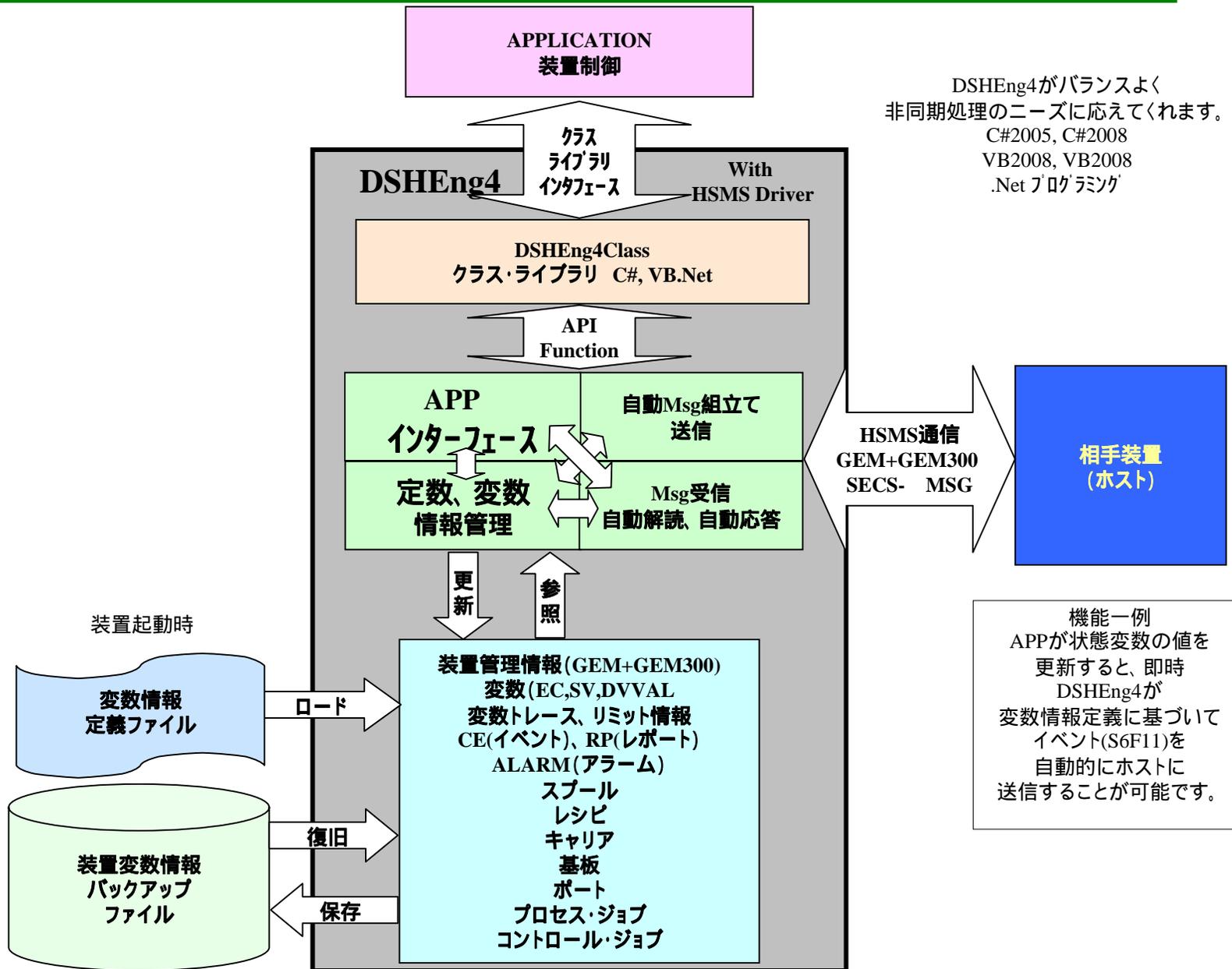


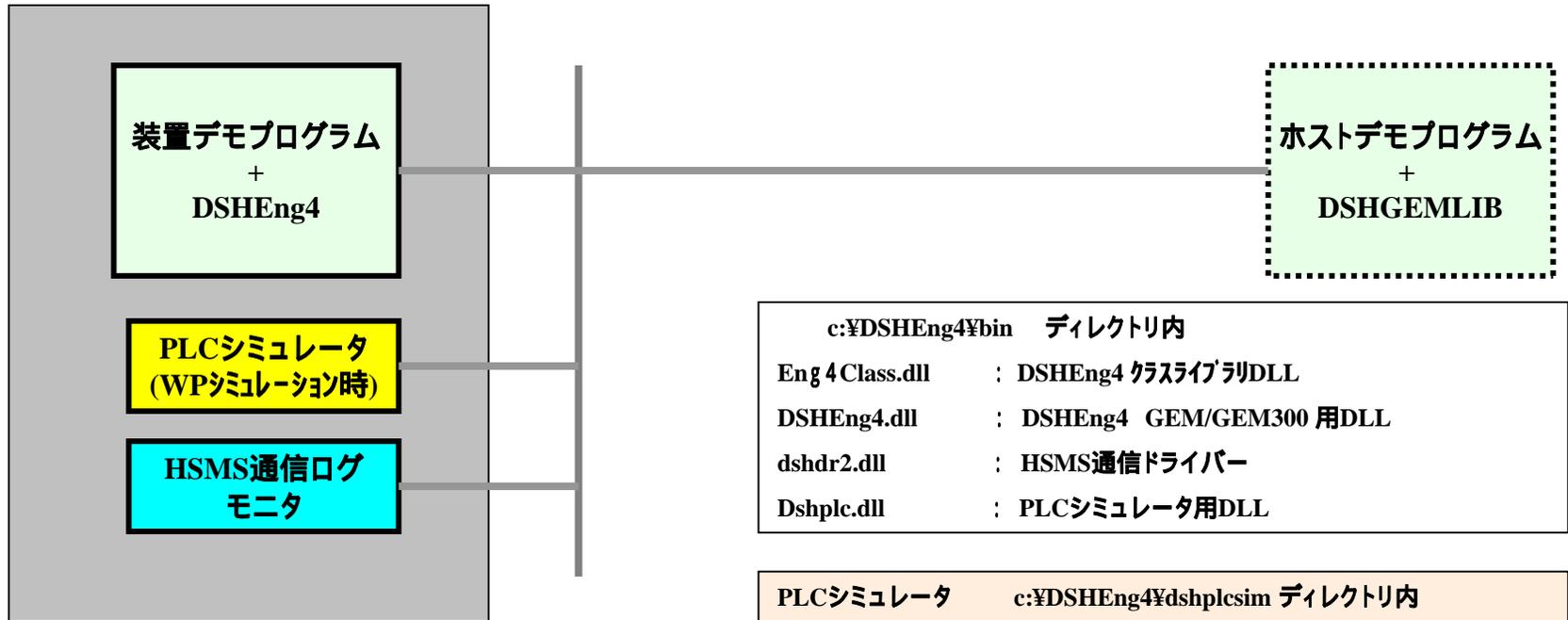
# DSHEng4クラス・ライブラリ使用の装置 デモプログラムの紹介 (2010年6月)

きめ細くGEM機能の確認ができます。

文書番号 DSHENG4-09-30391-00



# 装置制御デモプログラム with PLC Simulator



c:\¥DSHEng4¥bin ディレクトリ内	
Eng 4 Class.dll	: DSHEng4 クラスライブラリDLL
DSHEng4.dll	: DSHEng4 GEM/GEM300 用DLL
dshdr2.dll	: HSMS通信ドライバー
Dshplc.dll	: PLCシミュレータ用DLL

PLCシミュレータ	c:\¥DSHEng4¥dshplcsim ディレクトリ内
dshplcsim.exe	: Plc Simulator
Sample.pls	: I/O信号定義ファイル
Seq_300.prg	: シーケンスファイル

