶ FS hot start: affected tasks belonging to the FS resource
- ▶ ST hot start: affected tasks belonging to the ST resource

With a hot start, the process is continued exactly at the point at which it was previously stopped (runtime error or stop command). A hot start has no effect on the diagnostic list, i.e. the entries are retained.

A hot start is suitable for commissioning, for example. One of the stop commands (see [Stop commands](#)  552]) can be used to halt a targeted subprocess (e.g. for adjustment, step-by-step commissioning), which is then continued with a hot start.

Hot start in operating state "Resource in RUN state with task error"

Procedure for an FS/ST hot start while the FS/ST resource is in a "FS/ST resource is in a RUN state with a task error: Task in a STOP state":

Prerequisite: The error has been recovered

1. Operational entry is made in the diagnostic log
2. Affected tasks on the FS/ST resource are started

Only those resource tasks that were no longer run due to an error are started. A hot start does not affect any tasks on the FS/ST resource that have run without error.

3. Impact

The impact on variables and application parameters is described separately (see [Impact on variables and application parameters](#)  542]). Impact on "variable forcing" (see [Impact on active "variable forcing"](#)  543]) and on "Online changes" (see [Impact on active online changes](#)  545]) is also described separately.

Hot start after a stop command

Procedure for an FS/ST hot start after a stop command:

Prerequisite: No task errors are present.

1. Operational entry is made in the diagnostic log
2. Affected tasks on the FS/ST resource are started

All tasks on the selected resource are started. The user program is continued at the point at which it was previously stopped via the stop command

3. Impact

The impact on variables and application parameters is described separately (see [Impact on variables and application parameters](#)  542]). Impact on "variable forcing" (see [Impact on active "variable forcing"](#)  543]) and on "Online changes" (see [Impact on active online changes](#)  545]) is also described separately.

12.12.5 Restart commands

The effect of a restart (cold restart, warm restart) is resource-based, i.e. it affects the selected resource (ST/FS resource) on a PSS 4000-exclusive device). The main difference between a cold restart and a warm restart is the way in which it handles the non-volatile variables used to start a resource after a restart.

A cold restart can be used when a defined output status is to be brought about for a resource (FS resource/ST resource) in respect of the variables.

A warm restart can be used when the desired behaviour for a resource (FS resource/ST resource) is the same as that for a cold start/reboot command in respect of the non-volatile variables.

The following restarts are available:

- ▶ Cold restart for a project
- ▶ Warm restart for a project
- ▶ FS cold restart for the FS resource on a device
- ▶ ST cold restart for the ST resource on a device
- ▶ FS warm restart for the FS resource on a device
- ▶ ST warm restart for the ST resource on a device

As a restart is always preceded by an automatic stop (error on the ST/FS resource) or a manually executed stop (stop command), the behaviour of the affected process data and system sections is determined by the resource's STOP state. While a resource is in a STOP state, the process data and system sections always behave in the same way, so it does not matter how the STOP state was triggered (see [Stop commands – Behaviour during the STOP state](#) [ 552]).

Restart in the operating state "Resource in RUN state without error"

Procedure for a restart while the selected resource is in a RUN state without error:

1. Resources are stopped
 - Restarting the resources on a project's devices:
On all the project's devices, the resources (FS and ST resources) are run through to the end; any further execution is then stopped.
 - Restart for resource on a device:
The selected resource (FS/ST resource) on the device is run through to the end; any further execution is then stopped.
2. Diagnostic log
 - Restarting a project:
Operational entry is made in the diagnostic log for each of the project's resources
 - Restart for a resource on a device:
Operational entry is made in the diagnostic log for the selected resource
3. Diagnostic list
 - Restarting a project:
Messages for all the resources are deleted from the diagnostic list.
 - Restart for a resource on a device:

Messages for the selected resource are deleted from the diagnostic list

4. Resources are started

- Cold/warm restart for a project:

The resources (FS and ST resources) are started on all the project's devices.

- Cold/warm restart for a resource on a device:

The selected resource (FS/ST resource) on the device is started.

5. Impact

The impact on variables and application parameters is described separately (see [Impact on variables and application parameters](#)  542]). Impact on "variable forcing" (see [Impact on active "variable forcing"](#)  543) and on "Online changes" (see [Impact on active online changes](#)  545) is also described separately.

Restart after a stop command/error

Procedure for restarting after a stop command/error:

Prerequisite in the event of an error: The error has been recovered

1. Diagnostic log

- Restarting a project:

Operational entry is made in the diagnostic log for each of the project's resources

- Restart for a resource on a device:

Operational entry is made in the diagnostic log for the selected resource

2. Diagnostic list

- Restarting a project:

Messages for all the resources are deleted from the diagnostic list

- Restart for a resource on a device:

Messages for the selected resource are deleted from the diagnostic list

3. Resources are started

- Restarting a project:

The resources (FS and ST resources) are started on all the project's devices.

- Cold/warm restart of a device:

The selected resource (FS/ST resource) is started.

4. Impact

The impact on variables and application parameters is described separately (see [Impact on variables and application parameters](#)  542]). Impact on "variable forcing" (see [Impact on active "variable forcing"](#)  543) and on "Online changes" (see [Impact on active online changes](#)  545) is also described separately.

12.12.6 Stop commands

The following stop commands are available:

► Stopping a resource

The command is resource-based, i.e. it affects the selected resource on a device:

- Stop FS resource: Stops the FS resource on a device
- Stop ST resource: Stops the ST resource on a device

► Stopping a device

The command is device-based, i.e. it affects all resources (FS resource, ST resource) on the selected device.

► Stopping a project

The command is device-based, i.e. it affects all resources (FS resource, ST resource) on all a project's devices.

Procedure when stopping a resource

Stopping the selected resource (FS resource/ST resource) on a device initiates the following procedure:

1. All tasks on the selected resource are stopped

The effects on variables and application parameters processed via the selected resource are described separately (see [Effects on variables and application parameters](#) [542]). Effects on "variable forcing" (see [Effects on active "variable forcing"](#) [543]) and on "Online changes" (see [Effects on active online changes](#) [545]) are also described separately.

2. Operational entry in the diagnostic log for the selected resource
3. Message is entered in the diagnostic list

Procedure for stopping a device

Stopping a device initiates the following procedure:

1. All resources on the selected device are stopped

The effects on variables and application parameters processed via the selected resources are described separately (see [Effects on variables and application parameters](#) [542]). Effects on "variable forcing" (see [Effects on active "variable forcing"](#) [543]) and on online changes (see [Effects on active online changes](#)) are also described separately.

2. Operational entries are made in the diagnostic log for the resources on the device
3. Messages are entered in the diagnostic list

Procedure for stopping a project

Stopping a project initiates the following procedure:

1. All resources on all the project's devices are stopped

The effects on variables and application parameters processed via the selected resources are described separately (see [Effects on variables and application parameters](#) [542]). Effects on "variable forcing" (see [Effects on active "variable forcing"](#) [543]) and on online changes (see [Effects on active online changes](#)) are also described separately.

2. Operational entries are made in the diagnostic log for all of the project's resources
3. Messages are entered in the diagnostic list

Behaviour during the STOP state

The behaviour of the process data and system sections while a resource is in a STOP state is always the same, whether the STOP state was triggered by an error or a manual stop command. The table below describes the behaviour of the process data and system sections while a resource is in a STOP state:

Process data, system sections		Behaviour	
Process data	I-data of the module bus with I/O mapping	Bit modules	Inputs = current physical input signals
	(for "Stopping a device" see [1])	Byte modules	Inputs = current physical input values
	O-data of module bus with I/O mapping	Bit modules	Outputs = ZERO
	(for "Stopping a resource" see [2])	Byte modules	Outputs = switch-on values (see module's operating manual)
	O-PI variables of the stopped resource or resources	O-PI variables are marked as invalid (see [1] and [2])	
System sections	FS resource	See sections - Procedure for stopping a resource - Procedure for stopping a device - Procedure for stopping a project	
	ST resource		
	FS module bus	No effect	
	ST module bus		
	FS SafetyNET p		
	ST SafetyNET p		
	External communication e.g. "IP connections" system section, "PROFIBUS-DP Slave" system section		

[1]

For process data that is accessed by other PSS 4000-exclusive devices, substitute values are used in the other PSS 4000-exclusive devices, valid bits = FALSE (see [Validity of process data](#)  359).

[2]

Only the hardware outputs that are assigned to the O-PI variables on the stopped resource behave as described in the above table. Hardware outputs that are assigned to the O-PI variables on other (running) resources continue to be written by these variables.

12.13 Functions of the reset button

All PSS 4000-exclusive devices have a reset button.

The reset button has various functions:

- ▶ Perform a warm reset for a device
- ▶ Deliberate operator action to transfer the naming data and/or device project from the SD card
- ▶ Switching to recovery mode
- ▶ Manual call of commissioning mode to transfer Client certificates on PSS 4000-exclusive devices with system section OPC Server UA

12.13.1 Perform a warm reset for a device

A reset via the reset button of a PSS 4000-exclusive device corresponds to a warm reset. For a warm reset, the reset button on the device may be pressed for max. 5 s.

Further information is available in the flowchart for Operating status "Booting system sections".

The general effects of a warm reset are described separately (see [Reset commands](#)  547]).

12.13.2 Deliberate operator action to transfer the naming data and/or device project

The naming data and/or device project can be transferred from the SD card to the PSS 4000-exclusive device using the reset button. This procedure may be beneficial during service operations, for example, as there is no need to use PAS4000.

If you do not wish to use PAS4000, it is absolutely essential to perform the deliberate operator action using the reset button in the following cases:

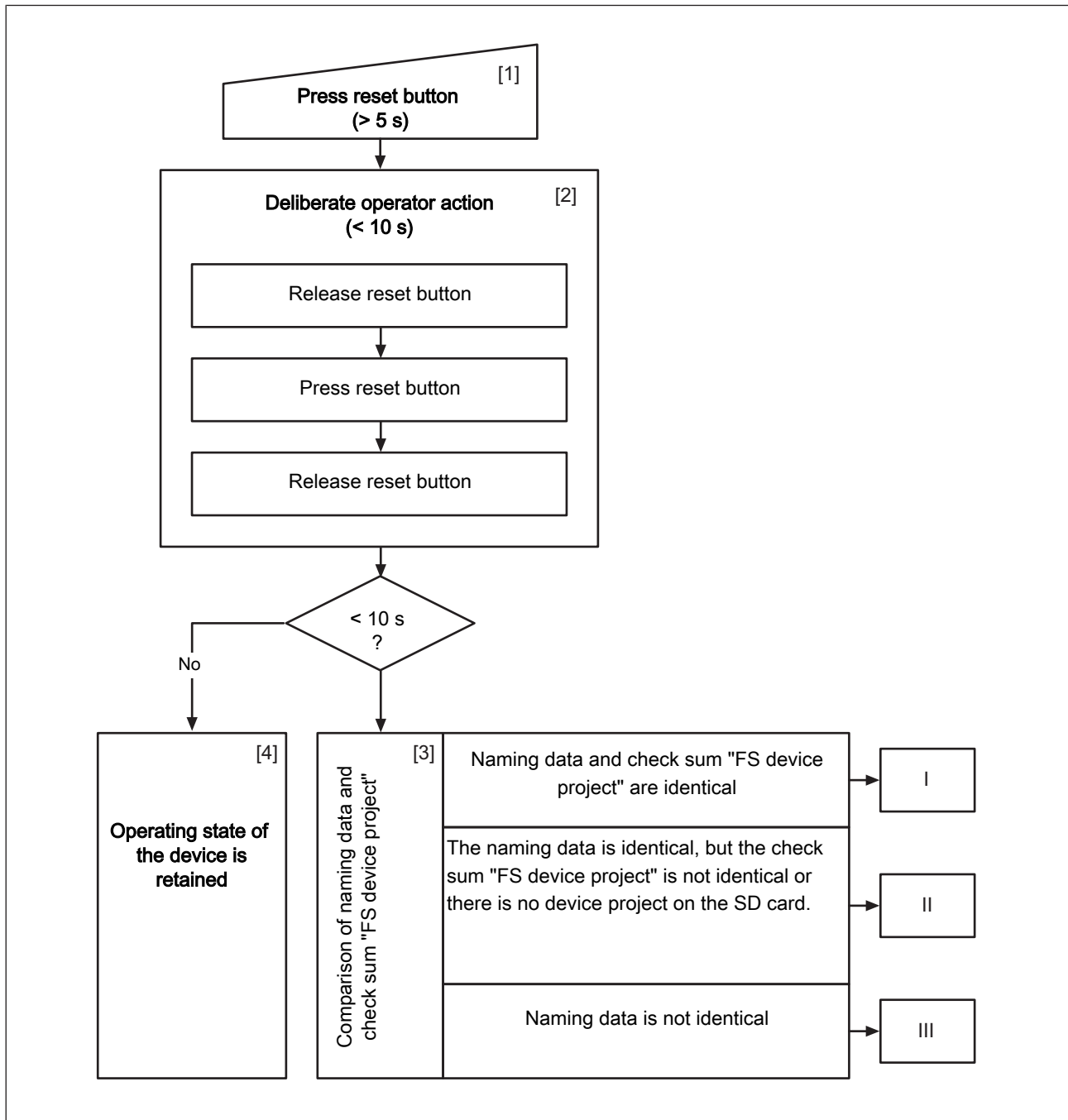
- ▶ New naming data is to be transferred from the SD card. This applies to all PSS 4000-exclusive devices, with the exception of the following:
 - PSSu H FS SN SD(-T)(-R)
 - PSSu H FS SN SD M12 (-T)(-R)
 - PSS67 IO1 16FDI
- ▶ For a device with identical naming data, the project is to be transferred from the SD card. This applies to all PSS 4000-exclusive devices, with the exception of the following:
 - PSSu H FS SN SD(-T)(-R)
 - PSSu H FS SN SD M12 (-T)(-R)
 - PSS67 IO1 16FDI

Note: If the SD card contains different naming data and a different device project, the deliberate operator action must be performed twice. The first time, only the naming data is transferred.

Prerequisites

- ▶ An SD card must be inserted on the device.
- ▶ The SD card may not be empty or defective.
- ▶ The device project on the SD card must match the product type.

Procedure for transferring the naming data and/or device project



[1]

The reset button on the device is pressed for longer than 5 s. The device remains unchanged in its current operating state.

[2]

Deliberate operator action within 10 s:

Release reset button – Press reset button – Release reset button

The start of deliberate operator action is displayed via a status LED on the device):

- ▶ "DIAG" LED on a PSSu system: flashing red-green
- ▶ "SD" LED on a PSS u2 system: flashing yellow

Note: As soon as the LED flashes, the reset button must be released within 5 s. otherwise the device reboots.

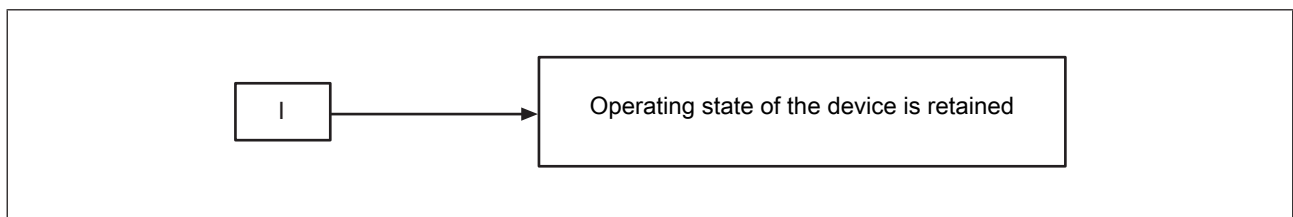
[3]

When the naming data and the check sum "FS device project" on the SD card are compared with the corresponding data on the device, a distinction is made between various cases. The sections marked with the Roman numerals I, II and III refer to the corresponding diagrams and description.

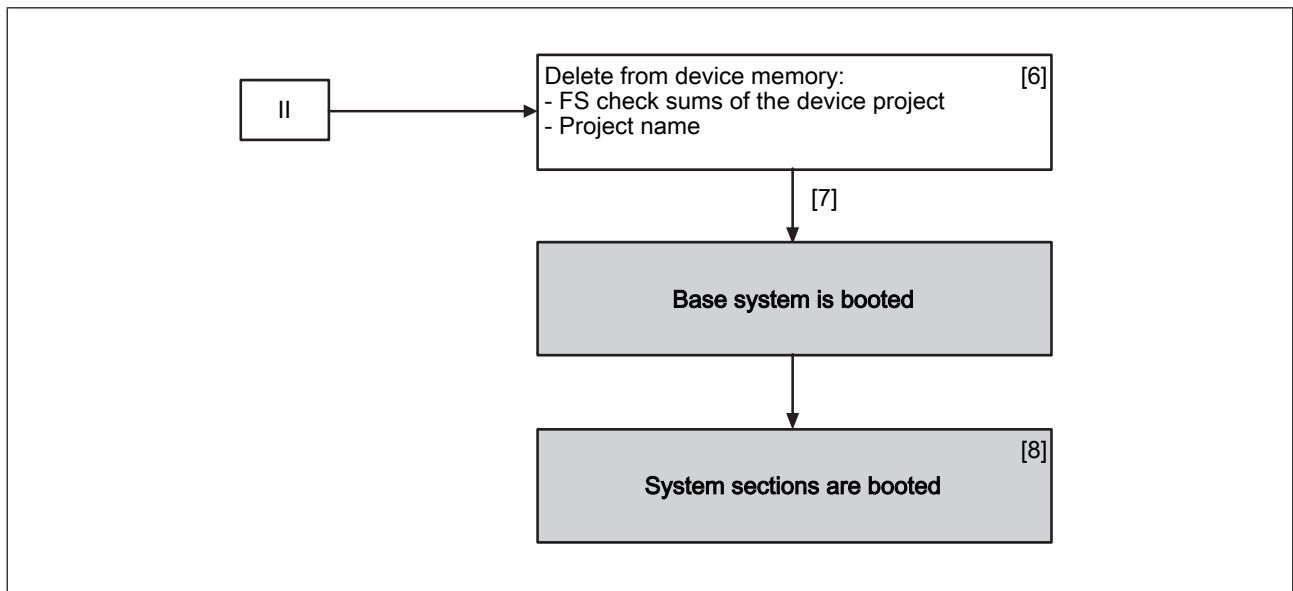
[4]

If deliberate operator action is not performed within 10 s, the device remains unchanged in its current operating state.

Procedures after the comparison



The naming data and check sum "FS device project" are identical. The device remains unchanged in its current operating state.



[6]

If the naming data is identical, but the check sum "FS device project" is not identical, or there is no device project on the SD card, the check sum "FS device project" and the project name are deleted from the device's device memory.

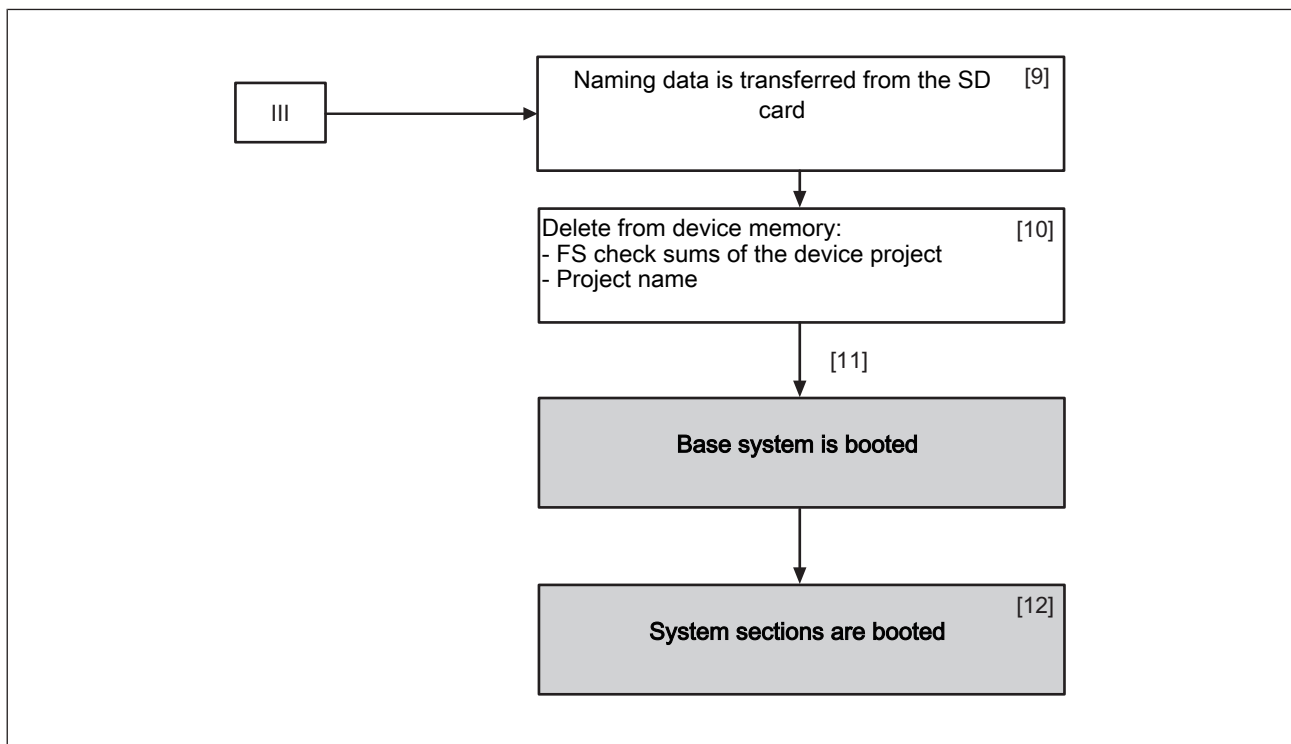
[7]

The device is restarted; the behaviour is the same as for a restart/reboot command (see Operating states and changes in operating state of a device).

[8]

If the SD card contains a device project that matches the product type, then in the operating state "Booting system sections", the device project is transferred from the SD card.

If the SD card does not contain a device project or contains a device project that does not match the product type, then the device starts without a device project. To ensure that the device contains a device project in this case, either a direct project download must be performed or the deliberate operator action must be performed again, using an SD card containing a device project that matches the product type.



[9]

If the naming data is not identical, then the naming data from the SD card is transferred to the device.

[10]

The check sum "FS device project" and the project name are deleted from the device's device memory.

[11]

The device is restarted; the behaviour is the same as for a restart/reboot command (see Operating states and changes in operating state of a device).

[12]

If the SD card contains a device project that matches the product type, then in the operating state "Booting system sections", the device project is transferred from the SD card. If the user program from the old device project contained non-volatile variables, the device will start up with the following non-volatile variable values:

Case	Variable with the attribute RETAIN PERSISTENT	Volatile variable [1]	Application parameters
Variable is available in the user program unchanged (resource assignment, FQN and data type are the same)	Saved value	Variable is set to initial value	Value configured for the application parameter
Variable in the user program is new	Variable is set to initial value		
Variable is no longer available in the user program	Variable is deleted		Application parameter is deleted

[1] If a non-retentive variable is used as the application parameter, then the "Application parameter" column applies.

If the SD card does not contain a device project or contains a device project that does not match the product type, then the device starts its system sections without a device project. To ensure that the device contains a device project, either a direct project download must be performed or the deliberate operator action must be performed again, using an SD card containing a corresponding device project.

12.13.3 Recovery mode

It is possible to switch to recovery mode in the event of a restart (switch on supply voltage) or when rebooting a device. In this case, the boot process has a scheduled interruption after the test phase and before the system components are booted (see [Operating status "Booting system sections" \[📖 418\]](#)).

No online operations are available in recovery mode.

Perform recovery mode

- ▶ Carry out either a restart or a reboot, while simultaneously holding the reset button pressed down for a while.

Note:

You can identify recovery mode via the status LED on the SD card:

- ▶ "SD Card" LED on a PSSu system: flashing green or lights up red
- ▶ "SD" LED on a PSS u2 system: flashing green or lights up red

If the reset button is not pressed down for long enough, the system does not switch to recovery mode. The device runs through all its usual boot phases.

Remove SD card

In recovery mode, the SD card can be removed without damaging any files on the SD card. There are two different scenarios:

- ▶ The SD card is in a consistent state (see [Journaling file system \[📖 613\]](#)):
 - "SD Card" LED on a PSSu system: flashing green.
 - "SD" LED on a PSS u2 system: flashing green.
- ▶ For some reason, the file system of the SD card cannot reconstruct a consistent state (see [Journaling file system \[📖 613\]](#)).
 - "SD Card" LED on a PSSu system: lights up red
 - "SD" LED on a PSS u2 system: lights up red

Exit recovery mode

- ▶ In recovery mode, press the reset button for > 10 s.
The device reboots.

12.13.4 Manual start of OPC UA commissioning mode

It is possible to switch to OPC UA commissioning mode in the event of a restart (switch on voltage supply) or when rebooting a device. In this case, the boot process has a scheduled interruption after the test phase and before the system components are booted (see [Operating status "Booting system sections" \[📖 418\]](#)).

Start commissioning mode manually

- ▶ Press the reset button twice within 3 s (press - release - press - release).



NOTICE

If you press and release the reset button only once within less than 3 seconds, a warm reset will be performed on the device (see [Perform a warm reset for a device \[📖 555\]](#)).

Extend commissioning mode

- ▶ If commissioning mode is active and the reset button is pressed twice again (press - release - press - release), the device remains in commissioning mode and the commissioning timeout is restarted.

You can configure the duration of the commissioning timeout in the Pilz NET Configurator.

Exit commissioning mode

- ▶ Commissioning mode is exited automatically when the commissioning timeout has expired.

12.14 Original reset

An original reset is always sensible or indeed necessary when a PSS 4000-exclusive device or PSS 4000-operable device is removed from an existing plant/machine, possibly to re-use it in another application.

If a PSS 4000-exclusive device/operable device is re-used in a project belonging to the automation system PSS 4000 following an original reset, a new device naming procedure will need to be carried out.

12.14.1 Effects of an original reset

An original reset has the following general effects on a PSS 4000-exclusive device or a PSS 4000-operable device:

- ▶ All the project-related data and working directories stored on the device will be deleted. This includes:
 - The project name
 - with a device for safety-related applications: the check sum "FS device project"
 - with a device with non-volatile memory: non-volatile variables
 - ▶ The device name is reset to the default device name (see [Naming data](#)  390]).
 - ▶ In devices with SD card, all the project-related data and working directories stored on the SD card are deleted. This includes:
 - The device project (includes the configuration of all related system sections)
 - with a device for safety-related applications: the check sum "FS device project"
- All data and working directories not from the system are also deleted. This includes:
- All user data that was saved on the SD card by the user including the /userfiles folder with all the directories and data contained within
 - If present, the /ep folder with the backup copy
- ▶ The messages on the device diagnostic list changes to "inactive" state. This change of state is logged in the device diagnostic log, but the change of state message may be missing for some diagnostic messages.
 - ▶ The device diagnostic log is retained. A log entry is added, which provides information about the execution of the original reset.
 - ▶ The current version of the PSS 4000 firmware remains on the device.
 - ▶ After an original reset the PSS 4000 firmware restarts on the device. The effect is as in a reboot command.

In addition to these general effects there also device-specific effects for

- ▶ PSS 4000-exclusive devices
- ▶ PSS 4000-operable devices

12.14.1.1 Effects on a PSS 4000-exclusive device

An original reset has the following device-specific effects on a PSS 4000-exclusive device:

- ▶ All outputs on the device are set to ZERO.
- ▶ Process data that is accessed by other devices is set to its substitute values in the other devices, valid bits = FALSE (see [Validity of process data](#) [ 359]).
- ▶ The device resets the IP address to the default setting and activates its Auto-IP mechanism (see [Auto-IP mechanism](#) [ 130]).
- ▶ In the case of a device with SD card, where the subdirectory /cfg contains a network configuration file (net.json), the network configuration file is deleted.
- ▶ Device with the system section PROFIBUS-DP Slave:
If the DIP switch on the device is set to 126_D, the device's station address is reset to 126_D.

12.14.1.2 Effects on a PSS 4000-operable device

The original reset has the following device-specific effects on a PSS 4000-operable device:

- ▶ An original reset only has an effect on the PSS 4000 firmware of a device.
- ▶ The device's network settings (IP address etc.) remain unchanged, i.e. the original reset does not influence the device's network settings (see [Naming data](#)  390).

12.14.2 Perform original reset

An original reset is performed in PAS4000.

12.15 Enable principle

With the enable principle, an ST data source can control an FS hardware output, provided there is an enable from an FS data source. In the event of an error, the FS data source must ensure that the FS hardware output controlled by the ST data source can no longer be set.

- ▶ Valid ST data sources are:
 - ST-O-PI variables
 - ST-I data from external communication (e.g. Modbus/TCP, fieldbuses)
- ▶ Valid FS data sources are:
 - FS-O-PI variables
 - FS-I data from external communication (e.g. FS bus system of another manufacturer)

Principle

In I/O mapping in PAS4000, the user maps an ST data source and an FS data source to the same FS data sink (e.g. FS hardware output). PSS 4000 performs the evaluation internally, based on this I/O mapping. This means that the evaluation is not performed in a PSSu system's resource, so no programming is required in the user program. The ST data source, the valid bit from the ST data source, the FS data source and the valid bit from the FS data source are logically linked and evaluated during the internal evaluation. The result of the logic operation controls the FS hardware output.

The following rules apply:

- ▶ Provided the FS data source and the valid bit from the FS data source are "TRUE", the FS hardware output can be switched on and off with the ST data source. The enable LED on the FS hardware output will light.
- ▶ As soon as the FS data source and/or the valid bit from the FS data source is "FALSE", the substitute value is used for the FS hardware output. With a digital FS hardware output this means that it is shut down safely. The enable LED on the FS hardware output will not be lit. Control via the ST data source is ignored.
- ▶ As soon as the valid bit from the ST data source is "FALSE", the substitute value is used for the FS hardware output. With a digital FS hardware output this means that it is shut down safely. The enable LED on the FS hardware output will light.

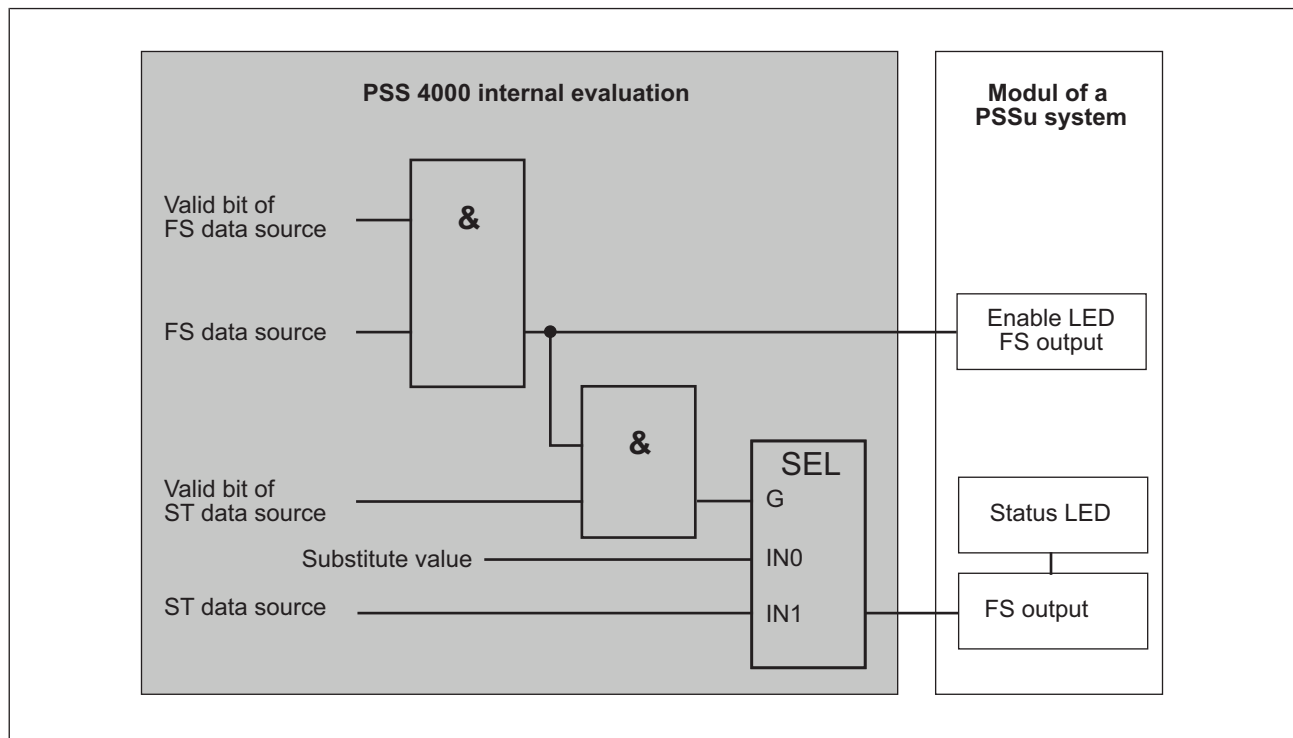


Fig.: Internal evaluation using a PSSu system as an example (principle)

Please note:

ST data that is processed on the ST resource is **not** safe. It may only be an **additional** criterion for a safety-related action (e.g. start/stop) and must **not** prevent safety functions. For example, if for whatever reason a stop signal has not been activated in the ST resource, or a start signal is continuously activated, this must not be able to compromise the safety of the machine or process.

12.16 Reaction times

The procedure when calculating reaction times depends on the PSS 4000-exclusive device you are using:

- ▶ [Reaction times for PSSu systems and PSS67 devices !\[\]\(687b6c142f51ac6f390f8bd444e38d03_img.jpg\) 571](#)
- ▶ [Reaction times for PSS u2 systems !\[\]\(861b7aaa71df51b93037a486c3b17630_img.jpg\) 587](#)

12.16.1 Reaction times for PSSu systems and PSS67 devices

The following description of the reaction times refers exclusively to the inputs and outputs on 1st generation PSS 4000-exclusive devices. If the inputs and outputs are distributed over various PSS 4000-exclusive devices, the processing time of the SafetyNET p interfaces and the SafetyNET p transmission time must also be taken into account.

If necessary, the relevant external communication relationships (e.g. communication via Modbus/TCP, UDP, PROFIBUS-DP), third party periphery devices, sensors/actuators and third party controllers must also be taken into account when determining the overall reaction times. These influences are not described here.



INFORMATION

The following description of reaction times is valid for PAS4000 from Version 1.2.0

The "reaction time" is the time that elapses between a defined event and a defined reaction that is triggered as a result.

12.16.1.1 Times that influence reaction times

The reaction times are determined by the following times:

Cycle time of a periodic task (t_{Task})

The task cycle time t_{Task} is defined by the user during the task configuration in PAS4000. It defines the time within which a task must be executed (see [Task basics](#)  364]).

Value range:

- ▶ PSS 4000-exclusive devices with the exception of head modules PSSu H FS2 SN SD(-T) and PSSu H PLC2 FS SN SD(-T):
 - ST tasks: 2 ms ... 2 000 000 ms
 - FS tasks: 6 ms ... 10 000 ms
- ▶ Head modules PSSu H FS2 SN SD(-T) and PSSu H PLC2 FS SN SD(-T):
 - ST tasks: 2 ms ... 1 000 ms
 - FS tasks: 6 ms ... 1 000 ms

The value range for the task cycle time t_{Task} depends on the device.

Execution time for a task (t_{ProcTask})

The execution time t_{ProcTask} is the time between the task cycle starting and the task execution ending, i.e. the process image of outputs and the resource global variables becoming available (see [Composition of data paths – Data paths with tasks](#)  579]).

Cycle time of module bus (t_{MBUS})

There is one cycle time each for the FS module bus and the ST module bus of a PSS 4000-exclusive device. They are calculated and displayed by PAS4000. The cycle time t_{MBUS} is the time needed to transport data between the head module and module.

Value range:

- ▶ ST module bus: 2 ms ... 45 ms
- ▶ FS module bus: 6 ms ... 30 ms

Please note:

- ▶ The cycle time of the respective module bus always corresponds to that of the task with the shortest cycle time, provided the task cycle time is less than/equal to the maximum possible cycle time of the corresponding module bus:

$$t_{\text{MBUS}} = t_{\text{Task min}}$$

- ▶ If only one task is executed on a PSS 4000-exclusive device, the module bus cycle time will correspond to the task's cycle time, provided the task cycle time is less than/equal to the maximum possible cycle time of the corresponding module bus:

$$t_{\text{MBUS}} = t_{\text{Task}}$$

Processing time for a module's input data (t_{ProcIM})

Processing time t_{ProcIM} : See technical details in the module's operating manual

The processing time t_{ProcIM} is the time that elapses on a module between the signal physically changing on the input and the I-datum being ready on the module bus. The processing time takes into account various influences such as internal runtimes, temperature drifts, spread of components, etc. It varies depending on the module.

If a filter time can be configured for the module then the following applies:

$$t_{\text{Proclm}} = t_{\text{ProclM}} + \text{configured filter time}$$

Processing time for a module's output data (t_{ProcOM})

Processing time t_{ProcOM} : See technical details in the module's operating manual

The processing time t_{ProcOM} is the time that a module needs internally to process O-data.

The processing time takes into account various influences such as internal runtimes, temperature drifts, spread of components, etc. It varies depending on the module.

Basic cycle time of SafetyNET p (t_{BasicSNp})

The basic cycle time t_{BasicSNp} describes the time it takes to run tests within a project as part of each cycle, to check whether data is available to send via SafetyNET p. The basic cycle time is calculated by PAS4000 and displayed during the build process.

► Project-internal SNp communication

The cycle times of the FS and ST system sections involved in SNp communication are taken into account when calculating the basic cycle time. The FS heartbeat cycle time $t_{\text{SNp FS-HBT}}$ is also taken into account, depending on the configuration (see [FS heartbeat \(FS-HBT\)](#) [190]). So the cycle times taken into account include t_{Task} , t_{MBUS} , t_{ExtCo} and $t_{\text{SNp FS-HBT}}$.

HBT.

The basic cycle time corresponds to the greatest common factor of the cycle times of the FS and ST system sections involved in SNp communication and, depending on the configuration, to the manually configured value for the FS heartbeat cycle time.

Example:

t_{BasicSNp} = Greatest common factor (t_{Task} , t_{MBUS} , $t_{\text{SNp FS-HBT}}$) in ms

- $t_{\text{Task}} = 10 \text{ ms}$
- $t_{\text{MBUS}} = 5 \text{ ms}$
- $t_{\text{SNp FS-HBT}}$ is 120 ms, for example

Result: $t_{\text{BasicSNp}} = 5 \text{ ms}$

► Cross-project SNp communication (PASconnect interface version 1.2.x)

The cycle times t_{Task} of the FS tasks and ST tasks involved in SNp communication are taken into account when calculating the basic cycle time.

The basic cycle time corresponds to the greatest common factor of the task cycle times (t_{Task}) of the tasks involved in SNp communication.

Example:

t_{BasicSNp} = Greatest common factor ($t_{\text{Task 1}}$, $t_{\text{Task 2}}$) in ms

- $t_{\text{Task 1}} = 10 \text{ ms}$
- $t_{\text{Task 2}} = 5 \text{ ms}$

Result: $t_{\text{BasicSNp}} = 5 \text{ ms}$

► Cross-project SNp communication (PASconnect interface version 1.3.x/1.4.x)

The cycle times t_{Task} of the FS tasks and ST tasks involved in SNp communication are taken into account when calculating the basic cycle time. The FS heartbeat cycle time $t_{\text{SNp FS-HBT}}$, configured manually in PAS4000, is also taken into account (see [FS heartbeat \(FS-HBT\)](#) [190]).

The basic cycle time corresponds to the greatest common factor of the task cycle times involved in SNp communication and to the configured value for the FS heartbeat cycle time.

Example:

t_{BasicSNp} = Greatest common factor ($t_{\text{Task 1}}$, $t_{\text{Task 2}}$, $t_{\text{SNpFS-HBT}}$) in ms

– $t_{\text{Task 1}} = 10 \text{ ms}$

– $t_{\text{Task 2}} = 10 \text{ ms}$

– $t_{\text{SNpFS-HBT}} = 110 \text{ ms}$

Result: $t_{\text{BasicSNp}} = 10 \text{ ms}$

SafetyNET p cycle time (t_{SNp})

With the cycle time t_{SNp} , a distinction is made in accordance with the following criteria:

- Cycle time with project-internal SNp communication and cycle time with cross-project SNp communication (PASconnect interface version 1.2.x)

The cycle time t_{SNp} refers to a transmit connection (Tx connection) of a PSS 4000 device; i.e. each Tx connection of a PSS 4000 device has its own cycle time t_{SNp} .

The cycle time describes the time in which a Tx connection is started cyclically in order to send data. The cycle time is an integer multiple of the basic cycle time t_{BasicSNp} . This is the basic cycle time of the project containing the transmitting PSS 4000 device.

- Cycle time with cross-project SNp communication (PASconnect interface version 1.3.x/1.4.x)

The cycle time t_{SNp} refers to a receive connection (Rx connection) of a PSS 4000 device; i.e. each Rx connection of a PSS 4000 device has its own cycle time t_{SNp} .

The cycle time describes the time in which the receiving PSS 4000 device cyclically checks whether input data from an Rx connection is available for further processing. The cycle time is an integer multiple of the basic cycle time t_{BasicSNp} . This is the basic cycle time of the project containing the receiving PSS 4000 device.

If necessary to make a clear distinction, the cycle time t_{SNp} is described as follows:

SNp communication	PASconnect interface version	Description of the cycle time
Project-internal	- - -	$t_{\text{SNp}} (\text{Tx})$
Cross-project	1.2.x	$t_{\text{SNp}} (\text{Tx})$
	1.3.x/1.4.x	$t_{\text{SNp}} (\text{Rx})$

A connection's cycle time is calculated by PAS4000 and displayed during the build process.

Maximum cycle time:

- ST SafetyNET p: Depends on the device
- FS SafetyNET p: Depends on the device

Maximum transmission delay time of SafetyNET p (t_{DelaySNp})

Within a project, the transmission delay time is regularly measured using the FS heartbeat (see [FS heartbeat \(FS-HBT\)](#) [190]). The time measured is the time it takes for data packets to be available in a device B once data transmission starts in a device A. The

measured time may not exceed the configured maximum transmission delay time t_{DelaySNp} . If the measured time exceeds the maximum transmission delay time, all FS receive connections (FS-Rx connections) received from device A are deactivated on device B (see Cyclical processing of connections). The substitute values are used for the receive data, with valid bits = FALSE.

Monitoring of the transmission delay time is independent of the cycle times of the SNp connections between the individual PSS 4000 devices.

- Project-internal SNp communication and cross-project SNp communication (PASconnect interface version 1.3.x/1.4.x)

The maximum transmission delay time must be configured manually in PAS4000.
Value range: 4 ms ... 2000 ms (default setting: 25 ms)

Note:

- The maximum transmission delay time should be kept as low as possible as it influences the maximum response time of SafetyNET p data subpaths (see [Data subpaths with SafetyNET p as data source](#) [582]). The default setting is an appropriate value for many applications.
- Problems with the transmission delay time may occur when
 - there are problems in the Ethernet infrastructure (e.g. switches, routers, cables)
 - the devices have a high number of SNp connections
 - the devices' system load is high, such as on PSS 4000 devices with many electronic modules or a large user program, for example

The Ethernet infrastructure should be checked first, before extending the transmission delay time.
- With cross-project SNp communication, the transmission delay time t_{DelaySNp} of the project containing the data receiver must be used to calculate the response time of SafetyNET p data subpaths.

- Cross-project SNp communication (PASconnect interface version 1.2.x)

The transmission delay time is calculated by PAS4000 and displayed during the build process.

The transmission delay time t_{DelaySNp} of the project containing the data receiver must be used to calculate the response time of SafetyNET p data subpaths (see [Data subpaths with SafetyNET p as data source](#) [582]).

Cycle time with external communication (t_{ExtCo})

External communication refers to data exchange in which external O-data leaves the PSS 4000 project and external I-data enters the PSS 4000 project (e.g. via Modbus/TCP, TCP, UDP, fieldbuses). Depending on the communication mode that is used, external O-data is sent from the O-data area of a PSS 4000-exclusive device. Depending on the communication mode that is used, external I-data can be found in the I-data area of a PSS 4000-exclusive device and is read from there.

In this description, the cycle times for send and receive are designated with t_{ExtCo} . Their meaning depends on the communication mode that is used:

- **Modbus/TCP Client connection**

The cycle time t_{ExtCo} describes the time the connection Client takes to start data transfer as part of each cycle.

The cycle time can be set separately in PAS4000 for each connection Client on a PSS 4000-exclusive device; in other words, each connection Client on a PSS 4000-exclusive device has its own cycle time t_{ExtCo} . It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

► **Modbus/TCP Server connection**

The cycle time t_{ExtCo} describes the time it takes to read data from a Server allocation table of a PSS 4000-exclusive device and for the data requested from a Server allocation table to be transferred to the connection Client, as part of each cycle.

The cycle time can be set separately in PAS4000 for each server allocation table on a PSS 4000-exclusive device; in other words, each server allocation table on a PSS 4000-exclusive device has its own cycle time t_{ExtCo} . It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

► **Raw TCP communication "Receive data" and "Send data"**

The cycle time t_{ExtCo} describes the time it takes to read in external I-data from a TCP connection and start transmitting data (external O-data) via a TCP connection as part of each cycle.

The cycle time can be set separately in PAS4000 for each TCP connection on a PSS 4000-exclusive device; in other words, each TCP connection has its own cycle time t_{ExtCo} . It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

► **Raw UDP communication "Receive data" and "Send data"**

The cycle time t_{ExtCo} describes the time it takes to cyclically read in external I-data from a UDP connection and to cyclically start transmitting data (external O-data) via an UDP connection as part of each cycle.

The cycle time can be set separately in PAS4000 for each UDP connection on a PSS 4000-exclusive device; in other words, each UDP connection has its own cycle time t_{ExtCo} . It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

► **PROFIBUS-DP Slave "Receive data" and "Send data":**

The cycle time t_{ExtCo} describes the time it takes to read in external I-data from the PROFIBUS-DP Master and start transmitting the requested data (external O-data) as part of each cycle.

The cycle time can be defined in PAS4000 for each "PROFIBUS-DP Slave" system section, i.e. each "PROFIBUS-DP Slave" system section has its own cycle time t_{ExtCo} for data transfer with PROFIBUS-DP. It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

► **PROFIBUS-DP Slave with PROFI-safe "Receive data" and "Send data"**

The cycle time t_{ExtCo} describes the time it takes to read in external I-data from the PROFIBUS-DP Master/F-host and start transmitting the requested data (external O-data) as part of each cycle.

The cycle time can be defined in PAS4000 for each "PROFIBUS-DP Slave with PROFI-safe" system section, i.e. each "PROFIBUS-DP Slave with PROFI-safe" system section has its own cycle time t_{ExtCo} for data transfer. The cycle time t_{ExtCo} applies to data transfer via both PROFIBUS-DP and PROFI-safe. It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

Please note:

If PROFI-safe communication is used, only a limited value range will be available for the cycle time t_{ExtCo} . In this case, this limited value range will also apply to PROFIBUS-DP Slave communication. The cycle time t_{ExtCo} depends on the device.

► **EtherNet/IP Adapter "Receive data" and "Send data"**

The cycle time t_{ExtCo} describes the time it takes to read in external I-data from the Scanner and start transmitting external O-data sent to the Scanner, as part of each cycle.

The cycle time can be determined in PAS4000 for the "Ethernet/IP Adapter" system section. It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

► **PROFINET "Receive data" and "Send data":**

The cycle time t_{ExtCo} describes the time it takes to read in external I-data from the PROFINET IO controller and start transmitting the requested data (external O-data) as part of each cycle.

The cycle time can be defined in PAS4000 for each "PROFINET IO DEVICE" system section, i.e. each "PROFINET IO DEVICE" system section has its own cycle time t_{ExtCo} for data transfer with PROFINET. It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

12.16.1.2 Processing rules during data processing

The following applies for data processing on a 1st generation PSS 4000-exclusive device:

► **Processing times of modules**

The processing times of modules (t_{ProcOM} and t_{ProcIM}) are constant. They are not influenced by the current task cycle.

► **Module bus cycle times**

The ST module bus and the FS module bus each have their individual cycle time. The cycle time of the respective module bus t_{MBUS} always corresponds to that of the task with the shortest cycle time, provided the task cycle time is less than/equal to the maximum possible cycle time of the relevant module bus:

$$t_{\text{MBUS}} = t_{\text{Task min}}$$

► **Task cycle times**

The task cycle times (t_{Task}) are constant. They are not influenced by the current task cycle.

12.16.1.3 Formation of data paths

Data subpaths are available to enable you to track a single I/O datum over a complete data path. A data path, therefore, is made up of the relevant data subpaths.

The response time of a data path is calculated by adding the response times of the relevant data subpaths. A data path always starts with the I-datum in a data source and ends with the O-datum in a data sink. A data source/data sink is always located in a system section (see [System sections](#) [22]).

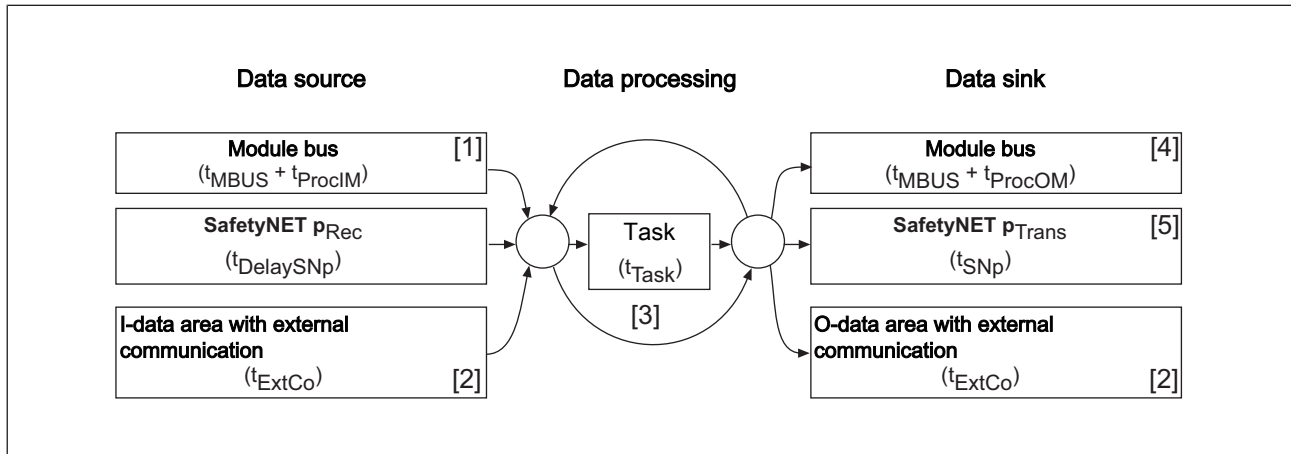


Fig.: Data paths on a PSS 4000-exclusive device

[1]

If the data source is an I-datum from a module, then the processing time for the I-datum on the module (t_{ProcIM}) needs to be considered in addition to the module bus cycle time (t_{MBUS}).

[2]

With external communication, the I-data from the I-data area is also referred to as "external I-data" and the O-data from the O-data area as "external O-data".

[3]

The following data paths are impossible:

- ▶ Data source: Module bus (I-datum) -> Data sink: module bus (O-datum)
- ▶ Data source: SafetyNET pRec -> Data sink: SafetyNET pTrans

[4]

If the data sink is an O-datum on a module, then the processing time for the O-datum on the module (t_{ProcOM}) needs to be considered in addition to the module bus cycle time (t_{MBUS}).

[5]

If several tasks with several SNp connections are used in a project, the cycle times t_{SNp} for these SNp connections may vary. Care must be taken to ensure that the correct cycle time t_{SNp} is used to calculate the response time of a data subpath. The calculation can be simplified by using the longest cycle time t_{SNp} for all of a device's data subpaths Task -> SafetyNET pTrans. In this case, the response time is overestimated.

Data paths with tasks

For data paths with tasks, please note:

► Task as data source

The response time for a datum starts at the beginning of the task cycle and ends after the execution time t_{ProcTask} with the process image of outputs and the resource global variables available in the data sink for further processing.

When the execution time t_{ProcTask} almost corresponds to the task cycle time t_{Task} , that is the maximum response time.

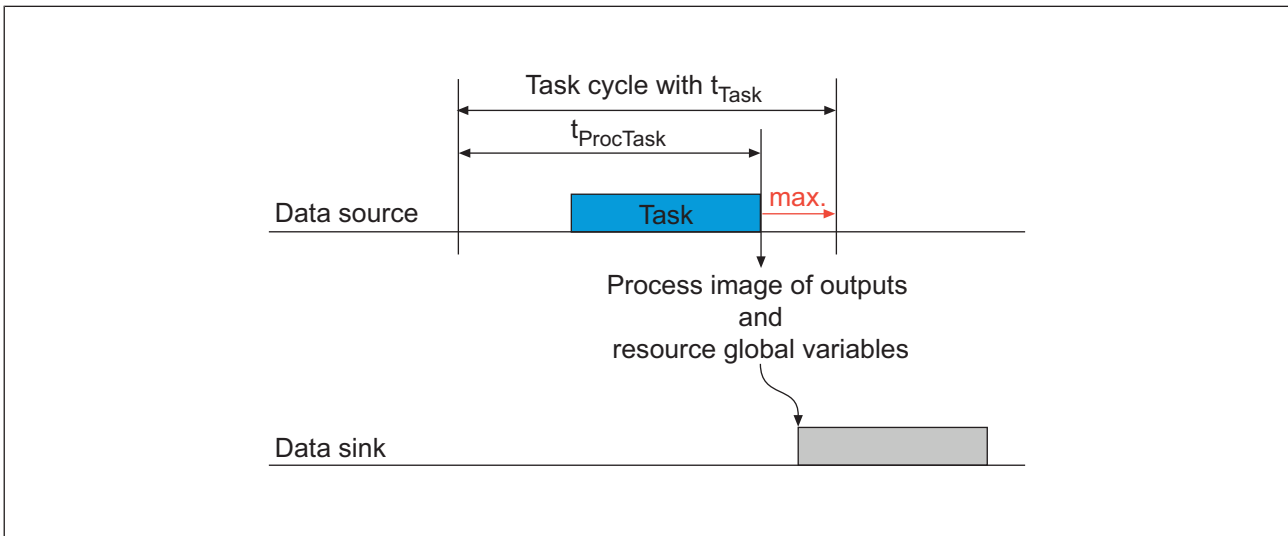


Fig.: Response time with a task's POU as data source

► Task as data sink

The response time for a datum starts at the beginning of the task cycle and ends after the execution time t_{ProcTask} with the process image of outputs and the resource global variables available for further processing.

