Pro-face™
by Schneider Electric

画面作成&ロジックプログラムソフトウェア
GP-Pro EX

All-in-One
はじめてのSTC6000テキスト

（表示器付きコントローラーSTC6000シリーズ）

メモ（以下は余白です。ご自由にお使いください）

はじめに

このたびは、自己学習テキスト「はじめてのSTC6000テキスト」（以下本テキストと言います）をご購読いただきまして、誠にありがとうございます。

本テキストは、STC6000シリーズ（以下「STC6000」と言います）の基本的なロジックプログラムおよび作画方法を習得いただくためのテキストです。

弊社カスタマーケアセンターへのお問合せや実際に現場で使われている例をもとに、最も一般的かつご要望の高いロジックプログラムおよび作画方法をユーザー様自身で効率良く、自習いただけるよう構成されています。GP-Pro EXのシミュレーション機能を使用すると、パソコン上で表示器画面の動作確認を行うことができます。

また、第2章から第4章の完成版サンプルプロジェクトをご用意していますので、ロジックプログラムや画面の作成例を確認することができます。

なお、実際のご使用にあたっては、弊社製品マニュアルをよくお読みいただき、正しい取り扱いと機能を、十分にご理解いただきますようお願い申し上げます。

■ 対象ソフトウェア
画面作成&ロジックプログラムソフトウェア「GP-Pro EX　Ver4.09.500以降」

■ 解説している実施環境
表示器付きコントローラー　：STC-6300TA シンク出力タイプ（PFXSTC6300TADDKE）
パソコン　：Windows10が動作するパソコン

本テキストは、以下の方を対象にしております。

・PLCに対して知識のある方
・表示器の画面作成の経験のある方

おことわり

（１）　本テキストは、シュナイダーエレクトリックホールディングス株式会社の著作物です。文章の一部または全部の転載利用はお断りします。

（２）　本テキスト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、弊社カスタマーケアセンターまでご連絡ください。

（３）　本テキストを運用した結果の影響及び第3者のいかなる請求にも、シュナイダーエレクトリックホールディングス株式会社は一切責任を負いません。

（４）　製品の改良のため、本テキストの記述と本テキスト内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、各製品のマニュアルをご参照いただくか弊社カスタマーケアセンターまでお問い合わせください。

（５）　本テキストで使用する製品が、記録・表示する情報の中にシュナイダーエレクトリックホールディングス株式会社及び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㈱デジタルがこれらの権利の再利用について、ユーザー及びその他の第三者に、何らの保証や許諾を与えるものではありません。

お問い合わせ

本テキストの内容に関するご質問は、「カスタマーケアセンター」までご連絡ください。

シュナイダーエレクトリックホールディングス株式会社

カスタマーケアセンター: 03-4578-4697
https://www.proface.com/ja/contact

平日 9:00～19:00
祝日（弊社指定の休業日を除く） 9:00～12:00、13:00～17:00

安全にご使用いただくために

設計上の危険事項

・外部電源の故障やSTC6000本体の故障およびSTC6000の誤作動の場合、システム全体が 安全側に働くようにSTC6000の外部で安全回路を設けてください。誤出力/誤動作により、事故の恐れがあります。

(1)非常停止回路、保護回路、正転/逆転などの相反する動作のインターロック回路、位置決めの上限/下限/走行限など機械の破損防止のインターロック回路などは、STC6000の外部で回路構成してください。
(2) STC6000でウォッチドッグタイマエラーなどの異常を検出したときは、ロジックプログラムの実行を停止します。また、STC6000で検出できない入出力制御部分などの異常時は、入出力部の動作について予想のできない動きとなる可能性があります。
このとき、機械の動作が安全側に働くよう、STC6000の外部でフェールセーフ回路を構成したり、機構を設けてください。
(3)出力ユニットのリレーやトランジスタなどの故障によっては、出力がONのままになったり、OFFのままになったりすることがあります。重大な事故につながるような出力信号については、外部で監視する回路を設けてください。

・STC6000の本体の電源を立ち上げる前に、STC6000本体に接続されているＩ/Ｏユニットおよび負荷制御電源を投入するように回路を構成してください。STC6000のプログラムがRUNした後に負荷制御用電源が立ち上がると、誤出力や誤動作により事故の恐れがあります。

・STC6000の表示機能とコントロール機能または、STC6000と外部に接続されている機器に通信異常および電源断が発生したとき、システムが安全側に働くようにロジックプログラムにてインターロック回路を構成してください。誤出力や誤動作により事故の恐れがあります。

・人的損害や物的損害をもたらす可能性があるスイッチは、絶対にタッチパネル上に作らないでください。本体、ユニット、ケーブル等の故障により、意図しない出力信号が出て重大な事故につながる可能性があります。重大な動作を行うスイッチはSTC6000本体以外の装置より行うようにシステム設計をしてください。

・STC6000とコントローラーの通信異常で機械が誤動作しないようにシステム設計を行ってください。人体に傷害を負ったり、物的損害の恐れがあります。

・STC6000は航空機器、航空宇宙機器、幹線通信機器、原子力制御機器、生命の維持に 関わる医療機器などの極めて高度な信頼性・安全性が求められ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。

・STC6000を運送機器（列車、自動車、船舶等）、防災防犯装置、各種安全装置、生命の維持に関わらない医療機器などの、機能・精度において高い信頼性・安全性が求められる用途で使用する場合は、組み込まれるシステム機器全般として、冗長設計、誤動作防止設計等の安全設計を施す必要があります。

取り付け上の警告事項

・ STC6000の解体は絶対に行わないでください。高電圧部分がSTC6000内部にあり、STC6000を解体すると感電の恐れがあります。
・ STC6000は改造しないでください。火災、感電の恐れがあります。
・可燃性ガスのあるところでは、使用しないでください。爆発の恐れがあります。

配線上の警告事項

・取り付け、配線などは、必ず電源が供給されていないことを確認してから行ってください。感電や機器の破損の恐れがあります。
・配線後は必ず付属の端子台カバーを取り付けてください。端子台カバーを取り付けないと感電の恐れがあります。
・マニュアルに記載された仕様以外での環境で使用しないでください。仕様の範囲外で使用すると、感電、火災、誤動作や製品の破損の恐れがあります。

立ち上げ・保守時の警告事項

・通電中に端子に触れないでください。感電の恐れや誤動作の原因になります。
・清掃や端子ネジの増し締めは、通電されていないことを確認してから行ってください。
通電中に行うと感電の恐れがあります。
・STC6000は時計のバックアップのためにリチウム電池を内蔵しています。電池の電圧が低下 すると、時計データが失われます。電池を交換する際は、STC6000の交換用電池 (PFXZGEBT1) のみを使用してください。電池を正しく安全に交換するために、ハードウェアマニュアルに記載がある設定手順に従ってください。電池を交換する前に、STC6000の電源をオフにしてください。

設計上の注意事項

・入出力信号線の配線は動力回路のケーブルとは、別ダクトにしてください。ノイズにより、誤作動の原因になります。

取り付け上の注意事項

・ケーブルは、コネクタに確実に装着してください。接触不良により、誤入力や誤出力の恐れがあります。

配線上の注意事項

・FG端子は、STC6000専用のＤ種接地工事を行ってください。感電や誤動作の恐れがあります。
・STC6000への配線は、定格電圧および端子配列を確認した上で正しく行ってください。定格と異なった電源の接続や誤った配線を行うと火災や故障の恐れがあります。
・端子ネジは規定のトルクで締め付けてください。端子ネジの締め付けがゆるいと短絡、火災や誤動作の恐れがあります。
・STC6000内に、切粉や配線くずなどの異物が入らないように注意してください。火災、故障や誤動作の恐れがあります。

立ち上げ・保守時の注意事項

・運転中のプログラム変更、強制変更、RUN 、STOP 、PAUSE 等の操作はマニュアルおよびオンラインヘルプを熟読し、十分に安全を確認してから行って ください。操作ミスにより機械の破損や事故の原因になります。
・Ｉ/Ｏユニットの脱着は、通電されていないことを確認してから行ってください。通電中に行うとＩ/Ｏユニットの故障や誤動作の原因になります。
・液晶ディスプレイ内部には、刺激性物質が含まれています。万一、破損により液状の物質が流出し皮膚に付着した場合は、すぐに流水で15分以上洗浄してください。
また、目に入った場合は、すぐに流水で15分以上洗浄した後、医師に相談してください。

廃棄時の注意事項

・STC6000を廃棄する時は、産業廃棄物として扱ってください。

序章：表示器付きコントローラー「STC6000シリーズ」とは

Ⅰ テキストのご紹介 序-2

Ⅱ STC6000シリーズ/GP-Pro EXのご紹介 序-4

1.STC6000シリーズの特徴

2.STC6000シリーズのメリット

3.STC6000シリーズの採用事例

第1章：GP-Pro EXの基礎知識（ロジックプログラム）

Ⅰ 開発環境を整えよう 1-2

1.開発環境

Ⅱ GP-Pro EXの基礎知識 1-3

1.プロジェクトの作成方法

2.STC6000シリーズの画面構成

3.各部の名称

Ⅲ まずは、これだけ！基本操作！ 1-10

1.ロジック画面作成の基本操作

2.行の挿入

3.命令の割り付け

4.変数登録

5.作画画面との連携

6.デバッグ

7.I/O設定

第2章：基本回路をマスターしよう

Ⅰ 基本回路 その1 2-2

1.【理解しよう】基本回路その１

（ON回路/OFF回路/AND回路/OR回路/NOT回路）

2.【やってみよう】基本回路その１(自己保持回路)

Ⅱ 基本回路 その2 2-12

1.【理解しよう】基本回路その2

（インターロック回路/タイマ回路/カウンタ回路/SET・REST回路）

Ⅲ 基本回路 その3 2-20

1.【理解しよう】基本回路その3

（転送命令/比較演算命令）

第3章：応用回路をマスターしよう

Ⅰ　応用回路　3-2

1.【理解しよう】応用回路その１

（演算命令/加算演算/インクリメント・デクリメント・間接指定）

2.【理解しよう】応用回路その2

（プログラム制御/ジャンプ命令/サブルーチン処理）

第4章：ライン操作盤をつくろう

Ⅰ　実習について　4-2

Ⅱ　【やってみよう】コンベアを前進させよう　4-3

Ⅲ　【やってみよう】コンベアを後退させよう　4-9

Ⅳ　【やってみよう】インターロックをかけよう　4-13

Ⅴ　【やってみよう】準備時間を調整しよう　4-17

付録：さらに便利に使おう

Ⅰ　STC6000シリーズのメモリ構成　付-2

Ⅱ　変数のアドレス方式　付-5

Ⅲ　STC6000シリーズの動作処理　付-6

Ⅳ　オフラインモード　付-9

Ⅴ　用語集　付-12

Pro-face™
by Schneider Electric

序章

表示器付きコントローラー「STC6000シリーズ」とは

Ⅰ　テキストのご紹介　序-2
Ⅱ　STC6000シリーズ/GP-Pro EXのご紹介　序-4
1.STC6000シリーズの特徴
2.STC6000シリーズのメリット
3.STC6000シリーズの採用事例

メモ（以下は余白です。ご自由にお使いください）

Ⅰ　テキストのご紹介

（1）テキストについて

本テキストは、STC6000の基本的なロジックプログラム/作画方法を習得いただくためのテキストです。
ロジックプログラム/作画実習をユーザー様ご自身で効率良く行なっていただけるよう構成されています。
また、シミュレーション機能を使用すると、パソコン上でロジックプログラム/表示器画面の動作確認を行うことができます。

（2）テキストの使い方

本テキストは、序章から付録までで構成されております。各章ごとにテーマを設け、STC6000のさまざまな機能の設定方法を「理解しよう」「やってみよう」の２ステップで習得できる内容になっています。

理解しよう　⇒　やってみよう

理解しよう ：各章で使われている機能を説明しています。画面に、どのような機能/パーツが配置されているのか、その機能/パーツの設計手順や注意点の説明をしています。

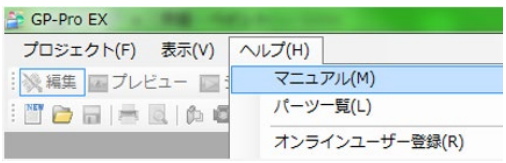
やってみよう：実習を通じて、機能/パーツの設定方法を習得いただきます。

※第2章から第4章の完成版サンプルプロジェクトをご用意しています。ロジックプログラムや画面の作成例を確認することができます。

ONE POINT

もっと詳しく知りたいときは

GP-Pro EXの操作等で困ったときは、マニュアルをご活用ください。
下図のように、GP-Pro EXのヘルプから開くことができます。



マニュアルは、以下の３種類を用意しています。

リファレンスマニュアル：GP-Pro EX（ソフトウェア）の設定手順や各設定項目の詳細が知りたいときにご覧ください。
接続機器マニュアル：表示器と接続機器との接続方法についてご説明します。
ハードウェアマニュアル：表示器の仕様や取り付け方法についてご説明します。

（3）テキストの構成

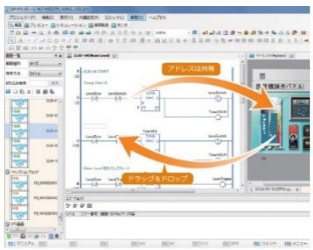
本テキストでは、各章ごとに用途別の画面を作画します。

序章：表示器付きコントローラー「STC6000シリーズ」とは

STC6000シリーズの特徴やメリット、採用事例を紹介してします。
（実習では、この章は行いません）

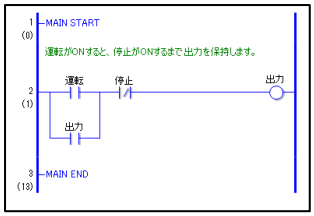
第１章：GP-Pro EXの基礎知識（ロジックプログラム）

GP-Pro EXを使ったSTC6000シリーズの画面作成方法の基礎知識を説明しています。



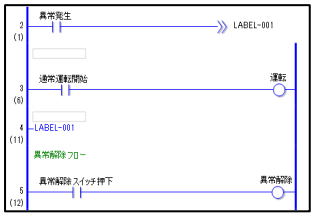
第２章：基本回路をマスターしよう

GP-Pro EX（ロジック）を使った基本回路の作成方法を習得できます。



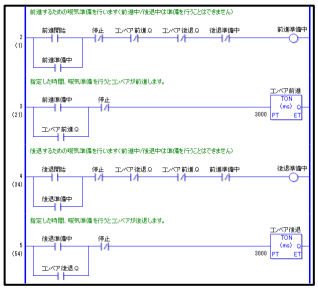
第３章：応用回路をマスターしよう

GP-Pro EX（ロジック）を使った応用回路の作成方法を習得できます。



第４章ライン操作盤をつくろう

ライン操作盤の作成を通じて、GP-Pro EX（ロジックプログラム）の実践的な使い方を習得できます。



付録：

STC6000シリーズ/GP-Pro EXのさらに便利な使い方や機能を紹介しています。

※第2章から第4章の完成版サンプルプロジェクトをご用意しています。ロジックプログラムや画面の作成例を確認することができます。

Ⅱ　STC6000シリーズ/GP-Pro EXのご紹介

1.STC6000シリーズの特徴

(1)STC6000シリーズとは

STC6000シリーズ(以下STC6000)とは、表示/操作/制御を一体化した表示器付きコントローラーです。DIO(入力:16点 / 出力:16点）を内蔵しており、各種I/O機器と直接接続できます。また、オプションのTM3モジュールを用いることで、DIOやAIOを拡張することが可能です。

STC6000の主な特長として以下のものがあります。

①コントロール機能内蔵

・STC6000本体内に制御プログラムを作成、実行できます。
・命令は厳選された120種類以上をサポート
・ロジックの作成仕様は国際標準規格IEC61131-3に準拠

②豊富な接続機器

・入出力機能内蔵（拡張可能）
・内蔵DIO以外にもEthernet/シリアル/USBでさまざまな機器とダイレクト接続できます。
・TM3モジュールを用いることで、DIOやAIOを拡張することができます。

③装置の付加価値アップ

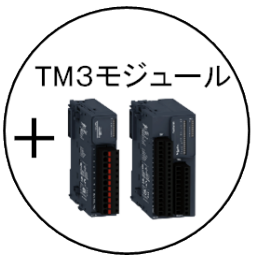
アルミベゼル採用し、スタイリッシュな見た目で、装置デザインの向上に貢献します。

④使うソフトは一本のみ

画面作成もロジックプログラム（ラダー）もGP-Pro EXひとつでできます。ソフト一本だから連携/管理が簡単です。

⑤サイバーセキュリティーを強化

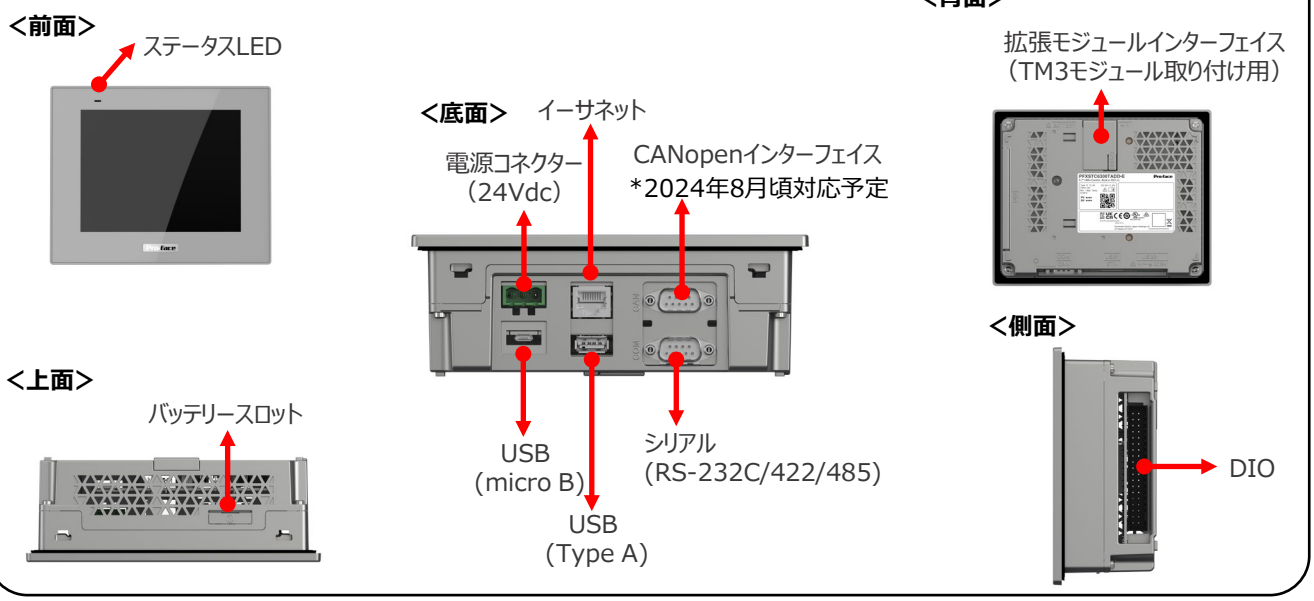
起動時に、悪意のあるソフトウェア等による問題がないか、セキュリティーチェックを行います。



5.7型 / VGA / 262,144色

ONE POINT

STC6000の外形(インターフェイスの位置)

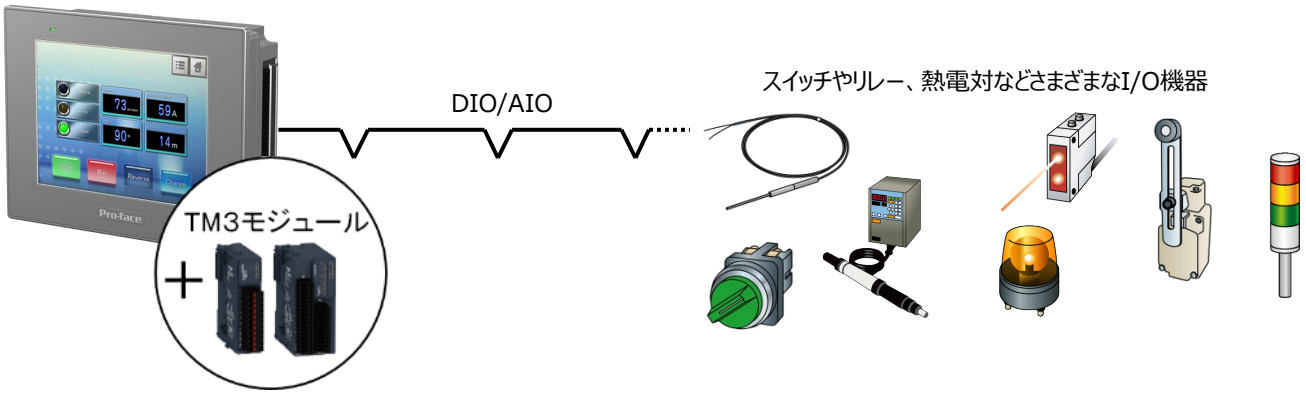


（2）コントローラー一体型で様々なシステム構築が可能

運転時のシステム構成には次のようなものがあります

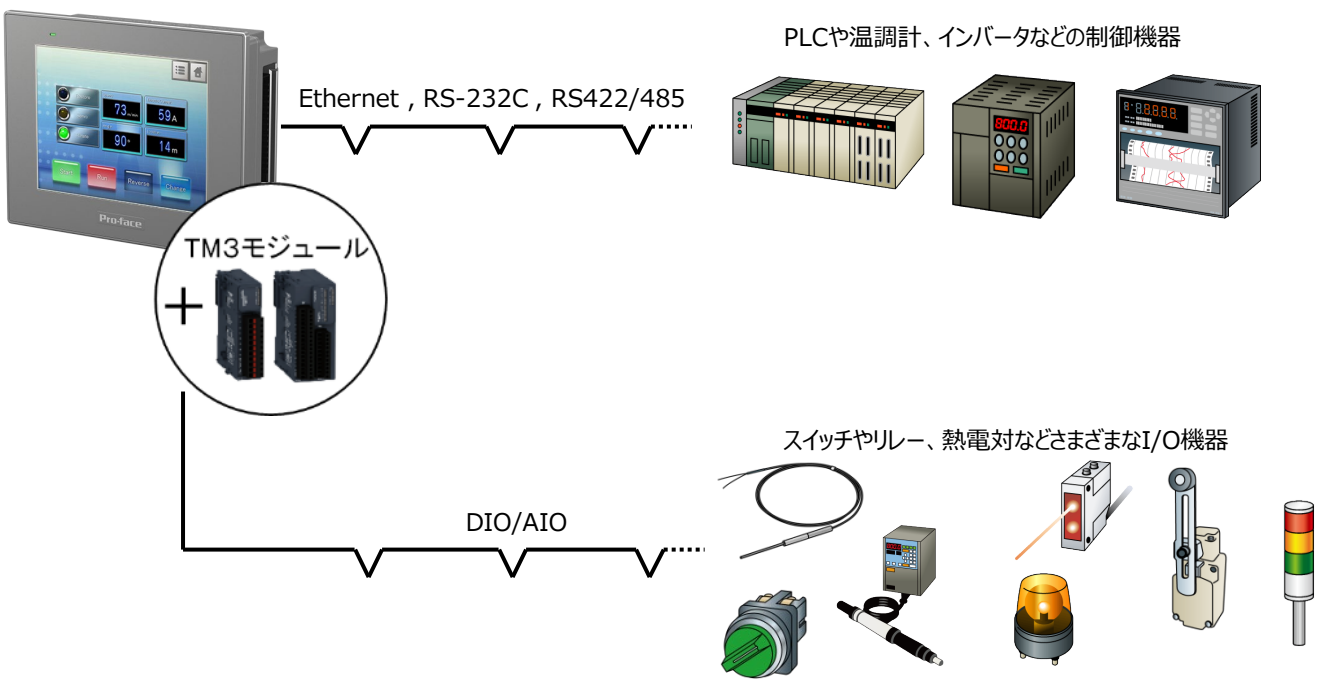
①単独システム

外部の制御機器を利用せず、STC6000だけですべての制御を行います。



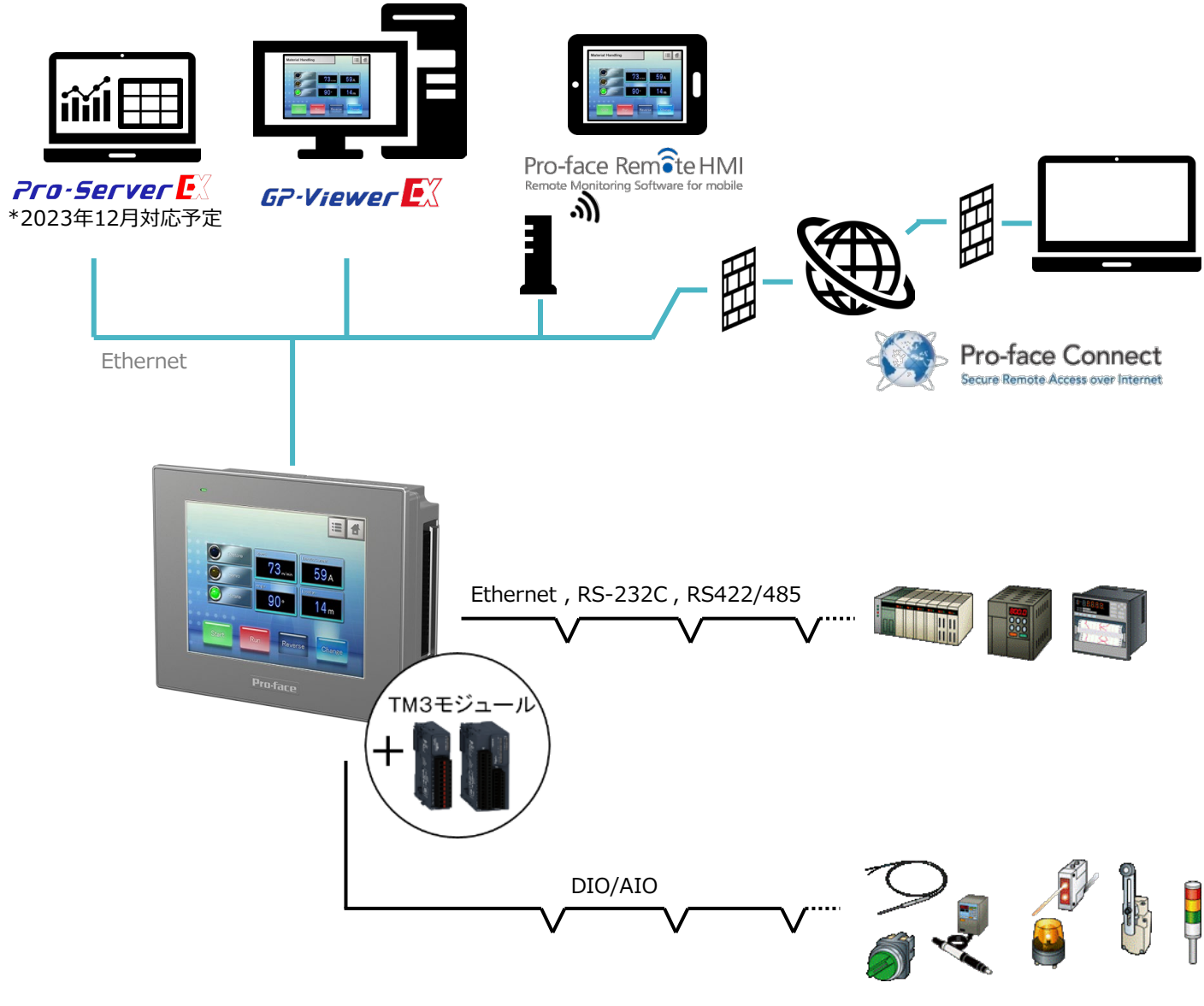
②外部制御機器接続システム

外部制御機器の顔として各種パラメータの表示/設定を行いながら、サブコントローラーとして追加プログラムを処理するといった使い方ができます。



③遠隔監視 / データ集約システム

PCやタブレットなどのデバイスを使った遠隔監視や接続機器のデータを集約して上位のパソコンへ送信するなど、エッジ端末やゲートウェイ機器として利用できます。



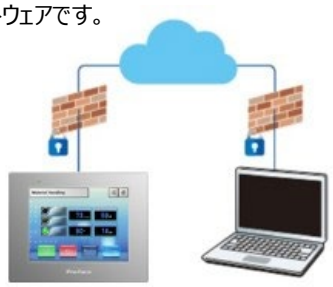
ONE POINT

セキュアリモートアクセス　Pro-face Connectとは

「Pro-face Connect」は、わずかな工数・コストで遠隔監視の実現をサポートするソフトウェアです。

Pro-face Connect を使用すれば、STC6000ががVPNルーター内蔵表示器になり、わずかな工数・コストでセキュリティー面で安全な遠隔監視・リモートメンテナンスを実現することができます。

遠隔監視サービス（保守/メンテナンス）の新たなビジネス展開や、立ち上げ時やトラブル時の対応もリモートで実現できるので、電話対応だけでは特定が難しい内容も、効率的に対応出来るので、出張コストやサポートコストの削減にも貢献します。

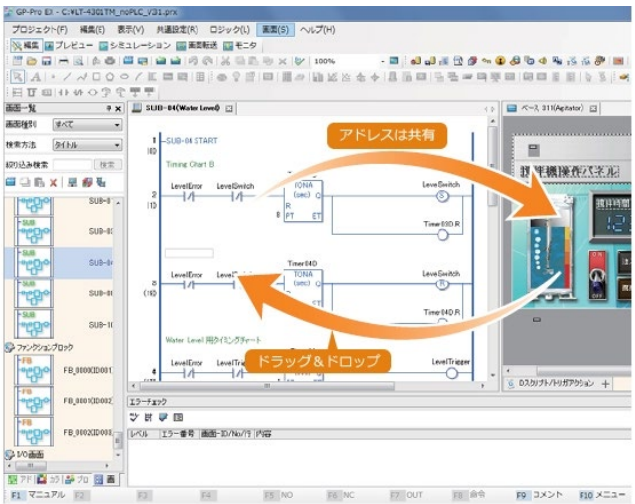


2. STC6000シリーズのメリット

（1）開発でのメリット

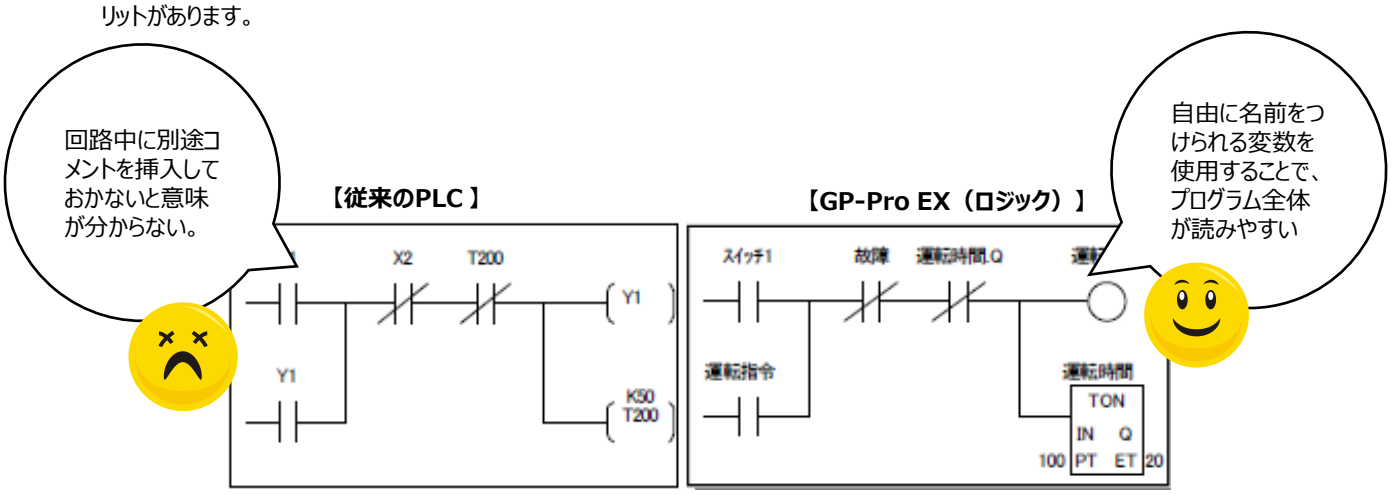
① 簡単な画面データ／ロジックプログラムの作成

作画とロジックプログラムの連携が可能で、スイッチやランプなどのパーツや命令、変数をドラッグ＆ドロップするだけで、配置や挿入ができます。画面やラダープログラムを効率的に作成でき、開発工数の削減に役立ちます。



② コメント不要で読みやすい

GP-Pro EX（ロジック）では自由に名前を付けられる変数を使用します。
それぞれの変数に自由に名前が付けられることで、ロジックプログラムの管理がしやすくなり、プログラム全体が読みやすくなるというメリットがあります。



ONE POINT

国際標準規格IEC61131-3準拠

国際電気標準会議（IEC）がPLC用のプログラム言語を定義した世界共通の標準規格です。国際標準規格IEC61131-3に準拠していることにより、GP-Pro EXには下記のような特長があります。

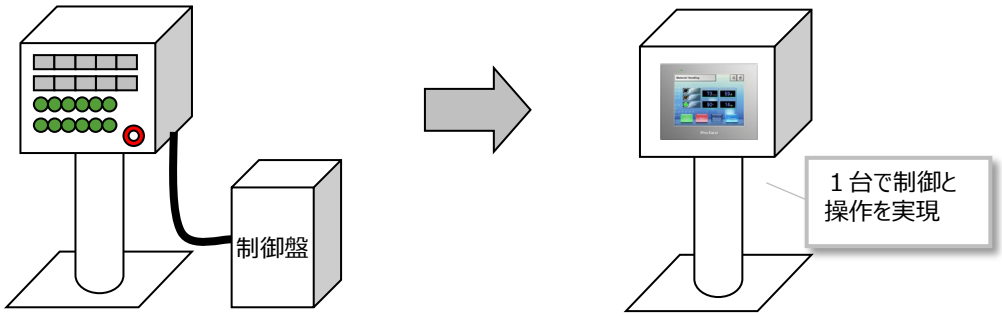
・制御システムの開発において最も普及しているラダー方式で、わかりやすい（命令語の形態や操作方法が共通）

・ファンクションブロック方式もサポートしており、電子回路を設計するように記述できるため、データの流れが直感的でわかりやすい

・デバイスアドレスを変数として作成できるため、コメント作成労力も不要で、管理もしやすい

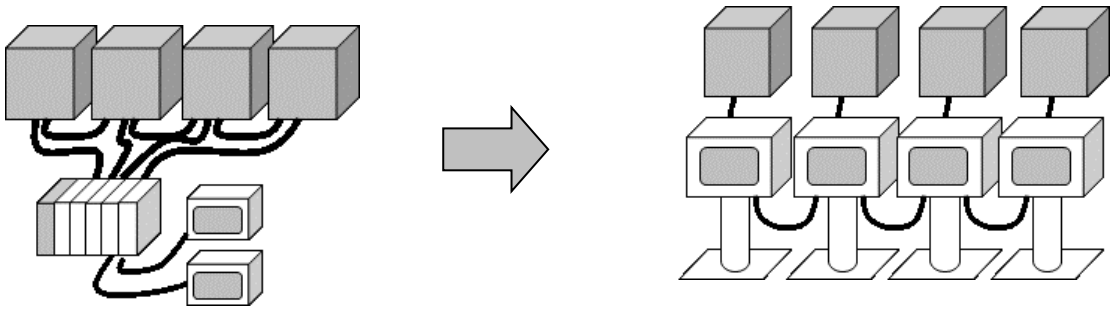
④ オールインワンでローコストな操作・制御盤製作を支援

操作盤機能と制御盤機能が一体になっているので、別々に盤を用意したり配線・設置場所を気にする必要がありません。
そのため、見えない部分で発生していた工数や費用を削減でき、コストを抑えられます。



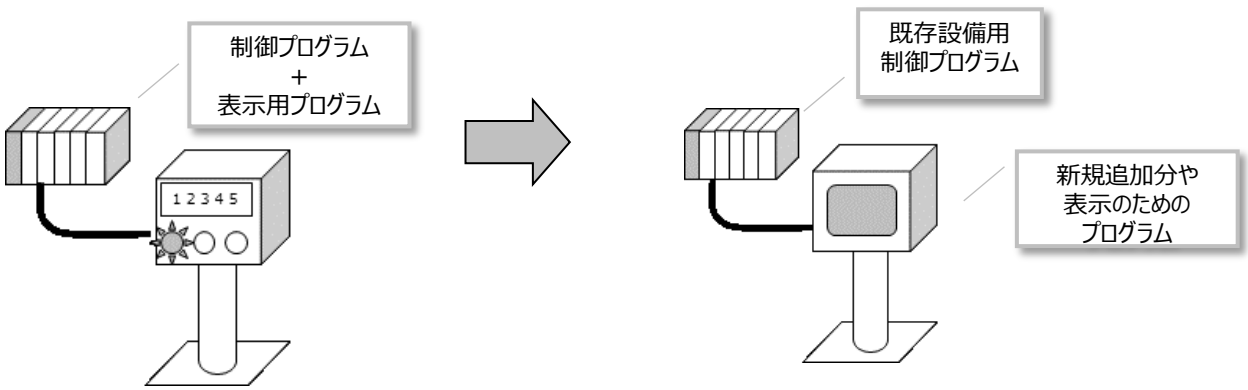
⑤ 設計の簡略化を実現

ユニットごとに完結するモジュール構造にすることができ、分散して制御が可能です。そのため、構成をシンプルにでき、設置・搬送の際に組み立て・レイアウト変更が簡単で、部分的な交換も可能です。



⑥ PLCの負担を軽減

STC6000内部でロジックを組めるので、本来の制御に関係のない表示のためのプログラムをPLCで組む必要はありません。
また、既存の設備に改造を加える際にも追加のサブコントローラーとして利用していただけば、既に稼動しているプログラムの修正は最小限度で済み、工数が削減できます。



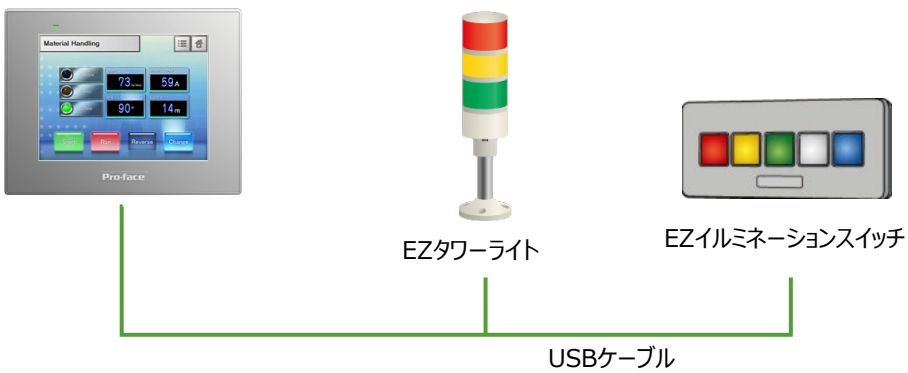
⑦ リモートI/O

STC6000は、TM3送信モジュール/受信モジュールを用いることで最大5Mのリモート I/O環境を構築することができます。
省配線やシステム設計の簡素化、メンテナンス性の向上などのメリットがあります。

*2023年12月予定のGP-Pro EX Ver. 4.09.550以降、最大7台まで接続可能

⑧ 積層式表示灯やスイッチ等の後付け/拡張が簡単

EZシリーズとUSBケーブル1本で接続できるので、積層式表示灯や外部スイッチの後付けや拡張が簡単です。



ONE POINT

Modicon TM3モジュール

Modicon TM3モジュールとは、STC6000にDIOやAIO等を拡張するためのモジュールです。DIO用モジュール(入力/出力/入出力混合)とアナログ用モジュール(入力/出力/入出力混合)を用意しています。