HSMS通信ログモニタ	c:\¥DSHEng4¥logmon ディレクトリ内
Logmon	: ログモニター
logon.dll	: ログモニター用DLL

装置デモ	c:\¥DSHEng4¥bin ディレクトリ内
EngCsDemo.exe	: C#2008言語
EngVbDemo.exe	: VB2008言語
EngCsDemo2005.exe	: C#2005言語
Equip.cnf	: 装置 Config File (起動ファイル)
EQINFO.fil	: 装置変数定義ファイル(EC,SV,DVVAL, CE etc)
Comm.def	: DSHDR2 HSMS通信環境定義ファイル

comm.def : DSHDR2 HSMS通信ドライバー 通信環境定義ファイル  
(¥DshEng4¥cnf)内  
Port-1, Device-1 を使用します。  
変更項目 : PORT-1 の IP の値を 相手(HOST)のコンピュータのIPにあわせてください。

次頁に画面例を紹介します。

# 装置デモ・WPシミュレーションの画面例

**formLog**

```

***** DSHeng4Class Library Demo Program Start *****
***** DshEng4 Engine Calss Library Start *****
** eng_class.start() ei = 0
* setup_eq_start()
* setup_eq_start() end
  transfer status =2
  carid status =0
  slotmap status =0
  association status =0
  access status =0
        
```

Log On/Off On 10

**DSHENG4-EngCalss - C#2008デモプログラム**

エンジン  
 バックアップ 復旧 なし

通信コントロール  
   通信状態 未確立

イベント通知  
 イベント通知(S6F11) ceid CE\_Communicating  
 周期(ms) 400 0

アラーム通知  
 アラーム通知(S5F1) alid AL\_AlarmTempOver  
復旧/発生 発生(1)

端末メッセージ  
 シミュレーション Simulation  スプール情報

変数情報

イベント・アラーム情報

プロセスプログラム・レシピ情報  
   
 S6F11表示指定 詳細(V)

キャリア・基板情報  
   
 SxFy Ack設定

コントロールジョブ・プロセスジョブ情報

ALL

**WP 自動シミュレーション**

画面左下の WP処理開始 ボタンのクリックで始めます。

処理状態 IDLE

半導体製造装置

Work

ポート-1     ポート-2

CarrierID CARID\_01 Usage   
 RCPID  Location   
 Pr JobID  CJobID

コントロール ONLINE, LOCAL

アクセスモード AUTO 処理済み回数   
  サイクル時間

アラーム通知 AL\_AlarmTempOver 発生  
アラーム

端末メッセージ This is a test message.

PLC Simulator  
 PLC I/O定義ファイル sample.pls  
 PLC TCP port 5930

PLOSim使用選択 使用しない

# 装置デモ・PLCシミュレータ画面例

DSH-PLcSim-2008-1 Trial S/N : 1214875261-T - effective 60 days after 2009-6-30 PLC Side 株式会社 データマップ

ファイル(F) 編集(E) 表示(O) 通信設定(S) ウィンドウ(W) ログファイル(L) ヘルプ(H)

I/O定義ファイル名 参照 D:\DshPlcSim\sample1 I/O操作開始 I/O操作終了 I/O定義ファイル参照 I/O定義ファイル終了 ログ表示 Log On

**接続入力On/Off マップ**

ON	OFF	選択	X_Port1RdyToLoad	AllOn	AllOff	同期
0000						
0010						
0020						
0030						
0040						
0050						
0060						
0070						
0080						
0090						
00A0						
00B0						
00C0						
00D0						
00E0						
00F0						
0100						
0110						
0120						
0130						
0140						
0150						
0160						
0170						
0180						
0190						
01A0						
01B0						
01C0						
01D0						
01E0						
01F0						

**接続出力On/Off マップ**

ON	OFF	選択	Y_PlcStart	AllOn	AllOff	同期
0000						
0010						
0020						
0030						
0040						
0050						
0060						
0070						
0080						
0090						
00A0						
00B0						
00C0						
00D0						
00E0						
00F0						
0100						
0110						
0120						
0130						
0140						
0150						
0160						
0170						
0180						
0190						
01A0						
01B0						
01C0						
01D0						
01E0						
01F0						

**接続入出力実行シーケンスプログラムファイル - C:\DSSHPLcSim\Yseq\_300.pr**

ステップ	制御対象名	On/Off	条件になる名	On/Off	注
60	X_Port1Foupclose	On			
61	Y_Port1Foupclose	Off			
62	X_Port1Foupclose	Off			
63	X_Port1MovedCarWork	Off			
64	X_Port1MovedCarDst	Off			
65	\$WAIT		Y_Port1RdyUnloadLamp	On	
66	Y_Port1Clamp	Off			
67	X_Port1Clamp	Off			
68					
69	X_Port1StartUnload	On			
70	Y_Port1RdyUnloadLamp	Off			
71					
72	\$DELAY	1			
73					
74	X_Port1UnloadEnd	On			
75					
76	\$DELAY	2			
77					
78	\$WAIT		Y_Port1Complete	On	
79	\$WAIT		Y_Port1Complete	On	

**PLC Operation Log**

```

#OUT - addr: 15 -> On ( Y_Port1Foupclose )
#IN - addr: 16 -> On ( X_Port1Foupclose )
#OUT - addr: 15 -> Off ( Y_Port1Foupclose )
#IN - addr: 14 -> Off ( X_Port1Foupclose )
#IN - addr: 18 -> Off ( X_Port1MovedCarWork )
#IN - addr: 17 -> Off ( X_Port1MovedCarDst )
#OUT - addr: 30 -> On ( Y_Port1RdyUnloadLamp )
#OUT - addr: 19 -> Off ( Y_Port1Clamp )
#IN - addr: 19 -> Off ( X_Port1Clamp )
#IN - addr: 31 -> On ( X_Port1StartUnload )
#OUT - addr: 30 -> Off ( Y_Port1RdyUnloadLamp )
#IN - addr: 32 -> On ( X_Port1UnloadEnd )
#OUT - addr: 32 -> On ( Y_Port1Complete )
#OUT - addr: 10 -> On ( Y_Port1RdyLoadLamp )
#OUT - addr: 20 -> On ( Y_Port2RdyLoadLamp )
    
```

**Carrier ID**

Carrier ID	Slotmap
CARD_01	slot-01
CARD_01	slot-02
CARD_02	slot-03
CARD_03	slot-04
CARD_04	slot-05
CARD_05	slot-06
CARD_06	slot-07
CARD_06	slot-08
CARD_07	slot-09
CARD_07	slot-10
CARD_08	slot-11
CARD_08	slot-12
CARD_09	slot-13
CARD_09	slot-14
CARD_10	slot-15
CARD_10	slot-16
CARD_10	slot-17
CARD_10	slot-18
CARD_10	slot-19
CARD_10	slot-20
CARD_10	slot-21
CARD_10	slot-22
CARD_10	slot-23
CARD_10	slot-24
CARD_10	slot-25

**ProcessData**

ProcessData	Slot
1111	slot-10
2222	slot-19
3333	slot-20
4444	slot-21

110 Y\_Port2RdyLoadLamp

通信は接続状態です。

## 装置デモ・ログモニター画面例

```
>
07/02 18:35:18 PT-01 Recv S6F12 len=0013 dvid=1234 blk=0000 sybt=00002fa1
<B[1]=x00>
07/02 18:35:18 PT-01 Send S6F11 len=0098 dvid=1234 W blk=0000 sybt=00002fa2
<L 3
<U4[4]=12037>
<U4[4]=15799>
<L 1
<L 2
<U4[4]=15856>
<L 7
<A[16]="2009070218351890">
<A[7]="LOT_000">
<L 1
<A[0]="">
<A[9]="LocationA">
<U1[1]=1>
<U1[1]=1>
<A[7]="PRODUCT">
>
>
>
>
07/02 18:35:19 PT-01 Recv S6F12 len=0013 dvid=1234 blk=0000 sybt=00002fa2
<B[1]=x00>
07/02 18:35:19 PT-01 Send S6F11 len=0067 dvid=1234 W blk=0000 sybt=00002fa3
<L 3
<U4[4]=12038>
<U4[4]=15800>
<L 1
<L 2
<U4[4]=15857>
<L 3
<A[16]="2009070218351976">
<A[7]="LOT_000">
<L 1
<A[0]="">
>
>
>
>
07/02 18:35:20 PT-01 Recv S6F12 len=0013 dvid=1234 blk=0000 sybt=00002fa3
<B[1]=x00>
07/02 18:35:20 PT-01 Send S6F11 len=0082 dvid=1234 W blk=0000 sybt=00002fa4
<L 3
<U4[4]=12039>
<U4[4]=15797>
<L 1
<L 2
<U4[4]=15792>
<L 8
```

