

DSHEng5 装置通信エンジン (GEM+GEM300)

ソフトウェア・パッケージ

アプリケーション・プログラム開発への手引き

2019年6月

株式会社データマップ

[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.	2019-06-28	初版	
2.			
3.			
4.			

目 次

1.概要.....	1
2.作業の流れ.....	2
3.通信環境定義ファイル(COMM. DEF)の作成.....	3
4.装置起動情報定義ファイルの作成.....	5
5.装置変数情報定義ファイルの作成とコンパイル.....	8
5. 1 装置変数情報の定義と定義ファイルの作成.....	9
5. 2 装置変数情報のコンパイル.....	12
5. 3 コンパイルによって生成されたヘッダーファイル例.....	13
6.アプリケーションプログラムの作成.....	14
6. 1 通信エンジン開始処理.....	15
6. 1. 1 DSHEng5 通信エンジンの起動(初期化)処理.....	15
6. 1. 2 予約変数の登録.....	16
6. 1. 3 ユーザプログラム処理対象受信1次メッセージの登録.....	17
6. 1. 4 通信確立の処理.....	17
6. 2 通常処理.....	18
6. 2. 1 変数のアクセス.....	18
6. 2. 1. 1 変数情報の管理場所とアクセスの方法.....	19
6. 2. 2 受信1次メッセージの処理.....	20
6. 2. 3 1次メッセージの送信.....	22
[SECS-IIメッセージと送信クラス一覧表]	23
6. 2. 4 収集イベント送信処理について.....	26
6. 2. 5 アラーム送信処理について.....	27
6. 2. 6 スプール機能処理について.....	28
6. 2. 7 トレース機能処理について.....	29
6. 2. 8 リミット監視とホスト通知処理について.....	30
6. 3 通信エンジン終了処理.....	31
<付録>.....	32
付録-A DSHEng5 通信制御エンジンライブラリ関連文書一覧表.....	32
付録-A1 DSHEng5 通信制御エンジンライブラリ関連文書一覧表.....	32
付録-A2 DSHEng5 GEM通信エンジン・クラス説明書.....	33
付録-A3 DSHEng5 エンジン・デモ・プログラム関連文書一覧表.....	33
付録-B SEMI スタンダードへの対応.....	34
付録-C 装置変数情報とバックアップファイル.....	35
付録-D SECS-IIの通信制御処理.....	36

1. 概要

本説明書は、ユーザが GEM、GEM300 サポート、DSHEng5 装置通信エンジンライブラリを半導体製造工場のシステムに導入する際、DSHEng5 を理解し、さらに、システム的设计、ソフトウェア開発を進めるための手助けとなるべき基本的な事項について具体的な手順と作業内容を説明するものです。

なお、DSHEng5 関連資料として **付録一A** の資料を参照することができます。

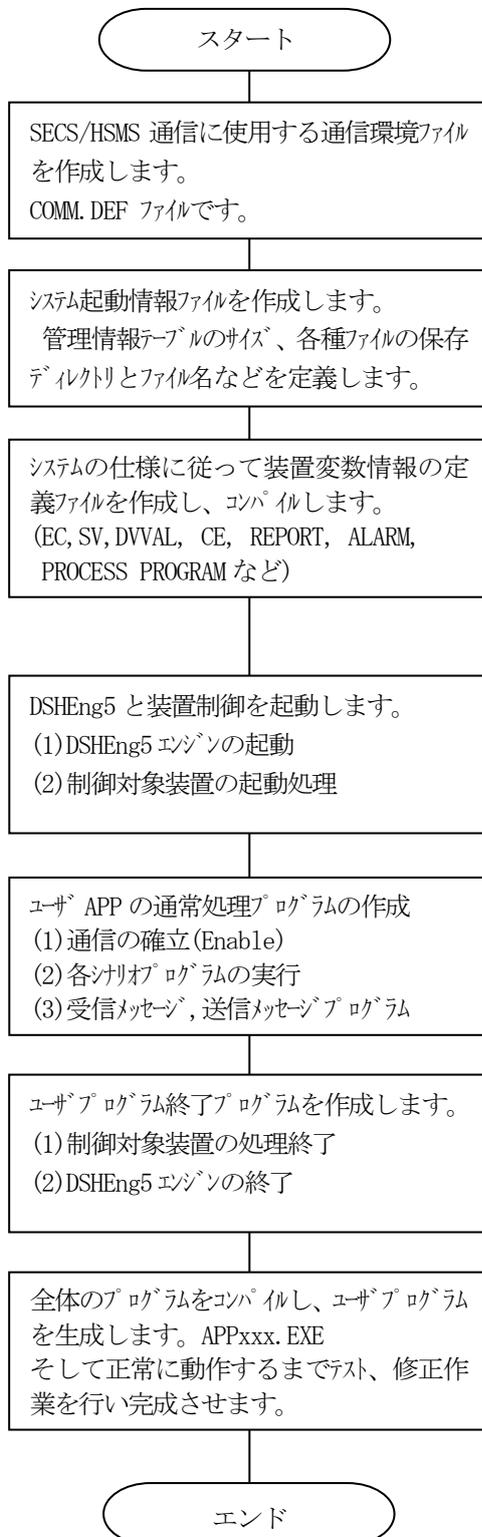
次章、「2. 作業の流れ」のフローチャートに従って作業を進めることになります。

[注釈]

以下、具体的なプログラミング説明においては、DSHEng5 エンジンが提供する .Net C# 言語のクラスを使っての記述になっています。

2. 作業の流れ

システムの外部仕様決定後、ソフトウェア設計そしてプログラミングを含めユーザーが行うための作業の大まかな流れを表します。



「DSHDR2 SECS/HSMS レベル2 通信制御ドライバユーザーズマニュアル」参照
装置が通信で使用する PORT, DEVICE の通信パラメータを COMM.DEF ファイル上に定義します。

DSHEng5 の起動時に必要な起動情報定義ファイルを作成します。装置単位で準備します。

「DSHEng5 装置変数情報定義仕様書」参照
編集プログラム、DshGemSet5.ee で作成編集可能
「装置変数情報定義ファイルコンパイル説明書」参照
CEID, REORT ID, VID とのリンク定義が容易にできます。
定義ファイルをコンパイルしオブジェクトファイルを生成します。

以下、ユーザ APP プログラムを作成します。
DSHEng5 内部情報と装置処理起動のための初期化処理です。
EngStart() 関数を使用します。

DSHEng5 API ならびにライブラリ関数を使ってプログラミングします。
通信の確立を行い、シリオの実行処理を行うためのプログラムです。
デモプログラム(Visual C++)を参考にしてください。

EngStop() 関数を使って行います。

- ・テストは DSHSIM シミュレータを使って基本的な通信テストを行います。
- ・DSHEng5 を使って対向する相手装置 (ホスト) の APP テストプログラムを作成し、実機に近い形態でテストを行います。
(DSHEng5 は装置/ホスト双方の通信機能を有しています。
対向用の通信エンジンは開発版を使用することができます)

3. 通信環境定義ファイル(COMM.DEF)の作成

通信環境定義ファイルは、DSHDR2.DLL 通信ドライバーに対し、通信ポート、デバイス ID (=セッション ID) プロトコルタイマー値などの通信環境情報を設定するためのテキストファイルです。本ファイル名は通信エンジンの EngAPI クラスの start() メソッドの引数になります。

「DSHDR2 SECS/HSMS レベル-2 通信制御ドライバーユーザーズマニュアル」の 3. 環境ファイル仕様 に従って作成します。

ファイルには、基本的にコマンド形式で設定項目とその値を表現します。3 種類のコマンドがあります。START メインコマンド行から始まり、END コマンドで終る行の間に指定するコマンドです。

(1) 一般的なコマンド

```
START DSH
  MAX_MSG_SIZE = x100040          # 最大メッセージサイズ (バイト数)          1MB+40
  MAX_TRANSACTION = 512           # 管理する最大トランザクション数 2 の n 乗の値
  LOG_LINE = 100000              # 1 個のログファイルに保存する最大行数(
  LOG_FILE = ¥DSHEng5¥log¥DSHDR2.LOG # ログファイルを保存するファイル名 (フルパス名)
  LOG_MODE = 0                   # save old log file 4 世代分のログファイルを残す
  MON_PORT = 9999                # 通信ログモニタープログラムとの TCP/IP 接続ポートを指定
END
```

MAX_MSG_SIZE の値は、仕様上の SECS-II メッセージのバイトサイズとログファイル関連のコマンドについて必要に応じて値を変えてください。

(2) ポート定義コマンド

DSHDR2 通信ドライバー上における通信ポートの番号とそのパラメータを定義します。

相手装置との通信は SECS-I または SECS-SS プロトコルで行いますが、装置との通信のためのポート番号です。(TCP/IP のポートではありません。) この通信ポートに 1 個以上のデバイスを定義することができます。ここでは、例として HSMS-SS プロトコルについて説明します。

```
START PORT
  PORT = 1                        # HSMS-SS のポート番号
  PROTOCOL = HSMS                 # HSMS-SS を指定
  PORT_MODE = ACTIVE              # Active ポート (Passive ポートでは Passive)
  TCP_PORT = 5001                 # このポートが使用する TCP/IP のポート
  IP = 192.168.1.2                # 接続相手の Passive ポートの IP アドレス
  T3 = 45                         # T3 タイムアウト時間 (秒単位, 小数点 1 位まで有効)
  T5 = 10                         # T5 “
  T6 = 5                          # T6 “
  T7 = 10                         # T7 “
  T8 = 5                          # T8 “
  LINKTEST = 60                   # Linktest の周期(秒単位)
END
```

