

DSHEng5 装置／ホスト通信エンジン (GEM+GEM300)

ソフトウェア・パッケージ

ユーザーズ・ガイド

(.Net C#. VB.Net クラス・ライブラリ標準)

2019年12月 (改訂-1)

株式会社データマップ

文書番号 DSHEng5-19-30300-01

[取り扱い注意]

この資料ならびにソフトウェアの一部または全部を無断で使用、複製することはできません。
本説明書に記述されている内容は予告なしで変更される可能性があります。

Windows は米国 Microsoft Corporation の登録商標です。

ユーザーが本ソフトウェアの使用によって生じた遺失履歴、(株) データマップの予見の有無を問わず発生した特別損害、付随的損害、間接損害およびその他の拡大損害に対して責任を負いません。

【改訂履歴】

番号	改訂日付	項目	概略
1.	2019. 6. 28	初版	
2.	2019. 12. 16	訂正と記述追加	誤字等の訂正 説明の追加など。
3.			
4.			

目 次

1. はじめに	1
1. 1 関連資料	3
1. 1. 1 DSHEng5 通信エンジン関連資料	3
1. 1. 2 SEMI スタンダード	5
1. 2 サポート範囲	6
2. DSHEng5 ソフトウェア構成	7
3. 装置管理情報	9
3. 1 装置定数 (EC, SV, DVVAL) 情報とクラス	11
3. 1. 1 装置定数 - class_EC クラス	13
3. 1. 2 装置状態変数 - classSV クラス	13
3. 1. 3 装置データバリュー変数 - classDV クラス	13
3. 2 収集イベント (CE)、レポート (REPORT) 情報	14
3. 2. 1 CE (収集イベント) - class_CE クラス	16
3. 2. 2 レポート - class_Report クラス	17
3. 3 アラーム情報	18
3. 4 スプール情報	20
3. 5 トレース情報	21
3. 6 プロセスプログラム (PP) 情報	23
3. 7 フォーマット付きプロセスプログラム (FP P) 情報	24
3. 8 レシピ (R C P) 情報	25
3. 9 キャリア (C A R) 情報	26
3. 10 基板 (SUBSTRATE) 情報	28
3. 11 プロセスジョブ (P R J) 情報	30
3. 12 コントロールジョブ (C J) 情報	32
4. DSHEng5 構成プログラムと機能概略	34
4. 1 プログラムモジュールの構成	34
4. 2 アプリケーションプログラム	35
4. 2. 1 アプリケーションプログラムの処理	35
4. 2. 1. 1 DSHEng5 の初期化処理	36
4. 2. 1. 2 装置管理情報 (変数など) アクセス処理	37
4. 2. 1. 3 対装置通信の開始/通信停止要求	38
4. 2. 1. 4 収集イベント通知処理	39
4. 2. 1. 5 アラーム通知処理	40
4. 2. 1. 6 APP が処理する通信メッセージの登録と処理	41
4. 2. 1. 7 1 次メッセージの送信	43
4. 2. 1. 8 DSHEng5 の終了処理	44
4. 2. 2 ユーザによる受信 1 次メッセージの処理とクラス	45
4. 2. 2. 1 ユーザが処理する受信 1 次メッセージの登録	46
4. 2. 2. 2 メッセージのエンコード / デコードクラス	48
4. 2. 2. 3 1 次メッセージ受信処理の流れ	51
4. 2. 3 DSHEng5 通信エンジンプログラム	53
4. 3 DSHEng5 通信エンジン構成プログラムの機能	54
4. 3. 1 DSHEng5 GEM 通信エンジン	55
4. 3. 1. 1 起動時の処理	55
4. 3. 1. 2 通常処理	58
4. 3. 1. 3 終了処理	59
4. 3. 2 HSMS 通信ドライバー (DSHDR2.DLL)	60

5. 個別機能	61
5. 1 状態管理機能	61
5. 1. 1 通信状態モデル	61
[通信状態制御用クラス一覧]	63
5. 1. 2 コントロール状態の管理	64
[コントロール関連メッセージ送信用クラス一覧]	66
5. 2 装置変数管理機能と装置への変数要求機能	67
[装置変数関連メッセージ送信用クラス一覧]	68
5. 3 収集イベント通知	69
[収集イベント関連メッセージ送信用クラス一覧]	71
5. 4 アラーム通知	72
5. 4. 1 アラーム状態モデル	72
5. 4. 2 アラーム処理と流れ	72
[アラーム関連メッセージ送信用クラス一覧]	73
5. 5 スプール機能	74
5. 5. 1 スプール状態モデル	74
5. 5. 2 スプーリング処理と流れ	76
[スプール関連メッセージ送信用クラス一覧]	77
5. 6 トレースデータ収集機能	78
[トレース関連メッセージ送信用クラス一覧]	79
5. 7 プロセスプログラム、レシピ管理機能	80
5. 7. 1 プロセスプログラム (PP) 管理機能	81
[プロセスプログラム関連メッセージ送信用クラス一覧]	82
5. 7. 2 書式付プロセスプログラム (FPP) 管理機能	83
[書式付プロセスプログラム関連メッセージ送信用クラス一覧]	84
5. 7. 3 レシピ (RCPID) 管理機能	85
[レシピ関連メッセージ送信用クラス一覧]	86
5. 8 キャリア管理と搬送関連機能	87
5. 8. 1 ロードポート搬送状態	87
[ロードポート関連メッセージ送信用クラス一覧]	92
5. 8. 2 キャリア状態モデル	93
[キャリアアクション関連メッセージ送信用クラス一覧]	98
5. 8. 3 アクセスモード管理	99
[アクセスモード関連メッセージ送信用クラス一覧]	100
5. 8. 4 ロード予約状態管理	101
5. 8. 5 ロードポート/キャリア関連状態管理	103
5. 9 プロセスジョブ管理機能	105
5. 9. 1 プロセスジョブ状態モデル	105
5. 9. 2 プロセスジョブの管理と処理の流れ	108
[プロセスジョブ関連メッセージ送信用クラス一覧]	109
5. 10 コントロールジョブ管理機能	110
5. 10. 1 コントロールジョブ状態モデル	110
5. 10. 2 コントロールジョブの管理と処理の流れ	113
[コントロールジョブ関連メッセージ送信用クラス一覧]	114
5. 11 端末サービス機能	115
[端末サービス関連メッセージ送信用クラス一覧]	115
5. 12 装置に対するリモートコマンドメッセージ送信処理	116
[リモートコマンド関連メッセージ送信用クラス一覧]	116
付録-A DSHeng5 - SECS-II 処理MSG一覧表	117



付録-B	DSHEng5 装置起動ファイルコマンド	119
付録-C	バックアップ対象情報と更新.....	120

図表目次

（表 1.1.1.B DSHEng5 .Net クラスライブラリ一覧表）	1
図 1-1 システムにおける DSHEng5 の位置	1
図 1-2 DSHEng5 の SEMI スタンダードサポート対象機能	2
表 1.1.1 DSHEng5 通信制御エンジン関連文書一覧表	エラー! ブックマークが定義されていません。
表 1.1.1.A DSHEng5 通信制御エンジン関数説明書一覧表	エラー! ブックマークが定義されていません。
表 1.1.1.B DSHEng5 .Net クラスライブラリ一覧表	エラー! ブックマークが定義されていません。
表 1.1.2 SEMI スタンダード関連資料一覧表	5
図 2-1 基本的なソフトウェア構成	7
表 3 装置管理情報一覧	9
表 3-1 変数と SECS-II メッセージの関連	10
図 3.1 変数情報の関連図	11
図 3.2 収集イベント情報関連図	14
図 3.3 アラーム情報関連図	18
図 3.4 スプール情報関連図	20
図 3.5 トレース情報関連図	21
図 3.6 プロセスプログラム情報関連図	23
図 3.7 フォーマット付きプロセスプログラム情報関連図	24
図 3.8 レシピ情報関連図	25
図 3.9 キャリア情報関連図	26
図 3.10 基板情報関連図	28
図 3.11 プロセスジョブ情報関連図	30
図 3.12 コントロールジョブ情報関連図	32
図 4 プログラムモジュール構成図	34
図 4.2.1.2 装置管理情報アクセス処理関連図	37
表 4.2.1.2 アクセス対象装置管理情報	37
図 4.2.1.3 通信状態モデル (Communication State Model)	38
図 4.2.1.7 APP による 1 次メッセージ送信処理関連図	43
図 4.2.2.1 APP への配信 1 次メッセージのソースファイルへの登録例	46
表 4.2.2.1 装置側デフォルト登録メッセージ一覧	47
図 4.2.2.3 APP の受信 1 次メッセージ処理	51
図 4.2.3 DSHEng5 の機能図	53
図 4.3 DSHEng5 プログラムの構成と位置	54
図 4.3.1.2-1 DSHEng5 の通常処理の関連図	58
図 4.3.1.2-4 DSHEng5 の 1 次メッセージの APP への配信関連図	58
図 4.3.1.2-5 DSHEng5 の APP からの 1 次メッセージ送信要求処理の関連図	58
図 4.3.1.2-6 DSHEng5 受信 1 次メッセージの内部自動処理の関連図	59
図 5.1.1 通信状態遷移図 (Communication State Model)	61
表 5.1.2 通信状態遷移定義表	62
図 5.1.1-1 通信状態管理に関する処理の流れ	63
表 5.1.1 通信状態制御用クラス	63
図 5.1.2 コントロール状態遷移図	64
表 5.1.2 コントロール状態遷移表	65
図 5.1.2-1 コントロール状態管理処理の流れ	66
表 5.1.2 コントロール関連メッセージ	66
図 5.2-1 変数アクセス操作	67
図 5.2-2 変数リミット値操作	67
表 5.2 装置変数関連メッセージ	68

図 5.3-1 CEID, RPTID, VID の関係	69
図 5.3-2 収集イベント処理の流れ	69
表 5.3 収集イベント関連メッセージ	71
図 5.4.1 アラーム ALIDn についての状態図	72
表 5.4 アラーム関連メッセージ送信用クラス	73
図 5.5.1 スプーリング状態遷移図	74
表 5.5.1 スプーリング状態遷移定義表	74
図 5.5.2 スプーリング処理の流れ	76
表 5.5 スプーリング関連メッセージ送信用クラス	77
図 5.6 トレース処理の流れ	78
表 5.6 トレース関連メッセージ送信用クラス	79
図 5.7 プロセスプログラム (レシピ) のタイプ	80
図 5.7.1-1 ホストからの S7F3 受信と情報設定	81
図 5.7.1-2 装置から S7F3 を送信	81
図 5.7.1-3 APP による PP 情報アクセス	82
表 5.7.1 プロセスプログラム関連メッセージ送信用クラス	82
図 5.7.2-1 ホストからの S7F23 受信と情報設定	83
図 5.7.2-2 装置から S7F23 を送信	83
図 5.7.2-3 APP による FPP 情報アクセス	84
表 5.7.2 書式付プロセスプログラム関連メッセージ送信用クラス	84
図 5.7.3-1 ホストからの S15F13 受信と情報設定	85
図 5.7.3-2 装置から S15F13 を送信	85
図 5.7.2-3 APP による RCP 情報アクセス	86
表 5.7.3 レシピ関連メッセージ送信用クラス	86
図 5.8.1 ロードポート搬送状態遷移図	87
表 5.8.1 ロードポート搬送状態遷移定義表	88
図 5.8.1-1 ホータ/上位プロセス指示によるポートサービス状態管理処理の流れ	92
図 5.8.1-2 装置によるポートサービス状態管理処理の流れ	92
表 5.8.1 ロードポート関連メッセージ送信用クラス	92
図 5.8.2 キャリア状態遷移図	93
表 5.8.2 キャリア状態遷移定義表	94
図 5.8.2-1 マニュアルモード キャリア状態管理処理の流れ	97
図 5.8.2-2 オートモード キャリア状態管理処理の流れ	98
表 5.8.2 キャリアアクション関連メッセージ送信用クラス	98
図 5.8.3 アクセスモード状態遷移図	99
図 5.8.3 アクセスモード管理処理の流れ	100
表 5.8.3 アクセスモード関連メッセージ送信用クラス	100
図 5.8.4 ロードポート予約状態遷移図	101
表 5.8.4 ロードポート予約状態定義表	101
図 5.8.4 ポート予約状態管理処理の流れ	102
図 5.8.5 ロードポート/キャリア関連状態遷移図	103
表 5.8.5 ロードポート/キャリア関連状態遷移定義表	103
図 5.9.1 プロセスジョブ状態遷移図	105
表 5.9.1 プロセスジョブ状態遷移定義表	106
図 5.9.2 プロセスジョブ管理処理の流れ	108
表 5.9 プロセスジョブ関連メッセージ送信用クラス	109
図 5.10.1 コントロールジョブ状態遷移図	110
表 5.10.1 コントロールジョブ状態遷移定義表	111
図 5.10.2 コントロールジョブ管理処理の流れ	113

表 5.10	コントロールジョブ関連メッセージ送信用クラス.....	114
図 5.11	S10F1 端末要求の流れ.....	115
表 5.11	端末サービス関連メッセージ送信用クラス.....	115
図 5.12	ホストコマンド S2F41 処理の流れ	116
表 5.12	リモートコマンド関連メッセージ送信用クラス.....	116
図 5.13-1	汎用サービス要求、完了通信の流れ	エラー! ブックマークが定義されていません。
図 5.13-2	レチクル搬送ジョブ通信の流れ	エラー! ブックマークが定義されていません。
表 5.13	レチクル制御関連メッセージ送信用クラス.....	エラー! ブックマークが定義されていません。

1. はじめに

GEM 通信エンジン (以下、DSHEng5 と呼びます) は、半導体製造工場で採用されている SEMI スタandard と SECS, HSMS 通信規約に基づき、製造装置の通信機能を有し、装置情報の管理に関連する全般的なサービスを提供し、GEM をはじめ GEM300 対応に必要な機能をサポートするソフトウェアパッケージです。

本パッケージは装置、ホスト双方の通信機能を持っています。

本説明書では、DSHEng5 エンジンについて、その概要を説明します。

ユーザが、プログラミングをより容易にするため、.NET の C#, VB 言語を使用することを前提にしています。

DSHEng5 が提供するクラスの説明については「DSHEng5 GEM 通信エンジン・クラス説明書 Vol-1~Vol-6」を参照ください。(表 1.1.1.B DSHEng5 に一覧表)

DSHEng5 通信エンジンが提供する基本的な機能は以下の通りです。

- (1) HSMS-SS プロトコル通信をサポートします。
- (2) SECS-II メッセージ送受信機能のためのプログラミング手段を提供します。
- (3) SEMI スタandard GEM 仕様に準拠する機能をサポートします。(全てではありません)
- (4) 同様に、GEM-300 仕様をサポートします。(全てではありません)
- (5) 装置管理情報 (変数、RECIPE, Cj, PRJ) などの情報一元管理を行います。
- (6) アプリケーションに装置管理情報をアクセスインタフェースを提供します。
- (7) 装置管理情報のバックアップと再開時の復旧機能を提供します。

システム上の DSHEng5 の位置は下図のとおりです。対装置との SECS/HSMS 通信制御と関連情報の全てを DSHEng5 が行います。

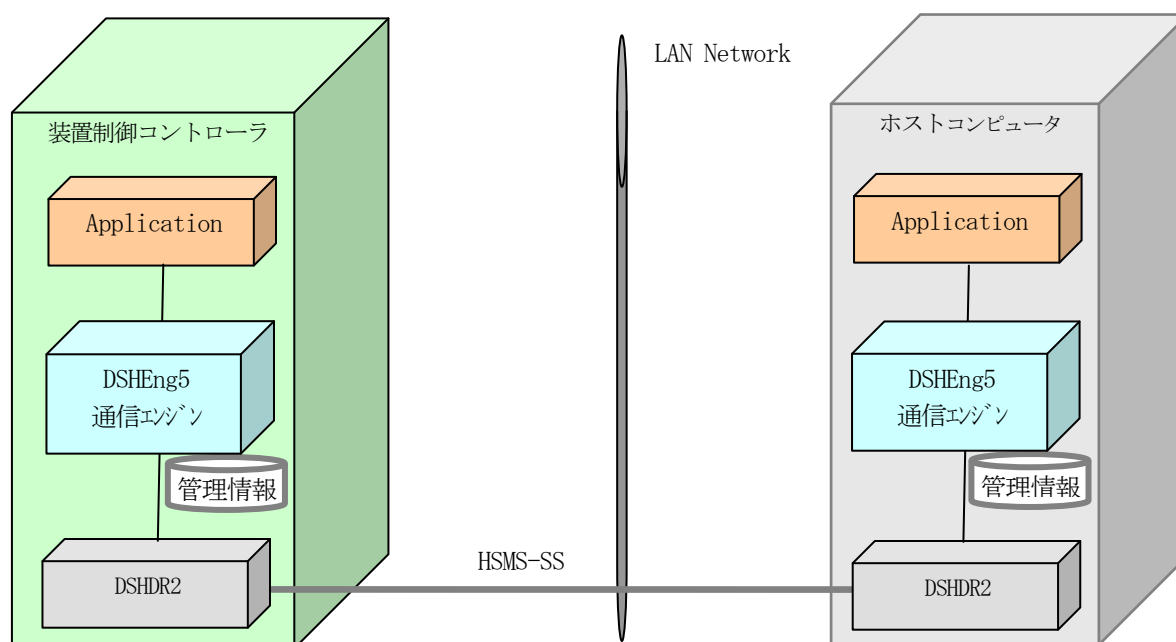


図 1-1 システムにおける DSHEng5 の位置

また、ユーザが求める独自仕様のための DSHEng5 のカスタマイズも可能です。

本パッケージが動作する OS 環境は Microsoft 社 Windows 8.1、Windows10 です。

DSHEng5 は、SEMI スタンド下の図の機能を実現することを念頭に設計されています。

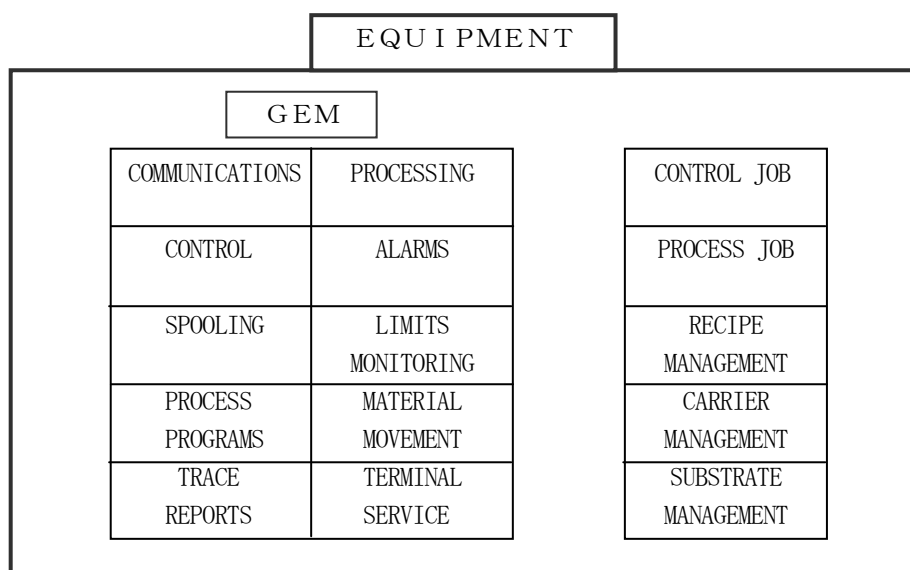
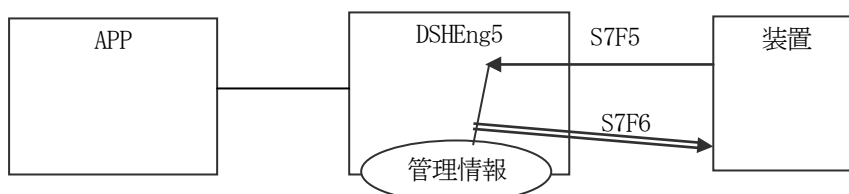


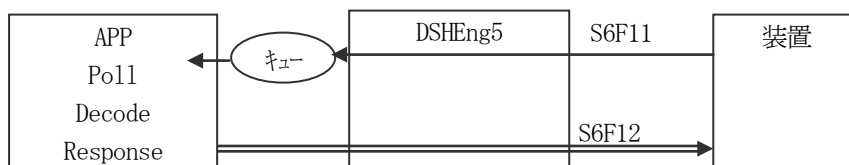
図 1-2 DSHEng5 の SEMI スタンドサポート対象機能

DSHEng5 は、装置間通信のやりとりに対する処理についてユーザができる限りシンプルにアプリケーションをプログラミングできるような仕組みと手段を提供します。

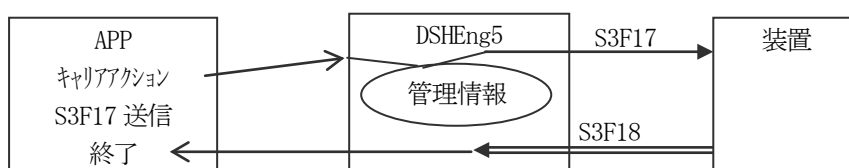
- (1) DSHEng5 が対応できる受信メッセージについてはユーザの手を煩わすことなく自動的に処理します。



- (2) APP が処理したい装置受信メッセージの場合、DSHEng5 が受信キューを通して APP に渡します。DSHEng5 は受信キューをポーリングするための関数とメッセージ内の情報を、プログラム処理がしやすい構造体にデコードするための関数を提供します。ユーザは SECS-II メッセージの構造を意識する必要がありません。また、応答メッセージの送信も応答情報を構造体に詰めるための関数ならびに応答するための関数を提供します。



- (3) ホストまたは装置が送信する S3F17, S7F3 メッセージなどの 1 次メッセージの送信は、専用クラスを使って簡単に送信できます。勿論、ユーザ自身で任意のメッセージを組み立て送信することも可能です。そしてそのためのメッセージエンコード用関数も使用することができます。



1. 1 関連資料

1. 1. 1 DSHeng5 通信エンジン関連資料

表 1. 1. 1-1 DSHeng5 通信制御エンジン一般関連文書一覧表

#	文書番号	文書名	注釈
1	DSHENG5-19-30300-00	DSHENG5 通信制御エンジン ユーザーズ・ガイド	DSHENG5 の全般的な機能の説明書です。
2	DSHENG5-19-30301-00	DSHENG5 起動ファイル定義仕様書	装置別の起動情報の定義方法の説明書です。
3	DSHENG5-19-30302-00	DSHENG5 装置管理情報定義仕様書 (変数、収集イベント、アラームその他)	DSHENG4 と同じ内容です。 定義ファイルはテキストファイルです。
4	DSHENG5-19-30303-00	装置管理情報定義ファイルコンパイル説明書	DSHENG4 と共通です。
5	DSHENG5-19-30304-00	DSHENG5 への手引き	DSHENG5 導入時に参考にする作業手順書です。
6	DSHENG5-19-30305-07	インストールと保存ファイル	製品インストール手順です。
7	DSHENG5-19-30308-00	DSHENG5, 起動ファイル、装置管理情報ファイル設定・編集プログラム説明書	DSHGEM-LIB, DSHeng5 共通
8	DSHENG5-19-30310-00	変数リミット監視機能 説明書	リミット監視の考え方、処理方法の説明書です。
9	DSHENG5-19-30351-00	バックアップファイル参照プログラム説明書	DOSコマンドでList構造表示します。

表 1. 1. 1-2 DSHeng5 GEM 通信エンジン・クラス説明書一覧表

#	文書番号	文書名	注釈
1	DSHENG5-19-30321-00	Vol-1 エンジン起動・停止、通信確立関連クラス (EngAPI、GEM 通信確立、予約装置変数関連)	クラスの構文、プロパティ、メソッドの詳しい説明書です。
2	DSHENG5-19-30322-00	変数情報関連クラス (EC, SV, DVVAL, CE, Report, Alarm)	
3	DSHENG5-19-30323-00	プロセス情報関連クラス (PP, FPP, RECIPE, PRJ, CJ, CARRIER, SUBSTRATE)	
4	DSHENG5-19-30324-00	SECS-II メッセージ送信クラス	
5	DSHENG5-19-30325-00	SECS-II 通信メッセージ情報保存クラス	
6	DSHENG5-19-30326-00	SECS-II 通信メッセージ エンコード/デコード処理クラス	

表 1.1.1-3 DSHEng4 互換クラス・ライブラリ関連文書一覧表

#	文書番号	文書名	注釈
1	DSHENG4-19-30361-00	ClassLib-Info-1 Vol-1 エンジン起動と管理情報クラス 編 Part-1	エンジン、装置起動 管理情報のアクセス
2	DSHENG4-19-30362-00	ClassLib-Info-2 Vol-1 エンジン起動と管理情報クラス 編 Part-2	管理情報のアクセス
3	DSHENG4-19-30363-00	ClassLib-Comm Vol-2 メッセージ通信クラス 編	GEM メッセージ送信
4	DSHENG4-19-30305-00	クラスライブラリ プログラミングの手引き	準備するファイルと開発ス テップ 手順も含む
5	DSHENG4-19-30306-00	クラス生成・消滅トレースと表示機能について	クラス・デバッグ用

表 1.1.1-4 DSHEng5 エンジン・HSMS 通信ドライバー関連文書一覧表

#	文書番号	文書名	注釈
1	DSHDR2-06-20000-02	DSHDR2 SECS/HSMS レベル2 通信制御ドライバー ユーザースマニュアル	SECS/HSMS 通信制御ドライバーの 説明書です。
2	DSHDR2-06-20040-0	DSHDR2 レベル2 通信ドライバー通信ログモニター説明書	リアルタイムで通信トランザクションをモタ ー画面で見ることができます。

表 1.1.1-5 DSHEng5 エンジン・デモ・プログラム関連文書一覧表

#	文書番号	文書名	注釈
1	DSHEng5-19-30500-00	DSHEng5 通信エンジン 装置 デモプログラムの紹介	全体構成、装置起動ファイル、 通信定義ファイル、画面の紹介
2	DSHEng5-19-30501-00	評価用デモプログラム インストールと保存ファイル	デモプログラム、setup.exe による インストール手順説明書です。
3	DSHEng5-19-30502-00	DSHEng5 通信エンジン・デモプログラム 説明書	デモプログラムの機能、操作画面、 WP シミュレーション関連の説明書です。
4	DSHEng5-19-30505-00	エンジン起動とメッセージ送受信処理 (要約)	GEM 通信関連処理の具体的な 内部説明書です。

1. 1. 2 SEMI スタンダード

表 1.1.2 SEMI スタンダード関連資料一覧表

番号	スタンダード技術資料名
1.	SEMI E4-0699 半導体製造通信スタンダード 1 (SECS-I)
2.	SEMI E5-1104 半導体製造通信スタンダード 1 (SECS-II)
3.	SEMI E30-1103 製造装置の通信及びコントロールのための包括的モデル (GEM)
4.	SEMI E37-0303 高速 SECS メッセージサービス (HSMS) 汎用サービス
5.	SEMI E37.1-96E 単一の選択セッションにおける高速 SECS メッセージサービス (HSMS-SS)
6.	SEMI E39-0703 オブジェクトサービススタンダード：概念、挙動およびサービス
7.	SEMI E39.1-0703 オブジェクトサービススタンダード (OSS) のための SECS-II プロトコル
8.	SEMI E40-0304 プロセス管理スタンダード
9.	SEMI E42-0704 レシピ管理スタンダード：コンセプト、挙動およびメッセージサービス
10.	SEMI E42.1-0704 レシピ管理スタンダード (RMS) のための SECS-II プロトコルスタンダード
11.	SEMI E40.1-0304 プロセス管理スタンダードの SECS-II のサポート
12.	SEMI E94-1104 コントロールジョブマネージメントの仕様
13.	SEMI E90-1104E2 基板トラッキング仕様
14.	SEMI E90.1-1104 SECS-II プロトコル基板トラッキングの暫定仕様

1. 2 サポート範囲

DSHeng5 は、ユーザの半導体製造装置を管理するホスト側ならびに装置側のプログラムの設計と製作を容易にするためのソフトウェアパッケージであり、以下のサポートを行います。

- (1) SEMI スタンダードに準拠した仕様についてサポートします。
前述の表 1.1.2 に示される技術資料の内容をサポートします。
(注) 全てをサポートするわけではありません。一部未サポートの部分がありますが、カスタマイズ可能な場合はサポートします。
- (2) GEM 関連機能
 - ・状態モデル：通信状態、コントロール状態、装置プロセッシング状態の管理と制御
 - ・変数情報管理とアクセス：装置定数(EC)、装置状態変数(SV)、データ変数(DWVAL)
 - ・収集イベント情報の管理とメッセージ送信：CEID、REPORTID、変数リンク情報
 - ・アラーム情報の管理とメッセージ送信
 - ・スプーリング機能
 - ・トレース機能
 - ・変数、イベント、レポート、アラーム情報等はテキストファイルでユーザが定義できます。
 - ・プロセスプログラムの管理
- (3) コントロールジョブ管理サービス機能
 - ・生成、状態管理、削除
- (4) プロセス管理サービス機能
 - ・生成、状態管理、削除
- (5) レシピ管理サービス機能
- (6) キャリア管理サービス機能
- (7) 基板トラッキング管理サービス機能
- (8) SECS-II メッセージ通信サービス処理
- (9) 装置管理情報のバックアップ機能と再起動時の復元
- (10) ユーザ固有の仕様に対してはカスタマイズも検討させていただきます。

2. DSEng5 ソフトウェア構成

DSHEng5 の基本的なソフトウェアシステムの構成を次に示します。

DSHEng5 エンジンのクラス呼び出しと、DSHEng5Class クラスライブラリを使った2つの構成を示します。

(1) DSEng5 エンジンのライブラリのクラスを使った場合の構成図

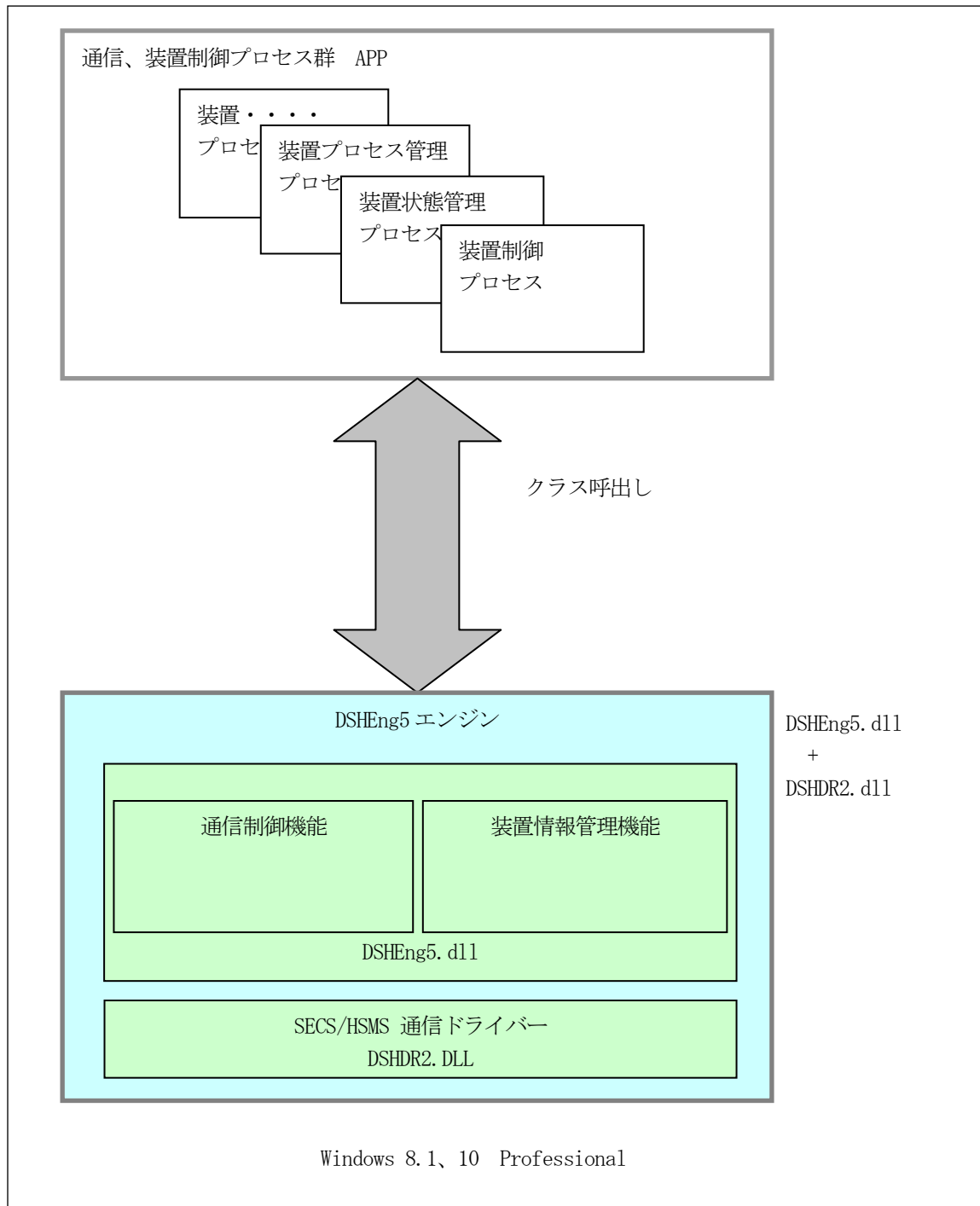


図 2-1 基本的なソフトウェア構成

(2) DSEng5Class クラス・ライブラリを使った場合の構成図 (DSEng4 の APP 用互換構成)

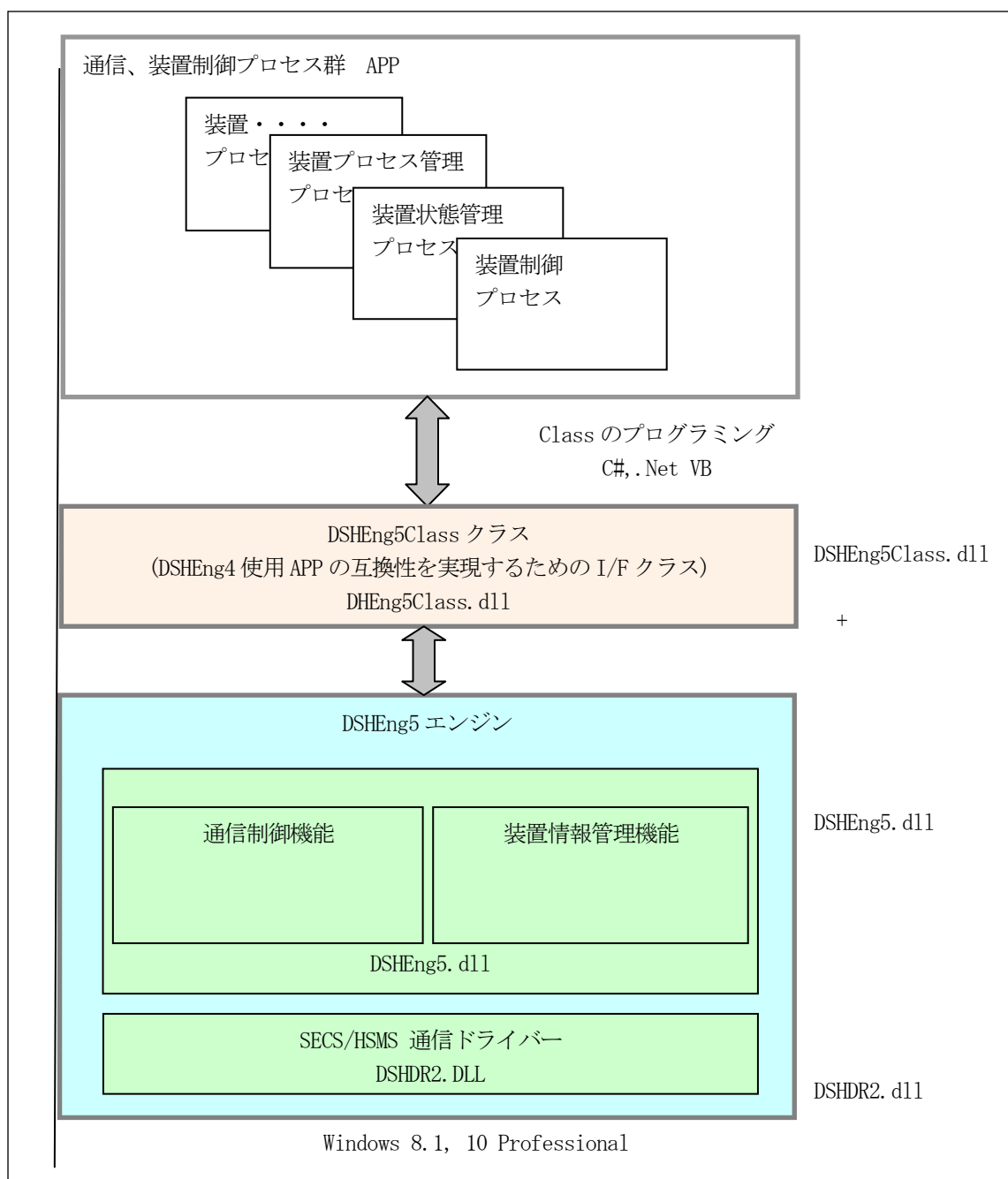


図 2-2 基本的なソフトウェア構成 (DSEng5Class ライブラリ使用)

3. 装置管理情報

(1) DSHEng5 はユーザに下表のシステム情報の管理とアクセス機能を提供します。

装置管理情報は装置別に定義する必要があります。

ここで述べる装置管理情報が DSHEng5 システムの基になります。

定義ファイル	○が定義可能
バックアップ	○がバックアップされる

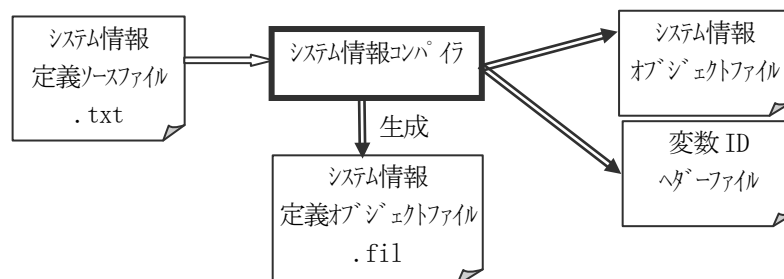
表 3 装置管理情報一覧

	情報の種類	定義ファイル	バックアップ	備考
1.	装置変数			
	(1) 装置定数(EC)	○	○	
	(2) 装置状態変数(SV)	○	○	
	(3) データ変数(DVVAL)	○	○	
2.	レポート(REPORT)	○	○	
3.	収集イベント(CE)	○	○	
4.	アラーム(ALM)	○		
5.	スプール(SPOOL)	○		
6.	トレース(TRACE)	○		
7.	プロセスプログラム(PP)	○	○	PP 情報使用の装置向け
8.	フォーマット付きプロセスプログラム(FPP)	○	○	FPP 情報使用の装置向け
9.	レシピ情報(RCP)	○	○	RCP 情報使用の装置向け
10.	キャリア情報(CAR)		○	
11.	基板情報(SUBST)		○	
12.	プロセスジョブ(PRJ)		○	
13.	コントロールジョブ(CJ)		○	

(2) DSHEng5 プログラムパッケージには、装置管理情報定義ファイルを編集し、コンパイルするためのプログラムツール DSHGEMSET.EXE プログラムが準備されています。(詳しくは

「SHGEM-LIB, DSHEng4 起動ファイル、管理情報ファイル設定・編集プログラム操作説明書」を参照)

コンパイラは、DSHEng5 初期化時に使用する変数などの定義情報を作成します。そして、さらに、ユーザプログラムで使用することができるヘッダファイルも作成してくれます。例えば、変数の名前に ID 値をマクロ定義した C 言語のヘッダファイルです。これにより、ユーザはマクロ定義した変数名を関数の引数として使用することができます。



(4) DSHEng5 は動作中、管理情報のバックアップを最大 4 世代分まで行います。

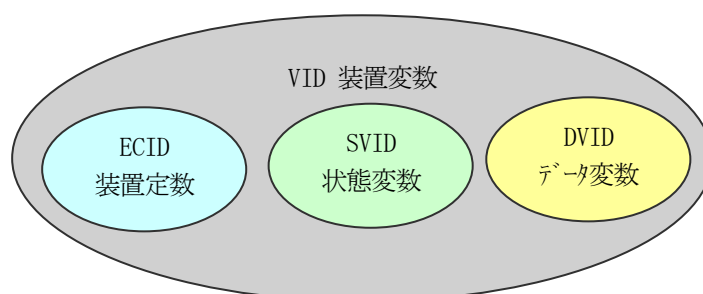
システム起動時にバックアップファイルが正常に保存されているかどうかを DSHEng5 ライブラリ関数を使って確認することができます。

装置変数には以下の3種類の変数があります。

- (1) 装置定数 (EC)
- (2) 装置状態変数 (SV)
- (3) 装置データ変数 (DVVAL)

