

# はじめに

このたびは、ロジックプログラム開発ソフト「Pro-Control Editor Ver.5.1」をご採用いただき、誠にありがとうございます。

この製品を正しくご使用いただくために、マニュアル類をよくお読みください。

また、マニュアル類は必ずご利用になる場所のお手元に保管し、いつでもご覧いただけるようにしておいてください。

## おことわり

- (1) 「Pro-Control Editor」(以下本製品といいます)のプログラムおよびマニュアル類は、すべて(株)デジタルの著作物であり、(株)デジタルがユーザーに対し「ソフトウェア使用許諾条件」に記載の使用権を許諾したものです。当該「ソフトウェア使用許諾条件」に反する行為は、日本国内外の法令により禁止されています。
- (2) 本書の内容については万全を期して作成しておりますが、万一お気づきの点がありましたら、(株)デジタル「サポートダイヤル」までご連絡ください。
- (3) 本製品を使用したことによるお客様の損害その他不利益、または第三者からのいかなる請求につきましても、当社はその責任を負いかねますので、あらかじめご了承ください。
- (4) 製品の改良のため、本書の記述と本製品のソフトウェアとの間に異なった部分が生じることがあります。最新の説明は、別冊ないし電子的な情報として提供していますので、あわせてご参照ください。
- (5) 本書は、(株)デジタルから日本国内仕様として発売された製品専用です。
- (6) 本製品が記録・表示する情報の中に、(株)デジタルおよび/または第三者が権利を有する無体財産権、知的所有権に関わる内容を含む場合がありますが、これは(株)デジタルがこれらの権利の利用について、ユーザーおよび/またはその他の第三者に、何らの保証や許諾を与えるものではありません。

© Copyright 2004 Digital Electronics Corporation. All rights reserved.

(株)デジタル 2004 August

商標・商号の権利については「商標権などについて」をご覧ください。

# 商標権などについて

本書に記載の社名、商品名は、各社の商号、商標(登録商標を含む)またはサービスマークです。本製品の表示・記述の中では、これら権利に関する個別の表示は省略しております。

商標等	権利者
Microsoft, MS, MS-DOS, Windows, Windows 95, Windows 98, Windows Me, Windows NT, Windows 2000, Windowsエクスプローラ, Microsoft Excel, Windows XP	米国Microsoft社
Intel, Pentium	米国Intel社
Pro-face, Flex Network	(株)デジタル
Ethernet	米国Western Digital社
NEC, PC-9800	日本電気(株)
IBM, PC/AT	米国IBM社
Adobe, Acrobat	アドビシステムズ社

なお、上記商号・商標類で、本書での表記と正式な表記が異なるものは以下の通りです。

本書での表記	正式な表記
Windows 95	Microsoft <sup>®</sup> Windows <sup>®</sup> 95 オペレーティングシステム
Windows 98	Microsoft <sup>®</sup> Windows <sup>®</sup> 98 オペレーティングシステム
Windows Me	Microsoft <sup>®</sup> Windows <sup>®</sup> Me オペレーティングシステム
Windows NT	Microsoft <sup>®</sup> Windows NT <sup>®</sup> オペレーティングシステム
Windows 2000	Microsoft <sup>®</sup> Windows <sup>®</sup> 2000 オペレーティングシステム
Windows XP	Microsoft <sup>®</sup> Windows <sup>®</sup> XP オペレーティングシステム
MS-DOS	Microsoft <sup>®</sup> MS-DOS <sup>®</sup> オペレーティングシステム

# 対応機種一覧

Pro-Control Editor Ver.5.1で対応している機種は以下のとおりです。本書では以下のようなシリーズ名または商品名を用いて説明します。下表の「GPタイプ」とは、GP-PRO/PB for Windows Ver.7.1で選択する時の名称です。

シリーズ名		商品名	型式	GPタイプ		
GLC100シリーズ		GLC100L	GLC100-LG41-24V	GLC100L		
		GLC100S	GLC100-SC41-24V	GLC100S		
GLC300シリーズ		GLC300T	GLC300-TC41-24V	GLC300		
GLC2000シリーズ	GLC2300シリーズ	GLC2300L	GLC2300-LG41-24V	GLC2300L		
		GLC2300T	GLC2300-TC41-24V	GLC2300		
	GLC2400シリーズ	GLC2400T	GLC2400-TC41-24V	GLC2400 <sup>1</sup> 「Rev.*-無し,1」		
				GLC2400 <sup>1</sup> 「Rev.*-2以降」		
	GLC2500シリーズ	GLC2500T	GLC2500-TC41-24V	GLC2500		
			GLC2500-TC41-200V	GLC2500		
	GLC2600シリーズ	GLC2600T	GLC2600-TC41-24V	GLC2600 <sup>1</sup> 「Rev.*-無し,1」		
				GLC2600 <sup>1</sup> 「Rev.*-2以降」		
		GLC2600-TC41-200V	GLC2600			
LTシリーズ	LT Type Aシリーズ	LT Type A1	GLC150-BG41-XY32SK-24V	LT TypeA		
		LTC Type A1	GLC150-SC41-XY32SK-24V	LTC TypeA		
		LT Type A2	GLC150-BG41-XY32SC-24V	LT TypeA		
	LT Type B/B+シリーズ	LT Type B	GLC150-BG41-FLEX-24V	LT TypeB/B+		
		LT Type B+	GLC150-BG41-XY32KF-24V			
		LTC Type B+	GLC150-SC41-XY32KF-24V	LTC TypeB+		
	LT Type Cシリーズ	LT Type C	GLC150-BG41-RSFL-24V	LT TypeC		
	LT Type Hシリーズ	LT Type H1		GLC150-BG41-ADK-24V	LT TypeH	
				GLC150-BG41-ADPK-24V		
				GLC150-BG41-ADTK-24V		
		LTC Type H1			GLC150-SC41-ADK-24V	LTC TypeH
					GLC150-SC41-ADPK-24V	
					GLC150-SC41-ADTK-24V	
	LT Type H2			GLC150-BG41-ADC-24V	LT TypeH	
GLC150-BG41-ADPC-24V						
GLC150-BG41-ADTC-24V						

1 リビジョンの見分け方については、参照「GLC2400/GLC2600をご使用のお客様へ」

# マニュアルの読み方

「GP-PRO/PB C-Package03」のマニュアルは7冊で構成されています。マニュアルの内容は別記の表をご覧ください。また、データファイルとして補足説明や機能の追加・修正情報が添付されている場合があります。

[スタート]ボタンをクリックし、[プログラム][Pro-face][ProPB3 C-Package]の順にポイントし、[お読みください]をクリックして表示された内容をご覧ください。

(株)デジタル製ハードウェアに関する詳しい説明は、各機種の「ユーザズマニュアル」(別売)をご覧ください。

<b>GP-PRO/PB C-Package03</b>	
導入ガイド	インストール方法と基本的なアプリケーションの開発方法について説明します。
<b>Pro-Control Editor Ver.5.1</b>	
ユーザズマニュアル	GLC/LTとの組み合わせに関するソフトウェア的な設定や変数、命令について説明します。
オペレーションマニュアル (本書)	準備から運転までの操作を習得するための演習とエラーメッセージの一覧を説明します。Pro-Control Editorで登録した変数をGP-PRO/PB で使用する方法などについても説明します。
<b>GP-PRO/PB for Windows Ver.7.1</b>	
オペレーションマニュアル	GP画面を作成するソフトウェアの操作手順と機能を説明します。
タグリファレンスマニュアル	GPの画面上機能を指定する「タグ」について説明します。
パーツリスト	GP画面を作成するソフトに用意されているパーツと図記号について説明します。
機器接続マニュアル (PLC接続マニュアル)	GPと各社PLC、温度調節器、インバータなどの接続について説明します。

GP-PRO/PB のマニュアルはGP画面の作成について書かれています。GLC/LTの作画を行う場合は、GPをGLC/LTに読み替えて操作してください。

上記のPDFマニュアル以外にもオンラインヘルプで詳しく説明していますのでご覧ください。

タグなどのアドレス設定時は標準インストール時にインストールされるレイアウトシートを利用されると便利です。

レイアウトシートには「デバイス割り付け表」と「タグレイアウトシート」があります。

それぞれMicrosoft Excel データとしてインストールされているのでご利用ください。

各ファイルの場所とファイル名を以下に示します。

なお、Microsoft Excel のご利用方法は該当商品マニュアルを参照ください。

フォルダ名	ファイル名	内容
Pro-face¥ propbwin¥sheet	Device1J.xls	デバイス割り付け表
	TAG1J.xls	タグレイアウトシート
	TAG2J.xls	
	TAG3J.xls	
	TAG4J.xls	

# お問い合わせ

Pro-Control Editor に関するご質問は、「サポートダイヤル」までお問い合わせください。

## お問い合わせ先

月～金 9:00～17:00

大阪 TEL (06)6613-3115

東京 TEL (03)5821-1105

名古屋 TEL (052)932-4093

月～金 17:00～19:00

専用ダイヤル TEL (06)6613-3206

土・日・祝日(12月31日～1月3日を除く) 9:00～17:00

専用ダイヤル TEL (06)6613-3206

ホームページからのお問い合わせには随時承ります。

URL <http://www.proface.co.jp/>

# 目次

はじめに .....	1
商標権などについて .....	2
対応機種一覧 .....	3
マニュアルの読み方 .....	4
お問い合わせ .....	5
表記のルール .....	10
使用上の注意 .....	12
制限事項について .....	13
以前のバージョンをご使用のお客様へ .....	14
GLC2400/GLC2600 をご使用のお客様へ .....	14

## 第 1 章 Pro-Control の基本事項

1.1 Pro-Control Editor とは .....	1-1
---------------------------------	-----

## 第 2 章 プログラムの作成

1. レッスンを始める前に .....	2-2
2. ロジックプログラムを作成する前にオプション画面で設定を行う .....	2-4
3. ツールバーのカスタマイズ .....	2-8
4. ファンクションキーでのプログラミング補助 .....	2-11
2.1 起動のしかた .....	2-13
2.2 変数の作成 .....	2-14
2.2.1 変数一覧の作成 .....	2-14
2.2.2 変数タイプの指定 .....	2-15
2.2.3 変数一覧のインポート / エクスポート .....	2-16
2.2.4 プログラムの保存 .....	2-21
2.3 行、命令、および分岐の挿入 .....	2-22
2.3.1 行の挿入 .....	2-22
2.3.2 行の削除 .....	2-24
2.3.3 命令の挿入 .....	2-25
2.3.4 命令の削除 .....	2-30
2.3.5 命令のコピー & 貼り付け .....	2-31
2.3.6 分岐の挿入 .....	2-32
2.3.7 初期化ロジックプログラム .....	2-34

---

2.4	命令への変数の割り付け	2-36
2.4.1	オペランド	2-36
2.4.2	変数の入力	2-37
2.4.3	作業の完了	2-39
2.5	ロジックプログラムのドキュメント化	2-42
2.5.1	プログラムコメントの追加	2-42
2.5.2	行コメントの追加	2-43
2.5.3	変数へのコメントの追加	2-44
2.5.4	[コメント一覧]ウィンドウ	2-45
2.6	行のコピー、切り取りおよび貼り付け	2-46
2.6.1	行のコピー	2-46
2.6.2	行の貼り付け	2-46
2.6.3	[切り取り]コマンドの使用	2-47
2.7	サブルーチンおよびラベル	2-48
2.7.1	サブルーチンの挿入	2-48
2.7.2	ラベルの挿入	2-50
2.8	ロジックプログラム内の移動	2-51
2.8.1	[検索]コマンド	2-51
2.8.2	[リファレンス]コマンド	2-52
2.8.3	[リファレンス]ダイアログボックスと他のウィンドウの併用	2-53
2.8.4	ブックマークの使用	2-54
2.8.5	[指定行へ移動]コマンドの使用	2-55
2.8.6	[指定ラベルへ移動]コマンドの使用	2-55
2.9	I/Oの割り付け	2-56
2.9.1	実I/Oへの変数の割り付け	2-56
2.9.2	[I/O設定]ウィンドウの割り付け解除	2-62
2.9.3	I/Oに割り付けた変数を命令で使用方法	2-62
2.9.4	I/O設定のインポート/エクスポート	2-63
2.10	プログラムエラーチェック	2-69
2.11	ロジックプログラムの印刷	2-71
2.12	ロジックプログラムのインポート/エクスポート	2-73
2.12.1	エクスポート	2-73
2.12.2	インポート	2-75
2.13	画面プログラムの開発	2-77

## 第3章 ロジックプログラムを実行する

3.1	コントローラの設定	3-1
3.1.1	ロジックプログラム書き込み	3-6
3.1.2	モニタモードへの移行	3-7
3.2	コントローラの運転 / 停止	3-8
3.3	システム変数によるプログラムのトラブルシューティング	3-10
3.4	システム変数の表示	3-10
3.5	ロジックプログラム読み込み	3-11
3.6	ロジックプログラム照合	3-12
3.7	プロパティ	3-12

## 第4章 モニタモードでの動作確認

4.1	編集を始める前に	4-1
4.2	モニタモード編集にカラーを使用する	4-2
4.3	ビット変数を ON/OFF する	4-3
4.4	ビット変数を強制的に ON/OFF する	4-4
4.5	変数値の変更	4-4
4.6	変数の属性変更	4-5
4.7	現在値表示リスト	4-6
4.8	オンラインエディット (対応機種 : GLC2000 シリーズ)	4-7
4.8.1	オンラインエディットの編集機能	4-7
4.8.2	データ保存	4-9

## 第5章 Editor と GP-PRO/PB

5.1	GP-PRO/PB での変数表示	5-2
5.1.1	Editor の起動手順	5-2
5.1.2	命令データの貼り付け	5-5
5.1.3	「ポンプチュートリアル」による作画例	5-12
5.2	GLC へのダウンロード	5-14
5.3	ポンププロジェクトの実行	5-15

## 第6章 Pro-Control と Pro-Server

6.1	GLC 変数のインポート	6-1
6.1.1	GLC 変数のインポート手順	6-2



## 第7章 エラーと警告

200-299 : ロジックのエラーと警告 .....	7-1
300-399 : 変数のエラーと警告 .....	7-4
400-499 : I/O のエラーと警告 .....	7-6
500-549 : 一般的な I/O ドライバのエラー .....	7-6
600-799 : PID 命令のエラー .....	7-7
800-899 : 特定の I/O ドライバのエラー .....	7-7
900-1000 : 特定の I/O ドライバの警告 .....	7-7

## 第8章 用語集

## 付録

付 . 1 固定変数モード .....	付 -1
---------------------	------

## 索引

# 表記のルール



本書は、以下のルールで表記します。

わかりにくいところなどは「サポートダイヤル」までお問い合わせください。「サポートダイヤル」では、(株)デジタル製品についての技術的なご質問・ご相談にお答えします。

なお、パソコンやWindowsそのものに関することは、パソコンをお買い上げの販売店、メーカーにお問い合わせください。


## 安全に関する注意表記

本製品のご使用上、安全に関して重要な説明には、以下の表示を添えています。

表示	意味内容
 警告	この表示を無視して誤った取り扱いをすると、人が死亡または重傷を負う可能性が想定される内容を示します。
 注意	この表示を無視して誤った取り扱いをすると、人が傷害を負ったり、物的損害の発生が想定される内容を示します。
<b>重要</b>	この表示の説明に従わない場合、機器の異常動作やデータの消失などの不都合が起こる可能性があります。
<b>強制</b>	必ず実施していただきたい操作、作業などを表します。
<b>禁止</b>	決して行ってはならない操作、作業などを表します。

## 説明のための表記

本書では、説明の便宜のため、以下のように表記します。

Pro-Control Editor	GLCまたはLTのロジックプログラムを作成/転送/モニタを行う機能をもったソフトウェアです。
コントローラ	GLCまたはLTに組み込まれている制御機能を指します。
GLC	GLCまたはLTを指します。
GP-PRO/PB (画面作成ソフト)	GP-PRO/PB for Windows Ver.7.1を指します。
外部通信機器	温調器やインバータなどの周辺機器を指します。ただし、Flex Network、DIOで接続する機器を除きます。
<b>重要</b>	この表示の説明に従わない場合、機器の異常動作やデータの消失などの不都合が起こる可能性があります。
	使用するに際して、ポイントとなる項目です。
	脚注で説明している語句についています。
<u>参照</u>	関連する説明が記載されている項目を示します。

## キーボード対応表

本書では、パソコンのキーを以下のように表記します。

機種によってやや異なりますが、この対応で読み替えてください。

機種 表記	PC/AT互換機		PC-9800シリーズ
	日本語 106キーボード	英語 101キーボード	
Esc	Esc	Esc	ESC
Tab	Tab ⇐⇒	Tab ⇐⇒	TAB
Ctrl	Ctrl	Ctrl	CTRL
Shift	Shift	Shift	SHIFT
Alt	Alt	Alt	GRPH
Delete	Delete	Delete	DEL
Back space	Back space	Back space	BS
日本語入力	Alt + 半角/全角	Alt + ~	CTRL + XFER

：日本語入力のオン / オフ操作は、使用する日本語FEPによって異なります。

## モデル環境

本書で、操作や機能を説明する場合のモデルとなるシステム構成は以下の通りです。

これ以外のシステム構成では、表示や各部の名称が異なることがありますが、同等の機能をもつものと読み替えてください。

機材・ソフト	モデルシステムの仕様	備考
パソコン	PC/AT互換機	
メモリ	64Mバイト	
マウス	PC/AT互換機対応マウス	
OS	Windows 98	
GLC	GLC2300シリーズ	
パソコンとGLCとの 接続方法	RS-232C	(株)デジタル製ケーブル GPW-CB02使用

# 使用上の注意

## 使用上の注意

本製品の使用について

誤動作や事故の原因となりますので、以下の点にご注意ください。



### 警告

本誌に記載されている参考回路やアプリケーション事例は参考用ですので、ご採用に際してはシステム・機器・装置の機能や安全性をご確認の上、ご使用ください。本誌に記載のない条件や環境での使用、および原子力用・電力用・航空宇宙用・医療用・乗用移動体用の機器あるいはシステムなどの極めて高度な信頼性・安全性が求められている用途への使用を想定しておりません。これらの用途には使用できません。

タッチパネルスイッチは非常停止用スイッチとして使えません。産業用ロボットほか、労働大臣が指定する産業用機械設備の非常停止用スイッチとしては、必ず人間が直接操作するスイッチを設置することが関係法令で義務づけられています。また、これ以外の装置設備でも、安全確保のため、必ず同様のスイッチを設置してください。GLC/LTの故障により重大な事故または損失の発生が予測される設備への適用に関しては、バックアップやフェイル・セーフ<sup>1</sup>を系統的に設置してください。

### 禁止

GLC/LTは一般工業等を対象とした汎用品として制作されたもので、人命にかかわるような状況下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。よって、人命や重大な物的損傷にかかわる制御には使用しないでください。

プログラム使用中に、パソコン本体の電源をOFFしないでください。

テキストエディタなどを使用して、本製品のプロジェクトファイルの中身を変更しないでください。

GLC/LTがサポートしていない機能を使用した画面は、GLC/LTに転送しないでください。

### 重要

本アプリケーションは、Administrator 権限でご使用ください。もし異なる場合、誤動作する恐れがあります。

## ディスクの取り扱いについて

ディスクの破損・故障を防ぐため、以下の点にご注意ください。

**強制** ・ パソコン本体の電源のON/OFFは、ディスクを抜いてから行ってください。

**禁止** ・ ディスクドライブのランプが点灯している時は、CD-ROMを取り出さないでください。  
・ CD-ROMの記録面、フロッピーディスクの磁性体面（シャッターの中）に手を触れないでください。  
・ 極端な高温や低温、湿気やホコリの多い場所にディスクを置かないでください。  
・ フロッピーディスクを、ステレオのスピーカーやテレビ、磁気治療器などに近づけないでください。

1 オペレータの操作ミスや、センサーやコントローラの誤動作による被害を最小限にする工夫をいいます。

# 制限事項について

本製品には、以下のような制限があります。

- GLC100、GLC300、LTシリーズでは、Pro-Server with Pro-Studio for Windows(2Way Driver)をサポートしていません。
- GP-PRO/PB はご使用のパソコン内部の文字フォントやグラフィック機能を使用して表示します。このため、これらの表示はGLC/LTへ転送後、GLC/LT上での表示とパソコン上での表示に多少の相違が生じる場合があります。あらかじめご了承ください。
- GLC100 では、GP-PRO/PB 機能の中で AUX 出力 / インチングタグ / t タグ AUX 出力 / バックアップ機能などその他のGP-370 で使用できない機能は使用できません。
- Pro-Control Editor の変数にはデバイスタイプがないため、GP-PRO/PB 機能のEタグ、Kタグの間接指定のデバイスタイプ&アドレスの指定はできません。
- GLC/LTのロジックタイムが長い場合、折れ線グラフのサンプリングタイムが守れない場合があります。
- 実数はEタグ、Kタグのfloatで使用してください。ただし、GLC/LTの変数(64ビット)とタグ(32ビット)の精度の違いによる誤差があります。
- メモリリンク方式を使用の場合、GLC/LT変数は、折れ線グラフの一括表示ができません。
- GLC/LTの変数は32bit デバイスのLow/High順で扱われます。
- GLC100では、Qタグのサブ表示は使用できません。
- GLC/LTのロジックタイムが長い場合、音声の再生時に音が途切れることがあります。
- 整数型変数のビット指定においてTタグ、Wタグのビット書き込み(「反転」以外)を行うと、該当する整数型変数の指定したビット以外はすべてクリア(0)します。
- ビット動作が反転のTタグに同一整数型変数の異なるビットを指定したものを複数重ねて配置した場合は、最後に配置したTタグのみ有効となります。
- GLC/LTの保持型変数のデータを保持するSRAMはリチウム電池でバックアップされていますが、バックアップ期間は初期状態(満充電)で約60日、電池寿命時で約6日となりますので、この期間以上にバックアップが必要な場合は、ホストでバックアップするシステム構成をとってください。
- GLC2400、GLC2500、GLC2600では、AUXとしてリセット入力以外は使用できません。
- オンラインエディットではSRAMに保存されているロジックプログラムを編集します。オンラインエディット後、電源OFF時のバッテリー切れなどによりSRAMのデータが失われた場合、FEPRMのデータが読み出されます。オンラインエディットした場合は、GLCのオフラインメニューで[FEPRMへのコピー]を実行するか、Pro-Control EditorでPRWファイルとして保存することにより、バックアップしてください。また、LTではオンラインエディットを行うことができません。
- オンラインエディットでは、通信パケット制限によって要素数の多い配列変数、多量の変数を含む行、多量の命令を含む行を扱えない場合があります。
- パソコンとGLC/LTとの実数の精度の違いによって、モニタリングで表示する値と入力した値が一致しない場合があります。
- ロジックプログラムと画面データで同じLSエリアを共有する場合は、全てLogicシンボルのLS変数(LS<\*>)で指定してください。

## 以前のバージョンをご使用のお客様へ

Ver.3.0以前のPro-Control Editorをご使用のお客様は下記の注意事項をご覧ください。

Ver.3.0までのPro-Control EditorではロジックプログラムはWLLファイルとして保存されていましたが、Ver.4.0以上ではロジックプログラムはGP-PRO/PB のプロジェクトファイルに含まれてPRWファイルとして保存されます。

Ver.3.0以下のバージョンで作成したロジックプログラムを流用する場合は、WLLファイルをPRWファイルへインポートする必要があります。

参照 「2.12. ロジックプログラムのインポート / エクスポート」

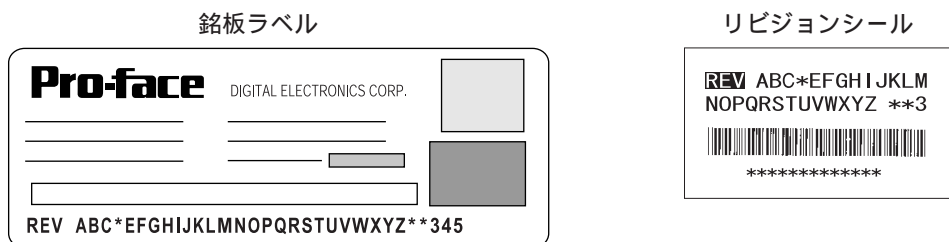
## GLC2400/GLC2600 をご使用のお客様へ

リビジョンは、GLC本体に貼り付けられている銘板ラベルまたはリビジョンシールで判定します。「REV」欄に「\*」マークおよびマジック等でマーキングがついている位置のアルファベットおよび数字がリビジョンになります。

数字部分について、複数の「\*」マークおよびマジック等でマーキングがついている場合は、最も大きな数がリビジョンになります。

### リビジョンの判定方法

下の例では、本来「D」および「1」「2」がある位置に「\*」マークがあるので、リビジョン「D-2」ということになります。



### リビジョンによる区別

下表のようにリビジョンの区別を定義します。

名称	ラベルの記載内容
「Rev.*-無し、1」	リビジョンの数字部分が表示されていない、または「1」の場合
「Rev.*-2以降」	リビジョンの数字部分が「2」以降の場合

# 第1章

## Pro-Control の基本事項

### 1.1 Pro-Control Editor とは

Pro-Control Editorは、GLCのロジックプログラムを作成するソフトです。(以下、単にEditorといたします)

Editorは、以下のような機能を含んでいます。

- ・DIOユニット用ドライバ
- ・ユニワイヤ拡張 I/F ユニット用ドライバ
- ・Flex Network I/F ユニット用ドライバ
- ・LT Type Hシリーズ用ドライバ
- ・ロジックプログラムの編集機能
- ・ロジックプログラム転送機能
- ・クロスリファレンスレポート
- ・モニタ機能
- ・オンラインエディット機能<sup>1</sup>
- ・イーサネット経由の通信<sup>1</sup>

Editorでは、国際基準 IEC61131-3 に準拠したロジックプログラムをWindowsの使いやすさを活かした簡単な操作で開発できます。

ロジックプログラムはGLCにダウンロードされ、GLC上で動作します。

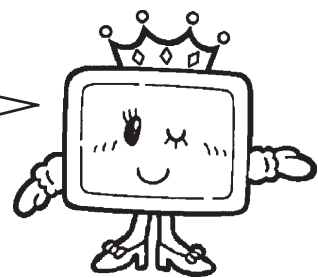
Editorで作成した変数は、画面作成ソフト「GP-PRO/PB for Windows Ver.7.1」に取り込まれ、GLCの表示機能（スイッチ、ランプなど）で変数名を共有できます。

---

<sup>1</sup> GLC2000シリーズのみサポートしています。

# MEMO

このページは、空白です。  
ご自由にお使いください。





# 第2章

# プログラムの作成

本章では、Editorを使って、プログラムモードでロジックプログラムを作成する方法を、順序を追って説明します。

Editorの起動方法については「GP-PRO/PB オペレーションマニュアル 1.2 起動から終了まで」を参照してください。また、Editorの各部の詳細説明は、ユーザーズマニュアル、オンラインヘルプを参照してください。

## 1. レッスンを始める前に

本章の各レッスンは、Editorの操作手順について、練習例に沿って説明しています。この様に練習例に沿って演習することをチュートリアルといいます。

ここでは、Editorを使ってファーストフードレストランのソフトドリンクマシンの動作を想定して制御するロジックプログラムを作成します。このマシンには、以下の機能があります。

- ・ボタンを1度押すことによって、大/中/小のカップサイズに適した量のドリンクを自動的に吐出する
- ・カップがディスペンサーの下にある場合だけ、氷またはソーダを吐出する
- ・機械に電源を投入したあとに、機械によって装填されたカップの数をカウントする

### ロジックプログラムおよび画面の完成例

本レッスンのロジックプログラム及び作画画面の完成品は、「C:\¥Program Files¥Pro-face¥ProPBWin¥Sample」のフォルダの中に「Soda.prw」ファイルとして付属しています。作成方法が分からない場合や検索などの練習の場合は、このファイルを開いて参考にしてください。Editorの各部の詳細説明は、オンラインヘルプを参照してください。

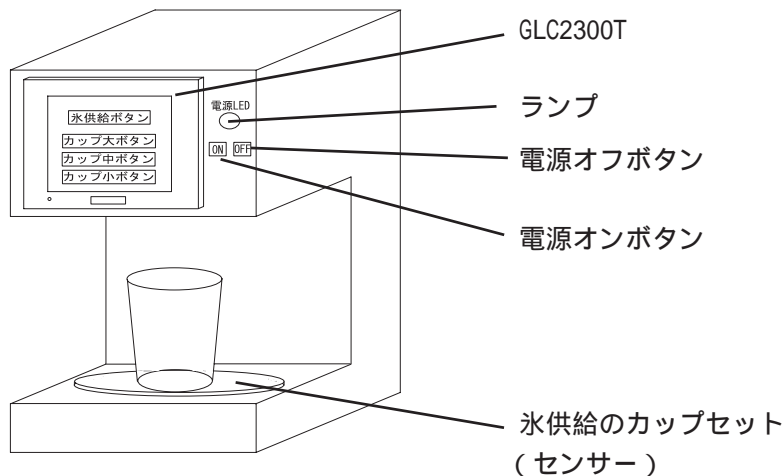
<ロジックプログラム>



<画面>

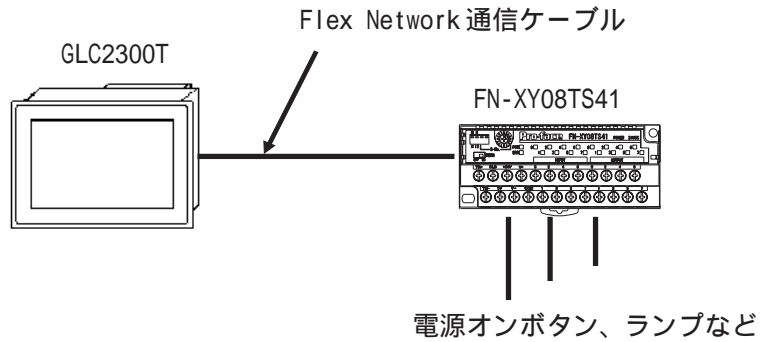


### ソフトドリンクマシン



## ハードウェア構成

## 構成図



## I/O 設定の割り付け

「氷供給ボタン」、「カップ大ボタン」、「カップ中ボタン」、「カップ小ボタン」は、GLC からのタッチパネル入力とするので割り付けません。

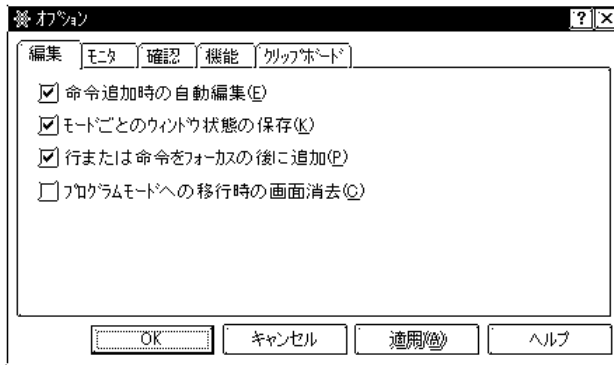
変数名	端子タイプ	端子番号
電源オンボタン	入力	I0
氷供給のカップセット	入力	I2
電源オフボタン	入力	I6
ランプ	出力	Q0
氷供給	出力	Q1
ソーダバルブ	出力	Q2

## 2. ロジックプログラムを作成する前にオプション画面で設定を行う

ロジックプログラムの作成を始める前に、好みにあった動作にEditorの操作設定を合わせることができます。[オプション]ダイアログボックスでプログラムを作成し実行する方法をカスタマイズできます。

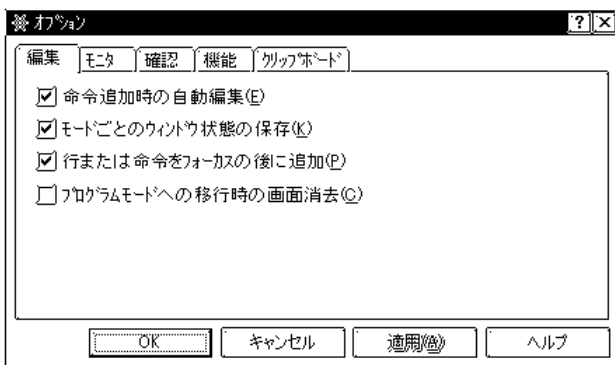
### オプション画面の設定手順

1. [ファイル]メニューから、[オプション]を選択します。[オプション]ダイアログボックスが表示されます。



2. 各項目のチェックボックスをクリックすると、その項目が選択またはクリアされます。  
[オプション]ダイアログボックスの中の各項目の意味は、次頁の表の通りです。

### 編集タブ



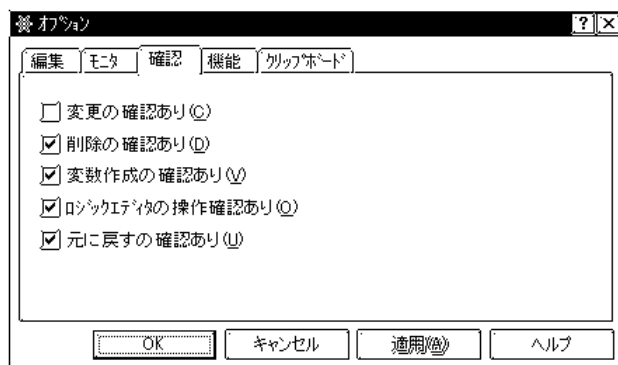
項目	説明
命令追加時の自動編集 (デフォルト=選択)	プログラムの中に挿入された新しい命令について、自動的に[オペランド]ウィンドウが開きます。
モードごとのウィンドウ状態の保存 (デフォルト=選択)	Editorの再起動やモード切り替えの際、前回の作業終了時に開いていたウィンドウが、そのときの状態で開かれます(ウィンドウのサイズ、位置など)。この設定は[現在値表示リスト]ウィンドウにも適用されます( [現在値表示リスト]ウィンドウは、現在のプログラムがモニタモードの場合、その変数の値を表示します)。
行または命令のフォーカスの後に追加 (デフォルト=選択)	この項目が選択されている場合、新しい命令は選択した命令の右側に挿入されます。また、ラベル、行、サブルーチンなどのオブジェクトは、選択している行の下に挿入されます。 この項目をクリアした場合、新しい命令は、左側、新しいオブジェクトは上側に挿入されます。命令の設定をされていない横ライン(接続線)を選択している場合は、設定に関わらず、命令はライン上に挿入されます。
プログラムモードへの移行時の画面消去 (デフォルト=クリア)	モニタモードからプログラムモードに移るときにEditorをクリアします。

## モニタタブ



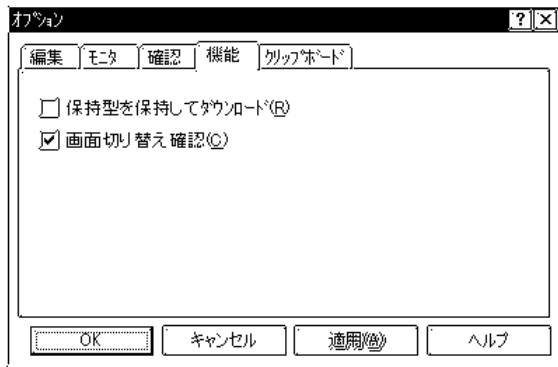
	項目	説明
モニタモードの更新	パワーフロー (デフォルト=選択)	コントローラの実行中にパワーフローが表示されます。パワーフローとは、コントローラが運転中に導通している行(ロジックプログラムで命令を記述する横方向のライン)を強調表示します。パワーフローの表示の更新はロジックの実行よりも遅れる場合がありますのでご注意ください。
	状態フロー (デフォルト=クリア)	コントローラの実行中に状態フローが表示されます。状態フローとは、コントローラが運転中に導通している命令を強調表示します。パワーフローと状態フローを同時に表示できます。
周期(ミリ秒) (デフォルト=500msec)		Editorがパワーフロー、状態フロー、現在値、およびステータスバーを更新するためにコントローラから新しいデータを要求する頻度を指定します。0~3000の範囲で設定してください。

## 確認タブ



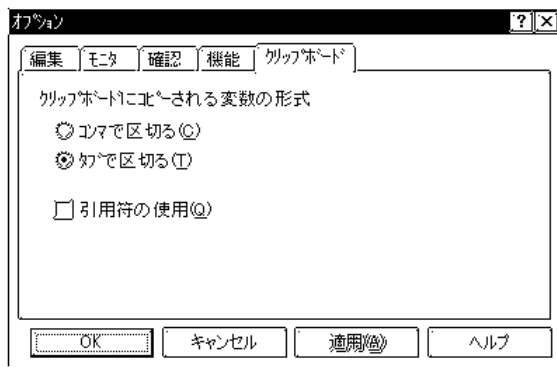
項目	説明
変更の確認あり (デフォルト=クリア)	Editorは、[更新]をクリックしたときにだけ変更を受け入れます。この項目を有効にした場合、Editorは変更の確認を要求します。
削除の確認あり (デフォルト=選択)	Editorは、プログラムの作成中に、すべての削除について確認を要求します。
変数の作成の確認あり (デフォルト=選択)	Editorは、プログラムの中での新しい変数の作成について確認を要求します。これは、オフライン環境にだけ適用されます。
ロジックエディタの操作確認あり (デフォルト=選択)	Editorは、コントローラの動作の変更(運転/停止、読み込み/書き込みなど)について確認するように要求します。
元に戻すの確認あり (デフォルト=選択)	[元に戻す]を実行する前に、確認するかどうかを指定します。

