



プログラマブルコントローラー

Modicon M221シリーズ

簡単プログラミング

はじめての **SoMachine Basic** テキスト



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SoMachine Basic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、シュナイダーエレクトリック社製プログラマブルコントローラー Modicon M221シリーズ（以下「M221」と言います）の基本的なラダープログラムの作成およびPro-face製表示器との連携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ラダー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。

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

プログラミングツール「SoMachine Basic」

画面作成ソフトウェア「GP-Pro EX」

#### ■ 解説している実施環境

プログラマブルコントローラー : Modicon M221シリーズ(TM221CE16R)

プログラマブル表示器 : GP-4114T(PFXGP4114T2D)

パソコン : Windows7が動作するパソコン

※本テキストは、「SoMachine Basic バージョン1.6」「GP-Pro EX バージョン4.08」で作成されています。ソフトウェアのバージョンにより、表記方法等が異なる場合があります。

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(旧：株式会社デジタル)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、シュナイダーエレクトリック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三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、シュナイダーエレクトリック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、何らかの保証や許諾を与えるものではありません。

## 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M221本体の故障およびM221の誤作動の場合、システム全体が安全側に働くようにM221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、M221の外部で回路構成してください。
- (2) M221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、M221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。

・M221の本体の電源を立ち上げる前に、M221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。M221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。

・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、絶対にプログラマブル表示器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プログラマブル表示器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。

・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。

・M221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。

・M221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。

## お問い合わせ

本テキストの内容に関するご質問は、「シュナイダーエレクトリック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(旧：株式会社デジタル)

シュナイダーエレクトリックカスタマーケアセンター：0570-056-800

<http://jpn.proface.co.jp/customer/contact.html>

平日 9:00～19:00

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



## **取り付け上の警告事項**

- ・M221の解体は絶対に行わないでください。高電圧部分がM221内部にあり、M221を解体すると感電の恐れがあります。
- ・M221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。通電中に行くと感電の恐れがあります。
- ・M221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。電池を誤って交換すると、電池が爆発する恐れがありますので、交換は行わないでください。
- ・交換が必要な場合には、お買い求めの代理店またはシュナイダーエレクトリックカスタマーケアセンターまでご連絡ください。

## **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、M221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。
- ・M221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。
- ・M221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。通電中に行くとI/Oユニットの故障や誤動作の原因になります。

## **廃棄時の注意事項**

- ・M221を廃棄する時は、産業廃棄物として扱ってください。

## 序章：プログラマブルコントローラー「Modicon M221シリーズ」

I. テキストのご紹介	序-2
II. Modicon M221シリーズ/SoMachine Basicのご紹介	序-4
1. Modicon M221シリーズの特徴	
2. プログラミングソフトウェア「SoMachine Basic」	

## 第1章：プログラミングツール「SoMachine Basic」の基礎知識

I. 開発環境を整えよう	1-2
1. 開発環境	
II. Modicon M221シリーズとPro-face製表示器	1-3
1. Pro-face製表示器との接続	
2. [Schneider Electric SA] Modbusシリアル接続について	
3. [Schneider Electric SA] Modbusイーサネット接続について	
III. SoMachine Basicの基礎知識	1-8
1. プロジェクトの作成方法	
2. SoMachine Basicの画面構成	
3. 新規プロジェクトの作り方	
IV. まずは、これだけ！基本操作！	1-18
1. プログラミングの基本操作	
2. ラングの追加	
3. 命令の割付け	
4. アドレスの割付け	
5. デバッグ	

## 第2章：基本回路をマスターしよう

I. 【理解しよう】基本回路 その1	2-2
1. ON回路	
2. OFF回路	
3. AND回路	
4. OR回路	
5. NOT回路	
II. 【やってみよう】自己保持回路	2-12
1. 【実習】自己保持回路とは	
2. 自己保持回路 完成例	
3. 【解説】自己保持回路	
III. 【理解しよう】基本回路 その2	2-16
1. インターロック回路	
2. SET/RSET回路	

## 第3章：応用回路をマスターしよう

I. 応用回路 その1	3-2
1. タイマー回路	
2. カウンター回路	
II. 応用回路 その2	3-8
1. データ転送回路	
2. インクリメント/デクリメント演算	
3. 比較演算回路	

## 第4章：ライン操作盤をつくろう

I. 実習について	4-2
1. ライン操作盤とは	
II. 実習の解説	4-3
1. 【設定】 SoMachine Basic (M221シリーズ)	
2. 【プログラミング】 SoMachine Basic (M221シリーズ)	
3. 【通信】 SoMachine Basic (M221シリーズ)	
4. 【システム設定】 GP-Pro EX (Pro-face製表示器)	
5. 【作画】 GP-Pro EX (Pro-face製表示器)	
6. 【シミュレーション】 GP-Pro EX (Pro-face製表示器)	
7. 【プロジェクト転送】 GP-Pro EX (Pro-face製表示器)	
8. 動作確認しよう	

## 付録：さらに便利に使おう

I. SoMachine Basicのプログラミングツール	
(1) オペレーションツール	付-2
(2) 命令一覧	付-3
(3) スマートコーディング	付-5
(4) PID	付-7
(5) スケジュールブロック	付-8
(6) データロギング	付-9



メモ(以下は余白です。ご自由にお使いください。)

# 序章

## プログラマブルコントローラー 「Modicon M221シリーズ」

I . テキストのご紹介.....	序-2
II . Modicon M221シリーズ/SoMachine Basicのご紹介.....	序-4
1. Modicon M221シリーズの特徴	
2. プログラミングソフトウェア「SoMachine Basic」	





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、「SoMachine Basic」の基本的なラダープログラム作成方法を習得いただくためのテキストです。  
ラダープログラム作成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、Pro-face製表示器との連携方法や作画ソフト「GP-Pro EX」の作画方法もあわせて解説しています。



### (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います。各章ごとにテーマを設け、「SoMachine Basic」/「Modicon M221シリーズ」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：ラダープログラムを回路別に説明しています。ラダープログラムの作成手順や作成例、注意点の説明をしています。

やってみよう：実習を通じて、ラダープログラムの設定方法を習得いただけます。



#### もっと詳しく知りたいときは

「SoMachine Basic」のTOP画面からアクセスできるビデオチュートリアル(動画)で使用方法を学ぶことができます。

SoMachine Basic 1.5 SP1 build 60148

なぜ登録が必要?



## (3)テキストの構成

本テキストでは、各章ごとに用途別のプログラムを作成します。

### 序章：プログラマブルコントローラ「Modicon M221シリーズ」

「M221シリーズ」の特徴やメリット、採用事例を紹介しています。



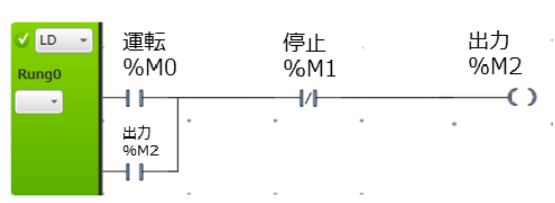
### 第1章：プログラミングツール「SoMachine Basic」の基礎知識

「SoMachine Basic」を使った「M221シリーズ」のプログラミング作成方法の基礎知識を説明しています。



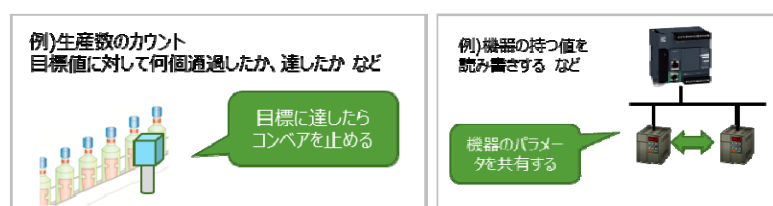
### 第2章：基本回路をマスターしよう

「SoMachine Basic」を使った基本回路の作成方法を習得できます。



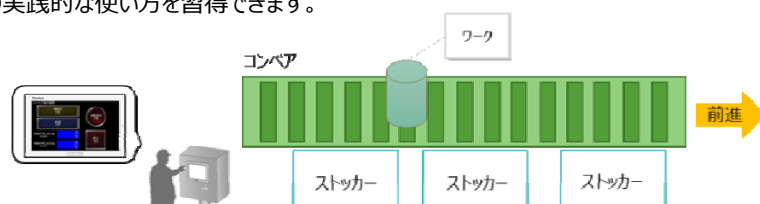
### 第3章：応用回路をマスターしよう

「SoMachine Basic」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らう

ライン操作盤の作成を通じて、「SoMachine Basic」の実践的な使い方を習得できます。



### 付録：さらに便利に使おう

Modicon M221シリーズ/SoMachine Basicのさらに便利な使い方や機能を紹介しています。

## Ⅱ. Modicon M221シリーズ/SoMachine Basicのご紹介

### 1. Modicon M221シリーズの特徴

#### (1) Modicon M221シリーズとは

M221は、CPUやI/Oなどを搭載したオールインワンタイプでコストメリットの高いシンプルな小型のPLCです。ラインナップは、コンパクトタイプとブックタイプがあり、いずれも、アナログ入力を標準で2チャンネル 装備しており、コストダウンやオプションユニットの後付けの手間を削減できます。

コンパクトタイプは扱い易いビルト型です。M221専用オプションの拡張カートリッジを本体に組み込むことで、外形を変えずに簡単にアナログなどの増設もできます。

ブックタイプは、幅70mmと圧倒的に小さく、狭いスペースでも設置可能です。プレス機やかくはん装置等のIO点数が少な目の装置に最適です。

【シンプル】  
シンプルなオール  
インワンの構成

【アナログI/F】  
アナログI/Oを多  
く使用する工作  
機械などに最適

【安全機能】  
セーフティ拡張  
モジュールで、簡単  
にSIL2/SIL3の  
安全レベルを実現

【無償】  
プログラミングツ  
ール「SoMachine  
Basic」は直感的  
な操作でプログラ  
ム作成が可能

CHECK!

#### Modicon M221シリーズのラインナップ

##### 【コンパクトタイプ(全18種)】

コンパクトタイプは扱い易いビルト型で、組み込み入出力点数が16点、24点、40点とあり、出力タイプはリレー、シンク、ソースから選択できます。また、本体組み込みカートリッジに対応しています。

16点

24点

40点



##### 【ブックタイプ(全10種)】

ブックタイプは、幅70mmと圧倒的に小さいので狭いスペースでも設置可能です。組み込み入出力点数は、16点と32点、出力はリレー、ソースから選択できます。

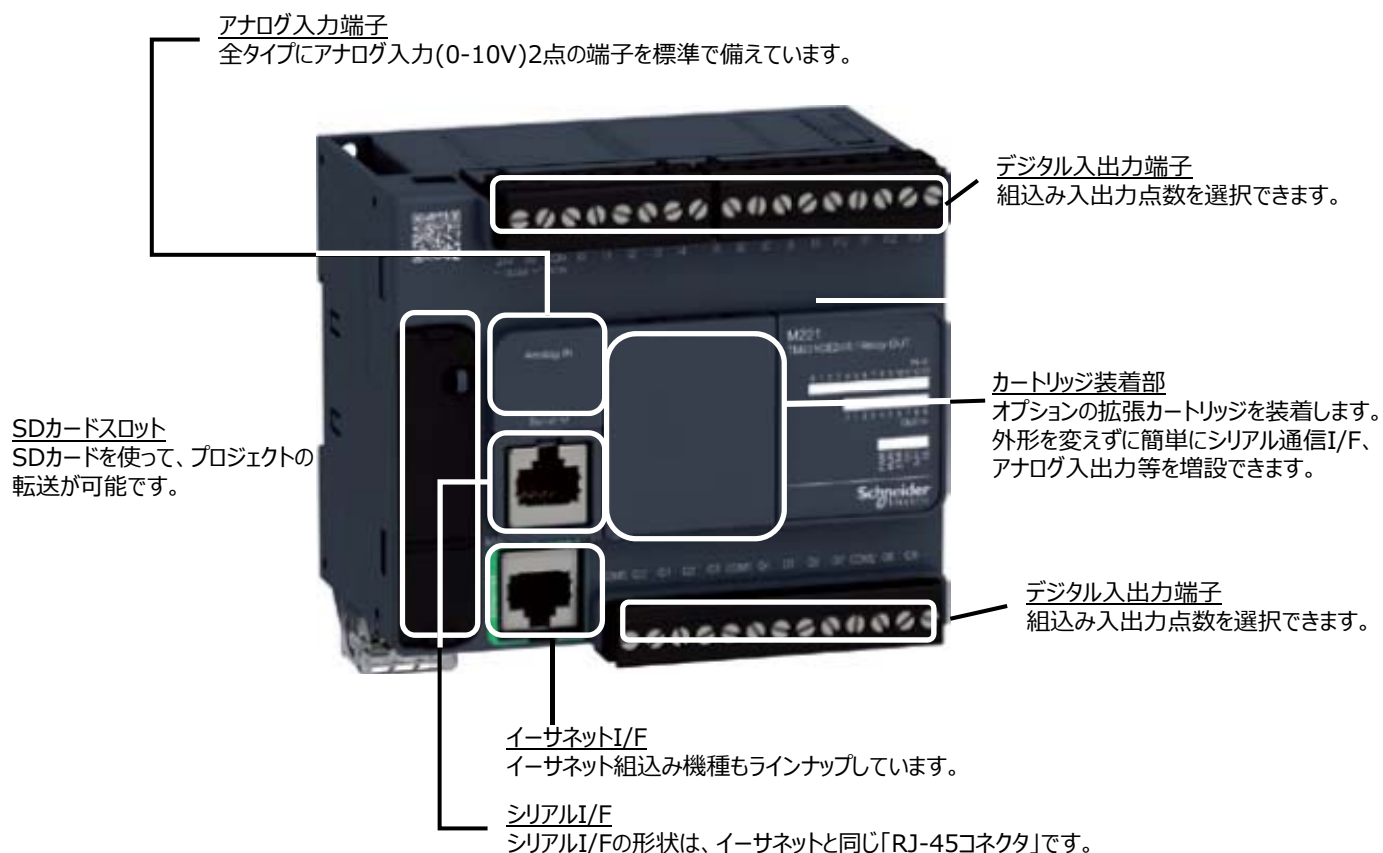
16点

32点



## (2)Modicon M221シリーズのインターフェイス

M221にはさまざまなインターフェイスを備えており、拡張性に優れております。

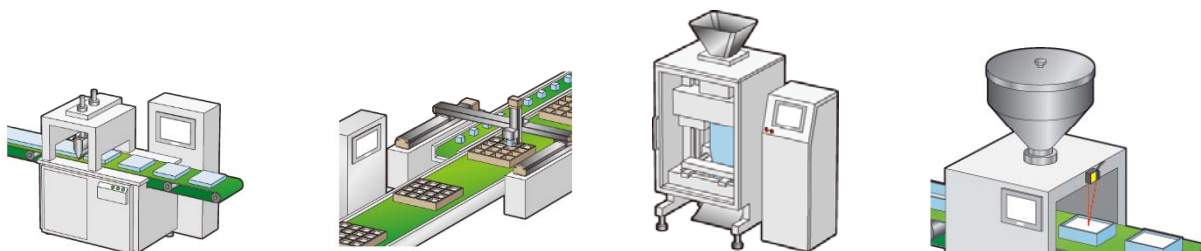


### M221シリーズ コンパクトタイプ(TM221CE24)



#### Modicon M221シリーズの採用事例

M221は、プレス機やかくはん装置、スライサー、検査装置等の、入出力点数の少な目の機械装置に最適です。  
スプリンクラーのようにFA用途だけに限らず農業等いろいろなところで採用いただけます。  
使用環境温度は、-10℃の低温度環境から55℃の間での使用が可能です。  
冷凍食品を扱う現場や暗所等の環境の厳しいところでも安心してお使いいただくことができます。





## 2. プログラミングソフトウェア「SoMachine Basic」

### (1) SoMachine Basicとは

SoMachine Basicは、M221専用プログラミングソフトウェアです。

【SoMachine Basicの特徴】

HPより無償でダウンロード

直感的な操作でプログラム作成が可能

ビデオチュートリアル(動画)で使用方法の学習可能！

IEC 61131-3に準拠したプログラム言語(LD、IL、Grafcet)

Windows 7, 8, 8.1, 10対応



機種選択

プログラミング

デバッグ



メンテナンス画面や  
シミュレーションで  
簡単デバッグ

CHECK!

