

SW-PLC ライブラリ

ユーザーズマニュアル

株式会社セイワ技研
2017 年 12 月 15 日

目次

概要.....	2
特徴.....	2
スタートガイド.....	3
ダミーPLC 機能.....	6
手順.....	6
PLC 接続機能.....	7
手順.....	7
スクリプト.....	8
スクリプト例.....	8
OnInit イベント.....	9
OnExit イベント.....	9
OnTimer イベント.....	9
OnDataChange イベント.....	9
動作の仕組み.....	10
ライブラリ.....	10
タグデータメモリ.....	10
コマンド実行.....	12
通信の効率化.....	14
ダミーPLC.....	14
SW-TagManager.....	15
動作環境.....	15
メイン画面.....	16
ファイルメニュー.....	16
ツールメニュー.....	18
コマンドボタン.....	19
タグリスト.....	20
右クリックメニュー.....	23
ログメッセージ出力.....	24
通信設定画面.....	25
ダミーPLC 動作設定.....	28
PLC 接続設定.....	28
コマンドライン引数.....	29

概要

SW-PLC ライブラリは、各社 PLC の通信プロトコルを実行するライブラリです。通信機能を提供するライブラリ DLL(SwPlcCom.dll, アドイン DLL)と、PLC のレジスタ割付を管理するツール SW-Tag Manager で構成されています。以降に、本ライブラリの使い方について説明します。

特徴

- 通信ライブラリ

各 PLC プロトコルに対応した通信アドイン DLL を提供します。接続先 PLC の機種やメーカーが変わってもアドイン DLL と設定ファイルを入れ替えるだけでプログラムは変更せずに対応できます。

- PLC レジスタをタグで管理

PLC のレジスタにタグ ID を割り付けて管理します。ユーザープログラムからはタグ ID にて PLC のレジスタにアクセスするため、アドレスを意識することなく操作できます。

- ダミーPLC 機能

SW-TagManager をダミーPLC として動作させることができます。本ライブラリを使用して作成されたユーザープログラムからこのダミーPLC に通信接続することで、PLC の実機が無くてもユーザープログラムの動作確認が可能です。

- PLC 接続機能

SW-TagManager から PLC へ直接接続することができます。PLC との接続確認、データの確認を行ったりすることができます。

- スクリプト機能

ダミーPLC や PLC 接続時の動作をスクリプトでカスタマイズすることができます。スクリプトは C#で記述できるため、様々な応用が可能です。ダミーPLC で PLC の動作をシミュレートしてユーザープログラムをデバッグしたり、PLC 接続時に PLC からデータを読み取ってファイルにロギングするなどが可能です。

スタートガイド

基本的な使い方について紹介します。ここでは C# のプログラムを例にしています。他の言語のプログラムについては別途サンプルプログラムファイルを参照してください。

1. タグ定義ファイルの作成

SW-TagManager を開いて、新規作成から新しいタグ定義シートを作成し、タグの追加を実行します。

2. 通信設定

接続設定から接続設定を行います。ここではダミー PLC へ接続するための設定を紹介します。ダミー PLC との通信はダミー PLC 専用プロトコル(SwCom)を使用します。

接続設定

ライブラリ名: SwCom

通信タイプ: TCP 接続

3. タグ定義ファイルの保存

タグ定義ファイルの保存 タグの割付が完了したら、ファイルに保存を実行してタグ定義ファイルを保存します。ここでは sample.xml として保存します。

4. プログラムの準備

C# で PLC ライブラリを使用する場合、下記の cs ファイルをプロジェクトに追加します。また、ライブラリ DLL を作成するプログラムから参照できる場所に置きます。通常は作成される exe と同じ場所に置きます。

- SwPlcComSdk.cs
- SwPlcCom.dll
- 接続する PLC に対応したアドイン DLL (ここではダミー PLC 用 SwCom.dll を使用します)

5. プログラムの作成

プログラム Visual Studio にて C#プロジェクトを作成し、下記コードを入力します。下記例は Windows フォームプロジェクトを作成し、フォーム上に配置したボタンイベント内でコードを実行しています。

```
using System;
using System.Windows.Forms;
using System.Diagnostics;
using SwPlcComSdk;

namespace Sample
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SwPlcCode result;
            int handle;
            // タグ定義ファイルを読み込んで通信ポートを開く
            result = SwPlcCom.SwPlcLoadFile(out handle, "sample.xml");
            if (result != SwPlcCode.SWPLC_OK)
            {
                return;
            }

            try
            {
                // Tag0～2 を読み込むコマンドを登録
                SwPlcCom.SwPlcRead(handle, 0);
                SwPlcCom.SwPlcRead(handle, 1);
                SwPlcCom.SwPlcRead(handle, 2);

                // Tag3～5 に書き込むコマンドを登録
                SwPlcCom.SwPlcWriteUShort(handle, 3, 30);
                SwPlcCom.SwPlcWriteUShort(handle, 4, 40);
                SwPlcCom.SwPlcWriteUShort(handle, 5, 50);
            }
        }
    }
}
```

```

// 登録されたコマンドを実行する
result = SwPlcCom.SwPlcExecute(handle);
if (result != SwPlcCode.SWPLC_OK)
{
    return;
}

// コマンド実行の応答を取得
// SwPlcResponse の 2 番目の引数に-1 を指定した場合、
// 全てのコマンド実行が完了するまで待機します
result = SwPlcCom.SwPlcResponse(handle, -1);
if (result == SwPlcCode.SWPLC_RESPONSE_DONE)
{
    // PLC から読み込んだデータを取得します
    ushort tag1 = 0;
    ushort tag2 = 0;
    ushort tag3 = 0;
    SwPlcCom.SwPlcGetUShort(handle, 0, ref tag1);
    SwPlcCom.SwPlcGetUShort(handle, 1, ref tag2);
    SwPlcCom.SwPlcGetUShort(handle, 2, ref tag3);
    Debug.Print("Tag0={0}, Tag1={1}, Tag2={2}", tag1, tag2, tag3);
}
}
finally
{
    SwPlcCom.SwPlcClosePort(handle);
}
}
}
}

