DSH-PLCSim　PLC シミュレータ

ユーザーズ・ガイド

２００８年6月

株式会社データマップ

文書番号　DSHPLCSIM-08-30304-00

［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.	2008.6	初版	
2.			
3.			
4.			

目　次

１．概要

本説明書は、DSH-PLCSim PLC シミュレータの役割、機能ならびに操作の概要について説明します。

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

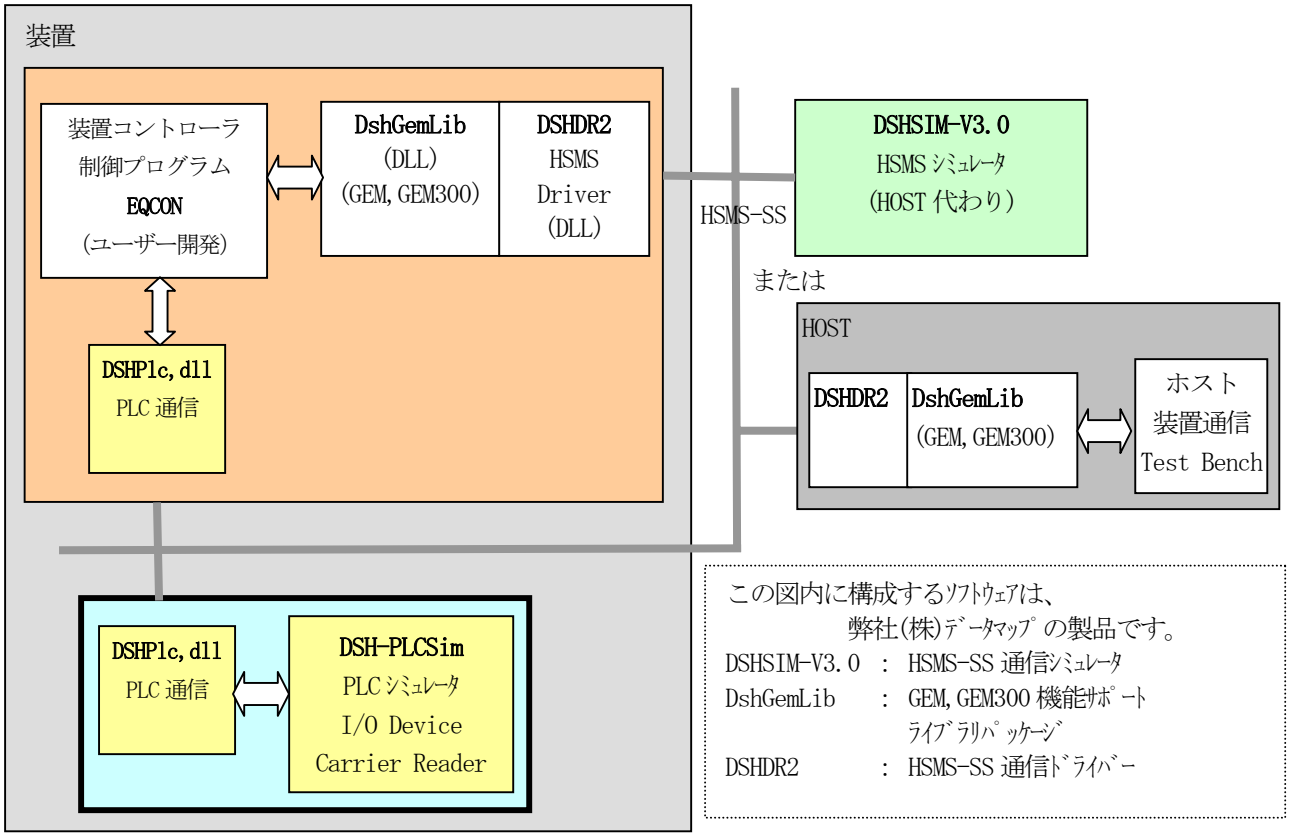
#	文書番号	文書名	注釈
1	DSHPLCSIM-08-30304-00	DSH-PLCSimﾕｰｻﾞｰｽﾞ・ｶﾞｲﾄﾞ	この文書です。
2	DSHPLCSIM-08-30305-00	DSH-PLCSim 操作ﾏﾆｭｱﾙ	操作についての具体的な説明書です。
3	DSHPLCSIM-08-30320-00	DSH-PLCSim ｲﾝﾀﾌｪｰｽ関数説明書	装置ｺﾝﾄﾛｰﾗ側が使用できるｲﾝﾀﾌｪｰｽ関数の説明書です。
4	DSHPLCSIM-08-30390-00	DSH-PLCSim PLC 制御デモ・プログラムについて (VB6.0 言語版)	VB6 のﾃﾞﾓﾌﾟﾛｸﾞﾗﾑの説明書です。
5	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++のﾃﾞﾓﾌﾟﾛｸﾞﾗﾑの説明書です。

１．１　DSH-PLCSim の目的と役割

DSH-PLCSim の目的は、半導体製造工場内に設置される半導体製造装置の制御プログラムの開発をトータルシステムとして総合的にシミュレーションできるようにするためのソフトウェアツールとして、きめ細かくその役割を果たすことです。

DSH-PLCSim システムで提供されるソフトウェアは DSH-PLCSim シミュレータと DSHPlc ライブラリです。

下のブロック図は、システムの一例ですが、開発過程において、PLC の代わりに DSHPLC シミュレータを利用した構成を示しています。



この例のシステムの中で、DSHPLC シミュレータは、EQCON に接続される仮想の PLC(Programmable Logic Controller)、キャリア ID リーダー、スロットマップリーダーの役割を果たします。

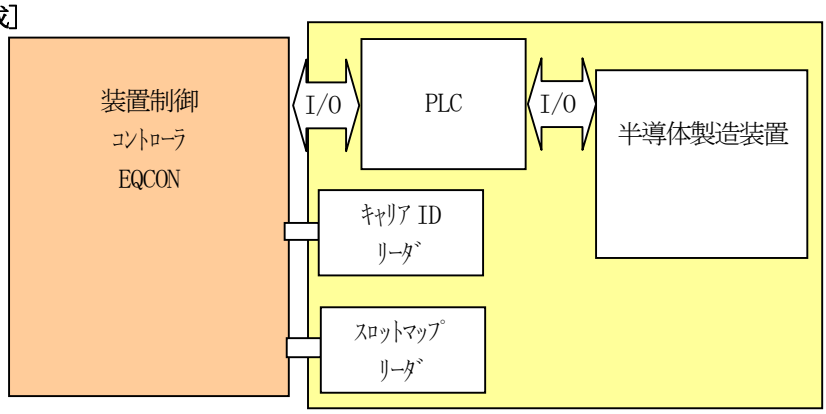
すなわち、DSH-PLCSim は、PLC、キャリア ID リーダー、スロットマップリーダーの役割と機能を論理的に受け持ち、EQCON のプログラムテストを基本的なテストから総合的なシーケンスのテストまでのプログラム処理を、バグが無くなるまで繰り返して簡単に実行できるようにするための環境を提供するものです。

なお、装置コントローラは、プログラム開発とテスト段階では、MELSEC のバスインタフェースライブラリプログラムの代わりに DSHPLC.DLL ライブラリプログラムを使用することになります。

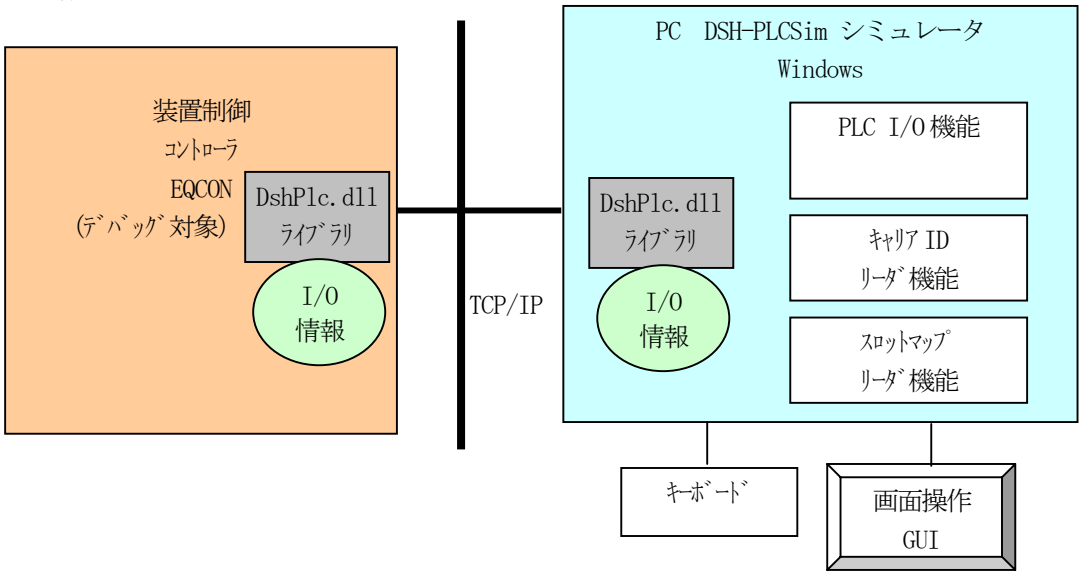
１．２　構成

実機と DSH-PLCSim におけるシステム構成は大体以下のようになります。

［実機の構成］



［DSH-PLCSim の構成］



DSH-PLCSim は、PLC, キャリア ID リーダー、スロットマップリーダーの基本的な３つの動作をシミュレートします。
EQCON との接続は、Ethernet を使って、TCP/IP プロトコルで行ないます。

