Panasonic®

IEC61131-3準拠
プログラマブルコントローラFPシリーズ用
ツールソフト

Control FPWIN Pro
導入ガイド

Control FPWIN Pro導入ガイド
ARCT1F405-2 '06·01月

http://www.nais-j.com/plc/
Panasonic... **NAIS** の新しいブランドです。

松下電工

⚠ 警告

本商品に添付されているディスクは、オーディオ用の CD プレーヤやパソコンのスピーカで絶対に再生しないでください。
大音量により、耳に傷害を与えたり、スピーカーを破損する恐れがあります。

著作権及び商品登録に関する記述

はじめに

このたびは、FPWIN Proをお買い上げいただき、
誠にありがとうございます。
このガイドは､FPWIN Proをはじめて使う方に､ソフトのセットアップや操作概要をお知らせするための「導入ガイド」として作成されています。十分に内容を理解していただいたうえ正しくご使用くださいますようお願い申し上げます。

なお、使用方法の詳細については、ヘルプをご覧ください。
命令の詳細については､「命令語マニュアル」をご覧ください。

●お願い

このマニュアルの内容に関しては万全を期しておりますが、ご不審な点や誤りなどお気付きの点がございましたらお手数ですが弊社までご連絡ください。

FPWIN Pro 導入ガイド　目次

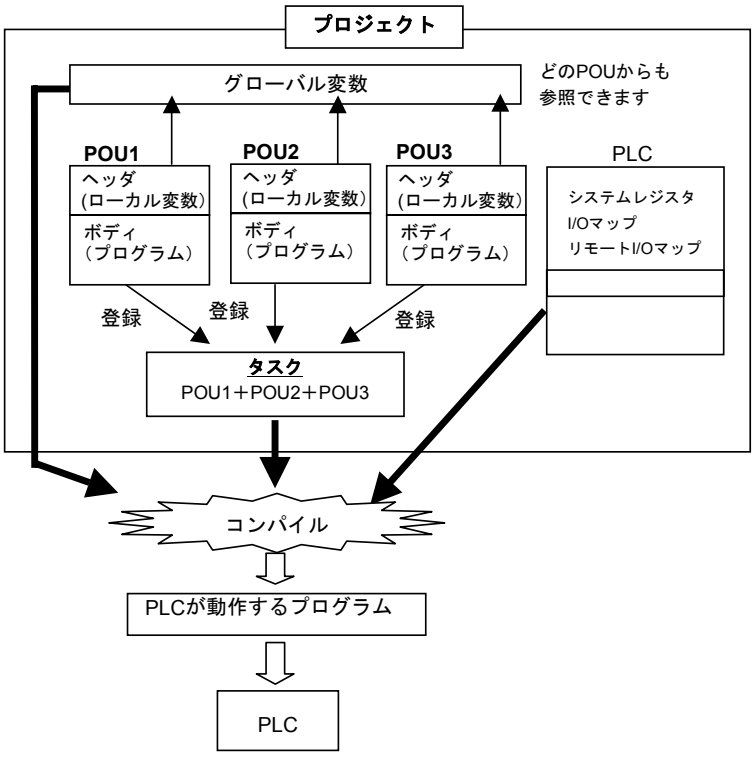
FPWIN Pro 基本編

FPWIN Pro 基本編

1章

FPWIN Proの概要

1.1 FPWIN Proの概念



■ いくつものプログラムを組み合わせて全体のプログラムを構成できます。

・ FPWIN Proでは、FPWIN GRと異なり、いくつものプログラムを組み合わせて全体のプログラムとすることができます。この内、１つのプログラムを**POU**と呼び、後述する５つの言語から選択して作成することができます。

・ **POU**は**タスク**に登録することにより、全体プログラムの１つとして認識されます。

■ **変数**を使ってプログラムできます。

・ 直接デバイスを指定する（X0、Y1、DT100などを使う）の代わりに、変数を使ったプログラムが作成できます。

・ POU内だけで有効な変数を**ローカル変数**、どのPOUからも参照できる変数を**グローバル変数**と呼びます。

・ 内部的なワークエリアとしてリレーやレジスタを使用する場合は、ローカル変数として使用すると、コンパイル時に自動的に適当なリレーやレジスタに割り当てられます。他のプログラムで再利用したいPOUを作成する場合には、直接デバイスを指定するより、ローカル変数を使用してプログラムを作成すると便利です。

・ 外部入力(X)、外部出力(Y)等、物理的に結線されるデバイスを変数として使用する場合には、必ずグローバル変数として登録する必要があります。

■ 作成したプログラムを**プロジェクト**として管理します。

・ いくつものPOU(プログラム)を登録したタスク、グローバル変数、システムレジスタ、I/Oマップなど、プログラムに関わる全情報をプロジェクトとして管理します。

・ コンピュータに保存するときは、ファイルではなく、フォルダとして管理されます。

■ 作成したプロジェクトは、高級言語プログラムのようにコンパイルが必要です。

・ 作成したプロジェクトは、コンパイルすることによりPLCが解読できるコードに変換されます。コンパイルする前のソースコード（ラダー図等）もPLC内に保管したい場合は、PLCが汎用メモリやコメントメモリを持っている必要があります。

・ 従ってFP0やFP1にソースコードを保管することはできません。コンピュータのディスク上での管理が必要です。

■ IEC61131-3規格に準拠しています。（後述）

・ ５つの言語(LD, FBD,SFC, ST, IL)に対応しています。

■ インポート機能により、過去の資産が活用できます。

・ FPWIN GRを使って作成されたプログラムも、インポート機能により活用できます。

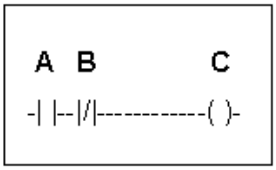
1.2 IEC61131-3規格について

この規格は、以下の様々な目的を実現するために、IEC（International Electro-technical Commission）によって策定されました。

・国際的レベルで標準化されたプログラム
・PLC機種に依存しないプログラム
・構造化プログラミングにより、解りやすく保守しやすいプログラム
・再利用性の高い、部品化されたプログラム
・厳密な文法チェックによる誤りの少ないプログラム

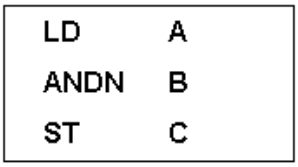
■ IEC61131-3規格では、以下の5つの言語が提供されています。

・LD（ラダーダイアグラム）
従来のラダープログラム。

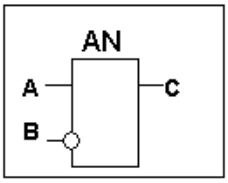


w

・IL（インストラクションリスト）
アセンブラに似た低級言語。



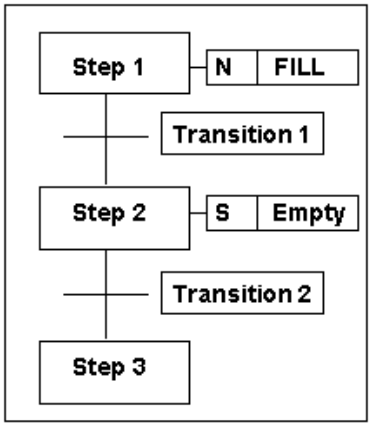
・FBD（ファンクションブロックダイアグラム）
ブロック間を配線する形式で回路図に似た表記法。



・ST（ストラクチャードテキスト）
数値演算等に向いた高級言語。
書式はPASCALに類似。

```
FOR i:=0 TO 100 DO
  SUM:=SUM+1;
END_FOR;
```

・SFC（シーケンシャルファンクションチャート）
プログラムの順序を図式的に表現できるグラフィック言語。
各ステップやトランジションの内部は、上記のLD、IL、STの各言語で記述可能。



2章

インストールと起動

2.1 使用動作環境

FPWIN Proをインストールするにあたっては、以下の環境をおすすめします。
OS：Windows95 OSR2（Ver. 4.00.950B以上）/98/Me/NT（Ver. 4.0以上）/2000/Xp

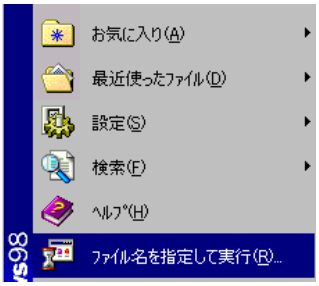
必要ハードディスク容量：	100MB以上
推奨CPU：	Pentium 100MHz以上
推奨搭載メモリ：	128MB以上（OSによる）
推奨画面解像度：	1024×768以上
推奨表示色：	High Color（16ビット）以上
対応PLC機種：	FP-e/FP0/FP Σ/FP1/FP-M/FP2/FP2SH/FP3/FP-C/FP10S/FP10SH/FP-X

2.2 インストールを実行する

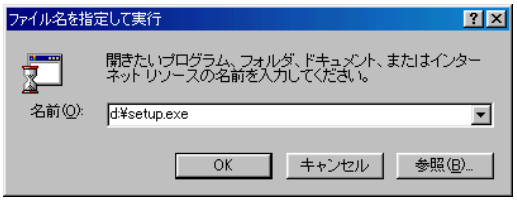
FPWIN Proを、ご使用のパソコンにインストールします。インストールは以下の手順で行ってください。

■操作手順

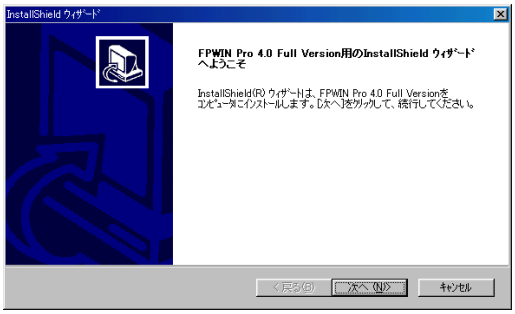
1. **起動中のアプリケーションを全て終了します。**
   （ウィルス検出ソフトなどの常駐ソフトも、全て終了してください。）
2. **セットアップCDを、パソコンのCDドライブにセットしてください。**
3. **「ファイル名を指定して実行」を選択します。**
   画面左下の「スタート」ボタンをクリックすると表示されるWindowsメニューから「ファイル名を指定して実行」を選択してください。



4. **実行ファイル名を入力します。**
   「ファイル名を指定して実行」を選択すると以下のダイアログボックスが表示されますので、
   d:¥setup.exe
   と入力し「OK」ボタンをクリックしてください。
   （この例では、CDドライブがDであると仮定して説明しています。）



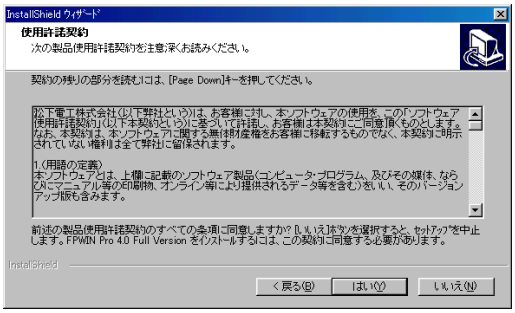
5. **確認メッセージが表示されます。**
   セットアッププログラムが起動すると、確認のダイアログボックスが表示されますので、内容を確認して「次へ(N)」ボタンをクリックしてください。中止する場合は「キャンセル」をクリックしてください。



6. ライセンス契約を確認します。

ライセンス契約の確認ダイアログボックスが表示されます。表示されているライセンス契約の、全ての条項に同意された場合は「はい(Y)」ボタンをクリックしてください。セットアップが開始されます。

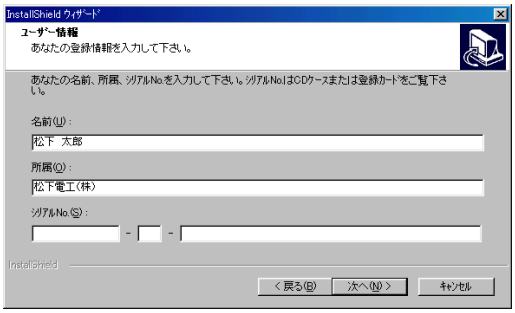
「いいえ(N)」が選択されると、FPWIN Proのセットアップは中止されます。



7. ユーザー情報を登録します。

ユーザーの情報ダイアログボックスが表示されますので、「名前」と「所属」、「シリアルNo.」を入力し、「次へ(N)」ボタンをクリックしてください。

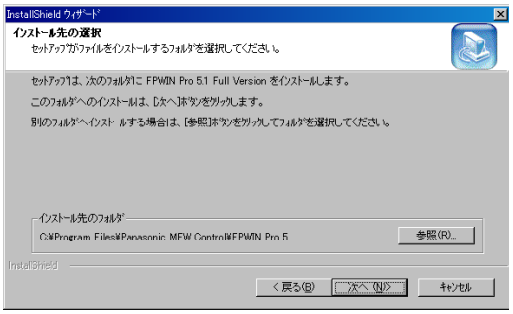
「シリアルNo.」は、FPWIN Proのパッケージに同梱のユーザカードに記述されていますので正しく入力してください。ここで入力した内容は、バージョン情報等で確認できます。



8. インストール先を選択します。

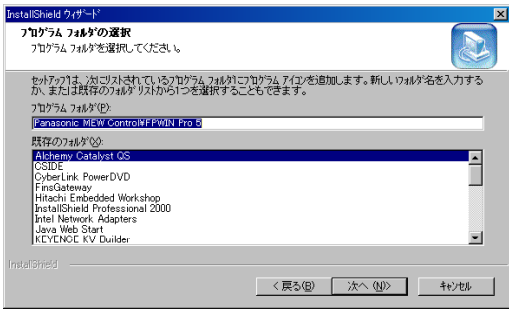
インストール先の確認ダイアログボックスが表示されます。表示されているフォルダにインストールする場合は「次へ(N)」ボタンをクリックしてください。

インストール先を変更したい場合は、「参照(R)」ボタンをクリックしてフォルダを指定してください。



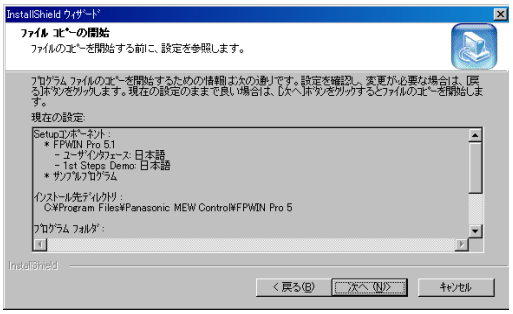
9. プログラムフォルダを選択します。

プログラムフォルダ名の確認ダイアログボックスが表示されます。表示されているフォルダ名で良ければ「次へ(N)」ボタンをクリックしてください。
変更する場合は「プログラムフォルダ」の部分を直接編集してください。

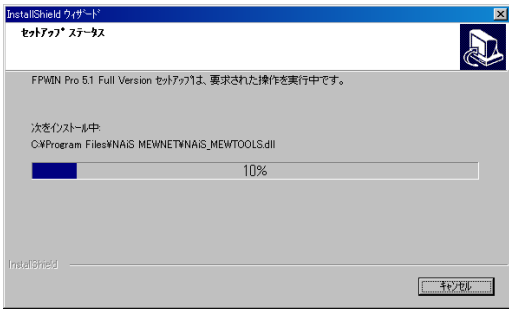


10. 設定を確認した後、インストールが開始されます。

これまでに設定した内容を確認するためのダイアログボックスが表示されますので、もう一度設定を確認してください。設定を変更したい場合は、「戻る(B)」ボタンで該当する設定項目の画面まで戻って設定をやり直してください。



設定に問題がなければ「次へ(N)」ボタンをクリックしてください。FPWIN Proのインストールが実行されます。



11. インストールの完了

必要なファイルのコピーが正常に終了したら、以下のダイアログボックスが表示されます。「完了」をクリックすると、インストールが終了します。

インストールが完了すると、デスクトップ上にショートカットアイコンが自動で生成されます。

2.3 FPWIN Proを起動する

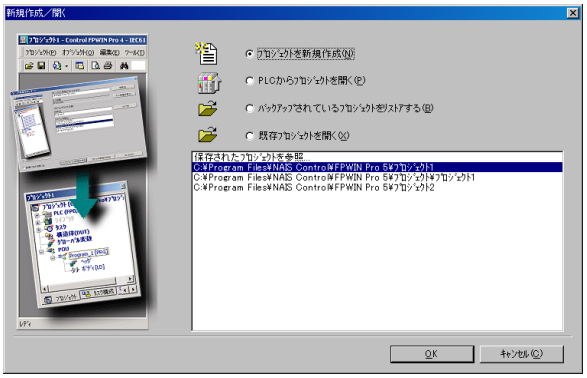
起動の手順

次のいずれかの方法で、FPWIN Proを起動してください。

・Windowsのスタートメニューから起動する。

・デスクトップ上のショートカットアイコンをダブルクリックして起動する。

起動すると、以下のダイアログボックスが表示されます。



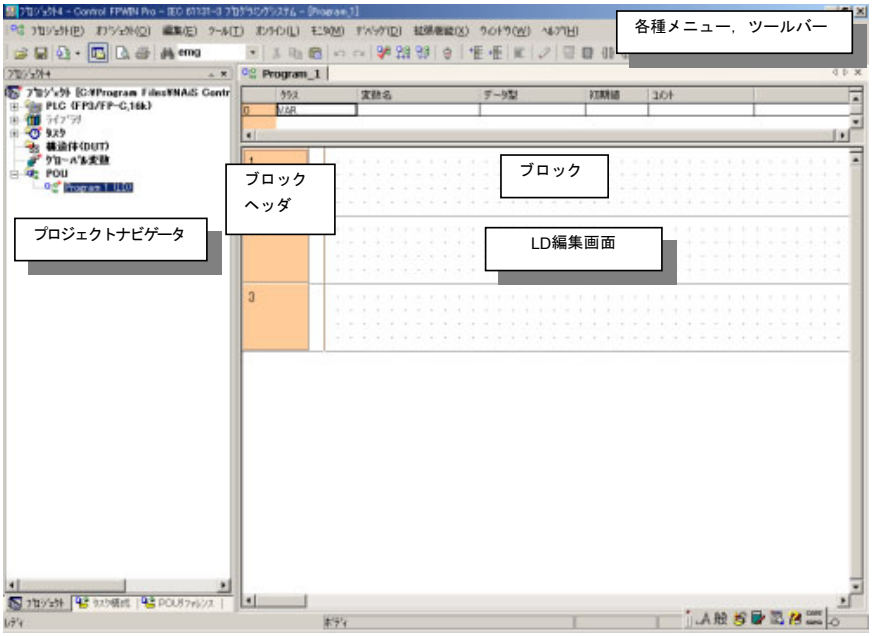
ここで新規にプロジェクトを作成するか、保存してあるプロジェクトを開くか、PLCからアップロードするかの選択を行います。

3章

プロジェクトについて

3.1　プロジェクトの概念

FPWIN Proでは、単にプログラムだけを管理するのではなく、付随する全ての情報を含めてプロジェクトとして管理します。
プロジェクトは、システムレジスタ、使用可能な命令一覧（ライブラリ）、プログラムの実行順序を管理するタスク、定義された変数を管理する変数テーブル、実際の動作プログラム等で構成されています。
これらの情報は、FPWIN Proの「プロジェクトナビゲータ」で一覧表示されています。
FPWIN Proで作成されたプログラムをはじめ、プロジェクトの内容全てを保存する場合は、メニューの「プロジェクトの保存」を選択します。

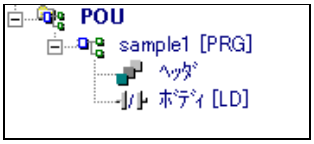


3.2 プロジェクトナビゲータの構成

プロジェクトナビゲータは、以下の要素で構成されています。

3.2.1 POU

POUは、Program Organization Unitの略で、プログラムの構成単位という意味になります。
ここで実際に作成するプログラムを登録します。



Ver.4

Ver.5

■POU の概念

FPWIN Proで作成されたプログラムは、全てPOUとして管理されます。
新しくプログラムを作成することを、FPWIN Proでは「POUの新規作成」と言います。
POUを新規作成するときには、まず、そのPOU内で使用する言語を選択します。
言語は、LD、FBD、SFC、ST、ILのいずれかを指定します。
プログラム作成途中に使用言語を変更することはできません。
POUは複数個作成でき、これらを一つにまとめて、もしくは必要な部分だけを選択して一つのプログラムとして完成させることができます。

■ヘッダ

ヘッダは、POU内で使用する変数を登録するエリアです。
ここに登録された変数のみ、そのPOU内で使うことができます。

■ボディ

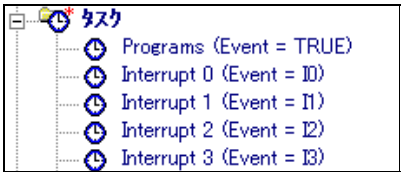
プログラム編集エリアです。ここに、実際の動作プログラムを記述します。
LD、SFC、STなど、POUの使用言語に応じた編集画面が表示されます。

v 5

Ver.4のときは、ヘッダとボディは別ウィンドウで表示されていましたが、
Ver.5では、ヘッダとボディが１つのウィンドウになり、セパレータで別れています。
したがって、Ver.5では、ナビゲータ上にヘッダとボディは表示されません。

3.2.2 タスク

FPWIN Proは、様々なプログラミング言語をサポートしています。
これらを用途に応じて使い分け、プログラムをいくつかのPOUに分けて作成することができます。
この時、それぞれのPOUをどの順番で実行させるかを登録するのがタスクです。
この導入ガイドでは、タスクの中でも特に重要な「Program」についてのみ解説します。
タスクには他にも、割り込みプログラムなどを登録できますが、これに関しての詳細はヘルプを参照してください。
タスクの中の「Program」に作成したPOUを登録することによって、そのPOUは変換対象となります。
POUを作成してもタスクに登録しなければ、そのPOUは変換されず実行もされません。
通常、POUを新規に作成すると、Programタスクへ自動的に登録されます。
POUは、タスクに登録された順番に実行されますが、この順番は後から変更することもできます。



3.2.3 グローバル変数

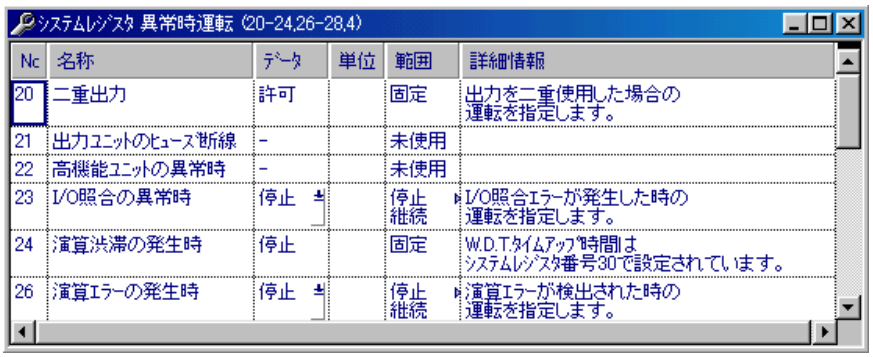
グローバル変数に関しては、「3.3 変数について」で解説します。

3.2.4 PLC

PLC動作用の各種環境設定です。

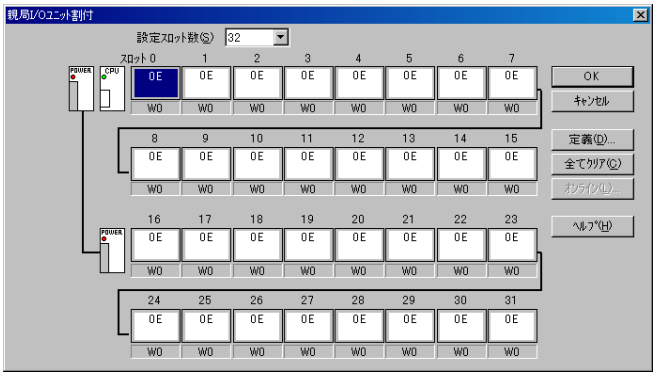
■システムレジスタ

エラー発生時の動作選択や通信ポートの設定、内部メモリの保持エリア設定など、PLC動作用の各種設定を行います。
システムレジスタは、PLC機種によって設定内容が異なる部分があります。



■I/O マップ

PLCのマザーボード上で、I/Oユニットをどのように割り付けるかの設定を行います。



■リモート I/O マップ

PLCのマザーボード上で、リモートI/OネットワークMEWNET-Fをどのように割り付けるかの設定を行います。

■プログラム

プロジェクトをPLCへダウンロードしたときに、実際にPLCへ格納されるプログラムコードが表示されます。

3.2.5 構造体（DUT）

外部機器との送受信データや位置決め制御用のデータテーブルなどデータを固まりとして扱う場合があります。
FPWIN Proでは、このようなデータのブロックを変数のグループとしてあらかじめ定義することができ「構造体」という名称で呼びます。

3.2.6 ライブラリ

ライブラリとは、プロジェクト内で使用できる応用命令一覧のことを示します。
FPWIN Proでは従来からのFPシリーズPLCの応用命令の他に、IEC命令などもサポートしています。
PLC機種によって使用できる命令が異なりますので、ライブラリの表示も異なります。
また、ユーザ側で作成されたライブラリを登録することもできます。
ライブラリには、以下のタイプがあります。

FP Library：	FPシリーズでサポートされている応用命令が登録されています。
FP Pulsed Library：	FPシリーズ応用命令のうち、微分実行型命令が登録されています。
FP Tool Library：	FPシリーズの応用命令には、オペランドにアドレスやサイズを指定するものがあります。FP Tool Libraryには、FPWIN Pro上で使用する変数をこれらの値に変換するためのファンクションが収められています。
IEC Standard Lib：	IECで標準的に定義された命令が登録されています。

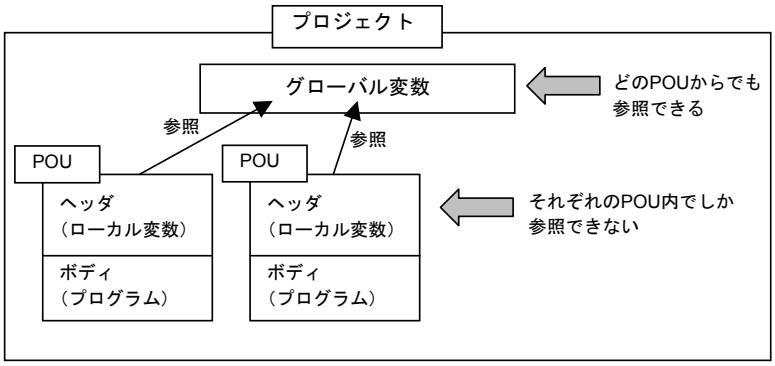
◆ここがポイント!

IEC命令は、IEC規格61131-3で定義された命令です。FPWIN Proでは、これらの命令をPLC本体が使用できる命令に変換しダウンロードすることにより、これらの命令をサポートしています。従って、IEC命令自体は、直接的にはサポートされていません。

3.3 変数について

3.3.1 グローバル変数とローカル変数の違い

グローバル変数は、どのPOUからでも参照できる変数です。プロジェクトナビゲータでも、POUの上位に位置しています。
これに対してローカル変数は、そのPOU内でしか参照できない変数です。



3.3.2 グローバル変数の定義

グローバル変数では、以下の項目を設定します。

■必須設定項目

・クラス（変数のクラス）：　通常の変数、値が保持される変数、値が固定の変数など。U

・変数名：　変数の名前です。半角英数字で100文字以内。（ひらがな、カタカナ、漢字は入力できません。）

・タイプ：　変数の型を設定します。（BOOL、INT、WORDなど）

・初期値：　変数の初期値を設定します。自動的に設定されますが、変更も可能です。

v 5

Ver.4のときは、変数名は半角英数字で16文字以内でしたが、Ver.5では、変数名は半角英数字で100文字以内と拡張されています。

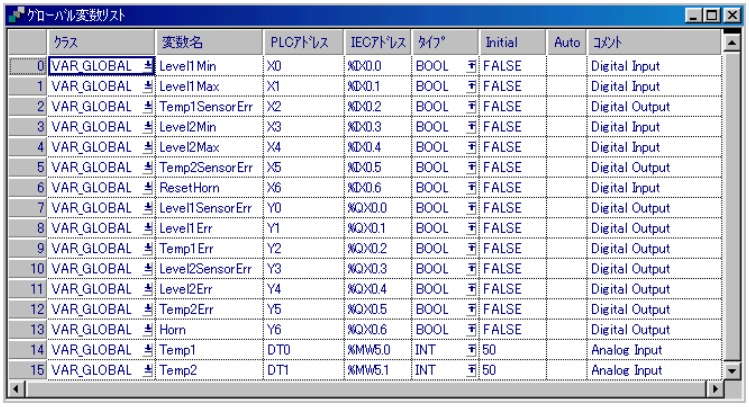
■任意設定項目

・PLCアドレス：　PLCのデバイスに割り当てる場合は設定します。設定されない場合はFPWIN Proによって自動的に割り当てられます。

・IECアドレス：　プログラム内で自動的に設定されるアドレスです。ユーザ側で入力する必要はありません。

・Autoextern（外部変数自動登録）：　ここをチェックすると、設定された変数が各POUのヘッダに自動で登録されます。

・コメント：　変数にコメントを書き込みます。ボディには表示されません。ひらがな、カタカナ、漢字を使って書くことも可能です。


グローバル変数リスト

	クラス	変数名	PLCアドレス	IECアドレス	タイプ	Initial	Auto	コメント
0	VAR_GLOBAL	Level1 Min	X0	%IX0.0	BOOL	FALSE		Digital Input
1	VAR_GLOBAL	Level1 Max	X1	%IX0.1	BOOL	FALSE		Digital Input
2	VAR_GLOBAL	Temp1SensorErr	X2	%IX0.2	BOOL	FALSE		Digital Output
3	VAR_GLOBAL	Level2Min	X3	%IX0.3	BOOL	FALSE		Digital Input
4	VAR_GLOBAL	Level2Max	X4	%IX0.4	BOOL	FALSE		Digital Input
5	VAR_GLOBAL	Temp2SensorErr	X5	%IX0.5	BOOL	FALSE		Digital Output
6	VAR_GLOBAL	ResetHorn	X6	%IX0.6	BOOL	FALSE		Digital Input
7	VAR_GLOBAL	Level1SensorErr	Y0	%QX0.0	BOOL	FALSE		Digital Output
8	VAR_GLOBAL	Level1Err	Y1	%QX0.1	BOOL	FALSE		Digital Output
9	VAR_GLOBAL	Temp1Err	Y2	%QX0.2	BOOL	FALSE		Digital Output
10	VAR_GLOBAL	Level2SensorErr	Y3	%QX0.3	BOOL	FALSE		Digital Output
11	VAR_GLOBAL	Level2Err	Y4	%QX0.4	BOOL	FALSE		Digital Output
12	VAR_GLOBAL	Temp2Err	Y5	%QX0.5	BOOL	FALSE		Digital Output
13	VAR_GLOBAL	Horn	Y6	%QX0.6	BOOL	FALSE		Digital Output
14	VAR_GLOBAL	Temp1	DT0	%MW5.0	INT	50		Analog Input
15	VAR_GLOBAL	Temp2	DT1	%MW5.1	INT	50		Analog Input

3.3.3 ローカル変数の定義

ローカル変数では、以下の項目を設定します。

■必須設定項目

・クラス（変数のクラス）：　通常の変数、値が保持される変数、値が固定の変数など。
・変数名：　変数の名前です。半角英数字100文字以内。（ひらがな、カタカナ、漢字は入力できません。）
・タイプ：　変数の型を設定します。（BOOL、INT、WORDなど）
・初期値：　変数の初期値を設定します。自動的に設定されますが、変更も可能です。

v 5

Ver.4のときは、変数名は半角英数字で16文字以内でしたが、Ver.5では、変数名は半角英数字で100文字以内と拡張されています。

■任意設定項目

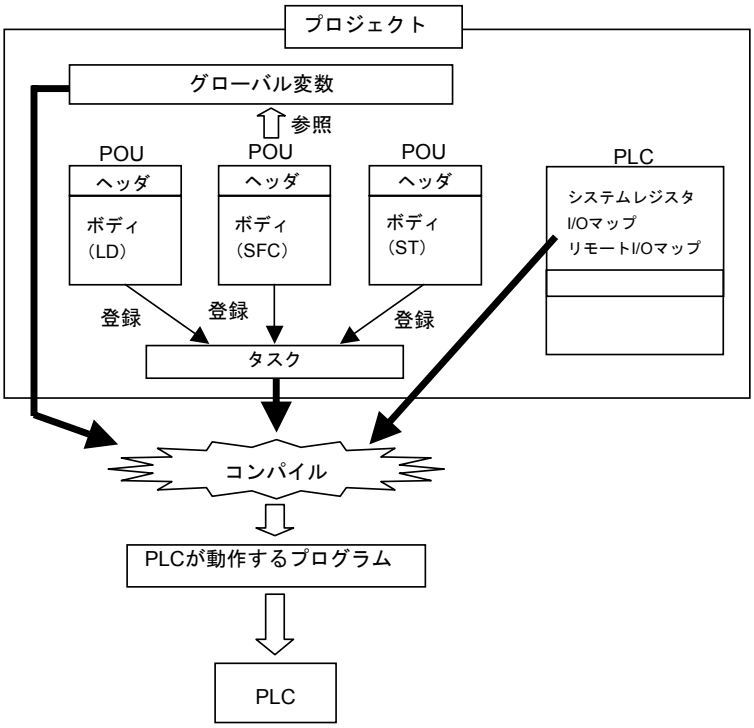
・コメント：　変数にコメントを書き込みます。ボディには表示されません。ひらがな、カタカナ、漢字を使って書くこともできます。

◆ご 注 意 !