これらの変数が SECS-II メッセージの中でどのように区別されるかについてですが、メッセージの中の変数 ID 名の表示によって以下のように区別されます。(SEMI スタンド資料参照)

- ①VID と表示されているものは、装置定数、装置状態変数、装置データ変数が全て対象になります。
- ②ECID と表示されているものは、装置定数だけが対象になります。
- ③SVID と表示されているものは、装置状態変数だけが対象になります。
- ④DVID と表示されているものは、装置データ変数だけが対象になります。



以上のことから、本 DSHEng5 システムにおいて、変数 ID の値はシステムの中でユニークであることを前提としています。即ち、同じ ID 値を有する複数の変数定義を行わないことが必要です。

表 3-1 変数と SECS-II メッセージの関連

#	SECS MSGID	メッセージ名	対象とする変数		
			装置定数	状態変数	データ変数
1.	S1F3, 4,	装置状態要求		○	
2.	S1F11, 12	状態変数一覧要求		○	
3.	S2F13, 14	装置定数要求	○		
4.	S2F15, 16	装置定数変更	○		
5.	S2F23, 24	トレース条件設定		○	
6.	S2F29, 30	装置定数名一覧要求	○		
7.	S2F33, 34	レポート設定	○	○	○
8.	S2F43, 44	変数リミット属性定義	○	○	○
9.	S2F45, 46	変数リミット属性一覧要求	○	○	○

DSHEng5 においては、変数アクセスのためのクラスはそれぞれの変数の種類に対応して準備されています。ほとんどのアクセスメソッドは引数として変数 ID を指定します。

3. 1 装置定数 (EC, SV, DVVAL) 情報とクラス

対装置との通信に使用される変数、装置制御に使用される変数を登録し、アクセスすることができます。装置変数と他プログラムとの関連を図 3.1 に示します。

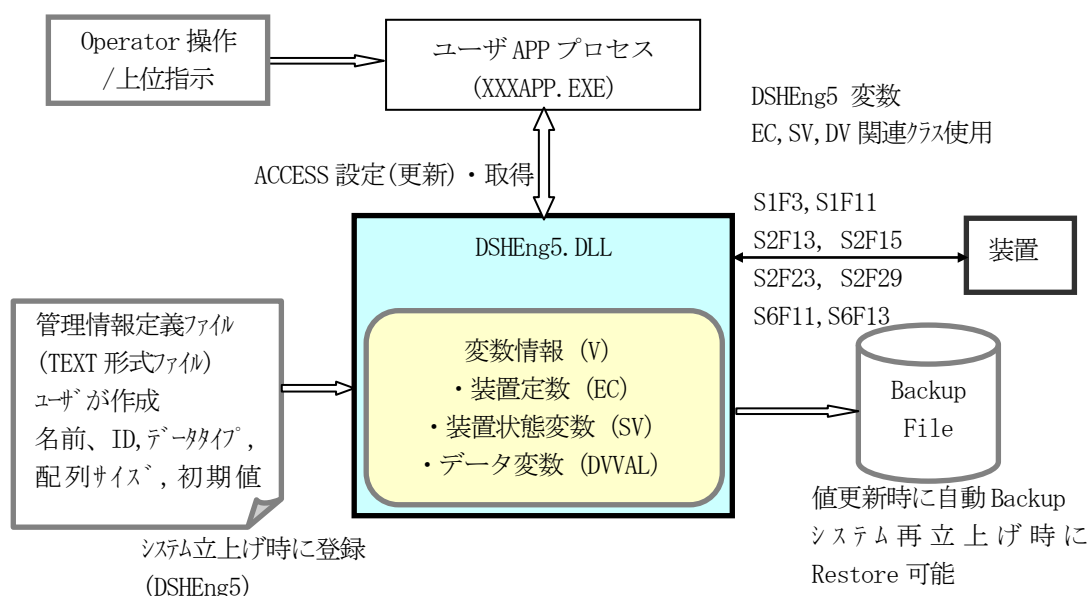


図 3.1 変数情報の関連図

- ・ユーザがシステムに登録する変数名、ID、タイプ、初期値などを管理情報定義ファイルに定義します。変数の種類は、装置定数(EC)、装置状態変数(SV)、データ変数 (DVVAL) の3種類です。
- ・DSHEng5 立上げ時、個別装置に対し装置管理情報定義ファイルの情報をシステム内部に登録します。
- ・DSHEng5 がこの情報を管理しますので、ユーザは、値の設定、取得だけを行えばいいことになります。変数 ID をキーにして変数アクセスクラス呼出しによって変数値の設定・取得操作を行います。
- ・値が更新された変数は、バックアップの指定がされていればバックアップファイルに自動的に保存されます。バックアップされた情報は、システム再立上げ時にシステム内部に復帰させることができます。
- ・ホストから S1F3 などの SECS-II メッセージによる変数値の参照を行えば変数の値を取得することができます。
- ・収集イベントのレポート ID にリンクされる変数があります。

次ページに管理情報定義ファイル内での変数定義の例と関連クラスを示します。

装置管理情報定義ファイルについての詳細は、「**DSHEng5 装置管理情報定義仕様書**」を参照ください。
(DSHEng4 装置通信エンジンと共通です。)

[定義例]

3種類の変数がありますが、基本的には、同じ書式になります。

例えば、装置定数 EC_Chamber1Temp（温度定数）は次のように定義します。

```
def_ec EC_Chamer1Temp{ // 装置定数定義開始と定数名です。
    ecid:      0x00000401 // ECID です。
    format:    U2[1]      // 温度データの型です。SECS のデータアイテム対応です。
    nominal:   40         // デフォルト温度値です。
    units:     Degree     // 値の単位（文字列）です。
    min:       10        // 取り得る最小値です。
    max:       100       // 取り得る最大値です。
    limit:     10, 80, 100 // リミット値（チェック用）です。
}
```

装置状態変数(SV)の定義は、def_sv で始めます。装置変数については、値が変化したときに装置がホストにイベントを通知するための収集イベント ID 名を指定することもできます。

データ変数(WVAL)の定義は、def_dv で始めます。

DSHEng5 は、定義に使用する英文字は大文字、小文字の区別はしません。

ただし、値を文字列で表す場合は、大文字、小文字の区別が必要になります。

3. 1. 1 装置定数 - class_EC クラス

メソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int get_id_name_list()	登録されている ID、名前リストを取得します。
4	public int get_id_by_name()	変数名からその ID を取得します
5	public int get_name()	変数 ID の名前を取得します
6	public int set()	指定 ID に変数情報を設定します。
7	public int get()	指定 ID の変数情報を取得します。
8	public int get_format()	指定 ID の変数値フォーマットを取得します。
9	public int get_size()	指定 ID の変数値のサイズを取得します。
10	public int get_units()	指定 ID の変数値の物理単位を取得します。
11	public int set_value()	当該変数インスタンスに値を設定します。
12	public int get_value()	当該変数インスタンスの値を取得します。
13	public int get_nominal()	指定 ID の規定値を取得します。
14	public int get_min_value()	指定 ID の最小値を取得します。(制限値)
15	public int get_max_value()	指定 ID の最大値を取得します。(制限値)
16	public int resize_V_array()	指定 ID の配列変数の配列サイズを変更します。
17	public int resize_V_Linklist()	format が L(List) 変数にリンクされる変数 ID 数の変更します。
18	public int set_V_Linklist()	format が L(List) 変数にリンクされる変数の ID を追加します。
19	public int set_V_Linklist()	format が L(List) 変数にリンクされる変数の ID を 1 個設定します。
20	public int set_V_Linklist_all()	format が L(List) 変数に全リンク変数 ID を一度に設定します
21	public int set_limit_info()	指定 ID 変数にリミット情報を設定します。
22	public int get_limit_info()	指定 ID 変数のリミット情報を取得します。
23	public int del_limit_info()	指定 ID 変数のリミット情報を削除します。
24	public int check_val()	指定 ID の値が最小、最大値の範囲内かどうかを調べます。

(参照資料) DSHeng5 GEM 通信エンジン・クラス説明書 vol-4 の 7.4

3. 1. 2 装置状態変数 - classSV クラス

メソッドは、3. 1. 1 の classEC のメソッドと同じです。

3. 1. 3 装置データバリュウ変数 - classDV クラス

メソッドは、3. 1. 1 の classEC のメソッドと同じです。

3.2 収集イベント (CE)、レポート (REPORT) 情報

収集イベント情報は変数情報と同様に、管理情報定義ファイル内に定義します。
 収集イベント、レポートと他プログラムとの関連を図 3.2 に示します。

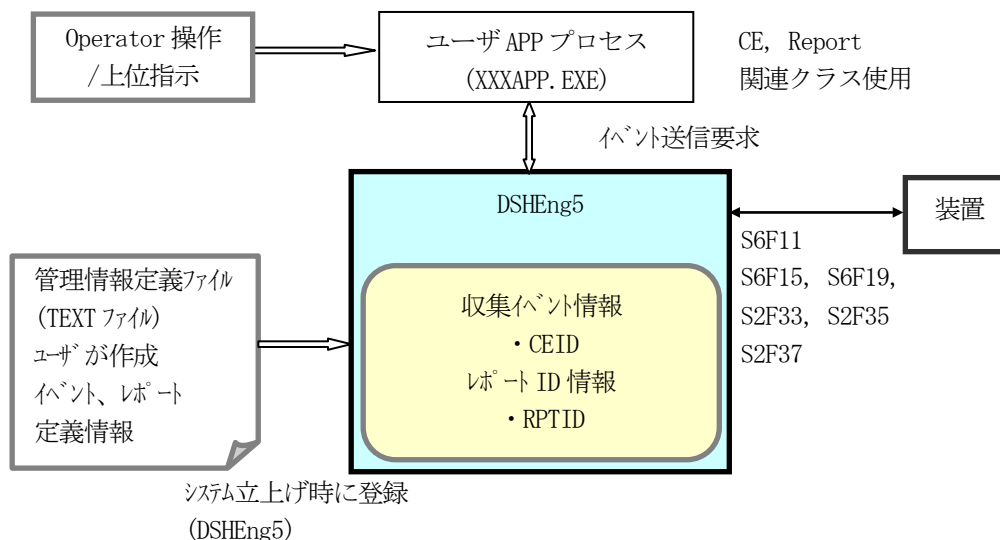
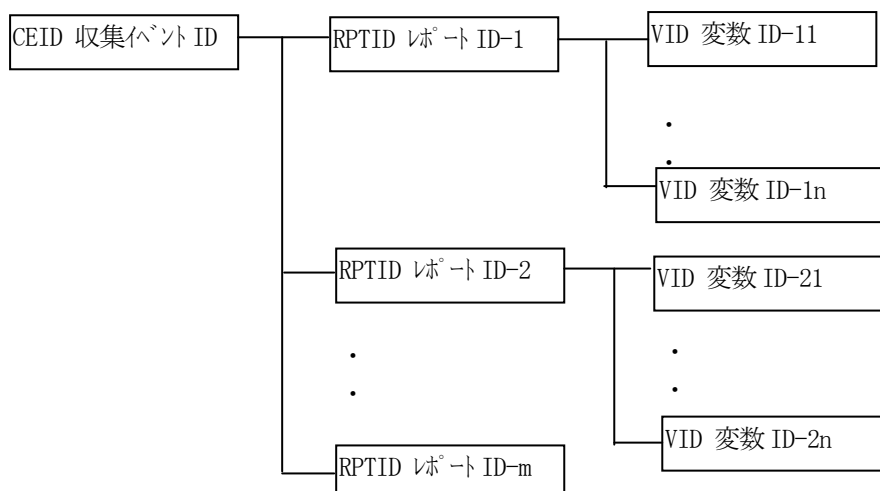


図 3.2 収集イベント情報関連図

- ・ユーザが、システムに登録するイベント名、ID、リンクレポート名を管理情報定義ファイルに定義します。
- ・また、イベント ID にリンクするレポート ID、レポート ID にリンクする変数の値を、ダイナミックに変更し、装置に通知することができます。
- ・イベント通知許可/禁止の変更指令もユーザからのクラスで行うことができます。
- ・収集イベント ID とそれにリンクされるレポート ID ならびに変数の関係は次のようになります。



次に CEID の定義例と関連クラスを示します。

[定義例]

1個の収集イベント(CE)を、例えば次のように定義します。

```
def_ce CE_Port1AccessMode{                               // 収集イベント定義開始とイベント名です。
    ceid:      2201                                       // CEIDです。
    enabled:   1                                           // 有効(1)/無効(0)の指定です。
    rptname:   RP_Port1AccessMode                         // リンクされるレポート名です。
}
```

収集イベントにリンクされるレポート(RPT)を、例えば次のように定義します。

```
def_report RP_Port1AccessMode{                           // レポート定義開始とレポート名です。
    rptid:     1201                                       // RPTID(レポートID)です。
    vname:     V_Port1AccessMode                         // レポートにリンクされるデータ変数名です。
}
```

3. 2. 1 CE (収集イベント) - cass_CE クラス

(1) class_CE クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int get_id_name_list()	登録されている ID、名前リストを取得します。
4	public int get_id_by_name()	変数名からその ID を取得します
5	public int get_name()	変数 ID の名前を取得します
6	public int set()	指定 ID に変数情報を設定します。
7	public int get_rp_list()	CEID にリンクされているレポート ID リストを取得します。
8	public int get_rp_name_list()	CEID にリンクされているレポート ID の名前リストを取得します。
9	public int set_ceed()	ceed(有効/無効) を設定します。
10	public int get_ceed()	ceed(有効/無効) を取得します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-2 の 4.1

(2) CE レポートを送信するメッセージは、S6F11 です。

S6F11 は、class_SendS6F11 を使って送信します。メソッドは次の通りです。

①非ブロックモード

```
public int SendS6F11(uint ceid, class_CALLBACK.callback_S6F11 callback, uint upara)
```

②ブロックモード

```
public int Send_wait(uint ceid)
```

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-4 の 7.4

3. 2. 2 レポート - class_Report クラス

class_Report クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int get_id_name_list()	登録されている ID、名前リストを取得します。
4	public int get_id_by_name()	変数名からその ID を取得します
5	public int get_name()	レポート ID の名前を取得します。
6	public int get()	指定 ID のレポート情報を取得します。
7	public int get_v_list()	レポート ID にリンクされている変数 ID リストを取得します。
8	public int get_v_name_list()	レポート ID にリンクされている変数 ID の名前リストを取得します。
9	public int get_TRP_CONTENT()	レポート ID に含む全てのリンク変数をリスト構造で取得します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-2 の 5.1

3.3 アラーム情報

アラーム情報は変数情報と同様に、管理情報定義ファイル内に定義されます。

アラーム情報と他プログラムとの関連を図 3.3 に示します。

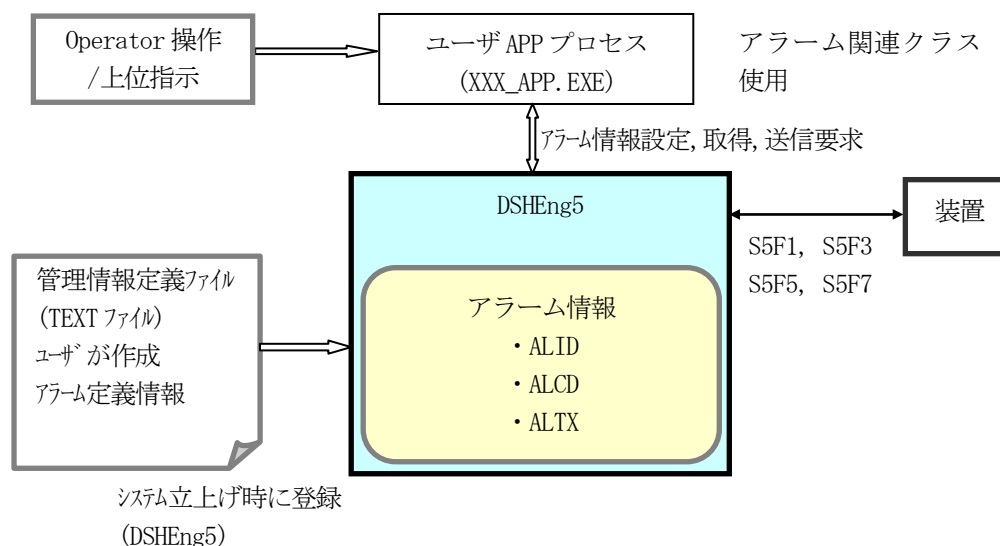


図 3.3 アラーム情報関連図

- ・ユーザがシステムに登録するアラーム名、ALID、ALCD、ALTX を管理情報定義ファイルに定義します。また、アラーム発生／復旧時に同時に送信される収集イベントを加えることができます。
- ・ホスト側では、アラーム情報が S5F1 メッセージで送信されてきます。DSHEng5 は S5F1 に含まれるアラーム情報を TAL_S5F1_INFO クラスのインスタンスにデコードするためのクラスを使用することができます。

アラーム定義例と関連クラスを示します。

[定義例]

アラーム情報は例えば次のように定義します。

```
def_alarm AL_AlarmTempOver {
    alid:      1 // アラームの定義開始とアラーム名です。
    altx:      "Chamber-1 Temperature Over" // ALID です。
    alcd:      2 // ALTX(アラームテキスト)です。
    ce_on:     CE_AlarmOn // ALCD です。
    ce_off:    CE_AlarmOff // 発生時に通知するイベント名です。(もしあれば)
} // 復旧時に通知するイベント名です。(もしあれば)
```

[関連クラス - class_Alarm]

class_Alarm クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int get_name_list()	登録されている ID 名リストを取得します。
4	public int get_id_name_list()	登録されている ID, 名前リストを取得します。
5	public int get_id_by_name()	名前からその ID を取得します。
6	public int get_name()	アラーム ID の名前を取得します。
7	public int get()	指定 ID のアラーム情報を取得します。
8	public int get_alcd()	指定 ID の ALCD を取得します。
9	public int get_altx()	指定 ID の ALTX を取得します。
10	public int set_ceid_on()	指定 ID のアラーム発生時に発する CEID を設定します。
11	public int set_ceid_off()	指定 ID のアラーム復旧時に発する CEID を設定します。
12	public int get_ceid_on()	指定 ID のアラーム発生時に発する CEID を取得します。
13	public int get_ceid_off()	指定 ID のアラーム復旧時に発する CEID
14	public int set_enabled()	指定 ID のアラームを有効または無効にします。
15	public int get_enabled()	指定 ID のアラームの有効/無効状態を取得します。

(参照資料) DSHeng5 GEM 通信エンジン・クラス説明書 vol-2 の 6.1

(2) アラームを送信するメッセージは、S5F1 です。

S5F1 は、class_SendS5F1 を使って送信します。メソッドは次の通りです。

①非ブロックモード

```
public int SendS5F1(uint alid, int on_off,
class_CALLBACK.callback_S5F2 callback, UInt32 upara)
```

②ブロックモード

```
public int SendS5F1_wait(uint alid, int on_off, ref int ack5)
```

(参照資料) DSHeng5 GEM 通信エンジン・クラス説明書 vol-4 の 8.1

3. 4 スプール情報

スプーリングは、装置がホストとの通信確立を失った間、装置が送信しようとしたが送信できなかった通信メッセージを一旦装置側でスプール情報保存領域に保存します。そして、通信確立が復帰した際に、ホストの要求に基づいて、装置が保存領域にスプールしたメッセージを順次ホストに送信します。

ホスト側はスプール対象メッセージを予め装置に通知することと、溜まっているスプールメッセージの送信要求を行うことができます。

他プログラムとの関連を図 3. 4 に示します。

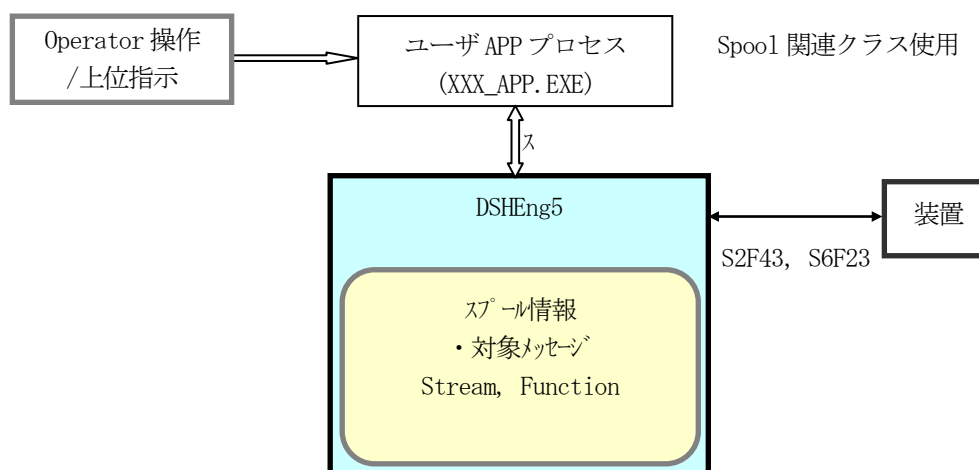


図 3. 4 スプール情報関連図

- ・ホストは登録したいスプール対象メッセージを S2F43 を使って装置に通知することができます。

[関連クラス - class_Spool]

class_Spool クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public void Dispose()	プロパティをクリアし、インスタンスを消滅させます。
2	public void clear()	スプール情報を削除します。
3	public int add_spool_SF ()	sstream, function のメッセージを追加します。
4	public int get()	Stream コードを指定して、function コードを取得します。
5	public static void purge()	Spool され、送信待ちのメッセージをすべて消去します。
6	public static int copy()	インスタンスを別のインスタンスにコピーします。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-2 の 3.20

3.5 トレース情報

ホストは、装置に対し、特定の装置状態変数の値を監視し、逐一ホストに送信するためのトレース機能を有します。

ホストはトレース設定情報を定義登録するとともに S2F23 メッセージを使ってトレース情報を設定するとともにトレース指令を出します。装置は、設定情報に従ってトレースした状態変数を S6F1 メッセージを使って報告してきます。

他プログラムとの関連を図 3.5 に示します。

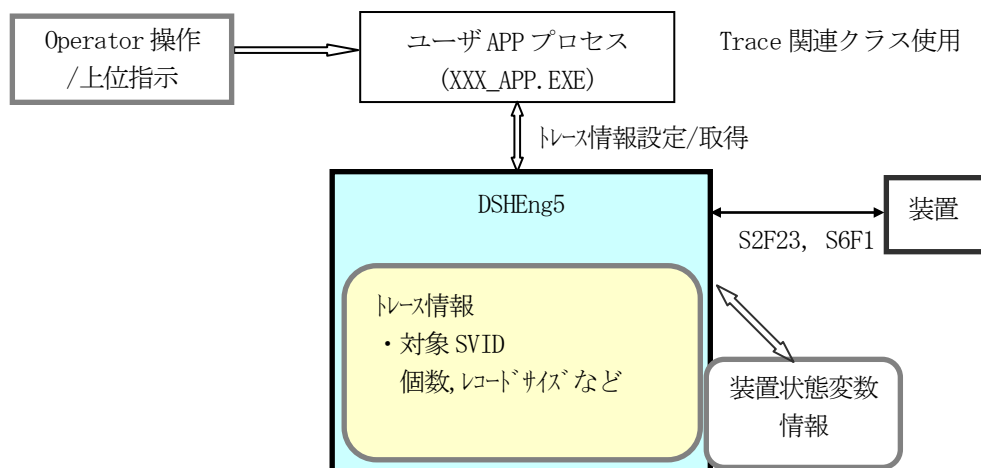


図 3.5 トレース情報関連図

トレース情報の定義例と関連クラス名を示します。

[関連クラス - class_Trace]

class_Trace クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int allocate()	トレース ID を登録します。
2	public int set()	指定 ID のトレース情報を設定します。
3	public int get()	指定 ID のトレース情報を取得します。
4	public void delete_all_id()	登録されているすべてのトレース ID を削除します。
5	public void delete()	指定された ID を削除します。
6	public int enable()	指定された ID のトレースモニター処理を開始します。
7	public int get_id_count()	登録されている ID 数を取得します。
8	public int get_id_list()	登録されている ID リストを取得します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-2 の 3.15

3.6 プロセスプログラム (PP) 情報

プロセスプログラム (PP) 情報は製造装置が処理に使用するレシピ情報です。

DSHEng5 は、登録された PP 情報を管理し、アプリケーションに PP 情報のアクセスサービスを提供します。

オート制御の場合 PP 情報は、装置処理開始前にホストが装置に対し S7F3 メッセージを使って与えます。そして装置が保持しているプロセスプログラム情報を S7F5 メッセージを使って取得することができます。

他プログラムとの関連を図 3.6 に示します。

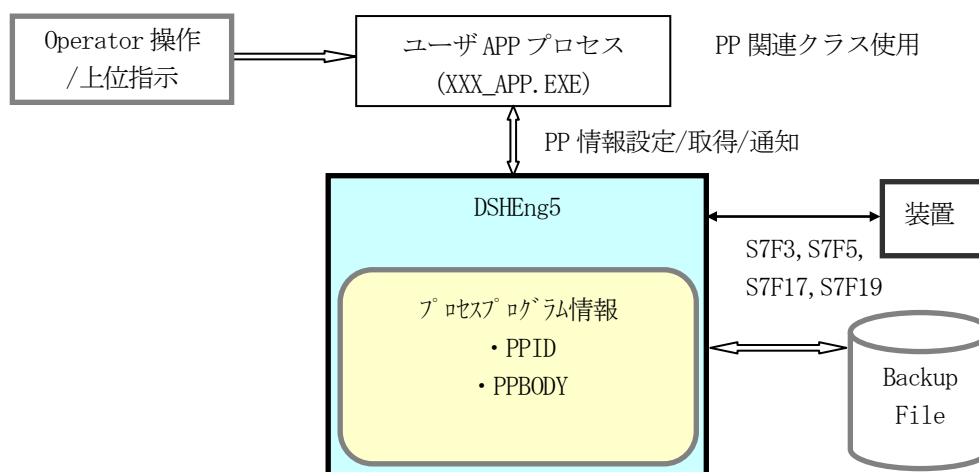


図 3.6 プロセスプログラム情報関連図

[関連クラス - class_PP]

class_PP クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int allocate_id()	PPID を登録します。
4	public int set()	指定 ID の PP 情報を設定します。
5	public int get()	指定 ID の PP 情報を取得します。
6	public int set_ppbody()	指定 ID の ppbody を設定します。
7	public string get_ppbody()	指定 ID の ppbody を取得します。
8	public void delete_all_id()	全登録 ID を削除します。
9	public void delete()	指定 ID の情報を削除します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-3 の 2.1

3. 7 フォーマット付きプロセスプログラム (FPP) 情報

DSHEng5 は、登録された FPP 情報を管理し、アプリケーションに FPP 情報のアクセスサービスを提供します。

オート制御の場合、FPP 情報は装置処理開始前にホストが装置に対し S7F23 メッセージを使って与えます。そして装置が保持しているプロセスプログラム情報を S7F25 メッセージなどを使って取得することができます。

他プログラムとの関連を図 3.7 に示します。

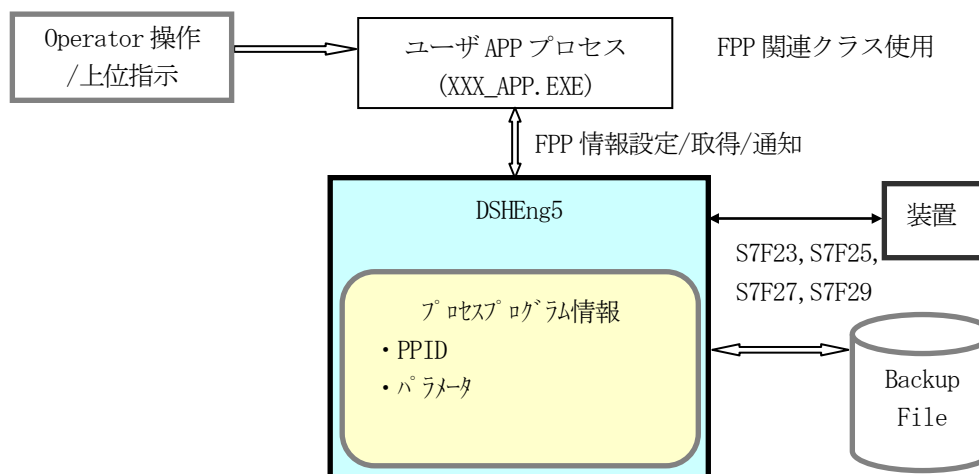


図 3.7 フォーマット付きプロセスプログラム情報関連図

[関連クラス - class_FPP]

class_FPP クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int allocate_id()	FPPID を登録します。
4	public int set()	指定 ID の FPP 情報を設定します。
5	public int get()	指定 ID の FPP 情報を取得します。
6	public int set_mdln()	指定 ID のモデル名を設定します。
7	public string get_mdln()	指定 ID のモデル名を取得します。
8	public int set_softrev()	指定 ID のソフトウェア版番号を設定します。
9	public int get_softrev()	指定 ID のソフトウェア版番号を取得します。
10	public void delete_all_id()	全登録 ID を削除します。
11	public void delete()	指定 ID の情報を削除します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-3 の 3.1

3.8 レシピ (RCP) 情報

レシピ (RCP) 情報は製造装置が処理に使用する情報です。

DSHEng5 は、登録された RCP 情報を管理し、アプリケーションに RCP 情報のアクセスサービスを提供します。

オート制御の場合 RCP 情報は装置処理開始前にホストが装置に対し S15F13 メッセージを使って与えます。そして装置が保持しているレシピ情報を S15F17 メッセージなどを使って取得することができます。

他プログラムとの関連を図 3.8 に示します。

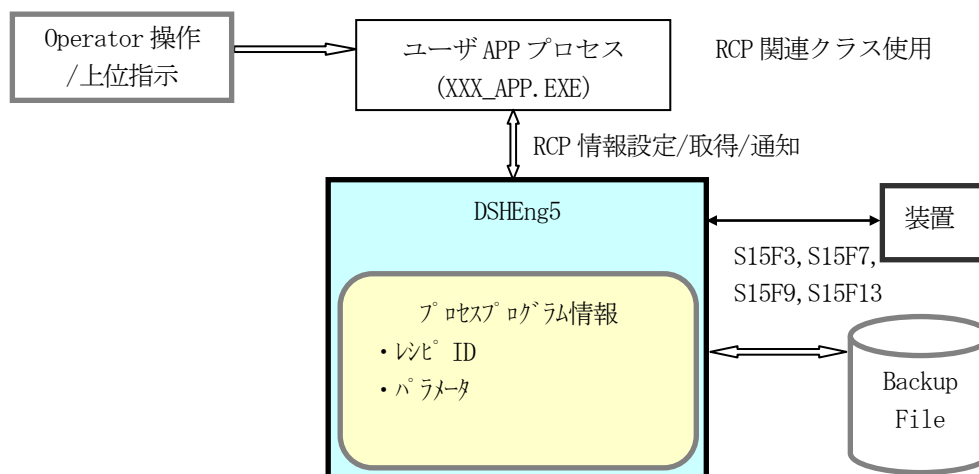


図 3.8 レシピ情報関連図

[関連クラス - class_RCP]

class_RCP クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int allocate_id()	レシピ ID を登録します。
4	public int set()	指定 ID のレシピ情報を設定します。
5	public int get()	指定 ID のレシピ情報を取得します。
6	public int set_rcpbody()	指定 ID のレシピボディを設定します。
7	public string get_rcpbody()	指定 ID のレシピボディを取得します。
8	public int set_state()	指定 ID のレシピ状態を設定します。
9	public int get_state()	指定 ID のレシピ状態を取得します。
10	public int rename_rcpid()	レシピ名を変えます。
11	public void delete_all_id()	全登録 ID を削除します。
12	public void delete()	指定 ID の情報を削除します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-3 の 4.1

3.9 キャリア (CAR) 情報

キャリア情報は製造装置での搬送ならびに処理の対象となる単位であり、内部のロットに収納されている基板情報が含まれます。

DSHEng5 は、登録されたキャリア情報を管理し、アプリケーションにキャリア情報のアクセスサービスを提供します。

キャリアの内部情報はキャリアのウエハの搭載時に決まり、装置においてはポートへのロード時に装置内部に登録され、その情報はホストから装置に対する S3F17 メッセージによって与えられます。

ユーザプログラムは、必要に応じてキャリア・オブジェクトの生成、パラメータ情報の設定、キャリア状態の更新などを DSHEng5 クラスを使って行うことになります。

他プログラムとの関連を図 3.9 に示します。

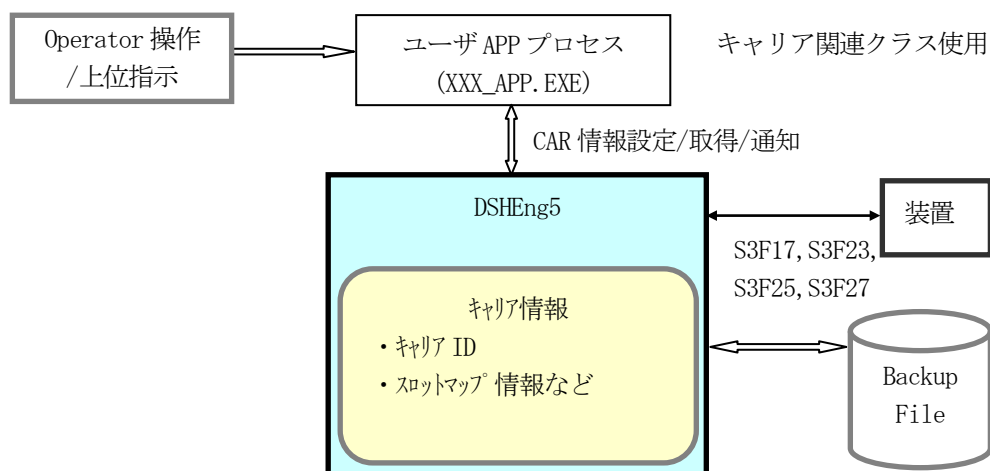


図 3.9 キャリア情報関連図

次にキャリア情報関連クラス名を示します。

[関連クラス - class_CAR]

class_CAR クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int allocate_id()	レシピ ID を登録します。
4	public int set()	指定 ID のレシピ情報を設定します。
5	public int get()	指定 ID のレシピ情報を取得します。
6	public int set_rcpbody()	指定 ID のレシピボディを設定します。
7	public string get_rcpbody()	指定 ID のレシピボディを取得します。
8	public int set_state()	指定 ID のレシピ状態を設定します。
9	public int get_state()	指定 ID のレシピ状態を取得します。
10	public int rename_rcpid()	レシピ名を変えます。
11	public void delete_all_id()	全登録 ID を削除します。
12	public void delete()	指定 ID の情報を削除します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-3 の 7.1

3. 10 基板 (SUBSTRATE) 情報

基板はキャリア内の基板スロット内に収納されて搬送され、処理対象単位になります。

DSHEng5 は、登録された基板情報を管理し、アプリケーションに基板情報のアクセスサービスを提供します。

基板情報はキャリアへのウェハーの搭載時に決まり、システムに登録され、その情報はホストから装置に対し S3F17 メッセージによって与えられます。

ユーザプログラムは、必要に応じて基板・オブジェクトの生成、パラメータ情報の設定、基板状態の更新などを DSHEng5 クラスを使って行うことになります。

他プログラムとの関連を図 3.9 に示します。

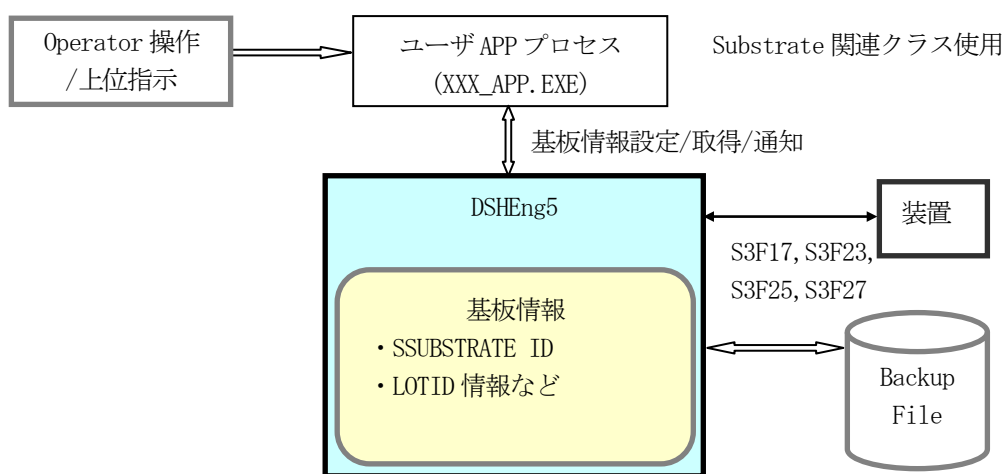


図 3.10 基板情報関連図

次ページに基板情報関連クラス名を示します。

[関連クラス - class_SUBST]

class_Trace クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int allocate_id()	基板 ID を登録します。
4	public int set()	指定 ID の基板情報を設定します。
5	public int get()	指定 ID の基板情報を取得します。
6	public void delete()	指定 ID の情報を削除します。
7	public void delete_all_id()	全登録 ID を削除します。

(参照資料) DSHeng5 GEM 通信エンジン・クラス説明書 vol-3 の 8.1

3. 11 プロセスジョブ (PRJ) 情報

プロセスジョブ情報は製造装置が処理に使用するプロセス処理単位であり、中には、処理されるキャリア、ウエハー、レシピ情報などが含まれます。

DSHEng5 は、登録されたプロセスジョブ情報を管理し、アプリケーションにプロセスジョブ情報のアクセスサービスを提供します。

オート制御の場合、プロセスジョブ情報は、装置処理開始前にホストから装置に対して S16F11 または S16F15 メッセージによって与えられます。

ユーザプログラムは、必要に応じてオブジェクトの生成、パラメータ情報の設定、状態の更新などを DSHEng5 クラスを使って行うことになります。

他プログラムとの関連を図 3. 10 に示します。

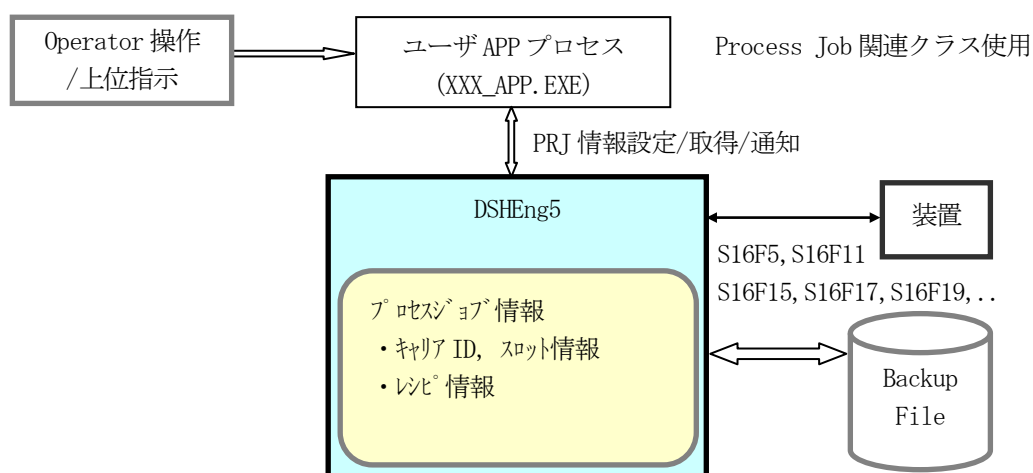


図 3. 11 プロセスジョブ情報関連図

次ページにプロセスジョブ情報関連クラス名を示します。

[関連クラス - class_PRJ]

class_PRJ クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int allocate_id()	プロセスジョブ ID を登録します。
4	public int set()	指定 ID のプロセスジョブ情報を設定します。
5	public int get()	指定 ID のプロセスジョブ情報を取得します。
6	public int set_mf()	指定 ID の MF 値を設定します。(MF : Material Format)
7	public string get_mf()	指定 ID の MF 値を取得します。
8	public int get_car_count()	指定 ID の処理対象キャリア数を取得します。
9	public int add_car_list()	指定 ID の処理キャリアリストに 1 個キャリア情報を追加します。
10	public int get_car_list()	指定 ID の処理対象のキャリア情報リストを取得します。
11	public int get_mid_count()	指定 ID の処理対象 MID 数を取得します。
12	public int add_mid_list()	指定 ID の処理 MID リストに 1 個 MID 情報を追加します。
13	public int get_mid_list()	指定 ID の処理対象の MID 情報リストを取得します。
14	public int set_prrecipemethod()	指定 ID のレシピ処理方法を設定します。
15	public int get_prrecipemethod()	指定 ID のレシピ処理方法を取得します。
16	public int set_rcp_info()	指定 ID のレシピ情報を設定します。
17	public int get_rcp_info()	指定 ID のレシピ情報を取得します。
18	public int set_prprocessstart()	指定 ID の起動方法を設定します。
19	public int get_prprocessstart()	指定 ID の起動方法を取得します。
20	public int get_ceid_count()	指定 ID の一時停止をする要因になる CEID の数を取得します。
21	public int add_ceid_list()	指定 ID の一時停止のイベントリストに CEID を追加します。
22	public int get_ceid_list()	指定 ID の一時停止の CE イベントリストの情報を取得します。
23	public int set_state()	指定 ID の state に値を設定します。
24	public int get_state()	指定 ID の state に値を取得します。
25	public void delete_all_id()	登録されている全登録 ID を削除します。
26	public void delete()	指定 ID を削除し、管理から外します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-3 の 5.1