機能タブ



項目	説明
<p>保持型を保持してダウンロード (デフォルト=無効)</p>	<p>ロジックプログラム書き込みを行うときに、保持型変数の値を保持することが可能です。 GP-PRO/PB から転送を行う場合でもこの項目は有効となります。その場合は、確認ダイアログボックスは表示されませんが、保持型変数の保持が行われなかった場合は、そのまま転送を続けるか、中断するかを確認メッセージダイアログボックスが表示されます。</p> <p>有効</p> <p>ロジックプログラム書き込みを行うときに、保持型変数の値を保持することが可能です。 確認タブの[ロジックエディタの操作確認あり]を指定していない場合は、保持型変数の保持を行うかどうかのメッセージダイアログボックスが表示されません。</p> <p>無効</p> <p>ロジックプログラム書き込みを行うときに、全ての変数の値が初期化されます。</p>
<p>画面切り替え確認 (デフォルト=有効)</p>	<p>ロジックプログラムまたはPLCで、「#Screen」と「LS[8]」、「LS0008」、「PLCに割り付けた切り替え画面番号デバイス」に切り替え画面番号をセットして画面切り替えを行った場合に、画面の切り替え完了の確認をするかどうかを指定します。</p> <p>有効</p> <ul style="list-style-type: none"> <li>・ダイレクトアクセス方式の場合 画面の切り替えを確認した（システムデータエリアの表示中画面番号と切り替え画面番号が一致した）後に「#Screen」と「LS[8]」、「LS0008」、「PLCに割り付けた切り替え画面番号デバイス」に0を書き込みます。</li> <li>・メモリリンク方式の場合 画面の切り替えを確認した（システムデータエリアの表示中画面番号と#Screenが一致した）後に#Screenに0を書き込みます。</li> </ul> <p>無効</p> <ul style="list-style-type: none"> <li>・ダイレクトアクセス方式の場合 画面の切り替えを確認しても、「#Screen」と「LS[8]」、「PLCに割り付けた切り替え画面番号デバイス」の値はセットされた切り替え画面番号を保持します。</li> <li>・メモリリンク方式の場合 画面の切り替えを確認しても、「#Screen」の値はセットされた切り替え画面番号を保持します。</li> </ul>

## クリップボードタブ



	項目	この項目が選択された場合の動作
クリップボードにコピーされる変数の形式	コンマで区切る (デフォルト=クリア)	Editorの変数一覧からクリップボードにコピーされたフィールドがコンマで区切られます。 例) My_variable,Discrete,adescription
	タブで区切る (デフォルト=選択)	Editorの変数一覧からクリップボードにコピーされたフィールドがタブで区切られます。 例) My_variable [TAB] Discrete [TAB] adescription
	引用符の使用 (デフォルト=クリア)	Editorの変数一覧からクリップボードにコピーされたフィールドが区切り記号で区切られ、引用符で囲まれます。 例) "My_variable","Discrete","adescription"

このチュートリアルでは、デフォルト設定値を使用します。

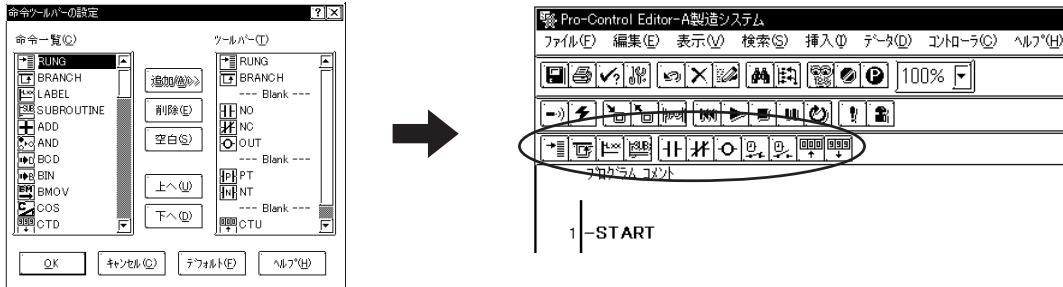
[キャンセル]をクリックし、[オプション]ダイアログボックスを閉じます。


キャンセルをクリックすることで、デフォルト設定値のままダイアログボックスを閉じることができます。

### 3. ツールバーのカスタマイズ

ロジックプログラムの作成を始める前に、ツールバーに利用頻度の高い命令のアイコンを配置させることができます。

下図のように[表示/命令ツールバーの設定]をクリックすると[命令ツールバーの設定]ダイアログボックスが表示され、ツールバーのアイコンをカスタマイズすることができます。



**MEMO** ・ ツールバーの  アイコンから[命令ツールバーの設定]ダイアログボックスを表示させることもできます。

#### 命令アイコン一覧

番号	アイコン	命令	内容
1		-	行の挿入
2		-	分岐の挿入
3		-	ラベル
4		-	サブルーチン
5		NO	a接点
6		NC	b接点
7		OUT	OUTコイル
8		NEG	反転コイル
9		SET	セットコイル
10		RST	リセットコイル
11		PT	立ち上がり接点
12		NT	立ち下がり接点
13		AND	論理積
14		OR	論理和
15		XOR	排他的論理和
16		NOT	ビット反転
17		MOV	転送
18		BMOV	ブロック転送
19		FMOV	フィル転送



番号	アイコン	命令	内容
20		SUM	合計
21		AVE	平均
22		BCNT	ビットカウント
23		ROL	左回転
24		ROR	右回転
25		SHL	左シフト
26		SHR	右シフト
27		RCL	キャリー付き左回転
28		RCR	キャリー付き右回転
29		SAL	算術左シフト
30		SAR	算術右シフト
31		ADD	加算
32		SUB	減算
33		MUL	乗算
34		DIV	除算
35		MOD	剰余算
36		INC	インクリメント
37		DEC	デクリメント
38		SQRT	平方根
39		EQ	比較: =
40		GT	比較: >
41		LT	比較: <
42		GE	比較: >=
43		LE	比較: <=
44		NE	比較:
45		TON	オンディレータイマ
46		TOF	オフディレータイマ
47	TP	パルスタイマ	
48		CTU	アップカウンタ
49		CTD	ダウンカウンタ
50		CTUD	アップダウンカウンタ

番号	アイコン	命令	内容
51		BCD	BCD変換
52		BIN	バイナリ変換
53		ENCO	エンコード
54		DECO	デコード
55		RAD	ラジアン変換
56		DEC	度変換
57		SCL	スケール変換
58		JMP	ジャンプ
59		JSR	ジャンプサブルーチン
60		RET	リターンサブルーチン
61		FOR	FOR
62		NEXT	NEXT
63		PID	PID演算
64		SIN	正弦
65		COS	余弦
66		TAN	正接
67		ASIN	逆正弦
68		ACOS	逆余弦
69		ATAN	逆正接
70		COT	余接
71		EXP	指数
72		LN	自然対数

## 4. ファンクションキーでのプログラミング補助

ロジックプログラムを作成する際に、ファンクションキーから ”命令の挿入 ” や ” 行の挿入 ” など利用頻度の高い機能をショートカットで呼び出すことができます。

呼び出すことができる機能については、下表のファンクションキー割り付け一覧を参照してください。

ファンクションキー割り付け一覧

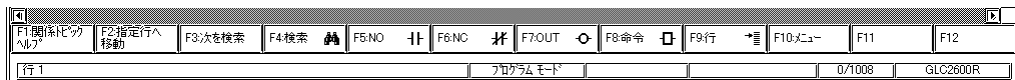
ファンクションキー	組み合わせ		
	-	Ctrlキー	Shiftキー
F1	関係トピックヘルプ	-	-
F2	指定行へ移動	-	-
F3	次を検索	-	-
F4	検索	-	-
F5	NO (a接点) 	-	NO-OR (a接点 OR) 
F6	NC (b接点) 	-	NC-OR (b接点 OR) 
F7	OUT (OUTコイル) 	PT-OR (立ち上がり接点 OR) 	PT (立ち上がり接点) 
F8	命令 (命令の挿入) 	NT-OR (立ち下がり接点 OR) 	NT (立ち下がり接点) 
F9	行 (行の挿入) 	分岐 (分岐の挿入) 	-
F10	メニュー	-	-



MEMO ・ [表示 / ツールバー / ファンクション] をクリックすると、Editor 下部にショートカットの可能な機能が表示されます。

### 組み合わせによる表示

#### ファンクションキーのみの場合



#### Ctrl キーと組み合わせた場合



#### Shift キーと組み合わせた場合



本演習を以下の順で説明します。

1. GP-PRO/PB C-Package を起動する

参照 「2.1 起動のしかた」

2. 新規作成で GLC のタイプ / 接続機器を決める

参照 「2.1 起動のしかた」

3. ロジックプログラムの開発

1. 変数を決める

Editor で作成するロジックプログラムの動作を設定する方法について説明します。また、使用する変数の作成と削除、初期値の設定方法を説明します。

参照 「2.2 変数の作成」

2. ロジックプログラムを作成する

行の作成、命令や分岐の挿入方法、行および行に関連づけられている命令や分岐の削除方法について説明します。

参照 「2.3 行、命令、および分岐の挿入」

3. ロジックプログラムに変数を割り付ける

ロジックプログラムの中の命令に変数を割り付ける方法について説明します。

参照 「2.4 命令への変数の割り付け」

4. コメントを入れる

ロジックプログラムにコメントを付ける方法について説明します。プログラム全体、特定の行、個別の命令にコメントを付ける方法です。

参照 「2.5 ロジックプログラムのドキュメント化」

5. 追加

行のコピー、切り取りおよび貼り付け方法を説明します。

参照 「2.6 行のコピー、切り取りおよび貼り付け」

6. サブルーチン

ロジックプログラムにサブルーチンやラベルを挿入する方法について説明します。

参照 「2.7 サブルーチンおよびラベル」

7. 検索

ロジックプログラムの中で、目的の回路をすばやく検索し移動する方法について説明します。

参照 「2.8 ロジックプログラム内の移動」

8. I/O 割り付け

ロジックプログラムの中の論理変数を実 I/O に割り付ける方法について説明します。

参照 「2.9 I/O の割り付け」

9. エラーチェック

ロジックプログラムのエラーをチェックする方法について説明します。

参照 「2.10 プログラムエラーチェック」

10. 印刷

ロジックプログラムを印刷する方法について説明します。

参照 「2.11 ロジックプログラムの印刷」

11. インポートとエクスポート

ロジックプログラムの「読み込み」と「書き込み」について説明します。

参照 「2.12 ロジックプログラムのインポート / エクスポート」

4. 画面プログラムの開発

ロジックプログラムとリンクした画面を GP-PRO/PB で作成します。

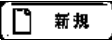
参照 「2.13 画面プログラムの開発」

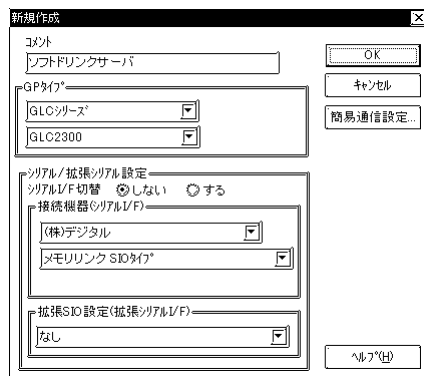
## 2.1 起動のしかた

Editorでロジックプログラムの作成を行うために、まずプロジェクトマネージャを起動します。

1. [スタート]ボタンをクリックし、[プログラム(P)]、[Pro-face]、[ProPB3 C-Package]の順にポイントし、[プロジェクトマネージャ]をクリックします。
2. プロジェクトマネージャが起動します。

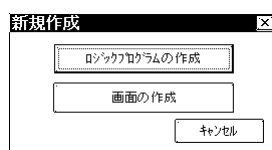


3. プロジェクトマネージャで[プロジェクト(P)]から[新規作成(N)]を選択するか、 **新規** ボタンをクリックします。ここで下記のように設定し、[OK]ボタンを押してください。



コメント	: ソフトドリンクサーバ
GP タイプ	: GLC シリーズ : GLC2300
シリアル I/F 切替	: しない
接続機器	: (株) デジタル : メモリリンク SIO タイプ <sup>*1</sup>
拡張 SIO 設定	: なし

4. ロジックプログラムまたは画面を作成するかの問い合わせがあります。[ロジックプログラムの作成]をクリックして、Editor を起動させます。



\*1 GLC を単体で使用する場合、[接続機器]には[(株) デジタル]の[メモリリンク SIO タイプ]を選択します。

## 2.2 変数の作成

Editorの動作の設定方法について説明します。また、その中で使用する変数の作成と削除、初期値の設定方法を説明します。

チュートリアルプログラムの完成品は、C:\Program Files\Pro-face\ProPBWin\Sampleのフォルダの中にSoda.prwファイルとして用意されています。

参照「Pro-Control Editor ユーザーズマニュアル 第2章 変数」

### 変数モードと固定変数モード

変数を作成する上で、Editorでは2つの動作モードがあります。本チュートリアルでは「変数モード」でロジックプログラム開発を進めます。

固定変数モードの詳細につきましては、参照「付録 固定変数モード」

### 2.2.1 変数一覧の作成

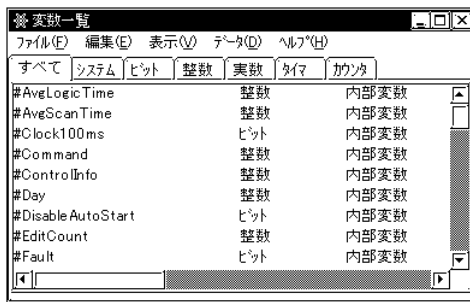
変数はロジックプログラムの作成中にいつでも追加できますが、あらかじめ設定することもできます。このチュートリアルで使用する変数一覧をここで作成しておくくと便利です。

#### 変数一覧の作成方法

メニュー内の詳細な説明は、オンラインヘルプをご参照ください。

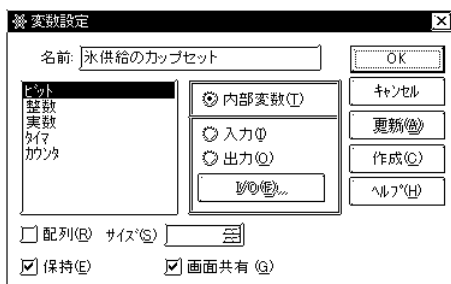
1. [データ]メニューから、[変数一覧]を選択します。

[変数一覧]ウィンドウが表示されます。



2. [編集]メニューから、[変数の追加]を選択します。

[変数設定]ダイアログボックスが表示されます。



3. 名前フィールドに変数名「氷供給のカップセット」と入力し、OKボタンを押します。変数名の制限事項については、参照「Pro-Control Editor ユーザーズマニュアル 第2章 変数」

## 2.2.2 変数タイプの指定

現在、[変数一覧]ウィンドウに変数「氷供給のカップセット」が表示されています。ここではビット、入力を指定します。変数タイプの詳細については、[参照](#)「Pro-Control Editor ユーザーズマニュアル 2.2 変数タイプ」

### 変数タイプの指定方法

1. [変数設定]ダイアログボックスから[ビット]を選択します。
2. [入力]を選択します。
3. 「保持」のチェックボックスをクリアにします。これで電断、GLC本体のコントローラリセットによるデータの保持は、非保持になります。
4. [作成]をクリックします。これで変数「氷供給のカップセット」がビット、入力として指定されました。

このとき、[変数設定]ダイアログボックスで変数「氷供給のカップセット」に対して行った変数タイプの設定が[変数一覧]上に表示されています。

[OK]をクリックした場合には、[変数設定]ダイアログボックスは閉じられます。行や命令の編集操作(挿入、ドラッグ&ドロップ、クリックなど)上メリットがありますので、[変数一覧]や[変数設定]ダイアログボックスは開いたままにしておいてください。

参考:[変数一覧]ウィンドウに表示する変数タイプを選択するには、[表示]メニューで表示する変数タイプを選択します。選択した変数タイプの横にチェックマークが表示されます。

これで変数を作成し、変数タイプの設定方法がわかりました。次に、下の表に示す変数を作成します。変数は[変数設定]ダイアログボックスで直接作成できます。

変数名	変数設定	I/Oタイプ	保持/非保持	画面共有変数
電源オンボタン	ビット	入力	非保持	非画面共有変数
氷供給のカップセット	ビット	入力	非保持	非画面共有変数
氷供給ボタン	ビット	内部変数	非保持	画面共有変数
カップ大ボタン	ビット	内部変数	非保持	画面共有変数
カップ中ボタン	ビット	内部変数	非保持	画面共有変数
カップ小ボタン	ビット	内部変数	非保持	画面共有変数
電源オフボタン	ビット	入力	非保持	非画面共有変数
氷供給	ビット	出力	非保持	非画面共有変数
ソーダバルブ	ビット	出力	非保持	非画面共有変数
ランプ	ビット	出力	非保持	非画面共有変数
ソーダ注入時間	タイマ	内部変数	保持	非画面共有変数
カップ大個数	カウンタ	内部変数	非保持	非画面共有変数
カップ中個数	カウンタ	内部変数	非保持	非画面共有変数
カップ小個数	カウンタ	内部変数	非保持	非画面共有変数

完了したら、[変数設定]ダイアログボックスを閉じます。

参考:変数名を入力ミスした場合は、[変数一覧]ウィンドウの中の編集メニューの[名前の変更]で変更できます。

[変数一覧]をウィンドウ表示中、[Insert]キーを押すと[変数設定]ダイアログが表示されますので、すばやく変数を作成できます。

## 2.2.3 変数一覧のインポート / エクスポート

変数一覧を CSV 形式のファイルとしてインポート、エクスポートすることができます。  
 変数一覧のデータが CSV 形式のファイルでエクスポートされるため、データの作成や編集が汎用表計算ソフトでも行うことができます。

### CSV ファイルのフォーマット

[変数一覧] ウィンドウの [ファイル/エクスポート] をクリックすると、変数一覧の情報が以下のように CSV 形式のファイルで出力されます。

ヘッダ情報		ドライバ情報									
//	ProductName	Pro-Control Editor									
//	FileVersion	5									
//	ProductVersion	5.00 Build (24)									
//	CompanyName	Digital Electronics Corp.									
//	LegalCopyright	Copyright(C) Digital Electronics Corp									
//	CSV FileVersion	1									
@@	Driver Type	Unit Offset	Variable	I/O Set							
GLC	0	0	1	0							
@@	Name	Data Type ID	Data Type	Array Size	I/O Type ID	I/O Type	I/O Address	I/O Offset	Attribute	comment	
1	カップ小ボタン	1	ビット		0	内部変数		0	G		
2	カップ小個数	12	カウンタ		0	内部変数		0			
3	カップ大ボタン	1	ビット		0	内部変数		0	G		
4	カップ大個数	12	カウンタ		0	内部変数		0			
5	カップ中ボタン	1	ビット		0	内部変数		0	G		
6	カップ中個数	12	カウンタ		0	内部変数		0			
7	ソーダバルブ	1	ビット		2	出力	%QX1.0.2	0			
8	ソーダ注入時間	11	タイマ		0	内部変数		0	R		
9	ランプ	1	ビット		2	出力	%QX1.0.0	0			
10	製氷機電源投入	1	ビット		0	内部変数		0			
11	電源オフボタン	1	ビット		1	入力	%IX1.0.6	0			
12	電源オンボタン	1	ビット		1	入力	%IX1.0.0	0			
13	氷供給	1	ビット		2	出力	%QX1.0.1	0			
14	氷供給のカップセット	1	ビット		1	入力	%IX1.0.2	0			
15	氷供給ボタン	1	ビット		0	内部変数		0	G		

変数情報

### ヘッダ情報

エクスポートした CSV ファイルは、Pro-Control Editor のヘッダ情報が添付されます。インポートを行う場合には、この情報はインポート先のプロジェクトには反映されません。ユーザーにて CSV ファイルの管理用情報として自由にご利用ください。

- ProductName : プロジェクト名を格納します。
- FileVersion : ファイルのバージョンを格納します。
- ProductVersion : 編集しないでください。
- CompanyName : 会社名を格納します。
- LegalCopyright : Digital Electronics Corp. (著作権)
- CSV FileVersion : 編集しないでください。



・ 変数一覧を CSV 形式のファイルで新規に作成する場合、  
 " ProductVersion " と " CSV FileVersion " の項目には何も  
 いれなくてください。



## ドライバ情報（入力必須）

GLCにオンライン接続されているユニットの情報です。

Driver Type : ドライバのタイプを以下のID番号で格納します。ドライバを未使用の場合は "0" を格納してください。

ドライバタイプ	ID番号
DIOドライバ	0
Flex Networkドライバ	1

Unit Offset : "0" を格納してください。

Variable Set : CSVファイルの変数名がインポート先の変数名と重複した場合の処理方法を以下のID番号で格納します。

処理方法	ID番号
上書き	0
追加	1



- ・ 本設定は「ファイル/オプション/確認タブ」の「ロジックエディタの操作確認あり」のチェックボックスが無効になっているとき有効になります。

I/O Set : CSVファイルのI/Oアドレスがインポート先のI/Oアドレスと重複した場合の処理方法を以下のID番号で格納します。

処理方法	ID番号
上書きか追加を随時メッセージダイアログボックスで確認する	0
上書き	1
追加	2

## 変数情報（入力必須）

I/Oに割り付けられている変数の情報です。

Name : 変数名を格納します。変数名をつける際の制限については、[参照](#)「Pro-Control Editor ユーザーズマニュアル 2.1 変数名」

Data Type ID : 変数タイプ（ビットや整数など）を以下のID番号で格納します。変数タイプの詳細については、[参照](#)「Pro-Control Editor ユーザーズマニュアル 2.2 変数タイプ」

変数タイプ	ID番号
ビット	1
整数	2
実数	3
タイマ	11
カウンタ	12

Data Type : "Data Type ID" に対応するコメントです。エクスポートした CSV ファイルにはコメントが入っていますが、新規で作成する場合などは不要です。

Array Size : 配列サイズを格納します。配列の詳細については、参照「Pro-Control Editor ユーザーズマニュアル 2.3 配列変数へのアクセス」

I/O Type ID : I/Oタイプ(入力や出力など)を以下の ID 番号で格納します。

I/Oタイプ	ID番号
内部変数	0
入力	1
出力	2

I/O Type : "I/O Type ID" に対応するコメントです。エクスポートした CSV ファイルにはコメントが入っていますが、新規で作成する場合などは不要です。

I/O Address : 以下のフォーマットで I/O アドレスを格納します。下線部の "%" と "1" は固定です。

I/O アドレスフォーマット: %AB1.C.D

A について

入出力端子について下表の ID 記号を格納します。

入出力端子	ID記号
入力端子	I
出力端子	Q

B について

ビット端子の場合は "X"、ワード端子の場合は "W" を格納します。

C について

Flex Network の場合は S-No. の番号を格納します。DIO ドライバの場合はユニット番号(0もしくは1)、ユニワイヤドライバの場合はエリア番号(0~15)になります。

D について

端子番号を格納します。

I/O Offset : "0" を格納してください。

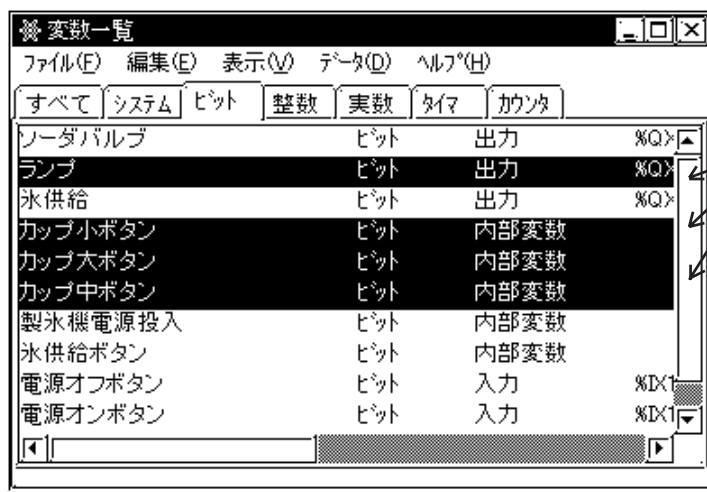
Attribute : 保持型、画面共有変数を以下の ID 記号で格納します。

変数属性	ID番号
保持型/画面共有変数	RG
保持型/非画面共有変数	R
非保持/画面共有変数	G
非保持/非画面共有変数	(空白)

Comment : コメントを格納します。

### エクスポート手順

1. エクスポートする変数を選択します。
2. [変数一覧]ウィンドウの[ファイル/エクスポート]をクリックします。



エクスポートする  
変数を選択します。



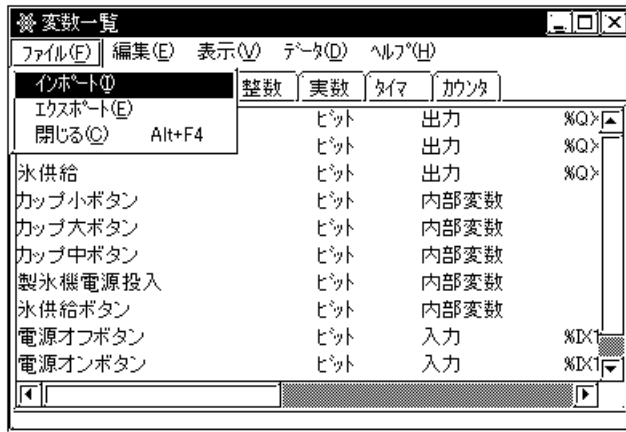
- ・ システム変数はインポート / エクスポートすることはできません。

3. CSVファイルの保存場所を指定して[保存]ボタンをクリックします。



## インポート手順

1. [変数一覧]ウィンドウの[ファイル/インポート]をクリックします。



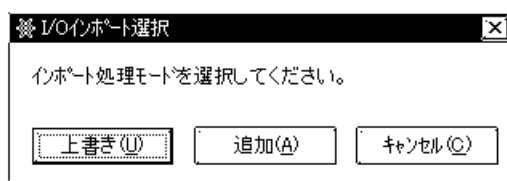
2. インポートするCSVファイルを指定して[開く]ボタンをクリックします。



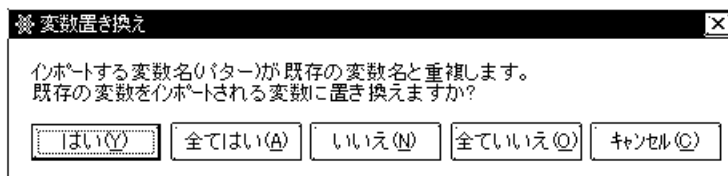
3. I/O Address の項目に何らかのアドレスが設定されている場合、以下のダイアログボックスが表示されます。CSV ファイルをどのようにインポートするかを選択します。

[上書き] : インポートするプロジェクトの変数一覧に上書きします。この場合、既存の変数一覧はすべて削除されますのでご注意ください。

[追加] : インポートするプロジェクトの変数一覧に追加します。



4. 変数が重複する場合、以下のダイアログボックスが表示されます。置き換えを行う場合には「はい」、置き換えを行わない場合には「いいえ」ボタンをクリックします。



5. エラーチェックが行われてインポート終了です。


**重要** ・ I/O設定からエクスポートされたCSVファイルを変数一覧へインポートしないでください。

## 2.2.4 プログラムの保存

作成データの安全性のため、ロジックプログラムを定期的に保存しておくことをお勧めします。

### 保存方法

ロジックプログラムエディタの[ファイル]メニューから「保存」を選択します。

参考: ツールバーのをクリックするか、またはCTRL+Sを押すことによってプログラムを保存することもできます。



- ・ ロジックプログラムを保存すると、Editorで作成した画面共有変数は自動的にシンボルエディタにGLCシンボルとして登録され、GP-PRO/PB の表示機能でも共有できるようになります。

参照 「GP-PRO/PB オペレーションマニュアル 4.7 シンボルエディタ」

## まとめ

このレッスンでは、以下の操作を学習しました。

- ・ 変数の作成と変数に関連したダイアログボックスの使い方
- ・ 変数タイプの決め方
- ・ 変数一覧のインポート / エクスポート
- ・ プログラムの保存

## 2.3 行、命令、および分岐の挿入

ロジックプログラムを作成する最初のステップは、行の挿入です。

画面には、下のような空白のプログラムが表示されます。使用するチュートリアルプログラムの完成品は、C:\Program Files\Pro-face\ProPBWin\Sampleのフォルダの中にSoda.prwファイルとして付属しています。



### 2.3.1 行の挿入

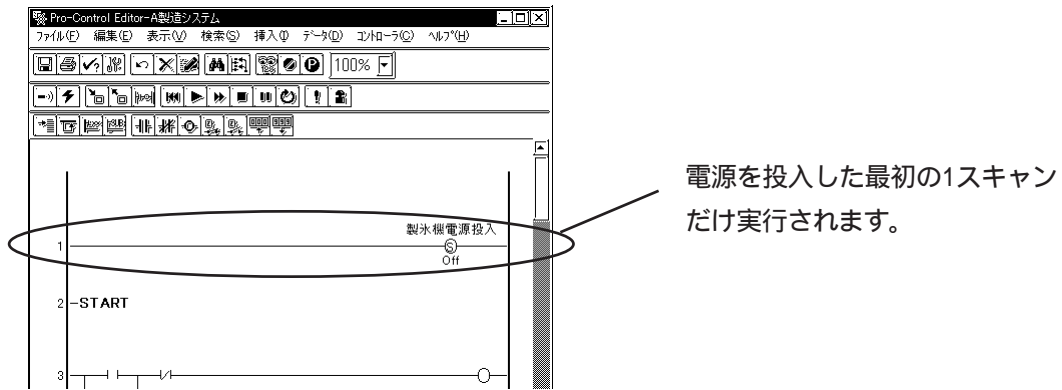
ロジックプログラムを新規作成します。

新しいプログラムの左側には、START、END、およびPEND というラベルが付いた3つの行があります。

- ・ START 行は、メインプログラム領域の開始を表します。
- ・ END 行は、メインプログラム領域の終わりを表します。
- ・ PEND 行は、全プログラム領域の終わりを表します。PEND 行の後には行を挿入できません。

#### 初期化プログラム

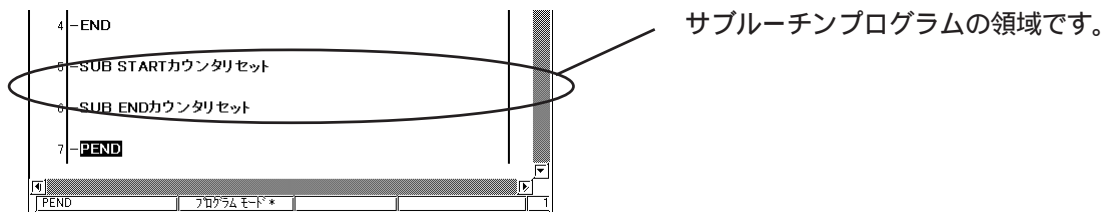
START と END の間の行は毎スキャン実行されますが、START の上に挿入される行は初期化プログラムを作成する領域です。この領域は、電源を投入した最初の1スキャンだけ実行されます。



#### サブルーチンプログラム

END と PEND の間の領域はサブルーチンプログラムのために予約されます。

START、END、およびPEND 行の詳細については、オンラインヘルプの[リファレンスガイド]を参照してください。




## 行の挿入手順

1. START の左側の行番号 1 をクリックします。  
1 行目が選択されます。
2. 右クリックします。  
ショートカットメニューが表示されます。
3. [行の挿入] を選択します。(または、「挿入」メニューより「行」を選択します。)  
新しい行が番号 2 (START 行の下) に表示されます。
4. 上の方法を繰り返して、START 行の下にさらに 5 つの行を挿入します。  
画面は、下記のようになります。

プログラム コメント

1	START
2	
3	
4	
5	
6	
7	
8	END
9	PEND

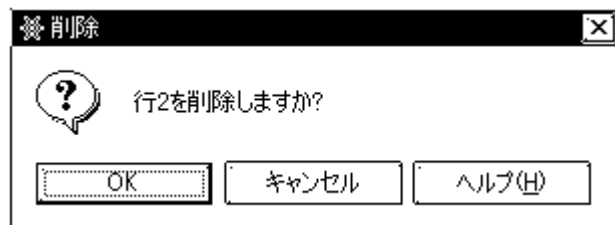


- ・ [挿入]メニューから[行]を選択するか、ツールバーの  をクリックすることでも、行を挿入することができます。

## 2.3.2 行の削除


### 行の削除手順

1. 削除する行を選択します。この例では、2行目の左の番号2(行番号)をクリックします。
2. [Delete]キーを押します。または、右クリックしショートカットメニューより[行の削除]を選択します。[削除]ダイアログボックスが表示されます。



3. [OK]をクリックします。



- ・ [編集]メニューから、[元に戻す]を選択するか、ツールバーの中の  をクリックすると、行単位に1つ前の状態に戻すことができます。



### 2.3.3 命令の挿入

ロジックプログラムに命令を挿入し、それらに変数を割り付ける方法は数とおり用意されています。このチュートリアルでロジックプログラムを作成する際に、これらの方法について説明し、実際に試してみます。

命令の詳細については、参照「Pro-Control Editor ユーザーズマニュアル 第4章 命令、オンラインヘルプ」

#### 命令を挿入する行の選択手順

- ここでは命令を2行目に挿入します。  
2行目のライン上のどこかをクリックします(ただし2という番号そのものをクリックしてはいけません)。選択された行が下のように表示されます。




- 選択した行にツールバーから命令を挿入できます。Editorのツールバーには、下に示すようなボタンがあります。これらのボタンをクリックすると選択した行に命令が挿入されます。ボタンの意味は、下表のとおりです。

	行の挿入
	分岐の挿入
	ラベル
	サブルーチン
	a接点
	b接点
	OUTコイル
	オンディレイタイマ
	オフディレイタイマ
	ダウンカウンタ
	アップカウンタ




- ・ ツールバーの命令アイコンはカスタマイズすることができます。設定方法は、[表示/命令ツールバーの設定]もしくはツールバーの アイコンをクリックします。  
参照「第2章 3. ツールバーのカスタマイズ」

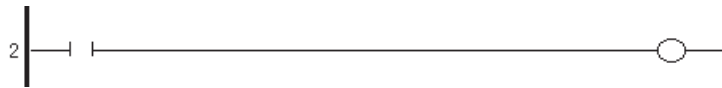
### 命令挿入手順 1: ツールバーからの挿入

1.  をクリックします。下のように入力ボックスが表示されます。




命令の上に入力ボックスが表示され、その中でカーソルがフラッシュしています。これは、オペランドです。通常では、このボックスは命令に関連付ける変数名を入力しますが、ここでは無視してください。

2.  をクリックします。2行目の右側にアウト・コイルが挿入されます。オペランドがフラッシュしていますが、ここでも無視してください。変数の入力方法はあとで説明します。参照「2.4 命令への変数の割り付け」



3. 2行目上のa接点(NO)とアウト・コイル(OUT)命令の間をクリックします。

4. b接点(NC)ボタン  をクリックします。  
2行目にb接点(NC)が挿入され、下のように表示されます。



参考: 各ツールボタンの機能の説明は、カーソルをそのボタンの上に置いたときにステータスバー上に表示されます。ツールバーは、頻繁に使用する命令を挿入するのに便利ですが、Editorで利用できるすべての命令を含んでいるわけではありません。次に説明する2つの方法を使って命令を挿入することもできます。

## 命令の挿入手順2：ダイアログボックスでのリスト選択による挿入

1. 3行目上のどこかを右クリックします。ショートカットメニューが表示されます。
2. [命令の挿入]を選択します。[命令の挿入]ダイアログボックスが表示されます。左側のボックスで選択されたカテゴリに属する命令が右側のボックスに表示されます。各命令が属するカテゴリについては、参照「Pro-Control Editor ユーザーズマニュアル」  
また、オペランドエリアで変数を作成することができます。

変数一覧に登録されている変数で、オペランドに割り付け可能な変数が表示されます。

チェックを入れると上記のプルダウンメニューにシステム変数も表示されます。



- ・ 行をダブルクリックするか、[挿入]をクリックして[命令]を選択するか、行を選択した後で[INSERT]キーを押すことで同様に[命令の挿入]ダイアログボックスが表示されます。
- ・ すべての命令は、左側で表示されている項目でカテゴリ別に分かれています。カテゴリの詳細については、参照「Pro-Control Editor ユーザーズマニュアル」
- ・ 左側のボックスで「すべて」を選択すると、右側のボックスでは全命令が一覧表示されます。

3. オンディレイタイマ (TON) を選択します。左側のボックスで「タイマ&カウンタ」を選択すると右側にオンディレイタイマが表示されます。



- ・ オペランドボタンを押すことで、オペランドエリアを表示/非表示させることができます。

4. オンディレータイマ(TON)を選択します。

[変数設定]ダイアログボックスと同様に、選択を確認する際に、[OK]または[更新]のどちらかをクリックします。ここでは、ロジックプログラムにさらに命令を挿入しますから、[命令の挿入]ダイアログボックスを開いたままにしておくために[更新]をクリックします。