STC6000には、背面には最大2個、TM3送信モジュール/受信モジュールを使用すれば最大3個*のTM3モジュールを追加接続することができます。

Modicon TM3モジュール：https://www.proface.com/ja/product/option/tm3

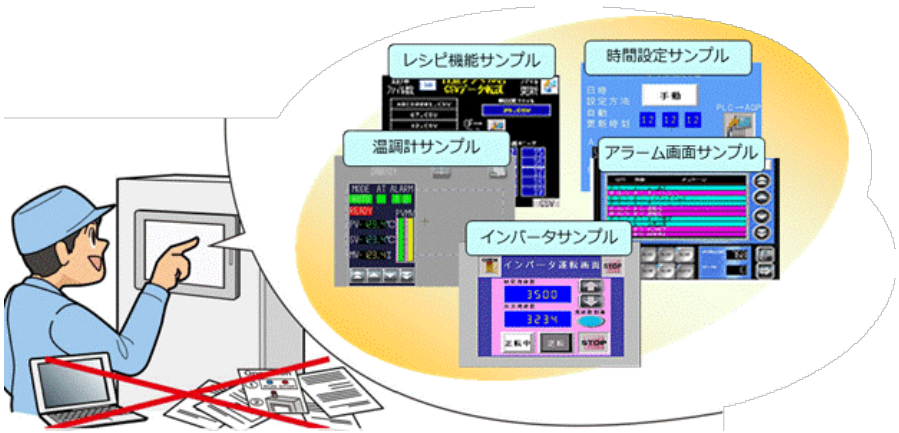
*2023年12月予定のGP-Pro EX Ver. 4.09.550以降、最大7台まで接続可能

(2)運用でのメリット

(3)インバータや調節計等の操作やアラーム表示を集約し、操作性を向上

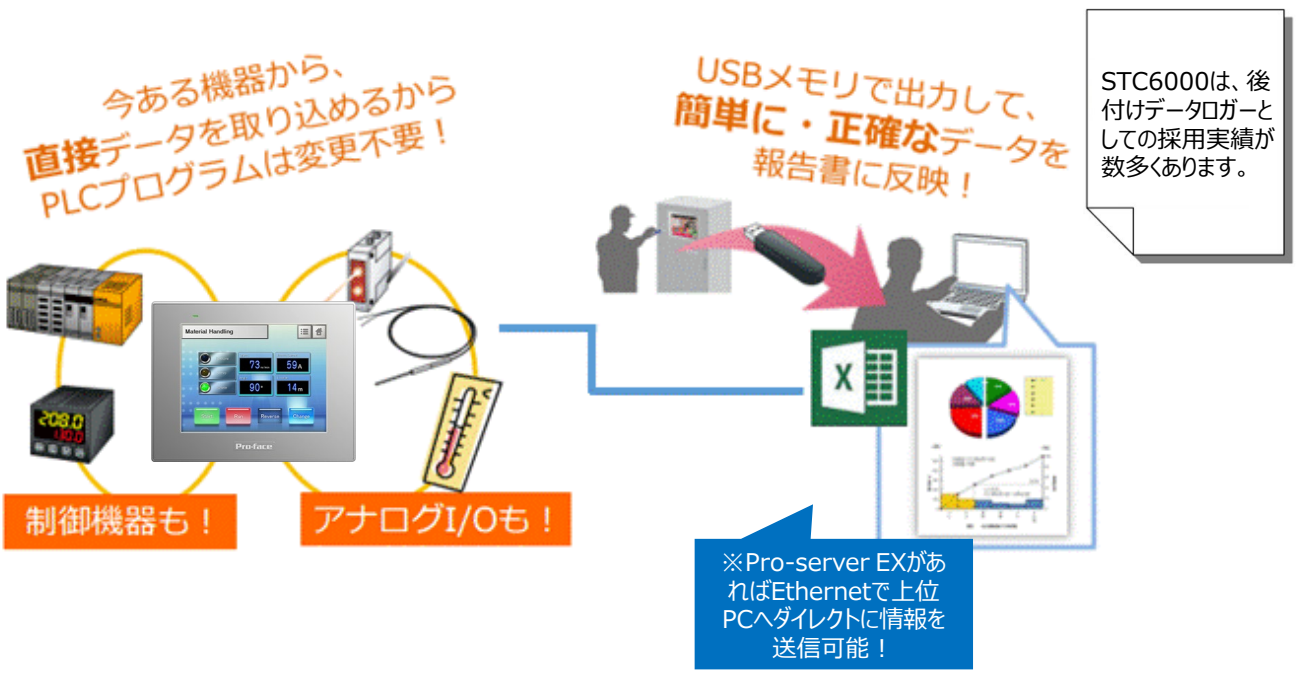
インバータや温調計をはじめ、サーボドライブ、ロボシリンダ、ロボットコントローラー等のよく使う機能を盛り込んだサンプル「コクピットパーツ」*を活用すると、今まで操作盤を開けての操作や専用ツールを用いての作業が、STC6000の画面上から行うことができ、操作性が向上します。また、アラームメッセージの内容や対処方法もSTC6000上に集約することでいつでも必要な情報を入手できます。

*STC6000で使用できるサンプル「コクピットパーツ」は、ST6000シリーズ用のものです。
また、「コクピットパーツ」をSTC6000で使用するには、「コクピットパーツ」であらかじめ、機種変更(STC-6300TA VGA 640＊480) を行ってください。



②トレーサビリティーの向上

STC6000は様々な機器と直接接続することにより、作業情報や装置情報から生産技能や製品品質向上に関わる情報の収集が簡単にできます。



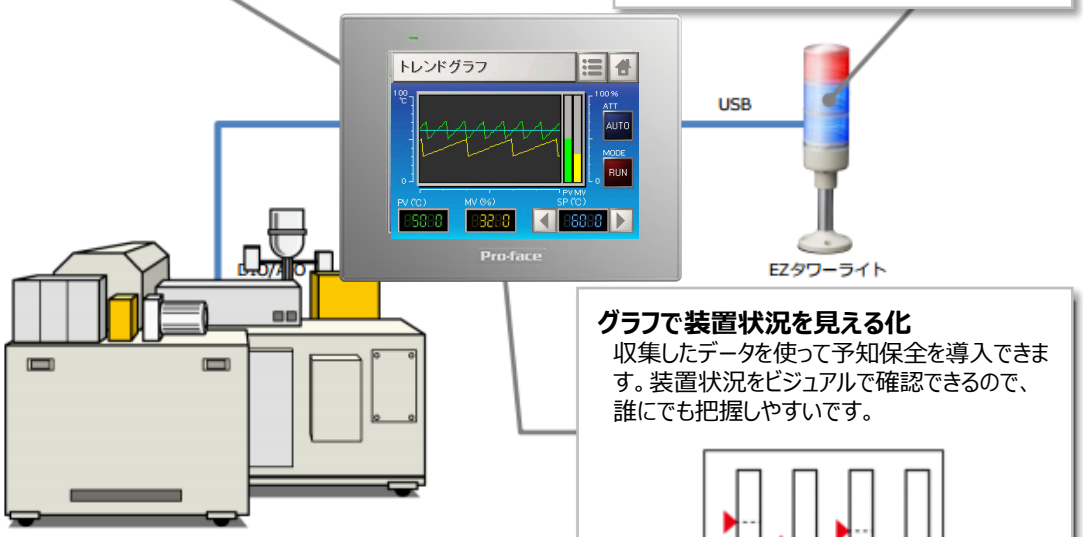
ONE POINT

予防保全に最適なSTC6000シリーズ

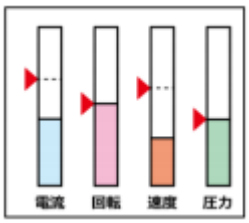
既設装置の改善活動を行うユーザ様にお勧めなのがSTC6000です。自動で装置の正常な状態を測定・記録を行うことで、異常の予兆を察知してメンテナンスすることができます。これにより急な装置停止を未然に防げるため、予知保全用に既設装置への採用実績が数多くあります。

後付けで簡単に予知保全
I/O内蔵だから直接機器につながります。PLCプログラムを改造せず、後付けで簡単にデータを収集できます。

目につきやすい表示色で注意喚起
EZタワーライトは表示色を現場で自由に変更できます。現場であまり使われていない色を選ぶことで、異常値の検出を目立たせることができます。



グラフで装置状況を見える化
収集したデータを使って予知保全を導入できます。装置状況をビジュアルで確認できるので、誰にでも把握しやすいです。

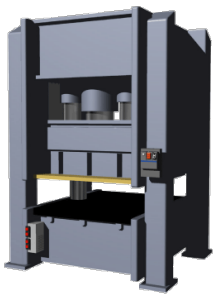


3. STC6000シリーズの採用事例

I/O内蔵でかつ本体が小さく、それでいてドライバサポート数が多いので、自動車、食品、医療、公共、化学と、さまざまな分野のさまざまな「小型装置」への採用実績が多数あります。

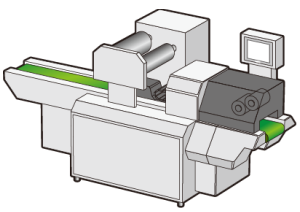
（1）プレス機/切断機

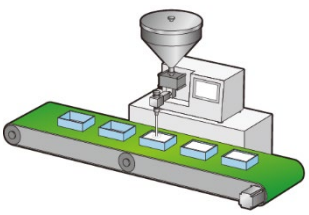
プレス機等のシンプルな装置では、STC6000を使用し、消耗品やメンテナンスパーツの使用回数や稼働時間をカウントし、メンテナンスにお役立ていただけます。また、STC6000は見やすいTFTカラー表示にアナログタッチパネルを標準装備しています。



（2）食品製造装置

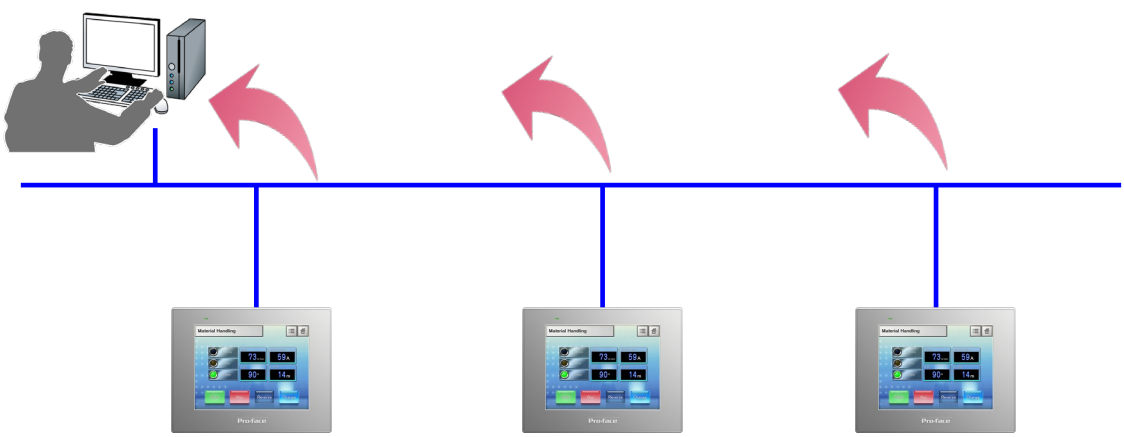
食品の製造装置において、センサーや機器のコントロールをSTC6000単体で行うことで、コストダウンと装置の小型化が実現できます。また、STC6000本体とTM3モジュールを分離接続(リモートI/O)できるので、小型装置だけでなく、中規模の装置でも使用することができます。





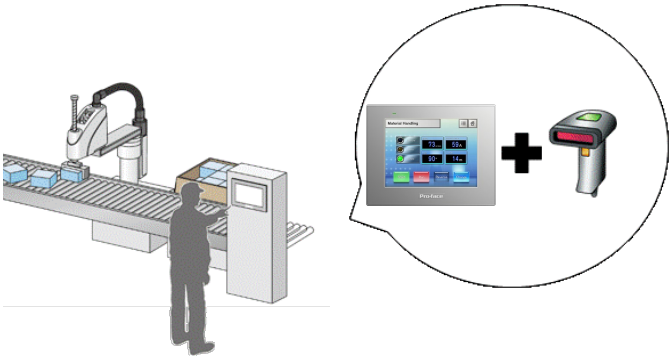
（3）小型工作機

STC6000はコンパクトなので組み込み用途に適し、独立したコントローラーとして利用できます。
また、STC6000と上位パソコンを接続することにより、必要に応じて、パソコンからSTC6000の内部データを吸い上げたり、STC6000へデータ書き込みも可能です。
※データの収集には別途Pro-Server EXが必要です。



（5）物流システム

I/Oの追加の際に全体のI/O構成を考えながら追加しなくても大丈夫です（ブザーやランプなど）STC6000に追加するだけでOKなので、柔軟性のあるシステム構築ができます。
また、STC6000には、演算、判断機能が入っているため、バーコードで読み取った製品情報を、STC6000のみの機能で生産物の仕分けや不良品選別などの端末としても最適です。



メモ（以下は余白です。ご自由にお使いください）

Pro-face™
by Schneider Electric

第1章

GP-Pro EXの基礎知識（ロジックプログラム）

Ⅰ　開発環境を整えよう　1-2
　1.開発環境
Ⅱ　GP-Pro EXの基礎知識　1-3
　1.プロジェクトの作成方法
　2.STC6000シリーズの画面構成
　3.各部の名称
Ⅲ　まずは、これだけ！基本操作！　1-10
　1.ロジック画面作成の基本操作
　2.行の挿入
　3.命令の割り付け
　4.変数登録
　5.作画画面との連携
　6.デバッグ
　7.I/O設定

メモ（以下は余白です。ご自由にお使いください）

I　開発環境を整えよう

1.開発環境

STC6000の画面とロジック作成には、以下の４つをご用意ください。

Windows
パソコン

GP-Pro EX
※V4.09.500以降

転送用ケーブル

STC6000シリーズ

（1）GP-Pro EX

GP-Pro EXは、STC6000の画面とロジックプログラム、どちらも作成できるソフトウェアです。
※STC6000は、GP-Pro EX Ver4.09.500以降に対応しています。

（2）パソコン

GP-Pro EXは以下の要件を満たしたパソコンにインストールしてください。
Windows PC（GP-Pro EXが快適に動作すること）

（3）転送用ケーブル

プロジェクトの転送方法は３つあります。

USBケーブル
（A-micro B）

オプションのUSBデータ転送ケーブル（PFXZUSCBMB2）または、
市販のUSBケーブル(Type A to micro B)を用いて転送することができます。

LANケーブル

LANケーブル（イーサネットケーブル）を用いて転送することができます。

USBストレージ

プロジェクトファイルを外部ストレージデバイス経由でパソコンから表示器に送受信することができます。

（4）STC6000シリーズ

STC6000とは、表示/操作/制御を一体化した表示器付きコントローラーです。DIO（入力:16点 / 出力:16点）IOを内蔵しており、各種I/O機器と直接接続できます。液晶サイズは5.7型で、シンク出力タイプとソース出力タイプが用意されています。

STC6000は標準でイーサネットのI/Fが付いています。

5.7型 / VGA / 262,144色

製品名：STC-6300TA（シンク）
型式：PFXSTC6300TADDKE

製品名：STC-6300TA（ソース）
型式：PFXSTC6300TADDCE

Ⅱ GP-Pro EXの基礎知識

1.プロジェクトの作成方法

（1）プロジェクト(*.prx)とは

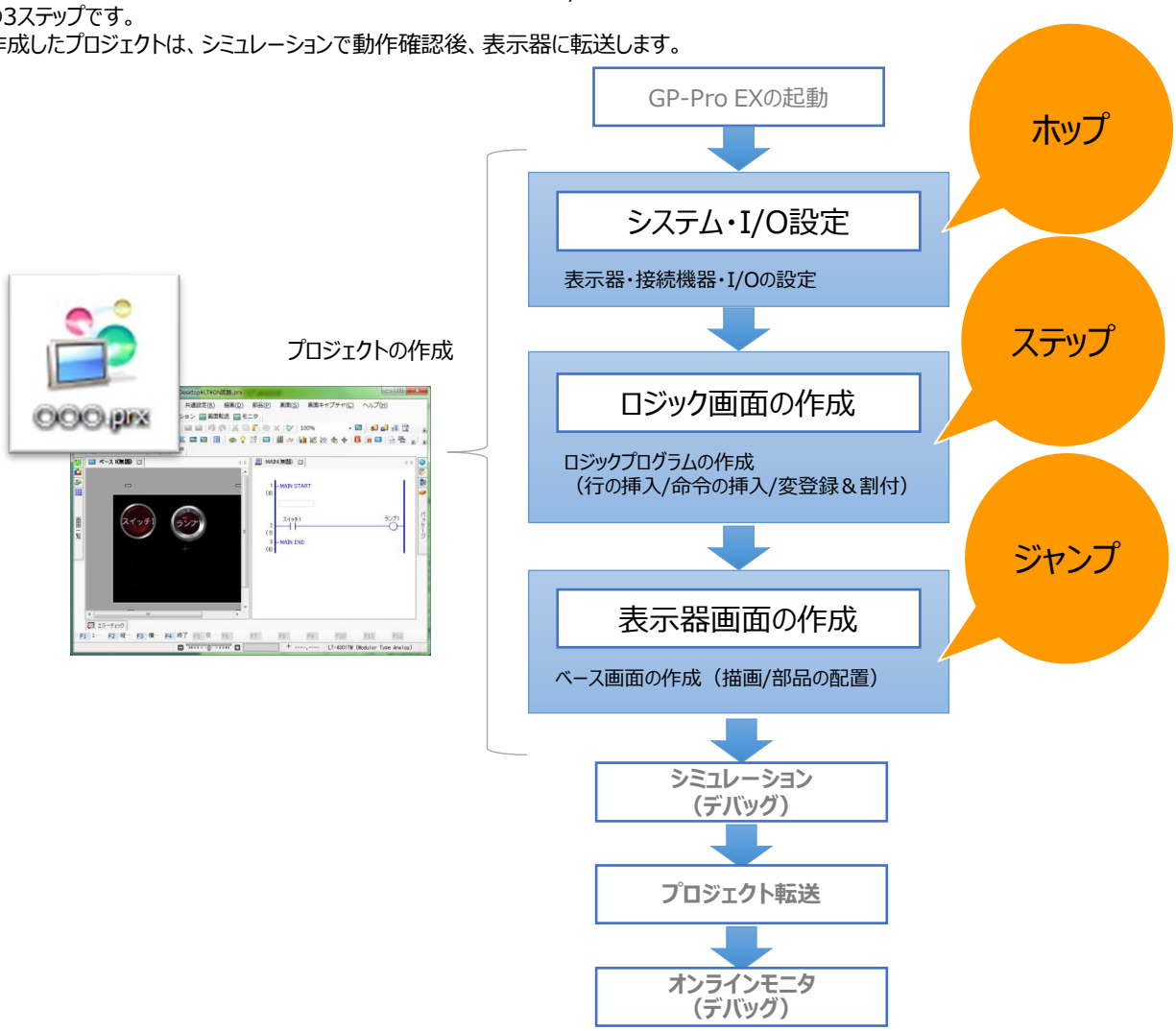
プロジェクトとは「画面データ」「ロジックプログラム」「通信プロトコル」「システム」を一つのまとめたものです。
プロジェクトはパソコン上に「.prx」の拡張子で保存されます。

GP-Pro EXで作ったプロジェクトを表示器に転送することで、
STC6000はさまざまなデータの表示や各種操作および制御、I/O入出力を行ないます。

〇〇〇.prx

（2）プロジェクトの作成の流れ

プロジェクト作成の手順は、GP-Pro EXの起動後、「システム・I/O設定」→「ロジック画面の作成」→「表示器画面の作成」の3ステップです。
作成したプロジェクトは、シミュレーションで動作確認後、表示器に転送します。



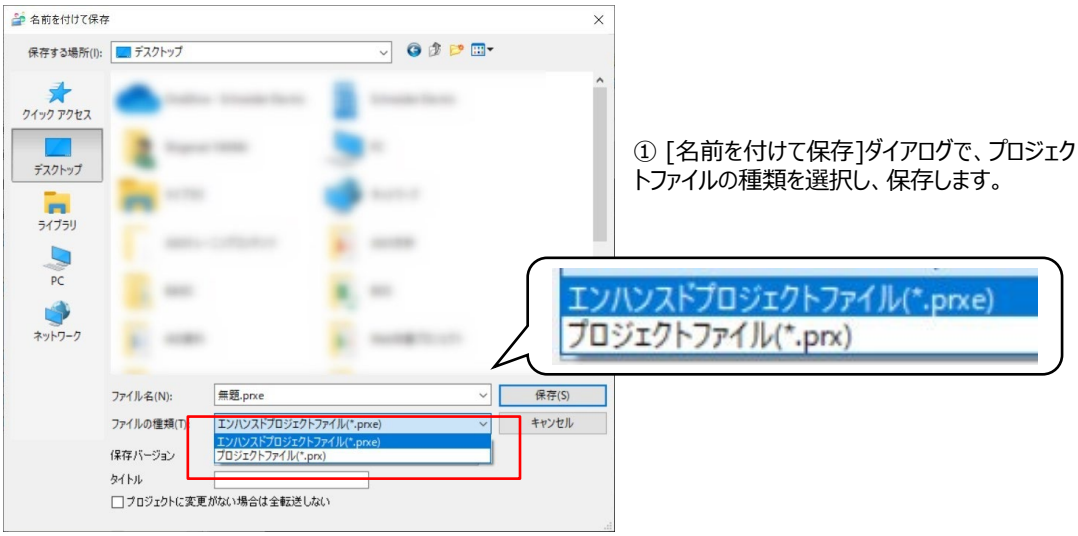
ONE POINT

エンハンスドプロジェクトファイル(.prxe)

エンハンスドプロジェクトファイル(.prxe)とは、セキュリティーを強化したプロジェクトファイルです。プロジェクトファイル全体を暗号化して保存します。
エンハンスドプロジェクトファイルは、GP-Pro EX Ver.4.09.450以上で使用することができます。

GP-Pro EXで、エンハンスドプロジェクトファイル形式で保存する場合は、[名前を付けて保存]をクリックして、[ファイルの種類]で[エンハンスドプロジェクトファイル(*.prxe)]を選択し、保存します。
なお、エンハンスドプロジェクトファイル(.prxe)または従来のプロジェクトファイル(.prx)のどちらで保存されても、表示器本体の動作や機能に違いはありません。

本テキストでは従来のプロジェクトファイル(.prx)で解説していますが、実際の使用においては、セキュリティーを強化したエンハンスドプロジェクトファイル(.prxe)の使用を推奨します。



① [名前を付けて保存]ダイアログで、プロジェクトファイルの種類を選択し、保存します。

【エンハンスドプロジェクトファイルとプロジェクトファイル】

	エンハンスドプロジェクトファイル	従来のプロジェクトファイル
	○○○.prxe	○○○.prx
拡張子	.prxe	.prx
対応GP-Pro EXバージョン	Ver.4.09.450～	Ver.1.0～ (Ver.4.09.450以降も利用可能)
暗号化	暗号化	非暗号化

（3）新規プロジェクトの作り方

新規プロジェクトは、システム設定（表示器タイプ・接続機器タイプ・I/O設定）を行った後は、エディタ上部にある「状態バー」に沿って、設定を進めます。

ここでは、以下の環境を例にシステム設定を行います。

・表示器付きコントローラー：STC-6300TA（シンク）
※型式 PFXSTC6300TADDKE
・接続機器：デジタル/メモリリンク（STC6000単独システム）

①GP-Pro EXの起動

デスクトップのGP-Pro EXのアイコンをダブルクリックします。

GP-Pro EX

新規プロジェクトを作成したら、名前を付けて保存してください。

②表示器タイプ

ウィザードに従い、表示器タイプを指定します。

シリーズ：STC6000Series
STC63※※Series（表示器の大きさ）
機種： STC-6300TA VGA(640*480)
設置方法：横型

③接続機器設定

接続機器を指定します。
STC6000を単独で使用する場合は、以下の設定を行います。

メーカー：デジタル
シリーズ：メモリリンク
ポート：COM1

④I/Oの設定

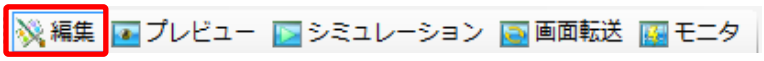
TM3モジュールを用いない場合は、「無」を選択します。

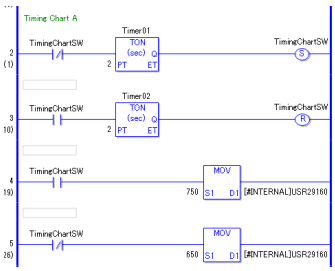
I/Oドライバ：無

⑤ロジック画面の作成

「状態バー」を「編集」にし、ロジック画面にて、ロジックプログラムを作成します。

【状態バー】

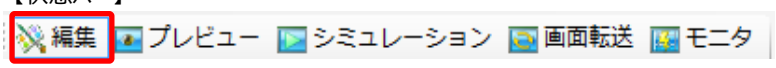


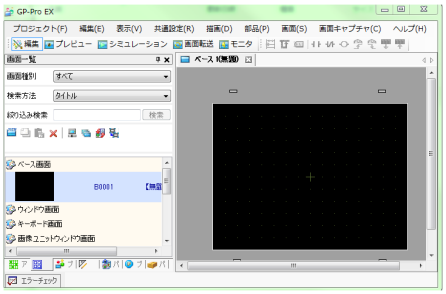


⑥ベース画面の作成

「状態バー」を「編集」にし、ベース画面で描画/部品を使って、表示器画面を作成します。

【状態バー】

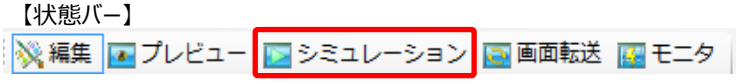


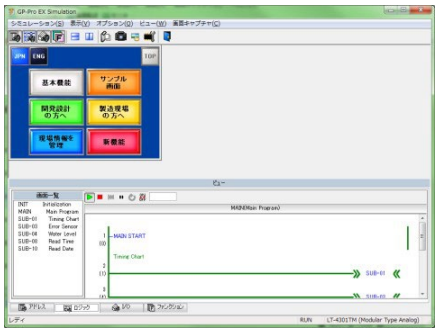


⑦シミュレーション（デバッグ）

シミュレーション機能を使い、パソコン上で動作確認します。
表示器画面だけでなく、ロジックプログラムの動作も確認できます。

【状態バー】

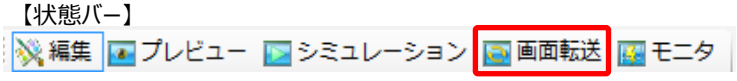




⑧画面転送

STC6000とパソコンを転送ケーブルでつなぎ、プロジェクトを転送します。
デフォルト設定は、USBケーブル転送です。

【状態バー】

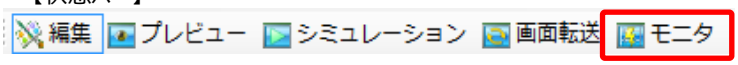


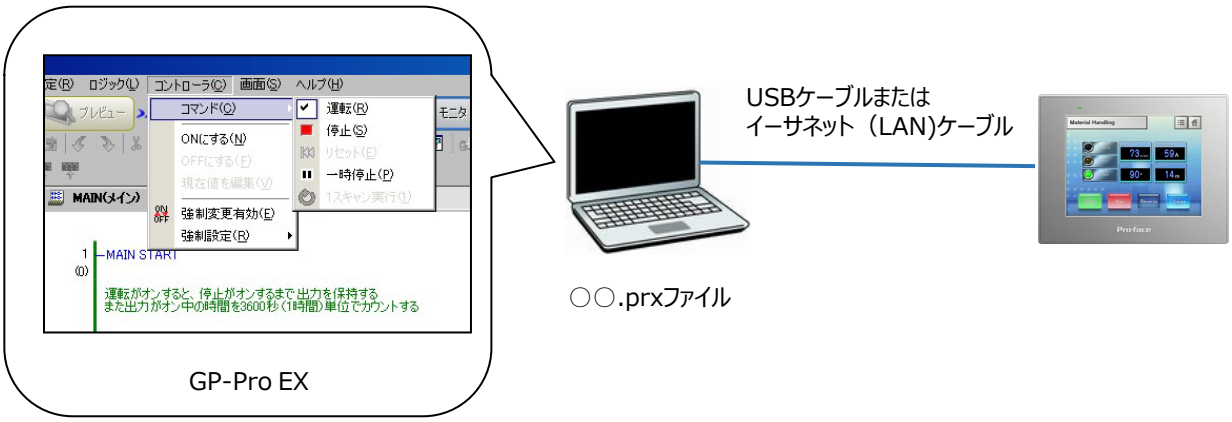
○○.prx

⑨オンラインモニタ（デバッグ）

オンラインモニタを使い、運転中のSTC6000内のロジックプログラムをパソコン上でモニターを行い動作確認を行います。

【状態バー】



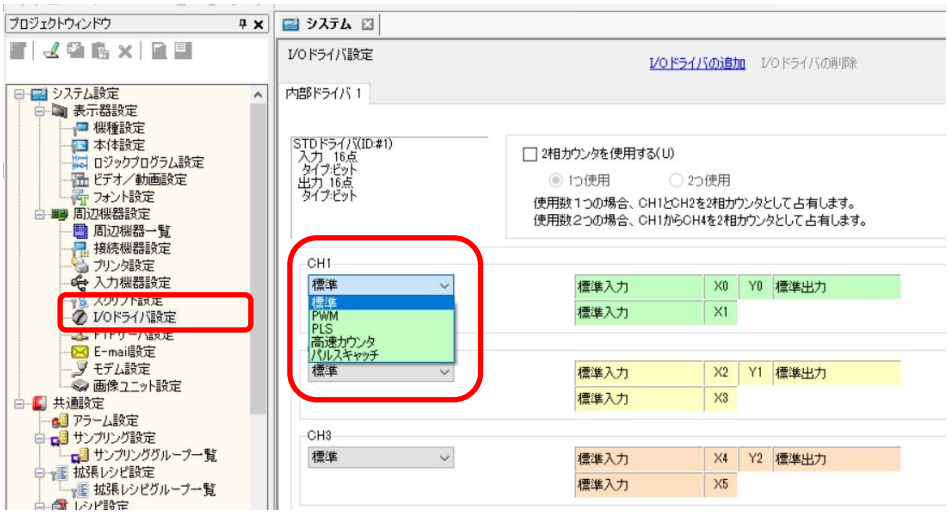


ONE POINT

特殊I/O

STC6000は、標準入出力以外に、下記のような特殊I/Oを持っています。特殊I/Oを活用することで、さらに高度な制御が可能です。特殊I/Oを設定する場合は、プロジェクトウィンドウ→I/Oドライバ設定にて、各設定を行います。

【特殊I/O 】
PID命令（PWM出力版）
PWM出力
パルス列出力
高速カウンタ（一致出力機能あり）



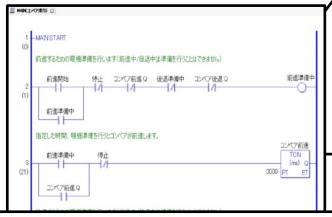
2.STC6000シリーズの画面構成

（1）STC6000シリーズの画面構成

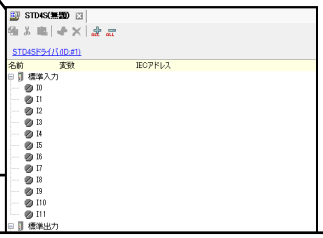
STC6000は、ベース画面（表示/操作）、ロジック画面（制御）、I/O画面（入出力）の3つの画面で構成されています。
それぞれの画面では、共有の変数を使用します。

ベース画面（表示/操作）

描画/部品を使って、表示や操作部分を作成します。
変数の状態を表示したり、ベース画面上のスイッチ等
を使って、操作することができます。



変数



ロジック画面（制御）

演算、判断、条件をラダー形式で作成します。
STC6000ではロジック画面のプログラムを実行します。

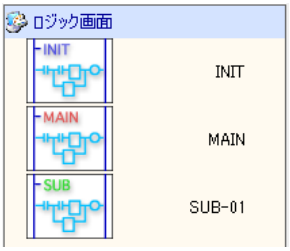
I/O画面（入出力）

変数の外部入出力への割付けを行います。

ONE POINT

ロジック画面の種類

ロジック画面には、以下の種類があります。



INIT画面：ロジックスタート後の最初の１スキャンだけ処理を行います。このエリアを利用して初期値の設定などを行います。

MAIN画面：メインのロジック処理のエリアです。STARTからENDに記述された内容を繰り返し処理を行います。

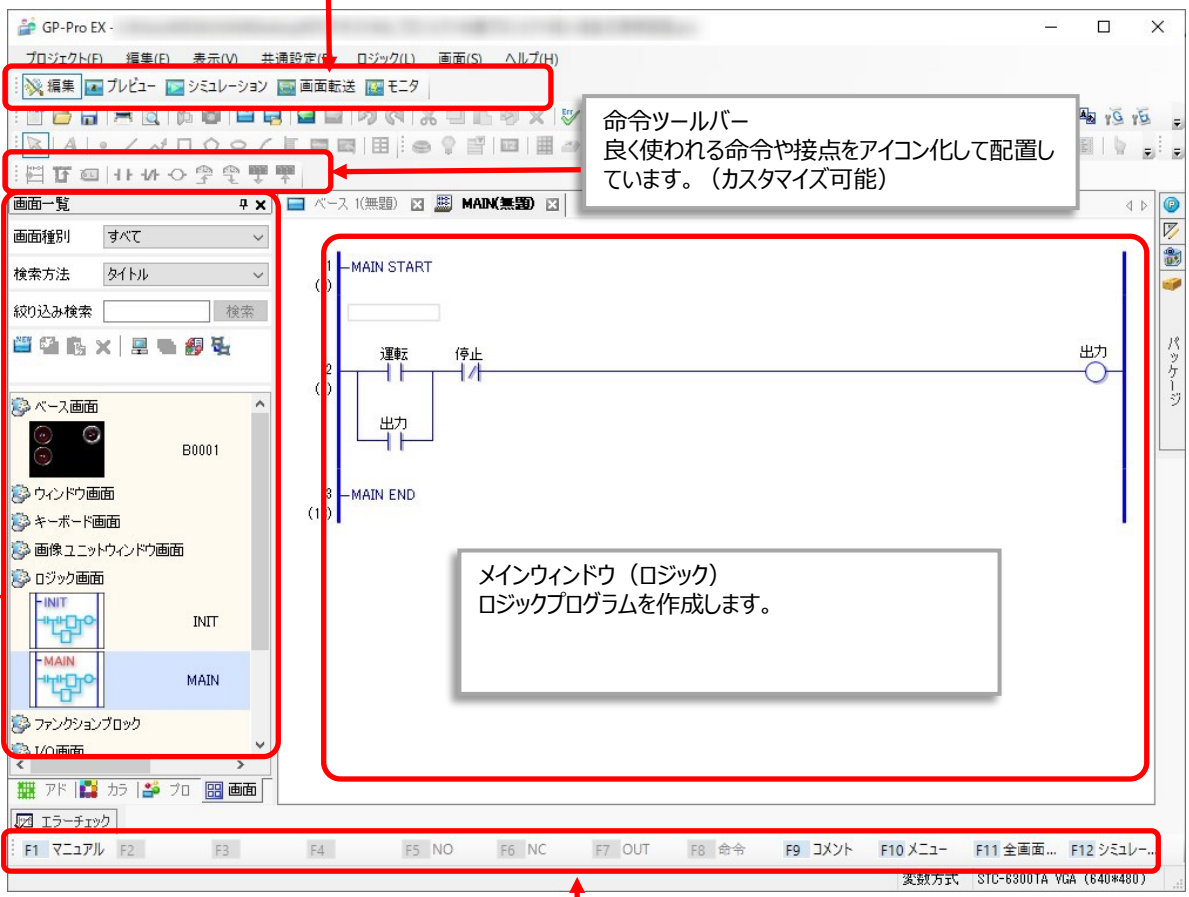
サブルーチン（SUB）：メインのプログラム中で呼び出されるサブルーチン（特定の処理のための小プログラム）を 記述します。サブルーチンを使用するにはメインプログラム中で「JSR」命令を利用し、ジャンプ先のサブルーチン画面を作成します。

ファンクションブロック：よく使うプログラムをブロック化して登録できます。ブロック化することで、よく利用する処理プログラムの作成時間短縮や、プログラム容量の節約ができます。また、ブロック化したデータにパスワードをかけることで、プログラム情報の保護が可能です。

3.各部の名称

（1）ロジック画面

状態バー
編集、プレビュー、シミュレーション、画面転送、モニタ、と左から右へ開発手順に沿ってアイコンが並んでいます。



命令ツールバー
良く使われる命令や接点をアイコン化して配置しています。（カスタマイズ可能）

メインウィンドウ（ロジック）
ロジックプログラムを作成します。

画面一覧ウィンドウ
既存の画面をサムネイルから直感的に選択し、開くことができます。

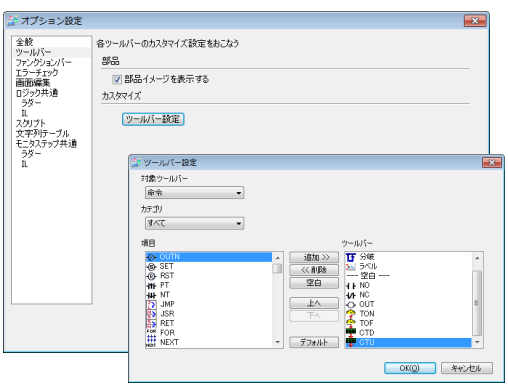
ファンクションスイッチ
ロジックプログラムの各操作でファンクションキー、ショートカットキーが使用できます。マウスに持ち替えず、キーボード作業することができます。

ONE POINT

ツールバーのカスタマイズ

命令ツールバーは使い勝手にあわせて、自由にカスタマイズすることができます。

①メニューバーの「表示」から「オプション設定」をクリックしてください。
②オプション設定ダイアログボックスが表示されます。
　左側のリストで「ツールバー」を選択し、「ツールバー設定」をクリックします。
③「ツールバー設定」ダイアログボックスで、対象ツールバーを指定してください。
④アイコン表示したいものを選択して追加をクリックします。
　逆に、不必要なものは、選択して削除をクリックします。



Ⅲ まずは、これだけ！基本操作！

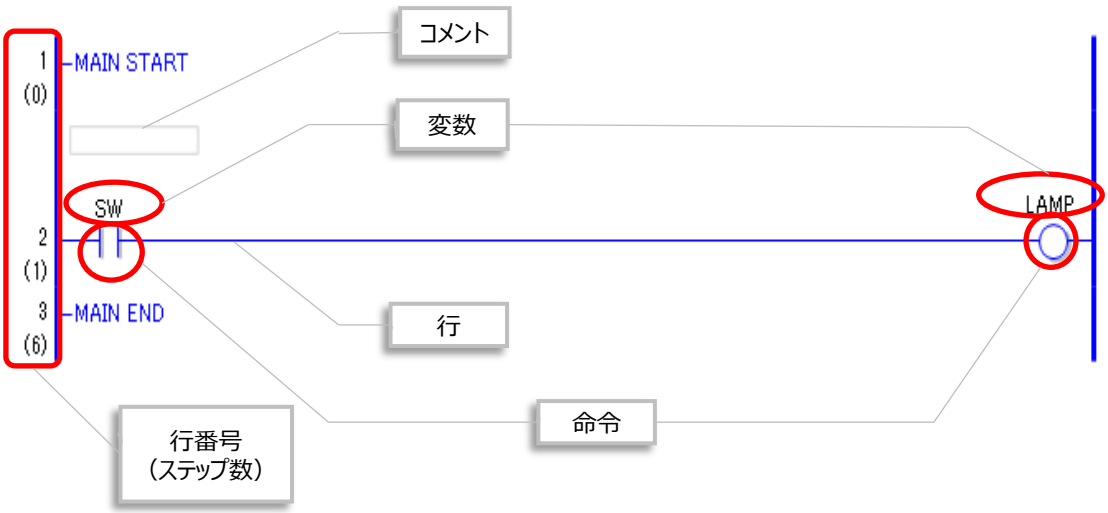
1.ロジック画面作成の基本操作

（1）代表的な基本操作

ロジック画面は、以下の順番に沿って行います。

① 行の挿入　　母線に水平な線（行）を挿入します。
② 命令の割付　　挿入した行に、命令/接点を割り付けます。
③ 変数の登録/割付　変数を作成し、命令に割り付けます。

ロジックプログラムでは、上から下、左から右に向かって、信号が流れます。信号が命令に到達すると、命令内容に従い、処理を行います（変数を変化させます）
また、接点では、割り当てた変数を変化させることで、接点をON/OFFします。