画面消去	開始	停止	終了	
接続=ON	192.168.1.16	9999	CODE=OFF LINK=OFF	598949
LOGMON.log				598800

装置 - ホスト間のHSMS通信トランザクションをリアルタイムでモニタリングします。

装置、ホスト以外のPCからもモニタリング可能です。

(DSHDR2 HSMS通信ドライバーと接続しています。)

## GEM 装置変数定義の内容 EC, SV, DVVAL, RP, CE, AL

一部分を紹介します。(テキストファイルです)

```
//---- EC - Equipment Constant -----
def_ec EC_Mdln{
    ecid: 1           // =0x00000001
    format: A[6..6]
    nominal: "DSH100"
}

def_ec EC_SoftRev{
    ecid: 2           // =0x00000002
    format: A[6..6]
    nominal: "REV001"
}

def_ec EC_InitCommState{
    ecid: 4           // =0x00000004
    format: U2[1]
    min: "0"
    max: "1"
    nominal: "0"
}

def_ec EC_InitControlState{
    ecid: 5           // =0x00000005
    format: U2[1]
    min: "0"
    max: "1"
    nominal: "0"
}
```

```
//---- SV - Equipment Status Variable -----
def_sv SV_Clock{
    svid: 8192        // =0x00002000
    format: A[16..16]
    nominal: ""
}

def_sv SV_CommunicationState{
    svid: 8193        // =0x00002001
    format: U1[1]
    nominal: "0"
}

def_sv SV_ControlState{
    svid: 8194        // =0x00002002
    format: U2[1]
    units: "state"
    min: "0"
    max: "2"
    nominal: "0"
    event: "CE_ControlState"
}

def_sv SV_CJExecName1{
    svid: 8195        // =0x00002003
    format: A[16..16]
    units: "CJOB"
    nominal: ""
}
```

```
//---- DV - Data Variable -----
def_dv DV_StartTime{
    dvid: 8300        // =0x0000206c
    format: A[19..19]
    nominal: ""
}

def_dv DV_EndTime{
    dvid: 8301        //
    =0x0000206d
    format: A[19..19]
    nominal: ""
}

def_dv DV_Temp1{
    dvid: 8302        // =0x0000206e
    format: A[8..8]
    units: "Degree"
    nominal: ""
}
```

詳細はEQINFO.TXTファイルを  
参照してください。

次頁にRP,CEを紹介します。

## RP(Report), CE(CollectionEvent)

一部分を紹介します。

```
//---- RP - Collection Event -----  
-----  
def_report RP_Communicating{  
    rptid: 10          // =0x0000000a  
    vname: "SV_Clock"      // vid=8192  
}  
  
def_report RP_ControlState{  
    rptid: 100         // =0x00000064  
    vname: "SV_Clock"      // vid=8192  
    vname: "SV_ControlState" // vid=8194  
}  
  
def_report RP_PortAccessMode{  
    rptid: 1200        // =0x000004b0  
    vname: "EC_PortAccessMode" // vid=256  
}  
  
def_report RP_Port1AccessMode{  
    rptid: 1201        // =0x000004b1  
    vname: "EC_Port1AccessMode" // vid=257  
}  
  
def_report RP_Port2AccessMode{  
    rptid: 1202        // =0x000004b2  
    vname: "EC_Port2AccessMode" // vid=258  
}
```

```
//---- CE - Collection Event -----  
def_ce CE_Communicating{  
    ceid: 2           // =0x00000002  
    enabled: 1  
    rptname: "RP_Communicating" // rptid=10  
}  
  
def_ce CE_SpoolDeactivated{  
    ceid: 999         // =0x000003e7  
    enabled: 1  
}  
  
def_ce CE_ControlState{  
    ceid: 100         // =0x00000064  
    enabled: 1  
    rptname: "RP_ControlState" // rptid=100  
}  
  
def_ce CE_AlarmOn{  
    ceid: 200         // =0x000000c8  
    enabled: 1  
}  
  
def_ce CE_AlarmOff{  
    ceid: 201         // =0x000000c9  
    enabled: 1  
}
```

### DSHEng4は

これらの情報を内部に取り込み、RP、CE、変数間のリンク関係を生成し、管理します。

変数の値の設定・参照のためのサービスをAPI関数によって提供します。

CEイベントは NotifyEvent() API関数を使ってCEIDを指定するだけでS6F11を送信してくれます。

(注) DSHGEMLIBと共通

次頁にAlarmを紹介します。

## AL(Alarm)

一部分を紹介します。

```
//----- AL - Alarm-----  
def_alarm AL_AlarmTempOver{  
  alid: 1           // =0x00000001  
  enabled: 1  
  alcd: "2"  
  altx: "Chamber Temperature Over"  
  ce_on: "CE_AlarmOn"      // ceid=200  
  ce_off: "CE_AlarmOff"    // ceid=201  
}  
  
def_alarm AL_AlarmPressure_1_Low{  
  alid: 101         // =0x00000065  
  enabled: 1  
  alcd: "2"  
  altx: "Chamber Pressure-1 Over"  
  ce_on: "CE_AlarmOn"      // ceid=200  
  ce_off: "CE_AlarmOff"    // ceid=201  
}  
  
def_alarm AL_AlarmPressure_2_Low{  
  alid: 102         // =0x00000066  
  enabled: 1  
  alcd: "2"  
  altx: "Chamber Pressure-2 Over"  
  ce_on: "CE_AlarmOn"      // ceid=200  
  ce_off: "CE_AlarmOff"    // ceid=201  
}
```