5. 先ほど挿入したオンディレータイマ(TON)命令の左側の行をクリックします。

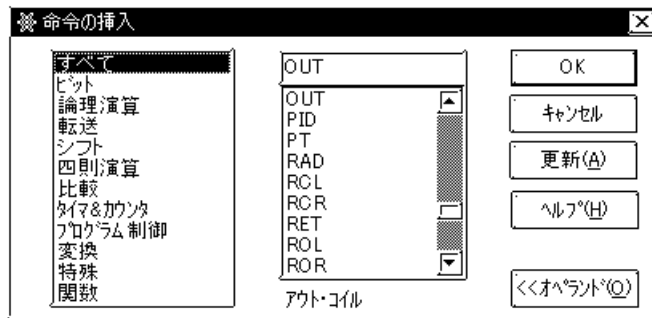
6. a接点(N0)が見つかるまで、[命令の挿入]ダイアログボックス内の命令のリストをスクロールします。

7. a接点(N0)をダブルクリックします。



## 命令挿入手順 3: ダイアログボックスでの入力による挿入

- 命令リストの上のフィールドにOUTと入力します。



- 命令リストが自動的にスクロールして、アウト・コイル(OUT)命令がリストの最上段に表示されます。また、その名前がダイアログボックスの右下隅に表示されます。
- オペランドボタンを押すことで、オペランドエリアを表示/非表示させることができます。

- オンディレータイマ(TON)命令の右側の行を選択します。

- [更新]をクリックします。

3行目が下のように表示されます。

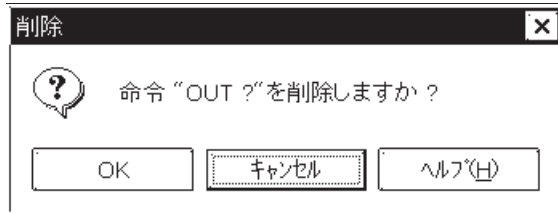


## 2.3.4 命令の削除

3行目に挿入したアウト・コイル(OUT)命令を削除します。


### 命令の削除手順

1. 3行目のアウト・コイル(OUT)命令を右クリックします。  
ショートカットメニューが表示されます。
2. [命令の削除]を選択します。  
[削除]ダイアログボックスが表示され、この命令を削除してもよいかどうかを尋ねます。



3. [OK]をクリックします。



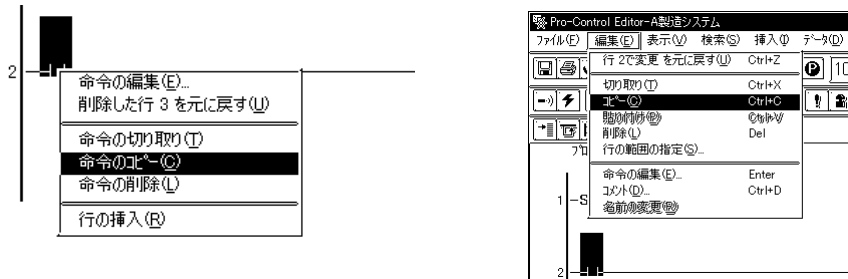
- ・ 削除する命令を選択して[Delete]キーを押すか、またはツールバーのをクリックすることでも削除できます。

## 2.3.5 命令のコピー & 貼り付け

ここでは、命令のコピー & 貼り付けを練習します。

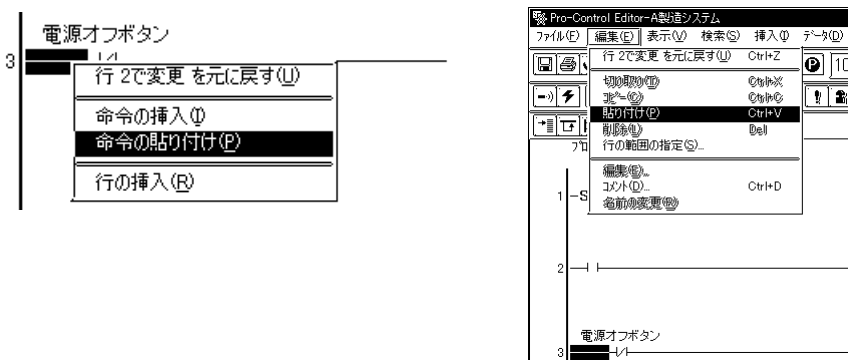
### 命令をコピーする手順

1. コピーしたい命令をクリックします。
2. 右クリックで[命令のコピー(P)]もしくは、メニューより[編集(E)] [コピー(C)]を選択します。

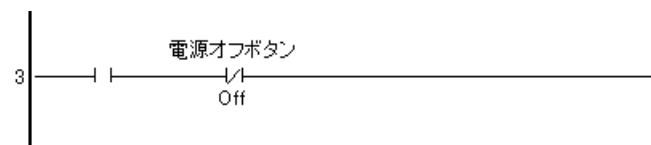


### 命令を貼り付けする手順

1. 命令を配置したい場所を選択します。
2. 右クリックで[命令のコピー(P)]もしくは、メニューより[編集(E)] [貼り付け(P)]を選択します。



3. 以下のように命令が貼り付けられます。

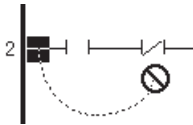


### 2.3.6 分岐の挿入

この練習では、分岐を2行目のa接点(N0)とb接点(NC)の間に挿入します。この分岐はソフトドリンクサーバのライトを自己保持するために設計されています。

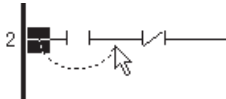
#### 分岐を挿入する手順

1. 行上の分岐を開始したいポイントにカーソルを置きます。この場合は、2行目のa接点(N0)のすぐ左です。
2. マウスをクリックし右にドラッグします。カーソルが $\odot$ に変わり、そこから破線が描かれます。

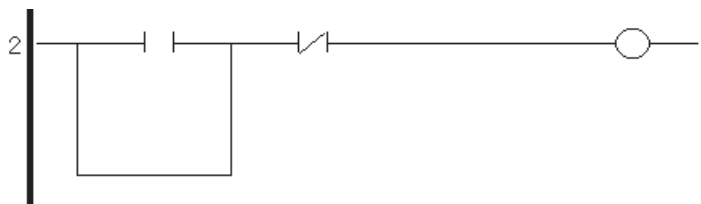


分岐の終点が無効な位置にある場合、カーソルが $\odot$ のように表示されます。分岐の終点が有効な位置にある場合、カーソルは通常に戻ります。カーソルが通常形で表示されるポイントでマウスを離すと、開始点とマウスを離した終点の間に分岐が挿入されます。カーソルが $\odot$ のように表示されているときにマウスを離すと、分岐は作成されません。

3. マウスをクリックして、カーソルがa接点(N0)とb接点(NC)の間に置かれ、 $\odot$ ではなく通常形で表示されるまで右にドラッグします。



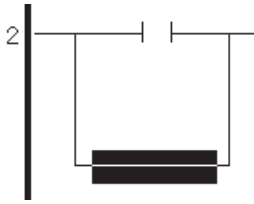
4. マウスを離します。これでa接点(N0)とb接点(NC)の間に分岐が挿入され、2行目は下のようになります。



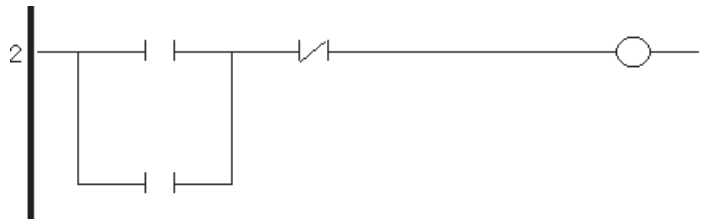


## 分岐に命令を挿入する手順

1. 分岐の下の部分をクリックして、分岐を選択します。



2. [命令の挿入]ダイアログボックスはまだ開かれていたはずですが、開いていない場合は、前述のいずれかの方法を使って、このダイアログボックスを開きます。
3. [命令の挿入]ダイアログボックスから a 接点 (NO) 命令を選択し、前述のいずれかの方法を使ってそれを挿入します。2 行目が下のように表示されます。



- ・ 命令が挿入されている分岐を削除するには、最初にその分岐の中のそれぞれの命令を選択して削除しなければなりません。

## 2.3.7 初期化ロジックプログラム

START行の上に挿入されているロジックは、初期化ロジックプログラムです。このロジックプログラムは、コントローラを運転したときに1回だけ実行されます。

### 初期化ロジック挿入手順

1. START行の上にある[プログラムコメント]フィールドをクリックします。  
このフィールドが表示されていない場合は、[表示]メニューから[コメント]を選択してから、[プログラム]を選択します。
2. ショートカットメニューから[行の挿入]を選択します。  
行がSTART行の上に挿入されます。  
参考：以降の行は1つずつ繰り下がります(2番目にあった行が3行目になります)。
3. 初期化行(1行目)を右クリックします。
4. ショートカットメニューから[命令の挿入]を選択します。
5. [命令の挿入] ダイアログボックスからSET 命令を選択し、[OK]をクリックします。



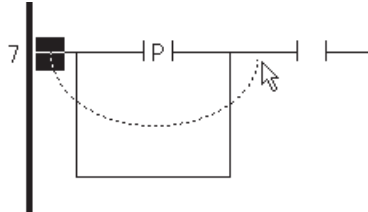
この行は、ソフトドリンクサーバの製氷機の電源をオンにするために使用します。これは一度オンに設定しておけば、ソフトドリンクサーバの電源がオンになっている間、オンのままになっています。

参考:[オプション]ダイアログボックスで[行または命令をフォーカスの後に追加]を選択していない場合、行などのオブジェクトは選択したライン上に挿入されます。初期化行を挿入するためにはSTART行を選択してください。

これでロジックプログラムの3行目、4行目、および初期化ロジックプログラムの1つの行が完成しました。次のページに示すように5行目～7行目を作成してください。ロジックプログラム中の | P | 命令とは、PT(立ち上がり接点)命令のことです。

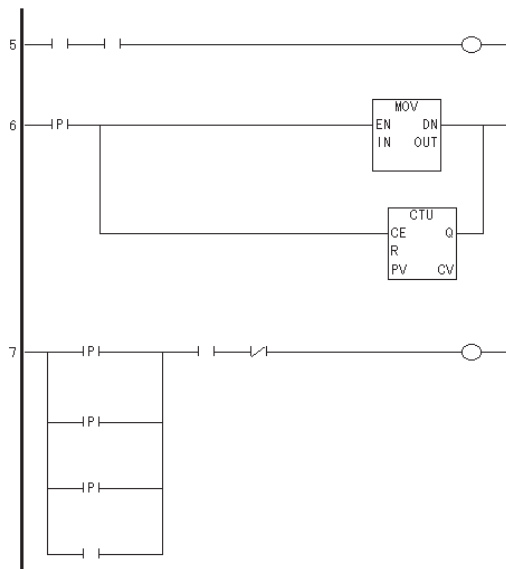
## 7行目の複数の分岐を挿入する手順

1. 最初の分岐を前述の方法で挿入します。
2. 2番目の分岐を挿入するために、前の分岐と同じポイントをクリックし、そこからドラッグします。
3. カーソルを、前の分岐を越えて、新しい分岐を挿入する行のポイントまでドラッグします。



マウスを離すと、新しい分岐が前の分岐の上に挿入されます。

下の例では、命令が5行目～7行目に挿入されています。



## まとめ

このレッスンでは、以下の操作を学習しました。

- ・ 行の挿入と削除
- ・ 命令の挿入と削除
- ・ 分岐の挿入と削除

## 2.4 命令への変数の割り付け

命令に変数を割り付ける方法を学びます。

「2.2 変数の作成」で、チュートリアルロジックプログラムで使用する変数一覧を作成しました。ここで[変数一覧]ウィンドウを開きます。

[変数一覧]ウィンドウのオープン手順

1. [データ]メニューから[変数一覧]を選択します。
2. このウィンドウを画面の左下隅に移動します。[命令の挿入]ダイアログボックスが開かれている場合は、キャンセルをクリックして閉じます。

### 2.4.1 オペランド

前のレッスンで、最初に命令を行に挿入したとき、カーソルがフラッシュしているフィールドが表示されたことを思い出してください。このフィールドがオペランドです。このボックスに、命令に関連付ける変数を入力します。

基本オペランドへのアクセス手順

1. 3行目のアウト・コイル(OUT 命令)をダブルクリックします。命令の上にテキストフィールドが表示され、その中のカーソルがフラッシュします。これがこの命令のオペランドです。

参考: オペランドは、命令をクリックして[Enter]を押すか、または命令を右クリックして、ショートカットメニューから[命令の編集]を選択することによってもアクセスすることができます。

応用命令には、2つ以上のオペランドがあります。たとえば、オンディレータイマ(TON 命令)には、2つのオペランドがあります。1つのオペランドで変数を割り付け、もう1つのオペランドでは、設定時間(ミリ秒単位)を入力します。

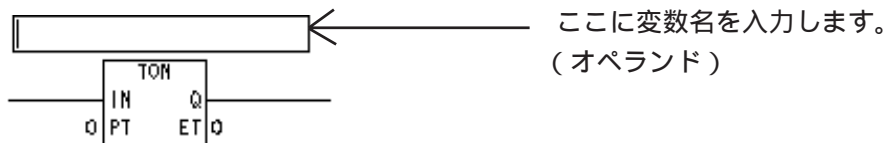
応用オペランドへのアクセス手順

1. 4行目のオンディレータイマ(TON 命令)をクリックします。命令が下のように表示されます。



オンディレータイマ(TON 命令)の上部の領域が黒く反転表示されています。この領域に、命令に割り付ける変数を入力します。下部の黒く反転表示されている領域は、設定時間(PT)の専用変数です。この領域には、設定時間(ミリ秒単位)を入力します。

2. オンディレータイマ(TON 命令)の上の黒く反転表示されている領域をダブルクリックします。これでオペランドが選択されました。このとき、命令にタイマ変数(変数名)を割り当てることができます。



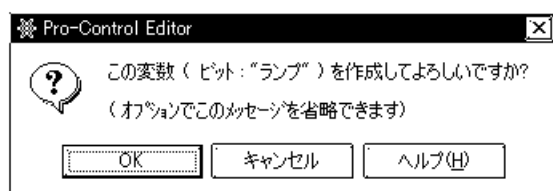
3. オンディレータイマ(TON命令)のPTのすぐ左にある領域をダブルクリックします。[現在値変更]ダイアログボックスが表示されます。このウィンドウには、タイマ出力(Q)がオンになるまでの設定時間(ミリ秒単位)を入力します。命令に変数を割り付ける方法については、次の項で説明します。
4. [現在値変更]ダイアログボックスを閉じます。

## 2.4.2 変数の入力

オペランドに変数を入力する1つの方法として、オペランドの中に直接入力することができます。

オペランドへのテキスト入力手順

1. 命令をダブルクリックして、3行目のアウト・コイル(OUT命令)のオペランドを選択します。
2. このボックスに「ランプ」と入力します。
3. [Enter]キーを押します。下のようなウィンドウが表示され、変数の作成を行うか確認します。



4. [OK]をクリックします。[変数一覧]ウィンドウのリストの中に「ランプ」という変数が表示されています。Editorは、命令に必要な変数タイプを自動的に変数に割り付けます。この場合、変数タイプとして内部変数のビットが割り付けられています。

参考: Editorは、命令に対して作成された新しい変数に必要な変数タイプを自動的に割り付けます。また、変数一覧にすでにある変数をオペランドの中に直接入力することもできます。入力が完了すると、自動的に変数が割り付けられます。

上記の方法を使って、変数「電源オフボタン」を3行目のb接点(NC命令)に割り付けます。

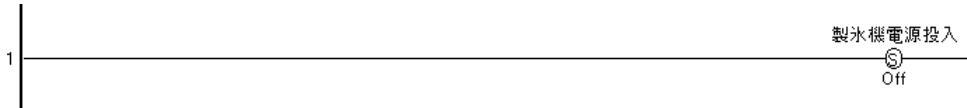
参考: コイル命令(OUT、SET、RST、NEG)に割り付けた変数を保持型に変えると自動的に保持型のコイル命令(M、SM、RM、NM)に変わります。

3行目は、下のように表示されます。



初期化行のセット・コイルに変数「製氷機電源投入」を割り付けます。この変数は、オペランドに直接入力することによって作成できます。

初期化行が下のように表示されます。

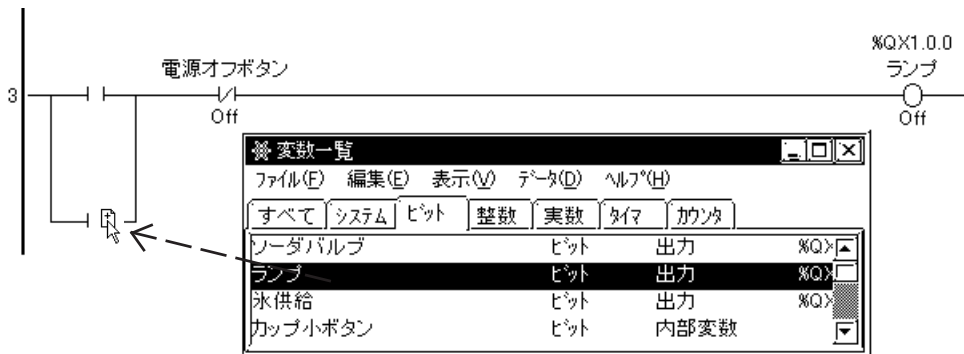



命令に変数を割り付けるもう1つの方法として、[変数一覧]ウィンドウの中の変数を命令にドラッグ&ドロップすることもできます。この方法は、多数の命令に同じ変数を割り付ける場合に便利です。この方法を使う利点は、「2.9 1/0の割り付け」で詳細を説明します。

#### [変数一覧]ウィンドウによるマッピング手順

1. [変数一覧]ウィンドウの「ランプ」をクリックし、そのままマウスボタンを押したままにします。
2. マウスボタンを押したまま、「ランプ」を3行目の分岐の上のa接点(N0命令)までドラッグします。分岐を挿入する場合と同様に、カーソルは最初は⓪になります。カーソルがこの状態のとき、命令に変数を割り付けることはできません。
3. カーソルを3行目の分岐の上のa接点(N0命令)の上にドラッグします。



カーソルは、下のように表示されます。



カーソルが  のときにマウスを離すと、変数が割り付けられます。これで変数「ランプ」が3行目の分岐のa接点(N0命令)に割り付けられました。

4. 変数「電源オンボタン」をクリックし、3行目上の他のa接点(N0命令)にドラッグします。  
3行目は、下のように表示されます。



[参考]カーソルを命令のごく近くまで移動すると、カーソルは自動的に  から  に変わります。この時点でマウスを離すと、変数とその命令に割り付けられます。定数を変数として割り付けることもできます。入力方法は、通常の変数の場合と全く同じですが、ウィンドウからドラッグできないため、手操作で入力する必要があることです。

### 2.4.3 作業の完了

これで、命令に変数を割り付ける方法がわかりました。プログラムの残りの行を完成させてください。完成した行は、次ページの図のようになります。

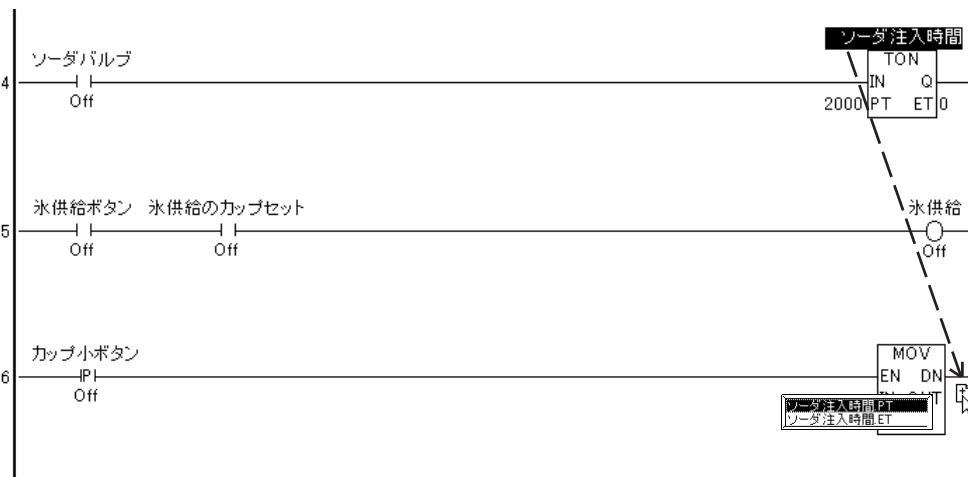
6行目のMOV命令と7行目のb接点(NC命令)には、それぞれ変数「ソーダ注入時間.PT」と「ソーダ注入時間.Q」が割り付けられています。これらの変数は、「ソーダ注入時間」変数が割り付けられているタイマのPTおよびQを参照します。

これらの変数を入力する手順には、以下の3つがあります。

- ・ オペランドを選択し、その中に変数「ソーダ注入時間」を直接入力する。
- ・ [変数一覧]ウィンドウから「ソーダ注入時間」変数をクリックしドラッグして入力する。
- ・ オペランドをコピーしたい命令にドラッグし、専用変数一覧から選択して入力する。この方法について以下に示します。

#### 操作手順

1. コピー元になるオペランドを選択します。
2. カウンタ・タイマの変数をコピーしたい命令にドラッグします。
3. 専用変数一覧ボックスより要素を選択し、ダブルクリックします。



**[重要]**

6行目、7行目などで使用される。「ソーダ注入時間.PT」や「ソーダ注入時間.Q」のように応用命令の専用変数は、変数名+拡張子の形で使用できます。

命令によりますが、拡張子として、

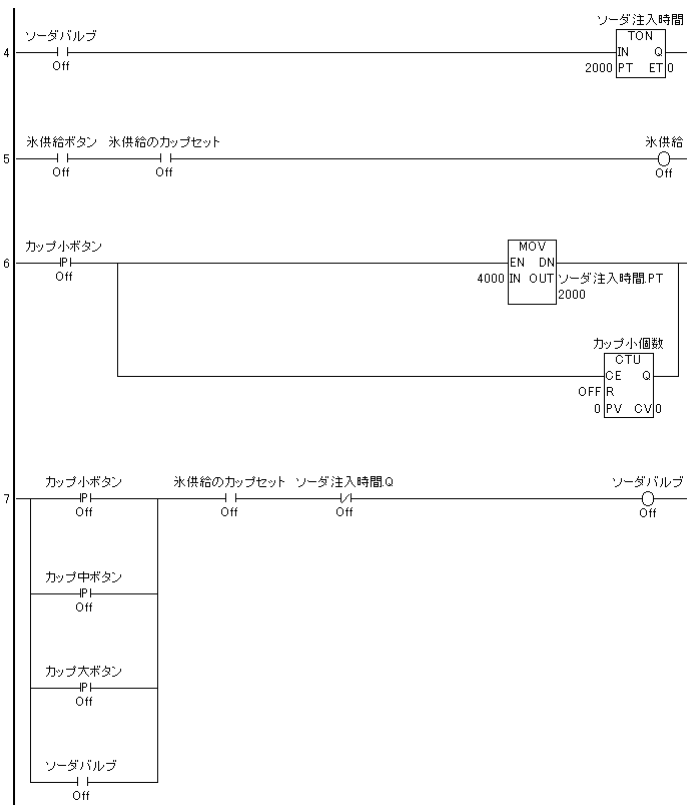
- \*\*\*.CV 現在値
- \*\*\*.PT 設定値
- \*\*\*.Q 出力ビット
- \*\*\*.R リセットビット

などがプログラム上で使用できます。

参照 「Pro-Control Editor ユーザーズマニュアル 2.2 変数タイプ」

<チュートリアルプログラムの作成例>

ここまでのレッスンで以下のロジックプログラムが作成されます。



**まとめ**

このレッスンでは、以下の操作を学習しました。

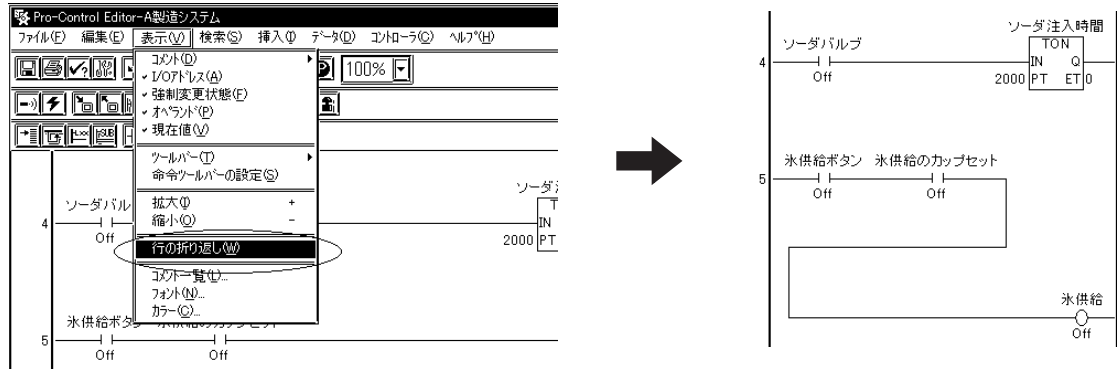
- ・命令に変数を割り付ける



### [ロジックプログラムの非表示部分について]

行が長くなりすぎて右端部分が画面内に収まらない場合、画面内で収まるように行を折り返して表示させることができます。

[表示/行の折り返し]をクリックすると、下図のように画面内に収まらない行が折り返して表示されます。



**重要** ・ 折り返し（左向き）の行や縦線には、命令や分岐の挿入はできません。



**MEMO** ・ ロジックプログラムを印刷する場合も範囲内で収まるように折り返して印刷されます。

## 2.5 ロジックプログラムのドキュメント化

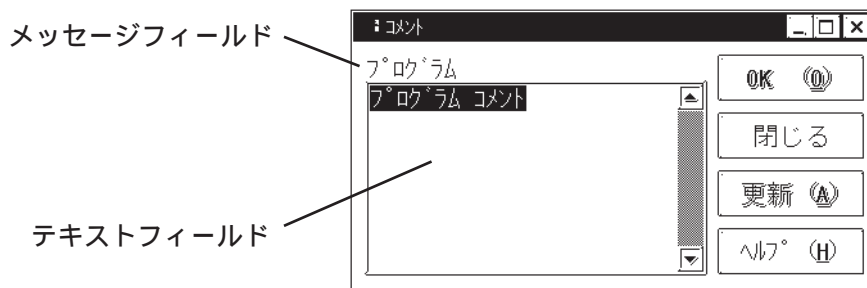
Editorでは、1. プログラム全体、2. 各行、3. 各変数にコメントを付加することができます。ロジックプログラムにコメントを付加しておく、再読性が上がりデバッグや追加修正の際に有効です。

### 2.5.1 プログラムコメントの追加

ロジックプログラムに追加する最初のコメントは、プログラムの機能のコメントです。

#### プログラムコメントの追加手順

1. 画面最上段の[プログラムコメント]をダブルクリックします。  
画面に[コメント]ダイアログボックスが表示されます。



このウィンドウでプログラムのコメントを入力してください。

参考:[コメント]ダイアログボックスのメッセージフィールドの上に、[プログラム]と表示されています。これは、このテキストフィールドがプログラムのコメントの入力ボックスであることを示しています。

2. [プログラムコメント]テキストを選択します。
3. 「このプログラムは、ファーストフードレストランのソフトドリンクサーバを制御します。」と入力します。
4. [OK]をクリックします。プログラムコメントがロジックプログラムの最上段に表示されます。表示されない場合は、画面を上スクロールしてください。

このプログラムは、ファーストフードレストランのソフトドリンクサーバを制御します。




参考:下側のステータスバーをダブルクリックしても、[プログラムコメント]の追加または編集ができます。

## 2.5.2 行コメントの追加

Editorでは、プログラムの各行にコメントを追加できます。下の例では、5行目にコメントを追加します。

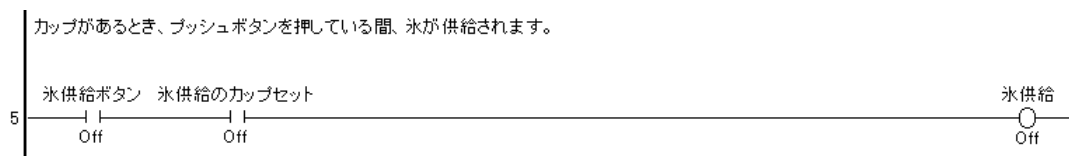
### 行コメントの追加手順

1. 5行目の左にある番号5を右クリックします。
2. ショートカットメニューから[コメント]を選択します。[コメント]ダイアログボックスが開きます。このウィンドウは、プログラムコメントを挿入したときにオープンしたウィンドウと同じです。唯一の違いは、メッセージフィールドの上のコメントが[プログラム]ではなく5行目になっていることです。

参考：編集メニューから[コメント]を選択するか、またはツールバーのをクリックすることでも、[コメント]ダイアログボックスをオープンできます。

プログラムの5行目は、アイスディスペンサーの制御のために使用されます。

3. [コメント]ダイアログボックスのテキストフィールドをクリックします。
4. 「カップがあるとき、プッシュボタンを押している間、氷が供給されます。」と入力します。
5. [更新]をクリックします。



[コメント]ダイアログボックスを開いておくと、プログラムの残りの行にコメントを追加するのが簡単になります。

### 3行目へのコメントの追加手順

1. 3行目上のオペランド以外の場所をクリックします。  
[コメント]ダイアログボックスの最上段のコメントが3行目が変わります。
2. テキストフィールドをクリックします。
3. 「電源オフボタンを押すまで、ライトは点灯したままになります」と入力します。
4. [更新]をクリックします。このチュートリアルでは、3行目および5行目だけを例にあげドキュメント化しました。

### 2.5.3 変数へのコメントの追加

ロジックプログラムの各変数にコメントを追加できます。ラベルまたは定数にはコメントを追加できません。

#### 変数コメントの追加手順

1. 本チュートリアルの流れでは、[変数一覧]ウィンドウが開かれているはずですが。開いていない場合は、[データ]メニューから[変数一覧]を選択して開きます。
2. [コメント]ダイアログボックスも開かれているはずですが。開いていない場合は、[編集]メニューから[コメント]を選択して開きます。
3. 変数「ソーダ注入時間」を表示しているオペランドをクリックします。  
[コメント]ダイアログボックスのメッセージフィールド上に「ソーダ注入時間」というメッセージが表示され、[変数一覧]ウィンドウでも「ソーダ注入時間」が反転表示されます。
4. [コメント]ダイアログボックスのテキストフィールドをクリックします。
5. 「ソーダ注入時間は、ソーダバルブを開けておく時間を決定します。動作時間は設定値に依存します。」と入力します。
6. [更新]をクリックします。

参考:変数にコメントを追加する方法として、変数をロジックプログラムから選択する方法のほかに、[変数一覧]ウィンドウから選択する方法もあります。

#### 変数一覧からコメントを追加する手順

この例では、電源オンボタンにコメントを追加します。

1. [変数一覧]ウィンドウの中の変数「電源オンボタン」をクリックします。  
[コメント]ダイアログボックスのメッセージフィールドに「電源オンボタン」というメッセージが表示されます。
2. [コメント]ダイアログボックスのテキストフィールドをクリックします。
3. 「電源オンボタンを押すと、ソフトドリンクサーバが起動します。」と入力します。
4. [更新]をクリックします。

このチュートリアルでは、「ソーダ注入時間」変数と「電源オンボタン」変数のコメントだけを例にあげ追加しました。他のすべての変数のコメントも、同じ方法で作成できます。

## 2.5.4 [コメント一覧]ウィンドウ

[コメント一覧]ウィンドウは、プログラム中のすべての変数および行のコメントを一覧表示します。

[コメント一覧]ウィンドウを表示する方法

[表示]メニューから、[コメント一覧]を選択します。

[コメント一覧]ウィンドウから詳細なコメントを表示する方法

[コメント一覧]ウィンドウの中の「ソーダ注入時間」変数をダブルクリックします。

[コメント]ダイアログボックスに、「ソーダ注入時間」変数の詳細なコメントが表示されます。

[変数一覧]、[コメント]、[コメント一覧]の表示内容はプログラム中の行や変数を選択するのに追従し選択表示されますが、[変数一覧]、[コメント]、[コメント一覧]表示内容を選択してもロジックプログラムの上は追従しません。

Editorでは、検索機能により特定の変数を簡単に見つけることができます。これは、「2.8 ロジックプログラム内の移動」で説明します。

### まとめ

このレッスンでは、以下の操作を学習しました。

- ・プログラム、行、および変数にコメントを追加する
- ・[コメント一覧]ウィンドウを表示する

## 2.6 行のコピー、切り取りおよび貼り付け

ロジックプログラムでは、いくつかの行で同じ命令シーケンスを入力しなければならないことがあります。すでに作成した行をコピーして貼り付けることによって、工数を削減できます。

### 2.6.1 行のコピー

以下のチュートリアルでは、2つの行が5行目と7行目の間に追加されます。これらの追加の行には、6行目と同じ命令があり、異なる変数が割り付けられています。

#### 行のコピー手順

1. 行の左の番号6をクリックして、6行目全体を選択します。
2. [編集]メニューから、[コピー]を選択します。

参考：切り取りまたはコピーする行の範囲を選択する方法。

選択する範囲の中の先頭行番号をクリックします。次に、[Shift]キーを押したまま、選択する最後の行番号を選択します。こうすると2つの行の間にあるすべての行が選択され、切り取りまたはコピーすることができます。一度の操作でコピーできる行の数は、25行程度が目安です(行内の命令数によって前後します)。

### 2.6.2 行の貼り付け

行全体が選択されている場合を除いて、Editorは、コピーまたは切り取りされた行を、現在選択されている行の下に貼り付けます。

[オプション]ダイアログボックスで[行または命令をフォーカスの後に追加]が選択されていない場合、コピーされた行は、現在の行の上に挿入されます。

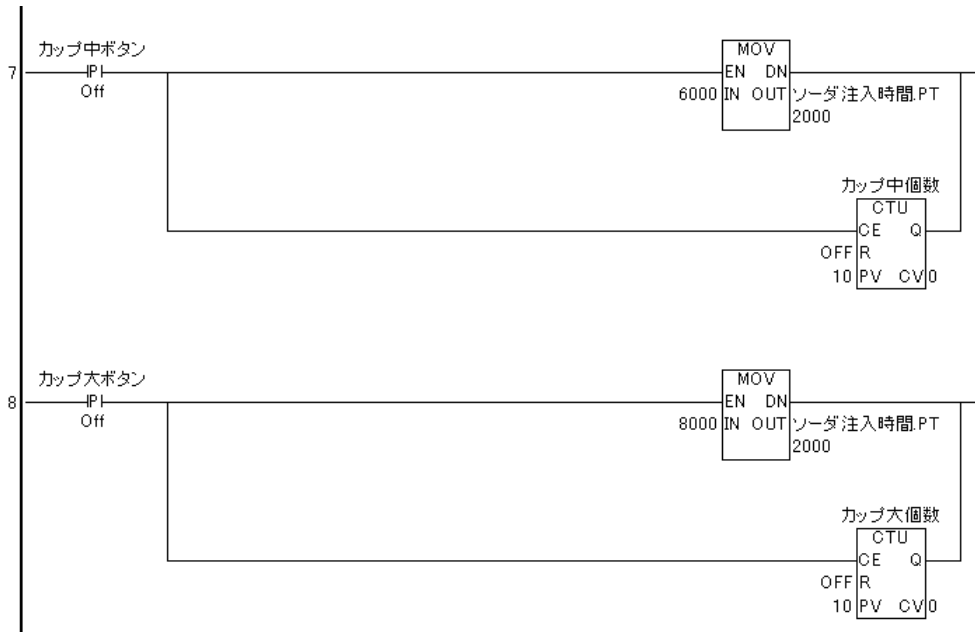
#### 【重要】

切り取り、コピーされた行は、一旦内部のクリップボードにロードされます。クリップボードから行を貼り付けるとき、行全体を選択している場合、Editorは、全行をクリップボード上の行に置換します。

#### 行の貼り付け手順

1. 6行目上のどこかをクリックします。
2. 編集メニューから[貼り付け]を選択します。6行目と7行目が同じになります。
3. 6行目上のどこかをクリックします。
4. 編集メニューから、[貼り付け]を選択します。6行目～8行目がすべて同じように表示されます。

参考:行を貼り付けしたとき、その行に関連付けられているすべての変数およびコメントもそのまま貼り付けられます。目的のプログラムに合わせて行を編集してください。このチュートリアルでは、7行目および8行目の変数を、下図のように変更してください。



5. 上の例のように行上のPT命令の変数名を変更します。

### 2.6.3 [切り取り]コマンドの使用

Editorでは、プログラムから行または行の一部を切り取り、それを他の部分へ移動できます。これは、[切り取り]コマンドを使って実行します。以下の練習では、4行目をプログラムの最後の行へ移動します。

#### [切り取り]コマンドの使用手順

1. 4行目の行番号をクリックします。4行目全体が選択されます。
2. 編集メニューから、「切り取り」を選択します。  
4行目全体が、ロジックプログラムから切り取られ、クリップボードに入れられます。
3. 8行目上のどこかをクリックします。
4. 編集メニューから、[貼り付け]を選択します。4行目が8行目の下に移動されます。  
プログラムの終わりは、下の図のようになります。