行番号：ロジックごとに、開始ラベルを1として連続した番号が表示されます。
ステップ数：6byteを1ステップとして計算したロジックプログラムのサイズです。

ONE POINT

よいロジックプログラムとは

よいロジックプログラムとは、プログラムの作成者以外が内容を容易に把握できる見やすいプログラムのことです。
（機能追加や改修時に現状の理解が早く行え、作業効率が上がります）

① コメントを入れる
GP-Pro EX（ロジック）では行ごとにコメントを入れることができます。回路の動作をコメントにすると、トラブル発生時や変更時に現状理解を素早く行うことができます。
※GP-Pro EX（ロジック）では自由に名前を付けられる変数を使用することで、ロジックプログラムの管理がしやすくなり、プログラム全体が読みやすくなります。

② サブルーチンを使って回路を分ける。
同様の処理を繰り返す場合はサブルーチンとしてモジュール化すると、読みやすくなり、変更時のメンテナンスも楽になります。また、必然的にメインプログラムは小さくなり、スキャンタイムもアップします。

③ プログラムの配置を標準化する。
回路ごとに、プログラムの配置を標準化します。例えば、現位置確認を最初に配置し、運転条件確認・リセット回路・手動操作回路・自動運転回路・出力回路と続け最後に表示器制御回路をまとめる。こうすればプログラムのどこに目的の回路があるのかが判りやすくなります。変数名もルール付けしておくとわかりやすいです。

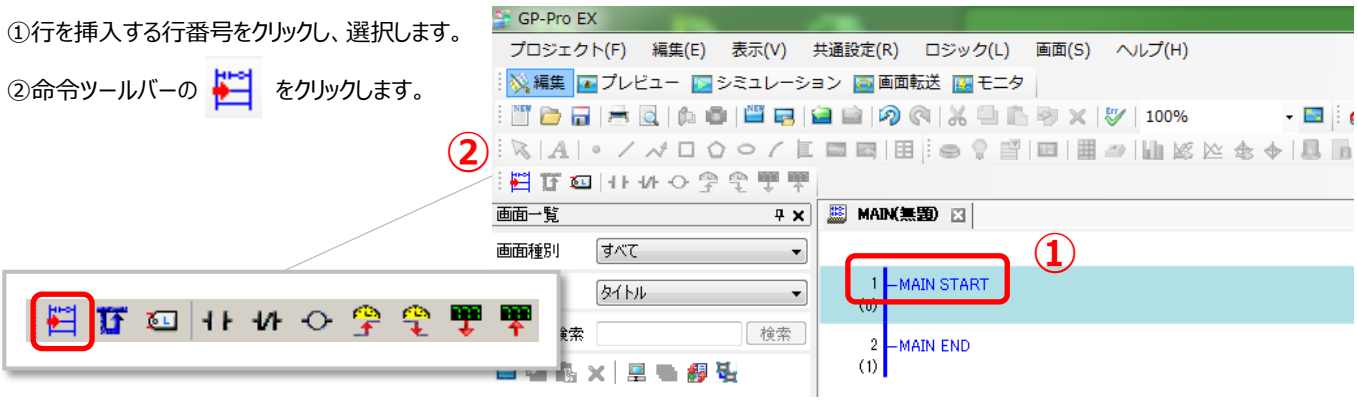
2.行の挿入

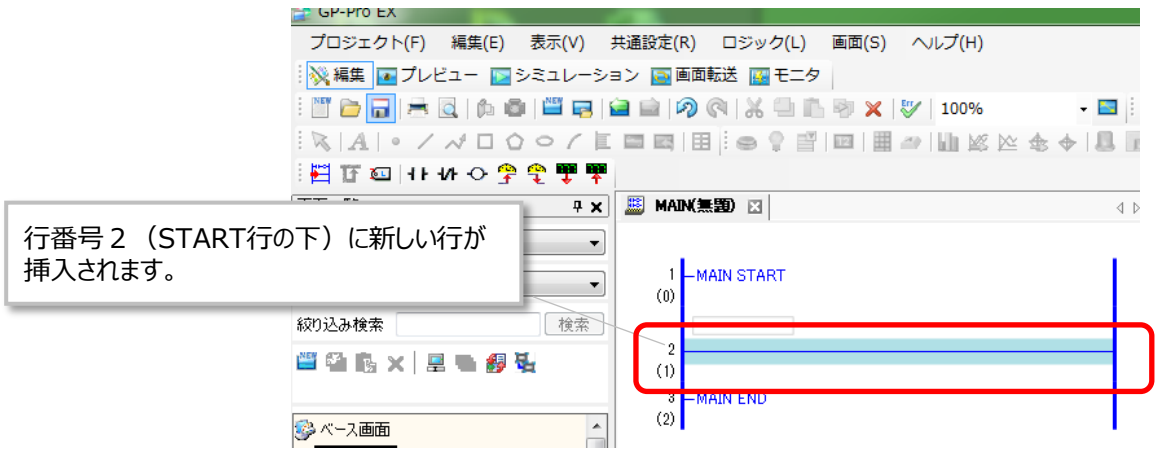
（1）行の挿入

行の挿入方法を説明します。

①行を挿入する行番号をクリックし、選択します。

②命令ツールバーの をクリックします。



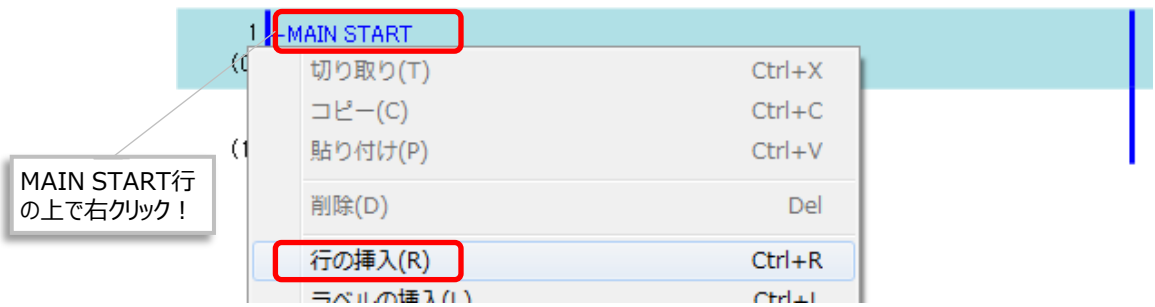


行番号２（START行の下）に新しい行が挿入されます。

ONE POINT

便利な行の挿入方法

選択した行の上で右クリックすると、ショートカットメニューが表示され、「行の挿入」を選択することができます。



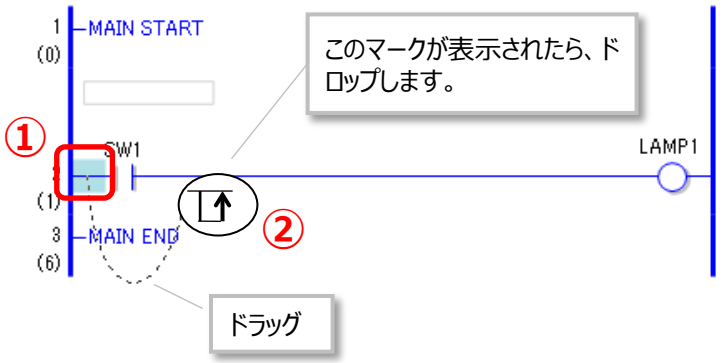
ショートカットキーから分岐を挿入する方法
行を挿入したい点をクリックし、「CTRL＋R」を押すことでも、行を挿入することができます。
※行挿入をファンクションキーに割り当てることもできます。

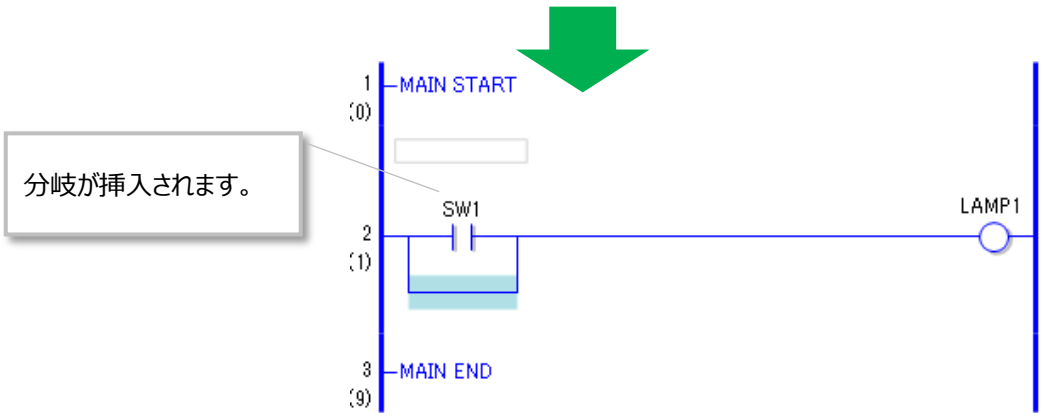
（2）分岐の挿入

行に分岐を挿入する方法について説明します。
（分岐を挿入する前に命令を割り付けておきます）

①行の上で分岐を開始したいポイントをクリックします。

②クリックしたまま、右にドラッグします。
（カーソルが⃠に変わり、そこから破線が描かれます）





ONE POINT

便利な分岐の挿入方法

分岐を開始したい点を右クリックし、ショートカットメニューの「挿入」から「分岐」を選択すると分岐を挿入することができます。

ショートカットキーから分岐を挿入する方法
分岐を開始したい点をクリックし、「CTRL＋B」を押すことでも、分岐を挿入することができます。

3.命令の割り付け

（1）命令とは

GP-Pro EXは、さまざまな命令語を用意しています。命令に信号が導通すると、個々の役割に応じて動作します。

各命令のカテゴリは、大きく分類すると、7カテゴリに分類されます。
①基本命令 ②タイマ命令 ③カウンタ命令 ④演算命令 ⑤関数命令 ⑥比較命令 ⑦変換命令

（2）代表的な命令

GP-Pro EXのロジックプログラムでは、多種多様の命令を用意しています。

【代表的な命令】 下記以外にも多種多様な命令を用意しています。

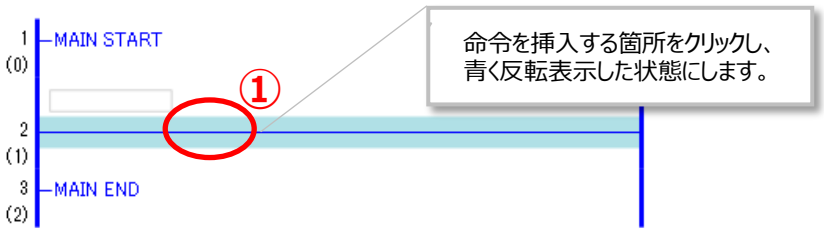
カテゴリ		命令の呼称	命令表記
基本命令	ビット基本	a接点	NO
		b接点	NC
		コイル出力	OUT
	パルス基本	立上り検出接点	PT
		立下がり検出接点	NT
	ロジック制御基本	ジャンプ	JMP
		サブルーチン処理開始	JSR
		サブルーチン処理強制終了	RET
タイマ命令	－	オンディレータイマ	TON
		オフディレータイマ	TOF
カウンタ命令	－	加算カウンタ	CTU
		減算カウンタ	CTD
演算命令	算術演算	加算演算	ADD
		減算演算	SUB
		インクリメント演算	INC
	転送	転送(コピー)	MOV
関数命令	演算関数	合計演算	SUM
		平均	AVE
比較命令	算術比較	比較(=)	EQ
		比較（<>）	NE
変換命令	数値	BCD変換	BCD
		BIN変換	BIN
		スケール変換	SCL

※各命令の詳細は、リファレンス・マニュアル（32 ロジック命令一覧）をご参照ください。

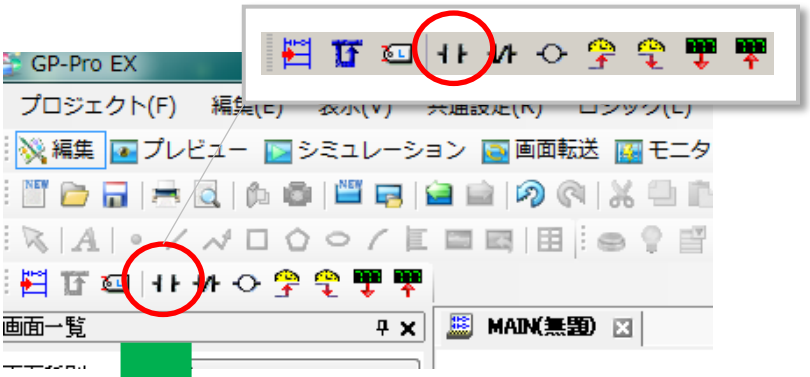
（3）命令の割り付け方法

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幾つか用意されています。
ここでは簡単な回路の作成を例に手順を説明します（ａ接点を挿入します）

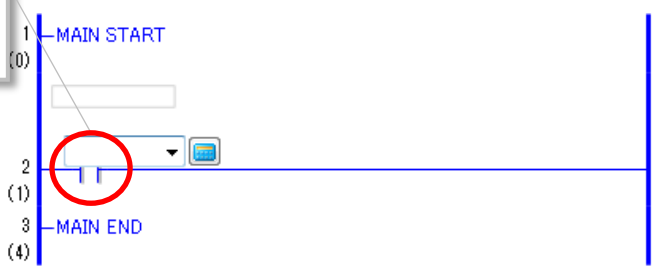
①命令を挿入する箇所（行）をクリック選択します。



②命令ツールバーの **をクリックします。**

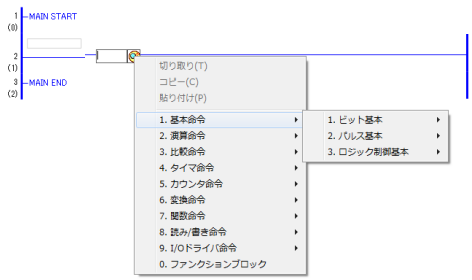


行の上に、命令が挿入されます。
（次の作業で挿入した命令に変数を割り当てます）



ONE POINT

便利な命令の割付け方法

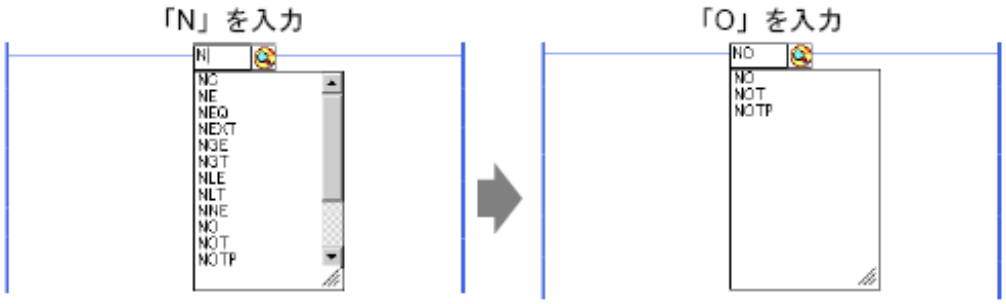


①アイコンから選択する方法

命令を挿入する行の上でダブルクリックします。

をクリックして命令を選択します。

②命令表記を利用して命令を挿入する方法
テキストボックスに直接命令を入力して挿入することができます。1文字入力するごとに、入力した文字列の命令候補が表示されます。



2.変数登録

（1）変数とは

変数とはデータの格納場所（入れ物）です。PLCプログラムのデバイスアドレスにあたります。従来のPLCでは、デバイスアドレスというPLCベンダー特有の名称（記号や番号）で扱われますが、GP-Pro EX（ロジック）ではデバイスアドレスの代わりに変数を使用します。
変数は任意の名前を付けてロジックプログラム上で使用します。変数には用途にあわせたいくつかの変数タイプ（ビット、整数、実数など）が用意されています。

それぞれの変数に自由に名前が付けられることで、ロジックプログラムの管理がしやすくなり、 プログラム全体が読みやすくなるというメリットがあります。

【 GP-Pro EXの変数と従来PLCのデバイスアドレス】

デバイス種別	外部出力	内部リレー	タイマ	データレジスタ
PLCメーカーA	X001	M100	T200	D0001
PLCメーカーB	01	1001	TIM000	DM0000
GP-Pro EX	運転開始	ランプ1	稼働時間	タンク容量

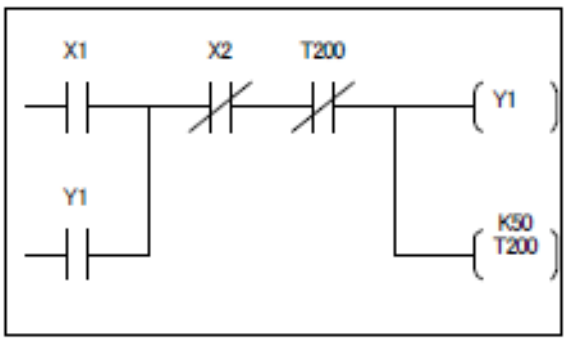
ONE POINT

どちらがわかりやすい？

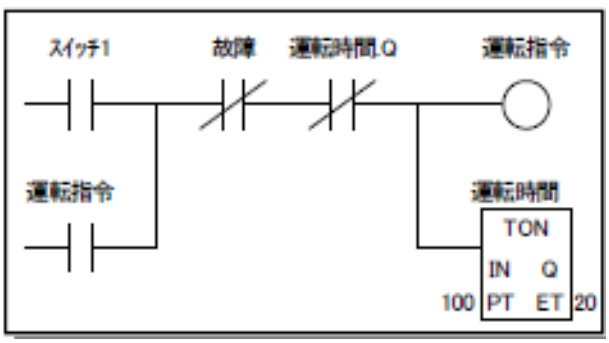
従来PLCラダープログラムソフトの場合、扱うメモリーデバイスの名前はメーカから提供されます。回路中（ソフト）のメモリーデバイスが持つ意味は回路中（ソフト）に別途コメントを挿入しておかないと意味が分かりません。

GP-Pro EX（ロジック）では、自由に名前をつけられる変数を使用することで、プログラム全体が読みやすくなります。

【従来のPLC 】



【GP-Pro EX（ロジック）】



（2）変数の種類(変数タイプ)

GP-Pro EX（ロジック）には、ユーザーが自由に作成できるユーザー変数と、あらかじめ用途が決まっているシステム変数があります。

①ユーザー変数

ユーザー変数のタイプには、ビット変数、整数変数、フロート変数、リアル変数、タイマ変数、カウンタ変数、時刻変数、日付変数、PID変数の9種類があります。

タイマ変数、カウンタ変数、時刻変数、日付変数、PID変数はそれぞれのロジック命令に対応した変数です。

ここでは、基本命令に対応しているビット変数、整数変数、フロート変数、リアル変数について説明します。

ビット変数：

ON/OFFを表す1ビットの長さの変数で、0 (OFF)か 1 (ON)の値を持ちます。

整数変数：

32ビットの長さの変数で、-2147483648～214783647の整数値を持ちます。

フロート変数：

32ビットの長さの変数で、±1.175494351e-38～±3.402823466e+38の浮動小数点と0の値を持ちます。小数点以下7桁まで使用できます。

リアル変数：

64ビットの長さの変数で、±2.2250738585072014e-308～±1.7976931348623158e+308の浮動小数点と0の値を持ちます。小数点以下15桁まで使用できます。

②システム変数

システム変数とは、あらかじめ機能を定義された変数です（変数名の最初に「＃」が付いています）
ロジック用のシステム変数と画面用のシステム変数があり、ユーザが任意の名前を付けて登録したり、変更や削除したりするなどの編集はできません。
このシステム変数は表示器画面やロジックプログラムを実行するコントローラーの状態を表現または操作します。

よく使うシステム変数

システム変数には、ロジックシステム変数(＃Lシステム変数)とHMIシステム変数(＃Hシステム変数)があり、表示器の状態を示し、動作に影響します。

変数名称	内容
#H_CurrentHour	日データ(現在値)
#H_CurrentMinute	分データ(現在値)
#H_CurrentSecond	秒データ(現在値)
#H_CurrentScreenNo	表示中画面番号
#H_ChangeScreenNo	切り替え画面番号
#L_ScanTime	ステップ開始から次スキャン開始までの時間

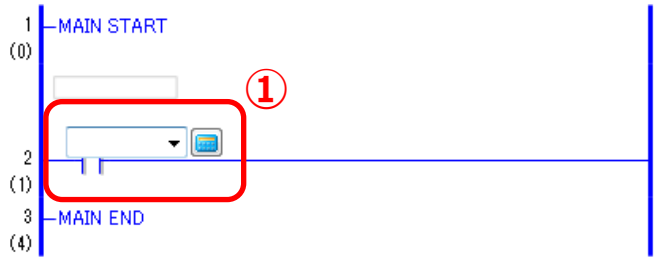
それぞれのシステム変数についての詳細は「リファレンス・マニュアル（付録）」をご参照ください。

（3）変数の登録方法

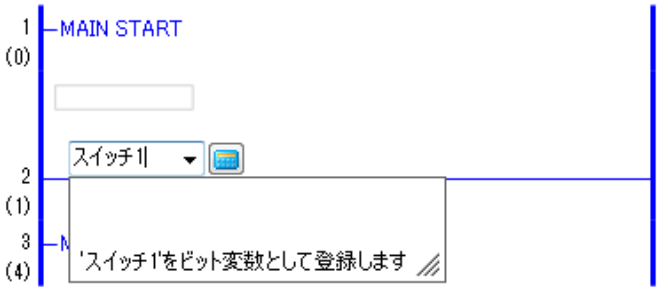
変数の登録は、ロジックプログラムの作成中に追加登録する方法とあらかじめ登録しておく方法があります。

【変数をロジックプログラムの作成中に登録する方法】
ロジックプログラム内で一連の流れの中で作成できるので、作業をスムーズに行うことができます。
ここでは、a接点に「スイッチ1」という変数を作成/割り当てます。

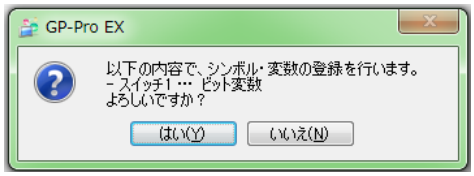
①命令を挿入すると、変数登録用のテキストボックスが表示されます。

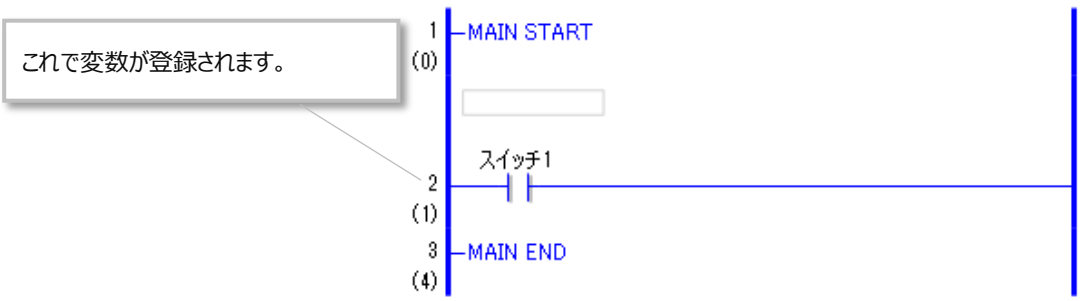


②変数名を入力します。
今回は“スイッチ１”と入力します。



③右図のようなメッセージが
表示されたら“はい”をクリックします。





変数名

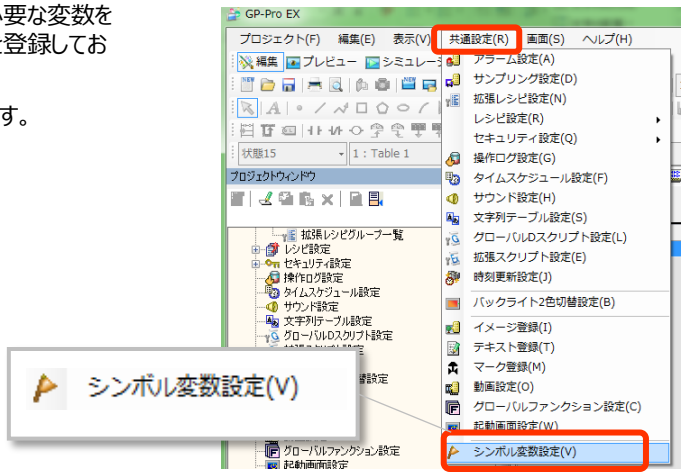
変数の名前は32文字以内です。全角文字と半角文字、大文字と小文字は区別されます。
以下の文字列は使用できませんのでご注意ください。

・空白文字列
・半角数字で始まる文字列
・以下の記号を含む文字列
+ - * / % & ^ | < > ¥ : @ . , " [ ] # ? =
・スペース(空白)や、[TAB]キー(タブ)、[DEL]キー(デリート)を含む文字列

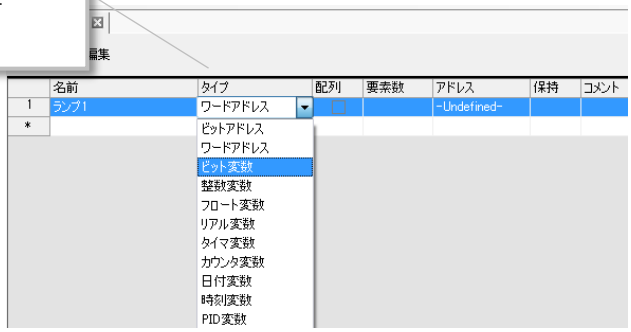
【変数を先に登録する方法】
ロジックプログラムを作成するよりも前に、シンボル編集設定（表）で必要な変数を登録しておき、必要なときにその変数を指定する方法です。先に変数を登録しておくので、無駄なく、一貫性のある変数名が作成できます。

もちろん、追加でロジックプログラムを作成する際に登録することもできます。

①「共通設定」-「シンボル変数設定」をクリックします。



シンボル変数編集画面が表示されます



②変数名とタイプを指定し、変数を登録します。

シンボル変数編集　ユーティリティ

	名前	タイプ	配列	要素数	アドレス	保持	コメント
1	コンベア後退	タイマ変数				☐	
2	コンベア前進	タイマ変数				☐	
3	後退開始	ビット変数	☐			☐	
4	後退準備中	ビット変数	☐			☐	
5	前進開始	ビット変数	☐			☐	
6	前進準備中	ビット変数	☐			☐	
7	停止	ビット変数	☐			☐	
8	ランプ1	ビット変数	☐			☐	
*							

ONE POINT

便利な変数の割付け方法

シンボル編集設定（表）に変数と登録した場合は、ロジックプログラムでは、命令を割りつけた際に表示されるカーソル▼をクリックし、候補から指定します。

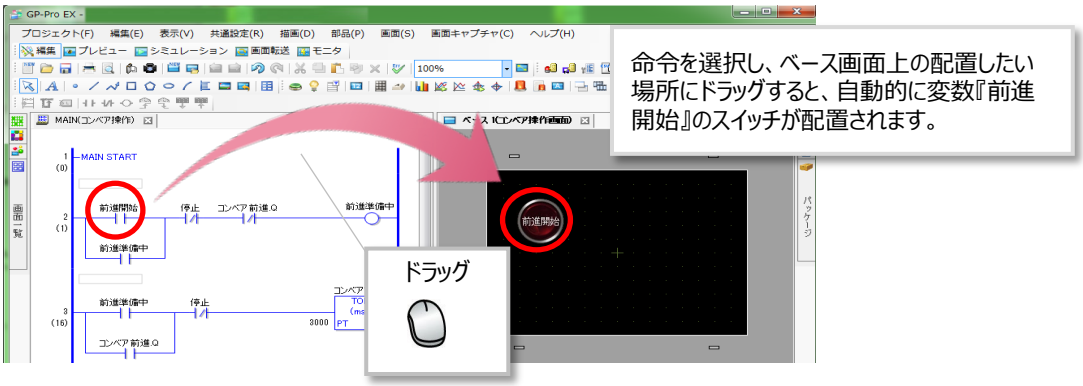
シンボル変数設定で登録した変数名のリストが表示されます。
さらに、文字を1文字入力すると、予測絞込み表示がされます。

5.作画画面との連携

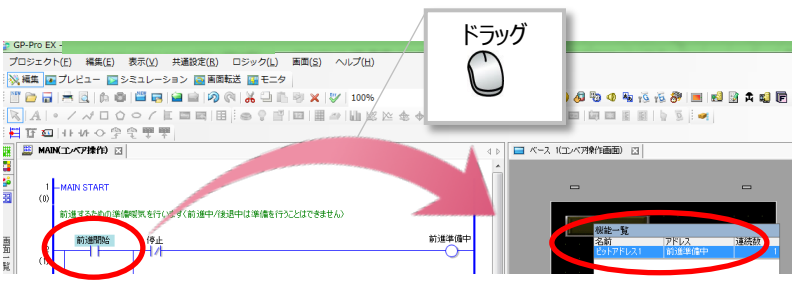
（1）ドラッグ＆ドロップで簡単なやり取り

作画とラダープログラムの連携が可能です。作画画面とラダープログラムの間でスイッチやランプなどのパーツをドラッグ＆ドロップすれば、シンボル変数の割り付け、命令やパーツの新規挿入/配置が可能に。画面やラダープログラムを効率的に作成でき、開発工数の削減に役立ちます。

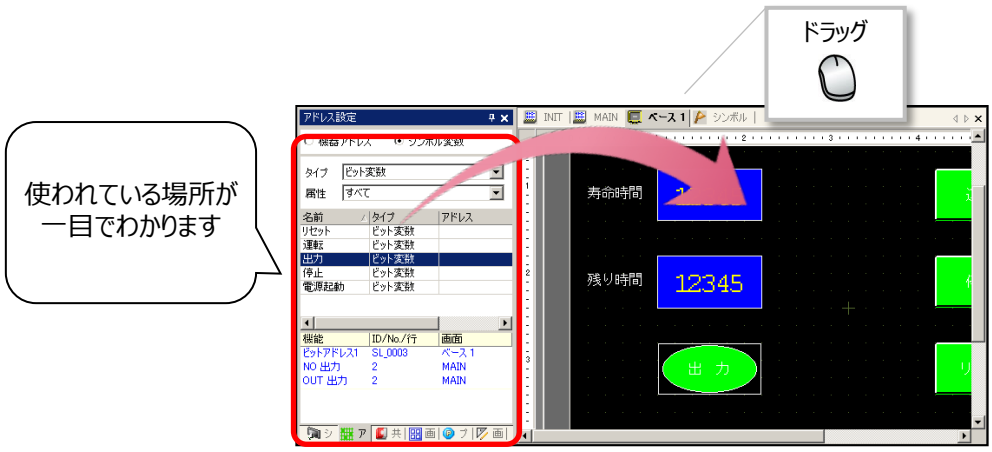
①命令からパーツ変換コピー
ロジックプログラムの命令を作画画面上にドラッグ＆ドロップすることで、該当するパーツに自動変換し、配置することができます。



②変数のコピー
ロジックプログラムの命令に既に割り付けている変数を、作画画面上の部品（スイッチやランプなど）にドラッグ＆ドロップすることで、作画画面上の部品の変数（アドレス）を変更することができます。



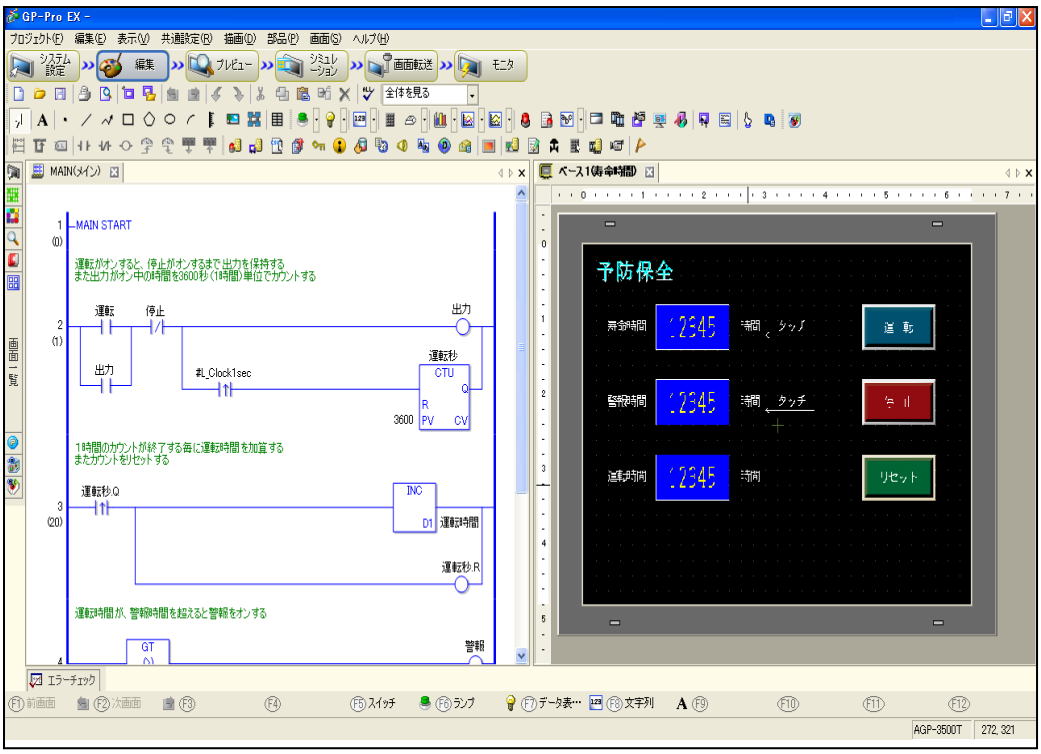
③アドレス設定からのコピー
アドレス（変数）の使用状況と使用場所がひと目で確認できます。また、ドラッグ＆ドロップで画面上の部品への割付ができます。



ONE POINT

画面ブロックの分割で作業効率UP！

画面ブロックを左右に分割し、ロジック画面とベース画面の両方を見ながら編集が行うことで作業効率を向上させることができます。



① 画面ブロックを左右に分割します。
右図のように画面タブを右クリックするとメニューが表示されるのでその中から選択します。

6.デバッグ

GP-Pro EXのロジックプログラムには下記のデバッグのための3つの機能が用意されています。

シミュレーション：GP-Pro EXで設定した内容（ロジック/ベース画面）を、STC6000に転送せずに、パソコン上で確認できます。

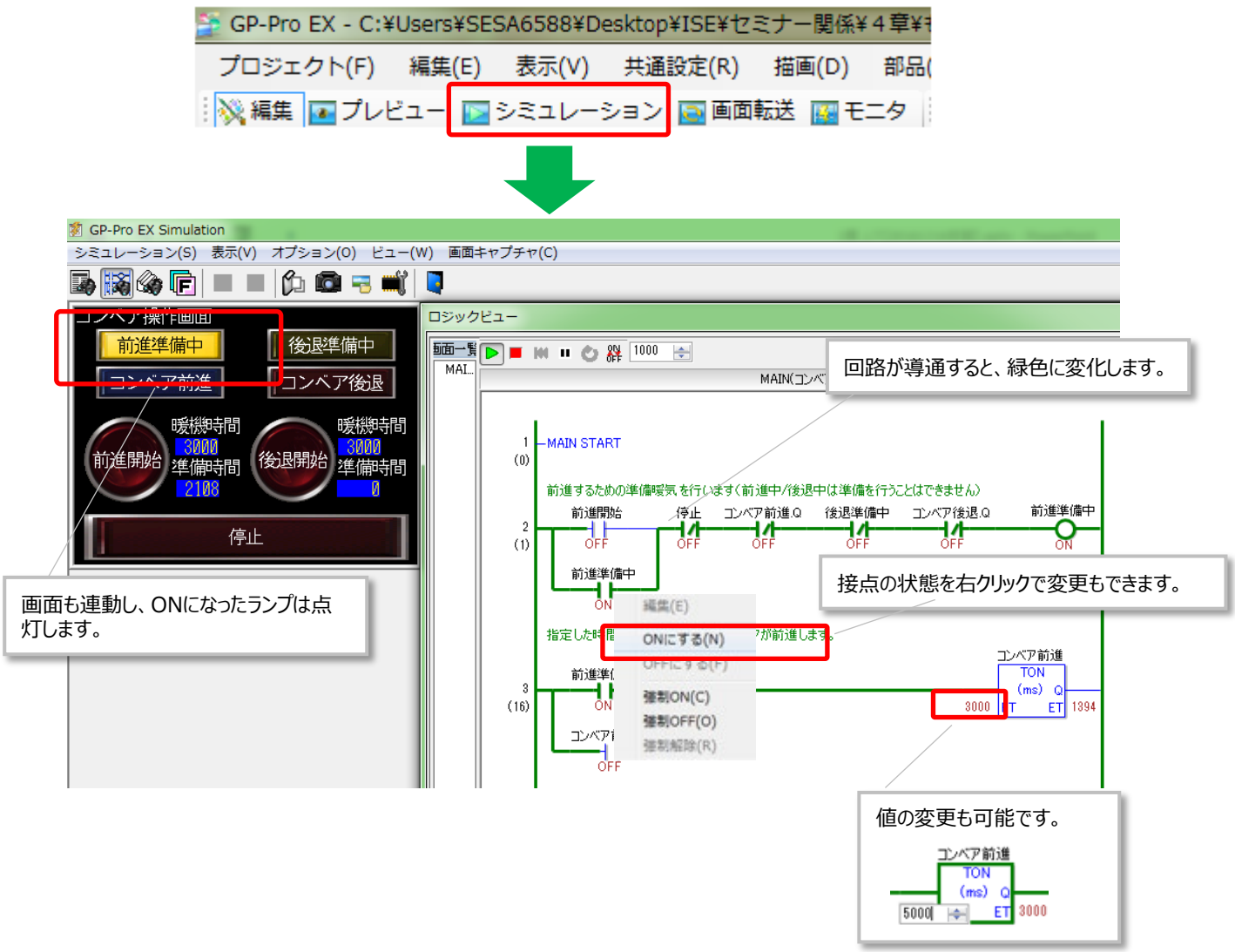
オンラインモニタ：プロジェクトをSTC6000に転送し、オンラインモニタ機能を使うと、運転中のSTC6000内のロジックプログラムをパソコン上でモニタすることができます。

ロジックモニタ：STC6000単体で、ロジックプログラムが正常に動作するかのどうかの確認や変数の現在値を一覧で確認できます。

（1）ソフトウェアでのデバッグ（シミュレーション）

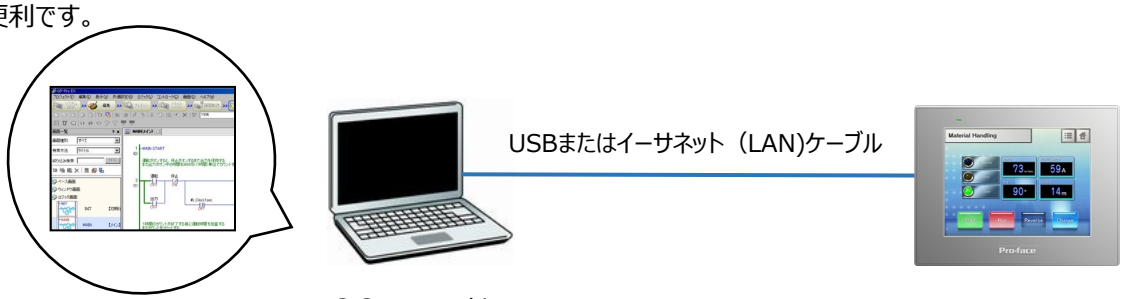
【シミュレーション】

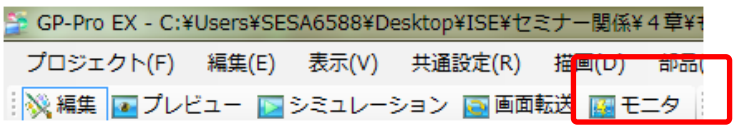
シミュレーション機能を使えば、ロジックプログラムと表示器画面の動作を確認することができます。