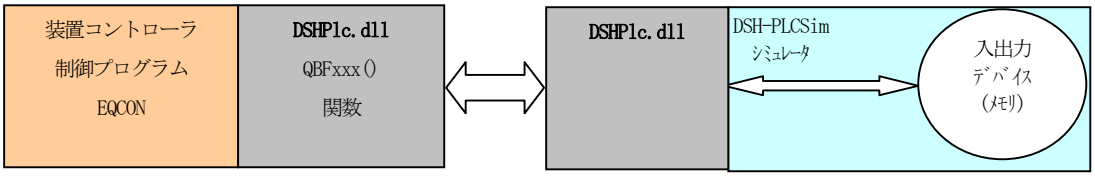
１．３　機能概要

ここでは、①装置コントローラ、②PLC シミュレータのそれぞれについての機能を簡単に説明します。

１．３．１　装置コントローラの概略入出力機能

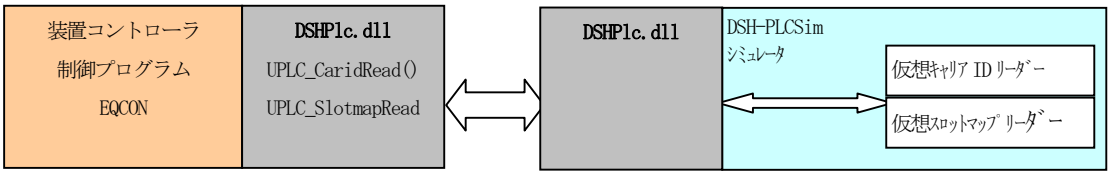
装置コントローラは、DSHPlc.dll ライブラリ関数を通して、PLC シミュレータとの入出力の制御を行ないます。

まず、接点入出力デバイス制御のためにMELSEC が提供するバスインタフェース関数があります。



（1）バスインタフェース関数
QBF_Open()関数を始めとしたバスインタフェース関数をサポートします。
MELSEC によって提供される関数のうち、Open, Close と入出力信号の Input,Output 関数のサポートを行ないます。

他に、DSH-PLCSim では、搬送単位であるキャリア ID リーダー機能と、その中に含まれるウェハー用スロットの有無信号であるスロットマップリーダー機能が提供されます。(DSH-PLCSim 画面の仮想リーダーから読みます)



（2）キャリア ID 読取り関数
UPLC_CarIdRead()関数を使います。

（3）スロットマップ読取り関数
UPLC_SlotmapRead()関数を使います。

１．３．２　PLC シミュレータ概略機能

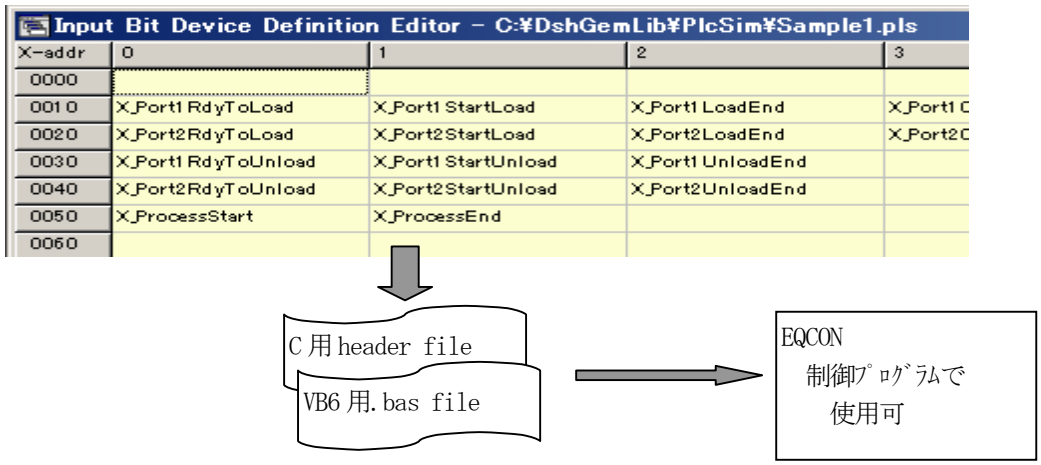
PLC シミュレータ機能として、以下の機能があります。

（1）I/O デバイス定義機能

入出力接点デバイス信号の定義を I/O マップ画面の操作でおこないます。
各信号には、名前とアドレスが与えられます。アドレスは入力、出力それぞれ　0～17FF(16 進)までの空間まで定義できます。
入力信号の名前には”X_”を、出力信号の名前には”Y_” を付けます。

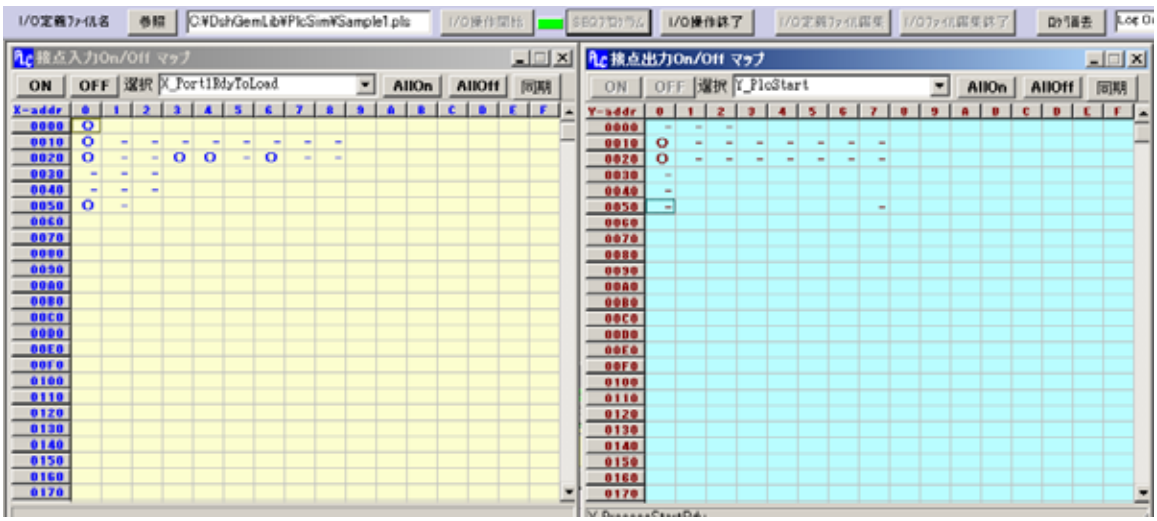
また、定義された I/O デバイス名とそれに与えられたデバイス番号が対応付けられた C, VB6 用ソースファイルが生成されます。EQCON のプログラムをヘダーファイルとして使用することができます。

入力信号定義画面例



（2）入出力制御機能

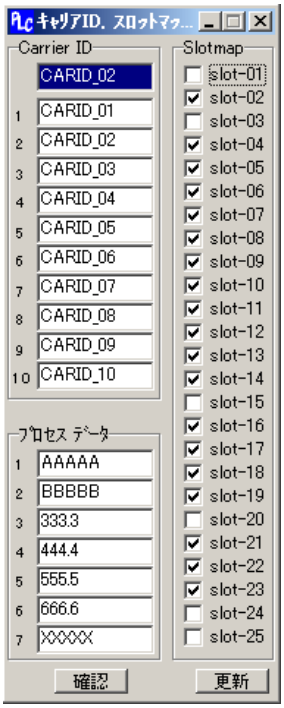
（1）で定義した I/O デバイスの I/O マップ制御画面上での操作であり、信号の On/Off の設定とその情報をシミュレータに接続されている装置コントローラへ送信します。EQCON からは信号読込み関数を使って信号の On/Off 状態を読み出すことができます。これらの操作はマウス操作で簡単にできます。