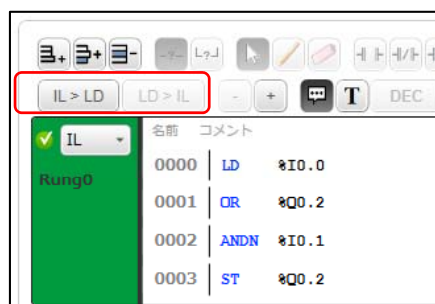
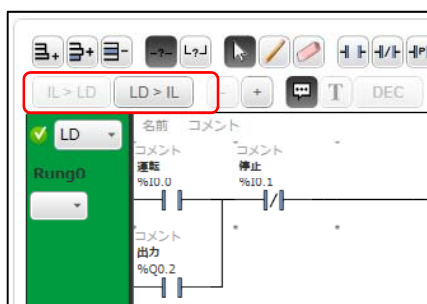
#### PLCプログラミング言語の グローバルスタンダード「IEC 61131-3」

「IEC 61131-3」とは、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。

この規格では、日本で普及しているラダーダイアグラムを含む5種類の言語が規定されており、使用者のスキルや目的に合った言語が使うことができます。

SoMachine Basicは、「IEC 61131-3」に準拠していることにより、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)という特長があります。ラダー方式だけでなく、IL(Instruction List)やSFC(Sequential Function Chart)にも対応しています。

SoMachine Basicでは、LDとILを1クリックで切り替えることができます。





メモ(以下は余白です。ご自由にお使いください。)

# 第1章

## プログラミングツール

## 「SoMachine Basic」の基礎知識

I. 開発環境を整えよう.....	1-2
1. 開発環境	
II. Modicon M221シリーズとPro-face製表示器.....	1-3
1. Pro-face製表示器との接続	
2. [Schneider Electric SA] Modbusシリアル接続について	
3. [Schneider Electric SA] Modbusイーサネット接続について	
III. SoMachine Basicの基礎知識.....	1-8
1. プロジェクトの作成方法	
2. SoMachine Basicの画面構成	
3. 新規プロジェクトの作り方	
IV. まずは、これだけ！基本操作！.....	1-18
1. プログラミングの基本操作	
2. ラングの追加	
3. 命令の割付け	
4. アドレスの割付け	
5. デバッグ	



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

M221の開発には、下記の4つをご用意ください。



SoMachine Basic



Modicon  
M221シリーズ



転送用ケーブル



Pro-face製表示器

#### (1) SoMachine Basic

Modicon M221専用のプログラミングソフトウェアです。直感的な操作で初心者でも安心してお使いいただくことができます。ホームページから無償でダウンロード可能です。

[https://www.proface.com/ja/product/schneider-plc/Modicon\\_M221/SoMachine](https://www.proface.com/ja/product/schneider-plc/Modicon_M221/SoMachine)

#### (2) Modicon M221シリーズ

Modicon M221は、CPUや入出力などを一体化したオールインワンタイプでコストメリットの高いシンプルな小型のPLCです。ラインナップは、コンパクトタイプとブックタイプがあり、いずれも、アナログ入力を標準で2チャンネル 装備しており、コストダウンやオプションユニットの後付けの手間を削減できます。

M221の詳細は弊社HPにてご確認ください。 [https://www.proface.com/ja/product/schneider-plc/Modicon\\_M221/top](https://www.proface.com/ja/product/schneider-plc/Modicon_M221/top)

#### (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-MiniB)

市販のUSBケーブル(Type A to mini 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。



SDカード

現場にパソコンや専用ツールを持ち込まなくても、SDカードでプロジェクトの更新ができます。

#### (4) Pro-face製表示器

GP4000シリーズ/SP5000シリーズ等のPro-face製表示器と接続することができます。



GP4000シリーズ  
/SP5000は全機種  
に標準でイーサネット  
のI/Fが付いています。

※GP-4201TWを  
除く



Pro-face製表示器の詳細は弊社HPにてご確認ください。

<https://www.proface.com/ja/product/hmi/top>



## II. Modicon M221シリーズとPro-face製表示器

### 1. Pro-face製表示器との接続

#### (1)「Schneider Electric SA Modbus」を使った接続

M221とPro-face製表示器とは、[Schneider Electric SA] Modbus方式を使って接続します。

ModbusとはModicon(現シュナイダーエレクトリック)が開発したFA機器間の通信プロトコルです。仕様は一般に公開されており、汎用的に採用されています。

[Schneider Electric SA] Modbusプロトコルは、(Modbus)シリアル接続と(Modbus TCP)イーサネット接続があり、Pro-face製表示器とはどちらを使っても接続できます。

##### 【[Schneider Electric SA] Modbusシリアル接続】

M221のシリアルI/Fの形状は、イーサネットと同じ「RJ-45コネクタ」です。



M221コンパクトタイプ(TM221CE16R)

シリアル通信



GP-4115T(PFXGP4115T2D)

※インターフェイス：RS-232C

GP-4116T(PFXGP4116T2D)

※インターフェイス：RS-422/485

##### 【[Schneider Electric SA] Modbusイーサネット接続】

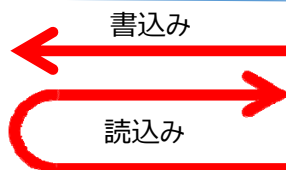
※本テキストでは、Pro-face製表示器をマスター(クライアント)、M221をスレーブ(サーバー)として説明します。

スレーブ(サーバー)



M221コンパクトタイプ(TM221CE16R)

イーサネット通信



マスター(クライアント)



GP-4114T(PFXGP4114T2D)

※インターフェイス：イーサネット

※イーサネット接続できるのは、M221のイーサネット組込み機種のみです。

CHECK!

#### Modicon M221シリーズと相性がぴったり「GP4100シリーズ(コンパクト)」

GP4100シリーズとは、4.3型の超小型表示器です。

GP4116とGP4115とGP4114の3種類があり、それぞれ、RS422/485、RS232C、イーサネットに対応しています。

お求めやすい価格になっているため、コストがネックで表示器の採用を見送っていた小型装置への組み込みに最適です。



## 2. [Schneider Electric SA] Modbusシリアル接続シリアル接続について

### (1)接続方法

①M221と表示器をシリアルケーブルで接続します。

※M221のシリアルI/Fの形状は、イーサネットと同じ「RJ-45コネクタ」です。



※ Pro-face製表示器とのシリアル接続の詳細は、GP-Pro EX機器接続マニュアルの[Schneider Electric SA](MODBUS SIO マスタ)をご参照ください。

### (2)SoMachine Basicの通信設定

SoMachine Basicの通信設定は、「設定タブ」で行います。

①「設定タブ」-「SL1(シリアルライン)」をクリックし、下記の設定を行います。



②「Modbus」の設定も行います。

※HMIをマスターにする場合は、M221をスレーブとして設定します。

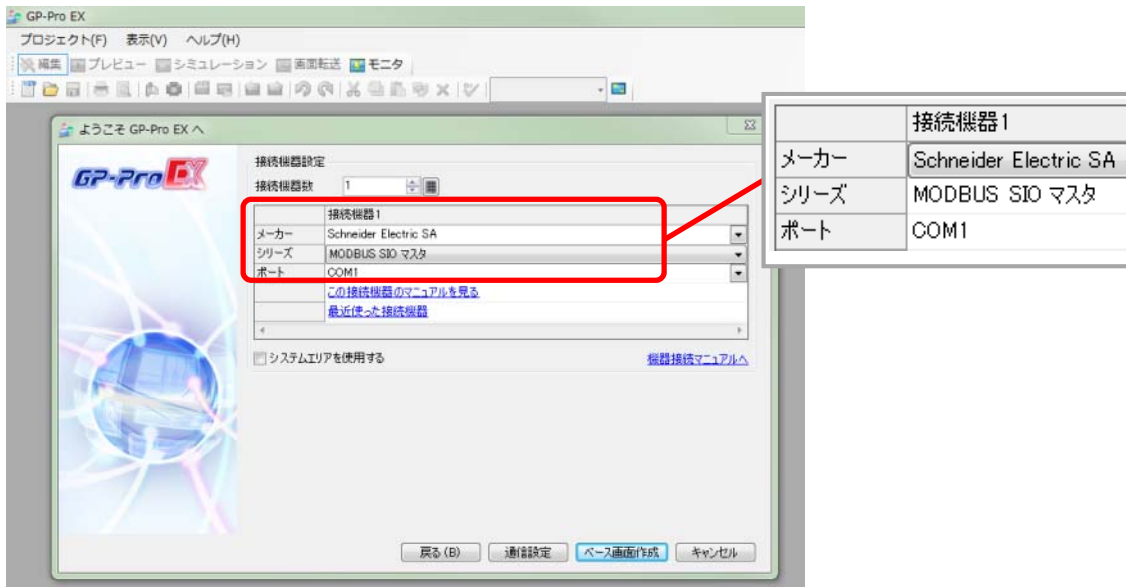


## (3)GP-Pro EXの通信設定

GP-Pro EXでは、「接続機器の指定」「通信設定」の2つを行います。

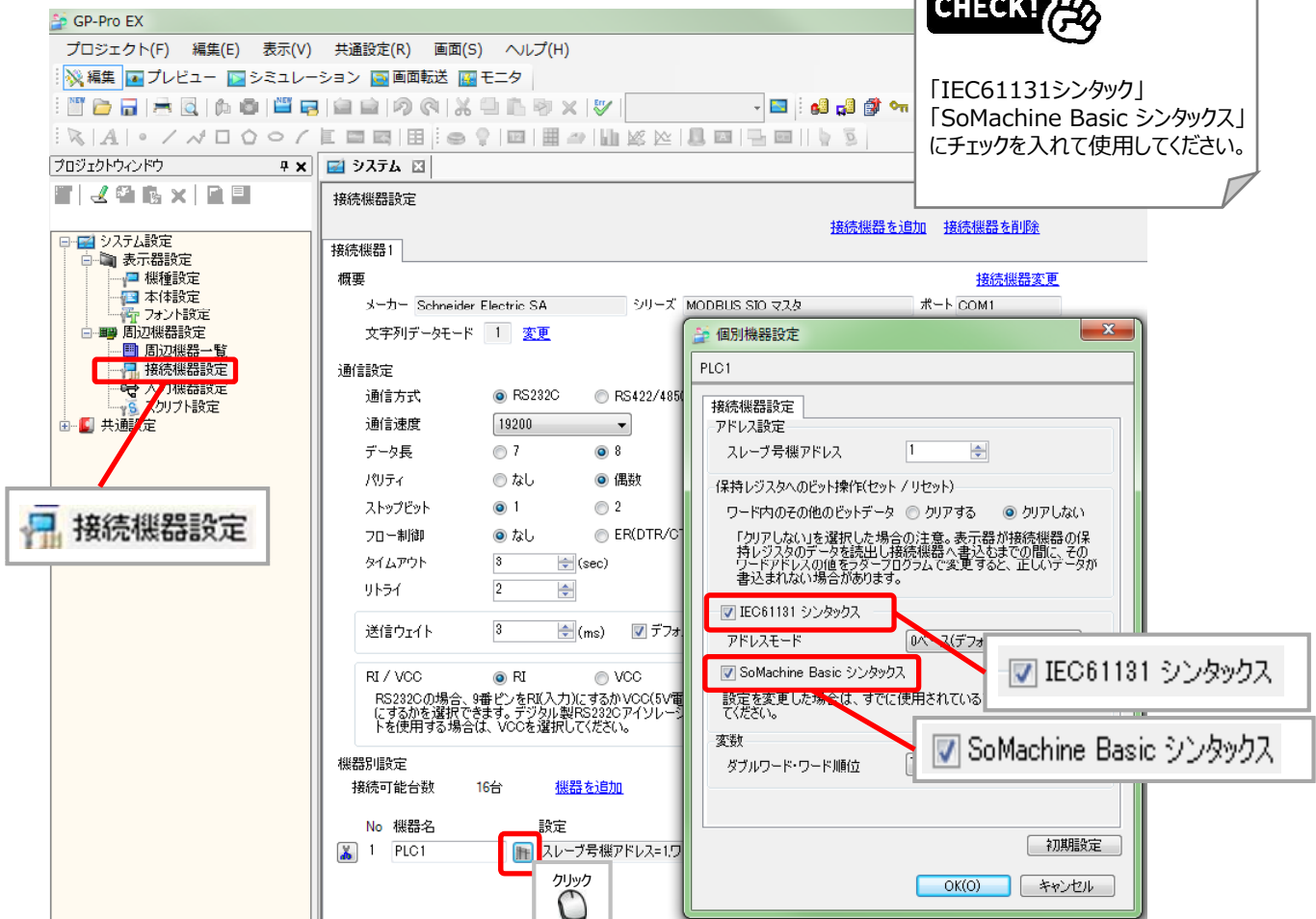
### ①接続機器の指定(通信プロトコル)

