

DSH-PLCSim PLC シミュレータ

PLC 制御デモ・プログラムについて (VB6.0 言語版)

2008年7月

株式会社データマップ

文書番号 DSHPLCSIM-08-30390-00

【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。
- ・ MELSEC は三菱電機株式会社殿の登録商標です。
- ・ 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.	2008.7	初版	
2.			
3.			
4.			

目 次

1 . 概要	1
[関連文書]	1
2 . 構成	2
3 . 開発環境と必要なファイル.....	3
3 . 1 デモプログラム側.....	3
3 . 2 DSH-PLCSim シミュレータ側	3
4 . デモ内容	4
4 . 1 I/O 定義ファイル	4
4 . 2 デモプログラムと PLC シミュレータ間の通信と制御	5
5 . 操作	6
5 . 1 PLC シミュレータ側	6
5 . 2 デモプログラム側.....	6
5 . 2 . 1 起動と画面.....	6
5 . 2 . 2 PLC シミュレータとの接続	7
5 . 2 . 3 マニュアル I/O 制御	8
(1) 入力デバイスの入力操作	8
(2) 出力デバイスの入出力操作	8
5 . 2 . 4 PLC シーケンスプログラムによる制御に対する処理	9
5 . 3 キャリア ID リーダー、スロットマップリーダー、プロセスデータリード操作	11
6 . デモプログラムのプログラミング・ノート.....	12
(1) 通信接続状態の監視	12
(2) シーケンスプログラムの進行処理	12

1. 概要

本説明書は、DSH-PLCSim PLC シミュレータと接続する VB6 言語によるアプリケーションデモプログラムについて説明します。

DSH-PLCSim がサポートする PLC は、三菱電機(株)製 MELSEC シリーズの PLC で、バスインタフェース関数が使用できるものを対象としています。

【関連文書】

DSH-PLCSim 関連文書一覧表

#	文書番号	文書名	注釈
1	DSHPLCSIM-08-30304-00	DSH-PLCSim ユーザーズ・ガイド	PLCシミュレータの機能などが記述されています。
2	DSHPLCSIM-08-30305-00	DSH-PLCSim 操作マニュアル	操作についての具体的な説明書です。
3	DSHPLCSIM-08-30320-00	DSH-PLCSim インタフェース関数説明書	装置コントローラ側が使用できるインタフェース関数の説明書です。
4	DSHPLCSIM-08-30390-00	DSH-PLCSim PLC 制御デモ・プログラムについて (VB6.0 言語版)	VB6 のデモプログラムの説明書です。
5	DSHPLCSIM-08-30391-00	DSH-PLCSim PLC 制御デモ・プログラムについて (VC++6.0 言語版)	Visual C++のデモプログラムの説明書です。

デモは 2 台の PC を使って行ないます。

1 台目の PC 上で PLC の役割を果たす DSH-PLCSim PLC シミュレータを動作させます。

他方の PC 上で、装置コンとローラ側のプログラムに相当するデモプログラムを動作させます

双方のプログラムには、DSHPLC.dll ライブラリを使用します。

アプリケーション側から言えば、このデモプログラムの目的は、DSHPLC.dll のライブラリの動作の確認です。すなわち、MELSEC PLC が提供する QBF インタフェース関数相当の関数を使って PLC との制御を実際にシミュレーションして動作することを確認することです。

