

**BA コントローラ用プログラミングツール
(形式:BA3CL1KW)
取扱説明書**

はじめに

本マニュアルは、Building Automation Controller(以下、「コントローラ」)に対してのプログラム作成、デバッグおよび設定の方法について説明しています。

ご使用になる前に本書をよくお読み頂き、正しくお使い下さい。

なお、本マニュアルは、Windows の操作や用語および IEC61131-3 仕様を理解している方を前提とした内容となっております。依って Windows, IEC61131-3 に関する操作や用語、仕様についてはそれぞれの資料を参照して下さい。

マニュアルについて

本マニュアルに記載されている記号、および共通注意事項は以下のとおりです。

■記号説明

注意

: 操作時の注意事項を記載しています。

MEMO

: 操作時のポイント事項を記載しています。

注意

- ・ 本書の内容に関しては、改良のために予告なしに仕様等変更することがありますのでご了承ください。
- ・ 詳細な処理の内容につきましては、プログラミングツール(BA3CL1KW)のオンラインヘルプ(eCLR)を参照してください。

目 次

1. 概 要	5
1.1. 特長	5
1.2. 仕 様	6
1.3. ソフトウェア構成および制限事項	8
2. ご注意事項	10
3. インストール／アンインストール	11
3.1. 動作環境	11
3.1.1. BA3CL1KW R3 より前	11
3.1.2. BA3CL1KW R3 以降	11
3.2. 対応製品	11
3.3. インストール方法	11
3.3.1. BA3CL1KW R3 より前	12
3.3.2. BA3CL1KW R3 以降	13
3.4. アンインストール方法	14
3.4.1. BA3CL1KW R3 より前	14
3.4.2. BA3CL1KW R3 以降	14
4. プログラミングツール	15
4.1. 画面構成および名称	15
4.1.1. 各部の名称	15
5. プログラムの作成	16
5.1. プログラミングツールの起動	16
5.2. 新規にプロジェクトを作成する	16
5.3. プログラムを作成する	19
5.4. プログラムをコンパイルする	25
6. コントローラとの接続	26
6.1. 通信ゲートウェイの起動	26
6.2. プロジェクトのダウンロード	27
6.3. 前回と異なるプロジェクトのダウンロード	28
6.4. ソースのダウンロード(BA3-CL2、BA3-CL3)	28
6.5. ソースのアップロード(BA3-CL2、BA3-CL3)	29
6.6. デバッグ	30
6.7. プロジェクトの保存	32
6.8. プロジェクトの保存(圧縮)	32
7. 通信ゲートウェイ	33
7.1. [ファイル]－[操作]	33
7.2. [ファイル]－[設定]	34
7.3. [操作]－[時刻設定]	34
7.4. [操作]－[POINT HISTORY]	35
7.5. [操作]－[コントローラ設定]	36
7.6. [操作]－[コントローラ設定]－[ネットワーク変数構成] (BA3-CL3 のみ)	37
7.7. [ヘルプ]－[バージョン情報]	38
8. ネットワーク変数構成ツール	39
8.1. 画面説明[NODE]タブ	40
8.2. 画面説明[Network Variables]タブ	41
8.3. 画面説明[Functional Blocks]タブ	42
8.4. 制限事項	43
8.5. 出荷時のネットワーク変数構成に戻す方法	43
9. ライセンスの登録	44

10. 付録.....	45
10.1. プロジェクトの構成要素.....	45
10.2. プログラム構成ユニット(POU)	45
10.3. ファンクションブロックのインスタンス	46
10.4. ファンクションとファンクションブロックの違い	46
10.5. 変数.....	47
10.6. データ型.....	47
10.6.1. 整数型リテラル.....	48
10.6.2. 実数型リテラル.....	48
10.6.3. 持続時間型リテラル.....	48
10.7. 予約語一覧	49
10.8. テンプレート(BA3-CL1,BA3-CL2,BA3-CL3 Template)	53
10.9. 旧ファームウェアへの対応	55
10.9.1. 接続機器のファームウェアバージョンの確認方法.....	55
10.9.2. 現在のプロジェクトを機器のファームウェアバージョンに合わせる方法.....	56
10.9.3. 「このリソースのビルド設定が整合していません」ダイアログが表示された場合	57

1. 概要

1.1. 特長

- コントローラに対してプログラムの作成、デバッグおよび各所設定を行うことができます。
- プログラミングは IEC61131-3 に準拠した FBD 言語を使用できます。
- プログラミングツールではコントローラが実行中の演算結果をオンラインでモニタすることができます。

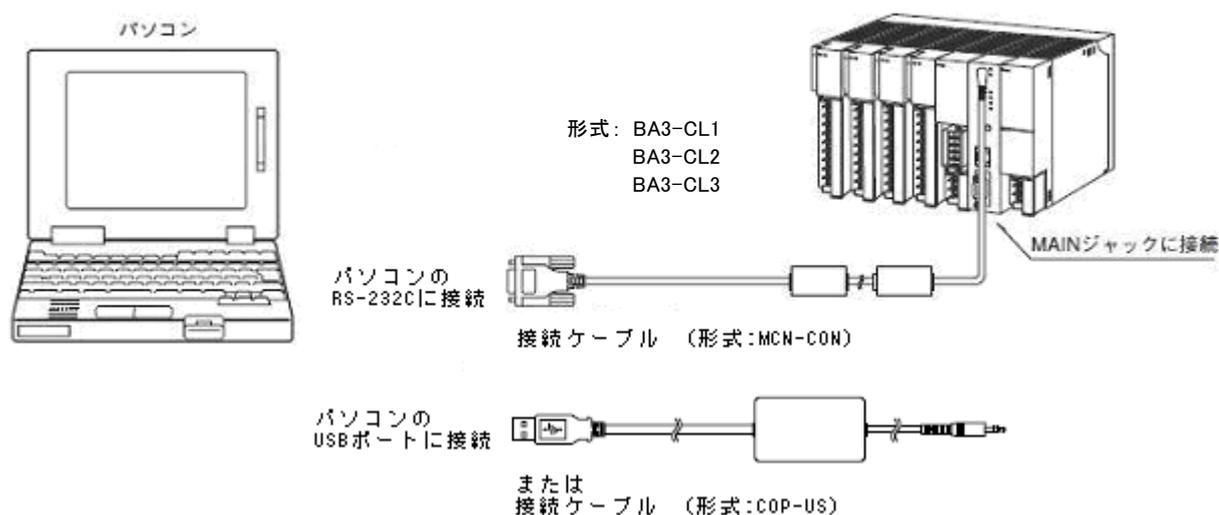


図 1-1 接続

1.2.仕様

■IECプログラム仕様

IEC61131-3 言語	FBD (Function Block Diagram)
データ数	最大 15000 (プロジェクト当たり)、1024 (POU 当たり)
プログラムコード	ワークシート毎に制限なし ワークシートは最大 255 (POU 当たり)
プロジェクト	「物理ハードウェア」数 プロジェクト内で最大1 「コンフィグレーション」数 物理ハードウェア内に最大1定義 「リソース」数 コンフィグレーション内に最大1定義 「タスク」数 リソース内に最大1定義
使用可能 IEC データ型	SINT(8), USINT(8), INT(16), UINT(16), DINT(32), UDINT(32), BOOL(1/8), BYTE(8), WORD(16), DWORD(32), REAL(32), STRING, Time, Arrays, Structures
ユーザ定義型	最大 1024 (プロジェクト当たり)
データ型のネスティング	最大 10 段
構造体データ型メンバー数	最大 1000
配列データ型の要素数	最大 32767
POUs	ライブラリ中の POUs を含み最大 1000 (プロジェクト当たり) インスタンス最大 750
IEC タスク数	最大 1 プログラムインスタンスはタスク当たり最大 500
ユーザファンクションブロック数	POUs 参照
ユーザファンクションブロックのネスティング	最大 128 段
ユーザファンクション数	POUs 参照
ユーザファンクションのネスティング	最大 128 段
ユーザファンクションブロック、ファンクションの入出力パラメータ数	最大 300
ライブラリの登録数	最大 61
変数名	最大長 30 文字(半角英数字)
文字列	最大長 255 文字(半角英数字)
プログラム実行方式	DEFAULT サイクリックスキャン
タスク ウォッチドッグ監視	プログラミングツール上での設定は 20ms ~ 5000ms (10ms 単位) 提供のプロジェクトテンプレートは 500ms で設定されています。
プログラム容量	BA3-CL1 : 転送時 128 KB, 実行時 244 KB BA3-CL2 : 転送時 512 KB, 実行時 256 KB BA3-CL3 : 転送時 512 KB, 実行時 256 KB
データ領域	実行時 244 KB
不揮発性データ領域	4 KB
コントローラへのプロジェクトソース転送機能	BA3-CL1 : なし BA3-CL2 : あり BA3-CL3 : あり

■メニュー（BA3CL1KW R3 より前）

「MULTIPROG」 スタートメニュー	M-SYSTEM → MULTIPROG 5.0 → MULTIPROG 上記以外のメニューは本製品では利用しません。
「MP_CommGateway (BA3-CL1)」 スタートメニュー	M-SYSTEM → MULTIPROG Tools → MP_CommGateway (BA3-CL1) BA3-CL1 と接続する場合に使用します。 (下記 MP_CommonGateway (BA3-CL2) と同時には使用できません)
「MP_CommGateway (BA3-CL2)」 スタートメニュー	M-SYSTEM → MULTIPROG Tools → MP_CommGateway (BA3-CL2) BA3-CL2 と接続する場合に使用します。 (上記 MP_CommonGateway (BA3-CL1) と同時には使用できません)

■メニュー（BA3CL1KW R3, R4）

「MULTIPROG」 スタートメニュー	M-SYSTEM → MULTIPROG 5 → MULTIPROG 上記以外のメニューは本製品では利用しません。
「MP_CommGateway (BA3-CL1)」 スタートメニュー	M-SYSTEM → MULTIPROG Tools → MP_CommGateway (BA3-CL1) BA3-CL1 と接続する場合に使用します。 (下記 MP_CommonGateway (BA3-CL2) と同時には使用できません)
「MP_CommGateway (BA3-CL2)」 スタートメニュー	M-SYSTEM → MULTIPROG Tools → MP_CommGateway (BA3-CL2) BA3-CL2 と接続する場合に使用します。 (上記 MP_CommonGateway (BA3-CL1) と同時には使用できません)

■メニュー（BA3CL1KW R5 以降）

「MULTIPROG」 スタートメニュー	M-SYSTEM → MULTIPROG 5 → MULTIPROG 上記以外のメニューは本製品では利用しません。
「MP_CommGateway (BA3-CL1)」 スタートメニュー	M-SYSTEM → MULTIPROG Tools → MP_CommGateway (BA3-CL1) BA3-CL1 と接続する場合に使用します。 (*1)
「MP_CommGateway (BA3-CL2)」 スタートメニュー	M-SYSTEM → MULTIPROG Tools → MP_CommGateway (BA3-CL2) BA3-CL2 と接続する場合に使用します。 (*1)
「MP_CommGateway (BA3-CL3)」 スタートメニュー	M-SYSTEM → MULTIPROG Tools → MP_CommGateway (BA3-CL3) BA3-CL3 と接続する場合に使用します。 (*1)

*1) 複数の MP_CommonGateway を同時には使用できません。

注意

- ・ ウォッチドッグ検出について
本コントローラでは、異常検出の手段として2つのウォッチドッグを搭載しています。
 - 1)タスク ウォッチドッグ
 - 検出条件
 - ・IEC プログラムがウォッチドッグ監視時間内に実行を終了できなかった。
 - 検出後の動作
 - ・コントローラはウォッチドッグ異常として IEC プログラムを停止します。
 - 2)システム ウォッチドッグ
 - 検出後の動作
 - ・コントローラは自動的に再起動します。
- ・ ゼロ(0)割り時の動作について
本コントローラでは、演算でゼロ割を検出しますとコントローラを停止します。
その為、必要に応じて除算がゼロ割をしないような措置を実施してください。

1.3. ソフトウェア構成および制限事項

弊社が提供するプログラミングツールの構成と制限事項を以下に示します。

■プログラミングツール本体（名称は MULTIPROG）

制限事項	制限内容
言語	FBD のみ それ以外の言語を使用時のプログラミングはサポート対象外となります。
新規プロジェクトの作成方法	「M-SYSTEM」グループの「BA3-CL1 Template」を使用して作成すること。 ユーザアプリケーションは以下の「論理 POU」に記述して下さい。 PLC_Default_PRG PLC_500ms_PRG PLC_1s_PRG PLC_5s_PRG PLC_20s_PRG
グローバル変数の宣言方法	リソース「BA3-CL1」は読み込み専用となっています。 一旦、「リソース」プロパティの「属性」で読み取り専用を解除してからグローバル変数の宣言して下さい。
プロジェクトツリーの リソース情報 (上記グローバル変数以外)	プロジェクトのリソース情報(グローバル変数の追加以外の操作)は変更しないで下さい。
ダウンロードオプション 「ブートプロジェクトの固定」	「ブートプロジェクトを固定」を有効(チェック)して「ダウンロード」すると送られたプロジェクトは自動的に「アクティブ」となります。
ダウンロードオプション 「ソースを含む」	「ソースを含む」を有効(チェック)にした「ダウンロード」には対応していません。従って「プロジェクト コントロール ダイアログ」にある「アップロード」も利用できません。

■ランタイムサポート（名称は eCLR）

制限事項	制限内容
EN/ENO 付きブロックの挿入	メニューの「オブジェクト」にある「EN/ENO 付きブロックの挿入」で行える機能はサポートしていません。

■BA3-CL1 通信ゲートウェイ（名称は MP_CommGateway (BA3-CL1)）

制限事項	制限内容
動作条件	プログラミングツールからターゲット(BA3-CL1)に接続するために使用します。そのためターゲットと接続する前にプログラムを起動しておいて下さい。
BA3-CL1 との接続方法	パソコンに接続されています RS-232C ポートを使用します。使用できるポートは COM1 ~ 9 の範囲です。

■BA3-CL2 通信ゲートウェイ（名称は MP_CommGateway (BA3-CL2)）

制限事項	制限内容
動作条件	プログラミングツールからターゲット(BA3-CL2)に接続するために使用します。そのためターゲットと接続する前にプログラムを起動しておいて下さい。
BA3-CL2 との接続方法	パソコンに接続されています RS-232C ポートを使用します。使用できるポートは COM1 ~ 9 の範囲です。

■BA3-CL3 通信ゲートウェイ（名称は MP_CommGateway (BA3-CL3)）

制限事項	制限内容
動作条件	プログラミングツールからターゲット(BA3-CL3)に接続するために使用します。そのためターゲットと接続する前にプログラムを起動しておいて下さい。
BA3-CL3 との接続方法	パソコンに接続されています RS-232C ポートを使用します。使用できるポートは COM1 ～ 9 の範囲です。

■BA3-CL3 ネットワーク変数構成ツール（名称は BA3CL3_NvConfigurator）

制限事項	制限内容
動作条件	ターゲット(BA3-CL3)のネットワーク変数(NVI, NVO, NCI)構成を変更するために使用します。
BA3-CL3 との接続方法	このツールは構成内容からコントローラダウンロードファイル(拡張子 cl3nvcfg)を生成します。ダウンロードファイルをターゲット(BA3-CL3)へ書き込むためには、前述の BA3-CL3 通信ゲートウェイを使用します。