GP-Pro EX起動時の接続機器設定で下記の設定を行います。



### ②通信設定

メニューバー「プロジェクト」「システム設定」「接続機器設定」で下記の設定を行います。



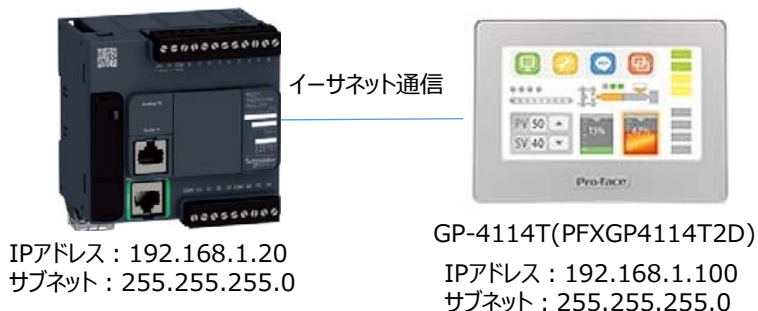
※「IEC61131シンタックス」「SoMachine Basic シンタックス」の詳細は、1-24を参照してください。

### 3. [Schneider Electric SA] Modbusイーサネット接続について

#### (1)接続方法

- ①Pro-face製表示器のIPアドレスを設定します。
- ②LANケーブルで接続します。

【接続例】



※ Pro-face製表示器とのイーサネット接続の詳細は、GP-Pro EX機器接続マニュアルの[Schneider Electric SA](MODBUS TCP マスタ)をご参照ください。

#### (2)SoMachine Basicの通信設定

SoMachine Basicの通信設定は、「設定タブ」で行います。

- ①M221本体のIPアドレスの指定

「設定タブ」-「ETH1」をクリックし、下記の設定を行います。

新編プロジェクト\* Schneider Electric SoMachine

M221 Controller (USB) プログラムエラーを検出しました

プロパティ **設定** プログラミング 表示

メッセージ

MyController (TM221CE24R)

デジタル入力

デジタル出力

アナログ入力

高速カウンタ (HSC)

IO Bus

**ETH1**

Modbus TCP

EtherNet/IP アダプター

SL1 (シリアルライン)

Modbus

Ethernet

デバイス名 M221

☐ DHCP による IP アドレス

☐ BOOTP による IP アドレス

☒ 固定 IP アドレス

IP アドレス 192 . 168 . 1 . 20

サブネットマスク 255 . 255 . 255 . 0

ゲートウェイアドレス 0 . 0 . 0 . 0

※PLCのIPアドレスを入力します。

セキュリティパラメーター

☒ プログラミングプロトコルを有効

☒ Ethernet/IP プロトコルを有効

☒ Modbus サーバーを有効

☒ 自動検出プロトコルを有効

☒ Modbus サーバーを有効





## Ⅲ. SoMachine Basicの基礎知識

### 1. プロジェクトの作成方法

#### (1)プロジェクト(\*.smbp)とは

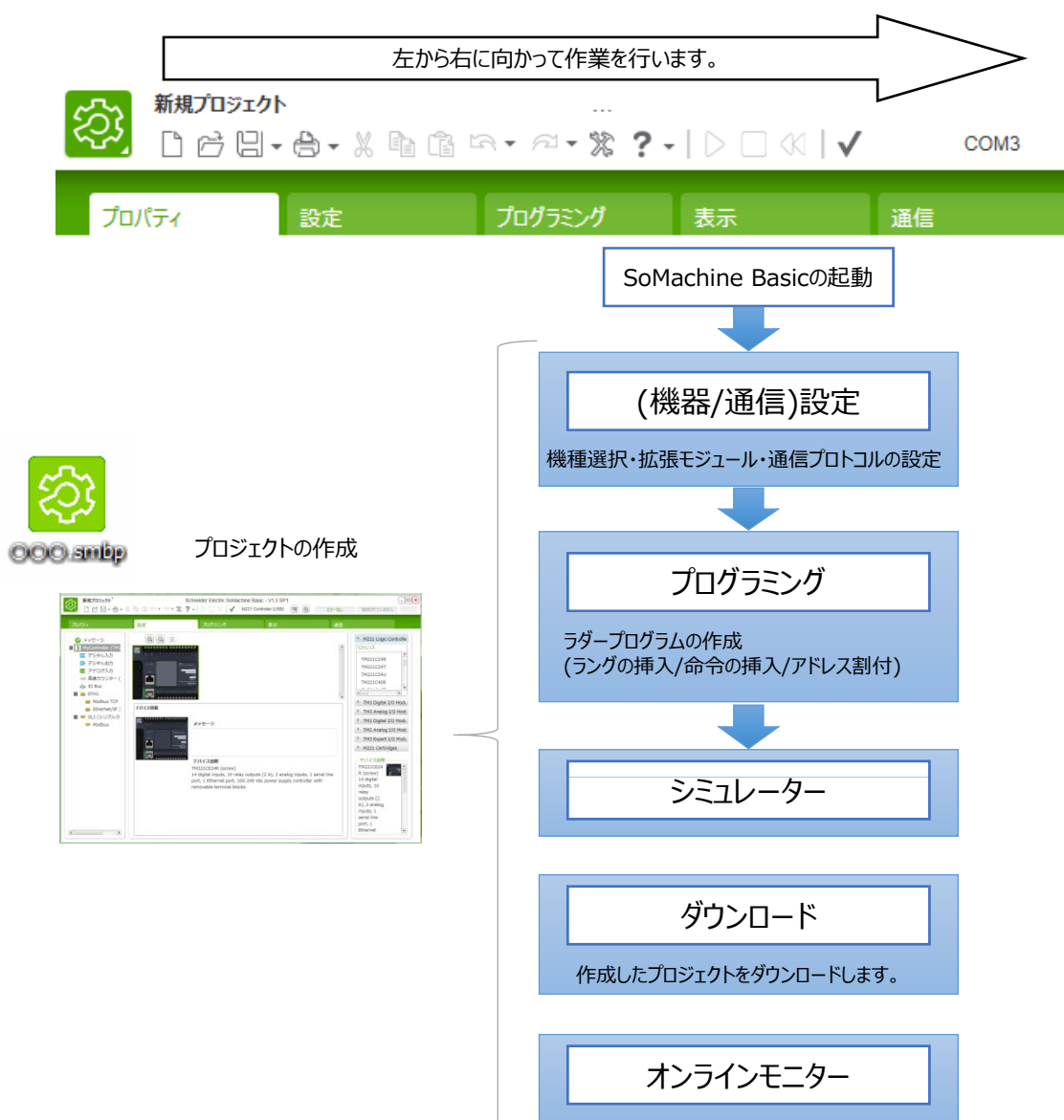
プロジェクトとは「プログラム」「通信プロトコル」「システム」を一つのまとめたものです。  
プロジェクトはパソコン上に「.smbp」の拡張子で保存されます。

SoMachine Basicで作ったプロジェクトをM221に転送することで、  
M221はシーケンス制御やI/O入出力を行ないます。



#### (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、「SoMachine Basic」の起動後、上部のタブに沿って設定を行います。  
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、M221に転送します。



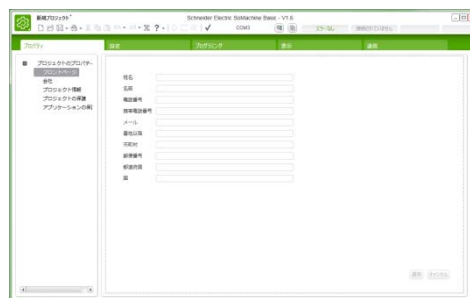
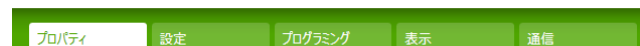
## 2. SoMachine Basicの画面構成

### (1) SoMachine Basicの5つ画面構成

SoMachine Basicは、5つの画面で構成されています。

#### ①プロパティ

プロジェクトに関する情報の登録とパスワード保護を指定できます。



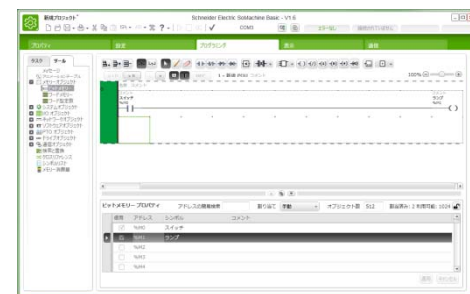
#### ②設定

使用するM221の指定、拡張モジュール等のハードウェアの設定を行います。  
また、Pro-face製表示器と接続するためのModbus通信の設定を行います。



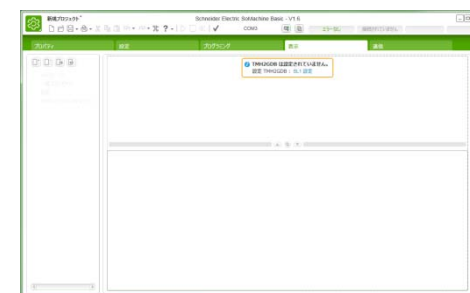
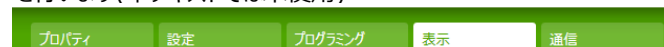
#### ③プログラミング

ラダー言語を使ってプログラミングを行います。



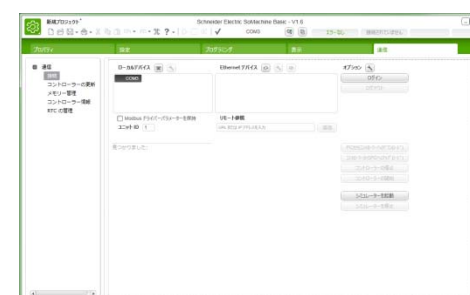
#### ④表示

オプションのシンプルディスプレイユニット(TM2H2GDB)を使用する場合の設定を行います(本テキストでは未使用)



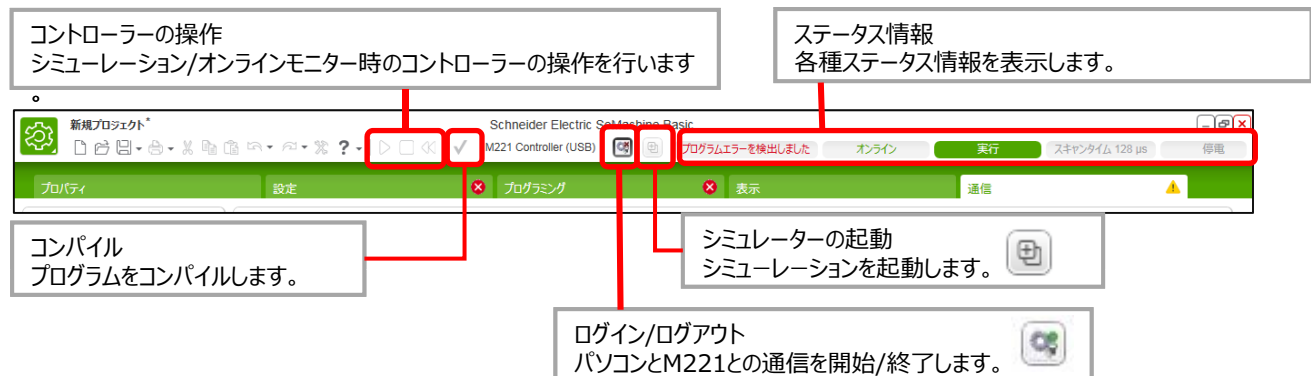
#### ⑤通信

PLCへのログインを行い、シミュレーター、プロジェクトの転送、オンラインモニター等を行うことができます。



## (2)ステータスエリアの構成

メインウィンドウ上部のステータスエリアには、現在のシステムステータスの情報が表示されます。また、よく使う機能をアイコン化し、ワンクリックで使用することができます。



## (3)設定タブの画面構成

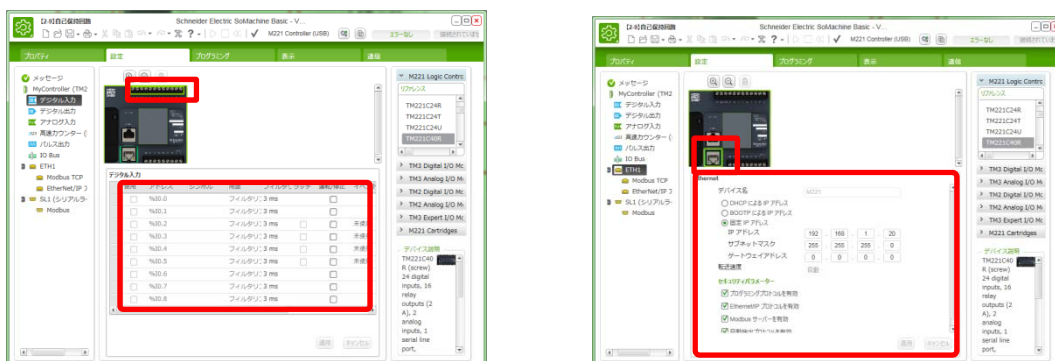
使用するM221の指定、拡張モジュール等のハードウェアの設定を行います。また、Pro-face製表示器と接続するためのModbus通信の設定を行います。