(3) デバイス定義コマンド

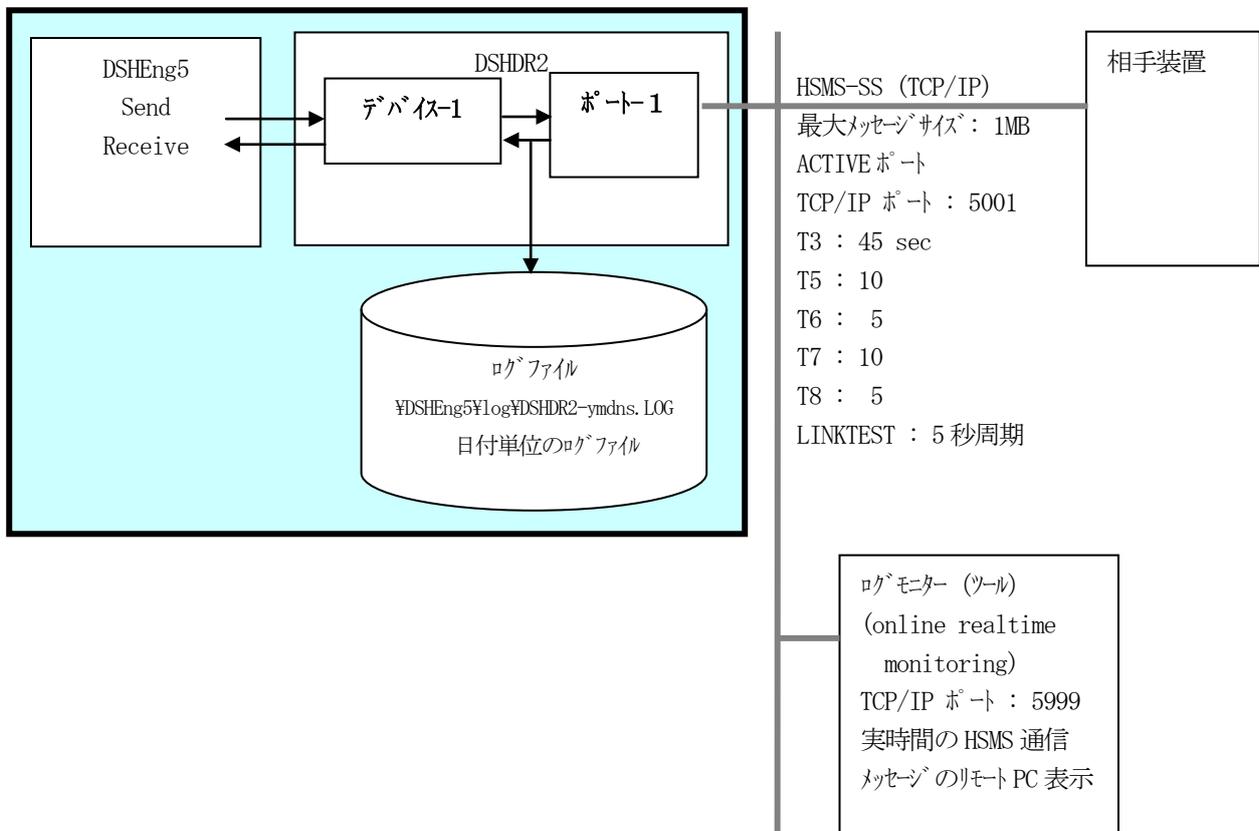
DSHDR2 通信ドライバー上における通信デバイスの番号とそのパラメータを定義します。

相手装置との通信は SECS-I または SECS-SS プロトコルで行いますが、(2) で説明したポート番号に含まれ、固有のデバイス ID (セッション ID) を有するデバイス番号です。

```
START DEVICE
  DEVICE = 1                # デバイス番号
  DVID = x1234              # デバイス ID (= セッション ID) 頭の x は 16 進表現の意
  PORT = 1                  # using port-1
END
```

DSHEng5 プログラムは相手装置との送受信をデバイス単位で行います。

以上述べてきたポート、デバイスなどのコマンドの関連を下図に示します。



4. 装置起動情報定義ファイルの作成

装置起動情報定義ファイルは、DSHEng5 の構成情報などの起動情報を設定するファイルで、ファイル拡張子 “.CNF” を持つテキストファイルです。

本定義ファイルに使用するコマンドと定義の仕方については、「DSHEng5 起動ファイル定義仕様書」を参照ください。

本定義ファイルの編集は、使用する DshGemSet5 プログラムを使って行うことができます。DshGemSet5 の機能、操作方法については、「DSHEng5, DSHEng4 起動ファイル、装置変数情報ファイル設定・編集プログラム説明書」を参照ください。コマンドは、次表の通りです。

#	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>	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:\dsheng4\log
2	LOG_MODE = <モード>	日付単位か、世代単位でのログ保存モードを指定します。	LOG_MODE = 0 (世代単位) LOG_MODE = daily (日付単位)
3	LOG_LIFE = <月数>	日付単位での古いファイルの保存期間を月数で指定する。(デフォルト=6)	LOG_LIFE = 6
4	LOG_FILE = <ファイル名>	装置のログファイル名を指定します。	LOG_FILE = “equip. log”
5	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
6	BKUP_PATH = <ディレクトリ名>	装置変数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:\dsheng4\backup”
7	INFO_FILE = <ファイル名>	装置変数情報定義ファイル名をフルパスで指定します。	INFO_FILE = “c:\dshgemfil\eqinfo. fil”
8	INFO_BACKUP = <1/0>	装置変数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
9	SPOOL_PATH = <ディレクトリ名>	装置変数情報の 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:\dsheng4\spool”
10	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
11	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
12	RCP_COUNT = <n>	RECIPE 最大管理数を n 個にします。(S15F13)	RCP_COUNT = 80
13	CAR_COUNT = <n>	CARRIER 最大管理数を n 個にします。	CAR_COUNT = 16
14	SUBST_COUNT = <n>	SUBSTRATE 最大管理数を n 個にします。	SUBST_COUNT = 250
15	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
16	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
17	TRACE_COUNT = <n>	TRACE 最大管理数を n 個にする。	TRACE_COUNT = 15
18	CAR_CAPACITY = <n>	1 個のキャリアの最大収納エーハ枚数を n 個にします。	CAR_CAPACITY = 25

19	COMM_PORT = <ポート番号>	DSHDR2 通信ドライバーで使用するポート番号を指定します。	PORT = 1
20	COMM_DEVICE = <デバイス番号>	DSHDR2 通信ドライバーで使用するデバイス番号を指定します。	DEVICE = 1
21	S1F13_SEND = <0/1/2>	通信確立方法を指定する。 (S1F13 のやり取りと確立)	0 : ホストからの S1F13 だけを待機する。 S1F14 ack=0 を返却し、無条件通信確立とする。 1 : ホストからの S1F13 については、上記 0 と同様に、自身の ENABLE によって、S1F13 を送信し、ack=0 の S1F14 受信によっても通信確立とする。 2 : ENABLE によって S1F13 を送信し、ack=0 の S1F14 受信で通信確立とする。 また、Enable の後、S1F13 受信で ack=0 応答し、通信確立とする。
22	RP_LISTV = <0/1>	Report にフォーマット = LIST が指定された場合の装置変数情報の定義の仕方を指定する。	0 : 旧仕様(ネスティングできない) 1 : LIST 変数を指定すれば、その LIST 変数にリンクできる(デフォルト)
23	NO_MBLK	S6F5 などの SECS-II でマルチブロックのメッセージを送信する際、予め Multi-Block Enquire メッセージを送信しないようにするために使用	0 : Multi-Block Enquire を送信する。 1 : Multi-Block Enquire を送信しない。 デフォルトは 0 (送信する)です。 (本コマンドが無い場合)
24	WBIT_S6F1	S6F1 送信時、W-bit の選択設定をする。 W-bit=0 の場合は、応答メッセージを期待しない。	0 : S6F1 の W-bit = 0 で送信する。 1 : S6F1 の W-bit = 1 で送信する。 デフォルトは 1 です。
25	WBIT_S6F11	S6F11 送信時、W-bit の選択設定をする。 W-bit=0 の場合は、応答メッセージを期待しない。	0 : S6F11 の W-bit = 0 で送信する。 1 : S6F11 の W-bit = 1 で送信する。 デフォルトは 1 です。
26	LEN_ALT_X	S5F1 の ALT_X のバイト長を設定する。	値の範囲は 40~128 とする。 デフォルトは 40 バイトです。 < 40 の場合は = 40 に >128 の場合は = 128 に設定する。
27	TRACE_STIME	S6F1 のトレーサデータの中の STIME のフォーマットを指定する。	12 : "YYMMDDHHNNSS" 12 桁表現 その他 : デフォルト "YYYYDDHHNNSSCC" 16 桁

(注) 装置変数、収集イベント、レポート、アラーム情報に関する登録個数は、装置変数情報ファイル内に定義される数が登録数になります。

次の画面は、DshGemSet5.exe 編集プログラムでの起動ファイルの編集画面です。

[DSHEng5 用設定画面]

no.	コメント	設定値	コメントの意味
1	LOG_PATH	.\log	ログファイル用ディレクトリ
2	LOG_MODE	DAILY	ログ収集モード(DAILY=日単位)/PLAIN
3	LOG_LIFE	6	ログ DAILY収集モードの場合のファイル寿命(月単位)
4	LOG_FILE	6	ログ PLAIN収集モードの場合のログファイル名
5	LOG_SIZE	100000	ログ PLAIN収集モードの場合ログファイル最大サイズ(行)
6	BKUP_PATH	.\backup	情報バックアップ用ディレクトリ
7	INFO_FILE		GEM装置変数データ定義ファイル名
8	INFO_BACKUP	0	GEM情報バックアップ指定(1 / 0)
9	PP_COUNT	0	プロセスプログラム最大保存数
10	FPP_COUNT	0	書式付プロセスプログラム最大保存数
11	RCP_COUNT	0	レジスタ最大保存数
12	CAR_COUNT	0	キャリア最大保存数
13	CAR_CAPACITY	0	キャリア容量(ウェハー枚数)
14	SUBST_COUNT	0	基板最大保存数
15	CJ_COUNT	0	コントロールジョブ最大保存数
16	PRJ_COUNT	0	プロセスジョブ最大保存数
17	TRACE_COUNT	0	トレース情報最大保存数
18	SPOOL_PATH	.\spool	スプール情報保存ディレクトリ
19	COMM_SIDE	HOST	ホスト/装置サイトの指定(HOST/EQUIP)DSGEMLIBのみ適用
20	COMM_PORT	0	HSMS通信ポート(DSHDR2のポート)
21	COMM_DEVICE	0	HSMS通信デバイス(DSHDR2のデバイス)
22	SIF13_SEND	0	通信確立方法 - 0 or 1=ホスト主導, 2=通常
23	MON_ENABLE	0	エンジンモニター接続有無(0=無, 1=有)
24	MON_PORT	0	エンジンモニター接続に使用するDSHDR2のPORT
25	MON_DEVICE	0	エンジンモニター接続に使用するDSHDR2のDEVICE
26	NO_MBLK	0	Multi-Block Inquireの無効(S6F5など) 1=無効
27	LEN_ALTX	0	ALTXの長さのバイト長を設定する(S5F1)