これらの信号は、（4）で述べる簡易シーケンスプログラムによってシナリオに基づく制御も可能です。

（3）キャリアＩＤリーダー，スロットマップリーダーと処理データ入力機能

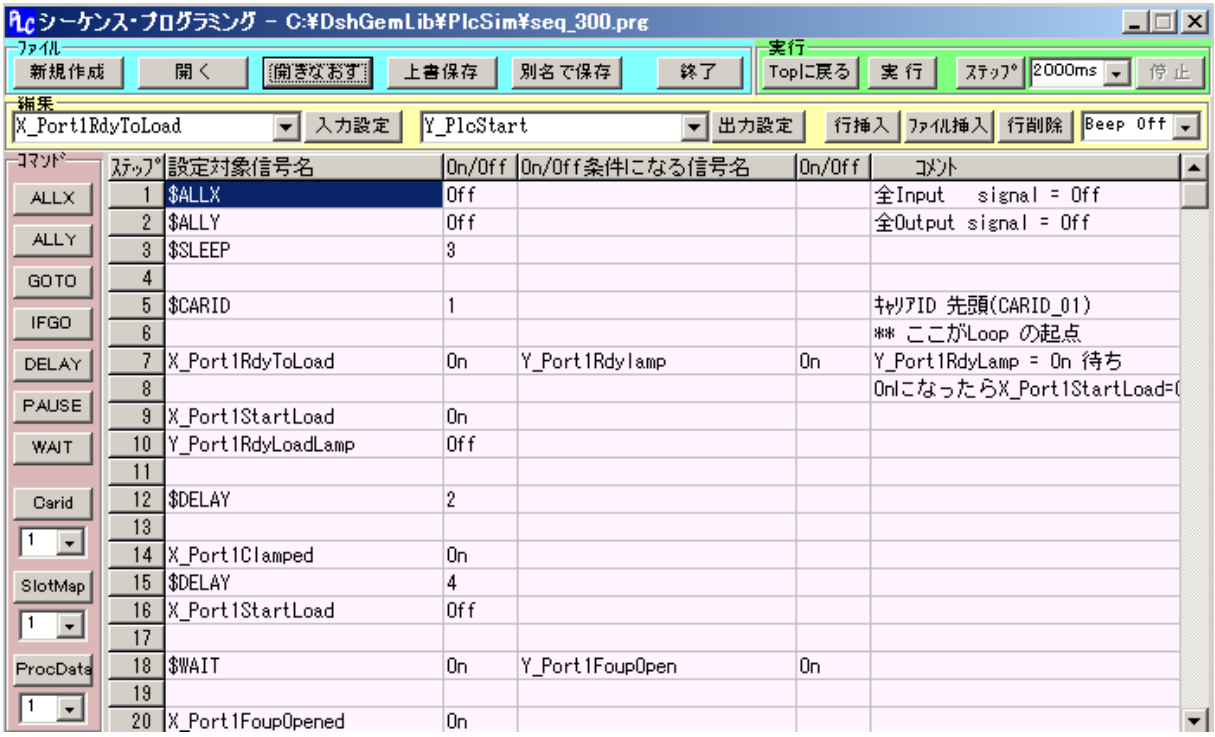
画面上にキャリアＩＤ、スロットマップ、プロセスデータの３種類の情報を設定できます。これらの情報は、EQCON から読み込むことができます。



（4）簡易シーケンスプログラム画面による自動シーケンスの実行機能

簡単なコマンドを使って、表の中に信号の On/Off 条件と設定を並べたり、ステップのジャンプ、時間遅延などの制御を行ないながら、制御シーケンスを作り、それを実行させることができます。

実行すると、自動的に表上に組まれたプログラムを実行してくれます。これで一通りの搬入、処理、搬出までのシナリオ制御を自動的に行なうことが可能です。



２．装置コントローラの入出力機能

ここでは、EQCON における機能を詳細に説明します。

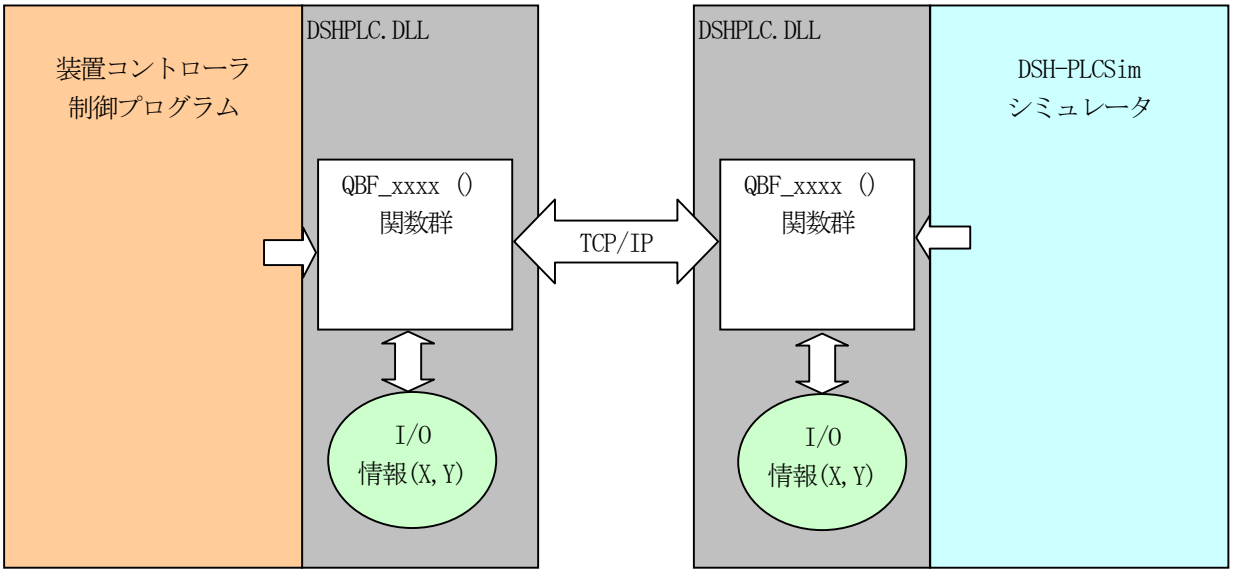
EQCON における機能は、対 DSH-PLCSim に対するインタフェース機能になります。
具体的には、EQCON の制御プログラムがどのように DSH-PLCSim との間で、提供される入出力インタフェース関数を使ってプログラミングするかということになります。

MELSEC をサポートするバスインタフェース関数(QBF_xxxx の名前の関数)の機能を準備しています。

インタフェース関数としては、接点入出力のためのものと、DSH-PLCSim のセットアップのために準備された関数を使用することができます。

また、ほかに DSH-PLCSim の固有の機能として、キャリア ID リーダー、スロットマップリーダーによる情報読取りのための関数も準備されていますので、そちらも使用することができます。

インタフェース関数の機能、使い方の詳しい内容については、**「DSHPLC.DLL インタフェース関数説明書」**に記述されていますので、そちらを参照してください。



EQCON と DSH-PLCSim との間のインタフェースは、DSHPLC.DLL ダイナミックライブラリに含まれる関数を使って取ります。

EQCON, DSH-PLCSim それぞれの DSHPLC.DLL には、I/O 情報を持っており、これらは、互いのインタフェース関数による制御によって、その内容の同期が取られています。

EQCON と DSH-PLCSim とは、TCP/IP プロトコルで接続されますが、Passive, Active の関係は以下のようになります。

EQCON ： Passive (サーバー)　　DSH-PLCSim ： Active(クライアント)

２．１　DSH-PLCSim 通信用インタフェース関数と使い方

EQCON が DSH-PLCSim との間で使用される入出力信号の Read/Write などのインタフェース関数について説明します。

（1）バスインタフェース関数（QBF_xxx）

DSHPLC.dll ライブラリが準備している MELSEC バスインタフェース関数として準備されている関数は次表の通りです。

	I/F 関数	機能
1	QBF_SetIOFile()	入出力定義ﾌｧｲﾙ名と DSH-PLCSim との TCP/IP 通信のための情報を設定します。 DSH-PLCSim 特有の関数です。
2	QBF_Open()	ﾊﾞｽまたは通信回線をｵｰﾌﾟﾝします。 DSH-PLCSim と接続します。
3	QBF_Close()	ﾊﾞｽまたは通信回線をｸﾛｰｽﾞします。 DSH-PLCSim との接続を切断します。
4	QBF_X_In_BitEx()	入力信号(X)を１点入力します。
5	QBF_X_In_WordEx()	入力信号(X)をﾜｰﾄﾞ単位で入力します。
6	QBF_Y_Out_BitEx()	出力信号(Y)を１点出力します。
7	QBF_Y_Out_WordEx()	出力信号(Y)をﾜｰﾄﾞ単位で出力します。
8	QBF_Y_In_BitEx()	出力信号(Y)を１点出力します。
9	QBF_Y_In_WordEx()	出力信号(Y)をﾜｰﾄﾞ単位で出力します。
10	QBF_ToBuf()	ﾃﾞﾊﾞｲｽﾒﾓﾘにﾃﾞｰﾀを書き込みます。 DSH-PLCSim では、ﾃﾞｰﾀ部分のみを書込み対象とします。
11	QBF_FromBuf()	ﾃﾞﾊﾞｲｽﾒﾓﾘからﾃﾞｰﾀを読み出します。 DSH-PLCSim では、ﾃﾞｰﾀ部分のみを読出し対象とします。
12	QBF_RefreshLinkDevice() QBF_WriteLinkDevice() QBF_SEND() QBF_RECV() QBF_UnitInfo() QBF_StartWDT() その他の関数	これらの関数については、何も処理しません。 返却値が必要な関数については、全て = 0 を返却します。