・POUのヘッダには、そのPOU内で使う変数を全て登録する必要があります。
グローバル変数として登録した変数も、POUヘッダに登録します。
その場合は、「グローバル変数」として登録済みであることを示す「外部変数（VAR_EXTERNAL）」として「POUヘッダ」に登録されます。
「グローバル変数」として、登録された変数を自動的に全てのPOUのヘッダに「外部変数」として登録することもできます。
・ローカル変数にはPLCアドレスを割り当てることができません。
変数にあらかじめPLCアドレスを割り当てたい場合は、グローバル変数として登録してください。

	クラス	変数名	データ型	初期値	コメント
0	VAR	sw1	BOOL	FALSE	
1	VAR	Lamp1	BOOL	FALSE	
2	VAR	sw2	BOOL	FALSE	
3	VAR	Lamp2	BOOL	FALSE	
4	VAR	Temp3	INT	0	
5	VAR	sw3	BOOL	FALSE	
6	VAR	Lamp3	BOOL	FALSE	
7	VAR				

■プロジェクトの構成



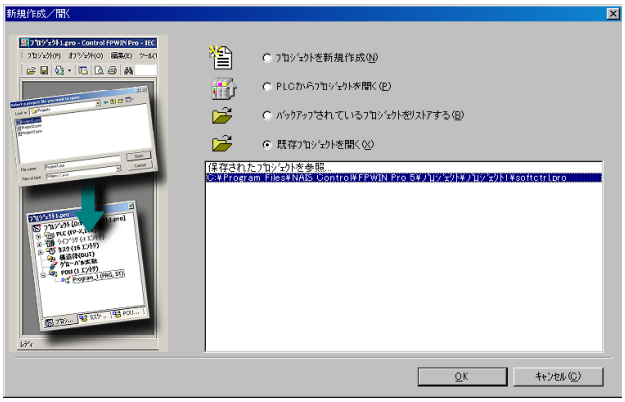
・PLCの動作プログラムは、POUのボディ部分に記述します。
・プログラム中で変数を使う場合は、変数が定義されたテーブルを参照します。
・作成したPOUはタスクに登録し、コンパイル（プログラム変換）を実行します。
・コンパイル時に、プログラムと一緒にPLC（システムレジスタ）のデータも変換され、PLCへ転送するデータが生成されます。

4章

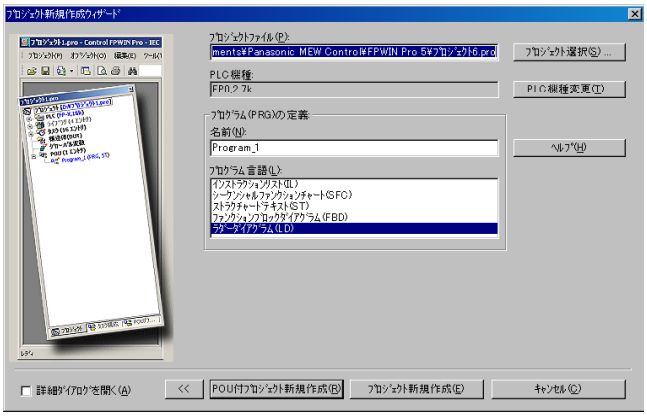
ラダーダイアグラムでプログラムを作成する

4.1　プロジェクトを新規作成する

FPWIN Proを起動すると、以下の画面が表示されます。



ここで「プロジェクトを新規作成」を選択し「OK」を押すと、以下の画面が表示されます。



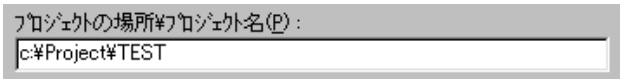
◆ご 注 意！

すでに存在するプロジェクトの下に、新規にプロジェクトを作成すると、後でプロジェクトを開くことが出来なくなります。この場合、エクスプローラ等でプロジェクト名がついているフォルダを適当な場所へ移動してください。

このウィザードで、以下の内容を設定してください。

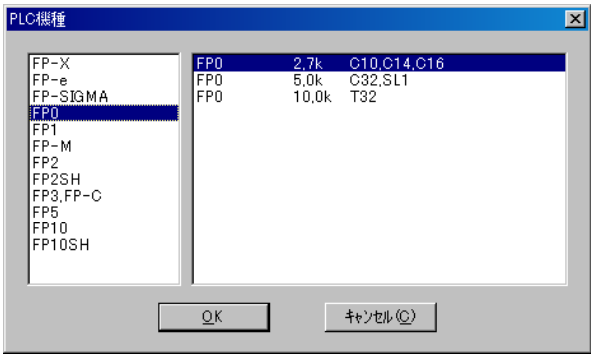
■作成するプロジェクトの場所と名前

「プロジェクトの場所¥プロジェクト名」のエリアに、プロジェクト名を入力します。
この例では、Cドライブに「Project」というフォルダをあらかじめ用意し、プロジェクト名を「TEST」と設定しています。



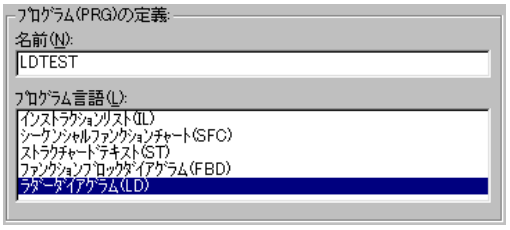
■プロジェクトの編集対象となるPLC機種

「PLC機種」のエリアに、使用するPLCの機種を選択します。
既に表示されているPLC機種を変更したい場合は、PLC機種変更(T)... ボタンをクリックすると以下のダイアログが表示されますので、ここで機種を変更してください。なお、PLC機種はプログラム作成中でも変換できます。



■プログラムの定義（POUに登録する名前と使用する言語）

「プログラム（PRG）の定義」エリアに、プロジェクトに登録されるPOU名と使用する言語を設定します。ここでは、POU名を「LDTEST」とし、言語はラダーダイアグラム(LD)を選択します。

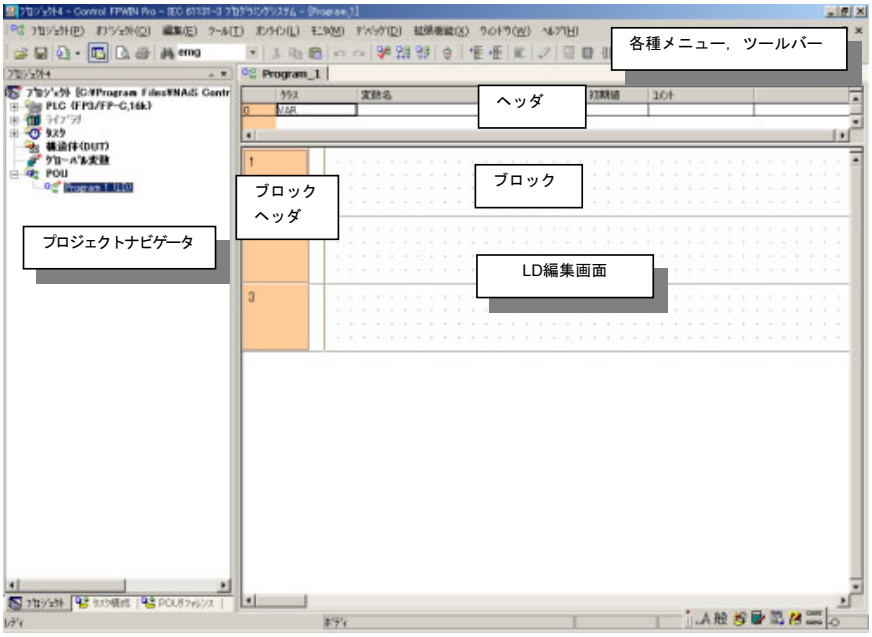


全ての設定が完了したら、POU付ﾌﾟﾛｼﾞｪｸﾄ新規作成(R) ボタンをクリックしてください。これで、最初の準備は完了です。

4.2 FPWIN Proの基本画面

プロジェクトを新規作成すると、以下の画面が表示されます。

■FPWIN Pro 起動時の画面



ここで、プロジェクトナビゲータの「POU」にある「+」をクリックしてください。
以下のようなツリーが表示されます。

最初に設定したPOU名「LDTEST」が表示されます。

4.3　ラダーを描く

4.3.1　LD編集用ツールバーについて

LD（ラダーダイアグラム）を編集するために、下記のツールバーを使います。

以下のボタンを特によく使いますので、役割を覚えておいてください。

■よく使用するアイコン

 　接点や応用命令を接続する線を描くために使われます。

 　応用命令やIEC命令等は、このアイコンで選択あるいは配置します。

 　接点（基本命令ST、AN、OR等）を配置します。

 　コイル（出力命令OT）を配置します。

 　入出力変数（応用命令のオペランド）を配置します。

 　編集画面上にコメントを配置します。

■編集の補助に使用するアイコン

 　現在アクティブなブロックの直前に、新しいブロックを挿入します。

 　現在アクティブなブロックの直後に、新しいブロックを追加します。

 　任意の場所を縦方向に広げます。

 　任意の場所を横方向に広げます。

v 5

Ver.4のときは、入力変数と出力変数のアイコンは区別されていましたが、
Ver.5では、入出力変数のアイコン１つになりました。

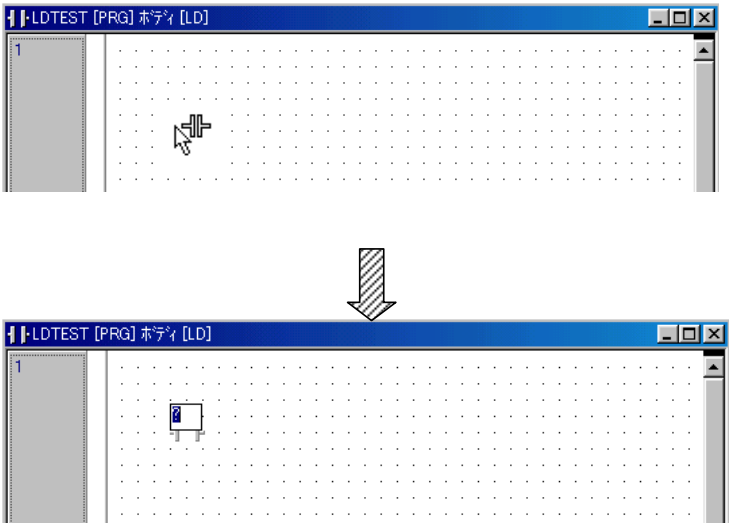
4.3.2　オブジェクトの配置

LD編集では、ツールバー上のオブジェクト（接点やコイル等）をクリックして、編集画面上の任意の場所で再度クリックすると、そこにオブジェクトが配置されます。　いわゆる「ドラッグ&ドロップ」とは違い、クリックした後にマウスのボタンを押し続ける必要はありません。

■操作手順

1. ツールバー上の接点  を、1回だけクリックします。

2. マウスポインタに選択したオブジェクトが付いてきますので、LD編集画面上の任意の位置（オブジェクトを置きたい場所）でもう一度クリックします。



3. 接点が配置されデバイス名の入力待ち状態となるので、ここでキーボードを用いて直接デバイス名を入力します。

◆ご 注 意 ！

デバイス名は必ず大文字で入力してください。小文字で入力すると変数と見なされ、変数を定義するためのダイアログが表示されます。
たとえば、「X0」と入力するとPLCアドレス、「x0」と入力すると変数と見なされます。
PLCのデバイスNo.を指定するときは、全て半角の大文字で入力してください。

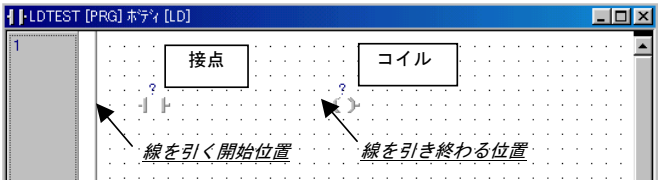
◆ここがポイント!

いくつか接点を配置して、後からデバイス名を入力することもできます。この場合、デバイス名のエリアには「？」が表示されたままとなります。

4.3.3 オブジェクトを接続する

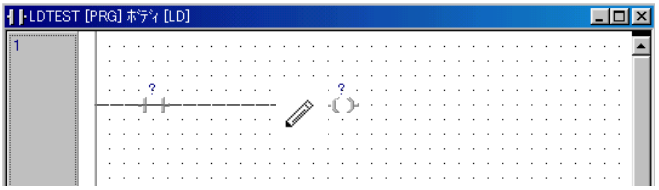
■操作手順

1. 基本操作１の手順に従って、以下の画面を作成します。



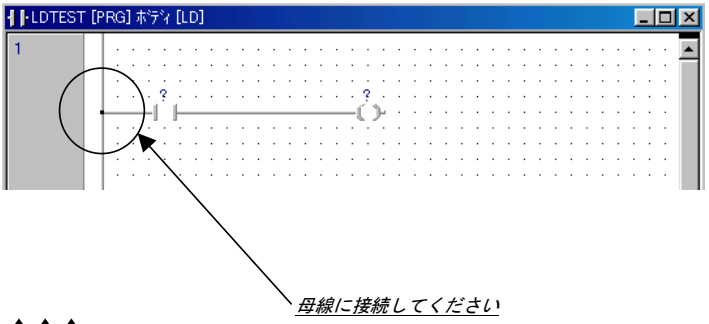
2. 線を描くためのアイコン  を一度だけクリックし、線を引く開始位置でもう一度クリックします。

クリックしたまま線を引き終わる位置までマウスを動かし、マウスボタンを離します。



3. 線を引き終わると、以下のような状態になります。

このように、LD編集画面上ではどこでも自由に線を描くことができます。



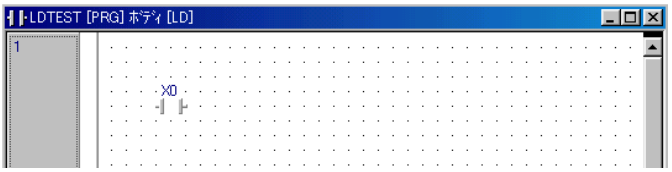
◆ご 注 意 !

接点の横の線は、必ず左側の母線につながるように描画してください。

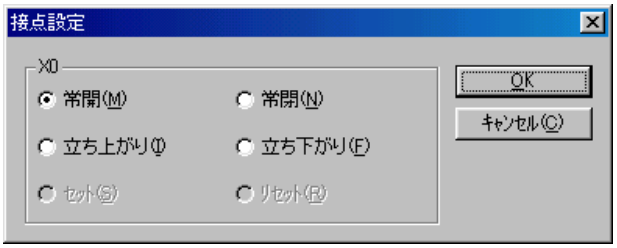
4.3.4 接点の設定を変える

■操作手順

1. 編集画面上に接点X0を配置します。



2. 接点をダブルクリックすると、以下のダイアログが表示されます。

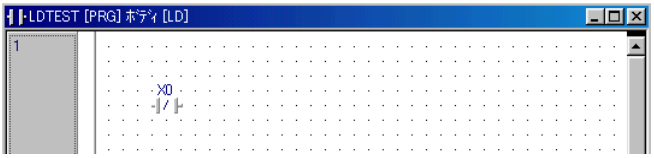


3. ここで、a接点やb接点の設定、立ち上がり／立ち下がりの設定を行います。

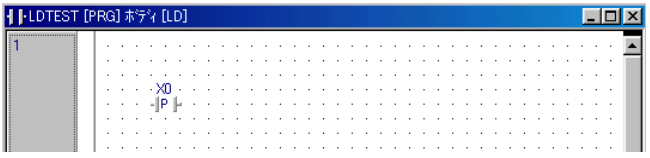
デフォルトは「常開」（a接点）に設定されています。

以下に、それぞれの設定を選択した場合の状態を示します。

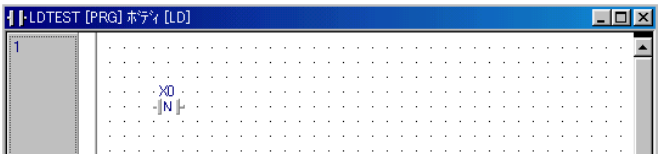
・常閉



・立ち上がり



・立ち下がり



4.3.5　応用命令の入力

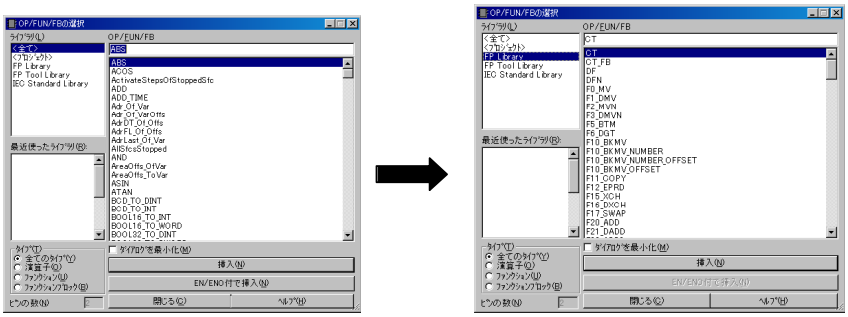
■操作手順

1. ツールバー上の  を、1回だけクリックします。

以下の命令選択用ダイアログボックスが表示されます。

FPWIN Proでは、接点以外の全ての命令はこのダイアログから入力します。

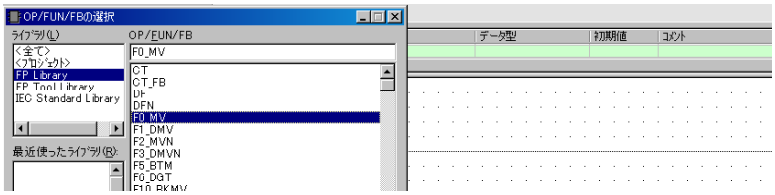
ここで、左上の「ライブラリ」の欄において、「FP Library」をダブルクリックしてください。右側のエリアに、PLCの標準的な命令一覧が表示されます。



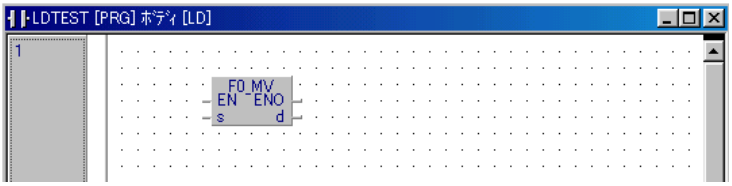
2. データ転送命令（F0 MV）を入力します。

ライブラリの中で、「F0_MV」の部分をダブルクリックするか、「挿入」ボタンをクリックしてください。これで、MV命令が選択されました。

このまま、マウスをLD編集画面上に移動させると、以下のようにマウスポインタにF0_MV命令が付いてきます。



任意の位置でマウスをクリックすると、以下のようにMV命令が配置されます。



各ピンには、以下に内容を接続します。

EN：　この命令を実行する条件を接続します。

ENO：　MV命令実行後に別の命令を起動したり、コイルをONしたりする場合に使用します。
（使用しない場合は、接続する必要はありません。）

s：　転送元

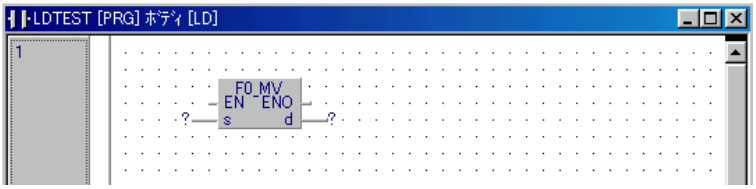
d：　転送先

3. オペランドを配置します。

最後に、このMV命令に必要なオペランドsとdを配置します。

オペランドの入力には、 VAR アイコンを使用します。

それでは、接点を配置したときと同じように VAR をクリックしてMV命令のsの端子横に配置し、 VAR をクリックしてdの端子横に配置してください。



後は、?の部分に実際のデバイス名を入力すれば完了です。

v 5

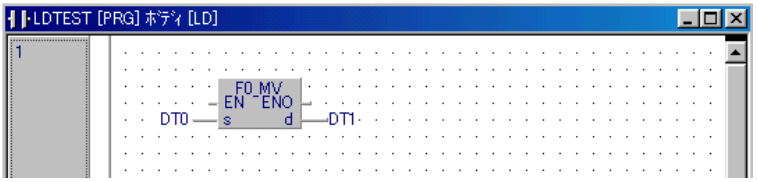
Ver.4のときは、オペランド入力のために、入力変数・出力変数のピンを接続する必要がありましたが、Ver.5では、はじめから各応用命令に入出力変数のピンが接続されています。

◆ここがポイント!

オペランドの配置とデバイス名の入力の順序は問いません。オペランドをまとめて配置してからデバイス名を入力することもできます。

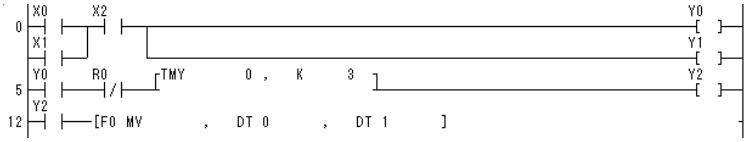
◆ご 注 意 !

デバイス名は必ず大文字で入力してください。



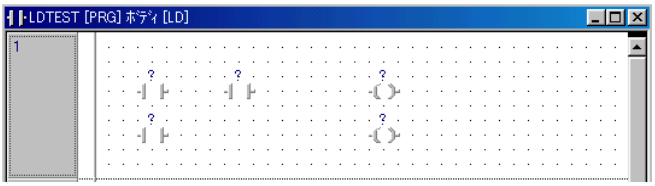
4.3.6 ラダープログラムの作成例

では､以下のプログラムを作成してみます。(FPWIN GRで作成したプログラムです。) PLCは、FP0の2.7Kタイプを選択しています。

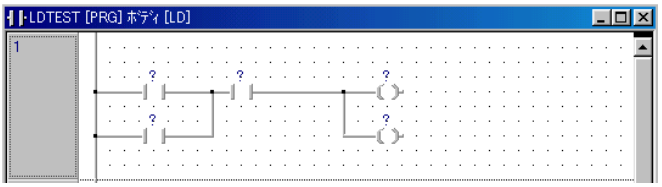


■操作手順

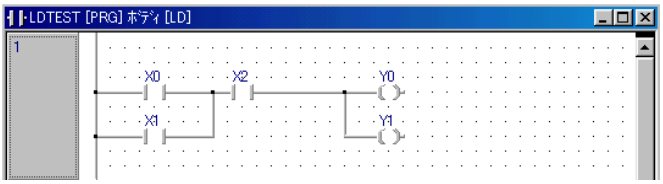
1. 最初のブロック1に、デバイス（接点、コイル）を編集画面に配置します。



2. デバイス同士を接続します。



3. デバイス名を入力します。



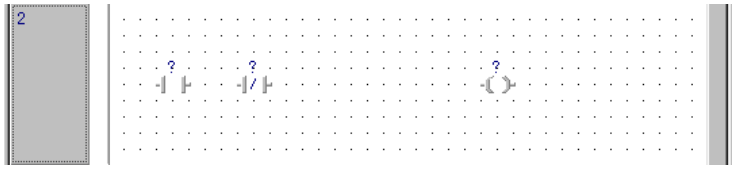
◆ここがポイント!

1.の後にそのままデバイス名を入力し、線を描画することもできます。
最終的に3.の絵が完成すれば良いので、順序でも編集を行うことができます。

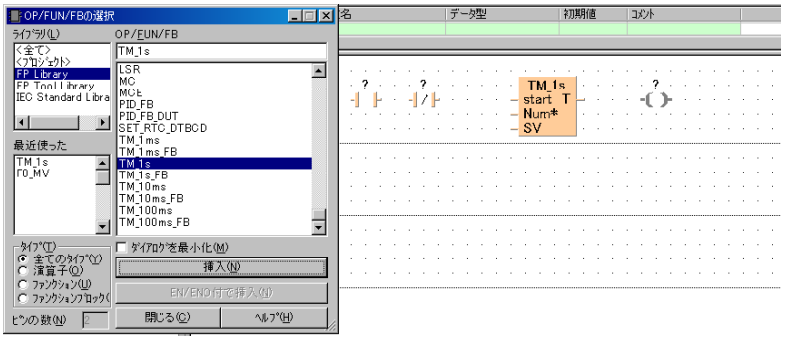
同様にして、次のブロック2にタイマ命令のあるブロックを描画します。

■操作手順

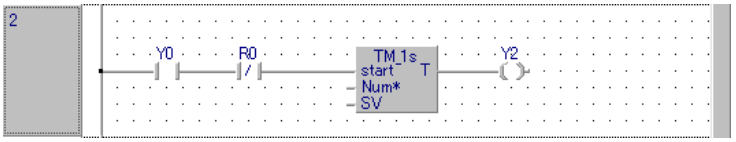
1. 接点とコイルを配置します。（左から2番目の接点は常閉です。）



2. タイマ命令を、命令選択用ダイアログボックスから選びます。



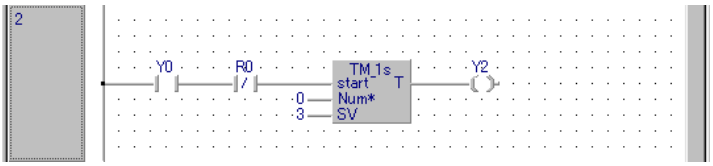
3. 各オブジェクトを接続し、タイマ命令のオペランドを  **で配置します。**



v 5

Ver.4のときは、オペランド入力のために、入力変数・出力変数のピンを接続する必要がありましたが、Ver.5では、はじめから各応用命令に入出力変数のピンが接続されています。

4. 各オブジェクトに、デバイス名等を入力します。



タイマ命令の各ピンには、以下の内容を接続します。

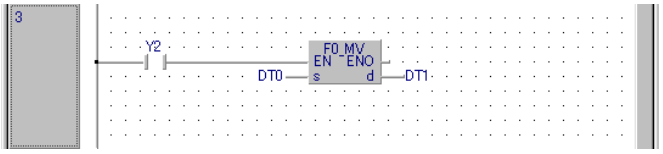
・start： タイマ命令を起動するトリガ

・Num*：タイマナンバー

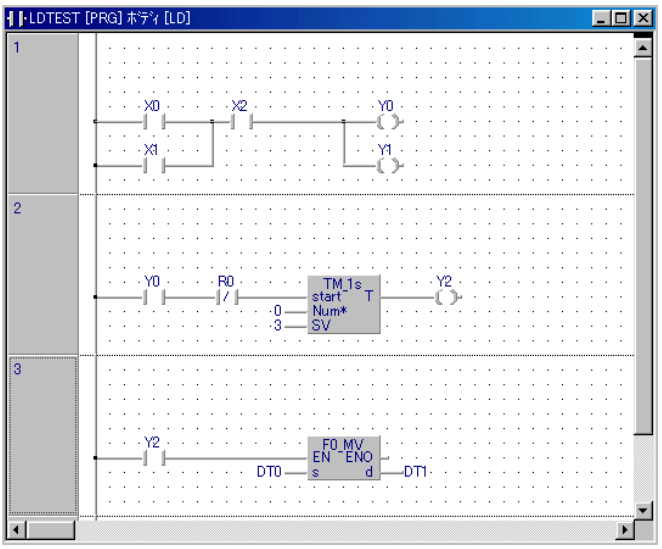
・SV： タイマの設定値

・T： タイマが起動したら有効になる

5. 最後にブロック3にMV命令のあるブロックを描画します。



6. これでラダーは完成しました。全体として、以下のようになります。



◆ご 注 意 ！

１ブロック内に描画できる要素数は、ラインを含めて１６０までです。

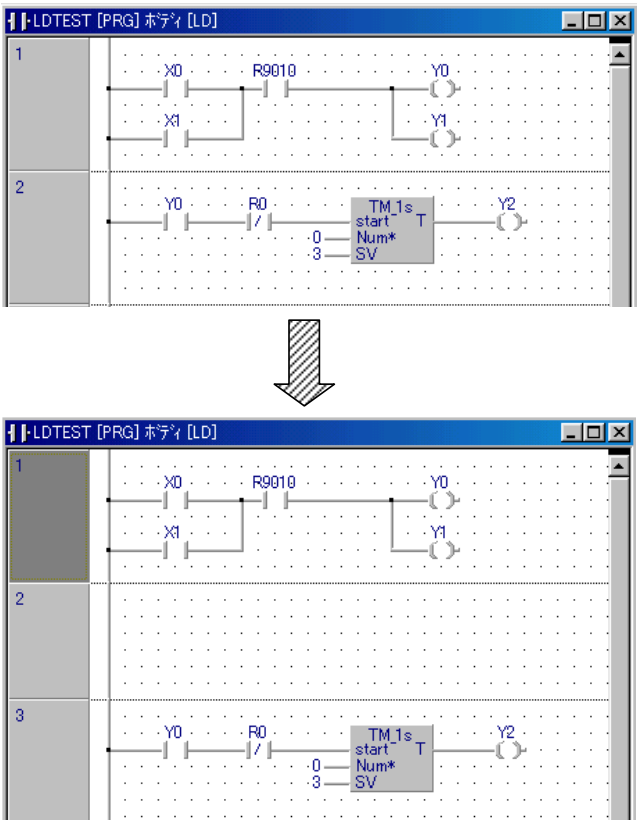
4.4 編集補助機能について

4.4.1 ブロックの追加・挿入・削除

ブロック1と2の間に新しいブロックを挿入する場合の手順を下記に示します。

■操作手順（追加・挿入）

1. **1のブロックヘッダをアクティブにしておき（マウスでクリックする）、**  **をクリックするか、2のブロックヘッダをアクティブにしておき、**  **をクリックします。**
2. **すると、以下のようにブロック1と2の間に新しいブロックが挿入されます。**



なお、ブロックの最後に何らかのオブジェクトが配置されると、新しいブロックが自動で追加されます。

■操作手順（削除）

1. **1のブロックヘッダをアクティブにしておき（マウスでクリックする）、[Delete]キーを押すと、１のブロックが削除されます。**

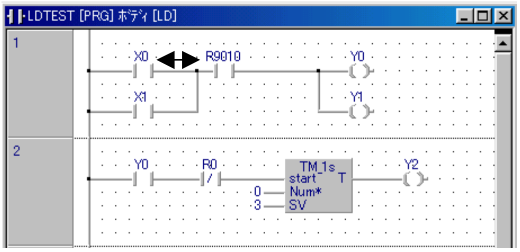
4.4.2 任意の場所を広げる

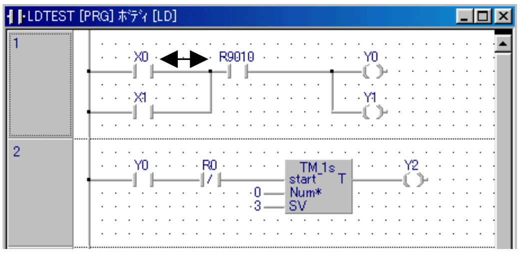
X0とR9010の間を広げて新しいオブジェクトを配置したい場合の手順を以下に示します。

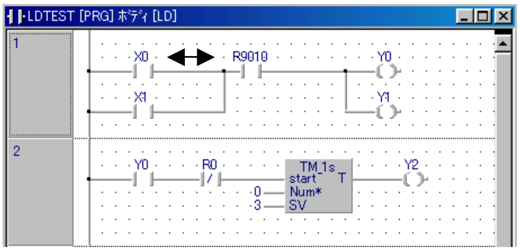
■操作手順

1. ツールバーの ⇔ をクリックして、編集画面上の広げたい場所へマウスを移動させます。

2. マウスをクリックし、場所を広げていきます。クリックした回数に応じて、場所の広がりは大きくなっていきます。







3. Shiftキーを押しながらマウスをクリックすると、逆に選択したエリアが狭くなります。

◆ご 注 意！

拡げたり、縮めたりできるのは、同一ブロック内のみです。他のブロックには影響されません。

4.5 コンパイルを実行する

ここでは先ほど完成したラダーのプログラム変換を行います。（FPWIN GRのPG変換に相当します。）
FPWIN Proでは、この作業を「コンパイル」と呼びます。

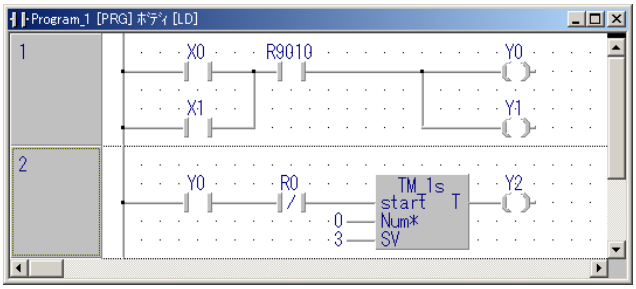
4.5.1 コンパイルについて

■FPWIN Proのコンパイルチェックには、次の2通りのチェックが存在します。

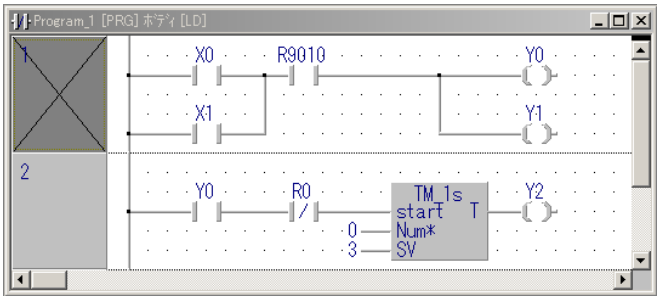
・オブジェクトチェック
ここでは、現在アクティブな画面に対してのみチェックを行います。
システムレジスタ設定画面がアクティブなら、システムレジスタのチェックを行いますが、この時、LD編集画面はチェックされません。
現在編集中の画面での簡単な文法チェックに有効な機能です。

・全コンパイル
システムレジスタやプログラムなど全ての情報をチェックして、PLCへ転送するためのデータを生成します。
全コンパイルはオブジェクトのチェック機能を含んでいますので、この機能のみ実行しても構いません。

■ブロックをコンパイル対象外にする

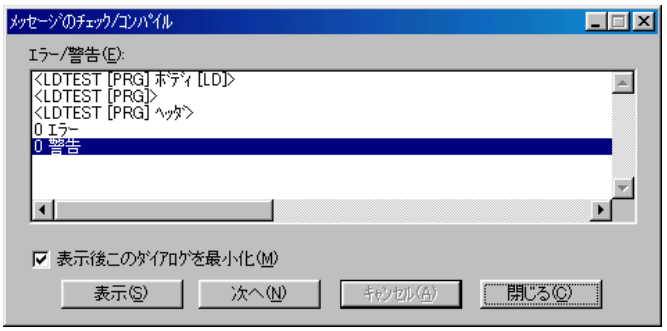


１のブロックヘッダを右クリックし、[ブロックをコンパイル対象／対象外にする］を選択することにより、コンパイル対象外にすることができます。



4.5.2 オブジェクトのチェックを行う

オブジェクトのチェックを行うには、チェックを行いたい編集画面をアクティブにして、  のアイコンをクリックするか、「オブジェクト」メニューの「チェック」を選択してください。
作成したLD編集画面をアクティブにして、オブジェクトのチェックを実行してください。以下の画面が表示されます。



上記のメッセージでは、問題なくチェックが終了しましたが、例えば次ページの例のようなラダーでチェックを行うと、エラーや警告が発生します。
エラーもしくは警告が表示された箇所をダブルクリックすると、実際の編集画面で該当する箇所が赤色で表示されます。

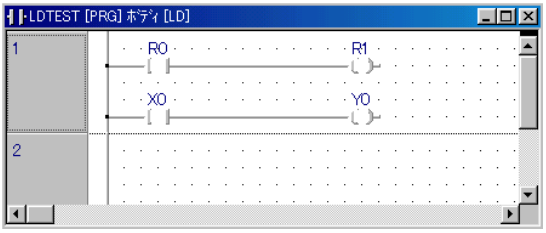
◆ご 注 意 !