デモする機能は DSH-PLCSim が提供する次に挙げる基本的な機能です。

(1) PLC としての機能

QBF_Open(), QBF_Close(), QBF_X_In_BitEx(), QBF_Y_OUT_BitEx()関数による入出力デバイスで制御機能です。

他に、DSH-PLCSim 独自の専用関数として、QBF_SetIOFile()と UPLC_CkCommReady()関数を使用します。

シーケンスプログラムによる自動制御機能のデモも簡単な例で実現します。

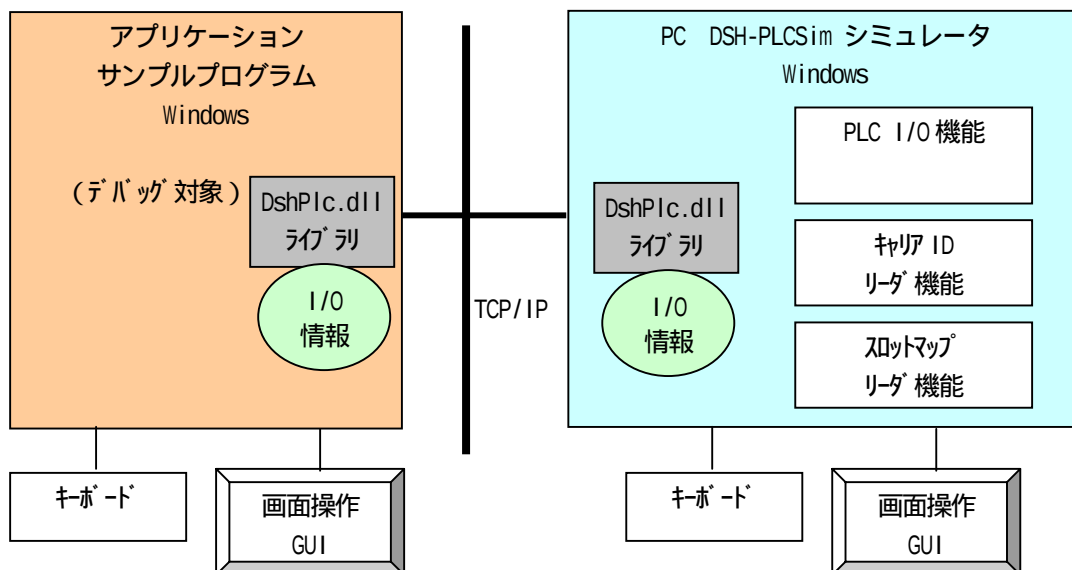
(2) DSH-PLCSim のキャリア ID リーダー、スロットマップリーダー、プロセスデータリードの機能

UPLC_CarIdRead(), UPLC_SlotmapRead(), UPLC_ProcDataRead()関数を使用します。

2. 構成

本デモプログラムが動作するシステム構成は次のようになります。

2 台の PC を使用し、図では左側の PC が装置コントローラに相当するデモプログラムのためのもので、右側の PC が DSH-PLCSim のためのものです。



双方の PC とも Windows の下で、GUI を使って操作します。

3 . 開発環境と必要なファイル

3 . 1 デモプログラム側

(1) 開発環境

Windows-2000、XP または VISTA
Microsoft Visual Basic 6.0

(2) プロジェクトファイル

PlcVB.vbp	- VB6 用デモプログラムファイル
formMain.frm	- メインプログラム
dshplc.bas	- QBF 関数のプロトタイプ
sample_1.bas	- I/O デバイス名のアドレス定義

(3) 実行ファイル

PlcVB.exe	- デモ実行プログラム
DSHPLC.dll	- ライブラリプログラム
sample_1.pls	- I/O デバイス定義ファイル

3 . 2 DSH-PLCSim シミュレータ側

(1) 実行ファイル

DSHPLCSIM.exe	- PLC シミュレータ
DSHPLC.dll	- ライブラリ
sample_1.pls	- I/O デバイス定義ファイル
sample_1.prg	- オート制御プログラム

4 . デモ内容

I/O デバイス信号の定義仕様と PLC シミュレータとの制御内容を示します。

4 . 1 I/O 定義ファイル

sample_1.pls ファイルを使用します。I/O デバイスの定義は次の通りです。

X-addr	入力デバイス名	Y-addr	出力デバイス名
0000	X_LoadEnd	0000	Y_LoadStart
0001	X_MovedSrc	0001	Y_MoveSrc
0002	X_FoupOpened	0002	Y_OpenFoup
0003	X_MovedWork	0003	Y_MoveWork
0004	X_ProcessStarted	0004	Y_StartProcess
0005	X_ProcessEnd	0005	Y_MoveDst
0006	X_MovedDst	0006	Y_CloseFoup
0007	X_FoupClosed	0007	Y_UnloadStart
0008	X_RdyUnload	0008	
0009	X_UnloadEnd	0009	