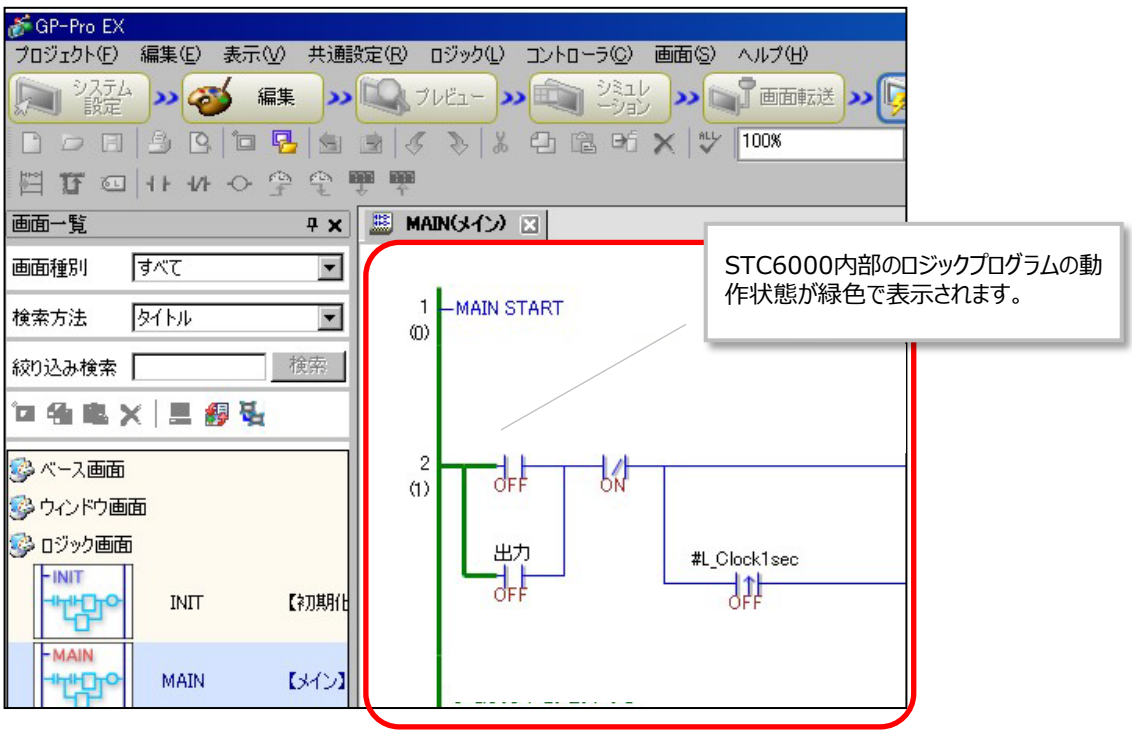
（2）ハードウェアでのデバッグ1（オンラインモニタ）

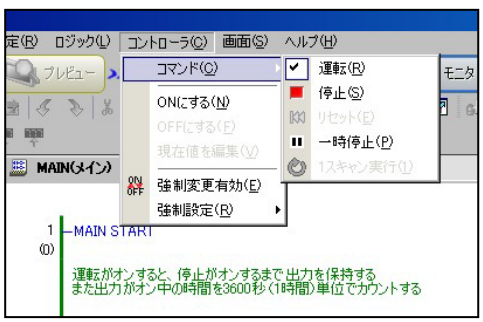
【オンラインモニタ】
運転中の表示器内のロジックプログラムをパソコン上でモニタすることができます。
シンボル変数のON/OFF状態やデバイスの数値を確認することができ、導通状態も色ですぐにわかるため、問題発生時の解析にも便利です。





状態バーの「モニタ」よりオンラインモニタを開始します。





ロジックプログラムを停止させる場合は、[コントローラ]メニューの[コマンド]－[停止]または[一時停止]をクリックします。

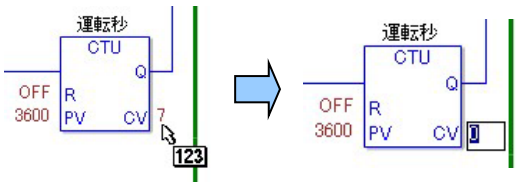
ONE POINT

オンラインモニタ中の現在値変更

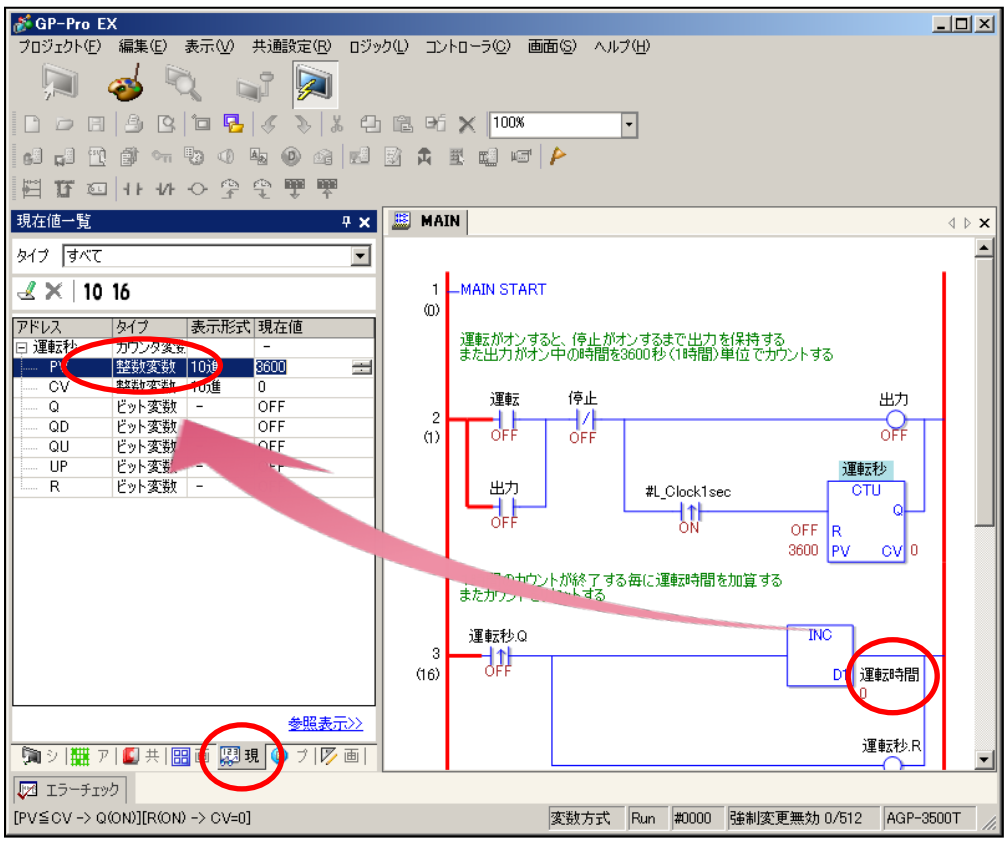
オンラインモニタ中は下記の方法で変数の現在値を変更し、動作確認やデバッグを行うことができます。

①接点のON/OFF
接点の変数名を右クリックし、メニューから
ONにする、またはOFFにするを選択します。
（強制ON、強制OFFも可能）

②データ値の変更
値を変更したいシンボル変数名をダブルクリックし
数値を入力します。



③現在値一覧ウインドウ
一覧表で複数のシンボル変数の値を変更します。
メニューバーの[表示]→[ワークスペース]→[現在値一覧ウインドウ]を選択し、ウインドウ内にシンボル変数をドラッグ＆ドロップします。

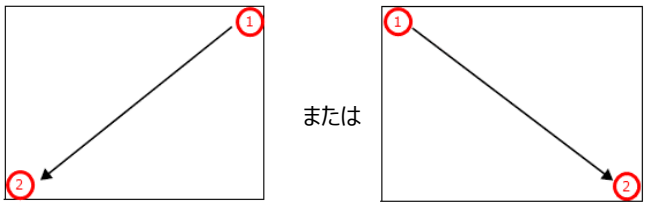


（3）ハードウェアでのデバッグ2（ロジックモニタ）

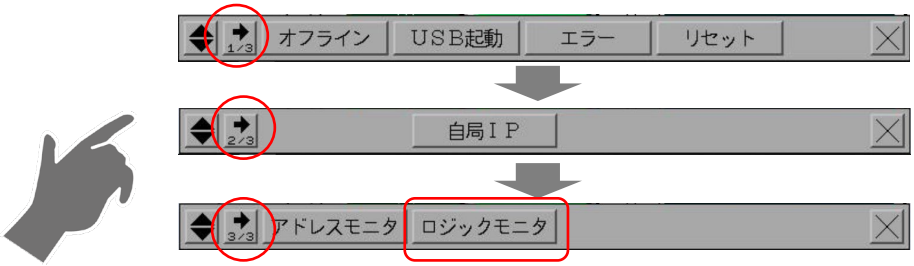
STC6000本体のみで、ロジックプログラムが正常に動作するかのどうかの確認や変数の現在値を一覧で確認できます。

【ロジックモニタの使い方】

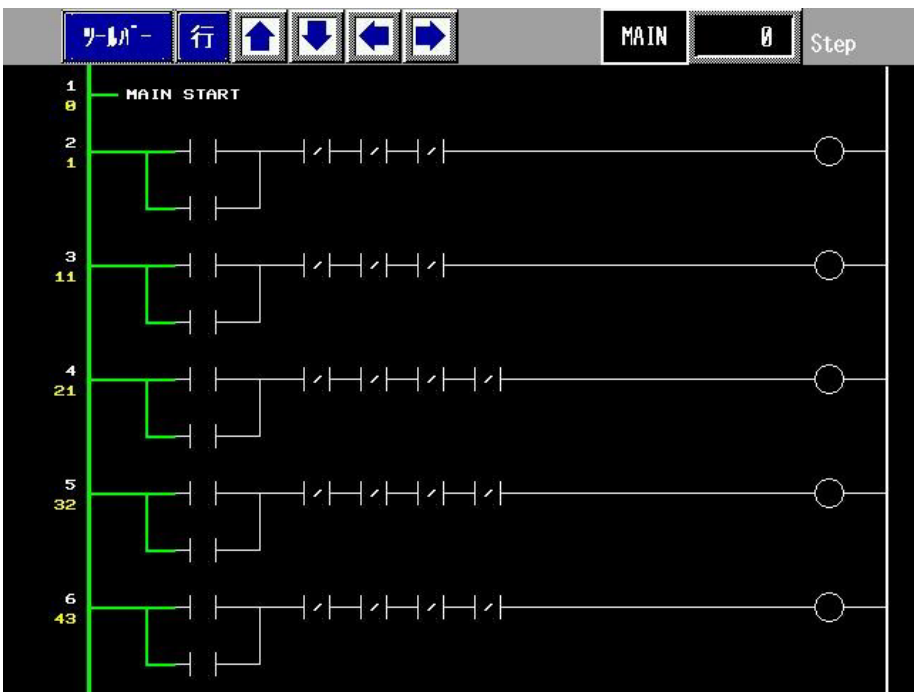
パネルの右上隅→左下隅または左上隅→右下隅(縦横40ドット以内)の順に0.5秒以内にタッチします。



画面にシステムメニューが表示されますので「ロジックモニタ」をタッチします。



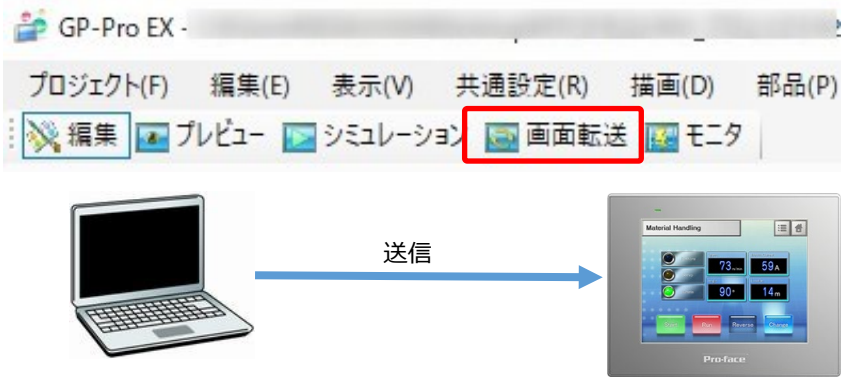
【ロジックモニタ】



同様にオフラインの【アドレスモニタ】では、転送したプロジェクトファイルで設定されているすべての変数の現在値をモニターすることも可能です。

（4）プロジェクトの転送

作成したプロジェクトをSTC6000に送信（転送）します。



○○.prxファイル

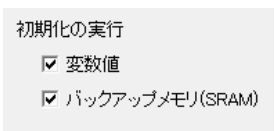
①状態バーの「画面転送」をクリックし、転送ツールを起動します。

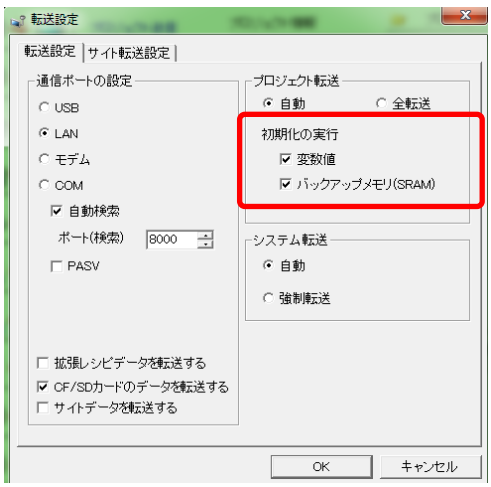
②「プロジェクト送信」をクリックします。

ONE POINT

保持転送

プロジェクトの送信時に初期化の実行で「変数値」「バックアップメモリ（SRAM）」にチェックを入れると、変数値/バックアップメモリは初期化されます。





※初期化の実行のチェックの有無に関わらず、下記の条件でプロジェクトを送信すると、「変数値」「バックアップメモリ（SRAM）」は初期化されます。

・操作ログ、アラーム、サンプリング、ファイリング、内部デバイスバックアップの設定を変更して送信したとき
・[プロジェクト転送]で[全転送]を選択して送信したとき
・[システム転送]で[強制転送]を選択して送信したとき
・異なるプロジェクトを送信したとき

「バックアップメモリ（SRAM）」：アラーム履歴やサンプリングなどのログデータを記憶します。
「変数値」：変数の情報を記憶します。

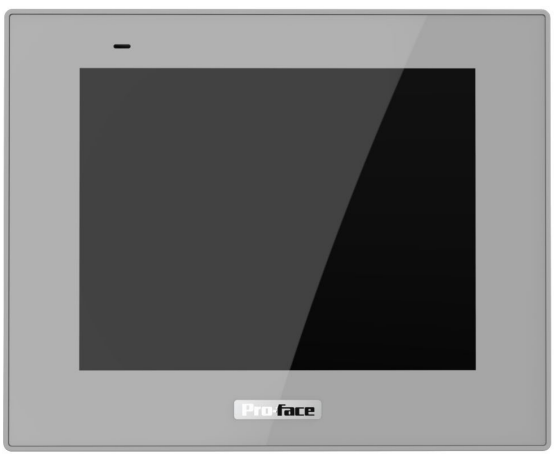
7. I/O設定

（1） I/Oとは

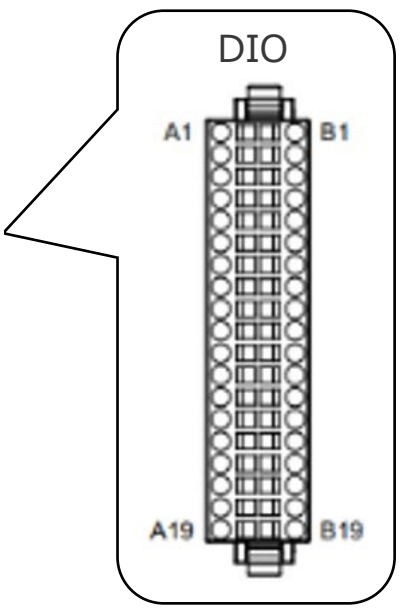
STC6000は、スイッチやランプ、ブザー等と直接接続できる入出力インターフェイスを搭載しています。その入出力インターフェイスのことをここではI/Oと呼びます。そのI/O設定について説明します。
STC6000のI/O構成は以下になります。

		STC-6300TA
シリアル		RS-232C/422/485 x1
USB		USB2.0（Type A）× 1 、USB 2.0（micro B）× 1
Ethernet		10BASE-T/ 100BASE-TX × 1
拡張IO		TM3モジュール（オプション） 背面2台まで / 拡張3台まで* / CANopen併用不可 *2023年12月予定のGP-Pro EX Ver. 4.09.550以降、最大7台まで接続可能
CANopen		CANopenマスター内蔵 *2024年8月頃対応予定
デジタル入力	入力点数	標準入力：16点 特殊入力：4点 *標準入力、標準出力のいずれか4点を使用 （特殊入出力最大4点まで）
	コモン構成	16点/1コモン
	高速カウンター入力	単相（4点）：100Kpps / 2相（1点 または 2点）：50Kpps 最小パルス幅：5μsec
	パルスキャッチ入力	最小パルス幅：5μsec
デジタル出力	出力点数	標準出力：16点 特殊出力：4点 *標準入力、標準出力の いずれか4点を使用 （特殊入出力最大4点まで）
	コモン構成	8 点/1コモンx 2
	パルス/PWM出力	最大65kHz
アナログ入出力		-

<前面>



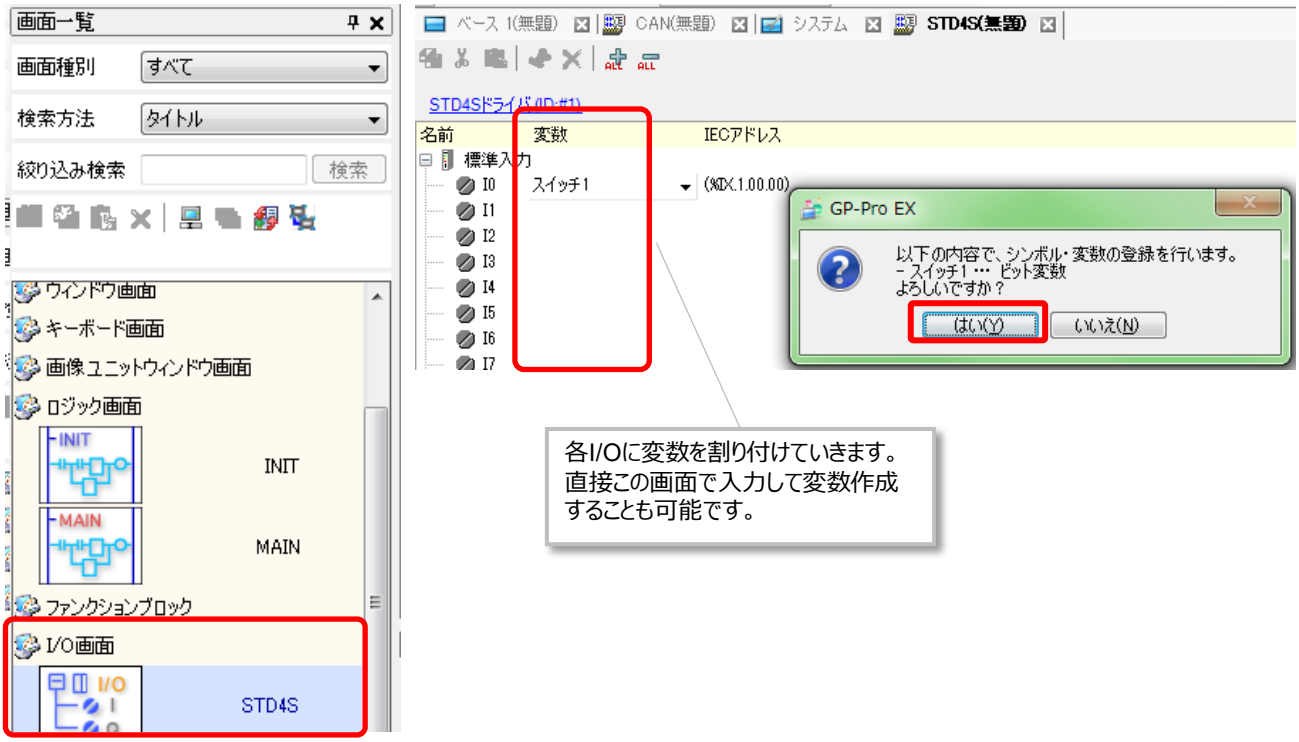
<側面>



※配線回路は、STC6000シリーズハードウェアマニュアルをご参照ください。

(2) I/Oの割り付け方法

I/O画面で、各I/Oに変数を割り付けることで、ロジックプログラムで使用することができるようになります。



メモ（以下は余白です。ご自由にお使いください）

Pro-face™
by Schneider Electric

第2章

基本回路をマスターしよう

Ⅰ　基本回路　その1　2-2

1.【理解しよう】基本回路その１

（ON回路/OFF回路/AND回路/OR回路/NOT回路）

2.【やってみよう】基本回路その１(自己保持回路)

Ⅱ　基本回路　その2　2-12

1.【理解しよう】基本回路その2

（インターロック回路/タイマ回路/カウンタ回路/SET・REST回路）

Ⅲ　基本回路　その3　2-20

1.【理解しよう】基本回路その3

（転送命令/比較演算命令）

メモ（以下は余白です。ご自由にお使いください）

Ⅰ　基本回路 その1

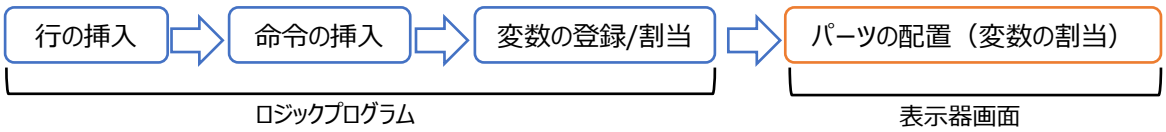
1.【理解しよう】基本回路その１

ここでは、GP-Pro EXのロジックプログラムでの、基本的な5つの回路の作成方法を解説します。本テキストを参照しながら、GP-Pro EXでロジックプログラムと表示器画面を作成し、シミュレーション機能で動作確認してください。

（1）ON回路

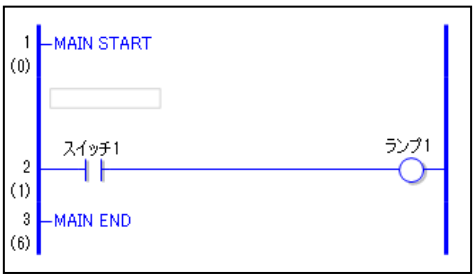
ON回路とは入力がONのときに出力がON、入力がOFFのときに出力がOFFになる回路です。

①作成手順

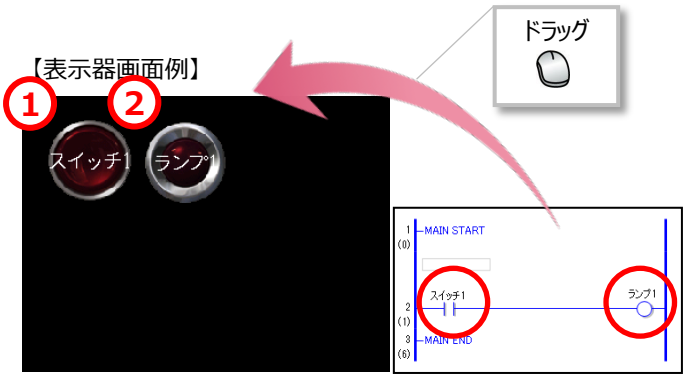


②作成例

【 ロジックプログラム例】



【表示器画面例】



【使用する命令】

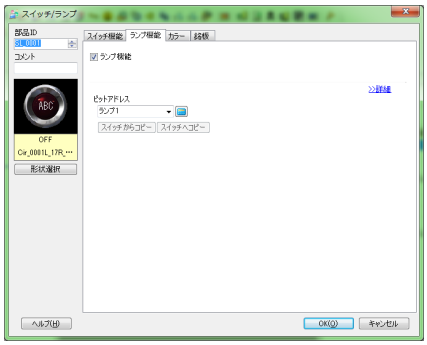
a 接点（NO）：スイッチを押さない場合、接点は開き、電気の流れは遮断されています。スイッチを押すことにより、接点が閉じ、電気が流れます。

コイル出力（OUT）：ON/OFF結果を出力する場合、OUT命令を用います。外部出力や内部コイルをON/OFFしたい場合に用います。

【スイッチ/ランプ設定例】

①スイッチ
・ビットアドレス（変数）：スイッチ1
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

②ランプ
・ビットアドレス（変数）：ランプ1
※上記以外の設定は任意です。



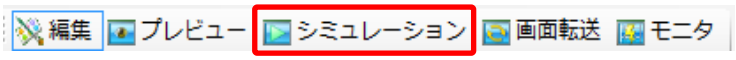
【使用する変数例】

変数名	変数タイプ
スイッチ1	ビット変数
ランプ1	ビット変数

③動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします。

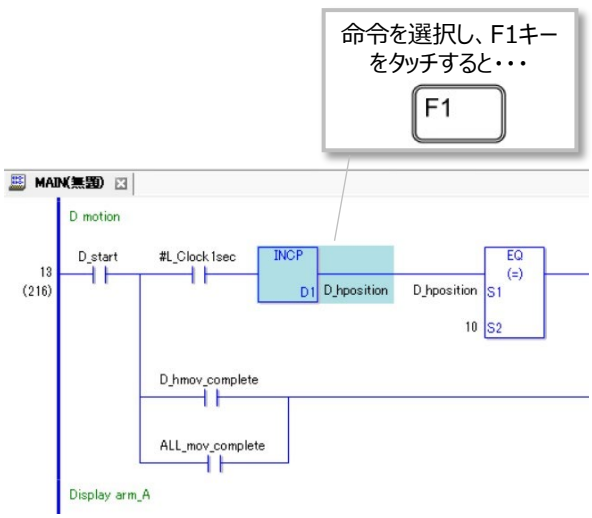
【状態バー】



ONE POINT

命令の詳しい使い方は「F1キー」で一発表示！

ロジックプログラム上で命令を選択し、パソコンの「F1キー」をタッチすると、命令の詳しい使い方や仕様などが即座に表示されます。特にロジックプログラムを開発中にわざわざ、マニュアルで該当のページを探さずに済むので大変便利です。



命令の詳しい使い方や仕様などが即座に表示されます。

32.10.6 INC / INCP(インクリメント演算)

記号/機能

ラダー命令名(呼称)	ラダー記号	機能	Step数
INC (インクリメント演算 レベル検出)	INC D1	演算	2～4

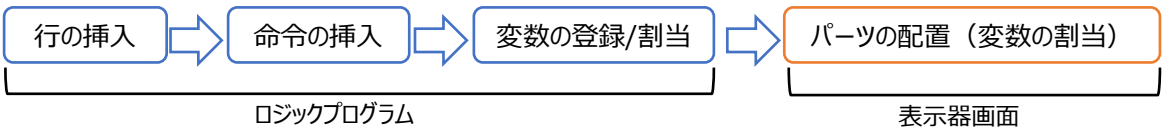
ラダー命令名(呼称)	ラダー記号	機能	Step数
INCP (インクリメント演算 立ち上がり検出)	INCP D1	演算	2～4

☞32.10.6.1 オペランド設定(D1) - INC / INCP
☞32.10.6.2 INC / INCP命令の解説
☞32.10.6.3 実行結果の確認 - INC / INCP

（2）OFF回路

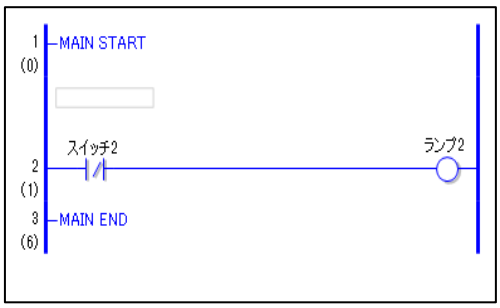
OFF回路とは入力がOFFのときに出力がON、入力がONのときに出力がOFFになるような回路です。

①作成手順

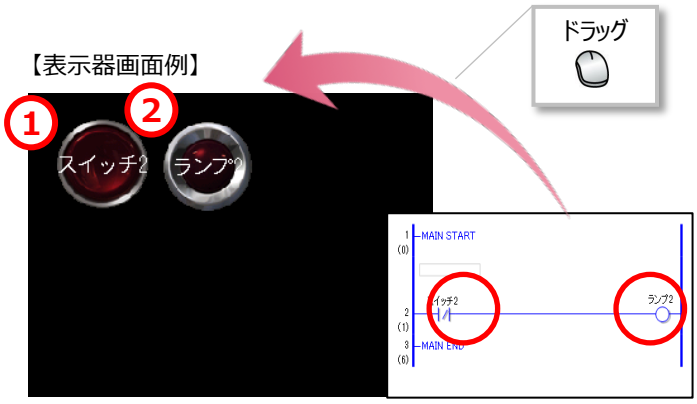


②作成例

【 ロジックプログラム例】



【表示器画面例】



【使用する命令】

b接点（NC）：スイッチを押さない場合、接点は閉じ、電気が流れています。スイッチを 押すことにより、接点が開き、電気の流れを遮断します。

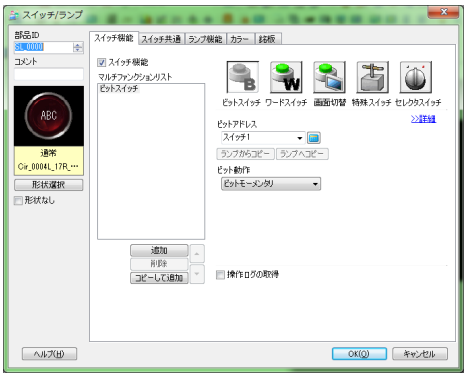
コイル出力（OUT）：ON/OFF結果を出力する場合、OUT命令を用います。外部出力や内部コイルをON/OFFしたい場合に用います。

【使用する変数例】

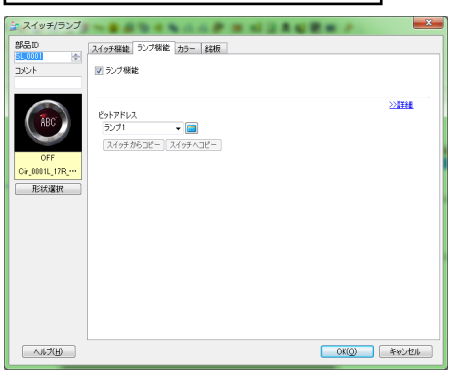
変数名	変数タイプ
スイッチ2	ビット変数
ランプ2	ビット変数

【スイッチ/ランプ設定例】

①スイッチ
・ビットアドレス（変数）：スイッチ2
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。



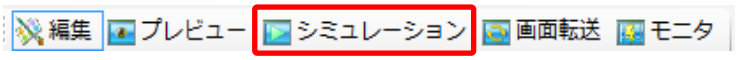
②ランプ
・ビットアドレス（変数）：ランプ2
※上記以外の設定は任意です。



③動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします。

【状態バー】

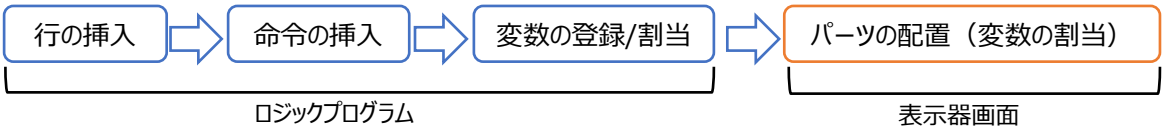


メモ（以下は余白です。ご自由にお使いください）

（3）AND回路

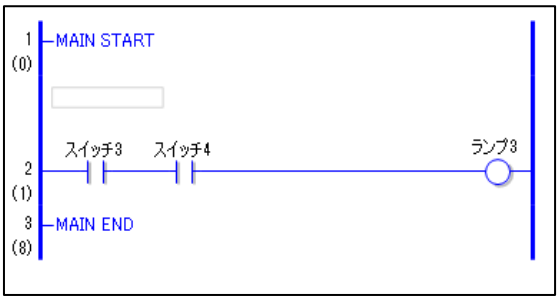
AND 回路 （論理積回路） とは、複数の入力（スイッチ）と1個の出力（コイル）があり、すべての入力がONされたときに出力されます。

①作成手順

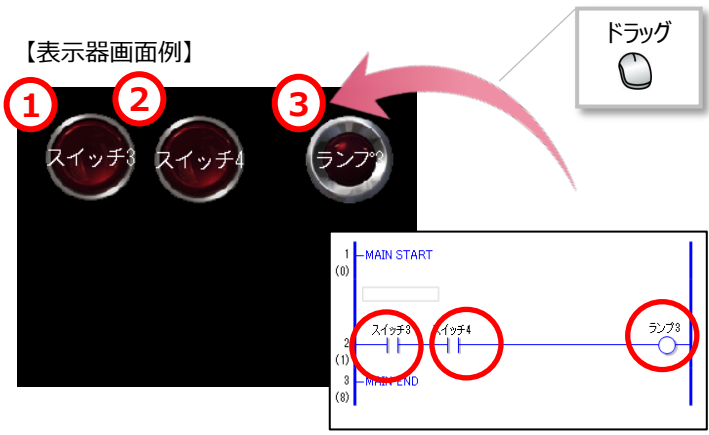


②作成例

【 ロジックプログラム例】



【表示器画面例】



【使用する命令】

ａ接点（NO） ：スイッチを押さない場合、接点は開き、電気の流れは遮断されています。スイッチを押すことにより、接点が閉じ、電気が流れます。

コイル出力（OUT） ：ON/OFF結果を出力する場合、OUT命令を用います。外部出力や内部コイルをON/OFFしたい場合に用います。

【スイッチ/ランプ設定例】

表示器画面上に配置するスイッチとランプは下記を参考に配置してください。

①スイッチ
・ビットアドレス（変数） ：スイッチ3
・ビット動作 ：反転
※上記以外の設定は任意です。

②スイッチ
・ビットアドレス（変数） ：スイッチ4
・ビット動作 ：反転
※上記以外の設定は任意です。

③ランプ
・ビットアドレス（変数） ：ランプ3
※上記以外の設定は任意です。

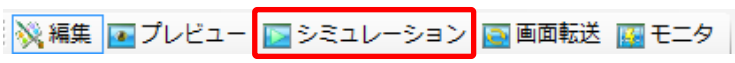
【使用する変数例】

変数名	変数タイプ
スイッチ3	ビット変数
スイッチ4	ビット変数
ランプ3	ビット変数

③動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします。

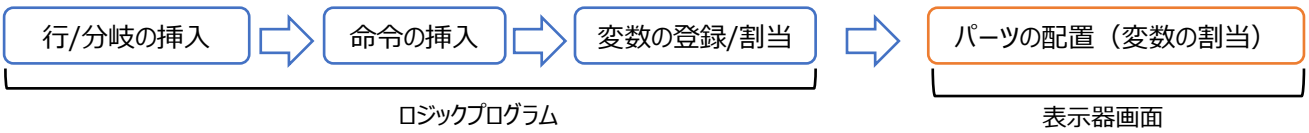
【状態バー】



（4）OR回路

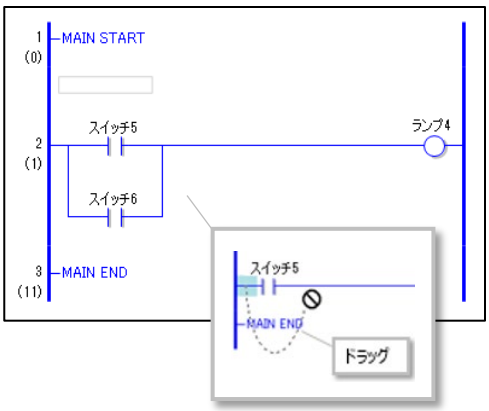
OR 回路 （論理和回路）とは 複数の入力（スイッチ）と1個の出力（コイル）があり、少なくとも1つの入力がONされたときに出力されます。

①作成手順

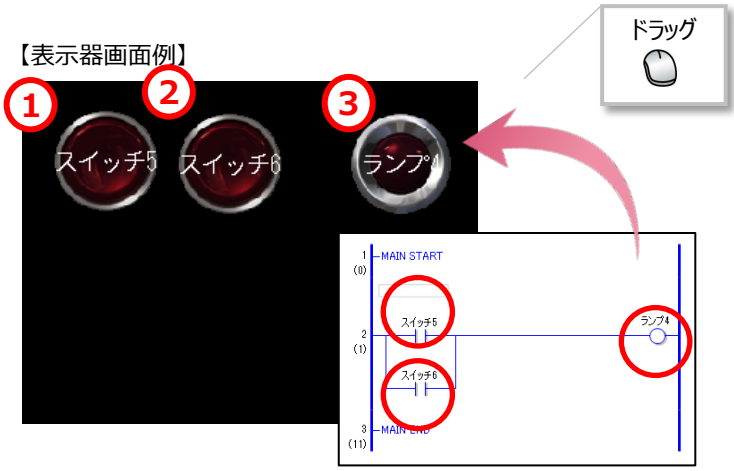


②作成例

【ロジックプログラム例】



【表示器画面例】



【使用する命令】

ａ接点（NO）：スイッチを押さない場合、接点は開き、電気の流れは遮断されています。スイッチを押すことにより、接点が閉じ、電気が流れます。

コイル出力（OUT）：ON/OFF結果を出力する場合、OUT命令を用います。外部出力や内部コイルをON/OFFしたい場合に用います。

【使用する変数例】

変数名	変数タイプ
スイッチ5	ビット変数
スイッチ6	ビット変数
ランプ4	ビット変数

【スイッチ/ランプ設定例】

表示器画面上に配置するスイッチとランプは下記を参考に配置してください。

①スイッチ
・ビットアドレス（変数）：スイッチ5
・ビット動作：反転
※上記以外の設定は任意です。

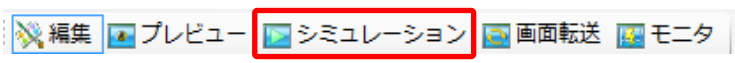
②スイッチ
・ビットアドレス（変数）：スイッチ6
・ビット動作：反転
※上記以外の設定は任意です。

③ランプ
・ビットアドレス（変数）：ランプ4
※上記以外の設定は任意です。

③動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします。

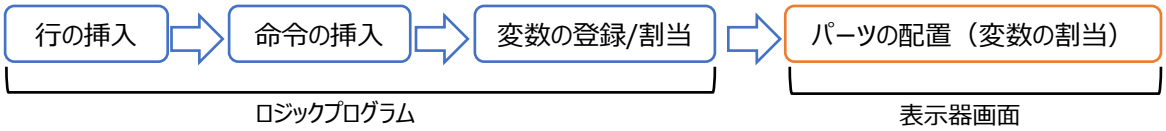
【状態バー】



（5）NOT回路

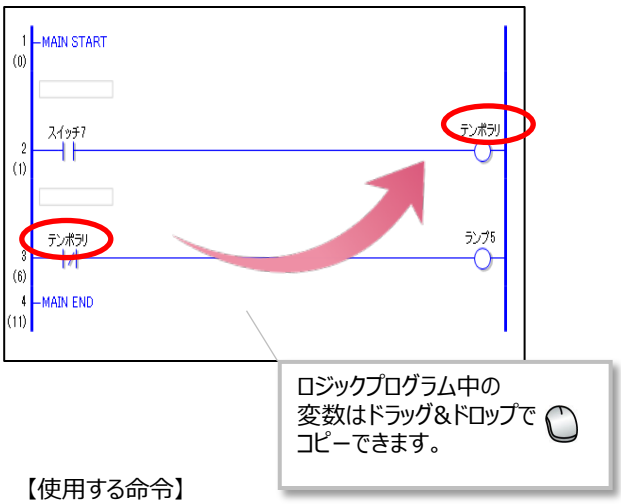
NOT 回路 （論理否定回路）とは 入力（スイッチ）と出力（コイル）があり、入力がONされたときに出力がOFFされ、入力がOFFされたときに出力がONされます。

①作成手順



②作成例

【ロジックプログラム例】



【表示器画面例】

【使用する命令】

ａ接点（NO）：スイッチを押さない場合、接点は開き、電気の流れは遮断されています。
スイッチを押すことにより、接点が閉じ、電気が流れます。

コイル出力（OUT）：ON/OFF結果を出力する場合、OUT命令を用います。外部出力や内部コイルをON/OFFしたい場合に用います。

b接点（NC）：スイッチを押さない場合、接点は閉じ、電気が流れています。スイッチを 押すことにより、接点が開き、電気の流れを遮断します。

【スイッチ/ランプ設定例】

表示器画面上に配置するスイッチとランプは下記を参考に配置してください。

①スイッチ
・ビットアドレス（変数）：スイッチ7
・ビット動作：反転
※上記以外の設定は任意です。

②ランプ
・ビットアドレス（変数）：ランプ5
※上記以外の設定は任意です。

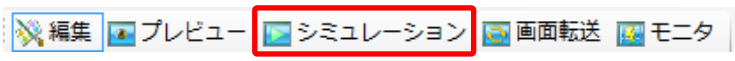
【使用する変数例】

変数名	変数タイプ
スイッチ7	ビット変数
テンポラリ	ビット変数
ランプ5	ビット変数

③動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします。

【状態バー】



2.【やってみよう】基本回路その１(自己保持回路)