2. ご注意事項

■取扱いについて

- ・本体の取外または取付を行う場合は、危険防止のため必ず、電源および入力信号を遮断して下さい。

■設置について

- ・屋内でご使用下さい。
- ・塵埃、金属粉などの多いところでは、防塵設計のきょう体に収納し、放熱対策を施して下さい。
- ・振動、衝撃は故障の原因となることがあるため極力避けて下さい。
- ・周囲温度が0～+55℃を超えるような場所、周囲湿度が30～90% RHを超えるような場所や結露するような場所でのご使用は、寿命・動作に影響しますので避けて下さい。

■配線について

- ・配線(電源線、入力信号線、出力信号線)は、ノイズ発生源(リレー駆動線、高周波ラインなど)の近くに設置しないで下さい。
- ・ノイズが重畳している配線と共に結束したり、同一ダクト内に収納することは避けて下さい。

■入出力カードの取扱いについて

- ・製品に添付の取扱説明書をご覧下さい。

■その他

- ・本器は電源投入と同時に動作しますが、すべての性能を満足するには10分の通電が必要です。
- ・制御・通信の二重化には対応していません。

注 意

- ・本コントローラは、一般産業用です。安全機器や事故防止システムなど人命や自然破壊など、より高い安全性が要求される用途、また、より高い信頼性が要求される用途には、必ずしも万全の機能を持つものではありません。
- ・非常停止回路・インタロック回路など特に安全性が要求される回路は本コントローラの外部で構成して下さい。
- ・安全にご使用いただくために、機器の設置や接続は、電氣的知識のある技術者が行って下さい。
- ・運転中のプログラム変更、強制出力、起動、停止などの操作は十分安全を確認してから行ってください。操作ミスにより機器の誤動作、機器の破損や事故の恐れがあります。
- ・コントローラに触れる前には、接地された金属などに触れておき人体などに帯電している静電気を放電させてください。過大な静電気は誤動作、故障の原因となります。
- ・電源を投入したままでのシステム変更(I/Oカードの着脱など)は行わないでください。通電中のシステム変更は誤動作、故障の原因になります。
- ・プログラムの書込み中に電源を切らないでください。機器の誤動作、故障の原因になります。

3. インストール／アンインストール

ここでは、プログラミングツールのインストール方法および動作環境のセットアップについて説明します。

3.1. 動作環境

プログラミングツールの動作環境は、次の基本ソフトウェア(OS)を保証します。

3.1.1. BA3CL1KW R3 より前

- ・Windows XP Professional ServicePack2 以降(日本語版)
及び
Internet Explorer 4.02 以上
Microsoft .NET Framework 2.0

3.1.2. BA3CL1KW R3 以降

- ・Windows XP Professional ServicePack3 以降 (日本語版)
- ・Windows Vista (32bit) ServicePack2 以降 (日本語版)
- ・Windows 7 (32bit/64bit) (日本語版)
- ・Windows 10 (32bit/64bit) (日本語版)
及び
Internet Explorer 6 以上または Microsoft Edge
Microsoft .NET Framework 2.0

注意

- ・上記以外の OS での動作確認は行っていません。
- ・必要ディスク容量については、使用する環境(作成されますプロジェクト等)に影響します。

3.2. 対応製品

弊社が提供するプログラミングツールが対応する製品を以下に示します。

- ・BA コントローラ (BA3-CL1)
- ・BA コントローラ (BA3-CL2)
- ・BA コントローラ (BA3-CL3)

3.3. インストール方法

以降の手順に従ってインストールを行って下さい。

注意

- ・インストールを行う前に、動作している他のアプリケーションをすべて終了して下さい。
- ・コンピュータ管理権限(Administrator)の属性を持つユーザでログオンして下さい。
- ・ユーザ名が漢字などの半角英数字以外を含んでいる場合はインストールに失敗することがあります、その際は、それらの文字が含まれないユーザでインストールを行ってください。

3.3.1. BA3CL1KW R3 より前

- 1) インストール CD をドライブにセットします。
自動起動が許可されている場合は自動的にインストール手順が表示されますのでご確認ください。
以降にインストールの手順を説明します。

インストールは

- ・プログラミングツール本体 (MULTIPROG)
- ・ランタイムサポート (eCLR)
- ・通信ゲートウェイ (MP_CommGateway)
- ・環境設定

の順で行われます。

各項目でインストールのキャンセルが指定できますが、キャンセルできるのはその項目のみで次の項目のインストール作業は開始されます。その場合、最終の環境設定などの作業で正常に処理されないことがあります。その際は、インストールされたものをアンインストールし再度インストール作業を行ってください。

- 2) スタートメニューの「ファイル名を指定して実行(R)」にて CD-ROM の setup.bat を実行します。



図 3-1 ファイル名を指定して実行

- 3) その後のインストール作業は画面に表示されます指示に従い行います。
インストールの最中に以下のダイアログメッセージが表示されることがあります。その場合は「OK」を押して処理を継続させてください。



図 3-2 インストール中の警告ダイアログ

- 4) インストール作業が終了しますと以下の画面表示になります。
ENTER キーを押して画面を閉じます。これで作業は終了しました。

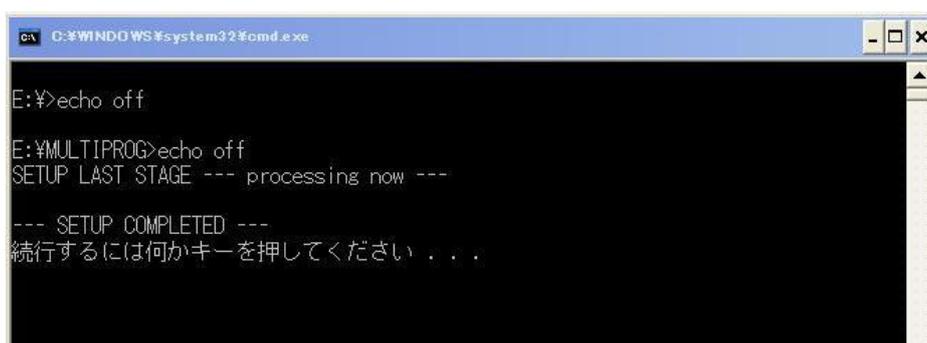


図 3-3 インストール終了

3.3.2. BA3CL1KW R3 以降

- 1) インストール CD をドライブにセットします。
自動起動が許可されている場合は自動的にインストール手順が表示されますのでご確認ください。
以降にインストールの手順を説明します。
- 2) スタートメニューの「ファイル名を指定して実行(R)」にて CD-ROM の setup.exe を実行します。

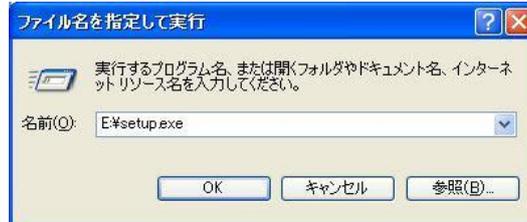


図 3-4 ファイル名を指定して実行

- 3) その後のインストール作業は画面に表示されます指示に従い行います。
次のダイアログが表示される場合は「インストール」を押してください。



図 3-5 セットアップ コンポーネントのインストール

- 4) インストールの最後に以下のダイアログメッセージが表示されることがあります。その場合は「OK」を押して処理を継続させてください(これらのダイアログ表示は他のウィンドウの背後になり隠れている場合がありますのでご注意ください)。



図 3-6 インストール中の警告ダイアログ

- 5) インストール作業が終了しますと以下の画面が表示されます。



図 3-7 インストール終了

「注意」
このセットアップでは、本製品とともに以下の共通モジュールがインストールされます。

- ・ Microsoft .NET Framework
- ・ Microsoft Visual C++ 2005 Redistributable
- ・ Microsoft Visual C++ 2008 Redistributable

これらは本製品の動作に必要なものなので本製品をアンインストールする迄システムから削除しないで下さい。

3.4. アンインストール方法

以降の手順に従ってアンインストールを行って下さい。

3.4.1. BA3CL1KW R3 より前

- 1) 「コントロールパネル」の「プログラムの追加と削除」から「MULTIPROG 5.0」を選択します。
- 2) 「変更／削除」を選択し、画面の指示に従って削除します。
- 3) 次に「MULTIPROG Communication Gateway」を同様の手順で削除します。

3.4.2. BA3CL1KW R3 以降

- 1) 「コントロールパネル」の「プログラムの追加と削除」から「BA3CL1KW R□」（□はリビジョン番号）を選択します。
- 2) 「変更／削除」を選択し、画面の指示に従い削除します。

4. プログラミングツール

ここでは、プログラミングツールの機能説明と使用方法について説明します。

4.1. 画面構成および名称

ここでは、各部の名称とその主な機能について説明します。

4.1.1. 各部の名称

【基本画面】

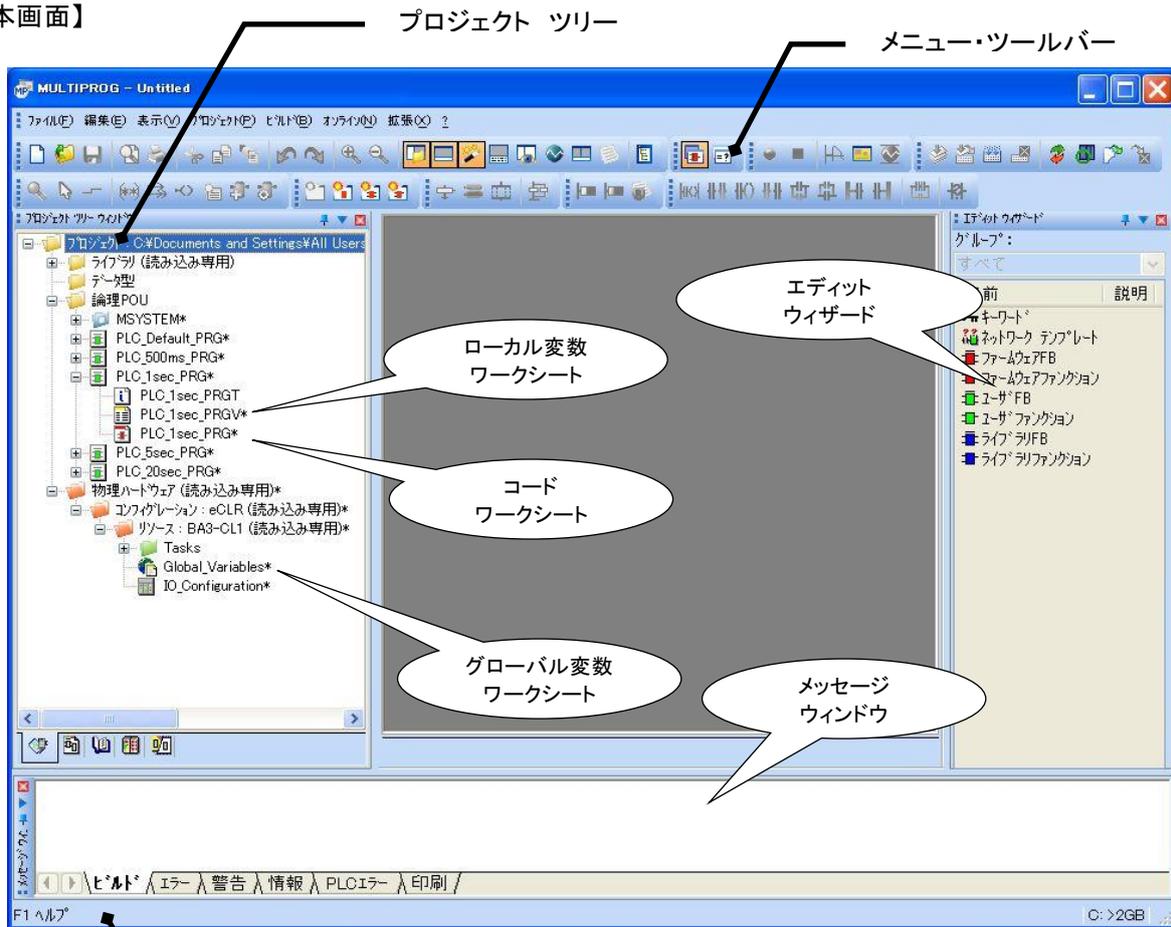


図 4-1 画面レイアウト

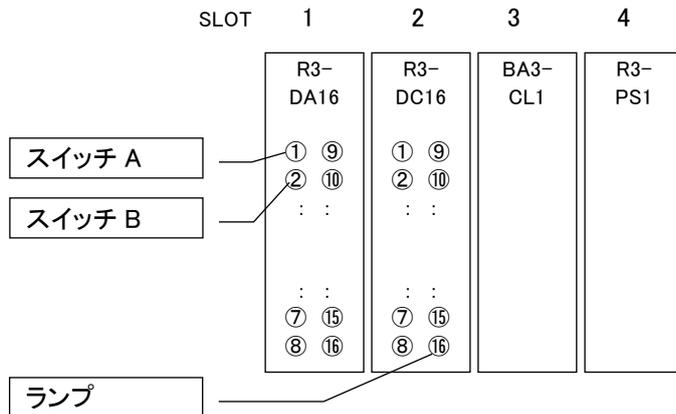
名称	機能
メニュー ツールバー	各メニューを選択します。 各機能呼び出せます。
プロジェクト ツリー	選択タブで選択されている情報を表示します。 例えばプロジェクトが選択されている場合はライブラリ、データ型、定義された POU などがツリー表示されます。
ローカル変数ワークシート	ダブルクリックすると POU のローカル変数を宣言するためのグリッドを表示します。
コードワークシート	ダブルクリックすると POU のコードを編集するウィンドウを表示します。
グローバル変数ワークシート	ダブルクリックするとグローバル変数を宣言するためのグリッドを表示します。
エディット ウィザード	グループを選択することで POU 一覧など関連情報が表示されます。
メッセージ ウィンドウ	ビルド結果、エラーなどが表示されます。
ステータスバー	現在状態を表示します。

5. プログラムの作成

ここでは簡単なプログラムの作成を通してプログラミングツールの使用方法を説明します。

本プログラミングツールではプログラムの管理をプロジェクトで行います。このプロジェクトではプログラムコード、変数設定などの POU 情報とタスク設定などのリソース情報を管理します。

ここでは下記の I/O 構成でサンプルプログラムを作成します。



5.1. プログラミングツールの起動

スタートメニューからプログラミングツールを起動します。

[スタート]—[プログラム]—[M-SYSTEM]—[MULTIPROG5]の順で選択し、「MULTIPROG」をクリックします。

5.2. 新規にプロジェクトを作成する

1) ファイルメニューから「新規プロジェクト」を選択します。



図 5-1 ファイルメニュー

2)今回は BA3-CL1 を使用するため、新規プロジェクトダイアログのテンプレート一覧より[M-SYSTEM]タブにある[BA3-CL1 Template]を選択します。BA3-CL2を使用する場合には[BA3-CL2 Template]、BA3-CL3を使用する場合には[BA3-CL3 Template]を選択します