**CHECK!**

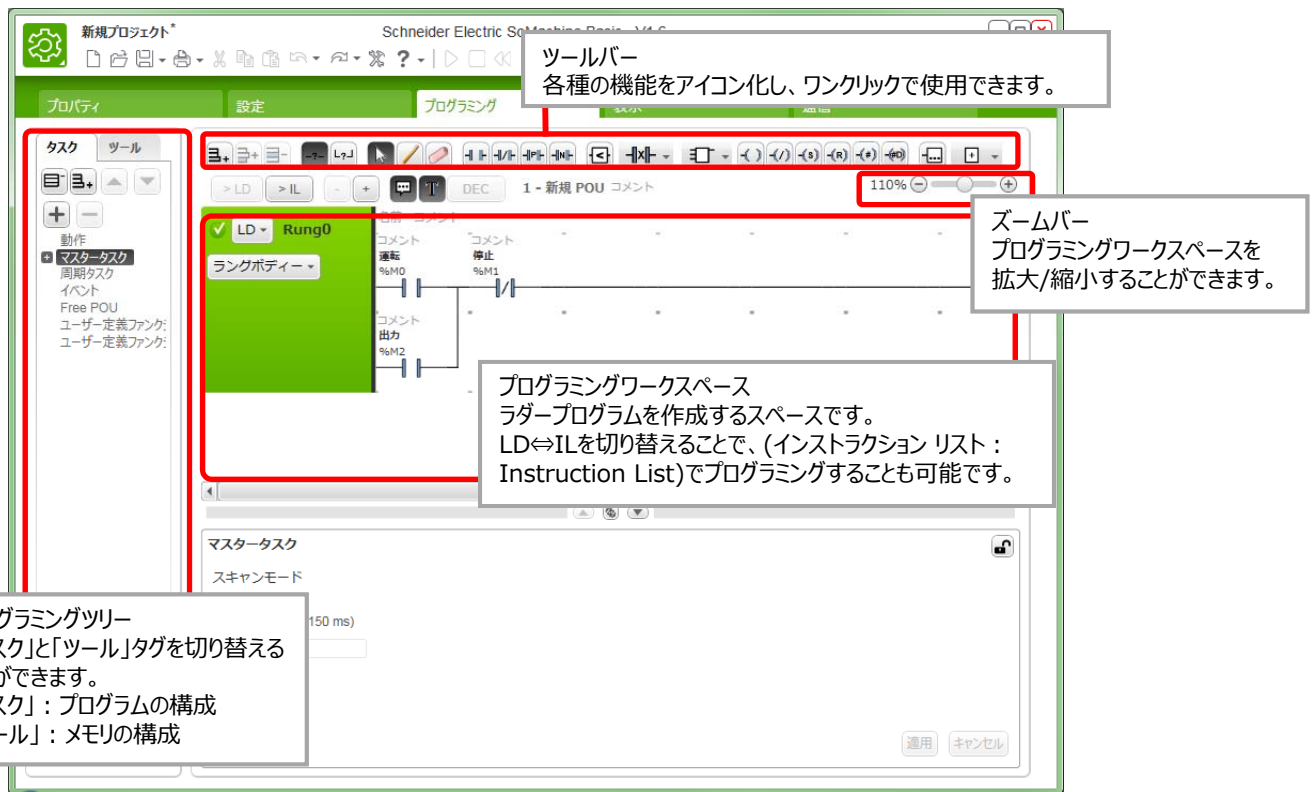
### 構成図の便利な使い方

構成図のI/Fや拡張モジュールの部分をダブルクリックすると、画面下部にそれぞれの情報が表示されます。



## (4)プログラミングタブの画面構成

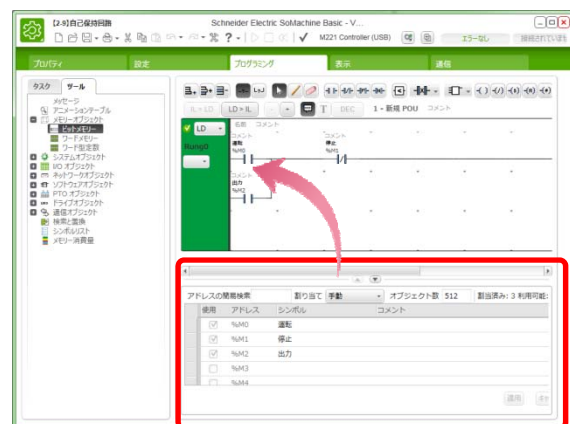
ラダー言語を使ってプログラミングを行います。



CHECK!

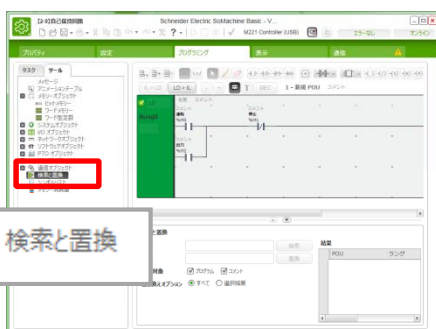
### プログラミングツール(ツール)

「プログラミングツール」を「ツール」にすると、各種アドレスの使用/未使用を画面下部で確認することができます。また、プログラミングワークスペースヘドラッグ & ドロップすることで、アドレスを割付けることができます。



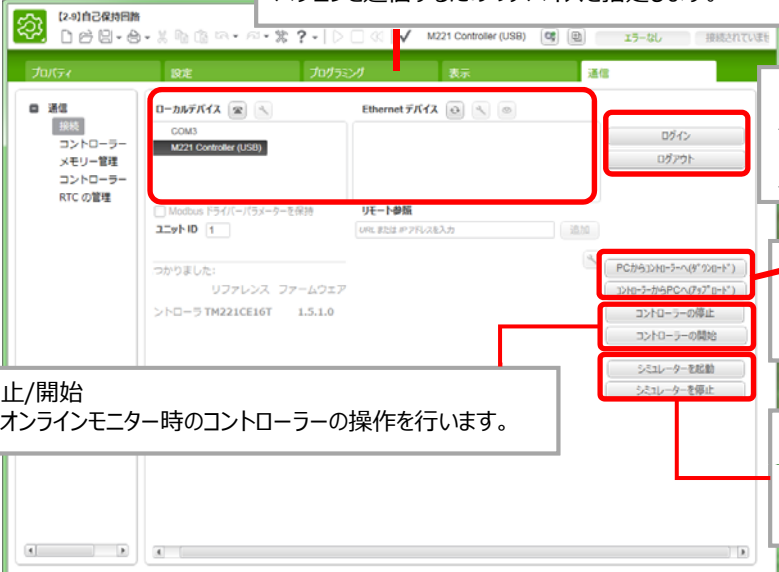
「検索と置換」では、プログラム/コメントを検索し、指定したものに一括変換ができます。

メモリ消費量では、メモリの使用状況を確認することができます。







## (5)通信タブの画面構成

PLCへログインを行うことができ、プロジェクトの転送、オンラインモニター等を行うことができます。



The screenshot shows the '通信' (Communication) tab with the following elements highlighted by red boxes and callouts:

- デバイス (Device):** A callout box pointing to the 'ローカルデバイス' (Local Device) section, stating: 'デバイス  
パソコンと通信するためのデバイスを指定します。' (Specify the device for communication with the PC).
- ログイン/ログアウト (Login/Logout):** A callout box pointing to the 'ログイン' (Login) and 'ログアウト' (Logout) buttons, stating: 'ログイン/ログアウト  
パソコンとM221との通信を開始/終了  
します。  
ステータスエリアの  でも可能です。' (Login/Logout starts/ends communication between the PC and M221. It is also possible in the Status Area using the .
- アップロード/ダウンロード (Upload/Download):** A callout box pointing to the 'PCからシミュレータへダウンロード' (Download from PC to Simulator) and 'シミュレータからPCへアップロード' (Upload from Simulator to PC) buttons, stating: 'アップロード/ダウンロード  
ログイン後、プロジェクトのダウンロード  
アップロードができます。' (Upload/Download: After login, you can download/upload the project).
- コントローラーの停止/開始 (Controller Stop/Start):** A callout box pointing to the 'コントローラーの停止' (Stop Controller) and 'コントローラーの開始' (Start Controller) buttons, stating: 'コントローラーの停止/開始  
シミュレーション/オンラインモニター時のコントローラーの操作を行います。' (Controller Stop/Start: Perform controller operations during simulation/online monitoring).
- シミュレーターの起動/終了 (Simulator Start/End):** A callout box pointing to the 'シミュレータを起動' (Start Simulator) and 'シミュレータを停止' (Stop Simulator) buttons, stating: 'シミュレーターの起動/終了  
シミュレータを起動/終了します。  
ステータスエリアの  でも可能です。' (Simulator Start/End: Start/stop the simulator. It is also possible in the Status Area using the .



メモ(以下は余白です。ご自由にお使いください。)

### 3. 新規プロジェクトの作り方

新規プロジェクトは、設定(PLCの指定・拡張モジュール・通信プロトコル)を行い、「プログラミング」「通信」の順に各種の設定を進めます。

ここでは、以下の環境を例にシステム設定を行います。

- ・PLC : Modicon M221(TM221CE16R)
- ・通信プロトコル : [Schneider Electric SA]Modbus TCP

スレーブ(サーバー)

マスター(クライアント)



IPアドレス : 192.168.1.20  
サブネット : 255.255.255.0

イーサネット通信



IPアドレス : 192.168.1.100  
サブネット : 255.255.255.0

#### (1) SoMachine Basicの起動

- ①デスクトップのSoMachine Basicのアイコンをダブルクリックします。



- ②新規プロジェクトの作成をクリックします。



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) ハードウェアの設定

- ①「設定」をクリックします。

【コントローラーの指定】

- ②右端の一覧から、TM221CE16Rをドラッグします。

コントローラー : TM221CE16R



- ③「確認」ウィンドウが表示されたら、「はい」をクリックします。



## 第1章

### 【IPアドレスの設定】

④左側から「ETH1」をクリックします。

⑤下記のようにM221自身のIPアドレスを指定します。

IPアドレス : 192.168.1.20  
サブネットマスク : 255.255.255.0

⑥「Modbusサーバーを有効」にチェックを入れます。

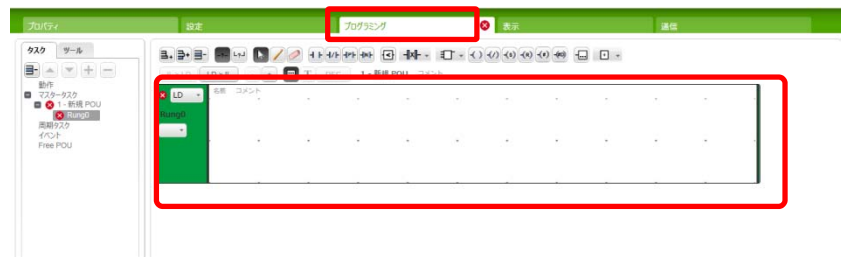
⑦「適用」をクリックします。



## (3)プログラミング


①「プログラミング」をクリックします。


②プログラミングワークスペースでプログラミングを行います。



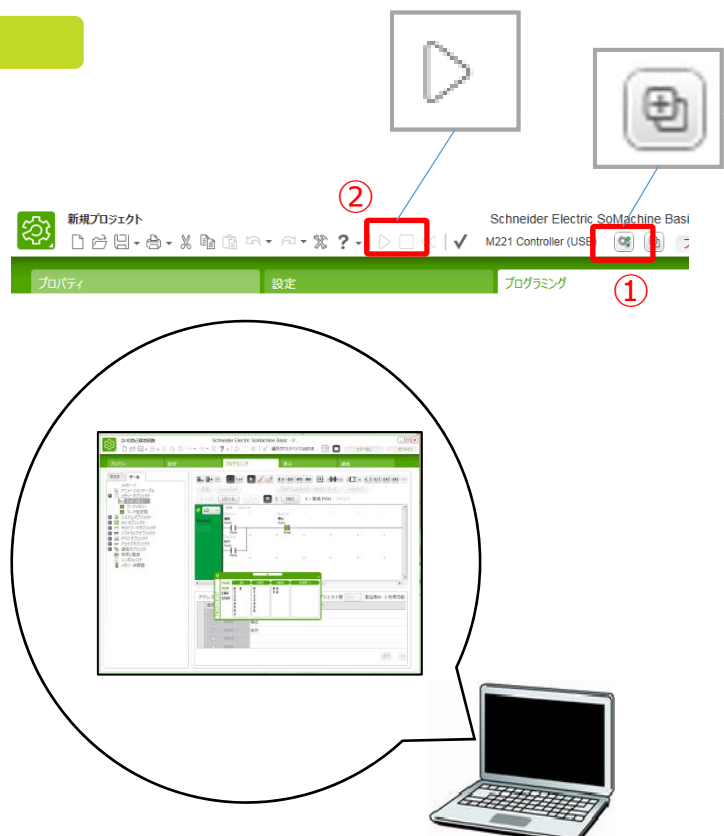
※プログラミングの詳細は、第2章以降を参照してください。

## (4)シミュレーター

①ステータスエリアの  をクリックします。  
(シミュレーターモードに切り替わります)

②ステータスエリアの  をクリックします。  
(シミュレーターが起動します)

③シミュレーターを使い、パソコン上で動作確認します。



※シミュレーターの詳細は、1-27を参照してください。



## (5)プロジェクトのダウンロード

ここでは、LANケーブルを使ったダウンロード方法を解説します。  
※あらかじめ、M221、パソコンにIPアドレスを指定しておきます。

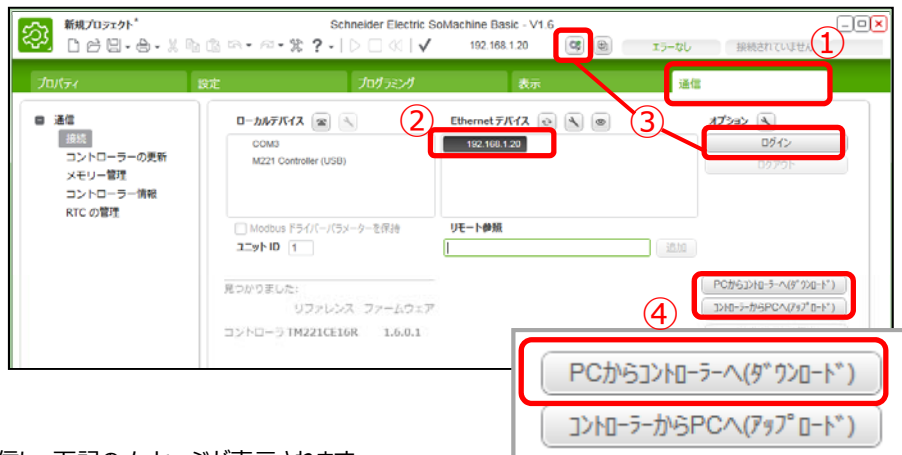


①「通信」をクリックします。

②「192.168.1.20」をクリックします。  
※M221とPCを接続するデバイス（方法）を指定します。

USBケーブルで転送する場合は、  
M221controller (USB)をクリックします。

③「ログイン」をクリックします。



※ログインすれば、M221とパソコンが通信し、下記のメッセージが表示されます。

⚠ PC とコントローラーのアプリケーションが違います  
PC とコントローラーのアプリケーションを比較

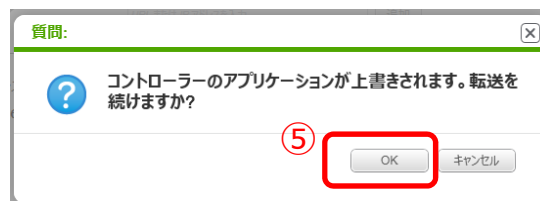
M221本体のラダープログラムとSoMachine Basicで設定しているラダープログラムに相違がある場合