エラーは必ず対処しなければ、プログラムをPLCへダウンロードできませんが、警告はそのままでもPLCへダウンロードすることができます。ただし、以後のデバッグ機能等で弊害が出ることがありますので、エラーも警告も0の状態にすることをおすすめします。

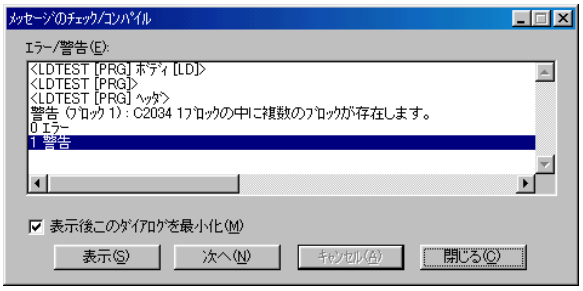
4.5.3 エラーや警告が発生する場合

■LD編集画面の1ブロックに、ラダーを2ブロック編集している。

編集画面の状態



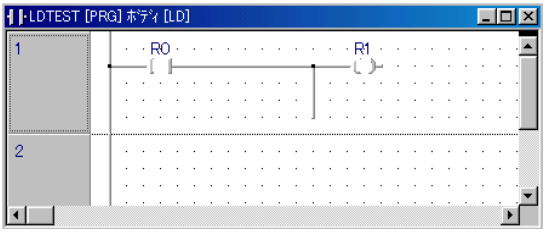
チェックの結果



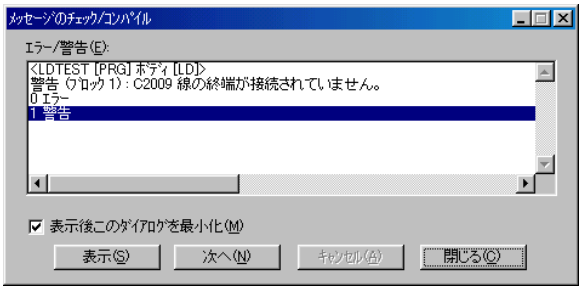
対策：2つ目のラダーブロックを、次のブロック2に移動してください。

■未接続の線が存在する。

編集画面の状態



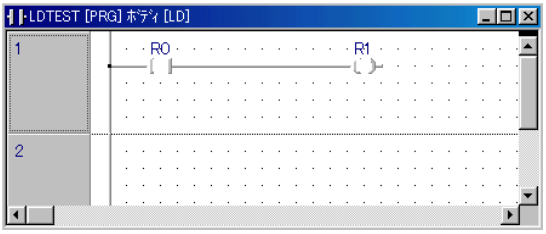
チェックの結果



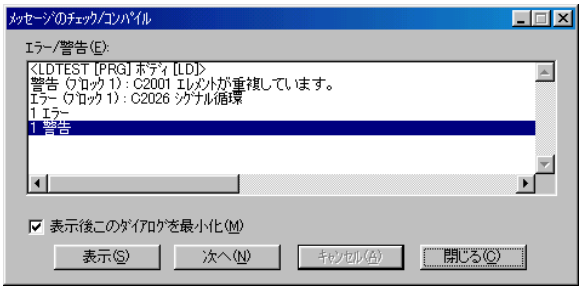
対策：未接続の線を消去してください。

■同じラインに2本の線が重なっている。

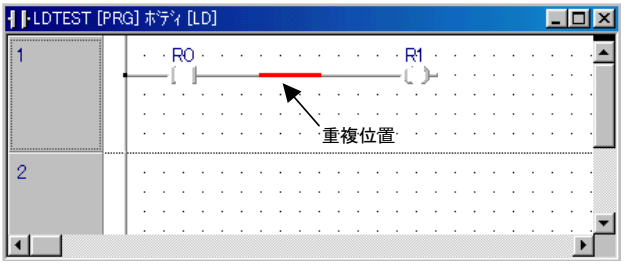
編集画面の状態



チェックの結果



ここで、警告の「エレメントが重複しています。」というメッセージの部分をダブルクリックしてください。該当部分が赤色で示されます。



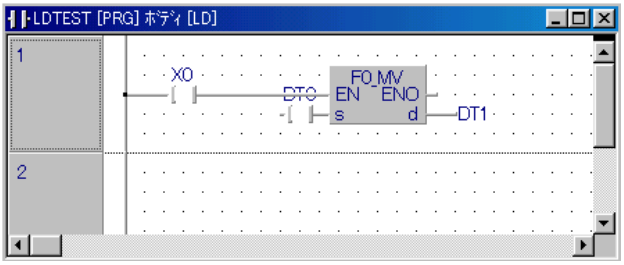
対策：重なった線を消去してください。

v 4.1

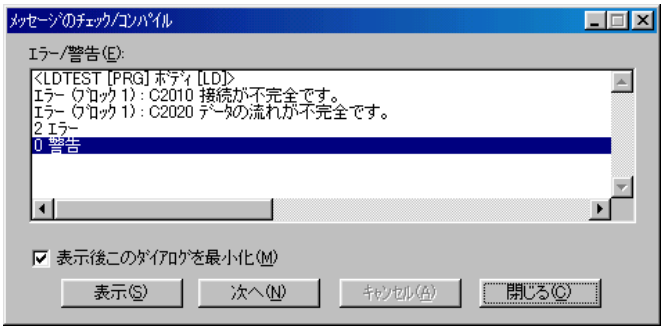
Ver.4.1以降、線の重複は１本の線と認識できるようになりました。

■応用命令で、オペランドを入力する場所に接点を配置した。

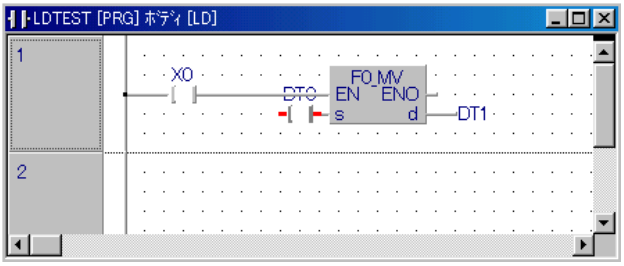
編集画面の状態



チェックの結果



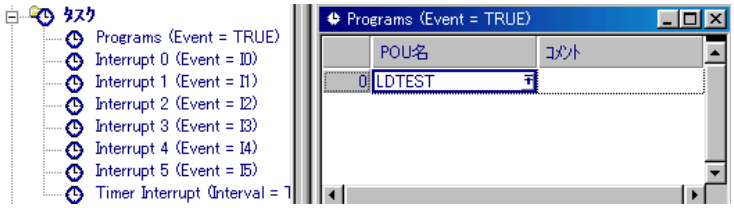
ここで、エラーの「接続が不完全です。」というメッセージ部分をダブルクリックしてください。該当部分が赤色で示されます。



対策：接点の部分を消去し、オペランドを入力してください。

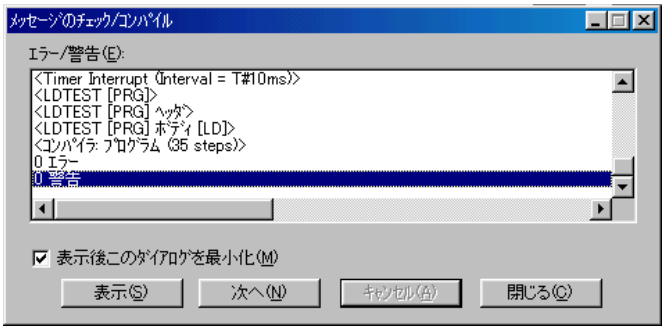
4.5.4 コンパイルを実行する

「オブジェクトのチェック」で問題がなければ、コンパイルを実行します。
コンパイルが問題なく終了すれば、そのプロジェクトはPLCへダウンロード可能となります。
(オブジェクトのチェックを行わずに、コンパイルを実行しても構いません。オブジェクトのチェックが未実施の場合は、コンパイル時に自動で実行されます。)
コンパイルを実行する前に、プロジェクトナビゲータの「タスク」内にある「Programs (Event = TRUE)」をダブルクリックし、以下の状態を確認してください。



ここに、POUの「LDTEST」が登録されているのが確認できます。
FPWIN Proは、このタスクに登録されたPOUのみのコンパイルを実行します。
POUを新規作成したとき、タスクへの登録は自動で行われましたので、このままコンパイルすることができます。
POUが複数個登録されている場合は、上から順番にコンパイルが実行されます。

コンパイルは、プロジェクトの全ての情報をチェックします。
 のアイコンをクリックするか、「プロジェクト」メニューの「全コンパイル」を選択してください。



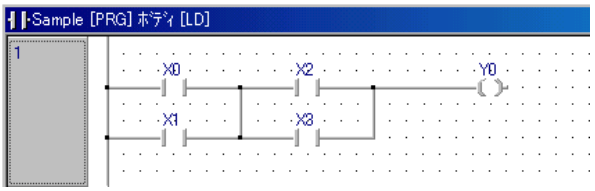
上記のように、「0エラー、0警告」でコンパイルは正常終了となります。
ここでエラーが発生した場合は、該当するメッセージをダブルクリックして、エラー箇所を修正してください。
なお、このダイアログを開いたままでも、LD編集画面やシステムレジスタ設定画面を修正できます。

4.6 コンパイルに関する注意事項

FPWIN Proにおけるコンパイルは、FPWIN GR（もしくはNPST-GR）のPG変換と同じような働きをしますが、実際の変換方法において異なる部分があります。
なかでも以下の項目は重要ですので、今までFPWIN GRやNPST-GRをお使いの場合はその違いをご確認ください。

4.6.1 生成されるニモニックコードが異なる

例えば、以下のようなラダーを編集して変換します。



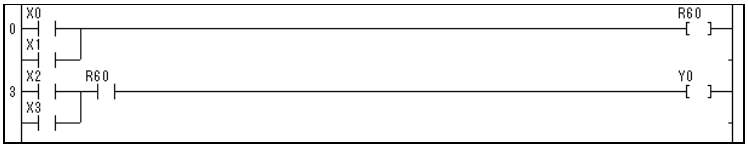
これはFPWIN GR（もしくはNPST-GR）の場合、以下のコードに変換されます。

```
ST X0
OR X1
ST X2
OR X3
ANS（アンドスタック）
OT Y0
```

FPWIN Proの場合、以下のコードに変換されます。
（実際はプログラムの先頭に数ステップのコードが付加されますが、それは省きます）

```
ST X0
OR X1
OT R60
ST X2
OR X3
AN R60
OT Y0
```

これを改めてラダーで表現し直すと、以下のようになります。



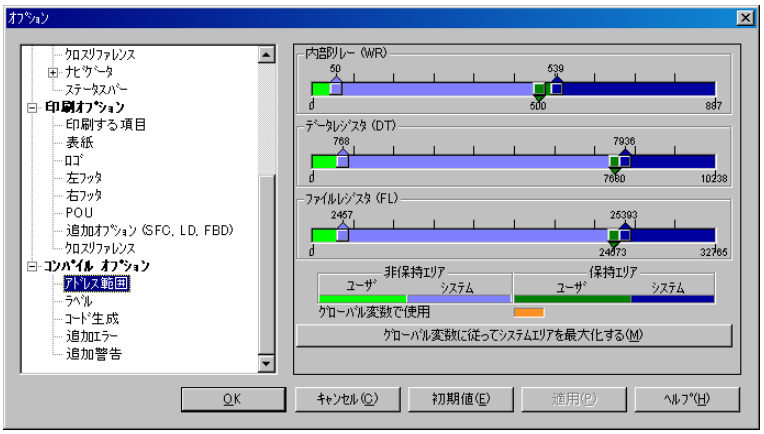
これらを比較すると、同じ動作をするプログラムが異なるコードで表現されています。また、実際のプログラムステップ数も異なります。（アンドスタックやオアスタックは、通常変換されません。）このように、FPWIN Proでコンパイルを行うと、FPWIN GR(NPST-GR)の場合とは、異なるコードが生成されています。

4.6.2 PLC本体の演算用メモリの使用

コンパイル実行時に、PLC内部メモリ（リレーやレジスタ等）を使用する場合があるので、実際にプログラムで使用できるPLC内部メモリは少なくなります。

まずは前頁ののプログラム中に、描画していないR60が使われていることを確認してください。これは、FPWIN Proがコンパイルを実行時に、PLCの内部メモリを使用したことを意味します。

「拡張機能」メニューの「オプション」メニューから、「コンパイルオプション」の「アドレス範囲」を選択してください。
以下のダイアログが表示されます。



上記の例では、内部リレー（R）は、WR6～60の55ワード（非保持）がコンパイル用に割り当てられており、実際のプログラムで使用することはできません。データレジスタ（DT）の場合は、1487ワード（保持と非保持）が割り当てられています。
これらの値は変更することもできますが、あまり小さくするとコンパイル時にプログラムを変換することができず、エラーになる可能性があります。

◆ここがポイント!

FPWIN Proでは、グローバル変数以外は内部メモリのアドレスを指定しないのが一般的です。従ってPLCの内部メモリは、ローカル変数に割り当てられたり、プログラムの変換補助用として使用されます。

5章

PLCにダウンロードする

5.1 PLCとオンラインにする

プログラムが完成し、コンパイルができたらPLCへダウンロードします。

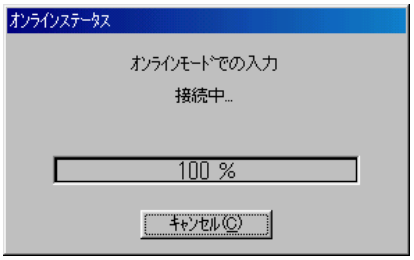
5.1.1 通信条件を設定する

まずは、「オンライン」メニューの「通信設定」を選択し、以下のダイアログを起動します。

ここで、使用する通信ポートやボーレート等を設定します。(PLC側の設定を確認してください。)
「OK」を押すと、現在の設定が保存されます。
「自動通信の設定内容」でチェックされた項目は、最初に設定された条件でPLCと通信が確立できなかった場合に、自動的に検索されます。

5.1.2 FPWIN Proをオンラインに移行する

前項で起動したダイアログ内の「オンラインモード」ボタンを押すか、ツールバーの  を選択すると、FPWIN ProはPLCと通信を開始します。

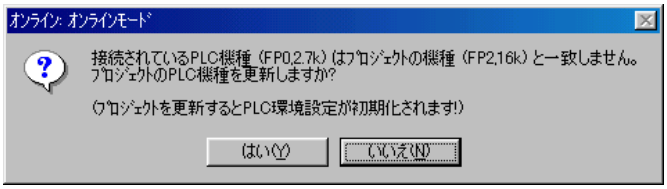


PLCとの通信に成功すると、新しくオンライン用のツールバーが表示されます。

設定された条件でPLCと通信が確立できなかった場合、自動通信設定機能が働きます。
「自動通信設定機能」でチェックされた項目の通信条件を変えながら、PLCとの通信確立を行います。

全てチェックしてもPLCと通信が確立できない場合はエラーメッセージが表示され、オンラインへの移行処理を中止します。

また通信が確立されても、プロジェクトに設定されたPLCと実際に接続されたPLC機種が異なる場合は、以下のメッセージが表示されます。



「はい」を選択すると、接続されたPLC機種に従ってシステムレジスタの初期化等を行い、オンラインモードへ移行します。
「いいえ」を選択すると、オンラインモードへの移行処理を中止します。

■よく使用するアイコン

オンライン用のツールバーの中で、以下のボタンは特によく使いますので、役割を覚えておいてください。


PLCへプロジェクトをダウンロードします。


オンライン編集時、プログラム変更部分をダウンロードします。


オンライン編集モードに移行します。


LD編集画面のモニタモードを切り替えます。

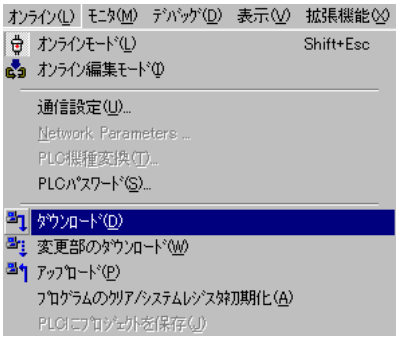
PLCの動作モードを切り替えます。

5.2 ダウンロードを実行する

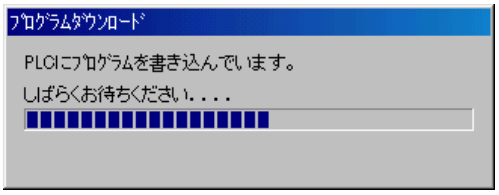
オンラインモードへ移行できたら、作成したプロジェクトをPLCへダウンロードします。

5.2.1 ダウンロードの手順

ダウンロードは、「オンライン」メニューから「ダウンロード」を選択するか、ツールバーの  を選択すると開始されます。



ダウンロード実行中は、以下の画面が表示されます。



このまま進行して、この画面が閉じられたら、ダウンロードは正常終了したことになります。
ダウンロード中にエラーが発生した場合は、エラーメッセージが表示されダウンロードは中止されます。

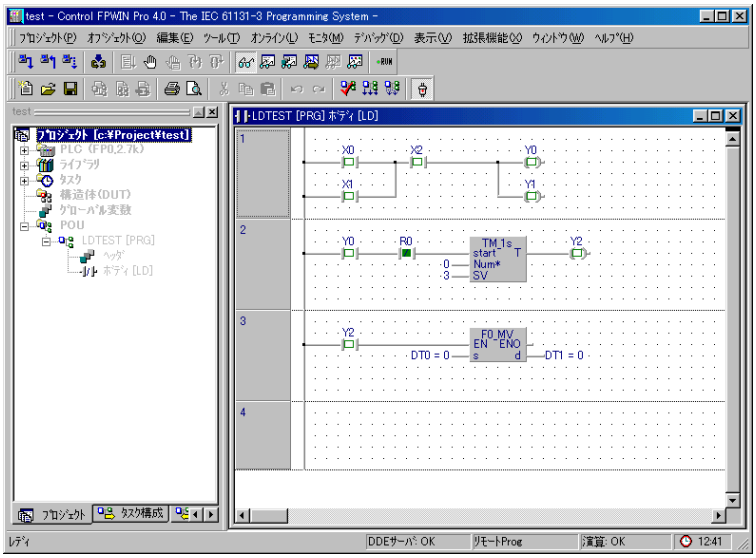
5.3 モニタを実行する

ダウンロードが正常終了したら、LD画面をモニタモードに移行します。

5.3.1 モニタモード移行の手順

モニタモードへ移行するには、「モニタ」メニューの「モニタ実行」を選択するか、ツールバーの  を選択します。

モニタモードへ移行されると、LD編集画面は以下のようになります。



5.4　オンライン編集を行う

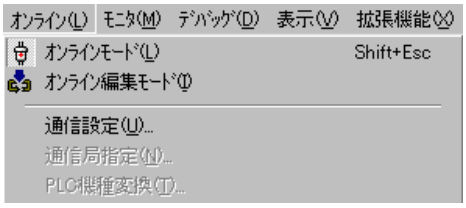
5.4.1　オンライン編集時の制限事項

FPWIN Proは、PLCとオンライン中にプログラムの編集をすることができます。
PLCがRUNモードであっても可能です。（FP1/FP－Mを除く）
この機能は、FPWIN GR（またはNPST-GR）の「RUN中書き換え」に相当します。

オンラインモード時、一度にPLC本体へダウンロードできるのは最大118ステップ分です。

5.4.2　オンライン編集の手順

LD編集画面がオンライン中に、「オンライン」メニューから「オンライン編集モード」を選択するか、ツールバーの  ボタンを選択します。

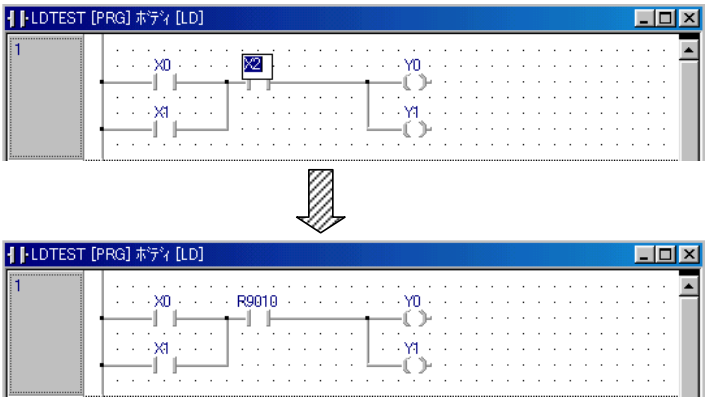


オンライン編集モードに移行すると、LD編集画面上の各オブジェクト（接点やコイル等）が変更できるようになります。
また、新しくブロックを追加したり、ラダーを追加することもできます。

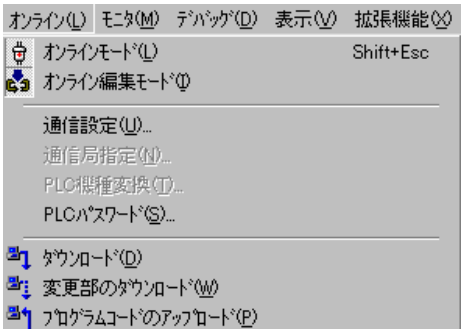
＜例＞

下記のように「X2」を「R9010」に変更する場合

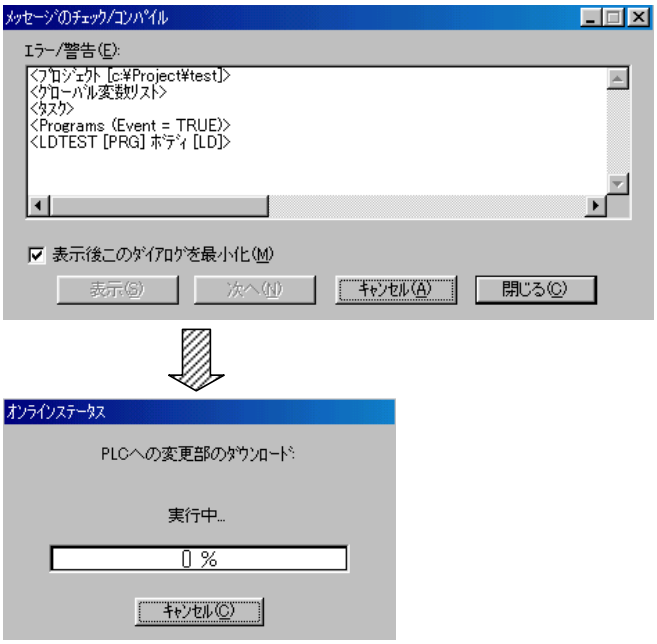
1. X2の部分をマウスでクリックし、変更できる状態にして、R9010を入力します。



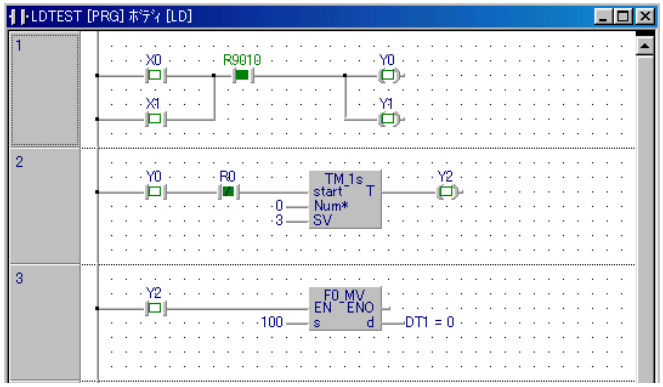
2. 変更が完了したら、「オンライン」メニューの「変更部のダウンロード」を選択するかツールバーの ボタンを選択してください。



3. FPWIN Proは、変更のあった部分を再コンパイルして、PLCへダウンロードします。



4. ダウンロード完了後の画面です。



5. オンライン編集モードを解除するには、もう一度メニューの「オンライン編集モード」かツールバーのボタンを選択してください。

6章

PLCからアップロードする

6.1 FPWIN Proにおけるアップロード

FPWIN Proでアップロードを実行する場合、次の2つの方法があります。

■プログラムのアップロード

現在PLC内に格納されているプログラム（命令コード）をアップロードします。
アップロードされた命令コードは、ニモニックNONラダー形式で表示されます。
（ラダーやその他の形式に変換できません。）
つまり、アップロードを行ってもPOUは再現されません。
プログラムのアップロードは、全てのPLCでサポートされます。

■プロジェクトのアップロード

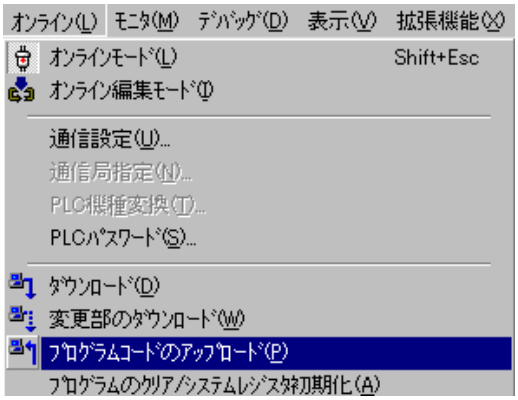
現在PLC内に格納されているプログラム（命令コード）と、PLCの拡張メモリに格納されたラダーの絵情報や変数の情報等をアップロードします。
この場合はプログラムのアップロードと違い、全てのPOU（ラダーなど）が再現されます。
プロジェクトのアップロードができるPLCは以下のタイプです。

・FP Σ
・FP2（オプションのコメントメモリ“AFP2201, AFP2202, AFP2203”装着時）
・FP2SH
・FP10SH（オプションのカードボード“AFP6208”“AFP6209A”装着時）

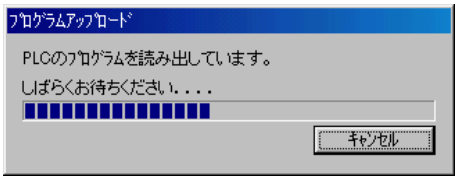
6.2 アップロードを実行する

6.2.1 プログラムをアップロードする

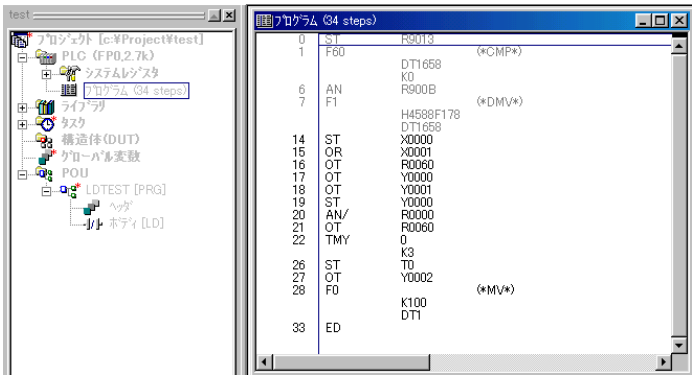
プログラムのアップロードは、FPWIN ProがPLCとオンライン状態で「オンライン」メニューの「プログラムコードのアップロード」を選択するか、ツールバーの  を選択して実行します。



アップロードが開始されます。



アップロードが終了後、プロジェクトナビゲータの「PLC」から「プログラム」をダブルクリックすると、アップロードしたプログラムがニモニックNONラダー形式で確認できます。

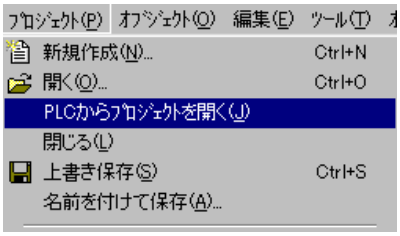


◆ご 注 意 !

アップロードされた内容は、この場所にのみ反映されます。
POUの編集画面には反映されません。

6.2.2 プロジェクトをアップロードする

プロジェクトのアップロードは、「プロジェクト」メニューから「PLCからプロジェクトを開く」を選択して実行します。



7章

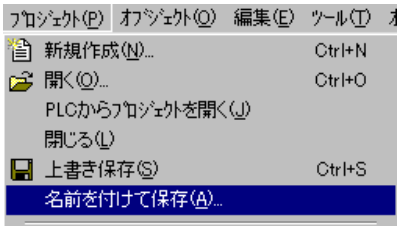
プロジェクトを保存する

7.1 プロジェクトの保存について

FPWIN Proでは、最初にプロジェクトを作成するときに「プロジェクト名」を設定しますが、
プロジェクトを作成した時点で、指定した場所に設定した名前のフォルダが生成されます。
プロジェクトを作成して保存せずに終了しても、このフォルダは残ります。
プロジェクトの保存を実行すると、このフォルダの中身が更新されることになります。

7.2　プロジェクトの保存を実行する

プロジェクトの保存は、「プロジェクト」メニューから「名前を付けて保存」を選択して実行します。
上書き保存する場合は「プロジェクト」メニューから「上書き保存」を選択するか  アイコンを選択してください。



◆ご 注 意 !

フォルダ名をエクスプローラ等で直接変更すると、プロジェクト名を変更することになります。
この作業については、特に問題ありません。
但し、フォルダの中にあるいくつかのファイルについては、名前を変更しないでください。

すでに存在するプロジェクトの下に、新規にプロジェクトを保存すると、後でプロジェクトを開くことが出来なくなります。この場合、エクスプローラ等でプロジェクト名がついているフォルダを適当な場所へ移動してください。

8章

その他の機能

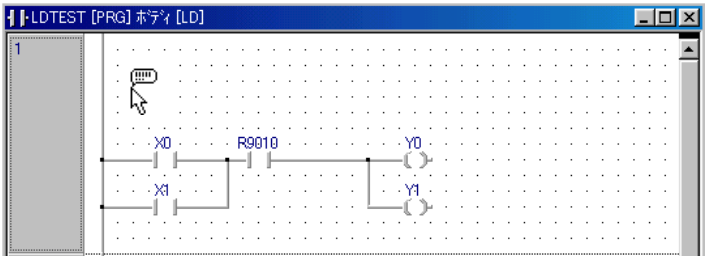
8.1 コメントの入力

FPWIN Proでは、編集画面上の任意の場所にコメントを書き込むことができます。
但し、FPWIN GRの「I/Oコメント」や「注釈文」と違い、コメントはどのデバイスにも付属しません。

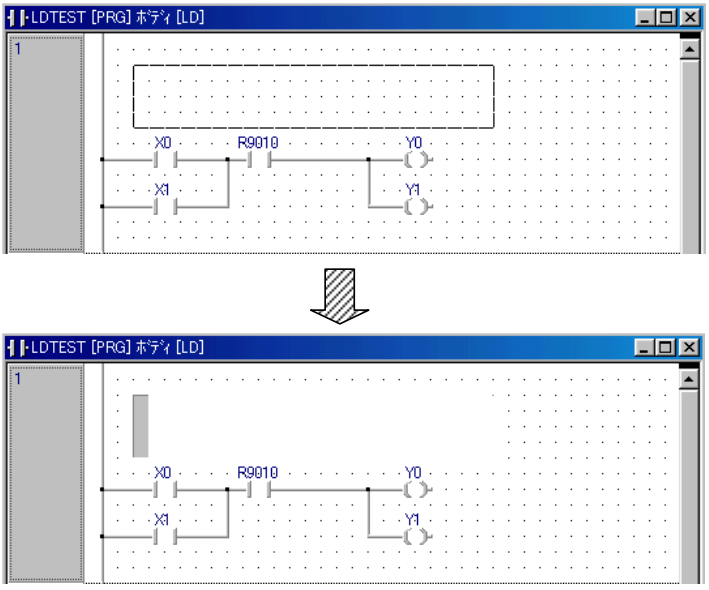
コメントの入力方法

■操作手順

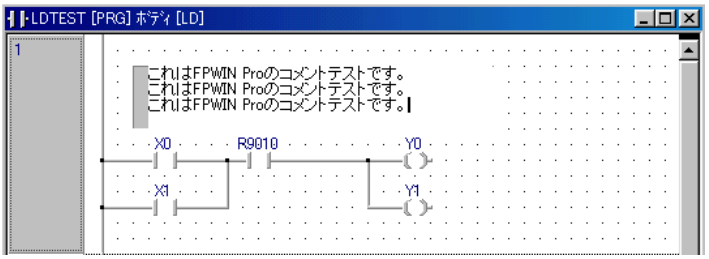
1. **コメントの入力は、LD編集画面がアクティブな状態でLD編集用ツールバーの**  **ボタンをクリックして、コメントを書き込みたい場所へマウスを移動させます。**



2. **マウスで四角形を描画する要領で、ある程度のコメントエリアを確保しておきます。**

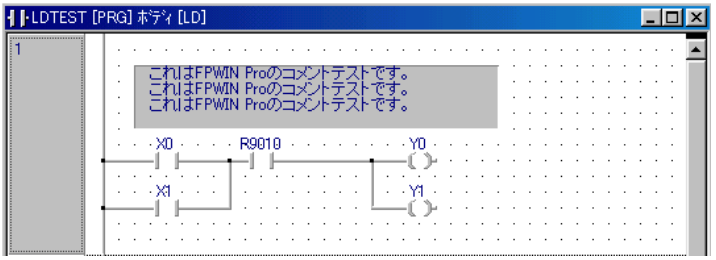


3. **キーボードから、任意のコメントを書き込みます。**



なお改行は、Ctrl + Enterキーで実行してください。

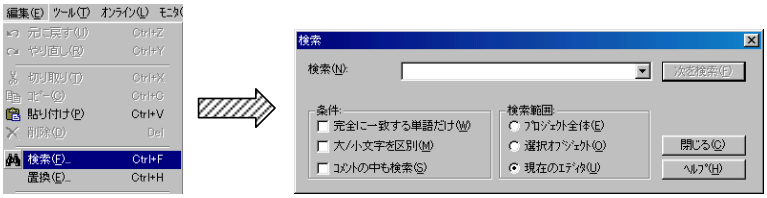
4. **最後にEnterキーで確定します。**



なお、コメントエリアの左側をマウスでクリックすると、そのまま画面上のどこにでも動かすことができます。

8.2 検索を実行する

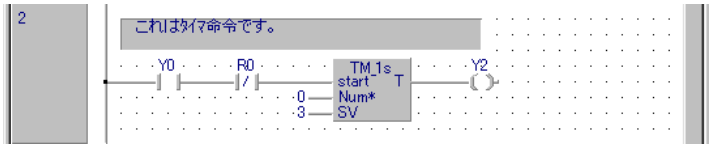
検索を実行するには、「編集」メニューの「検索」を選択してください。検索用のダイアログが表示されます。



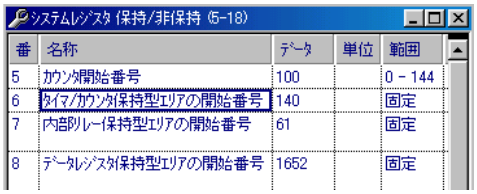
ここで、検索する文字列を入力し、 次を検索(F) ボタンを押してください。
該当する文字列にジャンプします。

8.2.1 検索範囲について

以下のような編集画面で、「ﾀｲﾏ」という文字を検索します。



・プロジェクト全体： プロジェクト内全ての「ﾀｲﾏ」という文字を検索します。
よって、例えば「システムレジスタ」にある「ﾀｲﾏ」という文字列も検索します。



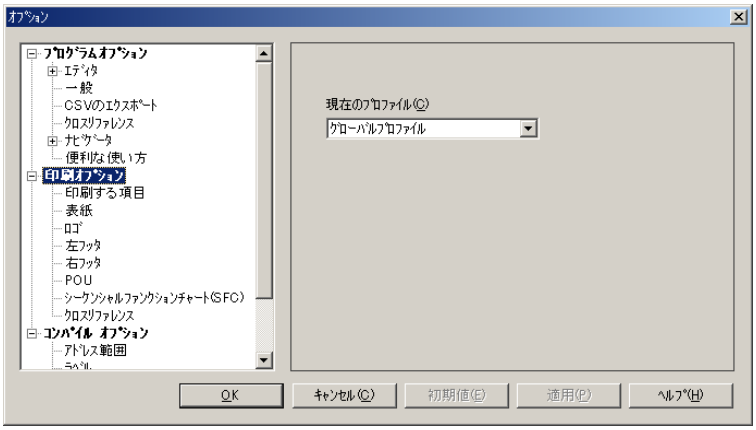
・選択オブジェクト：POUやPLCなどの範囲で、検索を実行します。
・現在のエディタ：現在アクティブな画面内のみ検索を実行します。

◆ご 注 意 !