PLC シミュレータとデモプログラムの双方で同じものを使用します。

互いに、QBF_SetIOFile()関数で I/O 定義ファイル名の引数に対して sample_1.pls を指定します。

I/O 定義ファイルの作成については、「DSH-PLCSim PLC シミュレータ 操作説明書」を参照してください。

4.2 デモプログラムと PLC シミュレータ間の通信と制御

(1) デバイス単位の On/Off 制御

デモプログラムでは、I/O デバイス名を選択し、画面のボタン操作で On/Off の制御を行います。

(2) シーケンスプログラムによる自動 I/O 制御

次のシーケンスの制御を自動で行います。

PLC	デバイスのやり取り	EQCON	注釈
Delay(3)	←	Y_LoadStart = ON	Load 開始
X_LoadEnd = ON	→		Load 完了
Y_LoadStart = OFF			
Delay(3)	←	Y_MoveSrc = ON	Src へ移動
X_MovedSrc = ON	→		同完了
	←	Y_OpenFoup = ON	Foup 開
X_FoupOpened = ON	→		同完了
	←	Y_MoveWork = ON	Work へ移動
X_MovedWork = ON	→		同完了
Delay(5)	←	Y_StartProcess = ON	処理開始
X_ProcessEnd = ON	→		処理終了
Delay(3)	←	Y_MoveDst = ON	Dst へ移動
X_MovedDst = ON	→		同完了
	←	Y_CloseFoup = ON	Foup 閉
X_FoupClosed = ON	→		同完了
X_RdyUnload = ON	→		
Delay(3)	←	Y_UnloadStart = ON	Unload 開始
X_UnloadEnd = ON	→		同完了
PAUSE			
全入力デバイス = OFF	→		
全出力デバイス = OFF	→		

PLC シミュレータは、シーケンスプログラム画面上で、sample_1.prg プログラムファイルを使用します。作成の仕方については、「DSH-PLCSim PLC シミュレータ 操作説明書」を参照してください。

デモプログラム内では、QBF_X_In_BitEx(), QBF_Y_Out_Ex()関数を使ってプログラムを組んで制御します。

(3) キャリア ID リーダー、スロットマップリーダー、プロセスデータ読み

ボタン操作でリードします。

DSHPLC.DLL ライブラリの関数 UPLC_CarIdRead(), UPLC_SlotmapRead(), UPLC_ProcDataRead()関数を使ってプログラムを組んで制御します。

5 . 操作

PLC シミュレータの操作については、「DSH-PLCSim PLC シミュレータ 操作説明書」を参照ください。

5 . 1 PLC シミュレータ側

操作で使用するファイルは以下の通りです。

- (1) I/O デバイス定義ファイルとして、sample_1.pls を指定し、I/O デバイス制御画面を開始します。
- (2) シーケンスプログラムファイルとして、sample_1.prg を使用します。