5. 装置変数情報定義ファイルの作成とコンパイル

次に、GEM スタンダードに含まれ管理対象となり、初期設定が必要な変数情報の定義ファイルを編集作成します。そして、それをコンパイルし、DSHEng5 で使用できるようにします。

装置変数情報の定義についての詳しい内容は、
「DSHEng5 装置変数情報定義仕様書」を、

また、編集、コンパイルのために使用する DshGemSet5 プログラムについての機能、操作方法については、
「DSHEng5, DSHEng3 起動ファイル、装置変数情報ファイル設定・編集プログラム説明書」を参照ください。

定義対象になる装置変数情報は以下の情報です。

番号	情報区分	情報名	必須は○印	
1	変数	EC 装置定数	○	
		SV 装置状態変数	○	
		DVVAL 装置データ変数	○	
2	収集イベント	CE 収集イベント	○	
		REPORT レポート	○	
3	アラーム	ALARM アラーム	○	
4	プロセス プログラム (レシピ)	PP プロセスプログラム		PP, FPP, RECIPE の 何れかを選択
5		FPP 書式付プロセスプログラム		
6		RECIPE レシピ		
7	トレース	TRACE 状態変数トレース		
8	スプール	SPOOL メッセージスプール		

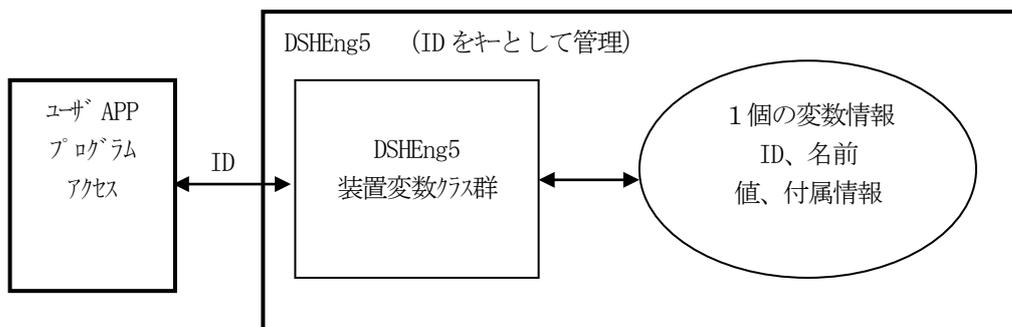
上の表で、番号1, 2, 3の情報の設定は必須です。他の情報については、APP プログラムが変数管理クラスのエンドソッドを使って必要に応じてDSHEng5を通して定義し登録することができます。

ここでは、必須定義情報について簡単に例を挙げて説明します。(詳しくは「装置変数情報定義仕様書」を参照)

装置変数情報の中の個々の情報には、基本的にシステム内部でユニークなIDと名前が与えられます。

(装置側の場合、ホストから与えられる管理情報がありますが、これにはIDがあるけれど名前は与えられないものもあります。)

装置変数情報のアクセスあるいは、直接関連するメッセージの送信においては、IDがキーとして使用されます。



5. 1 装置変数情報の定義と定義ファイルの作成

システム通信仕様書の中には、通常は、先に述べた装置変数情報の定義仕様書が提供されます。仕様書のフォーマットは様々ですが、大体次に示すようなものです。SV、CE、REPORT について簡単な例を使って説明します。

(1) 装置状態変数(SV)

SVID	名前	範囲	規定値	フォーマット	ユニット	内容
1000	SV_Clock			A[20]		装置現時刻 (YYYYMMDDhhmmsscc)
1001	SV_CommunicationState	0~5	0	U1		通信状態
1002	SV_ControlState	1~2	1	U1	state	制御状態
1003	SV_ReadyToLoad	1~2	1	U1		準備できたポートポート
1004	SV_AccessModeStatus	0~3	0	U1		ポートアクセスモード状態
.						
.						

(2) レポート ID

CEID	名前	イベント発生タイミング	リンク VID
102	RP_Communicating	通信が確立した	SV_Clock
			SV_CommunicationState
103	RP_ControlState	コントロール状態が変化した	SV_Clock
			SV_ControlState
104	RP_ReadyToLoad	ローダーが搬入可能状態になった	SV_Clock
			SV_LoadPortId
105	RP_PortAccessMode	ポートのアクセスモードが変更になった	SV_Clock
			SV_AccessModeStatus
.			
.			
.			

(3) 収集イベント(CE)

CEID	名前	イベント発生タイミング	リンクレポート ID
2	CE_Communicating	通信が確立した	RP_Communicating
3	CE_ControlState	コントロール状態が変化した	RP_ControlState
4	CE_ReadyToLoad	ローダーが搬入可能状態になった	RP_ReadyToLoad
5	CE_Port1AccessMode	ポート1のアクセスモードが変更になった	RP_Port1AccessMode
.			
.			
.			

装置変数情報定義ファイルはまず、ソースファイルで定義し、その後、コンパイルして DSHEng5 が解釈できるオブジェクトファイルにコンパイルすることになります。(編集ツールとして、DshGemSet5.exe が準備されています。)

定義情報ファイルの作成、編集そしてコンパイルは **DshGemSet5.exe** プログラムツールを使って行います。

例に出てきた 3 つの情報例に対する定義は、DSHEng5 の装置変数情報定義仕様に従ってそれぞれ以下のようにファイル内に記述します。

① SV 情報の定義

```
def_sv SV_Clock{
    svid:    1000
    format:  A[20]
    nominal: ""
}

def_sv SV_CommunicationState{
    svid:    1001
    format:  U1[1]
    min:     0
    max:     5
    nominal: 0
}

def_sv SV_ControlState{
    svid:    1002
    format:  U2[1]
    units:   "state"
    min:     "0"
    max:     "2"
    nominal: "0"
    event:   "CE_ControlState" // 値変化時に通知するための CEID
}

def_sv SV_LoadPortId{
    svid:    1003
    format:  U1[1]
    units:   ""
    min:     "1" // 数値の場合二重引用符で囲んでも OK
    max:     "2"
    nominal: "1"
}

def_sv SV_AccessModeStatus{
    svid:    1004
    format:  U1[1]
    units:   ""
    min:     "0"
    max:     "3"
    nominal: "0"
}
```

② レポート情報の定義

```

def_report RP_Communicating{
    rptid:    102
    vname:    SV_Clock
    vname:    SV_CommunicationStae
}

def_report RP_ControlState{
    rptid:    103
    vname:    SV_Clock
    vname:    SV_ControlState
}

def_report RP_ReadyToLoad{
    rptid:    104
    vname:    SV_Clock
    vname:    SV_LoadPortId
}

def_report RP_PortAccessMode{
    rptid:    105
    vname:    SV_Clock
    vname:    SV_PortAccessMode
}

```

③ CE 情報の定義

```

def_ce CE_Communicating{
    ceid:    2
    enabled: 1
    rptname: RP_Communicating
}

def_ce CE_ControlState{
    ceid:    3
    enabled: 1
    rptname: RP_ControlState
}

def_ce CE_ReadyToLoad{
    ceid:    4
    enabled: 1
    rptname: RP_ReadyToLoad
}

def_ce CE_PortAccessMode{
    ceid:    5
    enabled: 1
    rptname: RP_PortAccessMode
}

```

5. 2 装置変数情報のコンパイル

5. 1で定義した装置変数情報定義ファイルからDSHEng5が解釈できるオブジェクトファイルにコンパイルしなければなりません。コンパイルの方法としては、次の2つの方法があります。どちらもコンパイラーとしてDshCompile.exeプログラムを使用します。

