



オートメーションシステムPSS 4000、PSS
4000ファームウェアV1.24.x

PILZ
THE SPIRIT OF SAFETY

この資料はオリジナル資料です。

避けられない場合、読みやすくするため、本書では、男性形が選択されています（英語のhe, hisなど）。
当社はすべての人を差別することなく、平等と見なすことを保証します。

この資料に関するすべての権利はPilz GmbH & Co. KGが所有しています。複製は、ユーザの社内用途でのみ許可されます。本書を改善するための提案およびコメントをお待ちしています。

Pilz®、PIT®、PMI®、PNOZ®、Primo®、PSEN®、PSS®、PVIS®、SafetyBUS p®、
SafetyEYE®、SafetyNET p®、the spirit of safety®は、各国におけるPilz GmbH & Co. KGの登録
商標であり、保護されています。



SDはSecure Digitalの略号です。

1	はじめに	15
1.1	取扱説明書の有効性	15
1.2	記号の定義	16
1.3	サードパーティ製造業者のライセンス情報	17
2	オートメーションシステムPSS 4000	18
2.1	PSS 4000の概要	18
2.1.1	全般	18
2.1.2	SafetyNET p	19
2.1.3	オートメーションシステムPSS 4000に含まれる装置	20
2.1.3.1	システムの概要	20
2.1.3.2	システムセクション	22
2.1.3.3	PSS 4000専用装置	25
2.1.3.4	PSS 4000に対応する装置	26
2.1.3.5	PSS 4000対応装置	26
2.1.4	PAS4000	27
2.1.5	診断	28
2.1.6	OPCサーバ	29
2.2	PSS 4000のユニット特性	30
2.2.1	ネットワークシステムの集中表示	30
2.2.2	装置の拡張性	31
2.2.3	プログラミング内でのモジュール化	32
3	安全	33
3.1	その他の参照資料	33
3.2	用途	34
3.2.1	安全度水準 (SIL) およびパフォーマンスレベル (PL)	35
3.2.2	一般的な安全評価	36
3.3	有資格者の採用	37
3.4	安全入出力	38
4	セキュリティ	39
5	PSSuシステムの基本情報	40
5.1	構造	41
5.2	モジュールのレイアウト	44
5.3	モジュールの型番	46
5.4	モジュールバス (MBUS)	48
5.5	ベースユニット上の接続	50
5.5.1	入出力モジュールの接続レベル	51
5.5.2	接続レベルの色分け	52
5.6	コンパクトモジュール上の接続	53

5.7	PSSuシステムの増設の制限	54
5.7.1	PSSuシステムでのモジュール数と入出力数	55
5.7.2	最大電流負荷	57
5.7.2.1	電圧供給モジュールの最大電流負荷容量	57
5.7.2.2	温度に基づいた最大電流負荷容量	58
5.8	PSSuシステムのスロット	59
5.8.1	PSSuシステムでの設定／実際のハードウェアレジストリ	61
5.9	供給電圧と絶縁	64
5.10	供給グループ	67
5.11	最大周囲温度	69
5.12	ヘッドモジュールのステータスLED	70
6	Pss67装置に関する基本	77
6.1	構造	78
6.2	モジュールバス (MBUS)	79
6.3	PSS67装置のスロット	79
6.4	供給電源	80
6.5	ステータスLED	81
7	イーサネット	82
7.1	Auto IP機構	82
7.2	イーサネットインタフェース	83
7.2.1	RJ45インタフェース	84
7.2.2	RJ45インタフェースの割り付け	85
7.2.3	RJ45のコネクタと接続ケーブル	86
7.2.4	RJ45インタフェースのステータスLED	88
8	SafetyNET p	89
8.1	概要	89
8.1.1	SafetyNET pプロトコルのバージョン	90
8.1.2	ISO/OSIの7階層参照モデルのSafetyNET p	91
8.1.3	RTFN (リアルタイムフレームネットワーク)	92
8.1.4	安全関連および非安全関連データの転送	93
8.2	SafetyNET pのトポロジ	94
8.2.1	リニアトポロジ	94
8.2.2	スタートポロジ	95
8.2.3	動的構造	96
8.3	SafetyNET pコンポーネントの選択	97
8.4	SafetyNET pインタフェース	98
8.5	SafetyNET pを介したプロセスデータ交換	99
8.5.1	接続の周期的な処理	101
8.5.2	SafetyNET p受信タイムアウト	102

8.5.2.1	SNpプロトコルバージョン1: 通信エラーおよびFS通信の再開	103
8.5.2.2	SNpプロトコルバージョン2: 通信エラーおよびFS通信の再開	104
8.5.2.3	SNpプロトコルバージョン2: コンフィグレーション可能なFS SafetyNET p待機時間	105
8.5.3	データ転送の制限	107
8.6	プロジェクト内SafetyNET p通信	110
8.7	プロジェクト間SafetyNET p通信	111
8.8	FSハートビート (FS-HBT)	113
8.9	SafetyNET pスケジューリングモード	115
8.10	ルーターのないネットワーク	116
8.11	ルーターのあるネットワーク	118
8.11.1	システム変数PILZ_SNP_MULTICAST_TTL	120
8.12	無線LAN経由のSafetyNET p	121
8.13	SafetyNET pで最適化されるアベイラビリティ	122
8.14	同一プロジェクト名を持つプロジェクトとのデータ交換の防止	123
9	外部通信	124
9.1	外部通信の通信モード	124
9.1.1	フィールドバス	124
9.1.2	IP接続	124
9.2	Modbus/TCP	126
9.2.1	「IP接続」システムセクション	127
9.2.2	Modbus/TCP接続	128
9.2.3	Modbus/TCPサーバ接続 (サーバ接続)	129
9.2.4	ファンクションコード (クライアント接続)	130
9.2.5	ファンクションコードとModbus/TCPデータ領域の関係	132
9.2.6	Modbus/TCPを使ったデータ交換	133
9.2.6.1	送信データと受信データ	133
9.2.6.2	サーバロケーションテーブル	134
9.2.7	データ転送の制限	136
9.2.8	Modbus/TCPのI/Oデータ	138
9.2.9	Modbus/TCPアドレスのアドレス表示	142
9.2.9.1	送信データのアドレスフォーマット	142
9.2.9.2	受信データのアドレスフォーマット	142
9.2.9.3	サーバロケーションテーブルでのアドレス表示	142
9.2.10	Modbus/TCPのデータの整合性	143
9.2.11	データ領域0xと4x使用時の特殊な機能	144
9.3	Raw TCP	148
9.3.1	「IP接続」システムセクション	149
9.3.2	Raw TCP接続	150
9.3.3	Raw TCPを使ったデータ交換	151
9.3.4	受信データ用FIFOメモリ	152
9.3.5	Raw TCPによる受信データの処理	153

9.3.5.1	Raw TCPの一般的なアプリケーションシナリオ	154
9.3.5.2	受信データの処理時の基本的なワークフロー	156
9.3.5.3	FIFOメモリがイネーブルの場合の受信動作	157
9.3.5.4	FIFOメモリが無効の場合の受信動作	159
9.3.6	データ転送の制限	161
9.3.7	Raw TCPのI/Oデータ	162
9.3.8	Raw TCPアドレスのアドレスフォーマット	165
9.3.9	Raw TCPのデータの整合性	166
9.4	Raw UDP	167
9.4.1	「IP接続」システムセクション	168
9.4.2	Raw UDPの接続	169
9.4.3	Raw UDPを使ったデータ交換	170
9.4.4	受信データ用FIFOメモリ	171
9.4.5	データ転送の制限	172
9.4.6	Raw UDPのI/Oデータ	173
9.4.7	Raw UDPアドレスのアドレスフォーマット	176
9.4.7.1	送信データのアドレスフォーマット	176
9.4.7.2	受信データのアドレスフォーマット	176
9.4.8	Raw UDPのデータの整合性	177
9.5	ブロックを使用するUDP通信	178
9.5.1	「IP接続」システムセクション	179
9.5.2	UDPソケット	180
9.5.3	「ブロックを使用するUDP」経由のデータ交換	181
9.5.4	UDP送信バッファとUDP受信バッファ	182
9.5.5	データ転送の制限	183
9.5.6	UDPデータグラム送信時の時間動作	184
9.5.7	UDPデータグラム受信時の時間動作	186
9.5.8	ブロックを使用するUDP通信中の時間の要件	187
9.5.9	最大UDP送信速度と最大UDP受信速度	190
9.5.10	ブロックを使用するUDP通信中のデータの整合性	191
9.5.11	ブロックを使用する安全なUDP通信	192
9.6	PROFIBUS-DPスレーブ	192
9.6.1	「PROFIBUS-DPスレーブ」システムセクション	194
9.6.2	PROFIBUS-DPを経由したデータ交換	195
9.6.2.1	送信データと受信データ	195
9.6.2.2	送信データ用仮想PROFIBUSモジュール	197
9.6.2.3	受信データ用仮想PROFIBUSモジュール	199
9.6.3	データ転送の制限	201
9.6.4	PROFIBUS-DPスレーブのI/Oデータ	202
9.6.5	PROFIBUS-DPアドレスのアドレス表示	205
9.6.5.1	送信データのアドレスフォーマット	205
9.6.5.2	受信データのアドレスフォーマット	206

9.6.6	PROFIBUSの診断	207
9.6.6.1	システム内部の診断	208
9.6.6.2	外部診断データ	209
9.6.7	PROFIBUS-DPのデータの整合性	212
9.7	PROFIsafe搭載PROFIBUS-DPスレーブ	213
9.7.1	「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクション	214
9.7.2	「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションのPROFIBUS-DP	215
9.7.3	「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションのPROFIsafe	216
9.7.3.1	事前定義済みFモジュール	217
9.7.3.2	送信データ用仮想PROFIsafeサブモジュール	220
9.7.3.3	受信データ用仮想PROFIsafeサブモジュール	221
9.7.4	仮想PROFIBUSモジュールのGSDファイル	222
9.7.5	データ転送の制限	223
9.7.6	PROFIsafeのI/Oデータ	224
9.7.7	PROFIsafeアドレスのアドレスフォーマット	227
9.7.7.1	送信データのアドレスフォーマット	227
9.7.7.2	受信データのアドレスフォーマット	228
9.7.8	Fモジュールの無効化と再有効化	229
9.7.9	Fパラメータ	230
9.7.10	PROFIsafeの診断	231
9.7.10.1	F装置の装置ベースの診断	232
9.7.11	PROFIsafeのデータの整合性	234
9.8	EtherNet/IPアダプタ	235
9.8.1	「EtherNet/IPアダプタ」システムセクション	237
9.8.2	EtherNet/IPアダプタの接続	238
9.8.3	EtherNet/IPを使ったデータ交換	239
9.8.3.1	送信データと受信データ	239
9.8.3.2	送信データ用仮想モジュール	240
9.8.3.3	受信データ用仮想モジュール	241
9.8.4	データ転送の制限	242
9.8.5	EtherNet/IPアダプタのI/Oデータ	243
9.8.6	EtherNet/IPアドレスのアドレスフォーマット	246
9.8.6.1	送信データのアドレスフォーマット	246
9.8.6.2	受信データのアドレスフォーマット	247
9.8.7	EtherNet/IPの診断	248
9.8.8	EtherNet/IPのデータ整合性	250
9.8.9	サポートされているEtherNet/IPオブジェクトと共通のサービス	251
9.8.9.1	アイデンティティオブジェクト (クラスID 0x01)	252
9.8.9.2	アセンブリオブジェクト (クラスID 0x04)	254
9.8.9.3	TCP/IPインタフェースオブジェクト (クラスID 0xF5)	255
9.8.9.4	リンクオブジェクト (クラスID 0xF6)	256
9.8.9.5	共通のサービス	258

9.9	PROFINET IO DEVICE	259
9.9.1	「PROFINET IO DEVICE」システムセクション	260
9.9.2	PROFINETでの装置の組み込みとアドレス指定	261
9.9.3	PROFINET内のPSS 4000装置の識別	262
9.9.4	PROFINETを経由したデータ交換	263
9.9.4.1	データの送受信	263
9.9.4.2	送信データ用仮想PROFINETモジュール	264
9.9.4.3	受信データ用仮想PROFINETモジュール	266
9.9.5	GSDMLファイル	268
9.9.6	データ転送の制限	269
9.9.7	PROFINET IO DEVICEのI/Oデータ	270
9.9.8	PROFINET IOアドレスのアドレスフォーマット	273
9.9.8.1	送信データのアドレスフォーマット	273
9.9.8.2	受信データのアドレスフォーマット	274
9.9.9	PROFINET IOの診断	276
9.9.10	PROFINET IOのデータ整合性	277
10	プロジェクト	278
10.1	プロジェクト構造体	278
10.2	処理データの有効性	281
10.2.1	代入値	284
10.2.2	ユーザプログラムの有効なビットの評価	285
10.3	タスクの基本情報	286
10.3.1	タスクのプロパティ	287
10.3.2	タスクの実行	288
10.3.3	タスクの既定のコンフィグレーション	291
10.4	PSS 4000プロジェクト内のI/Oマッピングルール	292
10.5	PSS 4000プロジェクト間I/Oマッピングのルール	296
10.6	データの一貫性	300
10.6.1	一貫性の範囲	301
10.6.2	データの整合性に対するスケジューリングの影響	304
10.7	ブロックの安全ステータス	305
10.8	PAS4000による装置命名	307
10.8.1	装置の識別データ	308
10.8.2	命名データ	309
10.8.3	PSS 4000専用装置の装置命名	310
10.8.3.1	装置命名中のエラー動作	314
10.8.4	PSS 4000対応装置の装置命名	315
10.9	PSS 4000専用装置を交換する際の装置の命名	318
10.10	プロジェクトのダウンロード	319
10.10.1	プロジェクトの直接ダウンロード	319
10.10.2	プロジェクトの間接ダウンロード	325

10.11	シールされたFSブロック	327
11	オペレーティングモード	328
11.1	PAS4000とPSS 4000装置の間の通信	328
11.1.1	ポイントツーポイント接続の確立	329
11.1.2	リモート接続の確立	330
11.2	オペレーティングステータスと装置のオペレーティングステータスの変更	331
11.2.1	フローチャート	332
11.2.2	装置のオペレーティングステータス	333
11.2.2.1	装置停止	334
11.2.2.2	オペレーティングステータス「ベースシステムを起動中」	335
11.2.2.3	オペレーティングステータス「システムセクションを起動中」	336
11.2.2.4	オペレーティングステータス「装置がエラーなしで実行状態」	342
11.2.2.5	オペレーティングステータス「装置がエラーなしで停止状態」	343
11.2.2.6	オペレーティングステータス「Major FS error」	344
11.2.3	装置のオペレーティングステータスの変更	346
11.3	オペレーティングステータスとFSリソースのオペレーティングステータスにおける変化	347
11.3.1	フローチャート	348
11.3.2	FSリソースのオペレーティングステータス	349
11.3.2.1	FSリソースがエラーなしで実行状態	350
11.3.2.2	FSリソースがエラーなしで停止状態	351
11.3.2.3	FSリソースがエラーで実行状態: タスクが終了状態	352
11.3.2.4	FSリソースがエラーで実行状態: タスクが停止状態	353
11.3.2.5	Major FS error	354
11.3.2.6	Fatal error	355
11.3.3	FSリソースのオペレーティングステータスの変化	356
11.4	オペレーティングステータスとSTリソースのオペレーティングステータスにおける変化	359
11.4.1	フローチャート	360
11.4.2	STリソースのオペレーティングステータス	361
11.4.2.1	STリソースがエラーなしで実行状態	362
11.4.2.2	STリソースがエラーなしで停止状態	363
11.4.2.3	STリソースがエラーで実行状態: タスクが終了状態	364
11.4.2.4	STリソースがエラーで実行状態: タスクが停止状態	365
11.4.2.5	Minor ST error	366
11.4.3	STリソースのオペレーティングステータスの変化	367
11.5	オペレーティングステータスとFSモジュールバスPSSuのオペレーティングステータスの変更	370
11.5.1	フローチャート	371
11.5.2	FSモジュールバスPSSuのオペレーティングステータス	372
11.5.2.1	FSモジュールバスPSSuがRUN状態、エラーなし	373
11.5.2.2	FS入力モジュールがRUN状態、エラーなし	374
11.5.2.3	FSモジュールが入力エラーで実行状態	375
11.5.2.4	FS入力モジュールがモジュールエラーでSTOP状態	376

11.5.2.5	装置のFS入力モジュールがモジュールエラーでSTOP状態	377
11.5.2.6	FS出力モジュールがRUN状態、エラーなし	378
11.5.2.7	装置のすべてのFS出力が安全状態	379
11.5.2.8	Major FS error	380
11.5.3	FSモジュールバスPSSuのオペレーティングステータスの変更	381
11.6	オペレーティングステータスとSTモジュールバスPSSuのオペレーティングステータスの変化	383
11.6.1	フローチャート	384
11.6.2	STモジュールバスPSSuのオペレーティングステータス	385
11.6.2.1	STモジュールバスPSSuがRUN状態、エラーなし	386
11.6.2.2	ST入力モジュールがRUN状態、エラーなし	387
11.6.2.3	STモジュールが入力エラーで実行状態	388
11.6.2.4	ST入力モジュールがモジュールエラーでSTOP状態	389
11.6.2.5	ST出力モジュールがRUN状態、エラーなし	390
11.6.2.6	STモジュールが出力エラーで実行状態	391
11.6.2.7	ST出力モジュールがモジュールエラーでSTOP状態	392
11.6.2.8	Minor ST error	393
11.6.3	STモジュールバスPSSuのオペレーティングステータスの変更	394
11.7	オペレーティングステータスとFS SafetyNET pのオペレーティングステータスの変更	396
11.7.1	フローチャート	397
11.7.2	FS SafetyNET pのオペレーティングステータス	398
11.7.2.1	FS SafetyNET pがRUN状態、エラーなし	399
11.7.2.2	Major FS error	400
11.7.2.3	Major FS+ST error	401
11.7.2.4	SNpプロトコルバージョン1: FS SafetyNET pインタフェースがマイナーエラーでRUN状態	402
11.7.2.5	SNpプロトコルバージョン2: FS SafetyNET pインタフェースがマイナーエラーでRUN状態	403
11.7.3	FS SafetyNET pのオペレーティングステータスの変更	405
11.8	オペレーティングステータスとST SafetyNET pのオペレーティングステータスの変更	407
11.8.1	フローチャート	408
11.8.2	ST SafetyNET pのオペレーティングステータス	409
11.8.2.1	ST SafetyNET pがRUN状態、エラーなし	410
11.8.2.2	Major FS+ST error	411
11.8.2.3	ST SafetyNET pがマイナーエラーでRUN状態	412
11.8.3	ST SafetyNET pのオペレーティングステータスの変更	413
11.9	オペレーティングステータスと外部通信のオペレーティングステータスにおける変化	414
11.9.1	フローチャート	415
11.9.2	外部通信でのオペレーティングステータス	416
11.9.2.1	システムセクションがエラーなしで実行状態	417
11.9.2.2	外部通信のシステムセクションにマイナーエラー	418
11.9.2.3	システムセクションがエラーで実行状態	419
11.9.2.4	Major FS+ST error	420
11.9.2.5	Major FS error	421
11.9.3	外部通信のオペレーティングステータスにおける変化	422

11.10	リセット、再起動、起動および停止	424
11.10.1	概要	425
11.10.1.1	リセット、再起動、起動、停止のオプション	425
11.10.1.2	変数とアプリケーションパラメータへの影響	426
11.10.1.3	アクティブな「変数強制」への影響	427
11.10.1.4	「変数の制御」への影響	428
11.10.1.5	有効なオンライン変更時の影響	429
11.10.2	コールドスタートと再起動コマンド	430
11.10.3	リセットコマンド	432
11.10.4	ホットスタートコマンド	434
11.10.5	再起動コマンド	435
11.10.6	停止コマンド	437
11.11	リセットボタンの機能	439
11.11.1	装置のウォームリセットの実行	440
11.11.2	意図的なオペレータ操作による装置のオリジナルリセット	441
11.11.3	意図的なオペレータ操作による命名データ／装置プロジェクトの転送	443
11.11.4	リカバリモード	448
11.12	オリジナルリセット	449
11.12.1	オリジナルリセットの影響	450
11.12.1.1	PSS 4000専用装置での効果	451
11.12.1.2	PSS 4000対応装置での影響	452
11.12.2	オリジナルリセットの実行	453
11.13	イネーブルの原理	454
11.14	応答時間	456
11.14.1	応答時間に影響を与える時間	457
11.14.2	データ処理時の処理規則	463
11.14.3	データパスの構成	464
11.14.4	データサブパスの最大応答時間	466
11.14.4.1	モジュールバスがデータソースの場合のデータサブパスの最大応答時間	466
11.14.4.2	SafetyNET pがデータソースのデータサブパス	467
11.14.4.3	外部入力データがデータソースのデータサブパス	468
11.14.4.4	タスクがデータソースのデータサブパス	469
11.14.4.5	1台の装置上でのデータパスの計算例	470
11.14.4.6	3台の装置を使用したデータパスの計算例	471
11.15	システム時刻の同期	472
11.15.1	SNTPとのシステム時刻の同期	473
11.16	メモリ	474
11.16.1	メモリタイプ	474
11.16.2	メモリ構造	475
11.16.3	セキュリティSDカード	478
11.16.4	ジャーナリングファイルシステム	478
11.16.5	SDカードの関連付け	478

12	試運転	479
12.1	一般的な要件	479
12.2	初回試運転	480
12.3	再試運転	483
13	メンテナンス、修理、廃棄	486
13.1	メンテナンスと修理	486
13.2	廃棄	487
14	OPCサーバ (クラシック)	488
14.1	概要	488
14.1.1	設計オプション	489
14.1.2	OPCプロパティ	491
14.2	システム要件	492
14.2.1	Microsoft Windowsオペレーティングシステム搭載PC	492
14.3	ライセンス登録	493
14.4	取り付け	494
14.4.1	PCでのPSS 4000ファームウェアのインストール	495
14.4.1.1	PSS 4000ファームウェアの初回インストール	499
14.5	PCの通信とセキュリティ設定	500
14.5.1	DCOMコンフィグレーション	501
14.5.1.1	DCOMコンフィグレーションの開始	501
14.5.1.2	OPCサーバとOPCクライアントのDCOMコンフィグレーション	504
14.5.1.3	OPCServerのDCOMコンフィグレーション	510
14.5.1.4	OPCサーバのOPCEnumコンフィグレーション	519
14.5.1.5	ユーザ「Administrators」の特別な機能	528
14.5.1.6	ユーザ「NETWORK」の特別な機能	529
14.5.2	ファイアウォールのコンフィグレーション	530
14.6	OPCサーバの試運転	531
14.7	データ型の割り付け	532
14.8	PSS 4000ネームスペース内のデータへのアクセス	533
14.9	ネットワークデータ (OPCサーバ (クラシック))	533
15	OPCサーバUA	535
15.1	概要	535
15.1.1	設計オプション	536
15.1.2	OPC UAプロパティ	538
15.1.3	制限値	540
15.1.4	OPC UAクライアントのアクセス権	542
15.1.5	エンドポイントのセキュリティ設定	542
15.2	システム要件	543
15.2.1	Microsoft Windowsオペレーティングシステム搭載PC	543

15.3	ライセンス登録.....	544
15.4	インストール.....	545
15.4.1	PCへのPSS 4000ファームウェアのインストール.....	546
15.5	情報セキュリティ証明書.....	550
15.5.1	インストール中の証明書の作成.....	551
15.5.2	証明書ストアのディレクトリ構造.....	552
15.5.3	認証の手順.....	553
15.5.3.1	ローカルインストールの場合の認証.....	553
15.5.3.2	リモートインストールの場合の認証.....	553
15.5.3.3	サードパーティ製造業者のOPC UAクライアントの場合の認証.....	557
15.5.4	カスタム証明書の生成および使用.....	558
15.5.5	証明書の無効化.....	560
15.5.6	証明書の確認.....	560
15.6	ファイアウォールのコンフィグレーション.....	561
15.7	データ型の割り付け.....	562
15.7.1	PSS 4000ネームスペース内のデータへのアクセス.....	563
15.7.2	ネームスペース内の配列にアクセスする際の特別な機能.....	563
15.8	ネットワークデータ (OPCサーバUA).....	564
16	診断の基本情報	566
16.1	診断リスト.....	567
16.1.1	診断メッセージ.....	568
16.1.1.1	診断メッセージのステータス.....	570
16.1.1.2	重篤度と優先度.....	571
16.1.1.3	責任者.....	572
16.1.1.4	範囲.....	573
16.1.1.5	ロケーション情報.....	574
16.1.1.6	診断ID.....	579
16.2	診断ログ.....	581
16.2.1	ログエントリ.....	582
16.3	診断サーバ.....	584
17	PSS 4000ネームスペース	586
17.1	一般的なプロジェクトデータ.....	588
17.1.1	Projectinformation.....	589
17.1.2	user_prg.....	590
17.1.3	Diag.....	592
17.2	装置ベースのプロジェクトデータ.....	596
17.2.1	State.....	598
17.2.2	DeviceProject.....	607
17.2.3	Product.....	608
17.2.4	Network.....	610

17.2.5	FS resource.....	611
17.2.6	ST resource	613
17.3	PSS 4000ネームスペースのデータ型.....	615
18	付録	617
18.1	さまざまなサービスのポート番号に対する工場出荷時の既定設定	617
18.2	Ethernetパラメータの出荷時の既定の設定	618
	用語集	619

1 はじめに

1.1 取扱説明書の有効性

本書は、オートメーションシステムPSS 4000に対して有効です。本書の内容は、新しい取扱説明書が発行されるまで有効です。

最新のマニュアルは、ピルツホームページのダウンロード領域 (www.pilz.comのダウンロードページ) でも入手できます。

本書では、「オートメーションシステムPSS 4000」の略語として「PSS 4000」を使用することがあります。

1.2 記号の定義

特に重要な情報については、次のように区別して示しています。



危険！

この警告には必ず従ってください。重傷や死亡が発生する恐れのある差し迫った危険が存在する状況を警告し、推奨される予防措置を提示しています。



警告！

この警告には必ず従ってください。重傷や死亡が発生する恐れのある危険な状況を警告し、推奨される予防措置を提示しています。



注意！

比較的軽度の怪我や物的破損が発生する危険な状況を警告し、推奨される予防措置を提示しています。



重要

この記号は、製品または装置が損傷する可能性がある状況について説明しています。また、実施可能な予防措置も示しています。また、文中の特に重要な個所を強調表示しています。



情報

この記号は、アプリケーションに関するアドバイスを示し、特殊な機能に関する情報を提供します。



準備中

この機能または説明はまだ利用可能ではありません。
安全関連の機能や説明には関係しません。

1.3 サードパーティ製造業者のライセンス情報

各種製品には、さまざまなライセンスが設定されたオープンソースソフトウェアが含まれています。

詳細については、該当する製品の取扱説明書を参照してください。

2 オートメーションシステムPSS 4000

2.1 PSS 4000の概要

2.1.1 全般

PSS 4000は、非常に広範なオートメーション機能への対応に使用できるオートメーションシステムです。これらの機能に関しては、制御技術 (PLC)、モーションコントロール、および可視化の分野でそれぞれの要件があります。

PSS 4000は、小規模分散型のオートメーションターミナルから、制御レベルの複雑なネットワークまで、あらゆる段階のオートメーションで使用できます。

PSS 4000オートメーションシステムは、制御、モーションコントロール、および可視化のタスクのために、非安全関連機能 (ST機能) とともに、安全関連機能 (FS機能) にも対応しています。

2.1.2 SafetyNET p

SafetyNET pは、ネットワーク化されたオートメーション機能に使用します。SafetyNET pは、産業用途に適したイーサネットベースのマルチマスタバスシステムです。

SafetyNET pプロトコルには、安全関連データ用の安全データチャンネルが含まれています。このチャンネルは、IEC 61508のSIL 3に適合したデータ転送で認証されています。SafetyNET pでは、安全関連データと非安全関連データは、同一接続ケーブル上の異なるデータチャンネル経由で転送されます。

トポロジに関して、SafetyNET pは一般的なイーサネットアーキテクチャをサポートしているため、モジュール式のシステムアーキテクチャと個別のサブネットワークを実装することができます。必要に応じて、この方法を使用してオートメーション機能を分散型の独立した機能ユニットに分けることができます。このタイプのアーキテクチャの主な利点は、機能ユニットの再利用、変更、アベイラビリティにあります。

2.1.3 オートメーションシステムPSS 4000に含まれる装置

2.1.3.1 システムの概要

オートメーションシステムPSS 4000に含まれる装置を**PSS 4000装置**と呼びます。各装置の分類を次の図に示します。

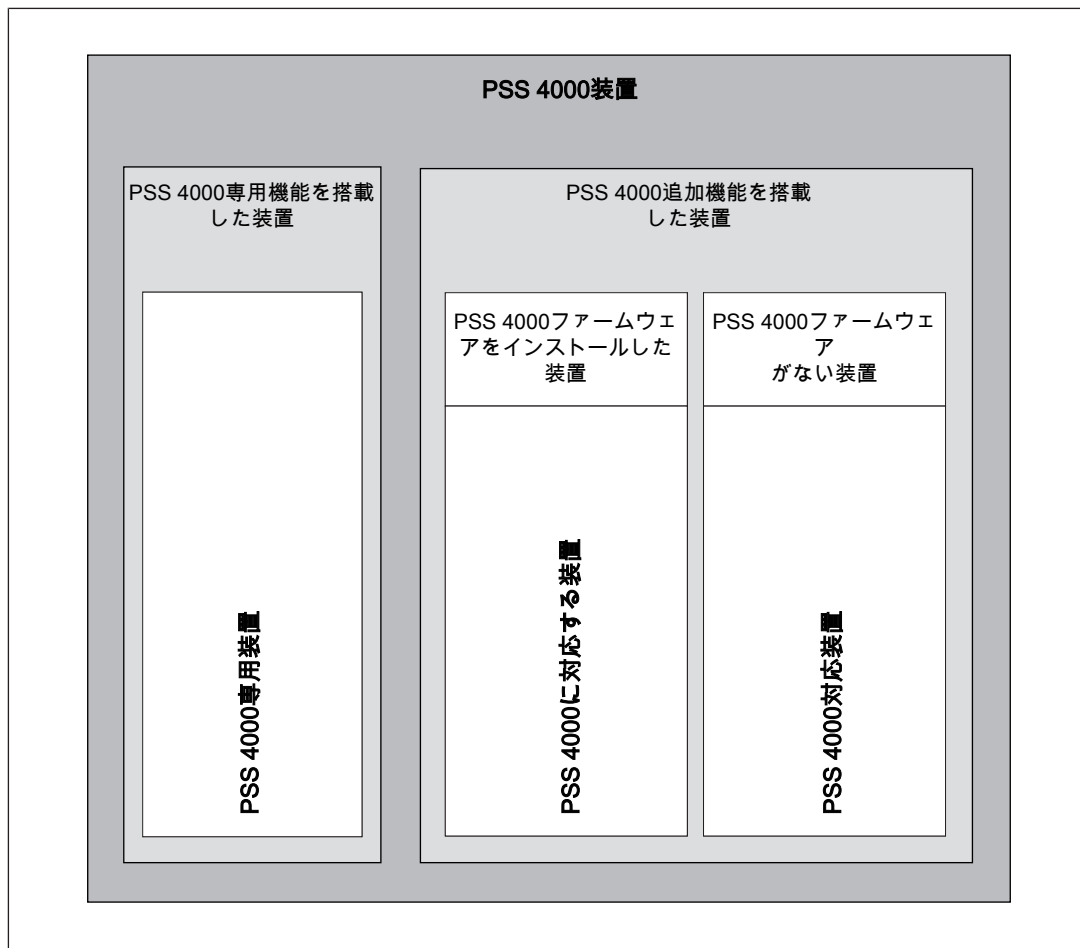


図: オートメーションシステムPSS 4000の装置

PSS 4000装置とは、PSS 4000のファームウェアがインストールされているか、PSS 4000のファームウェアをインストールできる装置です。これらの装置には、オートメーションシステムPSS 4000の基本プロパティがあります。重要な基本プロパティは次の通りです。

- ▶ PAS4000の「PSS 4000ネットワークをスキャン」を使用した装置の検出
- ▶ PSS 4000診断システムへの組み込み
- ▶ PSS 4000のプロジェクトダウンロードによってそれぞれのコンフィグレーションを取得
- ▶ PSS 4000のプロジェクトダウンロードによってそれぞれの装置プロジェクトを取得

PSS 4000装置はさらに次のように分類されます。

- ▶ PSS 4000専用機能を備えた装置 (PSSuシステムのヘッドモジュールなど) を「**PSS 4000専用装置**」といいます。

▶ 追加機能としてPSS 4000の機能を備えた装置は、さらに次のように分類されます。

– **PSS 4000に対応する装置**

PSS 4000のファームウェアがインストールされている装置です。

– **PSS 4000対応装置**

PSS 4000ファームウェアがインストールされていない装置のうち、PSS 4000ファームウェアをインストールできる装置です。

2.1.3.2 システムセクション

システムセクションは、定義済みの機能ユニットであり、装置のユーティリティをユーザが認識できる形で提供します。PSS 4000専用装置またはPSS 4000対応装置でシステムセクションを実行するには、対応するハードウェアの要件を満たしている必要があります。

PSS 4000専用装置またはPSS 4000対応装置で使えるシステムセクションは、装置によって異なります。ピルツのPSS 4000専用装置とPSS 4000対応装置の詳細については、該当する装置の取扱説明書を参照してください。

使用可能なシステムセクションの概要

▶ リソース

- FSリソースシステムセクション
安全関連制御プログラムの実行ユニット
- STリソースシステムセクション
非安全関連制御プログラムの実行ユニット

▶ モジュールバス

- FSモジュールバスPSSuシステムセクション
FS電子モジュール／FSコンパクトモジュールと、この装置上の他のシステムセクションとの間のデータ交換用ユニット
- STモジュールバスPSSuシステムセクション
ST電子モジュール／STコンパクトモジュールと、同じ装置上の他のシステムセクションとの間のデータ交換用ユニット

▶ SafetyNET p

- FS SafetyNET pシステムセクション
SafetyNET p経由でFS I/Oデータを伝送するためのユニット
- ST SafetyNET pシステムセクション
SafetyNET p経由でST I/Oデータを伝送するためのユニット

▶ 外部通信

- IP接続システムセクション
IP接続 (Modbus/TCP、Raw TCP、Raw UDPなど) を経由して外部I/Oデータを伝送するためのユニット
- PROFIBUS-DPスレーブシステムセクション
外部I/OデータをPROFIBUS-DPスレーブとして伝送するためのユニット
- PROFIsafe搭載PROFIBUS-DPスレーブシステムセクション
非安全関連外部I/OデータをPROFIBUS-DPスレーブとして伝送し、安全関連外部I/OデータをPROFIsafe F装置として伝送するためのユニット
- EtherNet/IPアダプタシステムセクション
非安全関連外部I/OデータをEtherNet/IPアダプタとして伝送するためのユニット

– PROFINET IO DEVICEシステムセクション

外部I/OデータをPROFINET IO Deviceとして伝送するためのユニット

▶ サーバ

– 診断サーバシステムセクション

診断データを提供するためのユニット

– システムセクションOPCサーバ (クラシック)

「OPC Classic」規格に適合した標準のソフトウェアインタフェースを備えたユニットで、可視化データを提供するために使用されます。

– システムセクションOPCサーバUA

「OPC UA」規格に適合した標準のソフトウェアインタフェースを備えたユニットで、可視化データを提供するために使用されます。

PSS 4000専用装置／対応装置のシステムセクションの概要

製品	FSリソース	STリソース	FSモジュールバスPSSu	STモジュールバスPSSu	FS SafetyNET p	ST SafetyNET p	IP接続	PROFIBUS-DPスレーブ	PROFIsafe搭載 PROFIBUS-DPスレー	EtherNet/IPアダプタ	PROFINET IO DEVICE	診断サーバ	OPCサーバ (クラシック)	OPCサーバ/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)														
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)														
PSSu H FS SN SD M12 (-T)(-R)	---	---	◆	◆	◆	◆	---	---	---	---	---	---	---	---
PSS67 PLC1 16FDI	◆	◆	◆	---	◆	◆	◆	---	---	◆ ¹⁾	◆ ¹⁾	---	---	---
PSS67 IO1 16FDI	---	---	◆	---	◆	---	---	---	---	---	---	---	---	---
PMI v8	---	---	---	---	---	---	---	---	---	---	---	◆	---	◆
PC用PSS 4000ファームウェア	---	---	---	---	---	---	---	---	---	---	---	◆	◆	◆

¹⁾ 装置には外部通信用のシステムセクションが2つありますが、一度に1つのシステムセクションしか使用できません。

2.1.3.3

PSS 4000専用装置

PSS 4000専用装置には、PSSuシステムとPSS67装置が含まれます。

装置で使用できるシステムセクションは、装置によって異なります（「[システムセクション](#)」 22」を参照）。詳細は、関連する装置の取扱説明書に記載しています。

パフォーマンスクラス

使用可能なパフォーマンスクラスは設計に応じて異なります。

設計	パフォーマンスクラス		
	I/O	multi	PLC
PSSu	x	x	x
PSS67	x	- - -	x

パフォーマンスクラスは、装置の基本的な機能を決定します。パフォーマンスクラスは次のように分類されます。

▶ PSS 4000 I/O (略称: I/O)

制御機能を備えていないPSS 4000専用装置

パフォーマンスクラス「分散型システムPSSu I/O」が文書化されているPSSuシステムがあります。PSS 4000の視点では、これはパフォーマンスクラスPSS 4000 I/Oに対応します。

▶ PSS 4000 multi (略称: multi)

コンパクト安全制御システムの機能を備えたPSS 4000専用装置

パフォーマンスクラス「制御システムPSS multi」が文書化されたPSSuシステムがあります。PSS 4000の視点では、これはパフォーマンスクラスPSS 4000 multiに対応します。

▶ PSS 4000 PLC (略称: I/O)

オートメーションシステムを制御するためのPSS 4000 CPU

パフォーマンスクラス「制御システムPSSu PLC」が文書化されているPSSuシステムがあります。PSS 4000の視点では、これはパフォーマンスクラスPSS 4000 PLCに対応します。

PSSuシステム

PSSuシステムは、必要な機能を満たせるよう、さまざまなモジュールで構築されています。モジュラ式構造により、変更や調整に容易に対応することができます。PSSuシステムでは、機能範囲を遡及的に適合させることができる（例: I/O数の増加、リレー出力の使用などのI/O技術の拡張／変更、アナログ評価、カウンタ機能、温度評価などのI/O機能の追加など）だけでなく、パフォーマンスクラスが低いPSSuシステム（例: PSS 4000 I/O）をパフォーマンスクラスが高いPSSuシステム（例: PSS 4000 PLC）に変換することもできます。

PSSuシステムは、PAS4000のPSSu Module Editorでコンフィグレーションします。

PSS67装置

原則として、PSS67装置はIP67アプリケーション用のコンパクト設計のPSSuシステムです。モジュラ設計ではないため、機能範囲を遡及的に適合させることはできません。

PSS67装置は、PAS4000のPSS67エディタでコンフィグレーションします。

2.1.3.4 PSS 4000に対応する装置

PSS 4000ファームウェアは、工場出荷時に、ピルツのさまざまなオペレーターミナルPMIにあらかじめインストールされています。PSS 4000ファームウェアが事前にインストールされているため、これらのオペレーターミナルはピルツの「PSS 4000対応装置」と見なされます。その結果、これらのオペレーターミナルには、OPCサーバUAや診断サーバなどのシステムセクションがあります。

PSS 4000ファームウェアをインストールすることで、対応するシステム要件を持つPCをサードパーティの「PSS 4000対応装置」にすることができます。この場合、PCはPSS 4000-PCとも呼ばれます。PSS 4000ファームウェアがインストールされると、PCはシステムセクションOPCサーバUAや診断サーバなどを取得します（「[使用可能なシステムセクション](#)」 23]を参照）。

2.1.3.5 PSS 4000対応装置

PSS 4000ファームウェアをインストールするには、PSS 4000対応装置が特定のシステム要件を満たしている必要があります。これらのシステム要件は、PSS 4000対応装置にインストールされるPSS 4000ファームウェアによって定義されます。

PSS 4000 OPCサーバの例:

- ▶ PSS 4000 OPCサーバは、Microsoft Windowsオペレーティングシステムを搭載したPCにインストールできます。PCにインストールするためのシステム要件については、「[Microsoft Windowsオペレーティングシステム搭載PC](#)」 543]を参照してください。

2.1.4 PAS4000

設計、コンフィグレーション、プログラミング、試運転および診断のすべてを、PAS4000 (Pilz Automation Suite 4000) で集中して行うことができます。

PAS4000は、プロジェクトの作成および編集に使用可能な次のさまざまなツールで構成されています。

- ▶ PSSu Module Editor

PSSu Module Editorでは、必要なPSSuシステムを構築およびコンフィグレーションします。

- ▶ I/O Mapping Editor

I/O Mapping Editorでは、ユーザプログラムとハードウェア間の接続を確立します。

- ▶ IL Editor

IL Editorでは、IL (Instruction List: 命令リスト) プログラミング言語でユーザプログラムをプログラミングします。

2.1.5 診断

機械のダウンタイムを最小限に抑えるためには、効果的な診断が必要です。診断により、エラーを素早く回復し、エラーを防止し、機械の状態を分析するのに必要なすべての情報が提供される必要があります。

PSS 4000では、システム診断とプロセス診断が区別されています。

▶ システム診断

ハードウェアのコンフィグレーションやユーザプログラムのエラー検出を含む、ハードウェアおよびファームウェアの診断です。

すべてのシステム診断がピルツによって提供されます。ユーザにできる調節は、ロケーション情報に付加情報を入力するなど、ごくわずかしかなかったりません。

▶ プロセス診断

手順と状態の診断。プロセス診断は、ユーザがユーザプログラム内に作成します。ピルツのブロックにはプロセス診断が標準装備されていて、ユーザは各自の要件を満たすように変更することができます。

プロジェクトの診断データを表示したい場合、診断サーバが必要です（「[診断サーバ](#)」 [584](#)」を参照）。プロジェクトには、診断サーバを1つだけ含むことができます。

プロジェクトにOPCサーバ（PSS 4000 OPCサーバ）が含まれる場合、OPCサーバがインストールされているPSS 4000装置が診断サーバになります。OPCサーバがない場合、診断リストまたは診断ログがPAS4000に表示されると、PAS4000は独自の診断サーバを起動します。

2.1.6 OPCサーバ

OPCは標準化されたソフトウェアインタフェースであるため、幅広い製造元のPCベースソフトウェアでデータを交換することができます。このOPCによる基盤は、規格の維持と展開に大きな役割を果たします。

ネットワーク化された設備において、OPCサーバは通常、診断データやプロセスデータなどを収集し、定義されたネームスペースでこれらを表示します。OPCクライアントが、OPCサーバで提供されるデータにアクセスして処理をし、処理の視覚化／オペレーティングデータの記録を実現します。

PSS 4000オートメーションシステムでは、PSS 4000ファームウェアのインストール中にPCまたはPMIオペレータ端末がOPCサーバシステムセクションを取得します。PSS 4000ファームウェアをインストールすることで、PCは「サードパーティ製造業者のPSS 4000対応装置」になり、PMIオペレータ端末PMIは「ピルツのPSS 4000対応装置」になります（「[システムの概要](#)」 20」を参照）。

オートメーションシステムPSS 4000は、OPC規格「OPCクラシック」と「OPC UA」の両方をサポートしています。

2.2 PSS 4000のユニット特性

2.2.1 ネットワークシステムの集中表示

PSS 4000は高度に自動化された集中型コントローラの利点と小型の分散型制御ユニットの利点を兼備しています。

▶ 自動化レベルを高めることによる利点:

- 中央から使用可能なソフトウェアツールPAS4000
- 分散型PSS 4000専用装置を含む一元的なコンフィグレーション
- 一元的に作成されたユーザプログラムを複数のPSS 4000専用装置に分配できる
- トラブルシューティング、試運転／再試運転、変更中に工場全体の概要を迅速に把握できる

▶ 自動化レベルを抑えることによる利点:

- 分散型機能ユニットを独立して試運転できる
- 可用性の向上 (「単一障害点」がない)

2.2.2 装置の拡張性

「拡張性」とは、要件の変更に対してある程度の適応力も持つ装置に対して使用する用語です。

「水平」拡張性と「垂直」拡張性を次のように区別しています。

▶ 水平拡張

機能を他の装置または追加の装置に分散します。たとえばPSS 4000では、水平拡張を使用して次のことが可能です。

- 制御機能をネットワーク化された設備のさまざまなPSS 4000専用装置に分散して、処理能力を向上させる
- ユーザプログラムを独立したサブファンクションに分割して、プロジェクトのPSS 4000専用装置に分散する
- プロジェクトを再使用可能な部分に分割する

▶ 垂直拡張

デバイスまたはデバイスの部品がより強力な代替装置に置き換えられます。たとえばPSS 4000では、垂直拡張を使用して次のことが可能です。

- 異なるパフォーマンスクラスのヘッドモジュールを使用して、PSSuシステムのパフォーマンスクラスを遡及的に適応する
- I/Oモジュールを搭載したモジュラ式PSS 4000専用装置を遡及的に拡張および適応させる

2.2.3 プログラミング内でのモジュール化

PSS 4000では、制御タスクを工場全体に分散できます。その結果、設備アーキテクチャをタスクの割り付けに沿ってモジュール化できます。モジュール化は、PSS 4000専用装置とネットワークのモジュラ式構造だけでなく、ユーザプログラムにも適用されます。コンポーネント指向のプログラミングでは、完全に設備を機能に分割できます。最大の柔軟性と適応性を維持しながら、標準化と再利用性をできるだけ高いレベルで実現することを目的としています。

モジュール化されたユーザプログラムを使用して、PSS 4000で次のようなことを実現できます。

- ▶ 変更がローカルにしか影響しない
- ▶ 制御タスクを標準化できる

ユーザプログラムを同じ構造を持つすべての機能に使用できます。変更が加えられると、関連するすべての装置に転送できます。

- ▶ 可用性が向上する (エラーの反応がローカルに限定されるため)

3 安全

3.1 その他の参照資料

使用している製品の取扱説明書と『安全マニュアルPSS 4000』で安全ガイドラインを参照してください。安全マニュアルには、安全関連の計画、設備の建設、運用に役立つチェックリストもあります。

また提供されている次の情報も参照してください。

- ▶ PSSuniversalの『取り付けマニュアル』
- ▶ PAS4000のオンラインヘルプ

印刷の現行規格が適用されます。

3.2 用途

すべてのパフォーマンスクラスのPSS 4000専用装置は、次の産業環境での使用を目的として設計されています。

- ▶ パフォーマンスクラスPSS 4000 PLCまたはPSS 4000 multiの制御システムのFSリソースを使用する安全関連アプリケーション (分散型タスク用のSafetyNET pへの接続を含む)
- ▶ パフォーマンスクラスPSS 4000 PLCの制御システムのSTリソースを使用する非安全関連アプリケーション (分散型タスク用のSafetyNET pや他の一般バスシステムへの接続を含む)

すべてのパフォーマンスクラスのPSS 4000専用装置は、本質的には、電源を切り離すことによって安全な状態になる機械安全回路で使用するのに適しています。

例:

- ▶ プレス
- ▶ トランスファライン
- ▶ 貯蔵タンク施設
- ▶ 非常停止機能
- ▶ 燃焼制御システム
- ▶ ケーブルカー／牽引操作
- ▶ 舞台技術

次のような使用は、明らかに不適切であるとみなされます。

- ▶ 製品部品の技術的または電氣的改造
- ▶ 取扱説明書で説明している分野以外での製品の使用
- ▶ 記載された技術の詳細情報に適合していない使用

予見できる誤用と不正使用

- ▶ 追加対策を施さない場合、オートメーションシステムPSS 4000は環境要件が過酷な箇所 (爆発危険箇所) で使用するのに適していません。
- ▶ オートメーションシステムPSS 4000は、電源切断時に安全な状態にならない設備には適していません。
- ▶ 旅客輸送システムの場合は、故障によって設備が停止した場合に備えて、適切な脱出手段を講じるか、自蔵式の救難具を備える必要があります。
- ▶ 電気式の解除装置を備えたロックゾーンの場合は、緊急解除装置を備える必要があります。
- ▶ 同じレベルの安全性を提供する適切な手段 (ミュート機能など) を講じるまで、安全装置を無効にしてはいけません。同じレベルの手段は、前の安全アセスメントで許可された場合のみ使用できます。

例:

- 動作モードセレクトスイッチによってトリガされるミュート機能がある場合には、セットアップモードで安全ゲートが開くことがあります。
- 対応するコンポーネントのミュート機能 (たとえば、制御を含むミュートセンサ、ユーザプログラム内のミュートブロックを使用するミュートセンサのミュート監視など) がある場合には、転送された資材によってライトカーテンを中断することができます。

3.2.1 安全度水準 (SIL) およびパフォーマンスレベル (PL)

すべてのパフォーマンスクラスのPSS 4000専用装置は、機械指令2006/42/ECの付属書Vに適合した安全コンポーネントです。アプリケーション領域とその規制によっては、すべてのパフォーマンスクラスのPSS 4000専用装置は、EN 62061のSIL 3およびEN ISO13849-1のPL e (Cat. 4) まで適合します。リスクアセスメントの関連技術データについては、それぞれの取扱説明書を参照してください。

SafetyNETpは、ネットワーク化されたオートメーション機能に使用します。この安全通信は、EN IEC 61508など、関連する規格に適合して開発されました。BGなどの独立認証機関が、EN ISO 13849-1のPL e (Cat. 4) またはEN IEC 62061のSIL 3までの安全関連アプリケーションでSafetyNET p概念のセキュリティおよび適合性を確認しました。モジュールのリスクアセスメントに必要な関連技術データについては、それぞれのモジュールの取扱説明書を参照してください。

3.2.2 一般的な安全評価

PSS 4000専用装置を使用する前に、機械指令に従って安全アセスメントを実施する必要があります。個別のコンポーネントとしてのPSS 4000専用装置は、EN/IEC 61508の目的のための安全関連システムです。たとえば、ハードウェアやファームウェアのエラーに対する機能の安全性を保証します。ただし、プロセスおよび設計全体、またはプロジェクトの安全性は保証しません。

プロジェクトの安全性は、ユーザの責任の下で確保するものとします。プログラミング時には、特段の注意を払い、規格と規制に従ってください。

問題のあるユーザプログラムにより、プロセス全体の安全性が損なわれることがあります。

耐用期間のすべての段階および安全ライフサイクル全体での設備全体の安全要件を定義してください。さらに、技術的および組織的見地から、これらをどのように実装するかも定義してください。

3.3 有資格者の採用

製品の組み立て、取り付け、プログラミング、試運転、運転、メンテナンス、取り外しを行うことができるのは、有資格者に限ります。

有資格者とは、トレーニング、経験、現職での活動により、資格のある知識豊富な人材を指し、必要な専門知識を有します。装置、システム、機械の検査、評価および運転を可能にするため、有資格者は最新技術だけでなく、国内、欧州、および国際的に適用される法律、指令、規格に通じている必要があります。

企業は、次の条件を満たす作業者にのみ業務を担当させる責任があります。

- ▶ 安全衛生および事故防止の基本的な規則に習熟している
- ▶ 「安全」の章にある情報を読んで内容を理解している
- ▶ 特定のアプリケーションに適用する包括規格および専門的な規格について優れた知識を有している

3.4 安全入出力

FSアプリケーションでは、短絡および断線によって設備／機械内で危険な状況が発生しないようにする必要があります。

各種テストをFS入出力モジュール上でコンフィグレーションすることができます。これらのテストを使用して、多くの短絡および断線を検出できます。テストの例を以下に示します。

- ▶ オンになった出力のオフテスト
- ▶ オフになった出力のオンテスト
- ▶ 出力間の短絡テスト
- ▶ 入力の接点間の短絡検出用テストパルス

配線のガイドラインと接続例については、入出力モジュールの取扱説明書またはデータシートで確認できます。

4 セキュリティ

プラント、システム、機械、ネットワークをサイバー脅威から保護するには、最先端の総合的な産業セキュリティコンセプトを実装し、継続的に維持する必要があります。

VDI/VDE 2182またはIEC 62443-3-2に適合するリスクアセスメントを実施し、細心の注意を払ってセキュリティ対策を計画してください。必要に応じて、ピルツカスタマーサポートにお問い合わせください。

詳細については、該当する製品の取扱説明書を参照してください。

PSS 4000ファームウェアを搭載したPCおよびオペレーターミナルPMIについては、「[ネットワークデータ \(OPCサーバ \(クラシック\)\)](#)」 [533](#)」および「[ネットワークデータ \(OPCサーバUA\)](#)」 [564](#)」を参照してください。

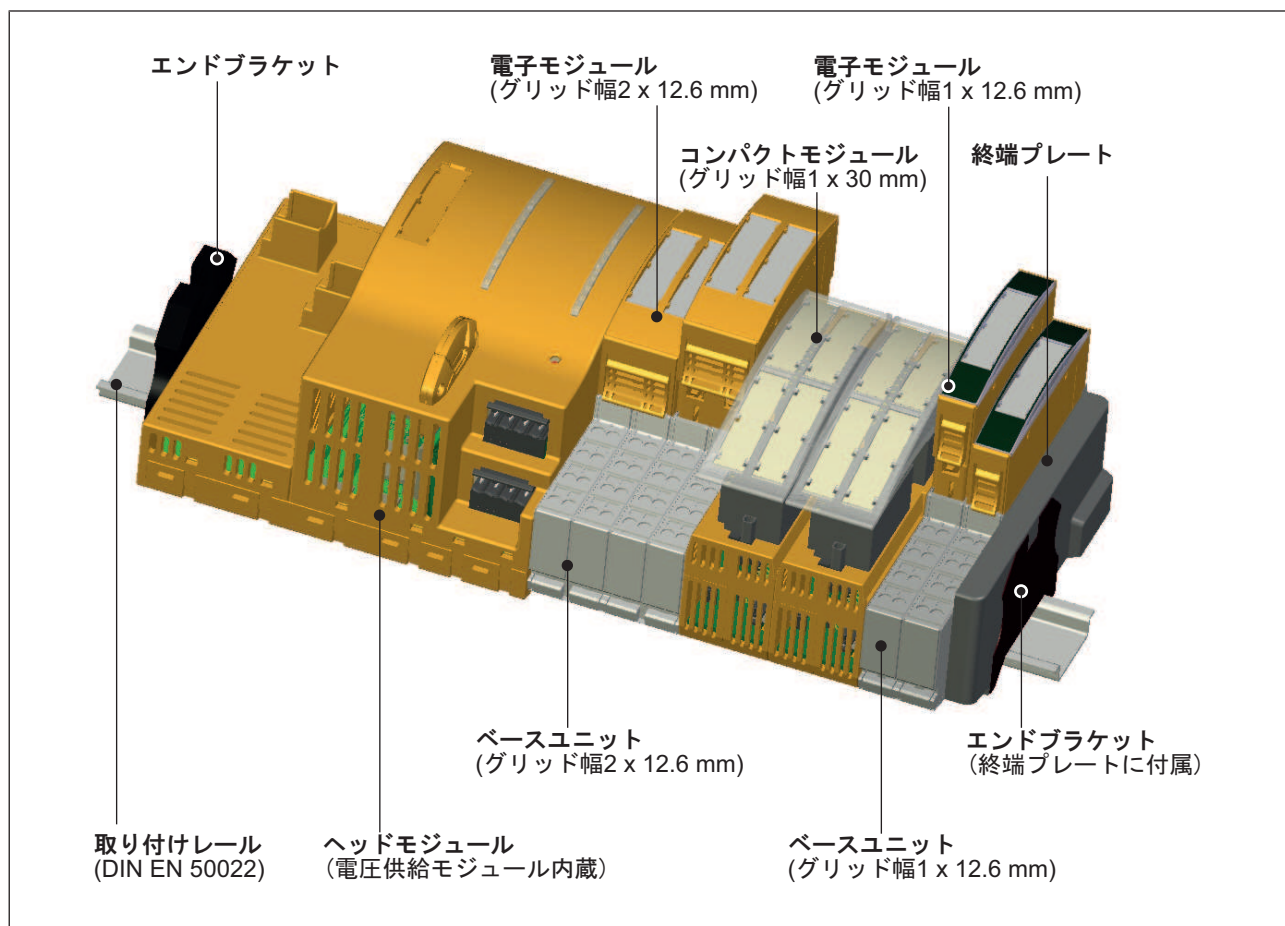
5 PSSuシステムの基本情報

PSSuシステムは、PSS 4000専用装置です。PSSuシステムは、モジュール設計のPSS 4000専用装置です。
装置で利用できるシステムセクションは、装置によって異なります (「[システムセクション](#)」 22]」を参照)。
詳細は、関連する装置の取扱説明書に記載しています。

5.1 構造

PSSuシステムは、次のコンポーネントで構成されています。

- ▶ 電源が内蔵されたヘッドモジュール
- ▶ 一般アプリケーションおよび安全アプリケーション用の入出力モジュール
- ▶ 電圧供給モジュール (必要な場合のみ)
- ▶ 始端部でシステムを固定するためのエンドブラケット
- ▶ バス終端抵抗と、システムを終端部で固定するためのエンドブラケットを備えた終端プレート



ヘッドモジュール

ヘッドモジュールは、PSSuシステムのデータトラフィック全体を調整し、パフォーマンスクラスを判断します。

パフォーマンスクラスは、装置の基本的な機能を決定します。パフォーマンスクラスは次のように分類されます。

- ▶ PSS 4000 I/O (略称: I/O)

制御機能を備えていないPSS 4000専用装置

パフォーマンスクラス「分散型システムPSSu I/O」が文書化されているPSSuシステムがあります。PSS 4000の視点では、これはパフォーマンスクラスPSS 4000 I/Oに対応します。

▶ PSS 4000 multi (略称: multi)

コンパクト安全制御システムの機能を備えたPSS 4000専用装置

パフォーマンスクラス「制御システムPSS multi」が文書化されたPSSuシステムがあります。PSS 4000の視点では、これはパフォーマンスクラスPSS 4000 multiに対応します。

▶ PSS 4000 PLC (略称: I/O)

オートメーションシステムを制御するためのPSS 4000 CPU

パフォーマンスクラス「制御システムPSSu PLC」が文書化されているPSSuシステムがあります。PSS 4000の視点では、これはパフォーマンスクラスPSS 4000 PLCに対応します。

パフォーマンスクラスに応じたPSSuシステムの機能:

▶ パフォーマンスクラスPSS 4000 I/OのPSSuシステム

- ヘッドモジュール、電子モジュールおよびコンパクトモジュール (またはそのいずれか) で構成
- パフォーマンスクラスPSS 4000 PLCまたはPSS 4000 multiの制御システムによるSafetyNET p経由のI/Oの制御

▶ パフォーマンスクラスPSS 4000 multiのPSSuシステム

- ヘッドモジュール、電子モジュールおよびコンパクトモジュール (またはそのいずれか) で構成
- FSリソースを使用できる
- 1つのタスクのみコンフィグレーションできる
- PAS 4000 Multiエディタでプログラミング

▶ パフォーマンスクラスPSS 4000 PLCのPSSuシステム

- ヘッドモジュール、電子モジュールおよびコンパクトモジュール (またはそのいずれか) で構成。ただし、SafetyNET pを使用して分散型I/Oを制御するためだけにPSSuシステムを使用する場合、電子モジュール／コンパクトモジュールは不要
- SafetyNET pを使用してネットワーク接続している場合、すべてのパフォーマンスクラスの分散型PSS 4000システムでのI/Oの制御
- STリソースとFSリソースを使用できる
- リソースでは最大9個のタスクをコンフィグレーションできる
- PAS4000 Multiエディタでプログラミングまたは、PAS4000でIEC 61131に従ってプログラミング

入出力モジュール

次の2種類の入出力モジュールがあります。

- ▶ 電子モジュールとベースユニット
- ▶ コンパクトモジュール

入出力モジュールは、さまざまな入出力機能で使用できます。

電圧供給モジュール

ヘッドモジュールは、残りのモジュールに供給電圧を提供します。追加の電源が必要になることもあります (例: 電位分離、電圧リフレッシュ)。

電圧供給モジュールは、電子モジュールとベースユニットから構成されています。

設計

電子モジュールとベースユニット:

電子モジュールは、電圧供給モジュールまたは入出力モジュールの機能を決定します。

▶ 電子モジュール

- ベースユニットに取り付け
- モジュールバス経由でヘッドモジュールと通信

ベースユニットは電子モジュールの接続装置です。

▶ ベースユニット

- 現場配線の接続に使用
- 次の接続タイプで使用:
 ケージ式端子またはスクリー式端子

コンパクトモジュール:

コンパクトモジュールには、機能ユニット (入力と出力、またはそのいずれか) と接続レベルが1つのハウジングに収められています。配線はスプリング式端子のある多ピンコネクタ経由で行います。このコネクタは、モジュールのコネクタストリップに取り付けます。

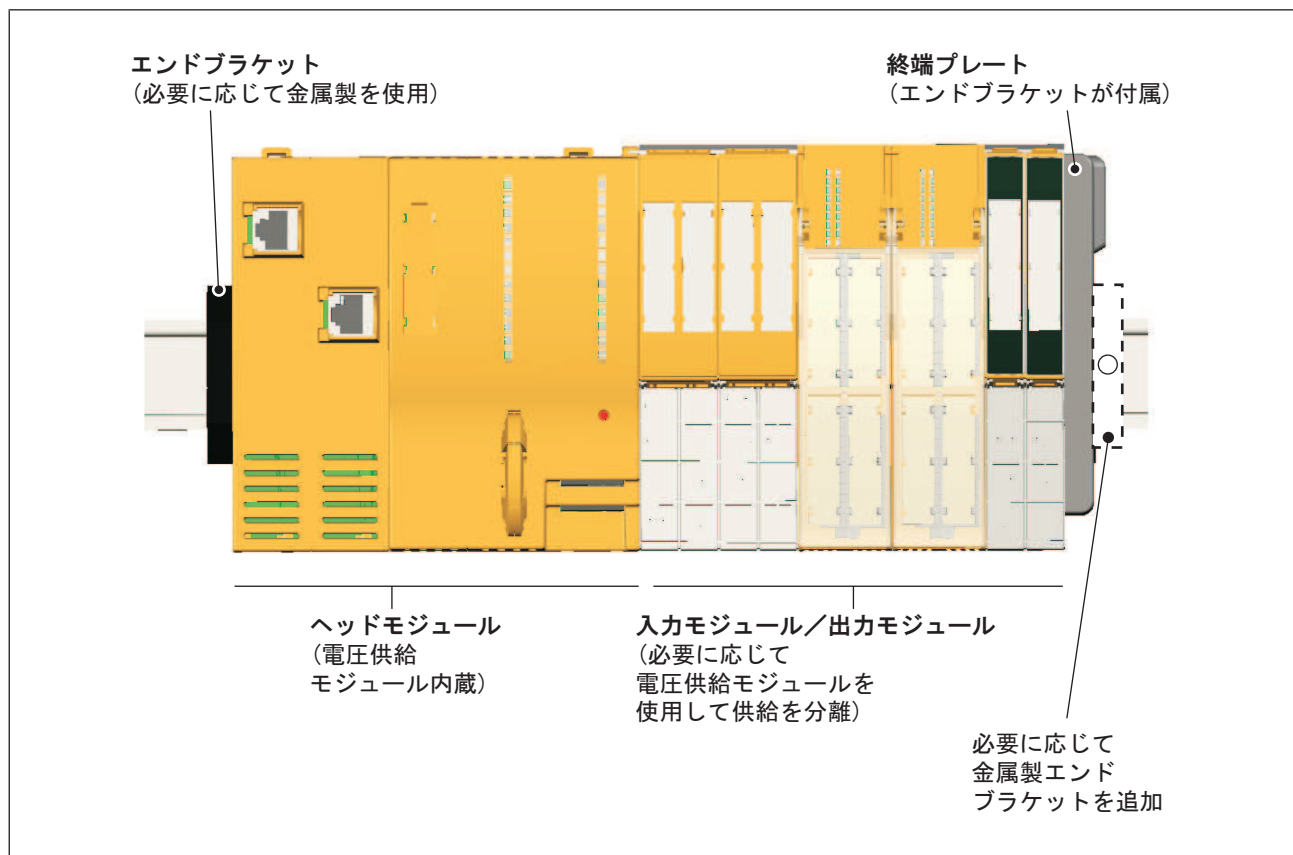
コンパクトモジュール

- ▶ ベースユニットは不要
- ▶ 現場配線の接続に使用
- ▶ モジュールバス経由でヘッドモジュールと通信

5.2 モジュールのレイアウト

- ▶ PSSuシステムの最初のモジュールはヘッドモジュールでなければなりません。電圧供給モジュールはヘッドモジュール内に内蔵されています。
 - モジュール供給電源(ヘッドモジュールと入出力モジュールへの供給)
 - 周辺電源 (センサおよびアクチュエータへの供給)
- ▶ FSおよびSTアプリケーションの入出力モジュールを右側に取り付けることができます。
 - 入出力モジュールは任意の順番で取り付けことができます。
 - FSアプリケーションおよびSTアプリケーションの入出力モジュールは必要に応じて組み合わせることができます。
 - 同じ機能を持つモジュールを機能別のグループに分けると、接続が分かりやすくなり、配線が容易になります。
 - スクリュー式端子のベースユニットとケージ式端子のベースユニットを1つのPSSuシステムに混在させることはできません。
 - スクリュー式端子のベースユニットは、コンパクトモジュールの左側に取り付けてはいけません。
 - 入出力モジュールの最大数は、システム制限によって決まります。
 - FSアプリケーションの入出力モジュールは、STアプリケーション用に専用設計されたヘッドモジュールとともに使用してはいけません。
- ▶ モジュール供給または周辺供給をリフレッシュするのに追加の電圧供給モジュールが必要になることがあります。
- ▶ 供給グループでは、各グループの開始位置に追加の電圧供給モジュールを取り付ける必要があります。供給グループに属しているモジュールをその右側に取り付けます。
- ▶ PSSuシステムの最終部品は、必ずモジュールバスの終端抵抗が含まれた終端プレートでなければなりません。
- ▶ システムは、システムの始端および終端で固定具を使用して取り付けレールに取り付けます。次のエンドブラケットが用意されていて、振動や衝撃に応じて選択することができます。
 - プラスチック製 (標準)
 - 金属製 (振動や衝撃が大きい場合)

固定具のレイアウト:



5.3 モジュールの型番

モジュールの型番は、機能に関する情報を提供します。型番は、複数の段階の文字と数字の組み合わせで構成されています。すべてのモジュール名はPSSuで始まります。この後に、次の指定が続きます。

▶ **H:** ヘッドモジュール

例: PSSu **H** PLC1 FS DP SN SD

▶ **E:** 電子モジュール:

– **S:** 一般モジュール

例: PSSu **E S** 4DI

– **F:** 安全モジュール

例: PSSu **E F** 4DI

▶ **K:** コンパクトモジュール:

– **S:** 一般モジュール

例: PSSu **K S** 16DI

– **F:** 安全モジュール

例: PSSu **K F** 16DI

▶ **B:** ベースユニット

例: PSSu **BP** 1/8S

ベースユニット	基本機能	サイズ	接続タイプ	追加機能
PSSu B...	PSSu BP... – 周辺機器用ベースユニット (入出力)	PSSu BP-C 1/8... – 1 x グリッド幅、8接続	PSSu BP-C 1/8C – ケージ式端子	PSSu BP 1/8S-J – 統合冷接点補償
	PSSu BP-C... – Cレール付きの周辺機器用ベースユニット	PSSu BP-C 1/12... – 1 x グリッド幅、12接続	PSSu BP-C 1/8S – スクリュー式端子	
	PSSu BS... – 電源用ベースユニット	PSSu BP-C 2/16... – 2 x グリッド幅、16接続		
	PSSu BS-R... – 電源リフレッシュ用ベースユニット	PSSu BP-C 2/8... – 2 x グリッド幅、8接続		

リンクモジュールには、アプリケーションに応じた個別の文字コードが含まれます。例:

▶ PSSu **XB** F-T、PSSu **XR** F-T

(ケーブルベースの下位区分)

PSSuシステムのアクセサリは、**PSSu A...**で識別されます(A:「Accessory」)。

PSSuモジュールには、次のような異なる製品タイプがあります。

▶ **基本タイプ**

PSSuモジュールの基本バージョンは、基本タイプモジュールと呼ばれます (例: PSSu **E F** DI OZ 2)。

▶ Tタイプ

温度と湿度の厳しい環境要件に適合するPSSuモジュールは、Tタイプモジュールと呼ばれます。基本タイプモジュールと比較して、機能性の違いはありません。これらは製品名の最後にある「-T」で識別されます。例: PSSu E F DI OZ 2-T

▶ Rタイプ

鉄道関連の用途に特化して設計され、鉄道関連の認証を受けているPSSuモジュールは、Rタイプモジュールと呼ばれます。これらは製品名にある「-R」で識別されます (例: PSSu E F DI OZ 2-R)。

5.4 モジュールバス (MBUS)

モジュールバスは、PSSuシステムのモジュールに電圧を供給し、ヘッドモジュールと電子／コンパクトモジュール間でデータを転送します。モジュールバスは、ベースユニット／コンパクトモジュールを組み合わせて、機械式ラッチで結合することで形成されます。

モジュールバスの構造は次の通りです。

▶ データバス

安全データ (FSモジュールバス) および一般データ (STモジュールバス) 用の2つの個別のバスシステム。

▶ 次のための供給電圧:

– モジュール供給

モジュール供給は、ヘッドモジュールおよび電子モジュール用の内部供給電圧です。

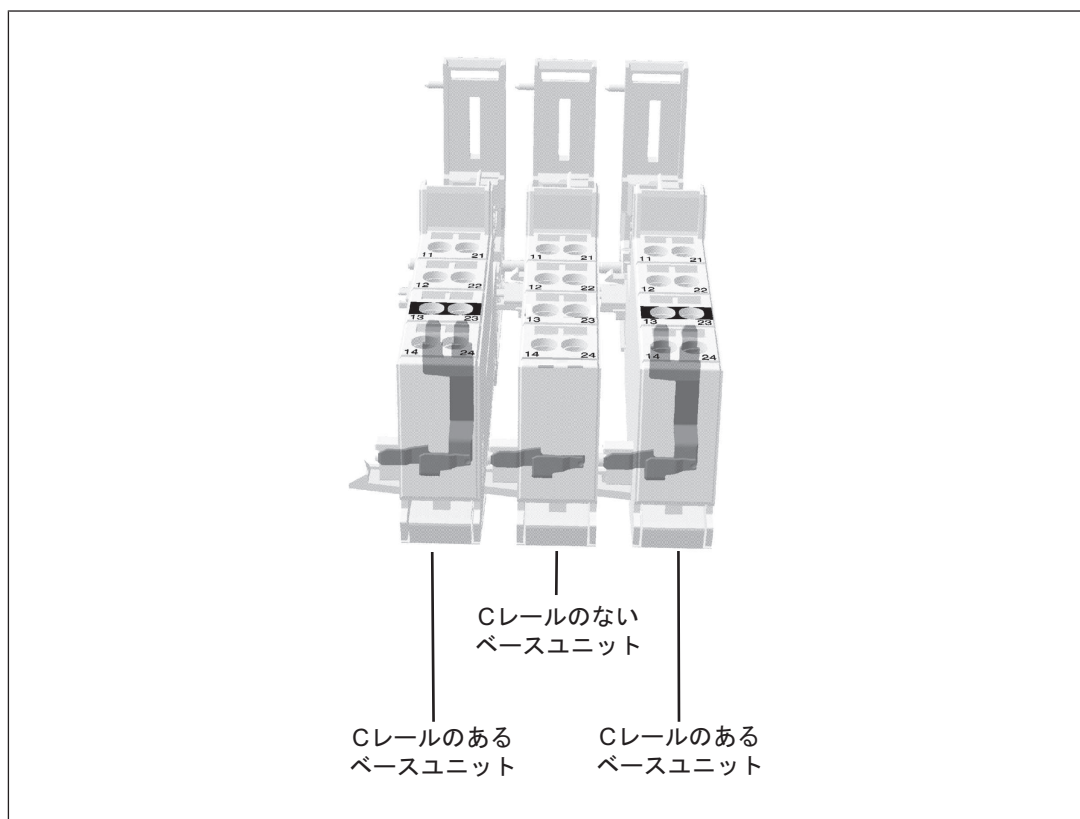
– 周辺供給

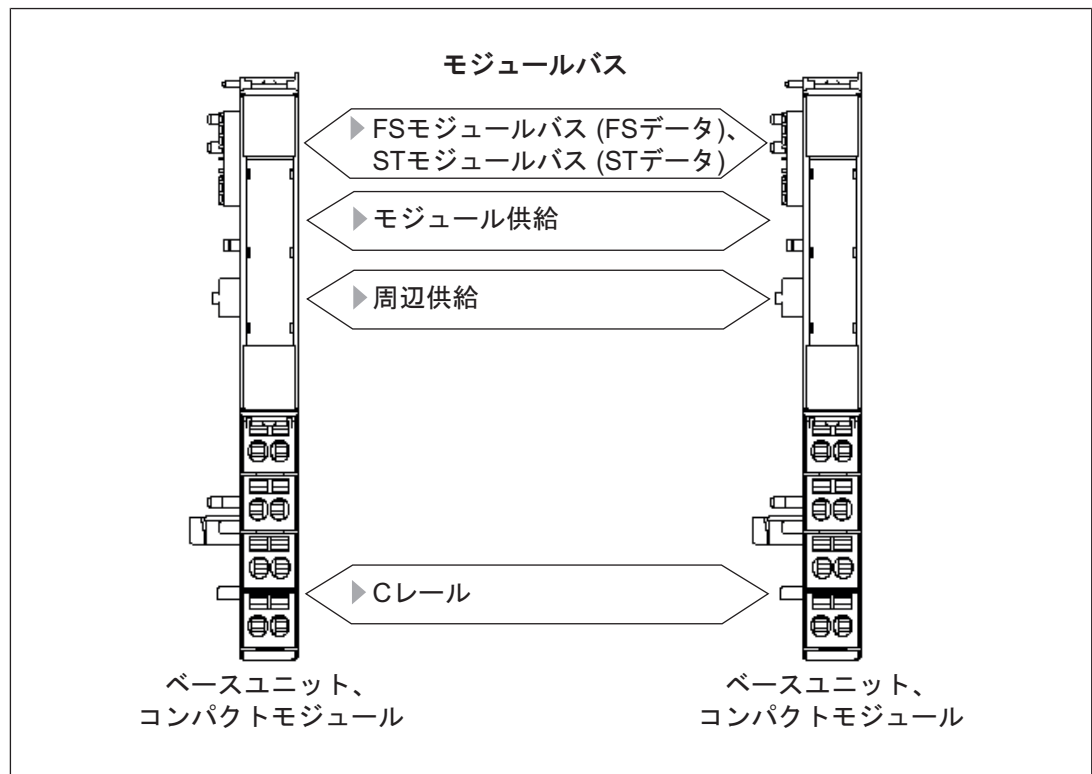
周辺供給は、電子モジュールのデジタル入出力のセンサおよびアクチュエータに24 VDCを供給します。コンパクトモジュールの場合、入力用の供給電圧は周辺供給から生成されます。出力の供給電圧は、モジュールの端子に供給される必要があります。モジュールバスの接続を切断して、供給グループを形成することができます。

▶ Cレール (「クロス接続」)

Cレールはオプションのフリー電源レールです。Cレールに対しては、説明に文字「-C」が含まれているベースユニット (例: PSSu BP-C 1/8 S, PSSu BP-C 1/8 C) を通じてアクセスできます。

Cレールの電圧供給は、他のベースユニット接続と同じ回路に属する必要があります。PEをCレールに接続することで、シンプルな形式でシールドを得ることができます。コンパクトモジュールの場合、Cレールはループスルーされず、端子では使用できません。





5.5 ベースユニット上の接続

ベースユニット上の接続は接続レベルと接続コラムで構成されています。接続には、2桁の数字が付けられています。

- ▶ 最初の桁は、ベースユニットの接続コラムを表しています (例: 接続23は2番目のコラムにあります)。
- ▶ 2桁目は、ベースユニットの接続レベルを表しています (例: 接続23は3番目のレベルにあります)。

ベースユニットの接続の機能は、電子モジュールによって異なります。

5.5.1 入出力モジュールの接続レベル

入出力モジュールで、接続レベルは、通常、次のように設定されています。

▶ 接続レベル1および4

入力および出力の接続

▶ 接続レベル2および3

共通供給用の接続 (周辺供給、アナログ入出力、またはシールド)

次の2つの接続レベルがある入出力モジュールも数多くあります。

▶ 接続レベル5および6

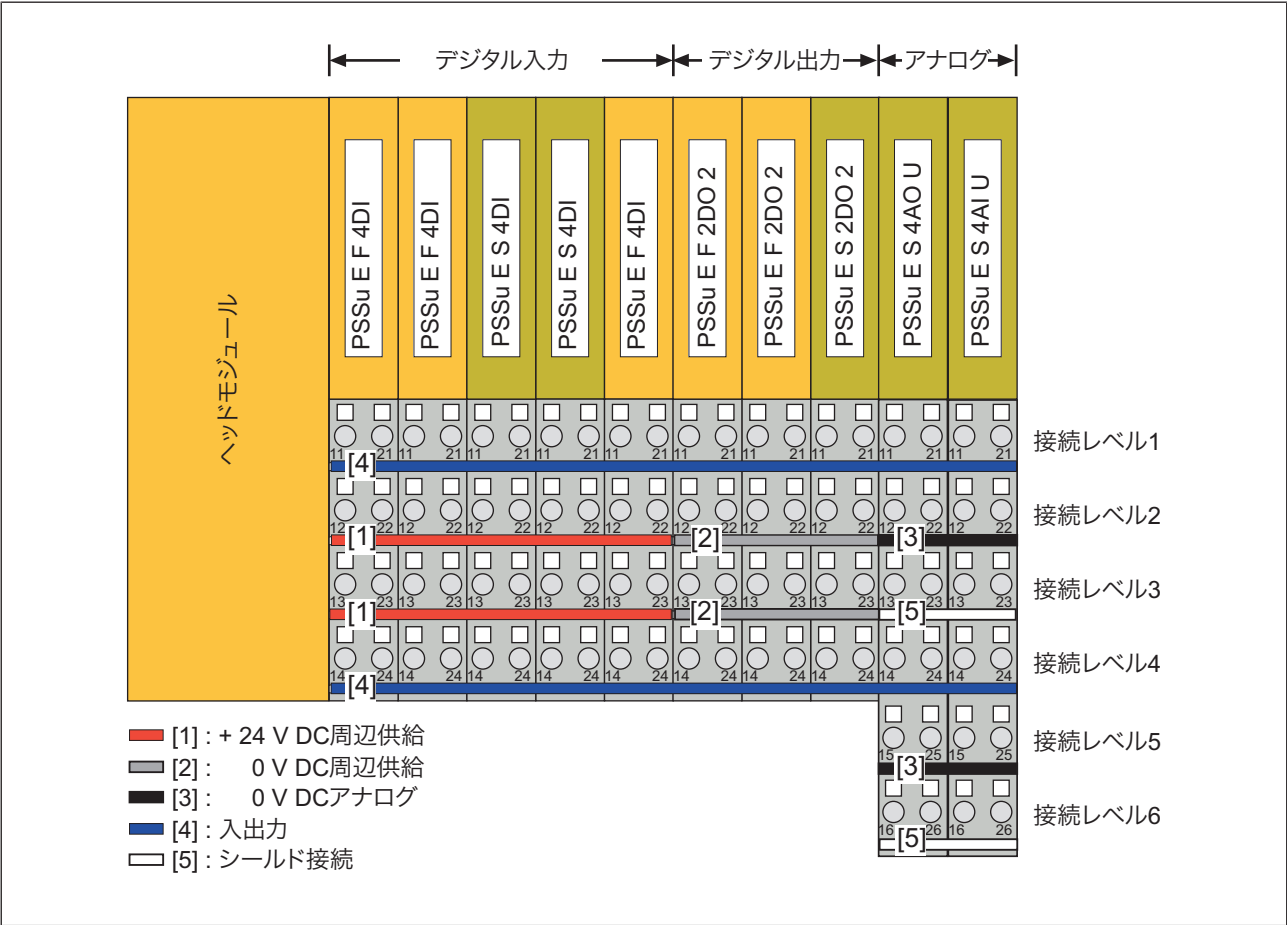
共通供給用の接続 (周辺供給、アナログ入出力、またはシールド)

入出力モジュールは任意の順番で取り付けることができます。ただし、接続図を一貫性のあるものにするためにも、同じタイプの入出力モジュールをグループに分類して、コンパクトモジュールが間に入らないようにすることをお勧めします (例を参照)。

5.5.2 接続レベルの色分け

接続レベルの色分けは、配線作業を容易にするためのものです。ラベル付け用に、さまざまなカラーマーカーを用意しています。カラーマーカーを使用して、各種供給にラベル付けすることができます。これにより、配線後も接続レベルをはっきりと目視確認できるようになります。

例:



5.6 コンパクトモジュール上の接続

コンパクトモジュールの接続は端子列に分かれています。ケージ式端子のある1列または3列コネクタは、装置の1列10ピンコネクタストリップに差し込むことができます。

接続の機能は、コンパクトモジュールによって異なります。

▶ 端子列1

入出力、周辺供給の接続 (24 Vおよび0 V)。

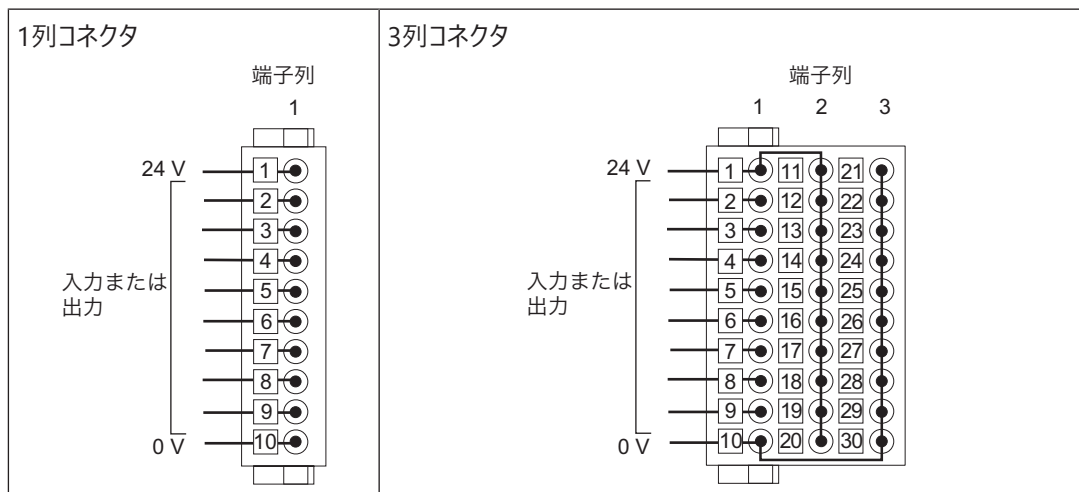
▶ 端子列2

すべての接続がリンクされていて、最初の接続レベルの24 V周辺供給接続にブリッジされています。

▶ 端子列3

すべての接続がリンクされていて、最初の接続レベルの0 V周辺供給接続にブリッジされています。

入出力モジュールは任意の順番で取り付けことができます。ただし、接続図を一貫性のあるものにするためにも、同じタイプの入出力モジュールをグループに分類することをお勧めします。



5.7 PSSuシステムの増設の制限

PAS4000のPSSuモジュールエディタは、PSSuシステムの構築に関するサポートを提供します。このツールは、データが入力されるのに合わせてシステム制限をチェックします。

PSSuシステムの増設は次の値によって制限されます。

- ▶ モジュール数
- ▶ FSモジュール数
- ▶ STモジュール数
- ▶ 安全入出力数
- ▶ 一般入出力数
- ▶ モジュールバスの電流負荷

システム制限については、以下で説明しています。

5.7.1 PSSuシステムでのモジュール数と入出力数

次のシステム制限が該当します。

モジュールタイプ	PSSuシステムあたりの最大数
- - -	合計64モジュール
STモジュール	64モジュール
FSモジュール	64モジュール
PSSu E F DIOZ 2	64モジュール
STビットモジュール	256 ST入力ビット 240 ST出力ビット
FSビットモジュール	256 FS入力ビット 256 FS出力ビット
STバイトモジュール	16モジュール
FSバイトモジュール	16モジュール
PSSu E S RS232	16モジュール
PSSu E S RS485	16モジュール
PSSu K S RS232	16モジュール
PSSu K S RS232 Modbus ASCII	16モジュール
PSSu K F FAU	2モジュール
PSSu K F EI	8モジュール 注意事項: タイプPSSu K F EI CVの複数のモジュールを同時に使用する場合、両方のモジュールタイプで構成されるモジュールを合計で8個まで使用できます。 ただし、タイプPSSu K F EI CVの上限である4モジュールは守る必要があります。 たとえば、次の組み合わせが許可されます。タイプPSSu K F EIモジュール6個とタイプPSSu K F EI CVモジュール2個。
PSSu K F EI CV	4モジュール
PSSu K F FCU	12モジュール

PSSuシステムには最大64個のモジュールを搭載できます。モジュール数をカウントする際には、次のことに注意してください。

- ▶ ヘッドモジュールとパッシブジャンクションモジュール (例: PSSu E PD、PSSu E PD1) はカウント対象外。
- ▶ PSSuシステムの次のモジュールはカウント対象。
 - 電圧供給モジュール
(ID: PSSu E F PS...)

- 一般モジュール
(ID: PSSu E S..., PSSu K S...)
- 安全モジュール
(ID: PSSu E F..., PSSu K F...)

5.7.2 最大電流負荷

5.7.2.1 電圧供給モジュールの最大電流負荷容量

システム供給	最大電流負荷
次を経由して供給されたモジュール供給 - PSSu E F PS(-T)	1.5 A
次を経由して供給されたモジュール供給 - PSSu E F PS1(-T)	2 A
次を経由して供給されたモジュール供給 - PSSu E F PS2(-T -R)	1.0 A
周辺供給	10 A
Cレール	10 A

▶ モジュール供給

モジュール供給の電流負荷は、電子モジュールとコンパクトモジュールによる総消費電流です。総消費電流が供給モジュールの電流負荷容量よりも高い場合は、追加の供給モジュールでモジュール供給をリフレッシュする必要があります (モジュールの「技術データ」を参照)。システムソフトウェアはこれを考慮します。

▶ 周辺供給

最大電流負荷:10 A

周辺供給の電流負荷は、入出力モジュールによって供給されたセンサとアクチュエータの消費電流の合計です。電流負荷が高い場合、過負荷を防止できるように追加の供給モジュールで周辺供給をリフレッシュする必要があります。デレーティング図を参照してください。

▶ Cレール

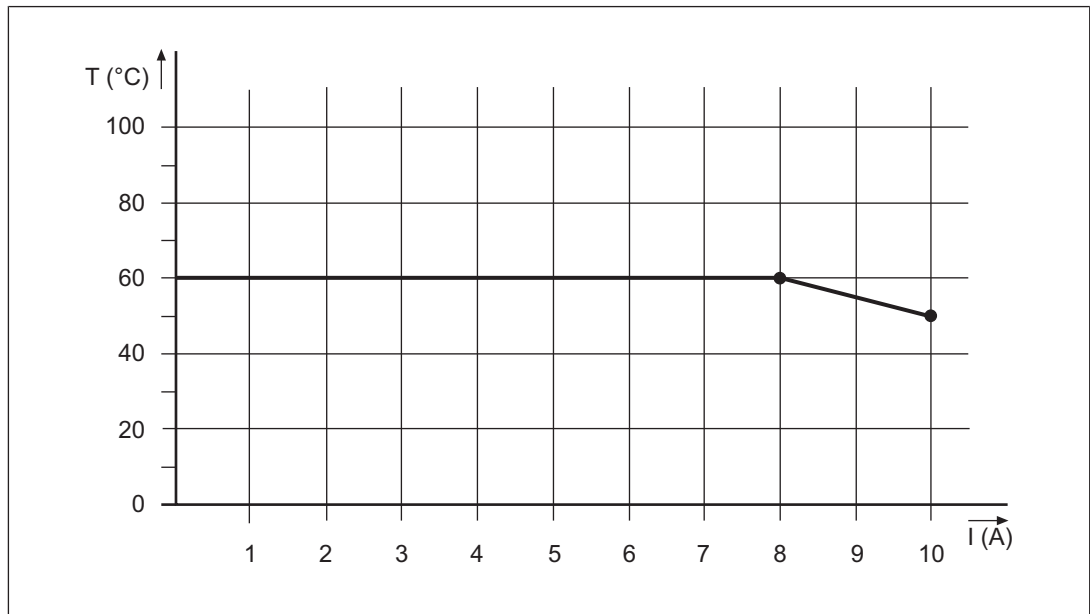
最大電流負荷:10 A

電流負荷が高い場合は、過負荷を防止するためにCレールは異なる供給を使用する必要があります。デレーティング図を参照してください。

5.7.2.2 温度に基づいた最大電流負荷容量

最大許容周囲温度以下の場合のみ、周辺供給およびCレールに関するモジュールバスの最大電流負荷容量を得ることができます。

周辺供給とCレールのデレーティング図: 温度Tと負荷電流Iの関係



Tタイプ: 周辺供給とCレールのデレーティング図: 温度Tと負荷電流Iの関係



5.8 PSSuシステムのスロット

スロットは、PSSuシステム上の電子モジュール、コンパクトモジュール、または電圧供給モジュールの位置を示します。

PSSuシステムの構造を確認するには、いくつかの方法があります。物理構造と論理構造は区別されています。したがって、物理スロットと論理スロットがあります。

物理スロット

スロットでは、物理スロット番号に対して次の規則が適用されます。

- ▶ カウントシーケンスは、ヘッドモジュールの後の最初のモジュールから始まる。
- ▶ カウントシーケンスは、スロット番号0から始まる。
- ▶ スロット番号では、モジュールは左から右に向かって昇順でカウントされる。

例

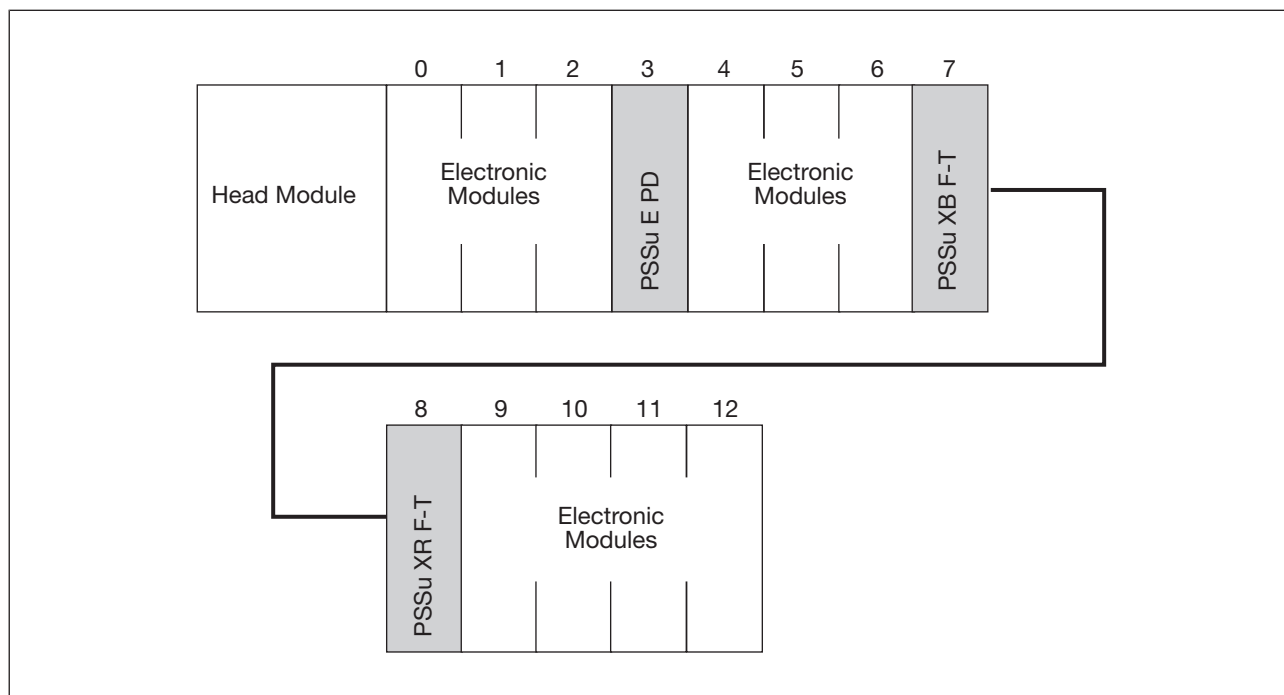


図: 物理スロット番号 (例)

論理スロット

論理スロット番号には、物理スロット番号と同じ規則が適用されますが、リンクモジュール (例: PSSu XB F-T) と電圧供給モジュール (例: PSSu E PD) はカウントされません。

例

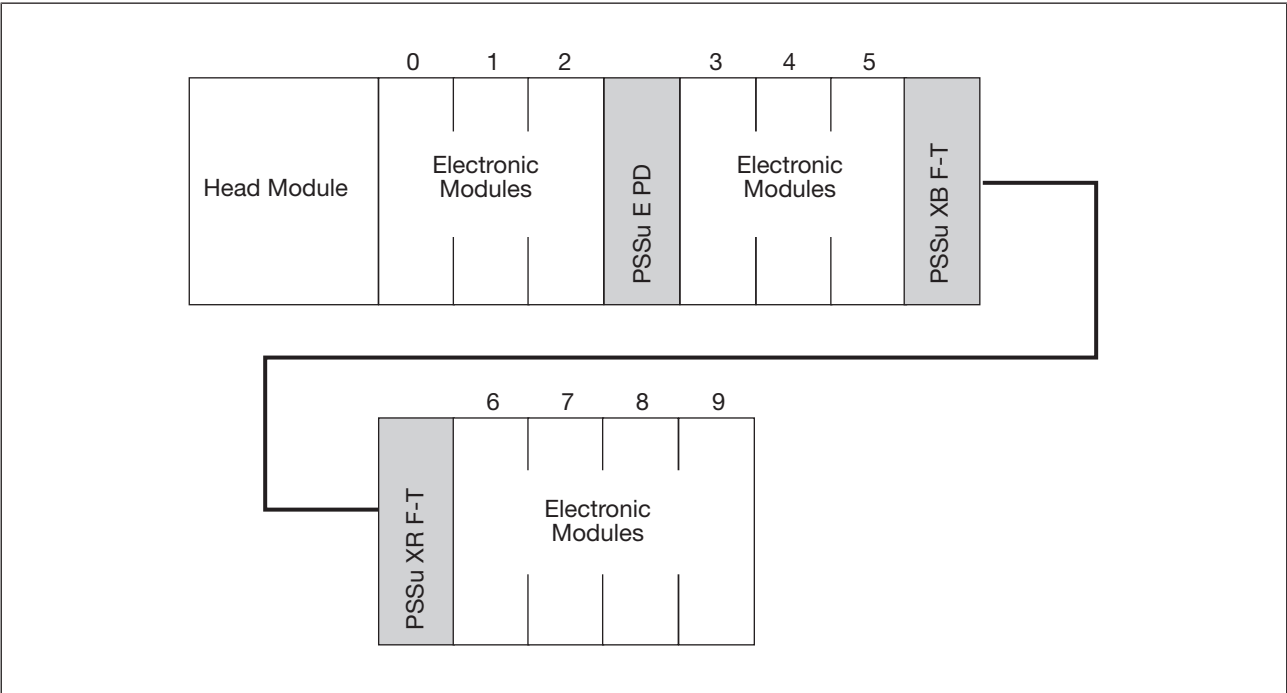


図: 論理スロット番号 (例)



情報

診断でスロット情報が提供される場合、この情報には論理スロット番号が含まれます。

5.8.1 PSSuシステムでの設定／実際のハードウェアレジストリ

PSSuシステム内の装置プロジェクトには、モジュールグループ内のモジュールのコンフィグレーションされた順序に関する詳細な情報が含まれます。

PSSuモジュールは、次のモジュールグループに分けられます。

- ▶ FSビットモジュール (例: PSSu E F 4DO 0.5)
- ▶ STビットモジュール (例: PSSu E S 2DO 2)
- ▶ FSバイトモジュール (例: PSSu E F AI U)
- ▶ STバイトモジュール (例: PSSu E S 2AI se)

PAS4000内のモジュールグループに対してコンフィグレーションされたモジュールの順序に関して、ローカルPSSuシステムにある実際のハードウェアレジストリが、設定のハードウェアレジストリと一致しない場合、コンフィグレーションエラーが発生します。



情報

電子モジュールには、2つの部品で構成されたコード化部品が付いています。最初にベースユニットに電子モジュールを挿し込んだ際、コード化部品の一部は電子モジュールに残り、もう一部はベースユニットに固定されます。これがベースユニットのコード化の仕様です。コード化部品は、電子モジュールの交換時などに、取り付けミスを防止できるように設計されています。電子モジュールは、同じ機械コードを使用したベースユニットにのみ挿入することができます。コードは、文字と色で識別されています。同じ文字でのコーディングは機械的に同じですが、色が異なることがあります。

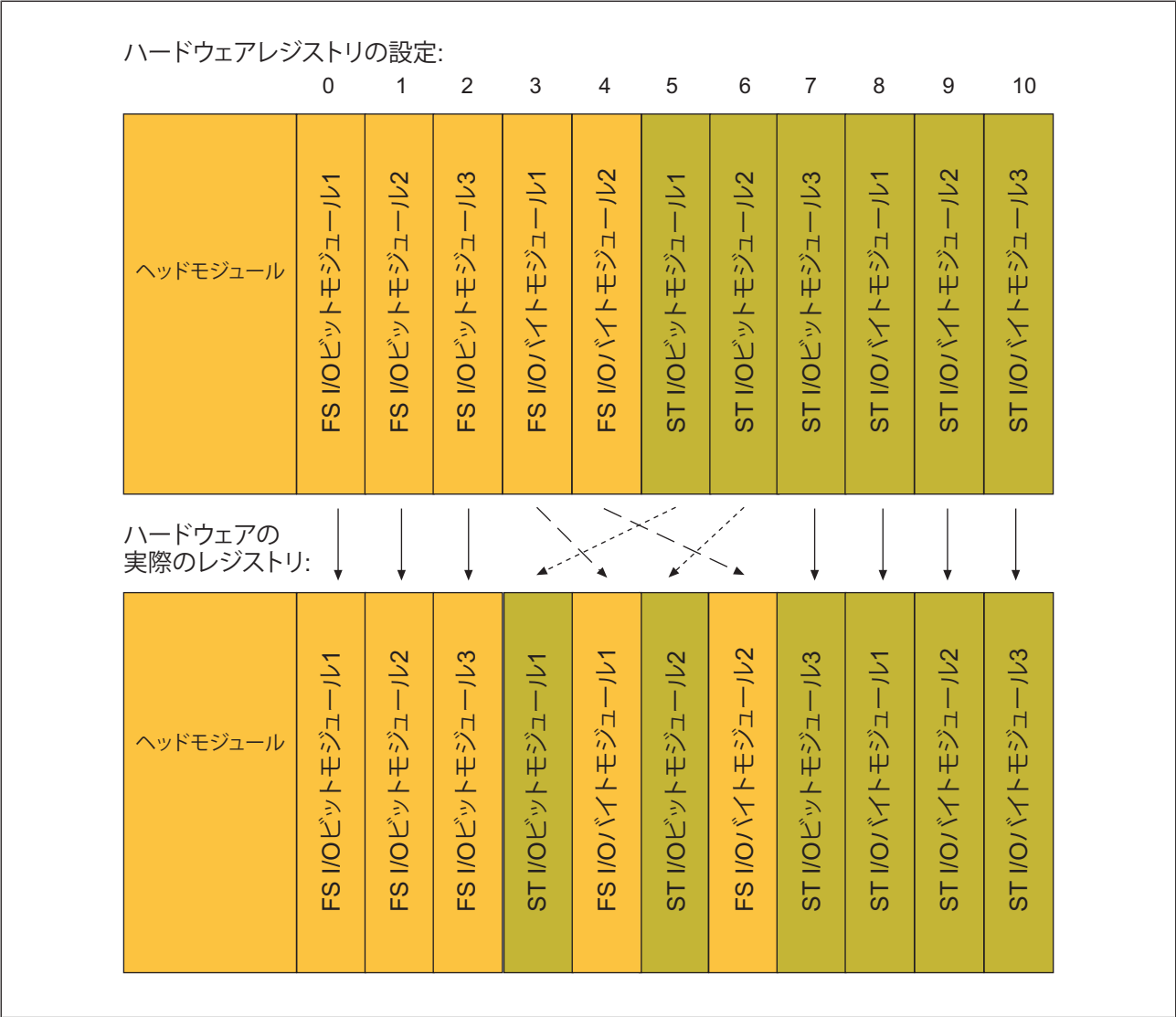


重要

注意:

モジュールを交換する場合、FS電子モジュールのコード化部品でコーディングしたベースユニットに、ST電子モジュールを挿入しないでください。FSモジュールのコード化部品は、それらが黄色ということで識別できます。ベースユニットからコード化部品を取り外した場合は、電子モジュール交換時の試運転／再試運転の仕様に適合して、システムを再試運転する必要があります。『PSS 4000 Safety Manual』を参照してください。

例1:コンフィグレーションエラーが発生しない設定／実際のハードウェアレジストリのバリエーション



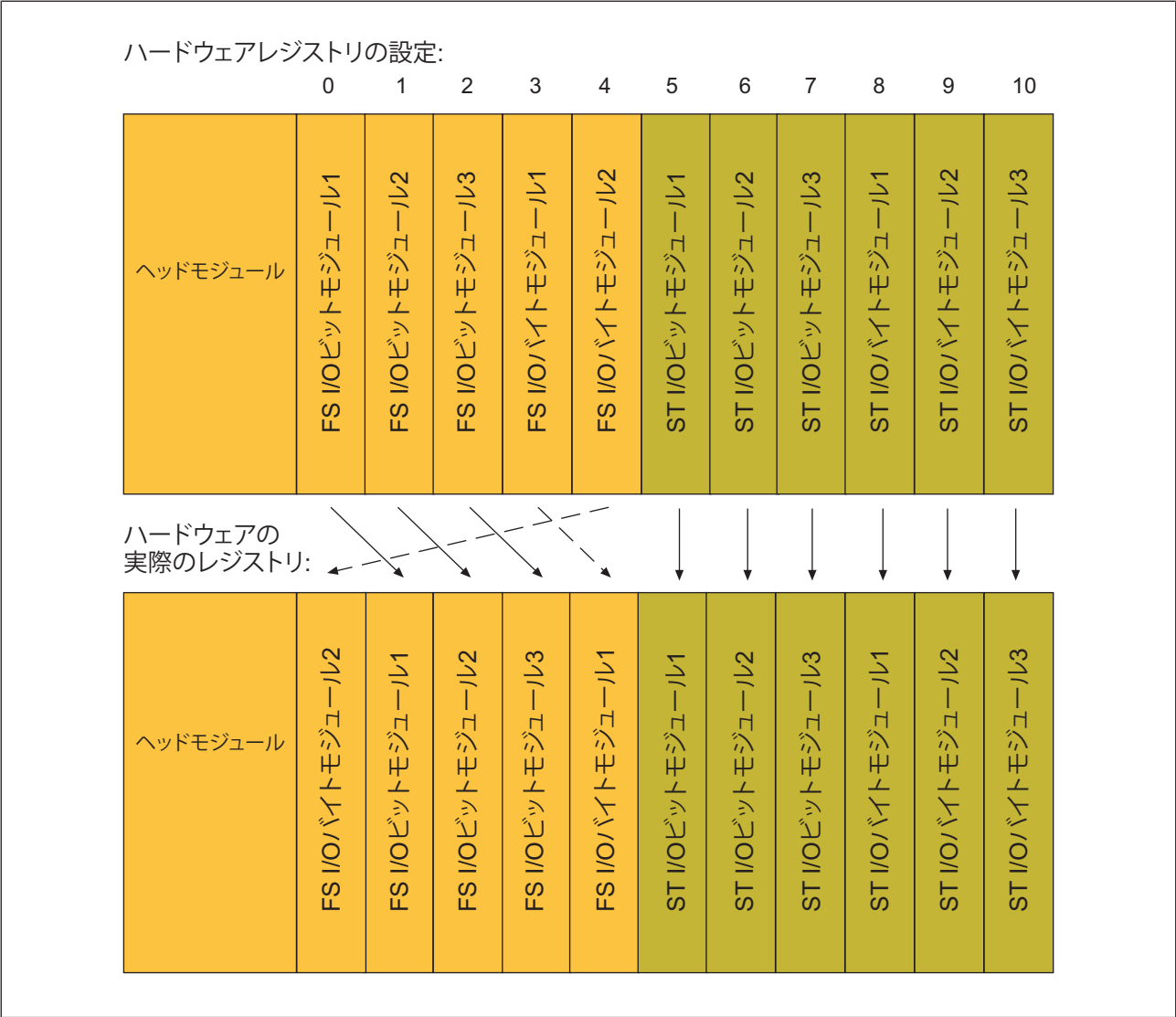
スロットという視点では、設定のハードウェアレジストリと実際のハードウェアレジストリは異なりますが、モジュールグループ内では順序が一致します。設定／実際のハードウェアレジストリの関係において、このバリエーションであれば、コンフィグレーションエラーが発生しません。



重要

この場合、診断メッセージには、ロケーションに関する誤った情報が含まれることに注意してください。

例2:コンフィグレーションエラーを招く、設定／実際のハードウェアレジストリのバリエーション



スロット、またはモジュールグループ内の順序という視点で、設定のハードウェアレジストリと実際のハードウェアレジストリは一致していません。設定／実際のハードウェアレジストリの関係において、このバリエーションでは、**コンフィグレーションエラー**が発生します。

5.9 供給電圧と絶縁

すべての電圧がヘッドモジュール経由で供給されます。さらに電圧が必要な場合（電圧リフレッシュや電位分離など）、これらは追加の電圧供給モジュール経由で供給されます。電圧は、モジュールバス経由でモジュールに分配されます。

▶ モジュール供給

モジュール供給は、ヘッドモジュール、コンパクトモジュールおよび電子モジュール用の内部供給電圧です。

▶ 周辺供給

周辺供給は、電子モジュールのデジタル入出力のセンサおよびアクチュエータに24 VDCを供給します。コンパクトモジュールの場合、入力用の供給電圧は周辺供給から生成されます。出力の供給電圧は、モジュールの端子に供給される必要があります。

モジュールバスの接続を切断して、供給グループを形成することができます。デジタル安全モジュールの場合、周辺供給を使用してテストパルスを提供します。安全入出力モジュールは、周辺供給が正しく接続されるまで起動しません。アナログ入出力モジュールの場合、周辺供給は入出力からガルバニック絶縁されています。

▶ Cレール

Cレールはオプションのフリー電源レールです。Cレールに対しては、説明に文字「-C」が含まれているベースユニット（例: PSSu BP-C 1/8 S、PSSu BP-C 1/8 C）を通じてアクセスできます。

Cレールの電圧供給は、他のベースユニット接続と同じ回路に属する必要があります。PEをCレールに接続することで、シンプルな形式でシールドを得ることができます。コンパクトモジュールの場合、Cレールはループスルーされず、端子では使用できません。Cレールで許容される供給:

- PE
- 0 V
- 画面
- -30 VDC～+30 VDC
- -48 VAC～+48 VAC

▶ アナログモジュールとカウンタモジュールの入出力

アナログモジュールとカウンタモジュールの入出力は、周辺供給からガルバニック絶縁されています。ベースユニットでは、出力の電源は周辺供給から生成されます。

▶ 接地 (グラウンド)

モジュール供給と周辺供給は、それぞれの地電位 (グラウンド) 用の個別の接続を備えています。

▶ 機能接地

ベースユニットとコンパクトモジュールの接点スプリングにより、モジュールの機能接地が取り付けレールに接続されます。

モジュール供給と周辺供給のための供給電圧は、VDE 0100, Part 410に適合する保護用の電氣的分離を備えた低電圧 (PELVまたはSELV) でなければなりません。電源の要件の詳細については、電圧供給モジュールまたはヘッドモジュールのデータシートを参照してください。

2つの電源を使用して電圧を個別に供給する場合、モジュール供給と周辺供給はガルバニック絶縁されます。ガルバニック絶縁が不要な場合は、周辺供給とモジュール供給に共通の電源を使用することができます。

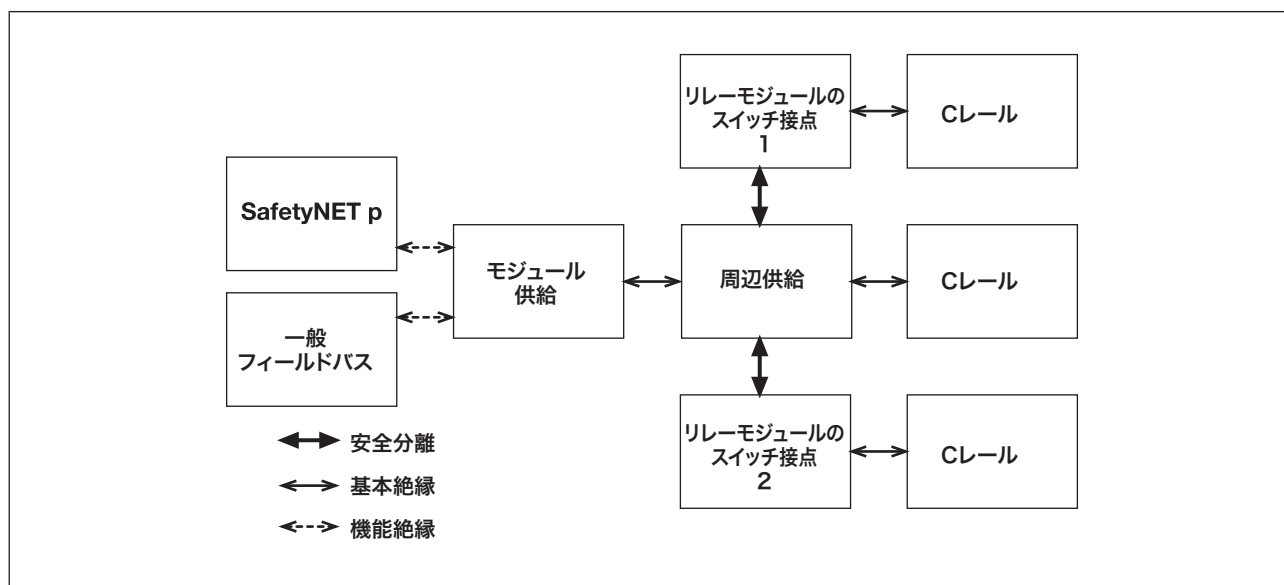
電源モジュール供給と周辺供給に個別の電源を使用するとアベイラビリティが向上し、周辺供給で短絡が発生した場合に、モジュール供給は維持され、ヘッドモジュールは診断目的などでフィールドバスシステムとデータを交換し続けることができます。



情報

PSSuniversalの取り付けマニュアルを参照してください。

PSSuシステムの電源の絶縁:



警告！

感電死の危険があります！

電圧が印加された状態で、通電している部品に接触すると、感電により重大または致命的な負傷につながる恐れがあります。

50 VACまたは120 VDCを超える電圧をPSSuシステム (リレーモジュールのスイッチ接点など) に接続する場合は、次のことに注意してください。

- － 特定の事故防止に関する規定が適用されます。
- － 保護接地 (PE) には、供給グループのCレールのみを接続する必要があります。



情報

FS出力回路は、最大限の安全を保証するように設計されています。この目的のために、社内では広範なテストが実施されています。テスト機能の実行時に周辺供給が低下すると、関連出力に不具合があると判断されます。

この場合、エラー対応がトリガされ、PSSuシステムにあるすべての安全関連システムセクションがSTOP状態に切り替わり、PSSuシステムにあるすべてのFS出力が安全状態に切り替わります。

繰り返し電圧が低下する場合は、高い可用性を維持するために、バッファ機能がある供給電圧から電源を確保する必要があります。

5.10 供給グループ

該当するベースユニットとともに使用した場合、任意の電子電圧供給モジュールを使用して供給を分離したり、供給グループを構成することができます。

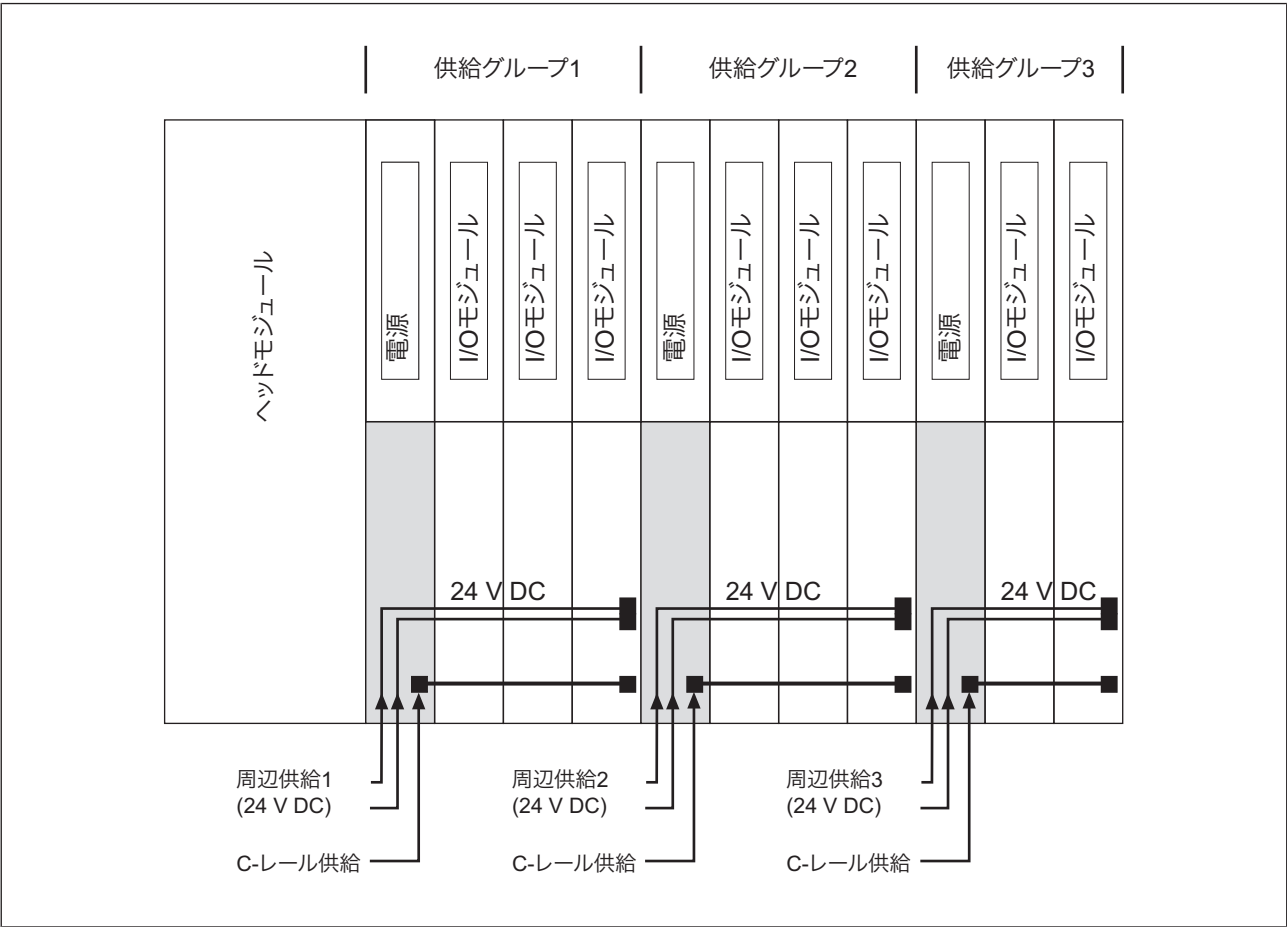
供給を分離するために、前 (左側) のモジュールへの供給 (周辺供給およびCレール供給) が遮断され、新しい供給が後続 (右側) のモジュールに提供されます。

新しい供給グループを使用して、新しい供給電圧を提供してモジュール供給をリフレッシュすることもできます。モジュール供給の0 V供給は、遮断されることなく、常にモジュールバス全体にわたってループスルーされます。

該当する電子電圧供給モジュールの説明書を必ず参照してください。

供給を分離するのに適しているモジュールを以下に示します。

電子モジュール	ベースユニット	モジュール供給	周辺供給 およびCレール
PSSu E F PS	PSSu BS-R 1/8 S PSSu BS-R 1/8 C	リフレッシュ中 (新しい供給電圧が提供され、0 V供給は遮断されない)	左側のモジュールの供給が遮断され、右側のモジュール用の新しい供給が提供される
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	リフレッシュなし (5 Vは分離されない)。	
PSSu E F BSW	PSSu BS 2/8 S PSSu BS 1/8 C		



最初の供給グループの供給電圧をヘッドモジュールに内蔵することもできます。

5.11 最大周囲温度

装置の取扱説明書の技術データに、最大許容周囲温度に関する情報が記載されています。環境要件が過酷な場所での使用に装置が適している場合、動作高度に基づいて、記載された周囲温度を考慮する必要があります。

技術データに記載されている値は、装置を動作高度 (海拔) 2000 mまでで使用する場合に適用されます。もっと高い場所で使用する場合は、周囲温度の制限 (規格IEC 61131-2) を考慮する必要があります。

動作高度 (海拔) [m]	装置の周囲温度の乗算係数
0 ～2000	1.0
3000	0.9
4000	0.8
5000	0.7

環境要件が過酷な場所での使用に適している装置には、製品名に「-T」または「-R」が付いている装置が含まれます。

例:

- ▶ 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

関連装置の取扱説明書の「意図した使用」に記載されている情報を参照してください。

5.12 ヘッドモジュールのステータスLED

ヘッドモジュールには、さまざまなシステムセクションのステータス情報を提供する多数のステータスLEDが含まれています。ただし、すべてのヘッドモジュールが下記のすべてのステータスLEDを提供しているとは限りません。

使用しているヘッドモジュールによっては、外部通信用にシステムセクションが使用可能になります。この場合、各オペレーティングステータスがステータスLEDによって示されます (例: PROFIBUS-DPスレーブ、PROFINET IO装置、PROFIsafe搭載PROFIBUS-DPスレーブ、EtherNet/IPアダプタ)。これらのステータスLEDの詳細については、関連するヘッドモジュールの取扱説明書を参照してください。

使用しているヘッドモジュールの取扱説明書を参照してください。



情報

診断テーブルと診断ログで、診断メッセージと処置を確認できます。

凡例



LED点灯



LED点滅



LED消灯

MBUS

[MBUS] LEDは、FSおよびSTモジュールバスのステータスを示します。

カラー	ステータス	意味
- - -		モジュールはなし
緑		FSおよびSTモジュールバスがエラーなく動作中
赤		オペレーティングステータス「PSSuシステムのすべてのFS出力が安全状態」または 1つ以上のモジュールが見つからない (動作中にモジュールが削除された、設定されたハードウェアレジストリが実際のものと一致しないなど)
		オペレーティングステータス「FSモジュールバスがエラーでSTOP状態: FSの重大エラー」

SDカードまたはメディア

LEDはSDカードのステータスを示し、装置識別にも使用されます。

カラー	ステータス	意味
---		モジュール電源のための供給電圧がない
赤		▶ SDカードが見つからない または ▶ SDカードが認識されない または ▶ SDカードが故障している または ▶ リカバリモード: 何らかの理由で、SDカードのファイルシステムが整合状態を再構築できませんでした。
		▶ 「装置プロジェクトを装置に関連付ける」機能: - 装置に、装置プロジェクトと一致する装置キーがありません。 または - 装置プロジェクトに、装置と一致するプロジェクトキーがありません。 および／または ▶ 「装置プロジェクトをSDカードに関連付ける」機能: 装置プロジェクトはSDカードに関連付けられますが、このSDカードが装置に挿入されていません。
緑		装置上の命名データおよび装置プロジェクトがSDカードのものと一致する
		▶ SDカードの製品型式が装置／ヘッドモジュールの製品型式と一致しない または ▶ SDカードに装置プロジェクトがない または ▶ リカバリモード: SDカードのファイルシステムが整合状態にあり、SDカードを取り外すことができます。
緑-赤		装置上の命名データおよび装置プロジェクトがSDカードのものと一致しない

カラー	ステータス	意味
オレンジ色		ユーザが装置識別を有効化 ▶ SafetyNET p (PAS4000) でPSS 4000装置として認識 または ▶ PROFINET (PROFINET用エンジニアリングツール) でPROFINET IO装置として認識

FS RUN

[FS RUN] LEDはFSリソースのステータスを示しています。

カラー	ステータス	意味
- - -		FSリソースが開始されていないか、STOP状態になっている
緑		オペレーティングステータス「FSリソースがRUN状態、エラーなし」: FSリソースのタスクは異常なしで実行中。 プロジェクトはライセンス登録済みです。
		オペレーティングステータス「FSリソースがエラーでRUN状態」: - タスクはTERMINATED状態 - タスクはSTOP状態 1つ以上のFSリソースタスクが実行されていない プロジェクトはライセンス登録済みです。
オレンジ色		オペレーティングステータス「FSリソースがRUN状態、エラーなし」: FSリソースのタスクは異常なしで実行中。 プロジェクトはライセンス未登録です。
		オペレーティングステータス「FSリソースがエラーでRUN状態」: - タスクはTERMINATED状態 - タスクはSTOP状態 1つ以上のFSリソースタスクが実行されていない プロジェクトはライセンス未登録です。

ST RUN

[ST RUN] LEDはSTリソースのステータスを示しています。

カラー	ステータス	意味
---		STリソースが開始されていないか、STOP状態になっている
緑		オペレーティングステータス「STリソースがRUN状態、エラーなし」: STリソースのタスクはエラーなしで実行中。 プロジェクトはライセンス登録済みです。
		オペレーティングステータス「STリソースがエラーでRUN状態」: - タスクはTERMINATED状態 - タスクはSTOP状態 1つ以上のSTリソースタスクが実行されていない プロジェクトはライセンス登録済みです。
オレンジ色		オペレーティングステータス「STリソースがRUN状態、エラーなし」: STリソースのタスクはエラーなしで実行中。 プロジェクトはライセンス未登録です。
		オペレーティングステータス「STリソースがエラーでRUN状態」: - タスクはTERMINATED状態 - タスクはSTOP状態 1つ以上のSTリソースタスクが実行されていない プロジェクトはライセンス未登録です。

DIAG

[DIAG] LEDは、PSSuシステム/PSS67装置のシステムセクションに異常があるかどうかを示しています。診断リストを使用して詳細な評価を行うことができます。

カラー	ステータス	意味
- - -		起動しているシステムセクションなし、モジュール電源なし
緑		装置に重大度が「エラー」または「警告」レベルのメッセージなし
		装置診断リストと装置診断ログを準備中
赤		1つ以上のシステムセクションに重大度が「エラー」レベルのメッセージがある (診断テーブルを参照)。
		▶ 1つ以上のFSシステムセクションに重大なFSエラーがある (診断リストを参照)。 または ▶ 内部エラーが発生したため、起動プロセスが停止しました。診断ができません。 リセット押しボタンが機能しません。
オレンジ色		装置に重大度が「警告」レベル以上のメッセージがある (診断リストを参照)
赤 - 緑		「作業者の意図的なアクション」の開始 (リセットボタンの機能)

FS FORCE

[FS FORCE] LEDはFSリソースに対する強制およびオンライン変更のステータスを示します。

カラー	ステータス	意味
- - -		FSリソースに対する強制は無効で、有効なオンライン変更はなし
黄		FSリソースで、強制が有効であるか、1つ以上のオンライン変更が有効 (またはその両方) です。

ST FORCE

[ST FORCE] LEDはSTリソースに対する強制およびオンライン変更のステータスを示します。

カラー	ステータス	意味
- - -		STリソースに対する強制は無効で、有効なオンライン変更はなし
黄		STリソースで、強制が有効であるか、1つ以上のオンライン変更が有効 (またはその両方) です。

FS SNp

[FS SNp] LEDは、安全関連のシステムセクションFS SafetyNET pのステータスを示します。

カラー	ステータス	意味
---		システムセクションFS SafetyNET pが開始していない
緑		オペレーティングステータス「FS SafetyNET pがRUN状態、エラーなし」
		オペレーティングステータス「FS SafetyNET pがマイナーエラーでRUN状態」
赤		オペレーティングステータス「FS SafetyNET pがエラーでSTOP状態: FSの重大エラー」
		オペレーティングステータス「FS SafetyNET pがエラーでSTOP状態: Major FS+ST error」

ST SNp

[ST SNp] LEDは、非安全関連のシステムセクションST SafetyNET pのステータスを示します。

カラー	ステータス	意味
---		システムセクションST SafetyNET pが開始していない
緑		オペレーティングステータス「ST SafetyNET pがRUN状態、エラーなし」
		オペレーティングステータス「ST SafetyNET pがマイナーエラーでRUN状態」
赤		オペレーティングステータス「ST SafetyNET pがエラーでSTOP状態: Major FS+ST error」
		

5 V、24 V

[5 V] LEDはモジュール電源のステータスを示します。

カラー	ステータス	意味
---		モジュール電源のための供給電圧はなし、または供給電圧に異常あり
緑		モジュールバスでモジュール電源を使用できる

[24 V] LEDは周辺供給のステータスを示します。

カラー	ステータス	意味
---		周辺供給のための供給電圧はなし、または供給電圧に異常あり
緑		モジュールバスで周辺供給を使用できる

X3:LNK、X3:TRF、X4:LNK、X3:TRF

PSSuシステムでは、1つのイーサネットインタフェース (X3)、または2つのイーサネットインタフェース (X3およびX4) を使用できます (「[イーサネットインタフェース](#)」[\[83\]](#)を参照)。イーサネットインタフェースは、ヘッドモジュールの表示部である2つのステータスLEDに割り付けられます。ステータスLEDは、接続と通信のさまざまなステータスを示します。

X3: LNK、X3: LNK

カラー	ステータス	意味
---		ネットワーク接続なし
緑		ネットワーク接続にはエラーがない

「LNK」の表記は「LINK」を表します。

X3: TRF、X4: TRF

カラー	ステータス	意味
---		データトラフィックなし
黄		データトラフィックはエラーなし

「TRF」の表記は「TRAFFIC」を表します。

6 Pss67装置に関する基本

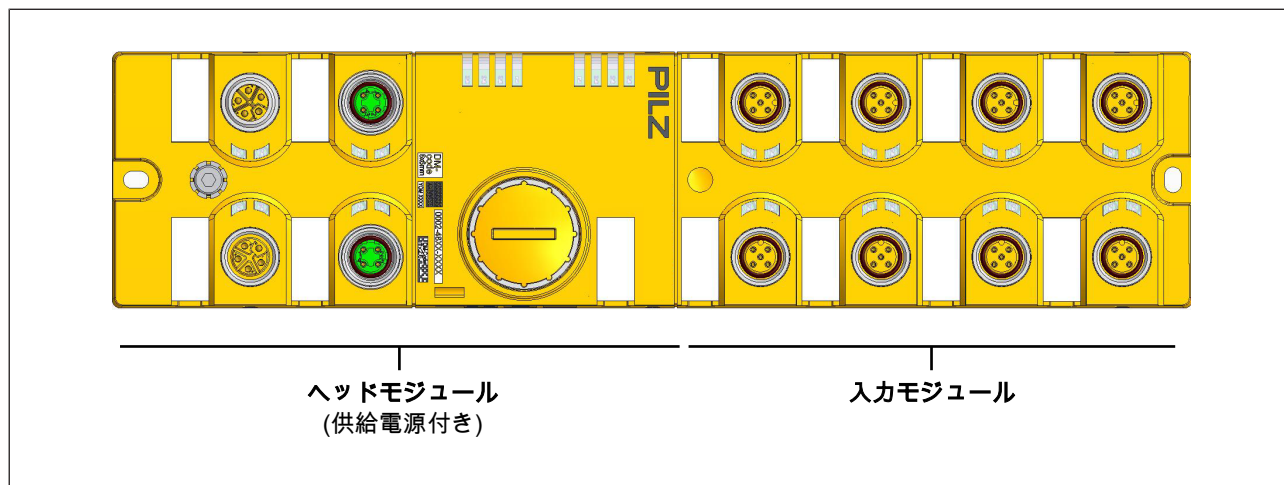
PSS67装置はPSS 4000専用装置です。PSS67装置は、コンパクト設計のPSS 4000専用装置です。

装置で利用できるシステムセクションは、装置によって異なります (「[システムセクション](#)」 22]」を参照)。
詳細は、関連する装置の取扱説明書に記載しています。

6.1 構造

原則として、PSS67装置はIP67アプリケーション用のコンパクト設計のPSSuシステムです。IP67ハウジングの内容:

- ▶ ヘッドモジュール
- ▶ 供給電源
- ▶ 一般アプリケーションおよび安全アプリケーション用の入力モジュール



ヘッドモジュール

ヘッドモジュールは、PSS67装置のデータトラフィック全体を調整し、そのパフォーマンスクラスを決定します。パフォーマンスクラスは、装置の基本的な機能を決定します。パフォーマンスクラスは次のように分類されます。

- ▶ PSS 4000 I/O (略称: I/O)
制御機能を備えていないPSS 4000専用装置
- ▶ PSS 4000 multi (略称: multi)
コンパクト安全制御システムの機能を備えたPSS 4000専用装置
- ▶ PSS 4000 PLC (略称: PLC)
オートメーション設備を制御するためのPSS 4000専用装置

パフォーマンスクラスに応じたPSS 67装置の機能:

- ▶ パフォーマンスクラスPSS 4000 I/OのPSS67装置
 - パフォーマンスクラスPSS 4000 PLCまたはPSS 4000 multiの制御システムによるSafetyNET p経由のI/Oの制御
- ▶ パフォーマンスクラスPSS 4000 PLCのPSS67装置
 - SafetyNET pを使用してネットワーク接続している場合、すべてのパフォーマンスクラスの分散型PSS 4000装置からのI/Oの制御
 - STリソースとFSリソースを使用できる
 - 1つのリソースにつき最大9つのタスク
 - PAS4000 Multiエディタでプログラミングまたは、PAS4000でIEC 61131に従ってプログラミング

入力モジュール

デジタル入力のあるPSS67装置を使用できます。

供給電源

ヘッドモジュールと入力モジュールには、供給電源を介してモジュール供給 (5V) と周辺供給 (24 V) が供給されます。

6.2 モジュールバス (MBUS)

ヘッドモジュールと入力モジュールは、内部モジュールバスを介して接続されます。内部モジュールバスは、ヘッドモジュールと入力モジュールの間のデータを転送します。

6.3 PSS67装置のロット

ロットは、PSS67装置上の入力／出力モジュールの場所を示します。

PSS67デバイスは、入力/出力モジュールのロットは決められた順序で単一ハウジング内に固定されています。番号はロット0から始まります。

ロットの詳細については、関連するPSS 67装置の取扱説明書を参照してください。

6.4 供給電源

PSS67装置の電圧:

- ▶ モジュール供給

モジュール供給は、ヘッドモジュールおよび入力モジュール用の内部供給電圧です。

- ▶ 周辺供給

周辺供給は、センサへの24VDC、およびデジタル入力モジュールのテストパルスを供給します。

アース (グラント) および機能アースの接続:

- ▶ 接地 (グラント)

モジュール供給と周辺供給は、それぞれの地電位 (グラント) 用の個別の接続を備えています。

- ▶ 機能アース

PSS67装置には、機能アースへの接続用の固定スクリーが含まれています。

モジュール供給と周辺供給の供給電圧は、VDE 0100, Part 410に適合する安全な電气的分離を備えた低電圧 (PELVまたはSELV) でなければなりません。電源のその他の要件については、該当するPSS 67装置の取扱説明書を参照してください。

2つの電源を使用して電圧を個別に供給する場合、モジュール供給と周辺供給はガルバニック絶縁されます。ガルバニック絶縁が不要な場合は、周辺供給とモジュール供給に共通の電源を使用することができます。

電源モジュール供給と周辺供給に個別の電源を使用すると、可用性が向上し、周辺供給で短絡が発生した場合でもモジュール供給は維持され、ヘッドモジュールは診断目的などでフィールドバスシステムとデータ交換を続けることができます。



重要

該当するPSS67装置の取扱説明書で供給電圧の接続および最大電流負荷容量に関する情報を参照してください。

6.5 ステータスLED

PSS67装置には、さまざまなシステムセクションのステータス情報を提供する多数のステータスLEDが含まれています。これらのステータスのLEDの意味は、PSSuシステムのヘッドモジュールのLEDと同じです（「[ヘッドモジュールのステータスLED](#)」 70」を参照）。



情報

PSS67装置とPSSuシステムの次の相違点に注意してください。

- PSS67装置では、イーサネットインターフェースは、X3およびX4ではなくX21およびX22です。ただし、ステータスLEDは、機能とラベルについては同じです。
- ステータスLED「MBUS」は、PSS67装置では意味がないのでPSS67装置には含まれていません。

詳細については、該当するPSS67装置の取扱説明書を参照してください。

7 イーサネット

7.1 Auto IP機構

Auto-IPメカニズムは、Ethernetネットワーク内で手動コンフィグレーションする代わりに、空いているIPアドレスを自動的に装置に割り付けるために使用されます。このために、IANAは範囲169.254.0.0/16のアドレスを提供しています。

PSS 4000専用装置

PSS 4000専用装置では、デリバリストータスで、またはオリジナルリセット後にAuto-IPメカニズムがアクティブになります。有効化は[システムセクションの起動](#)  [336](#)中に実行されます。起動プロセス中、Auto-IPメカニズムは、これらの装置に169.254.1.0～169.254.254.255の空いているIPアドレスを割り付けます。Auto-IPメカニズムは、装置の命名手順の間に無効化されます (「[PSS 4000専用装置での装置命名手順](#)  [310](#)」を参照)。

PSS 4000に対応する装置

PSS 4000対応装置では、169.254.0.0/16の範囲で空いているIPアドレスを選択します。

7.2 イーサネットインタフェース

PSS 4000装置には、イーサネットに接続するためのイーサネットインタフェースが備わっています。イーサネットインタフェースの仕様 (例: 2つの空きスイッチポートがある内部スイッチ)、接続技術 (例: RJ45)、イーサネットインタフェースの名前 (例: イーサネット、SafetyNET p) は装置によって異なります。使用している装置の取扱説明書の情報を参照してください。



重要

オートセンシング／オートネゴシエーションファンクションをサポートするイーサネットサブスクライバ (例: スイッチ、ハブ) のみを使用してください。それ以外の場合、二重モードの競合 (二重モードの不一致) が発生して、転送速度の大幅な低下や、散発的なデータ転送エラーが発生することがあります。

PSS 4000オートメーションシステムでは、イーサネットインタフェースを次のように使用します。

- ▶ SafetyNET p  89] 経由でデータ交換を行う接続
- ▶ Modbus/TCP、Raw UDPなど、IP接続で外部通信とデータ交換を行う接続 (「外部通信」を参照)。
- ▶ PAS4000 PCの接続 (= プログラミングインタフェース)
- ▶ OPCサーバ  488] の接続

7.2.1 RJ45インタフェース

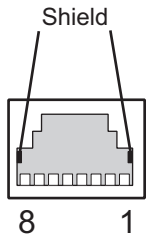
PSS 4000装置にはRJ45接続技術によるイーサネットインタフェース (RJ45インタフェース) を搭載できます。RJ45インタフェースの数は装置によって異なります。ピルツのPSS 4000専用装置とPSS 4000対応装置の詳細は、該当するユニットの取扱説明書に記載されています。

2つのRJ45インタフェースを搭載したPSS 4000専用装置には、2つの空いているスイッチポートを持つ内部スイッチがあります。そのため、これらの装置ではスイッチやハブを増設する必要がありません。

PSS 4000専用装置のRJ45インタフェースには、オートセンシングおよびオートクロス機能があります。オートセンシング機能では、データ転送中にビットレートを自動的に検出します。オートクロス機能により、接続ケーブルで、パッチケーブル (アンクロスドデータライン接続) とクロスオーバーケーブル (クロスオーバーデータライン接続) を区別する必要がなくなります。正しいデータライン接続が内部で自動的に確立されます。このため、パッチケーブルを端末装置とカスケードの接続ケーブルとして使用できます。

7.2.2 RJ45インタフェースの割り付け

RJ45インタフェースを搭載したPSS 4000専用装置には、次のインタフェースが割り付けられています。

RJ45ソケット (8ピン)	PIN	標準	クロスオーバ
	1	TD+ (送信+)	RD+ (受信+)
	2	TD- (送信-)	RD- (受信-)
	3	RD+ (受信+)	TD+ (送信+)
	4	n.c.	n.c.
	5	n.c.	n.c.
	6	RD- (受信-)	TD- (送信-)
	7	n.c.	n.c.
	8	n.c.	n.c.
	n.c.= 未接続		

RJ45コネクタ用ソケット (計画ビュー)

7.2.3 RJ45のコネクタと接続ケーブル

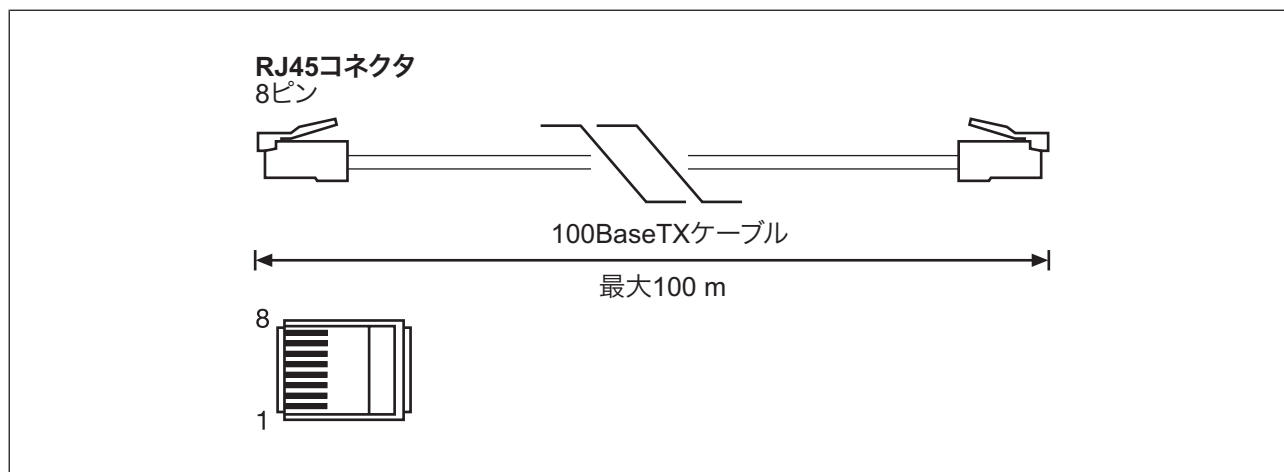


図: RJ45コネクタの割り付け (下からの平面図)

次に示すコネクタとケーブルの最低要件に**注意**してください。

100BaseTケーブル

ビットレート	100 MBit/s
最大セグメント長	100 m
中	STP、S/STP [1] (シールド／ダブルシールドツイストペアケーブル) 2ペア カテゴリ5以上 [2]
接続	シールド8ピンRJ45コネクタ
トポロジ	ポイントツーポイント／ハブ (ハブ、スイッチ)

[1]

産業環境のイーサネットアプリケーションの場合、ピルツはダブルシールドツイストペアケーブル (S/STP) を推奨します。

[2]

適切なTPケーブル:

カテゴリ5	最大100 MHz (10Base-T、100BaseT4、100BaseTX)
カテゴリ5e	最大100 MHz (1000Base-T)
カテゴリ6	最大250 MHz
カテゴリ6a	最大500 MHz (10GBase-T)
カテゴリ7	最大600 MHz
カテゴリ7a	最大1000 MHz

イーサネットアプリケーション用TPケーブルの波動インピーダンスは通常100 Ohmです。

この接続は、アンクロスドデータライン接続の100BaseTXケーブル (パッチケーブル) で確立できます。

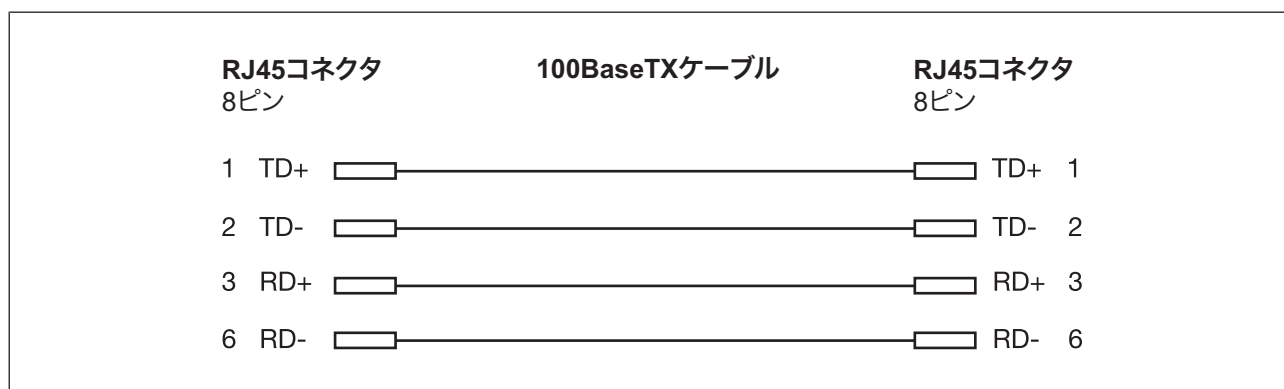


図: 100BaseTXケーブル (標準)



情報

接続ケーブルを自社で組み立てている場合、同じ製造元のコネクタとケーブルを使用することをお勧めします。これは、コネクタの寸法と配線接続線径に関して、製造元で指定される許容値に大幅な差異があるためです。

7.2.4 RJ45インタフェースのステータスLED

接続／通信のさまざまなステータスを示すために、一般的にRJ45バージョンのイーサネットインタフェースにはステータスLEDが備わっています。詳細は、関連する装置の取扱説明書に記載しています。

8 SafetyNET p

8.1 概要

SafetyNET pは、産業用途に適したイーサネットベースのマルチマスタバスシステムです。PSS 4000では標準でフィールドバスシステムとして使用され、分散型タスクを解決します。

このSafetyNET pプロトコルには安全データチャンネルが含まれており、IEC 61508のSIL 3に適合したデータ転送で認証されています。SafetyNET pでは、安全関連データと非安全関連データの両方が、同じバスケーブルを経由して転送されます。

SafetyNET pは「リアルタイムフレームネットワーク」(RTFN) 形式で使用できます。

SafetyNET pのアプリケーション層 (ISO/OSIの7階層モデルの第7層) はCANopenに基づいています。SafetyNET pは、CANバスプロトコルを標準化された産業アプリケーション層と組み合わせます。この規格には、通信の標準化と、分散型のオートメーションユニットをネットワーク化できるようにする技術および機能の性能データが含まれています。定義されたアプリケーションオブジェクトは、装置プロファイルで使用されます。

8.1.1 SafetyNET pプロトコルのバージョン

IEC 61784-3の改正には、SafetyNET pプロトコルに対する追加機能が含まれています。これらの追加機能は、SafetyNET pプロトコルバージョン2で既に考慮されています。

元のプロトコルバージョンは、引き続きSafetyNET pプロトコルバージョン1でサポートされます。



情報

備考:

- SafetyNET pプロトコルのバージョンは、PSS 4000プロジェクトのプロジェクトプロパティでコンフィグレーションされます。
- SafetyNET pプロトコルバージョン1を使用する装置とバージョン2を使用する装置を、1つのPSS 4000プロジェクト内で混在させることはできません。

8.1.2 ISO/OSIの7階層参照モデルのSafetyNET p

ISO/OSI (Open System Interconnection、開放型システム間相互接続)参照モデルは、すべてのデータ処理通信システムについて、あらゆるデータ通信機能を記述するための標準形式です。機能は分割されて、7つの異なる機能を持つ階層モデルを形成しています。フィールドバス通信は第1層、第2層、第7層によって記述されます。