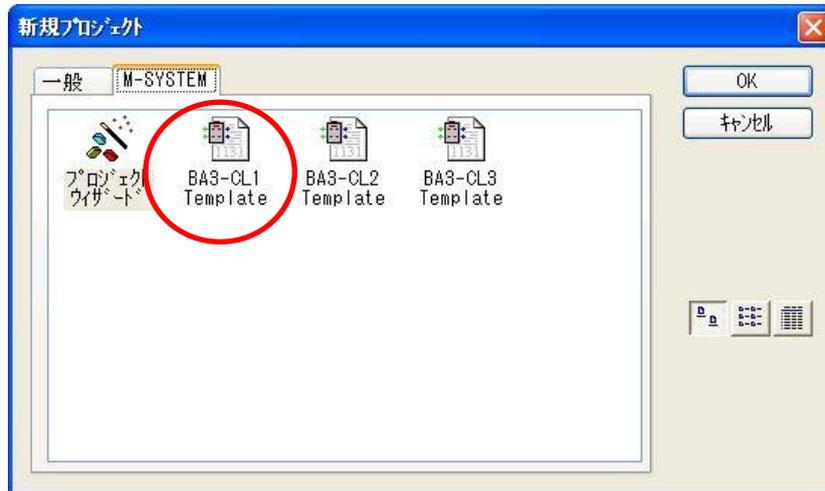


図 5-2 テンプレート

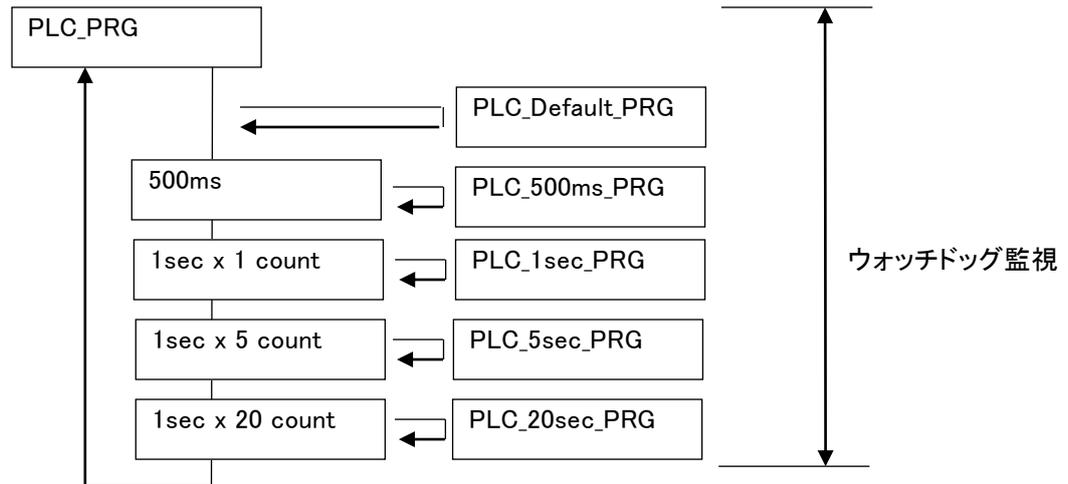
3)テンプレートにより次のようなプロジェクトが構築されます。



図 5-3 プロジェクトツリー

ユーザアプリケーションは次の POU に記述します。
下記の POU は PLC_PRG より呼び出されます。

POU Name	内容
PLC_Default_PRG	Default スキャン毎に呼び出し
PLC_500ms_PRG	500ms 毎に呼び出し
PLC_1sec_PRG	1sec 毎に呼び出し
PLC_5sec_PRG	5sec 毎に呼び出し
PLC_20sec_PRG	20sec 毎に呼び出し

**注意**

- ・ ウォッチドッグ検出について
PLC_PRG の開始から終了迄が監視対象となります。PLC_PRG から呼び出される各 POU のトータル処理時間がウォッチドッグの監視対象となります。
- ・ PLC_PRG から呼び出されるPOUについて
PLC_PRG からの呼び出しは順次呼び出しで実現しています。もしプログラムの実行時間が基準時間(上記の場合は 500ms,1sec)を超える場合は次の呼び出し時間に影響を与えます。

作成されるグローバル変数

Name	型	内容
SYSVAR_PULSE_100ms	BOOL	0.1 秒 ON、0.1 秒 OFF の繰り返し
SYSVAR_PULSE_500ms	BOOL	0.5 秒 ON、0.5 秒 OFF の繰り返し
SYSVAR_PULSE_1sec	BOOL	1 秒 ON、1 秒 OFF の繰り返し

5.3. プログラムを作成する

- 1) 作成されたプロジェクトにプログラムを作成していきます。今回は1秒周期にプログラムを作成します。プロジェクトツリーのノード PLC_1sec_PRG を開き「PLC_1sec_PRG」をダブルクリックし編集ウィンドウを開きます。

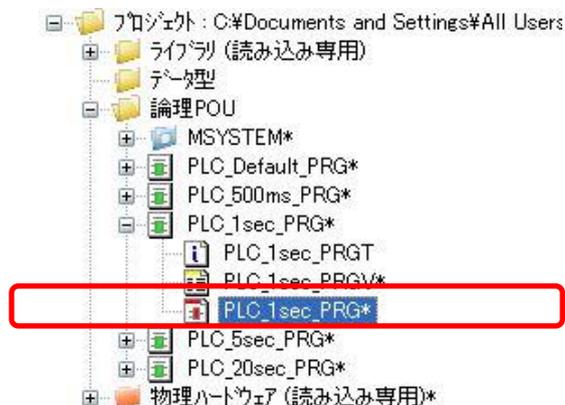


図 5-4 アプリケーション記述 POU

- 2) 開いた POU には既にテンプレートでサンプルコードが作成されています。サンプルコードは必要に応じて削除したり新しいアプリケーションで置換することができます。ここではサンプルコードを残し新しいコードを追加します。
コードはウィンドウ上のカーソル(+)の位置に挿入されます。

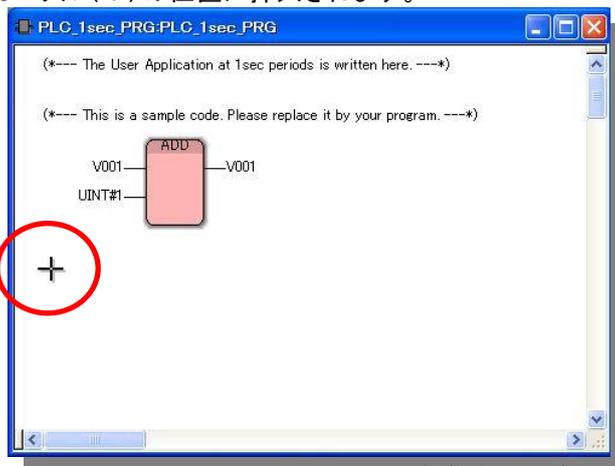


図 5-5 PLC_1sec_PRG 編集ウィンドウ

- 3) まずは今回のアプリケーションを説明するコメント(注釈)を追加します。コメントは挿入したい位置にカーソル(+)を移動しメニューの[オブジェクト]-[テキスト(コメント)]か下記のツールバーを押します。



図 5-6 ツールバー

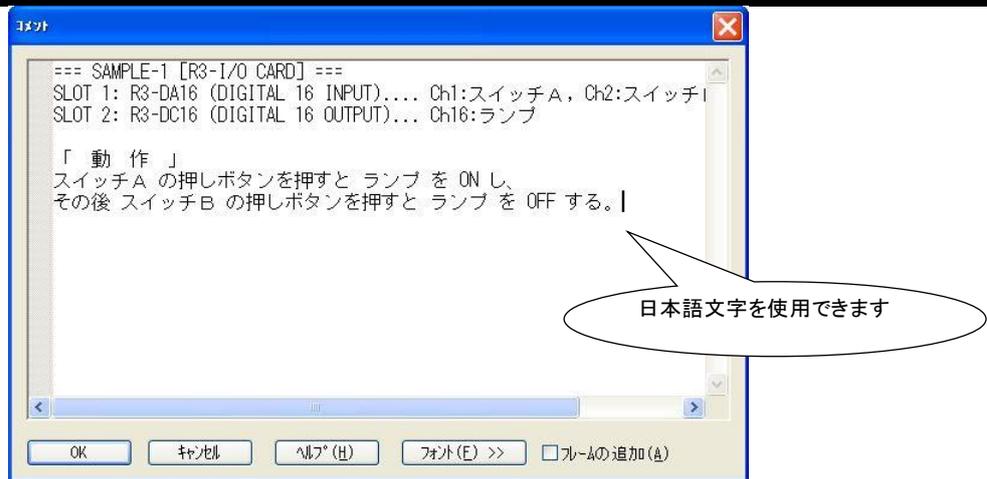


図 5-7 コメント編集ウィンドウ

- 4) 次に I/O カードから入力するためのデジタル入力「DIG_INPUT」ファンクションブロックのインスタンスを追加します。挿入する位置にカーソル(+)を移動しておき、挿入するファンクションブロックを画面右側のエディットウィザードのコンボボックスで<すべてのFU/FB>か<MsysFwLibBA3CL1>*1を選択し、表示されるPOU一覧から「DIG_INPUT」を選択(ダブルクリック)します。ここではファンクションブロックのインスタンス名を「DIG_INPUT_1」とします。

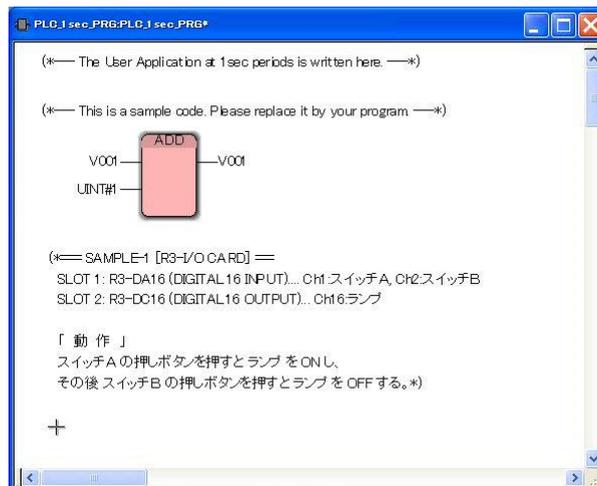


図 5-8 コメント挿入後

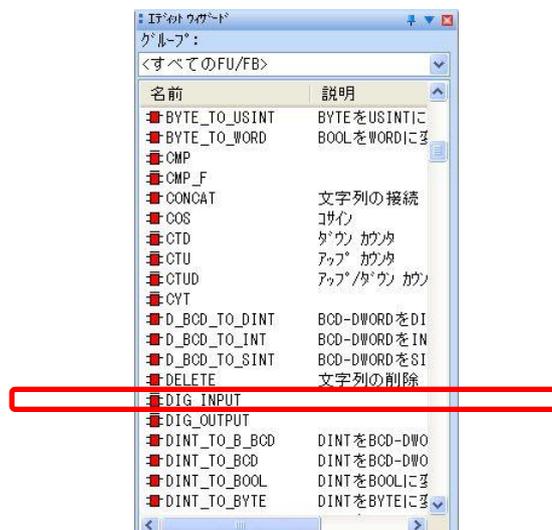


図 5-9 エディットウィザード

*1) BA3-CL2 の場合は<MsysFwLibBA3CL2>となります。

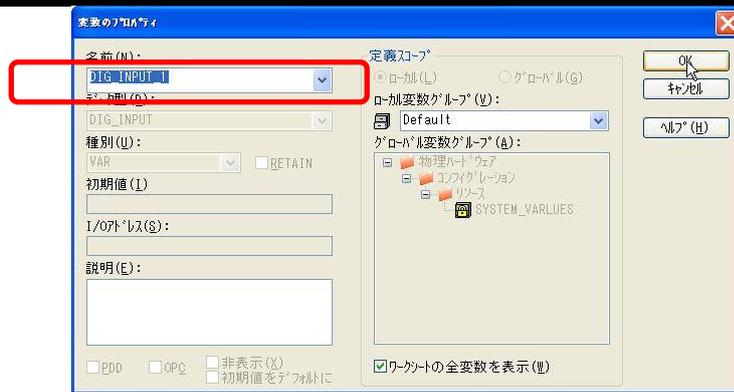
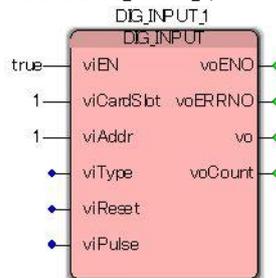


図 5-10 変数のプロパティ

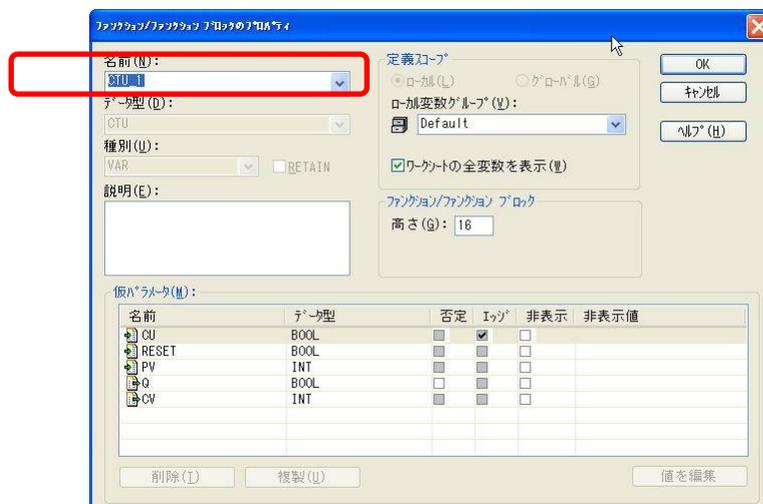
5) 次は挿入した「DIG_INPUT_1」の入出力パラメータを設定します。



6) 入出力パラメータ「viEN」は「true」、「viCardSlot」は「1」、「viAddr」は「1」を設定します。



7) 入力「スイッチ A」で ON して入力「スイッチ B」で OFF する動作を実現するために、ここではアップカウンタを利用します。アップカウンタ「CTU」ファンクションブロックのインスタンスを追加します。挿入する位置にカーソル(+)を移動しておき、挿入するファンクションブロックを画面右側のエディットウィザードのコンボボックスで<すべての FU/FB>を選択し、表示される POU 一覧から「CTU」を選択(ダブルクリック)します。ここではファンクションブロックのインスタンス名を「CTU_1」とします。



- 8) 「DIG_INPUT_1」の出力「vo」を「CTU_1」の入力「CU」とする為に接続します。
 接続はメニューの「オブジェクト」「オブジェクトの接続」か下記のツールバーを押して「接続モード」に移行します。
 次の「DIG_INPUT_1」の「vo」をクリックし続いて「CTU_1」の「CU」をクリックし接続します。ここで一旦「接続モード」を解除するために「右クリック」しておきます。