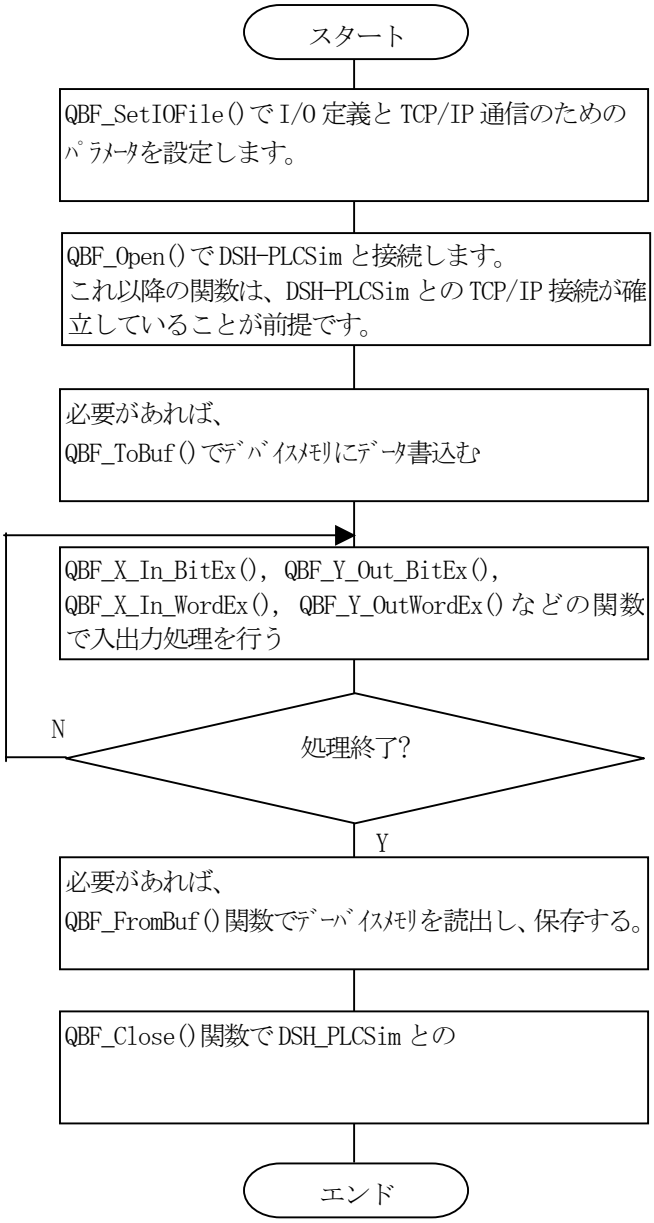
（2）その他の関数　(DSH-PLCSim との通信のための)

DSH-PLCSim シミュレータに対する固有の関数

	I/F 関数	機能
1	UPLC_CkCommReady()	DSH-PLCSim との接続が確立しているかどうかを調べるための関数です。
2	UPLC_XBitSyncReq()	DSH-PLCSim が有する入力信号(X)のﾃﾞｰﾀを読出し、ﾃﾞﾊﾞｲｽｺﾝﾄﾛｰﾗ側のﾃﾞﾊﾞｲｽﾒﾓﾘに格納します。
3	UPLC_YBitSyncReq()	DSH-PLCSim が有する出力信号(Y)のﾃﾞｰﾀを読出し、ﾃﾞﾊﾞｲｽｺﾝﾄﾛｰﾗ側のﾃﾞﾊﾞｲｽﾒﾓﾘに格納します。
4	UPLC_CarIdRead()	ｷｬﾘｱ ID を読出します。 読出すｷｬﾘｱ ID は、DSH-PLCSim の画面上に選択設定された値です。
5	UPLC_SlotmapRead()	ｽﾛｯﾄﾏｯﾌﾟ 情報 (25 ｽﾛｯﾄ分) を読出します。 読出すｽﾛｯﾄﾏｯﾌﾟ 情報は、DSH-PLCSim の画面上に選択設定された On/off 情報です。
6	UPLC_ProcDataRead()	指定した番号のﾌﾟﾛｾｽﾃﾞｰﾀを１個読出します。 読出すﾌﾟﾛｾｽﾃﾞｰﾀは文字列ﾃﾞｰﾀです。 画面には 7 個のﾃﾞｰﾀが設定されています。
7	UPLC_GetXDvName()	入力ﾃﾞﾊﾞｲｽ番号からそのﾃﾞﾊﾞｲｽ名を取得する。
8	UPLC_GetYDvName()	出力ﾃﾞﾊﾞｲｽ番号からそのﾃﾞﾊﾞｲｽ名を取得する。

２．２　インタフェース関数によるプログラミング手順

EQCON における、バスインタフェース関数のプログラミング手順は次の通りです。



３．PLC シミュレータの機能

DSH-PLCSim シミュレータは、PLC（シーケンサー）の機能をシミュレートするものです。

３．１　I/O デバイス定義機能

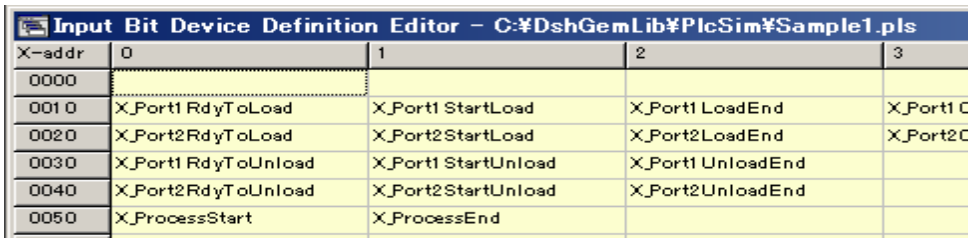
I/O デバイスの定義機能とは、PLC の X,Y デバイスメモリの各ビットをどのような信号名として使用するか、そのアドレス（＝デバイス番号）と信号名を対応付け、XY メモリのマップ上に割り付ける機能です。

システムの外部仕様が決まり、PLC の信号をどのように使用するかの I/O の割付が仕様上で決定されたものを DSH-PLCSim の I/O アドレスマップ画面上で入力操作し、割付情報の結果をディスクファイル上に保管します。

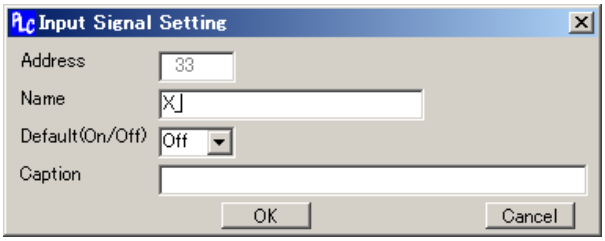
また、一度作成保存した I/O 定義ファイル内で定義した内容を変更、追加などの編集も I/O 定義操作画面上で操作することができます。

（1）I/O 定義画面

下に示す画面は、入力デバイスの設定画面です。
X_addr（16 進で 4 桁）に対応するセルを選択し、そこに入力デバイス名を入れます。

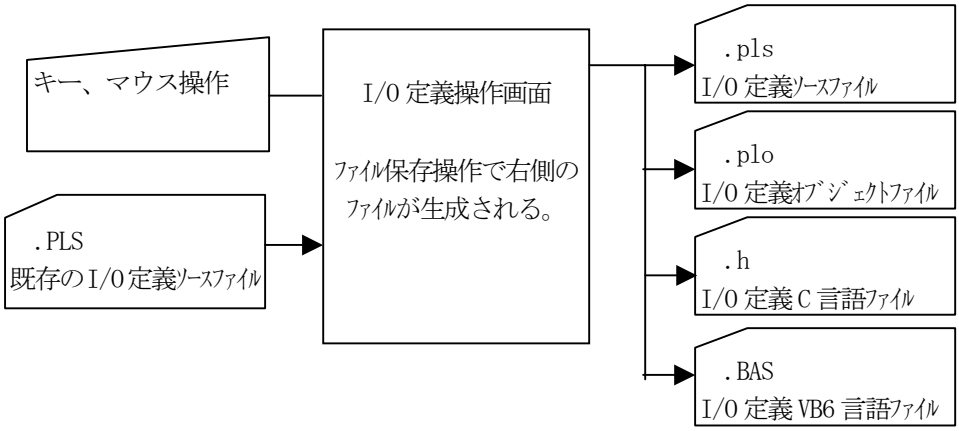


X_addr のセルのダブルクリックによって、ポップアップされる次画面を使って定義することもできます。



（2）生成されるファイル

画面に定義された I/O 定義情報をファイルに保存すると、下図に示すようなファイルが生成されます。



ファイル保存時に、例えば、ファイル名を"sample.pls" と指定された場合に生成されるそれぞれのファイル名の種類と内容は以下のようになります。

	保存ファイル名	拡張子
1	sample.pls ： I/O 定義のﾃｷｽﾄﾌｧｲﾙです。保存操作時に指定された名前で保存されます。	.PLS
2	sample.plo ： sample.pls をｺﾝﾊﾟｲﾙした結果ﾌｧｲﾙです。I/O 制御操作画面開始時にしようされます。	.PLO
3	sample.h ： C/C++言語ﾌﾟﾛｸﾞﾗﾑ用のﾍﾀﾞｰﾌｧｲﾙです。I/O の名前に対するｱﾄﾞﾚｽ値のﾏｸﾛ定義です。	.h
4	sample.bas ： VB6 用ﾌﾟﾛｸﾞﾗﾑ用のﾌｧｲﾙです。I/O の名前に対するｱﾄﾞﾚｽ値のﾏｸﾛ定義です。	.bas