```

6. プログラムの実行

SW-TagManager でダミーPLC を起動し、Tag0～2 へ適当な値を入力します。次に C# で作成したプログラムを実行します。上記プログラムでフォームのボタンを押すと、Tag0～2 から読み込んだデータがデバッグ出力され、Tag3～5 へ値が書き込まれるのを確認します。

7. 実機への接続

PLC 実機へ接続するには、タグマネージャの通信設定にて接続する PLC のプロトコルに対応したアドインライブラリを選択します。例えば MC プロトコル(三菱 PLC 用プロトコル)で接続する場合、McCom を選択したあとオプションパラメータの設定を行います。あとは、作成したタグ定義ファイルを上記プログラムで使用したファイルと入れ替えるだけで完了です。

ダミーPLC 機能

SW-TagManager をダミーPLC として動作させることができます。本ライブラリを使用して作成されたユーザープログラムからこのダミーPLC に通信接続することで、PLC の実機が無くてもユーザープログラムの動作確認が可能です。

ダミーPLC との通信は、ダミーPLC 通信専用プロトコルの SwCom で通信します。TCP 接続のみ可能です。ダミーPLC と通信するには、クライアント側で TCP 接続と通信アドインライブラリの SwCom を選択して接続してください。

手順

1. タグリストを作成します。
2. 設定画面のダミーPLC 動作設定で待ち受けポート番号の設定を行います。
3. コマンドボタンを PLC 接続ボタンを押下します。
4. 設定された通信条件で PLC へ接続されます。
5. PLC へ接続後は PLC から値を読み込んだり、値の書き込みができます。

PLC 接続機能

SW-TagManager から PLC へ直接接続することができます。PLC との接続確認、データ読み書きに利用することができます。

手順

1. タグリストを作成します。
2. 設定画面で通信設定を行います。
3. コマンドボタンを PLC 接続ボタンを押下します。
4. 設定された通信条件で PLC へ接続されます。
5. PLC へ接続後は PLC から値を読み込んだり、値の書き込みができます。

スクリプト

スクリプトを作成することで、値の自動更新やファイルへの保存が可能となります。スクリプトは C# で作成します。例えば、5 秒毎に値を更新したり、あるタグの値が変化したときに他のタグに値をセットするなどが可能です。また、C# の基本的な機能が使用できるため、定周期で PLC から値を読み取りファイルに保存するなどデータロガーのような使い方もできます。

スクリプト例

```
using SwTagManager.Scripting;
using System;
using System.IO;
using System.Text;

class PlcScript : PlcScriptBase
{
    void OnInit()
    {
        // タイマーを作成
        // 3 秒毎に OnTimer イベントが発生する
        SetTimer(1, 3000);
    }

    void OnExit()
    {
        // タイマーを破棄する
        KillTimer(1);
    }

    void OnTimer(int timerId)
    {
        if (timerId == 1)
        {
            // 3 秒毎に Tag1 のデータを読み取り
            var data = new PlcData[TagId = 1];
            if (PlcRead(data))
            {
                OutputLog("Tag1 = {0}", (ushort)data.Value);
            }
        }
    }

    void OnDataChange(int tagId, object value)
```

```

{
    // Tag2 が 1 に変化したとき
    if (tagId == 2 && (ushort)value == 1)
    {
        var data = new PlcData { TagId = 10, Value = (ushort)1};
        PlcWrite(data);
    }
}
}

```

OnInit イベント

OnInit はスクリプト起動時に1回呼び出されるイベントです。ここで各種変数の初期化などを行います。例では 3 秒のタイマーを作成しています。

OnExit イベント

OnExit はスクリプトが終了されたときに呼び出されるイベントです。後処理などを記述します。例ではタイマーを破棄しています。

OnTimer イベント

OnTimer は SetTimer で作成されたタイマーから、設定された時間が経過するたびに呼び出されるイベントです。例では 3 秒毎に Tag1 のデータを読み込んでログメッセージに出力しています。

OnDataChange イベント

OnDataChange は値が変化した時に呼び出されるイベントです。タグ定義にて通知 ON に設定されたタグが対象となります。例では、Tag2 の値が 1 に変化したとき、Tag10 に 1 を書き込んでいます。