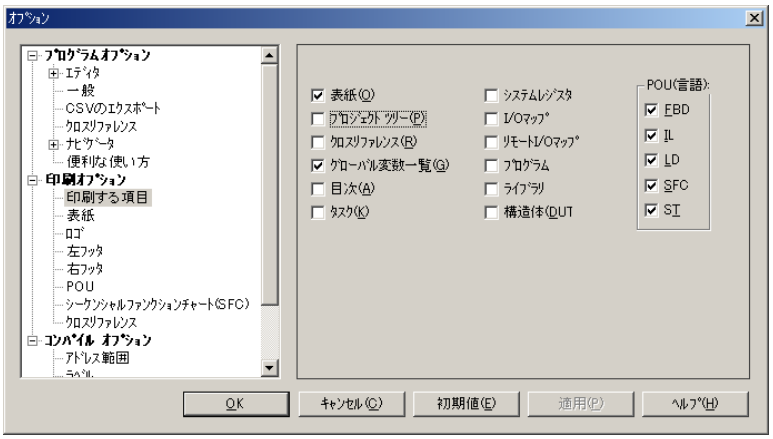
検索は、全角と半角を別の文字として行います。

8.3 印刷を実行する

印刷を実行する前に、出力する印刷内容を設定します。
(ラダー図、システムレジスタ設定など)
「拡張機能」メニューから「オプション」を選択し、「印刷オプション」を選択すると、以下のダイアログが表示されます。

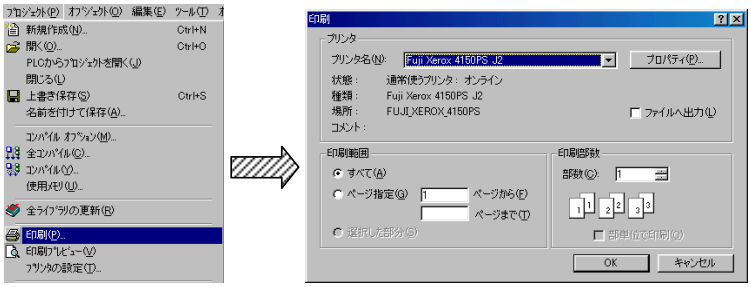


「印刷する項目」をクリックし、印刷内容を選択してください。



「印刷項目」で設定された内容が、印刷実行時に出力されます。

印刷は、「プロジェクト」メニューの「印刷」で実行されます。



FPWIN Pro 応用編

9章

変数を使ってプログラムを作成する

9.1 概要

■変数とは

変数とは、PLCにある入出力や、PLC内部のメモリ領域（DTやWR等）を識別するために使用する名称です。変数は、物理アドレスの代わりにプログラム上で使うことができます。

例えば、入力No.0を使ってスイッチの入力を確認するために、入力No.0から値を読み込む場合、IL（インストラクションリスト）では次のように記述します。

```
LD          X0
```

しかしながら、この記述方法には、次の2つの欠点があります。

・プログラムが複雑になってくると、それぞれの入出力の機能や内部メモリの割り当て内容を覚えておくのが難しくなる。

・アドレスの変更やPLC機種の変更に際し、プログラムを広範囲にわたって修正する必要がある。

Control FPWIN Proを使えばこのような問題は起こりません。例えば、変数宣言したSW0_ONをX0に割り付ける場合、プログラムへは次のように記述します

```
LD          SW0_ON
```

■グローバル変数とローカル変数

グローバル変数とローカル変数には、次のような違いがあります。

グローバル変数：　グローバル変数は、次のような場合に使用します
- ・外部入出力に変数を割り当てる場合
- ・プロジェクト内の複数のPOUで共通に変数を使用する場合
- ・外部機器との通信用に、変数に固定したPLC内部メモリアドレスを割り当てる場合

ローカル変数：　ローカル変数は、1つのPOU内でのみ有効です。変数に対する内部メモリの割り当ては、コンパイル時に自動的に行なわれます。ローカル変数は、各POUのPOUヘッダで宣言します。

変数全体の働きを把握するために、クロスリファレンスにより変数の宣言と属性のすべてを一覧表示できます。

9.2 グローバル変数

グローバル変数は、プロジェクトナビゲータのグローバル変数リストで宣言します。
外部入出力またはPLC内部メモリのアドレスを割りつけることができるのは、グローバル変数です。ローカル変数にアドレスを指定することはできません。

9.2.1 グローバル変数の宣言内容

変数宣言の手順の前に、グローバル変数リストにある各フィールドについて簡単に説明します。

■グローバル変数リストで指定できる内容

クラス

グローバル変数は3種類の変数種別に分けられます。

VAR_GLOBAL： 非保持型のグローバル変数です。
電源OFF時またはPROG.モード変更時に値は保持されません。
RUNモード切替時、[初期値]に指定した内容が設定されています。

VAR_GLOBAL_RETAIN： 保持型のグローバル変数です。
電源OFF時またはPROG.モード変更時にも値が保持されます。

VAR_GLOBAL_CONSTANT： 定数として使用するグローバル変数です。
VAR_GLOBAL_CONSTANTにはアドレスを指定することはできません。

変数名

プログラムで使用される識別名です。半角英数字100文字以内で指定します。
(Ver.5未満では、半角英数字16文字以内)

PLC アドレスおよび IEC アドレス

PLCアドレスは、変数に割り付ける物理アドレス（X0，Y0，DT0など）です。
アドレスは、PLCの外部入出力用として、あるいは、データレジスタの指定が必要な場合のみ登録します。不要なアドレス登録はしないことをおすすめします。
IECアドレスは、PLCアドレスから自動的に算出されますので、ユーザが入力する必要はありません。

データ型

アドレスを登録するとデフォルトのデータ型（例: 入力/出力に対しては「BOOL」）が表示されます。別のデータ型を選択することもできます。

省略文字	意味	範囲	データ長
BOOL	真理値	0（FALSE）または 1（TRUE）	1 ビット
INT	整数	−32,768～32,767	16 ビット
DINT	倍精度整数	−2,147,483,648～2,147,483,647	32 ビット
WORD	ワード	0～FFFF(H)	16 ビット
DWORD	倍精度ワード	0～FFFFFFFF(H)	32 ビット
REAL	実数	−1.175494 × 10E-38～−3.402823 ×10E38 と 1.175494 × 10E-38 ～ 3.402823 × 10E38	32 ビット
STRING	文字列	1～255 文字（ASCII）	1～255 バイト
TIME	時間（間隔）	T#0.01s～T#327.67s	16 ビット*1
TIME	時間（間隔）	T#0.01～21.474.836, 47s	32 ビット*2

*1：FP1 FPM FP3
*2：FP0 FPΣ FP2/2SH FP10SH

初期値

選択されたデータ型に対応したデフォルトの初期値が自動的に表示されます。この値は、PLC起動時に変数に割り当てられる値です。必要に応じて初期値を変更してください。

Autoextern（外部変数自動登録）

このフィールドにチェックを入れると、グローバル変数が、これ以後作成されるすべてのPOUヘッダに、外部変数（VAR_EXTERNAL）として自動的に挿入されます。
「拡張機能」メニューの「オプション」→「プログラムオプション」→「エディタ」→「宣言エディタ」をクリックして、「外部変数自動登録機能を全POUに直ちに適用する」のチェックボックスにチェックを入れると、グローバル変数の登録時に、全てのPOUヘッダに自動的に外部変数として挿入されます。

コメント

変数に関するコメントを記入します。

9.2.2　グローバル変数を宣言する

■操作手順

1. **プロジェクトナビゲータのグローバル変数をダブルクリックします。**
2. **グローバル変数リストで必要事項を各フィールドに登録します。**
   ＜Tab＞キーを押すと次のフィールドへ移動します。
3. **<Shift>キーと<Enter>キーを同時に押すか、**  **をクリックします。**
   リストの一番下に新しい宣言行が現れます。

◆ご 注 意 !

・変数名の先頭に数字は使用できません。
・PLCの物理アドレス(X0, Y0等)は、変数名としては使用できません。
・VAR_GLOBALの初期値は、PLCをPROGモードからRUNモードへ切り替えた時に設定されます。
・VAR_GOLBAL_RETAINの初期値は、PLCにプログラムをダウンロードして、最初にRUNモードになったときだけ変数に割り当てられます。「プロジェクト」メニュー →「コンパイルオプション」をクリックして開いたダイアログボックスの「ユーザ領域の保持型変数を初期化しない」を有効にすると、アドレスが割り付けられている変数は再初期化されません。

◆ここがポイント!

・「編集」メニュー→「変数の新規作成」→「先頭」（直前/直後/最後）をクリックして、グローバル変数リストの任意の位置に新しい宣言行を挿入することができます。

・「拡張機能」メニュー→「オプション」→「プログラムオプション」→「エディタ」→「宣言エディタ」の「新しい宣言が挿入されたとき、変数名を自動的にインクリメントする」を有効にしておくと、新規宣言の追加時に変数名とアドレスがコピーされます。変数名に数字の1が付加されます。すでに変数名の最後が数字のときは、その数字に1が加算されます。変数アドレスにも1が加算されます。「新しい宣言が挿入されたとき、変数名を自動的にインクリメントする」が有効のときだけ、「コメントと型情報をコピーする」を有効にすることができます。

・また、「拡張機能」メニュー→「オプション」→「プログラムオプション」→「エディタ」→「宣言エディタ」の中の「新しい宣言が挿入されたとき、変数名を自動的にインクリメントする」と「コメントと型情報をコピーする」を有効にしておくと、「編集」メニュー→「変数の新規作成」→「直後」をクリックして新しい宣言行を挿入するときに、現在の宣言行の内容がコピーされます。

・「編集」メニュー→「元に戻す」（切り取り/コピー/検索）などで、基本的なテキスト編集機能を使用することができます。

v 5

・最上段の「クラス」、「変数名」、「データ型」、「初期値」、「コメント」の各々をクリックすることにより各項目に対してアルファベット順に並べ替えることができます。

9.3 ローカル変数

ローカル変数は、POU(プログラム構成単位)ヘッダで宣言され、対応するPOUボディでのみ使用できます。これらは他のPOUヘッダで同じ変数名を宣言した場合、違う変数として扱われます。POUヘッダには、グローバル変数リストから挿入された変数と、POUヘッダで宣言されたローカル変数が含まれています。

9.3.1 POUヘッダにグローバル変数を挿入する

グローバル変数の宣言時に「Autoextern」フィールドにチェックを入れておくと、グローバル変数が以後新規作成されるPOUのヘッダに挿入されます。
「Autoextern」」フィールドにチェックを入れていない場合でも、あとで拡張機能・外部変数の宣言をクリックして、選択したグローバル変数を全てのPOUヘッダに外部変数として登録できます。

■操作手順

1. **プロジェクトナビゲータの「グローバル変数」をダブルクリックします。**
   グローバル変数リストが開きます。
2. **任意の変数が宣言されている行を選択します。**
3. **「拡張機能」→「外部変数の宣言」をクリックします。**
   確認メッセージが表示されます。
4. **「OK」をクリックします。**
   選択されていたグローバル変数は、プロジェクトナビゲータにあるすべてのPOUのヘッダにVAR_EXTERNALとして登録されます。

◆ご 注 意 !

・「拡張機能」メニュー→「外部変数の宣言」の操作に対しては、「編集」メニュー→「元に戻す」が機能しません。
・「拡張機能」メニューの「オプション」→「コンパイルオプション」→「アドレス範囲」で、コンパイラが変数用に使用するアドレス領域を指定することができます。グローバル変数に割り付けるアドレスはユーザエリアの領域を指定してください。

9.3.2 挿入された変数に適用される種別

グローバル変数リストからPOUヘッダに挿入された変数には、3種類の変数種別が適用されます。

VAR_EXTERNAL

グローバル変数リストで宣言されたVAR_GLOBAL変数を指しています。VAR_EXTERNALは、PRGかFBタイプのPOUのヘッダでのみ登録可能です。PLCをPRGモードからRUNモードに切り替えたときや、電源ON時には、VAR_GLOBALの初期値が割り当てられます。

VAR_EXTERNAL_CONSTANT

グローバル変数リストで宣言されたVAR_GLOBAL_CONSTANT変数を指しています。
VAR_EXTERNAL_CONSTANTは、PRGかFBタイプのPOUのヘッダでのみ指定可能です。このクラスの変数は、PLCのメモリにアドレスの割付けをしませんが、プログラムコードの中に定数を挿入します。

VAR_EXTERNAL_RETAIN

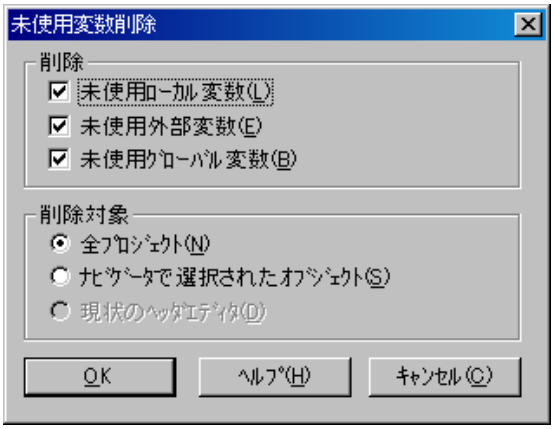
グローバル変数リストで宣言されたVAR_GLOBAL_RETAIN変数を指しています。
VAR_EXTERNAL_RETAINは、PRGかFBタイプのPOUのヘッダでのみ登録可能です。
「プロジェクト」メニュー→「コンパイル」オプションをクリックして開いたダイアログボックスの「ユーザ領域の保持型変数を初期化しない」を有効にすると、保持型領域に割り付けられている変数は再初期化されません。

9.3.3 POUヘッダからグローバル変数を消去する

「拡張機能」メニュー→「未使用変数削除」をクリックして、POUボディで使用されていない外部変数(VAR_EXTERNAL)をそのPOUヘッダから取り除くことができます。

■操作手順

1. **プロジェクトナビゲータから、削除する外部変数があるプロジェクトやPOUを選択します。**
2. **「拡張機能」メニューの「未使用変数削除」をクリックして、以下のダイアログを表示します。**



3. **削除欄にある「未使用外部変数」を選択します。**
4. **「削除対象」の下のオプションを選択します。**
5. **「OK」をクリックします。**

◆ご 注 意 !

ユーザライブラリのPOUは、この操作で削除できません。

9.3.4 ローカル変数の宣言内容

ローカル変数は、POUヘッダに宣言します。ローカル変数のアドレスは、コンパイラにより自動的に割り当てられます。ローカル変数の場合、7種類の変数種別が適用されます。
選択できる変数種別はPOUのタイプ（PRG, FUN, FB）により異なります。

VAR
演算処理の途中結果の保存などの用途として、各POUにおいて宣言できる変数です。VARの値は1つの処理の実行から次の処理の実行まで変わらず残っています。PLCがPROGモードからRUNモードへ切り替わった時や、電源ON時に、VARに初期値がセットされます。

VAR_CONSTANT
各POUにおいて宣言可能な定数変数です。VAR_CONSTANTにはアドレスがありませんが、定数がプログラムコードの中に挿入されます。

VAR_RETAIN
保持型のローカル変数です。
電源OFF時またはPROG.モード時にも値が保持されます。

VAR_INPUT
ファンクションやファンクションブロックで必要なパラメータの入力に使用する変数です。呼び出されたPOUが変数値をファンクションやファンクションブロック（PRGを除く）へ転送します。VAR_INPUTは、ファンクションもしくはファンクションブロックに対応するヘッダで宣言されます。入力変数の値は読み出せますが、書き込むことはできません。（強制入出力を除く）

VAR_OUTPUT
ファンクションブロックだけで使用される出力変数です。PLCがPROGモードからRUNモードへ切り替わった時や、電源ON時に、VAR_OUTPUTの初期値が設定されます。

VAR_OUTPUT_RETAIN
ファンクションブロックだけで使用可能な保持型の出力変数です。電源OFF時またはPROG.モード時にも値は保持されます。「プロジェクト」メニュー→「コンパイル」オプションをクリックして開いたダイアログボックスの「ユーザ領域の保持型変数を初期化しない」を有効にすると、アドレス割付けをした変数は初期化されません。

VAR_IN_OUT
入出力変数は、入力変数と出力変数の両方の役割を果たすもので、ファンクションブロックでのみ使用できます。

9.3.5 ローカル変数を宣言する

POUヘッダでのローカル変数の宣言は以下の手順で行って下さい。

■操作手順

1. **「編集」メニュー→「新規」→「POU」をクリックして、新規のPOUを作成します。POUの名前、型、プログラム言語を選択します。**
2. **プロジェクトナビゲータに表示された新規のPOUの名前をダブルクリックします。**
3. **POUに表示されたヘッダをダブルクリックします。**
4. **「クラス」フィールドの中の1つをクリックします。**
5. **任意の変数名をクリックして選択し、<Tab>キーを押します。**
6. **変数名を登録し、<Tab>キーを押します。**

7. **「データ構造」の任意のデータ型を選択します。**
8. **任意のライブラリを選択します。**
9. **「データ型」に表示されている任意のデータ型を選択します。**
10. **「OK」ボタンをクリックしまし、<Tab>キーを押します。**
    選択されたデータ型のデフォルトの初期値が自動的に表示されます。
    必要に応じて初期値を書き換えることができます。
11. **<Tab>キーを押します。**
12. **コメントを入力して<Enter>キーを押します。**
13. **<Shift>キーと<Enter>キーを同時に押します。**
    今登録した宣言行がPOUヘッダの最後の行ならば、その下に新たに宣言行が追加されます。「編集」メニュー→「変数の新規作成」またはツールバーのアイコンを用いて宣言行を追加することも可能です。

◆ここがポイント!

FBDとLDでは、「拡張機能」メニュー→「オプション」→「プログラムオプション」→「エディタ」→「LD/FBD」にある「新規変数入力時に変数定義ダイアログを開く」を有効にすると、POUボディで直接新しい「変数名」を宣言できます。この場合、POUヘッダでまだ宣言されていない変数名をPOUボディに入力して＜Enter＞キーを押すと、「変数の選択(新規作成モード)」ダイアログボックスが自動的に開き、変数の宣言ができます。

◆ご 注 意 !

・変数名の先頭に数字は使用できません。
・PLCアドレス(X0, Y0,など)は固有の用語なので、グローバル変数やローカル変数の変数名としては使用できません。

9.4　プログラム上に変数を配置する

グローバル変数リストやPOUヘッダでの変数宣言後、「変数の選択」ダイアログボックスを使ってプログラム上に変数を割り付けることができます。次の手順では、編集ウィンドウにPOUボディがすでに表示されているものとします。

■操作手順

1. **LDとFBDの場合：変数名フィールド**  **をクリックします。**
   ILの場合：オペランド部にカーソルを置きます。
2. **ツールバーの**  **をクリックします。または<F2>キーを押すかマウスの右ボタンをクリックします。**

左のダイアログが表示されます。

3. **「ライブラリ」にある宣言場所をクリックします。**
   編集中のPOUのヘッダ「ヘッダ」、グローバル変数リスト「グローバル変数」で宣言された変数を表示します。
   「＜全て＞」をクリックすると、全ての場所で宣言された変数を表示します。
4. **データ構造を選択します。**
   ここでクラスごとに分類されたデータ型が、「データ型」選択フィールドの表示されます。
   データ型：基本型（INT, WORDなど）、構造体（DUT）、
   　　FB（ファンクションブロック）
5. **データ型を選択します。**
   表示されている使用可能な変数のデータ型を、「データ型」選択フィールドで指定します。「ライブラリ」の「グローバル変数」を選択した場合、例えばデータ構造は「基本型」が、データ型は「ARRAY」が選択されると、グローバル変数リストで宣言された配列型変数が「変数」の一覧に表示されます。
6. **任意の変数をクリックします。**
   「詳細」に選択された変数のパラメータが表示されます。
7. **選択された変数をダブルクリックするか、「ボディに挿入」ボタンをクリックします。**
   POUボディのカーソル位置に、選択された変数が挿入されます。グローバル変数を選択した時、その変数がまだPOUヘッダに登録されていない場合は自動的に登録されます。

◆ここがポイント!

PLCのプログラムに変数を挿入する場合、POUヘッダに未登録のグローバル変数が使用できます。

■操作手順

1. 「変数の選択」ダイアログボックスの「ライブラリ」から、「グローバル変数」を選択します。

2. 右の一覧から挿入するグローバル変数を選択します。

3. 「ボディに挿入」ボタンをクリックします。

選択された変数は、編集状態のPOUのヘッダにコピーされます。

「広げる」ボタンをクリックすると、「変数の選択」ダイアログボックスが右に広がり、変数の変更や新規の変数宣言ができます。

9.5 変数を変更する

変数名、データ型などのパラメータの変更は、グローバル変数ではグローバル変数リストの中で、ローカル変数は個々のPOUヘッダの中で実行します。変数に関するすべての変更内容は、現在開いているプロジェクトのPOUヘッダとボディに反映させることができます。または、「拡張機能」メニュー→「変数の更新」をクリックして開くダイアログボックスで、変更内容を更新できます。このダイアログボックスでは、変更内容を更新するか否かを、各POUのヘッダ、ボディごとに選択できます。

「変数の選択」ダイアログボックスを使ってPOUボディから直接、グローバル変数やローカル変数の特定のパラメータを変更できます。

9.5.1 POUヘッダやグローバル変数リストの変数を変更する

グローバル変数やローカル変数の変更内容をPOUで有効にする方法は、以下の2つがあります。

・全POUヘッダ／ボディを自動更新する。
・各POUヘッダ／ボディごとに更新する。

■POU ヘッダやグローバル変数リストでの変数の変更

「拡張機能」メニュー→「オプション」→「プログラムオプション」→「エディタ」→「宣言エディタ」で、「保存時に変更された宣言を自動的に更新する」を有効にすると、グローバル変数リストやPOUヘッダにある変数のすべての変更内容は、開かれているプロジェクトの該当する変数を使用しているPOUヘッダ／ボディの全てで更新されます。

◆ご 注 意 !

・すべての変更内容が更新されるのは、グローバル変数リストやPOUヘッダに変更内容が登録されるときです。
・グローバルリストの変更は、グローバル変数リストでのみ可能です。POUヘッダでグローバル変数を変更するには、「拡張機能」メニュー→「オプション」→「プログラムオプション」→「エディタ」→「宣言エディタ」をクリックして、「保存時に変更された宣言を自動的に更新する」を無効にします。

■POU ヘッダとボディを個別に更新する

「拡張機能」メニュー→「変数の更新」をクリックして開いたダイアログボックスで、開いているプロジェクトのPOUにあるヘッダやボディごとに、個々の変数の変更内容を更新するか否かを選択することができます。

グローバル変数の更新

■操作手順

1. **プロジェクトナビゲータの「グローバル変数」をダブルクリックします。**
2. **新しい名前の登録などで変数を変更します。**
3. **変更された宣言行を選択します。**
4. **「拡張機能」メニュー→「変数の更新」をクリックします。**
   変更されたグローバル変数を使っている各POUヘッダ／ボディごとに更新するかを確認するメッセージボックスが開きます。
5. **「はい」をクリックすると、各POUヘッダ／ボディごとに更新されます。**
   あるいは、「いいえ」をクリックして、変更をキャンセルします。
   次のPOUヘッダ／ボディに対する確認メッセージボックスが開きます。

◆ご 注 意 !

グローバル変数は、グローバル変数リストで直接変更する以外の変更方法はありません。POUヘッダでグローバル変数を変更しても、「拡張機能」メニュー→「変数の更新」は無効表示になって更新できません。

POU ヘッダでの変数の更新

■操作手順

1. **プロジェクトナビゲータにあるPOUをダブルクリックし、ヘッダを開きます。**
2. **新しい名前の登録などで変数を変更します。**
3. **変更された宣言行を選択します。**
4. **「拡張機能」メニュー→「変数の更新」をクリックします。**
   POUボディの、変更された変数を更新するかどうかの確認メッセージボックスが開きます。
5. **「はい」をクリックして更新します。**
   POUボディで使われている該当する変数が、すべての個所で自動的に更新されます。

9.5.2 「変数の選択」ダイアログボックスで、変数を変更する

■操作手順

1. **POUボディの変数をクリックします。**
2. **<F2>キーを押すか、または**  **をクリックします。**
   「変数の選択」ダイアログボックスが開きます。
3. **「広げる」ボタンをクリックします。**
   「変数の選択」ダイアログボックスが右側に広がり、選択された変数のパラメータが表示されます。

「変数」の下に何も表示されていなければ､左側にある「データ型」から「ANY」または任意のデータ型をクリックします。

4. **任意の変数パラメータの内容を変更します。**
   このダイアログボックスで、「変数名」の変更はできません。新しい変数名を記入すると、「更新」ボタンが「宣言」ボタンに変わり、新規の変数が宣言できます。
5. **更新ボタンをクリックします。**
   変数の更新の確認メッセージボックスが表示されます。
6. **「はい」ボタンをクリックします。**
   変数は変更されました。ダイアログボックスの左側にある「詳細」に、変更された変数のパラメータが表示されます。
   さらに別の変数を変更するには、「変数」の下の一覧から任意の変数を選択して、ダイアログボックスの右側にあるフィールドにパラメータを表示させます。変数について表示されている「変数」、「ライブラリ」、「データ構造」、「データ型」の内容を登録します。

9.6 変数が使用しているPLCアドレスを取得する

ローカル変数を使用してプログラムを組んだ場合、実際にその変数に割り当てられるPLCアドレスは、コンパイル後にしかわかりません。変数に対して、コンパイル後に割り当てられるPLCアドレスを取得する方法としてFP Tool Libraryライブラリに以下の関数が用意されています。

- `Adr_Of_Var_I` ：入力変数が使用するPLCアドレスの先頭アドレスを取得。
- `Adr_Of_Var_O` ：出力変数が使用するPLCアドレスの先頭アドレスを取得。
- `AdrLast_Of_Var_I` ：入力変数が使用するPLCアドレスの最終アドレスを取得。
- `AdrLast_Of_Var_O` ：出力変数が使用するPLCアドレスの最終アドレスを取得。
- `Adr_Of_VarOffs_I` ：入力変数が使用するPLCアドレス（オフセット付）を取得。
- `Adr_Of_VarOffs_O` ：出力変数が使用するPLCアドレス（オフセット付）を取得。
- `AdrDT_Of_Offs_I` ：入力変数が使用するDT(データレジスタ)のアドレスを取得。
- `AdrDT_Of_Offs_O` ：出力変数が使用するDT(データレジスタ)のアドレスを取得。
- `AdrFL_Of_Offs_I` ：入力変数が使用するFL(ファイルレジスタ)のアドレスを取得。
- `AdrFL_Of_Offs_O` ：出力変数が使用するFL(ファイルレジスタ)のアドレスを取得。

◆ご 注 意！

- ST（ストラクチャードテキスト）で使用するときには、入力変数・出力変数の区別がなくなります。（各関数名の最後の `_I` や `_O` がなくなります）

- 上記以外にも、変数のサイズやメモリエリアを取得する関数があります。
  詳しくは、ヘルプのFP Toolライブラリ（FP Tool Library）をご参照ください。

10章

ファンクション（FUN）／ファンクションブロック（FB）を作成する

10.1 概要

ファンクション／ファンクションブロックは、一連の処理を部品化し登録することにより、あたかも一つの命令として使うことができる機能です。

■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

プログラム中で呼び出されると、一連の処理を実行して結果を返す処理です。
ファンクションでは、入力に対して処理結果は一意的に決まります。
ファンクション内部ではグローバル変数は使用できません。

ファンクションブロック

ファンクションと同様に一連の処理を行いますが、値を記憶しておくメモリ領域をファンクションブロック自身にもっています。そのため、同じ入力値に対して異なる結果を生じます。また、プログラム中に何度でも同じファンクションブロックを使用することができます。
その際、各々のファンクションブロックの呼び出しに名前（インスタンス名）をつけて区別します。プログラム中にいくつか配置することができ、番号をつけて区別するタイマ命令と同じような概念です。
ファンクションブロックを呼び出す回数は、PLCのSUB命令の数に制限されます。

10.2 ファンクション（FUN）を作成する

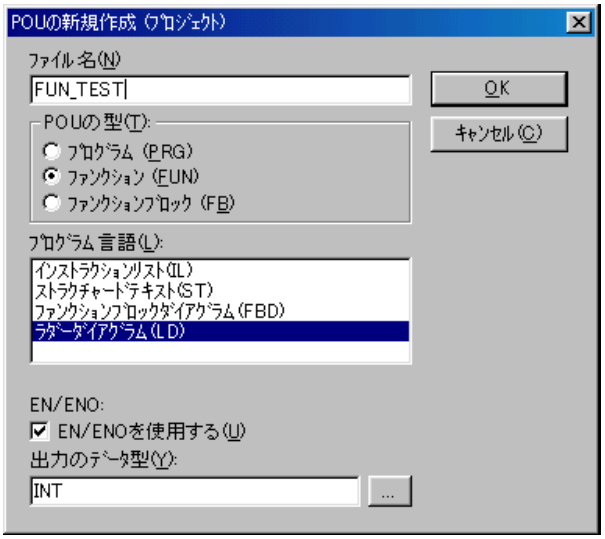
ファンクションの作成は、ファンクション（FUN）型のPOUを新規作成することから始めます。ヘッダ、ボディはこれまでと同様に編集します。

次の手順にしたがって、「３つの値を入力するとその平均値を返す」というファンクションを、ラダーダイアグラムを使って作成します。

■操作手順

1. ファンクション（FUN）型のPOUを新規作成します。

POUの型としてファンクション（FUN）を選択します。



「出力のデータ型」では、このファンクションを実行したときに出力される結果の「型」を指定します。

この例では「INT」を指定します。

「EN/ENOを使用する」が指定されると、EN/ENO端子付のファンクションが生成されます。ENはファンクションを実行するためのトリガ入力として、またENOは、次のファンクションへの実行トリガの出力として使用するものです。

2. 以下のようにPOUヘッダで任意の変数を宣言します。

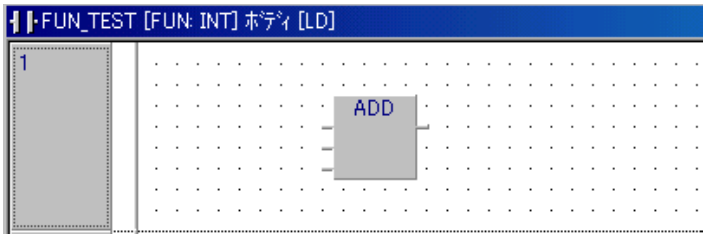
FUN_TEST

	クラス	変数名	データ型	初期値	コメント
0	VAR_INPUT	input_1	INT	0	
1	VAR_INPUT	input_2	INT	0	
2	VAR	input_3	INT	0	
3	VAR	work	INT	0	
4	VAR				