When the execution time t_{ProcTask} almost corresponds to the task cycle time t_{Task} , that is the maximum response time.

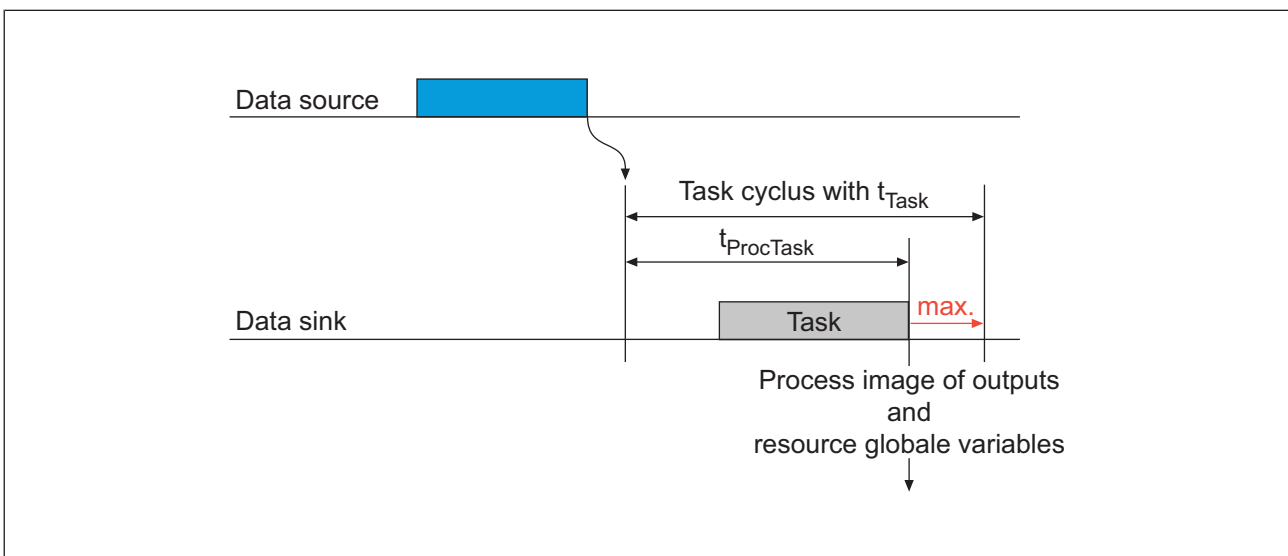


Fig.: Response time with a task as data sink

12.16.1.4 Maximum reaction times for data subpaths

The following description contains the formulas for calculating the maximum reaction time for all permitted data subpaths.

12.16.1.4.1 Maximum reaction time for data subpaths with module bus as data source**Module bus -> Task**

PSS 4000 firmware from Version 1.2.0 to 1.3.x:

$$t_{\text{MBUS\_Task\_max}} = t_{\text{ProclM}} + 2t_{\text{Task}} + t_{\text{MBUS}}$$

PSS 4000 firmware from version 1.4.0:

$$t_{\text{MBUS\_Task\_max}} = t_{\text{ProclM}} + 2t_{\text{Task}}$$

$t_{\text{MBUS\_Task\_max}}$	Max. reaction time for this data subpath
t_{ProclM}	Processing time for a module's I-data
t_{MBUS}	Module bus cycle time of the device
t_{Task}	Task cycle time of the task on the device

Module bus -> SafetyNET p_{Trans}

$$t_{\text{MBUS\_SNp\_max}} = t_{\text{ProclM}} + 2t_{\text{SNp}}$$

$t_{\text{MBUS\_SNp\_max}}$	Max. reaction time for this data subpath
t_{ProclM}	Processing time for a module's I-data
t_{MBUS}	Module bus cycle time of the device
t_{SNp}	Cycle time for the cyclical start of the SNp Tx connection (send data)

Module bus -> O-data area with external communication

$$t_{\text{MBUS\_ExtCo\_max}} = t_{\text{ProclM}} + 2t_{\text{ExtCo}}$$

$t_{\text{MBUS\_ExtCo\_max}}$	Max. reaction time for this data subpath
t_{ProclM}	Processing time for a module's I-data
t_{MBUS}	Module bus cycle time of the device
t_{ExtCo}	Cycle time for the cyclical start of data transfer from the O-data area with external communication

12.16.1.4.2 Data subpaths with SafetyNET p as data source

SafetyNET p_{Rec} -> Task

Project-internal SNp communication and
cross-project SNp communication (PASconnect interface version 1.2.x):

$$t_{\text{SNp\_Task\_max}} = t_{\text{DelaySNp}} + 2t_{\text{Task}}$$

$t_{\text{SNp\_Task\_max}}$	Max. reaction time for this data subpath
t_{DelaySNp}	► Project-internal SNp communication: Transmission delay time of SafetyNET p in a project ► Cross-project SNp communication: Transmission delay time of SafetyNET p on the project containing the receiver
t_{Task}	Task cycle time of the task on the device

Cross-project SNp communication (PASconnect interface version 1.3.x/1.4.x):

$$t_{\text{SNp\_Task\_max}} = t_{\text{DelaySNp}} + \max(t_{\text{SNp}}, t_{\text{Task}}) + t_{\text{Task}}$$

$t_{\text{SNp\_Task\_max}}$	Max. reaction time for this data subpath
t_{DelaySNp}	Transmission delay time of SafetyNET p on the project containing the receiver
t_{SNp}	Receiver's cycle time $t_{\text{SNp (Rx)}}$ for the cyclical test for the presence of receive data from the SNp Rx connection
t_{Task}	Task cycle time of the task on the device

SafetyNET p_{Rec} -> Module bus

PSS 4000 firmware from Version 1.2.0 to 1.3.x:

$$t_{\text{SNp\_MBUS\_max}} = t_{\text{DelaySNp}} + 3t_{\text{MBUS}} + t_{\text{ProcOM}}$$

PSS 4000 firmware from Version 1.4.0:

$$t_{\text{SNp\_MBUS\_max}} = t_{\text{DelaySNp}} + 2t_{\text{MBUS}} + t_{\text{ProcOM}}$$

$t_{\text{SNp\_MBUS\_max}}$	Max. reaction time for this data subpath
t_{DelaySNp}	Transmission delay time of SafetyNET p in a project
t_{MBUS}	Module bus cycle time of the device
t_{ProcOM}	Processing time for O-data on a module with outputs

SafetyNET p_{Rec} -> O-data area with external communication

$$t_{\text{SNp\_ExtCo\_max}} = t_{\text{Delay}} + 2t_{\text{ExtCo}}$$

$t_{\text{SNp\_ExtCo\_max}}$	Max. reaction time for this data subpath
t_{DelaySNp}	Transmission delay time of SafetyNET p in a project
t_{ExtCo}	Cycle time for the cyclical start of data transfer from the O-data area with external communication

12.16.1.4.3 Data subpaths with external I-data as data source

I-data area with external communication -> Module bus

$$t_{\text{ExtCo\_MBUS\_max}} = t_{\text{ExtCo}} + \max(t_{\text{ExtCo}}, t_{\text{MBUS}}) + t_{\text{MBUS}} + t_{\text{ProcOM}}$$

$t_{\text{ExtCo\_MBUS\_max}}$	Max. reaction time for this data subpath
t_{ExtCo}	Cycle time for cyclical reading of the external I-data from the I-data area with external communication
t_{MBUS}	Module bus cycle time of the device
t_{ProcOM}	Processing time for O-data on a module
$\max(t_{\text{ExtCo}}, t_{\text{MBUS}})$	The greater of the two values should be used

I-data area with external communication -> Task

$$t_{\text{ExtCo\_Task\_max}} = 2t_{\text{ExtCo}} + 2t_{\text{Task}}$$

$t_{\text{ExtCo\_Task\_max}}$	Max. reaction time for this data subpath
t_{ExtCo}	Cycle time for cyclical reading of the external I-data from the I-data area
t_{Task}	Task cycle time of the task on the device

I-data area with external communication -> SafetyNET p_{Trans}

$$t_{\text{ExtCo\_SNp\_max}} = 2t_{\text{ExtCo}} + 2t_{\text{SNp}}$$

$t_{\text{ExtCo\_SNp\_max}}$	Max. reaction time for this data subpath
t_{ExtCo}	Cycle time for cyclical reading of the external I-data from the I-data area with external communication
t_{SNp}	Cycle time for the cyclical start of the SNp Tx connection (send data)

I-data area with external communication -> O-data area with external communication

$$t_{\text{ExtCo1\_ExtCo2\_max}} = 2t_{\text{ExtCo1}} + 2t_{\text{ExtCo2}}$$

$t_{\text{ExtCo1\_ExtCo2\_max}}$	Max. reaction time for this data subpath
t_{ExtCo1}	Cycle time for cyclical reading of the external I-data from the I-data area (external communication with communication mode 1)
t_{ExtCo2}	Cycle time for the cyclical start of data transfer from the O-data area (external communication with communication mode 2)

12.16.1.4.4 Data subpaths with a task as data source**Task -> Module bus**

PSS 4000-Firmware from version 1.2.0 to 1.3.x:

$$t_{\text{Task\_MBUS\_max}} = 2t_{\text{MBUS}} + t_{\text{ProcOM}}$$

PSS 4000 firmware from version 1.4.0:

$$t_{\text{Task\_MBUS\_max}} = t_{\text{MBUS}} + t_{\text{ProcOM}}$$

$t_{\text{Task\_MBUS\_max}}$	Max. reaction time for this data subpath
t_{MBUS}	Module bus cycle time of the device
t_{ProcOM}	Processing time for O-data on a module

Task -> SafetyNET p_{Trans}

$$t_{\text{Task\_SNp\_max}} = 2t_{\text{SNp}}$$

$t_{\text{Task\_SNp\_max}}$	Max. reaction time for this data subpath
t_{SNp}	Transmitter's cycle time $t_{\text{SNp (Tx)}}$ for the cyclical start of the SNp Tx connection (send data)

Task1 -> Task2

$$t_{\text{Task1\_Task2\_max}} = 2t_{\text{Task2}}$$

$t_{\text{Task1\_Task2\_max}}$	Max. reaction time for this data subpath
t_{Task2}	Task cycle time of task2

Task -> O-data area with external communication

$$t_{\text{Task\_ExtCo\_max}} = 2t_{\text{ExtCo}}$$

$t_{\text{Task\_ExtCo\_max}}$	Max. reaction time for this data subpath
t_{ExtCo}	Cycle time for the cyclical start of data transfer from the O-data area with external communication

12.16.1.4.5 Example for calculating a data path on one device

The example was created for a PSS 4000-exclusive device with a PSS 4000 firmware from version 1.4.0.

For the data path in this example, the following applies:

- ▶ I-datum (e.g. HW input of a module) and O-datum (e.g. HW output of a module) are on the same device.
- ▶ The data path is composed as follows:
Module bus (I-datum) -> Task -> Module bus (O-datum)
- ▶ To calculate the maximum reaction time, the data path is divided into the following data subpaths:

- **Data subpath 1: Module bus (I-datum) -> Task**

The following applies for the maximum reaction time of data subpath 1

$$t_{\text{MBUS\_Task\_max}} = t_{\text{ProcIM}} + 2t_{\text{Task}}$$

- **Data subpath 2: Task -> Module bus (O-datum)**

The following applies for the maximum reaction time of data subpath 2:

$$t_{\text{Task\_MBUS\_max}} = t_{\text{MBUS}} + t_{\text{ProcOM}}$$

The following applies for the data path's maximum reaction time:

$$t_{\text{React\_max}} = t_{\text{ProcIM}} + 2t_{\text{Task}} + t_{\text{MBUS}} + t_{\text{ProcOM}}$$

12.16.1.4.6 Example for calculating a data path via three devices

The example was created for a PSS 4000-exclusive device with a PSS 4000 firmware from version 1.4.0.

For the data path in this example, the following applies:

- ▶ The I-datum (e. g HW input) is on device A. The datum is transferred to device B via SafetyNET p.
- ▶ On device B, the datum is processed in a task and is then transferred to device C via SafetyNET p.
- ▶ On device C, the datum is transferred to a module with outputs.
- ▶ The data path is composed as follows:

Module bus (I-datum) on device A -> SafetyNET p -> Task on device B -> SafetyNET p -> Module bus (O-datum) on device C

- ▶ To calculate the maximum reaction time, the data path is divided into the following data subpaths:

– **Data subpath 1: Module bus_(A) (I-datum) -> SafetyNET p_{Trans(A)}**

The following applies for the maximum reaction time of data subpath 1:

$$t_{\text{MBUS\_SNp\_max(A)}} = t_{\text{ProclM}} + 2t_{\text{SNp(A)}}$$

– **Data subpath 2: SafetyNET p_{Rec(B)} -> Task_(B)**

The following applies for the maximum reaction time of data subpath 2:

$$t_{\text{SNp\_Task\_max(B)}} = t_{\text{DelaySNp}} + 2t_{\text{Task(B)}}$$

– **Data subpath 3: Task_(B) -> SafetyNET p_{Trans(B)}**

The following applies for the maximum reaction time of data subpath 3:

$$t_{\text{Task\_SNp\_max(B)}} = 2t_{\text{SNp(B)}}$$

– **Data subpath 4: SafetyNET p_{Rec(C)} -> Module bus_(C) (O-datum)**

The following applies for the maximum reaction time of data subpath 4:

$$t_{\text{SNp\_MBUS\_max(C)}} = t_{\text{DelaySNp}} + 2t_{\text{MBUS(C)}} + t_{\text{ProcOM}}$$

The following applies for the data path's maximum reaction time:

$t_{\text{React\_max}}$	$ \begin{aligned} &= t_{\text{ProclM}} + 2t_{\text{SNp(A)}} \\ &+ t_{\text{DelaySNp}} + 2t_{\text{Task(B)}} \\ &+ 2t_{\text{SNp(B)}} \\ &+ t_{\text{DelaySNp}} + 2t_{\text{MBUS(C)}} + t_{\text{ProcOM}} \end{aligned} $

12.16.2 Reaction times for PSS u2 systems

The following description of the reaction times refers exclusively to the inputs and outputs on 2nd generation PSS 4000-exclusive devices. If the inputs and outputs are distributed over various PSS 4000-exclusive devices, the processing time of the SafetyNET p interfaces and the SafetyNET p transmission time must also be taken into account.



INFORMATION

The following description of reaction times is valid for PAS4000 from Version 1.25.0

The "reaction time" is the time that elapses between a defined event and a defined reaction that is triggered as a result.

12.16.2.1 Times that influence reaction times

Cycle time of a periodic task (t_{Task})

The task cycle time t_{Task} is defined by the user during the task configuration in PAS4000. It defines the time within which a task must be executed (see Task basics).

Value range:

- ▶ ST tasks: 2 ms ... 1 000 ms
- ▶ FS tasks: 6 ms ... 1 000 ms

The value range for the task cycle time t_{Task} depends on the device.

The cycle time is displayed by PAS4000 during the build process.

Execution time for a task (t_{ProcTask})

The execution time t_{ProcTask} is the time between the task cycle starting and the task execution ending, i.e. the process image of outputs and the resource global variables becoming available (see [Formation of data paths \[593\]](#)). The maximum reaction time is when the execution time t_{ProcTask} almost corresponds to the task cycle time t_{Task} .

The execution time can be displayed in PAS4000.

Cycle time of module bus PSS u2 (t_{MBUS})

The FS module bus PSS u2 and ST module bus PSS u2 of a 2nd generation device each have a cycle time.

- ▶ The cycle time of the ST module bus is the time needed to transport I/O data between the head module and ST modules.
- ▶ The cycle time of the FS module bus is the time needed to transport I/O data between the head module and FS modules.

Value range:

- ▶ ST module bus: 2 ms ... 100 ms
- ▶ FS module bus: 6 ms ... 100 ms

The cycle time is calculated by PAS4000 and is displayed during the build process.

Note:

- ▶ FS module bus oversampling rate and ST module bus oversampling rate

The module bus cycle time can be shortened by configuring a module bus oversampling rate for the respective module bus in PAS4000. If a module bus oversampling rate $x > 1$ is configured, then the module bus cycle time is x -times shorter than the module bus cycle time calculated automatically.

The module bus cycle time and therefore the reaction time are reduced, but the system load increases and the availability of the device may fall.

Time difference between processing of the send and receive data on the FS module ($t_{\text{FSTxDelay}}$)

$t_{\text{FSTxDelay}}$ is the minimum time interval between processing of the receive data and processing of the send data of the FS module bus PSS u2 in the FS module.

The minimum time difference is calculated by PAS4000 and is displayed during the build process.

Time difference between redundant I/O data on the FS module bus PSS u2 ($t_{FSRxDelay}$)

$t_{FSRxDelay}$ is the minimum time interval between redundant transmissions of I/O data on the FS module bus PSS u2

The minimum time difference is calculated by PAS4000 and is displayed during the build process.

Processing time for a module's input data (t_{ProcIM})

The processing time t_{ProcIM} is the time that elapses on a module between the signal physically changing on the input and the I-datum being ready on the module bus PSS u2. The processing time takes into account various influences such as internal runtimes, temperature drifts, spread of components, etc. The processing time varies depending on the module.

A distinction is made between the following processing times:

► **Worst-case processing time t_{ProcIM_wcd}**

t_{ProcIM_wcd} is the processing time for a module's I-data, without a fault occurring.

► **One-fault processing time t_{ProcIM_ofd}**

t_{ProcIM_ofd} is the processing time for a module's I-data when one fault occurs.

If additional times can be configured for the module's inputs (e.g. software filter time, maximum duration of the test pulse), then these configured times must be taken into account.

Information on determining t_{ProcIM} can be found in the operating manual of the relevant module.

Processing time for a module's output data (t_{ProcOM})

The processing time t_{ProcOM} is the time a module requires internally to process the O-data. The processing time takes into account the various influences such as internal run times, temperature drifts, spread of components, etc. The processing time varies depending on the module.

A distinction is made between the following processing times:

► **Worst-case processing time t_{ProcOM_wcd}**

t_{ProcOM_wcd} is the processing time for a module's O-data, without a fault occurring.

► **One-fault processing time t_{ProcOM_ofd}**

t_{ProcOM_ofd} is the processing time for a module's O-data when one fault occurs.

If additional times can be configured for the module's outputs (e.g.

switch-off delay), then these configured times must be taken into account.

Information on determining t_{ProcOM} can be found in the operating manual of the relevant module.

Processing time for I/O-data on the head module (t_{ProcHM})

The processing time t_{ProcHM} is the maximum time a head module requires to process the receive data and the send data of the FS module bus PSS u2

The maximum processing time is calculated by PAS4000 and is displayed during the build process.

Basic cycle time of SafetyNET p (t_{BasicSNp})

The basic cycle time t_{BasicSNp} describes the time it takes to run tests within a project as part of each cycle, to check whether data is available to send via SafetyNET p (project-internal communication).

The cycle times of the FS and ST system sections involved in SNp communication are taken into account when calculating the basic cycle time. The FS heartbeat cycle time $t_{\text{SNp FS-HBT}}$ is also taken into account, depending on the configuration (see [FS heartbeat \(FS-HBT\)](#) [190]). The cycle times that are taken into account are therefore t_{Task} , t_{MBUS} , t_{ExtCo} and $t_{\text{SNp FS-HBT}}$.

The basic cycle time corresponds to the greatest common factor of the cycle times of the FS and ST system sections involved in SNp communication and, depending on the configuration, to the manually configured value for the FS heartbeat cycle time.

The basic cycle time is calculated by PAS4000 and is displayed during the build process.

Example:

t_{BasicSNp} = Greatest common factor (t_{Task} , t_{MBUS} , $t_{\text{SNp FS-HBT}}$) in ms

- ▶ $t_{\text{Task}} = 10 \text{ ms}$
- ▶ $t_{\text{MBUS}} = 5 \text{ ms}$
- ▶ $t_{\text{SNp FS-HBT}}$ is 120 ms, for example

Result: $t_{\text{BasicSNp}} = 5 \text{ ms}$

SafetyNET p cycle time (t_{SNp})

With the cycle time t_{SNp} , a distinction is made in accordance with the following criteria:

- ▶ Cycle time for project-internal SNp communication

The cycle time t_{SNp} refers to a transmit connection (Tx connection) on a device; i.e. each Tx connection on a device has its own cycle time t_{SNp} .

The cycle time describes the time it takes to start a Tx connection as part of each cycle, in order to send data. The cycle time is an integer multiple of the basic cycle time t_{BasicSNp} . This is the basic cycle time of the project containing the transmitting PSS 4000 device.

- ▶ Cycle time with cross-project SNp communication (PASconnect interface version 1.3.x/1.4.x)

The cycle time t_{SNp} refers to a receive connection (Rx connection) on a PSS 4000 device; i.e. each Rx connection on a PSS 4000 device has its own cycle time t_{SNp} .

The cycle time describes the time in which the receiving PSS 4000 device cyclically checks whether input data from an Rx connection is available for further processing. The cycle time is an integer multiple of the basic cycle time t_{BasicSNp} . This is the basic cycle time of the project containing the receiving PSS 4000 device.

If necessary to make a clear distinction, the cycle time t_{SNp} is described as follows:

SNp communication	PASconnect interface version	Description of the cycle time
Project-internal	- - -	t_{SNp} (Tx)
Cross-project	1.3.x/1.4.x	t_{SNp} (Rx)

The value range for the cycle time t_{SNp} depends on the device (see Technical details in the operating instructions for the respective device).

A connection's cycle time is calculated by PAS4000 and displayed during the build process.

Maximum cycle time:

- ▶ ST SafetyNET p: depends on the device
- ▶ FS SafetyNET p: depends on the device

Maximum transmission delay time of SafetyNET p (t_{DelaySNp})

Within a project (project-internal SNp communication), the transmission delay time is regularly measured using the FS heartbeat (see [FS heartbeat \(FS-HBT\)](#) [📖 190]). The time measured is the time it takes for data packets to be available in a device B once data transmission starts in a device A. The measured time may not exceed the configured maximum transmission delay time t_{DelaySNp} .

If the measured time exceeds the maximum transmission delay time, all FS receive connections (FS-Rx connections) received from device A are deactivated on device B (see [Cyclical processing of connections](#)). The substitute values are used for the receive data, with valid bits = FALSE.

Monitoring of the transmission delay time is independent of the cycle times of the SNp connections between the individual devices.

The maximum transmission delay time must be configured manually in PAS4000 and is displayed in PAS4000 during the build process.

Value range: 4 ms ... 2000 ms (default setting: 25 ms)

Please note:

- ▶ The maximum transmission delay time should be kept as low as possible as it influences the maximum reaction time of SafetyNET p data subpaths (see [Data subpaths with SafetyNET p as data source](#) [📖 599]). The default setting is an appropriate value for many applications.
- ▶ Problems with the transmission delay time may occur when
 - There are problems in the Ethernet infrastructure (e.g. switches, routers, cables)
 - The devices have a high number of SNp connections
 - The devices' system load is high, such as on devices with many electronic modules or a large user program, for example

The Ethernet infrastructure should be checked first, before extending the transmission delay time.

12.16.2.2 Processing rules during data processing

The following applies for data processing on a 2nd generation device:

- ▶ Processing times of modules

The processing times of modules (t_{ProcOM} und t_{ProcIM}) are constant. They are not influenced by the current task cycle.

- ▶ Module bus cycle times PSS u2

The ST module bus PSS u2 and the FS module bus PSS u2 each has its own cycle time t_{MBUS} . The cycle time of the respective module bus is always set in such a way that the I-data of the module bus PSS u2 provides all tasks in time and the O-data of the module bus PSS u2 is received by the tasks in time. That's why the cycle time t_{MBUS} is always a divisor of the cycle time of all tasks.

- ▶ Task cycle times

The task cycle times (t_{Task}) are constant. They are not influenced by the current task cycle.

12.16.2.3 Formation of data paths

Data subpaths are available to enable you to track a single I/O datum over a complete data path. A data path, therefore, is made up of the relevant data subpaths.

The reaction time of a data path is calculated by adding the reaction times of the relevant data subpaths. A data path always starts with the I-datum in a data source and ends with the O-datum in a data sink. A data source/data sink is always located in a system section (see [System sections](#) [22]).

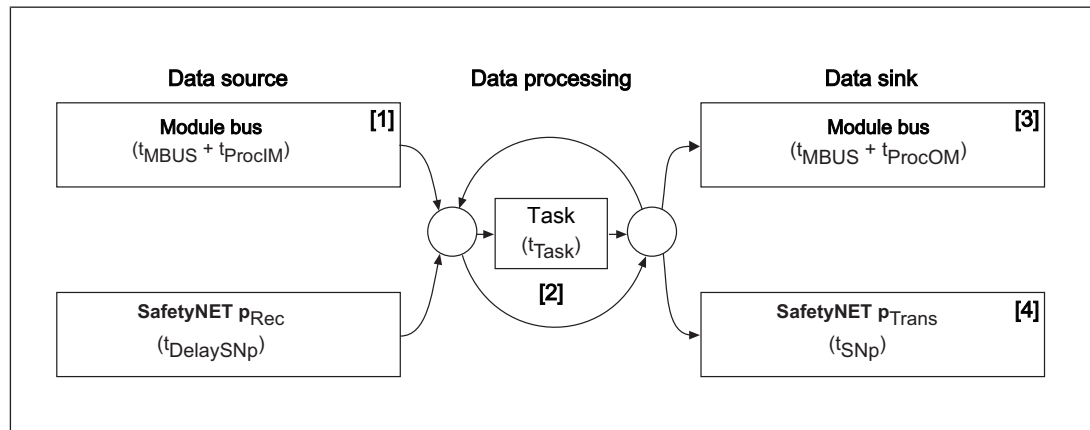


Fig.: Data paths on a 2nd generation PSS 4000-exclusive device

[1]

If the data source is an I-datum from a module, then besides the cycle time of the module bus (t_{MBUS}), the processing time for the I-datum on the module (t_{ProcIM}) must also be taken into account.

[2]

The following data paths are impossible:

- ▶ Data source: Module bus (I-datum) -> Data sink: module bus (O-Datum)
- ▶ Data source: SafetyNET pRec -> Data sink: SafetyNET pTrans

[3]

If the data sink is an O-datum on a module, then besides the cycle time of the module bus (t_{MBUS}), the processing time for the O-datum on the module (t_{ProcOM}) must also be taken into account.

[4]

If multiple tasks with multiple Snp connections are used in one project, the cycle times t_{SNp} of these Snp connections may vary. In this case, care must be taken to ensure that the correct cycle time t_{SNp} is used to calculate the reaction time of a data subpath. The calculation can be simplified by using the longest cycle time t_{SNp} for all of a device's data subpaths Task -> SafetyNET pTrans. In this case, the reaction time is overestimated.

Data paths with tasks

For data paths with tasks, please note:

▶ Task as data source

The reaction time for a datum starts at the beginning of the task cycle and ends after the execution time t_{ProcTask} with the process image of outputs and the resource global variables available in the data sink for further processing.

The maximum reaction time is when the execution time t_{ProcTask} almost corresponds to the task cycle time t_{Task} .

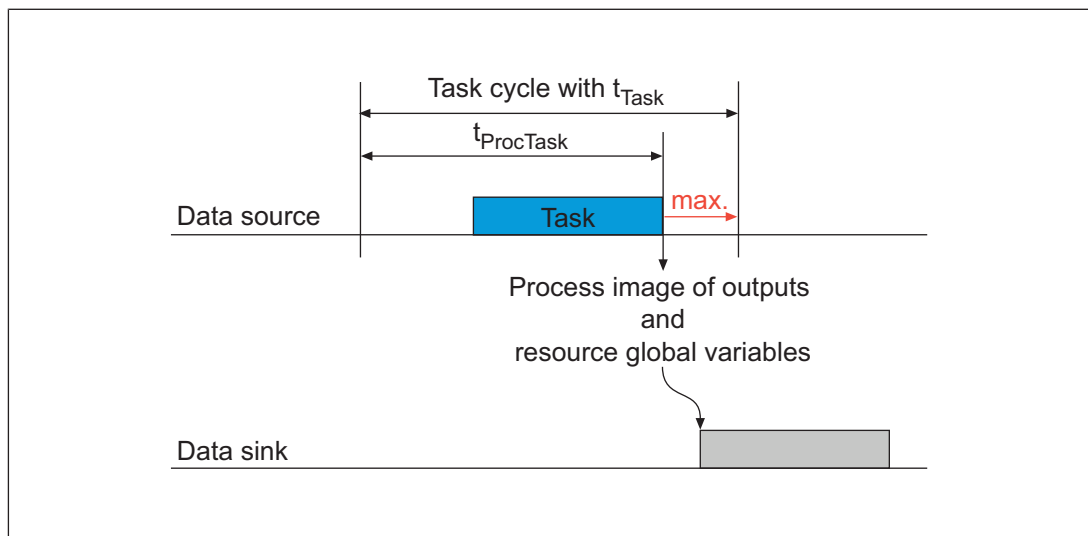


Fig.: Reaction time with a task's POU as data source

► Task as data sink

The reaction time for a datum starts at the beginning of the task cycle and ends after the execution time t_{ProcTask} with the process image of outputs and the resource global variables available for further processing.

The maximum reaction time is when the execution time t_{ProcTask} almost corresponds to the task cycle time t_{Task} .

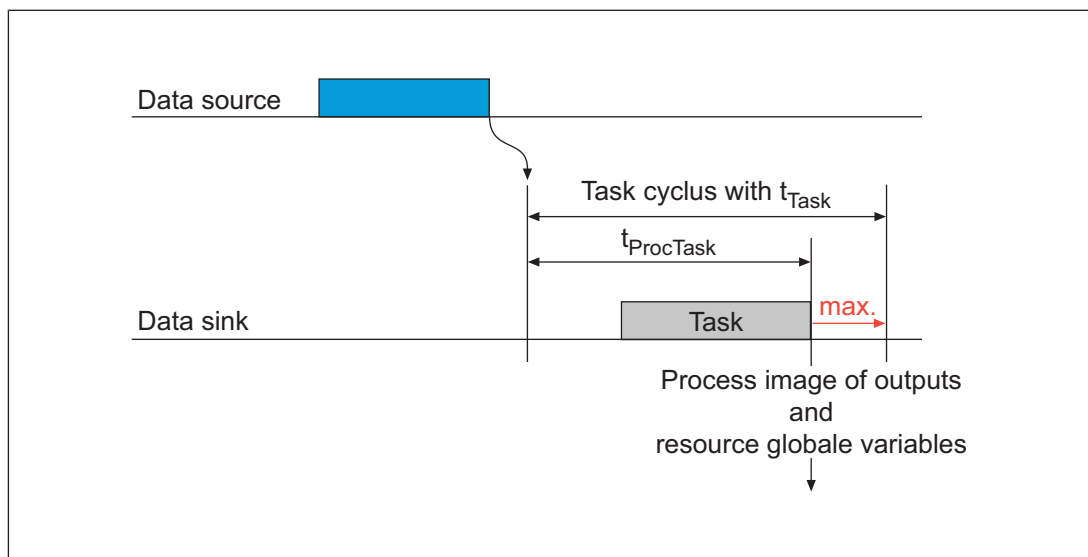


Fig.: Reaction time with a task as data sink

12.16.2.4 Principles for determining the maximum reaction time

Data subpaths are available to enable you to track a single I/O datum over a complete data path. A data path, therefore, is made up of the relevant data subpaths.

A data path always starts with the I-datum in a data source and ends with the O-datum in a data sink. A data source/data sink is always located in a system section (see [System sections](#) [22]).

The reaction time of a data subpath $t_{<i>}$ is viewed in the light of the following considerations:

$t_{<i>\_wcd}$	Worst-case reaction time, without a fault occurring on the data subpath $<i>$ or a fault occurring in a system section of the data subpath $<i>$ $<i>$: Data subpath wcd: worst case delay time
$t_{<i>\_ofd}$	One-fault reaction time, when one fault occurs on the data subpath $<i>$ or a fault occurs in a system section of the data subpath $<i>$ $<i>$: Data subpath ofd: one fault delay time

Principle for determining the maximum reaction time of a data path

$\text{Reaction time}_{\text{Data path\_max}} = t_{\text{Sum data subpaths\_wcd}} + \max(t_{<i>\_ofd} - t_{<i>\_wcd})$	
Reaction time _{Data path_max}	Maximum reaction time of a complete for data path
$t_{\text{Sum data subpaths\_wcd}}$	Sum of the worst-case reaction times of all relevant data subpaths, without a fault occurring
$\max(t_{<i>\_ofd} - t_{<i>\_wcd})$	Greatest difference value from one-fault reaction time $t_{<i>\_ofd}$ and worst-case reaction time $t_{<i>\_wcd}$ of a data subpath; i.e. with regard to the complete data path, the data subpath with the greatest difference value must be used for the calculation.
$t_{<i>\_ofd}$	One-fault reaction time of one data subpath $<i>$ when one fault occurs on this data subpath or a fault occurs in a system section of this data subpath
$t_{<i>\_wcd}$	Worst-case reaction time of a data subpath $<i>$, without a fault occurring

12.16.2.5 Maximum reaction times for data subpaths

The following description contains the formulas for calculating the maximum reaction time for all permitted data subpaths.

Both the worst-case reaction time and the one-fault reaction time of the data subpath must always be determined in order to calculate the reaction time of a data subpath.

12.16.2.5.1 Data subpaths with module bus PSS u2 as data source

PSS 4000 firmware from Version 1.25.x

Module bus PSS u2 -> Task

► Worst-case reaction time

$$t_{\text{MBUS2\_Task\_wcd}} = t_{\text{ProcIM\_wcd}} + 2t_{\text{MBUS}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + 2t_{\text{Task}}$$

$t_{\text{MBUS2\_Task\_wcd}}$	Worst-case reaction time for this data subpath
$t_{\text{ProcIM\_wcd}}$	Processing time for I-data on a module without a fault
t_{MBUS}	Module bus cycle time of the device
$t_{\text{FSTxDelay}}$	Minimum time difference between the transfer of receive data and send data on the module bus PSS u2
$t_{\text{FSRxDelay}}$	Minimum time difference between redundant I/O data when transferred on the FS module bus PSS u2
t_{Task}	Task cycle time of the task on the device

► One-fault reaction time

$$t_{\text{MBUS2\_Task\_ofd}} = t_{\text{ProcIM\_ofd}} + 2t_{\text{MBUS}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + 2t_{\text{Task}}$$

$t_{\text{MBUS2\_Task\_ofd}}$	One-fault reaction time for this data subpath
$t_{\text{ProcIM\_ofd}}$	Processing time for I-data on a module when one fault occurs
t_{MBUS}	Module bus cycle time of the device
$t_{\text{FSTxDelay}}$	Minimum time difference between the transfer of receive data and send data on the module bus PSS u2
$t_{\text{FSRxDelay}}$	Minimum time difference between redundant I/O data when transferred on the FS module bus PSS u2
t_{Task}	Task cycle time of the task on the device

Note:

► Minimum pulse duration

To ensure that a signal is safely read in and processed in the task, it must be present at the input at least during the minimum pulse duration. The minimum pulse duration corresponds to the sum of all one-fault reaction times of the data subpaths, from the input to the task.

► One-fault reaction time on data subpaths for PLe

The formula for calculating the one-fault reaction time is only guaranteed when all the following conditions are met:

- Two redundant FS inputs must be assigned two redundant input signals.
- The redundant FS inputs must be monitored for discrepancy. Discrepancy monitoring can either be configured in PAS4000 for a redundant terminal pair as a function of the FS module, or discrepancy monitoring must occur within the user program of the task in which the input signals (FS-I-data) are read in and processed.

► Worst-case and one-fault reaction times

The formulas for calculating the reaction times are only guaranteed under the following condition:

- The program that is executed in the task (data sink) must output the process image of outputs in the same task cycle in which the signal change at the input is detected and entered in the process image of inputs.

Module bus PSS u2 -> SafetyNET p

► Worst-case reaction time

$$t_{\text{MBUS2\_SNp\_wcd}} = t_{\text{ProcIM\_wcd}} + 2t_{\text{MBUS}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + 2t_{\text{SNp}}$$

$t_{\text{MBUS2\_SNp\_wcd}}$	Worst-case reaction time for this data subpath
$t_{\text{ProcIM\_wcd}}$	Processing time for a module's I-data without faults
t_{MBUS}	Module bus cycle time of the device
$t_{\text{FSTxDelay}}$	Minimum time difference between the transfer of receive data and send data on the module bus PSS u2
$t_{\text{FSRxDelay}}$	Minimum time difference when transferring redundant FS data on the FS module bus PSS u2
t_{SNp}	Cycle time for the cyclical start of the SNp Tx connection (send data) with project-internal SNp communication

► One-fault reaction time

$$t_{\text{MBUS2\_SNp\_ofd}} = t_{\text{ProcIM\_ofd}} + 2t_{\text{MBUS}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + 2t_{\text{SNp}}$$

$t_{\text{MBUS2\_SNp\_ofd}}$	One-fault reaction time for this data subpath
$t_{\text{ProcIM\_ofd}}$	Processing time for a module's I-data when one fault occurs
t_{MBUS}	Module bus cycle time of the device
$t_{\text{FSTxDelay}}$	Minimum time difference between the transfer of receive data and send data on the module bus PSS u2
$t_{\text{FSRxDelay}}$	Minimum time difference when transferring redundant FS data on the FS module bus PSS u2
t_{SNp}	Cycle time for the cyclical start of the SNp Tx connection (send data) with project-internal SNp communication

Note:

► One-fault reaction time on data subpaths for PLe

The formula for calculating the one-fault reaction time is only guaranteed when all the following conditions are met:

- Two redundant FS inputs must be assigned two redundant input signals.
- The redundant FS inputs must be monitored for discrepancy. Discrepancy monitoring can either be configured in PAS4000 for a redundant terminal pair as a function of the FS module, or discrepancy monitoring must occur within the user program of the task in which the input signals (FS-I-data) are read in and processed.

12.16.2.5.2 Data subpaths with SafetyNET p as data source

PSS 4000 firmware from Version 1.25.x

SafetyNET p_{Rec} -> Task

Project-internal SNp communication

► Worst-case reaction time

$$t_{\text{SNp\_Task\_wcd}} = t_{\text{DelaySNp}} + 2t_{\text{Task}}$$

$t_{\text{SNp\_Task\_wcd}}$	Worst-case reaction time for this data subpath
t_{DelaySNp}	Transmission delay time of SafetyNET p in a project
t_{Task}	Task cycle time of the task on the device

► One-fault reaction time

$$t_{\text{SNp\_Task\_ofd}} = t_{\text{DelaySNp}} + 2t_{\text{Task}}$$

$t_{\text{SNp\_Task\_ofd}}$	One-fault reaction time for this data subpath
t_{DelaySNp}	Transmission delay time of SafetyNET p in a project
t_{Task}	Task cycle time of the task on the device

Cross-project SNp communication (PASconnect interface version 1.3.x/1.4.x):

► Worst-case reaction time

$$t_{\text{SNp\_Task\_wcd}} = t_{\text{DelaySNp}} + \max(t_{\text{SNp}}, t_{\text{Task}}) + t_{\text{Task}}$$

$t_{\text{SNp\_Task\_wcd}}$	Worst-case reaction time for this data subpath
t_{DelaySNp}	Transmission delay time of SafetyNET p on the project containing the receiver
t_{SNp}	Receiver's cycle time $t_{\text{SNp (Rx)}}$ for the cyclical test for the presence of receive data from the SNp Rx connection
t_{Task}	Task cycle time of the task on the device

► One-fault reaction time

$$t_{\text{SNp\_Task\_ofd}} = t_{\text{DelaySNp}} + \max(t_{\text{SNp}}, t_{\text{Task}}) + t_{\text{Task}}$$

$t_{\text{SNp\_Task\_ofd}}$	One-fault reaction time for this data subpath
t_{DelaySNp}	Transmission delay time of SafetyNET p on the project containing the receiver
t_{SNp}	Receiver's cycle time $t_{\text{SNp (Rx)}}$ for the cyclical test for the presence of receive data from the SNp Rx connection
t_{Task}	Task cycle time of the task on the device

SafetyNET p_{Rec} -> Module bus PSS u2

► Worst-case reaction time

$$t_{\text{SNp\_MBUS2\_wcd}} = t_{\text{DelaySNp}} + 3t_{\text{MBUS}} + t_{\text{ProcHM}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + t_{\text{ProcOM\_wcd}}$$

$t_{\text{SNp\_MBUS2\_wcd}}$	Worst-case reaction time for this data subpath

t_{DelaySNp}	Transmission delay time of SafetyNET p in a project
t_{MBUS}	Module bus cycle time of the device
t_{ProcHM}	Maximum processing time for the processing of I/O-data on the head module
$t_{\text{FSTxDelay}}$	Minimum time difference between the transfer of receive data and send data on the module bus PSS u2
$t_{\text{FSRxDelay}}$	Minimum time difference between redundant I/O data when transferred on the FS module bus PSS u2
$t_{\text{ProcOM\_wcd}}$	Processing time for O-data of a module without faults

► One-fault reaction time

$$t_{\text{SNp\_MBUS2\_ofd}} = t_{\text{DelaySNp}} + 3t_{\text{MBUS}} + t_{\text{ProcHM}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + t_{\text{ProcOM\_ofd}}$$

$t_{\text{SNp\_Task\_ofd}}$	One-fault reaction time for this data subpath
t_{DelaySNp}	Transmission delay time of SafetyNET p in a project
t_{MBUS}	Module bus cycle time of the device
t_{ProcHM}	Maximum processing time for the processing of I/O-data on the head module
$t_{\text{FSTxDelay}}$	Minimum time difference between the transfer of receive data and send data on the module bus PSS u2
$t_{\text{FSRxDelay}}$	Minimum time difference between redundant I/O data when transferred on the FS module bus PSS u2
$t_{\text{ProcOM\_ofd}}$	Processing time for a module's O-data when one fault occurs

► One-fault reaction time on data subpaths for PLe

The formula for calculating the one-fault reaction time is only guaranteed when all the following conditions are met:

- The actuator must be connected to two redundant FS outputs.
- The actuator must be switched via two redundant output signals (FS-O-data).

12.16.2.5.3 Data subpath with a task as data source

PSS 4000 firmware from Version 1.25.x

Task1 -> Task2

► Worst-case reaction time

$$t_{\text{Task1\_Task2\_wcd}} = 2t_{\text{Task2}}$$

 $t_{\text{Task1\_Task2\_wcd}}$

Worst-case reaction time for this data subpath

 t_{Task2}

Task cycle time of task 2

► One-fault reaction time

$$t_{\text{Task1\_Task2\_ofd}} = 2t_{\text{Task2}}$$

 $t_{\text{Task1\_Task2\_ofd}}$

One-fault reaction time for this data subpath

 t_{Task2}

Task cycle time of task 2

Task -> Module bus PSS u2

► Worst-case reaction time

$$t_{\text{Task\_MBUS2\_wcd}} = 2t_{\text{MBUS}} + t_{\text{ProcHM}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + 2t_{\text{ProcOM\_wcd}}$$

 $t_{\text{Task\_MBUS2\_wcd}}$

Worst-case reaction time for this data subpath

 t_{MBUS}

Module bus cycle time of the device

 t_{ProcHM}

Maximum processing time for the I/O-data on the head module

 $t_{\text{FSTxDelay}}$

Minimum time difference between the transfer of receive data and send data on the module bus PSS u2

 $t_{\text{FSRxDelay}}$

Minimum time difference between redundant I/O data when transferred on the FS module bus PSS u2

 $t_{\text{ProcOM\_wcd}}$

Processing time for O-data on a module without faults

► One-fault reaction time

$$t_{\text{Task\_MBUS2\_ofd}} = 2t_{\text{MBUS}} + t_{\text{ProcHM}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + 2t_{\text{ProcOM\_ofd}}$$

 $t_{\text{Task\_MBUS2\_ofd}}$

One-fault reaction time for this data subpath

 t_{MBUS}

Module bus cycle time of the device

 t_{ProcHM}

Maximum processing time for the I/O-data on the head module

 $t_{\text{FSTxDelay}}$

Minimum time difference between the transfer of receive data and send data on the module bus PSS u2

 $t_{\text{FSRxDelay}}$

Minimum time difference between redundant I/O data when transferred on the FS module bus PSS u2

 $t_{\text{ProcOM\_ofd}}$

Processing time for O-data on a module when one fault occurs

Note:

► One-fault reaction time on data subpaths for PLe

The formula for calculating the one-fault reaction time is only guaranteed when all the following conditions are met:

- The actuator must be connected to two redundant FS outputs.
- The actuator must be switched via two redundant output signals (FS-O-data).

Task -> SafetyNET p_{Trans}

► Worst-case reaction time

$t_{\text{Task\_SNp\_wcd}} = 2t_{\text{SNp}}$

$t_{\text{Task\_SNp\_wcd}}$

Worst-case reaction time for this data subpath

t_{SNp}

Cycle time $t_{\text{SNp (Tx)}}$ of the transmitter for the cyclical start of the SNp-Tx connection (send data)

► One-fault reaction time

$t_{\text{Task\_SNp\_ofd}} = 2t_{\text{SNp}}$

$t_{\text{Task\_SNp\_ofd}}$

One-fault reaction time for this data subpath

t_{SNp}

Cycle time $t_{\text{SNp (Tx)}}$ of the transmitter for the cyclical start of the SNp-Tx connection (send data)

12.16.2.5.4 Example for calculating a data path on a PSS u2 system

In the example it is assumed that the FS-I datum (e.g. HW input on an FS module) and FS-O datum (e.g. HW output on an FS module) are on the same device.

The data path on the device is composed of the following:

FS module bus PSS u2 (I-datum) -> Task -> FS module bus PSS u2 (O-datum)

The data path is divided into the following data subpaths:

- ▶ FS module bus PSS u2 (I-datum) -> Task
- ▶ Task -> FS module bus PSS u2 (O-datum)

In accordance with the principle for determining the maximum reaction time of a data path (see [Principles for determining the maximum reaction time](#) [595]), the following applies for the example:

$t_{\max(A)}$	$= t_{\text{MBUS2\_Task\_wcd}} + t_{\text{Task\_MBUS2\_wcd}}$ $+ \max [$ $(t_{\text{MBUS2\_Task\_ofd}} - t_{\text{MBUS2\_Task\_wcd}}),$ $(t_{\text{Task\_MBUS2\_ofd}} - t_{\text{Task\_MBUS2\_wcd}})$ $]$

$t_{\text{MBUS2\_Task\_wcd}}$

Worst-case reaction time of the data subpath
FS module bus PSS u2 (I-datum) -> Task

$t_{\text{Task\_MBUS2\_wcd}}$

Worst-case reaction time of the data subpath
Task -> FS module bus PSS u2 (I-datum)

$\max [$

$(t_{\text{MBUS2\_Task\_ofd}} - t_{\text{MBUS2\_Task\_wcd}}),$
 $(t_{\text{Task\_MBUS2\_ofd}} - t_{\text{Task\_MBUS2\_wcd}})$
 $]$

With regard to the complete data path, the data subpath with the greatest difference value between the one-fault reaction time and the worst-case reaction time must be used for the calculation.

▶ Worst-case reaction times

- Data subpath FS module bus PSS u2 (I-datum) -> Task

$$t_{\text{MBUS2\_Task\_wcd}} = t_{\text{ProcIM\_wcd}} + 2t_{\text{MBUS}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + 2t_{\text{Task}}$$

- Data subpath Task -> FS module bus PSS u2 (O-datum)

$$t_{\text{Task\_MBUS2\_wcd}} = 2t_{\text{MBUS}} + t_{\text{ProcHM}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + t_{\text{ProcOM\_wcd}}$$

▶ One-fault reaction times

- Data subpath FS module bus PSS u2 (I-datum) -> Task

$$t_{\text{MBUS2\_Task\_ofd}} = t_{\text{ProcIM\_ofd}} + 2t_{\text{MBUS}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + 2t_{\text{Task}}$$

- Data subpath Task -> FS module bus PSS u2 (O-datum)

$$t_{\text{Task\_MBUS2\_ofd}} = 2t_{\text{MBUS}} + t_{\text{ProcHM}} - t_{\text{FSTxDelay}} - t_{\text{FSRxDelay}} + t_{\text{ProcOM\_ofd}}$$

For the maximum reaction time of the complete data path on device (A), the following applies:

$t_{\max(A)}$	$\begin{aligned} &= t_{\text{ProcIM\_wcd}} + 4t_{\text{MBUS}} - 2t_{\text{FSTxDelay}} - 2t_{\text{FSRxDelay}} + 2t_{\text{Task}} \\ &+ t_{\text{ProcHM}} + t_{\text{ProcOM\_wcd}} \\ &+ \max [\\ &\quad (t_{\text{MBUS2\_Task\_ofd}} - t_{\text{MBUS2\_Task\_wcd}}), \\ &\quad (t_{\text{Task\_MBUS2\_ofd}} - t_{\text{Task\_MBUS2\_wcd}}) \\ &] \end{aligned}$

12.16.2.5.5 Example for calculating a data path via three PSS u2 systems

In the example it is assumed that the complete data path runs across the 3 devices (A), (B) and (C).

The complete data path is composed as follows:

FS module bus PSS u2_(A) (I-datum) -> SafetyNET p_(Con_AB) -> Task_(B) -> SafetyNET p_(Con_BC) -> FS module bus PSS u2_(C) (O-datum)

The data path is divided into the following data subpaths:

- ▶ FS module bus PSS u2_(A) (I-datum) -> SafetyNET p_(Con_AB)
- ▶ SafetyNET p_(Con_AB) -> Task_(B)
- ▶ Task_(B) -> SafetyNET p_(Con_BC)
- ▶ SafetyNET p_(Con_BC) -> FS module bus PSS u2_(C) (O-datum)

In accordance with the principle for determining the maximum reaction time of a data path (see [Principles for determining the maximum reaction time](#) [595]), the following applies for the example:

$t_{\max(AC)}$	$= t_{\text{MBUS2\_SNp\_wcd}}(A)$ $+ t_{\text{SNp\_Task\_wcd}}(B)$ $+ t_{\text{Task\_SNp\_wcd}}(B)$ $+ t_{\text{SNp\_MBUS2\_wcd}}(C)$ $+ \max [$ $(t_{\text{MBUS2\_SNp\_ofd}}(A) - t_{\text{MBUS2\_SNp\_wcd}}(A)),$ $(t_{\text{SNp\_Task\_ofd}}(B) - t_{\text{SNp\_Task\_wcd}}(B)),$ $(t_{\text{Task\_SNp\_ofd}}(B) - t_{\text{Task\_SNp\_wcd}}(B)),$ $(t_{\text{SNp\_MBUS2\_ofd}}(C) - t_{\text{SNp\_MBUS2\_wcd}}(C))$ $]$

$t_{\text{MBUS2\_SNp\_wcd}}(A)$	Worst-case reaction time of the data subpath FS module bus PSS u2 _(A) (I-datum) -> SafetyNET p _(Con_AB)
$+ t_{\text{SNp\_Task\_wcd}}(B)$	Worst-case reaction time of the data subpath Task -> FS module bus PSS u2 (I-datum)
$t_{\text{Task\_SNp\_wcd}}(B)$	Worst-case reaction time of the data subpath SafetyNET p _(Con_AB) -> Task _(B)
$t_{\text{SNp\_MBUS2\_wcd}}(C)$	Worst-case reaction time of the data subpath SafetyNET p _(Con_BC) -> FS module bus PSS u2 _(C) (O-datum)
$\max [$ $(t_{\text{MBUS2\_SNp\_ofd}}(A) - t_{\text{MBUS2\_SNp\_wcd}}(A)),$ $(t_{\text{SNp\_Task\_ofd}}(B) - t_{\text{SNp\_Task\_wcd}}(B)),$ $(t_{\text{Task\_SNp\_ofd}}(B) - t_{\text{Task\_SNp\_wcd}}(B)),$ $(t_{\text{SNp\_MBUS2\_ofd}}(C) - t_{\text{SNp\_MBUS2\_wcd}}(C))$ $]$	With regard to the complete data path, the data subpath with the greatest difference value between the one-fault re- action time and the worst-case reaction time must be used for the calculation.