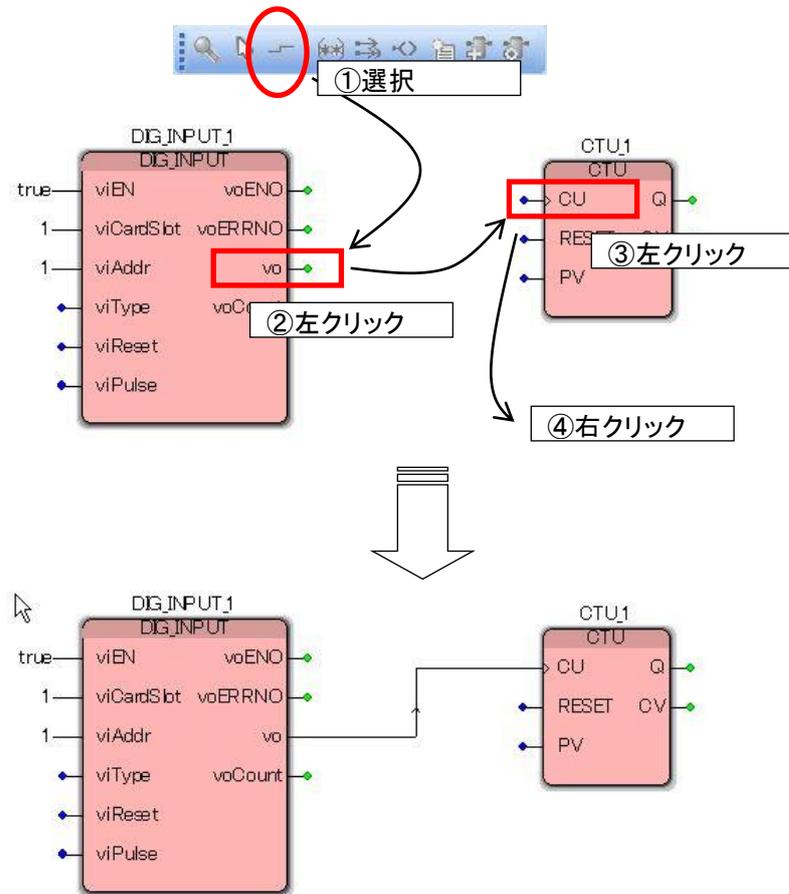
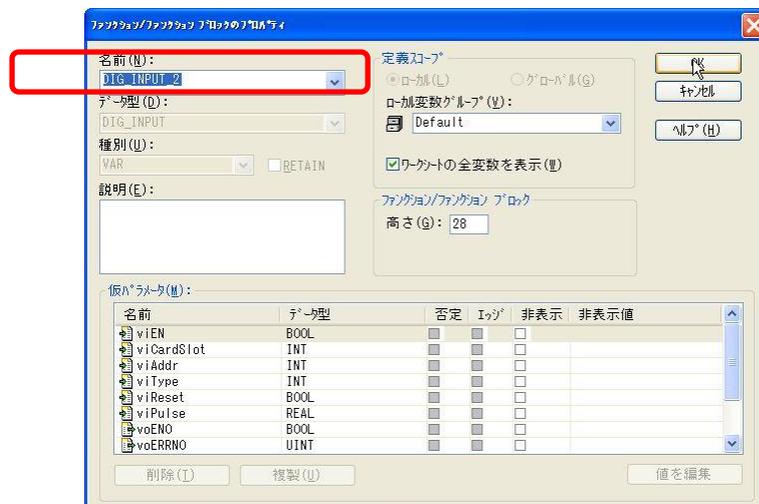
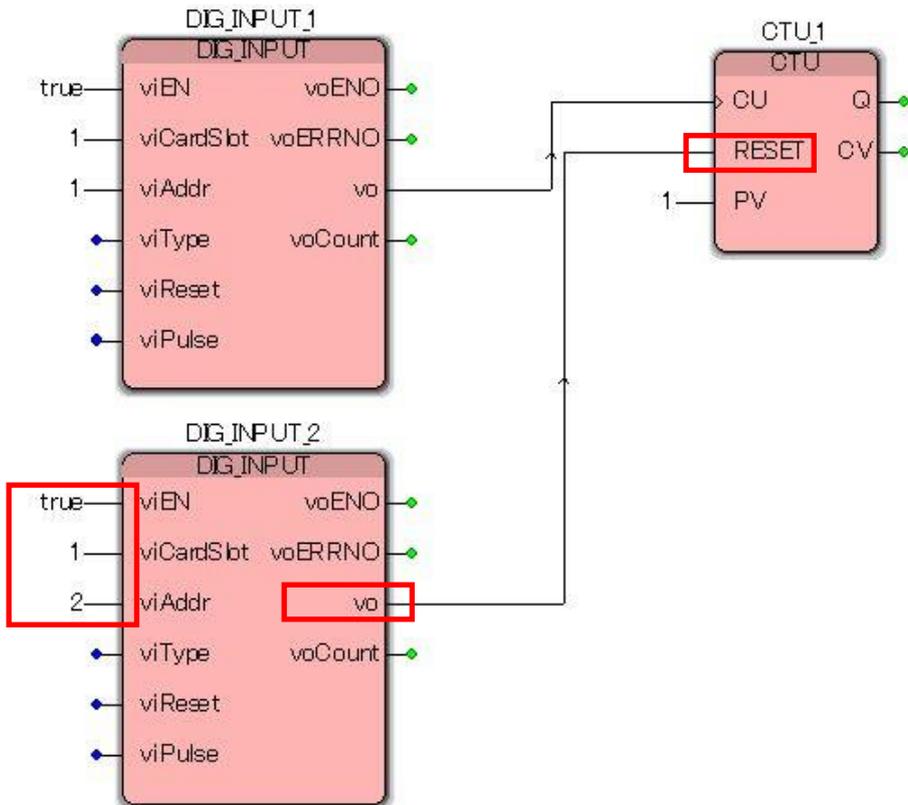


図 5-11 端子間の接続

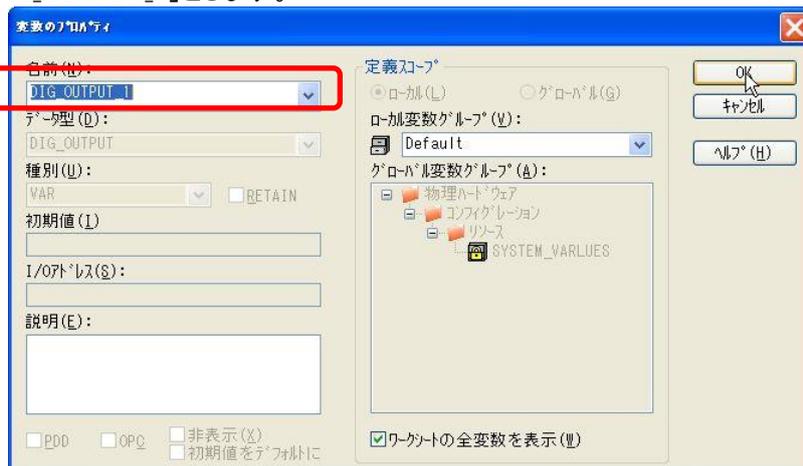
- 9) リセット入力のデジタル入力「DIG_INPUT」ファンクションブロックのインスタンスを追加します。
 挿入する位置にカーソル(+)を移動しておき、挿入するファンクションブロックを画面右側のエディットウィザードの POU 一覧から「DIG_INPUT」を選択(ダブルクリック)します。ここではファンクションブロックのインスタンス名を「DIG_INPUT_2」とします。



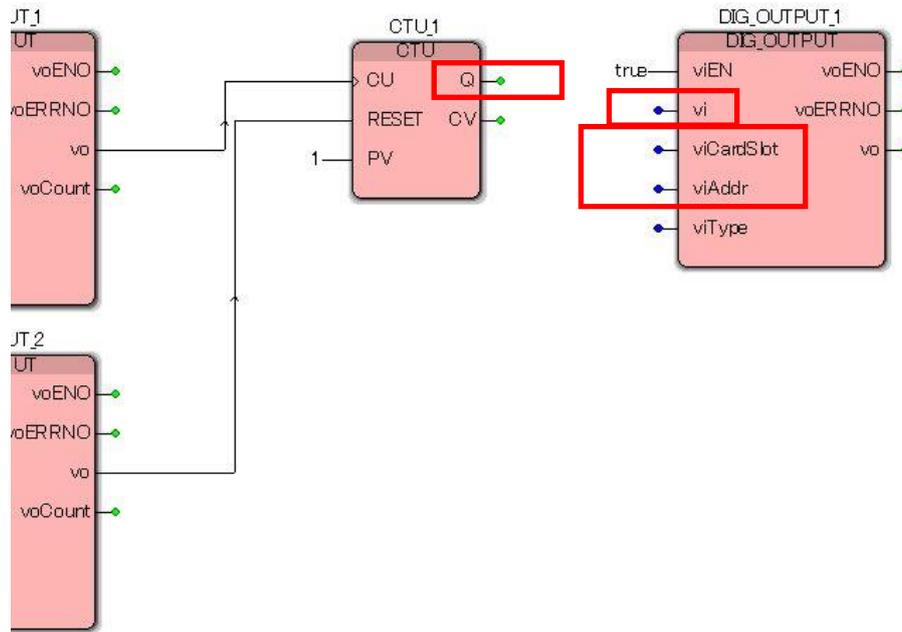
- 10) 「DIG_INPUT_2」の出力「vo」を「CTU_1」の入力「RESET」に接続します。
 次に残りの入力パラメータを設定していきます。「CTU_1」の入力「PV」にカウント1で出力Qを出力したいので「1」を設定します。「DIG_INPUT_2」の入力「viCardSlot」は「1」、「viAddr」は「2」をそれぞれ設定します。



- 11) 結果を出力するためにデジタル出力「DIG_OUTPUT」ファンクションブロックのインスタンスを追加します。挿入する位置にカーソル(+)を移動しておき、挿入するファンクションブロックを画面右側のエディットウィザードの POU 一覧から「DIG_OUTPUT」を選択(ダブルクリック)します。ここではファンクションブロックのインスタンス名を「DIG_OUTPUT_1」とします。



- 12) 「CTU_1」の出力「Q」を「DIG_OUTPUT_1」の入力「vi」に接続します。
次に「DIG_OUTPUT_1」の入力「viCardSlot」は「2」、「viAddr」は「16」をそれぞれ設定します。



- 13) 下図が完成したプログラムです。「DIG_INPUT_1(スイッチ A)」が入力されるとアップカウンタ「CTU_1」がカウントアップされ結果1回で設定値「PV」に達し出力「Q」がtrueとなります。その出力が「DIG_OUTPUT_1(ランプ)」にセットされ出力カードの該当出力を ON します。また「DIG_INPUT_2(スイッチ B)」が入力されると接続されているアップカウンタ「CTU_1」の RESET が実行されアップカウンタのカウント値がリセット(0)、出力「Q」は false になります。よって接続されている出力カードの該当出力も OFF になります。

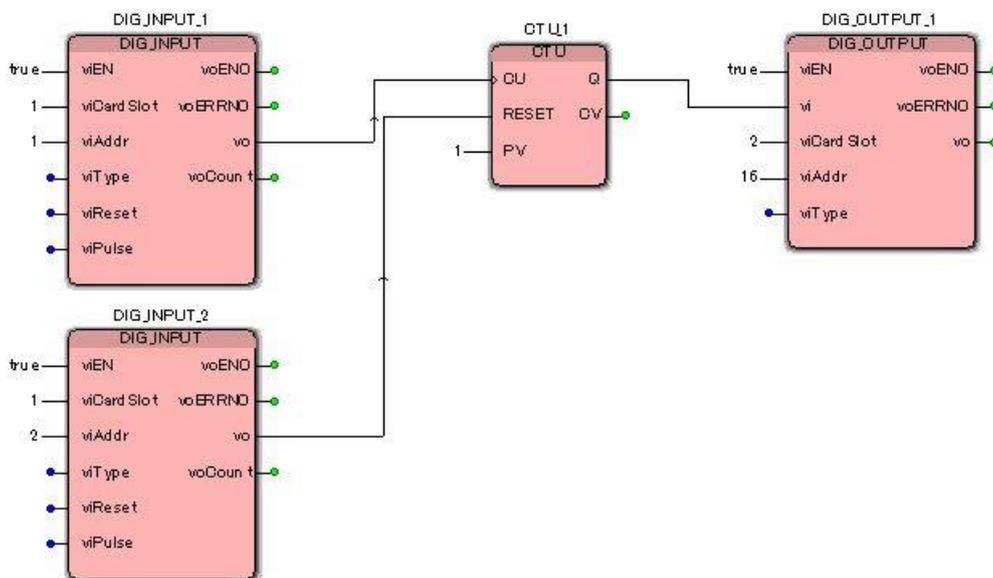


図 5-12 完成したプログラム

5.4. プログラムをコンパイルする

- 1) プログラムを実行するには実行イメージの作成(コンパイル)を行います。
コンパイルはメニューの[ビルド]-[メイク]か下記のツールバーを押します。



- 2) コンパイルが終了しますとメッセージウィンドウの「ビルド」タブに結果が表示されます。最後に「エラー」、「警告」の数が表示されます。この数が0個でない場合はそれぞれの「エラー」、「警告」タブに詳細が表示されていますので解決して再度コンパイルを行います。「警告」の中には表示のみでプログラムの修正を必要とせずそのまま処理を継続してもよいメッセージもあります。「情報」タブでは実行イメージのサイズが表示されます。

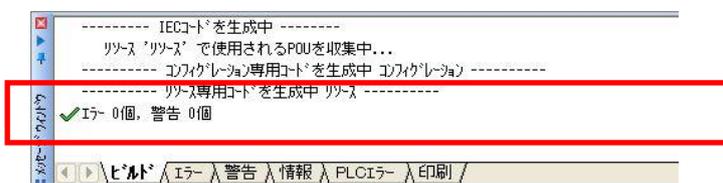


図 5-13 コンパイル結果 (ビルド)

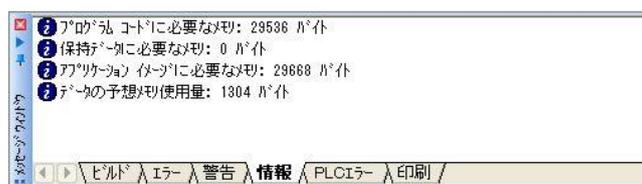


図 5-14 コンパイル結果 (情報)

注意

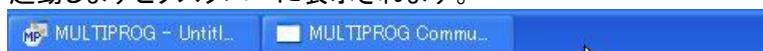
- ・ 実行イメージのサイズ
「情報」タブでは実行イメージのサイズや必要とするメモリ情報が表示されます。各項目がコントローラの制限値を超えますと、コントローラへの転送やコントローラでの実行ができません。各項目がコントローラの制限値を超えないようにプログラムの作成を行う必要があります。

6. コントローラとの接続

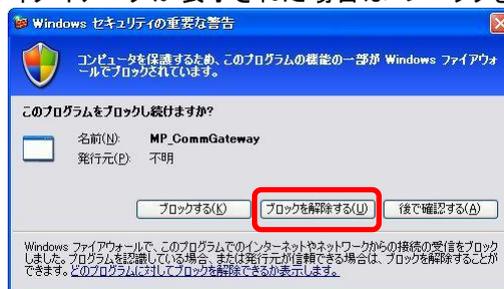
ここではプログラミングツールとコントローラとの接続方法を説明します。
プログラミングツールとコントローラを接続するには「通信ゲートウェイ」ソフトウェアを使用します。

6.1. 通信ゲートウェイの起動

- 1) スタートメニューからプログラミングツールを起動します。
[スタート]–[プログラム]–[M-SYSTEM]–[MULTIPROG Tools]の順で選択して
BA3-CL1 と接続の場合は「MP_CommGateway (BA3-CL1)」
BA3-CL2 と接続の場合は「MP_CommGateway (BA3-CL2)」
BA3-CL3 と接続の場合は「MP_CommGateway (BA3-CL3)」
のどちらかをクリックします。
- 2) ソフトウェアが起動するとタスクバーに表示されます。



もし Windows によりセキュリティダイアログが表示された場合は「ブロックを解除する」を選択してください。



注意

- ・ 通信ゲートウェイ
BA3-CL1、BA3-CL2、BA3-CL3 それぞれに専用の通信ゲートウェイが用意されています。適切な通信ゲートウェイが起動されていない場合には、これ以降の操作で「タイムアウト」エラーとなり通信が正常に行われません。