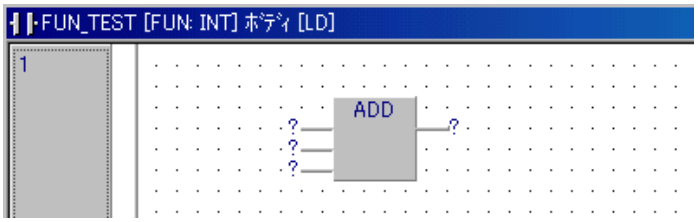
3. **まずは、3つの入力値を合計するブロックを作成します。**
4. **ボディをクリックし、ツールバーの**  **をクリックして、以下のダイアログを表示します。**

ADDを表示させるには、タイプで演算子を選択してください。

5. **ADDを選択し、「ピンの数」を「3」に設定します。（ピンの数は、ボディに配置後でも変更できます）**
6. **「挿入」ボタンをクリックし、ボディに配置します。**



7. **入力用の変数（ツールバー**  **）を左に、出力用の変数（ツールバー**  **）を右に配置します。**



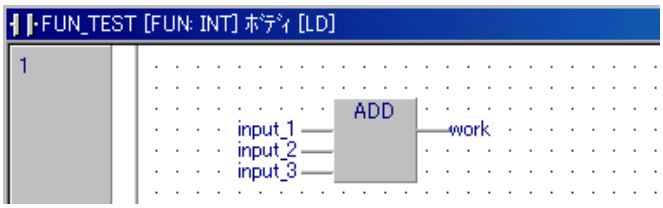
v 5

Ver.4のときは、オペランド入力のために、入力変数・出力変数のピンを接続する必要がありましたが、Ver.5では、はじめから各応用命令に入出力変数のピンが接続されています。

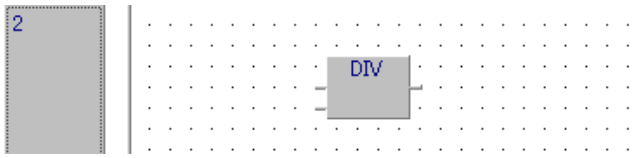
8. それぞれの変数ボックスをマウスでクリックして“？”を反転状態にした後、F2キーを押します。

以下のダイアログが表示されます。

9. ここでinput_1からinput_3を入力側に、3つの入力値の合計を保存するエリアとしてworkを選択して下さい。

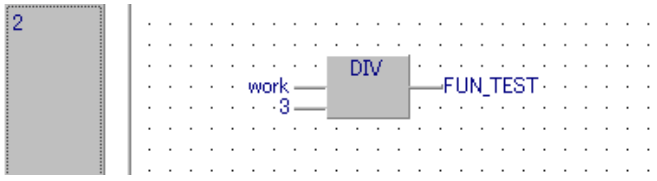


10. 次に、平均値を求めるブロックを作成します。4. と同様の操作でダイアログを表示させて、「DIV」を選択しボディに挿入します。（次のブロックに挿入して下さい）

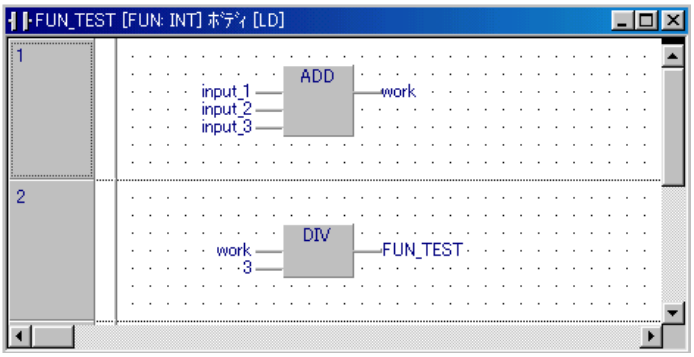


11. それぞれのピンに以下のように変数を配置して下さい。

ファンクションの出力は、ファンクション名を指定してください。出力変数をヘッダ内で定義する必要はありません。



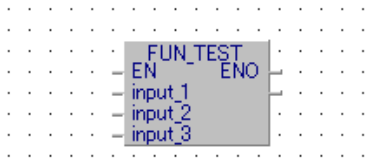
12. 以上で、プログラムは完成です。



13.「オブジェクト」メニューの「上書き保存」を選択し、プログラムを保存して下さい。

チェックが問題なく終了すれば、今回作成したファンクションは他のPOUからでも呼び出せる部品として登録されます。

【ボディに挿入されたファンクション】



10.3 ファンクションブロック（FB）を作成する

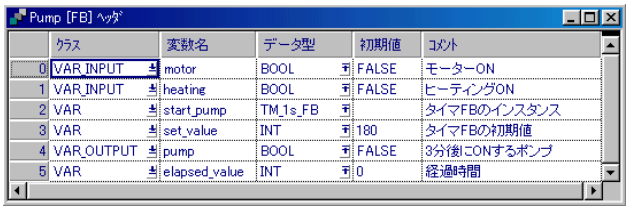
ファンクションブロック（FB）の作成は、ファンクションブロック（FB）型のPOUを新規作成することから始めます。ヘッダ、ボディの編集はこれまでと同様です。

＜例＞

次の手順にしたがって、モータ信号と加熱信号の入力が3分間ONし続けるごとに、ポンプスイッチが1回ONになるような、タイマを使った簡単なプログラムを作成します。

■操作手順

1. **ファンクショングロックダイアグラムでPOUを新規作成します。**
2. **以下のようにPOUヘッダで使用する変数をすべて宣言します。**



3. **「拡張機能」メニュー→「オプション」→「プログラムオプション」→「エディタ」→「LD/FBD」」をクリックします。**
   表示されたダイアログボックスの「自動接続モード時に入力／出力変数ボックスを自動追加」が有効になっているかを確認してください。
4. **ボディをクリックし、ツールバーの をクリックします。**
   「OP/FUN/FBの選択」ダイアログボックスが開きます。

編集ウィンドウの任意の位置で、＜Ctrl＞キーを押しながら挿入すると、何回かクリックすることで複数のプログラム記号を挿入できます。

5. **「タイプ」欄から「演算子」を選択します。**
6. **「OP/FUN/FB」の一覧から「AND」を選択します。**
7. **「挿入」ボタンをクリックするか、または選択した演算子をダブルクリックします。**

8. 編集ウィンドウの任意の位置でクリックします。

編集ウィンドウに演算子が挿入されます。

最初の入力変数の変数名フィールドに(「?」)が表示され、編集可能な状態になっています。

手順16の説明のように、すぐに変数の割付けができます。

次に、TM_1s_FBファンクションブロックを挿入します。

9. 「OP/FUN/FBの選択」ダイアログボックスの「ファンクションブロック」を有効にします。

10. 「OP/FUN/FB」の下の一覧から「TM_1s_FB」をクリックします。

11. 「挿入」ボタンをクリックするか、または選択したファンクションブロックをダブルクリックします。

12. 編集ウィンドウの任意の位置をクリックします。

TM_1s_FBファンクションブロックは、編集ウィンドウへ挿入されます。

入力フィールドは、FBインスタンス名の入力状態になっています。ここで、POUヘッダですでに宣言されているFBインスタンスを割り付けます。

POUボディにファンクションブロックを挿入したあとで、FBインスタンスの名前を直接入力フィールドに入力して<Enter>キーを押し、次に開く「変数の選択（新規作成モード）」ダイアログボックスの中にある広げるボタンをクリックすると、新規のFBインスタンスを宣言することもできます。

<Tab>キーを押して、次の入力フィールドに移動します。

13. <F2>キーを押します。

「変数の選択」ダイアログボックスが開きます。

FBタイプとして宣言されたすべてのFBインスタンスの名前は、「ライブラリ」で「<ヘッダ>」を選択すると「変数」一覧に表示されます。

14. start_pumpをクリックします。

15. 「ボディに挿入」ボタンをクリックします。

POUボディにある入力フィールドにFBインスタンスの名前が挿入されます。

入力フィールドに直接、すでに宣言されているFBインスタンスの名前を入力することもできます。

16. 変数を割り付けるために、入力フィールドをクリックして<F2>キーを押します。

POUヘッダで宣言されたすべての変数が、使用可能なデータ型ごとに「変数の選択」ダイアログボックスに表示されます。

POUヘッダで宣言された変数が「変数」の一覧に表示されない場合は、「データ型」から「ANY」か他のデータ型をクリックして選択します。

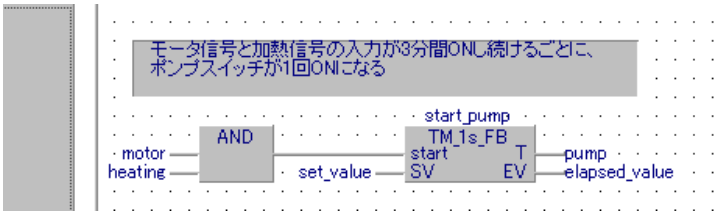
17. 「変数」から任意の変数をダブルクリックします。

POUボディの選択されている入力フィールドに変数が挿入されます。

プログラム記号が挿入された入力フィールドのすべてを順次クリックして、変数を割り付けます。

次に、編集ウィンドウのプログラム記号同士を接続します。

18. **ツールバーの  をクリックして、描画モード(ペン形のカーソル)にします。**
19. **AND演算子の出力ピンをクリックします。**
20. **start_pump の入力ピン“start”をクリックします。**
    接続線が描かれます。手動で接続線の形状を変更する場合は、線を引く途中で任意の位置でダブルクリックして「固定点」を設定できます。
    自動接続モードが無効のときのみ、固定点を解除できます。
21. **  をクリックして配置モードに切り替えます。**
    ブロックの任意の位置でコメントを挿入できます。
22. **  をクリックして編集ウィンドウの任意の位置にカーソルを置きます。**
    カーソル位置にコメント記号がつきます。
23. **任意の位置でマウスの左ボタンを押したままドラッグし、任意のサイズにコメントウィンドウを広げます。**
24. **「Switch on...」と入力し<Enter>キーを押します。**
    ＜Ctrl＞キーと＜Enter＞キーを同時に押すと改行されます。
    ウィンドウの右下の角でマウスの左ボタンを押したまま（両端が矢印のカーソル）ドラッグすることで、コメントウィンドウのサイズを変えることができます。後からコメントウィンドウを移動する事もできます。
    コメントウィンドウの左隅でマウスの左ボタンを押したまま任意の位置へドラッグすると、それにしたがってコメントウィンドウが移動します。
    これまでの手順で、以下のようなプログラムが完成しました。



25. **「オブジェクト」メニュー→「上書き保存」をクリックします。**

◆ここがポイント!

・編集中のPOUボディのユーザ側で作成したファンクション、ファンクションブロックを選択してダブルクリックすると、選択したファンクション、ファンクションブロックのヘッダ・ボディを開くことができます。
FBインスタンス名を選択、ダブルクリックすると、対応するファンクションブロックが開きます。
・チェックが問題なく終了すれば、今回作成したファンクションブロックは他のPOUからでも呼び出せる部品として登録されます。

◆ご 注 意 !

ファンクションブロックは、ファンクションと異なり、内部にデータを格納するメモリを所有しています。従って、ファンクションブロックの実行毎にインスタンス名をつけて区別する必要があります。
例えば、カウンタ機能をもつファンクションブロックを使用する場合、カウントする場所により、“counter1”, “counter2” …という別々のカウンタとして使用します。これらは機能（ロジック）は同じで内部に記憶しているカウンタ値が異なるだけです。
“counter1” や “counter2” のことをインスタンスと言います。

11章

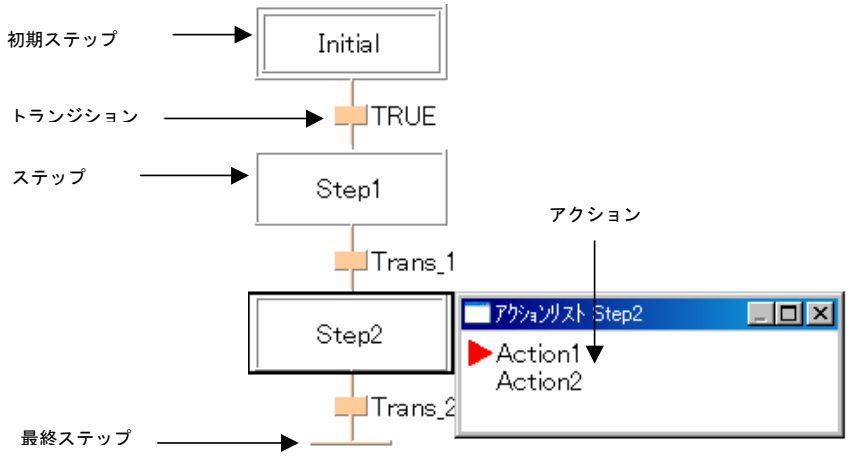
シーケンシャルファンクションチャート(SFC)でプログラムを作成する

11.1 概要

シーケンシャルファンクションチャート（SFC）では、複雑なプログラムを分かりやすく表記することができます。

11.1.1 シーケンシャルファンクションチャートを構成する要素

シーケンシャルファンクションチャート（SFC）は、一連のステップの流れを、ステップ・トランジション（遷移）・分岐等を使って表します。



■ステップ

ステップとは、上図では長方形のボックスとして表現される[Step1]，[Step2]のような部分タスクのことです。

上図を利用して制御の流れを説明します。

初期ステップ(Initial)が最初のステップとして起動し、トランジションTrans_0がONになると、Step1が起動されると同時に、初期ステップは停止します。次にトランジションTrans_1がONになると、Step2起動されると同時に、Step1は停止します。次にトランジションTrans_2がONになり最終ステップまで実行されると、再び初期ステップから一連のシーケンス処理が実行されます。

ステップごとに複数のアクション（ステップ内のプログラム）を連結させることができます。
（上図では、Step2に、アクションAction1とAction2が登録されています）

ステップがアクティブ（起動されている）のときは必ず、ステップ内の一連のアクションが実行されます。アクションが割り当てられていないステップは、直後につながっているトランジションの条件が成立するまで待ち状態となります。

IL, FBD, LD, STで作成されたアクションは、プロジェクトナビゲータのアクションプールに登録されます。同じアクションを複数のステップで使用することも可能です。

ブール型変数をアクションとして使用することもできます。この場合は、そのステップが起動している間、そのブール型変数をONします。

◆ご 注 意 !

・アクションとして登録されるプログラムでは、ラベルを使用できません。

■ステップフラグ

特定のステップが起動されている間だけONするフラグを、ステップフラグと呼びます。
ステップフラグは「ステップ名.X」で表現され、プログラム中で使用できます。

■マクロステップ

複数のステップを1つのマクロステップにまとめて、流れを見やすく表示することができます。
マクロステップは、下図のように横二重線の印がつけられています。

■トランジション

トランジションとは、条件付ジャンプのことです。
トランジションの条件が成立すると、次のステップが起動します。
トランジションは、次に示す記号で表します:

ブール型変数やIL, FBD, LD, STで作成されたプログラムを、トランジションに割り付けることができます。
ブール型変数の場合、割り付けられた変数の値がTRUEになると、トランジションの条件が成立したものとみなされます。
プログラムの場合、トランジション名と同じ名前の変数の値がTRUEになると、トランジションの条件が成立したものとみなされます。
トランジション名と同じ名前の変数は、Control FPWIN Proで自動的に宣言されます。

◆ご 注 意 !

トランジションにプログラムを割り付ける場合、以下の制限があります。

- **トランジションボディで作成できるプログラムは1ブロックのみです。**
- **「EN/END」付きファンクションは使用できません。**

11.1.2 分岐と並列分岐

■分岐

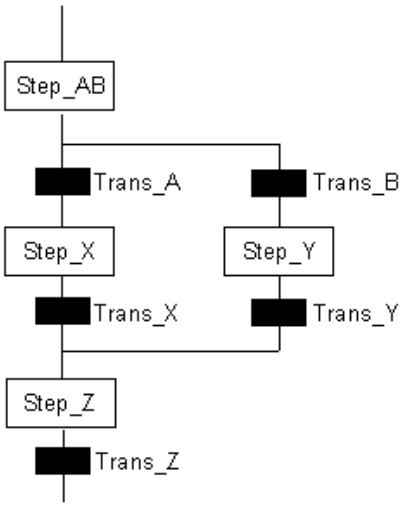
分岐の記号は、横一重線です。

2つのトランジションの内どちらかの条件(Trans_AあるいはTrans_B)が成立すると、該当する方を実行します。

両方のトランジションの条件が同時に満足すると、右側よりも左側が優先的に実行されます。

＜例＞

下図の分岐で、Trans_AとTrans_Bが同時に真になった場合、必ずTrans_A側のステップ(Step_X)だけが実行されます。

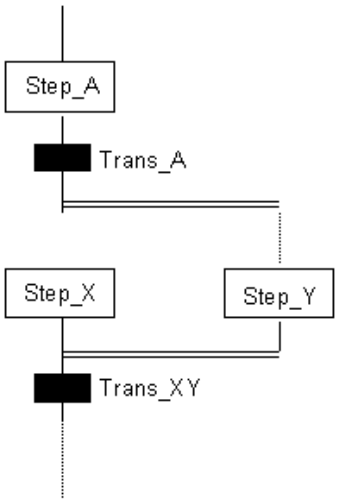


■並列分岐

並列分岐の記号は、横二重線です。

並列分岐直前のトランジションの条件が成立すると、（同時に）2つ以上の並列処理を実行します。

＜例＞



並列分岐により分岐したすべての処理は並列結合によりトランジション（Trans_XY）に再び結合されます。

◆ご 注 意 !

並列結合は、その直前のすべてのステップ（Step_X, Step_Y）が実行中であり、かつ、トランジション（Trans_XY）が成立したとき、1つのステップに結合します。

11.2 編集

SFCエディタでは、ツールバーのアイコンボタンを使ってエレメントやオブジェクトを挿入，表示，開くなどの編集ができます。
編集ウィンドウやプログラムを分割でき、ユーザが作成したプログラムから抜き出した様々なオブジェクトを同時に画面上に表示できます。

11.2.1 ツールバー上のアイコンボタン

SFCボディが開くと、以下のアイコンがツールバー上に表示されます。

■新たにオブジェクトを挿入するアイコン

「編集」メニュー→「挿入」でも同じ機能を選択できます。

-  ：トランジション付ステップを現在位置の直前に挿入
-  ：ステップを現在位置の直前に挿入
-  ：トランジションを現在位置の直前に挿入
-  ：左への分岐／並列分岐を現在位置の直前に挿入
-  ：右への分岐／並列分岐を現在位置の直前に挿入
-  ：左からの結合／並列結合を現在位置の直前に挿入
-  ：右からの結合／並列結合を現在位置の直前に挿入
-  ：ステップの直前にラベルを挿入
-  ：トランジションの直後にジャンプを挿入

■オブジェクトを表示あるいは開くアイコン

ツールメニューにある項目と同じ機能です。

-  ：マクロ構成リストを開く
-  ：ステップに連結されたアクションの編集
-  ：ステップのコメントを編集
-  ：アクション連結リストやトランジションのボディを開く
-  ：選択されたオブジェクトを開く
-  ：開かれたオブジェクトを閉じる
-  ：次の同種のエラーを検索
- 

：変数の選択ダイアログボックスを開く
（トランジションが編集状態のときのみ）

11.2.2 SFC記号の選択

ステップやトランジションを編集するには、最初に編集する位置のステップやトランジションを選択します。
選択方法には、次の3通りがあります。

・SFC記号をクリックします。
・<Shift>キーを押しながら複数のSFC記号をクリックして行きます。
・SFC記号をクリックしたあと、編集メニュー・マークをクリックします。

11.2.3 ステップ／トランジションを開く

次にオブジェクトの表示方法や開き方を説明します。

■操作手順

1.ステップを開く

すでにプログラムに割り付けられたステップは、下記のいずれかの方法にてアクション連結リスト等を開くことができます。

- 「ツール」メニュー→「オブジェクトを開く」をクリックする
- 選択したステップをダブルクリックする
- そのまま<Enter>キーを押す

上記の操作をした場合､デフォルトではアクション連結リストが開く設定になっていますが､「拡張機能」メニュー→「オプション」→「プログラムオプション」→「SFCエディタ」で開くダイアログボックスの､「SFCのボディ内でステップを開く」欄で変更することができます。

アクションリストを開く	アクション連結リスト
アクションリストの先頭アクションを開く	アクション連結リストに登録したアクションの内､先頭のアクションのボディ

2.トランジションを開く

すでにプログラムに割り付けられたトランジションは、下記のいずれかの方法でプログラムを開くことができます。

- 「ツール」メニュー→「オブジェクトを開く」をクリックする
- 選択したトランジションをダブルクリックする
- そのまま<Enter>キーを押す

まだプログラムに割り付けられていないトランジションに対して上記操作をすると、トラジションの新規作成ダイアグラムが開きますので、ここでトランジションに割り付けられるプログラムを作成します。

(PLCの内部メモリR0等をトランジションとして設定した場合は、上記操作をしても何も動作しません。)

3.マクロステップを開く

すでにプログラムに割り付けられたマクロステップは、下記のいずれかの方法で展開できます。

- 「ツール」メニュー→「オブジェクトを開く」をクリックする
- 選択したステップをダブルクリックする
- そのまま<Enter>キーを押す

11.2.4 SFC記号のプロパティの修正

「編集メニュー」→「修正」→... をクリックすると、カーソル位置のSFC記号のプロパティを修正できます。
トランジションでは、名前だけが変更できます。
一般のステップを初期ステップ、マクロステップ、最終ステップのいずれかに変更できます。

以下に、「ステップ」を「最終ステップ」に変更する方法を説明します。

■操作手順

1. 任意のステップをクリックします。
2.「編集」メニュー→「修正」→「最終ステップ」をクリックします。

SFCの最後のステップは、最終ステップと見なされます。
1つのSFCに複数の最終ステップを配置できます。
シーケンス処理が最終ステップを実行すると、初期ステップに戻って繰り返し実行されます。

11.2.5 編集ウィンドウの分割

編集ウィンドウのスクロールバーとタイトルバーに挟まれた黒い小さなバーを使って、編集ウィンドウを上下に分割できます。
一つのプログラムの内容を2つの画面に分けて、個別に表示，スクロール，モニタをすることができます。

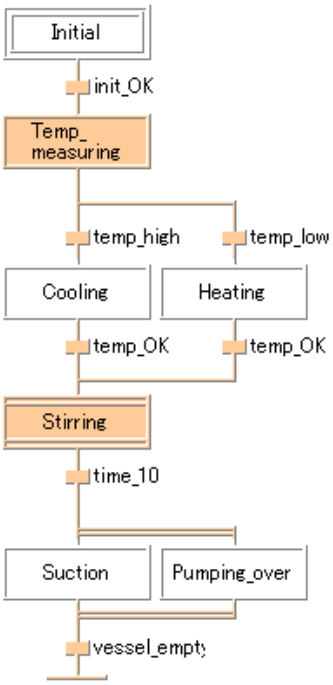
■操作手順

1. 編集ウィンドウのタイトルバーと、スクロールバーに挟まれた黒い小さなバー上にカーソルを置きます。
カーソルの両端が矢印の形になります。
2. マウスの左ボタンを押したまま、任意の位置までバーをドラッグします。
3. マウスボタンを放します。
ドラッグした位置で編集ウィンドウが分割され、プログラムが両方のウィンドウに表示されています。別々にウィンドウをスクロールできます。
必要に応じて、分割線をドラッグしてウィンドウの分割位置を調整してください。
4. プログラムの目的の場所がウィンドウ上に表示されるまでスクロールします。

編集ウィドウを閉じたり、編集ウィンドウのタイトルバーの真下まで分割線をドラッグしたりすると、ウィンドウの分割が解除されます。

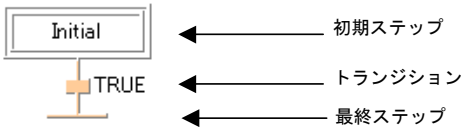
11.3 プログラムの例

ここでは実際に、SFCでプログラムを作成してみます。



■操作手順

1. **新規POUを作成します。（POUの型：PRG，プログラム言語：SFC）**
2. **プロジェクトナビゲータから、新規作成したPOU(PRG)をダブルクリックします。**
3. **SFCボディをダブルクリックします。**
   SFCボディが開きます。SFCボディ内にinitialという名前の初期ステップ，トランジション，TRUE，最終ステップが表示されます。



◆ご 注 意 !

・PLC上でプログラムの並列処理を実行する場合や、コンパイラを有効に活用するために大規模なSFCプログラムを小規模なプログラムに分割する場合に対応して、プロジェクトに複数のSFCプログラム（POU）を保有できるようになっています。
但し、タスクには複数のSFCプログラムが連続して登録される必要があります。

・手順では、ツールバーのアイコンボタンとメニューを主に使用しますが、キーボード操作でもできます。

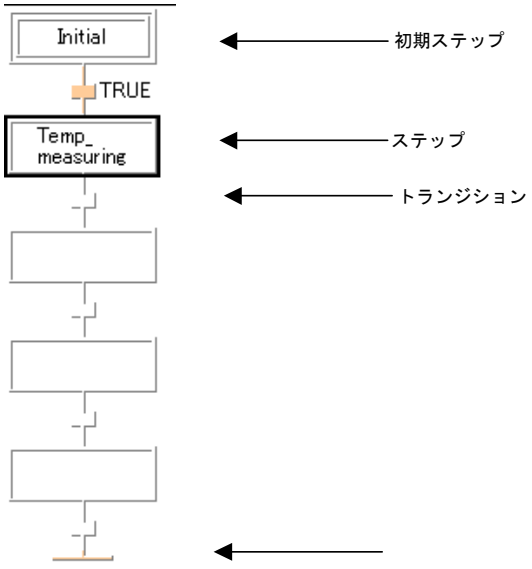
■ステップ／トランジションの挿入

4. 最終ステップをクリックします。

5. または「編集」メニュー→「挿入」→「ステップ／トランジション」をクリックします。
新規のステップと新規のトランジションが作成されます。

6. 手順5を、4回実行します。

7. 2番目のステップをクリックし、名前“Temp_measuring”を入力、<Enter>キーを押します。

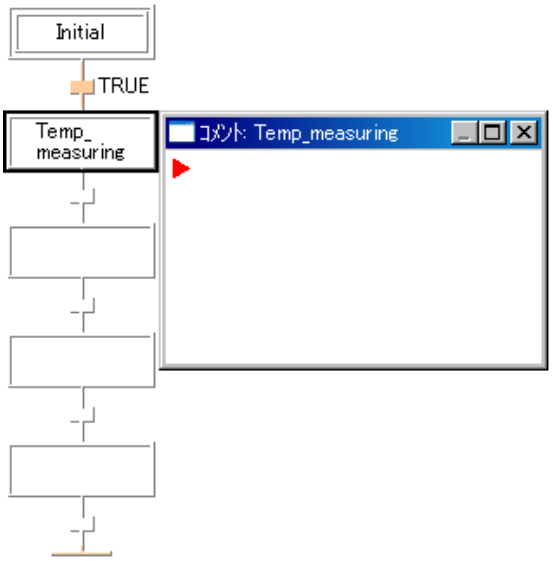


◆ご 注 意 !

Step名には日本語を使用できません。半角英数字及び‘__’を御使用下さい。

■コメントの挿入

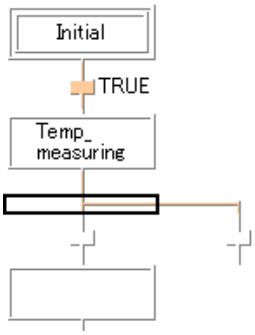
8. **または「ツール」メニュー→「ステップのコメント編集」をクリックします。**
   「コメント　ステップ名」入力フィールドが開きます。



9. **コメントを入力します。**
   <Enter>キーを押すと改行します。
10. **タイトルバーの右端にある×ボタンをクリックしてフィールドを閉じます。**
    必要に応じてステップごとにコメントを挿入できます。挿入したコメントは、「表示」メニュー→「詳細表示」を有効にするとステップ名の下に表示されます。

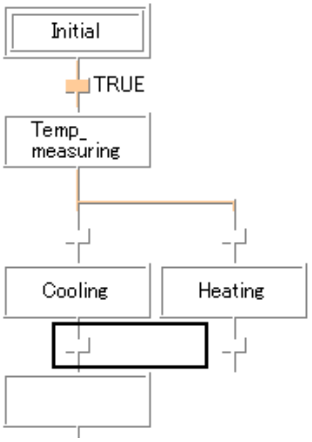
■分岐

11. **Temp_measuring ステップの直後のトランジションをクリックします。**
12. **または「編集」メニュー→「挿入」→「右分岐」をクリックします。**
    Temp_measuring ステップとトランジションの間に右分岐が挿入されます。

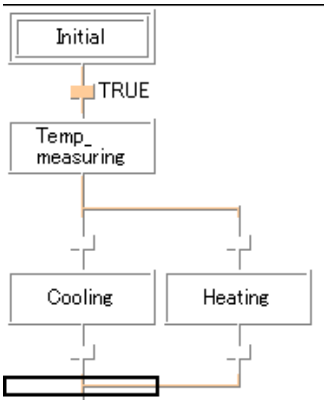


13. **右に分岐したトランジションの直後の矩形内をクリックします。**
14. **または「編集」→「挿入」→「ステップ／トランジション」をクリックします。**
15. **“Heating”と名前を入力し<Enter>キーを押します。**
16. **Heatingステップの左隣のステップをクリックします。**

17. **名前“Cooling”を入力し<Enter>キーを押します。**



18. **Coolingステップの直後のステップをクリックします。**
19. **または「編集」メニュー→「挿入」→「右結合」をクリックします。**



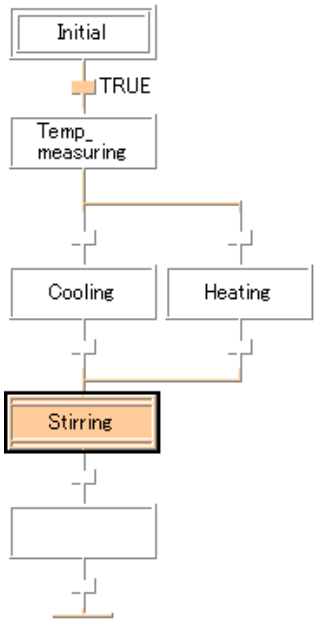
20. **分岐の直後のステップをクリックします。**
21. **“Stirring”と名前を入力し、<Enter>キーを押します。**

次にStirring ステップをマクロステップに変更します:

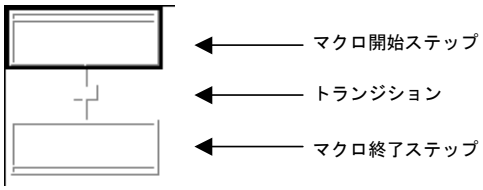
■マクロステップ

22.「編集」メニュー→「修正」→「マクロステップ」をクリックします。

複数のステップを選択し、「編集」メニュー→「マクロ」を実行することにより、マクロステップを生成することもできます。



23.  をクリックします。



マクロ内部においても、同様の編集が可能です。

■並列分岐

24. Stirring の直後のステップをクリックします。

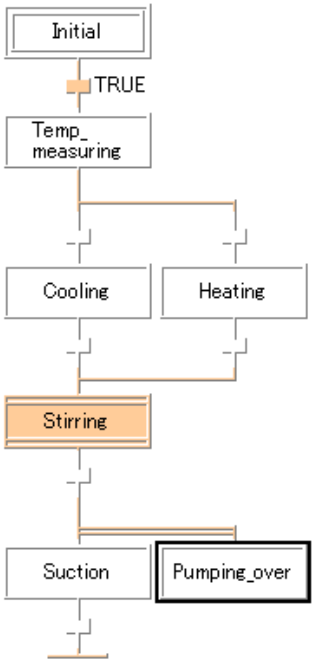
25. “Suction”と名前を入力し<Enter>キーを押します。

このステップの横に並列分岐を挿入します：

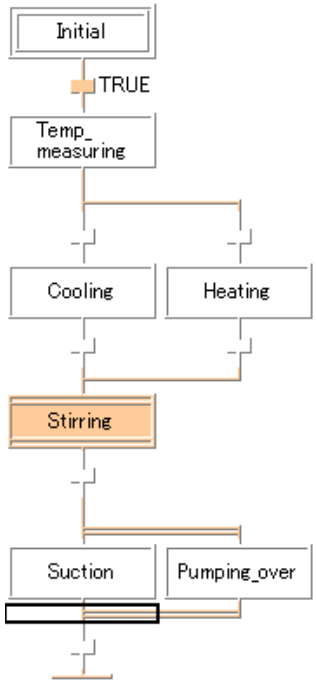
26. または「編集」メニュー→「挿入」→「右分岐」をクリックします。

27. Suctionの横にあるステップをクリックします。

28. 名前“Pumping_over”を入力し、<Enter>キーを押します。



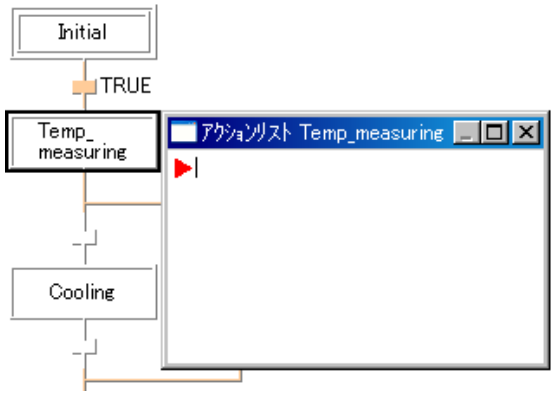
29. Sunctionの直後のトランジションをクリックします。
30. または「編集」メニュー→「挿入」→「右結合」をクリックします。



■アクションの編集

31. Temp_measuring ステップをクリックします。
32. をクリックします。

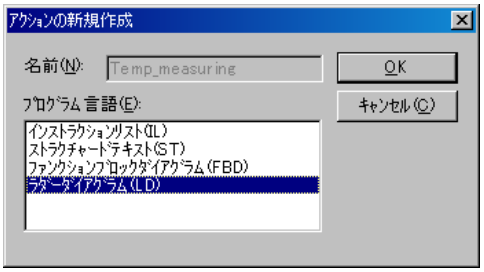
または<Enter>キーを押すか、「ツール」メニュー→「アクション連結編集」をクリックします。