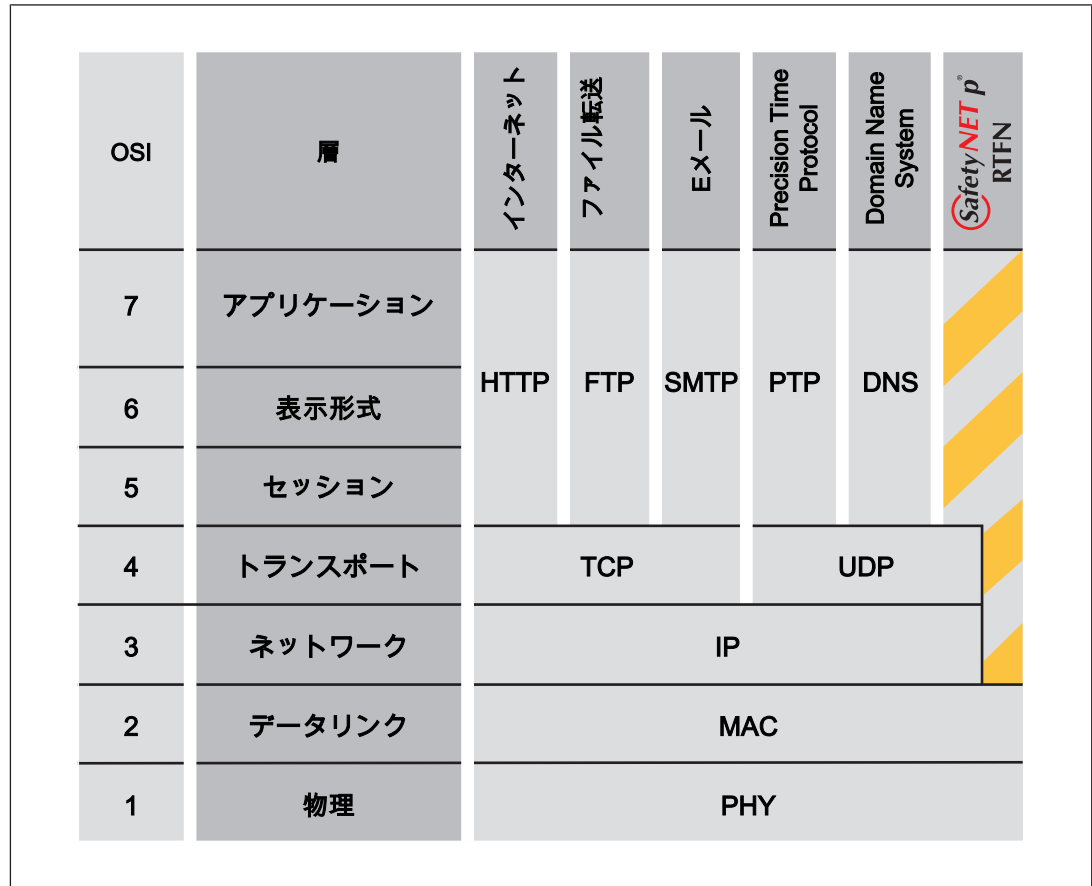


図: ISO/OSIの7階層参照モデルのSafetyNET p

8.1.3 RTFN (リアルタイムフレームネットワーク)

RTFNトランスポート層はプロセス制御とセルレベルで使用します。

RTFNはオフィスオートメーションで普及している標準インターネットと完全な互換性があります。同じトランスポートメカニズムに基づいているためです。RTFNは、Ethernetネットワーク内の他のユーティリティと制限なく混合されるように設計されているので、RTFNサブスライバはネットワークの他の装置 (ラップトップ、プリンタなど) と対等に通信します。



情報

RTFNに対応しない装置がネットワークに高負荷を与えると、SafetyNET pの可用性に悪影響を及ぼす可能性があります。そのため可用性要件が厳しい場合は、SafetyNET pと並行して他の通信を実行しないこと、または管理機能付きのスイッチを使用することをお勧めします。

8.1.4 安全関連および非安全関連データの転送

SafetyNET pを使用し、統合された安全通信層を経由して安全関連データを転送することができます。セキュリティメカニズムは、IEC 61508のSIL3に適合するように設計されています。安全関連データは、SafetyNET pテレグラム内にカプセル化されて送信されます。したがって、スイッチやケーブルなど、その他のすべてのネットワークコンポーネントを標準的なイーサネットコンポーネントとして設計できます。これらは安全関連データの転送に影響を及ぼしません。

PCや標準的な制御システムなど、非安全関連SafetyNET pサブスクリバが安全関連の通信に影響することはありません。したがって、安全関連サブスクリバと非安全関連サブスクリバをネットワーク内で混在させて運用できます。

8.2 SafetyNET pのトポロジ

SafetyNET pはイーサネットベースのバスシステムであるため、ライントポロジやスタートポロジなど、イーサネットで普及している有効なあらゆる構造に対応できます。また動的な構造も実装できます。

8.2.1 リニアトポロジ

リニアトポロジは、回線内の複数のバスサブスクライバをブランチなしで接続することで作成されます。このトポロジは、普及しているフィールドバス構造です。リニアトポロジは、広域に展開している設備 (例: コンベアシステム、相互接続型の製造セル) で有効です。

多くの場合、SafetyNET pサブスクライバは、2つのイーサネットインタフェース (2つの空きポートがあるオートセンシングスイッチ) を使用してリニアトポロジを実現します。したがって、スイッチなどのネットワークコンポーネントの追加は不要です。

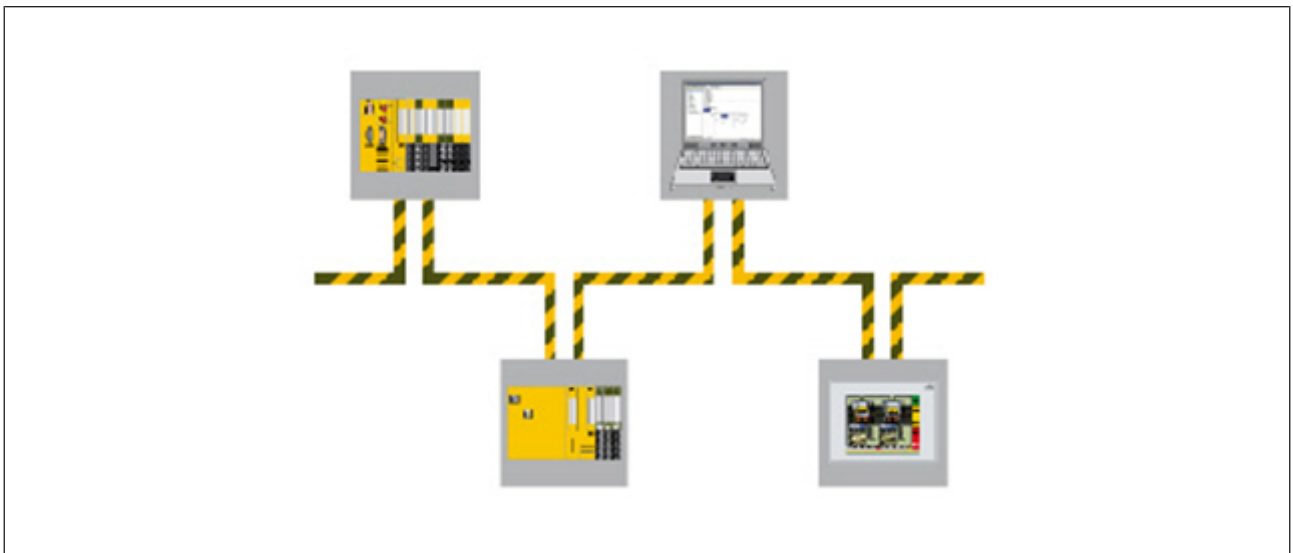


図: リニアトポロジ (原理)

8.2.2 スタートポロジ

スタートポロジは、中央部で複数のサブスクライバを接続するときに形成されます。スイッチは一般的に、中央信号配信機として使用されます。通信相手は、ポイントツーポイント接続でスイッチを経由して通信します。スタートポロジは、物理的に狭い範囲に装置が密集している現場 (例: 小規模製造セル、独立型の生産機械) に適しています。

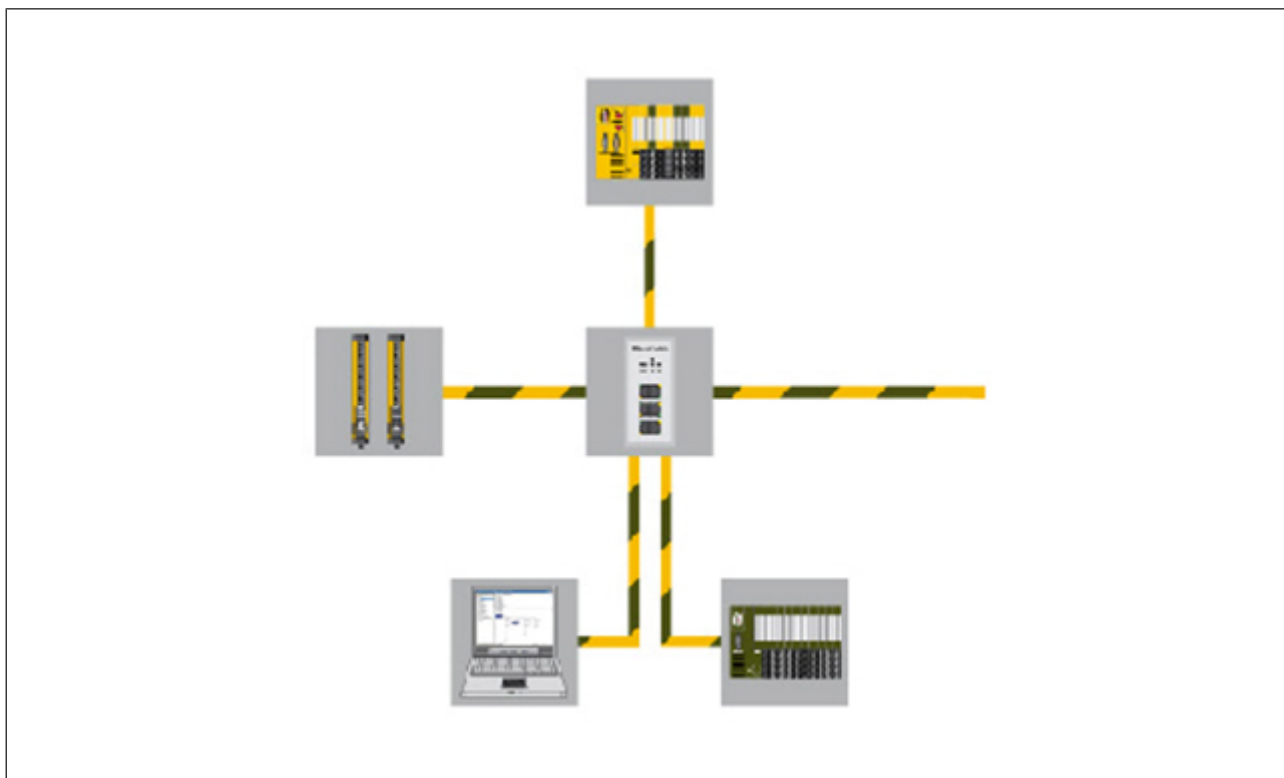


図: スタートポロジ (原理)

8.2.3 動的構造

SafetyNET pは、ホットスワップのサブスクライバに対して、一般的なイーサネットの容量を動的にサポートします。これはサービス、メンテナンス、診断のために特に役立ちます。

8.3 SafetyNET pコンポーネントの選択

SafetyNET pは、安定性が保証されたプラグインコネクタ、経済的なIP20配線を、制御キャビネット内でRJ45コネクタの形式で使用することで、使い慣れた強固なフィールドバス接続技術をサポートします。

パフォーマンスが十分である場合は、従来の標準的なイーサネットインフラを使用できます。このインフラには、コネクタ、ケーブル、ルータ、スイッチ、ゲートウェイ、および通信チャンネルが含まれます。

ピルツ製部品の使用をお勧めします。

原則、産業環境での使用に適していれば他の部品も使用できます。



情報

接続ケーブルをお客様ご自身で組み立てる場合は、同一製造業者のコネクタとケーブルを使用することを推奨しています。許容誤差により、コネクタの寸法と配線接続線径が大幅に異なる可能性があるためです。

8.4 SafetyNET pインタフェース

SafetyNET p経由でデータ交換を行う場合、物理的な接続はイーサネットインタフェース  83]で確立します。



重要

オートセンシング／オートネゴシエーション機能をサポートするSafetyNET pサブスクリバ (例: スイッチ、ハブ) のみを使用してください。それ以外の場合、二重モードの競合 (二重モードの不一致) が発生して、転送速度の大幅な低下や、散発的なデータ転送エラーが発生することがあります。

8.5 SafetyNET pを介したプロセスデータ交換

ST SafetyNET pおよび／またはFS SafetyNET pシステムセクションがあるPSS 4000装置であればどんなものでも、SafetyNET pを介したプロセスデータの交換が可能です。

PSS 4000装置のFS/ST SafetyNET pシステムセクションは、送信接続を介して送信データ領域から通信パートナーに処理データを送信し、受信接続を介して受信データ領域で通信パートナーから処理データを受信します。

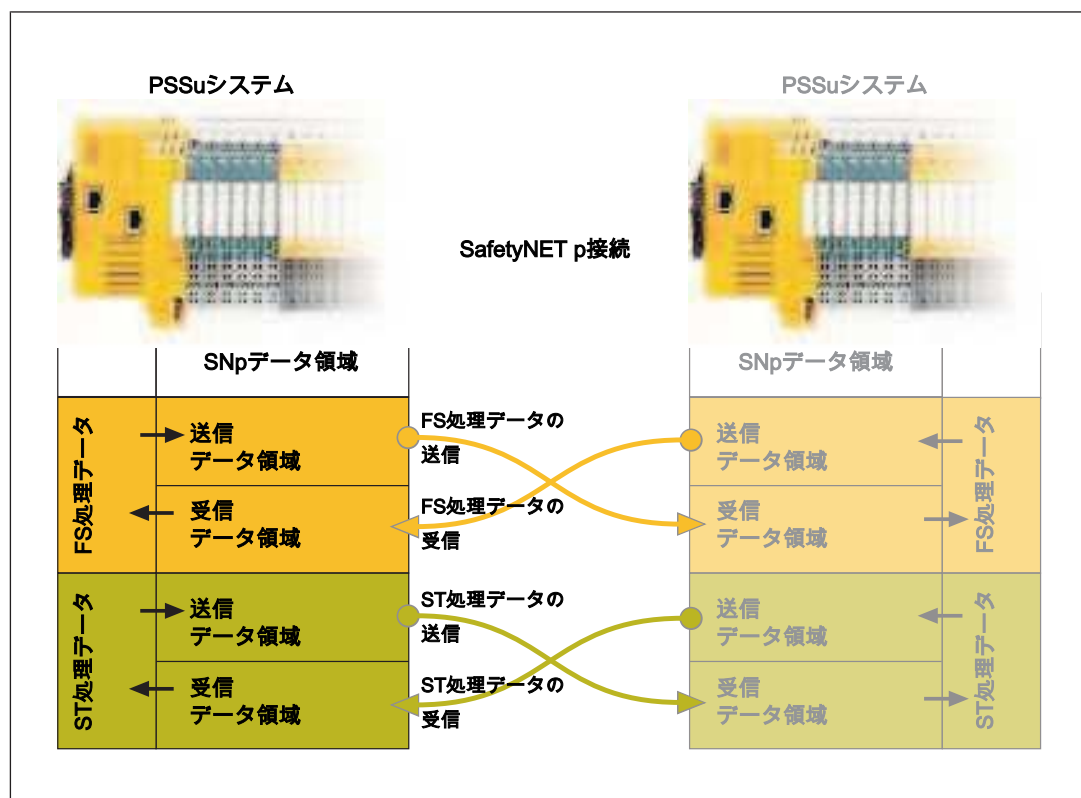


図: SafetyNET pを介した処理データ交換の原理 (例: PSSuシステム)

安全関連処理データ (FS処理データ) は、SafetyNET pの安全データチャンネルの安全接続 (FS接続) 経由で送信されます。非安全関連処理データ (ST処理データ) は、SafetyNET pの非安全関連データチャンネルの非安全関連接続 (ST接続) 経由で送信されます。

各接続は、送信部と受信部で構成されます。送信部には、送信側装置の送信データ領域の領域が含まれます。この領域はTXデータ領域と呼ばれます。受信部には、受信側装置の受信データ領域の領域が含まれます。この領域はRXデータ領域と呼ばれます。

したがって、各接続には、送信側装置上に独自のTXデータ領域があり、受信側装置上に独自のRXデータ領域があります。

▶ FS処理データの処理データ交換

FSプロセスデータは、FS接続を介して交換されます。つまり、FSプロセスデータは、装置の対応する (FS-)Txデータ領域からFS-Tx接続を介して送信され、受信側装置から見ると、対応する (FS-)Rxデータ領域で、FS-Rx接続を介して受信されます。

▶ ST処理データの処理データ交換

STプロセスデータは、ST接続を介して交換されます。つまり、STプロセスデータは、装置の対応する (ST-)Txデータ領域からST-Tx接続を介して送信され、受信側装置から見ると、対応する (ST-)Rxデータ領域で、ST-Rx接続を介して受信されます。

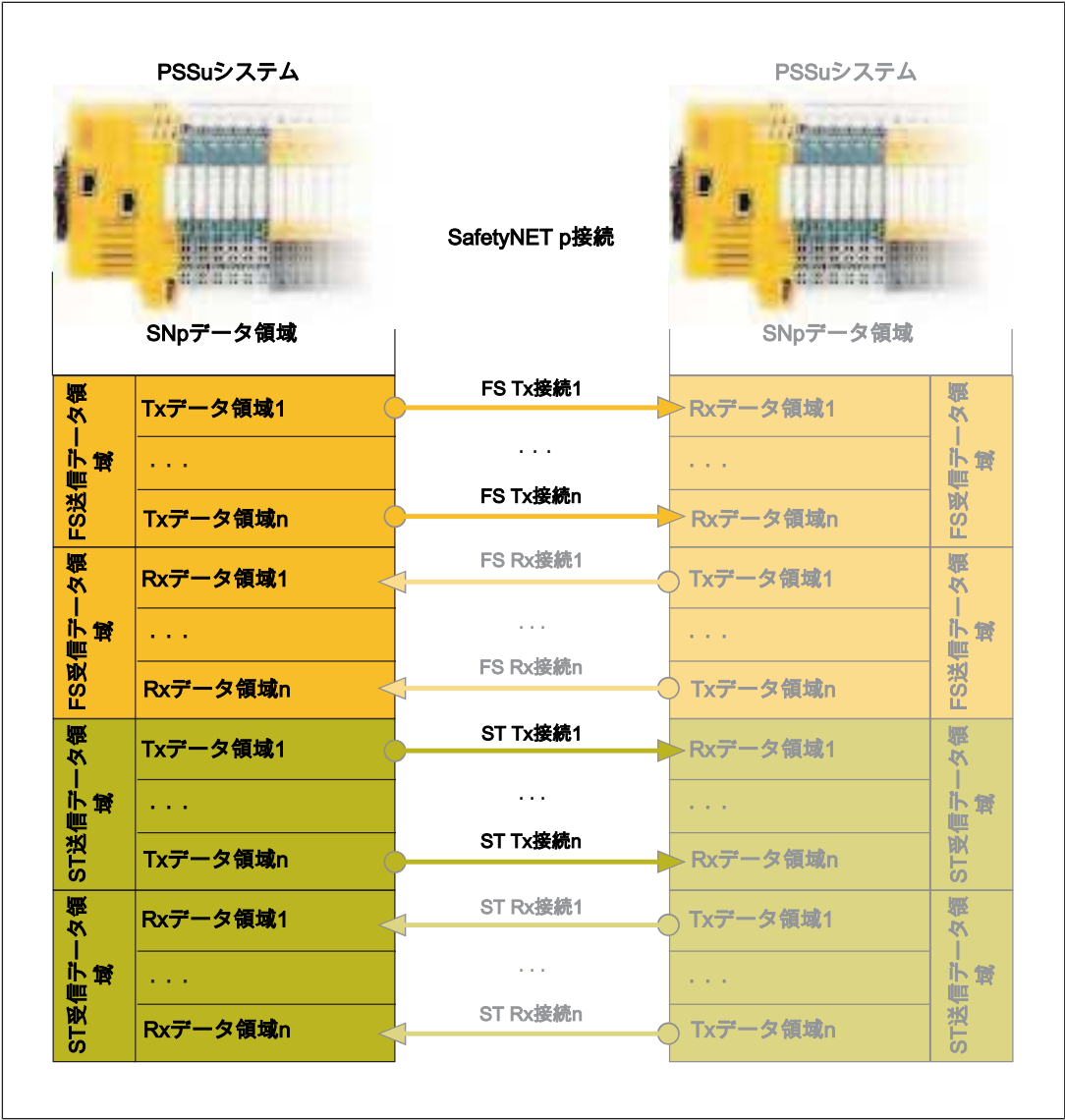


図: 接続を介した処理データ交換の原理 (例: PSSuシステム)

8.5.1 接続の周期的な処理

送信側PSS 4000装置のFS/ST SafetyNET pシステムセクションは、サイクル時間 t_{SNp} でTx接続を周期的に開始し、プロセスデータをSafetyNET p経由で送信します (「[応答時間](#)  456」を参照)。

受信側装置のFS/ST SafetyNET pシステムセクションでは、イベント起動のテストによって、新しいプロセスデータがTx接続に属しているRxデータ領域でさらに処理できるかどうかをチェックします。

Tx接続の開始後、コンフィグレーションされた最大伝送遅延時間 $t_{DelaySNp}$ の時間内に、プロセスデータが対応する受信データ領域内で使用できることが必要です。測定時間 (「[FSハートビート \(FS-HBT\)](#)  113」を参照) が、コンフィグレーションされた最大伝送遅延時間 $t_{DelaySNp}$ を超えると、送信装置から受信したすべてのFS受信接続 (FS-Rx接続) が受信装置で無効になります。代入値は、有効ビット = FALSEの状態で、受信データで使用されます。

システムは、プロジェクト内のすべてのTx接続について、これを監視します (「[応答時間](#)  456」参照)。

8.5.2 SafetyNET p受信タイムアウト

PSS 4000装置はSafetyNET p受信タイムアウトを使用して、受信データに通信エラーがないか監視します。これは、ST通信エラー（システムセクションST SafetyNET p）とFS通信エラー（システムセクションFS SafetyNET p）の両方に当てはまります。理解しやすいように、以下の情報では、受信側の装置から見たSNp通信について、監視が受信側でのみ行われる場合を取り上げています。

- ▶ プロセデータを受信する装置から見た場合、送信側装置もリモート装置と呼ぶことができます。
- ▶ 装置がリモート装置からプロセデータを取得する接続も、受信接続（Rx接続）と呼ぶことができます。
 - ST-Rx接続からの処理データは、対応するST-Rxデータ領域にあります。
 - FS-Rx接続からの処理データは、対応するFS-Rxデータ領域にあります。
- ▶ PSS 4000装置の関連するシステムセクションが、対応するRxデータ領域でさらに処理できる処理データを受け入れると、Rx接続は有効になります。
- ▶ PSS 4000装置の関連するシステムセクションが、対応するRxデータ領域でさらに処理できる処理データの受け入れに失敗するか無視すると、Rx接続は無効になります。
- ▶ Rx接続に属しているRxデータ領域では、SNp受信タイムアウト時間が終了する前に新しい処理データが存在していることが必要です。存在していない場合、PSS 4000装置の関連するシステムセクションが接続の喪失を検出し、Rx接続は無効になります。
- ▶ PSS 4000装置のST SafetyNET pシステムセクションは、1台のリモート装置に対するST-Rx接続と、複数のリモート装置に対するST-Rx接続を維持できます。
- ▶ PSS 4000装置のFS SafetyNET pシステムセクションは、1台のリモート装置に対するFS-Rx接続と、複数のリモート装置に対するFS-Rx接続を維持できます。

システムセクション（ST SafetyNET p/FS SafetyNET p）が通信エラーを検出すると、通信エラーの種類に応じてRx接続が無効になります。FS通信でエラーが発生した場合に無効化されるかどうか、SafetyNET pプロトコルのバージョンによって決まります。

通信エラー後、エラーの原因が存在しなくなると、ST通信（= ST-Rx接続の有効化）は常に自動再開されます（「[ST SafetyNET pがマイナーエラーでRUN状態](#)」 412」を参照）。通信エラーが発生した後にFS通信（= FS-Rx接続の有効化）が再開するかどうかは、SafetyNET pプロトコルのバージョンと通信エラーの種類によって決まります。

- ▶ [SNpプロトコルバージョン1: 通信エラーおよびFS通信の再開](#)  103
- ▶ [SNpプロトコルバージョン2: 通信エラーおよびFS通信の再開](#)  104

8.5.2.1 Snpプロトコルバージョン1: 通信エラーおよびFS通信の再開

Snpプロトコルバージョン1の通信エラーは、次のように分類されます。

▶ FS通信の継続的なエラー

受信側のPSS 4000装置が30秒以内に接続エラーを2回以上検出した場合、エラーは「継続的」とみなされ、リモート装置とのFS通信は永続的に無効になります。

▶ FS通信の単発エラー

受信側のPSS 4000装置が30秒以内に接続エラーを1回だけ検出した場合、エラーは「単発」とみなされ、リモート装置とのFS通信は一時的に無効になります。

PSS 4000装置のFS SafetyNET pシステムセクションは、Snp受信タイムアウトの時間内にリモート装置から新しいプロセスデータを受信できない場合、通信エラーを検出します。実際の処理は次の通りです。

▶ FS SafetyNET p待機時間 (1 s) が開始されます。

▶ PSS 4000装置のFS SafetyNET pシステムセクションが、FS SafetyNET pがマイナーエラーでRUN状態 402に切り替わります。

▶ エラーが回復されるか、エラーが存在しなくなると、早ければFS Snp待機時間 (1秒) が経過した後で、PSS 4000装置はリモート装置とのFS通信を再度有効にすることができます。

– 継続的なエラー

FS通信を再開するには、以下のいずれかのリセットを実行する必要があります。

– 単発エラー

FS通信は自動的に再開します。

8.5.2.2 Snpプロトコルバージョン2: 通信エラーおよびFS通信の再開

Snpプロトコルバージョン2の通信エラーは、次のように分類されます。

▶ FS通信の接続ベースエラー

このエラーは、単一のFS-Rx接続を介したFS通信に関係しています。FS-Rx接続は一時的に無効になります。

▶ FS通信の装置ベースエラー

このエラーは、2台のPSS 4000装置間のFS通信に関係しています。受信側の装置とリモート装置間のすべてのFS-Rx接続が、一時的に無効になります。

▶ FS通信の不特定エラー

このエラーは、1台のPSS 4000装置とそのすべてのリモート装置間のFS通信に関係しています。すべてのリモート装置に対するPSS 4000装置のすべてのFS-Rx接続が、一時的に無効になります。エラーの原因であるリモート装置を特定できない場合、エラーは「不特定」になります。

PSS 4000装置のFS SafetyNET pシステムセクションは、SafetyNET p受信タイムアウトの時間内にFS-Rx接続を介して使用できる新しいプロセスデータがない場合、通信エラーを検出します。通信エラーの種類によっては、複数のFS-Rx接続が接続されている場合もあります。実際の処理は次の通りです。

▶ **FS SafetyNET p待機時間** **105**が開始されます。

▶ PSS 4000装置のFS SafetyNET pシステムセクションが、**FS SafetyNET pがマイナーエラーでRUN状態** **403**に切り替わります。

▶ エラーが回復されるか、エラーが存在しなくなると、早ければFS SafetyNET p待機時間が経過した後、PSS 4000装置のFS SafetyNET pシステムセクションはFS通信を再度有効にすることができます。すべての通信エラーの後、FS通信は自動的に再開します (有効になります)。

8.5.2.3 SNpプロトコルバージョン2: コンフィグレーション可能なFS SafetyNET p待機時間

PSS 4000プロジェクトにSNpプロトコルバージョン2がコンフィグレーションされている場合、PSS 4000プロジェクトのFS SafetyNET p待機時間をコンフィグレーションできます。

FS SafetyNET p待機時間は、すべてのFS SafetyNET pシステムセクションにおいて、FS通信のすべてのエラーを検出するためのPSS 4000プロジェクトの待ち時間です。つまり、FS SafetyNET p待機時間により、PSS 4000プロジェクト内のすべてのFS-Rx接続で、すべての通信エラーを確実に検出できます。

SafetyNET pの待機時間中、影響を受けるFS-Rxデータ領域でさらに処理できる処理データは無視されます。

FS SafetyNET p待機時間の自動計算

PSS 4000プロジェクト (PAS4000) のプロパティで、FS SafetyNET p待機時間を自動的に計算するようにコンフィグレーションすることができます (= デフォルト設定)。システムは自動計算機能を使用して、プロジェクトの最小のFS SafetyNET p待機時間 (= 最小許容値) を決定します。FS SafetyNET pの最小待機時間は、プロジェクトのビルド時に表示されます。

ユーザによるFS SafetyNET p待機時間のコンフィグレーション

状況によっては、最小待機時間よりも長いFS SafetyNET p待機時間を使ってプロジェクトをコンフィグレーションすることが推奨される場合があります。たとえば、FS-Rx接続の無効化と再有効化が頻繁に行われる場合などです。PSS 4000プロジェクト (PAS4000) のプロジェクトマネージャーでは、以下の範囲でFS SafetyNET p待機時間をコンフィグレーションできます。

▶ FS SafetyNET pの最小待機時間

このコンフィグレーションでは、値はFS SafetyNET p待機時間の最小許容値を下回ってはなりません。

FS SafetyNET pの最小待機時間 $t_{\text{WaitTime_min}}$ は、次のように計算されます。

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

$t_{\text{WaitTime_min}}$:	PSS 4000プロジェクトのFS SafetyNET pの最小待機時間
$\max (t_{\text{Task}}; t_{\text{MBUS}}; t_{\text{ExtCo}}; t_{\text{SNp}})$:	計算には最大の値が使用される
t_{Task} :	PSS 4000プロジェクトの最大 t_{Task} を持つFSタスクのタスクサイクル時間
t_{MBUS} :	PSS 4000プロジェクトの最大 t_{MBUS} を持つFSモジュールバスのモジュールバスサイクル時間
t_{ExtCo} :	PROFIsafe通信のみ: PROFIsafe通信中の、出力データ領域から外部出力データへの周期的通信に対する、PSS 4000プロジェクト内の最長サイクル時間 t_{ExtCo}
t_{SNp} :	プロジェクト間SafetyNET p RTFN通信中の、PSS 4000プロジェクトのFS SNp-Tx接続 (FSデータ送信) の周期的な開始の最長サイクル時間
t_{BasicSNp} :	プロジェクト内の既存のSNp送信データに対する周期テストの基本サイクル時間

▶ FS SafetyNET pの最大待機時間

FS SNp待機時間をコンフィグレーションする場合、FS SafetyNET p待機時間の最大許容値を超えてはなりません。

最大FS SNp待機時間: 60000ms

8.5.3 データ転送の制限

データ転送の制限は、次のデータによって決まります。

▶ ST SafetyNET pシステムセクションあたりの最大ST接続数

ST-Tx接続の最大数は、ST SafetyNET pシステムセクションで管理できるST送信データ領域の数で決まります。ST送信データ領域の数は、常にST受信データ領域の数に対応しています。

ST SafetyNET pシステムセクションが管理できる送受信データ領域の数は、使用するPSS 4000装置(制御システムPSSu PLC、制御システムPSSu multi、分散型システムPSSu I/Oなどのヘッドモジュール)によって決まります。使用している装置の取扱説明書の情報を参照してください。

ST-Tx接続は1つのPSS 4000装置に対して、または複数のPSS 4000装置のSafetyNET pシステムセクションに対して確立できます。

例: 制御システムPSSu PLC:

制御システムPSSu PLCのST SafetyNET pシステムセクションは、最大64のST-Tx接続で送信データ領域を管理できます。つまり、同様に最大64のST-Tx接続で受信データ領域を管理できます。

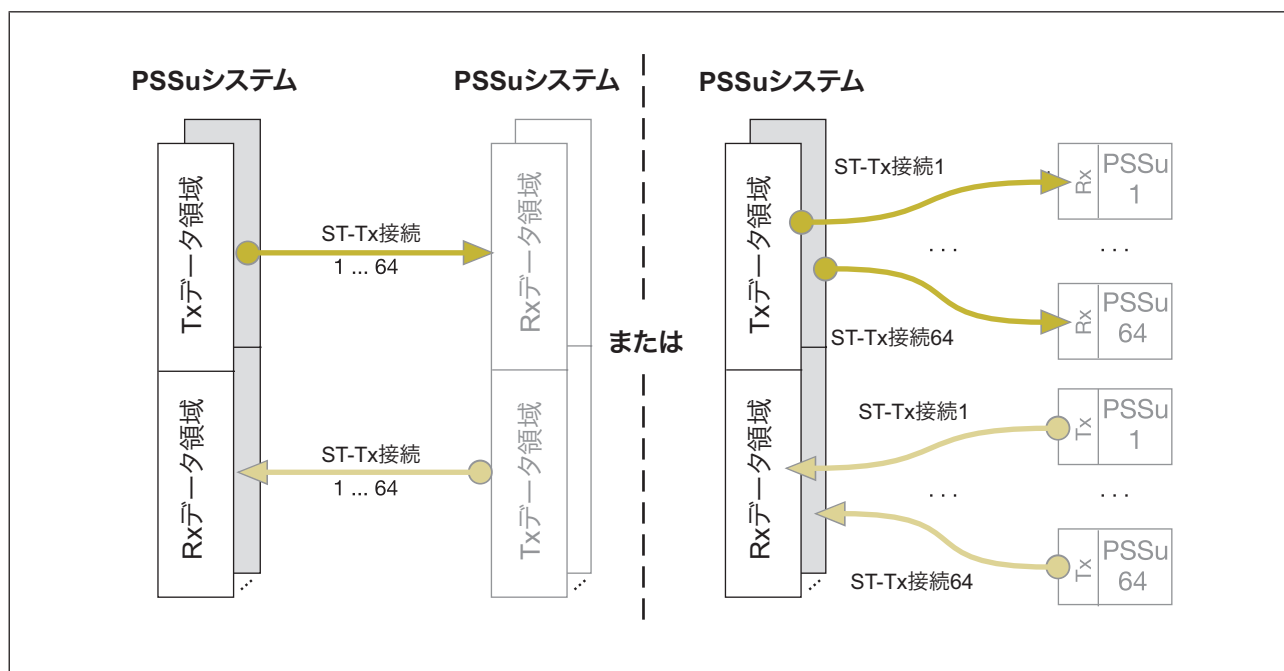


図: 制御システムPSSu PLCでのST-Tx接続の最大数 (例としてPSSuシステム)

▶ FS SafetyNET pシステムセクションあたりの最大FS接続数

FS-Tx接続の最大数は、FS SafetyNET pシステムセクションで管理できるFS送信データ領域の数で決まります。FS送信データ領域の数は、常にFS受信データ領域の数に対応しています。

FS SafetyNET pシステムセクションが管理できる送受信データ領域の数は、使用するPSS 4000装置(制御システムPSSu PLC、制御システムPSSu multi、分散型システムPSSu I/Oなどのヘッドモジュール)によって決まります。使用している装置の取扱説明書の情報を参照してください。

FS-Tx接続は1つのPSS 4000装置に対して、または複数のPSS 4000装置のFS SafetyNET pシステムセクションに対して確立できます。

例: 制御システムPSSu PLC:

制御システムPSSu PLCのFS SafetyNET pシステムセクションは、最大64のFS-Tx接続で送信データ領域を管理できます。つまり、同様に最大64のFS-Tx接続で受信データ領域を管理できます。

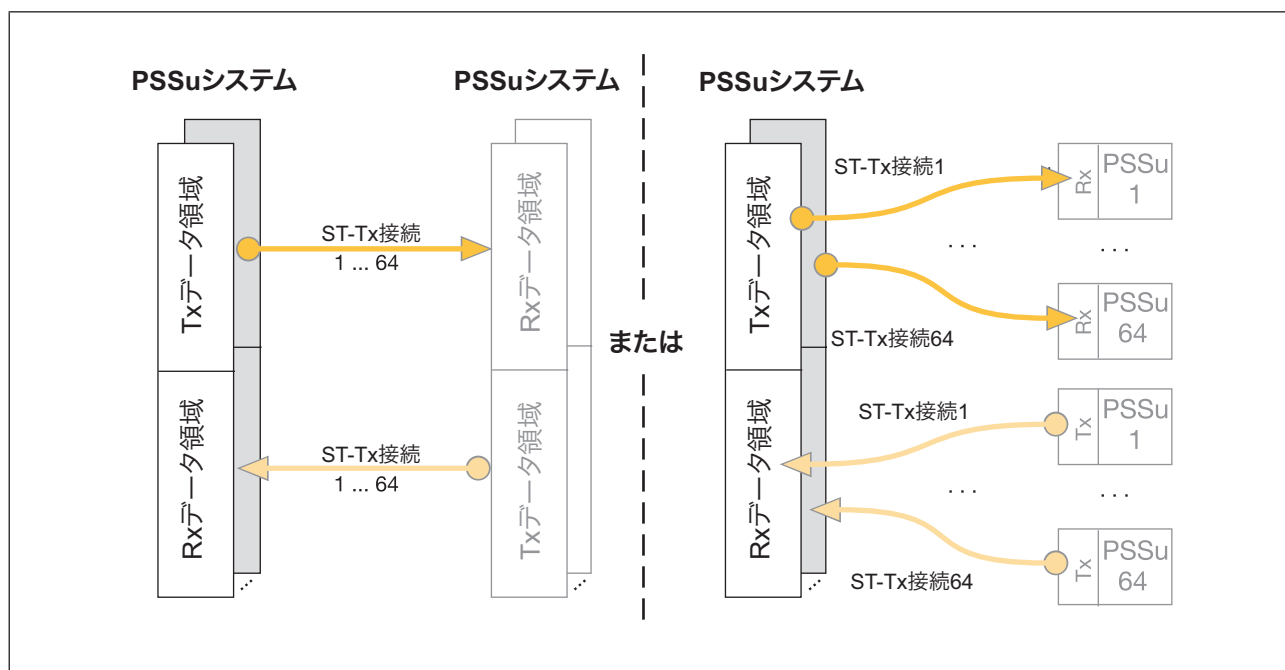


図: 制御システムPSSu PLCでのFS-Tx接続の最大数 (例としてPSSuシステム)

▶ 接続あたりの最大Byte数

接続あたりのByte数は、使用するPSS 4000装置によって決まります。

- Txデータ領域から送信されるSTプロセスデータ:
ST-Tx接続あたり246 Byte
- Rxデータ領域で受信するSTプロセスデータ:
ST-Tx接続あたり246 Byte
- Txデータ領域から送信されるFS処理データ:
FS-Tx接続あたり115 Byte
- Rxデータ領域で受信するFS処理データ:
FS-Tx接続あたり115 Byte

▶ 接続あたりの変数の最大数

接続で転送できる変数の数は、ビット要件により異なります。変数のビット要件は、データ型と使用可能な拡張 (有効なビット) により異なります。

接続で転送できる最大Byte数を超過して変数を転送する場合、変数は追加的な接続に自動的に分割されます。この処理は、データの一貫性に影響を及ぼすことがあるので注意してください ([「データの一貫性」](#) [300]を参照)。

– 基本データ型の変数

ビット要件は変数のデータ型により異なります (例: BOOL/SAFEBOOL: 1ビット、WORD/SAFEWORD: 16ビット)。

– 直接的な派生データ型の変数

ビット要件は、変数の派生元である基本データ型のビット要件に対応しています。

– ARRAYデータ型の変数

ビット要件は各ファンクションに必要なビット数の合計に対応しています。たとえば、1次元配列のビット要件は、ファンクションのビット要件にファンクション数を乗算して計算できます。
データの一貫性を保つために、配列のファンクションは異なる接続に分割されません。

– STRUCTデータ型の変数

ビット要件は各ファンクションに必要なビット数の合計に対応しています。
データの一貫性を保つために、構造のファンクションは複数の接続に分割されません。

– VALID拡張の変数

変数のビット要件は1ビット単位で増加します。有効ビットは一度だけ転送されます。これはI/Oデータ型の変数にも適用されます。



情報

ここに記載された制約は、プロジェクト内SNp通信に制限なく適用されます。

ここに記載されたいくつかの制約とは対照的に、[PSS 4000プロジェクト間I/Oマッピングのルール \[296\]](#) に記載されている情報は、プロジェクト間SNp通信に適用されます。

8.6 プロジェクト内SafetyNET p通信

プロジェクト内SNp通信は、すべてが同一プロジェクト内にあるPSS 4000装置間の、FS/ST SafetyNET pを介したデータ交換です。

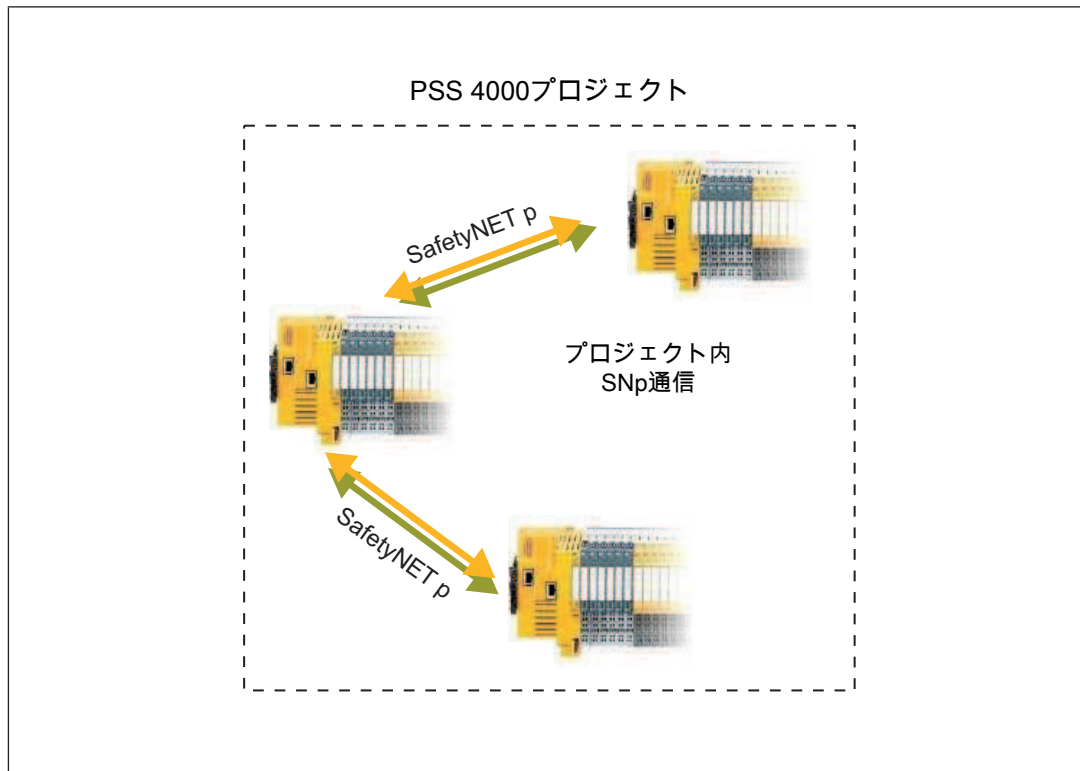


図: プロジェクト内SafetyNET p通信 (原理)

プロジェクト内SafetyNET p通信についての詳細は[PSS 4000プロジェクト内のI/Oマッピングルール](#)  [292](#)を参照してください。

8.7 プロジェクト間SafetyNET p通信

プロジェクト間SNp通信とは、FS/ST SafetyNET pを介して、異なるPSS 4000プロジェクトにあるPSS 4000装置間でデータを交換することです。

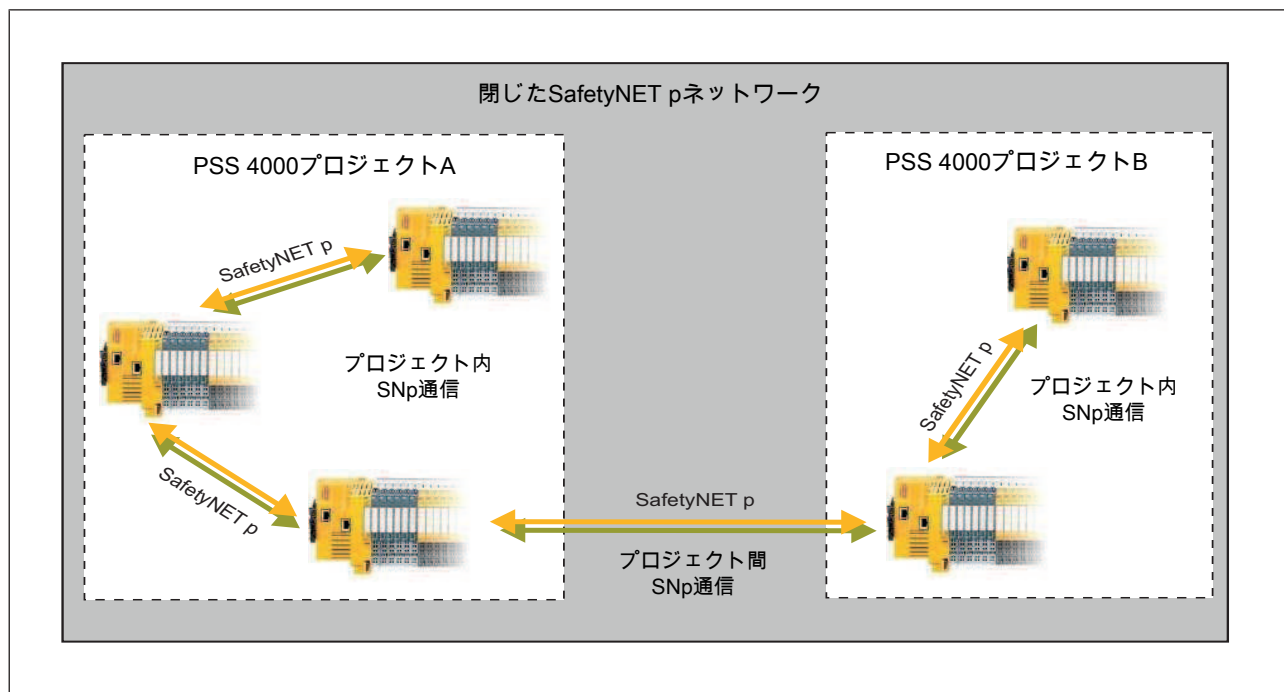


図: プロジェクト間SafetyNET p 通信 (原理)

プロジェクト間SNp通信を使用することにより、各PSS 4000プロジェクトは、他のPSS 4000プロジェクトの出力PI変数を読み取ることができ、他のPSS 4000プロジェクトの入力PI変数に書き込むこともできます。個々のPSS 4000プロジェクトは、この目的のためにPAS4000内に作成されます。PSS 4000プロジェクトのPI変数間のI/Oマッピングは、ソフトウェアツールPASconnectで行います。次に、それぞれ個別のPSS 4000プロジェクトをビルドして装置にダウンロードする必要があります。

プロジェクト間SafetyNET p通信についての詳細は[PSS 4000プロジェクト間I/Oマッピングのルール](#) [296]を参照してください。

プロジェクト間SNp通信では、閉じているSafetyNET pネットワーク内に1つのPASconnectプロジェクトのみが存在できます。PASconnectプロジェクトと同じ閉じているSafetyNET pネットワーク内のすべてのSafetyNET p対応装置は、PASconnectプロジェクトのサブプロジェクトに属している必要があります。これは装置がSafetyNET p経由で他の制御プロジェクトとデータを交換しない制御プロジェクトに属している場合も適用されます。

SafetyNET pデータのトラフィックがネットワーク境界を超えないことが保証される場合、SafetyNET pネットワークは「閉じている」とみなされます。

例: 閉じているSafetyNET pネットワーク内に3つの制御プロジェクトがあります。制御プロジェクト1は、SafetyNET p経由で制御プロジェクト2とデータを交換します。制御プロジェクト3はSafetyNET p対応装置を含んでいますが、SafetyNET p経由で他の制御プロジェクトとデータを交換しません。この場合、3つの制御プロジェクトのすべてが同じPASconnectプロジェクトのサブプロジェクトとしてコンフィグレーションされている必要があります。

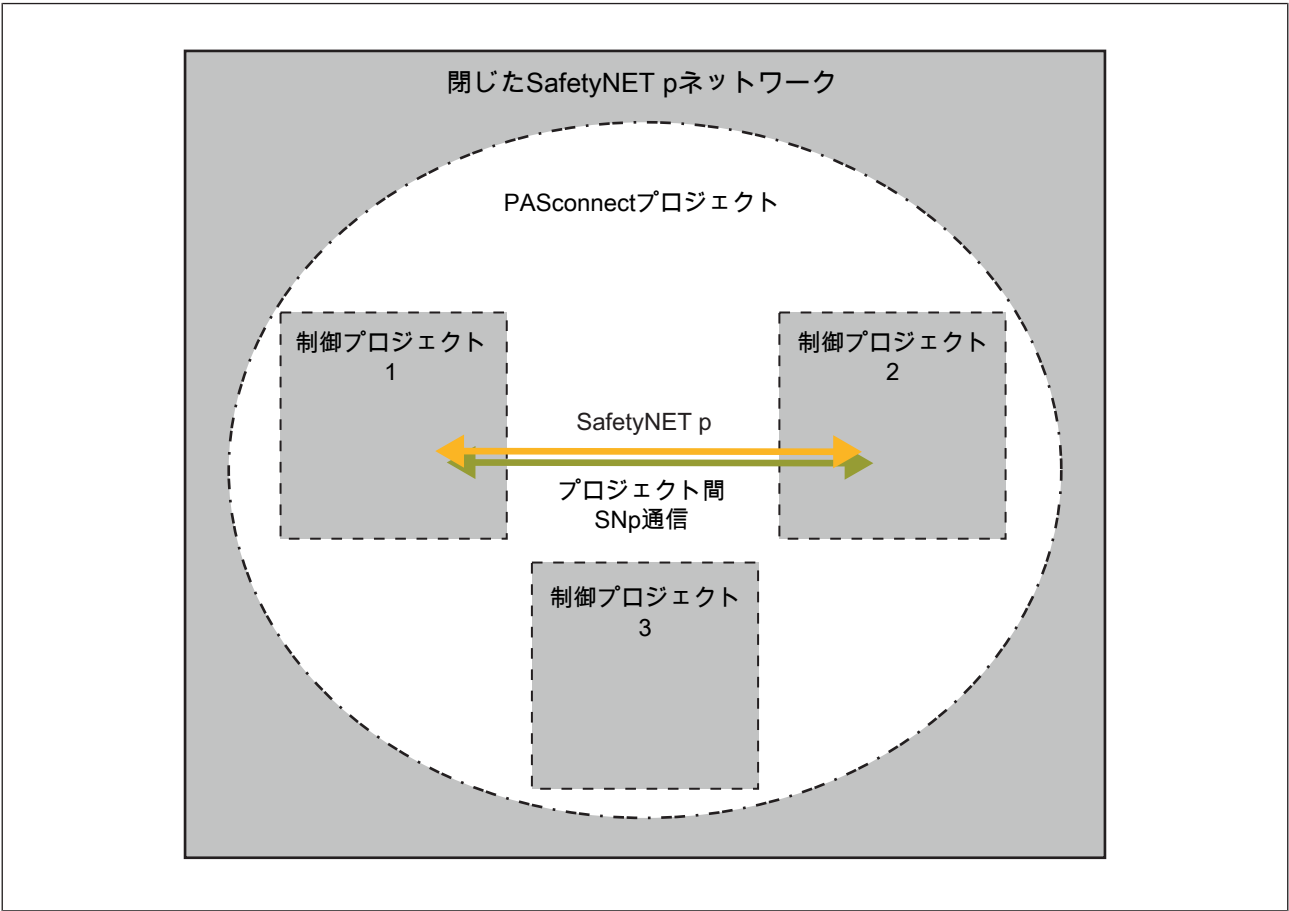


図: プロジェクト間のSnp通信のネットワーク要件

8.8 FSハートビート (FS-HBT)

FSハートビート (FS-HBT) は、FS Snp通信中に通信相手 (リモート装置) のアクセス可否を継続的に確認するのに使用される制御信号です。PSS 4000装置は、各サイクルの一部としてFSハートビートをそれぞれのリモート装置に送信し、一定時間内にリモート装置からFS-HBT応答が返ってくることを確認します。



情報

多くのプロジェクトでは、FSハートビートをコンフィグレーションする必要はありません。既定のコンフィグレーションで十分です。そのため、「FSハートビート」の問題への対処は必須ではありません。FSハートビートの既定のコンフィグレーションは、関連する専門知識を有している場合にのみ変更してください。

次の関係に注意してください。

- ▶ FSハートビートのサイクル時間によって、異常検出時間が決まります。つまり、エラーによって最大伝送遅延時間 t_{DelaySnp} を超過した場合、FSハートビートのサイクル時間に基づいて、これが認識および検出されます。従って、FSハートビートのサイクル時間が長いと、異常検出時間が長くなります。
- ▶ FSハートビートのサイクル時間に小さい値がコンフィグレーションされている場合や、装置がFS Snp経由で多数の装置に接続されている場合、FSハートビートが不要なネットワーク負荷を生成することがあります。
- ▶ FSハートビートのサイクル時間は、常に基本サイクル時間 t_{BasicSnp} の整数倍です。
 - コンフィグレーションによっては、基本サイクル時間 t_{BasicSnp} が変わると、FSハートビートのサイクル時間の値が変わります。この影響は、通常、望ましくありません。プロジェクト間FS Snp通信があるプロジェクトでは、この場合に、PASconnect内の接続されているプロジェクトもリビルドする必要があります。
 - 基本サイクル時間 t_{BasicSnp} は、STタスクを追加または変更した場合も変わる可能性があります。

FSハートビートのサイクル時間

PSS 4000装置がFS Snp接続経由でFSハートビートを送信するサイクル時間のことを、FSハートビートサイクル時間 ($t_{\text{Snp FS-HBT}}$) といいます。FSハートビートサイクル時間は、プロジェクト内のすべてのPSS 4000装置で同じです。

▶ FSハートビートサイクル時間の自動コンフィグレーション

- プロジェクトリンクのないプロジェクト (プロジェクト内FS Snp通信)

プロジェクト内FS Snp通信があるプロジェクトで、FSハートビートサイクル時間に自動コンフィグレーション (既定のコンフィグレーション) を使用した場合、PAS4000はFSハートビートサイクル時間に既定値を使用します。この場合、ユーザはこれ以上何もする必要はありません。

プロジェクト内FS Snp通信には自動コンフィグレーションが推奨されます。
- プロジェクト間FS Snp通信があるプロジェクトで、PASconnectインタフェースのバージョンが1.2.xの場合

このようなプロジェクトでは、FSハートビートサイクル時間は常に自動的に計算されます。

備考: PAS4000は、常にFSハートビートサイクル時間を基本サイクル時間 t_{BasicSNp} の整数倍として計算します。倍数は内部的に定義された値です。つまり、基本サイクル時間を変更すると、必然的にFSハートビートサイクル時間が変更されます。

▶ FSハートビートサイクル時間の手動コンフィグレーション

値の範囲: 6 ms～30000 ms

– プロジェクトリンクのないプロジェクト (プロジェクト内FS SNp通信)

プロジェクト内FS SNp通信があるプロジェクトでは、FSハートビートサイクル時間を手動でコンフィグレーションすることもできます。自動コンフィグレーションをお勧めしますが、場合によってはFSハートビートサイクル時間を手動で定義することも合理的です。例えば、FSハートビートによって生成されるネットワーク負荷を軽減する場合などです。

備考: 手動でコンフィグレーションしたFSハートビートサイクル時間の値は、基本サイクル時間 t_{BasicSNp} を計算するために内部的に使用されます。つまり、これにより、基本サイクル時間 t_{BasicSNp} が変わることがあります。

– プロジェクト間FS SNp通信があるプロジェクトで、PASconnectインタフェースのバージョンが1.3.x/1.4.xの場合

PASconnectインタフェースのバージョン1.3.x/1.4.xでは、FSハートビートサイクル時間は手動でのみコンフィグレーションできます。この手順を使用すると、プロジェクトリンクがあるプロジェクトを、可能な限り影響を生じさせることなく変更することができます。つまり、プロジェクトの変更は、このプロジェクトにリンクされたプロジェクトに影響を与えることはありません。特に、変更によって、装置内部の応答時間と、リンクされたプロジェクトの異常検出時間は変更されることはありません。

備考: 手動でコンフィグレーションしたFSハートビートサイクル時間の値は、基本サイクル時間 t_{BasicSNp} を計算するために内部的に使用されます。つまり、これにより、基本サイクル時間 t_{BasicSNp} が変わることがあります。



重要

FSハートビートサイクル時間 ($t_{\text{SNp FS-HBT}}$) の手動コンフィグレーションについては、次の点に注意してください。

- FSハートビートサイクル時間にコンフィグレーションした値が既定値または推奨値より大きい場合 (エラー)、これが認識および検出されるまでに時間がかかります。
- 手動でコンフィグレーションしたFSハートビートサイクル時間は、基本サイクル時間 t_{BasicSNp} の内部計算に影響します。手動でコンフィグレーションしたFSハートビートサイクル時間によって、基本サイクル時間の値が許容範囲外になった場合、PAS4000はビルドプロセス時にFSハートビートサイクル時間に適する値を提案します。

8.9 SafetyNET pスケジューリングモード

SafetyNET pスケジューリングモードを使用すると、プロジェクト内通信中の接続のサイクル時間に影響を与えることができます。SafetyNET pスケジューリングモードはPAS4000でコンフィグレーションされます。

スケジューリングモードは装置ごとに設定されます。1台の装置のスケジューリングモードの設定は、すべての装置のプロジェクト内Rx接続に影響します。コンフィグレーションは、装置のプロジェクト内FS接続およびプロジェクト内ST接続に適用されます。

装置のスケジューリングモードが変更されると、その結果、装置自体のSafetyNET pコンフィグレーションが変更され、装置にデータを送信するすべての装置のSafetyNET pコンフィグレーションも変更されます (プロジェクト内のTx接続)。

装置のスケジューリングモードを変更すると、その変更が少なくとも1つのプロジェクト内FS接続に影響する場合、チェックサム「FS装置プロジェクト」が影響を受けます。変更されるのは、スケジューリングモードが変更された装置、および装置にFSデータを送信する装置のチェックサム「FS装置プロジェクト」です。



情報

SafetyNET pスケジューリングモードは、FS SafetyNET pの最小待機時間には影響しません。

スケジューリングモード「SafetyNET p応答時間を短縮」

「SafetyNET p応答時間を短縮」が既定のコンフィグレーションです。

このスケジューリングモードは、PSS 4000ファームウェアバージョン1.22.xまでの自動コンフィグレーションに対応しています。

応答時間を短縮するため、このスケジューリングモードのデータは、スケジューリングモード「SafetyNET p受信負荷を低減」よりも頻繁に送信されます。

ただし、データの負荷が高い場合、状況によっては、受信側でデータパケットの処理速度が十分でないことがあります。この場合、受信側の影響を受けるシステムセクションで通信エラーが検出されます。

スケジューリングモード「SafetyNET p応答時間を短縮」は、受信装置が過負荷の場合にのみ選択解除する必要があります。

スケジューリングモード「SafetyNET p受信負荷を低減」

受信側のデータ負荷を管理するために、このスケジューリングモードのデータは、スケジューリングモード「SafetyNET p応答時間を短縮」よりも送信頻度が低くなります。ただし、このスケジューリングモードでは応答時間が長くなります。

注: このスケジューリングモードを使用するには、装置のSTリソースとFSリソースにそれぞれ最大1つのタスクが必要です。

8.10 ルーターのないネットワーク

プロジェクトに対してSafetyNET pネットワークを設計する際には、多数の規則を守る必要があります。ルータを使用していない場合に適用される事項:

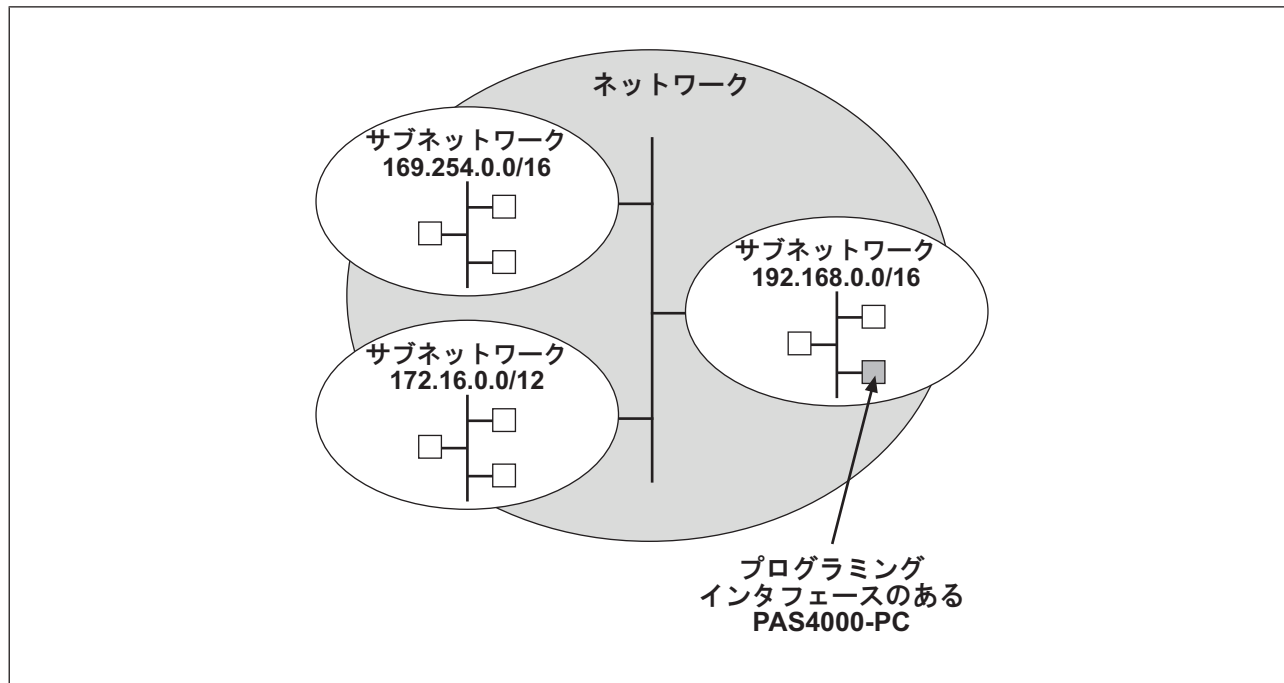


図: ルータを使用しないEthernetネットワークの例

▶ スキャン

PAS4000には、SafetyNET pサブスクリバを検出するさまざまなオプションがあります (ネットワークのスキャン、プロジェクトのスキャンなど)。

- 「ネットワークのスキャン」プロセスでは、プログラミングインタフェースと同じネットワークアダプタに接続されたネットワーク内のPSS 4000専用／対応装置をすべて検出します。
- 「プロジェクトのスキャン」プロセスでは、プロジェクト内でコンフィグレーションされており、プログラミングインタフェースと同じネットワークアダプタに接続されたネットワーク内に配置されたPSS 4000専用／対応装置をすべて検出します。

▶ PAS4000による装置の命名

装置の命名は、スキャンプロセス中に見つかった装置で実行できます。

▶ 処理データ交換

プロジェクト内のPSS 4000装置間の処理データ交換を有効にするには、これらの装置が同じサブネットワーク内にある必要があります。スイッチとハブは相互に接続できます。

▶ その他のオンラインアクション

その他のオンラインアクションには、スキャンと装置の命名を除くすべてのオンラインアクションが含まれます。例:

- プロジェクトの直接ダウンロード
- 診断 (診断リスト、診断ログ)
- デバッグ (表示、制御、変数の強制、オンラインモニタなど)

これらのオンラインアクションを実行できるようにするには、PAS4000-PCのプログラミングインタフェースが、プロジェクトのPSS 4000装置と同じサブネットワークになければなりません。スイッチとハブは相互に接続できます。

8.11 ルーターのあるネットワーク

ルーターは、Ethernet上で2つの異なるEthernetネットワークを接続します。EthernetネットワークはNet-ID (IPアドレスの一部) で識別されるため、Net-IDはサブスライバが同一ネットワーク内にあるかどうか、またはデータパケットを転送対象のルーターに転送する必要があるかどうかを判別するために使用できます。

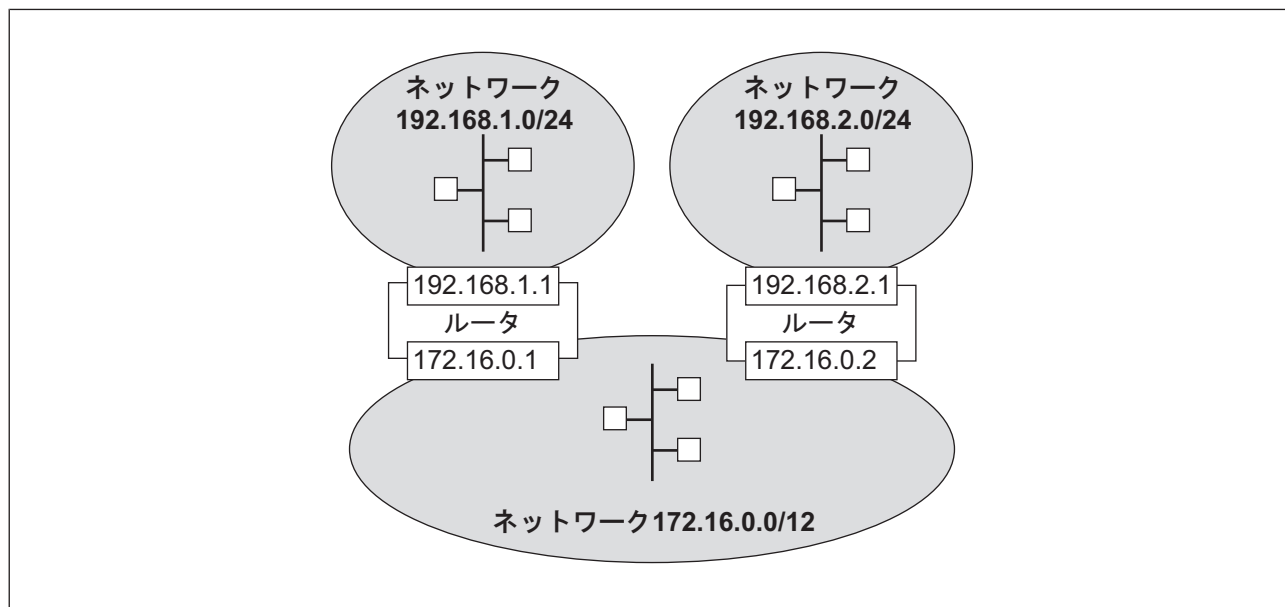


図: ルーターを用いたEthernetネットワーク接続例

SafetyNET pネットワークでルーターを使用している場合の注意点:

▶ ルーターの要件

SafetyNET pネットワークで利用できるルーターは、最大転送ユニット (MTU) が少なくとも302 Byteのものに限られます。

▶ スキャン

PAS4000には、SafetyNET pサブスライバを検出するさまざまなオプションがあります (ネットワークのスキャン、プロジェクトのスキャンなど)。

考えられるPSS 4000装置すべてを世界中から検索することがないよう、PAS4000のスキャン機能はIPマルチキャスト (マルチキャストアドレス224.0.23.61) に基づいて動作します。「Time To Live」の既定の設定 (TTL = 1) により、PAS4000を持つルーターへのスキャンが行われなくなります ([「システム変数 PILZ_SNP_MULTICAST_TTL」](#) [120] を参照)。

▶ PAS4000を用いた装置命名

ネットワークがルーターを通じて接続されている場合、「Time To Live」の既定の設定 (TTL = 1) では、装置の命名を中央に集中せず個々のネットワークから直接実行する必要があります ([「システム変数 PILZ_SNP_MULTICAST_TTL」](#) [120] を参照)。

▶ 処理データ交換

ルーターが正しくコンフィグレーションされていれば (ルーティングテーブルなど)、処理データ交換はPSS 4000専用/対応装置でも動作します。これはPAS4000-PCの側から見ると、ルーターの背後のネットワークにあります。

▶ その他のオンラインアクション

ルータが正しくコンフィグレーションされていれば (ルーティングテーブルなど)、その他のオンラインアクションは PSS 4000 専用 / 対応装置でも動作します。これは PAS4000-PC の側から見ると、ルータの背後のネットワークにあります。その他のオンラインアクションとは、スキャンと装置の命名を除くすべてのオンラインアクションです。例:

- プロジェクトの直接ダウンロード
- 診断 (診断リスト、診断ログ)
- デバッグ (表示、制御、変数の強制、オンラインモニタなど)



重要

MS Windows はインターネット上での「プライベートIPアドレス」 (プライベートIP) へのルーティングをサポートしていないため**注意してください**。プライベートIPアドレスはプライベートIPアドレス範囲にあり、IANAによって次のように定義されています。

- 10.0.0.0 ... 10.255.255.255
10.0.0.0/8: プライベートネットワーククラスAとサブネットマスク255.0.0.0
- 172.16.0.0 ... 172.31.255.255
172.16.0.0/12: プライベートネットワーククラスBとサブネットマスク255.240.0.0
- 192.168.0.0 ... 192.168.255.255
192.168.0.0/16: プライベートネットワーククラスCとサブネットマスク255.255.0.0

8.11.1 システム変数PILZ_SNP_MULTICAST_TTL

Time To Live (TTL) は、インターネットプロトコル (IP) のパラメータです。TTLはIPパケットのオペレーティング範囲の判断に使用できます。

送信者からターゲットの途中でIPパケットを渡すルータは、IPパケットのTTL値をそれぞれ1つずつ減らします。TTL値が0になったIPパケットはルータによる転送が行われず、破棄されます。

スキャンと装置の命名は、関連するTTLの設定に応じて実行されます。これら2つのアクションに対するTTLの値は、システム変数PILZ_SNP_MULTICAST_TTLを通じてPAS4000 PVでコンフィグレーションできます。ルータの正しいコンフィグレーションに加えて、PILZ_SNP_MULTICAST_TTLの設定は、装置のスキャンおよび命名時に、PAS4000 PCから見たルータの背後にあるPSS 4000装置に到達するための重要な要素です。



重要

PILZ_SNP_MULTICAST_TTLが1より大きい場合、他のネットワークのすべてのルータに対して介入できてしまうので注意が必要です。これにより、ネットワークの安全が脅かされる危険があります。

システム変数PILZ_SNP_MULTICAST_TTLのコンフィグレーションには、次のことが適用されます。

- ▶ システム変数PILZ_SNP_MULTICAST_TTLが使用できない場合の既定TTL: 1
システムが世界中のすべての潜在的なPSS 4000装置を検索しないようにするため、デフォルト設定ではルータ間のスキャンおよびルータ間の装置の命名が禁止されています。
PAS4000でデフォルト設定を使用する場合は、システム変数PILZ_SNP_MULTICAST_TTLを作成およびコンフィグレーションする必要はありません。起動時にシステム変数がない場合、PAS4000では自動的にデフォルト設定が使用されます。
- ▶ システム変数は、Windows 10オペレーティングシステムの**環境変数**でコンフィグレーションします。
Windowsの検索フィールドに「**環境変数**」と入力し、[環境変数を編集] を選択します。ダイアログボックスで、ユーザ自身のアカウントおよびすべてのユーザの変数をコンフィグレーションすることができます。すべてのユーザのシステム変数をコンフィグレーションするには、管理者権限が必要です。
- ▶ PILZ_SNP_MULTICAST_TTLの有効なコンフィグレーション: 1～64の範囲の整数値
 - コンフィグレーションが無効な場合 (文字、値が整数でない場合など)、PILZ_SNP_MULTICAST_TTLにはデフォルト設定が使用されます。
 - 64より大きな値がコンフィグレーションされた場合、PILZ_SNP_MULTICAST_TTLには64が使用されます。
- ▶ システム変数PILZ_SNP_MULTICAST_TTLの作成／変更は、PAS4000 PCを再起動するまで有効になりません。



重要

OPCサーバがPAS4000と同じPCにインストールされている場合、OPCサーバはインストール中にシステム変数PILZ_SNP_MULTICAST_TTLを64に設定します。

8.12 無線LAN経由のSafetyNET p

原則として、無線LAN規格に従ったワイヤレスネットワークでSafetyNET pを使用することができます。ただし、最初にさまざまな分析を実行して追加のサポート手段を講じることなくケーブルベースのネットワークを無線LANに置き換えることはお勧めしません。特に、安全なプロトコルを使用している場合は、無線LANはすべてのアプリケーションに適しているとは限らないことを考慮する必要があります。

SafetyNET pは、安全関連データ用の安全データチャンネル (FS通信) を備えています。FS通信の伝送エラーは常にSafetyNET pで検出され、システムセクションが適切なオペレーティング状態に切り替えられます。伝送面で可用性が低い無線LANでSafetyNET pが使用される場合、無線LAN通信でエラーが頻繁に発生すると、FS通信が無効化されることがあります。

次の点に注意してください。

- ▶ 無線LANを使用する前に、専門家による詳細なオンサイト分析の実施を手配する必要があります。この分析では、十分にエラーがない安定した無線伝送が可能かどうかを明確にする必要があります。この際に分析する必要があるのは主に、無線LAN伝送の可用性です。分析では、特に受信エリアでの可用性に注意する必要があります。
- ▶ 可能であれば、安全プロトコルの伝送に適した装置のみを無線LANルータとして使用してください。たとえば、常に高い信号品質とシームレスで冗長な無線LAN接続で動作する装置 (たとえばPRP) がこれに該当します。
- ▶ 可能な場合は、SafetyNET pプロトコルバージョン2を使用してください。無線LAN接続では通信エラーがより頻繁に発生することがあるので、SafetyNET pプロトコルバージョン2に優位性があります。バージョン2では、通信エラー後のFS通信の再開がサポートされています ([「SNpプロトコルバージョン2: 通信エラーおよびFS通信の再開」](#) [104] を参照)。
- ▶ SafetyNET pプロトコルバージョン2を使用し、エラーの原因を回復した後にFS通信を再開した場合でも、状況によっては、これらのエラーがPSS 4000装置の可用性に著しい影響を与える可能性があります。エラーが存在する間、影響を受けるプロセスデータで有効ビット = FALSEによる代入値が使用されることに注意してください。その結果、影響を受けるハードウェア出力が安全状態に切り替えられる場合があります。ユーザプログラムでは、たとえば、出力データを2倍以上の頻度で送信することで、これらの潜在的な悪影響に適切に対応する必要があります。
- ▶ ケーブルベースのネットワークと比較して、無線LANでは一般的に帯域幅が狭く遅延時間が長くなるので、無線LANを介したデータ送信のタスクサイクル時間は、同等のケーブルベースのネットワークよりも長く設定することが必要な場合があります。

個別のケースについて、これは次のようなことを意味する場合があります。

無線LAN経由のデータ送信の中断を避けるために、SafetyNET pのサイクル時間 (t_{SNp}) を増やす必要があります。ただし、SafetyNET pのサイクル時間は、PSS 4000プロジェクト内で直接設定することはできません。タスクサイクル時間を介して間接的にのみ変更することができます。

SafetyNET pのサイクル時間が、モジュールバス (t_{MBUS}) に対して計算されたサイクル時間よりも大きくなると、コントローラのモジュールバスのI/Oデータを、別のコントローラのブロックのPI変数に直接マッピングすることはできません。I/Oマッピングは2つの制御システムのPI変数の間でのみ行うことができます。

SafetyNET pのサイクル時間は、ブロックのタスクサイクル時間 (t_{Task}) を介して間接的に増やすことができます。詳細については、「[応答時間に影響を与える時間](#)」 [457] を参照してください。

8.13 SafetyNET pで最適化されるアベイラビリティ

SafetyNET pはイーサネットベースのバスシステムです。このため基本的には、イーサネットベースの会社ネットワーク（オフィス領域、構築サービス、物流）と同じネットワークで、オートメーション領域（機械環境、フィールドレベル）を運用することができます。ただし、いくつかの理由により、この2つの領域は混在させないことを強くお勧めします。

セキュリティ上の潜在的な問題とは別に、この2つの領域では、非常に異なる通信要件を考慮する必要があります。たとえば、オフィスでは大量のデータを管理するのが一般的であるのに対し、オートメーション領域で重要なのは主にリアルタイムの動作です。

イーサネットでは、サブスクライバが増えるごとにネットワークの負荷が高まり、さらにオフィスの装置は一般的にオートメーションの装置よりも高いネットワーク負荷を生み出す傾向があります。したがって、2つの領域は分離しておく必要があります。これは、確立されたさまざまな戦略（例：両方の領域の物理的な分離、ネットワークの区分化）で対応できます。

ネットワークをSafetyNET pで構築する場合、計画段階であっても慎重なアプローチが求められます。ネットワークの負荷を軽減してアベイラビリティを高めるには、特に以下の要件に対応して戦略を策定する必要があります。

- ▶ オートメーション領域 (SafetyNET p) の境界線の明確化 (オートメーション領域を上位レベルの会社ネットワーク (オフィス領域) に統合する必要がある場合)。
- ▶ 継続的または散発的に高いネットワーク負荷を発生させる装置のオートメーション領域 (SafetyNET p) からの除外。この対応ができない場合は、ネットワークの負荷がSafetyNET pに影響を及ぼさないように、これらの装置を少なくともSafetyNET pから区分する必要があります。

この方法では、たとえば、以下のような装置が対象になります。

- プリンタ
- ビデオカメラ
- ブロードキャストまたはマルチキャストテレグラムを頻繁に転送する装置



情報

制御システムのアベイラビリティを高めるために、継続的または散発的に高いネットワーク負荷を発生させる装置は、制御システムと同じネットワークで運用しないことをお勧めします。極端な状況では、これらの装置のネットワーク負荷により、FSの重大エラー（「[オペレーティングステータス「Major FS error」](#)」[\[344\]](#)」を参照）が発生する可能性もあります。

この推奨に従わなくても、制御システムの安全機能に影響が及ぶことはありません。

8.14 同一プロジェクト名を持つプロジェクトとのデータ交換の防止

PSS 4000プロジェクト間のデータ交換は、プロジェクト間SNp通信が使用されている場合にだけ許可されます。プロジェクト間SNp通信では、PASconnectによりPSS 4000プロジェクトが同一のプロジェクト名を持つことが阻止されます。

ただし、FSデータが処理されるPSS 4000プロジェクト間に実際のデータ交換がない場合、これらのプロジェクト名が同一であるという状況も発生します。そのようなプロジェクトが同じSafetyNET pネットワーク内で運用された場合、追加措置を取らない限りPSS 4000はデータ交換中の相互接続を防止できません。

同一のプロジェクト名を持つPSS 4000プロジェクトとプロセスFSデータは、閉じられたSafetyNET pネットワーク内で操作する必要があります。SafetyNET pデータのトラフィックがネットワーク境界を超えないことが保証される場合、SafetyNET pネットワークは「閉じている」とみなされます。



情報

PSS 4000プロジェクトが閉じられたSafetyNET pネットワーク内にあることを保証できない場合、これらのプロジェクトが同一のプロジェクト名を持つことはできません。プロジェクトの名前を変更します。

9 外部通信

9.1 外部通信の通信モード

外部通信とは、出力データ (外部出力データ) がPSS 4000プロジェクトを離れ、入力データ (外部入力データ) がPSS 4000プロジェクトに入るデータ交換のことです。これは、さまざまな通信モードのサポートを受けて実現されます。PSS 4000装置でサポートされる通信モードは、装置によって異なります。詳細は、関連するPSS 4000の取扱説明書に記載しています。

外部通信の通信モードは「フィールドバス」と「IP接続」に分けられます。

9.1.1 フィールドバス

「フィールドバス」という用語には、標準化されたプロトコル (例: PROFIBUS-DP) を搭載した産業向けフィールドバスも含まれます。フィールドバス経由の外部通信の場合、PSS 4000装置には、対応するシステムセクション (例: PROFIBUS-DPスレーブシステムセクション) が必要です。物理的な接続は、フィールドバスによって、定義されたインタフェースで確立します。

9.1.2 IP接続

「IP接続」という用語には、IPアドレスを使用するすべてのプロトコルと通信回線が含まれます (Modbus/TCP、Raw TCP、Raw UDP)。このタイプの外部通信では、PSS 4000装置に「IP接続」システムセクションが必要です。物理的な接続は、イーサネットインタフェースで確立します。



重要

IP接続の最大数には、次の規則が適用されます。

「IP接続」システムセクションを使用してPSS 4000装置で管理できるIP接続の最大数は、装置によって異なります。

ただし、「IP接続」システムセクションでは、最大32個のIP接続を管理できます。この上限は、「IP接続」システムセクションで、複数の異なるプロトコルや通信回線を使用している場合にも適用されます。

SafetyNET p接続は「IP接続」システムセクションで**管理されない**ため、この上限に影響しません。



重要

初期インストールの試行時にIP接続を確立できない場合や、IP接続が中断された場合、再インストールは即座には実行されません。

この動作は、IP接続が正常に初期化された結果として発生します。個々の初期化の間で経過する時間は、個々のIP接続のサイクル時間の最大公約数の倍数 (t_{ExtCo}) です。この時間によっては、特定の状況では、すべてのIP接続の復元にかかなりの時間がかかる可能性があります。

UDP経由の外部通信を使用する特別な機能

PSS 4000は、2つの異なるUDP経由の外部通信方法を提供します。

- ▶ Raw UDP
- ▶ ブロックを使用するUDP通信

IP接続システムセクションは両方のオプションに必要です。Raw UDP経由の通信は、IP接続のルールによって規制されますが、ブロックを使用するUDP通信は、独自のルールに従います。

次のテーブルに主な機能の比較を示します。

Raw UDP	ブロックを使用するUDP通信
IP接続システムセクションがあるPSS 4000装置が必要	
最大合計32 IP接続の上限に影響	- - -
t_{ExtCo} を使用した周期的データ転送	ユーザプログラム経由で制御されるデータ転送
PAS4000での接続のコンフィグレーション	PSS 4000システム制御ブロックのコンフィグレーション
- - -	STリソースまたはFSリソースでのPSS 4000システム制御ブロックの使用
コンフィグレーション済みデータ長 (= 固定データ長) を使用した接続経由のデータ転送	PSS 4000システム制御ブロックでコンフィグレーションできるデータ長。可変データ長でデータ転送可能。
使用可能なUDP接続: ▶ ユニキャスト (1:1接続) ▶ ブロードキャスト (1:「全」接続) ▶ マルチキャスト (1:n接続)	使用可能なUDP接続: ▶ ユニキャスト (1:1接続、優先接続タイプ) ▶ ブロードキャスト (1:「全」接続) ▶ マルチキャスト (1:n接続、送信のみ)
PAS4000ではIOマッピングが必要	PSS 4000システム制御ブロックとI/Oデータのマッピング

9.2 Modbus/TCP

Modbusは、ユーザ組織MODBUS-IDAが公開したフィールドバスのオープン規格です。

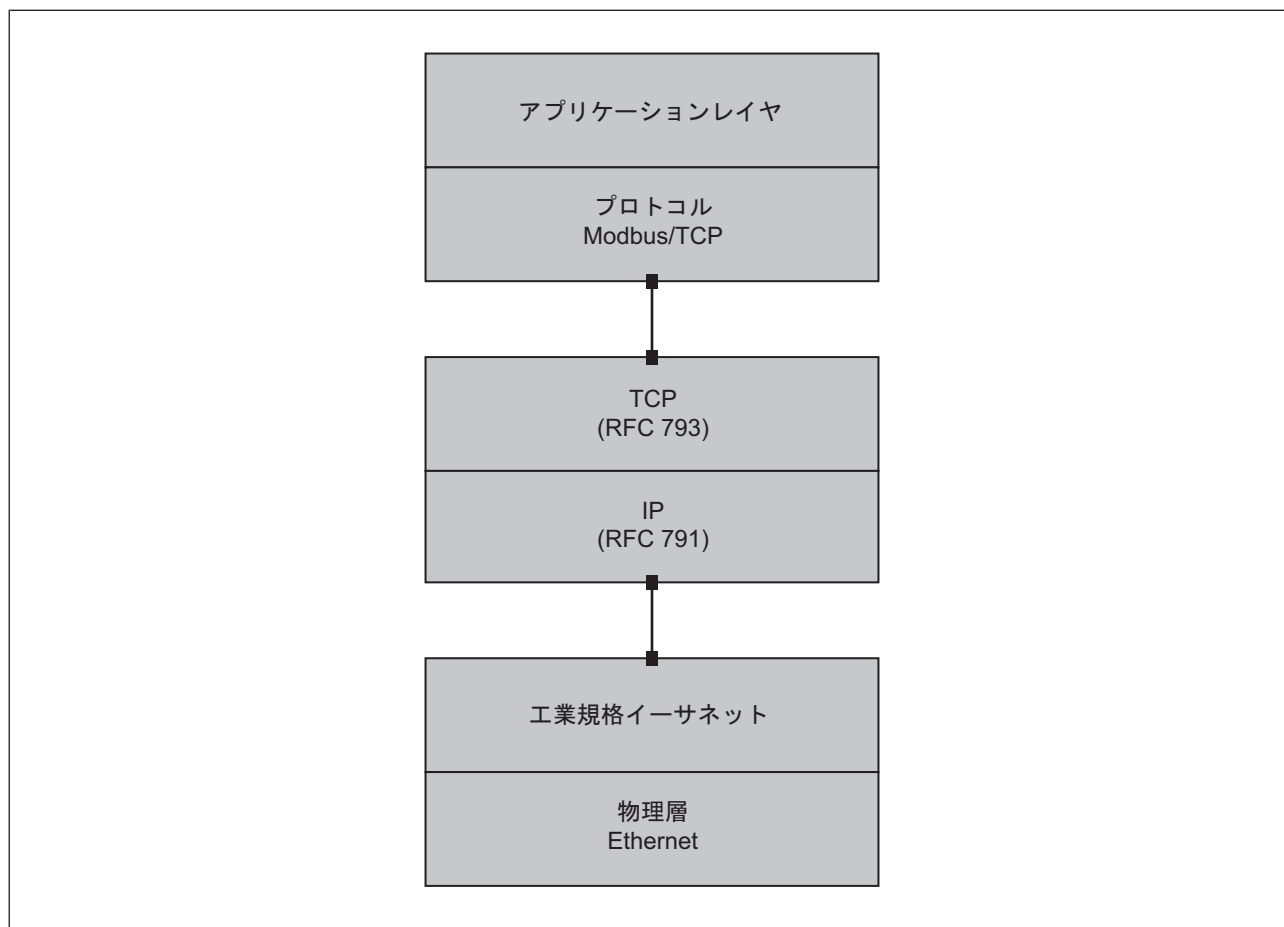


図: Modbus/TCP経由でのPSS 4000の通信

Modbus/TCPは産業用イーサネットに基づくプロトコルです (イーサネット経由のTCP/IP)。これは、クライアント／サーバ通信を使用するプロトコルです。データは、ファンクションコード (FC) を使用して、要求／応答のメカニズムで転送されます。

Modbus/TCPは接続指向のプロトコルです。したがって、ペイロードをModbus/TCP経由で転送するには、2つのModbus/TCPインタフェースの間で、最初に接続を確立する必要があります。接続を開始する側は「クライアント」と呼ばれます。クライアントが接続を確立する通信相手は「サーバ」と呼ばれます。装置の接続でクライアントとサーバのどちらの役割を使用するかは、接続のコンフィグレーション時に定義されます。したがって、サーバ／クライアントの役割は、その特定の接続に対してのみ適用されます。

Modbus/TCP経由でのデータ交換は、非安全関連アプリケーション用のオートメーションシステムPSS 4000で使用できます。

9.2.1 「IP接続」システムセクション

オートメーションシステムPSS 4000の場合、「IP接続」システムセクションがあるすべてのPSS 4000装置で、Modbus/TCP経由の外部通信を使用できます。この場合、PSS 4000装置の「IP接続」システムセクションが、通信に必要なタスクをすべて実行します。例:

- ▶ Modbus/TCP接続の管理
(「[Modbus/TCP接続](#)  128」を参照)
- ▶ Modbus/TCPデータ領域の管理
(「[Modbus/TCPデータ領域](#)  129」を参照)
- ▶ ファンクションコードのサポート
(「[ファンクションコード](#)  130」を参照)
- ▶ データ交換
(「[Modbus/TCPを使ったデータ交換](#)  133」を参照)

9.2.2 Modbus/TCP接続

PSS 4000の「IP接続」システムセクションでは、PSS 4000に対してコンフィグレーションするModbus/TCP接続を管理します。接続は、PAS4000のIP接続エディタでコンフィグレーションされます。

1台のPSS 4000に対して、複数のModbus/TCP接続をコンフィグレーションできます（「[データ転送の制限](#) [136]」を参照）。接続がコンフィグレーションされると、クライアントまたはサーバの役割で接続をコンフィグレーションできます。接続に対してクライアントの役割をコンフィグレーションする場合、この接続はPSS 4000の視点から「クライアント接続」と呼ばれます。それ以外の場合、この接続はPSS 4000の視点から「サーバ接続」と呼ばれます。

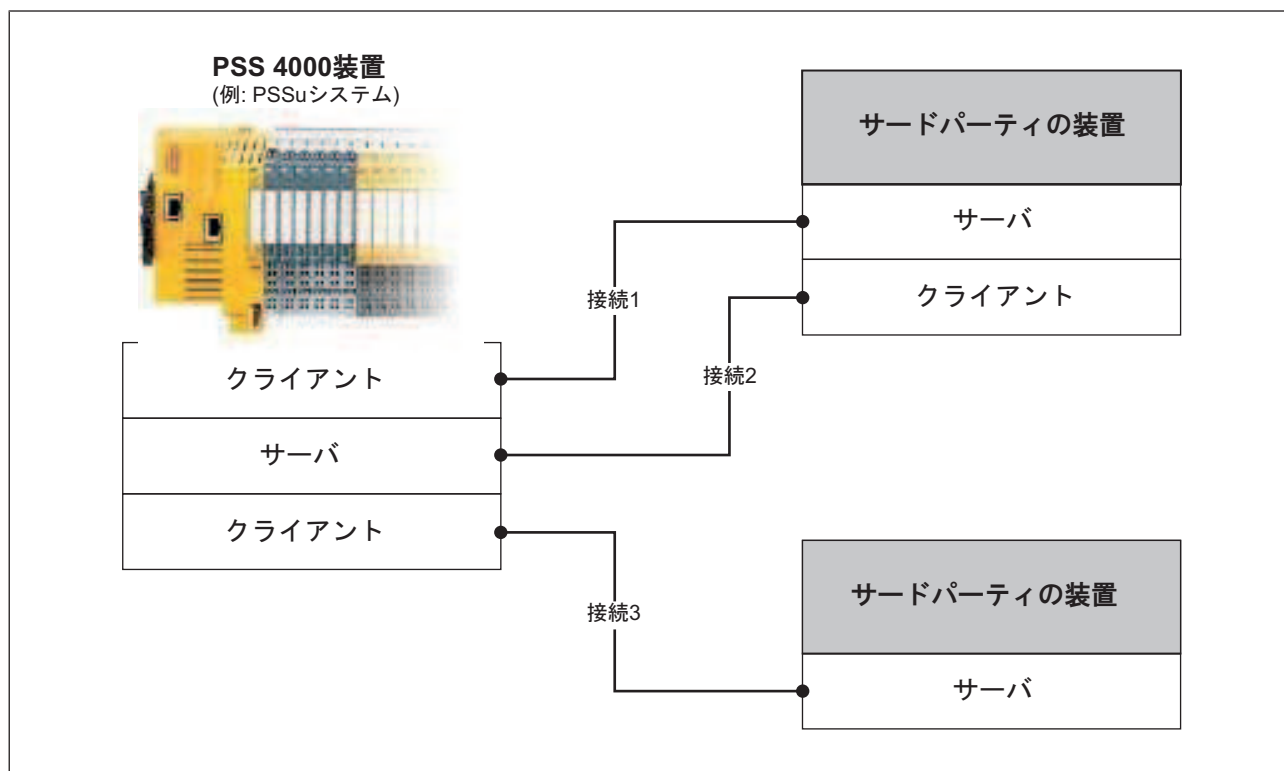


図: Modbus/TCP接続での役割の割り付け (例)

Modbus/TCP接続を経由したデータ交換に対して、ポート番号を定義する必要があります。PSS 4000専用のサーバ接続は、既定の設定としてポート502を使用します。ただし、異なるポート番号がコンフィグレーションされることもあります。



情報

既存のアプリケーション (Telnetなど) には、各アプリケーションがデータ転送時に参照できる固定ポート番号が割り付けられています。0～1023の範囲に予約ポート番号が含まれています。ユーザ独自のアプリケーションでは、これら予約ポートアドレスを使用しないでください。RFC 1700には、予約されたポート番号とユーティリティの詳細なリストがあります。

PSS 4000専用装置では、送信データと受信データが周期的に転送されます。周期的な転送では、接続クライアントとサーバの間で固定的な接続が維持されます。

9.2.3 Modbus/TCPサーバ接続 (サーバ接続)

PSS 4000の「IP接続」システムセクションは、サーバの役割との接続のために、次のModbus/TCPデータ領域をサポートします。

データ領域	Modbusの構文	例
コイル (ビット) 0x00000～0x65535 [読み取り／書き込み]	0x[xxxxx]	0x05888
ディスクリート入力 (ビット) 1x00000～1x65535 [読み取り専用]	1x[xxxxx]	1x05888
入力レジスタ (ワード／16ビット) 3x00000～3x65535 [読み取り専用]	3x[xxxxx]	3x05888
保持レジスタ (ワード／16ビット) 4x00000～4x65535 [読み取り/書き込み]	4x[xxxxx]	4x05888



情報

PSS 4000の「IP接続」システムセクションは、Modbus/TCPデータ領域のアドレス処理を「0」から開始します。他のメーカーの装置では、アドレス処理が「1」から開始されることがあります。

関連メーカーが提供する取扱説明書を参照してください。

9.2.4 ファンクションコード (クライアント接続)

PSS 4000の「IP接続」システムセクションは、クライアントの役割との接続のために、次のファンクションコード (FC) をサポートします。

ファンクションコード	ファンクション	
FC 01	コイルの読み取り	接続クライアントは、たとえば、デジタル入力 (0xからの受信データ) など、データ長 ≥ 1 ビットのビットデータを接続サーバから読み取ります。
FC 02	ディスクリット入力の読み取り	接続クライアントは、たとえば、デジタル入力 (1xからの受信データ) など、データ長 ≥ 1 ビットのビットデータを接続サーバから読み取ります。
FC 03	保持レジスタの読み取り	接続クライアントは、たとえば、アナログ入力、カウンタ入力 (4xからの受信データ) など、データ長 ≥ 1 ワードのワードデータを接続サーバから読み取ります。
FC 04	入力レジスタの読み取り	接続クライアントは、たとえば、アナログ入力、カウンタ入力 (3xからの受信データ) など、データ長 ≥ 1 ワードのワードデータを接続サーバから読み取ります。
FC 05	単一コイルの書き込み	接続クライアントは、たとえば、デジタル入力 (0xへの送信データ) など、データ長=1ビットの1ビットデータに接続サーバで書き込みを行います。
FC 06	単一レジスタの書き込み	接続クライアントは、たとえば、アナログ出力、カウンタ出力 (4xへの送信データ) など、データ長=1ワードの1ワードデータに接続サーバで書き込みを行います。
FC 15	複数コイルの書き込み	接続クライアントは、たとえば、デジタル出力 (0xへの送信データ) など、データ長>1ビットの複数ビットデータに接続サーバで書き込みを行います。
FC 16	複数レジスタの書き込み	接続クライアントは、たとえば、アナログ入力、カウンタ入力 (4xへの送信データ) など、データ長>1ワードの複数ワードデータに接続サーバで書き込みを行います。
FC 23	複数レジスタの読み取り／書き込み	接続クライアントは、テレグラム内で複数ワードデータの読み取りと書き込みを実行します (4xからの受信データおよび4xへの送信データ)。



情報

クライアント接続ごとに、ファンクションコードをコンフィグレーションする必要があります。この操作は、PAS4000のIP接続エディタで実行できます。

9.2.5 ファンクションコードとModbus/TCPデータ領域の関係

クライアント接続の場合、ファンクションコードとModbus/TCPデータ領域の間で、次の関係が存在します。

データ領域 (リモートアドレス)	送信 (クライアント)		受信 (クライアント)
0x コイル (ビット)	データ長	ファンクションコード	ファンクションコード
	= 1ビット	FC = 05 (単一コイルの書き込み)	FC = 01 (コイルの読み取り)
	> 1ビット	FC = 15 (複数コイルの書き込み)	
1x ディスクリート入力 (ビット)	---		FC = 02 (ディスクリート入力の読み取り)
3x 入力レジスタ (ワード)	---		FC = 04 (入力レジスタの読み取り)
4x 保持レジスタ (ワード)	データ長	ファンクションコード	FC = 03 (保持レジスタの読み取り)
	= 1ワード	FC = 06 (単一レジスタの書き込み)	
	> 1ワード	FC = 16 (複数レジスタの書き込み)	
送信データと受信データの両方にエントリがある場合、ファンクションコードFC = 23が使用されます (複数レジスタの読み取り／書き込み)。			

9.2.6 Modbus/TCPを使ったデータ交換

データ転送は、常に接続クライアントが開始します。クライアント接続の場合、データフローの方向 (送信／受信) とデータ領域 (コイル、ディスクリート入力、入力レジスタ、保持レジスタ) は、接続のコンフィグレーション時にファンクションコードの定義で確立されます。またデータ長も、接続のコンフィグレーション時に定義されます。

9.2.6.1 送信データと受信データ

送信時に、接続クライアントは、接続サーバに属しているデータ領域にデータを送信します。これは「データの書き込み」とも呼ばれます。この操作を正常に実行するには、クライアントの役割を持つ装置で目的のデータ宛先をコンフィグレーションする必要があります。

受信時に、接続クライアントは、接続サーバに属しているデータ領域にあるデータを要求し、受信します。これは「データの読み取り」とも呼ばれます。この操作を正常に実行するには、クライアントの役割を持つ装置でデータソースをコンフィグレーションする必要があります。

オートメーションシステムPSS 4000から見ると、受信データは外部入力データと呼び、送信データは外部出力データと呼びます (「[処理データの有効性](#)」[\[281\]](#)を参照)。

関連するデータ領域 (データの宛先、データソース) がサーバで使用可能になっているようにしてください (「[サーバアロケーションテーブル](#)」[\[134\]](#)を参照)。

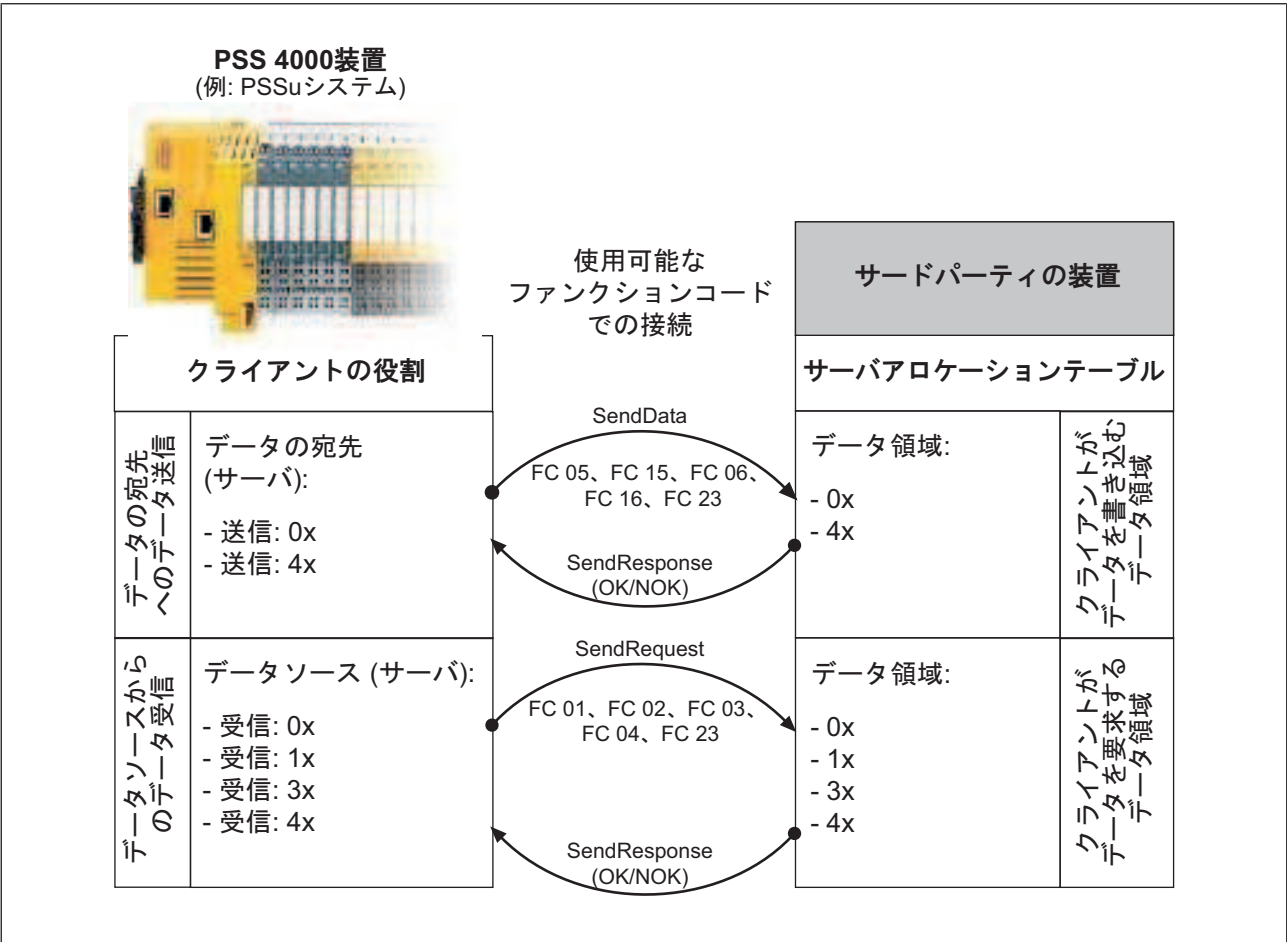


図: 「IP接続」システムセクションがあり、接続のためにクライアントの役割を持つPSS 4000 (原則)

9.2.6.2 サーバロケーションテーブル

接続クライアントで接続サーバにデータを送信し、要求データを接続サーバから受信できるようにするには、関連するデータ領域をサーバ上でコンフィグレーションする必要があります。

サーバ接続では、接続のコンフィグレーション時にサーバロケーションテーブルをコンフィグレーションすることで、この目的を達成します。

次のことが該当します。

- ▶ 接続クライアントは、サーバロケーションテーブルの書き込みデータ領域 (0x書き込み、4x書き込み) にデータを書き込みます。
- ▶ 接続クライアントは、サーバロケーションテーブルの読み取りデータ領域 (0x読み取り、1x、3x、4x読み取り) からデータを要求します。

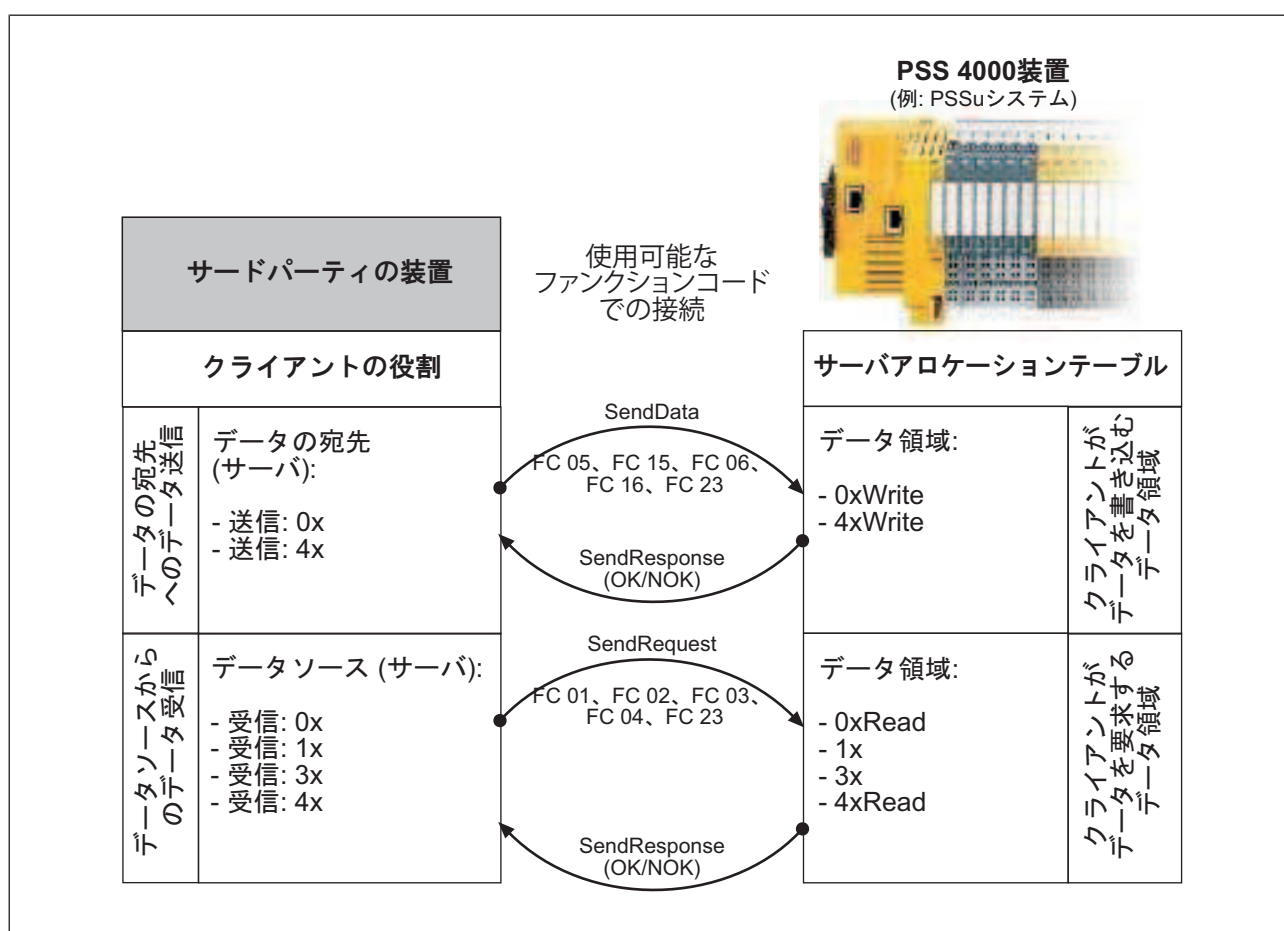


図: 「IP接続」システムセクションがあり、接続のためにサーバの役割を持つPSS 4000 (原則)

PAS4000でサーバロケーションテーブルをコンフィグレーションする場合の制限

各データ領域では、範囲0～ (65536: データ長) の開始アドレスと最大データ長8000を指定できます。

データ領域	開始アドレス	データ長
0x読み取り	0 ... (65536: データ長)	1 ... 8000
0x書き込み		
1x		
3x		
4x読み取り		
4x書き込み		

PAS4000サーバロケーションテーブルの構造の例:

データ領域	開始アドレス	データ長
0x読み取り	0	4
0x書き込み	100	4
1x	0	0
3x	0	0
4x読み取り	0	2
4x書き込み	100	2

9.2.7 データ転送の制限

Modbus/TCPの最大接続数

Modbus/TCPの最大接続数 (クライアント接続またはサーバ接続、あるいはその両方) は、装置によって異なります。詳細は、関連するPSS 4000の取扱説明書に記載されています。

また、PSS 4000装置1台のIP接続数は、32が一般的な上限であることにも注意してください。



重要

IP接続の最大数には、次の規則が適用されます。

「IP接続」システムセクションを使用してPSS 4000装置で管理できるIP接続の最大数は、装置によって異なります。

ただし、「IP接続」システムセクションでは、最大32個のIP接続を管理できます。この上限は、「IP接続」システムセクションで、複数の異なるプロトコルや通信回線を使用している場合にも適用されます。

SafetyNET p接続は「IP接続」システムセクションで**管理されない**ため、この上限に影響しません。

クライアント接続の制限

クライアント接続では、テレグラムあたりの最大データ長が制限されます。以下のテーブルでは、サポートされるテレグラムあたりの最大データ長について情報を示します。

データ転送		テレグラムあたりのデータ長
データの読み取り (ビット)	FC 01 (コイルの読み取り)	1～最大2000
	FC 02 (ディスクリート入力の読み取り)	
データの読み取り (ビット)	FC 05 (単一コイルの書き込み)	1ビット
	FC 15 (複数コイルの書き込み)	1～最大1968
データの読み取り (ワード)	FC 03 (保持レジスタの読み取り)	1～最大125
	FC 04 (入力レジスタの読み取り)	
データの書き込み (ワード)	FC 06 (単一レジスタの書き込み)	1ワード
	FC 16 (複数レジスタの書き込み)	1～最大123ワード
データの読み取りと 書き込み (ワード)	FC23 (複数レジスタの読み取り／書き込み)	読み取り1～最大125ワード 書き込み1～最大121ワード

FC 03、FC 04、およびFC 16は、複数のテレグラムでのデータ転送をサポートします。FC 03、FC 04、またはFC 16でのデータ転送時にサポートされるテレグラムあたりの最大データ長を超過した場合、データは複数のテレグラムに分散されます。以下のテーブルでは、複数のテレグラムでデータを転送した場合のデータ長について詳細を示します。

複数のテレグラムでのデータ転送		データ長
データの読み取り (ワード)	FC 03 (保持レジスタの読み取り)	126～最大2000ワード
	FC 04 (入力レジスタの読み取り)	
データの書き込み (ワード)	FC 16 (複数レジスタの書き込み)	124～最大2000ワード

クライアント接続がPAS4000でコンフィグレーションされる場合、テレグラムの多重転送に影響が及ぶ可能性があります。複数のテレグラムを転送する場合、接続サイクルごとに1つの (サブ) テレグラムを送信するかどうか、および次の (サブ) テレグラムを送信する前に各 (サブ) テレグラムに対する応答を待機する必要があるかどうかをコンフィグレーションすることができます。このようにコンフィグレーションしない場合は、接続サイクルですべての (サブ) テレグラムが応答を待機せずに連続して送信されます。転送が完了した後で、初めて (サブ) テレグラムへの応答が収集されます。データ転送は、こちらの方が最適化されます。



情報

サードパーティーメカの装置を通信相手として使用する場合は、データ長に関する制限が異なることがあります。関連ユニットの取扱説明書の情報を参照してください。

9.2.8 Modbus/TCPのI/Oデータ

「IP接続」システムセクションにより、Modbus/TCPを経由して、PSS 4000とサードパーティ装置の間でI/Oデータを交換できます。この操作を正常に実行するには、PAS4000のI/OマッピングエディタのI/Oデータに対応するModbus/TCPデータ領域にマッピングする必要があります。

Modbus/TCPデータ領域のI/Oマッピング

I/Oマッピングを使用して、どのデータソースからどのデータシンクにデータを供給するかを定義します。

Modbus/TCPデータ領域で可能なI/Oマッピング:

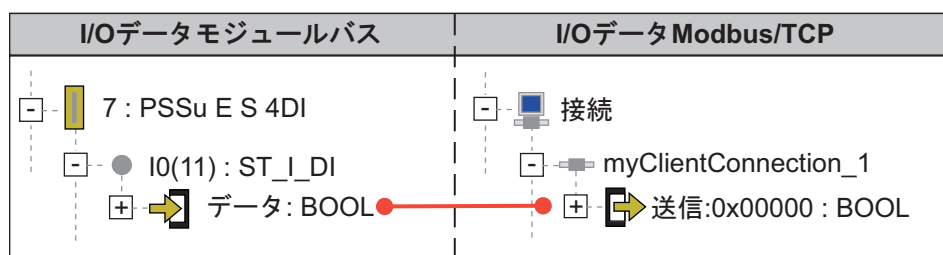
I/Oマッピング		Modbus/TCP			
		クライアントの役割		サーバの役割 (サーバロケーションテーブル)	
		送信 データ領域 (0x、4x)	受信 データ領域 (0x、1x、3x、4x)	読み取り データ領域 (0x、1x、3x、4x)	書き込み データ領域 (0x、4x)
モジュールバス から	入力データ (例: ハードウェア入力)	◆	---	◆	----
	出力データ (例: ハードウェア出力)	---	◆	---	◆
PI 変数 から	入力PI変数	---	◆	---	◆
	出力PI変数	◆	---	◆	---

◆ I/Oマッピング可
 --- I/Oマッピング不可

例

クライアント役割のI/Oマッピング

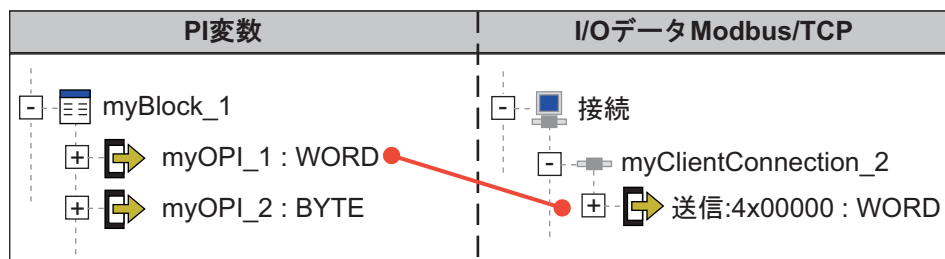
- ▶ ハードウェア入力の入力信号は、Modbus/TCP接続を経由して別のメーカーの装置に送信されます。
 - データフローの方向: ハードウェア入力 -> Modbus/TCP
 - 使用するファンクションコード: FC05/FC15 (送信)
 - 可能なI/Oマッピング (例):



●—● I/Oマッピング

- ▶ 出力PI変数の内容が、ユーザプログラムからModbus/TCP接続を經由して別のメーカーの装置に送信されます。

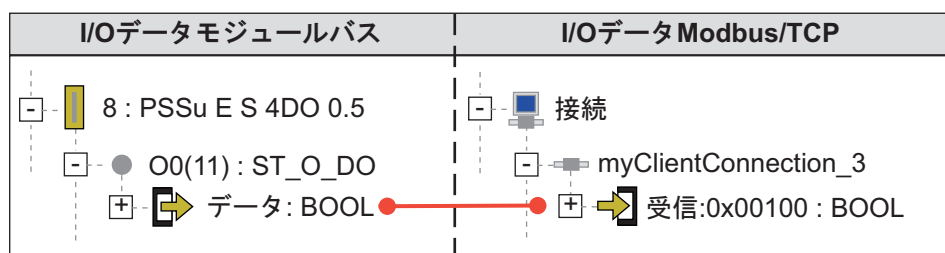
- データフローの方向: 出力PI変数 -> Modbus/TCP
- 使用するファンクションコード: FC06/FC16/FC23 (送信)
- 可能なI/Oマッピング (例):



●●● I/Oマッピング

- ▶ PSS 4000装置のハードウェア出力は、Modbus/TCP接続を通じて他の製造業者の装置により制御されます。

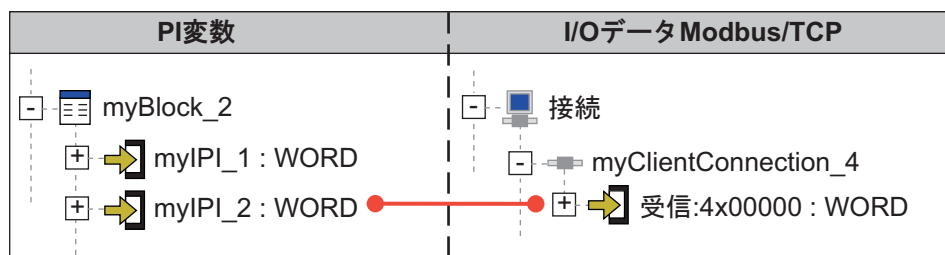
- データフローの方向: Modbus/TCP -> ハードウェア出力
- 使用するファンクションコード: FC01 (受信)
- 可能なI/Oマッピング (例):



●●● I/Oマッピング

- ▶ データは、Modbus/TCP接続を通じて他の製造業者の装置により要求されます。一度データを受信すると、ユーザプログラム内でさらに処理できるようになります。

- データフローの方向: Modbus/TCP -> 入力PI変数
- 使用するファンクションコード: FC03/FC23 (受信)
- 可能なI/Oマッピング (例):



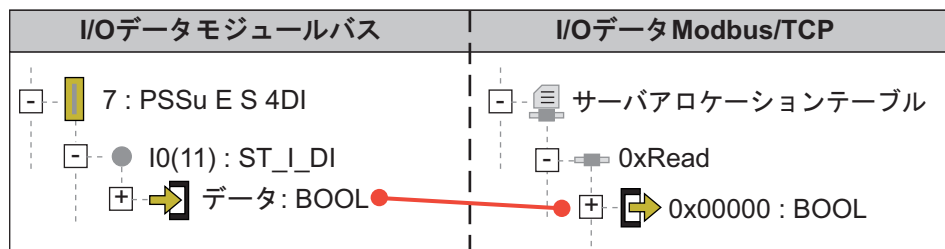
●●● I/Oマッピング

サーバの役割のI/Oマッピング (サーバロケーションテーブル)

- ▶ ハードウェア入力からの信号は、サーバロケーションテーブルの読み取りデータ領域で使用可能になります。データの一部は、Modbus/TCP接続を通じて他のメーカーの装置から読み取ることができます。

– データフローの方向: ハードウェア入力 -> Modbus/TCP

– 可能なI/Oマッピング:

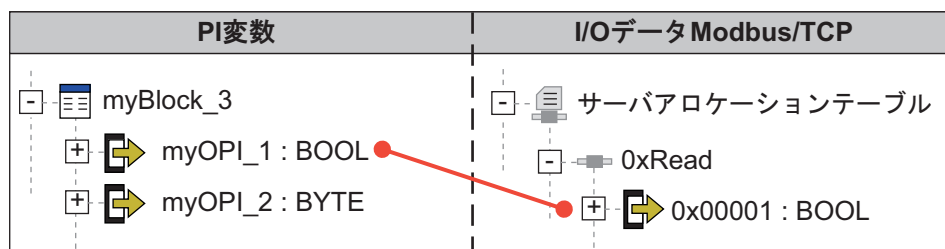


●●● I/Oマッピング

- ▶ 出力PI変数の内容は、サーバロケーションテーブルの読み取りデータ領域で使用可能になります。データの一部は、Modbus/TCP接続を通じて他のメーカーの装置から読み取ることができます。

– データフローの方向: 出力PI変数 -> Modbus/TCP

– 可能なI/Oマッピング:

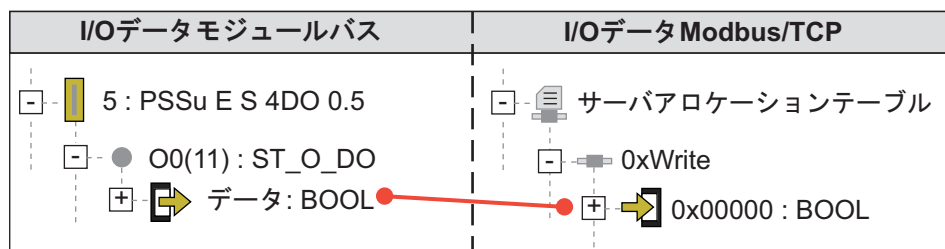


●●● I/Oマッピング

- ▶ PSS 4000装置のハードウェア出力は、Modbus/TCP接続を通じて他の製造業者の装置により制御されます。別のメーカーの装置は、サーバロケーションテーブルの書き込みデータ領域に書き込みます。

– データフローの方向: Modbus/TCP -> ハードウェア出力

– 可能なI/Oマッピング:

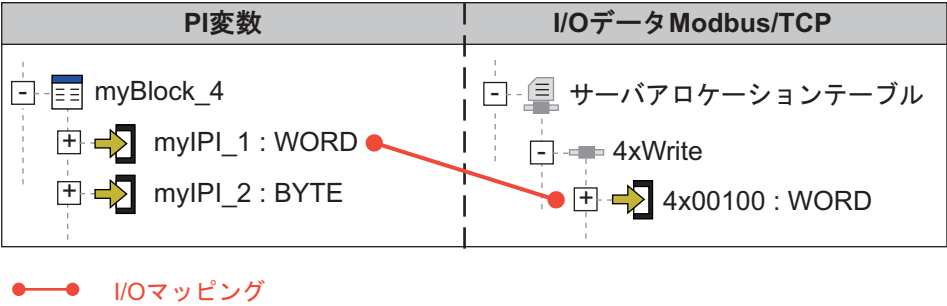


●●● I/Oマッピング

- ▶ 一部のデータは、Modbus/TCP接続を経由して、別のメーカーの装置により、サーバロケーションテーブルの書き込みデータ領域に書き込まれます。データは、ユーザプログラム内でさらに処理されます。

– データフローの方向: Modbus/TCP -> 入力PI変数

– 可能なI/Oマッピング:



9.2.9 Modbus/TCPアドレスのアドレス表示

PSS 4000装置のI/Oデータのアドレスは、PAS4000や診断など、複数の場所に表示されます。

9.2.9.1 送信データのアドレスフォーマット

Modbus/TCP送信データのアドレス (クライアントの役割) は、次の形式で構造化されます。

<デバイス名>.Modbus.Client.<接続名>.Send.<開始アドレス>

例:

myDevice_1.Modbus.Client.myConnection1.Send.0x00000

myDevice_1.Modbus.Client.myConnection2.Send.4x00005

9.2.9.2 受信データのアドレスフォーマット

Modbus/TCP受信データのアドレス (クライアントの役割) は、次の形式で構造化されます。

<デバイス名>.Modbus.Client.<接続名>.Receive.<開始アドレス>

例:

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

9.2.9.3 サーバロケーションテーブルでのアドレス表示

Modbus/TCPサーバロケーションテーブルのアドレス (サーバの役割) は、次の形式で構造化されます。

<デバイス名>.Modbus.Server.<サーバロケーションテーブルで選択できるデータ領域>.<開始アドレス>

例:

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

9.2.10 Modbus/TCPのデータの整合性

サードパーティの装置によって送信されるデータ (クライアントの役割) または提供されるデータ (サーバの役割) のデータ整合性については、これらの装置に責任があります。PSS 4000から見ると、これらの装置のデータの整合性に関してステートメントは作成できません。

クライアント接続から受信したデータのデータ整合性

PSS 4000専用／対応装置が接続クライアントである場合、この装置の視点に立つと、この接続からの受信データ (FC 01、FC 02、FC 03、FC 04、FC 23) は外部入力データになります。

1つのテレグラムで転送される場合 (「[データ転送の制限](#)」 136)を参照)、クライアント接続からの外部入力データは、PSS 4000で一貫したデータとして処理されます (「[データの一貫性](#)」 300)を参照)。

クライアント接続からの外部入力データは、整合性範囲タイプ「外部通信接続のST入力データ」の整合性範囲を形成します。正確な記述は、整合性範囲「Modbus/TCPサーバ接続のST-Iデータ」です (「[一貫性の範囲](#)」 301)を参照)。

サーバ接続から受信したデータのデータ整合性

PSS 4000専用／対応装置が接続サーバである場合、この装置の視点に立つと、この接続からの受信データ (データ領域0x書き込みまたは4x書き込み) は外部入力データになります。

1つのテレグラムで転送される場合 (「[データ転送の制限](#)」 136)を参照)、サーバ接続からの外部入力データは、PSS 4000で一貫したデータとして処理されます (「[データの一貫性](#)」 300)を参照)。

サーバ接続からの外部入力データは、整合性範囲タイプ「外部通信接続のST入力データ」の整合性範囲を形成します。正確な記述は、整合性範囲「Modbus/TCPサーバ接続のST-Iデータ」です (「[一貫性の範囲](#)」 301)を参照)。

クライアント接続の送信データのデータ整合性

PSS 4000専用／対応装置が接続クライアントである場合、この装置の視点に立つと、この接続からの送信データは外部出力データになります。

データソースの同じ整合性範囲から取得した場合 (「[データの一貫性](#)」 300)を参照)、クライアント接続の外部出力データ (FC 05、FC 15、FC 06、FC 16、FC 23) には整合性があります。

サーバ接続の送信データのデータ整合性

PSS 4000専用／対応装置が接続サーバである場合、この装置の視点に立つと、送信データ (データ領域0x読み取り、1x、3x、4x読み取り) は外部出力データになります。データソースの同じ整合性範囲から取得した場合 (「[データの一貫性](#)」 300)を参照)、サーバ接続の外部出力データには整合性があります。

9.2.11 データ領域0xと4x使用時の特殊な機能

一部の装置では、クライアント接続を使用して、サーバロケーションテーブルのデータ領域 (0x、4x) にデータを書き込んだ後、自動的に即座にデータを要求し、例えばディスプレイに表示します。

接続のサーバがPSS 4000専用／対応装置である場合、この「データミラーリング」は、装置に対して追加の措置を取った場合にのみ機能します。IEC 61131プログラミングとMultiプログラミングでは、操作に多少の違いがあります。

IEC 61131プログラミングの手順

- ▶ ブロックの作成
プログラムタイプPOUなど
- ▶ ブロック内での目的の入力PI変数の宣言
サーバロケーションテーブルの書き込みデータ領域 (0x書き込み、4x書き込み) へのアクセスには、入力PI変数を経由したI/Oマッピングが使用されます。
例: myIPI_1 AT %I* :WORD
- ▶ ブロック内での対応する出力PI変数の宣言
ブロック内で入力PI変数を復元するには、出力PI変数が必要です。これは、サーバロケーションテーブルの読み取りデータ領域 (0x読み取り、4x読み取り) に入力データを「データミラーリング」する場合の条件です。
例: myOPI_1 AT %Q* :WORD
- ▶ ブロック内で入力PI変数を出力PI変数に復元する方法による「データミラーリング」
例 (ILでのプログラミング):

RESTORE:	LD	myIPI_1
	ST	myOPI_1
- ▶ 入力PI変数とサーバロケーションテーブル (PAS4000のI/Oマッピングエディタ) の書き込みデータ領域 (0x書き込み、4x書き込み) のI/Oマッピング
- ▶ 出力PI変数とサーバロケーションテーブル (PAS4000のI/Oマッピングエディタ) の読み取りデータ領域 (0x読み取り、4x読み取り) のI/Oマッピング
- ▶ ブロックをPSS 4000専用／対応装置からリソースに割り付けます。
例: ST resource from myDevice_1

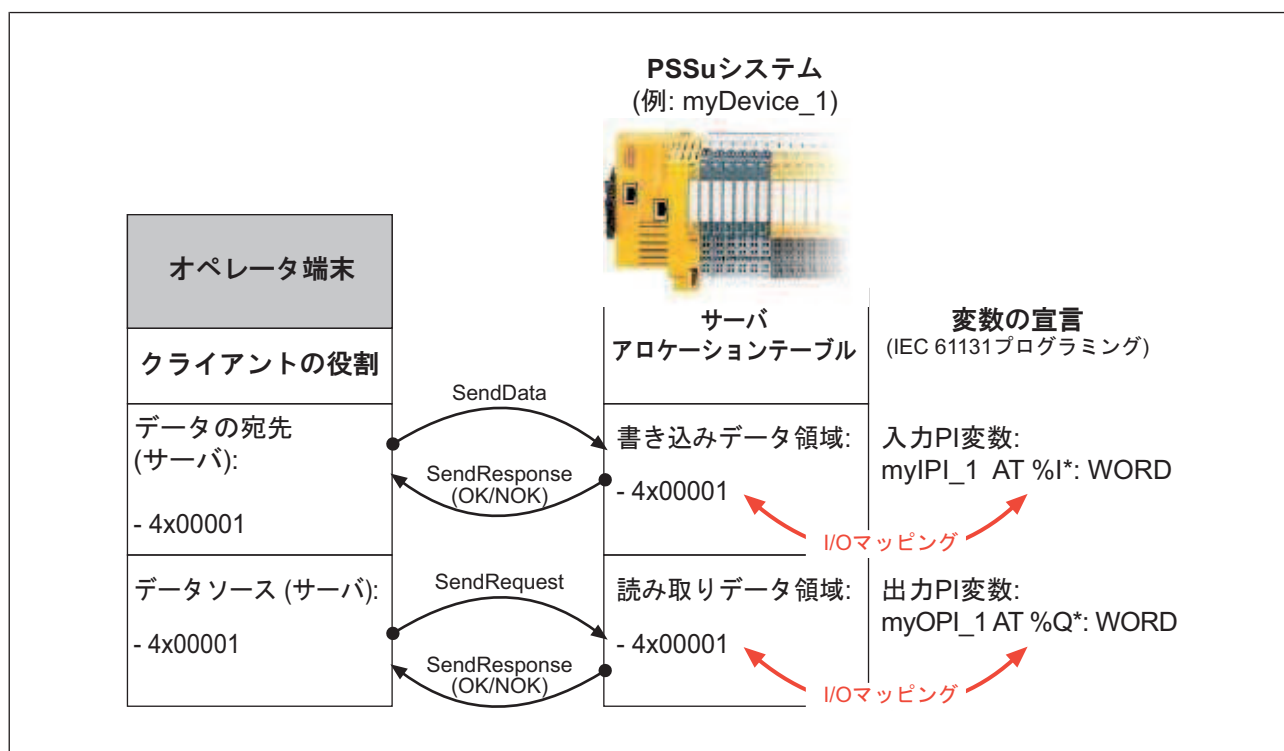


図: IEC 61131プログラミングでのデータミラーリング (例)

Multiプログラミングの手順

オプション1:

- ▶ MultiプログラムでのIEC 61131ベーシックブロックの作成
- ▶ IEC 61131ベーシックブロックに必要なPI変数の宣言と「データミラーリング」のプログラミング
この手順は、IEC-61131プログラミングの手順と同じです (上記を参照)。
- ▶ 入力PI変数とサーバアロケーションテーブルの書き込みデータ領域 (0x書き込み、4x書き込み) のI/Oマッピング
- ▶ 出力PI変数とサーバアロケーションテーブルの読み取りデータ領域 (0x読み取り、4x読み取り) のI/Oマッピング
- ▶ ブロックをPSS 4000専用／対応装置からリソースに割り付けます。
例: ST resource from myDevice_1

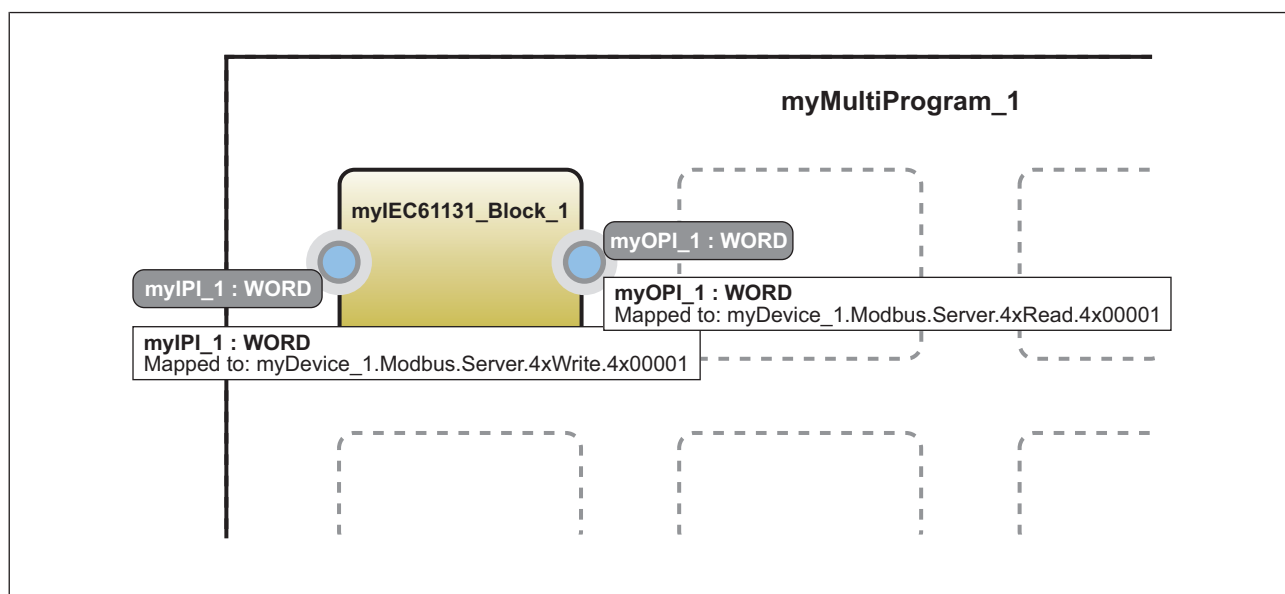
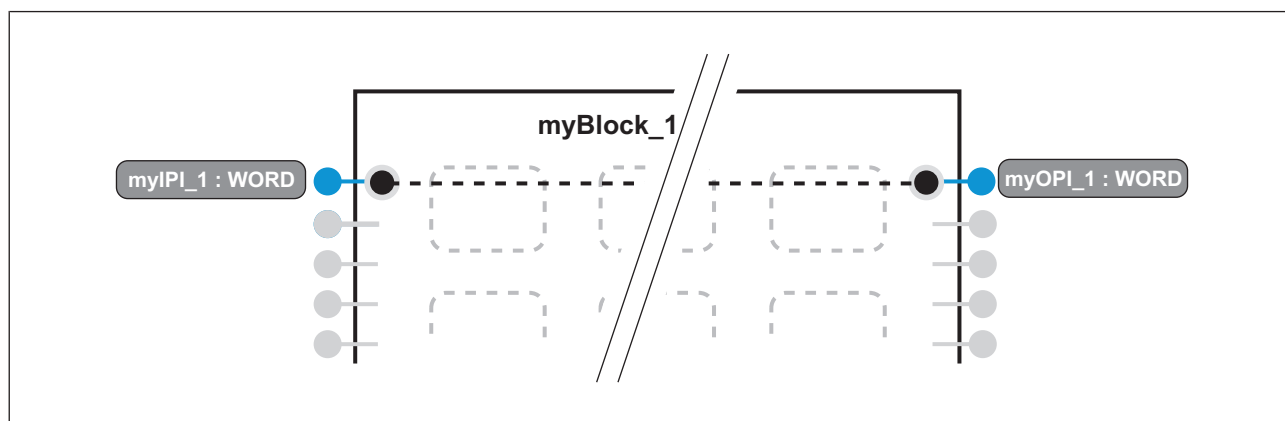


図: MultiプログラミングでのI/Oマッピング – IEC 61131ベーシックブロック (例)

オプション2:

- ▶ Multiプログラムでのコンポーネントブロック (CB) の作成
- ▶ ブロックインタフェースのコンフィグレーションで行う作業
 - 必要な入力PIポイントのコンフィグレーション。これは、入力PI変数を表します。
サーバロケーションテーブルの書き込みデータ領域 (0x書き込み、4x書き込み) へのアクセスには、入力PIポイントを経由したI/Oマッピングが使用されます。
例: myIPI_1: WORD
 - 対応する出力PIポイントのコンフィグレーション。これは出力PI変数を表します。
ブロック内で入力PIポイントを出力PIポイントに接続するには、出力PIポイントが必要です。これは、サーバロケーションテーブルの読み取りデータ領域 (0x読み取り、4x読み取り) に入力データを「データミラーリング」する場合の条件です。
例: myOPI_1: WORD
- ▶ 入力PIポイントとブロックの出力PIポイントの接続による「データミラーリング」

例:



▶ I/Oマッピングのコンフィグレーション

- 入力PIポイントとサーバロケーションテーブルの書き込みデータ領域 (0x書き込み、4x書き込み) の I/Oマッピング
- 出力PIポイントとサーバロケーションテーブルの読み取りデータ領域 (0x読み取り、4x読み取り) の I/Oマッピング

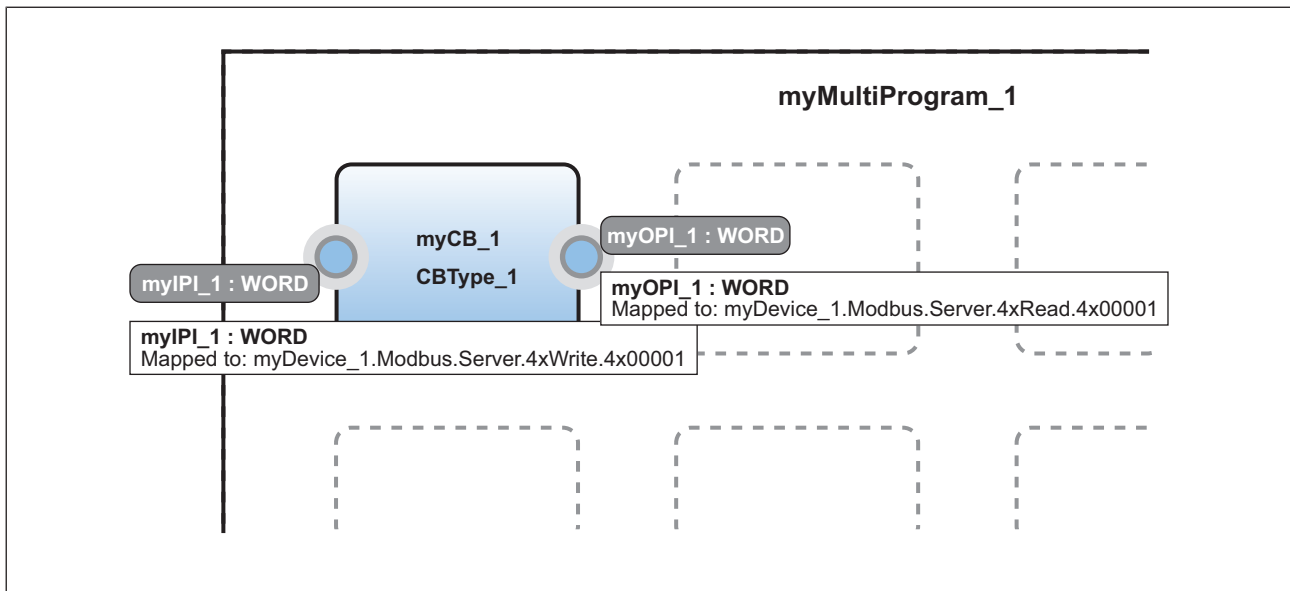


図: MultiプログラミングでのI/Oマッピング – コンポーネントブロック (例)

9.3 Raw TCP

伝送制御プロトコル (TCP) は、次の要件に適合する形で実装されています。

- ▶ RFC793
- ▶ RFC1122
- ▶ RFC2988
- ▶ RFC5681

「Raw TCP」は、上位層プロトコルなしで、TCP経由でデータを交換することを表します。

TCPは信頼性に優れた接続指向プロトコルです。2つの通信相手 (終点) 間の通信に使用されます。通信相手の通信サービスは、ポート番号経由で識別されます。TCPはアドレス処理にインターネットプロトコル (IP) を使用します。したがって、TCP接続は次の4つの値によって一意に識別されます。

- ▶ ソースのIPアドレス
- ▶ ソースのポート番号
- ▶ 宛先のIPアドレス
- ▶ 宛先のポート番号

このように指定されたTCP接続を介してデータを双方向に送信できます。

オートメーションシステムPSS 4000では、Raw TCP接続のIPアドレス、ポート番号、および送信方向は、接続のコンフィグレーション時に、PAS4000でコンフィグレーションされます。

Raw TCP経由でのデータ交換は、非安全関連アプリケーション用のオートメーションシステムPSS 4000で使用できます。

9.3.1 「IP接続」システムセクション

オートメーションシステムPSS 4000の場合、「IP接続」システムセクションがあるすべてのPSS 4000装置で、Raw TCP経由の外部通信を使用できます。この場合、PSS 4000装置の「IP接続」システムセクションが、通信に必要なタスクをすべて実行します。通信に必要なタスクには以下が含まれます。

- ▶ Raw TCP接続の管理
(「[Raw TCP接続](#)  150」を参照)
- ▶ Raw TCPを使ったデータ交換
(「[Raw TCPを使ったデータ交換](#)  151」を参照)

9.3.2 Raw TCP接続

PSS 4000装置の「IP接続」システムセクションでは、装置に対してコンフィグレーションするRaw TCP接続を管理します。接続は、PAS4000のIP接続エディタでコンフィグレーションされます。

PSS 4000装置に対して、以下のRaw TCP接続をコンフィグレーションできます。

- ▶ 送信接続

- ▶ 受信接続

受信データのためにFIFOメモリを有効にできます。

- ▶ 送受信接続

受信データのためにFIFOメモリを有効にできます。

各Raw TCP接続は、接続の確立時に、クライアントまたはサーバのどちらの役割を使用するかに応じてコンフィグレーションされます。

9.3.3 Raw TCPを使ったデータ交換

Raw TCPは接続指向のプロトコルです。したがって、使用可能なデータをRaw TCP経由で転送するには、2つのRaw TCP通信相手の間で、最初に接続を確立する必要があります。接続を開始する側は「クライアント」と呼ばれます。クライアントが接続を確立する通信相手は「サーバ」と呼ばれます。接続は、PAS4000のIP接続エディタでコンフィグレーションされます。

IP接続エディタで接続に対してクライアントの役割をコンフィグレーションする場合、この接続はPSS 4000の視点から「クライアント接続」と呼ばれます。それ以外の場合、この接続はPSS 4000の視点から「サーバ接続」と呼ばれます。このため、サーバ／クライアントの役割が適用されるのは特定の接続に限定され、有効であるのは通信の確立時に限定されます。

PSS 4000装置に対して、複数のRaw TCP接続をコンフィグレーションできます ([「データ転送の制限」](#) [161]を参照)。

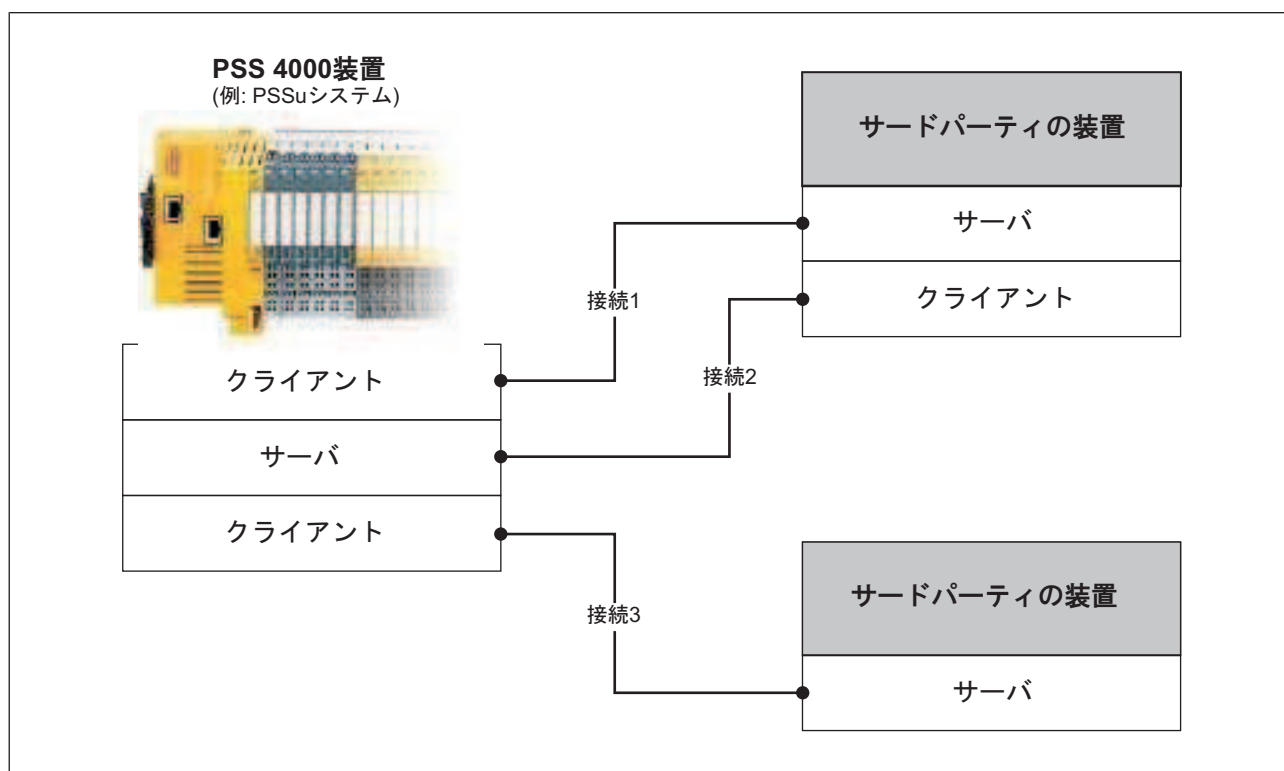


図: Raw TCP接続での役割の割り付け (例)