▶ Worst-case reaction times

- Data subpath FS module bus PSS u2_(A) (I-datum) -> SafetyNET p_(Con_AB)

$$t_{\text{MBUS2\_SNp\_wcd}}(A) = t_{\text{ProcIM\_wcd}} + 2t_{\text{MBUS}}(A) - t_{\text{FSTxDelay}}(A) - t_{\text{FSRxDelay}}(A) + 2t_{\text{SNp}}(\text{Con\_AB})$$

- Data subpath SafetyNET p_(Con_AB) -> Task_(B)

$$t_{\text{SNp\_Task\_wcd}}(B) = t_{\text{DelaySNp}} + 2t_{\text{Task}}(B)$$

– Data subpath Task_(B) -> SafetyNET p_(Con_BC)

$$t_{\text{Task\_SNp\_wcd (B)}} = 2t_{\text{SNp (Con\_BC)}}$$

– Data subpath SafetyNET p_(Con_BC) -> FS module bus PSS u2_(C) (O-datum)

$$t_{\text{SNp\_MBUS2\_wcd (C)}} = t_{\text{DelaySNp}} + 3t_{\text{MBUS (C)}} + t_{\text{ProcHM (C)}} - t_{\text{FSTxDelay (C)}} - t_{\text{FSRxDelay (C)}} + t_{\text{ProcOM\_wcd}}$$

► One-fault reaction times

– Data subpath FS module bus PSS u2_(A) (I-datum) -> SafetyNET p_(Con_AB)

$$t_{\text{MBUS2\_SNp\_ofd (A)}} = t_{\text{ProcIM\_ofd}} + 2t_{\text{MBUS (A)}} - t_{\text{FSTxDelay (A)}} - t_{\text{FSRxDelay (A)}} + 2t_{\text{SNp (Con\_AB)}}$$

– Data subpath SafetyNET p_(Con_AB) -> Task_(B)

$$t_{\text{SNp\_Task\_ofd (B)}} = t_{\text{DelaySNp}} + 2t_{\text{Task (B)}}$$

– Data subpath Task_(B) -> SafetyNET p_(Con_BC)

$$t_{\text{Task\_SNp\_ofd (B)}} = 2t_{\text{SNp (Con\_BC)}}$$

– Data subpath SafetyNET p_(Con_BC) -> FS module bus PSS u2_(C) (O-datum)

$$t_{\text{SNp\_MBUS2\_ofd (C)}} = t_{\text{DelaySNp}} + 3t_{\text{MBUS (C)}} + t_{\text{ProcHM (C)}} - t_{\text{FSTxDelay (C)}} - t_{\text{FSRxDelay (C)}} + t_{\text{ProcOM\_ofd}}$$

For the maximum reaction time of the complete data path on device (A), the following applies:

$t_{\text{max(AC)}}$	$ \begin{aligned} &= t_{\text{ProcIM_wcd}} + 2t_{\text{MBUS (A)}} - t_{\text{FSTxDelay (A)}} - t_{\text{FSRxDelay (A)}} + 2t_{\text{SNp (Con_AB)}} \\ &+ t_{\text{DelaySNp}} + 2t_{\text{Task (B)}} \\ &+ 2t_{\text{SNp (Con_BC)}} \\ &+ t_{\text{DelaySNp}} + 3t_{\text{MBUS (C)}} + t_{\text{ProcHM (C)}} - t_{\text{FSTxDelay (C)}} - t_{\text{FSRxDelay (C)}} + t_{\text{ProcOM_ofd}} \\ &+ \max [\\ &\quad (t_{\text{MBUS2_SNp_ofd (A)}} - t_{\text{MBUS2_SNp_wcd (A)}}), \\ &\quad (t_{\text{SNp_Task_ofd (B)}} - t_{\text{SNp_Task_wcd (B)}}), \\ &\quad (t_{\text{Task_SNp_ofd (B)}} - t_{\text{Task_SNp_wcd (B)}}), \\ &\quad (t_{\text{SNp_MBUS2_ofd (C)}} - t_{\text{SNp_MBUS2_wcd (C)}}) \\ &] \end{aligned} $

12.17 System time synchronisation

As devices in industrial automation systems are often involved in complex networks, each device is required to have the same system time. This is all the more important when precise data evaluation and processing of time-critical data (e.g. process data, reaction times for safety-related tasks, time stamp for logs) must not be delayed by the network load, the size of the data packets or mechanisms within the communication protocol.

In Ethernet-based industrial automation systems this requirement can be met using a system clock that provides the date and time. At least one real-time clock is used, depending on the applied technology, and this provides the reference time within the automation system (Server). All other network subscribers (Clients) synchronise their device time based on this reference time. In this way, each network subscriber has a standardised system time which is subject only to very minor deviations.

12.17.1 System time synchronisation with SNTP

The Simple Network Time Protocol (SNTP) is a synchronisation protocol used to communicate and synchronise time in Ethernet-based networks. SNTP operates in accordance with the Client-Server principle. It uses UDP as the communication protocol. The (time) Server uses the port with port number 123 for this service by default. On PSS 4000-exclusive devices the port is permanently set and cannot be modified by the user. The (time) Server is called an SNTP Server in PAS4000.

In the automation system PSS 4000, a project can be operated with or without an SNTP Server.

If an SNTP Server is configured for a project in PAS4000, SNTP is used to ensure the same time base for diagnostic messages and log entries for all devices within a project. This guarantees the correct chronological sequence of diagnostic messages coming from a variety of devices with their own real-time clock. STNP is also used to synchronise the time automatically after exchanging devices. PAS4000 only needs to be used to set the time on the new device if the exchanged unit is to be used as an SNTP Server.

The automation system PSS 4000 supports SNTP Unicast as a communication mechanism. This means that there can only be one SNTP Server in a project. All other network subscribers automatically become SNTP Clients (exception: PCs and PMIs can be excluded from automatic configuration as SNTP Client).

With the Unicast communication mechanism, an SNTP Client sends a time synchronisation request to the SNTP Server and receives time information from the SNTP Server.

On PSS 4000, time synchronisation between an SNTP Client and the configured SNTP Server occurs with a cycle time of ≤ 60 s. Users cannot configure this cycle time.

Any device in the project that has a real-time clock can be configured as the SNTP Server for the project. However, PAS4000 also supports the use of external devices. These may be special switches designed for this purpose, for example.

If **no** SNTP Server is configured for a project in PAS4000, then each device with a real-time clock operates with its own time base. As a result, there is no standard time base for diagnostic messages and log entries within the project. The correct chronological sequence of diagnostic messages cannot be guaranteed.

This also applies to projects with SNTP Server, in which PCs and PMIs are not configured as SNTP Client. In these projects, each PC and each PMI operates with its own time base, which may differ from the time base of the SNTP Server.

12.18 Memory

12.18.1 Memory types

A PSS 4000 device has various memories and memory areas. The memory types available on a PSS 4000 device and their memory size depends on the device. Details of the PSS 4000 devices from Pilz are available in the operating manual for the relevant device.

The table below contains the available memory types and their contents.

Memory type	Contents
Program and data memory	Program and data memory of the FS resource: ▸ FS user program ▸ Volatile FS variables Program and data memory of the ST resource: ▸ ST user program ▸ Volatile ST variables
Non-volatile FS memory	Non-volatile FS variables
Non-volatile ST memory	Non-volatile ST variables
SD card	▸ Device project ▸ Naming data
Security SD card	▸ Device project ▸ Naming data ▸ Data for special security functionality



NOTICE

When calculating the memory requirement please note that administrative data is always stored alongside variables and data; i.e. the actual memory requirement cannot be extrapolated from the bit width of data types.



INFORMATION

A PSS 4000 device needs write access to its SD card. If write protection is activated on the SD card, then the PSS 4000 device ignores write protection.

12.18.2 Memory structure

To ensure that the PSS 4000 firmware can manage its data, it requires various working directories in a directory on the data medium. The directory on the data medium is called the PSS 4000 data directory (PILZ_DATA_DIR).

With a PSS 4000-exclusive device, the PSS 4000 data directory is on the SD card. With a PSS 4000-operable device (e.g. PMI, PC), the PSS 4000 data directory is on the device. The PSS 4000 data directory has the same directory structure for all PSS 4000 devices, i.e. the PSS 4000 data directory contains different subdirectories with pre-defined folder names.

With regard to the assignment of memory space, the directories are differentiated into various directory types. There are directories with unlimited size and directories with limited size. Directories with limited size are used by functions that limit themselves with regard to the memory requirement. Functions that use the directories with unlimited size do not do this.

For SD cards, note the following: If the directories with unlimited size take up so much memory capacity that there is not enough memory capacity for the functions that use the directories with limited size, this may impair the performance of these functions.

Directory structure

Subdirectory	Content and use	Notes
/cfg	Naming data for the device, the firmware configuration and, with an extended network configuration, the network configuration file net.json	The directory is one of the directories with unlimited size.
/ps	Data for the device project	The directory is one of the directories with unlimited size.
/ep	Backup copy	The directory is one of the directories with unlimited size.
/debug	Debug data for the device	The directory is one of the directories with limited size. Please note for SD cards: The size of this directory is limited to max. 100 MByte. If this memory capacity is reached, old debug data is deleted.
/traces_st	Tracing configuration for the ST resource and trace files	The directory uses 40 % of the available memory.
/traces_fs	Tracing configuration for the FS resource and trace files	The directory uses 40 % of the available memory.
/CM-Stick	Security data	The directory is only available on a security SD card.

Subdirectory	Content and use	Notes
/userfiles	User-specific data and folder	The directory is available to the user in order to save certificates, documentation or other information relevant for the device or system, for example. The directory is one of the directories with unlimited size.



INFORMATION

When booting, all data and directories that are not relevant are deleted from the SD card, i.e.:

- data and directories that are not from the system and were stored by the user outside the /userfiles directory are deleted from the SD card.
- The following directories and files are retained on the SD card, along with the directories needed by functions of the PSS 4000 firmware:
 - /userfiles directory
 - CM-stick directory
 - "journal.jnl" file in the SD card's root directory
- Please note that only the directories of functions known to the device's PSS 4000 firmware remain on the SD card.

If the SD card of a device with a higher firmware version is used for a device, for example, the directories of all the functions available exclusively on the device with the higher firmware version will be deleted.



NOTICE

Impairing the performance of the file system

The performance of the file system decreases as more of the memory space is assigned. The performance is the lowest when the SD card is almost full. To maintain the performance of the file system, the memory capacity of directories with limited size are additionally limited by the fact that they may not occupy more than 80% of the SD card's total capacity.

Note that saving substantial data in the /userfiles directory can impair the performance of the file system.

Ensure that the SD card you use for your application has sufficient memory capacity.



NOTICE

Physically available disk space on an SD card

The memory size stated on an SD card is not the memory size that can physically be used by an application. Also there is always various administrative data stored on an SD card.

Please note that the physical disk space available to an application on the SD card is subject to variation for production and technological reasons.

You can display the memory usage in PAS4000. Further information is available in the online help under Resource assignment -> Show online resource information.

12.18.3 Security SD card

The security SD card contains not only the device project and the naming data, but also special data that is required for security functions. The security functions include in particular the protection of a PSS 4000 project against unauthorised access or use. In PAS4000, the device project can be linked with the security SD card for this purpose.

Security SD cards are available from Pilz, for example, under the designation "PASkey SD Card".

12.18.4 Journaling file system

All PSS 4000-exclusive devices use a proprietary journaling file system on their SD card. Before they are actually written, all changes are recorded in a specially reserved memory area, so that a consistent state can be reconstructed in the event of a potential power failure or system crash.



NOTICE

Damage to files on the SD card

Files may be damaged if the card is removed from the device or the power to the device is switched off as the SD card is being written.

Remove the SD card only in recovery mode or in switched-off state.

12.18.5 Bind SD card

An SD card is bound to a device when it is recognised by the device and the device can access the SD card.

With PSS 4000-exclusive devices, the SD card is bound either during the boot process or during operation, as soon as the SD card is inserted into the device.

A PSS 4000-exclusive device will not bind an SD card if it does not recognise the SD card or it cannot execute the journal of the journaling file system. In this case, all attempts to access the file system (e.g. save, read, modify or delete files) will fail.

13 Commissioning

13.1 General requirements

Secure the site in accordance with the regulations (barrier, warning signs etc.). The system may only be commissioned/recommissioned by competent persons.

- ▶ Please refer to the information and specifications stated in the relevant documentation (e.g. Safety manual PSS 4000, Installation Manual PSSuniversal, Installation Manual PSS u2, operating manuals of the relevant modules).
- ▶ During commissioning/recommissioning, make sure that no personal injury and/or material damage can occur, even if the plant/machine moves unintentionally (e.g. physical barrier for the danger zone or for the whole plant/machine if necessary).

13.2 Initial commissioning

The following description assumes that

- ▶ The following documentation has been considered when deciding on the installation site, mechanical structure and electrical connection
 - Installation Manual PSSuniversal / Installation Manual PSS u2
 - "SafetyNET p" chapter in this System Manual (if necessary)
 - Operating manuals for the respective modules
 - Installation guidelines for the respective standard fieldbus (if necessary)
 - Relevant documents from the risk analysis (e.g. assembly instructions, wiring diagram)
- ▶ All the required devices are fully assembled, wired and connected
- ▶ An SD card is inserted on all PSS 4000-exclusive devices
- ▶ The project in PAS4000 is complete
(See PAS4000 Online Help)
- ▶ The project has been downloaded
(See PAS4000 Online Help)
- ▶ The project download instructions have been followed and any errors have been recovered
(e.g. set hardware registry matches the actual hardware)
- ▶ Description is available of all the activities required for commissioning (e.g. check lists and test specifications from the risk analysis)
- ▶ The supply voltages to the plant/machine are **not** switched on at this point.

**WARNING!****Hazard due to use of an unreleased firmware version!**

When operated in accordance with their intended purpose (normal operation or productive mode), the following devices may only be used when they have a released firmware version from Pilz installed, otherwise a loss of the safety function may result.

- 2nd generation PSS 4000-exclusive devices (PSS u2)
- PSS 4000-exclusive devices with head module PSSu H FS2 SN SD(-T)
- PSS 4000-exclusive devices with head module PSSu H PLC2 FS SN SD(-T)

Ensure that your application only takes up its intended operation when a released firmware version from Pilz is installed on the stated PSS 4000-exclusive devices. For this purpose, check the status LEDs on the head module.

- **Restart:**

The LEDs in the head module's left-hand column light up in succession.

Green chase light: Released firmware version from Pilz

The device may be used in accordance with the intended purpose.

Red chase light: Test version

The device may only be used for test purposes.

- **"DIAG" LED flashes red quickly during operation:**

A test version of the firmware is installed. The device may only be used for test purposes.

Procedure**1. Check installation and wiring**

Check the assembly and wiring based on the specific assembly instructions and wiring diagram for the plant/machinery.

Please refer to the relevant documents from the risk analysis (see PSS 4000 Safety Manual).

2. Switch on supply voltages

Switch on the supply voltage to all devices together.

It may take a little time (max. 10 s for a PSS 4000-exclusive device) for all PSS 4000-exclusive devices to be ready for operation. You will know when a PSS 4000-exclusive device is ready for operation by the status LEDs.

3. Use commissioning aids

PAS4000 provides some important aids to help with any problems during commissioning:

- Watch variables

Watches the current state of variables dynamically

- Control variables

The value of variables is set once to a specified value. It can then be changed by the user program.

Make sure you heed the information and requirements stipulated in the PSS 4000 Safety Manual, even when controlling non-safety-related variables!

- Force variables

The value of variables is permanently assigned and cannot be changed by the user program. If "Force variables" is stopped, the variable will retain its force value until it is overwritten with a new value.

Make sure you heed the information and requirements stipulated in the PSS 4000 Safety Manual, even when forcing non-safety-related variables!

- Online changes

"Online changes" is an aid in PAS4000 when searching for programming errors. After a project download, individual POU's in the user program can be changed (application part and temporary variables in the declaration part), without the need to repeat the download. Online changes remain active until they are deactivated.

It is essential to heed the information and requirements stipulated in the PSS 4000 Safety Manual!

- Diagnostic log

Log that shows when diagnostic messages arrived, when they were cleared; It also contains entries regarding operation and debugging.

- Diagnostic list

List of current diagnostic messages (project diagnostic list, device diagnostic list)

- Project comparison

Option in PAS4000 to find out whether the original project in PAS4000 is identical to the project on the machine

The online project is compared with the offline project. Differences in the check sum "FS project" are displayed.

4. Test safety functions and fault detection facilities

If a project contains safety functions and fault detection facilities, use the test specification to test all the plant-specific safety functions and fault detection facilities.

It is **essential** that all the specified safety functions and fault detection facilities are fully tested.

It is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

5. Test FS blocks with mixed input interface

If FS blocks with mixed input interface have been used, test these using the defined test procedures (e.g. plausibility tests).

It is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

6. Optimise task cycle time

During commissioning you can optimise the task cycle time.

7. **Create backup copy**

Create a backup copy of the project, declare the project as the original project and save it in accordance with the project backup plan.

8. **Documentation**

- Document and log all actions taken during commissioning.

For all safety-related actions it is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

- Document the check sum "FS project" of the original project

9. **Normal operation**

The project is running normally once all devices are switched on and no errors are present.

13.3 Recommissioning

Recommissioning is always necessary when a change is made to the plant/machine or process. In all cases, the following changes must result in recommissioning:

- ▶ A safety requirement has changed
- ▶ A change due to the occurrence of a systematic error
- ▶ A change due to new operation or production requirements
- ▶ A change to the plant/machine by:
 - Adding or removing components (e.g. PSSu system, a PSSu system module)
 - Exchanging components (e.g. PSSu system, a PSSu system module, SD card of a PSS 4000-exclusive device)
- ▶ A change to the process cycle

The following description assumes that

- ▶ A risk analysis has been performed prior to any change
- ▶ The following documentation has been considered when adding, removing or exchanging components
 - Installation Manual PSSuniversal / Installation Manual PSS u2
 - "SafetyNET p" chapter in this System Manual (if necessary)
 - Operating manuals for the respective modules
 - Installation guidelines for the respective standard fieldbus (if necessary)
 - Relevant documents from the risk analysis for the changes
- ▶ All modified components are fully assembled/plugged in and wired
- ▶ An SD card is inserted on all PSS 4000-exclusive devices
- ▶ The changes to the project in PAS4000 are complete
(See PAS4000 Online Help)
- ▶ The project has been downloaded
(See PAS4000 Online Help)
- ▶ The project download instructions have been followed and any errors have been re-covered
(e.g. set hardware registry matches the actual hardware)
- ▶ Description is available of all the activities required for recommissioning (e.g. check lists and test specifications from the risk analysis)
- ▶ The supply voltages to the plant/machine are **not** switched on at this point.

**WARNING!****Hazard due to use of an unreleased firmware version!**

When operated in accordance with their intended purpose (normal operation or productive mode), the following devices may only be used when they have a released firmware version from Pilz installed, otherwise a loss of the safety function may result.

- 2nd generation PSS 4000-exclusive devices (PSS u2)
- PSS 4000-exclusive devices with head module PSSu H FS2 SN SD(-T)
- PSS 4000-exclusive devices with head module PSSu H PLC2 FS SN SD(-T)

Ensure that your application only takes up its intended operation when a released firmware version from Pilz is installed on the stated PSS 4000-exclusive devices. For this purpose, check the status LEDs on the head module.

- **Restart:**

The LEDs in the head module's left-hand column light up in succession.

Green chase light: Released firmware version from Pilz

The device may be used in accordance with the intended purpose.

Red chase light: Test version

The device may only be used for test purposes.

- **"DIAG" LED flashes red quickly during operation:**

A test version of the firmware is installed. The device may only be used for test purposes.

Procedure**1. Check installation and wiring**

Check the assembly and wiring of the modified components based on the specific assembly instructions and wiring diagram for the plant/machinery.

Please refer to the relevant documents from the risk analysis for the changes (see PSS 4000 Safety Manual).

2. Switch on supply voltages

Switch on the supply voltage to all devices together.

It may take a little time (max. 10 s for a PSS 4000-exclusive device) for all PSS 4000-exclusive devices to be ready for operation. You will know when a PSS 4000-exclusive device is ready for operation by the status LEDs.

3. Use commissioning aids

PAS4000 provides some important aids to help with any problems during recommissioning:

- Watch variables

Watches the current state of variables dynamically

- Control variables

The value of variables is set once to a specified value. It can then be changed by the user program.

Make sure you heed the information and requirements stipulated in the PSS 4000 Safety Manual, even when controlling non-safety-related variables!

- Force variables

The value of variables is permanently assigned and cannot be changed by the user program. If "Force variables" is stopped, the variable will retain its force value until it is overwritten with a new value.

Make sure you heed the information and requirements stipulated in the PSS 4000 Safety Manual, even when forcing non-safety-related variables!

- Online changes

"Online changes" is an aid in PAS4000 when searching for programming errors. After a project download, individual POU's in the user program can be changed (application part and temporary variables in the declaration part), without the need to repeat the download. Online changes remain active until they are deactivated.

It is essential to heed the information and requirements stipulated in the PSS 4000 Safety Manual!

- Diagnostic log

Log that shows when diagnostic messages arrived, when they were cleared; It also contains entries regarding operation and debugging.

- Diagnostic list

List of current diagnostic messages (project diagnostic list, device diagnostic list)

- Project comparison

The FS project contains various FS check sums, on which the user can base the project comparison.

- Block comparison

Each FS block contains a check sum on which the user can base the block comparison

4. Test safety functions and fault detection facilities

Following a change, if the safety analysis has shown that safety functions need to be validated and tested, the change itself must be tested, as must the course of the entire process. It must be ensured that only the desired changes are made to the project.

Please note:

During a project comparison, the check sum "FS project" can be used to determine whether any changes have been made to the FS project. If the "FS project" check sum is unchanged, then the FS project has not changed. If the "FS project" check sum has changed, then changes have been made to the FS project. Under no circumstances does the project comparison replace a prior safety analysis for changes. The project comparison can be used to confirm the analysis.

With any type of change to the FS project, you should note the procedure documented in the PSS 4000 Safety Manual (chapter entitled "Change, maintenance, decommissioning", under "Changes").

5. **Test FS blocks with mixed input interface**

If FS blocks with mixed input interface are affected by the changes, test these using the defined test procedures (e.g. plausibility tests).

It is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

6. **Optimise task cycle time**

During recommissioning you can optimise the task cycle time.

7. **Create backup copy**

Create a backup copy of the amended project, declare the amended project as the new original project and save it in accordance with the project backup plan.

8. **Documentation**

- Document all changes
- Document and log all actions taken during recommissioning.

For all safety-related actions it is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

- Document the check sum "FS project" of the new original project

9. **Normal operation**

The project is running normally once all devices are switched on and no errors are present.

14 Maintenance, repair, decommissioning

14.1 Maintenance and repair

Before starting maintenance or repair work, please note the following:

- ▶ Maintenance and repair work may only be carried out by competent persons.
- ▶ Make sure that power to the plant or machine is removed in accordance with the regulations.
- ▶ The plant or machine should be safeguarded against inadvertent reconnection.
- ▶ If you are exchanging components (e.g. PSSu system, a PSSu system module, SD card of a PSS 4000-exclusive device), then
 - Note the information and requirements stipulated in the PSS 4000 Safety Manual
 - Switch the device to a STOP state before exchanging components;
if you wish to exchange the SD card, this must only be done when the voltage is switched off
 - Note that if other devices access the process data on the PSS4000-exclusive device, then substitute values will be used in the other devices!
 - When exchanging the SD card, take care to only insert into the device the SD card that is intended for that device
 - When exchanging the SD card, take care to insert into the device the SD card containing the correct version of the device project
 - When exchanging the SD card, note that a new device naming procedure may need to be carried out (see [Device naming when exchanging devices](#) [400])
 - Note the information on recommissioning (see [Recommissioning](#) [619])

Maintenance and repair work on PSS 4000-exclusive devices

No maintenance work is necessary on a PSS 4000-exclusive device. Please send any faulty devices and/or modules back to Pilz.



INFORMATION

When exchanging modules and PSS 4000-exclusive devices, please comply with the information regarding assembly/removal as stipulated in the PSSuniversal Installation Manual.

14.2 Decommissioning

Please comply with the mission time T_M in the safety-related characteristic data of FS modules.

Note the information about assembly/disassembly in the product's documentation.

► PSSu systems:

- PSSuniversal Installation Manual (document number 21262)
- Operating manuals for the modules you are using

► PSS67 devices:

Operating manuals for the devices you are using.

When decommissioning, please comply with local regulations regarding the disposal of electronic devices (e.g. Electrical and Electronic Equipment Act).

15 OPC Server (classic)

15.1 Overview

OPC (Openness, Productivity and Collaboration) is a standardised software interface, through which PC-based software from a range of manufacturers can exchange data. The OPC Foundation is responsible for maintaining and propagating the standard. Around one hundred companies are currently members, including Pilz.

The OPC standard is now called "OPC Classic" and is specified for the Windows operating systems, depending on the platform. It is based on Windows technologies such as OLE (Object Linking and Embedding), COM (Component Object Model) and DCOM (Distributed COM).

On networked plants, an OPC Server will typically collect diagnostic data, process data etc. and display it in the defined namespace. An OPC Client is typically used for process visualisation and/or to record operating data. It accesses and then processes the data provided by the OPC Server. In order to process the data the OPC Client must contain the relevant OPC-enabled software.

In the automation system PSS 4000, a PC obtains the system section PSS 4000 OPC Server (classic) by installing the PSS 4000 firmware. By installing the PSS 4000 firmware, a PC becomes a "PSS 4000-operable device" (see [System overview \[📖 20\]](#)).

The following boundary conditions apply to the PSS 4000 OPC Server (classic):

- ▶ The PSS 4000 OPC Server (classic) supports various OPC Classic properties for access from OPC Classic Clients (see [OPC properties \[📖 628\]](#)).
- ▶ The PSS 4000 OPC Server (classic) can be installed on a PC (see [System requirements \[📖 629\]](#)).
- ▶ Each PSS 4000 project created using PAS4000 may contain up to 4 OPC Servers. OPC Classic Servers and OPC UA Servers may be combined in the process.
- ▶ A PSS 4000 project containing a 2nd generation device (PSS u2 system) may not contain a PSS 4000 OPC Server (classic).
- ▶ If a PSS 4000 firmware version $\geq 1.25.0$ is configured in the PSS 4000 project for a PC, then this PC cannot be configured as a PSS 4000 OPC Server (classic).
- ▶ If the PSS 4000 firmware is installed on a PC, then the system section PSS 4000 OPC Server (classic) will be available on the PC; this also applies for firmware versions $\geq 1.25.0$. A device project for a PSS 4000 OPC Server (classic) can be run on the PC. For this purpose, a firmware version $< 1.25.0$ must be configured in the project for the PC.
- ▶ It is possible to access an OPC Server simultaneously with more than one Client. The following applies:
 - Only OPC Classic Clients may access an OPC Classic Server.
 - Only OPC UA Clients may access an OPC UA Server.
- ▶ All project-based settings for a PSS 4000 OPC Server (classic) are made in PAS4000 and are transferred to the PSS 4000 OPC Server (classic) as a device project when the project is downloaded.

15.1.1 Design options

In a minimum configuration, the OPC Server will be on the same PC as the OPC Client with the visualisation software.

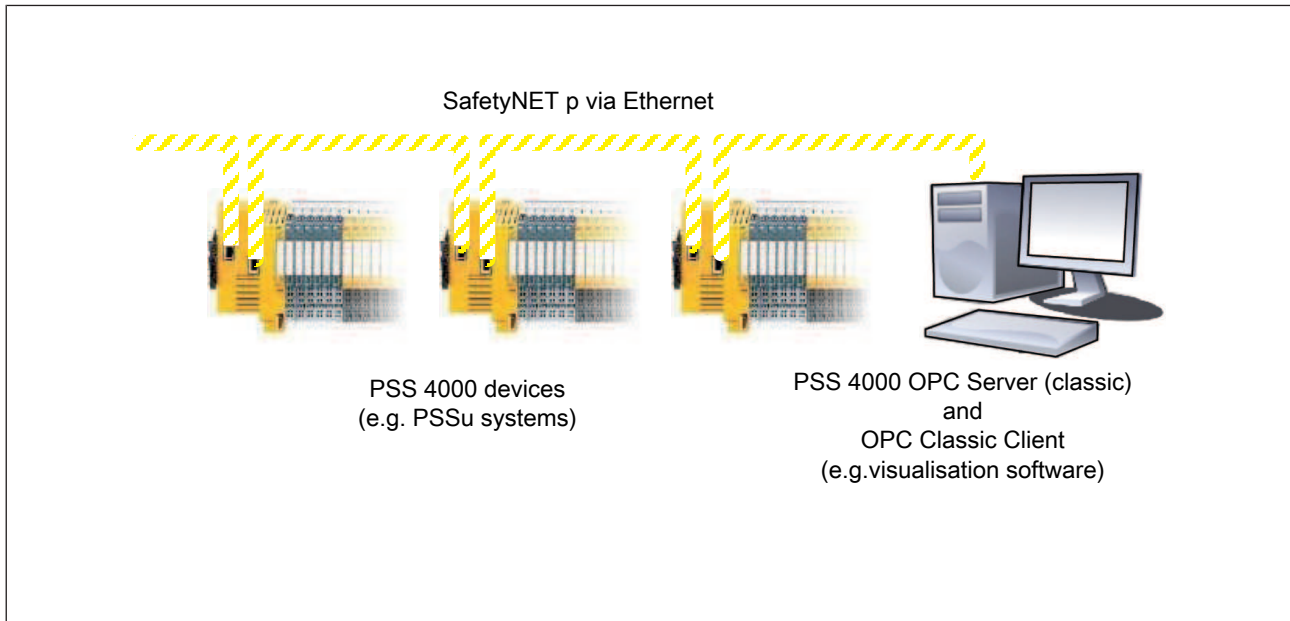


Fig.: Process visualisation with a minimum OPC Classic configuration (principle)

Other types of combinations are also conceivable, such as:

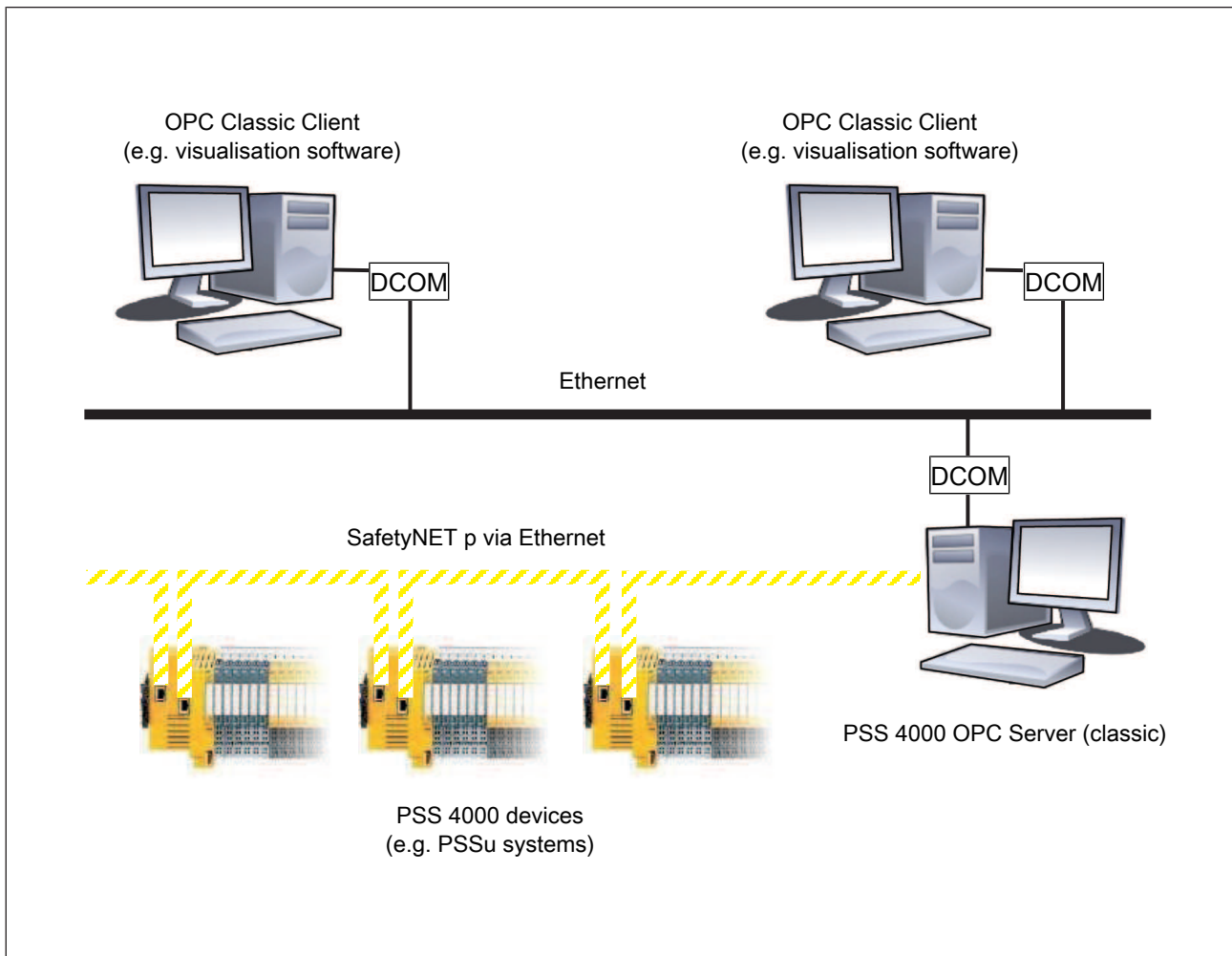


Fig.: Networking OPC Classic Servers and OPC Classic Clients via Ethernet (principle)

15.1.2 OPC properties

The PSS 4000 OPC Server (classic) supports the following OPC properties for access from OPC Classic Clients:

- ▶ OPC specification with
 - Data Access 1.0.a
 - Data Access 2.05
 - Data Access 3.0
- ▶ OPC interfaces with
 - All "Mandatory Interfaces" in accordance with the data access specification
 - Symbolic access to process data in the namespace
- ▶ OPC data types
 - VT_I1, VT_I2, VT_I4, VT_I8
 - VT_UI1, VT_UI2, VT_UI4, VT_UI8
 - VT_R4, VT_R8
 - VT_BOOL
 - VT_BSTR
 - VT_DATE

15.2 System requirements

Various system requirements must be met so that the PSS 4000 firmware can be installed on a device.

15.2.1 PC with Microsoft Windows operating system

The PSS 4000 firmware can be installed on a PC with a Microsoft Windows operating system:

- ▶ Microsoft Windows 10 in one of these versions
 - Windows 10 Home (64 Bit)
 - Windows 10 Pro (64 Bit)
 - Windows 10 Enterprise (64 Bit)

15.3 Licensing

Licensing of the OPC Server takes place when a project is licensed in PAS4000. Further information on licensing is available in PAS4000's online help.

15.4 Installation

The current installation files are available in the download area of the Pilz homepage (www.pilz.de -> Downloads). "Software" can be entered as the search term, for example. The software download is only available for registered users. Unregistered users can register free of charge. A valid E-mail address is required for registration.

Installation kits are available for the PSS 4000 firmware, depending on the application area:

- The PSS 4000 firmware is installed on a **PC**:

Required installation set:

PSS_4000_PC_<Version number>_<Identifier>.zip

(e.g. PSS_4000_PC_1_13_0_3000763B29.zip)

15.4.1 Install PSS 4000 firmware on a PC

The PSS 4000 firmware is installed on the PC that serves as the diagnostic unit. The diagnostic unit may also be used simultaneously as a display unit. When installing the PSS 4000 firmware, both the PSS 4000 OPC Server (classic) and the PSS 4000 OPC Server UA are installed.

The OPC Core Components are also installed automatically for the PSS 4000 OPC Server (classic). The OPC Core Components consist of various auxiliary programs and software components for OPC Servers. The PSS 4000 Firmware Controller is also installed.

For the PSS 4000 OPC Server UA, a self-signed certificate with public key and associated private key is created during installation.



INFORMATION

For compatibility reasons, the system section PSS 4000 OPC Server (classic) is also still included in the current PSS 4000 firmware versions. However, use of the PSS 4000 OPC Server (classic) is not recommended in new projects.

Refer to the table in the section entitled "Overview of the system sections of PSS 4000-operable devices", under [System sections](#)  22].



INFORMATION

- If the OPC Server is installed on the same PC as PAS4000, the PAS4000 firmware version must be identical to the version of the PSS 4000 firmware.
- If the OPC Server is installed on the same PC as PAS4000, we recommend that you install PAS4000 first, followed by the PSS 4000 firmware.
- If the OPC Server is installed on the same PC as PAS4000, the PSS 4000 firmware sets the system variable PILZ_SNP_MULTICAST_TTL to 64 during installation.
- If the OPC Server is installed on the same PC as PAS4000, PAS4000 may affect the project on the OPC Server. If an online action in PAS4000 is executed in another project, the project on the OPC Server may become unusable and may break the connections to the OPC Clients.
- An OPC Client (classic) can only access a PSS 4000 OPC Server (classic).
- An OPC UA Client can only access a PSS 4000 OPC Server UA.



NOTICE

Start OPC Server

Before an OPC Client can access the OPC Server, the Server must first be started. There are various procedures for doing this:

- Automatic start via the Autostart menu
When installing the PSS 4000 firmware, the default setting in the installation window is retained. This means that the PSS 4000 Firmware Controller is entered in the Autostart menu of the Windows operating system. Each time the PC is booted, the OPC Server is started automatically.
- Manual start
During installation, the option responsible for the entry in the Autostart menu is deselected in the installation window. This means that the OPC Server must be started manually via the PSS 4000 Firmware Controller after installation and each time the PC is booted.

Installing via the install program

- ⇒ Double-click on the install file "PSS_4000_PC_<Version number>_Installer.exe" and follow the on-screen instructions.

Uninstalling via the uninstall program

- ⇒ In the Windows program list right-click on **Pilz -> PSS 4000 firmware for PC <Version number>** and select **Uninstall**.

Silent install

It is also possible to have an unattended ("silent") installation of the PSS 4000 firmware. In this case the installation is carried out without any user interaction and the installation process runs in the background.

- ⇒ Start the installation by entering one of the commands described below in the Windows command line, the Windows search field or in a batch file.

Command for installation with default values:

"<Absolute directory of the install file>" /S /PilzEULA='Y'

Example:

```
"c:\myFiles\PSS_4000_PC_<Version number>_Installer.exe" /S /
PilzEULA='Y'
```

Command for installation with specifications:

"<Absolute directory of the install file>" /S /auto='<Value>' /lang='<Tool language>' /
PilzEULA='<Consent to EULA>' / D='<Absolute installation directory>'

Example:

```
"c:\myFiles\PSS_4000_PC_<Version number>_Installer.exe" /S /
auto='1' /lang='de' /PilzEULA='Y' /D=C:\myPrograms
```

Supported arguments:

Argument	Meaning
/S	Run file silently.
/auto	With this argument you can control whether the PSS 4000 firmware is started automatically when booting the PC. Argument is optional. Value range: 1: PSS 4000 firmware is started automatically when booting the PC 0: PSS 4000 firmware is not started automatically Default value: 1
/lang	Language for messages and language for the uninstall program, if the uninstall is not run silently. Argument is optional. Value range: de: German en: English Default value: If the language for the operating system is German, then German is the default language, otherwise English.
/PilzEULA	Consent to end user licence agreement for the PSS 4000 firmware; the text file containing the licence text is next to the install file in the folder "Licence\PSS4000_PC_x_x_x_EULA_End_User_Licence_Agreement". Please read the licence text and give your consent before starting the installation. Consent is a requirement for running the installation. Valid value: Y All other values or a missing argument will mean that the installation is not run.
/D	Installation directory; must be entered as the last argument, argument is optional Default value: %programfiles(x86)%\PILZ

Note: "Start /wait" can be added to the install command so that no further commands are started until the install command has been executed in full. Please note that there must be two double quotation marks between "start" and "/wait".

Example:

```
Start "" /wait "c:\myFiles\PSS_4000_PC_<Version number>_Installer.exe" /S /auto='1' /lang='de' /PilzEULA='Y' /D=C:\myPrograms
```

Silent uninstall

The uninstall can also be run silently.

⇒ Start the uninstall by entering the command described below in the Windows command line, the Windows search field or in a batch file.

Uninstall command:

"<Absolute directory of the uninstall file>" /S

Example:

```
"C:\Program Files (x86)\Pilz\PAS4000 1.20.0\uninstallerData\uninstall.exe" /S
```


Note: "Start /wait" can be added to the install command so that no further commands are started until the uninstall command has been executed in full. Please note that there must be two double quotation marks between "start" and "/wait".

Example:

```
Start "" /wait "C:\Program Files (x86)\Pilz\PAS4000 1.20.0\uninstallData\uninstall.exe" /S
```

15.4.1.1 First-time installation of the PSS 4000 firmware

With a first-time installation, proceed as follows:

1. Start the PC on which the PSS 4000 firmware is to be installed.
2. If you haven't already done so:

Create a directory on the PC, copy the installation kit (ZIP file) into the directory and unzip it. The following file will be available:

⇒ PSS_4000_PC_<Version number>_Installer.exe
(e.g. PSS_4000_PC_1_14_0_Installer.exe)

3. Double-click on the EXE file

Follow the on-screen instructions to complete the installation.

The following tasks are executed on the PC:

- The PSS 4000 OPC Server (classic) is installed along with the OPC Core Components
- The PSS 4000 OPC Server UA is installed
A self-signed certificate with a public key and associated private key is created.
- The PSS 4000 Firmware Controller is installed
- The software is registered in the registry file
- An icon is added to the task bar to display the status for the PSS 4000 Firmware Controller.

A change in state (start/stop) can be selected by right-clicking on the icon.

This completes the installation.



INFORMATION

For the PSS 4000 OPC Server (classic) please note: During registration (PC registry file), the installation program configures the DCOM settings for the OPC Server. If this is a first-time installation of the OPC Server on the PC, the DCOM setting for "anonymous remote access" will only be effective after the PC is restarted.

15.5 Communication and security settings for PCs

The settings described below are only required if

- ▶ The PSS 4000 OPC Server (classic) is being operated on a PC and
- ▶ OPC Classic Clients that are being operated on PCs access the OPC Server via Ethernet.

Example

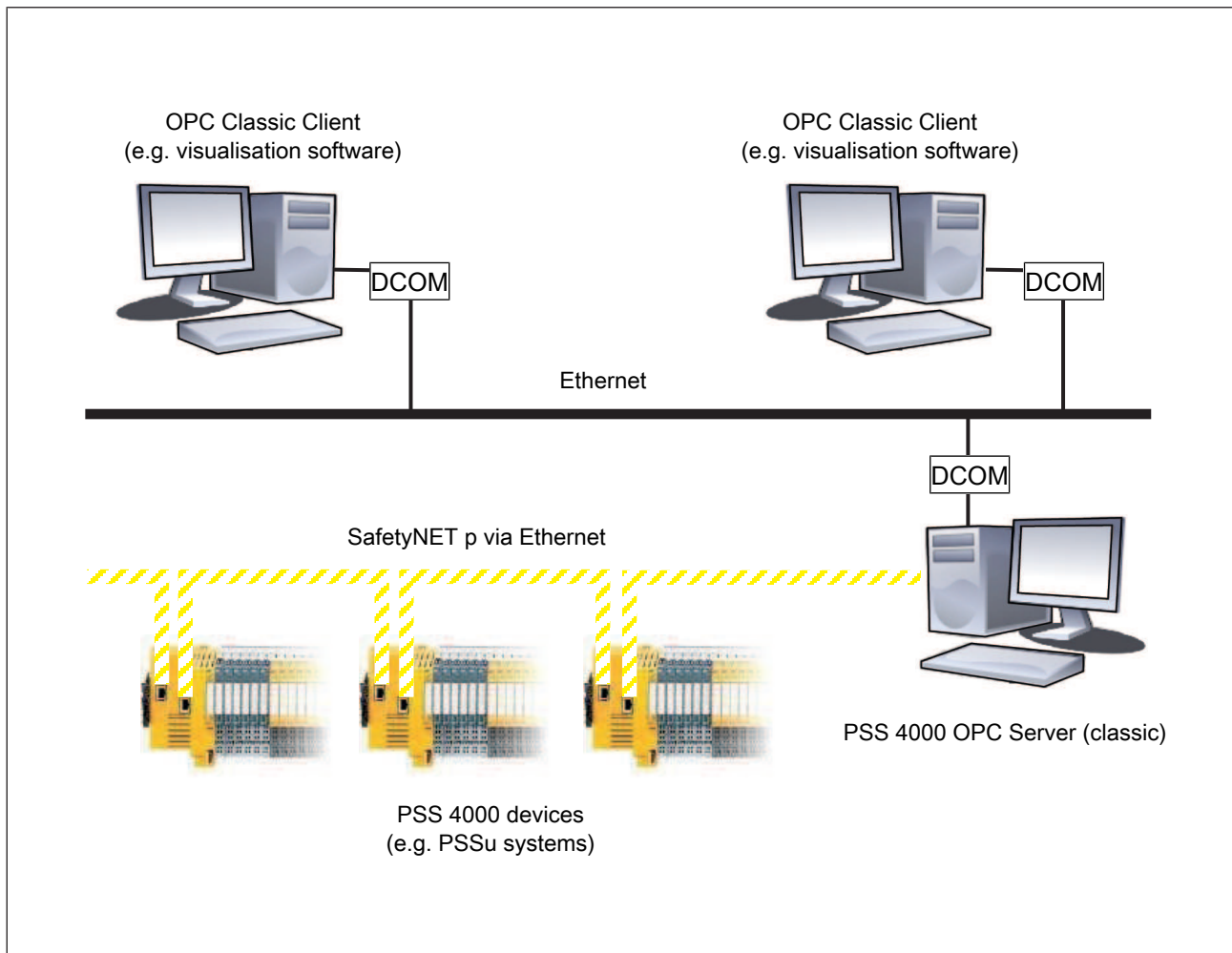


Fig.: OPC Classic Clients accessing the PSS 4000 OPC Server (classic) via Ethernet

To enable OPC Classic Clients to access the PSS 4000 OPC Server (classic) via Ethernet, various settings must be made in the DCOM configuration for all PCs involved in data exchange. These settings are documented below.

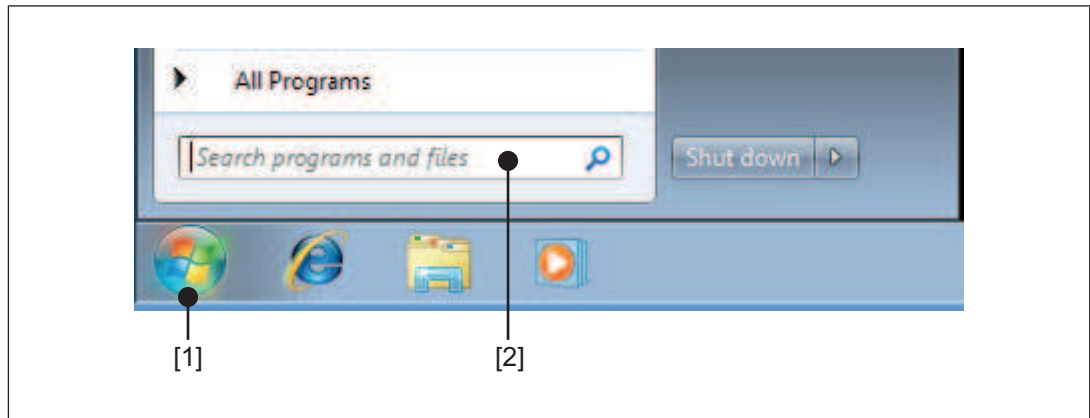
DCOM only permits authenticated access between PCs, so we recommend that the PCs (OPC Server, OPC Clients) are registered in the same domain.

The settings in the DCOM configuration grant access rights to the OPC Server for other network users. An incorrect configuration can compromise the safety and stability of the operating system. A network administrator can optimise these configurations to protect the OPC Server from unauthorised access.

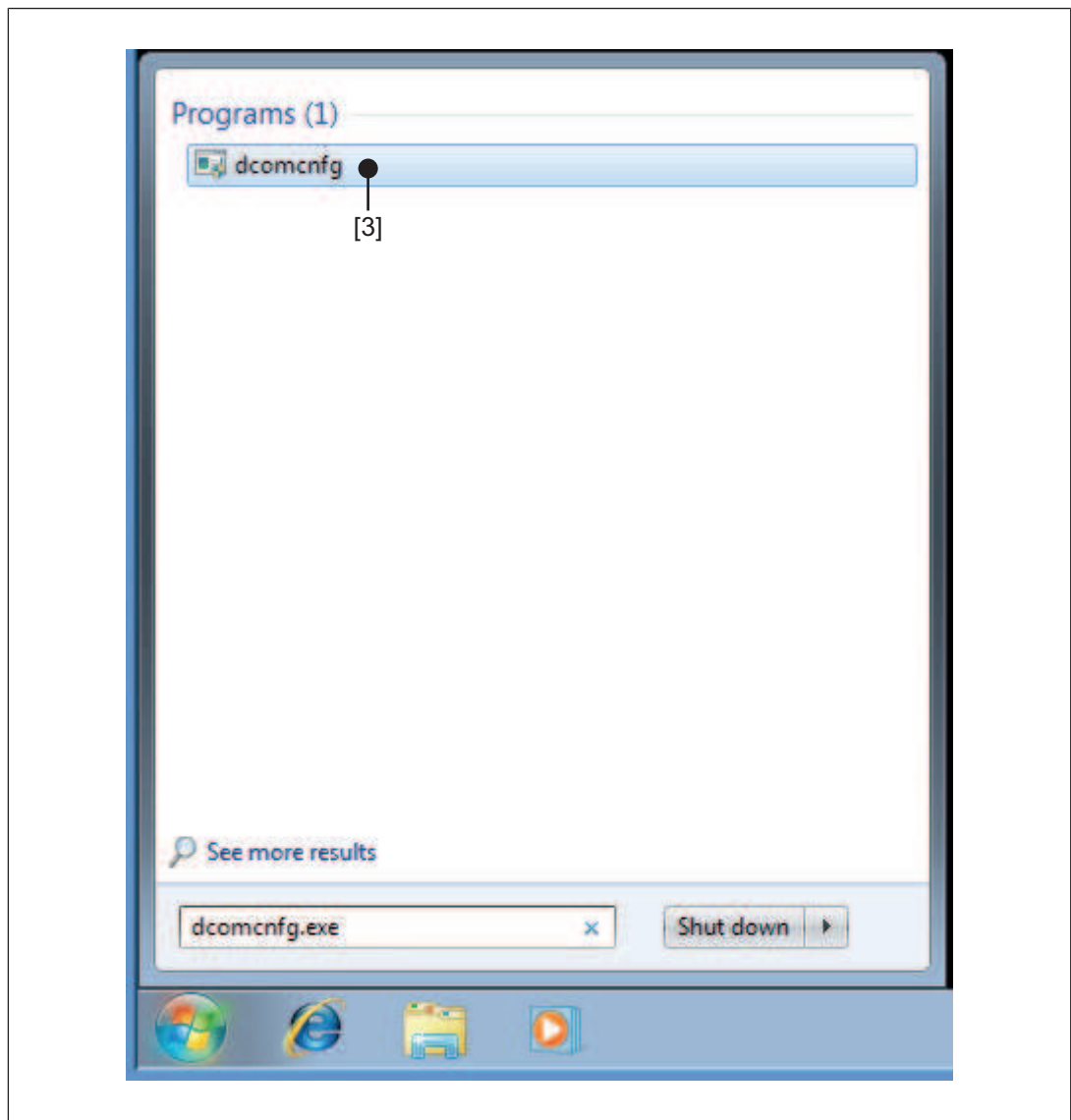
15.5.1 DCOM configuration

In the section below, the required settings and configurations are described using the dialogue boxes from Microsoft Windows 7. The dialogue boxes shown may vary depending on the Windows version and service pack. Further guidelines are available under Windows Help.

15.5.1.1 Start DCOM configuration



- Click the Windows Start symbol [1] and enter "dcomcnfg.exe" as a search criterion [2]. A window is opened, in which the program "dcomcnfg" is displayed as a search result.



- Select the program "dcomcnfg" [3] and confirm your selection.
The **Component Services** window opens.

Component services

Open the directory in the Navigator of the **Component Services** window (window's left-hand column)

Console Root / Component Services / Computers / My Computer.