参考: 行全体をプログラムの他の部分に移動するには、最初に行を選択し、その中央部をドラッグして、新しい位置まで移動できます。

## まとめ

このレッスンでは、以下の操作を学習しました。

- ・行をコピー、切り取りおよび貼り付ける

## 2.7 サブルーチンおよびラベル

JSR(ジャンプサブルーチン)またはJMP(ジャンプ)命令が挿入されると、コントローラは、そのサブルーチンまたはラベルまでジャンプして実行します。サブルーチンとラベルの主な違いとして、JSR命令では、ENDとPENDの間に作成された同一名サブルーチンを実行したあと、ロジックプログラムの中のJSR命令の次の位置に戻りますが、JMP命令ではJMP命令と同名のラベルにジャンプしロジックプログラムの実行を続け、そのままでは元のJMP命令の位置には戻りません。

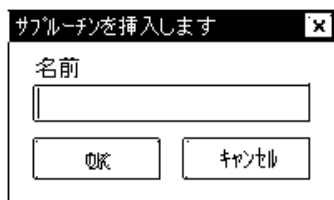
JMP命令およびJSR命令の詳細については、[参照](#)「Pro-Control Editor ユーザーズマニュアル 4.2.53 JMP (ジャンプ) /4.2.54 JSR (ジャンプサブルーチン)」

### 2.7.1 サブルーチンの挿入

すべてのロジックプログラムの終わりには、ENDおよびPENDというラベルが付いています。ENDラベルは、メインプログラム領域の終わりを知らせます。Editorは、スキャンごとにSTART～ENDまでの命令を実行します。ENDラベルとPEND(プログラムの終わり)ラベルの間の領域はサブルーチンのために予約されています。以下のチュートリアルで、サブルーチンを追加します。

#### サブルーチンの追加手順

1. ENDラベルをクリックします。
2. 挿入メニューから、サブルーチンを選択します。  
[サブルーチンを挿入します]ダイアログボックスが表示されます。



3. [サブルーチンを挿入します]ダイアログボックスの名前フィールドに、「カウンタのリセット」と入力します。サブルーチン名/ラベルは、最大半角32文字、全角16文字以内で設定されます。サブルーチン名/ラベルを数字で始めないでください。サブルーチン名は、“\_”以外の記号は使用できません。大文字と小文字、および全角と半角は区別しません。  
[参照](#)「2.2.1 変数一覧の作成」
4. [OK]をクリックします。ENDとPENDの間に、挿入されたサブルーチンが表示されます。

```

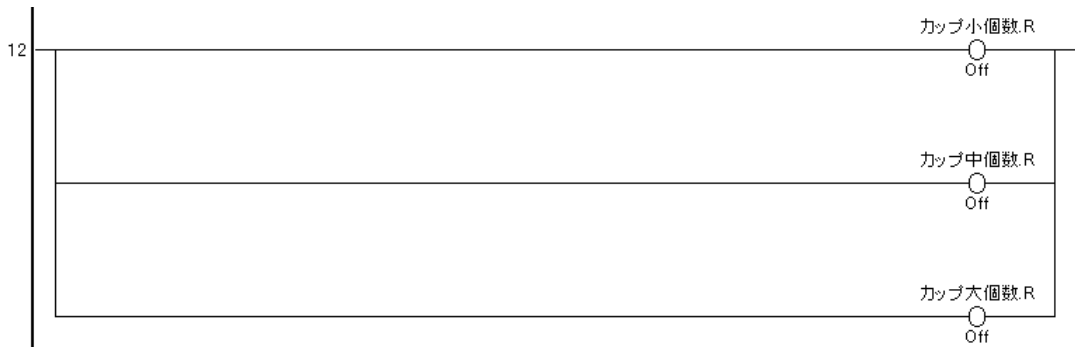
10 |—END
11 |—SUB STARTカウンタのリセット
12 |—SUB ENDカウンタのリセット
13 |—PEND

```

SUB STARTカウンタのリセットおよびSUB ENDカウンタのリセットというラベルが付いた2つの新しい行が、END行とPEND行の間に挿入されました。サブルーチンを挿入する場所は、これらの2つの行の間です。



5. SUB STARTカウンタのリセットラベルを右クリックします。
6. ショートカットメニューから[行の挿入]を選択します。行がSUB START行とSUB END行の間に挿入されます。
7. SUB START行とSUB END行の間にある行を右クリックします。
8. その行にアウト・コイル(OUT命令)を挿入します。
9. アウト・コイル(OUT命令)の両側に2つの分岐を挿入します。
10. それぞれの分岐にアウト・コイル(OUT命令)を挿入します。下の図は完成したサブルーチンです。  
これは、GLCに電源を入れるたびに各カウンタをリセットするために使用されます。



各アウト・コイル(OUT命令)に変数が割り付けられます。例のようにこれらの変数をここで割り付けてください。

以上でサブルーチンが完成しました。ロジックプログラムに2つ以上のサブルーチンを追加するには、SUB END行またはEND行を選択し、手順2)～6)を繰り返します。

サブルーチンをロジックプログラムの中の特定のポイントで実行する場合、JSR命令を挿入しなければなりません。

次にこの操作を練習します。このサブルーチンは、3行目の「ランプ」出力コイルがオンになったときに実行させます。したがって、この例では、JSR命令を4行目に置かなければなりません。

#### JSR命令の挿入手順

1. 3行目を選択します。
2. 挿入メニューから、[行]を選択します。
3. 4行目にPT命令を挿入します。
4. 変数「ランプ」をPT命令に割り付けます。
5. [命令の挿入]ダイアログボックスを使って、JSR命令をPT命令の右に挿入します。
6. JSR命令のオペランドの中に、サブルーチンの名前「カウンタのリセット」を入力します。行は下のように表示されます。



JSR命令「カウンタのリセット」が検出されたとき、サブルーチン「カウンタのリセット」にジャンプします。サブルーチンの実行が完了した後、5行目から実行を再開します。

参考：サブルーチンを削除するには、はじめにそのサブルーチンの中の個々の行を削除し、SUB START行を削除してください。(SUB END行は、SUB START行と一緒に削除されます。)

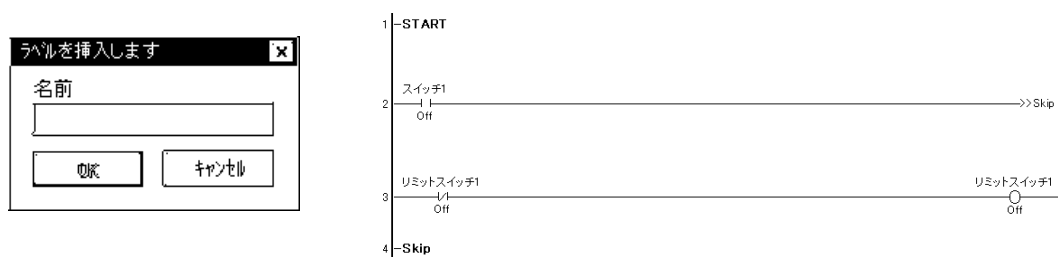
## 2.7.2 ラベルの挿入

JMP(ジャンプ)命令とジャンプ先のラベルをロジックプログラムの任意の部分に挿入できます。コントローラがJMP命令を実行したとき、JMP命令は、それに割り付けられている同名のラベルにジャンプし、そこからプログラムを実行します。

[オプション]ダイアログボックスの中で[行または命令をフォーカスの後に追加]が選択されているかどうかによって、ラベルは選択されている行の上または下に挿入されます。このチュートリアルではラベルを使用しませんが、ラベルを挿入するには、以下の手順でラベルは挿入できます。

### ラベルをロジックプログラムに割り付ける手順

1. 行上のどこかをクリックします。
2. [挿入]メニューから、[ラベル]を選択します。[ラベルを挿入します]ダイアログボックスが表示され、ラベルの名前の入力を要求してきます。



これは、ロジックプログラム中のJMP命令に割り付けられる変数と同名にしてください。JMP命令は、同名のラベルにジャンプを行います。

### JMP命令の挿入手順

1. ジャンプを開始する行の最後の命令の右側をクリックし、[挿入]メニューから[命令]を選択します。
2. [命令]ダイアログボックスのJMP命令をダブルクリックします。JMP命令は必ず行の最後の命令として挿入されます。Editorは、プログラムの中でこの命令を検出したとき、指定された同名のラベルにジャンプします。

## まとめ

このレッスンでは、以下の操作を学習しました。

- ・サブルーチンおよびラベルを作成する
- ・JMP(ジャンプ)命令およびJSR(ジャンプサブルーチン)命令を挿入する。

## 2.8 ロジックプログラム内の移動

ここではEditorで該当の行や変数をすばやく検索する方法を説明します。Editorには、検索のために、[検索]、[リファレンス]、[ブックマーク]、[指定行へ移動]、および[指定ラベルへ移動]などのコマンドがあります。


### 2.8.1 [検索]コマンド

[検索]コマンドは、ロジックの中の特定のテキストを検索します。

[検索]コマンドの使用手順

1. 開かれているウィンドウがあれば、ここでそれを閉じます。
2. [検索]メニューから、[検索]を選択します。[検索]ダイアログボックスが表示されます。



参考:[検索]ダイアログボックスは、ツールバーの中のをクリックすることによってオープンすることもできます。

検索に使用する一致パターンのタイプの指定ができます。

- Fill という語を探す場合、Editorは、fill という語を含むすべての語を検出します(小文字の fill や他の語の一部になっている場合、たとえば Fillet も検出されます)。
- [大小文字の区別]を選択した場合、Fill を検出しますが、fill を検出しません。また、[単語単位で探す]を選択した場合、Fill を検出しますが、Fillet を検出しません。

検索の範囲および方向の指定

- [選択範囲のみ]を選択している場合、検索の範囲は、プログラムの中の反転表示されている部分だけです。
  - [すべての範囲]を選択すると、検索の範囲は、プログラム全体を含みます。[すべての範囲]を選択することによって、選択範囲の最上段から検索を開始できます。また、[カーソルの位置から]を選択することによって、指定した位置から検索を開始できます。この例では、検索をプログラムの最初から開始します。
3. プログラムの中のSTARTラベルを選択します。
  4. [検索]ダイアログボックスの[検索する文字列]フィールドをクリックします。
  5. 「ソーダ」と入力します。

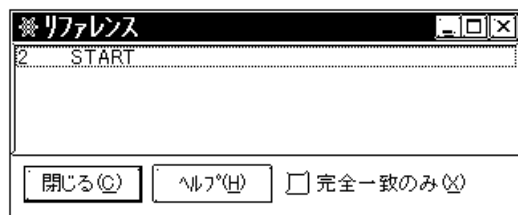
6. [すべての範囲]、[下へ]、または[カーソルの位置から]を選択します。
7. [検索]ボタンをクリックします。フォーカスは、検出した最初の一致、すなわち「ソーダ注入時間」変数の最初の部分に移ります。
8. [検索]ボタンをもう一度をクリックします。フォーカスは検出した2番目に一致する「ソーダ注入時間」に移動します。一致するエントリがなくなった時、ビープ音が鳴ります。  
参考：最初の[検索]コマンドの実行の後、[検索]メニューから[次を検索]を選択すれば、同じ条件で検索を続け、次に一致する文字列を検出できます。

## 2.8.2 [リファレンス]コマンド

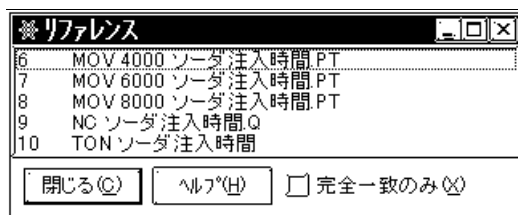
[リファレンス]コマンドは、ロジックプログラムの中の指定した変数が複数以上使用されている場合、使用されているすべての位置を検出します。このコマンドは、行番号とその変数が含まれる命令をリストします。このチュートリアルでは、STARTラベルを選択します。[リファレンス]コマンドは、プログラムの中のどこからでも実行できます。

### [リファレンス]コマンドの使用手順

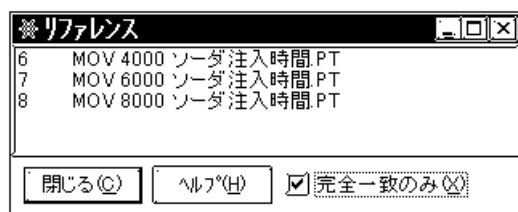
1. STARTラベルをクリックします。
2. [検索]メニューから、[リファレンス]を選択します。[リファレンス]ダイアログボックスが表示されます。



3. [リファレンス]ダイアログボックスのサイズを変更し、画面の右下隅に移動します。
4. 6行目の「ソーダ注入時間.PT」をクリックします。下のように[リファレンス]ダイアログボックスが表示されます。



5. [完全一致のみ]を選択します。[リファレンス]ダイアログボックスは下のように表示されます。





3. [変数一覧]ウィンドウの中の変数「ソーダ注入時間」をクリックします。

参考:[コメント一覧]および[リファレンス]ダイアログボックスの表示は、選択に従って変化します。[リファレンス]ダイアログボックスには、変数「ソーダ注入時間」に含まれるすべての項目が表示されています。また、これらのウィンドウの表示が変化しても、ロジックプログラムの表示位置は変化しません。[リファレンス]ダイアログボックスで任意の変数行を選択すると、ロジックプログラムの中の変数に対応するポイントが表示されます。

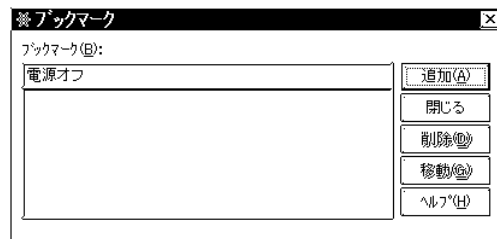
4. [リファレンス]ダイアログボックスの最初の行をクリックします。ロジックプログラムは、行上でユーザーが指定した命令および変数を反転表示しています。

## 2.8.4 ブックマークの使用

ロジックプログラムの中の特定のポイントを繰り返し参照し、そのたびにそこへスクロールするのが面倒である場合、ブックマークを使用します。ブックマークを設定するには、その設定位置を正確に指定しなければなりません。選択または反転表示できる項目はどれでもブックマークとして指定できます。この例では、3行目のNC命令をブックマークとして設定します。

### ブックマークの設定手順

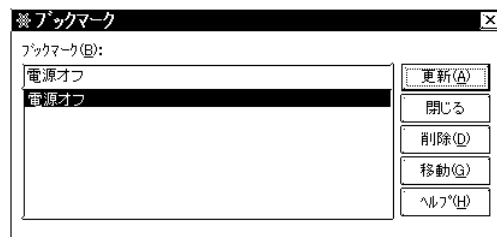
1. 3行目のNC命令をクリックします。
2. [検索]メニューから、[ブックマーク]を選択します。[ブックマーク]ダイアログボックスが表示されます。



3. [ブックマーク]フィールドに「電源オフ」と入力し、[追加]をクリックします。  
これでブックマークが設定されました。「電源オフ」を選択し[移動]を入力してブックマークに戻ると、プログラムのどこからでも必ず3行目のNC命令に戻ります。新しいブックマークを設定するときは、ロジックプログラムの新しいポイントを選択し、手順1)～3)を繰り返します。Editorでは、複数のブックマークが利用できます。

### ブックマークに飛ぶ手順

1. [検索]メニューから、[ブックマーク]を選択します。[ブックマーク]ダイアログボックスが表示されます。



2. [ブックマーク]ダイアログがリストから任意のブックマークを選択し、[移動]をクリックします。  
ロジックプログラムが表示されているときは、自動的にブックマークの場所に戻ります。  
参考: CTRL+Mを使って[ブックマーク]ダイアログボックスをオープンできます。

### ブックマークの位置の変更手順

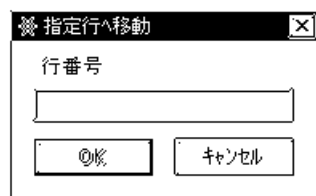
1. ロジックプログラムの中の新しい位置を選択します。
  2. 位置を変更したい任意のブックマークリストから選択します。
  3. [ブックマーク]ダイアログボックスの[更新]をクリックします。
- 以上の手順で、ブックマークの変更ができます。

## 2.8.5 [指定行へ移動] コマンドの使用

[指定行へ移動]コマンドを使って、選択位置をロジックプログラムの指定した行に移動できます。

### [指定行へ移動] コマンドの使用手順

1. [検索]メニューから、[指定行へ移動]を選択します。[指定行へ移動]ダイアログボックスが表示されます。



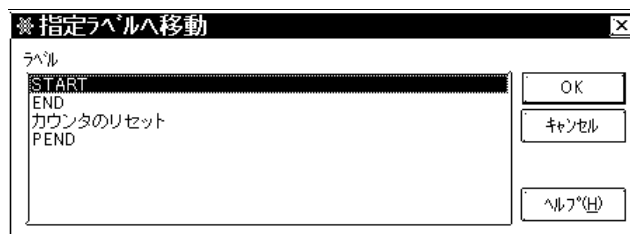
2. 行番号を入力します。
3. [OK]をクリックすると、指定した行に移動します。

## 2.8.6 [指定ラベルへ移動] コマンドの使用

[指定ラベルへ移動]コマンドを使って、ロジックプログラムの特定のラベルにジャンプできます。

### [指定ラベルへ移動] コマンドの使用手順

1. [検索]メニューから、[指定ラベルへ移動]を選択します。[指定ラベルへ移動]ダイアログボックスが表示されます。



2. 移動したいラベルを選択します。
3. [OK]をクリックすると、指定したラベルに移動します。

## まとめ

このレッスンでは、以下の操作を学習しました。

- ・ [検索]、[参照]、[ブックマーク]、[指定行へ移動]、および[指定ラベルへ移動]の各コマンドの使用方法

## 2.9 I/Oの割り付け

ロジックプログラムの作成が完了した後、選択した変数に実I/Oを割り付けることで、実際のI/O制御が可能です。このチュートリアルでは、使用方法を順序良く論理的に説明するため最初に変数を作成し、ロジックプログラムが完成した後に実I/Oを割り付ける流れを取りましたが、プログラミングを開始する前に実I/Oの配置が決まっていれば最初にI/O割り付けを行い、その後でプログラム上に実I/Oと共に変数を割り付けることができます。以下のレッスンでは両方の方法を説明します。

### 2.9.1 実I/Oへの変数の割り付け

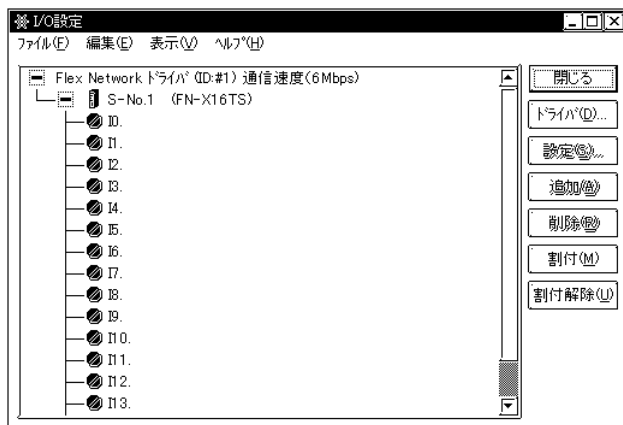
ロジックプログラムの変数を作成した後、いくつかの方法でそれをI/Oに割り付けることができます。割り付ける変数は以下の通りです。


「氷供給ボタン」、「カップ大ボタン」、「カップ中ボタン」、「カップ小ボタン」は、GLCからのタッチパネル入力とするので割り付けません。

変数名	端子タイプ	端子番号
電源オンボタン	入力	I0
氷供給のカップセット	入力	I2
電源オフボタン	入力	I6
ランプ	出力	Q0
氷供給	出力	Q1
ソーダバルブ	出力	Q2

#### I/O設定のオープン手順

[データ]メニューから、[I/O設定]を選択します。[I/O設定]ウィンドウが表示されます。

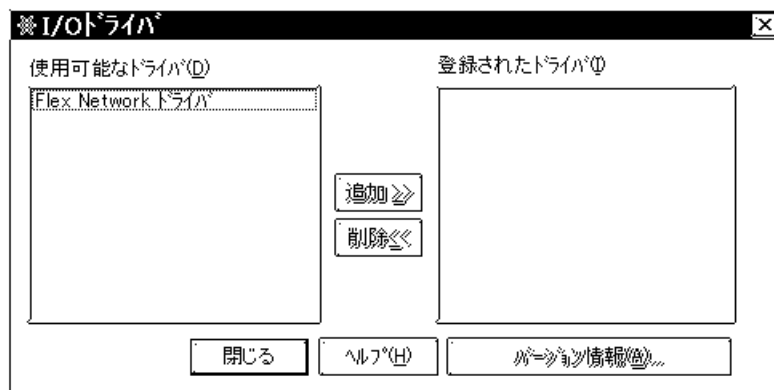


参考: ツールバーのをクリックするか、[変数設定]ダイアログボックスの[I/O(F)...]をクリックしても、[I/O設定]ウィンドウをオープンできます。



## ドライバの指定手順

1. [I/O設定]ウィンドウの[ドライバ]をクリックします。[I/Oドライバ] ウィンドウが表示されます。



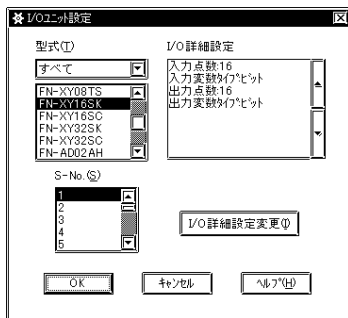
このウィンドウの左側の[使用可能なドライバ]には、現在選択されているGLCの機種で、使用可能なすべてのドライバがリストされます。同じく右側の[登録されたドライバ]には、プログラムに登録されているドライバがリストされます。現在何も登録されていません。

2. [使用可能なドライバ]から[Flex Network ドライバ]を選択します。
3. 任意のI/Oドライバを選択し[追加(>>)]をクリックするか、ドライバのタイトルをダブルクリックします。  
[登録されたドライバ]に選択したドライバが表示されます。
4. [閉じる]をクリックします。次頁のような[I/O設定]ウィンドウが表示されます。

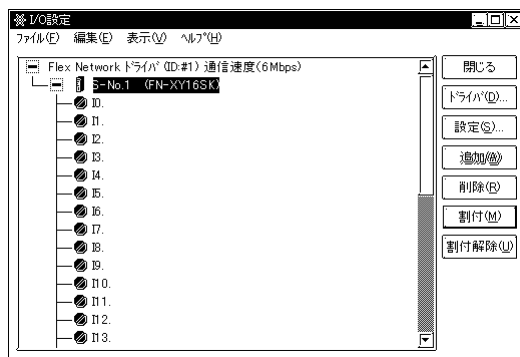
デフォルトの「型式」は「FN-X16TS」、[S-No.(号機番号)]は「1」に設定されています。この例では、[型式]を「FN-XY08TS」、[S-No.]を「1」に設定します。「FN-XY08TS」はそれぞれ8点ずつの入出力があります。

### Flex Network ドライバのセットアップ手順

1. [S-No.1 (FN-XY16TS)] を選択します。
2. [設定] をクリックします。設定ウィンドウが表示されます。



3. 「型式」から「FN-XY16SK」を選択します。
4. [OK] をクリックします。下のよう、[I/O 設定] ウィンドウが表示されます。



S-No.1 (FN-XY16SK) の下には、Flex Network ドライバに関連付けられている 16 個の入力端子と 16 個の出力端子が割り付けられています。このレッスンの後半で、それらに変数を割り付けます。

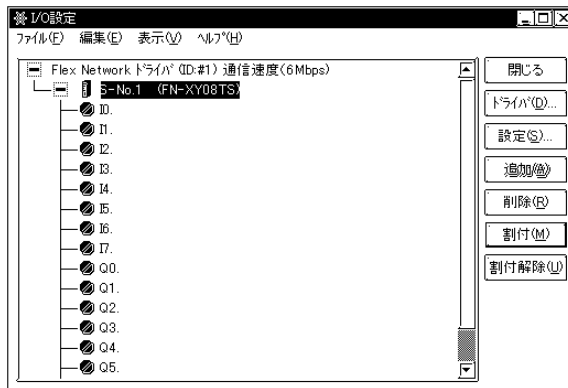
5. S-No.1 (FN-XY16SK) の隣の [ ] をクリックします。端子が非表示にされ、[ ] の代わりに [ ] が表示されます。



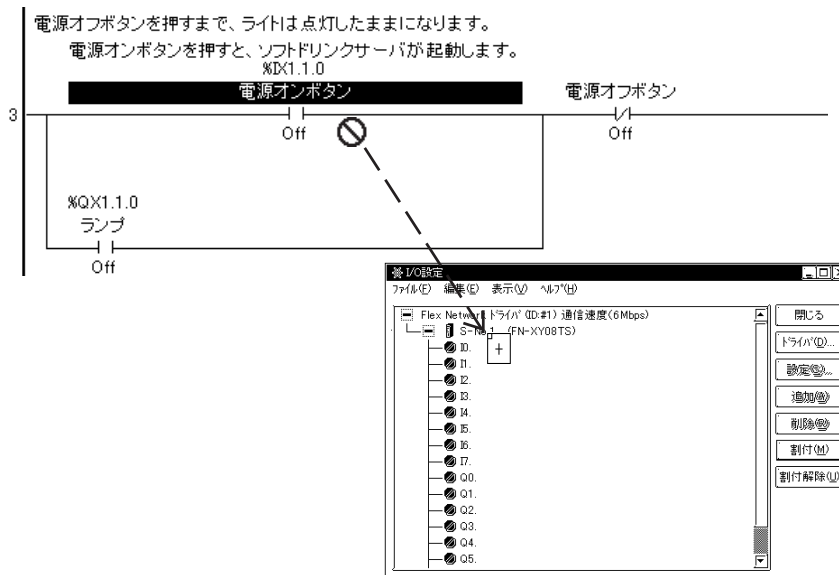
6. Flex Network ドライバは、最大 63 ユニット (2 回線使用時) まで接続できます。2 ユニット目の設定も、1 ユニット目と同様の設定です。

## ロジックプログラムの変数をクリックし、I/O 端子に割り付ける手順

1. S-No.1 (FN-XY08TS)の隣の $\oplus$ をクリックします。下のよう、[I/O設定]ウィンドウが表示されます。S-No.1 (FN-XY08TS)では、最初の8個の端子をビット入力のために使用します。



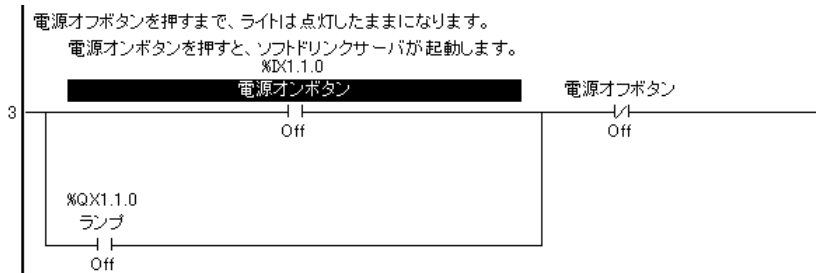
2. 3行目のa接点(N0命令)に設定された変数「電源オンボタン」を選択します。
3. 「電源オンボタン」をクリックして、端子 I0. にドラッグします。分岐を挿入する場合と同様に、カーソルは移動途中では $\oplus$ になります。カーソルがこの状態のときは、I/O 端子に変数を割り付けることはできません。



4. カーソルを端子 I0. の上にドラッグし、マウスを離します。変数「電源オンボタン」が端子 I0. に割り付けられます。

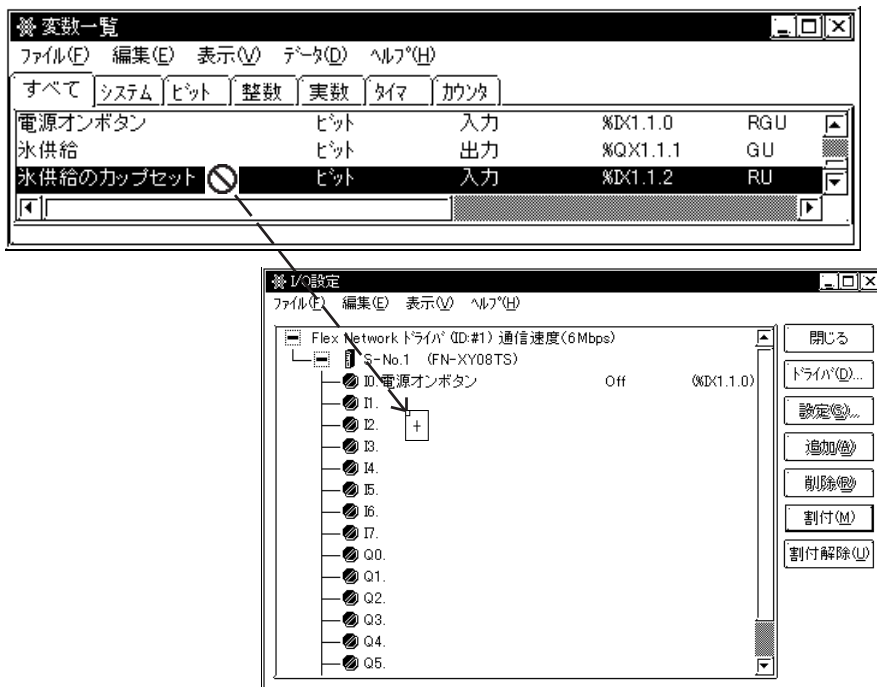


3行目のNO命令の上の変数「電源オンボタン」の上に、一連の数字および文字が表示されています。これは、変数の実 I/O アドレスです。I/O アドレスは IEC 規格に合致したフォーマットで表示されています。



**[変数一覧]ウィンドウから変数を I/O 端子に割り付ける手順**

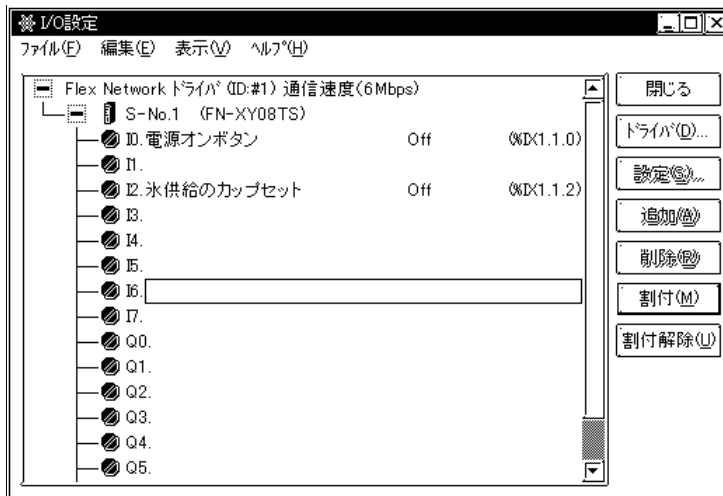
1. [I/O設定]ウィンドウは、オープンしたまま、[変数一覧]ウィンドウを開きます。
  2. 両方のウィンドウが見えるように、2つのウィンドウを配置します。
  3. [変数一覧]ウィンドウから、変数「氷供給のカップセット」をクリックし、[I/O設定]ウィンドウの端末 I2. にドラッグします。
  4. マウスを離します。変数「氷供給のカップセット」が入力端子 2 に割り付けられました。
- 参考: 上記の手順を使って、[コメント一覧]ウィンドウからの変数を I/O に割り付けることもできます。



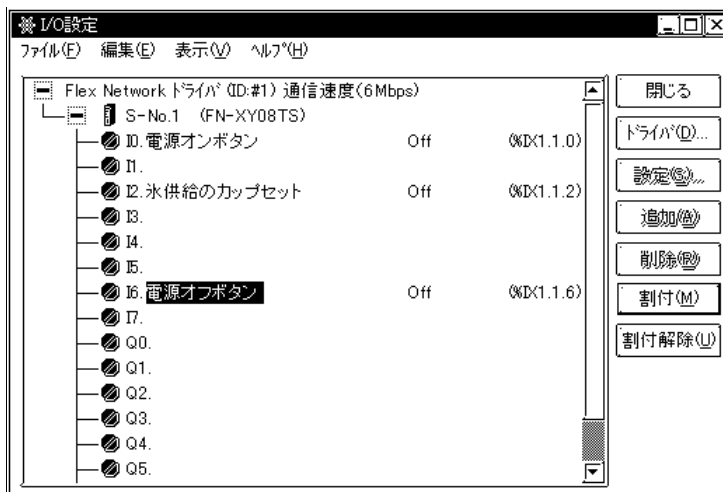
**[注意]** [変数一覧] [コメント一覧] ウィンドウから [I/O設定] ウィンドウに変数をドラッグして I/O 割り付けを行った場合、実 I/O の属性が優先され、変数属性が入力 / 出力などに変更されます。

## テキスト入力による割り付け手順

1. 端子 I6. をクリックします。
2. [Enter]を押します。端子テキストフィールドが起動します。



3. 「電源オフボタン」と入力します。
4. [Enter]を押します。「電源オフボタン」が入力端子 I6. に割り付けられます。

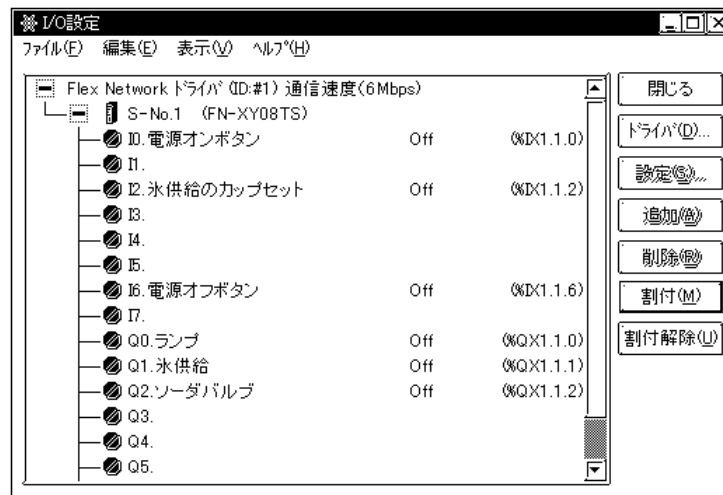


- ・ テキスト入力によって I/O 割り付けを行うと自動的に変数リストにも登録されます。

出力端子に変数を割り付ける方法は、入力端子への変数の割り付けと同じです。今までの手順を使って、以下の表の変数を入力および出力端子に割り付けてください。

変数名	端子タイプ	端子番号
ランプ	出力	Q0
氷供給	出力	Q1
ソーダバルブ	出力	Q2

下のように入力および出力ユニットが、[I/O設定]ウィンドウに表示されます。



## 2.9.2 [I/O 設定]ウィンドウの割り付け解除

[I/O 設定]ウィンドウの割り付け解除手順

1. [I/O 設定]ウィンドウの端子 I0. をクリックします。
2. [割り付け解除] をクリックします。「電源オンボタン」の端子 I0. への割り付けが解除されます。その後、この変数を任意の端子にも割り付けることができます。この練習では、この変数を端子 I0. に再び割り付けます。

## 2.9.3 I/O に割り付けた変数を命令で使用方法

新しいプログラムのためにI/Oを設定する最も簡単な方法は、変数を直接に端子上に入力する方法です。それらの変数は自動的に作成され、設定され、正しいI/Oポイントに割り付けられます。先にI/Oを設定してから、ロジックプログラムを作成する場合、下記の方法で、I/Oポイントを作成します。

I/O に割り付けた変数を命令で使用する手順

1. 変数をクリックし、I/O端子にドラッグする手順で示した方法を使って、変数をドライバの入力および出力端子に割り付けます。
2. ロジックプログラムを作成します。
3. [I/O 設定]ウィンドウから変数をクリックし、I/O を使用する命令にドラッグします。

## 2.9.4 I/O設定のインポート / エクスポート

I/O設定で割り付けられた変数をCSV形式のファイルとしてインポート、エクスポートすることができます。

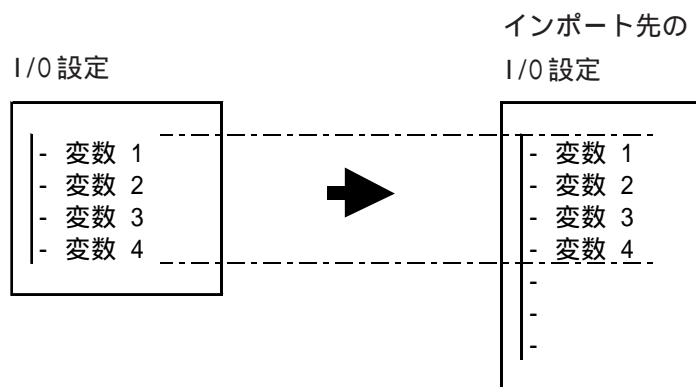
I/O設定のデータがCSV形式のファイルでエクスポートされるため、データの作成や編集が汎用表計算ソフトでも行うことができます。

用途として、DIOユニットからFlex Networkユニットへ、Flex Networkユニットから別のFlex Networkユニットなど、I/O設定に割り付けられた変数を別のドライバへ移す場合にご利用頂けます。