PSS 4000装置では、送信データと受信データが周期的に転送されます。周期的な転送では、接続クライアントとサーバの間で固定的な接続が維持されます。

オートメーションシステムPSS 4000から見ると、受信データは外部入力データと呼び、送信データは外部出力データと呼びます。

9.3.4 受信データ用FIFOメモリ

FIFOメモリは、PAS4000のコンフィグレーションを行うときに、受信データとのRaw TCP接続ごとに有効化できます。

FIFOメモリのサイズは装置により異なります。詳細は、関連するPSS 4000装置の取扱説明書に記載されています。

9.3.5 Raw TCPによる受信データの処理

TCPは、その設計上、連続したデータストリームを処理することを目的としています。データストリームがセグメントに分割されるのは、伝送上の理由のためだけです。

TCPを使用して周期的なプロセスデータを送信すると、プロセスデータが通常、同じサイズのデータパケット内で連続的かつ周期的に送信されるため、最初はTCPの設計と競合します。

TCPの元の設計にかかわらず、オートメーションシステムPSS 4000のRaw TCPは、プロセスデータを最初に最も重要なものとして、周期的に受信して処理できるように設計されています。

ただし、PSS 4000では、Raw TCPの受信動作をさまざまなアプリケーションシナリオに適合させることができます。これは、PAS 4000のコンフィグレーションにより可能です。

コンフィグレーションオプション:

- ▶ FIFOメモリを有効／無効にする
- ▶ 受信データは、コンフィグレーションされたデータ長を超えたり、下回ったりすることはできません

次の表に、コンフィグレーションオプションを慎重に組み合わせる方法の概要を示します。

コンフィグレーション	短いデータ長を受け入れない	短いデータ長を受け入れる
FIFOメモリの有効化	次のシナリオに適しているかどうか: 「処理データの送信」 (デフォルト設定) (「 Raw TCPの一般的なアプリケーションシナリオ [📖 154]」を参照してください)	次のシナリオに適しているかどうか: 「従来のTCPデータ通信」(たとえば、 可変データ長の要求応答プロトコル) (「 Raw TCPの一般的なアプリケーションシナリオ [📖 154]」を参照してください)
FIFOメモリの無効化	次のシナリオに適しているかどうか: 「プロセスデータを送信」(トランスミッタ オーバークロックが発生して接続が頻 繁に失敗する場合など、接続状態に 問題がある場合) (「 Raw TCPの一般的なアプリケーションシナリオ [📖 154]」を参照してください)	- 合理的なシナリオなし -

基本的な受信動作については、[受信データの処理時の基本的なワークフロー](#) [📖 156]を参照してください。FIFOメモリの影響の詳細については、[FIFOメモリがイネーブルの場合の受信動作](#) [📖 157]および[FIFOメモリが無効の場合の受信動作](#) [📖 159]を参照してください。

9.3.5.1 Raw TCPの一般的なアプリケーションシナリオ

Raw TCPは、さまざまな一般的なアプリケーションシナリオに対応できるように、PAS 4000でコンフィグレーションできます。

シナリオ:「プロセスデータの通信」

このタイプのデータ通信の目的:

TCP接続レシーバは、正しいデータ長に対応する場合に限り、バッファから現在の受信データを抽出して処理します。

レシーバのコンフィグレーション:

▶ FIFOメモリの有効化 (デフォルト設定)

特定の状況では、FIFOメモリを無効にすると役立つ場合があります。例を挙げると、トランスミッタのオーバークロックが発生して接続に頻繁に失敗するといった場合に、接続状態に問題がある場合です。この問題の詳細については、「レシーバの動作」を参照してください。

▶ 短いデータ長を受け入れない

レシーバの想定:

このコンフィグレーションでは、TCP接続レシーバはデータパケットが常に同じデータ長になることを想定しています。これを行うには、トランスミッタアプリケーションが固定データユニットを送信するようにする必要があります。つまり、トランスミッタは同じデータ長の周期的なプロセスデータを送信する必要があります。データ長は、PAS 4000でレシーバにコンフィグレーションされた受信データ長に対応する必要があります。

レシーバの動作:

各サイクル (t_{ExtCo}) で、TCP接続レシーバは、コンフィグレーションされた受信データ長のプロセスデータをバッファから抽出します。

▶ データ長が短い場合の動作:

何らかの理由で、レシーバが短いデータ長のプロセスデータを持つデータユニットを取得した場合、これはエラーと見なされます。接続が終了し、再確立されます。

▶ トランスミッタのオーバークロックが発生して接続が頻繁に失敗する場合の動作

トランスミッタがオーバークロックした場合、トランスミッタの送信速度は、レシーバがプロセスデータを受け入れることができる速度よりも早くなります。接続が頻繁に失敗する場合は、接続をそのたびに再確立する必要があります。

デフォルト設定 (FIFOメモリ有効) では、トランスミッタのオーバークロックと頻繁な接続失敗の両方により、バッファ内の古いデータの蓄積が増加します。FIFOの行動に従って、データは受信順に抽出されます。つまり、新しいデータが到着する前に古いデータを処理する必要があります。

この状況では、FIFOメモリを無効にすると便利な場合があります。FIFOモードがコンフィグレーションされていない場合、最も新しいデータパケットのプロセスデータが常に使用されます。

シナリオ:「従来のTCPデータ通信」**このタイプのデータ通信の目的:**

TCP接続レシーバは、バッファからすべてのデータを順番に抽出して処理することになっています。バッファ内のデータパケットのデータ長は変動する場合があります。

PSS 4000の特徴は、バッファから抽出されたデータが常にコンフィグレーションされた受信データ長に対応することです。したがって、このシナリオでは、ユーザはデータユニットを識別するために追加の基準を設定する必要があります。たとえば、PSS 4000ユーザプログラムにおけるプロトコルベースの基準によって、データ送信を要求と応答を介して制御することができます。また、たとえば、送信されたペイロードバイト数を各データパケットとともに送信し、それに応じてレシーバで評価できます。

レシーバのコンフィグレーション:

- ▶ FIFOメモリの有効化
- ▶ 短いデータ長を受け入れる

レシーバの想定:

このコンフィグレーションで、TCP接続レシーバは、アプリケーションで定義されたプロトコルベースの基準に従って、トランスミッタがデータを送信することを想定しています。データユニットは、それぞれのデータ長が異なる場合があります。そのため、レシーバにはデータパケットのデータ長に関する想定がありません。

レシーバの動作:

各サイクル (t_{ExtCo}) で、TCP接続レシーバは、受信データ長がコンフィグレーションされたデータをバッファから抽出します。データは、受信された順序、つまり最も古いデータから抽出されます。バッファに含まれるデータがコンフィグレーションされた受信データ長よりも少ない場合、残りのデータは、コンフィグレーションされた受信データ長まで0で埋められます。

9.3.5.2 受信データの処理時の基本的なワークフロー

TCPデータは非同期に受信され、バッファに保存されます。バッファからデータが抽出されるときは、常に同じ原則が適用されます。各サイクル (t_{ExtCo}) で、TCP接続レシーバは、接続の受信データ長としてコンフィグレーションされた正確なデータ量を抽出します。つまり、各サイクル (t_{ExtCo}) で、固定データ長を持つデータがバッファから抽出され、受信接続用にコンフィグレーションされたPSS 4000配列に転送されます ([Raw TCPのI/Oデータ \[162\]](#)を参照)。このデータ抽出手順は、FIFOモードがコンフィグレーションされているかどうかに関係なく使用されます。この手順は、データ長が固定されたデータパケットを含むバッファと、データ長が可変のデータパケットを含むバッファからデータを抽出する場合にも適用されます。

同一データ長のデータパケットを受信するときのバッファの内容

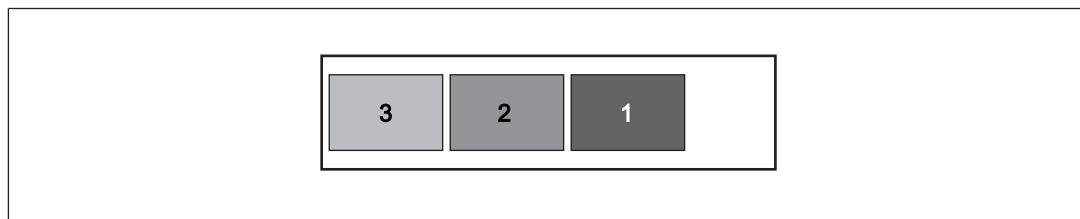


図: 同一データ長のデータパケットを含むバッファ

可変データ長のデータパケットを受信するときのバッファの内容:

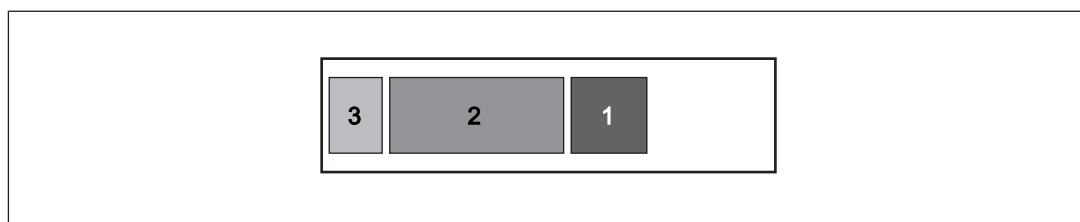


図: 可変データ長のデータパケットを含むバッファ

9.3.5.3 FIFOメモリがイネーブルの場合の受信動作

この動作は「元のTCP動作」と呼ばれます。この処理の目的は、新しいデータが到着したとき、古いデータがまだ処理されていなくても拒否される事態を防ぐことです。

各サイクル (t_{ExtCo}) で、TCP接続レシーバは、受信データ長がコンフィグレーションされたデータパケットをバッファから抽出し、受信接続用にコンフィグレーションされたPSS 4000配列に転送します。FIFOの動作に従って、データは受信された順序で抽出されます。つまり、データがPSS 4000装置で続いて処理されると、最も古いデータパケットのデータから処理が開始されます。

この手順でレシーバが過負荷になると、状況によってはレシーバが新しいデータを受信できなくなることがあります。この場合、TCPには、レシーバが過負荷であることをトランスミッタに通知する独自のメカニズムがあります。この時点で、トランスミッタは、レシーバがTCPメカニズムを介して再び受信可能であることを信号で通知するまで、新しいデータの送信を停止します。

FIFOメモリがイネーブルになっている場合の受信動作は許容可能な動作であり、TCP接続トランスミッタが固定データ長でデータユニットを送信する場合には、合理的な結果になります。この場合、バッファ内のデータパケットのデータ長は同じです。

トランスミッタが可変データ長のデータユニットを送信する場合は、レシーバに対して「短いデータ長を受け入れる」をコンフィグレーションする必要があります。この場合、バッファには異なるデータ長のデータパケットが含まれる可能性があります。ただし、データは常にコンフィグレーションされた受信データ長で抽出されるため、データは複数のデータパケットから抽出される可能性があります。この場合、合理的な結果を得るために、トランスミッタとレシーバで追加のアプリケーション関連の基準が必要です。バッファに含まれるデータがコンフィグレーションされた受信データ長よりも少ない場合、可変データ長を使用すると、残りのデータはコンフィグレーションされた受信データ長までゼロで埋められます。

データ長が固定されている場合の受け入れられた動作の例:

次の例は、データ長が同一または固定のデータパケットを含むバッファから、コンフィグレーションされた受信データ長でデータがいかにして抽出されるかを示しています。

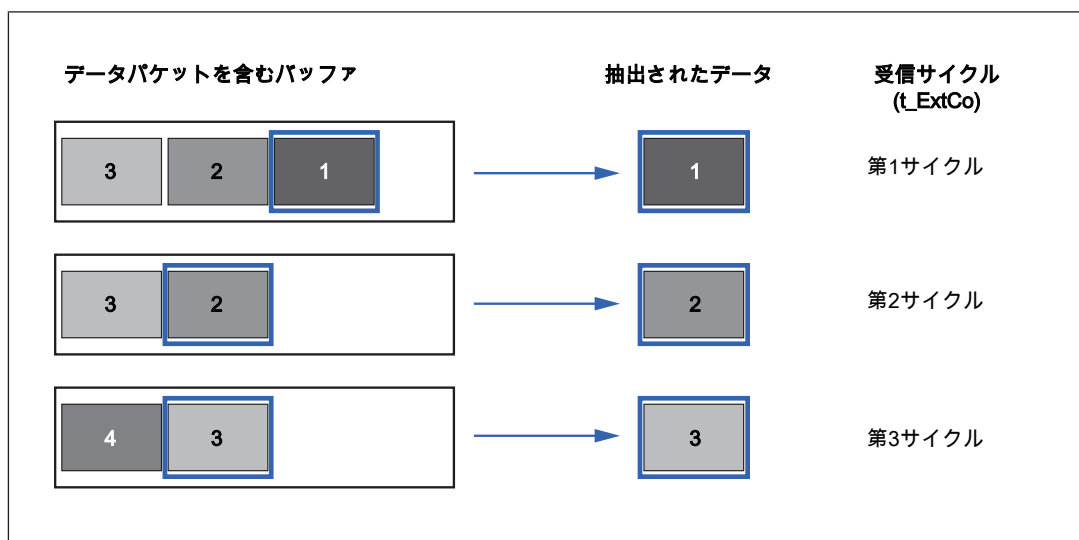


図: データ長が固定のデータパケットのデータを抽出するために使用されるFIFO動作 (例)

可変データ長での動作の例:

次の例は、データ長が可変のデータパケットを含むバッファから、コンフィグレーションされた受信データ長でデータがいかにして抽出されるかを示しています。

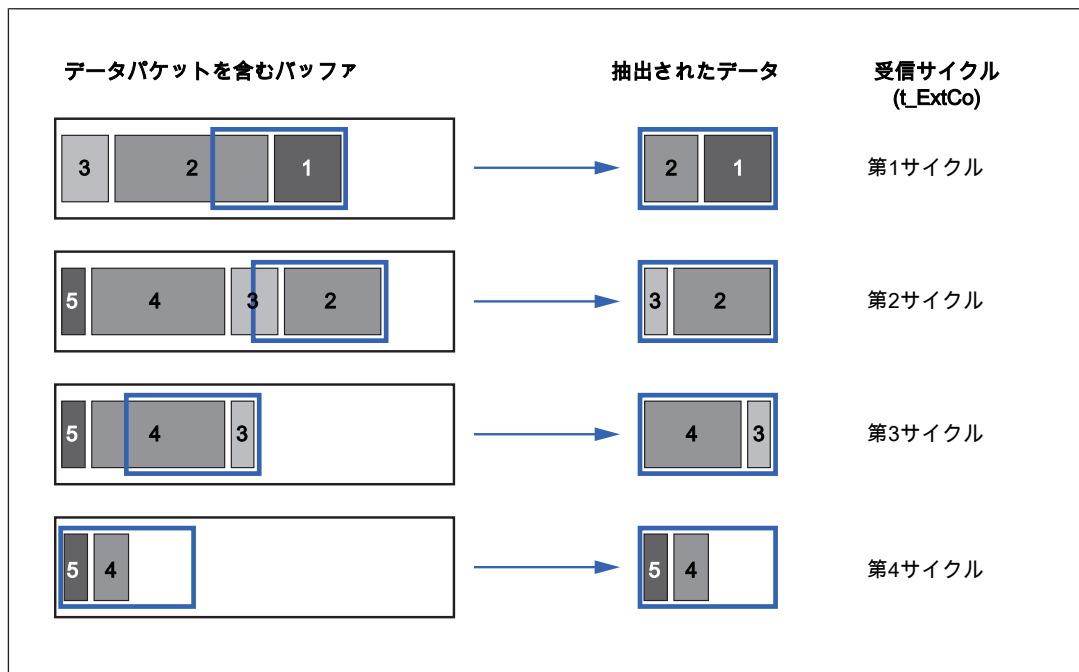


図: データ長が可変のデータパケットのデータを抽出するために使用されるFIFOの動作 (例)

次の点に注意してください:

この例では、追加の基準をアプリケーションで設定して、同類のデータを、抽出された後で識別または再構築できるようにする必要があります。

9.3.5.4 FIFOメモリが無効の場合の受信動作

この動作は「プロセスデータ動作」と呼ばれます。この手順の目的は、最新の受信データだけを処理することです。この目的のため、各サイクル (t_{ExtCo}) において、受信データが見つからなくなるまで、バッファの内容が検査され、必要な回数だけ拒否されます。検査中のデータ長は、コンフィグレーションされた受信データ長に対応します。最新の受信データを含むデータパケットが抽出され、後続の処理のために受信接続用にコンフィグレーションされた配列に転送されます。

FIFOメモリが無効になっている場合の受信動作は許容可能な動作であり、TCP接続トランスミッタが固定データ長でデータユニットを送信する場合には、合理的な結果になります。この場合、バッファ内のデータパケットのデータ長は同じです。

トランスミッタが可変データ長のデータユニットを送信する場合は、レシーバに対して「短いデータ長を受け入れる」をコンフィグレーションする必要があります。この場合、バッファには異なるデータ長のデータパケットが含まれる可能性があります。ただし、データは常にコンフィグレーションされた受信データ長で検査されるため、データが属する場所を考慮せずに拒否されます。バッファに含まれるデータがコンフィグレーションされた受信データ長よりも少ない場合、可変データ長を使用すると、残りのデータはコンフィグレーションされた受信データ長までゼロで埋められます。このシナリオに適した合理的なアプリケーションについては考慮していません。

データ長が固定されている場合の受け入れられた動作の例:

次の例は、データ長が同一または固定のデータパケットを含むバッファから、コンフィグレーションされた受信データ長でデータがいかにして抽出されるかを示しています。

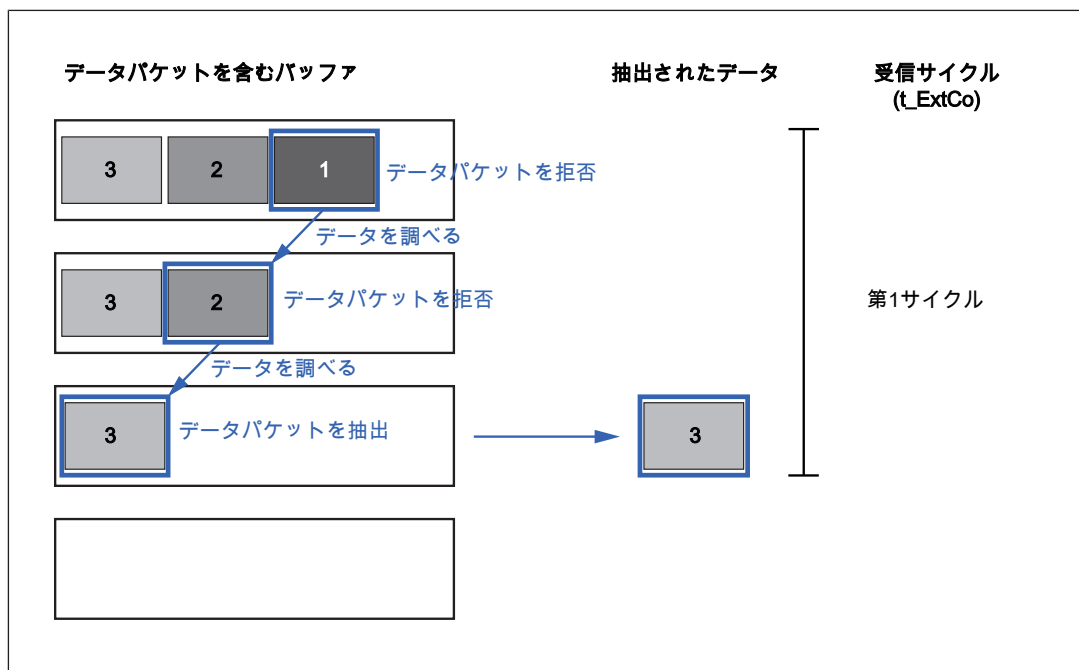


図: データ長が固定のデータパケットのデータを抽出するために使用されないFIFO動作 (例)

可変データ長での動作の例:

次の例は、データ長が可変のデータパケットを含むバッファから、コンフィグレーションされた受信データ長でデータがいかにして抽出されるかを示しています。

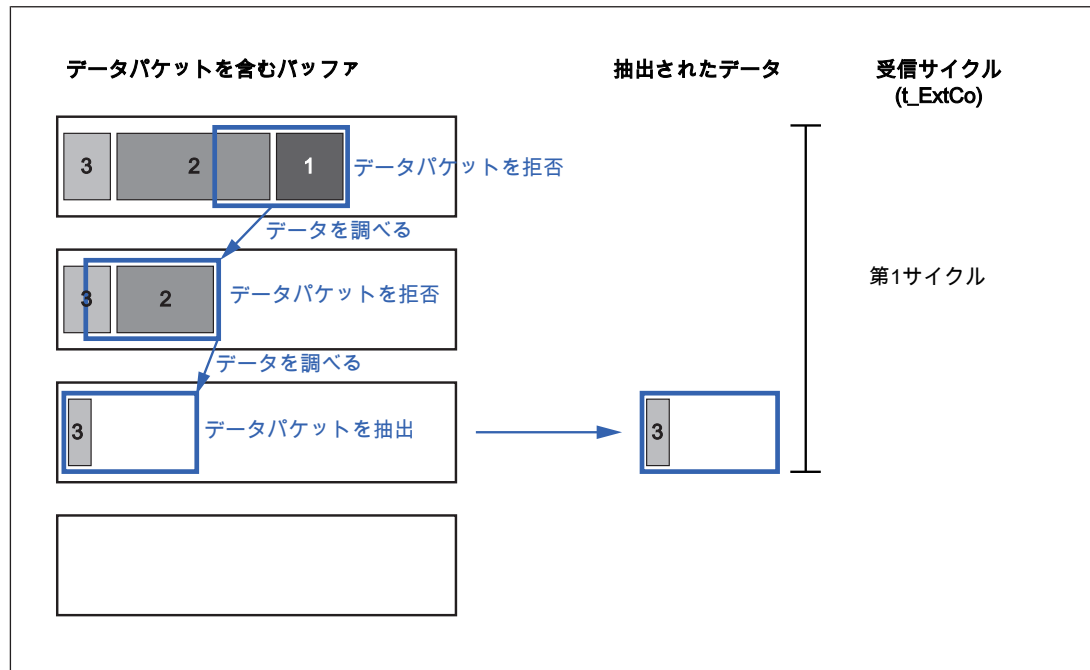


図: データ長が可変のデータパケットのデータを抽出するために使用されないFIFOの動作 (例)

この例は合理的なシナリオではないことに注意してください。

9.3.6 データ転送の制限

Raw TCP経由でデータを送信する場合、PSS 4000装置には次の制限が適用されます:

- ▶ Raw TCP接続の最大数は装置によって異なります。詳細は、関連するPSS 4000装置の取扱説明書に記載されています。また、PSS 4000装置1台のIP接続数は、32が一般的な上限であることにも注意してください。



重要

IP接続の最大数には、次の規則が適用されます。

「IP接続」システムセクションを使用してPSS 4000装置で管理できるIP接続の最大数は、装置によって異なります。

ただし、「IP接続」システムセクションでは、最大32個のIP接続を管理できます。この上限は、「IP接続」システムセクションで、複数の異なるプロトコルや通信回線を使用している場合にも適用されます。

SafetyNET p接続は「IP接続」システムセクションで**管理されない**ため、この上限に影響しません。

- ▶ PSS 4000装置から見ると、単一のサーバ接続がコンフィグレーションされている場合、このサーバを使用して単一のクライアントに接続できます。複数のクライアントで通信が必要な場合、PSS 4000装置から見ると、クライアントごとに独立したサーバ接続をコンフィグレーションする必要があります。
- ▶ 受信接続に対して、1～1024 Byteまでデータ長 (外部入力データ) をコンフィグレーションできます。
- ▶ 送信接続に対して、1～1024 Byteまでデータ長 (外部出力データ) をコンフィグレーションできます。
- ▶ 送受信接続に対して、1～1024 Byteまで各送信方向のデータ長をコンフィグレーションできます。
- ▶ デフォルトコンフィグレーション (PAS 4000) では、データパケットはコンフィグレーションしたデータ長より短くすることはできません。コンフィグレーションしたデータ長より短いTCPセグメントは拒否され、診断リストに診断メッセージが表示されます。
PAS 4000で「データ長が不足していても受信時に受け入れる」ようにコンフィグレーションされている場合は、可変データ長のデータパケットが受け入れられます。つまり、コンフィグレーションされた受信データ長より短いTCPセグメントが受け入れられます。
- ▶ 接続の受信データのFIFOメモリのサイズは装置により異なります。詳細は、関連するPSS 4000装置の取扱説明書に記載されています。

9.3.7 Raw TCPのI/Oデータ

「IP接続」システムセクションにより、Raw TCPを経由して、PSS 4000とサードパーティ装置の間でI/Oデータを交換できます。この操作を正常に実行するには、PAS4000のI/OマッピングエディタのI/Oデータに対応するRaw TCPデータ領域にマッピングする必要があります。

Raw TCPデータ領域のI/Oマッピング

I/Oマッピングを使用して、どのデータソースからどのデータシンクにデータを供給するかを定義します。

Raw TCPデータ領域で可能なI/Oマッピング:

I/Oマッピング		Raw TCP	
		受信データ (=外部入力データ)	送信データ (=外部出力データ)
モジュールバス	入力データ (例: ハードウェア入力)	---	◆
	出力データ (例: ハードウェア出力)	◆	---
PI 変数	入力PI変数	◆	---
	出力PI変数	---	◆

◆ I/Oマッピング可能

--- I/Oマッピング不可

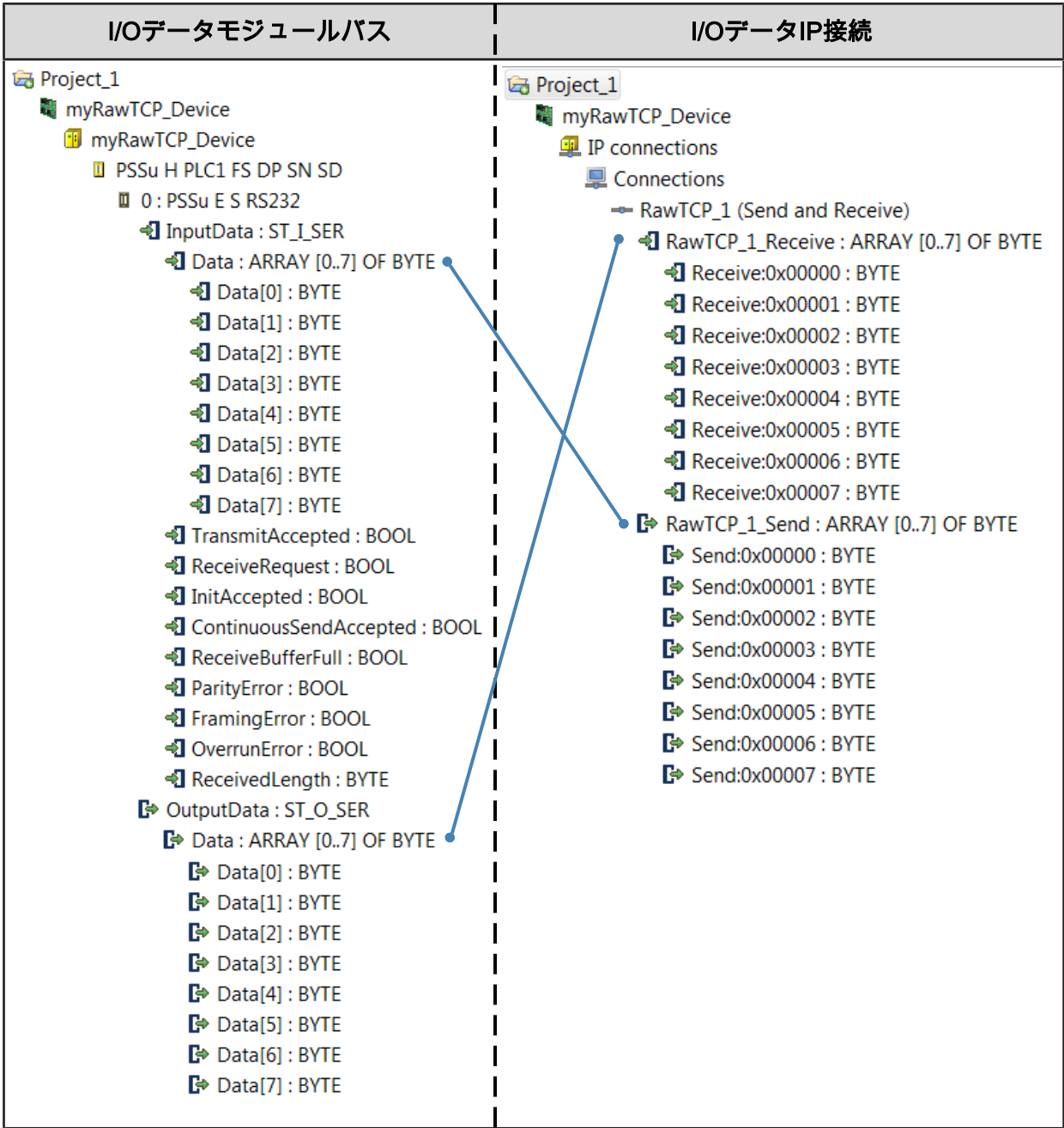
例

▶ I/OデータのI/Oマッピング

電子モジュールPSSu K S RS232の受信データは通信相手に送信され、通信相手が受信するデータはPSSu K S RS232で出力されます。

「RawTCP_1」の送受信接続は、この目的のためにコンフィグレーションされます。

可能なI/Oマッピング (例):



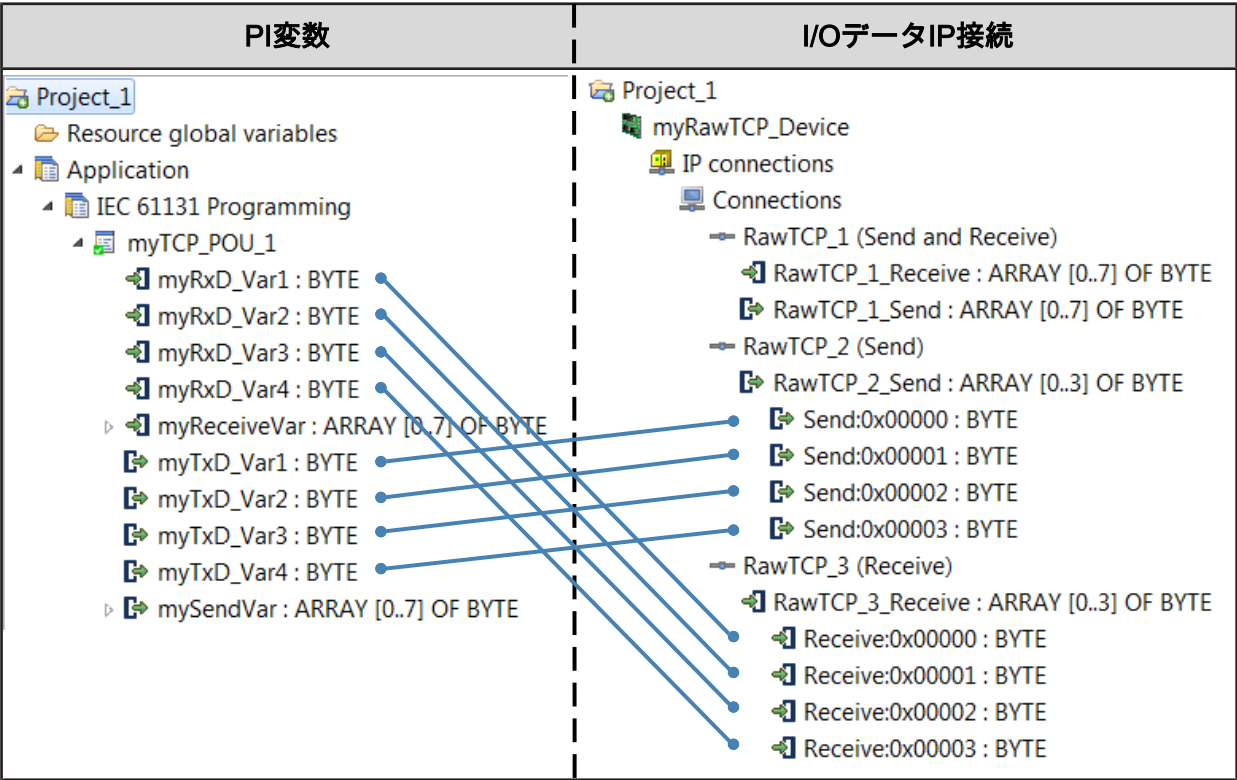
●—● I/Oマッピング

▶ PI変数のI/Oマッピング

ブロックの出力PI変数の内容は通信相手に送信され、入力PI変数は受信データに割り付けられます。この目的のために、2接続がコンフィグレーションされます。

- RawTCP_2 (送信接続)
- RawTCP_3 (受信接続)

可能なI/Oマッピング (例):



—●— I/Oマッピング

9.3.8 Raw TCPアドレスのアドレスフォーマット

PSS 4000装置のI/Oデータのアドレスは、PAS4000や診断など、複数の場所に表示されます。

送信データのアドレスフォーマット

Raw TCP送信データのアドレスは、次の形式で構造化されます。

<装置名>.TCP.<接続名>.Send.Byte[<ファンクション数>]

例:

myDevice_1.TCP.myTxConnection_1.Send.Byte[0]

...

myDevice_1.TCP.myTxConnection_1.Send.Byte[1023]

受信データのアドレスフォーマット

Raw UDP受信データのアドレスは、次の形式で構造化されます。

<装置名>.TCP.<接続名>.Receive.Byte.[<ファンクション数>]

例:

myDevice_1.TCP.myRxConnection_1.Receive.Byte[0]

...

myDevice_1.TCP.myRxConnection_1.Receive.Byte[1023]

9.3.9 Raw TCPのデータの整合性

受信データがサードパーティの装置で送信される環境では、このデータの整合性に対する責任は、これらの装置にあります。PSS 4000から見ると、これらの装置のデータの整合性に関してステートメントは作成できません。

Raw TCP接続の受信データのデータ整合性

PSS 4000専用／対応装置が、Raw TCP接続のデータの受信機である場合、PSS 4000から見ると、受信データは外部入力データになります。

Raw TCP接続からの外部入力データは、PSS 4000で一貫したデータとして処理されます ([「データの一貫性」](#)  300]を参照)。

Raw TCP接続からの外部入力データは、整合性範囲タイプ「外部通信接続のST入力データ」の整合性範囲を形成します。正確な記述は、整合性範囲「Raw TCP接続のST-Iデータ」です ([「一貫性の範囲」](#)  301]を参照)。

Raw TCP接続の送信データのデータ整合性

Raw TCP接続のPSS 4000専用／対応装置が、データを送信する場合、装置から見て、送信データは外部出力データです。

データソースの同じ整合性範囲から取得した場合 ([「データの一貫性」](#)  300]を参照)、Raw TCP接続の外部出力データには整合性があります。

9.4 Raw UDP

ユーザデータグラムプロトコル (UDP) はRFC768に適合して実装されます。「Raw UDP」は、上位層プロトコルなしで、UDP経由でデータを交換することを表します。

UDPは、さまざまな通信相手との通信に使用されます。通信相手の通信サービスは、ポート番号経由で識別されます。UDPはアドレス処理にインターネットプロトコル (IP) を使用します。つまり、UDPデータグラムはIPフレームにカプセル化され、IPアドレスを使用してデータパケットとして通信相手に送信されます。

UDPは「非接続指向」または「無接続型」のプロトコルです。特性:

- ▶ 通信相手との接続は、送信を開始するまで確立されません。
- ▶ 次の結果が保証されません。
 - 送信データパケットが宛先に到達すること
 - 送信データパケットが宛先に一度だけ到達すること
 - データパケットが送信した順序で到達すること

したがって、アプリケーションが損失したデータグラムや順序が混乱したデータグラムを識別しない場合、または通信相手側でアプリケーション自体が適切な制御および修正メカニズムを提供する場合にのみ、UDPを使用することができます。

UDPは無接続型のプロトコルですが、説明を簡単にするために、以下の説明では、通信相手との通信上の関係を「接続」と呼ぶことにします。

Raw UDP経由でのデータ交換は、非安全関連アプリケーション用のオートメーションシステムPSS 4000で使用できます。

9.4.1 「IP接続」システムセクション

オートメーションシステムPSS 4000の場合、「IP接続」システムセクションがあるすべてのPSS 4000装置で、Raw UDP経由の外部通信を使用できます。この場合、PSS 4000装置の「IP接続」システムセクションが、通信に必要なタスクをすべて実行します。次のようなものがあります。

- ▶ Raw UDP接続の管理
(「[Raw UDPの接続](#)」 169」を参照)
- ▶ データ交換
(「[Raw UDPを使ったデータ交換](#)」 170」を参照)

9.4.2 Raw UDPの接続

PSS 4000装置の「IP接続」システムセクションでは、装置に対してコンフィグレーションするRaw UDP接続を管理します。接続は、PAS4000のIP接続エディタでコンフィグレーションされます。

PSS 4000装置に対して、以下のRaw UDP接続をコンフィグレーションできます。

▶ **送信接続 (役割 = 送信機)**

- ユニキャスト (1:1接続)
- ブロードキャスト (1:「全」接続)
ループバックをコンフィグレーションできます。
- RFC 1112、IGMVP V1に適合したマルチキャスト (1:n接続)

▶ **受信接続 (役割 = 受信機)**

受信データのためにFIFOメモリを有効にできます。

- ユニキャスト (1:1接続)
- ブロードキャスト (1:「全」接続)
- RFC 1112、IGMVP V1に適合したマルチキャスト (1:n接続)

▶ **送受信接続 (役割 = 送信機／受信機)**

受信データのためにFIFOメモリを有効にできます。

- ユニキャスト (1:1接続)
- ブロードキャスト (1:「全」接続)
送信時のループバックをコンフィグレーションできます。
- RFC 1112、IGMVP V1に適合したマルチキャスト (1:n接続)

9.4.3 Raw UDPを使ったデータ交換

Raw UDPは接続指向ではありません。つまり、Raw UDPには、接続の確立と終了のメカニズムがありません。

PSS 4000装置は、PAS4000で送信接続 (役割 = 送信機) としてコンフィグレーションされた接続により、この接続の受信機の役割を持つ1つ以上の通信相手にデータを送信します。

PAS4000で受信接続 (役割 = 受信機) としてコンフィグレーションされた接続の場合、接続は、送信機の役割を持つ通信相手がデータを送信するまで待機します。

PAS4000で送受信接続 (役割 = 送信機／受信機) としてコンフィグレーションされた接続は、データの送信と受信の両方に使用されます。通信を確立する、つまり、この接続を経由して最初のテレグラムを送信する通信相手は、送受信接続に対してコンフィグレーションされる必要があります。この通信相手をイニシエータといいます。イニシエータ機能のない通信相手は、この接続を通じて最初のテレグラム受信を待機します。

オートメーションシステムPSS 4000から見ると、受信データは外部入力データと呼び、送信データは外部出力データと呼びます ([[処理データの有効性](#)  281])を参照)。

イニシエータ機能を使用する送受信接続は、PAS4000内で以下の一般的な規則に従ってコンフィグレーションされます。

役割	ネットワーク設定	
送信機／受信機	リモートポート番号	0以外
	リモートIPアドレス	0.0.0.0以外



情報

送受信接続のイニシエータは、リモートポート番号とリモートIPアドレスの両方の条件を満たす必要があります。

1つの条件のみを満たしていない場合、つまり、リモートポート番号が0である場合、またはリモートIPアドレスが0.0.0.0である場合、あるいはその両方である場合、送受信接続にはイニシエータ機能が導入されません。

9.4.4 受信データ用FIFOメモリ

FIFOメモリは、PAS4000のコンフィグレーションを行うときに、受信データとのRaw UDP接続ごとに有効化できます。

FIFOメモリがRaw UDP接続で有効化されている場合、受信データはすべてFIFOメモリで収集され、到着した順に処理されます。この処理により、新しいデータが到着したときに「古い」データが未処理のまま拒否される事態を防ぐことができます。

FIFOメモリのサイズは装置により異なります。詳細については、関連するPSS 4000の取扱説明書を参照してください。メモリのオーバーフローが発生すると、受信データは拒否され、診断リストに診断メッセージが表示されます。

9.4.5 データ転送の制限

Raw UDP経由でデータを転送する場合、PSS 4000装置には次の制限が適用されます。

- ▶ Raw UDP接続の最大数は装置によって異なります。詳細については、関連するPSS 4000の取扱説明書を参照してください。
- また、PSS 4000装置1台のIP接続数は、32が一般的な上限であることにも注意してください。



重要

IP接続の最大数には、次の規則が適用されます。

「IP接続」システムセクションを使用してPSS 4000装置で管理できるIP接続の最大数は、装置によって異なります。

ただし、「IP接続」システムセクションでは、最大32個のIP接続を管理できます。この上限は、「IP接続」システムセクションで、複数の異なるプロトコルや通信回線を使用している場合にも適用されます。

SafetyNET p接続は「IP接続」システムセクションで**管理されない**ため、この上限に影響しません。

- ▶ 受信接続に対して、1～1024 Byteまでデータ長 (外部入力データ) をコンフィグレーションできます。
- ▶ 送信接続に対して、1～1024 Byteまでデータ長 (外部出力データ) をコンフィグレーションできます。
- ▶ 送受信接続に対して、1～1024 Byteまで各送信方向のデータ長をコンフィグレーションできます。
- ▶ 各テレグラムで、コンフィグレーションできる最大データ長がサポートされます。つまり、送信接続から取得したデータは、複数のテレグラムに分散されません。
- ▶ テレグラムで送信されるデータのバイト数は、接続に対してコンフィグレーションされた数値と正確に一致する必要があります。
 - PSS 4000装置で受信したテレグラムのデータのバイト数が、接続に対してコンフィグレーションされた数値よりも大きかった場合、余ったバイト数のデータは受信が拒否されます。診断メッセージは診断リストには表示されません。これは、PSS 4000に対して定義された動作だからです。
 - PSS 4000装置で受信したテレグラムのデータのバイト数が、接続に対してコンフィグレーションされた数値よりも小さい場合、診断メッセージは診断リストに表示されます。
- ▶ 接続の受信データのFIFOメモリのサイズは装置により異なります。詳細については、関連するPSS 4000の取扱説明書を参照してください。

9.4.6 Raw UDPのI/Oデータ

「IP接続」システムセクションにより、Raw UDPを経由して、PSS 4000装置とサードパーティ装置の間でI/Oデータを交換できます。この操作を正常に実行するには、PAS4000のI/OマッピングエディタのI/Oデータに対応するRaw UDPデータ領域にマッピングする必要があります。

Raw UDPデータ領域のI/Oマッピング

I/Oマッピングを使用して、どのデータソースからどのデータシンクにデータを供給するかを定義します。

Raw UDPデータ領域で可能なI/Oマッピング:

I/Oマッピング		Raw UDP	
		受信データ (=外部入力データ)	送信データ (=外部出力データ)
モジュールバス	入力データ (例: ハードウェア入力)	---	◆
	出力データ (例: ハードウェア出力)	◆	---
PI 変数	入力PI変数	◆	---
	出力PI変数	---	◆

- ◆ I/Oマッピング可能
 --- I/Oマッピング不可

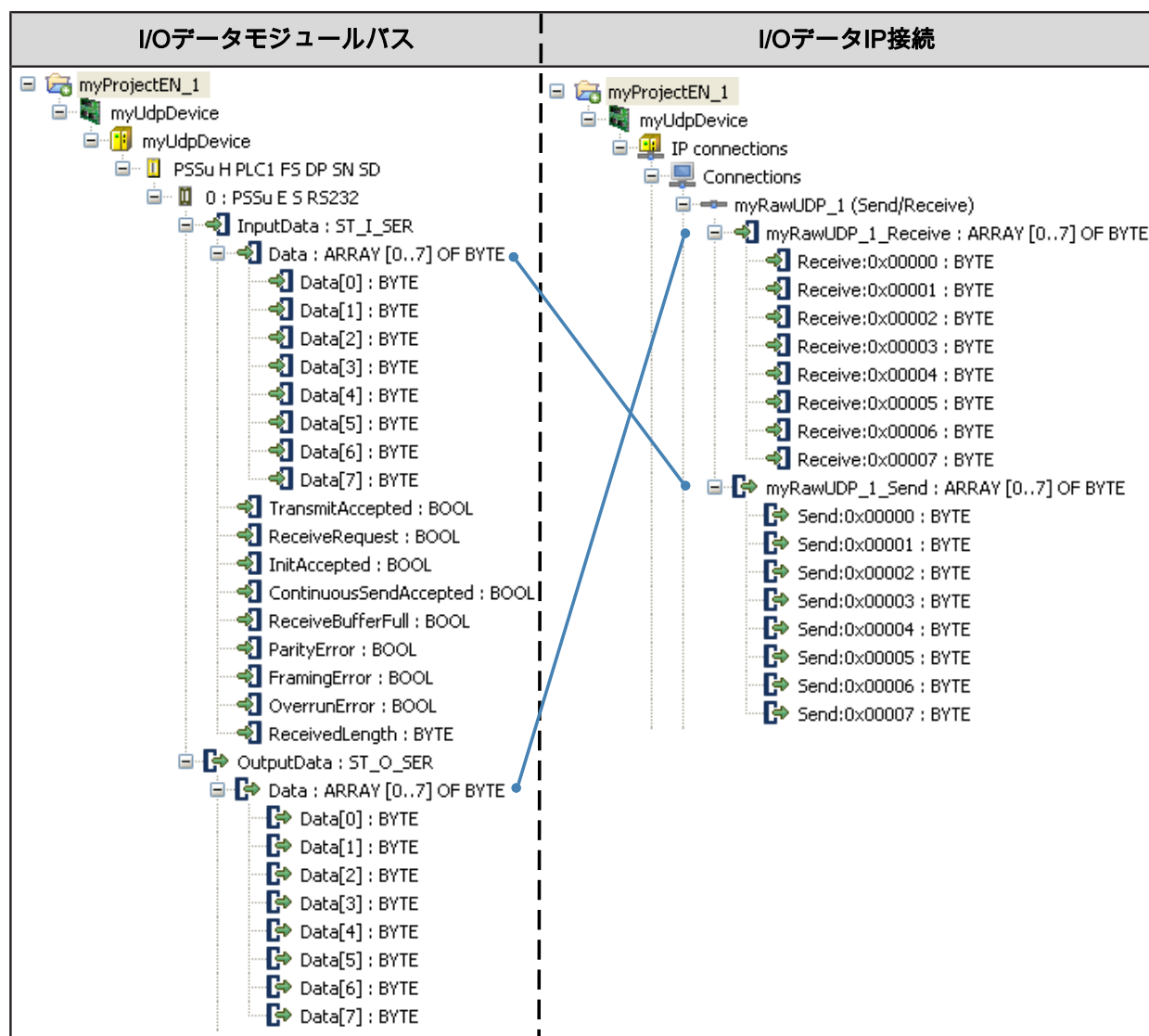
例

▶ I/OデータのI/Oマッピング

電子モジュールPSSu K S RS232の受信データは通信相手に送信され、通信相手が受信するデータはPSSu K S RS232で出力になります。

「myRawUDP_1」の送受信接続は、この目的のためにコンフィグレーションされます。

可能なI/Oマッピング (例):



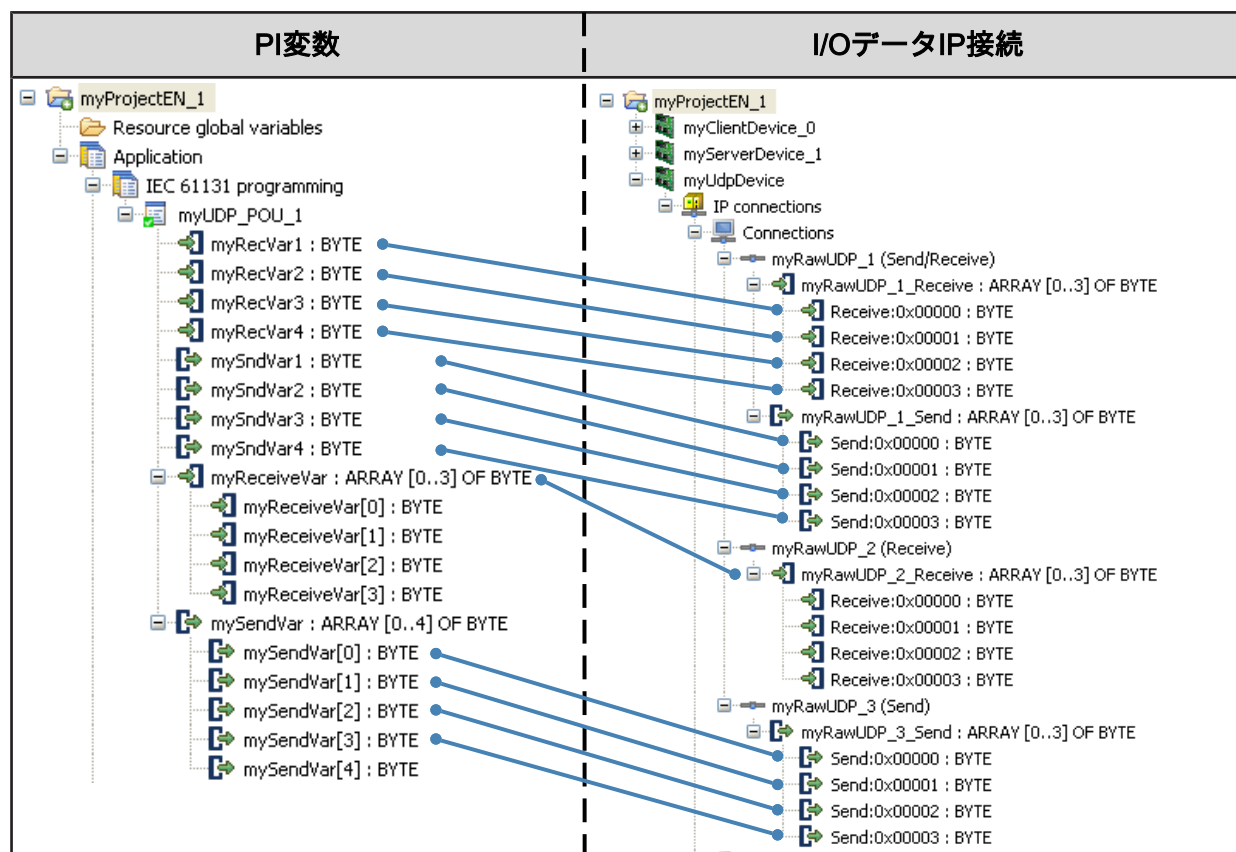
— I/Oマッピング

▶ PI変数のI/Oマッピング

ブロックの出力PI変数の内容は通信相手に送信され、入力PI変数は受信データに割り付けられます。この目的のために、3接続がコンフィグレーションされます。

- myRawUDP_1 (送受信接続)
- myRawUDP_2 (送信接続)
- myRawUDP_3 (受信接続)

可能なI/Oマッピング (例):



— I/Oマッピング

9.4.7 Raw UDPアドレスのアドレスフォーマット

PSS 4000装置のI/Oデータのアドレスは、PAS4000や診断など、複数の場所に表示されます。

9.4.7.1 送信データのアドレスフォーマット

Raw UDP送信データのアドレスは、次の形式で構造化されます。

<装置名>.UDP.<接続名>.Send.Byte[<ファンクション数>]

例:

myDevice_1.UDP.myTxConnection_1.Send.Byte[0]

～

myDevice_1.UDP.myTxConnection_1.Send.Byte[1023]

9.4.7.2 受信データのアドレスフォーマット

Raw UDP受信データのアドレスは、次の形式で構造化されます。

<装置名>.UDP.<接続名>.Receive.Byte.[<ファンクション数>]

例:

myDevice_1.UDP.myRxConnection_1.Receive.Byte[0]

～

myDevice_1.UDP.myRxConnection_1.Receive.Byte[1023]

9.4.8 Raw UDPのデータの整合性

受信データがサードパーティの装置で送信される環境では、このデータの整合性に対する責任は、これらの装置にあります。PSS 4000から見ると、これらの装置のデータの整合性に関してステートメントは作成できません。

Raw UDP接続の受信データのデータ整合性

PSS 4000専用／対応装置が、Raw UDP接続のデータの受信機である場合、PSS 4000から見ると、受信データは外部入力データになります。

Raw UDP接続からの外部入力データは、PSS 4000で一貫したデータとして処理されます ([「データの一貫性」](#)  [300](#))を参照)。

Raw UDP接続からの外部入力データは、整合性範囲タイプ「外部通信接続のST入力データ」の整合性範囲を形成します。正確な記述は、整合性範囲「Raw UDP接続のST-Iデータ」です ([「一貫性の範囲」](#)  [301](#))を参照)。

Raw UDP接続の送信データのデータ整合性

Raw UDP接続でPSS 4000専用／対応装置がデータの送信機である場合、PSS 4000から見ると、送信データは外部出力データになります。

データソースの同じ整合性範囲から取得した場合 ([「データの一貫性」](#)  [300](#))を参照)、Raw UDP接続の外部出力データには整合性があります。

9.5 ブロックを使用するUDP通信

ユーザデータグラムプロトコル (UDP) はRFC768に適合して実装されます。

UDPは、さまざまな通信相手との通信に使用されます。通信相手の通信サービスは、ポート番号経由で識別されます。UDPはアドレス処理にインターネットプロトコル (IP) を使用します。つまり、UDPデータグラムはIPフレームにカプセル化され、IPアドレスを使用してデータパケットとして通信相手に送信されます。

UDPは「非接続指向」または「無接続型」のプロトコルです。特性:

- ▶ 通信相手との接続は、送信を開始するまで確立されません。
- ▶ 次の結果が保証されません。
 - 送信データパケットが宛先に到達すること
 - 送信データパケットが宛先に一度だけ到達すること
 - データパケットが送信した順序で到達すること

したがって、アプリケーションが損失したデータグラムや分類されていないデータグラムを識別しない場合、または通信相手側でアプリケーション自体が適切な制御および修正メカニズムを提供する場合にのみ、UDPを使用することができます。

PSS 4000ユーザプログラムを使用してデータ転送を制御する場合は、ブロックを使用してUDP通信を使用するのが合理的です。UDPを介した周期的データ転送にはRaw UDPを使用するのが合理的です。

オートメーションシステムPSS 4000上で「ブロックを使用するUDP」を介してデータグラムを送受信するには、専用のPSS 4000システム制御ブロックをユーザプログラムで使用する必要があります。

- ▶ pssCreateUDPSocket - UDPソケットを作成
- ▶ PssReleaseUDPSocket - UDPソケットをリリース
- ▶ PssUDPSend - UDPデータグラムを送信
- ▶ PssUDPReceive - UDPデータグラムを受信

ブロックは、STリソースまたはFSリソースで使用できます。

ブロックを使用するUDP通信は、**非安全関連アプリケーション**用のオートメーションシステムPSS 4000で使用できます。安全関連アプリケーションでブロックを使用するUDP通信の使用を有効にするには、追加のセキュリティ対策が必要です。これらのセキュリティ対策は、アプリケーションを介して実装する必要があります。正しい実装はユーザの責任において行ってください。

9.5.1 「IP接続」システムセクション

オートメーションシステムPSS 4000では、「IP接続」システムセクションがあるすべてのPSS 4000装置で、ブロックを使用するUDP経由の外部通信を使用できます。この場合、PSS 4000装置の「IP接続」システムセクションが、通信に必要なタスクをすべて実行します。値には次のようなものがあります。

- ▶ UDPソケットの管理
- ▶ 「ブロックを使用するUDP」経由のデータ交換

9.5.2 UDPソケット

PSS 4000装置で専用のPSS 4000システム制御ブロックを使用してUDP経由でデータを交換するには、その前にIP接続システムセクションで少なくとも1つのUDPソケットが開かれている必要があります。UDPソケットを作成するには、PSS 4000システム制御ブロックpssCreateUDPSocketをタスクで実行する必要があります。その後、UDPソケット経由で他のPSS 4000システム制御ブロックとデータグラムを交換できます。単方向UDP通信 (送信専用または受信専用) と双方向UDP通信 (送受信) の両方をUDPソケット経由で実行できます。

UDPソケットは、PSS 4000装置のIP接続システムセクションによって管理されます。システムセクションは、最大9 UDPソケットまで管理できます。つまり、pssCreateUDPSocketを複数実行することで最大9個のUDPソケットをPSS 4000装置上で同時に開くことができます (「[データ転送の制限](#)」[\[183\]](#)」を参照)。

UDPソケットが作成されると、PSS 4000装置がソケットハンドルと呼ばれるUDPソケットに固有の基準値を生成します。生成されるソケットハンドルはブロックpssCreateUDPSocket上で発行されます。ユーザプログラムでブロックpssUDPSend/pssUDPReceive (データグラムの送受信) を実行するときには、データグラムの送受信に使用されるUDPソケットのソケットハンドルが指定されている必要があります。ブロックpssUDPSend/pssUDPReceiveは、常にブロックpssCreateUDPSocketと同じタスク内で実行される必要があります。

必要に応じて、UDPソケットを再度解放できます。これは、ユーザプログラムでブロックpssReleaseUDPSocketを実行することで行います。pssReleaseUDPSocketを実行するときには、解放されるUDPソケットのソケットハンドルを指定する必要があります。pssReleaseUDPSocketは、pssCreateUDPSocketと同じタスク内で実行される必要があります。UDPソケットが解放されると、このUDPソケット用に参照されているPSS 4000システム制御ブロックを使用してデータグラムの送受信を行うことができなくなります。

9.5.3 「ブロックを使用するUDP」経由のデータ交換

ユニキャスト通信では、ブロックを使用するUDP経由のデータ交換が推奨されます。詳細については、「[UDP経由の外部通信を使用する特別な機能](#)  125」を参照してください。1つの送信機と1つの受信機の間でデータが転送される1対1の通信関係は、ユニキャスト通信と呼ばれます。

PSS 4000では、UDPデータグラムの送信データは外部出力データと呼ばれ、UDPデータグラムの受信データは外部入力データと呼ばれます。

- ▶ PSS 4000システム制御ブロック「pssUDPSend」を使用したデータグラムの送信
pssUDPSendが実行されると、UDPデータグラムが生成され、タスクに関連付けられているUDP送信バッファに一時的に格納されます（「[UDP送信バッファとUDP受信バッファ](#)  182」を参照）。IP接続システムセクションは、UDPデータグラムが確実にUDP送信バッファから送信されるようにします。UDPデータグラムには、次のデータが含まれます。

- UDPデータグラムを受信したリモートデバイスのポートのポート番号
- 送信データの宛先のリモートデバイスのIPアドレス。
- ペイロード
- ペイロードのデータ長

pssUDPSendは、タスクの実行時間中に数回実行できます。ユーザプログラムがブロックが実行されるタイミングを決定します。つまり、各タスクサイクルでブロックを実行する必要はありません。pssUDPSendには、実行されるたびに異なるデータ（たとえば、異なるペイロード長、異なるリモートデバイス）を割り付けることができます。

- ▶ PSS 4000システム制御ブロック「pssUDPReceive」を使用したデータグラムの受信
IP接続システムセクションは、ネットワーク経由で受信したUDPデータグラムを読み取り、対象のタスクに関連付けられているUDP受信バッファにそれらが一時的に格納されるようにします。pssUDPReceiveが実行されるたびに、UDPデータグラムがUDP受信バッファから読み取られます（「[UDP送信バッファとUDP受信バッファ](#)  182」を参照）。ユーザプログラムがブロックが実行されるタイミングを決定します。つまり、各タスクサイクルでブロックを実行する必要はありません。UDPデータグラムには、次のデータが含まれます。

- PSS 4000装置のUDPソケットのポート番号
ポート番号は、UDPソケットを決定するために使用され、これによりUDPデータグラムの対象タスクが決定されます。
- 受信データの送信元のリモートデバイスのIPアドレス。
- ペイロード
- ペイロードのデータ長

9.5.4 UDP送信バッファとUDP受信バッファ

UDP経由のデータ交換のためにPSS 4000システム制御ブロックが実行される各タスクには、UDP送信バッファとUDP受信バッファがあります。

UDP送信バッファ

タスクのUDP送信バッファは、タスクの実行時間中にブロックpssUDPSendを実行することによって生成されるUDPデータグラムを一時的に格納するために使用されます。IP接続システムセクションは、タスクのUDP送信バッファからUDPデータグラムを取得し、ネットワーク経由でそれらを送信します。UDP送信バッファの必要なサイズは、コンフィグレーション済みの最大UDP送信速度から自動的に計算されます。

タスクの実行時間中にpssUDPSendが実行されるたびに、UDPデータグラムがタスクのUDP送信バッファに追加されます。

UDP受信バッファ

ネットワーク経由でIP接続システムセクションから受信したUDPデータグラムは、タスクのUDP受信バッファに一時的に格納されます。UDP受信バッファの必要なサイズは、コンフィグレーション済みの最大UDP受信速度から自動的に計算されます。

タスクの実行時間中にpssUDPReceiveが実行されるたびに、UDPデータグラムがタスクのUDP受信バッファから削除されます。UDPデータグラムは、最も古いものから時系列順に削除されます。

UDPデータグラムがタスクのUDP受信バッファから所定の時間内に取得されない場合、メモリ容量が不足している可能性があります。この場合、新しいUDPデータグラムをUDP受信バッファに格納できるようにするために、既存のUDPデータグラムが置き換えられます。UDPデータグラムは、最も古いUDPデータグラムから時系列順に置き換えられます。

9.5.5 データ転送の制限

「ブロックを使用するUDP」経由でデータを転送する場合、PSS 4000装置には次の制限が適用されます。

- ▶ IP接続システムセクションでは、最大9個のUDPソケットを管理できます。
9個のUDPソケットがすでにユーザプログラムで生成されているときに、別のUDPソケットの生成が試行された場合、「IP接続」システムセクションがそれを防止します (UDPソケットは生成されません)。
- ▶ ブロックpssCreateUDPSocketは、同じタスク内で数回実行できます。1つのタスクサイクル内で最大9個のUDPソケットを生成できます。
この制限の順守は監視されません。ただし、制限を超えると、正しい機能が保証できません。
- ▶ 1つのタスクサイクル内で最大9個のUDPソケットをリリースできます。
この制限の順守は監視されません。ただし、制限を超えると、正しい機能が保証できません。
- ▶ UDPデータグラムのペイロードの最大データ長が1464 Byteになるようにコンフィグレーションできます。
UDPデータグラムを送信するにはシステムデータ型UDP_DATAGRAMを使用する必要があります。このシステムデータ型を使用すると、この制限を超えることが防止されます。許可されるペイロードの最大値を超えるバイト数が含まれるUDPデータグラムを受信した場合、データグラムは拒否されます。
- ▶ 1つのタスクサイクル内で最大50個のUDPデータグラムを送信できます。UDPデータグラムはペイロード長が異なる場合があります。
最大50 UDPデータグラムという制限の順守は監視されません。ただし、制限を超えると、正しい機能が保証できません。
- ▶ 1つのタスクサイクル内で最大50個のUDPデータグラムを受信できます。UDPデータグラムはペイロード長が異なる場合があります。
最大50 UDPデータグラムという制限の順守は監視されません。ただし、制限を超えると、正しい機能が保証できません。
- ▶ 最大送信データ速度 (= 最大UDP送信速度) は、pssUDPSendが1回以上実行される各タスクについてPAS4000で定義する必要があります。
タスクの最大UDP送信速度には、128～8192の範囲の値をコンフィグレーションできます。既定の設定は、タスクサイクルあたり1024 Byteです。
- ▶ 最大受信データ速度 (= 最大UDP受信速度) は、pssUDPReceiveが1回以上実行される各タスクについてPAS4000で定義する必要があります。
最大UDP受信速度には、128～8192の範囲の値をコンフィグレーションできます。既定の設定は、ブロックを使用するUDP通信のサイクルあたり1024 Byteです。

9.5.6 UDPデータグラム送信時の時間動作

UDPデータグラムの送信を開始するには、タスクの実行時間中にpssUDPSendを実行します。UDPデータグラムを送信するときには次の条件が適用されます。

- ▶ pssUDPSendが実行された時点 (たとえば、タスクサイクル#m) からIP接続システムセクションがネットワーク経由でUDPデータグラムを送信する (タスクサイクル#m+3) までに最大4つのタスクサイクルが経過します。
- ▶ ブロックを使用するUDP通信のサイクル時間 (IP接続システムセクションがネットワーク経由でUDPデータグラムを送信するために使用する時間) は、pssUDPSendが実行されたタスクのタスクサイクル時間 t_{Task} に対応します。

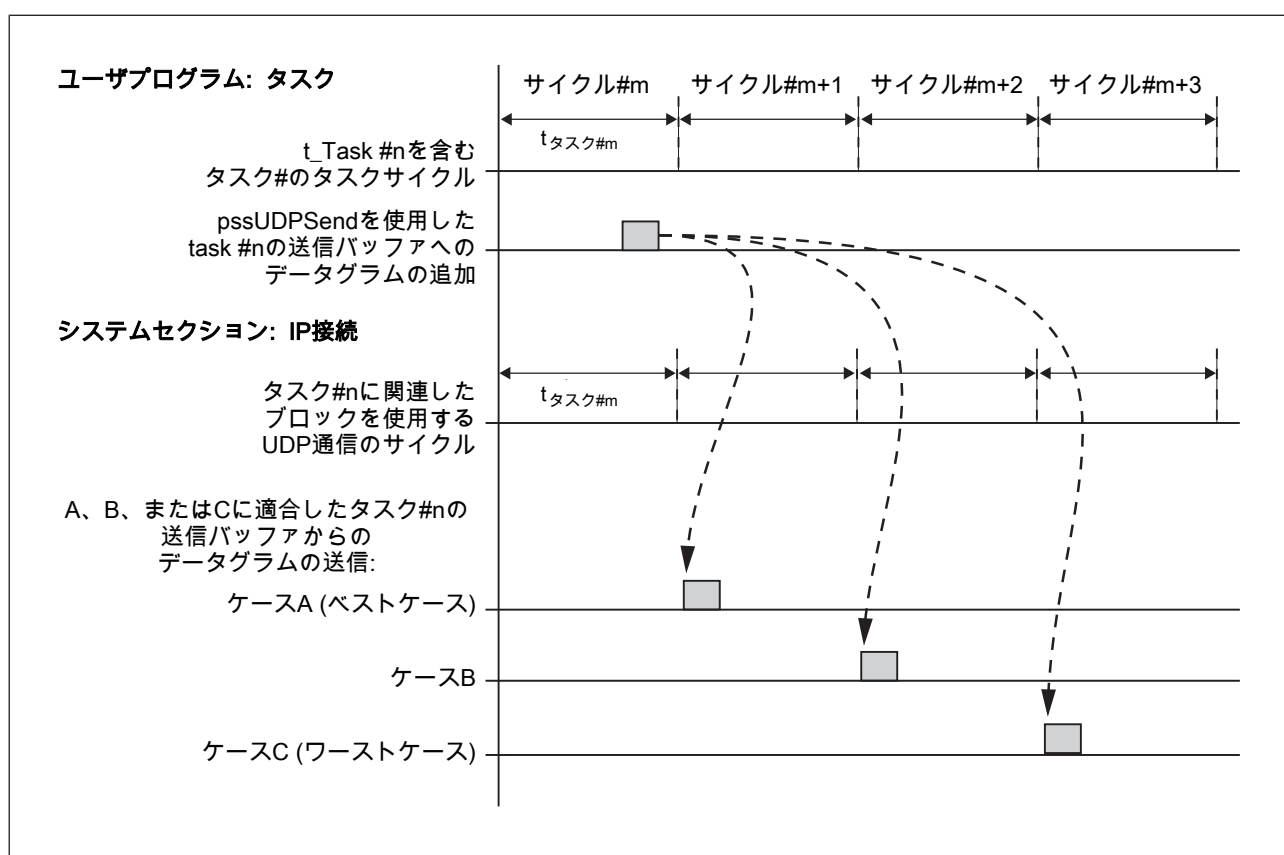


図: pssUDPSendを使用したUDPデータグラム送信時の時間動作

**情報**

同じタスクサイクル (たとえば、タスクサイクル#m) 内で、UDPソケットを作成するための `pssCreateUDPSocket` と、このUDPソケットを使用してUDPデータグラムを送信する `pssUDPSend` の両方を実行することができます。

UDPソケットは、その直後のタスクサイクル (タスクサイクル#m+1) で作成されます。IP接続システムセクションは、遅くとも、タスクサイクル#mに続くブロックを使用するUDP通信の3番目のサイクル (サイクル#m+3) までにこのソケットを使用するUDPデータグラムを送信します。

**重要**

タスクサイクル中に送信されるすべてのUDPデータグラムのペイロードバイト数は、最大でも、タスクでコンフィグレーションされている最大UDP送信速度 (タスクサイクルあたりのペイロードの最大バイト数) の半分になっている必要があります。つまり、可能な場合、コンフィグレーション済みの最大UDP送信速度の半分のみを使用する必要があります。この方法により、`pssUDPSend` の実行 (ユーザプログラム) とネットワーク経由のUDPデータグラムの送信 (IP接続システムセクション) の間の遅延を最小限に抑えることができます。

9.5.7 UDPデータグラム受信時の時間動作

UDPデータグラムを受信するときには次の条件が適用されます。

- ▶ IP接続システムセクションがネットワーク経由でUDPデータグラムを受信した時点から対象のタスクのUDP受信バッファでUDPデータグラムが使用可能になるまでに最大3つのタスクサイクルが経過します (「[ブロックを使用するUDP](#)」[経由のデータ交換](#)  181]」を参照)。

ブロックを使用するUDP通信のサイクル時間 (IP接続システムセクションがネットワークからUDPデータグラムを受信するために使用する時間) は、対象のタスクのUDPデータグラムのタスクサイクル時間 t_{Task} に対応します。

- ▶ pssUDPReceiveの実行からユーザプログラムでUDPデータグラムが使用可能になるまでの間に経過するタスクサイクルの数は、ユーザプログラムの設計によって異なります。



情報

同じタスクサイクル (たとえば、タスクサイクル#m) 内で、UDPソケットを作成するためのブロックpssCreateUDPSocketと、このUDPソケットを使用してUDPデータグラムを受信するブロックpssUDPReceiveの両方を実行することができます。

UDPソケットは、その直後のタスクサイクル (タスクサイクル#m+1) で作成されます。そのため、このUDPソケットを使用するUDPデータグラムがUDP受信バッファで使用可能になる最も早いタイミングは、1つおいて次のタスクサイクル (タスクサイクルm+2) です。



重要

ブロックを使用するUDP通信のサイクル中に受信されるすべてのUDPデータグラムのペイロードバイト数の平均は、最大でも、タスクでコンフィグレーションされている最大UDP受信速度の半分になっている必要があります。つまり、可能な場合、コンフィグレーション済みの最大UDP受信速度の半分のみを使用する必要があります。

この方法により、UDPデータグラムを受信するときの遅延を回避し、また対象のタスクのUDP受信バッファ内でUDPデータグラムが上書きされることを回避できます。

9.5.8 ブロックを使用するUDP通信中の時間の要件

ブロックを使用するUDP通信は、**非周期的な**伝送手順です。このため、ブロックを使用するUDPの通信時の時間要件はおおよそその時間のみ見積もることができます。原則的には、時間要件は、ユーザプログラムタスクに関連するUDPブロックが呼び出されたときに始まります。



重要

ブロックを使用するUDP通信は、外部通信としてカウントされ、IP接続システムセクションを使用しますが、それでも他のPSS 4000伝送手順とは異なる非周期的な伝送手順です。結果として、外部通信のサイクル時間 t_{ExtCo} が含まれる式を使用して時間要件を見積もることはできません。

時間動作への主な影響

- ▶ 時間動作は、UDPブロックが呼び出されるタスクの実行時間によって主に決まります。タスクの実行時間は、最大UDP送信速度と最大UDP受信速度（「[最大UDP送信速度と最大UDP受信速度](#)」[\[190\]](#)」を参照）および送受信したUDPデータグラムの数に応じて直線的に増加します。
- ▶ データグラムのサイズは、タスクの実行時間に対してほとんど影響しません。
- ▶ UDPブロックが異なるタスクで実行された場合、タスクの実行時間に特別な影響はありません。ただし、タスクは優先順位に従って実行され、その結果として実行時間が増加する場合があることに注意する必要があります（「[タスクのプロパティ](#)」[\[287\]](#)」および「[タスクの実行](#)」[\[288\]](#)」を参照）。

最大UDP送信速度と最大UDP受信速度の影響

コンフィグレーションされた最大UDP送信速度と最大UDP受信速度はタスクの実行時間を増加させます。この影響は、UDPデータグラムが実際に送信されたか受信されたかに関係なく存在します。次の関係が存在します。

$$\text{実行時間}_{\text{タスク (送信速度および受信速度)}} = \text{実行時間}_{\text{タスク}} + \text{時間}_{\text{UDP送信速度}} + \text{時間}_{\text{UDP受信速度}}$$



情報

実行時間 タスク (送信速度および受信速度) には、送受信されるUDPデータグラムの数の影響は含まれていません。

時間 UDP送信速度 および時間 UDP受信速度 は、おおよそその時間のみ見積もることができます。FSリソースとSTリソースを個別に考慮する必要があります。

FSリソースの時間要件を見積もるための一般的な規則

▶ 送信:

$$\text{時間}_{\text{UDP送信速度FS}} = (\text{バイト数}_{\text{最大UDP送信速度}} \times 0.9) \text{ マイクロ秒}$$

▶ 受信:

$$\text{時間}_{\text{UDP受信速度FS}} = (\text{バイト数}_{\text{最大UDP受信速度}} \times 1.3) \text{ マイクロ秒}$$

STリソースの時間要件を見積もるための一般的な規則

▶ 送信:

$$\text{時間}_{\text{UDP送信速度ST}} = (\text{バイト数}_{\text{最大UDP送信速度}} \times 0.3) \text{ マイクロ秒}$$

▶ 受信:

$$\text{時間}_{\text{UDP受信速度ST}} = (\text{バイト数}_{\text{最大UDP受信速度}} \times 0.24) \text{ マイクロ秒}$$

例

FSタスクのコンフィグレーション:

▶ 最大UDP送信速度 = 1500 Byte

▶ 最大UDP受信速度 = 2000 Byte

$$\text{実行時間}_{\text{タスク (送信速度および受信速度)}} = \text{実行時間}_{\text{タスク}} + (1500 \times 0.9) + (2000 \times 1.3) \text{ マイクロ秒}$$

結果:

FSタスクの実行時間は3950μs単位で増加します。

送受信されるUDPデータグラムの数の影響

送受信されるUDPデータグラムの数によってタスクの実行時間が増加しますが、UDPデータグラムのサイズはほとんど影響しません。次の関係が存在します。

$$\text{実行時間}_{\text{タスク (データグラム数)}} = \text{実行時間}_{\text{タスク}} + \text{時間}_{\text{UDP送信}} + \text{時間}_{\text{UDP受信}}$$



情報

実行時間_{タスク (データグラム数)} には、最大UDP送信速度および最大UDP受信速度の影響は含まれていません。

時間_{UDP送信} および時間_{UDP受信} は、おおよその時間のみ見積もることができます。FSリソースとSTリソースを個別に考慮する必要があります。最大100 Byteのパイロードを含むUDPデータグラムには次の一般的な規則が適用されます。

FSリソースの時間要件を見積もるための一般的な規則

▶ 送信:

$(\text{数}_{\text{データグラム}} \times 70) \text{ マイクロ秒} \leq \text{時間}_{\text{UDP送信FS}} \leq (\text{数}_{\text{データグラム}} \times 80) \text{ マイクロ秒}$

▶ 受信:

$(\text{数}_{\text{データグラム}} \times 100) \text{ マイクロ秒} \leq \text{時間}_{\text{UDP受信FS}} \leq (\text{数}_{\text{データグラム}} \times 120) \text{ マイクロ秒}$

これらの一般的な規則を使用して、タスクの実行時間は、タスク内で送信されるUDPデータグラムごとに約70～80μsずつ増加し、タスク内で受信されるUDPデータグラムごとに約100～120μsずつ増加すると推論できます。

STリソースの時間要件を見積もるための一般的な規則

▶ 送信:

$(\text{数}_{\text{データグラム}} \times 60) \text{ マイクロ秒} \leq \text{時間}_{\text{UDP送信ST}} \leq (\text{数}_{\text{データグラム}} \times 70) \text{ マイクロ秒}$

▶ 受信:

$(\text{数}_{\text{データグラム}} \times 80) \text{ マイクロ秒} \leq \text{時間}_{\text{UDP受信ST}} \leq (\text{数}_{\text{データグラム}} \times 100) \text{ マイクロ秒}$

これらの一般的な規則を使用して、タスクの実行時間は、タスク内で送信されるUDPデータグラムごとに約60～70マイクロ秒ずつ増加し、タスク内で受信されるUDPデータグラムごとに約80～100マイクロ秒ずつ増加すると推論できます。

全体的な時間要件

最大UDP送信速度および最大UDP受信速度の影響に加えて送受信されるUDPデータグラムの数の影響を考慮する必要があります。全体的な時間要件の計算には次の条件が適用されます。

$$\text{実行時間}_{\text{タスク全体}} = \text{実行時間}_{\text{タスク}} + \text{時間}_{\text{UDP送信速度}} + \text{時間}_{\text{UDP受信速度}} + \text{時間}_{\text{UDP送信}} + \text{時間}_{\text{UDP受信}}$$

9.5.9 最大UDP送信速度と最大UDP受信速度

PAS4000では、ブロックpssUDPSendおよびpssUDPReceiveを使用してUDP通信を実行する各タスクについて、最大UDP送信速度と最大UDP受信速度をコンフィグレーションする必要があります。

最大UDP送信速度

最大UDP送信速度は、pssUDPSendが1回以上実行されるときにタスクサイクル内で送信できるペイロードバイト数を制限します。

最大UDP送信速度の値を計算するための一般的な規則

タスクの平均送信速度を基にして、その値の2倍をPAS4000でタスクの最大UDP送信速度としてコンフィグレーションする必要があります。つまり、UDPデータグラムを使用してタスクサイクル内で送信されると予想されるペイロードバイト数は、コンフィグレーションされる最大UDP送信速度の半分以下になっている必要があります。IP接続システムセクションがネットワーク経由でUDPデータグラムを送信する際のサイクル時間は、タスクサイクル時間 t_{Task} と同じ大きさになります。

$$\text{最大UDP送信速度} \geq 2 \times \text{タスクサイクルあたりの最大送信ペイロードバイト数}$$

この一般的な規則を使用して最大UDP送信速度の値を計算し、PAS4000でコンフィグレーションした場合、UDPデータグラム送信時の遅延を回避できます。

最大UDP受信速度

最大UDP受信速度は、対象のタスクに関連して、タスクサイクル内でIP接続システムセクションがUDPデータグラムを介して受信できるペイロードバイト数を制限します。

最大UDP受信速度の値を計算するための一般的な規則

対象のタスクの平均受信速度を基にして、その値の2倍をPAS4000で対象のタスクの最大UDP受信速度としてコンフィグレーションする必要があります。つまり、UDPデータグラムを使用してタスクサイクル内で対象のタスク用に受信されると予想されるペイロードバイト数は、コンフィグレーションされる最大UDP受信速度の半分以下になっている必要があります。IP接続システムセクションが対象のタスク用のUDPデータグラムを受信する際のサイクル時間は、タスクサイクル時間 t_{Task} と同じ長さになります。

$$\text{最大UDP受信速度} \geq 2 \times \text{タスクサイクルあたりの最大受信ペイロードバイト数}$$

この一般的な規則を使用して最大UDP受信速度の値を計算し、PAS4000でコンフィグレーションした場合、UDPデータグラム受信時の遅延を防止できます。

9.5.10 ブロックを使用するUDP通信中のデータの整合性

この装置は、サードパーティの製造元の装置から送信されるUDPデータグラムの受信データ (ペイロード) のデータの整合性に責任を負います。PSS 4000から見ると、これらの装置のデータの整合性に関してステートメントは作成できません。

UDPデータグラムの受信データのデータ整合性

PSS 4000専用／対応装置が、UDPデータグラムからデータを受信する場合、装置から見て、受信データは外部入力データです。

UDPデータグラムからの外部入力データは、PSS 4000で整合性のあるデータとして処理されます (「[データの](#)
[一貫性](#)」 [300](#)」を参照)。

UDPデータグラムの外部入力データは、整合性範囲型「外部通信接続のST入力データ」の整合性範囲を形成します。正確な記述は、整合性範囲「ブロックを使用するUDP通信のST入力データ」です (「[一貫性の範囲](#)」 [301](#)」を参照)。

UDPデータグラムの送信データのデータ整合性

PSS 4000専用／対応装置が、UDPデータグラムを送信する場合、装置から見て、送信データは外部出力データです。

データソースの同じ整合性範囲から取得した場合、UDPデータグラムの外部出力データには整合性があります (「[データの整合性](#)」 [300](#)」を参照)。

9.5.11 ブロックを使用する安全なUDP通信

ブロックを使用するUDP通信は**非安全関連通信**です。これは特に、UDP経由のすべての伝送ルートに適用されます。安全関連アプリケーションでブロックを使用するUDP通信の使用を有効にするには、追加のセキュリティ対策が必要です。これらのセキュリティ対策は、アプリケーションを介して実装する必要があります。正しい実装はユーザの責任において行ってください。

次の点に注意してください。

▶ ブロックpssUDPSend

- FSリソースでブロックが実行される場合、ブロックは、安全関連の方法でデータグラムの生成元のデータを処理します。
- PSS 4000制御システムがデータグラムの生成中にエラーを検出した場合、IP接続システムセクションはデータグラムを送信しません。PSS 4000制御システムは、「FSの重大なエラー」に切り替わります。

▶ ブロックpssCreateUDPSocket、pssReleaseUDPSocket、およびpssUDPReceive

ブロックがFSリソース上で実行される場合でも、PSS 4000は、ブロックの安全関連の実行を保証できません。アプリケーションで追加のセキュリティ対策を実装する必要があります。

▶ UDP経由の伝送ルート

アプリケーションで伝送ルートの追加のセキュリティ対策 (ブラックチャネルの原則など) を実装する必要があります。



警告！

非安全関連通信による危険性

ブロックを使用するUDP通信は**非安全関連UDP通信**です。ブロックを使用するUDP通信からのデータは、追加のセキュリティ対策がないと安全関連タスクに**使用できません**。そのようにしないと、アプリケーションによっては重大な障害や死亡を招く可能性があります。

安全なUDP通信のために、次のことを確認する必要があります。

- 安全関連データがアプリケーションで使用されている。
- アプリケーションで伝送ルートの追加のセキュリティ対策 (ブラックチャネルの原則など) が実装されている。
- アプリケーションの安全を確認するために適切な手順 (たとえば、リスク分析と適切な試運転テスト、動作テスト) が使用されている。

9.6 PROFIBUS-DPスレーブ

PROFIBUSは、国際的な規格IEC 61158およびIEC 61784で通信が定義されたオープンなフィールドバス規格です。詳細な規定は、PROFIBUSユーザグループが公開した仕様で定義されています。これらの仕様は、PROFIBUS International (www.profibus.com を参照) から提供されています。

非安全関連アプリケーションの場合、PROFIBUS-DP対応のPSS 4000装置は、スレーブとしてPROFIBUS-DPに組み込むことができます。

PROFIBUS-DP対応のPSS 4000装置は、パフォーマンスレベルDP-V0、DP-V1、またはDP-V2で通信プロトコルPROFIBUS-DPをサポートするクラス1 DPマスタで使用できます。

PROFIBUS-DPスレーブとして、PROFIBUS-DP対応のPSS 4000装置には次の特徴があります。

- ▶ パフォーマンスレベルDP-V0での通信プロトコルPROFIBUS-DPの使用
- ▶ 装置アドレス126の使用時にDIPスイッチ (PSSuシステムのヘッドモジュール上など) またはPROFIBUS-DPマスタでアドレス処理される装置
- ▶ 選択可能な転送レート (9.6 kBits/s～12 MBits/s)
- ▶ 転送レートを自動検出するモードのサポート
(自動ボーモード)
- ▶ 同期モードのサポート
- ▶ フリーズモードのサポート
- ▶ PROFIBUS-DP診断テレグラムのサポート



情報

説明には、PROFIBUS規格によって設定および定義されているさまざまな専門用語が含まれています。「マスタ」や「スレーブ」といった用語が含まれます。

9.6.1 「PROFIBUS-DPスレーブ」システムセクション

オートメーションシステムPSS 4000の場合、「PROFIBUS-DPスレーブ」システムセクションがあるすべてのPSS 4000装置で、PROFIBUS-DP経由の外部通信を使用できます。この場合、「PROFIBUS-DPスレーブ」システムセクションが、通信に必要なタスクをすべて実行します。たとえば、これらのタスクには、送受信データ領域およびデータ交換の管理が含まれます（「[PROFIBUS-DPを経由したデータ交換](#)  195」を参照）。

接続は、PAS4000のPROFIBUS-DPスレーブエディタでコンフィグレーションされます。

9.6.2 PROFIBUS-DPを経由したデータ交換

「PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000装置は、PROFIBUS-DP経由でデータを交換する際に、PROFIBUS-DPスレーブとして動作します。これは、常にPROFIBUS-DPマスタとなり、データ転送を開始します。オートメーションシステムPSS 4000の視点では、PROFIBUS-DPマスタはシステムの外部にあります。

9.6.2.1 送信データと受信データ

PROFIBUS-DPマスタは、送信データ領域からPSS 4000装置の「PROFIBUS-DPスレーブ」システムセクションの受信データ領域にデータを送信し、PSS 4000装置の「PROFIBUS-DPスレーブ」システムセクションからのデータを要求します。PSS 4000装置の「PROFIBUS-DPスレーブ」システムセクションは、要求されたデータを送信データ領域からPROFIBUS-DPマスタの受信データ領域に送信します。オートメーションシステムPSS 4000では、これらのデータ領域を「仮想PROFIBUSモジュール」といいます。仮想PROFIBUSモジュールは、さまざまなデータ型で使用できます。

データ交換のために、PROFIBUS-DPマスタとPSS 4000装置の「PROFIBUS-DPスレーブ」システムセクションの両方に対して、対応する仮想PROFIBUSモジュールをコンフィグレーションする必要があります。送信データ用仮想PROFIBUSモジュールを、出力モジュールといいます。モジュール名には、「Output (出力)」を表す「O」の文字が入ります (8OX、OB、OWなど)。受信データ用仮想PROFIBUSモジュールを、入力モジュールといいます。モジュール名には、「Input (入力)」を表す「I」の文字が入ります (8IX、IB、IWなど)。

PROFIBUS DPマスタは、目的の仮想PROFIBUSモジュールをGSDファイルから取得します。このモジュールはPAS4000で提供され、ピルツホームページのダウンロード領域 (www.pilz.deのダウンロードページ) でも入手できます。たとえば「GSD」を検索語句として入力できます。ソフトウェアのダウンロードは、登録ユーザのみ実行できます。未登録ユーザの方は、無料で登録ができます。

PAS4000装置でPROFIBUS-DPスレーブがコンフィグレーションされる際に、PSS 4000装置の「PROFIBUS-DPスレーブ」システムセクションに必要な仮想PROFIBUSモジュールもコンフィグレーションされます。コンフィグレーション後に、仮想PROFIBUSモジュールが仮想スロットに配置されます (仮想スロット0～79)。

オートメーションシステムPSS 4000から見ると、受信データは外部入力データと呼び、送信データは外部出力データと呼びます (「[処理データの有効性](#)」 281」を参照)。

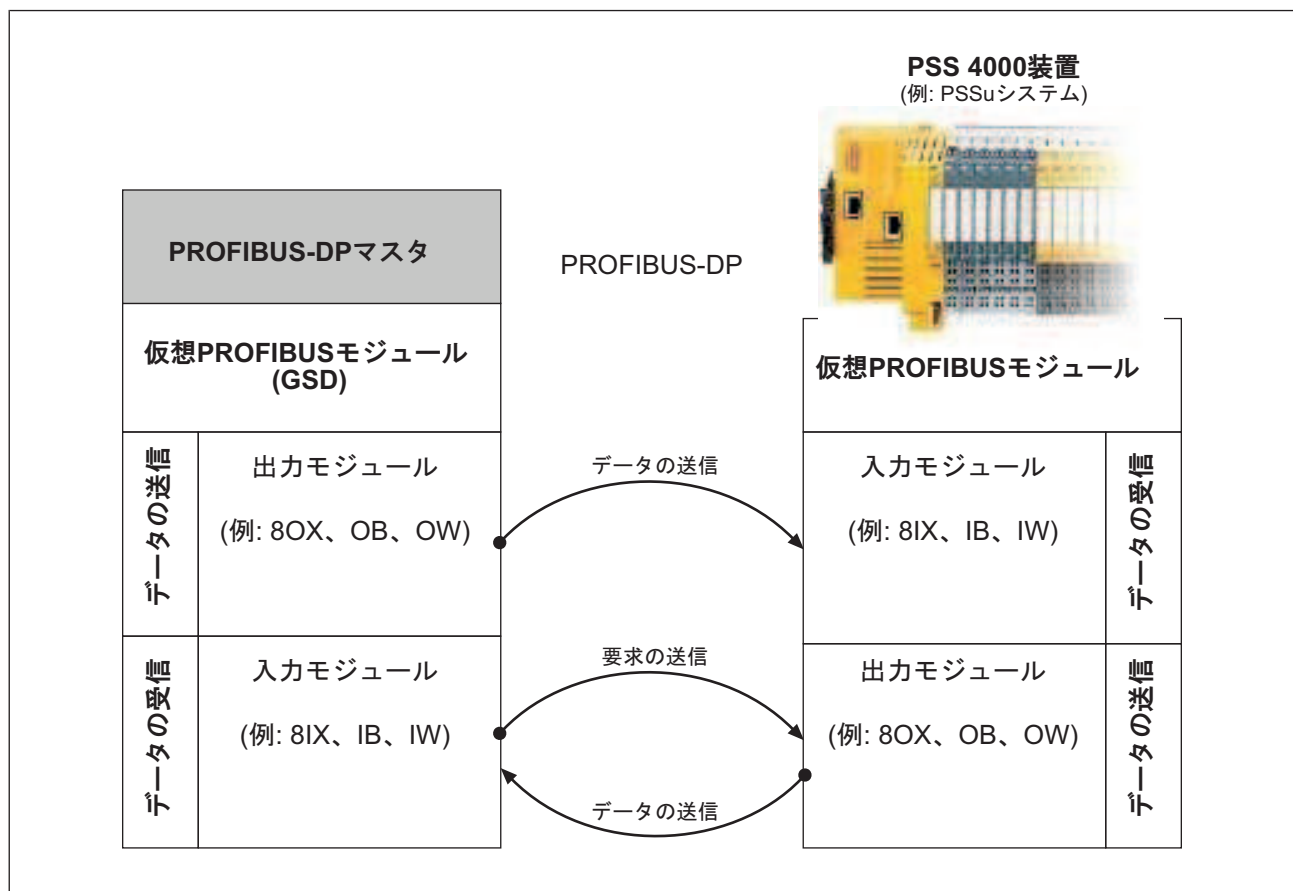


図: 仮想PROFIBUSモジュール経由でのデータ交換 (原則)

9.6.2.2 送信データ用仮想PROFIBUSモジュール

「PROFIBUS-DPスレーブ」システムセクションの送信データのために、PAS4000では、次の仮想PROFIBUSモジュール (出力モジュール) をコンフィグレーションできます。

仮想 PROFIBUSモジュール	データ型	アプリケーション
8OX	ARRAY [0..7] OF BOOL	仮想PROFIBUSビットモジュール データ型がBOOLの要素を8個持つ配列を送信
OB	BYTE	仮想PROFIBUSバイトモジュール データ型がBYTEのデータを送信
OW	WORD	仮想PROFIBUSバイトモジュール データ型WORDのデータを送信
OD	DWORD	仮想PROFIBUSバイトモジュール データ型DWORDのデータを送信
OBAn	ARRAY [0..(n-1)] OF BYTE	仮想PROFIBUS Byteモジュール データ型がBYTEの要素をn個持つ配列を送信 n = 2、4、8、16、32、64のいずれか
OWAn	ARRAY [0..(n-1)] OF WORD	仮想PROFIBUS Byteモジュール データ型がWORDの要素をn個持つ配列を送信 n = 2、4、8、16、32、64のいずれか



情報

ここでは、指定されているデータ型で記載されている仮想モジュールのみがサポートされていることに注意してください。LWORDデータ型はサポートされていません。

PROFIBUS-DPマスタは、データ交換時に、オートメーションシステムPSS 4000の出力モジュールからのデータを要求して、入力モジュール経由でこのデータを受信します。したがって、オートメーションシステムPSS 4000の出力モジュールには、PROFIBUS-DPマスタ内に対応する入力モジュールが必要です。またPROFIBUS-DPマスタの入力モジュールのデータ型は、オートメーションシステムPSS 4000の出力モジュールのデータ型に対応する必要があります。

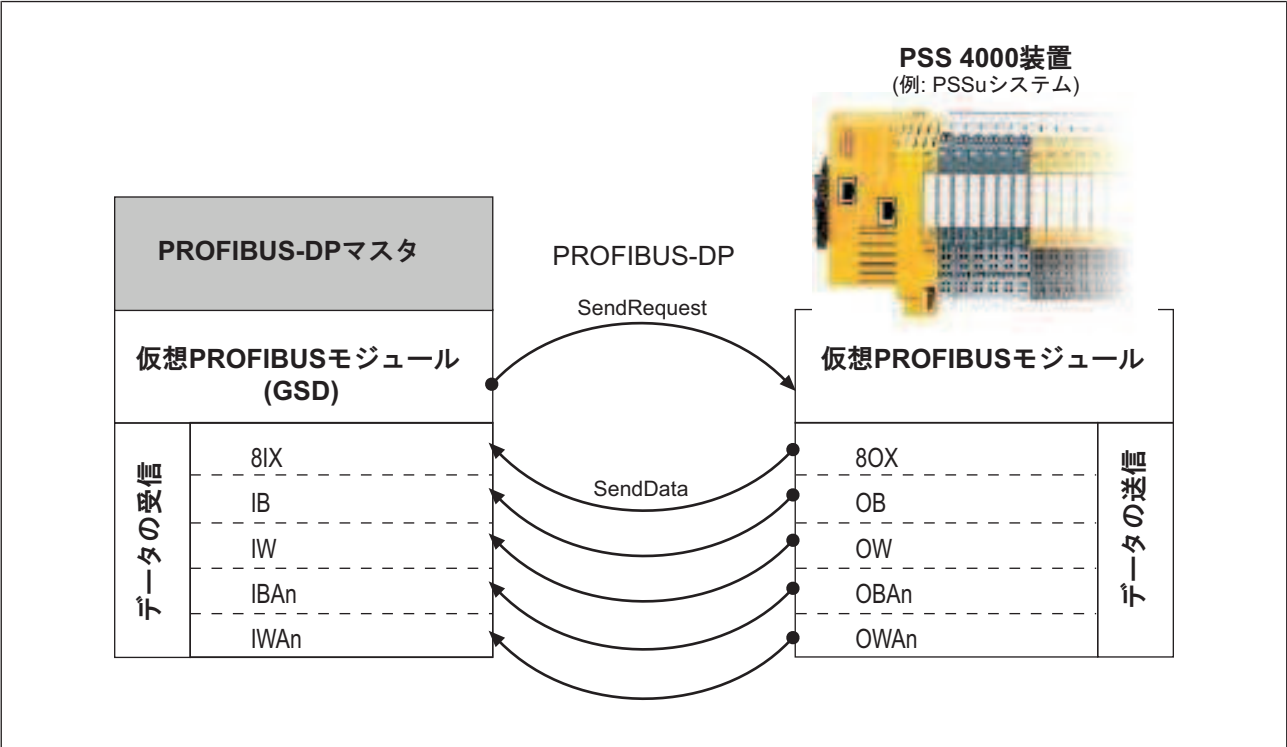


図: 仮想PROFIBUSモジュールと対応するデータ型 (PSS 4000の送信データ)

9.6.2.3 受信データ用仮想PROFIBUSモジュール

「PROFIBUS-DPスレーブ」システムセクションの受信データのために、PAS4000では、次の仮想PROFIBUSモジュール (入力モジュール) をコンフィグレーションできます。

仮想 PROFIBUSモジュール	データ型	アプリケーション
8IX	ARRAY [0..7] OF BOOL	仮想PROFIBUSビットモジュール データ型がBOOLの要素を8個持つ配列を受信
IB	BYTE	仮想PROFIBUSバイトモジュール データ型がBYTEのデータを受信
IW	WORD	仮想PROFIBUS Byteモジュール データ型WORDのデータを受信
ID	DWORD	仮想PROFIBUSバイトモジュール データ型DWORDのデータを受信
IBAn	ARRAY [0..(n-1)] OF BYTE	仮想PROFIBUS Byteモジュール データ型がBYTEの要素をn個持つ配列を受信 n = 2、4、8、16、32、64のいずれか
IWAn	ARRAY [0..(n-1)] OF WORD	仮想PROFIBUS Byteモジュール データ型がWORDの要素をn個持つ配列を受信 n = 2、4、8、16、32、64のいずれか



情報

ここでは、指定されているデータ型で記載されている仮想モジュールのみがサポートされていることに注意してください。LWORDデータ型はサポートされていません。

PROFIBUS-DPマスタは、データ交換時に、オートメーションシステムPSS 4000の入力モジュールに出力モジュール経由で送信データを送信します。したがって、PROFIBUS-DPマスタの出力モジュールには、オートメーションシステムPSS 4000内に対応する入力モジュールが必要です。また入力モジュールのデータ型は、PROFIBUS-DPマスタの出力モジュールのデータ型に対応する必要があります。

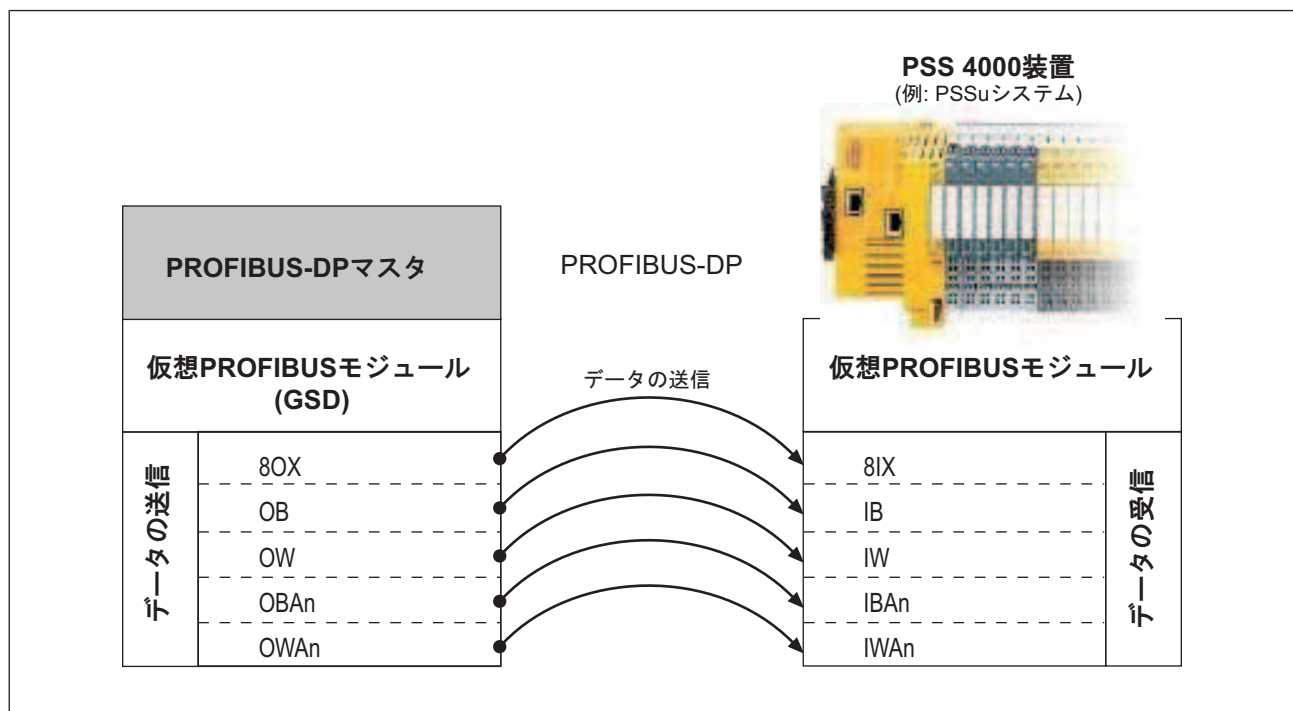


図: 仮想PROFIBUSモジュールとその対応するデータ型 (PSS 4000の受信データ)

9.6.3 データ転送の制限

仮想PROFIBUSモジュール経由のデータ転送には、以下の制限が適用されます。

- ▶ PSS 4000の「PROFIBUS-DPスレーブ」システムセクションに対して、仮想スロット0～79を使用して、最大80の仮想PROFIBUSモジュールをコンフィグレーションできます。
- ▶ PSS 4000の「PROFIBUS-DPスレーブ」システムセクションに対して、最大244 Byteの受信データ (=外部入力データ) をコンフィグレーションできます。
- ▶ PSS 4000の「PROFIBUS-DPスレーブ」システムセクションに対して、最大244 Byteの送信データ (=外部出力データ) をコンフィグレーションできます。

9.6.4 PROFIBUS-DPスレーブのI/Oデータ

「PROFIBUS-DPスレーブ」システムセクションにより、PSS 4000とPROFIBUS-DPマスタは、PROFIBUS-DPを経由してI/Oデータを交換できます。この操作を正常に実行するには、PAS4000のI/OマッピングエディタのI/Oデータを対応する仮想PROFIBUSモジュールにマッピングする必要があります。

仮想PROFIBUSモジュールのI/Oマッピング

I/Oマッピングを使用して、どのデータソースからどのデータシンクにデータを供給するかを定義します。

仮想PROFIBUSモジュールの可能なI/Oマッピング:

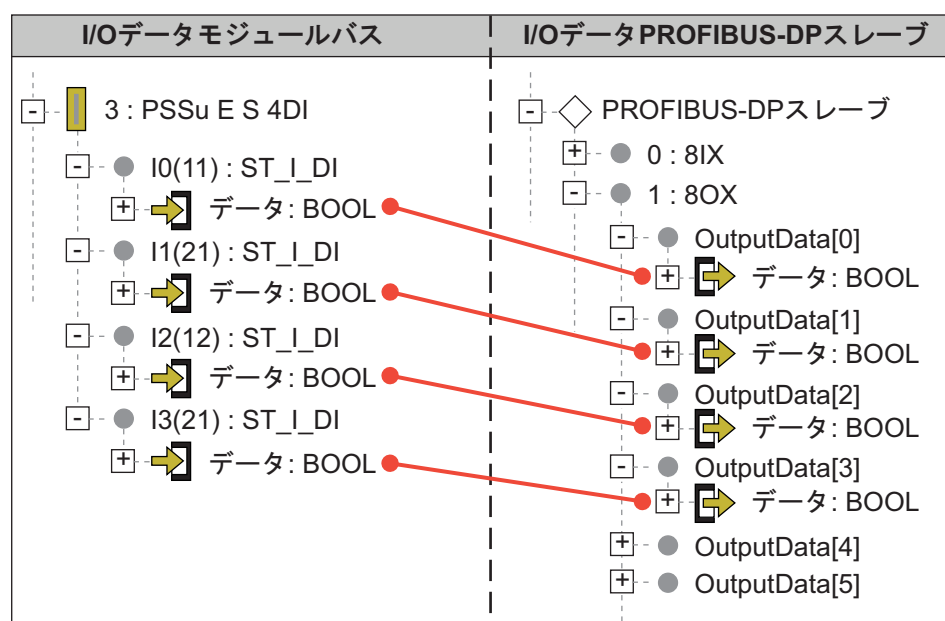
		仮想PROFIBUSモジュール	
		データの受信 (= 8IX、IB、IW、ID、IBAn、IWAn など、入力モジュールの外部入力データ)	データの送信 (= 8OX、OB、OW、OD、OBAn、 OWAnなど、出力モジュールの外部出力データ)
モジュールバス	入力データ (例: ハードウェア入力)	---	◆
	出力データ (例: ハードウェア出力)	◆	---
PI 変数	入力PI変数	◆	---
	出力PI変数	---	◆

- ◆ I/Oマッピング可
 --- I/Oマッピング不可

例

送信データがある仮想PROFIBUSモジュールのI/Oマッピング

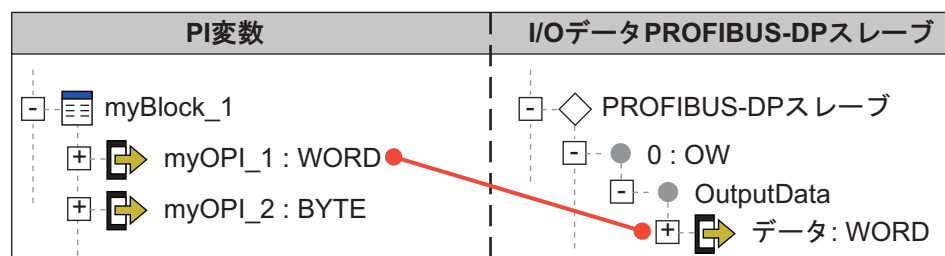
- ▶ PSSu E S 4DIのハードウェア入力の入力信号は、PROFIBUS DPマスタに転送されます。
 - データフローの方向:ハードウェア入力 -> PROFIBUS DP
 - 使用する仮想PROFIBUSモジュール:8OX
 - 可能なI/Oマッピング (例):



●—● I/Oマッピング

▶ ブロックの出力PI変数の内容は、PROFIBUS DPマスタに転送されます。

- データフローの方向: 出力PI変数 -> PROFIBUS DP
- 使用する仮想PROFIBUSモジュール: OW
- 可能なI/Oマッピング (例):

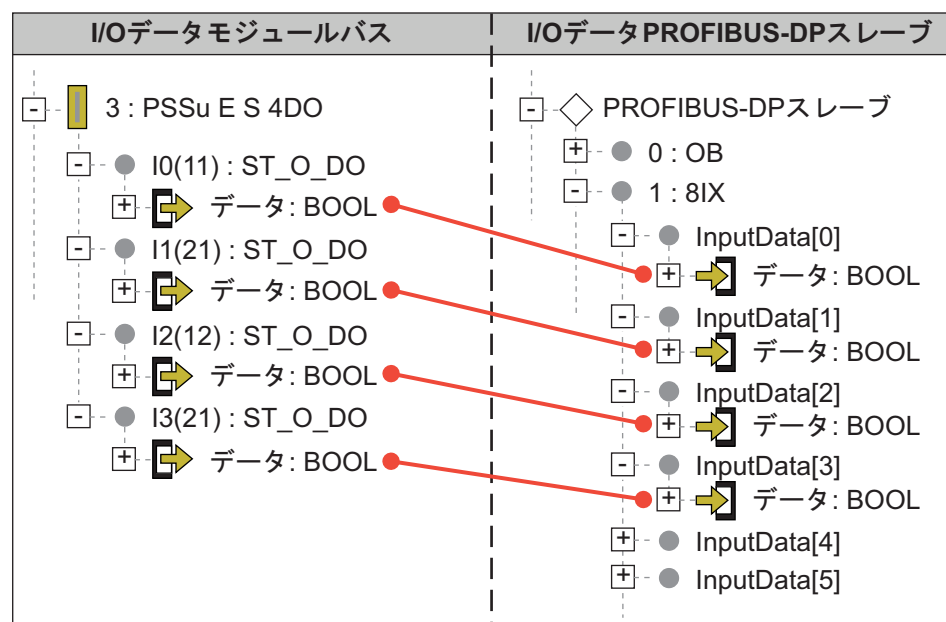


●—● I/Oマッピング

受信データがある仮想PROFIBUSモジュールのI/Oマッピング

▶ PROFIBUS-DPマスタは、PSSu E S 4DO 0.5のハードウェア出力を制御します。

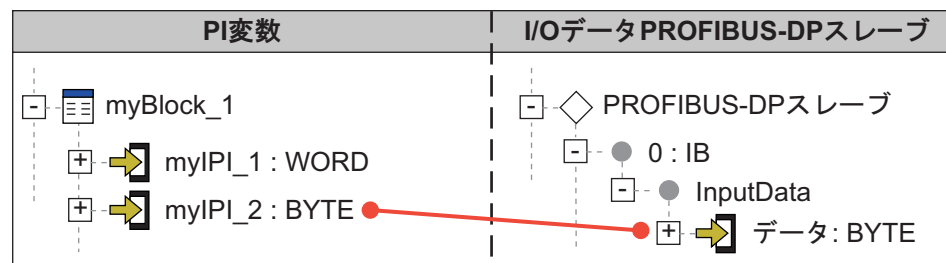
- データフローの方向: PROFIBUS-DP -> ハードウェア出力
- 使用する仮想PROFIBUSモジュール: 8IX
- 可能なI/Oマッピング (例):



●—● I/Oマッピング

▶ データは、PROFIBUS-DPマスタからPSS 4000装置に転送されます。一度データを受信すると、ブロック内でさらに処理できるようになります。

- データフローの方向: PROFIBUS-DP -> 入力PI変数
- 使用する仮想PROFIBUSモジュール: IB
- 可能なI/Oマッピング (例):



●—● I/Oマッピング

9.6.5 PROFIBUS-DPアドレスのアドレス表示

PSS 4000装置のI/Oデータのアドレスは、PAS4000や診断など、複数の場所に表示されます。

9.6.5.1 送信データのアドレスフォーマット

PROFIBUS-DP送信データのアドレス (仮想PROFIBUSモジュールの出力データ) は、次の形式で構造化されます。

<装置名>.ProfibusDP.<仮想スロット>.OutputData[<ファンクション番号>]

例:

▶ 8OXのアドレスフォーマット:

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]

▶ OBのアドレスフォーマット:

myDevice_1.ProfibusDP.3.OutputData

▶ OWのアドレスフォーマット:

myDevice_1.ProfibusDP.4.OutputData

▶ OBA2のアドレスフォーマット:

myDevice_1.ProfibusDP.5.OutputData[0]
myDevice_1.ProfibusDP.5.OutputData[1]

▶ OWA4のアドレスフォーマット:

myDevice_1.ProfibusDP.6.OutputData[0]
myDevice_1.ProfibusDP.6.OutputData[1]
myDevice_1.ProfibusDP.6.OutputData[2]
myDevice_1.ProfibusDP.6.OutputData[3]

9.6.5.2 受信データのアドレスフォーマット

PROFIBUS-DP受信データのアドレス (仮想PROFIBUSの入力データ) は、次の形式で構造化されます。

<装置名>.ProfibusDP.<仮想スロット>.InputData[<ファンクション番号>]

例:

▶ 8IXのアドレスフォーマット:

```
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]
```

▶ IBのアドレスフォーマット:

```
myDevice_1.ProfibusDP.8.InputData
```

▶ IWのアドレスフォーマット:

```
myDevice_1.ProfibusDP.9.InputData
```

▶ IBA4のアドレスフォーマット:

```
myDevice_1.ProfibusDP.10.InputData[0]
myDevice_1.ProfibusDP.10.InputData[1]
myDevice_1.ProfibusDP.10.InputData[2]
myDevice_1.ProfibusDP.10.InputData[3]
```

▶ IWA2のアドレスフォーマット:

```
myDevice_1.ProfibusDP.11.InputData[0]
myDevice_1.ProfibusDP.11.InputData[1]
```

9.6.6 PROFIBUSの診断

PROFIBUS-DPの診断は、PSS 4000のシステム診断を通じて実行できます (「[診断の基本情報](#)」[\[566\]](#)を参照)。

ただし、規格ではPROFIBUS-DPの独自の診断オプションが定義されています。PROFIBUS-DPマスタは、診断テレグラムを通じて、PROFIBUS-DPスレーブで現在の診断情報をポーリングできます。

規格では、診断テレグラムの診断情報は次の分野に分類されています。

- ▶ [システム内部の診断](#) [\[208\]](#)
- ▶ [外部診断](#) [\[209\]](#)

システムの内部診断データは診断テレグラムの最初の6 Byteに格納され、その意味は規格で厳密に指定されています。これらの必須の6 Byteの後には、外部診断データを含む1つ以上のブロックが続きます。

「PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000では、PROFIBUS DP診断テレグラムは、最小6 Byteの診断データ (必須の診断データ) と最大12 Byteの診断データ (6 Byteの必須診断データ + 6 Byteの外部診断データ) で構成されます。

診断情報は、GSDファイルの一部です。

9.6.6.1 システム内部の診断

規格に従い、診断テレグラムはシステム内部の診断データの6 byteで始まります (標準診断)。システム内部の診断データの構造、内容、意味は規格で厳密に指定されています。

システム内部の診断データの構造と内容

バイト番号	ビット番号	意味
1	0	Station_Non_Existent
	1	Station_Not_Ready
	2	Cfg_Fault
	3	Ext_Diag 0:外部診断データなし 1:外部診断データあり
	4	Not_Supported
	5	Invalid_Slave_Response
	6	Prm_Fault
	7	Master_lock
2	0	Prm_req
	1	Stat_Diag
	2	常に「1」
	3	WD_on
	4	Freeze_Mode
	5	Sync_Mode
	6	予約
	7	無効
3	6 ... 0	予約
	7	Ext_Diag_Overflow
4	7 ... 0	PROFIBUS-DPマスタのアドレス
5	7 ... 0	PROFIBUS-DPスレーブの識別番号:上位バイト PSSuシステム:0C
6	7 ... 0	PROFIBUS-DPスレーブの識別番号:下位バイト PSSuシステム:B0



情報

Byte 1～6の正確な意味については、IEC 61158-6-3 (5.3.1) とIEC 61158-6-5 (6.2.3.3.1) を参照してください。

9.6.6.2 外部診断データ

規格に従い、診断ブロックはヘッダバイトから始まります。ヘッダバイトは、診断ブロックのタイプと長さを示します。規格では、次の3通りのフォーマットが定義されています。

- ▶ 装置ベースの診断
- ▶ IDベースの診断
- ▶ チャンネルベースの診断

PSS 4000では、**装置ベースの診断**がサポートされています。「PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000装置の装置ベースの診断データには、ヘッダバイトを含め6 Byteが必要です。

「PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000装置の装置ベースの診断データの構造と内容

バイト番号	ビット番号	値	意味
n	0 ... 7	06h	ヘッダ
			ビット7と6: 00 「PROFIBUS-DPスレーブ」システムセクションがある装置: 装置ベースの診断のコード
			ビット5 ... 0: 000110 装置ベースの診断のバイト数 (ヘッダバイトを含む) 「PROFIBUS-DPスレーブ」システムセクションがある装置が6 Byteを送る
n + 1	予約済み		
n + 2	予約済み		
n + 3	予約済み		

バイト番号	ビット番号	値	意味
n + 4	0	0/1	装置に関するエラーメッセージ
			0 メッセージなし
			1 装置の1つ以上のシステムセクションに重大度が「エラー」レベルのメッセージがある または 装置の1つ以上のFSシステムセクションにFSの重大エラーがある PSS 4000専用装置では、次の [DIAG] ステータスLEDに対応しています。ステータス: 赤の点灯または点滅
	1	0/1	装置のFSまたはSTモジュールバスシステムセクションのエラー
			0 メッセージなし
			1 オペレーティングステータス「装置のすべてのFS出力が安全状態」 または 少なくとも1つのモジュールにアクセスできない (操作中にモジュールが削除された、設定されたハードウェアレジストリが実際の値と一致しないなど) または オペレーティングステータス「FSモジュールバスがエラーでSTOP状態: FSの重大エラー」 PSS 4000専用装置では、次の [MBUS] ステータスLEDに対応しています。ステータス: 赤の点灯または点滅
	2	0/1	PSS 4000装置の「FS SafetyNET p」システムセクションのエラー
			0 メッセージなし
			1 オペレーティングステータス「FS SafetyNET pがエラーでSTOP状態: FSの重大エラー」 または オペレーティングステータス「FS SafetyNET pがエラーでSTOP状態: Major FS+ST error」 PSS 4000専用装置では、次の [FS SNp] ステータスLEDに対応しています。ステータス: 赤の点灯または点滅
	3	0/1	装置の「ST SafetyNET p」システムセクションのエラー
			0 メッセージなし
			1 オペレーティングステータス「ST SafetyNET pがエラーでSTOP状態: Major FS+ST error」 PSS 4000専用装置では、次の [ST SNp] ステータスLEDに対応しています。ステータス: 赤の点灯または点滅
	4 ... 7	0	予約済み

バイト番号	ビット番号	値	意味
n + 5	0	0/1	装置に関する警告メッセージ
			0 メッセージなし
			1 装置に重大度が「警告」レベルのメッセージがある PSS 4000専用装置では、次の [DIAG] ステータスLEDに対応しています。ステータス: オレンジで点灯
	1	0/1	FSリソースに対する強制出力
			0 メッセージなし
			1 FSリソースに対する強制出力が有効であるか、1つ以上のオンライン変更が有効 (またはその両方) PSS 4000専用装置では、次の [FS FORCE] ステータスLEDに対応しています。ステータス: 黄色で点灯
	2	0/1	STリソースに対する強制出力
			0 メッセージなし
			1 STリソースに対する強制出力が有効であるか、1つ以上のオンライン変更が有効 (またはその両方) PSS 4000専用装置では、次の [ST FORCE] ステータスLEDに対応しています。ステータス: 黄色で点灯
	3 ... 7	0	予約済み

9.6.7 PROFIBUS-DPのデータの整合性

PROFIBUS-DPマスタとPSS 4000の間で交換されるデータのデータ整合性は、PROFIBUS-DPマスタのコンフィグレーションによって決まります。

PROFIBUS-DPは、3つのタイプのデータの整合性をサポートします。

▶ バイトの整合性

バイトは、その単位全体として送信されます。

▶ ワードの整合性

1ワード内の2 Byteは常に一緒に送信されます。

▶ バッファの整合性

データバッファ全体が、分割されないで1つの単位として送信されます。

PSS 4000の視点から見ると、データバッファは、送信データまたは受信データのそれぞれの仮想PROFIBUSモジュールに対応しています。

受信データの整合性

PSS 4000の視点では、仮想PROFIBUSモジュールの受信データは、外部入力データになります。

外部入力データが仮想PROFIBUSモジュールで使用可能になると、すぐに一貫したデータとして処理されます (「[データの一貫性](#)」 300」を参照)。PROFIBUSモジュールからの外部入力データは、整合性範囲タイプ「外部通信接続のST入力データ」の整合性範囲を形成します。正確な記述は、整合性範囲「PROFIBUS-DP接続のST-Iデータ」です (「[一貫性の範囲](#)」 301」を参照)。

送信データの整合性

PSS 4000の視点では、仮想PROFIBUSモジュールの送信データは、外部出力データになります。

データソースの同じ整合性範囲から取得した場合、仮想PROFIBUSモジュールの外部出力データには整合性があります。

PSS 4000では、仮想PROFIBUSモジュールの外部出力データは、常に1つの単位として処理されます。必要なデータの整合性を保証することは、PROFIBUS-DPマスタの責任です。

9.7 PROFIsafe搭載PROFIBUS-DPスレーブ

安全関連の通信のために、非安全関連のフィールドバスシステムPROFIBUS-DPに、PROFIsafe機能を追加することができます。この場合、ブラックチャンネルの原則に適合して、フィールドバスシステムに、PROFIsafeプロトコルが導入されます。PROFIsafe経由の通信は、国際規格IEC 61784-3-3で定義されています。

▶ PROFIBUS-DP

「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000は、**非安全関連**のアプリケーションのために、スレーブとしてPROFIBUS-DPに組み込むことができます。

▶ PROFIsafe

「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000は、**安全関連**のアプリケーションのために、F装置 (Fスレーブ) としてPROFIsafeに組み込むことができます。

PROFIBUS-DPスレーブおよび／またはF装置として動作するPSS 4000は、パフォーマンスレベルDP-V0、DP-V1、またはDP-V2で通信プロトコルPROFIBUS-DPをサポートするクラス1 DPマスタで使用できます。

PROFIBUS-DP用にサポートされる機能

- ▶ PROFIBUS-DPを通じたデータ交換に、パフォーマンスレベルDP-V0の通信プロトコルPROFIBUS-DPを使用
- ▶ PROFIsafeを通じたデータ交換に、パフォーマンスレベルDP-V1の通信プロトコルPROFIBUS-DPを使用
- ▶ 装置アドレス126をDIPスイッチで使用するときにDIPスイッチ (PSSuシステムのヘッドモジュール上など) またはPROFIBUS-DPマスタでアドレス処理される装置
- ▶ 選択可能な転送レート (9.6 kBits/s～12 MBits/s)
- ▶ 転送レートを自動検出するモードのサポート (自動ボーモード)
- ▶ 同期モードのサポート
- ▶ フリーズモードのサポート
- ▶ PROFIBUS-DP診断テレグラムのサポート
- ▶ 事前定義済み仮想PROFIBUSモジュールを経由したデータ交換 (入力モジュール、出力モジュール)

PROFIsafe用にサポートされる機能

- ▶ PROFIsafeを通じたデータ交換に、パフォーマンスレベルDP-V1の通信プロトコルPROFIBUS-DPを使用
- ▶ PROFIsafeアドレス0をDIPスイッチで使用するときにDIPスイッチ (PSSuシステムのヘッドモジュール上など) またはPAS4000で実行される、PROFIsafeアドレスの割り付け
- ▶ マスタツール (Siemens-Software HW Configなど) でのFパラメータの表示と編集
- ▶ 事前定義済み仮想Fモジュールを経由したデータ交換



情報

説明には、PROFIBUS規格によって設定および定義されているさまざまな専門用語が含まれています。「マスタ」や「スレーブ」といった用語が含まれます。

9.7.1 「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクション

オートメーションシステムPSS 4000の場合、「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションがあるすべてのPSS 4000で、PROFIBUS-DPおよび／またはPROFIsafe経由の外部通信を使用できます。この場合、「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションが、通信に必要なタスクをすべて実行します。例:

- ▶ 送受信データ領域の管理
- ▶ PROFIBUS-DPを経由したデータ交換
- ▶ PROFIsafeを経由したデータ交換

9.7.2 「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションのPROFIBUS-DP

「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000は、PROFIBUS-DP経由で**非安全関連データ**を送信します。PROFIBUS-DP経由でデータを交換する場合、これらはPROFIBUS-DPスレーブとして動作します。これは、常にPROFIBUS-DPマスタとなり、データ転送を開始します。オートメーションシステムPSS4000の視点では、PROFIBUS-DPマスタはシステムの外部にあります。つまり、受信データは外部入力データと呼ばれ、送信データは外部出力データと呼ばれます（「[処理データの有効性](#)」[\[281\]](#)を参照）。

送信データと受信データ

PROFIBUS-DPマスタは、送信データ領域からPSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションの受信データ領域にデータを送信し、PSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションからのデータを要求します。

PSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションは、要求されたデータを送信データ領域からPROFIBUS-DPマスタの受信データ領域に送信します。

オートメーションシステムPSS 4000では、これらのデータ領域を「仮想PROFIBUSモジュール」といいます。受信データ用仮想PROFIBUSモジュールを、入力モジュールといいます。モジュール名には、「Input (入力)」を表す「I」の文字が入ります (8IX、IB、IWなど)。送信データ用仮想PROFIBUSモジュールを、出力モジュールといいます。モジュール名には、「Output (出力)」を表す「O」の文字が入ります (8OX、OB、OWなど)。

PROFIBUS-DPマスタとPSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションの両方に対して、仮想PROFIBUSモジュールをコンフィグレーションする必要があります。PROFIBUS-DPマスタには、GSDファイルからの潜在的な仮想モジュールに必要なすべての情報が含まれます（「[仮想PROFIBUSモジュールのGSDファイル](#)」[\[222\]](#)を参照）。

PAS4000のPROFIsafe搭載PROFIBUS-DPスレーブエディタでのハードウェアコンフィグレーションの際に、PSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションに必要な仮想PROFIBUSモジュールがコンフィグレーションされます。コンフィグレーション後に、仮想モジュールが仮想スロットに配置されます。

「PROFIBUS-DPスレーブ」システムセクションの以下の機能とプロシージャが記述されています。これらは「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションにも適用されます。

- ▶ 受信データ用仮想PROFIBUSモジュール (入力モジュール)
（「[受信データ用仮想PROFIBUSモジュール](#)」[\[199\]](#)を参照）
- ▶ 送信データ用仮想PROFIBUSモジュール (出力モジュール)
（「[送信データ用仮想PROFIBUSモジュール](#)」[\[197\]](#)を参照）
- ▶ PROFIBUS-DPスレーブI/Oデータ
（「[PROFIBUS-DPスレーブのI/Oデータ](#)」[\[202\]](#)を参照）
- ▶ PROFIBUS-DPアドレスのアドレスフォーマット
（「[PROFIBUS-DPアドレスのアドレス表示](#)」[\[205\]](#)を参照）
- ▶ PROFIBUS診断
（「[PROFIBUSの診断](#)」[\[207\]](#)を参照）
- ▶ PROFIBUS-DPのデータ整合性
（「[PROFIBUS-DPのデータの整合性](#)」[\[212\]](#)を参照）

9.7.3 「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションのPROFIsafe

「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000は、PROFIsafe経由で**安全関連データ**を送受信します。PROFIsafe経由でデータを交換する場合、これらはF装置として動作します。これは、常にFホスト (Fマスタ) となり、データ転送を開始します。オートメーションシステムPSS 4000から見ると、Fホストはシステムの外部にあります。つまり、受信データは外部入力データと呼ばれ、送信データは外部出力データと呼ばれます (「[処理データの有効性](#)」 281」を参照)。

送信データと受信データ

Fホストは、送信データ領域からPSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションの受信データ領域にデータを送信し、PSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションからのデータを要求します。

PSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションは、要求されたデータを送信データ領域からFホストの受信データ領域に送信します。

オートメーションシステムPSS 4000では、これらのデータ領域を「仮想PROFIBUSモジュール」といいます。PROFIsafe経由でのデータ交換のために、さまざまな事前定義済み仮想Fモジュールを使用できます。Fモジュールは、受信データ用および送信データ用の仮想PROFIsafeサブモジュールで構成されています。Fモジュールには全体で24 Byteのデータ長があり、これらの8 Byteには受信データ用仮想PROFIsafeサブモジュールが割り付けられ、別の8 Byteには送信データ用仮想PROFIsafeサブモジュールが割り付けられます。受信データ用仮想PROFIsafeサブモジュールは入力サブモジュールと呼ばれます。これらのサブモジュールでは、「Input (入力)」を表す「I」がモジュール名に付けられます (例: 8I_SAFEBOOL、I_SAFEBYTE、I_SAFEWORD、I_SAFEDWORD、I_SAFEINT)。送信データ用仮想PROFIsafeサブモジュールは出力サブモジュールと呼ばれます。これらのサブモジュールでは、「Output (出力)」を表す「O」がモジュール名に付けられます (例: 8O_SAFEBOOL、O_SAFEBYTE、O_SAFEWORD、O_SAFEDWORD、O_SAFEINT)。

FホストとPSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションの両方に対して、仮想PROFIBUSモジュールをコンフィグレーションする必要があります。Fホストは、GSDファイルから必要な仮想PROFIBUSモジュールを取得します (「[仮想PROFIBUSモジュールのGSDファイル](#)」 222」を参照)。

PAS4000のPROFIsafe搭載PROFIBUS-DPスレーブエディタでのハードウェアコンフィグレーションの際に、PSS 4000の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションに必要なFモジュールがコンフィグレーションされます。コンフィグレーション後、Fモジュールは仮想PROFIsafeサブモジュール用のサブスロットで仮想スロットに配置されます。

9.7.3.1 事前定義済みFモジュール

事前定義済みFモジュールは、PAS4000のPSS 4000に対して使用できます。これらのFモジュールには、仮想PROFIsafeサブモジュール (I/Oサブモジュール) が事前に割り付けられます。

PSS 4000のFモジュールには、PROFIsafe Fホスト内に対応するFモジュールが必要です。Fモジュールの割り付けは次の通りです。

F_32IX_32IX_32OX_32OX

PSS 4000のFモジュールF_32IX_32IX_32OX_32OXには、Fホスト内に対応するF_32OX_32OX_32IX_32IXが必要です。

Fホスト		PSS 4000		
F_32OX_32OX_32IX_32IX		F_32IX_32IX_32OX_32OX (既定の割り付け)		
		仮想サブスロット	I/Oサブモジュール	
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

PSS 4000のFモジュールF_32IX_IWA2_32OX_OWA2には、Fホスト内に対応するF_32OX_OWA2_32IX_IWA2が必要です。

Fホスト		PSS 4000		
F_32OX_OWA2_32IX_IWA2		F_32IX_IWA2_32OX_OWA2 (既定の割り付け)		
		仮想サブスロット	I/Oサブモジュール	
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

PSS 4000のFモジュールF_IWA4_OWA4には、Fホスト内に対応するF_OWA4_IWA4が必要です。

Fホスト		PSS 4000		
F_OWA4_IWA4		F_IWA4_OWA4		
		仮想サブスロット	I/Oサブモジュール	
Output	Integer16	0	I_SAFEINT	Input
	Integer16	1	I_SAFEINT	
	Integer16	2	I_SAFEINT	
	Integer16	3	I_SAFEINT	

Fホスト		PSS 4000		
Input	Integer16	4	O_SAFEINT	Output
	Integer16	5	O_SAFEINT	
	Integer16	6	O_SAFEINT	
	Integer16	7	O_SAFEINT	

9.7.3.2 送信データ用仮想PROFIsafeサブモジュール

FモジュールにSAFEBOOLデータ型の出力サブモジュールが含まれている場合、これらのサブモジュールには、PAS4000内の別のデータ型の出力サブモジュールが割り付けられている可能性があります。これはたとえば、2つのFモジュール、F_32IX_32IX_32OX_32OXとF_32IX_IWA2_32OX_OWA2に該当します。コンフィグレーションに対して、以下の出力サブモジュールを使用できます。

出力サブモジュール	データ型	用途
8O_SAFEBOOL	ARRAY [0..7] OF SAFEBOOL	仮想ビットモジュール SAFEBOOLデータ型の8つのファンクションでの配列送信
O_SAFEBYTE	SAFEBYTE	仮想バイトモジュール SAFEBYTEデータ型のデータ送信
O_SAFEWORD	SAFEWORD	仮想WORDモジュール SAFEWORDデータ型のデータ送信
O_SAFEDWORD	SAFEDWORD	仮想DWORDモジュール SAFEDWORDデータ型のデータ送信
O_SAFEINT	SAFEINT	仮想INTモジュール SAFEINTデータ型のデータ送信



情報

ここに記載されている指定されたデータ型の仮想モジュールのみがサポートされていることに注意してください。「SAFELWORD」データ型はサポートされていません。

9.7.3.3 受信データ用仮想PROFIsafeサブモジュール

FモジュールにSAFEBOOLデータ型の入力サブモジュールが含まれている場合、これらのサブモジュールには、PAS4000内の別のデータ型の入力サブモジュールが割り付けられている可能性があります。これはたとえば、2つのFモジュール、F_32IX_32IX_32OX_32OXとF_32IX_IWA2_32OX_OWA2に該当します。コンフィグレーションに対して、以下の入力サブモジュールを使用できます。

入力サブモジュール	データ型	用途
8I_SAFEBOOL	ARRAY [0..7] OF SAFEBOOL	仮想ビットモジュール BOOLデータ型の8つのファンクションでの 配列受信
I_SAFEBYTE	SAFEBYTE	仮想バイトモジュール SAFEBYTEデータ型のデータ受信
I_SAFEWORD	SAFEWORD	仮想WORDモジュール SAFEWORDデータ型のデータ受信
I_SAFEDWORD	SAFEDWORD	仮想DWORDモジュール SAFEDWORD データ型のデータ受信
I_SAFEINT	SAFEINT	仮想INTモジュール SAFEINTデータ型のデータ受信



情報

ここに記載されている指定されたデータ型の仮想モジュールのみがサポートされていることに注意してください。「SAFELWORD」データ型はサポートされていません。

9.7.4 仮想PROFIBUSモジュールのGSDファイル

「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションで使用可能なすべての仮想PROFIBUSモジュールは、対応するGSDファイルに含まれています。



情報

「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションに属すGSDファイルのみを使用してください。このファイルには、PROFIBUS-DPマスタ用の入力モジュールと出力モジュールの両方、PROFIsafe FホストのFモジュールが含まれます。

GSDファイルはPAS4000で提供され、ピルツホームページのダウンロードエリア(www.pilz.deのダウンロードページ)でも入手できます。たとえば「GSD」を検索語句として入力できます。ソフトウェアのダウンロードは、登録ユーザのみ実行できます。未登録ユーザの方は、無料で登録ができます。

9.7.5 データ転送の制限

「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000では、データ転送は仮想PROFIBUSモジュール経由で実行されます (入力モジュール、出力モジュール、Fモジュール)。装置の「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションには、以下の制限が適用されます。

- ▶ 仮想スロット0～79で最大80の仮想PROFIBUSモジュールをコンフィグレーションできます。
- ▶ Fモジュールは1つのみコンフィグレーションできます。
注意:Fモジュールは、使用可能な仮想PROFIBUSモジュールの合計数を1つずつではなく、5つずつ削減します。
- ▶ Fモジュールには合計24 Byteが必要です。このうち、
 - 入力サブモジュールには正確に8 Byteが予約されています (受信データ用仮想PROFIsafeサブモジュール)。
 - 出力サブモジュールには正確に8 Byteが予約されています (送信データ用仮想PROFIsafeサブモジュール)。
- ▶ Fモジュールでは、仮想PROFIsafeサブモジュールで最大16のサブスロットを使用できます。
- ▶ 最大244 Byteの受信データ (=外部入力データ) をコンフィグレーションできます。
- ▶ 最大244 Byteの送信データ (=外部出力データ) をコンフィグレーションできます。

9.7.6 PROFIsafeのI/Oデータ

「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションにより、PSS 4000とFホストは、PROFIsafeを経由して安全関連のI/Oデータを交換できます。この操作を正常に実行するには、PAS4000のI/OマッピンググエディタのI/OデータをFモジュールの仮想PROFIsafeサブモジュールにマッピングする必要があります。

FモジュールでのI/Oマッピング

I/Oマッピングを使用して、どのデータソースからどのデータシンクにデータを供給するかを定義します。

Fモジュールで使用可能なI/Oマッピング:

I/Oマッピング		Fモジュール	
		受信データ (= 8I_SAFEBOOL、I_SAFEBYTE、 I_SAFEWORD、I_SAFEDWORD、 I_SAFEINTなど、仮想PROFIsafeサブモ ジュールの外部入力データ)	送信データ (= 8O_SAFEBOOL、O_SAFEBYTE、O_SAFEWORD、 O_SAFEDWORD、O_SAFEINTな ど、仮想PROFIsafeサブモジュールの 外部出力データ)
モジュールバス	入力データ (例: ハードウェア入力)	---	◆
	出力データ (例: ハードウェア出力)	◆	----
PI 変数	入力PI変数	◆	---
	出力PI変数	---	◆

◆ I/Oマッピング可能

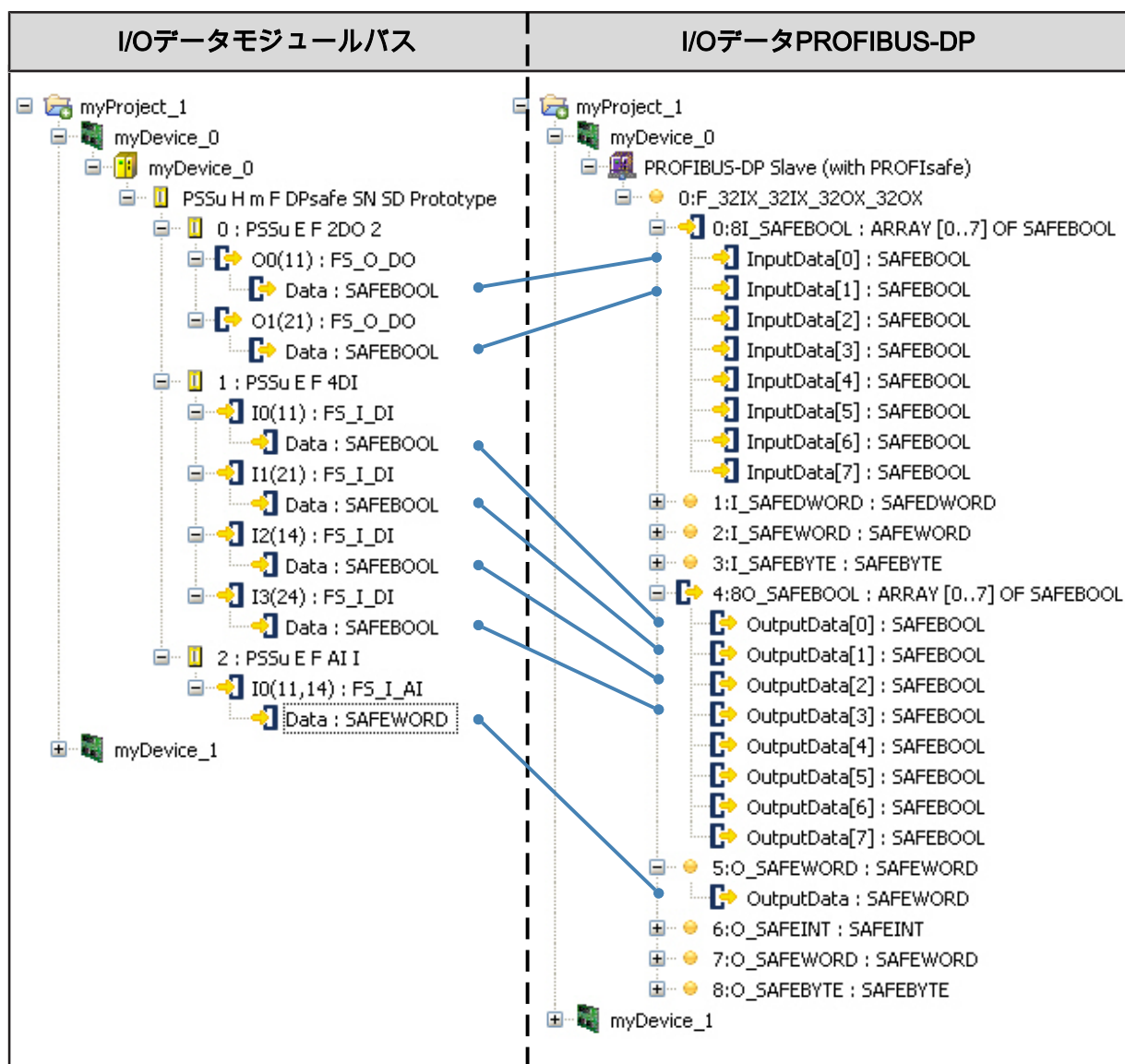
--- I/Oマッピング不可

例

▶ I/OデータのI/Oマッピング

FホストはPSSu E F 2DO 2を有効化して、PSSu E F 4DIおよびPSSu E F AI Iの入力データがFホストに送信されます。この目的のために、必要な仮想I/Oサブモジュールで、FモジュールF_32IX_32IX_32OX_32OXをコンフィグレーションします。

可能なI/Oマッピング (例):

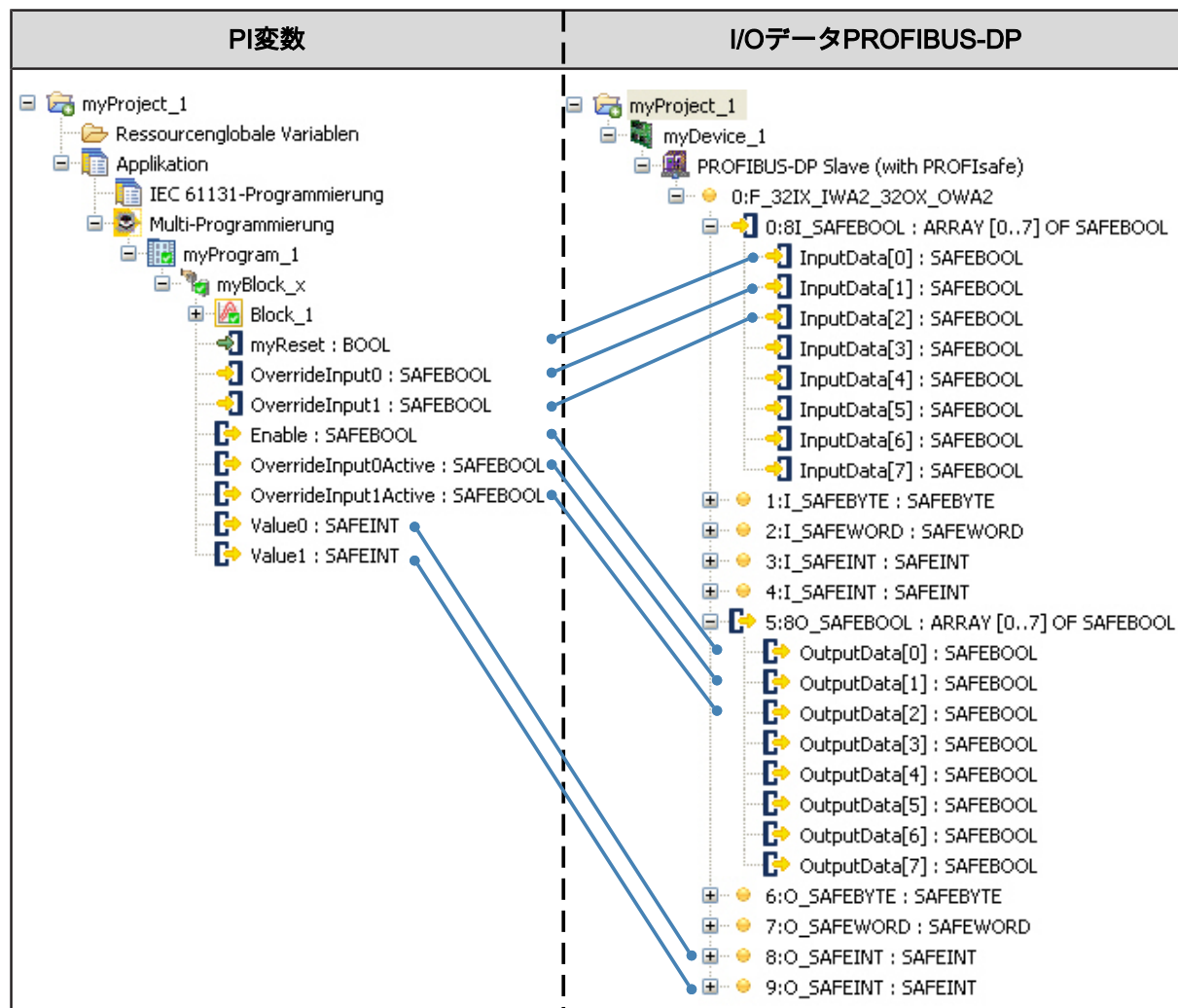


— I/Oマッピング

▶ PI変数のI/Oマッピング

ブロックの出力PI変数の内容はFホストに送信され、Fホストから着信した受信データは入力PI変数に割り付けられます。FモジュールF_32IX_IWA2_32OX_OWA2は、この目的のために使用されます。

可能なI/Oマッピング (例):



— I/Oマッピング

9.7.7 PROFIsafeアドレスのアドレスフォーマット

PSS 4000装置のI/Oデータのアドレスは、PAS4000や診断など、複数の場所に表示されます。

9.7.7.1 送信データのアドレスフォーマット

PROFIsafe送信データのアドレス (仮想PROFIsafeサブモジュールの出力データ) は、次の形式で構造化されます。

<装置名>.ProfibusDP.<仮想スロット>.<仮想サブスロット>.OutputData[<ファンクション番号>]

例:

▶ **8O_SAFEBOOLのアドレスフォーマット:**

```
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]
```

▶ **O_SAFEBYTEのアドレスフォーマット:**

```
myDevice_1.ProfibusDP.1.5.OutputData
```

▶ **O_SAFEWORDのアドレスフォーマット:**

```
myDevice_1.ProfibusDP.1.6.OutputData
```

▶ **O_SAFEDWORDのアドレスフォーマット:**

```
myDevice_1.ProfibusDP.4.3.OutputData
```

▶ **O_SAFEINTのアドレスフォーマット:**

```
myDevice_1.ProfibusDP.4.4.OutputData
```

9.7.7.2 受信データのアドレスフォーマット

PROFIsafe受信データのアドレス (仮想PROFIsafeサブモジュールの入力データ) は、次の形式で構造化されます。

<装置名>.ProfibusDP.<仮想スロット>.<仮想サブスロット>.InputData[<ファンクション番号>]

例:

▶ **8I_SAFEBOOLのアドレスフォーマット:**

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]