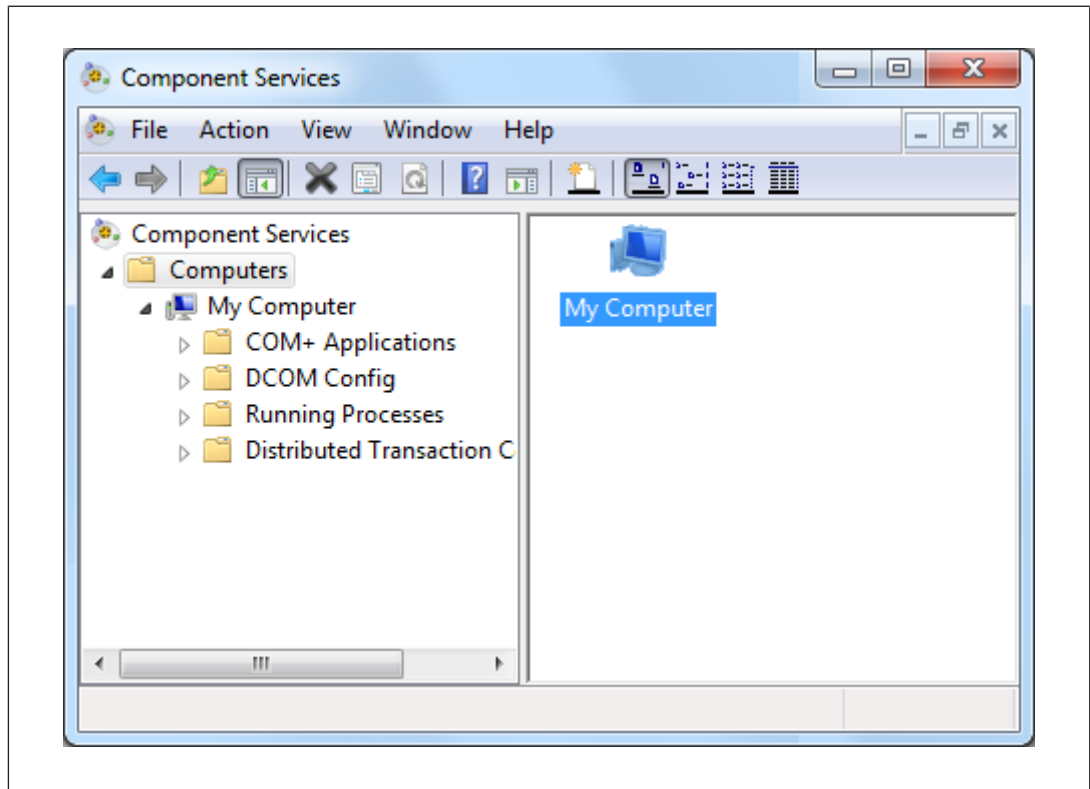


Fig.: **Component Services** window

- For further details of the configuration, please refer to the sections:
 - DCOM configuration for the OPC Server and OPC Clients [641]
 - DCOM configuration for the OPC Server [647]
 - OPCEnum configuration for the OPC Server [656]

15.5.1.2 DCOM configuration for the OPC Server and OPC Clients

The following communication and security settings must be made on all **all** PCs (OPC Server, OPC Clients) involved in data exchange. These settings are called the "general DCOM configuration" and specify the access that other PCs may have to a PC.

- ▶ Open the **Component Services** window
(see [Start DCOM configuration](#) [638]).
- ▶ Right-click on **My Computer** and select **Properties**.
The **My Computer Properties** window opens.

Default Properties tab

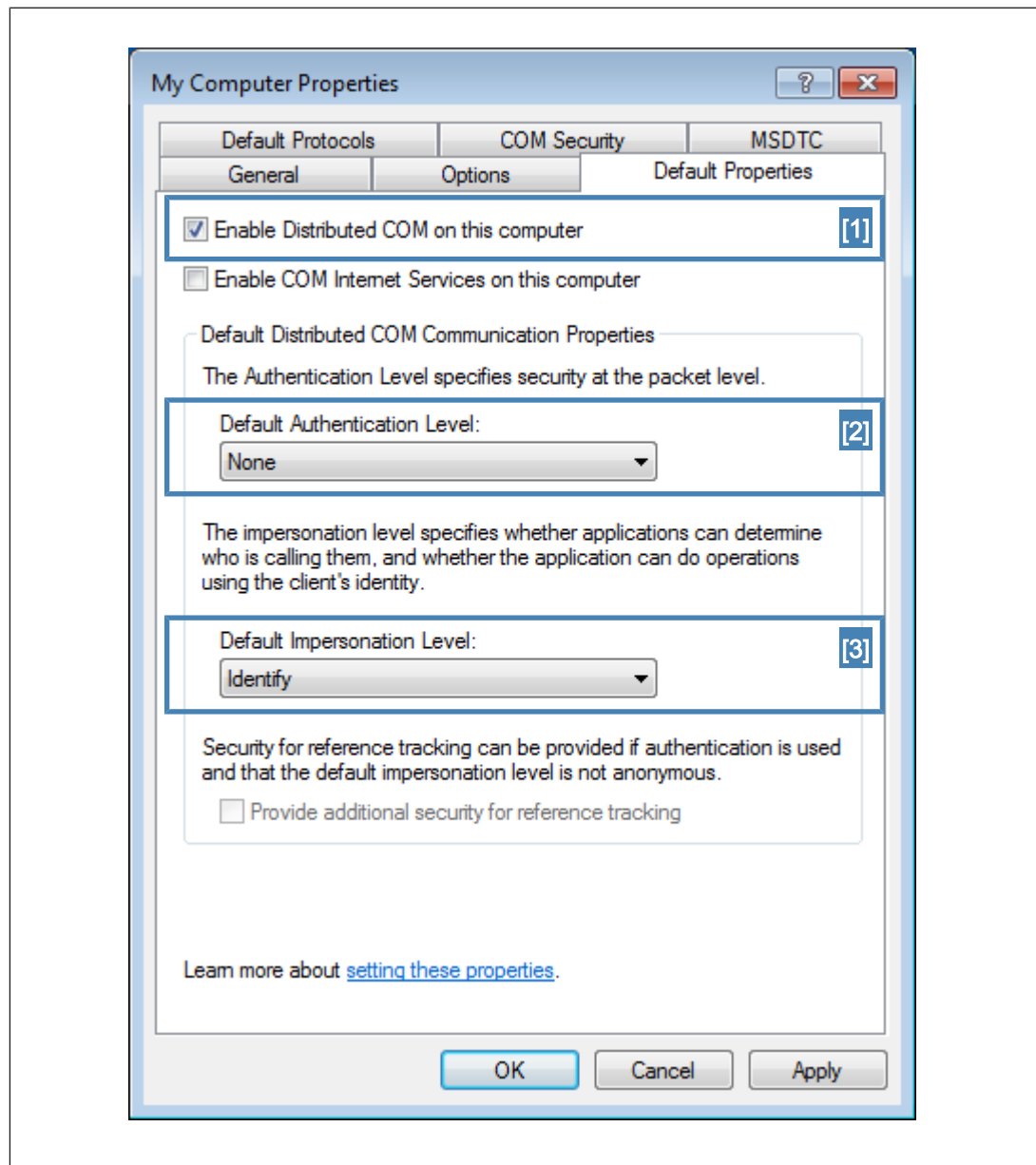


Fig.: **Default Properties** tab

- ▶ Select **Enable DCOM (Distributed COM) on this Computer** at [1].
- ▶ Select **None** under **Default Authentication Level** [2].

- Select **Accept Identity** or **Identify** under **Default Impersonation Level** [3].

Default Protocols tab

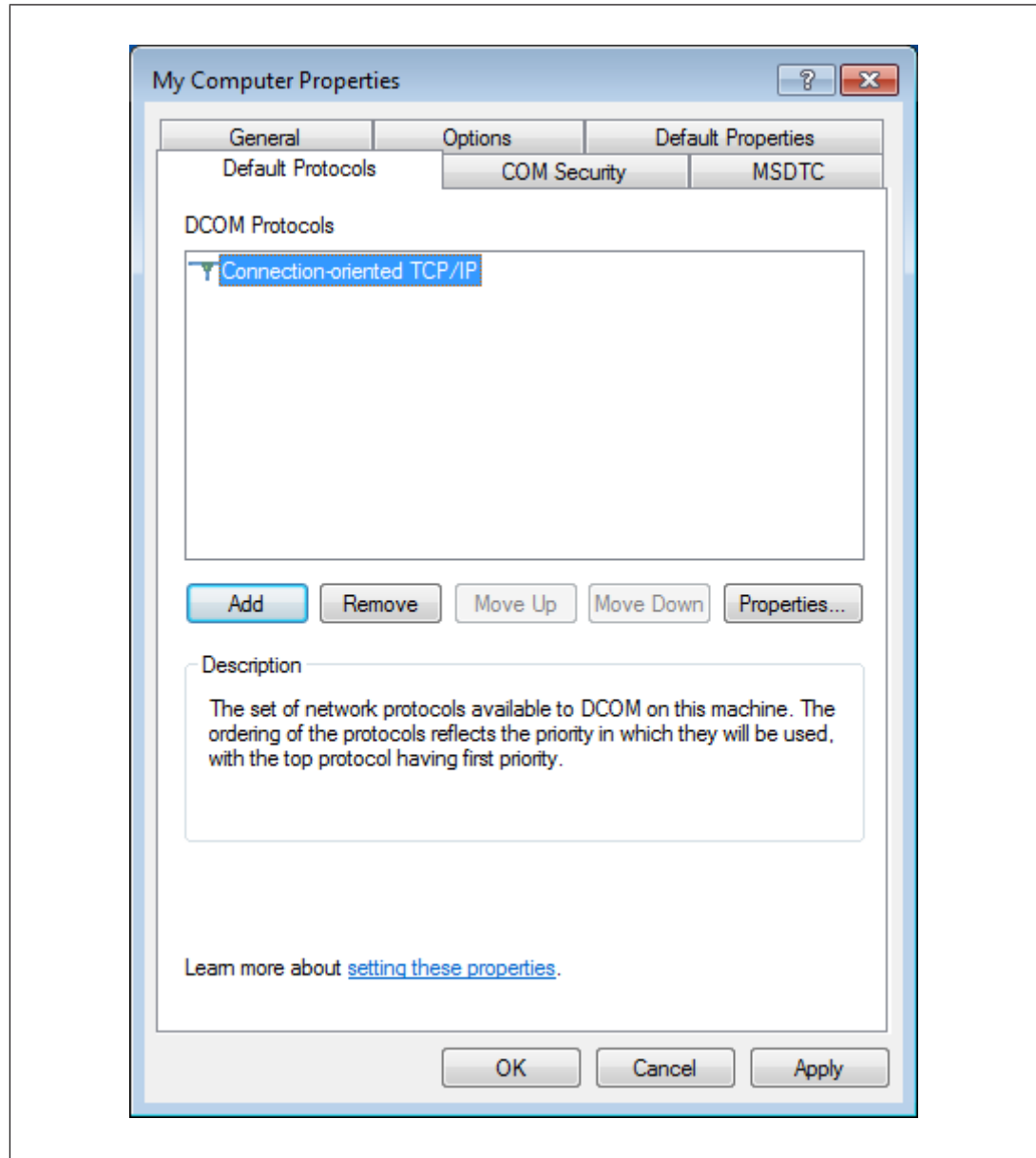


Fig.: **Default Protocols** tab

- If the connection protocol **Connection-oriented TCP/IP** does not appear in the list in the **Default Protocols** tab, you can add it as follows:
 - Click on the **Add** button
The **Select DCOM protocol** dialogue box opens.
 - Under **Protocol Sequence**, select the **Connection-oriented TCP/IP** protocol and click **OK**.
The **Default Protocols** tab is reactivated.
- Highlight the **Connection-oriented TCP/IP** protocol and move it to the top of the list.
This will speed up the connection setup.

- Remove any unneeded protocol entries, as these will cause delays when establishing the connection.

COM Security tab

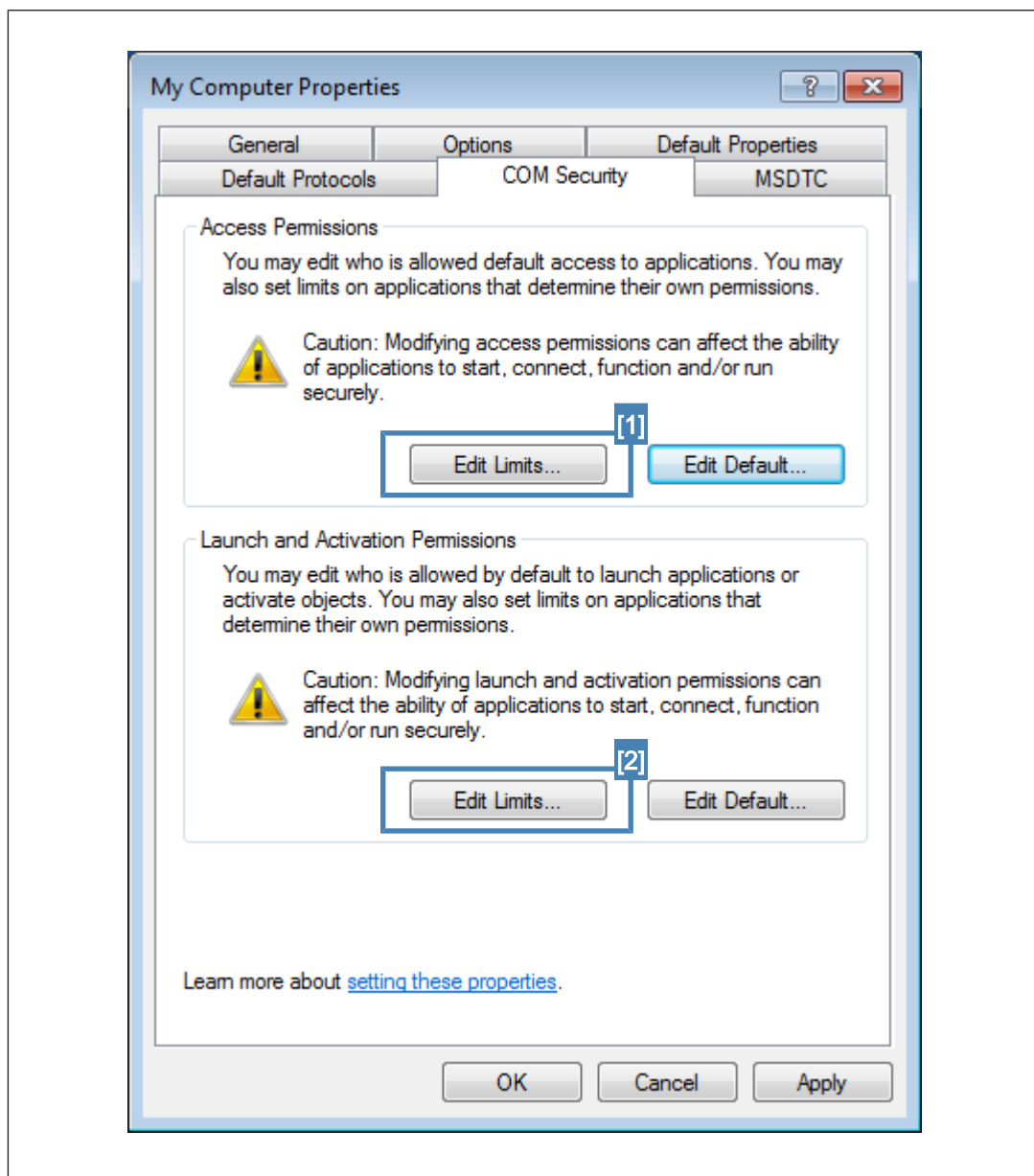


Fig.: **COM Security** tab

- Under **Access Permissions**, click on **Edit Limits** [1]
The **Access Permissions** window opens.

COM Security window -> Access Permissions

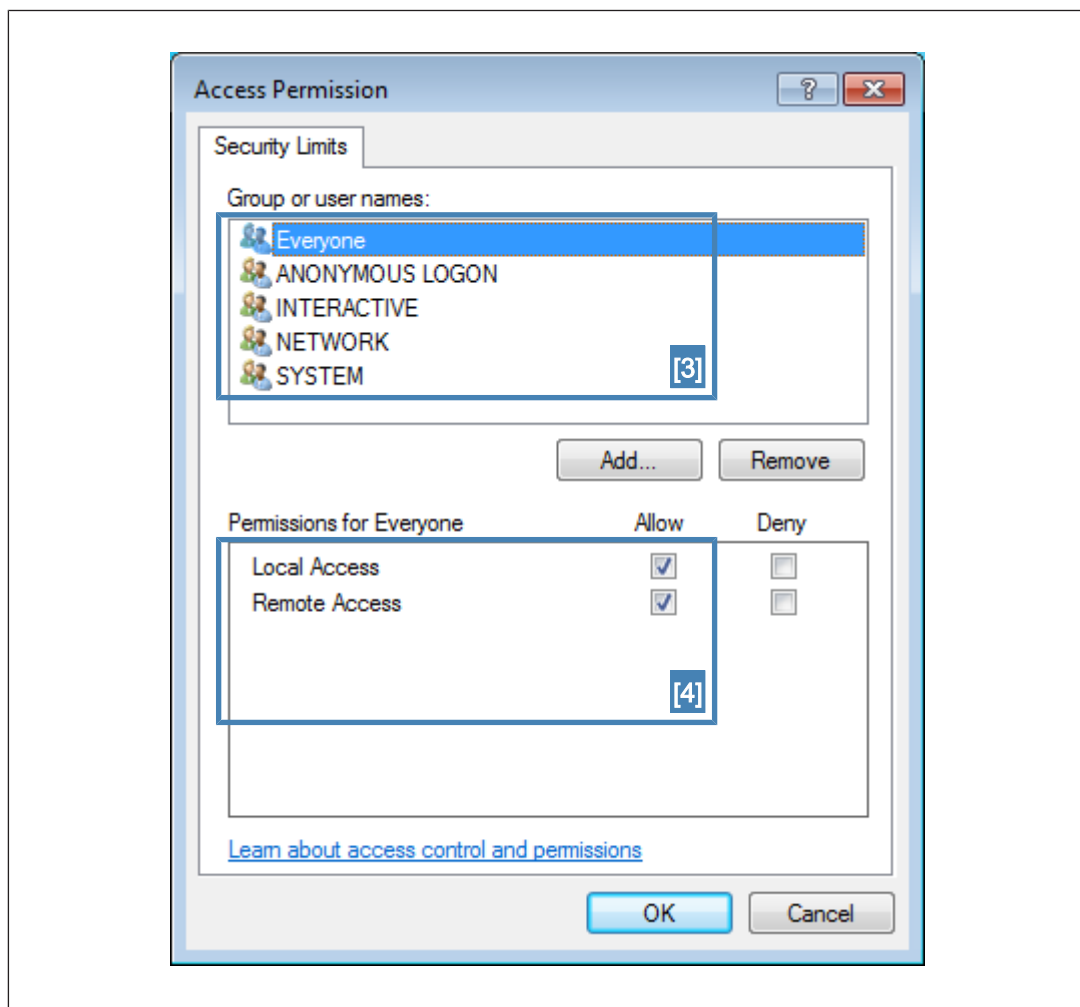


Fig.: **COM Security window -> Access Permissions**

- Check whether all the following group or user names [3] are available:

- ANONYMOUS LOGIN
- INTERACTIVE
- Any
- NETWORK
- SYSTEM

- If one of the group or user names is not available, add it (**Add... - Advanced – Search Now**).

Please note the additional information for the NETWORK user (see [Special feature of the user "NETWORK" \[665\]](#)).

Then confirm with **OK**.

The **Access Permissions** window is reactivated.

- Select each user [3] individually and tick **Allow** for the permissions [4]
- Local access
- Remote access

- ▶ Click **OK**.
The **COM Security** tab is reactivated.
- ▶ On the **Com Security** tab, under **Start and Activation Permissions**, click on **Edit Limits** [2].
The **Start and Activation Permissions** window opens.

COM Security window -> Start and Activation Permissions

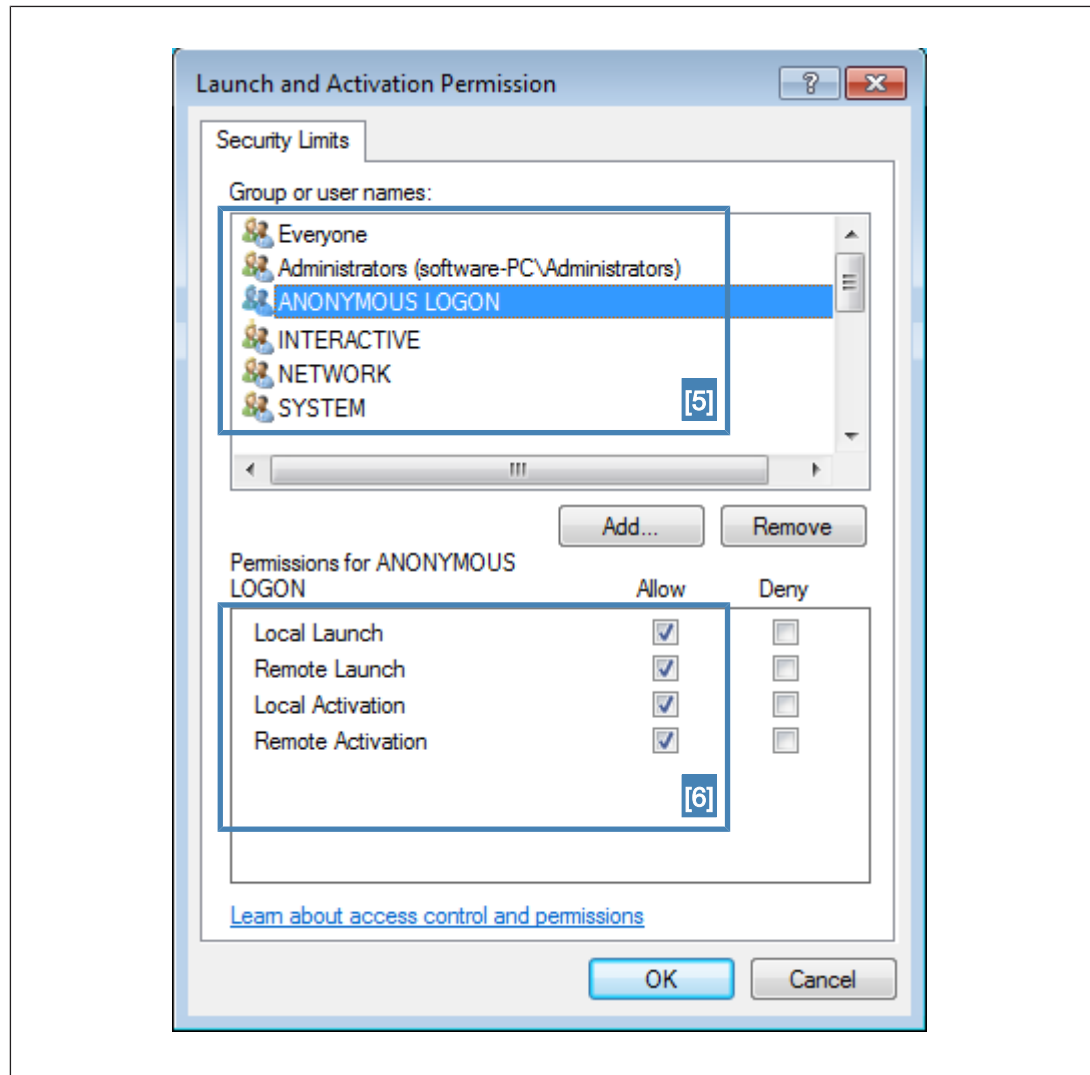


Fig.: **COM Security window -> Start and Activation Permissions**

- ▶ Check whether all the following group or user names [5] are available:
 - Administrators (...)
 - ANONYMOUS login
 - INTERACTIVE
 - Any
 - NETWORK
 - SYSTEM

- ▶ If the "Administrators" user is not available, add it. To add the "Administrators" user, the search path must be adapted (see [Special feature of the user "Administrators" \[664\]](#)).
- ▶ If one of the other group or user names is not available, you can add it (**Add... - Advanced – Search Now**)

Please note the additional information for the NETWORK user (see [Special feature of the user "NETWORK" \[665\]](#))

Then confirm with **OK**.

The **Launch Permission** window is reactivated.

- ▶ Select each user [5] individually and tick **Allow** for the permissions [6]
 - Local launch
 - Remote launch
 - Local activation
 - Remote activation.

- ▶ Click **OK**.

The **COM Security** tab is reactivated.

- ▶ Click **OK** to save all the settings and close the **My Computer Properties** window.

This completes the general DCOM configuration for this workstation.

- ▶ Close the **Component Services** window.

The settings will take effect the next time the PC is restarted.

15.5.1.3 DCOM configuration for the OPCServer

The following settings are made automatically when the OPC Server is installed. Check these settings in the event of an error and modify the stated settings if necessary. These settings are only relevant for the PC containing the OPC Server.

- ▶ Open the **Component Services** window.
(see [Start DCOM configuration](#) [638])
- ▶ Open the **DCOM Configuration** directory [1] in the **Component Services** window.

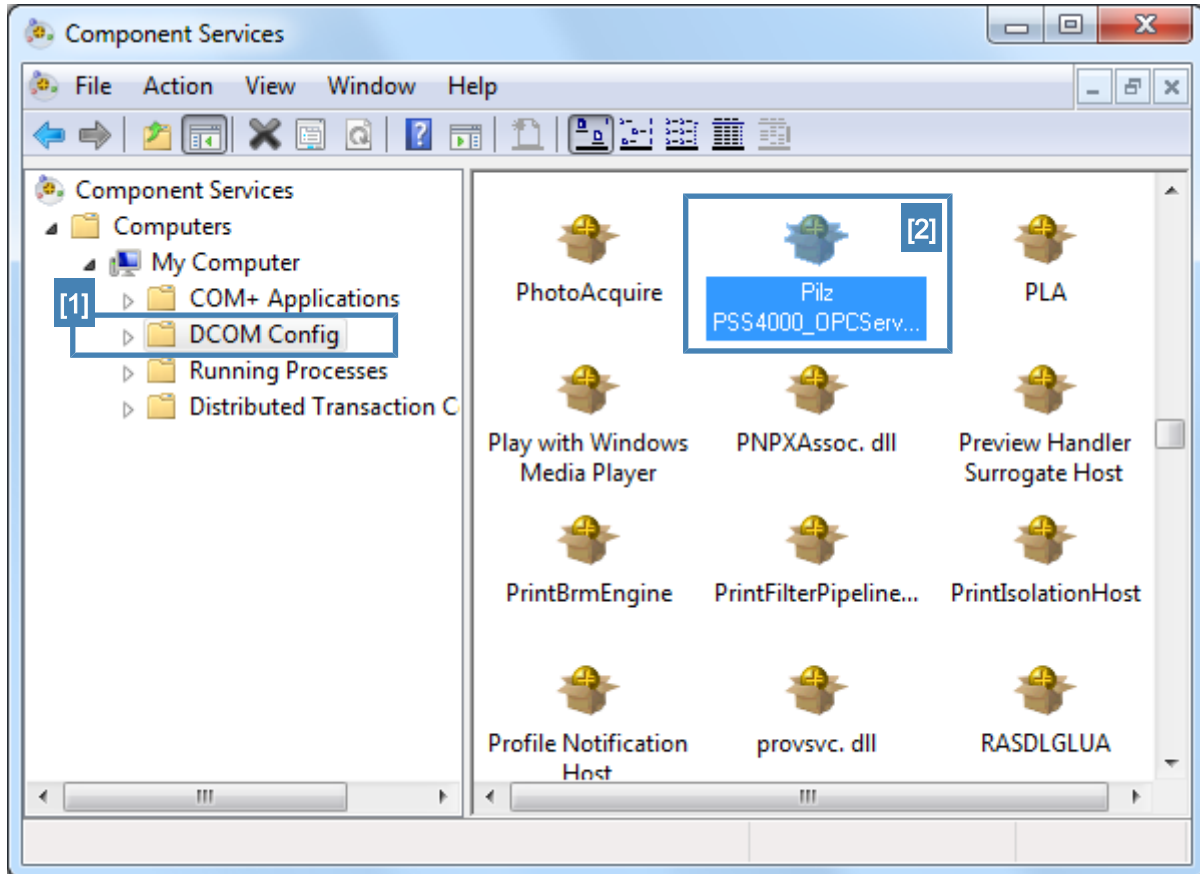


Fig.: **Component Services** window (DCOM configuration on the OPC Server)

- ▶ Right-click on the application **Pilz PSS4000_OPCTServerDA.DA** [2] and select **Properties**.
The **Pilz PSS4000_OPCTServerDA.DA Properties** window opens.

General tab

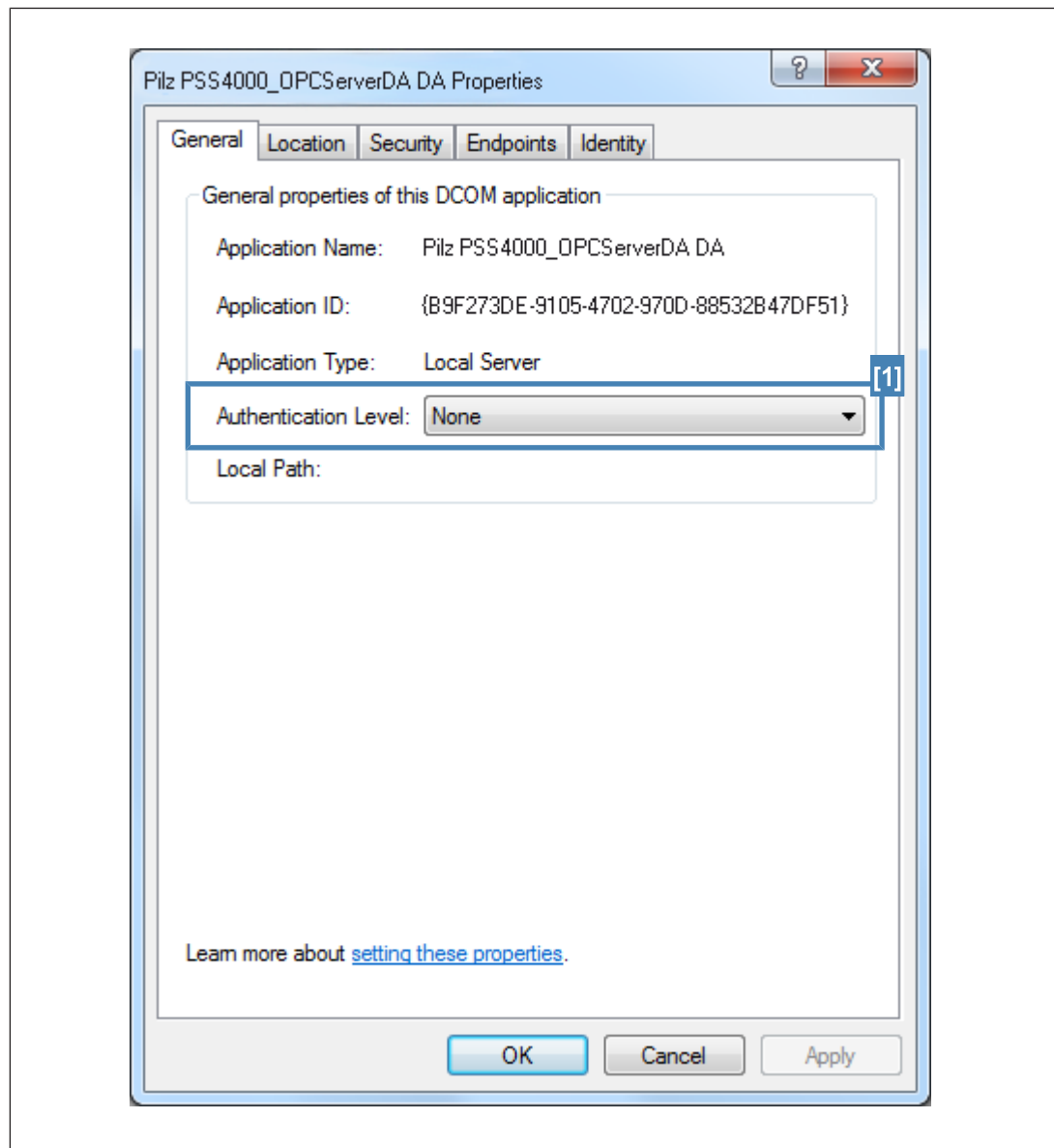


Fig.: **General** tab (DCOM configuration on the OPC Server)

- Select **None** under **Authentication Level** [1].

Location tab

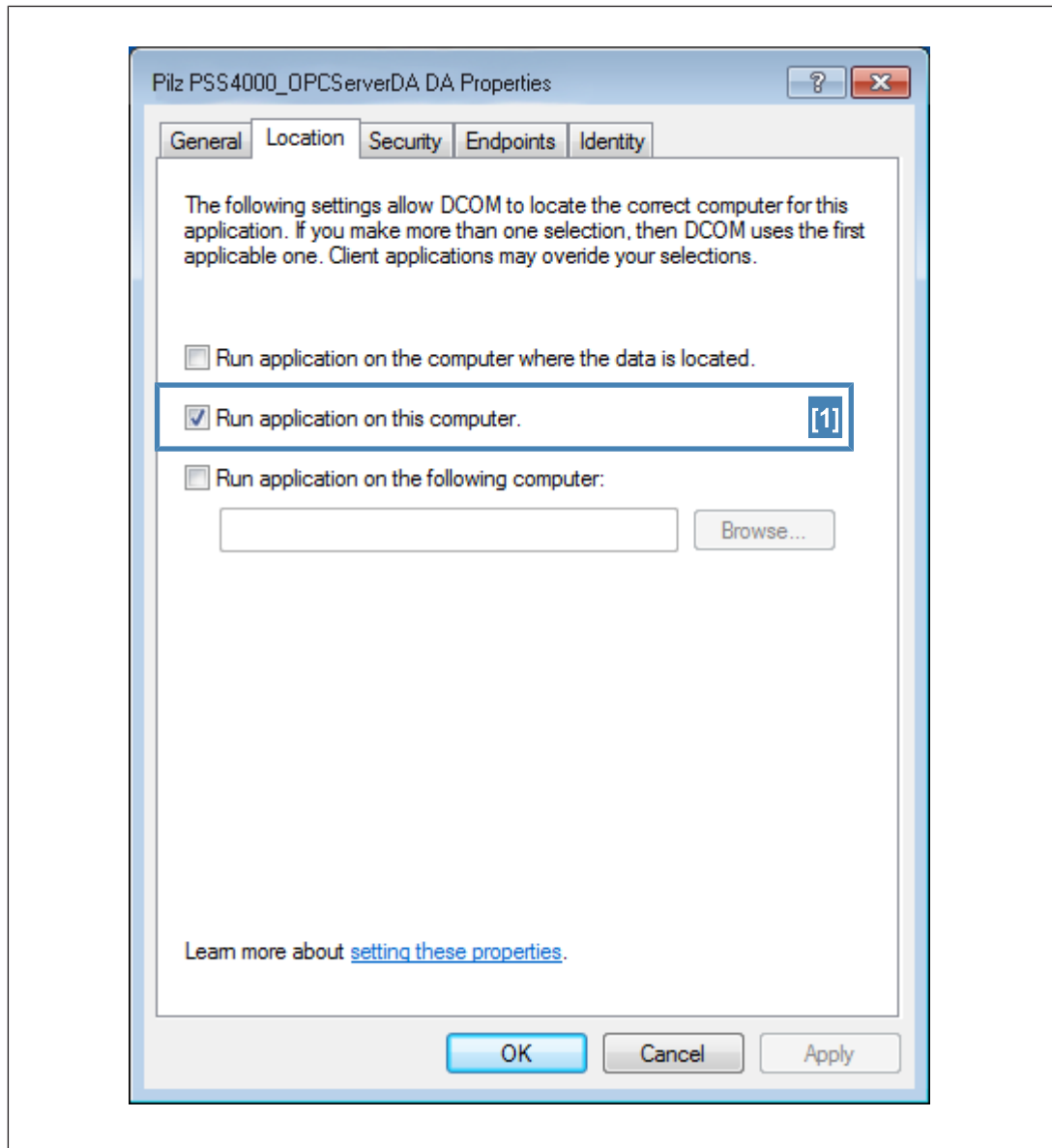


Fig.: **Location** tab (DCOM configuration on the OPC Server)

- Enable **Run application on this computer** [1].

Security tab

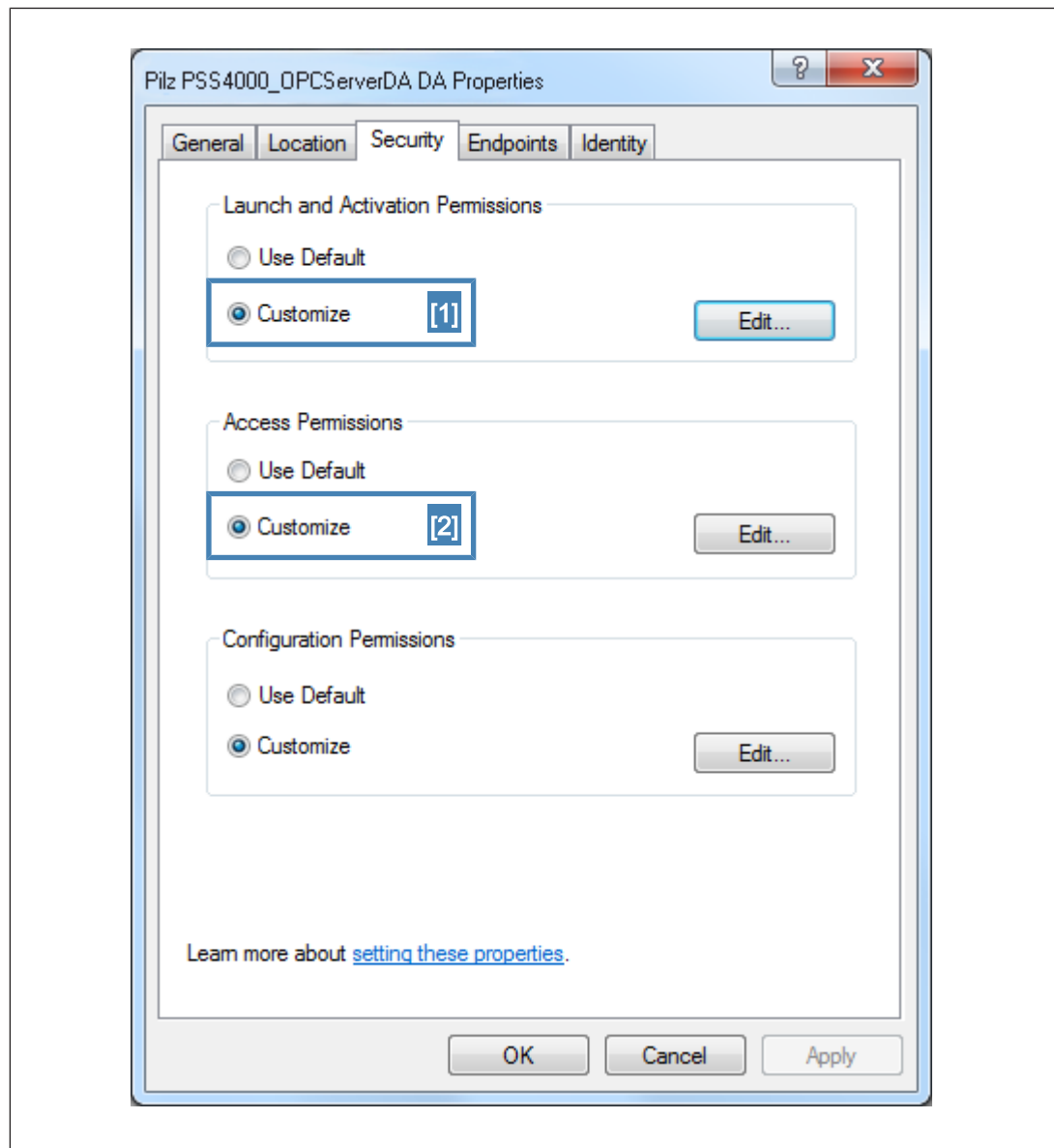


Fig.: **Security** tab (DCOM configuration on the OPC Server)

- ▶ Select **Customize** for **Launch and Activation Permissions** [1] and **Access Permissions** [2] and then click **Apply**.
- ▶ Click on **Edit** under **Launch and Activation Permissions** [1].
The **Launch and Activation Permissions** window is opened.

Security-> Launch and Activation Permissions window

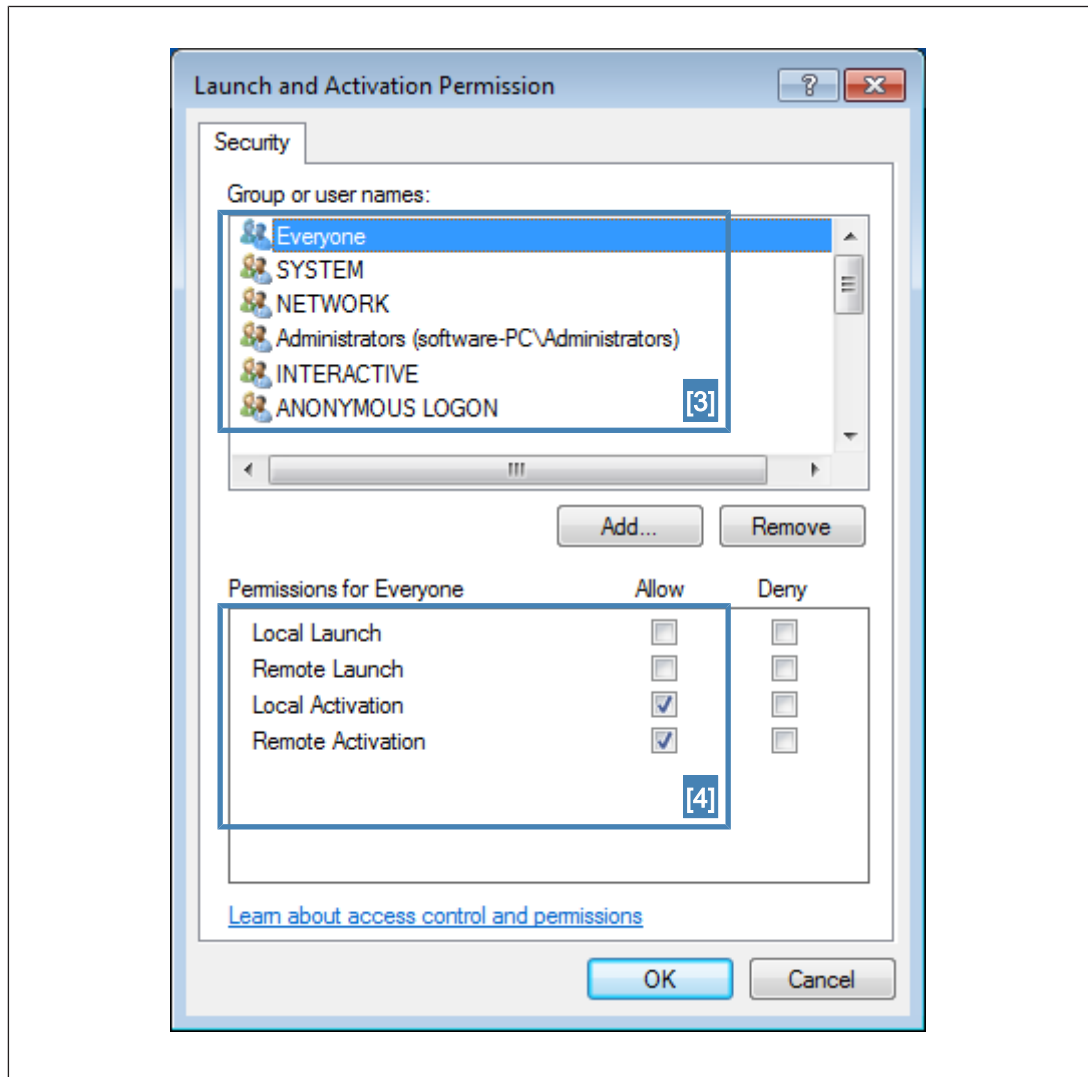


Fig.: **Launch and Activation Permissions window**

- ▶ Check whether all the following group or user names [3] are available:
 - Administrators (...)
 - ANONYMOUS LOGON
 - INTERACTIVE
 - Any
 - NETWORK
 - SYSTEM
 - ▶ If the "Administrators" user is not available, add it. To be able to add the "Administrators" user, the search path has to be adapted (see [Special feature of the user "Administrators" \[664\]](#)).
 - ▶ If one of the other group or user names is not available, add it also (**Add... - Advanced – Search Now**).
- Please note the additional information for the NETWORK user (see [Special feature of the user "NETWORK" \[665\]](#)).

Then confirm with **OK**.

The **Launch and Activation Permissions** window is reactivated.

- ▶ Select each user [3] individually and tick **Allow** for the permissions [4].
 - Local activation
 - Remote activation
- ▶ Click **OK**.

The **Security** tab is reactivated.

- ▶ Click on **Edit** under **Access Permissions** [2] on the **Security** tab.
The **Access Permissions** window opens.

Security -> Access Permissions window

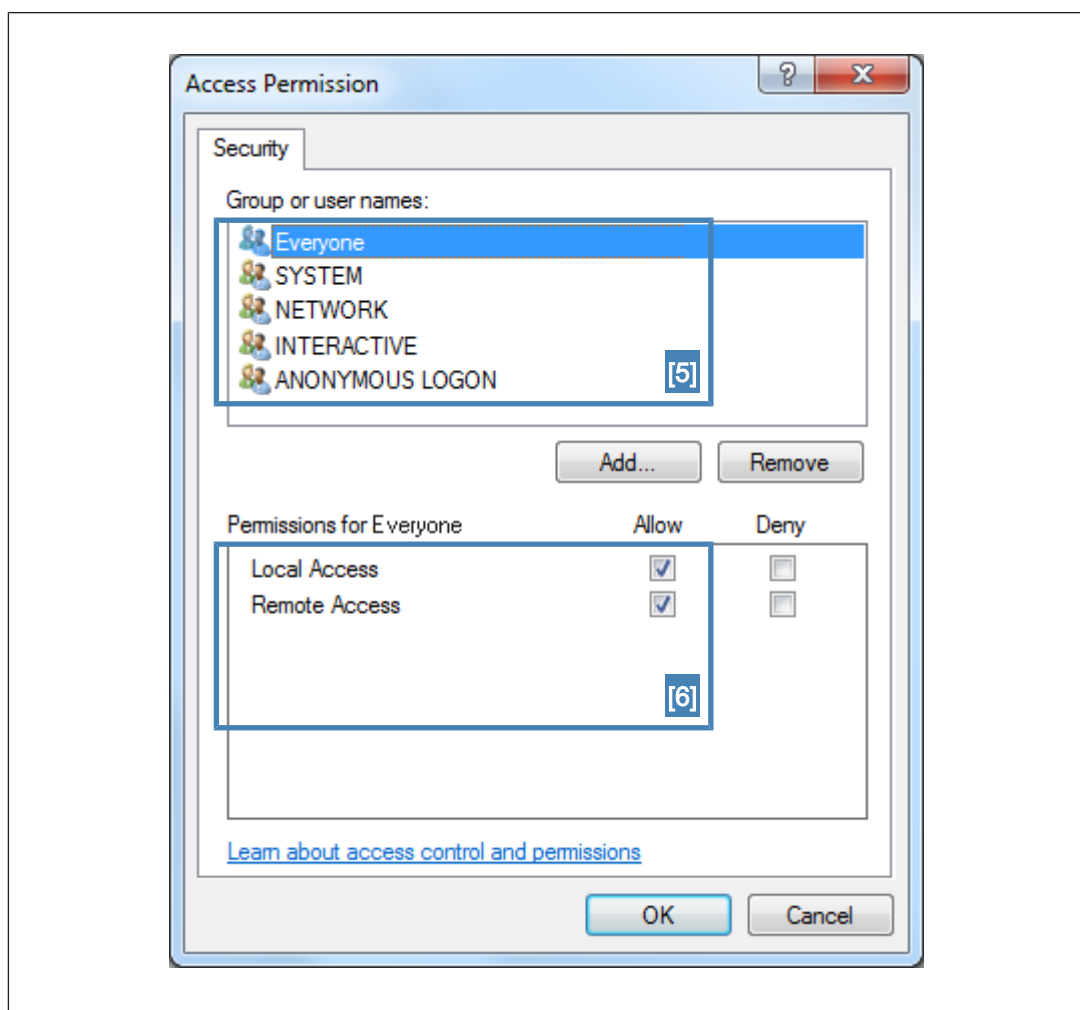


Fig.: **Access Permissions** window

- ▶ Check whether all the following group or user names [5] are available:
 - ANONYMOUS LOGON
 - INTERACTIVE
 - Any
 - NETWORK

– SYSTEM

- ▶ If one of the group or user names is not available, add it (**Add... - Advanced – Search Now**).

Please note the additional information for the NETWORK user Special feature of the "NETWORK" user.

Then confirm with **OK**.

The **Access Permissions** window is reactivated.

- ▶ Select each user [5] individually and tick **Allow** for the permissions [6].

- Local Access

- Remote Access

- ▶ Click **OK**.

The **Safety** tab is reactivated.

Identity tab

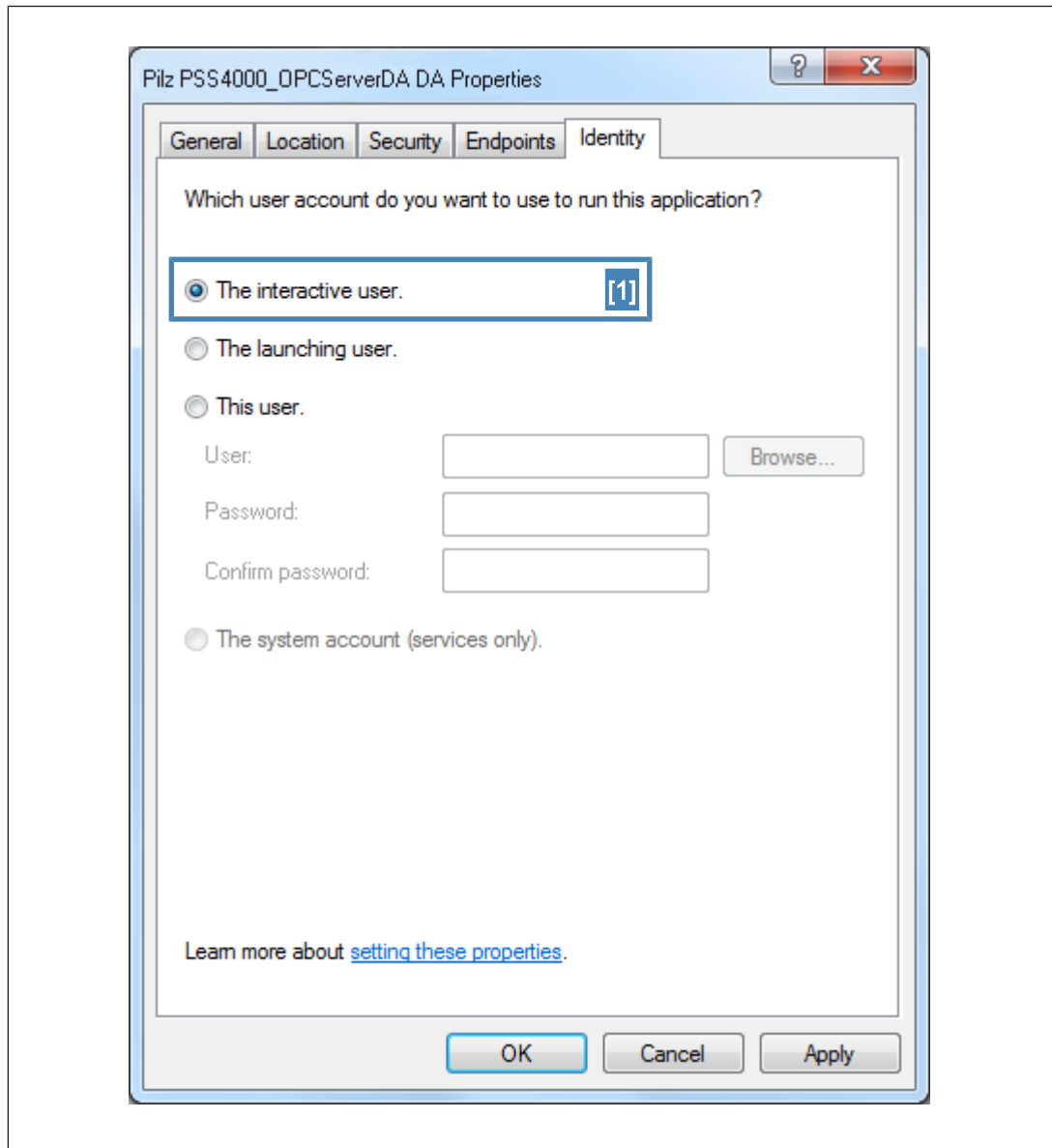


Fig.: **Identity** tab (DCOM configuration on the OPC Server)

- Select **Interactive User** [1].