3. 12 コントロールジョブ (CJ) 情報

コントロールジョブ情報は製造装置が処理に使用するコントロール処理単位であり、中には、処理されるべきプロセスジョブ、キャリア情報などが含まれます。

DSHEng5 は、登録と登録済みのコントロールジョブ情報を管理し、ユーザにコントロールジョブ情報のアクセスサービスを提供します。

オート制御の場合、コントロールジョブ情報は装置の処理開始前にホストから装置に対し S14F9 メッセージによって与えられます。

ユーザプログラムは、必要に応じてオブジェクトの生成、パラメータ情報の設定、状態の更新などを DSHEng5 クラスを使って行うことになります。

他プログラムとの関連を図 3. 11 に示します。

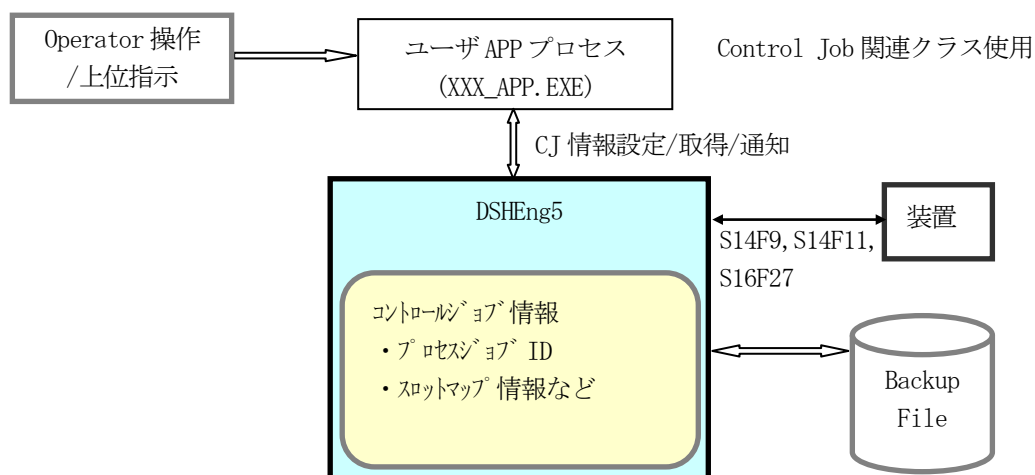


図 3. 12 コントロールジョブ情報関連図

次ページにコントロールジョブ情報関連クラス名を示します。

[関連クラス - class_CJ]

class_CJ クラスの情報にアクセスするメソッドは下表の通りです。

	メソッド名	説明
1	public int get_id_count()	登録されている ID 数を取得します。
2	public int get_id_list()	登録されている ID リストを取得します。
3	public int allocate_id()	コントロールジョブ ID を登録します。
4	public int set()	指定 ID のコントロールジョブ情報を設定します。
5	public int get()	指定 ID のコントロールジョブ情報を取得します。
6	public void delete()	指定 ID のコントロールジョブを削除します。
7	public int set_TMTRL_OUT_STAT()	指定 ID の MtrlOutByStatus 情報を設定します。
8	public int get_TMTRL_OUT_STAT()	指定 ID の MtrlOutByStatus 情報を取得します。
9	public int set_state()	指定 ID の state を設定します。
10	public int get_state()	指定 ID の state を取得します。
11	public int set_start_method()	指定 ID の StartMethod を設定します。
12	public int get_start_method()	指定 ID の StartMethod を取得します。
13	public int set_processing_order_mgmt()	指定 ID の ProcessingOrderMgmt を設定します。
14	public int get_processing_order_mgmt()	指定 ID の ProcessingOrderMgmt を取得します。
15	public int set_data_collect_plan()	指定 ID の DataCollectionPlan を設定します。
16	public int get_data_collect_plan()	指定 ID の DataCollectionPlan を取得します。
17	public int set_TMTRL_OUT_SPEC()	指定 ID の MtrlOutSpec を設定します。
18	public int get_TMTRL_OUT_SPEC()	指定 ID の MtrlOutSpec を取得します。
19	public int set_TPAUSE_EVENT()	指定 ID の PauseEvent を設定します。
20	public int get_ TPAUSE_EVENT()	指定 ID の PauseEvent を取得します。
21	public int set_TCTRL_SPEC_LIST()	指定 ID の ProcessingCtrlSpec を設定します。
22	public int get_ TCTRL_SPEC_LIST()	指定 ID の ProcessingCtrlSpec を取得します。
23	public int set_TPRJ_STATE_LIST()	指定 ID の PRJobStatusList を設定します。
24	public int get_ set_TPRJ_STATE_LIST()	指定 ID の PRJobStatusList を取得します。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-3 の 6.1

4. DSEng5 構成プログラムと機能概略

4. 1 プログラムモジュールの構成

主なプログラムのモジュール構成は下図のようになります。

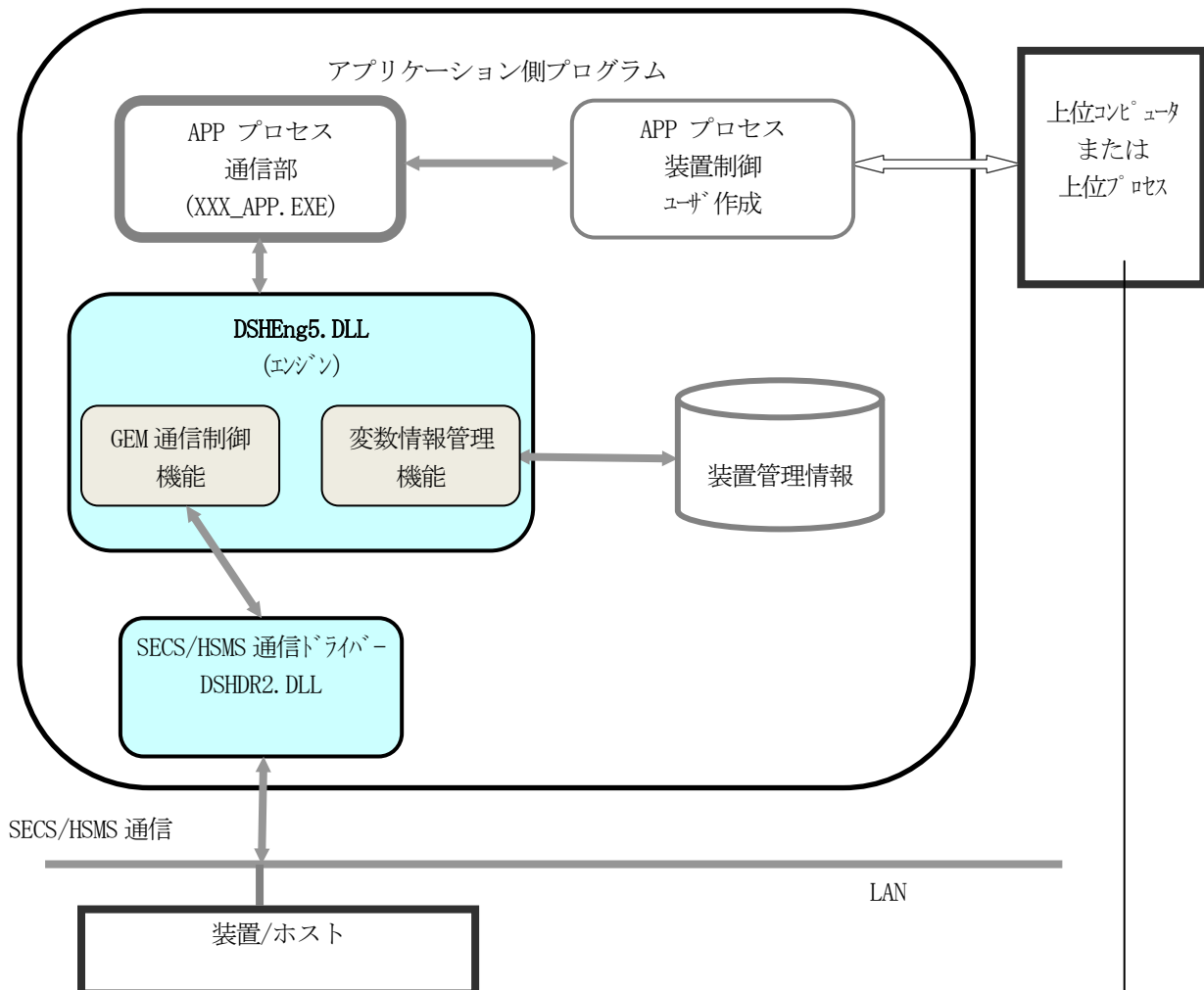


図 4 プログラムモジュール構成図

4. 2 アプリケーションプログラム

4. 1 で述べた APP 通信プロセスに位置するアプリケーションプログラムが DSHEng5 に対して行う処理に説明します。

4. 2. 1 アプリケーションプログラムの処理

ユーザは下表に示す処理を順番に行うためのプログラムを作成することになります。

順番	処理項目	使用クラス	DSHEng5 の処理
1	DSHEng5 の初期化処理	eng_class = new EngAPI() eng_class.start() DSHDR2 の comm. def, 装置構成ファイル equip. cnf バックアップファイルの復旧指定 class_EnableComm クラス 通信確立 (Enable)	(EngAPI のインスタンス) ①DSHEng5 を起動します。 ②DSHDR2 HSMS 通信ドライバの起動 ③装置関連情報を初期設定します。 (変数情報) ④装置変数定義ファイルの取込み ⑤バックアップ情報の復元が指定ならば 装置変数の復旧 ⑥S1F13, S1F14 のやり取りで通信確立処理を行います。
2	通常処理	DSHEng5 通信エンジンクラスを使って御処理	①エンジンのクラスによる管理情報のアクセスと SECS メッセージの送信処理 ②相手装置からの SECS メッセージの受信処理 ③その他
3	DSHEng5 の終了処理	eng_class.stop()	①装置の全処理を終了させます。 ②DSHDR2 HSMS 通信ドライバを停止 ③1. の初期化処理で準備したの制御情報領域 (メモリ) を開放

4. 2. 1. 1 DSEng5の初期化処理

初期化処理の内容は次の通りです。

まず、EngAPI クラスのインスタンス `eng_class` を生成します。

`eng_class.start()` メソッドを呼出すことによって行います。

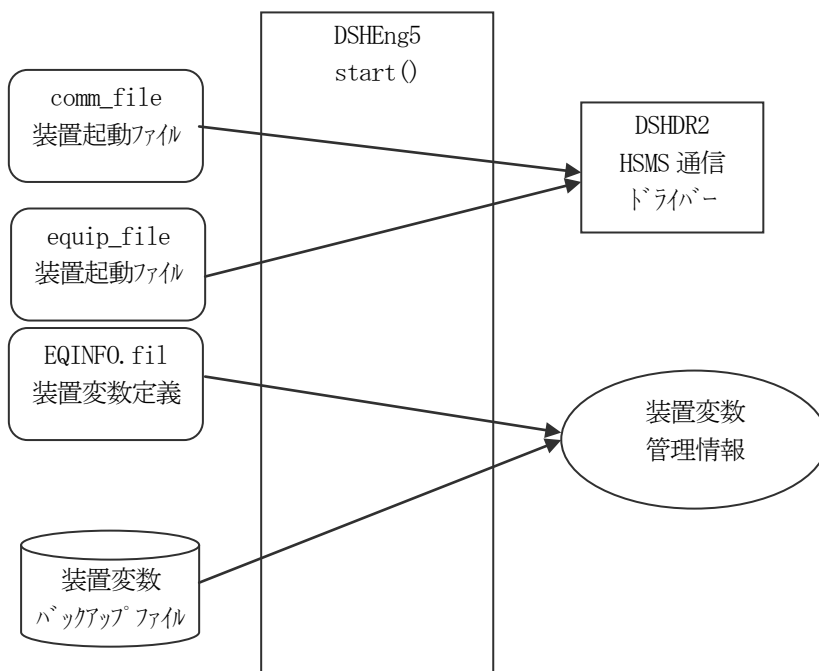
```
public int start(string equip_file, string comm_file, int bkup_flag, ref string err_str)
    equip_file   : 装置起動ファイル名(.cnf)装置制御に必要な基本情報が定義されている。
    comm_file    : 通信環境定義ファイルに従って DSHDR2 通信ドライバーを起動します。
    bkup_flag    : 前回バックアップされている管理情報を回復させるかどうかを指定します。(0/1)
```

なお、`equip_file` で指定されたファイルの中に、装置の管理情報定義ファイル名(.fil)、DSHDR2 ドライバーで装置が通信に使用するポート、デバイス番号が指定されます。

その後、装置管理情報ファイルから装置変数、イベント、アラーム情報などを DSEng5 内部に生成しセットアップします。また、DSHDR2 通信ドライバーとの通信を行うためにポート、デバイスを開きます。

処理の内容を簡略に示すと以下のような制御になります。

入力情報であるファイル名などから、矢印が示す情報のセットアップになります。



4. 2. 1. 2 装置管理情報 (変数など) アクセス処理

APP の通常処理になりますが、その1つに、管理情報のアクセスがあります。例えば、ある装置定数の現在値を取得する場合、class_EC クラスの get_value() メソッドを使って、指定した ECID が有する値を取得します。

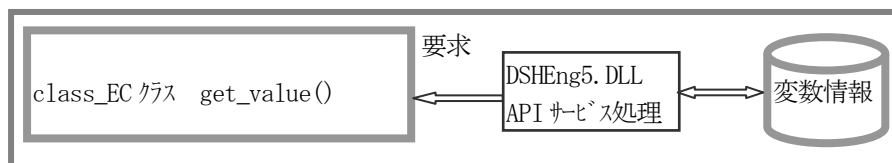
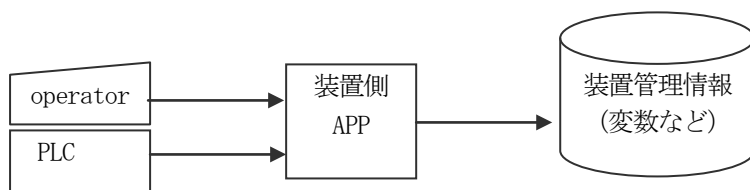


図 4. 2. 1. 2 装置管理情報アクセス処理関連図

装置変数は、DSHEng5 によって管理されていますので、APP は、DSHEng5 のクラスを使って変数情報を取得することになります。

ホスト側は、装置からの SECS-II 通信メッセージの受信によって装置状態変数あるいはデータ変数の値を更新することになります。(S1F4, S6F11 受信時など) 一方、装置側は、オペレータからの入力、あるいは、PLC などからの入出力信号によって変数の値を更新することになります。

装置変数の値の更新は、変数値設定関数を使って行います。



アクセスできる装置管理情報には以下のものがあります。

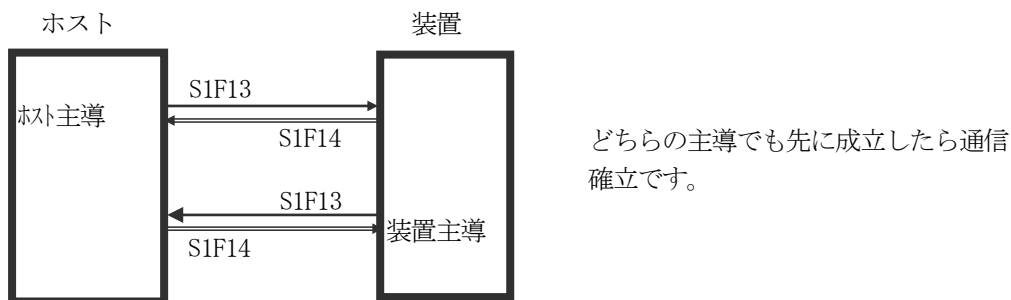
表 4. 2. 1. 2 アクセス対象装置管理情報

	情報の種類	備考
1.	装置変数	
	(1) 装置定数 (EC)	
	(2) 装置状態変数 (SV)	
	(3) データ変数 (DVVAL)	
2.	収集イベント (CE)	
3.	アラーム (ALM)	
4.	スプール (SPOOL)	
5.	トレース (TRACE)	
6.	プロセスプログラム (PP)	PP 情報使用の装置向け (S7F3)
7.	フォーマット付きプロセスプログラム (FPP)	FPP 情報使用の装置向け (S7F23)
8.	レシピ情報 (RCP)	RCP 情報使用の装置向け (S15F13)
9.	キャリア情報 (CAR)	
10.	プロセスジョブ (PRJ)	
11.	コントロールジョブ (CJ)	

4. 2. 1. 3 対装置通信の開始／通信停止要求

相手装置との通信開始は、単に class_EnableComm クラスの **Enable()** メソッドを呼出して行います

本関数によって通信状態を ENABLED にし、その後の S1F13, 14 メッセージのやり取りの成功で COMMUNICATING 状態にします。



通信停止は、**Disable()** メソッドを使って行い、装置との通信状態を DISABLE 状態にします。

相手装置との通信確立処理は、DSHEng5 が全て行います。APP は、通信状態変数の参照によって通信が確立しているかどうかを確認することになります。

通信状態が COMMUNICATING 状態でない状態で装置から S1F13 以外の 1 次メッセージを受信した場合、DSHEng5 は無条件に Function=0 のメッセージを返します。

DSHEng5 は、次の通信状態遷移図に従って、装置との通信状態制御ならびに管理を行います。

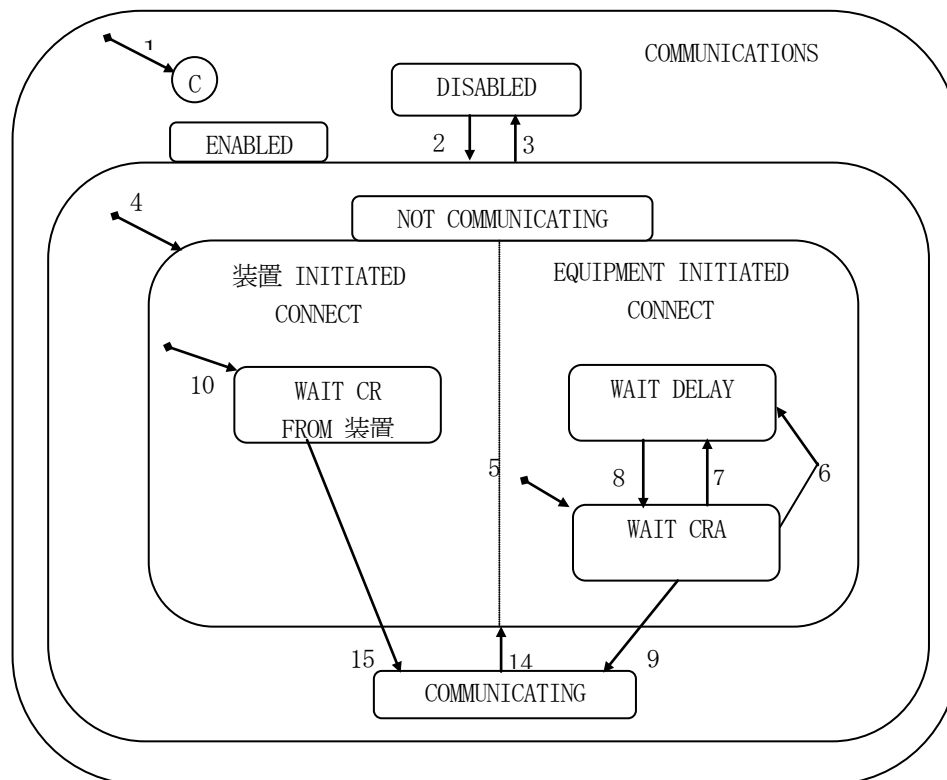


図 4. 2. 1. 3 通信状態モデル (Communication State Model)

4. 2. 1. 4 収集イベント通知処理

APP は装置状態の変化に基づいて、変数値を更新するとともに、変数が収集イベントにリンクされているケースでは、関連収集イベントをホストに通知することがあります。

収集イベントの通知は、`class_SendS6F11` クラスの `SendS6F11()` または `SendS6F11_wait()` メソッドで、引数として CEID を指定して DSHeng5 に依頼します。

(参照資料) DSHeng5 GEM 通信エンジン・クラス説明書 vol-6 の 7.4



DSHeng5 は、APP からの関数呼出しで与えられたイベント ID (CEID) の定義情報内にリンクされているレポート ID、レポート ID にリンクされている変数 ID に与えられている変数値から S6F11 メッセージを組立て、そして送信します。

APP 側のプログラミングにおいては、イベント ID を指定してクラスを実行すればよく、S6F11 のメッセージを組立てる処理をすることもなく、収集イベントメッセージを送信することができます。

S6F11 ← CEID ← LINK RPTID (複数個の場合もあり) ← LINK VID (複数個の場合もあり)

ユーザが実際に `SendS6F11()` メソッドの引数である CEID を指定するとき、装置管理情報定義ファイルのコンパイル時に得られたヘッダファイル (ex. `eng_id.cs`) に定義された CEID の名前を使うことができます。

例えば、オンライン切替の時に CE 名が `CE_Online` で、CEID の値が =100 の場合、次のようにイベントを通知することができます。

① `eng_id.cs` ファイルには、次のように定数定義されます。

(`EQ_INFO.txt` のコンパイルでファイルが生成されます)

```
public const uint CE_ControlState = 100;
```

② 次のどちらの表現でもオンライン切替イベントを通知することができます。

```
class_SendS6F11.SendS6F11(CE_Online, CE_callback, upara );
```

```
class_SendS6F11.SendS6F11(100, CE_callback, upara );
```

収集イベント関連で、ホストからの以下のメッセージによる指示、要求に対し、DSHeng5 が自動的に要求された処理を行い応答メッセージを作成し、送信してくれます。(APP が関与しなくてもかまいません)

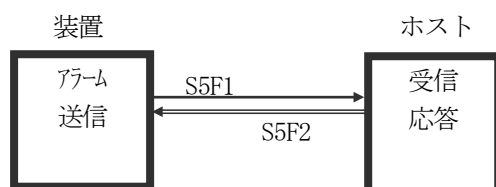
メッセージ	目的	DSHeng5 の自動処理
S2F33	レポートの定義	レポート ID にリンクされる変数 ID の設定を行う。
S2F35	リンクイベントレポート	CEID にリンクされるレポート ID の設定を行う。
S2F37	有効・無効イベントレポート	CEID の有効/無効の設定を行う。
S6F15	イベントレポート要求	指定された CEID のレポート情報を応答する。(S6F11 と同内容)
S6F19	個別レポート要求	指定された RPID にリンクされている変数の値を応答する。

4. 2. 1. 5 アラーム通知処理

APP は装置状態の変化に基づいて変数値を更新しますが、その中で変数値が異常で作業者あるいは装置の安全に関わる場合、ホストにアラーム通知を行う必要があります。

ホストへのアラームの通知は、アラーム ID(ALID)と発生/復旧の識別を引数に指定して classSendS5F1 クラスの SendS5F1 () メソッドを呼出して行います。

(参照資料) DSHEng5 GEM 通信エンジン・クラス説明書 vol-6 の 8.1



DSHEng5 はクラスの引数で与えられたアラーム ID (ALID) の定義情報から、ALCD, ALTX の値を取り出し、S5F1 メッセージに組込んだ上でアラーム通知を行います。

また、もし ALID にリンクされている収集イベント (CEID) があれば、その収集イベント通知 (S6F11) も行います。

ユーザは、APP のプログラミングをする上で、S5F1 のメッセージ構造を意識することなく、また、メッセージの組立て処理をすることもなく、アラームメッセージを送信することができます。

ユーザが実際に class_SendS5F1.SendS5F1 () メソッドの引数である ALID を指定するときは、装置管理情報定義ファイルのコンパイラ時に得られたソースファイル (ENG_DEF.cs) 上に定義された ALID の名前を使うことができます。

例えば、圧力 A オーバーのアラームが発生したとして、名前が AL_PressureAOver で ALID の値が = 230 の場合、次のようにホストにアラームを通知することができます。

① eng_id.cs には、次のように定義されます。

```
public const uint AL_AlarmPressure_1_Low = 101;
```

② 次のどちらの表現でもこのアラームを通知することができます。

```
class_SendS5F1.SendS5F1 (AL_AlarmPressure_1_Low, 1, AL_callback, upara );
```

```
class_SendS5F1.SendS5F (101, 1, AL_callback, upara );
```

3 番目の引数が発生/復旧のフラグです。(1=発生、0=復旧)

アラーム関連で、ホストからの以下のメッセージによる指示、要求に対し、DSHEng5 が自動的に要求された処理を行い応答メッセージを作成し、送信してくれます。

メッセージ	目的	DSHGEMLIB の自動処理
S5F3	アラーム有効・無効の設定	指定された ALID の有効/無効の設定
S5F6	アラームリストの要求	指定された ALID の ALARM 一覧情報を応答する。

4. 2. 1. 6 APP が処理する通信メッセージの登録と処理

初期化時に APP は通信確立後装置から送信されてくる 1 次メッセージの中で APP が直接処理したいメッセージのメッセージ ID (Stream, Function) を DSHEng5 に伝えておきます。

APP は、配信してもらう 1 次メッセージの指定し、`class_TPRI_PASS` クラスの `set_stream_func()` メソッドを使って DSHEng5 に登録します。

DSHEng5 は、装置から APP によって指定されたメッセージを受信した場合、このメッセージを APP に渡します。

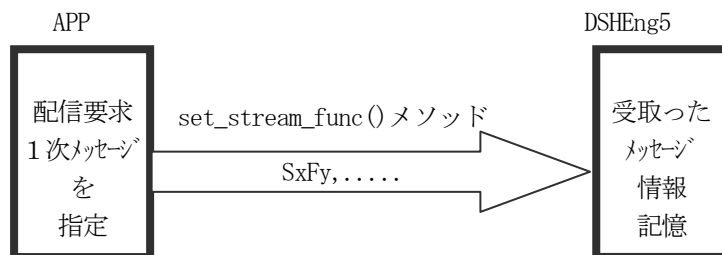


図 4. 2. 1. 6 APP への配信メッセージの登録

APP は DSHEng5 から渡されたメッセージをデコードし、然るべき処理を行います。

デコードのためのクラスについては、DSHEng5 GEM 通信エンジン・クラス説明書 vol-6 を参照ください。

(DSHEng5 がサポートする 1 次、2 次メッセージの Encode / Decode クラス (`class_SxFy`) についての説明があります)

処理が終わった後、APP は受取ったメッセージに対する 2 次メッセージを DSHEng5 ライブラリ関数を使って装置に応答送信します。

応答送信には、DSHEng5 が提供する

DSHEng5 は、APP が直接処理する 1 次メッセージのほとんどについて、APP ができるだけシンプルに処理できるように、前処理と後処理を行うためのクラスとメソッドを用意しています。

1 次メッセージに含まれている情報をプログラムが処理しやすい構造体内にデコードしたり、応答情報が格納されている構造体のデータから応答メッセージをエンコードしたりするためのライブラリ関数です。

装置から受信し、APP に配信する 1 次メッセージは下図の矢印の方向に渡され処理されます。

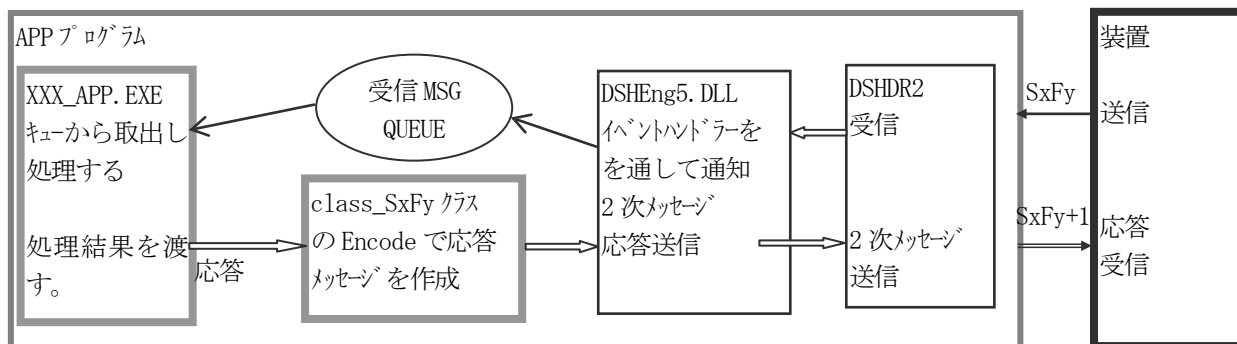


図 4. 2. 1. 6 APP 配信 1 次メッセージの処理関連図

DSHEng5 からユーザへの 1 次メッセージの配信と処理方法は、簡単にまとめると以下のようになります。

- (1) ユーザは予め、装置からの受信メッセージの中で、APP が直接処理をしたいメッセージを DSHEng5 に登録しておきます。また、ユーザは受信メッセージを受け取るためのイベントハンドラー DSHEng5 に渡しておきます。
- (2) DSHEng5 は、相手装置からユーザ向けメッセージを受取ったら、(1) のイベントハンドラーを呼出し、受信メッセージをユーザに渡します。
- (3) ユーザーのイベントハンドラーでは、自身で生成した受信メッセージキューに受信したメッセージ情報をエンキューします。
- (4) 一方、ユーザは周期的に、受信キューをポーリングし、受信メッセージを取出すことができたなら、その受信メッセージの処理を行います。

処理にあたって、DSHMSG 構造体に含まれるメッセージをデコードすることによって、そのメッセージのために準備されているクラスのインスタンスに保存します。

- (5) 受信メッセージの処理終了後、応答メッセージを作り、送信することになります。応答情報をクラスのインスタンスに設定し、その情報からメッセージにエンコードして SECS-II メッセージを作成し、DSHMSG 構造体に保存します。
- (6) 応答メッセージを送信します。 EngAPI. SendResponse () メソッドを使用します。

(参照資料)

参照事項	参照資料 (説明書)
メッセージの Encode/Decode	DSHEng5 GEM 通信エンジン・クラス説明書 vol-6 SECS-II メッセージ別にクラス名とメソッドの説明があります。
応答メッセージの送信	DSHEng5 GEM 通信エンジン・クラス説明書 vol-4 の 18.2 DSHMSG 構造体に保存された応答メッセージを送信します。

4. 2. 2 で更に詳しく説明します。

4. 2. 1. 7 1次メッセージの送信

ユーザは、アプリケーションプログラム内で、装置に対して任意の SECS-II の 1 次メッセージを作成し、DSHEng5 と DSHDR2 を介して装置に送信することができます。(DSHEng5 がサポートするメッセージの送信は大体 DSHEng5 のクラスとして準備されていますので、ユーザはクラスの方法呼出しで簡単に送信できます。)

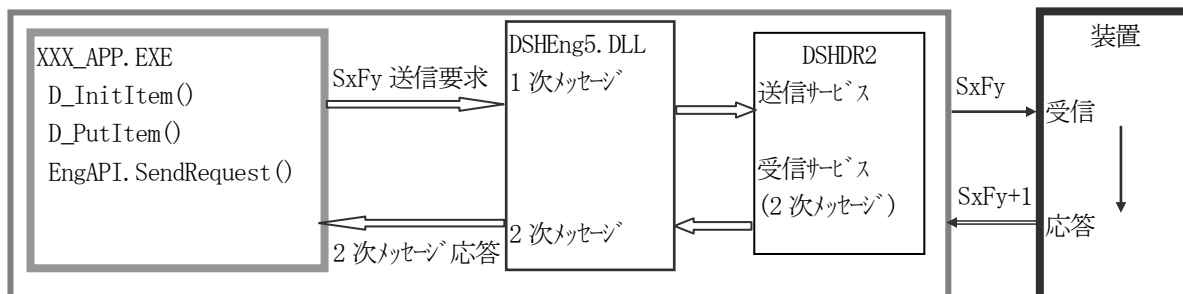
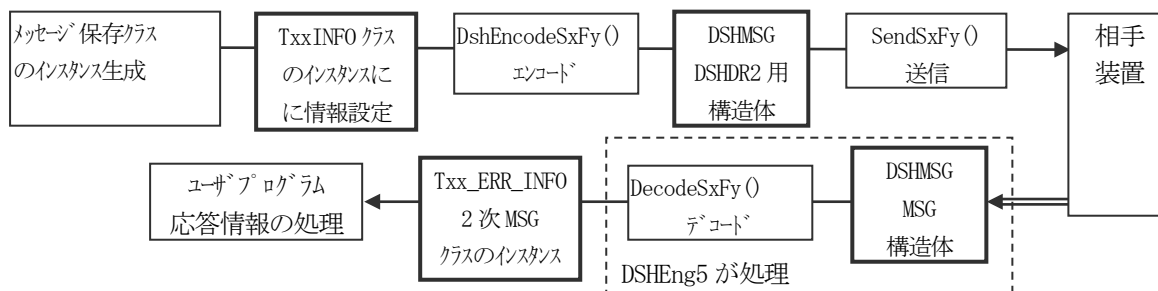


図 4. 2. 1. 7 APP による 1 次メッセージ送信処理関連図

APP は以下の DSHDR2 通信ドライバークラスならびに DSHEng5 クラスを使用することができます。

- (1) DSHDR2 HSMS 通信ドライバークラス
 メッセージ組立て関数 : D_InitItemPut(), D_PutItem()
 (通常の GEM メッセージについてはそれぞれエンコード関数が準備されています。このメッセージ組み立て関数を使用するのは DSHEng5 がサポートしていないメッセージのケースになります。)
- (2) DSHEng5 がサポートしていないメッセージの送信
 クラス : EngAPI, メソッド : SendRequest()
 (注) XXX_APP.EXE 内で直接 DSHDR2 の D_SendRequest() を使ってメッセージを送信することはできません。
- (3) DSHEng5 がサポートするメッセージ送信専用のクラス
 S7F3をはじめとしてS6F11などのGEMで規定されている基本的なメッセージについてはメッセージ毎に専用クラスが準備されています。
 class_SendS7F3.SendS7F3(), class_SendS6F11.SendS6F11() メソッドなど。
- (4) 1次メッセージ生成のためのエンコード関数、2次応答メッセージのエンコード関数が準備されています。
 メッセージに含めたい、あるいは含まれる情報をクラスのインスタンス内に保存し、その後、送信メッセージの生成、応答メッセージの処理を行います



4. 2. 1. 8 DSEng5 の終了処理

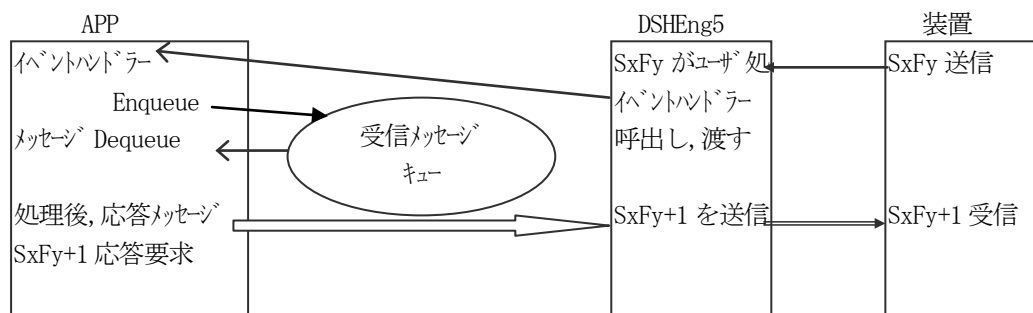
DSEng5 の終了処理は、EngAPI クラスの stop() メソッドを使って行います。

本関数によって以下の処理を行います。

- (1) 装置の HSMS 通信を停止させます。(DSHDR2 ドライバーの停止)
- (2) DSEng5 が抱えている装置管理変数情報をバックアップファイルに保存します。
- (3) DSEng5 が装置管理に使用していたメモリを全て開放します。
- (4) ログファイルを閉じます。

4. 2. 2 ユーザによる受信1次メッセージの処理とクラス

装置に送信されてくる1次メッセージのいくつかはユーザプログラム自身で処理する必要のあるものがあります。DSHEng5は、ユーザが処理したい1次メッセージをシンプルな方法でDSHEng5から受け取りそして処理できるようにするための仕組みをユーザに提供します。



- (1) ユーザは予めユーザが処理したい1次メッセージをDSHEng5に登録しておきます。また、DSHEng5がユーザ向けメッセージを受信した後、それを受取るためのイベントハンドラーを渡しておきます。
- (2) DSHEng5はユーザ向けメッセージを受信したら、(1)で指定されたイベントハンドラーに受信メッセージを呼び出してユーザに渡します。ユーザは、イベントハンドラーの中で、受信したメッセージを受信メッセージキューに格納します。
- (3) 一方、ユーザは、受信メッセージ処理のためのスレッドを作り、そのスレッドで受信メッセージキューをポーリングしながら受信メッセージを取り出し、処理します。
- (4) 処理が終わったら、応答メッセージを作り、DSHEng5に送信します。

DSHEng5は、ユーザがAPPプログラムの中でメッセージをできるだけ効率よく処理するために様々なライブラリ関数を提供します。下の図は、ポーリングで得られたメッセージのAPPプログラムでの処理と順番を示しています。

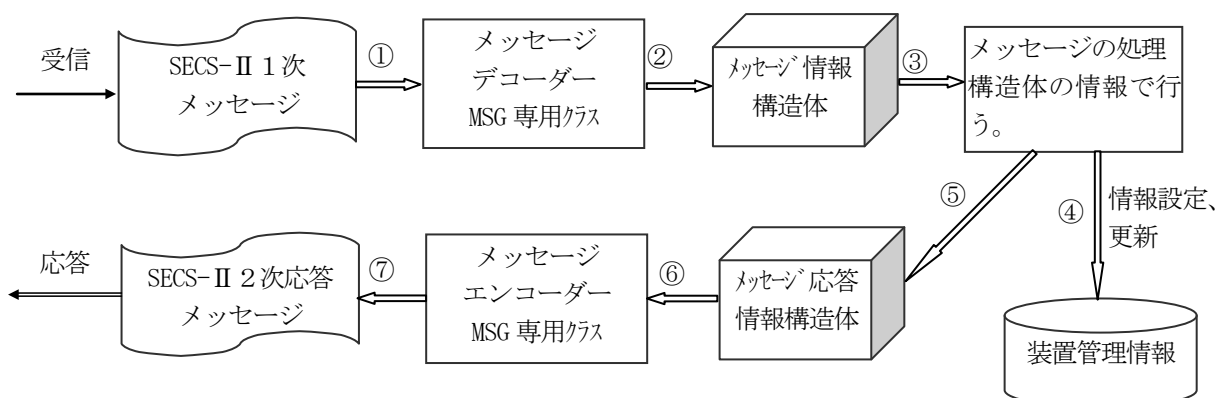


図 4.2.2 処理の流れとデコード、エンコードメッセージ

メッセージのデコード、エンコードに使用される関数については4. 2. 2. 2で示します。

④の装置管理情報のアクセスはDSHEng5のクラスを使って行います。

4. 2. 2. 1 ユーザが処理する受信1次メッセージの登録

ユーザは、自身で処理した受信1次メッセージのメッセージIDをDSHEng5に登録しておきます。

DSHEng5は装置/ホストの両方の処理機能を有しており、そのどちらか一方の機能を選択して通信制御を行います。

ユーザ処理メッセージの登録は、DSHEng5のclass_TPRI_PASSクラスのset_stream_func()メソッドを使って行います。

1次メッセージの登録のためのコーディングは、例えば、次のように行います。

```
public int set_SECS2_msg()
{
    int ei = 0;
    while (true)          // 1個づつ設定します。
    {                    // >>> この内容をシステムの通信仕様によって変更してください
        if ((ei = class_TPRI_PASS.set_stream_func(1, 15)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(1, 17)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(2, 17)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(2, 23)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(2, 41)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(2, 43)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(2, 45)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(2, 49)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(3, 17)) != 0) break;
        (途中省略)
        if ((ei = class_TPRI_PASS.set_stream_func(15, 3)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(15, 5)) != 0) break;
        if ((ei = class_TPRI_PASS.set_stream_func(15, 13)) != 0) break;
        break;
    }
    return ei;
}
```