▶ **I_SAFEBYTEのアドレスフォーマット:**

myDevice_1.ProfibusDP.1.1.InputData

▶ **I_SAFEWORDのアドレスフォーマット:**

myDevice_1.ProfibusDP.1.2.InputData

▶ **I_SAFEDWORDのアドレスフォーマット:**

myDevice_1.ProfibusDP.4.0.InputData

▶ **I_SAFEINTのアドレスフォーマット:**

myDevice_1.ProfibusDP.4.1.InputData

9.7.8 Fモジュールの無効化と再有効化

Fモジュールに送信される送信データは、Fホスト経由で選択的に安全状態に移行できます。この操作はFモジュールの無効化／再有効化と呼ばれます。

▶ Fモジュールの無効化

- VALIDビットはFALSEになります (「[処理データの有効性](#)  281」を参照)。
- Fモジュールの入力サブモジュールは安全値 (=ゼロ) に設定されます。

▶ Fモジュールの再有効化

- PROFIsafeモジュールエラーがない場合、VALIDビットがTRUEになります。
- Fモジュールの入力サブモジュールには処理値が含まれます。

9.7.9 Fパラメータ

Fパラメータは、それぞれのFデバイスに合わせてPROFIsafeプロトコルを調整するために使用できます。Fパラメータは、GSDファイルの一部です。

ネットワークコンフィグレーションでは、マスタツール (例: Siemens Software HW Config) を使用して、Fパラメータの表示とコンフィグレーションを行い、これらのパラメータをFデバイスにダウンロードできます。Fホストは、Fパラメータを使用してFデバイスとの接続を確認します。さまざまなFパラメータで、FホストのFパラメータの内容がFデバイスの内容と一致しない場合 (「[PROFIsafeの診断](#)」 231]を参照)、通知が表示されます。

Fデバイスとして、PSS 4000装置は以下のFパラメータをサポートします。

Fパラメータ	既定のコンフィグレーション	マスタツールでのコンフィグレーション	意味
F_Check_SeqNr	1	コンフィグレーション不可	整合性チェック (CRC計算) には、シーケンス番号が含まれます。
F_Check_iPar	チェックなし	コンフィグレーション不可	iParameterなし
F_SIL	SIL3	コンフィグレーション不可	FデバイスのSIL
F_CRC_Length	3 Octet	コンフィグレーション不可	CRC2キーの長さ
F_Block_ID	F_iPar_CRCなし	コンフィグレーション不可	パラメータのタイプID
F_Par_Version	V2モード	コンフィグレーション不可	Fデバイス内に実装されるPROFIsafeバージョン
F_Source_Add	1	値の範囲: 1～65534	FホストのFアドレス (マスタツールにより自動的に割り付け)
F_Dest_Add	1	値の範囲: 1～65534	FデバイスのFアドレス
F_WD_Time	200 ms (PROFIsafe搭載 PROFIBUS-DPスレーブの 既定の t_{ExtCo} = 100 ms)	値の範囲: F_WD_Time_min～ 65535	Fデバイスの監視時間。 有効な現在の安全テレグラムは、監視時間 F_WD_Timeの時間内に、Fホストから到着する 必要があります。 監視時間が守られた場合、安全処理データが PROFIsafe経由で他のシステムセクションに配信 されます。 監視時間を超過した場合、PROFIsafe経由のデ ータ転送が停止され、有効ビット = FALSEで代 入値が影響を受ける処理データに使用されます。 F_WD_Timeは、PROFIsafe搭載PROFIBUS- DPスレーブのサイクル時間 (t_{ExtCo}) の少なくとも2 倍に対応する必要があります。 F_WD_Time_minは装置により異なります。

9.7.10 PROFIsafeの診断

PROFIsafeの診断は、PSS 4000のシステム診断を通じて実行できます（「[診断の基本情報](#)  566」を参照）。

ただし、規格ではPROFIBUS-DPの独自の診断オプションが定義されています。PROFIBUS-DPマスタは、診断テレグラムを通じて、PROFIBUS-DPスレーブで現在の診断情報をポーリングできます。

「PROFIBUS-DPスレーブ」とPROFIsafeシステムセクションがあるPSS 4000装置の診断テレグラムには、PROFIBUS診断の先頭の12 Byteの診断データが格納されます。これはそれぞれ6 Byteの2つのブロックに分割されます（「[PROFIBUSの診断](#)  207」を参照）。PROFIsafe診断から診断情報を取得できる場合は、ブロックに5 Byteが追加されます。

診断テレグラムの構造:

ブロック1	Byte 1～6	システム内部の診断データ PROFIBUS-DPスレーブ - システム内部の診断  208を参照	PROFIBUSの診断
ブロック2	Byte 7～12	装置ベースの診断データ PROFIBUS-DPスレーブ - 外部診断  209を参照	
ブロック3	Byte 13～17	F装置の装置ベースの診断	PROFIsafeの診断

診断情報は、GSDファイルの一部です。

9.7.10.1 F装置の装置ベースの診断

F装置またはFモジュールは、PROFIBUS-DPによって定義された診断メカニズムを経由して診断情報を登録できます。F装置として、PSS 4000装置は診断テレグラムの装置ベースの診断データとともに5 Byteを送信します。

F装置上の装置ベースの診断データの構造と内容

バイト番号	ビット番号	値	意味
n	0 ... 7	0x05	ヘッダ
			ビット7とビット6:00 「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000装置:装置ベースの診断のコード
			ビット5 = 0:000101 装置ベースの診断のバイト数 (ヘッダバイトを含む) 「PROFIsafe搭載PROFIBUS-DPスレーブ」システムセクションがあるPSS 4000装置は5 Byteを送信します。
n + 1	0 ... 7	0x81	Status_Type
			ビット7:1 「ステータス」の診断ID
			ビット6 = 0:0000001 「ステータスメッセージ」ステータスタイプのコード
n + 2	0 ... 7	0 ... 254	PROFIsafe Fホストから見たFモジュールスロット (「事前定義済みFモジュール」 [217] を参照)
n + 3	0 ... 7	0x01/0x02	0x01 ステータスメッセージ有/有効
			0x02 ステータスメッセージ無/有効
n + 4	0 ... 7	0x40 ... 0x47	PROFIsafeアラーム ([1] を参照)

[1] PSS 4000装置1台がサポートするPROFIsafeアラーム:

16進	数値	診断情報:
0x40	64	安全宛先アドレスの不一致 (F_Dest_Add) Fホストでパラメータ「F_Dest_Add」に入力された値が、ローカルのPROFIsafeアドレスと一致しません。
0x41	65	安全宛先アドレスが有効ではない (F_Dest_Add) Fホストでパラメータ「F_Dest_Add」に入力された値が無効です。 Fホストの有効なエントリ:1 ... 65534
0x42	66	安全ソースアドレスが有効ではない (F_Source_Add) Fホストでパラメータ「F_Source_Add」に入力された値が無効です。 Fホストの有効なエントリ:1 ... 65534

16進	数値	診断情報:
0x43	67	安全ウォッチドッグ時間の値が0 msである (F_WD_Time) Fホストでパラメータ「F_WD_Time」に入力された値 (0 ms) が無効です。 Fホストの有効なエントリ [ms]:1 ... 65534
0x44	68	パラメータ「F_SIL」が特定の装置のアプリケーションからのSILを超過した Fホストでパラメータ「F_SIL」に入力された値が、PSS 4000の最大SIL (SIL 3) を超えています。 Fホストの有効なエントリ (最大):「SIL3」
0x45	69	パラメータ「F_CRC_Length」が生成された値と一致しない Fホストでパラメータ「F_CRC_Length」に入力された値がサポートされていません。 Fホストの有効なエントリ:「3 octet CRC」
0x46	70	設定されたFパラメータのバージョンが正しくない Fホストでパラメータ「F_Par_Version」に入力された値がサポートされていません。 Fホストの有効なエントリ:「V2モード」
0x47	71	CRC1異常 Fホストと仮想Fモジュールの「CRC1」チェックサムが一致しません。

9.7.11 PROFIsafeのデータの整合性

Fホストが送信する受信データのデータ整合性に対しては、Fホストに責任があります。PSS 4000から見ると、Fホストのデータの整合性に関してステートメントは作成できません。

受信データの整合性

PSS 4000の視点では、Fモジュールの受信データは、外部入力データになります。

外部入力データがFモジュールで使用可能になると、すぐに一貫したデータとして処理されます (「[データの一貫性](#)」 300]を参照)。仮想Fモジュールの外部入力データは、整合性範囲タイプ「外部通信接続のFS入力データ」の整合性範囲を形成します。正確な記述は、整合性範囲「仮想PROFIsafe FモジュールのFS入力データ」です (「[一貫性の範囲](#)」 301]を参照)。

送信データの整合性

PSS 4000の視点では、Fモジュールの送信データは、外部出力データになります。

データソースの同じ整合性範囲から取得した場合 (「[データの一貫性](#)」 300]を参照)、Fモジュールの外部出力データには整合性があります。

PSS 4000では、Fモジュールの外部出力データは、常に1つの単位として処理されます。必要なデータの整合性を保証することは、Fホストの責任です。

9.8 EtherNet/IPアダプタ

EtherNet/IP (Ethernet Industrial Protocol: EIP) は、Ethernetベースのフィールドバスです。EtherNet/IPでは、従来のEthernetが、オートメーションアプリケーション用アプリケーションレイヤとして産業用プロトコル (Common Industrial Protocol: CIP) で拡張された産業用のオープン標準が定義されています。

産業用のオープン標準であるEtherNet/IPは、ControlNet International (CI) とOpen DeviceNet Vendor Association (ODVA) が、Industrial Ethernet Association (IEA) の支援を受けて共同で策定したものです。EtherNet/IPは、IEC 61158シリーズとして国際規格化されています。

Common Industrial Protocol (CIP)

EtherNet/IPネットワークでは、CIPメッセージデータの転送にImplicitメッセージとExplicitメッセージが使用されます。

Implicitメッセージは一般に、I/Oデータを含む小型のデータパケットであり、時間が重視されるデータ転送用です。ImplicitメッセージはUDPを使って転送されます。

時間が重視されないメッセージは、Explicitメッセージで転送されます。Explicitメッセージには、たとえばコンフィグレーションや情報のデータが含まれます。TCP/IP通信方式が使用されます。

CIPはオブジェクト指向です。ODVAでは、装置プロファイルを使用して、特定の装置クラスの装置でサポートする必要があるオブジェクトと属性を定義しています。オプションやメーカー定義のオブジェクトや属性も可能です。CIPでは、コンフィグレーションの共通の基準として電子データシート (EDS) が使用されます。

CIPまたはEtherNet/IPの詳細情報は、ODVA (www.odva.org) が提供しています。

ピルツのEtherNet/IP対応PSS 4000装置

安全関連**以外**のアプリケーションでは、ピルツのEtherNet/IP対応のPSS 4000装置をEtherNet/IPにアダプタとして組み込むことができます。

ピルツのEtherNet/IP対応PSS 4000装置は、次のEtherNet/IPアダプタをサポートしています。

- ▶ ODVAの「Common Industrial Protocol仕様書」Volume 1に準拠する通信アダプタ装置プロファイル、装置タイプ12。
2012年4月公開のEdition 3.12に従って開発と適合性試験を実行済み
- ▶ EtherNet/IPスキャナとの周期的データ交換
 - ImplicitメッセージのI/Oデータを対象としたI/Oサーバ機能
 - I/OデータはUDPを使って送信され、ポート2222を使用します。
- ▶ 「Explicitメッセージサーバ」としての非周期的データ交換
 - Explicitデータを対象としたExplicitメッセージサーバ機能
 - 次のオブジェクトとの非周期的データ交換のサポート: アイデンティティオブジェクト、TCP/IPオブジェクト、およびリンクオブジェクト
- ▶ アイデンティティオブジェクト
- ▶ TCP/IPオブジェクト
- ▶ リンクオブジェクト
- ▶ 次の要素を備えたアセンブリオブジェクト
 - 出力アセンブリオブジェクトインスタンス

– 入力アセンブリオブジェクトインスタンス

9.8.1 「EtherNet/IPアダプタ」システムセクション

オートメーションシステムPSS 4000では、「EtherNet/IPアダプタ」システムセクションがあるすべてのPSS 4000装置と、EtherNet/IPを使った外部通信が可能です。この場合、PSS 4000装置の「EtherNet/IPアダプタ」システムセクションが、通信に必要なタスクをすべて実行します。次のようなものがあります。

- ▶ サポート対象オブジェクトの管理
(「[サポートされているEtherNet/IPオブジェクトと共通のサービス](#)  251」を参照)
- ▶ EtherNet/IPを使ったデータ交換
(「[EtherNet/IPを使ったデータ交換](#)  239」を参照)

9.8.2 EtherNet/IPアダプタの接続

現在、PSS 4000装置あたり1つのEtherNet/IPアダプタ接続だけがサポートされています。このEtherNet/IPアダプタ接続で、1つの出力アセンブリオブジェクトインスタンスおよび1つの入力アセンブリオブジェクトインスタンスとのデータ交換が可能です (「[アセンブリオブジェクト](#)」 254]」を参照)。

9.8.3 EtherNet/IPを使ったデータ交換

「EtherNet/IPアダプタ」システムセクションがあるPSS 4000装置は、EtherNet/IPアダプタとしてデータ交換をサポートします。データ転送は必ずEtherNet/IPスキャナから開始します。オートメーションシステムPSS 4000から見た場合、EtherNet/IPスキャナはシステムの外にあります。

9.8.3.1 送信データと受信データ

EtherNet/IPでは、実際の使用可能なデータ（プロセスデータ）がアセンブリオブジェクト経由で交換されます。「EtherNet/IPアダプタ」システムセクションがあるPSS 4000装置は、出力アセンブリオブジェクトインスタンスと入力アセンブリオブジェクトインスタンスがそれぞれ1つつあるアセンブリオブジェクトをサポートしています。PSS 4000では、EtherNet/IPの用語「入力」と「出力」は、スキャナから見た定義です。この定義のメリットは、スキャンツールとPAS4000の両方で用語が同じであることです。つまり、スキャンツールの出力アセンブリオブジェクトインスタンスがPAS4000の出力アセンブリオブジェクトインスタンスでもあり、スキャンツールの入力アセンブリオブジェクトインスタンスがPAS4000の入力アセンブリオブジェクトインスタンスでもあります。

PSS 4000では、出力アセンブリオブジェクトインスタンスを「スキャナ出力アセンブリオブジェクトインスタンス」と呼び、入力アセンブリオブジェクトインスタンスを「スキャナ入力アセンブリオブジェクトインスタンス」と呼びます。

スキャナ出力アセンブリオブジェクトインスタンス

スキャナ出力アセンブリオブジェクトインスタンスには、スキャナが生成し、ネットワークに配置するデータ、つまりPSS 4000装置（アダプタ）で受信されるデータが含まれます。スキャナ出力アセンブリオブジェクトインスタンスを使用して転送されるデータの長さは、スキャナとPSS 4000装置の両方で同じコンフィグレーションになっている必要があります。PSS 4000装置では、PAS4000のEtherNet/IPアダプタエディタでこのコンフィグレーションを行います。

スキャナ出力アセンブリオブジェクトインスタンスに含まれるデータは、PSS 4000装置から見ると受信データ（＝外部入力データ）であるので、EtherNet/IPアダプタエディタで仮想入力モジュールのコンフィグレーションを行う必要があります。仮想入力モジュールのモジュール名には、「Input（入力）」を表す「I」の文字が入ります（16IX、IB、IW、IDなど）。

PSS 4000装置では、必ずスキャナから実行／アイドルヘッダが送信されることが想定されます。

スキャナ入力アセンブリオブジェクトインスタンス

スキャナ入力アセンブリオブジェクトインスタンスには、スキャナがネットワークから使用するデータ、つまりPSS 4000（アダプタ）から送信されるデータが含まれます。スキャナ入力アセンブリオブジェクトインスタンスを使用して転送されるデータの長さは、スキャナとPSS 4000の両方で同じコンフィグレーションになっている必要があります。PSS 4000では、PAS4000のEtherNet/IPアダプタエディタでこのコンフィグレーションを行います。

スキャナ入力アセンブリオブジェクトインスタンスに含まれるデータは、PSS 4000から見ると送信データ（＝外部出力データ）であるので、EtherNet/IPアダプタエディタで仮想出力モジュールのコンフィグレーションを行う必要があります。仮想出力モジュールのモジュール名には、「Output（出力）」を表す「O」の文字が入ります（16OX、OB、OW、ODなど）。

PSS 4000のステータス情報をスキャナに送信するためのステータスワードをオプションとしてコンフィグレーションできます。

9.8.3.2 送信データ用仮想モジュール

PSS 4000 (アダプタ) から見ると、スキャナ入力アセンブリオブジェクトインスタンスには送信データ (= 外部出力データ) が含まれます。PAS4000で、PSS 4000の送信データに次の仮想モジュールをコンフィグレーションできます。

仮想モジュール	データ型	アプリケーション
16OX	ARRAY [0..15] OF BOOL	仮想EtherNet/IPビットモジュール データ型がBOOLの要素を16個持つ配列を送信
OBAn	ARRAY [0...(n-1)] OF BYTE	仮想EtherNet/IP Byteモジュール データ型がBYTEの要素をn個持つ配列を送信 n = 2、4、8、16、32、64のいずれか
OW	WORD	仮想EtherNet/IP Byteモジュール データ型WORDのデータ送信
OWAn	ARRAY [0...(n-1)] OF WORD	仮想EtherNet/IP Byteモジュール データ型がWORDの要素をn個持つ配列を送信 n = 2、4、8、16、32、64のいずれか
OD	DWORD	仮想EtherNet/IP Byteモジュール データ型DWORDのデータ送信



情報

ここでは、指定されているデータ型で記載されている仮想モジュールのみがサポートされていることに注意してください。LWORDデータ型はサポートされていません。

スキャナ入力アセンブリオブジェクトインスタンスを使用して転送されるデータの長さは、スキャナとPSS 4000の両方で同じコンフィグレーションになっている必要があります。

9.8.3.3 受信データ用仮想モジュール

PSS 4000 (アダプタ) から見ると、スキャナ出力アセンブリオブジェクトインスタンスには受信データ (= 外部入力データ) が含まれます。PAS4000で、PSS 4000の受信データに次の仮想モジュールをコンフィグレーションできます。

仮想モジュール	データ型	アプリケーション
16IX	ARRAY [0..15] OF BOOL	仮想EtherNet/IPビットモジュール データ型がBOOLの要素を16個持つ配列を受信
IBAn	ARRAY [0...(n-1)] OF BYTE	仮想EtherNet/IP Byteモジュール データ型がBYTEの要素をn個持つ配列を受信 n = 2、4、8、16、32、64のいずれか
IW	WORD	仮想EtherNet/IP Byteモジュール データ型WORDのデータ受信
IWAn	ARRAY [0...(n-1)] OF WORD	仮想EtherNet/IP Byteモジュール データ型がWORDの要素をn個持つ配列を受信 n = 2、4、8、16、32、64のいずれか
ID	DWORD	仮想EtherNet/IP Byteモジュール データ型DWORDのデータ受信



情報

ここでは、指定されているデータ型で記載されている仮想モジュールのみがサポートされていることに注意してください。LWORDデータ型はサポートされていません。

スキャナ出力アセンブリオブジェクトインスタンスを使用して転送されるデータの長さは、スキャナとPSS 4000の両方で同じコンフィグレーションになっている必要があります。

9.8.4 データ転送の制限

EtherNet/IPでデータを送信する場合、PSS 4000 (アダプタ) には次の制限が適用されます。

- ▶ PSS 4000 1台につき、スキャナへの接続 (Ethernet/IPアダプタ接続) を最大1接続コンフィグレーションで
きます。
- ▶ PSS 4000は、最大でスキャナ出力アセンブリオブジェクトインスタンスを1つとスキャナ入力アセンブリオブ
ジェクトインスタンスを1つサポートしています。
- ▶ スキャナ出力アセンブリオブジェクトインスタンスには、総データ長が0～504 Byteの仮想入力モジュール
(外部入力データ) をコンフィグレーションできます。
スキャナ出力アセンブリオブジェクトインスタンスの総データ長は最大510 Byteです (仮想入力モジュール
に504 Byte、実行／アイドルヘッダに4 Byte、シーケンスカウントに2 Byte)。
- ▶ スキャナ入力アセンブリオブジェクトインスタンスには、総データ長が0～508 Byteの仮想出力モジュール
(外部出力データ) をコンフィグレーションできます。
スキャナ入力アセンブリオブジェクトインスタンスの総データ長は最大510 Byteです (仮想出力モジュール
に508 Byte、シーケンスカウントに2 Byte)。

9.8.5 EtherNet/IPアダプタのI/Oデータ

「EtherNet/IPアダプタ」システムセクションによって、PSS 4000装置とEtherNet/IPスキャナの間でEtherNet/IPを使ったI/Oデータの交換が可能です。そのためには、PAS4000のI/Oマッピングエディタで、I/Oデータを、対応する仮想モジュールにマッピングする必要があります。

仮想モジュールのI/Oマッピング

I/Oマッピングを使用して、どのデータソースからどのデータシンクにデータを供給するかを定義します。

仮想モジュールに可能なI/Oマッピング:

I/Oマッピング		仮想モジュール	
		スキャナ出力アセンブリオブジェクトインスタンスのデータ (= 16IX、IBAn、IWAn、IDなど、入力モジュールの外部入力データ)	スキャナ入力アセンブリオブジェクトインスタンスのデータ (= 16OX、OBAn、OWAn、ODなど、出力モジュールの外部出力データ)
モジュールバス	入力データ (例: ハードウェア入力)	---	◆
	出力データ (例: ハードウェア出力)	◆	---
PI 変数	入力PI変数	◆	---
	出力PI変数	---	◆

◆ I/Oマッピング可能

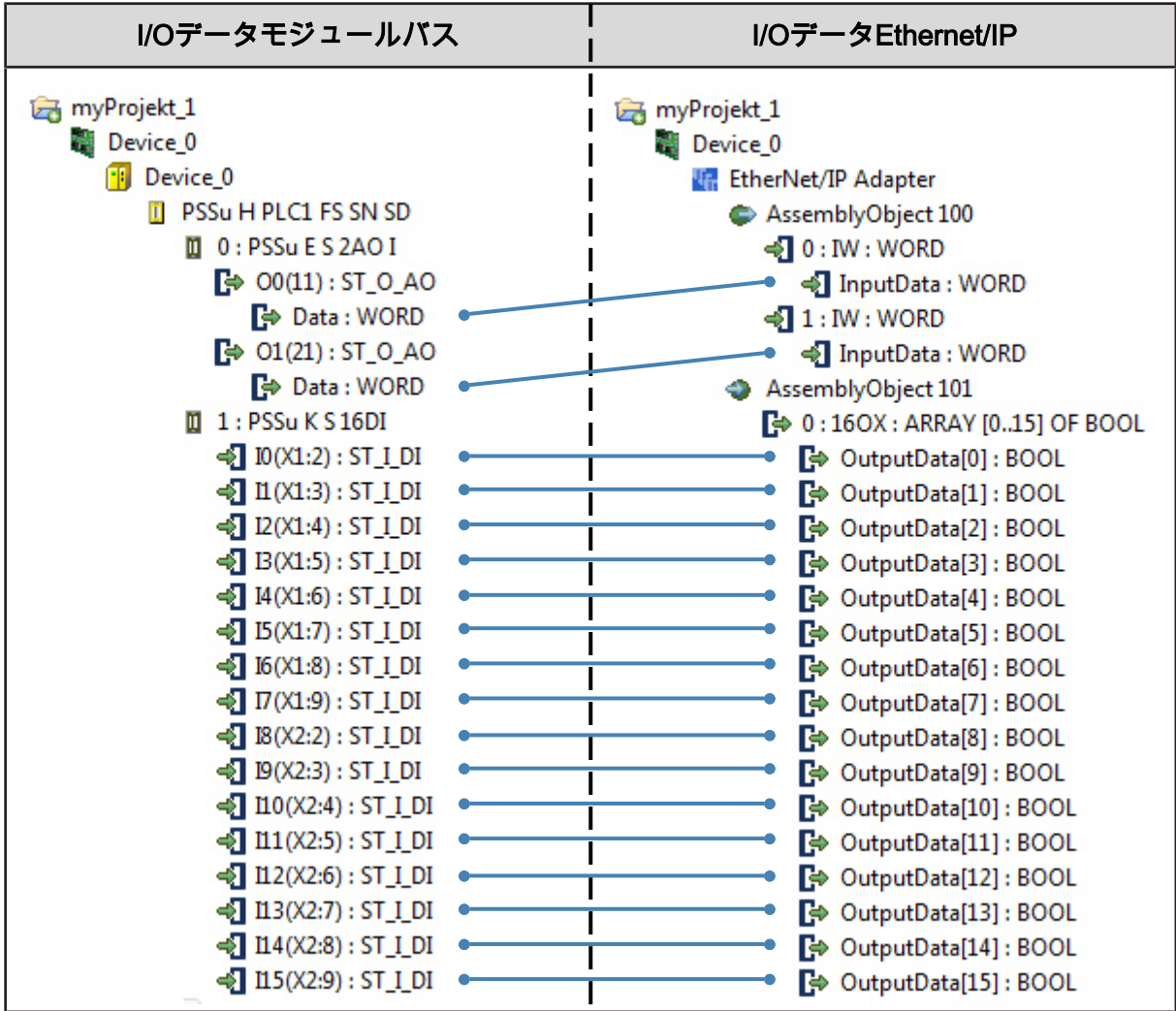
--- I/Oマッピング不可

例

▶ I/OデータのI/Oマッピング

PSSu K S 16DIのハードウェア入力の入力信号をスキャナに送信し、スキャナがPSSu E S 2 AO Iのハードウェア出力を制御します。

可能なI/Oマッピング (例):

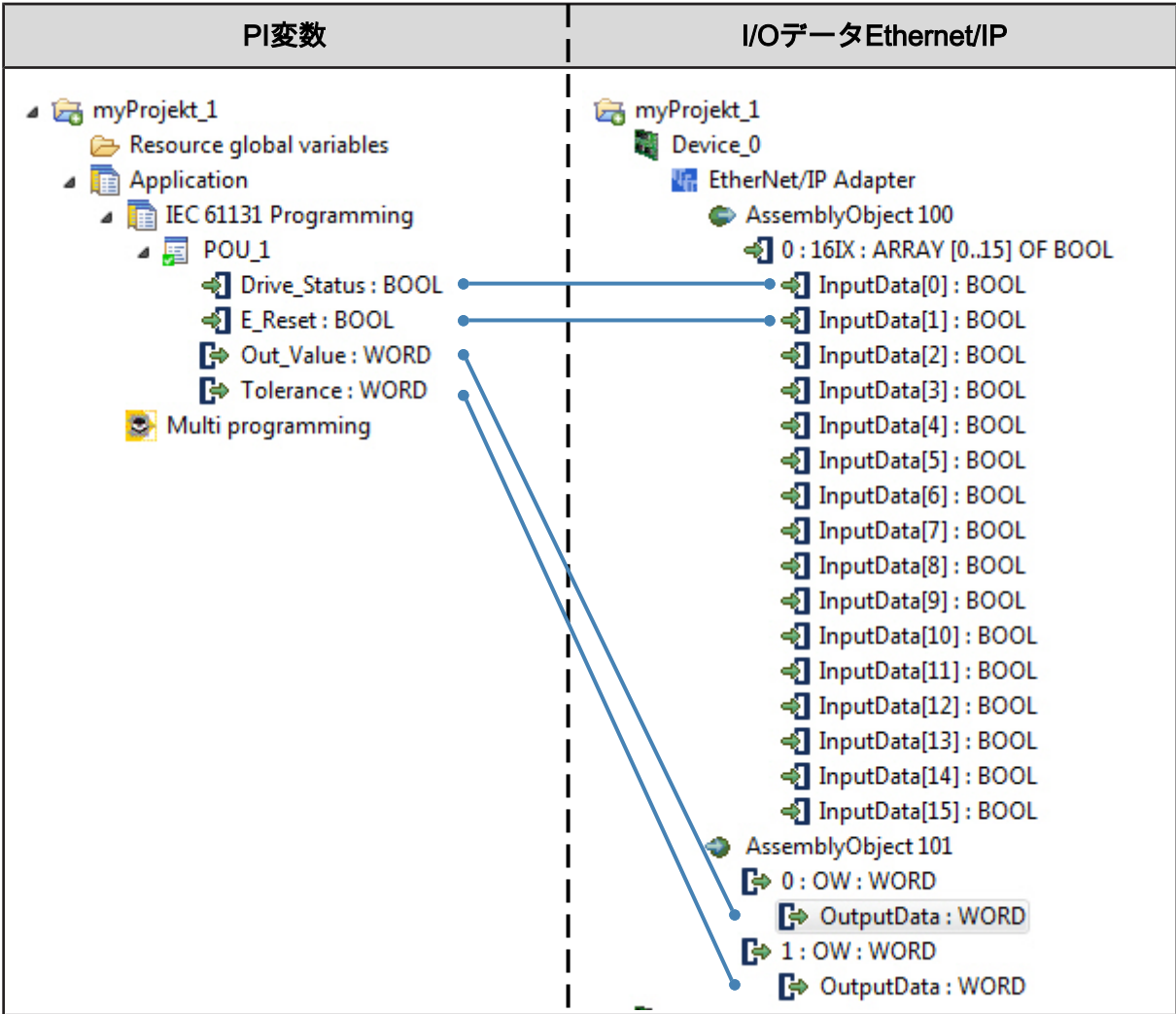


● I/Oマッピング

▶ PI変数のI/Oマッピング

ブロックの出力PI変数の内容をスキャナに送信し、入力PI変数を受信データに割り付けます。

可能なI/Oマッピング (例):



— I/Oマッピング

9.8.6 EtherNet/IPアドレスのアドレスフォーマット

PSS 4000装置のI/Oデータのアドレスは、PAS4000や診断など、複数の場所に表示されます。

9.8.6.1 送信データのアドレスフォーマット

EtherNet/IPの送信データ (仮想モジュールの出力データ) のアドレスは次のフォーマットになっています。

<装置名>.EtherNetIP.AObj<インスタンス番号>.<仮想スロット>.OutputData[<ファンクション番号>]

例:

▶ 16OXのアドレスフォーマット:

```
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]
```

▶ OBA2のアドレスフォーマット:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule4.OutputData[0]
myDevice_1.EtherNetIP.AObj101.VirtualModule4.OutputData[1]
```

▶ OWのアドレスフォーマット:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule5.OutputData
```

▶ OWA4のアドレスフォーマット:

```
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]
```

▶ ODのアドレスフォーマット:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule7.OutputData
```

9.8.6.2 受信データのアドレスフォーマット

EtherNet/IPの受信データ (仮想モジュールの入力データ) のアドレスは次のフォーマットになっています。

<装置名>.EtherNetIP.AObj<インスタンス番号>.<仮想スロット>.InputData[<ファンクション番号>]

例:

▶ 16IXのアドレスフォーマット:

```
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]
```

▶ IBA2のアドレスフォーマット:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule9.InputData[0]
myDevice_1.EtherNetIP.AObj100.VirtualModule9.InputData[1]
```

▶ IWのアドレスフォーマット:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule10.InputData
```

▶ IWA4のアドレスフォーマット:

```
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]
```

▶ IDのアドレスフォーマット:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule12.InputData
```

9.8.7 EtherNet/IPの診断

オプションとして、PAS4000でスキャナ入力アセンブリオブジェクトインスタンスのステータスワードの送信をコンフィグレーションできます。ステータスワードは、装置ベースのステータス情報を含み、診断のためにスキャナに送信されます。

ステータスワードの構造と内容

Octet 1には、EtherNet/IPアダプタシステムセクションがあるPSS 4000装置のエラーメッセージが含まれます。

ビット	値	意味
0	0/1	装置に関するエラーメッセージ
		0 メッセージなし
		1 ▶ 装置の1つ以上のシステムセクションに重大度が「エラー」レベルのメッセージがある または ▶ 装置の1つ以上のFSシステムセクションにFSの重大エラーがある PSS 4000専用装置では、次の [DIAG] ステータスLEDに対応しています。ステータス: 赤の点灯または点滅
1	0/1	装置のFSまたはSTモジュールバスシステムセクションのエラー
		0 メッセージなし
		1 ▶ オペレーティングステータス「装置のすべてのFS出力が安全状態」または ▶ 少なくとも1つのモジュールにアクセスできない (オペレーションの最中にモジュールが削除された、設定されたハードウェアレジストリが実際のものと一致しないなど)、または ▶ オペレーティングステータス「FSモジュールバスがエラーでSTOP状態: FSの重大エラー」 PSS 4000専用装置では、次の [MBUS] ステータスLEDに対応しています。ステータス: 赤の点灯または点滅
2	0/1	装置の「FS SafetyNET p」システムセクションのエラー
		0 メッセージなし
		1 ▶ オペレーティングステータス「FS SafetyNET pがエラーでSTOP状態: Major FS error」 または ▶ オペレーティングステータス「FS SafetyNET pがエラーでSTOP状態: Major FS+ST error」 PSS 4000専用装置では、次の [FS SNp] ステータスLEDに対応しています。ステータス: 赤の点灯または点滅

ビット	値	意味
3	0/1	装置の「FS SafetyNET p」システムセクションのエラー
		0 メッセージなし
		1 オペレーティングステータス「ST SafetyNET pがエラーでSTOP状態: Major FS+ST error」 PSS 4000専用装置では、次の [ST SNp] ステータスLEDに対応しています。ステータス: 赤の点灯または点滅
4	0	予約済み
5	0	予約済み
6	0	予約済み
7	0	予約済み

オクテット2には、EtherNet/IPアダプタシステムセクションがあるPSS 4000装置の追加ステータス情報が含まれます。

ビット	値	意味
0	0/1	装置に関する警告メッセージ
		0 メッセージなし
		1 装置に重大度が「警告」レベルのメッセージがある PSS 4000専用装置では、次の [DIAG] ステータスLEDに対応しています。ステータス: オレンジで点灯
1	0/1	FSリソースに対する強制出力
		0 メッセージなし
		1 FSリソースに対する強制出力は有効 PSS 4000専用装置では、次の [FS FORCE] ステータスLEDに対応しています。ステータス: 黄色で点灯
2	0/1	STリソースに対する強制出力
		0 メッセージなし
		1 STリソースに対する強制出力は有効 PSS 4000専用装置では、次の [ST FORCE] ステータスLEDに対応しています。ステータス: 黄色で点灯
4	0	予約済み
5	0	予約済み
6	0	予約済み
7	0	予約済み

9.8.8 EtherNet/IPのデータ整合性

EtherNet/IP接続経由でスキャナから送信される受信データ (アセンブリオブジェクトのスキャナ出力インスタンス) のデータの整合性はスキャナによって維持されます。PSS 4000から、スキャナのデータの整合性に関して通知を行うことはできません。

スキャナ出力アセンブリオブジェクトインスタンスによる受信データのデータ整合性

PSS 4000専用／対応装置が、EtherNet/IP接続からデータを受信する場合、装置から見て、受信データは外部入力データです。

EtherNet/IP接続からの外部入力データは、PSS 4000で整合性のあるデータとして処理されます。

EtherNet/IP接続からの外部入力データは、整合性範囲タイプ「外部通信接続のST-Iデータ」の整合性範囲を形成します。正確には、整合性範囲「EtherNet/IPアセンブリオブジェクトインスタンスのST-Iデータ」です。

スキャナ入力アセンブリオブジェクトインスタンスによる送信データのデータ整合性

EtherNet/IP接続のPSS 4000専用／対応装置が、データを送信する場合、装置から見て、送信データは外部出力データです。

EtherNet/IP接続の外部出力データは、整合性範囲がデータソースと同じである場合、整合性があると見なされます。

9.8.9 サポートされているEtherNet/IPオブジェクトと共通のサービス

EtherNet/IPを使用したデータ交換では、EtherNet/IPのサブスクライバが、CIPオブジェクトのコレクションで構成されます。CIPオブジェクトは、クラス、インスタンス、および属性に分類されます。

「EtherNet/IPアダプタ」システムセクションがあるPSS 4000装置は、アダプタとして次のオブジェクトクラスをサポートします。

クラスID	オブジェクトクラスの名前
0x01	アイデンティティオブジェクト
0x02	メッセージルータオブジェクト
0x04	アセンブリオブジェクト
0x06	コネクションマネージャオブジェクト
0xF4	ポートオブジェクト
0xF5	TCP/IPインタフェースオブジェクト
0xF6	(ネットワーク固有) リンクオブジェクト

適切な診断ツール (Rockwellのツールなど) を使用して、属性の内容を表示できます。さまざまな[共通のサービス](#)  [258](#)がサポートされています。

9.8.9.1 アイデンティティオブジェクト (クラスID 0x01)

アイデンティティオブジェクトは、「EtherNet/IPアダプタ」システムセクションがある各PSS 4000にあります。
装置の識別情報と、装置に関するその他の一般的な情報が含まれます。

アイデンティティオブジェクトインスタンスのインスタンス属性

属性ID	名前	値	説明
1	ベンダID	181 (0x00B5)	ピルツのメーカID
2	装置のタイプ	12 (0x000C)	製品タイプに関する一般的な情報 (12 = 通信アダプタ)
3	製品コード	既定値: 1 (0x0001)	製品のEtherNet/IP製品コード (例: PSSu H PLC1 FS SN SDは2070)
4	バージョン	- - -	製品のPSS 4000ファームウェアのバージョン <メジャーバージョン番号>.<マイナーバージョン番号> (例: 1.8) バージョン番号の3桁目は表示されない
	メジャーバージョン	<メジャーバージョン番号>	
	マイナーバージョン	<マイナーバージョン番号>	

属性ID	名前	値		説明
5	ステータス	ビット	コンテンツ	製品の現在のステータス
		0	0	未使用
		1	0	予約済み
		2	1	EtherNet/IPがコンフィグレーション済み
		3	0	予約済み
		4～7	0	未使用
			1	
			2	1つ以上のEtherNet/IP接続に問題がある
			3	確立されているEtherNet/IP接続なし
			4	未使用
			5	1つ以上のFSシステムセクションに重大なFSエラーがある (診断リストを参照) または 1つ以上のシステムセクションに重篤度が「エラー」レベルのメッセージがある (診断リストを参照)
			6	1つ以上のEtherNet/IP接続がRUN状態
			7	1つ以上のEtherNet/IP接続が確立済み、すべてのEtherNet/IP接続がアイドル状態
			8、9	予約済み
			10～15	未使用
		8	0	PSS 4000に重篤度が「警告」レベルのメッセージなし
			1	装置に重篤度が「警告」レベルのメッセージあり (診断リストを参照)
		9	0	未使用
		10	0	いずれのFSシステムセクションにも重大なFSエラーはなく、 いずれのシステムセクションにも重篤度が「エラー」レベルのメッセージなし
			1	1つ以上のFSシステムセクションに重大なFSエラーがある (診断リストを参照) または 1つ以上のシステムセクションに重篤度が「エラー」レベルのメッセージがある (診断リストを参照)
		11	0	未使用
		12～15	0	
6	シリアル番号	<シリアル番号>		製品のシリアル番号 (例: PSSuシステムのヘッドモジュールのシリアル番号)
7	製品名	既定値: PSS 4000		製品のEtherNet/IP製品名 (例: PSSu H PLC1 FS SN SD)

9.8.9.2 アセンブリオブジェクト (クラスID 0x04)

スキャナとアダプタの間のデータ交換は、アセンブリオブジェクトのスキャナ出力アセンブリオブジェクトインスタンスとスキャナ入力アセンブリオブジェクトインスタンスを使って行われます。

スキャナ出力アセンブリオブジェクトインスタンスの属性

属性ID	名前	値	説明
1	属性のサポートなし		
2			
3	データ	<データ>	この属性には、コンフィグレーションに従ってスキャナが生成し、ネットワークに送信するデータが含まれる。PSS 4000から見ると、これは外部入力データである。
4	サイズ	<データ長>	コンフィグレーションに従ったデータ長

スキャナ入力アセンブリオブジェクトインスタンスの属性

属性ID	名前	値	説明
1	属性のサポートなし		
2			
3	データ	<データ>	この属性には、コンフィグレーションに従ってスキャナがネットワークから取り込むデータが含まれる。PSS 4000から見ると、これは外部出力データである。
4	サイズ	<データ長>	コンフィグレーションに従ったデータ長

9.8.9.3 TCP/IPインタフェースオブジェクト (クラスID 0xF5)

TCP/IPインタフェースオブジェクトは、装置のTCP/IPプロパティのスキャンに使用できます (IPアドレス、サブネットマスク、ゲートウェイアドレスなど)。EtherNet/IPアダプタシステムセクションがあるPSS 4000装置では、[命名データ \[309\]](#)に含まれるすべてのTCP/IPプロパティがサポートされています。「EtherNet/IPアダプタ」システムセクションがあるPSS 4000のTCP/IPインタフェースオブジェクトは、装置の命名時に命名データからこのコンフィグレーションを取得します。

TCP/IPインタフェースオブジェクトの属性

属性ID	名前	値	説明
1	ステータス	0	EtherNet/IPを使用したデータ交換のコンフィグレーションなし
		1	EtherNet/IPを使用したデータ交換のコンフィグレーションあり
2	コンフィグレーション機能	32 (0x00000020)	装置の命名を通じたTCP/IPプロパティのコンフィグレーション
3	コンフィグレーション制御	0	装置の命名を通じたTCP/IPプロパティのコンフィグレーションは固定
4	物理リンクオブジェクト	<物理リンクオブジェクトへのパス>	装置には、物理リンクオブジェクトのインスタンスが1つある。この属性には、この物理リンクオブジェクトのインスタンスへのパスが含まれる。
		0	装置には、物理リンクオブジェクトのインスタンスが複数ある。
5	インタフェースのコンフィグレーション	<IPアドレス、サブネットマスク、ゲートウェイアドレスなど>	命名データに従ったTCP/IPプロパティ この属性は「読み取り専用」であるため、設定サービスは使用できない。
6	ホスト名	<装置名>	装置の命名の装置名 この属性は「読み取り専用」であるため、設定サービスは使用できない。
		0	ホストのコンフィグレーションがなく、ホストをコンフィグレーションできない
7	属性のサポートなし		
8	TTL値	1	IPマルチキャストを使用したEtherNet/IPパケットのTime To Live (TTL)

属性ID	名前	値	説明
9	Mcast Config	- - -	EtherNet/IPのマルチキャストパケットに使用するIPマルチキャストアドレスのコンフィグレーション 属性の総データ長: 8 bytes
	Alloc control (1 Byte)	0	既定の割り付けアルゴリズムを使用してIPマルチキャストのアドレスを割り付け
		1	「Num Mcast」と「Mcast Start Addr」を使用してIPマルチキャストアドレスを割り付け
		2	予約済み
	予約 (1 Byte)		
	Num Mcast (2 Byte)	<装置によって異なる>	割り付けるIPマルチキャストアドレス数 開始アドレスについては「Mcast Start Addr」を参照
	Mcast Start Addr (4 Byte)	<装置によって異なる>	「Num Mcast」で定義されているIPマルチキャストアドレス数の開始アドレス

9.8.9.4 リンクオブジェクト (クラスID 0xF6)

リンクオブジェクトには、割り付けられている物理インタフェースに関する情報が含まれます。

リンクオブジェクトの属性

属性ID	名前	値		説明
1	インタフェース速度	0		接続不良または 接続なし
		<通信速度>		現在の通信速度 (10、100、1000、10000 MBit/s)
2	インタフェースステータス フラグ	ビット	内容	- - -
		0	0	Ethernet接続なし
			1	Ethernet接続あり
		1	0	半二重モード
			1	全二重モード
		2	0	オートネゴシエーション実行中
			1	オートネゴシエーション失敗
		4	2	オートネゴシエーションは失敗したが、通信速度は検出
			3	オートネゴシエーション成功
			4	オートネゴシエーションの実行なし
		5	0	サポートなし
		6	0	ハードウェアエラー不検出
			1	ハードウェアエラー検出
3	物理アドレス	<MACアドレス>		装置のMACアドレス

属性ID	名前	値	説明
4、5、6	属性のサポートなし		
7	インタフェースタイプ	0	不明
		1	内部 (サポートなし)
		2	ツイストペア
		3	光ファイバケーブル
8	インタフェース状態	0	不明
		1	インタフェースは有効で、データ送受信可能
		2	無効
		3	テストモード
9	属性のサポートなし		
10	インタフェースラベル	<インタフェース名>	インタフェース名のテキスト (例: 「Port X3」)

9.8.9.5 共通のサービス

クラスとインスタンスの属性に対して、次の共通のサービスがサポートされています。

▶ **取得サービス**

診断のために、取得サービス**Get_Attribute_All** (サービスコード0x01) と**Get_Attribute_Single** (サービスコード0x0E) で属性の内容をポーリングできます。Rockwellなどサードパーティのツールが、これらのサービスをサポートしています。サポートされていない属性がアクセスされた場合、「属性のサポートなし」のエラーコード (0x14) が発行されます。

▶ **設定サービス**

設定サービスは、次の属性に使用できます。

- スキャナ出力アセンブリオブジェクトインスタンス - 属性3
周期的通信に含まれない場合、属性3に設定サービスを使用できます。
- コネクションマネージャオブジェクト - 属性1～8

▶ **リセットサービス**

サービスコード0x00または0x01を使用してリセットサービスがEtherNet/IPスキャナから呼び出されると、該当するPSS 4000のすべてのEtherNet/IPアダプタ接続がリセットされます。診断ログと診断リストにエントリが追加されます。

9.9 PROFINET IO DEVICE

PROFINETは、産業用イーサネットに基づいたオープンフィールドバス規格です。PROFINET通信は、国際規格IEC 61158、IEC 61158-5-10 (アプリケーション層サービス定義)、IEC 61158-6-10 (アプリケーション層プロトコル仕様)、IEC 61784-1 Type 10 (通信プロファイル)、IEC 61784-2 (PROFINET IO) で定義されています。詳細な規定は、ユーザ組織PROFIBUS PROFINETインターナショナル (PI) が公開した仕様で定義されています。

ピルツのPROFINET対応PSS 4000装置

非安全関連アプリケーションの場合、PROFINET対応のPSS 4000装置は、IO装置としてPROFINETに組み込むことができます。PROFINET対応PSS 4000装置は、適合性クラスA (CC-A) のIO装置です。PSS 4000装置は、対応するGSDMLファイル (XMLベースのGeneral Station Descriptionファイル) によってPROFINETプロジェクトに組み込まれます。

PROFINET IOシステムは、仕様に従って最低1つのIOコントローラを構成し、最低1つのIO装置を制御します。PSS 4000装置は1つのPROFINET IOコントローラのみに組み込み可能です。つまり複数アクセス (共有装置) はサポートされていません。

PROFINET IOデバイスとして、PSS 4000装置は以下をサポートします。

- ▶ 周期的プロセスデータ交換用IO Data Communication Relations (IO Data CR)
- ▶ リアルタイムのアラーム送信用Alarm Communication Relations (Alarm CR)
- ▶ PROFINET IO診断テレグラム
- ▶ 事前定義済み仮想PROFINETモジュールを経由したデータ交換 (入力モジュール、出力モジュール)

9.9.1 「PROFINET IO DEVICE」システムセクション

オートメーションシステムPSS 4000の場合、「PROFINET IO DEVICE」システムセクションがあるすべてのPSS 4000装置で、PROFINET経由の外部通信を使用できます。この場合、PSS 4000装置の「PROFINET IO DEVICE」システムセクションが、通信に必要なタスクをすべて実行します。値には次のようなものがあります。

- ▶ 送受信データ領域の管理
- ▶ PROFINETを経由したデータ交換

9.9.2 PROFINETでの装置の組み込みとアドレス指定

PSS 4000装置は、PROFINETのエンジニアリングツール (STEP 7 – HW Configなど) によってPROFINETプロジェクトに組み込まれます。

設備のコンフィグレーション中、IO装置は、MACアドレス、IPアドレス、および装置ID (PROFINET装置名) を使用してエンジニアリングツールでアドレス指定されます。アドレス解決は、固有のMACアドレスが割り付けられている装置IDに基づきます。

設備のコンフィグレーション後、エンジニアリングツールはデータ交換に必要なすべての情報をIOコントローラに読み込みます。これには、接続されたIO装置のIPアドレスが含まれます。IOコントローラは、コンフィグレーションされたIO装置を装置IDによって認識し、定義されたIPアドレスをそれらに割り付けます。



重要

無効化されたPROFINET IO装置は、PROFINET IOコントローラとの通信を妨げます

PROFINET IOコントローラとPSS 4000装置の間の通信は、PSS 4000装置のPROFINET IO DEVICEシステムセクションが有効化されている場合にのみ可能です。PROFINET IO DEVICEシステムセクションは、PSS 4000装置の少なくとも1つの仮想PROFINETモジュールをコンフィグレーションすることで、有効化されます。

PSS 4000装置で少なくとも1つの仮想PROFINETモジュールがコンフィグレーションされていることを確認してください。コンフィグレーションされていない場合、PROFINET IOコントローラはPSS 4000装置を検出できません。

PROFINET IO DEVICEシステムセクションを持つPSS 4000装置については、以下の点に注意してください。

▶ PROFINETの装置名

- PSS 4000装置がPROFINETのエンジニアリングツールによって組込まれた場合、PROFINETの要件に従って装置名は表示されません (既定の設定)。
- PSS 4000装置のPROFINET装置名はPROFINETのエンジニアリングツールによって割り付けられ、PROFINET内でのみ意味を持ちます。つまりPROFINET装置名は、PAS4000での装置命名中にPSS 4000装置に対してコンフィグレーションされた装置名には影響しません。
- PSS 4000装置のPROFINET装置名は、装置交換にあたり、PSS 4000装置にSDカードがあり、置き換えられた装置でSDカードを再利用する場合に保持されます。

▶ PROFINET IPコンフィグレーション

- 各PSS 4000装置には、PROFINETでの個別のIPコンフィグレーションがあります。PROFINET IPコンフィグレーションは、PROFINET用のエンジニアリングツールまたはDHCPサーバを介してのみ実行できます。
デフォルト設定: IPアドレス、サブネットマスク、および (デフォルト) ゲートウェイは、0.0.0.0です。

- PROFINET-IPコンフィグレーションはPROFINETでのみ有効です。つまり、PROFINET-IPコンフィグレーションは、PAS4000での装置命名時にPSS 4000装置にコンフィグレーションしたIPコンフィグレーションに影響を与えません。
- PROFINET-IPコンフィグレーションは、PAS4000での装置命名時にPSS 4000装置にコンフィグレーションしたIPコンフィグレーションと同じにすることができます。ただし、2つのIPコンフィグレーションが異なっている場合もありません。
- PROFINET IPコンフィグレーションは、装置交換にあたり、PSS 4000装置にSDカードがあり、置き換えられた装置でSDカードを再利用する場合に保持されます。

▶ MACアドレス

Link Layer Discovery Protocol (LLDP) による「ネイバーディスカバリ (近隣探索)」の場合、PROFINETは装置のMACアドレスを必要としませんが、PROFINETへの装置の接続に使用されている物理イーサネットポートのMACアドレスは必要とします。

- PSS 4000装置には、2つの物理イーサネットポートを備えた統合されたスイッチがあります。両方のポートは2つのMACアドレスを使用し、それらは装置のMACアドレスの後ろに続きます。つまり、PSS 4000装置には3つの連続したMACアドレスがあります。
- 装置のMACアドレスはPSS 4000装置上にあります。

9.9.3 PROFINET内のPSS 4000装置の識別

PROFINET内のPROFINET IO装置を識別することは可能です。識別するために、対象のPROFINET IO装置宛にPROFINETのエンジニアリングツール (STEP 7 – HW Configなど) から照会が送信されるので、装置はこの照会に応答する必要があります。このプロセスは「装置識別」とも呼ばれます。

PSS 4000装置に対して (PROFINET) 装置識別が実行中の場合、SDカードのステータスLEDはオンになります。装置の [SD CARD] ステータスLEDはオレンジ色に点滅します。



重要

無効化されたPROFINET IO装置は、PROFINET IOコントローラとの通信を妨げます

PROFINET IOコントローラとPSS 4000装置の間の通信は、PSS 4000装置のPROFINET IO DEVICEシステムセクションが有効化されている場合にのみ可能です。PROFINET IO DEVICEシステムセクションは、PSS 4000装置の少なくとも1つの仮想PROFINETモジュールをコンフィグレーションすることで、有効化されます。

PSS 4000装置で少なくとも1つの仮想PROFINETモジュールがコンフィグレーションされていることを確認してください。コンフィグレーションされていない場合、PROFINET IOコントローラはPSS 4000装置を検出できません。

9.9.4 PROFINETを経由したデータ交換

PROFINET IO DEVICEシステムセクションを備えたPSS 4000装置は、PROFINETでIO装置として動作します。データ転送を開始するのは、常にPROFINET IOコントローラです。オートメーションシステムPSS 4000から見た場合、PROFINET IOコントローラはシステムの外にあります。

9.9.4.1 データの送受信

PROFINET IOコントローラは、送信データ領域からPSS 4000装置のPROFINET IO DEVICEシステムセクションの受信データ領域にデータを送信し、PSS 4000装置のPROFINET IO DEVICEシステムセクションからのデータを要求します。PSS 4000装置のPROFINET IO DEVICEシステムセクションは、要求されたデータを送信データ領域からPROFINET IOコントローラの受信データ領域に送信します。

オートメーションシステムPSS 4000では、これらのデータ領域を「仮想PROFINETモジュール」と呼びます。仮想PROFINETモジュールは、さまざまなデータ型で使用できます。PROFINET IOコントローラおよびPSS 4000装置のPROFINET IO DEVICEシステムセクションのどちらも、対応する仮想PROFINETモジュールがデータ交換用に必要です。送信データ用仮想PROFINETモジュールを、出力モジュールと呼びます。モジュール名には、「Output (出力)」を表す「O」の文字が入ります (8OX、16OX、32OX、OB、OW、ODなど)。

受信データ用仮想PROFINETモジュールを、入力モジュールと呼びます。モジュール名には、「Input (入力)」を表す「I」の文字が入ります (8IX、16IX、32IX、IB、IW、IDなど)。

PSS 4000装置は、PROFINETのコンフィグレーションツール (STEP 7 – HW Configなど) 内にPROFINETプロジェクト専用に組み込まれます。たとえば、ここでPROFINET-IPアドレスやPROFINET装置名が割り付けられます。必要な仮想PROFINETモジュールは、GSDMLファイルによって使用可能になります ([「GSDMLファイル」 \[268\]](#)を参照)。

オートメーションシステムPSS 4000では、仮想PROFINETモジュールはPAS4000のPROFINET IO DEVICEエディタでコンフィグレーションされます。コンフィグレーション後に、仮想PROFINETモジュールが仮想スロットに配置されます (仮想スロット0～64)。

オートメーションシステムPSS 4000から見た場合、PROFINET経由の通信は外部通信です。受信データは外部入力データと呼ばれ、送信データは外部出力データと呼ばれます ([「処理データの有効性」 \[281\]](#)および[「代入値」 \[284\]](#)を参照)。

9.9.4.2 送信データ用仮想PROFINETモジュール

PROFINET IO DEVICEシステムセクションの送信データのために、PAS4000では、次の仮想PROFINETモジュール (出力モジュール) をコンフィグレーションできます。

仮想 PROFINETモジュール	データ型	アプリケーション
8OX	ARRAY [0..7] OF BOOL	仮想PROFINETビットモジュール データ型BOOLの8個のファンクションで 配列を送信
16OX	ARRAY [0..15] OF BOOL	仮想PROFINETビットモジュール データ型BOOLの16個のファンクションで 配列を送信
32OX	ARRAY [0..31] OF BOOL	仮想PROFINETビットモジュール データ型BOOLの32個のファンクションで 配列を送信
OB	BYTE	仮想PROFINETバイトモジュール データ型がBYTEのデータを送信
OW	WORD	仮想PROFINETバイトモジュール データ型WORDのデータを送信
OD	DWORD	仮想PROFINETバイトモジュール データ型DWORDのデータを送信
OBAn	ARRAY [0..(n-1)] OF BYTE	仮想PROFINETバイトモジュール データ型がBYTEのファンクションをn個持 つ配列を送信 n = 2、4、8、16、32、64のいずれか
OWAn	ARRAY [0..(n-1)] OF WORD	仮想PROFINETバイトモジュール データ型がWORDのファンクションをn個 持つ配列を送信 n = 2、4、8、16、32、64のいずれか
ODAn	ARRAY [0..(n-1)] OF DWORD	仮想PROFINETバイトモジュール データ型がDWORDのファンクションをn 個持つ配列を送信 n = 2、4、8、16、32、64のいずれか



情報

ここでは、指定されているデータ型で記載されている仮想モジュールのみがサポートされていることに注意してください。LWORDデータ型はサポートされていません。

PROFINET IOコントローラは、データ交換時に、オートメーションシステムPSS 4000の出力モジュールからのデータを要求して、入力モジュール経由でこのデータを受信します。したがって、オートメーションシステムPSS 4000の出力モジュールには、PROFINET IOコントローラ内に対応する入力モジュールが必要です。またPROFINET IOコントローラの入力モジュールのデータ型は、オートメーションシステムPSS 4000の出力モジュールのデータ型に対応する必要があります。

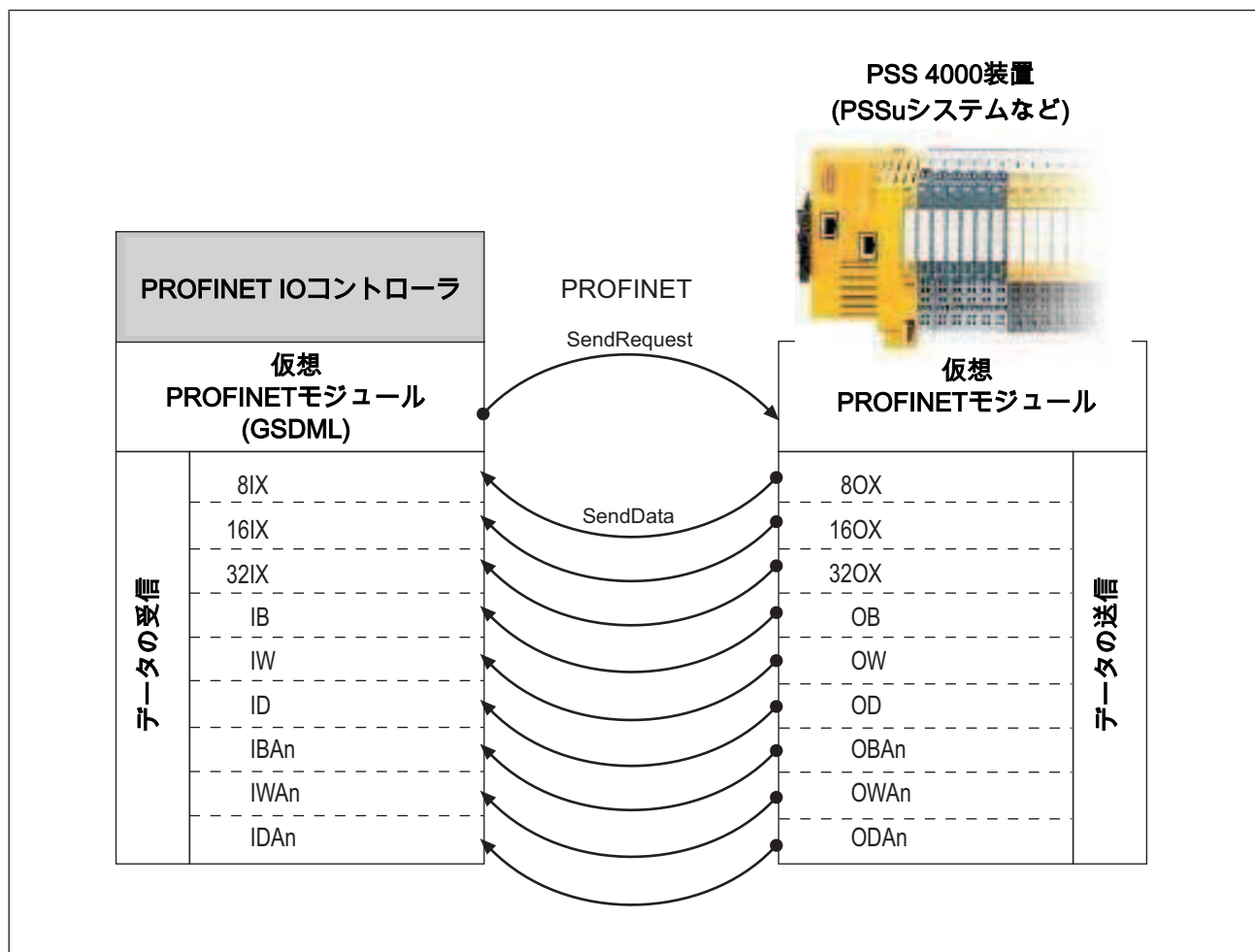


図: 仮想PROFINETモジュールとそれらに対応するデータ型 (PSS 4000の送信データ)

9.9.4.3 受信データ用仮想PROFINETモジュール

PROFINET IO DEVICEシステムセクションの受信データのために、PAS4000では、次の仮想PROFINETモジュール (入力モジュール) をコンフィグレーションできます。

仮想 PROFINET IOモジュール	データ型	アプリケーション
8IX	ARRAY [0..7] OF BOOL	仮想PROFINETビットモジュール データ型BOOLの8個のファンクションで 配列を受信
16IX	ARRAY [0..15] OF BOOL	仮想PROFINETビットモジュール データ型BOOLの16個のファンクションで 配列を受信
32IX	ARRAY [0..31] OF BOOL	仮想PROFINETビットモジュール データ型BOOLの32個のファンクションで 配列を受信
IB	BYTE	仮想PROFINETバイトモジュール データ型がBYTEのデータを受信
IW	WORD	仮想PROFINETバイトモジュール データ型WORDのデータを受信
ID	DWORD	仮想PROFINETバイトモジュール データ型DWORDのデータを受信
IBAn	ARRAY [0..(n-1)] OF BYTE	仮想PROFINETバイトモジュール データ型がBYTEのファンクションをn個持 つ配列を受信 n = 2、4、8、16、32、64のいずれか
IWAn	ARRAY [0..(n-1)] OF WORD	仮想PROFINETバイトモジュール データ型がWORDのファンクションをn個 持つ配列を受信 n = 2、4、8、16、32、64のいずれか
IDAn	ARRAY [0..(n-1)] OF DWORD	仮想PROFINETバイトモジュール データ型がDWORDのファンクションをn 個持つ配列を受信 n = 2、4、8、16、32、64のいずれか



情報

ここでは、指定されているデータ型で記載されている仮想モジュールのみがサポートされていることに注意してください。LWORDデータ型はサポートされていません。

PROFINET IOコントローラは、データ交換時に、オートメーションシステムPSS 4000の入力モジュールに出力モジュール経由で送信データを送信します。したがって、PROFINET IOコントローラの出力モジュールには、オートメーションシステムPSS 4000内に対応する入力モジュールが必要です。また入力モジュールのデータ型は、PROFINET IOコントローラの出力モジュールのデータ型に対応する必要があります。

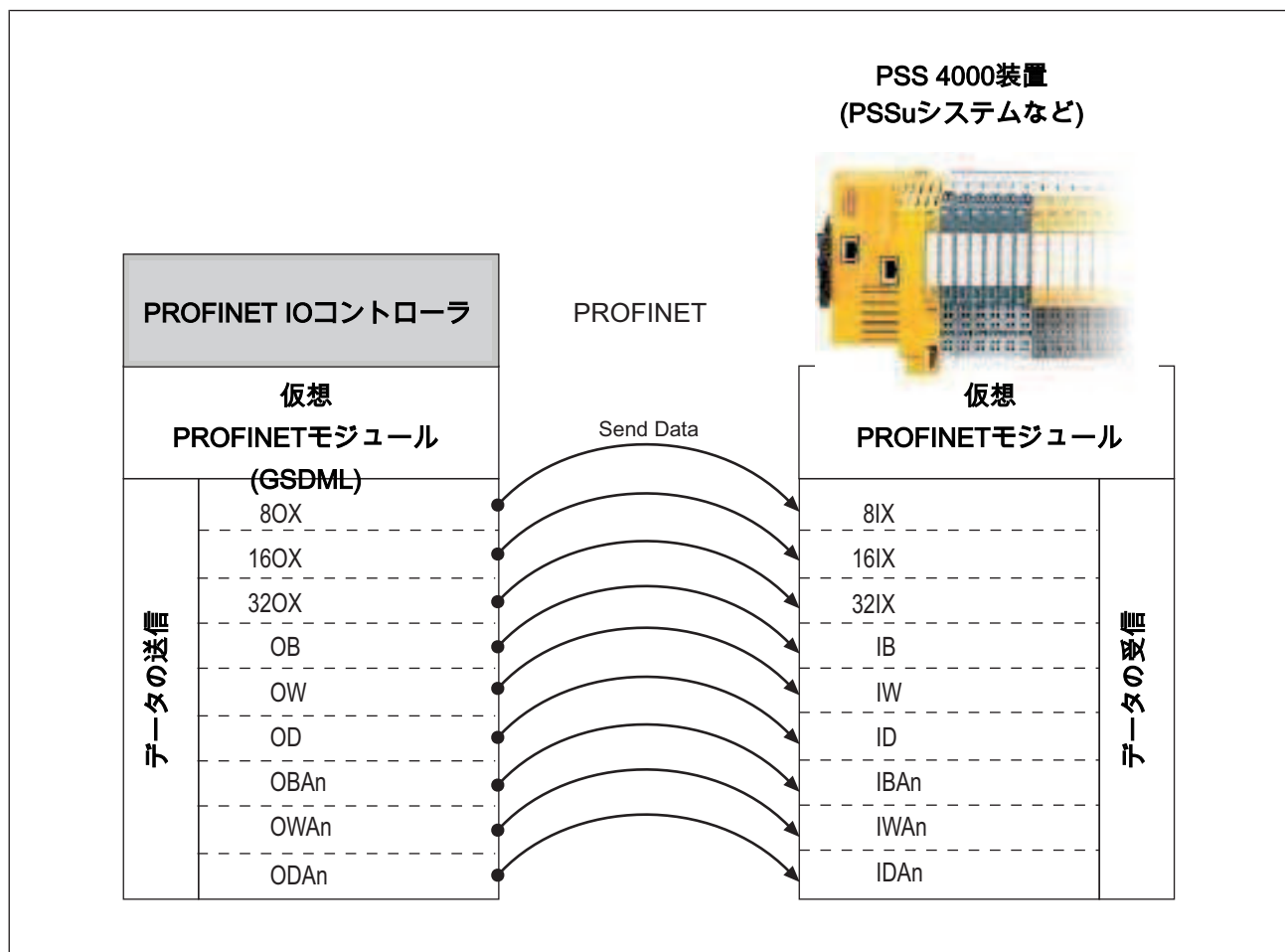


図: 仮想PROFINETモジュールとそれらに対応するデータ型 (PSS 4000の受信データ)

9.9.5 GSDMLファイル

PROFINET装置のすべての関連データは、対応するGSD (General Station Description) ファイルに含まれています。このファイルはXMLベースで、GSDMLファイルとしても知られています。このファイルには、PROFINET装置のプロパティと機能が記述されていて、これらはエンジニアリングおよびデータ交換において重要です。

PROFINET IO DEVICEシステムセクションを備えたPSS 4000装置のGSDMLファイルには、次のようなデータが含まれています。

- ▶ 使用可能なすべての仮想PROFINETモジュール
- ▶ さまざまな装置ベースのPROFINETアラームへの処置が含まれた診断メッセージ

GSDMLファイルはPAS4000に付属して提供されます。また、ピルツホームページのダウンロードエリア (www.pilz.comのダウンロードページ) でも入手できます。

9.9.6 データ転送の制限

仮想PROFINETモジュール経由のデータ転送には、以下の制限が適用されます。

- ▶ PSS 4000のPROFINET IO DEVICEシステムセクションに対してコンフィグレーション可能な仮想モジュールは最大で64台
- ▶ PSS 4000のPROFINET IO DEVICEシステムセクションに対してコンフィグレーション可能な受信データ (= 外部入力データ) は最大で1440 Byte
- ▶ PSS 4000のPROFINET IO DEVICEシステムセクションに対してコンフィグレーション可能な送信データ (= 外部出力データ) は最大で1440 Byte
- ▶ 外部出力データ用にコンフィグレーションされた各仮想PROFINETモジュールには、純粋なペイロードの他に、データフロー方向 (送信、受信) ごとに追加バイトが必要
- ▶ 外部入力データ用にコンフィグレーションされた各仮想PROFINETモジュールには、純粋なペイロードの他に、データフロー方向 (送信、受信) ごとに追加バイトが必要
- ▶ PSS 4000のPROFINET IO DEVICEシステムセクションには、各データフロー方向 (送信、受信) につき常に6 Byteが必要

バイト要求はPAS4000でのコンフィグレーション中に計算され表示されます。

例

PSS 4000のPROFINET IO DEVICEシステムセクションに対して、仮想32IXモジュールがコンフィグレーションされました。

	データフロー方向「受信」 (外部入力データ)	データフロー方向「送信」 (外部出力データ)
仮想32IXモジュール		
純粋なペイロードへのバイト要求	4 Byte	- - -
仮想モジュールごとのバイト要求	1 Byte	1 Byte
PROFINET IO DEVICEシステムセクション		
システムセクションごとのバイト要求	6 Byte	6 Byte
コンフィグレーションされた全モジュールに対する総バイト要求	11 Byte	7 Byte

計算されたバイト要求により、以下のバイト数が今後の仮想入出力モジュールのコンフィグレーションのために残ります。

- ▶ 仮想入力モジュール (外部入力データ、データフロー方向「受信」):
1440 Bytes - 11 Bytes = 1429 Bytes
- ▶ 仮想出力モジュール (外部出力データ、データフロー方向「送信」):
1440 Bytes - 7 Bytes = 1433 Bytes

9.9.7 PROFINET IO DEVICEのI/Oデータ

PROFINET IO DEVICEシステムセクションにより、PSS 4000とPROFINET IOコントローラは、PROFINETを経由してI/Oデータを交換できます。この操作を正常に実行するには、PAS4000のI/OマッピングエディタのI/Oデータを対応する仮想PROFINETモジュールにマッピングする必要があります。

仮想PROFINETモジュールのI/Oマッピング

I/Oマッピングを使用して、どのデータソースからどのデータシンクにデータを供給するかを定義します。

仮想PROFINETモジュールの可能なI/Oマッピング:

送信元／送信先		仮想PROFINETモジュール	
		受信データ (= 8IX、16IX、32IX、IB、IW、ID、IBAn、IWAn、IDAnなどの入力モジュールの外部入力データ)	送信データ (= 8OX、16OX、32OX、OB、OW、OD、OBAn、OWAn、ODAnなどの出力モジュールの外部出力データ)
モジュールバス	入力データ (例: ハードウェア入力)	---	◆
	出力データ (例: ハードウェア出力)	◆	---
PI 変数	入力PI変数	◆	---
	出力PI変数	---	◆

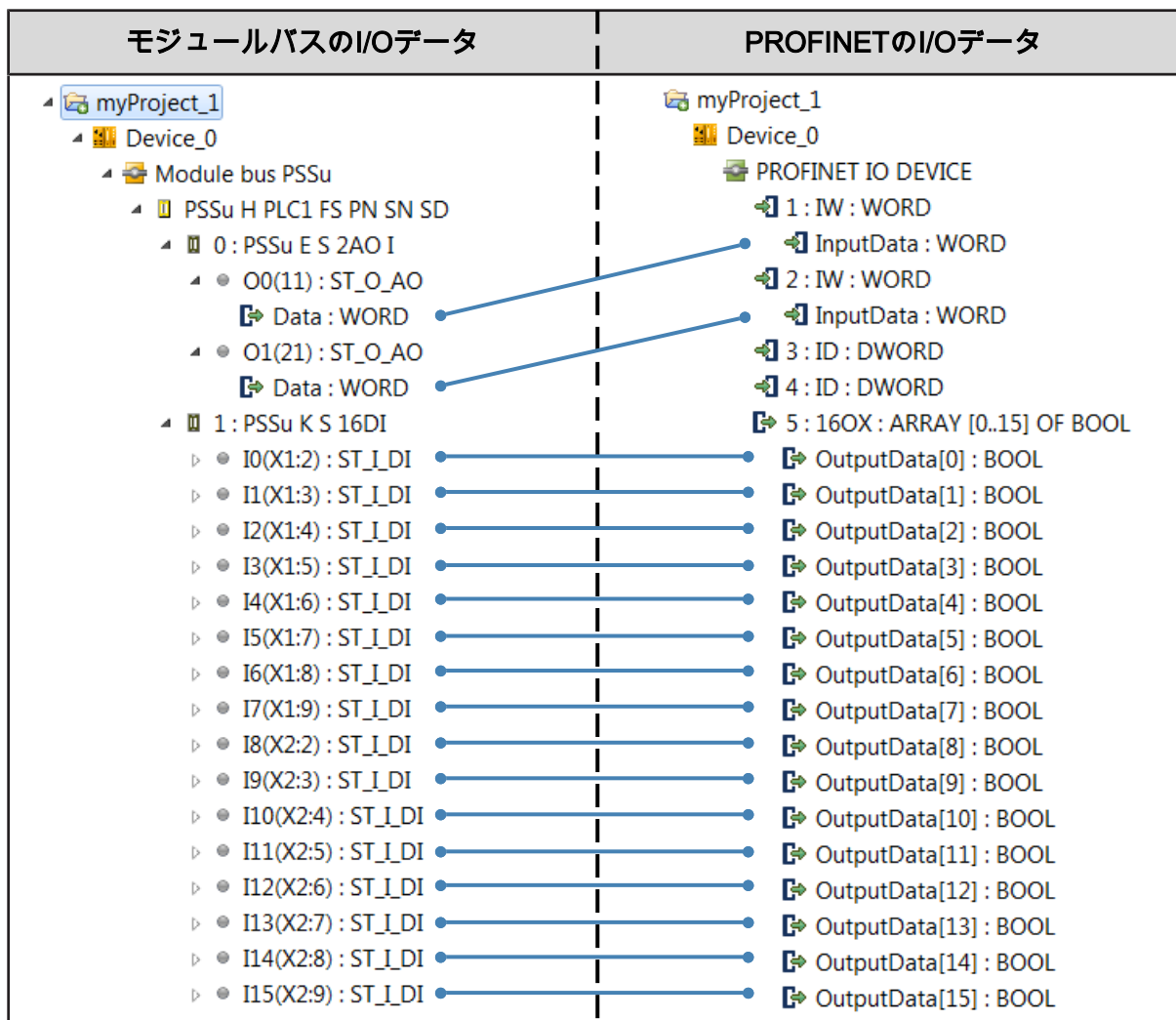
- ◆ I/Oマッピング可
 --- I/Oマッピング不可

例

▶ I/OデータのI/Oマッピング

PSSu K S 16DIのハードウェア入力の入力信号をPROFINET IOコントローラに送信し、PROFINET IOコントローラがPSSu E S 2 AO Iのハードウェア出力を制御します。

可能なI/Oマッピング (例):

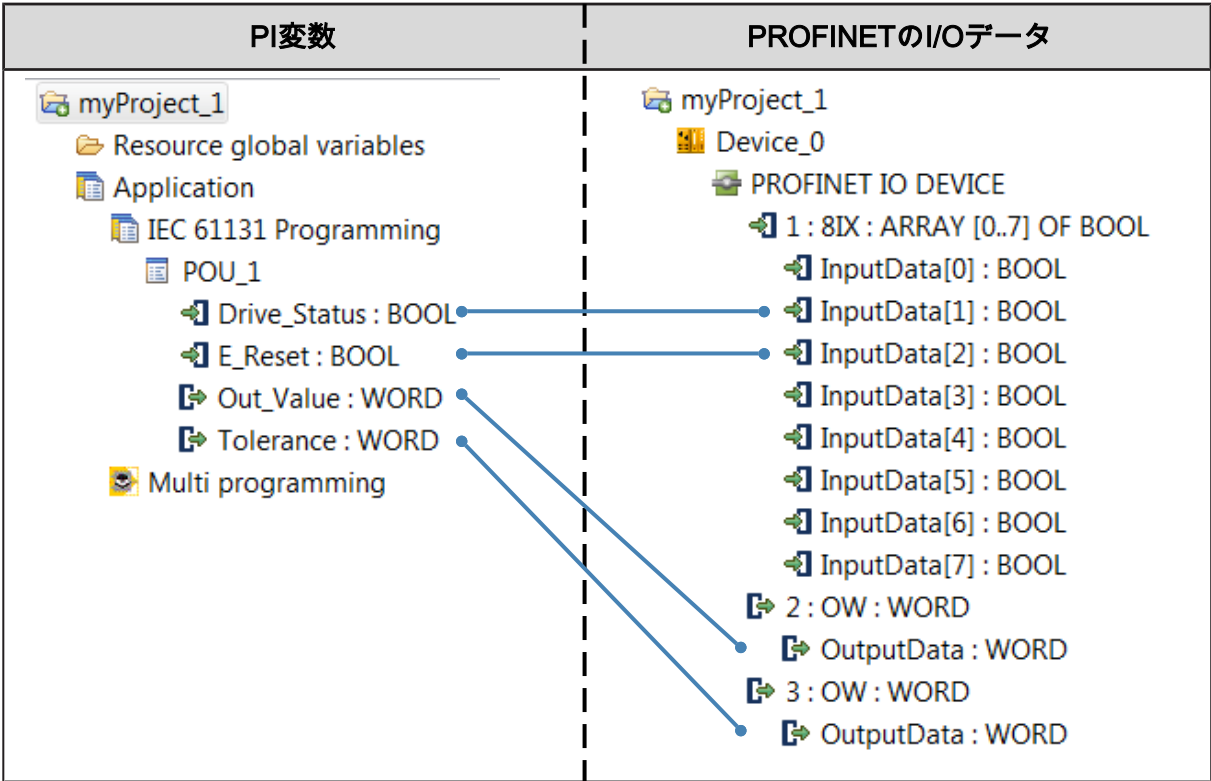


●—● I/Oマッピング

▶ PI変数のI/Oマッピング

ブロックの出力PI変数の内容をPROFINET IOコントローラに送信し、入力PI変数を受信データに割り付けます。

可能なI/Oマッピング (例):



● I/Oマッピング

9.9.8 PROFINET IOアドレスのアドレスフォーマット

PSS 4000装置のI/Oデータのアドレスは、PAS4000や診断など、複数の場所に表示されます。

9.9.8.1 送信データのアドレスフォーマット

PROFINET送信データのアドレス (仮想PROFINETモジュールの出力データ) は、次の形式で構造化されます。

<装置名>.ProfinetIODev.<仮想スロット>.OutputData[<ファンクション番号>]

例:

▶ 8OXのアドレスフォーマット:

```
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]
```

▶ 16OXのアドレスフォーマット:

```
myDevice_1.ProfinetIODev.2.OutputData[0]
...
myDevice_1.ProfinetIODev.2.OutputData[15]
```

▶ 32OXのアドレスフォーマット:

```
myDevice_1.ProfinetIODev.3.OutputData[0]
...
myDevice_1.ProfinetIODev.3.OutputData[31]
```

▶ OBのアドレスフォーマット:

```
myDevice_1.ProfinetIODev.4.OutputData
```

▶ OBA4のアドレスフォーマット:

```
myDevice_1.ProfinetIODev.5.OutputData[0]
...
myDevice_1.ProfinetIODev.5.OutputData[3]
```

▶ OWのアドレスフォーマット:

```
myDevice_1.ProfinetIODev.6.OutputData
```

▶ OWA2のアドレスフォーマット:

```
myDevice_1.ProfinetIODev.7.OutputData[0]
myDevice_1.ProfinetIODev.7.OutputData[1]
```

▶ ODのアドレスフォーマット:

```
myDevice_1.ProfinetIODev.8.OutputData
```

▶ ODA4のアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.9.InputData[0]
...
myDevice_1.ProfinetIODevice.9.InputData[3]
```

9.9.8.2 受信データのアドレスフォーマット

PROFINET受信データのアドレス (仮想PROFINETモジュールの入力データ) は、次の形式で構造化されます。

<装置名>.ProfinetIODevice.<仮想スロット>.InputData[<ファンクション番号>]

例:

▶ 8IXのアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.10.InputData[0]
myDevice_1.ProfinetIODevice.10.InputData[1]
myDevice_1.ProfinetIODevice.10.InputData[2]
myDevice_1.ProfinetIODevice.10.InputData[3]
myDevice_1.ProfinetIODevice.10.InputData[4]
myDevice_1.ProfinetIODevice.10.InputData[5]
myDevice_1.ProfinetIODevice.10.InputData[6]
myDevice_1.ProfinetIODevice.10.InputData[7]
```

▶ 16IXのアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.11.InputData[0]
...
myDevice_1.ProfinetIODevice.11.InputData[15]
```

▶ 32IXのアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.12.InputData[0]
...
myDevice_1.ProfinetIODevice.12.InputData[31]
```

▶ IBのアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.13.InputData
```

▶ IBA2のアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.14.InputData[0]
myDevice_1.ProfinetIODevice.14.InputData[1]
```

▶ IWのアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.15.InputData
```

▶ IWA8のアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.16.InputData[0]
...
myDevice_1.ProfinetIODevice.16.InputData[7]
```

▶ IDのアドレスフォーマット:

```
myDevice_1.ProfinetIODevice.17.InputData
```

▶ IDA4のアドレスフォーマット:

myDevice_1.ProfinetIODev.18.InputData[0]

...

myDevice_1.ProfinetIODev.18.InputData[3]

9.9.9 PROFINET IOの診断

PROFINET IOの診断は、PSS 4000のシステム診断を通じて実行できます (「[診断の基本情報](#)」[\[566\]](#)を参照)。

ただし、個別の診断メカニズムはPROFINETで定義します。PROFINETで、IO装置はアラームおよびステータスメッセージをIOコントローラに送信できます。

PROFINET IO DEVICEシステムセクションのあるPSS 4000装置は、PROFINETアラーム形式の装置ベースの診断をサポートしています。診断メッセージは各PROFINETアラームタイプに割り付けられて、装置経由で送信されます。診断メッセージは装置固有のGSDMLファイルに格納され、対策情報も含まれます。また各診断メッセージは、装置のステータスLEDのライト信号にも対応しています。

適切なPROFINET診断ツールにより、各PROFINETアラームに対応する診断メッセージと対策情報を表示できます。

PAS4000 (PROFINET IO DEVICEエディタ) でのコンフィグレーション中に装置ベースの診断が有効化されていると、PROFINET IO装置として、装置がPROFINETアラームを使って以下の装置ステータスの信号を送信できます。

アラームタイプ (PROFINET IO DEVICEエディタで有効化)	メッセージテキスト (GSDMLファイル)	LED表示 (装置)	
装置に関するエラーメッセージ	1つ以上のシステムセクションに重大度が「エラー」レベルのメッセージがある (診断リストを参照)	DIAG	赤の点灯または赤の点滅
モジュールバスでのエラー	オペレーティングステータス「装置のすべてのFS出力が安全状態」または、少なくとも1つのモジュールにアクセスできない (操作中にモジュールが削除された、設定されたハードウェアレジストリが実際の値と一致しないなど)	MBUS	赤の点灯または赤の点滅
FS SafetyNET pでのエラー	オペレーティングステータス「FS SafetyNET pがエラーでSTOP状態: FSの重大エラー	FS SNp	赤の点灯または赤の点滅
ST SafetyNET pでのエラー	オペレーティングステータス「ST SafetyNET pがエラーでSTOP状態: FSおよびSTの重大エラー	ST SNp	赤の点灯または赤の点滅
装置に関する警告メッセージ 注: 装置に対するいずれかのエラーメッセージが同時に現れた場合は、警告メッセージの信号は送信されません。	装置に重大度が「警告」レベル以上のメッセージがある (診断リストを参照)	DIAG	オレンジ色のLED
FSリソースに対する強制、オンライン変更	FSリソースで、強制が有効であるか、1つ以上のオンライン変更が有効 (またはその両方) です。	FS FORCE	黄色で点灯
STリソースに対する強制、オンライン変更	STリソースで、強制が有効であるか、1つ以上のオンライン変更が有効 (またはその両方) です。	ST FORCE	黄色で点灯

9.9.10 PROFINET IOのデータ整合性

PROFINET IOコントローラが送信する受信データのデータ整合性に対しては、PROFINET IOコントローラに責任があります。PSS 4000から見た場合、PROFINET IOコントローラのデータ整合性に関するステートメントは作成できません。

受信データの整合性

PSS 4000から見た場合、仮想PROFINETモジュールの受信データは、外部入力データになります。外部入力データが仮想PROFINETモジュールで使用可能になると、すぐに一貫したデータとして処理されます（「[データのー貫性](#)」 300」を参照）。仮想PROFINETモジュールの外部入力データは、整合性範囲型「外部通信接続のST入力データ」の整合性範囲を形成します。正確な記述は、整合性範囲「PROFINET IO接続のST-Iデータ」です（「[ー貫性の範囲](#)」 301」を参照）。

送信データの整合性

PSS 4000から見た場合、仮想PROFINETモジュールの送信データは、外部出力データになります。データソースの同じ整合性範囲から取得した場合（「[データのー貫性](#)」 300」を参照）、仮想PROFINETモジュールの外部出力データには整合性があります。

PSS 4000では、仮想PROFINETモジュールの外部出力データは、常に1つの単位として処理されます。必要なデータの整合性を保証することは、PROFINET IOコントローラの責任です。

10 プロジェクト

10.1 プロジェクト構造体

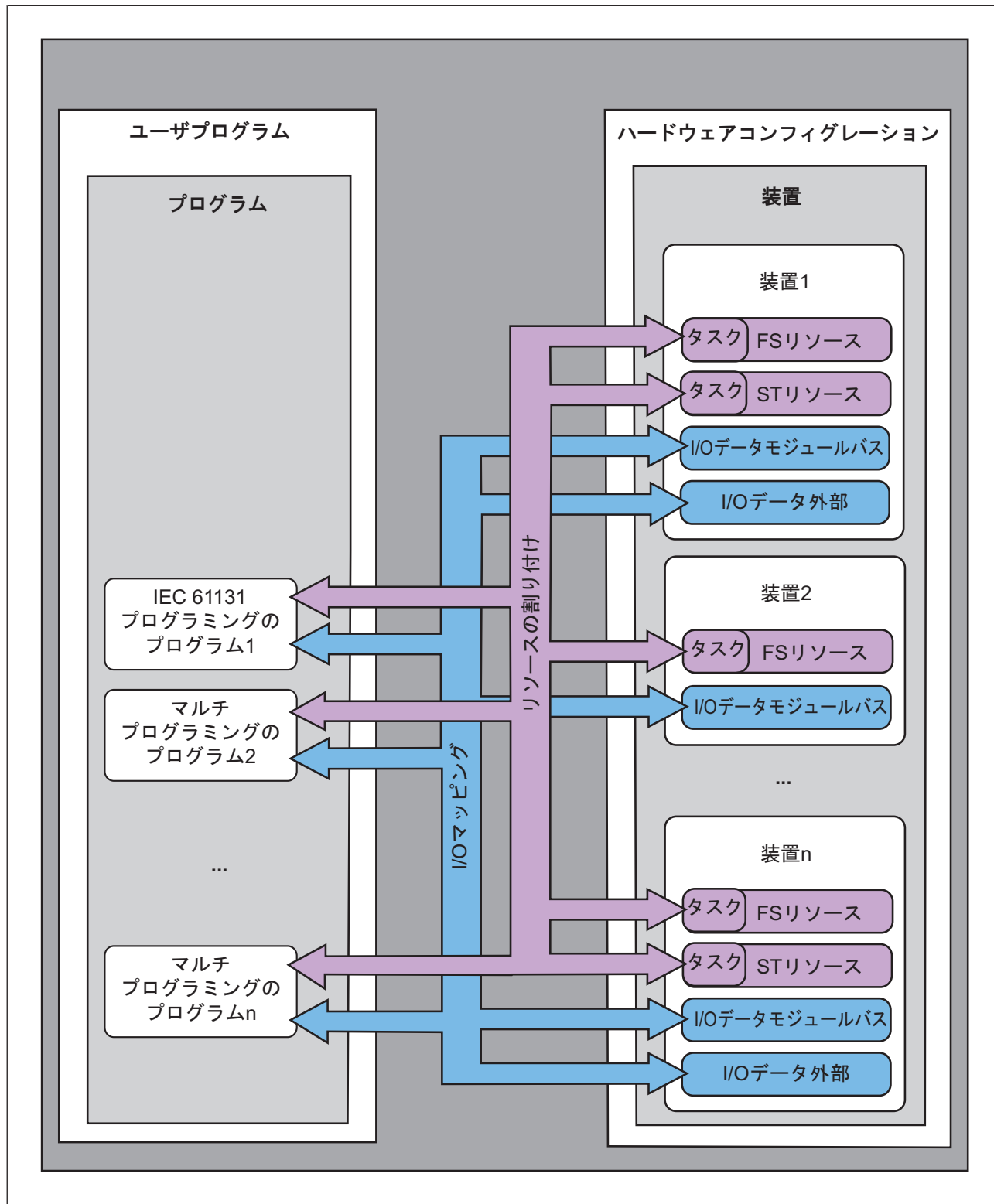


図: プロジェクト構造体

オートメーションシステムPSS 4000を使用して設備を操作するには、プロジェクトをPAS4000で作成する必要があります。

プロジェクトはハードウェアコンフィグレーションとユーザプログラムで構成されています。

ハードウェアのコンフィグレーションは、プロジェクトで使用される装置と、装置で使用される設定を指定します。たとえば、コンフィグレーションは各PSSuシステムに対して指定されています。装置によって処理または提供されるモジュールバスI/Oデータは、I/Oモジュールの指定から派生しています。外部通信は、外部通信を使用する装置 (例: EtherNet/IP) にもコンフィグレーションできます。これにより外部I/Oデータが生成され、装置によって処理または提供されます。

ユーザプログラムは1つ以上のプログラムで構成されています。パフォーマンスクラスPSS 4000 PLCの制御システムのプログラムは、IEC 61131プログラミングとMultiプログラミングで作成できます。パフォーマンスクラスPSS 4000 multiの制御システムのプログラムは、Multiプログラミングのみで作成できます。

▶ IEC 61131プログラミング

IEC 61131に従ってプログラムする場合、ユーザプログラムの構成に3つのタイプのプログラム構成要素が使用できます。

- プログラム (PRG)
- ファンクションブロック (FB)
- ファンクション (FUN)

プログラムは、より高度な構造レベルを形成します。ファンクションおよびファンクションブロックは、プログラム内で呼び出すことができます。

ファンクションブロックとファンクションは、ユーザプログラム内で個々のタスクを実行します。

POUは、必要に応じてIL (インストラクションリスト) またはSTL (構造化テキスト) プログラミング言語で記述できます。

▶ Multiプログラミング

Multiエディタで、コンポーネントモデルに応じてMultiプログラミングを実行します。ユーザプログラムを構造化するために3つのタイプのブロックが利用できます。

- Multiプログラム

Multiプログラムは、ユーザプログラムを構造化するために排他的に使用されます。部品ブロックと基本ブロックは、Multiプログラムで呼び出すことができます。

- 部品ブロック (CB)

部品ブロックは、部品ブロックおよび／または基本ブロックの結合に使用されます。この方法で、オートメーション機能に対応する複数の分離したサブソリューションを組み合わせることができます。コンポーネントブロックは、設備と機械の構造を再構成するために利用できます。

部品ブロックは、簡単に再利用できる自己完結型のユニットです。

- 基本ブロック (BB)

基本ブロックは、複雑なオートメーションソリューションの実装に使用されます。



情報

一般的に、用語「ブロック」は、タイププログラム、ファンクションブロック、およびファンクションのMultiプログラム、部品ブロック、基本ブロック、およびプログラム構成要素に使用されている用語です。

ブロックおよびI/Oハードウェアデータ間の接続は、「PI変数」で行われます。PI変数は、ブロック間のデータ交換にも使用できます。

リソースグローバル変数を使用しない場合、使用するハードウェアに関係なくユーザプログラムを作成できます。ユーザプログラムのどのセクションが、どのタスクのどのリソース長で実行されるのかという情報は、プログラミングが完了するまではリソース割り付けで定義する必要はありません。タスクは、実行時の優先度など、ユーザプログラムのセクションのランタイムプロパティを決定します。

プログラミングが完了すると、PI変数とオートメーションシステムI/Oデータ間の接続を設定する必要があります。これはI/Oマッピングで実行されます。

10.2 処理データの有効性

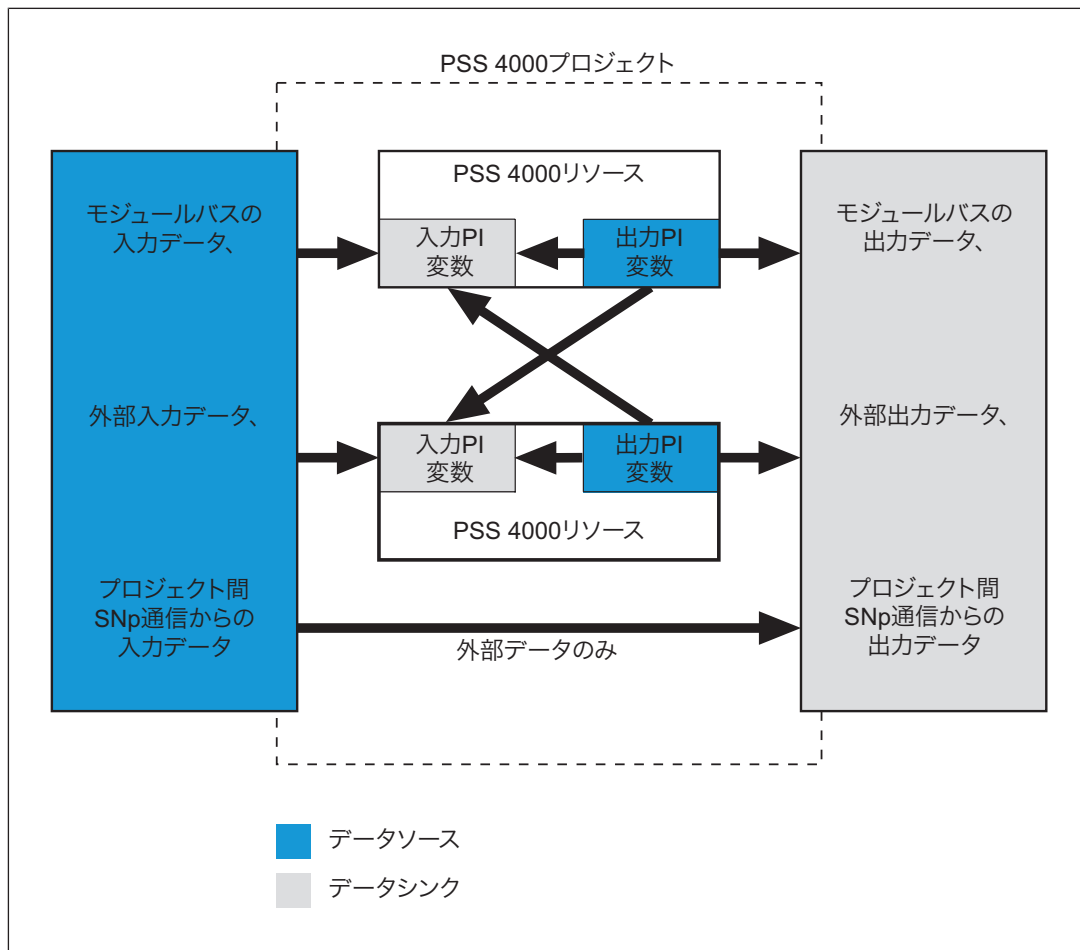


図: プロジェクト内のデータフロー

プロセスデータには、I/OデータとPI変数が含まれます。プロジェクト内で、プロセスデータはデータソースとデータシンクの間で交換されます。

データソース:

- ▶ モジュールバスの入力データ
PSS 4000専用装置のモジュールなど、FS/STモジュールバスのシステムセクションから取得された入力データ
- ▶ 外部入力データ
EtherNet/IPなど、外部通信のためにシステムセクションから取得された入力データ
- ▶ 出力PI変数
FSリソースまたはSTリソースのシステムセクションなど、PSS 4000リソースから取得された出力PI変数。リソースは、同じ装置にある場合も異なる装置にある場合もあります。
- ▶ プロジェクト間SNp通信からの入力データ
他のPSS 4000プロジェクトからの出力データ (出力PI変数など)

データシンク:

- ▶ モジュールバスの出力データ
PSS 4000専用装置のモジュールにダウンロードされる場合など、FS/STモジュールバスのシステムセクションに書き込まれる出力データ
- ▶ 外部出力データ
EtherNet/IPなど、外部通信のためにシステムセクションから外部に転送される出力データ
- ▶ 入力PI変数
FSリソースまたはSTリソースのシステムセクションなど、PSS 4000リソースから取得される入力PI変数
- ▶ プロジェクト間SNp通信からの出力データ
他のPSS 4000プロジェクトからの入力データ (入力PI変数など)

矢印を使ってデータフローが図に示されます。I/Oマッピングは、どのデータソースがどのデータシンクにデータを供給するかを定義します。

データフロー内で、データソース、またはデータソースとデータシンク間の通信チャンネルに問題が発生した可能性があります。したがって、データシンクは、データソースから有効なプロセス値を受信していません。受信したプロセス値が有効であるかどうかをデータシンクが識別できるように、データシンク内のほとんどすべてのプロセス値に「有効ビット」と呼ばれる値が付いています。プロジェクト間SNp通信からの入力データのみ有効ビットがありません。

データソース内の処理データが無効とマークされた場合、またはデータソース内の処理データにアクセスできない場合、有効ビットはFALSEになります。

有効ビットがFALSEである場合、プロセス値が無効になり、使用することができません。FALSEであった場合、データシンクにはプロセス値の代わりに使用される代入値があります。

有効ビットがFALSEに設定される状況:

- ▶ モジュールバスの入力データの場合
 - 入力時のエラー (例: テストパルスエラー)
 - I/Oモジュールのエラー (例: モジュールのオーバーヒート)
 - モジュールバスでのエラー
 - モジュールバスの起動
- ▶ 外部入力データの場合
 - 通信の停止
 - 通信のタイムアウト
- ▶ 出力PI変数
 - タスクがSTOP状態
 - リソースがSTOP状態

備考:

- ▶ 起動動作
起動時に、データソース (例: ローカルモジュールバス) がまだ有効なデータを提供していないにもかかわらず、装置のリソースでユーザプログラムがすでに実行中になっていることがあります。この場合、ユーザプログラムは代入値 [284] で処理を継続します。データソースがあるシステムセクションが稼働状態になると、ただちに、代入値ではなく現在のデータで処理が実行されます。有効ビットを評価して、この信号の

変更により不要な応答が発生しないことを確認できます (「[ユーザプログラムでの有効なビットの評価](#)」 [ 285]を参照)。

ユーザプログラムでデータ进行处理している各システムセクション (STリソース、FSリソース、STモジュールバス、FSモジュールバス、ST SafetyNET p、FS SafetyNET p、EtherNet/IPなどの外部通信のためのシステムセクション) から有効ビットを取得して、ランダムサンプルとして評価する必要があります。たとえば、入力PI変数がローカルFSモジュールバス上の入力データに割り付けられ、入力PI変数の有効ビットがTRUEである場合、ローカルFSモジュールバスが実行中であると結論付けることができます。個々のFS入力モジュールがエラーステータスになっている可能性があります。入力PI変数が別の装置のSTモジュールバスにある入力データに割り付けられており、その入力PI変数の有効ビットがTRUEである場合、別の装置のSTモジュールバスと、ST SafetyNET p経由の通信が、両方とも稼働中であると結論付けることができます。他の装置にある個々のST入力モジュールがエラーステータスになっている可能性があります。

▶ オンライン操作:「強制」

有効なビットが宣言された入力PI変数を強制する際、有効なビットは自動的にTRUEの値に強制されます。

例外:

- ユーザが有効ビットの強制値をFALSEに変更する場合。
- ユーザが有効ビットの強制を無効化する場合。
- 入力PI変数は、書き込み／読み取り保護を設定したブロックや保護ブロックに属しています。この場合、有効なビットを強制することはできません。

10.2.1 代入値

処理データの代入値は固定されています。

代入値には、次のデータを使用できます。

- ▶ ゼロ (= 安全状態)
- ▶ 最後の有効な値

処理データ	代入値
FSリソースの入力PI変数	ゼロ
FSビットモジュールの出力	ゼロ
FSバイトモジュールの出力データ	ゼロ
STリソースの入力PI変数	ゼロ
STビットモジュールの出力	ゼロ
STバイトモジュールの出力データ	ゼロ
ST出力データ: PROFIBUS-DPスレーブ	ゼロ
ST出力データ: Modbus/TCPサーバ接続	最後の有効な値
ST出力データ: Modbus/TCPクライアント接続	ゼロ
ST出力データ: Raw TCP	ゼロ
ST出力データ: Raw UDP	ゼロ
ST出力データ: EtherNet/IPアダプタ	ゼロ
ST-Oデータ: PROFINET IO DEVICE	ゼロ
FS出力データ: PROFIsafe搭載PROFIBUS-DPスレーブ	ゼロ

10.2.2 ユーザプログラムの有効なビットの評価

データソースが有効な処理データを提供できない場合、データシンクの有効ビットがFALSEに設定され、データシンクに代入値が割り付けられます。この処理は、有効ビットを評価して監視できます。

入力PI変数の有効ビットは、ユーザプログラムで評価できます。他のデータシンクから取得した有効ビットの監視は、まだ実行できません。

入力PI変数がマルチファンクション変数 (たとえば、I/Oデータ型ST_I_AIの構造体) である場合、構造体全体の有効ビットは、構造の個々のファンクションの有効ビットと同じになります。

Multiプログラミングでの有効ビットの評価

PSS 4000システムブロック「VALID」が有効ビットの評価に使用されます。

IEC 61131プログラミングでの有効ビットの評価

有効ビットを評価するために、入力PI変数に対して、VALID拡張子を宣言する必要があります。

10.3 タスクの基本情報

タスクはブロックのランタイムプロパティ (Multiプログラミング: Multiプログラム、コンポーネントブロック、ベーシックブロック、IEC 61131プログラミング: プログラムタイプPOU) を決定します。たとえば、条件、頻度、および実行される優先度です。

1つのタスクに1つ以上のブロックが割り付けられます。

リソースには最大9個のタスクを含めることができます。タスク数は装置に応じて変わります。詳細については、装置の取扱説明書を参照してください。

タスクの実行はシステムによって構成されます。タスクのプロパティがタスクを実行するタイミングと順番に影響します。



情報

多くのプロジェクトでは、タスクのプロパティをコンフィグレーションする必要はありません。既定のコンフィグレーションで十分です (「[既定のタスクのコンフィグレーション](#)」 291) を参照)。したがって、「タスク」の問題に対処する必要はありません。

10.3.1 タスクのプロパティ

タスクのプロパティはタスクコンフィグレーションで定義されます。

▶ 名前

名前は ID の規則 に従い、リソースごとに一意でなければなりません。

▶ タイプ

以下のタイプが使用できます。「周期性タスク」と「周期性タスク、排他的」。いずれのタスクタイプの動作も同じですが、「周期性タスク、排他的」は1つのブロック (サブブロックでも可) にだけ割り付けることができます。

周期性タスクは、定義可能なタスクサイクル時間の間に1回だけ実行されます。実行の間隔はさまざまです。

実行がタスクサイクル時間の間に完了しなかった場合、オペレーティングステータス「リソースがタスクエラーでRUN状態」が表示されます。

▶ 優先順位

各タスクの優先度を定義できます。優先度は、リソースの他のタスクと比較した各タスクの重要度です。つまり、リソースのタスクが実行される順序は優先度で決まります。優先度には、高、中、低があります。リソースに優先度を割り付けることができる回数は指定されています。たとえばタスクが9個あるリソースでは、優先度の高いタスクを3個、中程度のタスクを3個、低いタスクを3個使用できます。

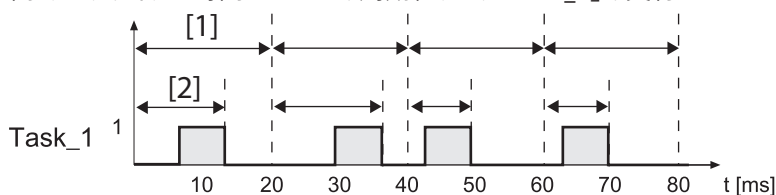
多くのタスクを高い優先度でプログラミングすると、PAS4000では、特定のオンライン機能 (変数の監視、オンラインモニタなど) の実行や、PSS 4000 OPCサーバから装置上のデータへのアクセスが低速になる可能性があります。

▶ タスクサイクル時間 (周期性タスクの場合のみ)

タスクサイクル時間は、タスクを実行する必要がある期間を定義します。

タスクサイクル時間は、二次的な優先度としても機能します。タスクの優先度が同じ場合は、タスクサイクル時間の最も短いタスクが先に実行されます。

例: タスクサイクル時間20 msでの周期性タスク「Task_1」の実行



[1] タスクサイクル時間

[2] 実行時間



情報

制御システムのアベイラビリティを向上させるため、タスクサイクル時間の値は最大実行時間に20 %を加算した値以上に設定することをお勧めします。

最大実行時間 は、設備の動作があらゆるオペレーティングステータスを経由するように、比較的長い時間 (たとえば、6時間超) をかけて決定してください。

この推奨に従わなくても、制御システムの安全機能に影響が及ぶことはありません。

10.3.2 タスクの実行

オートメーションシステムPSS 4000のコントローラは、プリエンプティブスケジューリングを使ったマルチタスクシステムです。1つのリソースで最大9個のタスクを使用できます。タスク数は装置に応じて変わります。詳細については、装置の取扱説明書を参照してください。

タスクの実行はシステムによって構成されます。タスクのプロパティがタスクを実行するタイミングと順番に影響します。優先度の高いタスクが保留になっている場合、優先度の低いタスクの実行は中断されます。

タスクを実行する際は、次の規則が適用されます。

▶ 実行時間

実行時間は、サイクルが開始された時点からタスクの実行が完了した時点までの時間です。タスクの実行が別のタスクによって中断された場合、実行時間は増加します。

▶ 装置で制御される実行

プロジェクト内の各装置は、タスクの実行を独立して制御します。装置ごとにFSリソースとSTリソースを1つずつ持つことができ、リソースには最大9個のタスクを含めることができます。タスク数は装置に応じて変わります。詳細については、装置の取扱説明書を参照してください。

リソースに優先度を割り付けることができる回数は指定されています。たとえばタスクが9個あるリソースでは、優先度の高いタスクを3個、中程度のタスクを3個、低いタスクを3個使用できます。

リソースには、同じ優先度で同じサイクル時間のタスクを2つ含めてはなりません。例外: タスクタイプが「周期性タスク」のタスクと「周期性タスク、排他的」のタスクには、同じ優先度とサイクル時間を割り当てることができます。

▶ 優先度による順番

タスクは優先度に従って実行されます。優先度が高いタスクが最初に実行され、次に優先度が中位のタスクが実行され、最後に低いタスクが実行されます。タスクの優先度が同じである場合、タスクサイクル時間によって決められます。タスクサイクル時間が最も短いタスクが最初に実行されます。優先度とタスクサイクル時間が同じタスクが複数存在している場合、実行の順序はランダムになります。

▶ FSリソースとSTリソースの連携

装置にFSリソースとSTリソースがある場合は、どちらも同じCPUによって実行されます。次の順序になります。

- 1.FSタスク、高優先度
- 2.STタスク、高優先度
- 3.FSタスク、中優先度
- 4.STタスク、中優先度
- 5.FSタスク、低優先度
- 6.STタスク、低優先度

優先度が高いFSタスクの実行が常に優先されます。

▶ 保留中のタスクの定期的なチェック

システムクロックパルス毎に、システムは実行が保留されているタスクがないかどうかを確認し、実行の順序を定義し直します。あるタスクが実行されているときに、それよりも優先度の高いタスクが保留されている場合は、実行中のタスクが中断されます。

▶ 実行順序

タスクの内部で複数のプログラムを実行する必要がある場合、プログラムは名前のアルファベット順に実行されます。

Multiプログラミング: Multiプログラムの内部での基本ブロックの実行順序は、PAS4000により定義されます。

▶ プロセスイメージとリソースグローバル変数の読み取りと出力

タスクの実行が開始されると、タスク内で、ブロックで必要とされるプロセスイメージ (入力PIおよび出力PI) のセクションからコピーが作成されます。必要なリソースグローバル変数もコピーされます。

タスクが実行されると、コピーから取得した値のみが読み取られ、書き込まれます。

タスクが完了するまで、言い換えると、すべてのタスクのブロックが実行されるまで、コピーから取得したデータは、リソースグローバル変数と出力 (出力PI) のプロセスイメージに転送されません。

この手順に従うと、タスクの実行中は、データを外部から変更できません。実行中にタスクが中断された場合でも、それ以降はコピー内のデータで処理が続行されます。

▶ タスクの中断

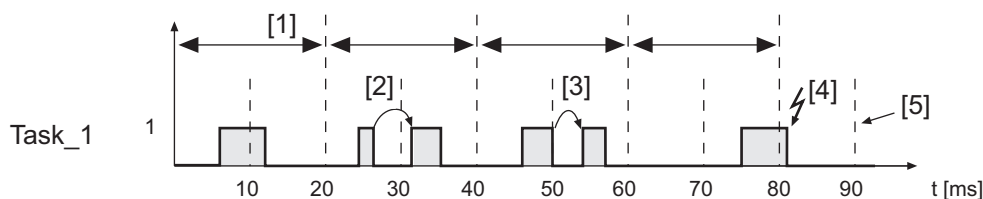
優先度が高いタスクの他にも、タスクを中断できるイベントが存在しています。また、ブロックがシステムファンクションの実行待ち状態である場合なども、タスクは中断されます。したがって、ブロックを実行する場合、優先度が高いタスクでブロックが以前に実行されたかどうか、現在のデータがPIで使用可能であるかどうかは保証されません。特定のタスクが、別のタスクから取得したデータを処理する場合、カウンタを評価する方法などで、別のタスクから受け取る必須データが、すでに処理されていることを確認する必要があります。

▶ タスクコンフィギュレーションによるサイクル時間の超過

優先度が低いタスクでサイクル時間が不必要に超過するのを防ぐには、リソースで複数のタスクをコンフィギュレーションする際に次の規則に従う必要があります。タスクのサイクル時間は、優先度が上がるにつれて短くなるようにします。例5を参照してください。

例1

周期性タスク「Task_1」の実行 (タスクサイクル時間20 ms、システムクロックパルス10 ms)



[1] タスクのサイクル時間

[2]、[3] 中断

[4] エラー

[5] システムクロックパルス

システムクロックパルス毎に、システムは実行を中断しているタスクの有無を確認します。タスク「Task_1」は、20 msごとに実行可能になります。ただし、優先度が高いタスクが最初に実行されるため、いつでもすぐ実行できるとは限りません。また、たとえば、プログラムがシステムファンクションの実行を待っている場合

[2] や、優先度が高いタスクがシステムクロックパルスに存在している場合 [3] に、これらが原因となって、タスクが中断されることもあります。タスクサイクル時間の期間内にタスクの実行が完了しない場合、エラーがトリガされます [4]。

例2

中優先度のタスクが実行されています。高優先度のタスクがシステムクロックパルスに存在します。最初のタスクが中断され、優先度が高いタスクが実行されます。次に最初のタスクの実行が続行されます。

例3

タスクサイクル時間が20 msの高優先度のタスクが実行されています。タスクサイクル時間が10 msの高優先度のタスクがシステムクロックパルスに存在します。最初のタスクが中断され、タスクサイクル時間が短いタスクが実行されます。

例4

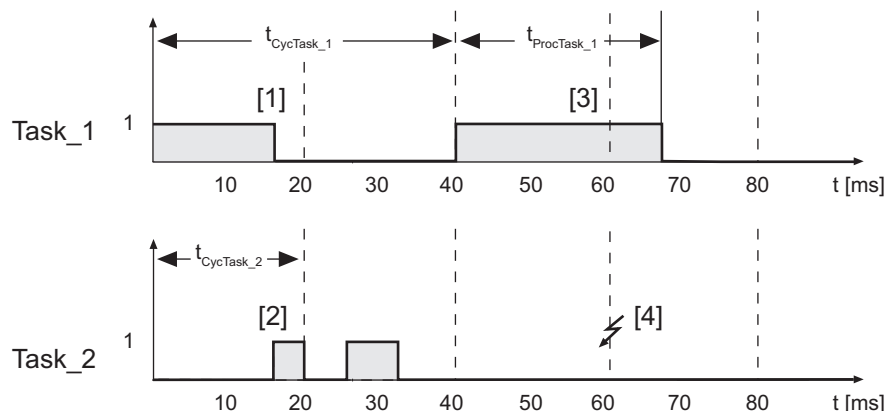
中優先度のSTタスクが実行されています。中優先度のFSタスクがシステムクロックパルスに存在します。STタスクが中断され、FSタスクが最初に実行されます。

例5

1つのリソースで2つのタスク、「Task_1」および「Task_2」。

サイクル時間40 msと優先度「高」が「Task_1」にコンフィグレーションされています。

サイクル時間20 msと優先度「低」が「Task_2」にコンフィグレーションされています。



優先度が高いため、「Task_1」が完全に実行されてから [1]、「Task_2」の実行が開始されます [2]。

「Task_1」の実行時間 $t_{ProcTask_1}$ が「Task_2」のサイクル時間 $t_{CycTask_2}$ よりも短い場合にのみ、「Task_2」はサイクル時間内に実行できます。

「Task_1」の実行時間 $t_{ProcTask_1}$ が「Task_2」のサイクル時間 $t_{CycTask_2}$ よりも長い場合 [3]、「Task_2」は時間内に実行できず、サイクル時間は超過します [4]。

サイクル時間の超過を防ぐには、次の選択肢があります。

- ▶ 「Task_2」の優先度を「Task_1」より高くします。
- ▶ 「Task_2」のサイクル時間を「Task_1」のサイクル時間以上の長さにします。
- ▶ 「Task_1」で実行されるブロックを最適化し、「Task_1」の実行時間が「Task_2」のサイクル時間より短くなるようにします。

10.3.3 タスクの既定のコンフィグレーション

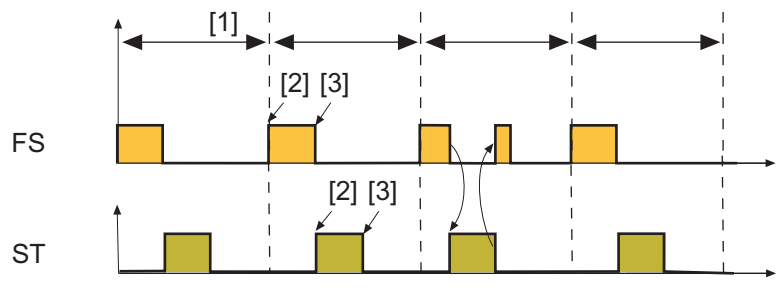
多くのプロジェクトでは、タスクのプロパティをコンフィグレーションする必要はありません。既定のコンフィグレーションで十分です。

既定のコンフィグレーションでは、装置上でFSタスクが最初に行われ、次にSTタスクが実行されるよう設計されています。たとえば、FSブロックがシステムファンクションの実行待ち状態となる状況が発生することがあります。この場合、FSタスクの実行は中断され、STタスクが実行されます。

タスクの内部で複数のプログラムを実行する必要がある場合、プログラムは名前のアルファベット順に実行されます。

Multiプログラミング: Multiプログラムの内部での基本ブロックの実行順序は、PAS4000により定義されます。

FSタスクとSTタスクは、タスクサイクル時間100 msの期間内に一回実行されます。



[1] タスクのサイクル時間:

[2] リソースグローバル変数と入力 (入力PI) / 出力 (出力PI) のプロセスイメージが読み取られます。

[3] リソースグローバル変数の出力と出力 (出力PI) のプロセスイメージ

10.4

PSS 4000プロジェクト内のI/Oマッピングルール

I/Oマッピングを行う際は、どのデータソースがどのデータシンクにマッピングされるかを考慮する必要があります。また、データソースとデータシンクのデータ型も考慮に入れるようにしてください。



情報

これは、プロジェクト内のデータフローに対してI/Oマッピングが表示される場所です。これらのI/Oマッピングは、PAS4000内で行われます。

プロジェクト間データフロー（プロジェクト間SNp通信）に対するI/OマッピングはPASconnectで行われます。それに適用される特別なルールについては「[PSS 4000プロジェクト間I/Oマッピングのルール](#) [📖 296]」を参照してください。

データソースとデータシンク

I/Oマッピングは、どのデータソースがどのデータシンクにデータを供給するかを定義します。次の図では、プロジェクト内部で理論的に可能なデータの流れを矢印で示しています。

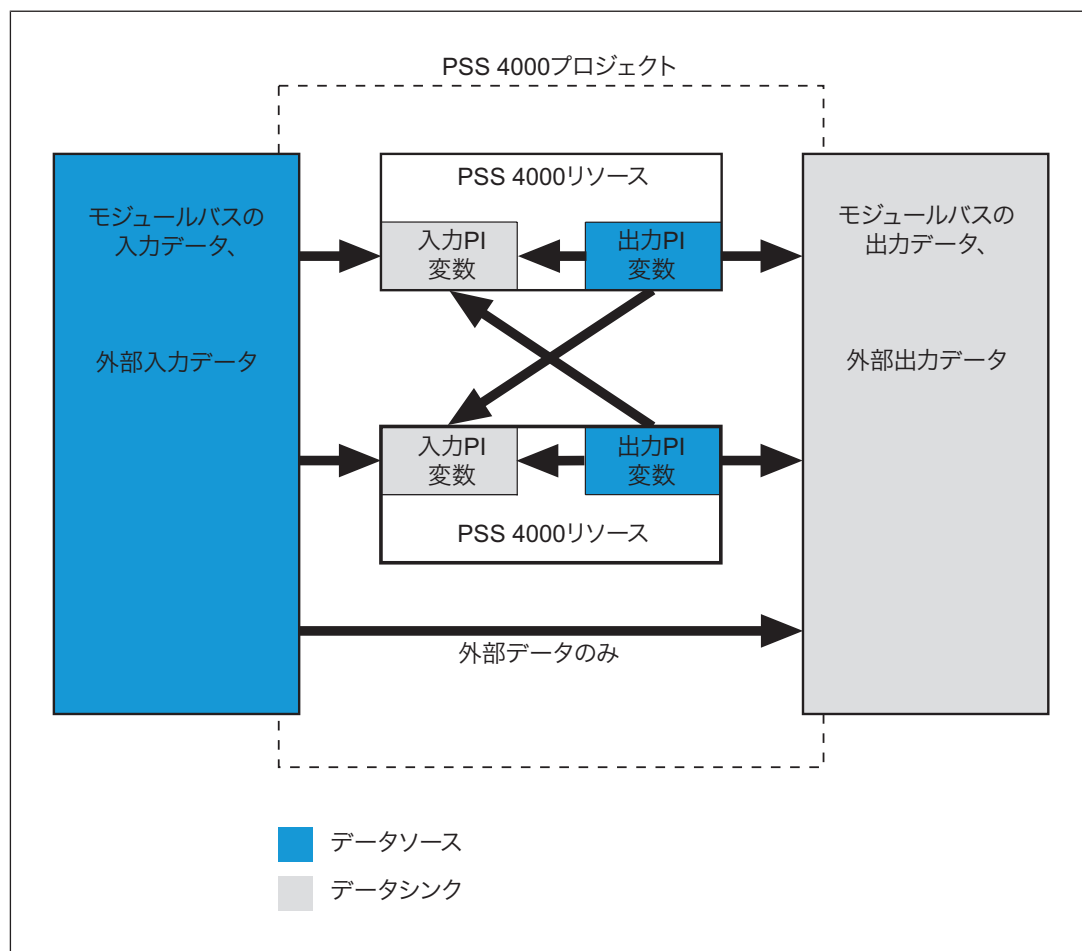


図: プロジェクト間SNp通信のないプロジェクト内データフロー

プロセスデータには、I/OデータとPI変数が含まれます。プロジェクト内で、プロセスデータはデータソースとデータシンクの間で交換されます。

データソース:

▶ モジュールバスの入力データ

PSS 4000専用装置のモジュールなど、FS/STモジュールバスのシステムセクションから取得された入力データ

▶ 外部入力データ

EtherNet/IPなど、外部通信のためにシステムセクションから取得された入力データ

▶ 出力PI変数

FSリソースまたはSTリソースのシステムセクションなど、PSS 4000リソースから取得された出力PI変数。リソースは、同じ装置にある場合も異なる装置にある場合もあります。

データシンク:

▶ モジュールバスの出力データ

PSS 4000専用装置のモジュールにダウンロードされる場合など、FS/STモジュールバスのシステムセクションに書き込まれる出力データ

▶ 外部出力データ

EtherNet/IPなど、外部通信のためにシステムセクションから外部に転送される出力データ

▶ 入力PI変数

FSリソースまたはSTリソースのシステムセクションなど、PSS 4000リソースから取得される入力PI変数

許可されているI/Oマッピングを示す概要です。

I/Oマッピング		送り先 (データシンク)					
		モジュールバスのST出力データ	モジュールバスのFS出力データ	ST入力PI変数	FS入力PI変数	外部ST出力データ	外部FS出力データ
送り元 (データソース)	モジュールバスのST入力データ	---	---	◆	---	◆	---
	モジュールバスのFS入力データ	---	---	+	◆	+	◆
	ST出力PI変数	◆	++	◆	---	◆	---
	FS出力PI変数	+	◆	+	◆	+	◆
	外部ST入力データ	◆	++	◆	---	---	---
	外部FS入力データ	+	◆	+	◆	---	---

凡例:

--- 不可 (PAS4000により禁止)

◆ 可

++ 可、ただしFSデータは非安全関連タスクでのみ使用可

+

可、ただしSTデータは安全に関連するタスクでは依然使用不可

I/Oマッピングを行う際の注意点

▶ FSデータソースとSTデータシンク間のI/Oマッピング

FSデータソースのデータがSTデータシンクのデータにマッピングされた場合、STデータシンクのデータは安全関連タスクで使用することはできません。

つまり、ST出力データ (STハードウェア出力、外部通信用ST出力データなど) がFSリソースを介して制御されている場合、この出力データは**非安全関連タスク**にのみ使用できます。

PAS4000は、このようなI/Oマッピングに対して警告します。

STデータシンクのへの使用をアプリケーションが必要とする場合は、以下のような適切な方法により安全性を確認する必要があります。

- 個別のリスク分析と
- 適切な試運転テスト、動作テスト

正しい実装はユーザの責任において行ってください。

▶ STデータソースとFSデータシンク間のI/Oマッピング

STデータソースのデータがFSデータシンクのデータにマッピングされている場合、FSデータシンクからのデータを安全関連タスクで使用することはできません。PAS4000は、このようなI/Oマッピングに対して警告します。該当するI/Oマッピングの例:

- ST出力PI変数からモジュールバスのFS出力データへのI/Oマッピング
- 外部ST入力データからモジュールバスのFS出力データへのI/Oマッピング

アプリケーションでFSデータシンクを安全関連タスクに使用する必要がある場合は、以下のような適切な方法により安全性を確認する必要があります。

- 個別のリスク分析と
- 適切な試運転テスト、動作テスト

正しい実装はユーザの責任において行ってください。

▶ 同じタスク内のPI変数とPI変数の間のI/Oマッピング

出力PI変数を記述した後、次のタスクサイクルが開始されるまで、マッピングした入力PI変数に値が読み込まれるようにすることはできません。

データ型

データソースとデータシンクのデータ型に対してI/Oマッピングを行う際は、次の事項が全般的に適用されます。

- ▶ データソースとデータシンクのデータ型が匿名データ型の場合、これらは同じ構造でなければなりません。
- ▶ データソースとデータシンクのデータ型が名前付きデータ型の場合、これらは同一でなければなりません。
- ▶ FSとSTの間のI/Oマッピングは上記のルールに従って許可されています。すなわち、たとえばBOOLとSAFEBOOLの間などもI/Oマッピングが行われます。

データ型がARRAY (配列) またはSTRUCT (構造体) のI/OデータまたはPI変数には、次の追加の規則が適用されます。

▶ データシンクとしての入力PI変数

入力PI変数のデータ型がARRAY (配列) またはSTRUCT (構造体) の場合、個々のファンクションへのI/Oマッピングは許可**されません**。ただし、データソースを配列または構造体のファンクションにすることはできます。

▶ データシンクとしての出力データ

出力データのデータ型がARRAY (配列) またはSTRUCT (構造体) の場合、I/Oマッピングは、配列／構造体全体に対しても、配列／構造体の個々のファンクションに対しても可能です。

10.5 PSS 4000プロジェクト間I/Oマッピングのルール

PSS 4000プロジェクト間のデータ交換は、I/Oマッピングを使用して、ソフトウェアツールPASconnectでコンフィグレーションされます。I/Oマッピングを行う際は、どのデータソースがどのデータシンクにマッピングされるかを考慮する必要があります。また、データソースとデータシンクのデータ型も考慮に入れるようにしてください。

データソースとデータシンク

I/Oマッピングは、どのデータソースがどのデータシンクにデータを供給するかを定義します。次の図では、プロジェクト間で理論的に可能なデータの流れを矢印で示しています。

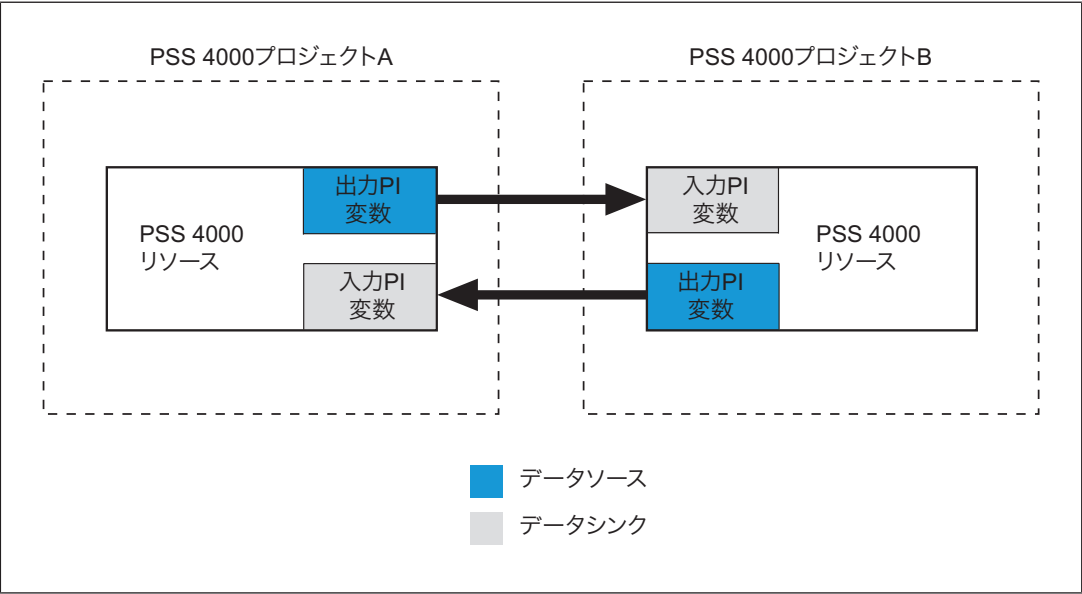


図: プロジェクト間のデータフロー

データソース:

- ▶ 出力PI変数
プロジェクト装置のFSリソースまたはSTリソースにあるユーザプログラムからの任意のローカル出力PI変数リソースのグローバル変数として宣言されている出力PI変数は、プロジェクト間SNp通信に有効なデータソースではありません。

データシンク:

- ▶ 入力PI変数
入力PI変数にVALID拡張子が宣言されておらず (有効ビット)、プロジェクト内で入力PI変数へのI/Oマッピングがまだ行われていない場合の、プロジェクト装置のFSリソースまたはSTリソースにあるユーザプログラムからの任意のローカル入力PI変数リソースのグローバル変数として宣言されている入力PI変数は、プロジェクト間SNp通信に有効なデータシンクではありません。

一覧は許可されているI/Oマッピングを示します。

I/Oマッピング		送り先 (データシンク)	
		ST入力PI変数	FS入力PI変数
送り元 (データソース)	ST出力PI変数	◆	---
	FS出力PI変数	+	◆

凡例:

- 不可 (PASconnectにより禁止)
- ◆ 可
- + 可、ただしSTデータは安全に関連するタスクでは依然使用不可

I/Oマッピングを行う際の注意点

▶ FSデータソースとSTデータシンク間のI/Oマッピング

FSデータソースのデータがSTデータシンクのデータにマッピングされている場合、STデータシンクからのデータを安全関連タスクで使用することはできません。PASconnectは、このようなI/Oマッピングに対して警告します。

STデータシンクのへの使用をアプリケーションが必要とする場合は、以下のような適切な方法により安全性を確認する必要があります。

- 個別のリスク分析と
- 適切な試運転テスト、動作テスト

正しい実装はユーザの責任において行ってください。

▶ データ転送の制限

SafetyNET pに関するPSS 4000のシステム説明に記載されている[データ転送の制限](#) [📖 107]が、PSS 4000プロジェクト間のSNp通信に適用されます。

プロジェクト内SNp通信とは対照的に、2つの装置間のプロジェクト間通信では、以下の通り各方向のタスクにつき最大FS-Tx接続1つとST-Tx接続1つがあります。

- 装置Aから装置Bへのタスクごとに1つのFS-Tx接続
- 装置Bから装置Aへのタスクごとに1つのFS-Tx接続
- 装置Aから装置Bへのタスクごとに1つのST-Tx接続
- 装置Bから装置Aへのタスクごとに1つのST-Tx接続

1つのFS-Tx接続で最大118 BytesのFS処理データ (FSデータ型のデータ) の転送が可能

1つのST-Tx接続で最大251 BytesのST処理データ (STデータ型のデータ) の転送が可能

▶ データ型

データソースとデータシンクのデータ型に対してI/Oマッピングを行う際は、次の事項が適用されます。

- データソースとデータシンクのデータ型は、基本データ型にする必要があります。派生データ型間のI/Oマッピングは許可されていません。例: ARRAYデータ型 (配列) またはSTRUCTデータ型 (構造)
- データソースとデータシンクのデータ型名は一致する必要がありますが、SAFEの接頭語は考慮しません。
- FSとSTの間のI/Oマッピングは上記のルールに従って許可されています。すなわち、たとえばBOOLとSAFEBOOLの間などもI/Oマッピングが行われます。

▶ PSS 4000プロジェクト間のI/Oマッピング

PSS 4000プロジェクト間でI/Oマッピングを実行する場合は、データソースをデータシンクにマッピングするときに特別な注意を払う必要があります。特に、データソースとデータシンクを正しく識別するには、データソースとデータシンクの選択元のプロジェクトをメモしておくことが重要です。

▶ PSS 4000プロジェクト間のI/Oマッピングの変更点

PASconnectプロジェクトのI/Oマッピングが変更された場合は、影響を受けるPSS 4000プロジェクトを再ビルドする必要があります。影響を受けるプロジェクトは、PASconnectインタフェースのバージョンによって異なります（以下を参照してください）。プロジェクトが再ビルドおよびダウンロードされない場合、PSS 4000プロジェクト内のSafetyNET pコンフィギュレーションの整合性がなくなります。この場合、伝送されるデータが受信側で想定されているデータと一致しないことがあります。

PSS 4000プロジェクトがビルドされた後で、試運転（PSS 4000安全マニュアルの「試運転と有効性」を参照）または再試運転（PSS 4000安全マニュアルの「変更、メンテナンス、廃棄」の「変更」を参照）を行う必要があります。

▶ PASconnectインタフェースのバージョン1.2.x: I/Oマッピングの変更の影響を受けるPSS 4000プロジェクト

PASconnectプロジェクトのI/Oマッピングが変更された場合、I/Oマッピングが変更された双方のPSS 4000プロジェクト、およびデータをこれらのプロジェクトと交換しているすべてのプロジェクトを再ビルドする必要があります。

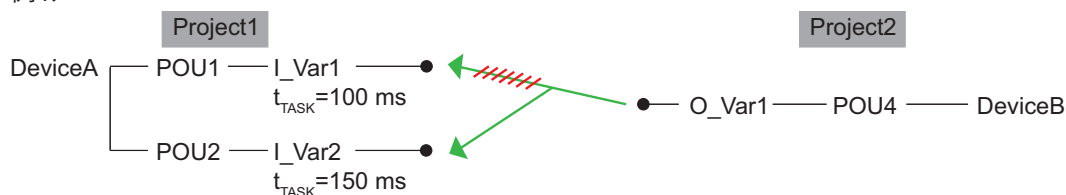
▶ PASconnectインタフェースのバージョン1.3.x/1.4.x: I/Oマッピングの変更の影響を受けるPSS 4000プロジェクト

新しいI/Oマッピングを作成したり、既存のI/Oマッピングを削除した場合は、結果として、関係していた2つのPSS 4000プロジェクトのPASconnect出力データが変更されるため、両方のプロジェクトを再ビルドする必要があります。

一方のI/Oマッピングを変更した場合は、必ずしも両方のPSS 4000プロジェクトを再ビルドする必要はありません。次の要件が満たされている場合は、一方のI/Oマッピングを変更したときに、もう一方のPSS 4000プロジェクトのPASconnect出力データを変更する必要はありません。

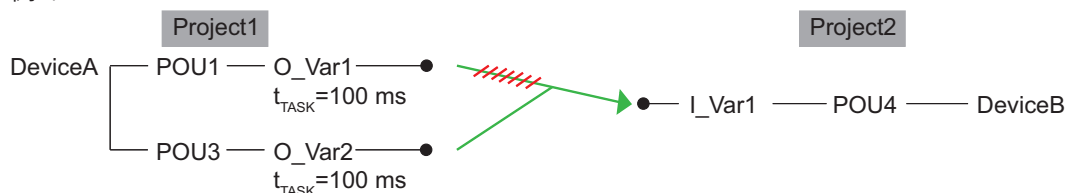
- 新しい変数が古い変数と同じ装置に存在している必要がある。
- 出力PI変数の場合は、タスクサイクル時間を変更してはいけない。
- PASconnectプロジェクトで、オプション **[ロケーション情報を含める]** を無効にする必要がある。

例1:



変更後、Project 1の入力PI変数I_Var1とI_Var2は同じ装置上にあるため、Project 2を再ビルドする必要はありません。

例2:



変更後、Project 1の出力PI変数O_Var1とO_Var2は同じ装置上にあり、POUのタスクサイクル時間が同じであるため、Project 2を再ビルドする必要はありません。

▶ 排他性

PSS 4000プロジェクト内のI/Oマッピングに使用される入力PI変数は、プロジェクト間SNp通信には使用できません。

10.6 データの一貫性

PSS 4000におけるデータの整合性は、一般的に、同じタスクサイクルに記録され、定義された期間中に最新と見なされて関連付けられたデータを一緒に保持することであると理解されています。

データソースのデータの整合性

データの整合性の観点から、データソース（「[プロセスデータの有効性](#)  281」を参照）は「整合性の範囲」に分割されます。データソースの整合性の範囲には、整合性のあるデータの量が含まれます。つまり、データソースのプロセスデータが、データソースの同じ整合性の範囲からのデータであり、かつ有効である場合、そのデータは整合性があるということになります。

データとプロジェクト内SafetyNET p通信との整合性

1つのPSS 4000プロジェクト内で転送される整合性範囲内のプロセスデータは、単一SafetyNET p接続を使用して転送される限り整合性があります。

プロジェクト内部のデータ転送中に、記述された[SafetyNET p制限](#)  107を超過した場合、整合性範囲のデータは自動的に複数の接続に分かれます。この場合、データの整合性は保証されなくなります。ビルド処理中に、整合性の範囲のプロセスデータがいくつかの接続に分かれた場合、PAS4000は警告を発行します。

データとプロジェクト間SafetyNET p通信との整合性

プロジェクトをまたいで別のPSS 4000プロジェクトに転送される整合性範囲内のプロセスデータは、単一SafetyNET p接続を使用して転送される限り整合性があります（「[PSS 4000プロジェクト間I/Oマッピングのルール](#)  296」の「データ転送の制限」を参照）。

プロジェクト間SNp通信中に、単一SNp接続を使用してプロセスデータを転送できない場合は、PASconnectがSNp接続を阻止します。

データシンクのデータの整合性

データシンクの処理データ（「[処理データの有効性](#)  281」を参照）は、データソースの同じ整合性の範囲からのデータである場合、整合性があります。さまざまな整合性の範囲からのデータである場合、データシンクの処理データ間の関係に関してステートメントは作成できません。

10.6.1 一貫性の範囲

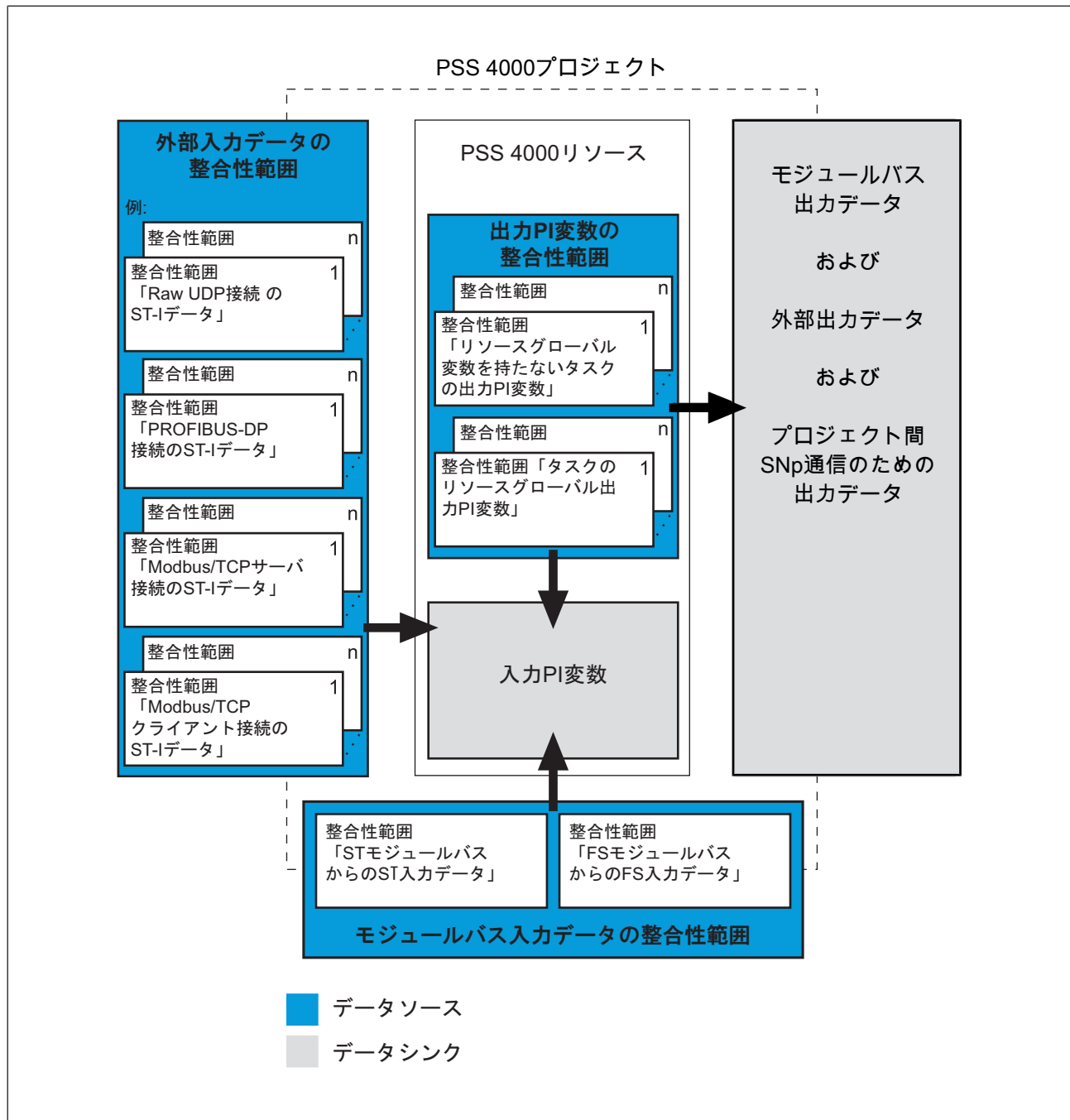


図: プロジェクトの整合性の範囲の概要

プロジェクトには、次の整合性の範囲があることがあります。

▶ **モジュールバス入力データ**

- 整合性の範囲「FSモジュールバスのFS入力データ」
- 整合性の範囲「STモジュールバスのST入力データ」

▶ **外部入力データ**

- 整合性の範囲「外部通信接続のST入力データ」

- 整合性の範囲「外部通信接続のFS入力データ」

外部通信を介して外部装置からPSS 4000装置に転送されたデータの対応するシステムセクションとのデータの整合性に関してステートメントは作成できません。通信モード (例: Modbus/TCP、PROFIBUS-DP、Raw UDP)、および通信モードで利用できるオプションによって決まります。

データがPSS 4000で外部入力データとして使用できるようになると、整合性のあるデータとして扱われます。

あらゆる通信モードまたは外部通信接続タイプの外部入力データに対して整合性の範囲があります。外部入力データに対するそれぞれの整合性の範囲は、そのデータの整合性の観点から同じように動作します。多くの場合、外部入力データに対して全般的な整合性の範囲のみが指定されます (整合性の範囲「外部通信接続のST入力データ」または整合性の範囲「外部通信接続のFS入力データ」)。ただし、必要に応じて、各通信モードまたは各接続タイプに対する整合性の範囲は個別に指定される場合もあります。

整合性の範囲「外部通信接続のST入力データ」の例:

- 整合性の範囲「Modbus/TCPクライアント接続のST入力データ」
- 整合性の範囲「Modbus/TCPサーバ接続のST入力データ」
- 整合性の範囲「PROFIBUS-DP接続のST入力データ」
- 整合性の範囲「Raw UDP接続のST入力データ」

整合性の範囲「外部通信接続のFS入力データ」の例:

- 整合性の範囲「仮想PROFIsafe FモジュールのFS入力データ」

▶ 出力PI変数

- 整合性の範囲「リソースグローバル変数を使用しないタスクの出力PI変数」

このような整合性の範囲には、同一タスクで実行されるブロックの出力PI変数すべてが含まれます。このタイプの整合性の範囲では、このタスクで実行されるリソースグローバル出力PI変数との関係に関してステートメントは作成できません。

- 整合性の範囲「タスクのリソースグローバル出力PI変数」

整合性の範囲「リソースグローバル変数を使用しないタスクの出力PI変数」に対するリソースグローバル変数との関係に関してステートメントは作成できません。これは、同一タスク (下記の例にある「Task 1」を参照) 内で実行される場合も適用されます。

次の点もリソースグローバル出力PI変数のデータの整合性に適用されます。

- 同一タスクで実行されるブロックから排他的にアクセスされるリソースグローバル出力PI変数は整合性があり、整合性の範囲を形成します (下記の「Task 1」および整合性の範囲 [1] および [2] を参照)。

- 異なるタスクで実行されるブロックからアクセスされるリソースグローバル出力PI変数の共通部分は整合性があり、整合性の範囲を形成します (下記の「Task 2」、「Task 3」および整合性の範囲 [3] を参照)。

- 入力PI変数とリソースグローバル入力PI変数にマッピングされる整合性の範囲、および同一タスクで実行されるブロックからアクセスされる整合性の範囲内の変数のデータの整合性に関してステートメントは作成できません。

例:

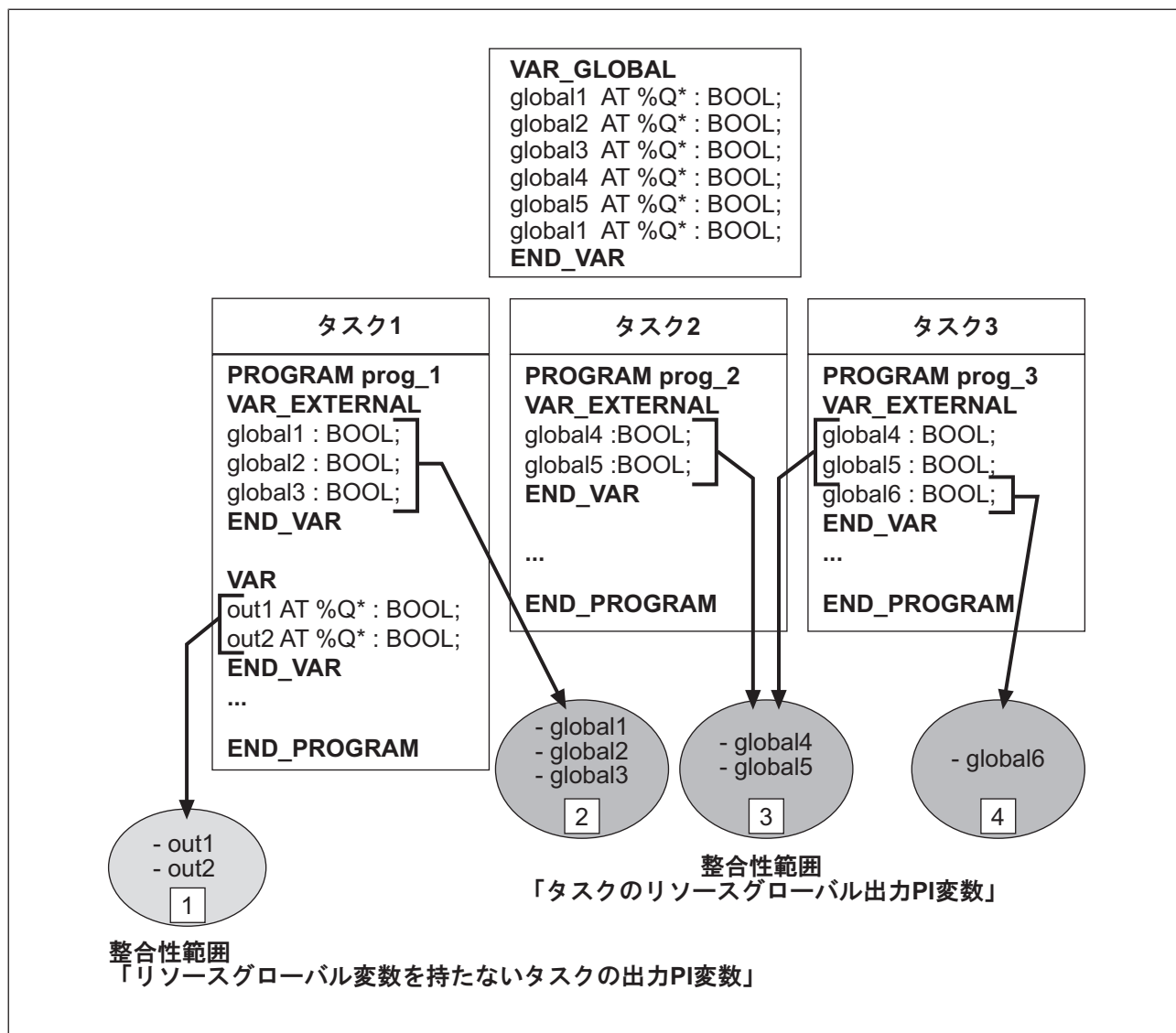


図: 出力PI変数の整合性の範囲の例

10.6.2 データの整合性に対するスケジューリングの影響

スケジューリングは、プロジェクトのデータが処理される順番を規制し、さらに、さまざまなサイクル時間、実行時間、処理時間を考慮して、適切なデータフローを保証します（「[応答時間](#)  457」を参照）。

PAS4000では、コンフィグレーション可能なサイクル時間（例：外部通信を使用するサイクル時間）がシステムで自動的に計算される場合、プロジェクトに最適な条件や応答時間がシステムによって決定されます。整合性の範囲では、データの整合性が保証されます。

ユーザがコンフィグレーション可能なサイクル時間をコンフィグレーションした場合、コンフィグレーションされた条件に対する最適な応答時間がシステムによって決定されます。自動的に決定される応答時間と比べて悪くなる可能性があります。整合性の範囲では、データの整合性が保証されます。

10.7 ブロックの安全ステータス

IEC 61131プログラミングでは、定義済みの安全状態があるファンクションブロックタイプ／ファンクションタイプPOUをプログラミングすることができます。Multiプログラミングでは、安全状態があるブロックをプログラミングできません。ただし、Multiプログラミングでは、IEC 61131でプログラミングされたPOUは、基本ブロック (IEC 61131基本ブロック) として使用されます。これらの基本ブロックには、安全状態があります。

非安全関連タスクは、STブロックでプログラミングできます。STブロックは、STリソースまたはFSリソースで実行できます。PAS4000では、STブロックを緑色で示します。

安全関連タスクは、FSブロックでプログラミングする必要があります。FSブロックは、FSリソース上でのみ実行できます。PAS4000では、FSブロックを黄色で示します。

IEC 61131プログラミングでは、FSデータ型変数が宣言されるか、FS-POUが呼び出される／インスタンス化されると、ただちにPOUがFS-POUになります。

FSブロック内で非安全関連データ (STデータ) にアクセスできる場合、ブロックは「混合入力インタフェースがあるFSブロック」になります。PAS4000では、これらのブロックに特別なマーキングが付けられます。

- ▶ Multiプログラミング: 緑色と黄色が半々
- ▶ IEC 61131プログラミング: 白と黄色が半々

この特別なマーキングは、潜在的な危険としてブロックを識別することが目的です。

IEC 61131プログラミングでは、次の状況で、FS-POUが混合入力インタフェース付きのFS-POUになります。

- ▶ FSデータ型ではない入力PI変数の宣言
- ▶ FSデータ型ではない、タイプがVAR_INPUT、VAR_IN_OUT、またはVAR_EXTERNALである変数の宣言
- ▶ FSデータ型ではない、入力PI変数またはVAR_EXTERNALタイプの変数を宣言するPOUの呼び出し／インスタンス化

混合入力インタフェース付きのFSブロックを使用する場合は、次の点に注意してください。

- ▶ STリソースのデータは**安全ではありません**。これは、安全に関するアクション (開始／停止など) の追加基準にすぎず、安全機能を**妨げない**ようにする必要があります。たとえば、何らかの理由でSTリソースの停止信号が有効化されていない場合、または開始信号が継続して有効化されている場合、マシンまたはプロセスの安全が損なわれないようにする必要があります。
- ▶ 安全を確認するために、次のような方法で適切な手順を確立する必要があります。
 - 個別のリスク分析と
 - 適切な試運転テスト、実行可能性テスト
- ▶ 最初のインスタンスでは、混合インタフェース付きのFS-POUを完全にテストすることをお勧めします。必要な場合は、認証機関 (例: BG、TÜV) のテストと認証を受けただうえで、PAS4000を使用して保護してください。
- ▶ 正しい実装はユーザの責任において行ってください。



重要

混合入力インターフェースを持つFS-POUは、保護されると、潜在的危険（白と黄色が半々）の表示ではなくなり、黄色で表示されます。ブロック内部の安全に関わらないデータ（STデータ）へのアクセスが安全上重要でないアクセスであることは、製造者または認証機関の責任において保証する必要があります。

10.8 PAS4000による装置命名

装置の命名は、PAS4000のユーザによって実行されます。プロジェクト内で装置を一意に指定するために必要なデータは、プロジェクト内の各装置に対して定義されています。このデータを**命名データ**  309]と

装置の命名中にプロジェクト内の装置へ命名データを転送できるようにするには、装置がネットワーク上で識別可能でなければなりません。これには、工場設定として各装置に格納された固定データを使用します。このデータを、**装置識別データ**  308]といいます。

装置の命名中に区別が行われる装置タイプ:

▶ **PSS 4000専用装置** 310]

PSS 4000専用装置は、すべての装置識別データとすべての命名データをサポートします。

▶ **PSS 4000対応装置** 315]

PSS 4000対応装置は、装置識別データと命名データの一部をサポートします。



情報

ネットワークがルータを通じて接続されている場合、システム変数
PILZ_SNP_MULTICAST_TTLがそれに応じてコンフィグレーションされている場合にのみ、
装置の命名を集中的に実行できます (「**ルータを使用するネットワーク**  118]」を参
照)。

10.8.1 装置の識別データ

PSS 4000専用装置: PSSuシステム、PSS67装置

装置識別データ	意味	例
製品型式	装置の種類 (PSSuシステムのヘッドモジュールなど)	PSSu H PLC1 FS SN SD
型番	ピルツに装置を注文するための6桁の番号 (PSSuシステムのヘッドモジュールなど)	312070
シリアル番号	ピルツで装置を識別するための6桁の番号 (PSSuシステムのヘッドモジュールなど)	100076

ピルツのPSS 4000対応装置: オペレータターミナルPMI

装置識別データ	意味	例
製品型式	装置の種類	PMI v812
注文番号	ピルツに製品を注文するために使用する6桁の番号	266812
シリアル番号	ピルツで装置を識別するための6桁の番号	002020

サードパーティのPSS 4000対応装置 (PCなど)PC

装置識別データ	意味
製品型式	装置の種類: PSS 4000-PC
型番	型番には「0」を使用
シリアル番号	PSS 4000ファームウェアを最初に起動すると、ランダムな6桁の番号が生成され、その後さまざまな識別処理でこの番号をシリアル番号として内部で使用

10.8.2 命名データ

PSS 4000専用装置: PSSuシステム、PSS67装置

命名データ	既定の設定	意味
装置名	接頭語 (PSSなど)、6桁の型番、6桁のシリアル番号の組み合わせ フォーマット: <接頭語>_<型番>_<シリアル番号> 例: PSS_312070_100076	▶ プロジェクト内部で装置を一意に識別 ▶ プロジェクトを直接ダウンロードする際に、ネットワーク上で装置を識別するために使用 ▶ アドレスの一部を形成 (I/Oマッピングなど)
IPアドレス	Ethernet規格に適合するIPアドレス (「 Ethernet/パラメータの出荷時の既定の設定 」  618]を参照)	シリアル番号と組み合わせて使用し、装置の命名中にネットワーク上で装置を識別
サブネットマスク	Ethernet規格に適合する設定 (「 Ethernet/パラメータの出荷時の既定の設定 」  618]を参照)	Ethernet規格に適合
ゲートウェイアドレス	Ethernet規格に適合する設定 (「 Ethernet/パラメータの出荷時の既定の設定 」  618]を参照)	▶ Ethernet規格に適合 ▶ オプションとして使用
常用DNSサーバのアドレス		
代用DNSサーバのアドレス		

PSS 4000対応装置: オペレータターミナルPMI v8、PC

命名データ	意味、内容、既定の設定
装置名	▶ プロジェクト内部で装置を一意に識別 ▶ プロジェクトを直接ダウンロードする際に、ネットワーク上で装置を識別するために使用 既定の装置名は、Windowsオペレーティングシステムの設定中にユーザが定義したコンピュータ名またはホスト名に対応します。
IPアドレス	装置の命名の際、生成されたシリアル番号とともに使用してネットワーク上の装置を識別します。 Ethernet規格に適合するIPアドレス
サブネットマスク	Ethernet規格に適合するサブネットマスク
ゲートウェイアドレス	オプションとして使用 (装置により異なる)
常用DNSサーバのアドレス	オプションとして使用 (装置により異なる)
代用DNSサーバのアドレス	

10.8.3 PSS 4000専用装置の装置命名

PSS 4000専用装置での装置の命名で、区別可能なケース:

▶ **工場出荷時の状態またはオリジナルリセット後のPSS 4000専用装置**

工場設定の既定値は、工場出荷時の状態またはオリジナルリセットが行われた装置に格納されます (「[命名データ](#)」[\[309\]](#)」を参照)。Auto-IPメカニズムが有効化されます (「[Auto-IPメカニズム](#)」[\[82\]](#)」を参照)。装置プロジェクトがありません。

▶ **装置名の変更**

命名プロセスをすでに経ているPSS 4000専用装置の装置名は、遡って変更できます。この場合、ユーザ固有の命名データがすでに装置上にあります。装置プロジェクトが存在している場合もあります。ただし、装置の命名中に削除されます。

Ethernetパラメータも変更されるかどうかは、装置の命名手順とは無関係です。

▶ **Ethernetパラメータの変更**

命名プロセスをすでに経ているPSS 4000専用装置のEthernetパラメータは、遡って変更できます。この場合、ユーザ固有の命名データがすでに装置上にあります。装置プロジェクトが存在している場合もあります。これは装置の名前指定時に保持されます。

変更されるEthernetパラメータが1つだけであるか複数であるかは、装置の命名手順とは無関係です。通常は、IPアドレスのみが変更されます。ただし、装置名も変更された場合の処理は「装置名の変更」で述べたようになります。



情報

以下の説明では、準備作業がすべて行われていることを前提としています。例:

- SDカードがPSS 4000専用装置に挿入されていること。
- 関連する命名データがPAS4000で編集されていること。

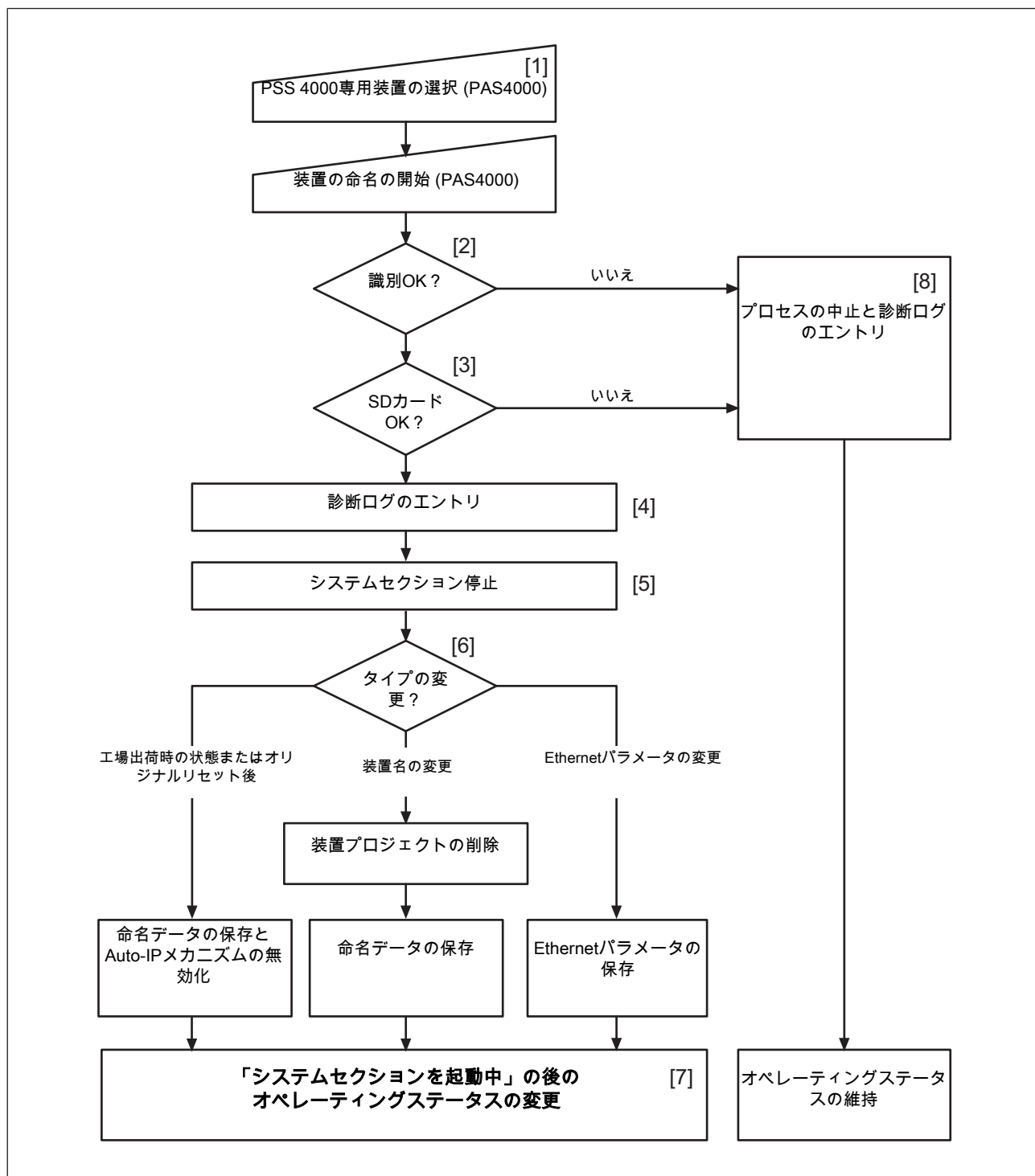


図: PSS 4000専用装置での装置命名フローチャート

PSS 4000専用装置での装置命名手順

以下の説明は、フローチャート「PSS 4000専用装置での装置命名フローチャート」の図を参照しています。

[1]

装置の命名を実行する装置が、PAS4000で選択されています。PAS4000でテストを実行し、選択した装置を現場で識別できます。このテストを「装置識別」といいます。装置の識別中、選択した装置ではSDカードのLEDが有効化されます。ステータスLED (装置の [SD CARD] LEDなど) がオレンジ色に点滅します。

[2]

いったん装置の命名を開始すると、装置は、ローカルのシリアル番号とIPアドレスが、PAS4000の命名データ転送の宛先として指定したシリアル番号およびIPアドレスと一致しているか確認します。

[3]

装置はSDカードがアクセス可能かどうか、すなわちSDカードが挿入されていること、故障していないことを確認します。

[4]

診断ログ内のエントリは、装置の命名が行われたことを示します。

[5]

装置のシステムセクションは停止しています。PSS 4000専用装置では、これはオペレーティングステータス「装置がSTOP状態、エラーなし」に対応しています。

[6]

その他の処理は、どの命名データが変更されたかに応じて変わります。

▶ **工場出荷時の状態またはオリジナルリセット後のPSS 4000専用装置**

装置のメモリとSDカードにある工場設定の既定命名データは、コンフィグレーション済みの命名データに置き換えられます。Auto-IPメカニズムは無効化されます。

▶ **装置名の変更**

既存の命名データは、装置メモリとSDカードから削除されます。SDカードに装置プロジェクトがすでに含まれている場合、その装置プロジェクトはSDカードから削除されます。FSリソースのある装置では、装置プロジェクトのFSチェックサムとプロジェクト名も、装置メモリから削除されます。その後、SDカードに新しい命名データが格納されます。

▶ **Ethernetパラメータの変更**

既存の命名データと既存の装置プロジェクトは保持されます。IPアドレスとその他すべてのEthernetパラメータは、装置メモリとSDカードから削除されます。その後、新しいIPアドレスとその他すべてのEthernetパラメータがSDカードに格納されます。

[7]

装置は、自動的に**オペレーティングステータス「システムセクションを起動中」**  [336](#)に切り替わります。

その他の処理は、どの命名データが変更されたかに応じて変わります。

▶ **工場出荷時の状態のPSS 4000専用装置と装置名の変更**

オペレーティングステータスが「システムセクションを起動中」の間、その効果の1つとして、新しい命名データがSDカードから装置メモリにロードされます。装置は装置プロジェクトなしで起動されます。SDカードのステータスLED (装置の [SD CARD] LEDなど) が緑色に点滅します。

装置が装置プロジェクトを取得できるように、プロジェクトをダウンロードする必要があります (**プロジェクトの直接ダウンロード**  [319](#)) および **プロジェクトの間接ダウンロード**  [325](#)を参照)。

▶ **Ethernetパラメータの変更**

オペレーティングステータスが「システムセクションを起動中」の間、その効果の1つとして、新しいEthernetパラメータがSDカードから装置メモリにロードされます。装置プロジェクトを使用できるようになるとすぐに、装置が装置プロジェクトとともに起動されます。

ネットワークの他の装置がこの装置とデータを交換する場合、新しいプロジェクトのダウンロードは、IPアドレスや他のEthernetパラメータの変更後に実行する必要があります。そうしなければ、他の装置はこの装置にアクセスできなくなります。

[8]

いずれかのチェックが不合格になると、処理が中止されます。エントリが診断リストに追加され、装置のオペレーティングステータスは変更されないままになります。

10.8.3.1 装置命名中のエラー動作

装置の命名中にエラーが発生した場合、一般に適用されるエラー動作:

- ▶ 装置命名処理の中止
- ▶ PSS 4000専用装置のすべてのシステムセクションを安全状態に切り替え
- ▶ PSS 4000専用装置のFS出力をすべて安全状態に切り替え
- ▶ 診断リストにエントリを追加

PSS 4000専用装置でコールドスタート／再起動コマンドを実行する必要があります。この場合、PSS 4000専用装置は「古い」命名データと「古い」装置プロジェクト (使用可能な場合) で起動されます。

10.8.4 PSS 4000対応装置の装置命名

PSS 4000対応装置の装置命名を行う際の注意事項:

▶ 装置名

PAS4000の装置命名中に編集された装置名は、PAS4000の診断リストと診断ログにのみ表示されます。装置の命名中、オペレーティングシステムのレベルで割り付けられた装置名 (コンピュータ名) は上書きされません。

▶ IPアドレス

装置のオペレーティングシステムのレベルでのIPアドレスは、装置命名によって上書きされません。PSS 4000対応装置のIPアドレスについては、次の点に注意してください。

- PAS4000でのハードウェアコンフィグレーションおよび装置の命名中に装置のIPアドレスを定義して、PSS 4000プロジェクト内の他の装置がこの装置と通信できるようにしておく必要があります。
- PAS4000でのハードウェアコンフィグレーションおよび装置命名中に定義される装置のIPアドレスは、装置のオペレーティングシステムレベルでコンフィグレーションされたIPアドレスと一致している必要があります。
- IPアドレスを変更する場合は、装置のオペレーティングシステムレベルで変更する必要があります (Microsoft Windows 10: Windowsの検索フィールドに「ネットワーク接続」と入力し、**[ネットワーク接続の表示]**を選択します。PSS 4000プロジェクトへの接続を確立するために使用されているアダプタを右クリックし、**[プロパティ]**を選択します。**[ネットワーク]**タブで **[インターネットプロトコルバージョン4 (TCP/IPv4)]**を選択し、**[プロパティ]**をクリックします。
- PSS 4000プロジェクトで、PSS 4000対応装置のオペレーティングシステムレベルのIPアドレスを変更する必要がある場合、そのPSS 4000プロジェクトに属していない他の装置は、既存のEthernet通信関係を通じて、PSS 4000対応装置にアクセスすることはできなくなります。このEthernet通信関係に対するIPアドレスは、関連装置で適応させる必要があります。



重要

プロジェクトで、PCがPAS4000-PC (PAS4000がインストールされたPC) とPSS 4000-PC (OPC サーバが動作するPC) の両方を兼ねている場合、プログラミングインタフェース (PAS4000-PC) のIPアドレスをPSS 4000-PCのIPアドレスとしても使用することになります。

PSS 4000対応装置での装置命名で、区別可能なケース:

▶ 初期装置命名中のPSS 4000対応装置

PSS 4000ファームウェアをインストールした後、工場設定の命名データが格納されます (**[命名データ]**  309)を参照)。装置プロジェクトがありません。

▶ 装置名の変更

装置名をさかのぼって変更する場合、PSS 4000対応装置には、ユーザ指定の命名データがすでに存在します。装置プロジェクトが存在している場合もあります。ただし、装置の命名中に削除されます。

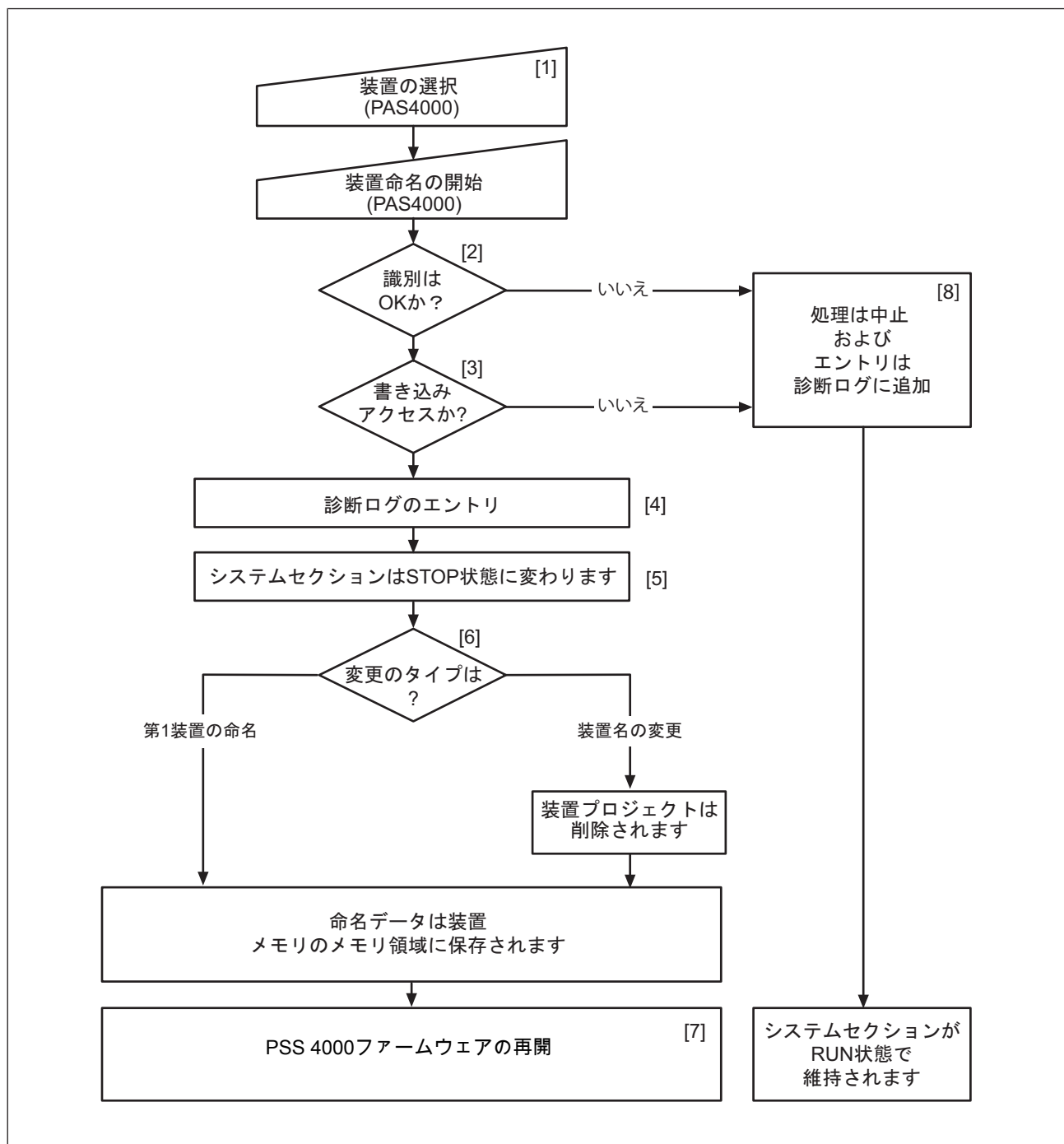


図: PSS 4000対応装置での装置命名フローチャート

PSS 4000対応装置での装置命名手順

以下の説明は、フローチャート「PSS 4000対応装置での装置命名フローチャート」の図を参照しています。

[1]

装置命名を実行するPSS 4000対応装置が、PAS4000で選択されています。PAS4000でテストを実行し、選択した装置を識別できます。このテストを「装置識別」といいます。装置識別中、選択した装置でポップアップウィンドウが開き、次のような内容が表示されます。

PSS 4000装置名: <装置名>

ポップアップウィンドウは、装置識別を明示的に終了するまで消えません。

[2]

いったん装置の命名を開始すると、装置は、ローカルのシリアル番号とIPアドレスが、PAS4000の命名データ転送の宛先として指定したシリアル番号およびIPアドレスと一致しているか確認します。

[3]

装置のセキュリティ設定で、関連するメモリ領域への「書き込み」権限があるか確認するテストが行われます。

[4]

診断ログ内のエントリは、装置の命名が行われたことを示します。

[5]

PSS 4000ファームウェアのシステムセクションで装置が停止します (OPCサーバ、診断サーバなど)。

[6]

その他の処理は、どの命名データが変更されたかに応じて変わります。

▶ **初期装置命名中のPSS 4000対応装置**

工場設定の命名データは、装置の関連するメモリ領域から削除されます。その後、メモリ領域に新しい命名データが格納されます。次に、装置に装置プロジェクトを保存するために、プロジェクトをダウンロードする必要があります。

▶ **装置名の変更**

既存の命名データと、すべての装置プロジェクトは、装置の関連メモリ領域から削除されます。その後、メモリ領域に新しい命名データが格納されます。次に、装置に装置プロジェクトを保存するために、プロジェクトをダウンロードする必要があります。

[7]

装置の命名後、命名データを転送するために、PCまたはオペレータターミナルPMIのPSS 4000ファームウェアを手動で再開する必要があります。

[8]

いずれかのチェックが不合格になると、処理が中止されます。診断リストにエントリが追加され、PSS 4000ファームウェアのシステムセクション (OPCサーバ、診断サーバなど) はRUN状態のままになります。

10.9 PSS 4000専用装置を交換する際の装置の命名

PSS 4000専用装置を交換する場合、PAS4000で新規装置の命名手順を厳密に実行する必要はありません。既存の装置の装置命名データと装置プロジェクトは、交換する装置の命名データと装置プロジェクトを含む既存のSDカードを再度使用することで、交換する装置に転送できます。



情報

PSS 4000専用装置を安全関連アプリケーションに使用している場合、装置を交換するには、「PSS 4000安全マニュアル」に記載されている変更に関する指示を必ず遵守していただかなければなりません。

前提条件

交換する装置は、工場出荷時の状態のままであるか、交換の前に装置でオリジナルリセットを実行している必要があります。

手順

1. 既存の装置をSTOP状態に切り替え、電圧供給コネクタを取り外します。
2. 既存の装置を取り外します。
3. SDカードを既存の装置から取り外し、交換する装置に挿入します。
4. 交換する装置の取り付け
5. 電圧供給コネクタを交換するユニットに差し込む
6. 以下の手順に従い、装置の命名データと装置プロジェクトを転送:
 - 交換する装置のリセットボタンを5秒以上押します。
 - 装置命名のための作業者の意図的なアクションの実行:
リセットボタンを放す - リセットボタンを押す - リセットボタンを放す
作業者の意図的なアクションは10秒以内に実行してください。そうしなければ転送は中止されます。

命名データと装置プロジェクトが転送されると、交換する装置は再起動されます。動作はコールドスタート／再起動コマンドの場合と同じです。

10.10 プロジェクトのダウンロード

10.10.1 プロジェクトの直接ダウンロード

プロジェクトの直接ダウンロード中、プロジェクトは装置プロジェクトに分割されてPSS 4000専用装置にダウンロードされ、可能な場合にはプロジェクトのPSS 4000対応装置にダウンロードされます。

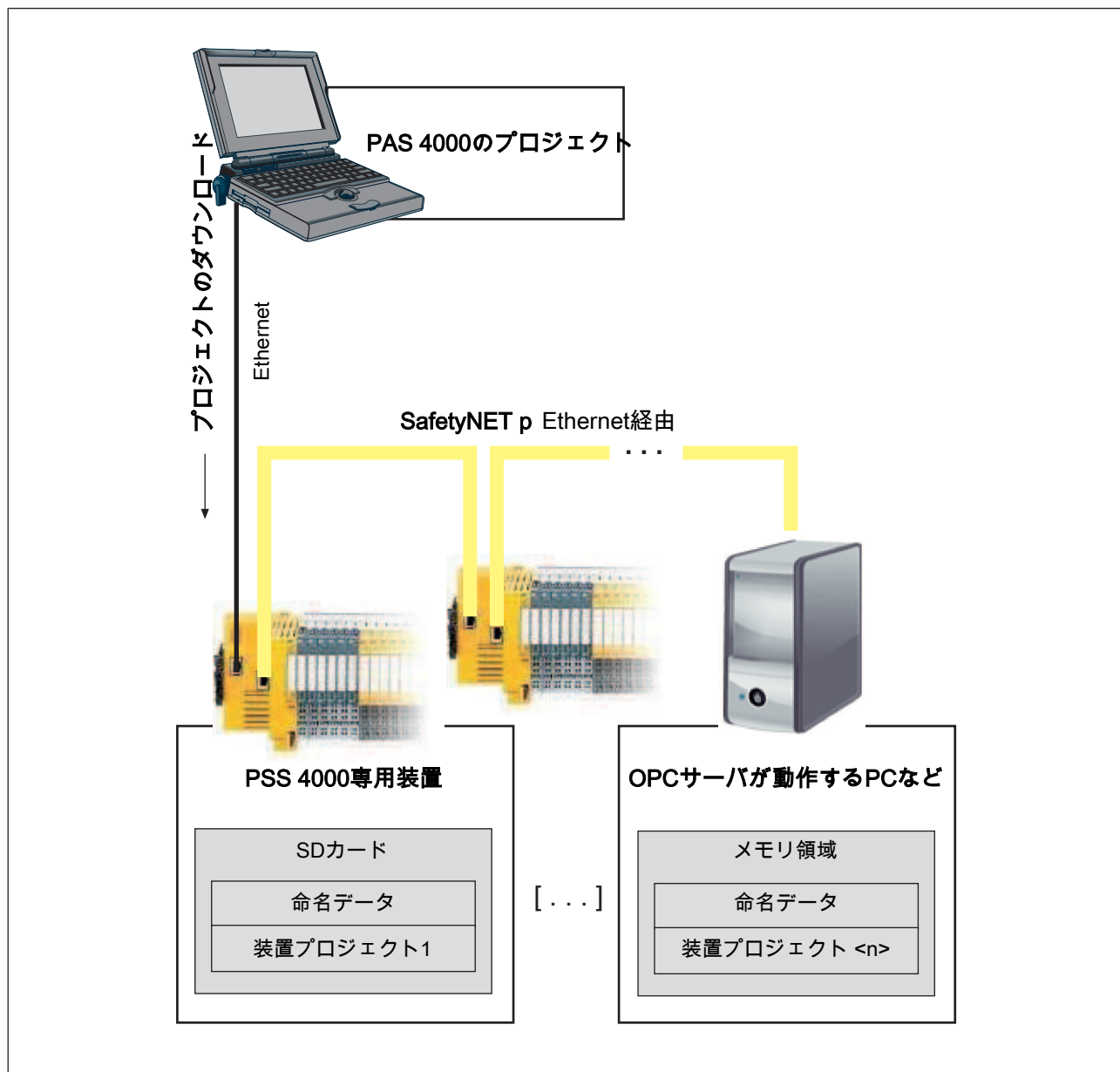


図: プロジェクトの直接ダウンロードの原則 (例)

プロジェクトのダウンロード中に、プロジェクトのバックアップコピーをプロジェクトの装置にダウンロードすることもできます。バックアップコピーのダウンロードは、プロジェクトデータのダウンロードに影響しません。バックアップコピーのダウンロードに失敗しても、プロジェクトのダウンロードは引き続き実行されます。

前提条件

- ▶ PAS4000-PCと、プロジェクトのPSS 4000装置を含むネットワークの間に、物理的な接続 (Ethernetケーブル) があること (「PAS4000-PCと装置の接続の確立」を参照)

- ▶ PAS4000でプロジェクトが有効化されていること
- ▶ プロジェクトの各装置において、ハードウェアコンフィグレーションにコンフィグレーションされているPSS 4000ファームウェアバージョンが、装置の実際のPSS 4000ファームウェアバージョンと同等以下であること
- ▶ 装置の命名を実行しておくこと（「PAS4000における装置命名の実行」を参照）
- ▶ プロジェクトをビルドしておくこと（「プロジェクトのビルド」を参照）
- ▶ SDカードがあるプロジェクト装置では、SDカードが挿入されていること
- ▶ 関連するメモリ領域に対する書き込みアクセスがある
例: PCの関連メモリ領域のセキュリティ設定で「書き込み」権限が設定されています。

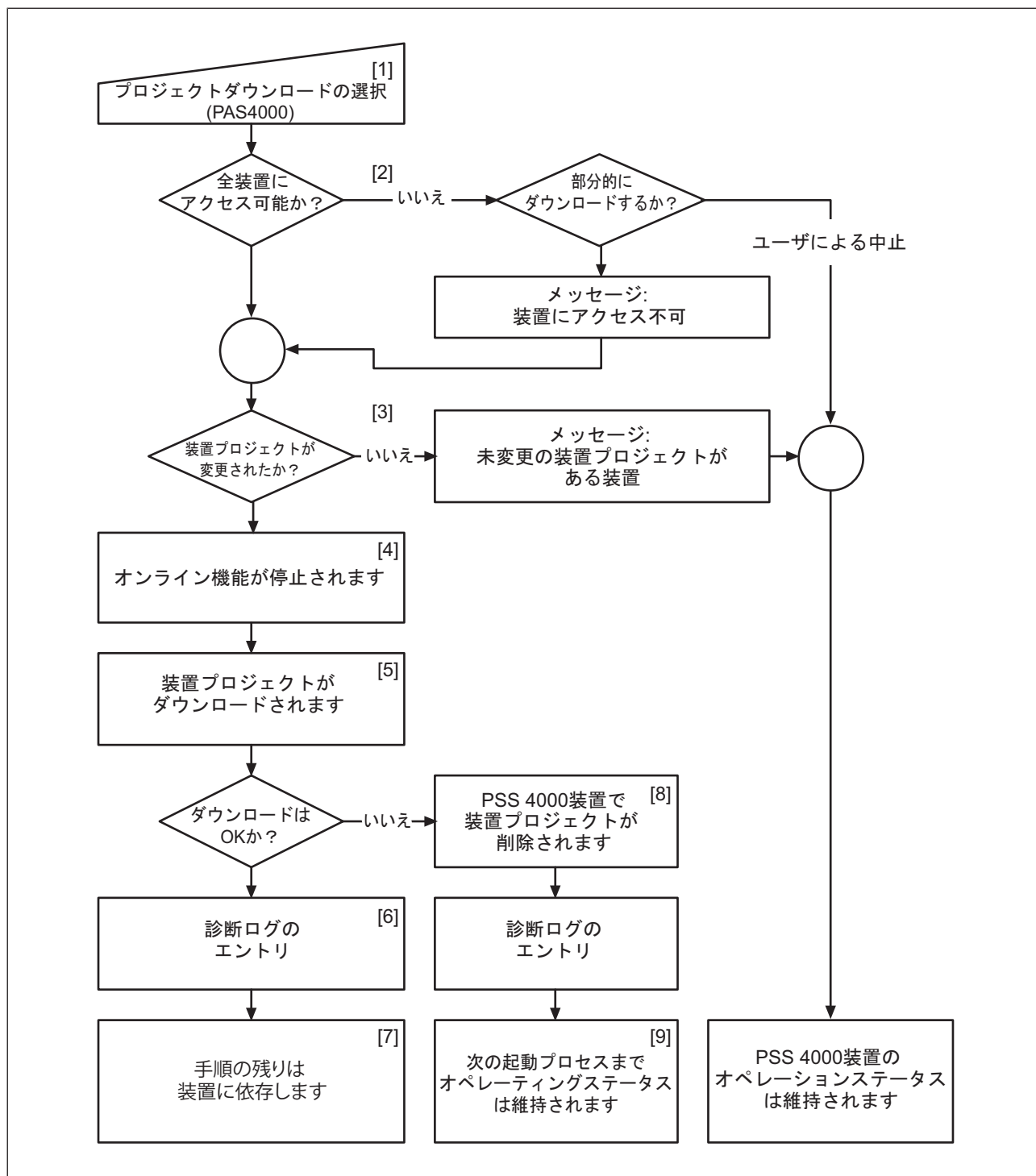


重要

いずれかのシステムセクションで内部エラーが検出された場合、PSS 4000装置への直接ダウンロードは実行されないで注意してください。内部エラーに対応して、影響を受けるシステムセクションのオペレーティングステータスが変わります。

- オペレーティングステータス: 重大なFSエラー
影響を受けるシステムセクション: 外部通信用システムセクションのFSリソース、FSモジュールバス、FS SafetyNET pおよび安全通信チャンネル (PROFIsafeシステムセクションを備えたPROFIBUS-DPスレーブのPROFIsafeなど):
- SafetyNET pを用いた通信のオペレーティングステータス: FSおよびSTの重大エラー
影響を受けるシステムセクション: FS SafetyNET pおよびST SafetyNET p:
- 外部通信でのオペレーティングステータス: FSおよびSTの重大エラー
(このオペレーティングステータスは、システムセクションに安全関連の通信チャンネルがある場合にのみ発生します。たとえばPROFIsafeシステムセクションを備えたPROFIBUS-DPスレーブのPROFIsafeなど)
影響を受けるシステムセクション: 外部通信用システムセクション (PROFIsafeを備えたPROFIBUS-DPスレーブなど)。

プロジェクトの直接ダウンロード手順



[1]

PAS4000で「プロジェクトの直接ダウンロード」を選択します。

[2]

PAS4000は、プロジェクトの装置がすべてアクセス可能であるか確認します (たとえば、装置のスイッチが入っているか、装置へのネットワーク接続ケーブルが差し込まれているかなど)。プロジェクトの装置の中にアクセスできないものがある場合、次のような選択肢があります:

- ▶ ユーザがプロジェクトの直接ダウンロードを中止する
全装置の現在のオペレーティング状態は維持されます。
- ▶ プロジェクトの直接ダウンロードを継続する
プロジェクトの直接ダウンロードに関してアクセス可能な全装置が選択されます。ダウンロードから除外された装置がPAS4000に表示されます。

[3]

PAS4000は各装置を確認して、既存の装置プロジェクトと新しい装置プロジェクトの間に違いがあるか調べます。

装置プロジェクトが変更されていない場合、装置はダウンロードから除外されます。装置プロジェクトが変更されていない装置はPAS4000に表示されます。これらの装置の現在のオペレーティングステータスは保持されます。

[4]

オンライン機能 (変数の強制や変数の監視など) は停止します。装置は現在のオペレーティングステータスのままです。

[5]

関連装置の装置プロジェクトをダウンロードします。

- ▶ PSS 4000専用装置

PSS 4000専用装置のSDカードに装置プロジェクトがすでに存在する場合、装置プロジェクトがSDカードから削除され、装置プロジェクトのFSチェックサムとプロジェクト名が装置メモリから削除されます。その後、新しい装置プロジェクトがSDカードに格納されます。

PSS 4000専用装置でオンライン変更が有効な場合、ダウンロードによってこれらオンライン変更は無効化されます。

- ▶ PSS 4000対応装置

使用可能な場合、PSS 4000対応装置で、まず装置プロジェクトが関連するメモリ領域から削除され、次に新しい装置プロジェクトがメモリ領域に格納されます。

[6]

プロジェクトの直接ダウンロードが成功すると、診断ログにエントリが作成されます。

[7]

装置プロジェクトが無事にダウンロードされた後の手順は、装置により異なります。

- ▶ PSS 4000専用装置

装置は自動的に再起動されます。

- ▶ PSS 4000対応装置

装置のPSS 4000ファームウェアは自動的に再起動されます。

PSS 4000専用装置を起動する際の注意事項:

- ▶ 起動の影響はウォームリセットの場合と同じです。
- ▶ プロジェクトのダウンロードより前に停止コマンドのいずれかを用いて停止させたPSS 4000専用装置は、自動的に起動されません。これらの装置は手動で再起動する必要があります。
- ▶ 旧装置プロジェクトのユーザプログラムに不揮発性および揮発性の変数が含まれている場合、装置は以下のような値で起動されます。

ケース	属性がRETAIN PERSISTENTの変数	揮発性 変数 [1]	アプリケーションパラメータ
ユーザプログラム内で、変数は変更されずに使用できます (リソース割り付け、FQN、データ型が同じです)。	保存済みの値	変数は初期値に設定される	アプリケーションパラメータに 設定された値
ユーザプログラム内で変数が新規になる	変数は初期値に設定される		
変数がユーザプログラム内で使用不可になる	変数は削除される		アプリケーションパラメータが 削除される

[1] 保持されない変数がアプリケーションパラメータとして使用されている場合、「アプリケーションパラメータ」列が適用されます。

注

プロジェクトが装置にダウンロードされたら、次の条件が適用されます。

- ▶ 変数が使用されるブロックのリソース割り付け
- ▶ 変数のFQN (PSS 4000名前空間での正式名称) および
- ▶ そのデータ型

これらの値が以前のダウンロードと比較して変更されていない場合、属性RETAIN PERSISTENTがある変数の内容は維持されます。

[8]

装置プロジェクトがPSS 4000装置にダウンロードされる際に、さまざまなエラーが発生することがあります。

例:

- ▶ ユーザによる中止
- ▶ タイムアウト
- ▶ 装置プロジェクトの保存中にエラーが発生する

影響

PSS 4000専用装置では、装置プロジェクトがSDカードから削除され、装置プロジェクトのプロジェクト名とFSチェックサムが装置メモリから削除されます。PSS 4000対応装置では、関連するメモリ領域から装置プロジェクトが削除されます。診断ログにエントリが追加されます。

処置

プロジェクトの再ダウンロード

[9]

装置のオペレーティングステータスは、次の起動プロセスまで保持されます。次にシステムを起動すると (コールドスタート／再起動コマンドの後など)、装置は装置プロジェクトなしで起動されます。



重要

プロジェクトの直接ダウンロード後に、プロジェクトの全体的なデータの整合性を保証する

プロジェクトのダウンロード後、プロジェクト装置のデータは自動的に整合性が維持されます。つまり、装置では、プロジェクトのダウンロードが中断または失敗した場合でも、装置プロジェクトのデータは常に整合性が保たれます。

プロジェクトのダウンロードが正常に実行され、すべてのプロジェクト装置にアクセスできる場合、PSS 4000プロジェクトのすべての装置の装置プロジェクトも相互に整合性があります。

プロジェクトのダウンロードが中断または失敗した場合、またはすべてのプロジェクト装置にアクセスできなかった場合に、すべての装置の装置プロジェクト間で相互に整合性が保たれるようにすることはユーザの責任です。

この場合は、次の手順に従います。

- プロジェクトのすべての装置で同じプロジェクトステータスがコミットされていることを確認します。
これを行うには、PAS4000のオンラインネットワークエディタで各プロジェクト装置のオンライン情報を表示します。プロジェクト情報「プロジェクト名」と「ビルド年月日」がすべてのプロジェクト装置で一致していることを確認します。
- 装置のプロジェクト情報が一致しない場合は、プロジェクトを再びダウンロードします。プロジェクトのすべての装置にアクセスできることを確認してから、プロジェクトを再度ダウンロードします。

10.10.2 プロジェクトの間接ダウンロード

プロジェクトの間接ダウンロードでは、プロジェクトは装置プロジェクトに分割されます。装置プロジェクトは、命名データとともにSDカードまたはディレクトリに保存されます。間接ダウンロードを完了するには、ユーザは、SDカードを装置に挿入し、オペレータが意図的な手動操作でプロジェクトを装置に転送する必要があります。

これにより、PAS4000と装置を接続しなくても、データを操作できるようになります。



情報

間接ダウンロードはPMIには使用できません。PMIは、プロジェクトの直接ダウンロードでのみ装置プロジェクトを取得できます。

2つのオプションがあります。

▶ SDカードでのプロジェクトの間接ダウンロードの実行

装置プロジェクトおよび対応する命名データは、SDカードに保存され、後で装置に挿入してプロジェクトを実行できます。これにより、PAS4000と装置を接続しなくても、SDカードを操作できるようになります。この手順は、特にPSS 4000専用装置 (PSSuシステムなど) で役立ちます。PC用にSDカードを作成することもできますが、これらのSDカードを直接使用することはできません (データをSDカードからPCに手動でコピーする必要があります)。

▶ ディレクトリへのプロジェクトの間接ダウンロードの実行

すべての装置プロジェクトおよび対応する命名データは、ディレクトリに保存されます。ディレクトリは、ローカルドライブ、ネットワークドライブ、またはUSBスティックなどの大容量記憶装置上に置くことができます。そこからデータをSDカードにコピーし、SDカードを装置に挿入してプロジェクトを実行できます。PC装置プロジェクトをPCにコピーできます。

「ディレクトリへの間接ダウンロード」アクションを使用して、装置プロジェクトと、プロジェクトのすべてのPSS 4000専用装置 (PSSuシステムなど) の対応する命名データが格納されたSDカードを作成することもできます。いつでも特定の装置にSDカードが必要な場合は、関連するデータをSDカードに移動する必要があります。そうすれば適切なSDカードが利用可能になります。



重要

プロジェクトの間接ダウンロード後に、プロジェクトの全体的なデータの整合性を保証する

プロジェクトの間接ダウンロード後、プロジェクト装置のデータは自動的に整合性が維持されます。つまり、装置では、SDカードからの装置プロジェクトの転送に失敗した場合や、コピー時にファイルが損傷した場合でも、装置プロジェクトのデータは常に整合性が保たれます。

すべての装置の装置プロジェクトが互いに一貫していることを確認するのは、ユーザの責任です。

この場合は、次の手順に従います。

- プロジェクトのすべての装置で同じプロジェクトバージョンが転送されていることを確認します。
これを行うには、PAS4000のオンラインネットワークエディタで各プロジェクト装置のオンライン情報を表示します。プロジェクト情報の「プロジェクト名」と「ビルド年月日」がすべてのプロジェクト装置で一致していることを確認します。
PAS4000の代わりに、PSS 4000 OPCサーバUAを使用して「プロジェクト名」と「ビルド年月日」を調べることができます。
- 装置のプロジェクト情報が一致しない場合は、PAS4000でプロジェクトの間接ダウンロードを再度実行し、装置プロジェクトを装置に再度配布します。

10.11 シールされたFSブロック

保護されたブロックは、プログラミング補助機能です。保護されたブロックは、変更できず、操作から保護されています。

保護されたFSブロックプロセスは、頻繁に必要な安全関連タスク (非常停止など) を処理します。それらは、ユーザが作成するか、ビルツから入手できます。

認証済みの保護されたFSブロックを使用すると、必須の検査の対象になる機械で 사용되는場合でも、含まれるユーザプログラムの再テストが不要になるので、プロジェクトの認証処理の期間を短縮することができます。次のブロックが存在します。

▶ 保護されたユーザブロック

保護されたユーザブロックの追加の情報を保存できます。これには、次のような内容が含まれています。

- 保護されたブロックをテストした担当者の名前
- 会社名
- 第三者認証機関の名前
- 認証番号

▶ 保護されたビルツブロック

ビルツは、第三者認証機関 (BG、TÜV など) によってテストおよび認証されたさまざまなブロックを提供します。

第三者認証機関によって認証されたブロックを使用することをお勧めします。これらのブロックの想定される動作状態がユーザプログラムに必要な状態と一致していることを確認する必要があります。

11 オペレーティングモード

11.1 PAS4000とPSS 4000装置の間の通信

オンライン操作を実行するために、PAS4000がインストールされているコンピュータ (PAS4000-PC) はプロジェクトのPSS 4000装置 (PSS 4000専用装置および／またはPSS 4000対応装置) を含むネットワークに接続されている必要があります。

詳細については、PAS4000のオンラインヘルプを参照してください。

11.1.1 ポイントツーポイント接続の確立

ポイントツーポイント接続では、PAS4000は個々のPSS 4000専用／対応装置に加えて、ネットワークにあるすべてのPSS 4000専用／対応装置にアクセスできます。

原理

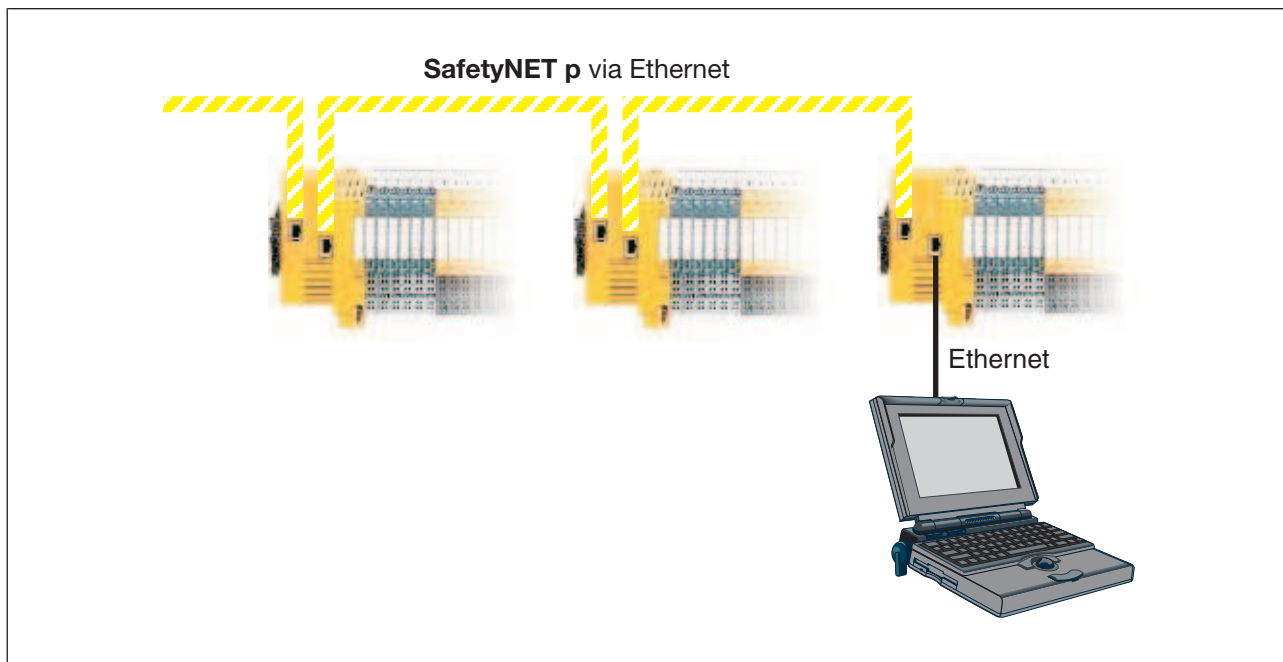


図: PAS4000でのポイントツーポイント接続の原理

「ルータのないネットワーク [📖 116]」の情報を参照してください。

11.1.2 リモート接続の確立

リモート接続では、PAS4000は個々のPSS 4000専用／対応装置に加えて、ネットワークにあるすべてのPSS 4000専用／対応装置にアクセスできます。



重要

データセキュリティリスク

診断などのためにリモート接続で通信する場合は、データセキュリティに関する特殊な機能について注意してください (『PSS 4000安全マニュアル』を参照)。

原理

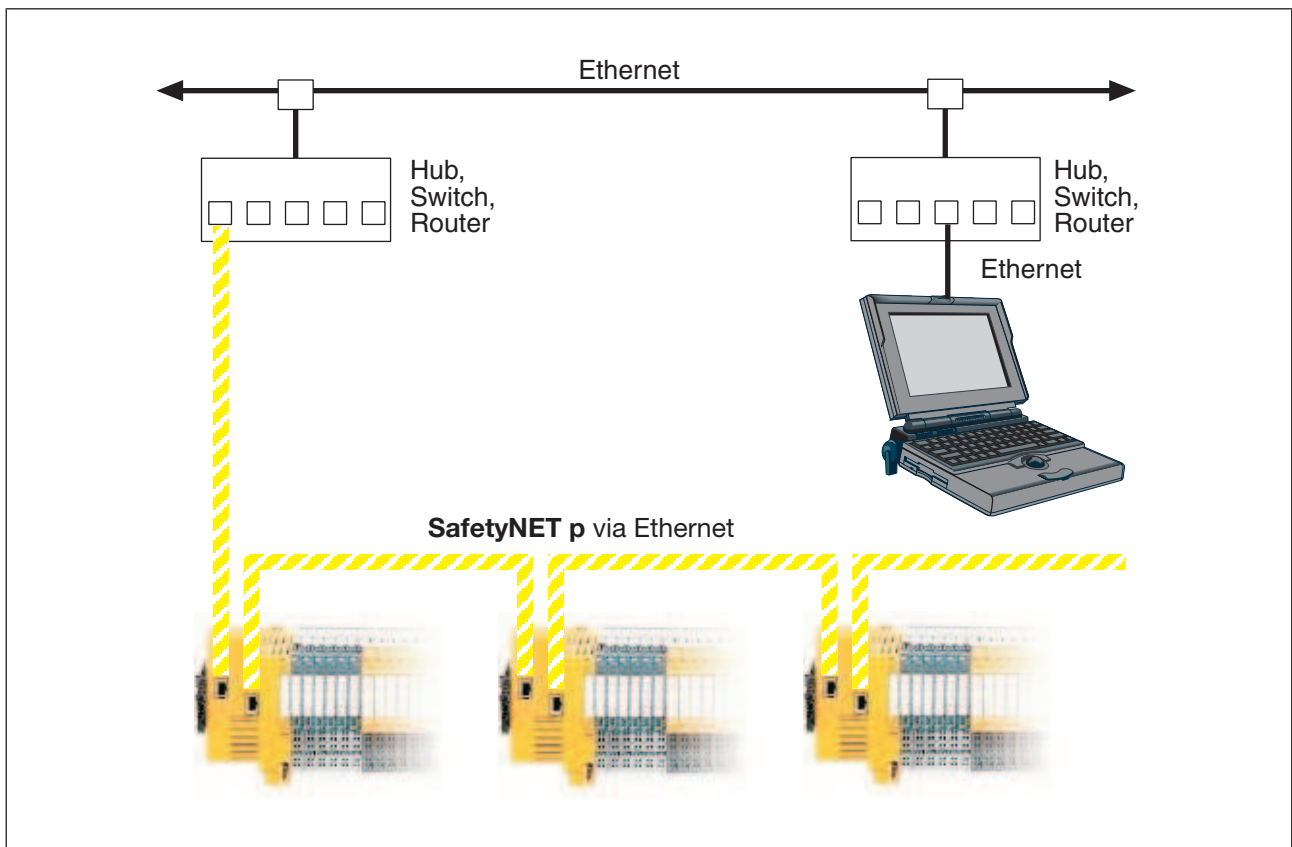


図: PAS4000でのリモート接続の原理

ルータを使用している場合は、制限事項に注意してください (「[ルータのあるネットワーク](#)」[\[118\]](#)」を参照)。

11.2 オペレーティングステータスと装置のオペレーティングステータスの変更

オペレーティングステータスやオペレーティングステータスの変更には、PSS 4000専用装置のすべてのシステムセクションに影響を与えるものもあれば、あるシステムセクションだけに影響が限定されるものもあります。PSS 4000専用装置にはシステムセクションのすべて、またはサブセットのみを含めることができます (「[システムセクション](#)」 22]」を参照):

以下の説明では、PSS 4000専用装置のすべてのシステムセクションに影響するオペレーティングステータスのみを扱います。

図「オペレーティングステータスとPSS 4000専用装置のオペレーティングステータスの変更」(「[フローチャート](#)」 332]」を参照) にオペレーティングステータス (グレー) の概要を示します。

- ▶ 角括弧内の数字はオペレーティングステータスのさまざまな変更を示します。
- ▶ ローマ数字「I」または「II」が付いたセクションは接続ポイントで、他の図を参照しています。
- ▶ ローマ数字「III」、「IV」、「V a/b」、「VI a/b」、「VII」が付いたセクションは、オペレーティングステータスとシステムセクションのオペレーティングステータスの変更を示す図を参照しています。

以下の説明では、PSS 4000専用装置に有効な装置プロジェクトがあることを前提としています

11.2.1 フローチャート

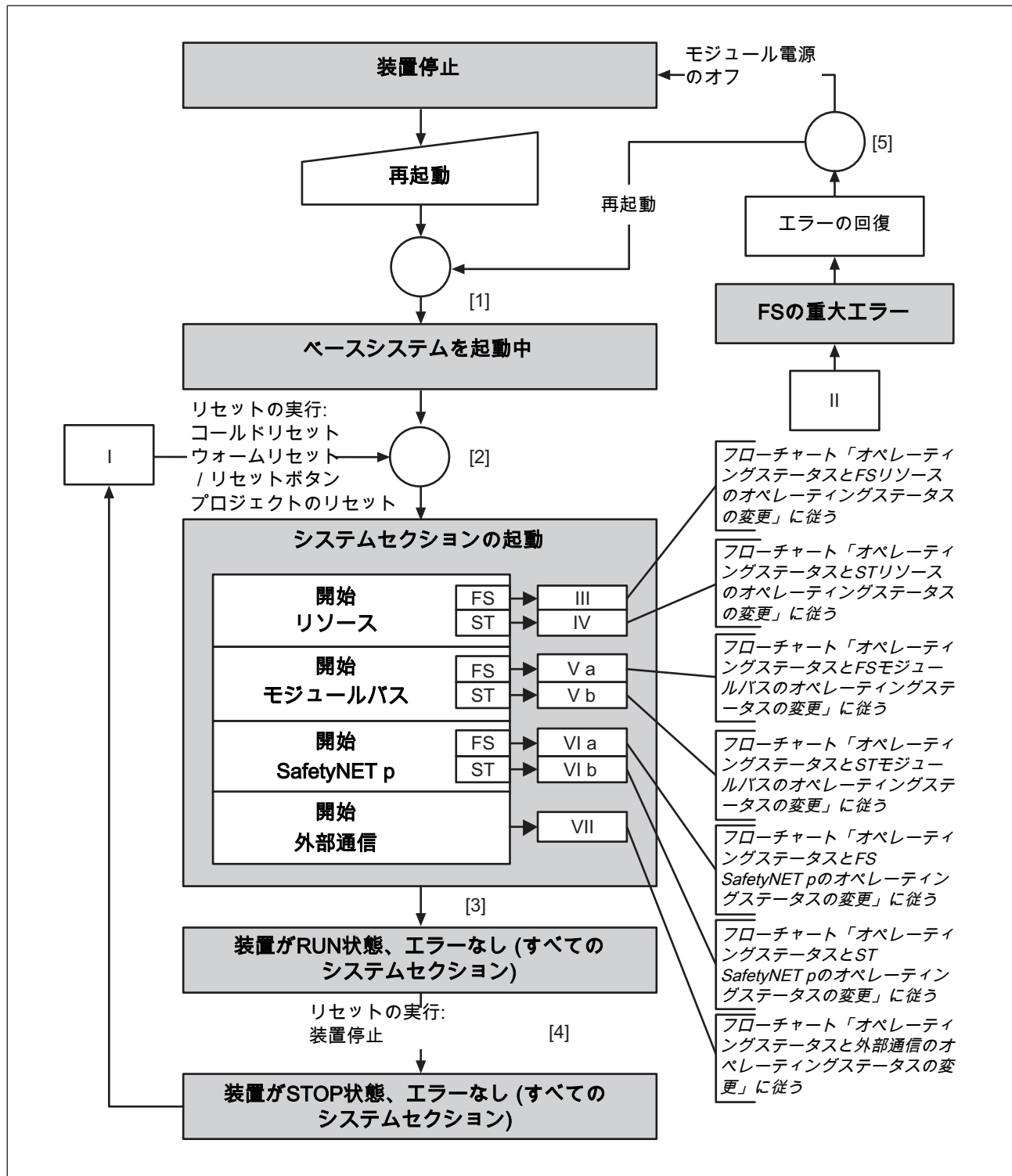


図: オペレーティングステータスとPSS 4000専用装置のオペレーティングステータスの変更

11.2.2 装置のオペレーティングステータス

PSS 4000専用装置は次のオペレーティングステータスに変更します (「フローチャート」を参照)。

- ▶ オペレーティングステータス「装置停止」
- ▶ オペレーティングステータス「ベースシステムを起動中」
- ▶ オペレーティングステータス「システムセクションを起動中」
- ▶ オペレーティングステータス「装置がRUN状態、エラーなし」
- ▶ オペレーティングステータス「装置がSTOP状態、エラーなし」
- ▶ オペレーティングステータス「FSの重大エラー」



情報

各オペレーティングステータスの説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「[リセット、再起動、起動、停止のオプション](#)」 424]を参照)。

11.2.2.1 装置停止

「装置停止」はPSS 4000専用装置に電源が入っていない状態です。モジュール電源の供給電圧をオンにすると (コールドスタート)、PSS 4000専用装置がオペレーティングステータス「ベースシステムを起動中」に切り替わります。

11.2.2.2 オペレーティングステータス「ベースシステムを起動中」

各種起動テスト (メモリテストなど) が実行されます。起動プロセス中にエラーがあると、PSS 4000専用装置は起動しません。装置の [DIAG] LEDは赤色に点滅します。

処置

- ▶ 技術データ (モジュール電源の供給電圧、周囲温度など) を確認します
- ▶ 装置の交換が必要な場合があります

11.2.2.3 オペレーティングステータス「システムセクションを起動中」

「システムセクションを起動中」は、次のいずれかのフェーズを示すステータスです。

- ▶ 基本システムの起動後に、PSS 4000専用装置の全システムセクションがRUN状態になるまで
- ▶ リセットコマンド (コールドリセット、リセットボタン／ウォームリセット) のトリガ後、PSS 4000専用装置の全システムセクションがRUN状態に戻るまで
- ▶ 基本システムの起動後、またはリセットコマンドのトリガ後、システムセクションがエラー状態になるまで
これらのエラー状態については、それぞれのシステムセクションで説明します。

テストフェーズ

テストフェーズは、システムセクションの実際の起動に先行します。このテストフェーズでは、ハードウェア自己診断が実行され、装置のファームウェアが検証されます。エラーが検出された場合、または装置のファームウェアが破損している場合、オペレーティングシステムは起動プロセスを停止します。

このような場合、ステータスLED「DIAG」が赤色に点滅し、診断機能が利用できなくなり、リセットボタンが機能しなくなり、すべてのオンライン操作ができなくなります。この状態を終了するには、電圧供給をOFFにする (装置を再起動する) 必要があります。

さまざまな自己診断がエラーなしで実行された場合は、SDカードの整合性がチェックされます。ジャーナリングファイルシステムのジャーナルをSDカードで実行できない場合、システムセクションはSDカードなしで起動されます (下のフローチャートの[1]を参照)。詳細については、「[ジャーナリングファイルシステム](#) [📖 478]」および「[SDカードの関連付け](#) [📖 478]」を参照してください。

テストフェーズ中にリセットボタンを押すと、テストとチェックの後に起動プロセスが停止します。装置がリカバリモードに切り替わります。詳細については、「[リカバリモード](#) [📖 448]」を参照してください。

実際の「システムセクションの起動」を開始します。

SDカードは、システムセクションの起動に使用する命名データと装置プロジェクトを決定するために使用されます。

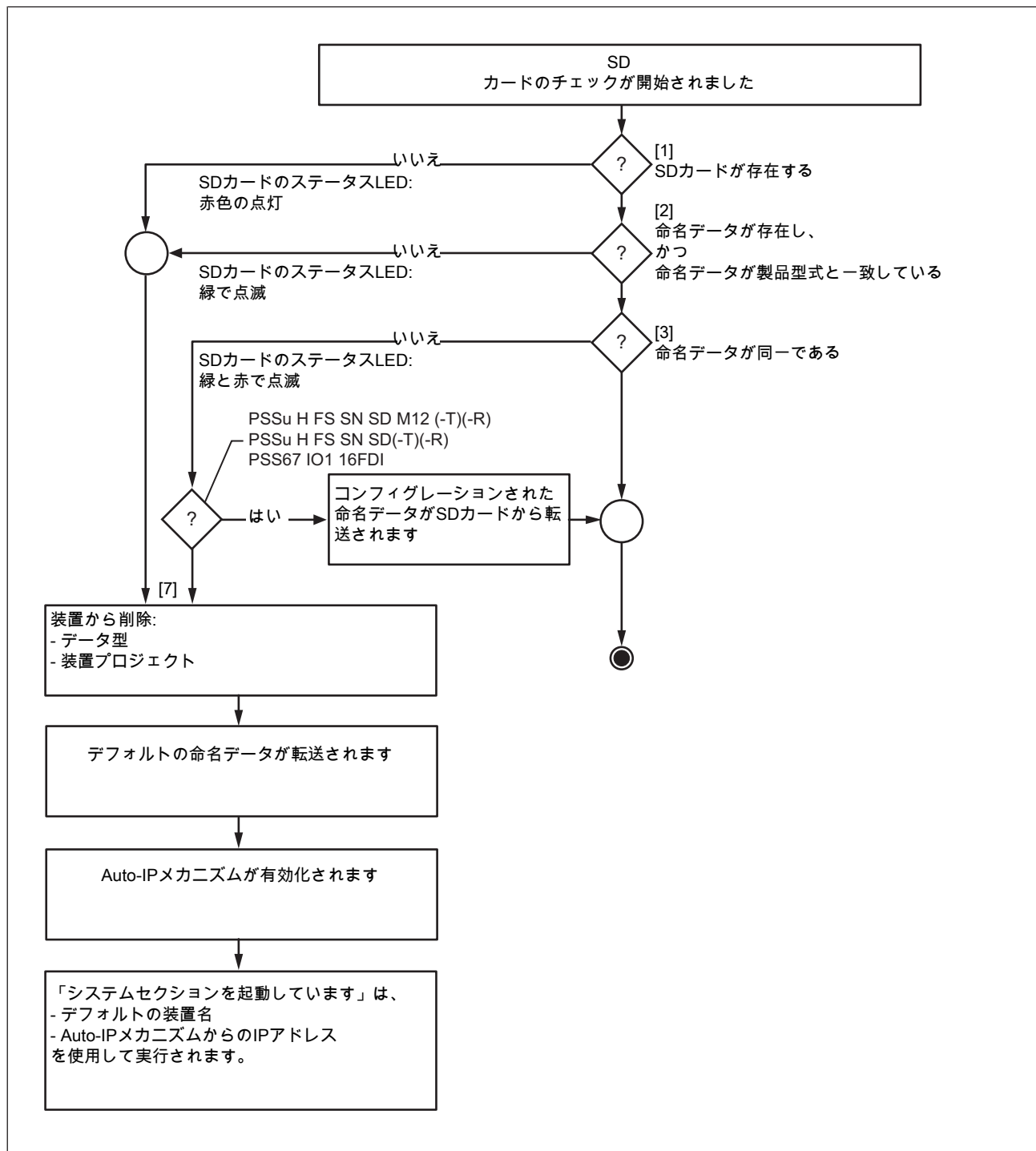


図: SDカードのチェックのフローチャート (パート1)

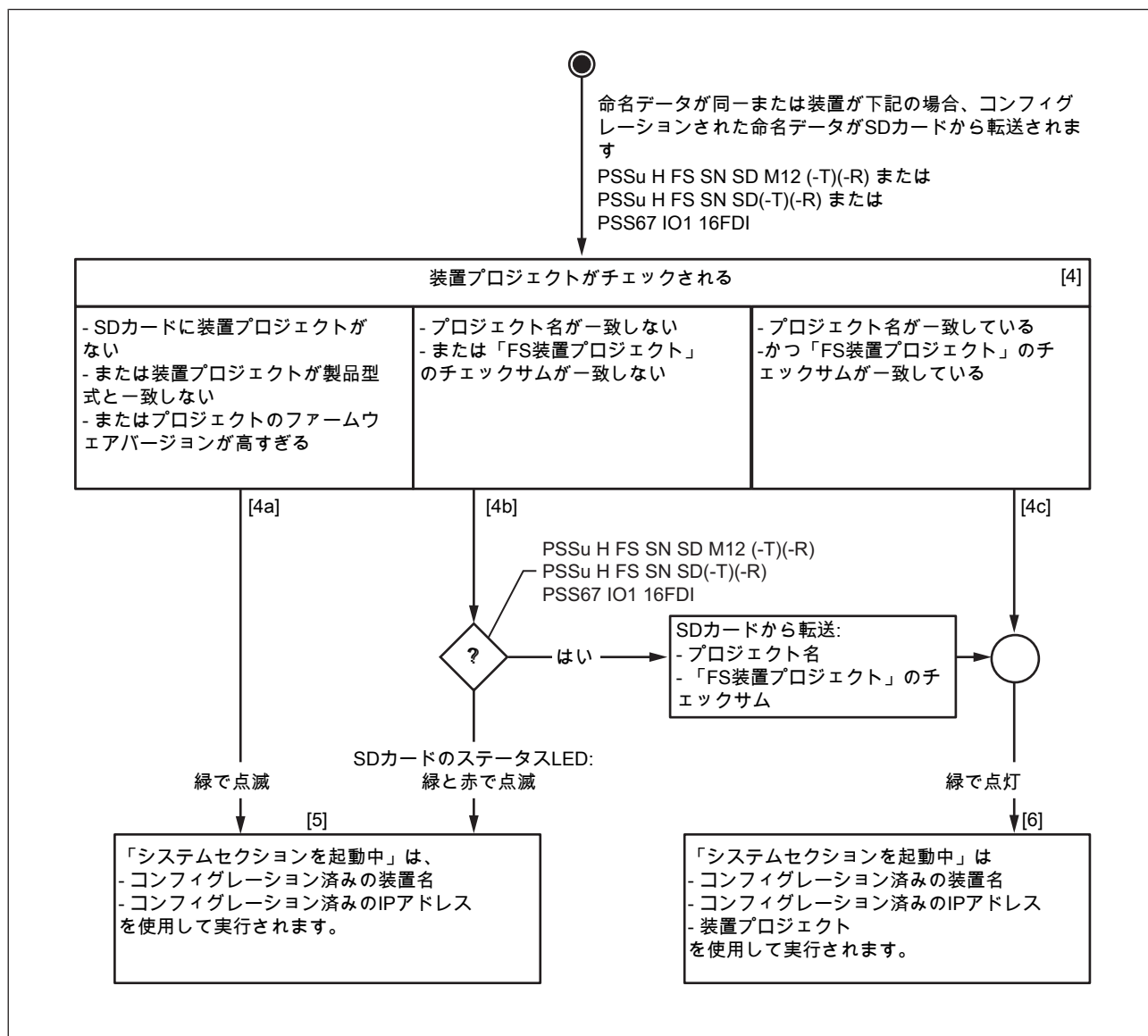


図: SDカードのチェックのフローチャート (パート2)

[1]

最初のテストでは、装置のSDカードにアクセスできるかどうかチェックされます。つまり、SDカードについて次の内容がチェックされます。

- ▶ 挿入済みか
- ▶ 不具合がないか
- ▶ データが空でないか

SDカードにアクセスできない場合、装置は装置プロジェクトなしで、工場出荷時のデフォルトの命名データで起動します ([7] を参照)。

[2]

テストでは、SDカードに命名データ (「命名データ」[\[309\]](#)) が含まれているかどうかと、SDカードにある命名データが装置の製品タイプに一致しているかどうか確認されます。

SDカードに命名データがない場合、または命名データが装置の製品タイプに一致していない場合、装置は装置プロジェクトなしで、工場出荷時のデフォルトの命名データで起動します ([7] を参照)。

[3]

テストでは、装置の命名データがSDカードの命名データと一致するかどうかを確認されます。ケースは次のように分類されます。

▶ 同一の命名データ:

装置プロジェクトは、次の手順に従ってテストされます ([4] を参照)。

▶ 次のいずれかの装置の同一でない命名データ:

- PSSu H FS SN SD(-T)(-R)
- PSSu H FS SN SD M12 (-T)(-R)
- PSS67 IO1 16FDI

装置の命名データは削除され、装置はSDカードの命名データを採用します。装置プロジェクトは、次の手順に従ってテストされます ([4] を参照)。

▶ 他のすべての装置の同一ではない命名データ:

装置は装置プロジェクトなしで、工場出荷時のデフォルトの命名データで起動します ([7] を参照)。

[4]

装置プロジェクトに対して、さまざまなテストが実行されます。その他の手順は、テスト結果により異なります。

▶ 次のうち、少なくとも1つの条件を満たす場合は、[4a] の手順に従います。

- SDカードに装置プロジェクトがない。
たとえば、この状態は装置の命名時に発生することがあります。
- SDカードの装置プロジェクトが装置の製品タイプと一致しない。
たとえば、この状態はSDカードを交換した場合に発生することがあります。
- 装置は、SDカードの装置プロジェクトに対して、さらに高いファームウェアバージョンを要求している。

これらのどの場合でも、装置はユーザがコンフィグレーションした命名データ (装置名、IPアドレス) で起動しますが、装置プロジェクトがない状態での起動になります ([5] を参照)。装置で装置プロジェクトを取得するには、プロジェクトの直接ダウンロードを実行するか ([「プロジェクトの直接ダウンロード」](#)  319) を参照)、「意図的なオペレータ操作による命名データ／装置プロジェクトの転送」を実行する必要があります ([「意図的なオペレータ操作」](#)  443) を参照)。

▶ 次のうち、少なくとも1つの条件を満たす場合は、[4b] の手順に従います。

- 装置のプロジェクト名がSDカードのプロジェクト名と一致しない。
たとえば、この状態はSDカードを交換した場合に発生することがあります。
- 装置が、PSSu H FS SN SD(-T)(-R)、PSSu H FS SN SD M12 (-T)(-R)、または PSS67 IO1 16FDIではなく、装置のチェックサム「FS装置プロジェクト」がSDカードのチェックサム「FS装置プロジェクト」と一致しない。
この状態はSDカードを交換した場合に発生することがあります。

装置はユーザがコンフィグレーションした命名データ (装置名、IPアドレス) で起動しますが、装置プロジェクトがない状態での起動になります ([5] を参照)。装置で装置プロジェクトを取得するには、プロジェクトの直接ダウンロードを実行するか ([「プロジェクトの直接ダウンロード」](#)  319) を参照)、「意図的なオペレータ操作による命名データ／装置プロジェクトの転送」を実行する必要があります ([「意図的なオペレータ操作」](#)  443) を参照)。

PSSu H FS SN SD(-T)(-R)、PSSu H FS SN SD M12 (-T)(-R)、またはPSS67 IO1 16FDIは、装置メモリからプロジェクト名とチェックサム「FS装置プロジェクト」を削除して、SDカードのプロジェクト名とチェックサム「FS装置プロジェクト」を採用します。装置はユーザがコンフィグレーションした命名データ (装置名、IPアドレス) とSDカードの装置プロジェクトで起動します ([6] を参照)。

- ▶ 次のうち、少なくとも1つの条件を満たす場合は、[4c] の手順に従います。

SDカードおよび装置で、プロジェクト名とチェックサム「FS装置プロジェクト」が一致している。

これは、装置が「正常な状態」になっていること、つまり、円滑に操作を実行するために必要な要件を装置がすべて満たしたことを表しています。たとえば、これらの要件には次のようなものがあります。

- SDカードにアクセスできない
- 装置命名データと装置プロジェクトがSDカードから装置に正常に転送された
- SDカードが正しく交換されていない

装置はユーザがコンフィグレーションした命名データ (装置名、IPアドレス) とSDカードの装置プロジェクトで起動します ([6] を参照)。



情報

起動すると、関連しないすべてのデータとディレクトリがSDカードから削除されます。つまり、次のように処理されます。

- ユーザによって「/userfiles」ディレクトリ以外に保存された、システム由来でないデータとディレクトリは、SDカードから削除されます。
- 次のディレクトリとファイルは、PSS 4000ファームウェアの機能に必要なディレクトリとともにSDカードに保持されます。
 - /userfilesディレクトリ
 - CM-stickディレクトリ
 - SDカードのルートディレクトリ内の「journal.jnl」ファイル
- 装置のPSS 4000ファームウェアが認識している機能のディレクトリのみがSDカードに保持されることに注意してください。
たとえば、ファームウェアバージョンが高い装置のSDカードが装置で使用されている場合、ファームウェアバージョンが高い装置のみで利用できるすべての機能のディレクトリは削除されます。

[7]

次の場合には [7] の手順に従います。

- ▶ SDカードが装置に取り付けられていない。またはデータがないか、整合性がないか、不具合がある。
- ▶ SDカードに命名データがない。またはSDカードの命名データが装置の製品タイプと一致しない。たとえば、この状態は、使用しているSDカードが不適切である場合、または異なる製品タイプの装置を使用している場合に発生することがあります。

- ▶ 装置がPSSu H FS SN SD(-T)(-R)、PSSu H FS SN SD M12 (-T)(-R)、またはPSS67 IO1 16FDIではなく、SDカードの命名データが装置の命名データと一致しない。たとえば、この状態は、新しい装置の命名時に命名データが変更されると発生することがあります。

[7] の手順に従うと、次の操作が実行されます。

- ▶ 装置の命名データが削除されます。
- ▶ 装置にあるプロジェクト名とチェックサム「FS装置プロジェクト」が削除されます。
- ▶ 装置は工場出荷時の既定の命名データを有効にします。
- ▶ 装置はAuto-IPメカニズムを有効にします。
- ▶ オペレーティングステータス「システムセクションを起動中」が実行されます。装置は次の内容を使用します。
 - 工場出荷時の既定の装置名
 - Auto-IPメカニズム経由で割り付けられたIPアドレス

「システムセクションを起動中」での動作:

- ▶ 起動処理で異常が見つからなかった場合、すべてのシステムセクションが起動されます。起動は並行して同時に実行されます (フローチャートのセクションIII～VIを参照)。
- ▶ 起動処理で異常が見つかった場合、影響を受けるシステムセクションがいずれかのエラー状態になります。影響を受けないすべてのシステムセクションは、起動処理を続行してRUN状態になります (フローチャートのセクションIII～VIIを参照)。
- ▶ 起動処理は、次の時点で完了になります。
 - すべてのシステムセクションが正常にコンフィグレーションをロードした (「DIAG」LEDが「緑で点滅」から「緑で点灯」に切り替わる)。
 - システムセクションが異常を記録して (「DIAG」LEDが「緑の点滅」から「オレンジ色の点灯」、「赤の点灯」、または「赤の点滅」に切り替わる)、エラー状態に切り替わった。

これらの場合でも、影響を受けないシステムセクションでは、コンフィグレーションが正常にロードされています。
- ▶ ステータスLED

装置のステータスLEDには、システムセクションのステータス情報が示されます。

11.2.2.4 オペレーティングステータス「装置がエラーなしで実行状態」

このオペレーティングステータスは、すべてのシステムセクションがRUN状態でエラーが発生せずに動作していることが特徴です (フローチャートのセクションIII～VIIを参照)。

▶ ステータスLED

装置のステータスLEDには、システムセクションのステータス情報が示されます。

11.2.2.5 オペレーティングステータス「装置がエラーなしで停止状態」

このオペレーティングステータスは、PAS4000を使用して明確な停止をトリガすることによって達成されます (装置は停止)。

影響

- ▶ システムセクションは現在の処理を終了します。それ以降の処理が停止します。すべてのシステムセクションがSTOP状態になります。停止状態中の動作については、個別に説明します (「[停止コマンド – STOP 状態中の動作](#)」 437]を参照)。

▶ ステータスLED

装置のステータスLEDには、システムセクションのステータス情報が示されます。

処置

PSS 4000専用装置の起動には、次のコマンドを使用できます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトの起動 (PAS4000)

11.2.2.6 オペレーティングステータス「Major FS error」

エラーによりオペレーティングステータスが「FSの重大エラー」になった場合、その影響を受けるPSS 4000専用装置のセクションが定義された通りに終了します。定義された状態になります。

考えられる原因

▶ 妥当性エラー

マイクロプロセッサ間の不規則性 (異なる計算結果) など

▶ メモリエラー

- ▶ 極端に高いネットワーク負荷 (継続的または散発的に高いネットワーク負荷を発生させる装置が原因です。
たとえば、プリンタ、ビデオカメラ、ブロードキャストやマルチキャストテレグラムを頻繁に転送する装置などです)

影響

- ▶ 安全関連システムセクションの重大なエラーの場合、他のすべての安全関連システムセクションは停止するか安全状態に切り替えられます。
 - FSリソースはSTOP状態になり、FSリソースのタスクは実行されなくなります。
 - すべてのハードウェアFS出力は安全状態に切り替えられます。
 - SafetyNET p経由のFSプロセスデータ通信が停止します。
- ▶ FSリソースを介して値が決定され、他のリソースを介してアクセスされるプロセスデータの場合、有効ビット = FALSEの状態、代入値が使用されます。
- ▶ FSリソースを介して値が決定される変数 (揮発性変数、不揮発性変数) の操作方法の詳細については、PAS4000のオンラインヘルプを参照してください。
- ▶ ハードウェアST出力は影響を受けません。
- ▶ 装置のステータスLED
 - 「FS RUN」LEDはオフになり、「FS SNp」LEDライトは赤色に点灯し、「MBUS」LEDは赤色に点滅します。

処置

1. 診断

たとえば、PAS4000を使用して、プロジェクトの診断リストまたは診断ログを読み取ります。

2. エラーの回復

3. コールドスタート／再起動コマンドの実行



重要

次のオペレーティングステータスのコマンドは実行されません。「FSの重大エラー」

PSS 4000専用装置が、オペレーティングステータス「FSの重大エラー」になっていて、PAS4000が次のいずれかのコマンドを送信した場合、装置はコマンドを実行しません。

影響を受けるコマンド:

- プロジェクトの直接ダウンロード
- オリジナルリセット
- コールド／ウォームリセット
- コールド／ウォーム再起動
- 停止コマンドのいずれか

このような場合、装置はPAS4000にエラーメッセージを送信します。

11.2.3 装置のオペレーティングステータスの変更

オペレーティングステータスの変更の説明は、図「オペレーティングステータスと装置のオペレーティングステータスの変更」(「フローチャート」を参照)を参照しています。

[1] コールドスタート／再起動コマンド後のオペレーティングステータスの変更

PSS 4000専用装置のモジュール電源をオンにすると(コールドスタート)、または再起動コマンド(PAS4000)を使用すると、装置がまず「ベースシステムを起動中」オペレーティングステータスに切り替わり、各種起動テストを実行してから、「システムセクションを起動中」ステータスに切り替わります。

基本的に、「ベースシステムを起動中」後のオペレーティングステータスの変更は、装置がどのようなステータスであってもコールドスタート／再起動コマンドで強制的に実行できます(フローチャートのセクションIIを参照)。可用性のために、オペレーティングステータスを強制的に変更するのは、本当に必要なときのみとすることをお勧めします。たとえば、重大なエラーとなったエラーの回復後です。

[2] リセット後のオペレーティングステータスの変更

基本的に、リセット(コールドリセット、リセットボタン／ウォームリセットまたはプロジェクトリセット)は、PSS 4000専用装置がどのようなステータスにあってもトリガできます(フローチャートのセクションIを参照)。その影響(変数の操作方法など)は使用するリセットの種類によって異なります。リセット後、装置はオペレーティングステータス「システムセクションを起動中」に切り替わります。

可用性のために、リセットをトリガするのは、本当に必要なときのみとすることをお勧めします。

リセットの影響については、個別に説明します(「リセットコマンド」 432」を参照)。

[3] 「システムセクションを起動中」から「装置がRUN状態、エラーなし」への切り替え

このオペレーティングステータスの変更は、PSS 4000専用装置のすべてのシステムセクションがRUN状態でエラーなく動作するようになると同時に起こります。

影響

- ▶ ミリ秒タイマが開始します。
- ▶ 不揮発性および揮発性の変数の内容は、「リセット、再起動、起動および停止オプション」に記載されています(「[ローカルおよびリソースグローバル変数への影響](#)」 426」を参照)。
- ▶ PII/PIOのすべての値がゼロに設定されます。

モジュールバスがFSリソースおよび／またはSTリソースによってすでに起動している可能性があります。この場合、PIIには現在の入力値を含めることができます。

- ▶ 診断リストにエントリを追加

[4] 「装置がRUN状態、エラーなし」から「装置がSTOP状態、エラーなし」への切り替え

このオペレーティングステータスの変更が起こるには、PAS4000を使用して明確にトリガされる必要があります(装置の停止など)。

停止の影響については、個別に説明します(「[停止コマンド](#)」 437」を参照)。

[5] 「FSの重大エラー」から「システムセクションを起動中」への切り替え

エラーを回復した後、PSS 4000専用装置は、コールドスタート／再起動コマンドでのみ起動できます。コールドスタート／再起動コマンドを実行すると、装置は「ベースシステムを起動中」になり、起動テストの実行後、ステータス「システムセクションを起動中」の最終的な起動処理に進みます。

11.3 オペレーティングステータスとFSリソースのオペレーティングステータスにおける変化

以下の説明では、PSS 4000専用装置のFSリソースのオペレーティングステータスのみを扱います。必要に応じて装置のオペレーティングステータスを参照します。

図「オペレーティングステータスとFSリソースのオペレーティングステータスの変更」(「[フローチャート !\[\]\(1fa9f096610b70a789c42d45c70a53f9_img.jpg\) 348](#)」を参照) にオペレーティングステータス (グレー) の概要を示します。

- ▶ 角括弧内の数字はオペレーティングステータスのさまざまな変更を示します。
- ▶ ローマ数字「I」または「II」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「[フローチャート !\[\]\(9ab6aad62cedd266a2eb5858982152fa_img.jpg\) 348](#)」を参照)。その影響はFSリソースだけではなく、PSS 4000専用装置のすべてのシステムセクションに及ぶためです。
- ▶ ローマ数字「III」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「[フローチャート](#)」を参照)。PSS 4000専用装置のシステムセクションを起動すると、FSリソースが起動するためです。

11.3.1 フローチャート

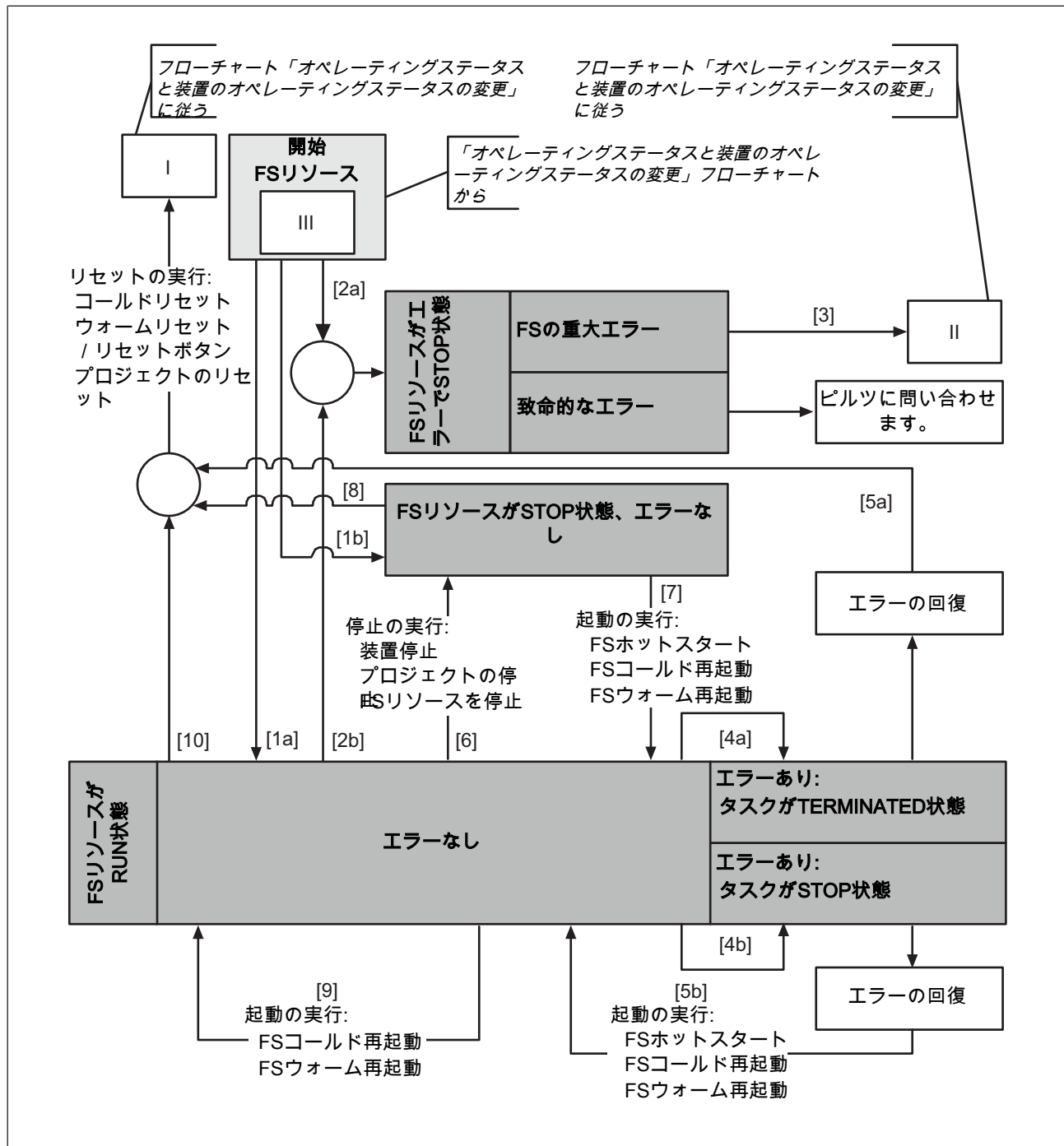


図: オペレーティングステータスとFSリソースのオペレーティングステータスの変更

11.3.2 FSリソースのオペレーティングステータス

各種オペレーティングステータスの説明では、PSS 4000専用装置に有効な装置プロジェクトがあることを前提としています

FSリソースは次のオペレーティングステータスに変更します (「[フローチャート](#)  348」を参照)。

- ▶ オペレーティングステータス「FSリソースがRUN状態」:
 - FSリソースがRUN状態、エラーなし
 - FSリソースがエラーでRUN状態: タスクがTERMINATED状態
 - FSリソースがエラーでRUN状態: タスクがSTOP状態
- ▶ オペレーティングステータス「FSリソースがSTOP状態、エラーなし」
- ▶ オペレーティングステータス「FSリソースがエラーでSTOP状態」:
 - 致命的なエラー
 - FSの重大エラー



情報

各オペレーティングステータスの説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「[リセット、再起動、起動、停止のオプション](#)  424」を参照)。

11.3.2.1 FSリソースがエラーなしで実行状態

このオペレーティングステータスは、次のオプションのいずれかを通して達成されます。

- ▶ 以下にトリガされ、正常に起動プロセスが実行された後
 - 装置のコールドスタートまたは装置の再起動コマンド (PAS4000を使用)
 - 装置のリセットボタンの使用
 - リセットのトリガ (PAS4000を使用)
(コールドリセット、ウォームリセット、プロジェクトリセット)
- ▶ 起動または再起動後 (PAS4000を使用)
(FSコールド再起動、FSウォーム再起動、FSホットスタート)

影響

- ▶ オペレーティングステータス「FSリソースがRUN状態、エラーなし」で、FSリソースのタスクがエラーなく実行されます。
- ▶ 装置のステータスLED:
[FS RUN] LEDが緑色に点灯します。

11.3.2.2 FSリソースがエラーなしで停止状態

このオペレーティングステータスは、PAS4000を使用して停止をトリガすることによって達成されます (装置は停止、FSリソースは停止、プロジェクトは停止)。

影響

- ▶ リソースがSTOP状態の間のハードウェア入力、ハードウェア出力、プロセスデータ、システムセクションの動作については、個別に説明します (「[停止コマンド – STOP状態中の動作](#)」 437」を参照)。
- ▶ ローカル変数への影響とリソースグローバル変数への影響は、別々に記載します (「[ローカルおよびリソースグローバル変数への影響](#)」 426」を参照)。
- ▶ 装置のステータスLED:
「FS-RUN」LEDが消灯します。

処置

- ▶ FSリソースの (再) 起動:
 - FSホットスタート (PAS4000)
 - FSCールド再起動 (PAS4000)
 - FSウォーム再起動 (PAS4000)
- ▶ 装置のリセット:
 - ウォームリセット (PAS4000) またはリセットボタン (装置)
 - コールドリセット (PAS4000)
- ▶ プロジェクトのリセット
 - プロジェクトのウォームリセット (PAS4000)
 - プロジェクトのコールドリセット (PAS4000)

11.3.2.3 FSリソースがエラーで実行状態: タスクが終了状態

エラーによりこのオペレーティングステータスになった場合、影響を受けるタスクは、定義された通りに終了します。定義された状態になります。

考えられる原因

(IEC 61131-3、テーブルE1も参照してください)

- ▶ 実行時間エラー
実行時間が超過したためのタイムアウトエラーなど

影響

- ▶ 影響を受けるFSリソースのタスクは実行されなくなり、他のすべてのFSリソースのタスクは引き続き実行されます。
- ▶ FSリソースは引き続きRUN状態になります。
- ▶ ローカル変数として宣言されるか、停止したタスクのPOU内でVAR_EXTERNALとして参照される出力PI変数には、次の操作が適用されます。
I/Oマッピングされたデータシンク (入力PI変数、ハードウェア出力など) および有効ビット = FALSEで代入値が使用されます。
- ▶ Multiブロック内で出力PIポイントとして作成されるか、VAR_EXTERNALとして参照されるすべての出力PI変数には、次の操作が適用されます。
I/Oマッピングされたデータシンク (入力PI変数、ハードウェア出力など) および有効ビット = FALSEで代入値が使用されます。
- ▶ 次の点がリソースグローバル変数 (出力PIではない) に適用されます。リソースグローバル変数は終了したタスクのブロック内で「VAR_EXTERNAL」を介して参照されます。
有効となり、保存される最新の値を含みますが、初期値には設定されません。
- ▶ 終了したタスクのブロック内でリソースグローバル変数が「VAR_EXTERNAL」を介して参照される場合、そこにある値を上書きできます。この上書きの結果として、リソースグローバル出力PI変数が再び有効としてマークされます。
- ▶ 変数 (揮発性変数、不揮発性変数) の操作方法の詳細については、PAS4000のオンラインヘルプを参照してください。
- ▶ 装置のステータスLED:
[FS-RUN] LEDは緑色に点滅します。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. リセットの実行:
ウォームリセットまたはコールドリセットを使用します

11.3.2.4 FSリソースがエラーで実行状態: タスクが停止状態

エラーによりこのオペレーティングステータスになった場合、影響を受けるタスクは、定義された通りに停止します。定義された状態になります。

考えられる原因

(IEC 61131-3、テーブルE1も参照してください)

▶ 実行時エラー

変数の値が定義された範囲を超えている、機能実行の数値の結果がデータ型の値の範囲を超えている、0による除算など

影響

- ▶ 影響を受けるFSリソースのタスクは実行されなくなり、他のすべてのFSリソースのタスクは引き続き実行されます。
- ▶ FSリソースは引き続きRUN状態になります。
- ▶ ローカル変数として宣言されるか、停止したタスクのPOU内でVAR_EXTERNALとして参照される出力PI変数には、次の操作が適用されます。
I/Oマッピングされたデータシンク (入力PI変数、ハードウェア出力など) および有効ビット = FALSEで代入値が使用されます。
- ▶ Multiブロック内で出力PIポイントとして作成されるか、VAR_EXTERNALとして参照されるすべての出力PI変数には、次の操作が適用されます。
I/Oマッピングされたデータシンク (入力PI変数、ハードウェア出力など) および有効ビット = FALSEで代入値が使用されます。
- ▶ 次の点がリソースグローバル変数 (出力PI変数ではない) に適用されます。リソースグローバル変数は停止したタスクのブロック内で「VAR_EXTERNAL」を介して参照されます。
有効となり、保存される最新の値を含みますが、初期値には設定されません。
- ▶ 停止したタスクのブロック内でリソースグローバル変数が「VAR_EXTERNAL」を介して参照される場合、そこにある値を上書きできます。この上書きの結果として、リソースグローバル出力PI変数が再び有効としてマークされます。
- ▶ 変数 (揮発性変数、不揮発性変数) の操作方法の詳細については、PAS4000のオンラインヘルプを参照してください。
- ▶ 装置のステータスLED:
[FS-RUN] LEDは緑色に点滅します。

処置

1. 診断

PAS4000を使用した診断リストまたは診断ログの読み取りなど

2. エラーの回復

3. リセットまたは起動 (再起動) の実行:

FSホットスタート、FSウォームリスタート、またはFSコールドリスタートを使用します

いずれかのリセットコマンドも使用できます。リセットコマンドは、FSリソースだけではなく、装置のすべてのシステムセクションに影響します。

11.3.2.5 Major FS error

エラーによりこのオペレーティングステータスになった場合、影響を受けるFSリソースのセクションは定義された通りに終了します。定義された状態になります。

FSの重大エラーはPSS 4000専用装置のすべての安全関連部に影響します (「[オペレーティングステータス「FSの重大エラー」](#)を参照  344])。

11.3.2.6 Fatal error

エラーによりオペレーティングステータスが「致命的なエラー」になった場合、PSS 4000専用装置上で実行されているすべてのプロセスを順番に可能な限り終了します。安全関連システムセクションは、停止されるか、安全状態に切り替えられます。

考えられる原因

- ▶ 初期化エラー

- ▶ 内部ファームウェアエラー

(たとえば、メモリエラーが検出されても、そのために装置のファームウェアの重大なエラーになると、ファームウェアが正常に実行されなくなることがあります)

- ▶ ハードウェアの欠陥

影響

- ▶ FSハードウェア出力が安全状態に切り替えられます。

- ▶ 変数の内容およびSTハードウェア入力/STハードウェア出力のステータスがランダムになります。

- ▶ 装置のステータスLED:

LED「FS RUN」、「ST RUN」、「FS SNp」、「ST SNp」、および「DIAG」のステータスがランダムになります。

処置

ユーザがエラーを回復することはできません。装置がこのオペレーティングステータスになっている場合、次の手順に従います。

1. エラーが発生した状況を書き留めます。
2. PAS4000を使用して診断リストおよび診断ログを読み込むか出力するために、コールドスタート／再起動コマンドを実行します。

ただし、これは、通信領域の故障による致命的なエラーが発生しなかった場合のみ可能です。

3. ピルツに問い合わせます。

11.3.3 FSリソースのオペレーティングステータスの変化

オペレーティングステータスの変更に関する説明では、装置に有効な装置プロジェクトがあることを前提としています。数字は、図「オペレーティングステータスとFSリソースのオペレーティングステータスの変更」(「フローチャート」[\[348\]](#))を参照)のオペレーティングステータスの変更に対応しています。



情報

オペレーティングステータスのそれぞれの変更に関する説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します(「リセット、再起動、起動、停止のオプション」[\[424\]](#))を参照)。

[1a] 「システムセクションを起動中」から「FSリソースがRUN状態、エラーなし」への切り替え

「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです(「フローチャート」を参照)。システムセクションの起動時に、安全関連システムセクションでエラーが見つからなかった場合、FSリソースが起動され、ステータス「FSリソースがRUN状態、エラーなし」になります。

[1b] 「システムセクションを起動中」から「FSリソースがSTOP状態、エラーなし」への切り替え

「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです(「フローチャート」を参照)。さまざまな状況により、FSリソースがオペレーティングステータス「FSリソースがRUN状態、エラーなし」ではなく、「FSリソースがSTOP状態、エラーなし」になることがあります。

- ▶ STOPコマンド([6]を参照)を手動で実行することにより、FSリソースはオペレーティングステータス「FSリソースがSTOP状態、エラーなし」になります。この場合、起動コマンド([7]を参照)ではなく、リセットコマンド([8]を参照)を実行すると、ただしこの場合、FSリソースはいかなるリセットコマンドを使用してもRUN状態にならず、再び「FSリソースがSTOP状態、エラーなし」になります。
- ▶ FSリソースがオペレーティングステータス「FSリソースがエラーでRUN状態: タスクがSTOP状態」になります。エラーを回復した後で起動コマンド([5b]を参照)ではなく、リセットコマンドを実行します。ただしこの場合、FSリソースはいかなるリセットコマンドを使用してもRUN状態にならず、「FSリソースがSTOP状態、エラーなし」になります。

[2a] 「システムセクションを起動中」からエラーステータスへの切り替え

システムセクションの起動時に、安全関連システムセクションでエラーが見つかり(「フローチャート」を参照)、FSリソースは、エラーに応じてオペレーティングステータス「FSの重大エラー」または「致命的なエラー」になります。

[2a] 「FSリソースがRUN状態、エラーなし」からエラーステータスへの切り替え FSリソースがRUN状態になっているときに、安全関連システムセクションでエラーが見つかり、FSリソースはエラーに応じてオペレーティングステータス「FSの重大エラー」または「致命的なエラー」になります。

[3] 「FSの重大エラー」から「システムセクションを起動中」への切り替え

エラーを回復した後、FSリソースは、コールドスタート／再起動コマンドでのみ起動できます。コールドスタート／再起動コマンドを実行すると、PSS 4000専用装置は「ベースシステムを起動中」になり、起動テストの実行後、ステータス「システムセクションを起動中」の最終的な起動処理に進みます(「フローチャート」を参照)。

[4a] 「FSリソースがRUN状態、エラーなし」から「FSリソースがエラーでRUN状態: タスクがTERMINATED状態」への切り替え

FSリソースタスクでエラー (例: 実行時間エラー) が見つかったと、そのタスクは中止されて、それ以上は実行されません。他のFSリソースタスクは、すべて引き続き実行されます。

[4b] 「FSリソースがRUN状態、エラーなし」から「FSリソースがエラーでRUN状態: タスクがSTOP状態」への切り替え

FSリソースタスクでエラー (例: 実行時エラー) が見つかったと、そのタスクは中止されて、それ以上は実行されません。他のFSリソースタスクは、すべて引き続き実行されます。

[5a] 「FSリソースがRUN状態、エラーなし」の後、「FSリソースがエラーでRUN状態: タスクがTERMINATED状態」からの切り替え

エラーを回復した後、FSリソースを次のいずれかのリセットコマンドで再起動できます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[5b] 「FSリソースがRUN状態、エラーなし」の後、「FSリソースがエラーでRUN状態: タスクがSTOP状態」からの切り替え

異常を回復した後、影響を受けたタスクを次のいずれかの (再) 起動コマンドで再起動できます。

- ▶ FSホットスタート
- ▶ FSウォーム再起動
- ▶ FSウォームスタート

[6] 「FSリソースがRUN状態、エラーなし」から「FSリソースがSTOP状態、エラーなし」への切り替え

FSリソースがエラーなしでRUN状態になっている場合、ユーザはFSリソースを明示的にSTOP状態に切り替えることができます。STOP状態はユーザが明示的にFSリソースを再起動するまで維持されます。

設備セクションの試運転を行う場合や、調整を行う場合など、オペレーティングステータスでこの変更をトリガすると便利な場合があります。

FSリソースのSTOP状態は、次のいずれかによりトリガできます。

- ▶ プロジェクトの停止 (PAS4000)
- ▶ 装置の停止 (PAS4000)
- ▶ FSリソースの停止 (PAS4000)

[7] 「FSリソースがSTOP状態、エラーなし」から「FSリソースがRUN状態、エラーなし」への切り替え

オペレーティングステータスでのこの変更は、次のいずれかのコマンドで実現できます。

- ▶ FSホットスタート (PAS4000)
- ▶ FSコールド再起動 (PAS4000)
- ▶ FSウォーム再起動 (PAS4000)

[8] ステータス「FSリソースがSTOP状態、エラーなし」でのリセット

このオペレーティングステータス ([8] を参照) では、いずれのリセットコマンドを実行しても、システムセクションの起動時にFSリソースが再び「FSリソースがSTOP状態、エラーなし」に戻ります ([1b] を参照)。いずれのリセットコマンドを使用しても、オペレーティングステータスを「FSリソースがRUN状態、エラーなし」に切り替えることはできません。

[9] ステータス「FSリソースがRUN状態、エラーなし」での再起動

試運転の場合など、変数を定義済みの値に設定する必要がある場合、いずれかの再起動コマンドを使用できます (「[ローカルおよびリソースグローバル変数への影響](#)」 426」を参照)。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ FSウォーム再起動 (PAS4000)
- ▶ FScールド再起動 (PAS4000)

[10] 「FSリソースがRUN状態、エラーなし」から「システムセクションを起動中」への切り替え

オペレーティングステータスでのこの変更は、装置のすべてのシステムセクションに影響を及ぼします (「フローチャート」を参照)。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

11.4 オペレーティングステータスとSTリソースのオペレーティングステータスにおける変化

以下の説明では、PSS 4000専用装置のSTリソースのオペレーティングステータスのみを扱います。必要に応じて装置のオペレーティングステータスを参照します。

図「オペレーティングステータスとSTリソースのオペレーティングステータスの変更」(「[フローチャート](#)  360」を参照) にオペレーティングステータス (グレー) の概要を示します。

- ▶ 角括弧内の数字はオペレーティングステータスのさまざまな変更を示します。
- ▶ ローマ数字「I」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「[フローチャート](#)」を参照)。その影響はSTリソースだけではなく、PSS 4000専用装置のすべてのシステムセクションに及ぶためです。
- ▶ ローマ数字「IV」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「[フローチャート](#)」を参照)。PSS 4000専用装置のシステムセクションを起動すると、STリソースが起動するためです。

11.4.1 フローチャート

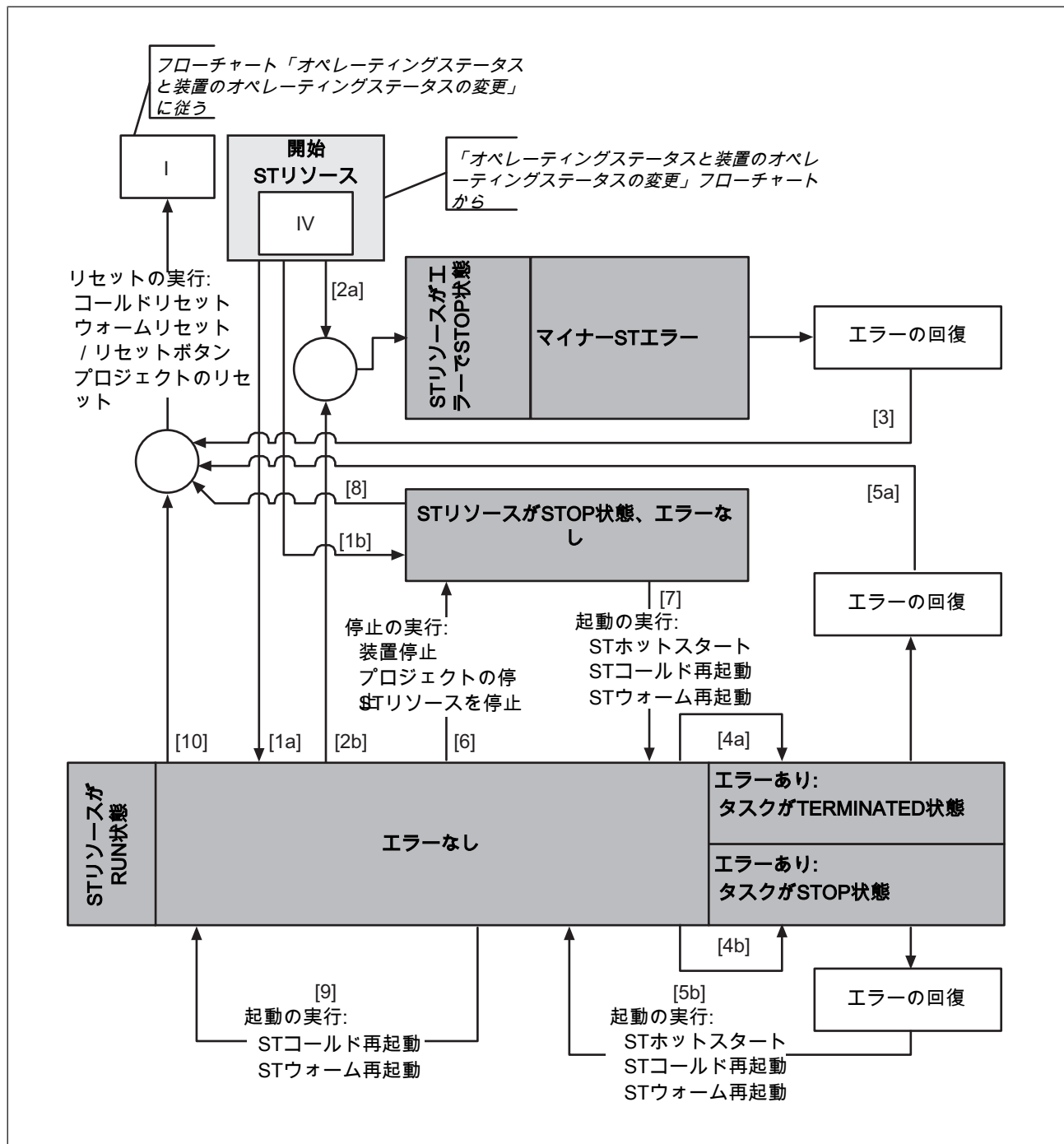


図: オペレーティングステータスとSTリソースのオペレーティングステータスの変更

11.4.2 STリソースのオペレーティングステータス

各種オペレーティングステータスの説明では、PSS 4000専用装置に有効な装置プロジェクトがあることを前提としています

STリソースは次のオペレーティングステータスに変更します (「[フローチャート](#)」[\[📖 360\]](#)を参照)。

- ▶ オペレーティングステータス「STリソースがRUN状態」:
 - STリソースがRUN状態、エラーなし
 - STリソースがエラーでRUN状態: タスクがTERMINATED状態
 - STリソースがエラーでRUN状態: タスクがSTOP状態
- ▶ オペレーティングステータス「STリソースがSTOP状態、エラーなし」
- ▶ オペレーティングステータス「STリソースがエラーでSTOP状態」:
 - マイナーSTエラー



情報

各オペレーティングステータスの説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「[リセット、再起動、起動、停止のオプション](#)」[\[📖 424\]](#)を参照)。

11.4.2.1 STリソースがエラーなしで実行状態

このオペレーティングステータスは、次のオプションのいずれかを通して達成されます。

- ▶ 以下にトリガされ、正常に起動プロセスが実行された後
 - 装置のコールドスタートまたは装置の再起動コマンド (PAS4000を使用)
 - 装置のリセットボタンの使用
 - リセットのトリガ (PAS4000を使用)
(コールドリセット、ウォームリセット、プロジェクトリセット)
- ▶ 起動または再起動後 (PAS4000を使用)
(STコールド再起動、STウォーム再起動、STホットスタート)

影響

- ▶ オペレーティングステータス「STリソースがRUN状態、エラーなし」で、STリソースのタスクがエラーなく実行されます。
- ▶ 装置のステータスLED:
[ST RUN] LEDが緑色に点灯します。

11.4.2.2 STリソースがエラーなしで停止状態

このオペレーティングステータスは、PAS4000を使用して停止をトリガすることによって達成されます (装置は停止、STリソースは停止、プロジェクトは停止)。

影響

- ▶ リソースがSTOP状態の間のハードウェア入力、ハードウェア出力、プロセスデータ、システムセクションの動作については、個別に説明します (「[停止コマンド – STOP状態中の動作](#)」 437」を参照)。
- ▶ ローカル変数への影響とリソースグローバル変数への影響は、別々に記載します (「[ローカルおよびリソースグローバル変数への影響](#)」 426」を参照)。
- ▶ 装置のステータスLED:
[ST RUN] LEDが消灯します。

処置

- ▶ STリソースの (再) 起動:
 - STホットスタート (PAS4000)
 - STコールド再起動 (PAS4000)
 - STウォーム再起動 (PAS4000)
- ▶ 装置のリセット:
 - ウォームリセット (PAS4000) またはリセットボタン (装置)
 - コールドリセット (PAS4000)
- ▶ プロジェクトのリセット
 - プロジェクトのウォームリセット (PAS4000)
 - プロジェクトのコールドリセット (PAS4000)

11.4.2.3 STリソースがエラーで実行状態: タスクが終了状態

エラーによりこのオペレーティングステータスになった場合、影響を受けるタスクは、定義された通りに終了します。定義された状態になります。

考えられる原因

(IEC 61131-3、テーブルE1も参照してください)

▶ 実行時間エラー

タスクサイクル時間内にタスクが完了しなかった

影響

- ▶ 影響を受けるSTリソースのタスクは実行されなくなり、他のすべてのSTリソースのタスクは引き続き実行されます。
- ▶ STリソースは引き続きRUN状態になります。
- ▶ ローカル変数として宣言されるか、停止したタスクのPOU内でVAR_EXTERNALとして参照される出力PI変数には、次の操作が適用されます。
I/Oマッピングされたデータシンク (入力PI変数、ハードウェア出力など) および有効ビット = FALSEで代入値が使用されます。
- ▶ Multiブロック内で出力PIポイントとして作成されるか、VAR_EXTERNALとして参照されるすべての出力PI変数には、次の操作が適用されます。
I/Oマッピングされたデータシンク (入力PI変数、ハードウェア出力など) および有効ビット = FALSEで代入値が使用されます。
- ▶ 次の点がリソースグローバル変数 (出力PI変数ではない) に適用されます。リソースグローバル変数は停止したタスクのブロック内でVAR_EXTERNALを介して参照されます。
有効となり、保存される最新の値を含みますが、初期値には設定されません。
- ▶ 停止していないタスクのPOU内でリソースグローバル変数がVAR_EXTERNALを介して参照される場合、そこにある値を上書きできます。この上書きの結果として、リソースグローバル出力PI変数が再び有効としてマークされます。
- ▶ 変数 (揮発性変数、不揮発性変数) の操作方法の詳細については、PAS4000のオンラインヘルプを参照してください。
- ▶ 装置のステータスLED:
[ST-RUN] LEDは緑色に点滅します。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. リセットの実行:
ウォームリセットまたはコールドリセットを使用します

11.4.2.4 STリソースがエラーで実行状態: タスクが停止状態

エラーによりこのオペレーティングステータスになった場合、影響を受けるタスクは、定義された通りに終了します。定義された状態になります。

考えられる原因

(IEC 61131-3、テーブルE1も参照してください)

▶ 実行時エラー

変数の値が定義された範囲を超えている、機能実行の数値の結果がデータ型の値の範囲を超えている、0による除算など

影響

- ▶ 影響を受けるSTリソースのタスクは実行されなくなり、他のすべてのSTリソースのタスクは引き続き実行されます。
- ▶ STリソースは引き続きRUN状態になります。
- ▶ ローカル変数として宣言されるか、停止したタスクのPOU内でVAR_EXTERNALとして参照される出力PI変数には、次の操作が適用されます。
I/Oマッピングされたデータシンク (入力PI変数、ハードウェア出力など) および有効ビット = FALSEで代入値が使用されます。
- ▶ Multiブロック内で出力PIポイントとして作成されるか、VAR_EXTERNALとして参照されるすべての出力PI変数には、次の操作が適用されます。
I/Oマッピングされたデータシンク (入力PI変数、ハードウェア出力など) および有効ビット = FALSEで代入値が使用されます。
- ▶ 次の点がリソースグローバル変数 (出力PI変数ではない) に適用されます。リソースグローバル変数は停止したタスクのブロック内でVAR_EXTERNALを介して参照されます。
有効となり、保存される最新の値を含みますが、初期値には設定されません。
- ▶ 停止していないタスクのPOU内でリソースグローバル変数がVAR_EXTERNALを介して参照される場合、そこにある値を上書きできます。この上書きの結果として、リソースグローバル出力PI変数が再び有効としてマークされます。
- ▶ 変数 (揮発性変数、不揮発性変数) の操作方法の詳細については、PAS4000のオンラインヘルプを参照してください。
- ▶ 装置のステータスLED:
[ST-RUN] LEDは緑色に点滅します。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. (再) 起動の実行
STホットスタート、STウォーム再起動、またはSTコールド再起動を使用します
いずれのリセットコマンドも使用できます。リセットコマンドは、STリソースだけではなく、装置のすべてのシステムセクションに影響します。

11.4.2.5 Minor ST error

エラーによりこのオペレーティングステータスになった場合、影響を受けるSTリソースのセクションは定義された通りに終了します。定義された状態になります。

考えられる原因

- ▶ 不揮発性変数のメモリのエラー

影響

- ▶ 影響を受けるSTリソースのタスクは実行されなくなり、他のすべてのSTリソースのタスクは停止します。
- ▶ STリソースはSTOP状態になります。
- ▶ 値がこのタスクによって決定されるプロセスデータの場合、代入値を使用します (PI変数を使用、同じく、対応する有効ビット = FALSE)。
- ▶ 変数 (揮発性変数、不揮発性変数) の操作方法の詳細については、PAS4000のオンラインヘルプを参照してください。
- ▶ 装置のステータスLED:
[ST RUN] LEDが消灯します。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. リセットの実行

11.4.3 STリソースのオペレーティングステータスの変化

オペレーティングステータスの変更に関する説明では、装置に有効な装置プロジェクトがあることを前提としています。数字は、図「オペレーティングステータスとSTリソースのオペレーティングステータスの変更」(「フローチャート」[360])を参照)のオペレーティングステータスの変更に対応しています。



情報

オペレーティングステータスのそれぞれの変更に関する説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します(「リセット、再起動、起動、停止のオプション」[424])を参照)。

[1a] 「システムセクションを起動中」から「STリソースがRUN状態、エラーなし」への切り替え

「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです(「フローチャート」を参照)。システムセクションの起動時に、安全関連システムセクションでエラーが見つからなかった場合、STリソースが起動され、ステータス「STリソースがRUN状態、エラーなし」になります。

[1b] 「システムセクションを起動中」から「STリソースがSTOP状態、エラーなし」への切り替え

「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです(「フローチャート」を参照)。さまざまな状況により、システムセクションの起動時に、STリソースがオペレーティングステータス「STリソースがRUN状態、エラーなし」ではなく、「STリソースがSTOP状態、エラーなし」になることがあります。

- ▶ STOPコマンド ([6] を参照) を手動で実行することにより、STリソースはオペレーティングステータス「STリソースがSTOP状態、エラーなし」になります。この場合、起動コマンド ([7] を参照) ではなく、リセットコマンド ([8] を参照) を実行すると、ただしこの場合、STリソースはいかなるリセットコマンドを使用してもRUN状態にならず、再び「STリソースがSTOP状態、エラーなし」になります。
- ▶ STリソースがオペレーティングステータス「STリソースがエラーでRUN状態: タスクがSTOP状態」になります。エラーを回復した後で起動コマンド ([5b] を参照) ではなく、リセットコマンドを実行します。ただしこの場合、STリソースはいかなるリセットコマンドを使用してもRUN状態にならず、「STリソースがSTOP状態、エラーなし」になります。

[2a] 「システムセクションを起動中」からエラーステータス「マイナーSTエラー」への切り替え

システムセクションの起動時にエラーが見つかった(「フローチャート」を参照)、STリソースはオペレーティングステータス「マイナーSTエラー」になります。

[2b] 「STリソースがRUN状態、エラーなし」からエラーステータス「マイナーSTエラー」への切り替え

STリソースがRUN状態の間にエラーが見つかった、STリソースはオペレーティングステータス「マイナーSTエラー」になります。

[3] 「マイナーSTエラー」から「システムセクションを起動中」への切り替え

エラーを回復した後、同時にFSエラーが存在しない場合はSTリソースを次のいずれかのリセットコマンドで再起動できます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (PSS 4000専用装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[4a] 「STリソースがRUN状態、エラーなし」から「STリソースがエラーでRUN状態: タスクがTERMINATED状態」への切り替え

STリソースタスクでエラー (例: 実行時間エラー) が見つかり、そのタスクは中止されて、それ以上は実行されません。他のSTリソースタスクは、すべて引き続き実行されます。

[4b] 「STリソースがRUN状態、エラーなし」から「STリソースがエラーでRUN状態: タスクがSTOP状態」への切り替え

STリソースタスクでエラー (例: 実行時エラー) が見つかり、そのタスクは中止されて、それ以上は実行されません。他のSTリソースタスクは、すべて引き続き実行されます。

[5a] 「STリソースがRUN状態、エラーなし」の後、「FSリソースがエラーでRUN状態: タスクがTERMINATED状態」からの切り替え

エラーを回復した後、STリソースを次のいずれかのリセットコマンドで再起動できます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (PSS 4000専用装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[5b] 「STリソースがRUN状態、エラーなし」の後、「FSリソースがエラーでRUN状態: タスクがSTOP状態」からの切り替え

エラーを回復した後、影響を受けたタスクを次のいずれかの (再) 起動コマンドで再起動できます。

- ▶ STホットスタート
- ▶ STウォーム再起動
- ▶ STウォームスタート

いずれのリセットコマンドも使用できます。リセットコマンドは、STリソースだけでなく、PSS 4000専用装置全体に影響します。

[6] 「STリソースがRUN状態、エラーなし」から「STリソースがSTOP状態、エラーなし」への切り替え

STリソースがエラーなしでRUN状態になっている場合、ユーザはSTリソースを明示的にSTOP状態に切り替えることができます。STOP状態はユーザが明示的にSTリソースを再起動するまで維持されます。

設備セクションの試運転を行う場合や、調整を行う場合など、オペレーティングステータスでこの変更をトリガすると便利な場合があります。

STリソースのSTOP状態は、次のいずれかのコマンドによりトリガできます。

- ▶ プロジェクトの停止 (PAS4000)
- ▶ 装置の停止 (PAS4000)
- ▶ STリソースの停止 (PAS4000)

[7] 「STリソースがSTOP状態、エラーなし」から「STリソースがRUN状態、エラーなし」への切り替え

オペレーティングステータスでのこの変更は、次のいずれかのコマンドで実現できます。

- ▶ STホットスタート (PAS4000)
- ▶ STコールド再起動 (PAS4000)
- ▶ STウォーム再起動 (PAS4000)

[8] ステータス「STリソースがSTOP状態、エラーなし」でのリセット

このオペレーティングステータス ([8] を参照) では、いずれのリセットコマンドを実行しても、システムセクションの起動時にSTリソースが再び「STリソースがSTOP状態、エラーなし」に戻ります ([1b] を参照)。いずれのリセットコマンドを使用しても、オペレーティングステータスを「STリソースがRUN状態、エラーなし」に切り替えることはできません。

[9] ステータス「STリソースがRUN状態、エラーなし」での再起動

試運転の場合など、変数を定義済みの値に設定する必要がある場合、いずれかの再起動コマンドを使用できます (「[ローカルおよびリソースグローバル変数への影響](#)」 426] を参照)。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ STウォーム再起動 (PAS4000)
- ▶ STコールド再起動 (PAS4000)

[10] 「STリソースがRUN状態、エラーなし」から「システムセクションを起動中」への切り替え

オペレーティングステータスでのこの変更は、PSS 4000専用装置のすべてのシステムセクションに影響を及ぼします (「[フローチャート](#)」を参照)。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (PSS 4000専用装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

11.5 オペレーティングステータスとFSモジュールバスPSSuのオペレーティングステータスの変更

以下の説明では、PSS 4000専用装置のFSモジュールバスPSSuのオペレーティングステータスのみを扱います。必要に応じて装置のオペレーティングステータスを参照します。

図「オペレーティングステータスとFSモジュールバスPSSuのオペレーティングステータスの変更」(「[フローチャート 371](#)」を参照) にオペレーティングステータス (グレー) の概要を示します。

- ▶ 角括弧内の数字はオペレーティングステータスのさまざまな変更を示します。
- ▶ ローマ数字「I」または「II」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (フローチャート)。その影響はFSモジュールバスPSSuだけではなく、PSS 4000専用装置のすべてのシステムセクションに及ぶためです。
- ▶ ローマ数字「Va」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (フローチャートを参照)。PSS 4000専用装置のシステムセクションを起動すると、FSモジュールバスPSSuが起動するためです。

11.5.1 フローチャート

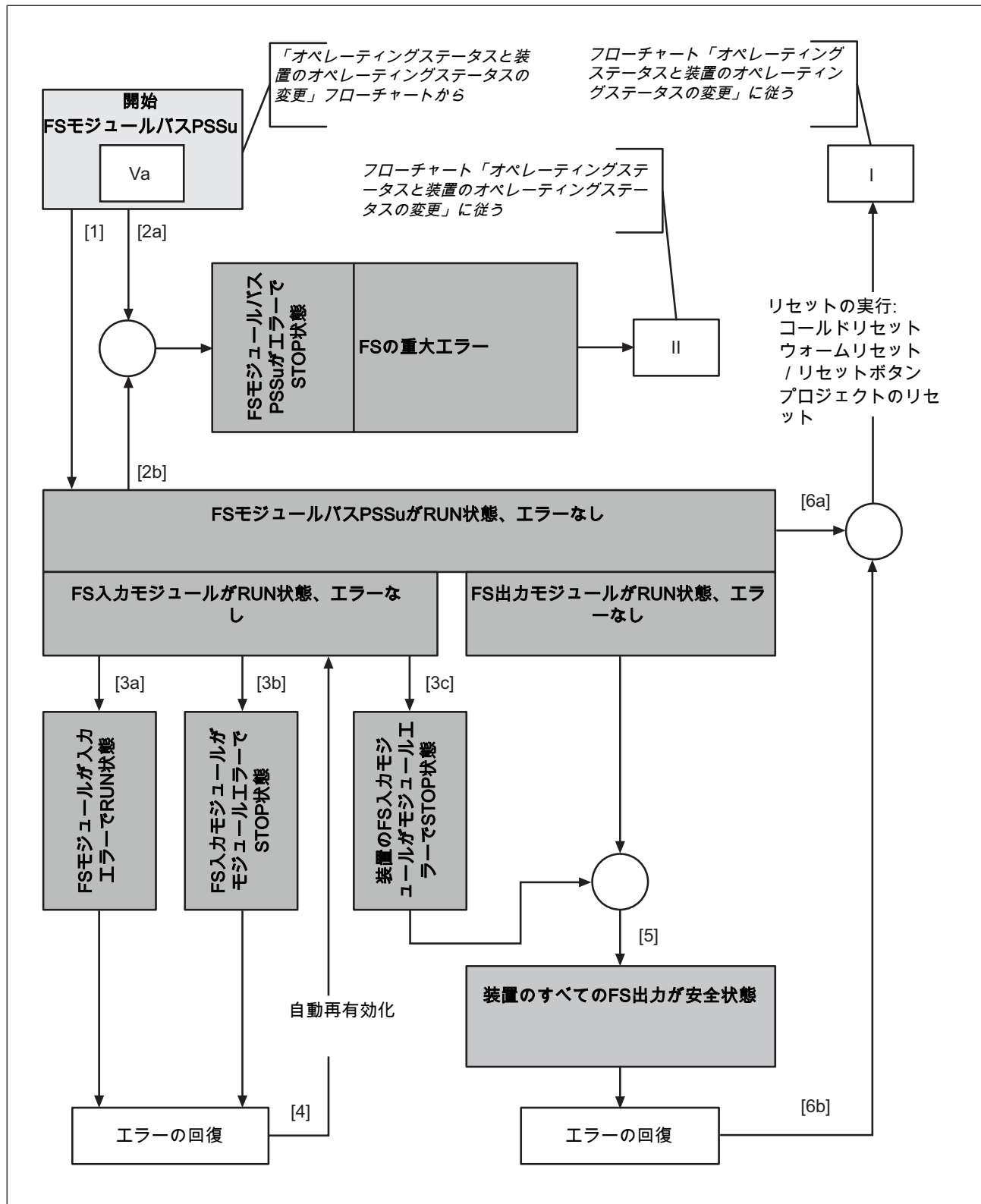


図: オペレーティングステータスとFSモジュールバスPSSuのオペレーティングステータスの変更

11.5.2 FSモジュールバスPSSuのオペレーティングステータス

各種オペレーティングステータスの説明では、PSS 4000専用装置に有効な装置プロジェクトがあることを前提としています

FSモジュールバスPSSuは次のオペレーティングステータスに変更します (「[フローチャート](#)  371」を参照)。

▶ オペレーティングステータス「FSモジュールバスがRUN状態、エラーなし」が以下を伴う場合

- 「FS入力モジュールがRUN状態、エラーなし」
- 「FS入力モジュールが入力エラーでRUN状態」
- 「FS入力モジュールがモジュールエラーでSTOP状態」
- 「装置のFS入力モジュールがモジュールエラーでSTOP状態」
- 「FS出力モジュールがRUN状態、エラーなし」
- 「装置のすべてのFS出力が安全状態」

▶ オペレーティングステータス「FSモジュールバスPSSuがエラーでSTOP状態」:

- 「FSの重大エラー」



情報

各オペレーティングステータスの説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「[リセット、再起動、起動、停止のオプション](#)  424」を参照)。

11.5.2.1 FSモジュールバスPSSuがRUN状態、エラーなし

このオペレーティングステータスは、次のオプションのいずれかを通して達成されます。

- ▶ 以下にトリガされ、正常に起動プロセスが実行された後
 - 装置のコールドスタートまたは装置の再起動コマンド (PAS4000を使用)
 - 装置のリセットボタンの使用
 - リセットのトリガ (PAS4000を使用)
(コールドリセット、ウォームリセット、プロジェクトリセット)

影響

- ▶ FSモジュールバスがエラーのないRUN状態になります。
- ▶ FSモジュールがエラーなく動作しています。つまり、
 - 入力の入力信号／入力値が正しく記録され、FSモジュールバスPSSuを介してエラーなく送信されています。
 - 出力が正しく有効化されています。
- ▶ 装置のステータスLED:
STモジュールバスPSSuもエラーなく動作している場合、[MBUS] LEDが緑色に点灯します。

11.5.2.2 FS入力モジュールがRUN状態、エラーなし

オペレーティングステータスは、モジュールバスのステータスに関わらずモジュールの動作準備が完了するとすぐに達成されます。FSモジュールバスPSSuは、FS入力モジュールがヘッドモジュールヘデータを伝送するための入力データを供給する場所です。

11.5.2.3 FSモジュールが入力エラーで実行状態

FS入力エラーは影響を受けるFSハードウェア入力にのみ影響します。FSモジュールバスはオペレーティングステータス「FSモジュールバスPSSuがRUN状態、エラーなし」を維持します。

考えられる原因

- ▶ テストパルスエラー
- ▶ 入力エラー

影響

- ▶ 代入値は、有効ビット = FALSEと共に入力信号／入力値で使用されます。
- ▶ FS入力モジュールはRUN状態を維持し、影響を受けないFSハードウェア入力は引き続き処理されます。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. エラーが回復されると、FSハードウェア入力は自動的に再有効化されます。

11.5.2.4 FS入力モジュールがモジュールエラーでSTOP状態

モジュールのエラーは、影響を受けるFS入力モジュールにのみ影響します。FSモジュールバスはオペレーティングステータス「FSモジュールバスPSSuがRUN状態、エラーなし」を維持します。

考えられる原因

- ▶ 温度エラー
- ▶ 電圧エラー (モジュール電源)
- ▶ モジュールが動作中に切断された
- ▶ 欠陥のあるモジュールが動作中に検出された

影響

- ▶ 代入値は、有効ビット = FALSEの状態、モジュールのすべての入力信号／入力値で使用されます。
- ▶ 装置上の影響を受けないFS入力モジュールは引き続き処理されます。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. エラーが回復されると、FS入力モジュールは自動的に再有効化されます。

11.5.2.5 装置のFS入力モジュールがモジュールエラーでSTOP状態

FS入力エラーはすべてのFS入力モジュールおよびPSS 4000専用装置のFS出力に影響します。FSモジュールバスはオペレーティングステータス「FSモジュールバスPSSuがRUN状態、エラーなし」を維持します。

考えられる原因

- ▶ 装置が、FS入力エラーを検出し、エラーの場所を特定できない (PSSu E F AI Uなど)。

影響

- ▶ 代入値は、有効ビット = FALSEの状態、すべての装置の入力信号／入力値で使用されます
- ▶ 装置上のすべてのFSハードウェア出力が安全状態に切り替わります。
- ▶ 代入値は、有効ビット = FALSEの状態、モジュールのFS出力で使用されます。
- ▶ 装置のステータスLED:
「MBUS」LEDが赤色に点灯します。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. リセットの実行
(ウォームリセット/リセットボタン、コールドリセット、プロジェクトのリセット)

11.5.2.6 FS出力モジュールがRUN状態、エラーなし

オペレーティングステータスは、モジュールバスのステータスに関わらずモジュールの動作準備が完了するとすぐに達成されます。FS出力モジュールは、FSモジュールバスPSSuが提供する出力データを処理します。

11.5.2.7 装置のすべてのFS出力が安全状態

エラーによりこのオペレーティングステータスになった場合、エラーの影響は、エラーが検出されたPSS 4000専用装置のFSハードウェア出力に制限されます。FSモジュールバスはオペレーティングステータス「FSモジュールバスPSSuがRUN状態、エラーなし」を維持します。

考えられる原因

- ▶ 出力エラー
接点間の短絡、短絡回路、出力テストのエラーなど
- ▶ 温度エラー
- ▶ 電圧エラー (モジュール電源)
- ▶ モジュールが動作中に切断された
- ▶ 欠陥のあるモジュールが動作中に検出された
- ▶ オペレーティングステータス「装置のFS入力モジュールがモジュールエラーでSTOP状態」

影響

- ▶ 装置上のすべてのFSハードウェア出力が安全状態に切り替わります。
- ▶ 代入値はモジュールのFS出力に使用されます。
- ▶ 装置のステータスLED:
「MBUS」LEDが赤色に点灯します。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. リセットの実行
(ウォームリセット/リセットボタン、コールドリセット、プロジェクトのリセット)

11.5.2.8 Major FS error

エラーによりこのオペレーティングステータスになった場合、FSモジュールバスPSSuの、影響を受けるセクションは定義された通りに終了します。定義された状態になります。

FSの重大エラーはPSS 4000専用装置のすべての安全関連部に影響します (「[オペレーティングステータス「FSの重大エラー」](#)を参照  344])。

11.5.3 FSモジュールバスPSSuのオペレーティングステータスの変更

オペレーティングステータスの変更に関する説明では、装置に有効な装置プロジェクトがあることを前提としています。数字は、図「オペレーティングステータスとFSモジュールバスPSSuのオペレーティングステータスの変更」(「フローチャート [371]」を参照) のオペレーティングステータスの変更に対応しています。



情報

オペレーティングステータスのそれぞれの変更に関する説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「リセット、再起動、起動、停止のオプション [424]」を参照)。

[1] 「システムセクションを起動中」から「FSモジュールバスPSSuがRUN状態、エラーなし」への切り替え
「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです (「フローチャート」を参照)。システムセクションの起動時に、安全関連システムセクションでエラーが見つからなかった場合、FSモジュールバスPSSuが起動され、ステータス「FSモジュールバスPSSuがRUN状態、エラーなし」になります。

[2a] 「システムセクションを起動中」から「FSの重大エラー」への切り替え

システムセクションの起動時に、安全関連システムセクションでエラーが見つかり、FSモジュールバスPSSuがオペレーティングステータス「FSの重大エラー」になります。

ステータス「システムセクションを起動中」とステータス「FSの重大エラー」(「フローチャート」を参照) はどちらもPSS 4000専用装置全体に影響するオペレーティングステータスです。

[2b] 「FSモジュールバスPSSuがRUN状態、エラーなし」から「FSの重大エラー」への切り替え

FSモジュールバスPSSuがRUN状態の間に安全関連システムセクションでエラーが見つかり、FSモジュールバスPSSuがオペレーティングステータス「FSの重大エラー」になります。

オペレーティングステータス「FSの重大エラー」(「フローチャート」を参照) はPSS 4000専用装置全体に影響するオペレーティングステータスです。

[3a] 「FS入力モジュールがRUN状態、エラーなし」から「FSモジュールがエラーでRUN状態」への切り替え

FSハードウェア入力に入力エラーが見つかり、FS入力モジュールがオペレーティングステータス「FSモジュールが入力エラーでRUN状態」になります。

[3b] 「FS入力モジュールがRUN状態、エラーなし」から「FS入力モジュールがモジュールエラーでSTOP状態」への切り替え

FS入力モジュールにエラーが見つかり、装置の関連するFS入力モジュールがオペレーティングステータス「FS入力モジュールがモジュールエラーでSTOP状態」になります。

[3c] 「FS入力モジュールがRUN状態、エラーなし」から「装置のFS入力モジュールがモジュールエラーでSTOP状態」への切り替え

FS入力エラーが見つかり、PSSu専用装置でそれをFSハードウェア入力のいずれかに一意に割り付けできない場合、装置のFS入力モジュールはすべてオペレーティングステータス「FS入力モジュールがモジュールエラーでSTOP状態」になり、装置のすべてのFS出力もオペレーティングステータス「装置のすべてのFS出力が安全状態」になります ([5] を参照)。

[4] 「FS入力モジュールが入力エラーでRUN状態」または

「FS入力モジュールがモジュールエラーでSTOP状態」のいずれかのオペレーティングステータスから「FS入力モジュールがRUN状態、エラーなし」への切り替え

関連するエラー（入力エラー、モジュールエラー）が回復されると、FS入力モジュールの、影響を受けるセクションが自動的に再有効化され、モジュールはオペレーティングステータス「FS入力モジュールがRUN状態、エラーなし」になります。

[5] 「FS出力モジュールがRUN状態」から「装置のすべてのFS出力が安全状態」への切り替え

FS出力モジュールまたはFS出力にエラーが見つかった場合、またはPSS 4000専用装置のFS入力モジュールがオペレーティングステータス「装置のFS入力モジュールがモジュールエラーでSTOP状態」になった場合、装置のFS出力はすべて安全状態に切り替わります。

[6a] 「FSモジュールバスPSSuがRUN状態、エラーなし」から「システムセクションを起動中」への切り替え

オペレーティングステータスでのこの変更は、PSS 4000専用装置のすべてのシステムセクションに影響を及ぼします（「フローチャート」を参照）。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[6b] 「装置のすべてのFS出力が安全状態」から「システムセクションを起動中」への切り替え

オペレーティングステータスでのこの変更は、PSS 4000専用装置のすべてのシステムセクションに影響を及ぼします（「フローチャート」を参照）。

異常を回復した後、オペレーティングステータスのこの変更は、次のコマンドのいずれかを使用してトリガできます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

11.6 オペレーティングステータスとSTモジュールバスPSSuのオペレーティングステータスの変化

以下の説明では、PSS 4000専用装置のSTモジュールバスPSSuのオペレーティングステータスのみを扱います。必要に応じて装置のオペレーティングステータスを参照します。

図「オペレーティングステータスとSTモジュールバスPSSuのオペレーティングステータスの変更」(「[フローチャート](#)  384」を参照) にオペレーティングステータス (グレー) の概要を示します。

- ▶ 角括弧内の数字はオペレーティングステータスのさまざまな変更を示します。
- ▶ ローマ数字「I」または「II」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (フローチャート)。その影響はSTモジュールバスPSSuだけではなく、PSS 4000専用装置のすべてのシステムセクションに及ぶためです。
- ▶ ローマ数字「Vb」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「フローチャート」を参照)。PSS 4000専用装置のシステムセクションを起動すると、STモジュールバスPSSuが起動するためです。

11.6.1 フローチャート

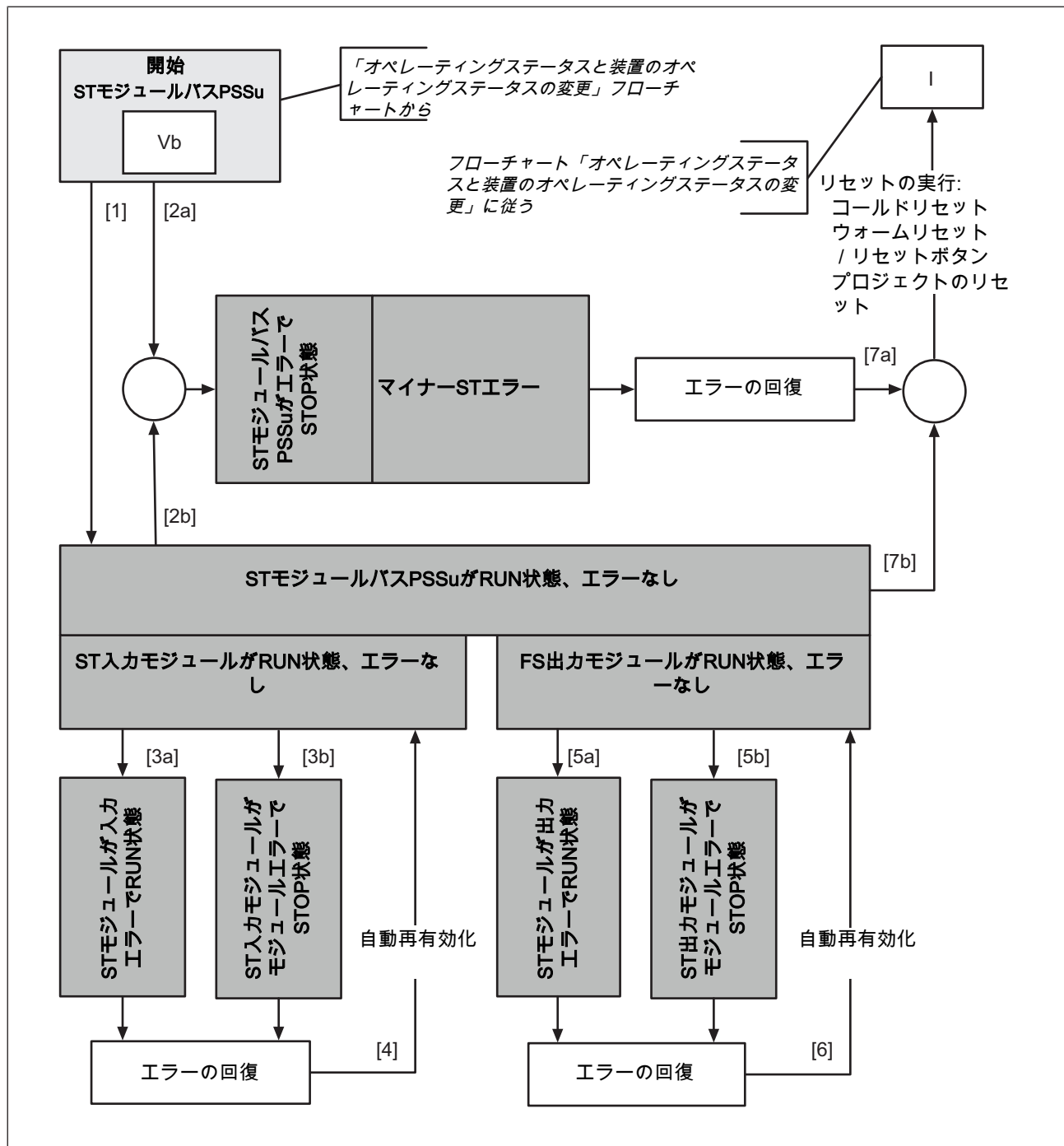


図: オペレーティングステータスとSTモジュールバスのオペレーティングステータスの変更

11.6.2 STモジュールバスPSSuのオペレーティングステータス

各種オペレーティングステータスの説明では、PSS 4000専用装置に有効な装置プロジェクトがあることを前提としています

STモジュールバスは次のオペレーティングステータスに変更します (「[フローチャート](#)」[\[384\]](#)を参照)。

▶ オペレーティングステータス「STモジュールバスPSSuがRUN状態、エラーなし」が以下を伴う場合

- 「ST入力モジュールがRUN状態、エラーなし」
- 「STモジュールが入力エラーでRUN状態」
- 「ST入力モジュールがモジュールエラーでSTOP状態」
- 「ST出力モジュールがRUN状態、エラーなし」
- 「STモジュールが出力エラーでRUN状態」
- 「ST出力モジュールがモジュールエラーでSTOP状態」