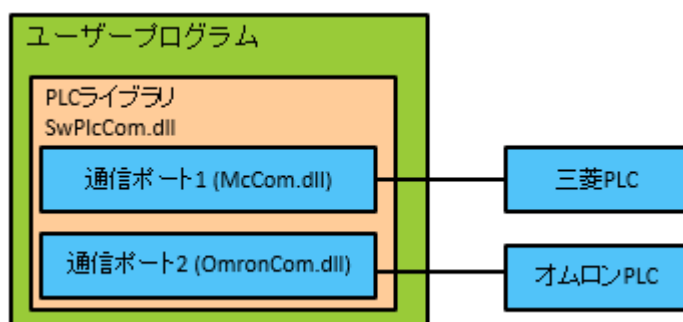
その他詳細についてはスクリプトリファレンスを参照してください。

動作の仕組み

ライブラリをプログラムに組み込む際に、ライブラリの動作について理解するとより効率の良いプログラムができます。以降にライブラリの動作について説明します。

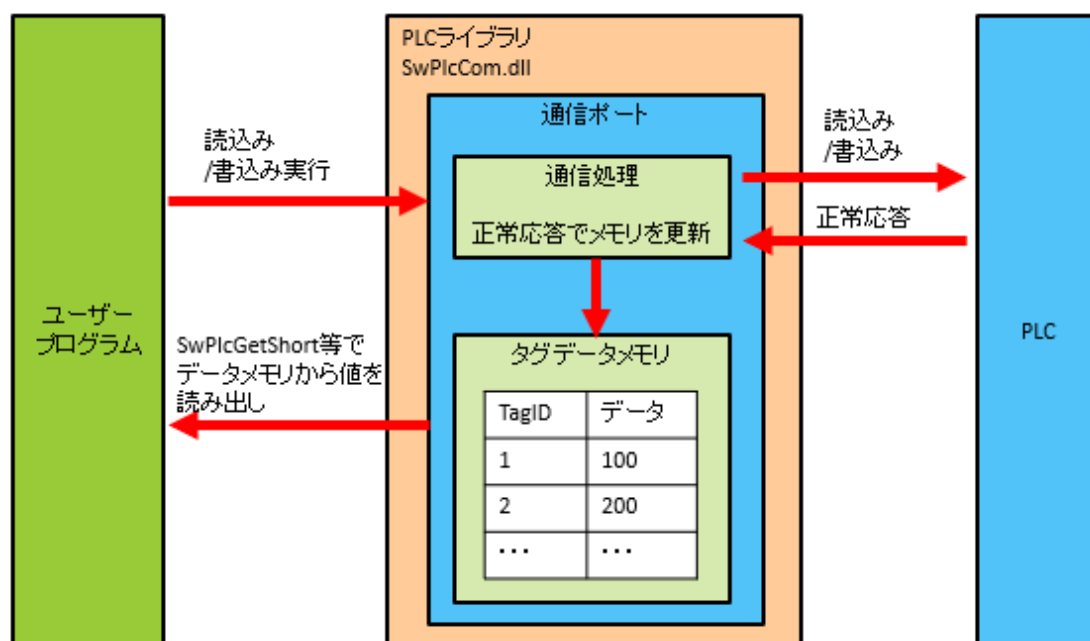
ライブラリ

下記にライブラリの構成を示します。



タグデータメモリ

通信ポートが作成された後は、各通信ポートは独立したスレッドで動作します。各スレッド内では定義されたタグデータを保持する共有メモリ領域が確保されます。Read/Write コマンドが実行され、通信が正常に完了したときにこの共有メモリのデータが更新されます。下記に動作概念図を示します。



シリアル通信

シリアル通信を指定して通信ポートを作成するとシリアルポートが開かれます。シリアルポートは SwClosePort が実行されるまでオープン状態となり、SwPlcComStatus の戻り値は常に SWPLC_PORT_OPEN となります。SwPlcClosePort を実行するとシリアルポートが閉じられます。

UDP 通信

UDP 通信を指定して通信ポートを作成するとライブラリ内で UDP ポートが開かれます。UDP ポートは SwClosePort が実行されるまでオープン状態となり、SwPlcComStatus の戻り値は常に SWPLC_PORT_OPEN となります。SwPlcClosePort を実行すると UDP ポートが閉じられます。

TCP 通信

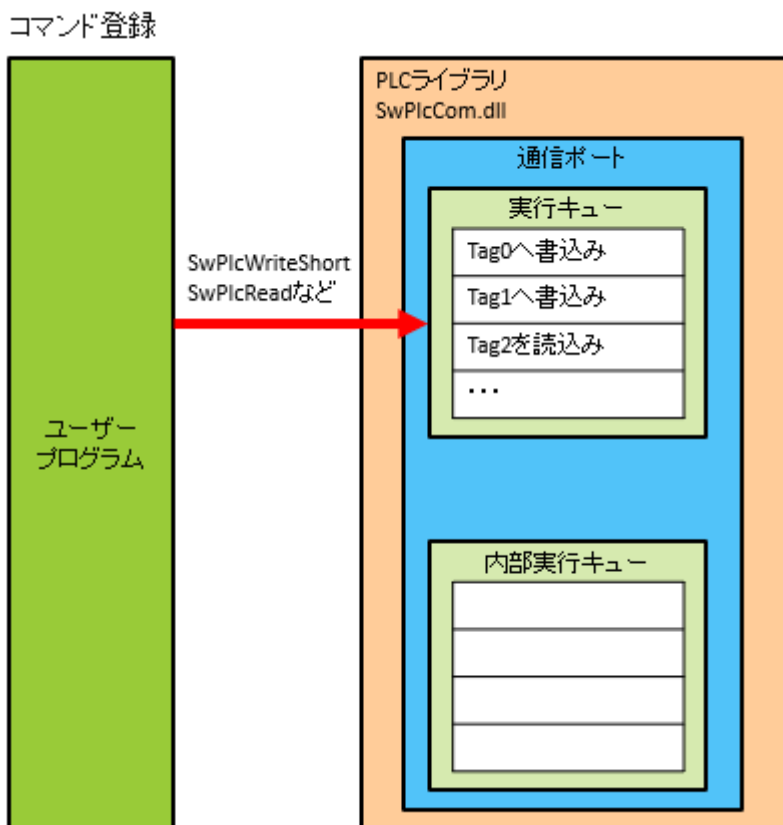
TCP 通信を指定して通信ポートを作成するとライブラリ内で TCP ポートが開かれます。TCP ポートが開かれた後、ライブラリは指定された接続先に TCP 接続を試行します。TCP 接続に成功すると状態は SWPLC_PORT_CONNECTED となります。TCP 接続が切断されると状態は SWPLC_PORT_CLOSE となります。TCP 接続が切断された後は、ライブラリ内で再び TCP 接続が試行されます。TCP 接続試行時の接続待ち時間と試行する時間間隔は通信パラメータで設定します。SwPlcClosePort を実行すると TCP 接続を切断し TCP ポートが閉じられます。