### ドライバ間の端子数が異なる場合のインポート

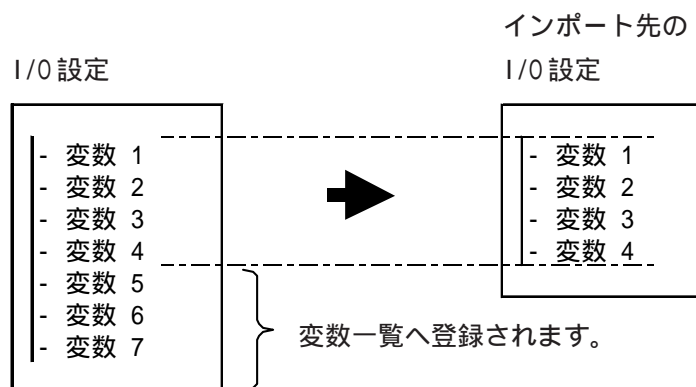
#### 対応する端子が存在する場合

DIOユニット(16点/16点)からFlex Network 64点 I/Oユニット(32点/32点)へインポートする場合のように、インポート先のドライバに対応する端子が存在する場合は、問題なくすべての変数がインポートされます。



#### 対応する端子が存在しない場合

インポート先のドライバに対応する端子が存在しない変数は、I/O設定へインポートされません。インポートされなかった変数については、変数一覧に登録されますので、必要に応じて変数一覧からI/O設定へ変数の割り付けを行ってください。



### CSV ファイルのフォーマット

[I/O設定]ウィンドウから[エクスポート]ボタンをクリックすると、I/Oに割り付けられている変数の情報が以下のようなフォーマットで、CSV形式のファイルで出力されます。

ヘッダ情報		ドライバ情報										
//	ProductName	Pro-Control Editor										
//	FileVersion	5										
//	ProductVersion	5.00 Build (24)										
//	CompanyName	Digital Electronics Corp.										
//	LegalCopyright	Copyright(C) Digital Electronics Corp.										
//	CSV FileVersion	1										
@@	Driver Type	Unit	Offset	Variable	I/O Set							
GLC		0	0	1	0							
@@	Name	Data Type	ID	Data Type	Array Size	I/O Type	ID	I/O Type	I/O Address	I/O Offset	Attribute	comment
1	電源オンボタン	1	ビット			1	入力		%I1.0.0	0		
2	氷供給のカップセット	1	ビット			1	入力		%I1.0.2	0		
3	電源オフボタン	1	ビット			1	入力		%I1.0.6	0		
4	ランプ	1	ビット			2	出力		%Q1.0.0	0		
5	氷供給	1	ビット			2	出力		%Q1.0.1	0		
6	ソーダバルブ	1	ビット			2	出力		%Q1.0.2	0		
		変数情報										

#### ヘッダ情報

エクスポートした CSV ファイルは、Pro-Control Editor のヘッダ情報が添付されます。インポートを行う場合には、この情報はインポート先のプロジェクトには反映されません。ユーザーにて CSV ファイルの管理用情報として自由にご利用ください。

- ProductName : プロジェクト名を格納します。
- FileVersion : ファイルのバージョンを格納します。
- ProductVersion : 編集しないでください。
- CompanyName : 会社名を格納します。
- LegalCopyright : Digital Electronics Corp. (著作権)
- CSV FileVersion : 編集しないでください。



・ I/O 設定を CSV 形式のファイルで新規に作成する場合、”ProductVersion” と ”CSV FileVersion” の項目には何もいれないでください。

#### ドライバ情報 (入力必須)

GLCにオンライン接続されているユニットの情報です。

Driver Type : ドライバのタイプを以下のID番号で格納します。ドライバを未使用の場合は ”0” を格納してください。

ドライバタイプ	ID番号
DIOドライバ	0
Flex Networkドライバ	1



Unit Offset : "0" を格納してください。  
 Variable Set : CSVファイルの変数名がインポート先の変数名と重複した場合の処理方法を以下の ID 番号で格納します。

処理方法	ID番号
上書き	0
追加	1



- ・ 本設定は「ファイル/オプション/確認タブ」の「ロジックエディタの操作確認あり」のチェックボックスが無効になっているとき有効になります。

I/O Set : CSVファイルのI/Oアドレスがインポート先のI/Oアドレスと重複した場合の処理方法を以下の ID 番号で格納します。

処理方法	ID番号
上書きか追加を随時メッセージダイアログボックスで確認する	0
上書き	1
追加	2

#### 変数情報（入力必須）

I/Oに割り付けられている変数の情報です。

Name : 変数名を格納します。変数名をつける際の制限については、[参照](#)「Pro-Control Editor ユーザーズマニュアル 2.1 変数名」  
 Data Type ID : 変数タイプ（ビットや整数など）を以下の ID 番号で格納します。変数タイプの詳細については、[参照](#)「Pro-Control Editor ユーザーズマニュアル 2.2 変数タイプ」

変数設定	ID番号
ビット	1
整数	2
実数	3
タイマ	11
カウンタ	12

Data Type : "Data Type ID" に対応するコメントです。エクスポートした CSV ファイルにはコメントが入っていますが、新規で作成する場合などは不要です。

Array Size : 配列サイズを格納します。配列の詳細については、[参照](#)「Pro-Control Editor ユーザーズマニュアル 2.3 配列変数へのアクセス」

I/O Type ID : I/Oタイプ(入力や出力など)を以下のID番号で格納します。

I/Oタイプ	ID番号
内部変数	0
入力	1
出力	2

I/O Type : "I/O Type ID"に対応するコメントです。エクスポートしたCSVファイルにはコメントが入っていますが、新規で作成する場合などは不要です。

I/O Address : 以下のフォーマットでI/Oアドレスを格納します。下線部の"% "と"1 "は固定です。

I/Oアドレスフォーマット: %AB1.C.D

Aについて

入出力端子について下表のID記号を格納します。

入出力端子	ID記号
入力端子	I
出力端子	Q

Bについて

ビット端子の場合は"X"、ワード端子の場合は"W"を格納します。

Cについて

Flex Networkの場合はS-No.の番号を格納します。DIOドライバの場合はユニット番号(0もしくは1)、ユニワイヤドライバの場合はエリア番号(0~15)になります。

Dについて

端子番号を格納します。

I/O Offset : "0"を格納してください。

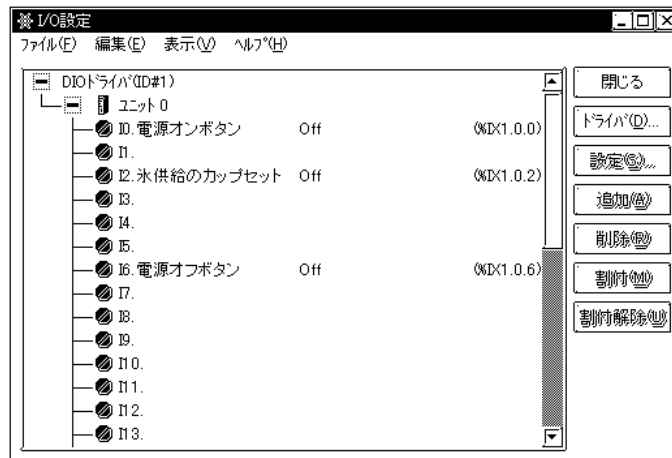
Attribute : 保持型、画面共有変数を以下のID記号で格納します。

変数属性	ID番号
保持型/画面共有変数	RG
保持型/非画面共有変数	R
非保持/画面共有変数	G
非保持/非画面共有変数	(空白)

Comment : コメントを格納します。

## 設定手順 (DIOユニット Flex Network)

GLC100S にオンライン接続されている DIO ユニットから、GLC2300 にオンライン接続されている Flex Network[FN-XY16SK]へのを例にして説明します。

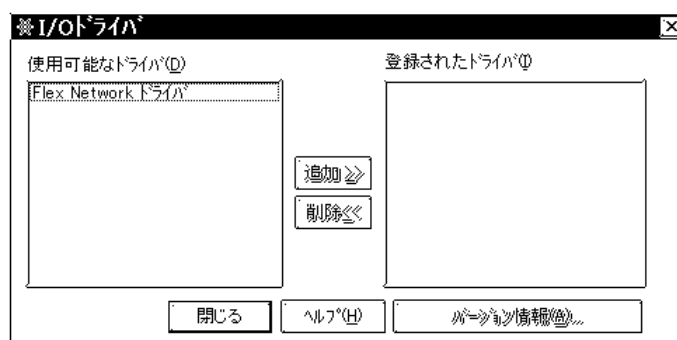


1. [I/O設定]ウィンドウで[エクスポート]ボタンをクリックします。DIOドライバに割り付けられている変数を CSV ファイルに書き出して保存します。

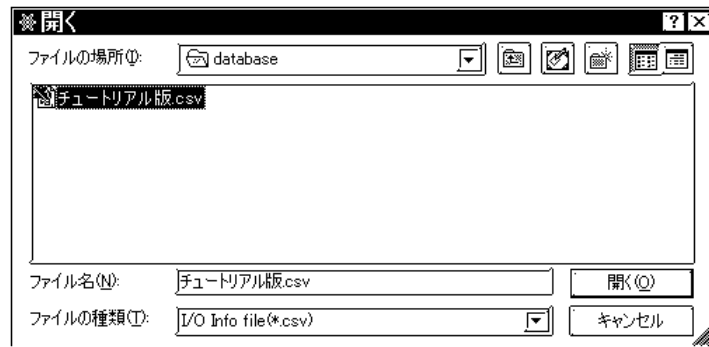


MEMO ・ 保存した CSV ファイルは、汎用表計算ソフトで編集できます。

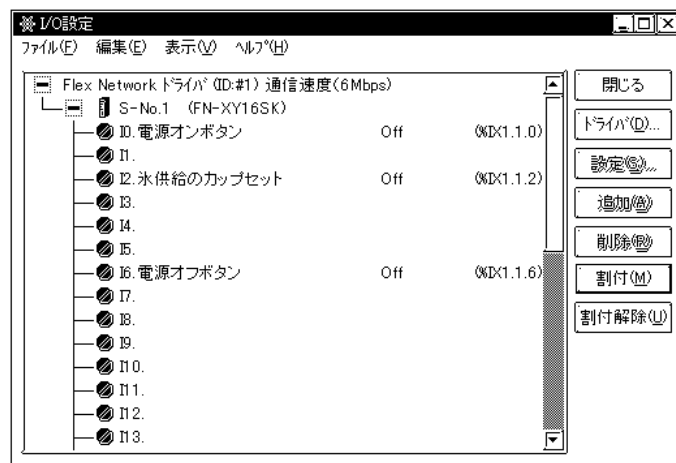
2. プロジェクトマネージャの「新規作成 /GP タイプ」で GLC100S から GLC2300 へ機種変更します。参照 「2.1 起動のしかた」
3. Editor から [I/O 設定] ウィンドウを開き、Flex Network ドライバを設定します。



4. [S-No.1(FN-XY16SK)]を選択して、[インポート]ボタンをクリックします。保存したCSVファイルを選択して[開く]ボタンをクリックします。



5. CSV ファイルから変数を取り込み、Flex Network へ割り付けを行います。



**重要** ・ 変数一覧からエクスポートされた CSV ファイルを I/O 設定へインポートしないでください。

## まとめ

このレッスンでは、以下の操作を学習しました。

- ・ I/O 設定の CSV ファイル出力
- ・ DI0 ユニットから Flex Network ドライバへの移行
- ・ 変数の I/O への割り付け

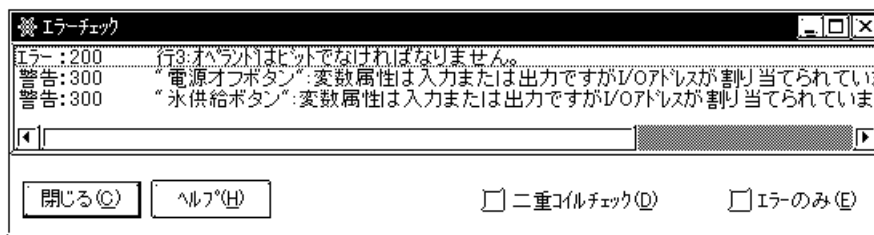
## 2.10 プログラムエラーチェック

ロジックプログラムをGLCに転送して実行する前に、プログラムにエラーがないことを確認しておかなければなりません。そのためにエラーチェックを行います。

エラーの詳細については、[参照](#) 「第7章 エラーと警告」

### エラーチェック実行手順

[ファイル]メニューから、[エラーチェック]を選択します。[エラーチェック]ウィンドウが表示されます。



[エラーチェック]ウィンドウは、Editorがプログラムで検出できるすべてのエラーおよび考えられる問題箇所をリストします。問題箇所は、警告としてリストされます。ウィンドウの右下隅には、[エラーのみ]というマークが付いているチェックボックスがあります。このボックスをチェックすると、Editorがプログラムの中で検出したエラーだけが表示され、警告は表示されません。

コントローラは警告を含むプログラムをGLCに書き込むことはできますが、エラーを含むプログラムはGLCに書き込むことはできません。これらのエラーを修正しなければなりません。

参考: エラーチェックは、ツールバーの  をクリックすることによって実行することもできます。

[エラーチェック]ウィンドウは、エラーおよび警告を、ロジックプログラムで検出された順序に表示します。つまり、1行目のエラーが最初に表示され、次に2行目以下のエラーが順に表示されます。[エラーチェック]ウィンドウのエラーまたは警告をダブルクリックすると、問題の行を直接表示します。

- ロジックの問題である場合、プログラムのその部分が表示されます。
- I/O割り付けに関する問題である場合、[I/O設定]ウィンドウが表示されます。

### エラーの修正手順

1. [エラーチェック]ウィンドウのエラー行をダブルクリックします。9行目のオペランドが反転表示され、それに割り付けられる変数がないことを知らせます。
2. 変数として「ソーダバルブ」と入力します。  
それぞれのエラーおよび警告に関する詳細については、オンラインヘルプまたは「付録A: エラーと警告」を参照してください。  
[エラーチェック]ウィンドウにリストされているエラーの修正が終わったら、もう一度エラーチェックを実行します。まだ残っているエラーが表示されます。エラーをすべて修正した後、プログラムをコントローラに書き込むことができます。

### まとめ

このレッスンでは、以下の操作を学習しました。

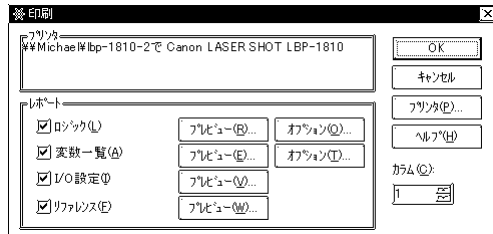
- ・ ロジックプログラムのエラーチェックの方法  
これでプログラムをGLCに転送して実行する準備が完了しました。  
このあとの手順は、「3.1 コントローラの設定」で詳しく説明します。

## 2.11 ロジックプログラムの印刷

Editor では、ロジックプログラムを印刷することができます。

### ロジックプログラムの印刷手順

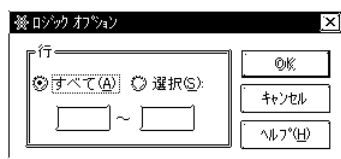
1. [ファイル]メニューから、[印刷]を選択します。[印刷オプション]ウィンドウが表示されます。プレビュー機能で印刷イメージを確認することができます。



レポートを印刷する際のカラムの数(1~4)を選択できます。[レポート]セクションの下に、[ロジック]、[変数一覧]、[I/O設定]、および[リファレンス]というラベルが付いた4つのチェックボックスがあります。これらのチェックボックスは、ロジックプログラムを印刷するときに、以下のオプションを提供します。

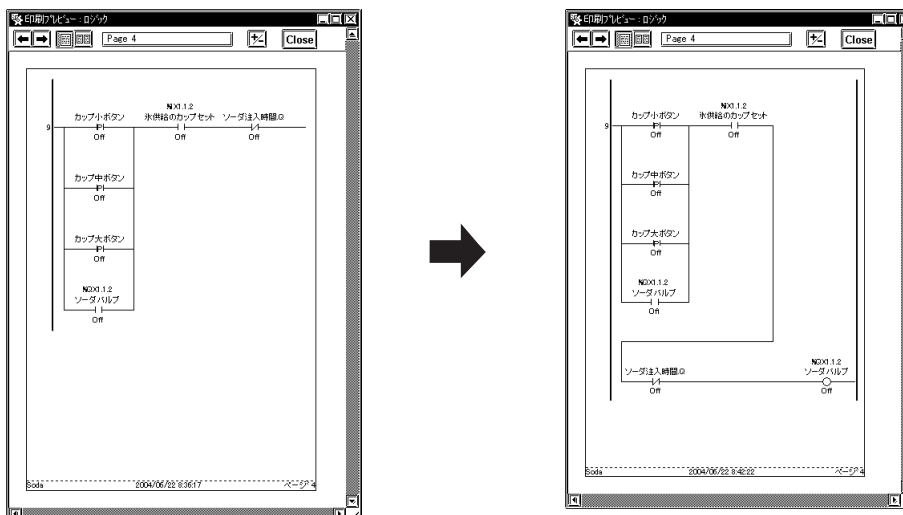
ロジック：

この項目では、ロジックプログラムの行を印刷できます。その隣の[オプション]をクリックすると、下のようなウィンドウが表示されます。



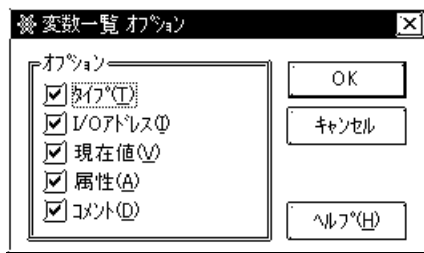
プログラムのすべての行を印刷するには[すべて]を選択し、プログラムの特定の範囲の行を印刷するには[選択]を選択し、その範囲を入力します。

[表示]メニューの拡大、縮小で印刷するロジックの大きさを調整できます。縮小しても右端部分が画面に収まりきれない場合、[表示/行の折り返し]をクリックすると、画面内に収まるように行を折り返して印刷することができます。



変数一覧：

この項目では、変数一覧のオプションが設定できます。目的のチェックボックスをクリックして、変数一覧に含める項目を選択します。




変数一覧オプション：

このダイアログでは、変数の設定を印刷できます。

オプション	説明
タイプ	変数タイプを表示します。
I/Oアドレス	割り付けられた変数すべてのI/Oアドレスを表示します。
現在値	すべての変数の現在値を表示します。
属性	保持、画面共有変数の設定を表示します。
コメント	変数に付加されたコメントを表示します。



- ・ このダイアログでは、全変数の全項目を表示するリファレンスを印刷できます。
- ・ ツールバーのをクリックすることによっても、プログラムを印刷できます。

## まとめ

このレッスンでは、以下の操作を学習しました。

- ・ ロジックプログラムを印刷する



## 2.12 ロジックプログラムのインポート / エクスポート

Editor では、ロジックプログラムのみをエクスポートし、ロジックプログラムファイル (\*.wll) として保存することができます。

また逆に、ロジックプログラムファイル (\*.wll) をインポートして別のプロジェクトファイル (\*.prw) のロジックプログラムとして使用できます。

インポート/エクスポートできるロジックプログラムは、プロジェクトで作成したすべてのロジックプログラムはもちろん、ロジックプログラム的一部分をエクスポートする「部分エクスポート」、ロジックプログラム的一部分をインポートする「挿入インポート」などがあります。

### 2.12.1 エクスポート

エクスポートできるロジックプログラムは、以下の3種類があります。

- ・ サブルーチンを含む、すべてのロジックプログラム (\*.wll)
- ・ 選択した一部のロジックプログラム (\*.wlp)
- ・ サブルーチン部分のロジックプログラム (\*.wlf)

#### エクスポート手順

上記3種類のエクスポート手順を説明します。

#### サブルーチンを含む、すべてのロジックプログラム

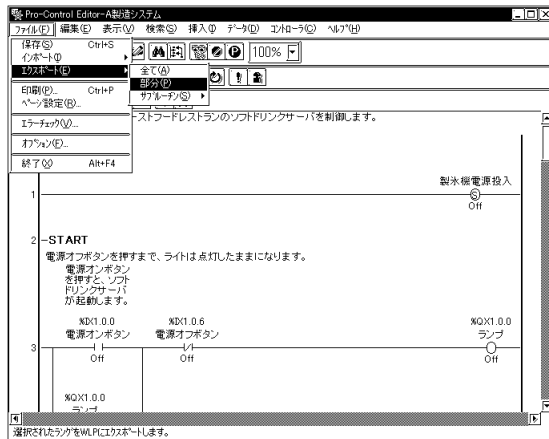
1. [ファイル]メニューから、[エクスポート / 全て]を選択します。



2. [名前を付けて保存]ウィンドウでファイル名を入力します。
3. [保存]をクリックします。  
ロジックプログラムがWLLファイルとして保存されます。

### 選択した一部分のロジックプログラム

1. [ファイル]メニューから、[エクスポート / 部分]を選択します。



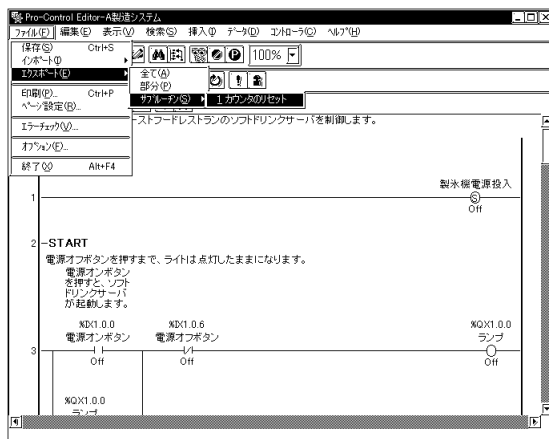
2. エクスポートする行の範囲を設定し、[OK]ボタンをクリックします。



3. [名前を付けて保存]ウィンドウでファイル名を入力します。
4. [保存]をクリックします。  
ロジックプログラムがWLPファイルとして保存されます。

### サブルーチン部分のロジックプログラム

1. [ファイル]メニューから、[エクスポート/サブルーチン]を選択します。[サブルーチン]を選択すると、ロジックプログラムで作成されたサブルーチンの一覧が表示されます。一覧からエクスポートするサブルーチンを選択してください。



2. [名前を付けて保存]ウィンドウでファイル名を入力します。
3. [保存]をクリックします。  
ロジックプログラムがWLFファイルとして保存されます。

## 2.12.2 インポート

エクスポートされたロジックプログラムをインポートしてやる方法として、以下の3種類があります。

- ・ サブルーチンを含む、すべてのロジックプログラムをインポートする「上書き」
- ・ 選択した一部分のロジックプログラムをインポートする「挿入」
- ・ サブルーチン部分をインポートする「サブルーチン」

サブルーチンを含む、すべてのロジックプログラムをインポートする場合、現プロジェクトのロジックプログラムに「上書き」されますのでご注意ください。

行をインポートする位置は、「ファイル / オプション / 編集」で設定することができます。

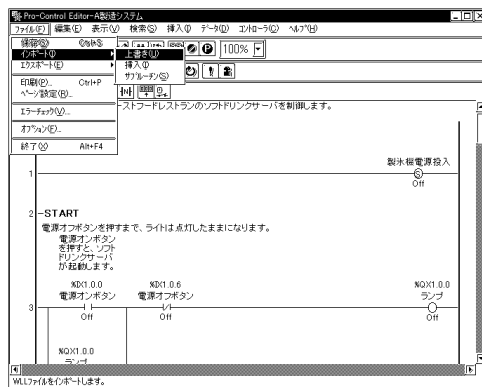
参照 「第2章 プログラムの作成 ロジックプログラムを作成する前にオプション画面で設定を行う」

### インポート手順

上記3種類のインポート手順を説明します。

#### サブルーチンを含む、すべてのロジックプログラム

1. [ファイル]メニューから、[インポート / 上書き]を選択します。



2. [開く]ウィンドウでインポートしたいWLLファイルを選択します。

3. [開く]をクリックします。

ロジックプログラムがインポートされ、インポートしたロジックプログラムで使用されている変数に変数一覧に登録されます。

4. ロジックプログラムを保存すると、画面共有変数がロジックシンボルとしてシンボルエディタに登録されます。

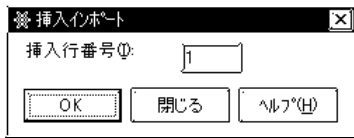
参照 「GP-PRO/PB オペレーションマニュアル 4.7 シンボルエディタ」

#### 選択した一部分のロジックプログラム

1. [ファイル]メニューから、[インポート / 挿入]を選択します。



2. ロジックプログラムを挿入する場所（行番号）を指定します。



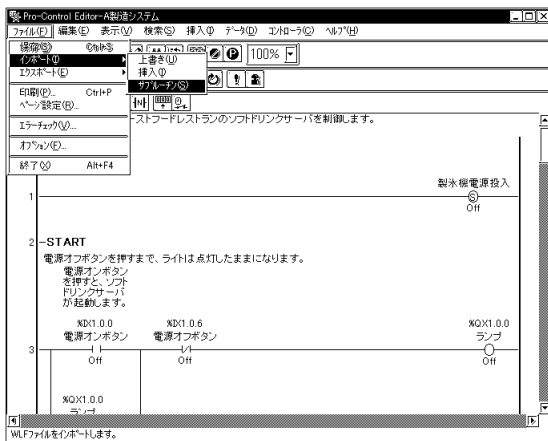
3. [開く]ウィンドウでインポートしたいWLPファイルを選択します。
4. [開く]をクリックします。  
ロジックプログラムがインポートされ、インポートしたロジックプログラムで使用されている変数が変数一覧に登録されます。
5. ロジックプログラムを保存すると、画面共有変数がロジックシンボルとしてシンボリエディタに登録されます。

参照 「GP-PRO/PB オペレーションマニュアル 4.7 シンボリエディタ」

**重要** ・ インポートするロジックプログラムとインポートされるロジックプログラムに同名の変数が存在する場合、変数タイプはインポートされる側の変数タイプに変更されます。

### サブルーチン部分のロジックプログラム

1. [ファイル]メニューから、[インポート/サブルーチン]を選択します。



2. [開く]ウィンドウでインポートしたいWLPファイルを選択します。
3. [開く]をクリックします。  
ロジックプログラムがインポートされ、インポートしたロジックプログラムで使用されている変数が変数一覧に登録されます。
4. ロジックプログラムを保存すると、画面共有変数がロジックシンボルとしてシンボリエディタに登録されます。

参照 「GP-PRO/PB オペレーションマニュアル 4.7 シンボリエディタ」

**重要** ・ インポートするロジックプログラムとインポートされるロジックプログラムに同名の変数が存在する場合、変数タイプはインポートされる側の変数タイプに変更されます。

## まとめ

このレッスンでは、以下の操作を学習しました。

- ・ ロジックプログラムをインポート / エクスポートする

## 2.13 画面プログラムの開発

GP-PRO/PB で、「氷供給ボタン」、「カップ大ボタン」、「カップ中ボタン」、「カップ小ボタン」を作成します。下図は画面の完成例です。

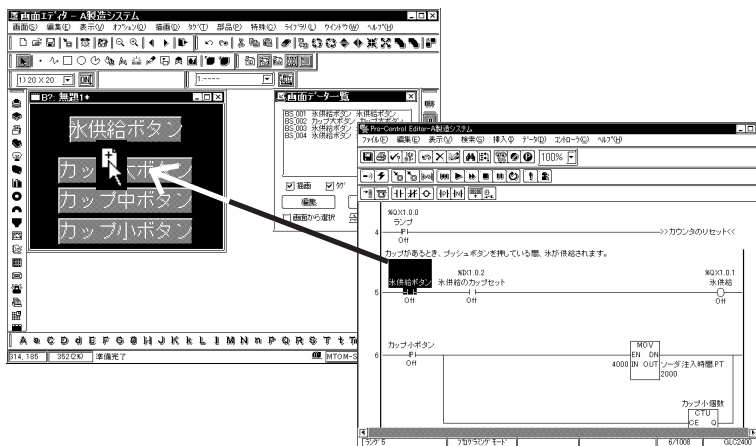


### GP-PRO/PB の起動

1. プロジェクトマネージャから[作画 / 画面]をクリックし、GP-PRO/PB を起動します。
2. メニューバーの[画面 / 新規作成]をクリックし、[ベース画面]が選択されていることを確認して[OK]ボタンを押します。

### ドラッグ&ドロップによる作画手順

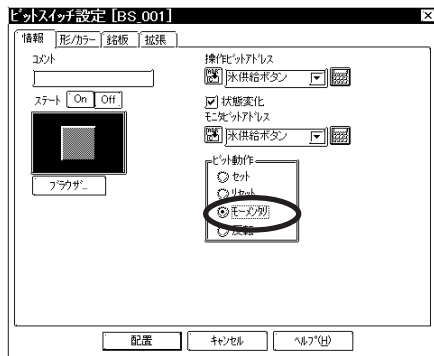
1. ロジックプログラム中の「氷供給ボタン」を選択し、GP-PRO/PB へドラッグします。



2. GP-PRO/PB でドロップすると「部品選択」ダイアログボックスが表示されるので、「ビットスイッチ」を選択して[OK]ボタンを押します。



3. [ビットスイッチ設定]ダイアログボックスが表示されます。[ビット動作]を[モーメンタリ]に設定してください。また、操作ビットアドレスが「氷供給ボタン」になっていることを確認して[配置]してください。



4. 「氷供給ボタン」スイッチの完成です。同様に「カップ大ボタン」、「カップ中ボタン」、「カップ小ボタン」を作成してください。



5. 作画データを GLC2300 に転送します。  
参照 「GP-PRO/PB オペレーションマニュアル 第7章 データ転送」

# 第3章

# ロジックプログラム を実行する

エラーのないロジックプログラムが完成したら、GLCにダウンロードしてコントローラ(ランタイム)上で実行することができます。

本章では、コントローラの設定方法、プログラムの書き込み方、モニタモードでの実行方法について説明します。

## 3.1 コントローラの設定

ロジックプログラムをGLCに書き込む前に、コントローラが正しく設定されていることを確認してください。GLCでプログラムを実行する場合、コントローラの設定にはシステムの設定、通信の設定、メモリという3つの項目があります。

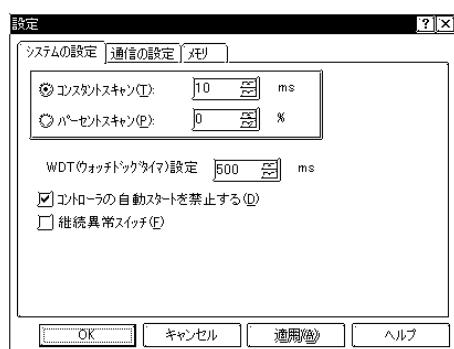
### コントローラの設定手順

[コントローラ]メニューで[設定]を選択します。

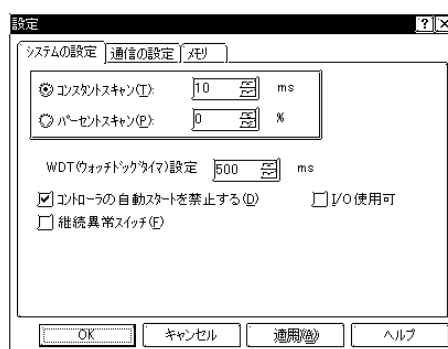
システムの設定

[システムの設定]タグを選択します。

<GLC100/GLC300>



<GLC2000 シリーズ、LT シリーズ>



[システム設定]に設定した値は、コントローラでロジックプログラムを実行する際の設定値になります。設定後にプログラムを実行すると、手動で変更しない限り、コントローラでは常にこの設定が適用されます。設定内容は個別の実行プログラム毎に設定されます。

[システム設定]タブの各項目の内容は次ページの表の通りです。

項目	内容
コンスタントスキャン	システム変数 #TargetScanに、スキャンタイムをms単位で指定します。 注：コンスタントスキャンモードでも、ロジックタイムがスキャンタイムの50%を超える場合はコンスタントなスキャンを保証できません。10ms単位で設定します。
パーセントスキャン	システム変数 #PercentAllocに、適切なスキャンタイムを、総処理時間に占める比率で指定します。求められたスキャンタイムの1msの位は切り上げられます。
WDT (ウォッチドッグタイム) 設定値	ロジックプログラムの異常でスキャンタイムが延び、ここで設定した値を超えてしまうと「メジャー異常」でお知らせします。システム変数 #WatchdogTimeでも同機能の設定が行えます。参照「Pro-Control Editorユーザーズマニュアル 第3章 システム変数」
コントローラの自動 スタートを禁止する	GLCのオフラインメニューで「電源ON時の動作モード」を [DEFAULT]に設定した場合のみ本設定は有効になります。*1 電断後にコントローラが起動するとき、コントローラがロジックプログラムを自動的に実行しないことを設定します。システム変数 #DisableAutoStartでも同機能の設定が行えます。参照「Pro-Control Editorユーザーズマニュアル 第3章 システム変数」
継続異常スイッチ	コントローラにマイナー異常が発生したときに、ロジックプログラムの実行を終了するかどうかを設定します。システム変数 #FaultOnMinorでも同機能の設定が行えます。参照「Pro-Control Editorユーザーズマニュアル 第3章 システム変数」
I/O使用可 (GLC2000シリーズ/ LTシリーズのみ)	GLCの本体やI/Oユニットの外部I/Oへの入出力を可能にする動作です。通常ロジックプログラムのダウンロードをおこなった後、GLCを運転状態にただけでは外部I/Oの入出力をおこなうことができません。これは、操作やロジックプログラムのミスなどで、機械が突然動作することを防止するための安全を優先する考え方の機能です。



- ・ コンスタントスキャン、パーセントスキャンについては、「Pro-Control Editor ユーザーズマニュアル 第1章 コントローラ機能」を参照してください。
- ・ システム変数の詳細については、「Pro-Control Editor ユーザーズマニュアル 第3章 システム変数」を参照してください。
- ・ コントローラの運転/停止操作においても「I/O使用可」が選択できません。詳細は、「3.2 コントローラの運転/停止」を参照してください。

\*1「電源ON時の動作モード」は、「動作環境」「コントローラメニュー」「コントローラ設定1」で設定します。ここで、[運転]または[停止]に設定した場合、Pro-Control Editorの設定は無視され、オフラインでの設定が優先されます。



## 通信の設定

「通信の設定」タブをクリックします。



- ・ 画面転送設定と同じ(M)  
GP-PRO/PB で「画面の転送 / 転送設定」での設定した項目が有効になります。
- ・ COM  
シリアル通信でのシリアルポート、伝送速度を設定します。ただし、GLC100 は伝送速度 115.2Kbps をサポートしていません。

イーサネット経由でロジックプログラムの書き込み / 読み込み、およびモニタモードの実行ができます。

## 転送設定

[コントローラ]メニューで、[設定]を選択します。[設定]メニューの[通信の設定]タブで「イーサネット」または「イーサネット：自動取得」を選択します。「画面転送設定と同じ」を選択すると、GP-PRO/PB の「転送」の設定が有効になります。

項目	内容
イーサネット	イーサネット経由で通信するGLCのIPアドレス、ポート番号を設定します。[ロジックプログラム書き込み]、[ロジックプログラム読み込み]または[モニタモードへの移行]を実行するとイーサネット経由で通信をはじめます。
イーサネット：自動取得	[ロジックプログラム書き込み]、[ロジックプログラム読み込み]または[モニタモードへの移行]を実行するとイーサネット上のGLCを検索し、検索結果が一覧表示されます。検索結果より通信したいGLCのIPアドレスを選択します。[ロジックプログラム書き込み]では、複数のGLCを選択できます。



- ・ Editorの[コントローラ]メニューの[コマンド/オンライン接続]からでも、イーサネット通信をはじめます。

(例) イーサネット：自動取得

イーサネット上にオンライン接続されているGLCを一覧表示します。

IPアドレス	ポート
160.187.3.45	8000
160.187.3.46	8000

検索

追加

キャンセル ヘルプ(H)

IPアドレスを指定して[追加]を選択すると指定したIPアドレスのGLCとオンライン接続します。

IPアドレス	ポート
160.187.3.45	8000
160.187.3.46	8000

検索

160.187.3.47 追加

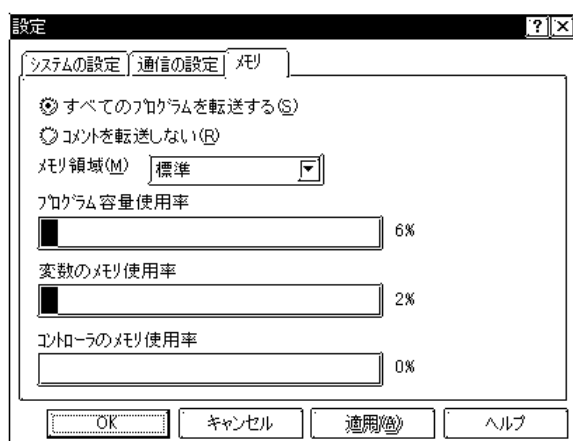
キャンセル ヘルプ(H)



- MEMO ・ LAN カードを2枚以上使用する環境で「イーサネット：自動取得」を指定すると、目的のGLCが検索できない場合があります。これは、OSが先に見つけたLANカードに接続されたGLCを検索するためです。  
転送設定を「イーサネット」に指定し、GLCのIPアドレスを直接入力し転送してください。

## メモリ

[メモリ]タブをクリックします。



- すべてのプログラムを転送する  
コメントを含むすべてのロジックプログラムを転送します。  
GLCより読み込むとロジックプログラムのコメントを読み込みめます。
- コメントを転送しない  
GLCに書き込むファイルのサイズを小さくすることができます。コメントを転送しなかった場合は、GLCより読み込んでもコメントはありません。
- メモリ領域  
「標準」と「変数優先」が選択できます。本機能はGLC2400シリーズ<sup>1</sup>/GLC2600シリーズ<sup>1</sup>のみ対応で、メモリ領域の配分を選択する機能です。  
変数領域を優先すると、定数、ラベル、PT/NT命令領域を縮小し、変数領域を拡張します。  
下表は、ユーザーが作成可能な最大個数です。(転送時にコメントを転送しない場合)