表の 3, 4 の .h, .bas ファイルは、EQCON 制御プログラムのソースファイルとして使用できるようにフォーマットが整えられて生成されます。

（3）ファイルの内容(例)

① .pls – I/O 定義ソースファイル

```
//----- Input Device Bit Definition
def_in_bit  X_Port1RdyToLoad{
            addr:    0x10
            nominal: 0
            caption: ""
            }
def_in_bit  X_Port1StartLoad{
            addr:    0x11
            nominal: 0
            caption: ""
            }
//----- Output Device Bit Definition
def_out_bit Y_PlcStart{
            addr:    0x1
            nominal: 0
            caption: ""
            }
def_out_bit Y_PlcStop{
            addr:    0x2
            nominal: 0
            caption: ""
            }
```

② .plo – I/O 定義オブジェクトファイル

```
INB:NAME00000010X_Port1RdyToLoadADDR00000010NOMI00000000CAPT00000000
INB:NAME00000010X_Port1StartLoadADDR00000011NOMI00000000CAPT00000000
INB:NAME0000000eX_Port1LoadEndADDR00000012NOMI00000000CAPT00000000
INB:NAME0000000eX_Port1ClampedADDR00000013NOMI00000000CAPT00000000
INB:NAME00000011X_Port1FoupOpenedADDR00000014NOMI00000000CAPT00000000
INB:NAME00000011X_Port1FoupClosedADDR00000015NOMI00000000CAPT00000000
INB:NAME00000013X_Port1MovedCarWorkADDR00000016NOMI00000000CAPT00000000
INB:NAME00000012X_Port1MovedCarDstADDR00000017NOMI00000000CAPT00000000
```

③ .h – C/C++用ファイル

```
/*-------------------------------------------------------*/
/*  C:¥DshGemLib¥PlcSim¥Sample1.h - PlcSim Addr Definition*/
/*-------------------------------------------------------*/
#define  X_Port1RdyToLoad              0x0010          // 16
#define  X_Port1StartLoad              0x0011          // 17
#define  X_Port1LoadEnd                0x0012          // 18
#define  X_Port1Clamped                0x0013          // 19
#define  X_Port1FoupOpened             0x0014          // 20
#define  X_Port1FoupClosed             0x0015          // 21
#define  X_Port1MovedCarWork           0x0016          // 22
#define  X_Port1MovedCarDst            0x0017          // 23
#define  X_Port1LoadReq                0x0018          // 24
#define  X_Port2RdyToLoad              0x0020          // 32
#define  X_Port2StartLoad              0x0021          // 33
```

④ .bas – VB6 用ファイル

```
Attribute VB_Name = "PLC_addr.bas"
'-------------------------------------------------------
' C:¥DshGemLib¥PlcSim¥Sample1.bas - PlcSim Addr definition
'-------------------------------------------------------
Public Const X_Port1RdyToLoad =        16              ' 0x00000010
Public Const X_Port1StartLoad =        17              ' 0x00000011
Public Const X_Port1LoadEnd =          18              ' 0x00000012
Public Const X_Port1Clamped =          19              ' 0x00000013
Public Const X_Port1FoupOpened =       20              ' 0x00000014
Public Const X_Port1FoupClosed =       21              ' 0x00000015
Public Const X_Port1MovedCarWork =     22              ' 0x00000016
Public Const X_Port1MovedCarDst =      23              ' 0x00000017
Public Const X_Port1LoadReq =          24              ' 0x00000018
Public Const X_Port2RdyToLoad =        32              ' 0x00000020
Public Const X_Port2StartLoad =        33              ' 0x00000021
```

３．２　I/O 制御画面での制御機能

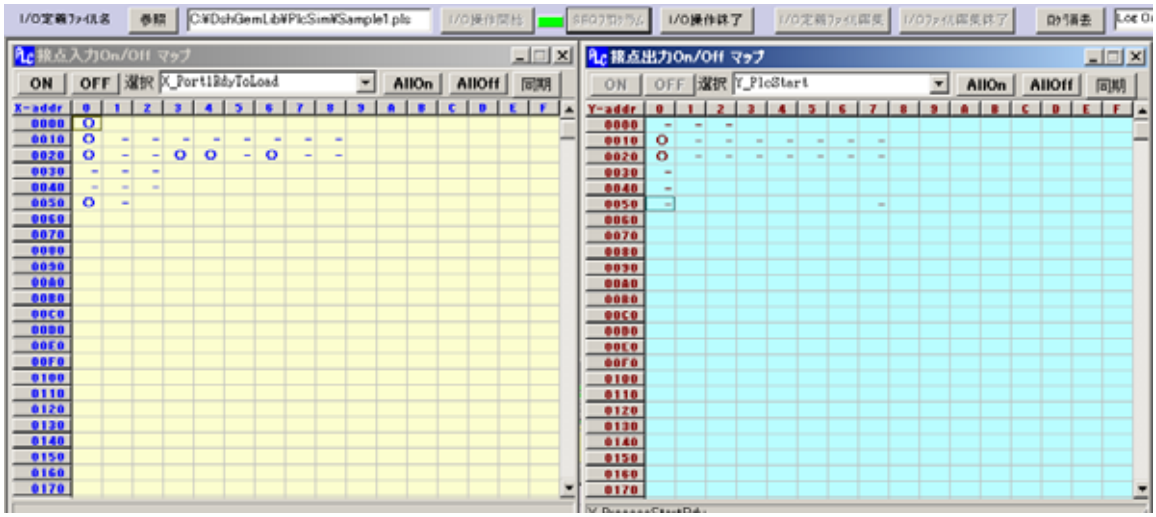
３．１で作成した I/O 定義ファイルの入出力制御を実際に EQCON と接続して実行する機能です。

３．２．１　I/O 制御の開始

I/O 定義ファイルをメイン画面で指定し、I/O 操作開始操作を行なうと、DSH-PLCSim は次のような準備処理をします。

（1）相手の EQCON との TCP/IP の接続を行ないます。
TCP/IP の接続上は、EQCON が Passive で、DSH-PLCSim が Active として接続します。
（注）DSH-PLCSim 側では、接続のための EQCON の **IP** と **ポート**を通信設定メニューで設定できます。

（2）入力、出力のマップ画面を別々に表示し個別に I/O 信号を選択し、On/Off の制御ができるようにします。



マップ画面上では、信号が On になっているものは、〇で表示され、Off になっているものは － で表示されます。

３．２．２　I/O 制御の内容と操作

DSH-PLCSim における入力と出力信号の操作は、それぞれの画面で行ないますが、操作の方法は同じです。

（1）信号の On/Off 制御

２つの方法があります。

①信号名をコンボボックスで選択し、ON または OFF ボタンをクリックする方法

②目的の信号が割当てられているマップ上のセルをダブルクリックする方法

この場合、On<->Off が交互に変化します。

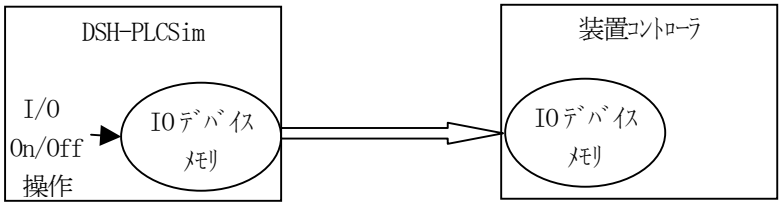
また、全信号を ON/OFF したい場合は、それぞれ、ALLON、ALLOFF ボタンのクリックします。