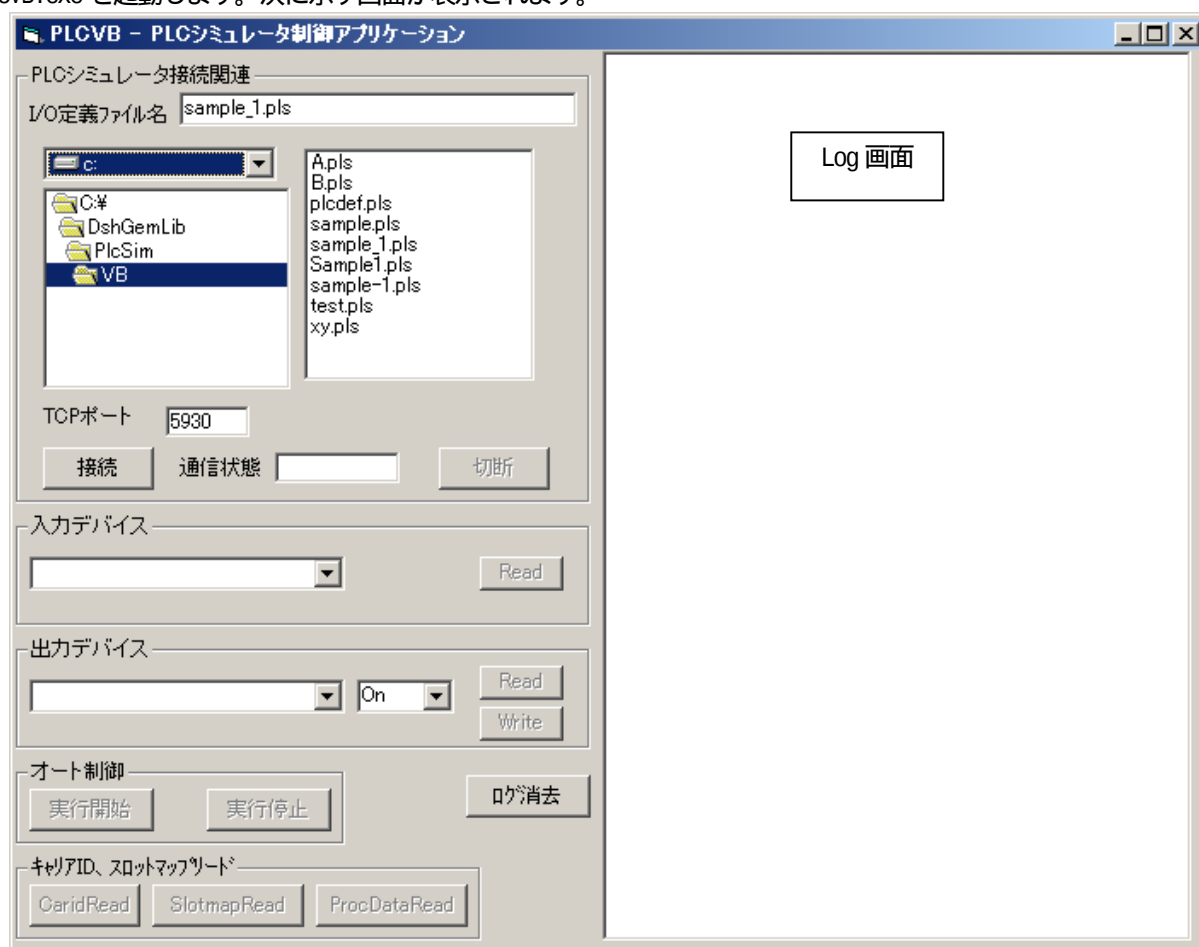
5 . 2 デモプログラム側

lcvb.exe を VB6 開発環境で生成します。

plcvb.exe が保存されている同じディレクトリに sample_1.pls ファイルをコピーしておいてください。

5 . 2 . 1 起動と画面

plcvb.exe を起動します。次に示す画面が表示されます。

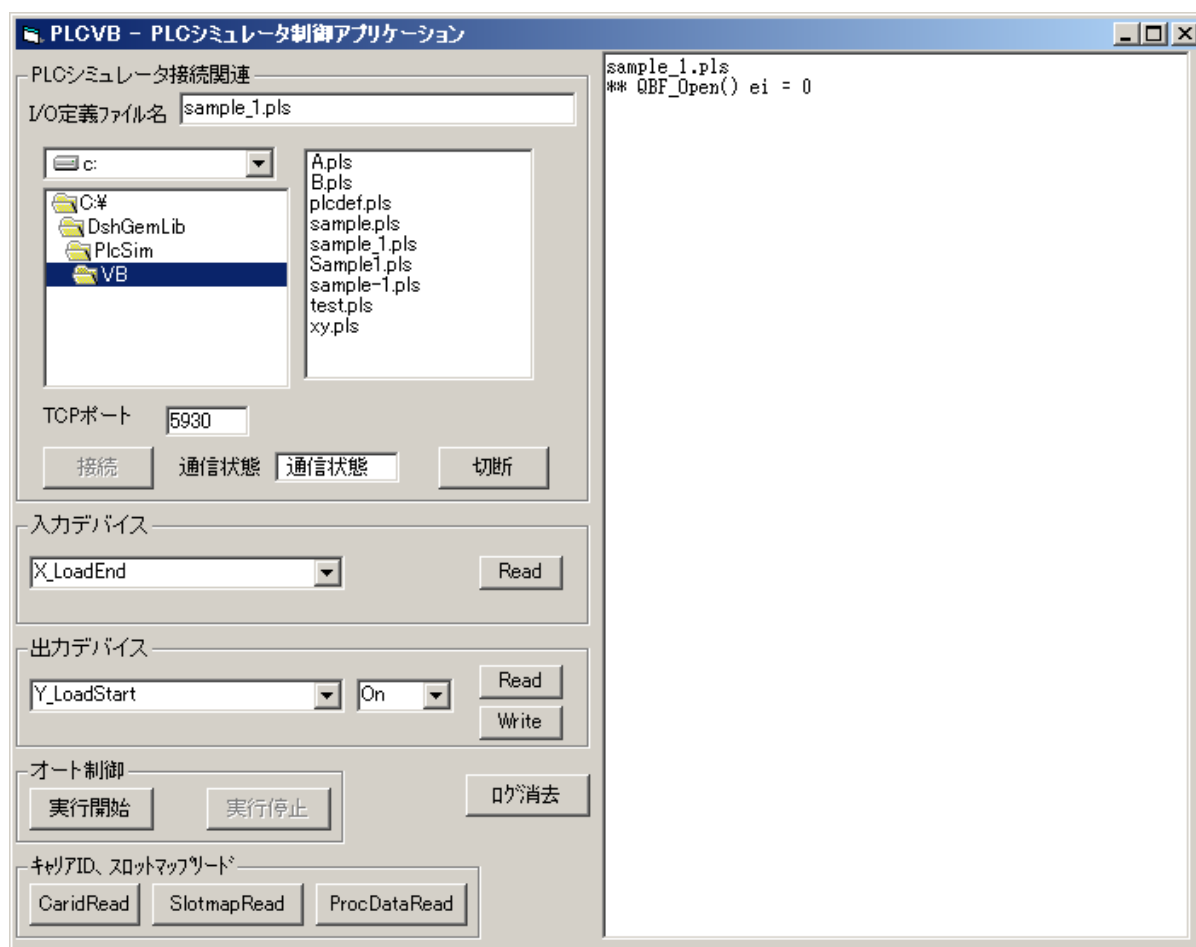


この状態では、**接続**ボタンだけが有効になっています。

5.2.2 PLC シミュレータとの接続

- (1) PLC シミュレータ側において、I/O デバイス制御が開始されていることを確認します。
 また、PLC の通信設定メニューの中の TCP/IP の設定情報、IP、ポートの内容を確認します。
 PLC シミュレータの設定 : IP - デモプログラム側の PC のネットワークアドレスになっていること。
 ポート - デモプログラムと同じ値になっていること。
- (2) 画面の接続ボタンをクリックします。
 他の操作ボタンが有効になります。