「アクションリスト」ダイアログボックスが開きます。

33. アクション名(例：Temp_measureing)を入力して、 をクリックします。

（アクション名入力後に、改行しないでください）



「アクションの新規作成」ダイアログボックスが開きます。

編集済みのアクション、または宣言済みの変数をステップに割り付けるには「アクション連結リスト」にて、<F2>キーを押して「アクション名リスト」を開きます。POUのアクションプールにある使用可能なアクションと、POUヘッダに登録済みでステップに割り付けできるブール型変数が表示されます。その中の1つを選択すると「アクション連結リスト」に登録されます。

34. 任意のプログラム言語をクリックします。

35.「OK」ボタンをクリックします。

アクションボディの編集ウィンドウが開きます。

選択したプログラム言語でアクションを編集します。編集終了後にアクションウィンドウを保存して、閉じます。

プロジェクトナビゲータにあるPOUのアクションプールに、新しいアクションが表示されます。

・プログラムをアクションとして使用する場合、そのPOUボディにラベルを使用できません。

36. SFC編集ウィンドウへ戻ります。

ステップが指定色に変化します。

色の変更は、「拡張機能」メニュー→「オプション」→「プログラムオプション」→「エディタ」→「SFCエディタ」→「フォーマット」をクリックします。

■トランジション

37. initializing ステップの直後のトランジションをクリックします。

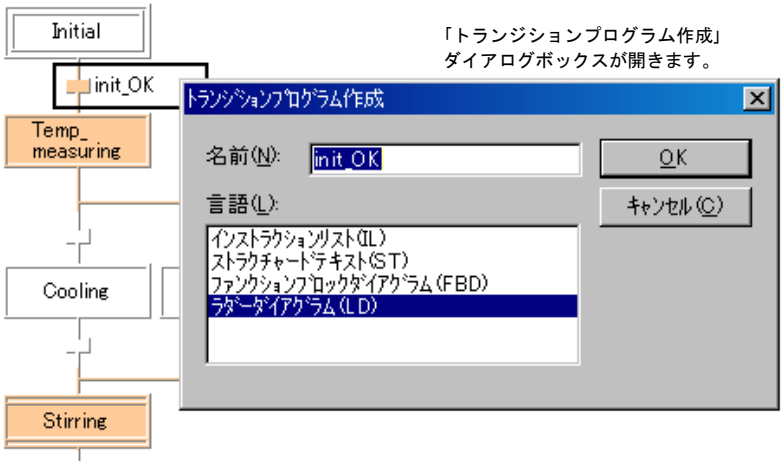
38. 選択された矩形の内部をクリックします。

SFCプログラムの最初のトランジションには、デフォルト値のTRUEが選択されています。

トランジションにBOOL型変数を割り付けるには、<F2>キーを押して「変数の選択」ダイアログボックスを開き、POUヘッダで宣言済みの変数を選択するか、新しく変数を宣言して割り付けます。

以下の例ではトランジションにプログラムを割り付けます。

39. “init_OK”と文字を入力して、 をクリックします。



40. 任意のプログラム言語を選択します。

41. OKボタンをクリックします。

トランジション名がトランジションの横に表示され、編集ウィンドウが開きトランジションボディが表示されます。選択したプログラム言語を使って、トランジションに割り付けるプログラムを作成できます。

トランジションの処理結果は、トランジション名(<F2>キーで割り付け可能)に書き込まれる必要があります。トランジション名に真理値TRUEが記述されると常に遷移条件が成立します。

42.「オブジェクト」メニュー→上書き保存をクリックします。

トランジションにプログラムを割り付ける場合、以下の制限があります。

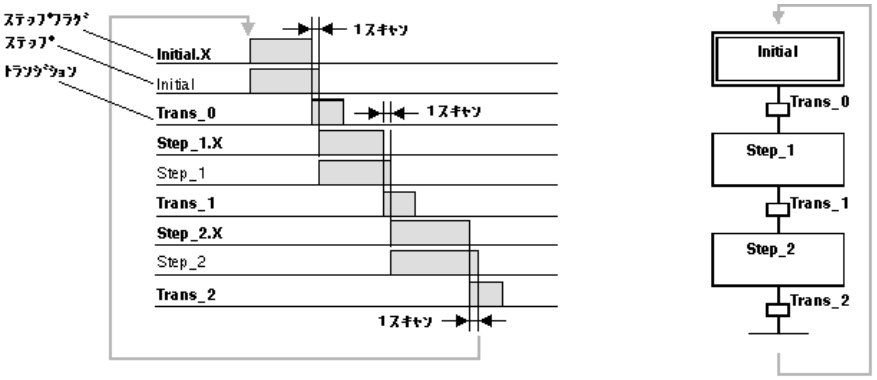
・トランジションボディで作成できるプログラムは1ブロックのみです。

・「EN/ENO」付きファンクションは使用できません。

11.4 ステップフラグ設定の時間変化

コンパイラは、SFCの各ステップごとに、BOOL型のステップフラグを自動的に割り付けます。ステップフラグは、ステップの状態が稼動中（TRUE），休止中（FALSE）のどちらかを示します。ステップの名前は、step_1.X，step_2.Xなどのように、ステップ名と添え字“X”で表記されます。

ステップ(Step_1)が起動すると､対応するステップフラグ(Step_1.X)がTRUEにセットされます。ステップ(Step_1)直後のトランジションの条件(Trans_1)が成立すると、直ちにステップフラグ(Step_1.X)はFALSEにセットされますが、現在のステップ(Setp_1)は、あと１スキャン動作してから、次のステップ(Setp_2)に移行します。
したがって、各ステップは常に最低でも2回以上は実行されます。



次の節では、ステップフラグによるフラグの切り替え動作について例を用いて説明します。

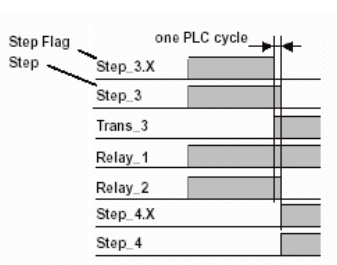
＜例＞

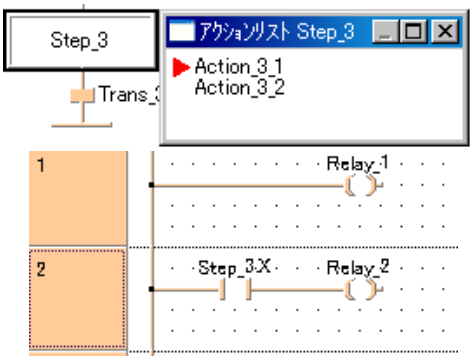
アクションは、ブール型変数やIL，FBD，LD，STの各プログラム言語で作成したプログラムで構成されます。

BOOL型変数を使って、アクション修飾ができます。

次のステップへ処理が移行する時に、ボディで使われているブール型変数の現在の状態（TRUE）または（FALSE）は保持されます。

次のステップへの移行時に、BOOL型変数にFALSEをセットするには、ステップフラグを使用して以下のように行ないます。





ステップフラグが、次のステップを実行する直前にRelay_2をリセットします。

トランジションが、次のステップに処理をジャンプさせた直後にRelay_1にTRUEが保持されます。

Step_3は、Action_3_1とAction_3_2で構成されています。

Step_3が実行中は、Relay_1とRelay_2の出力がセットされています。

Trans_3が成立時には、これらのリレーはTRUEの状態のままです。

Step_4が起動時には、Relay_1はTRUEのままですが、Relay_2はステップフラグStep_3.Xによりリセットされています。

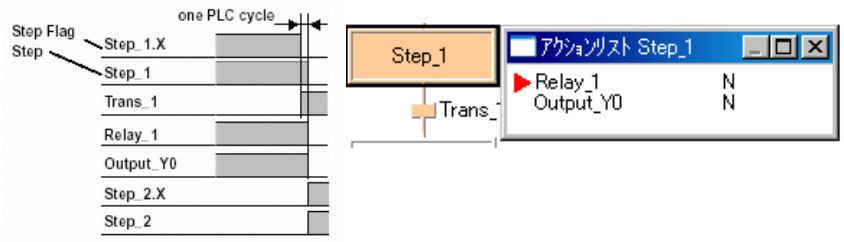
11.5 アクションの動作記号

アクションにBOOL型変数を宣言する際に、アクション連結リストで次の動作記号を使用して、BOOL型変数の動作を設定できます。

動作記号	内容
N	非保持
R	リセット(OFF保持)
S	セット（ON保持）
P	パルス

11.5.1 動作記号N

BOOL型変数は、動作記号N（非保持アクション）が割り付けられると、次のステップに移行する際にFALSEにリセットされます。すなわち、Nと記述されたブール型変数の現在値は保持されません。



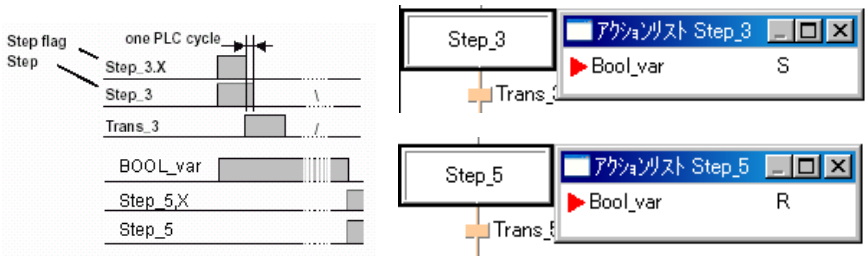
Step_1は、Relay_1とOutput_Y0のBOOL型変数がアクションに割り付けられており、2つの変数ともNが修飾されています。
Step_1が実行されている間、TRUEにセットされたRelay_1とOutput_Y0の2つの変数は、次のステップの実行直前にFALSEにセットされます。

◆ご 注 意 !

・アクション名に続いてTabキーを入力した後、[F2]キーを押すと「アクション動作リスト」が表示され、上記の動作記号が選択できます。
・特にアクション動作記号を記述をしなかった場合、PLCにより自動的にNとみなされます。

11.5.2 動作記号RとS

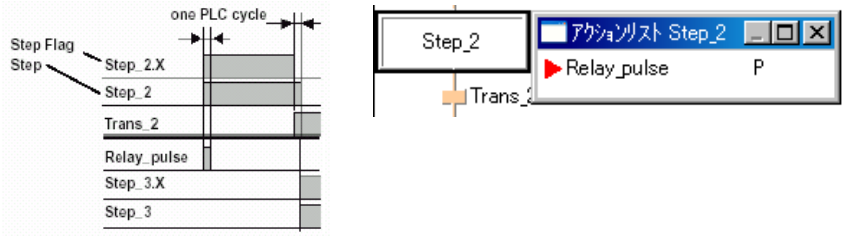
BOOL型変数に動作記号R(リセット)が割り付けられると、ステップが起動するとすぐにFALSEにセットされます。動作記号S(セット)が割り付けられると、ステップが起動するとすぐにTRUEにセットされます。
次のステップへの移行後も、変数値は変更されず、そのまま保持されます。



Step_3は、BOOL型変数Bool_barにSがアクション動作記号として割り付けられています。
Step_3が起動すると同時に、変数Bool_barがTRUEにセットされます。
この状態は、Step_5内にてリセットされるまで保持されつづけます。

11.5.3 動作記号P

BOOL型変数に、動作記号P（パルス）が割り付けられると、ステップが起動すると同時にPLCの1スキャンの間だけTRUEにセットされます。



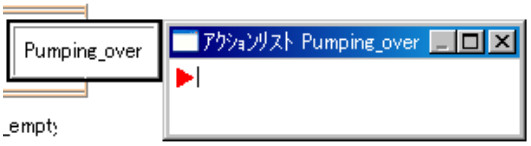
Step_2は、ブール変数Relay_pulseにPがアクション動作記号として割り付けられます。
Step_2が起動すると同時に、変数Relay_pulseがTRUEにセットされ、1スキャンサイクル後FALSEにセットされます。
以降、Step_2の実行中は、変数Replay_pulseはFALSEのままです。

11.5.4　アクションの動作

サンプルプログラムのステップpumping_over P.11-10を使って、ブール型変数へのアクション動作記号の入力手順を簡単に説明します。

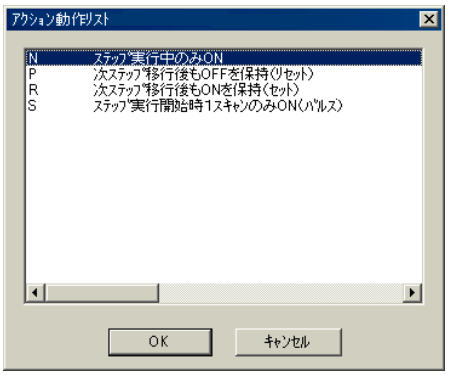
■操作手順

1. **ステップpump_overをクリックします。**
2. **または、「ツール」メニュー→「オブジェクトを開く」をクリックします。**



「アクションリスト」ダイアログボックスが開きます。

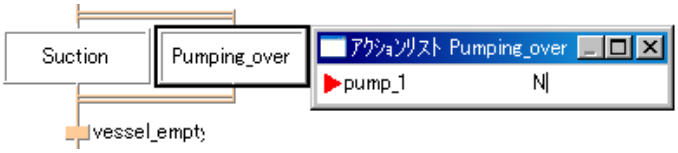
3. **<F2>キーを押して「アクション名リスト」ダイアログボックスを開き、ブール型変数pump_1を選択します。**
4. **<F2>キーを押します。**



アクションの動作記号が表示されたリストが表示されます。

5. **「動作記号」Nを選択します。**
6. **「OK」をクリックします。**

ステップpumping_overが稼動中は、変数pump_1の値はTRUE を保持します。
トランジションvessel_emptyが起動すると、変数pump_1の値は1スキャン後FALSEにセットされます。



11.6 マクロ

SFCプログラムの複数のステップ、トランジションをまとめて1つのマクロにする編集方法を、3種類に分けて簡単に説明します。

11.6.1 マクロの編集

シーケンシャルファンクションチャートのサンプルプログラムでマクロの編集について説明しました。本節では、それ以外のマクロの編集方法について説明します。

■操作手順：マクロ化

1. マクロの先頭となるステップをクリックして選択します。

2. <Shift>キーを押しながら、マクロの最後となるステップをクリックして選択します。

3.「編集」メニュー→「マクロ生成」をクリックします。

すると、選択されたすべてのステップとトランジションは、まとめて1つのマクロになります。

必要に応じて、再びマクロを開くことができます。

■操作手順：マクロを開く

1. マクロをクリックして選択します。

2.「編集」メニュー→「マクロを開く」をクリックします。

■操作手順：マクロ終了ステップの作成

1. マクロの最後のステップをクリックして選択します。

2.「編集」メニュー→「修正」→「マクロ終了ステップ」をクリックします。

マクロ作成時に、自動的に定義されるマクロ終了ステップを誤って削除しても、マクロ終了用としてこのステップを再定義できます。

◆ご 注 意 !

マクロの範囲を指定する際、マクロの最初と最後になるSFC記号は、ステップでなければなりません。

ステップ以外だと、「編集」メニュー→「マクロ生成」をクリックしてもマクロ化されません。

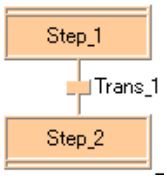
11.6.2 マクロの表示

「ツール」メニュー→「マクロ構成リスト」をクリックすると、ダイアログボックスが開いて選択されているPOUの定義されたすべてのマクロが一覧表示されます。
一覧から編集するマクロを選択し開きます。

■操作手順

1. **POUボディにある任意のマクロステップをクリックします。**
2. **「ツール」メニュー→「マクロ構成リスト」、または  をクリックします。**
   マクロ構成リストが開いて、選択されたPOUの中で定義されたマクロのすべてが表示されます。

3. **任意のマクロ名をクリックします。**
4. **OKボタンをクリックします。**
   選択されたマクロの内容が編集ウィンドウに表示されます。
   表示されているものにステップ，トランジション，分岐を挿入することができます。



5. **任意のステップをクリックします。**
6. **「ツール」メニュー→「オブジェクトを閉じる」または  をクリックします。**
   元のSFC編集ウィンドウへ戻ります。

11.7 SFCプログラムのチェック

「オブジェクト」メニュー→「チェック」、または  をクリックすると、SFCプログラムを随時チェックできます。

これまで登録されたプログラム全体を、文法エラーや宣言エラーがないかチェックします。
(例：宣言されていない変数がないか、同じステップ名が使われてないか等)
エラーごとに、独立してエラー／警告リストに表示されます。
エラー／警告リストからエラーを選択して表示ボタンをクリックすると、POUボディにあるエラーの個所が設定色に変わります。

同一のエラーが複数発生した場合、「ツール」メニュー→「エラーの検索」、または  をクリックすると、その時表示されているエラーから次のエラー発生個所までジャンプして見ることができます。

◆ご 注 意 !

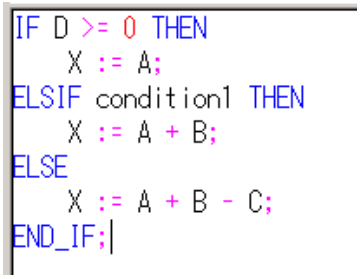
まず最初に、表示されたエラーを修正して、「オブジェクト」メニュー→「チェック」をクリックしてチェックを繰り返してください。
他のすべてのエラーは、最初に発生したエラーに関連して発生している可能性があります。

12章

ストラクチャードテキスト(ST)でプログラムを作成する

12.1 概要

ストラクチャードテキスト（ST）はラダーなどのグラフィック形式の言語と異なり、テキスト形式でプログラムを作成する言語です。
他の言語（LD,IL,SFC,FBD）と同様に、弊社のPLC全機種に対応しています。



またSTエディタでは、Windowsで使用されるテキストエディタの標準的な編集機能（切り取り，コピー，貼り付け，検索，置き換え等）が使用できます。
さらに命令入力補助機能（テンプレート）により、ファンクションやファンクションブロック，IF文やFOR文などの入力が格段に簡単になりました。

ただし、プログラミングの記述方法に制限がありますので、後述の ［ＳＴ作成における注意事項]をお読み頂き、充分ご理解くださいますようお願い致します。

12.2 STエディタの準備

STエディタを起動するには、以下の2通りの方法があります。

・FPWIN Proを起動しST言語のPOUを作成する。

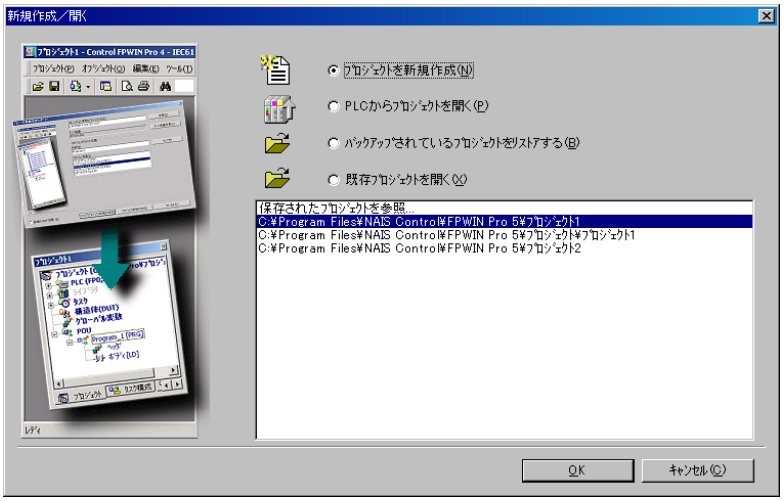
・編集中のプロジェクトにST言語のPOUを追加する。

12.2.1 FPWIN Proを起動しST言語のPOUを作成する。

【手順】

1．デスクトップ上にFPWIN Proのアイコンがある場合はそれをダブルクリックするか、スタートメニューからFPWIN Proを起動します。

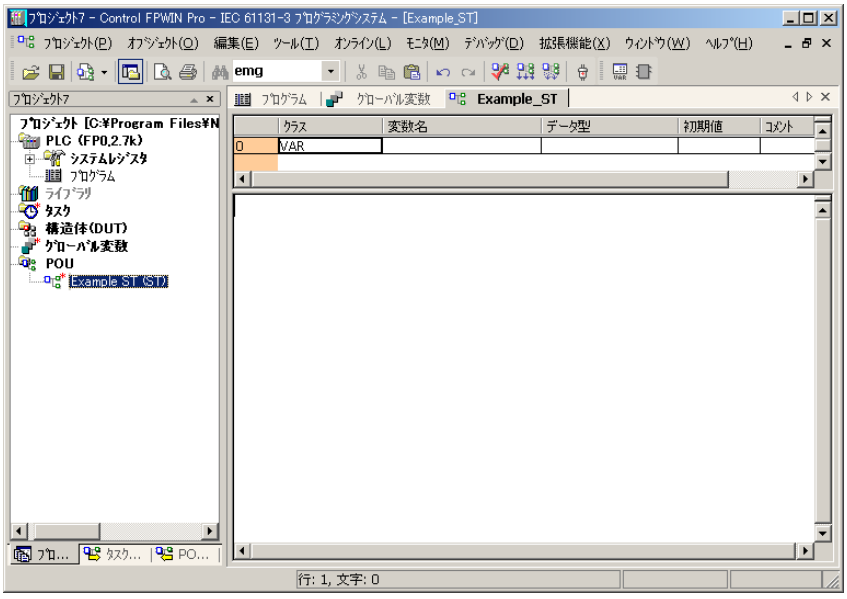
2．以下の画面が表示されますので、「プロジェクトを新規作成」を選んでOKをクリックしてください。



３．以下の画面が表示されますので、プロジェクト名とプログラム名を入力しプログラム言語に「ストラクチャードテキスト（ST)」を選択してください。

設定後、「POU付プロジェクト新規作成」ボタンをクリックしてください。

４．以下のようにST編集用のエディタが表示されます。



12.2.2　編集中のプロジェクトにST言語のPOUを追加する。

FPWIN Proで編集中のプロジェクトに新しくST言語のPOUを追加するには、以下の手順をおこないます。

[手順]

この例では既にPOUが3つ登録されています。

1．ナビゲータ上の「POU」を右クリックして、コンテキストメニューから「POUの新規作成」を選択するかもしくは、「編集」メニューから「新規作成」→「POU」を選択します。

2．以下の画面が表示されますので「ファイル名」を任意に設定（この例ではExample_ST）し、

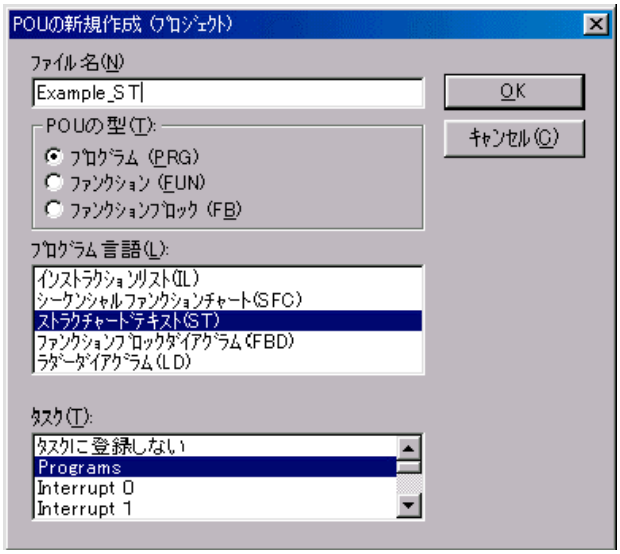
・POUの型：　　プログラム（PRG）

・プログラム言語：　ストラクチャードテキスト（ST）

・タスク：　　Programs

となっていることを確認して「OK」をクリックしてください。

ST編集用のエディタが表示されます。

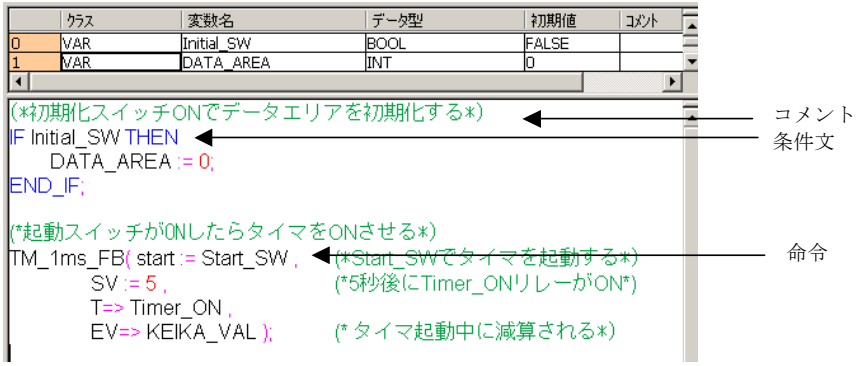


12.3 プログラムを作成する

12.3.1 STの記述方法

STエディタでは、基本的にキーボードによるテキスト入力でプログラムを作成します。

＜例＞




```
IF Initial_SW THEN
    DATA_AREA := 0;
END_IF;
```

IF や FOR などの命令文は、次の文との間に
1 つ以上のスペースが必要です。
演算子の場合は特に必要ありません。
(DATA_AREA:=0;でもかまいません)

```
IF Initial_SW THEN DATA_AREA := 0;
END_IF;
```

左記の条件文は、このように記述してもかまいません。
命令の最後には必ずセミコロン「；」を付けてください。

・詳細な編集方法、知っておくと得する便利な編集方法は、
「12-3-2.命令を入力する」を参照してください。

・STエディタで使用できる命令／演算子／オペランドを知りたい場合は、
「12-3-3.STで使用できるオペランド／命令文／演算子」を参照してください。

・STエディタでの変数の登録方法、登録した変数の使用方法を知りたい場合は、
「12-3-4.変数を使用する」を参照してください。

・STエディタでのコメントの入力方法を知りたい場合は
「12-3-5.コメントを入力する」を参照してください。

12.3.2 命令を入力する

ここでは実際に、STで各種命令を入力する方法を示します。

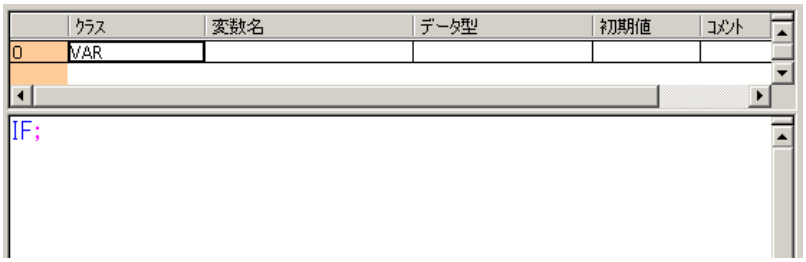
12.3.2.1 命令文を入力する

STでサポートされる各種命令文（「12-3-3.STで使用できるオペランド／命令文／演算子」を参照してください）は基本的にキーボードで直接入力しますが、”挿入用テンプレート（ショートカット)”を使用すると編集が容易になります。

＜例＞１．"IF"命令文を入力する。

■操作手順

１．ST編集画面で、最初にキーボードで"IF"を入力します。

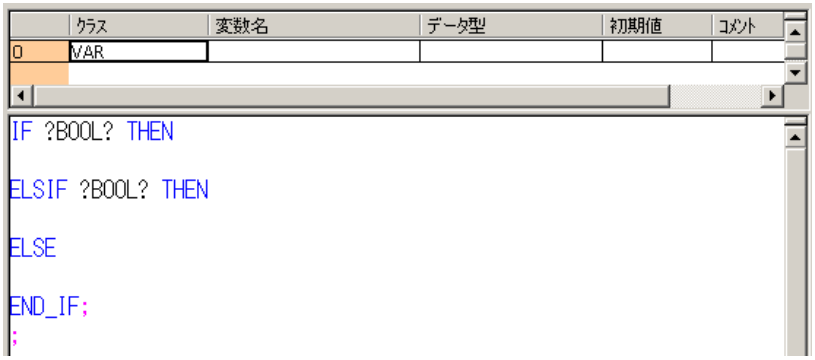


２．"IF"命令のどこかにカーソルがセットされた状態で、<Ctrl> + <F1>キーを押します。

（挿入用テンプレート）

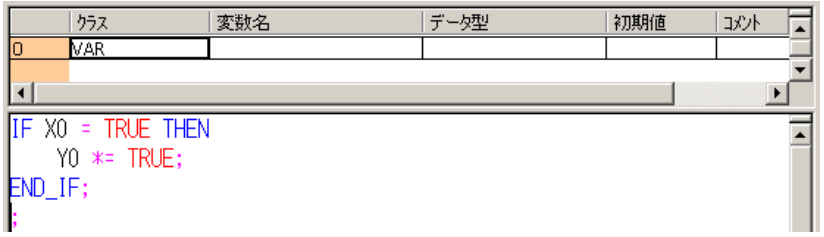
すると、以下のようなIF文が表示されます。

"?"で囲まれた部分に任意の変数やデバイスを入力します。



３．必要に応じて"IF"～"END_IF"までの間を編集してください。

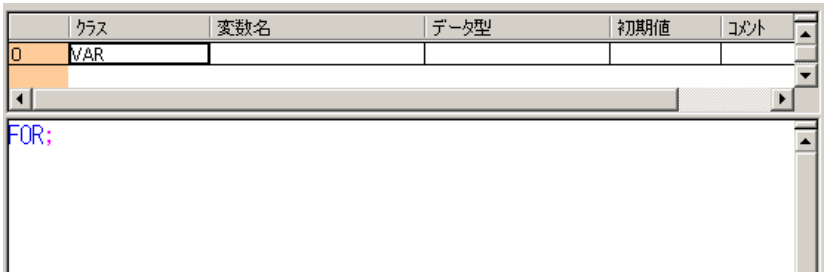
【例】 X0がONした場合はY0をONさせる。



＜例＞２．"FOR"命令文を入力する。

■操作手順

１．ST編集画面で、最初にキーボードで"FOR"を入力する。

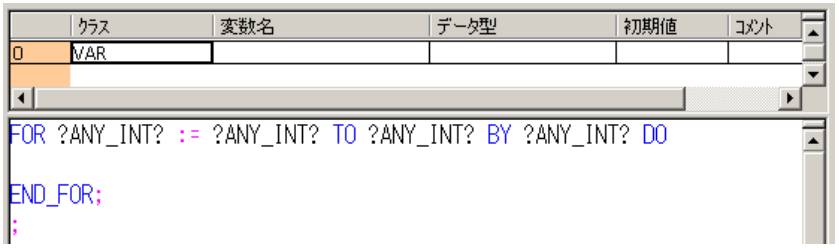


２．"FOR"命令のどこかにカーソルがセットされた状態で、<Ctrl> + <F1>キーを押します。

（挿入用テンプレート）

すると、以下のようなFOR文が表示されます。

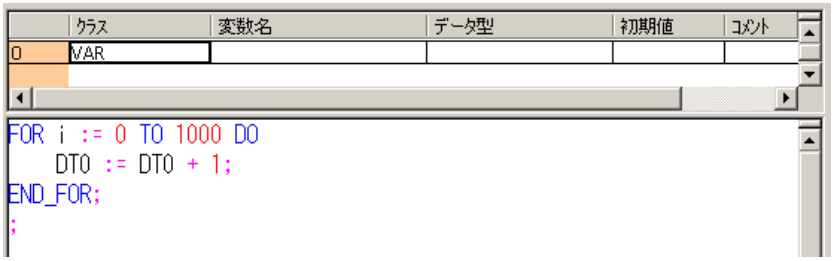
"?"で囲まれた部分に任意の変数やデバイスを入力します。



３．必要に応じて"FOR"～"END_FOR"までの間を編集してください。

【例】 iを0～1000まで1ずつ増加させて、DT0の値をiずつ増加させる。

（この例では、INT型の変数としてiを使用しています）



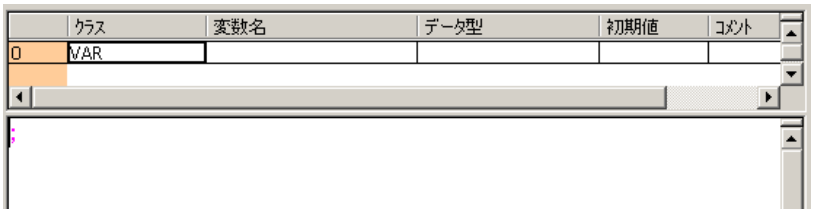
12.3.2.2 OP/FUN/FBを入力する

ST編集画面でのOP（演算子）/FUN（ファンクション）/FB（ファンクションブロック）の入力方法を以下に示します。

＜例＞データ転送命令 F0(MV)命令を入力する。
（この例では[F0(MV) DT0, DT1]を入力します）

■操作手順

1．ST編集画面の命令を入力する位置にカーソルをセットします。

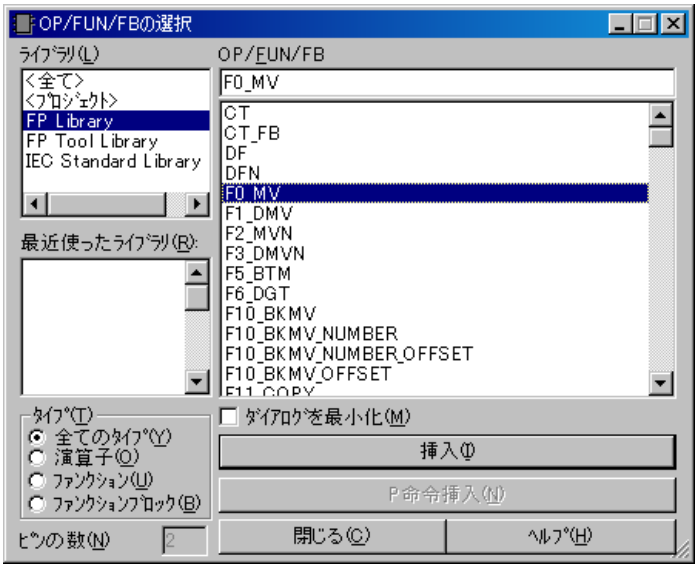


2．ツールバーの  ボタンをクリックするか、「ツール」メニューの「命令の選択」を選択します。

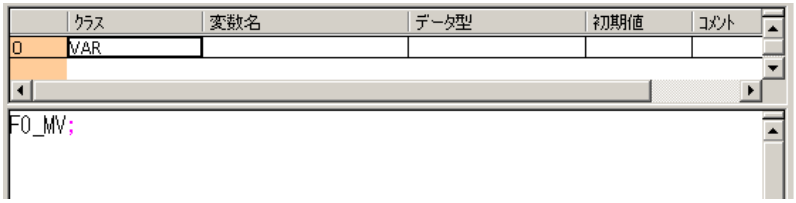
以下のダイアログが表示されます。

３．「ライブラリ」の選択エリアで「FP Library」の中から「F0_MV」を選択し、ダブルクリックするか「ボディに挿入」ボタンをクリックしてください。

（ライブラリ選択エリアではデフォルトで<全て>が選択されていますが、このままでも"F0_MV"命令はリストに存在します）



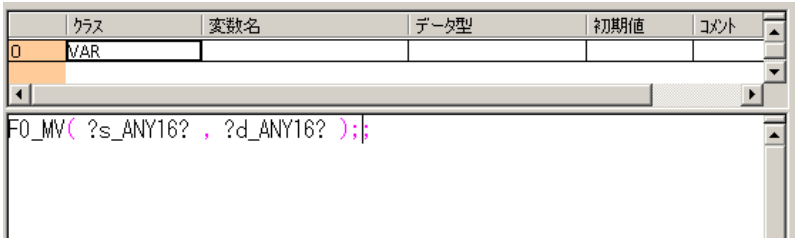
４．以下のように、F0_MV命令が挿入されます。



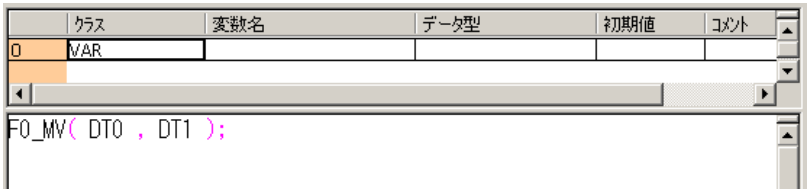
５．"F0_MV"命令のいずれかの位置にカーソルがセットされた状態で、<Ctrl> + <F1>キーを押します。

（挿入用テンプレート）

すると以下のように、入力が必要なオペランドが自動的に表示されます。



６．'?'で囲まれた文字（入力するデータの型を示しています）を入力して完成です。



◆ここがポイント!