操作された信号の状態は、I/O マップ上の当該セルに On の場合は、”○”で、Off の場合は “－“ が表示されます。

（注）入出力デバイスに対する On/off 制御は３．４で説明するシーケンスプログラムによっても制御できます。

（2）EQCON への送信

（1）で制御された入力信号は、EQCON に送信され、EQCON の DSHPLC.DLL が有する入力または出力デバイスメモリに反映されます。（ただし、DSH-PLCSim と EQCON が接続状態である必要があります。）



（3）I/O 操作のログ表示

（1）で行なわれた操作は、ログ表示画面に随時表示されます。

表示と同時に、dshplc.log ファイルに記録保存されます。

３．２．３　装置コントローラからの I/O 要求応答機能

EQCON との通信確立の後、EQCON からの入出力要求に対し、DSH-PLCSim は自動的に応答します。

（1）入力デバイスの読出し要求

EQCON からは、入力デバイスの１点入力、データ単位（16 ビット）の入力に対し、DSH-PLCSim は、I/O デバイスメモリに保存されている情報を自動的に応答します。

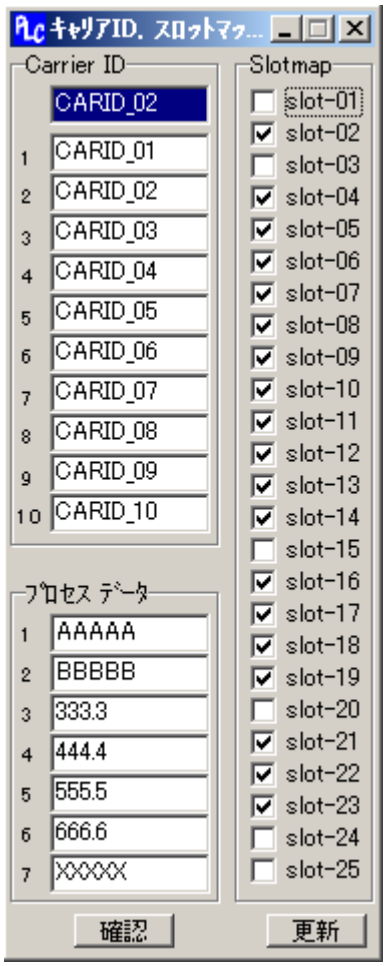
（2）出力デバイスに対する制御情報

EQCON から送信されてくる出力デバイスの１点出力、データ単位（16 ビット）の出力に対し、DSH-PLCSim は送られてきた情報を I/O デバイスメモリに反映させるとともに、出力画面のマップ表示にも反映させます。

３．３　キャリア ID、スロットマップ、プロセスデータ操作機能

本機能は、PLC とは直接関係がありませんが、以下の情報に対する設定と EQCON の読出し要求に対する応答機能があります。

1. キャリア ID の設定、選択と、EQCON への応答
2. スロットマップ（収納されているウェハーの有無）情報の設定と、EQCON への応答
3. プロセスデータの設定と、EQCON への応答

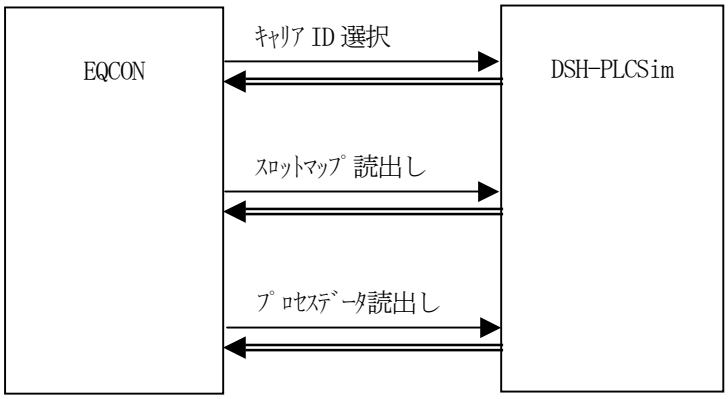


[ｷｬﾘｱ ID]
10 個分準備されており、
それらの中の１つを選択します。
ｷｰ入力でｷｬﾘｱ ID を変更できます。
選択はｼｰｹﾝｽﾌﾟﾛｸﾞﾗﾑでも可能です。

[Slotmap]
25 ｽﾛｯﾄ分準備されており、各ｽﾛｯﾄの在荷状態を変更できます。
在荷状態の変更はｼｰｹﾝｽﾌﾟﾛｸﾞﾗﾑでも可能です。

[ﾌﾟﾛｾｽﾃﾞｰﾀ]
7 個のﾃﾞｰﾀが準備されています。
ｷｰ入力でﾃﾞｰﾀを変更できます。
ﾌﾟﾛｾｽﾃﾞｰﾀの変更はｼｰｹﾝｽﾌﾟﾛｸﾞﾗﾑでも可能です。

EQCON との情報のやり取りは次のようになります。

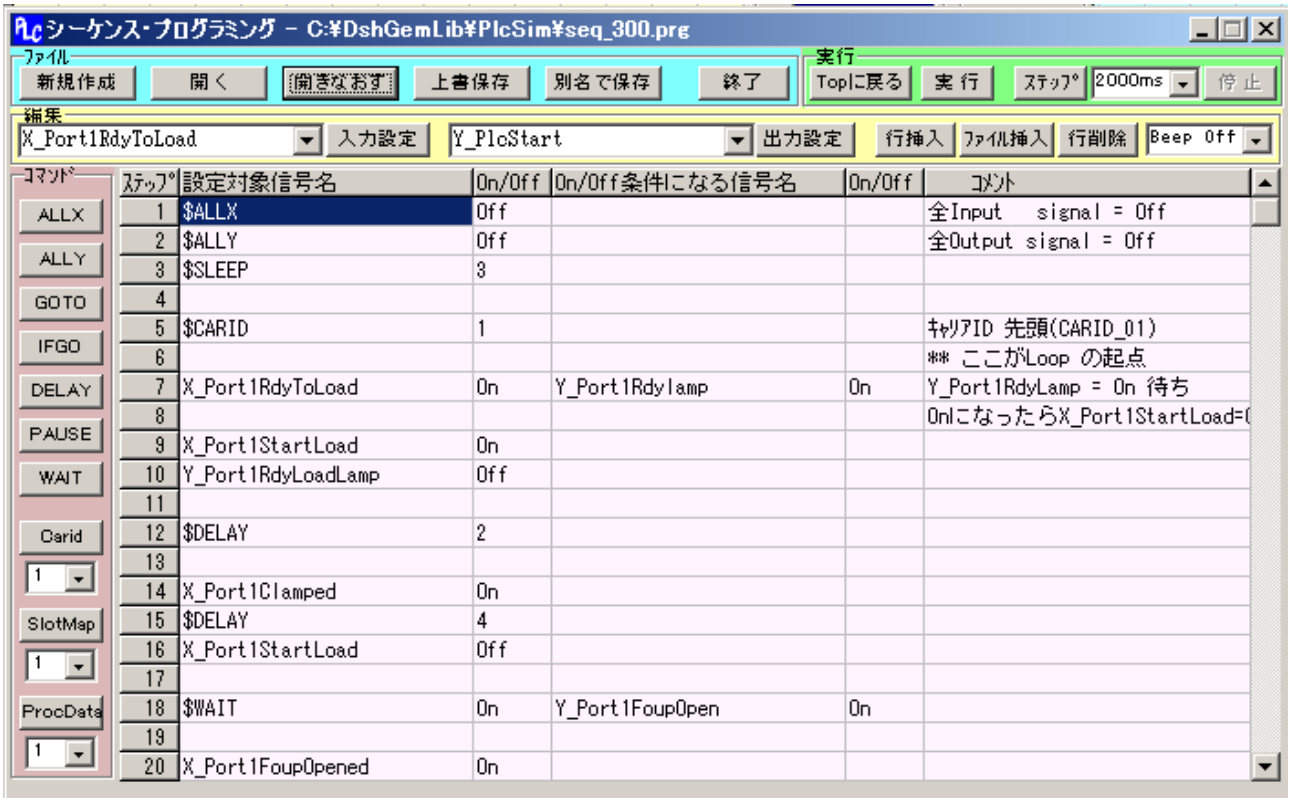


３．４　簡易シーケンスプログラム機能

ここまでは、I/O 制御画面上での操作について説明してきましたが、もう一つ、強力な機能としてシーケンスプログラムによる I/O の自動制御があります。

この機能は、画面に表示される表の中に、コマンドとパラメータを設定し、表の上から順にコマンド群から成るプログラムを実行するものです。実行ステップをビジュアルに確認しながら EQCON の機能をテストすることができます。