✓ PC とコントローラー アプリケーションは同一です  
接続が確立されました

M221本体のラダープログラムとSoMachine Basicで設定しているラダープログラムが同じ場合

④「PCからコントローラへ」(ダウンロード)をクリックします。

⑤質問は「OK」をクリックします。



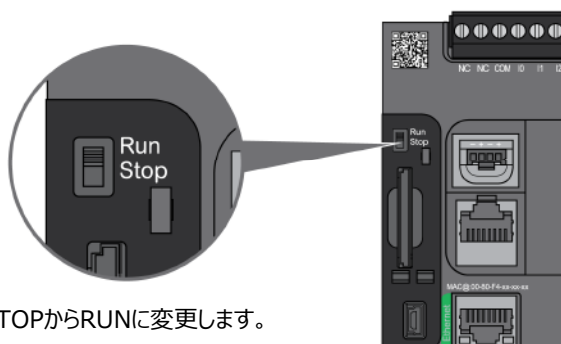
⑥正しく転送が完了すれば、下記のメッセージが表示されます。

✓ PC とコントローラー アプリケーションは同一です  
接続が確立されました



**M221本体でプログラムをRUNさせるには**

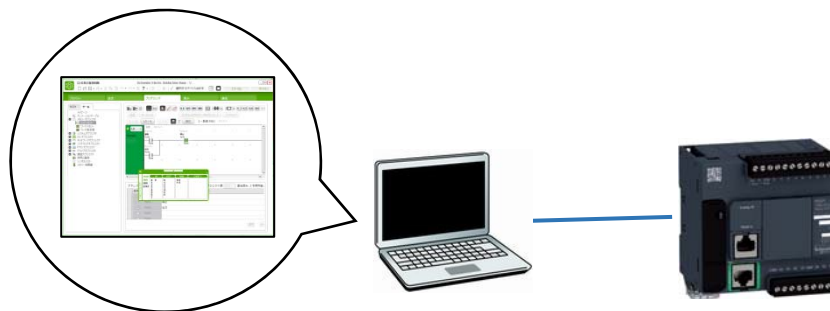
M221シリーズには運転/ 停止ハードウェアスイッチが備わっており、コントローラを運転または停止状態にすることができます。



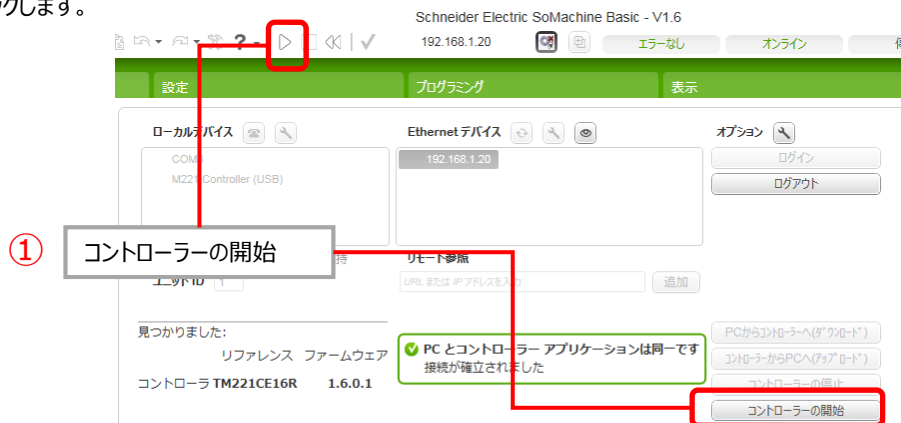


## (6)オンラインモニター

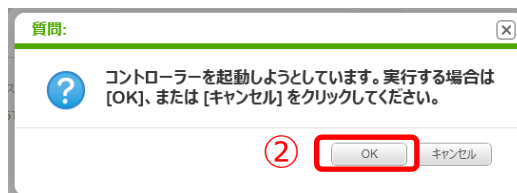
パソコン上で、M221のラダープログラムの動作を確認したい場合に使用します。



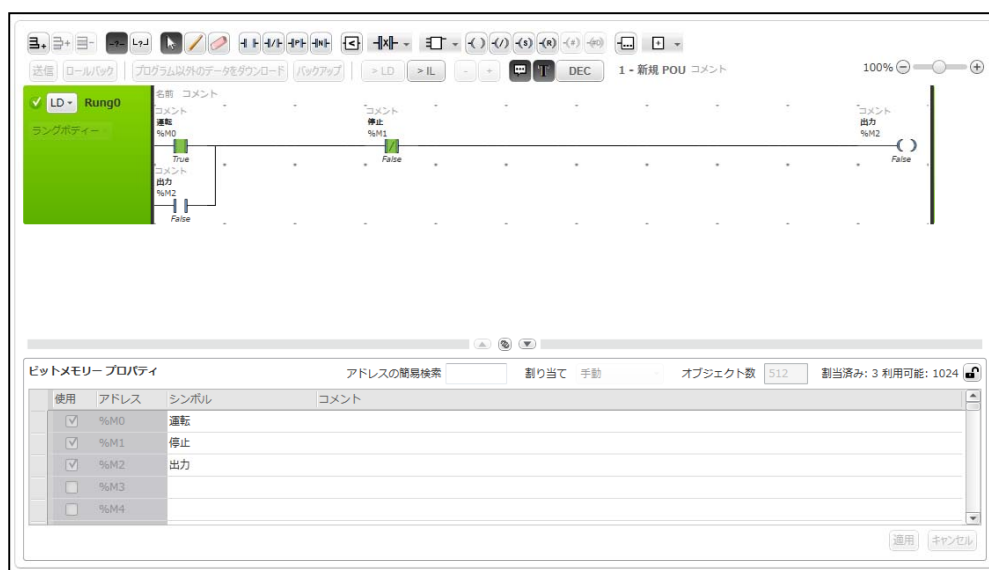
① (ログイン後に)「コントローラーの開始」をクリックします。



②「質問」は「OK」をクリックします。



③ オンラインモニターを使い、運転中のM221内のプログラムの動作確認を行います。



CHECK!


ちょっと知りたいときは、「Shift + F1」

SoMachine Basicでは、作業しているときに「Shift+F1キー」を押下すれば、いつでも見たい項目をコンテキストヘルプで確認することができます。

【ポップアップヘルプの表示方法】

① 不明点があれば、「Shift+F1キー」を押下



② マウスカーソルが  に変化



③ 知りたい箇所をクリック



④ 知りたい項目がポップアップで表示！

The screenshot shows the SoMachine Basic V1.6 SP2 interface. Three callout boxes highlight specific help features:

- 設定ウィンドウの概要 (Overview of the Settings Window):** A window showing the structure of the settings window, including sections for '設定' (Settings) and 'プログラム' (Program).
- 演算ブロック (Operation Block):** A window showing the '演算ブロック' (Operation Block) section, which provides context help for the selected block in the ladder logic diagram.
- メモリーオブジェクト (Memory Object):** A window showing the 'メモリーオブジェクト' (Memory Object) section, which provides context help for the selected memory object in the variable declaration table.

プロパティ	編集可能	値	初期値
使用	いいえ	True/False	False
アドレス	いいえ	ビットオブジェクトを参照してください。	N/A
シンボル	可	有効なシンボル	なし
値	可	ビットオブジェクトを参照してください。	0
コメント	可	任意のコメント	なし

※F1キーのみを押下すれば、ヘルプが表示されます。

## IV. まずは、これだけ！基本操作！

### 1. プログラミングの基本操作

#### (1) 代表的な基本操作

プログラミングでは、以下の順番に沿って行います。

- ① ラング(Rung)の追加・・・2本の垂直線の間にかかれた命令の集合体です。SoMachine Basicでは、ラング単位でプログラミングします。
- ② 命令の配置・・・・・・・・・・ ラング内に、命令/接点を配置します。
- ③ アドレスの割付け・・・・・・・・ アドレスを命令に割り付けます。

ラダープログラムでは、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(アドレスを変化させます)

また、接点では、割当てたアドレスを変化させることで、接点をON/OFFします。



#### 「タスク」「POU」「Rung」

SoMachine Basicのプログラムは、「タスク」「POU」「Rung」で構成されています。

タスク：タスクには下記の種類があります。各POUをタスクに登録することで実行されます。

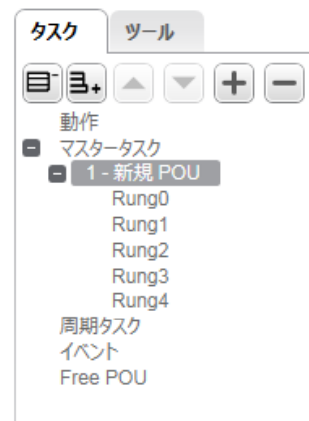
- ・マスタータスク：メインの処理のエリアです。  
タスク内のPOUに記述された内容を繰り返し処理を行います。
- ・周期タスク：サブルーチンを定期的に処理します。
- ・イベント：指定した条件を満たした場合にのみ実行されます。

POU (Program Organization Unit)

プログラムの実行単位です。POU内はRung単位で区分されています。

Rung(ラング)


論理式を構成する命令の集まり。

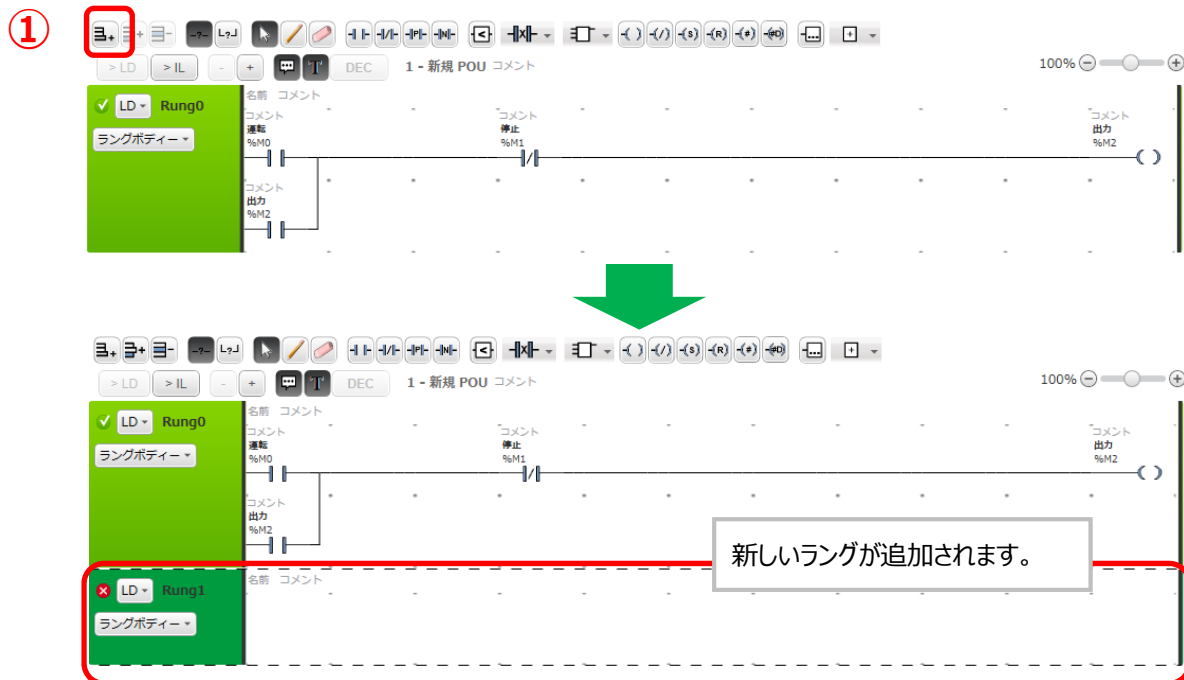


## 2. ラングの追加

### (1) 新規ラングの追加

新規ラングの追加方法を説明します。

① 新規ラングの追加  をクリックします。

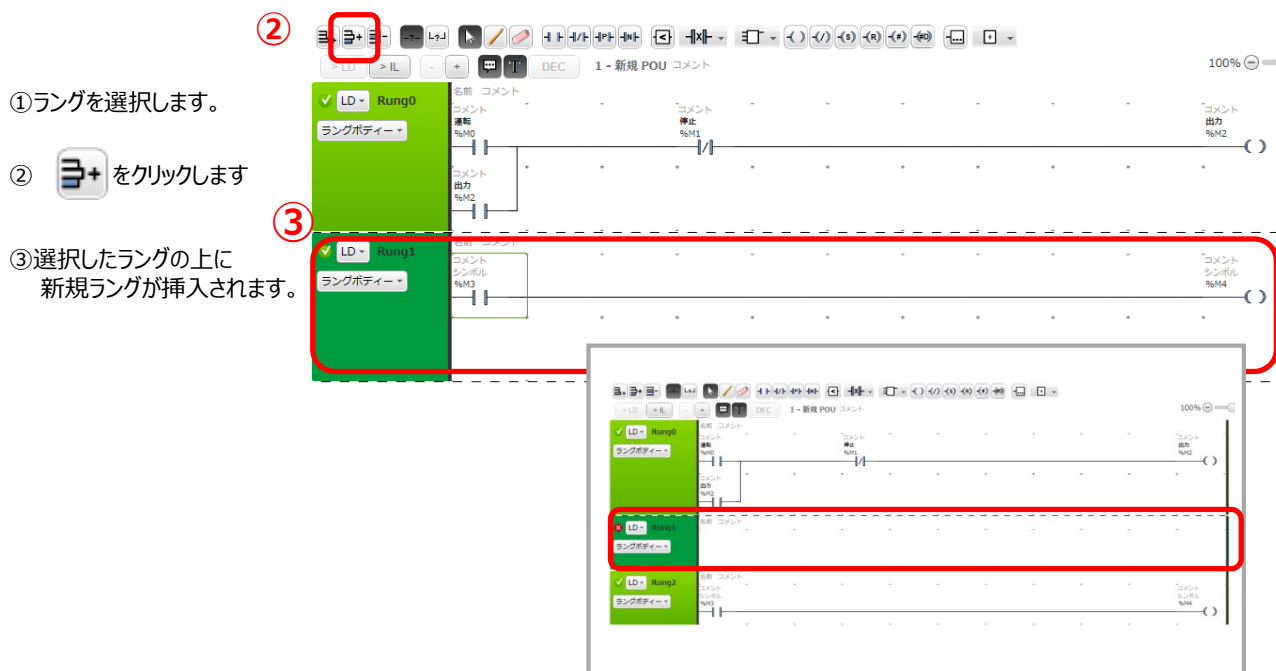


CHECK!