Endpoints tab

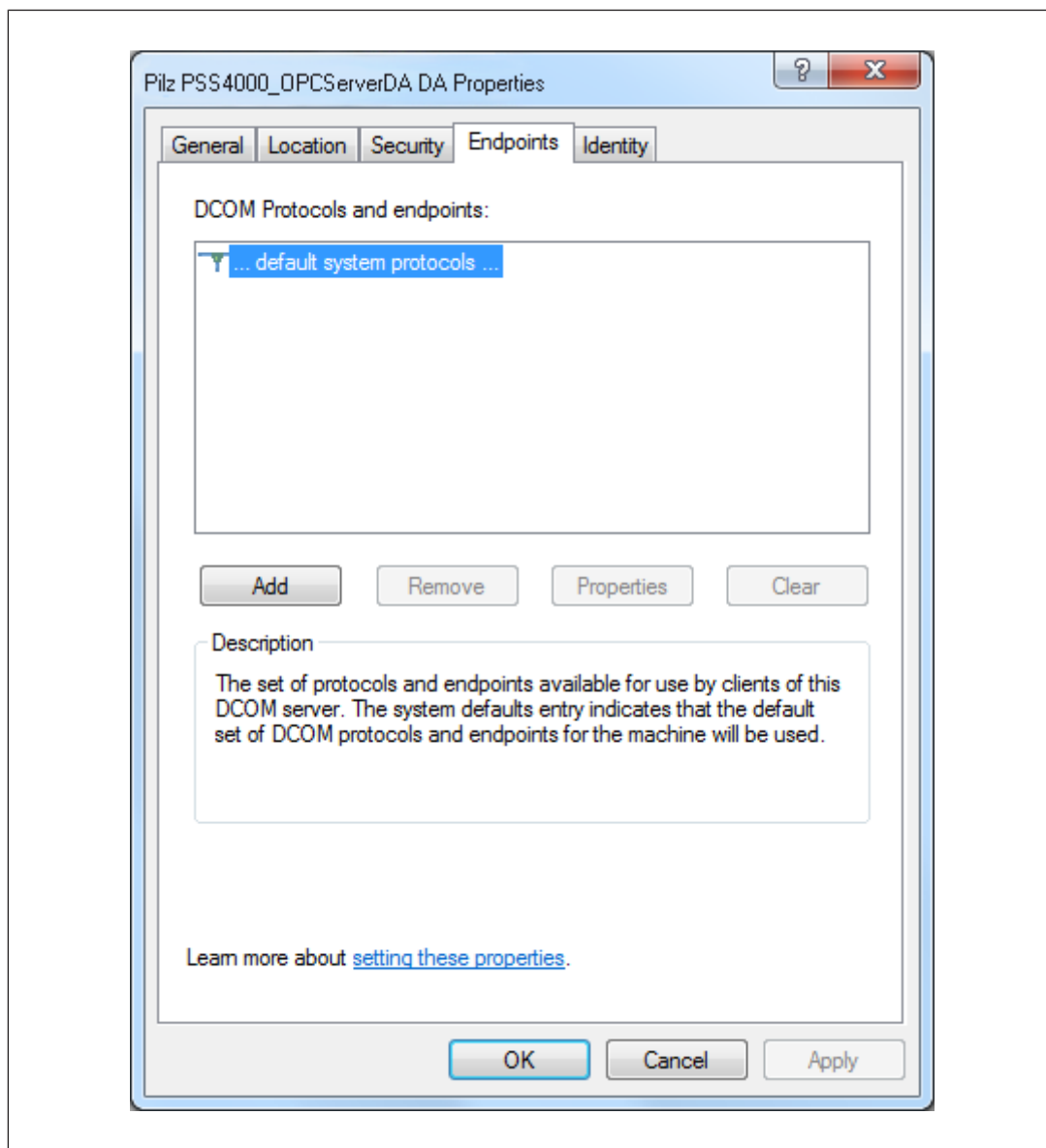


Fig.: **Endpoints** tab (connection protocol on the OPC Server)

- ▶ If **Default system protocols** is not available in the list, add the connection protocol as follows:
 - Click on **Add** and select the **Connection-oriented TCP/IP** protocol under **Protocol Sequence**.
 - Click **OK**.
The **Endpoints** tab is reactivated.
 - ▶ Click **OK** to save all the settings and close the **PVIS OPC Server DA Properties** window.
- This completes the DCOM configuration for the OPC Server.

15.5.1.4 OPCEnum configuration for the OPC Server

The following settings are only relevant for the PC containing the OPC Server.

- Open the **Component Services** window.
(see [Start DCOM configuration](#) [638]).
- Open the **DCOM Configuration** directory [1] in the **Component Services** window.
You may receive several DCOM configuration warnings, but you can ignore all of them for this configuration. Each time click **No**.

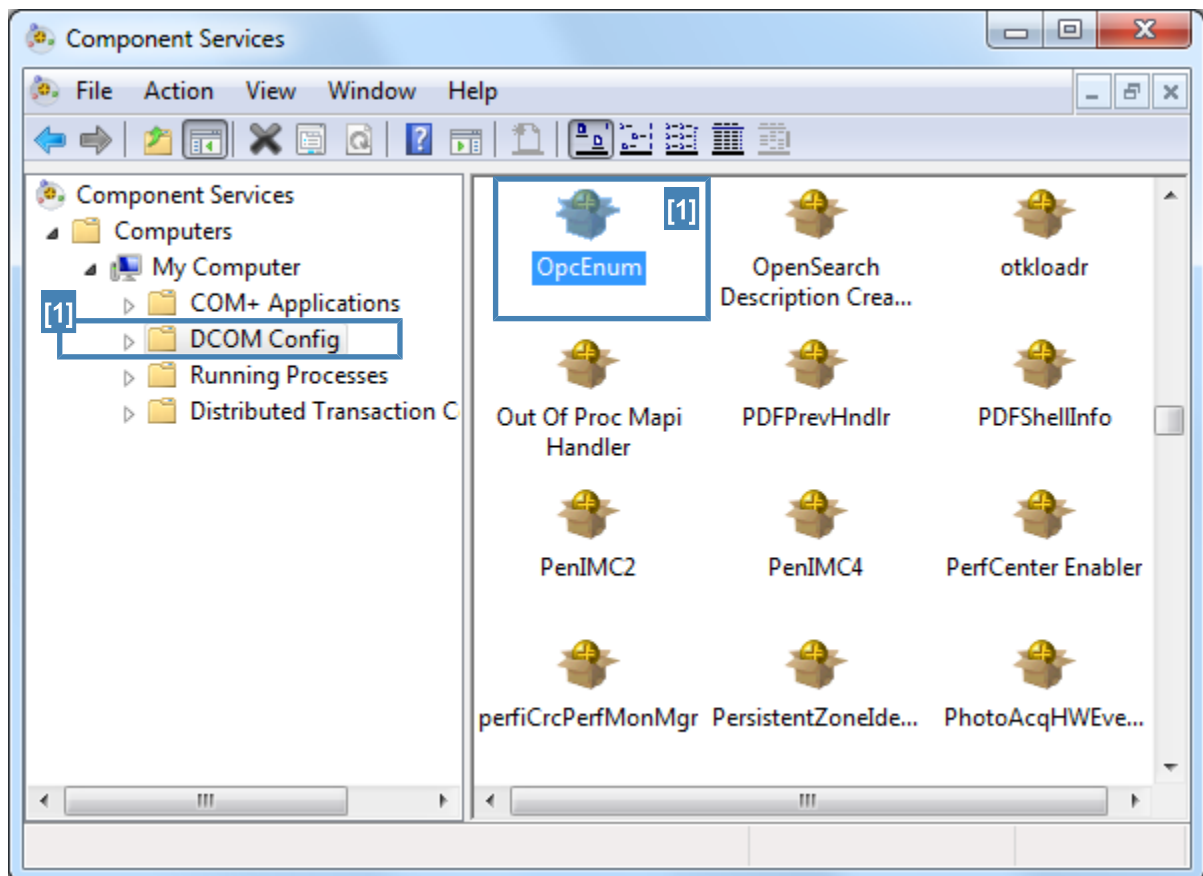


Fig.: **Component Services** window (DCOM configuration for OPCEnum of the OPC Server)

- Right-click on the **OpcEnum** application [2] and select **Properties**.
The **OpcEnum Properties** window opens.

General tab

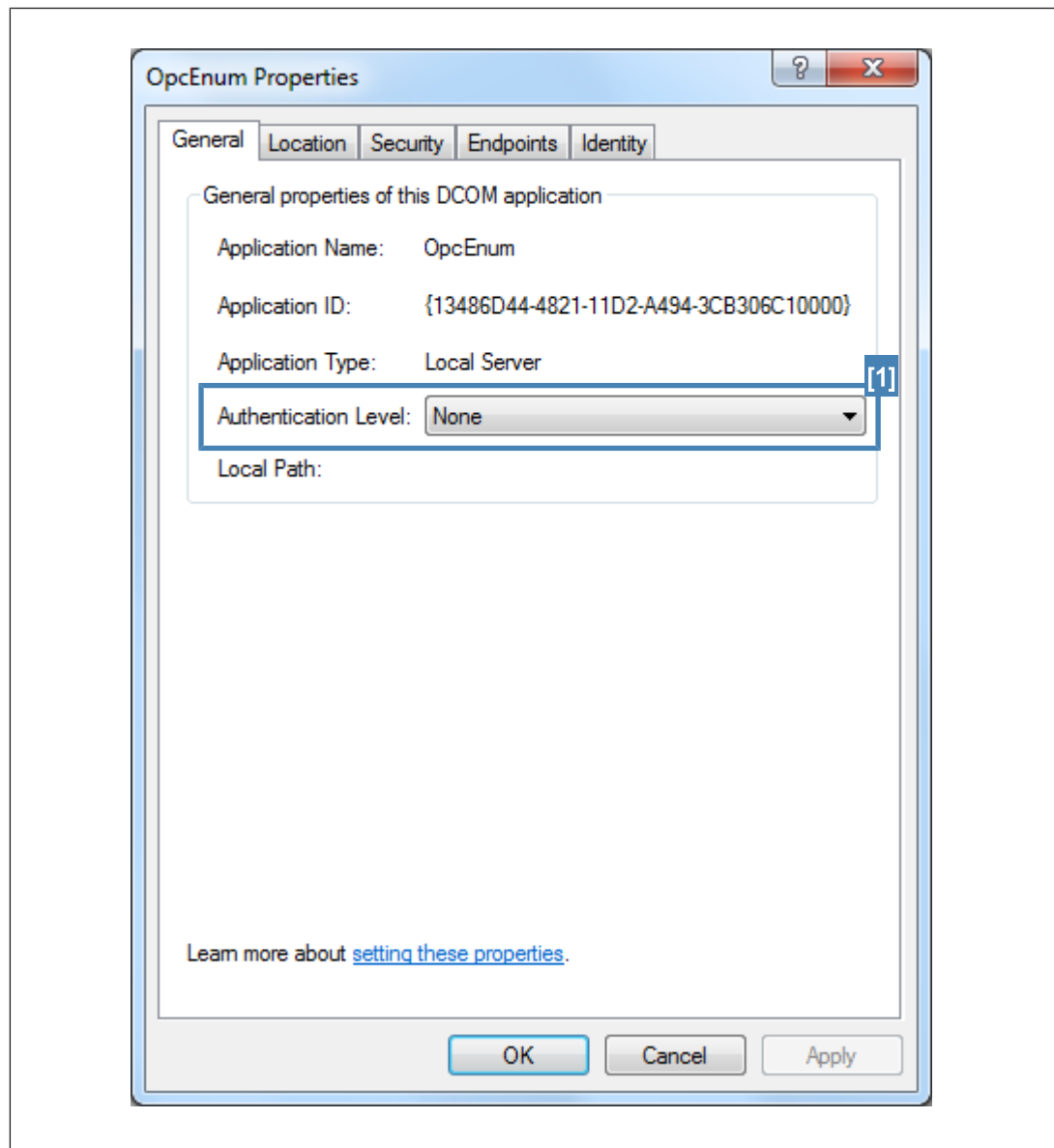


Fig.: **General** tab (DCOM configuration for OPCEnum of the OPC Server)

- Select **None** under **Authentication Level** [1].

Location tab

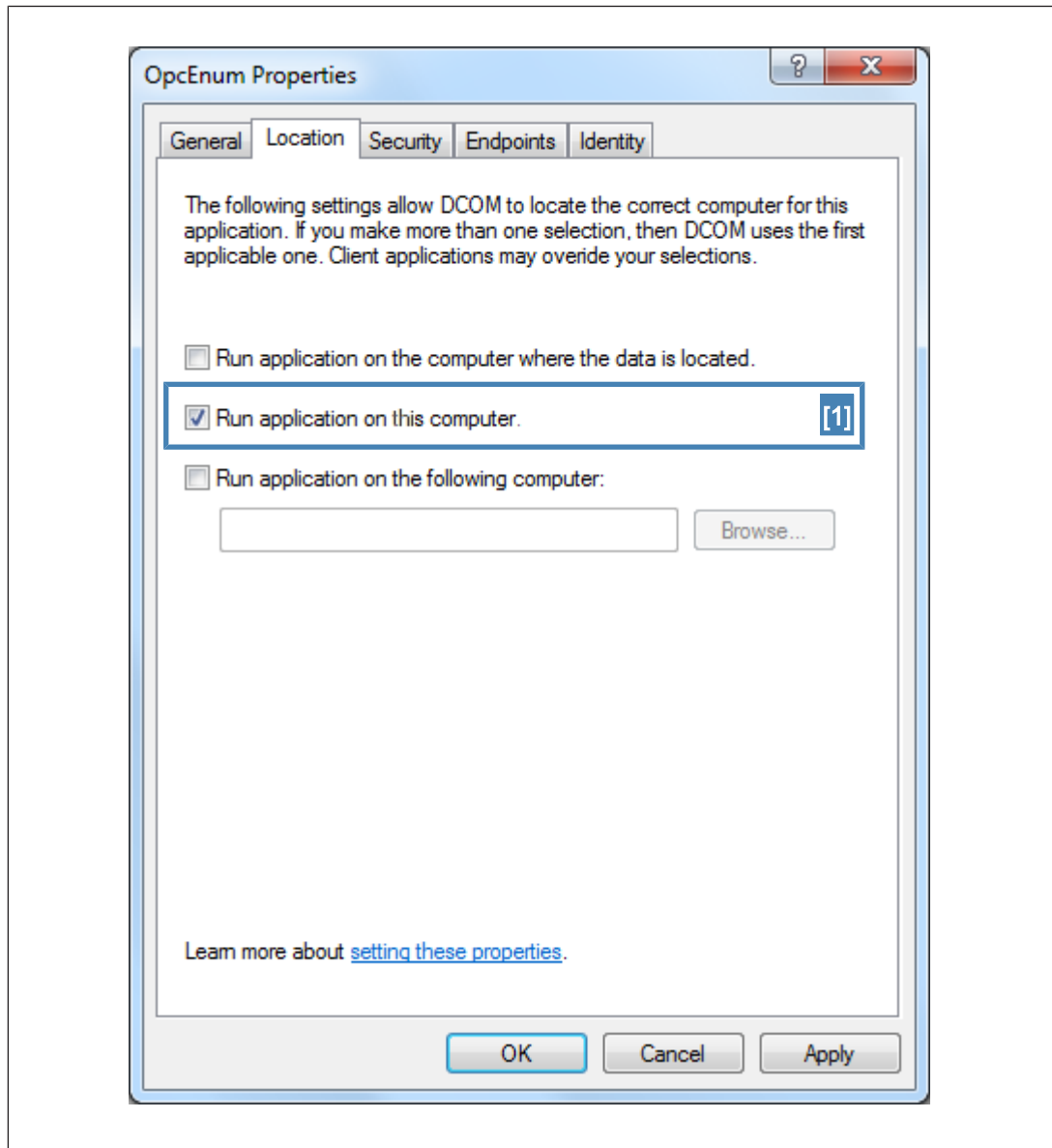


Fig.: **Location** tab (DCOM configuration for OPCEnum of the OPC Server)

- Enable **Run application on this Computer** [1].

Security tab

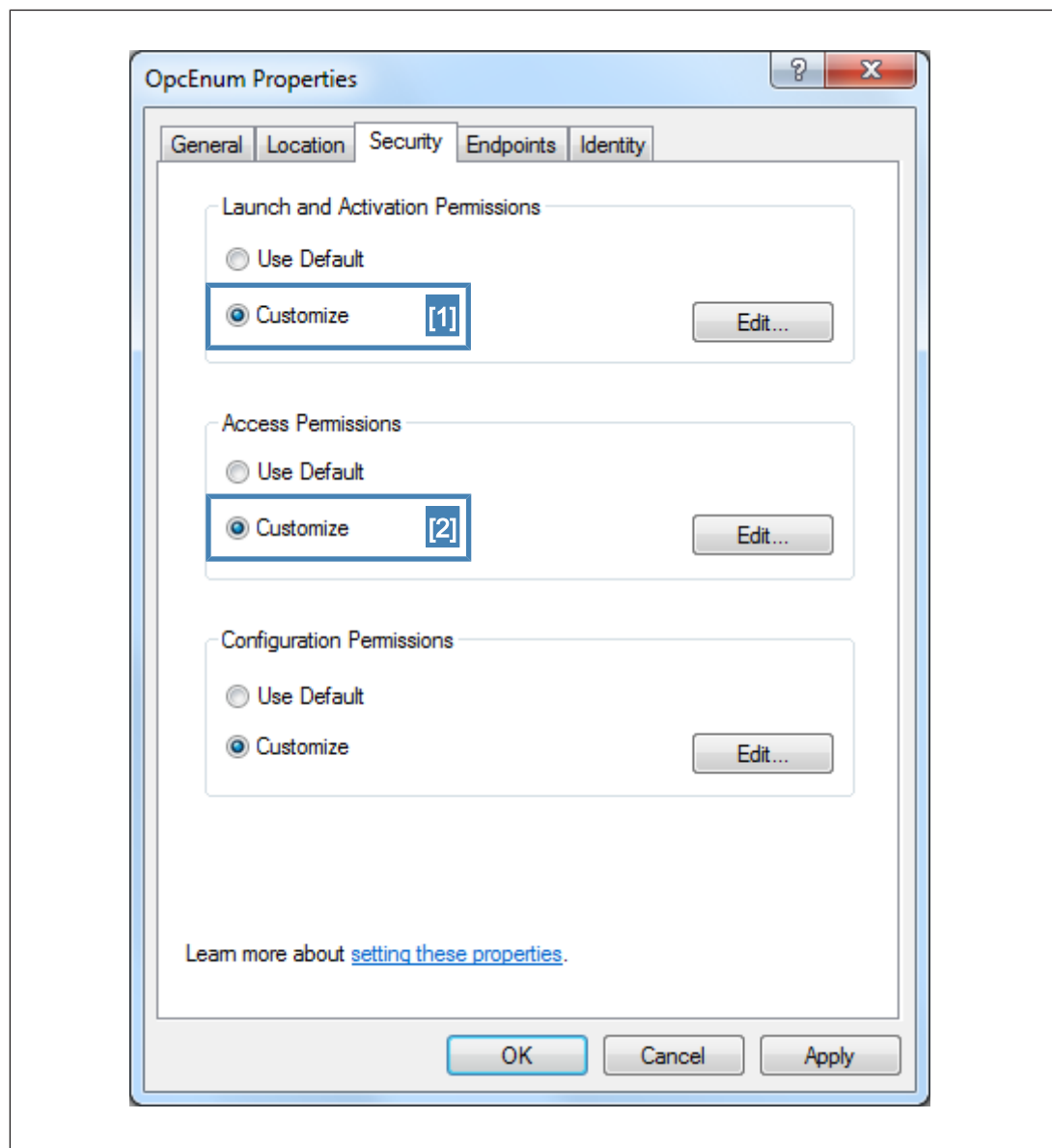


Fig.: **Security** tab (DCOM configuration for OPCEnum of the OPC Server)

- ▶ Select **Customize** for **Launch and Activation Permissions** [1] and **Access Permissions** [2] and then click **Apply**.
- ▶ Click on **Edit** in the **Launch and Activation Permissions** field [1].
The **Launch and Activation Permissions** window is opened.

Security window -> Launch and Activation Permissions

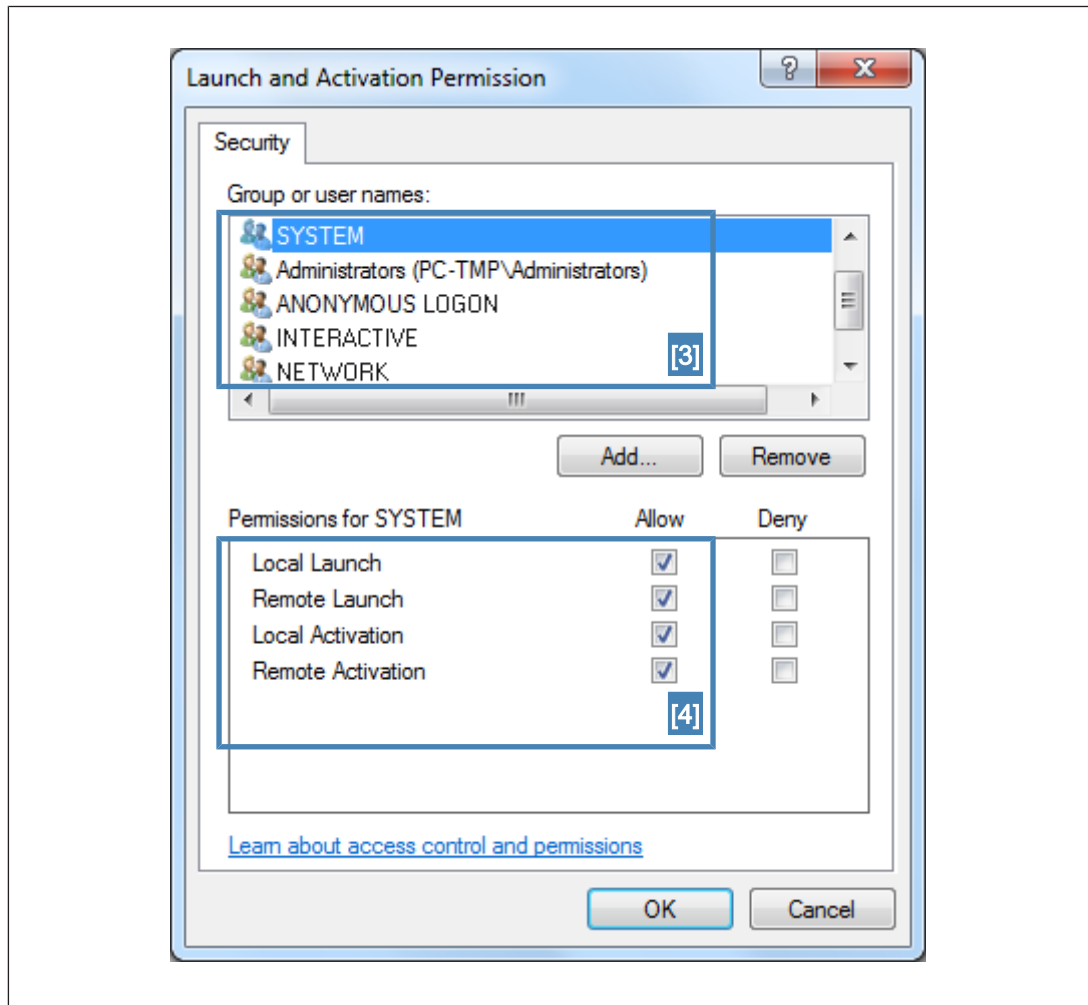


Fig.: **Launch and Activation Permissions** window (COM security for OPCEnum)

- Check whether all the following group or user names [3] are available:
 - Administrators (...)
 - ANONYMOUS LOGIN
 - INTERACTIVE
 - Any
 - NETWORK
 - SYSTEM
- If the "Administrators" user is not available, add it. To be able to add the "Administrators" user, the search path must be adapted (see Special feature of the user "Administrators").
- If one of the other group or user names is not available, you can add it (**Add... - Advanced – Search Now**).

Please note the additional information for the NETWORK user (see Special feature of the "NETWORK" user).

Then confirm with **OK**.

The **Launch and Activation Permissions** window is reactivated.

- ▶ Select each user [3] individually and tick **Allow** for the permissions [4]
 - Local launch
 - Remote launch
 - Local activation
 - Remote activation
- ▶ Click **OK**.
The **Security** tab is reactivated.
- ▶ On the **Security** tab, under **Access Permissions** [2], click on **Edit**.
The **Access Permission** window opens.

Security window -> Access Permissions

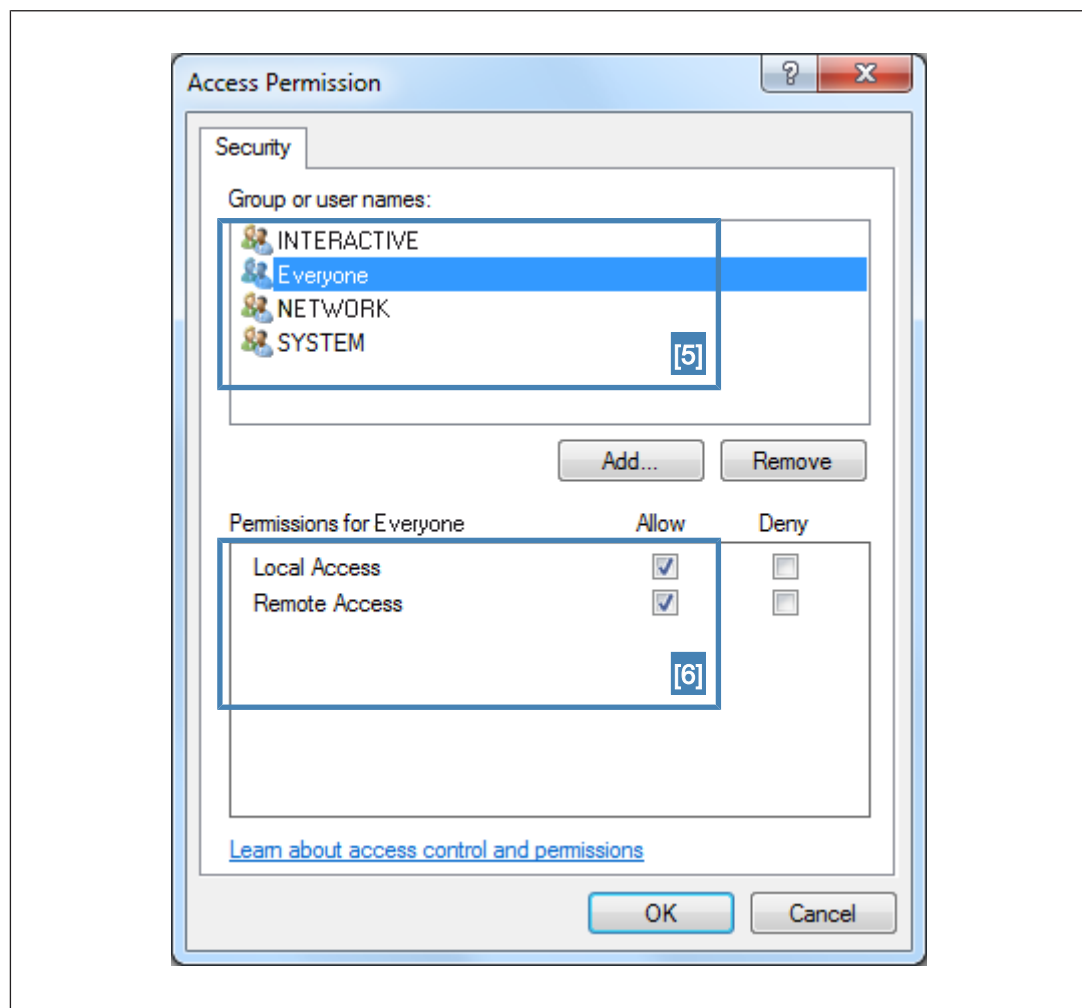


Fig.: **Access Permissions** window (COM Security for OPCEnum)

- ▶ Check whether all the following group or user names [5] are available:
 - INTERACTIVE
 - Any
 - NETWORK
 - SYSTEM

- If one of the group or user names is not available, add it (**Add... - Advanced – Search Now**).

Please note the additional information for the NETWORK user (see Special feature of the "NETWORK" user).

Then confirm with **OK**.

The **Access Permission** window is reactivated.

- Select each user [5] individually and tick **Allow** for the permissions [6]
 - Local Access
 - Remote Access
- Click **OK**.
The **Security** tab is reactivated.

Identity tab

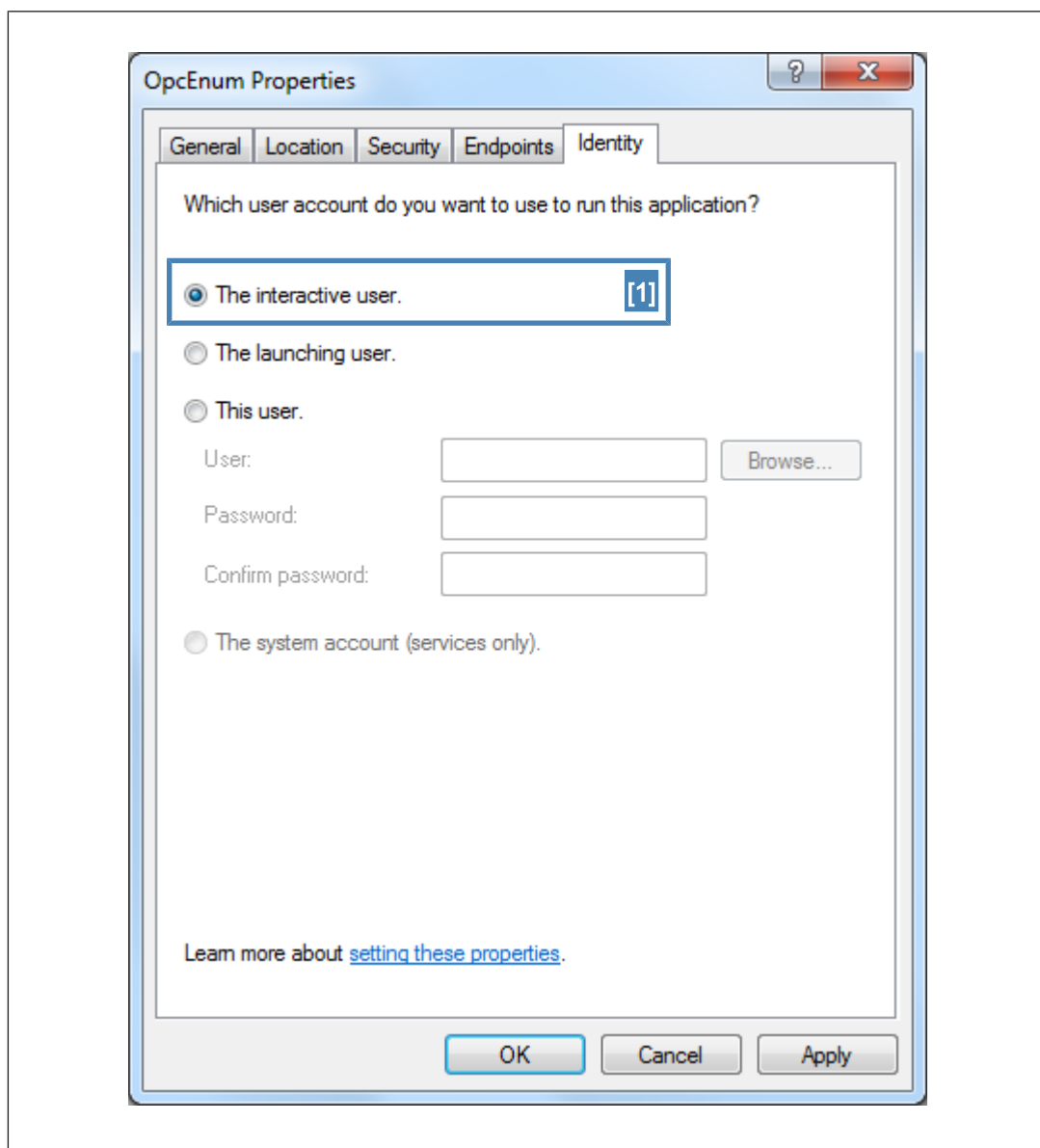


Fig.: **Identity** tab (COM Security for OPCEnum)

- Select **Interactive User** [1].

Endpoints tab

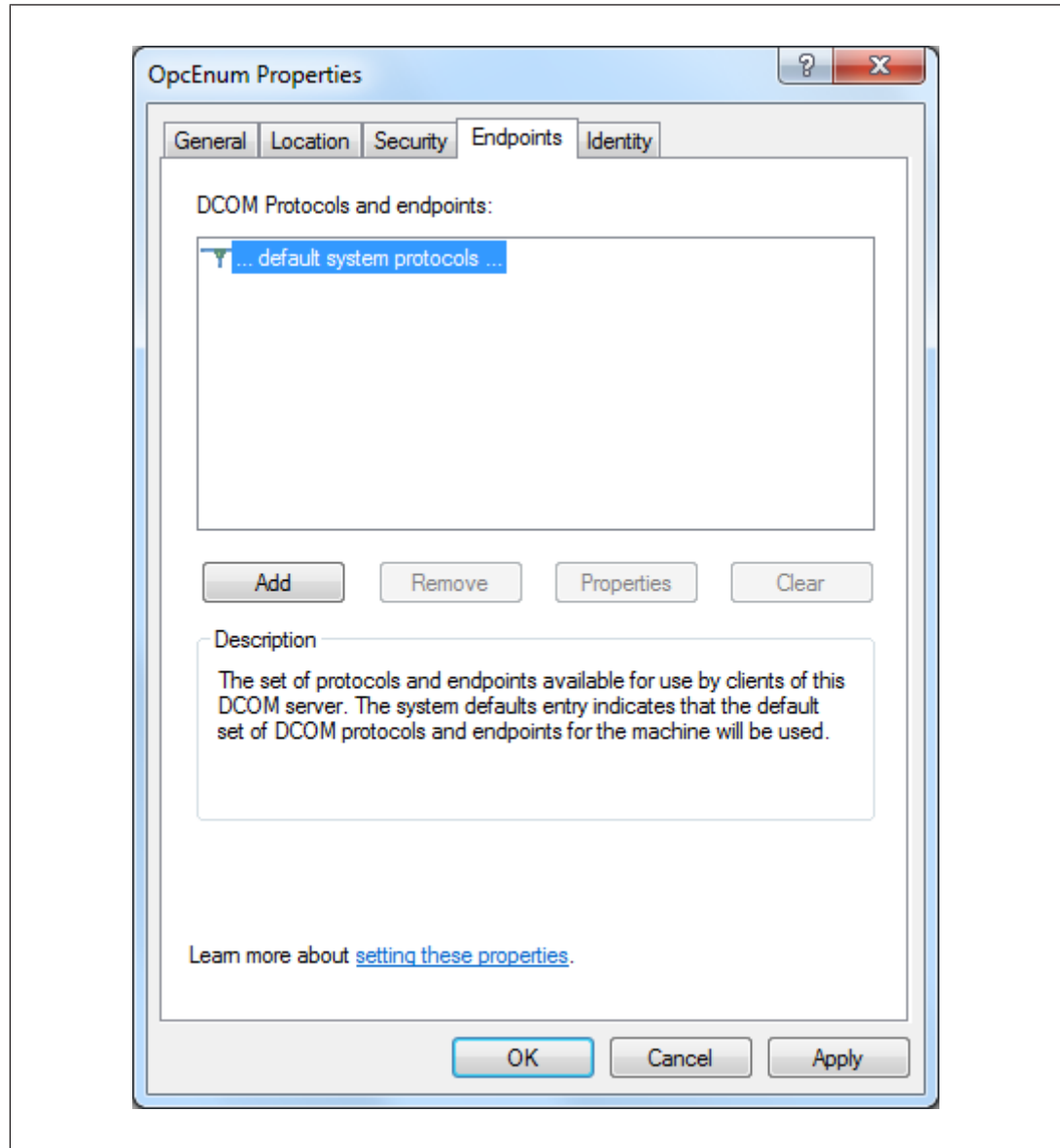


Fig.: **Endpoints** tab (connection protocols for OPCEnum)

- If **Default system protocols** is not available in the list, add the connection protocol as follows:
 - Click on **Add** and select the **Connection-oriented TCP/IP** protocol under **Protocol Sequence**.
 - Click **OK**.
The **Endpoints** tab is reactivated.
 - Click **OK** to save all the settings and close the **OpcEnum Properties** window.
This completes the OPCEnum configuration for the OPC Server.
 - Close the **Component Services** window.
- The settings will take effect the next time the PC is restarted.

15.5.1.5 Special feature of the user "Administrators"

To be able to add the "Administrators" user to the group or user names of a permission, the search path must first be adapted:

- ▶ In the relevant launch permissions window click **Add...** and then **Paths...**
- ▶ Select the PC on which you are performing the DCOM configuration (in this example "PC-TMP" [1]) as a (search) path and confirm with **OK**.
- ▶ Enter "Administrators" [2] as an object name and click **OK**.

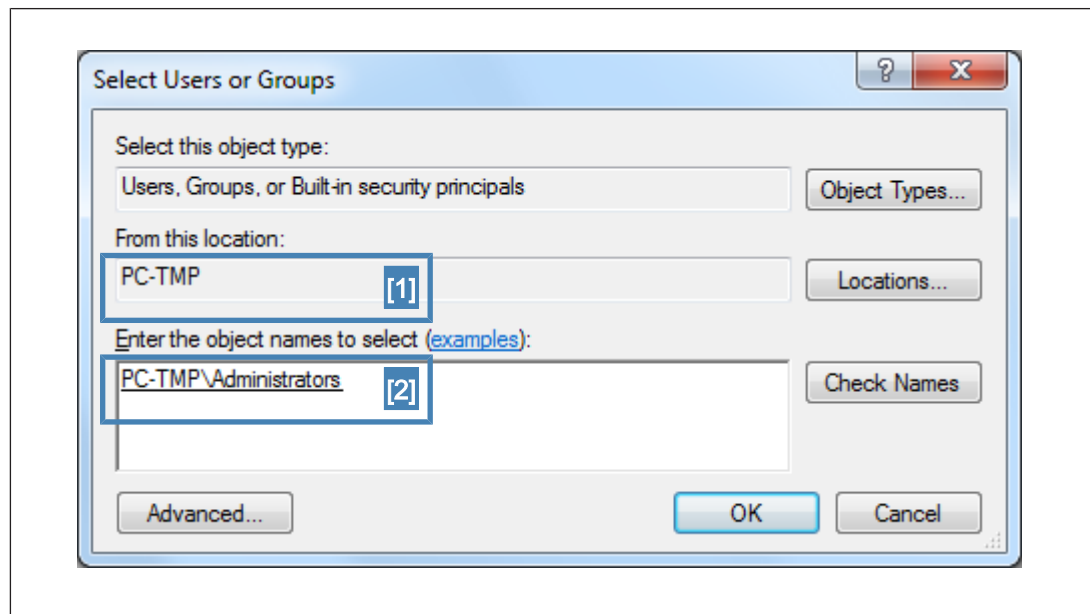


Fig.: "Select Users or Groups" window

15.5.1.6 Special feature of the user "NETWORK"

If you want to add the NETWORK user in one of the permission windows (e. g. Launch Permission window, Access Permission window), you may receive several users for selection when entering the user name "Network". In this case, select the "Network" user.

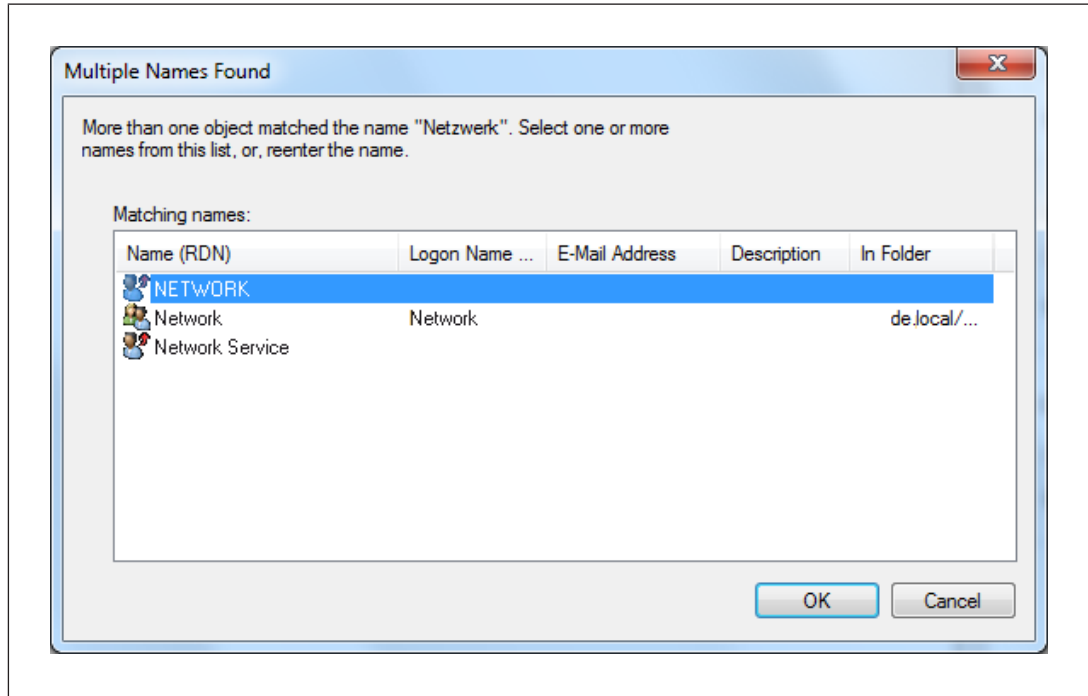


Fig.: Select NETWORK user

15.5.2 Firewall configuration

A firewall or any other security mechanisms on the PC and network may prevent the OPC Server from communicating correctly with the connected devices. In this case it will be necessary to adjust the security settings and firewall settings. Current documentation on the required security settings is available on the website for the OPC Foundation, under "Downloads" and "White Papers": www.opcfoundation.org

Firewalls allow access to explicitly registered programs. The following programs must have access:

- ▶ All OPC Clients
- ▶ Microsoft Management Console
- ▶ Pilz PVIS OPC Server DA.exe (only for PVIS OPC Server)
- or
- ▶ PSS4000_OPCTServerDA.exe (only for PSS 4000 with OPC Server (classic) system section)
- ▶ OPCenum.exe

A port must also be enabled for data exchange:

- ▶ Name: DCOM
- ▶ Port number 135
- ▶ Protocol: TCP

These enables grant access rights to the PC for other users on the network or from the Internet. A network administrator can optimise these Firewall settings to protect the server from unauthorised access.

15.6 Commissioning the OPC Server

To commission the OPC Server, proceed as follows:

1. Configure the OPC Server in PAS4000 (see PAS4000 Online Help).
2. Perform a project download.

As the project is downloaded, the device project for the OPC Server is created and is stored on the OPC Server. The device project for the OPC Server contains the OPC configuration, which consists of the configuration data and the complete namespace for all the devices configured on the project.

Once the project has been downloaded, the OPC Server makes the complete namespace for all devices configured on the project available to the OPC Clients.

15.7 Data type assignment

Various data types are assigned through the OPC standard. The table below contains the data types supported via the OPC Server and their assignment to the data types available in the user program on the PSS 4000:

PSS 4000 namespace	OPC Classic	User program (PSS 4000)
Boolean	VT_BOOL	BOOL
SByte	VT_I1	SINT
Int16	VT_I2	INT
Int32	VT_I4	DINT
Int64	VT_I8	LINT
Byte	VT_UI1	USINT
		BYTE
UInt16	VT_UI2	UINT
		WORD
UInt32	VT_UI4	UDINT
		DWORD
UInt64	VT_UI8	ULINT
		LWORD
Float	VT_R4	REAL
Double	VT_R8	LREAL
String	VT_BSTR	---
DateTime	VT_DATE	TIME
		DATE
		TIME_OF_DAY
		DATE_AND_TIME

15.8 Access to data in the PSS 4000 namespace

The PSS 4000 OPC Server collects the general project data and device-based project data from a PSS 4000 project and presents it in the PSS 4000 namespace. The PSS 4000 namespace is a special memory area with a predefined structure. An OPC Client can access the data in the PSS 4000 namespace via this predefined structure.

Further information on the PSS 4000 namespace and its structure can be found under [PSS 4000 namespace](#) [ 741].

15.9 Network data (OPC Server (classic))

Protocol	Direction [*]	Transport protocol	Port no.	Can be deactivated	Time-critical	Description
SNTP	In	UDP (IPv4)	123	Yes Default: inactive	No	Affected devices: ▶ PMI/PC with PSS 4000 firmware Application: ▶ Time synchronisation ▶ SNTP Server
SNTP	Out	UDP (IPv4)	123	Yes Default: inactive	No	Affected devices: ▶ PMI/PC with PSS 4000 firmware Application: ▶ Time synchronisation ▶ SNTP Client
SNp	In/out	UDP (IPv4)	40000	No	Yes	Affected devices: ▶ PMI/PC with PSS 4000 firmware Application: ▶ Communication via SafetyNET p for reading and retrieving – Process data – Status information – Event messages

Protocol	Direction [*]	Transport protocol	Port no.	Can be deactivated	Time-critical	Description
SNp MSC	In	UDP (IPv4)	40000	No	No	Affected devices: ▶ PMI/PC with PSS 4000 firmware Application: ▶ SafetyNET p communication with IP-Multicast for – Scan network – Device naming – Device identification Implemented security measure: ▶ Multicast addresses limited to: 224.0.23.61
SNp MSC	Out	UDP (IPv4)	40000	No	No	Affected devices: ▶ PC with PSS 4000 firmware for PC Application: ▶ SafetyNET p communication with IP-Multicast for – Scan network – Device naming – Device identification Implemented security measures: ▶ Multicast addresses limited to: 224.0.23.61

[*]

in: The communication partner starts communication with the product.

out: The product starts communication with the communication partner.

Also refer to the information under [Firewall configuration](#) [ 666].

16 OPC Server UA (PC, PMI)

16.1 Overview

OPC stands for Openness, Productivity, Collaboration and UA for Unified Architecture. OPC UA is an industrial standard of the OPC Foundation. OPC UA has properties such as platform independence, scalability, high availability, internet capability and more.

OPC UA provides a standardised software interface which enables applications from a wide variety of manufacturers to exchange data. The OPC Foundation is responsible for maintaining and propagating the standard.

On networked plants, an OPC Server will typically collect diagnostic data, process data etc. and display it in the defined namespace. An OPC Client is typically used for process visualisation and/or to record operating data. It accesses and then processes the data provided by the OPC Server. In order to process the data the OPC Client must contain the relevant OPC-enabled software (e.g. visualisation system PASvisu from Pilz, Zenon from Copa Data).

In the automation system PSS 4000, a PC obtains the system section PSS 4000 OPC Server UA by installing the PSS 4000 firmware. By installing the PSS 4000 firmware, a PC becomes a "PSS 4000-operable device from a third-party manufacturer" (see [System overview](#) [📖 20]).

The PSS 4000 OPC Server UA reads the process data, status information and event messages from the connected PSS 4000-exclusive devices and displays them in the PSS 4000 namespace. There they can be retrieved by an OPC UA-Client.



NOTICE

The PSS 4000 OPC Server UA can only be used with OPC UA Clients (e.g. visualisation system PASvisu from Pilz). OPC Clients (classic) cannot access the PSS 4000 OPC Server UA.

The following boundary conditions apply to the PSS 4000 OPC Server UA:

- ▶ The PSS 4000 OPC Server UA supports various OPC UA properties for access from OPC UA Clients (see [OPC UA properties](#) [📖 674]).
- ▶ The PSS 4000 OPC Server UA can be installed on a PC (see [System requirements](#) [📖 679]).
- ▶ Each project created with PAS4000 may contain up to 4 OPC Servers. OPC Classic Servers and OPC UA Servers may be combined in the process.
- ▶ It is possible to access an OPC Server simultaneously with more than one Client. The following applies:
 - Only OPC Classic Clients may access an OPC Classic Server.
 - Only OPC UA Clients may access an OPC UA Server.
- ▶ All project-based settings for the PSS 4000 OPC Server UA are made in PAS4000 and are transferred to the PSS 4000 OPC Server UA as a device project when the project is downloaded.

16.1.1 Design options

In a minimum configuration, the PSS 4000 OPC Server UA will be on the same PC as the OPC UA Client containing the visualisation software.

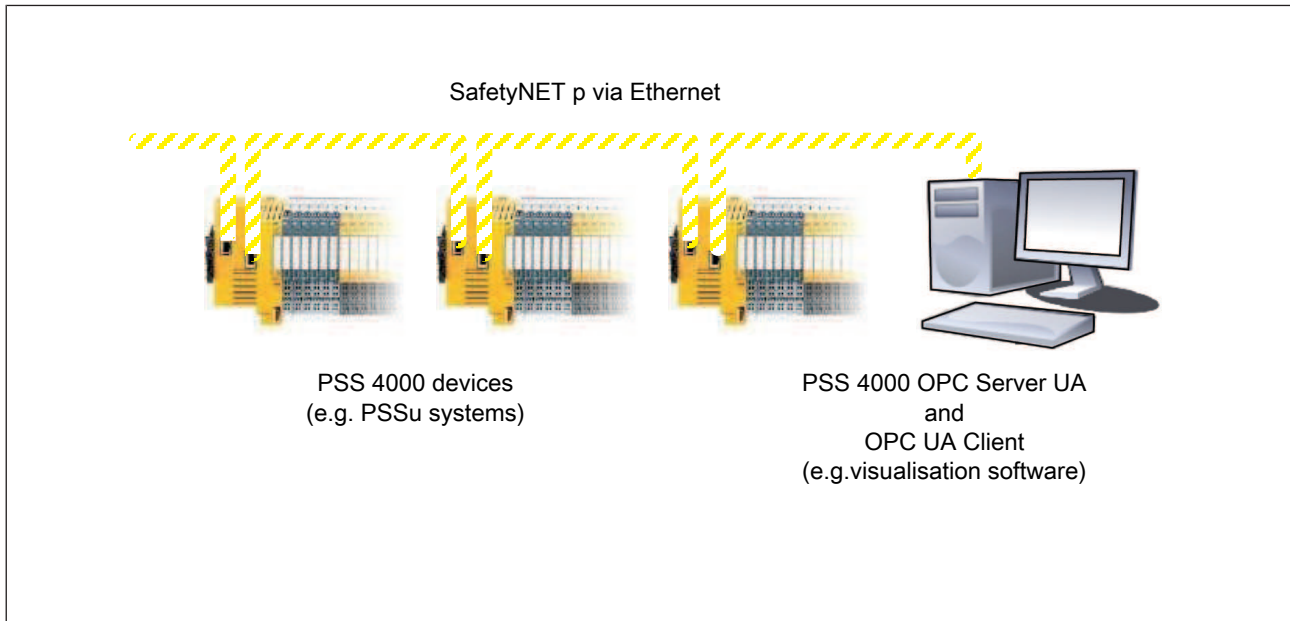


Fig.: Process visualisation with a minimum OPC UA configuration (principle)

Other types of combinations are also conceivable, such as:

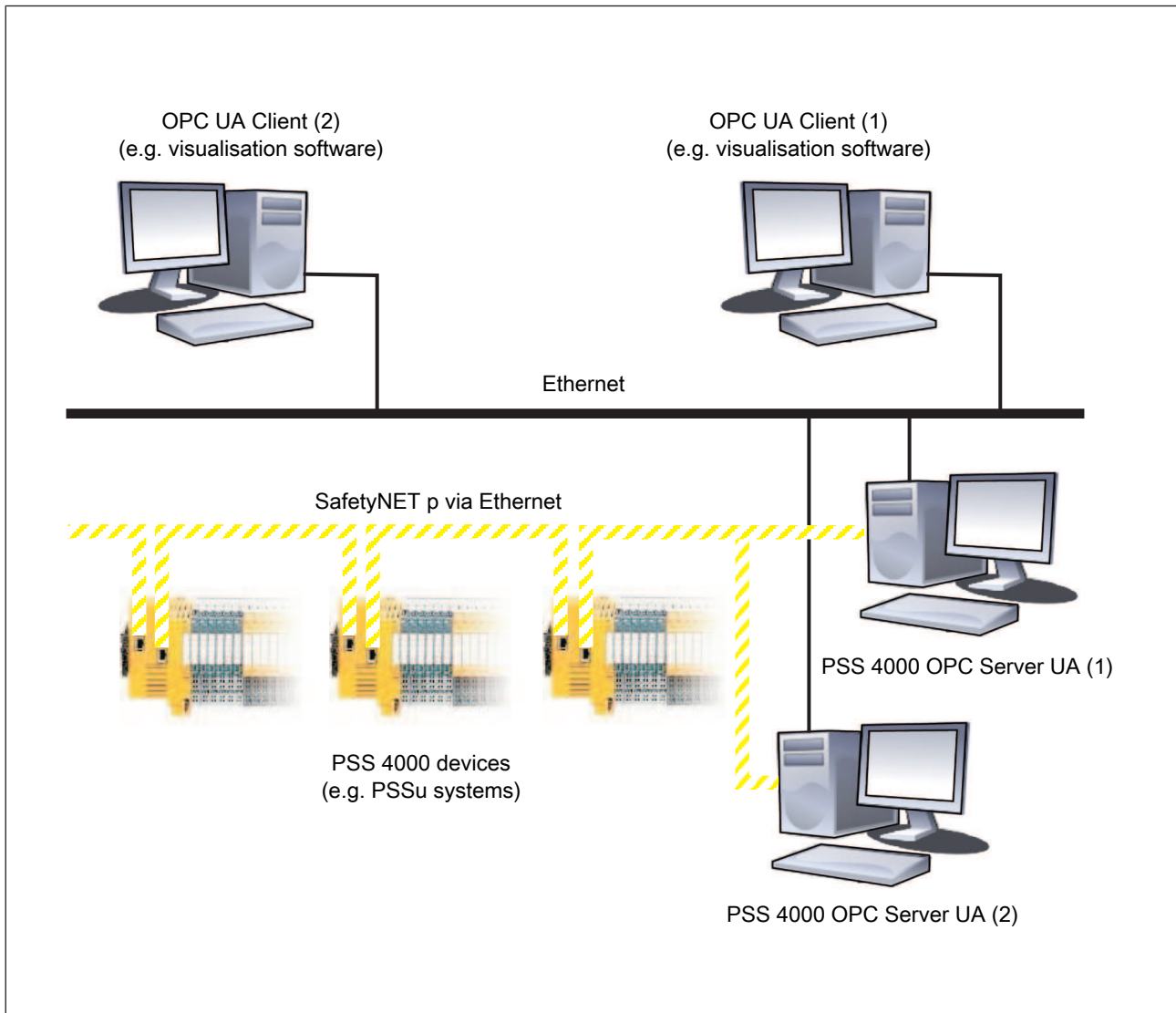


Fig.: Process visualisation with OPC UA with redundant design and networking via Ethernet (principle)

16.1.2 OPC UA properties

The PSS 4000 OPC Server UA supports the following OPC UA properties for access from OPC UA Clients:

- ▶ Supports the UA binary protocol via TCP
- ▶ End point URL of the PSS 4000 OPC Server UA (address in the network):
opc.tcp://<IP address>:<Port>/Pilz/PSS4000OPCServerUA

<IP address>	IP address of the PSS 4000 OPC Server UA, which was configured with the naming data in PAS4000
<Port>	Port number of the PSS 4000 OPC Server UA, which was configured in PAS4000. ▶ Default setting when an OPC UA Client has read/write access (RW access): 4853 ▶ Default setting when an OPC UA Client has read only access (RO access): 4854

Note: You can adjust the port numbers in PAS4000. Further information is available under [Access rights of OPC UA Clients](#)  678].