 同時に、I/O デバイス定義ファイル sample_1.pls が DSHPLC.dll によって内部に読み込まれます。
 PLC シミュレータと接続できると、通信状態が **接続状態** 表示されます。
 未接続中は、**-----**が表示されます。

画面は次のように変わります。

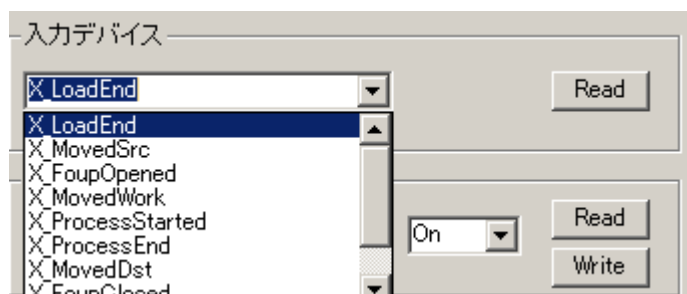


5.2.3 マニュアルI/O 制御

PLC シミュレータとの通信が可能になったあと、入力デバイスの入力ならびに出力デバイスの入出力操作が可能になります。

(1) 入力デバイスの入力操作

入力デバイス名選択コンボボックスリストから入力したいデバイス名を選択し、**Read** ボタンをクリックします。



デモプログラムは、QBF_X_In_BitEx()関数を使ってシミュレータに対し、指定された入力デバイスの On/Off 状態を取得します。

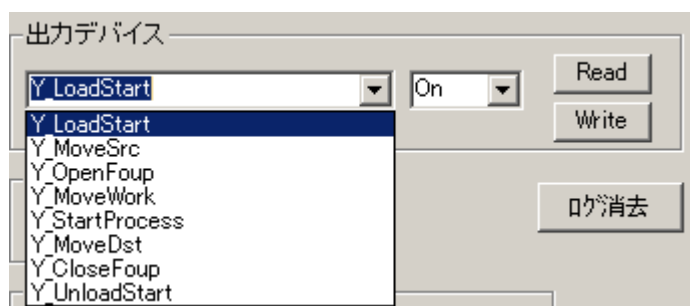
取得された状態はログ画面に、例えば次のように表示されます。

```
** QBF_X_In_BitEx() addr=0x0, 状態=0 (X_LoadEnd)
```