機種	GLC2300、GLC2500、LT、 GLC2400/GLC2600の「Rev.*-2以降」 <sup>1</sup>	GLC2400/GLC2600の「Rev.*-無し、1」 <sup>1</sup>	
		標準	変数優先
メモリ領域	-	標準	変数優先
変数	4405個 <sup>2</sup>	4045個 <sup>2</sup>	4408個 <sup>2</sup>
ラベル	3987個	3987個	2045個
定数	8192個 <sup>2</sup>	8192個 <sup>2</sup>	2048個 <sup>2</sup>
NT/PT命令	8192個	8192個	2048個

- プログラム容量使用率  
現在編集中的ロジックプログラムの容量が、GLCで使用可能なメモリ領域のうち何パーセントを占めているかを示します。
- 変数のメモリ使用率  
現在登録されている変数の容量が、GLCで使用可能なメモリ領域のうち何パーセントを占めているかを示します。
- コントローラのメモリ使用率  
現在使用中の命令および行の容量が、GLCで使用可能なメモリ領域のうち何パーセントを占めるかを示します。

<sup>1</sup> リビジョンが「Rev.\*-無し、1」の機種です。リビジョンの見分け方について、[参照](#)  
「GLC2400/GLC2600をご使用のお客様へ」

<sup>2</sup> 整数型で作成した場合の最大個数です。

### 3.1.1 ロジックプログラム書き込み

Editorを使ってロジックプログラムを作成しエラーがない状態になったら、ロジックプログラム書き込み、GLCでの実行が可能になります。

GLCへのロジックプログラムの書き込みには以下の方法があります。

- ・GP-PRO/PB の[転送]ウィンドウで画面データとロジックプログラムを転送する。
- ・Editorでロジックプログラムのみを転送する。

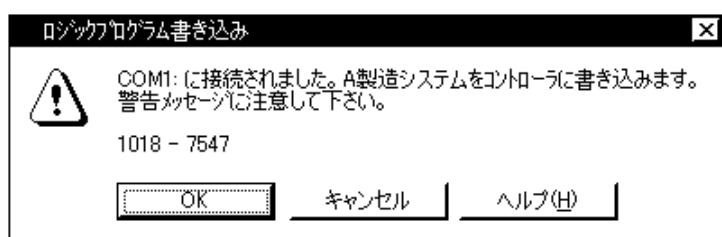
ただし、GLCへロジックプログラムを書き込む前に、GLCをセットアップしておく必要があります。Editorの[転送]ウィンドウで、プロジェクトファイルと共にシステムを転送することによりセットアップできます。転送についての詳細は、「GP-PRO/PB オペレーションマニュアル 第7章 画面を転送する」を参照してください。

ここではEditorでロジックプログラムのみを転送する方法について説明します。

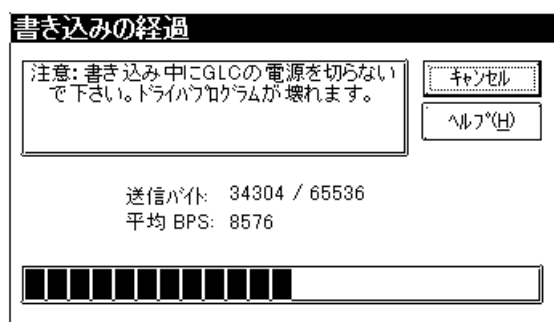
#### ロジックプログラム書き込み手順

1. [コントローラ]メニューで[ロジックプログラム書き込み]を選択します。[ロジックプログラム書き込み]ウィンドウが表示されます。コントローラへ書き込むようプロンプトで要求されます。

ロジックプログラム書き込み前に、Editorは自動的にエラーのチェックを実行します。プログラムにエラーがあると、ロジックプログラム書き込みはできません。



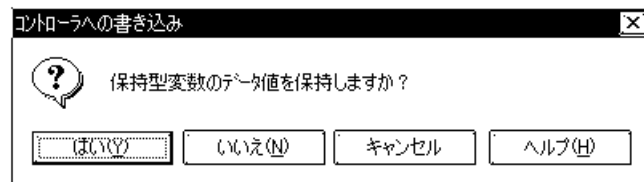
2. [OK]をクリックします。[ダウンロードの進行状況]ウィンドウが表示されます。GLCへのファイルの転送を開始し、進行状況が表示されます。



参考：PRW ファイルをコントローラに書き込むと、Flex Network ドライバなどの必要な I/O ドライバがダウンロードされます。ドライバに変更がない場合は、ドライバのダウンロードはスキップされます。

参考：転送する前にコメントを削除すれば、ダウンロードしたファイルのサイズを縮小することができます。参照「3.1 コントローラの設定」

参考：保持型変数を保持してダウンロードすることが可能です。オプションダイアログの機能タブにて設定した場合、以下のダイアログが表示されます。



「はい」をクリックすると、保持型変数の値を保持してダウンロードを行います。「いいえ」をクリックすると、変数値をすべてクリアしてダウンロードを行います。



- ・ 確認タブの「ロジックエディタの操作の確認あり」を選択していない場合は表示されません。

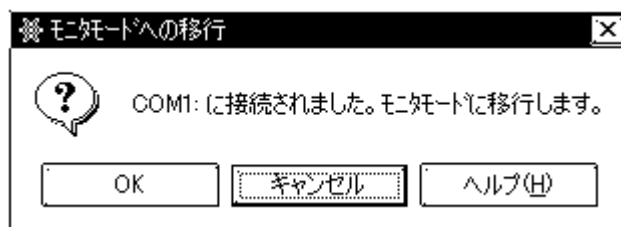
### 重要

- ・ GLC300/GLC2000 シリーズ / LT シリーズでは、ロジックプログラムの書き込みのあと GLC がリセットされます。
- ・ Administrator 権限でない場合、転送に失敗する恐れがあります。Administrator 権限でご使用ください。

## 3.1.2 モニタモードへの移行

### モニタモードへの移行手順

1. [コントローラ]メニューで、[モニタモードへの移行]を選択します。モニタモードへ移行するかどうかの確認のダイアログボックスが表示されます。



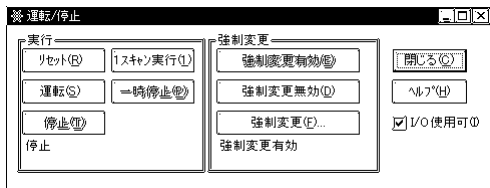
2. [OK]をクリックします。モニタモードでオンライン接続された GLC 上のプログラムの運転 / 停止ができます。

## 3.2 コントローラの運転 / 停止

モニタモードの場合、Editor側からコントローラの運転/停止を切り替えることができます。この時点でコントローラのモニタを実行し始めます。前述のように、コントローラへモニタモードで接続してから、運転/停止やモニタモードの機能を実行してください。

### コントローラの運転 / 停止手順

1. [コントローラ]メニューで、[運転 / 停止]を選択します。プログラムモードでは、この項目は使用できません。[運転 / 停止]ダイアログボックスが表示されます。

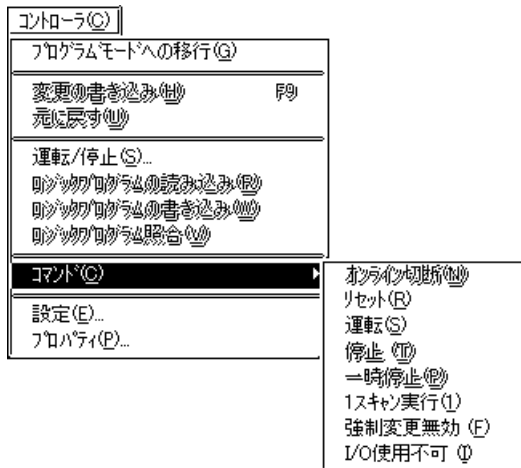


項目	内容
運転	[運転] ボタンをクリックするとコントローラによるプログラム実行が開始されます。最初にプログラムのスキャンが始まり、すべてのラダープログラムが順に実行されます。最初のスキャンで初期設定プログラムが実行されます。
停止	[停止] ボタンをクリックするとコントローラが終了します。
リセット	[リセット] ボタンをクリックすると、コントローラが実行ファイルを再ロードし、I/Oおよび変数を初期化します。
1スキャン実行	[1スキャン実行] ボタンをクリックすると、プログラムのスキャンが1回行われます。この機能はアプリケーションのトラブルシューティングやデバッグに有効です。
一時停止	[一時停止] ボタンをクリックすると、コントローラはプログラムをスキャンするのを一時停止しますが、I/Oリフレッシュは実行されません。
強制変更有効	強制変更を有効にします。
強制変更無効	強制変更を無効にします。
強制変更	ロジックプログラムで強制変更に登録したすべての変数をリストアップします。
I/O使用可	GLCの本体やI/Oユニットの外部I/Oへの入出力を可能にする動作です。通常ロジックプログラムのダウンロードをおこなった後、GLCを運転状態にただけでは外部I/Oの入出力をおこなうことができません。これは、操作やロジックプログラムのミスなどで、機械が突然動作することを防止するための安全を優先する考え方の機能です。

**重要**

- ・ 運転 / 停止の移行時に内部的に I/O 使用可のチェックをおこなっています。よって運転状態で「I/O使用可」を変更しても反映されません。一端、必ず停止にして「I/O使用可」の変更を行い、運転に戻してください。
- ・ I/O使用可の情報はSRAMに保存されます。したがってSRAMバックアップの電池切れで初期化されます。

[コントローラ(C)]メニューおよびツールバーより選択することもできます。



	接続
	モニタモードへの移行
	ロジックプログラム書き込み
	ロジックプログラム読み込み
	変更の書き込み (オンラインエディット時のみ使用)
	コントローラリセット
	運転
	停止
	一時停止
	1スキャン実行
	強制変更有効
	I/O使用可

- 重要** ・ コントローラリセットにより、保持型の変数はリセットされます。特別な初期設定が必要な場合は、ロジックプログラム上でMOV命令などを使って初期値を設定してください。

### 3.3 システム変数によるプログラムのトラブルシューティング

コントローラの起動後にアプリケーションがうまく動作しない場合、システム変数を使用してアプリケーションのトラブルシューティングを実行することができます。システム変数 #FaultCode、#FaultRung、#IOFault、#IOStatus、#ScanCount は、コントローラが I/O に発生するトラブルの検出に最も有効です。

項目	内容
#FaultCode	最新のエラー状況を示します。立ち上がり一回目のスキャン実行時に、「0」にリセットされます。
#FaultRung	エラーが発生した行の番号が表示されます。
#IOFault	I/Oドライバでエラーが検出されるとONになります。
#IOStatus	I/O固有のエラーを表示する配列です。これらのエラーには、ドライバごとに異なるエラーコードがつけられます。エラーの詳細については、ドライバのオンラインヘルプを参照してください。#IOFaultがONになった場合のみ、エラーが#IOStatusで表示されます。
#ScanCount	コントローラが起動してから実行したスキャン回数を示します。この変数は、モニターを開始してから常に増加しています。増加していない場合は、コントローラが動作していません。

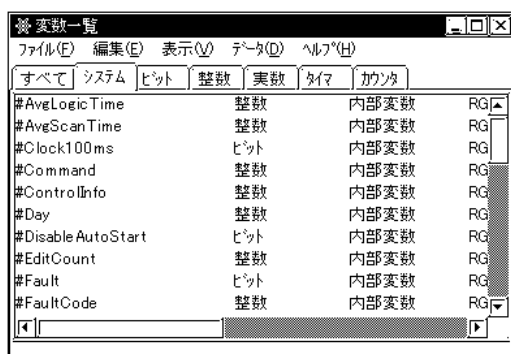
システム変数の詳細については、「Pro-Control Editor ユーザーズマニュアル 第3章 システム変数」を参照してください。

### 3.4 システム変数の表示

システム変数を表示させて、I/Oの状態、スキャン時間、コントローラの状態に関する情報を見ることができます。参照「Pro-Control Editor ユーザーズマニュアル 第3章 システム変数」

#### システム変数の表示手順

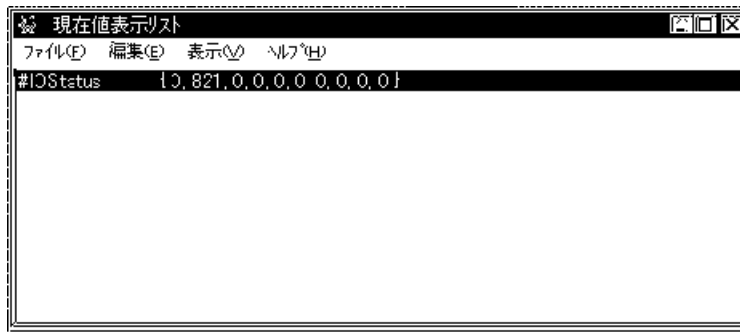
1. [データ]メニューで[変数一覧]を選択します。[変数一覧]ウィンドウが表示されます。このウィンドウにはPro-control Editorのシステム変数(#で始まる変数)がすべて表示されます。表示されない場合は、[表示]メニューで[システム]を選択してください。





2. [データ]メニューで[現在値表示リスト]を選択すると、[現在値表示リスト]ウィンドウが表示されます。
3. モニタするシステム変数をクリックして、[変数一覧]ウィンドウから[現在値表示リスト]ウィンドウへドラッグします。

ロジックプログラムのスキャン中にエラーが発生した場合は、モニターした変数が当該のエラーを表示します。以下は、ドライバ1でI/Oエラー821が発生した例です。



## 3.5 ロジックプログラム読み込み

GLC本体にあるロジックプログラムの編集や保存を行いたい場合はコントローラから読み込みます。

GLCからロジックプログラムを読み出すには以下の方法があります。

- ・ GP-PRO/PB の[転送]ウィンドウで画面データとロジックプログラムを受信する。
- ・ Editor でロジックプログラムのみを受信する。

ここではEditorでロジックプログラムのみを転送する方法について説明します。

GP-PRO/PB の[転送]ウィンドウで受信する方法については、「GP-PRO/PB オペレーションマニュアル 第7章 画面を転送する」を参照してください。

### コントローラからの読み込み手順

1. コントローラがモニタモードで接続されている場合は、[コントローラ]メニューで[プログラムモードへの移行]を選択します。

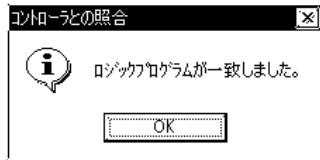
**重要** ・ プログラムに初期化していない値が含まれている場合、プログラムから読み込む前に、コントローラを終了してください。

2. [コントローラ]メニューで、[ロジックプログラム読み込み]を選択します。コントローラに書き込まれたプログラムがEditorで開かれます。これでプログラムに変更を加えたり、「\*.prw」ファイルとして保存できるようになります。

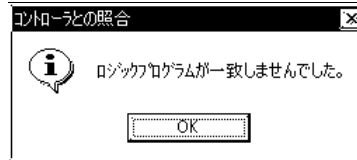
## 3.6 ロジックプログラム照合

Editorで編集中のプログラムと、現在動作中のプログラムを照合します。照合結果は以下のダイアログにて表示します。

照合結果が一致した場合



照合結果が不一致の場合

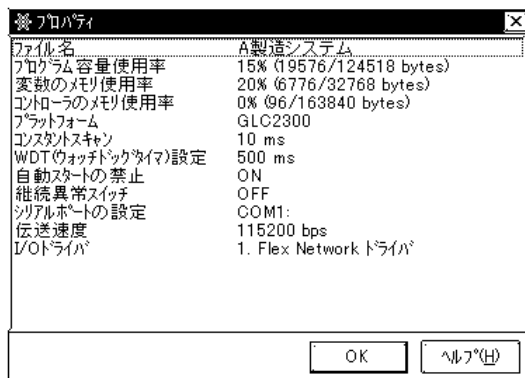


- ・ 変数の値、コメントは照合されません。
- ・ プロジェクト名が異なる場合、同じプログラムでも一致しません。

## 3.7 プロパティ

コントローラメニューの[プロパティ]を選択するとGLCのプログラム情報がリストボックスで表示されます。

以下はプログラム情報のプロパティボックスです。



# 第4章

# モニタモードでの 動作確認

コントローラで実行しているプログラムにモニタモードで変更を加え、その変更内容をすぐに実施することができます。

本章での解説や例示では、C:\Program Files\Pro-face\ProPBWin\SampleにあるSoda.prwを使用します。このファイルでは、ラダーのカラーやオプションをシステムのデフォルト値に設定しているものとしします。

モニタモード移行時、本体のロジックプログラムと保存データが一致しないとアップロードします。そのため、照合機能にて一致した場合でも必ずデータを保存してください。

## 4.1 編集を始める前に

### ロジックプログラムの実行手順

1. Soda.prwを開きます。このファイルはサンプルプログラムとして、Editorに添付されています。  
デフォルト設定のパスはC:\Program Files\Pro-face\ProPBWin\Sampleです。
2. コントローラにプログラムを書き込みます。
3. コントローラをモニタモード接続します。
4. コントローラを運転します。コントローラの操作の詳細は、[参照](#)「第3章 ロジックプログラムを実行する」

### モニタモードでGLCに加えられるプログラムの変更

コントローラにモニタモードで接続中、以下の変更をプログラムに加えることができます。

- ・変数の状態の変更
- ・変数の値の変更

## 4.2 モニタモード編集にカラーを使用する

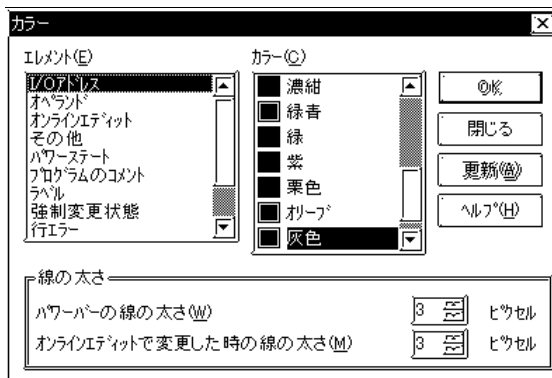
Editor では、デフォルト設定のカラーを使用して、特定の状態を示したり、モニタモードの動作中にロジックプログラムに変更を加えることができます。

デフォルト設定のカラーは以下のとおりです：

項目	内容
緑	回路が導通しています。
赤	行にエラーが発生していることを示します。
紫	オンラインエディットしていることを示します。

Editor のカラーのデフォルト設定の変更手順

1. [表示]メニューで[カラー]を選択します。[カラー]ダイアログボックスが表示されます。



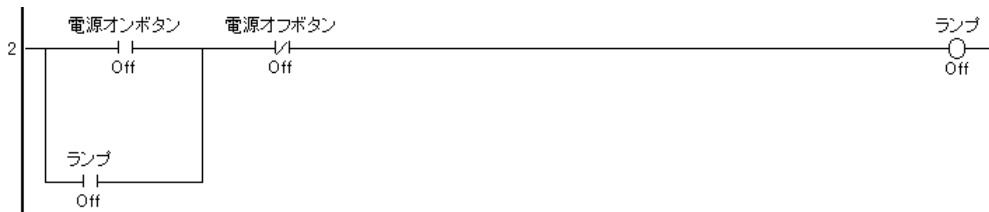
2. 変更したい[エレメント]と[カラー]を選択して、[更新]をクリックします。

## 4.3 ビット変数を ON/OFF する

ロジックプログラムの実行中は、ビット変数を手動で ON / OFF することができます。強制 ON にした場合は ON 状態が保持されますが、手動でビットを ON にした場合スキャンが行われると、ビット変数の状態がプログラムの影響を受けるため、プログラムに依存します。

### ビット変数の ON / OFF 方法

1. 2 行目の出力コイルに割り付けた変数「ランプ」をマウス右ボタンでクリックします。
2. ショートカットメニューで[ON にする]を選択します。変数「ランプ」が ON になり、行が導通します。



3. 2 行目の出力コイルに割り付けた変数「ランプ」をマウス右ボタンでクリックします。
4. ショートカットメニューで[OFF にする]を選択します。「ランプ」変数が OFF になり、導通状態はクリアされます。

参考 : [オプション]ダイアログボックスの[モニタ]タブで「パワーフロー」が選択されていないと、導通状態はロジックプログラムに表示されません。

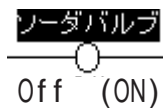
参照 「第2章 ロジックプログラムを作成する前にオプション画面で設定を行う」

## 4.4 ビット変数を強制的に ON/OFF する

コントローラにモニタモードで接続している間は、ビット変数を強制的に ON/OFF することができます。ビット変数を強制的に ON/OFF すると、強制 ON/OFF を手動で解除するまでは状態が保持されます。整数変数、文字型変数のビット指定の状態は変更できません。4.3 のビット変数の ON/OFF の場合は、プログラムの演算結果に依存しましたが、強制 ON/OFF の場合は、プログラムの演算結果に依存しません。

ビット変数の強制的な ON / OFF 方法

- 1.9 行目の出力コイルの変数「ソーダバルブ」を右クリックします。
2. ショートカットメニューで[強制 ON]を選択します。
3. [強制変更]ウィンドウで[OK]をクリックします。



変数が ON となり、プログラムでは OFF になりません。

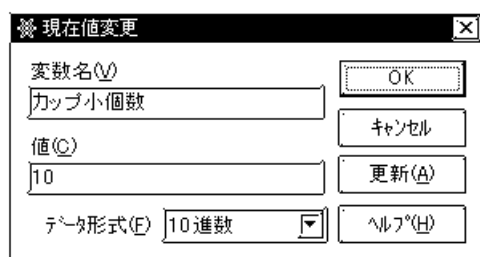
参考：強制 ON/OFF を指定したのにロジックプログラムに反映されない場合は、強制変更無効の設定になっています。有効にするには、[運転 / 停止]ダイアログボックスで[強制変更有効]ボタンをクリックするか[コントローラ]メニューおよびツールバーより切り替えることができます。デフォルトは、強制変更有効となっています。

## 4.5 変数値の変更

コントローラにモニタモードで接続している間は、ロジックプログラムに含まれる変数の値を設定することができます。

変数値の変更手順

1. [データ]メニューで[現在値変更]を選択すると、[現在値変更]ダイアログボックスが表示されます。
2. ロジックプログラムの変数「カップ小個数」をクリックします。下のような[現在値変更]ダイアログボックスが表示されます。



3. [値]フィールドを選択してから、5を入力してください。
4. [更新]をクリックします。「カップ小個数」の値が5になりました。他の値を変更するか、または[閉じる]をクリックして[現在値変更]ウィンドウを閉じます。



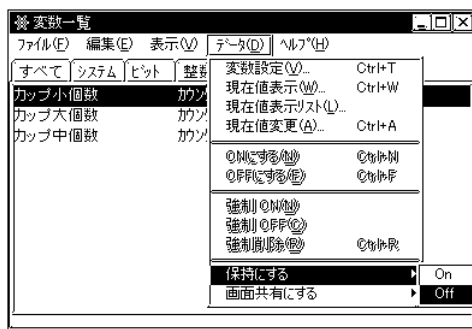
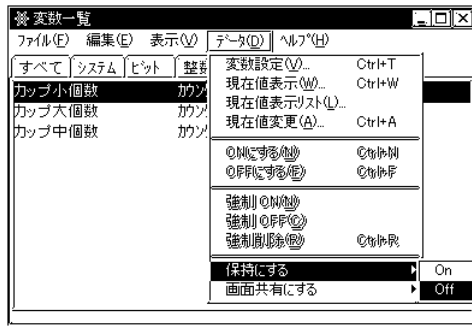
- ・ 現在値は、10進数、16進数、8進数、2進数のいずれかのフォーマットで入力できます。[現在値表示リスト]で選択することもできます。
- ・ [現在値変更]ダイアログボックスとともに、[変数一覧]ウィンドウか[現在値表示リスト]ウィンドウを使用すれば、変数をすばやく検索して設定することができます。

## 4.6 変数の属性変更

変数一覧の[データ]メニューに変数の保持、画面共有変数の属性を変更するメニューがあります。このメニューはプログラムモードのみ有効となります。

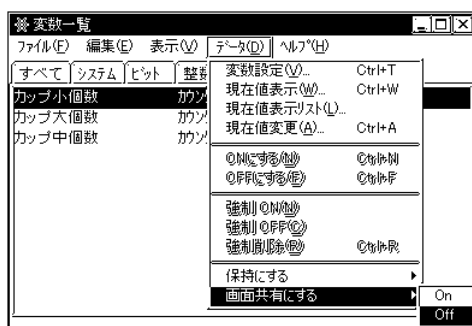
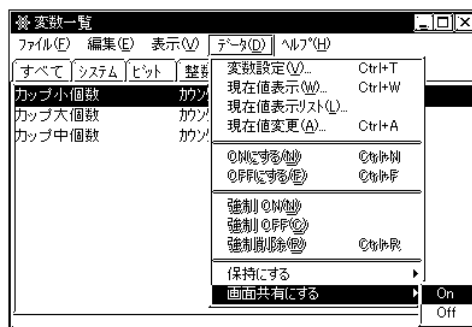
### 変数(保持)の属性変更手順

1. [データ]メニューで[変数一覧]を選択すると、[変数一覧]ウィンドウが表示されます。変更したい変数を選択し、属性を変更します。システム変数の保持は変更できません。



### 変数(画面共有変数)の属性変更手順

1. [データ]メニューで[変数一覧]を選択すると、[変数一覧]ウィンドウが表示されます。変更したい変数を選択し、属性を変更します。



## 4.7 現在値表示リスト

### 表示モード一括変更

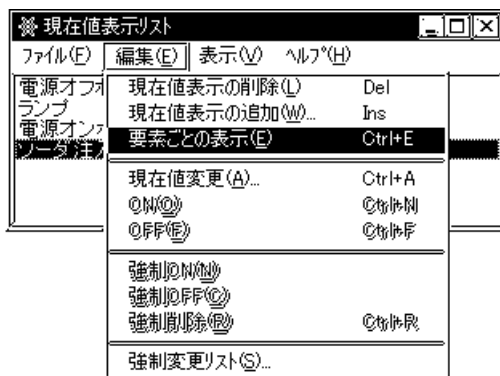
[データ]メニューから[現在値表示リスト]を選択します。選択した変数の表示モードを一括で変更が可能です。



### 配列の要素ごとの表示

現在値表示リストで配列の場合は、[要素ごとの表示]で配列タイマ、カウンタの値を要素ごとに表示します。

1. [データ]メニューで[変数一覧]を選択し、[データ]メニューから[現在値表示リスト]を選択します。
2. [現在値表示リスト]メニューから[要素ごとの表示]を選択します。





## 4.8 オンラインエディット（対応機種：GLC2000シリーズ）

モニタモードにおいてロジックプログラムの実行中にロジックプログラムを変更することができます。

オンラインエディットでは、編集機能として6種類の編集ができます。

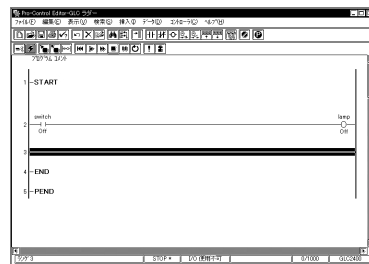
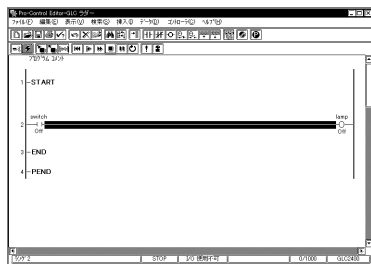
1. 行の追加
2. 行の置換
3. 行の削除
4. ラベルの追加
5. サブルーチンの追加
6. 変数の追加

### 4.8.1 オンラインエディットの編集機能

#### 行の追加

指定行間に1行追加します。

[挿入]メニューの[行]を選択します。

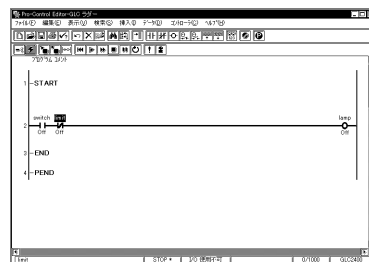


この時、変数追加をした場合は変数追加命令も同時に行われます。

#### 行の置換

既存行の行を編集します。

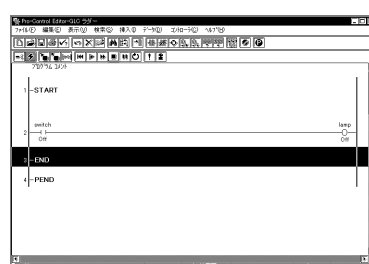
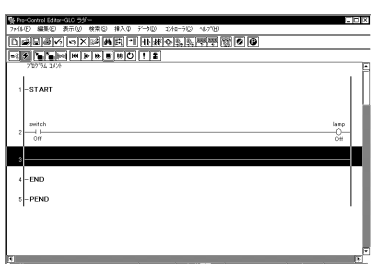
命令の挿入、置換、削除ができます。



この時、変数追加をした場合は変数追加命令も同時に行われます。

#### 行の削除

選択した行を削除します。

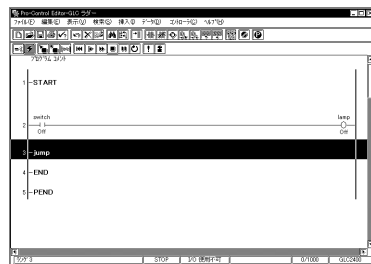
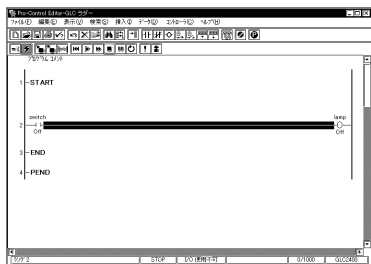


この時、変数は削除されません。

### ラベルの追加

ラベルを追加します。

[挿入]メニューの[ラベル]を選択します。

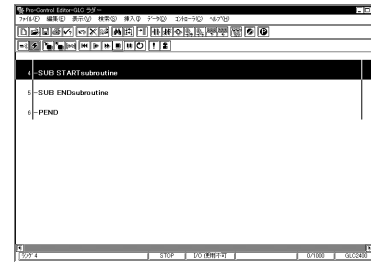


### サブルーチンの追加

サブルーチンを追加します。

サブルーチンは、ENDラベルとPENDラベルの間に挿入されます。

[挿入]メニューの[サブルーチン]を選択します。



### 変数の追加

新規の変数を追加します。

[データ]メニューの[変数設定]または命令の挿入時に追加することができます。

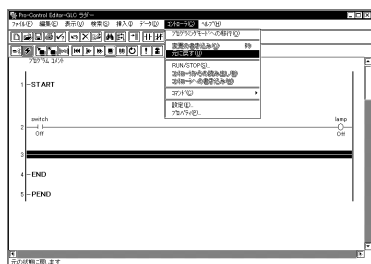
### 編集ロジックプログラムの書き込み

[コントローラ]メニューの[変更の書き込み]で編集したロジックプログラムをGLCに書き込みます。または編集後、他の行を編集すると書き込まれます。



### 編集ロジックプログラムを元に戻す

直前に編集したロジックプログラムを行単位で元に戻します。



## 4.8.2 データ保存

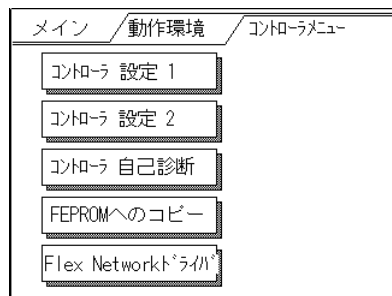
ロジックプログラムは、Editor で作成後、[ ロジックプログラム書き込み ] で一旦 FEPRM に書き込まれます。書き込み後最初の GLC 起動時に、FEPRM の内容を SRAM にコピーします。オンラインエディットでは、この SRAM に保存されているロジックプログラムを編集します。オンラインエディット後 GLC を再起動した場合、通常 GLC は SRAM に保存されているデータを読み込みますが、SRAM に保存されたデータは電源 OFF 時のバッテリー切れ<sup>1</sup>などにより失われることがあります。その場合、次回 GLC 起動時には FEPRM に保存されているロジックプログラムが読み出されます。

SRAM のデータが失われた場合に備え、GLC のオフラインメニューで [ FEPRM へのコピー ] を実行するか、Editor で PRW ファイルとして保存することにより、バックアップしてください。

### FEPRM へのコピー

GLC のオフラインメニュー [ FEPRM へのコピー ] を選択します。

注意：オフラインメニューへ移行すると、GLC のロジックプログラムや表示機能は停止し、その後、初期状態から起動します。



FEPRM へ編集したロジックプログラムをコピーすることで、SRAM に保存されているロジックプログラムが失われても FEPRM からロジックプログラムを読み込むことでシステムは継続して運転が可能となります。

- 重要**
- FEPRM へのコピーを実行していない状態で GLC を再起動すると “No Backup logic program in FEPRM” という警告メッセージが表示されます。通常は SRAM に保存されているデータを読み込みますが、SRAM のデータが失われた場合は、FEPRM に保存されているオンラインエディットによる編集前のデータを読み込みます。SRAM のデータが失われた場合に備えて、FEPRM へのコピーを必ず実行してください。
  - SRAM のデータが失われたとき、FEPRM からロジックプログラムの読み込みは自動的に行われます。ただし、この状態のときは、マイナー異常のエラーが発生しますので自動的に FEPRM のロジックプログラムで実行すると問題があるシステムでは [ マイナー異常時の運転 ] を選択してロジックプログラムが自動的に実行しないように設定してください。

<sup>1</sup> リチウム電池の寿命は電池周囲温度 40 以下で 10 年以上、50 以下で 4.1 年以上、60 以下で 1.5 年となります。バックアップ期間は初期状態 (満充電) で約 60 日、電池寿命時で約 6 日です。

### Editor での保存

Editor でオンラインエディット終了後、プログラムモードに移って、編集したロジックプログラムを [保存] することで、PRW ファイルとして保存することができます。PRW ファイルとして保存されているロジックプログラムを GLC にダウンロードすることで編集後のロジックプログラムを実行することができます。

# 第5章

## Editor と GP-PRO/PB

GP-PRO/PB では、Editorで作成した変数とリンクした操作画面を作成して、コントローラや稼動しているロジックプログラムを操作したりモニタリングをすることができます。本章では、GP-PRO/PB でのGLC用画面の作成方法について説明します。本章は簡単なチュートリアル形態になっており、タンクからの水をポンプで汲み上げるために設計したロジックプログラムとリンクした画面をGP-PRO/PB で作成します。

## 5.1 GP-PRO/PB での変数表示

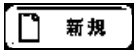
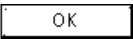
GLC用の変数をGP-PRO/PB でインポートする方法について説明します。

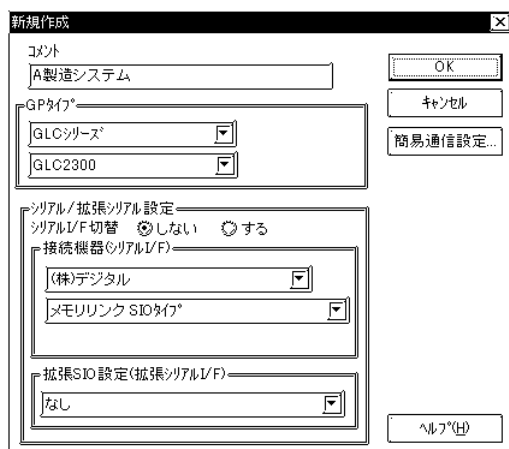
### 5.1.1 Editor の起動手順

1. [スタート]ボタンをクリックして、[プログラム(P)] [Pro-face] [ProPB3 C-Package]の順にポイントし、[プロジェクトマネージャ]をクリックします。
2. プロジェクトマネージャが起動します。

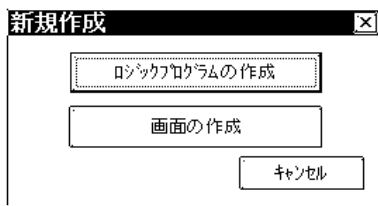


#### 新規プロジェクトの作成

1. プロジェクトマネージャで[プロジェクト(P)]から[新規作成(N)]を選択するか、 **新規** をクリックします。
2. コメント、GPタイプ、接続機器、シリアル I/F 切替、拡張 SIO 設定を設定し、 **OK** で実行します。



3. ロジックプログラムまたは画面を作成するかどうかの問い合わせがあります。



ロジックプログラムの作成 : Editor が起動します。

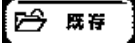
画面の作成 : GP-PRO/PB が起動します。

キャンセル : プロジェクトマネージャに戻ります。



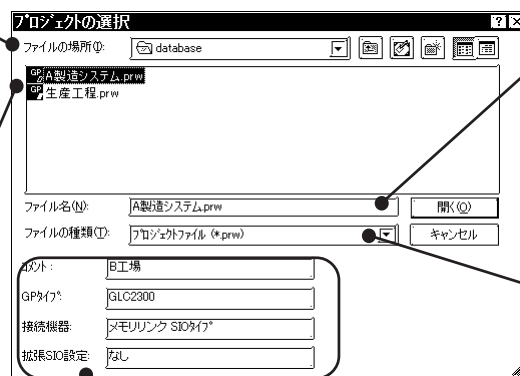
- プロジェクトファイルを新規作成後、保存をせずに別プロジェクトファイルを作成または選択しようとする、保存するかどうかの問い合わせがあります。 **はい(Y)** をクリックすると、[名前を付けて保存]のダイアログボックスが表示されます。 **いいえ(N)** をクリックすると保存せずに画面を開く操作に移ります。