**DSHEng4は**

ここで定義されるアラーム情報  
に従ってNotifyAlarm()関数で  
ALIDと発生/復帰の指定だけで  
S5F1を送信可能にします。

他に以下の情報を変数定義ファイル内に  
定義できます。  
SPOOL  
TRACE  
RECIPE

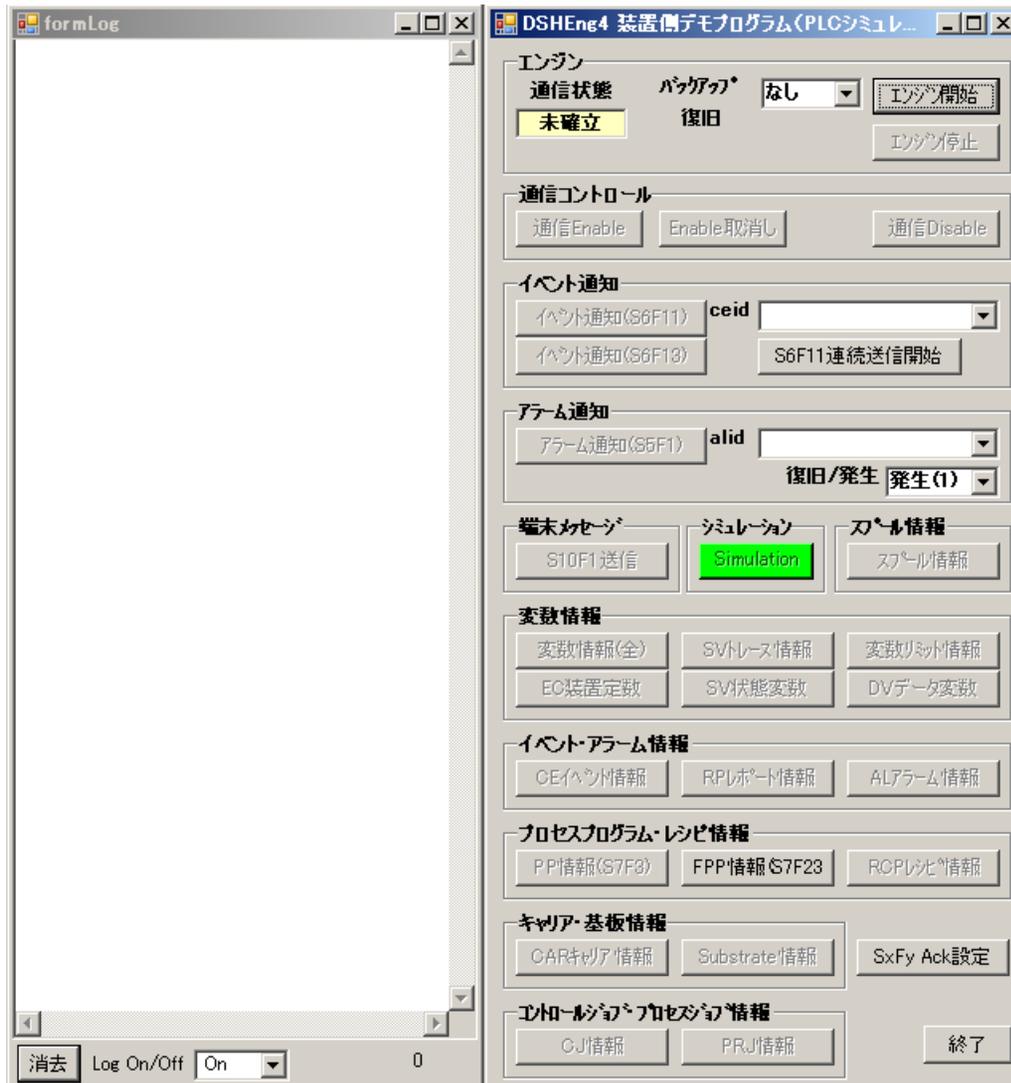
**EQINFO.txtファイルは**

Dshedit.exe  
編集プログラムを使って  
画面で編集できます。

詳細はEQINFO.TXTファイルを  
参照してください。

デモプログラムの起動は次頁です。

## デモプログラムの起動とメイン画面



### デモプログラムの開始

VB : %DSHEng4%\bin\%EngCsDemo.exe

または

C# : %DSHEng4%\bin\%EngVbDemo.exe

を起動します。

左の画面が表示されます。

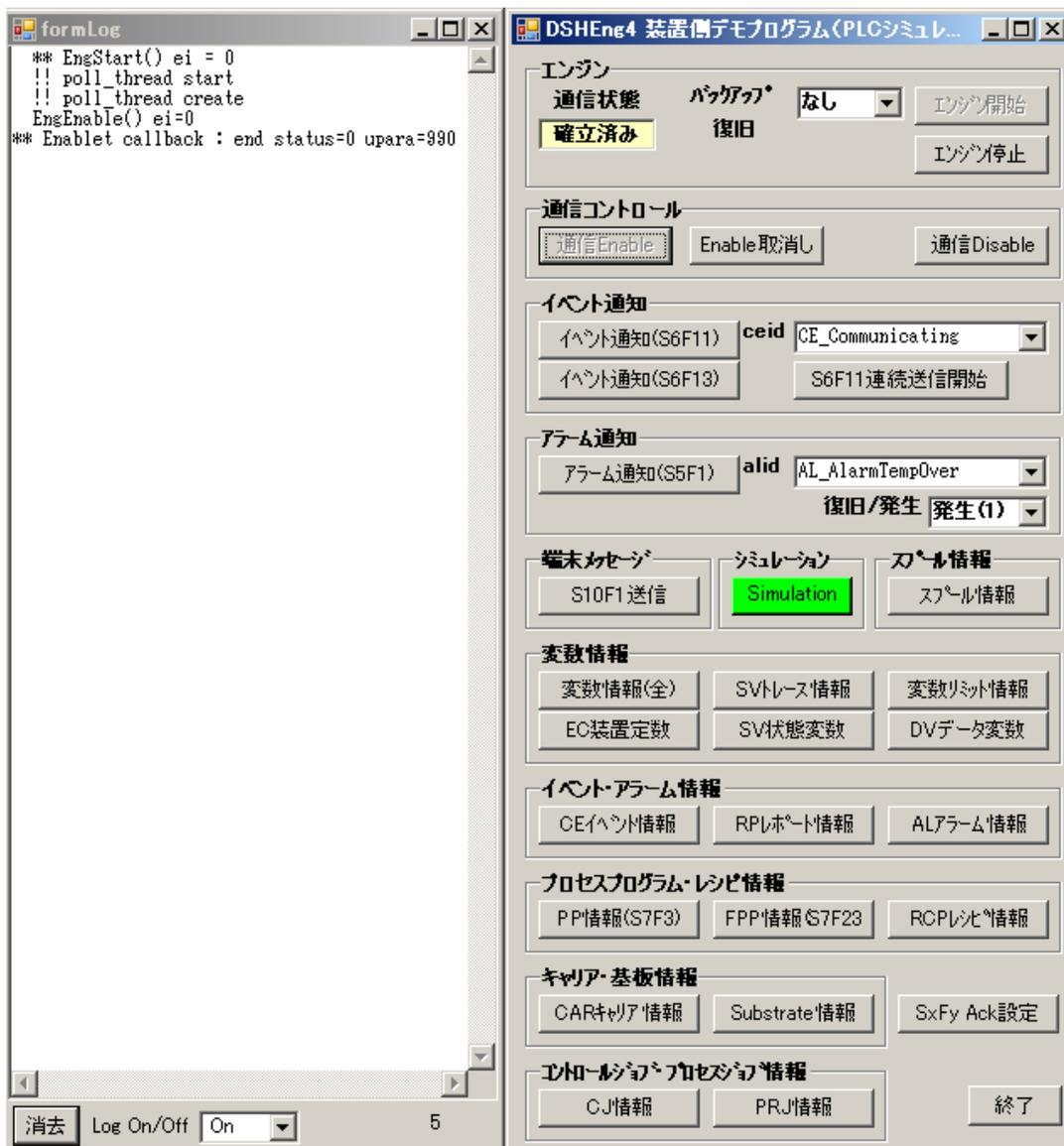
通信を開始するために次の順にボタンをクリックします。

エンジン開始 ボタン、

通信Enable ボタン

次頁のように装置通信が可能になります。

## 通信確立後のメイン画面



通信Enable

のクリック後、相手ホストとの間で通信確立ができたなら、通信状態が“確立済み”になります。(S1F13,14のやり取りの成功)

その後、以下の操作ができます。

CE(S6F11)収集イベントの送信  
ceidを選択し、イベント通知(S6F11)をクリックす

ると、選択されたceidのレポートをホストに送信します。  
( DSHEng4がリンクされている変数の値を付けて送信します。)

Alarmの送信  
alidと復旧/発生を選択し、アラーム通知(S5F1)ボタンをクリックします。

通信Disable は通信確立を解消します。

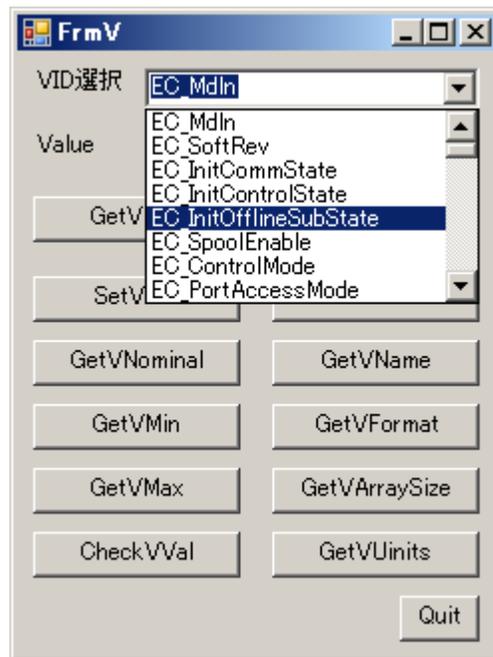
装置停止 は装置制御を停止します。

エンジン停止 はDSHEng4を終了します。

他のボタンをクリックすると、それに対応する機能の操作画面が表示されます。

EC, SV, DVVALの設定・参照 など  
TRACE, LIMIT, CE, REPORT, ALARM,  
PP, RECIPE, CARRIER, SUBSTRATE,  
CJ, PRJ などの情報の生成、設定、参照などの操作画面です。