- ▶ Data Access (DA) support
- ▶ Support for generic access to process data in the PSS 4000 namespace with pre-defined syntax
- ▶ Support for symbolic access to process data in the PSS 4000 namespace
- ▶ Support for OPC UA services and service sets
(in accordance with OPC Unified Architecture – Part 4: Services)

Service set	Supported services	Meaning
Discovery	▶ (Discovery Server functions)	Discovery Server functions Note: If the PSS 4000 OPC Server UA is being installed on a PC, then the Discovery Server will also be installed.
Secure Channel	▶ OpenSecureChannel ▶ CloseSecureChannel	Connection management between OPC UA Clients and OPC UA Servers
Session	▶ CreateSession ▶ ActivateSession ▶ CloseSession ▶ Cancel	Connection management between OPC UA Clients and OPC UA Servers
View	▶ Browse ▶ BrowseNext ▶ TranslateBrowsePathsToNodeIds	Retrieval of information in the address space
Attributes	▶ Read ▶ Write	Read and write data and metadata

Service set	Supported services	Meaning
MonitoredItem	▶ CreateMonitoredItems ▶ ModifyMonitoredItems ▶ SetMonitoringMode ▶ SetTriggering ▶ DeleteMonitoredItems	See [1]
Subscription	▶ CreateSubscription ▶ ModifySubscription ▶ SetPublishingMode ▶ Publish ▶ Republish ▶ DeleteSubscription	See [1]

[1] Service sets **"Subscription"** and **"MonitoredItems"**

An OPC UA Client can subscribe to data provided by the OPC UA Server. The purpose of a subscription is to group data using monitored items.

Please note the limit values for the service sets "Subscription" and "MonitoredItem" (see [Limit values](#)  676)

16.1.3 Limit values

Designation		Limit values
Service set "Subscription" (see also OPC UA properties [674])		
PublishingInterval		min. 100 ms
		max. 15000 ms
		Note: ▶ PublishingInterval < 100 ms If, for a subscription, an OPC UA Client uses a value for "PublishingInterval" that is less than 100 ms, the PSS 4000 OPC Server UA uses the minimum value of 100 ms for "PublishingInterval" ▶ PublishingInterval > 15000 ms If, for a subscription, an OPC UA Client uses a value for "PublishingInterval" that is greater than 15000 ms, the PSS 4000 Server UA uses the maximum value of 15000 ms for "PublishingInterval".
Maximum number of subscriptions per resource		max. 50
Maximum number of monitored items per subscription; i.e. maximum number of variables that can be requested per subscription		500 per resource Note: With an array, each element is counted, irrespective of the access type (see Special features when accessing arrays in the namespace [701]).
Service Set "MonitoredItem" (see also OPC UA properties [674])		
SamplingInterval		min. 0 ms
		max. 15000 ms
		Notes: ▶ SamplingInterval = 0 The PSS 4000 OPC Server UA uses the fastest reasonable update interval. ▶ SamplingInterval > 0 and < 100 ms If, for a monitored item, an OPC UA Client uses a value for "SamplingInterval" that is greater than 0 ms and less than 100 ms, the PSS 4000 OPC Server UA uses the minimum value of 100 ms for "SamplingInterval". ▶ "SamplingInterval" > 15000 ms If, for a monitored item, an OPC UA Client uses a value for "SamplingInterval" that is greater than 15000 ms, the PSS 4000 OPC Server UA uses the maximum value of 15000 ms for "SamplingInterval". For further guidance see [1]

[1] Notes on "SamplingInterval":

- ▶ The PSS 4000 OPC Server UA uses a common update interval for each subscription of a device (e.g. PSS 4000 device, diagnostic server).

- ▶ Update interval

The lowest possible value for the update interval is 100 ms.

The update interval is calculated from the value of the lowest PublishingInterval for a subscription divided by 2 and the values of the update intervals that result from the SamplingInterval and QueueSize of the monitored items.

- The SamplingInterval of the monitored item is > 0:

Update interval = SamplingInterval / 2

- The SamplingInterval of the monitored item is 0:

Update interval = (PublishingInterval of the subscription / queue size) / 2

- The SamplingInterval of the monitored item is 1:

Update interval = PublishingInterval of the subscription / 2

- ▶ The shortest delay between a variable changing on a PSS 4000 device and the modified variable being available on the OPC UA Client occurs when the monitored item is generated with SamplingInterval = 0 and QueueSize = 1.

16.1.4 Access rights of OPC UA Clients

An OPC UA Client can have either read/write access (RW access) or read-only access (RO access) to the PSS 4000 OPC Server UA. Access rights are controlled via the port that is used:

- ▶ Default setting when an OPC UA Client has read/write access (RW Client): 4853
- ▶ Default setting when an OPC UA Client has read only access (RO Client): 4854

The port numbers can be adjusted in PAS4000.



NOTICE

Preventing write access for RO Clients

An OPC UA Client with RO access (RO Client) to the OPC UA Server must have no possibility of using a port for RW access.

Use a firewall to prevent unwanted write access by OPC UA Clients to the OPC UA Server. Configure the firewall so that access by a RO Client to a read/write end point is prevented; i.e. communication between RO Client and OPC UA Server must not occur via a port that has been configured for RW access in PAS4000.

16.1.5 Security settings for end points

For end points, various encryption levels (security policy and message security modes) are supported. You find an overview and further information on the currently available Security Policies at [Security policies of OPC UA Servers](#) [📖 697].

16.2 System requirements

Various system requirements must be met so that the PSS 4000 firmware can be installed on a device.

16.2.1 PC with Microsoft Windows operating system

The PSS 4000 firmware can be installed on a PC with a Microsoft Windows operating system:

- ▶ Microsoft Windows 10 in one of these versions
 - Windows 10 Home (64 Bit)
 - Windows 10 Pro (64 Bit)
 - Windows 10 Enterprise (64 Bit)

16.3 Licensing

Licensing of the OPC Server takes place when a project is licensed in PAS4000. Further information on licensing is available in PAS4000's online help.

16.4 Installation

The current installation files are available in the download area of the Pilz homepage (www.pilz.de -> Downloads). "Software" can be entered as the search term, for example. The software download is only available for registered users. Unregistered users can register free of charge. A valid E-mail address is required for registration.

Installation kits are available for the PSS 4000 firmware, depending on the application area:

- The PSS 4000 firmware is installed on a **PC**:

Required installation set:

PSS_4000_PC_<Version number>_<Identifier>.zip

(e.g. PSS_4000_PC_1_13_0_3000763B29.zip)

16.4.1 Install PSS 4000 firmware on a PC

The PSS 4000 firmware is installed on the PC that serves as the diagnostic unit. The diagnostic unit may also be used simultaneously as a display unit. When installing the PSS 4000 firmware, both the PSS 4000 OPC Server (classic) and the PSS 4000 OPC Server UA are installed.

The OPC Core Components are also installed automatically for the PSS 4000 OPC Server (classic). The OPC Core Components consist of various auxiliary programs and software components for OPC Servers. The PSS 4000 Firmware Controller is also installed.

For the PSS 4000 OPC Server UA, a self-signed certificate with public key and associated private key is created during installation.



INFORMATION

- If the OPC Server is installed on the same PC as PAS4000, the PAS4000 firmware version must be identical to the version of the PSS 4000 firmware.
- If the OPC Server is installed on the same PC as PAS4000, we recommend that you install PAS4000 first, followed by the PSS 4000 firmware.
- If the OPC Server is installed on the same PC as PAS4000, the PSS 4000 firmware sets the system variable PILZ_SNP_MULTICAST_TTL to 64 during installation.
- If the OPC Server is installed on the same PC as PAS4000, PAS4000 may affect the project on the OPC Server. If an online action in PAS4000 is executed in another project, the project on the OPC Server may become unusable and may break the connections to the OPC Clients.
- An OPC Client (classic) can only access a PSS 4000 OPC Server (classic).
- An OPC UA Client can only access a PSS 4000 OPC Server UA.



NOTICE

Start OPC Server

Before an OPC Client can access the OPC Server, the Server must first be started. There are various procedures for doing this:

- Automatic start via the Autostart menu
When installing the PSS 4000 firmware, the default setting in the installation window is retained. This means that the PSS 4000 Firmware Controller is entered in the Autostart menu of the Windows operating system. Each time the PC is booted, the OPC Server is started automatically.
- Manual start
During installation, the option responsible for the entry in the Autostart menu is deselected in the installation window. This means that the OPC Server must be started manually via the PSS 4000 Firmware Controller after installation and each time the PC is booted.

Installing via the install program

⇒ Double-click on the install file "PSS_4000_PC_<Version number>_Installer.exe" and follow the on-screen instructions.

Define target directory for the certificate store on the PC

When installing the PSS 4000 firmware, the certificate store can either be created in the public space or in the private space of the current user.

For security reasons, we recommend that the certificate store is installed in the private space (see also Directory structure of the certificate store).



NOTICE

Potential security risk due to certificate store in the public space

Creating the certificate store in the PC's public space is a potential security risk. In this case, all of the PC's users have access to the certificate store and therefore to the private key of the OPC Server UA. Also, all users can store client certificates in the "trusted" directory, in order to allow communication connections to the OPC Server UA.

When carrying out your security risk analysis, consider that creating the certificate store in the public space can represent a security vulnerability for cyber threats.

Uninstalling via the uninstall program

⇒ In the Windows program list right-click on **Pilz -> PSS 4000 firmware for PC <Version number>** and select **Uninstall**.

Silent install

It is also possible to have an unattended ("silent") installation of the PSS 4000 firmware. In this case the installation is carried out without any user interaction and the installation process runs in the background.

⇒ Start the installation by entering one of the commands described below in the Windows command line, the Windows search field or in a batch file.

Command for installation with default values:

"<Absolute directory of the install file>" /S /PilzEULA='Y'

Example:

```
"c:\myFiles\PSS_4000_PC_<Version number>_Installer.exe" /S /
PilzEULA='Y'
```

Command for installation with specifications:

"<Absolute directory of the install file>" /S /auto='<Value>' /lang='<Tool language>' /
PilzEULA='<Consent to EULA>' /localOPC='<Storage location of the certificate store>' /
D='<Absolute installation directory>'

Example:

```
"c:\myFiles\PSS_4000_PC_<Version number>_Installer.exe" /S /
auto='1' /lang='de' /PilzEULA='Y' /localOPC='1' /D=C:\myPrograms
```

Supported arguments:

Argument	Meaning
/S	Run file silently.
/auto	With this argument you can control whether the PSS 4000 firmware is started automatically when booting the PC. Argument is optional. Value range: 1: PSS 4000 firmware is started automatically when booting the PC 0: PSS 4000 firmware is not started automatically Default value: 1
/lang	Language for messages and language for the uninstall program, if the uninstall is not run silently. Argument is optional. Value range: de: German en: English Default value: If the language for the operating system is German, then German is the default language, otherwise English.
/PilzEULA	Consent to end user licence agreement for the PSS 4000 firmware; the text file containing the licence text is next to the install file in the folder "Licence\PSS4000_PC_x_x_x_EULA_End_User_Li- cense_Agreement". Please read the licence text and give your consent before starting the installation. Consent is a requirement for running the installation. Valid value: Y All other values or a missing argument will mean that the installation is not run.

Argument	Meaning
/localOPC	Target directory for the certificate store of the PSS 4000 OPC Server UA, the argument is optional. Value range: 1: Storage location of the certificate store in the private space C:\<%LOCALAPPDATA%>\Pilz\PSS4000OPCServerUA\pki 0: Storage location of the certificate store in the public space: C:\<%ALLUSERSPROFILE%>\Pilz\PSS4000OPCServerUA\pki Default value: If the argument is not used, then the certificate store is created in the private space. Notes: ▶ During installation, if the certificate store is created in the private space, but the OPC Server UA is then started by another user, then on start up the system will search for the certificate store in the private space of the user that started the OPC Server UA. In this case, the certificate store must be created manually under <%LOCALAPPDATA%>\Pilz\PSS4000OPCServerUA of the user who starts the OPC Server UA. ▶ Potential security risk due to certificate store in the public space Creating the certificate store in the PC's public space is a potential security risk. In this case, all of the PC's users have access to the certificate store and therefore to the private key of the OPC Server UA. Also, all users can store client certificates in the "trusted" directory, in order to allow communication connections to the OPC Server UA. When carrying out your security risk analysis, consider that creating the certificate store in the public space can represent a security vulnerability for cyber threats.
/D	Installation directory; must be entered as the last argument, argument is optional Default value: %programfiles(x86)%\PILZ

Note: "Start /wait" can be added to the install command so that no further commands are started until the install command has been executed in full. Please note that there must be two double quotation marks between "start" and "/wait".

Example:

```
Start "" /wait "c:\myFiles\PSS_4000_PC_<Version number>_Installer.exe" /S /auto='1' /lang='de' /PilzEULA='Y' /localOPC='1' /D=C:\myPrograms
```

Silent uninstall

The uninstall can also be run silently.

⇒ Start the uninstall by entering the command described below in the Windows command line, the Windows search field or in a batch file.

Uninstall command:

"<Absolute directory of the uninstall file>" /S

Example:

```
"C:\Program Files (x86)\Pilz\PAS4000 1.20.0\uninstallerData\uninstall.exe" /S
```

Note: "Start /wait" can be added to the install command so that no further commands are started until the uninstall command has been executed in full. Please note that there must be two double quotation marks between "start" and "/wait".

Example:

```
Start "" /wait "C:\Program Files (x86)\Pilz\PAS4000 1.20.0\uninstallerData\uninstall.exe" /S
```

16.5 Information security certificates

Networked automation systems place demands on information security. Information security encompasses the authenticity, confidentiality, integrity and availability of data as well as protection of access to data.

On the PSS 4000 OPC Server UA from Pilz, information security is established with the aid of certificates compliant with the X.509 standard. The certificates are used to safeguard communication between an OPC UA Client and the OPC UA Server; i.e. with a certificate, data traffic from a unidirectional communication connection is encrypted and communication between one subscriber and another is authenticated.

A separate certificate is required for each subscriber in the Client-Server communication.

An OPC UA Client cannot communicate with the PSS 4000 OPC Server UA until the PSS 4000 OPC Server UA has a valid certificate from the OPC UA Client and the OPC UA Client has a valid certificate from the PSS 4000 OPC Server UA.

16.5.1 Create certificate during installation

A self-signed certificate with a public key and associated private key is created as standard when the PSS 4000 OPC Server UA is installed. This certificate is intended for commissioning and should not be used in normal operation.

We recommend you create your own certificates for normal operation. Further information is available in the chapter entitled [Generating and using custom certificates](#) [ 694].



INFORMATION

In rare cases the situation may arise in which an OPC UA Client from a third-party manufacturer does not accept the certificate created during installation. In this case, you will need to generate and use your own certificates even for commissioning.

16.5.2 Directory structure of the certificate store

A self-signed certificate with a public key and associated private key is created as standard when the PSS 4000 OPC Server UA is installed (see [Create certificate during installation \[687\]](#)). The public key and non-public (private) key are stored in a certificate store on the installation device. A pre-defined directory structure is created as the certificate store during installation. The location of the certificate store depends on the device on which the PSS 4000 OPC Server UA is installed. However, the directory structure is always the same:

<basepath>\pki

\own	Location for the public part of the self-signed certificate The public part of the self-signed certificate is always stored in this folder during installation.
\private	Location for the non-public (private) part of the self-signed certificate The non-public part of the self-signed certificate is always stored in this folder during installation.
\rejected	Location for unknown or rejected certificates
\trusted	Location for trusted certificates



INFORMATION

The certificates are only created with a new installation. In the case of an update, all existing certificates in the certificate store are retained.



NOTICE

Use the directory structure even if you generate your own certificates.

Storage location when installed on a PC

When the PSS 4000 firmware was installed, either the public space or the private space of the current user was defined as the storage location for the certificate store. The certificate store is created based on this configuration.

- Storage location of the certificate store in the private space (default setting):

C:\<%LOCALAPPDATA%>\Pilz\PSS4000OPCServerUA\pki

Note: During installation, if the certificate store is created in the private space, but the OPC Server UA is then started by another user, then on start-up the system will search for the certificate store in the private space of the user that started the OPC Server UA. In this case, the certificate store must be created manually under <%LOCALAPPDATA%>\Pilz\PSS4000OPCServerUA of the user who starts the OPC Server UA.

- Storage location of the certificate store in the public space:

C:\ProgramData\Pilz\PSS4000OPCServerUA\pki

16.5.3 Procedure for authentication

To ensure that client-server communication can take place, the communication partners must exchange their certificates. For the certificates to be exchanged, the certificate files must be stored in the relevant folder in the certificate store of the OPC UA Client and PSS 4000 OPC Server UA (see [Directory structure of the certificate store](#)  688).

There are different procedures for mutual authentication of the relevant PSS 4000 OPC Server UA and OPC UA Client. The procedures can be differentiated according to the following criteria:

- ▶ Using PASvisu Server as OPC UA Client
 - PSS 4000 OPC Server UA and PASvisu Server are on the same device (see [Authentication in the case of local installation](#)  689)
 - PSS 4000 OPC Server UA and PASvisu Server are installed on different devices (see [Authentication in the case of remote installation](#)  689)
- ▶ Using OPC UA Client of a third-party manufacturer (see [Authentication in the case of OPC UA Clients of a third-party manufacturer](#)  693)

16.5.3.1 Authentication in the case of local installation

The PASvisu Server is used as an OPC UA Client. If the PSS 4000 OPC Server UA and the PASvisu Server are installed on the same device, then the certificates are exchanged automatically during installation:

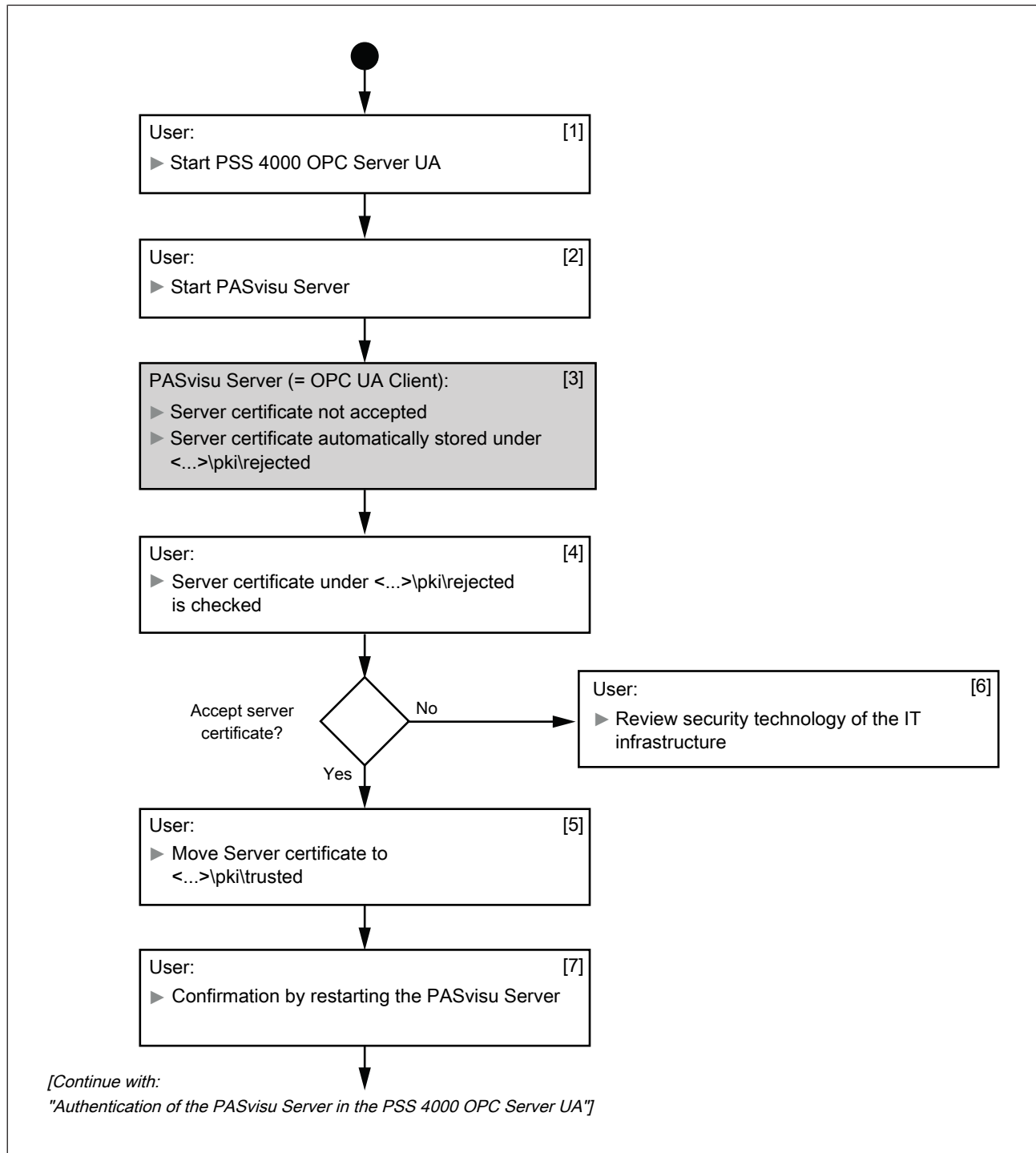
- ▶ The certificate of the PSS 4000 OPC Server UA (cert_Pilz_PSS 4000_OPC_Server_UA.der) is copied to the "trusted" folder of the PASvisu.
- ▶ The certificate of the PASvisu Server (cert_Pilz_PASvisu_OPC_UA.der) is copied to the "trusted" folder of the PSS 4000 OPC Server UA

The user does not need to do anything else.

16.5.3.2 Authentication in the case of remote installation

The PASvisu Server is used as an OPC UA Client. If the PSS 4000 OPC Server UA and the PASvisu Server are installed on different devices, then the certificates must be exchanged by the user.

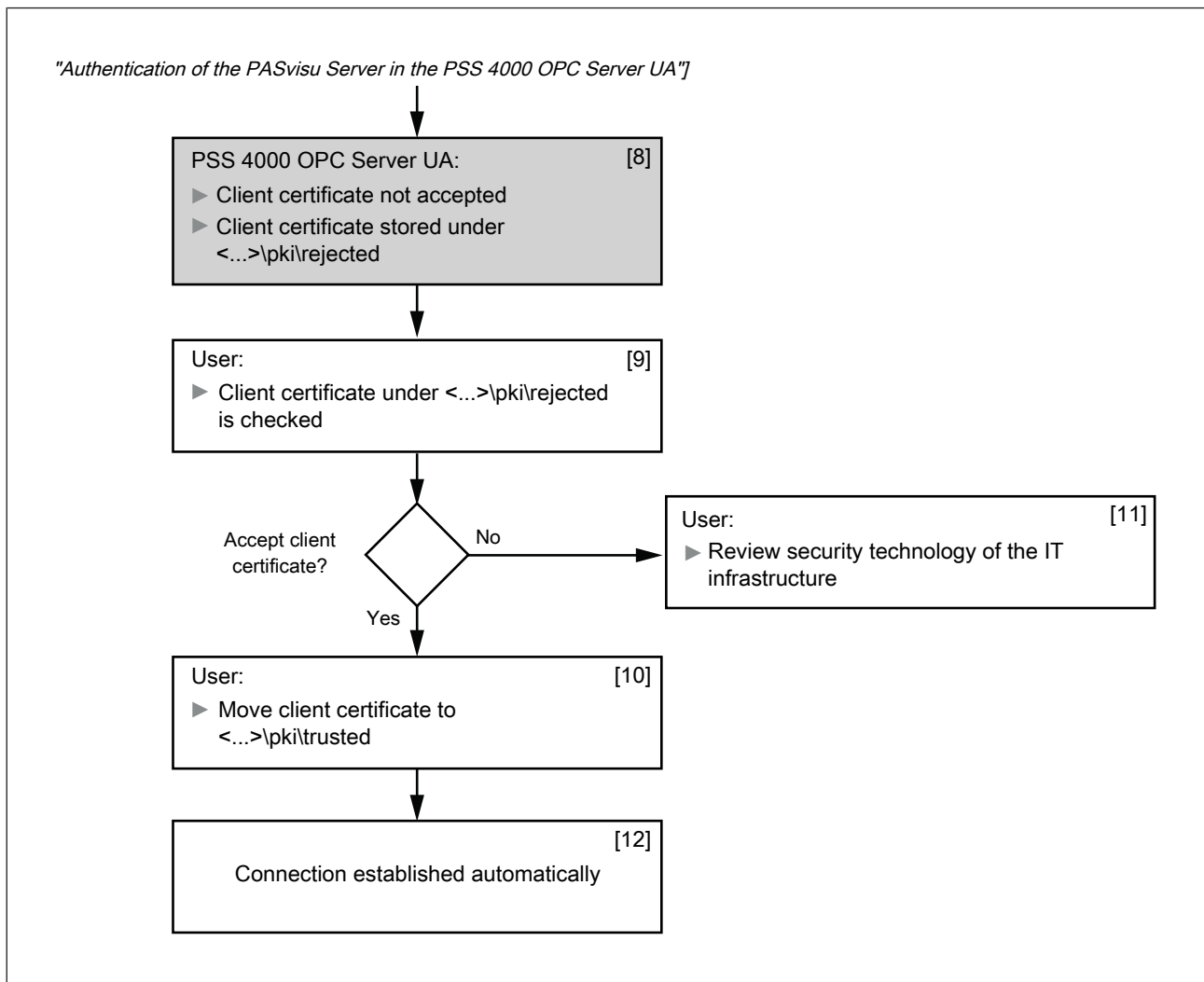
Procedure



Legend

- [1] First time the PSS 4000 OPC Server UA is started after being installed by the user
- [2] First time the PASvisu Server is started after being installed by the user

- [3] The PASvisu Server attempts to establish a connection to the PSS 4000 OPC Server UA. As the PASvisu Server does not have a valid certificate from the PSS 4000 OPC Server UA after installation, the PASvisu Server reacts as follows:
- ▶ Server certificate is not accepted by the PASvisu Server
 - ▶ Server certificate is saved in the certificate store of the PASvisu Server in the directory "<...>\pkilrejected".
The name of the certificate file is composed of the certificate's common name, domain component and fingerprint.
Structure of the file name: <CN>@<DC>_[Fingerprint].der
 - CN: Common Name
 - DC: Domain Component
 - Fingerprint: 40-digit HEX value
- Example:
 Pilz_PSS4000_OPC_Server_UA_self_signed_@192_168_1_101_[0ff981ea8ba58ad04bad16f89a3cc9c0c60e7fac].der
 - CN: Pilz PSS4000 OPC Server UA (self signed)
 - DC: 192.168.1.101
 - Fingerprint: 0ff981ea8ba58ad04bad16f89a3cc9c0c60e7fac
- Note:
 Any characters in the CN and DC that are not in the range A ... Z, a ... z and 0 ... 9 are replaced with an underscore.
- ▶ Connection to the PSS 4000 OPC Server UA is not established
- [4] The user must check whether the server certificate is trustworthy (see [Verifying a certificate](#)  696)
- [5] If the user considers the server certificate to be trustworthy, he must move it from the "rejected" directory to the "trusted" directory in the certificate store of the PASvisu Server.
 The PSS 4000 OPC Server UA is now authenticated on the PASvisu Server.
- [6] If there are doubts about the trustworthiness of the server certificate, the security technology of the IT infrastructure should be reviewed.
- [7] For the PSS 4000 OPC Server UA to receive a valid certificate from the PASvisu Server, a connection between the two must first be established.



Legend

[8] As the PSS 4000 OPC Server UA does not have a valid certificate from the PASvisu Server after installation, the PSS 4000 OPC Server UA reacts as follows:

- ▶ The client certificate is not accepted by the PSS 4000 OPC Server UA
- ▶ The client certificate is saved in the certificate store of the PSS 4000 OPC Server UA in the directory "<...>\pkilrejected".

The name of the certificate file is composed of the common name, domain component and fingerprint.

Structure of the file name: <CN>_<DC>_[Fingerprint].der

- CN: Common Name
- DC: Domain Component
- Fingerprint: 40-digit HEX value

Example:

Pilz_PASvisu_OPC_UA_192_168_1_170_[c0340c57b-cf7a6e7d1097735d3358351b942755c].der

- CN: Pilz PASvisu OPC UA
- DC: 192.168.1.170
- Fingerprint: c0340c57bcf7a6e7d1097735d3358351b942755c

Note:

Any characters in the CN and DC that are not in the range A ... Z, a ... z and 0 ... 9 are replaced with an underscore.

► Connection to the PASvisu Server is not established

[9] The user must check whether the client certificate is trustworthy (see [Verifying a certificate](#) [696]).

[10] If the user considers the client certificate to be trustworthy, he must move it from the "rejected" directory to the "trusted" directory of the PSS 4000 OPC Server UA certificate store.

The PASvisu Server is now authenticated on the PSS 4000 OPC Server UA.

[11] If there are doubts about the trustworthiness of the client certificate, the security technology of the IT infrastructure should be reviewed.

[12] Following mutual authentication, the PASvisu Server automatically establishes the connection to the PSS 4000 OPC Server UA.



NOTICE

During authentication, a certificate (client or server certificate) is only saved in the directory "<...>\pki\rejected" directory under the following conditions:

- There is no certificate with the same fingerprint in the directory "<...>\pki\rejected"
- There is no valid certificate with the same fingerprint in the directory "<...>\pki\trusted"

16.5.3.3

Authentication in the case of OPC UA Clients of a third-party manufacturer

If the OPC UA Client of a third-party manufacturer is used with the PSS 4000 OPC Server UA from Pilz, then the certificates must be exchanged by the user. For the OPC UA Client, follow the instructions in the third-party manufacturer's operating manual.

Procedure

► OPC UA Client

Follow the authentication instructions in the third-party manufacturer's operating manual.

Note:

Some OPC UA Clients need information about the directory in which the certificate of the OPC UA Server is stored. After installation, the certificate of the PSS 4000 OPC Server UA from Pilz is located in the directory <...>\pki\own (see [Directory structure of the certificate store](#) [688]).

► PSS 4000 OPC Server UA

The procedure for the PSS 4000 OPC Server UA from Pilz is in principle the same as the procedure when the PASvisu Server is used as the OPC UA Client (see [Authentication in the case of remote installation](#) [689]).

16.5.4 Generating and using custom certificates

It is possible to use a self-generated certificate instead of the certificate that is created when the PSS 4000 OPC Server UA and/or the PASvisu Server is installed. This requires a suitable tool from a third-party manufacturer. The tool must be used to generate a certificate that has a public part and a non-public (private) part.



INFORMATION

The certificate that is created when the PSS 4000 OPC Server UA and/or the PASvisu Server is installed is intended for commissioning. We recommend you create custom certificates for normal operation.

Requirements of certificates

The PSS 4000 OPC Server UA does not validate its custom server certificate. For this reason you should make sure that both the server certificate and client certificates contain the following options:

- ▶ CA: FALSE
- ▶ If the parameter NID_key_usage is used, the value cRLSign must not be set.
- ▶ Encryption algorithm
Information on the supported encryption algorithms is available under [Security policies of OPC UA Servers](#) [697].

The PASvisu Server supports the same security policies as the PSS 4000 OPC Server UA.



INFORMATION

It is essential that you note the following documentation when generating and using custom certificates:

- OPC UA specification of the OPC Foundation
- BSI specification
- Documentation of the relevant OPC UA Client
- Documentation of the tool used to create the certificate

Incorporate certificates

For a certificate to be effective, the public part and the private part of the certificate must have the correct file name and the parts must be located in the correct directory of the certificate store (see Directory structure of the certificate store).

- ▶ File name of the public part:
 - PSS 4000 OPC Server UA: "cert_Pilz_PSS4000_OPC_Server_UA.der"
 - PASvisu Server: "cert_Pilz_PASvisu_OPC_UA.der"

- ▶ File name of the private part:
 - PSS 4000 OPC Server UA: "private_key_Pilz_PSS4000_OPC_Server_UA.pem"
 - PASvisu Server: "private_key_Pilz_PASvisu_OPC_UA.pem"

Location for the public and private part of a certificate

The certificate store is to be on the PC, in the private space of the current user:

- ▶ <%LOCALAPPDATA%>\Pilz\PSS4000OPCServerUA\pki
 - \own\cert_Pilz_PSS4000_OPC_Server_UA.der
 - \private\private_key_Pilz_PSS4000_OPC_Server_UA.pem
- ▶ <%LOCALAPPDATA%>\Pilz\PASvisu Builder\pki
 - \own\cert_Pilz_PASvisu_OPC_UA.der
 - \private\private_key_Pilz_PASvisu_OPC_UA.pem

The certificate store is to be on the PC, in the public space:

- ▶ C:\ProgramData\Pilz\PSS4000OPCServerUA\pki
 - \own\cert_Pilz_PSS4000_OPC_Server_UA.der
 - \private\private_key_Pilz_PSS4000_OPC_Server_UA.pem
- ▶ C:\ProgramData\Pilz\PASvisu Builder\pki
 - \own\cert_Pilz_PASvisu_OPC_UA.der
 - \private\private_key_Pilz_PASvisu_OPC_UA.pem



NOTICE

Potential security risk due to certificate store in the public space

Creating the certificate store in the PC's public space is a potential security risk. In this case, all of the PC's users have access to the certificate store and therefore to the private key of the OPC Server UA. Also, all users can store client certificates in the "trusted" directory, in order to allow communication connections to the OPC Server UA.

When carrying out your security risk analysis, consider that creating the certificate store in the public space can represent a security vulnerability for cyber threats.

16.5.5 Invalidating a certificate

A certificate is invalidated if the user removes the certificate from the directory <...>\pki\trusted (see [Directory structure of the certificate store](#) [📖 688]).

16.5.6 Verifying a certificate

Before a certificate is copied from the directory <...>\pki\rejected to the directory <...>\pki\trusted during authentication, the user should verify the trustworthiness of the certificate (see [Authentication in the case of remote installation](#) [📖 689]).

► PASvisu Server

The content of the certificate in the directory <...>\pki\rejected of the PASvisu Server should be compared with the content of the certificate in the directory <...>\pki\own of the PSS 4000 OPC Server UA. It is sufficient just to compare the fingerprints.

► PSS 4000 OPC Server UA

The content of the certificate in the directory <...>\pki\rejected of the PSS 4000 OPC Server UA should be compared with the content of the certificate in the directory <...>\pki\own of the PASvisu Server. It is sufficient just to compare the fingerprints.

Procedure in the case of a PC

► Double-click on a certificate to view its content.

The contents of the two certificates to be compared must be identical.



INFORMATION

When establishing the connection, the PSS 4000 OPC Server UA checks whether the local system time is within the validity range (duration) of the client certificate. If this is not the case, a connection cannot be established.

16.5.7 Security policies of OPC UA Servers

Various security policies can be activated for a PSS 4000 OPC Server UA, depending on the firmware version. If several security policies are activated, all activated security policies can be used for communication with the OPC Clients.



NOTICE

Inadequate security due to deprecated security policies or security policy "None"

It is possible to use the security policy "None". In this case, communication between OPC Server and OPC Client is completely unprotected. Some of the security policies that can be activated are deprecated and no longer represent the state of the art for the protection of communication. It is better to use the current security policies, in order to provide optimum protection for communication between OPC Server and OPC Client.

Security policy	Available in firmware version	
	< 1.25.0	≥ 1.25.0
Not recommended None	---	◆
Deprecated: Basic128Rsa15 Sign	◆	◆
Deprecated: Basic128Rsa15 SignAndEncrypt	◆	◆
Deprecated: Basic256 Sign	◆	◆
Deprecated: Basic256 SignAndEncrypt	◆	◆
Aes128-Sha256-RsaOaep Sign	---	◆
Aes128-Sha256-RsaOaep SignAndEncrypt	---	◆
Basic256-Sha256 Sign	---	◆
Basic256-Sha256 SignAndEncrypt	---	◆
Aes256-Sha256-RsaPss Sign	---	◆
Aes256-Sha256-RsaPss SignAndEncrypt	---	◆

Notes

- ▶ If the PSS 4000 OPC Server UA has a firmware version $\geq 1.25.0$ and a project is transmitted to the OPC Server, in which a firmware version $< 1.25.0$ is configured for the OPC Server, then all security policies are activated automatically, with the exception of "None". In other words, the current security policies can be used for older projects, thereby increasing the security of the communication.
- ▶ If the firmware of a PSS 4000 OPC Server UA is updated from a version $< 1.25.0$ to a version $\geq 1.25.0$, then a new certification exchange is required.

- ▶ If a firmware version < 1.25.0 is configured for an OPC Server, then all four deprecated security policies must always be activated:
 - Basic128Rsa15 Sign
 - Basic128Rsa15 SignAndEncrypt
 - Basic256 Sign
 - Basic256 SignAndEncrypt

16.6 Firewall configuration

A firewall or any other security mechanisms on the PC and network may prevent the OPC Server from communicating correctly with the connected devices. In this case it is necessary to adjust the security settings and firewall settings.

Firewalls allow access to explicitly registered programs. The following programs must have access:

- ▶ All OPC UA Clients
- ▶ PSS4000_OPCTServerUA.exe

Ports must also be enabled for data exchange:

- ▶ Port number of the OPC Server UA - Discovery servers: 4840
- ▶ Port number of the PSS 4000 OPC Server UA when an OPC UA Client (RW Client) has read/write access (RW access):
Default setting: 4853
- ▶ Port number of the PSS 4000 OPC Server UA when an OPC UA Client (RO Client) has read only access (RO access):
Default setting: Port 4854



NOTICE

Preventing write access for RO Clients

An OPC UA Client with RO access (RO Client) to the OPC UA Server must have no possibility of using a port for RW access.

Use a firewall to prevent unwanted write access by OPC UA Clients to the OPC UA Server. Configure the firewall so that access by a RO Client to a read/write end point is prevented; i.e. communication between RO Client and OPC UA Server must not occur via a port that has been configured for RW access in PAS4000.

- ▶ Log: TCP

This enables grant access rights to the PC for other users on the network or from the Internet. A network administrator can optimise these Firewall settings to protect the server from unauthorised access.

Please also read the information in the section entitled [Network data \(OPC Server UA\)](#)  721].

16.7 Data type assignment

Various data types are defined through the OPC UA standard. The table below contains the data types supported via the PSS 4000 OPC Server UA and their assignment to the data types available in the user program on the PSS 4000:

PSS 4000 namespace	OPC UA	User program (PSS 4000)
Boolean	Boolean	BOOL
SByte	SByte	SINT
Byte	Byte	BYTE
		USINT
Double	Double	LREAL
		TIME
Float	Float	REAL
Int16	Int16	INT
Int32	Int32	DINT
		FIXED
Int64	Int64	LINT
		LFIXED
String (language-dependent text, e.g. diagnostic messages)	LocalizedText	---
String (variables from the user pro- gram)	String	USTRING
		UCHAR
UInt16	UInt16	WORD
		UINT
UInt32	UInt32	DWORD
		UDINT
UInt64	UInt64	LWORD
		ULINT
DateTime	DateTime	DATE
		TIME_OF_DAY
		DATE_AND_TIME

16.7.1 Access to data in the PSS 4000 namespace

The PSS 4000 OPC Server collects the general project data and device-based project data from a PSS 4000 project and presents it in the PSS 4000 namespace. The PSS 4000 namespace is a special memory area with a predefined structure. An OPC Client can access the data in the PSS 4000 namespace via this predefined structure.

Further information on the PSS 4000 namespace and its structure can be found under [PSS 4000 namespace](#) [ 741].

16.7.2 Special features when accessing arrays in the namespace

An array in the namespace is usually represented as a single node; i.e. the individual elements of an array are not available to access as separate nodes. However, newer OPC UA Clients are able to use the IndexRange function to access the individual elements of an array. This type of access is also called "index-based access".

To enable OPC UA Clients that do not support index-based access to access individual elements of an array, an array can be represented in the namespace with all its elements; i.e. in the namespace of the PSS 4000 OPC Server UA, all the elements of an array are available to access as separate nodes. Please note that this type of access leads to a higher memory requirement on the OPC UA Server, as each element of an array must be made available in the namespace as an individual node.

The way in which arrays are represented in the namespace can be configured for a PSS 4000 OPC Server UA via PAS4000.

16.8 Network data (OPC Server UA)

Protocol	Direction [*]	Transport protocol	Port no.	Can be deactivated	Time-critical	Description
SNTP	In	UDP (IPv4)	123	Yes Default: inactive	No	Affected devices: ▶ PMI/PC with PSS 4000 firmware Application: ▶ Time synchronisation ▶ SNTP Server
SNTP	Out	UDP (IPv4)	123	Yes Default: inactive	No	Affected devices: ▶ PMI/PC with PSS 4000 firmware Application: ▶ Time synchronisation ▶ SNTP Client
SNp	In/out	UDP (IPv4)	40000	No	Yes	Affected devices: ▶ PMI/PC with PSS 4000 firmware Application: ▶ Communication via SafetyNET p for reading and retrieving – Process data – Status information – Event messages
SNp MSC	In	UDP (IPv4)	40000	No	No	Affected devices: ▶ PMI/PC with PSS 4000 firmware Application: ▶ SafetyNET p communication with IP-Multicast for – Scan network – Device naming – Device identification Implemented security measure: ▶ Multicast addresses limited to: 224.0.23.61

Protocol	Direction [*]	Transport protocol	Port no.	Can be deactivated	Time-critical	Description
SNp MSC	Out	UDP (IPv4)	40000	No	No	Affected devices: ▶ PC with PSS 4000 firmware Application: ▶ SafetyNET p communication with IP-Multicast for – Scan network – Device naming – Device identification Implemented security measures: ▶ Multicast addresses limited to: 224.0.23.61
OPC UA DA	In	TCP (IPv4)	1 ... 65535 Default: 4853, 4854	Yes Default: inactive	No	Affected devices: ▶ PMI/PC with PSS 4000 firmware Default setting for port numbers: ▶ OPC UA Clients with RW access: 4853 ▶ OPC UA Clients with RO access: 4854 Implemented security measures: ▶ Mutual authentication of PSS 4000 OPC Server UA and relevant OPC UA Client via certificates ▶ Certificate in accordance with standard X.509 with signature and encryption method
OPC Server UA - Discovery Server	In	TCP (IPv4)	4840	Yes Default: active	No	Affected devices: ▶ PC with PSS 4000 firmware for PC

[*]

in: The communication partner starts communication with the product.

out: The product starts communication with the communication partner.

Also refer to the information under Firewall configuration.

17 OPC Server UA (device-internal)

17.1 Overview

OPC stands for Openness, Productivity, Collaboration and UA for Unified Architecture. OPC UA is an industrial standard of the OPC Foundation. OPC UA has properties such as platform independence, scalability, high availability, internet capability and more.

OPC UA provides a standardised software interface which enables applications from a wide variety of manufacturers to exchange data. The OPC Foundation is responsible for maintaining and propagating the standard.

In the automation system PSS 4000, a PSS 4000-exclusive device with system section OPC Server UA has a device-internal PSS 4000 OPC Server UA (see [System sections \[📖 22\]](#)).

The device-internal OPC UA Server reads the process data, status information and event messages from the PSS 4000-exclusive device and displays them in its namespace. The namespace of a PSS 4000-exclusive device is therefore limited and only contains the process data, status information and event messages from the PSS 4000-exclusive device. An OPC UA Client can retrieve and display the device-related information from the device-related namespace.

The following boundary conditions apply to the device-internal OPC UA Server:

- ▶ A PSS 4000-exclusive device with system section OPC Server UA has exactly one instance of the PSS 4000 OPC Server UA.
- ▶ It is possible to access the device-internal OPC UA Server of a PSS 4000-exclusive device with several OPC UA Clients simultaneously.
- ▶ The device-internal OPC UA Server of a PSS 4000-exclusive device supports various OPC UA properties for access from OPC UA Clients (see [OPC UA properties \[📖 708\]](#)).
- ▶ The device-internal OPC UA Server of a PSS 4000-exclusive device is deactivated in its delivery state and after an original reset (see Activation/deactivation of the device-internal OPC UA Server).
- ▶ When the device project is transferred using the system software PAS4000, an application-dependent namespace is also transferred to a PSS 4000-exclusive device. The application-dependent namespace contains the nodes and elements that are supported by the PSS 4000-exclusive device.
- ▶ The port for OPC UA Clients to access the namespace of a PSS 4000-exclusive device cannot be changed. The port can be activated/deactivated using the system software PAS4000.
- ▶ Certificate management for information security is carried out via the Pilz NET Configurator. The Pilz NET Configurator is installed together with the system software PAS4000 and can be called up in the Online Network Editor ("Manage certificates" button).
- ▶ The access type to ARRAYS is "Complete ARRAY". The "Index-based" access type is not supported.



NOTICE

Potential security risk due to improper use of OPC UA.

To guarantee security when using OPC UA, we strongly recommend that you familiarise yourself with this subject. Different levels of knowledge are required, depending on the application. As a minimum you should have read and understood the document "OPC 10000-2: UA Part 2: Security". The document can be found here:

<https://reference.opcfoundation.org/>

17.1.1 Design options

In the minimum configuration, an OPC UA Client accesses the internal OPC UA Server of a PSS 4000-exclusive device. An OPC UA Client can access the internal OPC UA Server of multiple PSS 4000-exclusive devices.

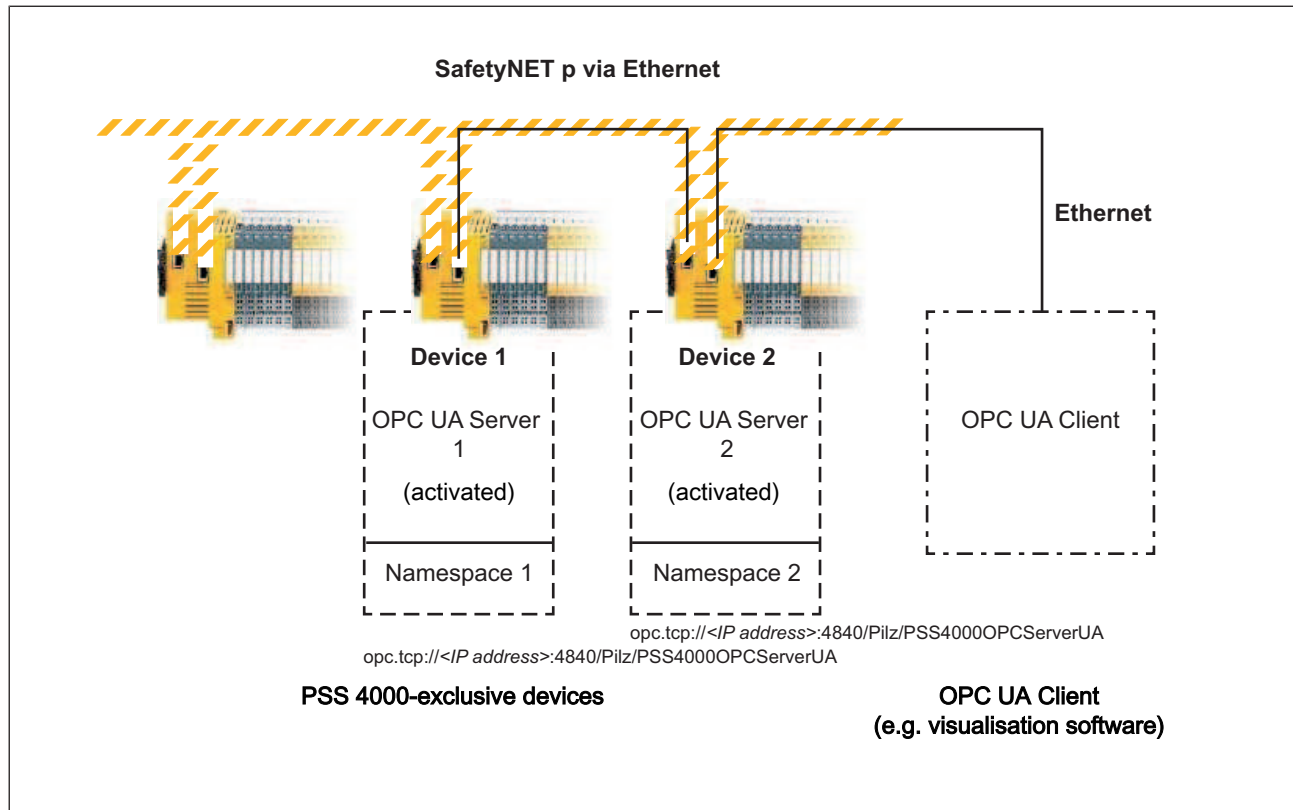


Fig.: Process visualisation with an OPC UA Client and the OPC UA Server of multiple PSS 4000-exclusive devices (principle)

Other types of combinations are conceivable, for example:

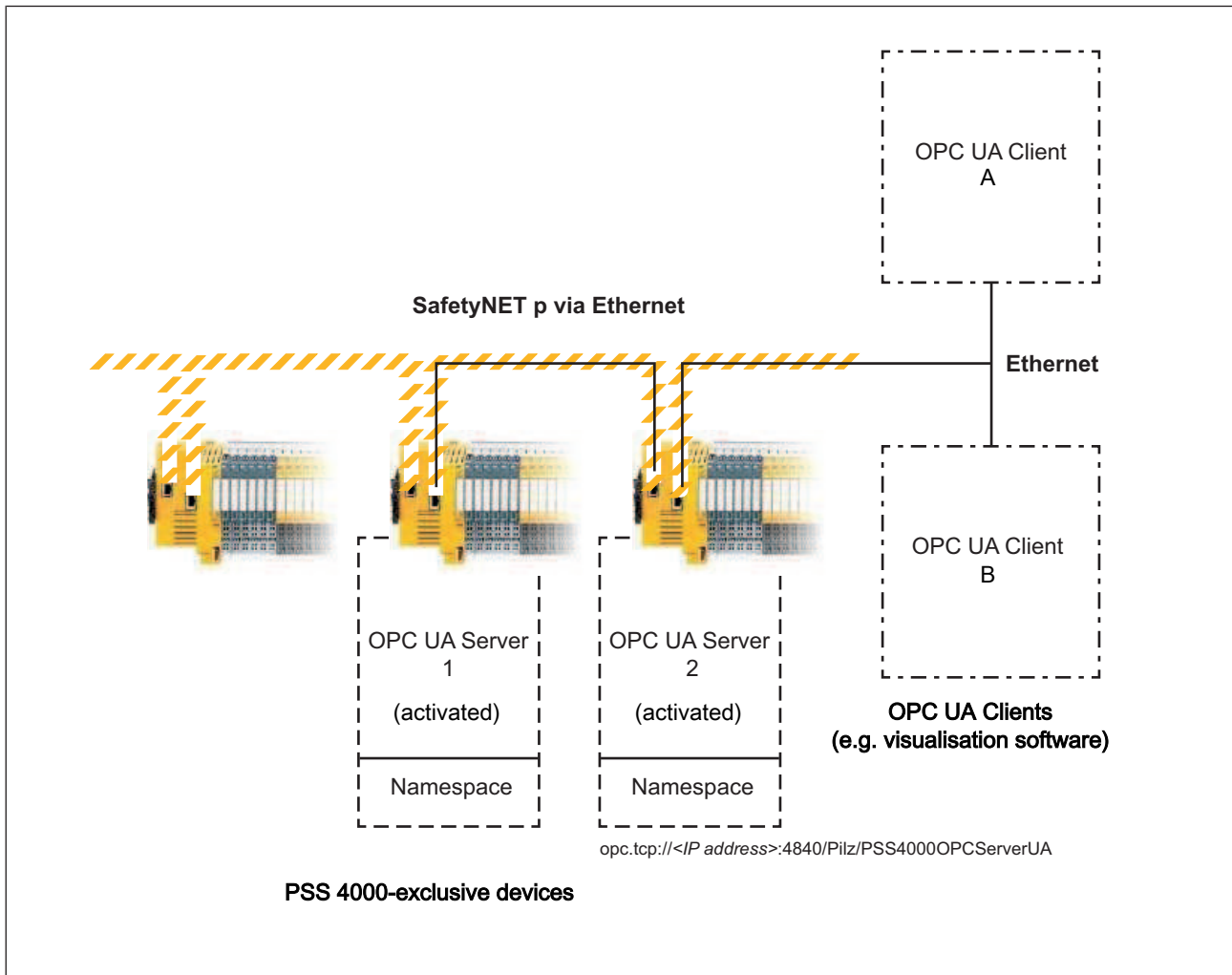


Fig.: Process visualisation with OPC UA Clients and the OPC UA Server of a PSS 4000-exclusive device (principle)

17.1.2 OPC UA properties

The PSS 4000 OPC Server UA supports the following OPC UA properties for access from OPC UA Clients:

- ▶ Supports the UA binary protocol via TCP
- ▶ End point URL of the PSS 4000 OPC Server UA (address in the network):
opc.tcp://<IP address>:<Port>/<Root name>

Note:

An end point URL must be specified manually. An end point URL cannot be obtained via Multicast DNS (mDNS).

<IP address> IPv4 address of the PSS 4000-exclusive device on which the OPC UA Server is located

Note: The IP address that was used during device naming for the PSS 4000-exclusive device is used.

<Port> 4840

Port number of the port that enables OPC UA Clients to access the supported Service Sets of the OPC UA Server

Note: The port number is permanently set on the PSS 4000-exclusive device and cannot be changed. Further information is available under [Access rights of OPC UA Clients \[711\]](#).