DshGemSet5.exe、装置変数情報編集プログラムからは、コンパイルボタンのクリックで簡単にコンパイルできます。

DshCompile.exeの起動書式は次のようになります。

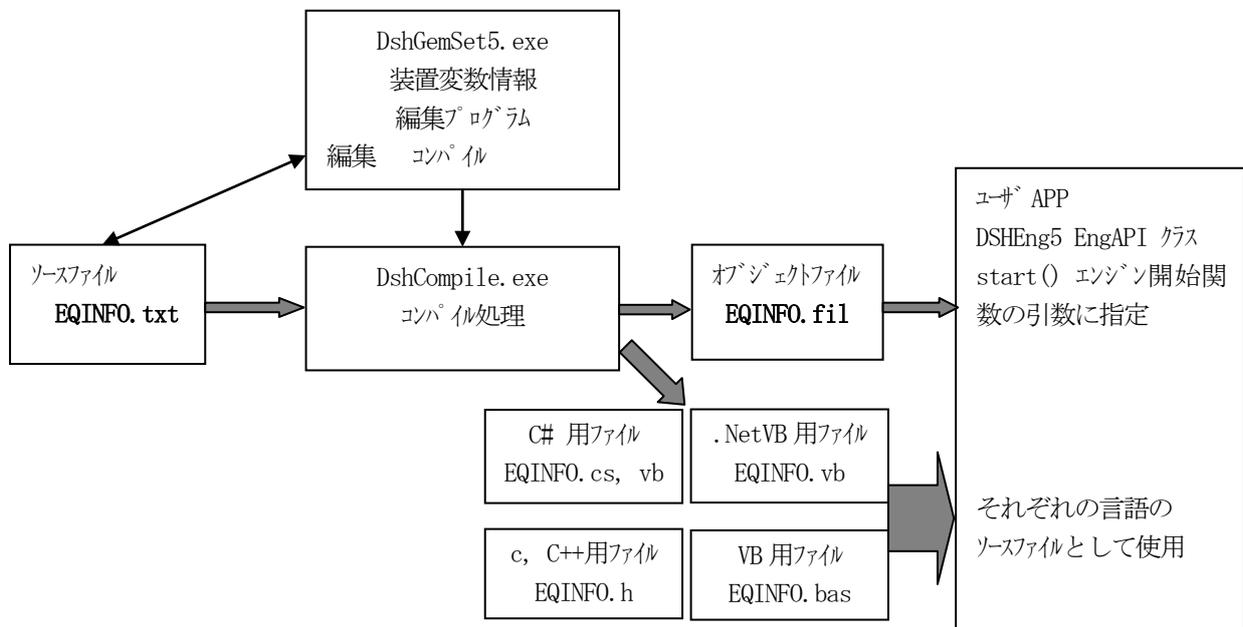
DshCompile.exe <ソースファイル名> <ログファイル名> <エンド通知ファイル名>	
ソースファイル名	: 装置変数情報定義ファイル名(ソースファイル)
ログファイル名	: コンパイルエラーメッセージ、終了時のメッセージなどのログ用ファイル名です。 省略可能です。省略時は dsh_conf.log に記録されます。
エンド通知ファイル名	: DshGemSet5.exe に終了通知するためのファイル名です。 省略してください。

コンパイルの結果、正常に終了した場合は、ソースファイル名の拡張子が .FIL のオブジェクトファイルが生成されます。DSHEng5の装置起動時に、このオブジェクトファイルを装置のセットアップメソッドの引数として与えます。

例えば、EQINFO.txtファイルのコンパイルによって得られるオブジェクトファイルはEQINFO.filになります。

コンパイルの結果、副次的な生成物として、ユーザプログラムのヘダーファイルとして使用することができるファイルが生成されます。C、C++、VB6、VB2008、C#2008 言語用のファイルです。ファイルには、各情報の名前とその値がマクロで定義されています。(次ページ参照)

作業の関連は下図の通りです。



DshGemSet5.exeによる編集でEQINFO.txtファイルを作成し、その後、DshCompile.exeでコンパイルし、EQINFO.filを生成します。そして、それをstart()メソッドで使用します。

5. 3 コンパイルによって生成されたヘッダファイル例

生成されるソースファイルの一部の例を示します。

.cs ファイル(C#2008 用)

```
public const int SV_Clock = 65536 // 0x00010000
public const int SV_ReadyToLoad = 65600 // 0x00010040
public const int CE_ReadyToLoad = 65600 // 0x00010040
public const int RP_ReadyToLoad = 65792 // 0x00010100
```

.vb ファイル(VB2008 用)

```
Public Const SV_Clock As Integer 65536 ' 0x00010000
Public Const SV_ReadyToLoad As Integer 65600 ' 0x00010040
Public Const CE_ReadyToLoad As Integer 65600 ' 0x00010040
Public Const RP_LoadPort As Integer 65792 ' 0x00010100
```

.H ファイル(c, C++用)

```
#define SV_Clock 65536 // 0x00010000
#define SV_ReadyToLoad 65600 // 0x00010040
#define CE_ReadyToLoad 65600 // 0x00010040
#define RP_LoadPort 65792 // 0x00010100
```

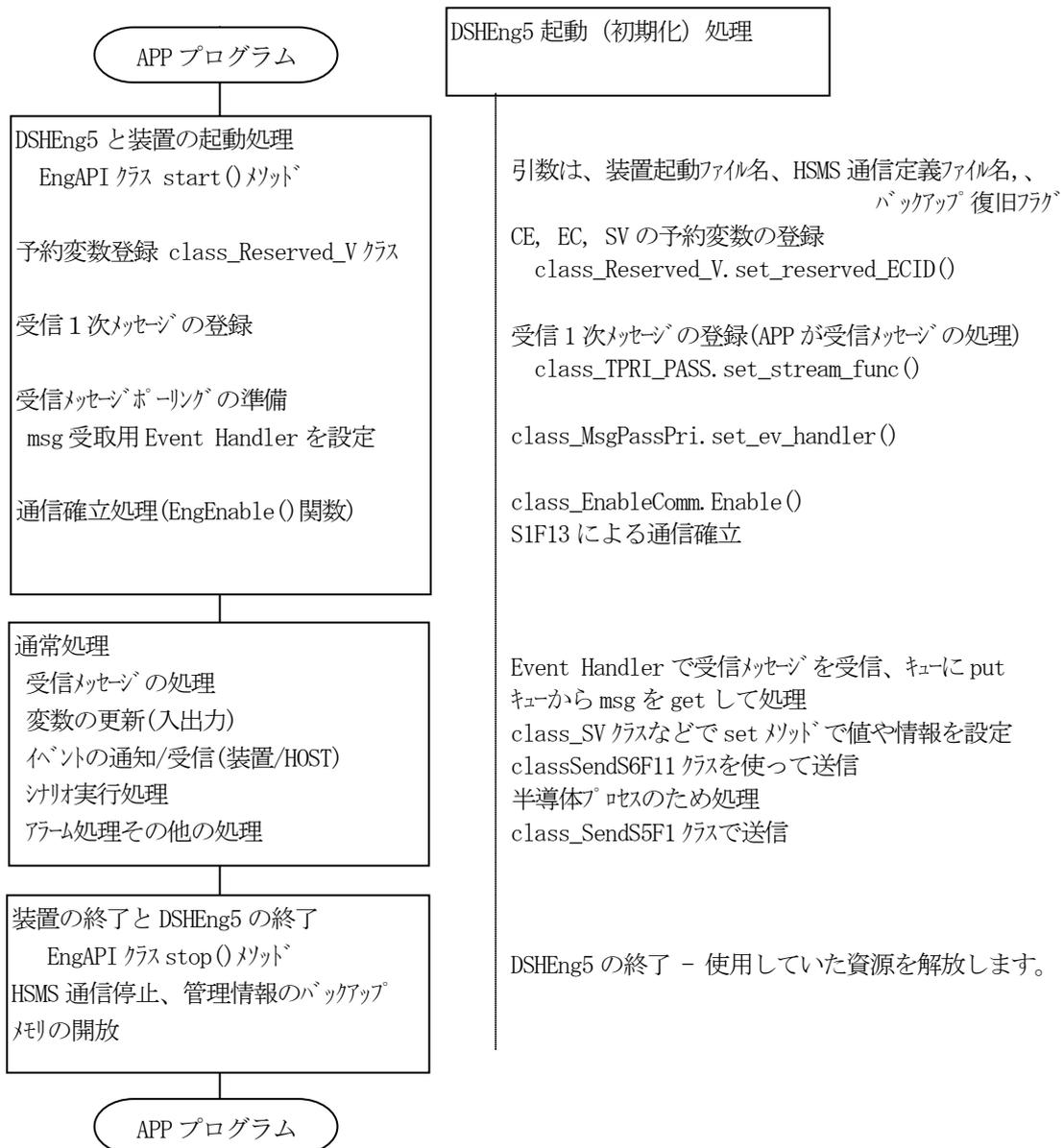
.BAS ファイル(VB6. 0用)

```
Public Const SV_Clock = 65536 ' 0x00010000
Public Const SV_ReadyToLoad = 65600 ' 0x00010040
Public Const CE_ReadyToLoad = 65600 ' 0x00010040
Public Const RP_ReadyToLoad = 65600 ' 0x00010040
```

6. アプリケーションプログラムの作成

ユーザが作成するアプリケーションプログラムが、DSHEng5 関連で行うべき、あるいは行うことができる処理について説明します。(デモプログラムの参照によって、より詳細について理解できます)
大きく3つに分けて説明します。

- (1) DSHEng5 の初期化処理
 - ・ DSHEng5 と装置制御の初期起動処理
- (2) DSHEng5 を使用しての通常処理
 - ・ イベント通知、アラーム通知
 - ・ 1次メッセージの受信処理
 - ・ 装置変数情報の値の更新など
- (3) DSHEng5 の終了処理
 - ・ 装置の終了処理
 - ・ DSHEng5 の終了処理



6. 1 通信エンジン開始処理

最初に、DSHEng5 を起動し、その後、制御する装置を起動します。
起動には EngAPI クラスの start() メソッドを使用します。

6. 1. 1 DSHEng5 通信エンジンの起動（初期化）処理

EngAPI クラスのインスタンスを生成し、start() メソッドを実行します。
メソッドの書式は次の通りです。

```
int start (
    string conf_file,           // 装置起動ファイル名
    string comm_def,           // DSHDR2 ドライバー用通信環境定義ファイル名
    int restore_bkup,          // バックアップ 情報復元指定フラグ
    ref string err_str         // エラー発生時のエラー情報領域ポインタ
);
```