## 変数(V)情報操作画面



VID選択用コンボボックスにシステムに登録されている変数のID(EC,SV,DVVAL)が表示されます。

### ID単位の操作

VIDを選択し、操作したいボタンをクリックします。

SetVVal

値を設定する場合は、Value欄に値を設定した後、このボタンをクリックします。

GetVVal

などの変数情報取得用のボタンクリックの場合は、取得結果はメイン画面のログ部に表示されます。

EC, SV, DVVALの操作画面については、それぞれの変数の操作になるだけで、基本的に本画面の操作と同じ操作になりますので、説明を省略します。

### メイン画面のログ表示

```
EngGetVVal() ei = 0
    value = DSH100
EngSetVVal() ei = 0
    fmt=10 asize=6
EngGetVVal() ei = 0
    value = MODELA
```

DSHEng4はAPI関数を介して変数IDをキーにして全変数の情報のアクセスが可能です。

値の設定、変更、情報(単位、データ範囲など)の参照が可能です。

## SVトレース(Trace)設定画面

The screenshot shows the 'frmTrace' application window. It has a title bar with standard Windows window controls. The main area is divided into several sections:

- Trace情報**: A section containing a 'TRID選択' dropdown menu with 'TRACE1' selected. Below it are four input fields: 'DSPER' (10), 'DSPER\_TIME' (10), 'TOTSMP' (9), and 'REPGSZ' (3). At the bottom of this section is an 'SV-COUNT' input field with the value '1'.
- SVID選択**: A dropdown menu with 'SV\_Clock' selected. Below it are two buttons: 'SVIDをリストに追加' and 'SVIDリストのリセット'.
- SVID リスト(設定用)**: A large empty rectangular area intended for a list of SVIDs.
- index**: A small input field containing the number '0'.
- Buttons**: A grid of buttons including 'AllocTrInfo', 'SetTrInfo', 'GetTrInfo', 'DelTrInfo', 'EnableTrace', and 'Quit'. There are also 'SetTrInfoX', 'GetTrInfoX', and 'DelTrInfoX' buttons.

The screenshot shows a dialog box titled 'Allocate Trace ID'. It has a text input field containing 'TRACE1'. Below the input field are two buttons: 'OK' and 'Cancel'. At the bottom of the dialog, there is a message: 'AllocateするTRIDを入力してください。'

### SV\_Clockのtrace msg例

```
06/20 16:17:23 PT-01 Send S6F1 len=0098
d=1234 W blk=0000 sybt=0000008f
<L 4
<A[6]="TRACE1">
<U2[2]=2>
<A[16]="2009063016272349">
<L 3
<A[16]="2009063016272349">
<A[16]="2009063016272349">
<A[16]="2009063016272349">
>
>
```

本画面は、装置側でも変数トレース機能を試すことができるように設けられています。

以下のように操作します。

1. **AllocTrInfo** ボタンをクリックし、TraceIDを割当てます。(左の画面)
2. **DSPER, TOSMP, REPGSZ, SV\_COUNT** を設定します。  
SV\_COUNTは同時にトレース対象とするSVの数です。
3. 次にトレース対象にするSVIDを**SVID選択**コンボボックスで決めます。  
そして、**SVIDのリストに追加** をクリックし、下のリストボックスに表示します。  
これをSV\_COUNT分だけ行います。
4. **SetTrInfo** をクリックし、システムに登録します。(TRID選択のTRIDに対し)  
これで設定OKです。
5. **EnableTrace** をクリックすると設定した条件でトレースさせ、結果をホストにS6F1で送信することができます。

ホスト側のデモプログラムで結果を確認できます。

(本来はホストからS2F23で設定される機能です)

## 変数リミット(Limit)情報設定画面

LimitID	Lowerdb	Upperdb
0	20	40
1	41	60
2	61	80

### メイン画面のログ表示 (SetVLimit)

```
----- Limit Information -----  
VID      = 8303  
limit count = 3  
format   = 2A  
asize    = 1  
i = 0  
  limit_id = 0  
  upperdb  = 40  
  lowerdb  = 20  
i = 1  
  limit_id = 1  
  upperdb  = 60  
  lowerdb  = 41  
i = 2  
  limit_id = 2  
  upperdb  = 80  
  lowerdb  = 61  
EngSetVLimit() ei = 0  
vid = DV_Temp1_bin
```

本画面は、装置側でも変数リミット設定機能を試すことができるように設けられています。

以下のように操作します。

1. VID選択を変数のコンボボックスで行います。例えば、DV\_Temp1\_binを選択します。
2. ID Countに設定する変数リミットID数を入力します。  
本画面では最大3個になります。
3. LimitID, Lowerdb, Upperdb値をキー入力  
で設定します。(ID count分だけ)
4. **SetVLimit** をクリックし、リミット情報を  
選択された変数に登録します。  
これで設定OKです。
5. 設定値の確認は **GetVLimit** のクリック  
でできます。
6. リミット情報の消去は **DelVLimit** の  
クリックで行います。

(本来はホストからS2F45で設定される機能です。  
値のバンド幅間の変移によってイベントレポート  
が装置から報告される機能です。)

## 収集イベント(CE)情報操作画面



### メイン画面のログ表示 (GetCeAllRpid)

```
EngGetCeAllRpid() id = 15600(0x3CF0) ei = 1  
rpcount = 1  
0 - rpid = 15600(0x3CF0) RP_ReadyToLoad
```

本画面は、CE収集イベントの情報の確認と有効かどうかの設定を行うためのものです。

それぞれのボタンによる操作について説明します。

1. **GetCeList** はシステムに登録されている全CEIDとCE名リストを表示します。
2. **GetCeName** はCEID選択で選択されているCEIDの名前を取得し表示します。
3. **SetCeEnabled** は選択されているCEIDを有効にします。
4. **GetCeEnabled** は選択されているCEIDが有効かどうかを確認します。
5. **GetCeRpid** は、右側の順位で指定されている位置にリンクされているレポートIDを取得します。
6. **GetCeAllRpid** は、選択されているCEIDにリンクされているレポートIDを全て取得し、表示します。

## レポート(RP)情報操作画面



メイン画面のログ表示  
(GetRpAllVid)

```
EngGetRpAllVid() id = 15792(0x3DB0) ei = 8
vicount = 8
0 - vid = 8192(0x2000) SV_Clock
1 - vid = 8199(0x2007) SV_LoadPortTransferStatus
2 - vid = 8200(0x2008) SV_LoadPortId
3 - vid = 8201(0x2009) SV_LoadCarId
4 - vid = 8202(0x200A) SV_LoadCarIdStatus
5 - vid = 8203(0x200B) SV_LoadSlotMapStatus
6 - vid = 8204(0x200C) SV_LoadPortAssociationStatus
7 - vid = 8205(0x200D) SV_LoadCarAccessStatus
```

本画面は、レポートの情報の確認を行うためのものです。

それぞれのボタンによる操作について説明します。

1. **GetRpList** はシステムに登録されている全RPIDとレポート名リストを表示します。
2. **GetRpName** はRPID選択で選択されているRPIDの名前を取得し表示します。
3. **GetRpVid** は、右側の順位で指定されている位置にリンクされている変数IDを取得します。
4. **GetRpAllVid** は、選択されているRPIDにリンクされている変数IDを全て取得し、表示します。