<Root name> Pilz/PSS4000OPCUAServer

- ▶ Supports Data Access (DA)

Note: Historical Access (HA) is not supported.

- ▶ Support for generic access to process data in the PSS 4000 namespace with pre-defined syntax
- ▶ Support for symbolic access to process data in the PSS 4000 namespace
- ▶ Support for OPC UA Services and Service Sets
(in accordance with OPC Unified Architecture – Part 4: Services)

Service Set	Supported services	Meaning
Discovery	▶ Multicast Discovery	Service for determining the implemented end points of an OPC UA Server
Secure Channel	▶ OpenSecureChannel ▶ CloseSecureChannel	Services for opening/closing a communication channel that guarantees security requirements (see Limit values [710])
Session	▶ CreateSession ▶ ActivateSession ▶ CloseSession ▶ Cancel	Connection management between OPC UA Clients and OPC UA Servers (see Limit values [710])

Service Set	Supported services	Meaning
NodeManagement	▶ AddNodes ▶ AddReferences ▶ DeleteNodes ▶ DeleteReferences	Node management for the nodes in the namespace and their references (see Limit values  710)
View	▶ Browse ▶ BrowseNext ▶ TranslateBrowsePathsToNodeIds	Services for navigation in the namespace
Attributes	▶ Read ▶ Write	Services for accessing attributes that are parts of nodes
Method	▶ Call	Service for calling methods
MonitoredItem	▶ CreateMonitoredItems ▶ ModifyMonitoredItems ▶ SetMonitoringMode ▶ SetTriggering ▶ DeleteMonitoredItems	See [1]
Subscription	▶ CreateSubscription ▶ ModifySubscription ▶ SetPublishingMode ▶ Publish ▶ Republish ▶ TransferSubscriptions ▶ DeleteSubscription	See [1]

[1] Service Sets **"Subscription"** and **"MonitoredItems"**

An OPC UA Client can subscribe to data provided by the OPC UA Server. The purpose of a subscription is to group data using monitored items.

Please note the limit values for the service sets "Subscription" and "MonitoredItem" (see [Limit values](#)  710).

17.1.3 Limit values



NOTICE

Validity of the stated limit values

Please note that only values within the stated ranges are permitted for the stated limit values.

The stated limit values are not guaranteed values. The stated limit values are supported in principle, but not in all combinations.

Name		Limit values
Service Set "Secure Channel"		
	SecureChannelNumber	0 ... 40
	SecurityTokenLifetime	1 ... 600 000 ms
Service Set "Session"		
	SessionNumber	0 ... 40
	SessionTimeout	1 ... 360 000 ms
Service Set "NodeManagement"		
	NodesPerRead	1 ... 500
	NodesPerWrite	1 ... 500
	NodesPerMethodCall	1 ... 10
	NodesPerBrowse	1 ... 100
	NodesPerTranslateBrowsePathsToNodeIds	1 ... 100
	MonitoredItemsPerCall	1 ... 500
	ReferencesPerNode	0 ... 100
Service Set "Subscription" (see also OPC UA properties [708])		
	SubscriptionNumber	0 ... 25
	PublishingIntervalLimits	100 ... 3 600 000 ms
Service Set "MonitoredItem" (see also OPC UA properties [708])		
	MonitoredItemNumber	0 ... 12 500
	MonitoredItemsPerSubscription	0 ... 500
	SamplingIntervalLimits	220 ... 1 800 000 ms
	QueueSizeLimits	1 ... 100

Note:

To ensure that changes to variable values are available on an OPC UA Client as quickly as possible, the delay must be kept to a minimum. To do this, create MonitoredItems with a Sampling Interval of 0 and a QueueSize of 1.

17.1.4 Access rights of OPC UA Clients

An OPC UA Client has read/write access (RW access) to the device-internal OPC UA Server. The port number for access by an OPC UA Client (RW Client) is permanently set on the PSS 4000 device (head module). The port number cannot be adjusted.

► Setting for the port number: 4840



NOTICE

Potential security risk due to write access by OPC UA Clients

An OPC UA Client (RW Client) with RW access to the device-internal OPC UA Server must not have the ability to perform unauthorised or unwanted write access.

Use a firewall to prevent unwanted or undesirable write access by OPC UA Clients to the device-internal OPC UA Server. Configure the firewall so that unauthorised access by a RW Client to a read/write end point is prevented.

Please also note the information and requirements under [Information security certificates](#) [📖 715] and [OPC UA commissioning mode for certificate management](#) [📖 715].

17.1.5 Security settings for end points

For end points, various encryption levels (Security Policy and Message Security Modes) are supported. You find an overview and further information on the currently available Security Policies at [Security Policies of the device-internal PSS 4000 OPC Server UA](#) [📖 716].

17.1.6 Diagnostics

The device's diagnostic messages and log entries can be retrieved in the PSS 4000 namespace of the device-internal OPC UA Server. This is also possible when there is no device project stored on the device.

Diagnostic list:

- ▶ In the namespace under:
pss4000.net.<DeviceName>.Device.Diag.List:
- ▶ Maximum 128 messages

Diagnostic log:

- ▶ In the namespace under:
pss4000.net.<DeviceName>.Device.Diag.Log:
- ▶ Maximum 256 entries.

Language:

- ▶ No device projects exists
The diagnostics will be displayed in English.
- ▶ Device projects exists
The diagnostics will be displayed in the language that was used as the diagnostic language when building the project.

17.1.7 Audit log

The following actions are logged in the diagnostic log:

- ▶ The OPC UA Server has switched to commissioning mode due to a user command.
- ▶ The OPC UA Server has switched to commissioning mode because there are no trusted certificates available.
- ▶ The OPC UA Server has exited commissioning mode because trusted certificates are now available.
- ▶ The OPC UA Server has exited commissioning mode because the maximum duration of commissioning has been exceeded.
- ▶ The commissioning timeout of the OPC UA Server was changed from ... s to ... s.
- ▶ The Trust List was changed. Trusted Certs: ..., Trusted CRLs: ..., Issuer Certs: ..., Issuer CRLs:
- ▶ Could not update the Trust List because a certificate is invalid. (Error code: ...; Subject: ...)
- ▶ The current size of the Trust List (... exceeds 80 % of the maximum size (... Bytes).
- ▶ The certificate for the OPC UA application instance was changed.
- ▶ The certificate for the OPC UA application instance and the corresponding private key were changed.
- ▶ The new certificate for the OPC UA application instance was rejected because it is invalid.
- ▶ The certificate for the OPC UA peer with IP address ... is invalid. (Error code: ...)

The same entry can be logged a maximum of 3 times per minute. After the last entry, it is 1 minute before the same entry can be logged again. This prevents an attacker from triggering a large number of entries and thereby pushing a relevant security entry out of the diagnostic log (maximum 256 entries).

17.2 System requirements

The device-internal OPC UA Server is available on PSS 4000-exclusive devices with system section OPC Server UA (see [System sections \[📖 22\]](#)).

17.3 Activation/deactivation of the device-internal OPC UA Server

The device-internal OPC UA Server is always deactivated on a device in its delivery state or after an original reset. To be more precise, the OPC Server port through which Clients access the Server's data is deactivated.

The device-internal OPC UA Server is activated/deactivated in the hardware configuration and firmware configuration:

► Hardware configuration

In PAS4000, the property **OPC Server port activated** is set to "Yes" or "No" in the device properties. To do this, right-click in the Project Manager on the device (not on the head module) and select **Properties**.

Default setting: Not activated

► Firmware configuration

Certain device properties must also be configurable if no device project exists. These properties are summarised in the firmware configuration. The property **OPC Server port activated** is part of the firmware configuration. The firmware configuration is stored on the device and SD card.

Default setting: Not activated

– During device naming

During device naming with the PSS 4000 project, the firmware configuration is transferred to the device together with the device naming data. Information as to whether the OPC Server port should be activated is taken from the device's hardware configuration.

– Through a specific online action

When the properties of a device are displayed in the Online Network Editor of PAS4000, the property **OPC Server port activated** can be changed. To do this, click on the name of the property.

If the OPC UA Server is deactivated in the firmware configuration although it is activated in the hardware configuration, a malfunction occurs. A corresponding diagnostic message is displayed and the OPC UA Server port is closed.

If the OPC UA Server is activated in the firmware configuration although it is deactivated in the hardware configuration, only a diagnostic message of "Information" severity is displayed.

17.4 Ethernet mode

The device-internal OPC UA Server can only be operated via the "Switched mode" Ethernet mode (see [Ports' Ethernet mode](#)  138]; i.e. advanced network settings such as VLAN, for example, are not available.

Switched mode is the default setting.

17.5 Information security certificates

Networked automation systems place demands on information security. Information security encompasses the authenticity, confidentiality, integrity and availability of data as well as protection of access to data.

On Pilz devices with an OPC UA application, information security is established using certificates compliant with the X.509 standard.

A separate certificate is required for each subscriber in the Client-Server communication. An OPC UA Client can only communicate securely with the OPC UA Server if the OPC UA Server trusts the certificate of the OPC UA Client and the OPC UA Client trusts the certificate of the OPC UA Server.

The certificates for information security are managed via the Pilz NET Configurator. The Pilz NET Configurator is installed together with the system software PAS4000 and can be called up in the Online Network Editor ("Manage certificates" button). The Pilz NET Configurator has its own help, in which management of the certificates is described.

17.5.1 OPC UA commissioning mode for certificate management

The OPC UA Server of a PSS 4000-exclusive device can be switched to a commissioning mode. In commissioning mode, any Clients can connect to the OPC UA Server.



NOTICE

Potential security risk due to OPC UA commissioning mode

In OPC UA commissioning mode, there is a risk that an unauthorised user or attacker could install a certificate they control and therefore compromise the system. For this reason, only use commissioning mode in a trusted network environment.

Commissioning mode is required, for example, if the certificates on the OPC UA Server are to be managed using the Pilz NET Configurator tool and the OPC UA Server does not yet trust the tool instance.

OPC UA commissioning mode only affects the device-internal PSS 4000 OPC Server UA of a PSS 4000-exclusive device. Operation of the other system sections is not affected.

If the OPC UA Server is in commissioning mode and no other diagnostic messages are present, then the DIAG LED flashes red.

Switch to commissioning mode

► Automatic

If the Trusted Certs list in the Trust List is empty, the OPC UA Server automatically switches to commissioning mode. This happens in the following cases:

- A device is started in its delivery status (new device).
- The device has undergone an original reset.
- The last trustworthy certificate on the device has been deleted.

Commissioning mode is exited automatically as soon as there is a certificate in the list.

► Manual

Commissioning mode can be started manually using the reset button. To do this, press the reset button twice within 3 s (press - release - press - release).

Commissioning mode is exited automatically when the commissioning timeout has expired. The duration of the commissioning timeout can be configured in the Pilz NET Configurator (120 ... 3600 s).

If commissioning mode is active and the reset button is pressed twice again (press - release - press - release), the device remains in commissioning mode and the commissioning timeout is restarted.

If the OPC UA Server has automatically switched to commissioning mode and the reset button is then pressed twice, the rules for manually switching to commissioning mode will apply. This means that even if certificates are added to the Trusted Certs list, commissioning mode is only exited once the commissioning timeout has expired.

As long as certificates are located in the Trusted Certs list on the PSS 4000-exclusive device, commissioning mode can only be started manually (see Manual start of OPC UA commissioning mode).

17.5.2 Security Policies of the device-internal PSS 4000 OPC Server UA

Various Security Policies can be activated for the device-internal PSS 4000 OPC Server UA. If several Security Policies are activated, all activated Security Policies can be used for communication with the OPC UA Clients.



NOTICE

Potential security risk due to the Security Policy "None"

It is possible to use the Security Policy "None". In this case, communication between the OPC UA Server and OPC UA Client is completely unprotected.

When carrying out your security risk analysis, consider that using the Security Policy "None" can represent a security vulnerability for cyber threats. It is preferable to use one of the Security Policies with high security.

Supported Security Policies and Message Security Modes:

- ▶ None
 - Not recommended
 - Default setting: not activated
- ▶ Basic256-Sha256 Sign
 - Default setting: not activated
- ▶ Basic256-Sha256 SignAndEncrypt
 - Default setting: activated, cannot be deactivated

17.5.3 RSA key size

The Rivest-Shamir-Adleman (RSA) algorithm is used to encrypt certificates. The used key size can be configured in the Pilz NET Configurator.

Supported key sizes:

- ▶ 2048 Bit
- ▶ 3072 Bit

► 4096 Bit

17.6 Firewall configuration

A firewall or any other security mechanisms on the OPC UA Client and network may prevent the device-internal OPC UA Server from communicating correctly with the connected devices. In this case it is necessary to adjust the security settings and firewall settings.

Firewalls allow access to explicitly registered programs. The following programs must have access:

- ▶ All OPC UA Clients with access to the device-internal OPC UA Server of a PSS 4000-exclusive device

Ports must also be enabled for data exchange:

- ▶ Port number of the device-internal OPC UA Server
Fixed setting: Port 4840
- ▶ Port number of the OPC Server UA - Discovery Servers:
Fixed setting: Port 4840
- ▶ Protocol: TCP

These enables grant access rights to the device-internal OPC UA Server for other users on the network or from the Internet. A network administrator can optimise the Firewall settings to protect the device-internal OPC UA Server from unauthorised access.

Please also refer to the network data in the operating manual for the relevant PSS 4000-exclusive device.

17.7 Data type assignment

Various data types are defined through the OPC UA standard. The table below contains the data types supported by the device-internal OPC UA Server and their assignment to the data types available in the user program on the PSS 4000:

PSS 4000 namespace	OPC UA	User program (PSS 4000)
Boolean	Boolean	BOOL
SByte	SByte	SINT
Byte	Byte	BYTE
		USINT
Double	Double	LREAL
		TIME
Float	Float	REAL
Int16	Int16	INT
Int32	Int32	DINT
		FIXED
Int64	Int64	LINT
		LFIXED
String (language-dependent text, e.g. diagnostic messages)	LocalizedText	- - -
String (Variables from the user pro- gram)	String	USTRING
		UCHAR
UInt16	UInt16	WORD
		UINT
UInt32	UInt32	DWORD
		UDINT
UInt64	UInt64	LWORD
		ULINT
DateTime	DateTime	DATE
		TIME_OF_DAY
		DATE_AND_TIME

17.7.1 Access to data in the PSS 4000 namespace

The OPC UA Server of a PSS 4000-exclusive device collects the general device-based project data and the project data from the device project and presents it in the PSS 4000 namespace. The PSS 4000 namespace is a specific area with a predefined structure. An OPC UA Client can access the data in the PSS 4000 namespace of the PSS 4000-exclusive device via this predefined structure.

Further information on the PSS 4000 namespace and its structure can be found under [PSS 4000 namespace](#) [ 741].

17.7.2 Special features when accessing arrays in the namespace

An array in the namespace is usually represented as a single node; i.e. the individual elements of an array are not available to access as separate nodes. On a PSS 4000-exclusive device with system section OPC Server UA, only the complete array can be accessed ("Complete ARRAY" access type).

Note: There are OPC UA Clients that are able to use the IndexRange function to access the individual elements of an array. This access type is also called index-based access. A PSS 4000-exclusive devices with system section OPC Server UA does not support the "Index-based" access type.

17.8 Network data (OPC Server UA)

You'll find the network data for the device-internal OPC UA Server of a PSS 4000-exclusive device in the operating manual for the relevant device.

Also refer to the information under [Firewall configuration](#) [ 699].

18 Diagnostic basics

Effective diagnostics are required in order to keep machine downtimes to a minimum. Diagnostics must provide all the information necessary to recover errors quickly, prevent errors and analyse machine conditions.

The automation system PSS 4000 distinguishes between system diagnostics and process diagnostics.

► System diagnostics

Diagnostics on hardware and firmware, including errors in the hardware configuration or user program.

The system diagnostics are provided by Pilz in full. Only a few adjustments can be made by the user, e.g. entering additional information for location information.

► Process diagnostics

Diagnostics on procedures and states. The process diagnostics are created by the user within the user program. Pilz blocks are supplied with process diagnostics, which users can adapt to suit their own requirements.

During operation, system messages and process messages are entered in a [Diagnostic list](#)  723]. Each device has its own device diagnostic list. The diagnostic server combines the device diagnostic lists of all the devices in the project to form the project diagnostic list.

The diagnostic list only contains current messages. Once an error has been recovered, the message is deleted from the diagnostic list.

The order in which the diagnostic messages appear in the diagnostic list is determined by the priority of the diagnostic message. That makes it easy to see which message is the most important.

In system diagnostics, a distinction is made between the first error and subsequent errors. Only the first error is reported. This avoids unnecessary messages in the diagnostic list.

If a PSS 4000-exclusive device exits the RUN state, the system sections switch to a STOP state at a slight offset. Diagnostic messages may be displayed as a result, which appear superfluous but actually represent the operating states correctly.

As well as describing the event that has occurred and indicating its location, diagnostic messages also contain suggested remedies. These tell you what to do, which area is affected and who is responsible for rectifying the problem.

The [Diagnostic log](#)  736] records when a diagnostic message arrived and when it was cleared. The diagnostic log also contains information on any changes in the operating state, e.g. if a device has been restarted, as well as information for Pilz technical support. Each device has its own device diagnostic log. The diagnostic server combines the device diagnostic logs of all the devices in the project to form the project diagnostic log.

The [Diagnostic server](#)  739] forwards the project diagnostic list and log to the diagnostic clients (e.g. PSS 4000 OPC Server). The diagnostic clients then provide the information to the displays (e.g. PMI with PASvisu project).

The project diagnostic list and project diagnostic log are also displayed in PAS4000. The diagnostic list and diagnostic log in PAS4000 may vary slightly from the diagnostic list and diagnostic log on a display (e.g. PMI with PASvisu project), because they also contain messages/entries regarding online operations, for example.

The device diagnostic lists and logs are not displayed.

18.1 Diagnostic list

During operation, [Diagnostic messages](#)  724 are entered in diagnostic lists.

Diagnostic lists only contain current messages, i.e. diagnostic messages with an "active" state (see [State of diagnostic messages](#)  726).

A new diagnostic message is entered in the diagnostic list if the state of the diagnostic message switches to "active". The diagnostic message is deleted once its state switches to "inactive".

Each device has its own device diagnostic list. The device diagnostic lists from all the devices in a project are combined within a project diagnostic list.

► Device diagnostic list

- Generated by the device's firmware
- Contains a device's diagnostic messages (raw data only, no texts)
- Memory for at least 128 diagnostic messages
- If the maximum number of diagnostic messages is reached, the diagnostic message with the lowest priority is deleted
- If so many process diagnostic messages occur within a short time that they can't be stored quickly enough, a corresponding diagnostic message is entered in the device diagnostic list and all other process diagnostic messages are ignored
- If so many system diagnostic messages occur within a short time that they can't be stored quickly enough, a corresponding diagnostic message is entered in the device diagnostic list; an attempt will still be made to process as many system diagnostic messages as possible, but some messages may be missed
- Deleted in the event of an original reset
- Cannot be displayed; used as the basis for the project diagnostic list

► Project diagnostic list

- Generated by the [Diagnostic server](#)  739
- Contains plain text diagnostic messages for all the devices in a project
- Memory for at least 256 diagnostic messages
- If the maximum number of diagnostic messages is reached, the diagnostic message with the lowest priority is deleted
- Always up-to-date as it is continuously synchronised with the device diagnostic lists
- Should the diagnostic server lose the connection to a device, all the device's diagnostic messages are deleted from the project diagnostic list and a corresponding diagnostic message is entered in the project diagnostic list
- The diagnostic list in PAS4000 may vary slightly from the diagnostic list on a display (e.g. PMI with PASvisu project), because it also contains messages regarding online actions, for example

18.1.1 Diagnostic messages

The following diagnostic messages are available:

- ▶ System messages

Message that is issued when an event occurs in system diagnostics.

- ▶ Process messages

Message that is issued when an event occurs in process diagnostics, i.e. the set condition is met.

During operation, diagnostic messages are entered in a diagnostic list.

A diagnostic message contains a description of the event. Each diagnostic message includes a remedy. The remedy describes how to react to the event. The remedy contains a maximum of 8 steps, the so-called "Actions".

Components of a diagnostic message:

- ▶ Time stamp

Time at which the diagnostic message was activated. In the diagnostic message, the time is stored in universal time (UTC). When the diagnostic message is displayed, the time stamp is displayed in the local time of the display device. The local time is calculated from the world time, based on the time zone set on the display device.

- ▶ Diagnostic identifier

In system diagnostics, the diagnostic identifier is used for unique identification of a diagnostic message. In process diagnostics, the diagnostic identifier is used for unique identification of the diagnostic item responsible for the diagnostic message.

(see [Diagnostic identifier](#)  734)

- ▶ Severity and priority

Importance of the diagnostic message

(see [Severity and priorities](#)  727)

- ▶ Scope

Approximate description of the kind of event

(see [Scopes](#)  729)

- ▶ Description (message text)

Description of what has happened

- ▶ Location information

Information about the location at which the event occurred (see [Location information](#)  730).

The diagnostic message's location information contains the following details:

- Device
- Equip-ID
- Address
- Location description

- ▶ Parameter list

Contains all the information required by technical support to evaluate a diagnostic message. This is the diagnostic identifier and all parameters.

- ▶ State

Indicates the state of the diagnostic message: "active" or "inactive"
(see [State of diagnostic messages](#)  726])

Action components

- ▶ Scope

Approximate description of the kind of action
(see [Scopes](#)  729])

- ▶ Responsibility

Information about who should undertake the action
(see [Responsibilities](#)  728])

- ▶ Description (action text)

Description of what has to be done

- ▶ Location information

Information about the location at which the action is to be taken (see [Location information](#)  730]).

The location information for action components contains the following details:

- Device
- Equip-ID
- Address
- Location description

18.1.1.1 State of diagnostic messages

The state of a diagnostic message depends on the event that triggered it.

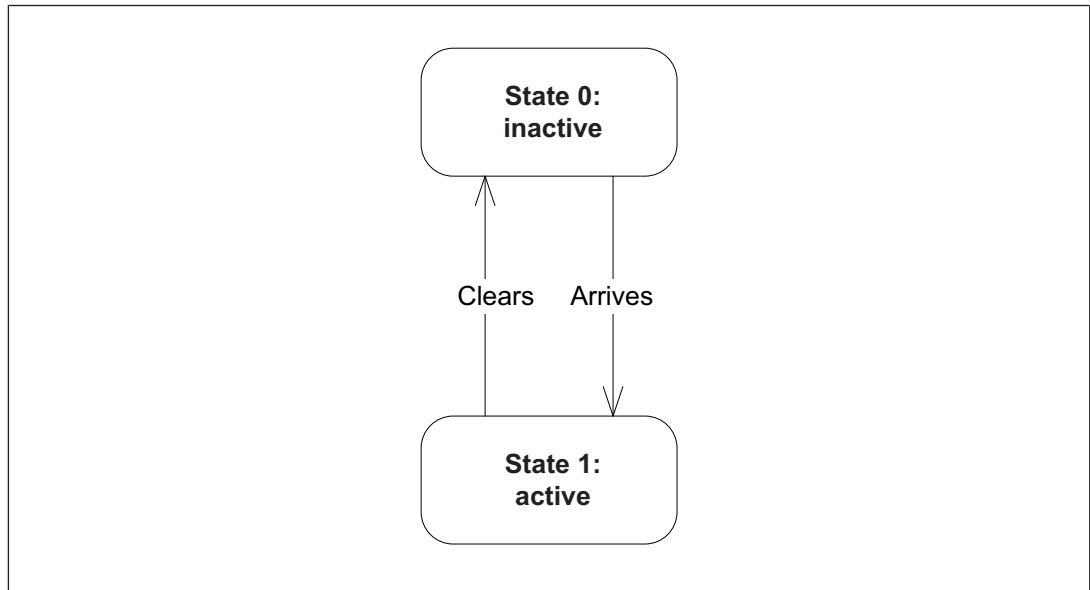


Fig.: State of diagnostic messages

► System diagnostics

If an event is present (e.g. short circuit at an input), the diagnostic message becomes "active" (change of state: "arrives"). If the fault has been rectified, the diagnostic message becomes "inactive" (status change: "clears").

► Process diagnostics

If the programmed condition is met, the diagnostic message becomes "active" (change of state: "arrives"). If the condition is no longer met, the diagnostic message becomes "inactive" (change of state: "clears").

The diagnostic list only contains diagnostic messages with the state 1.

Each time the state of a diagnostic message changes, a log entry is triggered in the diagnostic log.

Changes in state are symbolised by the following character strings:

- <+> for "arrived"
- <-> for "cleared"

18.1.1.2 Severity and priority

Diagnostic messages are differentiated according to their importance. They are divided into three levels of severity:



Error

Diagnostic message to which the user must react immediately. This diagnostic message provides information on system conditions in which one or more functions have failed or are adversely affected.



Warning

Diagnostic message which demands the user's attention. This diagnostic message warns the user about potential or imminent critical system conditions.



Status information

Diagnostic message which describes the current condition of the process and/or system.. This diagnostic message is provided for information; operation is neither impeded nor adversely affected.

Priorities allow a more precise classification. Each level of severity has 5 priorities. Priority 1 is the highest priority and priority 15 the lowest.

Severity	Priority	Name of priority
Error	1	
	2	
	3	Error
	4	
	5	
Warning	6	
	7	
	8	Warning
	9	
	10	
Status information	11	
	12	
	13	Status information
	14	
	15	

The user can assign names to these priorities. The names for priority 3 "Error", priority 8 "Warning" and priority 13 "Status information" are pre-defined by Pilz because they are used in system diagnostics and in Pilz-generated process diagnostics.

These pre-defined names can be modified by the user but cannot be deleted.

18.1.1.3 Responsibilities

The responsibility stands for the person or group who is supposed to perform the remedial action.

Some responsibilities are pre-defined by Pilz and a maximum of 15 responsibilities can be defined by the user. The pre-defined responsibilities are used in system diagnostics and in Pilz-generated process diagnostics. The user can change the name of the pre-defined responsibilities, but cannot delete them.

Pre-defined responsibilities:

► Operator

This action involves the release, actuation or operation of a safety device.

Responsible person: Plant operator, for example

► Electrical maintenance engineer

This action refers to the mechanics/electrics/electronics outside the Pilz devices, e.g. rectify short circuit, check supply voltage, check switch contacts.

Responsible person: Maintenance engineer, for example

► Device exchange

The device is defective and must be exchanged.

Responsible person: Maintenance engineer, for example

► Project engineer

This action can only be performed with PAS4000 or requires some programming knowledge, e.g. check or adapt hardware configuration, check or adapt programming.

Responsible person: Maintenance engineer, programmer, project engineer, for example

► Communication

This action can be performed by users with a technical understanding of SafetyNET p/IP connections/fieldbuses, who can check the bus load or check cable lengths, for example.

Responsible person: Maintenance engineer, bus specialist, for example

► ---

No details of responsibility are possible or necessary.

18.1.1.4

Scopes

The scope roughly describes the kind of event/action. The scope is stated for diagnostic messages and remedial actions.

Some scopes are pre-defined by Pilz and a maximum of 15 scopes can be defined by the user. The pre-defined scopes are used in system diagnostics and in Pilz-generated process diagnostics. The user can change the name of the pre-defined scopes, but cannot delete them.

Pre-defined scopes:

► Hardware

Affects the mechanics, electrics, electronics or operating system.

Examples: Defective device, periphery error (e.g. wiring, actuator/sensor or supply voltage is faulty), bus error (e.g. error in SafetyNET p or Ethernet)

► Project configuration

Affects the user program and hardware configuration.

Examples: Division by zero in the user program

► Operation

Affects sequence or operation.

Examples: Sequence error (cylinder fails to reach an end position), safety device was triggered (e.g. E-STOP operated)

► Plant

Affects the hardware or project configuration.

► ---

No details of the scope are possible or necessary.

18.1.1.5 Location information

Location information is available for diagnostic messages and for the individual remedial actions. The diagnostic message contains details of the location in which the event occurred. The information for the action includes details of where the action is to be taken.

The location information contains the following details:

- ▶ **Device**
Relevant element
- ▶ **Equip-ID**
The equipment identifier is used to identify a piece of hardware in a project and should be unique.
- ▶ **Address**
The address is generated automatically.
- ▶ **Location description**
Additional information about a location, e.g. information about the control cabinet in which a device is located.

The information content depends on the location. The following locations are possible:

- ▶ **Module/device**
- ▶ **Terminal**
- ▶ **Sensor/actuator**
- ▶ **Block instance**
- ▶ **Position in the user program**
- ▶ **Other locations**

The individual locations and location information details are described below.



INFORMATION

If "<???" is displayed in the diagnostic list or diagnostic log instead of location information, either the current project has not been downloaded to all devices or not all the I/O mappings have been performed (see [Diagnostic Server](#) [ 739]).

Module/device

Only used in system diagnostic messages and log entries.

- ▶ **Device**
Text that the user has entered for the module/device as "Text for device" location information; the default setting is the product type of the module/device (e.g. "PSSu E F 4DI"). In rare cases, on PSS 4000-exclusive devices it is not possible to determine which module has triggered a diagnostic message. In this case, the text "anonymous module" is displayed as the device.
- ▶ **Equip-ID**
Equip-ID of the module/device (e.g. "-4A1"), entered by the user
- ▶ **Address**
Automatically generated address of the module/device

- Location description

Additional information on the module/device, entered by the user

Terminal

In the system diagnostics or process diagnostics, if the underlying PI variable is mapped to an I/O datum from the module bus:

- Device

"Device" location information for the module/device on which the terminal is located (see "Device" under "Module/device")

- Equip-ID

Equip-ID of the module/device on which the terminal is located, including details of the terminal (e.g. "-4A1:12", whereby "12" refers to the terminal)

- Address

Automatically generated terminal address

- Location description

Additional information on the module/device, entered by the user; no location description can be entered for the terminal itself

In the process diagnostics, if the underlying PI variable is mapped to an external I/O datum:

- Device

"Device" location information for the module/device containing the external I/O datum (see "Device" under "Module/device")

- Equip-ID

Equip-ID of module/device containing the external I/O datum (see "Equip-ID" under "Module/device")

- Address

Automatically generated address of the external I/O datum

- Location description

Location description of the module/device containing the external I/O datum (see "Location description" under "Module/device")

In the process diagnostics, if the underlying PI variable is mapped to another PI variable:

- Device

"Device" location information for the module/device on which the resource of the mapped PI variable is located (see "Device" under "Module/device")

- Equip-ID

Equip-ID of module/device on which the resource of the mapped PI variable is located (see "Equip-ID" under "Module/device")

- Address

Automatically generated address of the mapped PI variable

- Location description

Location description of the module/device on which the resource of the mapped PI variable is located (see "Location description" under "Module/device")

Sensor/actuator

In the system diagnostics or process diagnostics, if the underlying PI variable is mapped to an I/O datum from the module bus:

- **Device**
Text that the user has entered for the sensor/actuator as the location information "Text for Device"
- **Equip-ID**
Equip-ID of the sensor/actuator, entered by the user
- **Address**
Automatically generated address of the terminal to which the sensor/actuator is connected
- **Location description**
Additional information on the sensor/actuator, entered by the user

In the process diagnostics, if the underlying PI variable is mapped to an external I/O datum:

- **Device**
"Device" location information for the module/device containing the external I/O datum (see "Device" under "Module/device")
- **Equip-ID**
Equip-ID of module/device containing the external I/O datum (see "Equip-ID" under "Module/device")
- **Address**
Automatically generated address of the external I/O datum
- **Location description**
Location description of the module/device containing the external I/O datum (see "Location description" under "Module/device")

In the process diagnostics, if the underlying PI variable is mapped to another PI variable:

- **Device**
"Device" location information for the module/device on which the resource of the mapped PI variable is located (see "Device" under "Module/device")
- **Equip-ID**
Equip-ID of module/device on which the resource of the mapped PI variable is located (see "Equip-ID" under "Module/device")
- **Address**
Automatically generated address of the mapped PI variable
- **Location description**
Location description of the module/device on which the resource of the mapped PI variable is located (see "Location description" under "Module/device")

Block instance

Only used in process diagnostic messages and log entries.

- **Device**
Name of the type of Multi block/POU
- **Equip-ID**
Not used

► Address

FQN (full name in the PSS 4000 namespace) of the Multi block/POU

► Location description

Not used

Position in the user program

Only used in system diagnostic messages and log entries.

► Device

Name of the type of Multi block/POU

► Equip-ID

Not used

► Address

FQN (full name in the PSS 4000 namespace) of the Multi block/POU, with additional positional data

Positional data on the Multi block: X and Y co-ordinate in the Multi Editor; X is the column number and Y is the row number. The row and column numbers can be shown in the Multi Editor.

Positional data with IL-POU and STL-POU: Row number

Positional data on LD-POU: Network number

If a Multi block/POU is sealed, the FQN (full name in the PSS 4000 namespace) will end with details of the last unsealed Multi block/POU, before the affected Multi block/POU.

The position names the point at which the sealed Multi block/POU is called.

► Location description

Not used

Other locations

If the location information listed above isn't suitable, the location information can be freely defined.

► Device

Process diagnostics: Any; defined in the type or instance of the diagnostic item

System diagnostics: Names the location (e.g. "Hardware configuration" or "User program")

► Equip-ID

Process diagnostics: Any; defined in the type or instance of the diagnostic item

System diagnostics: Not used

► Address

Not used

► Location description

Process diagnostics: Any; defined in the type or instance of the diagnostic item

System diagnostics: Not used

18.1.1.6 Diagnostic identifier

In system diagnostics, the diagnostic identifier is used for unique identification of a diagnostic message. In process diagnostics, the diagnostic identifier is used for unique identification of the diagnostic item responsible for the diagnostic message. In process diagnostics, the uniqueness is limited to the project.

The diagnostic identifier consists of three parts:

<Kind> - *<Class>* - *<Number>*

► Kind

A letter indicates the kind of diagnostics

S: System diagnostics

P: Process diagnostics, generated by Pilz

C: Process diagnostics, generated by the user

► Class

A decimal figure of a maximum of 4 digits indicates the class. The meaning of class depends on the kind of diagnostics.

– System diagnostics

The class defines the kind of error, e.g. device error, user program

Classes in system diagnostics:

100 ... 110 Device error

120 Error caused by a device project

121 Error caused by the naming data

122 Error caused by a boot command

123 Error caused by firmware update

200 ... 220 Error in the hardware registry/hardware configuration

300 Error in the FS or ST resource system section

303 Licensing is missing

310 Error in the VISU resource system section

400 Periphery error

410 Faulty supply voltage

600 ... 602 Error in the FS or ST SafetyNET p system section

700 ... 710 Error in the FS or ST module bus system section

801 Internal error in PAS4000

908 Ethernet error

910, 911 Error in the IP connections system section

912, 913 Error in the PROFIBUS-DP Slave system section

914, 915 Error in the EtherNet/IP Adapter system section

916, 917 Error in the PROFIsafe system section

918, 919 Error in the PROFINET IO DEVICE system section

9000 Operational entries in the diagnostic log

9001 Self test entries in the diagnostic log

9900 Debugging entries in the diagnostic log

- Process diagnostics, generated by Pilz

The class indicates the Pilz block to which the diagnostic message belongs. The class refers to the type of block, not to an instance.

- Process diagnostics, generated by the user

The user can decide what the class means. We recommend that the class is used along similar lines to that of the Pilz process diagnostics.

► Number

Unique number of a diagnostic message/diagnostic item within a class. The number is a 4-digit decimal figure.

18.2 Diagnostic log

The diagnostic log records when a diagnostic message arrived or when it was cleared. The diagnostic log also contains information on any changes in the operating state, e.g. if a device has been restarted, as well as information for Pilz technical support.

Each device has its own device diagnostic log. The [Log entries \[737\]](#) for all the devices in a project are combined within the project diagnostic log.

► Device diagnostic log

- Generated by the device's firmware
- Contains a device's log entries (raw data only, no texts)
- Memory for at least 256 log entries;
one memory area is reserved for system, process and self test entries and the other memory area for operational entries (see [Log entries \[737\]](#))
- If the maximum number of log entries has been reached, the oldest log entry is deleted;
this is the case respectively within the memory area for system, process and self test entries and the memory area for operational entries
- Cannot be displayed; used as the basis for the project diagnostic log

► Project diagnostic log

- Generated by the [Diagnostic server \[739\]](#)
- Contains plain text log entries for all the devices in a project
- Memory for at least 4096 log entries;
one memory area is reserved for system, process and self test entries and the other memory area for operational entries (see [Log entries \[737\]](#))
- If the maximum number of log entries has been reached, the oldest log entry is deleted;
this is the case respectively within the memory area for system, process and self test entries and the memory area for operational entries. (This is why a project diagnostic log may contain very old operational entries while at the same time containing new system, process and self test entries or vice versa.)
- Always up-to-date as it is continuously synchronised with the device diagnostic logs
- Should the diagnostic server lose the connection to a device, all of the device's previous log entries are retained in the project diagnostic log. There will also be a log entry showing the loss of the connection.
- The diagnostic log in PAS4000 may vary slightly from the diagnostic log on a display (e.g. PMI with PASvisu project), because it also contains entries regarding online actions, for example

18.2.1 Log entries

The following log entries are available:

- ▶ System entries

Information about system messages changing state, i.e. when a system message arrived or when it was cleared. System entries are part of the system diagnostics.

- ▶ Process entries

Information about process messages changing state, i.e. when a process message arrived or when it was cleared. Process entries are part of the process diagnostics.

- ▶ Operational entries

Information about changes in the operating status (e.g. when a device was restarted) or any other changes relevant to the operation (e.g. executing a project download). Operational entries are part of the system diagnostics.

- ▶ Self test entries

Information about a fault occurring during the self test in the "Booting base system" operating status, which has meant that the device could not be started.

- ▶ Debugging entries

Information for Pilz technical support. Debugging entries are part of the system diagnostics and are only visible when enabled by technical support.

During operation, log entries are entered in the diagnostic log.

With system entries and process entries, the log entry matches the diagnostic message, except that some information is left out, such as the remedy for example. The time stamp means something different and the change in state is displayed rather than the state.

Components of a log entry (operational entries and debugging entries are not included in the list):

- ▶ Time stamp

Time at which the state of the diagnostic message changed (see [State of diagnostic messages](#)  726]). In the log entry, the time is stored in universal time (UTC). When the log entry is displayed, the time stamp is displayed in the local time of the display device. The local time is calculated from the world time, based on the time zone set on the display device.

- ▶ Change of state

Symbol for change of state (see [State of diagnostic messages](#)  726])

<+> for "arrived"

<-> for "cleared"

or

<#> for operational entries, self test entries and debugging entries

- ▶ Diagnostic identifier

Diagnostic identifier for the diagnostic message (see [Diagnostic identifier](#)  734])

- ▶ Severity and priority

Importance of the diagnostic message (see [Severity and priorities](#)  727])

- ▶ Scope

Approximate description of the kind of event (see [Scopes](#)  729])

► Description

Description of what has happened

► Location information

Information about the location at which the event occurred (see [Location information](#) [ 730]).

The diagnostic log's location information contains the following details:

- Device
- Equip-ID
- Address
- Location description

18.3 Diagnostic server

A diagnostic server is required if you wish to display a project's diagnostic data.

The diagnostic server forwards the project diagnostic list and log to the diagnostic clients (e.g. PSS 4000 OPC Server). The diagnostic clients then provide the information to the displays (e.g. PMI with PASvisu project).

The diagnostic server gathers the raw data in the device diagnostic lists and device diagnostic logs of every device in the project and uses it to generate the project diagnostic list and project diagnostic log in plain text. In other words, the diagnostic server takes the references to message texts and location information in the device diagnostic lists and device diagnostic logs and replaces them with legible texts.

The diagnostic server always needs the current project data: user program, hardware configuration, resource assignment and I/O mapping.

If the current project data is unavailable to the diagnostic server, references to message texts and location information cannot be replaced correctly. In this case they will be replaced with the wrong message texts/location information or by "<???" if no text is available. This can also happen if the diagnostic server has the current project data, but a device within the project is still operating with an older version of the project.

► Show on display

If the project diagnostic list and log are to be shown on a display (e.g. PMI with PASvisu project), then the project will need an OPC Server (PSS 4000 OPC Server UA).

The PSS 4000 device on which the OPC Server is installed also provides the diagnostic server system section automatically. Each OPC Server has its own diagnostic server.

The texts displayed for system diagnostics and process diagnostics are those which were available when the project was built in PAS4000 and which were transferred when the project was downloaded.

► Display in PAS4000

PAS4000 has its own diagnostic server.

The diagnostic texts supplied with PAS4000 are displayed for system diagnostics. The texts displayed for process diagnostics are the diagnostic texts which were available when the project was built.



INFORMATION

The diagnostic list and diagnostic log are displayed correctly if the current project has been downloaded to the diagnostic server and to all the devices in the project.

If the computer on which the OPC Server is installed is used as the diagnostic server, you will also need to download the current project to this computer (this happens automatically during a direct project download if the OPC Server is connected to SafetyNET p during download).

We also recommend that the version of the PSS 4000 firmware on the device containing the diagnostic server should be equal to or higher than the version in the project.

If the texts are to be displayed in PAS4000, you will need to activate the current project in PAS4000. Always use the latest version of PAS4000.

After an extended period of operation, log entries which are no longer available on the project devices may still be stored in the project diagnostic log on the diagnostic server. These old log entries will be deleted when the device with the diagnostic server has a warm reset or when the PAS4000 is cold started.

19 PSS 4000 namespace

The PSS 4000 OPC Server collects the general project data and device-based project data from a PSS 4000 project and presents it in the PSS 4000 namespace. The PSS 4000 namespace is a special memory area with a predefined structure. An OPC Client can access the data in the PSS 4000 namespace via this predefined structure.

For the most part the namespace can be viewed using the browser on an OPC Client.

The general and device-based project data that was transferred with the device project the last time a PSS 4000 project was downloaded is always available to the PSS 4000 OPC Server in the PSS 4000 namespace.

Access to information in the PSS 4000 namespace

An OPC Client can access the information in the PSS 4000 namespace, depending on which PSS 4000 OPC Server is used.

► PSS 4000 OPC Server when installed on a PC or PMI

The focus is on the PSS 4000 project and the general and device-related project data of all devices in a PSS 4000 project.

► PSS 4000-exclusive device with PSS 4000 OPC Server

The focus is on the device project data and the project data of the device.

If no device project has been downloaded to the device, the OPC UA Server uses a default configuration, which contains the device-specific elements of the namespace.

Access to variables in the PSS 4000 namespace

An OPC Client can access local variables (VAR) and resource global variables (VAR_GLOBAL) in the user program of the PSS 4000 project. Whether read or write access is possible does not depend on the variable's data type but on the resource on which the variable is located:

Variables	Access right
PI variables on FS resources	Read
All variables on FS resources, without PI variables	
PI variables to ST resources	
All variables to ST resources, without PI variables	Read and write

19.1 Basics

A namespace is defined via its URI (Unique Resource Identifier). The namespace URI describes a node set resource. Each namespace has a namespace index. The allocation of the namespace indices to the namespace URIs is stored in a table on the Server.

Namespace table of the OPC UA Server:

Namespace URI	Description
http://opcfoundation.org/UA/	Basic specifications defined by the OPC Foundation, such as basic data types for example. ▶ The OPC UA Server supports a reduced namespace for a minimal OPC UA Server (NS0). ▶ The OPC A Server supports a reduced namespace for BaseEvent communication in the OPC UA Server (NS0).
urn:<DNS name>:Pilz:PSS4000OPCServerUA	URI for unique identification of the Server application (NS1)
urn:Pilz:PSS4000OPCServerUA:PSS4000Namespace	URI for the PSS 4000 namespace (NS2)
http://www.pilz.com/UA	OPC UA variable "NamespaceArray" for PSS 4000-exclusive devices with system section OPC Server UA (NS3)

The application name of the OPC Server is **PSS 4000 OPC Server UA**.

The namespace supported by a PSS 4000 OPC Server UA is language-independent; i.e. the default language setting is English.

19.2 Structure of the namespace

In the following section, the PSS 4000 namespace is represented in a tree structure. The tree branches into various levels, with corresponding sublevels. The branch points are called nodes and are labelled. The nodes are used for data structuring (= elements). The elements are found exclusively on the lowest level. The full name that uniquely identifies an element in the namespace is called the fully qualified name (FQN). The FQN consists of the hierarchical node name (stated path) and the name of the element. The display format may vary, depending on the browser on the OPC Client. However, the structure remains the same.

Due to the complexity, the section below can only describe the individual areas consecutively. To ensure that you still retain an overview, the complete path through which the area's node is accessed is stated at the start of each area. When stating the path, a dot is used as the separator between the nodes.

For example, the path to the "State" area is entered as follows:
pss4000.net.myDevice1.Device.State.

So the path to the "Diag" element in the "State" area is:
pss4000.net.myDevice1.Device.State.Diag

Abbreviation of hierarchical path information

On PSS 4000-exclusive devices with system section OPC Server UA, the hierarchical path information for elements may be truncated. With elements, this can lead to ambiguous node IDs. In this case, PAS4000 registers an error. The affected nodes can be listed in the namespace of the PSS 4000-exclusive device via the OPC UA variable "AmbiguousNodes" (see [OPCServer](#)  787]).

19.3 Main area of the PSS 4000 namespace

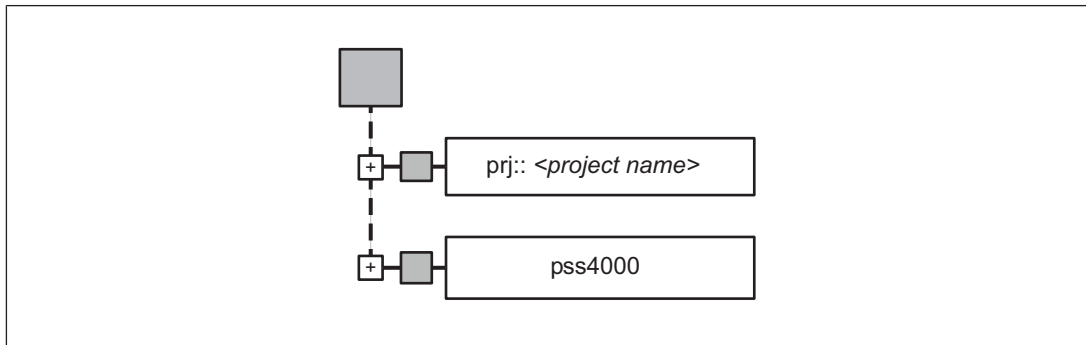


Fig.: Node names in the main area

Node name	Meaning
<i>prj::<project name></i>	Area for the general project data containing the project name that was configured in PAS4000. e.g.: "myProject"
pss4000	Area for the device-based project data

Definition of symbols



This symbol indicates that the node or element is only available for a PSS 4000 OPC Server that was installed on a PC or PMI.



This symbol indicates that the node or element is only available on a PSS 4000-exclusive device with the system section OPC Server UA.

<...>

Markers are shown in angle brackets in the diagrams and tables. They will be replaced with the respective current name. <device> becomes myDevice1, for example

19.4 General project data

Area structuring

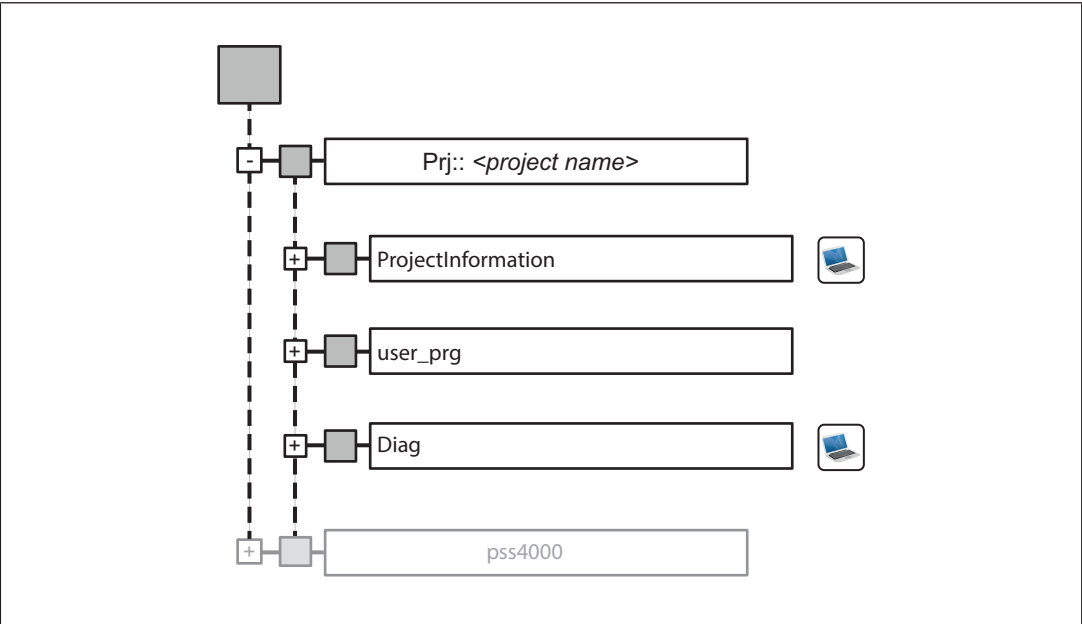


Fig.: Structuring of "General project data"

Node name (path information)		Meaning	
prj::<project name>.	ProjectInformation		Area containing information on PMC project <project name>
	prj::<project name>.ProjectInformation.		
	user_prg		Area containing information about the program (e.g. variables of POU's, variables of component blocks, resource global variables)
	prj::<project name>.user_prg.		
	Diag		Area containing diagnostic list messages and entries in the project's diagnostic log <project name>
prj::<project name>.Diag.			

19.4.1 Projectinformation

Availability: 

Stated path: prj::<project name>.ProjectInformation.

Node number	Node name	Meaning
1	ChecksumFSProject	Check sum "FS project" Data type: UInt32
	prj::<project name>.ProjectInformation.ChecksumFSProject	
2	ChecksumFSProjectAppParams	Check sum "Application parameters of the FS project" Data type: UInt32
	prj::<project name>.ProjectInformation.ChecksumFSProjectAppParams	
3	ChecksumFSProjectWithoutAppParams	Check sum "FS project without application parameters" Data type: UInt32
	prj::<project name>.ProjectInformation.ChecksumFSProjectWithoutAppParams	



INFORMATION

The check sums "FS project", "Application parameters of the FS project" and "FS project without application parameters" are displayed as a hexadecimal value in PAS4000 and in the PASvisu Server. To ensure that a check sum can be displayed correctly on a 3rd party visualisation system, please select: "Hexadecimal display" for the UInt32 value.

19.4.2 user_prg

Please note the following information regarding the availability of program-related information:

- ▶  PSS 4000-exclusive device with system section OPC Server UA
 - Access to resource global variables of the FS and ST resource, which were created by the user on this device.
 - Access to VAR type variables of the FS and ST resource, which were created by the user on this device. This includes the PI variables.
 - NO access to cross-device information
- ▶  PSS 4000 OPC Server when installed on a PC/PMI
 - Access to all resource global variables created by the user
 - Access to all VAR type variables created by the user, including PI variables

Stated path: prj::<project name>.user_prg.

Node num-ber	Node name	Meaning	
1	var_global	Area for the project's resource global variables The area includes all resource global variables created by the user.	
		Access rights to resource global variables in the OPC server:	
		PI variables on FS resources	Read
		All variables on FS resources, without PI variables	
		PI variables to ST resources	
		All variables to ST resources, without PI variables	Read and write
	prj::<project name>.user_prog.var_global.		
1.1	<var_global name>	Name of a resource global variable	
	prj::<project name>.user_prog.var_global.<var_global name>		

Node num-ber	Node name	Meaning	
2	app	Path of the project's variables The area includes the project's path with the user-created variables of type VAR, including the PI variables. Note: VAR_IN_OUT, VAR_EXTERNAL and VAR_TEMP type vari-ables are not displayed in the OPC Server.	
		Access rights to the variables in the OPC Server:	
		PI variables on FS resources	Read
		All variables on FS resources, without PI variables	
		PI variables to ST resources	
		All variables to ST resources, without PI variables	Read and write
		Path to a single-element variable: prj::<project name>.user_prog.app.<Path of variable>.<Name of variable>	
Path to a multi-element variable: ▶ Path of a STRUCT type variable: prj::<project name>.user_prog.app.<Path of variable>.<Name of variable>.<Name of struc-ture element> ▶ Path of an ARRAY type variable when access configuration is "Complete ARRAY": prj::<project name>.user_prog.app.<Path of array>.<Name of array> ▶ Availability:  Path of an ARRAY type variable when access configuration is "Index-based": prj::<project name>.user_prog.app.<Path of array>.<Name of array>.<Index of array ele-ment> Note regarding the "Complete ARRAY" and "Index-based" access types: ▶  PSS 4000-exclusive device with system section OPC Server UA: Only the "Complete ARRAY" access type is available. ▶  PSS 4000 OPC Server when installed on a PC/PMI: Both access types are available. The "Complete ARRAY" or "Index-based" access config-uration is set when the OPC UA Server is configured in PAS4000.			
prj::<project name>.user_prog.app			

19.4.3 Diag

Availability: 

Stated path: prj::<project name>.Diag.