本メソッドで、DSHDR2 HSMS 通信ドライバーで使用する通信環境定義ファイル名、装置起動ファイル名、バックアップ情報を復元するかどうかを指定します。

start() メソッドは、以下の処理を行います。

- (1) 装置制御に必要な制御ブロックのためのメモリを準備します。
conf_file ファイル内に情報の最大管理数が指定されているものについては、指定された管理数分の領域を準備します。(プロセッサプログラム、キャリア、基板、CJ、PJ 情報など)
- (2) conf_file 内に指定されている装置変数情報定義ファイル名から装置管理のための定義情報を読み出し、(1) で準備した情報領域内に登録します。
- (3) restore_bkup フラグが $\neq 0$ 以外の値の場合、バックアップされた装置変数情報を復元します。
先に DSHEng5 が停止した際にバックアップ保存された管理情報を DSHEng5 の管理情報内に読み込みます。
- (3) DSHDR2 通信ドライバーの起動は、comm_def で指定された通信環境情報に基づいて行います。
ここでは、ドライバーを起動し、ポート、デバイスの通信開始も行われます。

6. 1. 2 予約変数の登録

start()メソッドによる装置の起動処理が正常に終わった後、ユーザは、DSHEng5 で予約されている次の装置変数情報の登録を行う必要があります。

予約は、class_Reserved_V クラスのメソッドを使って行います。(メソッドは全て static です)
下で示す定数名は、class_const クラスで定義されています。

予約する必要がある情報として、次の3種類のものがあります。

- 収集イベント (CE)
- 装置定数 (EC)
- 装置状態変数 (SV)

(1) 予約収集イベント ID の登録

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

index 値	定数名	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	スプールデータ送信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

(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	定数名	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	定数名	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

6. 1. 3 ユーザプログラム処理対象受信1次メッセージの登録

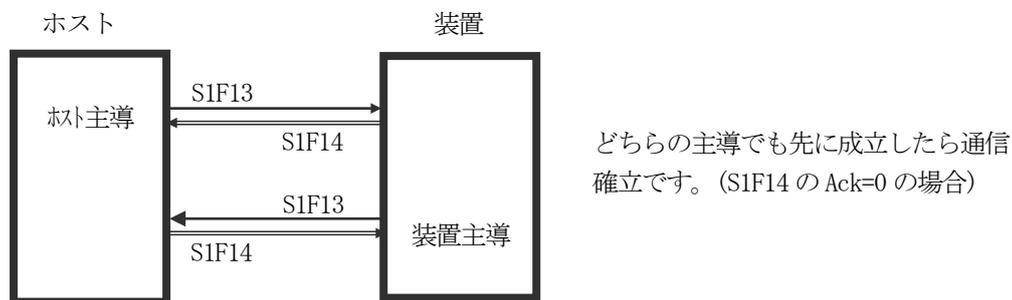
6. で説明したユーザが処理する対象メッセージの登録を class_TPRL_PASS クラスの set_stream_func() メソッドを使って行います。

6. 1. 4 通信確立の処理

ホスト、装置ともに相手装置との通信を行うためには、S1F13, 14 のやり取りによる通信確立のための通信を行う必要があります。

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

本メソッド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、装置との通信状態を DISABLED 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。

6.1.2 で説明した class_Reserved_V クラスの get_reserved_SVID(class_const.SVX_RSV_COMMUNICATING) を使って通信状態値を取得し、通信確立したかどうかを判断します。

```
public const int ST_COMM_DISABLED = 0;
public const int ST_COMM_ENABLED = 1;
public const int ST_NOT_COMMUNICATING = 2;
public const int ST_WAIT_CRA = 3;
public const int ST_WAIT_DELAY = 4;
public const int ST_COMMUNICATING = 5; // 通信確立状態
```

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。

6. 2 通常処理

通常処理については、ユーザがシステムの仕様に基づいてプログラムを設計し、作成しますが、ここでは、シナリオなどを除く一般的な処理について説明します。

6. 2. 1 変数のアクセス

変数には EC (装置定数)、SV (装置状態変数) そして DVVAL (装置データ変数) があります。これら変数は装置の起動処理時に DSHEng5 内に登録されます。

ユーザプログラムは、これら変数値の取得、設定操作を次の API メソッドで行います。

	変数	クラス名	取得メソッド	設定メソッド
1	EC	class_EC	get ()	set ()
2	SV	class_SV	get ()	set ()
3	DVVAL	class_DV	get ()	set ()
4	V (EC, SV, V)	class_V	get ()	set ()

上の表で、4 の class_V のメソッドは、EC, SV, DVVAL の ID がそれぞれユニークであるという条件の中で、EC, SV, DVVAL のどの ID に対してもアクセスできるメソッドです。

これらのメソッドは以下の処理も行います。

(1) 設定値のチェック (最小、最大値による)

変数の設定に関して、もし、その変数に最小値、最大値の指定がある場合、DSHEng5 は設定する際、設定しようとしている値が、この最大値、最小値の範囲内にあるかどうかを調べます。もし、範囲から外れていれば、値を設定しないで、エラーを返却します。

(最小、最大値の設定は、装置変数情報定義ファイルでの指定、または、set_min(), set_max() メソッドによる設定による。

(2) 収集イベント (S6F11) の通知 (装置側)

設定しようとした変数値が元の値と違って、しかも、収集イベント ID (CEID) が指定されていた場合、DSHEng5 は自動的にその指定収集イベントをホストに通知します。

(3) 変数値のバックアップ

DSHEng5 は、変数の値が変更になった場合、情報のバックアップ処理を自動的に行います。

バックアップの処理は、装置起動情報ファイルの INFO_BACKUP コマンドが =1 に指定されていた場合に適用されます。DSHEng5 によるバックアップの対象は、変数だけでなくプロセスプログラム、レシピ、コントロールジョブ、プロセスジョブ、キャリア、基板管理情報も含まれます。

6. 2. 1. 1 変数情報の管理場所とアクセスの方法

各変数は **EngAPI** クラスの下表に示すプロパティに管理されています。

	変数	EngAPU クラス内プロパティ	
1	EC	public static class_EC M_EC	
2	SV	public static class_SV M_SV	
3	DV	public static class_DV M_DV	
4	CE	public static class_CE M_CE	
5	Report	public static class_Report M_RP	
6	Alarm	public static class_AL M_AL	
7	PP	public static class_PP M_PP	
8	FPP	public static class_FPP M_FPP	
9	Recipe	public static class_RCP M_RCP	
10	Carrier	public static class_CAR M_CAR	
11	Substrate	public static class_SUBST M_SUBST	
12	PRJ	public static class_PRJ M_PRJ	
13	CJ	public static class_CJ M_CJ	
14	Trace	public static class_TRACE M_TRACE	
15	Spool	public static class_Spool M_SPOOL	

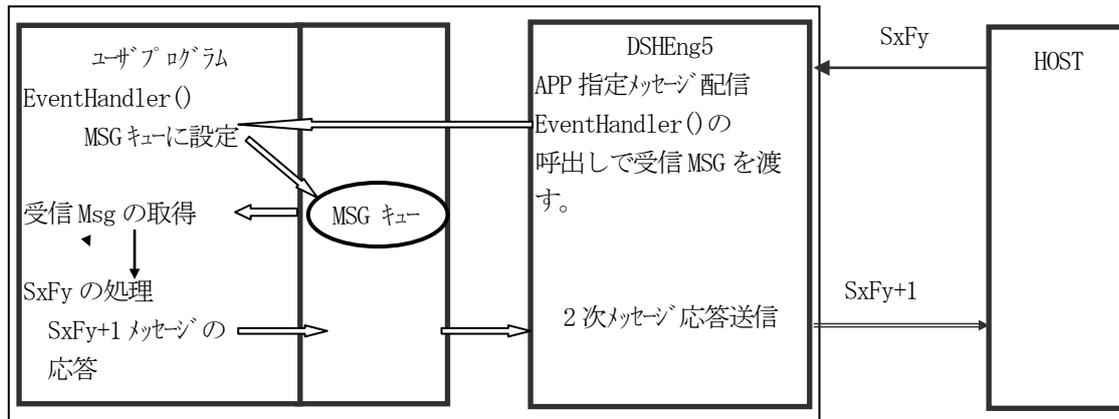
DSHEng5 に管理されている装置変数は、EngAPI.M_xx に対し、そのクラスが提供するメソッドを使って変数情報のアクセスができます。

(例) SV の SV_clock (id=8192, format = A (ascii)) の値を clock_buff に取得するコーディングは以下のようになります。

```
string clock_buff = "";
int result = EngAPI.M_SV.get_value( SV_clock, ref clock_buff );
```

6. 2. 2 受信1次メッセージの処理

6. 1. 3で登録されたユーザ処理対象1次メッセージの処理の流れは次のようになります。



MSG キューは、受信メッセージが一時的に保存される待ち行列で、装置 ID 単位で個別に設けられます。

受信メッセージの取り出しと応答2次メッセージの送信については、既に述べた6. 2を参照ください。受信メッセージの処理の中では、例えば、ホストからのレシピ生成、S15F13を受信した場合を例に説明します。

ユーザプログラムは、受信したS15F13メッセージが有するレポートIDと変数値を取り出し、必要があれば変数の値を更新する必要があります。

DSHEng5は、S15F13のデコードメソッドDshDecodeS15F13()を提供していますので、ユーザはこのメソッドを使って、ユーザが処理しやすい構造体の中にS15F13に含まれる情報を展開することができます。

DshDecodeS15F13メソッドのプロトタイプと使用する構造体は次の通りです。

(1) S15F13のデコードメソッド

```

class S15F13.DecodeS15F13(
    ref DSHMSG msg, // SECSメッセージ情報構造体のポインタ
    TRCP_INFO ref pinfo // デコードした情報を格納するクラスのインスタンス
);
  
```

(2) レシピ情報格納構造体

```

typedef struct { // TRCP_INFO - Recipe Information
    int index;
    int state; // レシピ状態
    string name; // name
    string rcpspec; // rcpid
    int para_count; // # of pparameter
    TRCP_PARA **para_list; // parameter list
    string rcpsbody;
} TRCP_INFO; // Recipe Information
  