## アラーム(AL)情報操作画面



### メイン画面のログ表示 (GetAlcd, GetAltx)

```
EngGetAlcd() id = 1(0x1)
    alcd = 2
EngGetAltx() ei=0 id=1 (0x1)
    altx = Chamber Temperature Over
```

本画面は、アラーム情報の確認と有効かどうかの設定を行うためのものです。

それぞれのボタン操作について説明します。

1. **GetAlList** はシステムに登録されている全ALIDとAL名リストを表示します。
2. **GetAlName** はALID選択で選択されているALIDの名前を取得し表示します。
3. **SetAlEnabled** は選択されているALIDを有効にします。
4. **GetAlEnabled** は選択されているALIDが有効かどうかを確認します。
5. **GetAlcd** は選択されているALIDのALCDを取得します。
6. **GetAltx** は、選択されているALIDのALTXを取得します。
7. **GetAlCeOn** は、選択されているALIDのアラームが発生した時にレポートされるCEIDを取得します。
8. **GetAlCeOff** は、選択されているALIDのアラームが復旧した時にレポートされるCEIDを取得します。

## レシビ(RCP)情報操作画面

### メイン画面のログ表示

#### (GetRcpInfo)

```
EngGetRcpInfo() ei = 0
rcpspec = RCP100
para count = 2
i-1
rcpparm = rcppara1
rcpparval(bin) = 54321
i-2
rcpparm = rcppara2
rcpparval(bin) = 12345
rcpbody = rcpbody-Sample
```

(注) ボタンの末尾の X は、IDに対するインデクスをキーにして情報を参照するものです。インデクス値はIDがシステムに登録された時点(Allocateされた)にIDに対して発行されます。

11. **SendS15F17** は、レシビ検索要求 S15F17を送信します。
12. **GetRcpList** は、システムに登録されている全レシビのIDをリスト表示します。
13. **GetRcpId** **GetRcpIndex** は、RCPIDからインデクス、インデクスからRCPIDを取得するためのボタンです。

本画面は、レシビ情報の設定と確認を行うためのものです。

それぞれのボタン操作について説明します。

1. **AllocRcpInfo** は新規にレシビを設けるためのボタンです。IDを左の画面で指定してそれをシステム内に生成します。
2. **SetRcpInfo** はRCPID選択で選択されているRCPIDに対し、レシビパラメータ上に入力されているパラメータを設定します。
3. **GetRcpInfo** は選択されているRCPIDの情報を取得し表示します。
4. **DelRcpInfo** は選択されているRCPIDをシステムから削除します。
5. **SetRcpState** **GetRcpState** は選択されているRCPの状態の設定、取得用です。
6. **SendS15F3** は、レシビアクション要求 S15F3を送信します。
7. **SendS15F5** は、レシビのRename要求 S15F5を送信します。
8. **SendS15F7** は、レシビスペース要求 S15F7を送信します。
9. **SendS15F9** は、レシビステータス要求 S15F9を送信します。
10. **SendS15F13** は、レシビの生成要求 S15F13を送信します。選択されているレシビの情報を送ります。

## コントロールジョブ(CJ)情報操作画面

frmCj

CjID選択 --- index 0

AllocCjInfo GetCjList

SetCjInfo SetCjInfoX

GetCjInfo GetCjInfoX

DelCjInfo DelCjInfoX

GetCjId GetCjIndex

SetState SetStateX

GetState GetStateX

Quit

設定情報

objspec equipment

objtype ControlJob

carid CARID03

prjid PRJ0001

d collection plan D1234

pause ceid 200

order mgmt 1

start(0/1) 1

State Queued

(注) ボタンの末尾の X は、IDに対するインデクスをキーにして情報を参照するものです。インデクス値はIDがシステムに登録された時点(Allocateされた)にIDに対して発行されます。

Allocate CJID

CJID-2

OK Cancel

AllocateするCJIDを入力してください。

### メイン画面のログ表示 (GetCjInfo)

```
EngGetCjInfo() ei = 0
cjid = CJID-1
objspec = equipment
objtype = ControlJob
objid = CJID-1
attr_count= 12
i-1
ObjID
objid = CJID-1
i-2
CarrierInputSpec
count = 1
j-1
car in spec = CARID03
i-3
CurrentPRJob
j-1
current pr job = PRJ0001
```

本画面は、コントロールジョブ(CJ)情報の設定と確認を行うためのものです。

それぞれのボタン操作について説明します。

1. **AllocCjInfo** は新規にコントロールジョブを設けるためのボタンです。IDを左の画面で指定してそれをシステム内に生成します。
2. **SetCjInfo** はCjID選択で選択されているCJに対して設定情報上に入力されている属性情報を設定します。
3. **GetCjInfo** は選択されているCJIDの情報を取得し表示します。
4. **DelCjInfo** は選択されているCJIDをシステムから削除します。
5. **SetState** **GetState** は選択されているCJの状態の設定、取得用です。
6. **GetCjList** は、システムに登録されている全コントロールジョブのIDをリスト表示します。
7. **GetCjId** **GetCjIndex** は、CJIDからインデクス、インデクスからCJIDを取得するためのボタンです。

# プロセスジョブ(PRJ)情報操作画面

## メイン画面のログ表示 (GetPrjInfo)

```
EngGetPrjInfo() ei = 0
prjid = PRJID-1
carcount = 1
i-1
car_id = CARID03
slot_count= 25
j-1
slotid = 10
j-2
slotid = 11
j-3
slotid = 12
j-4
slotid = 13
j-5
slotid = 14
j-6
```

(注) ボタンの末尾の X は、IDに対するインデクスをキーにして情報を参照するものです。インデクス値はIDがシステムに登録された時点(Allocateされた)にIDに対して発行されます。

本画面は、プロセスジョブ(PRJ)情報の設定と確認を行うためのものです。

それぞれのボタン操作について説明します。

1. **AllocPrjInfo** は新規にプロセスジョブを設けるためのボタンです。IDを左の画面で指定してそれをシステム内に生成します。
2. **SetPrjInfo** はPrjID選択で選択されているPRJに対し、設定情報上に入力されている情報を設定します。
3. **GetPrjInfo** は選択されているPRJIDの情報を取得し表示します。
4. **DelPrjInfo** は選択されているPRJIDをシステムから削除します。
5. **SetState** **GetState** は選択されているPRJの状態の設定、取得用です。
6. **GetPrjList** は、システムに登録されている全プロセスジョブのIDをリスト表示します。
7. **GetPrjId** **GetPrjIndex** は、PRJIDからインデクス、インデクスからPRJIDを取得するためのボタンです。

# キャリア(CAR)情報操作画面

The main window 'frmCar' contains the following elements:

- CarID選択: CARID03 (dropdown), index: 0
- IdStatus: 0, MapStatus: 0, AccStatus: 0
- Location: SOURCE, Usage: PRODUCT
- SlotCount: 25, Mid: K1234, Substid: K1234
- Buttons: AllocCarInfo, GetCarList, SetCarInfo, GetCarInfo, SetCarInfoX, GetCarInfoX, DelCarInfo, DelCarInfoX, SetIdStatus, GetIdStatus, SetIdStatusX, GetIdStatusX, SetMapStatus, GetMapStatus, SetMapStatusX, GetMapStatusX, SetLocation, GetLocation, SetLocationX, GetLocationX, SetUsage, GetUsage, SetAccStatus, GetAccStatus, SetUsageX, GetUsageX, SetAccStatusX, GetAccStatusX, SetSlotMap, GetSlotMap, SetContent, GetContent, SetSlotMapX, GetSlotMapX, SetContentX, GetContentX, slot pos(1,2...25) with value 1, and a Quit button.
- A table with columns 'slot id' and values 1 through 25.