（1）【実習】自己保持回路とは

自己保持回路とは、「入力がONした状態を自ら保つ回路」のことです。要求仕様を参考にして、ロジックプログラムと表示器画面を作成しましょう。また、作成後はシミュレーションで動作確認してください。

【要求仕様】

下記の２つの要求を満たすロジックプログラムと表示器画面を作成しましょう
（新規プロジェクトで作成してください/完成画面例を参考にしてください）

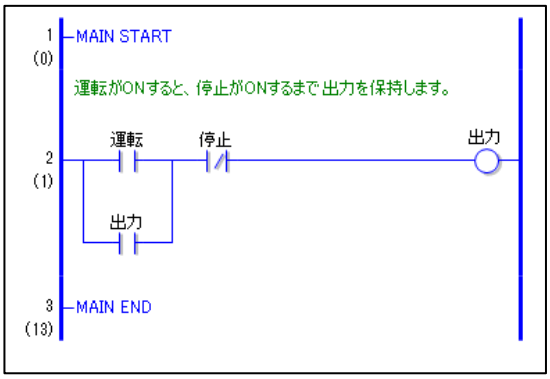
①表示器画面上の運転スイッチをタッチすると、出力ランプが点灯します。
（運転スイッチをOFFしても、出力ランプは点灯した状態を保持します）
②表示器画面上の停止スイッチをタッチすると、出力ランプが消灯します。

表示器

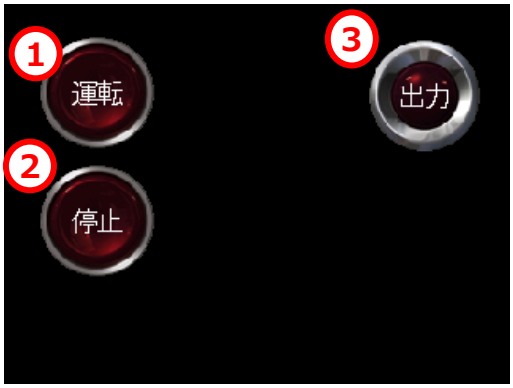
【表示器タイプ】	【接続機器設定】
シリーズ：STC6000Series STC63※※Series（表示器の大きさ） 機種： STC-6300TA VGA(640*480) 設置方法：横型	メーカー：デジタル シリーズ：メモリリンク ポート：COM1

（2）自己保持回路　完成画面例

【ロジックプログラム例】



【表示器画面例】



【使用する命令】

a接点（NO）　b接点（NC）　コイル出力（OUT）

【使用する変数例】

変数名	変数タイプ
運転	ビット変数
出力	ビット変数
停止	ビット変数

【スイッチ/ランプ設定例】

表示器画面上に配置するスイッチとランプは下記を参考に配置してください。

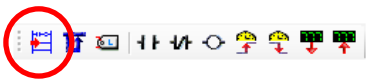
①スイッチ
・ビットアドレス（変数）：運転
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

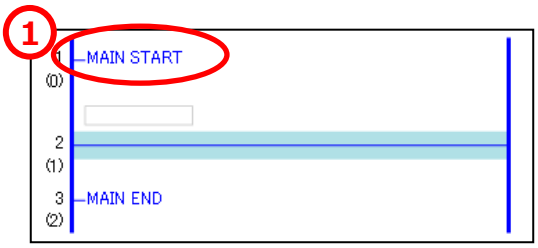
②スイッチ
・ビットアドレス（変数）：停止
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

③ランプ
・ビットアドレス（変数）：出力
※上記以外の設定は任意です。

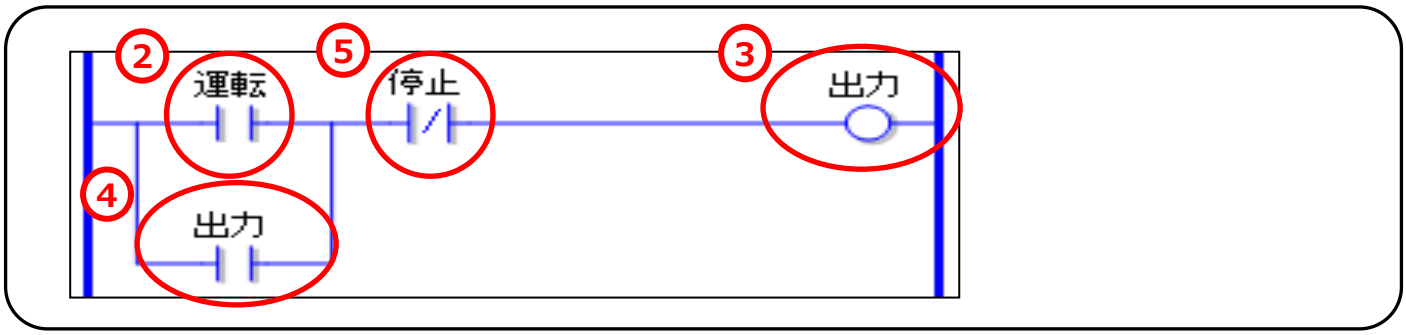
（3）【解説】 自己保持回路（ロジックプログラム）

①「MAIN START」をクリックで選択し、
「行の挿入」アイコンをクリックします。

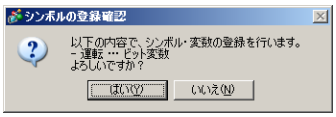




行が挿入されたら、②～⑥の設定を行います。

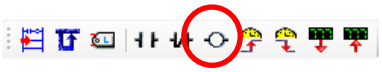


② NO(a接点）アイコンをクリックします。
変数名を『運転』と入力し、ビット変数として登録します。



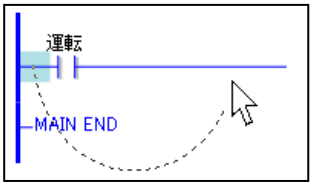
シンボルの登録確認では「はい」をクリックし、ビット変数として登録します。

③ OUT(出力コイル）のアイコンをクリックします。
『出力』と入力し、ビット変数として登録します。

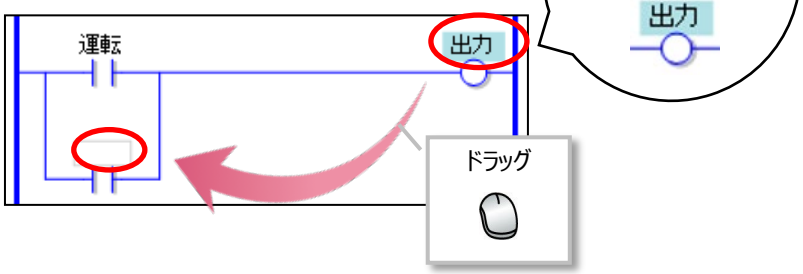


これで、『運転』接点がONしたら、『出力』コイルが出力します。

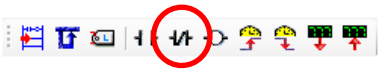
④ ②をまたぐように、a接点の左側の行から、右側の行の上までドラッグします。
（マウスカーソルが矢印に変わる場所まで）

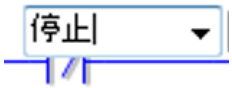


⑤ 分岐上にNO(a接点）を挿入してから、コイル『出力』の変数をドラッグでコピーします。



⑥ 次に、停止処理を挿入します。
NC（ｂ接点）を挿入します。
変数名を『停止』と入力し、ビット変数として登録します。





（4）【解説】 自己保持回路（表示器画面）

【表示器画面例】

表示器画面上の①スイッチをタッチすると、ビット変数「運転」がONし、③ランプのビット変数「出力」がONして回路が導通します。
②スイッチをタッチすると、ビット変数「停止」がONし、運転と出力間の回路が切れるため、③ランプ「出力」はOFFします。
つまり、「停止」がリセット信号の役割を果たしています。

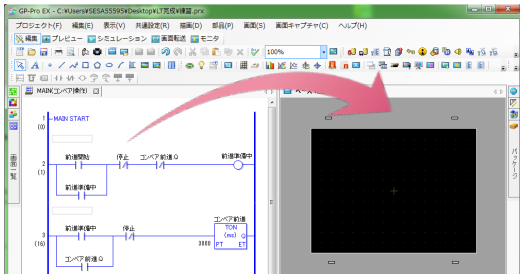
下記の設定を参考に表示器画面を作成してください。

【スイッチ/ランプの設定】

①スイッチ
・ビットアドレス（変数） ：運転
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

②スイッチ
・ビットアドレス（変数） ：停止
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

③ランプ
・ビットアドレス（変数） ：出力
※上記以外の設定は任意です。



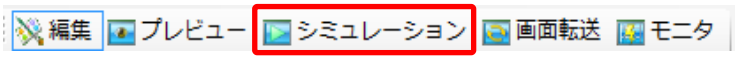
ロジックプログラムから命令をドラッグすれば、簡単にパーツを配置できます。

（5）動作確認しよう（シミュレーション）

動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします。

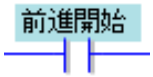
【状態バー】



ドラッグ＆ドロップでのパーツの配置

ロジックプログラムから、表示器画面に命令をパーツに変換し/配置する際は、変数だけでなく、命令全体をドラッグします。

○

× 

Ⅱ　基本回路 その2

1.【理解しよう】基本回路その2

ここでは、GP-Pro EXのロジックプログラムを使った「インターロック」「タイマ」「カウンタ」「セット/リセット」回路の作成方法を解説します。作成例を参照しながら、GP-Pro EXでロジックプログラムと表示器画面を作成し、シミュレーション機能で動作確認してください。

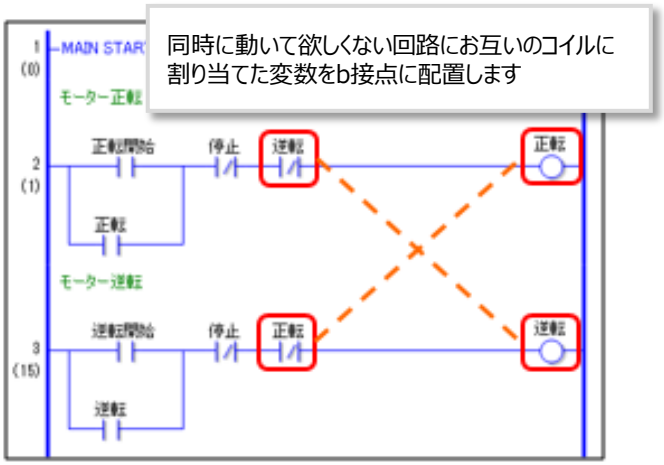
（1）インターロック回路

インターロック回路とは、自動制御において安全を確保するために非常に重要で、「やってはいけないことを、できないようにする」ために作成するものです。たとえば、エレベーターの扉が開いているときには上昇や下降ができないようにしたり、モーターが正転中に無理やり逆転できないようにしたり、機械が無理な動作するために不可欠な回路です。プログラムは、自己保持回路の応用になります。

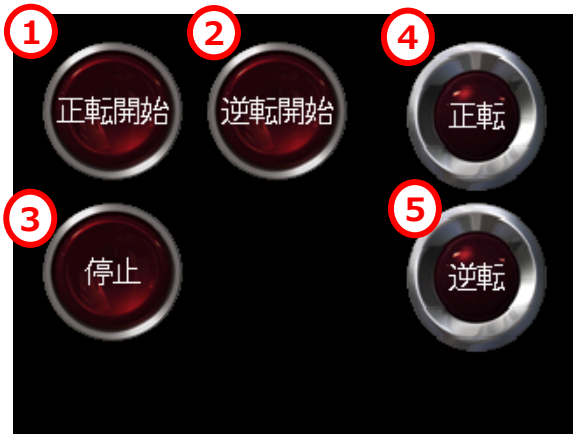
表示器画面上の①スイッチをタッチすると、ビット変数「正転開始」がONし、④ランプのビット変数「正転」がONして回路が導通します。
②スイッチをタッチしても、ビット変数「正転」がONしているので、⑤ランプのビット変数「逆転」はONしません。
逆も同様に、ビット変数「逆転」がONしているときは「正転」はONしません。相互の動作がインターロックとなっています。

①作成例

【 ロジックプログラム例】



【表示器画面例】



【使用する命令】

a接点（NO）　b接点（NC）　コイル出力（OUT）

①スイッチ
・ビットアドレス（変数）：正転開始
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

②スイッチ
・ビットアドレス（変数）：逆転開始
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

③スイッチ
・ビットアドレス（変数）：停止
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

④ランプ
・ビットアドレス（変数）：正転
※上記以外の設定は任意です

⑤ランプ
・ビットアドレス（変数）：逆転
※上記以外の設定は任意です

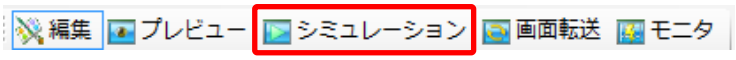
【使用する変数例】

変数名	変数タイプ
正転	ビット変数
正転開始	ビット変数
逆転	ビット変数
逆転開始	ビット変数
停止	ビット変数

②動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします）

【状態バー】

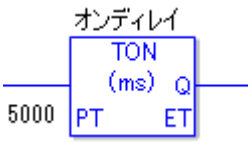


メモ（以下は余白です。ご自由にお使いください）

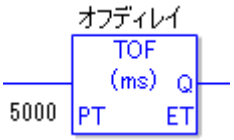
（2）タイマ回路

タイマ回路とは、設定時間後に接点が動作する回路のことです。GP-Pro EXでは、下記の3種類のタイマを用意しています。

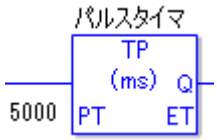
①タイマの種類　※図は5000ms(＝5秒）のタイマ動作の例です。



【オンディレイタイマ】（TON）
設定した時間、遅れて出力（タイマ.QがON）します。導通後、現在値（ET）が設定値（PT）に到達すると出力（タイマ.QがON）します。



【オフディレイタイマ】（TOF）
設定した時間、遅れて出力（タイマ.QがOFF）します。導通後、現在値（ET）が設定値（PT）に到達すると出力（タイマ.QがOFF）します。



【パルスタイマ】（TP）
設定した時間だけ、出力（タイマ.QがON）します。導通後、現在値（ET）が設定値（PT）に到達すると出力（タイマ.QはOFF）します。

②タイマ変数（構造体変数）

TON/TOF/TP命令に割りつけた変数はタイマ変数になります。タイマ変数は、それぞれの役割を持った要素変数の集まり（構造体変数）です。要素変数の役割は下記になります。

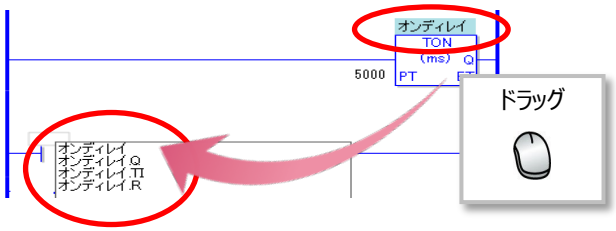
要素変数名	変数タイプ	役割
変数名.TI	ビット変数	タイマ計測時にON（タイマ変数に信号が導通するとONします）
変数名.Q	ビット変数	タイマ計測完了時にON（計測が完了（変数名.ET＝変数名.PT）するとONします）
変数名.PT	整数変数	タイマの設定値
変数名.ET	整数変数	タイマの現在値

ONE POINT

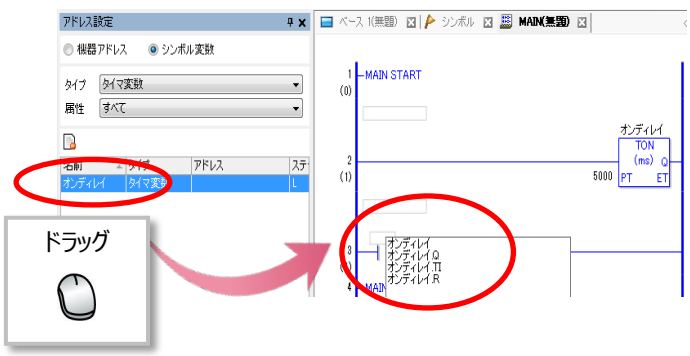
要素変数の使い方

タイマ変数やカウンタ変数の構成する要素変数は下記のようにロジックプログラム内にて割り当て使用します。

①タイマ変数を割り当てたい命令まで、ドラッグすると、変数候補リストが表示されます。リストから要素変数を指定します。



①アドレス設定から「シンボル変数」「タイマ変数」にして表示される一覧からドラッグすると、変数候補リストが表示されます。リストから要素変数を指定します。

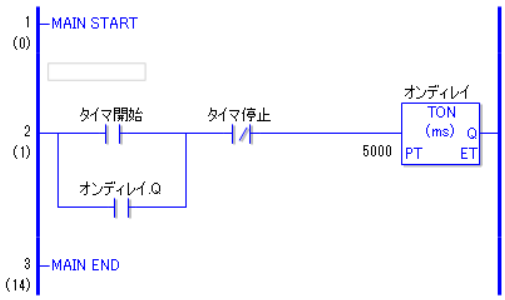


③作成例

ここでは、オンディレイタイマを使った回路について説明します。

表示器画面上の①スイッチをタッチすると、ビット変数「タイマ開始」がONし、タイマ計測が開始されます。
計測値（タイマ.ET」がタイマ.PTに到達すると、③ランプのビット変数「オンディレイ.Q」がONして回路が導通します。
（5秒間タッチしつづけると、運転ランプが点灯します）
また、画面上からタイマ設定値の変更が可能です。

【タイマ回路】



※例ではタイマ設定値を5000(ms)に設定されていますが、
変数を使って任意に変更可能です。

【使用する命令】

a接点
（NO）

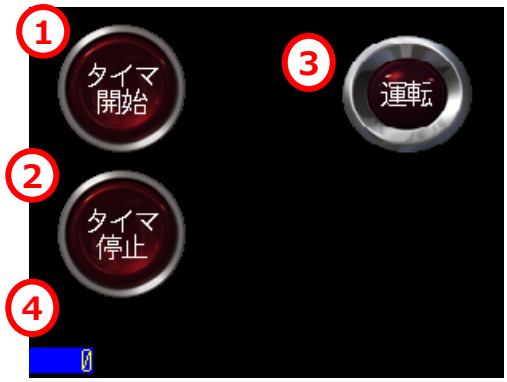
b接点
（NC）

コイル出力
（OUT）

【使用する変数例】

変数名	変数タイプ
タイマ開始	ビット変数
タイマ停止	ビット変数
オンディレイ	タイマ変数

【表示器画面例】



①スイッチ
・ビットアドレス（変数）：タイマ開始
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

②スイッチ
・ビットアドレス：タイマ停止
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

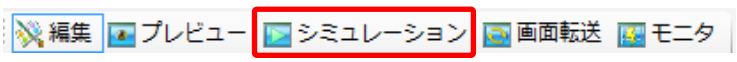
③ランプ
・ビットアドレス（変数）：オンディレイ.Q
※上記以外の設定は任意です。

④データ表示器
・ワードアドレス：オンディレイ.PT
※上記以外の設定は任意です。

④動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます（またはF12キーをタッチします）
状態バーの「シミュレーション」をクリックします。

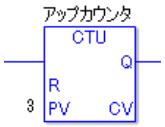
【状態バー】



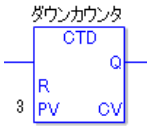
（3）カウンタ回路

カウンタ回路とは、設定回数に到達すると接点が動作する回路です。GP-Pro EXでは、下記の3種類のタイマを用意しています。

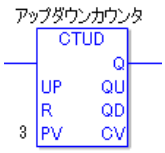
①カウンタの種類　※図は3回のカウンタ動作の例です。



【加算カウンタ】（CTU）
0から設定回数まで１ずつ加算し、設定回数に到達すると出力（カウンタ.QがON）します。
導通しているあいだ、CV（現在値）を加算し設定値（PV）に達すると出力（カウンタ.QがON）します。



【減算カウンタ】（CTD）
設定回数から減算し、0になると出力（カウンタ.QがON）します。
導通しているあいだ、CV（現在値）を減算し、０以下になると出力（カウンタ.QがON）します。



【加減算カウンタ】（CTUD）
一つの命令で、アップカウンタとダウンカウンタを切り替えて使うことができます。「.UP」ビットのON時は、加算カウンタ(CTU)と同様に、現在値を加算します。
「.UP」ビットのOFF時は、減算カウンタ(CTD)と同様に、現在値を減算します。

②カウンタ変数（構造体変数）

CTU/CTD/CTUD命令に割りつけた変数はカウンタ変数になります。カウンタ変数は、それぞれの役割を持った要素変数の集まり（構造体変数）です。要素変数の役割は下記になります。

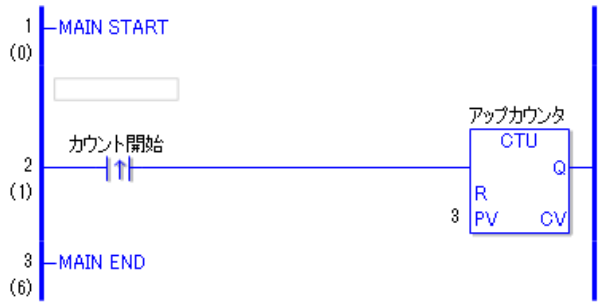
要素変数名	変数タイプ	役割
変数名.R	ビット変数	現在値リセット。0クリア。
変数名.Q	ビット変数	カウンタ計測完了時にON（現在値が設定値に達するとONします）
変数名.PV	整数変数	カウンタの設定値
変数名.CV	整数変数	カウンタの現在値

②作成例

ここでは、アップカウンタを使った回路について説明します。

表示器画面上の①スイッチをタッチすると、ビット変数「カウント開始」がONします。
設定回数（例：3）タッチすると②ランプのビット変数「アップカウンタ.Q」がONします。
また、画面上からタイマ設定値の変更が可能です。

【カウンタ回路】



※例ではカウント設定値を3に設定されていますが、変数を使って任意に変更可能です。

【表示器画面例】

【使用する命令】

立ち上がり検出接点（PT）

PT命令のビット変数がONになったときに、1スキャンのみONします。次回のスキャンではビット変数がON状態でもOFFとなります。ONの回数を数える時などに使用できます。

①スイッチ
・ビットアドレス（変数）：カウント開始
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

②ランプ
・ビットアドレス（変数）：アップカウンタ.Q
※上記以外の設定は任意です。

③データ表示器
・ワードアドレス（変数）：アップカウンタ.PV
※上記以外の設定は任意です。

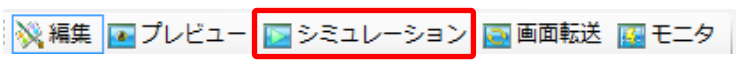
【使用する変数例】

変数名	変数タイプ
カウント開始	ビット変数
アップカウンタ	カウンタ変数

③動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます（またはF12キーをタッチします）
状態バーの「シミュレーション」をクリックします。

【状態バー】



（4） SET/RST回路

SET/RST回路とは、入力条件がONすると、指定した変数をONにし、入力条件がOFFになってもONのまま保持します。
指定デバイスをOFFにするときは、RST命令を使用します。回路の動きは、前述した自己保持回路と同様です。

※SETとRSTは必ずペアで使ってください。

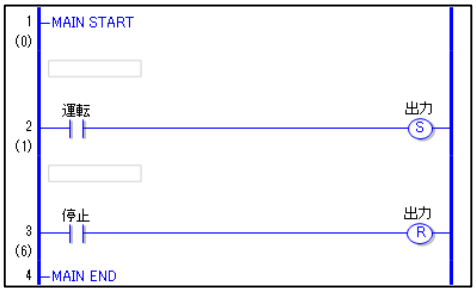
【 セット出力】（SET）
入力の状態に関わらず一度ONされると、ON状態を保ちます。

【リセット出力】（RST）
入力の状態に関わらず一度ONされると、OFF状態を保ちます。

①作成例

表示器画面上の①スイッチをタッチすると、ビット変数「運転」がONし、③ランプのビット変数「出力」がONして回路が導通します。
②スイッチをタッチすると、ビット変数「停止」がONし、運転と出力間の回路が切れるため、③ランプ「出力」はOFFします。
つまり、「停止」がリセット信号の役割を果たしています。

【 SET/RST回路】



【表示器画面例】

【使用する命令】

a接点（NO）

【変数】

変数名	変数タイプ
運転	ビット変数
出力	ビット変数
停止	ビット変数

①スイッチ
・ビットアドレス（変数）：運転
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

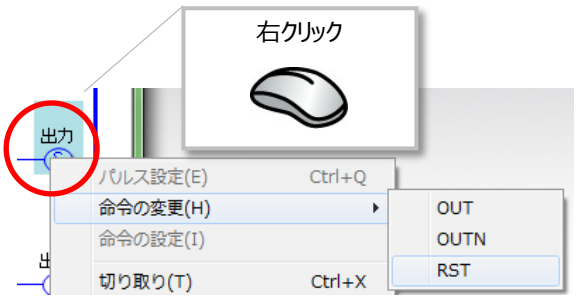
②スイッチ
・ビットアドレス（変数）：停止
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

③ランプ
・ビットアドレス（変数）：出力
※上記以外の設定は任意です。

ONE POINT

命令の変更

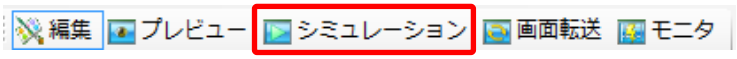
変更したい命令アイコンを選択して右クリックすると、
同系列の命令がリストに出てきます。
SET/RST命令を作成する際は、使用する変数も
同じなので、先にSETを作成してからコピーし、右クリックで
SETをRSTに変更する方法が時短になります。



②動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます（またはF12キーをタッチします）
状態バーの「シミュレーション」をクリックします。

【状態バー】



メモ（以下は余白です。ご自由にお使いください）

Ⅲ　基本回路 その3

1.【理解しよう】基本回路その3

ここでは、GP-Pro EXのロジックプログラムの「転送命令」「比較演算命令」を使った回路の作成方法を説明します。
作成例を参照しながら、GP-Pro EXでロジックプログラムと表示器画面を作成し、シミュレーション機能で動作確認してください。

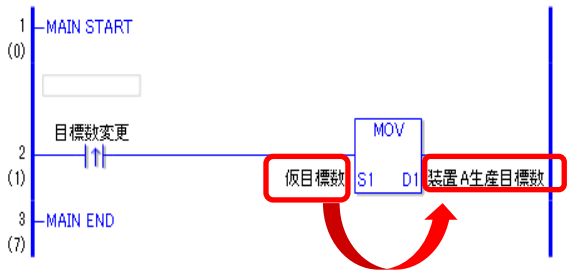
（1）転送命令（MOV(P))を使用した基本回路

転送命令は、同じ変数タイプ（またはPLCアドレス）に格納されたデータをコピーする命令です。
また、定数を変数に格納することもできます。

①作成例

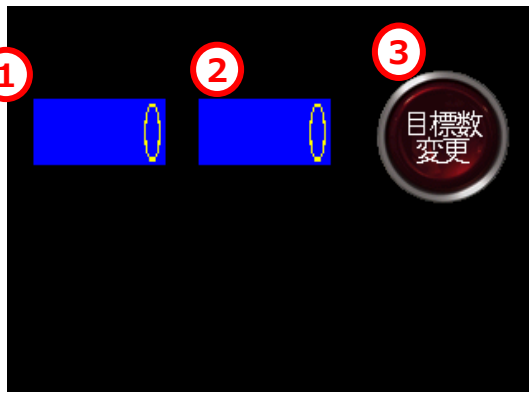
表示器画面上の①データ表示器に任意の数値を入力すると整数変数『仮目標数』の値が更新されます。
②スイッチをタッチすると、整数変数『仮目標数』の値が整数変数『装置A生産目標数』にコピーされます。

【転送命令（MOV(P))を使用した基本回路】

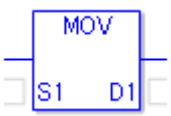


S1→D1へ、転送します。

【表示器画面例】



【使用する命令】



転送
（MOV）

立ち上がり検出接点
（PT）

①データ表示器
ワードアドレス（変数）：仮目標数
※上記以外の設定は任意です。

①データ表示器
ワードアドレス（変数）：装置A生産目標数
※上記以外の設定は任意です。

②スイッチ
・ビットアドレス（変数）：目標数変更
・ビット動作：ビットモーメンタリ

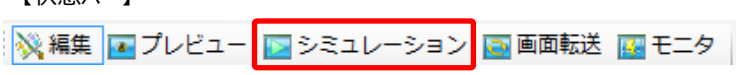
【使用する変数例】

変数名	変数タイプ
目標数変更	整数変数
仮目標数	整数変数
装置A生産目標数	ビット変数

②動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます（またはF12キーをタッチします）
状態バーの「シミュレーション」をクリックします。

【状態バー】



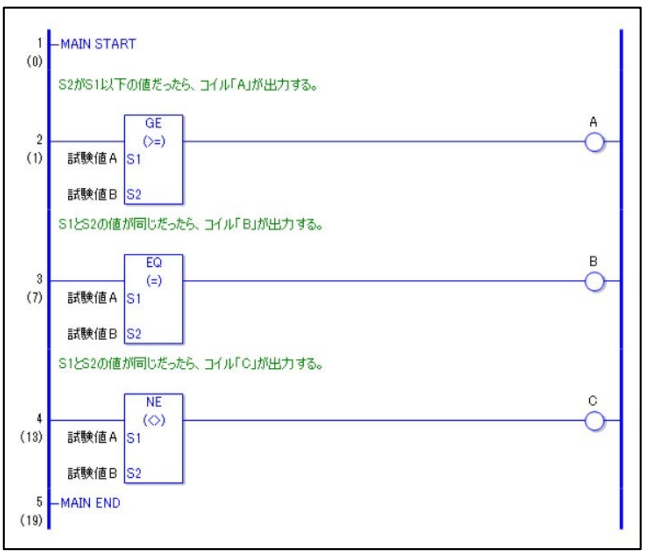
（2）比較演算命令を使用した基本回路

比較演算命令は、2つのデータの大小比較（＝、<>、>、<、>＝、<＝）を行い、
条件が成立している間、（または、条件が成立すると）導通します。
条件が成立しない場合はOFFとなり接点はOFFのままです。

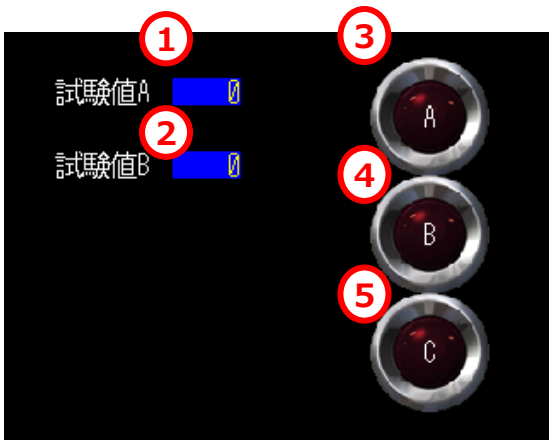
①作成例

表示器画面上の①データ表示器に任意の数値を入力すると整数変数『試験値A』の値が更新されます。
②データ表示器に任意の数値を入力すると整数変数『試験値B』の値が更新されます。
『試験値A』＞＝『試験値B』なら③ランプのビット変数『A』がONします。
『試験値A』＝『試験値B』なら④ランプのビット変数『B』がONします。
『試験値A』≠『試験値B』なら⑤ランプのビット変数『C』がONします。

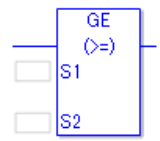
【比較演算命令を使用した基本回路】



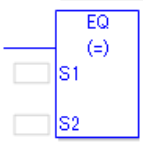
【表示器画面例】



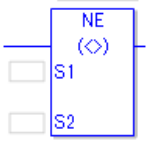
【使用する命令】



比較(＞＝)　（GE）
S1とS2が比較され、比較の結果S1＞＝S2なら導通します。



比較(＝)　（EQ）
S1とS2が比較され、比較の結果S1＝S2なら導通します。



比較()　（NE）
S1とS2が比較され、比較の結果S1≠S2なら導通します。

コイル出力（OUT）

【使用する変数例】

変数名	変数タイプ
試験値A	整数変数
試験値B	整数変数
A	ビット変数
B	ビット変数
C	ビット変数

①データ表示器
ワードアドレス（変数）：試験値A
入力許可：チェックあり
※上記以外の設定は任意です。

②データ表示器
ワードアドレス（変数）：試験値B
入力許可：チェックあり
※上記以外の設定は任意です。

③ランプ
・ビットアドレス（変数）：A
※上記以外の設定は任意です。

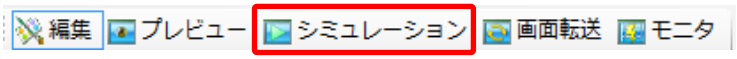
④ランプ
・ビットアドレス（変数）：B
※上記以外の設定は任意です。

⑤ランプ
・ビットアドレス（変数）：C
※上記以外の設定は任意です。

②動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます（またはF12キーをタッチします）
状態バーの「シミュレーション」をクリックします。

【状態バー】



メモ（以下は余白です。ご自由にお使いください）

メモ（以下は余白です。ご自由にお使いください）

Pro-face™
by Schneider Electric

第3章

応用回路をマスターしよう

Ⅰ　応用回路　3-2

1.【理解しよう】応用回路その１

（演算命令/加算演算/インクリメント・デクリメント・間接指定）

2.【理解しよう】応用回路その2

（プログラム制御/ジャンプ命令/サブルーチン処理）

メモ（以下は余白です。ご自由にお使いください）

I　応用回路

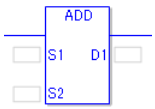
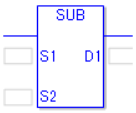
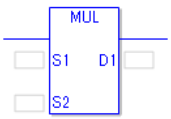
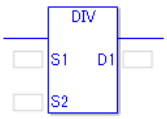
1.【理解しよう】応用回路 その1

ここでは、GP-Pro EXのロジックプログラムで、データを加工したり、効率よくプログラムするための応用回路について説明します。
各ページを参照しながら、GP-Pro EXでロジックプログラムと表示器画面を作成し、シミュレーション機能で動作確認してください。

（1）演算命令（算術演算）

演算命令を使えば、ロジックプログラム上で数値データ同士の加算/減算/乗算/除算等ができます。

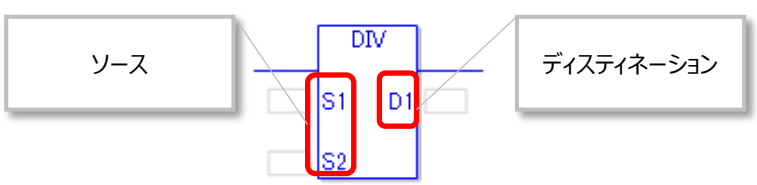
①演算命令の種類

命令名	呼称	ラダー記号	機能
ADD	加算演算		S1とS2が加算され、D1に結果が格納されます。
SUB	減算演算		S1からS2が減算され、D1に結果が格納されます。
MUL	乗算演算		S1とS2が乗算され、D1に結果が格納されます。
DIV	除算演算		S1からS2が除算され、D1に結果が格納されます。

ONE POINT

S1,S2,D1とは

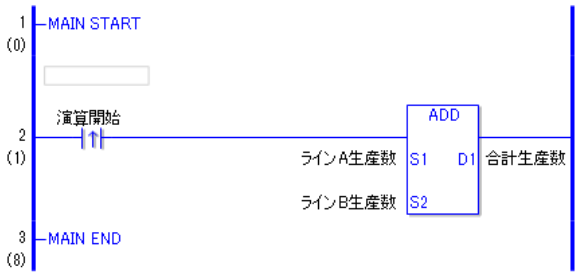
S1やS2の[S]とは、ソース（Source)の頭文字です。ソースとは、送信元（演算で使用する元データ）のことです。
D1の[D]とは、ディスティネーション(Destination)の頭文字です。演算結果を格納する宛先のことです。



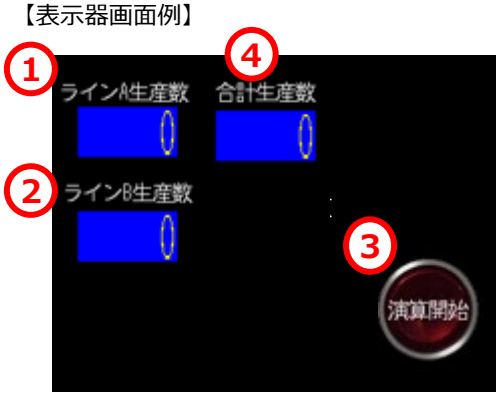
（2）加算演算

表示器画面上の①ラインA生産数と②ラインB生産数を入力し、③演算開始をタッチすると、③合計生産数に演算結果が格納されます。
下図を参考にロジックプログラムと表示器画面を作成してください（作成後はシミュレーションで動作確認を行ってください）

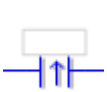
【ロジックプログラム例】（加算）



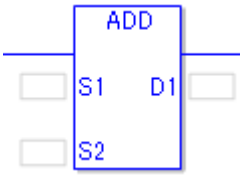
【表示器画面例】



【使用する命令】



立上り検出接点
（PT）



加算演算
（ADD）

【使用する変数例】

変数名	変数タイプ
演算開始	ビット変数
ラインA生産数	整数変数
ラインB生産数	整数変数
合計生産数	整数変数

【スイッチ/ランプ設定例】

①データ表示器
・ワードアドレス（変数）　：ラインA生産数
・データ形式：16ビット DEC
・入力許可：チェックあり
※上記以外の設定は任意です。

②データ表示器
・ワードアドレス（変数）　：ラインB生産数
・データ形式：16ビット DEC
・入力許可：チェックあり
※上記以外の設定は任意です。

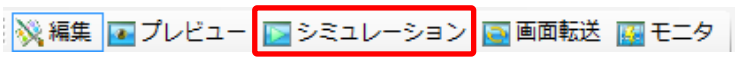
③スイッチ
・ビットアドレス（変数）　：演算開始
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

④データ表示器
・ワードアドレス（変数）　：合計生産数
・データ形式：16ビット DEC
・入力許可：チェックなし
※上記以外の設定は任意です。

動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします）

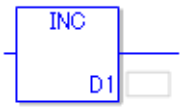
【状態バー】



（3）インクリメント/デクリメント

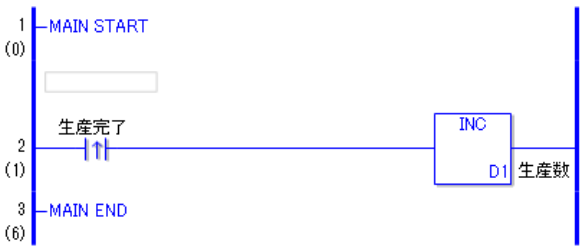
ここでは、変数の値を1ずつ増やすインクリメント命令と１ずつ減らすデクリメント命令について説明します。

【その他の演算命令】

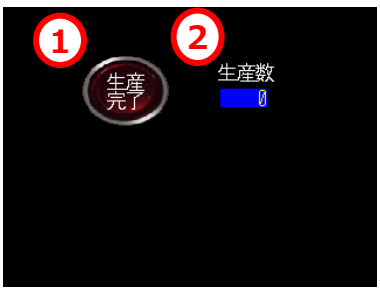
命令名	呼称	ラダー記号	
INC	インクリメント演算 レベル検出		INC命令を実行すると、D1を1加算します。
DEC	デクリメント演算 レベル検出		DEC命令を実行すると、D1を1減算します。

表示器画面上の①演算完了をタッチすると、②生産数が１ずつ増加します。
下図を参考にロジックプログラムと表示器画面を作成してください（作成後はシミュレーションで動作確認を行ってください）

【 ロジックプログラム例】（インクリメント演算）

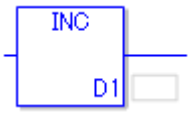


【表示器画面例】



【使用する命令】

立上り検出接点
（PT）



インクリメント演算
（INC）

【スイッチ/ランプ設定例】

①スイッチ
・ビットアドレス（変数）　：生産完了
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

【使用する変数例】

変数名	変数タイプ
生産完了	ビット変数
生産数	整数変数

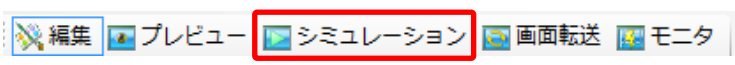
②データ表示器
・ワードアドレス：生産数
・データ形式：16ビット DEC
・入力許可：チェックなし
※上記以外の設定は任意です。