▶ オペレーティングステータス「STモジュールバスPSSuがエラーでSTOP状態」:

- 「マイナーSTエラー」



情報

各オペレーティングステータスの説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「[リセット、再起動、起動、停止のオプション](#)」[\[424\]](#)を参照)。

11.6.2.1 STモジュールバスPSSuがRUN状態、エラーなし

このオペレーティングステータスは、次のオプションのいずれかを通して達成されます。

- ▶ 以下にトリガされ、正常に起動プロセスが実行された後
 - 装置のコールドスタートまたは装置の再起動コマンド (PAS4000を使用)
 - 装置のリセットボタンの使用
 - リセットのトリガ (PAS4000を使用)
(コールドリセット、ウォームリセット、プロジェクトリセット)

影響

- ▶ STモジュールバスPSSuがエラーのないRUN状態にあります。
- ▶ STモジュールが正常に動作しています。つまり、
 - 入力の入力信号／入力値が正しく記録され、STモジュールバスPSSuを介してエラーなく送信されています。
 - 出力が正しく有効化されています。
- ▶ 装置のステータスLED:
FSモジュールバスPSSuもエラーなく動作している場合、[MBUS] LEDが緑色に点灯します。

11.6.2.2 ST入力モジュールがRUN状態、エラーなし

オペレーティングステータスは、モジュールバスのステータスに関わらずモジュールの動作準備が完了するとすぐに達成されます。STモジュールバスPSSuは、ST入力モジュールがヘッドモジュールヘデータを伝送するための入力データを供給する場所です。

11.6.2.3 STモジュールが入力エラーで実行状態

入力エラーは影響を受けるSTハードウェア入力にのみ影響します。STモジュールバスはオペレーティングステータス「STモジュールバスPSSuがRUN状態、エラーなし」を維持します。

考えられる原因

- ▶ 入力エラー

影響

- ▶ 代入値は、有効ビット = FALSEと共に入力信号／入力値で使用されます。
- ▶ ST入力モジュールはRUN状態を維持し、影響を受けないSTハードウェア入力は引き続き処理されます。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. エラーが回復されると、STハードウェア入力は自動的に再有効化されます。

11.6.2.4 ST入力モジュールがモジュールエラーでSTOP状態

モジュールのエラーは、影響を受けるST入力モジュールにのみ影響します。STモジュールバスはオペレーティングステータス「STモジュールバスPSSuがRUN状態、エラーなし」を維持します。

考えられる原因

- ▶ 温度エラー
- ▶ 電圧エラー (モジュール電源)
- ▶ モジュールが動作中に切断された
- ▶ 欠陥のあるモジュールが動作中に検出された

影響

- ▶ 代入値は、有効ビット = FALSEの状態、モジュールのすべての入力信号／入力値で使用されます。
- ▶ 装置上の影響を受けないST入力モジュールは引き続き処理されます。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. エラーが回復されると、ST入力モジュールは自動的に再有効化されます。

11.6.2.5 ST出力モジュールがRUN状態、エラーなし

このオペレーティングステータスは、モジュールバスのステータスに関わらずモジュールの動作準備が完了するとすぐに達成されます。ST出力モジュールは、STモジュールバスが提供する出力データを処理します。

11.6.2.6 STモジュールが出力エラーで実行状態

ST出力エラーは影響を受けるSTハードウェア出力にのみ影響します。STモジュールバスはオペレーティングステータス「STモジュールバスPSSuがRUN状態、エラーなし」を維持します。

考えられる原因

- ▶ 出力エラー
(過負荷、短絡など)

影響

- ▶ 代入値は、有効ビット = FALSEの状態、出力信号／出力値で使用されます。
- ▶ ST出力モジュールはRUN状態を維持し、影響を受けないSTハードウェア出力は引き続き処理されます。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. エラーが回復されると、STハードウェア出力は自動的に再有効化されます。

11.6.2.7 ST出力モジュールがモジュールエラーでSTOP状態

ST出力モジュールのモジュールエラーは、影響を受けるST出力モジュールにのみ影響します。STモジュールバスはオペレーティングステータス「STモジュールバスPSSuがRUN状態、エラーなし」を維持します。

考えられる原因

- ▶ 温度エラー
- ▶ 電圧エラー (モジュール電源)
- ▶ モジュールが動作中に切断された
- ▶ 欠陥のあるモジュールが動作中に検出された

影響

- ▶ 代入値は、有効ビット = FALSEの状態、モジュールのすべての出力信号／出力値で使用されます。
- ▶ 装置上の影響を受けないST出力モジュールは引き続き処理されます。
- ▶ 装置のステータスLED:
「MBUS」LEDが赤色に点灯します。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. エラーが回復されると、ST出力モジュールは自動的に再有効化されます。

11.6.2.8 Minor ST error

エラーによりこのオペレーティングステータスになった場合、STモジュールバスPSSuの、影響を受けるセクションは定義された通りに終了します。定義された状態になります。

考えられる原因

- ▶ ハードウェアレジストリエラー
(設定されたハードウェアレジストリが実際の値と一致しないなど)
- ▶ モジュールバスPSSu起動時に欠陥のあるSTモジュールが検出された

影響

- ▶ STモジュールバスPSSuの起動時にエラーが検出されると、ST出力が有効化されず、ST入力を読み取られません。
- ▶ 他の装置がこの装置のST I/Oデータにアクセスすると、有効ビット = FALSEの状態で、代入値がすべての信号／値に使用されます。

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど
2. エラーの回復
3. エラーを回復したら、任意のリセットコマンドで装置を再起動する必要があります。

11.6.3 STモジュールバスPSSuのオペレーティングステータスの変更

オペレーティングステータスの変更に関する説明では、装置に有効な装置プロジェクトがあることを前提としています。数字は、図「オペレーティングステータスとSTモジュールバスPSSuのオペレーティングステータスの変更」(図 [384] を参照) のオペレーティングステータスの変更に対応しています。



情報

オペレーティングステータスのそれぞれの変更に関する説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「リセット、再起動、起動、停止のオプション [424]」を参照)。

[1] 「システムセクションを起動中」から「STモジュールバスPSSuがRUN状態、エラーなし」への切り替え
「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです (図を参照)。システムセクションの起動時にエラーが見つからなかった場合、STモジュールバスPSSuが起動され、ステータス「STモジュールバスPSSuがRUN状態、エラーなし」になります。

[2a] 「システムセクションを起動中」から「マイナーSTエラー」への切り替え

システムセクションの起動時にエラーが見つかり、STモジュールバスPSSuがオペレーティングステータス「マイナーSTエラー」になります。

オペレーティングステータス「マイナーSTエラー」は、PSS 4000専用装置のSTモジュールバスPSSuにのみ影響するオペレーティングステータスです。

[2b] 「STモジュールバスPSSuがRUN状態、エラーなし」から「マイナーSTエラー」への切り替え

STモジュールバスPSSuがRUN状態の間にエラーが見つかり、STモジュールバスPSSuがオペレーティングステータス「マイナーSTエラー」になります。

オペレーティングステータス「マイナーSTエラー」は、PSS 4000専用装置のSTモジュールバスPSSuにのみ影響するオペレーティングステータスです。

[3a] 「ST入力モジュールがRUN状態、エラーなし」から「ST入力モジュールが入力エラーでRUN状態」への切り替え

STハードウェア入力に入力エラーが見つかり、ST入力モジュールがオペレーティングステータス「ST入力モジュールが入力エラーでRUN状態」になります。

[3b] 「ST入力モジュールがRUN状態、エラーなし」から「ST入力モジュールがモジュールエラーでSTOP状態」への切り替え

ST入力モジュールにモジュールエラーが見つかり、影響を受けるST入力モジュールがオペレーティングステータス「ST入力モジュールがモジュールエラーでSTOP状態」になります。

[4] 「ST入力モジュールが入力エラーでRUN状態」または

「ST入力モジュールがモジュールエラーでSTOP状態」のいずれかのオペレーティングステータスから「ST入力モジュールがRUN状態、エラーなし」への切り替え

関連するエラー (入力エラー、モジュールエラー) が回復されると、ST入力モジュールの、影響を受けるセクションが自動的に再有効化され、モジュールはオペレーティングステータス「ST入力モジュールがRUN状態、エラーなし」になります。

[5a] 「ST出力モジュールがRUN状態、エラーなし」から「ST出力モジュールが出力エラーでRUN状態」への切り替え

STハードウェア出力に出力エラーが見つかったと、ST出力モジュールがオペレーティングステータス「STモジュールが出力エラーでRUN状態」になります。

[5b] 「ST出力モジュールがRUN状態、エラーなし」から「ST出力モジュールがモジュールエラーでSTOP状態」への切り替え

ST出力モジュールにモジュールエラーが見つかったと、影響を受けるST出力モジュールがオペレーティングステータス「ST出力モジュールがモジュールエラーでSTOP状態」になります。

[6] 「ST出力モジュールが出力エラーでRUN状態」または「ST出力モジュールがモジュールエラーでSTOP状態」のいずれかのオペレーティングステータスから「ST出力モジュールがRUN状態、エラーなし」への切り替え
関連するエラー (出力エラー、モジュールエラー) が回復されると、ST出力モジュールの、影響を受けるセクションが自動的に再有効化され、モジュールはオペレーティングステータス「ST出力モジュールがRUN状態、エラーなし」になります。

[7a] 「マイナーSTエラー」から「システムセクションを起動中」への切り替え

オペレーティングステータスでのこの変更は、PSS 4000専用装置のすべてのシステムセクションに影響します (セクション「オペレーティングステータスと装置のオペレーティングステータスの変更」の図8-3を参照)。エラーを回復したら、次のいずれかのリセットコマンドでSTモジュールバスPSSuを再起動できます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[7b] 「STモジュールバスPSSuがRUN状態、エラーなし」から「システムセクションを起動中」への切り替え
オペレーティングステータスでのこの変更は、PSS 4000専用装置のすべてのシステムセクションに影響を及ぼします (図を参照)。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

11.7 オペレーティングステータスとFS SafetyNET pのオペレーティングステータスの変更

以下の説明では、PSS 4000専用装置のFS SafetyNET pシステムセクションのオペレーティングステータスのみを扱います。必要に応じて装置のオペレーティングステータスを参照します。

図「オペレーティングステータスとFS SafetyNET pのオペレーティングステータスの変更」(「[フローチャート](#)  397」を参照) にオペレーティングステータス (グレー) の概要を示します。

- ▶ 角括弧内の数字はオペレーティングステータスのさまざまな変更を示します。
- ▶ ローマ数字「I」または「II」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「[フローチャート](#)」を参照)。その影響はシステムセクションFS SafetyNET pだけではなく、PSS 4000専用装置のすべてのシステムセクションに及ぶためです。
- ▶ ローマ数字「VIa」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「[フローチャート](#)」を参照)。PSS 4000専用装置のシステムセクションを起動すると、システムセクションFS SafetyNET pが起動するためです。

11.7.1 フローチャート

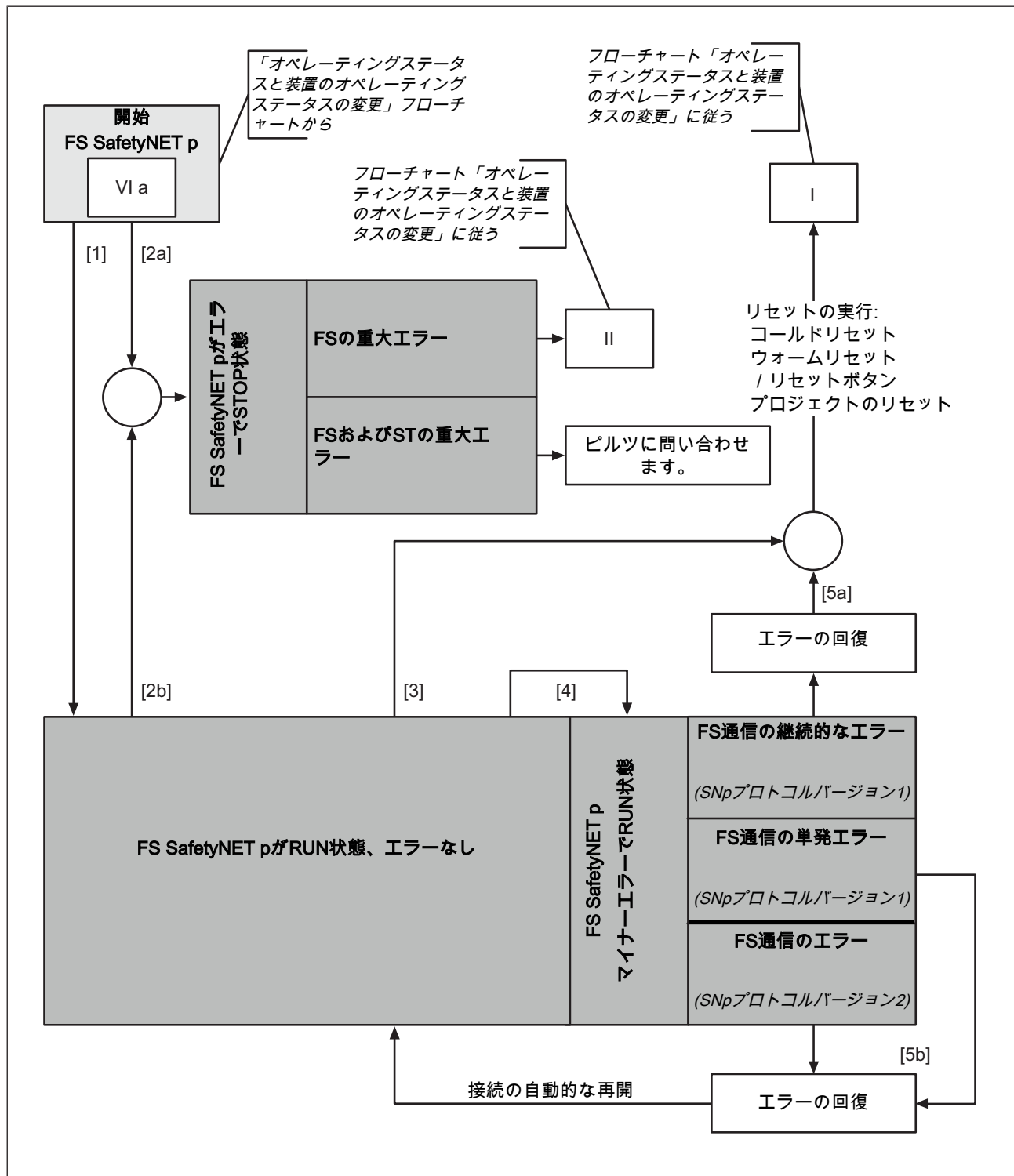


図: オペレーティングステータスとFS SafetyNET pのオペレーティングステータスの変更

11.7.2 FS SafetyNET pのオペレーティングステータス

各種オペレーティングステータスの説明では、PSS 4000専用装置に有効な装置プロジェクトがあることを前提としています

システムセクションFS SafetyNET pは、次のオペレーティングステータスに変更します（「[フローチャート \[397\]](#)」を参照）。

- ▶ オペレーティングステータス「FS SafetyNET pがRUN状態、エラーなし」
- ▶ オペレーティングステータス「FS SafetyNET pがエラーでSTOP状態」:
 - 「FSの重大エラー」
 - 「FSおよびSTの重大エラー」
- ▶ オペレーティングステータス「FS SafetyNET pがマイナーエラーでRUN状態」
 - SafetyNET pプロトコルバージョン1:
 - 「FN通信の継続的なエラー」
 - 「外部通信の単発エラー」
 - SafetyNET pプロトコルバージョン2:
 - FS通信の接続ベースエラー
 - FS通信の装置ベースエラー
 - FS通信の不特定エラー



情報

各オペレーティングステータスの説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します（「[リセット、再起動、起動、停止のオプション \[424\]](#)」を参照）。

11.7.2.1 FS SafetyNET pがRUN状態、エラーなし

このオペレーティングステータスは、次のオプションのいずれかを通して達成されます。

- ▶ 以下にトリガされ、正常に起動プロセスが実行された後
 - 装置のコールドスタートまたは装置の再起動コマンド (PAS4000を使用)
 - 装置のリセットボタンの使用
 - リセットのトリガ (PAS4000を使用)
(コールドリセット、ウォームリセット、プロジェクトリセット)

影響

- ▶ オペレーティングステータス「FS SafetyNET pがRUN状態、エラーなし」では、保留中のFSプロセスデータを SafetyNET pでエラーなく送信できます。
- ▶ 装置のステータスLED:
[FS SNp] LEDが緑色に点灯します。

11.7.2.2 Major FS error

エラーによりこのオペレーティングステータスになった場合、システムセクションFS SafetyNET pの、影響を受ける部分が定義された通りに終了します。定義された状態になります。

FSの重大エラーはPSS 4000専用装置のすべての安全関連部に影響します (「[オペレーティングステータス「FSの重大エラー」](#)を参照  344])。

11.7.2.3 Major FS+ST error

エラーによりオペレーティングステータスが「FSおよびSTの重大エラー」になった場合、その影響を受けるシステムセクションFS SafetyNET pの部分が定義された通りに終了します。定義された状態になります。

考えられる原因

- ▶ 内部エラー

影響

- ▶ システムセクションFS SafetyNET pおよびシステムセクションST SafetyNET pはどちらもSTOP状態に切り替わります。装置ではプロセスデータが交換できなくなります。
- ▶ 代入値は、有効ビット = FALSEの状態で、影響を受けるプロセスデータで使用されます。
- ▶ 装置のステータスLED:
「FS SNp」および「ST SNp」LEDが赤色に点滅します。

処置

ピルツに問い合わせます。

11.7.2.4 Snpプロトコルバージョン1: FS SafetyNET pインタフェースがマイナーエラーでRUN状態

エラーによりFS通信にマイナーエラーが発生した場合、影響を受けるセクションが定義された通りに停止または終了します。定義された状態になります。

考えられる原因

▶ FS通信の継続的なエラー

30秒以内に接続エラーが2回以上発生した場合、エラーは「継続的」と見なされます。

継続的なエラーの例:

- 装置がSafetyNET pから取り外されたため、デバイスと一時的に通信できません。
- 装置のバスコネクタが抜かれました。

▶ FS通信の単発故障

30秒以内に接続エラーが1回だけ発生した場合、エラーは「単発」と見なされます。

単発エラーの例:

- バスケーブルでの一時的な単発エラー

影響

- ▶ 受信側の装置とリモート装置間のFS通信が無効化されています。つまり、影響を受けるRxデータ領域でさらに処理できるプロセスデータが、装置に受け入れられていないか、無視されているということです。
- ▶ 他のすべての装置とのFSプロセスデータ交換は継続します。
- ▶ システムセクションFS SafetyNET pは引き続きRUN状態になります。
- ▶ 代入値は、有効ビット = FALSEの状態、影響を受けるプロセスデータで使用されます。
- ▶ 装置のステータスLED:
「FS Snp」LEDは緑色に点滅します。

処置

1. 診断

PAS4000を使用した診断リストまたは診断ログの読み取りなど

2. エラーの回復

3. 接続を自動的にリセットまたは再開

- FS通信での継続的なエラー:

FS SafetyNET p待機時間 (1 s) が経過し、エラーの原因が存在しなくなったら、リセット (ウォームリセット/リセットボタン、コールドリセット、プロジェクトリセット) を実行します。

- FS通信の単発エラー:

FS SafetyNET p待機時間 (1 s) が経過し、エラーの原因が存在しなくなると、FS通信は自動的に再開します。

FS通信が再開すると、出力でプロセスの値が出力されます。

11.7.2.5 Snpプロトコルバージョン2: FS SafetyNET pインタフェースがマイナーエラーでRUN状態

エラーによりFS通信にマイナーエラーが発生した場合、影響を受けるセクションが定義された通りに停止または終了します。定義された状態になります。

考えられる原因

- ▶ FS通信の接続ベースエラー

このエラーは、単一接続を介したFS通信に関係しています。

- ▶ FS通信の装置ベースエラー

このエラーは、2台の装置間のFS通信に関係しています。

- ▶ FS通信の不特定エラー

このエラーは、1台の装置とそのすべてのリモート装置間のFS通信に関係しています。エラーの原因であるリモート装置を特定できない場合、エラーは「不特定」になります。

例:

- ▶ 装置がSafetyNET pから取り外されたため、デバイスと一時的に通信できません。
- ▶ 装置のバスコネクタが抜かれました。
- ▶ SafetyBUS pインフラのエラー
- ▶ バスケーブルのエラー
- ▶ 電源容量不足
- ▶ 一貫性がない装置プロジェクト

影響

- ▶ エラーの種類

- FS通信の接続ベースエラー

接続は一時的に無効化されています。つまり、対応するRxデータ領域でさらに処理できるプロセスデータが、装置で受け入れられていないか、無視されているということです。

他の接続を介したFS通信は影響を受けません。

- FS通信の装置ベースエラー

受信側の装置とリモート装置間のFS通信が一時的に無効化されています。つまり、影響を受けるRxデータ領域でさらに処理できるプロセスデータが、装置で受け入れられていないか、無視されているということです。

他のリモート装置とのFS通信は影響を受けません。

- FS通信の不特定エラー

すべてのリモート装置とのFS通信は一時的に無効化されています。つまり、装置はリモート装置からこれ以上プロセスデータを受け付けません。

- ▶ システムセクションFS SafetyNET pは引き続きRUN状態になります。

- ▶ 代入値は、有効ビット = FALSEの状態、影響を受けるプロセスデータで使用されます。

- ▶ 装置のステータスLED:

「FS Snp」LEDは緑色に点滅します。

処置

1. 診断

PAS4000を使用した診断リストまたは診断ログの読み取りなど
診断メッセージは、どのエラーがあるかを特定するのに使用できます。

2. エラーの回復

3. FS通信の自動再開

FS SafetyNET p待機時間が経過し、エラーの原因が存在しなくなると、FS通信は自動的に再開します。FS通信が再開すると、出力でプロセスの値が出力されます。

11.7.3 FS SafetyNET pのオペレーティングステータスの変更

オペレーティングステータスの変更に関する説明では、装置に有効な装置プロジェクトがあることを前提としています。数字は、図「オペレーティングステータスとFS SafetyNET pのオペレーティングステータスの変更」(「フローチャート」[\[397\]](#))を参照)のオペレーティングステータスの変更に対応しています。



情報

オペレーティングステータスにおけるそれぞれの変更に関する説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「リセット、再起動、起動、停止のオプション」[\[424\]](#))を参照)。

[1] 「システムセクションを起動中」から「FS SafetyNET pがRUN状態、エラーなし」への切り替え

「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです (「フローチャート」を参照)。システムセクションの起動時に、安全関連システムセクションでエラーが見つからなかった場合、システムセクションFS SafetyNET pが起動され、ステータス「FS SafetyNET pがRUN状態、エラーなし」になります。

[2a] 「システムセクションを起動中」からエラーステータスへの切り替え

システムセクションの起動時に、安全関連システムセクションでエラーが見つかる (「フローチャート」を参照)、FS SafetyNET pはSTOP状態になり、エラーに応じてオペレーティングステータス「FSの重大エラー」または「FSおよびSTの重大エラー」になります。

[2b] 「FS SafetyNET pがRUN状態、エラーなし」からエラーステータスへの切り替え

FS SafetyNET pがRUN状態になっているときに、安全関連システムセクションでエラーが見つかる、FS SafetyNET pはSTOP状態になり、エラーに応じてオペレーティングステータス「FSの重大エラー」または「FSおよびSTの重大エラー」になります。

[3] 「FS SafetyNET pがRUN状態、エラーなし」から「システムセクションを起動中」への切り替え

オペレーティングステータスでのこの変更は、PSS 4000専用装置のすべてのシステムセクションに影響を及ぼします (「フローチャート」を参照)。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[4] 「FS SafetyNET pがRUN状態、エラーなし」から「FS SafetyNET pがマイナーエラーでRUN状態」への切り替え

オペレーティングステータスでのこの変更は、FS通信エラーが原因で発生しました。このエラーを招く異常は、次のように分類されます。

- ▶ SafetyNET pプロトコルバージョン1:
 - FS通信の継続的なエラー
 - FS通信の単発故障

▶ SafetyNET pプロトコルバージョン2:

- FS通信の接続ベースエラー:
- FS通信の装置ベースエラー:
- FS通信の不特定エラー

[5a] 「FS SafetyNET pがエラーでRUN状態」から「FS SafetyNET pがRUN状態、エラーなし」への切り替え

SafetyNET pプロトコルバージョン1のみ:

オペレーティングステータス「FS SafetyNET pがエラーでRUN状態」は、FS通信の継続的なエラーが原因で発生しました。エラーを回復したら、次のいずれかのリセットコマンドでシステムセクションを再起動できます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[5b] 「FS SafetyNET pがエラーでRUN状態」から「FS SafetyNET pがRUN状態、エラーなし」への切り替え

▶ SafetyNET pプロトコルバージョン1:

オペレーティングステータス「FS SafetyNET pがエラーでRUN状態」は、FS通信の一時的な単発エラーが原因で発生しました。FS SafetyNET p待機時間が経過し、異常が回復されるか、エラーが存在しなくなると、FS通信は自動的に再開します。

▶ SafetyNET pプロトコルバージョン2:

「FS SafetyNET pがエラーでRUN状態」への切り替えは、FS通信の接続ベースエラー、装置ベースエラー、または不特定エラーが原因で発生しました。FS SafetyNET p待機時間が経過し、異常が回復されるか、エラーが存在しなくなると、影響を受ける接続は自動的に再開します。

11.8 オペレーティングステータスとST SafetyNET pのオペレーティングステータスの変更

以下の説明では、PSS 4000専用装置のST SafetyNET pシステムセクションのオペレーティングステータスのみを扱います。必要に応じて装置のオペレーティングステータスを参照します。

図「オペレーティングステータスとST SafetyNET pのオペレーティングステータスの変更」(「[フローチャート \[408\]](#)」を参照) にオペレーティングステータス (グレー) の概要を示します。

- ▶ 角括弧内の数字はオペレーティングステータスのさまざまな変更を示します。
- ▶ ローマ数字「I」または「II」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「[フローチャート](#)」を参照)。その影響はシステムセクションST SafetyNET pだけではなく、PSS 4000専用装置のすべてのシステムセクションに及ぶためです。
- ▶ ローマ数字「VIb」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています (「[フローチャート](#)」を参照)。PSS 4000専用装置のシステムセクションを起動すると、システムセクションST SafetyNET pが起動するためです。

11.8.1 フローチャート

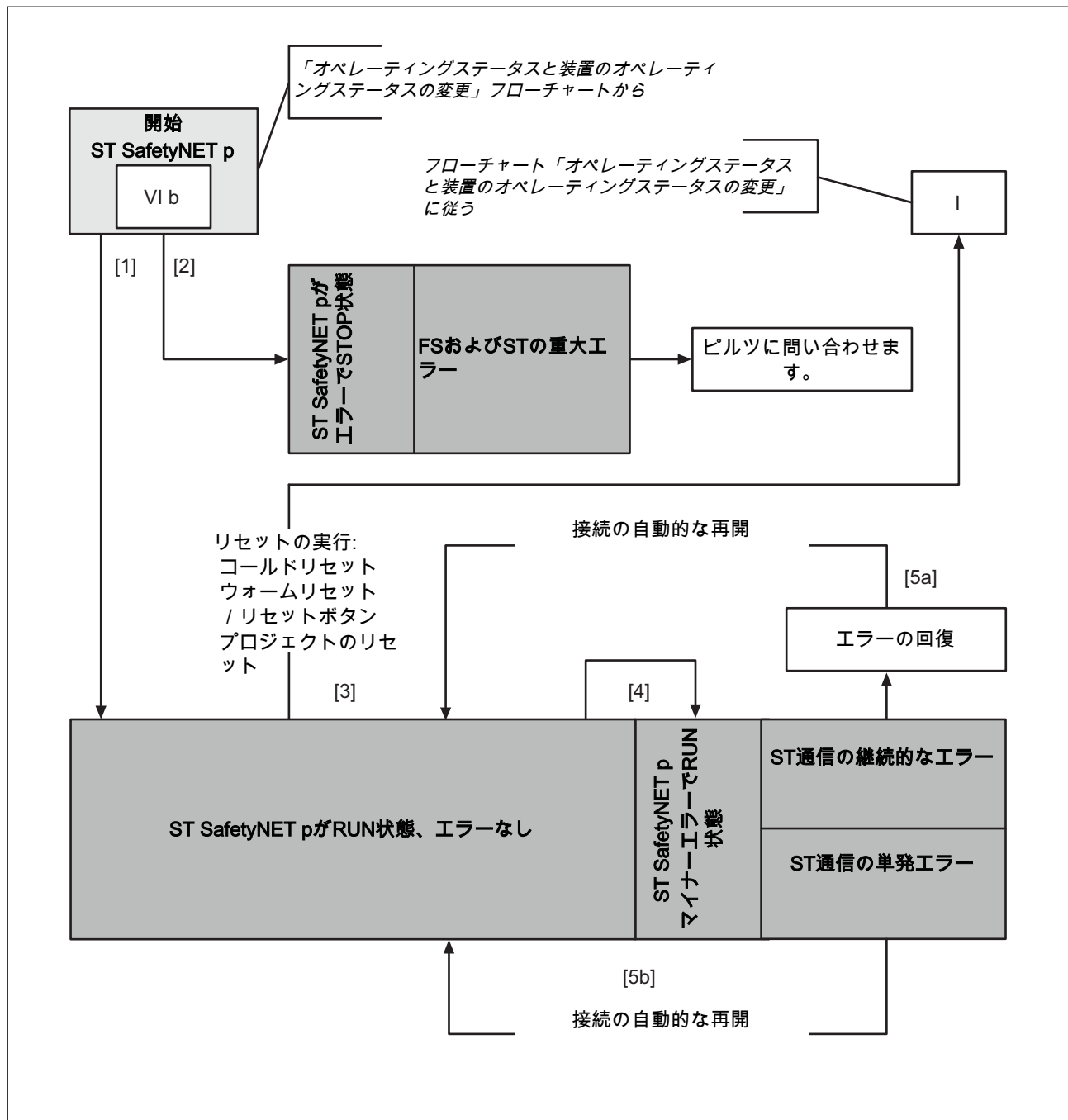


図: オペレーティングステータスとST SafetyNET pのオペレーティングステータスの変更

11.8.2 ST SafetyNET pのオペレーティングステータス

各種オペレーティングステータスの説明では、PSS 4000専用装置に有効な装置プロジェクトがあることを前提としています

システムセクションSafetyNET pは、次のオペレーティングステータスに変更します
(「[フローチャート](#)」 [408](#)」を参照)。

- ▶ オペレーティングステータス「ST SafetyNET pがRUN状態、エラーなし」
- ▶ オペレーティングステータス「ST SafetyNET pがエラーでSTOP状態」:
 - 「FSおよびSTの重大エラー」
- ▶ オペレーティングステータス「ST SafetyNET pがマイナーエラーでRUN状態」:
 - 「ST通信での継続的なエラー」
 - 「ST通信で一度限りのエラー」



情報

各オペレーティングステータスの説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「[リセット、再起動、起動、停止のオプション](#)」 [424](#)」を参照)。

11.8.2.1 ST SafetyNET pがRUN状態、エラーなし

このオペレーティングステータスは、次のオプションのいずれかを通して達成されます。

- ▶ 以下にトリガされ、正常に起動プロセスが実行された後
 - 装置のコールドスタートまたは装置の再起動コマンド (PAS4000を使用)
 - 装置のリセットボタンの使用
 - リセットのトリガ (PAS4000を使用)
(コールドリセット、ウォームリセット、プロジェクトリセット)

影響

- ▶ オペレーティングステータス「ST SafetyNET pがRUN状態、エラーなし」では、保留中のSTプロセスデータを SafetyNET pでエラーなく送信できます。
- ▶ 装置のステータスLED:
[ST SNp] LEDが緑色に点灯します。

11.8.2.2 Major FS+ST error

エラーによりオペレーティングステータスが「FSおよびSTの重大エラー」になった場合、その影響を受けるシステムセクションFS SafetyNET pの部分が定義された通りに終了します。定義された状態になります。

考えられる原因

- ▶ 内部エラー

影響

- ▶ システムセクションFS SafetyNET pおよびシステムセクションST SafetyNET pはどちらもSTOP状態に切り替わります。装置ではプロセスデータが交換できなくなります。
- ▶ 代入値は、有効ビット = FALSEの状態で、影響を受けるプロセスデータで使用されます。
- ▶ 装置のステータスLED:
「FS SNp」および「ST SNp」LEDが赤色に点滅します。

処置

ピルツに問い合わせます。

11.8.2.3 ST SafetyNET pがマイナーエラーでRUN状態

ST通信で軽度のエラーに至るような異常が発生した場合、影響を受けるセクションは定義通りに停止または終了されて、定義された状態になります。

考えられる原因

▶ ST通信の継続的なエラー

継続的なエラーの例:

- 装置がSafetyNET pから取り外されたため、デバイスと一時的に通信できません。
- 装置のバスコネクタが抜かれました。

▶ ST通信の単発エラー

単発エラーの例:

- バスケーブルでの一時的な単発エラー

影響

- ▶ STプロセスデータは、影響を受ける装置とは交換されなくなりますが、他のすべての装置とは引き続き交換されます。
- ▶ システムセクションST SafetyNET pは引き続きRUN状態になります。
- ▶ 代入値は、有効ビット = FALSEの状態で、影響を受けるプロセスデータで使用されます。
- ▶ 装置のステータスLED:
「FS SNp」LEDは緑色に点滅します。

処置

1. 診断

PAS4000を使用した診断リストまたは診断ログの読み取りなど

2. エラーの回復

3. 接続の再開

- ST通信での継続的なエラー:
接続は自動的に再開されます。
- ST通信での単発的な切断エラー:
接続は自動的に再開されます。

11.8.3 ST SafetyNET pのオペレーティングステータスの変更

オペレーティングステータスの変更に関する説明では、装置に有効な装置プロジェクトがあることを前提としています。数字は、図「オペレーティングステータスとST SafetyNET pのオペレーティングステータスの変更」(「フローチャート [408]」を参照) のオペレーティングステータスの変更に対応しています。



情報

オペレーティングステータスのそれぞれの変更に関する説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「リセット、再起動、起動、停止のオプション [424]」を参照)。

[1] 「システムセクションを起動中」から「ST SafetyNET pがRUN状態、エラーなし」への切り替え
「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです (「フローチャート」を参照)。システムセクションの起動時にエラーが見つからなかった場合、システムセクションST SafetyNET pが起動され、ステータス「ST SafetyNET pがRUN状態、エラーなし」になります。

[2] 「システムセクションを起動中」から「FSおよびSTの重大エラー」への切り替え
システムセクションの起動時にエラーが見つかる (「フローチャート」を参照)、システムセクションST SafetyNET pがSTOP状態になり、オペレーティングステータス「FSおよびSTの重大エラー」になります。

[3] 「ST SafetyNET pがRUN状態、エラーなし」から「システムセクションを起動中」への切り替え
オペレーティングステータスでのこの変更は、PSS 4000専用装置のすべてのシステムセクションに影響を及ぼします (「フローチャート」を参照)。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[4] 「ST SafetyNET pがRUN状態、エラーなし」から「ST SafetyNET pがマイナーエラーでRUN状態」への切り替え

オペレーティングステータスでのこの変更は、ST通信エラーが原因で発生しました。このエラーを招く異常は、次のように分類されます。

- ▶ ST通信の継続的なエラー
- ▶ ST通信の単発エラー

[5a] 「ST SafetyNET pがエラーでRUN状態」から「ST SafetyNET pがRUN状態、エラーなし」への切り替え

オペレーティングステータス「ST SafetyNET pがエラーでRUN状態」は、ST通信での継続的なエラーが原因で発生しました。異常が回復されると、接続は自動的に再開されます。

[5b] 「ST SafetyNET pがエラーでRUN状態」から「ST SafetyNET pがRUN状態、エラーなし」への切り替え

オペレーティングステータス「ST SafetyNET pがエラーでRUN状態」は、ST通信の一時的な単発エラーが原因で発生しました。接続は自動的に再開されます。

11.9 オペレーティングステータスと外部通信のオペレーティングステータスにおける変化

以下の説明では、外部通信でのオペレーティングステータスのみを扱います（「外部通信の通信モード」を参照）。オペレーティングステータスは、各システムセクション（システムセクション「IP接続」、システムセクション「EtherNet/IPアダプタ」など）で同一です。必要に応じて、PSS 4000専用装置のオペレーティングステータスに言及します。

図「オペレーティングステータスと外部通信とのオペレーティングステータスの変更」（「[フローチャート 415](#)」を参照）にオペレーティングステータス（グレー）の概要を示します。

- ▶ 角括弧内の数字はオペレーティングステータスのさまざまな変更を示します。
- ▶ ローマ数字「I」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています（「フローチャート」を参照）。その影響は各外部通信のシステムセクションだけではなく、PSS 4000専用装置のすべてのシステムセクションに及ぶためです。
- ▶ ローマ数字「VII」が付いたセクションは、図「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照しています（「フローチャート」を参照）。PSS 4000専用装置のシステムセクションを起動すると、各外部通信のシステムセクションが起動するためです。

11.9.1 フローチャート

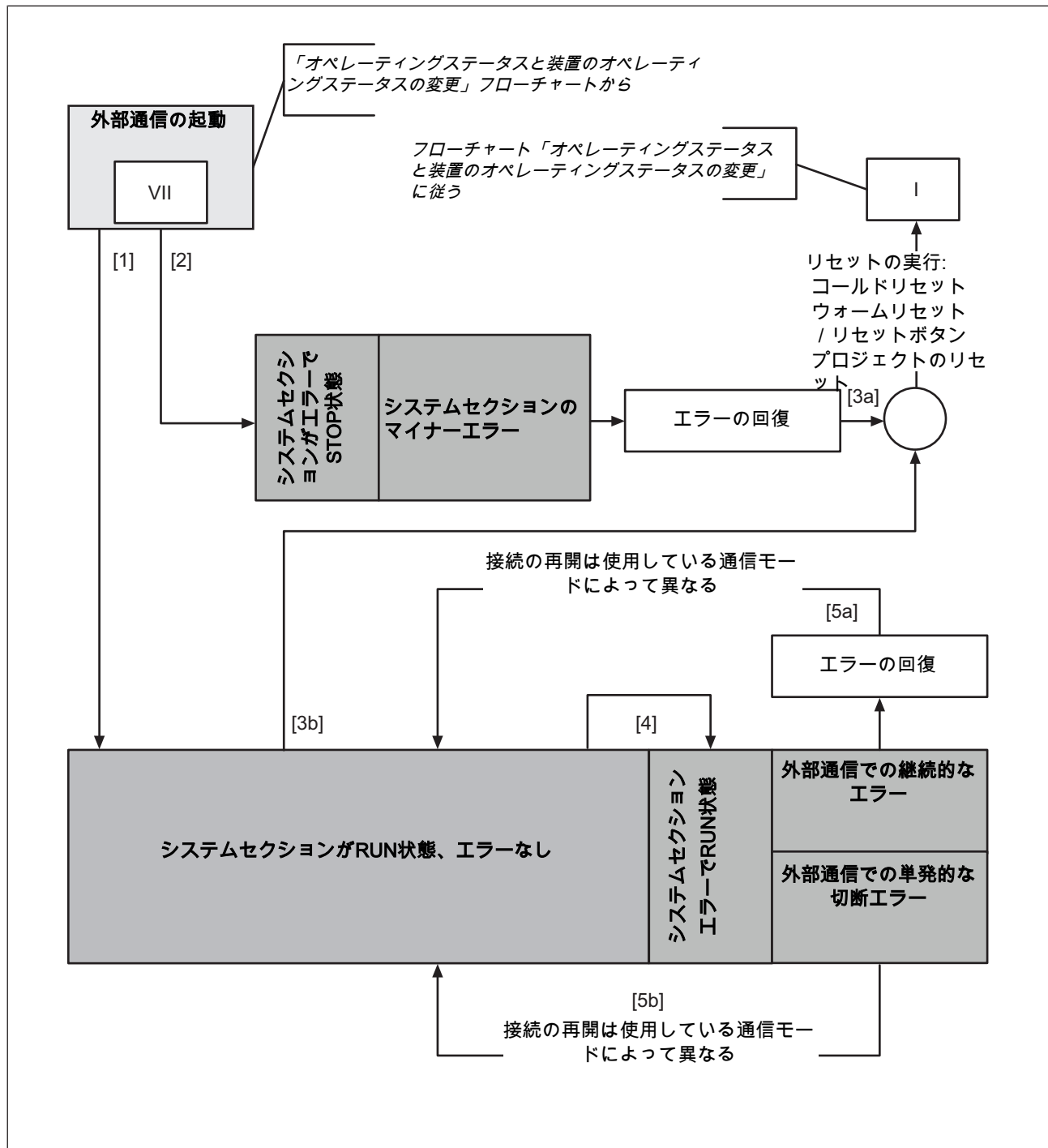


図: オペレーティングステータスと外部通信とのオペレーティングステータスの変更

11.9.2 外部通信でのオペレーティングステータス

各種オペレーティングステータスの説明では、PSS 4000専用装置に有効な装置プロジェクトがあることを前提としています

各外部通信 (例: Modbus/TCP、Raw UDP、EtherNet/IP) のシステムセクションは、次のオペレーティングステータスに変更します (「[フローチャート \[415\]](#)」を参照)。

- ▶ オペレーティングステータス「システムセクションがRUN状態、エラーなし」
- ▶ オペレーティングステータス「システムセクションがエラーでSTOP状態」
 - 「外部通信のシステムセクションにマイナーエラー」
- ▶ オペレーティングステータス「システムセクションがエラーでRUN状態」
 - 「外部通信での継続的なエラー」
 - 「外部通信での単発エラー」
- ▶ FSの重大エラー
(PROFIsafeなど、安全通信メディアを経由する通信に限られます)
- ▶ FSおよびSTの重大エラー
(PROFIsafeなど、安全通信メディアを経由する通信に限られます)



情報

- 各オペレーティングステータスの説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「[リセット、再起動、起動、停止のオプション \[424\]](#)」を参照)。
- 一部の通信モードでは、装置のステータスLEDで、さまざまなオペレーティングステータスを表示できます。ステータスLEDの存在とその意味は、使用する通信モードにより異なります。
例:
Modbus/TCP、Raw TCP、およびRaw UDPは、産業向けイーサネットに基づくプロトコルです。これらのプロトコルには、オペレーティングステータスを示すステータスLEDはありません。
PROFIBUS-DPスレーブでは、2つのステータスLED「SF」および「BF」でさまざまなステータスが表示されます。
PROFINET IO装置では、ステータスLED「BF」でさまざまなステータスが表示されます。
PROFIsafeでは、「SO」ステータスLEDでさまざまなステータスが表示されます。
EtherNet/IPアダプタでは、2つのステータスLED [MS] および [NS] でさまざまなステータスが表示されます。

11.9.2.1 システムセクションがエラーなしで実行状態

このオペレーティングステータスは、次のオプションのいずれかを通して達成されます。

- ▶ 以下にトリガされ、正常に起動プロセスが実行された後
 - 装置のコールドスタートまたは装置の再起動コマンド (PAS4000)
 - 装置のリセットボタンの使用
 - リセットのトリガ (PAS4000を使用)
(コールドリセット、ウォームリセット、プロジェクトリセット)

影響

- ▶ 各外部通信 (Modbus/TCP、Raw UDP、EtherNet/IPなど) のオペレーティングステータスが「システムセクションがRUN状態、エラーなし」のとき、保留中のプロセスデータは、各通信モードを使用してエラーなく送信されます。
- ▶ 装置のステータスLED:
使用している通信モードによって異なる

11.9.2.2 外部通信のシステムセクションにマイナーエラー

エラーによりこのオペレーティングステータスになった場合、システムセクションの、影響を受ける部分が定義された通りに終了します。定義された状態になります。

考えられる原因

- ▶ コンフィグレーションエラー

影響

- ▶ 各外部通信のシステムセクションがSTOP状態になります。対応する通信モードを使用して、プロセスデータを装置と交換できません。
- ▶ 代入値は、有効ビット = FALSEの状態、影響を受けるプロセスデータで使用されます。
- ▶ 装置のステータスLED:
使用している通信モードによって異なる

処置

1. 診断
PAS4000を使用した診断リストまたは診断ログの読み取りなど。
2. エラーの回復
通信相手のコンフィグレーションの同期など
3. リセットの実行

11.9.2.3 システムセクションがエラーで実行状態

エラーにより外部通信にエラーが発生した場合、影響を受けるシステムセクションが定義された通りに停止または終了します。定義された状態になります。

考えられる原因

▶ 外部通信での継続的なエラー

継続的なエラーの例:

- 通信相手が削除されたために、通信相手にアクセスできない。
- 装置のバスコネクタが抜かれた。

▶ 外部通信での単発的な切断エラー

単発エラーの例:

- バスケーブルでの一時的な単発エラー、テレグラムの損失

影響

- ▶ 処理データは通信相手と交換されなくなります。
- ▶ システムセクションは引き続きRUN状態になります。
- ▶ 代入値は、有効ビット = FALSEの状態、影響を受けるプロセスデータで使用されます。
- ▶ 装置のステータスLED:

使用している通信モードによって異なる

処置

1. 診断

PAS4000を使用した診断リストまたは診断ログの読み取りなど

2. エラーの回復

3. 接続の再開

接続を再開するオプションは、使用する通信モードにより異なります。また、多くの場合、通信相手のコンフィグレーションオプションによっても異なります。

接続の再開が通信モードでサポートされる場合、対応するシステムセクションは自動的に接続を再開します。それ以外の場合は、次のいずれかのリセットコマンドでシステムセクションを再起動する必要があります。

- ウォームリセット (PAS4000) またはリセットボタン (装置)
- コールドリセット (PAS4000)
- プロジェクトのウォームリセット (PAS4000)
- プロジェクトのコールドリセット (PAS4000)



情報

次の点に注意してください。

- 状況によっては、通信相手を再起動する必要があります。
- 通信相手が原因で異常が発生した場合があります。この場合は、通信相手側のエラーを回復してからでなければ、装置のシステムセクションで接続を復旧できません。

11.9.2.4

Major FS+ST error

このオペレーティングステータスは、システムセクションに安全関連の通信チャンネルがある場合にのみ発生します (たとえばPROFIsafeシステムセクションを備えたPROFIBUS-DPスレーブのPROFIsafeなど)。エラーによりこのオペレーティングステータスになった場合、影響を受けるシステムセクションは、定義された通りに終了します。

考えられる原因

- ▶ 内部エラー

影響

- ▶ システムセクション (PROFIsafeを備えたPROFIBUS-DPスレーブなど) はSTOP状態に切り替わります。
- ▶ 代入値は、有効ビット = FALSEの状態で、影響を受けるプロセスデータで使用されます。

処置

ピルツに問い合わせます。

11.9.2.5 Major FS error

このオペレーティングステータスは、システムセクションに安全関連の通信チャンネルがある場合にのみ発生します (たとえばPROFIsafeシステムセクションを備えたPROFIBUS-DPスレーブのPROFIsafeなど)。エラーによりこのオペレーティングステータスになった場合、システムセクションの安全関連通信チャンネルを介した通信は、定義された通りに終了します (システムセクションPROFIsafe搭載PROFIBUS-DPスレーブのPROFIsafeなど)。定義された状態になります。

FSの重大エラーはPSS 4000専用装置のすべての安全関連部に影響します (「[オペレーティングステータス「FSの重大エラー」](#)を参照  354)。

11.9.3 外部通信のオペレーティングステータスにおける変化

オペレーティングステータスの変更に関する説明では、装置に有効な装置プロジェクトがあることを前提としています。数字は、図「オペレーティングステータスと外部通信のオペレーティングステータスの変更」(「[フローチャート \[415\]](#)」を参照) のオペレーティングステータスの変更に対応しています。



情報

オペレーティングステータスのそれぞれの変更に関する説明では、オペレーティングステータスの変更をトリガするオプションのリストを示します。これらのオプションについては、個別に説明します (「[リセット、再起動、起動、停止のオプション \[424\]](#)」を参照)。

[1] 「システムセクションを起動中」から「システムセクションがRUN状態、エラーなし」への切り替え

「システムセクションを起動中」は、PSS 4000専用装置のオペレーティングステータスの1つです (「[フローチャート](#)」を参照)。システムセクションの起動時にエラーが見つからなかった場合、各外部通信のシステムセクションが起動され、ステータス「システムセクションがRUN状態、エラーなし」になります。

[2] 「システムセクションを起動中」から「外部通信のシステムセクションにマイナーエラー」への切り替え

システムセクションの起動時にエラーが見つかる (「[フローチャート](#)」を参照)、システムセクションはSTOP状態になり、オペレーティングステータス「外部通信のシステムセクションにマイナーエラー」になります。

[3a] 「外部通信のシステムセクションにマイナーエラー」から「システムセクションを起動中」への切り替え

エラーを回復した後、外部通信のシステムセクションは、任意のリセットコマンドで再起動できます。オペレーティングステータスでのこの変更は、装置のすべてのシステムセクションに影響を及ぼします (「[フローチャート](#)」を参照)。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[3b] 「システムセクションがRUN状態、エラーなし」から「システムセクションを起動中」への切り替え

オペレーティングステータスでのこの変更は、装置のすべてのシステムセクションに影響を及ぼします (「[フローチャート](#)」を参照)。

オペレーティングステータスでのこの変更は、次のいずれかのコマンドを使用してトリガできます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[4] 「システムセクションがRUN状態、エラーなし」から「システムセクションがエラーでRUN状態」への切り替え

オペレーティングステータスでのこの変更は通信エラーが原因で発生しました。このエラーを招く異常は、次のように分類されます。

- ▶ 外部通信での継続的なエラー
- ▶ 外部通信での単発的な切断エラー

[5a] 「システムセクションがエラーでRUN状態」から「システムセクションがRUN状態、エラーなし」への切り替え

オペレーティングステータス「システムセクションがエラーでRUN状態」は、外部通信の継続的なエラーが原因で発生しました。エラーの回復後、接続を再開するオプションは、使用する通信モードにより異なります。また、多くの場合、通信相手のコンフィグレーションオプションによっても異なります。

エラーの実際の原因が通信相手にない場合、外部通信のシステムセクションは、次のいずれかのコマンドで再起動できます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

[5b] 「システムセクションがエラーでRUN状態」から「システムセクションがRUN状態、エラーなし」への切り替え

オペレーティングステータス「システムセクションがエラーでRUN状態」は、外部通信の一時的な単発エラーが原因で発生しました。接続を再開するオプションは、使用する通信モードにより異なります。また、多くの場合、通信相手のコンフィグレーションオプションによっても異なります。

エラーの実際の原因が通信相手にない場合、外部通信のシステムセクションは、次のいずれかのコマンドで再起動できます。

- ▶ ウォームリセット (PAS4000) またはリセットボタン (装置)
- ▶ コールドリセット (PAS4000)
- ▶ プロジェクトのウォームリセット (PAS4000)
- ▶ プロジェクトのコールドリセット (PAS4000)

11.10 リセット、再起動、起動および停止

PSS 4000は、「段階的縮小」の原則に従って動作します。段階的縮小の原則の目的は、可能であれば、制御プロセスへの介入（異常発生時など）の影響を関連領域だけに制限することです。可能であれば、関連しない領域は中断されることなく、かつ安全機能に影響を与えることなく運転し続けるようにします。

たとえば試運転時には、特定の設備／機械セクションを起動／停止するためのさまざまなオプションを利用できます。制御プロセスへの介入の影響は、使用する特定のコマンドによって異なります。

11.10.1 概要

11.10.1.1 リセット、再起動、起動、停止のオプション

リセット、再起動、起動、停止オプションの概要とPAS4000の対応するコマンドを示すテーブル:

オプション	PAS4000
装置のコールドスタート (装置のすべてのシステムセクションに影響)	- - -
プロジェクトの再起動 (プロジェクトのすべての装置のすべてのシステムセクションに影響)	再起動
装置の再起動 (装置のすべてのシステムセクションに影響)	再起動
プロジェクトのリセット (プロジェクトのすべての装置のすべてのシステムセクションに影響)	コールドリセット
	ウォームリセット
装置のリセット (装置のすべてのシステムセクションに影響)	コールドリセット
	ウォームリセット
装置のリセットボタンを使用したリセット	- - -
プロジェクトの再起動 (すべての装置のすべてのリソースに影響)	コールド再起動
	ウォーム再起動
リソースの再起動 (装置のリソースに影響)	FSコールド再起動
	STコールド再起動
	FSウォーム再起動
	STウォーム再起動
リソースでタスクを開始 (リソースの1つまたは複数のタスクに影響)	FSホットスタート
	STホットスタート
プロジェクト停止 (プロジェクトのすべての装置のすべてのリソースに影響)	停止
装置の停止 (装置のすべてのリソースに影響)	停止
リソースの停止 (装置のリソースに影響)	FS停止
	ST停止

11.10.1.2 変数とアプリケーションパラメータへの影響

以下のテーブルには、ローカル変数 (VAR)、リソースグローバル変数 (VAR_GLOBAL。出力PI変数ではありません)、およびアプリケーションパラメータでのリセット、再起動、起動、停止の影響の概要を記載しています。

記載したコマンドのいずれかを実行した後、影響を受けるセクション (リソース、システムセクションのタスクなど) は、次のような変数値で起動されます。

操作	属性RETAIN PERSISTENT を持つ変数	変数 非保持 [1]	アプリケーションパラメータ
コールドスタート 再起動	保存済みの値	変数は初期値に設定される	アプリケーションパラメータに設 定された値
ウォーム再起動 ウォームリセット			
コールド再起動 コールドリセット	変数は初期値に設定される		
ホットスタート	保存済みの値		
停止	保存済みの値は維持される		

[1] 保持されない変数がアプリケーションパラメータとして使用されている場合、「アプリケーションパラメータ」列が適用されます。

ヒント

- ▶ 属性がRETAIN PERSISTENTである変数の内容はタスクの実行後に保存され、電源障害が発生した場合または再起動された場合でも保持されます。
- ▶ 属性がRETAIN PERSISTENTである変数の保存された内容は、電源障害が発生する前の、最後から2番目のタスク実行から取得できます。
複数の変数が保存される場合、一部の値が2番目のタスク実行から取得され、別の値が最後の実行から取得されます。

11.10.1.3 アクティブな「変数強制」への影響

アクティブな「変数の強制」に対するリセット、再起動、起動、停止の影響の概要を示すテーブル:

操作	「変数の強制」への影響
コールドスタート 再起動	「変数の強制」が停止 その後「変数の強制」を手動で開始する必要がある
ウォーム再起動 ウォームリセット	
コールド再起動 コールドリセット	
ホットスタート	「変数の強制」が停止 その後「変数の強制」がシステムで自動的に開始される
停止	「変数の強制」が停止 「変数の強制」を再起動する方法は、その後の (再) 起動アクション／リセット アクションに応じて異なる - ウォーム再起動／ウォームリセット - コールド再起動／コールドリセット - ホットスタート

11.10.1.4 「変数の制御」への影響

「変数の制御」は1度だけ実行されます。リセット、再起動、起動または停止コマンドの影響があるのは、「変数の制御」がまだ実行されていない場合に限られます。

「変数の制御」に対する、リセット、再起動、起動、停止による影響の概要を示すテーブル:

操作	「変数の制御」への影響
コールドスタート 再起動	まだ実行されていない「変数の制御」アクションは削除
ウォーム再起動 ウォームリセット	
コールド再起動 コールドリセット	
ホットスタート	まだ実行されていない「変数の制御」アクションはホットスタート後に実行
停止	「変数の制御」は実行されない 「変数の制御」アクションが実行されるかどうかは、その後の (再) 起動アクション／リセットアクションに応じて異なる - ウォーム再起動／ウォームリセット - コールド再起動／コールドリセット - ホットスタート

11.10.1.5 有効なオンライン変更時の影響

オンライン変更がアクティブな場合のリセット、再起動、起動、停止の影響の概要を示すテーブル:

アクション	有効なオンライン変更の影響
コールドスタート 再起動	オンライン変更は無効
ウォーム再起動 ウォームリセット	
コールド再起動 コールドリセット	
ホットスタート	オンライン変更は引き続き有効
停止	オンライン変更は引き続き有効 その後の影響は、以降の (再) 起動操作 / リセット操作によって決まります。 -ウォーム再起動 / ウォームリセット -コールド再起動 / コールドリセット -ホットスタート

11.10.2 コールドスタートと再起動コマンド

コールドスタート (モジュール電源の供給電圧をオフにしてオンにする) と再起動コマンドの影響は装置ベースです。すなわち、PSS 4000専用装置のすべてのシステムセクション (FSリソース、STリソースなど) に影響します。

手順

コールドスタート／再起動コマンドの実行で開始される手順:

1. 処理の中止
2. 装置がオペレーティングステータス「ベースシステムを起動中」になる
3. オペレーティングステータス「システムセクションを起動中」になり、すべてのシステムセクションが起動する
4. 診断ログに運転エントリが記録される
5. 影響
変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#)」[\[426\]](#)を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#)」[\[427\]](#)」を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#)」[\[429\]](#)」を参照) も別々に記載します。

コールドスタート／再起動コマンド実行中の動作

以下のテーブルに、処理データとシステムセクションの動作を示します。

処理データ、システムセクション		動作	
プロセスデータ	モジュールバスの入力データとI/Oマッピング [1]	ビットモジュール	入力 = 現在の物理入力信号
		バイトモジュール	入力 = 現在の物理入力値
	モジュールバスの出力データ	ビットモジュール	出力 = ゼロ
		バイトモジュール	出力 = スイッチオン値 (モジュールの取扱説明書を参照)
	出力PI変数とI/Oマッピング [1]	出力PI変数が無効としてマークされる	
システムセクション	FSリソース	「手順」を参照	
	STリソース		
	FSモジュールバスPSSu		
	STモジュールバスPSSu		
	FS SafetyNET p		
	ST SafetyNET p		
	外部通信 例: 「IP接続」システムセクション、 「PROFIBUS-DPスレーブ」システムセクション		

[1]

他の装置によってアクセスされたプロセスデータは、他の装置の代入値、有効ビット = FALSEに設定されます (「[プロセスデータの有効性](#)」 281」を参照)。

11.10.3 リセットコマンド

リセット (コールドリセット、ウォームリセット) の影響は装置ベースです。すなわち、PSS 4000専用装置のすべてのシステムセクション (FSリソース、STリソースなど) に影響します。コールドリセットとウォームリセットの主な違いは、不揮発性変数の処理方法です。

コールドリセットは、変数に関する定義済みの出力ステータスを装置に適用する場合に使用します。

ウォームリセットは、不揮発性変数に関する装置での目的の動作がコールドスタート／再起動コマンドと同じ場合に使用します。

使用可能なリセット:

▶ コールドリセット

- プロジェクトのすべての装置のコールドリセット
- 装置のコールドリセット

▶ ウォームリセット:

- プロジェクトのすべての装置のウォームリセット
- 装置のウォームリセット
(PAS4000またはリセットボタン)

手順

リセット実行時に開始される処理:

1. すべてのシステムセクションの停止

- プロジェクトのリセット:
プロジェクトのすべての装置上ですべてのリソースおよび他のすべてのシステムセクションのタスクが最後まで実行され、それ以降の実行は停止されます。
- 装置のリセット:
すべてのリソースおよび装置の他のすべてのシステムセクションのタスクが最後まで実行され、それ以降の実行は停止されます。

2. システムセクションの起動

- プロジェクトのリセット:
プロジェクトの装置はすべて、オペレーティングステータス「システムセクションを起動中」に切り替わります。これには、すべてのシステムセクションと、リセット前にRUN状態またはタスクエラーが発生していたすべてのリソースのすべてのタスクの起動も含まれます。
- 装置のリセット:
選択された装置はオペレーティングステータス「システムセクションを起動中」に切り替わります。これには、すべてのシステムセクションと、リセット前にRUN状態またはタスクエラーが発生していたすべてのリソースのすべてのタスクの起動も含まれます。

3. 診断ログ

- プロジェクトのリセット:
プロジェクトの装置ごとに、運転エントリが診断ログに記録されます。

– 装置のリセット:

選択した装置の運転エントリが診断ログに記録されます。

4. 診断リスト

– プロジェクトのリセット:

診断リストが削除されます。

– 装置のリセット:

選択した装置のメッセージが診断リストから削除されます。

5. 影響

変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#)」[\[426\]](#)を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#)」[\[427\]](#)を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#)」[\[429\]](#)を参照) も別々に記載します。

リセット中の動作

以下のテーブルに、リセット中の処理データとシステムセクションの動作を示します。

処理データ、システムセクション		動作	
プロセスデータ	モジュールバスの入力データとI/Oマッピング [1]	ビットモジュール	入力 = 現在の物理入力信号
		バイトモジュール	入力 = 現在の物理入力値
	モジュールバスの出力データ	ビットモジュール	出力 = ゼロ
		バイトモジュール	出力 = スイッチオン値 (モジュールの取扱説明書を参照)
	出力PI変数とI/Oマッピング [1]	出力PI変数が無効としてマークされる	
システムセクション	FSリソース	「手順」を参照	
	STリソース		
	FSモジュールバスPSSu		
	STモジュールバスPSSu		
	FS SafetyNET p		
	ST SafetyNET p		
	外部通信 例: 「IP接続」システムセクション、 「PROFIBUS-DPスレーブ」システムセクション		

[1]

他の装置によってアクセスされたプロセスデータは、他の装置の代入値、有効ビット = FALSEに設定されます (「[プロセスデータの有効性](#)」[\[281\]](#)を参照)。

11.10.4 ホットスタートコマンド

ホットスタート (FSホットスタート、STホットスタート) は、PSS 4000専用装置の以下に示すシステムセクションに影響を与えます。

- ▶ FSホットスタート: FSリソースに属する影響を受けたタスク
- ▶ STホットスタート: STリソースに属する影響を受けたタスク

ホットスタートでは、プロセスは以前に停止したポイント (実行時エラーまたは停止コマンドによる) から続行されます。ホットスタートは、診断リストに影響を与えません。つまり、エントリは維持されます。

ホットスタートは、試運転などに適しています。いずれかの停止コマンド (「[停止コマンド](#)」 437]」を参照) を使用して対象のサブプロセスを停止し (調整時やステップごとの試運転時など)、その後ホットスタートで続行することができます。

オペレーティングステータス「リソースがタスクエラーでRUN状態」でのホットスタート

FS/STリソースが「FS/STリソースがタスクエラーでRUN状態: タスクがSTOP状態」である場合の、FS/STホットスタートの手順

前提条件: エラーが回復されている

1. 運転エントリが診断ログに記録される
2. FS/STリソースの影響を受けたタスクが起動される

エラーによって実行されなくなったリソースタスクのみが起動されます。ホットスタートは、エラーなく正常に実行されているFS/STリソースのタスクに影響を与えません。

3. 影響
変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#)」 426]」を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#)」 427]」を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#)」 429]」を参照) も別々に記載します。

停止コマンド後のホットスタート

停止コマンド後のFS/STホットスタートの手順:

前提条件: タスク異常が発生していないこと

1. 運転エントリが診断ログに記録される
2. FS/STリソースの影響を受けたタスクが起動される

選択したリソースのすべてのタスクが起動されるユーザプログラムは、停止コマンドにより以前に停止したポイントから続行されます。

3. 影響
変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#)」 426]」を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#)」 427]」を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#)」 429]」を参照) も別々に記載します。

11.10.5 再起動コマンド

再起動 (コールド再起動、ウォーム再起動) の影響はリソースベースです。すなわち、PSS 4000専用装置の選択したリソース (ST/FSリソース) に影響します。コールド再起動とウォーム再起動の主な違いは、再起動の後、リソースの再起動に使用する不揮発性変数の扱い方です。

コールド再起動は、変数に関する定義済みの出力ステータスをリソース (FSリソース／STリソース) に適用する場合に使用します。

ウォーム再起動は、不揮発性変数に関するリソース (FSリソース／STリソース) の目的の動作がコールドスタート／再起動コマンドと同じ場合に使用します。

使用可能な再起動:

- ▶ コールド再起動: プロジェクト
- ▶ ウォーム再起動: プロジェクト
- ▶ FSコールド再起動: 装置のFSリソース
- ▶ STコールド再起動: 装置のSTリソース
- ▶ FSウォーム再起動: 装置のFSリソース
- ▶ STウォーム再起動: 装置のSTリソース

再起動の前には必ず自動停止 (ST/FSリソースのエラー) または手動停止 (停止コマンド) が行われるため、関連するプロセスデータとシステムセクションの動作は、リソースのSTOP状態によって決まります。リソースがSTOP状態の間、プロセスデータとシステムセクションは常に同じように動作するので、STOP状態へのトリガがどのようにかかったかは無関係です (「[停止コマンド – STOP状態中の動作](#)」 437]」を参照)。

オペレーティングステータス「リソースがRUN状態、エラーなし」での再起動

選択したリソースがRUN状態でエラーが発生していない場合の再起動の手順:

1. リソースが停止される
 - プロジェクトの装置上でのリソースの再起動:
プロジェクトのすべての装置で、リソース (FSおよびSTリソース) が最後まで実行され、それ以降の実行は停止されます。
 - 装置上でのリソースの再起動:
装置上で選択したリソース (FS/STリソース) が最後まで実行され、それ以降の実行は停止されます。
2. 診断ログ
 - プロジェクトの再起動:
プロジェクトのリソースごとに、運転エントリを診断ログに記録
 - 装置上でのリソースの再起動:
選択したリソースの運転エントリを診断ログに記録
3. 診断リスト
 - プロジェクトの再起動:
すべてのリソースのメッセージが診断リストから削除

- 装置上でのリソースの再起動:
選択したリソースのメッセージを診断リストから削除
- 4. リソースの起動
 - プロジェクトのコールド／ウォーム再起動:
プロジェクトのすべての装置上でリソース (FSおよびSTリソース) が起動されます。
 - 装置上でのリソースのコールド／ウォーム再起動:
装置上で選択したリソース (FS/STリソース) が起動されます。
- 5. 影響
変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#) [426]」を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#) [427]」を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#) [429]」を参照) も別々に記載します。

停止コマンド／エラー後の再起動

停止コマンド／エラー後の再起動の手順:

エラー時の前提条件: エラーが回復されている

1. 診断ログ
 - プロジェクトの再起動:
プロジェクトのリソースごとに、運転エントリを診断ログに記録
 - 装置上でのリソースの再起動:
選択したリソースの運転エントリを診断ログに記録
2. 診断リスト
 - プロジェクトの再起動:
すべてのリソースのメッセージが診断リストから削除
 - 装置上でのリソースの再起動:
選択したリソースのメッセージを診断リストから削除
3. リソースの起動
 - プロジェクトの再起動:
プロジェクトのすべての装置上でリソース (FSおよびSTリソース) が起動されます。
 - 装置のコールド／ウォーム再起動:
選択したリソース (FS/STリソース) が起動されます。
4. 影響
変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#) [426]」を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#) [427]」を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#) [429]」を参照) も別々に記載します。

11.10.6 停止コマンド

次の停止コマンドを使用できます。

▶ リソースの停止

コマンドはリソースベースで、装置上で選択したリソースに影響します。

- FSリソースの停止: 装置のFSリソースを停止
- STリソースの停止: 装置のSTリソースを停止

▶ 装置停止

コマンドは装置ベースで、選択した装置のすべてのリソース (FSリソース、STリソース) に影響します。

▶ プロジェクトの停止

コマンドは装置ベースで、プロジェクトのすべての装置の、すべてのリソース (FSリソース、STリソース) に影響します。

リソース停止の手順

装置上で選択したリソース (FSリソース／STリソース) を停止すると、次の処理が開始されます。

1. 選択したリソースのすべてのタスクが停止される

選択したリソースで処理される変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#)」 426」を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#)」 427」を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#)」 429」を参照) も別々に記載します。

2. 選択したリソースの運転エントリを診断ログに記録

3. メッセージが診断リストに入力される

装置停止の手順

装置を停止すると、次の処理が開始されます。

1. 選択した装置のすべてのリソースが停止される

選択したリソースで処理される変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#)」 426」を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#)」 427」を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#)」 429」を参照) も別々に記載します。

2. 装置のリソースの運転エントリが診断ログに記録される

3. メッセージが診断リストに入力される

プロジェクト停止の手順

プロジェクトを停止すると、次の処理が開始されます。

1. プロジェクトのすべての装置ですべてのリソースが停止される

選択したリソースで処理される変数とアプリケーションパラメータへの影響は、別々に記載します (「[変数とアプリケーションパラメータへの影響](#)」 426」を参照)。「変数の強制」への影響 (「[有効な「変数強制」への影響](#)」 427」を参照) と「オンライン変更」への影響 (「[有効な「オンライン変更」への影響](#)」 429」を参照) も別々に記載します。

2. プロジェクトのすべてのリソースについて、運転エントリが診断ログに記録される

3. メッセージが診断リストに入力される

STOP状態中の動作

リソースがSTOP状態になっているときのプロセスデータとシステムセクションの動作は、STOP状態のトリガがエラーであっても手動の停止コマンドであっても、常に同じです。以下のテーブルに、リソースがSTOP状態になっているときのプロセスデータおよびシステムセクションの動作を示します。

処理データ、システムセクション		動作	
プロセスデータ	I/Oマッピングがあるモジュールバスの入力データ	ビットモジュール	入力 = 現在の物理入力信号
	(「装置停止」については [1] を参照)	バイトモジュール	入力 = 現在の物理入力値
	I/Oマッピングがあるモジュールバスの出力データ	ビットモジュール	出力 = ゼロ
	(「リソースの停止」については [2] を参照)	バイトモジュール	出力 = スイッチオン値 (モジュールの取扱説明書を参照)
	停止したリソースの出力PI変数	出力PI変数は無効としてマークされる ([1] と [2] を参照)	
システムセクション	FSリソース	以下のセクションを参照 - リソース停止の手順 - 装置停止の手順 - プロジェクト停止の手順	
	STリソース		
	FSモジュールバスPSSu		
	STモジュールバスPSSu	影響なし	
	FS SafetyNET p		
	ST SafetyNET p		
	外部通信 例: 「IP接続」システムセクション、 「PROFIBUS-DPスレーブ」システムセクション		

[1]

他のPSS 4000専用装置によってアクセスされたプロセスデータは、他のPSS 4000専用装置の代入値、有効ビット = FALSEに設定されます (「[プロセスデータの有効性](#)」 281」を参照)。

[2]

上のテーブルに示すように動作するのは、停止したリソースの出力PI変数に割り付けられているハードウェア出力だけです。他のリソース (実行中のリソース) の出力PI変数に割り付けられているハードウェア出力は、これらの変数によって引き続き記録されます。

11.11 リセットボタンの機能

すべてのPSS 4000専用装置にリセットボタンがあります。

リセットボタンには、次のようなさまざまな機能があります。

- ▶ 装置のウォームリセットの実行
- ▶ 作業者の意図的なアクション
 - 装置のオリジナルリセットの実行
 - SDカードからの命名データ／装置プロジェクトの転送
- ▶ リカバリモードへの切り替え

11.11.1 装置のウォームリセットの実行

PSS 4000専用装置のリセットボタンを使用したリセットはウォームリセットに相当します。ウォームリセットを実行するには、装置のリセットボタンを押します (5秒以上押さない)。

詳細については、「[オペレーティングステータス「システムセクションを起動中」](#)  [336](#)」のフローチャートを参照してください。

ウォームリセットの全般的な影響については、個別に説明します ([「リセットコマンド」](#)  [432](#))を参照)。

11.11.2 意図的なオペレータ操作による装置のオリジナルリセット

オリジナルリセットは、PSS 4000専用装置のリセットボタンを使用して実行できます。

前提条件

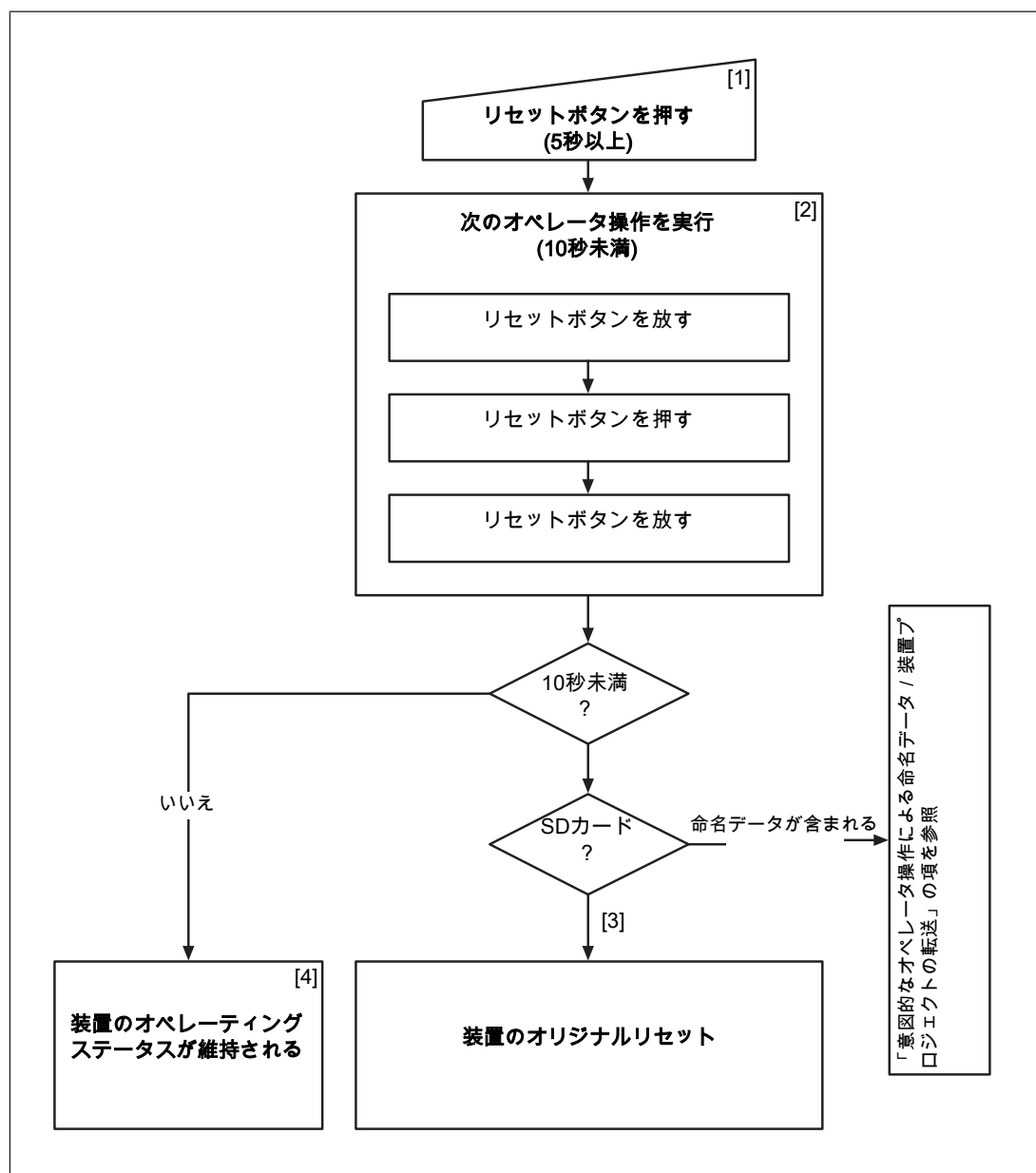
装置のSDカードが挿入されていないか、空である。



情報

SDカードに不具合がある場合も、オリジナルリセットが実行されます。

オリジナルリセットの手順



[1]

装置のリセットボタンを5秒以上押し続けます。装置は現在のオペレーティングステータスのまま変わりません。

[2]

次の意図的なオペレータ操作を10秒以内に実行します。

リセットボタンを放す – リセットボタンを押す – リセットボタンを放す

意図的なオペレータ操作の開始が、「DIAG」LED (赤／緑で点滅) によって示されます。

備考: [DIAG] LEDが赤／緑で点滅したら、5秒以内にリセットボタンを放す必要があります。そうしないと、装置が再起動します。

[3]

SDカードがチェックされます。空のSDカードが挿入されている場合、またはSDカードが挿入されていない場合、オリジナルリセットが実行されます ([「オリジナルリセット」 \[📖 449\]](#)を参照)。

備考:

SDカードに命名データが含まれている場合、オリジナルリセットは実行されません ([「意図的なオペレータ操作による命名データ／装置プロジェクトの転送」 \[📖 443\]](#)を参照)。

[4]

意図的なオペレータ操作を10秒以内に実行しなかった場合、装置は現在のオペレーティングステータスのまま変わりません。

11.11.3 意図的なオペレータ操作による命名データ／装置プロジェクトの転送

リセットボタンを使用して、命名データや装置プロジェクトをSDカードからPSS 4000専用装置に転送できます。PAS4000を使用する必要がないため、たとえばサービス操作中にこの手順が役に立つことがあります。

PAS4000を使用しない場合、次のようなケースでは、リセットボタンを使用して意図的なオペレータ操作を実行することが不可欠です。

- ▶ 新しい命名データがSDカードから転送される。これは、以下を除くすべてのPSS 4000専用装置に適用されます。
 - PSSu H FS SN SD(-T)(-R)
 - PSSu H FS SN SD M12 (-T)(-R)
 - PSS67 IO1 16FDI
- ▶ 同一の命名データがある装置については、プロジェクトがSDカードから転送される。これは、以下を除くすべてのPSS 4000専用装置に適用されます。
 - PSSu H FS SN SD(-T)(-R)
 - PSSu H FS SN SD M12 (-T)(-R)
 - PSS67 IO1 16FDI

備考: SDカードに異なる命名データおよび異なる装置プロジェクトが含まれている場合、意図的なオペレータ操作を2回実行する必要があります。最初の操作では、命名データのみが転送されます。

前提条件

- ▶ SDカードが装置に挿入されていること。
- ▶ SDカードが空でなく、不具合もないこと。
- ▶ SDカードの装置プロジェクトが製品型式と一致すること。

命名データや装置プロジェクトの転送手順

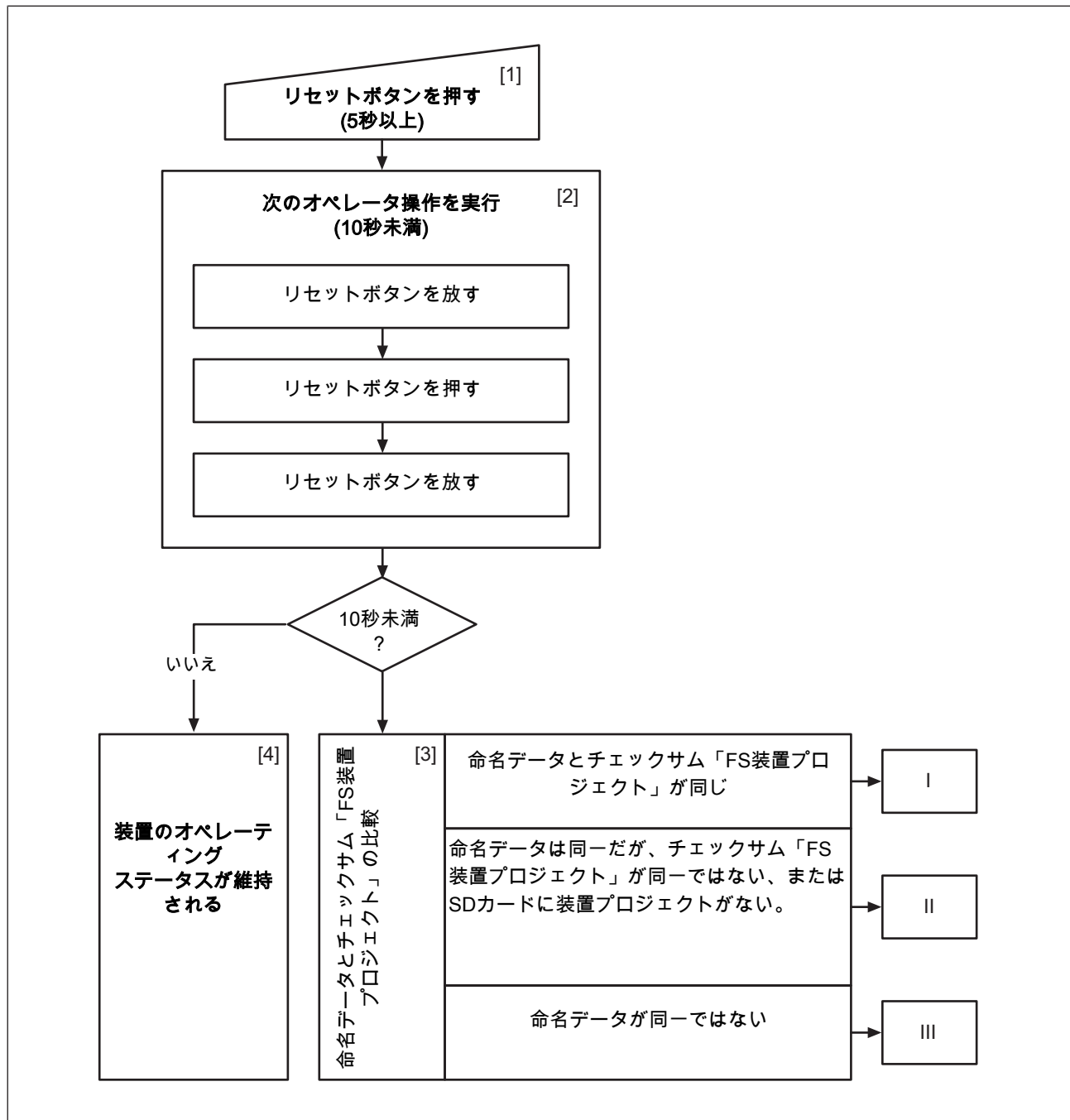


図:

[1]
装置のリセットボタンを5秒以上押し続けます。装置は現在のオペレーティングステータスのまま変わりません。

[2]
次の意図的なオペレータ操作を10秒以内に実行します。
リセットボタンを放す – リセットボタンを押す – リセットボタンを放す
意図的なオペレータ操作の開始が、「DIAG」LED (赤／緑で点滅) によって示されます。

備考: [DIAG] LEDが赤／緑で点滅したら、5秒以内にリセットボタンを放す必要があります。そうしないと、装置が再起動します。

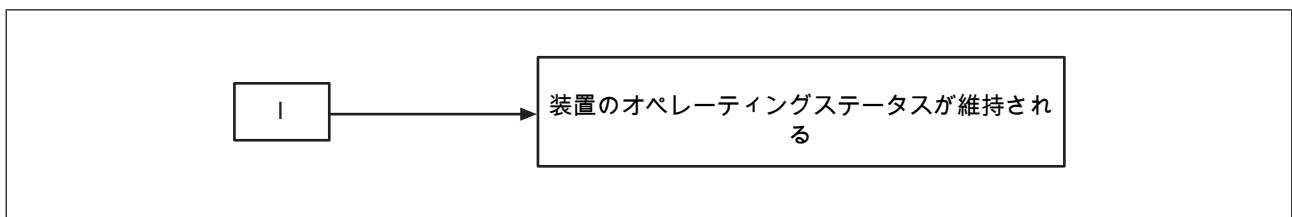
[3]

SDカード上の命名データとチェックサム「FS装置プロジェクト」が装置上の対応するデータと比較され、それぞれの状況に応じて処理が行われます。図の「I」、「II」、「III」は、それぞれの図と説明に対応しています。

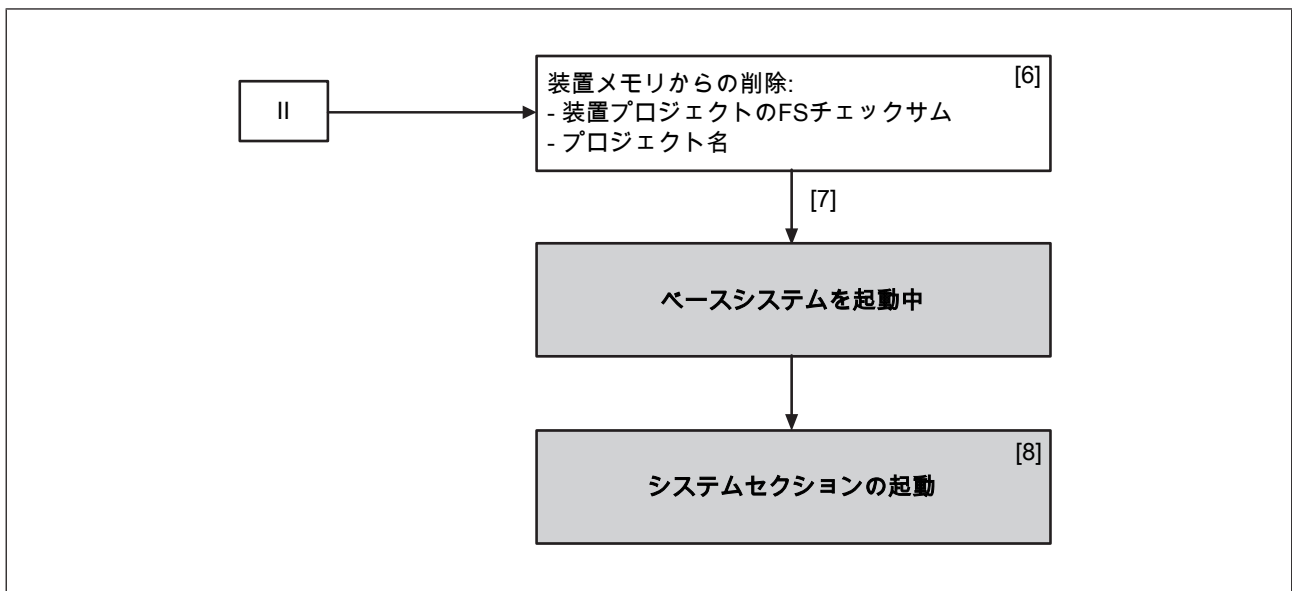
[4]

意図的なオペレータ操作を10秒以内に実行しなかった場合、装置は現在のオペレーティングステータスのまま変わりません。

比較後の手順



命名データとチェックサム「FS装置プロジェクト」が同じです。装置は現在のオペレーティングステータスのまま変わりません。



[6]

命名データは同じでも、チェックサム「FS装置プロジェクト」が同じではない場合、または、SDカードに装置プロジェクトがない場合、チェックサム「FS装置プロジェクト」とプロジェクト名は装置の装置メモリから削除されます。

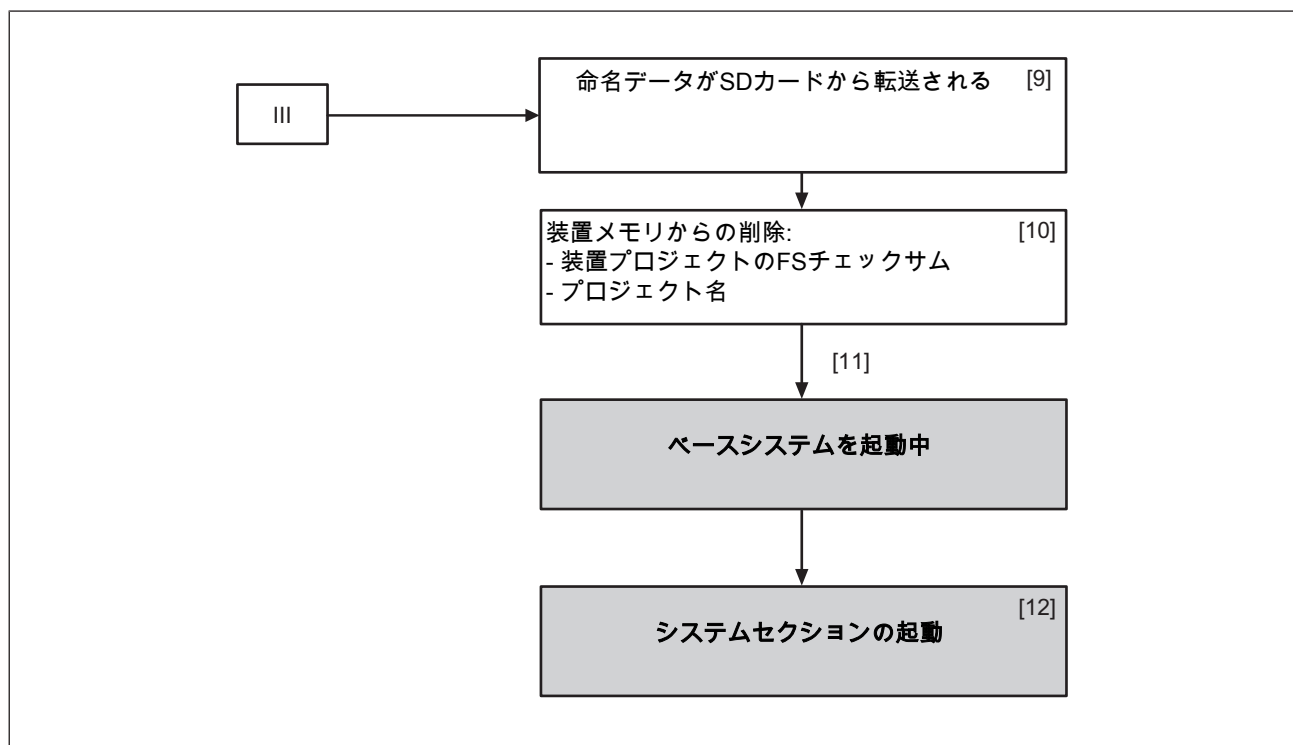
[7]

装置が再起動されます。この動作は、再起動コマンドの場合と同じです（「オペレーティングステータスと装置のオペレーティングステータスの変更」を参照）。

[8]

SDカードに製品型式に一致する装置プロジェクトがなく、オペレーティングステータス「システムセクションを起動中」になっている場合、装置プロジェクトはSDカードから転送されます。

SDカードに装置プロジェクトがない場合、または製品型式の異なる装置プロジェクトが含まれている場合、装置は装置プロジェクトなしで起動します。この場合、装置に装置プロジェクトがあることを確認するために、プロジェクトの直接ダウンロードを実行するか、または製品型式が一致する装置プロジェクトを含むSDカードを使用して意図的なオペレータ操作をもう一度実行する必要があります。



[9]

命名データが異なる場合は、SDカード上の命名データが装置に転送されます。

[10]

チェックサム「FS装置プロジェクト」とプロジェクト名は装置の装置メモリから削除されます。

[11]

装置が再起動されます。この動作は、再起動コマンドの場合と同じです（「[オペレーティングステータスと装置のオペレーティングステータスの変更](#)  331」を参照）。

[12]

SDカードに製品型式に一致する装置プロジェクトがなく、オペレーティングステータス「システムセクションを起動中」になっている場合、装置プロジェクトはSDカードから転送されます。旧装置プロジェクトのユーザプログラムに不揮発性の変数が含まれている場合、装置は以下のような不揮発性変数値で起動されます。

ケース	属性がRETAIN PERSISTENTの変数	揮発性 変数 [1]	アプリケーションパラメータ
ユーザプログラム内で、変数は変更されずに使用できます (リソース割り付け、FQN、データ型が同じです)。	保存済みの値	変数は初期値に設定される	アプリケーションパラメータに 設定された値
ユーザプログラム内で変数が新規になる	変数は初期値に設定される		
変数がユーザプログラム内で使用不可になる	変数は削除される		アプリケーションパラメータが 削除される

[1] 保持されない変数がアプリケーションパラメータとして使用されている場合、「アプリケーションパラメータ」列が適用されます。

SDカードに装置プロジェクトがない場合、または製品型式の異なる装置プロジェクトが含まれている場合、装置は装置プロジェクトなしでそのシステムセクションを起動します。装置に装置プロジェクトがあることを確認するために、プロジェクトの直接ダウンロードを実行するか、または対応する装置プロジェクトを含むSDカードを使用して意図的なオペレータ操作をもう一度実行する必要があります。

11.11.4 リカバリモード

再起動時 (電源電圧をONにする) または装置の再起動時に、リカバリモードに切り替えることができます。この場合、テストフェーズの後、システムコンポーネントが起動される前に、起動プロセスがスケジュールされた割り込みを行います (「[オペレーティングステータス「システムセクションを起動中」](#)」 336」を参照)。

リカバリモードではオンライン操作は使用できません。

リカバリモードの実行

- ▶ リセットボタンをしばらく押したままにして、再起動または装置の再起動を実行します。

備考:

リカバリモードは、「SDカード」ステータスLED (緑色の点滅または赤色の点灯) で識別できます。リセットボタンを十分に長押ししないと、システムはリカバリモードに切り替わりません。装置は通常の起動フェーズをすべて実行します。

SDカードの取り外し

リカバリモードでは、SDカード上のファイルを損傷することなくSDカードを取り外すことができます。次の2種類のシナリオがあります。

- ▶ SDカードが整合状態です (「[ジャーナリングファイルシステム](#)」 478」を参照)。
「SDカード」ステータスLEDが緑色に点滅します。
- ▶ 何らかの理由で、SDカードのファイルシステムは整合状態を再構築できません (「[ジャーナリングファイルシステム](#)」 478」を参照)。
「SDカード」ステータスLEDが赤色に点灯します。

リカバリモードの終了

- ▶ リカバリモードで、リセットボタンを10秒以上押します。
装置が再起動します。

11.12 オリジナルリセット

オリジナルリセットを行う意味があり、また本当に必要になるのは、PSS 4000専用装置またはPSS 4000対応装置が既存のプラント／マシンから取り外された (他のアプリケーションで再使用が可能) 場合です。PSS 4000専用装置／対応装置が、オリジナルリセットの後に、自動化システムPSS 4000に属するプロジェクトで再使用される場合、装置の命名手順を新しく実行する必要があります。

11.12.1 オリジナルリセットの影響

PSS 4000専用装置またはPSS 4000対応装置に対するオリジナルリセットの全般的な影響:

- ▶ 装置に保存されているプロジェクト関連データと作業ディレクトリはすべて削除されます。例:
 - プロジェクト名
 - 安全関連アプリケーション用装置の場合: チェックサム「FS装置プロジェクト」
 - 不揮発性メモリを使用する装置の場合: 不揮発性変数
- ▶ 装置名は既定の装置名にリセットされます (「命名データ  309」を参照)。
- ▶ SDカードがある装置では、SDカードに保存されているプロジェクト関連データと作業ディレクトリはすべて削除されます。例:
 - 装置プロジェクト (関連するシステムセクションすべてのコンフィグレーションを含む)
 - 安全関連アプリケーション用装置の場合: チェックサム「FS装置プロジェクト」

システムのものではないすべてのデータと作業ディレクトリも削除されます。例:

- ユーザがSDカードに保存したすべてのユーザデータ (/userfilesフォルダとそこにあるすべてのディレクトリとデータを含む)
- バックアップコピーのある/epフォルダ (ある場合)
- ▶ 装置診断リストのメッセージが「inactive」の状態に変わります。ステータスの変更は、装置の診断ログに記録されません。
- ▶ 装置の診断ログは保持されます。
- ▶ PSS 4000ファームウェアの現在のバージョンが装置に残ります。
- ▶ オリジナルリセットの後、PSS 4000ファームウェアは装置で再起動します。動作は再起動コマンドの場合と同じです。

全般的な影響以外にも、次のような装置固有の効果があります。

- ▶ PSS 4000専用装置
- ▶ PSS 4000に対応する装置

11.12.1.1 PSS 4000専用装置での効果

PSS 4000専用装置に対するオリジナルリセットの装置特有の効果:

- ▶ 装置のすべての出力がゼロに設定されます。
- ▶ 他の装置によってアクセスされたプロセスデータは、他の装置の代入値、有効ビット = FALSEに設定されます (「[プロセスデータの有効性](#)」 281」を参照)。
- ▶ 装置はIPアドレスを既定の設定にリセットし、Auto-IPメカニズムを有効化します (「[Auto-IPメカニズム](#)」 82」を参照)。
- ▶ システムセクションPROFIBUS-DPスレーブがある装置:
装置のDIPスイッチが126_Dに設定されている場合、装置のステーションアドレスは126_Dにリセットされます。

11.12.1.2 PSS 4000対応装置での影響

オリジナルリセットは、PSS 4000対応装置に次のような装置固有の効果を与えます。

- ▶ オリジナルリセットは、装置のPSS 4000ファームウェアに対してのみ有効です。
- ▶ 装置のネットワーク設定 (IPアドレスなど) は変更されません。つまり、オリジナルリセットは、装置のネットワーク設定 (「命名データ」 309)を参照) に影響しません。

11.12.2 オリジナルリセットの実行

装置に応じて、オリジナルリセットを次のように実行できます。

▶ PSS 4000専用装置

オリジナルリセットは、PAS4000で、または装置のリセットボタンを用いて行います。

▶ PSS 4000に対応する装置

オリジナルリセットはPAS4000で行われます。

11.13 イネーブルの原理

イネーブルの原理により、FSデータソースからのイネーブルが存在する場合、STデータソースはFSハードウェア出力を制御できます。エラー発生時に、FSデータソースは、STデータソースによって制御されるFSハードウェア出力が設定されないようにする必要があります。

- ▶ 有効なSTデータソース:
 - ST出力PI変数
 - 外部通信からのST入力データ (例: Modbus/TCP、フィールドバス)
- ▶ 有効なFSデータソース:
 - FS出力PI変数
 - 外部通信からのFS入力データ (例: 他の製造業者のFSバスシステム)

原理

PAS4000のI/Oマッピングでは、ユーザはSTデータソースとFSデータソースを同じFSデータシンク (FSハードウェア出力など) にマッピングします。PSS 4000は、このI/Oマッピングに基づき内部で評価を行います。すなわち、評価はPSSuシステムのリソースでは行われなため、ユーザプログラム内でのプログラミングは不要です。STデータソース、STデータソースの有効なビット、FSデータソース、FSデータソースの有効なビットは、内部評価の際に論理的にリンクされて評価されます。ロジックオペレーションの結果がFSハードウェア出力を制御します。

次の規則が適用されます。

- ▶ FSデータソースと、FSデータソースの有効なビットが「TRUE」であれば、FSハードウェア出力はSTデータソースによりONとOFFを切り替えることができます。FSハードウェア出力のイネーブルLEDが点灯します。
- ▶ FSデータソース、FSデータソースの有効なビット、またはその両方が「FALSE」になると、FSハードウェア出力にはただちに代入値が使用されます。デジタルFSハードウェア出力の場合、これは安全にシャットダウンされたことを示します。FSハードウェア出力のイネーブルLEDは点灯しません。STデータソースを通じた制御は無視されます。
- ▶ STデータソースの有効なビットが「FALSE」になると、FSハードウェア出力にはただちに代入値が使用されます。デジタルFSハードウェア出力の場合、これは安全にシャットダウンされたことを示します。FSハードウェア出力のイネーブルLEDが点灯します。

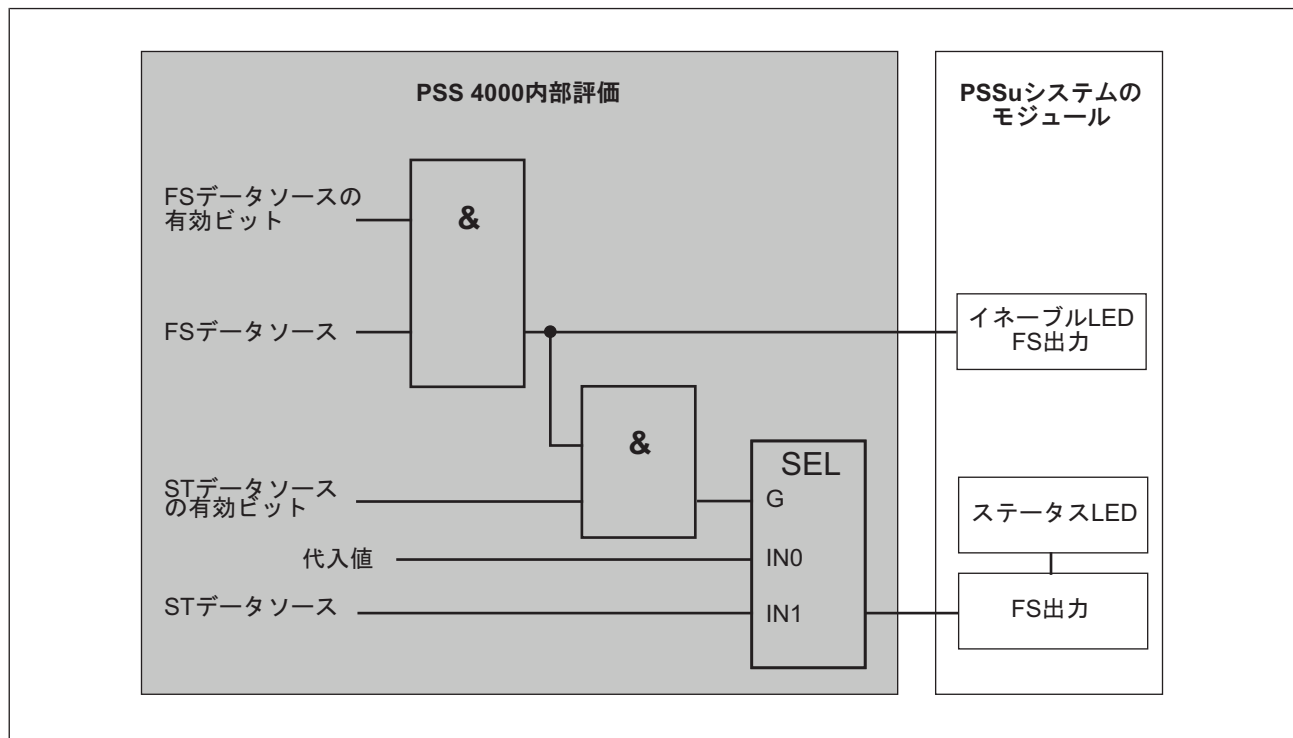


図: PSSuシステムを例として使用した内部評価 (原則)

注:

STリソースで処理されるSTデータは、安全ではありません。これは、安全に関するアクション (開始／停止など) の追加基準にすぎず、安全機能を妨げないようにする必要があります。たとえば、何らかの理由でSTリソースの停止信号が有効化されていない場合、または開始信号が継続して有効化されている場合、マシンまたはプロセスの安全が損なわれないようにする必要があります。

11.14 応答時間

応答時間に関する次の説明は、PSS 4000専用装置の入出力にのみ該当します。入出力がさまざまなPSS 4000専用装置に分散される場合は、SafetyNET pインタフェースの処理時間とSafetyNET pの通信時間も考慮する必要があります。

全体応答時間を判断する際には、必要に応じて、関連する外部通信関係 (例: Modbus/TCP、UDP、PROFIBUS-DP経由での通信)、サードパーティ製周辺装置、センサ／アクチュエータおよびサードパーティ製制御システムも考慮する必要があります。これらの影響については、このマニュアルでは説明していません。



情報

応答時間に関する次の説明は、PAS4000バージョン1.2.0以降に該当します。

「応答時間」とは、定義済みイベントと、結果としてトリガされる定義済み応答との間の経過時間です。

11.14.1 応答時間に影響を与える時間

応答時間は、次の時間によって決定されます。

周期性タスクのサイクル時間 (t_{Task})

タスクサイクル時間 t_{Task} は、ユーザがPAS 4000でタスクをコンフィグレーションするときに定義します。これは、タスクが実行される必要がある制限時間を決定します (「[タスクの基本情報](#)」[\[286\]](#)を参照)。

値の範囲:

- ▶ STタスク: 2 ms ~ 2 000 000 ms
- ▶ FSタスク: 6 ms ~ 10 000 ms

タスクサイクル時間 t_{Task} の値の範囲は装置によって異なります。

タスクの実行時間 ($t_{ProcTask}$)

実行時間 $t_{ProcTask}$ は、タスクサイクルが開始してからタスクの実行が終了するまで、つまり、出力のプロセスイメージとリソースのグローバル変数が使用可能になるまでの時間です (「[データベースの構成](#)」-「[タスクでのデータベース](#)」[\[464\]](#)を参照)。

モジュールバスのサイクル時間 (t_{MBUS})

PSS 4000専用装置のFSモジュールバスとSTモジュールバスには、それぞれサイクル時間が1つずつあります。これはPAS4000によって計算され、表示されます。サイクル時間 t_{MBUS} は、ヘッドモジュールとモジュールの間でデータを伝送するのに必要な時間です。

値の範囲:

- ▶ STモジュールバス: 2 ms ~ 45 ms
- ▶ FSモジュールバス: 6 ms ~ 30 ms

次の点に注意してください:

- ▶ それぞれのモジュールバスのサイクル時間は、常にサイクル時間が最も短いタスクと同じになります。ただし、タスクのサイクル時間は、該当するモジュールバスの最大サイクル時間以下でなければなりません ($t_{MBUS} = t_{Task\ min}$)。
- ▶ PSS 4000専用装置でタスクが1つだけ実行された場合、モジュールバスのサイクル時間はタスクのサイクル時間と同じになります。ただし、タスクのサイクル時間は、該当するモジュールバスの最大サイクル時間以下でなければなりません ($t_{MBUS} = t_{Task}$)。

モジュールの入力データの処理時間 (t_{ProcIM})

処理時間 t_{ProcIM} : モジュール取扱説明書の技術データを参照してください。

処理時間 t_{ProcIM} は、入力およびモジュールバスに用意されている入力データで物理的に変化する信号間の、モジュールにおける遅延時間です。処理時間には、内部ランタイム、温度ドリフト、コンポーネントの分散などのさまざまな影響が考慮されており、処理時間はモジュールによって異なります。

モジュールにフィルタ時間をコンフィグレーションできる場合は、次のことが適用されます。

$$t_{ProcIM} = t_{ProcIM} + \text{コンフィグレーションされたフィルタ時間}$$

モジュールの出力データの処理時間 (t_{ProcOM})

処理時間 t_{ProcOM} : モジュールの取扱説明書の「技術データ」を参照

処理時間 t_{ProcOM} は、モジュールが出力データを処理するために内部的に必要とする時間です。処理時間では、内部ランタイム、温度ドリフト、部品の分散状態などのさまざまな影響が考慮されます。この時間はモジュールによって異なります。

SafetyNET pの基本サイクル時間 (t_{BasicSNp})

基本サイクル時間 t_{BasicSNp} は、各サイクルの一部としてプロジェクト内でテストを実行し、SafetyNET p経由でデータを送信できるかどうかを確認するのに必要な時間です。基本サイクル時間は、PAS4000によって計算され、ビルドプロセス中に表示されます。

▶ プロジェクト内SNp通信

SNp通信に関連するFSおよびSTシステムセクションのサイクル時間は、基本サイクル時間を計算する際に考慮されます。コンフィグレーションに応じて、FSハートビートサイクル時間 $t_{\text{SNp FS-HBT}}$ も考慮されます（「FSハートビート (FS-HBT) [113]」を参照）。したがって、考慮されるサイクル時間には t_{Task} 、 t_{MBUS} 、 t_{ExtCo} 、および $t_{\text{SNp FS-HBT}}$ が含まれます。

基本サイクル時間は、SNp通信に関連するFSおよびSTシステムセクションのサイクル時間の最大公約数に対応し、コンフィグレーションによっては、FSハートビートサイクル時間の手動でコンフィグレーションされた値に対応します。

例:

$$t_{\text{BasicSNp}} = \text{最大公約数} (t_{\text{Task}}, t_{\text{MBUS}}, t_{\text{SNp FS-HBT}}) \text{ (ms)}$$

$$- t_{\text{Task}} = 10 \text{ ms}$$

$$- t_{\text{MBUS}} = 5 \text{ ms}$$

$$- t_{\text{SNp FS-HBT}} \text{ は } 120 \text{ ms} \text{ です。たとえば次のようになります。}$$

$$\text{結果: } t_{\text{BasicSNp}} = 5 \text{ ms}$$

▶ プロジェクト間SNp通信 (PASconnectインタフェースのバージョン1.2.x)

SNp通信に関連するFSタスクとSTタスクのサイクル時間 t_{Task} は、基本サイクル時間を計算する際に考慮されます。

基本サイクル時間は、SNp通信に関連するタスクのタスクサイクル時間 (t_{Task}) の最大公約数に相当します。

例:

$$t_{\text{BasicSNp}} = \text{最大公約数} (t_{\text{Task 1}}, t_{\text{Task 2}}) \text{ (ms)}$$

$$- t_{\text{Task 1}} = 10 \text{ ms}$$

$$- t_{\text{Task 2}} = 5 \text{ ms}$$

$$\text{結果: } t_{\text{BasicSNp}} = 5 \text{ ms}$$

▶ プロジェクト間SNp通信 (PASconnectインタフェースのバージョン1.3.x/1.4.x)

SNp通信に関連するFSタスクとSTタスクのサイクル時間 t_{Task} は、基本サイクル時間を計算する際に考慮されます。PAS4000で手動コンフィグレーションされたFSハートビートサイクル時間 $t_{SNp\ FS-HBT}$ も考慮されます（「FSハートビート (FS-HBT)」[\[113\]](#)を参照）。

基本サイクル時間は、SNp通信に関連するタスクサイクル時間の最大公約数と、FSハートビートサイクル時間にコンフィグレーションされた値に対応します。

例:

$t_{BasicSNp} = \text{最大公約数}(t_{Task\ 1}, t_{Task\ 2}, t_{SNpFS-HBT}) \text{ (ms)}$

– $t_{Task\ 1} = 10 \text{ ms}$

– $t_{Task\ 2} = 10 \text{ ms}$

– $t_{SNpFS-HBT} = 110 \text{ ms}$

結果: $t_{BasicSNp} = 10 \text{ ms}$

SafetyNET pサイクル時間 (t_{SNp})

サイクル時間 t_{SNp} は、次の条件に従って区別されます。

▶ プロジェクト内SNp通信のサイクル時間および

プロジェクト間SNp通信のサイクル時間 (PASconnectインタフェースのバージョン1.2.x)

サイクル時間 t_{SNp} は、PSS 4000装置の送信接続 (Tx接続) を表します。つまり、PSS 4000装置の各Tx接続にはそれぞれ固有のサイクル時間 t_{SNp} があります。

サイクル時間は、データを送信するために、Tx接続を開始するのにかかる時間です。サイクル時間は、基本サイクル時間 $t_{BasicSNp}$ の整数倍です。これは送信側PSS 4000装置を含むプロジェクトの基本サイクル時間です。

▶ プロジェクト間SNp通信のサイクル時間 (PASconnectインタフェースのバージョン1.3.x/1.4.x)

サイクル時間 t_{SNp} は、PSS 4000装置の受信接続 (Rx接続) を表します。つまり、PSS 4000装置の各Rx接続にはそれぞれ固有のサイクル時間 t_{SNp} があります。

サイクル時間は、Rx接続からの入力データが以降の処理に使用できるかどうかを、受信側PSS 4000装置が周期的にチェックする時間表します。サイクル時間は、基本サイクル時間 $t_{BasicSNp}$ の整数倍です。これは受信側PSS 4000装置を含むプロジェクトの基本サイクル時間です。

明確に区別する必要がある場合、サイクル時間 t_{SNp} は次のように記述されます。

SNp通信	PASconnectインタフェースのバージョン	サイクル時間の説明
プロジェクト内	- - -	$t_{SNp} \text{ (Tx)}$
プロジェクト間	1.2.x	$t_{SNp} \text{ (Tx)}$
	1.3.x/1.4.x	$t_{SNp} \text{ (Rx)}$

接続のサイクル時間は、PAS4000によって計算され、ビルドプロセス中に表示されます。

最大サイクル時間:

- ▶ ST SafetyNET p: 装置により異なる
- ▶ FS SafetyNET p: 装置により異なる

SafetyNET pの最大伝送遅延時間 (t_{DelaySNp})

プロジェクト内では、伝送遅延時間はFSハートビートを使用して定期的に測定されます (「FSハートビート (FS-HBT) [113]」を参照)。測定される時間は、装置Aでデータ伝送が開始された後、装置Bでデータパケットが使用可能になるまでにかかる時間です。この時間は、コンフィグレーションされた最大伝送遅延時間 t_{DelaySNp} を超えることはできません。

測定されるこの時間が最大伝送遅延時間を超えると、装置Aから受信したすべてのFS受信接続 (FS-Rx 接続) が装置Bで無効になります (「接続の周期的な処理 [101]」を参照)。代入値は、有効ビット = FALSEの状態、受信データで 사용됩니다。

通信遅延時間の監視は、個々のPSS 4000装置間のSNp接続のサイクル時間とは無関係です。

▶ プロジェクト内SNp通信および

プロジェクト間SNp通信 (PASconnectインタフェースのバージョン1.3.x/1.4.x)

最大伝送遅延時間は、PAS4000で手動コンフィグレーションする必要があります。

値の範囲: 4 ms ~ 2000 ms (既定の設定: 25 ms)

備考:

- 最大伝送遅延時間は、SafetyNET pデータサブパスの最大応答時間に影響するため、可能な限り短くする必要があります (「SafetyNET pがデータソースのデータサブパス [467]」を参照)。デフォルト設定は、多くのアプリケーションに適した値です。
- 次の場合、伝送遅延時間に問題が発生することがあります
 - Ethernetインフラに問題がある (スイッチ、ルータ、ケーブルなど)
 - 装置のSNp接続の数が多くなっている
 - 装置のシステム負荷が高い。例えば、多数の電子モジュールを搭載したPSS 4000装置や大規模なユーザプログラムなど

通信遅延時間を延長する前にまずEthernetインフラを確認する必要があります。

- プロジェクト間SNp通信では、データの受信側を含むプロジェクトの伝送遅延時間 t_{DelaySNp} を、SafetyNET pデータサブパスの応答時間の計算に使用する必要があります。

▶ プロジェクト間SNp通信 (PASconnectインタフェースのバージョン1.2.x)

伝送遅延時間は、PAS4000によって計算され、ビルドプロセス中に表示されます。

データの受信側を含むプロジェクトの伝送遅延時間 t_{DelaySNp} を、SafetyNET pデータサブパスの応答時間の計算に使用する必要があります (「SafetyNET pがデータソースのデータサブパス [467]」を参照)。

外部通信でのサイクル時間 (t_{ExtCo})

外部通信とは、外部出力データがPSS 4000プロジェクトを離れ、外部入力データがModbus/TCP、UDP、フィールドバスなどを経由してPSS 4000プロジェクトに入るデータ交換のことです。使用する通信モードに従って、外部出力データは、PSS 4000専用装置の出力データ領域から送信されます。使用している通信モードによっては、外部入力データはPSS 4000 CPUの入力データ領域にあり、そこから読み取られます。

この説明では、送信および受信のサイクルタイムは t_{ExtCo} で示しています。これらの意味は、使用している通信モードによって異なります。

▶ Modbus/TCPクライアント接続

サイクル時間 t_{ExtCo} は、各サイクルの一部として、接続クライアントがデータ転送を開始するのにかかる時間です。

PAS 4000で、サイクル時間は、PSS 4000専用装置の各接続クライアント用に個別に設定できます。つまり、PSS 4000専用装置の各接続クライアントには、それぞれ独自のサイクル時間 t_{ExtCo} があります。自動的に計算することも、ユーザが設定することもできます (PAS4000でコンフィグレーション)。

▶ **Modbus/TCPサーバ接続:**

サイクル時間 t_{ExtCo} は、各サイクルの一部として、PSS 4000専用装置のサーバロケーションテーブルからデータを読み取り、サーバロケーションテーブルからリクエストされたデータが接続クライアントに転送されるのにかかる時間です。

PAS 4000で、サイクル時間は、PSS 4000専用装置の各サーバロケーションテーブルで個別に設定できます。つまり、PSS 4000専用装置の各サーバロケーションテーブルには、それぞれ独自のサイクル時間 t_{ExtCo} があります。自動的に計算することも、ユーザが設定することもできます (PAS4000でコンフィグレーション)。

▶ **Raw TCP通信の「受信データ」と「送信データ」**

サイクル時間 t_{ExtCo} は、各サイクルの一部として、外部データをTCP接続から読み取り、TCP接続経由でデータ (外部出力データ) 送信を開始するのにかかる時間です。

PAS 4000で、サイクル時間は、PSS 4000専用装置の各TCP接続用に個別に設定できます。つまり、各TCP接続には、それぞれ独自のサイクル時間 t_{ExtCo} があります。自動的に計算することも、ユーザが設定することもできます (PAS4000でコンフィグレーション)。

▶ **Raw UDP通信の「受信データ」と「送信データ」**

サイクル時間 t_{ExtCo} は、各サイクルの一部として、外部入力データをUDP接続から周期的に読み取り、UDP接続経由でデータ (外部出力データ) 送信を周期的に開始するのにかかる時間です。

PAS 4000で、サイクル時間はPSS 4000専用装置の各UDP接続用に個別に設定できます。つまり、各UDP接続には、それぞれ独自のサイクル時間 t_{ExtCo} があります。自動的に計算することも、ユーザが設定することもできます (PAS4000でコンフィグレーション)。

▶ **PROFIBUS-DPスレーブの「受信データ」と「送信データ」:**

サイクル時間 t_{ExtCo} は、各サイクルの一部として、外部入力データをPROFIBUS-DPマスタから読み取り、要求されたデータ (外部出力データ) の送信を開始するのにかかる時間です。

サイクル時間は、「PROFIBUS-DPスレーブ」システムごとにPAS 4000で定義できます。つまり、各「PROFIBUS-DPスレーブ」システムにはPROFIBUS-DPによるデータ転送用のそれぞれのサイクル時間 t_{ExtCo} があります。自動的に計算することも、ユーザが設定することもできます (PAS4000でコンフィグレーション)。

▶ **PROFIsafe搭載PROFIBUS-DPスレーブの「受信データ」と「送信データ」**

サイクル時間 t_{ExtCo} は、各サイクルの一部として、外部入力データをPROFIBUS-DPマスタ/Fホストから読み取り、要求されたデータ (外部出力データ) の送信を開始するのにかかる時間です。

サイクル時間は、「PROFIsafe付きPROFIBUS-DPスレーブ」システムセクションごとにPAS 4000で定義できます。つまり、各「PROFIsafe付きPROFIBUS-DPスレーブ」システムセクションにはそれぞれのデータ転送用のサイクル時間 t_{ExtCo} があります。サイクル時間 t_{ExtCo} は、PROFIBUS-DPとPROFIsafeの両方を介したデータ転送が対象です。自動的に計算することも、ユーザが設定することもできます (PAS4000でコンフィグレーション)。

次の点に注意してください:

PROFIsafe通信を使用する場合、サイクル時間 t_{ExtCo} の値の範囲に制限があります。この場合、同じ値の範囲の制限が、PROFIBUS-DPスレーブの通信にも適用されます。サイクル時間 t_{ExtCo} は、装置によって異なります。

▶ **EtherNet/IPアダプタの「受信データ」と「送信データ」**

サイクル時間 t_{ExtCo} は、各サイクルの一部として、外部入力データをスキャナから読み込み、スキャナへ送信された外部出力データの送信を開始するのにかかる時間です。

このサイクル時間は、PAS 4000で「EtherNet/IPアダプタ」システムセクションに指定できます。自動的に計算することも、ユーザが設定することもできます (PAS4000でコンフィグレーション)。

▶ **PROFINETの「受信データ」と「送信データ」**

サイクル時間 t_{ExtCo} は、各サイクルの一部として、外部入力データをPROFINET IOコントローラから読み取り、要求されたデータ (外部出力データ) の送信を開始するのにかかる時間です。

サイクル時間は、「PROFINET IO DEVICE」システムセクションごとにPAS 4000で定義できます。つまり、各「PROFINET IO DEVICE」システムセクションにはPROFINETによるデータ転送用のそれぞれのサイクル時間 t_{ExtCo} があります。自動的に計算することも、ユーザが設定することもできます (PAS4000でコンフィグレーション)。

11.14.2 データ処理時の処理規則

PSS 4000専用装置上でのデータ処理には以下が適用されます。

▶ モジュールの処理時間

モジュールの処理時間 (t_{ProcOM} および t_{ProcIM}) は一定です。現在のタスクサイクルの影響は受けません。

▶ モジュールバスのサイクル時間

STモジュールバスとFSモジュールバスにはそれぞれ個別のサイクル時間があります。モジュールバスのサイクル時間 t_{MBUS} は、常にサイクル時間が最も短いタスクと同じになります。ただし、そのタスクのサイクル時間は、該当するモジュールバスの最大サイクル時間以下でなければなりません。

$$t_{\text{MBUS}} = t_{\text{Task min}}$$

▶ タスクのサイクル時間

タスクのサイクル時間 (t_{Task}) は一定です。現在のタスクサイクルの影響は受けません。

11.14.3 データパスの構成

データサブパスを使用して、データパス全体にわたって1つのI/Oデータを追跡することができます。このため、データパスは関連するデータサブパスで構成されています。

データパスの応答時間は、関連するデータサブパスの応答時間を加算することで計算されます。データパスは、常にデータソース内の入力データで開始し、データシンク内の出力データで終了します。データソース／データシンクは、必ずシステムセクションにあります（「[システムセクション](#)」[\[22\]](#)」を参照）。

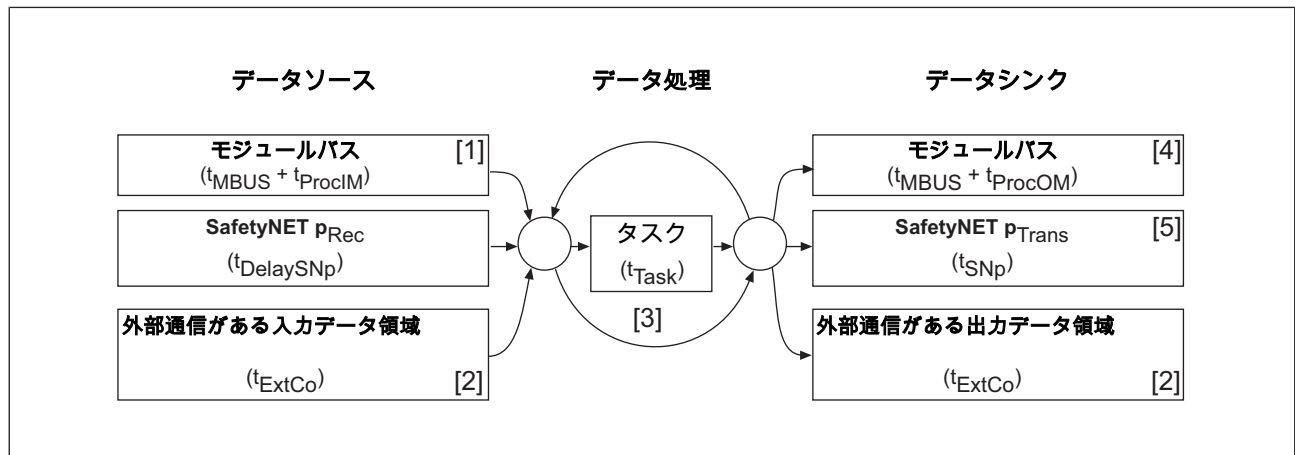


図: PSS 4000専用装置のデータパス

[1]

データソースがモジュールからの入力データの場合は、モジュールバスのサイクル時間 (t_{MBUS}) に加え、モジュール上の入力データの処理時間 (t_{ProcIM}) も考慮する必要があります。

[2]

外部通信がある場合、入力データ領域からの入力データのことを「外部入力データ」、出力データ領域からの出力データのことを「外部出力データ」ともいいます。

[3]

次のデータパスは不可能です。

- ▶ データソース: モジュールバス (入力データ) -> データシンク: モジュールバス (出力データ)
- ▶ データソース: SafetyNET p_{Rec} -> データシンク: SafetyNET p_{Trans}

[4]

データシンクがモジュール上の出力データの場合は、モジュールバスのサイクル時間 (t_{MBUS}) に加え、モジュール上の出力データの処理時間 (t_{ProcOM}) も考慮する必要があります。

[5]

1つのプロジェクトで複数のSNp接続を持つ複数のタスクが使用されている場合、これらのSNp接続のサイクル時間 t_{SNp} は異なる場合があります。データサブパスの応答時間を計算するために、正しいサイクル時間 t_{SNp} が使用されるように注意する必要があります。この計算は、装置のすべてのデータサブパス (タスク -> SafetyNET p_{Trans}) に最長サイクル時間 t_{SNp} を使用することで簡素化できます。この場合、応答時間は過大評価されます。

タスクのデータパス

タスクのデータパスの場合は、次のことに注意してください。

▶ データソースとしてのタスク

データの応答時間は、タスクサイクルが開始してから実行時間 t_{ProcTask} が経過し、その後の処理のためにデータシンク内で出力のプロセスイメージとリソースのグローバル変数ができるようになるまでの時間です。

実行時間 t_{ProcTask} がタスクサイクル時間 t_{Task} とほぼ一致する場合、それが最大応答時間です。

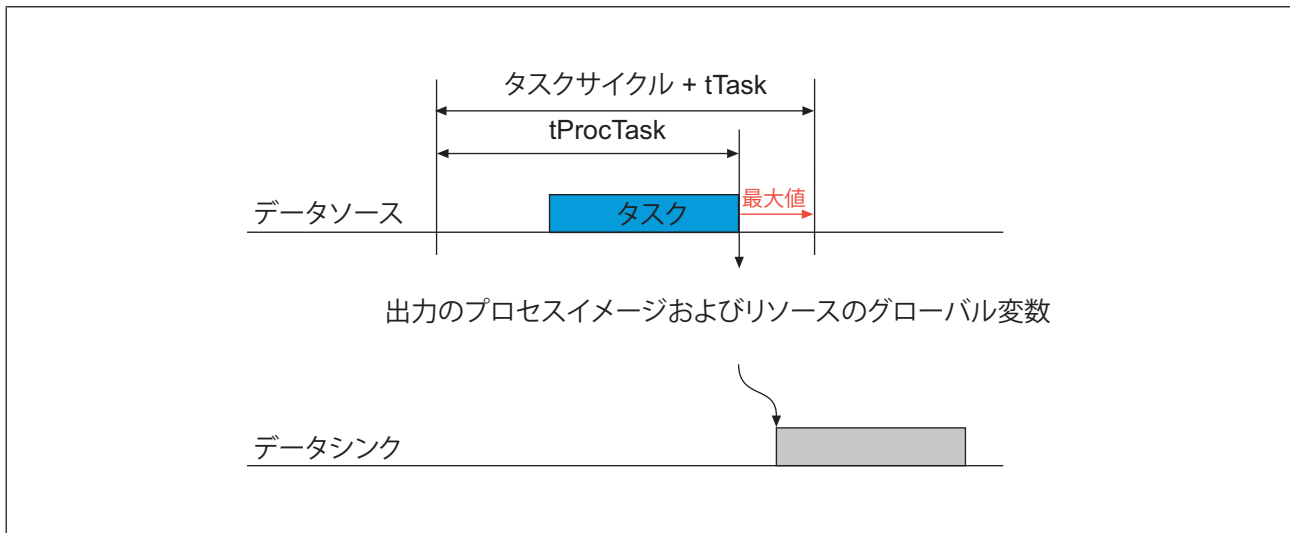


図: データソースとしてのタスクのPOUの応答時間

▶ データシンクとしてのタスク

データの応答時間は、タスクサイクルが開始してから実行時間 t_{ProcTask} が経過し、その後の処理のために出力のプロセスイメージとリソースのグローバル変数ができるようになるまでの時間です。

実行時間 t_{ProcTask} がタスクサイクル時間 t_{Task} とほぼ一致する場合、それが最大応答時間です。

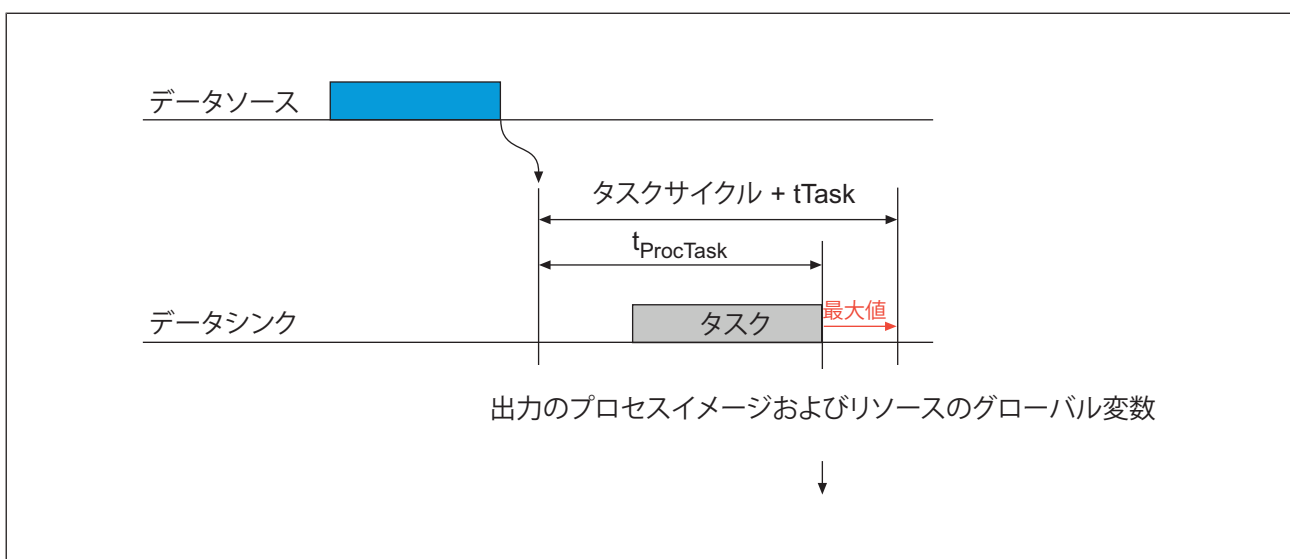


図: データシンクとしてのタスクの応答時間

11.14.4 データサブパスの最大応答時間

次の説明では、使用できるすべてのデータサブパスの最大応答時間を計算するための公式を示しています。

11.14.4.1 モジュールバスがデータソースの場合のデータサブパスの最大応答時間

モジュールバス -> タスク

PSS 4000ファームウェアバージョン1.2.0～1.3.x:

$$t_{\text{MBUS_Task_max}} = t_{\text{ProclM}} + 2t_{\text{Task}} + t_{\text{MBUS}}$$

PSS 4000ファームウェアバージョン1.4.0以降:

$$t_{\text{MBUS_Task_max}} = t_{\text{ProclM}} + 2t_{\text{Task}}$$

$t_{\text{MBUS_Task_max}}$	このデータサブパスの最大応答時間
t_{ProclM}	モジュールの入力データの処理時間
t_{MBUS}	装置のモジュールバスサイクル時間
t_{Task}	装置上のタスクのタスクサイクル時間

モジュールバス -> SafetyNET p_{Trans}

$$t_{\text{MBUS_SNp_max}} = t_{\text{ProclM}} + 2t_{\text{SNp}}$$

$t_{\text{MBUS_SNp_max}}$	このデータサブパスの最大応答時間
t_{ProclM}	モジュールの入力データの処理時間
t_{MBUS}	装置のモジュールバスサイクル時間
t_{SNp}	SNp Tx接続 (送信データ) の周期的開始のサイクル時間

モジュールバス -> 外部通信がある出力データ領域

$$t_{\text{MBUS_ExtCo_max}} = t_{\text{ProclM}} + 2t_{\text{ExtCo}}$$

$t_{\text{MBUS_ExtCo_max}}$	このデータサブパスの最大応答時間
t_{ProclM}	モジュールの入力データの処理時間
t_{MBUS}	装置のモジュールバスサイクル時間
t_{ExtCo}	外部通信がある出力データ領域からデータ転送を周期的に開始するサイクル時間

11.14.4.2 SafetyNET pがデータソースのデータサブパス

SafetyNET p_{Rec} -> タスク

プロジェクト内SNp通信および

プロジェクト間SNp通信 (PASconnectインタフェースのバージョン1.2.x):

$$t_{\text{SNp_Task_max}} = t_{\text{DelaySNp}} + 2t_{\text{Task}}$$

$t_{\text{SNp_Task_max}}$	このデータサブパスの最大応答時間
t_{DelaySNp}	▶ プロジェクト内SNp通信: プロジェクト内のSafetyNETpの伝送遅延時間 ▶ プロジェクト間SNp通信: 受信側を含むプロジェクトにおけるSafetyNET pの伝送遅延時間
t_{Task}	装置上のタスクのタスクサイクル時間

プロジェクト間SNp通信 (PASconnectインタフェースのバージョン1.3.x/1.4.x):

$$t_{\text{SNp_Task_max}} = t_{\text{DelaySNp}} + \max(t_{\text{SNp}}, t_{\text{Task}}) + t_{\text{Task}}$$

$t_{\text{SNp_Task_max}}$	このデータサブパスの最大応答時間
t_{DelaySNp}	受信側を含むプロジェクトにおけるSafetyNET pの伝送遅延時間
t_{SNp}	SNp Rx接続からの受信データの存在を周期的にテストするための受信機のサイクル時間 $t_{\text{SNp (Rx)}}$
t_{Task}	装置上のタスクのタスクサイクル時間

SafetyNET p_{Rec} -> モジュールバス

PSS 4000ファームウェアバージョン1.2.0～1.3.x:

$$t_{\text{SNp_MBUS_max}} = t_{\text{DelaySNp}} + 3t_{\text{MBUS}} + t_{\text{ProcOM}}$$

PSS 4000ファームウェアバージョン1.4.0以降:

$$t_{\text{SNp_MBUS_max}} = t_{\text{DelaySNp}} + 2t_{\text{MBUS}} + t_{\text{ProcOM}}$$

$t_{\text{SNp_MBUS_max}}$	このデータサブパスの最大応答時間
t_{DelaySNp}	プロジェクト内のSafetyNETpの伝送遅延時間
t_{MBUS}	装置のモジュールバスサイクル時間
t_{ProcOM}	出力があるモジュール上の出力データの処理時間

SafetyNET p_{Rec} -> 外部通信がある出力データ領域

$$t_{\text{SNp_ExtCo_max}} = t_{\text{Delay}} + 2t_{\text{ExtCo}}$$

$t_{\text{SNp_ExtCo_max}}$	このデータサブパスの最大応答時間
t_{DelaySNp}	プロジェクト内のSafetyNETpの伝送遅延時間

t_{ExtCo} 外部通信がある出力データ領域からデータ転送を周期的に開始するサイクル時間

11.14.4.3 外部入力データがデータソースのデータサブパス

外部通信がある入力データ領域 -> モジュールバス

$$t_{ExtCo_MBUS_max} = t_{ExtCo} + \max(t_{ExtCo}, t_{MBUS}) + t_{MBUS} + t_{ProcOM}$$

$t_{ExtCo_MBUS_max}$ このデータサブパスの最大応答時間
 t_{ExtCo} 外部通信がある入力データ領域から外部入力データを周期的に読み取るサイクル時間
 t_{MBUS} 装置のモジュールバスサイクル時間
 t_{ProcOM} モジュール上の出力データの処理時間
 $\max(t_{ExtCo}, t_{MBUS})$ 2つの値のうち大きい方が使用される

外部通信がある入力データ領域 -> タスク

$$t_{ExtCo_Task_max} = 2t_{ExtCo} + 2t_{Task}$$

$t_{ExtCo_Task_max}$ このデータサブパスの最大応答時間
 t_{ExtCo} 外部通信がある入力データ領域から外部入力データを周期的に読み取るサイクル時間
 t_{Task} 装置上のタスクのタスクサイクル時間

外部通信がある入力データ領域 -> SafetyNET p_{Trans}

$$t_{ExtCo_SNp_max} = 2t_{ExtCo} + 2t_{SNp}$$

$t_{ExtCo_SNp_max}$ このデータサブパスの最大応答時間
 t_{ExtCo} 外部通信がある入力データ領域から外部入力データを周期的に読み取るサイクル時間
 t_{SNp} SNp Tx接続 (送信データ) の周期的開始のサイクル時間

外部通信がある入力データ領域 -> 外部通信がある出力データ領域

$$t_{ExtCo1_ExtCo2_max} = 2t_{ExtCo1} + 2t_{ExtCo2}$$

$t_{ExtCo1_ExtCo2_max}$ このデータサブパスの最大応答時間
 t_{ExtCo1} 入力データ領域からの外部入力データの周期的読み取りのサイクル時間(通信モード1での外部通信)
 t_{ExtCo2} 出力データ領域からのデータ転送の周期的開始のサイクル時間(通信モード2での外部通信)

11.14.4.4 タスクがデータソースのデータサブパス

タスク -> モジュールバス

PSS 4000ファームウェアバージョン1.2.0～1.3.x:

$$t_{\text{Task_MBUS_max}} = 2t_{\text{MBUS}} + t_{\text{ProcOM}}$$

PSS 4000ファームウェアバージョン1.4.0以降:

$$t_{\text{Task_MBUS_max}} = t_{\text{MBUS}} + t_{\text{ProcOM}}$$

$t_{\text{Task_MBUS_max}}$	このデータサブパスの最大応答時間
t_{MBUS}	装置のモジュールバスサイクル時間
t_{ProcOM}	モジュール上の出力データの処理時間

タスク -> SafetyNET p_{Trans}

$$t_{\text{Task_SNp_max}} = 2t_{\text{SNp}}$$

$t_{\text{Task_SNp_max}}$	このデータサブパスの最大応答時間
t_{SNp}	SNp Tx接続 (送信データ) の周期的開始の送信機のサイクル時間 $t_{\text{SNp (Tx)}}$

タスク1 -> タスク2

$$t_{\text{Task1_Task2_max}} = 2t_{\text{Task2}}$$

$t_{\text{Task1_Task2_max}}$	このデータサブパスの最大応答時間
t_{Task2}	タスク2のタスクサイクル時間

タスク -> 外部通信がある出力データ領域

$$t_{\text{Task_ExtCo_max}} = 2t_{\text{ExtCo}}$$

$t_{\text{Task_ExtCo_max}}$	このデータサブパスの最大応答時間
t_{ExtCo}	外部通信がある出力データ領域からデータ転送を周期的に開始するサイクル時間

11.14.4.5 1台の装置上でのデータパスの計算例

この例は、PSS 4000ファームウェアバージョン1.4.0以降のPSS 4000専用装置のものです。

この例のデータパスには、次のことが当てはまります。

- ▶ 入力データ (例: モジュールのハードウェア入力) および出力データ (例: モジュールのハードウェア出力) は同じ装置上にあります。
- ▶ データパスは次のように構成されています。
モジュールバス (入力データ) -> タスク -> モジュールバス (出力データ)
- ▶ 最大応答時間を計算するために、データパスは次のデータサブパスに分割されています。

- データサブパス1: モジュールバス (入力データ) -> タスク

データサブパス1の最大応答時間には、次のことが当てはまります。

$$t_{\text{MBUS_Task_max}} = t_{\text{ProclM}} + 2t_{\text{Task}}$$

- データサブパス2: タスク -> モジュールバス (出力データ)

データサブパス2の最大応答時間には、次のことが当てはまります。

$$t_{\text{Task_MBUS_max}} = t_{\text{MBUS}} + t_{\text{ProcOM}}$$

データパスの最大応答時間には、次のことが当てはまります。

$t_{\text{React_max}} = t_{\text{ProclM}} + 2t_{\text{Task}} + t_{\text{MBUS}} + t_{\text{ProcOM}}$
--

11.14.4.6 3台の装置を使用したデータパスの計算例

この例は、PSS 4000ファームウェアバージョン1.4.0以降のPSS 4000専用装置のものです。

この例のデータパスには、次のことが当てはまります。

- ▶ 入力データ (ハードウェア入力など) が装置Aにあります。データはSafetyNET p経由で装置Bに転送されます。
- ▶ 装置Bで、データはタスクで処理され、その後SafetyNET p経由で装置Cに転送されます。
- ▶ 装置Cで、データは出力があるモジュールに転送されます。
- ▶ データパスは次のように構成されています。

装置A上のモジュールバス (入力データ) -> SafetyNET p -> 装置B上のタスク -> SafetyNET p -> 装置C上のモジュールバス (出力データ)

- ▶ 最大応答時間を計算するために、データパスは次のデータサブパスに分割されています。

– データサブパス1: モジュールバス_(A) (入力データ) -> SafetyNET p_{Trans(A)}

データサブパス1の最大応答時間には、次のことが当てはまります。

$$t_{\text{MBUS_SNp_max(A)}} = t_{\text{ProclM}} + 2t_{\text{SNp(A)}}$$

– データサブパス2: SafetyNET p_{Rec(B)} -> タスク_(B)

データサブパス2の最大応答時間には、次のことが当てはまります。

$$t_{\text{SNp_Task_max(B)}} = t_{\text{DelaySNp}} + 2t_{\text{Task(B)}}$$

– データサブパス3: タスク_(B) -> SafetyNET p_{Trans(B)}

データサブパス3の最大応答時間には、次のことが当てはまります。

$$t_{\text{Task_SNp_max(B)}} = 2t_{\text{SNp(B)}}$$

– データサブパス4: SafetyNET p_{Rec(C)} -> モジュールバス_(C) (出力データ)

データサブパス4の最大応答時間には、次のことが当てはまります。

$$t_{\text{SNp_MBUS_max(C)}} = t_{\text{DelaySNp}} + 2t_{\text{MBUS(C)}} + t_{\text{ProcOM}}$$

データパスの最大応答時間には、次のことが当てはまります。

$t_{\text{React_max}}$	$ \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} $
-------------------------	---

11.15 システム時刻の同期

産業オートメーションシステムの装置は複雑なネットワークに組み込まれることが多いため、各装置は同じシステム時刻に合わせる必要があります。正確なデータ評価と時間が重視されるデータ処理 (例: 処理データ、安全関連タスクの応答時間、ログのタイムスタンプ) では、ネットワークの負荷やデータパケットのサイズ、通信プロトコル内のメカニズムなどで処理が遅れることは許されないため、システム時刻の一致が特に重要になります。

イーサネットベースの産業オートメーションシステムでは、日時を提供するシステムクロックを使用して、この要件に対応することができます。応用技術に応じて、少なくとも1つのリアルタイムクロックを使用します。このクロックが、オートメーションシステム (サーバ) 内に参照時刻を提供します。その他のすべてネットワークサブスクリバ (クライアント) は、この参照時刻に基づいて装置の時刻を同期することになります。この方法により、各ネットワークサブスクリバは、標準化されたシステム時刻を利用できるようになり、時刻のずれは非常にわずかに抑えられます。

11.15.1 SNTPとのシステム時刻の同期

Simple Network Time Protocol (SNTP) は、Ethernetベースのネットワークで通信と時刻の同期に使用する同期プロトコルです。SNTPはクライアント／サーバの原則に適合して動作します。これは、通信プロトコルとしてUDPを使用します。(時刻) サーバは、このサービスのために既定でポート番号123のポートを使用します。PSS 4000専用装置では、ポートは固定的に設定され、ユーザが変更することはできません。(時刻) サーバは、PAS 4000ではSNTPサーバと呼ばれます。

オートメーションシステムPSS 4000では、SNTPサーバの有無に関係なくプロジェクトを実行できます。

PAS 4000でプロジェクトに対してSNTPサーバをコンフィグレーションした場合、SNTPを使用してプロジェクト内の全装置で診断メッセージとログエントリの時刻が同じになるようにします。これにより、独自のリアルタイムクロックを持つさまざまな装置から、正しい時間の順序で診断メッセージが受信されることが保証されます。またSNTPは、装置の交換後に、自動的に時刻を同期するために使用されます。交換したユニットをSNTPサーバとして使用する場合、PAS 4000は、新しい装置で時刻を設定するためにのみ使用する必要があります。

オートメーションシステムPSS 4000は、SNTPユニキャストを通信メカニズムとしてサポートします。これは、プロジェクト内では1台のSNTPサーバのみが存在できることを意味します。その他のすべてのネットワークサブスクライバは、自動的にSNTPクライアントになります (例外: PCSおよびPMIは、SNTPクライアントとしての自動コンフィグレーションから除外できます)。