Diagnostic list:

Node number	Node name	Meaning
1	List	Area containing the project's diagnostic list messages <i><project name></i>
	prj:: <i><project name></i> .Diag.List.	
1.1	Number	Number of messages for a severity level
	prj:: <i><project name></i> .Diag.List.Number.	
1.1.1	Error	Number of messages of "Error" severity Data type: UInt32
	prj:: <i><project name></i> .Diag.List.Number.Error	
1.1.2	Warning	Number of messages of "Warning" severity Data type: UInt32
	prj:: <i><project name></i> .Diag.List.Number.Warning	
1.1.3	Information	Number of messages of "Status information" severity Data type: UInt32
	prj:: <i><project name></i> .Diag.List.Number.Information	
1.2	Line	Lines in the diagnostic list
	prj:: <i><project name></i> .Diag.List.Line.	
1.2.1	<LineNumber>	Information on line number Value range: 1 ... 100
	prj:: <i><project name></i> .Diag.List.Line. <i><line number></i> .	
1.2.1.1	PriorityNo	Number of priority Data type: Byte
	prj:: <i><project name></i> .Diag.List.Line. <i><line number></i> .PriorityNo	
1.2.1.2	Priority	Name of priority Data type: String
	prj:: <i><project name></i> .Diag.List.Line. <i><line number></i> .Priority	
1.2.1.3	Description	Message text Data type: String
	prj:: <i><project name></i> .Diag.List.Line. <i><line number></i> .Description	
1.2.1.4	ScopeNo	Number of scope Data type: Byte
	prj:: <i><project name></i> .Diag.List.Line. <i><line number></i> .ScopeNo	

Node number	Node name	Meaning	
1.2.1.5	Scope	Name of scope Data type: String	
	prj::<project name>.Diag.List.Line.<line number>.Scope		
1.2.1.6	Timestamp	Time stamp Data type: DateTime	
	prj::<project name>.Diag.List.Line.<line number>.Timestamp		
1.2.1.7	Did	Diagnostic identifier Data type: String	
	prj::<project name>.Diag.List.Line.<line number>.Did		
1.2.1.8	StateNo	Status number of diagnostic message	
		Status	Meaning
		1	Message is active
		Data type: Byte	
	prj::<project name>.Diag.List.Line.<line number>.StateNo		
1.2.1.9	 State	Status of diagnostic message: Data type: String	
1.2.1.10	ParameterList	Parameter list Datentyp: String	
	prj::<project name>.Diag.List.Line.<line number>.ParameterList		
1.2.1.11	Location	Location information	
	prj::<project name>.Diag.List.Line.<line number>.Location.		
1.2.1.11.1	Address	Address Data type: String	
	prj::<project name>.Diag.List.Line.<line number>.Location.Address		
1.2.1.11.2	EquipmentId	Equip-ID Data type: String	
	prj::<project name>.Diag.List.Line.<line number>.Location.EquipmentId		
1.2.1.11.3	Device	Device Data type: String	
	prj::<project name>.Diag.List.Line.<line number>.Location.Device		
1.2.1.11.4	Description	Location description Data type: String	
	prj::<project name>.Diag.List.Line.<line number>.Location.Description		
1.2.1.12	Action	Remedial actions	
	prj::<project name>.Diag.List.Line.<line number>.Action.		

Node number	Node name	Meaning
1.2.1.12.1	<action number>	Number of remedy Value range: 1 ... 8
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.	
1.2.1.12.1.1	Description	Action text Data type: String
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Description	
1.2.1.12.1.2	ScopeNo	Number of scope Data type: Byte
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.ScopeNo	
1.2.1.12.1.3	Scope	Name of scope Data type: String
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Scope	
1.2.1.12.1.4	ResponsibilityNo	Number of responsibility Data type: Byte
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.ResponsibilityNo	
1.2.1.12.1.5	Responsibility	Name of responsibility Data type: String
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Responsibility	
1.2.1.12.1.6	Location	Location information
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Location.	
1.2.1.12.1.6.1	Address	Address Data type: String
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Location.Address	
1.2.1.12.1.6.2	EquipmentId	Equip-ID Data type: String
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Location.EquipmentId	
1.2.1.12.1.6.3	Device	Device Data type: String
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Location.Device	
1.2.1.12.1.6.4	Description	Location description Data type: String
	prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Location.Description	

Diagnostic log:

Node number	Node name	Meaning
2	Log	Area containing the project's diagnostic log entries <i><project name></i>
	prj:: <i><project name></i> .Diag.Log.	
2.1	Number	Number of log entries Data type: UInt32
	prj:: <i><project name></i> .Diag.Log.Number	
2.2	Line	Lines in the diagnostic log
	prj:: <i><project name></i> .Diag.Log.Line.	
2.2.1	<i><line number></i>	Information on line number Value range 1 ... "Number" (see 2.1; max. 4096)
	prj:: <i><project name></i> .Diag.Log.Line. <i><line number></i> .	
2.2.1.1	PriorityNo	Number of priority Data type: Byte
	prj:: <i><project name></i> .Diag.Log.Line. <i><line number></i> .PriorityNo	
2.2.1.2	Priority	Name of priority Data type: String
	prj:: <i><project name></i> .Diag.Log.Line. <i><line number></i> .Priority	
2.2.1.3	Description	Message text Data type: String
	prj:: <i><project name></i> .Diag.Log.Line. <i><line number></i> .Description	
2.2.1.4	ScopeNo	Number of scope Data type: Byte
	prj:: <i><project name></i> .Diag.Log.Line. <i><line number></i> .ScopeNo	
2.2.1.5	Scope	Name of scope Data type: String
	prj:: <i><project name></i> .Diag.Log.Line. <i><line number></i> .Scope	
2.2.1.6	Timestamp	Time stamp Data type: DateTime
	prj:: <i><project name></i> .Diag.Log.Line. <i><line number></i> .Timestamp	
2.2.1.7	Did	Diagnostic identifier Data type: String
	prj:: <i><project name></i> .Diag.Log.Line. <i><line number></i> .Did	

Node number	Node name	Meaning	
2.2.1.8	StateChangeNo	Status number for the change of state	
		Status	Meaning
		0	Message cleared
		1	Message arrived
		101	Log entry implemented (for operational entries, self test entries and debugging entries)
		Data type: Byte	
prj::<project name>.Diag.Log.Line.<line number>.StateChangeNo			
2.2.1.9	StateChange	Change of state	
		Display	Meaning
		cleared	Message cleared
		arrived	Message arrived
		logged	Log entry implemented (for operational entries, self test entries and debugging entries)
		Data type: String	
prj::<project name>.Diag.Log.Line.<line number>.StateChange			
2.2.1.10	ParameterList	Parameter list Datentyp: String	
	prj::<project name>.Diag.Log.Line.<line number>.ParameterList		
2.2.1.11	Location	Location information	
	prj::<project name>.Diag.Log.Line.<line number>.Location.		
2.2.1.11.1	Address	Address Data type: String	
	prj::<project name>.Diag.Log.Line.<line number>.Location.Address		
2.2.1.11.2	EquipmentId	Equip-ID Data type: String	
	prj::<project name>.Diag.Log.Line.<line number>.Location.EquipmentId		
2.2.1.11.3	Device	Device Data type: String	
	prj::<project name>.Diag.Log.Line.<line number>.Location.Device		
2.2.1.11.4	Description	Location description Data type: String	
	prj::<project name>.Diag.Log.Line.<line number>.Location.Description		

19.5 Device-based project data

Area structuring

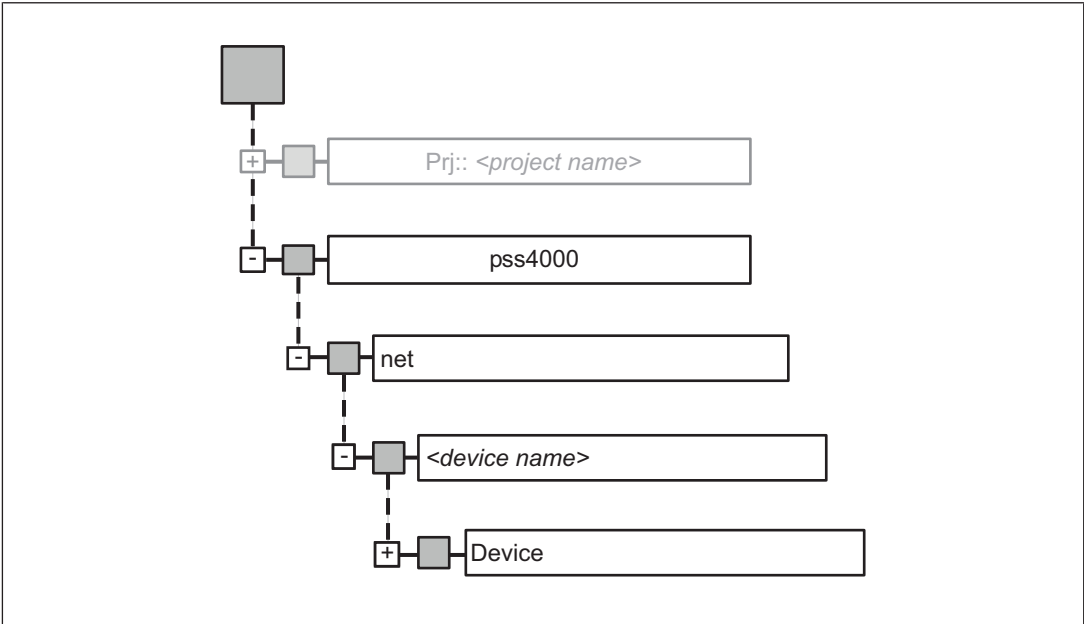


Fig.: Structuring of "Device-based project data"

Node name		Meaning
pss4000.	net	Network area
	pss4000.net.	
	<device name>	Area of device containing the device name configured during device naming PSS 4000-exclusive/operable device)
	pss4000.net.<device name>.	
	Device.	Area containing device-based information
	pss4000.net.<device name>.Device.	

PILZ | 755

Node number	Node name	Meaning	
4	Network	Area containing the device's network parameters <i><device name></i>	
	pss4000.net. <i><device name></i> .Device.Network.		
5	FS resource	Area containing information about the device's FS resource tasks <i><device name></i>	
	pss4000.net. <i><device name></i> .Device.FS-Resource.		
6	ST resource	Area containing information about the device's ST resource tasks <i><device name></i>	
	pss4000.net. <i><device name></i> .Device.ST-Resource.		
7	Diag		Area containing diagnostic information about the device <i><device name></i>
	pss4000.net. <i><device name></i> .Device.Diag.		
8	OPCServer		Area containing information about the device's OPC Server <i><device name></i>
	pss4000.net. <i><device name></i> .Device.OPCServer.		

19.5.1.1 State

Path information: pss4000.net.<device name>.Device.State.

Node number	Node name	Meaning
1	Diag	Status of the system sections on a device (see [1])
		Data type: Byte
	pss4000.net.<device name>.Device.State.Diag	

[1]

State number	State name	Meaning
0	PREPARING	Device diagnostic list and device diagnostic log are being prepared.
1	DELETE MODE	Start of "deliberate operator action" (function of reset button)
2	INTERNAL ERROR	A major FS error is present for at least one FS system section (see diagnostic list).
3	ERROR	A message of "Error" severity is present for at least one system section (see diagnostic table).
4	WARNING	A message of at least "Warning" severity is present for the device (see diagnostic list).
5	ERROR FREE	No message of "Error" or "Warning" severity is present for the device.

Node number	Node name	Meaning
2	STRun	Status of ST resource (see [2])
		Data type: Byte
	pss4000.net.<device name>.Device.State.STRun	

[2]

State number	State name	Meaning
0	STOP	None of the tasks on the ST resource are started or no tasks are configured.
1	PARTIAL RUN LICENSED	Operating state "ST resource in RUN state with error": - Task in TERMINATED state or - Task in STOP state At least one ST resource task is not running. The project is licensed.
2	RUN LICENSED	Operating state "ST resource in RUN state without error": The ST resource tasks are running without error. The project is licensed.
3	PARTIAL RUN UNLICENSED	Operating state "ST resource in RUN state with error": - Task in TERMINATED state or - Task in STOP state At least one ST resource task is not running. The project is unlicensed.
4	RUN UNLICENSED	Operating state "ST resource in RUN state without error": The ST resource tasks are running without error. The project is unlicensed.

Node number	Node name	Meaning
3	FSRun	Status of FS resource (see [3])
		Data type: Byte
	pss4000.net.<device name>.Device.State.FSRun	

[3]

State number	State name	Meaning
0	STOP	None of the tasks on the FS resource are started or no tasks are configured.
1	PARTIAL RUN LICENSED	Operating state "FS resource in RUN state with error": - Task in TERMINATED state or - Task in STOP state At least one FS resource task is not running. The project is licensed.
2	RUN LICENSED	Operating state "FS resource in RUN state without error": The FS resource tasks are running without error. The project is licensed.
3	PARTIAL RUN UNLICENSED	Operating state "FS resource in RUN state with error": - Task in TERMINATED state or - Task in STOP state At least one FS resource task is not running. The project is unlicensed.
4	RUN UNLICENSED	Operating state "FS resource in RUN state without error": The FS resource tasks are running without error. The project is unlicensed.

Node number	Node name	Meaning
4	STForce	Status of forcing and online changes on the ST resource (see [4])
		Data type: Byte
	pss4000.net.<device name>.Device.State.SOPROFIsafe	

[4]

State number	State name	Meaning
0	ONLINE CHANGE IN-ACTIVE	No online change is active on the ST resource. Note: Forcing on the ST resource is not supported.
1	ONLINE CHANGE ACTIVE	At least one online change is active on the ST resource.

Node number	Node name	Meaning
5	FSForce	Status of forcing and online changes on the FS resource (see [5])
		Data type: Byte
	pss4000.net.<device name>.Device.State.FSForce	

[5]

State number	State name	Meaning
0	FORCING OR ONLINE CHANGE INACTIVE	On the FS resource, forcing is inactive and no online change is active.
1	FORCING OR ONLINE CHANGE ACTIVE	On the FS resource, forcing is active and/or at least one online change is active.

Node number	Node name	Meaning
6	STSafetyNETp	State of the system section ST SafetyNET p (see [6])
		Data type: Byte
	pss4000.net.<device name>.Device.State.STSafetyNETp	

[6]

State number	State name	Meaning
0	CONFIGURATION INVALID	Operating state "ST SafetyNET p in STOP state with error: major FS+ST error"
1	STOP	ST SafetyNET p has not started.
2	PARTIAL RUN	Operating state: "ST SafetyNET p in RUN state with minor error"
3	RUN	Operating state "ST SafetyNET p in RUN state without error"
4	INTERNAL ERROR	Operating state "ST SafetyNET p in STOP state with error: major FS+ST error"

Node number	Node name	Meaning
7	FSSafetyNETp	State of the system section FS SafetyNET p (see [7])
		Data type: Byte
	pss4000.net.<device name>.Device.State.FSSafetyNETp	

[7]

State number	State name	
0	CONFIGURATION INVALID	Operating state "FS SafetyNET p in STOP state with error: major FS error"
1	STOP	FS SafetyNET p has not started.
2	PARTIAL RUN	Operating state "FS SafetyNET p in RUN state with minor error"
3	RUN	Operating state "FS SafetyNET p in RUN state without error"
4	INTERNAL ERROR	Operating state "FS SafetyNET p in STOP state with error: major FS+ST error"

Node number	Node name	Meaning
8	Module bus	State of the system sections FS and ST module bus (see [8])
		Data type: Byte
	pss4000.net.<device name>.Device.State.Modulebus	

[8]

State number	State name	Meaning
0	ERROR	Operating state "Safe state of all FS outputs on the device" or at least one module cannot be accessed (e.g. a module has been removed during operation, set/actual hardware registry does not match).
1	INACTIVE	No modules are configured and no modules are available.
2	RUN	FS and ST module bus are operating without error.
3	INTERNAL ERROR	Operating state "FS module bus in STOP state with error: major FS error"

Node number	Node name	Meaning
9	Media	Status of data medium and device identification (see [9])
		Data type: Byte
	pss4000.net.<device name>.Device.State.Media	

[9]

State number	State name	
0	BLINK TEST ACTIVATED	Device identification activated by user: ▶ Identification as PSS 4000 device in SafetyNET p (PAS4000) or ▶ Identification as PROFINET IO Device in PROFINET (engineering tool for PROFINET)
1	CARD NOT READABLE	▶ SD card is missing. or ▶ SD card was not detected. or ▶ SD card is defective.
2	WRONG PRODUCT TYPE OR NO DATA ON CARD	▶ Product type on the SD card does not match the product type of the device/head module. or ▶ No device project and/or no naming data on the SD card.
3	DATA ON CARD AND ON DEVICE DO NOT MATCH	Naming data and device project on the device and SD card do not match.
4	DATA ON CARD AND ON DEVICE DO MATCH	Naming data and device project on the device and SD card match.
5	BOOT COMMAND	Reserved
6	BIND DEVICE PROJECT ERROR	▶ "Bind device projects to devices" function: – The device does not have the device key that matches the device project. or – The device project does not have the project key that matches the device. and/or ▶ "Bind device projects to SD cards" function: The device project is bound to an SD card, but this SD card is not inserted in the device.

Node number	Node name	Meaning
10	SFProfibus	"SF" status LED (PROFIBUS-DP) The evaluation can only be performed in combination with the LEDs of the "BF" status LED (see operating manual for the device/head module). (see [10])
		Data type: Byte
	pss4000.net.<device name>.Device.State.SFProfibus	

[10]

State number	State name	Meaning
0	FAULT	LED "lights red"
1	DATA EXCHANGE	LED "off"

Node number	Node name	Meaning
11	BFProfibus	"BF" status LED (PROFIBUS-DP) The evaluation can only be performed in combination with the LEDs of the "SF" status LED (see operating manual for the device/head module). (see [11])
		Data type: Byte
	pss4000.net.<device name>.Device.State.BFProfibus	

[11]

State number	State name	Meaning
0	NO CONNECTION	LED "lights red"
1	PARAMETERIZATION FAULT	LED "flashes red"
2	DATA EXCHANGE	LED "lights up green"
3	NOT CONFIGURED	LED "off"

Node number	Node name	Meaning
12	SOPROFIsafe	"SO" status LED (PROFIsafe) Status of PROFIsafe in accordance with PROFIsafe specification (see [12])
		Data type: Byte
	pss4000.net.<device name>.Device.State.SOPROFIsafe	

[12]

State number	State name	Meaning
0	INACTIVE	LED "off" No F-module is configured or an F-module is configured, but it is neither in "Operator Acknowledge Requested" nor "Safe Operation" state
1	OPERATOR ACKNOWLEDGE REQUESTED	LED "flashes green" "Operator Acknowledgement" is requested by at least one F-module (OA_C requested)
3	SAFE OPERATION	LED "lights up green" PROFIsafe communication is ok

Node number	Node name	Meaning
13	MSEtherNetIP	Status LED "MS" (EtherNet/IP) Module status in accordance with EtherNet/IP specification (see [13])
		Data type: Byte
	pss4000.net.<device name>.Device.State.MSEtherNetIP	

[13]

State number	State name	Meaning
0	MAJOR FAULT	LED "lights up red" A message of "Error" severity is present for at least one system section or a major FS error is present for at least one FS system section (see diagnostic list).
1	MINOR FAULT	LED "flashes red quickly" At least one message of "Warning" severity is present for the device, no message of "Error" severity is present and no invalid data is being downloaded to the scanner.
2	OPERATIONAL	LED "lights up green" No message of "Error" or "Warning" severity is present for the device
3	INACTIVE	LED "off" No supply voltage or device inactive or device not configured

Node number	Node name	Meaning
14	NSEtherNetIP	Status LED "NS" (EtherNet/IP) Network status in accordance with EtherNet/IP specification (see [14])
		Data type: Byte
	pss4000.net.<device name>.Device.State.NSEtherNetIP	

[14]

State number	State name	Meaning
0	CONNECTION TIMEOUT	LED "flashes red quickly" Connection in timeout
1	NO CONNECTIONC	LED "flashes green quickly" No network connection
2	CONNECTED	LED "lights up green" Network connection is available and EtherNet/IP communication is ok
3	INACTIVE	LED "off" No data traffic or EtherNet/IP not configured or no IP address configured

Node number	Node name	Meaning
15	BFProfinet	Status LED "BF" (PROFINET IO Bus Failure) Bus status in accordance with PROFINET IO specification (see [15])
		Data type: Byte
	pss4000.net.<device name>.Device.State.BFProfinet	

[15]

State number	State name	Meaning
0	NO CONNECTION	LED "lights up red" No connection to the PROFINET IO Controller
1	PARAMETERIZATION FAULT	LED "flashes red" Connection to PROFINET IO Controller available, no data exchange due to faulty parameter settings
2	OPERATIONAL	LED "lights up green" The PROFINET IO Device is in "Operational" state, data exchange possible
3	DEACTIVATED	LED "off" The PROFINET IO Device is deactivated; i.e. no virtual PROFINET IO modules configured.

Node number	Node name	Meaning
16	ModulebusPSSu2	Status of the system sections FS and ST module bus PSSu 2 (see [16])
		Data type: USINT
	pss4000.net.<device name>.Device.State.ModulebusPSSu2	

[16]

State number	State name	Meaning
0	INACTIVE	The module bus is just being started or the system section "Module bus PSS u2" does not exist on the device.
1	ERROR FREE	FS and ST module bus are operating without error.
2	REMEDIALABLE ERROR	There is at least one message of "Error" severity present for the FS or ST module bus, which can be recovered by the user (see diagnostic list).
3	INTERNAL ERROR	The FS or ST module bus has an internal error.

19.5.1.2 DeviceProject

Path information: pss4000.net.<device name>.Device.DeviceProject.

Node number	Node name	Meaning	
1	Name	Name of the PSS 4000 project to which the device <i><device name></i> belongs Data type: String	
	pss4000.net.<device name>.Device.DeviceProject.Name		
2	ChecksumFS	Check sum "FS device project" for the device <i><device name></i> Data type: UInt32	
	pss4000.net.<device name>.Device.DeviceProject.ChecksumFS		
3	ChecksumFSProjectAppParams	Check sum "Application parameters of the FS device project" for the device <i><device name></i> Data type: UInt32	
	pss4000.net.<device name>.Device.DeviceProject.ChecksumFSProjectAppParams		
4	ChecksumFSProjectWithoutAppParams	Check sum "FS device project without application parameters" for the device <i><device name></i> Data type: UInt32	
	pss4000.net.<device name>.Device.DeviceProject.ChecksumFSProjectWithoutAppParams		
5	BuildTime	Time at which the PSS 4000 project was built Data type: DateTime	
	pss4000.net.<device name>.Device.DeviceProject.BuildTime		
6	Version	Version of the PSS 4000 project to which the device <i><device name></i> belongs Data type: String	
	pss4000.net.<device name>.Device.DeviceProject.Version		
7	VersionPAS4000	PAS4000 version with which the PSS 4000 project was built. Data type: String	
	pss4000.net.<device name>.Device.DeviceProject.VersionPAS4000		
8	LicenceState	Status information on the licensing state of the project	
		Status	Meaning
		0	Unlicensed or no project active
		1	Licensed
		Data type: Byte	
	pss4000.net.<device name>.Device.DeviceProject.LicenceState		

19.5.1.3

Product

Path information: pss4000.net.<device name>.Device.Product.

Node number	Node name	Meaning
1	SerialNo	Serial number of the device <device name> PSSu systems/PSS u2 systems: Serial number of the head module Data type: String
	pss4000.net.<device name>.Device.Product.SerialNo	
2	OrderNo	Order number of the device <device name> PSSu systems/PSS u2 systems: Order number of the head module Data type: String
	pss4000.net.<device name>.Device.Product.OrderNo	
3	ProductType	Product type of the device <device name> PSSu systems/PSS u2 systems: Description of the head module Data type: String
	pss4000.net.<device name>.Device.Product.ProductType	
4	FirmwareVersion	Firmware version of the device <device name> PSSu systems: Firmware version of the head module Data type: String
	pss4000.net.<device name>.Device.Product.FirmwareVersion	
5	FirmwareRevision	Revision number of the firmware of the device <device name> PSSu systems/PSS u2 systems: Revision number of the firmware of the head module Data type: String
	pss4000.net.<device name>.Device.Product.FirmwareRevision	
6	FirmwareUpdate	Time at which the firmware for the device <device name> was last updated PSSu systems/PSS u2 systems: Time at which the firmware for the head module was last updated Data type: DateTime
	pss4000.net.<device name>.Device.Product.FirmwareUpdate	

Node number	Node name	Meaning
7	HardwareVersion	Hardware version of the device <i><device name></i> Display format: <i><MajorHardwareVersion>.<MinorHardwareVersion></i> Example: 3.1 PSSu systems: Hardware version of the head module Data type: String
	pss4000.net. <i><device name></i> .Device.Product.HardwareVersion	
8	MajorHardwareVersion	Major hardware version of the device <i><device name></i> PSSu systems/PSS u2 systems: Major hardware version of the head module Data type: Byte
	pss4000.net. <i><device name></i> .Device.Product.MajorHardwareVersion	
9	MinorHardwareVersion	Minor hardware version of the device <i><device name></i> PSSu systems/PSS u2 systems: Minor hardware version of the head module Data type: Byte
	pss4000.net. <i><device name></i> .Device.Product.MinorHardwareVersion	
10	PilzSecSDcardSerial-Number	Serial number of the security SD card (e.g. PAS-key SD Card) of the device <i><device name></i> Data type: String
	pss4000.net. <i><device name></i> .Device.Product.PilzSecSDcardSerialNumber	
11	DeviceKey	Device key If a device key has been written to the device for the function "Bind device projects to devices", part of the device key is displayed. Further information on this subject is available in the online help. Data type: String
	pss4000.net. <i><device name></i> .Device.Product.DeviceKey	

Node number	Node name	Meaning	
12	DeviceKeyRestricts-Projects	Information on whether projects that are to be executed on the device require a matching project key. Further information on the function "Bind device projects to devices" is available in the online help.	
		Display	Meaning
		FALSE	Project does not require a project key
		TRUE	Project requires suitable project key
		Data type: Boolean	
pss4000.net.<device name>.Device.Product.DeviceKeyRestrictsProjects			

19.5.1.4 Network

Path information: pss4000.net.<device name>.Device.Network.

Node number	Node name	Meaning	
1	IPAddressV4	IP address of device <i><device name></i> Data type: String	
	pss4000.net. <i><device name></i> .Device.Network.IPAddressV4		
2	SubnetMaskV4	Subnet mask Data type: String	
	pss4000.net. <i><device name></i> .Device.Network.SubnetMaskV4		
3	GatewayV4	IP address of Standard Gateway Data type: String	
	pss4000.net. <i><device name></i> .Device.Network.GatewayV4		
4	DHCP	DHCP setting for the device <i><device name></i>	
		Display	Meaning
		FALSE	DHCP not activated
		TRUE	DHCP activated
		Data type: Boolean	
	pss4000.net. <i><device name></i> .Device.Network.DHCP		
5	DNS1V4	Address of preferred DNS Server Data type: String	
	pss4000.net. <i><device name></i> .Device.Network.DNS1V4		
6	DNS2V4	Address of alternative DNS Server Data type: String	
	pss4000.net. <i><device name></i> .Device.Network.DNS2V4		
7	MACAddress	MAC address of device <i><device name></i> Data type: String	
	pss4000.net. <i><device name></i> .Device.Network.MACAddress		
8	SNTP	SNTP setting for the device <i><device name></i>	
		Display	Meaning
		FALSE	Device is not configured as SNTP server of the project
		TRUE	Device works in the project as SNTP Server
		Data type: Boolean	
	pss4000.net. <i><device name></i> .Device.Network.SNTP		

Node number	Node name	Meaning
9	SNTPV4	IP address of the SNTP Server from which the device obtains its time Data type: String
	pss4000.net.<device name>.Device.Network.SNTPV4	
10	RealTimeClock	Current time on the device Data type: DateTime
	pss4000.net.<device name>.Device.Network.RealTimeClock	

19.5.1.5

FS resource

Path information: pss4000.net.<device name>.Device.FS-Resource.

Node number	Node name	Meaning	
1	UserProgramMemoryUsed	Memory used for the user program on the device <i><device name></i> (including active on-line changes)	
	Data type: UInt32		
pss4000.net. <i><device name></i> .Device.FS-Resource. UserProgramMemoryUsed			
2	UserProgramMemorySize	Size of the memory for the user program on the device <i><device name></i>	
	Data type: UInt32		
pss4000.net. <i><device name></i> .Device.FS-Resource. UserProgramMemorySize			
3	<i><task name></i>	Name of the task in the FS resource of the device <i><device name></i>	
	pss4000.net. <i><device name></i> .Device.FS-Resource. <i><task name></i> .		
3.1	State	Status of the task	
		Status	Meaning
		1	Task in STOP state
		2	Task in RUN state
		3	Task in TERMINATED state
		Data type: UInt32	
pss4000.net. <i><device name></i> . Device.FS-Resource. <i><task name></i> .State			
3.2	ProcessingTime	Task execution time t_{ProcTask} The execution time is only available for periodic tasks (see Reaction times  570])	
	pss4000.net. <i><device name></i> . Device.FS-Resource. <i><task name></i> . ProcessingTime.		

Node number	Node name	Meaning
3.2.1	Minimum	Minimum execution time The displayed time is the shortest execution time that has been detected since the task was started. Data type: UInt32
	pss4000.net.<device name>. Device.FS-Resource.<task name>. ProcessingTime.Minimum	
3.2.2	Maximum	Maximum execution time The displayed time is the longest execution time that has been detected since the task was started. Data type: UInt32
	pss4000.net.<device name>. Device.FS-Resource.<task name>. ProcessingTime.Maximum	
3.2.3	Current	Current task execution time Data type: UInt32
	pss4000.net.<device name>. Device.FS-Resource.<task name>. ProcessingTime.Current	

19.5.1.6

ST resource

Path information: pss4000.net.<device name>.Device.ST-Resource.

Node number	Node name	Meaning								
1	UserProgramMemoryUsed	Memory used for the user program on the device <i><device name></i> (including active on-line changes) Data type: UInt32								
	pss4000.net. <i><device name></i> .Device.ST-Resource. UserProgramMemoryUsed									
2	UserProgramMemorySize	Size of the memory for the user program on the device <i><device name></i> Data type: UInt32								
	pss4000.net. <i><device name></i> .Device.ST-Resource. UserProgramMemorySize									
3	<i><task name></i>	Name of the task in the ST resource of the device <i><device name></i>								
	pss4000.net. <i><device name></i> .Device.ST-Resource. <i><task name></i> .									
3.1	State	Status of the task								
		<table><tr><th>Status</th><th>Meaning</th></tr><tr><td>1</td><td>Task in STOP state</td></tr><tr><td>2</td><td>Task in RUN state</td></tr><tr><td>3</td><td>Task in TERMINATED state</td></tr></table>	Status	Meaning	1	Task in STOP state	2	Task in RUN state	3	Task in TERMINATED state
		Status	Meaning							
		1	Task in STOP state							
		2	Task in RUN state							
		3	Task in TERMINATED state							
		Data type: UInt32								
pss4000.net. <i><device name></i> .Device.ST-Resource. <i><task name></i> .State										
3.2	ProcessingTime	Task execution time t_{ProcTask} The execution time is only available for periodic tasks (see Reaction times  570])								
	pss4000.net. <i><device name></i> . Device.ST-Resource. <i><task name></i> . ProcessingTime.									

Node number	Node name	Meaning
3.2.1	Minimum	Minimum execution time The displayed time is the shortest execution time that has been detected since the start of the task. Data type: UInt32
	pss4000.net.<device name>. Device.ST-Resource.<task name>. ProcessingTime.Minimum	
3.2.2	Maximum	Maximum execution time The displayed time is the longest execution time that has been detected since the start of the task. Data type: UInt32
	pss4000.net.<device name>. Device.ST-Resource.<task name>. ProcessingTime.Maximum	
3.2.3	Current	Current task execution time Data type: UInt32
	pss4000.net.<device name>. Device.ST-Resource.<task name>. ProcessingTime.Current	

19.5.1.7

Diag

Availability: 

Path information: pss4000.net.<device name>.Device.Diag.

Device diagnostic list

Note:

- ▶ An OPC UA Client can use its browser to access any element of the device diagnostic list.
- ▶ The elements can always be accessed, even if there is no device project on the PSS 4000-exclusive device.
- ▶ It is possible to access a maximum of 128 elements.

Node number	Node name	Meaning
1	List	Area containing the device's diagnostic list messages <i><device name></i>
	pss4000.net. <i><device name></i> .Device.Diag.List.	
1.1	Number	Number of messages for a severity level
	pss4000.net. <i><device name></i> .Device.Diag.List.Number.	
1.1.1	Error	Number of messages of "Error" severity Data type: UInt32
	pss4000.net. <i><device name></i> .Device.Diag.List.Number.Error	
1.1.2	Warning	Number of messages of "Warning" severity Data type: UInt32
	pss4000.net. <i><device name></i> .Device.Diag.List.Number.Warning	
1.1.3	Information	Number of messages of "Status information" severity Data type: UInt32
	pss4000.net. <i><device name></i> .Device.Diag.List.Number.Information	
1.2	Line	Lines in the diagnostic list
	pss4000.net. <i><device name></i> .Device.Diag.List.Line.	
1.2.1	<LineNumber>	Information on line number Value range: 1 ... 100
	pss4000.net. <i><device name></i> .Device.Diag.List.Line. <i><line number></i> .	
1.2.1.1	PriorityNo	Number of priority Data type: Byte
	pss4000.net. <i><device name></i> .Device.Diag.List.Line. <i><line number></i> .PriorityNo	

Node number	Node name	Meaning	
1.2.1.2	Priority	Name of priority Data type: String	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Priority		
1.2.1.3	Description	Message text Data type: String	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Description		
1.2.1.4	ScopeNo	Number of scope Data type: Byte	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.ScopeNo		
1.2.1.5	Scope	Name of scope Data type: String	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Scope		
1.2.1.6	Timestamp	Time stamp Data type: DateTime	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Timestamp		
1.2.1.7	Did	Diagnostic identifier Data type: String	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Did		
1.2.1.8	StateNo	Status number of diagnostic message	
		Status	Meaning
		1	Message is active
		Data type: Byte	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.StateNo		
1.2.1.9	 State	Status of diagnostic message: Data type: String	
1.2.1.10	ParameterList	Parameter list Datentyp: String	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.ParameterList		
1.2.1.11	Location	Location information	
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Location.		

Node number	Node name	Meaning
1.2.1.11.1	Address	Address Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Location.Address	
1.2.1.11.2	EquipmentId	Equip-ID Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Location.EquipmentId	
1.2.1.11.3	Device	Device Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Location.Device	
1.2.1.11.4	Description	Location description Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Location.Description	
1.2.1.12	Action	Remedial actions
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.	
1.2.1.12.1	<action number>	Number of remedy Value range: 1 ... 8
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.	
1.2.1.12.1.1	Description	Action text Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.Description	
1.2.1.12.1.2	ScopeNo	Number of scope Data type: Byte
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.ScopeNo	
1.2.1.12.1.3	Scope	Name of scope Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.Scope	
1.2.1.12.1.4	ResponsibilityNo	Number of responsibility Data type: Byte
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.ResponsibilityNo	
1.2.1.12.1.5	Responsibility	Name of responsibility Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.Responsibility	

Node number	Node name	Meaning
1.2.1.12.1.6	Location	Location information
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.Location.	
1.2.1.12.1.6.1	Address	Address Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.Location.Address	
1.2.1.12.1.6.2	EquipmentId	Equip-ID Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.Location.EquipmentId	
1.2.1.12.1.6.3	Device	Device Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.Location.Device	
1.2.1.12.1.6.4	Description	Location description Data type: String
	pss4000.net.<device name>.Device.Diag.List.Line.<line number>.Action.<action number>.Location.Description	

Device diagnostic log

Note:

- ▶ An OPC UA Client can use its browser to access any element of the device diagnostic log.
- ▶ The elements can always be accessed, even if there is no device project on the PSS 4000-exclusive device.
- ▶ It is possible to access a maximum of 256 elements.

Node number	Node name	Meaning
2	Log	Area containing the device's diagnostic log entries <i><device name></i>
	pss4000.net.<device name>.Device.Diag.Log.	
2.1	Number	Number of log entries Data type: UInt32
	pss4000.net.<device name>.Device.Diag.Log.Number	
2.2	Line	Lines in the diagnostic log
	pss4000.net.<device name>.Device.Diag.Log.Line.	
2.2.1	<line number>	Information on line number Value range 1 ... "Number" (see 2.1; max. 4096)
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.	
2.2.1.1	PriorityNo	Number of priority Data type: Byte
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.PriorityNo	
2.2.1.2	Priority	Name of priority Data type: String
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.Priority	
2.2.1.3	Description	Message text Data type: String
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.Description	
2.2.1.4	ScopeNo	Number of scope Data type: Byte
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.ScopeNo	
2.2.1.5	Scope	Name of scope Data type: String
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.Scope	

Node number	Node name	Meaning	
2.2.1.6	Timestamp	Time stamp Data type: DateTime	
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.Timestamp		
2.2.1.7	Did	Diagnostic identifier Data type: String	
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.Did		
2.2.1.8	StateChangeNo	Status number for the change of state	
		Status	Meaning
		0	Message cleared
		1	Message arrived
		101	Log entry implemented (for operational entries, self test entries and debugging entries)
	Data type: Byte		
pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.StateChangeNo			
2.2.1.9	StateChange	Change of state	
		Display	Meaning
		cleared	Message cleared
		arrived	Message arrived
		logged	Log entry implemented (for operational entries, self test entries and debugging entries)
	Data type: String		
pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.StateChange			
2.2.1.10	ParameterList	Parameter list Datentyp: String	
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.ParameterList		
2.2.1.11	Location	Location information	
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.Location.		
2.2.1.11.1	Address	Address Data type: String	
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>.Location.Address		

Node number	Node name	Meaning
2.2.1.11.2	EquipmentId	Equip-ID Data type: String
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>. Location.EquipmentId	
2.2.1.11.3	Device	Device Data type: String
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>. Location.Device	
2.2.1.11.4	Description	Location description Data type: String
	pss4000.net.<device name>.Device.Diag.Log.Line.<line number>. Location.Description	

19.5.1.8 OPCServerAvailability: 

Path information: pss4000.net.<device name>.Device.OPCServer.

Node number	Node name	Meaning
1	UsedNamespaceBasicMemory	Memory in the device's namespace basic memory <device name> that is used by namespace elements Data type: UInt32
	pss4000.net.<device name>.Device.OPCServer.UsedNamespaceBasicMemory	
2	SizeOfNamespaceBasicMemory	Maximum size of the device's namespace basic memory <device name> Data type: UInt32
	pss4000.net.<device name>.Device.OPCServer.SizeOfNamespaceBasicMemory	
3	NumberOfOmmitedNodes	Number of nodes that are omitted because there is insufficient memory in the device's namespace basic memory <device name> Data type: UInt32
	pss4000.net.<device name>.Device.OPCServer.NumberOfOmmitedNodes	
4	AmbiguousNodes	FQN of the first 30 ambiguous nodes Data type: String Array
	pss4000.net.<device name>.Device.OPCServer.AmbiguousNodes	

19.6 Data types in the PSS 4000 namespace

The data types documented below can appear in the PSS 4000 namespace. Please note that the data types of the PSS 4000 namespace must be assigned to the data types available on the relevant client/server. Appropriate assignment lists are available for Pilz clients and servers. Please refer to the documentation for the relevant Pilz clients and servers.

Data types	Meaning
Boolean	Bit value/Boolean Data width: 1 bit Value range: TRUE/FALSE
Byte	Positive integer Data width: 8 bits (1 byte) Value range: 0 ... 255
Double	Floating point number with double precision (in accordance with IEEE 754) Data width: 64 bits (8 bytes)
Float	Floating point number with single precision (in accordance with IEEE 754) Data width: 32 bits (4 bytes)
Int16	Integer Data width: 16 bits (1 word) Value range: -32768 ... 32767
Int32	Integer Data width: 32 bits (2 words) Value range: - 2147483648 ... 2147483647
Int64	Integer Data width: 64 bits (4 words) Value range: - 9223372036854775808 ... 9223372036854775807
LocalizedText	Text with a language identifier Coding see "String"
SByte	Integer Data width: 8 bits (1 byte) Value range: -128 ... 127
String	Character string in Unicode format
UInt16	Positive integer Data width: 16 bits (1 word) Value range: 0 ... 65535
UInt32	Positive integer Data width: 32 bits (2 words) Value range: 0 ... 4294967295
UInt64	Positive integer Data width: 64 bits (4 words) Value range: 0 ... 18446744073709551615
DateTime	Positive integer Data width: 64 Bit (4 words) The value displayed for DateTime is interpreted as the number of 100 nanosecond intervals that have elapsed since 1601-01-01 00:00:00.

20 Appendix

20.1 Factory default settings for port numbers for various services

Port numbers have standard default settings for various services.

Service		Default setting	Meaning
SafetyNET p		40000	SafetyNET p protocol
DNS		53	Domain Name Service Protocol
OPC via DCOM	Connection Setup (Server)	135	Distributed Component Object Model Protocol
	Communication (Server)	Dynamic	
DHCP		67, 68	Dynamic Host Configuration Protocol
Modbus/TCP		502	Modbus over TCP
SNTP		123	Simple Network Time Protocol

20.2 Factory default settings for network parameters

The Auto IP mechanism is activated on PSS 4000-exclusive devices in delivery status or after an original reset. The IP parameters are set as follows:

Parameter	Default setting	Meaning
IP address	Auto-IP mechanism is active: 169.254.1.0 to 169.254.254.255	The Auto-IP mechanism assigns the IP address randomly.
Subnet mask	255.255.0.0	The Auto IP mechanism assigns the stated Subnet mask.
Address of Standard Gateway	deactivated (0.0.0.0)	Configurable as an option when SafetyNET p is divided into subnetworks through Gateways and, from the communication partner's perspective, the product is behind a Gateway
Address of preferred DNS server	deactivated (0.0.0.0)	Configurable as an option when the Domain Name Service is used
Address of alternative DNS server		

There are other network parameters besides the IP parameters. These network parameters can only be configured via the extended network configuration (see [Extended network configuration](#)  137). If there is no extended network configuration on a PSS 4000-exclusive device, the following default settings are used for these parameters:

Parameter	Default setting	Meaning
Ports' Ethernet mode	Switched mode	see Ports' Ethernet mode  138]
Maximum transmission unit (MTU)	1500	see ETH parameter " <i>mtu</i> " under ETH_INTERFACE  144]
802.1Q tag	Not configured	see ETH parameter " <i>ieee_802_1q</i> " under ETH_INTERFACE  144]
DSCP (DSCP bits in the TOS field)	0	see DSCP_RULE  148]

20.3 Network data

Depending on the product you are using, please note the following network data:

Protocol	Direction [*]	Transport protocol	Port no.	Can be deactivated	Time-critical	Description
SNTP	In	IPv4	123	Yes	No	Time synchronisation with SNTP (SNTP Server)
SNTP	Out	IPv4	123	Yes	No	Time synchronisation with SNTP (SNTP Client)
SNp	In/out	IPv4	40000	No	Yes	Communication via SafetyNET p for reading and retrieving ▸ Process data ▸ Status information ▸ Event messages
SNp MSC	In	IPv4	40000	No	No	SafetyNET p communication with IP-Multicast for ▸ Scan network ▸ Device naming ▸ Device identification
Modbus/TCP	Out	IPv4	1 ... 65535	Yes	Yes	Communication via Modbus/TCP with Client connection
Modbus/TCP	In	IPv4	1 ... 65535	Yes	Yes	Communication via Modbus/TCP with Server connection
Raw TCP	Out	IPv4	1 ... 65535	Yes	Yes	Communication via Raw TCP with Client connection
Raw TCP	In	IPv4	1 ... 65535	Yes	Yes	Communication via Raw TCP
Raw UDP send/receive	In/out	IPv4	1 ... 65535	Yes	Yes	Send/receive communication via Raw UDP
UDP using blocks	In/out	IPv4	1 ... 65535	Yes	Yes	UDP communication using blocks (PSS 4000 system control blocks)

[*]

in: The communication partner starts communication with the product.

out: The product starts communication with the communication partner.

You'll find network data for the OPC server under [Network data \(OPC Server UA\)](#) [ 702] and [Network data \(OPC Server \(classic\)\)](#) [ 669].

Auto IP mechanism

Technology for configuration-free and/or automatic assignment of a free IP address from the address range 169.254.0.0/16

Autocrossing

Automatic crossing of the send and receive lines on twisted pair interfaces; devices that support this function can be connected via a 1:1 wired cable (patch cable) instead of a crossed cable (crossover cable).

Autosensing

Automatic detection of data rate and duplex mode

Basic diagnostic item (BDI)

Diagnostics: Part of the user program in which a process state that is to be monitored for diagnostic purposes is defined. The basic diagnostic item also contains information and texts for one or more diagnostic messages and diagnostic log entries.

BDI

Abbreviation for basic diagnostic item

Bit module

Electronic/compact module whose data requires one bit each in the process image. Example: Digital input/output modules

Block

Umbrella term for Multi blocks and POU's

Build

The code for the individual devices (device projects) is built from the project.

Byte module

Electronic module whose data requires different data widths in the process image (bit, byte and/or word). Example: Counter modules

Certification

Procedure by which a product's compliance with the applicable regulations and standards is tested by a notified body; proven compliance is confirmed through the issue of a certificate.

Certification body

Notified body in accordance with Article 14 of the Directive 2006/42/EC; otherwise: Accredited institution, often a government body, which examines and confirms a product's compliance with applicable regulations and standards.

Check sum

Sum, which is formed via an object, e.g. a project or program organisation unit (POU). This can be used to detect whether the object has been modified.

Consistency range

Amount of consistent data in a data source (e.g. consistency range "FS-I data of an FS module bus")

Control variables

The value of variables is set once to a specified value. It can then be changed by the user program.

Crossover cable

Cross wired cables for wiring Ethernet components and end devices such as PLC or network components (hubs, switches)

Data consistency

Keeping together data that is recorded at a certain time and is considered current and related during a defined period.

Debugging

Troubleshooting and fault recovery during programming or commissioning.

Device Editor

Software tool used to define the settings for a device and call up information from the device

Device project

Part of the project that is downloaded to a device.

Diagnostic item

Diagnostics: Generic term for basic diagnostic item and summary diagnostic item

Diagnostic list

Diagnostics: List of a project's current diagnostic messages

Diagnostic log

Diagnostics: Log that shows when a project's diagnostic messages arrived and when they were cleared. It also contains operational entries, self test entries and debugging entries.

Download

Transfer of data to a device, e.g. from a PC to a device

Enable principle

Configuration that enables the ST section to have write access to the FS section; FS outputs can be switched via the ST section.

Ethernet infrastructure

All the hardware (e.g. cables, plug-in connections, adapters, switches, routers, hubs) and software (e.g. network services and their configuration) required for data transfer in an Ethernet network.

Failsafe (FS)

Safety-related, failsafe; a failsafe system switches to a safe condition in the event of an error.

FB

Abbreviation for Function Block

Firmware

Software created by the manufacturer to run on a device and perform device-specific functions.

Firmware update

Updating the device firmware

Force variables

The value of variables is permanently assigned and cannot be changed by the user program.

FQN

Abbreviation for fully qualified name

FS

Abbreviation for FailSafe

FS project

The "FS project" includes all the safety-related elements of a project from the user program, the hardware configuration, the resource assignment and I/O mapping.

FS resource

Execution unit for safety-related control code

Fully qualified name (FQN)

Full name that uniquely identifies an element in the namespace

Function block (FB)

IEC 61131-3: Programmable controller programming language element consisting of: 1) the definition of a data structure partitioned into input, output, and internal variables; and 2) a set of operations to be performed upon the elements of the data structure when an instance of the function block type is invoked.

FW

Abbreviation for FirmWare

Hardware input

Physical device input

Hardware output

Physical device output

Hub

Hardware of the Ethernet infrastructure, for star-shape connection of several Ethernet subscribers, data packets received at one Port, will be transmitted to all other Ports.

HW

Abbreviation for HardWare

I/O mapping

Mapping of a PI variable to I/O data from the module bus, I/O data from the external communication or another PI variable

IL

Abbreviation for Instruction List

Instruction list (IL)

Text-based, machine-oriented programming language; all operations are shown in text format

I-PI variable

Variable in the process image, which can be mapped to I data from the module bus, I data from external communication or an O-PI variable.

O-PI variable

Variable in the process image, which can be mapped to O data from the module bus, O data from external communication or an I-PI variable.

Patch cable

1:1-wired cable with a max. length of 5 m; it is primarily used to connect Ethernet components within an enclosure (control cabinet etc.).

Person, competent

A person who, through training, experience and current professional activity, has acquired the necessary knowledge and authorisation to test, assess and handle devices, systems, plant and machinery in accordance with the general standards and guidelines for safety technology

PI variable

Variable in the process image which can be mapped to I/O data from the module bus, I/O data from the external communication or another PI variable.

POU

Abbreviation for Program Organisation Unit

PRG

Abbreviation for PRoGram

Process value

Actual value of process data; in contrast to the substitute value

Program (PRG)

IEC 61131-1: "Logical assembly of all the programming language elements and constructs necessary for the intended signal processing required for the control of a machine or process by a PLC-system."

Program organisation unit (POU)

In accordance with IEC 61131-3: Function, function block or program

PSS 4000 project

Project created in PAS4000 to control and monitor a system (plant, machine). Project = User program + hardware configuration + network configuration

Quality of process data

Information on whether process data relates to the "process value" or a "substitute" value.

Real-time system

System in which the validity of a calculated result depends not only on logical accuracy but also on the time at which the result is provided. A response that is too early or too late is an incorrect response.

Resource

Unit that executes code, e.g. FS resource, ST resource

Resource assignment

Assignment of program type program organisation units or basic blocks to the resources

Router

Hardware of the Ethernet infrastructure, for connecting different networks, based on the Net-ID (= section of the IP address) the router decides which data packets are to be transmitted to a different network.

Safe condition

Condition in which power is removed from the safety circuits.

SafetyNET p infrastructure

see Ethernet infrastructure

Sealed

Property of project elements (e.g. program organisation units, component blocks). Project element is sealed, certified if necessary, and may not be modified and displayed.

ST

Abbreviation for STandard

ST project

The ST project includes all the elements of a project that do not belong to the FS project.

ST resource

Execution unit for non-safety-related control code

Standard (ST)

Not safety-related, in contrast to failsafe

Standard bus system

Non-safety-related bus system, e.g. DeviceNet or PROFIBUS, in contrast to SafetyBUS p or SafetyNET p from Pilz (see also "Fieldbus")

Substitute value

If the process data cannot be determined due to an error, it is replaced by substitute values.

SW

Abbreviation for SoftWare

Switch

Hardware of the Ethernet infrastructure, for star-shape connection of several Ethernet subscribers, data packets received at one Port, will be analysed and only transmitted to the Port where the subscriber is registered. Broadcast messages are forwarded.

Task

Execution control element providing for periodic or triggered execution of a group of associated IEC 61131 programs and/or basic blocks (see IEC 61131).

Test pulse

The test pulse is wired to an input in order to detect wiring errors in that input, e.g. open circuit, short circuit to 24 V DC or 0 V or shorts between two inputs.

Timeliness

Occurrence of an event within a defined time window; events that occur earlier or later are invalid.

Variable list

List containing the variables for which control, force or watch variables is to be executed.

Support

Technical support is available from Pilz round the clock.

Americas

Brazil

+55 11 97569-2804

Canada

+1 888 315 7459

Mexico

+52 55 5572 1300

USA (toll-free)

+1 877-PILZUSA (745-9872)

Asia

China

+86 400-088-3566

Japan

+81 45 471-2281

South Korea

+82 31 778 3390

Australia and Oceania

Australia

+61 3 95600621

New Zealand

+64 9 6345350

Europe

Austria

+43 1 7986263-444

Belgium, Luxembourg

+32 9 3217570

France

+33 3 88104003

Germany

+49 711 3409-444

Ireland

+353 21 4804983

Italy, Malta

+39 0362 1826711

Scandinavia

+45 74436332

Spain

+34 938497433

Switzerland

+41 62 88979-32

The Netherlands

+31 347 320477

Türkiye

+90 216 5775552

United Kingdom

+44 1536 460866

You can reach our international hotline on:

+49 711 3409-222

support@pilz.com

Reporting security vulnerabilities or security incidents

If you would like to report a security vulnerability or a security incident in connection with a Pilz product, please contact our **Pilz Product Security Incident Response Team (PSIRT)**.

You can reach us at: www.pilz.com/psirt

Pilz develops environmentally-friendly products using ecological materials and energy-saving technologies. Offices and production facilities are ecologically designed, environmentally-aware and energy-saving. So Pilz offers sustainability, plus the security of using energy-efficient products and environmentally-friendly solutions.



www.pilz.com/facebook



www.pilz.com/linkedin



www.pilz.com/xing



www.pilz.com/youtube



CECE, CHRE, CMSE®, IndustrialPI®, Leansafe®, MYZEL®, PAS4000®, PAScale®, PASconfig®, Pilz®, PIIT®, PMCPrimo®, PMCProtego®, PMCTendo®, PMD®, PMV®, PNOZ®, Primo®, PSN®, PSS®, PVS®, SafetyBUS p®, SafetyNET p®, THE SPIRIT OF SAFETY® are registered and protected trademarks of Pilz GmbH & Co. KG in some countries. We would point out that product features may vary from the details stated in this document, depending on the status at the time of publication and the scope of the equipment. We accept no responsibility for the validity, accuracy and entirety of the text and graphics presented in this information. Please contact our Technical Support if you have any questions.

We are represented internationally. Please refer to our homepage www.pilz.com for further details or contact our headquarters.

Headquarters: Pilz GmbH & Co. KG, Felix-Wankel-Straße 2, 73760 Ostfildern, Germany
Telephone: +49 711 3409-0, E-Mail: info@pilz.com, Internet: www.pilz.com

PILZ
THE SPIRIT OF SAFETY