6.2. プロジェクトのダウンロード

- 1) プロジェクトをダウンロードするためには「プロジェクト コントロール ダイアログ」を表示します。メニューの[オンライン]-[プロジェクト コントロール]が下記のツールバーを押します。



- 2) プロジェクトをダウンロードするために「ダウンロード」を押します。



注意

- ・ ダイアログの「状態」表示
ダイアログの状態が「オン」、「停止[Stop]」または「実行中」であることを確認します。もし「タイムアウトしました」が表示されている場合はコントローラとの接続あるいは通信ゲートウェイプログラムの再起動を行う必要があります。
- ・ プログラムや変数宣言の変更は、変数値や動作が変化する場合があるため、安全面や動作結果に注意する必要があります。

- 3) ダウンロード完了後は次のように表示されます。プログラムの実行は必要に応じて「コールド」ボタンを押します。



- 4) プログラムが実行されると状態に「実行中」と表示されます。確認できたらこのダイアログを閉じることをお勧めします。このダイアログはコントローラの情報を確認するために頻繁に通信を行っており、そのためこのダイアログが表示されたまま次で説明しますオンラインでのモニタを行うと表示の更新やコンピュータのレスポンスが著しく低下することがあります。



注意

- 「プロジェクト コントロール ダイアログ」の表示
「ダウンロード」などの操作が終わった後はこのダイアログを閉じることをお勧めします。このダイアログが表示されている間はコントローラの情報を確認するために頻繁に通信を行っております。そのため、このダイアログが表示されたままオンラインでのモニタなどを行うと表示の更新やコンピュータのレスポンスが著しく低下することがあります。

6.3. 前回と異なるプロジェクトのダウンロード

コントローラへ以前のプロジェクトと異なるプロジェクトをダウンロードするためには「プロジェクト コントロール ダイアログ」にて次の操作を必要とします。

- 1) プロジェクトが動作中の場合は「停止」を押します。
- 2) コントローラ内に存在する以前のプロジェクトを消去するために「リセット」を押します。リセットの確認ダイアログでは「はい」を押します。

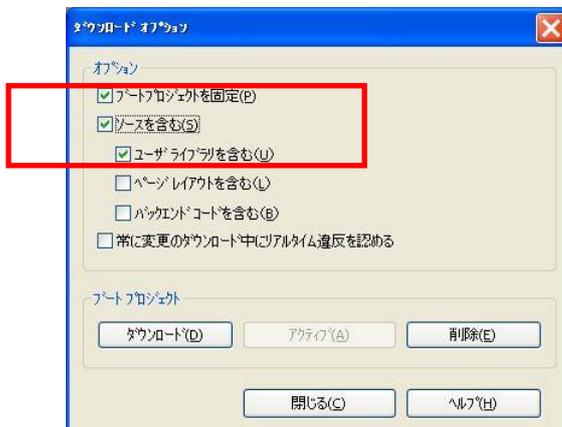


- 3) 新しいプロジェクトをダウンロードするために「ダウンロード」を押します。

6.4. ソースのダウンロード(BA3-CL2、BA3-CL3)

コントローラへ現在プロジェクトのソースを含めてダウンロードすることができます。その場合は「ダウンロード」の操作の前に次の操作を必要とします。

- 1) 「プロジェクト コントロール ダイアログ」の「その他」を押します。
- 2) ダウンロードオプションダイアログの「ブートプロジェクトの固定」と「ソースを含む」(このとき「ユーザライブラリを含む」以外はチェックしないでください)をチェックして閉じます。ここでの設定はダイアログを閉じても記憶されますのでダウンロードのたびに設定する必要はありません。
また、ダウンロード操作は、このダイアログにある「ダウンロード」でも操作できますは、今回はリソースダイアログで操作します。

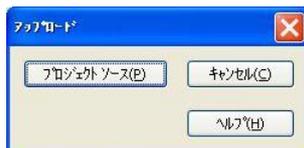


- 3) リソースダイアログの「ダウンロード」を押します。自動的にプロジェクトの圧縮を行いコントローラへダウンロードされます。

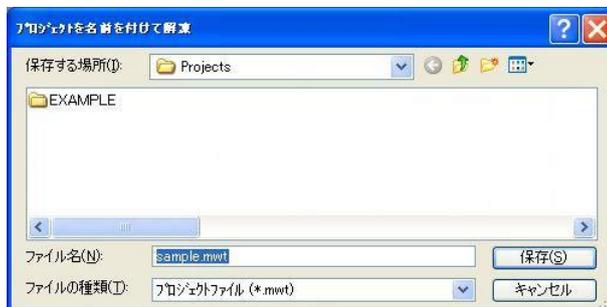
6.5. ソースのアップロード(BA3-CL2、BA3-CL3)

コントローラへプロジェクトのソースを含めてダウンロードされている場合に、そのダウンロードされたプロジェクトをコントローラからアップロードできます。

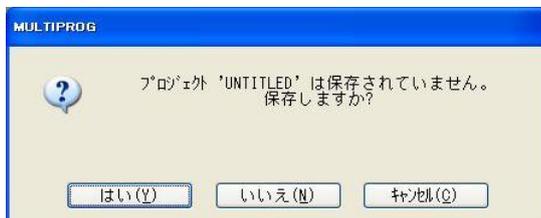
- 1) アップロードを行うため新規にプロジェクト(テンプレートで BA3-CL2 あるいは BA3-CL3 を選択)を作成します。このプロジェクトはアップロード操作のために必要なもので最終的には破棄します。
- 2) リソースダイアログの「アップロード」を押します。次のアップロードダイアログでは「プロジェクトソース」を押します。



- 3) アップロードが終了すると、アップロードされたプロジェクトをパソコン上に格納するためのファイル名を求められます。



- 4) アップロード操作のために作成したプロジェクトの保存を求められるので、「いいえ」を押して破棄(保存しない)します。



- 5) アップロードされたプロジェクトが開きアップロード操作が完了します。

注意

- ・ リソースダイアログ「アップロード」の表示
「アップロード」はコントローラにソースをダウンロードされていなければ有効になりません。したがって「アップロード」が有効でない場合はコントローラ内にソースが転送されていないことを意味しています。

6.6. デバッグ

- 1) プログラムの実行状態を確認するために「デバッグ」機能を利用できます。この「デバッグ」機能では実行中のプログラムの変数値をモニタすることや強制的に値を代入するなど可能にします。「デバッグ」状態を設定するためには、デバッグしたい POU コード編集ウィンドウを表示させた状態で、メニューの[オンライン]-[デバッグ]か下記のツールバーを押します。



- 2) 「デバッグ」状態では「変数の現在値」が表示されます。接続線は現在値を色で表現し true は赤色、false は青色で視覚的に確認することができます。

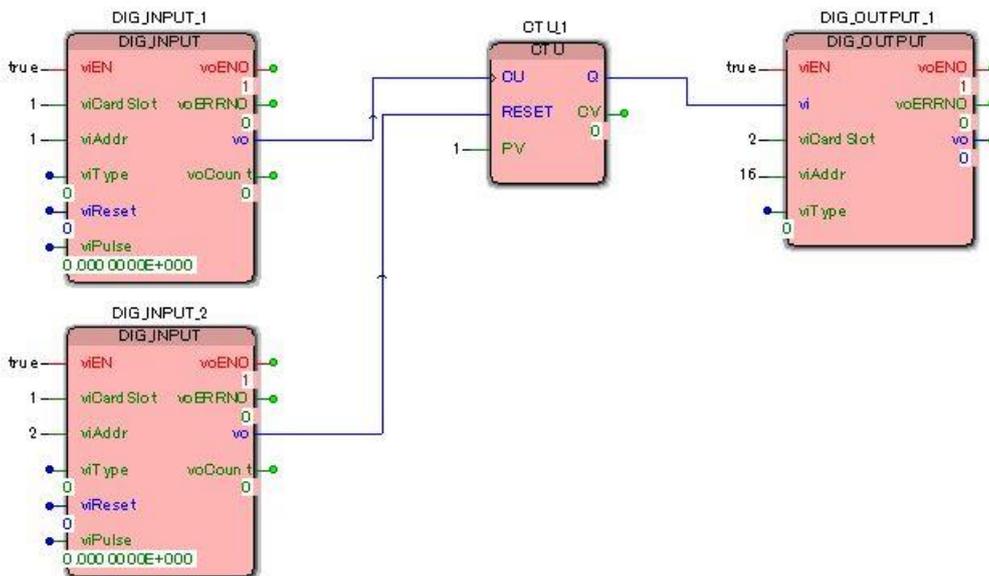


図 6-1 デバッグ時の表示例（まだ入力スイッチ A,スイッチ B を ON していない）

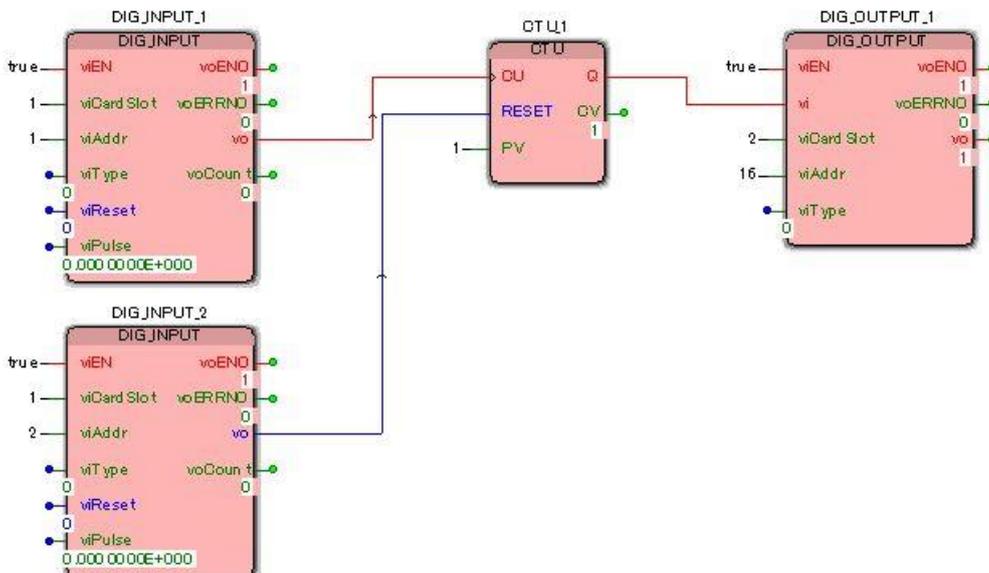


図 6-2 デバッグ時の表示例（入力スイッチ A を ON）

6.7. プロジェクトの保存

- 1) 新規に作成したプロジェクトは名称が「Untitled」となっています。保存するには希望するプロジェクト名を付けることができます。保存方法はメニューの[ファイル]－[プロジェクトを名前を付けて保存／圧縮]を選択します。

「制限事項」

- ・ プロジェクト名の長さは半角24文字以内です。
- ・ プロジェクトパス全体の最大長は半角 171 文字です。
- ・ 使用できない文字

パス	:\. =;<>[]/? * ¥:\. =;<>[]/? *-
プロジェクト名	¥:\. =;<>[]/? *-

プロジェクトの保存先は
(Windows XP の場合)

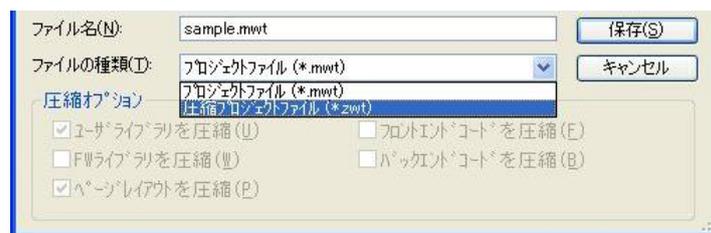
C:\¥Documents and Settings¥All Users¥Documents¥MULTIPROG¥Projects
に保存されます。

プロジェクトは複数のファイルで構成されており、プロジェクト名を名称とする mwt ファイルとプロジェクト名を名称とするフォルダに情報を格納しています。

そのため、プロジェクトをバックアップする際は、次で説明します「プロジェクトの保存(圧縮)」を行い、結果として作成される圧縮ファイル(.zwt)をバックアップします。

6.8. プロジェクトの保存(圧縮)

- 1) プロジェクトの保存の際にすべての構成情報を1つのファイルにまとめる(圧縮)ことができます。保存方法はメニューの[ファイル]－[プロジェクトを名前を付けて保存／圧縮]を選択しダイアログの「ファイルの種類」コンボボックスから「圧縮プロジェクトファイル」を選択します。
また、ここで圧縮したファイルを開く際は「プロジェクトを zwt ファイルのあるフォルダへ解凍しますか？」のようなダイアログが表示され圧縮ファイルを解凍します。



7. 通信ゲートウェイ

通信ゲートウェイは、プログラミングツールとコントローラ間の通信を行う機能のほかに次の独自機能を持ちます。



メニュー	サブメニュー	機能
ファイル		
	操作	START/STOP 操作 (起動時に自動的に START しています)
	オプション	(選択不可)
	設定	接続ポートの選択とそのパラメータ設定
	アプリケーションの終了	
操作		
	時刻設定	ターゲット(コントローラ)側の時計の設定
	LONWORKS	(選択不可)
	R3-I/O	(選択不可)
	POINT HISTORY	POINT_HISTORY Function Block の結果取得
	コントローラ設定	各種コントローラの設定
ヘルプ		
	バージョン情報	

注意

- 通信ゲートウェイの操作を行う際は MULTIPROG を終了してから実施されることを推奨します。MULTIPROG が稼働したままですと動作が遅くなったり、通信異常発生の原因となります。通信ゲートウェイを操作する際は、一旦 MULTIPROG を終了し、通信ゲートウェイの操作終了後に再度 MULTIPROG を起動してください。

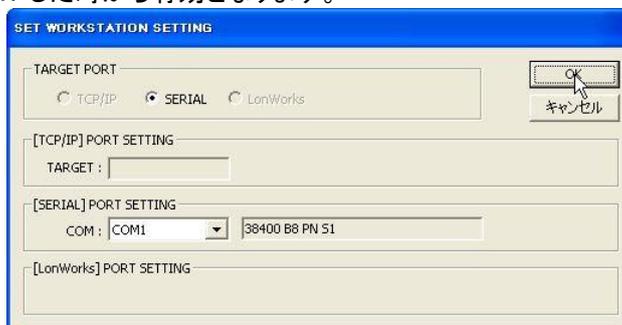
7.1. [ファイル]－[操作]

現在接続中の通信回線の状態及び手動での切断(STOP)と接続(START)を行うことができます。



7.2. [ファイル]—[設定]

パソコンとコントローラを接続する方法を設定します。ここでの設定は次回起動時か前述[ファイル]—[操作]で再度 START した時から有効となります。



■ SERIAL 接続

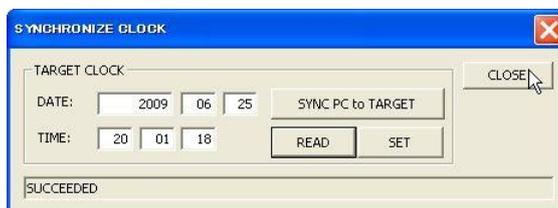
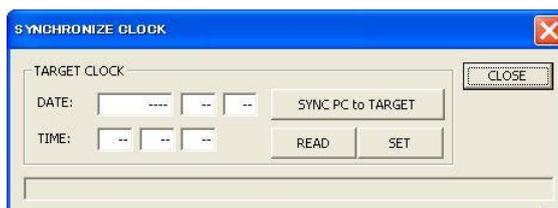
設定項目	設定範囲
COM ポート	COM1 ~ COM9 「パラメータ: 38400 B8 PN S1」