図 4. 2. 2. 1 APP への配信1次メッセージのソースファイルへの登録例

ユーザはここに載っている以外のメッセージをこのソースに追加することができます。また、APPが処理する必要のないものはソースファイルから削除することもできます。

DSHEng5は受信メッセージキューからのメッセージ情報の取り出し手段と、メッセージを処理した後に応答メッセージを装置に送るための関数を提供します。

応答メッセージ送信処理用関数は次の一覧表に示されるソースファイル上に準備されています。

ユーザは、このファイル上のプログラムをそのまま使用することができます。また、処理を追加することもできます。

表 4.2.2.1 装置側デフォルト登録メッセージ一覧

番号	メッセージ	メッセージ用途		
1.	S1F15	オフライン要求		
2.	S1F17	オンライン要求		
3.	S2F41	ホストコメント送信		
4.	S2F43	スケジュールの設定		
5.	S2F45	変数リミット属性定義		
6.	S2F49	Enhanced Remote Command		
7.	S3F17	キャリアアクション要求		
8.	S3F23	ポートグループアクション要求		
9.	S3F25	ポートアクション要求		
10.	S3F27	Change Access		
11.	S7F1	プロセスプログラムポート問合せ		
12.	S7F3	プロセスプログラム送信		
13.	S7F23	フォーマット付プロセスプログラム送信		
14.	S10F3	端末表示、シングルロック		
15.	S10F5	端末表示、マルチロック		
16.	S14F9	Create Object Request (Cj)		
17.	S14F11	Delete Object Request (Cj)		
18.	S15F3	Recipe Namespace action Req		
19.	S15F5	Recipe Namespace rename Req		
20.	S15F13	Recipe Create Request		
21.	S16F5	Process Job Cmd Request		
22.	S16F11	PrJob Create Enh		
23.	S16F15	PrJob Multi Create		
24.	S16F17	PrJob Deque		
25.	S16F27	Control Job Command Request		

表 4.2.2.2 ホスト側デフォルト登録メッセージ一覧

番号	メッセージ	メッセージ用途		
1.	S5F1	アラーム報告送信		
2.	S6F11	イベントレポート送信		
3.	S6F13	注釈付イベントレポート送信		
4.	S7F1	プロセスプログラムポート問合せ		
5.	S7F3	プロセスプログラム送信		
6.	S7F23	フォーマット付プロセスプログラム送信		
7.	S7F25	フォーマット付プロセスプログラム要求		
8.	S7F27	プロセスプログラム妥当性送信		
9.	S10F1	端末要求		
10.	S15F3	レシピネームスペースアクション要求		
11.	S15F5	レシピネームスペースリネーム要求		
12.	S15F13	レシピ生成要求		

4. 2. 2. 2 メッセージのエンコード / デコードクラス

DSHeng5 は、APP が受信 1 次メッセージを処理するための Encode / Decode クラスを提供します。
 メッセージ情報の構造体へのデコード、そして構造体からのメッセージへのエンコードなどのための関数です。

	class	msg ID	Encode method	Decode method	情報保存 class / ack
1	class_S1F3	S1F3	EncodeS1F3 ()	DecodeS1F3 ()	TVID_LIST
		S1F4	EncodeS1F4 ()	DecodeS1F4 ()	TVID_LIST, TV_VALUE_LIST
2	class_S1F11	S1F11	EncodeS1F11 ()	DecodeS1F11 ()	TVID_LIST
		S1F12	EncodeS1F12 ()	DecodeS1F12 ()	TVID_LIST, TSV_NAME_LIST
3	class_S1F13	S1F13	EncodeS1F13 ()	DecodeS1F13 ()	(EQ side)
		S1F14	EncodeS1F14 ()	DecodeS1F14 ()	
4	class_S1F15	S1F15	-	-	-
		S1F16	EncodeS1F16 ()	DecodeS1F16 ()	ack
5	class_S1F17	S1F17	-	-	-
		S1F18	EncodeS1F18 ()	DecodeS1F18 ()	ack
6	class_S2F13	S2F13	EncodeS2F13 ()	DecodeS2F13 ()	
		S2F14	EncodeS2F14 ()	DecodeS2F14 ()	ack
7	class_S2F15	S2F15	EncodeS2F15 ()	DecodeS2F15 ()	TVID_VAL_LIST
		S2F16	EncodeS2F16 ()	DecodeS2F16 ()	ack
8	class_S2F17	S2F17	EncodeS2F17 ()	DecodeS2F17 ()	-
		S2F18	EncodeS2F18 ()	DecodeS2F18 ()	string(日付、時刻)
9	class_S2F23	S2F23	EncodeS2F23 ()	DecodeS2F23 ()	TTRACE_INFO
		S2F24	EncodeS2F24 ()	DecodeS2F24 ()	ack
10	class_S2F29	S2F29	EncodeS2F29 ()	DecodeS2F29 ()	TVID_LIST
		S2F30	EncodeS2F30 ()	DecodeS2F30 ()	TVID_LIST, TEC_NAME_LIST
11	class_S2F31	S2F31	EncodeS2F31 ()	DecodeS2F31 ()	string(日付、時刻)
		S2F32	EncodeS2F32 ()	DecodeS2F32 ()	ack
12	class_S2F33	S2F33	EncodeS2F33 ()	DecodeS2F33 ()	TRP_LIST
		S2F34	EncodeS2F34 ()	DecodeS2F34 ()	ack
13	class_S2F35	S2F35	EncodeS2F35 ()	DecodeS2F35 ()	TRP_LIST
		S2F36	EncodeS2F36 ()	DecodeS2F36 ()	ack
14	class_S2F37	S2F37	EncodeS2F37 ()	DecodeS2F37 ()	TEDER_INFO
		S2F38	EncodeS2F38 ()	DecodeS2F38 ()	ack
15	class_S2F41	S2F41	EncodeS2F41 ()	DecodeS2F41 ()	TRCMD_INFO
		S2F42	EncodeS2F42 ()	DecodeS2F42 ()	TRCMD_ERR_INFO
16	class_S2F43	S2F43	EncodeS2F43 ()	DecodeS2F43 ()	TSPool_INFO
		S2F44	EncodeS2F44 ()	DecodeS2F44 ()	TSPool_ERR_INFO
17	class_S2F45	S2F45	EncodeS2F45 ()	DecodeS2F45 ()	TLIMIT_LIST
		S2F46	EncodeS2F46 ()	DecodeS2F46 ()	TLIMIT_ERR_LIST
18	class_S2F47	S2F47	EncodeS2F47 ()	DecodeS2F47 ()	TVID_LIST
		S2F48	EncodeS2F48 ()	DecodeS2F48 ()	TLIMIT_RSP_LIST
19	class_S2F49	S2F49	EncodeS2F49 ()	DecodeS2F49 ()	TERCMD_INFO
		S2F48	EncodeS2F48 ()	DecodeS2F48 ()	TRCMD_ERR_INFO

20	class_S3F17	S3F17	EncodeS3F17 ()	DecodeS3F17 ()	TCACT_INFO
		S3F18	EncodeS3F18 ()	DecodeS3F18 ()	TCACT_ERR_INFO
21	class_S3F23	S3F23	EncodeS3F23 ()	DecodeS3F23 ()	TPORTG_INFO
		S3F24	EncodeS3F24 ()	DecodeS3F24 ()	TCACT_ERR_INFO
22	class_S3F25	S3F25	EncodeS3F25 ()	DecodeS3F25 ()	TPORT_INFO
		S3F26	EncodeS3F26 ()	DecodeS3F26 ()	TCACT_ERR_INFO
23	class_S3F27	S3F27	EncodeS3F27 ()	DecodeS3F27 ()	TACCESS_INFO
		S3F28	EncodeS3F28 ()	DecodeS3F28 ()	TACCESS_ERR_INFO
24	class_S5F1	S5F1	EncodeS5F1 ()	DecodeS5F1 ()	TAL_INFO
		S5F2	EncodeS5F2 ()	DecodeS5F2 ()	ack
25	class_S5F3	S5F3	EncodeS5F3 ()	DecodeS5F3 ()	TAL_S5F3_INFO
		S5F4	EncodeS5F4 ()	DecodeS5F4 ()	ack
26	class_S5F5	S5F5	EncodeS5F5 ()	DecodeS5F5 ()	TAL_S5F5_INFO
		S5F6	EncodeS5F6 ()	DecodeS5F6 ()	TAL_S5F6_LIST
27	class_S5F7	S5F7	EncodeS5F7 ()	DecodeS5F7 ()	-
		S5F8	EncodeS5F8 ()	DecodeS5F8 ()	TAL_S5F6_LIST
28	class_S6F1	S6F1	EncodeS6F1 ()	DecodeS6F1 ()	TTRACE_MON
		S6F2	EncodeS6F2 ()	DecodeS6F2 ()	ack
29	class_S6F11	S6F11	EncodeS6F11 ()	DecodeS6F11 ()	ceid
		S6F12	EncodeS6F12 ()	DecodeS6F12 ()	ack
30	class_S6F15	S6F15	EncodeS6F15 ()	DecodeS6F15 ()	ceid
		S6F16	EncodeS6F16 ()	DecodeS6F16 ()	TCE_CONTENT
31	class_S6F19	S6F19	EncodeS6F19 ()	DecodeS6F19 ()	rpil
		S6F20	EncodeS6F20 ()	DecodeS6F20 ()	TRP_CONTENT
32	class_S6F23	S6F23	EncodeS6F23 ()	DecodeS6F23 ()	rsdc
		S6F24	EncodeS6F24 ()	DecodeS6F24 ()	rsda
33	class_S7F1	S7F1	EncodeS7F1 ()	DecodeS7F1 ()	TPPINQ_INFO
		S7F2	EncodeS7F2 ()	DecodeS7F2 ()	ack
34	class_S7F3	S7F3	EncodeS7F3 ()	DecodeS7F3 ()	ppid
		S7F4	EncodeS7F4 ()	DecodeS7F4 ()	TPP_INFO
35	class_S7F17	S7F17	EncodeS7F17 ()	DecodeS7F17 ()	TPPID_LIST
		S7F18	EncodeS7F18 ()	DecodeS7F18 ()	ack
36	class_S7F19	S7F19	EncodeS7F19 ()	DecodeS7F19 ()	-
		S7F20	EncodeS7F20 ()	DecodeS7F20 ()	TPPID_LIST
37	class_S7F23	S7F23	EncodeS7F23 ()	DecodeS7F23 ()	TFPP_INFO
		S7F24	EncodeS7F24 ()	DecodeS7F24 ()	ack
38	class_S7F25	S7F25	EncodeS7F25 ()	DecodeS7F25 ()	fppid
		S7F26	EncodeS7F26 ()	DecodeS7F26 ()	TFPP_INFO
39	class_S7F27	S7F27	EncodeS7F27 ()	DecodeS7F27 ()	TPP_PVS_LIST
		S7F28	EncodeS7F28 ()	DecodeS7F28 ()	ack
40	class_S7F29	S7F29	EncodeS7F29 ()	DecodeS7F29 ()	lenth
		S7F30	EncodeS7F30 ()	DecodeS7F30 ()	ack
41	class_S10F1	S10F1	EncodeS10F1 ()	DecodeS10F1 ()	tid, text
		S10F2	EncodeS10F2 ()	DecodeS10F2 ()	ack
42	class_S10F3	S10F3	EncodeS10F3 ()	DecodeS10F3 ()	tid, text
		S10F4	EncodeS10F4 ()	DecodeS10F4 ()	ack

43	class_S10F5	S10F5	EncodeS10F5 ()	DecodeS10F5 ()	TTERM_MULTI_INFO
		S10F6	EncodeS10F6 ()	DecodeS10F6 ()	ack
44	class_S14F9	S14F9	EncodeS14F9 ()	DecodeS14F9 ()	TCJ_INFO
		S14F10	EncodeS14F10 ()	DecodeS14F10 ()	TOBJ_S14_ERR_INFO
45	class_S14F11	S14F11	EncodeS14F11 ()	DecodeS14F11 ()	TCJ_INFO
		S14F12	EncodeS14F12 ()	DecodeS14F12 ()	TOBJ_S14_ERR_INFO
46	class_S15F1	S15F1	EncodeS15F1 ()	DecodeS15F1 ()	-
		S15F2	EncodeS15F2 ()	DecodeS15F2 ()	int rmgmnt
47	class_S15F3	S15F3	EncodeS15F3 ()	DecodeS15F3 ()	TRCP_ACT_INFO
		S15F4	EncodeS15F4 ()	DecodeS15F4 ()	TRCP_ERR_INFO
48	class_S15F5	S15F5	EncodeS15F5 ()	DecodeS15F5 ()	TRCP_RENAME_INFO
		S15F6	EncodeS15F6 ()	DecodeS15F6 ()	TRCP_ERR_INFO
49	class_S15F7	S15F7	EncodeS15F7 ()	DecodeS15F7 ()	string
		S15F8	EncodeS15F8 ()	DecodeS15F8 ()	TRCP_S15F8_INFO
50	class_S15F9	S15F9	EncodeS15F9 ()	DecodeS15F9 ()	string objspec
		S15F10	EncodeS15F10 ()	DecodeS15F10 ()	TRCP_S15F10_INFO
51	class_S15F13	S15F13	EncodeS15F13 ()	DecodeS15F13 ()	TRCP_INFO
		S15F14	EncodeS15F14 ()	DecodeS15F14 ()	TRCP_ERR_INFO
52	class_S15F17	S15F17	EncodeS15F17 ()	DecodeS15F17 ()	string rpid
		S15F18	EncodeS15F18 ()	DecodeS15F18 ()	TRCP_S15F18_INFO
53	class_S16F5	S16F5	EncodeS16F5 ()	DecodeS16F5 ()	TPRJ_CMD_INFO
		S16F6	EncodeS16F6 ()	DecodeS16F6 ()	TPRJ_ERR_INFO
54	class_S16F11	S16F11	EncodeS16F11 ()	DecodeS16F11 ()	TPRJ_INFO
		S16F12	EncodeS16F12 ()	DecodeS16F12 ()	TPRJ_ERR_INFO
55	class_S16F15	S16F15	EncodeS16F15 ()	DecodeS16F15 ()	TPRJ_LIST
		S16F16	EncodeS16F16 ()	DecodeS16F16 ()	TMPRJ_ERR_INFO
56	class_S16F17	S16F17	EncodeS16F17 ()	DecodeS16F17 ()	TPRJ_DEQ_INFO
		S16F18	EncodeS16F18 ()	DecodeS16F18 ()	TMPRJ_ERR_INFO
57	class_S16F19	S16F19	EncodeS16F19 ()	DecodeS16F19 ()	-
		S16F20	EncodeS16F20 ()	DecodeS16F20 ()	TPRJ_STATE_LIST
58	class_S16F21	S16F21	EncodeS16F21 ()	DecodeS16F21 ()	-
		S16F22	EncodeS16F22 ()	DecodeS16F22 ()	UInt16 pr_j_space
59	class_S16F27	S16F27	EncodeS16F27 ()	DecodeS16F27 ()	TCJ_CMD_INFO
		S16F28	EncodeS16F28 ()	DecodeS16F28 ()	TCJ_CMD_ERR_INFO

4. 2. 2. 3 1次メッセージ受信処理の流れ

DSHDR2 通信ドライバーが受信した1次メッセージはイベントハンドラーを介して受信メッセージ情報キューに格納されます。

ユーザ PP プログラムは、キューから受信メッセージ情報を取り出し、処理することになります。

実際のメッセージ処理に関連する全体の処理の流れは概略次のようになります

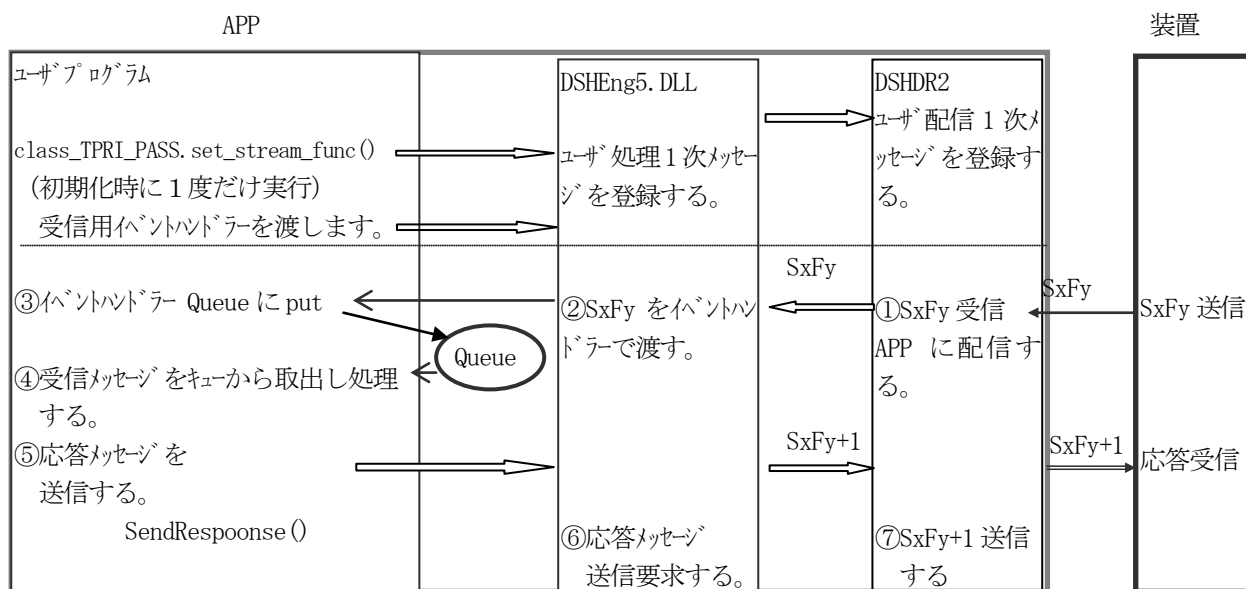
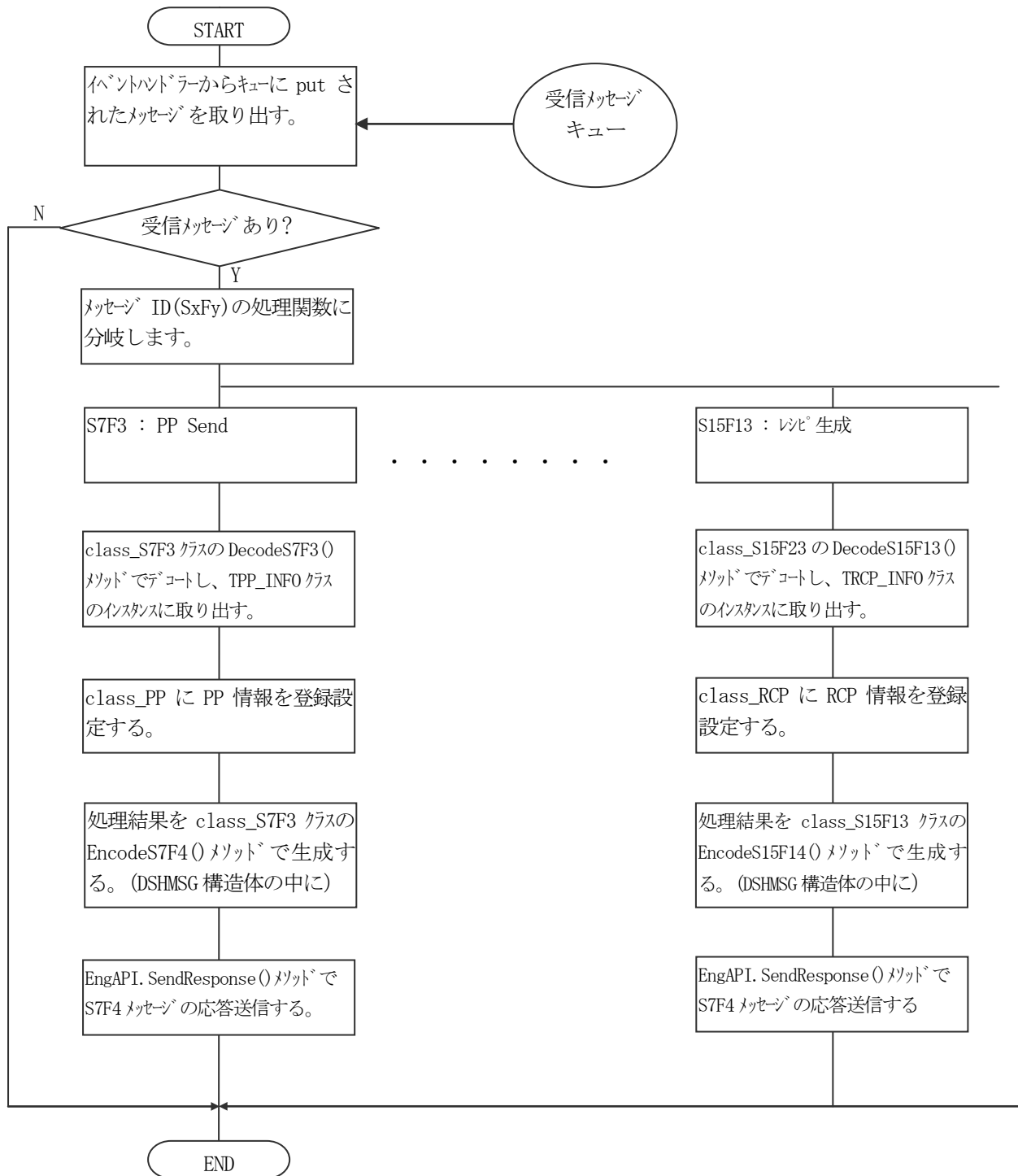


図 4. 2. 2. 3 APP の受信1次メッセージ処理

- (1) APP が処理したい1次メッセージの登録と受信メッセージ受取用のイベントハンドラーの設定
DSHEng5 を起動した後、APP は class_TPRI_PASS.set_stream_func() メソッドを使って APP が受取り処理したいメッセージ群を DSHEng5 に登録しておきます。(4. 2. 2. 1 参照)
また、メッセージ受信用イベントハンドラーを DSHEng5 に渡しておきます。
- (2) 相手装置 (ホスト) から送信されてきた1次メッセージの処理
 - ① DSHEng5 は相手装置からのメッセージが (1) で登録されたものであれば APP から指定されたイベントハンドラーを通してメッセージ情報を APP に渡します。
 - ② イベントハンドラーは、受け取った受信メッセージ情報を Queue に入れます。
 - ③ その後 APP の受信メッセージを処理するプログラムがキューから受信メッセージを取り出し処理します。
処理は、メッセージをデコードする DSHEng5 クラスを使ってメッセージ情報をメッセージ専用クラスのインスタンスに取得し処理を行います。(前節 4. 2. 2. 2 参照)
 - ④ メッセージの処理終了後相手装置に対し送信する応答メッセージの応答情報を作成します。
 - ⑤ 各メッセージの応答メッセージを生成するために設けられたクラスのエンコードメソッドを使って応答メッセージを作成し送信します。
応答メッセージは与えられた ACK を含めた応答情報から応答メッセージを組み立て、DSHEng5 に応答メッセージ送信を要求します。
 - ⑥ DSHEng5 は⑤の要求に基づいて装置に対し応答メッセージを送信します。
これで1次メッセージの処理が終了することになります。

(3) ユーザ側受信処理の流れは、例えば次のフローチャートのようにになります。



4. 2. 3 DSHEng5 通信エンジンプログラム

DSHEng5 通信エンジンは APP プログラムによって使用されるクラス群であり、内部で装置管理情報を管理します。以下は、装置側での機能図です。

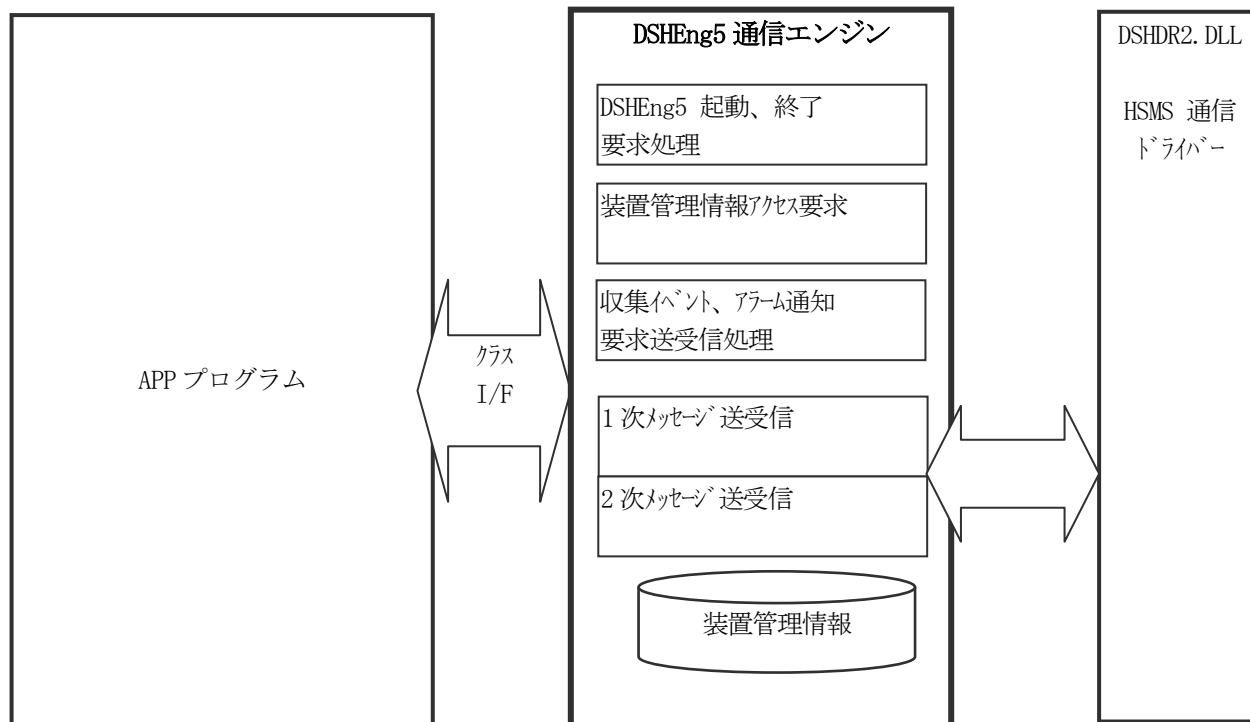


図 4. 2. 3 DSHEng5 の機能図

DSHEng5 のクラスは以下の目的のために使用することができます。

- (1) APP からの DSHEng5 の起動、終了要求処理を行います。
- (2) APP と DSHEng5 との間でクラスを介して装置管理情報交換を行います。
 - ①装置管理情報のアクセス
 - ②装置への各種問合せならびに要求指令の通知など
- (3) APP は 4. 2. 2 で述べた DSHEng5 から提供される通信メッセージ関連情報の処理を行うクラス群を提供します。
 - ①DSHEng5 から渡された 1 次メッセージのポーリング関数
 - ② 1 次メッセージに対する応答 2 次メッセージの DSHEng5 への送信要求
 - ③1 次メッセージのデコード、2 次メッセージへのエンコード
 - ④メッセージ情報構造体の生成、解放、複製などのための関数

ユーザへ提供する各種クラスについての詳しい内容は、**GEM 通信エンジン・クラス説明書(Vo1-1~6)** を参照してください。

4. 3 DSEng5 通信エンジン構成プログラムの機能

DSEng5 通信エンジンプログラムは下図のように構成されています。

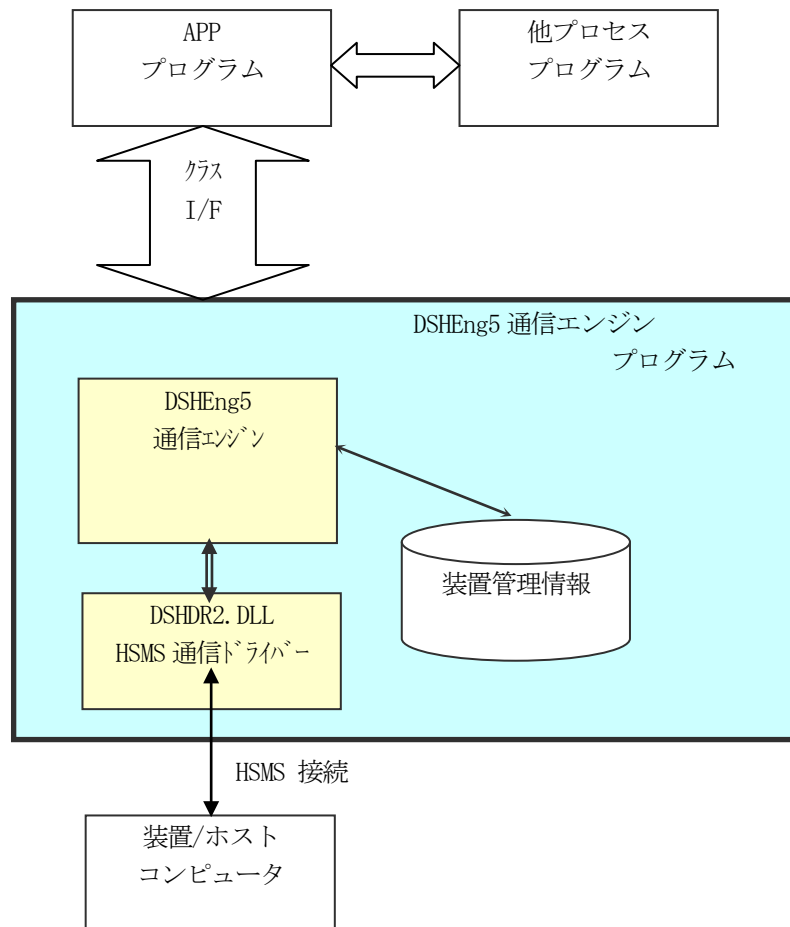


図 4.3 DSEng5 通信エンジンの構成と位置

4. 3. 1 DSHEng5 GEM 通信エンジン

本プログラムは、対装置通信サービスならびに装置管理情報の管理を行います。

先に説明しましたが、ここでは、更に詳しく説明します。

ユーザは基本的に以下の処理を行います。

- ・ 初期化処理
- ・ 通常処理
- ・ 終了処理

4. 3. 1. 1 起動時の処理

APP (アプリケーション) が、DSHEng5 のメイン・クラスである EngAPI クラスの start() メソッドを使って起動と初期化処理を行います。

start() メソッドの呼び出しで行います。

```
public int start(string sys_conf, string comm_file, int restore_bkup, ref string err_msg)
    sys_conf      : 装置起動情報定義ファイル名 (.cnf) 装置制御に必要な基本情報が定義されている。
    comm_file     : HSMS 通信定義ファイルです。
    restore_bkup  : 前回バックアップされている管理情報を回復させるかどうかを指定します。(0/1)
    err_msg       : 検出したエラーメッセージ
```

DSHEng5 は EngStart() の呼出しを受けて、与えられた comm_file, sys_conf, restore_bkup の指定に従って、以下の処理を行います。

- ① 装置を管理するための情報領域を準備します。
V(EC, SV, DVVAL)、CE、REPORT、ALARM など装置情報領域
- ② ログ、バックアップ、スプール関連ディレクトリの準備
- ③ 装置変数定義情報の読み込み
- ④ バックアップ情報の復旧指定があれば、復旧処理をします。
- ⑤ HSMS 通信のための準備

以上の処理結果を APP に返却します。

(1) 管理領域の準備

変数、収集イベント、レポート、アラーム情報ならびに、sys_conf ファイル内に指定されている設定値に基づいて、管理情報領域を準備します。設定例を示します。

PP_COUNT = 100	SUBST_COUNT = 250
FPP_COUNT = 64	PRJ_COUNT = 30
RCP_COUNT = 200	CJ_COUNT = 32
CAR_COUNT = 80	TRACE_COUNT = 28
CAR_CAPACITY = 25	

(2) ログ、バックアップ、スプール関連ディレクトリの準備

sysconf ファイル内の指定に基づいてディレクトリの有無を調べ、存在していなければディレクトリを生成します。また、指定された名前のログファイルをオープンします。

```
BKUP_PATH = "c:\%DSHEng5%backup"  
SPOOL_PATH = "c:\%DSHEng5%spool"  
LOG_PATH = "c:\%DSHEng5%log"  
LOG_FILE = "equip.log"
```

(3) 装置変数定義情報の読み込み

sysconf ファイル内に指定された装置変数定義ファイルの内容に従って 変数 (EC, SV, DVVAL)、CE、REPORT、ALARM などの定義情報に従って管理情報領域を設けて内容を登録します。登録する情報は、ID、名前、それからその変数が有する属性の値です。

```
INFO_FILE = "c:\%DSHEng5%cnf%eqinfo.fil"
```

ID の検索にはハッシュ機能を使用します。

(4) バックアップ情報の復旧

resore_bkup の指定が =1 に指定されたい場合、DSHEng5 は、(2) で、バックアップディレクトリに指定された保存場所に各管理情報のバックアップファイルを検索し、存在すれば、そのバックアップファイルに保存されている情報を管理情報領域に復元します。バックアップ情報については、**付録-C**を参照してください。

(5) HSMS 通信のための準備

相手装置との HSMS-SS 通信は、DSHDR2.DLL 通信ドライバーを介して行います。まず、comm_file 通信環境定義ファイルに従って DSHDR2 通信ドライバーをスタートします。そして、sysconf に指定されたファイル内に指定されているポートとデバイスを開きます。ポートが開かれると、通信ドライバーは相手装置との HSMS 通信接続処理を開始します。

```
COMM_PORT = 1  
COMM_DEVICE = 1
```

次頁には、DSHEng5 の予約変数の設定について説明します。

EngStart() メソッドによる DSHEng5 の起動が正常に終了した後、次に、DSHEng5 で予約されている変数の ID を DSHEng5 に対して行います。収集イベント(CE)、装置定数 (EC) ならびに装置状態変数(SV) の予約変数について、ID をどの予約変数にするかを設定します。

設定は、**class_Reserved_V** クラスのメソッドを使用します。

詳しくは、「GEM 通信エンジン・クラス説明書 Vol-1 5.1」を参照ください。

① 収集イベント(CE)

set_reserved_CEID() メソッドを使って、次の予約 CEID を登録します。

index 値	マクロ名	収集イベント
0	CEX_RSV_COMMUNICATING	ホストと通信確立時に通知するイベント ID
1	CEX_RSV_SPOOL_END	スプール送信の終了時に通知するイベント ID
2	CEX_RSV_LIMIT	変数リミット監視におけるホスト通知用イベント ID
3		
4		

② 装置定数(EC)

set_reserved_ECID() メソッドを使って、次の予約 EC を登録します。

index 値	マクロ名	装置定数
0	ECX_RSV_MDLN	S1F14 で使用する装置モデル名
1	ECX_RSV_SOFTREV	S1F14 で使用するソフトウェアバージョンコード
2	ECX_RSV_SPOOL_MAX	最大スプール数
3	ECX_RSV_SPOOL_OVERWRITE	スプールのプール数
4	ECX_RSV_INIT_COMMSTATE_	エンジン起動時の自動通信 Enable の指定用変数

③ 装置状態変数(SV)

set_reserved_SVID() メソッドを使って、次の予約 SV を登録します。

index 値	マクロ名	装置状態変数
0	SVX_RSV_CLOCK	システムの日付時刻変数(DSHEng5 が値を更新する)
1	SVX_RSV_COMMUNICATING	通信状態
2	SVX_RSV_SPOOL_STATE	スプール状態
3	SVX_RSV_SPOOL_TOTAL	スプール合計
4	SVX_RSV_SPOOL_ACTUAL	実スプール数(貯えられた)
5	SVX_RSV_SPOOL_STIME	スプール開始時刻
6	SVX_RSV_SPOOL_FTIME	スプール満杯時刻
7	SVX_RSV_LIMIT_V	リミット監視対象変数 ID
8	SVX_RSV_LIMIT_DVVAL	同変数値(文字列)
9	SVX_RSV_LIMIT_ID	同リミット ID
10	SVX_RSV_LIMIT_DIR	同遷移方向

また、ユーザが処理対象にしたい SECS-II 受信 1 次メッセージの登録を行う必要があります。
class_TPRI_PASS クラスの set_stream_func() メソッドを使って行います。
詳しくは、「GEM 通信エンジン・クラス説明書 Vol-1 4.3」を参照ください。

4. 3. 1. 2 通常処理

以下の通常処理を行います。

(1) APP からの装置管理情報のアクセスサービス

前述の 表 4. 2. 1. 2 アクセス対象管理情報に記載されている情報サービスを行います。

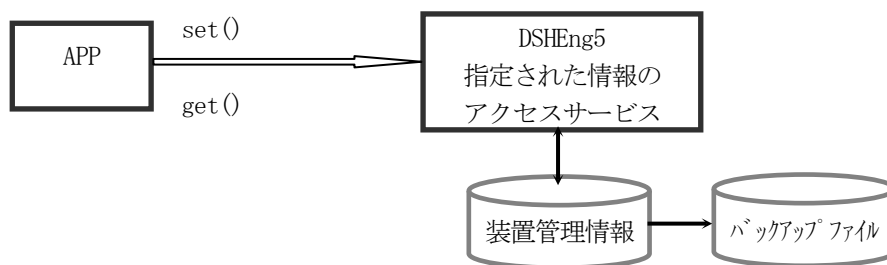


図 4. 3. 1. 2-1 DSHEng5 の通常処理の関連図

(2) 収集イベント通知の処理

4. 2. 1. 4の収集イベント通知処理の説明を参照ください。

(3) アラーム通知の処理

4. 2. 1. 5のアラーム通知処理の説明を参照ください。

(4) 受信1次メッセージのAPP への配信処理

予め APP から指定されていたメッセージを装置から受信したときにそれを APP 側に配信します。
APP から渡される応答メッセージの送信も行います



図 4. 3. 1. 2-4 DSHEng5 の1次メッセージのAPP への配信関連図

(5) APP からの1次メッセージ送信要求処理

APP から要求される1次メッセージの送信と2次メッセージの受信サービスを行います。



図 4. 3. 1. 2-5 DSHEng5 のAPP からの1次メッセージ送信要求処理の関連図

(6) APP の処理対象でない 1 次メッセージは DSHEng5 が自動処理します。

APP 側から配信指定されていないメッセージで DSHEng5 が処理できるものを受信した際、DSHEng5 は自動的に処理します。

対象になるメッセージは装置からの装置管理情報の設定、参照要求に応えるもの、例えば、S7F5 のように DSHEng5 に自動応答することができるメッセージです。

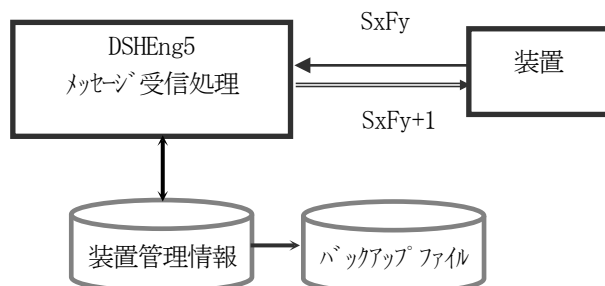


図 4. 3. 1. 2-6 DSHEng5 受信 1 次メッセージの内部自動処理の関連図

4. 3. 1. 3 終了処理

DSHEng5 の終了は、APP からの次の EngAPI クラスのインスタンスのメソッドの呼出しで行われます。

(1) stop() メソッドを使って DSHEng5 を終了させます。

DSHEng5 は、以下の処理を行います。

- ①相手装置との HSMS 通信を停止させます。各装置が使用していたポート、デバイスの停止です。
- ②装置変数情報をバックアップします。
- ③装置の管理情報用に使用していた情報領域を開放します。
- ④ログファイルを閉じます。

4. 3. 2 HSMS 通信ドライバー (DSHDR2.DLL)

HSMS-SS の通信ドライバーです。

通信プロトコルの制御と管理を行います。

DSHEng5 のユーザは、DSHDR2 の `D_SendRequest()`、`D_SendResponse()`、`D_Receive()` 送受信関数を直接使用することはできません。(送受信は、DSHEng5 のクラスを通しておこないます。)

DSHEng5 がサポートしていない例外的な 1 次メッセージの送信は、`EngSendPrimary()` メソッドを使って送信することになります。

詳しくは、SECS/HSMS レベル 2 通信ドライバーの説明書を参照してください。

5. 個別機能

5. 1 状態管理機能

通信確立状態ならびに装置コントロール状態について管理します。

5. 1. 1 通信状態モデル

通信状態は、以下 (1)、(2) で記述する状態モデルに従って管理を行います。

APP は通信開始時に、Enable() メソッドを使って DSHEng5 に対し通信開始を指令します。指令を受けた DSHEng5 はそれ以降の通信開始処理に関するすべてを行います。

通信停止は、APP からの Disable() メソッドを使って行います。

APP は、装置とホストとの間で通信状態が確立したかどうかを **SV_CommunicationState** 状態変数の値を参照することによって判断することができます。(値が、ST_COMMUNICATION になっていれば通信確立状態です。)

(SVID=SV_CommunicationState を引数にして classSV の get_value() メソッドを使って値を取得します。)