The 'Allocate CARID' dialog box shows: CARID-2, OK, Cancel, and the text 'AllocateするCARIDを入力してください。'

本画面は、キャリア(CAR)情報の設定と確認を行うためのものです。

それぞれのボタン操作は基本的にCJなどの情報操作画面のものと同様に行ないます。ここでは、各ボタンの機能だけを簡単に説明します。

## メイン画面のログ表示 (GetCarInfo)

```
EngGetCarInfo() ei = 0
carid = PRJID-1
----- Car Info -----
carid = PRJID-1
usage = PRODUCT
location = SOURCE
map_status = 0
id_status = 0
acc_status = 0
slot_count = 25

1 slot_id = 1 status = 0
mid = K1234
substid = K12341
2 slot_id = 2 status = 0
```

(注) ボタンの末尾の X は、IDに対するインデクスをキーにして情報を参照するものです。インデクス値はIDがシステムに登録された時点(Allocateされた)にIDに対して発行されます。

1. **AllocCarInfo** 新規キャリアの登録
2. **SetCarInfo** **GetCarInfo** 情報設定・取得  
設定は画面上部の入力情報を設定します。
3. **DelCarInfo** キャリアID削除
4. **SetSlotMap** **GetSlotMap** Slot分の情報設定と取得
5. **SetContent** **GetContent** 各スロットに対するLotID、SubstID(基板)の設定・取得
6. その他のボタンは種々の状態、Location、Usageなど情報の設定・取得用です。
7. **GetCarList** システムに登録されているキャリアIDのリスト表示

## 基板(Subst)情報操作画面

The main window 'frmSubst' contains the following elements:

- SubstID 選択**: Dropdown menu with 'index' 0.
- 設定情報**: Fields for AcquiredID (SUBST100), LotID (LOT-111), SubstLocID (LOCID-001), Source (LOC\_SRC), Destination (LOC\_DEST), BatchLocID (BLOC-01), PosInBatch (POS-001), and various status dropdowns (State, IDStatus, Material, ProcState, LocState, Type, Usage).
- Loc History**: Table with columns locID, TimeIn, TimeOut. Current entry: locID: LOCID-1, TimeIn: 200907011701, TimeOut: 200907011701.
- Buttons**: A grid of buttons for actions like AllocSubstInfo, GetSubstList, SetPosInBat, etc.
- Allocate SUBSTID**: A dialog box with 'SUBSTID-2' entered and 'OK'/'Cancel' buttons.

本画面は、基板 (SUBST)情報の設定と確認を行うためのものです。

それぞれのボタン操作は基本的にCJなどの情報操作画面のものと同様に行ないます。ここでは、各ボタンの機能だけを簡単に説明します。

1. **AllocSubstInfo** 新規基板の登録
2. **SetSubstInfo** **GetSubstInfo** 情報設定・取得設定は画面上部入力情報を設定します。
3. **DelSubstInfo** キャリアID削除
4. その他のボタンは種々の状態、Location、Usageなど情報の設定・取得用です。
5. 右上のLocHistoryのブロックでは、基板の現在位置と入、出時刻を設定、追加、参照するためのボタンです。
6. **GetSubstList** システムに登録されているキャリアIDのリスト表示

### メイン画面のログ表示 (GetSubstInfo)

```
EngGetSubstInfo() ei = 0
  substid = SUBSTID-2
----- SUBST Information -----
  substid      = SUBSTID-2
  acquiredID   = SUBST100
  lotID        = LOT-111
  locID        = LOCID-001
  source       = LOC_SRC
  destination  = LOC_DEST
  batch locID  = POS-001
  pos in batch = 0
  status       = 0
  Id_status    = 0
  material status = 0
  proc state   = 0
  loc state    = 0
  type        = 0
  usage       = 0
```

(注) ボタンの末尾の X は、IDに対するインデクスをキーにして情報を参照するものです。インデクス値はIDがシステムに登録された時点(Allocateされた)にIDに対して発行されます。

## 応答メッセージAck設定画面

受信した1次メッセージに対して応答するAckの値を設定します。

S1F15 oflack	通常	S10F3 ackc10	通常
S1F17 onlack	3	S10F5 ackc10	通常
S2F23 rsack	通常	S14F9 objack	通常
S2F41 hcack	通常	S14F11 objack	通常
S2F43 rsack	通常	S14F19 svcack	通常
S2F45 lvack	通常	S14F21 dataack	通常
S2F49 hcack	通常	S15F3 rmack	通常
S3F17 caack	通常	S15F5 rmack	通常
S3F23 caack	通常	S15F13 rmack	通常
S3F25 caack	通常	S16F5 acka	True
S3F27 caack	通常	S16F11 acka	True
S3F35 rpmack	通常	S16F15 acka	True
S7F1 ackc7	通常	S16F17 acka	True
S7F3 ackc7	通常	S16F27 acka	True
S7F23 ackc7	通常		

デフォルトに戻す      設定      Quit

本画面は、装置側が受信する1次メッセージに対して応答メッセージを相手に返しますが、そのときに返すAckアイテムの値を故意にエラーを示す値を返すようにするための画面です。

以下のように操作します。

1. Ackの値をエラーにしたいメッセージのコンボボックスの選択を変えて行います。
2. その後、設定ボタンをクリックします。これでOKです。

例えば、S1F17に対し、onlack = 3 を返すようにするためには、左の画面のようにS1F17のonlackを3に選択し、設定ボタンをクリックします。

ログモニターの通信ログを下に示します。

### ログモニターの表示

```
07/02 15:28:21 PT-01 Recv S1F17 len=0010 dvid=1234 W blk=0000 sybt=00000062
07/02 15:28:21 PT-01 Send S1F18 len=0013 dvid=1234 blk=0000 sybt=00000062
<B[1]=x03>
```

## WPシミュレーション画面

The screenshot shows the 'WP 自動シミュレーション' (WP Automatic Simulation) window. At the top, a green banner reads '画面左下の WP処理開始 ボタンのクリックで始めます。' (Start by clicking the WP processing start button in the bottom left corner). The main area displays a simulation of a wafer processing station with a central 'Work' area and two ports labeled 'ポート-1' and 'ポート-2'. The status bar at the top indicates 'IDLE'. The interface includes various control buttons such as '処理開始' (Start), '処理終了' (End), 'SMap照合' (SMap Check), 'FOUP開' (FOUP Open), 'CarID照合' (CarID Check), 'ポート終了' (Port End), '搬出開始' (Unload Start), 'ポート開始' (Port Start), '終了予約' (End Reservation), 'アラーム' (Alarm), 'アラーム確認' (Alarm Confirmation), 'RESET', '一旦停止' (Stop), '再開' (Restart), and '終了' (End). Below the simulation area, there are input fields for 'CarrierID' (set to 'CARID\_01'), 'Usage', 'RCPID', 'Location', 'PrJobID', and 'CJobID'. There are also buttons for 'ONLINE', 'OFFLINE', 'REMOTE', and 'LOCAL', and a section for 'アクセスモード' (Access Mode) with 'AUTO' and 'MANUAL' options. The 'PLC Simulator' section includes a file path 'sample.pls' and a 'PLC TCP port' of '5930'. At the bottom, there is a 'WP処理開始' button and a 'PLCSim使用選択' (PLCSim Use Selection) dropdown menu set to '使用しない' (Do not use).