(2) 出力デバイスの入出力操作

出力デバイスの出力は、(1)の入力デバイスの入力操作と同様に行ないます。

出力は、出力デバイス名選択コンボボックスリストから出力したいデバイス名を選択し、そして、On/Off コンボボックス上の On または Off を選択し、**Write** ボタンをクリックします。



デモプログラムは、QBF_Y_Out_BitEx()関数を使ってシミュレータに対し、指定された出力デバイスの On/Off 状態を設定します。

出力した結果はログ画面に、例えば次のように表示されます。

```
** QBF_Y_Out_BitEx() addr=0x0, 状態=1 (Y_LoadStart)
```

出力された On/Off 状態は PLC シミュレータの画面で確認できます。

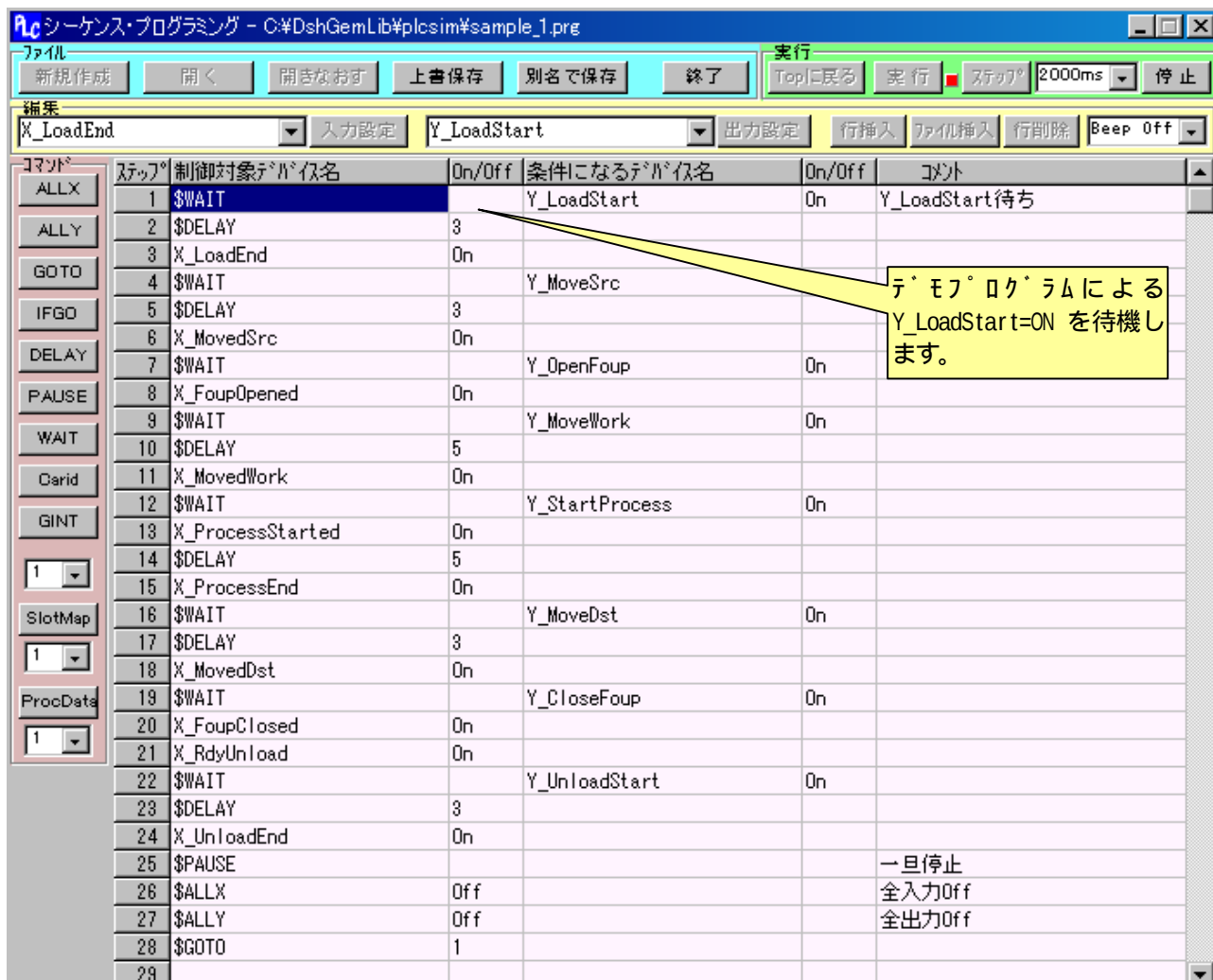
入出力関数実行の結果、エラーが発生した場合は、その旨の表示を行ないます。

5.2.4 PLC シーケンスプログラムによる制御に対する処理

4.2で説明しましたが、PLC シミュレータ側で行なわれるシーケンスプログラムによる自動制御動作に対し、その相手をするための処理を行います。

操作は、以下の順に行ないます。

- (1) 先に、PLC シミュレータ側の操作で、シーケンスプログラムを開始しておきます。
sample_1.prg シーケンスプログラムファイルを開いて、**実行**ボタンのクリックします。
PLC シミュレータのシーケンスプログラム画面は次のようになります。