SNTPクライアントは、ユニキャスト通信メカニズムにより、時刻の同期要求をSNTPサーバに送信し、SNTPサーバから時刻の情報を受信します。

PSS 4000では、SNTPクライアントとコンフィグレーションされたSNTPサーバ間の時間同期は、サイクル時間 ≤ 60 sで実行されます。ユーザはこのサイクル時間をコンフィグレーションできません。

リアルタイムクロックを持つプロジェクト内の任意の装置を、プロジェクトのSNTPサーバとしてコンフィグレーションできます。ただし、PAS 4000は外部装置の使用もサポートします。たとえば、この目的のために特別に設計されたスイッチを使用できます。

PAS 4000内でプロジェクトに対してSNTPサーバをコンフィグレーションしていない場合、リアルタイムクロックがある各装置は、それぞれの時刻で動作することになります。したがって、診断メッセージおよびログエントリの標準的な時刻はプロジェクト内に存在しません。診断メッセージが正しい時間の順序になることは保証されません。

これはPCとPMIがSNTPクライアントとしてコンフィグレーションされていない、SNTPサーバを使用しているプロジェクトにも適用されます。これらのプロジェクトでは、各PCと各PMIがそれぞれの時刻で動作します。これは、SNTPサーバの時刻とは異なる場合があります。

11.16 メモリ

11.16.1 メモリタイプ

PSS 4000装置には、さまざまなメモリおよびメモリ領域があります。PSS 4000装置で使用するメモリタイプとメモリサイズは、装置によって異なります。ピルツのPSS 4000装置の詳細については、該当する装置の取扱説明書を参照してください。

以下のテーブルに、使用可能なメモリタイプとその内容を示します。

メモリタイプ	目次
プログラムとデータメモリ	FSリソースのプログラムとデータメモリ: ▶ FS制御プログラム ▶ 揮発性のFS変数 STリソースのプログラムとデータメモリ: ▶ ST制御プログラム ▶ 揮発性のST変数
不揮発性FSメモリ	不揮発性FS変数
不揮発性STメモリ	不揮発性ST変数
SDカード	▶ 装置プロジェクト ▶ 命名データ
セキュリティSDカード	▶ 装置プロジェクト ▶ 命名データ ▶ 特別なセキュリティ機能のデータ



重要

メモリ要件を計算するときは、管理データは常に変数およびデータと一緒に格納されることに注意してください。つまり、実際のメモリ要件をデータ型のビット幅から推定することはできません。



情報

PSS 4000装置にはSDカードの書き込みアクセスが必要です。SDカードで書き込み保護が有効な場合、PSS 4000装置は書き込み保護を無視します。

11.16.2 メモリ構造

PSS 4000ファームウェアがデータを管理できるようにするには、データメディア上のディレクトリにさまざまな作業ディレクトリが必要です。データメディア上のディレクトリは、PSS 4000データディレクトリ (PILZ_DATA_DIR) と呼ばれます。

PSS 4000専用装置では、PSS 4000データディレクトリがSDカードにあります。PSS 4000対応装置 (PMI、PCなど) では、装置にPSS 4000データディレクトリがあります。PSS 4000データディレクトリのディレクトリ構造は、すべてのPSS 4000装置で同じです。つまりPSS 4000データディレクトリには、フォルダ名が事前定義された異なるサブディレクトリが含まれます。

メモリ容量の割り付けに関しては、ディレクトリがさまざまなディレクトリタイプに区別されます。ディレクトリサイズは、無制限の場合と制限付きの場合があります。サイズ制限付きのディレクトリは、メモリ要件関連の制限がある機能で使用されます。サイズ無制限のディレクトリを使用する機能では、メモリ要件関連の制限はありません。

SDカードの場合は、次の点に注意してください。サイズ無制限のディレクトリによるメモリ使用量が多すぎて、サイズ制限付きディレクトリを使用する機能のメモリ容量が不足すると、これらの機能のパフォーマンスが低下する可能性があります。

ディレクトリ構造

サブディレクトリ	内容と使用	備考
/cfg	装置の命名データ	ディレクトリは、サイズ無制限のディレクトリの1つです。
/ps	装置プロジェクトのデータ	ディレクトリは、サイズ無制限のディレクトリの1つです。
/ep	バックアップコピー	ディレクトリは、サイズ無制限のディレクトリの1つです。
/debug	装置のデバッグデータ	ディレクトリは、サイズ制限付きディレクトリの1つです。 SDカードの場合は、次の点に注意してください。 このディレクトリのサイズ上限は100 MByteです。このメモリ容量に達すると、古いデバッグデータが削除されます。
/traces_st	STリソースおよびトレースファイルのトレースコンフィグレーション	ディレクトリは使用可能なメモリの40%を使用します。
/traces_fs	FSリソースおよびトレースファイルのトレースコンフィグレーション	ディレクトリは使用可能なメモリの40%を使用します。
/CM-Stick	セキュリティデータ	このディレクトリは、セキュリティSDカードでのみ使用可能です。

サブディレクトリ	内容と使用	備考
/userfiles	ユーザ固有のデータとフォルダ	このディレクトリは、装置やシステムなどに関連する証明書、文書、またはその他の情報をユーザが保存するために使用できます。 ディレクトリは、サイズ無制限のディレクトリの1つです。



情報

起動すると、関連しないすべてのデータとディレクトリがSDカードから削除されます。つまり、次のように処理されます。

- ユーザによって「/userfiles」ディレクトリ以外に保存された、システム由来でないデータとディレクトリは、SDカードから削除されます。
- 次のディレクトリとファイルは、PSS 4000ファームウェアの機能に必要なディレクトリとともにSDカードに保持されます。
 - /userfilesディレクトリ
 - CM-stickディレクトリ
 - SDカードのルートディレクトリ内の「journal.jnl」ファイル
- 装置のPSS 4000ファームウェアが認識している機能のディレクトリのみがSDカードに保持されることに注意してください。
たとえば、ファームウェアバージョンが高い装置のSDカードが装置で使用されている場合、ファームウェアバージョンが高い装置のみで使用できるすべての機能のディレクトリは削除されます。



重要

ファイルシステムのパフォーマンスの低下

割り付けられるメモリ容量が増えるにつれて、ファイルシステムのパフォーマンスが低下します。SDカードがほぼいっぱいになると、パフォーマンスが大幅に低下します。ファイルシステムのパフォーマンスを維持するために、少なくなったディレクトリのメモリ容量がさらに少なくなります。これは、SDカードのその時点での総容量の80%までしか使用できないためです。

/userfilesディレクトリに大量のデータを保存すると、システムのパフォーマンスが低下する可能性があります。

アプリケーションに使用するSDカードに十分なメモリ容量があることを確認してください。



重要

SDカード上の物理的に使用可能なディスク容量

SDカードに記載されているメモリサイズは、アプリケーションが物理的に使用できるメモリサイズではありません。また、SDカードには常にさまざまな管理データが保存されています。

アプリケーションが使用可能なSDカード上の物理ディスク容量は、製造上および技術的な理由により異なることに注意してください。

PAS4000でメモリ使用量を表示できます。詳細については、「オンラインヘルプのリソースの割付け」->「オンラインリソース情報の表示」を参照してください。

11.16.3 セキュリティSDカード

セキュリティSDカードには、装置プロジェクトと命名データだけでなく、セキュリティ機能に必要な特別なデータも含まれています。セキュリティ機能には特に、不正なアクセスや使用からPSS 4000プロジェクトを保護する機能が含まれます。PAS4000では、セキュリティ機能のために装置プロジェクトとセキュリティSDカードをリンクできます。

セキュリティSDカードはピルツから入手できます。たとえば「PASkey SDカード」と指定します。

11.16.4 ジャーナリングファイルシステム

すべてのPSS 4000専用装置は、SDカード上で独自のジャーナリングファイルシステムを使用します。実際に書き込まれる前に、すべての変更が特別に予約されたメモリ領域に記録されるため、停電やシステムクラッシュが発生した場合も整合性のある状態を再構築できます。



重要

SDカード上のファイルの破損

SDカードの書き込み中に、SDカードを装置から取り外したり、装置の電源をオフにすると、ファイルが破損する可能性があります。

SDカードの取り外しは、リカバリモードまたはスイッチOFF状態でのみ行ってください。

11.16.5 SDカードの関連付け

SDカードは、装置によって認識されたときに装置に関連付けられ、その後装置はSDカードにアクセスできるようになります。

PSS 4000専用装置では、装置の起動処理中または動作中に、SDカードを装置に挿入すると直ちに関連付けられます。

PSS 4000専用装置がSDカードを認識しない場合、またはジャーナリングファイルシステムのジャーナルを実行できない場合、装置はSDカードを関連付けません。この場合、ファイルシステムへのアクセス試行 (ファイルの保存、読み取り、変更、削除など) はすべて失敗します。

12 試運転

12.1 一般的な要件

規制 (安全障壁、警告サインなど) に適合させて、サイトの安全性を確保します。システムの試運転／再試運転は、有資格者だけが行うことができます。

- ▶ 関連する取扱説明書に記載された情報と仕様を参照してください

(例: 『PSS 4000安全マニュアル』、『PSSuniversal取り付けマニュアル』、使用しているモジュールの取扱説明書)。

- ▶ 試運転／再試運転の際には、設備／機械が意図せずに動いてしまうような場合も含めて、人身事故や物品の破損がないように十分に注意してください

(例: 危険な現場、または設備／機械全体に対して、必要に応じて、物理的なガードを設置します)。

12.2 初回試運転

以下の説明では、次のことを前提としています

- ▶ 設置場所、機械構造、および電気接続を決定する際に次の資料が考慮されている
 - PSSUniversal取り付けマニュアル
 - このシステムマニュアルの「SafetyNET p」の章 (必要な場合)
 - 使用しているモジュールの取扱説明書
 - 使用している標準フィールドバスの取り付けガイドライン (必要な場合)
 - リスク分析からの関連資料 (組み立て説明書、配線図など)
- ▶ 必要なすべての装置が完全に組み立てられ、配線され、接続されている
- ▶ SDカードがPSS 4000専用装置に挿入されている
- ▶ PAS4000のプロジェクトが完了している
(PAS4000オンラインヘルプを参照)
- ▶ プロジェクトがダウンロードされている
(PAS4000オンラインヘルプを参照)
- ▶ プロジェクトのダウンロード手順に従って、エラーが回復済み
(登録済みハードウェアが実際のハードウェアと一致するなど)
- ▶ 試運転に必要なすべての作業に関する説明が利用できる (リスク分析のチェックリストやテスト仕様など)
- ▶ 設備／機械への供給電圧がこの時点でオンになっていない

手順

1. 取り付けと配線の確認

設備／機械の組み立て説明書と配線図に基づいて、組み立てと配線を確認します。

リスク分析関連の取扱説明書を参照してください (『PSS 4000安全マニュアル』を参照)。

2. 供給電圧のスイッチオン

すべての装置で供給電圧を同時にオンにします。

すべてのPSS 4000専用装置で操作が可能になるには、多少時間がかかることがあります (PSS 4000専用装置ごとに最大10秒)。ステータスLEDによって、いつPSS 4000専用装置で操作が可能になるかわかります。

3. 試運転に役立つツールの使用

試運転の際に問題に対応できるように、PAS4000には便利で重要なツールが備わっています。

– 変数の監視

変数の現在の状態を動的に監視します。

– 変数の制御

変数の値は、指定された値に一度設定されます。その後は、ユーザプログラムによって変更することができます。

非安全関連変数を制御する場合でも、『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認してください。

– 変数の強制

変数の値は永久的に割り付けられ、ユーザプログラムによって変更することはできません。「変数の強制」を停止した場合、新しい値で上書きされるまで、変数では強制値が保持されます。

非安全関連変数を強制する場合でも、『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認してください。

– オンライン変更

「オンライン変更」は、プログラミングのエラーを探すときのPAS4000の補助機能です。プロジェクトのダウンロード後、ユーザプログラムの個々のPOUは変更が可能で(宣言部のアプリケーション部と一時変数)、ダウンロードを繰り返す必要はありません。オンライン変更は、無効化するまで有効なままです。

『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認することが必要です。

– 診断ログ

診断メッセージを受信した時間と消去した時間を示すログ。

操作とデバッグに関する情報も含まれています。

– 診断リスト

現在の診断メッセージのリスト(プロジェクト診断リスト、装置診断リスト)

– プロジェクトの比較

PAS4000のオリジナルプロジェクトが機械のプロジェクトと同じであるかどうかを確認するPAS4000のオプションです。

オンラインプロジェクトがオフラインプロジェクトと比較されます。チェックサム「FSプロジェクト」の違いが表示されます。

4. 安全機能と異常検出機能のテスト

プロジェクトに安全機能と異常検出機能がある場合、テストの仕様に基づいて、設備固有の安全機能と異常検出機能をすべてテストします。

指定された安全機能と異常検出機能をすべて完全にテストすることが**必要**です。

『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認することが**必要**です。

5. 混合入カインタフェース付きFSブロックのテスト

混合入カインタフェース付きFSブロックを使用していた場合は、定義されているテスト手順(妥当性テストなど)に従って、これらをテストします。

『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認することが**必要**です。

6. タスクのサイクル時間の最適化

試運転の際に、タスクのサイクル時間を最適化できます。

7. バックアップコピーの作成

プロジェクトのバックアップコピーを作成して、オリジナルのプロジェクトとしてプロジェクトを宣言し、プロジェクトのバックアップ計画に従って保存します。

8. 文書化

- － 試運転時に行ったすべての操作のログを記録し、情報を出力します。
安全関連の操作の場合、常に『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認することが必要です。
- － オリジナルプロジェクトのチェックサム「FSプロジェクト」を文書化する

9. 通常の動作

すべての装置がオンになり、エラーが存在していなかった場合、プロジェクトは正常に実行されています。

12.3 再試運転

設備／機械またはプロセスを変更したときには常に再試運転が必要です。どのような場合でも、次の変更を行うと再試運転が必要になります。

- ▶ 安全要件が変更された
- ▶ システムエラーの発生による変更
- ▶ 新しいオペレーションまたは生産要件による変更
- ▶ 次の操作による設備／機械に対する変更:
 - 部品の追加または削除 (PSSuシステム、PSSuシステムモジュールなど)
 - 部品の交換 (PSSuシステム、PSSuシステムモジュール、PSS 4000専用装置のSDカードなど)
- ▶ プロセスサイクルに対する変更

以下の説明では、次のことを前提としています

- ▶ 変更前にリスク分析が実行されている
- ▶ 部品の追加、取り外し、または交換を行う際に次の資料が考慮されている
 - PSSuniversal取り付けマニュアル
 - このシステムマニュアルの「SafetyNET p」の章 (必要な場合)
 - 使用しているモジュールの取扱説明書
 - 使用している標準フィールドバスの取り付けガイドライン (必要な場合)
 - 変更のリスク分析からの関連資料
- ▶ 修正されたすべての部品が完全に組み立て／接続され、配線されている
- ▶ SDカードがPSS 4000専用装置に挿入されている
- ▶ PAS4000でのプロジェクトに対する変更が完了している
(PAS4000オンラインヘルプを参照)
- ▶ プロジェクトがダウンロードされている
(PAS4000オンラインヘルプを参照)
- ▶ プロジェクトのダウンロード手順に従って、エラーが回復済み
(登録済みハードウェアが実際のハードウェアと一致するなど)
- ▶ 再試運転に必要なすべての作業に関する説明が利用できる (リスク分析のチェックリストやテスト仕様など)
- ▶ 設備／機械への供給電圧がこの時点でオンになっていない

手順

1. 取り付けと配線の確認

設備／機械の組み立て説明書と配線図に基づいて、変更するコンポーネントの組み立てと配線を確認します。

変更のリスク分析に関連する取扱説明書を参照してください (『PSS 4000安全マニュアル』を参照)。

2. 供給電圧のスイッチオン

すべての装置で供給電圧を同時にオンにします。

すべてのPSS 4000専用装置で操作が可能になるには、多少時間がかかることがあります (PSS 4000専用装置ごとに最大10秒)。ステータスLEDによって、いつPSS 4000専用装置で操作が可能になるかわかります。

3. 試運転に役立つツールの使用

再試運転の際に問題に対応できるように、PAS4000には便利で重要なツールが備わっています。

– 変数の監視

変数の現在の状態を動的に監視します。

– 変数の制御

変数の値は、指定された値に一度設定されます。その後は、ユーザプログラムによって変更することができます。

非安全関連変数を制御する場合でも、『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認してください。

– 変数の強制

変数の値は永久的に割り付けられ、ユーザプログラムによって変更することはできません。「変数の強制」を停止した場合、新しい値で上書きされるまで、変数では強制値が保持されます。

非安全関連変数を強制する場合でも、『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認してください。

– オンライン変更

「オンライン変更」は、プログラミングのエラーを探すときのPAS4000の補助機能です。プロジェクトのダウンロード後、ユーザプログラムの個々のPOUは変更が可能で (宣言部のアプリケーション部と一時的変数)、ダウンロードを繰り返す必要はありません。オンライン変更は、無効化するまで有効なままです。

『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認することが必要です。

– 診断ログ

診断メッセージを受信した時間と消去した時間を示すログ。

操作とデバッグに関する情報も含まれています。

– 診断リスト

現在の診断メッセージのリスト (プロジェクト診断リスト、装置診断リスト) です。

– プロジェクトの比較

FSプロジェクトに含まれるさまざまなFSチェックサムを基準として、ユーザがプロジェクトを比較することができます。

– ブロック比較

各FSブロックには、ユーザがブロック比較の基準として使用できるチェックサムが含まれています。

4. 安全機能と異常検出機能のテスト

変更の後に、安全分析で、安全機能の妥当性確認およびテストを実行する必要があることが示された場合、変更自体および一連のプロセス全体をテストする必要があります。プロジェクトに必要な変更のみが加えられていることを確認する必要があります。

次の点に注意してください。

プロジェクトの比較の際に、チェックサム「FSプロジェクト」を使用して、FSプロジェクトに変更が加えられたかどうか確認できます。「FSプロジェクト」のチェックサムに変更がない場合、FSプロジェクトは変更されていません。「FSプロジェクト」のチェックサムに変更がある場合、FSプロジェクトが変更されています。いかなる状況でも、プロジェクトの比較により、変更に対する先行の安全分析を置き換えることはできません。プロジェクトの比較は分析の確認に使用できます。

FSプロジェクトに変更がある場合は、PSS 4000安全マニュアル（「変更、メンテナンス、廃棄」の章の「変更」の項）に記載された手順に留意してください。

5. 混合入カインタフェース付きFSブロックのテスト

混合入カインタフェース付きFSブロックが変更の影響を受ける場合は、定義されているテスト手順（妥当性テストなど）に従って、これらをテストします。

『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認することが必要です。

6. タスクのサイクル時間の最適化

再試運転の際に、タスクのサイクル時間を最適化できます。

7. バックアップコピーの作成

変更したプロジェクトのバックアップコピーを作成して、変更したプロジェクトを新しいオリジナルのプロジェクトとして宣言し、プロジェクトのバックアップ計画に従って保存します。

8. 文書化

- － すべての変更の情報を記録する。
- － 再試運転時に行ったすべての操作のログを記録し、情報を出力する。
安全関連の操作の場合、常に『PSS 4000安全マニュアル』に記載された情報と要件を十分に確認することが必要です。
- － 新しいオリジナルプロジェクトのチェックサム「FSプロジェクト」を文書化する。

9. 通常の動作

すべての装置がオンになり、エラーが存在していなかった場合、プロジェクトは正常に実行されています。

13 メンテナンス、修理、廃棄

13.1 メンテナンスと修理

メンテナンスや修理を始める前に、以下の点に注意してください。

- ▶ メンテナンスと修理の作業は、有資格者にのみ許可されています。
- ▶ 設備または機械の電源が、規制に従って取り外されていることを確認します。
- ▶ 不注意による通電に備えて、設備または機械には安全対策を施す必要があります。
- ▶ お客様が部品 (PSSuシステム、PSSuシステムモジュール、PSS 4000専用装置のSDカードなど) を交換する場合は、以下に従ってください。
 - 「PSS 4000安全マニュアル」に記載された情報と要件を確認してください。
 - 部品を交換する前に、装置をSTOP状態に切り替えます。
SDカードを交換する場合、必ず電圧がオフになっているときに実行してください。
 - 他の装置がPSS 4000専用装置上のプロセスデータにアクセスする場合、他の装置では代入値が使用されることに注意してください。
 - SDカードを交換するときには、その装置向けのSDカードのみを装置に挿入するように注意してください。
 - SDカードを交換するときには、装置プロジェクトの正しいバージョンを含んだSDカードを装置に挿入するように注意してください。
 - SDカードを交換する場合、新しい装置の命名を行う必要があることに注意します (「[装置交換時の装置命名](#)」 318」を参照)。
 - 再試運転の情報を確認します (「[再試運転](#)」 483」を参照)。

PSS 4000専用装置のメンテナンスと修理

PSS 4000専用装置でのメンテナンス作業は必要ありません。故障のある装置やモジュールはピルツに送り返してください。



情報

モジュールまたはPSS 4000専用装置を交換する場合は、『PSSuniversal取り付けマニュアル』で説明している組み立て／取り外し手順に従ってください。

13.2 廃棄

FSモジュールの安全関連特性データの処理時間 t_M を守ってください。

『PSSuniversal取り付けマニュアル』および関連モジュールの取扱説明書で説明している組み立て／取り外し手順に従ってください。

PSS67装置の場合は、使用している装置の取扱説明書を参照してください。

廃棄時は、電子装置の廃棄に関する地域の規則 (廃電子・電気機器法など) に従ってください。

14 OPCサーバ (クラシック)

14.1 概要

OPC (Openness, Productivity and Collaboration) は標準化されたソフトウェアインタフェースであるため、幅広い製造元のPCベースソフトウェアでデータを交換することができます。このOPCによる基盤は、規格の維持と展開に大きな役割を果たします。現在ビルツを含めて、約100社が参加しています。

OPC規格は現在「OPCクラシック」と呼ばれ、プラットフォームによっては、Windowsオペレーティングシステム用に指定されます。この規格は、OLE (Object Linking and Embedding)、COM (Component Object Model)、DCOM (Distributed COM) など、Windowsの技術に基づいています。

ネットワーク化された設備において、OPCサーバは通常、診断データやプロセスデータなどを収集し、定義されたネームスペースでこれらを表示します。OPCクライアントは、一般的にプロセスの視覚化やオペレーティングデータの記録に使用されます。このクライアントは、OPCサーバが提供するデータにアクセスしてプロセスを実行します。データを処理するには、OPCクライアントに関連するOPC対応ソフトウェアが含まれている必要があります。

オートメーションシステムPSS 4000では、PCが、PSS 4000ファームウェアをインストールすることで、システムセクションPSS 4000 OPCサーバ (クラシック) を取得します。PSS 4000ファームウェアをインストールすると、PCは「サードパーティのPSS 4000対応装置」になります (「[システムの概要](#)」 20)を参照)。

PSS 4000 OPCサーバ (クラシック) には、次のような境界条件があります。

- ▶ PSS 4000 OPCサーバ (クラシック) は、OPCクラシッククライアントからのアクセスのためのさまざまなOPCクラシックプロパティをサポートしています (「[OPCプロパティ](#)」 491)を参照)。
- ▶ PSS 4000 OPCサーバ (クラシック) は、PCにインストールできます (「[システム要件](#)」 492)を参照)。
- ▶ PAS4000で作成された各プロジェクトには、最大4つのOPCサーバを含めることができます。プロセスで、OPCクラシックサーバとOPC UAサーバを組み合わせることができます。
- ▶ 複数のクライアントとともにOPCサーバに同時にアクセスすることができます。
次のことが該当します。
 - OPCクラシッククライアントのみがOPCクラシックサーバにアクセスできます。
 - OPC UAクライアントのみがOPC UAサーバにアクセスできます。
- ▶ PSS 4000 OPCサーバ (クラシック) のプロジェクトベースの全設定はPAS4000で設定され、プロジェクトのダウンロード時に装置プロジェクトとしてPSS 4000 OPCサーバ (クラシック) に転送されます。

14.1.1 設計オプション

最小限のコンフィグレーションでは、OPCサーバは視覚化ソフトウェアとともに、OPCクライアントと同じPCに配置されます。

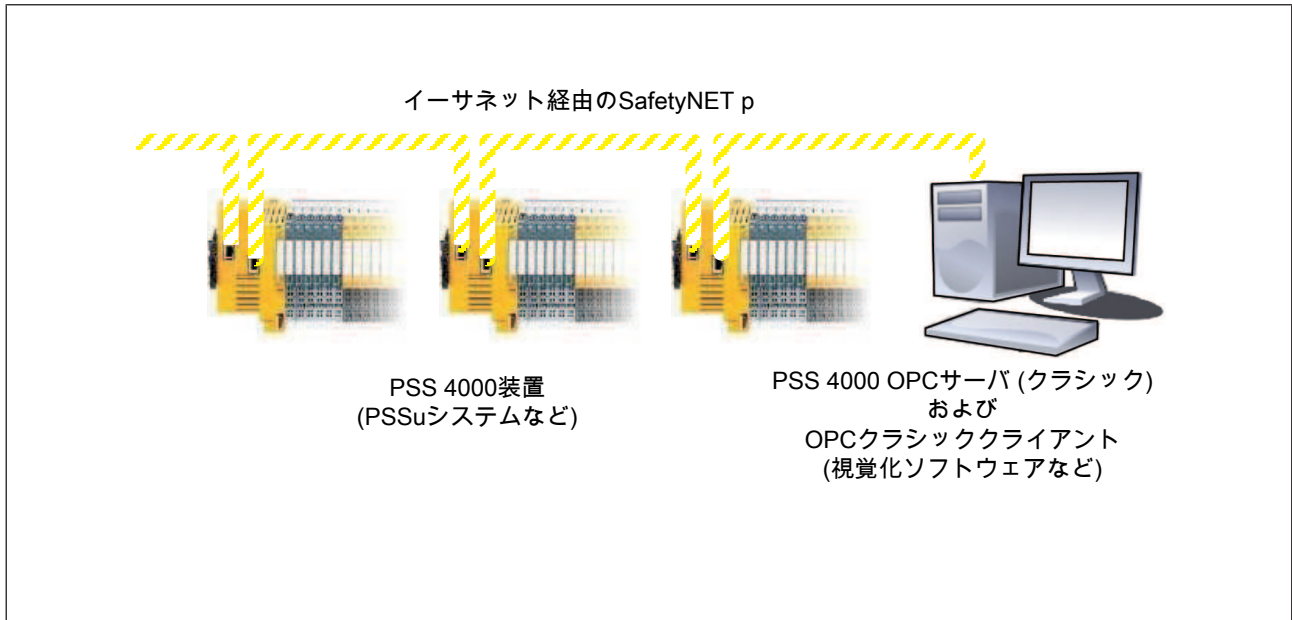


図: 最小限のOPCクラシックコンフィグレーションでの処理の視覚化 (原則)

その他のタイプの組み合わせも考えられます (以下を参照)。

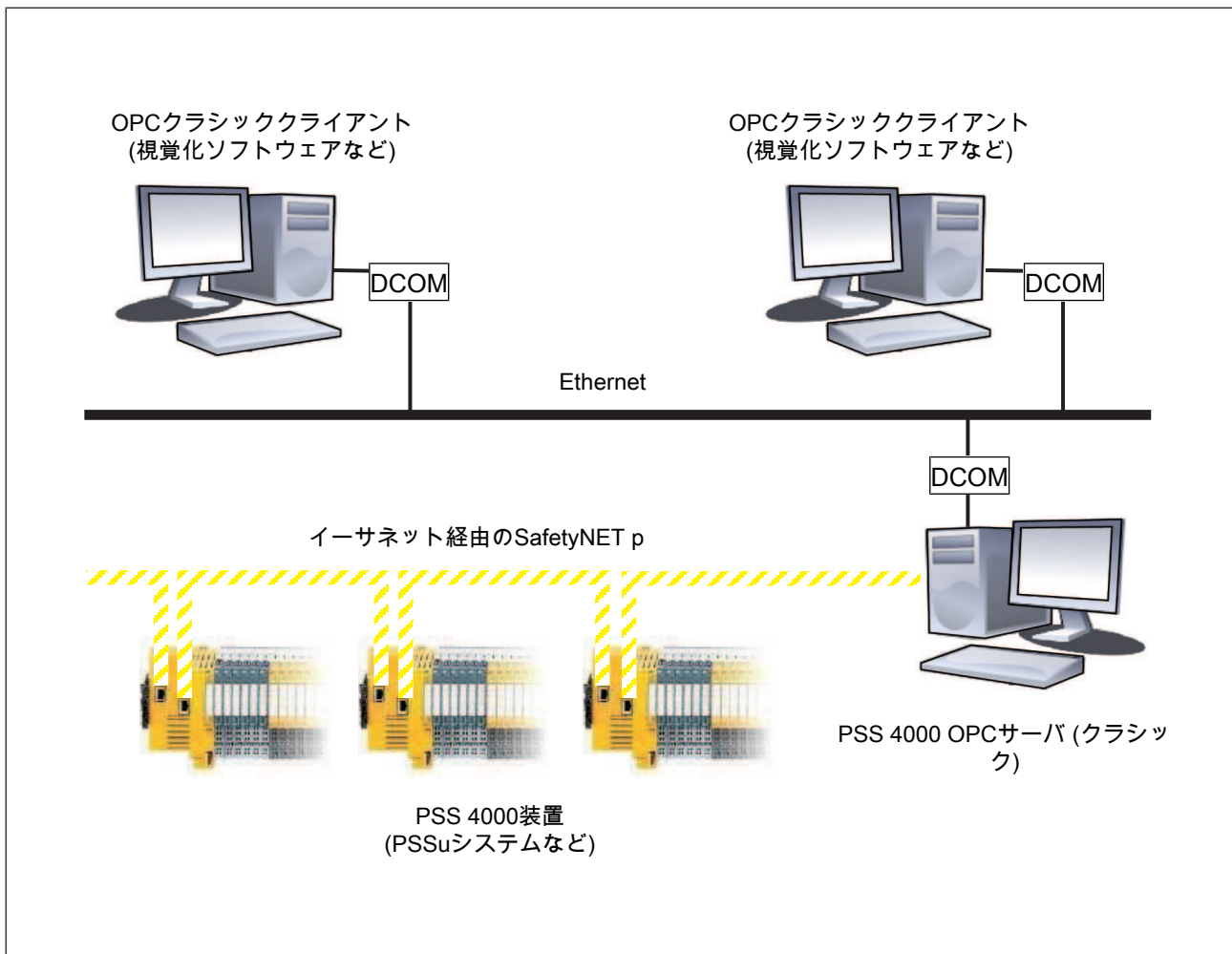


図: イーサネット経由でのOPCクラシックサーバとOPCクラシッククライアントのネットワーク化 (原則)

14.1.2 OPCプロパティ

PSS 4000 OPCサーバ (クラシック) は、次のOPCプロパティをサポートして、OPCクラシッククライアントからのアクセスを実現します。

- ▶ 次を含むOPC仕様
 - Data Access 1.0.a
 - Data Access 2.05
 - Data Access 3.0
- ▶ 次を含むOPCインタフェース
 - データアクセス仕様に適合したすべての「必須インタフェース」
 - ネームスペースでデータを処理するためのシンボリックアクセス
- ▶ OPCデータ型
 - 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

14.2 システム要件

PSS 4000ファームウェアを装置にインストールできるためには、さまざまなシステム要件を満たす必要があります。

14.2.1 Microsoft Windowsオペレーティングシステム搭載PC

PSS 4000ファームウェアは、Microsoft Windowsオペレーティングシステムを搭載したPCにインストールできます。

▶ 以下のいずれかのバージョンのMicrosoft Windows 10

- Windows 10 Home (64ビット)
- Windows 10 Pro (64ビット)
- Windows 10 Enterprise (64ビット)

14.3 **ライセンス登録**

OPCサーバのライセンスは、PAS4000でプロジェクトのライセンスが登録されると有効になります。ライセンスの詳細については、PAS4000のオンラインヘルプを参照してください。

14.4 取り付け

最新のインストールファイルは、ピルツホームページのダウンロードエリア (www.pilz.comのダウンロードページ) から入手できます。たとえば「ソフトウェア」を検索語句として入力できます。

ソフトウェアのダウンロードは、登録ユーザのみが実行できます。未登録ユーザの方は、無料で登録ができます。登録には有効なEメールアドレスが必要です。

PSS 4000ファームウェアのためのインストールキットは、次のアプリケーションエリアごとに用意されています。

- ▶ PSS 4000ファームウェアがPCにインストールされます。

必要なインストールセット:

PSS_4000_PC_<バージョン番号>_<ID>.zip

(e.g. PSS_4000_PC_1_13_0_3000763B29.zipなど)

14.4.1 PCでのPSS 4000ファームウェアのインストール

PSS 4000ファームウェアは、診断ユニットとして機能するPCにインストールされます。診断ユニットは、同時に表示ユニットとしても使用できます。PSS 4000ファームウェアをインストールするときには、PSS 4000 OPCサーバ (クラシック) およびPSS 4000 OPCサーバUAの両方がインストールされます。

PSS 4000 OPCサーバ (クラシック) のOPCコアコンポーネントも自動的にインストールされます。OPCコアコンポーネントは、OPCサーバ向けのさまざまな補助的プログラムとソフトウェアコンポーネントで構成されています。PSS4000ファームウェアコントローラもインストールされます。

PSS 4000 OPCサーバUAの場合、公開キーを含む自己署名証明書および関連付けられたプライベートキーが、インストール時に作成されます。



情報

- OPCサーバをPAS4000と同じPCにインストールする場合、PAS4000ファームウェアのバージョンをPSS 4000ファームウェアのバージョンと同じにする必要があります。
- OPCサーバをPAS4000と同じPCにインストールする場合、PAS4000を最初にインストールし、次にPSS 4000ファームウェアをインストールすることをお勧めします。
- OPCサーバをPAS4000と同じPCにインストールする場合、PSS 4000ファームウェアはインストール中にシステム変数PILZ_SNP_MULTICAST_TTLを64に設定します。
- OPCサーバをPAS4000と同じPCにインストールする場合、PAS4000がOPCサーバ上のプロジェクトに影響する可能性があります。PAS4000のオンラインアクションが別のプロジェクトで実行された場合、OPCサーバ上のプロジェクトが使用不可になり、OPCクライアントとの接続が切断される可能性があります。
- OPCクライアント (クラシック) は、PSS 4000 OPCサーバ (クラシック) にのみアクセスできます。
- OPC UAクライアントは、PSS 4000 OPCサーバUAにのみアクセスできます。



重要

OPCサーバの起動

OPCクライアントがOPCサーバにアクセスできるためには、最初にサーバを起動する必要があります。この操作を行うには、さまざまな手順があります。

– [スタートアップ] メニューからの自動起動

PSS 4000ファームウェアのインストール時に、インストールウィンドウ内の既定の設定が維持されています。つまり、Windowsオペレーティングシステムの [スタートアップ] メニューに [PSS 4000ファームウェアコントローラ] が登録されます。PCを起動するたび、OPCサーバが自動的に起動されます。

– 手動起動

インストール時に、インストールウィンドウで [スタートアップ] メニューへの登録のためのオプションは選択されていません。これは、インストール後およびPCを起動するたびに、PSS 4000ファームウェアコントローラ経由でOPCサーバを手動で起動する必要があることを意味します。

インストールプログラムを使用したインストール

⇒ インストールファイル「PSS_4000_PC_<バージョン番号>_Installer.exe」をダブルクリックして、画面上の指示に従います。

アンインストールプログラムを使用したアンインストール

⇒ Windowsプログラムリストで、[Pilz] -> [PSS4000 Firmware for PC <バージョン番号>] を右クリックし、[アンインストール] を選択します。

サイレントインストール

PSS 4000ファームウェアを無人（「サイレント」）でインストールすることもできます。この場合、ユーザの操作なしでインストールが実行され、インストールプロセスはバックグラウンドで実行されます。

⇒ Windowsコマンドライン、Windows検索フィールドまたはバッチファイルに以下のいずれかのコマンドを入力して、インストールを開始します。

既定値を使用したインストールのコマンド:

"<インストールファイルの絶対ディレクトリ>" /S

例:

"c:¥myFiles¥PSS_4000_PC_<バージョン番号>_Installer.exe" /S

仕様を指定してインストールするためのコマンド:

"<インストールファイルの絶対ディレクトリ>" /S /auto='<値>' /lang='<ツールの言語>' /D='<絶対インストールディレクトリ>'

例:

"c: ¥myFiles¥PSS_4000_PC_ <バージョン番号>_Installer.exe" /S /auto='1' /lang='de' /D=C: ¥myPrograms

サポートされている引数:

引数	意味	大文字／小文字の区別
/S	ファイルをサイレントに実行します。	はい
/auto	この引数によって、PCの起動時にPSS 4000ファームウェアを自動的に起動するかどうかを制御できます。引数はオプションです。 値の範囲: 1: PCの起動時にPSS 4000ファームウェアが自動的に起動する 0: PSS 4000ファームウェアが自動的に起動しない 既定値: 1	なし
/lang	メッセージの言語とアンインストールプログラムの言語 (アンインストールがサイレント実行されていない場合)。引数はオプションです。 値の範囲: de: ドイツ語 en: 英語 既定値: オペレーティングシステムの言語がドイツ語の場合は、ドイツ語がデフォルトの言語になり、それ以外の場合は英語になります。	いいえ
/D	インストールディレクトリ: 最後の引数として入力する必要があります。引数はオプション 既定値: %programfiles(x86)%¥PILZ	はい

注: インストールコマンドに「Start /wait」を追加することで、インストールコマンドの実行が完了するまで、それ以上のコマンドが開始されないようにすることができます。「start」と「/wait」の間には二重引用符が2つ必要です。

例:

Start "" /wait "c: ¥myFiles¥PSS_4000_PC_ <バージョン番号>_Installer.exe" /S /auto='1' /lang='de' /D=C: ¥myProgramms

サイレントアンインストール

アンインストールはサイレントに実行することもできます。

⇒ Windowsコマンドライン、Windows検索フィールドまたはバッチファイルに以下のコマンドを入力して、アンインストールを開始します。

アンインストールコマンド:

" <アンインストールファイルの絶対ディレクトリ>" /S

例:

"C: ¥Program Files (x86)¥Pilz¥PAS4000 1.20.0¥uninstallerData¥uninstall.exe" /S

注: インストールコマンドに「Start /wait」を追加することで、アンインストールコマンドの実行が完了するまで、それ以上のコマンドが開始されないようにすることができます。「start」と「/wait」の間には二重引用符が2つが必要です。

例:

Start "" /wait "C: ¥Program Files (x86)¥Pilz¥PAS4000 1.20.0¥uninstallerData¥uninstall.exe" /S

14.4.1.1 PSS 4000ファームウェアの初回インストール

初回のインストール時には、次の手順に従います。

1. PSS 4000のファームウェアをインストールするPCを起動します。

2. まだ次の手順を実行していない場合:

PC上でディレクトリを作成し、このディレクトリにインストールキット (ZIPファイル) をコピーして内容を展開します。次のファイルが使用できるようになります。

⇒ PSS_4000_PC_<バージョン番号>_Installer.exe
(たとえば、PSS_4000_PC_1_14_0_Installer.exe)

3. EXEファイルをダブルクリックします。

画面上の指示に従ってインストールを完了します。

以下のタスクがPCで実行されます。

- PSS 4000 OPCサーバ (クラシック) が、OPCコアコンポーネントとともにインストールされます。
- PSS 4000 OPCサーバUAがインストールされます。
公開キーを含む自己署名証明書および関連付けられたプライベートキーが作成されます。
- PSS 4000ファームウェアコントローラがインストールされます。
- ソフトウェアがレジストリファイルに登録されます。
- PSS 4000ファームウェアコントローラのステータスを表示するアイコンがタスクバーに追加されます。
状態 (起動／停止) の変更は、アイコンを右クリックして選択できます。

これでインストールは完了です。



情報

PSS 4000 OPCサーバ (クラシック) の場合は、以下の点に注意してください。(PCのレジストリファイルへの) 登録時に、インストールプログラムによってOPCサーバのDCOM設定が設定されます。PCに初めてOPCサーバをインストールした場合、「匿名リモートアクセス」のDCOM設定は、PCの再起動後に初めて有効になります。

14.5 PCの通信とセキュリティ設定

ここで説明する設定は、次の場合にのみ必要になります。

- ▶ PSS 4000 OPCサーバ (クラシック) をPC上で実行する場合、かつ
 - ▶ PC上で実行するOPCクラシッククライアントがイーサネット経由でOPCサーバにアクセスする場合
- 例

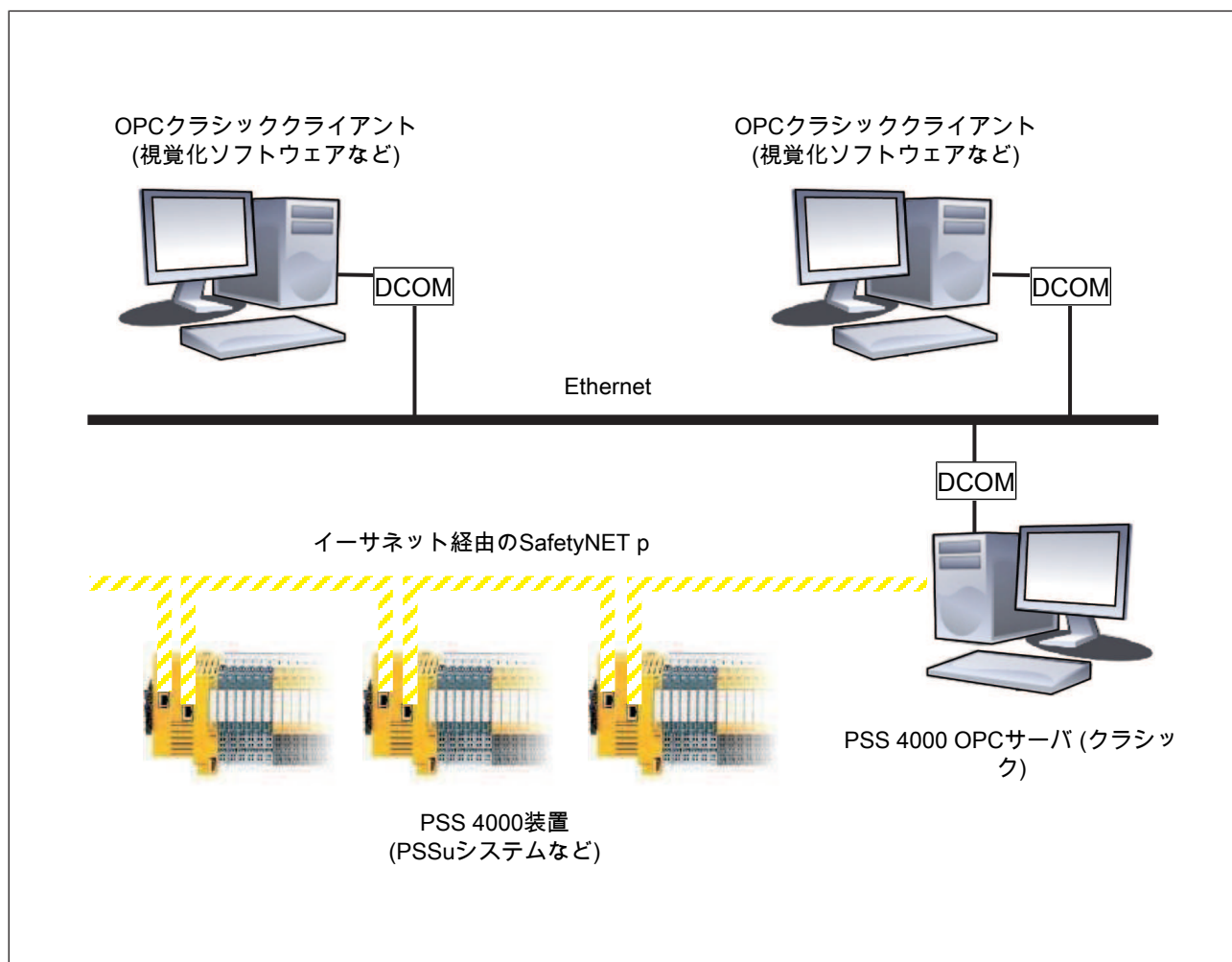


図: イーサネット経由でPSS 4000 OPCサーバ (クラシック) にアクセスするOPCクラシッククライアント

OPCクラシッククライアントがイーサネット経由でPSS 4000 OPCサーバ (クラシック) にアクセスできるようにするには、データ交換に使用するすべてのPCに対して、DCOMコンフィグレーションでさまざまな設定を行う必要があります。これらの設定については、これ以降で説明します。

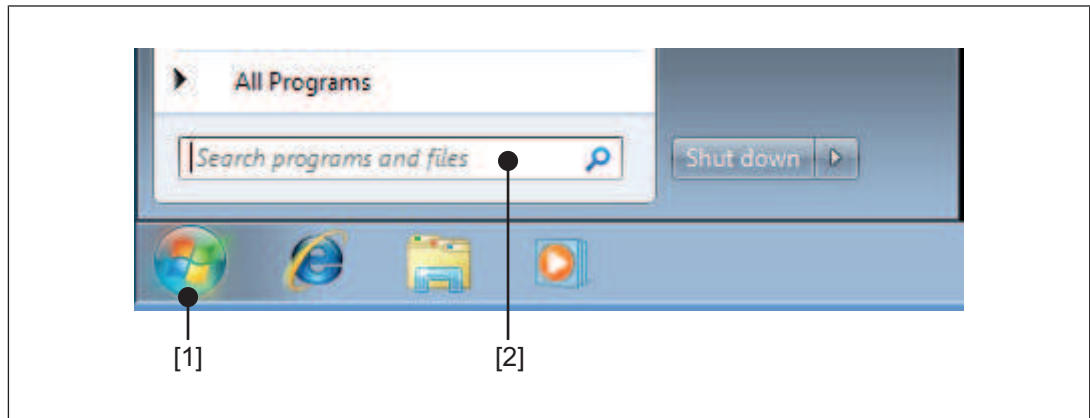
DCOMでは、PC間で認証されたアクセスのみが許可されます。したがって、PC (OPCサーバ、OPCクライアント) を同じドメインに登録することをお勧めします。

DCOMコンフィグレーションの設定は、他のネットワークユーザにOPCサーバへのアクセス権を付与するものです。不適切なコンフィグレーションにより、オペレーティングシステムの安全性と安定性が低下する可能性があります。ネットワーク管理者は、これらのコンフィグレーションを最適化して、不正アクセスからOPCサーバを保護することができます。

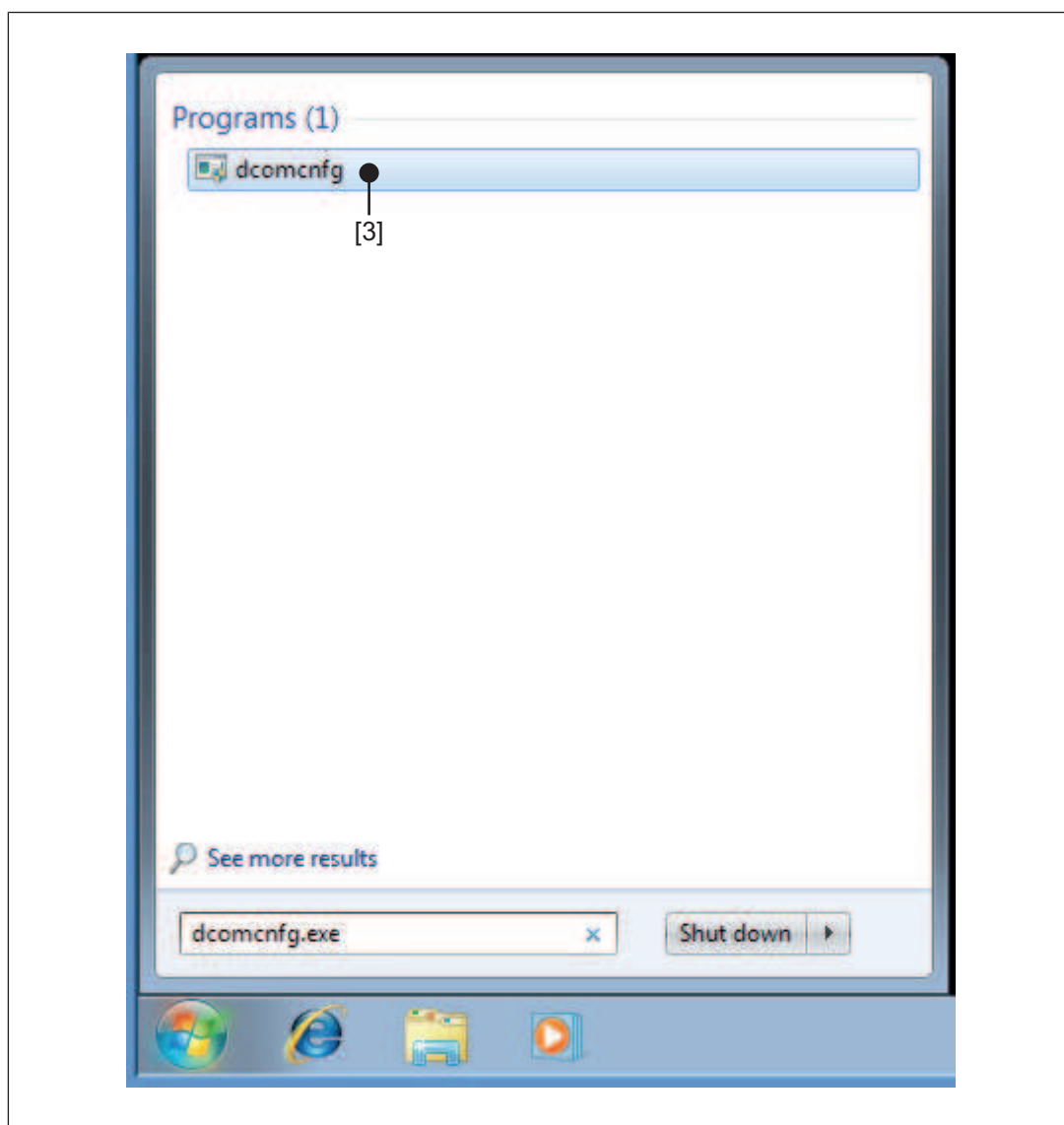
14.5.1 DCOMコンフィグレーション

以下のセクションでは、必要な設定およびコンフィグレーションについて、Microsoft Windows 7のダイアログボックスを使用して説明しています。表示されるダイアログボックスは、Windowsのバージョンとサービスパックによって異なる場合があります。Windowsヘルプでは、詳細なガイドラインも確認できます。

14.5.1.1 DCOMコンフィグレーションの開始



- ▶ Windowsのスタートシンボル [1] をクリックして、検索条件として「dcomcnfg.exe」 [2] を入力します。ウィンドウが開き、プログラムdcomcnfgが検索結果として表示されます。



- ▶ プログラム「dcomcnfg」 [3] を選択して、この選択を確認します。
[コンポーネントサービス] ウィンドウが開きます。

コンポーネントサービス

[コンポーネントサービス] ウィンドウのナビゲータ (ウィンドウの左側の列) でディレクトリを開きます。

Console Root / Component Services / Computers / My Computer

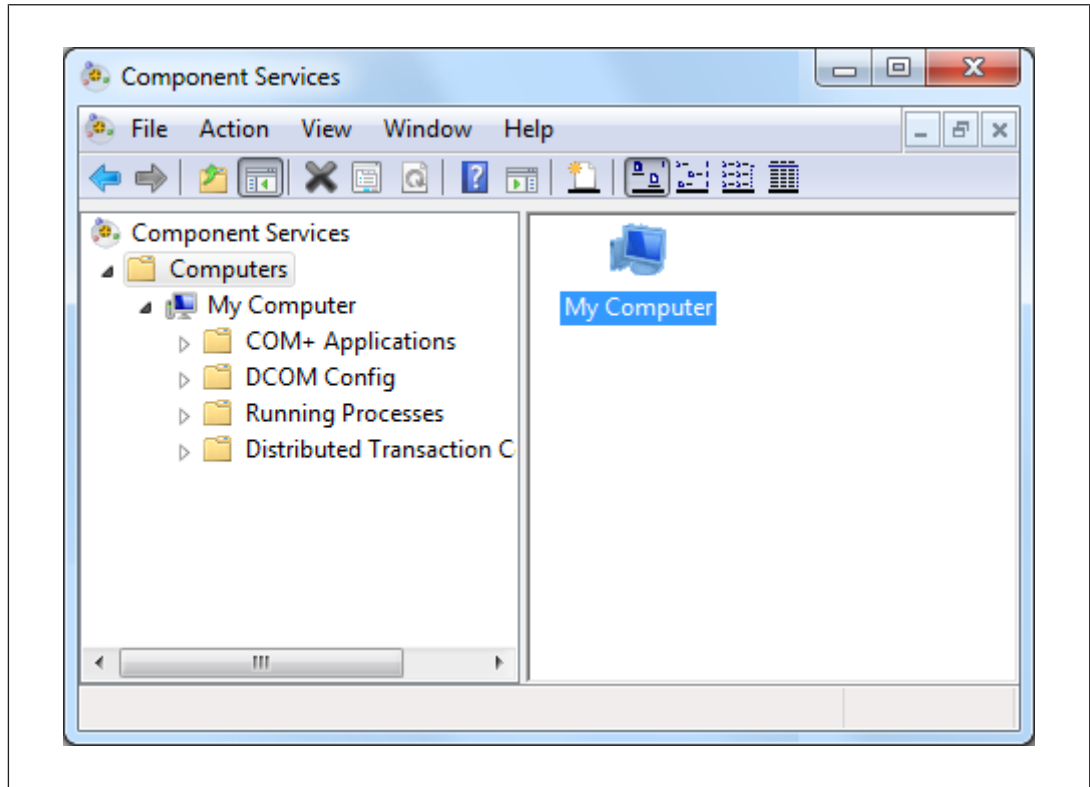


図: [コンポーネントサービス] ウィンドウ

- ▶ コンフィグレーションの詳細については、次のセクションを参照してください。
 - OPCサーバとOPCクライアントのDCOMコンフィグレーション [\[📖 504\]](#)
 - OPCサーバのDCOMコンフィグレーション [\[📖 510\]](#)
 - OPCサーバのOPCEnumコンフィグレーション [\[📖 519\]](#)

14.5.1.2 OPCサーバとOPCクライアントのDCOMコンフィグレーション

以下の通信設定とセキュリティ設定を、データ交換に使用するすべてのPC (OPCサーバ、OPCクライアント)で行う必要があります。これらの設定は「全般DCOMコンフィグレーション」と呼ばれ、特定のPCに対して他のPCが持つことのできるアクセス権を指定します。

- ▶ **[コンポーネントサービス]** ウィンドウを開きます。
(「**DCOMコンフィグレーションの開始**」[\[501\]](#)」を参照してください。)
- ▶ **[マイコンピュータ]** を右クリックして **[プロパティ]** を選択します。
[マイコンピュータのプロパティ] ウィンドウが開きます。

[既定のプロパティ] タブ

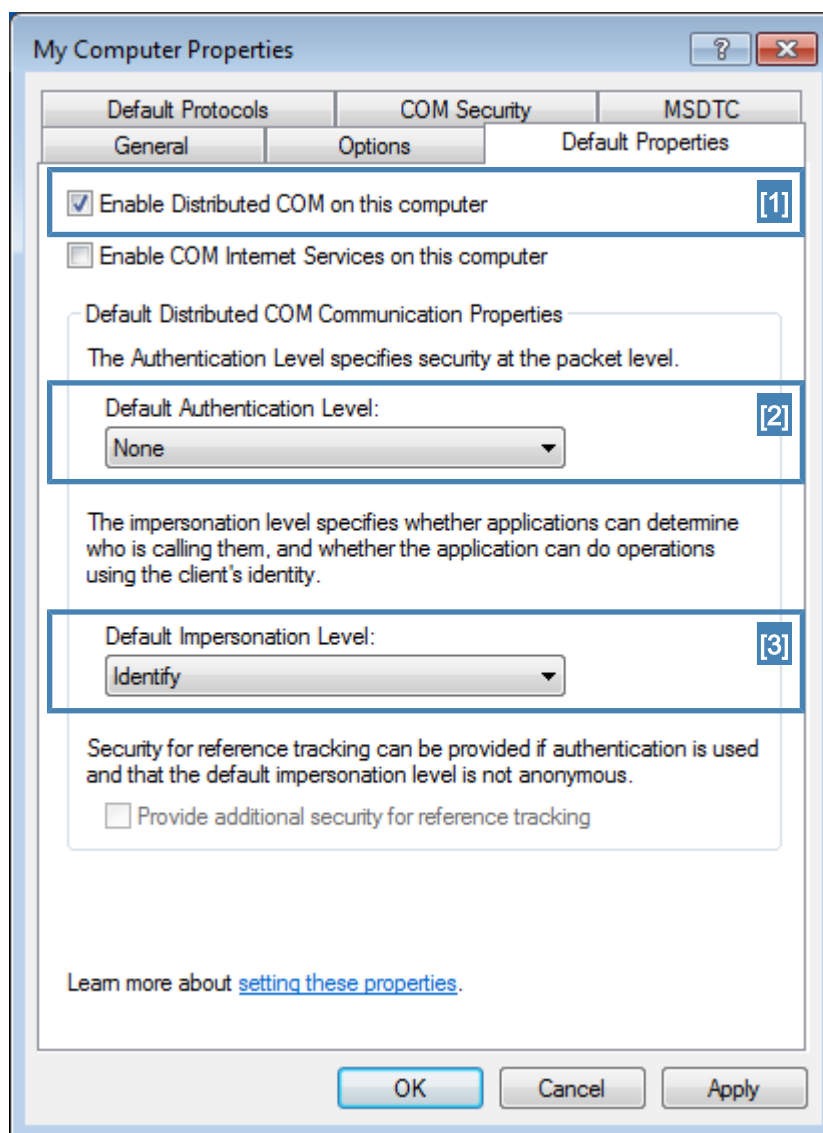


図: [既定のプロパティ] タブ

- ▶ **[このコンピュータ上で分散COMを有効にする]** をオンにします [1]。

- ▶ **[既定の認証レベル]** で **[なし]** を選択します [2]。
- ▶ **[既定の偽装レベル]** で **[IDを承認]** または **[ID]** を選択します [3]。

[既定のプロトコル] タブ

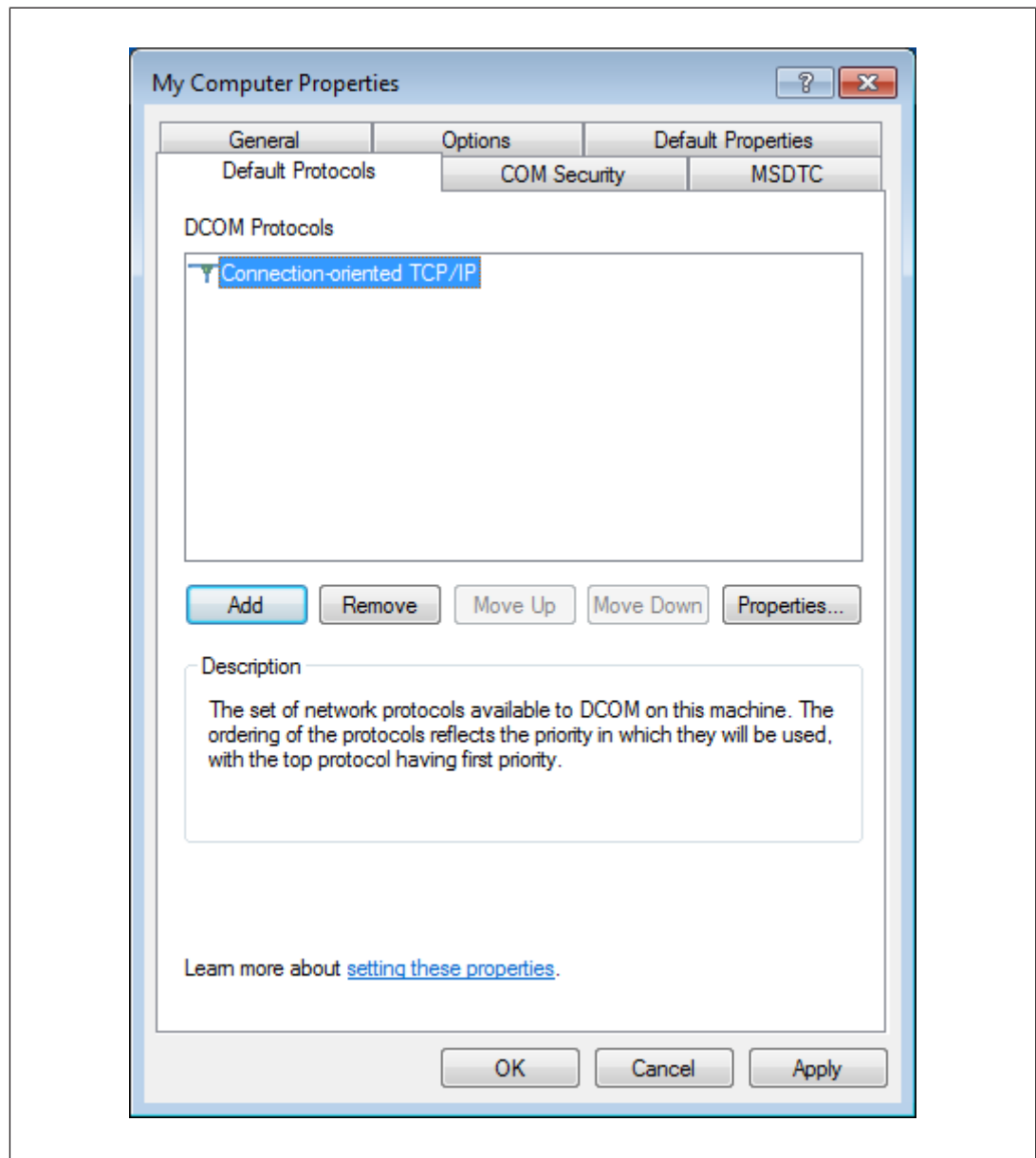


図: [既定のプロトコル] タブ

- ▶ 接続プロトコル **[接続指向TCP/IP]** が [既定のプロトコル] タブのリストに表示されない場合、次の手順でこの項目を追加できます。
 - **[追加]** ボタンをクリックします。
 - **[DCOMプロトコルの選択]** ダイアログボックスが開きます。
 - **[プロトコルシーケンス]** から **[接続指向TCP/IP]** プロトコルを選択して **[OK]** をクリックします。
 - **[既定のプロトコル]** タブが再び有効になります。
- ▶ **[接続指向TCP/IP]** プロトコルを強調表示して、リストの一番上に移動します。

この手順で、接続のセットアップ時間が短縮されます。

- ▶ 不要なプロトコルエントリは、接続の確立時に遅延の原因となるため削除します。

[COMセキュリティ] タブ

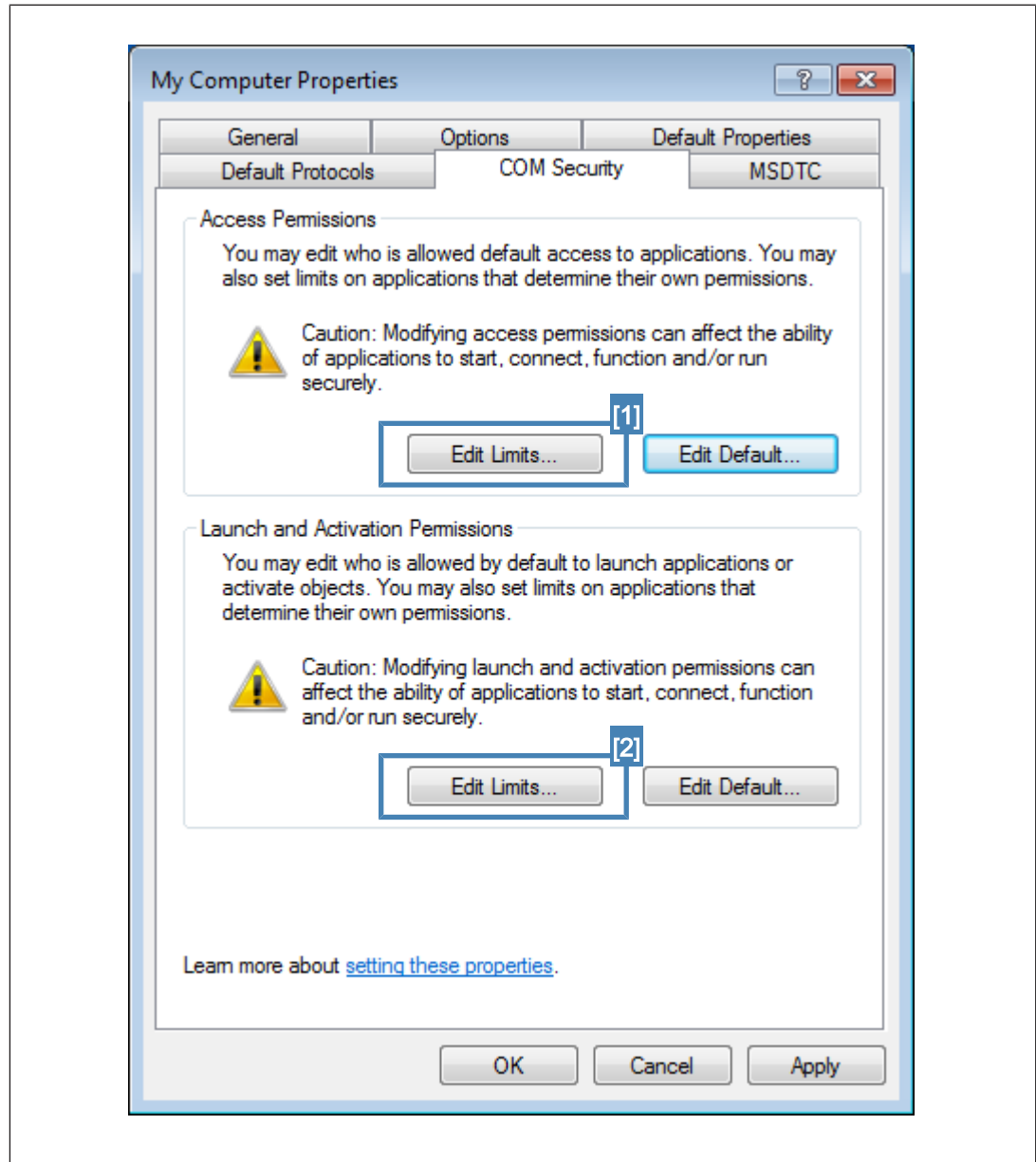


図: [COMセキュリティ] タブ

- ▶ **[アクセス許可]** の **[制限の編集]** をクリックします [1]。
[アクセス許可] ウィンドウが開きます。

[COMセキュリティ] ウィンドウ -> [アクセス許可]

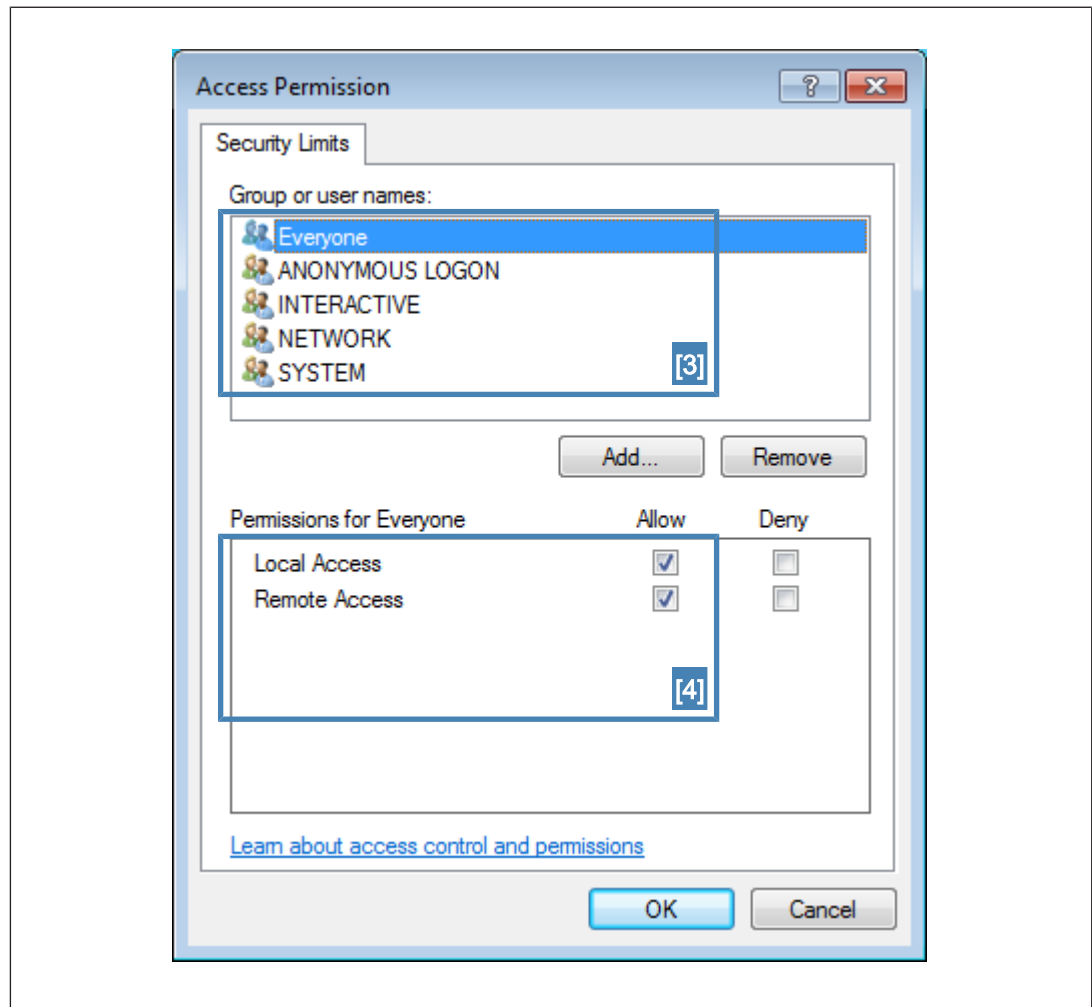


図: [COMセキュリティ] ウィンドウ -> [アクセス許可]

- ▶ 以下のすべてのグループ名またはユーザ名 [3] が使用可能になっているかどうかを確認します。
 - ANONYMOUS LOGIN
 - INTERACTIVE
 - 任意
 - NETWORK
 - SYSTEM
- ▶ グループ名またはユーザ名のいずれかが使用可能になっていない場合、その名前を追加します ([追加...] - [詳細設定] - [今すぐ検索])。

NETWORKユーザには、追加情報があることに注意してください (「ユーザ「NETWORK」の特別な機能」[529](#)を参照)。

[OK]で選択を確定します。

[アクセス許可] ウィンドウが再び有効になります。
- ▶ 各ユーザ [3] を個別に選択し、[許可] をオンにしてアクセスを許可します [4]。
 - ローカルアクセス

– リモートアクセス

▶ **[OK]** をクリックします。

[COMセキュリティ] タブが再び有効になります。

▶ **[COMセキュリティ]** タブの **[起動とアクティブ化の許可]** で **[制限の編集]** をクリックします [2]。
[起動とアクティブ化の許可] ウィンドウが開きます。

[COMセキュリティ] ウィンドウ -> **[起動とアクティブ化の許可]**

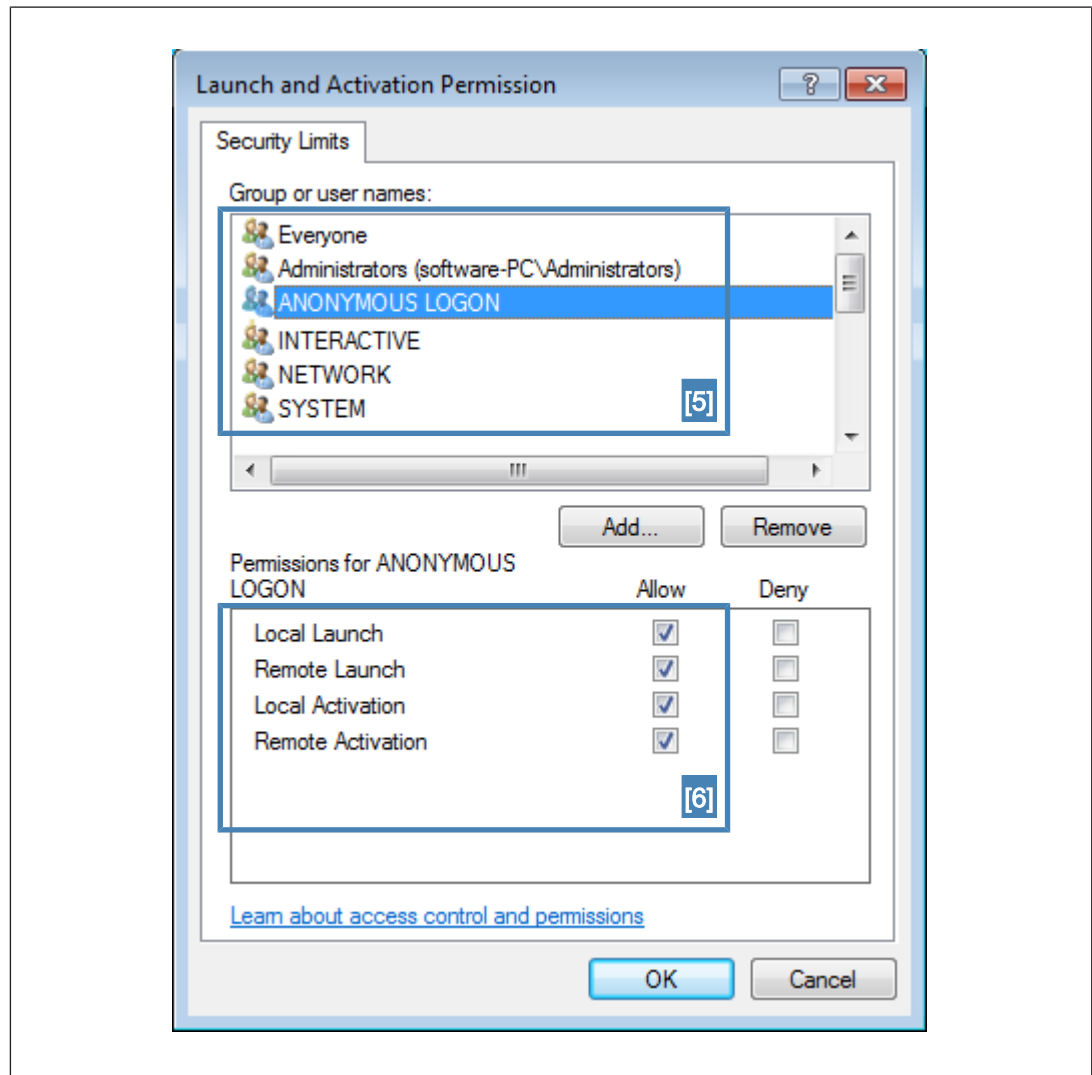


図: **[COMセキュリティ]** ウィンドウ -> **[起動とアクティブ化の許可]**

▶ 以下のすべてのグループ名またはユーザ名 [5] が使用可能になっているかどうかを確認します。

- Administrators (...)
- ANONYMOUS LOGON
- INTERACTIVE
- 任意
- NETWORK

– SYSTEM

- ▶ Administratorsユーザを使用できない場合は、このユーザを追加します。Administratorsユーザを追加するには、検索パスを調整する必要があります (「ユーザ「Administrators」の特別な機能  528」を参照)。
- ▶ 他のグループ名またはユーザ名のいずれかが使用可能になっていない場合は、その名前を追加できます ([追加...] - [詳細設定] - [今すぐ検索])。

NETWORKユーザには、追加情報があることに注意してください (「ユーザ「NETWORK」の特別な機能  529」を参照)

[OK] で選択を確定します。

[起動許可]ウィンドウが再び有効になります。

- ▶ 各ユーザ [5] を個別に選択し、[許可]をオンにして許可します[6]。
 - ローカルからの起動
 - リモートからの起動
 - ローカルからのアクティブ化
 - リモートからのアクティブ化

- ▶ [OK] をクリックします。

[COMセキュリティ] タブが再び有効になります。

- ▶ [OK] をクリックしてすべての設定を保存し、[マイコンピュータのプロパティ] ウィンドウを閉じます。
この手順で、このワークステーションの全般DCOMコンフィグレーションは完了です。
- ▶ [コンポーネントサービス] ウィンドウを閉じます。
この設定は、次回のPCの再起動時に有効になります。

14.5.1.3 OPCServerのDCOMコンフィグレーション

以下の設定は、OPCサーバのインストール時に自動的に設定されます。エラーの場合はこれらの設定を確認し、表示された設定を必要に応じて変更します。これらの設定が該当するのは、OPCサーバを搭載したPCのみです。

- ▶ **[コンポーネントサービス]** ウィンドウを開きます。
(「**DCOMコンフィグレーションの開始**」[\[501\]](#)を参照してください。)
- ▶ **[コンポーネントサービス]** ウィンドウで**DCOM Config**ディレクトリ **[1]** を開きます。

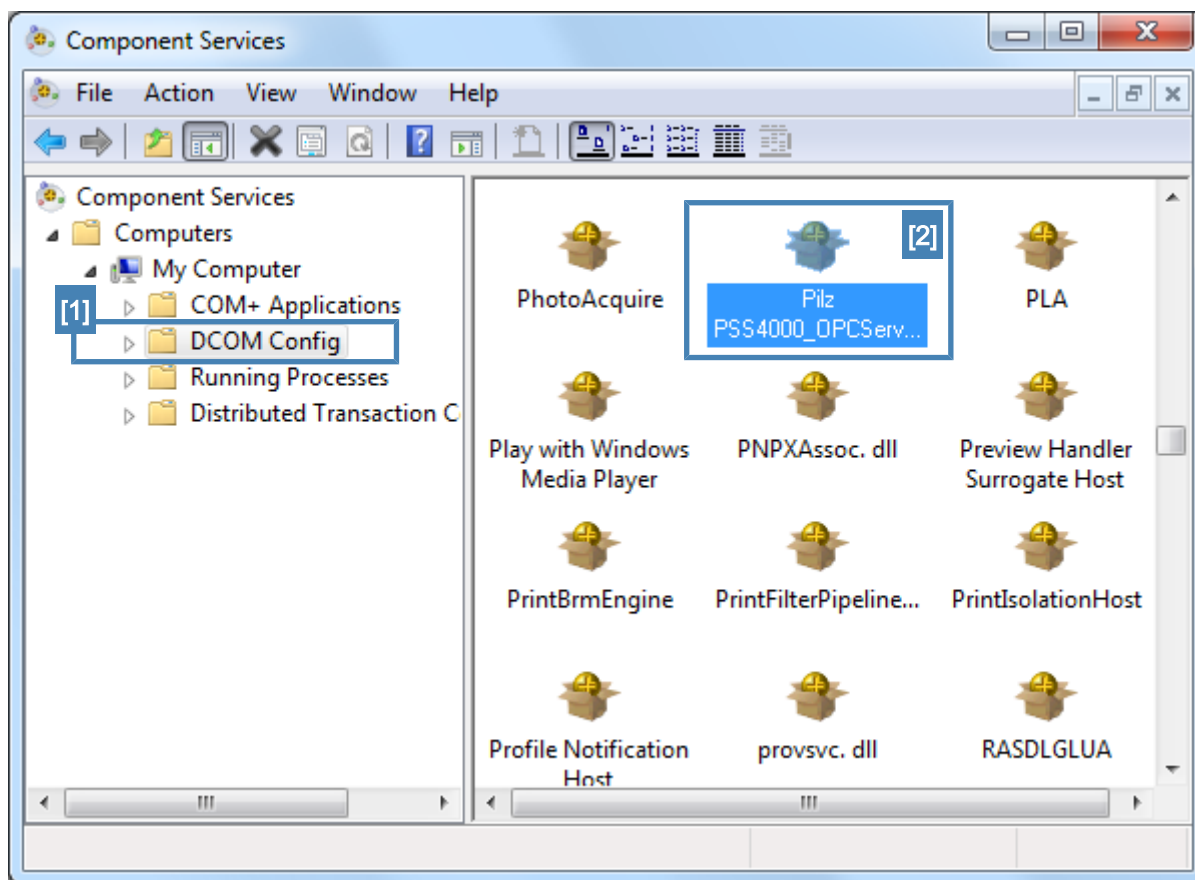


図: **[コンポーネントサービス]** ウィンドウ (OPCサーバのDCOMコンフィグレーション)

- ▶ アプリケーション**Pilz PSS4000_OPCTServerDA.DA** **[2]** を右クリックして、**[プロパティ]** を選択します。
[Pilz PSS4000_OPCTServerDA.DAのプロパティ] ウィンドウが開きます。

[一般事項] タブ

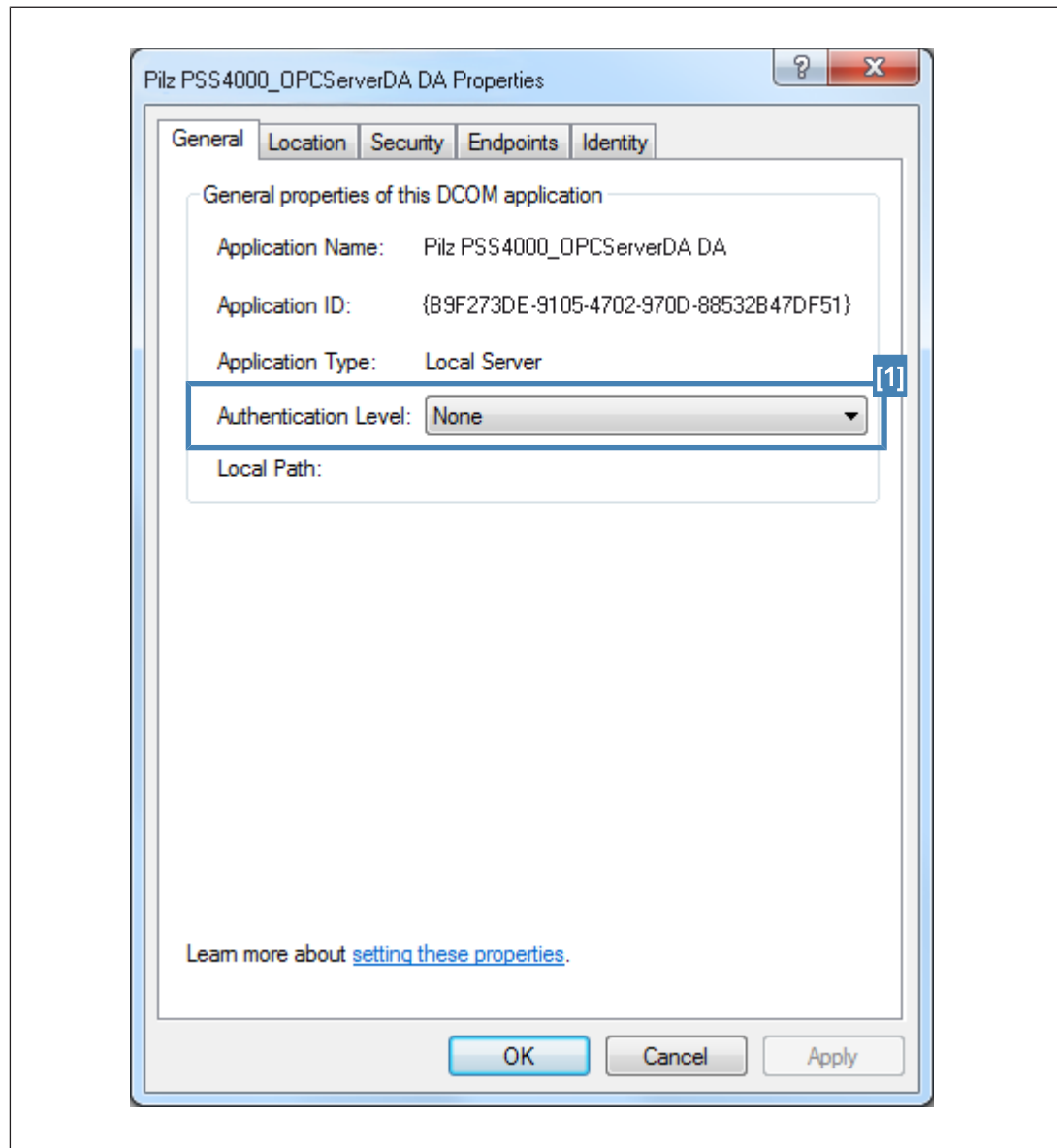


図: [全般] タブ (OPCサーバのDCOMコンフィグレーション)

- ▶ [認証レベル] で [なし] を選択します [1]。

[ロケーション] タブ

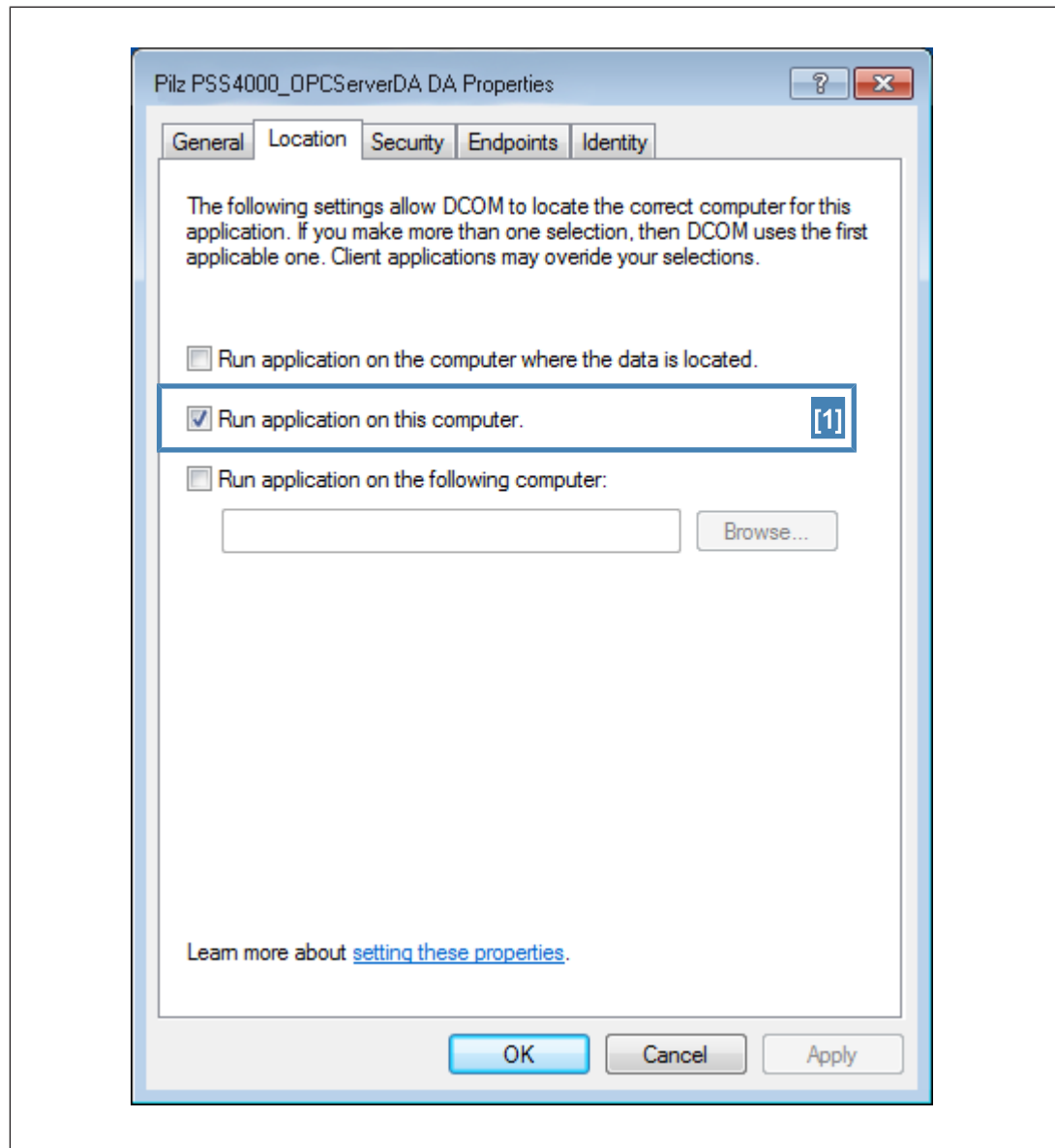


図: [ロケーション] タブ (OPCサーバのDCOMコンフィグレーション)

- ▶ **[このコンピュータ上でアプリケーションを実行する]** を有効にします [1]。

[セキュリティ] タブ

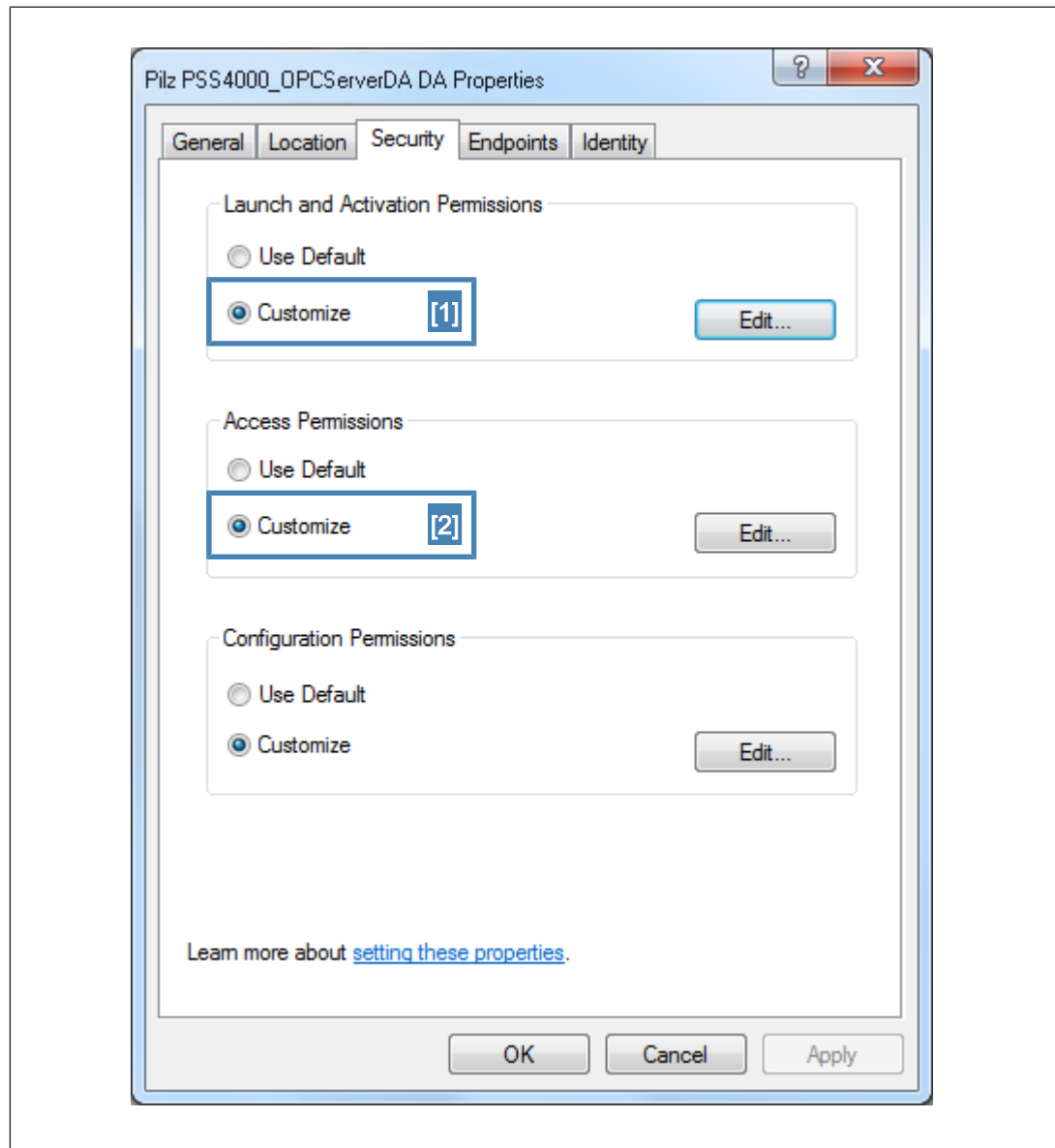


図: [セキュリティ] タブ (OPCサーバのDCOMコンフィグレーション)

- ▶ [起動とアクティブ化の許可][1]および[アクセス許可][2]で[カスタマイズ]を選択して、[適用]をクリックします。
- ▶ [起動とアクティブ化の許可] フィールドで [編集] をクリックします [1]。
[起動とアクティブ化の許可] ウィンドウが開きます。

[セキュリティ] -> [起動とアクティブ化の許可]ウィンドウ

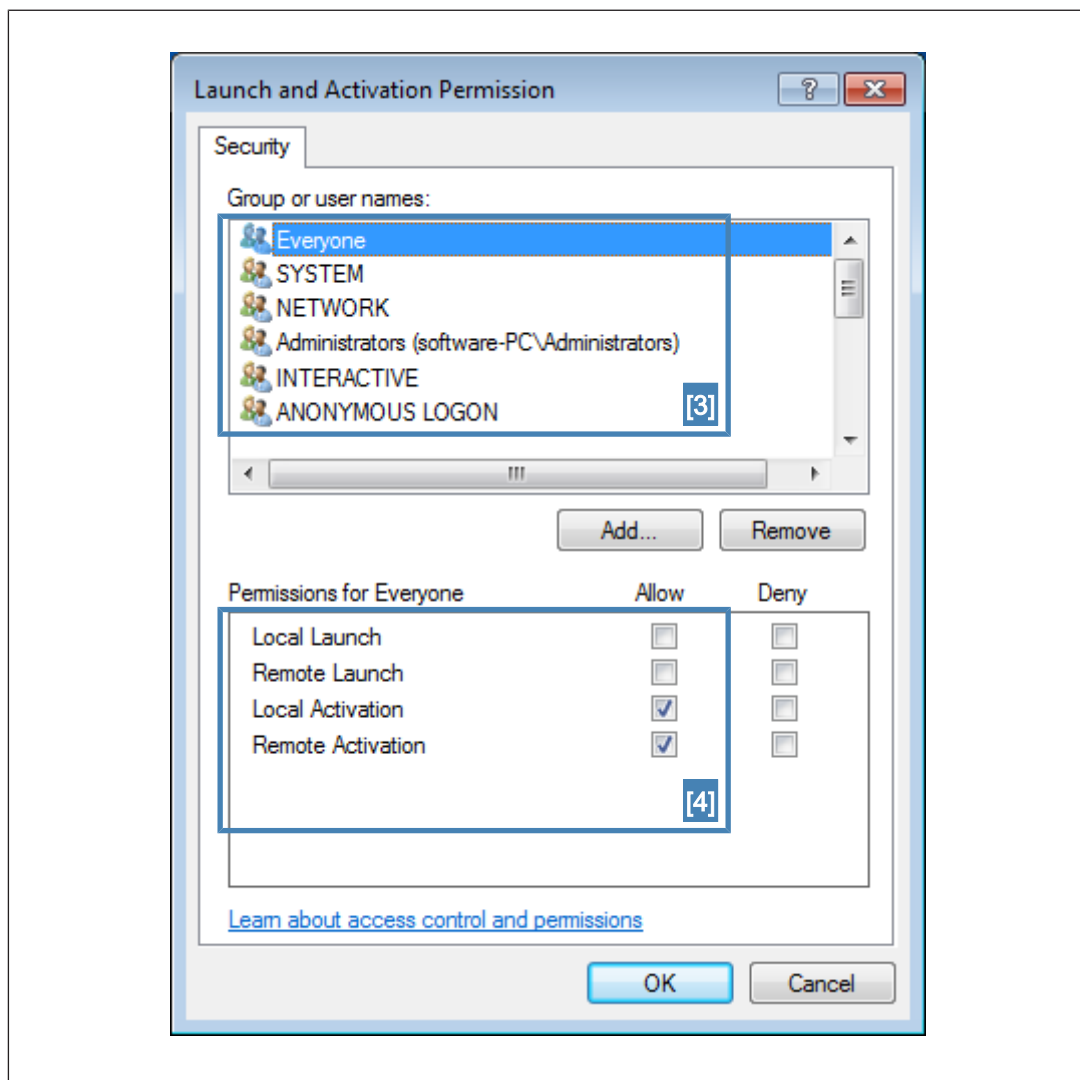


図: [起動とアクティブ化の許可] ウィンドウ

- ▶ 以下のすべてのグループ名またはユーザ名 [3] が使用可能になっているかどうかを確認します。
 - Administrators (...)
 - ANONYMOUS LOGON
 - INTERACTIVE
 - 任意
 - NETWORK
 - SYSTEM
- ▶ Administratorsユーザを使用できない場合は、このユーザを追加します。Administratorsユーザを追加するには、検索パスを調整する必要があります (「ユーザ「Administrators」の特別な機能 [528]」を参照)。
- ▶ 他のグループ名またはユーザ名のいずれかが使用可能になっていない場合は、その名前も追加します ([追加...] - [詳細設定] - [今すぐ検索])。

NETWORKユーザには、追加情報があることに注意してください (「ユーザ「NETWORK」の特別な機能」[\[529\]](#)を参照)。

[OK]で選択を確定します。

[起動とアクティブ化の許可]ウィンドウが再び有効になります。

- ▶ 各ユーザを個別に選択し [3]、[許可]をオンにして許可します[4]。
 - ローカルからのアクティブ化
 - リモートからのアクティブ化

- ▶ [OK]をクリックします。

[セキュリティ]タブが再び有効になります。

- ▶ [セキュリティ]タブの [アクセス許可] [2] で [編集]をクリックします。
[アクセス許可]ウィンドウが開きます。

[セキュリティ] -> [アクセス許可] ウィンドウ

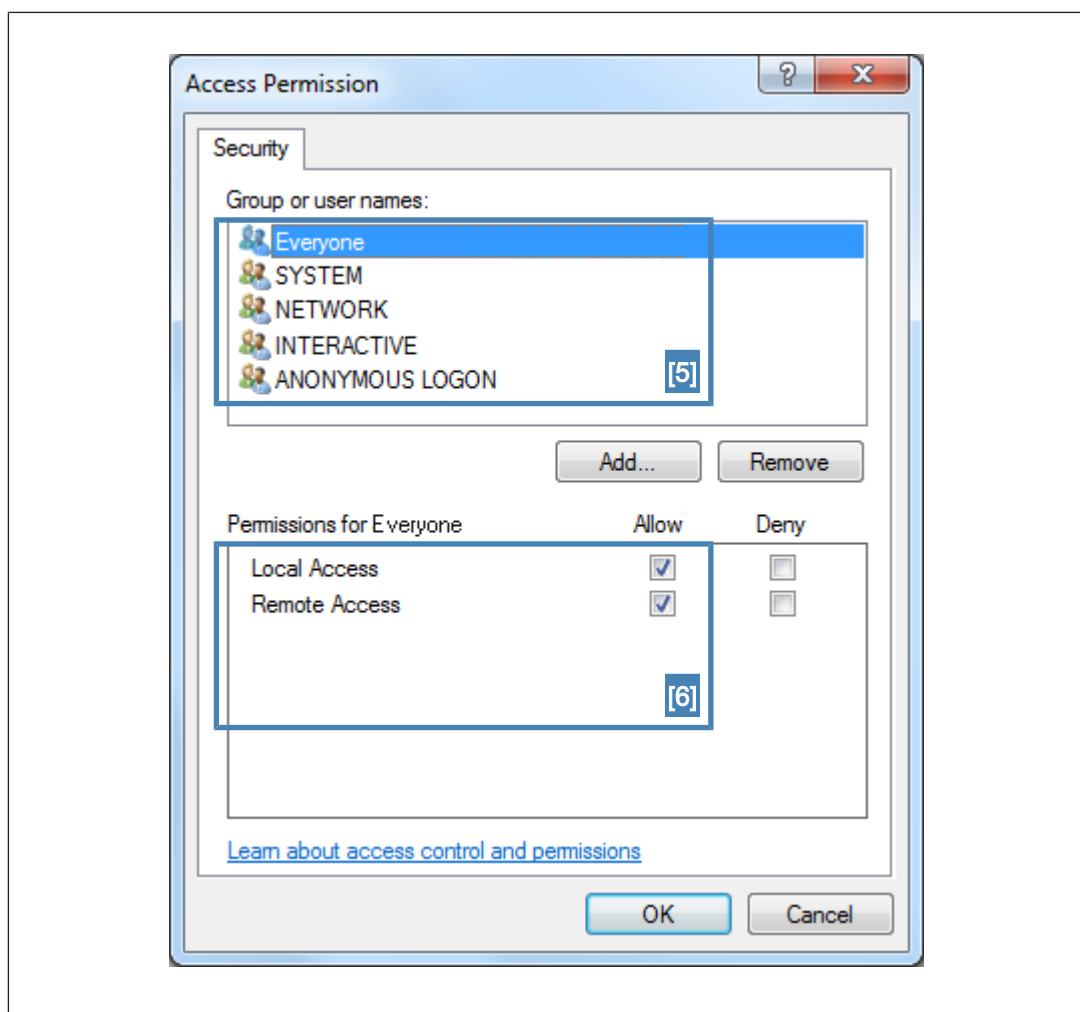


図: [アクセス許可] ウィンドウ

- ▶ 以下のすべてのグループ名またはユーザ名 [5] が使用可能になっているかどうかを確認します。
 - ANONYMOUS LOGON

- INTERACTIVE

- 任意

- NETWORK

- SYSTEM

- ▶ グループ名またはユーザ名のいずれかが使用可能になっていない場合、その名前を追加します (**[追加...]** - **[詳細設定]** - **[今すぐ検索]**)。

NETWORKユーザには、追加情報があることに注意してください (「NETWORKユーザの特別な機能」を参照)。

[OK]で選択を確定します。

[アクセス許可] ウィンドウが再び有効になります。

- ▶ 各ユーザを個別に選択し [5]、**[許可]** をオンにしてアクセスを許可します [6]。

- ローカルアクセス

- リモートアクセス

- ▶ **[OK]** をクリックします。

[安全] タブが再び有効になります。

[ID] タブ

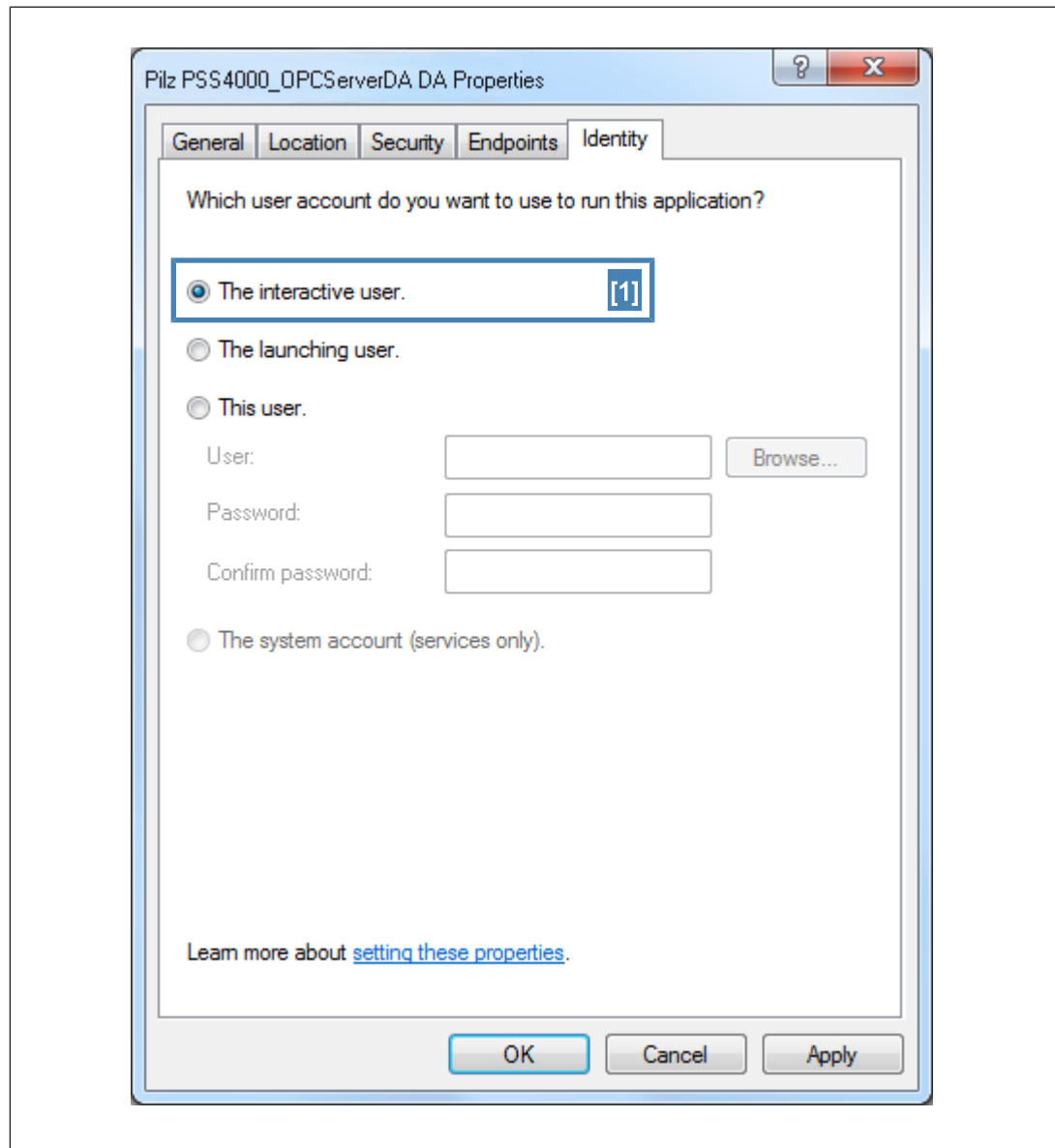


図: [ID] タブ (OPCサーバのDCOMコンフィグレーション)

- ▶ **[対話ユーザ]** を選択します [1]。

[エンドポイント] タブ

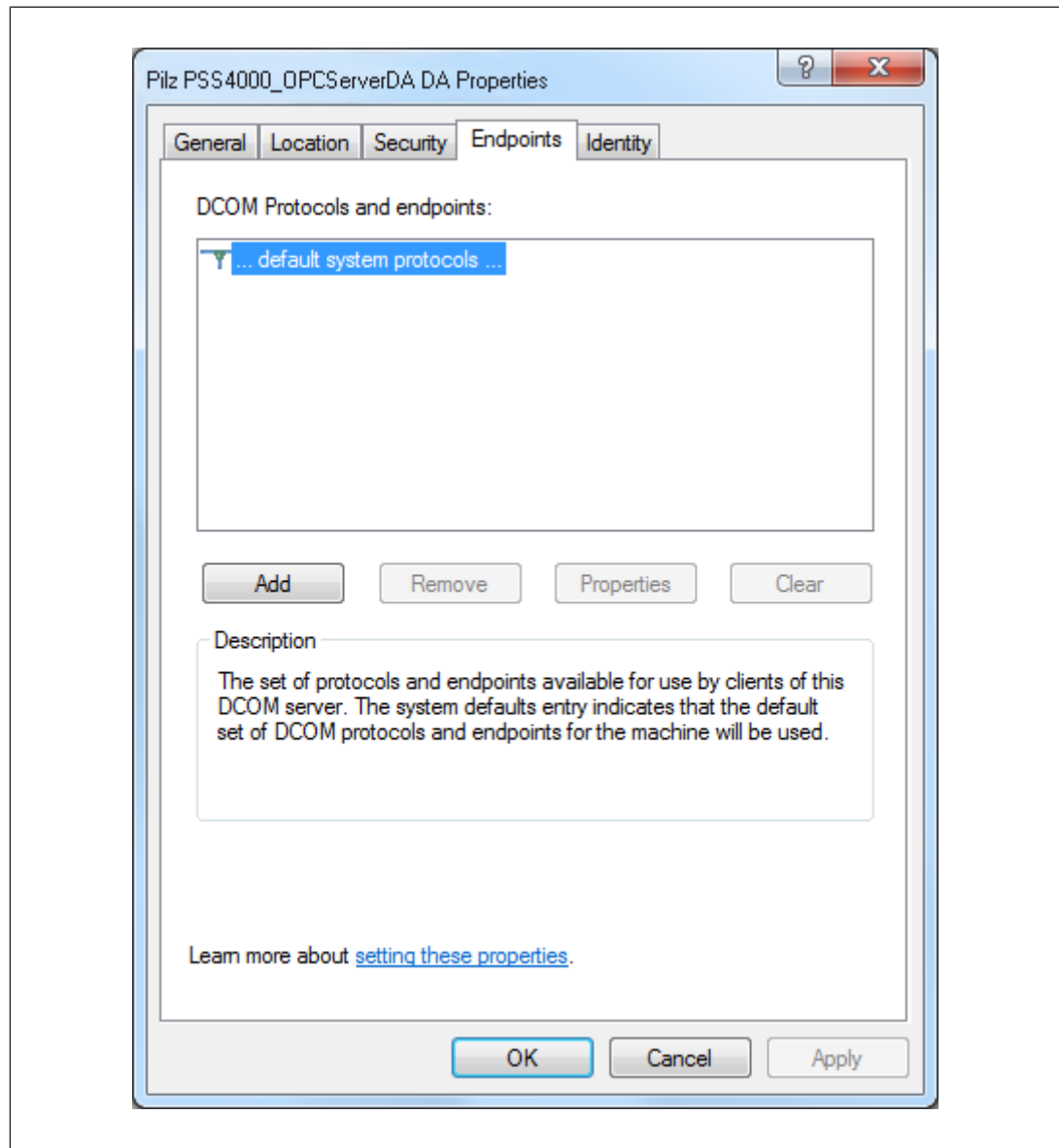


図: [エンドポイント] タブ (OPCサーバの接続プロトコル)

- ▶ **[既定のシステムプロトコル]** がリストに表示されない場合、以下の手順でこの接続プロトコルを追加します。
 - **[追加]** をクリックし、**[プロトコルシーケンス]** で **[接続指向TCP/IP]** プロトコルを選択します。
 - **[OK]** をクリックします。
 - **[エンドポイント]** タブが再び有効になります。
 - ▶ **[OK]** をクリックしてすべての設定を保存し、**[PVIS OPC Server DAのプロパティ]** ウィンドウを閉じます。
- これでOPCサーバのDCOMコンフィグレーションは完了です。

14.5.1.4 OPCサーバのOPCEnumコンフィグレーション

次の設定が該当するのは、OPCサーバを搭載したPCのみです。

- ▶ **[コンポーネントサービス]** ウィンドウを開きます。

(「[DCOMコンフィグレーションの開始](#)」[\[501\]](#)」を参照してください。)

- ▶ **[コンポーネントサービス]** ウィンドウで**DCOM Config**ディレクトリ [\[1\]](#) を開きます。

DCOMコンフィグレーションに関する警告が出されることがありますが、このコンフィグレーションに関しては、これらの警告をすべて無視してかまいません。毎回、**[いいえ]**をクリックします。

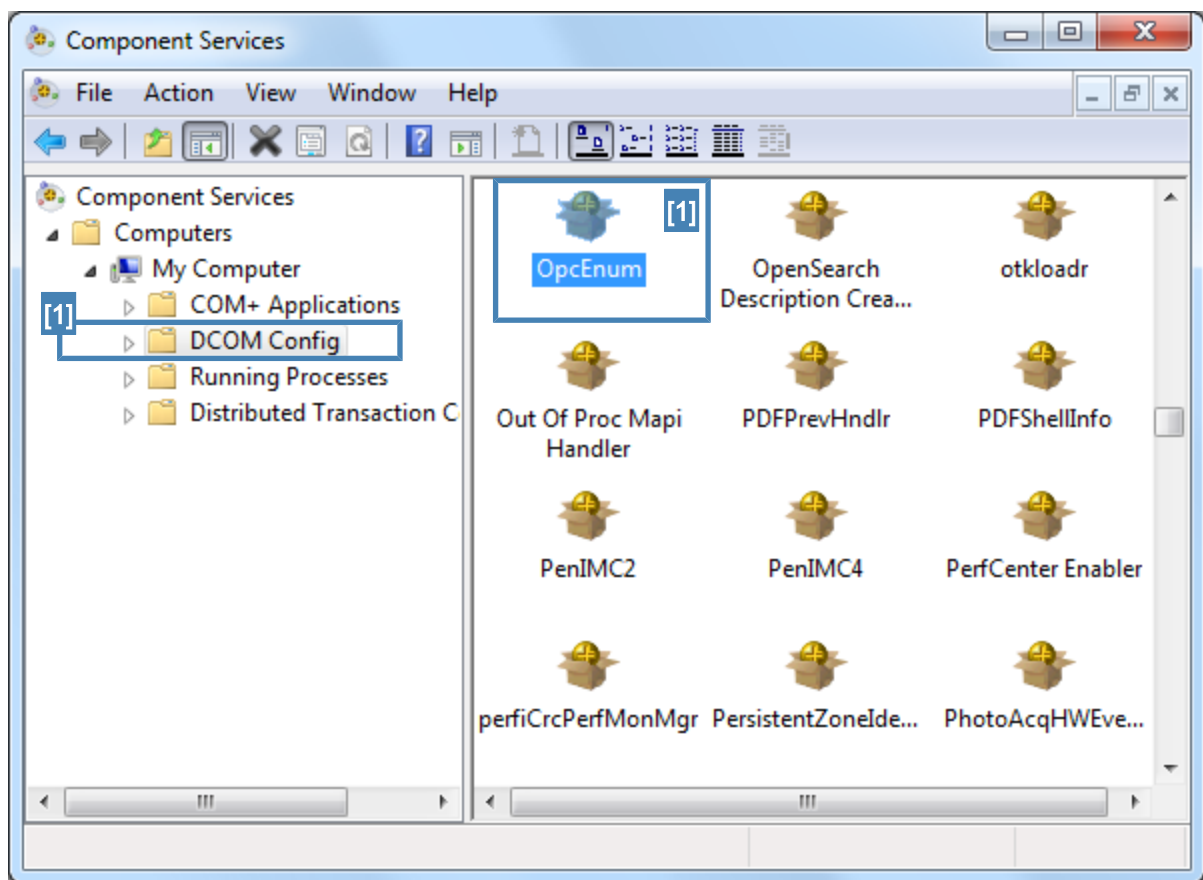


図: **[コンポーネントサービス]** ウィンドウ (OPCサーバのOPCEnumに対するDCOMコンフィグレーション)

- ▶ **[OpcEnum]** アプリケーション [\[2\]](#) を右クリックして **[プロパティ]** を選択します。
[OpcEnumのプロパティ] ウィンドウが開きます。

[一般事項] タブ

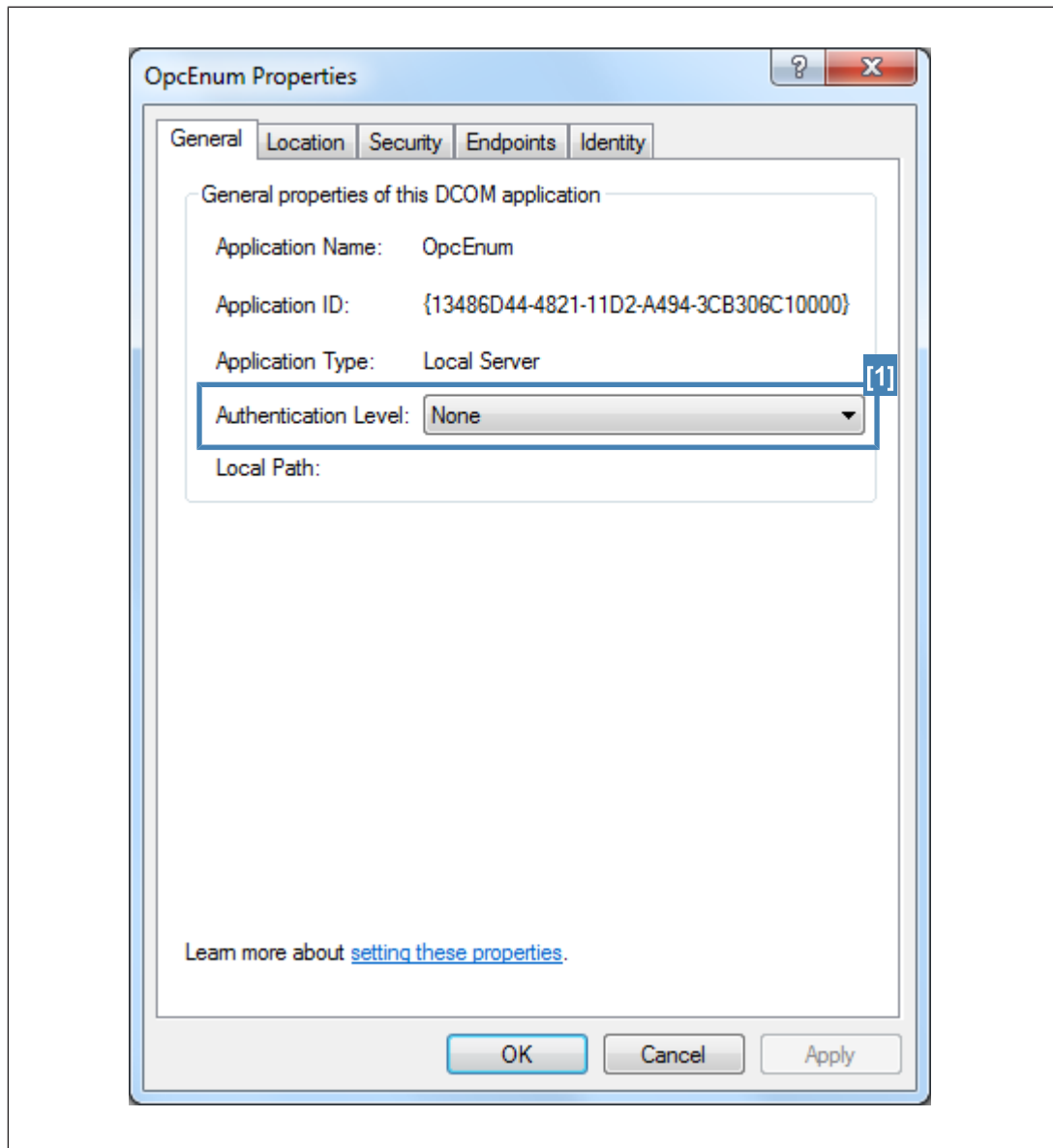


図: [全般] タブ (OPCサーバのOPCEnumに対するDCOMコンフィグレーション)

- ▶ [認証レベル] で [なし] を選択します [1]。

[ロケーション] タブ

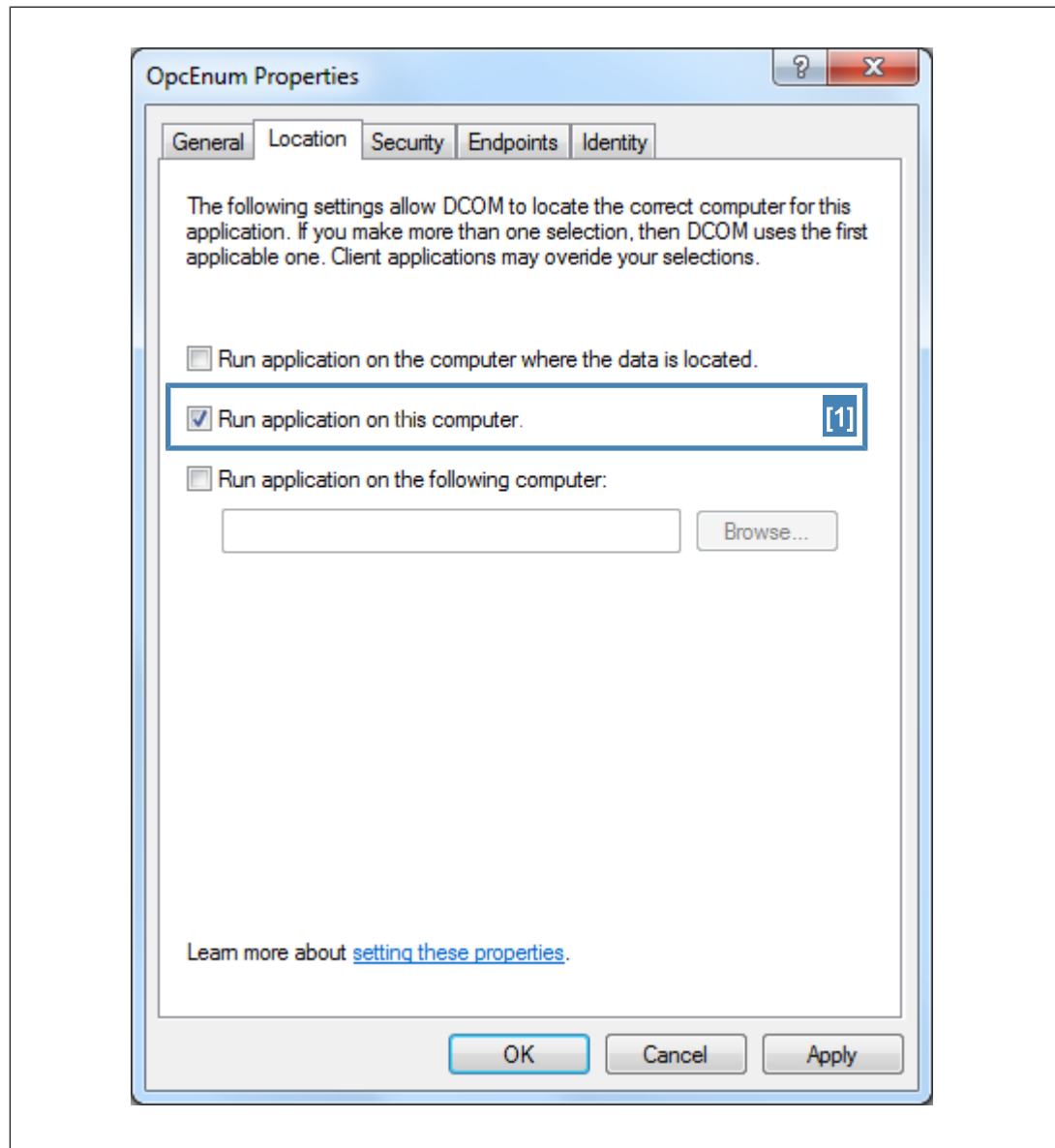


図: [ロケーション] タブ (OPCサーバのOPCEnumに対するDCOMコンフィグレーション)

- ▶ **[このコンピュータ上でアプリケーションを実行する]** を有効にします [1]。

[セキュリティ] タブ

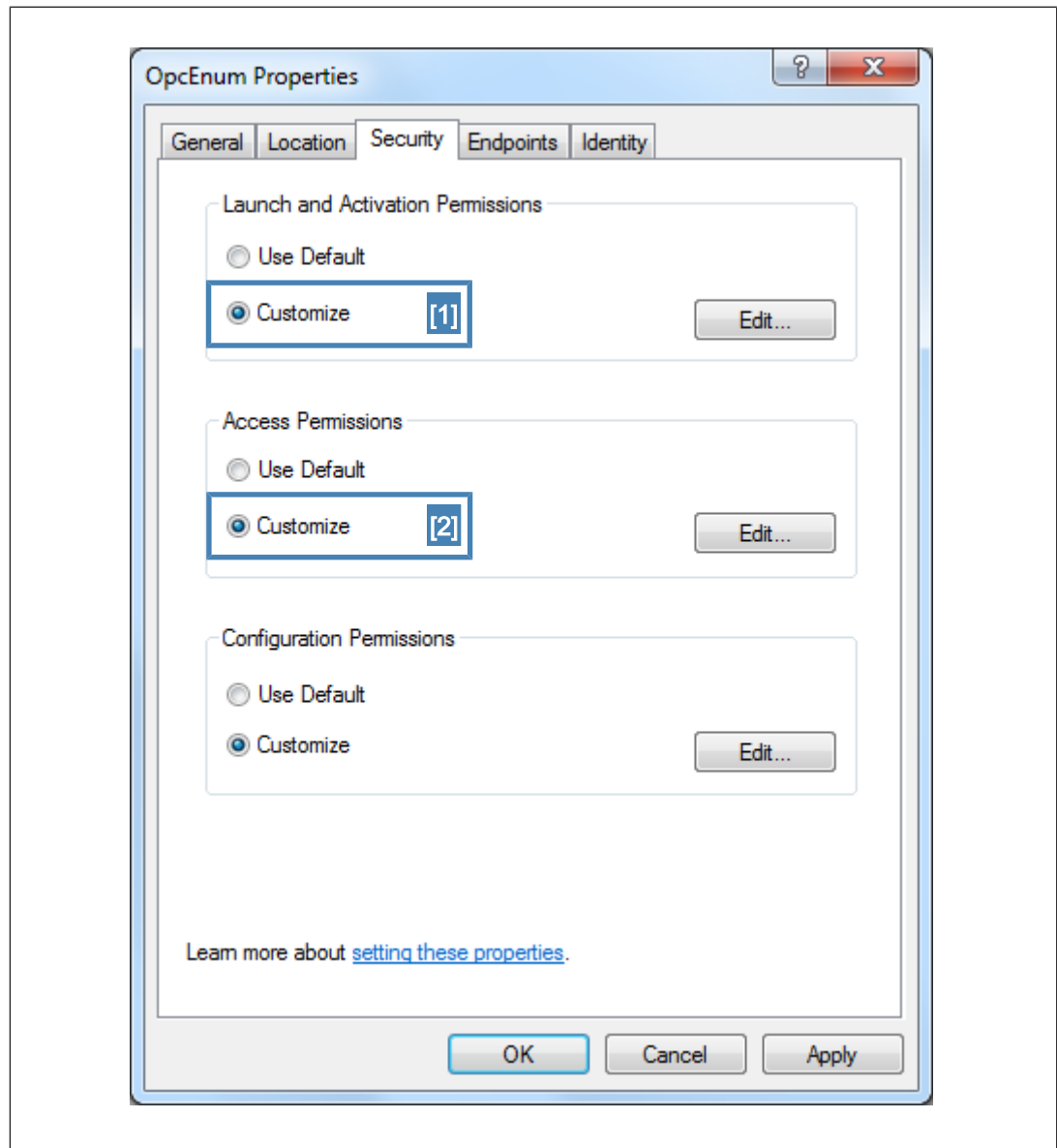


図: [セキュリティ] タブ (OPCサーバのOPCEnumに対するDCOMコンフィグレーション)

- ▶ **[起動とアクティブ化の許可]** [1]および**[アクセス許可]** [2]で[カスタマイズ]を選択して、**[適用]**をクリックします。
- ▶ **[起動とアクティブ化の許可]** フィールドで**[編集]**をクリックします [1]。
[起動とアクティブ化の許可]ウィンドウが開きます。

[セキュリティ]ウィンドウ -> [起動とアクティブ化の許可]

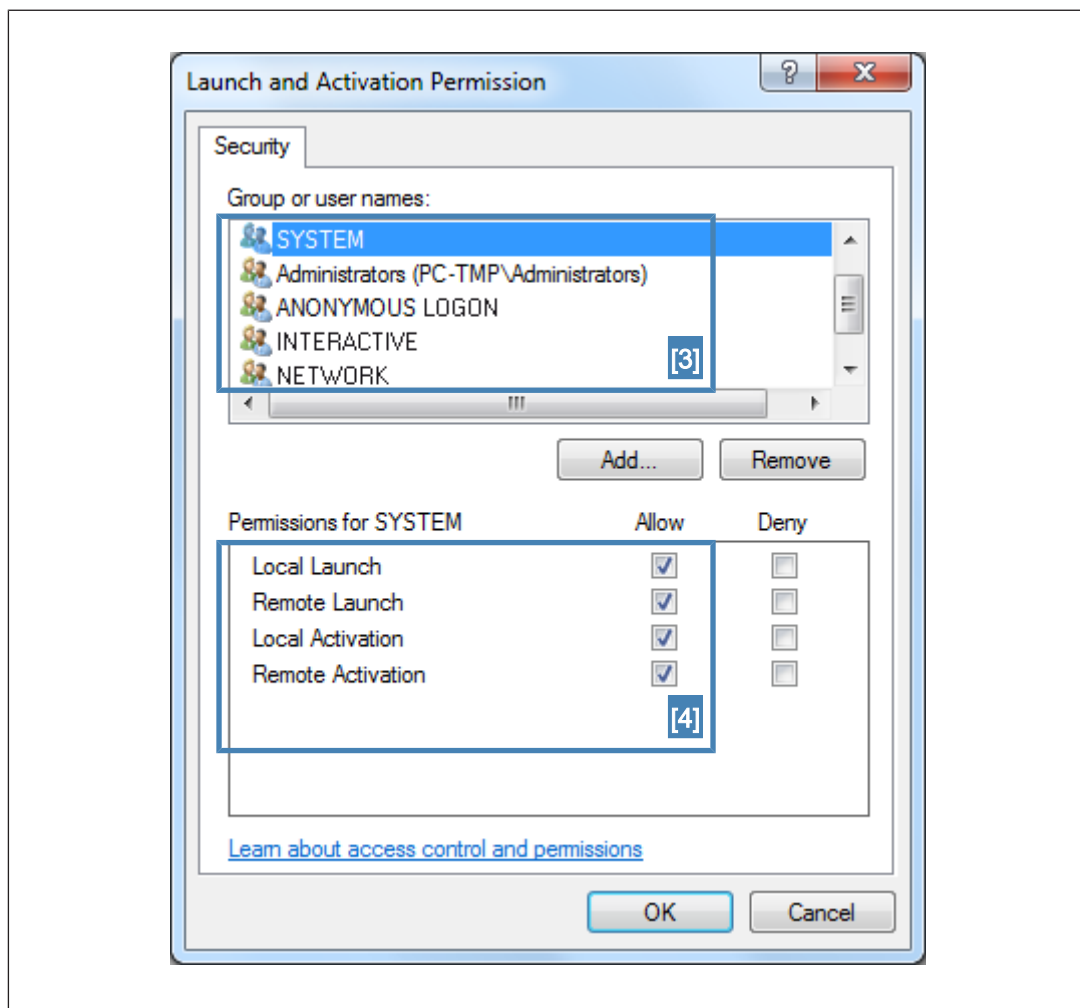


図: [起動とアクティブ化の許可] ウィンドウ(OPCEnumのCOMセキュリティ)

▶ 以下のすべてのグループ名またはユーザ名 [3] が使用可能になっているかどうかを確認します。

- Administrators (...)
- ANONYMOUS LOGIN
- INTERACTIVE
- 任意
- NETWORK
- SYSTEM

▶ Administratorsユーザを使用できない場合は、このユーザを追加します。Administratorsユーザを追加するには、検索パスを調整する必要があります (「ユーザAdministratorsの特別な機能」を参照)。

▶ 他のグループ名またはユーザ名のいずれかが使用可能になっていない場合は、その名前を追加できます ([追加...] - [詳細設定] - [今すぐ検索])。

NETWORKユーザには、追加情報があることに注意してください (「NETWORKユーザの特別な機能」を参照)。

[OK] で選択を確定します。

[起動とアクティブ化の許可] ウィンドウが再び有効になります。

- ▶ 各ユーザ [3] を個別に選択し、**[許可]** をオンにしてアクセスを許可します [4]。

- ローカルからの起動
- リモートからの起動
- ローカルからのアクティブ化
- リモートからのアクティブ化

- ▶ **[OK]** をクリックします。

[セキュリティ] タブが再び有効になります。

- ▶ **[セキュリティ]** タブの **[アクセス許可]** [2] で、**[編集]** をクリックします。

[アクセス許可] ウィンドウが開きます。

[セキュリティ] ウィンドウ -> [アクセス許可]

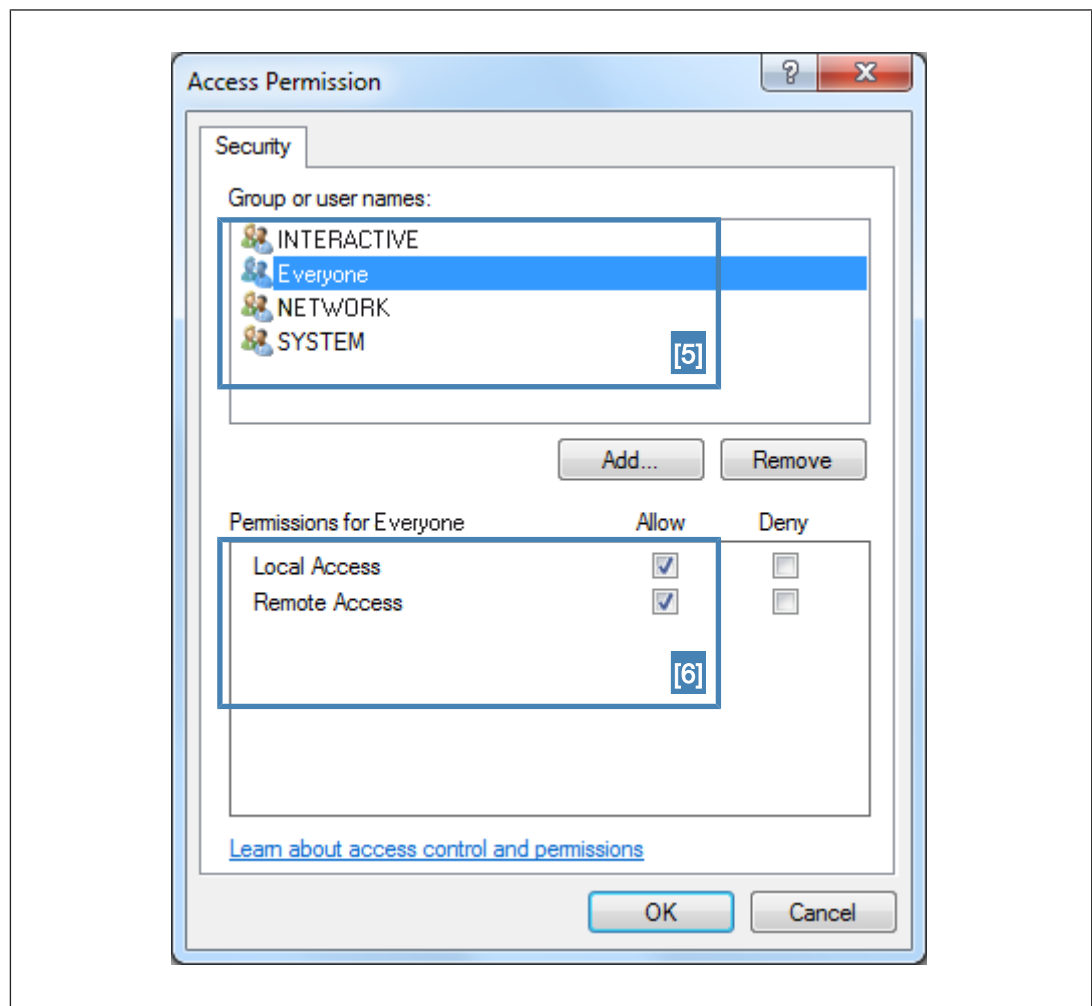


図: **[アクセス許可]** ウィンドウ (OPCEnumのCOMセキュリティ)

- ▶ 以下のすべてのグループ名またはユーザ名 [5] が使用可能になっているかどうかを確認します。

- INTERACTIVE
- 任意

– NETWORK

– SYSTEM

- ▶ グループ名またはユーザ名のいずれかが使用可能になっていない場合、その名前を追加します (**[追加...]** - **[詳細設定]** - **[今すぐ検索]**)。

NETWORKユーザには、追加情報があることに注意してください (「NETWORKユーザの特別な機能」を参照)。

[OK] で選択を確定します。

[アクセス許可] ウィンドウが再び有効になります。

- ▶ 各ユーザ [5] を個別に選択し、**[許可]** をオンにしてアクセスを許可します [6]。

– ローカルアクセス

– リモートアクセス

- ▶ **[OK]** をクリックします。

[セキュリティ] タブが再び有効になります。

[ID] タブ

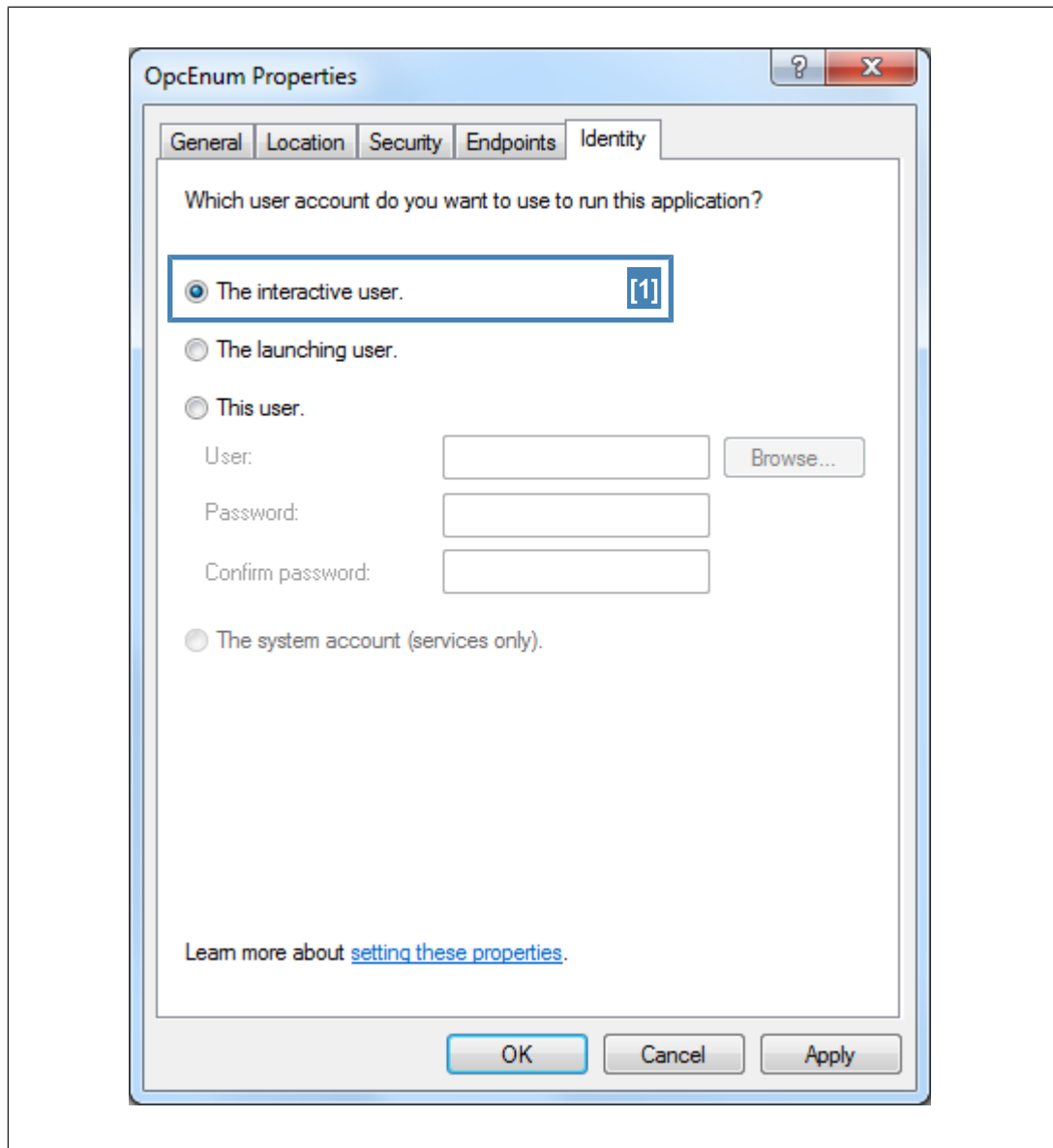


図: [ID] タブ (OPCEnumのCOMセキュリティ)

- ▶ **[対話ユーザ]** を選択します [1]。

[エンドポイント] タブ

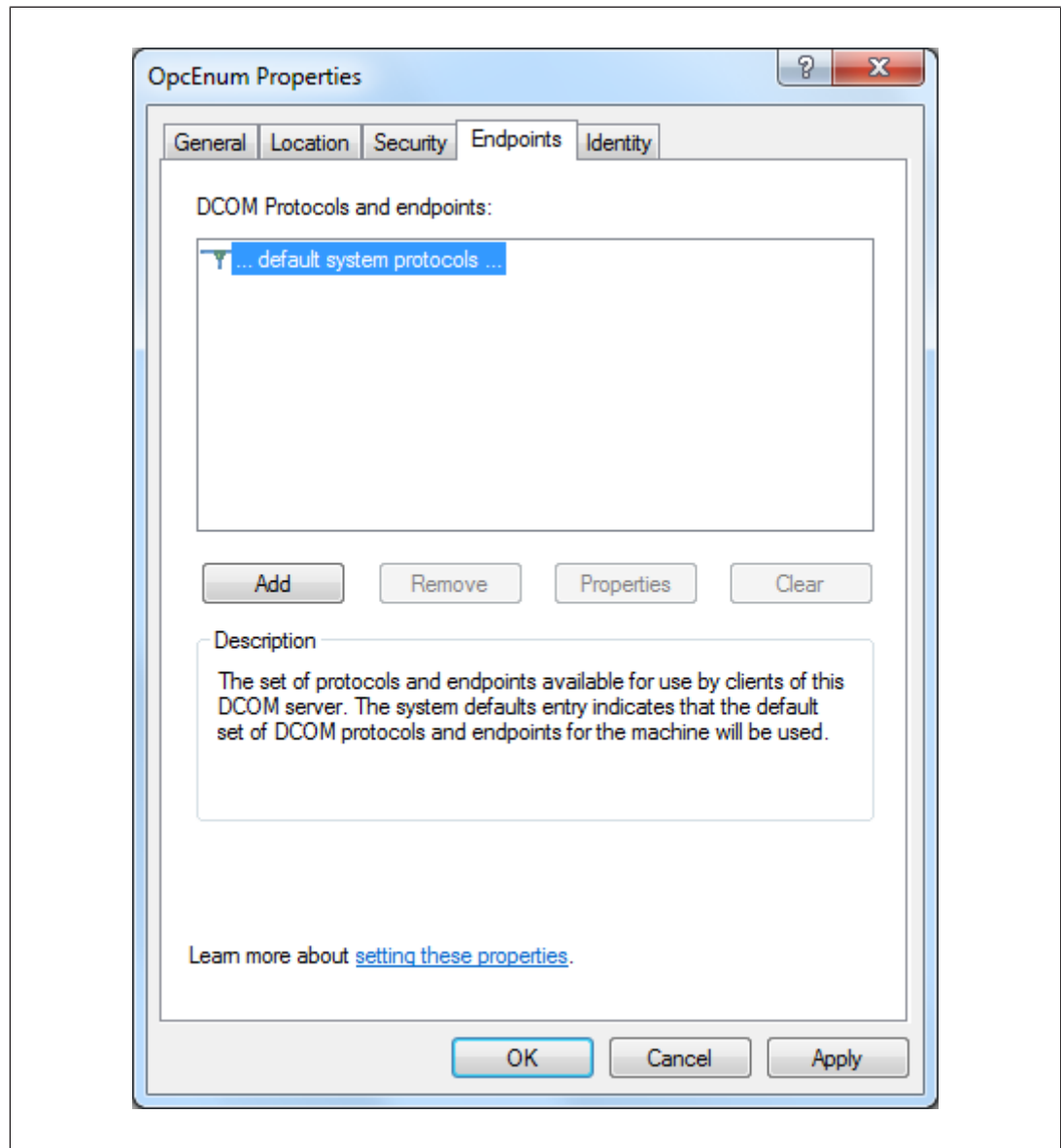


図: [エンドポイント] タブ (OPCEnumの接続プロトコル)

- ▶ **[既定のシステムプロトコル]** がリストに表示されない場合、以下の手順でこの接続プロトコルを追加します。
 - **[追加]** をクリックし、**[プロトコルシーケンス]** で **[接続指向TCP/IP]** プロトコルを選択します。
 - **[OK]** をクリックします。
 - [エンドポイント]** タブが再び有効になります。
 - ▶ **[OK]** をクリックしてすべての設定を保存し、**[OpcEnumのプロパティ]** ウィンドウを閉じます。
これでOPCサーバのOPCEnumコンフィグレーションは完了です。
 - ▶ **[コンポーネントサービス]** ウィンドウを閉じます。
- この設定は、次回のPCの再起動時に有効になります。