注意

- 設定変更
この設定操作を行うには通信を停止する必要があります。
手順は
 - 1) 通信の停止 [ファイル][操作]で「STOP」を押します。
 - 2) このダイアログでの設定を変更します。
 - 3) 通信を再開 [ファイル][操作]で「START」を押します。

7.3. [操作]—[時刻設定]

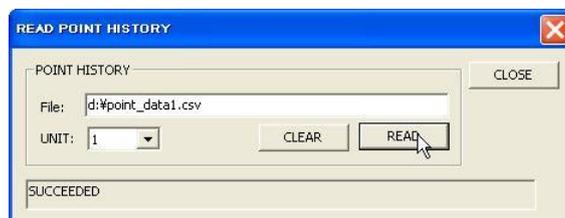
コントローラ側の時計の確認と設定を行うことができます。



ボタン	操作方法
READ	ボタンを押すとコントローラの現在時刻を取得し表示します。
SET	ダイアログの各テキストボックスに設定したい時刻情報を入力し、ボタンを押すことでコントローラに設定します。
SYNC PC to TARGET	ボタンを押すことでパソコンの現在時刻をコントローラに設定します。

7.4. [操作]—[POINT HISTORY]

コントローラ側のファンクションブロック(POINT_HISTORY)で収集した情報をパソコンに取り込みます。ただしコントローラ側で記録中のデータを読み出すと正確な読み出しができませんので、パソコンへの履歴データの取り込みは POINT_HISTORY ファンクションブロックが対象とする UNIT 番号に対して記録を行っていない間に行ってください。



ボタン	操作方法
CLEAR	削除したい UNIT 番号を指定してボタンを押すとコントローラで記憶している履歴情報を削除します。
READ	取得したい UNIT 番号と取得したデータを保存するパソコン側ファイル名を入力し、ボタンを押します。

■書き出されるファイルフォーマット

行	内容	備考
1	Unit: 1	対象のユニット番号 BA3-CL1 は 1～10 BA3-CL2 は 1～50 BA3-CL3 は 1～50
2	Title: [sample]	タイトル文字列
3	Total: 100	履歴最大件数
4	Count: 0	履歴データの件数
5	DATE, msec, Flags, viBoolValue1, viBoolValue2, viBoolValue3, viBoolValue4, viUSIntValue, viRealValue	データの表題 日付時刻 ミリ秒 [x 10ms] フラグ (システムで使用) 値 (BOOL) 値 (BOOL) 値 (BOOL) 値 (BOOL) 値 (USINT) 値 (REAL)
6 :	1970/01/01 00:00:00,0,0,0,0,0,0,0,0.000000	履歴データ (最大100件)

7.5. [操作]—[コントローラ設定]

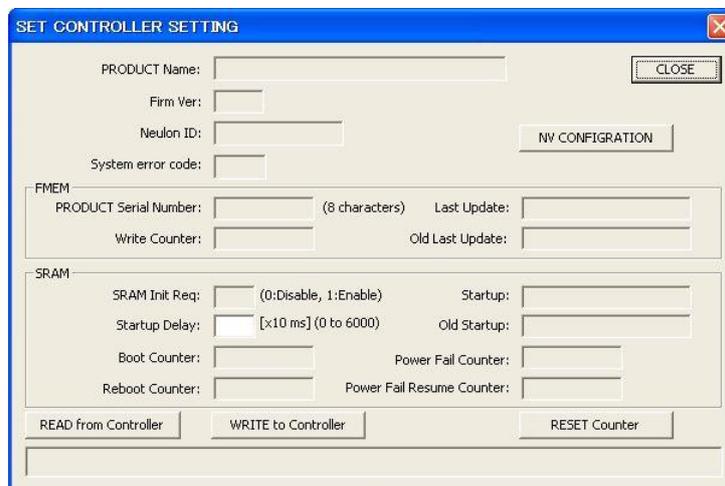
コントローラの各種設定項目の表示、設定ができます。

以下は「READ from Controller」ボタンを押しコントローラから取得した情報を表示した例。

設定項目	機能	詳細
Startup Delay	起動遅延時間 (BA3-CL2 の場合は 起動遅延に加え、起動時 LonWorks 出力変数現在値の送信を有効にし ます)	起動してから処理を開始するまでの時間を10ミリ秒単位で設定します。1以上の設定を行うと「起動 LonWorks 出力変数現在値の送信」機能も有効になります。 機能を有効にした場合は、起動時に処理開始前に遅延を行いその後 LonWorks 出力変数現在値をバインドされている相手に送信します。 0を設定することで機能無効にします。

7.6. [操作]—[コントローラ設定]—[ネットワーク変数構成] (BA3-CL3 のみ)

BA3CL3_NvConfigurator ツールで作成された「コントローラ ダウンロード ファイル」(拡張子.cl3nvcfg)をコントローラに書き込むことができます。



以下は「NV CONFIGURATION」ボタンを押し表示されるダイアログです。書き込むネットワーク変数構成ファイルを指定し「UPDATE」ボタンを押すとコントローラへの書き込みが開始されます。



設定項目	機能	詳細
File	ファイル名の入力	コントローラに書き込む BA3CL3_NvConfigurator で作成されたコントローラ ダウンロード ファイル (拡張子 .cl3nvcfg) を指定します。 例: f:¥BA3CL3.cl3nvcfg

ネットワーク変数構成の書き込みが完了しましたら、新しい構成を有効にするためにコントローラの電源再投入を再投入してください。構成変更後の初回の起動は、しばらく時間がかかります。正常に起動が行われると全面上部の RUN-LED が点灯します。

この操作は次の情報に影響を与えます。

- LonWorks コミッション、バインド情報はクリアされます。
- LonWorks NCI は初期値に設定されます。

そのため LonMaker ではコミッション操作から行う必要があります。

注意

- ・ ネットワーク変数構成の UPDATE 操作を行う際は下記の事項を厳守してください。
 - コントローラで動作しているソフトロジックプログラムは[停止]するか[リセット]操作でプログラムを削除してください。UPDATE 操作はソフトロジックプログラムが動作している状態で行うと期待しない動作

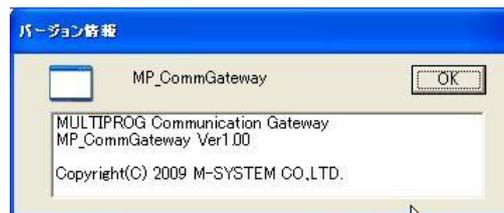
や重大な問題が発生する危険があります。

●UPDATE 操作を行うには MULTIPROG を終了する必要があります。 MULTIPROG が動作していると

UPDATE 処理が失敗することがあります。

7.7. [ヘルプ]—[バージョン情報]

本アプリケーションのバージョン情報を表示します。



8. ネットワーク変数構成ツール

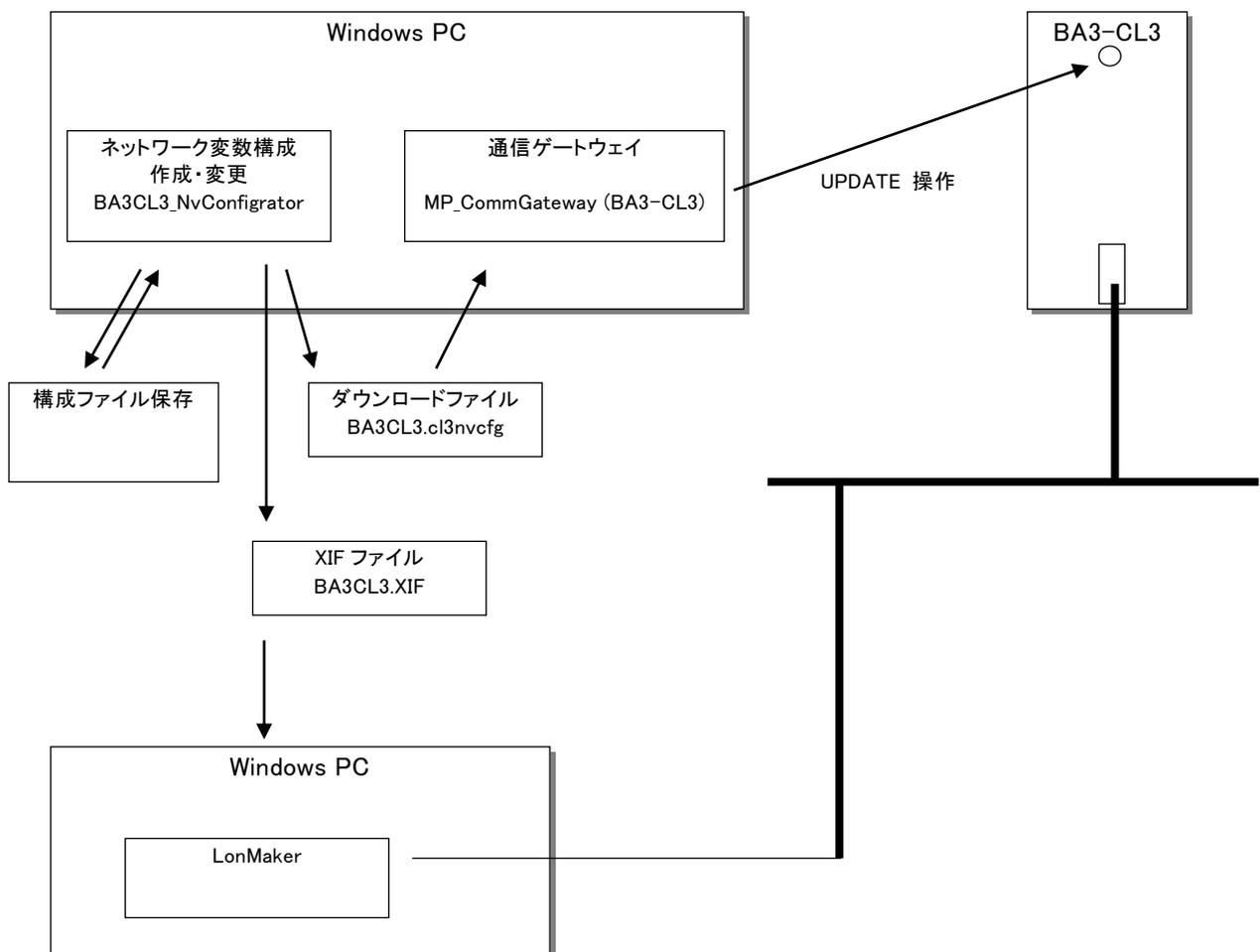
BA3-CL3 ではネットワーク変数の構成を変更できるようになっています。

注意

- このツールは BA3-CL3 専用です。BA3-CL1, BA3-CL2 ではネットワーク変数の構成を変更することはできません。

次の手順でネットワーク変数の構成を変更できます：

- 1) 専用のツール「BA3CL3_NvConfigurator」を使用してネットワーク変数の構成を行います。構成の終わりに[Create FILEs]ボタン操作でコントローラ ダウンロード ファイルを出力しておきます。
- 2) 通信ゲートウェイを操作して上記で出力したコントローラ ダウンロード ファイルをコントローラにダウンロード(書き込み)します。
- 3) ダウンロードされた新しい構成を有効にするために手でコントローラを再起動します。



8.1. 画面説明[NODE]タブ

起動すると次の画面が表示されます。基本情報は[NODE]タブ、ネットワーク変数定義は[Network Variables]タブ、ファンクショナルブロック定義は[Functional Blocks]タブで行います。

項目	機能
[ファイルから読込] ボタン	以前に保存したネットワーク変数構成を読み出します。指定はフォルダを選択します。
[ファイルへ保存] ボタン	現在のネットワーク変数構成を保存します。保存先はフォルダを指定します。
[NODE] タブ	
Title	タイトル ASCII 文字最大 30 文字 (*1)
Version	バージョン情報 ASCII 文字最大 30 文字 (*1)
Author	作成者情報 ASCII 文字最大 30 文字 (*1)
Company	会社情報 ASCII 文字最大 30 文字 (*1)
Description	説明文 ASCII 文字最大 30 文字 (*1)
ProgramID	ProgramID を指定します。最終桁は 0:出荷時構成で予約、1~9:A,B,C,D,E,F:ユーザ用
minSendTime	minSendTime を設定します。デフォルトは 0 範囲(0:Disable, 1 ~ 6553.4sec)
Location	ロケーション文字列 ASCII 文字最大 30 文字 (*1)
[出荷時変数構成] ボタン	現在の構成を破棄し出荷時と同じ変数構成を自動生成します。
[全てを NVO に構成] ボタン	現在の構成を破棄し全ての変数を NVO とした構成を自動生成します。
[全てを NVI に構成] ボタン	現在の構成を破棄し全ての変数を NVI とした構成を自動生成します。
[型混在構成] ボタン	現在の構成を破棄し規定の型構成で自動生成します。
XIF File	出力する XIF ファイルの名称を指定します。(拡張子は XIF)
Controller Download File	出力するコントローラ ダウンロード ファイルの名称が表示されます。ファイル名は[XIF File]で指定したファイル名に拡張子 cl3nvcfg が付加されます。
[Create FILEs] ボタン	XIF と Controller Download File を出力します。

*1)使用可能な文字は '0' ~ '9', 'a' ~ 'z', 'A' ~ 'Z', '_', '.', '#', '\$', '%', '&', '|', '(', ')', '[', ']'

注意

- 初期の XIF File に表示されていますパス("c:%BA3CL3.XIF")はご使用の OS により不適切な場合があります (Windows7 など)。[Create FILEs]ボタンを押す前に、XIF 及び Controller Download ファイルを出力したいパスに変更してください。

8.2. 画面説明[Network Variables]タブ

ネットワーク変数の定義を行います。

No	Dir	SNVT	service	Name	FbIdx	MemIdx	maxSendTime	maxRcvTime
4	1:nvo	8:SNVT_count	2:unackd	nv004	1	1	0	0
5	1:nvo	8:SNVT_count	2:unackd	nv005	2	1	0	0
6	1:nvo	8:SNVT_count	2:unackd	nv006	3	1	0	0
7	1:nvo	8:SNVT_count	2:unackd	nv007	4	1	0	0
8	1:nvo	8:SNVT_count	2:unackd	nv008	5	1	0	0
9	1:nvo	8:SNVT_count	2:unackd	nv009	6	1	0	0
10	1:nvo	8:SNVT_count	2:unackd	nv010	7	1	0	0
11	1:nvo	8:SNVT_count	2:unackd	nv011	8	1	0	0
12	0:nvi	8:SNVT_count	2:unackd	nv012	9	1	0	0
13	0:nvi	8:SNVT_count	2:unackd	nv013	10	1	0	0
14	0:nvi	8:SNVT_count	2:unackd	nv014	11	1	0	0
15	0:nvi	8:SNVT_count	2:unackd	nv015	12	1	0	0
16	0:nvi	8:SNVT_count	2:unackd	nv016	13	1	0	0
17	0:nvi	8:SNVT_count	2:unackd	nv017	14	1	0	0
18	0:nvi	8:SNVT_count	2:unackd	nv018	15	1	0	0
19	0:nvi	8:SNVT_count	2:unackd	nv019	16	1	0	0
20	2:nci	8:SNVT_count	2:unackd	nci001	0	0	0	0
21	2:nci	8:SNVT_count	2:unackd	nci002	0	0	0	0