プログラム画面はつぎのような画面です。この画面上でプログラミングと実行の全ての操作ができます。



プログラムコマンドとして以下のものがあります。

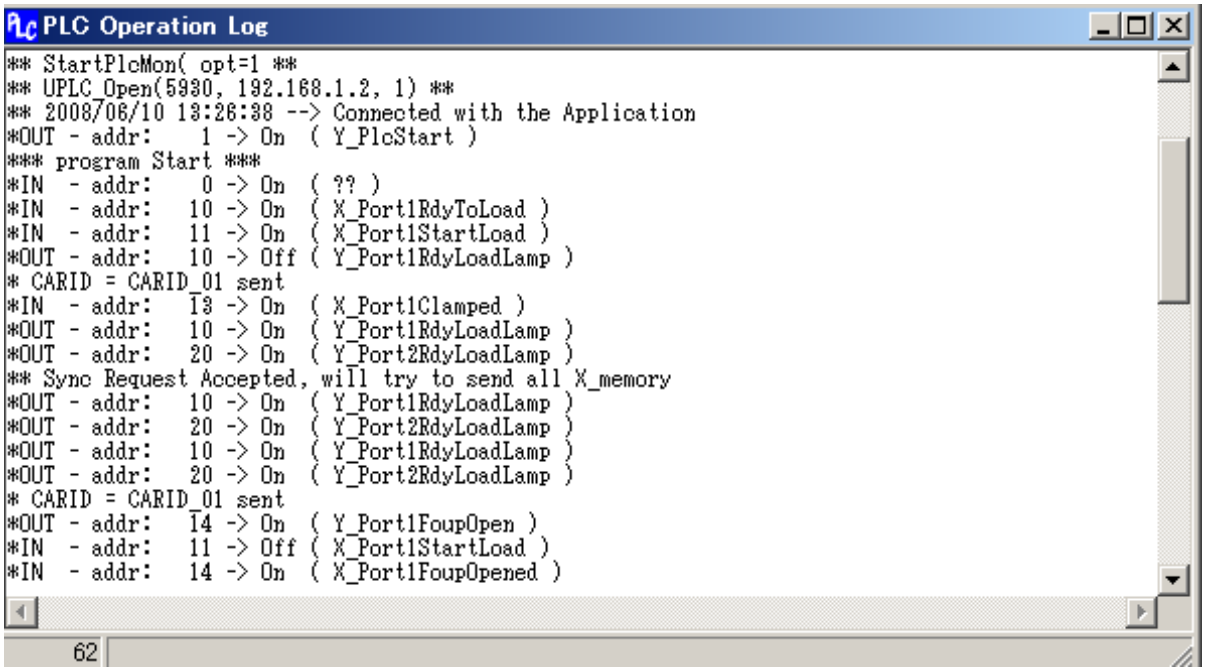
	ｺﾏﾝﾄﾞ	機能
1	X_signal On/Off	入力信号の On/Off、他の信号の On/Off の条件付き On/Off も可能です。
2	Y_SIgnal On/Off	出力信号の On/Off、他の信号の On/Off の条件付き On/Off も可能です。
3	$ALLX	入力信号を全て On/Off にします。
4	$ALLY	出力信号を全て On/Off にします。
5	$DELAY	秒単位で処理を遅らせます。
6	$GOTO	無条件に別のｽﾃｯﾌﾟにｼﾞｬﾝﾌﾟします。
7	$IFGO	信号の On/Off 条件によってｼﾞｬﾝﾌﾟします。
8	$WAIT	信号が On/Off になるのを待機します。
9	$PAUSE	実行を一旦停止します。
10	$CARID	ｷｬﾘｱ ID の選択をします。
11	$SLOTMAP	ｽﾛｯﾄﾏｯﾌﾟ情報を変更します。
12	$PROCDATA	ﾌﾟﾛｾｽﾃﾞｰﾀを変更します。

詳しくは、４章で説明します。

３．５　ログ表示機能

画面上で操作された、例えば入出力の On/Off 制御した内容は、ログ画面に随時表示されます。
また、EQCON から要求された要求内容も同様に表示されます。
そして、これらのログ情報は、ログファイル上にログメッセージにタイムスタンプ付きで記録されます。

［ログ画面］



［ログファイル］

```
08-06-10 13:24:26 ** StartPlcMon( opt=1 **
08-06-10 13:24:26 ** UPLC_Open(5930, 192.168.1.2, 1) **
08-06-10 13:26:38 ** 2008/06/10 13:26:38 --> Connected with the Application
08-06-10 13:26:39 *OUT - addr:    1 -> On  ( Y_PlcStart )
08-06-10 13:26:40 *** program Start ***
08-06-10 13:26:40 *IN  - addr:    0 -> On  ( ?? )
08-06-10 13:26:40 *IN  - addr:   10 -> On  ( X_Port1RdyToLoad )
08-06-10 13:26:40 *IN  - addr:   11 -> On  ( X_Port1StartLoad )
08-06-10 13:26:40 *OUT - addr:   10 -> Off ( Y_Port1RdyLoadLamp )
08-06-10 13:26:40 * CARID = CARID_01 sent
08-06-10 13:26:42 *IN  - addr:   13 -> On  ( X_Port1Clamped )
08-06-10 13:26:43 *OUT - addr:   10 -> On  ( Y_Port1RdyLoadLamp )
08-06-10 13:26:43 *OUT - addr:   20 -> On  ( Y_Port2RdyLoadLamp )
08-06-10 13:26:43 ** Sync Request Accepted, will try to send all X_memory
08-06-10 13:26:43 *OUT - addr:   10 -> On  ( Y_Port1RdyLoadLamp )
08-06-10 13:26:43 *OUT - addr:   20 -> On  ( Y_Port2RdyLoadLamp )
08-06-10 13:26:43 *OUT - addr:   10 -> On  ( Y_Port1RdyLoadLamp )
08-06-10 13:26:43 *OUT - addr:   20 -> On  ( Y_Port2RdyLoadLamp )
08-06-10 13:26:43 * CARID = CARID_01 sent
08-06-10 13:26:45 *OUT - addr:   14 -> On  ( Y_Port1FoupOpen )
08-06-10 13:26:47 *IN  - addr:   11 -> Off ( X_Port1StartLoad )
08-06-10 13:26:47 *IN  - addr:   14 -> On  ( X_Port1FoupOpened )
```

４．簡易シーケンスプログラムによる自動テスト機能

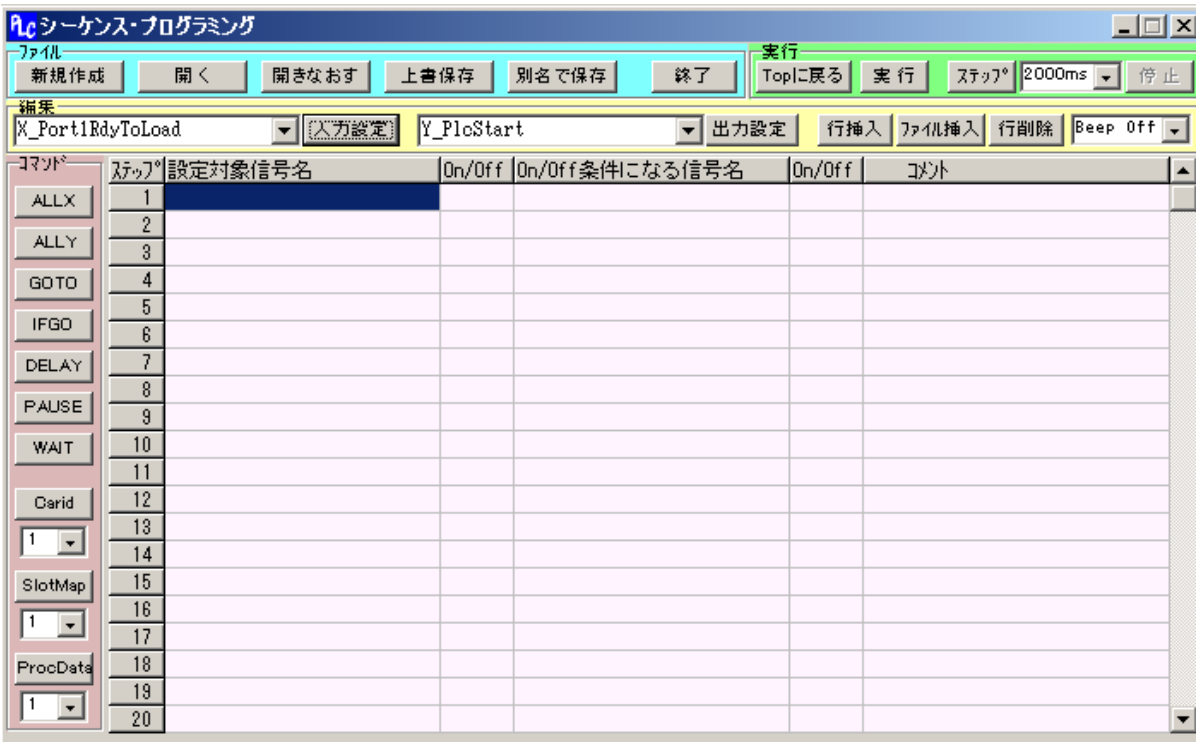
I/O 制御をプログラムのシーケンスに従って自動的にテスト実行させるための機能です。