14.5.1.5 ユーザ「Administrators」の特別な機能

許可するグループ名またはユーザ名にAdministratorsユーザを追加するには、最初に検索パスを調整する必要があります。

- ▶ 関連する起動アクセス許可ウィンドウで、**[追加...]**、**[パス...]**の順にクリックします。
- ▶ DCOMコンフィグレーションで (検索) パスとして設定するPC (この例では「PC-TMP」[1]) を選択し、**[OK]**で確定します。
- ▶ オブジェクト名として「Administrators」[2]を入力し、**[OK]**をクリックします。

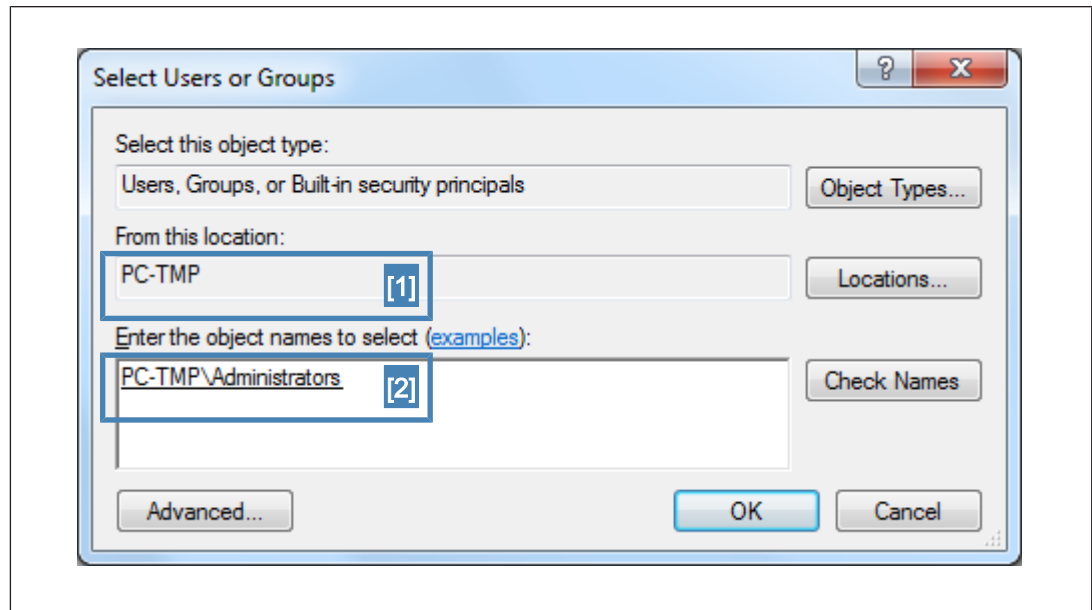


図: [ユーザまたはグループの選択] ウィンドウ

14.5.1.6 ユーザ「NETWORK」の特別な機能

許可ウィンドウ ([起動許可]ウィンドウ、[アクセス許可]ウィンドウなど) でNETWORKユーザを追加する場合、ユーザ名「Network」を入力する際に、選択肢として複数のユーザが表示されることがあります。この場合は、「Network」ユーザを選択します。

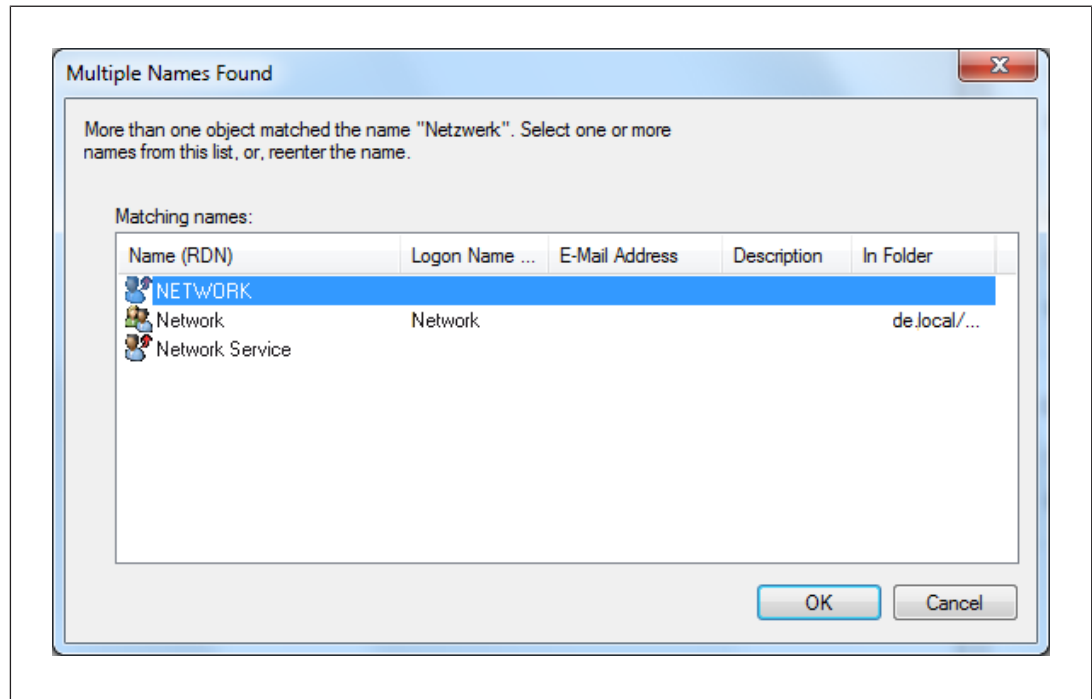


図: NETWORKユーザの選択

14.5.2 ファイアウォールのコンフィグレーション

PCおよびネットワークのファイアウォールや、その他のセキュリティメカニズムにより、OPCサーバが接続された装置と正常に通信できないことがあります。このような場合は、セキュリティ設定とファイアウォール設定を調整する必要があります。必要なセキュリティ設定に関する最新の資料は、OPC FoundationのWebサイト (www.opcfoundation.org) にアクセスして、[Downloads] の [White Papers] で入手できます。

ファイアウォールでは、明示的に登録されたプログラムへのアクセスが許可されます。次のプログラムにはアクセス権が必要です。

- ▶ すべてのOPCクライアント
- ▶ Microsoft管理コンソール
- ▶ Pilz PVIS OPC Server DA.exe (PVIS OPCサーバのみ)
- または
- ▶ PSS4000_OPCTServerDA.exe (OPCサーバ (クラシック) システムセクションがあるPSS 4000のみ)
- ▶ OPCenum.exe

またデータ交換のためにポートを有効にする必要があります。

- ▶ 名前: DCOM
- ▶ ポート番号135
- ▶ プロトコル: TCP

これらの有効化により、ネットワーク上またはインターネット上の他のユーザに対して、PCへのアクセス権が付与されます。ネットワーク管理者は、これらのファイアウォール設定を最適化して、不正アクセスからサーバを保護することができます。

14.6 OPCサーバの試運転

OPCサーバの試運転を行うには、次の手順に従います。

1. PAS4000でOPCサーバをコンフィグレーションします (PAS4000オンラインヘルプを参照)。
2. プロジェクトのダウンロードを実行します。
プロジェクトのダウンロード時に、OPCサーバの装置プロジェクトが作成され、OPCサーバに格納されます。OPCサーバの装置プロジェクトには、OPCコンフィグレーションが含まれています。このコンフィグレーションは、プロジェクト上でコンフィグレーションされる全装置のコンフィグレーションデータおよびネームスペースで構成されています。

プロジェクトのダウンロード後、OPCサーバは、プロジェクト上でコンフィグレーションされる全装置のネームスペースをOPCクライアントでできるようにします。

14.7 データ型の割り付け

OPC規格により、さまざまなデータ型が割り付けられます。以下のテーブルに、OPCサーバでサポートされるデータ型、PSS4000のユーザプログラムで使用できる類似のデータ型を示します。

PSS 4000ネームスペース	OPCクラシック	ユーザプログラム (PSS 4000)
BOOL値	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

14.8 PSS 4000ネームスペース内のデータへのアクセス

PSS 4000 OPCサーバは、PSS 4000プロジェクトから一般的なプロジェクトデータおよび装置ベースのプロジェクトデータを収集し、PSS 4000ネームスペースに提示します。PSS 4000ネームスペースは、定義済み構造の特別なメモリ領域です。OPCクライアントは、この定義済み構造を介してPSS 4000ネームスペースのデータにアクセスできます。

PSS 4000ネームスペースとその構造についての詳細は [PSS 4000ネームスペース](#)  586] を参照してください。

14.9 ネットワークデータ (OPCサーバ (クラシック))

プロトコル	方向 [*]	転送プロトコル	ポート番号	無効化機能あり	瞬時の処理が必要	説明
SNTP	入力	UDP (IPv4)	123	はい デフォルト: 無効	いいえ	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載した PMI/PC アプリケーション: ▶ 時刻同期 ▶ SNTPサーバ
SNTP	出力	UDP (IPv4)	123	はい デフォルト: 無効	いいえ	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載した PMI/PC アプリケーション: ▶ 時刻同期 ▶ SNTPクライアント
SNp	入力/ 出力	UDP (IPv4)	40000	いいえ	はい	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載した PMI/PC アプリケーション: ▶ 読み取りおよび取得用の SafetyNET p経由の通信 – プロセスデータ – ステータス情報 – イベントメッセージ

プロトコル	方向 [※]	転送プロトコル	ポート番号	無効化機能あり	瞬時の処理が必要	説明
SNp MSC	入力	UDP (IPv4)	40000	いいえ	いいえ	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載した PMI/PC アプリケーション: ▶ 以下のためのIPマルチキャストとの SafetyNET p通信 – ネットワークのスキャン – 装置の命名 – 装置の識別 実装されているセキュリティ対策: ▶ マルチキャストアドレスは次の値に制限: 224.0.23.61
SNp MSC	出力	UDP (IPv4)	40000	いいえ	いいえ	影響を受ける装置: ▶ PC用PSS 4000ファームウェア搭載PC アプリケーション: ▶ 以下のためのIPマルチキャストとの SafetyNET p通信 – ネットワークのスキャン – 装置の命名 – 装置の識別 実装されているセキュリティ対策: ▶ マルチキャストアドレスは次の値に制限: 224.0.23.61

[※]

入力: 通信相手が装置との通信を開始します。

出力: 装置が通信相手との通信を開始します。

「ファイアウォールのコンフィグレーション」 530」の項も参照してください。

15 OPCサーバUA

15.1 概要

OPCは、オープン性 (Openness)、生産性 (Productivity)、協調作業 (Collaboration) を表し、UAは統合アーキテクチャ (Unified Architecture) を表します。OPC UAは、プラットフォーム非依存、拡張性、高可用性、インターネット機能などの多くの特性を持つ現在クラシックOPCと呼ばれることが多いOPC規格を補完するものです。

OPC UAは、標準化されたソフトウェアインタフェースを提供し、幅広い製造業者のアプリケーションでデータの交換を可能にします。このOPCによる基盤は、規格の維持と展開に大きな役割を果たします。

ネットワーク化された設備において、OPCサーバは通常、診断データやプロセスデータなどを収集し、定義されたネームスペースでこれらを表示します。OPCクライアントは、一般的にプロセスの視覚化やオペレーティングデータの記録に使用されます。このクライアントは、OPCサーバが提供するデータにアクセスしてプロセスを実行します。データを処理するために、OPCクライアントには、関連するOPC対応ソフトウェアが必要です (ピルツの視覚化ソフトウェアPASvisu、Copa DataのZenonなど)。

オートメーションシステムPSS 4000では、PCが、PSS 4000ファームウェアをインストールすることで、システムセクションPSS 4000 OPCサーバUAを取得します。PSS 4000ファームウェアをインストールすると、PCは「サードパーティ製造業者のPSS 4000対応装置」になります (「[システムの概要](#)」 [20] を参照)。

PSS 4000 OPCサーバUAは、接続されたコントローラからプロセスデータ、ステータス情報、およびイベントメッセージを読み取り、PSS 4000ネームスペースで表示します。これは、OPC UAクライアントで取得できます。



重要

PSS 4000 OPCサーバUAは、OPC UAクライアント (ピルツの視覚化システムPASvisuなど) でのみ使用できます。OPCクライアント (クラシック) は、PSS 4000 OPCサーバUAにアクセスできません。

PSS 4000 OPCサーバUAには、次のような境界条件があります。

- ▶ PSS 4000 OPCサーバUAは、さまざまなOPC UAプロパティをサポートして、OPC UAクライアントからのアクセスを実現します (「[OPC UAプロパティ](#)」 [538] を参照)。
- ▶ PSS 4000 OPCサーバUAは、PCにインストールできます (「[システム要件](#)」 [543] を参照)。
- ▶ PAS4000で作成された各プロジェクトには、最大4つのOPCサーバを含めることができます。プロセスで、OPCクラシックサーバとOPC UAサーバを組み合わせることができます。
- ▶ 複数のクライアントとともにOPCサーバに同時にアクセスすることができます。次のことが該当します。
 - OPCクラシッククライアントのみがOPCクラシックサーバにアクセスできます。
 - OPC UAクライアントのみがOPC UAサーバにアクセスできます。
- ▶ PSS 4000 OPCサーバUAのプロジェクトベースの全設定はPAS4000で行われ、プロジェクトのダウンロード時に装置プロジェクトとしてPSS 4000 OPCサーバUAに転送されます。

15.1.1 設計オプション

最小限のコンフィグレーションでは、PSS 4000 OPCサーバUAは視覚化ソフトウェアを含んでいるOPC UAクライアントと同じPCに配置されます。

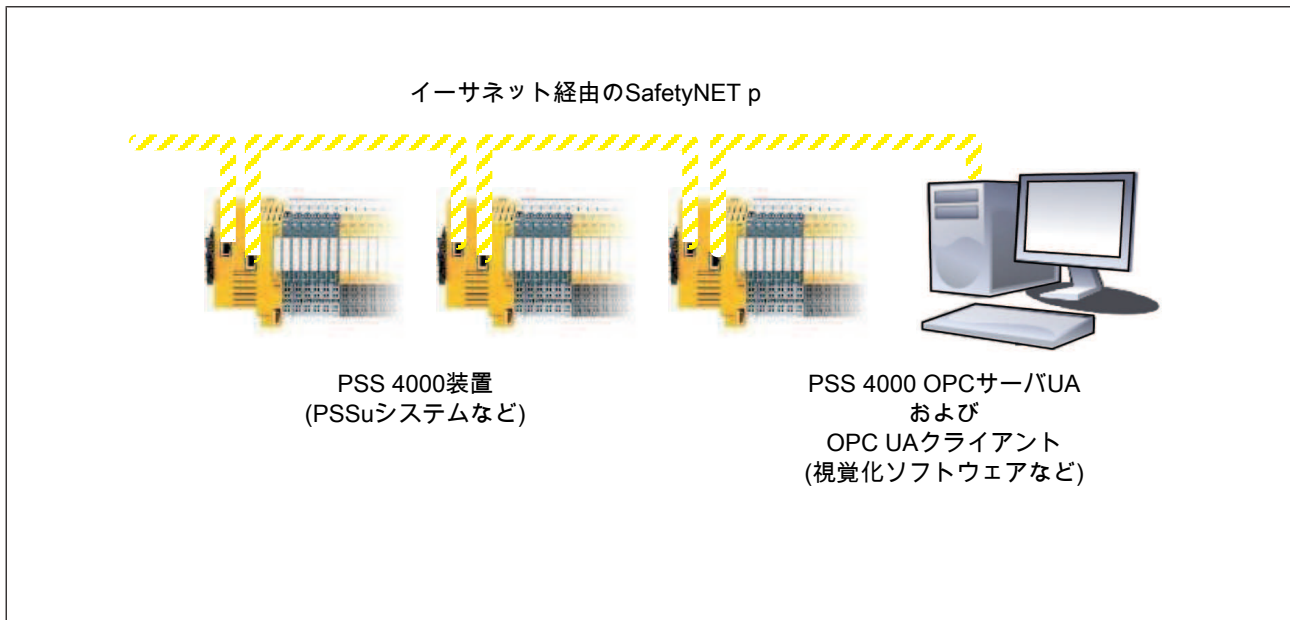


図: 最小限のOPC UAコンフィグレーションでのプロセスの視覚化 (原則)

その他のタイプの組み合わせも考えられます (以下を参照)。

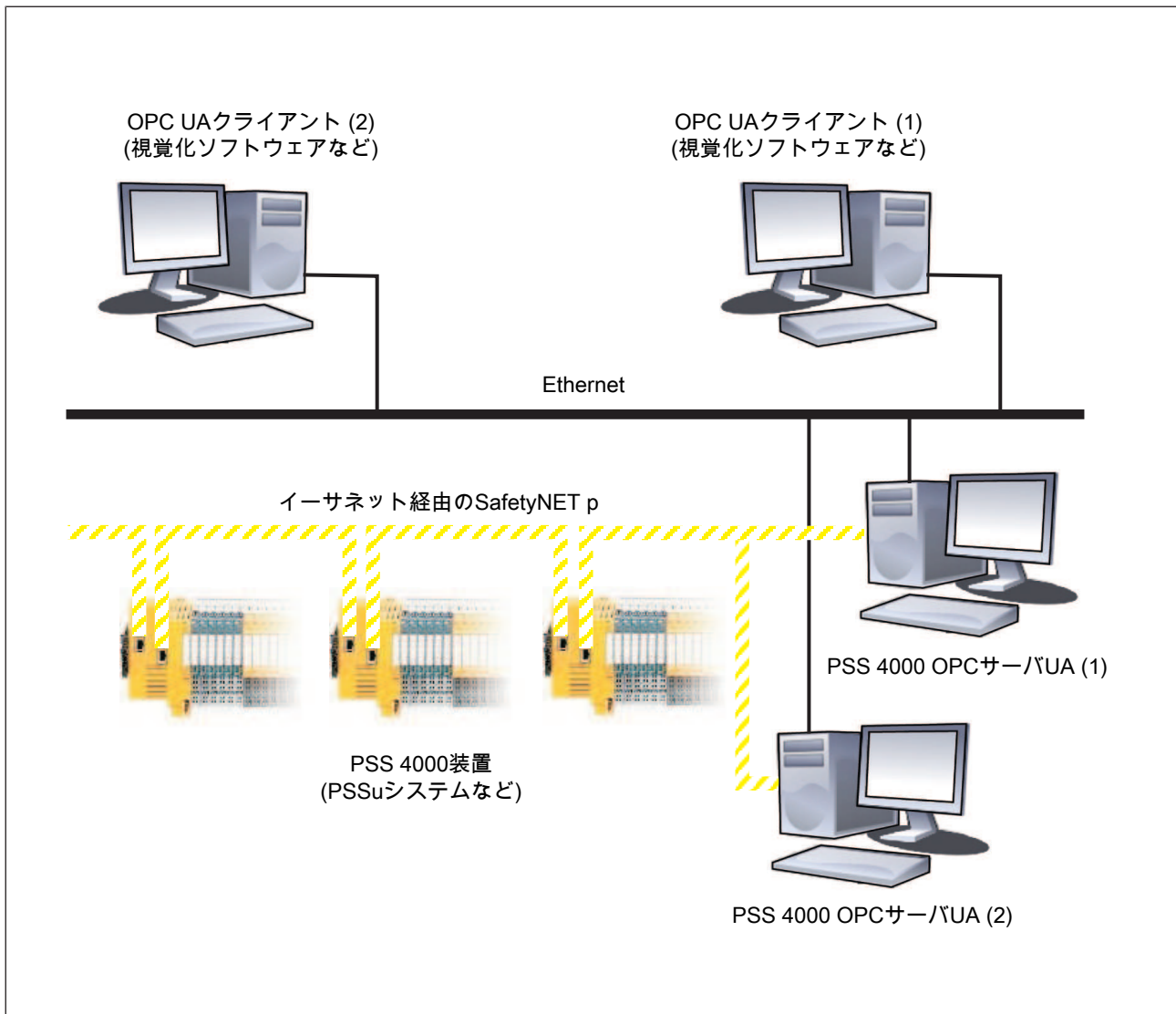


図: 冗長設計、およびイーサネット経由でのネットワークングを備えたOPC UAによるプロセスの視覚化 (原則)

15.1.2 OPC UAプロパティ

PSS 4000 OPCサーバUAは、次のOPC UAプロパティをサポートして、OPC UAクライアントからのアクセスを実現します。

▶ TCPによるUAバイナリプロトコルのサポート

▶ PSS 4000 OPCサーバUAのエンドポイントURL (ネットワーク内のアドレス):

opc.tcp: //<IPアドレス>:<ポート>/Pilz/PSS4000OPCServerUA

<IPアドレス> PAS4000の命名データでコンフィグレーションされたPSS 4000 OPCサーバUAのIPアドレス

<ポート> PAS4000でコンフィグレーションされたPSS 4000 OPCサーバUAのポート番号。

▶ OPC UAクライアントが読み取り／書き込みアクセス権 (RWアクセス) を持つ場合のデフォルト設定: 4853

▶ OPC UAクライアントが読み取り専用アクセス (ROアクセス) を持つ場合のデフォルト設定: 4854

注: PAS4000でポート番号を調整できます。詳細については、「[OPC UAクライアントのアクセス権 \[542\]](#)」を参照してください。

▶ データアクセス (DA) サポート

▶ 事前に定義された構文を使用したPSS 4000ネームスペース内のプロセスデータへの汎用アクセスのサポート

▶ PSS 4000ネームスペース内のプロセスデータへのシンボリックアクセスのサポート

▶ OPC UAサービスおよびサービスセットのサポート
(OPC Unified Architecture –パート4: サービスに準拠)

サービスセット	サポートされているサービス	意味
Discovery	▶ (Discovery Serverの機能)	Discovery Serverの機能 注: PSS 4000 OPCサーバUAがPCにインストールされる場合、Discovery Serverもインストールされます。
セキュアチャンネル	▶ OpenSecureChannel ▶ CloseSecureChannel	OPC UAクライアントとOPC UAサーバ間の接続管理
Session	▶ CreateSession ▶ ActivateSession ▶ CloseSession ▶ キャンセル	OPC UAクライアントとOPC UAサーバ間の接続管理
表示	▶ 検索 ▶ BrowseNext ▶ TranslateBrowsePathsToNodeIds	アドレススペースの情報の取得

サービスセット	サポートされているサービス	意味
Attributes	▶ 読み取り ▶ 書き込み	データとメタデータの読み取りと書き込み
MonitoredItem	▶ CreateMonitoredItems ▶ ModifyMonitoredItems ▶ SetMonitoringMode ▶ SetTriggering ▶ DeleteMonitoredItems	[1] を参照
サブスクリプション	▶ CreateSubscription ▶ ModifySubscription ▶ SetPublishingMode ▶ Publish ▶ Republish ▶ DeleteSubscription	[1] を参照

[1] サービスセット「サブスクリプション」および「MonitoredItems」

OPC UAクライアントは、OPC UAサーバから提供されるデータをサブスクライブできます。サブスクリプションの目的は、監視対象項目を使用してデータをグループ化することです。

サービスセット「サブスクリプション」および「MonitoredItem」の制限値に注意してください（「[制限値](#)」 540」を参照）。

15.1.3 制限値

指定		制限値
サービスセット「サブスクリプション」(「 OPC UAプロパティ 」  538]も参照)		
	PublishingInterval	100 ms以上
		15000 ms以下
		注: ▶ PublishingInterval < 100 ms サブスクリプションでは、OPC UAクライアントが「PublishingInterval」に100 ms未満の値を使用する場合、PSS 4000 OPCサーバUAで 사용되는「PublishingInterval」の最小値は100 msです ▶ PublishingInterval > 15000 ms サブスクリプションでは、OPC UAクライアントが「PublishingInterval」に15000 msより大きい値を使用する場合、PSS 4000 OPCサーバUAで 使用される「PublishingInterval」の最大値は15000 msです。
	リソースごとの最大サブスクリプション数	最大50
	サブスクリプションごとの監視対象項目の最大数。 つまり、サブスクリプションごとにリクエストできる変数の最大数	リソースあたり500 注: 配列では、アクセスタイプに関係なく、各要素がカウントされます (「 ネームスペース内の配列にアクセスする際の特別な機能 」  563]を参照)。
サービスセット「MonitoredItem」(「 OPC UAプロパティ 」  538]も参照)		
	SamplingInterval	0 ms以上
		15000 ms以下
		注: ▶ SamplingInterval = 0 PSS 4000 OPCサーバUAは、最も高速で妥当な更新間隔を使用します。 ▶ SamplingInterval > 0 かつ < 100 ms 監視対象項目では、OPC UAクライアントが「SamplingInterval」に0 msより大きくかつ100 ms未満の値を使用する場合、PSS 4000 OPCサーバUAで 使用される「SamplingInterval」の最小値は100 msです。 ▶ 「SamplingInterval」 > 15000 ms 監視対象項目では、OPC UAクライアントが「SamplingInterval」に15000 msより大きい値を使用する場合、PSS 4000 OPCサーバUAで 使用される「SamplingInterval」の最大値は15000 msです。 詳細については、[1]を参照してください

[1] 「SamplingInterval」に関する備考:

- ▶ PSS 4000 OPCサーバUAは、(PSS 4000装置、診断サーバなどの) 装置の各サブスクリプションの共通更新間隔を使用します。
- ▶ 更新間隔
更新間隔の最小値は100 msです。
更新間隔は、サブスクリプションのPublishingIntervalの最小値を2で割った値と、監視対象の項目のSamplingIntervalとQueueSizeから得られる更新間隔の値から計算されます。
 - 監視対象項目のSamplingIntervalが0より大きい:
更新間隔 = $\text{SamplingInterval} / 2$
 - 監視対象項目のSamplingIntervalが0:
更新間隔 = $(\text{サブスクリプション} / \text{キューサイズのPublishingInterval}) / 2$
 - 監視対象項目のSamplingIntervalが1:
更新間隔 = サブスクリプションのPublishingInterval / 2
- ▶ SamplingInterval = 0およびQueueSize = 1で監視対象項目が作成された場合、PSS 4000装置で変数に変更された少し後に、変更された変数がOPC UAクライアントで使用可能になります。

15.1.4 OPC UAクライアントのアクセス権

OPC UAクライアントは、PSS 4000 OPCサーバUAへの読み取り／書き込みアクセス (RWアクセス) または読み取り専用アクセス (ROアクセス) のいずれかを持つことができます。アクセス権は、使用されるポートを介して制御されます。

- ▶ OPC UAクライアントが読み取り／書き込みアクセス権を持つ場合のデフォルト設定 (RWクライアント): 4853
 - ▶ OPC UAクライアントが読み取り専用アクセス権を持つ場合のデフォルト設定 (ROクライアント): 4854
- ポート番号はPAS4000で調整できます。



重要

ROクライアントの書き込みアクセスを禁止する

OPC UAサーバへのROアクセスを持つOPC UAクライアント (ROクライアント) は、RWアクセス用のポートを使用できません。

ファイアウォールを使用して、OPC UAクライアントからOPC UAサーバへの不要な書き込みアクセスを防止してください。読み取り／書き込みエンドポイントへのROクライアントからのアクセスが防止されるようにファイアウォールを設定します。つまり、PAS4000でRWアクセス用にコンフィグレーションされたポートを経由してROクライアントとOPC UAサーバ間の通信が行われることがないようにします。

15.1.5 エンドポイントのセキュリティ設定

次の暗号化レベル (セキュリティポリシーおよびメッセージセキュリティモード) がサポートされています。

- ▶ Basic 128Rsa15 – Sign
- ▶ Basic 128Rsa15 – SignAndEncrypt
- ▶ Basic256 – Sign
- ▶ Basic256 – SignAndEncrypt

15.2 システム要件

PSS 4000ファームウェアを装置にインストールできるためには、さまざまなシステム要件を満たす必要があります。

15.2.1 Microsoft Windowsオペレーティングシステム搭載PC

PSS 4000ファームウェアは、Microsoft Windowsオペレーティングシステムを搭載したPCにインストールできます。

▶ 以下のいずれかのバージョンのMicrosoft Windows 10

- Windows 10 Home (64ビット)
- Windows 10 Pro (64ビット)
- Windows 10 Enterprise (64ビット)

15.3 **ライセンス登録**

OPCサーバのライセンスは、PAS4000でプロジェクトのライセンスが登録されると有効になります。ライセンスの詳細については、PAS4000のオンラインヘルプを参照してください。

15.4 インストール

最新のインストールファイルは、ピルツホームページのダウンロードエリア (www.pilz.comのダウンロードページ) から入手できます。たとえば「ソフトウェア」を検索語句として入力できます。

ソフトウェアのダウンロードは、登録ユーザのみが実行できます。未登録ユーザの方は、無料で登録ができます。登録には有効なEメールアドレスが必要です。

PSS 4000ファームウェアのためのインストールキットは、次のアプリケーションエリアごとに用意されています。

- ▶ PSS 4000ファームウェアがPCにインストールされます。

必要なインストールセット:

PSS_4000_PC_<バージョン番号>_<ID>.zip

(e.g. PSS_4000_PC_1_13_0_3000763B29.zipなど)

15.4.1 PCへのPSS 4000ファームウェアのインストール

PSS 4000ファームウェアは、診断ユニットとして機能するPCにインストールされます。診断ユニットは、同時に表示ユニットとしても使用できます。PSS 4000ファームウェアをインストールするときには、PSS 4000 OPCサーバ (クラシック) およびPSS 4000 OPCサーバUAの両方がインストールされます。

PSS 4000 OPCサーバ (クラシック) のOPCコアコンポーネントも自動的にインストールされます。OPCコアコンポーネントは、OPCサーバ向けのさまざまな補助的プログラムとソフトウェアコンポーネントで構成されています。PSS4000ファームウェアコントローラもインストールされます。

PSS 4000 OPCサーバUAの場合、公開キーを含む自己署名証明書および関連付けられたプライベートキーが、インストール時に作成されます。



情報

- OPCサーバをPAS4000と同じPCにインストールする場合、PAS4000ファームウェアのバージョンをPSS 4000ファームウェアのバージョンと同じにする必要があります。
- OPCサーバをPAS4000と同じPCにインストールする場合、PAS4000を最初にインストールし、次にPSS 4000ファームウェアをインストールすることをお勧めします。
- OPCサーバをPAS4000と同じPCにインストールする場合、PSS 4000ファームウェアはインストール中にシステム変数PILZ_SNP_MULTICAST_TTLを64に設定します。
- OPCサーバをPAS4000と同じPCにインストールする場合、PAS4000がOPCサーバ上のプロジェクトに影響する可能性があります。PAS4000のオンラインアクションが別のプロジェクトで実行された場合、OPCサーバ上のプロジェクトが使用不可になり、OPCクライアントとの接続が切断される可能性があります。
- OPCクライアント (クラシック) は、PSS 4000 OPCサーバ (クラシック) にのみアクセスできます。
- OPC UAクライアントは、PSS 4000 OPCサーバUAにのみアクセスできます。



重要

OPCサーバの起動

OPCクライアントがOPCサーバにアクセスできるためには、最初にサーバを起動する必要があります。この操作を行うには、さまざまな手順があります。

– [スタートアップ] メニューからの自動起動

PSS 4000ファームウェアのインストール時に、インストールウィンドウ内の既定の設定が維持されています。つまり、Windowsオペレーティングシステムの [スタートアップ] メニューに [PSS 4000ファームウェアコントローラ] が登録されます。PCを起動するたび、OPCサーバが自動的に起動されます。

– 手動起動

インストール時に、インストールウィンドウで [スタートアップ] メニューへの登録のためのオプションは選択されていません。これは、インストール後およびPCを起動するたびに、PSS 4000ファームウェアコントローラ経由でOPCサーバを手動で起動する必要があることを意味します。

インストールプログラムを使用したインストール

⇒ インストールファイル「PSS_4000_PC_<バージョン番号>_Installer.exe」をダブルクリックして、画面上の指示に従います。

アンインストールプログラムを使用したアンインストール

⇒ Windowsプログラムリストで、[Pilz] -> [PSS4000 Firmware for PC <バージョン番号>] を右クリックし、[アンインストール] を選択します。

サイレントインストール

PSS 4000ファームウェアを無人（「サイレント」）でインストールすることもできます。この場合、ユーザの操作なしでインストールが実行され、インストールプロセスはバックグラウンドで実行されます。

⇒ Windowsコマンドライン、Windows検索フィールドまたはバッチファイルに以下のいずれかのコマンドを入力して、インストールを開始します。

既定値を使用したインストールのコマンド:

"<インストールファイルの絶対ディレクトリ>" /S

例:

"c:¥myFiles¥PSS_4000_PC_<バージョン番号>_Installer.exe" /S

仕様を指定してインストールするためのコマンド:

"<インストールファイルの絶対ディレクトリ>" /S /auto='<値>' /lang='<ツールの言語>' /D='<絶対インストールディレクトリ>'

例:

"c: ¥myFiles¥PSS_4000_PC_<バージョン番号>_Installer.exe" /S /auto='1' /lang='de' /D=C: ¥myPrograms

サポートされている引数:

引数	意味	大文字／小文字の区別
/S	ファイルをサイレントに実行します。	はい
/auto	この引数によって、PCの起動時にPSS 4000ファームウェアを自動的に起動するかどうかを制御できます。引数はオプションです。 値の範囲: 1: PCの起動時にPSS 4000ファームウェアが自動的に起動する 0: PSS 4000ファームウェアが自動的に起動しない 既定値: 1	なし
/lang	メッセージの言語とアンインストールプログラムの言語 (アンインストールがサイレント実行されていない場合)。引数はオプションです。 値の範囲: de: ドイツ語 en: 英語 既定値: オペレーティングシステムの言語がドイツ語の場合は、ドイツ語がデフォルトの言語になり、それ以外の場合は英語になります。	いいえ
/D	インストールディレクトリ: 最後の引数として入力する必要があります。引数はオプション 既定値: %programfiles(x86)%¥PILZ	はい

注: インストールコマンドに「Start /wait」を追加することで、インストールコマンドの実行が完了するまで、それ以上のコマンドが開始されないようにすることができます。「start」と「/wait」の間には二重引用符が2つ必要です。

例:

Start "" /wait "c: ¥myFiles¥PSS_4000_PC_<バージョン番号>_Installer.exe" /S /auto='1' /lang='de' /D=C: ¥myProgramms

サイレントアンインストール

アンインストールはサイレントに実行することもできます。

⇒ Windowsコマンドライン、Windows検索フィールドまたはバッチファイルに以下のコマンドを入力して、アンインストールを開始します。

アンインストールコマンド:

"<アンインストールファイルの絶対ディレクトリ>" /S

例:

"C: ¥Program Files (x86)¥Pilz¥PAS4000 1.20.0¥uninstallerData¥uninstall.exe" /S

注: インストールコマンドに「Start /wait」を追加することで、アンインストールコマンドの実行が完了するまで、それ以上のコマンドが開始されないようにすることができます。「start」と「/wait」の間には二重引用符が2つが必要です。

例:

Start "" /wait "C: ¥Program Files (x86)¥Pilz¥PAS4000 1.20.0¥uninstallerData¥uninstall.exe" /S

15.5 情報セキュリティ証明書

ネットワーク接続されたオートメーション技術システムでは、情報セキュリティに対する要求が高まりつつあります。情報セキュリティには、データの信頼性、機密性、整合性、可用性と、データへのアクセスの保護が含まれます。

ピルツのPSS 4000 OPCサーバUAでは、X.509規格に準拠した証明書を利用して情報セキュリティが確立されます。証明書は、OPC UAクライアントとOPC UAサーバの間の通信を保護するために使用されます。つまり、証明書は、一方向の通信接続のデータトラフィックを暗号化し、一方の通信サブスライバを他方の通信サブスライバに認証します。

クライアントサーバ接続のサブスライバごとに個別の証明書が必要です。PSS 4000 OPCサーバUAがOPC UAクライアントから有効な証明書を取得し、OPC UAクライアントがPSS 4000 OPCサーバUAから有効な証明書を取得するまで、OPC UAクライアントはPSS 4000 OPCサーバUAと通信できません。

15.5.1 インストール中の証明書の作成

PSS 4000 OPCサーバUAのインストール時には、公開キーを含む自己署名証明書および関連付けられたプライベートキーが規格として作成されます。この証明書は試運転のために使用します。通常の操作では使用しないでください。

通常の操作に独自の証明書を作成することをお勧めします。詳細については、「[カスタム証明書の生成および使用 \[📖 558\]](#)」を参照してください。



情報

まれに、サードパーティ製のOPC UAクライアントが、インストール時に作成された証明書を認証しない場合があります。この場合、試運転でも独自の証明書を作成して使用する必要があります。

15.5.2 証明書ストアのディレクトリ構造

PSS 4000 OPCサーバUAのインストール時には、公開キーを含む自己署名証明書および関連付けられたプライベートキーが規格として作成されます（「[インストール中の証明書の作成 \[551\]](#)」を参照）。公開キーおよび非公開（プライベート）キーは、インストール先の装置の証明書ストアに保存されます。インストール中に証明書ストアとして事前定義されたディレクトリ構造が作成されます。証明書ストアの場所は、PSS 4000 OPCサーバUAがインストールされている装置によって異なります。ただし、ディレクトリ構造は常に同じです。

<basepath>%pki

%own	自己署名証明書の公開部分のストアの場所。 自己署名証明書の公開部分は、インストール中に常にこのフォルダに格納されます。
%private	自己署名証明書の非公開（プライベート）部分のストアの場所。 自己署名証明書の非公開部分は、インストール中に常にこのフォルダに格納されます。
%rejected	未知の証明書または拒否された証明書のストアの場所。
%trusted	信頼された証明書のストアの場所。



情報

証明書は、新規インストールの場合にのみ作成されます。更新の場合、証明書ストア内のすべての既存の証明書が保持されます。



重要

独自の証明書を生成する場合でも、ディレクトリ構造を使用します。

PCにインストールされた場合のストアの場所。

証明書ストアは、PCの公開エリアに作成されます。

▶ C:%ProgramData%Pilz%PSS4000OPCServerUA%pki

15.5.3 認証の手順

クライアントとサーバの間で通信可能なことを確認するには、通信パートナーが証明書を交換する必要があります。証明書を交換するには、証明書ファイルをOPC UAクライアントとPSS 4000 OPCサーバUAの証明書ストアの該当するフォルダに格納する必要があります（「[証明書ストアのディレクトリ構造](#)」 552」を参照）。

該当するPSS 4000 OPCサーバUAとOPC UAクライアントの相互認証には、さまざまな手順があります。手順は、次の条件に従って区別できます。

- ▶ PASvisuサーバをOPC UAクライアントとして使用
 - PSS 4000 OPCサーバUAとPASvisuサーバが同じ装置上にある
（「[ローカルインストールの場合の認証](#)」 553」を参照）
 - PSS 4000 OPCサーバUAとPASvisuサーバが、異なる装置にインストールされている
（「[リモートインストールの場合の認証](#)」 553」を参照）
- ▶ サードパーティ製造業者のOPC UAクライアントの使用
（「[サードパーティ製造業者のOPC UAクライアントの場合の認証](#)」 557」を参照）

15.5.3.1 ローカルインストールの場合の認証

PASvisuサーバは、OPC UAクライアントとして使用されます。PSS 4000 OPCサーバUAとPASvisuサーバが同じ装置にインストールされている場合、インストール中に証明書が自動的に交換されます。

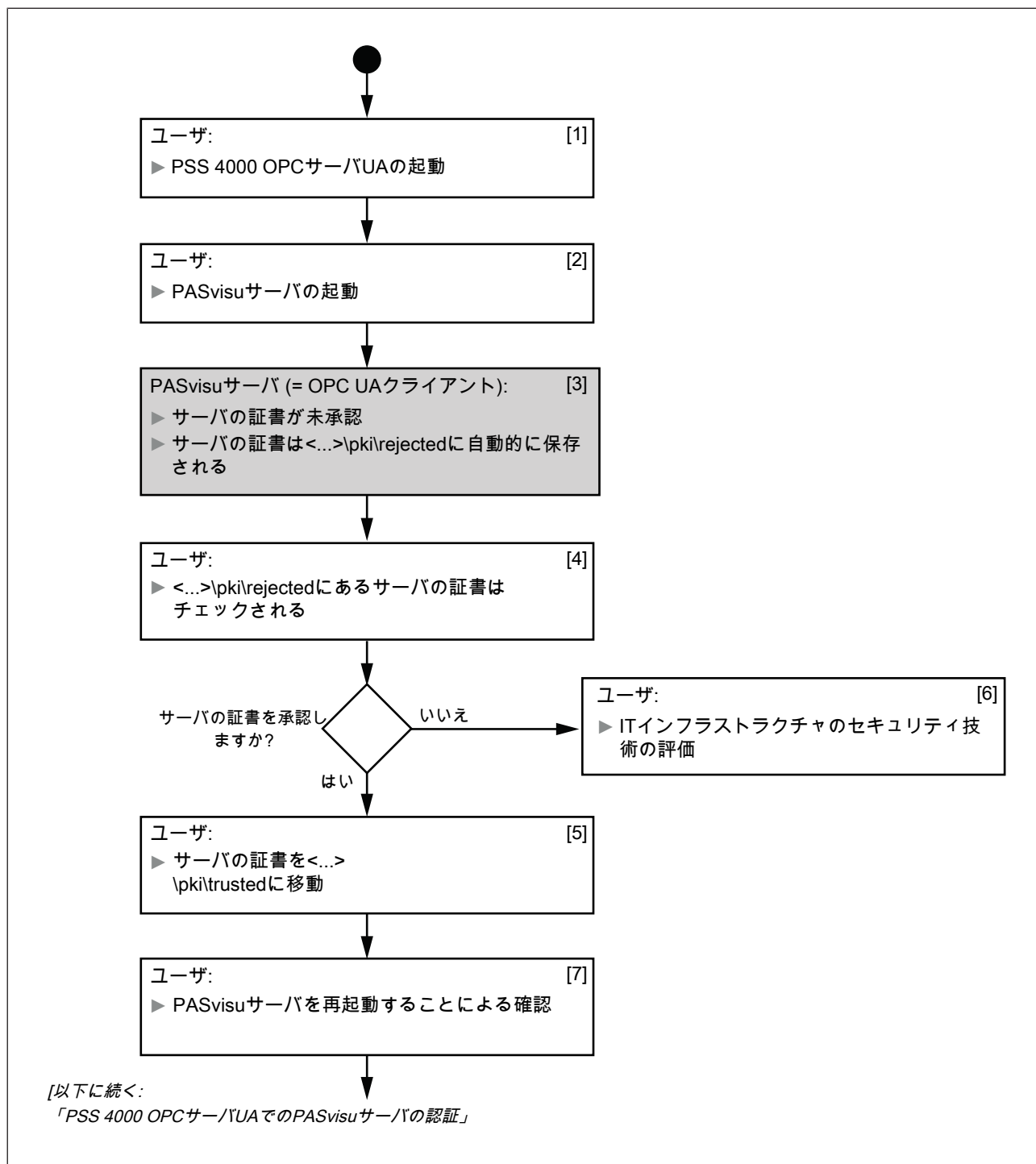
- ▶ PSS 4000 OPCサーバUAの証明書 (cert_Pilz_PSS 4000_OPC_Server_UA.der) は、PASvisuの「trusted」フォルダにコピーされます。
- ▶ PASvisuサーバの証明書 (cert_Pilz_PASvisu_OPC_UA.der) は、PSS 4000 OPCサーバUAの「trusted」フォルダにコピーされます。

ユーザが他の操作を行う必要はありません。

15.5.3.2 リモートインストールの場合の認証

PASvisuサーバは、OPC UAクライアントとして使用されます。PSS 4000 OPCサーバUAとPASvisuサーバが異なる装置にインストールされている場合は、ユーザが証明書を交換する必要があります。

手順



凡例

- [1] ユーザによるインストール後にPSS 4000 OPCサーバUAが初めて起動されたとき
- [2] ユーザによるインストール後にPASvisuサーバが初めて起動されたとき

[3] PASvisuサーバが、PSS 4000 OPCサーバUAへの接続の確立を試みます。PASvisuサーバには、インストール後にPSS 4000 OPCサーバUAからの有効な証明書がないので、PASvisuサーバは次のように応答します。

▶ サーバ証明書はPASvisuサーバによって受け入れられません

▶ サーバ証明書は、PASvisuサーバの証明書ストアのディレクトリ「<...>%pki%rejected」に保存されます。

証明書ファイルの名前は、証明書の共通名、ドメインコンポーネント、およびフィンガープリントで構成されます。

ファイル名の構造: <CN>@ <DC>_[フィンガープリント].der

- CN: 共通名

- DC: ドメインコンポーネント

- フィンガープリント: 40桁の16進数値

例:

Pilz_PSS4000_OPC_Server_UA_self_signed_@192_168_1_101_[0ff981ea8ba58ad04bad16f89a3cc9c0c60e7fac].der

- CN: Pilz PSS4000 OPC Server UA (自己署名済み)

- DC: 192.168.1.101

- フィンガープリント: 0ff981ea8ba58ad04bad16f89a3cc9c0c60e7fac

注:

CNおよびDC内のA～Z、a～z、および0～9の範囲にない文字は、すべてアンダーラインに置き換えられます。

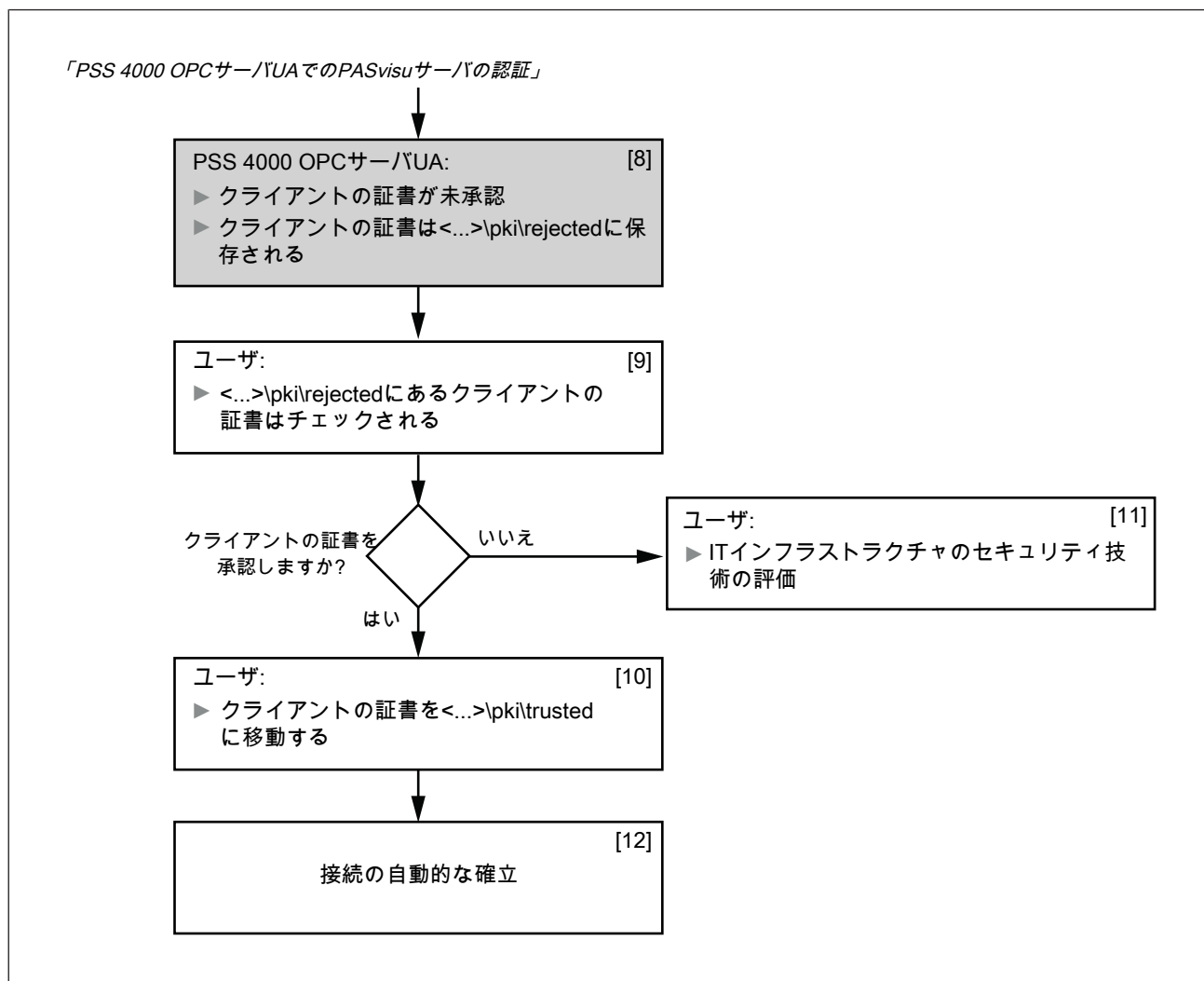
▶ PSS 4000 OPCサーバUAへの接続は確立されません。

[4] ユーザがサーバ証明書を信頼できるかどうかを確認する必要があります (「[証明書の確認](#)」を参照)。

[5] ユーザは、サーバ証明書を信頼できると見なした場合、PASvisuサーバの証明書ストアの「rejected」ディレクトリからその証明書を「trusted」ディレクトリに移動する必要があります。PSS 4000 OPCサーバUAは、PASvisuサーバ上で認証されるようになりました。

[6] サーバ証明書の信頼性について疑問がある場合は、ITインフラストラクチャのセキュリティ技術を確認する必要があります。

[7] PSS 4000 OPCサーバUAがPASvisuサーバから有効な証明書を受信するには、まずこの2つの間の接続を確立する必要があります。



凡例

- [8] PSS 4000 OPCサーバUAには、インストール後にPASvisuサーバからの有効な証明書がないので、PSS 4000 OPCサーバUAは次のように応答します。
- ▶ クライアント証明書は、PSS 4000 OPCサーバUAによって受け入れられません。
 - ▶ クライアント証明書は、PSS 4000 OPCサーバUAの証明書ストアのディレクトリ「<...>\pki\rejected」に保存されます。
- 証明書ファイルの名前は、共通名、ドメインコンポーネント、およびフィンガープリントで構成されます。
- ファイル名の構造: <CN>_<DC>_[フィンガープリント].der
- CN: 共通名
 - DC: ドメインコンポーネント
 - フィンガープリント: 40桁の16進数値

例:

Pilz_PASvisu_OPC_UA_192_168_1_170_[c0340c57bcf7a6e7d1097735d3358351b942755c].der

- CN: Pilz PASvisu OPC UA

- DC: 192.168.1.170

- フィンガープリント: c0340c57bcf7a6e7d1097735d3358351b942755c

注:

CNおよびDC内のA～Z、a～z、および0～9の範囲にない文字は、すべてアンダーラインに置き換えられます。

▶ PASvisuサーバへの接続が確立されません。

[9] ユーザがクライアント証明書を信頼できるかどうかを確認する必要があります (「[証明書の確認](#)」 [560](#)」を参照)。

[10] ユーザは、クライアント証明書を信頼できると見なした場合、PSS 4000 OPCサーバUAの証明書ストアの「rejected」ディレクトリからその証明書を「trusted」ディレクトリに移動する必要があります。

PASvisuサーバは、PSS 4000 OPCサーバUA上で認証されるようになりました。

[11] クライアント証明書の信頼性について疑問がある場合は、ITインフラストラクチャのセキュリティ技術を確認する必要があります。

[12] 相互認証に続いて、PASvisuサーバがPSS 4000 OPCサーバUAへの接続を自動的に確立します。



重要

次の条件下では、認証中に、証明書 (クライアントまたはサーバ証明書) はディレクトリ「<...>%pkirejected」のみに保存されます。

- ディレクトリ「<...>%pkirejected」に同じフィンガープリントの証明書がない。
- ディレクトリ「<...>%pkitrusted」に同じフィンガープリントの有効な証明書がない。

15.5.3.3

サードパーティ製造業者のOPC UAクライアントの場合の認証

サードパーティ製造業者のOPC UAクライアントがピルツのPSS 4000 OPCサーバUAで使用される場合、ユーザが証明書を交換する必要があります。OPC UAクライアントについては、サードパーティ製造業者の取扱説明書の手順に従ってください。

手順

▶ OPC UAクライアント

サードパーティ製造業者の取扱説明書の認証手順に従ってください。

注:

一部のOPC UAクライアントでは、OPC UAサーバの証明書が格納されるディレクトリに関する情報が必要です。インストール後に、ピルツのPSS 4000 OPCサーバUAの証明書はディレクトリ<...>%pkidownに置かれます (「[証明書ストアのディレクトリ構造](#)」 [552](#)」を参照)。

▶ PSS 4000 OPCサーバUA

ピルツのPSS 4000 OPCサーバUAの手順は原則として、PASvisuサーバをOPC UAクライアントとして使用する場合と同じです (「[リモートインストールの場合の認証](#)」 [553](#)」を参照)。

15.5.4 カスタム証明書の生成および使用

PSS 4000 OPCサーバUAやPASvisuサーバのインストール時に作成される証明書の代わりに、自己生成証明書を使用できます。これには、サードパーティ製造業者の適切なツールが必要です。公開部分と非公開(プライベート)部分がある証明書の作成に、このツールを使用する必要があります。



情報

PSS 4000 OPCサーバUAやPASvisuサーバのインストール時に作成される証明書は、試運転のために使用します。通常の操作にはカスタム証明書を作成することをお勧めします。

証明書の要件

PSS 4000 OPCサーバUAは、そのカスタムサーバ証明書を有効にしません。このため、サーバ証明書とクライアント証明書の両方に次のオプションが含まれていることを確認する必要があります。

- ▶ CA: FALSE
- ▶ パラメータNID_key_usageが使用されている場合、値cRLSignを設定しないようにする必要があります。
- ▶ 暗号化アルゴリズム: sha1RSA - 1024ビット

PSS 4000 OPCサーバUAとPASvisuサーバは、署名アルゴリズム「sha1WithRSAEncryption」を使用します。キーの長さは1024ビットです。



情報

カスタム証明書を作成して使用する場合は、次の資料に注意する必要があります。

- OPC FoundationのOPC UAの仕様
- BSIの仕様
- 該当するOPC UAクライアントの資料
- 証明書を作成するために使用されたツールの資料

証明書の組み込み

証明書を有効にするには、証明書の公開部分およびプライベート部分に正しいファイル名が指定され、各部分が証明書の保管場所の正しいディレクトリに配置されている必要があります (「[証明書ストアのディレクトリ構造](#)」 552]を参照)。

- ▶ 公開部分のファイル名:
 - PSS 4000 OPCサーバUA: 「cert_Pilz_PSS4000_OPC_Server_UA.der」
 - PASvisuサーバ: "cert_Pilz_PASvisu_OPC_UA.der"

▶ プライベート部分のファイル名:

- PSS 4000 OPCサーバUA: 「private_key_Pilz_PSS4000_OPC_Server_UA.pem」
- PASvisuサーバ: "private_key_Pilz_PASvisu_OPC_UA.pem"

証明書の公開部分およびプライベート部分の保管場所

PCにインストールされた場合の保管場所:

- ▶ <%ALLUSERSPROFILE%>%Pilz%PSS4000OPCServerUA%pkc
– %own%cert_Pilz_PSS4000_OPC_Server_UA.der
– %private%private_key_Pilz_PSS4000_OPC_Server_UA.pem
- ▶ <%ALLUSERSPROFILE%>%Pilz%PASvisu Builder%pkc
– %own%cert_Pilz_PASvisu_OPC_UA.der
– %private%private_key_Pilz_PASvisu_OPC_UA.pem

15.5.5 証明書の無効化

ユーザがディレクトリ<...>%pki%trustedから証明書を削除した場合、証明書が無効になります (「[証明書ストアのディレクトリ構造](#)」[\[📖 552\]](#)を参照)。

15.5.6 証明書の確認

認証中にディレクトリ<...>%pki%rejectedから証明書をディレクトリ<...>%pki%trustedにコピーする前に、ユーザが証明書の信頼性を確認する必要があります (「[リモートインストールの場合の認証](#)」[\[📖 553\]](#)を参照)。

▶ PASvisuサーバ

PASvisuサーバのディレクトリ<...>%pki%rejected内の証明書の内容をPSS 4000 OPCサーバUAのディレクトリ<...>%pki%own内の証明書の内容と比較する必要があります。フィンガープリントを比較するだけで十分です。

▶ PSS 4000 OPCサーバUA

PSS 4000 OPCサーバUAのディレクトリ<...>%pki%rejected内の証明書の内容をPASvisuサーバのディレクトリ<...>%pki%own内の証明書の内容と比較する必要があります。フィンガープリントを比較するだけで十分です。

PCの場合の手順

- ▶ 証明書をダブルクリックして、その内容を表示します。
比較する2つの証明書の内容が一致している必要があります。



情報

PSS 4000 OPCサーバUAが接続を確立するときに、ローカルシステムの時刻がクライアント証明書の有効な範囲 (期間) 内かどうかを確認します。このようになっていない場合、接続を確立できません。

15.6 ファイアウォールのコンフィグレーション

PCおよびネットワークのファイアウォールや、その他のセキュリティメカニズムにより、OPCサーバが接続された装置と正常に通信できないことがあります。このような場合は、セキュリティ設定とファイアウォール設定を調整する必要があります。

ファイアウォールでは、明示的に登録されたプログラムへのアクセスが許可されます。次のプログラムにはアクセス権が必要です。

- ▶ すべてのOPC UAクライアント
- ▶ PSS4000_OPCTServerUA.exe

また、データ交換のためにポートを有効にする必要があります。

- ▶ OPCサーバUA - Discovery Serverのポート番号: 4840
- ▶ OPC UAクライアント (RWクライアント) に読み取り／書き込みアクセス (ROアクセス) がある場合のPSS 4000 OPCサーバUAのポート番号:
デフォルト設定: 4853
- ▶ OPC UAクライアント (ROクライアント) に読み取り専用アクセス (ROアクセス) がある場合のPSS 4000 OPCサーバUAのポート番号:
デフォルト設定: ポート4854



重要

ROクライアントの書き込みアクセスを禁止する

OPC UAサーバへのROアクセスを持つOPC UAクライアント (ROクライアント) は、RWアクセス用のポートを使用できません。

ファイアウォールを使用して、OPC UAクライアントからOPC UAサーバへの不要な書き込みアクセスを防止してください。読み取り／書き込みエンドポイントへのROクライアントからのアクセスが防止されるようにファイアウォールを設定します。つまり、PAS4000でRWアクセス用にコンフィグレーションされたポートを経由してROクライアントとOPC UAサーバ間の通信が行われることがないようにします。

- ▶ ログ: TCP

これらを有効化することで、ネットワーク上またはインターネット上の他のユーザに対して、PCへのアクセス権が付与されます。ネットワーク管理者は、これらのファイアウォール設定を最適化して、不正アクセスからサーバを保護することができます。

「[ネットワークデータ \(OPCサーバUA\)](#)」[\[564\]](#)」の項も参照してください。

15.7 データ型の割り付け

OPC UA規格により、さまざまなデータ型が割り付けられます。以下のテーブルに、PSS 4000 OPCサーバUAでサポートされているデータ型と、PSS 4000のユーザプログラムで使えるデータ型への割り付けを示します。

PSS 4000ネームスペース	OPC UA	ユーザプログラム (PSS 4000)
BOOL値	BOOL値	BOOL
SByte	SByte	SINT
Byte	Byte	BYTE
		USINT
Double	Double	LREAL
		TIME
Float	Float	REAL
Int16	Int16	INT
Int32	Int32	DINT
Int64	Int64	LINT
String	LocalizedText	- - -
UInt16	UInt16	WORD
		UINT
UInt32	UInt32	DWORD
		UDINT
UInt64	UInt64	LWORD
		ULINT
DateTime	DateTime	DATE
		TIME_OF_DAY
		DATE_AND_TIME

15.7.1 PSS 4000ネームスペース内のデータへのアクセス

PSS 4000 OPCサーバは、PSS 4000プロジェクトから一般的なプロジェクトデータおよび装置ベースのプロジェクトデータを収集し、PSS 4000ネームスペースに提示します。PSS 4000ネームスペースは、定義済み構造の特別なメモリ領域です。OPCクライアントは、この定義済み構造を介してPSS 4000ネームスペースのデータにアクセスできます。

PSS 4000ネームスペースとその構造についての詳細は [PSS 4000ネームスペース](#)  [586](#) を参照してください。

15.7.2 ネームスペース内の配列にアクセスする際の特別な機能

ネームスペース内の配列は通常、1つのノードとして表されます。つまり、配列の個々の要素には、個別のノードとしてアクセスできません。ただし、新しいOPC UAクライアントでは、IndexRange関数を使用して配列の個々の要素にアクセスできます。このタイプのアクセスは、「インデックスベースのアクセス」とも呼ばれます。インデックスベースのアクセスをサポートしないOPC UAクライアントが配列の個々の要素にアクセスできるようにするために、すべての要素を含むネームスペースで配列を表すことができます。つまり、PSS 4000 OPCサーバUAのネームスペースでは、配列のすべての要素に個別のノードとしてアクセスできます。このタイプのアクセスでは、OPC UAサーバのメモリ要件が高くなることに注意してください。これは、配列の各要素をネームスペースで個別のノードとして使用できるようにするためです。

配列がネームスペースに表示される方法は、PAS4000を使用してPSS 4000 OPC サーバ UA用にコンフィグレーションできます。

15.8 ネットワークデータ (OPCサーバUA)

プロトコル	方向 [°]	転送プロトコル	ポート番号	無効化機能あり	瞬時の処理が必要	説明
SNTP	入力	UDP (IPv4)	123	はい デフォルト: 無効	いいえ	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載したPMI/PC アプリケーション: ▶ 時刻同期 ▶ SNTPサーバ
SNTP	出力	UDP (IPv4)	123	はい デフォルト: 無効	いいえ	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載したPMI/PC アプリケーション: ▶ 時刻同期 ▶ SNTPクライアント
SNp	入力/出力	UDP (IPv4)	40000	いいえ	はい	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載したPMI/PC アプリケーション: ▶ 読み取りおよび取得用のSafetyNET p経由の通信 – プロセスデータ – ステータス情報 – イベントメッセージ
SNp MSC	入力	UDP (IPv4)	40000	いいえ	いいえ	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載したPMI/PC アプリケーション: ▶ 以下のためのIPマルチキャストとのSafetyNET p通信 – ネットワークのスキャン – 装置の命名 – 装置の識別 実装されているセキュリティ対策: ▶ マルチキャストアドレスは次の値に制限: 224.0.23.61

プロトコル	方向 [*]	転送プロトコル	ポート番号	無効化機能あり	瞬時の処理が必要	説明
SNp MSC	出力	UDP (IPv4)	40000	いいえ	いいえ	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載したPC アプリケーション: ▶ 以下のためのIPマルチキャストとのSafetyNET p通信 – ネットワークのスキャン – 装置の命名 – 装置の識別 実装されているセキュリティ対策: ▶ マルチキャストアドレスは次の値に制限: 224.0.23.61
OPC UA DA	入力	TCP (IPv4)	1 ... 65535 デフォルト: 4853、4854	はい デフォルト: 無効	いいえ	影響を受ける装置: ▶ PSS 4000ファームウェアを搭載したPMI/PC ポート番号のデフォルト設定: ▶ RWアクセスを持つOPC UAクライアント: 4853 ▶ ROアクセスを持つOPC UAクライアント: 4854 実装されているセキュリティ対策: ▶ 証明書を使用したPSS 4000 OPCサーバUAと関連するOPC UAクライアントの相互認証 ▶ 署名および暗号化方式によるX.509規格に適合した証明書
OPCサーバUA - Discovery Server	入力	TCP (IPv4)	4840	はい デフォルト: 有効	いいえ	影響を受ける装置: ▶ PC用PSS 4000ファームウェア搭載PC

[*]

入力: 通信相手が装置との通信を開始します。

出力: 装置が通信相手との通信を開始します。

「[ファイアウォールのコンフィグレーション](#)」 561」の項も参照してください。

16 診断の基本情報

機械のダウンタイムを最小限に抑えるためには、効果的な診断が必要です。診断により、エラーを素早く回復し、エラーを防止し、機械の状態を分析するのに必要なすべての情報が提供される必要があります。

オートメーションシステムPSS 4000では、システム診断とプロセス診断が区別されています。

▶ システム診断

ハードウェアのコンフィグレーションやユーザプログラムのエラー検出を含む、ハードウェアおよびファームウェアの診断です。

すべてのシステム診断がピルツによって提供されます。ユーザにできる調節は、ロケーション情報に付加情報を入力するなど、ごくわずかしかなりません。

▶ プロセス診断

手順と状態の診断。プロセス診断は、ユーザがユーザプログラム内に作成します。ピルツのブロックにはプロセス診断が標準装備されていて、ユーザは各自の要件を満たすように変更することができます。

オペレーション中に、システムメッセージおよびプロセスメッセージが[診断リスト](#)  [567](#)に入力されます。装置ごとに独自の診断リストがあります。診断サーバは、プロジェクト内の全装置の装置診断リストを結合して、プロジェクトの診断リストを作成します。

診断リストには現在のメッセージのみが含まれます。エラーが回復されると、診断リストからメッセージが削除されます。

診断リストに表示される診断メッセージの順序は、診断メッセージの優先順位によって決まります。これにより、最も重要なメッセージを簡単に確認できます。

システム診断では、最初のエラーとその後のエラーが区別されます。最初のエラーだけが報告されます。これにより、診断リスト内に不要なメッセージが入らないようになります。

PSS 4000専用装置がRUN状態を終了すると、システムセクションはわずかなオフセットでSTOP状態に切り替わります。結果として診断メッセージが表示される場合があります。これは不必要に見えますが、実際にはオペレーティングステータスを正しく表しています。

発生したイベントやその位置の記述の他に、診断メッセージには推奨される処置も含まれます。これらは何をすべきか、どのエリアが影響を受けているか、誰が問題の解決を担当するかを示しています。

[診断ログ](#)  [581](#)は、診断メッセージが発生した日時とクリアされた日時を記録します。また、たとえば装置が再起動されたなどの場合には、オペレーティングステータスの変更についての情報も、ピルツのテクニカルサポートのための情報と共に診断ログに含まれます。

装置ごとに独自の診断ログがあります。診断サーバは、プロジェクト内の全装置の装置診断ログを結合して、プロジェクトの診断ログを作成します。

[診断サーバ](#)  [584](#)は、そのプロジェクト診断リストとログを診断クライアント (PSS 4000 OPCサーバなど) に転送します。その後、診断クライアントが表示機器 (PASvisuプロジェクトを含むPMIなど) に情報を提供します。

プロジェクト診断リストとプロジェクト診断ログも PAS4000 に表示されます。PAS4000の診断リストと診断ログは、オンラインオペレーションに関するメッセージ／エントリなども含むため、表示機器 (PASvisuプロジェクトを含むPMIなど) の診断リストおよび診断ログと多少異なる場合があります。

装置診断リストとログは表示されません。

16.1 診断リスト

動作中に、[診断メッセージ](#)  [568](#)が診断リストに入力されます。

診断リストには、現在のメッセージ（「アクティブ」状態の診断メッセージ）のみが含まれます（「[診断メッセージのステータス](#)  [570](#)」を参照）。

診断メッセージの状態が [active] に変わると、診断リストに新しい診断メッセージが入力されます。診断メッセージは、状態が [inactive] に変わると削除されます。

装置ごとに独自の診断リストがあります。1つのプロジェクトのすべての装置の診断リストは、プロジェクト診断リストにまとめられます。

▶ 装置の診断リスト

- 装置のファームウェアによって作成される
- 装置の診断メッセージを含む（未加工データのみ、テキストなし）
- 少なくとも128件の診断メッセージに対応できるメモリ
- 診断メッセージの数が最大数を超えると、最も優先度が低い診断メッセージが削除される
- 保存が間に合わないほど多数のプロセス診断メッセージが短時間に発生した場合、対応する診断メッセージが装置診断リストに入力されて、その他のすべてのプロセス診断メッセージはすべて無視される
- 保存が間に合わないほど多数の診断メッセージが短時間に発生した場合、対応する診断メッセージが装置診断リストに入力されて、その他のすべてのプロセス診断メッセージはすべて無視される。できる限り多くのシステム診断メッセージを処理しようとし続けるが、一部のメッセージは取得されない場合がある
- オリジナルリセットのイベントによって削除される
- 表示できない。プロジェクトの診断リストのベースとして使用される

▶ プロジェクト診断リスト

- [診断サーバ](#)  [584](#)によって生成
- プロジェクトの全装置に関するブレーンテキストの診断メッセージを含む
- 少なくとも256件の診断メッセージに対応できるメモリ
- 診断メッセージの数が最大数を超えると、最も優先度が低い診断メッセージが削除される
- 装置の診断リストと継続的に同期されるため、常に最新状態
- 診断サーバが装置への接続を失った場合、装置のすべての診断メッセージがプロジェクト診断リストから削除され、対応する診断メッセージがプロジェクト診断リストに入力される
- PAS4000の診断リストは、オンラインアクションなどのメッセージも含むため、表示機器（PASvisuプロジェクトを含むPMIなど）の診断リストと多少異なる可能性がある

16.1.1 診断メッセージ

次の診断メッセージを使用できます。

- ▶ システムメッセージ

イベント発生時にシステム診断に発行されるメッセージ

- ▶ プロセスメッセージ

イベント発生時にプロセス診断に発行されるメッセージ (設定条件が満たされた場合)

オペレーションの最中に、診断メッセージが診断リストに入力されます。

診断メッセージには、イベントの説明が記載されます。各イベントメッセージには対処方法が含まれます。対処方法には、イベントに対する対処方法が表示されます。処置は、最大8つの手順 (「アクション」) で構成されます。

診断メッセージのコンポーネント:

- ▶ タイムスタンプ

診断メッセージが出された日時診断メッセージでは、時刻は標準時 (UTC) で保存されます。診断メッセージを表示すると、タイムスタンプは表示装置の現地時間で表示されます。現地時間は、表示装置で設定されているタイムゾーンに基づいて世界標準時から計算されます。

- ▶ 診断ID

システム診断では、診断IDは診断メッセージの一意のIDとして使用されます。プロセス診断では、診断IDは診断メッセージに関連する診断項目の一意のIDとして使用されます (「[診断ID](#)」を参照)。

- ▶ 重篤度と優先度

診断メッセージの重要度
(「[重大度と優先度](#)」を参照)

- ▶ 範囲

イベントの種類の大まかな説明
(「[範囲](#)」を参照)

- ▶ 説明 (メッセージテキスト)

発生した内容の説明

- ▶ ロケーション情報

イベントが発生したロケーションに関する情報 (「[ロケーション情報](#)」を参照)。
診断メッセージのロケーション情報に含まれる詳細:

- 装置
- 装置ID
- アドレス
- ロケーションの説明

- ▶ パラメータリスト

テクニカルサポートが診断メッセージを評価するために必要なすべての情報が含まれます。これは診断IDとすべてのパラメータです。

▶ 状態

診断メッセージの状態を示します。[active] または [inactive]

(「[診断メッセージの状態](#)  570」を参照)

アクションの構成要素

▶ 範囲

アクションの種類の大まかな説明

(「[範囲](#)  573」を参照)

▶ 責任者

このアクションを取るべき責任者

(「[責任者](#)  572」を参照)

▶ 説明 (アクションテキスト)

対処方法に関する説明

▶ ロケーション情報

アクションを実施するべきロケーションに関する情報 (「[ロケーション情報](#)  574」を参照)。

アクションの構成要素のロケーション情報には次の詳細が含まれます。

- 装置
- 装置ID
- アドレス
- ロケーションの説明

16.1.1.1

診断メッセージのステータス

診断メッセージの状態は、そのトリガになったイベントに応じて異なります。

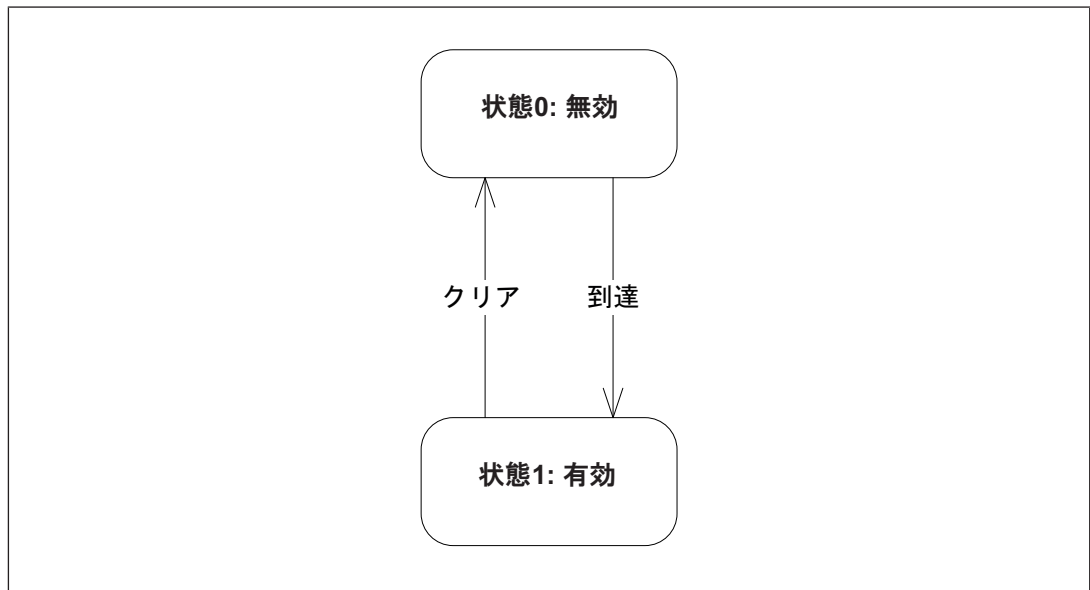


図: 診断メッセージのステータス

▶ システム診断

イベントが発生している場合 (入力での短絡など)、診断メッセージは [active] になります (状態の変化: [arrives])。異常が修正されると、診断メッセージは [inactive] になります (ステータスの変化: [clears])。

▶ プロセス診断

プログラムされた条件が満たされると、診断メッセージは [active] に変わります (状態の変化: [arrives])。条件を満たさなくなると、診断メッセージは [inactive] になります (状態の変化: [clears])。

診断リストには、状態1の診断メッセージのみが含まれます。

診断メッセージの状態が変化するたびに、診断ログにログエントリがトリガされます。

状態の変化は次のように記号で表されます。

- ▶ 「arrived」の場合は <+>
- ▶ 「cleared」の場合は <->

16.1.1.2

重篤度と優先度

診断メッセージはその重要度によって区別されます。これらは、次の3つのレベルの重篤度に分けられます。

**エラー**

ユーザが即座に対処しなければならない診断メッセージ。この診断メッセージは、1つまたは複数の機能に異常があるか、または悪影響が及んでいるシステムの状態に関する情報を提供します。

**警告**

ユーザの注意を促す診断メッセージ。この診断メッセージは、潜在的または切迫した重大なシステム状態に関してユーザに警告します。

**ステータス情報**

プロセスやシステムの現在の状況を説明する診断メッセージ。この診断メッセージは情報を提供するためのもので、オペレーションが妨げられたり、悪影響が及んでいたりするものではありません。

優先度を使用すると、診断メッセージをより具体的に区別できます。各レベルに、5種類ずつの優先度があります。優先度1が最高の優先度で、優先度15が最低です。

重篤度	優先度	優先度の名前
エラー	1	
	2	
	3	エラー
	4	
	5	
警告	6	
	7	
	8	警告
	9	
	10	
ステータス情報	11	
	12	
	13	ステータス情報
	14	
	15	

ユーザはこれらの優先度に名前を割り付けることができます。優先度3の名前「エラー」、優先度8の名前「警告」、優先度13の名前「ステータス情報」は、システム診断およびピルツが作成したプロセス診断で使用されており、ピルツによって事前定義されています。

これらの事前定義された名前は、ユーザが変更することはできますが、削除することはできません。

16.1.1.3 責任者

責任者は、処置アクションを実施する責務を担う担当者または担当グループです。

一部の責任者はピルツによって事前定義されており、ユーザは最大15の責任者を定義できます。事前定義された責任者はシステム診断、およびピルツの作成したプロセス診断で使用されます。ユーザは、事前定義された責任者の名前を変更することはできますが、削除することはできません。

事前定義の責任者

▶ オペレータ

このアクションは、安全装置の解除、駆動、または操作に関連します。

担当者:設備のオペレータなど

▶ 電気設備のメンテナンスエンジニア

このアクションは、短絡の修正、供給電圧の確認、スイッチ接点の確認など、ピルツ製装置以外の機械／電気／電子に関連します。

担当者:メンテナンスエンジニアなど

▶ 装置の交換

装置に不具合があり、交換が必要です。

担当者:メンテナンスエンジニアなど

▶ プロジェクトエンジニア

このアクションは、PAS4000を用いて行うか、またはハードウェアコンフィグレーションの確認や適応、プログラミングの確認や適応など、何らかのプログラミングの知識を必要とします。

担当者:メンテナンスエンジニア、プログラマ、プロジェクトエンジニアなど

▶ 通信

このアクションは、SafetyNET p、IP接続、フィールドバスについて技術的に理解し、バスの負荷やケーブル長などを点検できるユーザが実行できます。

担当者:メンテナンスエンジニア、バス専門家など

▶ ---

責任者を限定するのは不可能か、またはその必要はありません。

16.1.1.4 範囲

範囲はイベント／アクションの種類を大まかに記述したものです。範囲は診断メッセージおよび処置アクションに記載されます。

一部の範囲はピルツによって事前定義されており、ユーザは最大15の範囲を定義できます。事前定義された範囲はシステム診断、およびピルツの作成したプロセス診断で使用されます。ユーザは、事前定義された範囲の名前を変更することはできますが、削除することはできません。

事前定義の範囲:

▶ ハードウェア

機械、電気、電子、またはオペレーティングシステムに影響します。

例:故障した装置、周辺機器のエラー (配線、アクチュエータ／センサ、供給電圧の異常など)、バスエラー (SafetyNET pやイーサネットのエラーなど)

▶ プロジェクトのコンフィグレーション

ユーザプログラムとハードウェアのコンフィグレーションに影響します。

例:ユーザプログラム内でのゼロによる除算

▶ 命令

シーケンスまたは命令に影響します。

例:シーケンスエラー (シリンダが最終位置に達しないなど)、安全装置がトリガされたなど (非常停止ボタンが操作された)

▶ 設備

ハードウェアまたはプロジェクトのコンフィグレーションに影響します。

▶ ---

対象を限定するのは不可能か、またはその必要はありません。

16.1.1.5 ロケーション情報

ロケーション情報は、診断メッセージおよび個々の処置アクションに表示されます。診断メッセージには、そのイベントが発生したロケーションの詳細が含まれます。アクションの情報には、そのアクションを実行する位置に関する詳細が含まれます。

ロケーション情報に含まれる詳細:

- ▶ 装置

関連するファンクション

- ▶ 装置ID

装置IDはプロジェクト内のハードウェアを識別するもので、それぞれ一意でなければなりません。

- ▶ アドレス

アドレスは自動的に生成されます。

- ▶ ロケーションの説明

装置がある制御盤の情報など、ロケーションに関する追加の情報です。

情報の内容は、ロケーションに応じて異なります。使用可能なロケーション:

- ▶ モジュールまたは装置

- ▶ 端子部

- ▶ センサ／アクチュエータ

- ▶ ブロックインスタンス

- ▶ ユーザプログラム内の位置

- ▶ 他のロケーション

個別のロケーションとロケーション情報の詳細を、以下に記載します。



情報

診断リストまたは診断ログに、ロケーション情報ではなく「<??>」と表示されている場合、現行のプロジェクトがすべての装置にダウンロードされていないか、またはI/Oマッピングが一部完了していません (「[診断サーバ](#)」 584)を参照)。

モジュールまたは装置

システム診断メッセージとログエントリでのみ使用されます。

- ▶ 装置

ユーザが、モジュールまたは装置の「装置用テキスト」ロケーション情報として入力したテキスト。既定の設定は、モジュールまたは装置の製品型式 (「PSSu E F 4DI」など) です。

まれに、PSS 4000専用装置で、診断メッセージのトリガになったモジュールを判断できない場合があります。この場合は装置として「匿名モジュール」と表示されます。

- ▶ 装置ID

ユーザが入力したモジュールまたは装置の装置ID (「-4A1」など)

- ▶ アドレス

モジュールまたは装置の自動生成アドレス

▶ ロケーションの説明

ユーザが入力したモジュールまたは装置の追加情報

端子部

システム診断またはプロセス診断で、ベースとなるPI変数がモジュールバスのI/Oデータにマッピングされている場合:

▶ 装置

端子があるモジュールまたは装置の「装置」ロケーション情報 (「モジュールまたは装置」の「装置」を参照)

▶ 装置ID

端子があるモジュールまたは装置の装置IDで、端子の詳細が含まれる (たとえば [-4A1:12] の場合、「12」は端子を指す)

▶ アドレス

自動生成される端子アドレス

▶ ロケーションの説明

ユーザが入力するモジュールまたは装置の付加情報。端子自体のロケーションの説明は入力不可

プロセス診断で、ベースとなるPI変数が外部I/Oデータにマッピングされている場合:

▶ 装置

外部I/Oデータを含むモジュールまたは装置の「装置」ロケーション情報 (「モジュールまたは装置」の「装置」を参照)

▶ 装置ID

外部I/Oデータを含むモジュールまたは装置の装置ID (「モジュールまたは装置」の「装置ID」を参照)

▶ アドレス

外部I/Oデータの自動生成アドレス

▶ ロケーションの説明

外部I/Oデータを含むモジュールまたは装置のロケーションの説明 (「モジュールまたは装置」の「ロケーションの説明」を参照)

プロセス診断で、ベースとなるPI変数が他のPI変数にマッピングされている場合:

▶ 装置

マッピングされたPI変数のリソースがあるモジュールまたは装置の「装置」ロケーション情報 (「モジュールまたは装置」の「装置」を参照)

▶ 装置ID

マッピングされたPI変数のリソースがあるモジュールまたは装置の装置ID (「モジュールまたは装置」の「装置ID」を参照)

▶ アドレス

マッピングされたPI変数の自動生成アドレス

▶ ロケーションの説明

マッピングされたPI変数のリソースがあるモジュールまたは装置のロケーションの説明 (「モジュールまたは装置」の「ロケーションの説明」を参照)

センサ／アクチュエータ

システム診断またはプロセス診断で、ベースとなるPI変数がモジュールバスのI/Oデータにマッピングされている場合:

- ▶ 装置
ユーザが、センサまたはアクチュエータの「装置用テキスト」ロケーション情報として入力したテキスト。
- ▶ 装置ID
ユーザが入力したセンサまたはアクチュエータの装置ID
- ▶ アドレス
センサまたはアクチュエータが接続されている端子の自動生成アドレス
- ▶ ロケーションの説明
ユーザが入力したセンサまたはアクチュエータの追加情報

プロセス診断で、ベースとなるPI変数が外部I/Oデータにマッピングされている場合:

- ▶ 装置
外部I/Oデータを含むモジュールまたは装置の「装置」ロケーション情報 (「モジュールまたは装置」の「装置」を参照)
- ▶ 装置ID
外部I/Oデータを含むモジュールまたは装置の装置ID (「モジュールまたは装置」の「装置ID」を参照)
- ▶ アドレス
外部I/Oデータの自動生成アドレス
- ▶ ロケーションの説明
外部I/Oデータを含むモジュールまたは装置のロケーションの説明 (「モジュールまたは装置」の「ロケーションの説明」を参照)

プロセス診断で、ベースとなるPI変数が他のPI変数にマッピングされている場合:

- ▶ 装置
マッピングされたPI変数のリソースがあるモジュールまたは装置の「装置」ロケーション情報 (「モジュールまたは装置」の「装置」を参照)
- ▶ 装置ID
マッピングされたPI変数のリソースがあるモジュールまたは装置の装置ID (「モジュールまたは装置」の「装置ID」を参照)
- ▶ アドレス
マッピングされたPI変数の自動生成アドレス
- ▶ ロケーションの説明
マッピングされたPI変数のリソースがあるモジュールまたは装置のロケーションの説明 (「モジュールまたは装置」の「ロケーションの説明」を参照)

ブロックインスタンス

プロセス診断メッセージとログエントリでのみ使用されます。

▶ 装置

Multiブロック／POUのタイプ名

▶ 装置ID

未使用

▶ アドレス

Multiブロック／POUのFQN (PSS 4000名前空間での正式名称)

▶ ロケーションの説明

未使用

ユーザプログラム内の位置

システム診断メッセージとログエントリでのみ使用されます。

▶ 装置

Multiブロック／POUのタイプ名

▶ 装置ID

未使用

▶ アドレス

位置データが付加されたMultiブロック／POUのFQN (PSS 4000名前空間での正式名称)

Multiブロックの位置データ: MultiエディタのX軸とY軸; Xは列数で、Yは行数。行数と列数はMultiエディタに表示できます。

IL-POUおよびSTL-POUの位置データ: 行数

LD-POUの位置データ: ネットワーク番号

Multiブロック／POUが保護されている場合、FQN (PSS 4000名前空間での正式名称) は、影響を受けるMultiブロック／POUの前の、保護されていない最後のMultiブロック／POUの詳細で終了します。

この位置は、保護されたMultiブロック／POUが呼び出されたポイントを示します。

▶ ロケーションの説明

未使用

他のロケーション

上記のロケーション情報が適切でない場合、ロケーション情報を自由に定義できます。

▶ 装置

プロセス診断: 任意 (診断項目のタイプまたはインスタンスで定義)

システム診断: ロケーション指定 (「ハードウェアのコンフィグレーション」や「ユーザプログラム」など)

▶ 装置ID

プロセス診断: 任意 (診断項目のタイプまたはインスタンスで定義)

システム診断: 未使用

▶ アドレス

未使用

▶ ロケーションの説明

プロセス診断: 任意 (診断項目のタイプまたはインスタンスで定義)

システム診断: 未使用

16.1.1.6 診断ID

システム診断では、診断IDは診断メッセージの一意のIDとして使用されます。プロセス診断では、診断IDは診断メッセージに関連する診断項目の一意のIDとして使用されます。プロセス診断では、その一意性はプロジェクトに限定されます。

診断IDは次の3つの部分から成ります。

<種類> - <クラス> - <番号>

▶ 種類

診断の種類を示す1文字

S: システム診断

P: ピルツによって生成されたプロセス診断

C: ユーザによって生成されたプロセス診断

▶ クラス

クラスを示す最大4桁の10進数。クラスの意味は、診断の種類に応じて変わります。

– システム診断

クラスはエラーの種類 (装置エラー、ユーザプログラムなど) を定義します。

システム診断のクラス:

100～110 装置エラー

120 装置プロジェクトで発生したエラー

121 命名データで発生したエラー

122 起動コマンドで発生したエラー

123 ファームウェアの更新で発生したエラー

200～220 ハードウェアレジストリ／ハードウェアコンフィグレーションのエラー

300 FSまたはSTリソースシステムセクションのエラー

303 ライセンスが見つからない

310 VISUリソースシステムセクションのエラー

400 周辺機器エラー

410 供給電圧の異常

600～602 FSまたはST SafetyNET pシステムセクションのエラー

700～710 FSまたはSTモジュールバスシステムセクションのエラー

801 PAS4000の内部エラー

908 Ethernetエラー

910、911 IP接続システムセクションのエラー

912、913 PROFIBUS-DPスレーブシステムセクションのエラー

914、915 EtherNet/IPアダプタシステムセクションのエラー

916、917 PROFIsafeシステムセクションのエラー

918、919 PROFINET IO DEVICEシステムセクションのエラー

9000 診断ログの運転エントリ

9001 診断ログのセルフテストエントリ

9900 診断ログのデバッグエントリ

– ピルツによって生成されたプロセス診断

クラスは、診断メッセージが属するピルツブロックを示します。クラスは、インスタンスではなくブロックのタイプを指します。

– ユーザによって生成されたプロセス診断

クラスの意味は、ユーザが決定できます。ピルツのプロセス診断のクラスと同様の方法でを使用することを推奨します。

▶ 番号

クラス内の診断メッセージ／診断項目の一意の番号。番号は4桁の10進数です。

16.2 診断ログ

診断ログは、診断メッセージの発生時と問題の解決時に記録されます。また、たとえば装置が再起動されたなどの場合には、オペレーティングステータスの変更についての情報も、ピルツのテクニカルサポートのための情報と共に診断ログに含まれます。

装置ごとに独自の診断ログがあります。プロジェクトのすべての装置の[ログエントリ](#)  [582](#)は、プロジェクト診断ログにまとめられます。

▶ 装置診断ログ

- 装置のファームウェアによって作成される
- 装置のログエントリを含む (未加工データのみ、テキストなし)
- 少なくとも256件のログエントリに対応できるメモリ。
メモリ領域の1つは、システム、プロセスおよびセルフテストのエントリに使用し、他のメモリ領域は運転エントリに使用します (「[ログエントリ](#)  [582](#)」を参照)。
- ログエントリの最大数に達した場合、最も古いログエントリが削除される。
これは、システム、プロセス、セルフテストのエントリ用のメモリ領域、運転エントリ用のメモリ領域でそれぞれ行われます。
- 表示できない。プロジェクト診断ログのベースとして使用される

▶ プロジェクト診断ログ

- [診断サーバ](#)  [584](#)によって生成
- プロジェクトの全装置に関するプレーンテキストのログエントリを含む
- 4096以上のログエントリ用のメモリ。
1つのメモリ領域が、システム、プロセス、およびセルフテストエントリ用に予約され、もう1つのメモリ領域が、オペレーションエントリ用に予約される (「[ログエントリ](#)  [582](#)」を参照)。
- ログエントリの最大数に達した場合は、最も古いログエントリが削除される。
これは、システム、プロセス、セルフテストエントリのメモリ領域内、およびオペレーションエントリのメモリ領域内でそれぞれ行われます。(このため、プロジェクト診断ログには非常に古いオペレーションエントリが含まれ、同時に新しいシステム、プロセス、およびセルフテストエントリが含まれる場合があります、また、その逆の場合もあります)。
- 装置診断ログと継続的に同期されるため、常に最新状態
- 診断サーバが装置への接続を失った場合、装置の以前のログエントリはすべてプロジェクト診断ログに保持される。接続が解除されたことを示すログエントリも表示される。
- PAS4000の診断ログは、オンラインアクションなどのエントリも含むため、表示機器 (PASvisuプロジェクトを含むPMIなど) の診断ログと多少異なる可能性がある

16.2.1 ログエントリ

次のログエントリがあります。

▶ システムエントリ

状態の変化 (システムメッセージが生成されたり、クリアされたときなど) のシステムメッセージに関する情報。システムエントリはシステム診断の一部です。

▶ プロセスエントリ

状態の変化 (プロセスメッセージが生成されたり、クリアされたときなど) のプロセスメッセージに関する情報。プロセスエントリはプロセス診断の一部です。

▶ 運転エントリ

オペレーティングステータスの変化 (装置が再起動されたときなど) やオペレーションに関連するその他の変化 (プロジェクトダウンロードの実行など) に関する情報。運転エントリはシステム診断の一部です。

▶ セルフテストエントリ

オペレーティングステータスが「ベースシステム起動中」の場合に、セルフテスト中に発生した異常に関する情報。装置を起動できないことを示します。

▶ デバッグエントリ

ピルツのテクニカルサポート用の情報。デバッグエントリはシステム診断の一部で、テクニカルサポートによって有効にされた場合にのみ表示されます。

オペレーションの最中に、ログエントリが診断ログに入力されます。

システムエントリとプロセスエントリでは、ログエントリは診断メッセージと一致しますが、処置などの一部の情報は省略されます。タイムスタンプの意味は異なり、状態ではなく状態の変化が表示されます。

ログエントリのコンポーネント (運転エントリとデバッグエントリはリストに含まれません):

▶ タイムスタンプ

診断メッセージの状態が変化した日時 (「[診断メッセージの状態](#)  570」を参照)。ログエントリでは、時刻は標準時 (UTC) で保存されます。ログエントリを表示すると、タイムスタンプは表示装置の現地時刻で表示されます。現地時刻は、表示装置で設定されているタイムゾーンに基づいて世界標準時から計算されます。

▶ 状態の変化

状態変化の記号 (「[診断メッセージの状態](#)  570」を参照)

「arrived」の場合は <+>

「cleared」の場合は <->

または

運転エントリ、セルフテストエントリ、デバッグエントリの場合は <#>

▶ 診断ID

診断メッセージの診断ID (「[診断ID](#)  579」を参照)

▶ 重篤度と優先度

診断メッセージの重要度 (「[重篤度と優先度](#)  571」を参照)

▶ 範囲

イベントの種類の大きな説明 (「[範囲](#)  573」を参照)

▶ 説明

発生した内容の説明

▶ ロケーション情報

イベントが発生したロケーションに関する情報 (「[ロケーション情報](#) [ 574]」を参照)。

診断ログのロケーション情報に含まれる詳細:

- 装置
- 装置ID
- アドレス
- ロケーションの説明

16.3

診断サーバ

プロジェクトの診断データを表示したい場合、診断サーバが必要です。

診断サーバは、そのプロジェクト診断リストとログを診断クライアント (PSS 4000 OPCサーバなど) に転送します。その後、診断クライアントが表示機器 (PASvisuプロジェクトを含むPMIなど) に情報を提供します。

診断サーバは、プロジェクト内の全装置の装置診断リストと装置診断ログの未加工データを収集し、そのデータを用いてプロジェクト診断リストとプロジェクト診断ログをプレーンテキストで生成します。言い換えると、診断サーバは、装置診断リストと装置診断ログのメッセージテキストとロケーション情報への参照を取得し、それを読みやすいテキストに置き換えます。

診断サーバには、現在のプロジェクトデータ (ユーザプログラム、ハードウェアのコンフィグレーション、リソースの割り付け、I/Oマッピング) が常に必要です。

診断サーバが現在のプロジェクトデータを使用できない場合、メッセージテキストとロケーション情報への参照を正しく置き換えることができません。この場合は、不正確なメッセージテキスト／ロケーション情報に置き換えられるか、テキストが利用できない場合には「<??>」になります。これは、診断サーバに現在のプロジェクトデータがあるが、プロジェクト内の装置がまだプロジェクトの古いバージョンで動作している場合にも発生します。

▶ 表示機器上に表示

プロジェクト診断リストとログを表示機器 (PASvisuプロジェクトを含むPMIなど) に表示する場合、プロジェクトにはOPCサーバ (PSS 4000 OPCサーバUA) が必要になります。

OPCサーバがインストールされているPSS 4000装置にも、診断サーバシステムセクションが自動的に表示されます。各OPCサーバには、独自の診断サーバがあります。

システム診断とプロセス診断に表示されるテキストは、プロジェクトがPAS4000でビルドされたときに使用可能になっていたものと、プロジェクトがダウンロードされた時に転送されたものです。

▶ PAS4000での表示

PAS4000には独自の診断サーバがあります。

システム診断を表示するテキストは、PAS4000に付属の診断テキストです。プロセス診断を表示するテキストは、プロジェクトがビルドされたときに使用可能になっていた診断テキストです。

**情報**

診断リストと診断ログは、現在のプロジェクトが診断サーバとプロジェクト内の全装置にダウンロードされていれば、正しく表示されます。

OPCサーバがインストールされているコンピュータが診断サーバとして使用されている場合は、このコンピュータにも現在のプロジェクトをダウンロードする必要があります (ダウンロード中にOPCサーバがSafetyNET pに接続されていれば、これはプロジェクトの直接ダウンロード中に自動的に行われます)。

また、診断サーバを含む装置のPSS 4000ファームウェアのバージョンを、プロジェクトのバージョンと同等かそれ以上にしておくことを推奨します。

テキストをPAS4000に表示する場合は、PAS4000で現在のプロジェクトを有効化する必要があります。必ずPAS4000の最新バージョンをお使いください。

長時間のオペレーションの後、プロジェクト装置で使えなくなったログエントリが、診断サーバのプロジェクト診断ログにまだ保存されている可能性があります。このような古いログエントリは、診断サーバのある装置がウォームリセットされるか、またはPAS4000がコールドスタートされると削除されます。

17 PSS 4000ネームスペース

PSS 4000 OPCサーバは、PSS 4000プロジェクトから一般的なプロジェクトデータおよび装置ベースのプロジェクトデータを収集し、PSS 4000ネームスペースに提示します。PSS 4000ネームスペースは、定義済み構造の特別なメモリ領域です。OPCクライアントは、この定義済み構造を介してPSS 4000ネームスペースのデータにアクセスできます。

多くの場合、ネームスペースはOPCクライアントのブラウザで表示できます。

最後にダウンロードされた全般的なプロジェクトデータと装置ベースのプロジェクトデータを、PSS 4000 OPCサーバはPSS 4000ネームスペースでいつでも使用できます。

PSS 4000ネームスペース内の変数へのアクセス

OPCクライアントは、PSS 4000プロジェクトのユーザプログラムで、ローカル変数 (VAR) とリソースグローバル変数 (VAR_GLOBAL) にアクセスできます。読み取りアクセスまたは書き込みアクセスが可能かどうかは、変数のデータ型ではなく、変数が格納されているリソースによって決まります。

変数	アクセス権
FSリソースのPI変数	読み取り
FSリソースのすべての変数、PI変数なし	
STリソースのPI変数	
STリソースのすべての変数、PI変数なし	読み取りと書き込み

PSS 4000ネームスペース構造の表現

以下のセクションでは、PSS 4000ネームスペースはツリー構造で表されています。このツリーは、対応する下位レベルを含めて、さまざまなレベルに分岐します。分岐ポイントはノードと呼ばれ、ラベルが付けられます。これらのノードは、データ構造 (= ファンクション) を表すために使用されます。ファンクションは最下位のレベルのみで見つかります。ネームスペース内でファンクションを一意に識別する完全な名前は、完全修飾名 (FQN) と呼ばれています。FQNは、階層的なノード名 (指定パス) とファンクション名で構成されます。表示形式は、OPCクライアントのブラウザにより異なることがあります。ただし、構造は同じです。

複雑になるため、以下のセクションでは、個々の領域のみの説明を順番に記載していきます。全体像を完全に把握できるように、各領域の説明を始める際に、その領域のノードにアクセスする場合の完全なパスを示します。パスを示す際に、ノード間を区切る文字としてピリオドを使用しています。

たとえば、State領域へのパスは次のように入力します。

```
pss4000.net.myDevice1.Device.State
```

したがって、State領域のDiagファンクションのFQNは次のようになります。

```
pss4000.net.myDevice1.Device.State.Diag
```

記号の定義

図とテーブルでは、マーカを山括弧で囲んで示しています。これらはそれぞれ現在の名前に置き換えられます。たとえば、<装置> はmyDevice1に置き換えます。

PSS 4000ネームスペースの主要領域の構造化

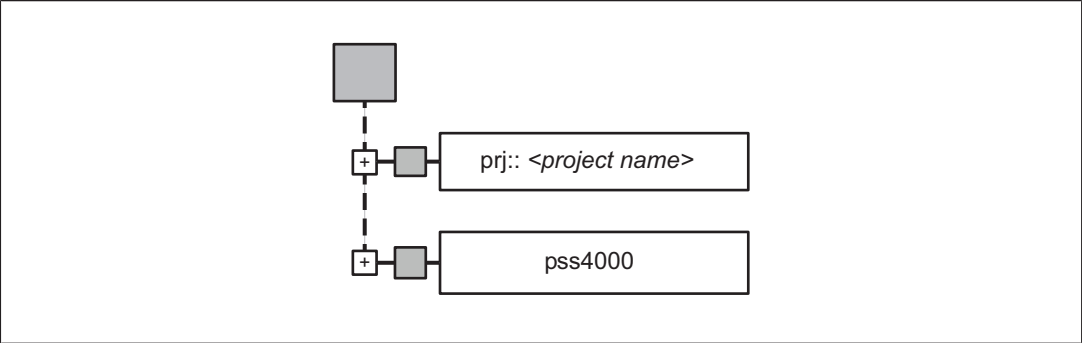


図: 主要領域内のノード名

ノード名	意味
<i>prj::<プロジェクト名></i>	PAS4000でコンフィグレーションされたプロジェクト名が含まれる、一般的なプロジェクトデータの領域 例: 「myProject」
pss4000	装置ベースのプロジェクトデータの領域

17.1 一般的なプロジェクトデータ

領域の構造

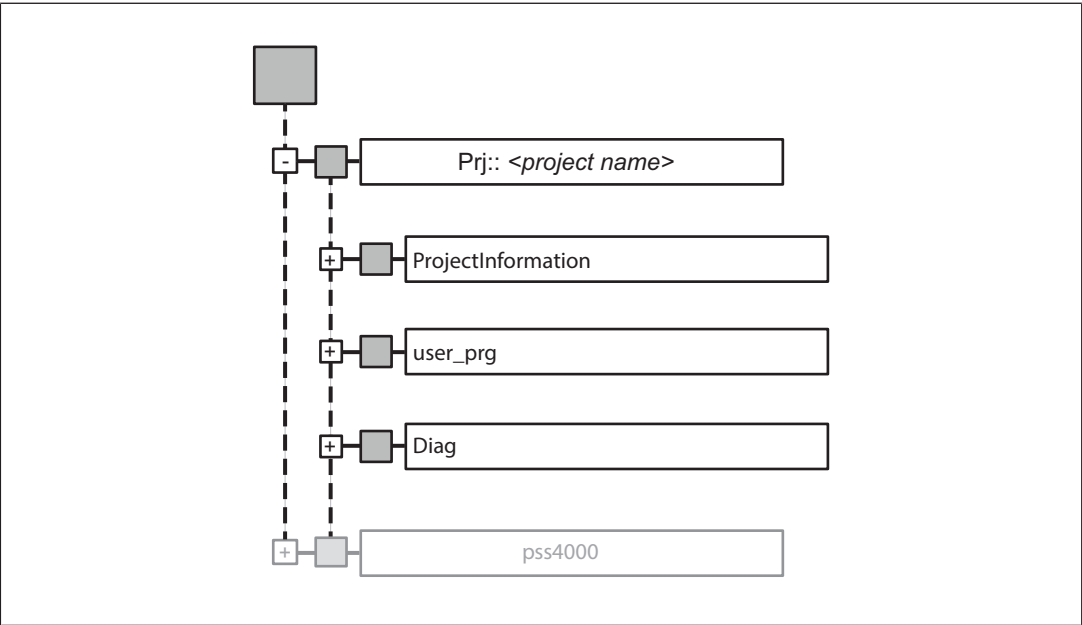


図: 「一般的なプロジェクトデータ」の構造

ノード名 (パスステートメント)		意味
prj::<project name>	ProjectInformation	プロジェクトベースの情報を含む領域
	(prj::<プロジェクト名>.ProjectInformation.)	
	user_prg	プログラム情報を含む領域 (POUの変数、コンポーネントブロックの変数、リソースグローバル変数など)
	(prj::<プロジェクト名>.user_prg.)	
	Diag	診断リストメッセージおよび診断ログエントリを含む領域
	(prj::<プロジェクト名>.Diag.)	

17.1.1 Projectinformation

指定パス: prj::<プロジェクト名>ProjectInformation

ノード番号	ノード名	意味
1	ChecksumFSProject	チェックサム「FSプロジェクト」 データ型: UInt32
	(prj::<プロジェクト名>.ProjectInformation.ChecksumFSProject)	
2	ChecksumFSProjectAppParams	チェックサム「FSプロジェクトのアプリケーションパラメータ」 データ型: UInt32
	(prj::<プロジェクト名>.ProjectInformation.ChecksumFSProjectAppParams)	
3	ChecksumFSProjectWithoutAppParams	チェックサム「アプリケーションパラメータがないFSプロジェクト」 データ型: UInt32
	(prj::<プロジェクト名>.ProjectInformation.ChecksumFSProjectWithoutAppParams)	



情報

チェックサム「FSプロジェクト」、「FSプロジェクトのアプリケーションパラメータ」、および「アプリケーションパラメータがないFSプロジェクト」は、PAS4000およびPASvisuサーバで16進数値として表示されます。サードパーティの視覚化システムでチェックサムを正しく表示できるようにするには、UInt32値に対して [16進表示] を選択してください。

17.1.2 user_prg

指定パス: prj::<プロジェクト名>.user_prg.

ノード番号	ノード名 (指定パス)	意味	
1	var_global	プロジェクトのリソースグローバル変数の領域 この領域には、ユーザが作成したすべてのリソースグローバル変数が含まれます。	
		OPCサーバでのリソースグローバル変数へのアクセス権:	
		FSリソースのPI変数	読み取り
		FSリソースのすべての変数、PI変数なし	
		STリソースのPI変数	
		STリソースのすべての変数、PI変数なし	読み取りと書き込み
	(prj::<プロジェクト名>.user_prog.var_global.)		
1.1	<グローバル変数名>	リソースグローバル変数の名前	
	prj::<プロジェクト名>.user_prog.var_global.<グローバル変数名>		

ノード番号	ノード名 (指定パス)	意味	
2	app	プロジェクトの変数のパス この領域には、プロジェクトのパスに加えて、ユーザが作成したVAR型の変数 (PI変数など) が含まれます。 次の点に注意してください: VAR_IN_OUT型、VAR_EXTERNAL型、およびVAR_TEMP型の変数はOPCサーバでは表示されません。	
		OPCサーバでの変数へのアクセス権:	
		FSリソースのPI変数	読み取り
		FSリソースのすべての変数、PI変数なし	
		STリソースのPI変数	
		STリソースのすべての変数、PI変数なし	読み取りと書き込み
		シングルエレメント変数のFQN: prj:: <プロジェクト名>.user_prog.app. <変数のパス>. <変数名> マルチエレメント変数のFQN: ▶ STRUCTタイプ変数のFQN: prj:: <プロジェクト名>.user_prog.app. <変数のパス>. <変数名>. <構造エレメント名> ▶ アクセスコンフィグレーションが「完全ARRAY」の場合のARRAYタイプ変数のFQN: prj:: <プロジェクト名>.user_prog.app. <配列のパス>. <配列名> ▶ アクセスコンフィグレーションが「インデックスベース」の場合のARRAYタイプ変数のFQN: prj:: <プロジェクト名>.user_prog.app. <配列のパス>. <配列名>. <配列要素のインデックス> 注: OPCサーバをPAS 4000でコンフィグレーションすると、「完全ARRAY」または「インデックスベース」アクセスコンフィグレーションが設定されます。	

17.1.3

Diag

指定パス: (prj::<プロジェクト名>.Diag)

診断リスト:

ノード番号	ノード名 (指定パス)	意味
1	List	診断リストメッセージを含む領域
	prj:: <プロジェクト名>.Diag.List.)	
1.1	Number	重大度レベルのメッセージ数
	(prj:: <プロジェクト名>.Diag.List.Number.)	
1.1.1	Error	重大度「エラー」のメッセージ数 データ型: UInt32
1.1.2	Warning	重大度「警告」のメッセージ数 データ型: UInt32
1.1.3	Information	重大度「ステータス情報」のメッセージ数 データ型: UInt32
1.2	Line	診断リストの行
	(prj:: <プロジェクト名>.Diag.List.Line.)	
1.2.1	<行番号>	行番号の情報 値の範囲: 1 ... 100
	(prj:: <プロジェクト名>.Diag.List.Line. <行番号>.)	
1.2.1.1	PriorityNo	優先度の数値 データ型: Byte
1.2.1.2	Priority	優先度の名前 データ型: String
1.2.1.3	説明	メッセージテキスト データ型: String
1.2.1.4	ScopeNo	範囲の数値 データ型: Byte
1.2.1.5	Scope	範囲の名前 データ型: String
1.2.1.6	Timestamp	タイムスタンプ データ型: DateTime
1.2.1.7	Did	診断ID データ型: String

ノード番号	ノード名 (指定パス)	意味
1.2.1.8	StateNo	診断メッセージのステータス番号
		ステータス
		1
		意味 メッセージは有効である
		データ型: Byte
1.2.1.9	 State	診断メッセージのステータス: データ型: String
1.2.1.10	ParameterList	パラメータリスト データ型: String
1.2.1.11	Location	ロケーション情報
	(prj::<プロジェクト名>.Diag.List.Line.<行番号>.Location.)	
1.2.1.11.1	Address	アドレス データ型: String
1.2.1.11.2	EquipmentId	装置ID データ型: String
1.2.1.11.3	Device	装置 データ型: String
1.2.1.11.4	説明	場所の説明 データ型: String
1.2.1.12	Action	処置アクション
	(prj::<プロジェクト名>.Diag.List.Line.<行番号>.Action.)	
1.2.1.12.1	<アクション番号>	処置番号 値の範囲: 1 ... 8
	(prj::<プロジェクト名>.Diag.List.Line.<行番号>.Action.<アクション番号>.)	
1.2.1.12.1.1	説明	アクションテキスト データ型: String
1.2.1.12.1.2	ScopeNo	範囲の数値 データ型: Byte
1.2.1.12.1.3	Scope	範囲の名前 データ型: String
1.2.1.12.1.4	ResponsibilityNo	責任者番号 データ型: Byte
1.2.1.12.1.5	Responsibility	責任者名 データ型: String

ノード番号	ノード名 (指定パス)	意味
1.2.1.12.1.6	Location (prj::<プロジェクト名>.Diag.List.Line.<行番号>.Action.<アクション番号>.Location.)	ロケーション情報
1.2.1.12.1.6.1	Address	アドレス データ型: String
1.2.1.12.1.6.2	EquipmentId	装置ID データ型: String
1.2.1.12.1.6.3	Device	装置 データ型: String
1.2.1.12.1.6.4	説明	場所の説明 データ型: String

診断ログ:

ノード番号	ノード名 (指定パス)	意味
2	Log (prj::<プロジェクト名>.Diag.Log.)	診断ログエントリを含む領域
2.1	Number	ログエントリの番号 データ型: UInt32
2.2	Line (prj::<プロジェクト名>.Diag.Log.Line.)	診断ログの行
2.2.1	<行番号> (prj::<プロジェクト名>.Diag.Log.Line.<行番号>.)	行番号の情報 値の範囲1〜「番号」(2.1を参照。最大4096)
2.2.1.1	PriorityNo	優先度の数値 データ型: Byte
2.2.1.2	Priority	優先度の名前 データ型: String
2.2.1.3	説明	メッセージテキスト データ型: String
2.2.1.4	ScopeNo	範囲の数値 データ型: Byte
2.2.1.5	Scope	範囲の名前 データ型: String

ノード番号	ノード名 (指定パス)	意味	
2.2.1.6	Timestamp	タイムスタンプ データ型: DateTime	
2.2.1.7	Did	診断ID データ型: String	
2.2.1.8	StateChangeNo	状態遷移のステータス番号	
		ステータス	意味
		0	メッセージがクリアされた
		1	メッセージが受信された
		101	ログエントリが実行された (運転エントリ、自己診断エントリ、デバッ グエントリなど)
		データ型: Byte	
2.2.1.9	StateChange	状態の変化	
		表示	意味
		cleared	メッセージがクリアされた
		arrived	メッセージが受信された
		logged	ログエントリが実行された (運転エントリ、自己診断エントリ、デバッ グエントリなど)
		データ型: String	
2.2.1.10	ParameterList	パラメータリスト データ型: String	
2.2.1.11	Location	ロケーション情報	
	(prj::<プロジェクト名>.Diag.Log.Line.<行番号>.Location.)		
2.2.1.11.1	Address	アドレス データ型: String	
2.2.1.11.2	EquipmentId	装置ID データ型: String	
2.2.1.11.3	Device	装置 データ型: String	
2.2.1.11.4	説明	場所の説明 データ型: String	

17.2 装置ベースのプロジェクトデータ

領域の構造

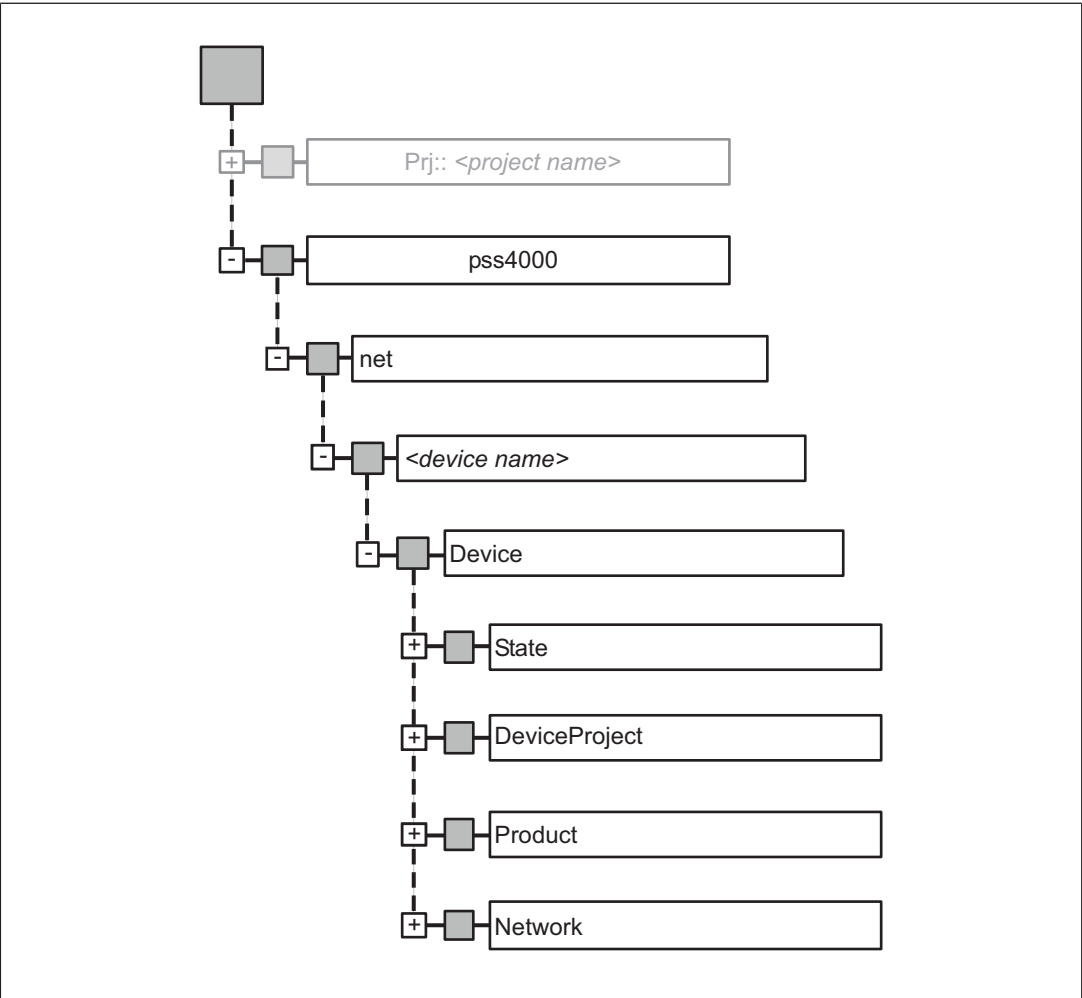


図: 「装置ベースのプロジェクトデータ」の構造

ノード名		意味	
pss4000.	net.	ネットワーク領域	
	<装置名>	装置の命名時にコンフィグレーションされた装置名を含む装置領域 (PSS 4000 専用／対応装置)	
	Device.	装置ベースの情報を含む領域	
	State	装置の診断ステータス情報を含む領域 (pss4000.net.<装置名>.Device.State.)	
	DeviceProject	装置プロジェクトの情報を含む領域 (pss4000.net.<装置名>.Device.DeviceProject.)	
	Product	製品情報を含む領域 (pss4000.net.<装置名>.Device.Product.)	
	Network	装置のネットワークパラメータを含む領域 (pss4000.net.<装置名>.Device.Network.)	
	FS-Resource	FSリソースのタスク情報を含む領域 (pss4000.net.<装置名>.Device.FS-Resource.)	
	ST-Resource	STリソースのタスク情報を含む領域 (pss4000.net.<装置名>.Device.ST-Resource.)	

17.2.1 State

指定パス: pss4000.net.<装置名>.Device.State.

ノード番号	ノード名	意味	
1	Diag	PSS 4000装置のシステムセクションのステータス	
		ステータス	意味
		0	装置診断リストと装置診断ログを準備中
		1	「作業者の意図的なアクション」の開始 (リセットボタンの機能)
		2	1つ以上のFSシステムセクションに重大なFSエラーがある (診断リストを参照)
		3	1つ以上のシステムセクションに重大度が「エラー」レベルのメッセージがある (診断テーブルを参照)
		4	装置に重大度が「警告」レベル以上のメッセージがある (診断リストを参照)
		5	装置に重大度が「エラー」または「警告」レベルのメッセージなし
		データ型: Byte	

ノード番号	ノード名	意味	
2	STRun	STリソースのステータス	
		ステータス	意味
		0	STリソースがまだ起動されていない
		1	オペレーティングステータス「STリソースがエラーでRUN状態」: -タスクがTERMINATED状態、または -タスクがSTOP状態 少なくとも1つのSTリソースタスクが未実行。 プロジェクトはライセンス登録済みです。
		2	オペレーティングステータス「STリソースがエラーなしでRUN状態」: STリソースのタスクはエラーなしで実行中。 プロジェクトはライセンス登録済みです。
		3	オペレーティングステータス「STリソースがエラーでRUN状態」: -タスクがTERMINATED状態、または -タスクがSTOP状態 少なくとも1つのSTリソースタスクが未実行。 プロジェクトのライセンス登録なし。
		4	オペレーティングステータス「STリソースがエラーなしでRUN状態」: STリソースのタスクはエラーなしで実行中。 プロジェクトのライセンス登録なし。
	データ型: Byte		

ノード番号	ノード名	意味	
3	FSRun	FSリソースのステータス	
		ステータス	意味
		0	FSリソースがまだ起動されていない
		1	オペレーティングステータス「FSリソースがエラーでRUN状態」: -タスクがTERMINATED状態、または -タスクがSTOP状態 少なくとも1つのFSリソースタスクが未実行。 プロジェクトのライセンス登録あり。
		2	オペレーティングステータス「FSリソースがエラーなしでRUN状態」: FSリソースのタスクはエラーなしで実行中。 プロジェクトはライセンス登録済み。
		3	オペレーティングステータス「FSリソースがエラーでRUN状態」: -タスクがTERMINATED状態、または -タスクがSTOP状態 少なくとも1つのFSリソースタスクが未実行。 プロジェクトのライセンス登録なし。
		4	オペレーティングステータス「FSリソースがエラーなしでRUN状態」: FSリソースのタスクはエラーなしで実行中。 プロジェクトはライセンス登録なし。
		データ型: Byte	
4	STForce	STリソースに対する強制、オンライン変更のステータス	
		ステータス	意味
		0	STリソースに対する強制は無効で、有効なオンライン変更はなし
		1	STリソースに対する強制が有効であるか、1つ以上のオンライン変更が有効 (またはその両方)
		データ型: Byte	

ノード番号	ノード名	意味	
5	FSForce	FSリソースに対する強制、オンライン変更のステータス	
		ステータス	意味
		0	FSリソースに対する強制は無効で、有効なオンライン変更はなし
		1	FSリソースに対する強制が有効であるか、1つ以上のオンライン変更が有効 (またはその両方)
		データ型: Byte	
6	STSafetyNETp	システムセクションST SafetyNET p RTFNのステータス	
		ステータス	意味
		0	オペレーティングステータス「ST SafetyNET p RTFNがエラーでSTOP状態: FSおよびSTの重大エラー」
		1	ST SafetyNET p RTFNが起動していない
		2	オペレーティングステータス「ST SafetyNET p RTFNがマイナーエラーでRUN状態」
		3	オペレーティングステータス「ST SafetyNET p RTFNがエラーなしでRUN状態」
		4	オペレーティングステータス「ST SafetyNET p RTFNがエラーでSTOP状態: FSおよびSTの重大エラー」
		データ型: Byte	

ノード番号	ノード名	意味	
7	FSSafetyNETp	システムセクションFS SafetyNET p RTFNのステータス	
		ステータス	意味
		0	オペレーティングステータス「FS SafetyNET p RTFNがエラーでSTOP状態: FSの重大エラー」
		1	FS SafetyNET p RTFNが起動していない
		2	オペレーティングステータス「FS SafetyNET p RTFNがマイナーエラーでRUN状態」
		3	オペレーティングステータス「FS SafetyNET p RTFNがエラーなしでRUN状態」
		4	オペレーティングステータス「FS SafetyNET p RTFNがエラーでSTOP状態: FSおよびSTの重大エラー」
		データ型: Byte	
8	Module bus	システムセクションFSおよびSTモジュールバスのステータス	
		ステータス	意味
		0	オペレーティングステータス「装置のすべてのFS出力が安全な状態」または、少なくとも1つのモジュールにアクセスできない (操作中にモジュールが削除された、設定されたハードウェアレジストリが実際の値と一致しないなど)
		1	モジュールはなし
		2	FSおよびSTモジュールバスが異常なく動作中
		3	オペレーティングステータス「FSモジュールバスがエラーでSTOP状態: FSの重大エラー」
		データ型: Byte	

ノード番号	ノード名	意味	
9	メディア	データメディアと装置識別のステータス	
		ステータス	意味
		0	ユーザが装置識別を有効化 ▶ SafetyNET p (PAS4000) でPSS 4000装置として認識、 または ▶ PROFINET (PROFINET用エンジニアリングツール) でPROFINET IO装置として認識
		1	リムーバブルディスクが見つからない または リムーバブルディスクが検出されない または リムーバブルディスクに不具合がある
		2	リムーバブルデータメディアの製品型式がPSS 4000 専用／対応装置と一致しない (PSSuシステム: ヘッドモジュールの製品型式) または リムーバブルデータメディアで使用可能な装置プロジェクトが存在しない
		3	PSS 4000 専用／対応装置とリムーバブルディスクメディアで、命名データと装置プロジェクトが一致しない
		4	PSS 4000 専用／対応装置とリムーバブルディスクメディアで、命名データと装置プロジェクトが一致
		5	予約済み
		6	▶ 「装置プロジェクトを装置に関連付ける」機能: 装置プロジェクトには特定の装置キーを装備する装置が必要ですが、この装置キーが装置に存在しません。 および／または ▶ 「装置プロジェクトをSDカードに関連付ける」機能: 装置プロジェクトはSDカードに関連付けられますが、このSDカードが装置に挿入されていません。
データ型: Byte			

ノード番号	ノード名	意味
10	SFProfibus	「SF」ステータスLED (PROFIBUS-DP) 「BF」ステータスLEDのLEDと対比して初めて評価が可能になります (装置／ヘッドモジュールの取扱説明書を参照)。
		ステータス
		0
		1
		データ型: Byte
11	BFProfibus	「BF」ステータスLED (PROFIBUS-DP) 「SF」ステータスLEDのLEDと対比して初めて評価が可能になります (装置／ヘッドモジュールの取扱説明書を参照)。
		ステータス
		0
		1
		2
		3
		データ型: Byte
12	SOPROFIsafe	「SO」ステータスLED (PROFIsafe) PROFIsafe仕様に適合したPROFIsafeのステータス
		ステータス
		0
		1
		3
		データ型: Byte

ノード番号	ノード名	意味	
13	MSEtherNetIP	ステータスLED「MS」(EtherNet/IP) EtherNet/IP仕様に適合したモジュールステータス	
		ステータス	意味
		0	LED「赤で点灯」 1つ以上のFSシステムセクションに重大なFSエラーがある
		1	LED「赤で速い点滅」 装置に重大度が「警告」レベルのメッセージが1つ以上あり、重大度が「エラー」レベルのメッセージはなく、無効なデータがスキャナにダウンロードされていない
		2	LED「緑で点灯」 装置に重大度が「エラー」または「警告」レベルのメッセージなし
		3	LED「消灯」 供給電圧なし または 装置が無効 または 装置のコンフィグレーションなし
	データ型: Byte		

ノード番号	ノード名	意味	
14	NSEtherNetIP	ステータスLED「NS」(EtherNet/IP) EtherNet/IP仕様に適合したネットワークステータス	
		ステータス	意味
		0	LED「赤で速い点滅」 接続がタイムアウト
		1	LED「緑で速い点滅」 ネットワーク接続なし
		2	LED「緑で点灯」 ネットワーク接続が使用可能で EtherNet/IP通信に異常なし
		3	LED「消灯」 データトラフィックなし または EtherNet/IPのコンフィグレーションなし または IPアドレスのコンフィグレーションなし
		データ型: Byte	
15	BFProfinet	ステータスLED「BF」(PROFINET IOバスのエラー) PROFINET IO仕様に準拠したバスステータス	
		ステータス	意味
		0	LED「赤で点灯」 PROFINET IOコントローラへの接続なし
		1	LED「赤で点滅」 PROFINET IOコントローラへの接続が使用可能、間違ったパラメータ設定が原因でデータ交換なし
		2	LED「緑で点灯」 PROFINET IO装置は「動作」状態、データ交換が可能
		3	LED「消灯」 PROFINET IO装置は無効化 (仮想PROFINET IOモジュールのコンフィグレーションなし)。
		データ型: Byte	

17.2.2 DeviceProject

指定パス: pss4000.net.<装置名>.Device.DeviceProject

ノード番号	ノード名	意味	
1	名前	装置 <装置名> が属しているPAS4000プロジェクトの名前 データ型: String	
2	ChecksumFS	装置 <装置名> の「FS装置プロジェクト」のチェックサム データ型: UInt32	
3	ChecksumFSProjectAppParams	装置 <装置名> の「FS装置プロジェクトのアプリケーションパラメータ」のチェックサム データ型: UInt32	
4	ChecksumFSProjectWithoutApp-Params	装置 <装置名> の「アプリケーションパラメータがないFS装置プロジェクト」のチェックサム データ型: UInt32	
5	BuildTime	PAS4000プロジェクトのビルドにかかった時間 データ型: DateTime	
6	Version	装置 <装置名> が属しているPAS4000プロジェクトのバージョン データ型: String	
7	LicenceState	プロジェクトのライセンスステータスのステータス情報	
		ステータス	意味
		0	ライセンス登録なし、または有効なプロジェクトなし
		1	ライセンス登録済み
		データ型: Byte	

17.2.3 Product

指定パス: pss4000.net.<装置名>.Device.Product.

ノード番号	ノード名	意味
1	SerialNo	装置 <装置名> のシリアル番号、 PSSuシステム: ヘッドモジュールのシリアル番号 データ型: String
2	OrderNo	装置 <装置名> の注文番号、 PSSuシステム: ヘッドモジュールの注文番号 データ型: String
3	ProductType	装置 <装置名> の製品型式、 PSSuシステム: ヘッドモジュールの名前 データ型: String
4	FirmwareVersion	装置 <装置名> のファームウェアバージョン、 PSSuシステム: ヘッドモジュールのファームウェアバージョン データ型: String
5	FirmwareRevision	装置 <装置名> のファームウェアの改訂番号、 PSSuシステム: ヘッドモジュールのファームウェアの改訂番号 データ型: String
6	FirmwareUpdate	装置 <装置名> のファームウェアの最終更新日時、 PSSuシステム: ヘッドモジュールのファームウェアの最終更新日時 データ型: DateTime

ノード番号	ノード名	意味	
7	HardwareVersion	装置 <装置名> のハードウェアバージョン、 表示形式: <MajorHardwareVersion>.<MinorHardwareVersion> 例: 3.1 PSSuシステム: ヘッドモジュールのハードウェアバージョン データ型: String	
8	MajorHardwareVersion	装置のメジャーハードウェアバージョン <装置名>、 PSSuシステム: ヘッドモジュールのメジャーハードウェアバージョン データ型: Byte	
9	MinorHardwareVersion	装置のマイナーハードウェアバージョン <装置名>、 PSSuシステム: ヘッドモジュールのマイナーハードウェアバージョン データ型: Byte	
10	PilzSecSDcardSerialNumber	装置のセキュリティSDカード (PASkey SDカードなど) のシリアル番号 <装置名> データ型: String	
11	DeviceKey	装置キー 機能「装置プロジェクトを装置に関連付ける」について、 装置に装置キーが書き込まれている場合は、装置キーの一部が表示されます。この項目の詳細については、オンラインヘルプを参照してください。 データ型: String	
12	DeviceKeyRestrictsProjects	装置で実行するプロジェクトに一致するプロジェクトキーが必要かどうかについての情報。機能「装置プロジェクトを装置に関連付ける」の詳細については、オンラインヘルプを参照してください。	
		表示	意味
		FALSE	プロジェクトにはプロジェクトキーがありません。
		TRUE	プロジェクトには一致するプロジェクトキーが必要です。
		データ型: BOOL値	

17.2.4 Network

指定パス: pss4000.net.<装置名>.Device.Network.

ノード番号	ノード名	意味	
1	IPAddressV4	装置 <装置名> のIPアドレス データ型: String	
2	SubnetMaskV4	サブネットマスク データ型: String	
3	GatewayV4	標準ゲートウェイのIPアドレス データ型: String	
4	DHCP	装置 <装置名> のDHCP設定:	
		表示	意味
		FALSE	DHCPは有効化されていない
		TRUE	DHCPは有効化されている
		データ型: BOOL値	
5	DNS1V4	優先DNSサーバのアドレス データ型: String	
6	DNS2V4	代替DNSサーバのアドレス データ型: String	
7	MACAddress	装置 <装置名> のMACアドレス データ型: String	
8	SNTP	装置 <装置名> のSNTP設定	
		表示	意味
		FALSE	装置はプロジェクトのSNTPサーバとしてコンフィグレーションされていない
		TRUE	装置はプロジェクト内でSNTPサーバとして機能している
		データ型: BOOL値	
9	SNTPV4	装置が時刻を取得するSNTPサーバのIPアドレス データ型: String	
10	RealTimeClock	装置の現在の時刻 データ型: DateTime	

17.2.5 FS resource

指定パス: pss4000.net.<装置名>.Device.FS-Resource.

ノード番号	ノード名 (指定パス)	意味
1	UserProgramMemoryUsed	装置 <装置名> のユーザプログラムに使用されるメモリ (有効なオンライン変更を含む) データ型: UInt32
	(pss4000.net.<装置名>.Device.FS-Resource. UserProgramMemoryUsed)	
2	UserProgramMemorySize	装置<装置名>のユーザプログラムのメモリのサイズ データ型: UInt32
	(pss4000.net.<装置名>.Device.FS-Resource. UserProgramMemorySize)	
3	<タスク名>	装置 <装置名> のFSリソースにあるタスクの名前
	pss4000.net.<装置名>.Device.FS-Resource.<タスク名>.)	
3.1	State	タスクの状態
		ステータス
		意味
		1
		2
		3
		データ型: UInt32
	(pss4000.net.<装置名>.Device.FS-Resource.<タスク名>.State)	
3.2	ProcessingTime	タスクの実行時間 t_{ProcTask} 実行時間は、周期性タスクの場合にのみ使用できる (「応答時間 [457]」を参照)。
	(pss4000.net.<装置名>.Device.FS-Resource.<タスク名>. ProcessingTime.)	

ノード番号	ノード名 (指定パス)	意味
3.2.1	Minimum	最小の実行時間 表示される時間は、タスクの開始以来、検出された最も短い実行時間です。 データ型: UInt32
3.2.2	Maximum	最大の実行時間 表示される時間は、タスクの開始以来、検出された最も長い実行時間です。 データ型: UInt32
3.2.3	Current	現在のタスク実行時間 データ型: UInt32

17.2.6 ST resource

指定パス: pss4000.net.<装置名>.Device.ST-Resource.

ノード番号	ノード名 (指定パス)	意味
1	UserProgramMemoryUsed	装置 <装置名> のユーザプログラムに使用されるメモリ (有効なオンライン変更を含む) データ型: UInt32
	(pss4000.net.<装置名>.Device.ST-Resource. UserProgramMemoryUsed)	
2	UserProgramMemorySize	装置<装置名>のユーザプログラムのメモリのサイズ データ型: UInt32
	(pss4000.net.<装置名>.Device.ST-Resource. UserProgramMemorySize)	
3	<タスク名>	装置のSTリソースにあるタスクの名前 <装置名> pss4000.net.<装置名>.Device.ST-Resource.<タスク名>.)
3.1	State	タスクの状態
		ステータス
		意味
		1
		2
		3
	データ型: UInt32	
3.2	ProcessingTime	タスクの実行時間 $t_{ProcTask}$ 実行時間は、周期性タスクの場合にのみ使用できる (「応答時間 [457]」を参照)。
	(pss4000.net.<装置名>.Device.ST-Resource.<タスク名>. ProcessingTime.)	
3.2.1	Minimum	最小の実行時間 表示される時間は、タスクの開始以来、検出された最も 短い実行時間。 データ型: UInt32

ノード番号	ノード名 (指定パス)	意味
3.2.2	Maximum	最大の実行時間 表示される時間は、タスクの開始以来、検出された最も長い実行時間。 データ型: UInt32
3.2.3	Current	現在のタスク実行時間 データ型: UInt32

17.3 PSS 4000ネームスペースのデータ型

PSS 4000ネームスペースでは、以下で説明するデータ型が表示されることがあります。PSS 4000ネームスペースのデータ型は、該当するクライアント／サーバで使用可能なデータ型に割り付ける必要があります。ピルツのクライアントとサーバには、適切な割り付けリストを使用できます。該当するピルツのクライアントとサーバの資料を参照してください。

データ型	意味
BOOL値	ビット値／BOOL値 データ幅: 1ビット 値の範囲: TRUE/FALSE
Byte	正の整数 データ幅: 8ビット (1バイト) 値の範囲: 0～255
Double	倍精度の浮動小数点数 (IEEE754に適合) データ幅: 64ビット (8バイト)
Float	単精度の浮動小数点数 (IEEE754に適合) データ幅: 32ビット (4バイト)
Int16	整数 データ幅: 16ビット (1ワード) 値の範囲: -32768～32767
Int32	整数 データ幅: 32ビット (2ワード) 値の範囲: - 2147483648～2147483647
Int64	整数 データ幅: 64ビット (4ワード) 値の範囲: - 9223372036854775808～9223372036854775807
LocalizedText	言語識別子付きのテキスト コーディングについては、「String」を参照
SByte	整数 データ幅: 8ビット (1バイト) 値の範囲: -128～127
String	Unicode形式の文字列
UInt16	正の整数 データ幅: 16ビット (1ワード) 値の範囲: 0～65535
UInt32	正の整数 データ幅: 32ビット (2ワード) 値の範囲: 0～4294967295
UInt64	正の整数 データ幅: 64ビット (4ワード) 値の範囲: 0～18446744073709551615

データ型	意味
DateTime	正の整数 データ幅: 64ビット (4ワード) DateTimeに表示される値は、1601-01-01 00:00:00から経過した100ナノ秒間隔の回数として解釈されます。

18 付録

18.1 さまざまなサービスのポート番号に対する工場出荷時の既定設定

各種サービスの既定のポート番号を以下に示します。

サービス		既定の設定	意味
SafetyNET p		40000	SafetyNET pプロトコル
DNS		53	Domain Name Serviceプロトコル
OPC via DCOM	接続のセットアップ (サーバ)	135	Distributed Component Object Modelプロトコル
	通信 (サーバ)	動的	
DHCP		67、68	Dynamic Host Configuration Protocol
Modbus/TCP		502	Modbus over TCP
SNTP		123	Simple Network Time Protocol

18.2 Ethernetパラメータの出荷時の既定の設定

PSS 4000専用装置の工場出荷時またはオリジナルリセット後には、Auto IPメカニズムが有効になっています。Ethernetのパラメータは次のように設定されます。

パラメータ	既定の設定	意味
IPアドレス	Auto-IPメカニズムが有効: 169.254.1.0～169.254.254.255	Auto IPメカニズムによって、IPアドレスがランダムに割り付けられます。
サブネットマスク	255.255.0.0	Auto IPメカニズムによって、指定のサブネットマスクが割り付けられます。
スタンダードゲートウェイのIPアドレス	無効化 (0.0.0.0)	SafetyNET pがゲートウェイ経由でサブネットワークに分割され、通信の相手から見て当該製品がゲートウェイの後ろにある場合、オプションとして設定可能です。
常用DNSサーバのアドレス	無効化 (0.0.0.0)	DNSを使用する場合、オプションとして設定可能です。
代用DNSサーバのアドレス		

Auto IP機構

コンフィグレーションが不要で、169.254.0.0/16 のアドレス範囲から空いているIPアドレスを自動的に割り付ける技術。

BDI

基本診断項目 (Basic Diagnostic Item) の略語。

FB

ファンクションブロック (Function Block) の略語。

FQN

完全修飾名の略語

FS

安全 (FailSafe) の略語。

FSプロジェクト

「FSプロジェクト」には、ユーザプログラムのプロジェクトのすべての安全関連ファンクション、ハードウェアコンフィグレーション、リソースの割り付け、I/Oマッピングが含まれています。

FSリソース

安全関連制御コードの実行ユニット。

FW

ファームウェア (FirmWare) の略語。

HW

ハードウェア (HardWare) の略語。

I/Oマッピング

モジュールバスからI/Oデータ、あるいは外部通信または他のPI変数からI/Oデータに対するPI変数のマッピング。

IL

インストラクションリスト (Instruction List) の略語。

I-PI変数

モジュールバスからIデータ、あるいは外部通信またはO-PI変数からIデータにマッピング可能な、プロセスイメージ内の変数。

O-PI変数

モジュールバスからOデータ、あるいは外部通信またはI-PI変数からOデータにマッピング可能な、プロセスイメージ内の変数。

PI変数

モジュールバスからI/Oデータ、あるいは外部通信または他のPI変数からI/Oデータにマッピング可能な、プロセスイメージ内の変数。

POU

プログラム構成要素 (Program Organisation Unit) の略語。

PRG

プログラム (PRoGram) の略語。

PSS 4000プロジェクト

システム (設備、機械) を制御および監視するためにPAS4000で作成されたプロジェクト。プロジェクト=ユーザプログラム+ハードウェアコンフィグレーション+ネットワークコンフィグレーション。

SafetyNET pインフラ

「イーサネットインフラ」を参照

ST

一般 (STandard) の略語。

STプロジェクト

STプロジェクトには、FSプロジェクトに属していないすべてのプロジェクトファンクションが含まれています。

STリソース

非安全関連制御コード用の実行ユニット。

SW

ソフトウェア (SoftWare) の略語。

イーサネットインフラ

イーサネットネットワークにおけるデータ転送に必要なすべてのハードウェア (ケーブル、プラグイン接続、アダプタ、スイッチ、ルータ、ハブなど) とソフトウェア (ネットワークサービスとそのコンフィグレーションなど)。

イネーブルの原理

STセクションがFSセクションへの書き込みアクセスを持つことを可能 (イネーブル) にするコンフィグレーション。FS出力は、STセクション経由で切り替えることができます。

インストラクションリスト (IL)

テキストベースの機械指向プログラミング言語。すべての命令はテキスト形式で示されます。

オートセンシング

データ速度と全二重モードの自動検出

クロスオーバーケーブル

イーサネットコンポーネントとPLCまたはネットワークコンポーネント (ハブ、スイッチ) などの端末機器を配線するクロスケーブル。

シールされた

プロジェクトファンクションのプロパティ (プログラム構成要素、コンポーネントブロックなど)。プロジェクトファンクションはシールされ、必要に応じて認証されており、変更および表示することはできません。

スイッチ

複数のイーサネット端末をスター型接続するイーサネットインフラのハードウェア。1つのポートで受信されたデータパケットは分析され、サブスクライバが登録されているポートのみに転送されます。ブロードキャストメッセージが転送されます。

ダウンロード

装置にデータを転送すること (PCから装置への転送など)。

タスク

関連するIEC 61131プログラムや基本ブロック (「IEC 61131」を参照) のグループを定期的に実行するか、トリガして実行するための実行制御ファンクション。

チェックサム

プロジェクトまたはプログラム構成要素 (POU) など、オブジェクトを経由して形成される和。オブジェクトが変更されているかどうかを検出するために使用できます。

データの一貫性

定義された期間中、特定の時刻に記録されたデータと最新であり関連しているとみなされるデータをまとめておくこと。

テストパルス

テストパルスは入力に結び付けられて、その入力で断線、24 V DCまたは0 Vへの短絡、あるいは2つの入力間の短絡などの配線エラーを検出します。

デバイスエディタ

装置の設定を定義したり、装置から情報を呼び出すためのソフトウェアツール。

デバッグ

プログラミング中または試運転中のトラブルシューティングと不具合の回復。

ハードウェア出力

物理装置の出力。

ハードウェア入力

物理装置の入力。

バイトモジュール

データがプロセスイメージ内の異なるデータ幅 (ビット、バイト、またはワード) を必要とする電子モジュール。例: カウンタモジュール。

パッチケーブル

最大長が5 mの1:1の配線ケーブル。主に筐体内のイーサネットコンポーネント (制御盤など) の接続に使用されます。

ハブ

複数のイーサネット端末をスター型接続するイーサネットインフラのハードウェア。1つのポートで受信されたデータパケットは、その他すべてのポートに転送されます。

ビットモジュール

プロセスイメージ内の1ビットを必要とするデータを使用する電子／コンパクトモジュール。例: デジタル入出力モジュール

ビルド

各装置 (装置プロジェクト) のコードは、プロジェクトからビルドされます。

ファームウェア

メーカが作成した、装置で稼働する、装置固有の機能を実行するソフトウェア。

ファームウェアアップデート

装置のファームウェアを更新すること。

ファンクションブロック (FB)

IEC 61131-3: 以下で構成されるプログラマブルコントローラのプログラミング言語ファンクションを指します。1) 入力、出力、および内部変数に区分化されたデータ構造の定義、および2) ファンクションブロックタイプのインスタンスが呼び出されたとき、データ構造のファンクションで実行される一連の命令。

プログラム (PRG)

IEC 61131-1: 「機械の制御またはPLCシステムによる処理を行うための信号処理に必要なすべてのプログラミング言語要素および構文の論理的な集まり。」

プログラム構成要素 (POU)

IEC 61131-3に適合する機能、ファンクションブロック、またはプログラム。

ブロック

マルチブロックとPOUを指す包括的用語。

リアルタイムシステム

計算結果の妥当性が、論理的な正確さだけではなく、結果が提供される時間にも依存するシステム。早すぎるかまたは遅すぎる応答は、正しくない応答です。

リソース

FSリソース、STリソースなどのコードを実行するユニット

リソースの割り付け

プログラム型プログラム構成要素または基本ブロックをリソースに割り付けること。

ルータ

さまざまなネットワークに接続するためのイーサネットインフラストラクチャのハードウェア。ネットID (IPアドレスのセクション) を基にして、ルータが異なるネットワークに送信するデータパケットを決定します。

安全 (FS)

安全関連のフェイルセーフ。安全 (フェイルセーフ) システムにより、エラー発生時に安全な状態に切り替わります。

安全な状態

安全回路から電源を取り外した状態。

一貫性の範囲

データソースに含まれている一貫したデータの量 (「FSモジュールバスのFS-Iデータ」などの一貫性の範囲)。

一般 (ST)

安全 (フェイルセーフ) の対義語で、安全関連ではない一般制御を表します。

一般バスシステム

ピルツのSafetyBUS pやSafetyNETpと相対する、DeviceNetやPROFIBUSなどの非安全関連バスシステム (「フィールドバス」も参照してください)。

完全修飾名 (FQN)

ネームスペースでファンクションを一意に識別する完全な名前

基本診断項目 (BDI)

診断: 診断目的で監視される処理状態を定義するユーザプログラムの一部。基本診断項目には、1つ以上の診断メッセージおよび診断ログエントリの情報とテキストも含まれます。

自動交差

ツイストペアインタフェースにおける送受信回線の自動交差。この機能をサポートする装置は、クロスケーブル (クロスオーバーケーブル) ではなく、1:1の配線ケーブル (パッチケーブル) で接続できます。

処理データの品質

処理データが「処理値」または「代入値」のどちらに関連しているのかについての情報。

処理値

処理データの実行値。代入値の対義語です。

診断リスト

診断: プロジェクトの現在の診断メッセージのリスト。

診断ログ

診断: プロジェクトの診断メッセージを受信した時間と消去した時間を示すログ。運転エントリ、自己診断エントリ、およびデバッグエントリも含まれます。

診断項目

診断: 基本診断項目と要約診断項目を指す一般用語。

装置プロジェクト

装置にダウンロードされるプロジェクトの一部。

代入値

エラーにより処理データが求められない場合は、代入値で置き換えられます。

適時性

定義された時間ウィンドウ内でイベントが発生すること。前後に発生するイベントは無効です。

認証

製品が、該当する規則および規格に適合していることを確認するための認証機関によるテスト手順。認証書の発行によって適合していることが証明されます。

認証機関

指令2006/42/ECの14条に従った認証機関。または、製品が、該当する規則および規格に適合していることを検査および承認する認定協会 (通常、政府機関)。

変数の強制

変数の値は永久的に割り付けられ、ユーザプログラムによって変更することはできません。

変数の制御

変数の値は、指定された値に一度設定されます。その後は、ユーザプログラムによって変更することができます。

変数リスト

制御、強制、または監視が実行される変数が記載されたリスト。

有資格者

トレーニング、経験、および現在の専門的な活動を通じて、安全技術の一般的な規格およびガイドラインに従ってデバイス、システム、設備、および機械をテスト、評価、操作するために必要な知識を有し、それを行う認可を得ている人。