### 既存プロジェクトから選択

プロジェクトマネージャで[プロジェクト(P)]から[選択(S)]を選択するか、 をクリックします。

目的のプロジェクトファイルのあるフォルダを選択します。

現在のフォルダや既存プロジェクト名がリスト表示されます。




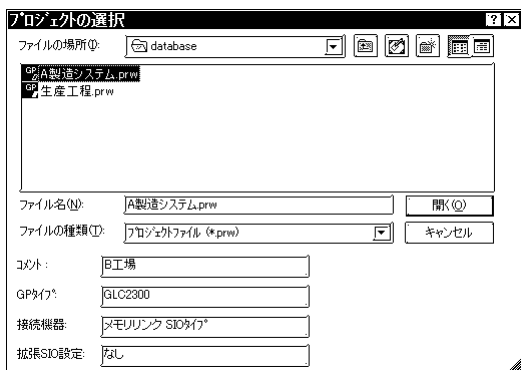
リストから選択したプロジェクトファイル名が表示されます。プロジェクトファイル名を入力して指定することもできます。

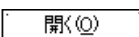
目的のファイルの種類を選択します。

選択されているプロジェクトファイルに付けられたコメント、選択されているGPタイプと接続機器、拡張 SIO が表示されます。

参考 : 外部機器 (PLC、温調器、インバータなど) を接続しない場合は、「メモリリンク SIO タイプ」を選択します。

1. プロジェクトマネージャで[プロジェクト(P)]から[選択(S)]を選択するか、 をクリックします。
2. プロジェクトリストから選択します。または、プロジェクトファイル名を入力します。



3.  で実行します。

### ロジックプログラムの起動アイコン

作成 : Editor でロジックプログラムを作成します。

モニタ : ロジックプログラムをモニタリングします。

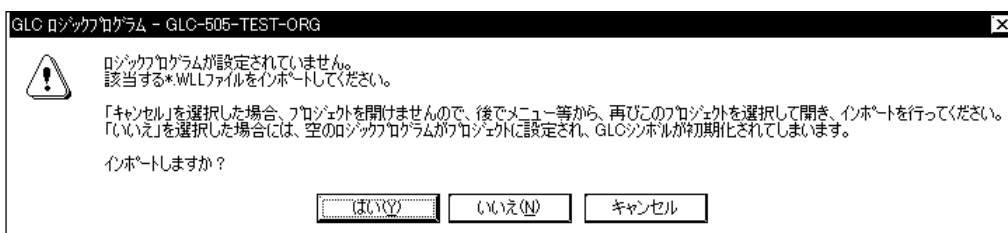
I/O : I/O 設定を行います。

変数 : 変数一覧を表示します。



- ・ GPタイプがGLC/LTシリーズの場合のみロジックプログラムの各アイコンが有効になります。

旧バージョンで作成されたプロジェクトファイルを選択する際に、一旦 WLL ファイルをインポートしているプロジェクトを選択すると、再度WLLファイルを割り付ける作業が必要になります。その際、以下のダイアログボックスが表示されます。





## 5.1.2 命令データの貼り付け

はじめに作画で使用する部品やタグのアドレスに変数を割り付けるために、作成したロジックプログラムを一度保存してください。保存することにより、変数はGP-PRO/PB にインポートされます。

Editorで作成したロジックプログラムの命令をコピーしてGP-PRO/PB 上に貼り付けることにより、その命令に該当する部品を配置できます。

また逆に、画面に配置した部品をコピーしてロジックプログラム上に貼り付けることにより、その部品に該当する命令を挿入できます。

### 命令 - 部品間の変換

命令と部品はそれぞれ該当する種類が決まっています。

### 命令から部品への変換

各命令が変換される部品の種類は以下のとおりです。

命令	→	部品
NO(a接点)		ビットスイッチ
NC(b接点)		ビットスイッチ
PT(立ち上がり接点)		ビットスイッチ
NT(立ち下がり接点)		ビットスイッチ
OUT/M(アウトコイル)		ランプ
NEG/NM(反転コイル)		ランプ
SET/SM(セットコイル)		ランプ
RST/RM(リセットコイル)		ランプ
CTU(アップカウンタ)		数値表示器/グラフ/設定表示器
CTD(ダウンカウンタ)		数値表示器/グラフ/設定表示器
CTUD(アップダウンカウンタ)		数値表示器/グラフ/設定表示器
TON(オンディレータイマ)		設定表示器
TOF(オフディレータイマ)		設定表示器
TP(パルスタイマ)		設定表示器

部品から命令への変換

各部品が変換される命令の種類は以下のとおりです。

部品	命令
ビットスイッチ	NO(a接点) NC(b接点) PT(立ち上がり接点) NT(立ち下がり接点)
ランプ	NO(a接点) NC(b接点) PT(立ち上がり接点) NT(立ち下がり接点) OUT/M(アウトコイル) NEG/NM(反転コイル) SET/SM(セットコイル) RST/RM(リセットコイル)
数値表示器/グラフ/設定表示器	CTU(アップカウンタ) CTD(ダウンカウンタ) CTUD(アップダウンカウンタ)
設定表示器	TON(オンディレータイマ) TOF(オフディレータイマ) TP(パルスタイマ)

ロジックプログラムの命令を画面に貼り付ける

ロジックプログラムの命令をコピーし、画面に貼り付けます。貼り付け時に、変換する部品の種類をリストから選択します。

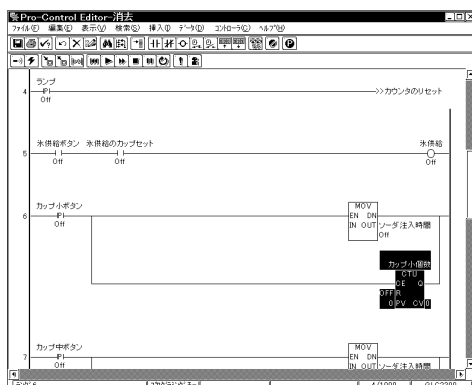
**重要**

- 命令をコピーする前に命令に変数を割り付けておく必要があります。変数の割り付けられていない命令を画面に貼り付けることはできません。
- 命令を貼り付ける前に、必ずロジックプログラムを保存しておいてください。

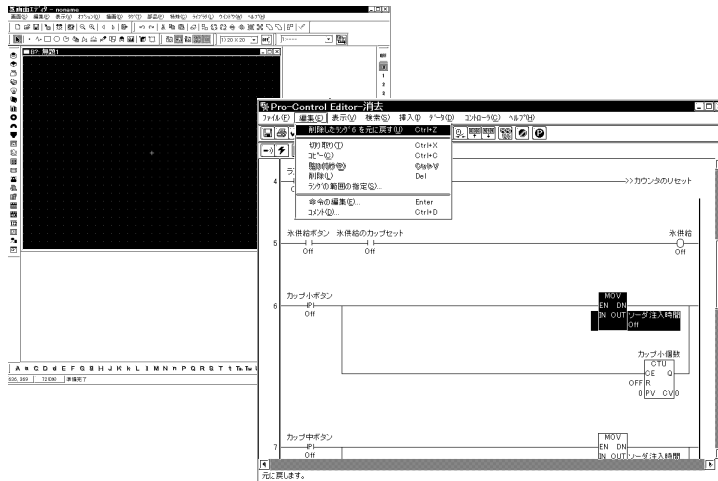


画面に命令を貼り付けた後にロジックプログラム側で命令を変更しても、既に貼り付けた命令には反映されません。

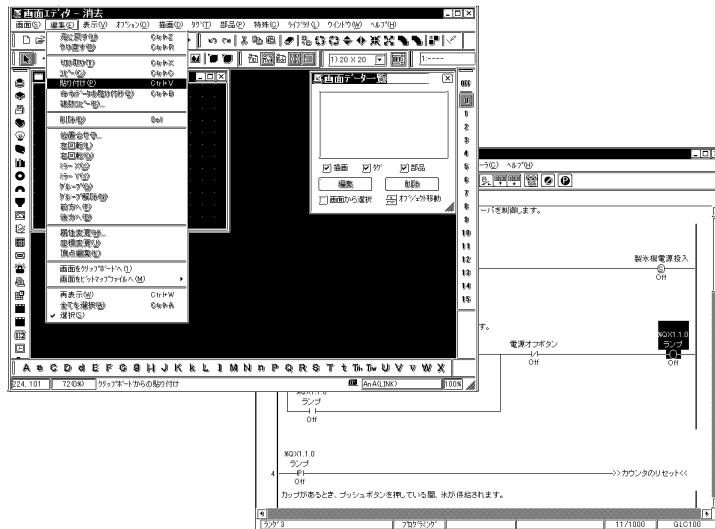
1. Editor で目的の命令を選択します。



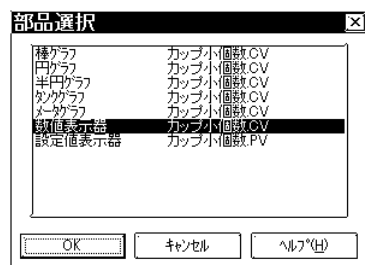
2. Editor で[編集(E)]から[コピー(C)]を選択します。選択した命令がクリップボードにコピーされます。



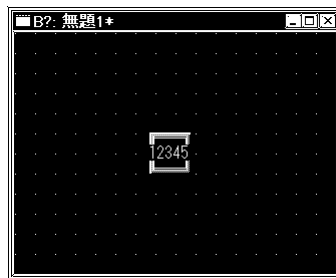
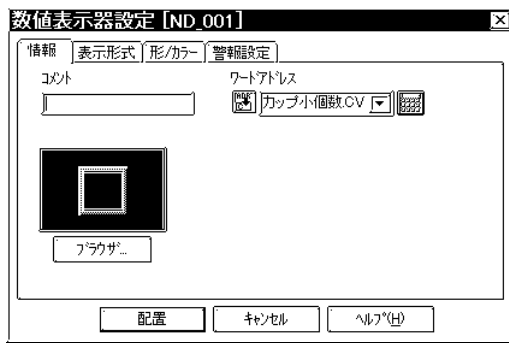
3. GP-PRO/PB で[編集(E)]から[貼り付け(P)]を選択します。



4. 変換する部品を選択し、 で確定します。コピーした命令に応じて変換する部品がリスト表示されます。変換できる部品が固定の場合は、表示されません。



5. GP-PRO/PB で[編集(E)]から[貼り付け(P)]を選択します。



### 画面に配置した部品をロジックプログラムに貼り付ける

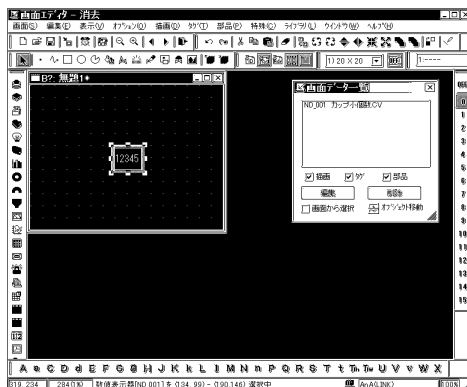
画面に配置した部品をコピーし、ロジックプログラムに貼り付けます。貼り付け時に、変換する命令の種類をリストから選択します。

**重要** ・ 部品をコピーする前に部品に変数( GLCシンボル )を割り付けておく必要があります。変数の割り付けられていない部品をロジックプログラムに貼り付けることはできません。

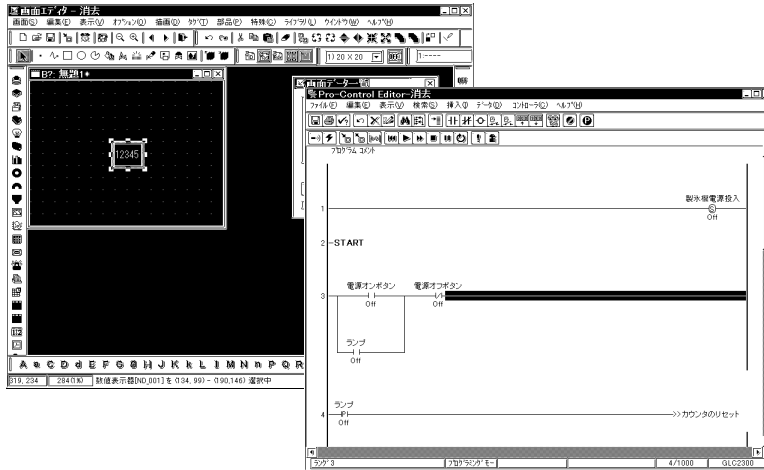


**MEMO** ・ 変数はロジックプログラムを保存することによりシンボルエディタに GLCシンボルとして登録されます。  
 ・ ロジックプログラム側に命令を貼り付けた後に画面側で命令を変更しても、既に貼り付けた命令には反映されません。

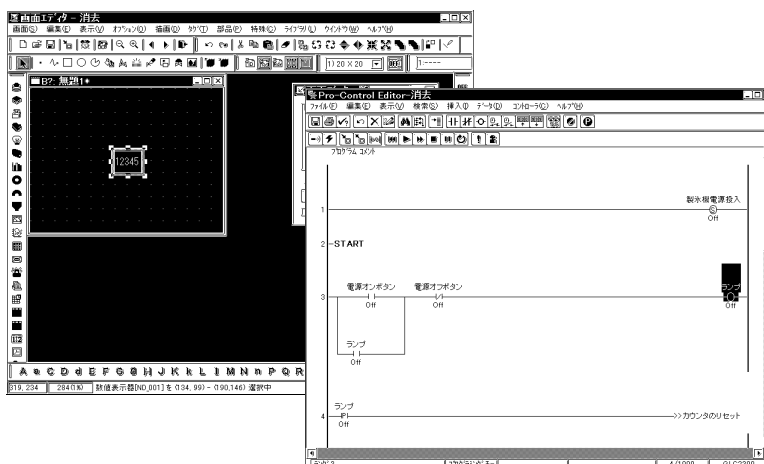
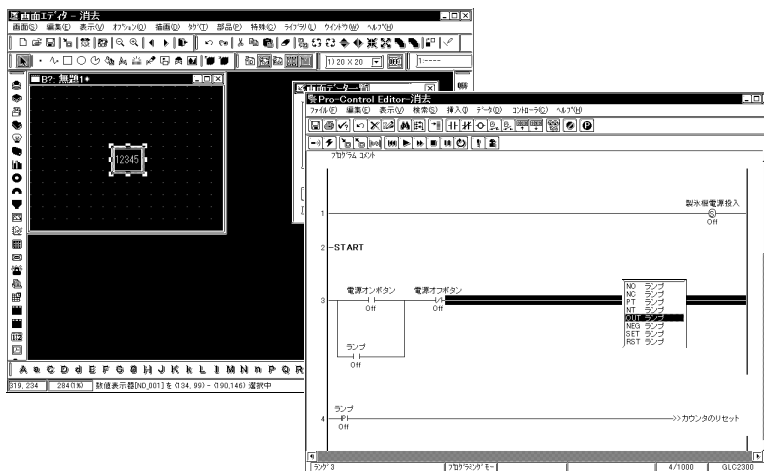
1. GP-PRO/PB で目的の部品を選択します。



- GP-PRO/PB で[編集(E)]から[コピー(C)]を選択します。選択した部品がクリップボードにコピーされます。
- Editor で命令を挿入したい行を選択し、[編集(E)]から[貼り付け(P)]を選択します。



- 変換する命令を選択し、ダブルクリックで確定します。コピーした部品に応じて変換する命令がリスト表示されます。



### ドラッグ&ドロップによる操作

ドラッグ&ドロップで命令データのコピーおよび貼り付け、または部品のコピーおよび貼り付けを行うことができます。

- 重要** ・ ドラッグ&ドロップする前に命令または部品に変数を割り付けておく必要があります。変数の割り付けられていない命令や部品はドラッグ&ドロップできません。

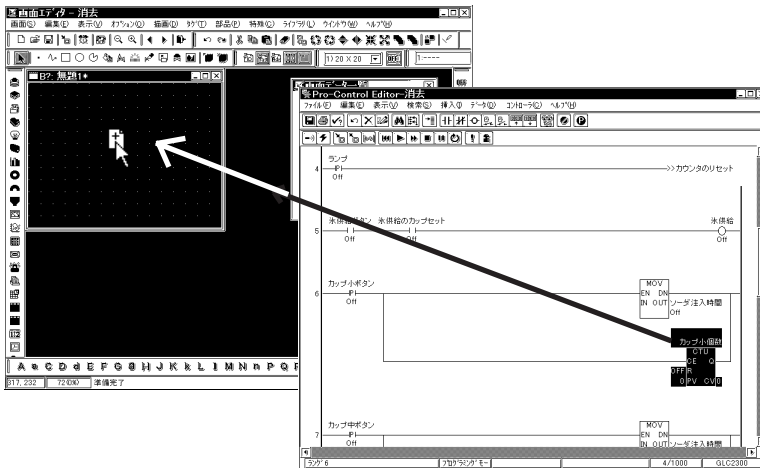
### ロジックプログラムの命令から部品へのドラッグ&ドロップ

Editorで作成したロジックプログラムの命令をGP-PRO/PB 上にドラッグすることにより、その命令に該当する部品を配置できます。

- 重要** ・ ドラッグ&ドロップする前に、必ずロジックプログラムを保存しておいてください。



- ・ 画面に命令を貼り付けた後にロジックプログラム側で命令を変更しても、既に貼り付けた命令には反映されません。

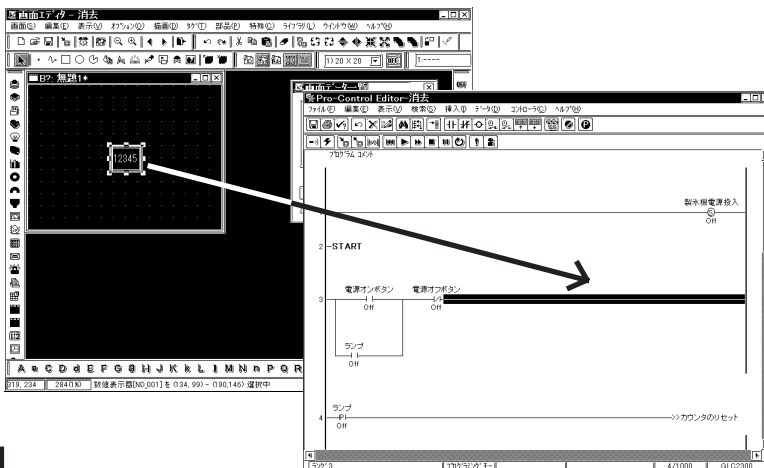


### 部品から命令へのドラッグ&ドロップ

画面に配置した部品をロジックプログラム上にドラッグすることにより、その部品に該当する命令を挿入できます。**[Ctrl]** キーを押しながらドラッグします。



- ・ ロジックプログラム側に命令を貼り付けた後に画面側で命令を変更しても、既に貼り付けた命令には反映されません。

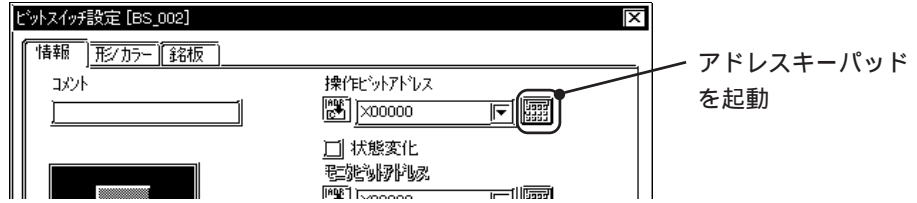


アドレスキーパッドから入力する

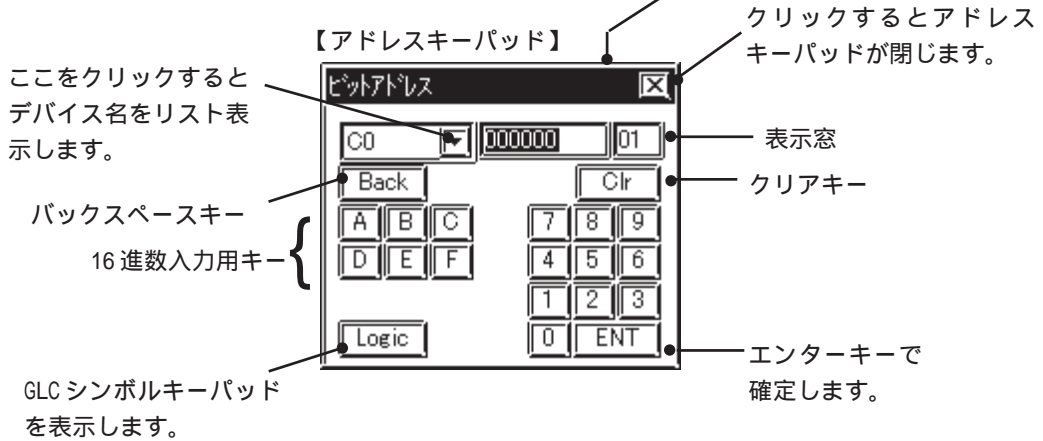
Editor では整数変数に修飾語(.X[m])をつけることでビット指定ができます。同様にGP-PRO/PB でも、インポートした整数変数に修飾語(.X[m])を付けることでビット指定できます。

参照 「Pro-Control Editor ユーザーズマニュアル 2.3 配列変数へのアクセス」

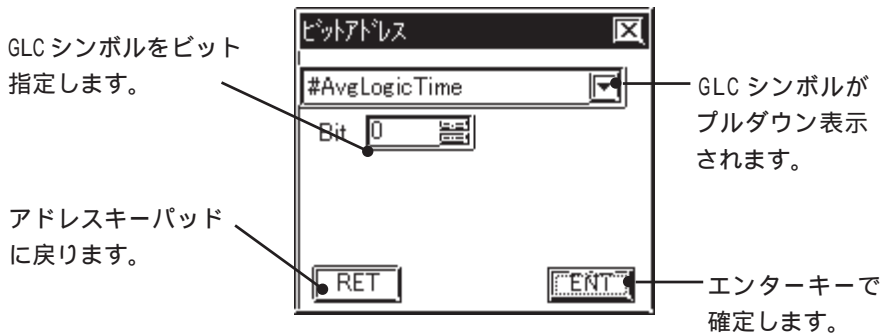
ロジックプログラムの変数 (GLCシンボル) をビット指定する場合は、アドレスキーパッドから GLC シンボルキーパッドを表示して指定します。



タイトルバーをマウスでドラッグすると、アドレスキーパッドを任意の位置へ移動できます。



【GLCシンボルキーパッド】



MEMO ・ Editor で整数変数のビット指定を使用する / しないに関わらず、GP-PRO/PB には通常の整数変数のみインポートされます。GP-PRO/PB で整数変数にビット単位でアクセスしたい場合は、GP-PRO/PB でビット指定します。

## 変数の制限事項

GP-PRO/PB で GLC の変数を扱う場合、以下のような制限があります。

- 通常のシンボルのエクスポートでは GLC シンボルは出力しません。
- 通常のシンボルをコピーし、貼り付けした時に、コントローラ変数内に存在する変数は設定できません。
- 通常のシンボルの入力では GLC シンボル内に存在する変数は設定できません。
- GP タイプの変更で GLC から GLC 以外の GP に変更された場合、GLC シンボルが設定されている時、GLC シンボルは通常変数に変更され自動的に割り当てられたアドレスは全て削除され、GLC シンボルを設定していた画面は、自動的に転送準備が必要な状態となります。GLC シンボルの割り付けを見直してください。
- GLC シンボルを含む画面をシミュレーションした場合、シミュレーション画面のデバイス情報には GLC シンボルを設定したデバイスは表示されません。
- GLC シリーズでは、Editor の変数にデバイスタイプがないので GP-PRO/PB 機能の E タグ、K タグの間接指定のデバイスタイプ & アドレスの指定はできません。
- GLC の変数は 32bit デバイスの Low/High 順で扱われます。
- GP-PRO/PB では GLC の配列の '[' ] ' を '<>' で表します。
- GP-PRO/PB で扱える変数の数は 2048 個です。配列の 1 要素を 1 つの変数として扱われます。画面共有変数が 2048 個を越える場合、GP-PRO/PB のタグ/部品で使用しない変数は非画面共有変数にしてください。

### 5.1.3 「ポンプチュートリアル」による作画例

ポンプチュートリアルは、タンクから水をポンプで汲み上げるために設計したロジックプログラムとリンクした画面を GP-PRO/PB で作成します。

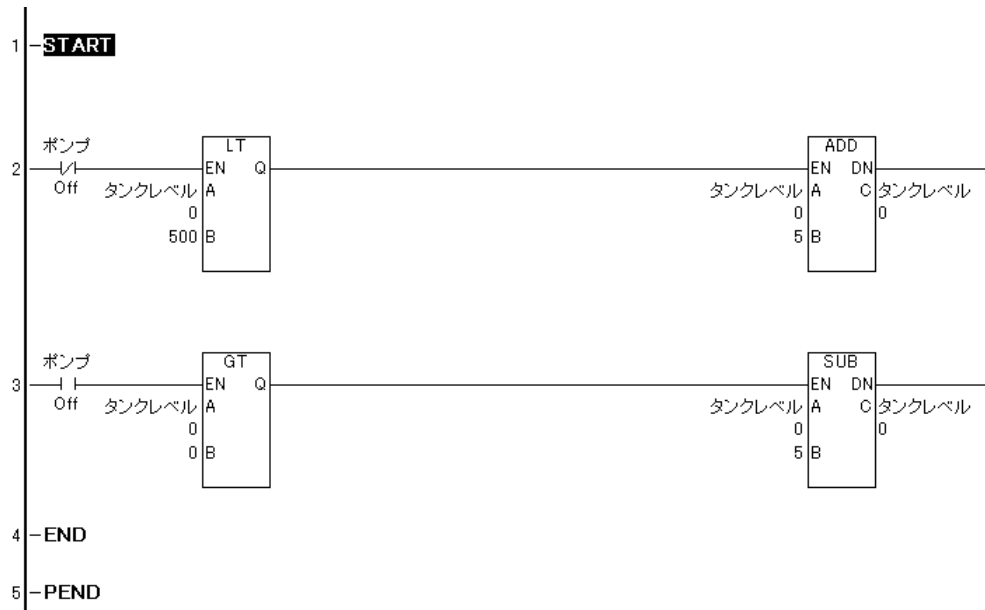
#### Editor の起動

1. プロジェクトマネージャから [ロジックプログラム/作成] をクリックします。また、[作画/画面] をクリックして、Editor と GP-PRO/PB の両画面を開きます。

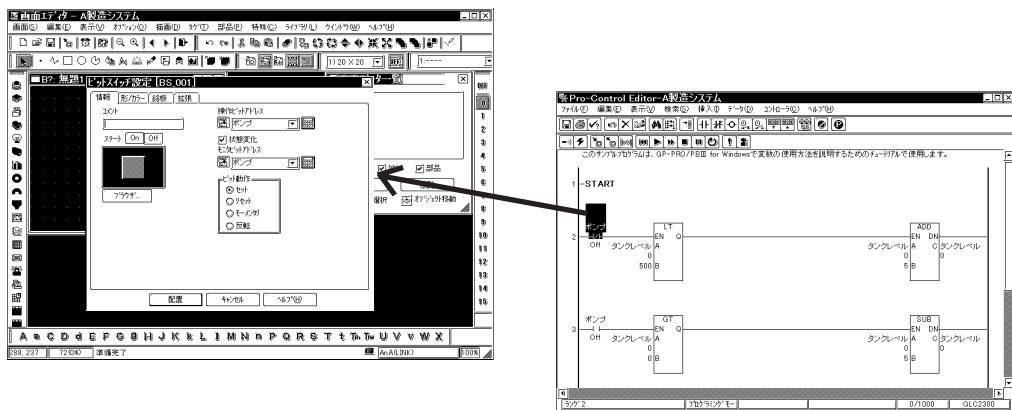




2. Editorで以下のロジックプログラムを作成してください。完成版は、「pump.prw」ファイルにありますので参照してください。

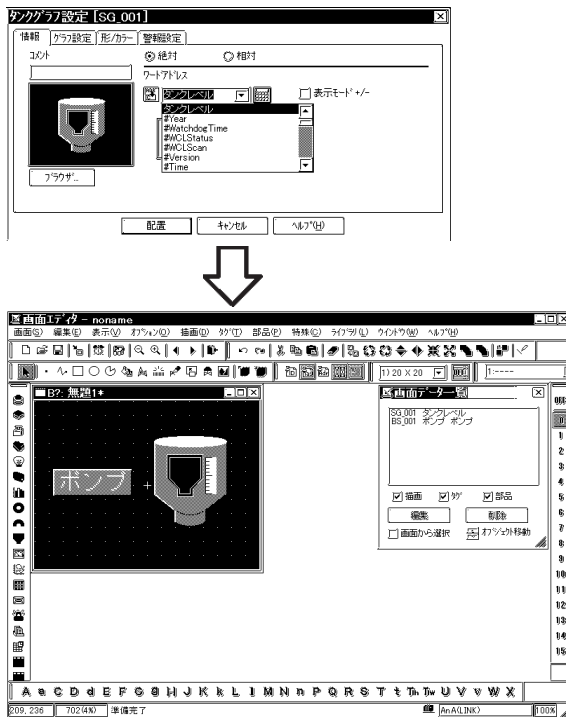


3. Editorの変数をインポートするために保存を行ってください。GP-PRO/PB で変数を使用できるようになります。
4. Editorの「ポンプ」命令を選択し、GP-PRO/PB へドラッグします。GP-PRO/PB でドロップすると「部品選択」ダイアログボックスが表示されるので、「ビットスイッチ」を選択して[OK]します。



5. 操作ビットアドレスが「ポンプ」になっていることを確認して[配置]してください。GP-PRO/PB に「ポンプ」ビットスイッチが作成されます。
6. GP-PRO/PB でタンクグラフを作成します。「ツールバー」から「タンクグラフ」をクリックするか、「メニューバー」から[部品(P)/タンクグラフ(G)]をクリックします。

7. 「プルダウンメニュー」から「タンクレベル」を選択して[配置]します。

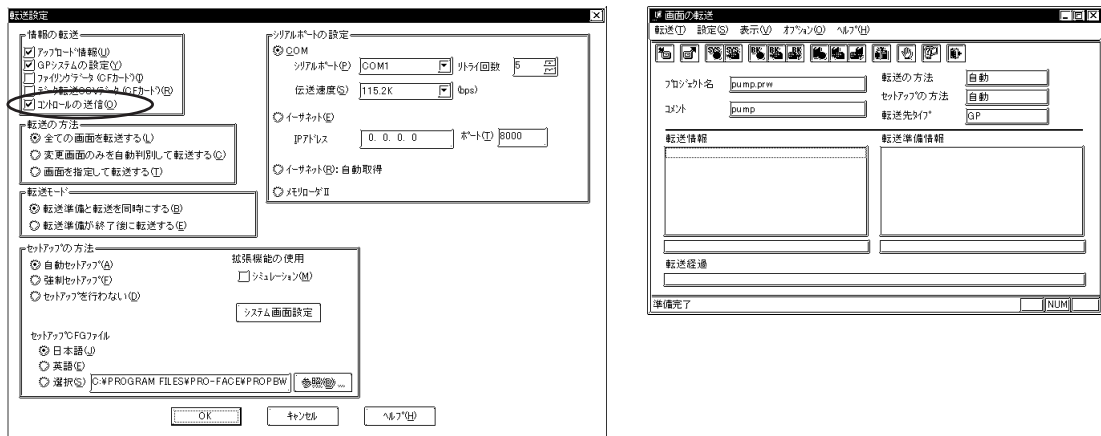


## 5.2 GLC へのダウンロード

コントロールアプリケーションを動作させる前に、GP-PRO/PB で作成した画面をGLCへダウンロードします。

画面作成ソフト GP-PRO/PB で作成した画面を GLC にダウンロードする手順

1. [画面の転送]をクリックします。
2. 設定メニューの[設定]で「情報の転送 / コントロールの送信」にチェックが入っているのを確認し、[画面の転送]ウィンドウで転送を行ってください。エラーが発生した場合にはそのエラーとともに、ダウンロードの進行状況が表示されます。



**重要** ・ ダウンロード中、他のプログラムはコンピュータの同じポートを使用することはできません。

## 5.3 ポンププロジェクトの実行

サンプルプログラムが動作しているのを確認するには、コントローラを実行してください。

コントローラのロジックプログラムを GLC にダウンロードする手順

1. GP-PRO/PB を起動します。
2. Pump.prw を選択し、[コントロールの作成]をクリックします。
3. [コントローラ]メニューで[ロジックプログラム書き込み]を選択します。
4. しばらくの間[ダウンロードの進行状況]ウィンドウが表示されます。
5. [コントローラ]メニューで[モニタモードへの移行]を選択します。
6. [コントローラ]メニューで[運転 / 停止]を選択します。コントローラのコントロールパネルが表示されます。
7. [運転]をクリックします。

**重要** ・ 本チュートリアルでは外部への入出力を行いませんが、I/O機器を接続して外部への入出力を行う場合は、[コントローラ / コマンド]メニューで「I/O 使用可」の設定を行ってください。  
参照 「3.2 コントローラの運転 / 停止」

これで、GLC にダウンロードされたコントローラがロジックプログラム Pump.prw を実行します。  
参照 「第3章 ロジックプログラムを実行する」

動作確認

1. GLC 画面上の[ON]ボタンをタッチします。ポンプでタンクから汲み出すにつれて、液レベルが下がるのを確認してください。
2. GLC 画面上の[OFF]ボタンをタッチします。ポンプがそれ以上タンクから汲み出さず、液レベルが上がるのを確認してください。

上記内容が問題なく動作している場合は正常です。動作しない場合は、もう一度、手順を1からやり直してください。

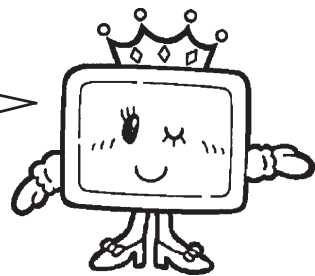
### まとめ

このレッスンでは、以下の操作を学習しました。

- ・ プロジェクトマネージャを開く方法
- ・ コントローラとリンクして動作する GP-PRO/PB によるプロジェクトの作成方法
- ・ 変数を、GP-PRO/PB にインポートする方法
- ・ Editor の変数を GP-PRO/PB の作画オブジェクト(部品、タグなどの要素)にリンクさせる方法
- ・ GP-PRO/PB と Editor を組み合わせた GLC での実行手順

MEMO

このページは、空白です。  
ご自由にお使いください。



# 第6章

## Pro-Control と Pro-Server

Pro-Server を使用すれば、イーサネット経由で変数のリード/ライトおよび2Way機能(配信、アクション機能など)を実行できます。

Pro-Server の詳細については「Pro-Server with Pro-Studio for Windows オペレーションマニュアル」を参照してください。

本章では、Pro-Server で GLC 変数を使用する方法について説明します。

### 6.1 GLC 変数のインポート

Pro-Studio で変数を扱うためには、GP-PRO/PB にインポートされた GLC シンボルを、Pro-Studio を使ってインポートします。

インポートした GLC シンボルは GLC ローカルシンボルとして登録されます。

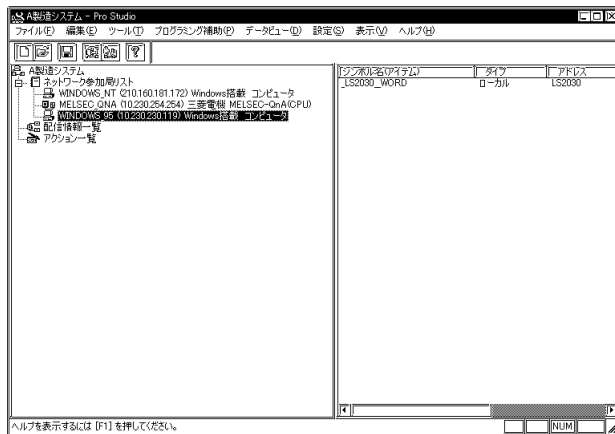
この GLC ローカルシンボルは、

- ・ GLC タイプの参加局のみ存在するシンボル
- ・ 編集・削除はできない
- ・ デバイスの種類はビットと 32 ビット 16 進のみとなります。

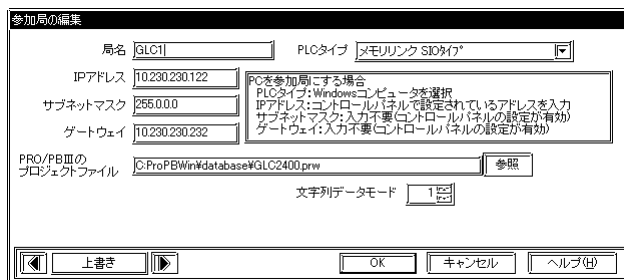
## 6.1.1 GLC 変数のインポート手順

Pro-Studio で GLC シンボルをインポートする方法について説明します。

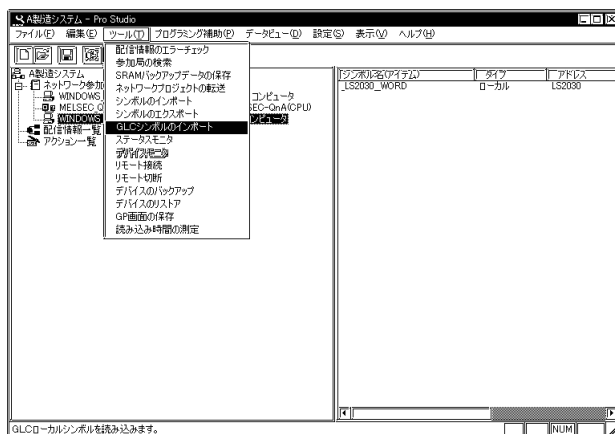
1. Pro-Studio を起動します。
2. ネットワーク参加局から GLC シンボルをインポートする参加局を選択します。