シーケンスを実行するには、ある出力デバイスが ON になったらという条件で、ある入力デバイスを ON にするような判断した上で制御するというような論理で進める必要があります。また、実機に近い形でテストするには、時間的な遅延などを行なう必要もあります。

このような自動テスト機能を簡易なコマンド（命令）で簡単にプログラミングし実行することを可能にする機能です。キャリア ID, スロットマップ、プロセスデータに対するプログラミングもできます。

４．１　プログラミングと操作画面

画面上で、入出力選択用コンボボックス、ボタン、マウスとグリッドへのキー入力操作でプログラミングします。



（1）プログラム設定グリッド（表）画面

画面の表には、６個の列がありますが、それぞれの列の用途は次の通りです。

列	見出し	意　味
1	ｽﾃｯﾌﾟ	ﾌﾟﾛｸﾞﾗﾑの実行ｼｰｹﾝｽです。実行は、正順に進みます。
2	設定対象信号名	On/Off したい信号名または制御ｺﾏﾝﾄﾞを設定します。
3	On/Off	設定対象信号の On/Off または制御ｺﾏﾝﾄﾞのﾊﾟﾗﾒｰﾀを設定します。
4	On/Off 条件になる信号名	設定対象信号を On/Off にする際に他の信号の On/Off 条件になります。 また、制御ｺﾏﾝﾄﾞにﾊﾟﾗﾒｰﾀにも使用されます。
5	On/Off	条件になる信号の On/Off 条件を設定します。
6	ｺﾒﾝﾄ	注釈の欄です。ﾌﾟﾛｸﾞﾗﾑ実行上の意味はありません。

（2）入出力デバイス名選択コンボボックス

３．１で説明したI/Oデバイス定義ファイルに登録された入出力信号名が入力、出力それぞれにコンボボックスのリストに表示されます。

コンボボックスをプルダウンし、コマンドとして設定したい信号名を選択します。そして、入力設定または出力設定ボタンを使って所望のステップと列に信号名を設定します。

［設定例］

ｽﾃｯﾌﾟ	設定対象信号名	On/Off	On/Off 条件になる信号名	On/Off	
1					
2	X_Port1StartLoad	On	Y_Port1RdyToLoad	On	
3			Y_Port1RdyLoadLamp	On	
4					
5	$DELAY	5			
6	Y_Port1RdyToLoad	Off			

上の例は、Y_Port1RdyToLoad=On で、かつ、Y_Port1RdyLoadLamp=On の状態になったら X_Port1StartLoad＝On にし、ｽﾃｯﾌﾟ を５に進め、５秒間遅延の後、ｽﾃｯﾌﾟ -6 に進みます。
因みに、通常、条件になる出力信号の On は EQCON が制御します。
（ﾌﾞﾗﾝｸのｽﾃｯﾌﾟ は無処理（no operation）となり、次のｽﾃｯﾌﾟ に進みます。）

（3）制御コマンドボタン

制御コマンドの設定は制御ボタンを使って行ないます。
コマンド名は、設定対象信号名の列に設定されます。
そして、コマンドのパラメータはコマンドによって所定の列に設定することになります。
コマンド名は、大文字で、ボタンの表示名の頭に **$** を付けたものになります。

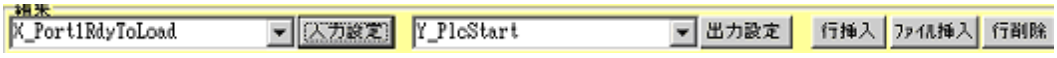
設定対象信号名	On/Off	On/Off 条件になる信号名	On/Off	機 能
$ALLX	On/Off			全入力ﾃﾞﾊﾞｲｽを On/Off する。
$ALLY	On/Off			全出力ﾃﾞﾊﾞｲｽを On/Off する。
$DELAY	<t>			<t>秒間遅延します。
$GOTO	<step>			<step>位置に無条件ｼﾞｬﾝﾌﾟ します。
$IFGO	<step>	<I/O 信号名>	On/Off	<I/O 信号名>の On/Off 条件で<step>ｼﾞｬﾝﾌﾟ します
$WAIT		<I/O 信号名>	On/Off	<I/O 信号名>の On/Off 条件になるまで待機します。
$PAUSE				一旦停止します。
$CARID	<index>			<index>(1～10)のｷｬﾘｱ ID を選択します。
$SLOTMAP	<index>		On/Off	<index>のｽﾛｯﾄを On/Off します。
$PROCDATA	<index>	<ﾃﾞｰﾀ値>		<index>(1～7)の値を<ﾃﾞｰﾀ値>にｾｯﾄします。

（注） <ﾃﾞｰﾀ値>は文字列ﾃﾞｰﾀです。

４．２　編集と保存

（１）編集

編集ボタンブロックを使って操作します。



４．１のプログラミング操作の中で、コマンドの挿入、削除などの編集作業が必要になります。

ステップ行の挿入は、行挿入ボタン、ステップ行の削除は、行削除ボタンのクリックで行ないます。行挿入ではブランク行が挿入されます。また、行削除では、そのとき選択されているステップ行を削除し、後ろのコマンド群を前方向にプルアップします。

この際、＄GOTO,　$IFGO コマンドのジャンプ先は、挿入、削除によって変わる可能性がありますが、そのジャンプ先の調整も自動的に行われます。

既に作成されているファイルを行の間に挿入したい場合は、ﾌｧｲﾙ挿入ボタンを使用します。この場合の挿入によって影響を受ける$Go, $IFGO のジャンプ先ステップの調整も自動的に行なわれます。

（２）保存形式と保存

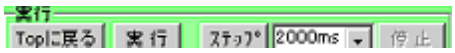
ファイルボタンブロックのボタンを使用します。

画面上にプログラミングされたプログラムをファイルに保存することができます。また、保存したプログラムを再度開いて使用することができます。
プログラムファイルは、拡張子.prg で保存されます。
形式は、プログラム画面の表に表示されているステップ順に、2 列目から 6 列目までのセルの内容をカンマ切りされて保存されます。
下に、サンプルを示します。

```
$ALLX,                  Off, ,                  ,    全Input   signal = Off,
$ALLY,                  Off, ,                  ,    全Output signal = Off,
$SLEEP,                 3,   ,                  ,    ,
,                       ,    ,                  ,    ,
$CARID,                 1,   ,                  ,    ｷｬﾘｱ ID 先頭(CARID_01),
,                       ,    ,                  ,    ** ここが Loop の起点,
X_Port1RdyToLoad,       On,  Y_Port1Rdylamp,    On,  Y_Port1RdyLamp = On 待ち,
,                       ,    ,                  ,    On になったら X_Port1StartLoad=On,
X_Port1StartLoad,       On,  ,                  ,    ,
Y_Port1RdyLoadLamp,     Off, ,                  ,    ,
,                       ,    ,                  ,    ,
$DELAY,                 2,   ,                  ,    ,
,                       ,    ,                  ,    ,
X_Port1Clamped,         On,  ,                  ,    ,
$DELAY,                 4,   ,                  ,    ,
X_Port1StartLoad,       Off, ,                  ,    ,
,                       ,    ,                  ,    ,
$WAIT,                  On,  Y_Port1FoupOpen,   On,  ,
,                       ,    ,                  ,    ,
X_Port1FoupOpened,      On,  ,                  ,    ,
Y_Port1FoupOpen,        Off, ,                  ,    ,
```

４．３　シーケンスプログラムの実行と停止

実行はボタンブロックのボタンのクリック行ないます。

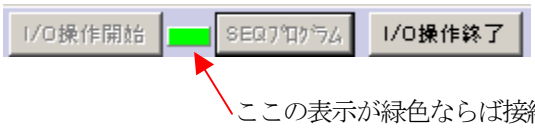


（1）実行

実行開始は、画面上の開始したいステップを選択し、実行ボタンのクリックで開始します。
実行中は、コマンドとコマンドの実行にインターバルをとることができます。ｽﾃｯﾌﾟボタンの右横にあるインターバル（単位はms）の値を選択できます。このことによって、実行状態を目視しながらステップを確認しテストすることができます。

ｽﾃｯﾌﾟボタンを使うとコマンドを１個ずつステップ順に実行できます。

EQCONとの通信が必要なコマンドの実行の条件として、DSH-PLCSimとEQCONとの間でTCP/IP通信接続が確立していることが必要です。接続しているかどうかの確認は、メニューバーの下のランプで確認できます。



ここの表示が緑色ならば接続中です。

（2）停止

プログラムの実行停止は、一旦停止も含め、次のことによって停止します。
①プログラムが終端に達したとき。
②停止ボタンがクリックされたとき。
③$PAUSEコマンドに達したとき。