減算の場合は、INC命令をDEC命令に置き換えます。

動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします(またはF12キーをタッチします)

【状態バー】



（4）間接指定

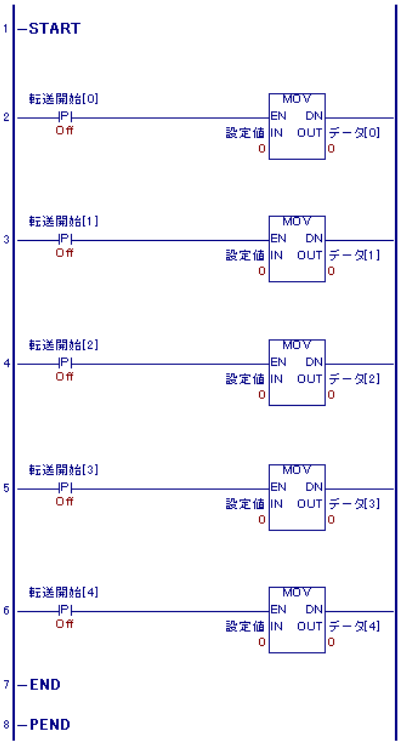
①間接指定とは

変数を配列変数として作成し、要素番号を指定することを間接指定といいます。

間接指定を利用すると、ロジックプログラムを簡略化できます。また、繰り返し処理が省略できるので、処理速度の向上や制御動作の安定が図れます。

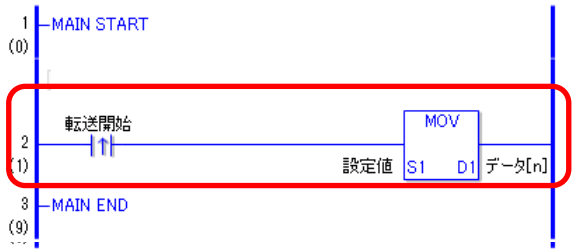
【間接指定を使わないプログラム】

複数の変数、命令を繰り返し使うことで、プログラム作成の手間がかかり、サイズも大きくなってしまいます。



【間接指定を使ったプログラム】

変数、命令ともにシンプルにでき、プログラムサイズもコンパクトにまとまります。またその分処理速度もアップします。

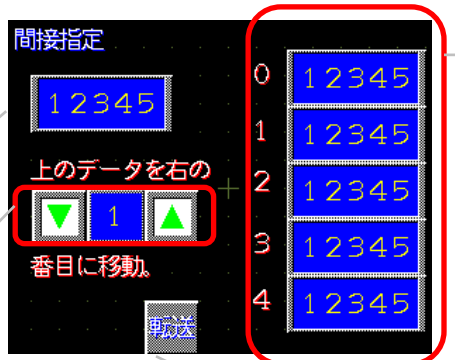


【使用する変数例】

変数名	変数タイプ	配列	要素数
転送開始	ビット変数		
設定値	整数変数		
データ	整数変数	✔	5
n	整数変数		

「ｎ」という整数変数を作成し、この「ｎ」の中身の値を変化させています。

【表示器画面例】



「設定値」に値を書き込むデータ表示器

整数変数「n」の値を表示するデータ表示器とその両脇に「n」の値を加減算するワードスイッチ。
（左：－1、右：＋1）
右の５個の数値表示器のうち、上から何番目（n番目）に「設定値」を書込むかを指定します。

それぞれの整数配列の値を表示するデータ値表示器

データ[0]
データ[1]
データ[2]
データ[3]
データ[4]

ｎの値が[]（中括弧）の中の値として入力されます。

何番目の数値表示器に書込むかを指定したら、このスイッチをタッチして「転送開始」のビットをＯＮします。

②配列変数

配列変数とは同じ変数タイプの入れ物を複数個一括登録したものです。

ひとつの変数で一度に複数の入れ物が確保されるため、同種、同サイズの変数を一括作成するのに便利です。
ひとつひとつの入れ物のことは「要素」と呼び、先頭から通し番号で何番目か決めます。

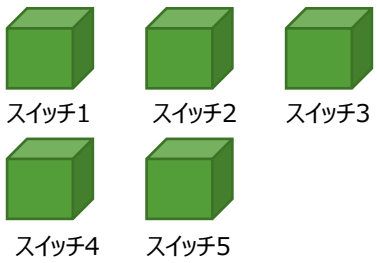
以下にその例を示します。
例）ある装置に入力用スイッチが５つあり、それぞれのスイッチの変数を作成する場合

■配列を使用しない場合

名前	タイプ	配列	要素数
スイッチ1	ビット変数	☐	
スイッチ2	ビット変数	☐	
スイッチ3	ビット変数	☐	
スイッチ4	ビット変数	☐	
スイッチ5	ビット変数	☐	

５つのビット変数をいちいち
登録するのは面倒です。
また登録ミスの恐れもあります。

ビット変数のイメージ



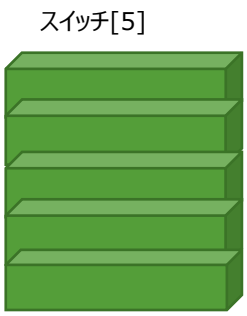
■配列を使用した場合（ビット配列）

名前	タイプ	配列	要素数
スイッチ	ビット変数	☑	5

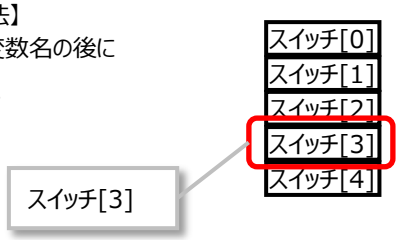
配列を使用すると１回の登録で複数個分の変数を登録できます。複数のセンサやスイッチを種類別に分類したい場合などに便利です。

上のように要素数を指定して配列として登録することで５
ビット分のビット変数が一括で登録されます。

配列変数のイメージ



【配列変数の使用方法】
各要素の指定は変数名の後に
[] に指定します。
例：変数名[※]

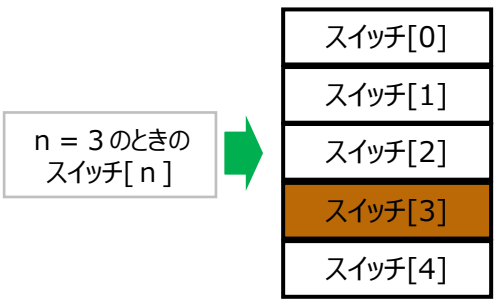


ONE POINT

配列変数の要素指定（間接）

通常、配列変数を使用する際は、スイッチ［1］というように、直接指定します。

間接指定では、直接要素番号で指定するのではなく、整数変数を使い、指定します。
例えば「ｎ」という整数変数を作成し、この「ｎ」の中身の値が変化すると、その状態に応じた「ｎ」の値が、要素番号になります。
例：　配列変数スイッチ[ｎ] は、ｎ＝3のときにスイッチ[3] を示します。



ONE POINT

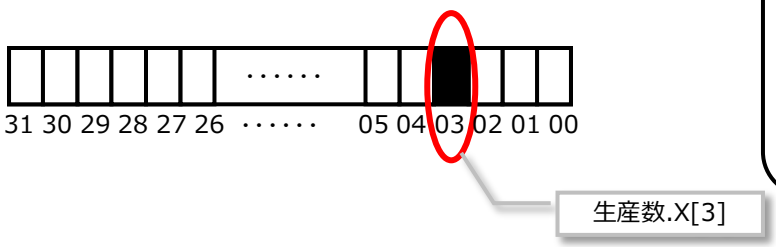
修飾語による要素指定

整数変数は修飾語を追記することで、配列と同様の以下のような扱い方もできます。
ビット、バイト、ワード単位で分割して使用できるため、メモリの節約になります。

■ 1ビット毎のデータ（修飾語＝***.X[n]）

ひとつの整数変数を１ビット毎に32個のビットデバイスとして扱います。
各ビットデバイスの取り得る値は０または１です。

例：整数変数「生産数」の４ビット目を指定する場合

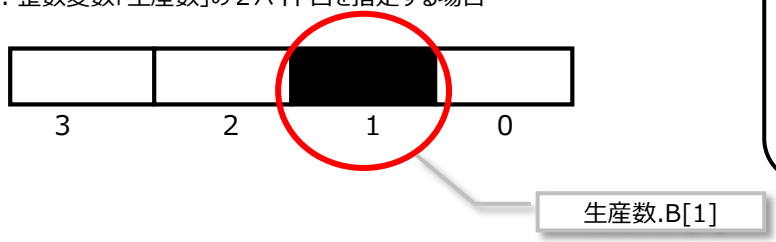


使用例

■ 8ビット毎のデータ（修飾語＝***.B[n]）

ひとつの整数変数を８ビット毎に４個のバイトデバイスとして扱います。
各バイトデバイスの取り得る値の範囲は０～２５６です。

例：整数変数「生産数」の２バイト目を指定する場合

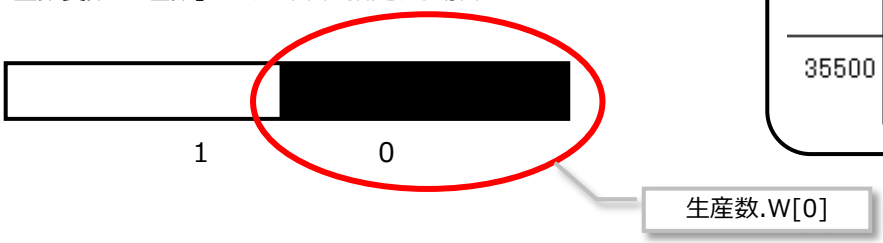


使用例

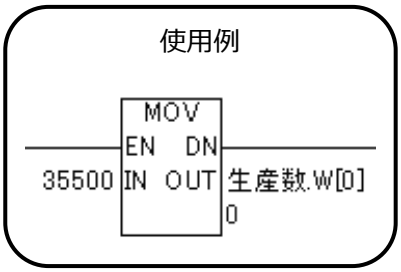
■ 16ビット毎のデータ（修飾語＝***.W[n]）

ひとつの整数変数を16ビット毎に２個のワードデバイスとして扱います。
各ワードデバイスの取り得る値の範囲は０～６５５３５です。

例：整数変数「生産数」の０ワード目を指定する場合



使用例



2.【理解しよう】応用回路 その2

（1）プログラム制御

ここまでのプログラムは、アドレスの順番に記述されている命令を単純に実行するものでした。
プログラムを効率良く記述したり、高度な処理を実現するには、命令の実行順序を制御する必要があります。

ここでは、メインロジック画面内にラベルを指定し、条件により、ラベル位置までジャンプし、実行するジャンプ（JMP）命令と、
あらかじめ作成したサブルーチン画面を条件によりメインロジック画面に呼び出し、実行するサブルーチン処理（JSR）命令について説明します。

ラダー命令名	呼称	ラダー記号	機能
JMP	ジャンプ		メインプログラム中にラベル指定した場所まで、ジャンプし、実行します。
JSR	サブルーチン処理		メインプログラムからサブルーチンプログラムを呼び出し、実行します。

スキャンタイムの変動

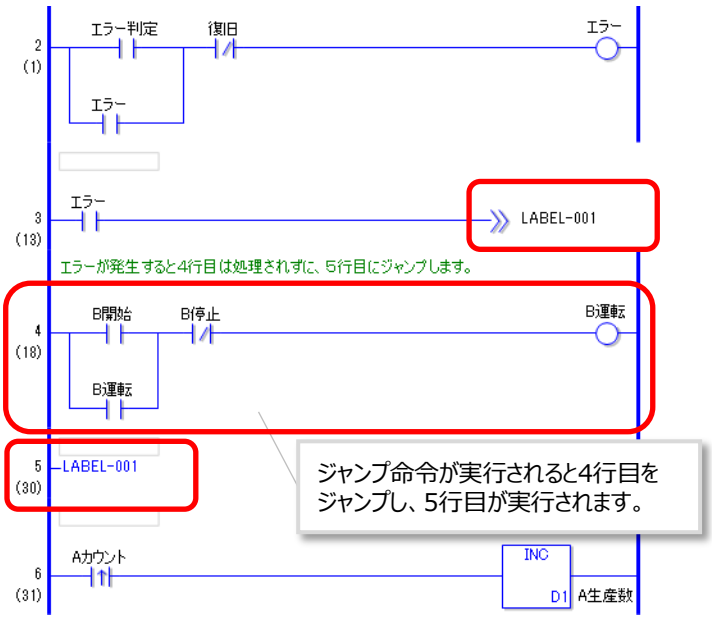
ロジックプログラム内で、ジャンプ命令やサブルーチン処理命令を使うと、一時的にスキャンタイムが変動する場合がありますのでご注意ください。

メモ（以下は余白です。ご自由にお使いください）

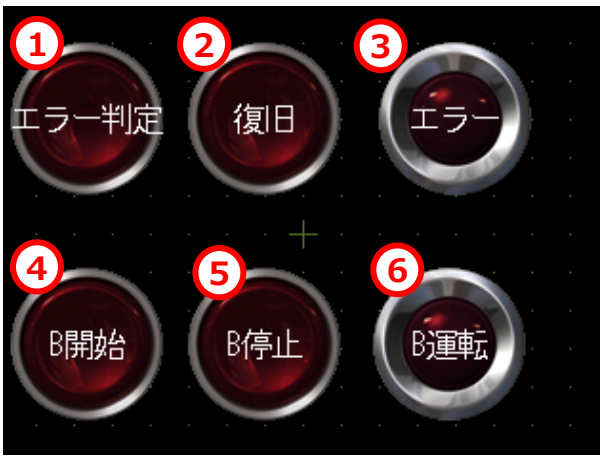
（2）ジャンプ（JMP）命令

ジャンプ命令では、メインプログラム中にラベル指定した場所まで、ジャンプし、実行します。
サンプル表示器画面上の①エラー判定スイッチをタッチすると、変数「エラー」がONし、ジャンプ命令（JMP）が実行され（3行目）、４行目を処理せず、ラベル指定した行（５行目）までジャンプします。ジャンプ命令が実行されている間は、表示器画面上の④~⑥は動作しません。

【 ロジックプログラム例】

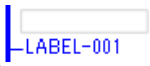


【表示器画面例】



【使用する命令】

ジャンプ命令（JMP）/ラベル（LABEL）
メインプログラム中にラベル指定した場所まで、ジャンプし、実行します。



【使用する変数例】

変数名	変数タイプ
エラー判定	ビット変数
復旧	ビット変数
エラー	ビット変数
B開始	ビット変数
B停止	ビット変数
B運転	ビット変数

【スイッチ/ランプ設定例】

①スイッチ
・ビットアドレス（変数）　：エラー判定
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

②スイッチ
・ビットアドレス（変数）　：復旧
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

③ランプ
・ビットアドレス（変数）　：エラー
※上記以外の設定は任意です。

④スイッチ
・ビットアドレス（変数）　：B開始
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

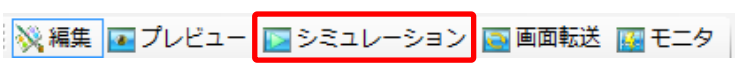
⑤スイッチ
・ビットアドレス（変数）　：B停止
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

⑥ランプ
・ビットアドレス（変数）　：B運転
※上記以外の設定は任意です。

動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします）

【状態バー】

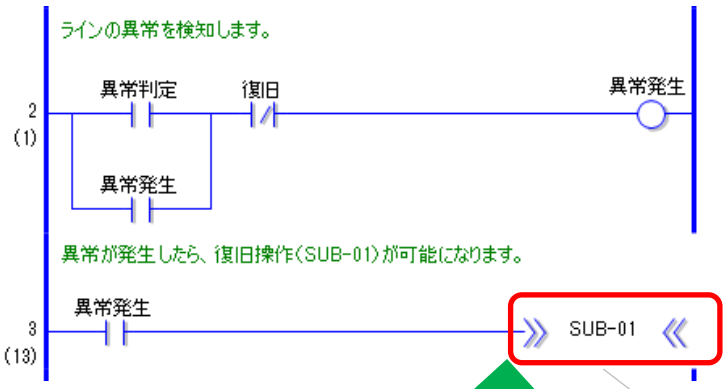


（3）サブルーチン処理（ JSR）命令

メインプログラムからサブルーチンプログラムを呼び出し、実行します。
同じ処理をサブルーチンとして作成することで、メインプログラム内で同じ処理を行うプログラムを何度も記述することがなくなります。そのため、メインプログラムをシンプルにすることができ、不具合発生率の低減やプログラムが読みやすくなるなどの利点があります。

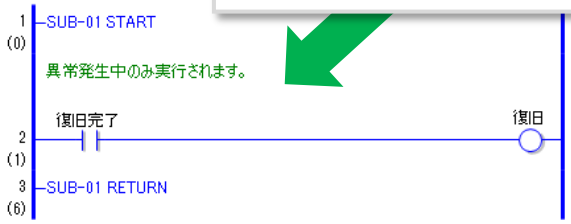
サンプル表示器画面上で変数「異常発生」がONすると、サブルーチン（SUB-01）を呼び出し/実行します（復旧完了することができます）

【 ロジックプログラム例】
MAIN



サブルーチン処理命令が実行されると、サブルーチン（SUB-01）が実行されます。

SUB-01



SUB-01の処理が終了すると、MAINのJSR命令の次の行へ戻り、処理を続けます。

【表示器画面例】

【使用する命令】

サブルーチン処理開始（JSR）
メインプログラムからサブルーチンプログラムを呼び出し、実行します。

サブルーチン（SUB-○○）
メインプログラムから呼び出すプログラムです。

【スイッチ/ランプ設定例】

①スイッチ
・ビットアドレス（変数） ：異常判定
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

②スイッチ
・ビットアドレス（変数） ：異常発生
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

③ランプ
・ビットアドレス（変数） ：復旧完了
※上記以外の設定は任意です。

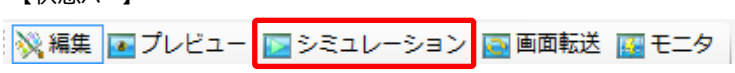
【使用する変数例】

変数名	変数タイプ
異常判定	ビット変数
異常発生	ビット変数
復旧	ビット変数
復旧完了	ビット変数

動作確認しよう(シミュレーションの起動)

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします（またはF12キーをタッチします）

【状態バー】



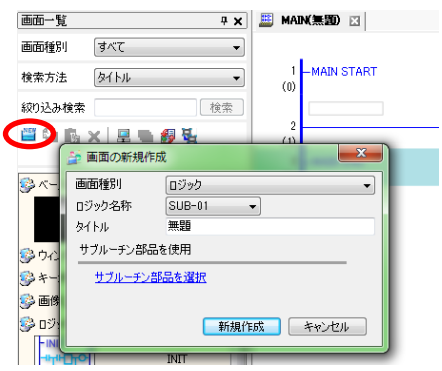
メモ（以下は余白です。ご自由にお使いください）

ONE POINT

サブルーチンの作り方

サブルーチンとは、プログラムの中で何度も必要になる定型的な処理を一つのプログラムにまとめて、メインプログラムから呼び出せるようにしたものです。

画面一覧ワークスペースの をクリックし、「画面の新規作成」から作成します。



ONE POINT

ファンクションブロック

ファンクションブロックとは、よく使う演算処理などのロジックプログラムを1つの関数としてまとめたものです。関数化することでロジックプログラムが簡素化でき、プログラム作成の工数が削減できます。

1、ファンクションブロックの作成方法

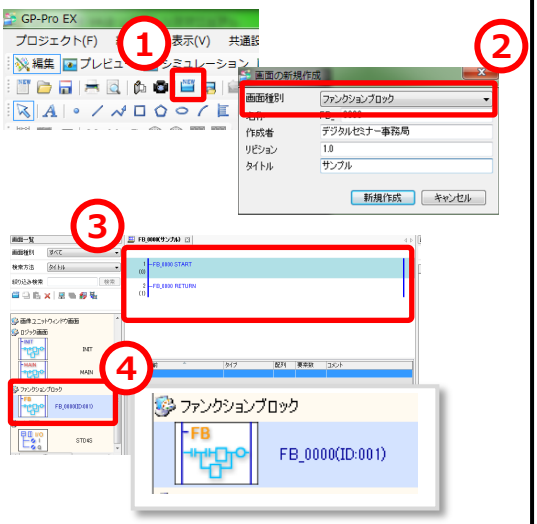
（1）新規に作成する

① NEW をクリックします。

②「画面の新規作成」ダイアログボックスの「画面種別」で「ファンクションブロック」を選択し、「新規作成」をクリックします。

③ロジックプログラムを作成します。

④作成したファンクションブロックは画面一覧に登録されます。



（2）作成済みのロジックプログラムを流用し、作成する

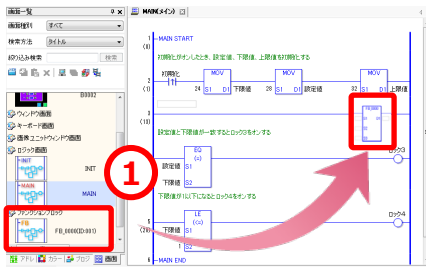
①ファンクションブロックにしたい行を選択し、右クリックします。

②ショートカットメニューから「ファンクションブロックとして」登録をクリックします。

③作成したファンクションブロックは画面一覧に登録されます。

2、ファンクションブロックを挿入する

①「画面一覧」の「ファンクションブロック」から、挿入する行にドラッグ＆ドロップします。



※ファンクションブロックを挿入する位置をクリックし、「命令の挿入」「ファンクションブロックの挿入」でもファンクションブロックを配置できます。

メモ（以下は余白です。ご自由にお使いください）

Pro-face™
by Schneider Electric

第4章

ライン操作盤をつくろう

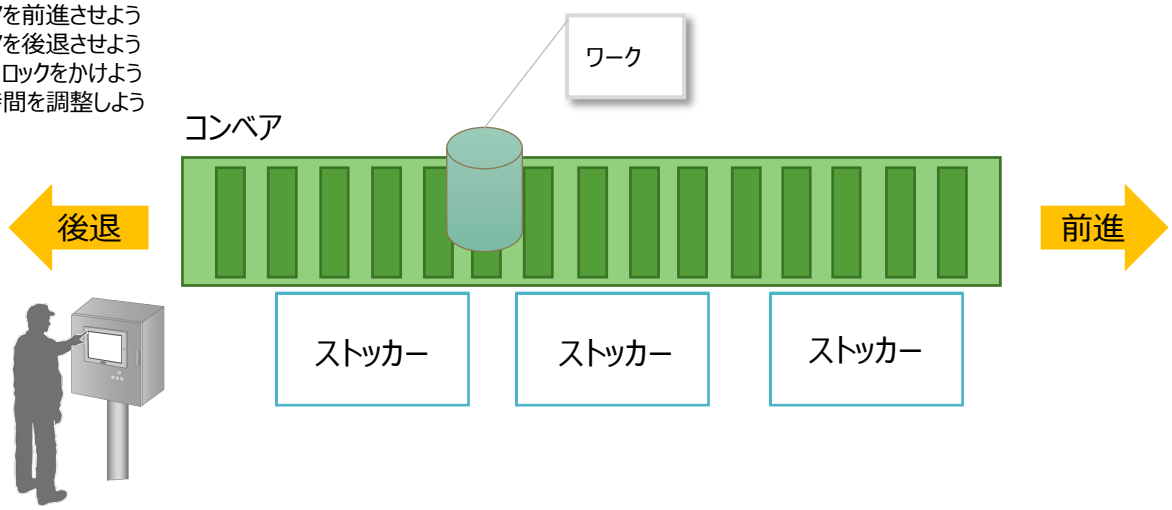
Ⅰ　実習について　4-2
Ⅱ　【やってみよう】コンベアを前進させよう　4-3
Ⅲ　【やってみよう】コンベアを後退させよう　4-9
Ⅳ　【やってみよう】インターロックをかけよう　4-13
Ⅴ　【やってみよう】準備時間を調整しよう　4-17

メモ（以下は余白です。ご自由にお使いください）

Ⅰ　実習について

この章では、ベルトコンベアを使ったライン装置の操作盤を想定し、STC6000のロジックプログラムと表示器画面を作成します。
ライン装置の機能を4つに分け、それぞれ要求仕様と完成画面例を参考に実習を行います。
実習後は、シミュレーション機能を使って、動作確認してください。

【ライン装置の4つの機能】
①コンベアを前進させよう
②コンベアを後退させよう
③インターロックをかけよう
④準備時間を調整しよう



1.ライン操作盤とは

（1）完成画面例と動作イメージ

①コンベアを前進させよう（4-3ページ）
・「前進開始」スイッチをタッチします。
・「前進準備中」ランプが3秒間点灯します（3秒間暖機を行います）
・「前進準備中」ランプが消灯すると、「コンベア前進」ランプが点灯します。
（コンベアの前進が開始されます）
・「停止」スイッチをタッチすると「前進準備中」「コンベア前進」が消灯します
（準備または前進を即座に停止します）

②コンベアを後退させよう（4-9ページ）
・「後退開始」スイッチをタッチします。
・「後退準備中」ランプが3秒間点灯します（3秒間暖機を行います）
・「後退準備中」ランプが消灯すると、「コンベア後退」ランプが点灯します。
（コンベアの後退が開始されます）
・「停止」スイッチをタッチすると「後退準備中」「コンベア後退」が消灯します
（準備または後退を即座に停止します）

③インターロックをかけよう（4-13ページ）
・「コンベア前進」ランプが点灯中は、後退の操作を開始することができません。
・「コンベア後退」ランプが点灯中は、前進の操作を開始することができません。

④準備時間を調整しよう（4-16ページ）
・表示器画面上のデータ表示器を使って前進準備時間を変更します。
・表示器画面上のデータ表示器を使って後退準備時間を変更します。

（2）システム環境（実習環境）

ここでは、以下の環境を例にシステム設定を行います。
・表示器付きコントローラー：STC-6300TA（シンク）
※型式 PFXSTC6300TADDKE
・接続機器：デジタル/メモリリンク（STC6000単独システム）

Ⅱ 【やってみよう】コンベアを前進させよう

1.要求仕様

下記の要求仕様を満たすロジックプログラムおよび表示器画面を作成しましょう。

①コンベアを前進させよう
・「前進開始」スイッチをタッチします。
・「前進準備中」ランプが3秒間点灯します（3秒間暖機を行います）
・「前進準備中」ランプが消灯すると、「コンベア前進」ランプが点灯します。
（コンベアの前進が開始されます）
・「停止」スイッチをタッチすると「前進準備中」「コンベア前進」ランプが消灯します（準備または前進を即座に停止します）

（1）使用する変数

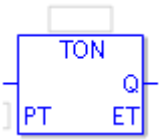
変数名	変数タイプ
前進開始	ビット変数
前進準備中	ビット変数
停止	ビット変数
コンベア前進	タイマ変数

（2）使用する命令

a接点
（NO）

b接点
（NC）

コイル出力
（OUT）

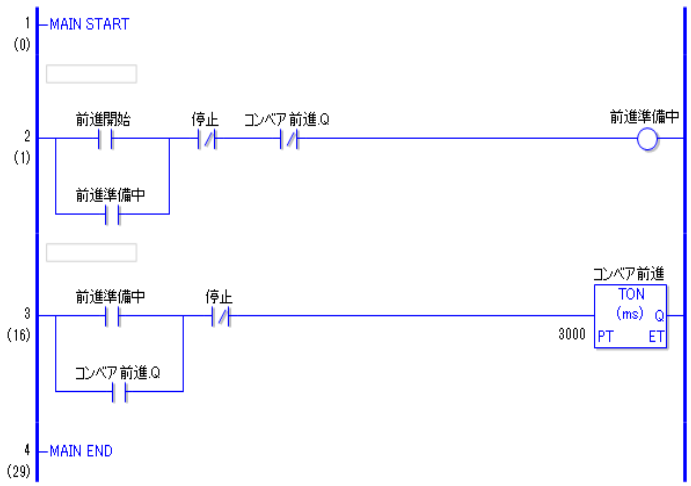


オンディレイタイマ
（TON）

（3）完成画面例

新規プロジェクトの作成は、システム設定（表示器タイプ・接続機器タイプ・I/O設定）を行った後は、エディタ上部にある「状態バー」に沿って、設定を進めます。

【ロジックプログラム例】



【表示器画面例】

2.解説　作成方法（システム設定）

①GP-Pro EXの起動

デスクトップのGP-Pro EXのアイコンをダブルクリックします。

GP-Pro EX

新規プロジェクトに作成名前をつけて保存しましょう。

②表示器タイプ

ウィザードに従い、表示器タイプを指定します。

シリーズ：STC6000Series
　　STC63※※Series（表示器の大きさ）
機種：　STC-6300TA VGA(640*480)
設置方法：横型

③接続機器設定

接続機器を指定します。
STC6000を単独で使用する場合は、以下の設定を行います。

メーカー：デジタル
シリーズ：メモリリンク
ポート：COM1

④I/Oの設定

EXモジュール/TM3ユニットを用いない場合は、「なし」を選択します。

I/Oドライバ：無

⑤ロジック画面作成

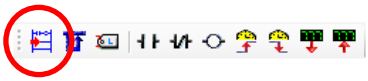
ロジック画面作成をクリックし、ロジック画面を表示します。

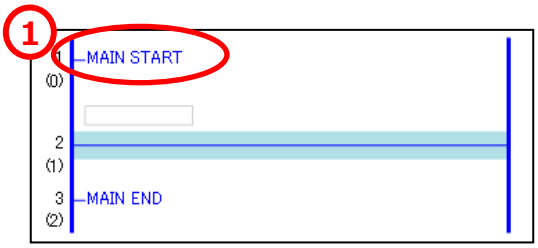
3.解説　作成方法（ロジックプログラム）

（1）前進準備

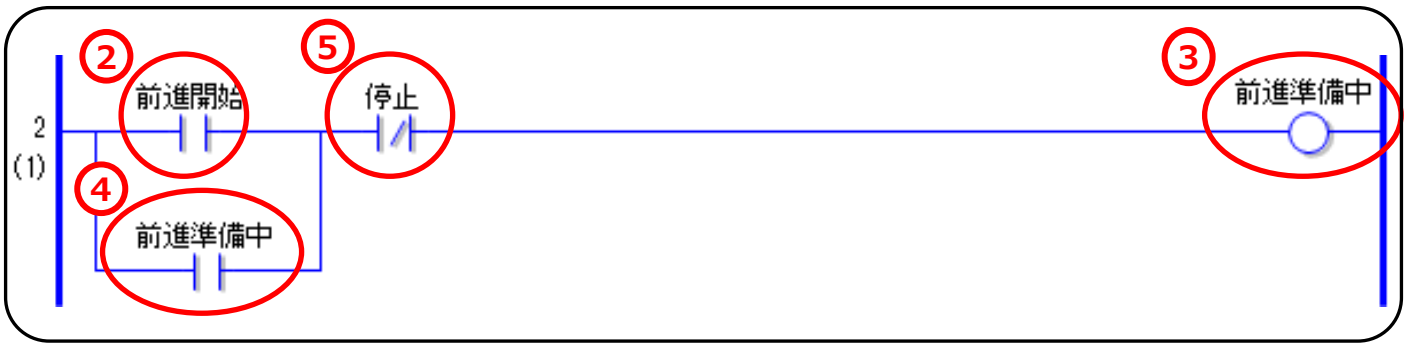
「前進開始」がONすると、「前進準備中」もONし、分岐上にある自分自身の接点によってバイパス回路を作り、動作を保持する
ロジックプログラム（自己保持回路）を作ります。

①「MAIN START」をクリックで選択し、
「行の挿入」アイコンをクリックします。

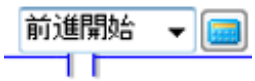




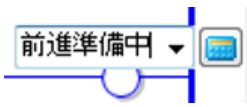
行が挿入されたら、②～⑥の設定を行います。



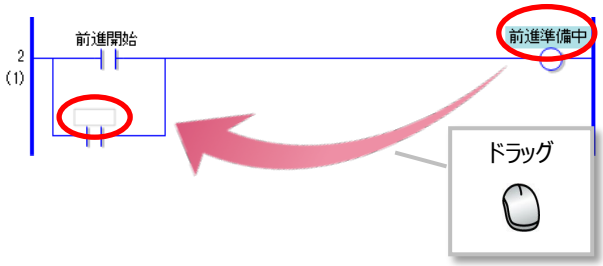
② NO(a接点）アイコンをクリックします。
変数名を『前進開始』と入力し、ビット変数として登録します。



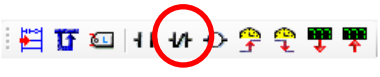
③ OUT(出力コイル）のアイコンをクリックします。
『前進準備中』と入力し、ビット変数として登録します。

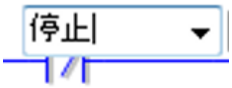


④ ②をまたぐように分岐し、NO(a接点）を挿入してから、
コイル『前進準備中』の変数をドラッグで、コピーします。



⑤次に、停止処理を行います
a接点『前進開始』とコイル『前進準備中』の間に、
NC（ｂ接点）を挿入します。
変数名を『停止』と入力し、ビット変数として登録します。

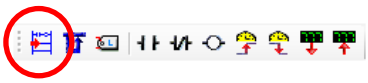


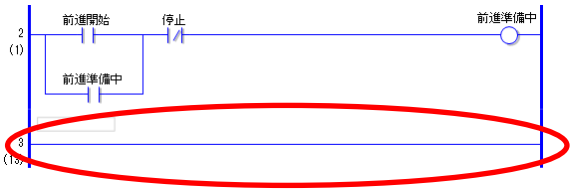


（2）前進

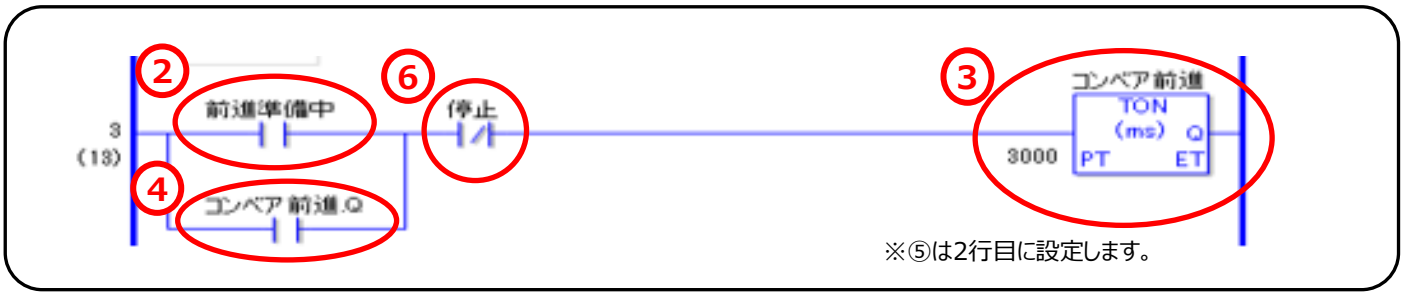
「前進準備」が3秒間行われると、「前進」がONするロジックプログラムを作ります（オンディレイタイマを使用します）

①「行2」をクリックで選択し、
「行の挿入」アイコンをクリックします。



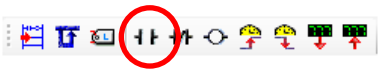


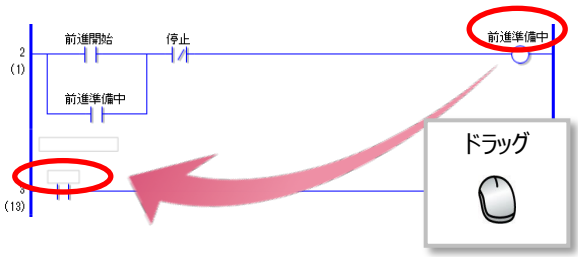
行が挿入されたら、②～⑥の設定を行います。



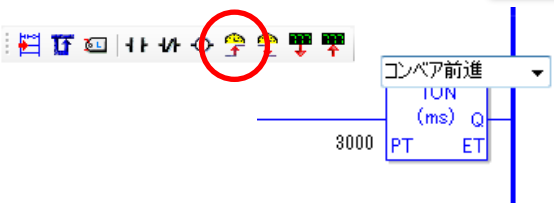
※⑤は2行目に設定します。

② NO(a接点）アイコンをクリックします。
変数『前進準備中』をコイルからa接点にドラッグ＆ドロップでコピーします。

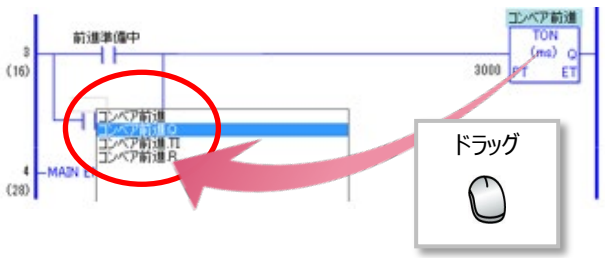




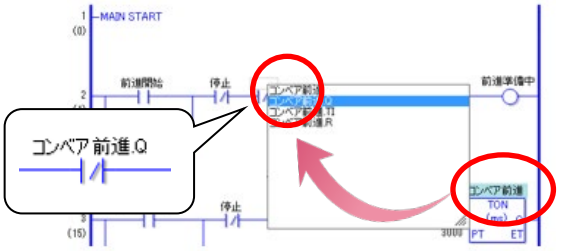
③ 3行目を選択し、TON(オンディレイタイマ）命令を挿入し、変数名を『コンベア前進』と入力しタイマ変数として登録します。
（PT（設定値）は、「3000ms」にします。
これにより、オンディレイタイマに信号が導通すると3秒間、遅れてONします。



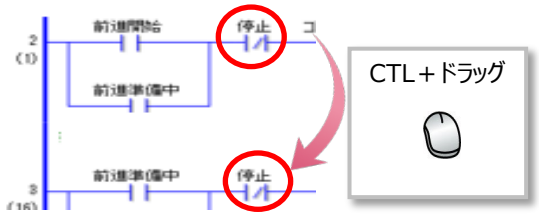
④ ②をまたぐように分岐し、NO(a接点）を挿入してから、
オンディレイタイマ『コンベア前進』の変数名をドラッグし、コピーします。
変数名『コンベア前進』をドラッグすると複数の変数候補リストが
表示されます。
TON命令が出力するとONになる『コンベア前進.Q』を選択してください。



⑤続いて完了処理を挿入します（2行目）
2行目の『停止』の右隣にNC(b接点）を挿入し、TON命令の変数名『コンベア前進』をドラッグし、変数候補リストから、『コンベア前進.Q』を選択してください。