```

(注) ppara_count = 0の場合、パラメータがないことを意味します。

(3) レシピ情報の DSHeng5 への登録と設定

装置は、受信したレシピ情報を DSHeng5 内に登録、設定します。

具体的には、EngAllocRcpInfo() メソッドを使って、まず、レシピ ID を登録します。

その後、EngSetRcpInf() メソッドを使って、TRCP_INFO の内容を DSHeng5 のレシピ情報として設定します。

そして、必要な処理を行った後、DshFreeTRCP_INFO() メソッドを使って、TRCP_INFO 内に使用したメモリを開放します。

(4) S15F13 に対する S15F14 の応答

応答情報は、次の構造体内に応答情報を生成し、それを引数として、ULIB 内の DshResponseS15F14() メソッドを使用して S15F14 メッセージを送信します。

```
typedef struct{                                     // TRCP_ERR_INFO – Response Information – S15F14
    int      rmack;                                  // U1
    int      err_count;
    TERR_INFO **err_list;
} TRCP_ERR_INFO;

typedef struct{                                     // TERR_INFO – Response Error Information
    int      errcode;
    string  errtext;
} TERR_INFO;
```

この構造体に情報を設定するための次のメソッドが準備されています。

DshInitTRCP_ERR_INFO()、DshPutTRCP_ERR_INFO()

応答が終了した後、DshFreeTRCP_ERR_INFO() メソッドで、TRCP_ERR_INFO 内で使用したメモリを開放します。

DSHeng5 ではこのようなメッセージの内容をユーザプログラム処理を容易にするためのメッセージデコードメソッドや応答メッセージに設定する情報のためのメソッドが準備されています。詳しくは、「APP インタフェースライブラリメソッド説明書」を参照ください。

6. 2. 3 1次メッセージの送信

1次メッセージの送信は、DSHEng5 が標準でサポートするメッセージについては、それぞれの送信クラスが準備されています。

送信モードとして、非ブロックとブロックの2つのモードがあります。

(1) 非ブロックモード

DSHEng5 に通信メッセージの送信要求を行い、その受付が終了したらコントロールが直ちに戻りそのまま処理を続けることになります。

送信が完了すると、DSHEng5 が送信要求時に与えられたコールバック関数を呼び出して、送信結果、応答情報を APP に渡します。

(2) ブロッカーモード

送信要求を行った後、コントロールはそのままブロックされ、送信完了（応答メッセージの受信）時にコントロールが戻ります。

応答情報は、要求時に渡したクラスのインスタンスなどの中に設定されます。

ユーザプログラムから送信要求される GEM その他の標準的なメッセージについては、DSHEng5 の送信クラスとメソッドとして下表の通りのメソッドが準備されています。

各メッセージ送信クラスについての詳しい説明は、次の説明書を参照ください。

文書番号 DSHENG5-19-30324-00 DSHEng5 GEM 通信エンジン・クラス説明書 Vol-4
SECS-II メッセージ送信クラス

次ページに各メッセージ ID (S, F) 別の送信クラスの一覧表を示します。

[SECS-II メッセージと送信クラス一覧表]

DSHEng5 がサポートするメッセージの一覧表を示します。

	メッセージ	クラス名	機能概略
1	S1F3, 4	class_SendS1F3	S1F3 送信 Selected Equipment Status Request
		-	S1F4 応答
2	S1F11, 12	class_SendS1F11	S1F11 送信 Status Variable Namelist Request
		-	S1F12 応答
3	S1F15, 16	class_SendS1F15	S1F15 送信 Request OFF-LINE
		class_SendS1F16Response	S1F16 応答
4	S1F17, 18	class_SendS1F17	S1F17 送信 Request ON-LINE
		class_SendS1F18Response	S1F18 応答
5	S2F13, 14	class_SendS2F13	S2F13 送信 Equipment Constant Request
		-	(S2F14 はエンジンが自動応答)
6	S2F15, 16	class_SendS2F15	S2F15 送信 New Equipment Constant Send
		-	(S2F16 はエンジンが自動応答)
7	S2F23, 24	class_SendS2F23	S2F23 送信 Trace Initialize Send
		class_SendS2F24Response	S2F24 応答
8	S2F29, 30	class_SendS2F29e	S2F29 送信 Equipmeny Constant Namelist Request
		-	(S2F30 はエンジンが自動応答)
9	S2F31, 32	class_SendS2F31	S2F31 送信 Date and Time Set Requist
		-	(S2F32 はエンジンが自動応答)
10	S2F33, 34	class_SendS2F33	S2F33 送信 Define Report
		-	(S2F34 はエンジンが自動応答)
11	S2F35, 36	class_SendS2F35	S2F35 送信 Link Event Report
		-	(S2F36 はエンジンが自動応答)
12	S2F37, 38	class_SendS2F37	S2F37 送信 Enable/Disabel Event Report
		-	(S2F38 はエンジンが自動応答)
13	S2F41, 42	class_SendS2F41	S2F41 送信 Host Command Send
		class_SendS2F42Response	S2F42 応答
14	S2F43, 44	class_SendS2F43	S2F43 送信 Reset Spooling Stream and Function
		class_SendS2F44Response	S2F44 応答
15	S2F45, 46	class_SendS2F45	S2F45 送信 Define Variable Limit Attributes
		class_SendS2F46Response	S2F46 応答
16	S2F47, 48	class_SendS2F47	S2F47 送信 Variable Limit Attributes Request
		-	(S2F48 はエンジンが自動応答)
17	S2F49, 50	class_SendS2F49	S2F49 送信 Enhanced Remote Command
		class_SendS2F50Response	S2F50 応答
18	S3F17, 18	class_SendS3F17	S3F17 送信 Carrier Action Request
		class_SendS3F18Response	S3F18 応答
19	S3F23, 24	class_SendS3F23	S3F23 送信 Port Group Action Request
		class_SendS3F24Response	S3F24 応答
20	S3F25, 26	class_SendS3F25	S3F25 送信 Port Action Request
		class_SendS3F26Response	S3F26 応答
21	S3F27, 28	class_SendS3F27	S3F27 送信 Change Access
		class_SendS3F28Response	S3F28 応答

22	S5F1, 2	class_SendS5F1	S5F1 送信 Alarm Report Send
		class_SendS5F2Response	S5F2 応答
23	S5F3, 4	class_SendS5F3	S5F3 送信 Enable/Disabel Alarm Send
		-	(S5F4 はエンジンが自動応答)
24	S5F5, 6	class_SendS5F5	S5F5 送信 List Alarm Request
		-	(S5F6 はエンジンが自動応答)
25	S6F1, S6F2	-	(S6F1 はエンジンが自動応答) Trace Data Send
		class_SendS6F2Response	S6F2 応答
26	S6F11, S6F12	class_SendS6F11	S6F11 送信 Event Report Send
		class_SendS6F12Response	S6F12 応答
27	S6F15, S6F16	class_SendS6F15	S6F15 送信 Event Report Request
		-	(S6F16 はエンジンが自動応答)
28	S6F19, S6F20	class_SendS6F19	S6F19 送信 Individual Report Data
		-	(S6F20 はエンジンが自動応答)
29	S6F23, S6F24	class_SendS6F23	S6F23 送信 Request Spooled Data
		-	(S6F24 はエンジンが自動応答)
30	S7F1, S7F2	class_SendS7F1	S7F1 送信 Process Program Load Inquire
		class_SendS7F2Response	S7F2 応答
31	S7F3, S7F4	class_SendS7F3	S7F3 送信 Process Program Send
		class_SendS7F4Response	S7F4 応答
32	S7F5, S7F6	class_SendS7F5	S7F5 送信 Process Program Data
		class_SendS7F6Response	S7F6 応答 (通常エンジン自動応答)
33	S7F17, S7F18	class_SendS7F17	S7F17 送信 Delete Process Program Send
		-	(S7F18 はエンジンが自動応答)
34	S7F19, S7F20	class_SendS7F19	S7F19 送信 Current EPPD Request
		-	(S7F20 はエンジンが自動応答)
35	S7F23, S7F24	class_SendS7F23	S7F23 送信 Formatted Process Program Send
		class_SendS7F24Response	S7F24 応答
36	S7F25, S7F26	class_SendS7F25	S7F25 送信 Formatted Process Program Data
		class_SendS7F26Response	S7F26 応答 (通常エンジン自動応答)
37	S10F1, S10F2	class_SendS10F1	S10F1 送信 Terminal Request
		class_SendS10F2Response	S10F2 応答
38	S10F3, S10F4	class_SendS10F3	S10F3 送信 Treminal Display, Single
		class_SendS10F4Response	S10F4 応答
39	S10F5, S10F6	class_SendS10F5	S10F5 送信 Terminal Display, Multi-Block
		class_SendS10F6Response	S10F6 応答
40	S14F9, S14F10	class_SendS14F9	S14F9 送信 Create Object Request
		class_SendS14F10Response	S14F10 応答
41	S14F11, S14F12	class_SendS14F11	S14F11 送信 Delete Object Request
		class_SendS14F12Response	S14F12 応答
42	S15F3, S15F4	class_SendS15F3	S15F3 送信 Recipe Name Space Action Request
		class_SendS15F4Response	S15F4 応答
43	S15F5, S15F6	class_SendS15F5	S15F5 送信 Recipe Name space Rename Request
		class_SendS15F6Response	S15F6 応答
44	S15F7, S15F8	class_SendS15F7	S15F7 送信 Recipe Space Request
		-	(S15F8 はエンジンが自動応答)