・"F0_MV(DT0, DT1)"は、全てキーボードで直接入力しても構いません。
命令の種類や型を覚えている場合は、命令選択用のダイアログを開いたり<Ctrl> + <F1>キーでオペランドを表示させる必要はありません。
または"F0_MV"までキーボードで入力して、<Ctrl> + <F1>キーでオペランドを表示させることもできます。

12.3.3 STで使用できるオペランド／命令文／演算子

■STで使用できるオペランドとデータ型を以下に示します。
これらのオペランドは、演算子，命令と共に使用されます。

名称	データ型	例
定数値	値 文字列 時間	55, -3.14159 'This is a sample text.' T#8d_3h_23m
変数	変数 配列の要素 DUT（構造体）の要素 構造体内の配列要素	Var1 Array1[5] Dut1.Var1 Dut1.Array1[i+5]
ファンクション	ファンクションコール	Fun1(a,b,c)

STでPLCデバイスを記述する（グローバル変数を定義する，プログラム中で直接使用する）場合、以下のように“全て大文字で記述”してください。
(小文字で記述するとコンパイル時に変数と見なされ、変数宣言がされてないというエラーになります)

	名称	例（全て大文字で記述）	補足
リレー	外部入力　X	X0	
	外部出力　Y	Y0	
	内部リレー R	R0	
	リンクリレー　L	L0	
	タイマ／カウンタ　T/C	T0/C200	
メモリエリア	外部入力　WX	WX0	ダブルワード指定時はDWX0
	外部出力　WY	WY0	ダブルワード指定時はDWY0
	内部リレー WR	WR0	ダブルワード指定時はDWR0
	リンクリレー　WL	WL0	ダブルワード指定時はDWL0
	データレジスタ　DT	DT0	ダブルワード指定時はDDT0
	リンクレジスタ　LD	LD0	ダブルワード指定時はDLD0
	タイマ／カウンタ　SV 設定値エリア	SV0	ダブルワード指定時はDSV0
	タイマ／カウンタ　EV 経過値エリア	EV0	ダブルワード指定時はDEV0
	インデックスレジスタ　I	IX, IY	I0 – IDは使用不可
定数	10進定数	100, 1000, 255, -15	数字を直接記述
	16進定数	16#0000, 16#50AC, 16#FFFF	先頭に「16#」を付加する
	10進定数（浮動小数点型）	0.0, 100.0, 12.3, -0.5	小数点を付けると実数扱いになる。

データ型	内容	ビット数	範囲
BOOL	0:FALSE, 1:TRUE	1	-
DINT	倍精度整数	32	-2147483648 ～ 2147483647
DWORD	32ビット文字列	32	-
INT	整数	16	-32769 ～ 32767
REAL	実数 （IEEE754基準）	32	-3.402823*E38 ～ -1.175494*E-38, 0.0, +1.175494*E-38 ～ +3.402823*E38
STRING	文字列	-	最大255文字
TIME	接続時間	16（FP1,3など） 32（FP0, Σ ,2/2SH,10SH）	0.01 ～ 327.67秒 0.01 ～ 21474836.47秒
WORD	16ビット文字列	16	-

■STで使用できる命令一覧を以下に示します。

命令	例	説明
:= （代入）	y := (a + b + c)/3;	右辺の値が左辺の変数に代入されます。 この例では、a,b,cの平均値がyに代入されます。
（FUN呼び出し）	Y := SIN(x);	単純に1つの引数を与えるファンクションを呼び出しています。 この例では入力変数xの正弦を計算して変数Yに代入します。
（FB呼び出し）	Y := LIMIT(MN:=0,IN:=X,MX:=100);	引数に代入文を含むファンクションを呼び出しています。 この例では、引数INがMNとMXの間の値であればそのままYに代入されます。
	Ton1(IN:=Start1, PT:=T#300ms, End1:=Ton1.Q; Ev1:=Ton1.ET);	入力パラメータを直接代入し、FBを呼び出しています。 この例では、変数Start1がONすると300ms後に変数End1がONします。 変数Ev1には経過値が代入されます。

IF
（条件分岐）
	IF a>=0 THEN
	b:=0;
	ELSIF a>=100 THEN
	b:=1;
	ELSE
	b:=2;
	END_IF;
		記述された真理値にしたがって、分岐された代入文が実行されます。
		IF文を使用する場合は、END_IFを入力する必要があります。

CASE （複数選択）	CASE a OF 0: b=0; 1,2: b=1; 3,4,10..20: b=2; 100..110: b=3; ELSE b=4; END_CASE;	記述された変数値にしたがって、実行される文が選択されます。 変数aは、INT型（またはDINT型）でなければなりません。
FOR （繰り返し）	FOR i:=0 TO 100 DO SUM:=SUM+a[i]; END_FOR;	設定された増分で繰り返しを実行します。 この例ではiを1ずつ増加させて、100になったら処理を抜けます。
	FOR i:=0 TO 100 BY 10 DO SUM:=SUM+a[i]; END_FOR;	この例ではiを10ずつ増加させて、100になったら処理を抜けます。

WHILE
（繰り返し）
	i:=0;
	WHILE i<=100 AND a[i]<100 DO
	i:=i+10;
	END_WHILE;
		WHILE文以下の条件が一致している間、処理を繰り返します。
		条件は処理を実行する前にチェックされます。

REPEAT
（繰り返し）
	i:=0;
	REPEAT
	i:=i+10;
	UNTIL i>100 OR a[i]>=100
	END_REPEAT;
		UNTIL文以下の条件を満たすようになるまで、処理を繰り返します。
		条件は処理を実行した後にチェックされます。

EXIT （中止）	EXIT;	無条件に繰り返し処理を中止します。
RETURN （ジャンプ元へ戻る）	RETURN;	呼び出し元のPOUへプログラムを戻します。

これらの命令は、全てのPLCで使用することができます。

■STで使用できる演算子を以下に示します。

演算子	扱えるデータ型	例	結果	演算順位
()		(1+2)*(3+4)	21	最上位
** （べき乗）	REAL	3.0 ** 2.0	9.000435	
- （符号反転） NOT （論理否定）	INT, DINT, REAL BOOL, WORD, DWORD	-DATA0 （DATA0は100とする） NOT DATA1 （DATA1は16#0064とする）	-100 16#FF9B	
* （乗算） / （除算） MOD （剰余）	INT, DINT, REAL	10 * 5 20 / 4 12 MOD 10	50 5 2	
+ （加算） - （減算）	INT, DINT, REAL	10 + 20 5 – 1 2 - 10	30 4 -8	
> < >= <= （比較）	ANY (ANY_BITは不可)	1 > 3 1 < 3 5 >= 5 2 <= 1	FALSE TRUE TRUE FALSE	
= （等式） <> （不等式）	ANY	X0 = X1 （X0:ON状態，X1:OFF状態） 100 <> 100	FALSE FALSE	
&,AND （論理積）	BOOL, WORD, DWORD	16#A5F8 AND 16#B0B0	16#A0B0	
XOR （排他的論理和）	BOOL, WORD, DWORD	16#A5F8 XOR 16#B0B0	16#1548	
OR （論理和）	BOOL, WORD, DWORD	16#5555 OR 16#AAAA	16#FFFF	最下位

これらの演算子は、全てのPLCで使用することができます。
(ただしREALを扱えるのは、FP0, FP Σ, FP10SH, FP2/2SHのみです)

【補足】

なお、三角関数（SIN, COS, TAN など）や指数，対数，平方根などの算術命令はIEC命令として定義されており、FP-X, FP0, FP Σ, FP10SH, FP2/2SH（実数を扱えるPLC）で使用することができます。

演算子	扱えるデータ型	例	結果
ABS （絶対値）	ANY_NUM	Result := ABS(-5.0) （以下、ResultはREAL型の変数）	5.0
SQRT （平方根）	REAL	Result := SQRT(4.0)	2.0
LN （自然対数）	REAL	Result := LN(100.0)	4.605168
LOG （対数）	REAL	Result := LOG(100.0)	1.999999
EXP （自然指数）	REAL	Result := EXP(3.0)	20.08554
SIN （正弦）	REAL （角度データはラジアンで指定）	Result := SIN(1.0)	0.841471
COS （余弦）	REAL （角度データはラジアンで指定）	Result := COS(1.0)	0.5403023
TAN （正接）	REAL （角度データはラジアンで指定）	Result := TAN(1.0)	1.557408
ASIN （逆正弦）	REAL （角度データはラジアンで指定）	Result := ASIN(1.0)	1.570796
ACOS （逆余弦）	REAL （角度データはラジアンで指定）	Result := ACOS(1.0)	0.0
ATAN （逆正接）	REAL （角度データはラジアンで指定）	Result := ATAN(1.0)	0.7853

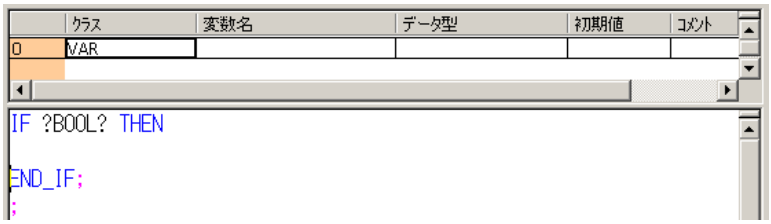
なお、これらと同等の命令がFPWIN Proシリーズ命令（FP Library）にも定義されています。どちらを使用していただいても問題ありません。

12.3.4 変数を使用する

LDなどの他の編集画面と同様に、STエディタでも変数を使用したプログラミングが可能です。あらかじめ登録された変数を使用することもできますし、プログラミング中に新しく変数を登録していくこともできます。

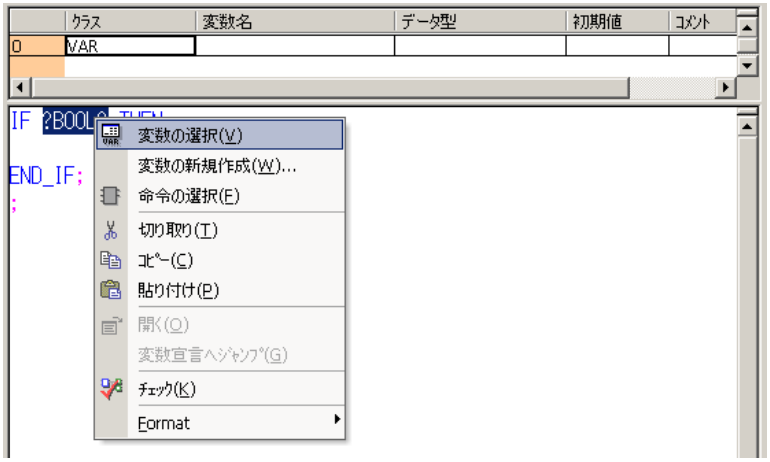
<例>１．登録された変数を使用する。

以下のSTプログラムで、"?BOOL?"の部分に変数を入力します。



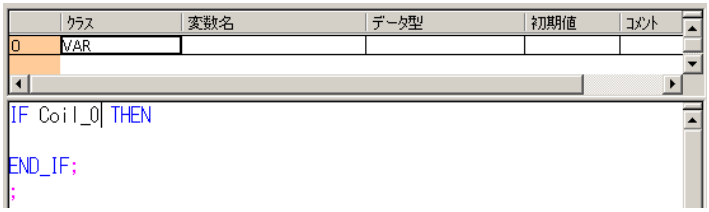
■操作手順

１．ツールバーの  をクリックするか<F2>キーを押す、または右クリックで「変数の選択」を選択します。



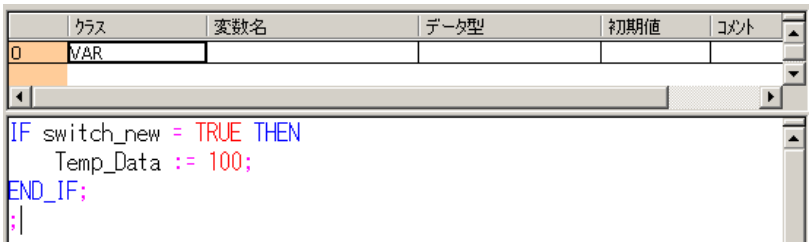
２．以下のダイアログが表示されますので、任意の変数を選択しダブルクリックするか「ボディに挿入」ボタンをクリックします。

３．ST編集画面に変数が入力されます。



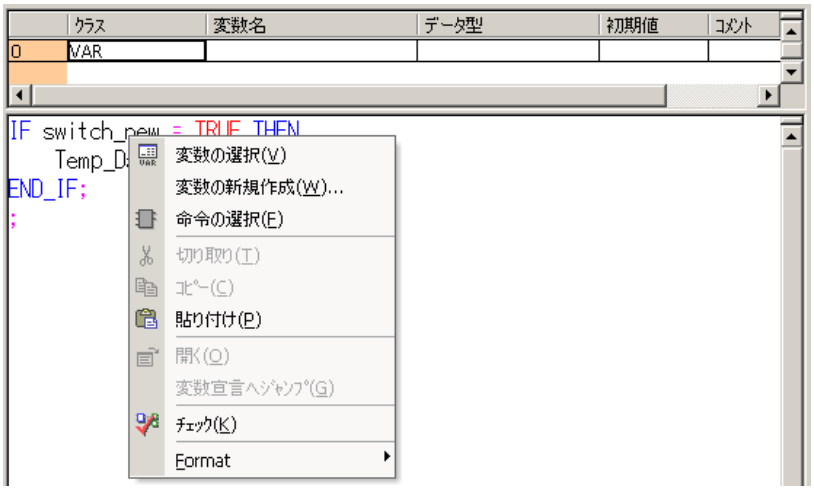
<例>２．プログラミング中に変数を新規登録する。

以下のSTプログラムで、新たに編集した変数"switch_new"をヘッダに登録します。
(新規の変数を登録していない場合は、プログラムチェック時にエラーが表示されます)



■操作手順

１．「ツール」メニューから「変数の新規作成」を選択するか、または右クリックで「変数の選択」を選択します。



２．以下の画面が表示されますので、「変数名」「データ型」「初期値」を入力してください。

３．この例では、変数名は"switch_new"，データ型は"BOOL"，初期値は"FALSE"に設定します。

・データ型は、入力エリア横の … ボタンをクリックするとデータ型選択用のダイアログが表示されます。

入力エリアに直接"BOOL"と入力しても構いません。

４．以下のように入力が完了したら、「宣言」ボタンをクリックしてください。

"switch_new"が新しく変数（この例ではローカル変数）として登録されます。

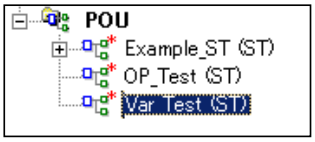
<例>３．ヘッダに変数を登録する。

あらかじめPOUヘッダに変数を登録しておくことにより、プログラミング作業がスムーズにおこなえます。

以下に、POUヘッダに変数を登録する手順を示します。

■操作手順

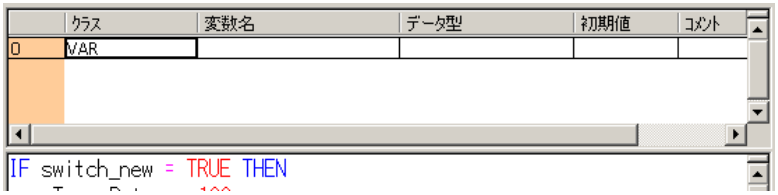
１．プロジェクトナビゲータのPOUの１つをダブルクリックします。



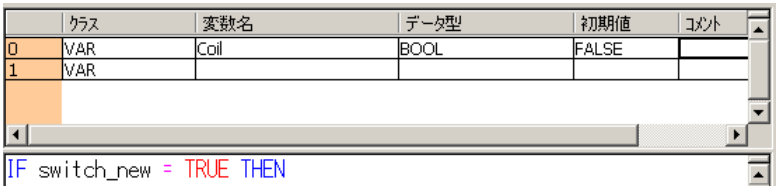
２．以下の画面が表示されますので、ここで変数を登録します。

変数名，データ型，初期値を入力してください。（コメントは任意です）

本説明では、例として変数名：Coil，データ型：BOOL，初期値：FALSEで設定します。



３．入力後に、ツールバーの  ボタンをクリックするか「オブジェクト」メニューの「チェック」を選択して登録に問題がないかチェックしてください。

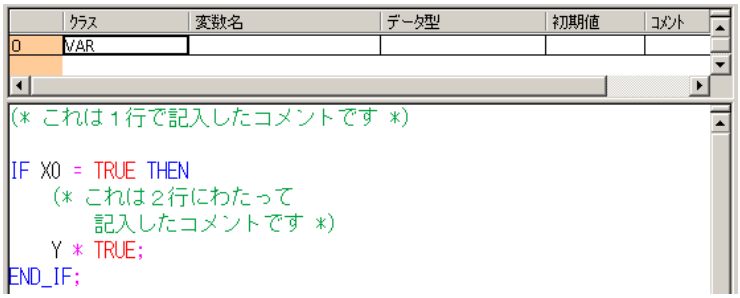


なお変数についての詳細は、第9章「変数を使ってプログラムを作成する」を参照してください。

12.3.5　コメントを入力する

ST編集画面の任意の行に、コメントを入力することができます。
コメントは、アスタリスク付きのカッコ"(*"と"*)"で囲みます。

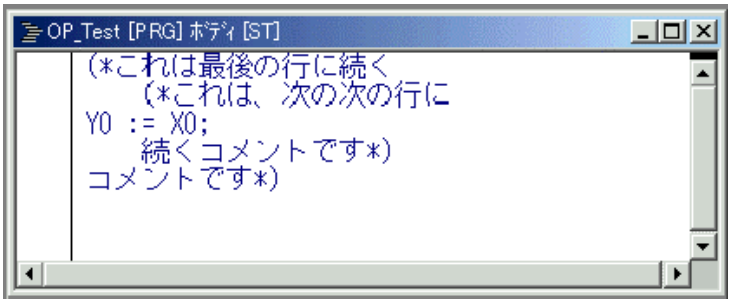
＜例＞



◆ご 注 意 !

v 5

・Ver.5では下記のようなコメントの入れ子（ネスティング）が可能ですが、Ver.5未満では、サポートしていません。



上記の例では、最初の”(*”の次にまた”(*”が検出されますので、Ver.5未満ではチェック時に文法エラーになります。”(*”　と　”*)”　は、必ずペアで使用してください。

12.3.6 ST作成における注意事項

ここでは、STでプログラムを編集するにあたっての注意事項を示します。

■コードの記述について

・IFやFORなどの命令文は、次の文との間に1つ以上のスペースが必要です。

<例>

```
IF switch_0 THEN → OK
IFswitch_0THEN   → NG
```

・1つの命令文または演算式に対して、末尾に必ず1つのセミコロン「;」が必要です。

・STの命令文または演算式は、半角の大文字で記述してください。

・X,Y,R,T,C,L,DT,WR,LD,FLなどのPLCデバイスは、半角の大文字で記述してください。
（小文字だと変数と見なされ、これをPOUヘッダで宣言しないとチェック時にエラーになります）

<例>

x0→変数
X0→PLCの外部入力X0

・PLCデバイスをダブルワードで表現する場合、先頭に「D」を付加してください。

<例>

DT0のダブルワード→`DDT0`
WR0のダブルワード→`DWR0`

・BOOL型変数について、TRUEとFALSEは使用可能ですが、これに数値の"0"と"1"を使用することはできません。（TRUEは省略することもできます）

<例>

```
IF switch_0 = TRUE THEN
```

は

```
IF switch_0 THEN
```

と記述してもかまいません。

・STでEN/ENO付ファンクションは使用できません。

■演算の順位について

・次の演算式に対して、変数の値は　A:=1.0; B:=2.0, C:=3.0; D:=4.0とします。

```
X := A + B - C * SQRT(D);
```

演算結果は、-3です。

括弧が入ると演算順位が変わり、たとえば次のようになります。

```
X := A + (B - C) * SQRT(D);
```

演算結果は、-1です。

■PLCの動作に影響する記述の制限

・"FOR"や"WHILE"などを使用して長時間にわたるループは記述しないでください。

PLCがRUNモードになるとSTプログラムは先頭から最後まで実行され、続けてI/Oリフレッシュや通信サービスが実行され、また先頭からプログラムが実行されます。
この動作はPLCがPROGモードになるまで繰り返されます。
よって"FOR"や"WHILE"などを使用して長時間にわたるループを発生させると、I/Oリフレッシュや通信サービスが行われなくなり、PLCのウオッチドグタイマがタイムアップする可能性があります。

<例>

（iとjはINT型の変数）

```
FOR i:=0 TO 32767 DO
   j:=j+1;
END_FOR;
```

上記プログラムはFOR文のループ時間が長くなり、PLCのウォッチドグタイマがタイムアウトを起こします。
（FP0やΣの場合はERR.のLEDが点灯、FP2SHなどはALARMのLEDが点灯します）

・WHILEなどの反復文において、外部入出力（X,Y）を制御条件に使用しないでください。

（使用してもコンパイルエラーにはなりません）
PLCはタスクの最後に到達するまでI/Oリフレッシュを行いませんので、この文は無限ループになりPLCのウオッチドグタイマがタイムアップする可能性があります。

<例>

```
WHILE X0=TRUE DO
   DT0:=DT1;
END_WHILE;
```

上記のような記述だと、X0がONすると　DO文以降が繰り返され、I/Oリフレッシュが実行されず、プログラムが最後まで実行されません。
従って実際にX0がOFFしてもプログラムには反映されないばかりか、ウォッチドグタイマがタイムアップしてPLCがエラーになります。

■PLCの動作に影響する記述の制限

・"FOR"や"WHILE"などを使用して長時間にわたるループは記述しないでください。

PLCがRUNモードになるとSTプログラムは先頭から最後まで実行され、続けてI/Oリフレッシュや通信サービスが実行され、また先頭からプログラムが実行されます。
この動作はPLCがPROGモードになるまで繰り返されます。

■変数が実際に使用しているPLCアドレスを取得したい場合

・ローカル変数を使用してプログラムを組んだ場合、実際にその変数に割り当てられるPLCアドレスは、コンパイル後にしかわかりません。しかし、命令には、PLCのアドレスを指定しなければならない命令があります。この場合、Adr_Of_～という命令を使用します。

- ・Adr_Of_Var　　：変数が使用する先頭アドレスを取得します。
- ・Adr_Of_VarOffs　：変数が使用する先頭アドレスから＋何ワードかのアドレスを取得します。
- ・AdrLast_Of_Var　：変数が使用する最終アドレスを取得します。

■文字列を使用するときの注意事項

<例>STRING 型の変数”Send_Char”を 10 文字分確保して、’ABCDEF’で初期化する。

	クラス	変数名	データ型	初期値	コメント
0	VAR	Send_Char	STRING[10]	'ABCDEF'	送信文字
1	VAR				

・通常、文字列を変数として宣言した場合、下記のように、実際には宣言した文字数＋2ワード分が確保されます。

Send_Char

10（確保された文字数）		DT** + 0
6（実際に使用される文字数）		DT** + 1
42(H) ‘B’	41(H) ‘A’	DT** + 2
44(H) ‘D’	43(H) ‘C’	DT** + 3
46(H) ‘F’	45(H) ‘E’	DT** + 4
未使用(8文字目)	未使用(7文字目)	DT** + 5
未使用(10文字目)	未使用(9文字目)	DT** + 6

上位ﾊﾞｲﾄ　　下位ﾊﾞｲﾄ

上記の場合、5ワード（10文字分）＋2ワードで合計7ワード分確保されています。

上記のように文字列の前に２ワード確保するようなデータテーブルを持つ文字列命令(CONCAT等)を使用するときは、先頭の2ワードを意識する必要はありません。
しかし、F159_MTRN（シリアル送信命令）など上記とは異なったデータテーブルをもつ応用命令で使用するときは注意が必要です。

<例>F159 命令を使って、Send_Char で定義した文字列’ABCDEF’を COM1 から送信する。

F159_MTRN(s_Start, n_Number, d_Port);

s_Start	:送信エリア先頭No.
n_Number	:送信する文字数（バイト数）
d_Port	:送信COMポートNo.

変数のアドレスを参照する場合は、Adr_Of_～という命令を使用します。

F159_MTRN（シリアル送信命令）命令での送信用データテーブルは以下の形式になっています。

送信開始時、送信バイト数		DT** + 0
2文字目	1文字目	DT** + 1
4文字目	3文字目	DT** + 2
6文字目	5文字目	DT** + 3
8文字目	7文字目	DT** + 4
10文字目	9文字目	DT** + 5

上位ﾊﾞｲﾄ　　下位ﾊﾞｲﾄ

この命令のデータテーブルは、実際の文字列の前に１ワードしかありません。従ってAdr_Of_VarOffs命令（＊注）を使って間接的にアドレスを指定することになります。正しく送信を実行させるには以下のように指定する必要があります。

F159_MTRN(s_Start:= Adr_Of_VarOffs(Var:= NAME, Offs:= 1), n_Number:= 6, d_Port:= 1);

◆ご 注 意！

Adr_Of_VarOffs命令：
変数Varの先頭からオフセット値Offsを加算したメモリエリア番号が出力されます。

■その他

・以下のようなIF文においては変数aが100以上の場合、a:=a+1は実行されません。
しかしこの場合においても常にUserFunc(a)は実行されます。

```
IF a<100 AND UserFunc(a) THEN
        a := a + 1;
END_IF;
;
```

演算を短縮させたい場合は以下のように記述してください。
この場合、aが100以上の場合はUserFunc(a)は実行されません。

```
IF a<100 THEN
    IF UserFunc(a) THEN
        a := a + 1;
    END_IF;
END_IF;
;
```

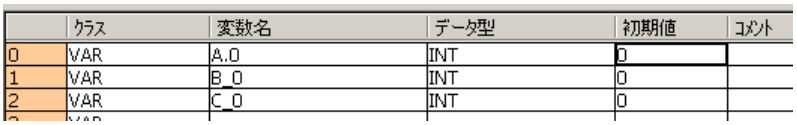
12.4 プログラムのチェック／コンパイルを実行する

12.4.1 プログラムチェックを実行する

プログラム編集中は、各オブジェクト（POU等）のチェックを随時行ってください。
命令の記述ミスや、宣言されていない変数を使用しているなどのエラーをチェックできます。

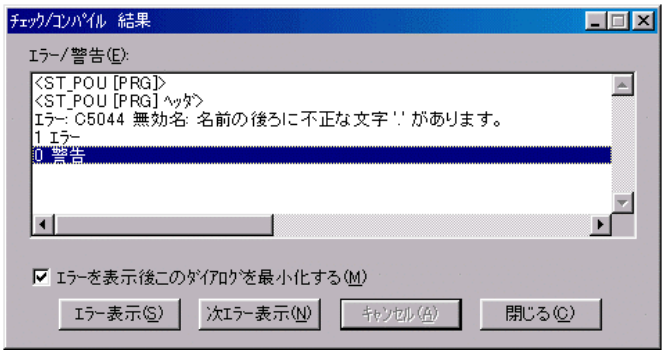
プログラムチェックは、チェックの対象となるオブジェクトをアクティブにして「オブジェクト」メニューの「チェック」を選択するか、ツールバーの  ボタンをクリックして実行します。

【例】１．チェックを実行します。



	ｸﾗｽ	変数名	データ型	初期値	ｺﾒﾝﾄ
0	VAR	A.0	INT	0	
1	VAR	B_0	INT	0	
2	VAR	C_0	INT	0	

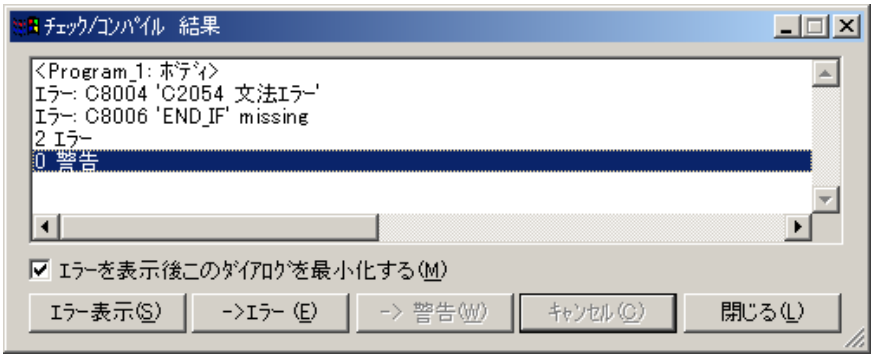
結果、以下の画面が表示されます。
これは、変数には使用できない文字"."を使っているためエラーになります。



【例】２．以下のようなプログラムをアクティブにして、チェックを実行します。

```
IF X0 THEN Y0:= TRUE;
```

結果、以下の画面が表示されます。



IF文は必ずEND_IFとペアで使用しなければなりません。
この場合は、IF文に対するEND_IF文が存在しないためエラーになります。

◆ここがポイント!