リトライ処理

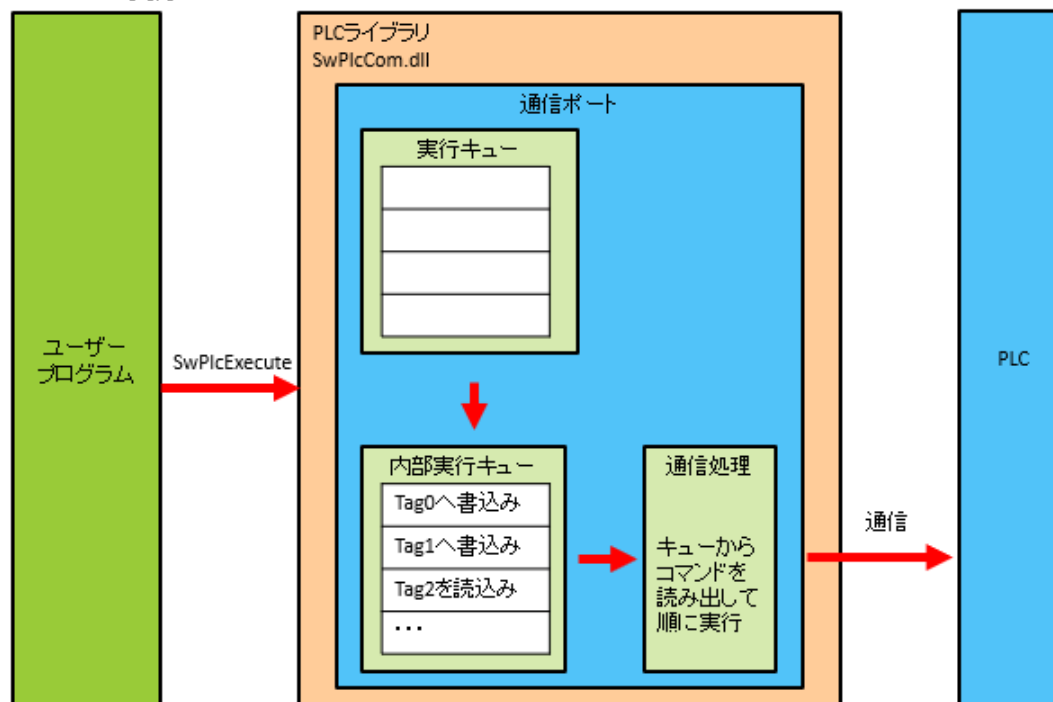
通信コマンドを送信した後、設定されたタイムアウト時間を経過しても PLC から応答がない場合、設定された回数だけ通信処理のリトライが実行されます。通信をリトライしても PLC から応答が無い場合、コマンドを実行した対象のタグ状態は、SWPLC_TAG_STATUS_ERROR となります。タグの状態は SwPlcGetShort などのデータ取得関数の戻り値で確認できます。

コマンド実行

コマンド実行時の動作について説明します。読み書きのコマンドは一旦実行キューに登録され、SwPlcExecuteにより実行されます。SwPlcExecuteが実行されたあとは、登録したコマンドは全て内部実行キューに移動され、実行キューはクリアされます。実行キューがクリアされた後は、次のコマンドの登録が可能となります。但し、次にコマンドの実行は前回実行したコマンドが全て終了した後で実行可能となります。下記にコマンド実行の概念図を示します。

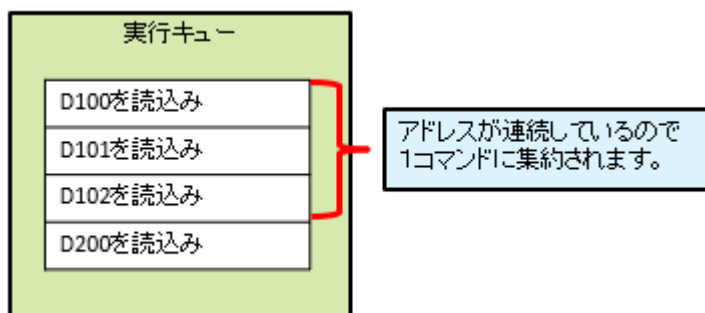


SwPlcExecute実行



通信の効率化

実行キューにコマンドを登録するときは、PLC レジスタのアドレスが連続するように登録すると効率よく通信が実行されます。下記例のようなコマンドを実行した場合、D100～D102 の読み込みは 1 コマンドに集約されて実行されます。1 コマンドに集約される最大の個数は各 PLC によって異なります。