45	S15F9, S15F10	class_SendS15F9	S15F9 送信 Recipe Status Request
		-	(S15F10 はエンジンが自動応答)
46	S15F13, S15F14	class_SendS15F13	S15F13 送信 Recipe Create Request
		class_SendS15F14Response	S15F14 応答
47	S15F17, S15F18	class_SendS15F17	S15F17 送信 Recipe Retrieve Request
		class_SendS15F18Response	S15F18 応答
48	S16F5, S16F6	class_SendS16F5	S16F5 送信 Process Job Command Request
		class_SendS16F6Response	S16F6 応答
49	S16F11, S16F12	class_SendS16F11	S16F11 送信 PrJobCreateEnh
		class_SendS16F12Response	S16F12 応答
50	S16F15, S16F16	class_SendS16F15	S16F15 送信 PrJobMultiCreate
		class_SendS16F16Response	S16F16 応答
51	S16F17, S16F18	class_SendS16F17	S16F17 送信 PrJobDeque
		class_SendS16F18Response	S16F18 応答
52	S16F19, S16F20	class_SendS16F19	S16F19 送信 PrGetAllJobs
		class_SendS16F20Response	S16F20 応答
53	S16F21, S16F22	class_SendS16F21	S16F21 送信 PrGetSpace
		class_SendS16F22Response	S16F22 応答
54	S16F27, S16F28	class_SendS16F27	S16F27 送信 Control Job Command Request
		class_SendS16F28Response	S16F28 応答

6. 2. 4 収集イベント送信処理について

装置側では、装置状態変数値の変化の要因によってホストに対し収集イベント通知メッセージ、S6F11を送信する必要があります。

装置側から収集イベントを通知する手段として2つの方法があります。

- DSHEng5 が装置状態変数の変化に伴って自動的に収集イベントを通知します。
- ユーザプログラムが class_SendS6F11 の SendS6F11() または、SendS6F11_wait() メソッドを使ってイベントを通知します。

(1) DSHEng5 による自動イベント通知

ユーザプログラムが装置状態変数の値を変更することによって、予めその変数値の変化によって通知する CEID (収集イベント ID) が指定されていた場合、DSHEng5 は、指定されている CEID の定義内容に従って S6F11 メッセージを組み立てホストに対し自動的に送信します。

自動通知の仕組みは次の通りです。

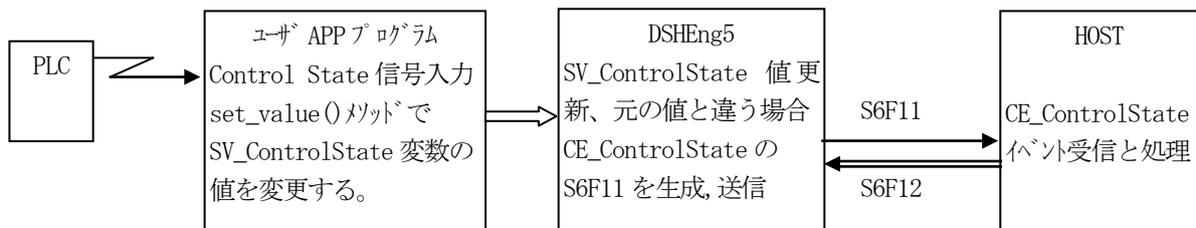
ここでは、仮に状態変数を **SV_ControlState**、通知イベント ID を **CE_ControlState** として説明します。

① 装置変数情報定義ファイル上に予め、状態変数の変化に対する収集イベント ID を指定しておきます。
SV_ControlState の値が変化した際に自動的に CE_ControlState の収集イベントを通知するように指定しておきます。(定義については、5. 1 (3) - ①の定義例を参照)

② ユーザが SV_ControlState の値を次のメソッドで変更します。

```
set_value(SV_ControlState, new_control_state); // new_control_state= 新変数値
```

この SV 値設定メソッドによって DSHEng5 は、SV_ControlState の SVID を持つ装置状態変数値の更新を行います。そして、新しい設定値が元の値と違っていた場合、DSHEng5 は、自動的に CE_ControlState のイベント通知を行います。



(2) ユーザプログラムの class_SendS6F11.SendS6F11() メソッドによるイベント通知

ControlState 信号を入力した後、set_value() メソッドで値を更新し、ユーザプログラムの判断で、class_SendS6F11.SendS6F11() メソッドを使って収集イベントをホストに通知します。収集イベントに関係する引数としては CEID (収集イベント ID) だけ指定すればよく、メッセージの組み立ては、DSHEng5 が全て行ってくれます。

class_SendS6F11 クラスの次のメソッド

① 非ブロックモード

```
public int SendS6F11( uint ceid, class_CALLBACK.callback_S6F12 callback, UInt32 upara)
```

② ブロックモード

```
public int SendS6F11_wait(uint ceid, ref int ack6)
```

6. 2. 5 アラーム送信処理について

装置側では、人間あるいは装置の安全に関わる危険な状態が発生し、それを検出した際、また、危険な状態が解除された際、ホストに対しアラーム通知メッセージ、 S5F1 を送信する必要があります。

class_SendS5F1 クラスの SendS5F1() または SendS5F1_wait() メソッドを使って送信します。

```
class_SendS5F1 クラスの次のメソッド  
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)
```

引数の中に、on_off がありますが、発生／復旧を 1 / 0 で指定します。

装置変数情報の中のアラーム情報の定義において、ce_on, ce_off コマンドで、そのアラームの発生／復旧時に通知する収集イベント ID が指定されていれば、DSHEng5 は、アラームと同時に収集イベントの通知(S6F11 の送信)も自動的に行います。

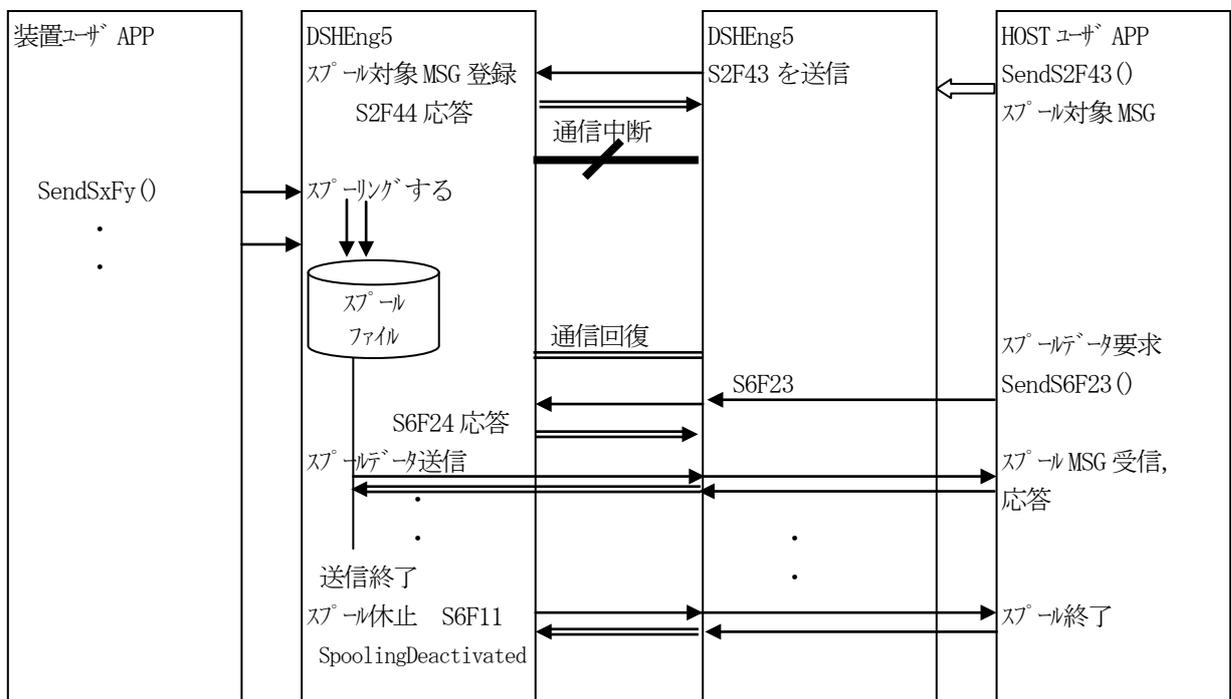
6. 2. 6 スプール機能処理について

スプール処理は、装置側での処理であり、通信が中断した後、装置側からホストへ送信しようとしたメッセージの中、スプール対象に指定されたメッセージをディスクメモリに一時的に保存し、通信が復旧した際に、ホストの指示に従ってスプールしたメッセージをホストに送信するものです。

ホストからのスプール対象メッセージの指定は、S2F43 メッセージで与えられます。DSHEng5 は受信した S2F43 メッセージの内容に従ってスプールの準備をします。(ユーザプログラムは関知する必要がありません。)

DSHEng5 では、スプール対象メッセージを装置変数情報定義ファイルの中で指定することができます。そして、ユーザによってスプール対象メッセージの登録あるいは取り消しを行うこともできます。

スプール処理の流れは次のようになります。



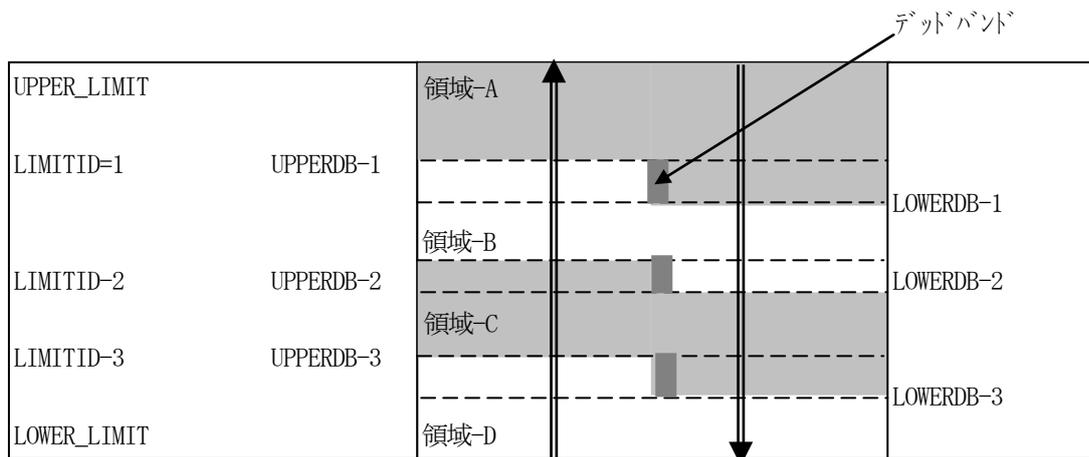
このチャートで判りますように、装置側でのほとんどのスプール処理は、DSHEng5 が自動的に行ってくれます。