- (2) その後、デモプログラム側はオート制御の中の**実行開始**ボタンをクリックし、シーケンスを開始します。



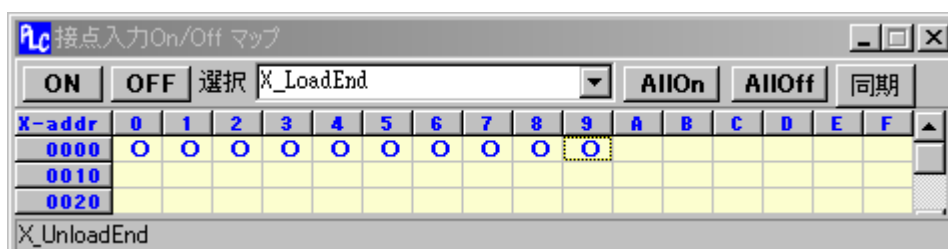
これで、デモプログラムは、QBF_Y_Out_BitEx()関数を使って Y_LoadStart 信号を ON にします。
(PLC シミュレータは Y_LoadStart 信号が ON になるのを \$WAIT コマンドで待機しています。)

- (3) 双方で実行開始が行なわれると、4.2の制御シーケンスに従って制御が進行していきます。
PLC シミュレータ側では、I/O デバイス制御画面上の信号がシーケンスの進行と共に信号が On 表示になっていく様子を目視できます。

シミュレータ側では、シーケンス-24 の X_UnloadEnd 信号=ON の出力の後、\$PAUSE コマンドで実行停止します。

また、デモプログラム側では、X_UnloadEnd=ON の入力確認でオート制御処理を終了し、停止します。

最終的に、PLC シミュレータ側の I/O デバイス制御画面は次のようになります。

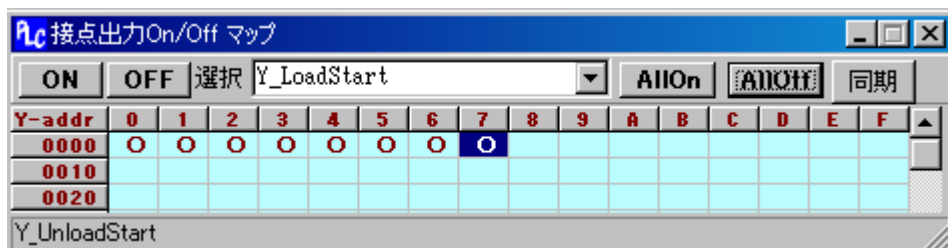


PLC 接点入力On/Off マップ

ON OFF 選択 X_LoadEnd AllOn AllOff 同期

X-addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	O	O	O	O	O	O	O	O	O	O						
0010																
0020																