ダミーPLC

ダミーPLC は SwCom という専用プロトコルを使用して通信が行われます。ダミーPLC は TCP 接続のみ可能です。ダミーPLC を起動すると専用の TCP サーバーが起動され待ち受け状態となります。クライアントプログラムは通信ライブラリで SwCom と TCP 接続を設定して接続します。ダミーPLC とは 1:1 の接続のみ可能で、複数のクライアントプログラムを 1 つのダミーPLC で同時に受け付けることはできません。

SW-TagManager

SW-TagManager の各機能について説明します。

動作環境

サポート OS

- * Windows 7 (32bit/64bit)

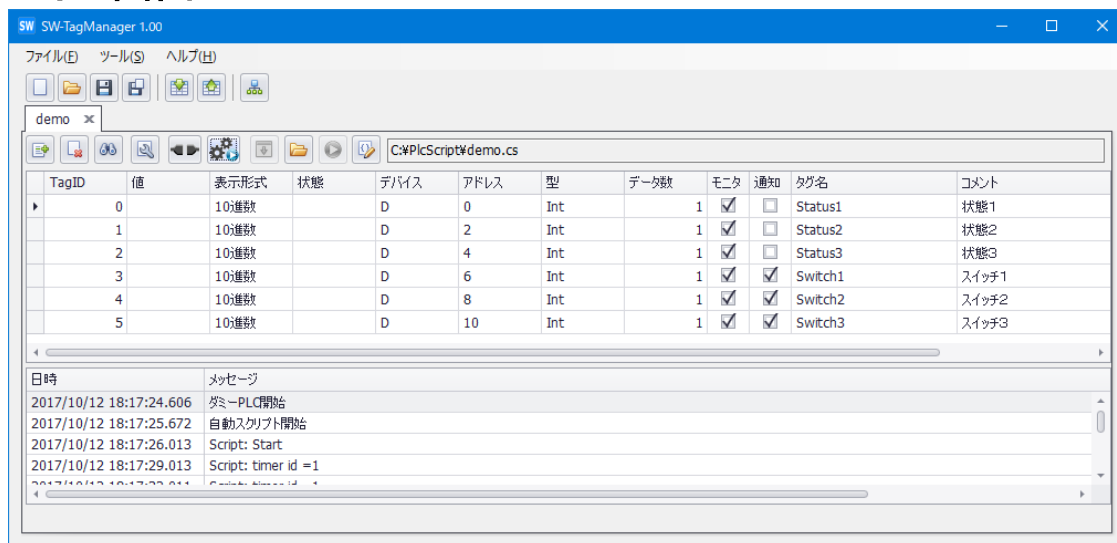
- * Windows 8 (32bit/64bit)

- * Windows 8.1 (32bit/64bit)

- * Windows 10 (32bit/64bit)

*その他 ST-TagManager を実行するには、.NET Framework 4.5 以上のバージョンをインストールする必要があります。

メイン画面



ファイルメニュー

[新規作成]

タグ定義が新規で作成されます。

[開く]

タグ定義ファイルを読み込んで表示します。

[上書き保存]

選択されているタブのタグ定義をファイルに上書き保存します。

[名前を付けて保存]

選択されているタブのタグ定義をファイル名を指定して保存します。

[CSV インポート]

CSV 形式のタグリストファイルを読み込んで表示します。

CSV ファイルは下記の項目を順番にカンマ区切りで出力したファイルです。

No.	項目	説明
1	タグ ID	0 以上の整数
2	表示形式	10 または 16
3	デバイス	デバイス名文字列

4	アドレス	10 進表記または 16 進表記の数値(16 進表記の場合は末尾に H を付加 例: 1FH)
5	型	データ型(Short/UShort/Int/UInt/Long/ULong/Float/Double/Bit/Bit2)
6	データ数	1 以上の整数
7	モニタ	0: モニタしない 1: モニタする
8	通知	値が変化した時の通知 0: しない 1: する
9	タグ名	タグ名文字列
10	コメント	コメント文字列

CSV 例

```
0,10,D,100,Short,1,1,0,Status1,状態 1
1,10,D,101,Short,1,1,0,Status2,状態 2
2,10,D,102,Short,1,1,0,Status3,状態 3
3,10,D,200,Short,1,1,1,Switch1,スイッチ 1
4,10,D,201,Short,1,1,1,Switch2,スイッチ 2
5,10,D,202,Short,1,1,1,Switch3,スイッチ 3
```

[CSV エクスポート]

選択されたタブのタグリストを CSV 形式でファイルに出力します。

[通信中のポートヘアタッチ]

他のプログラムで通信中のポートヘアタッチし、タグデータメモリの状態を画面に表示します。通信中のポートと同じタグ定義ファイルを選択し、アタッチ実行ボタンを押下するとタグデータがリアルタイムに表示されます。PLC と通信実行中のプログラムで、PLC とのデータの読み書きの状態を確認したいなどプログラムのデバッグに利用できます。