複数のエラーが表示された場合は、まず「チェック／コンパイル　結果」ダイアログの最初に表示されたエラーを修正し、再度チェックを実行してください。
多くのエラーは最初に出てきたエラーに関連して発生しています。

12.4.2 コンパイルを実行する

各オブジェクトのチェックが問題なく終了したら、コンパイルを実行してください。
コンパイルは、「プロジェクト」メニューの「全コンパイル」を選択するかツールバーの  ボタンをクリックして実行します。
コンパイルでは、各オブジェクトのチェックでは発見できないエラーも全てチェックされます。

コンパイルが正常終了すると、作成したプロジェクトをPLCへ転送できます。

12.5 LD（ラダー）とSTの対応表（例）

LDシンボル	名称	STコード
X0 Y0 --| |----------[ ]	スタート	Y0 := X0;
X0 Y0 --| / |----------[ ]	スタートノット	Y0 := NOT(X0);
X0 X1 Y0 --| |----| |----------[ ]	アンド	Y0 := X0 AND X1;
X0 Y0 --| |----------[ ] X1 | --| |--	オア	Y0 := X0 OR X1;
X0 Y0 --| |----------<A>	オルタネートアウト	Y0 := ALT(X0);
X0 X1 Y0 --| |----| |----------[ ] X2 X3 | --| |----| |--	アンドスタック	Y0 := (X0 OR X2) AND (X1 OR X3); （STではANSという命令は使えません）
X0 X1 Y0 --| |----| |----------[ ] X2 X3 --| |----| |--	オアスタック	Y0 := (X0 AND X1) OR (X2 AND X3); （STではORSという命令は使えません）
X0 Y0 --| |--(DF)----------[ ]	立ち上がり微分	Y0 := DF(X0);
X0 Y0 --| |--(DF/)----------[ ]	立ち下がり微分	Y0 := DFN(X0);
X0 Y0 --| |----------<S>	セット	IF X0 THEN Y0 := TRUE; END_IF; （STにSETという命令はありません）
X0 Y0 --| |----------<RST>	リセット	IF X0 THEN Y0 := FALSE; END_IF; （STにRSTという命令はありません）
X0 --| |----------[TMY 0, K5] T0 Y0 --| |----------[ ]	オンディレイタイマ	TM_1s_FB(Start :=X0 , SV :=5 , T => Y0, EV => DT0); TMXの場合 : TM_100ms_FB TMRの場合 : TM_10ms_FB TMLの場合 : TM_1ms_FB
X0 CT 1008 --| |----------| | X1 | 100 | --| |----------| |	カウンタ	CT(Count:= X0 , Reset:= X1 , Num:= 1008 , SV:= 100);
| = DT0, DT1 | Y0 -- | |--------[ ]	16ビットデータ比較（スタート）	Y0 := (DT0 = DT1);
X0 |= DT0, DT1| Y0 --| |- | |-------[ ]	16ビットデータ比較（アンド）	Y0 := X0 AND (DT0 = DT1);
X0 --| |----[F0 MV , DT0, DT1]	16ビットデータ転送命令	IF X0 THEN F0_MV(DT0, DT1); END_IF;
R9010 --| |----[F0 MV , DT0, DT1]	16ビットデータ転送命令（常時実行）	F0_MV(DT0, DT1); または DT1 := DT0;
R9010 --| |----[F1 DMV , DDT0, DDT2]	32ビットデータ転送命令（常時実行）	F1_DMV(DDT0, DDT2); または DDT2 := DDT0; （ダブルワードは、DDT**と記述します）

13章

モニタ機能について

13.1 概要

プログラムをコンパイルしてダウンロードした後、その動きをモニタする方法が複数あります。
モニタモードでは、変数値の変更，経過値の表示，設定（強制入出力等）ができます。
Control FPWiN Pro では以下のような様々なタイプのモニタをサポートしています。

・データモニタ
・ヘッダモニタ
・プログラムモニタ
・制御モニタ

さらに、以下に示すステータスのモニタもサポートしています。
これらの詳細に関しては、ヘルプを参照してください。

・PLCステータス
・PCリンクステータス（MEWNET-P/W）
・ネットワークの状態（MEWNET-P）
・特殊内部リレー
・特殊データレジスタ
・共有メモリ

モニタはオンラインモード以外では操作できません。

13.2 データモニタ・強制入出力

13.2.1 データモニタ

「モニタ」メニュー→「データモニタ」をクリックすると、PLCのプログラムが稼動中にその変数値を表示できます。
必要に応じて、変数値の変更や強制設定ができます。

■操作手順

1. **「オンライン」メニュー →「オンラインモード」、または** **をクリックします。**
2. **「オンライン」メニュー →「PLC動作モード変更」、または** **をクリックして、PLCをRUNモードに切り替えます。**
3. **「モニタ」メニュー →「データモニタ」、または** **をクリックします。**
   以下のダイアログボックスが開きます。

4. **「グローバル変数」、または表示させたい変数のあるPOU名をクリックします。**
5. **任意のコンポーネントをダブルクリックします。**
   選択されたコンポーネントの変数名が表示されます。
6. **任意の変数を選択します（複数選択可）。**
   次に示すものを、直接「変数」フィールドに入力できます。
   POU名
   グローバル変数：
   ・グローバル変数リストの変数名（例:Key_1）
   ・PLCアドレス（例:X0, Y2）
   ・IECアドレス（例:%IX0.0）：
   ローカル変数：
   ・POU名．変数名（場合によってはPOU名．FB名．変数名）

7. OKボタンをクリックします。

変数の値の表記形式，状態，アドレスが表示されます。

モニタ画面には以下のように表示されます。

＜例＞

Key_1　2#0　to　%IX0.0

Key_1： 変数

2#： 2進数

0： 値（FALSE）

%IX0.0： IECアドレス

「変数名」にPOU名を入力すると、POUにあるすべての変数がモニタできます。

POU名の後ろにカーソルを置き、OKボタンをクリックします。

POUが、正符号に「値 ： アドレス」の形で表示されます。

POUをダブルクリックするか、または「ツール」メニュー→「広げて表示／たたんで表示」をクリックすると、すべての変数の内容をたたんで表示したり、広げて表示できます。

8. 「編集」メニュー→「項目追加」→「直前／直後」、または  か  をクリックします。

9. モニタに必要な変数をすべて登録し終わるまで、手順 4 からの操作を繰り返します。

■変数値の表記形式の変更

■操作手順

1. データモニタのウィンドウにある、表記形式を変えたい箇所（モニタ欄）をクリックして選択します。

2. 「ツール」メニュー→「表示」→「10進数」、または  をクリックします。

「ツール」メニュー→「表示」→「2 進数」、または  をクリックします。

「ツール」メニュー→「表示」→「16進数」、または  をクリックします。

「ツール」メニュー→「表示」→「ASCII」、または  をクリックします。

値の変更

■操作手順

1. モニタ中のモニタ欄をクリックします。

2. 新しい値を入力します。

3. <Enter>キーを押します。

13.2.2 強制入出力

強制入出力：（ブール型変数にのみ可能です）

■操作手順

1. **モニタ中のモニタ欄をクリックします。**
2. **「ツール」メニュー→「強制入出力」をクリックします。**
3. **強制する値（0もしくは1）を入力します。**
4. **<Enter>キーを押します。**

   変数の色が指定色になります。

   すでに強制設定している変数の値を変更する場合は、前述の値の変更手順にしたがって変更してください。

強制解除

■操作手順

1. **モニタ中のモニタ欄をクリックします。**
2. **「ツール」メニュー→「強制解除」をクリックします。**

   変数の色が白色になります。

13.2.3 その他の操作

POU／変数の変更

■操作手順

1. **データモニタウィンドウにある別のPOU／変数を、表示するモニタ欄をクリックします。**
2. **「ツール」メニュー→「名前の変更」をクリック、または <Ctrl>キーと <Enter>キーを同時に押します。**
3. **<F2>キーを押して、新たにモニタする変数を選択します。**

   手順4へ進みます。

   または新しい名前を入力し、手順5へ進みます。
4. **任意の変数をダブルクリックします。**
5. **OKボタンをクリックします。**

データモニタの内容の保存

■操作手順

1. **「ツール」メニュー→「ファイルに保存」をクリックします。**
2. **ファイル名を入力します。**
3. **OKボタンをクリックします。**

「ツール」メニュー→「ファイルから読込み」をクリックすると、保存したファイルを開くことができます

POU／変数の削除

■操作手順

1. **リストから削除するPOU、または個々の変数をクリックして選択します。**
2. **「編集」メニュー→「削除」をクリックします。**

   リスト上の全ての登録データを削除するには、「編集」メニュー→「全てクリア」をクリックします。

13.3 POUヘッダモニタ

画面上にPOUボディを開いておくと、このボディに対応するPOUヘッダで宣言された変数をモニタすることができます。

■操作手順

1. **対象となるPOUをダブルクリックして、ヘッダとボディを開きます。**
2. **「モニタ」メニュー→「ヘッダモニタ」、または  をクリックします。**
   最終節で説明するのと同じデータモニタウィンドウが開きます。
   編集オプションも全く同じものです。

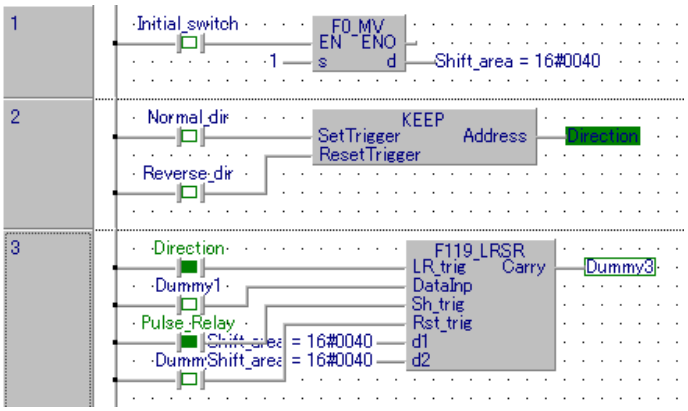
13.4 編集ウィンドウからのモニタ

13.4.1 編集ウィンドウでの変数値モニタ

「モニタ」メニュー→「モニタ実行」、または  をクリックすると、編集ウィンドウで変数の値がモニタできます。オンラインモードを解除するまで、変数値は反転表示になります。
モニタを終了するには、「モニタ」メニュー→「モニタ実行」、または  を再度クリックして「モニタ実行」のチェックマークを外します。

編集ウィンドウに表示される全ての変数値は、モニタ用に自動的に記録され、プログラムコードの右に表示されます。
BOOL型変数はプログラム中に直接、反転表示（TRUE）、通常表示（FALSE）のどちらかで表示されます。

編集ウィンドウをスクロールすると、新しく表示されたページの変数がモニタ用に自動的に記録され、以前のページが消去されます。



解説図の中で、先頭ブロックのBOOL型変数値は「FALSE」であることを示しています。

モニタ中には、接点に対して次の内容が適用されます。
・モニタ用に表示された矩形の状態によって、接点の状態が切り替わる様子を示します。
矩形内が設定色に変わった時、接点は閉じた状態で、矩形内が空白に変わった時は、接点が開いている状態であること示します。

・変数名の色の状態によって、現在値を示します。
変数の色が、モニタメニューで設定した色の時：TRUE
変数の色がテキストの設定色の時：FALSE

コイルに対しては次の内容が適用されます。
・モニタ用に表示された矩形内の色や、変数名の色の変化で現在値を示します。

入出力変数に対しては次の内容が適用されます。
・現在値は、変数名の右隣に表示されます。

13.4.2 オンラインモードでの変数値の変更

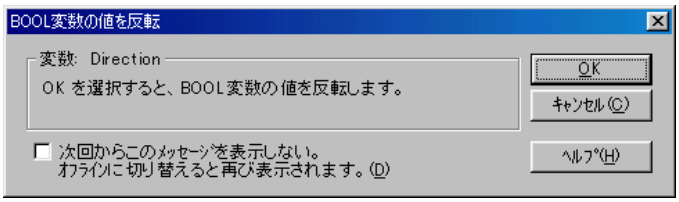
オンラインモードでPOUのボディをモニタ中、変数をダブルクリックすると、直接、変数値を変更できます。

ブール型変数

■操作手順

1. 値を変更する変数の名前をダブルクリックします。

2. 「BOOL型変数を変更します」と警告ダイアログボックスが開きますので、「OK」ボタンを押して照合します。



「次回からこのメッセージを表示しない」にチェックを入れると、オンラインモードを解除するまではBOOL型変数に対してこの警告は表示されません。

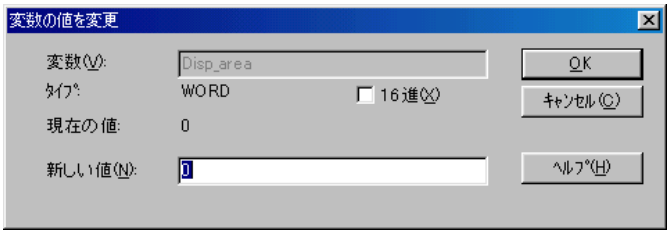
3. 「OK」ボタンをクリックすると、実行中のプログラムでBOOL型変数の値が反転ます。

BOOL 型以外の変数

■操作手順

1. 値を変更する変数の名前をダブルクリックします。

次のダイアログウィンドウが開きます。



2. 変数値を変更します。

3. 「OK」ボタンをクリックすると、実行中のプログラムで変数値を変更します。

13.5 タイムチャートモニタ

「モニタ」メニュー→「タイムチャート」をクリックすると、接点の現状や変数値を時間軸上に表示できます。

13.5.1 タイムチャートモニタの種類

■トレンドディスプレイ

PLCから任意のタイミングでデータを直接読み込んで、リアルタイムにグラフ表示する機能です。この機能は、弊社の全てのPLCでご使用できます。

■サンプリングトレース

PLCに対して、サンプリングを行う接点，データ，サンプリングの条件を登録します。
登録後にモニタを開始すると、PLCが接点やデータのサンプリングを行い、その結果をFPWIN Proに返します。FPWIN Proは、PLCから受信したサンプリングデータを元に、グラフ表示を行います。

ただしこの機能は、トレースメモリを持つPLCでしか使うことができません。

- FP2（オプションメモリAFP2203装着時）
- FP2SH
- FP10SH
- FP3（AFP3211C）

◆ここがポイント!

- これら2つの機能では、16までの接点か変数をウィンドウに表示することができます。複数のウィンドウを開くこともできます。

- トレンドディスプレイは、FPWIN Proがサンプリングを行うのに対し、トレースでサンプリングを行うのはPLC自身です。このためトレースではPLCのスキャン毎のサンプリングが可能になるので、より詳細なデバッグが行えます。

13.5.2 トレンドディスプレイ

Control FPWIN ProはPLCからデータを読み込み、タイムチャートの画面上に表示します。
表示する変数を、最初に選択しておきます。

変数の表示

■操作手順

1.「オンライン」メニュー→「オンラインモード」または  をクリックします。

2. 必要に応じてPLCをRUNモードに切り替えます。（必須ではありません）

3.「モニタ」メニュー→「タイムチャート」→「トレンドディスプレイ」または  をクリックします。

モニタウィンドウに表示する変数を、ダイアログボックスから選択します。

4. 変数パスのPOUをダブルクリックします。

5. 変数をダブルクリックします。

6. OKボタンをクリックします。

7. 変数が定数でなければ、最大値／最小値を設定してOKボタンをクリックします。

8.「編集」メニュー→「挿入」または  をクリックして、手順 4 から手順 7 を繰り返します。

16個までのBOOL型変数と3個までのWORD型変数が選択できます。

変数の削除

■操作手順

1.「編集」メニュー→「削除」あるいは  をクリックします。

2. 変数をダブルクリックします。

3.「削除」ボタンをクリックします。

◆ご 注 意！

・変数名の代わりに物理アドレス（DT0等）を入力することもできます。
・時間軸は設定できません。

13.5.3 サンプリングトレース

サンプリングトレースを使うと、変数の登録，パラメータの設定についてのオプションが使用できます。
(サンプリングタイム，トリガの状態，ゼロ点)
サンプリングトレースを始める前に、サンプリングする変数，サンプリングの開始時間，モニタ内容を設定する必要があります。

■操作手順

1. **「オンライン」メニュー→「オンラインモード」または**  **をクリックします。**
2. **「モニタ」メニュー→「タイムチャート」→「サンプリングトレース」または**  **をクリックします。**
   モニタモードで表示する変数を選択します。
3. **「編集」メニュー→「挿入」または**  **をクリックする。**
4. **変数パスの選択にあるPOUをダブルクリックします。**
5. **任意の変数をダブルクリックします。**
6. **「OK」ボタンをクリックします。**

一覧からの変数の削除

■操作手順

1. **「編集」メニュー→「削除」をクリックします。**
   「サンプル変数の削除」ダイアログボックスが開きます。
2. **削除する変数をクリックします。**
3. **OKボタンをクリックします。**

◆ご 注 意 !

・変数名の代わりに物理アドレス（DT0等）を登録することもできます。
・座標（最大値・最小値）は自動縮尺表示です。（設定することはできません）
・16個までのBOOL型変数と3個のWORD型変数が選択できます。
　選択した変数を一覧から削除もできます。

■パラメータの設定

「編集」メニュー→「サンプリングトレース設定」をクリックして、下記の設定を行います。

トリガビットの設定

トリガポイント（記録開始条件）の設定には3つの方法があります。

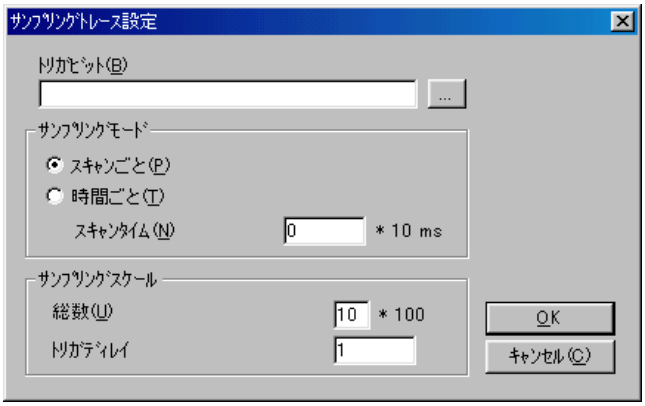
・BOOL型変数を使ったトリガの設定：
ビットマスクに、サンプリング開始時点が登録されているブール型変数を登録します。X0, Y0などが設定できます。

・「ツール」メニュー→「トリガ」をクリックして、手動でトリガを設定：
このときトリガボックスは空白のままです。

・PLCのプログラム（F155_SMPL参照）ごとにトリガを設定。
このときビットマスクボックスは空のままです。

サンプリングモードの設定

設定は、「編集」→「サンプリングトレース設定」で行います。



・スキャンごと：
PLCのスキャンサイクルごとに、変数値をバッファに格納します。
サンプルトレースウィンドウの時間軸は、1スキャンサイクルを1目盛とます。
・時間ごと：
変数値が記録されるスキャン時間を入力します。

サンプリングトレースウィンドウの時間軸上に、ms（ミリ秒）単位で1目盛に対する時間間隔を設定できます。

サンプリングスケールの設定

・総数：

サンプリング数。（最大1000）

・トリガディレイ：

例：　総数 = 1000

トリガディレイ = 800

PLCにトリガが設定されると、トリガ信号が発生する以前のデータ200個とトリガ信号が発生した後のデータ800個が記録されます。手動トリガで、トリガがONした後の全データを取得したい場合は、総数とトリガディレイを同数に設定してください。

4.「OK」ボタンをクリックします。

■パラメータ／変数のダウンロード：

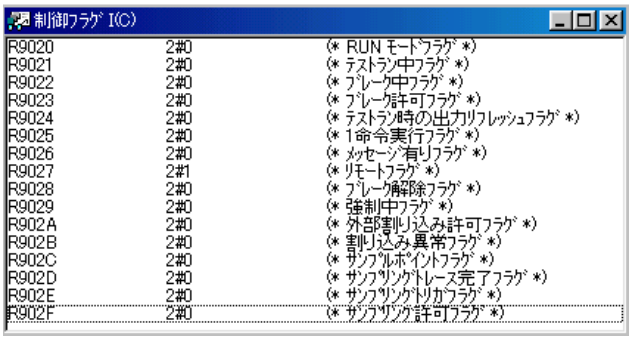
1.「ツール」メニュー→「パラメータのダウンロード」をクリックします。

サンプリングトレースフラグ（R902CからR902F）にFALSEがセットされます

■レジスタのモニタ：

1.「ツール」メニュー→「ステータス」をクリックします。

あるいは「モニタ」メニュー→「特殊内部リレー」→「制御フラグⅠ」をクリックします。



レジスタ　内容

DT9028

FP0(10k), FP2/2SH,
FP10S/FP10SH
サンプリング時間　0から3000 x 10 ms

R902C　サンプリング開始
スキャンサイクルごと　= 0 (FALSE)
設定時間間隔ごと　= 1 (TRUE)

R902D　サンプリングトレース完了
サンプリング完了時　= 1 (TRUE)

R902E　サンプリングトリガフラグ　= 1 (TRUE)
サンプリングトリガが設定されているとき

R902F　サンプリング　ON
フラグ　= 1 (TRUE)
PLC上でサンプル処理が実行されるとき(記録値)

■PLC のサンプリング処理実行

1.「ツール」メニュー→「アクティブにする」をクリックします。

ステータスバーにエディタ情報を表示している場合、「有効」の表示が0から1へ変わります。（サンプリングを作動させると、フラグ（R902F）にTRUEがセットされます。）

■手動によるトリガの実行

1.「ツール」メニュー→「トリガ」をクリックします。

ステータスバーにエディタ情報を表示している場合、「トリガ」の表示が0から1へ変わります。（サンプリングトリガ用フラグ（R902E）にTRUEがセットされます。）

「ビットマスク」フィールドにトリガ変数が登録されていないか、F155_SMPLファンクションを使ってトリガをアクティブにしていない場合は、ユーザ自身でトリガ時間を設定する必要があります。

■PLC からサンプリングトレース値を読み込む

1. ステータスバーにエディタ情報を表示している場合、「終了」の表示が1から0へ変わるまで（サンプリングトレース完了フラグ（R902D）がTRUEになるまで）待機し、「ツール」メニュー→「値のアップロード」をクリックします。

■ドライブにあるデータの読み込み／保存

1. アップロードした値を保存する場合は、タイムチャートモニタ画面をアクティブにしたまま、「ツール」メニュー→「ファイルに保存」をクリックします。

2. ファイル名（*.sam）を入力します。

3.「OK」ボタンをクリックします。

サンプリングトレースウィンドウの内容をファイルに保存して、後から参照することができます。保存したファイルのデータを参照する場合は、タイムチャートモニタ画面をアクティブにしたまま、「ツール」メニュー→「ファイルから読み込み」をクリックします。

14章

ユーザライブラリについて

14.1 概要

頻繁に使用するようなPOU、ファンクション(FUN)、ファンクションブロック(FB)を、ユーザライブラリに保存し、再利用することができます。Control FPWIN Proでは、最大50種類までのユーザライブラリを作成できます。
ユーザライブラリはハードディスク内に保存されます。他のユーザも、プロジェクトファイルへライブラリを呼び出せます。この場合は、新しいライブラリが作成されるのではなく、単にライブラリを参照しているだけです。
ライブラリオブジェクトを修正すると、その修正は、すべてのユーザに有効となります。

ユーザライブラリには、「オープン」「変更」「インストール」の3つのステータスがあります。どのステータであるかは、プロジェクトナビゲータに表示されています。ステータスによって、ライブラリの編集や使用が可能になります。

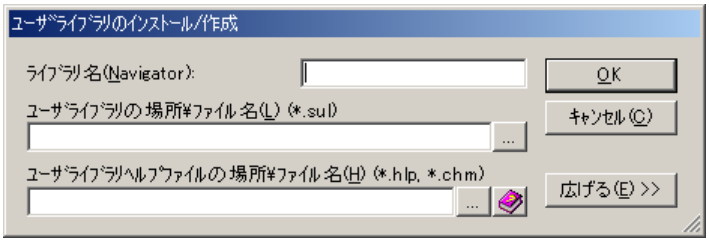
ユーザライブラリの作成方法を次章で説明します。

14.2 ユーザライブラリ作成手順

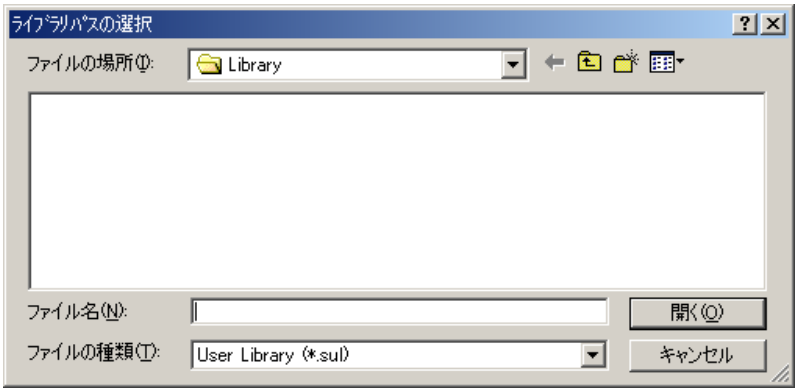
ここでは、例として、あらかじめ作成されたPumpというファンクションブロック(FB)を、Main_Pumpとうライブラリ名で登録する方法を説明します。

■操作手順

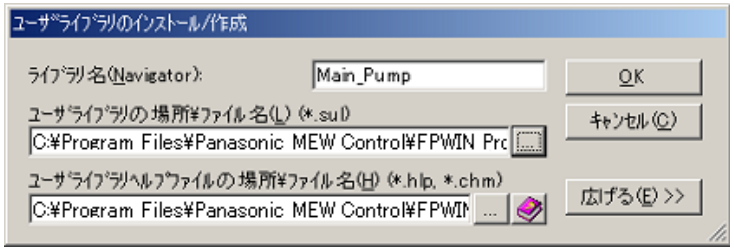
1. **プロジェクトナビゲータをアクティブにして、「編集」メニュー→「ライブラリ」→「インストール／作成」をクリックするか、プロジェクトウィンドウ」中の「ライブラリ」を右クリックし「ライブラリ」→「インストール／作成」をクリックしてください。**
   以下のウィンドウが表示されます。



2. **「ユーザライブラリの場所¥ファイル名(L) 」の項目の参照ボタン ... をクリックして、今から作成するユーザライブラリを保存するフォルダを指定してください。**



3. **ファイル名に “Main_Pump”と入力し、「開く」をクリックしてください。**
   以下の画面に切り替わります。

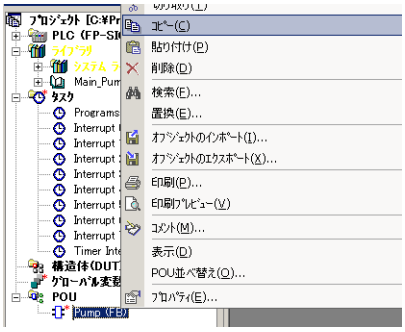


ヘルプファイルを別途作成している場合は、上記同様にヘルプファイルが保存されているフォルダのパスを指定してください。ヘルプがない場合は、そのまま[OK]ボタンをおしてください。

4. **プロジェクトナビゲータを見ると、Main_Pump というライブラリが表示されています。**

現在の状態では、Main_Pumpのライブラリは空の状態です。
これから、ライブラリMain_Pumpに、ファンクションブロックPumpを登録します。

5. **プロジェクトナビゲータ内のPOU内のPumpを右クリックし「コピー」を選択した後、同じくプロジェクトナビゲータ内のライブラリ内のMain_Pumpを右クリックし「貼り付け」してください。**



Main_Pump[オープン] の表示が、Main_Pump[変更] に切り替わります。

6. **Main_Pump[変更] を右クリックして「ライブラリ」→「閉じる」を選択し、保存してください。**Main_Pump[インストール] に表示が変更されます。

以降は、新規プロジェクトの作成時から、このライブラリを再利用することができるようになります。

改訂履歴

＊マニュアル番号は、表紙下に記載されています。

発行日付	マニュアル番号	改訂内容
2004 年 12 月	ARCT1F405	初版
2005 年 10 月	ARCT1F405-1	第 2 版
2006 年　1 月	ARCT1F405-2	第 3 版

ご注文に際してのお願い

本資料に記載された製品および仕様は、製品の改良などのために予告なしに変更（仕様変更、製造中止を含む）することがありますので、記載の製品のご使用のご検討やご注文に際しては、本資料に記載された情報が最新のものであることを、必要に応じ当社窓口までお問い合わせのうえ、ご確認くださいますようお願いします。
なお、本資料に記載された仕様や環境・条件の範囲を超えて使用される可能性のある場合、または記載のない条件や環境での使用、あるいは鉄道・航空・医療用などの安全機器や制御システムなど、特に高信頼性が要求される用途への使用をご検討の場合は、当社窓口へご相談いただき、仕様書の取り交わしをお願いします。

受入検査 〕
・ご購入または納入品につきましては、速やかに受入検査を行っていただくとともに、本製品の受入検査前または検査中の扱いにつきましては、管理保全に十分なご配慮をお願いします。

保証期間 〕
・本製品の保証期間は、ご購入後あるいは貴社のご指定場所への納入後１年間とさせていただきます。
なお電池や光源ランプなどの消耗品、補材については除かせていただきます。

保証範囲 〕
・万一、保証期間中に本製品に当社側の責による故障や瑕疵が明らかになった場合、当社は代替品または必要な交換部品の提供、または瑕疵部分の交換、修理を、本製品のご購入あるいは納入場所で無償で速やかに行わせていただきます。
ただし、故障や瑕疵が次の項目に該当する場合は、この保証の対象範囲から除かせていただくものとします。

1. 貴社側が指示した仕様、規格、取扱い方法などに起因する場合。
2. ご購入後あるいは納入後に行われた当社側が係わっていない構造、性能、仕様などの改変が原因の場合。
3. ご購入後あるいは契約時に実用化されていた技術では予見することが不可能な現象に起因する場合。
4. カタログや仕様書に記載されている条件・環境の範囲を逸脱して使用された場合。
5. 本製品を貴社の機器に組み込んで使用される際、貴社の機器が業界の通念上備えられている機能、構造などを持っていれば回避できた損害の場合。
6. 天災や不可抗力に起因する場合。

また、ここでいう保証は、ご購入または納入された本製品単体の保証に限るもので、本製品の故障や瑕疵から誘発される損害は除かせていただくものとします。
以上の内容は、日本国内の取引および使用を前提とするものです。

日本以外での取引および使用に関し、仕様、保証、サービスなどについてのご要望、ご質問は当社窓口まで別途ご相談ください。

制御機器関連お問い合わせ一覧

平成17年11月1日現在

●在庫・納期・価格など販売に関するお問い合わせは

松下制御機器株式会社

東京　〒105―8301　東京都港区東新橋1丁目5番1号　松下電工東京本社ビル8階　TEL.(03)6218―1919
大阪　〒571―8686　大阪府門真市大字門真1048番地　TEL.(06)6900―2740

東北営業所	〒981-3133	仙台市泉区泉中央1丁目23番地4号　ノースファンシービル5F	☎022―371―0766	FAX.022―371―7303
関東営業所	〒370-0071	高崎市小八木町1519番地	☎027―363―2033	FAX.027―362―6491
首都圏デバイス営業所	〒105-8301	東京都港区東新橋1丁目5番1号　松下電工東京本社ビル8階	☎03―6218―1920	FAX.03―6218―1931
東部グローバル営業所	〒105-8301	東京都港区東新橋1丁目5番1号　松下電工東京本社ビル8階	☎03―6218―1923	FAX.03―6218―1931
東京SCソリューション営業所	〒105-8301	東京都港区東新橋1丁目5番1号　松下電工東京本社ビル8階	☎03―6218―1922	FAX.03―6218―1941
茨城営業課	〒310-0851	水戸市千波町海道付2313番地	☎029―243―8868	FAX.029―243―8857
首都圏北営業所	〒330-0843	さいたま市大宮区吉敷町4丁目13番2号　大宮ダイヤビル6F	☎048―643―4735	FAX.048―643―4741
首都圏西営業所	〒190-0012	立川市曙町3丁目5番3号	☎042―528―2241	FAX.042―528―1963
松本営業課	〒399-0004	松本市市場3番10号	☎0263―28―0790	FAX.0263―28―0799
横浜SCソリューション営業所	〒220-0022	横浜市西区花咲町7丁目150番　ウエインズ＆イッセイ横浜ビル8F	☎045―321―1235	FAX.045―322―7080
東部車載営業所	〒105-8301	東京都港区東新橋1丁目5番1号　松下電工東京本社ビル8階	☎03―6218―1930	FAX.03―6218―1951
名古屋デバイス営業所	〒450-8611	名古屋市中村区名駅南2丁目7番55号　松下電工名古屋ビル北館6F	☎052―581―8861	FAX.052―581―6753
名古屋SCソリューション営業所	〒450-8611	名古屋市中村区名駅南2丁目7番55号　松下電工名古屋ビル北館6F	☎052―581―8861	FAX.052―581―6753
三重営業課	〒514-8555	津市大字藤方1668番地　松下電工(株)津工場内	☎059―246―8991	FAX.059―246―8991
豊田SCソリューション営業所	〒448-0857	刈谷市大手町2丁目29番地　INOビル2F	☎0566―62―6861	FAX.0566―62―6866
静岡営業所	〒420-0803	静岡市葵区千代田7丁目7番5号	☎054―261―7711	FAX.054―262―7342
浜松営業課	〒432-8052	浜松市東若林町1522番地	☎053―442―0531	FAX.053―442―0682
北陸営業所	〒920-8203	金沢市鞍月4丁目117番	☎076―268―9546	FAX.076―268―9547
富山営業課	〒930-0008	富山市神通本町2丁目2番19号	☎076―441―1910	FAX.076―441―1457
中部車載営業所	〒450-8611	名古屋市中村区名駅南2丁目7番55号　松下電工名古屋ビル北館6F	☎052―581―8861	FAX.052―581―6753
静岡営業課	〒420-0803	静岡市葵区千代田7丁目7番5号	☎054―261―7711	FAX.054―262―7342
京滋営業所	〒601-8127	京都市南区上鳥羽北花名町34番地	☎075―681―0237	FAX.075―671―2338
近畿デバイス営業所	〒571-8686	門真市大字門真1048番地	☎06―6900―2737	FAX.06―6900―5180
西部グローバル営業所	〒571-8686	門真市大字門真1048番地	☎06―6900―2737	FAX.06―6900―5180
近畿SCソリューション営業所	〒571-8686	門真市大字門真1048番地	☎06―6900―2733	FAX.06―6900―5180
姫路営業課	〒670-0055	姫路市神子岡前1丁目2番1号	☎0792―91―3927	FAX.0792―91―0612
中四国営業所	〒730-8577	広島市中区中町7番1号	☎082―247―9084	FAX.082―247―5925
岡山営業課	〒700-0973	岡山市下中野337番106号	☎086―245―3701	FAX.086―245―3731
四国営業課	〒761-0113	高松市屋島西町字百石1960番地	☎087―841―4473	FAX.087―843―0718
九州営業所	〒810-8530	福岡市中央区薬院3丁目1番24号	☎092―522―5545	FAX.092―523―9515
北九州営業課	〒802-0011	北九州市小倉北区重住3丁目2番10号	☎093―932―0652	FAX.093―931―2749
熊本営業課	〒860-0072	熊本市花園1丁目5番5号	☎096―353―4676	FAX.096―356―8797

上記の営業所の他に松下電工営業所でもお取り扱いいたしております。
松下制御機器のインターネットホームページ　http://www.mac-j.co.jp/

●技術に関するお問い合わせは

◆ 制御機器コールセンター

☎0120―101―550　※お問い合せ商品/リレー・機器用センサ・スイッチ・コネクタ・プログラマブルコントローラ・プログラマブル表示器・画像処理装置・タイマ・カウンタ・温度調節器
・サービス時間/9:00―17:00(11:30―13:00、当社休業日除く)

●FAX……06―6904―1573(24時間受付)

●webでのお問い合わせ…(制御機器WEB)http://www.nais-j.com/

●このマニュアルに使われている用紙は古紙配合率100％の再生紙を使用しております。
●この印刷物は環境にやさしい植物性大豆油インキを使用しています。

古紙配合率100%再生紙を使用しています　大豆油を主成分としたインキで印刷しています

●在庫・納期・価格など販売に関するお問い合わせは

●技術に関するお問い合わせは

制御機器コールセンター
0120-101-550
※お問い合せ商品 / リレー・機器用センサ・スイッチ・コネクタ・
プログラマブルコントローラ・プログラマブル表示器・
画像処理装置・タイマ・カウンタ・温度調節器
※サービス時間／9:00ー17:00〈11:30ー13:00、当社休業日除く〉
●FAX ····· 06-6904-1573（24時間受付）
Webでのお問い合わせ（制御機器WEB）http://www.nais-j.com/

松下電工株式会社　制御機器本部
制御デバイス事業部
〒571-8686 大阪府門真市門真1048
TEL.(06)6908-1131〈大代表〉

このマニュアルの記載内容は平成17年12月現在のものです。

■ARCT1F405-2　200601-3ZT　Printed in Japan