(1) 状態遷移図

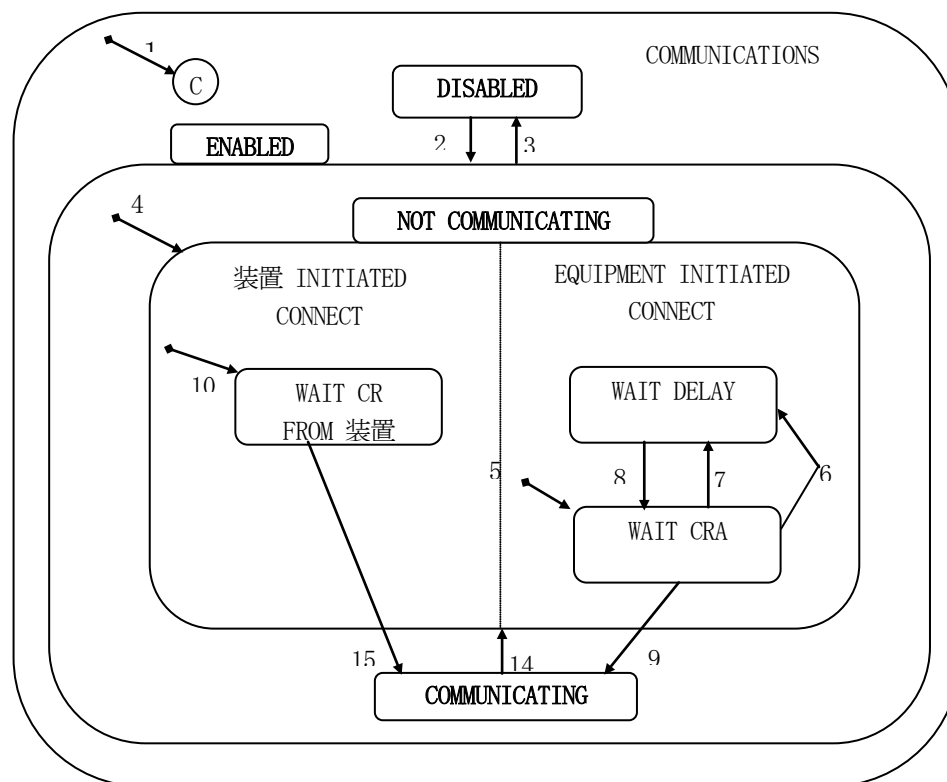


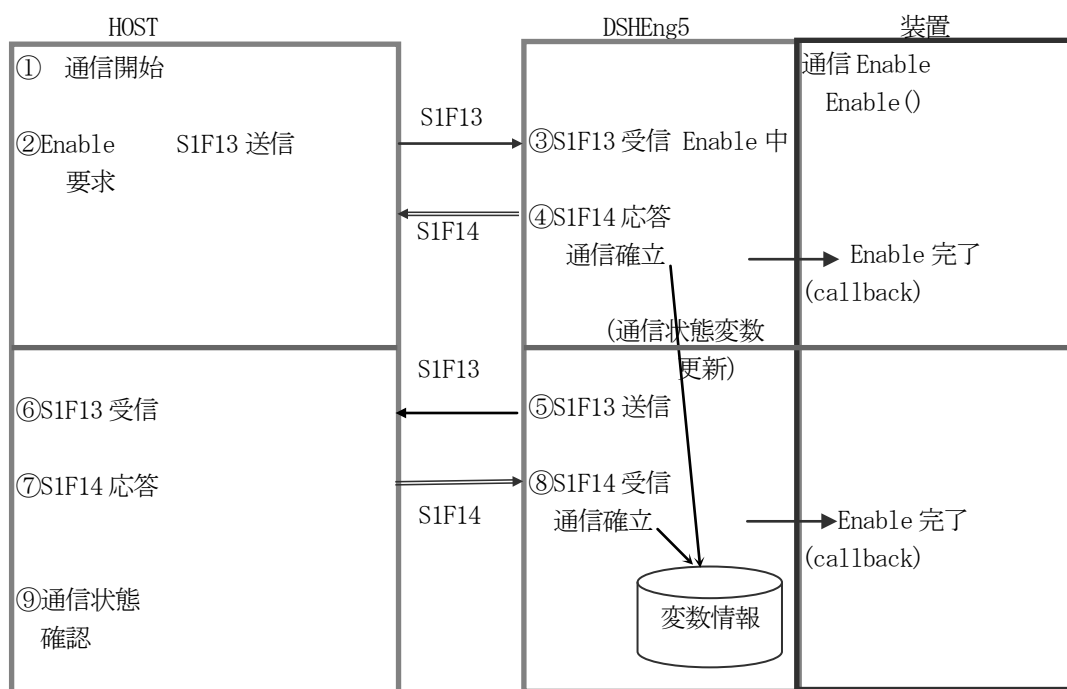
図 5.1.1 通信状態遷移図 (Communication State Model)

(2) 状態遷移定義

表 5.1.2 通信状態遷移定義表

#	現在の状態	トリガー	新しい状態	動作	コメント
1	(通信実行に入る)	システムの初期化	システムデフォルト	なし	システムのデフォルトは Enabled or Disabled に設定される。
2	DISABLED (通信無効)	オペレータが通信有効に切り替える	ENABLED (通信有効)	なし	SECS-II 通信が有効になる。
3	ENABLED (通信有効)	オペレータが通信無効に切り替える	DISABLED (通信無効)	なし	SECS-II 通信が禁止される。
4	(通信有効に入る)	通信有効状態に入る	NOTCOMMUNICATING (通信中断)	なし	システムの初期状態から通信有効に入ってもよいし、オペレータが通信有効に切り替えてもよい。
5	(装置開始接続に入る)	(通信中断状態に入る)	WAITCRA (通信確立要求確認待ち)	通信初期化。 CommDelay タイマを時間切れにセット。S1F13 を送信)	通信確立開始。
6	WAITCRA (通信確立要求確認待ち)	通信トランザクションの失敗	WAITDELAY (遅延タイマタイムアウト待ち)	CommDelay タイマを初期化する。送信するために入れておいたメッセージを全てキューから出す。)	適切な場合にはキューからだされたメッセージは生成順にスプールバッファに入れる。タイマが時間切れになるのを待つ。
7	WAITDELAY (遅延タイマタイムアウト待ち)	通信遅延タイマがタイムアウトになる。	WAITCRA (通信確立要求確認待ち)	S1F13 送信	S1F14 を待つ。ホストからの S1F13 を受けることもできる。
8	WAITDELAY (遅延タイマタイムアウト待ち)	S1F13 以外のメッセージを受信する。	WAITCRA (通信確立要求確認待ち)	メッセージを捨てる。応答はしない。CommDelay タイマを時間切れにセット。(S1F13 を送信)	通信を確立するチャンスがあることを意味する。
9	WAITCRA (通信確立要求確認待ち)	待っていた COMMACK=0 の S1F14 を受信する。	COMMUNICATING (通信実行)	なし	通信が確立される。
10	(ホスト開始接続に入る)	(通信中断状態に入る。)	WAITCRFROM 装置 (ホストからの通信確立待ち)	なし	ホストからの S1F13 を待つ。
11	WAITCRFROM 装置 (ホストからの通信確立待ち)	S1F13 受信	WAITTXCOMPLETE (通信確立完了待ち)	COMMACK=0 の S1F14 を送信	ホストからの通信確立。
12	WAITTXCOMPLETE (通信確立完了待ち)	S1F14 の送信に失敗した。	WAITCRFROM 装置 (ホストからの通信確立待ち)	なし	ホストからの S1F13 を待つ。
13	WAITTXCOMPLETE (通信確立完了待ち)	S1F14 の送信に成功した。	COMMUNICATING (通信実行)	なし	通信が確立した。
14	COMMUNICATING (通信実行)	通信の失敗	NOTCOMMUNICATING (通信中断)	送信のためにキューに入っていたメッセージを全てデキューする。	デキューされたメッセージは必要に応じてスプールされる。

(3) 通信状態の管理と処理の流れ



注) ③の S1F13 はホスト主導、⑤の S1F13 は装置主導
どちらかの通信が成立すれば通信確立となる。

図 5.1.1-1 通信状態管理に関する処理の流れ

[通信状態制御用クラス一覧]

使用するクラスは `class_EnableComm` で、以下のメソッドを使用します。

表 5.1.1 通信状態制御用クラス

メッセージ	目的	使用するメソッド
S1F13	通信確立	Enable()
-	通信確立処理中止	CancelEnable()
-	通信 Disable	Disable()

詳しくは、「GEM 通信エンジン・クラス説明書 Vol-1 4.2」を参照ください。

5. 1. 2 コントロール状態の管理

装置のコントロール状態は、以下の（１）、（２）の状態モデルによって、ユーザAPPが主体になって管理します。

すなわち、状態変数を設け、その値を状態遷移のイベントが発生したときに更新するようになります。

（１）状態遷移図

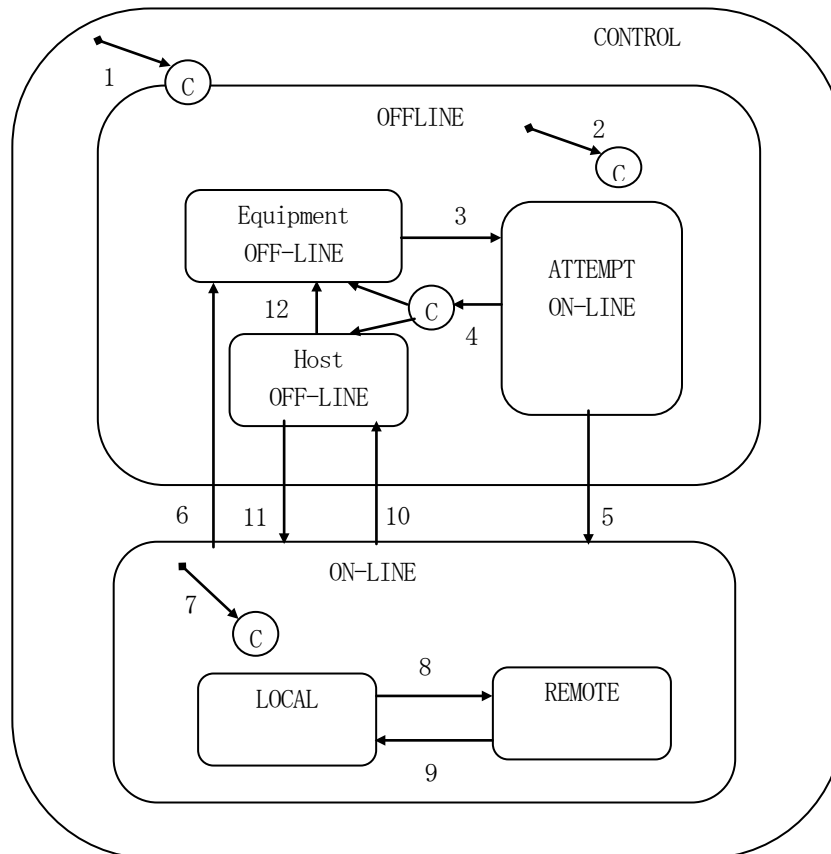


図 5.1.2 コントロール状態遷移図

(2) コントロール状態遷移定義

表 5.1.2 コントロール状態遷移表

#	現在の状態	トリガー	新しい状態	動作	コメント
1	(未定義)	コントロール状態に入る (システム立上げ)	CONTROL (下位状態は設定により異なる)	なし	装置はデフォルト設定値 ON-LINE or OFF-LINE に入る
2	(未定義)	OFF-LINE 状態に入る	OFF-LINE (下位状態は設定により異なる)	なし	装置はデフォルト設定値 OFF-LINE のどんな下位状態にもなる
3	EQUIPMENT OFF-LINE (装置オフライン)	オペレータがスイッチを ON-LINE に切り替える	ATTEMP-ONLINE オンライン試行	なし	オンライン試行状態にある ときはいつでも SIF1 が 送信されることに注意
4	ATTEMPT-ONLINE (オンライン確立試行 1)	SIF0	設定条件により異なる 新しい状態	なし	通信の喪失、返信タイムアウト、 もしくは SIF0 の受信による。 設定条件により装置オンライン、 もしくはホストオンラインに移行する。
5	ATTEMPT-ONLINE (オンライン確立試行)	装置はホストから期待した SIF2 を受信する。	ON-LINE オンライン	なし	装置は遷移 7 でオンラインに 移行することを通知される。
6	ON-LINE (オンライン)	オペレータがスイッチをオンライン に切り替える。	EQUIPMENT-OFFLINE 装置オフライン	なし	“装置オンライン” イベント発生 オンラインのとき、イベント返信 メッセージは捨てられる。
7	(未定義)	ON-LINE 状態に入る。	ONLINE (下位状態は設定により異なる)	なし	“コントロール状態ローカル” または “コントロール状態リモート” イベント発生。 イベントレポートは実際に移行した オンラインの下位状態を示す。
8	LOCAL (ローカル)	オペレータがフロントパネルの スイッチをリモートにセットする。	REMOTE リモート	なし	“コントロール状態リモート” イベント発生。
9	REMOTE (リモート)	オペレータがフロントパネルの スイッチをローカルモードにセットする。	LOCAL ローカル	なし	“コントロール状態ローカル” イベント発生。
10	ON-LINE (オンライン)	装置はオフライン切替メッセージ SIF15 をホストから受信する。	装置 OFF-LINE ホストオフライン	なし	“ホストオフライン” イベント発生
11	装置 OFF-LINE (ホストオフライン)	装置はオンライン移行要求 SIF17 を了解する。	ON-LINE オンライン	なし	装置は遷移 7 でオンラインに 移行することを通知される。
12	装置 OFF-LINE (ホストオフライン)	オペレータがスイッチをオンライン に切り替える。	EQUIPMENT-OFFLINE 装置オフライン	なし	“装置オンラインイベント” 発生。

(3) コントロール状態管理と処理の流れ

Host による処理処理例として、次の図の番号の順に行われるようになります。
S1F15 について記してありますが、S1F17 も同様です。

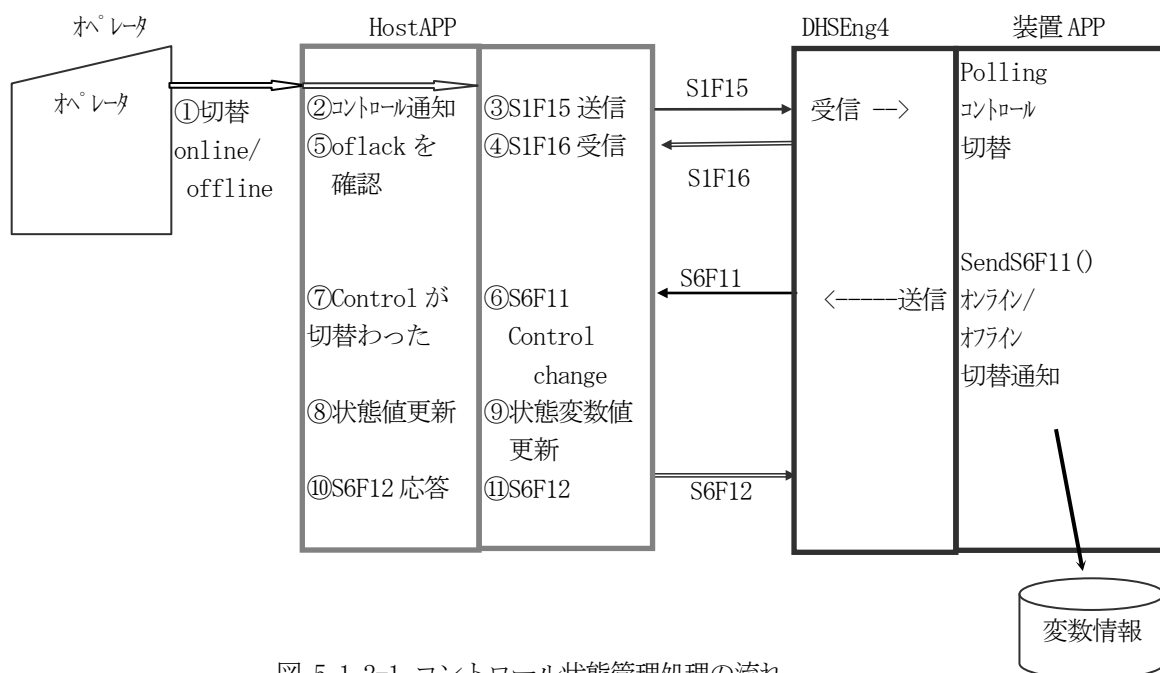


図 5.1.2-1 コントロール状態管理処理の流れ

[コントロール関連メッセージ送信用クラス一覧]

コントロールメッセージを送信するために以下のクラスがあります。(Host → 装置)

表 5.1.2 コントロール関連メッセージ

メッセージ	目的	クラス
S1F15	オンライン要求	class_SendS1F15.SendS1F15()
S1F17	オンライン要求	class_SendS1F17.SendS1F17()

5. 2 装置変数管理機能と装置への変数要求機能

装置変数として以下の種類のものがあります。

- (1) 装置定数 (EC)
- (2) 装置状態変数 (SV)
- (3) 装置データ変数 (DVVAL)

DSHEng5 はユーザに装置変数に対する以下のアクセスクラスとメソッドを提供します。関数は変数の種類別に設けられています。

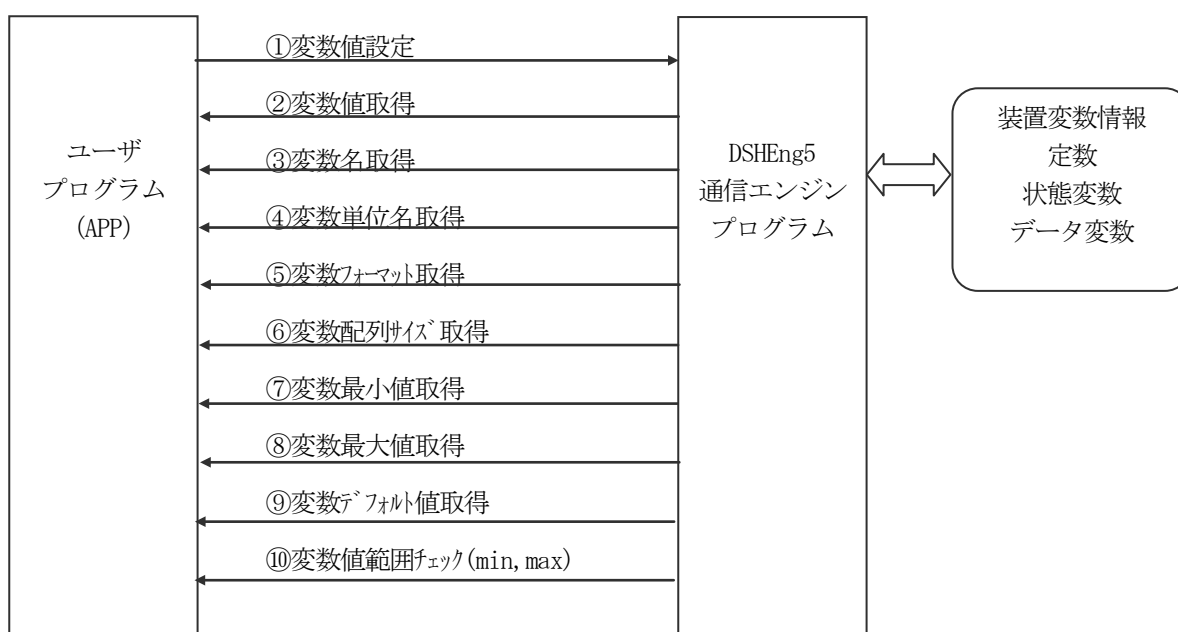


図 5.2-1 変数アクセス操作

また、変数のリミット（限界値）値の設定、参照ならびにチェッククラスも準備されています。

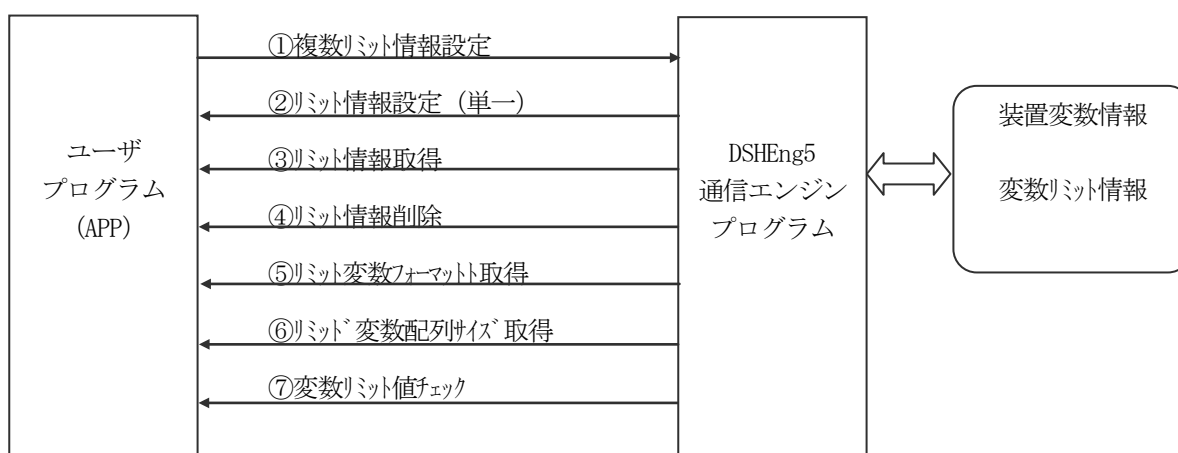


図 5.2-2 変数リミット値操作

変数アクセス関数の詳細については、DSHEng5 GEM 通信エンジン・クラス説明書 Vol-1 2 を参照してください。

[装置変数関連メッセージ送信用クラス一覧]

装置への変数要求メッセージは以下の通りです。

受信時、DSHEng5 が自動的に応答を返すメッセージもあります。

表 5.2 装置変数関連メッセージ

メッセージ	目的
S1F3	装置状態要求
S1F11	状態変数一覧要求
S2F13	装置定数要求
S2F15	装置定数変更
S2F29	装置定数名一覧要求
S2F45	変数リミット属性定義
S2F47	変数リミット属性一覧要求

5.3 収集イベント通知

収集イベント情報については3.2で説明しましたが、その準備と操作は次のように行われます。

- (1) 装置管理情報定義ファイル内に装置がホストに通知すべき収集イベントを全て定義します。
定義方法など詳しい内容は「**装置管理情報定義仕様書**」を参照してください。
ファイルに定義された情報は、DSHEng5 立上げ時の処理によってシステム内部に登録されます。
- (2) 1個の収集イベントは以下の情報によって構成されます。
 - ① 1個のイベント ID (CEID)
 - ② 0個または1個以上のレポート ID (RPTID)
 - ③ 各レポート ID にリンクされている 0 個以上の装置変数データ (VID)

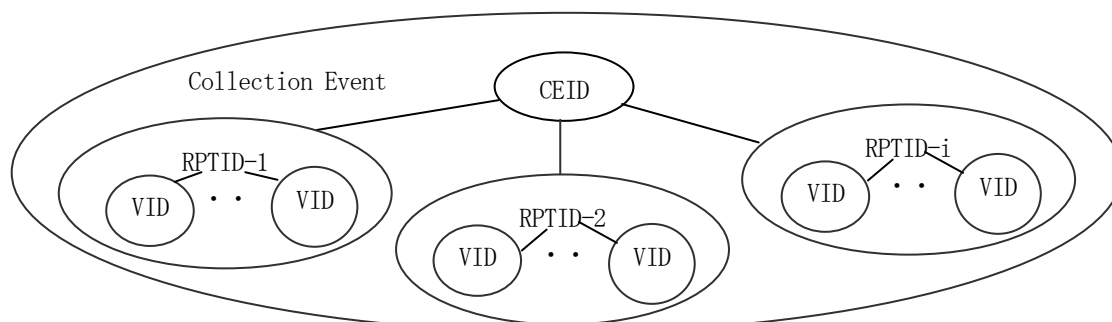


図 5.3-1 CEID, RPTID, VID の関係

装置変数の値は装置の状態の変化（外部入出力信号による）あるいはホストまたは装置オペレータの設定値変更操作によって更新されます。

- (3) 変数値変化の検出は基本的に装置側で行われ、以下のように処理されます。

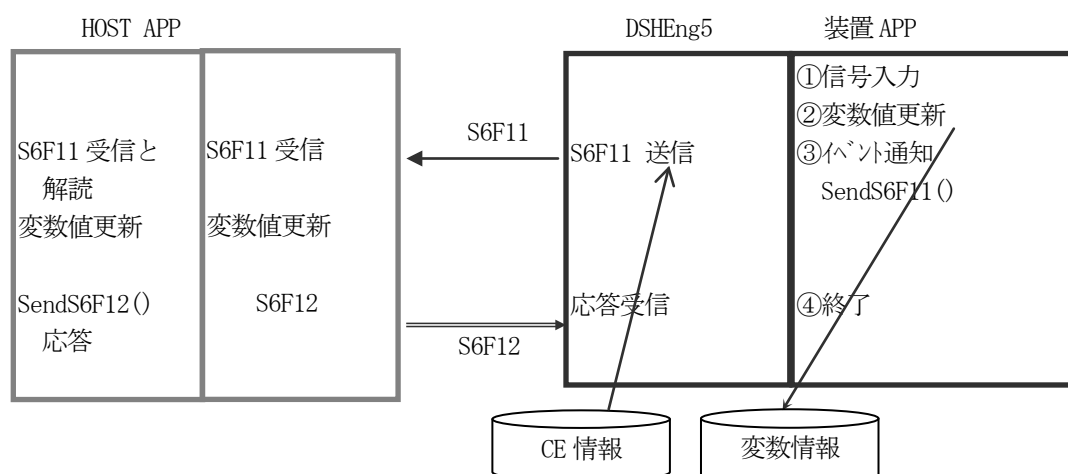
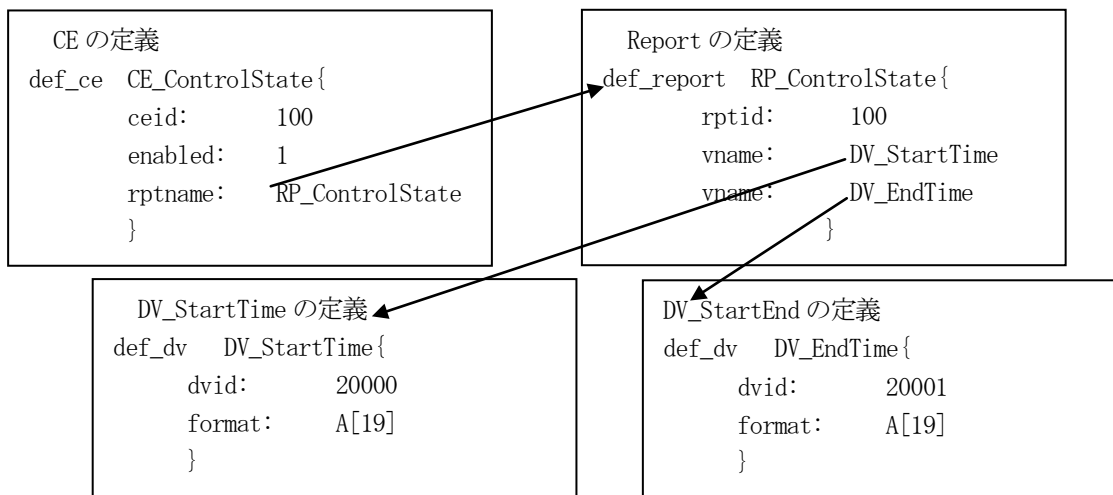


図 5.3-2 収集イベント処理の流れ

- (4) 装置は変数値の変化を検出し、その値を `set_value()` メソッドなどで更新し、その後、イベント ID を指定して `SendS6F11()` メソッドを使ってホストに S6F11 メッセージを送信します。
- (5) 要求を受取った DSHeng5 は、それを APP 側に知らせます。APP は S6F11 を受信して解釈し処理をします。DSHeng5 内に定義された CEID、レポート ID の定義情報から VID を特定し、TCE_CONTENT クラスに S6F11 メッセージ内に含まれる変数情報にデコードします。
例をあげて説明すると以下のようになります。

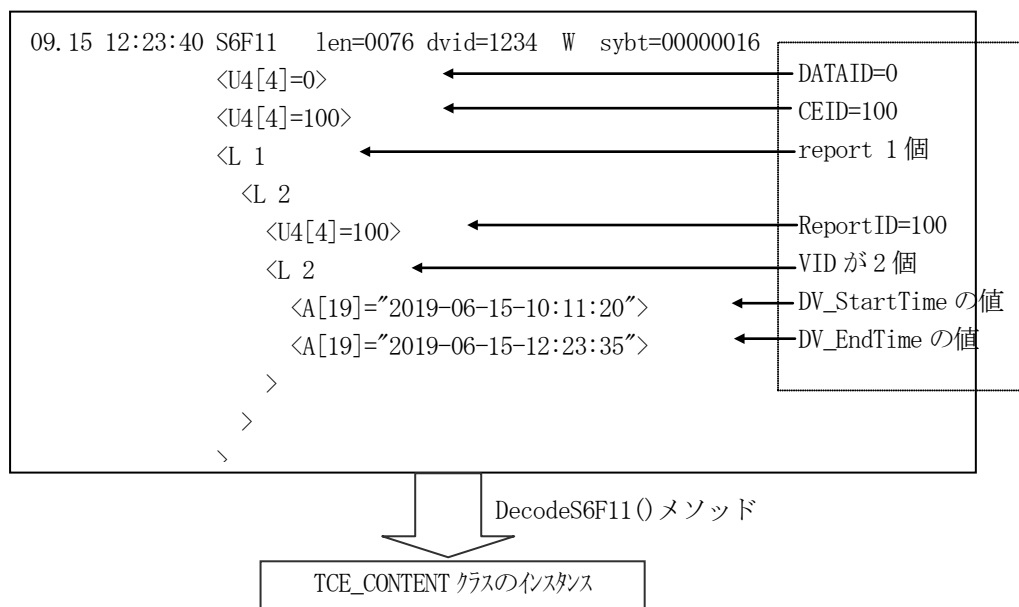


- ①CE_ControlState 収集イベントは CEID=100 であり、リンクしているレポートは RP_ControlState 1 個です
- ②RP_ControlState は RPTID=100 であり、リンクしている装置データ変数は DV_StartTime と DV_EndTime の 2 つです
- ③DV_StartTime と DV_EndTime 変数はそれぞれ DVID が 20000, 20001 であり、それぞれフォーマット-ASCII (fmt 10) で最大 19 文字の値を持ちます。
それぞれの変数の値が、次のように設定されているとします。
- ```

DV_StartTime = "2019-06-15-10:11:20"
DV_EndTime = "2019-06-15-12:23:35"

```

この状態で、装置側 DSHeng5 は収集イベント CE\_ControlState を S6F11 にエンコードし通知します。ホスト側では、その S6F11 を受信し、APP に渡します。そして、S6F11 に含まれるレポート ID、変数値を CEID 定義情報から、TCE\_CONTENT クラスのインスタンス内にデコードし、処理します。



### [収集イベント関連メッセージ送信用クラス一覧]

以下の送信クラスを使って装置に対し直接的に収集イベント情報の要求や設定を行うことができます。

表 5.3 収集イベント関連メッセージ

| メッセージ | 目的            | 使用するクラス     | ホスト(H)/装置(E) |
|-------|---------------|-------------|--------------|
| S2F33 | レポートの定義       | TRP_LIST    | H            |
| S2F35 | リンクイベントレポート   | TCE_LIST    | H            |
| S2F37 | 有効・無効イベントレポート | TEDER_INFO  | H            |
| S6F15 | イベントレポート要求    | TCE_CONTENT | H            |
| S6F19 | 個別レポート要求      | TRP_CONTENT | H            |
| S6F11 | イベントレポート送信    | SendS6F11() | E            |

## 5. 4 アラーム通知

装置コントローラから装置で起こるアラーム状態がホストに通知されてきます。DSHEng5 はアラーム情報の受信と処理を簡潔に実現するための手段を提供します。

### 5. 4. 1 アラーム状態モデル

(1) アラーム状態遷移

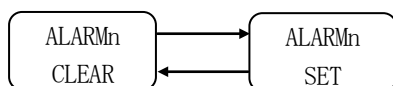


図 5. 4. 1 アラーム ALIDn についての状態図

### 5. 4. 2 アラーム処理と流れ

アラーム情報は 3. 3 で概要を説明しましたが、その準備と処理の流れは次のようになります。

- (1) 装置管理情報定義ファイル内に装置から通知されるアラーム ID とその内容を全て定義します。定義方法など詳しい内容は「**装置管理情報定義仕様書**」を参照してください。ファイルに定義された情報は、DSHEng5 立上げ時の処理によってシステム内部に登録されます。
- (2) 1 個のアラーム情報は以下の要素で構成されます。
  - ①アラーム ID (ALID)
  - ②アラームコード(ALCD)
  - ③アラームテキスト(ALTX)
- (3) 装置はアラーム通知を行う必要がある入力信号の変化を検出した時、class\_SendS5F1 クラスの SendS5F1 () メソッドを使ってホストに対しアラーム通知を行います。

処理の流れは、下の図の番号順になります。

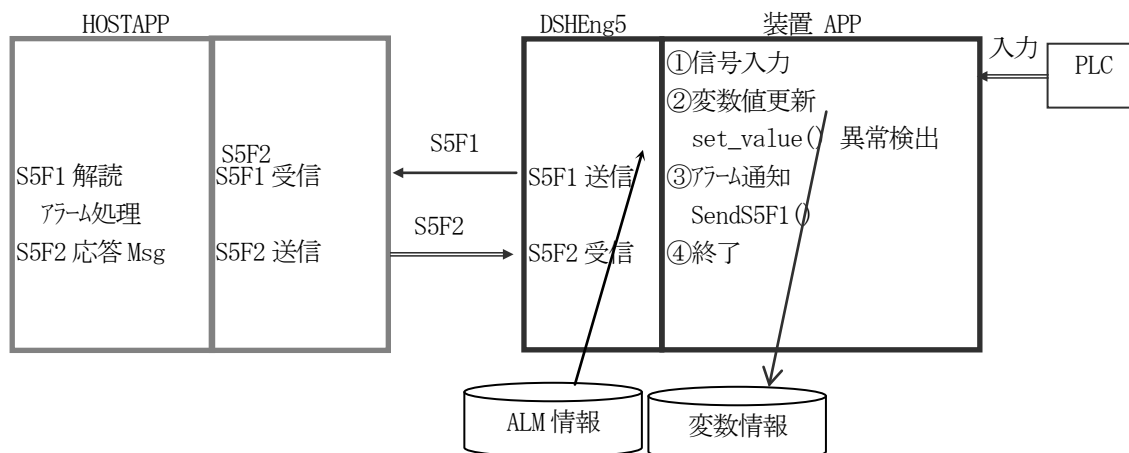


図 5. 4. 2-1 アラーム処理の流れ



## [アラーム関連メッセージ送信用クラス一覧]

装置に対し、直接アラーム情報の設定と要求メッセージ送信のためのクラスがあります。

表 5.4 アラーム関連メッセージ送信用クラス

| メッセージ | 目的           | 使用するクラス / メソッド            | ホスト(H)/装置(E) |
|-------|--------------|---------------------------|--------------|
| S5F3  | アラーム有効・無効の設定 | class_SendS5F3.SendS5F3() | H            |
| S5F5  | アラームリストの要求   | class_SendS5F5.SendS5F5() | H            |
| S5F1  | アラーム報告送信     | class_SendS5F1.SendS5F1() | E            |

## 5.5 スプール機能

ホストとの通信が中断している間、装置が送信しようとしたメッセージの中で、指定されたメッセージ ID を一旦ディスク退避領域に保存し、通信が回復した時に退避したメッセージをまとめてホストに送信するための機能です。

### 5.5.1 スプール状態モデル

#### (1) 状態遷移図

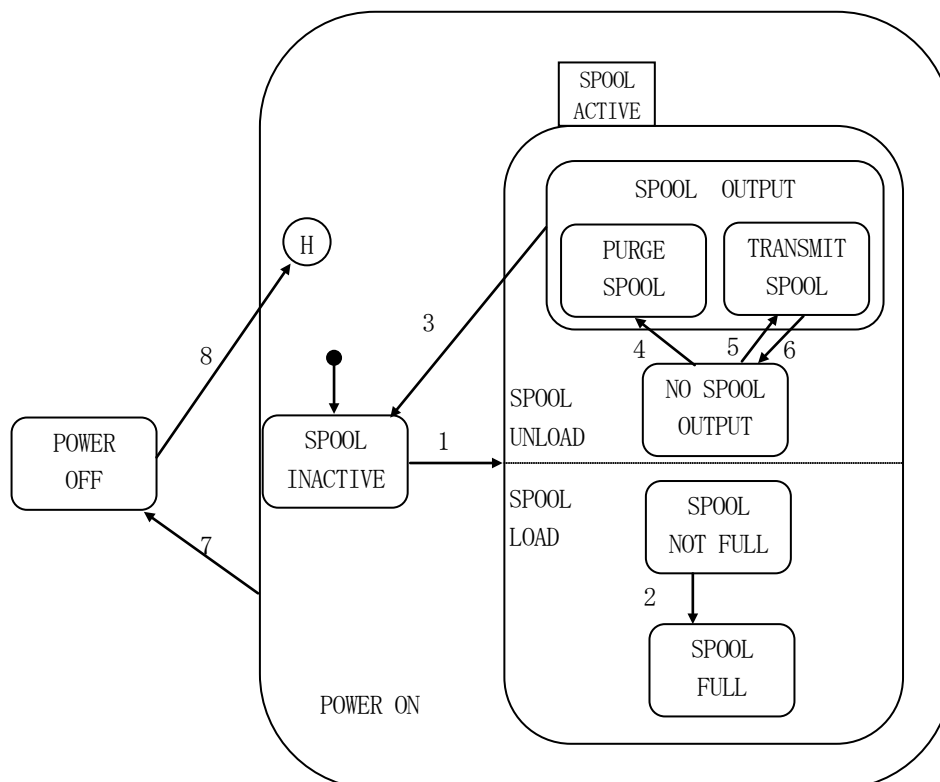


図 5.5.1 スプーリング状態遷移図

#### (2) 状態遷移定義表

表 5.5.1 スプーリング状態遷移定義表

| # | 現在の状態                   | トリガー                                                                                          | 新しい状態                 | 動作                                                                                                                                 | コメント                                                                                                       |
|---|-------------------------|-----------------------------------------------------------------------------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| 1 | SPOOL INACTIVE (スプール休止) | 通信状態は COMMUNICATING から NOT COMMUNICATING へあるいは WAITCRA から WAIT DELAY へ変わり Enable Spool が真である。 | SPOOL ACTIVE (スプール活動) | SpoolCountActual および SpoolCountTotal は初期化される。ホストとのオペレーションは全てアポートされる。SpoolStartTime (SV) は現在の時刻にセットされる。スプーリングがアクティブになっているとホレタに警告する。 | あらゆる OR サブスタートのデフォルト状態に入る。送信できなかったメッセージは送信キューに残され Spool Active 状態で処理される。収集イベント Spooling Activated が発生している。 |

|   |                                  |                                  |                                  |                                                     |                                                                                                     |
|---|----------------------------------|----------------------------------|----------------------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| 2 | SPOOL NOT FULL<br>(スプール空き)       | 作成メッセージがスプールエリアに入らない。            | SPOOL FULL                       | SpoolFullTime(SV) を現在時刻にセットし、スプールがまんばいであることをホータに報告。 | スプーリングエリアに入らないメッセージは遷移の後で取り扱う。シュウウイイベントは発生しない。                                                      |
| 3 | SPOOL OUTPUT<br>(スプール出力)         | スプールエリアが空になった。                   | SPOOL<br>INACTIVE                | スプーリング処理を無効にしスプーリングが終了したことをホータに知らせる。                | 収集イベント SpoolingDeactivated を発生。AND 下位状態スプーロードからも遷移する。                                               |
| 4 | NO SPOOL<br>OUTPUT<br>(スプール出力なし) | w/RSDC=1 の S6F23 を受信した。          | PURGE SPOOL<br>(スプーラー掃)          | 動作なし。                                               | パージング（一掃）処理を開始する。これはホータの要求に基づいているので収集イベントは発生しない。                                                    |
| 5 | NO SPOOL<br>OUTPUT<br>(スプール出力なし) | w/RSDC=0 の S6F23 を受信した。          | TRANSMIT<br>SPOOL<br>(スプール転送)    | 動作なし。                                               | スプールからのメッセージに転送を開始する。これはホータの要求に基づいているので収集イベントは発生しない。                                                |
| 6 | TRANSMIT<br>SPOOL<br>(スプール転送)    | 通信の喪失もしくは MaxSpoolTransmit に達した。 | NO SPOOL<br>OUTPUT<br>(スプール出力なし) | スプール転送処理を一時停止する。                                    | 通信の喪失の場合、イベント Spool Transmit Failure が発生する。MaxSpoolTransmit の値に達した場合には収集イベントは発生しない。                 |
| 7 | POWER ON<br>(電源 ON)              | 装置の電源が遮断                         | POWER OFF<br>(電源 OFF)            | 動作なし。                                               | スプーリングコンテキストはこの遷移の前に不揮発性の記憶装置に保持されている。                                                              |
| 8 | POWER OFF<br>(電源 OFF)            | 装置の電源がもう一度投入された                  | POWER ON<br>(電源 ON)              | スプーリングコンテキストは不揮発性の記憶装置から取り出される。                     | スプーリングが電源 OFF の前に活動状態であった場合、それを続行する。電源 OFF 時にスプール転送が活動状態にあったときは立ち上がると通信状態は中断状態であるから遷移#6 が起こると予想される。 |