[終了]

タブをすべて閉じて SW-TagManager を終了します。

ツールメニュー

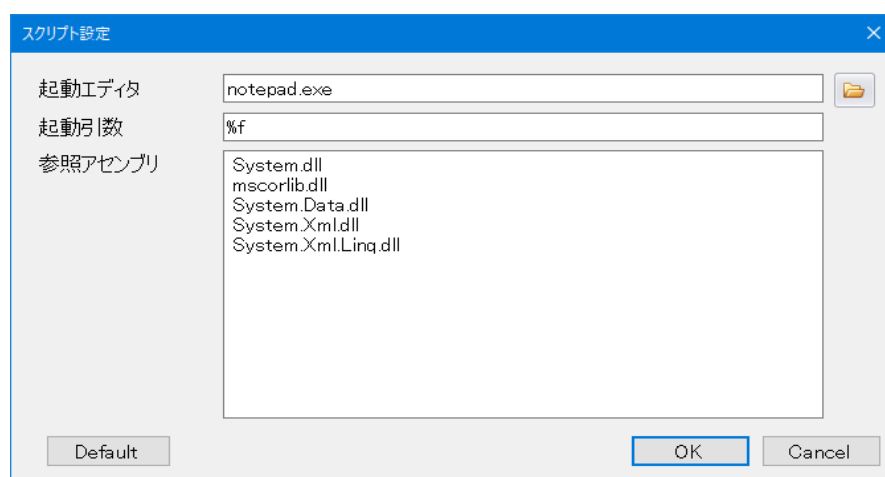
[スクリプトテンプレート作成]

スクリプトのテンプレートを指定したファイルに作成します。スクリプトを作成する際は、このテンプレートファイルを元に作成してください。

[Enum テンプレート作成]

各プログラム言語(C++/C#/VB)に対応した Enum 定義をクリップボードに出力します。タグ ID は数値のため、プログラム内でそのまま使用するには数が多くなると管理が大変です。タグ ID を定数定義してプログラム内で使用するのが基本ですが、本機能を利用すれば定数定義を作成する手間が省けます。Enum 定義で出力される定数の名前はタグ名に入力した名前が出力されます。

[スクリプト設定]



- 起動エディタ

スクリプト編集で起動するエディタプログラムのパスを設定します。

- 起動引数

外部エディタを起動する時に付加する引数です。%f はスクリプトファイルパスに変換されます。

- 参照アセンブリ

スクリプトで使用するアセンブリを設定します。アセンブリを追加する場合は、1行ずつアセンブリファイル名を入力してください。

コマンドボタン



[Tag 追加]

Tag 追加画面が表示されます。

[行削除]

タグリストで選択された行を削除します。

[検索]

タグリスト内の文字列を検索できます。

通信設定

通信設定画面が表示されます。

[PLC 接続]

PLC への通信接続が開始されます。通信設定にて設定された条件で接続されます。

[ダミーPLC 起動]

現在のタグ定義でダミーPLC を起動します。ダミーPLC を起動すると TCP サーバーが起動し、クライアントからの接続待ち状態となります。ダミーPLC は通信設定-ダミーPLC 動作設定で設定された TCP ポートで待ち受け状態となります。クライアントからダミーPLC へ接続するには通信方式を TCP に設定し、通信プロトコルで SwCom を使用してください。

[全データを PLC から読み込み]

PLC 接続時に PLC からタグ定義された全てのデータを読み込みます。

[スクリプトファイル選択]

スクリプトファイルを選択します。

[スクリプト実行]

選択したスクリプトを実行します。

[スクリプト編集]

選択したスクリプトをエディタで開きます。

[アタッチ実行]

通信中ポートへのアタッチを実行します。[通信中のポートへアタッチ]からタグ定義ファイルを選択したときに、このコマンドボタンが表示されます。

タグリスト

TagID	値	表示形式	状態	デバイス	アドレス	型	データ数	モニタ	通知	タグ名	コメント
▶ 0		10進数		D	0	Int	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Status1	状態1
1		10進数		D	2	Int	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Status2	状態2
2		10進数		D	4	Int	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Status3	状態3
3		10進数		D	6	Int	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Switch1	スイッチ1
4		10進数		D	8	Int	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Switch2	スイッチ2
5		10進数		D	10	Int	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Switch3	スイッチ3

● TagID

タグ ID を数値で入力します。

● 値

PLC 接続、ダミーPLC 起動、アタッチを実行したときに値が表示されます。PLC 接続時とダミーPLC 起動時は値の書き込みも可能です。値を書込むには、値を直接入力して Enter キーまたはセルフフォーカスの移動で確定します。また、右クリックメニューの[選択項目を変更する]から一括書き込みも可能です。PLC 接続時はモニタありに設定されたタグは自動で PLC から値が読み取られて表示が更新されます。モニタする周期は通信設定-PLC 接続設定のモニタ通信間隔で設定します。

● 表示形式

値の表示形式を選択します。10 進数または 16 進数表示を選択します。

● 状態

値の状態が表示されます。OK 表示の時は正常に通信が実行されている状態です。Error 表示の時は正常に通信出来ていない状態を示します。None 表示は 1 度も通信が実行されていない状態を示します。