項目	機能
NVs	定義済みネットワーク変数の数が表示されます。 最大は 254 です。 この表示にはシステムで予約しているネットワーク変数 No.0 ~3 (4 つの変数)も含まれています。 ユーザはシステム予約分を除いた残りの 250 変数を定義できます。
[挿入 1NV] ボタン	現在選択している行の直前に新しい行を挿入します。
[追加 1NV] ボタン	現在選択している行の直後に新しい行を挿入します。
[削除] ボタン	現在選択している1行あるいは複数行を削除します。
No	NV Index に対応します。 システムで0~3を使用しますので最初は4となります。
Dir	入出力の方向を指定します。 0:NVI, 1:NVO, 2:NCI を指定します。
SNVT	Standard Network Variable Type を指定します。 (*1)
service	初期の SERVICE が表示されます。 LonMaker の設定で運用されます。
Name	ネットワーク変数名を指定します。 ASCII 最大 16 文字 (*2)
FbIdx	属する Functional Block の Index を指定します。 範囲は 0~255 (*3)
MemIdx	属する Functional Block のメンバー番号を指定します。 0:指定なし, 1~255:メンバー (*3)
maxSendTime	maxSendTime の初期値を設定します。 0:未使用, 1~6553.4sec
maxRcvTime	maxRcvTime の初期値を設定します。 0:未使用, 1~6553.4sec

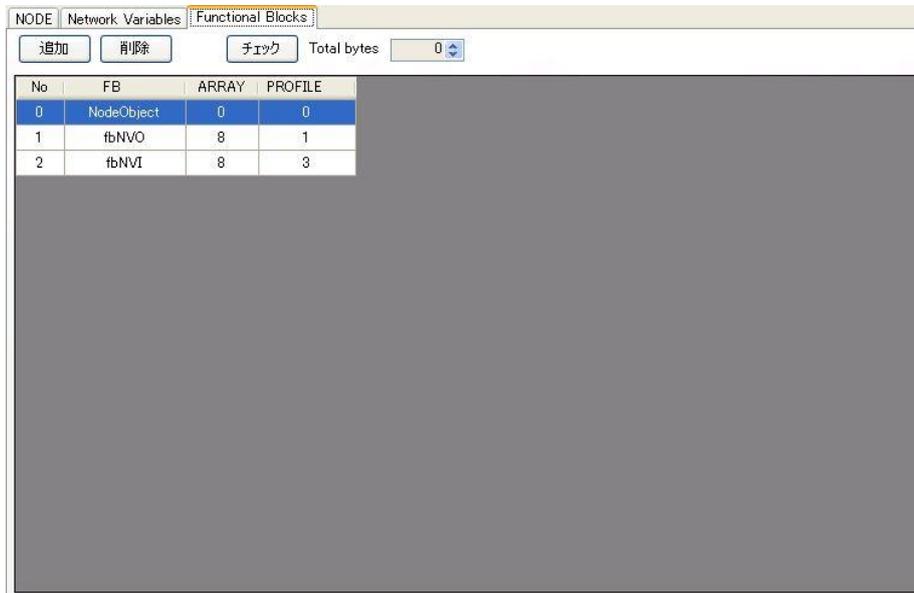
*1)設定可能な SNVT の種類は LonMark Resource File Ver.13 です。

*2)使用可能な文字は '0' ~ '9', 'a' ~ 'z', 'A' ~ 'Z', '_'

*3)NCI は FbIdx=0, MemIdx=0 と設定してください。 また、NVI,NVO では属する Functional Block の定義が必要となります。

8.3. 画面説明[Functional Blocks]タブ

ファンクショナル ブロックの定義を行います。



項目	機能
[追加] ボタン	現在選択している行の直後に新しい行を挿入します。
[削除] ボタン	現在選択している1行あるいは複数行を削除します。
[チェック] ボタン	現在の定義のダウンロードイメージにした場合のサイズを計算します。
Total bytes	[チェック]ボタンで算出されたサイズが表示されます。 最大 512 バイト
No	連番です。 Functional Block の Index とは対応していません。 (*1)
FB	ファンクショナルブロック名称。 ASCII 最大 16 文字 (*2)
ARRAY	配列の要素数を指定します。 0:非配列、2~255:配列要素数 要素数 1 の配列は非配列と同等なので 0:非配列 と設定してください。
PROFILE	プロファイル番号を設定します。 既定の番号は LonMark 協会から公開されています。

*1) [Network Variables]タブの FbIdx は、最初に定義された Functional Block が 0 となり ARRAY 数も含めて算出します。

例えば、上画面では

NodeObject	FbIdx: 0 (FbIdx の最初は 0)
fbNVO	FbIdx: 1 ~ 8 (配列の最初が 1)
fbNVI	FbIdx: 9 ~ 16 (配列の最初が 9)

となります。

*2)使用可能な文字は '0' ~ '9', 'a' ~ 'z', 'A' ~ 'Z', '_'

8.4. 制限事項

■ CP(Configuration Property)

構成プロパティは全て NodeObject に属しています。また、ネットワーク変数に関わる SCPTmaxSendTime は配列で実装されており、SCPTmaxSendTime[4] が nv004 に対応しています。この構成は変更できません。

■ NCI 変数

このネットワーク変数構成ツールでは、全ての NCI 定義は virtual functional block へ属します。個別のファンクションブロックへの割り当てはできません。

■ ファンクショナルブロック

本ツールで使用するファンクショナルブロックは、ご使用になる LonMaker が参照するリソースに定義されている必要があります。定義のないファンクショナルブロックやユーザ定義のファンクショナルブロックおよび変数型を使用すると正常な表示が行われな場合があります。新規のファンクショナルブロック、変数型の定義は Echelon 社から提供のリソースエディタで行うことができます。

BA3-CL3 で使用しますユーザ定義ファンクショナルブロックと変数型は CD 内の以下のフォルダに定義ファイルがあります。このファイルを適切な場所にコピーして使用してください。

```
lonworks¥types¥User¥m-system¥
  R000DF840Axxxxxx.ENU
  R000DF840Axxxxxx.fmt
  R000DF840Axxxxxx.fpt
  R000DF840Axxxxxx.ls
  R000DF840Axxxxxx.typ
```

LonMaker をデフォルトでインストールされているコンピュータの場合は次の場所にコピーします。

```
c:¥LonWorks¥types¥User¥
```

```
  m-system¥
```

←フォルダを作成して5つのファイルをコピーする

```
  R000DF840Axxxxxx.ENU
  R000DF840Axxxxxx.fmt
  R000DF840Axxxxxx.fpt
  R000DF840Axxxxxx.ls
  R000DF840Axxxxxx.typ
```

8.5. 出荷時のネットワーク変数構成に戻す方法

側面のディップスイッチ(SW2-2:SRAM clear)を ON にして電源投入します。しばらくすると全面上部の ERR-LED が点滅を始めます。その後ディップスイッチの状態を元に戻し(SW2-2 を OFF)で電源を再投入します。

この操作は次の内容がクリアあるいは出荷時設定に戻されます。

- ・ ソフトロジックプログラムはクリア
- ・ ネットワーク変数構成は出荷時設定
- ・ LonWorks コミッション、バインド情報はクリア
- ・ LonWorks NCI は初期値

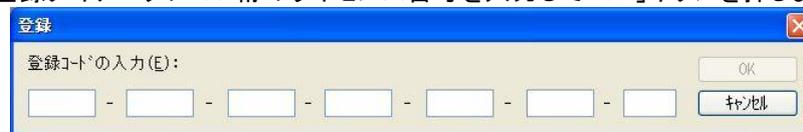
9. ライセンスの登録

プログラミングツールへライセンスを登録する方法を説明します。

- 1) ライセンスの登録はメニュー「？」を選択し、表示されるメニューの「登録」を選択します。
- 2)



- 3) 下記の登録ダイアログに 33 桁のライセンス番号を入力して「OK」ボタンを押します。



10. 付録

10.1. プロジェクトの構成要素

IEC61131-3 準拠のソフトウェアモデルは以下の要素で階層化されています。

- コンフィグレーション
 - リソース
 - タスク
- 1) コンフィグレーション
最上位のレベルにありコントローラに必要なすべての情報を提供します。
 - 2) リソース
コンフィグレーション内に位置しプログラムの実行に必要な機能を提供します。
リソースでは以下の定義ができます。
 - ・グローバル変数の定義
 - ・タスクの定義
 - 3) タスク
関連付けられたプログラムのタイムスケジュールを決定します。従って、プログラムの実行にはプログラムをタスクに関連付ける必要があります。

10.2. プログラム構成ユニット(POU)

Program Organization Unit は IEC61131-3 の言語構築ブロックです。それはプログラムコードを持つ小さい独立したソフトウェアの単位です。この POU 名はプロジェクト内でユニークでなければなりません。

POU は以下の3つのタイプがあります。

- ファンクション
- ファンクション ブロック
- プログラム

ファンクション

ファンクションとは複数の入力パラメータと1つの出力パラメータを持つ POU です。同じ値で呼び出されたファンクションの戻り値は常に同じ結果を戻します。この戻り値には基本データ型を指定できます。

ファンクション内では他のファンクションは呼び出せませんがファンクションブロックは呼び出せません。再帰呼び出しは使えません。

ファンクション ブロック

ファンクションブロックとは複数の入力／出力パラメータと内部メモリ(値の保存)を持つ POU です。同じ値で呼び出されたファンクションブロックの戻り値はその内部メモリの値により異なります。

ファンクションブロック内では他のファンクションブロックまたはファンクションを呼び出せません。再帰呼び出しは使えません。

プログラム

プログラムはファンクションやファンクションブロックを論理的に組み合わせた POU です。プログラムはファンクションブロックと同じく内部メモリを持つことができ、動作と使い方もファンクションブロックと似ています。プログラムを実行するにはタスクと関連付ける必要があります。

プログラム内ではファンクションとファンクションブロックを呼び出せません。再帰呼び出しは使えません。

10.3. ファンクションブロックのインスタンス

FB(ファンクションブロック)のコードはユニークなPOU名を持たせることで、別のPOUやプロジェクトからこのFBを呼び出すこと(再利用)ができます。FBはコードを一度だけ定義しておき、別のPOUからそのFBを呼び出す際にはそのFBのコピー(インスタンス)を作成します。このFBのインスタンスは呼び出されるとFBの内部メモリを呼び出されたインスタンスに割り当てます。これは各インスタンス毎に異なるメモリ領域を使用することを意味しており、このメモリ領域はカウンタやフリップフロップのように状態の記憶が必要な場合に利用されます。

10.4. ファンクションとファンクションブロックの違い

ファンクションとファンクションブロックの主な違いは

1) 内部メモリの存在

ファンクションには内部メモリ(変数の保存)を持たないので同じ入力値に対して常に同じ結果を生成します。対比的にファンクションブロックでは同じ入力値に対して繰り返し実行する場合に違う結果を生成することができます。例えばカウンタであれば前回の値を内部メモリが保持することで実行する毎に増加した値を結果として生成することができます。内部メモリにはファンクションブロック内の出力変数と内部変数が保持されます。

2) 出力パラメータの数

ファンクションでは単一の出力パラメータに結果を返すだけです。対比的にファンクションブロックでは複数の出力パラメータに対して結果を返すことができます。

10.5. 変数

変数はアクセス範囲により次の2種類が使用できます。

- ローカル変数
- グローバル変数

ローカル変数は宣言した POU 内でのみ使用でき、グローバル変数は対応するプロジェクトのどの POU からでも使用できます。

ローカル変数

使用する POU の「ローカル変数」ワークシートで宣言します。

グローバル変数

「リソース」の「グローバル変数」ワークシートで宣言します。

グローバル変数の宣言はこのワークシートにて VAR_GLOBAL として宣言します。

また、使用する POU では「ローカル変数」ワークシートにて VAR_EXTERNAL として宣言します。

10.6. データ型

データ型は変数が持てる値の種類を定めます。データ型の種類には次の3種類があります。

- 基本データ型
- 普遍的データ型
- ユーザ定義データ型

その説明に続いて各データ型のリテラルの形式について説明します。リテラルとは定数であるデータ型に与える値を表現するときに使います。

基本データ型

IEC61131-3 の基本データ型です。利用できる型を次に示します。

データ型	説明	サイズ	範囲
BOOL	ブール	1	TRUE, FALSE または 1, 0
SINT	単精度整数	8	-128 ~ 127
INT	整数	16	-32768 ~ 32767
DINT	倍精度整数	32	-2,147,483,648 ~ 2,147,483,647
USINT	符号なし単精度整数	8	0 ~ 255
UINT	符号なし整数	16	0 ~ 65535
UDINT	符号なし倍精度整数	32	0 ~ 4,294,967,295
REAL	実数	32	$\pm 1.18 \times 10^{-38} \sim \pm 3.40 \times 10^{38}$
TIME	持続時間	32	4,294,976,295ms ~ 4,294,976,295s
BYTE	8ビットのビット文字列	8	0x00 ~ 0xFF
WORD	16ビットのビット文字列	16	0x0000 ~ 0xFFFF
DWORD	32ビットのビット文字列	32	0x00000000 ~ 0xFFFFFFFF
STRING	文字列	80	

普遍的データ型

基本データ型のグループでできたデータ型です。例えば ANY_BIT または ANY_INT が該当します。

ユーザ定義データ型

異なるデータ型のグループで特定の目的のために集められ ARRAY または STRUCT として定義されます。

10.6.1. 整数型リテラル

整数型リテラルは 123 や -123 のように10進値を使って表現できます。

それ以外には

2進法	2#1111_1111 (10進の 255)
	2#0001_0000 (10進の 16)
8進法	8#377 (10進の 255)
	8#020 (10進の 16)
16進法	16#FF (10進の 255)
	16#10 (10進の 16)