### ラングの挿入方法

既存のラング間に新たにラングを挿入したい場合は、新規ラングの挿入を  クリックします。

選択したラングの上に新たなラングが挿入されます。

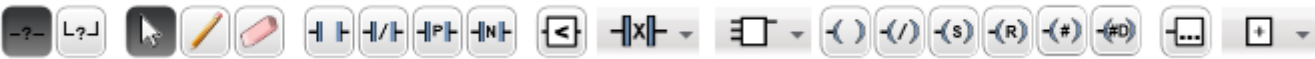


3. 命令の割付け

(1)命令とは

SoMachine Basic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。

各命令は、ツールバーにアイコン化されており、ラング内に配置します。配置後はアドレスを割付けます。



(2)代表的な命令


【代表的な命令】 下記以外にも様々な命令を用意しています。

カテゴリ		命令の呼称	図
基本命令	接点	a接点	
		b接点	
		立上がり接点	
		立下がり接点	
	コイル(出力)	コイル出力	
		反転コイル出力	
		セットコイル	
		リセットコイル	
比較	比較ブロック	比較ブロックを配置し、比較式を入力します。	
ファンクション	ファンクションブロック	タイマー	
		カウンター	
		その他のファンクションも用意しています。	
演算 (データ加工)	オペレーションブロック	オペレーションブロックを配置し、操作式を入力することで、算術演算、データ転送、インクリメント/デクリメント等のデータ加工ができます。	 スマートコーディングアシスタントも使用できます。

※命令の一覧は、「付-3」をご参照ください。

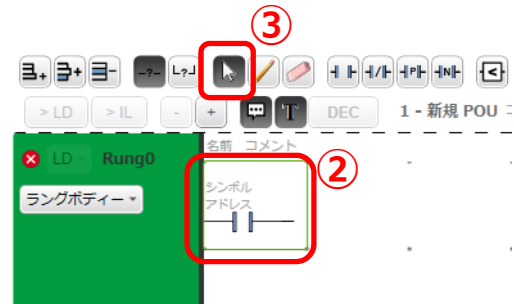
### (3)命令の配置方法


ラダープログラムに命令を配置し、それらにアドレスを割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  をクリックします。



② 命令を配置する箇所(ラング内)をクリックします。



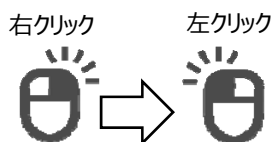
③ 命令ツールバーの  をクリックし、選択を解除します。



#### 便利な命令の解除方法

命令はアイコンをクリックすると、解除するまで、連続して配置することができます。  
すばやく選択を解除するには、下記の方法があります。

① マウスを右クリックすると命令の選択を解除できます。



※右クリックすると、解除できますが、同時にショートカットメニューが表示されますので、左クリックすると通常操作ができます。

② パソコンの「ESC」キーをタッチすると、命令の選択を解除できます。



#### ラダーエディターの環境設定

システム設定「ラダーエディター」を使えば、ラダーエディターの各種の変更ができます。


例：「ツール選択の保護」で「ポインターをリセット」を選択すると、命令を配置するごとに、通常操作に戻ります。



## (4)分岐の挿入


ラング内に分岐を挿入する方法について説明します。

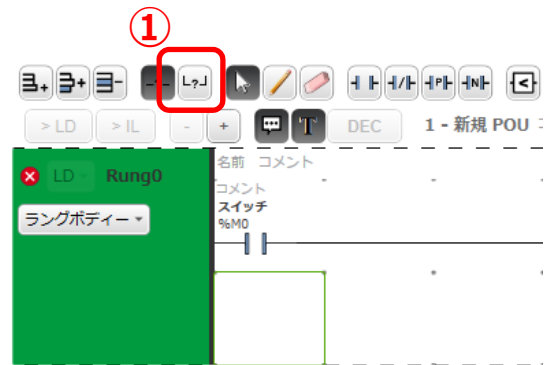
(分岐を挿入する前にあらかじめ命令を配置しておきます)

①  をクリックし、分岐モードにします。

② 命令ツールバーの  をクリックします。

③ ②で選択した命令を分岐したい箇所に配置します。

④ 分岐の挿入が完了すれば、 をクリックし、通常モード (分岐なし)に戻します。



**CHECK!** 

通常モードと分岐モードの切替えは、  
F2⇔Shift + F2  
でも行うことができます。


**CHECK!** 

### 便利な分岐の挿入方法

通常モード(分岐なし)では、「ラインの描画」 を使用すると分岐を挿入することができます。

① 命令ツールバーから  をクリックします。

② 分岐箇所をドラッグしてラインを描画します。

ラインを消去する場合は、 をクリックし、消去箇所をドラッグします。



## 4. アドレスの割付け

### (1) アドレスの種類

SoMachine Basicには、用途にあわせたアドレスが、オブジェクト単位で区分されています。各オブジェクトのアドレスは、「プログラミングツリー」「ツール」から、確認することができます。

また、アドレスには、あらかじめ機能を定義されたシステムオブジェクトがあります。

アドレスに任意の名前を付けてプログラム上で使用することができます。それぞれのアドレスに自由に名前が付けられることで、ラダー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。



### (2) 代表的なオブジェクトとアドレス

※下記以外にもオブジェクトとアドレスを用意しています。

オブジェクトの種類	種類	アドレス	内容
メモリオブジェクト	ビットメモリ	%M0～%M1023	ON/OFFを表す1ビットの長さのアドレス
	ワードメモリ	%MW0～%MW7999	16ビットの長さのアドレス
		%MD0～%MD7998	32ビットの長さのアドレス(ダブルワード)
		%MF0～%MW7998	32ビットの長さのアドレスで浮動小数点と0の値を持つ
システムオブジェクト	システムビット	%S0～%S159	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。機能の変更や削除したりするなどの編集はできません。
	システムワード	%SW0～%SW233	
I/Oオブジェクト	デジタル入力	%I0.0～%I0.※	デジタル入力用の1ビットの長さのアドレス
	デジタル出力	%Q0.0～%Q0.※	デジタル出力用の1ビットの長さのアドレス
	アナログ入力	%IW.0～%IW.1	アナログ入力用のアドレス
ソフトウェアオブジェクト	タイマー	%TM0～%TM254	タイマー命令に割付ける構造体アドレス
	カウンター	%C0～%C254	カウンター命令に割付ける構造体アドレス



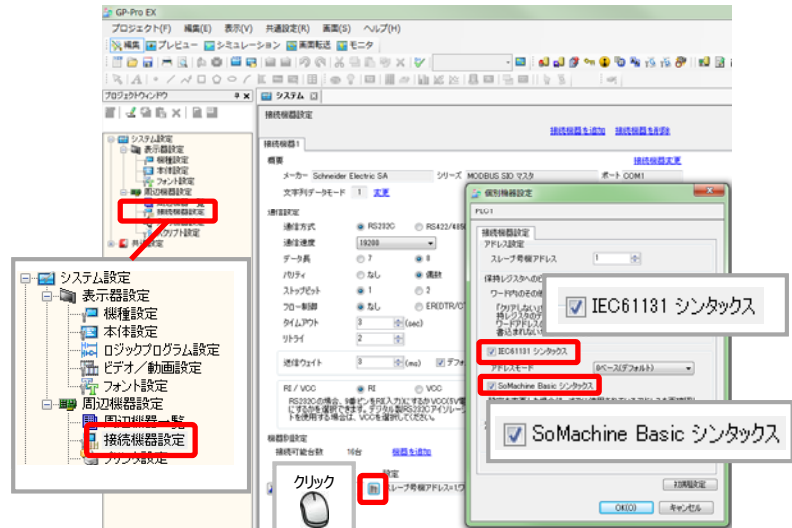
CHECK!

## 「IEC61131 シンタックス」「SoMachine Basic シンタックス」

M221とPro-face製表示器とは、  
[Schneider Electric SA] のModbus方式を  
使って接続します。

GP-Pro EXの接続機器設定では、  
「IEC61131シンタックス」  
「SoMachine Basic シンタックス」  
にチェックを入れて使用してください。

メニューバー「プロジェクト」「システム設定」「接続機器設定」で下記  
の設定を行います。



シンタックスにチェックを入れると、SoMachine Basicのオブジェクト(アドレス)であるビットメモリ(%M)、ワードメモリ(%MW)等を  
GP-Pro EXでそのまま利用して作画することができます。

### 【 SoMachine Basicのオブジェクト(アドレス) の表記】

**プロパティ**

タスク ツール

- メッセージ
- アニメーションテーブル
- メモリオブジェクト
  - ビットメモリ
  - ワードメモリ
  - ワード型定数
- システムオブジェクト
  - システムビット
  - システムワード
  - 入力チャンネルステータス
  - 出力チャンネルステータス
  - IOScanner ステータス
- I/O オブジェクト
  - デジタル入力
  - デジタル出力
  - アナログ入力
  - アナログ出力

**ビットメモリ プロパティ**

使用	アドレス	シンボル
<input checked="" type="checkbox"/>	%M0	
<input type="checkbox"/>	%M1	
<input type="checkbox"/>	%M2	

**デジタル入力 プロパティ**

使用	アドレス	シンボル
<input type="checkbox"/>	%IO.0	
<input type="checkbox"/>	%IO.1	
<input type="checkbox"/>	%IO.2	

**ワードメモリ プロパティ** %MW %MD %MF

使用	重複使用	アドレス	シンボル
<input type="checkbox"/>	<input type="checkbox"/>	%MW0	
<input type="checkbox"/>	<input type="checkbox"/>	%MW1	
<input type="checkbox"/>	<input type="checkbox"/>	%MW2	

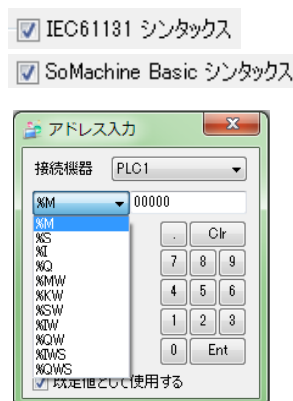
**デジタル出力 プロパティ**

使用	アドレス	シンボル
<input type="checkbox"/>	%Q0.0	
<input type="checkbox"/>	%Q0.1	
<input type="checkbox"/>	%Q0.2	

### 【シンタックスのチェックなし(GP-Pro EXのアドレス指定)】



### 【シンタックスのチェックあり(GP-Pro EXのアドレス指定)】



### 【ドラッグ＆ドロップで割付ける方法】

ここでは、a接点に「%M0」を割付けます。

- ①プログラミングツリーのツールをクリックします。
- ②メモリオブジェクトから、ビットメモリをクリックします。
- ③画面下部のビットメモリプロパティから「%M0」をa接点にドラッグ＆ドロップします。

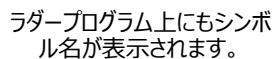


**CHECK!**

アドレスに任意の名前を付けてプログラム上で使用することができます。それぞれのアドレスに自由に名前が付けられることで、ラダー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。

### 【シンボル名の付け方】

ビットメモリプロパティのシンボル欄に直接、シンボル名を登録し、画面右下にある適用をクリックします。



## 【パソコンのキーボードから入力する方法】

ラング内に配置した命令にパソコンのキーボードから直接アドレスをタイプし、入力します。

ここでは、a接点に「%M0」を割り当てます。

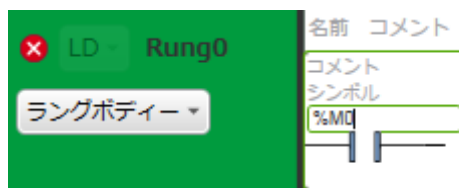
- ① a接点の「アドレス」をダブルクリックします。



- ② 「%」に続いて、アドレスを入力します。

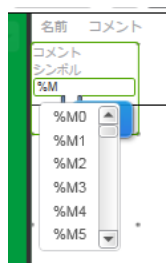


- ③ 「Enter」を押下し、入力を確定します。



**CHECK!**

アドレス入力すると、予測絞り込み表示がされるのでリストから選択することもできます。



**CHECK!**

## 便利なアドレスの割付け方法

あらかじめシンボル名が登録されている場合は、アドレスだけでなく、シンボル名を入力することもできます。

- ① a接点の「シンボル」をクリックします。
- ② 「シンボル名」を入力します。



## 5. デバッグ

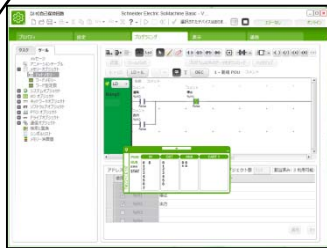
SoMachine Basicにはデバッグのための3つの機能が用意されています。

シミュレーター：SoMachine Basicで設定した内容(プロジェクト)を、M221に転送せずに、パソコン上で確認できます。

オンラインモニター：プロジェクトをM221に転送し、オンラインモニター機能を使うと、運転中のM221内のプログラムをパソコン上でモニターすることができます。

シンプルディスプレイユニット：オプションのシンプルディスプレイユニット(TM2H2GDB)を使って、プログラムが正常に動作するかどうかの確認やアドレスの現在値を確認できます。

### (1)シミュレーター(ソフトウェアでのデバッグ)



シミュレーターには便利なショートカットキーも用意しています。

**[Ctrl] + B** シミュレーター起動


**[Ctrl] + W** シミュレーター停止

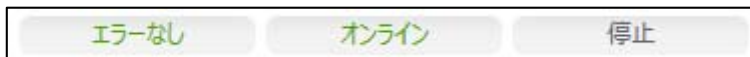
**[Ctrl] + L** コントローラー起動


**[Ctrl] + M** コントローラー停止

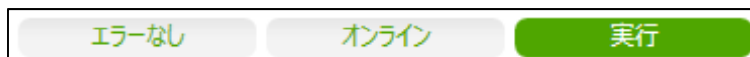
#### 【シミュレーター】

シミュレーターを使えば、ラダープログラムの動作をパソコン上で確認することができます。

①ステータスエリアの  をクリックします(シミュレーターモードに切り替わります)



②ステータスエリアの  をクリックします(シミュレーターが開始されます)



回路が導通すると、緑色に変化します。

接点の状態を右クリックで変更もできます。

CHECK!