● デバイス

デバイス名を設定します。デバイス名は半角英数字で 31 文字以内で設定します。

● アドレス

アドレスを設定します。アドレスの末尾に H を付けた場合は 16 進数として、O(オー)を付けた場合は 8 進数として認識されます。

100 --> 100, 10H --> 16, 10O --> 8

- 型

値のデータ型を選択します。下記のデータ型から選択してください。

データ型	説明
Short	ワードデバイスを符号付き 16 ビット整数として読み書きします。
UShort	ワードデバイスを符号無し 16 ビット整数として読み書きします。
Int	2 ワードデバイスを符号付き 32 ビット整数として読み書きします。2 ワードデバイスは連続した 2 つのデータレジスタで構成されます。
UInt	2 ワードデバイスを符号無し 32 ビット整数として読み書きします。2 ワードデバイスは連続した 2 つのデータレジスタで構成されます。
Long	4 ワードデバイスを符号無し 64 ビット整数として読み書きします。4 ワードデバイスは連続した 4 つのデータレジスタで構成されます。
ULong	4 ワードデバイスを符号無し 64 ビット整数として読み書きします。4 ワードデバイスは連続した 4 つのデータレジスタで構成されます。
Float	2 ワードデバイスを 32 ビット浮動小数点数として読み書きします。2 ワードデバイスは連続した 2 つのデータレジスタで構成されます。
Double	4 ワードデバイスを 64 ビット浮動小数点数として読み書きします。4 ワードデバイスは連続した 4 つのデータレジスタで構成されます。
Bit	ビットデバイスを指定するときに使用するデータ型です。
Bit2	ビットデバイスを指定するときに使用するデータ型です。アドレス+ビット位置でビットデバイスを指定する PLC で使用します。ビットデバイスの指定方法は各アドインのヘルプを参照してください。Bit2 を選択した時は、アドレスは 10 進数で入力する必要があります。ビットデバイスのアドレスは下 2 桁が 00～15 のビット位置、2 桁目以降がアドレス番号を示します。例: 10015 の場合、アドレス 100, ビット位置 15 となります。

- データ数

データ数を設定します。1 以上の数値を設定してください。1 の場合は通常の値型のタグとなります。2 以上が設定された場合、配列型タグとなります。配列型タグをプログラムから操作する時は、SwPlcWriteShortArray や SwPlcGetShortArray などの配列操作関数の関数を使用してください。

- モニタ

チェックボックスを On にするとタグがモニタ登録されます。モニタ登録されたタグは通信ポートが開かれた後、定周期で PLC から値が読み込まれます。大量のタグをモニタ登録すると負

荷が増え、定周期で読み込みできなくなりますので常時モニタが必要なタグのみ登録してください。

- 通知

チェックボックスを On にするとタグの値が変化した時に通知されます。通常はモニタ登録と合わせて On に設定すると自動で値が監視できて便利です。通知は SwPlcSetNotify で設定されたコールバック関数に通知されます。スクリプト利用時は、OnDataChange イベントにて通知されます。

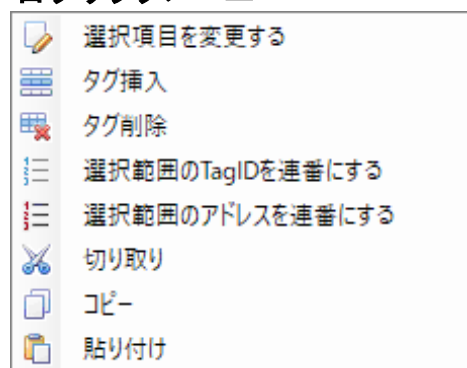
- タグ名

タグ名を設定します。Enum テンプレート出力で使用されます。(省略可)

- コメント

タグのコメントを設定します。(省略可)

右クリックメニュー



タグリスト上で右クリックしたときに表示されるメニューです。

- 選択項目を変更する
タグリストで選択されている項目を一括変更します。
- タグ挿入
タグリストで現在のフォーカス位置に行を挿入します。
- タグ削除
タグリストで選択されている行を削除します。
- 選択範囲の TagID を連番にする
選択されている範囲の TagID が選択範囲の先頭 TagID を基準に連番になります。
- 選択範囲のアドレスを連番にする
選択されている範囲のアドレスが選択範囲の先頭アドレスを基準に連番になります。
- 切り取り
選択された行を切り取ります。
- コピー
選択された行をコピーします。
- 貼り付け
切り取りまたはコピーされた行を現在のフォーカス位置へ挿入します。

ログメッセージ出力

日時	メッセージ
2017/10/12 18:17:24.606	ダミーPLC開始
2017/10/12 18:17:25.672	自動スクリプト開始
2017/10/12 18:17:26.013	Script: Start
2017/10/12 18:17:29.013	Script: timer id =1
2017/10/12 18:17:29.013	Script: timer id =1

PLC 接続、ダミーPLC 起動時に各種メッセージが出力されます。また、スクリプトから OutputLog 関数を実行したときも、この領域にメッセージが出力されます。

通信設定画面

通信設定

通信設定 | PLC動作設定 | PLC接続設定

ポート名: TestPort

ライブラリ: SwCom ヘルプ

通信タイプ: ☒ TCP ☐ UDP ☐ Serial

パラメータ名	設定値	説明
Address	127.0.0.1	IPアドレス
Port	40001	接続先ポート番号
ConnectionTimeout	2000	接続待ちタイムアウト値(ms)、(省略可、デフォルト値=2000ms)
ConnectionRetryTime	2000	接続リトライ間隔(ms) (0:リトライ無し)、(省略可、デフォルト値=2000ms)
Timeout	2000	タイムアウト値(ms)
Retry	0	リトライ回数(0:リトライ無し)
LogFilePath	C:\Log	通信ログ保存パス
LogFileName	TestLog	通信ログファイル名
LogSize	100000	通信ログファイル最大サイズ
LogNum	3	通信ログファイル数