⑥最後に、停止処理を挿入します。
『停止』のNC(b接点）をドラッグでコピーします（CTL＋ドラッグ）



4.解説　作成方法（表示器画面）

（1）前進/前進準備の操作

ここではコンベア操作画面で、前進操作用のスイッチとランプを作成します。

スイッチとランプの設定内容は、下図を参考に作成してください。

【 表示器画面例】

①ランプ
・ビットアドレス（変数）：前進準備中
・銘板：前進準備中
※上記以外の設定は任意です。

②ランプ
・ビットアドレス（変数）：コンベア前進.Q
・銘板：コンベア前進
※上記以外の設定は任意です。

③スイッチ
・ビットアドレス（変数）：前進開始
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

④スイッチ
・ビットアドレス：停止
・ビット動作：ビットモーメンタリ
※上記以外の設定は任意です。

（2）表示器画面の作成に便利な方法　その１

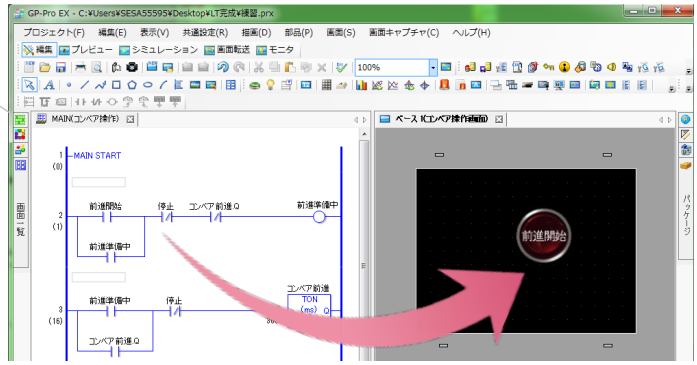
表示器画面の作成に便利な方法を、２つ紹介します。

【ロジックプログラムの命令をパーツ変換/配置する方法】

①　2画面表示にします。
「表示」 -「画面ブロック」-「横2画面表示」をクリックします。

作画画面とロジックプログラムを上下ブロックに分けて同時に表示させることができます。

②命令を選択し、ベース画面上にドラッグすると、自動的にパーツに変換し、配置されます。



※すべての部品が初期状態の形状で貼り付けられますので、任意でお好みの形状に変更してください。

ドラッグ＆ドロップでのパーツの貼り付け

ロジックプログラムから、表示器画面に命令をパーツに変換し/配置する際は、変数だけでなく、命令全体をドラッグします。

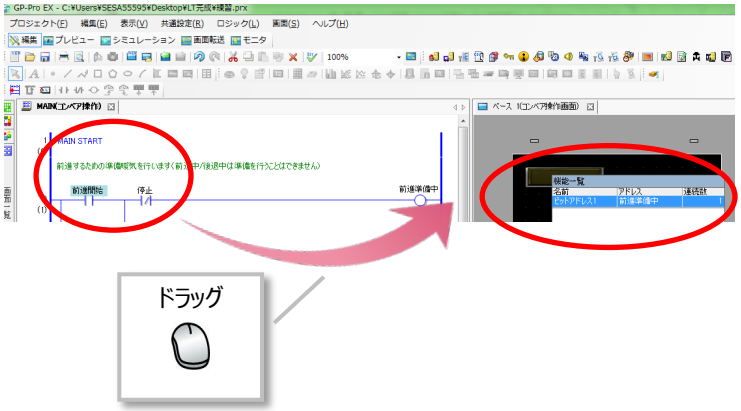
○ 前進開始　　× 前進開始

（3）表示器画面の作成に便利な方法　その２

【変数のみコピーする方法】

既に表示器画面に、パーツが配置されている場合は、変数のみをロジックプログラムからコピーすることができます。

①変数を選択し、ベース画面上のコピーしたいパーツに
ドラッグすると、変数のみをコピーすることができます。

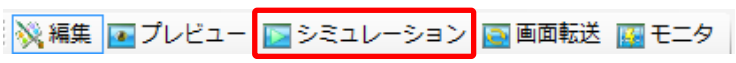


5.動作確認しよう（シミュレーション）

（1）シミュレーションの起動

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします(またはF12キーをタッチします)

【状態バー】



Ⅲ 【やってみよう】コンベアを後退させよう

1.要求仕様

下記の要求仕様を満たすロジックプログラムおよび表示器画面を作成しましょう。

②コンベアを後退させよう
・「後退開始」スイッチをタッチします。
・「後退準備中」ランプが3秒間点灯します（3秒間暖機を行います）
・「後退準備中」ランプが消灯すると、「コンベア後退」ランプが点灯します。
（コンベアの後退が開始されます）
・「停止」スイッチをタッチすると「後退準備中」「コンベア後退」が消灯します
（準備または後退を即座に停止します）

（1）使用する変数

作成状況		変数名	変数タイプ
作成済み		前進開始	ビット変数
作成済み		前進準備中	ビット変数
作成済み		停止	ビット変数
作成済み		コンベア前進	タイマ変数
		後退開始	ビット変数
		後退準備中	ビット変数
		コンベア後退	タイマ変数

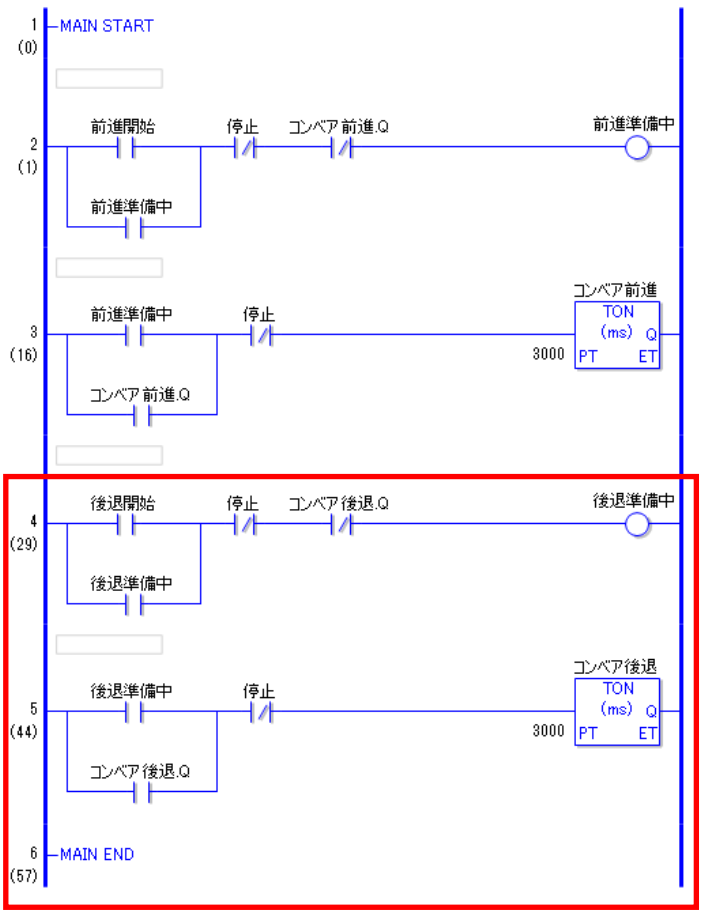
（2）使用する命令

a接点（NO）　b接点（NC）　コイル出力（OUT）　オンディレイタイマ（TON）

（3）完成画面例

前進の作成方法を参照して、コンベアの後退処理を作成してみましょう。

【ロジックプログラム例】



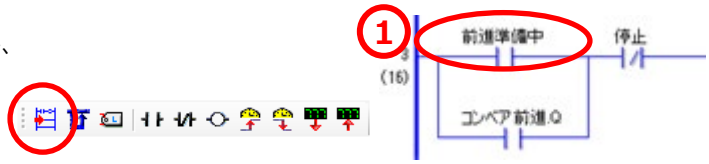
【表示器画面例】

2.解説　作成方法（ロジックプログラム）

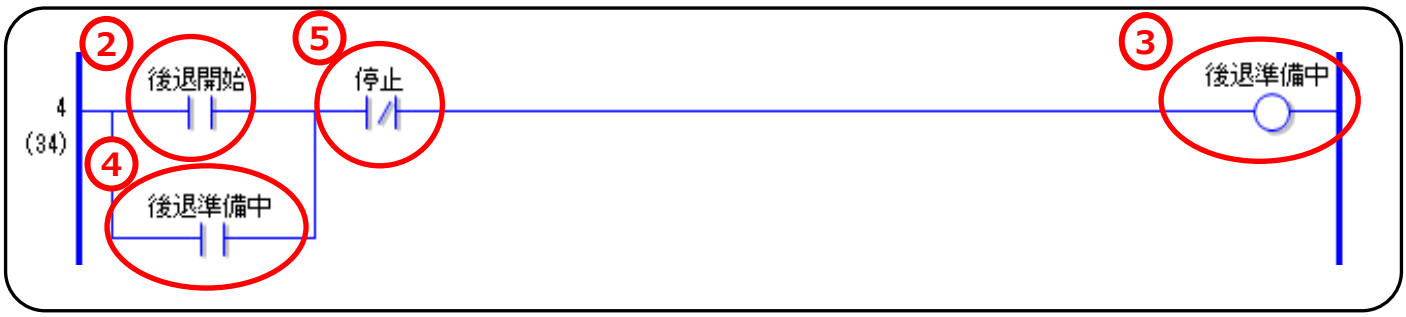
（1）後退準備

「後退開始」がONすると、「後退準備中」がONし、分岐上にある自分自身の接点によってバイパス回路を作り、動作を保持するロジックプログラムを作ります。

①３行目「前進準備中」行をクリックで選択し、
「行の挿入」アイコンをクリックします。



行が挿入されたら、②～⑥の設定を行います。

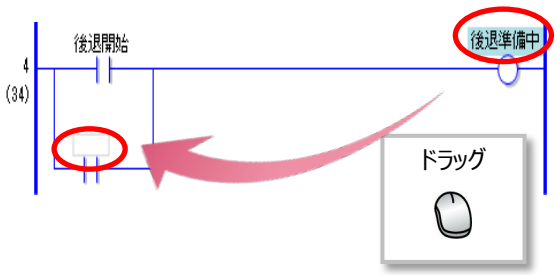


②　NO(a接点）アイコンをクリックします。
変数名を『後退開始』と入力し、ビット変数として登録します。

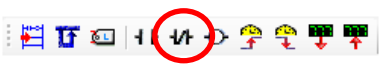
③　OUT(出力コイル）のアイコンをクリックします。
『後退準備中』と入力し、ビット変数として登録します。

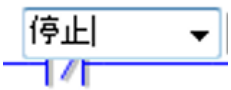
これで、『後退開始』接点がONしたら、
『後退準備中』コイルが出力します。

④　②をまたぐように分岐し、NO(a接点）を挿入してから、
コイル『後退準備中』の変数名をドラッグし、コピーします。



⑤次に、停止処理を挿入します。
4行目『後退開始』の横にNC（ｂ接点）を挿入します。
変数名に『停止』と入力します。

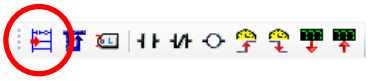


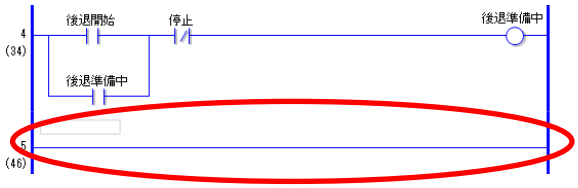


（2）後退

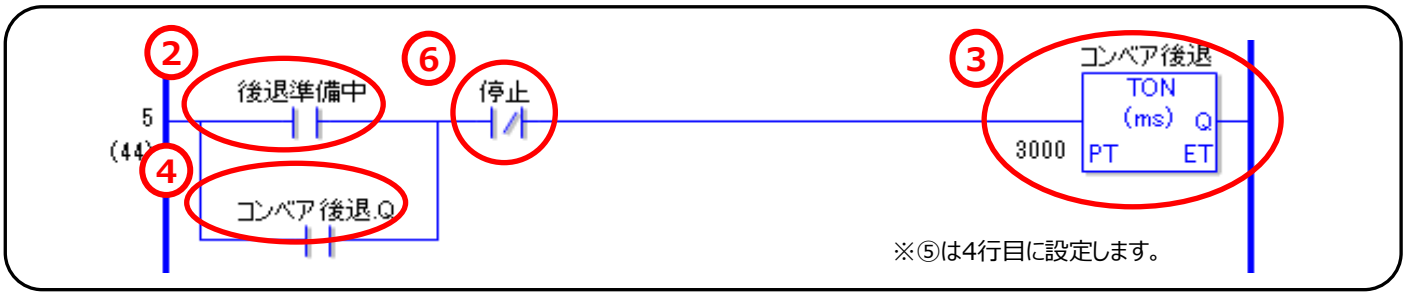
「後退準備」が3秒間行われると、「後退」がONするロジックプログラムを作ります（オンディレイタイマを使用します）

①「行4」をクリックで選択し、
「行の挿入」アイコンをクリックします。

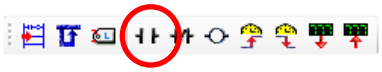


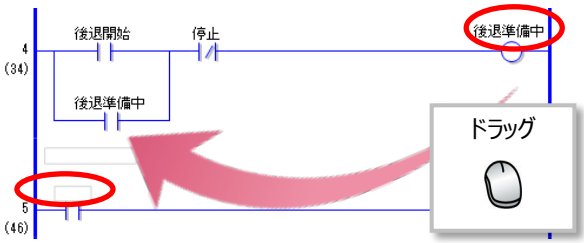


行が挿入されたら、②～⑥の設定を行います。

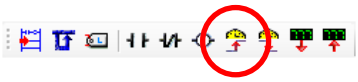


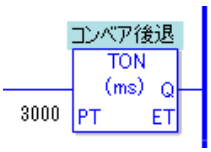
② NO(a接点）アイコンをクリックします。
変数『後退準備中』をコイルからa接点にドラッグでコピーします。





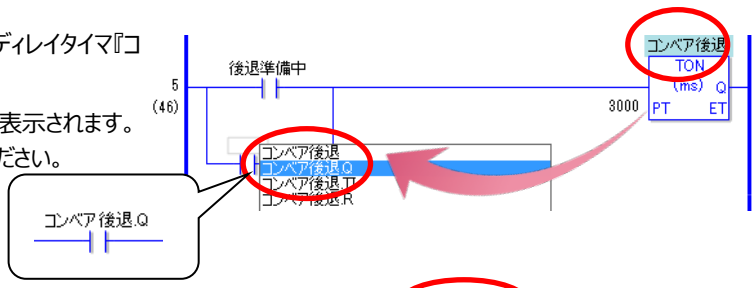
③ ３行目を選択し、TON(オンディレイタイマ）命令を挿入し、
変数名を 『コンベア後退』と入力しタイマ変数として登録します。
（PT（設定値）は、「3000ms」にします。
これにより、オンディレイタイマに信号が導通すると3秒間、遅れてONします。



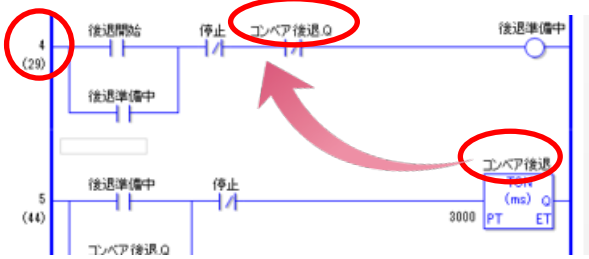


④ ②をまたぐように分岐し、NO(a接点）を挿入してから、オンディレイタイマ『コンベア後退』の変数をドラッグし、コピーします。

変数名『コンベア後退』をドラッグすると 複数の変数候補リストが表示されます。
TON命令が出力するとONになる『コンベア後退.Q』を選択してください。



⑤続いて完了処理を挿入します（４行目）
4行目の『停止』の隣にNC(b接点）を挿入し、TON命令の変数名『コンベア後退』をドラッグし、変数候補リストから、『コンベア後退.Q』を選択してください。

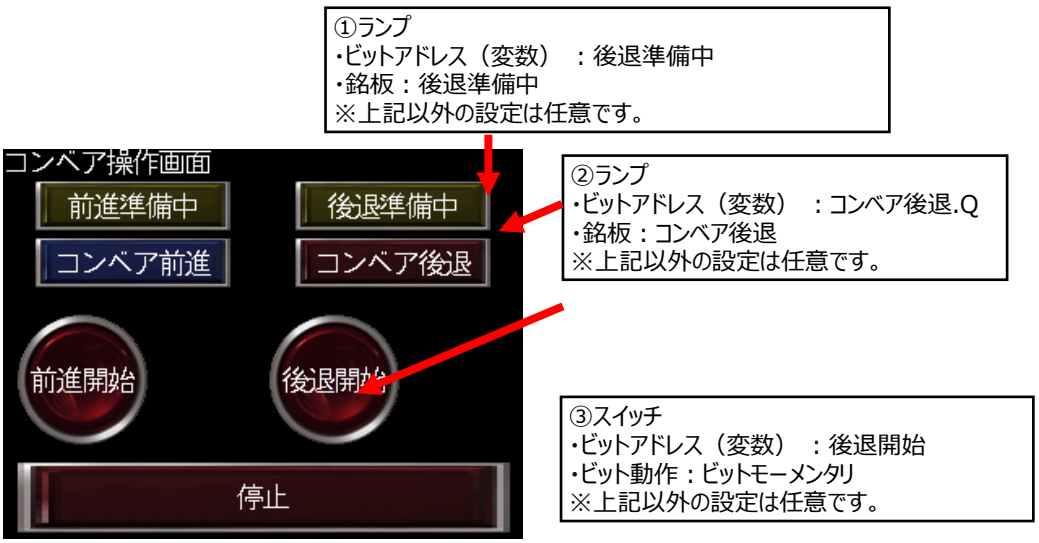


⑥最後に、停止処理を挿入します。
『停止』のNC(b接点）をドラッグでコピーします（CTL＋ドラッグ）

3.解説　作成方法（表示器画面）

スイッチとランプの設定内容は、下図を参考に作成してください。

【表示器画面例】

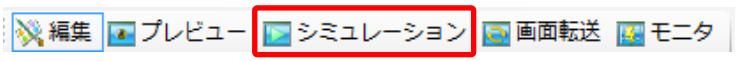


4.動作確認しよう（シミュレーション）

（1）シミュレーションの起動

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします(またはF12キーをタッチします)

【状態バー】



Ⅳ 【やってみよう】インターロックをかけよう

1.要求仕様

下記の要求仕様を満たすロジックプログラムおよび表示器画面を作成しましょう。

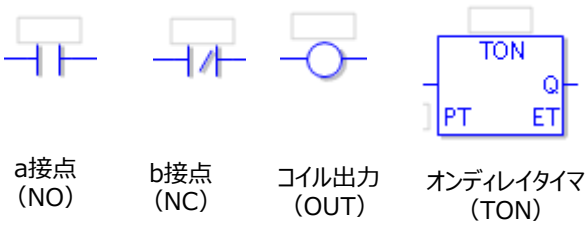
③インターロックをかけよう
・「コンベア前進」ランプが点灯中は、後退の操作を開始することができません。
・「コンベア後退」ランプが点灯中は、前進の操作を開始することができません。

（1）使用する変数

作成状況	変数名	変数タイプ
作成済み	前進開始	ビット変数
作成済み	前進準備中	ビット変数
作成済み	停止	ビット変数
作成済み	コンベア前進	タイマ変数
作成済み	後退開始	ビット変数
作成済み	後退準備中	ビット変数
作成済み	コンベア後退	タイマ変数

※この実習では、変数の新規作成はありません。

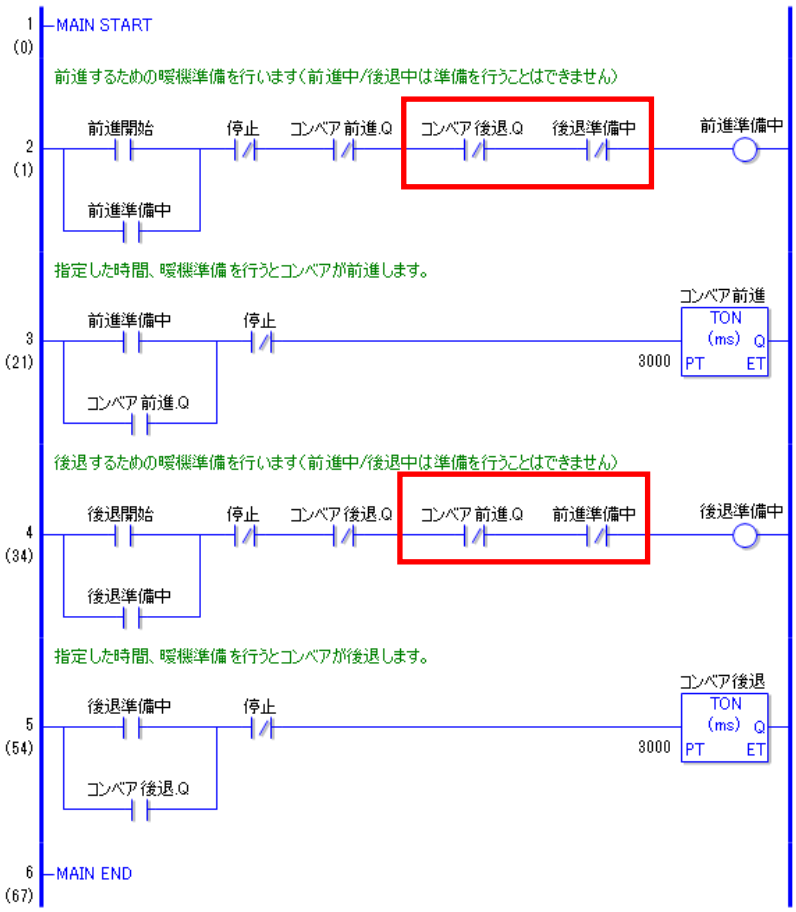
（2）使用する命令



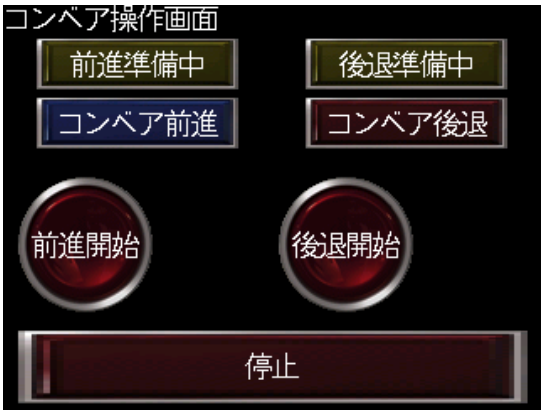
a接点（NO）　b接点（NC）　コイル出力（OUT）　オンディレイタイマ（TON）

（3）完成画面例

【ロジックプログラム例】



【表示器画面例】



※この実習では、表示器画面の作成/編集はありません。

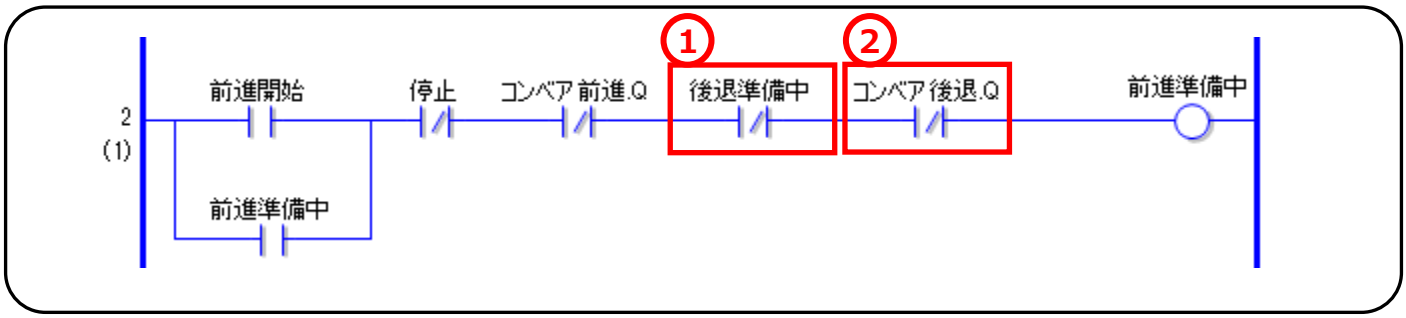
2.解説　作成方法

インターロック回路は、同時に動いて欲しくない回路にお互いのコイルをb接点に配置することで完成します。
「前進および前進準備」と「後退および後退準備」が同時に動かないようにするために、各行にそれぞれのb接点を配置していきます。

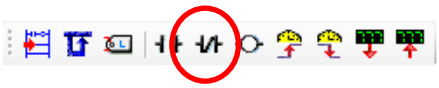
（1）インターロック　その１

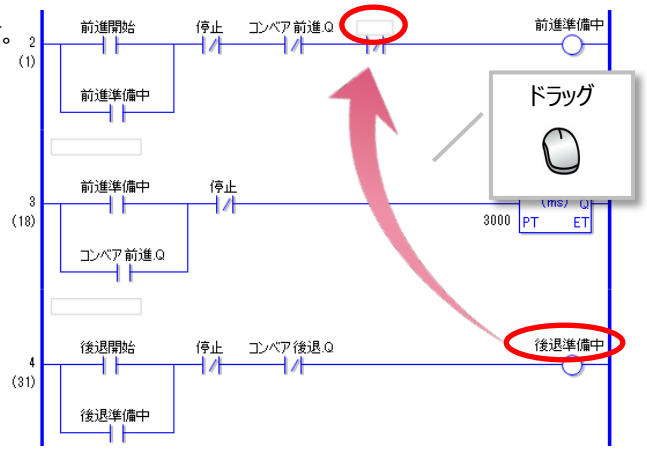
まずは『「コンベア前進」ランプが点灯中は、後退および後退準備（暖機）を開始することができません。』
という要求仕様を満たすために、2行目に『後退準備中』および『コンベア後退.Q 』をb接点に配置します。

2行目に①～②の設定を行います。

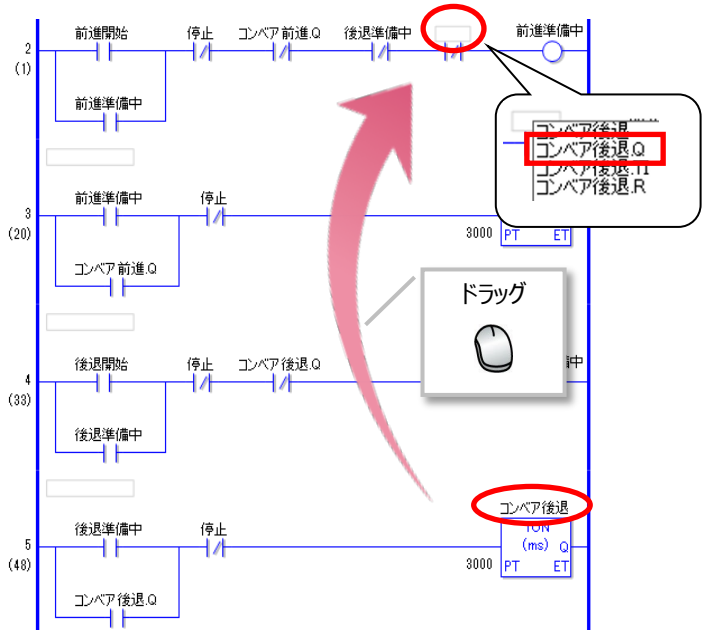


①NC（ｂ接点）を挿入します。変数名『後退準備中』を割り当てます。
（下の行のものをコピー＆ペースト）





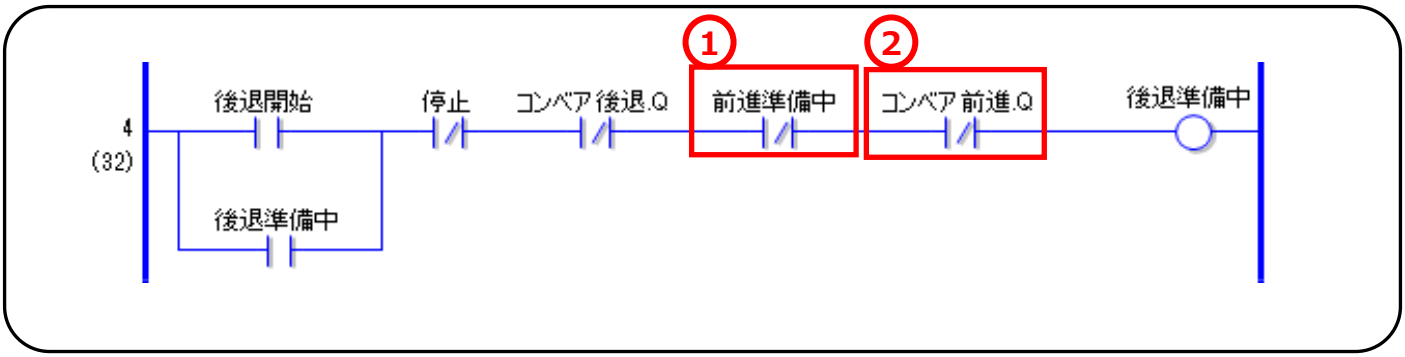
②NC（ｂ接点）を挿入します。TON命令の変数名『コンベア後退』をドラッグし、変数候補リストから『コンベア後退.Q』を選択してください。



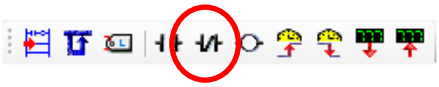
（2）インターロック　その2

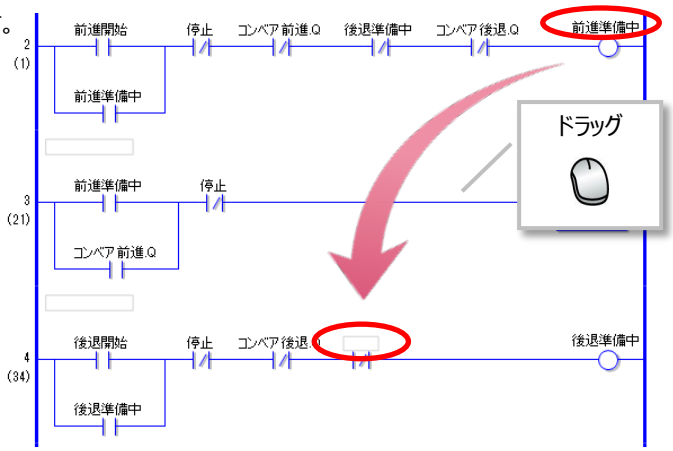
『「コンベア後退」ランプが点灯中は、前進および前進準備（暖機）を開始することができません。』
という要求仕様を満たすために、4行目に『前進準備中』および『コンベア前進.Q 』をb接点に配置します。

4行目に①～②の設定を行います。

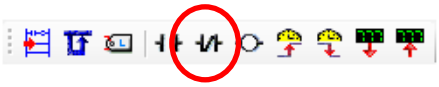


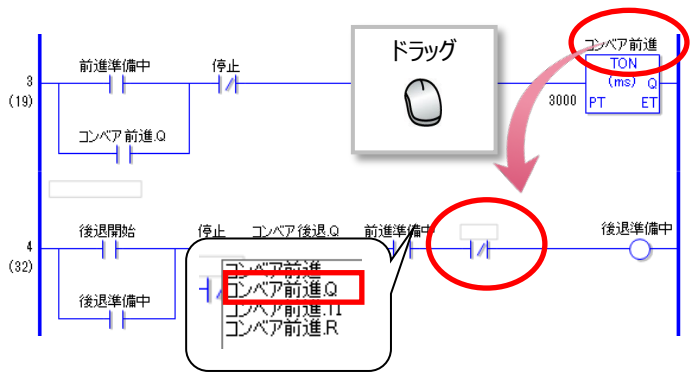
①NC（ｂ接点）を挿入します。変数名『前進準備中』を割り当てます。
（上の行のものをコピー＆ペースト）





②NC（ｂ接点）を挿入します。TON命令の変数名『コンベア後進』をドラッグし、変数候補リストから『コンベア前進.Q』を選択してください。

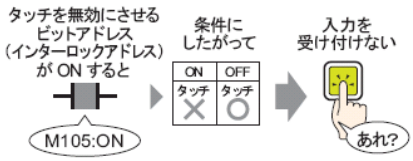


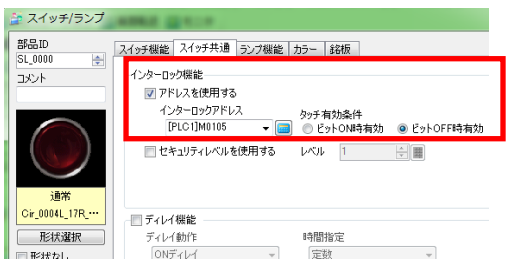


ONE POINT

表示器画面のスイッチのインターロック

表示器画面上に配置したスイッチの設定で、誤操作防止のためのインターロックを設定することもできます。



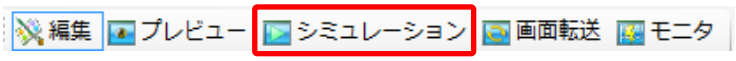


3.動作確認しよう（シミュレーション）

（1）シミュレーションの起動

シミュレーションを使えば、パソコン上で動作確認できます。
状態バーの「シミュレーション」をクリックします(またはF12キーをタッチします)

【状態バー】



メモ（以下は余白です。ご自由にお使いください）

V 【やってみよう】準備時間を調整しよう

1.要求仕様

下記の要求仕様を満たすロジックプログラムおよび表示器画面を作成しましょう。

④準備時間を調整しよう
・表示器画面上のデータ表示器を使って前進準備時間を変更します。
・表示器画面上のデータ表示器を使って後退準備時間を変更します。

（1）使用する変数

作成状況	変数名	変数タイプ
作成済み	前進開始	ビット変数
作成済み	前進準備中	ビット変数
作成済み	停止	ビット変数
作成済み	コンベア前進	タイマ変数
作成済み	後退開始	ビット変数
作成済み	後退準備中	ビット変数
作成済み	コンベア後退	タイマ変数

※この実習では、変数/命令の新規作成はありません。

（2）使用する命令

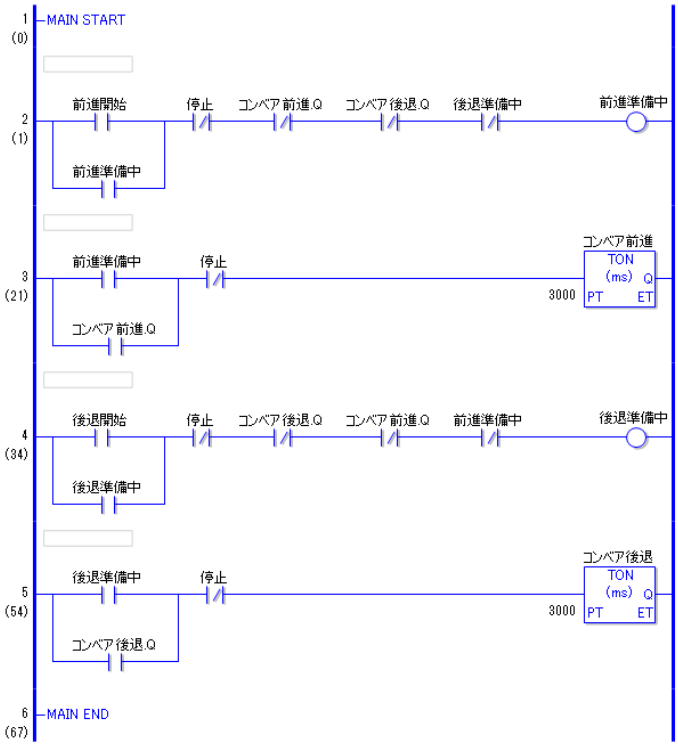
a接点（NO）　b接点（NC）　コイル出力（OUT）　オンディレイタイマ（TON）

※この実習では、変数/命令の新規作成はありません。

（3）完成画面例

【ロジックプログラム例】

※この実習では、ロジックプログラムの作成/編集はありません。



【表示器画面例】

2.解説　作成方法（表示器画面）

【表示器画面例】
※ロジックプログラムの変更はありませんので、表示器画面のみの作成となります。

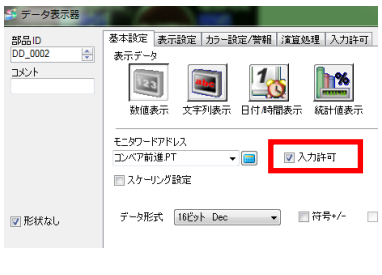
暖機時間（○○.PT）をタッチし、テンキーから暖機の時間を設定します。
準備時間では、経過時間を表示します。

①データ表示器
・ワードアドレス：コンベア前進.PT
・入力許可：チェックあり
※上記以外の設定は任意です。

②データ表示器
・ワードアドレス：コンベア前進.ET
・入力許可：チェックなし
※上記以外の設定は任意です。

③データ表示器
・ワードアドレス：コンベア後退.PT
・入力許可：チェックあり
※上記以外の設定は任意です。

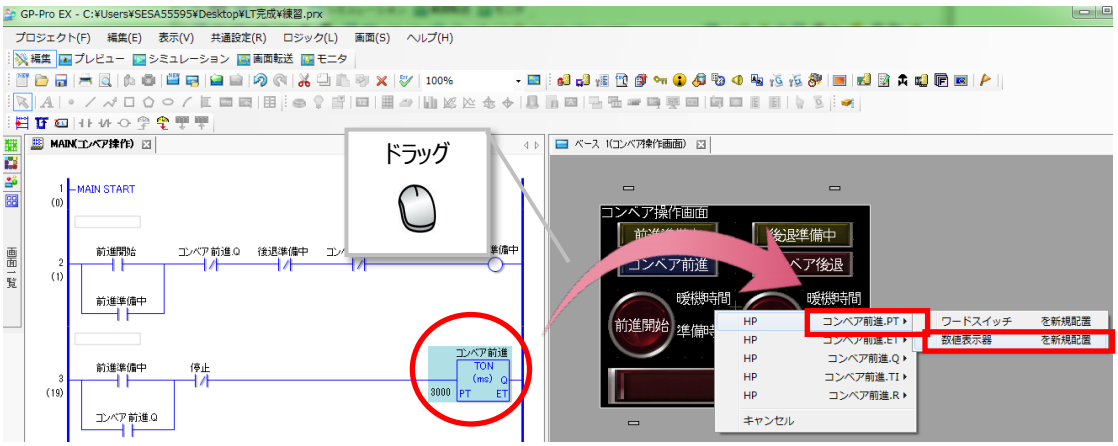
③データ表示器
・ワードアドレス：コンベア後退.ET
・入力許可：チェックなし
※上記以外の設定は任意です。



①②のデータ表示器には、「入力許可にチェック」を入れてください。
（タッチすると、テンキーが表示されます）

（1）表示器画面の作成（データ表示器）

①のデータ表示器を配置します（スイッチ同様に、ロジックプログラム上の命令を選択して作画画面へドラッグ＆ドロップで配置します）



タイマ命令を選択してドラッグすると、候補の変数と部品がリスト表示されます。
ここでは、暖機時間をテンキー入力できる数値表示器を配置したいので、『コンベア前進.PT』「数値表示器」を選択します。
また、テンキーからの入力を行う場合はデータ表示器の配置後、「入力許可」にチェックを入れます。
※すべての部品が初期状態の形状で貼り付けられますので、任意でお好みの形状に変更してください。

②③④のデータ表示器も上記を参考に配置してください。

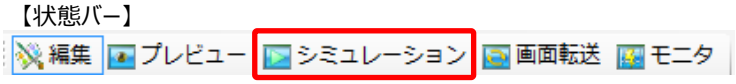
3.動作確認しよう（シミュレーション）

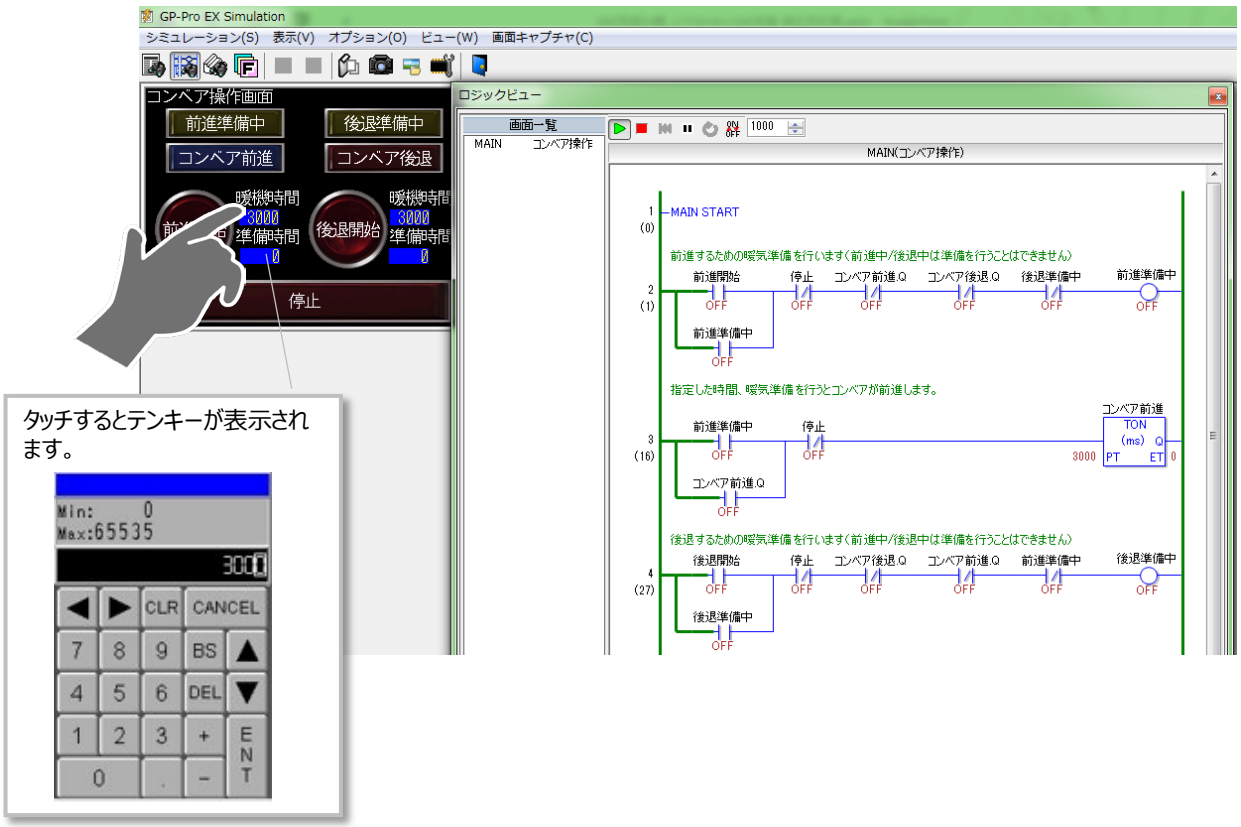
（1）シミュレーションの起動

シミュレーションを使えば、パソコン上で動作確認できます。

①「シミュレーション」をクリックします(またはF12キーをタッチします)