## 5. 5. 2 スプーリング処理と流れ

スプーリング処理の流れは5. 5. 1の状態遷移仕様によって下図のようになります。

ホストは装置に対してスプーリングする対象メッセージを S2F43 で通知します。そして、S6F23 で装置にスプーリングされているメッセージをホストに送信するように指示します。

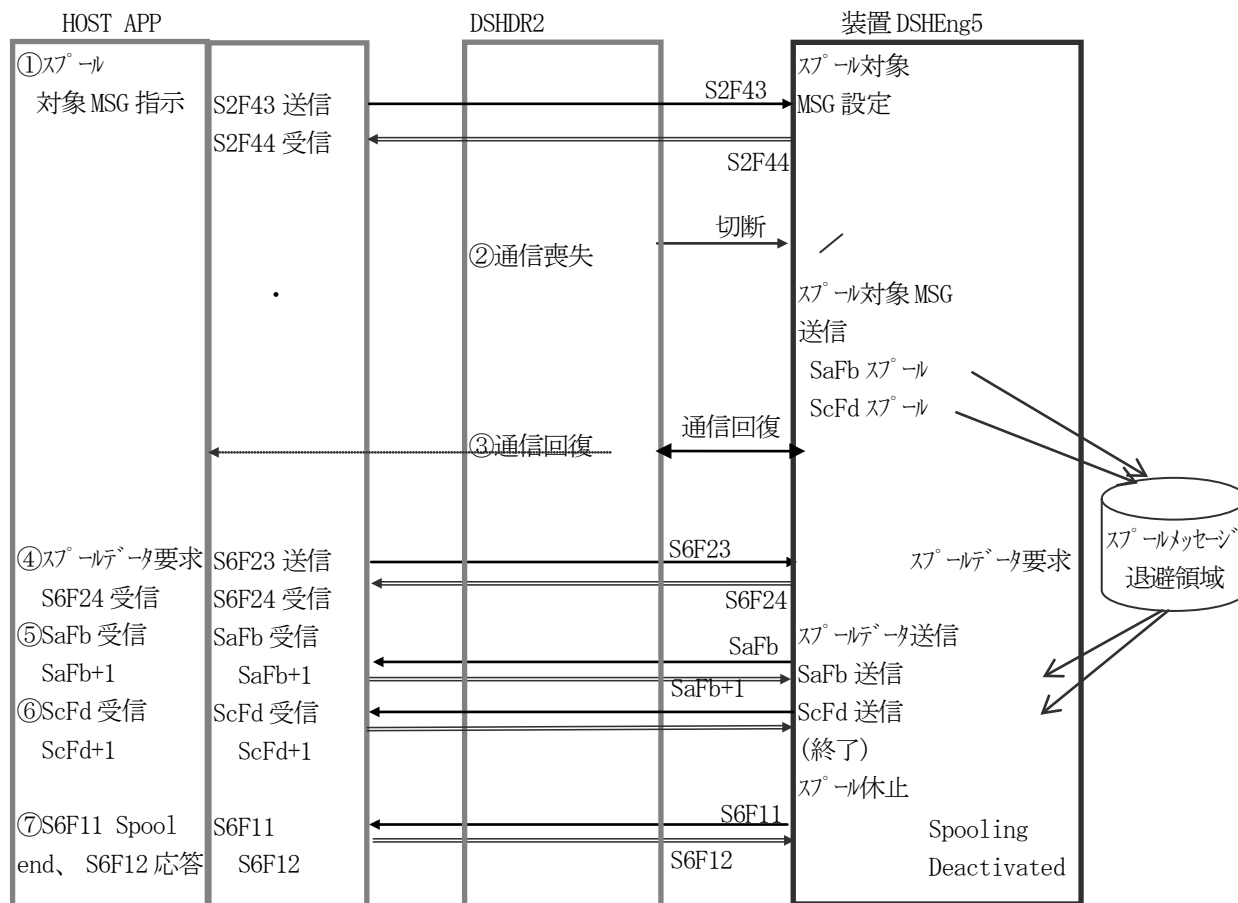


図 5.5.2 スプーリング処理の流れ

- ① ホストから S2F43 メッセージを使ってスプーリング対象メッセージのストリーム、ファンクションの情報を装置に設定します。装置は装置管理情報領域にその情報を記憶します。
- ② その後装置はホストとの通信が喪失された時に、①で指定された送信メッセージのスプーリングが開始します。この例では SaFb, ScFd メッセージが装置側でスプーリングされます。
- ③ ホストとの通信が回復します。
- ④ ホストから S6F23 スプーリングデータ要求(rsdc=1)を行います。装置では、先に退避したメッセージを送信します。
- ⑤ 装置が SaFb を送信
- ⑥ 装置が ScFd を送信
- ⑦ 装置はスプーリングデータの送信が終わったら、収集イベント Spooling Deactivated を S6F11 で送信します。ホストはスプーリングデータの終了を確認して一連の処理は終了します。

## [スプール関連メッセージ送信用クラス一覧]

スプール関連メッセージ送信のためのクラスがあります。

表 5.5 スプール関連メッセージ送信用クラス

| メッセージ | 目的        | 使用するクラス/メソッド                                | ホスト(H)/装置(E) |
|-------|-----------|---------------------------------------------|--------------|
| S2F43 | スプールの設定   | class_SendS2F43.SendS2F43()<br>TSPPOOL_INFO | H            |
| S6F23 | スプールデータ要求 | class_SendS6F23.SendS6F23()                 | H            |
| S6F11 | 収集イベント通知  | class_SendS6F11.SendS6F11()                 | E            |

## 5.6 トレースデータ収集機能

ホストは S2F23 メッセージを使って装置に対し指定した装置状態変数を周期的にトレース監視し、その結果を S6F1 メッセージで報告させることができます。これは状態変数の値をサンプリングするための機能です。

ホストは 1 個以上の状態変数を、指定したレコードサイズ単位で指定合計数だけ指定周期でサンプリングするように指示できます。

処理の流れは概略下図のようになります。

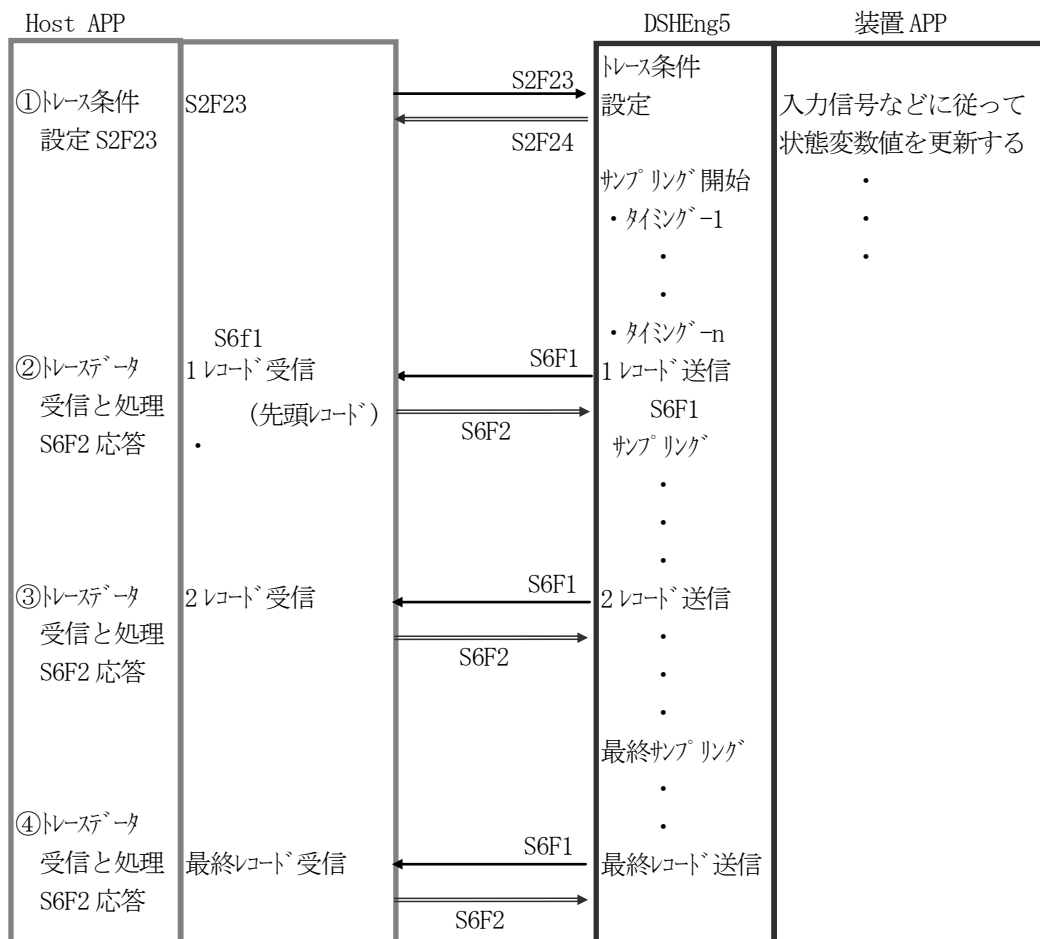


図 5.6 トレース処理の流れ

- ① 最初にホストから S2F23 メッセージを使って、トレースする対象の SVID ならびにトレース周期、合計サンプリング数、送信レコードサイズ情報を送信します。  
装置は、S2F23 に含まれている SVID の評価をし、問題がなければそのメッセージを受け入れます。  
装置は S2F23 内に含まれる SVID に対応する状態変数の入力を指定周期に合わせて必要な準備を行います。そしてホストに S2F24 の応答送信し、同時に、トレースのためのサンプリングを指定周期で行います。
- ②～③ DSHEng5 はサンプリングした結果を S6F1 にエンコードし、ホストに送信します。ホストは、装置から送信されるトレース情報 S6F1 を受信し、処理します。これを totsamp 分のサンプリングが終了するまで続けます。
- ④ 最終のトレースデータの受信です。

S2F23 に含まれるトレース条件は次のとおりです。

**dsper** : 周期(sec)                                **totsmp** : 合計サンプル数  
**repgsz** : グループレコードサイズ            **svid** : 装置変数(1個以上)  
dsper 時間間隔でサンプルし、repgsz 回毎に S6F1 で結果をホストに報告する。  
サンプル回数が totsmp に達したら、最後の情報を送信し、トレースを終了します。

## [トレース関連メッセージ送信用クラス一覧]

トレース関連メッセージ送信のためのクラスがあります。

表 5.6 トレース関連メッセージ送信用クラス

| メッセージ | 目的        | 使用するクラス/メソッド                                         | ホスト(H)/装置(E) |
|-------|-----------|------------------------------------------------------|--------------|
| S2F23 | トレース条件設定  | class_SendS2F23.SendS2F23()<br>TTRACE_INFO           | H            |
| S6F1  | トレースデータ送信 | class_SendS6F1.SendS6F1()<br>TTRACE_MON - trace data | E            |

## 5. 7 プロセスプログラム、レシピ管理機能

DSHEng5 では、プロセスプログラム情報として、以下の3種類のタイプの情報のサポートをします。

- |                        |                |
|------------------------|----------------|
| (1) プロセスプログラム (PP)     | — S7F3 メッセージ   |
| (2) 書式付プロセスプログラム (FPP) | — S7F23 メッセージ  |
| (3) レシピ情報 (RCP)        | — S15F13 メッセージ |

各システムにおいては、上のいずれか1つのタイプを採用することになります。

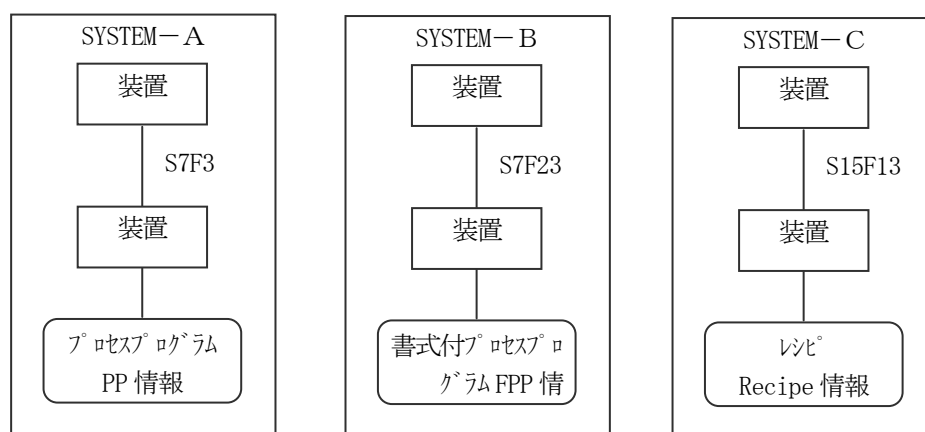


図 5.7 プロセスプログラム (レシピ) のタイプ

DSHEng5 においては、どのタイプに対しても独立の管理を行います。

- 管理情報領域
- DSHEng5 に対するクラス



## 5. 7. 1 プロセスプログラム (PP) 管理機能

プロセスプログラム、PP 情報については 3. 6 の説明を参照してください。

DSHEng5 は以下の処理を行います。

### (1) ホストからメッセージ受信処理

S7F1, S7F3, S7F5, 7F17, S7F19 の受信処理があります。ここでは、S7F3 の受信処理について下のチャートで説明します。

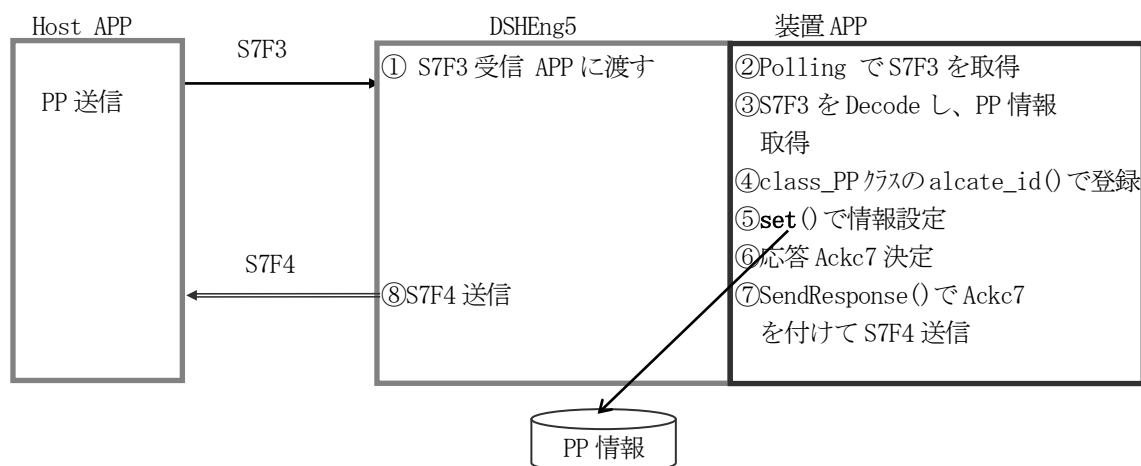


図 5. 7. 1-1 ホストからの S7F3 受信と情報設定

### (2) 装置からのメッセージ送信処理

S7F1, S7F3, S7F5 の送信処理があります。ここでは、S7F3 メッセージの送信処理について説明します。  
(PP 情報が予め、装置管理情報の中に登録されていることが前提です)

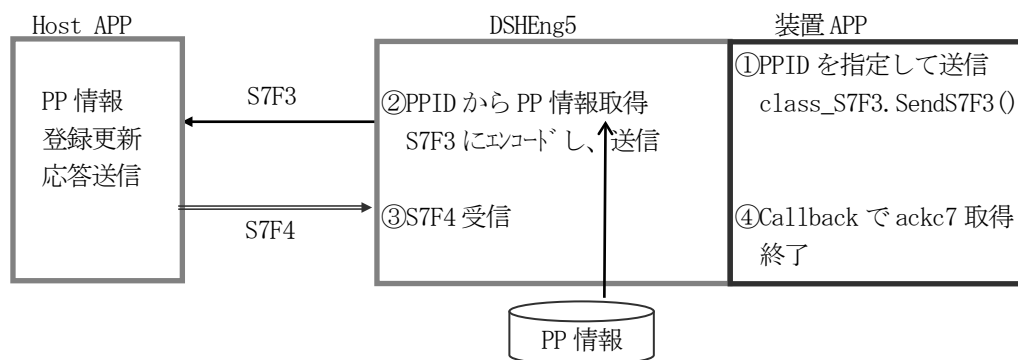


図 5. 7. 1-2 装置から S7F3 を送信

### (3) PP 情報のアクセス

APP は DSHEng5 クラスを使って、PP 情報の設定、取得、削除操作をすることができます。

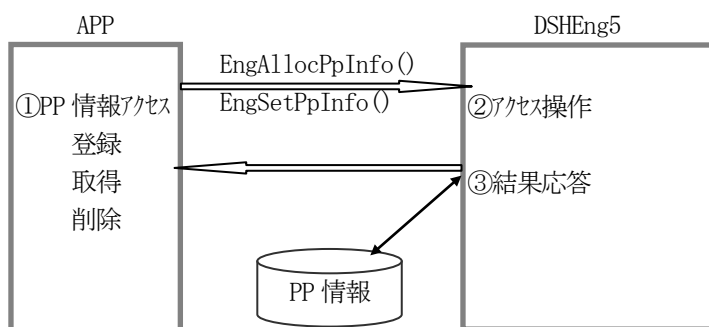


図 5.7. 1-3 APP による PP 情報アクセス

### [プロセスプログラム関連メッセージ送信用クラス一覧]

プロセスプログラム関連メッセージ送信のためのクラスがあります。

表 5.7.1 プロセスプログラム関連メッセージ送信用クラス

| メッセージ | 目的              | 使用するクラス/メソッド                               | ホスト(H)/装置(E) |
|-------|-----------------|--------------------------------------------|--------------|
| S7F1  | プロセスプログラムポート問合せ | class_SendS7F1. SendS7F1()                 | H, E         |
| S7F3  | プロセスプログラム送信     | class_SendS7F3. SendS7F3()<br>TPP_INFO     | H, E         |
| S7F5  | プロセスプログラム要求     | class_SendS7F5. SendS7F5()<br>TPP_INFO     | H, E         |
| S7F17 | プロセスプログラム削除指示   | class_SendS7F17. SendS7F17()               | H            |
| S7F19 | プロセスプログラム一覧要求   | class_SendS7F19. SendS7F19()<br>TPPID_LIST | H            |

なお、S7F17, S7F19 の受信に対しては、DSHEng5 が自動応答します。

## 5. 7. 2 書式付プロセスプログラム (FPP) 管理機能

書式付プロセスプログラム、FPP (Formatted Process Program) 情報については 3. 7 で説明したとおりです。

DSHEng5 は以下の処理を行います。

### (1) ホストからメッセージ受信処理

S7F23, S7F25 の受信処理があります。ここでは、S7F23 の受信処理について下のチャートで説明します。

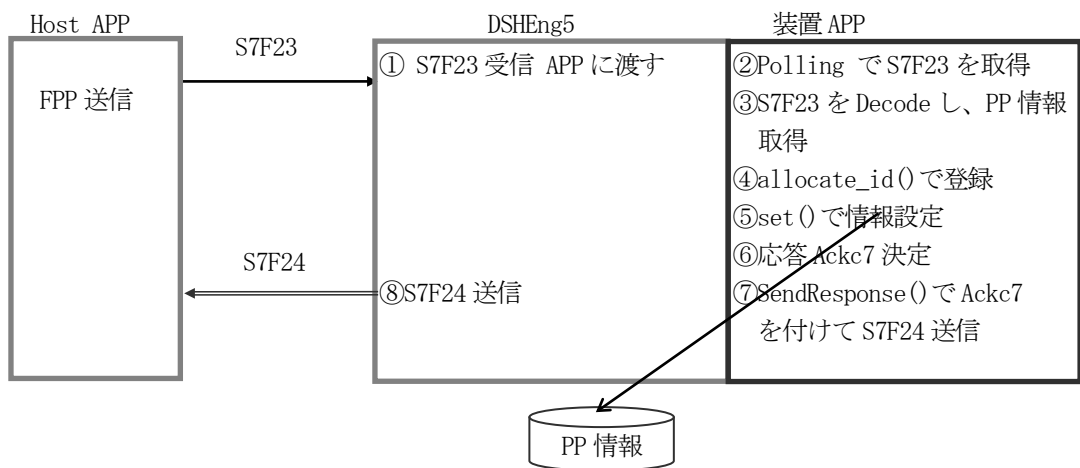


図 5. 7. 2-1 ホストからの S7F23 受信と情報設定

### (2) 装置からのメッセージ送信処理

S7F23, S7F25 の送信処理があります。ここでは、S7F23 メッセージの送信処理について説明します。

(FPP 情報が予め、装置管理情報の中に登録されていることが前提です)

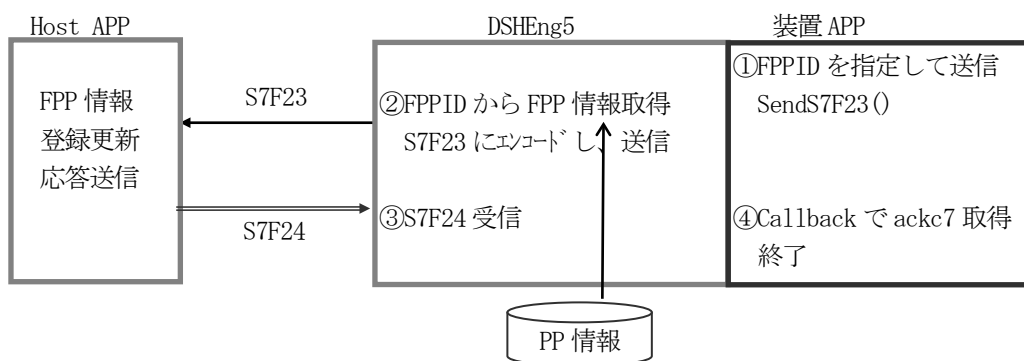


図 5. 7. 2-2 装置から S7F23 を送信

### (3) FPP 情報のアクセス

APP は class\_FPP クラスを使って、FPP 情報の設定、取得、削除操作をすることができます。

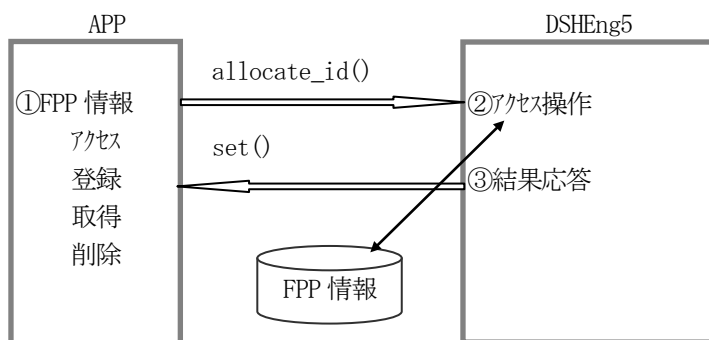


図 5.7.2-3 APP による FPP 情報アクセス

### [書式付プロセスプログラム関連メッセージ送信用クラス一覧]

書式付プロセスプログラム関連メッセージ送信のためのクラスがあります。

表 5.7.2 書式付プロセスプログラム関連メッセージ送信用クラス

| メッセージ | 目的                 | 使用するクラス / メソッド                            | ホスト(H)/装置(E) |
|-------|--------------------|-------------------------------------------|--------------|
| S7F23 | フォーマット付プロセスプログラム送信 | class_SendS7F23. SendS7F23()<br>TFPP_INFO | H, E         |
| S7F25 | フォーマット付プロセスプログラム要求 | class_SendS7F25. SendS7F25()<br>TFPP_INFO | H, E         |

### 5. 7. 3 レシピ (RCPID) 管理機能

レシピ、RCPID 情報については 3. 8 で説明したとおりです。

DSHEng5 は以下の処理を行います。

#### (1) ホストからメッセージ受信処理

S15F3, S15F5, S15F7, S15F9, S15F13, S15F17, S15F19 の受信処理があります。ここでは、S15F13 の受信処理について下のチャートで説明します。

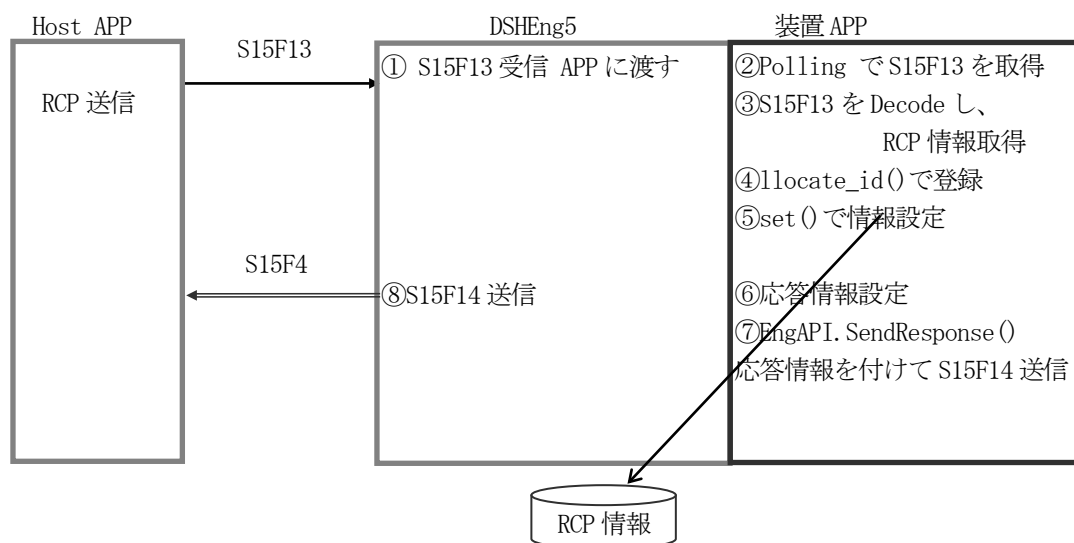


図 5. 7. 3-1 ホストからの S15F13 受信と情報設定

#### (2) 装置からのメッセージ送信処理

S15F3, S15F5, S15F7, S15F9, S15F13, S15F17, S15F19 の送信処理があります。ここでは、S15F13 メッセージの送信処理について説明します。

(RCP 情報が予め、装置管理情報の中に登録されていることが前提です)

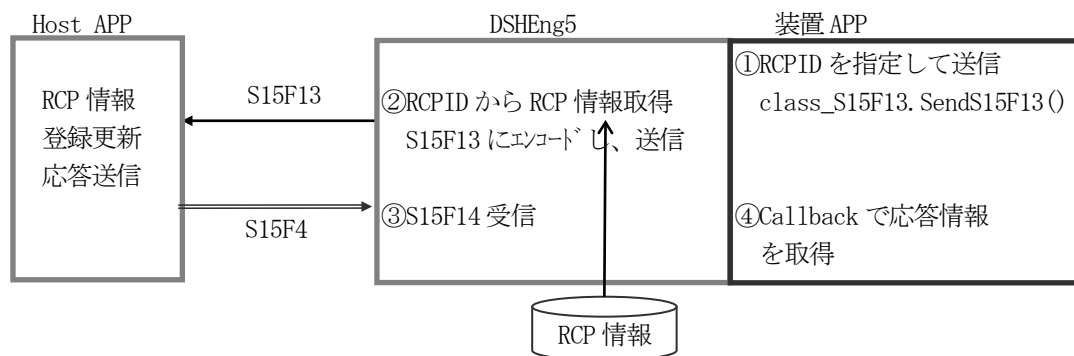


図 5. 7. 3-2 装置から S15F13 を送信

### (3) RCP 情報のアクセス

APP は DSHEng5 クラスを使って、RCP 情報の設定、取得、削除操作をすることができます。

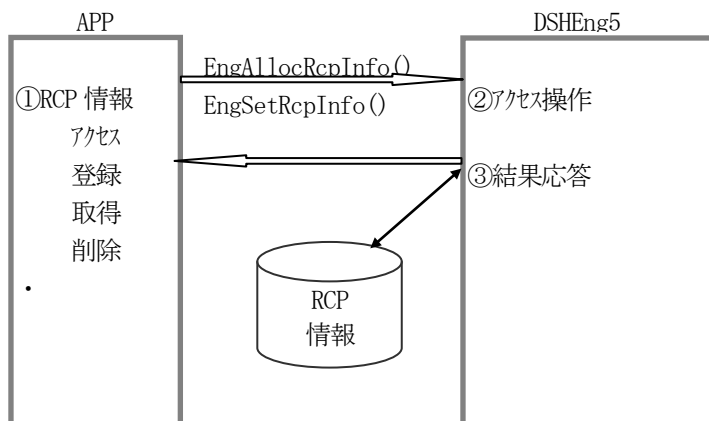


図 5.7.2-3 APP による RCP 情報アクセス

### 【レシピ関連メッセージ送信用クラス一覧】

レシピプログラム関連メッセージ送信のためのクラスがあります。

表 5.7.3 レシピ関連メッセージ送信用クラス

| メッセージ  | 目的                          | 使用するクラス         | ホスト(H)/装置(E) |
|--------|-----------------------------|-----------------|--------------|
| S15F3  | Recipe Namespace action Req | EngSendS15F3()  | H, E         |
| S15F5  | Recipe Namespace rename Req | EngSendS15F5()  | H, E         |
| S15F7  | Recipe Space Request        | EngSendS15F7()  | H, E         |
| S15F9  | Recipe Status Request       | EngSendS15F9()  | H, E         |
| S15F13 | Recipe Create Request       | EngSendS15F13() | H, E         |
| S15F17 | Recipe Retrieve Req         | EngSendS15F17() | H, E         |

## 5. 8 キャリア管理と搬送関連機能

キャリア管理に関連する以下の状態管理機能について記述します。

- (1) ロードポート搬送状態
- (2) キャリア状態
- (3) アクセスモード
- (4) ロードポート予約状態
- (5) ロードポート／キャリア関連状態

### 5. 8. 1 ロードポート搬送状態

ホストは装置のロードポートの管理を行います。

ロードポートの搬送状態を装置状態変数として定義し、以下のロードポートの状態遷移仕様に基づいてユーザが管理し、DSHEng5 エンジンにそれを反映させることになります。

- (1) 状態遷移図

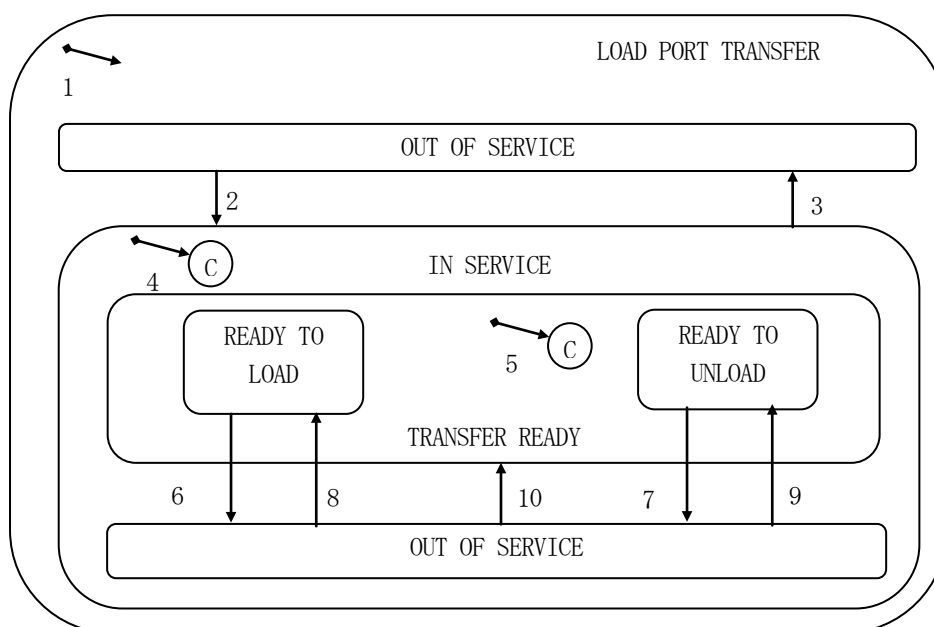


図 5. 8. 1 ロードポート搬送状態遷移図

(2) 状態遷移定義

表 5.8.1 ロードポート搬送状態遷移定義表

| # | 現在の状態          | トリガー                                                                                                                                                        | 新しい状態                                            | 動作 | コメント                                                                                                                                                                                                                              |
|---|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | (状態なし)         | システムのリセット                                                                                                                                                   | OUT OF SERVICE<br>あるいは<br>IN SERVICE<br>(HISTRY) |    | この遷移はシステムリセットの前のそのときの搬送ステータスが何であったかが基になる。<br>このイベント報告の際、入手可能であることが求められるデータは PortID, PortTransferState                                                                                                                             |
| 2 | OUT OF SERVICE | ホストあるいはオペレータがこのロードポートに対してパラメータ値 IN SERVICE で ChangeServiceStatus サービスを依頼した。                                                                                 | IN SERVICE                                       |    | この時点でロードポートは受け渡しに使用可能となる。<br>このイベント報告の際、入手可能であることが求められるデータは PortID, PortTransferState                                                                                                                                             |
| 3 | IN SERVICE     | ホストあるいはオペレータがこのロードポートに対してパラメータ値 OUT OF SERVICE で ChangeServiceStatus サービスを依頼した。                                                                             | OUT OF SERVICE                                   |    | ロードポートは搬送に使用できなくなる。状態遷移後にこのロードポートをキャリアの授受に使用しようとするするとアラームになる。<br>このイベント報告の際、入手可能であることが求められるデータは PortID, PortTransferState                                                                                                         |
| 4 | IN SERVICE     | ホストあるいはオペレータがこのロードポートにする ChangeServiceStatus サービス値を IN SERVICE で依頼した。<br>システムセット:装置の再初期化によってこの遷移を発生させることができる。                                               | TRANSFER READY<br>あるいは<br>TRANSFER BLOCKED       |    | IN SERVICE へ入った際のデフォルト状態。キャリアあるいはロードポートが伽矢リアの授受に使用できないなら、状態は TRANSFER BLOCKED になる。そうでなければ TRANSFER READY である。<br>このイベント報告の際、入手可能であることが求められるデータは PortID, PortTransferState                                                         |
| 5 | TRANSFER READY | サービス:ホストあるいはオペレータがこのロードポートにする。<br>ChangeServiceStatus サービスを IN SERVICE で依頼した。<br>システムセット:装置の再初期化によってこの遷移を発生させることができる。<br>授受の失敗:受け渡し失敗したら遷移#10によってこの遷移が発生する。 | READY To LOAD<br>あるいは<br>READY TO UNLOAD         |    | TRANSFER READY 状態に入った際、キャリアがあるなら下位状態は READY TO UNLOAD であり、そうでないならば下位状態は READY TO LOAD である。<br>このイベント報告の際、入手可能であることが求められるデータは PortID<br>状態が READY To UNLOAD の場合、このイベントの際、入手可能であることが求められるデータは PortID, CarrierID, PortTransferState, |



|   |                  |                                                                                                                                                                                                                                                               |                     |  |                                                                                                                                                                                      |
|---|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6 | READY TO LOAD    | <p>手動 :装置が手動ロード搬送開始の論理的な指示を認識する。このトリガーをユーザが構成することができ、その例を表8に示す。</p> <p>自動 :PIOロード搬送が始まり、PIO準備完了信号が出る。</p> <p>内部バッファ :このポートに対するCarrierOut サービスが開始した。</p>                                                                                                       | TRANSFER<br>BLOCKED |  | <p>CarrierOut サービスがキューイングされ、装置ポートポートが TRANSFER BLOCKED 状態にあるときは、装置はポートポートを TRANSFER BLOCKED 状態のままにしておく必要がある。</p> <p>このイベント報告の際、入手可能であることが求められるデータは<br/>PortID, PortTransferState</p> |
| 7 | READY TO UNLOAD  | <p>手動 :装置が手動アンロード搬送開始の論理的な指示を認識する。このトリガーをユーザが構成することができ、その例を表8に示す。</p> <p>自動 :PIOアンロード搬送が始まり、PIO準備完了信号が出る。</p> <p>内部バッファ :このポートに対するCarrierIn サービスが開始した。</p> <p>CarrierReCreate サービスによる :CarrierReCreate サービスモジュールは、ホストまたは操作員によって発行された。</p>                     | TRANSFER<br>BLOCKED |  | <p>CarrierIn サービスがキューイングされ、装置ポートポートが TRANSFER BLOCKED 状態にあるときは、装置はポートポートを TRANSFER BLOCKED 状態のままにしておく。</p> <p>このイベント報告の際、入手可能であることが求められるデータは<br/>PortID, PortTransferState</p>       |
| 8 | TRANSFER BLOCKED | <p>手動 :キャリアアンロードの移載が完了し、ロードポートが空き、ロード用の移載に準備が整っている。これら2つの条件が満たされ、感知信号がキャリアは存在しないことを示し、且つ搬送が完了したことをオペレータが論理的に意思表示したことを示す。</p> <p>自動 :PIOによるアンロードの移載がPIO完了信号とともに終了する。</p> <p>ナインバッファ :キャリアはロードポートから内部バッファへの移動を終了し、このロードポートに対するCarrierOut サービスがキューイングされていない。</p> | READY TO LOAD       |  | <p>このときで外部から、あるいは装置内部の材料異動機構によってキャリアをロードポートへ移動できる。</p> <p>このイベント報告の際、入手可能であることが求められるデータは<br/>PortID, PortTransferState</p>                                                           |

|    |                     |                                                                                                                                                                                                                                                                                  |                 |  |                                                                                                                                 |
|----|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--|---------------------------------------------------------------------------------------------------------------------------------|
| 9  | TRANSFER<br>BLOCKED | <p>手動 :キャリアに収納される基板の処理が完了した、あるいは CANCEL Carrier/ CancelCarrierAtPort サービスがポートのポート /アンポート位置へ戻った。</p> <p>自動 :キャリアに属している基板の処理が完了した、あるいは CANCEL Carrier/ CancelCarrierAtPort サービスがポートのポート /アンポート位置へ戻った。これは PIO のアンポートを要求する信号で表明される。</p> <p>内部バッファ動作 : キャリアは内部バッファからポートへの移動を完了した。</p> | READY TO UNLOAD |  | <p>このとき、ロードポート上のキャリアをポートポートから外部へアンポートできる。</p> <p>このイベントの際、入手可能であることが求められるデータは<br/>PortID, CarrierID,<br/>PortTransferState,</p> |
| 10 | TRANSFER<br>BLOCKED | <p>移載は失敗し、キャリアはポートもアンポートもされなかった。</p>                                                                                                                                                                                                                                             | TRANSFER READY  |  | <p>遷移#5 によって決定される。TRANSFER READY の下位状態。</p> <p>このイベント報告の際、入手可能であることが求められるデータは<br/>PortID, PortTransferState</p>                 |

表 5.8.1-1 キャリア搬送境界面

| 授受の種類  | 授受の方法  | 開始の境界                                                                                                                     | 終了の境界                                                                                                                                           |
|--------|--------|---------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| LOAD   | MANUAL | この開始境界はユーザが定義する。開始境界の既知の例は次のものを含むがそれに限定しない。キャリア有無センサのキャリア検知、ロッドポートドアの解放、ロッドポートまたは装置端末のスイッチを介して装置へのホペレタの入力。                | この終了境界はユーザが定義する。終了境界の既知の例は次のものを含むがそれに限定しない。キャリア有無センサおよびキャリア載置センサがキャリアを検知してからのプリセットされた設定可能な時間、ロッドポートドアの閉鎖、ロッドポートまたは装置端末のスイッチを介して装置へのホペレタの入力。     |
|        | AUTO   | ロッドの場合 PIO の信号“READY”がアクティブ。 SEMI E84 参照                                                                                  | ロッド終了時には PIO の信号が“COMPT”になる。 SEMI E84 参照                                                                                                        |
| UNLOAD | MANUAL | この開始境界はユーザが定義する。開始境界の既知の例は次のものを含むがそれに限定しない。キャリア有無及びキャリア載置センサがキャリアを検知しなくなった。ロッドポートドアの解放、ロッドポートまたは装置端末のスイッチを介して装置へのホペレタの入力。 | この終了境界はユーザが定義する。終了境界の既知の例は次のものを含むがそれに限定しない。キャリア有無センサおよびキャリア載置センサがキャリアを検知しなくなったからのプリセットされた設定可能な時間、ロッドポートドアの閉鎖、ロッドポートまたは装置端末のスイッチを介して装置へのホペレタの入力。 |
|        | AUTO   | アンロッドの場合 PIO の信号“READY”がアクティブ。 SEMI E84 参照                                                                                | アンロッド終了時には PIO の信号が“COMPT”になる。 SEMI E84 参照                                                                                                      |