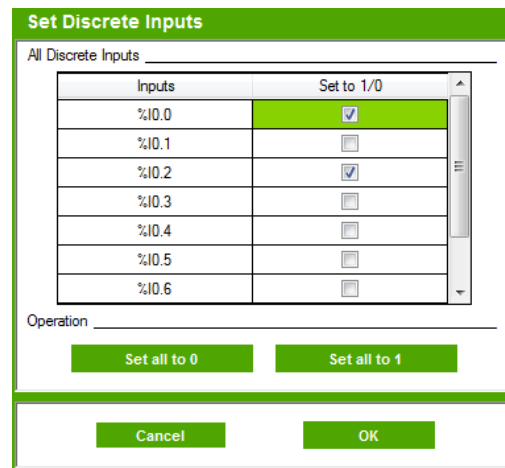
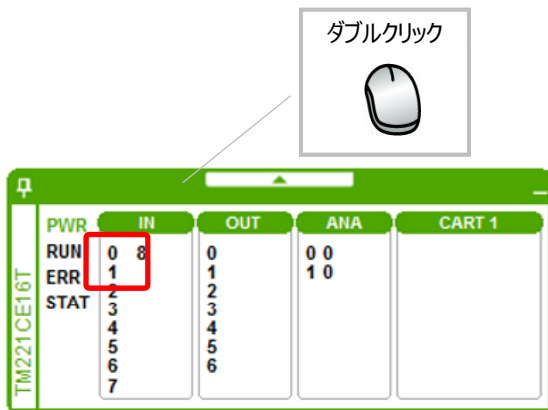


## シミュレータの便利な機能

下記の2つの機能を使えば、より効率的にデバッグを行うことができます。

### 【シミュレーター・I/Oマネージャーウィンドウ】

シミュレーターでプログラム実行中のコントローラー、および拡張モジュールの I/O を制御、および監視できます。また、LED ステータスの確認も行うことができます。



### 【アニメーションテーブル】

アニメーションテーブルを使用すれば、デバッグ(シミュレーター/オンラインモニター)時に、アドレスの現在値を変更し、動作確認を行うことができます。

- ① プログラミングツリーの「ツール」から右クリックし、「新規アニメーションテーブルの追加」を追加します。
- ② アニメーションテーブルで、アドレスを追加します。



- ③ アドレスの値を変更することができます。

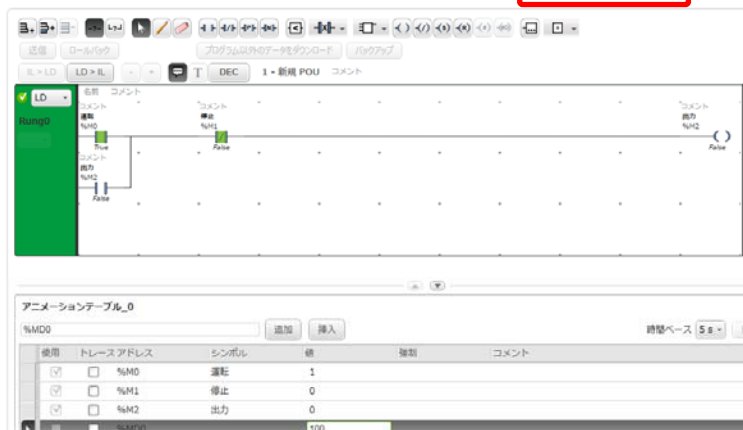
②



CHECK!

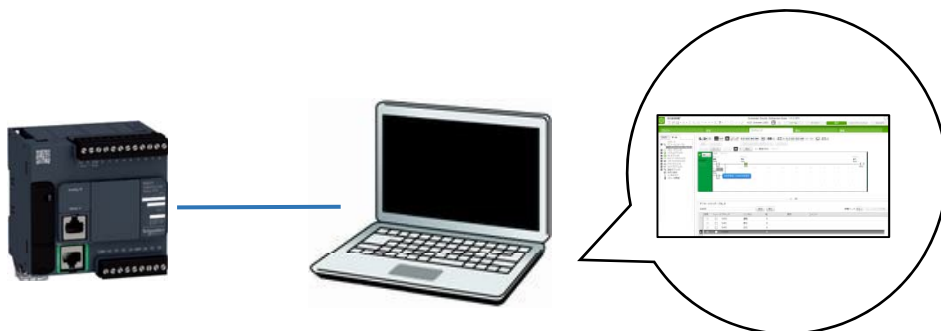


アニメーションテーブルは、次項のオンラインモニターでも活用できます。



## (2) オンラインモニター(ハードウェアでのデバッグ)

オンラインモニターを使えば、運転中のM221内のプログラムをパソコン上でモニターすることができます。アドレスの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。また、RUN中もプログラムの変更もできます。



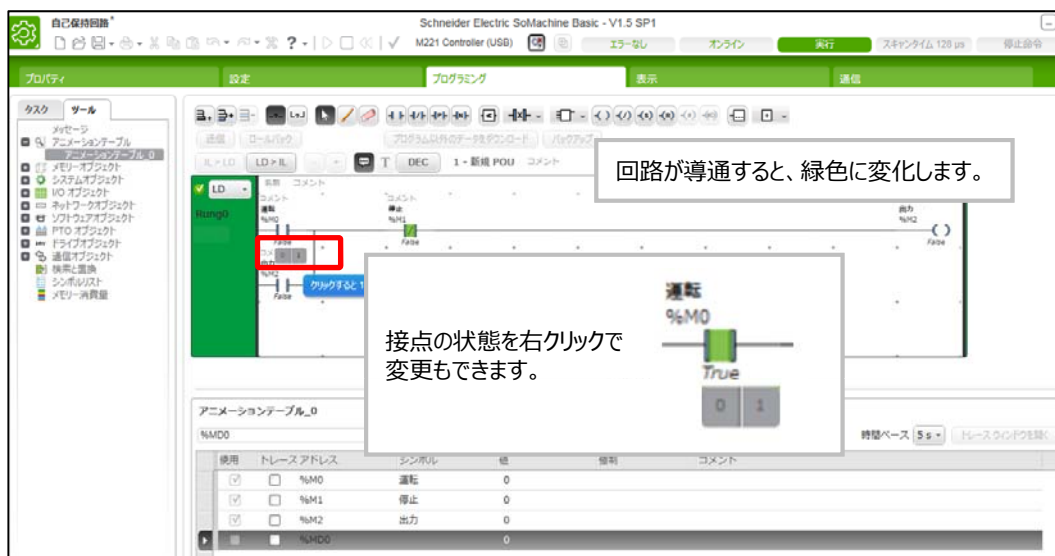
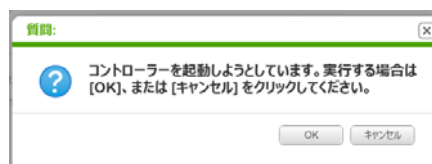
### 【オンラインモニター】

①「通信」タブをクリックします。

②ログインをクリックします。  
(ログインすることで、PCと接続されます)

③コントローラーの開始をクリックします。

④質問は、「OK」をクリックします。

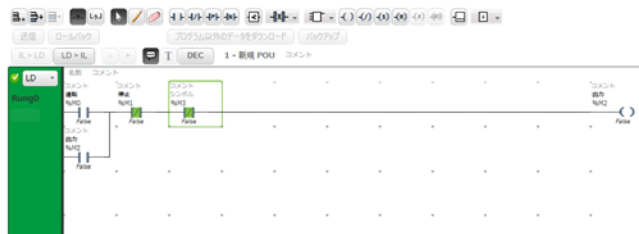


CHECK!

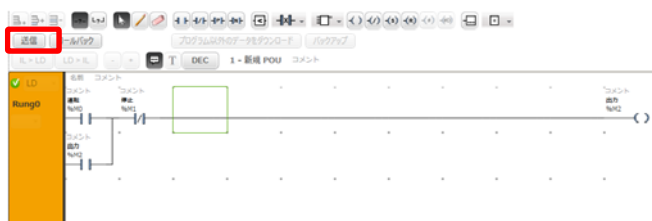
## オンラインモニター(RUN)中のRungの編集

SoMachine Basicでは、コントローラーを実行した状態のまま、Rungを改造することが可能です。  
ここでは、b接点をオンライン中に削除します。

①b接点を削除します。



②変更したラングはオレンジ色で表示されます。

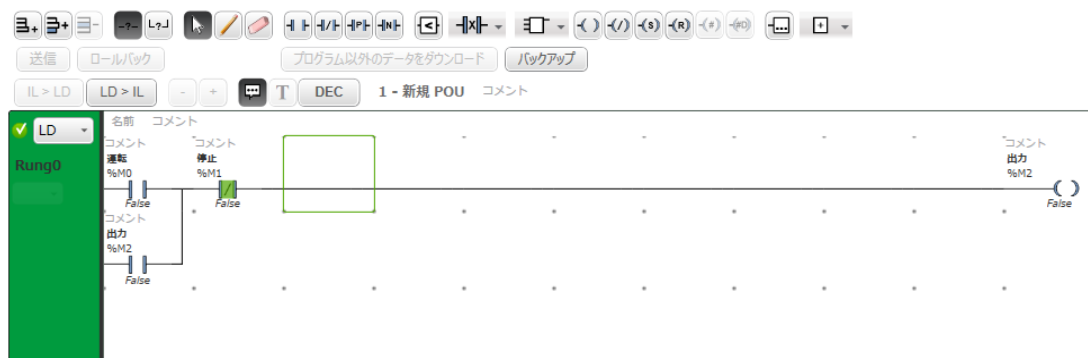
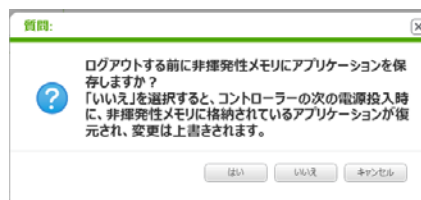


③変更箇所を反映するには、「送信」をクリックします。  
コントローラーに送信され、変更箇所が反映されます。  
(ラングが緑色になります)



※この時点では変更箇所の反映はまだ一時的です。

④ログアウトする際に、変更箇所をコントローラーに上書きするかを確認するメッセージが表示されますので、確定する場合は、「はい」を選択します。





メモ(以下は余白です。ご自由にお使いください。)



# 第2章

## 基本回路をマスターしよう

I. 【理解しよう】基本回路 その1 .....	2-2
1. ON回路	
2. OFF回路	
3. AND回路	
4. OR回路	
5. NOT回路	
II. 【やってみよう】自己保持回路 .....	2-12
1. 【実習】自己保持回路とは	
2. 自己保持回路 完成例	
3. 【解説】自己保持回路	
III. 【理解しよう】基本回路 その2 .....	2-16
1. インターロック回路	
2. SET/RSET回路	



メモ(以下は余白です。ご自由にお使いください。)

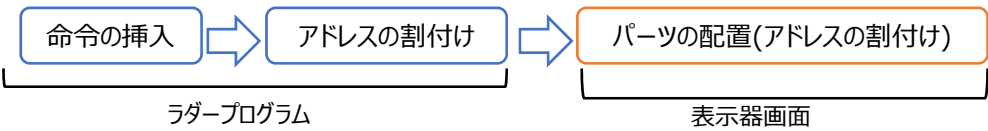
# I.【理解しよう】基本回路 その1

ここでは、SoMachine Basicのプログラムでの、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、SoMachine Basicでラダープログラムを作成し、シュミレーターで動作確認してください。あわせて、GP-Pro EXの画面作成方法もご参照ください。

## 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

【ラダープログラム例】



【表示器画面例】



【使用する命令】



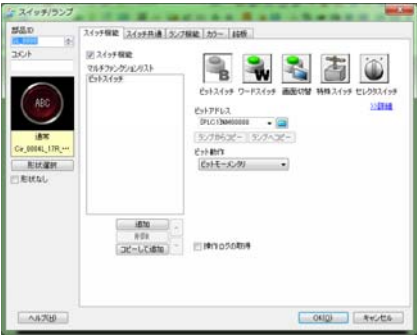
a 接点(NO)：スイッチを押さない場合、接点は開き、電気の流れは遮断されています。スイッチを押すことにより、接点が閉じ、電気が流れます。



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 (銘板:スイッチ1)
- ・ビットアドレス：%M0
  - ・ビット動作：ビットモーメンタリ



【使用するアドレス(メモリ)】

種類	アドレス	シンボル名
ビットメモリ	%M0	スイッチ1
ビットメモリ	%M1	ランプ1

- ②ランプ (銘板:ランプ1)
- ・ビットアドレス：%M1



### (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。

#### CHECK!

シミュレーターには便利な  
ショートカットキーも用意しています。

**Ctrl** + B シミュレーター起動

**Ctrl** + W シミュレーター停止

**Ctrl** + L コントローラー起動

**Ctrl** + M コントローラー停止

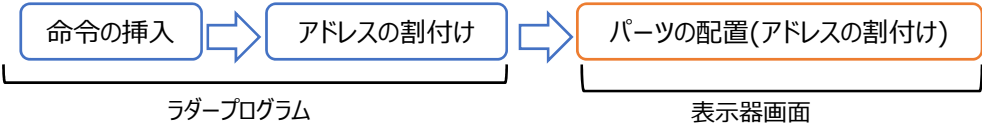


メモ(以下は余白です。ご自由にお使いください。)

2. OFF回路

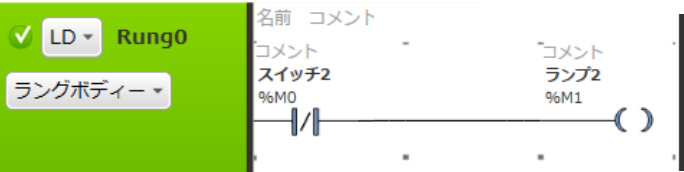
OFF回路とは入力OFFのときに出力がON、入力がONのときに出力がOFFになる回路です。

(1)作成手順



(2)作成例

【ラダープログラム例】



【使用する命令】



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

【使用するアドレス例】

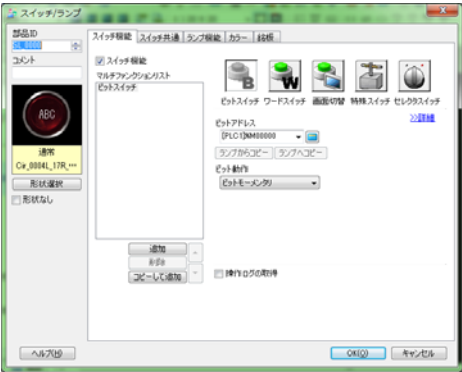
種類	アドレス	シンボル名
ビットメモリ	%M0	スイッチ2
ビットメモリ	%M1	ランプ2