【状態バー】





タッチするとテンキーが表示されます。

メモ（以下は余白です。ご自由にお使いください）

Pro-face™
by Schneider Electric

付録

さらに便利に使おう

Ⅰ　STC6000シリーズのメモリ構成　付-2
Ⅱ　変数のアドレス方式　付-5
Ⅲ　STC6000シリーズの動作処理　付-6
Ⅳ　オフラインモード　付-9
Ⅴ　用語集　付-12

メモ（以下は余白です。ご自由にお使いください）

Ⅰ　STC6000シリーズのメモリ構成

（1）STC6000シリーズのメモリ構成

ここでは、STC6000の内部メモリについて説明します。STC6000には、3つの内部メモリがあり、それぞれ下記のプログラムやデータを格納します。

STC6000シリーズ

STC6000シリーズの内部メモリ（全モデル共通）

	表示器画面（表示/操作）	コントロール（制御）
アプリケーションメモリ（FLASH EPROM）	作画プログラム（作画画面） 64MB	ロジックプログラム ※1 132KB（15,000ステップに相当）
SRAM	バックアップメモリ 320KB （アラーム履歴やログデータ等を記憶）	変数エリア 64KB
内部デバイス（#Internal）	LSエリア（10,000ワード） USRエリア（30,000ワード）	

※1　最大60,000ステップに切り替え可能です。その場合は、作画プログラム容量が1Mバイト少なくなります（64MB→63MB）

ONE POINT

内部デバイス

内部デバイスとは、表示器内に用意されているデバイスです。

表示器の内部デバイス（#INTERNAL）は、下図のようにLSエリアとUSRエリアの2つの領域で構成されています。
特にUSRエリアは、自由に使える30,000ワード（USR0からUSR29,999）を用意していますので、表示器内部で演算した値を一時的に格納したり、表示器内部で一時的に処理(制御)する場合のデータ格納先として適しています。

そのデータを参照する場合、[接続機器]で[#INTERNAL]を選択し、アドレスを入力します。

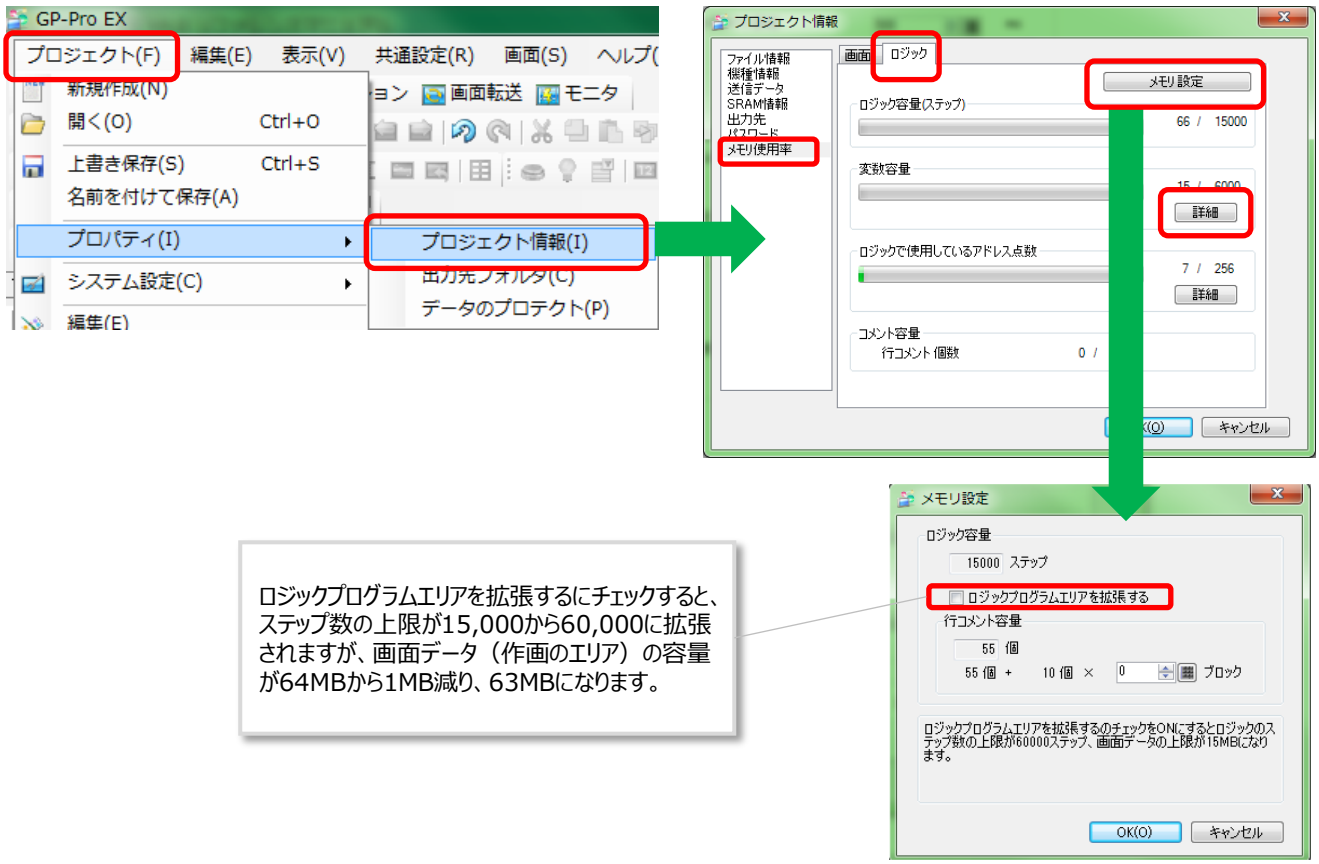
内部デバイス（#INTERNAL）

内部デバイスの参照方法

（2）ロジックプログラム容量（ステップ数）

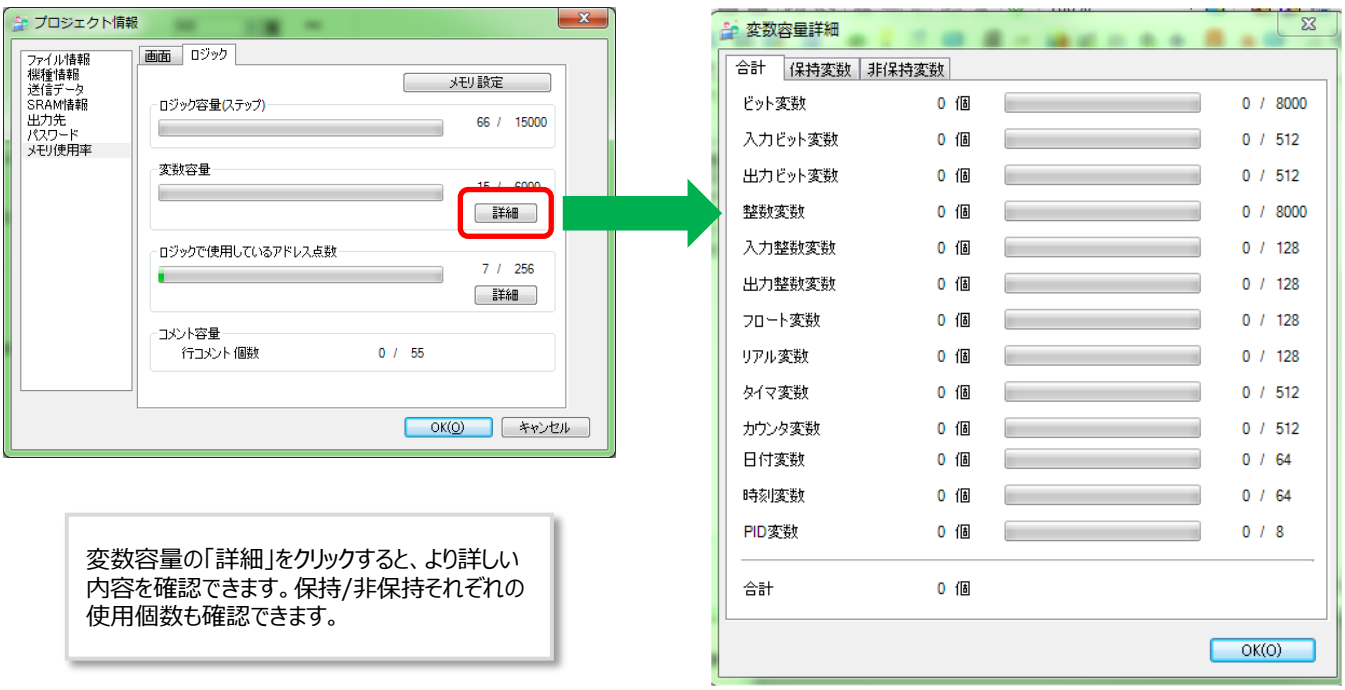
＜ロジックプログラムのメモリ使用率の確認方法＞

（1）GP-Pro EXメニューバーの「プロジェクト」の中の「プロパティ」から「プロジェクト情報」を選択します



ロジックプログラムエリアを拡張するにチェックすると、ステップ数の上限が15,000から60,000に拡張されますが、画面データ（作画のエリア）の容量が64MBから1MB減り、63MBになります。

（3）変数容量



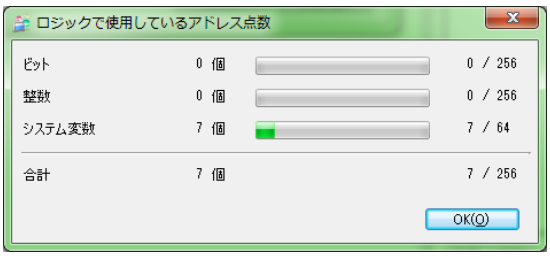
変数容量の「詳細」をクリックすると、より詳しい内容を確認できます。保持/非保持それぞれの使用個数も確認できます。

（4）ロジックプログラムの変数とPLCアドレス

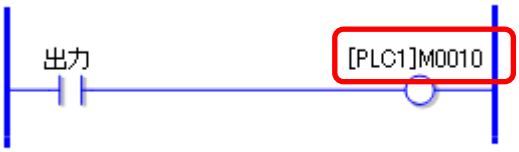
ロジックプログラムでPLCアドレスを使用することができます。
ビットのON/OFF状態や数値データをPLCアドレスと共有することで、PLC側の制御プログラムとの連携ができます。

①ロジックで使用できるアドレスの登録制限は下記の通りです。

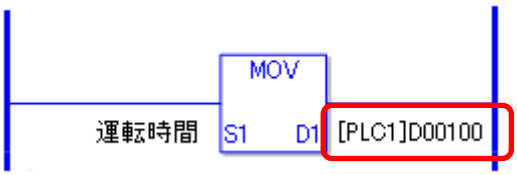
ワードデバイス：256個
ビットデバイス：256個



②使用例



例）出力ビットON/OFF状態をPLCの
ビットアドレスM10に送る。



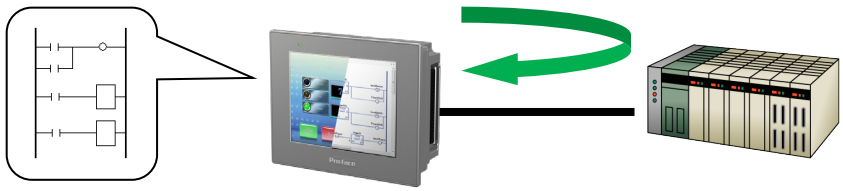
例）運転時間の値をPLCの
ワードアドレスD100に書き込む。

接続機器

PLCを接続する場合は、システム設定の接続機器設定にて使用する PLCのドライバを選択します。

ロジック上でPLCアドレスを直接、演算命令や比較命令で処理すると、 変数やGP内部アドレスを使用する場合と比べて、処理が遅れることが あります。
これはロジックプログラムのスキャンタイムに通信サイクルタイム、（PLCアドレスのデータを読み書きする時間）が加わるためです。

ロジック処理時間
（スキャンタイム）
＋
通信サイクルタイム



Ⅱ　変数のアドレス方式

変数登録には、これまで説明してきた「変数方式」と「アドレス方式」があります。「アドレス方式」とは、PLCのデバイスアドレス同様の名称で扱うことができる変数です。

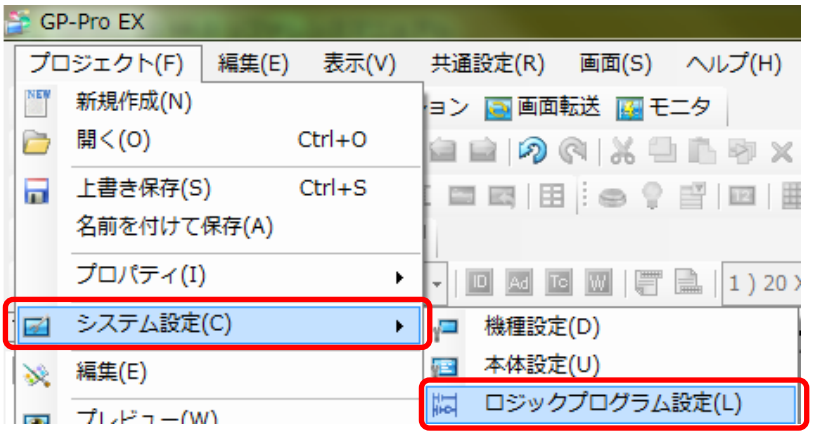
【アドレス方式の変数】

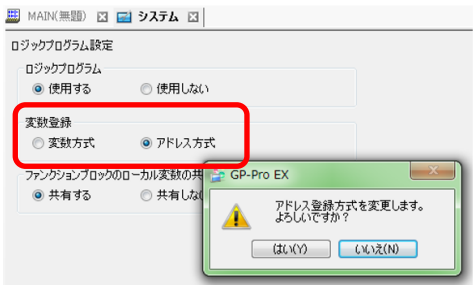
タイプ	ロジックアドレス			表示	サイズ	備考
ビット変数	X_0000	～	X_0511	10進	512	入力
	Y_0000	～	Y_0511	10進	512	出力
	M_0000	～	M_7999	10進	8000	内部
整数変数	I_0000	～	I_0127	10進	128	入力
	Q_0000	～	Q_0127	10進	128	出力
	D_0000	～	D_7999	10進	8000	内部
フロート変数	F_0000	～	F_0127	10進	128	内部
リアル変数	R_0000	～	R_0127	10進	128	内部
タイマ変数	T_0000	～	T_0511	10進	512	内部
カウンタ変数	C_0000	～	C_0511	10進	512	内部
日付変数	N_0000	～	N_0063	10進	64	内部
時刻変数	J_0000	～	J_0063	10進	64	内部
PID変数	U_0000	～	U_0007	10進	8	内部

「アドレス方式」にすると、アドレス（変数）を割り当てる際に、「#LOGIC」を指定することができます。

①アドレス方式の設定方法

「アドレス方式」を使用する際は、新規プロジェクト作成時に「システム設定」「ロジックプログラム設定」にて設定します。





アドレス方式の注意点

・アドレス方式の設定は、新規プロジェクト作成時のみです（変数登録後はアドレス方式に変更できません）

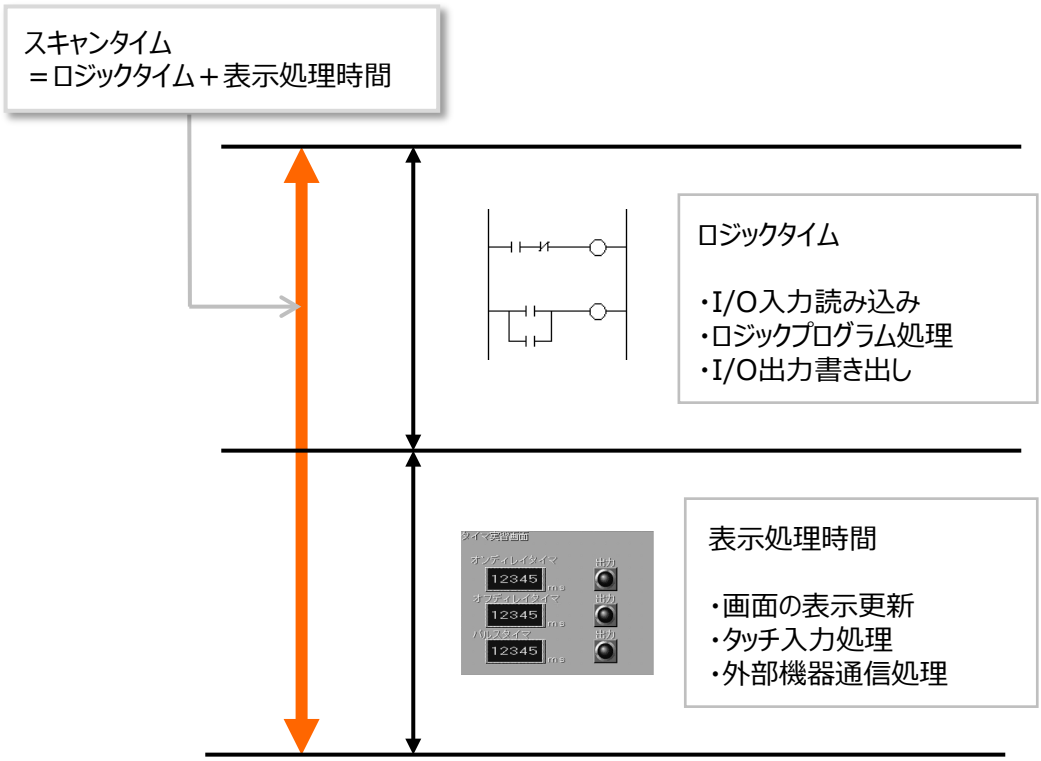
Ⅲ STC6000シリーズの動作処理

（1）STC6000シリーズの動作処理

STC6000では、大きく分け以下の２つの動作が順番に繰り返されます。

1. ロジックタイム（I/O入力＋ロジックプログラムの処理＋I/O出力）
2. 表示処理時間（画面の表示更新＋タッチパネル入力処理＋通信処理）

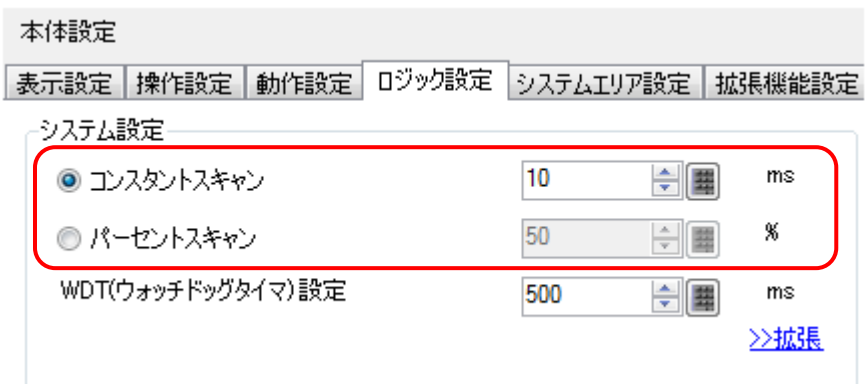
この一連の動作をスキャンといい、この動作に要する時間をスキャンタイムといいます。



（2）スキャンモード

スキャンタイムは、作成されたロジックプログラムや画面データによって変化しますので、最適なスキャンタイムで実行されるよう調節する必要があります。

制御（ロジックタイム）優先のコンスタントスキャンモードと、画面（表示処理時間）優先のパーセントスキャンモードの2種類あり、状況に応じてどちらかを選択して使用します（初期設定は制御優先です）

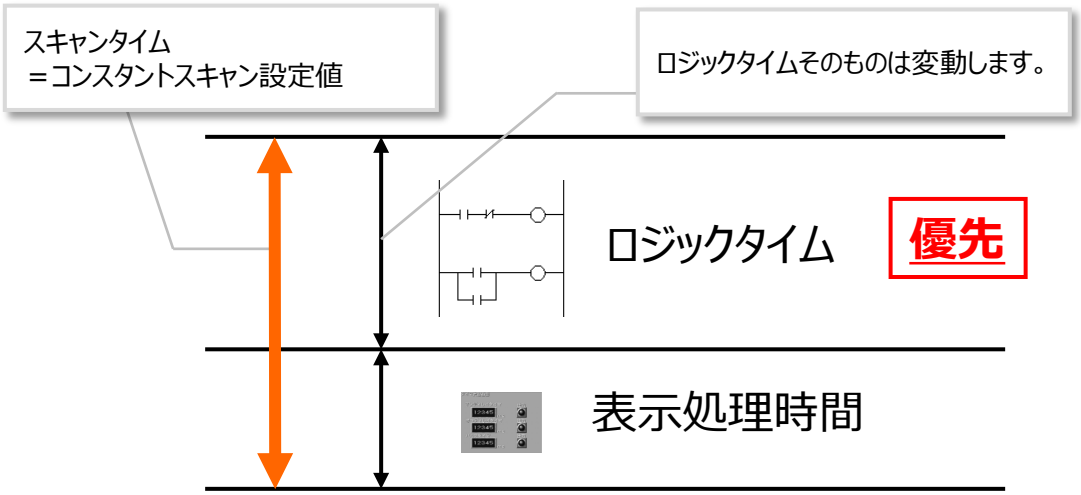


（3）コンスタントスキャンモード

スキャンタイムが一定になるように設定するモードです。最小設定値は10msです。

設定されたスキャンタイム内でロジックタイムが優先的に処理され、表示処理は残りの時間で処理されます。そのため、設定によって表示処理が遅くなることがあります。

制御を優先した設定と言え、監視がメインでタッチ操作が少なく、制御を優先するシステムに適しています。



表示処理時間 ＝ コンスタントスキャン設定値（ms） － ロジックタイム（可変）

スキャンタイムは設定した値が絶対ではなく、ロジックタイムが設定したスキャンタイムの50%を超えると、ロジックタイムを0.5で割り算した値に自動調整されます。プログラム容量が多くなり、万が一設定したスキャンタイムよりもロジックタイムが長くなっても、必ずスキャンタイムが一定時間になるように適切な値に調整されます。

例）

・コンスタントスキャン設定値：40ms

・ロジックタイム：30ms

計算式）

30ms（ロジックタイム）÷0.5=60ms(スキャンタイム)

60ms（スキャンタイム）-30ms（ロジックタイム）=30ms(表示処理時間)

スキャンタイムは40msから60msに自動調整され、一定周期で実行されます。

画面表示処理時間は30msとなります。

ONE POINT

ロジックタイムの実行時間をモニタする方法

コンスタントスキャンモードまたは、パーセントスキャンモードでスキャンタイムを

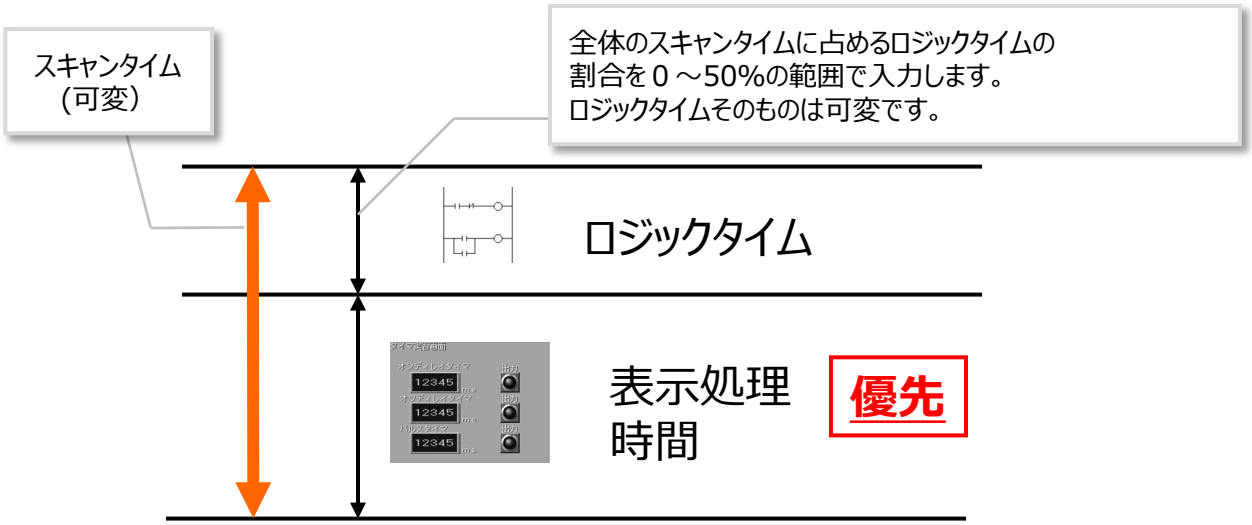
調節するには、実際のロジックタイムの実行時間を予め確認しておく必要があります。

システム変数「＃AvgLogicTime」を画面上に数値表示すると現在の実行時間の確認ができます。

（4）パーセントスキャンモード

ロジックタイムの長短に応じてスキャンタイムを可変させて動作するモード です。

ロジックタイムのスキャンタイムに占める割合(%)を決めることにより、ロジックタイムの増加による表示処理時間の圧迫を防ぐことができるため、画面の操作スピードや切り替えスピードを優先するシステムに適しています。



スキャンタイム(ms) ＝ ロジックタイム ÷ パーセントスキャン設定値（%）

スキャンタイムの中を占めるロジックタイムの割合を設定し、その設定値によってスキャンタイムが可変します。

つまりロジックタイムが長くなると、その分表示処理時間も長くなります。

したがって、ロジックタイムが長くなればなるほど、表示器上の表示更新速度は速くなりますが、ロジックプログラムの処理周期は遅くなります。

例）

・パーセントスキャン設定値：40%

・ロジックタイム：20ms

計算式)

(20ms（ロジックタイム）÷40(パーセントスキャン設定値)×100 = 50ms(スキャンタイム)

50ms(スキャンタイム) - 20ms（ロジックタイム） = 30ms(表示処理時間)

表示優先にしたい場合

・設定値（%）はロジックタイム内の個々の命令処理時間には影響を与えません。

・設定値の範囲は０～50%までです。

パーセントスキャン設定値を50%に設定した場合は、表示とロジックプログラムの処理は同じ時間になりますので、表示処理を優先する処理にはなりません。
表示処理を優先したくて本モードを選択した場合は、必ず50%未満の値を設定してください。

Ⅳ　オフラインモード

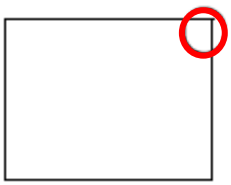
（1）オフラインモードとは

STC6000の初期設定をはじめ、様々な動作環境の設定や自己診断を行うことができます。
（詳細は各機種のユーザーズマニュアルをご参照ください）

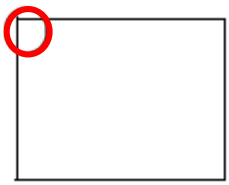
（2）オフラインモードの入り方

電源投入時

起動中画面表示後に、パネルの右上隅または左上隅(縦横40ドット以内)を3以上タッチします。

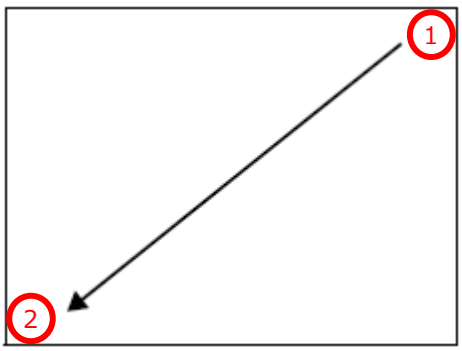


または

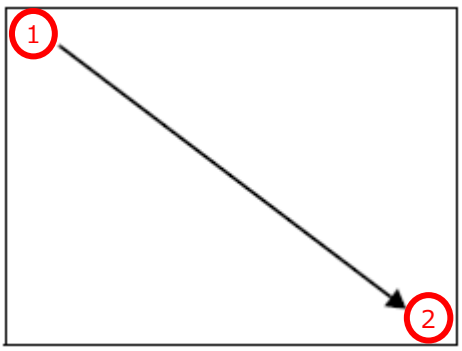


運転時

パネルの右上隅→左下隅または左上隅→右下隅(縦横40ドット以内)の順に0.5秒以内にタッチします。



または



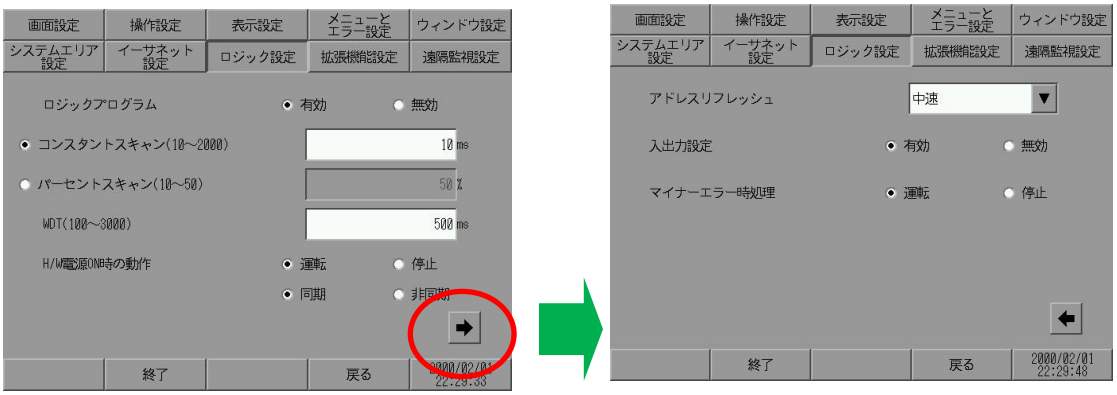
画面にシステムメニューが表示されますので「オフライン」をタッチします。

（3）よく使うオフラインモード

STC6000固有のオフラインメニューについて説明します。

①オフラインモードからの設定 その1

メインメニュー
↓
本体設定
↓
ロジック設定



・コンスタントスキャン、パーセントスキャンについては「付　スキャンタイムについて」をご参照ください。

・WDT (ウォッチドッグタイム)

スキャンタイムに対するウォッチドッグタイムを100～3000msの範囲で設定します。

100ms単位で数値を入力してください。0を入力すると無効です。

スキャンタイムがこの値を超えると、メジャー異常が発生しますので、スキャンタイムより長めの値を設定してください。

・H/W電源ON時の動作

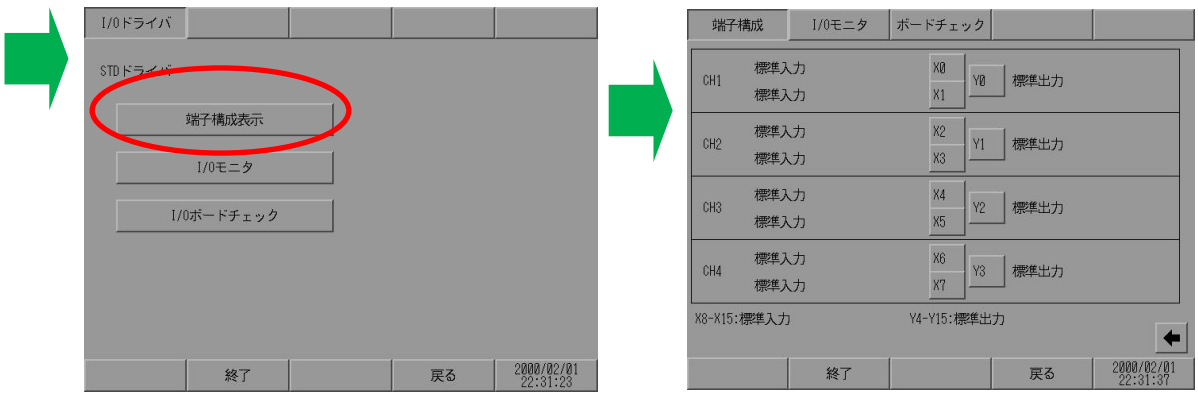
電源ON時の動作モード(運転、停止)を設定します。

運転：電源投入時、必ずプログラムはRUN状態

停止：電源投入時、必ずプログラムはSTOP状態

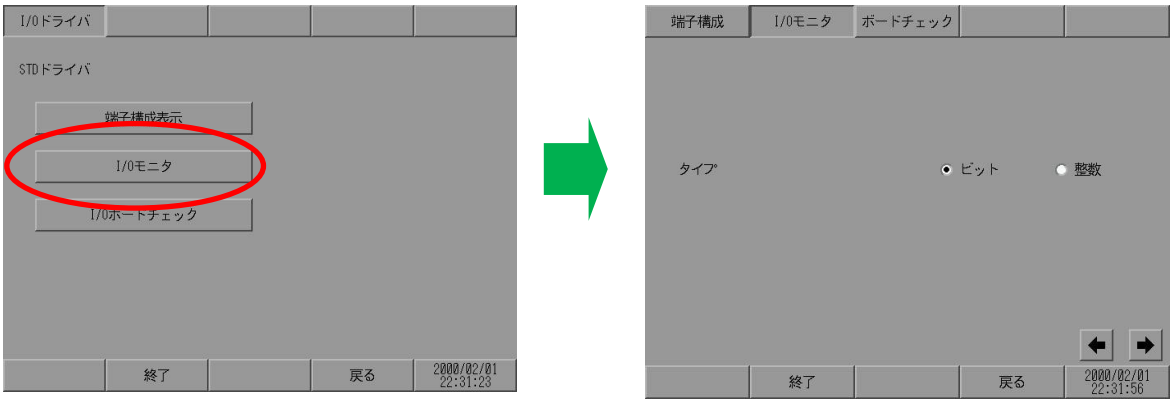
①オフラインモードからの設定 その2

メインメニュー
↓
周辺機器
↓
I/0ドライバ
↓
○○ドライバ



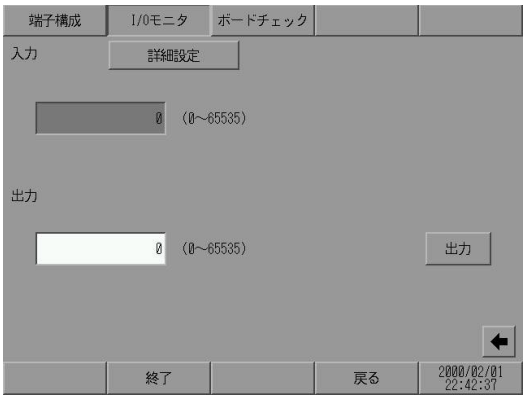
設定されている入力端子(X)、出力端子(Y)の端子構成を表示します。

標準入力と標準出力のモニタリング結果をビットまたは整数で表示選択できます。ただし特殊I/Oを使用している場合は[整数]表示できません。

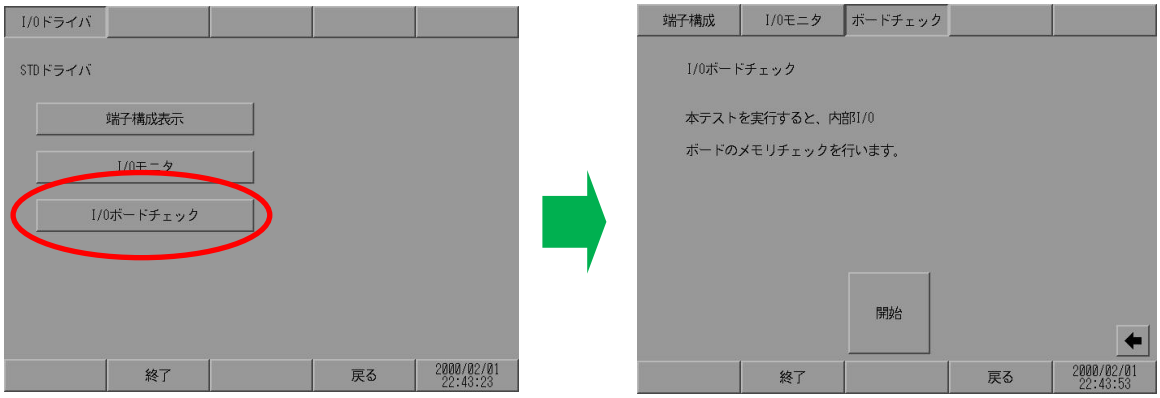


ビットモニタ

整数モニタ



I/Oボードが正常に動作するかどうかのチェックを行います。異常がある場合はエラーが表示されます。



V 用語集

用語	意味
CANopen	耐ノイズ性強化を考慮して設計されたデータ転送規格CAN(Controller Area Network）をベースにしたオープンネットワークです。 STC6000はマスターの通信ドライバーを搭載していますので、CANopenスレーブ機器と通信することができます。*2024年8月頃対応予定
スキャンタイム	ロジックプログラムの実行周期のことです。画面表示機能（画面表示、タッチパネル処理、通信処理）に必要な最低時間を確保するために、最小で10msとなります。 また表示画面の部品数やロジックプログラムの容量が増えた場合、その分スキャンタイムは延びます。
I/O	入出力インターフェイスのことです。
シンク出力	STC6000を選定するとき、出力タイプを選びます。STC6000と接続する機器の仕様にあわせて選定します。シンクタイプはマイナスがコモンとなりますので、電源のマイナス側に負荷の出力線をつなぎ、出力ONしたときに負荷からSTC6000側に電流が流れます。
ソース出力	プラスコモンとなり、電源のプラス側に負荷の出力線をつなぎ、出力0NしたときにSTC6000から負荷側に電流が流れます。
特殊I/O	「標準」以外のI/Oのこと。PWM出力、PLS出力、高速カウンタ入力、パルスキャッチ入力のことを指します。
PWM出力	特殊I/O機能のひとつで、主にモーター制御に使用します。パルス幅制御とも呼ばれ、加える電圧は一定に保ったまま、ONする時間を変化させることでモーターの回転数を制御します GP-Pro EXでは、ONしている時間とOFFしている時間の比率（%）と周期を設定し、指定のプログラムを作成することで簡単にPWM制御が実現できます。
PLS(パルス列）出力	特殊I/O機能のひとつで、位置決めや回転の制御に使用します。GP-ProEXでは、設定出力周波数を設定出力パルス数出力する通常パルス出力と、設定出力周波数まで徐々に周波数を上げていく加減速パルス出力を用意しています。
高速カウンタ（入力）	特殊I/O機能のひとつで、エンコーダーやサーボアンプと接続して高速パルスを取得することができます。また、一致出力機能もあるため、モータの回転数などを取り込み、到達したときに外部に出力することも可能ですのでフィードバック制御が可能です。
パルスキャッチ（入力）	特殊I/O機能のひとつで、主に割り込み処理が必要なときに使用します。ロジックのスキャンタイムより短い10μs以上(5μs以上のON)のパルスを取り込むことが可能です。
ピンアサイン	ピンの配置に対応するピンの信号が決められています。この事を指します。ピンと信号の配置を ピンアサイン と言います。
インクリメント	変数の値を 1 増やす処理のこと。実行回数や生産数など、数える処理に使用します。
デクリメント	変数の値を 1 減らす処理のこと。
ソース（S）	送信元（演算で使用する元データ）のことです。
ディスティネーション（D）	演算結果を格納する宛先のことです。
オペランド	演算における値のこと。GP-Pro EXロジックの命令におけるS1やD1に格納される値のことをオペランドと呼びます。

メモ（以下は余白です。ご自由にお使いください）

メモ（以下は余白です。ご自由にお使いください）

Pro-face™
by Schneider Electric

発行　：2023年08月（1.00）
発行元：シュナイダーエレクトリックホールディングス株式会社トレーニングセンター