(3) ロード搬送状態管理と処理の流れ

(3) - 1 ホストからの切替

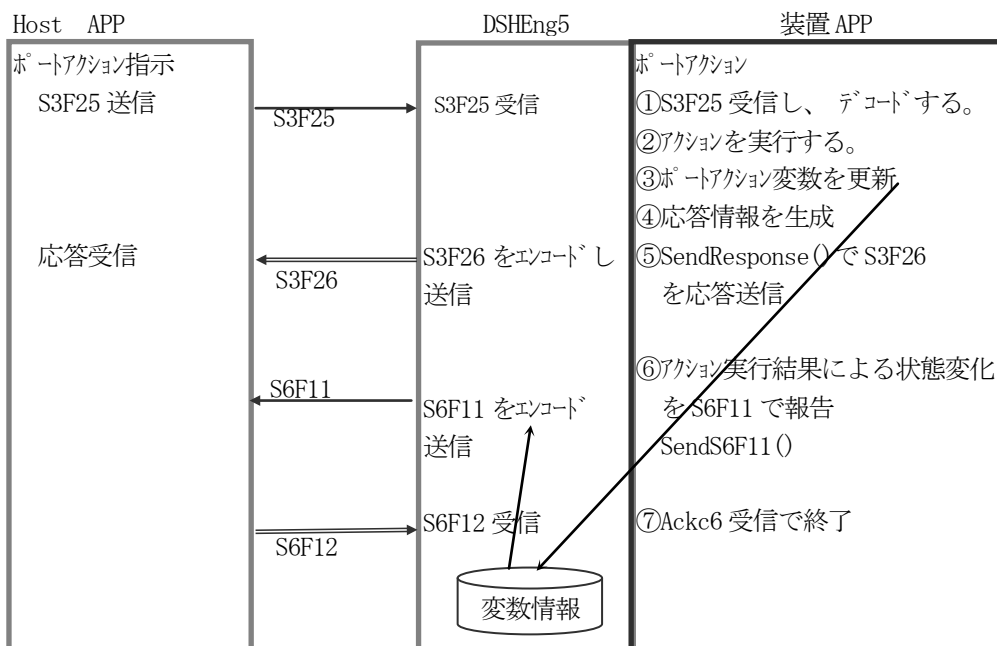


図 5.8. 1-1 ホータ/上位プロセス指示によるポートサービス状態管理処理の流れ

(3) - 2 装置からの報告による切替

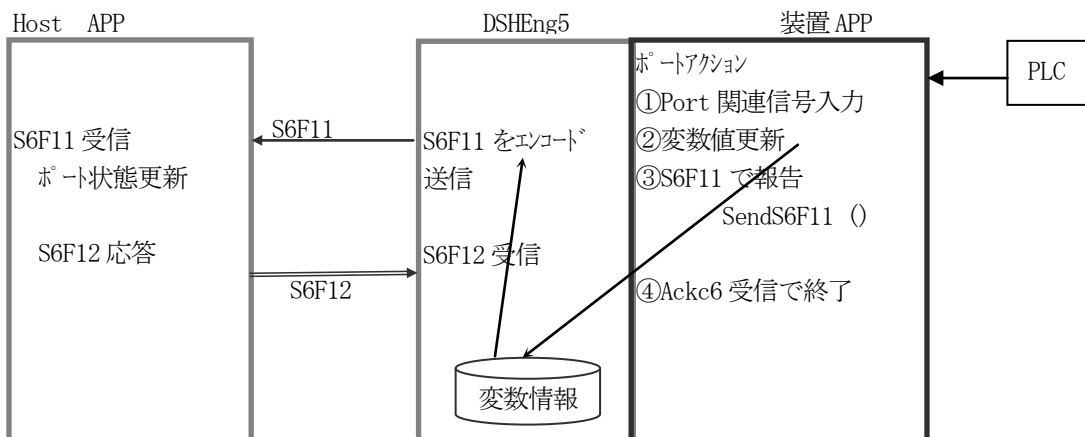


図 5.8. 1-2 装置によるポートサービス状態管理処理の流れ

[ロードポート関連メッセージ送信用クラス一覧]

ロードポート関連メッセージ送信のためのクラスがあります。

表 5.8.1 ロードポート関連メッセージ送信用クラス

| メッセージ | 目的         | 使用するクラス                                     | ホスト(H)/装置(E) |
|-------|------------|---------------------------------------------|--------------|
| S3F25 | ポートアクション要求 | class_SendS3F25, SendS3F25 ()<br>TPORT_INFO | H            |

## 5. 8. 2 キャリア状態モデル

キャリアの状態管理は以下述べるキャリア状態遷移仕様に基づいて行われます。

オペレータ操作または上位プロセスからの指令によってキャリアオブジェクト生成、装置への情報通知、状態管理、削除までの処理を行います。

多くの処理は DSHeng5 が提供するサービス機能を使用し実行することになります。

### (1) 状態遷移図

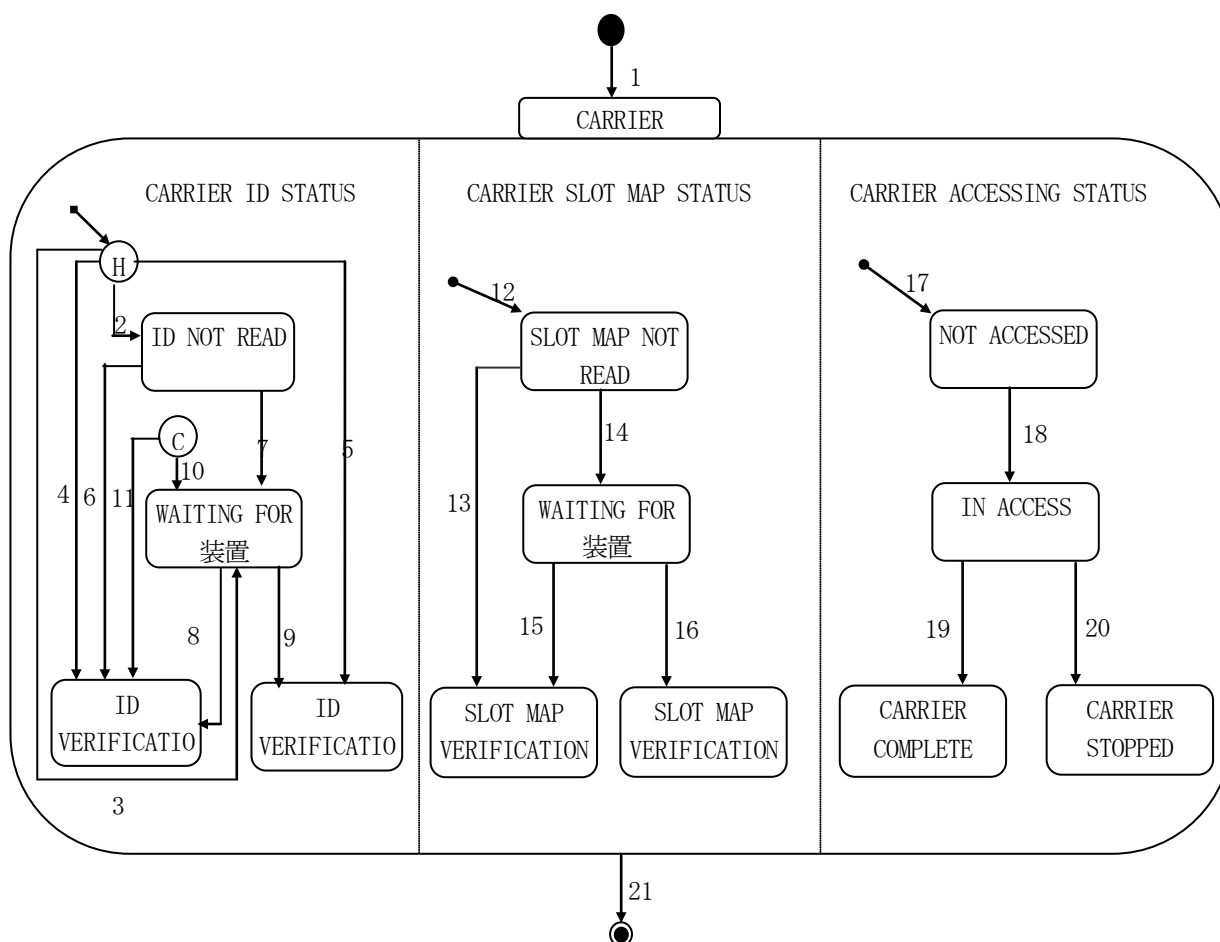


図 5.8.2 キャリア状態遷移図

(2) 状態遷移定義

表 5.8.2 キャリア状態遷移定義表

| # | 前の状態        | トリガー                                                                               | 新しい状態                | アクション                                                       | コメント                                                                                                                                                                            |
|---|-------------|------------------------------------------------------------------------------------|----------------------|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | (状態なし)      | キャリアがインスタンス化される。                                                                   | CARRIER              | なし                                                          | この遷移にはイベントレポートは不要                                                                                                                                                               |
| 2 | (状態なし)      | 正常：Bind サービスあるいは Carrier Notification サービスを受信する。                                   | ID NOT READ          | なし                                                          | このイベント報告に必要なデータ：<br>CarrierID,<br>CarrierIDStatus                                                                                                                               |
| 3 | (状態なし)      | seijou：現在装置に存在していない。CarrierID の読取りに成功する。<br>異常：CarrierID の読取りに成功するが、送チンによる照合に失敗する。 | WAITING FOR 装置       | なし                                                          | このイベント報告に必要なデータ：<br>PortID<br>CarrierID<br>CarrierIDStatus<br>正常時に、この遷移は bind サービスが発行されていない場合（ホストによる照合）には ID 読取り成功の後発生する、あるいは異常時としては Bind サービスが実行された後、ID 読取りが成功し装置による照合に失敗する場合。 |
| 4 | (状態なし)      | ID 読取り失敗もしくは UnknownCarrierIDEvents: Proceed WithCarrier サービスを受信する。                | ID VERIFICATION OK   | ProceedWithCarrier サービスで与えられる CarrierID を持つキャリアがインスタンス化される。 | このイベント報告に必要なデータ：<br>CarrierID<br>CarrierIDStatus<br>この遷移は bind サービスが受信されなかった場合にのみ起こる。                                                                                           |
| 5 | (状態なし)      | ID 読取り失敗もしくは UnknownCarrierIDEvents: CancelCarrier サービスを受信する。                      | ID VERIFICATION FAIL | CancelCarrier サービスで与えられる CarrierID を持つキャリアがインスタンス化される。      | このイベント報告に必要なデータ：<br>CarrierID<br>CarrierIDStatus<br>この遷移は bind サービスが受信されなかった場合にのみ起こる。                                                                                           |
| 6 | ID NOT READ | キャリア ID の読込みに成功し、装置は CarrierID の照合に成功する。                                           | ID VERIFICATION OK   | なし                                                          | このイベント報告に必要なデータ：<br>PortID<br>CarrierID,<br>CarrierIDStatus                                                                                                                     |
| 7 | ID NOT READ | キャリアの読込みに失敗する。                                                                     | WAITING FOR 装置       | なし                                                          | このイベント報告に必要なデータ：<br>PortID<br>CarrierID<br>CarrierIDStatus                                                                                                                      |

|    |                   |                                                                                                                                                                         |                          |                             |                                                                                                       |
|----|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|-----------------------------|-------------------------------------------------------------------------------------------------------|
| 8  | WAITING FOR 装置    | ProceedWithCarrier サービスを受信する。                                                                                                                                           | ID VERIFICATION OK       | なし                          | このイベント報告に必要なデータ:<br>PortID<br>CarrierID<br>CarrierIDStatus                                            |
| 9  | WAITING FOR 装置    | CancelCarrier サービスを受信する。                                                                                                                                                | ID VERIFICATION FAIL     | なし                          | このイベント報告に必要なデータ:<br>PortID<br>CarrierID<br>CarrierIDStatus                                            |
| 10 | ID NOT READ       | BypassReadID 変数が偽にセットされており、IDリードが使用不可あるいは実装されていない時にキャリアが置かれる。                                                                                                           | WAITING FOR 装置           | ProceedWithCarrier を待つ      | このイベント報告に必要なデータ:<br>PortID<br>CarrierID<br>CarrierIDStatus                                            |
| 11 | ID NOT READ       | BypassReadID 変数が真にセットされており、IDリードが使用不可あるいは実装されていない時にキャリアが置かれる。                                                                                                           | ID VERIFICATION OK       | なし                          | このイベント報告に必要なデータ:<br>PortID<br>CarrierID<br>CarrierIDStatus                                            |
| 12 | (状態なし)            | キャリアがインスタンス化される。                                                                                                                                                        | SLOT MAP NOT READ        | なし                          | この遷移にはイベントが必要でない。                                                                                     |
| 13 | SLOT MAP NOT READ | 相対がスロットマップの読み取りおよび照合に成功する。                                                                                                                                              | SLOT MAP VERIFICATION OK | なし                          | このイベント報告に必要なデータ:<br>PortID (有効な場合)<br>CarrierID,<br>LocatioID<br>CarrierAccessStatus<br>SlotMapStatus |
| 14 | SLOT MAP NOT READ | 正常なホストでの照合:スロットマップの読み取りに成功し、装置はホストの称号を待っている。<br>装置での照合の失敗:スロットマップの読み取りに成功したが、装置での照合には失敗した。<br>スロットマップでの読み取り失敗:スロットマップが読み取れない。<br>キャリア基板の位置異常:スロットマップの読み取りで基板位置の異常が判明した。 | WAITING FOR 装置           | SlotMap 属性に新しいスロットマップを保存する。 | このイベント報告に必要なデータ:<br>PortID (有効な場合)<br>CarrierID<br>LocatioID<br>SlotMap(有効な場合)<br>理由<br>SlotMapStatus |
| 15 | WAITING FOR 装置    | ProceedWithCarrier サービスを受け付ける。                                                                                                                                          | SLOTMAP VERIFICATION OK  | 指示通りキャリアを進める。               | このイベント報告に必要なデータ:<br>PortID (有効な場合)<br>CarrierID<br>LocatioID<br>SlotMapStatus                         |

|    |                |                                                                                                                                                                        |                           |                           |                                                                                                          |
|----|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------------------|----------------------------------------------------------------------------------------------------------|
| 16 | WAITING FOR 装置 | CancelCarrier サービスを受信する。                                                                                                                                               | SLOTMAP VERIFICATION FAIL | キャリアをアンロードするための準備をする。     | このイベント報告に必要なデータ:<br>PortID (有効な場合)<br>CarrierID<br>LocationID<br>CarrierAccessingStatus<br>SlotMapStatus |
| 17 | (状態なし)         | キャリアオブジェクトがインスタント化される。                                                                                                                                                 | NOT ACCESSED              | なし                        | No Event is required for this transition.                                                                |
| 18 | NOT ACCESSED   | 装置がキャリアへのアクセスを開始する。                                                                                                                                                    | IN ACCESS                 | なし                        | このイベント報告に必要なデータ:<br>CarrierID<br>CarrierAccessingStatus                                                  |
| 19 | IN ACCESS      | 装置がキャリアへのアクセスを正常に終了する。                                                                                                                                                 | CARRIER COMPLITE          | なし                        | このイベント報告に必要なデータ:<br>CarrierID<br>CarrierAccessingStatus                                                  |
| 20 | IN ACCESS      | 装置がキャリアへのアクセスを異常終了する。                                                                                                                                                  | CARRIER STOPPED           | なし                        | このイベント報告に必要なデータ:<br>CarrierID<br>CarrierAccessingStatus                                                  |
| 21 | CARRIER        | 正常 :キャリアが装置からアンロードされる。<br>サービスによる異常 :キャリアのロードの前に CancelBind サービスあるいは CancelCarrierNotification サービスを受け付ける。<br>そおうちによる異常 :沿おう地での照合が失敗し、装置は自分で開始する CancelBind サービスを実行する。 | (状態なし)                    | 装置はキャリアオブジェクトのインスタンスを消滅さ。 | このイベント報告に必要なデータ:<br>CarrierID                                                                            |



(3) キャリア状態管理と処理の流れ

マニュアルモード、自動モードについてキャリア管理処理の概略の流れについて記述します。

ここに記述されている処理は単なる一例であり、実際には各システムの独自の仕様に基づいて処理されることとなります。

キャリアの状態は、状態変数(SV)、キャリア情報の中の状態データに反映され、それに基づいて装置からホストに対し、S6F11 で状態情報を報告することとなります。

(3) - 1 マニュアルモード

マニュアルモードではホストとの間が通信状態であれば、キャリアの状態を逐一装置から S6F11 で通知することとなります。装置は、ロードされたキャリアが ID リーダで読取られた段階で、キャリア ID が発生し、それに対応する CARID 変数にセットします。同時にキャリア情報として DSHeng5 に登録されることとなります。

登録以降、キャリア ID の状態は搬送、装置による処理などによって状態が変わり、それがキャリア情報の中の状態情報そして、イベント ID にリンクされている装置状態変数にも反映更新されます。

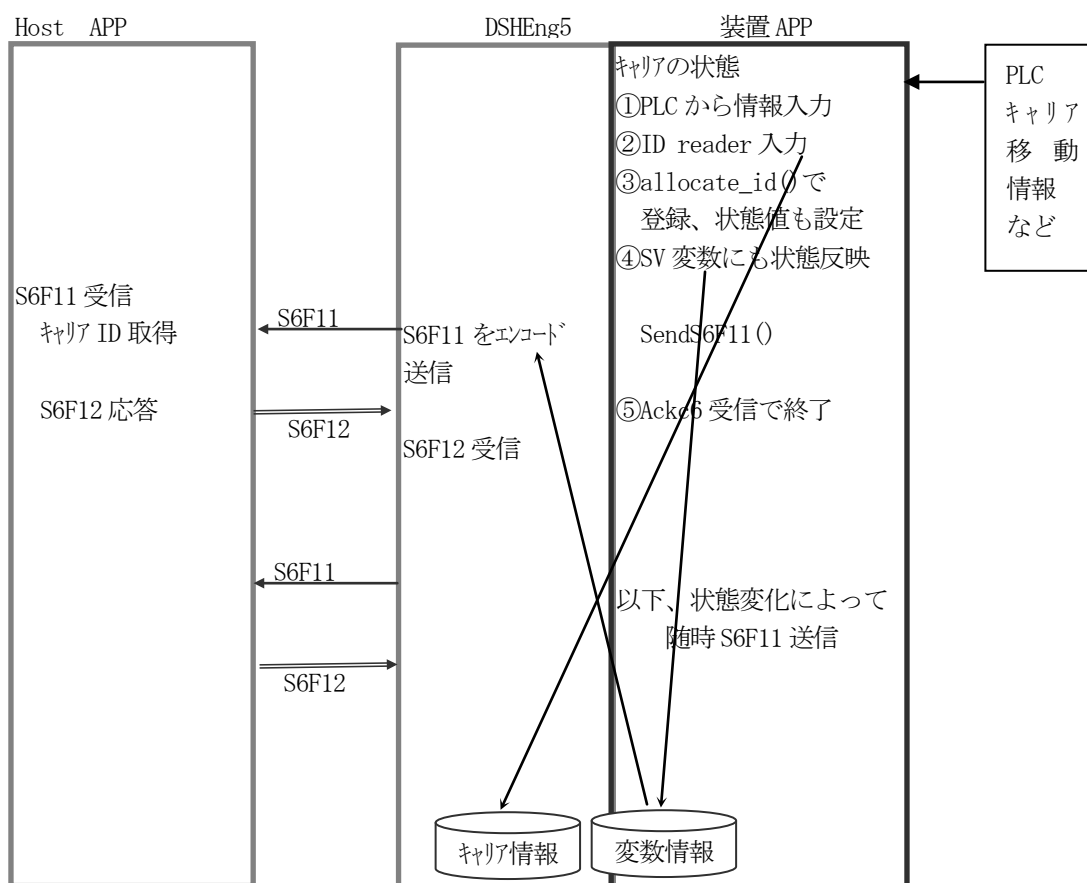


図 5. 8. 2-1 マニュアルモード キャリア状態管理処理の流れ

(3) - 2 オートモード

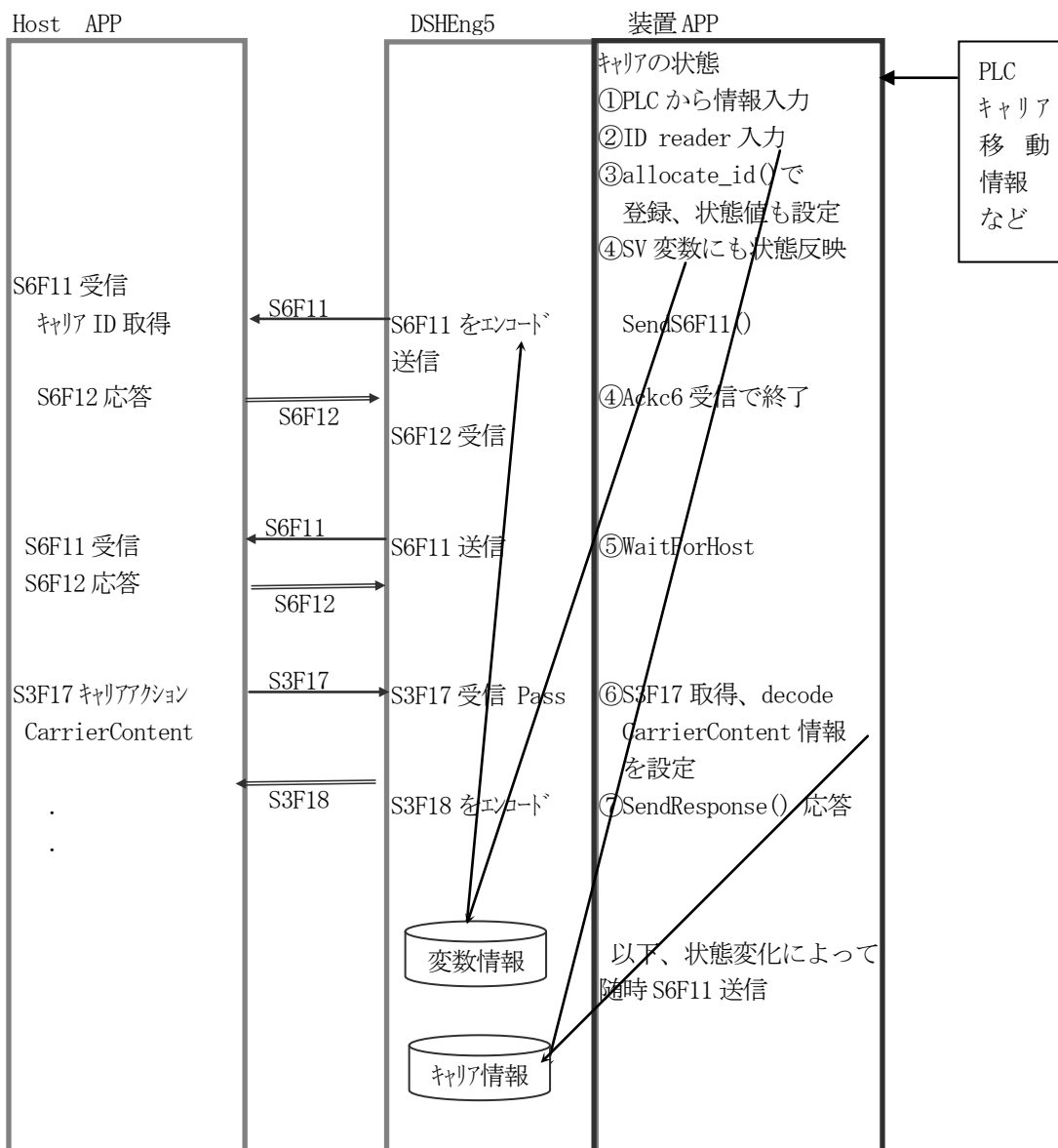


図 5.8.2-2 オートモード キャリア状態管理処理の流れ

**[キャリアアクション関連メッセージ送信用クラス一覧]**

キャリアアクション関連メッセージ送信のためのクラスがあります。

表 5.8.2 キャリアアクション関連メッセージ送信用クラス

| メッセージ | 目的          | 使用するクラス / メソッド                            | ホスト (H) / 装置 (E) |
|-------|-------------|-------------------------------------------|------------------|
| S3F17 | キャリアアクション要求 | class_SendS3F17.SendS3F17()<br>TCACT_INFO | H                |

### 5. 8. 3 アクセスモード管理

アクセスモードの管理は、状態遷移の定義に基づいて行われます。

#### (1) 状態遷移図

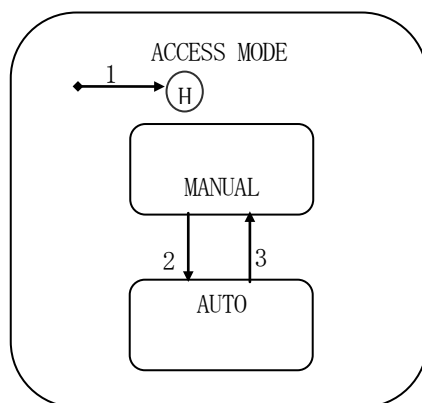


図 5.8.3 アクセスモード状態遷移図

#### (2) 状態遷移定義

表 5.8.3 アクセスモード状態定義表

| # | 前の状態   | トリガー                                                                            | 新しい状態                 | アクション                        | コメント                                                                                                                         |
|---|--------|---------------------------------------------------------------------------------|-----------------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 1 | (状態なし) | システムの再起動                                                                        | MANUAL あるいは AUTO (履歴) | アクセスモードがシステムがリセットされる前の状態に戻る。 | このイベント報告に必要なデータ:<br>PortID<br>AccessMode                                                                                     |
| 2 | MANUAL | ホストまたはオペレータが AUTO と指定して ChangeAccess サービスを実行した。このトリガーはキャリアを除いていつでも発生可能である。      | AUTO                  |                              | この状態遷移後はマニュアル搬送は許されない。オペレータは製造装置のコンソールからもこのトランザクションを引き起こす。<br>このイベント報告に必要なデータ:<br>PortID<br>AccessMode                       |
| 3 | AUTO   | ホストまたはオペレータが MANUAL と指定して ChangeAccess サービスを実行した。このトリガーはキャリア搬送時を除いていつでも発生可能である。 | MANUAL                |                              | オペレータは製造装置のコンソールからあるいはポートポートのマニュアルスイッチでもこのトランザクションを引き起こす。<br>この状態遷移後は自動搬送は許されない。<br>このイベント報告に必要なデータ:<br>PortID<br>AccessMode |

(3) 状態と処理の流れ

ロードポートのアクセスモード状態を保持するための装置状態変数が装置管理情報内に定義されていることが条件です。モード切替は、オペレータコンソールからの切替またはホストからのポートアクション指令によって行われます。ロードポートモード管理に関する処理の流れは概略下図のようになります。

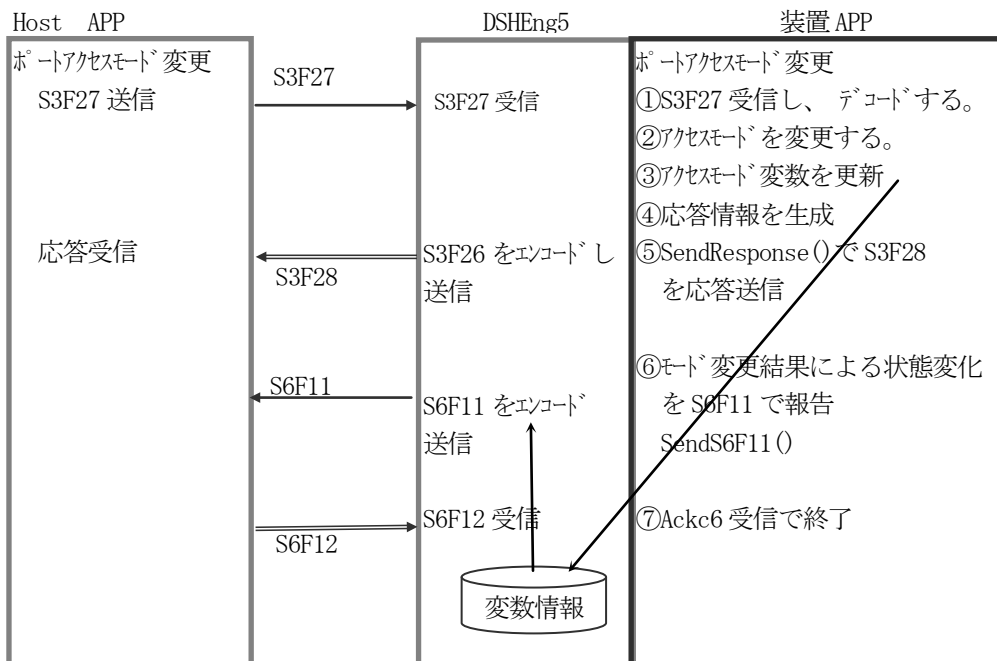


図 5.8.3 アクセスモード管理処理の流れ

[アクセスモード関連メッセージ送信用クラス一覧]

アクセスモード関連メッセージ送信のためのクラスがあります。

表 5.8.3 アクセスモード関連メッセージ送信用クラス

| メッセージ | 目的            | 使用するクラス / メソッド                               | ホスト(H)/装置(E) |
|-------|---------------|----------------------------------------------|--------------|
| S3F27 | Change Access | class_SendS3F27. SendS3F27()<br>TACCESS_INFO | H            |

## 5. 8. 4 ロード予約状態管理

### (1) 状態遷移図

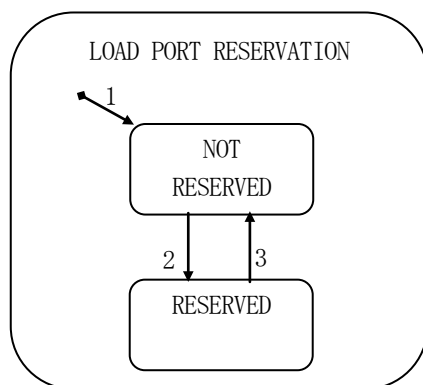


図 5.8.4 ロードポート予約状態遷移図

### (2) 状態遷移定義

表 5.8.4 ロードポート予約状態定義表

| # | 前の状態         | トリガー                                                                                                                       | 新しい状態        | アクション                                            | コメント                                                                                                      |
|---|--------------|----------------------------------------------------------------------------------------------------------------------------|--------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| 1 | (状態なし)       | システムのリセット                                                                                                                  | NOT RESERVED |                                                  | この遷移にはイベント報告は必要ない。                                                                                        |
| 2 | NOT RESERVED | サービス：サービスで予約する際は、ホストあるいはオペレータが製造装置に ReserveAtPort サービスあるいは Bind サービスを依頼する。<br>キャリアの排出：装置が物理的にキャリアアウト操作の開始によるトリガーの発生        | RESERVED     | ユーザーが予約視認表示を使うように装置を構成する場合は、このロードポートで視認表示を作動させる。 | このイベント報告に必要なデータ： PortID<br>LoadPortReservationState<br>キャリアアウトあるいは bind サービスがこの遷移のトリガーなら、CarrierID が含まれる。 |
| 3 | RESERVED     | サービス：サービスで予約をキャンセルする際は、ホストあるいはオペレータが CANCELBind サービスあるいは CancelReservationAtPort サービスを依頼する。<br>キャリアの到着：キャリアが予約されたポートに到着する。 | NOT RESERVED | ユーザーが予約視認表示を使うように装置を構成する場合は、このロードポートで視認表示を解除さ。   | このイベント報告に必要なデータ： PortID<br>LoadPortReservationState                                                       |

(3) 状態と処理の流れ

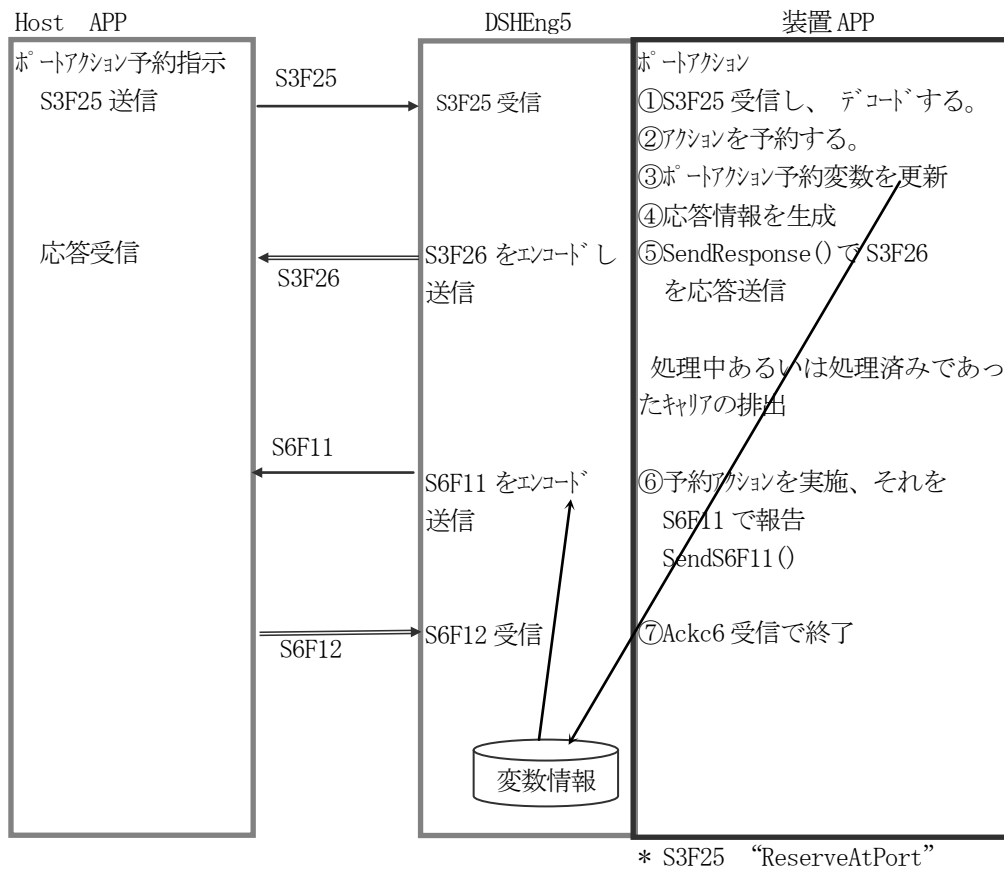


図 5.8.4 ポート予約状態管理処理の流れ

## 5. 8. 5 ロードポート/キャリア関連状態管理

### (1) 状態遷移図

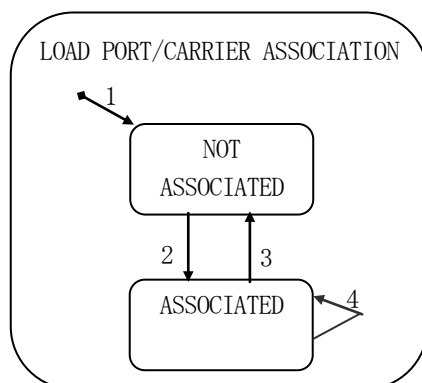


図 5.8.5 ロードポート/キャリア関連状態遷移図

### (2) 状態遷移定義

表 5.8.5 ロードポート/キャリア関連状態遷移定義表

| # | 前の状態           | トリガー                                                                                                                                                                                                                                                                                                                                 | 新しい状態          | アクション                 | コメント                                                                                                                                                                                                                        |
|---|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | (状態なし)         | システムのリセット                                                                                                                                                                                                                                                                                                                            | NOT ASSOCIATED |                       | この遷移にはイベント報告は不要。                                                                                                                                                                                                            |
| 2 | NOT ASSOCIATED | <p>正常サービス：正常状況においてサービスが関連づける際は、ポートがふさがっていないければ、ホストは製造装置にBindサービスを送信する。</p> <p>異常サービス：異常状況において、サービスが関連づける際は、ポートがふさがっていないければ、ホストは製造装置にProceedWithCarrierサービスを送信する。</p> <p>CarrierID 読取り：CarrierID の読取りによって関連づけが起きる場合、製造装置はCarrierID の読取りが実行され時点で関連を作る。</p> <p>既知キャリア：製造装置が既に知っているキャリアがロードポートにロードされる。これはCarrierOutサービスが開始されたときに起こる。</p> | ASSOCIATED     | このロードポートが相関視認表示が作動する。 | <p>製造装置が CarrierID を読み取る前に Bind サービスが実行された場合、製造装置はCarrierID の照合を実行できる。ロードポートへのCarrierID お関連づけが一旦成立したらそのロードポートがまたNOT ASSOCIATEDへ戻るまで関連づけはできない。</p> <p>このイベント報告に必要なデータ：<br/>PortID<br/>CarrierID<br/>PortAssociationState</p> |

|   |            |                                                                                                                                                                                  |                |                                                                                                                                                                            |                                                                                                                        |
|---|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| 3 | ASSOCIATED | <p>サービス：ポート関連のキャンセルを要する際は、キャリアがそのポートに到着する前、または搬送シーケンスが開始する前に、ホストが製造装置へ CancelBind サービスを創始すると完了する。</p> <p>キャリアアンロード：ポートからキャリアを取り除く、あるいは装置がキャリアを内部バッファへ移動することによる関連づけのキャンセルが可能。</p> | NOT ASSOCIATED | このポートが相関視認表示が解除する。                                                                                                                                                         | <p>キャリアのアンロードは処理の前起こる場合も後に起こる場合もある。ポートへは別の関連づけができる。</p> <p>このイベント報告に必要なデータ：</p> <p>PortID<br/>PortAssociationState</p> |
| 4 | ASSOCIATED | <p>製造装置によるキャリア照合が失敗し、キャリアはポート上のキャリアの ID 値を仮定する。</p> <p>内部バッファ：キャリアがアンロードされ、キューイングされていた CarrierOut サービスが始まる。</p>                                                                  | ASSOCIATED     | <p>Bind サービスによって関連づけられていた既存の CarrierID は製造装置によって関連づけを解かれ、ポートには新しい CarrierID が関連づけられている。製造装置は CancelCarrier コマンドあるいは ProceedWithCarrier コマンドのどちらかをホストから受信するまでアクションを控える。</p> | <p>この遷移は Bind コマンドがしようされたときだけ発生する。</p> <p>このイベント報告に必要なデータ：</p> <p>PortID<br/>CarrierID<br/>PortAssociationState</p>    |