X_UnloadEnd



PLC 接点出力On/Off マップ

ON OFF 選択 Y_LoadStart AllOn AllOff 同期

Y-addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	O	O	O	O	O	O	O	O								
0010																
0020																

Y_UnloadStart

デモプログラム側のログ画面には次のように信号の変化が表示されます。

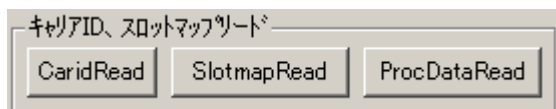
```

** Auto Control Start **
step-1 ==> Y_LoadStart      = ON
step-2 ==> X_LoadEnd        = ON
step-3 ==> Y_MoveSrc        = ON
step-4 ==> X_MovedSrc       = ON
step-5 ==> Y_MoveOpenFoup   = ON
step-6 ==> X_FoupOpened     = ON
step-7 ==> Y_MoveWork       = ON
step-8 ==> X_MovedWork      = ON
step-9 ==> Y_StartProcess   = ON
step-10 ==> X_ProcessStarted = ON
step-11 ==> X_ProcessEnd    = ON
step-12 ==> Y_MoveWork      = ON
step-13 ==> X_MovedDst      = ON
step-14 ==> Y_CloseFoup     = ON
step-15 ==> X_FoupClosed    = ON
step-16 ==> X_RdyUnload     = ON
step-17 ==> Y_MoveCloseFoup = ON
step-18 ==> X_UnloadEnd     = ON
** End of Auto Control

```

5.3 キャリアIDリーダー、スロットマップリーダー、プロセスデータリード操作

PLC シミュレータの“キャリア ID、スロットマップ...”画面に表示されている情報を dshplc ライブラリの専用関数を使って取得します。操作は、次のボタンのクリックで行ないます。



(1) キャリア ID の取得操作

CaridRead ボタンをクリックします。デモプログラムは、UPLC_CarIdRead() 関数を使って取得します。取得された ID は次のようにログ画面に表示されます。

```
** UPLC_CarIdRead() - ID = CARID_03
```

(2) スロットマップ情報の取得操作

SlotmapRead ボタンをクリックします。デモプログラムは、UPLC_SlotmapRead() 関数を使って取得します。取得されたスロットマップ情報は次のようにログ画面に表示されます。

```
** UPLC_SlotmapRead() OK
1234567890123456789012345
100111111110111111111101
```

(3) プロセスデータの取得操作

ProcDataRead ボタンをクリックします。デモプログラムは、UPLC_ProcDataRead() 関数を使って取得します。取得されたデータ (7 個分) は次のようにログ画面に表示されます。

```
** Procdata( 0 ) = 111.1
** Procdata( 1 ) = 222.2
** Procdata( 2 ) = 333.3
** Procdata( 3 ) = 444.4
** Procdata( 4 ) = 555.5
** Procdata( 5 ) = 666.6
** Procdata( 6 ) = 777.7
```

6 . デモプログラムのプログラミング・ノート

(1) 通信接続状態の監視

接続ボタンで PLC シミュレータとの通信接続を開始した後、実際に TCP/IP プロトコルで接続できたかどうかの監視と通信状態表示の更新を行ないます。

監視は、UPLC_CkCommRead()関数を使って 5 0 0 MS 間隔で行ないます。

TIMER1 でインターバルをとり、そのイベント発生時に監視処理を行ないます

(2) シーケンスプログラムの進行処理

オート制御の処理は、4 . 2 で示した PLC シミュレータ間の信号のやり取りを状態番号で表現して、その状態番号 (1 ~ 1 8) における処理を必要に応じて行なうことにします。この状態番号領域として、next_step (Integer) を使用します。

オート制御ボタンでの開始時に、next_step = 1 にセットします。

その後は、TIMER2 で 5 0 0 MS の周期で、next_step の値にしたがって、出力デバイスの ON 出力あるいは入力デバイスの状態を入力して next_step の値を進めていきます。

詳しくは、プログラムファイルの中の TIMER2 イベントハンドラーの処理を参照してください。