3. [参加局の編集] ダイアログボックスが表示されます。[GP-PRO/PB のプロジェクトファイル] にインポート元のプロジェクトファイルを指定してください。



4. メニューバーより [ツール / GLC シンボルのインポート] を選択します。



5. 以下のダイアログボックスが表示され、GLCシンボルがインポートされます。

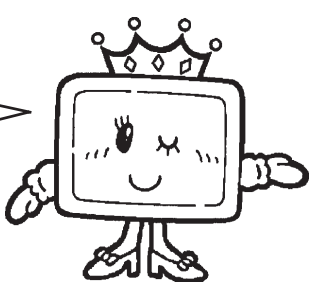


**重要**

- ・ インポートしたGLCシンボルの編集はできません。
- ・ GLCシンボルをインポートした場合は、必ずネットワークプロジェクトファイルをGLCに転送してください。
- ・ EditorでGLCの変数(GLCシンボル)を追加/削除した場合は、必ずGLCシンボルを再インポートしてください。  
EditorではGLCの変数が追加/削除されるごとに全ての変数情報をメモリに再マッピングしています。古いマッピング情報でPro-ServerがGLCにアクセスすると誤動作の原因にもなります。

MEMO

このページは、空白です。  
ご自由にお使いください。





# 第7章

# エラーと警告

プログラムでエラーチェックが行われると、[エラーチェック]ウィンドウにエラーや警告が表示されます。これらのエラーや警告は、プログラムのロジック、変数、I/Oなどに起こる問題に関するものです。エラーの原因ごとにエラーコードが割り付けられています。このコードの値を参考にすると、エラーや警告の発生原因が特定できます。

## 200-299：ロジックのエラーと警告

ロジックプログラムの命令に関する情報については、メインウィンドウで選択してから[ヘルプ]メニューで[関係トピックヘルプ]を選択して[F1]キーを押してください。

### エラー 200

変数名はビットでなければなりません。

命令にビットのオペランドが必要です。以下のいずれかです：

ビット値

ビット配列の要素

整数値のビット要素

### エラー 201

変数名はカウンタでなければなりません。命令にカウンタの値が必要です。

### エラー 202

変数名はタイマでなければなりません。命令にタイマの値が必要です。

### エラー 203

・・・は整数または実数でなければなりません。命令に、変数または定数として整数または実数が必要です。

### エラー 204

・・・は定数でない整数または実数でなければなりません。命令に整数または実数の変数が必要です。命令を定数にすることはできません。

### エラー 205

・・・は整数でなければなりません。命令に現在値が整数である変数または整数の定数が必要です。

エラー 206

・・・は整数でなければなりません。しかし配列は許されません。命令に現在値が整数である変数または整数の定数が必要です。命令を配列にすることはできません。

エラー 207

・・・は定数でない整数でなければなりません。命令に整数の変数が必要です。命令を定数にすることはできません。

エラー 208

変数名がラベルでなければなりません。命令にラベル名が必要です。そのラベル名が存在しなければなりません。

エラー 209

変数名がサブルーチンでなければなりません。命令にサブルーチン名が必要です。

エラー 210

ラベルが範囲外のところで使用されています。指定したラベルは存在しますが、有効範囲外です。

エラー 211

サブルーチンは自分自身をコールできません。サブルーチンの中でJSR命令で同じサブルーチンをコールします。

エラー 212

・・・は・・・と同じタイプでなければなりません。2個の変数名が同じタイプ(整数、実数など)でなければなりません。

エラー 213

・・・は・・・と同じサイズでなければなりません。2個の変数名が同じサイズでなければなりません。つまり2個の変数名はどちらも、「同じ数の要素をもつ配列」または「非配列」でなければなりません。

エラー 214

XはYと同じサイズまたは、整数でなければなりません。2個の変数名が同じサイズであるか、または2番目の変数名の方が大きいサイズのように見なされる整数でなければなりません。

エラー 215

Xは整数または実数またはビットの配列でなければなりません。命令に変数、定数、または配列として、整数、実数、またはビットが必要です。

エラー 216

・・・は定数でない整数または実数、またはビット配列でなければなりません。命令に変数または完全な配列として、整数、実数、またはビットが必要です。定数にすることはできません。

警告 217

両方の変数名は定数です。命令が2個の定数を比較しています。

警告 218

入力変数名が出力命令に使用されました。変数は入力にマークをしているのに([変数設定]ウィンドウを参照)出力命令で使用されています。I/O割り付けを確認してください。

警告 219

設定値が " 0 " です。

警告 220

設定値が " 0 " です。

警告 224

変数名に保持型のデータを使用できません。命令の変数名に割り付けた変数に保持型データを使用できません。

## 警告 225

・・・は整数配列でなければなりません。命令のオペランドに割り付ける変数は、整数配列でなければなりません。

## エラー 230

・・・は定数でない実数でなければなりません。

## エラー 232

・・・は・・・の配列参照範囲を超えています。BMOV/FMOV 命令はオペランド・・・で出力オペランドに割りついている配列変数の要素を指定しますが、その配列変数の要素数以上の要素番号を整数定数で指定しています。参照範囲内で指定してください。

## エラー 234

・・・は整数定数でなければなりません。

## エラー 235

・・・の値は・・・よりも大きくなければなりません。

## エラー 236

・・・と・・・は異なる値にしてください。

## エラー 237

・・・は整数で表現できる範囲を超えています。

## エラー 250

重複ラベルは使用できません。同じラベルが、2回以上定義されています。プログラムの別の部分であっても重複はできません。

## 警告 251

空サブルーチン実行上意味がありません。サブルーチンに行がありません。

## 警告 252

空ラングは実行上意味がありません。行に命令がありません。空の行の修正を選択しないと、プログラムに効果が生じません。

## 警告 253

空分岐は実行上意味がありません。分岐に命令がありません。空の分岐の修正を選択しないと、プログラムに効果が生じません。

## エラー 254

出力、演算などの制御命令が行の最後になければなりません。命令の右側に何もありません。

## 警告 255

・・・が複数のタイマで使用されています。同一のタイマ変数が複数のタイマ命令で使用されています。結果は不定となります。[リファレンス]ダイアログボックスを使用して他のタイマ命令を検索し変数名と変更してください。

## エラー 256

・・・が複数のカウンタで使用されています。同一のカウンタ変数が複数のカウンタ命令で使用されています。結果は不定となります。[リファレンス]ダイアログボックスを使用して他の命令を検索し変数名を変更してください。

## エラー 257

行上の最後の命令は出力命令でなければなりません。命令が出力命令になっていません(したがって変数名の値が変わりません)。

## エラー 258

複数の出力命令を使用できません。出力命令は右側に他の命令をもつことはできません。

エラー 259

分岐の最後の命令は出力命令でなければなりません。出力命令は右側に他の命令をもつことはできません。

エラー 260

ネスティングのレベルが最大を超えました。行の分岐レベルが多すぎます(レベルの最大数は25です)。行を小さくいくつか分割してみてください。

エラー 262

プログラムが大きすぎます。GLCのフラッシュメモリの使用可能な容量を超えています。

警告 263

・・・が複数のコイルで使用されています。同一変数が複数のコイルで使用されています。ロジックプログラムの処理順で最後に変数が割り付けられた命令の結果が有効になります。

エラー 264

対応するNEXT命令が見つかりません。FOR命令とNEXT命令は対で使用してください。

エラー 265

対応するFOR命令が見つかりません。FOR命令とNEXT命令は対で使用してください。

エラー 266

FOR、NEXT命令は、他の命令と同一行上では使用できません。FOR、NEXT命令と同一行上にある命令を他の行へ移動させてください。

エラー 267

この命令は現在のプラットフォームでは対応しておりません。

エラー 268

FOR-NEXTループからの脱出はできません。

エラー 269

行のメモリ使用率が・・・%超えています。

エラー 270

ラベルの最大数を超えました。最大数は2048です。

エラー 271

変数の最大数を超えました。最大数は8192です。

エラー 272

定数の最大数を超えました。

エラー 273

NT命令およびPT命令の最大数を超えました。最大数は2048です。

エラー 274

PID命令の最大数を超えました。最大数は100です。

## 300-399：変数のエラーと警告

警告 300

変数タイプは入力または出力ですがI/Oアドレスが割り付けられていません。変数は入力または出力にマークをしているのに( [変数設定]ウィンドウを参照) I/Oに割り付けられません。

エラー 301

タイプが設定されていません。変数に変数タイプが割り付けられていません。変数タイプを割り付けるには、[変数設定]ウィンドウを使用してください。

## エラー 302

ラベルが見つかりません。存在しないラベルがジャンプサブルーチン(JSR)命令の飛び先として指示されています。

## エラー 303

参照される変数はタイマまたはカウンタでなければなりません。タイマまたはカウンタの変数の要素を指定したにもかかわらず、別のタイプの変数になっています。[変数設定]ウィンドウを参照してください。

## エラー 304

変数タイプは整数でなければなりません。変数を使用して配列の要素かまたは修飾語を指定しました。この変数は整数でなければなりません。[変数設定]ウィンドウを参照してください。

## エラー 305

配列変数でない変数に配列参照されました。配列の要素を指定しましたが、その変数が配列として指定されていません。[変数設定]ウィンドウを参照してください。

## エラー 306

配列の範囲を超えて参照しています。配列のサイズと等しいかそれ以上の定数を使用して、配列の要素を指定しました。(有効な要素には0からサイズ-1の番号がついています。)サイズは[変数設定]ウィンドウで変更できます。

## エラー 308

修飾参照が範囲を超えています。範囲を超えたビット、バイト、ワードの要素を指定しています。

## エラー 309

変数の参照が正しくありません。カウンタの変数にタイマの参照を指定したり、その逆を指定しています。

## 警告 310

・・・がすでに存在します。置き換えできません。その名前の変数はすでに存在しています。[変数のインポート状態]ウィンドウで[OK]をクリックすると、元からあった変数は新しい変数に置き換えられます。

## エラー 311

クリップボードのバッファのフォーマットが認識されません。現在のクリップボードの中身は、[変数一覧]ウィンドウへの貼り付けには適していません。

## エラー 312

警告が多すぎます。[変数のインポート状態]ウィンドウでは、一定の数の警告しか表示しません。このメッセージが表示されたら、他にも表示されていない警告があるということです。

## 警告 313

・・・の右括弧 "]" がありません。配列には、[]で囲んだサイズが必要です。たとえば、整数 [10] など。

## 警告 314

配列のサイズは・・・で無効です。サイズ1が使用されます。この変数は配列を意図しているようですが、サイズが認識されません。サイズは、整数 [10] のように、[]で囲んだ整数にしなければなりません。

## 警告 315

・・・に未知の変数・・・があります。変数タイプは未定義となります。この変数は Editor の変数タイプとして認識されません。以下の原因が考えられます：

- ・スペル間違い
- ・先頭か末尾にブランクがある

警告 316

・・・に未サポートの配列・・・があります。設定した配列を無視します。この変数では配列は設定できません。

エラー 317

変数名が無効です... 変数名として無効な文字を入力しました。

エラー 318

エラー数が多すぎます。

エラー 320

I/O変数が多すぎます。

エラー 321

変数が多すぎます。変数の数を減らしてください。

エラー 328

変数を正常に作成できませんでした。'(変数名)'を非画面共有変数に変更しています。変数一覧にて変数の属性を確認してください。

エラー 329

I/Oアドレスに該当するターミナルが見つかりませんでした。変数一覧をインポートしたときにCSVファイルで変数情報のI/Oアドレスに相当するターミナルが存在しない。またはアドレスが不正です。

## 400-499 : I/O のエラーと警告

エラー 400

変数名がすでに割り付けられています。この変数は複数のI/Oポイントに割り付けられています。[I/O設定]ウィンドウを参照してください。

## 500-549 : 一般的な I/O ドライバのエラー

エラー 500

WLLファイルが破損しているか、WLLファイルのダウンロード中に障害が発生した可能性があります。

エラー 501

I/Oターミナルに割り当てられている変数に、内部変数タイプの変数が割り当てられています。入力もしくは出力タイプに変更してください。

エラー 502

出力ターミナルに割り当てられている変数に、入力タイプの変数が割り当てられています。出力タイプに変更してください。

エラー 503

入力ターミナルに割り当てられている変数に、出力タイプの変数が割り当てられています。入力タイプに変更してください。

エラー 504

整数ターミナルに割り当てられている変数に、ビットタイプの変数が割り当てられています。整数タイプに変更してください。

**エラー 505**

ビットターミナルに割り当てられている変数に、整数タイプの変数が割り当てられています。ビットタイプに変更してください。

**エラー 506**

ドライバがコントローラの変数を確認しないときに表示されます。

**エラー 507**

ターミナルに変数が割り当てられていないときに表示されます。

**エラー 508**

ドライバでサポートされていないGLCタイプが選択されたときに表示されます。

**600-799 : PID 命令のエラー****エラー 600**

コントロールブロック変数は、要素数7個以上の整数配列型に設定してください。

**エラー 601**

PIDのパラメータは、整数型に設定してください。

**800-899 : 特定の I/O ドライバのエラー**

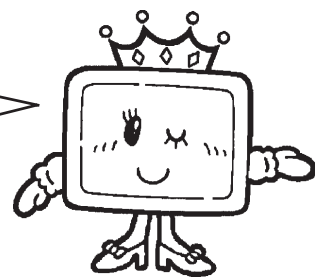
I/O ドライバに関するエラーについては、I/O ドライバのオンラインヘルプを参照してください。

**900-1000 : 特定の I/O ドライバの警告**

I/O ドライバに関する警告については、I/O ドライバのオンラインヘルプを参照してください。

MEMO

このページは、空白です。  
ご自由にお使いください。





# 第8章

# 用語集

## 16進数

16が1単位となる整数の表記法です。この値は、前に16#をつけて入力することができます。たとえば、16#FFは255になります。

## I/O

Input(入力)/Output(出力)。コントローラは、(株)デジタル製I/Oユニットやサードパーティの提供するI/Oユニットにより現実のデバイスに接続されます。

## I/Oアドレス

I/Oユニットに割り付けるときのアドレス値です。I/Oアドレスのフォーマットは、割り付けるドライバによって異なります。

## IEC61131-3

国際電気標準会議(IEC)の制定した標準で、命令リスト(IL)、ラダーロジック図(LD)、機能ブロック図(FBD)、構造化テキスト(ST)、順次機能チャート(SFC)の5つの制御言語について印刷および表示方法を定義します。

## WDT(ウォッチドッグタイマ)

一定時間内にEND行まで実行できなかったとき、WDT(ウォッチドッグタイマ)がメジャー異常を検出します。WDT(ウォッチドッグタイマ)の設定は[設定]ダイアログボックスで設定できます。

## エラー

エラーには、メジャー異常、マイナー異常、I/Oエラーの3つがあります。

メジャー異常が発生すると、コントローラは、すぐにロジックプログラムの実行を停止します。Editorには、「メジャー異常」と表示されます。状態をクリアするには、コントローラを[運転/停止]ダイアログボックスでリセットしてください。

マイナー異常は、軽微なエラーです。

I/Oエラーは、I/Oの読み込み/書き込みのエラーです。

## 強制変更(Forces)

ビット変数は強制的にONまたはOFFにできます。これはロジックで実行する動作に優先するものです。たとえば、変数が強制的にOFFにされた場合、ロジックがONにしようとしても、OFFのままです。プログラムの強制リストは、[強制リスト]ウィンドウで見ることができます。

## クリップボード(Clipboard)

Windowsの機能で、コピーや貼り付けのための一時記憶場所です。異なるアプリケーション間でも、ひとつのアプリケーション内でも使用可能な機能です。

### コメント(Descriptions)

コメントは最大32767文字(半角)のテキストで、プログラムの一部についてコメントするものです。コメントの要約は、[コメント一覧]ウィンドウで見ることができます。

### コントローラ

コントローラはロジックプログラムを実行し、I/Oを制御します。コントローラは表には現れず、GLCの拡張タスクとして実行します。Editorはモニタモードでコントローラを監視します。

### サブルーチン(Subroutine)

個々に別の名前をついたロジックのグループ。サブルーチンはENDとPENDというラベルの間に位置します。[挿入]メニューで[サブルーチン]をクリックすると、[サブルーチン開始]と[サブルーチン終了]の両方のマークが作成されます。2つのラベルの間にロジックを挿入することができますが、サブルーチン内に別のサブルーチンを入れることはできません。

サブルーチンはJump Subroutine( JSR )(ジャンプサブルーチン)という命令で呼び出されます。サブルーチンはどこからでも何度でも呼び出すことができますが、自分自身のサブルーチンからは呼び出すことはできません。

サブルーチンは、コードを一回書き込むだけですむ点が便利です。サブルーチン名が必要です。

### サブルーチン名

サブルーチン名には、文字・数字・下線を32文字まで使用できます。サブルーチン名を数字から始めることはできません。

### システム変数(System Variables)

「システム変数」は事前に定義された特殊な変数で、コントローラの状態に関する情報を提供し、コントローラの操作に反映させます。自動で作成され削除不可能な点を除けば、通常の変数とよく似た働きをします。

### 実数(Real)

小数点を含む数や特定の記数法で表す数です。Editorでは実数の範囲を $\pm 2.25e^{-308}$ から $\pm 1.79e^{-308}$ とします。有効数字は最大15桁です。

### 状態フロー(State Flow)

モニタ時パラメータに基づいて個々の命令の動作を緑色で強調表示します。パラメータの状態に応じて、それぞれの接点が強調表示されます。

### 整数(Integer)

32ビットの情報の含まれる記憶単位。整数には、-2,147,483,648から2,147,483,647(16進数で16#00000000から16#FFFFFF)までの値が割り付けられます。整数には小数点を含むことはできません。

### 接続線

命令の間の水平線です。新規の命令を挿入する場合は、挿入したい位置の接続線にフォーカスをあわせませす。

### ビット(Discrete Point)

OFFまたはONのいずれかの状態を持つ変数です。

### [現在値表示リスト]ウィンドウ(Data Watch List Window)

登録した変数のモニタリング状況がこのウィンドウに表示されます。[オプション]ダイアログボックスで更新の基準を調整することができます。

### 定数

42(整数)や3.14159(実数)などの数です。

### ドラッグ(Drag)

マウスの左ボタンを押したままマウスを動かしてから離します。マウスポインタにより、そこがドラッグできる有効な場所かどうかわかります。

### 内部変数(Internal Variable)

I/Oに割り付けられている以外の変数です。

### バイト(Byte)

8ビットの情報をもつ記憶単位。バイトには0から255の値を割り当てます。Editorの整数は4バイトから成ります。

### 配列

(1つの名前を割り付けた)同じタイプの複数の要素です。

### 変数名(Parameter)

(命令に割り付けられている、変数、変数の要素、定数、ラベル名)への入力または(命令に割り付けられている、変数、変数の要素、定数、ラベル名)からの出力。

### パワーフロー(Power Flow)

ロジックプログラムの実行する流れ。

### ビット(Bit)

基本的な記憶単位、値は1または0。

### フォーカス(Focus)

ロジックプログラムでの選択事項を強調表示するための黒色のカーソルのことです。

### 分岐

行に並列接続してロジックプログラムを実行します。

### ブックマーク(Bookmark)

ロジックプログラムのどこにでも配置できる非表示のマーカーで、プログラムのその箇所にすぐに戻ることができます。

### プログラムモード

ロジックプログラムを含む拡張子が .prw のファイルを編集します。

### 変数(Variable)

カウンタ、I/Oの値などの現在値です。理解しやすい変数名をつけてください。

通常、変数は自動的に作成されます。パラメータボックスか[I/O設定]ウィンドウで新規の名前を入力すると、Editorは適切なタイプで新規の変数を自動的に作成します。[変数一覧]ウィンドウで不要な変数を削除できます。変数名は文字、数字、下線を含めて最大20文字で定義しますが、数字で始めることはできません。

### 母線

メインウィンドウの両端にある2本の垂直な線です。

### 命令(Instruction)

ロジックプログラムの基本的な要素(ビット命令、ビット演算命令、データ操作命令、演算命令、タイマとカウンタ、プログラム制御命令)です。コントローラに特定の機能を実行させる命令です。Editorの命令は、IEC61131-3の仕様に基づいています。

### モニタモード

Editorはコントローラに書き込んだロジックプログラムをモニタします。

たとえば: Power\_Off\_pushbutton、ResetButton、ALARM2 などです。

#### 要素(Element)

要素は変数全体ではなくある一部分の名前のことです。以下のような部分が該当します：

- ・タイマの変数やカウンタの変数の要素
- ・配列の要素
- ・整数の一部；変更子を参照してください。

たとえば：FillTimer.ET、LimitSwitches<5>、LimitSwitches<Index>、Flags.X<12>、SensorArray<Position>.X<Index>

#### ラベル名(Label Name)

最大32文字から成る名前で、ロジックプログラム内の位置を識別したりラベルを付けたりするための名前です。ラベル名を数字から始めることはできません。

#### 行

1つ以上の命令で2本の母線の間で作成する回路です。

#### ロジックプログラム

アプリケーションプログラム中の行の集合。Editorではラダー言語をサポートしています。

#### ワード(Word)

16ビットからなる記憶単位。ワードには0から65535までの値が割り付けられます。

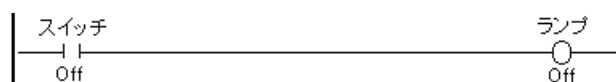
# 付録

## 付 .1 固定変数モード

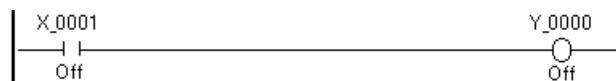
変数を作成する上で、Editorでは下図のように2つの動作モードがあります。固定変数モードは、ロジックプログラム開発においてI/Oやカウンタのデータを格納するエリアのアドレスを自動で作成するモードです。

ロジックプログラムではデバイスアドレスで使用します。

変数モード : I/Oやカウンタのデータを格納するエリアを変数として、ユーザーが任意に定義することができます。ロジックプログラムでは下図のように名前で使用することができます。



固定変数モード : 一般のPLCのデータ格納エリアのようにデバイスアドレスというエリアとして、ユーザーが擬似的に定義することができます。ロジックプログラムでは下図のようにデバイスアドレスで使用します。

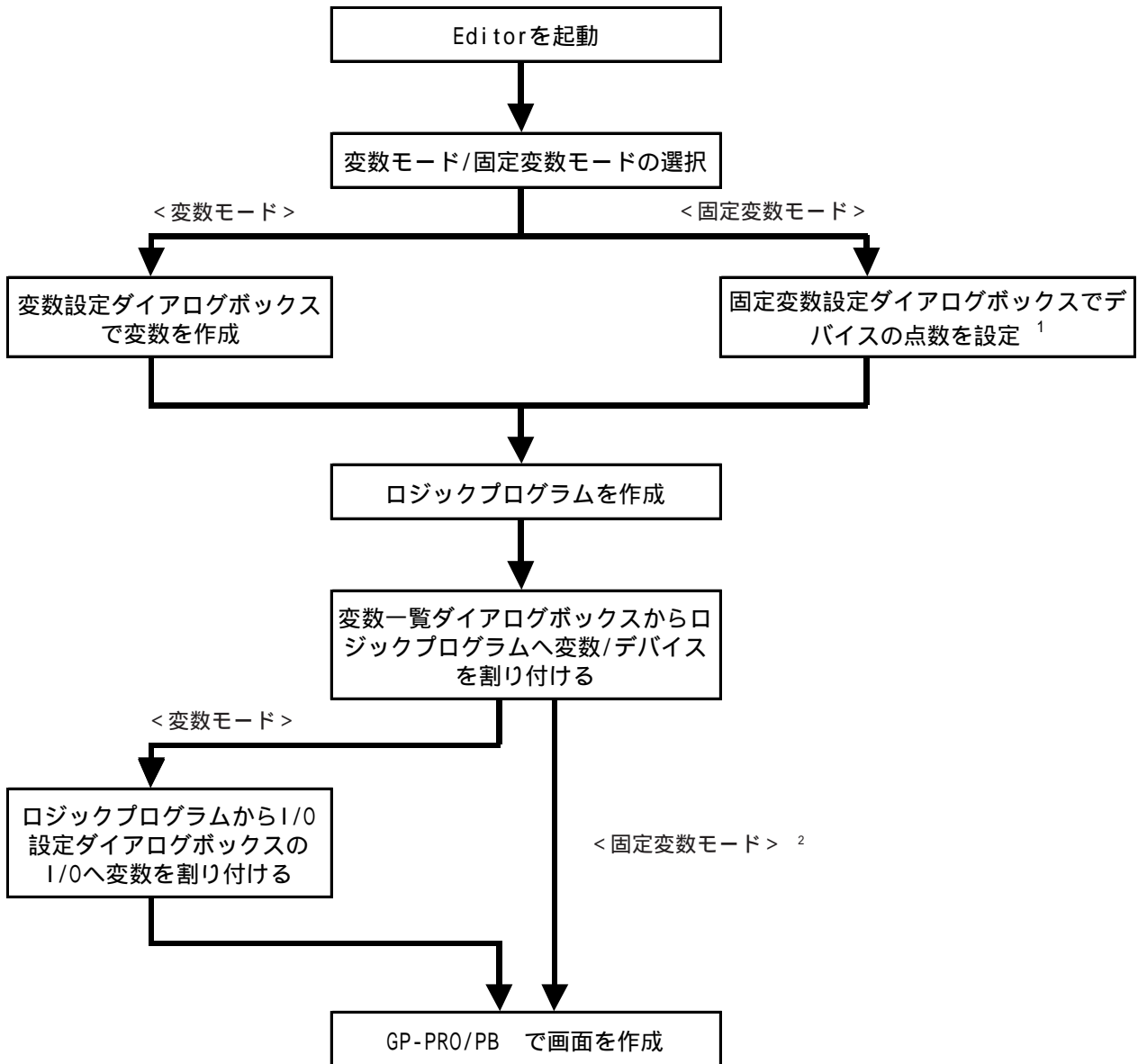


- 重要**
- 固定変数モードでは、配列を指定することができません。ただし、PIDデバイスとLSエリアのみ配列要素を持っています。配列について、参照「Pro-Control Editor ユーザーズマニュアル 2.3 配列変数へのアクセス」
  - 固定変数モードでは、PLCのようにオフセット指定でのアクセスはできません。
  - 固定変数モードでは、変数のインポート / エクスポートを行うことができません。

### ロジックプログラム開発の概略手順

固定変数モードでは、ビット型や整数型などの変数タイプやI/Oへの割り付け情報があらかじめ各デバイスで定義されているために、変数モードの開発手順と多少異なります。

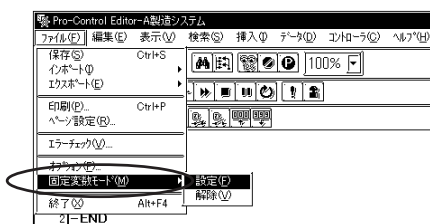
下図に変数モード / 固定変数モードの開発手順を示します。



- 1 次項の "固定変数設定" ダイアログボックスでデバイス点数や属性を設定します。
- 2 固定変数モードでは、前行程の "固定変数設定" ダイアログボックスで入出力タイプを定義しているため、I/O設定ダイアログボックスでの割り付け作業は必要ありません。

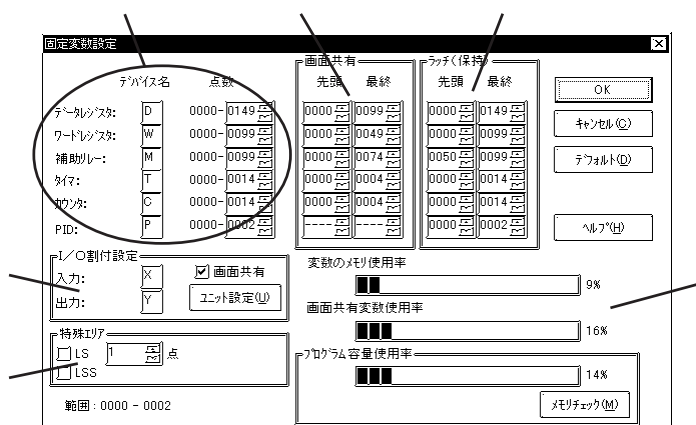
## 固定変数モードへの移行とデバイスアドレスの設定

固定変数モードへは、「ファイル/固定変数モード/設定」をクリックすると移行します。変数モードへ戻る場合は、解除をクリックしてください。



## デバイスアドレスの設定

固定変数モードで使用するデバイスアドレスの点数を下図の「固定変数設定」ダイアログボックスで設定します。



## デバイス名および点数

各デバイスのシンボル記号を英文字の1文字で任意に設定することができます。また、ここで設定する点数は、右側のボックスで設定する画面共有デバイスの点数とラッチ(保持)デバイスの点数を含む、総和のデバイス点数になります。

各デバイスの型、タイプは下表のようになります。

シンボル記号 (デフォルト)	デバイス名	型	タイプ	最大点数
D	データレジスタ	整数型	内部変数	1000
W	ワードレジスタ	実数型	内部変数	1000
M	補助リレー	ビット型	内部変数	1000
T	タイマ	タイマ型	内部変数	500
C	カウンタ	カウンタ型	内部変数	500
P	PID	整数型配列	内部変数	100
X	入力	ビット型	入力	-
XW		整数型	入力	-
Y	出力	ビット型	出力	-
YW		整数型	出力	-

### 重要

- 使用しないデバイスについては、点数 "0000" の状態から入力エリア横の下ボタンを押すか、点数に "-" (ハイフン) を入力してください。点数の入力エリアがグレーアウトし、該当デバイスを使用しない設定となります。
- デバイスのシンボル記号を変更するときは、英文字1文字を入力してください。

### 画面共有

で設定した範囲内で画面と共有するデバイスの点数を設定します。本項目で設定する画面共有は変数モードで”画面共有変数”にあたるデバイスです。画面共有変数の詳細は、参照「Pro-Control Editor ユーザーズマニュアル 2.2 変数タイプ」

### ラッチ(保持)

で設定した範囲内でラッチ(保持)するデバイスの点数を設定します。本項目で設定するラッチ(保持)は変数モードで”保持型変数”にあたるデバイスです。保持型変数の詳細は、参照「Pro-Control Editor ユーザーズマニュアル 2.2 変数タイプ」

### I/O 割付設定

入出力デバイスのシンボル記号を任意に設定することができます。また、入出力デバイスを画面共有にする場合は右上のチェックボックスにチェックを入れてください。

”ユニット設定”ボタンを押すと、I/O設定ダイアログボックスが表示され、ドライバの設定を行うことができます。

- 重要**
- ・ I/O設定ダイアログボックスでは、デバイスの割付や割付解除はできません。ドライバの設定のみ可能です。
  - ・ 複数のFlexNetworkが接続されている場合のデバイスアドレスについて、”S-No.”の小さい順番から割り付けられます。  
例) S-No.1 入出力8点ユニット      Y0000 ~ Y0007  
      S-No.5 出力16点ユニット      Y0008 ~ Y0023

### 特殊エリア

LSまたはLSSエリアを使用する場合は、チェックを入れてください。その際は、使用する点数も設定してください。

### メモリの使用率

変数のメモリ使用率、画面共有変数のメモリの使用率を示します。この使用率が100%を超えないようにロジックプログラムを開発してください。



- ・ プログラム容量使用率については、「メモリチェック」ボタンを押してから使用率を確認してください。プログラムを変更した場合には、瞬時に反映されない場合があります。

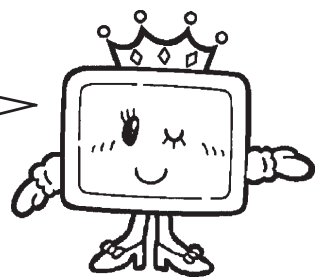


### 固定変数モードの制限事項

- ・ 固定変数設定ダイアログボックスで設定した値が設定可能範囲を超えた場合、設定範囲内の最小値 / 最大値として設定されます。
  - 重要** ・ 使用しないデバイスについては、点数 " 0000 " の状態から入力エリア横の下ボタンを押すか、点数に " - " (ハイフン) を入力してください。点数の入力エリアがグレーアウトし、該当デバイスを使用しない設定となります。
  - ・ デバイスのシンボル記号を変更するときは、英文字 1 文字を入力してください。
- ・ 変数一覧ダイアログボックス、変数設定ダイアログボックス、I/O設定ダイアログボックスでデバイスのタイプや属性を変更することはできません。  
変更する際は、前述した固定変数設定ダイアログボックスで変更を行ってください。
- ・ 変数モードでプログラムを作成している場合、固定変数モードに移行すると新規作成状態になります。
- ・ 固定変数モードで作成したプロジェクトを機種変更した場合、必ず固定変数モードの再設定を行ってください。

# MEMO

このページは、空白です。  
ご自由にお使いください。



## 索引

- E**
- Editor の設定 ..... 2-4
  - END ..... 2-48
- G**
- GLC シンボル ..... 5-11
  - GP-PRO/PB for Windows95  
Ver.2.1 ..... 13, 5-12
- I**
- I/O ..... 2-56, 3-10
  - I/O 設定 ..... 2-57, 2-58, 2-62, 2-69
- J**
- JMP 命令 ..... 2-48
  - JSR 命令 ..... 2-48, 2-49
- M**
- MOV 命令 ..... 2-53
- P**
- PEND ..... 2-48
  - Pro-Control Editor ..... 1
- ア**
- アウト・コイル ..... 2-30, 2-49
- エ**
- エラーチェック ..... 2-69
  - エレメント ..... 4-2
- オ**
- 応用命令 ..... 2-36, 2-40
  - オプション ..... 4-3
- カ**
- 拡張子 ..... 2-40
  - 確認タブ ..... 2-5, 2-6
  - カラー ..... 4-2
  - カラム ..... 2-71
- キ**
- キーボード ..... 11
  - キーボード対応表 ..... 11
  - 基本命令 ..... 2-36
- ク**
- クリップボードタブ ..... 2-7
- コ**
- コメントリスト ..... 2-45
  - コントローラ ... 3-1, 3-8, 3-10, 3-11, 3-12,  
4-1, 4-2, 4-3, 4-4, 4-5, 4-7
- サ**
- サンプルプログラム ..... 4-1, 5-15
- シ**
- システムの設定 ..... 3-1
  - 指定ラング ..... 2-55
  - ジャンプ ..... 2-48
  - 出力命令 ..... 7-2
  - 使用上の注意 ..... 12, 13
  - 初期化 ..... 2-34
  - 進行状況 ..... 3-6, 5-15
- セ**
- セーブ ..... 2-21
  - 接点 ..... 2-26, 2-28
  - 説明のための表記 ..... 10
  - 選択範囲 ..... 2-51
- ソ**
- ソースファイル ..... 4-1
  - 挿入 ..... 2-27, 2-49
- タ**
- ダイアログボックス ..... 2-34, 2-45, 2-46
  - タイプ ..... 2-51
  - ダウンロード ..... 3-1, 3-6, 5-14

<b>テ</b>		<b>メ</b>	
データ	4-4, 4-5	命令	2-25, 2-36
ディスク	12	命令から部品への変換	5-5
テキスト	2-61	命令パラメータボックス	2-37
テキストフィールド	2-42	メッセージフィールド	2-42
デフォルト	4-2	メモリ	3-5
<b>ト</b>		<b>モ</b>	
動作確認	5-15	モデル	11
ドライバ	2-57, 7-7	モデル環境	11
ドラッグ&ドロップ	5-10	モニタータブ	2-5
<b>ニ</b>		<b>ユ</b>	
入力端子	2-62	ユニット	2-59, 2-62
<b>ハ</b>		<b>リ</b>	
配列	7-2, 7-5	リファレンス	2-52, 2-54, 2-71
パラメータ	7-3	<b>ロ</b>	
反転表示	2-54	ロジックプログラム	2-22, 2-50, 2-54, 2-62
<b>ヒ</b>			
表記のルール	10		
<b>フ</b>			
ブックマーク	2-54		
部品から命令への変換	5-6		
プログラミングモード	3-8		
分岐	2-31, 2-32, 7-3		
<b>ヘ</b>			
編集タブ	2-4		
変数	1-1, 2-13, 2-14, 2-36, 2-42, 2-46, 2-48, 2-51, 2-56, 2-59, 2-69, 2-71, 2-73, 5-2, 6-1, 付-1		
変数コメント	2-44		
変数タイプ	7-5, 2-15		
変数リスト	2-37, 2-38, 2-44, 2-60, 2-14, 2-15, 2-16, 2-22, 2-25, 2-30, 2-36		
<b>ホ</b>			
保持型	3-9		
<b>マ</b>			
マニュアルの読み方	4		