6. 2. 8 リミット監視とホスト通知処理について

監視対象の変数の値が、ある領域から別の領域へ遷移する際に、領域の境界に UPPERDB と LOWERDB で与えられたデッドバンドを抜け出したときにイベントを通知し、ホストに伝える機能の変数リミット監視機能です。

変数値の領域は、最大8つに分割できます。これにより設けることができるデッドバンドは最大7つになります。

下の図は、4つの領域に分割されたケースです。(3つの LIMITID が存在します。)



例えば、領域-A について説明すると、UPPERDB-1 と LOWERDB-1 で挟まれた部分が領域-A と他の領域間を遷移する際のデッドバンドになります。そして、次のような変数値の変化による遷移でイベントを発生させます。

- ①変数値が 領域-A 以外の領域から UPPERDB-1 の値以上になり、領域-A に遷移したときにイベントを発生させます。
- ②変数値が領域-A から LOWERDB-1 の値以下になり、他の領域に遷移したときにイベントを発生させます。

UPPER_LIMIT、LOWER_LIMIT は、それぞれ、その変数に与えられた MAX、MIN の値と同じです。

領域-B、C、D についても領域-A と同様になります。

また、DSHEng5 では、リミット監視イベントをホストに通知するだけでなく、DSHEng5 のアプリケーションプログラムに対しても通知する機能を提供します。

リミット監視機能については、

文章番号 DSHEng5-19-30310-00 「変数リミット監視機能 説明書」

を参照してください。

6. 3 通信エンジン終了処理

6. 1. 1において start()メソッドで起動した装置の処理と DSHEng5 とを終了させるための処理です。

start()時に生成した EngAPI クラスのインスタンスで、stop()メソッドを実行します。

```
public void stop();
```

本メソッドは、装置処理のために使用したメモリ、管理情報などをシステムに返却するとともに装置処理のために生成されていたスレッドの終了も行います。

DSHEng5 エンジン処理のために使用した管理領域のためのメモリの開放、処理のために生成されていたスレッドの終了などを行います。それから DSHDR2 通信ドライバーの停止も行います。

<付録>

付録-A DSHEng5 通信制御エンジンライブラリ関連文書一覧表

付録-A1 DSHEng5 通信制御エンジンライブラリ関連文書一覧表

#	文書番号	文書名	注釈
1	DSHEng5-19-30300-00	DSHEng5 通信制御エンジンライブラリ (SECS/HSMS) 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 装置変数情報定義仕様書 (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	装置変数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	DSHEng5-19-30305-00	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、装置変数情報ファイル設定・編集プログラム説明書	DSHGEM-LIB, DSHEng4 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	DSHEng5-19-30340-00	ユーザ作成ライブラリメソッド 2次メッセージ応答メソッド一覧表	C, C++言語によるプログラミング .Net 用クラスライブラリを使用しない
10	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・
11	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 構造で表示します・

付録-A2 DSHeng5 GEM 通信エンジン・クラス説明書

Vol 番号	文書番号	内容
Vol-1	DSHENG5-19-30321-00	DSHeng5 GEM 通信エンジン説明書 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)
Vol-2	DSHENG5-19-30322-00	変数情報関連クラス (EC, SV, DVVAL, CE, Report, Alarm)
Vol-3	DSHENG5-19-30323-00	プロセス情報関連クラス (PP, FPP, RECIPE, PRJ, CJ, CARRIER, SUBSTRATE)
Vol-4	DSHENG5-19-30324-00	SECS-II メッセージ送信クラス
Vol-5	DSHENG5-19-30325-00	SECS-II 通信メッセージ情報保存クラス
Vol-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

付録-A3 DSHeng5 エンジン・デモ・プログラム関連文書一覧表

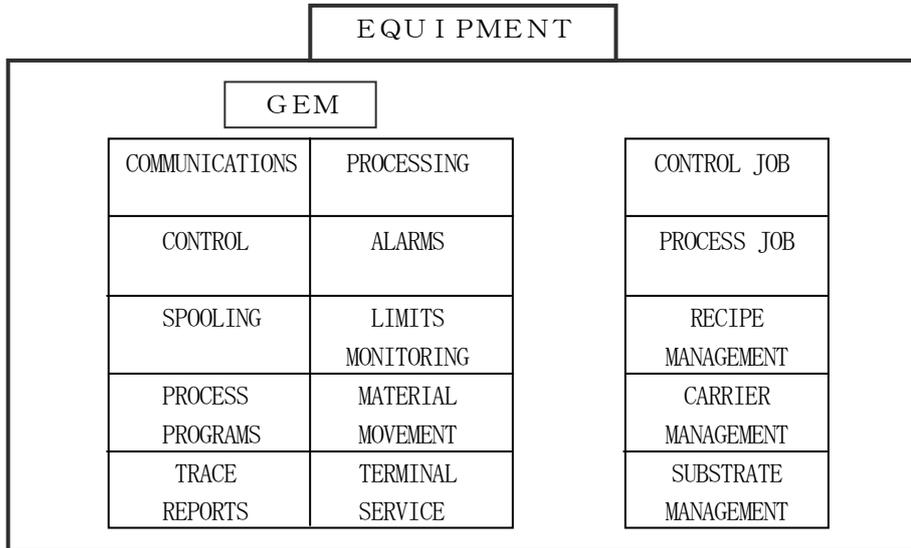
#	文書番号	文書名	注釈
1	DSHeng5-19-30501-00	クラス・ライブラリ・デモプログラム説明書	(本ドキュメントです。)
2	DSHeng5-19-30502-00	DSHeng5 デモプログラム インストールと保存ファイル	C#, .Net VB デモプログラムです。

付録-B SEMI スタンドへの対応

通信プロトコルは、SECS-I ならびに HSMS-SS をサポートします。

SECS-II メッセージのエンコード、デコードと装置変数情報（変数など）の管理を行います。

SEMI スタンドが提唱する GEM ならびに CJ, PJ, RMS, CMS、STS に関するスタンドをサポートします。



GEM スタンドに含まれない製造工場特有なメッセージの送受信も可能なように DSHEng5 はユーザが SECS-II メッセージの送受信をするための API メソッドも準備されています。

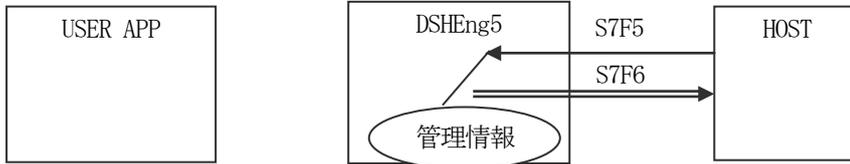
付録-C 装置変数情報とバックアップファイル

- (1) 装置固有管理情報（変数、収集イベント、アラームなど）の定義ファイルによる登録
変数、収集イベント情報などについて、名前と ID、それに付随するパラメータ（フォーマット、値、リンク情報など）の情報をテキストファイル上に DSHEng5 が定める書式で定義します。変数別に作成したものの内容をシステムに登録します。
専用の管理情報編集ソフトウェアツール、DshGemSet5.exe プログラムを使って編集可能です。
- (2) 装置変数情報のアクセス機能ならびにバックアップ機能
（1）で定義し、DSHEng5 内に登録された装置変数情報に対するアクセス機能が準備されています。ID をキーにした値の取得、更新設定などのアクセスを DSHEng5 が提供するクラスのメソッドを使って自由に行うことができます。
また、DSHEng5 は、装置変数情報をファイルに随時バックアップします。このバックアップファイルは終了後、次の開始時に復元し、前回の終了時の情報を継続して使用することを可能にします。

付録-D SECS-II の通信制御処理

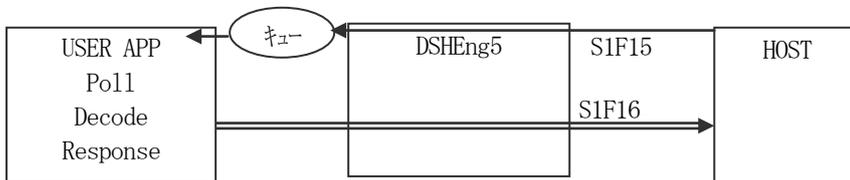
DSHEng5 は、装置間の SECS-II メッセージ通信処理について、ユーザができる限りシンプルに通信プログラミングできるように仕組みと手段を提供します。

- ① DSHEng5 が自身で対応できる受信 1 次メッセージに対しユーザの手を煩わすことなく自動的に処理します。

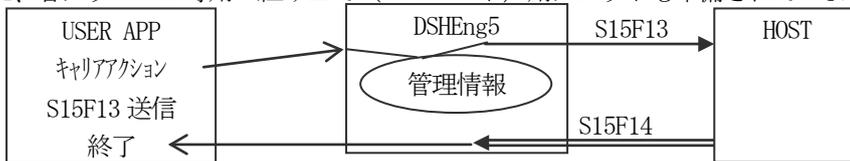


- ② APP が処理したい受信メッセージの場合、DSHEng5 が受信キューを通して APP に渡します。

DSHEng5 は受信キューをポーリングするためのメソッドを提供します。また、メッセージ内の情報を、プログラムが処理しやすいように、メッセージ専用の構造体にデコードするためのメソッドも提供します。これによってユーザは 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メソッドならびに応答するためのメソッドを提供します。



- 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には、専用 API メソッド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。また、各メッセージ専用の組み立て (エンコード) 用メソッドも準備されています。



メッセージの送信については、通信のトランザクションの完了を待機する方法として、プログラムの制御をそのままブロックして待つ方法と、プログラムの制御を完了を待たずに進め、完了を指定したコールバックメソッドを呼び出させる方法の 2 つの方法があります。