オプション:

OK Cancel

[ポート名]

通信ポート名を設定します。ポート名は半角英数字で設定してください。

[ライブラリ]

通信ライブラリを選択します。通信先 PLC に対応したライブラリを選択してください。通信ライブラリ選択コンボボックス横のヘルプボタンを押下するとライブラリのオプション設定に関するヘルプが表示されます。

[通信タイプ]

接続先に合わせた通信タイプを選択します。

- TCP

TCP で通信します。

- UDP

UDP で通信します。

- シリアル

シリアルポートで通信します。

[パラメーター]

選択した通信タイプにより設定項目が変わります。

- Address (TCP/UDP)

接続先の IP アドレスを設定します。

- Port (TCP/UDP)

接続先の TCP ポート番号を設定します。

- ConnectionTimeout (TCP)

TCP 接続の接続待ちタイムアウト時間をミリ秒単位で設定します。

- ConnectionRectyTime (TCP)

TCP 接続試行リトライ時のリトライ間隔をミリ秒単位で設定します。

- ComPort (Serial)

シリアル通信の COM ポート番号を設定します。(COM1～)

- Baudrate (Serial)

シリアル通信のボーレートを設定します。(1200 / 2400 / 4800 / 9600 / 19200 / 38400 / 57600 / 115200 / 230400)

- DataLen (Serial)

シリアル通信のデータ長を設定します。(7 / 8)

- StopBit (Serial)

シリアル通信のストップビット長を設定します。(1 / 2)

- Parity (Serial)

シリアル通信のパリティビットを設定します。(NONE / ODD / EVEN)

- Flow (Serial)

シリアル通信のフロー制御を設定します。(NONE / RTSCTS)

- Timeout

通信コマンド送信後に PLC からの応答待ち時間をミリ秒単位で設定します。設定された時間が経過しても応答が無い場合、通信タイムアウトとなります。

- Retry

通信タイムアウトが発生した場合に通信リトライを実行する回数を設定します。

- LogFilePath

通信ログファイルの出力先パスを設定します。

- LogFileName

通信ログファイルのファイル名を設定します。空文字を設定した場合は、通信ログは出力されません。

- LogSize

通信ログファイルのサイズを設定します。通信ログファイルがサイズを超えた場合は新しいファイルに通信ログが出力され、古い通信ログファイルは番号を付けてリネームされます。リネームされた通信ログファイルは、LogNum で設定されたファイル数でローテーションされます。

- LogNum

通信ログファイルを残す最大数を設定します。

- 例 LogNum=3 の時

sample-0.log --- 出力中ファイル

sample-1.log --- 1 つ前のファイル

sample-2.log --- 2 つ前のファイル

sample-3.log --- 3 つ前のファイル

sample-0.log が一杯になったときは sample-3.log が削除され他はファイル番号が+1 されます。

[オプション]

通信オプション文字列を設定します。ライブラリごとに設定値は異なります。オプションテキストボックス横の“...”ボタンを押下するとオプション設定詳細画面が表示されます。オプション設定詳細画面では予め用意されたプリセットを読み込んでオプション設定を行うことができます。各ライブラリのオプション設定については、ライブラリのヘルプを参照してください。

ダミーPLC 動作設定

- IP アドレス

ダミーPLC を起動すると TCP 待ち受け状態となります。ここで設定された IP アドレスで待ち受け状態となります。デフォルト値は 0.0.0.0 です。待ち受け IP アドレスを指定したい場合は 0.0.0.0 を任意の IP アドレスに変更してください。

- ポート番号

ダミーPLC の TCP 待ち受けポート番号を設定します。

PLC 接続設定

- モニタ通信間隔

PLC 接続機能利用時のモニタ通信間隔をミリ秒単位で設定します。ここで設定された時間周期でモニタ設定されたタグのデータが PLC から読み込みされます。

コマンドライン引数

SW-TagManager を起動するときにコマンドライン引数を指定して起動することができます。

書式

SwTagManager.exe タグ定義ファイル [-c タグ定義ファイル スクリプトファイル] [-s タグ定義ファイル スクリプトファイル]

[-a タグ定義ファイル]

-c : PLC 接続モードで起動

-s : ダミーPLC モードで起動

-a : プロセスヘアタッチした状態で起動

例 1:

Sample.xml と Sample2.xml を開いた状態で起動

SwTagManager.exe "Sample.xml" "Sample2.xml"

例 2:

Sample.xml を PLC 接続モードで起動した状態で起動

SwTagManager.exe -c "Sample.xml"

例 3:

Sample.xml を PLC 接続モードで起動し、Sample.cs スクリプトを起動

SwTagManager.exe -c "Sample.xml" "Sample.cs"

例 4:

Sample1.xml、Sample1.cs、Sample2.xml、Sample2.cs を PLC 接続モードで起動し、スクリプトを起動

SwTagManager.exe -c "Sample 1.xml" "Sample 1.cs" -c "Sample 2.xml" "Sample 2.cs"