単純な WP (Wafer Processing) のシミュレーションをPLCSimシミュレータを使って行います。

### PLCSim使用選択

“使用しない” は、自動的にシーケンスを進めます  
“使用する” は、PLCSimと連携します。

本画面の **WP開始** ボタンのクリックでWP処理を開始します。

以降、処理の進行はHOSTとのHSMS通信を開始し、自動的に処理が進行します。

ポートが2個あり、交互にキャリアが搬入され、処理が終わったら、搬入したポートから搬出されて1サイクルの終了です。

**終了予約** ボタンをクリックし、次に搬出終了した時点で、終了させることができます。

処理はつぎのように進行します。

上の処理状態の欄に状態が表示されます。進行に従って、ロード開始、ロード終了、Car照合などボタンの状態が変化します。ポートから先、WORKまでのキャリア(Wafer)の移動状態も表示されます。

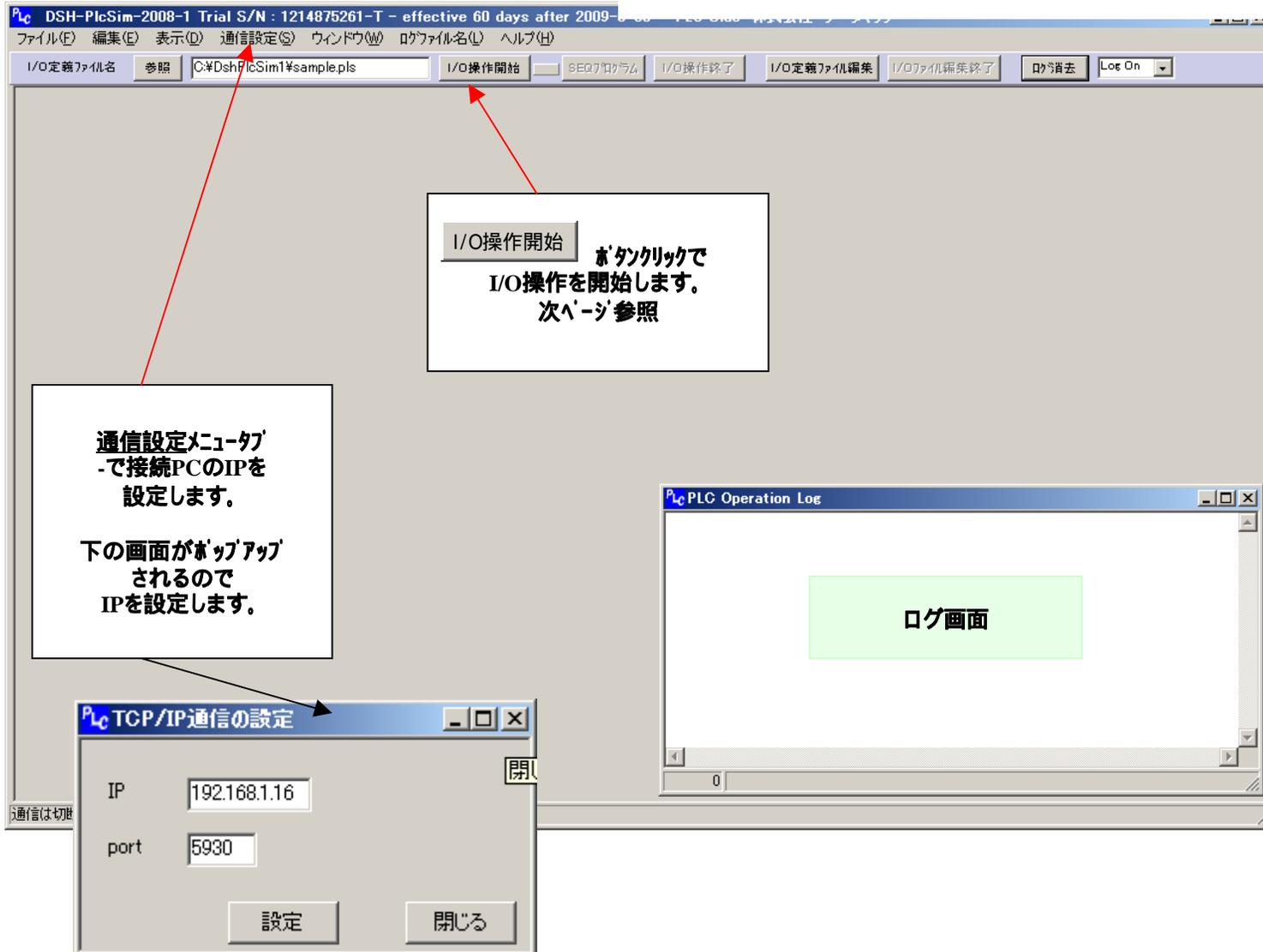
処理中にS5F1,S10F1を受信したら、それらが表示されます。

S1F15,S1F17,S3F25,S3F27受信によってコントロールモードなどの表示が変わります。

また装置が送信したS6F11、ホストから受信するS3F17,S15F13,S16F11,S14F9,S14F11のメッセージはメイン画面にログ表示されます。

# PLCシミュレータ起動と操作

PLCシミュレータの開始  
¥DSHEng4¥dshplcsim¥dshplcsim.exeを起動すると下の画面が表示されます。



# PLCシミュレータ操作

I/O操作開始ボタンによってこの画面が表示されます。  
次に自動的にテストできるようにシーケンスプログラムを開いて実行します。

The screenshot displays the PLC simulation software interface. At the top, the title bar reads "DSH-PlcSim-2008-1 Trial S/N : 1214875261-T - effective 60 days after 2009-6-30 PLC Side 株式会社 データマップ". Below the title bar is a menu bar with options like "ファイル(F)", "編集(E)", "表示(D)", "通信設定(S)", "ウィンドウ(W)", "ログファイル名(L)", and "ヘルプ(H)".

Two windows are open:

- PLC 接点入力On/Off マップ**: A table showing input points (X-addr) from 0000 to 0170. The "ON" and "OFF" columns are currently empty.
- PLC 接点出力On/Off マップ**: A table showing output points (Y-addr) from 0000 to 0140. The "ON" and "OFF" columns are currently empty.

A red arrow points from the "I/O操作開始" button in the top toolbar to the "PLC 接点出力On/Off マップ" window.

A text box with a black border contains the following text:

SEQプログラム ボタンをクリックすると  
右下のシーケンスプログラム画面が表示  
されます。

The "シーケンスプログラミング" window is also open, showing a toolbar with buttons like "新規作成", "開く", "開きなおす", "上書き保存", "別名で保存", "終了", "実行", "再開", "ステップ", "停止", "BP設定", "全BP解除", and "Oms". Below the toolbar is a table for sequence programming with columns for "コメント", "ステップ", "制御対象ポイント名", "On/Off", "条件になるポイント名", "On/Off", and "コメント".

A red arrow points from the "開く" button in the "シーケンスプログラミング" window to a text box:

開く 開きなおす ボタンをクリック  
し、seq\_300.prgファイルを開きます。  
次頁に続きます。

At the bottom left of the main interface, a status bar indicates "通信は切断状態です。"

# PLCシミュレータ操作/シーケンス開始

シーケンス・プログラミング - C:\MSHPLC\Sim\seq\_300.prg

実行

Topに戻る 実行 再開 ストップ 停止 BP設定 全BP解除 Ono

ステップ	ステップ名	On/Off	条件になるステップ名	On/Off	注
1	SALLX	Off			
2	SALLY	Off			
3	SDELAY	5			
4	//** Main				
5	SLABEL_000				
6	SCALL	SLABEL_100			Port-1 Load
7	SDELAY	3			
8					
9	SCALL	SLABEL_200			
10	SDELAY	3			
11					
12	SGOTO	SLABEL_000			
13					
14	//** Port-1				
15	SLABEL_100				
16	\$CARDID	1			
17	X_Port1RdyToLoad	On	Y_Port1RdyLoadLamp	On	
18					
19	X_Port1StartLoad	On			
20	Y_Port1RdyLoadLamp	Off			

PLC Operation Log

```

09-07-09 18:42:54.38 ** End of PLC I/O definition file compilation.
09-07-09 18:42:54.42 number of error = 0
09-07-09 18:42:54.44 <PlcSim Compilation result >
09-07-09 18:42:54.44 source file (in) = C:\MSHPLC\Sim\Wsample.pls
                                (out) = C:\MSHPLC\Sim\Wsample.PLD
                                (comment) = I/O definition loading
    
```

Topに戻る 実行 の順にボタンをクリックして、シーケンスを先頭から実行します。

この後、装置デモプログラムからの信号を待機します。装置との接続が済むと矢印の      のランプが点灯します。