10.6.2. 実数型リテラル

実数型リテラルは10進値を使って表現できます。整数型リテラルとは小数点があるかないかで区別できます。大きな値や小さな値を表現するには指数法を使うこともできます。

例えば 12.34 , +234.5 , -0.00123 , -1.23E-10 , 0.98E20 , 0.34e+12

10.6.3. 持続時間型リテラル

持続時間型リテラルには長短の2つの書式が用意されています。

書式で使われる文字は d=日、h=時間、m=分、s=秒、ms=ミリ秒

短い書式の例

T#12d3h4s	12日と3時間と4秒
T#3s45ms	3秒と45ミリ秒

長い書式の例

TIME#12d3h4s	12日と3時間と4秒
TIME#3s45ms	3秒と45ミリ秒

10.7. 予約語一覧

	記号		記号	
A	ABS	D	COLD_RESTART	
	ACOS		CONCAT	
	ADD		CONSTANT	
	ADD_T_T		CONTINUE	
	AND		COS	
	ANDN		CSV	
	ANY		CTD	
	ANY_BIT		CTU	
	ANY_DATE		CTUD	
	ANY_INT		D_BCD_TO_DINT	
	ANY_NUM		D_BCD_TO_DINT	
	ANY_REAL		D_BCD_TO_INT	
	ARRAY		D_BCD_TO_INT	
	ASIN		D_BCD_TO_SINT	
	AT		D_BCD_TO_SINT	
	ATAN		DATE	
	B		B_BCD_TO_DINT	DATE_AND_TIME
			B_BCD_TO_DINT	DELETE
B_BCD_TO_INT		DERIVAT		
B_BCD_TO_INT		DEVICE		
B_BCD_TO_SINT		DINT		
B_BCD_TO_SINT		DINT_TO_B_BCD		
BCD_TO_DINT		DINT_TO_B_BCD		
BOOL		DINT_TO_BCD		
BOOL_TO_BYTE		DINT_TO_BOOL		
BOOL_TO_DINT		DINT_TO_BYTE		
BOOL_TO_DWORD		DINT_TO_D_BCD		
BOOL_TO_INT		DINT_TO_D_BCD		
BOOL_TO_REAL		DINT_TO_DWORD		
BOOL_TO_SINT		DINT_TO_INT		
BOOL_TO_UDINT		DINT_TO_LREAL		
BOOL_TO_UINT		DINT_TO_REAL		
BOOL_TO_USINT		DINT_TO_SINT		
BOOL_TO_WORD		DINT_TO_STRING		
BOOL8		DINT_TO_TIME		
BY		DINT_TO_UDINT		
BYTE		DINT_TO_UINT		
BYTE_TO_BOOL		DINT_TO_USINT		
BYTE_TO_DINT		DINT_TO_W_BCD		
BYTE_TO_DWORD		DINT_TO_W_BCD		
BYTE_TO_INT		DINT_TO_WORD		
BYTE_TO_REAL		DIV		
BYTE_TO_SINT		DIV_T_A		
BYTE_TO_STRING		DIV_T_AI		
BYTE_TO_UDINT		DIV_T_AN		
BYTE_TO_UINT		DIV_T_R		
BYTE_TO_USINT		DO		
BYTE_TO_WORD		DT		
C		CAL	DWORD	
		CALC	DWORD_TO_BOOL	
	CALCN	DWORD_TO_BYTE		
	CASE	DWORD_TO_DINT		
	CLR_OUT	DWORD_TO_INT		

	記号		記号
	DWORD_TO_REAL		INT_TO_WORD
	DWORD_TO_SINT		INTEGRAL
	DWORD_TO_STRING	J	JMP
	DWORD_TO_UDINT		JMPC
	DWORD_TO_UINT		JMPCN
	DWORD_TO_USINT		LD
	DWORD_TO_WORD		LDN
E	ELEMENTARY		LE
	ELSE		LE_STRING
	ELSIF		LEFT
	EN		LEN
	END_CASE		LIMIT
	END_FOR		LIMIT_DINT
	END_IF		LIMIT_INT
	END_PROGRAM		LIMIT_REAL
	END_REPEAT		LIMIT_SINT
	END_SPECIFIC		LIMIT_STRING
	END_STRUCT		LINT
	END_TYPE		LN
	END_VAR		LOG
	END_WHILE		LREAL
	ENO		LREAL_TO_DINT
	EQ		LREAL_TO_REAL
	EQ_STRING		LREAL_TO_UDINT
	EXIT		LT
	EXP		LT_STRING
	EXPT		LWORD
F	F_TRIG		MAX
	FALSE		MAX_DINT
	FIND		MAX_INT
	FOR		MAX_REAL
	FPID		MAX_SINT
G	GE		MAX_STRING
	GE_STRING		MEMCPY
	GET_SYM		MEMSET
	GT		MID
	GT_STRING		MIN
H	HOT_RESTART	M	MIN_DINT
I	IF		MIN_INT
	IMEMCPY		MIN_REAL
	INSERT		MIN_SINT
	INT		MIN_STRING
	INT_TO_B_BCD		MOD
	INT_TO_B_BCD		MOVE
	INT_TO_BOOL		MUL
	INT_TO_BYTE		MUL_T_AI
	INT_TO_D_BCD		MUL_T_AN
	INT_TO_D_BCD		MUL_T_R
	INT_TO_DINT	N	NE
	INT_TO_DWORD		NE_STRING
	INT_TO_REAL		NEG
	INT_TO_SINT		NOT
	INT_TO_STRING	O	OF
	INT_TO_UDINT		OR
	INT_TO_UINT		ORN
	INT_TO_USINT	P	PDD
	INT_TO_W_BCD		PID

	記号		記号
	PLC_STOP		SHL_WORD
	PROGRAM		SHR
R	R		SHR_BYTE
	R_TRIG		SHR_DWORD
	RD_BOOL_BY_SYM		SHR_WORD
	RD_BYTE_BY_SYM		SIN
	RD_DINT_BY_SYM		SINT
	RD_DWORD_BY_SYM		SINT_TO_B_BCD
	RD_INPUT_GROUP		SINT_TO_B_BCD
	RD_INT_BY_SYM		SINT_TO_BOOL
	RD_SINT_BY_SYM		SINT_TO_BYTE
	RD_STRING_BY_SYM		SINT_TO_D_BCD
	RD_WORD_BY_SYM		SINT_TO_D_BCD
	REAL		SINT_TO_DINT
	REAL_TO_BOOL		SINT_TO_DWORD
	REAL_TO_BYTE		SINT_TO_INT
	REAL_TO_DINT		SINT_TO_REAL
	REAL_TO_DWORD		SINT_TO_STRING
	REAL_TO_INT		SINT_TO_UDINT
	REAL_TO_LREAL		SINT_TO_UINT
	REAL_TO_SINT		SINT_TO_USINT
	REAL_TO_STRING		SINT_TO_W_BCD
	REAL_TO_UDINT		SINT_TO_W_BCD
	REAL_TO_UINT		SINT_TO_WORD
	REAL_TO_USINT		SQRT
	REAL_TO_WORD		SR
	REPEAT		ST
	REPLACE		STN
	RET		STRING
	RETAIN		STRING_TO_BYTE
	RETC		STRING_TO_DINT
	RETCN		STRING_TO_DWORD
	RETURN		STRING_TO_INT
	RIGHT		STRING_TO_REAL
	ROL		STRING_TO_SINT
	ROL_BYTE		STRING_TO_TIME
	ROL_DWORD		STRING_TO_UDINT
	ROL_WORD		STRING_TO_UINT
	ROR		STRING_TO_USINT
	ROR_BYTE		STRING_TO_WORD
	ROR_DWORD		STRUCT
	ROR_WORD		SUB_T_T
	RS		TAN
S	S		THEN
	SEL		TIME
	SEL_BOOL		TIME_OF_DAY
	SEL_BYTE		TIME_TO_DINT
	SEL_DINT		TIME_TO_STRING
	SEL_DWORD		TO
	SEL_INT		TOD
	SEL_REAL		TOF
	SEL_SINT		TON
	SEL_STRING		TP
	SEL_WORD		TRUE
	SHL		TRUNC_DINT
	SHL_BYTE		TRUNC_INT
	SHL_DWORD		TRUNC_SINT
			T

	記号		記号	
	TYPE		WHILE	
U	UDINT		WORD	
	UDINT_TO_BOOL		WORD_TO_BOOL	
	UDINT_TO_BYTE		WORD_TO_BYTE	
	UDINT_TO_DINT		WORD_TO_DINT	
	UDINT_TO_DWORD		WORD_TO_DWORD	
	UDINT_TO_INT		WORD_TO_INT	
	UDINT_TO_LREAL		WORD_TO_REAL	
	UDINT_TO_REAL		WORD_TO_SINT	
	UDINT_TO_SINT		WORD_TO_STRING	
	UDINT_TO_STRING		WORD_TO_UDINT	
	UDINT_TO_UINT		WORD_TO_UINT	
	UDINT_TO_USINT		WORD_TO_USINT	
	UDINT_TO_WORD		WR_BOOL_BY_SYM	
	UINT		WR_BYTE_BY_SYM	
	UINT_TO_BOOL		WR_DINT_BY_SYM	
	UINT_TO_BYTE		WR_DWORD_BY_SYM	
	UINT_TO_DINT		WR_INT_BY_SYM	
	UINT_TO_DWORD		WR_OUTPUT_GROUP	
	UINT_TO_INT		WR_SINT_BY_SYM	
	UINT_TO_REAL		WR_STRING_BY_SYM	
	UINT_TO_SINT		WR_WORD_BY_SYM	
	UINT_TO_STRING			
	UINT_TO_UDINT		X	
	UINT_TO_USINT		XOR	
	UINT_TO_WORD		XORN	
	ULINT			
	UNTIL			
	USINT			
	USINT_TO_BOOL			
	USINT_TO_BYTE			
	USINT_TO_DINT			
	USINT_TO_DWORD			
	USINT_TO_INT			
	USINT_TO_REAL			
	USINT_TO_SINT			
	USINT_TO_STRING			
	USINT_TO_UDINT			
	USINT_TO_UINT			
	USINT_TO_WORD			
	V	VAR		
		VAR_EXTERNAL		
		VAR_EXTERNAL_FB		
VAR_EXTERNAL_PG				
VAR_GLOBAL				
VAR_GLOBAL_FB				
VAR_GLOBAL_PG				
VAR_IN_OUT				
VAR_INPUT				
VAR_OUTPUT				
W	W_BCD_TO_DINT			
	W_BCD_TO_DINT			
	W_BCD_TO_INT			
	W_BCD_TO_INT			
	W_BCD_TO_SINT			
	W_BCD_TO_SINT			
	WARM_RESTART			

■これ以外に拡張ファンクションブロックの名称及びテンプレート(「BA3-CL1 Template」, 「BA3-CL2 Template」, 「BA3-CL3 Template」)で作成されますグローバル変数名、POU名が対象となります。

10.8. テンプレート(BA3-CL1,BA3-CL2,BA3-CL3 Template)

テンプレート(BA3-CL1 Template, BA3-CL2 Template または BA3-CL3 Template)で構築されますプロジェクトを説明します。

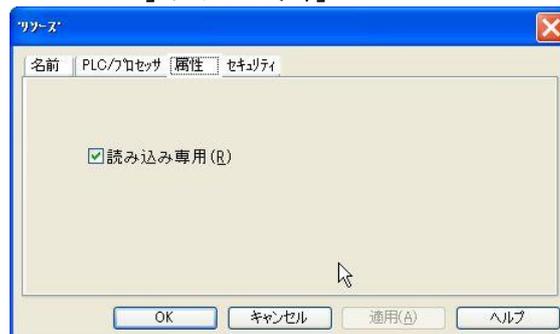
属性が「読み取り専用」に設定されています項目の内容は変更しないでお使い下さい。

(例外として「Global_Variable」はテンプレートで用意されています変数を変更削除、変更しないよう注意しユーザの変数を追加することができます)

① 構築された「プロジェクトツリー」



② 構築された「リソース:BA3-CL1」の「プロパティ」



③ 構築された「リソース:BA3-CL1」の「設定」

BA3-CL1のリソース設定

通信

タイプ: TCP/IP

パラメータ: 127.0.0.1

バージョン

ビルド設定: BA3-CL1 v1.0.0 (Core: 2.1.1)

ビルド設定動作の更新:

自動更新(△)

更新前に確認(ⓑ)

更新しない(Ⓝ)

オンライン更新

周期: 10 ms (範囲 0.60000)

コンピュータオプション

スタックチェック

記列境界チェック

OK(ⓐ) キャンセル(ⓑ) ヘルプ(Ⓝ)

④ 構築された「タスク:DEFAULT」の「設定」

タスクのタスク設定

イベント(E):

フライオブライ(P):

ウォッチドック時間(W): 500 ms ウォッチドックを有効にする(N)

OK キャンセル

10.9. 旧ファームウェアへの対応

最新のプログラミングツールは、接続する機器の最新ファームウェア(本体内のソフトウェア)に対応していません。新規作成の場合は最新のファームウェアに対応したライブラリを自動的に選択しています。しかし、納入済みの機器やリプレースなどで旧ファームウェアを使用する場合があります。

ここでは、旧ファームウェアの機種を使用する場合の操作方法を説明します。

注意

使用しているファームウェアに適合していない設定の場合は、予測できない動作になることが有ります。この章での操作は、上級者向けであり影響度を理解のうえ実施してください。

10.9.1. 接続機器のファームウェアバージョンの確認方法

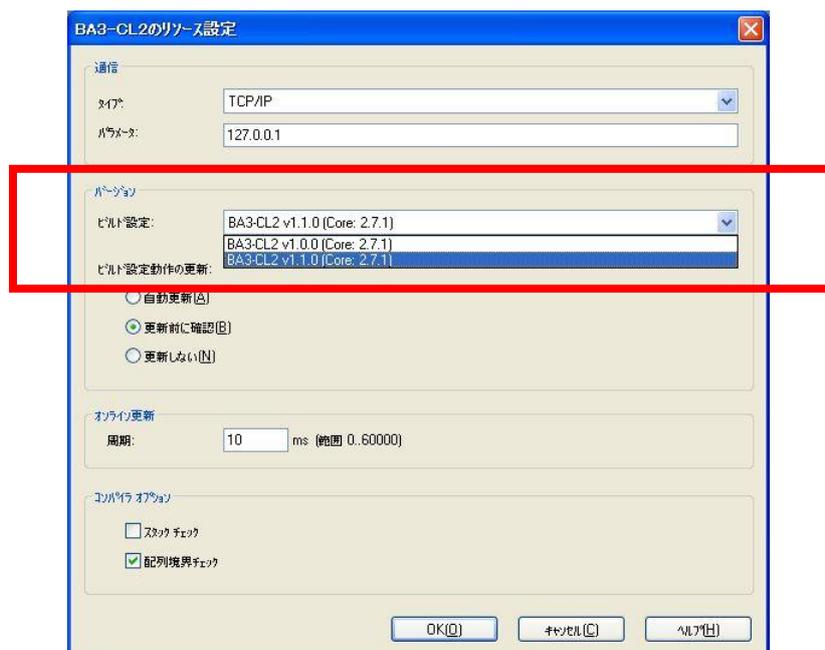
- ① 接続の準備
接続する機種に合ったテンプレートでプロジェクトを作成します。
- ② 機器に通信ケーブルを接続して、プログラミングツールのプロジェクトコントロールダイアログを表示します。そのダイアログで「情報」ボタンを押します。



10.9.2. 現在のプロジェクトを機器のファームウェアバージョンに合わせる方法

その機器に合わせたプログラムを作成するためには、ファームウェアバージョンに適合したリソースをプロジェクトに設定する必要があります。

- ① プロジェクトツリーウィンドウのリソースを選択します。初期は「読み取り専用」が設定されているので「プロパティ」の「属性」で「読み取り専用」のチェックを外しておきます。
- ② リソースの「設定」を選択すると次のダイアログが表示されます。ビルド設定をファームウェアバージョンに適合した適切な設定に変更します。



適合ファームウェアバージョン	ビルド設定
V1.0.0	BA3-CL2 v1.0.0 (Core: 2.7.1)
V1.1.0	BA3-CL2 v1.1.0 (Core: 2.7.1)

■ 適合しない設定で機器にダウンロードした場合に表示される警告ダイアログ



- ③ プロジェクトをコンパイルしてエラーがないことを確認しておきます。

10.9.3. 「このリソースのビルド設定が整合していません」ダイアログが表示された場合

プロジェクトを読み込んだ際、次のダイアログが表示された場合には、10.9.1「接続機器のファームウェアバージョンの確認方法」により使用機器のバージョンを確認いただき、10.9.2「現在のプロジェクトを機器のファームウェアバージョンに合わせる方法」に従い適合するリソースをプロジェクトに設定してください。