## 5.9 プロセスジョブ管理機能

### 5.9.1 プロセスジョブ状態モデル

(1) 状態遷移図

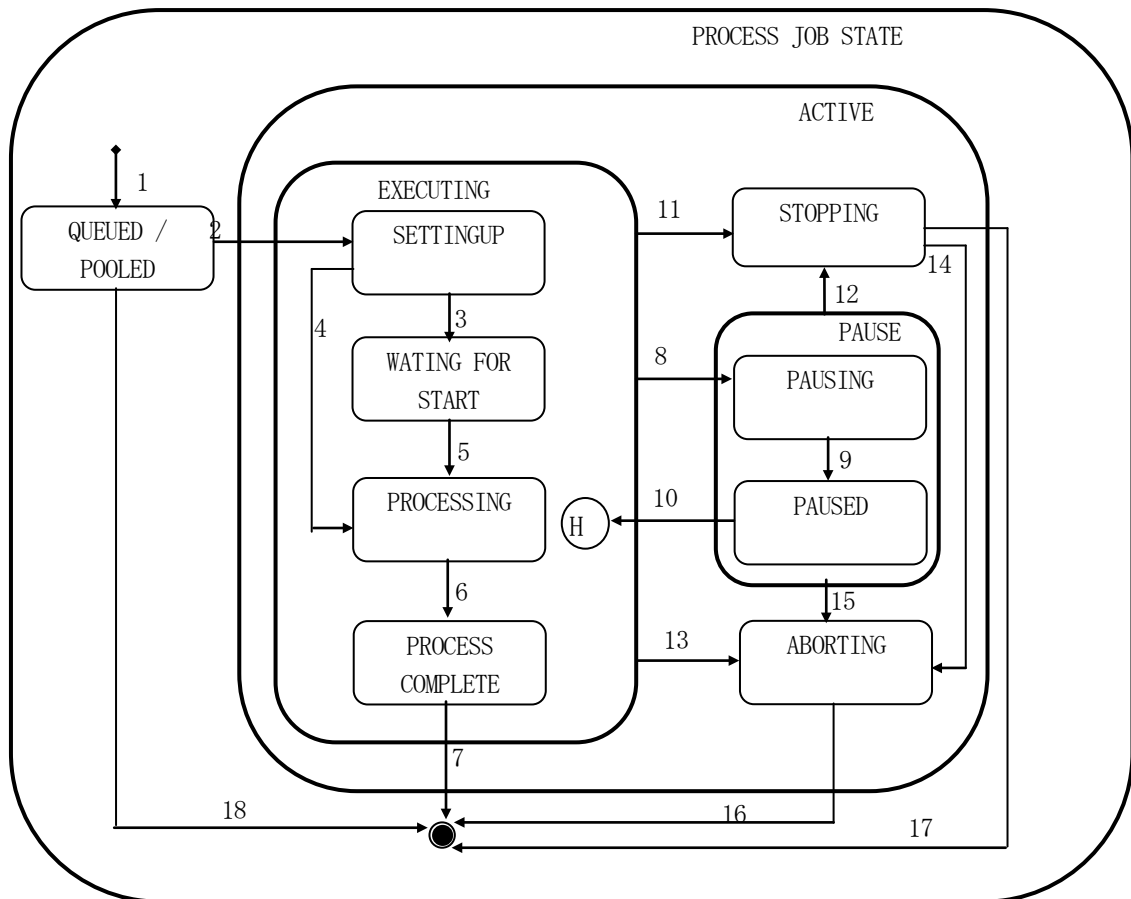


図 5.9.1 プロセスジョブ状態遷移図

(2) 状態遷移表

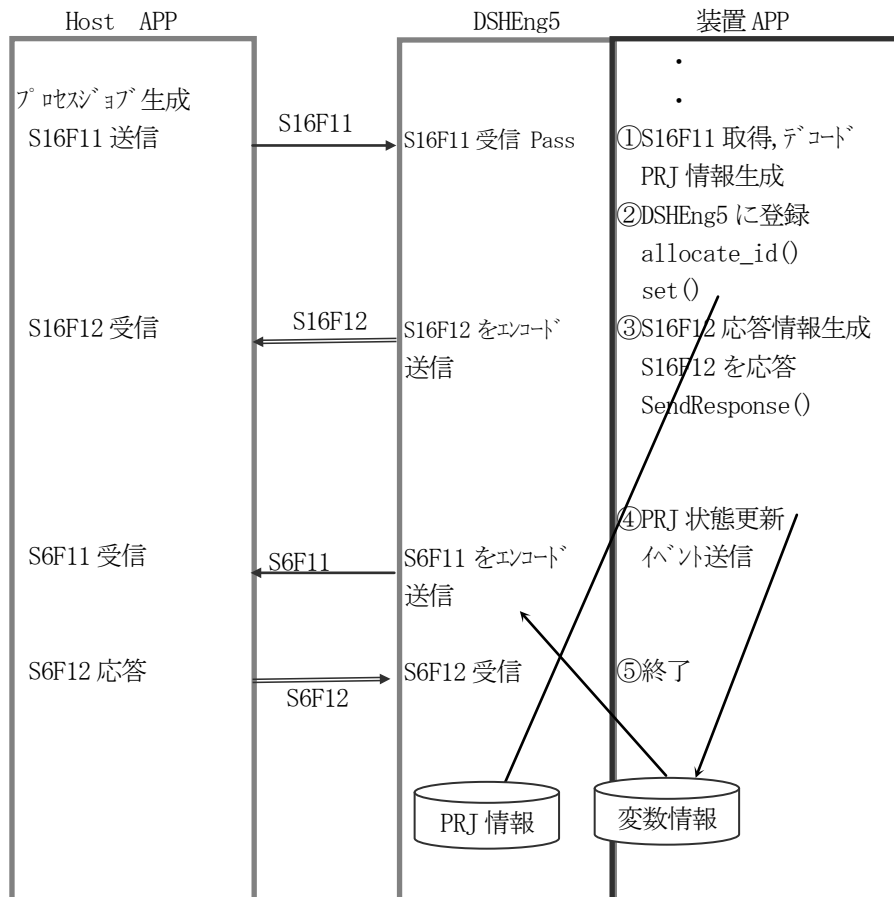
表 5.9.1 プロセスジョブ状態遷移定義表

| #  | 前ステート             | トリガー                                                                  | 新ステート             | アクション                                                                                                                       |
|----|-------------------|-----------------------------------------------------------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 1  | (状態なし)            | プロセス実行資源が PR Job 生成要求を受け付ける                                           | QUEUED/POOLED     | 1. ジョブが、ジョブキュー/プールとなる。<br>2. プロセスジョブ生成を確認する。                                                                                |
| 2  | QUEUED/POOLED     | プロセス実行資源がプロセスジョブに割当てられた。                                              | SETTING UP        | 1. プロセスジョブをジョブキュー/プールから外す。<br>2. PR ジョブセットアップイベントがトリガーされる。<br>3. すべての要求されているリリース事前処理が行われる。<br>4. ジョブ材料が到着すると、すべての材料処理が行われる。 |
| 3  | SETTING UP        | ジョブ材料が存在し、処理リリースがプロセスジョブはプロセスジョブの開始準備が完了し、PRProcessStart 属性が設定されていない。 | WAITING FOR START | スタートイベントを待っている PR Job Waiting がトリガーされる。                                                                                     |
| 4  | SETTING UP        | 材料があり、処理できる状態になった。“PRProcessStart”属性がセットされている。                        | PROCESSING        | 1. Pr Job Processing イベントがトリガーされる。<br>2. 材料を処理する。                                                                           |
| 5  | WAITING FOR START | Job Start 指令                                                          | PROCESSING        | 1. Pr Job Processing イベントがトリガーされる。<br>2. 材料を処理する。                                                                           |
| 6  | PROCESSING        | 材料の処理が終了した。                                                           | PROCESS COMPLETE  | 1. PR Job Processing Complete イベントがトリガーされる。<br>2. プロセス実行資源がすべての要求されているリリース事後処理を行う。                                          |
| 7  | PROCESS COMPLETE  | ジョブ材料がプロセス実行資源を離れ、リリース事後処理が完了した。または同一材料に対する別のジョブが取って変わった。             | (no state)        | 1. PR Job Complete がトリガーされる。<br>2. プロセスジョブが削除される。                                                                           |
| 8  | EXECUTING         | プロセス実行資源がプロセス一時停止動作を開始した。(PAUSE 命令を受取ったか、内部一時停止を開始した。)                | PAUSING           | プロセス実行資源が、最初の都合のよい時間で停止する。                                                                                                  |
| 9  | PAUSING           | プロセス実行資源がジョブを一時停止した。                                                  | PAUSED            | なし                                                                                                                          |
| 10 | PAUSED            | プロセス実行資源がジョブを再開した。                                                    | EXECUTING         | プロセス実行資源が一時的に停止した動作を再開する。                                                                                                   |

|    |               |                                                                  |            |                                                                                                                                                  |
|----|---------------|------------------------------------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 11 | EXECUTING     | プロセス実行資源がプロセス停止動作を開始した。(PAUSE 命令を受取ったか、内部一時停止を開始した。)             | STOPPING   | プロセス実行資源が現在実行中の動作を最初の都合のよいところで停止する。                                                                                                              |
| 12 | PAUSED        | プロセス実行資源がプロセス停止動作を開始した。(STOP 命令を受取ったか、内部停止を開始した。)                | STOPPING   | プロセス実行資源が現在実行中の動作を最初の都合のよいところで停止する。                                                                                                              |
| 13 | EXECUTING     | プロセス実行資源がプロセスアボート動作を開始した。(ABORT 命令を受取ったか、内部 ABORT を開始した。)        | ABORTING   | プロセス実行資源が現在実行中の動作を直ちに終了する。                                                                                                                       |
| 14 | STOPPING      | プロセス実行資源がプロセスアボート動作を開始した。(ABORT 命令を受取ったか、内部 ABORT を開始した。)        | ABORTING   | プロセス実行資源が現在実行中の動作を直ちに終了する。                                                                                                                       |
| 15 | PAUSED        | プロセス実行資源がプロセスアボート動作を開始した。(ABORT 命令を受取ったか、内部 ABORT を開始した。)        | ABORTING   | プロセス実行資源が現在実行中の動作を直ちに終了する。                                                                                                                       |
| 16 | ABORING       | プロセス実行資源がアボート手順が完了し、いつかの装置に対しては、その関連する基板がエアリカハリの一部分として移動させられている。 | (no state) | <ol style="list-style-type: none"> <li>1. PR Job Complete がトリガ-される。</li> <li>2. プロセスジョブが削除される。</li> </ol>                                        |
| 17 | STOPPING      | プロセス実行資源の停止手順が完了した。                                              | (no state) | <ol style="list-style-type: none"> <li>1. PR Job Complete がトリガ-される。</li> <li>2. プロセスジョブが削除される。</li> </ol>                                        |
| 18 | QUEUED/POOLED | "CANCEL", "ABORT" または"STOP"命令を受取った。                              | (no state) | <ol style="list-style-type: none"> <li>1. プロセスジョブを queue/pool から取り除く。</li> <li>2. PR Job Complete がトリガ-される。</li> <li>3. プロセスジョブを削除する。</li> </ol> |

## 5. 9. 2 プロセスジョブの管理と処理の流れ

概略の流れの一例です。



\*PRJ Multi-Create の場合 S16F15, S16F16 メッセージを使用

図 5.9.2 プロセスジョブ管理処理の流れ

## [プロセスジョブ関連メッセージ送信用クラス一覧]

プロセスジョブ関連メッセージ送信のためのクラスがあります。

表 5.9 プロセスジョブ関連メッセージ送信用クラス

| メッセージ  | 目的                      | 使用するクラス / メソッド                                    | ホスト(H)/装置(E) |
|--------|-------------------------|---------------------------------------------------|--------------|
| S16F5  | Process Job Cmd Request | class_SendS16F3. SendS16F3()<br>TPRJ_CMD_INFO     | H            |
| S16F11 | PrJob Create Enh        | class_SendS16F11. SendS16F11()<br>TPRJ_INFO       | H            |
| S16F15 | PrJob Multi Create      | class_SendS16F15. SendS16F15()<br>TPRJ_LIST       | H            |
| S16F17 | PrJob Deque             | class_SendS16F17. SendS16F17()<br>TPRJ_DEQ_INFO   | H            |
| S16F19 | Pr Get All Job          | class_SendS16F19. SendS16F19()<br>TPRJ_STATE_LIST | H            |
| S16F21 | Pr Get Space            | class_SendS16F21. SendS16F21()                    | H            |

## 5. 10 コントロールジョブ管理機能

### 5. 10. 1 コントロールジョブ状態モデル

(1) 状態遷移図

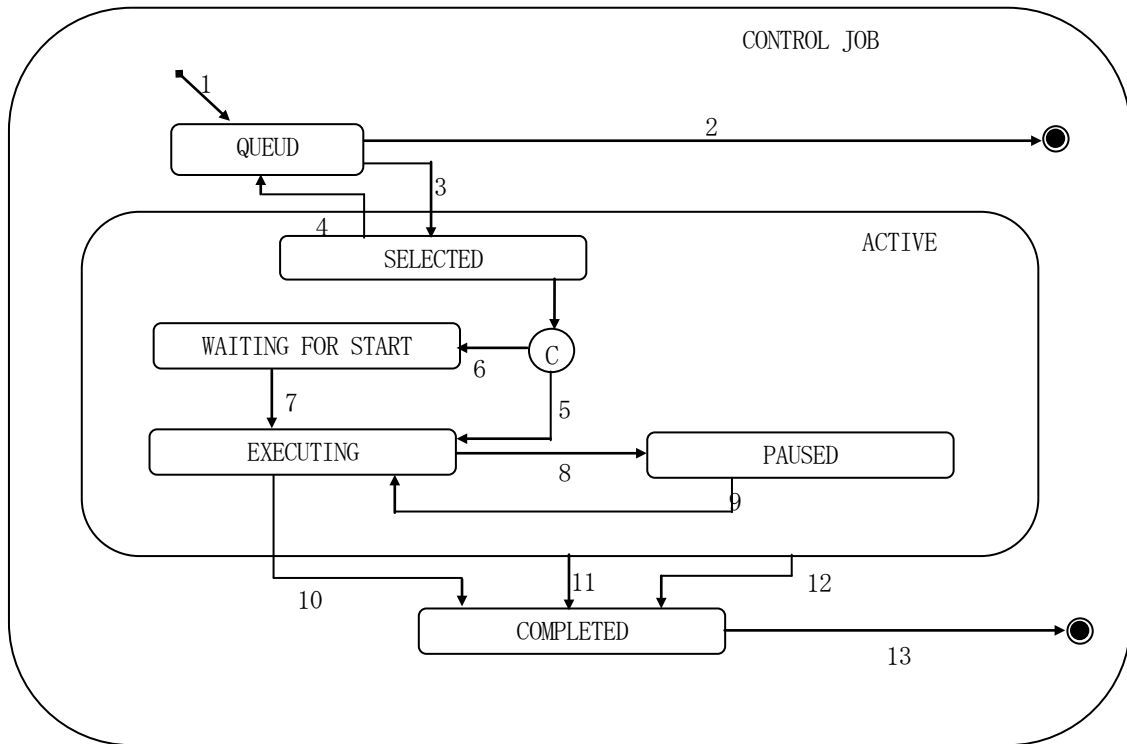


図 5. 10. 1 コントロールジョブ状態遷移図

(2) 状態遷移表

表 5.10.1 コントロールジョブ状態遷移定義表

| # | 前状態                     | トリガー                                                                                                                                                                                                   | 新しい状態                   | 処理                                                                | コメント                                                                                                        |
|---|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 1 | (No State)              | オペレータコンソールを通してホスト<br>あるいはオペレータから<br>"Create" 命令を受信。                                                                                                                                                    | QUEUED                  | ControlJob を生成しそれを<br>コントロールジョブ待ち<br>行列の最後に置く。                    | ジョブ待ち行列が一杯の<br>場合、"Create" 要求は拒<br>否される。                                                                    |
| 2 | QUEUED                  | オペレータコンソールを通してホスト<br>あるいはオペレータから<br>"Cancel", "Abort" ある<br>いは"Stop" 命令を受信。                                                                                                                            | (no state)              | 待ちを離れてジョブを<br>終了する。<br>"ControlJobCanceled"<br>イベントをホストに送信す<br>る。 | 他のコントロールジョブが待ち<br>行列中でキャンセルされたジョ<br>ブの後に待っている場合<br>は、キャンセルされたコントロ<br>ールジョブが待ちを離れた後で<br>ギャップを埋めて前にシフト<br>する。 |
| 3 | QUEUED                  | 処理資源が次の<br>ControlJob 作業を開始す<br>る量を持つ。                                                                                                                                                                 | SELECTED                | 待ちの頭にあるジョブ<br>を選択して待ちから外<br>す。"Selected" イベント<br>をホストに送信する。       | 装置においては材料を必<br>要としない。                                                                                       |
| 4 | SELECTED                | オペレータコンソールを通してホスト<br>あるいはオペレータから<br>"De-select" 命令を受<br>信,<br>コントロールジョブのための材<br>料は未着                                                                                                                   | QUEUED                  | 非選択ジョブをジョブ待<br>ち行列の頭に移動して<br>頭にあったジョブは<br>SELECTEDジョブとなる。         | この命令、待ち行列の頭<br>にあるジョブのための資<br>源が利用可能でない場合<br>は<br>拒否されなければならない。<br>Queue Model を参照                          |
| 5 | SELECTED                | 最初の (あるいは唯一の)<br>プロセッサジョブが材料を要求<br>しない場合に最初のプロセ<br>ッサジョブのための材料が到<br>着あるいはケースの中にあ<br>る、この遷移はそのプロセ<br>ッサジョブのための資源が利<br>用可能になると直ちに起<br>きなければならない。<br>ControlJob の中<br>の "StartMethod" 属性は<br>Auto に設定される。 | EXECUTING               | "Execution began" イベント<br>をホストに送信。                                | キャリアに関連するプロセッサ<br>ジョブはキャリアについての識別<br>子とウェハースロットマップが確認<br>されるまで起動しない。<br>材料を使わないプロセッサ<br>ジョブは直ちに起動できる。       |
| 6 | SELECTED                | コントロールジョブの中<br>の "StartMethod" 属性が<br>ユーザ始動に設定されてい<br>る以外は遷移5に同じ。                                                                                                                                      | WAITING<br>FOR<br>START | "Job Waiting for<br>start" イベントをホストおよ<br>び/またはオペレータに送<br>信。       |                                                                                                             |
| 7 | WAITING<br>FOR<br>START | User START 命令受信                                                                                                                                                                                        | EXECUTING               | 遷移5に同じ。                                                           | 遷移5に同じ。                                                                                                     |

|    |           |                                                                                                          |            |                                        |                                         |
|----|-----------|----------------------------------------------------------------------------------------------------------|------------|----------------------------------------|-----------------------------------------|
| 8  | EXECUTING | オペレータコンソールを通してあるいは ControlJob を通してホストあるいはオペレータから "Pause" メッセージを受信、PauseEvent が発生。                        | PAUSED     | "Paused" イベントをホストに送信。                  | 開始していないプロセスジョブにこの状態で変更可能                |
| 9  | PAUSED    | オペレータコンソールを通してホストあるいはオペレータから "Resume" メッセージを受信。                                                          | EXECUTING  | 起動プロセスジョブ開始。<br>"Resumed" イベントをホストに送信。 |                                         |
| 10 | EXECUTING | ControlJob のために規定するすべての ProcessJob が完了                                                                   | COMPLETED  | "Complete" イベントをホストに送信。                | 後処理の完了を含んでよい。                           |
| 11 | ACTIVE    | オペレータコンソールまたは ControlJob を通してホストあるいはオペレータから "SJStop" メッセージを受信あるいは ControlJob 下のすべてのプロセスジョブが停止し材料プロセスが停止。 | COMPLETED  | "Stopped" イベントをホストに送信。                 |                                         |
| 12 | ACTIVE    | オペレータコンソールを通してホストあるいはオペレータから "SJAbort" メッセージを受信あるいは ControlJob 下のすべてのプロセスジョブが中断し材料プロセスが中断。               | COMPLETED  | "Aborted" イベントをホストに送信。                 |                                         |
| 13 | COMPLETED | ControlJob の削除。                                                                                          | (No state) |                                        | 装置は COMPLETED ジョブのために自動的にこの機能を実施すべきである。 |



## 5. 10. 2 コントロールジョブの管理と処理の流れ

概略の流れの一例です。

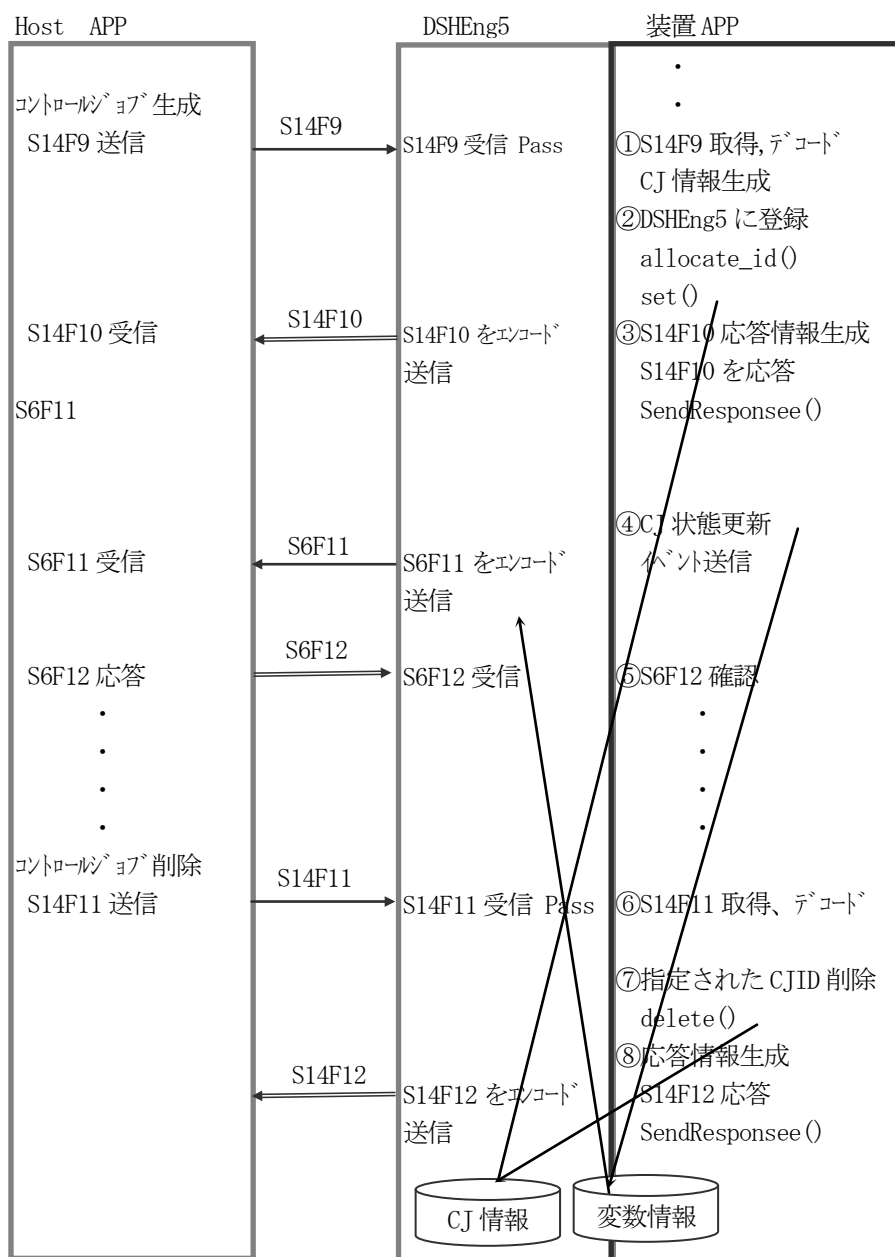


図 5. 10. 2 コントロールジョブ管理処理の流れ

## [コントロールジョブ関連メッセージ送信用クラス一覧]

コントロールジョブ関連メッセージ送信のためのクラスがあります。

表 5.10 コントロールジョブ関連メッセージ送信用クラス

| メッセージ  | 目的                          | 使用するクラス / メソッド                                 | ホスト(H)/装置(E) |
|--------|-----------------------------|------------------------------------------------|--------------|
| S14F9  | Create Object Request (Cj)  | class_SendS14F9.SendS14F9()<br>TCJ_INFO        | H            |
| S14F11 | Delete Object Request (Cj)  | class_SendS14F11.SendS14F11()<br>TCJ_INFO      | H            |
| S16F27 | Control Job Command Request | class_SendS16F27.SendS16F27 ()<br>TCJ_CMD_INFO | H            |

## 5. 11 端末サービス機能

装置はホストに端末情報を送信することができます。

S10F1, S10F2 メッセージが使用されます。

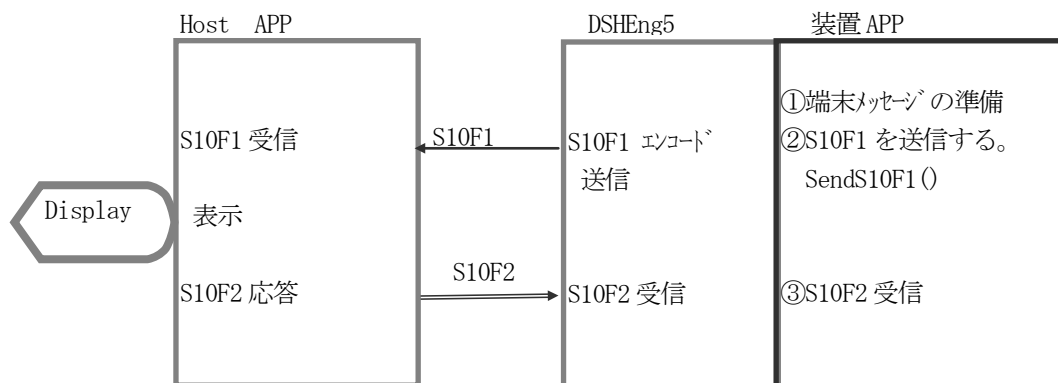


図 5.11 S10F1 端末要求の流れ

### [端末サービス関連メッセージ送信用クラス一覧]

端末サービス関連メッセージ送信のためのクラスがあります。

表 5.11 端末サービス関連メッセージ送信用クラス

| メッセージ | 目的            | 使用するクラス / メソッド                                    | ホスト(H)/装置(E) |
|-------|---------------|---------------------------------------------------|--------------|
| S10F1 | 端末要求          | class_SendS10F1. SendS10F1 ()                     | E            |
| S10F3 | 端末表示、シングルブロック | class_SendS10F3. SendS10F3 ()                     | H            |
| S10F5 | 端末表示、マルチブロック  | class_SendS10F5. SendS10F5 ()<br>TTERM_MULTI_INFO | H            |

## 5. 12 装置に対するリモートコマンドメッセージ送信処理

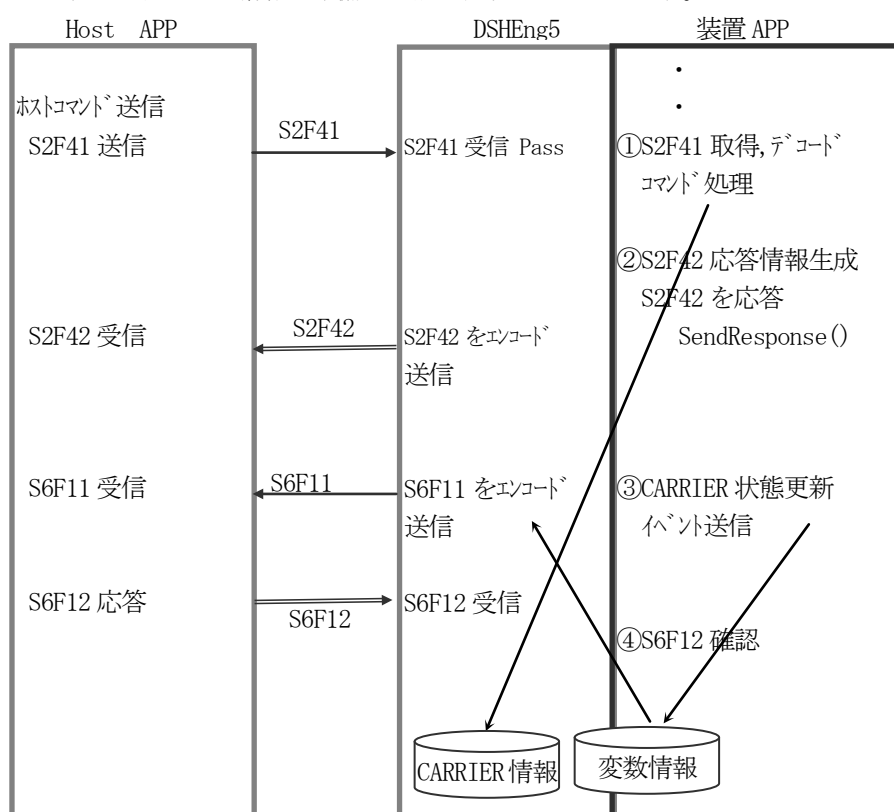
ここではこれまで直接記述対象にならなかったホストから発信される以下の要求コマンドメッセージの送信処理について説明します。

|   | メッセージ | 機能                      |
|---|-------|-------------------------|
| 1 | S2F41 | ホストコマンド送信               |
| 2 | S2F49 | Enhanced Remote Command |

S2F41 または S2F49 メッセージを使ってリモートコマンドを装置に送信します。

DSHEng5 はこれらのメッセージを送信するためのクラスとメッセージ組み立てのためのライブラリ関数を提供します。

DSHEng5 では次のようにメッセージ情報の準備と送信を行うことになります。



\*Enhanced Remote Command の場合 S2F49, S2F50 を使用

図 5.12 ホストコマンド S2F41 処理の流れ

### [リモートコマンド関連メッセージ送信用クラス一覧]

リモートコマンド関連メッセージ送信のためのクラスがあります。

表 5.12 リモートコマンド関連メッセージ送信用クラス

| メッセージ | 目的         | 使用するクラス | ホスト(H)/装置(E) |
|-------|------------|---------|--------------|
| S2F41 | ホストコマンド送信  |         | H            |
| S2F49 | 拡張リモートコマンド |         | H            |

関連 API、ライブラリ関数の詳細については「APP インタフェース ライブラリ関数説明書」を参照ください。

## 付録-A DSHEng5 - SECS-II 処理MSG 一覧表

表の欄の記述の意味は右のとおりです。

DSHEng5 欄： ○はDSHEng5 が送受信処理をする。  
 ENG\_APP 欄： ○はAPP が受信処理する。  
 ◎はAPP がDSHEng5 クラスを使って要求する。(送信, 制御要求)  
 備考欄： API は要求をDSHEng5 クラスで要求する。  
 DSHDR2 は通信ドライバが処理する。

1次メッセージ処理分担一覧表

| 1次MSG | 2次MSG | 方向   | 役割                      | DSHEng5 | XXX_APP | 備考 |
|-------|-------|------|-------------------------|---------|---------|----|
| S1F1  | S1F2  | H←→E | オンライン確認                 | ○       | ◎       |    |
| S1F3  | S1F4  | H→E  | 装置状態要求                  |         | ◎       |    |
| S1F11 | S1F12 | H→E  | 状態変数一覧要求                |         | ◎       |    |
| S1F13 | S1F14 | H←→E | 通信確立                    | ○       | ◎       |    |
| S1F15 | S1F16 | H→E  | オンライン要求                 |         | ◎       |    |
| S1F17 | S1F18 | H→E  | オンライン要求                 |         | ◎       |    |
|       |       |      |                         |         |         |    |
| S2F13 | S2F14 | H→E  | 装置定数要求                  |         | ◎       |    |
| S2F15 | S2F16 | H→E  | 装置定数変更                  |         | ◎       |    |
| S2F17 | S2F18 | H←→E | 時刻要求                    |         | ◎       |    |
| S2F23 | S2F24 | H→E  | トレース条件設定                |         | ◎       |    |
| S2F29 | S2F30 | H→E  | 装置定数名一覧要求               |         | ◎       |    |
| S2F31 | S2F32 | H→E  | 時刻設定要求                  |         | ◎       |    |
| S2F33 | S2F34 | H→E  | レポート設定                  |         | ◎       |    |
| S2F35 | S2F36 | H→E  | イベントレポート設定              |         | ◎       |    |
| S2F37 | S2F38 | H→E  | イベントレポート有効/無効設定         |         | ◎       |    |
| S2F39 | S2F40 | H→E  | マルチブロッグ問合せ              | ○       |         |    |
| S2F41 | S2F42 | H→E  | ホストコメント送信               |         | ◎       |    |
| S2F43 | S2F44 | H→E  | スプールの設定                 |         | ◎       |    |
| S2F45 | S2F46 | H→E  | 変数リミット属性定義              |         | ◎       |    |
| S2F47 | S2F48 | H→E  | 変数リミット属性一覧要求            |         | ◎       |    |
| S2F49 | S2F50 | H→E  | Enhanced Remote Command |         | ◎       |    |
|       |       |      |                         |         |         |    |
| S3F17 | S3F18 | H→E  | キャリアアクション要求             |         | ◎       |    |
| S3F23 | S3F24 | H→E  | ポートグループアクション要求          |         | ◎       |    |
| S3F25 | S3F26 | H→E  | ポートアクション要求              |         | ◎       |    |
| S3F27 | S3F28 | H→E  | Change Access           |         | ◎       |    |
|       |       |      |                         |         |         |    |
| S5F1  | S5F2  | H←E  | アラーム報告                  |         | ○       |    |
| S5F3  | S5F4  | H→E  | アラーム有効/無効設定             |         | ◎       |    |
| S5F5  | S5F6  | H→E  | アラームリスト要求               |         | ◎       |    |
|       |       |      |                         |         |         |    |
| S6F1  | S6F2  | H←E  | トレースデータ送信               |         | ○       |    |
| S6F5  | S6F6  | H←E  | マルチブロッグデータ問合せ           | ○       |         |    |
| S6F11 | S6F12 | H←E  | イベントレポート送信              |         | ○       |    |

|        |        |      |                             |   |    |        |
|--------|--------|------|-----------------------------|---|----|--------|
| S6F13  | S6F14  | H←E  | 注釈付きイベントポート送信               |   | ○  |        |
| S6F15  | S6F16  | H→E  | イベントポート要求                   |   | ◎  |        |
| S6F19  | S6F20  | H→E  | 個別ポート要求                     |   | ◎  |        |
| S6F23  | S6F24  | H→E  | スプールデータ要求                   |   | ◎  |        |
|        |        |      |                             |   |    |        |
| S7F1   | S7F2   | H←→E | プロセスログラムポート問合せ              |   | ◎◎ |        |
| S7F3   | S7F4   | H←→E | プロセスログラム送信                  |   | ◎◎ |        |
| S7F5   | S7F6   | H←→E | プロセスログラム要求                  |   | ◎◎ |        |
| S7F17  | S7F18  | H→E  | プロセスログラム削除指示                |   | ◎  |        |
| S7F19  | S7F20  | H→E  | プロセスログラム一覧要求                |   | ◎  |        |
| S7F23  | S7F24  | H←→E | フォーマット付プロセスログラム送信           |   | ◎◎ |        |
| S7F25  | S7F26  | H←→E | フォーマット付プロセスログラム要求           | ○ | ◎  |        |
| S7F27  | S7F28  | H←E  | プロセスログラム妥当性送信               |   | ○  |        |
| S7F29  | S7F30  | H←E  | プロセスログラム妥当性問合せ              |   | ○  |        |
|        |        |      |                             |   |    |        |
| S9F1   | -      | H←E  | 未定義デバイスID                   | ○ |    | DSHDR2 |
| S9F3   | -      | H←E  | 未定義ストリームタイプ                 |   | ○  | “      |
| S9F5   | -      | H←E  | 未定義ファンクションタイプ               |   | ○  | “      |
| S9F7   | -      | H←E  | 不正データ                       |   | ○  | “      |
| S9F9   | -      | H←E  | T3タイムアウト                    | ○ |    | “      |
| S9F11  | -      | H←E  | データが長すぎる                    |   | ○  | “      |
| S9F13  | -      | H←E  | 会話タイムアウト                    |   | ○  | “      |
|        |        |      |                             |   |    |        |
| S10F1  | S10F2  | H←E  | 端末要求                        |   | ○  |        |
| S10F3  | S10F4  | H→E  | 端末表示、シングルブロック               |   | ◎  |        |
| S10F5  | S10F6  | H→E  | 端末表示、マルチブロック                |   | ◎  |        |
|        |        |      |                             |   |    |        |
| S14F9  | S14F10 | H→E  | Create Object Request (Cj)  |   | ◎  |        |
| S14F11 | S14F12 | H→E  | Delete Object Request (Cj)  |   | ◎  |        |
|        |        |      |                             |   |    |        |
| S15F1  | S15F2  | H←→E | Recipe management m-blk inq | ○ |    |        |
| S15F3  | S15F4  | H←→E | Recipe Namespace action Req |   | ◎◎ |        |
| S15F5  | S15F6  | H←→E | Recipe Namespace rename Req |   | ◎◎ |        |
| S15F7  | S15F8  | H←→E | Recipe Space Request        | ○ | ◎  |        |
| S15F9  | S15F10 | H←→E | Recipe Status Request       | ○ | ◎  |        |
| S15F13 | S15F14 | H←→E | Recipe Create Request       |   | ◎◎ |        |
| S15F17 | S15F18 | H←→E | Recipe Retrieve Req         | ○ | ◎  |        |
|        |        |      |                             |   |    |        |
| S16F5  | S16F6  | H→E  | Process Job Cmd Request     |   | ◎  |        |
| S16F11 | S16F12 | H→E  | PrJob Create Enh            |   | ◎  |        |
| S16F15 | S16F16 | H→E  | PrJob Multi Create          |   | ◎  |        |
| S16F17 | S16F18 | H→E  | PrJob Deque                 |   | ◎  |        |
| S16F19 | S16F20 | H→E  | Pr Get All Job              |   | ◎  |        |
| S16F21 | S16F22 | H→E  | Pr Get Space                |   | ◎  |        |
| S16F27 | S16F28 | H→E  | Control Job Command Request |   | ◎  |        |

## 付録-B DSHEng5 装置起動ファイルコマンド

起動ファイルはDSHEng5 エンジン起動時にエンジンが動作する環境条件を指定するテキスト形式のファイルです。起動ファイル上には以下のコマンドを使って環境条件情報を定義します。

| #  | コマンド名とフォーマット           | 機能                                      | コマンド例                                    |
|----|------------------------|-----------------------------------------|------------------------------------------|
| 1  | LOG_PATH = <ディレクトリ名>   | 個別装置のログファイルの保存ディレクトリ名を指定します。            | LOG_PATH = c:\¥DSHEng5¥log               |
| 2  | LOG_FILE = <ファイル名>     | 個別装置のログファイル名を指定します。                     | LOG_FILE = "gem_host.log"                |
| 3  | LOG_SIZE = <行数>        | ログファイルに保存する最大行数を指定します。                  | LOG_SIZE = 100000                        |
| 4  | BKUP_PATH = <ディレクトリ名>  | 装置管理情報のバックアップファイルを保存するディレクトリを指定します。     | BKUP_PATH = "C:\¥DSHEng5¥backup"         |
| 5  | INFO_FILE = <ファイル名>    | 装置管理情報定義ファイル名をフルパスで指定します。               | INFO_FILE =<br>"c:\¥ DSHEng5¥eqinfo.fil" |
| 6  | INFO_BACKUP = <1/0>    | 装置管理情報のバックアップを行うかどうかを 1,0 で指定します。       | BACKUP_FLAG = 1                          |
| 7  | SPOOL_PATH = <ディレクトリ名> | SPOOL ファイルを保存するディレクトリを指定します。            | SPOOL_PATH = "C:\¥DSHEng5¥spool"         |
| 8  | PP_COUNT = <n>         | PP(プロセスプログラム)最大管理数を n 個にします。(S7F3)      | PP_COUNT = 64                            |
| 9  | FPP_COUNT = <n>        | FPP(書式付プロセスプログラム)最大管理数を n 個にします。(S7F23) | PP_COUNT = 80                            |
| 10 | RCP_COUNT = <n>        | RECIPE 最大管理数を n 個にします。(S15F13)          | RCP_COUNT = 80                           |
| 11 | CAR_COUNT = <n>        | CARRIER 最大管理数を n 個にします。                 | CAR_COUNT = 16                           |
| 12 | SUBST_COUNT = <n>      | SUBSTRATE 最大管理数を n 個にします。               | SUBST_COUNT = 250                        |
| 13 | PRJ_COUNT = <n>        | PRJ(プロジェクト)最大管理数を n 個にします。              | PRJ_COUNT = 16                           |
| 14 | CJ_COUNT = <n>         | CJ(コントロールジョブ)最大管理数を n 個にします。            | CJ_COUNT = 16                            |
| 15 | TRACE_COUNT = <n>      | TRACE 最大管理数を n 個にする。                    | TRACE_COUNT = 15                         |
| 16 | CAR_CAPACITY = <n>     | 1 個のキャリアの最大収納ウェハ枚数を n 個にします。            | CAR_CAPACITY = 25                        |
| 17 | COMM_SIDE = <サイト名>     | 通信サイトを HOST/EQUIPMENT で指定します。           | COMM_SIDE = HOST<br>(固定)                 |
| 18 | COMM_PORT = <ポート番号>    | DSHDR2 通信ドライバで使用するポート番号を指定します。          | PORT = 1                                 |
| 19 | COMM_DEVICE = <デバイス番号> | DSHDR2 通信ドライバで使用するデバイス番号を指定します。         | DEVICE = 1                               |
| 20 | S1F13_SEND = <0/1/2>   | 通信確立方法を定義する。<br>ホストのみ、ホスト主導、通常          | S1F13_SEND = 2                           |

## 付録-C バックアップ対象情報と更新

バックアップ対象になる情報は以下の情報です。バックアップ情報は、装置起動ファイルの BKUP\_PATH コマンドで指定されたディレクトリに保存されます。(付録-B参照)

バックアップ対象管理情報は次の情報です。(バックアップは無条件に行なわれます。)

|    | 情報名                                  | ファイル名                                           |
|----|--------------------------------------|-------------------------------------------------|
| 1  | 変数定義情報<br>装置定数<br>装置状態変数<br>装置データ変数) | ec_bkup0. bkp<br>sv_bkup0. bkp<br>dv_bkup0. bkp |
| 2  | レポート情報                               | rp_bkup0. bkp                                   |
| 3  | 収集イベント情報                             | ce_bkup. bkp                                    |
| 4  | プロセスプログラム情報                          | pp_bkup0. bkp                                   |
| 5  | フォーマット付きプロセスプログラム情報                  | fpp_bkup0. bkp                                  |
| 6  | レシピ情報                                | rcp_bkup0. bkp                                  |
| 7  | キャリア情報                               | car_bkup0. bkp                                  |
| 8  | 基板情報                                 | subst_bkup0. bkp                                |
| 9  | コントロールジョブ情報                          | cj_bkup0. bkp                                   |
| 10 | プロセスジョブ情報                            | prj_bkup0. bkp                                  |

バックアップは、各管理情報が、クラスで値が更新された後、1秒間以内に行なわれます。

バックアップ情報は、最大4世代前までの情報がファイルに保存されます。世代は、各管理情報についてファイルが更新される度にファイルの世代も更新されます。

バックアップファイル名の末尾に世代番号が付けられます。世代番号は、0が最新のバックアップ情報であり、3が最も古い世代になります。