【表示器画面例】

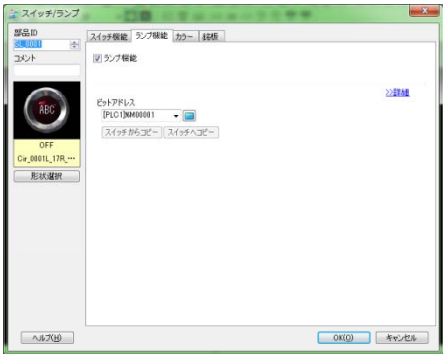


【スイッチ/ランプ設定例】

- ①スイッチ（銘板:スイッチ2）
- ・ビットアドレス：%M0
  - ・ビット動作：ビットモーメンタリ



- ②ランプ（銘板:ランプ2）
- ・ビットアドレス：%M1



※上記以外の設定は任意です。

### (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。



シミュレーターには便利な  
ショートカットキーも用意しています。

**[Ctrl] + B** シミュレーター起動

**[Ctrl] + W** シミュレーター停止

**[Ctrl] + L** コントローラー起動

**[Ctrl] + M** コントローラー停止

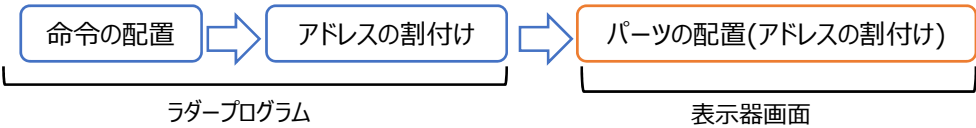


メモ(以下は余白です。ご自由にお使いください。)

3. AND回路

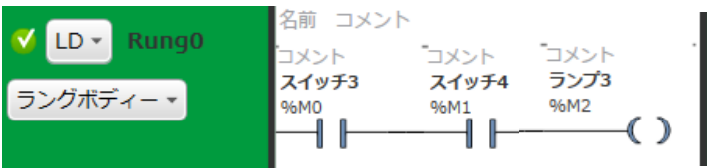
AND 回路（論理積回路）とは、複数の入力(スイッチ)と1個の出力(コイル)があり、すべての入力がONされたときに出力されます。

(1)作成手順



(2)作成例

【ラダープログラム例】



【表示器画面例】



【使用する命令】



a 接点：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：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。

【使用するアドレス例】

種類	アドレス	シンボル名
ビットメモリ	%M0	スイッチ3
ビットメモリ	%M1	スイッチ4
ビットメモリ	%M2	ランプ3

【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/ランプ（銘板:スイッチ3）  
・ビットアドレス：%M0  
・ビット動作：ビット反転  
※上記以外の設定は任意です。

②スイッチ/ランプ（銘板:スイッチ4）  
・ビットアドレス：%M1  
・ビット動作：ビット反転  
※上記以外の設定は任意です。

③ランプ（銘板:ランプ3）  
・ビットアドレス：%M2  
※上記以外の設定は任意です。

### (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。

#### CHECK!

シミュレーターには便利な  
ショートカットキーも用意しています。


**Ctrl** + B シミュレーター起動

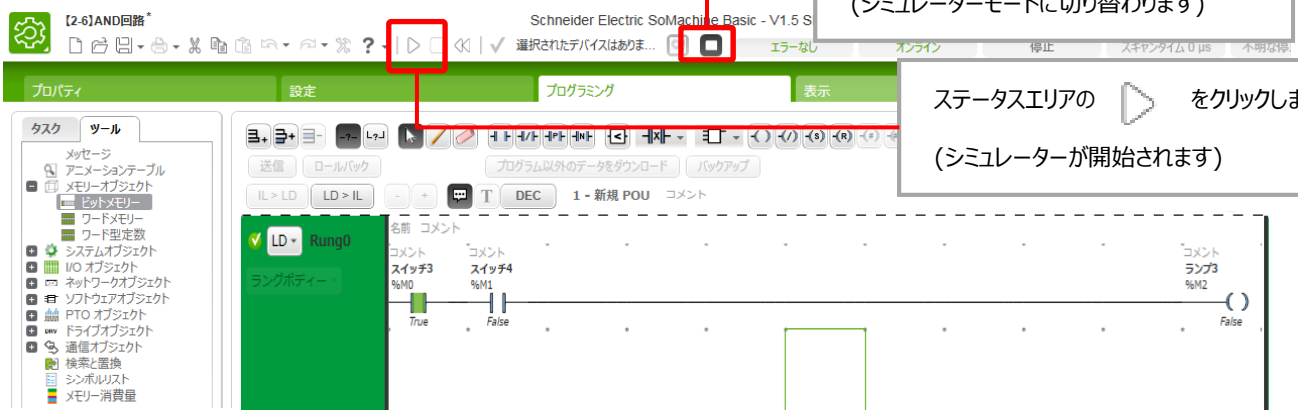
**Ctrl** + W シミュレーター停止

**Ctrl** + L コントローラー起動

**Ctrl** + M コントローラー停止

ステータスエリア  をクリックします。  
(シミュレーターモードに切り替わります)

ステータスエリアの  をクリックします。  
(シミュレーターが開始されます)



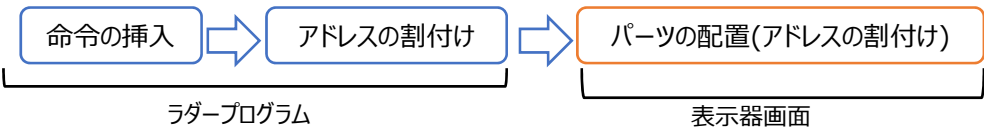
メモ(以下は余白です。ご自由にお使いください。)



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つの入力が入力ONされたときに出力されます。

(1)作成手順



(2)作成例



【表示器画面例】



【使用する命令】



a 接点：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：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。

【使用するアドレス例】

種類	アドレス	シンボル名
ビットメモリ	%M0	スイッチ5
ビットメモリ	%M1	スイッチ6
ビットメモリ	%M2	ランプ4

【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/ランプ（銘板:スイッチ5）  
・ビットアドレス：%M0  
・ビット動作：ビット反転  
※上記以外の設定は任意です。

②スイッチ/ランプ（銘板:スイッチ6）  
・ビットアドレス：%M1  
・ビット動作：ビット反転  
※上記以外の設定は任意です。

③ランプ（銘板:ランプ4）  
・ビットアドレス：%M2  
※上記以外の設定は任意です。

### (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。



シミュレーターには便利な  
ショートカットキーも用意しています。

**[Ctrl] + B** シミュレーター起動

**[Ctrl] + W** シミュレーター停止

**[Ctrl] + L** コントローラー起動

**[Ctrl] + M** コントローラー停止

ステータスエリア をクリックします。  
(シミュレーターモードに切り替わります)

ステータスエリアの をクリックします。  
(シミュレーターが開始されます)

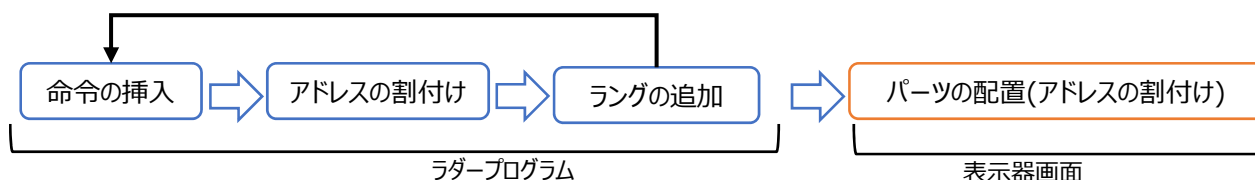


メモ(以下は余白です。ご自由にお使いください。)

## 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

【ラダープログラム例】



【表示器画面例】



【使用する命令】



a 接点：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：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：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。

【使用するアドレス例】

種類	アドレス	シンボル名
ビットメモリ	%M0	スイッチ7
ビットメモリ	%M1	テンポラリ
ビットメモリ	%M2	ランプ5

【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/ランプ (銘板:スイッチ7)

・ビットアドレス：%M0

・ビット動作：ビット反転

※上記以外の設定は任意です。

②ランプ (銘板:ランプ5)

・ビットアドレス：%M2

※上記以外の設定は任意です。

## (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。



シミュレーターには便利な  
ショートカットキーも用意しています。

**Ctrl** + B シミュレーター起動

**Ctrl** + W シミュレーター停止

**Ctrl** + L コントローラー起動

**Ctrl** + M コントローラー停止

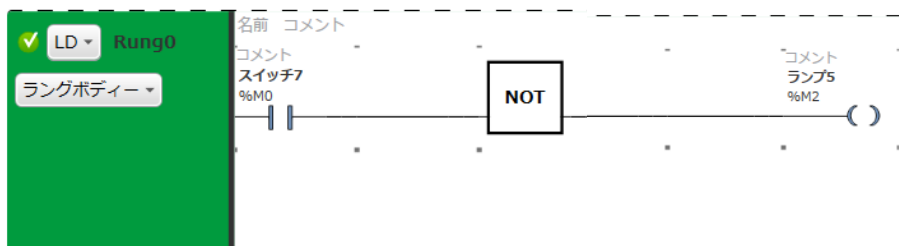
ステータスエリア  をクリックします。  
(シミュレーターモードに切り替わります)

ステータスエリアの  をクリックします。  
(シミュレーターが開始されます)

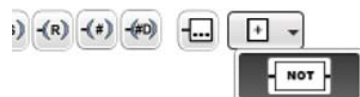
### CHECK!

#### NOT命令

SoMachine Basicには、NOT命令が用意されています。NOT命令を使用すると、1つのラングでNOT回路を作成することができます。



NOT命令はツールバーの左端「他のラダー項目」から挿入することができます。



Ⅱ.【やってみよう】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、ラダー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ターで動作確認してください。

- 【要求仕様】
- 下記の2つの要求を満たすラダー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。

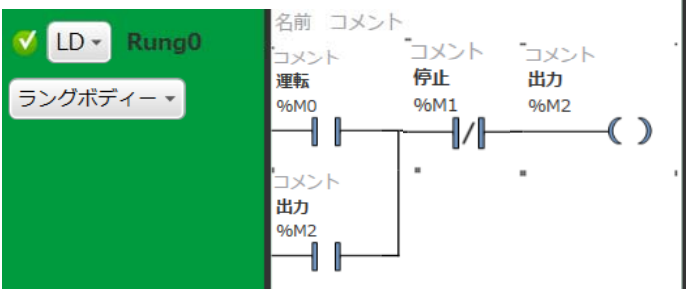
【PLC】  
シリーズ：Modicon M221  
機種：TM221CE16R  
IPアドレス：192.168.1.20

【Pro-face製表示器】  
製品名：GP-4114T(PFXGP4114T2D)  
接続機器：[Schneider Electric SA]Modbus TCP マスタ  
IPアドレス：192.168.1.100

2. 自己保持回路 完成例

下記は、「SoMachine Basicのラダープログラムの作成例」と「GP-Pro EXを使った表示器画面の作成例」です。  
まずは要求仕様と作成例を参考に設定を行いましょう。

【ラダープログラム例】



【表示器画面例】



【使用する命令】



【使用するアドレス例】

種類	アドレス	シンボル名
ビットメモリ	%M0	運転
ビットメモリ	%M1	停止
ビットメモリ	%M2	出力

【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（銘板:運転）  
・ビットアドレス：%M0  
・ビット動作：ビットモーメンタリ
- ②スイッチ（銘板:停止）  
・ビットアドレス：%M1  
・ビット動作：ビットモーメンタリ
- ③ランプ（銘板:出力）  
・ビットアドレス：%M2

※上記以外の設定は任意です。

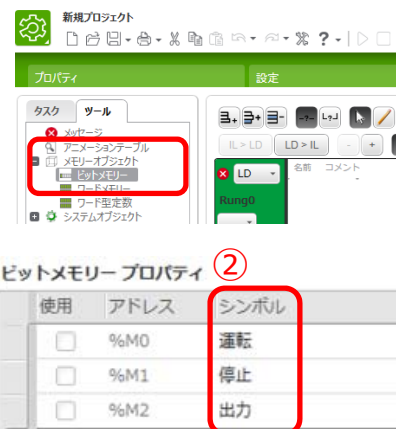
### 3. 【解説】自己保持回路

#### (1) ラダープログラム

【使用するアドレスにシンボル名を付けます】

①「プログラムツリー」「ツール」から「メモリオブジェクト」「ビットメモリー」をクリックします。

①

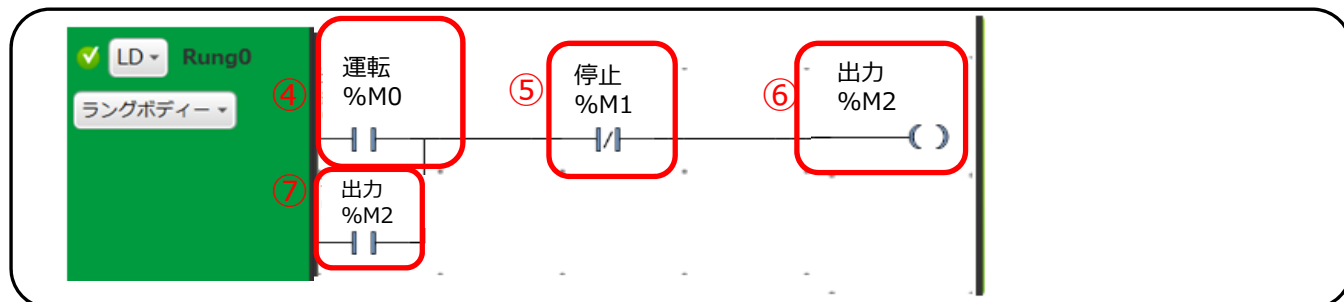



②「ビットメモリープロパティ」で、各アドレスに対して、シンボル名を追加します。

③「適用」をクリックします。

【シンボル名を追加したら、下記に沿って、設定を行います】

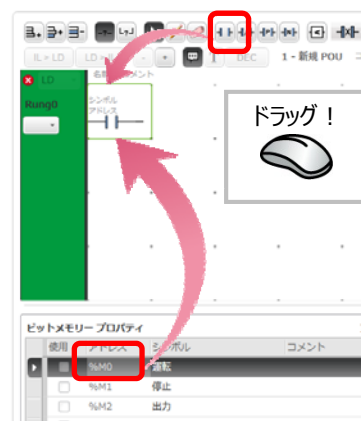
適用




④  をクリックします。


ラングに配置します。

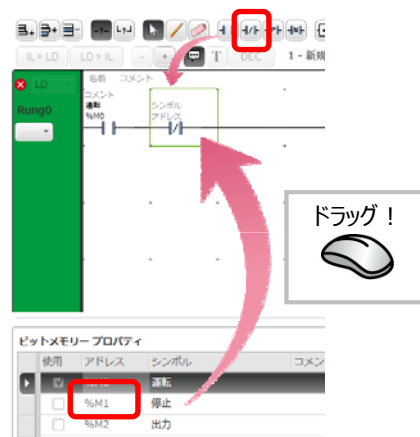
ビットメモリープロパティの「%M0」を  にドラッグし、割付けます。




⑤  をクリックします。


ラングに配置します。

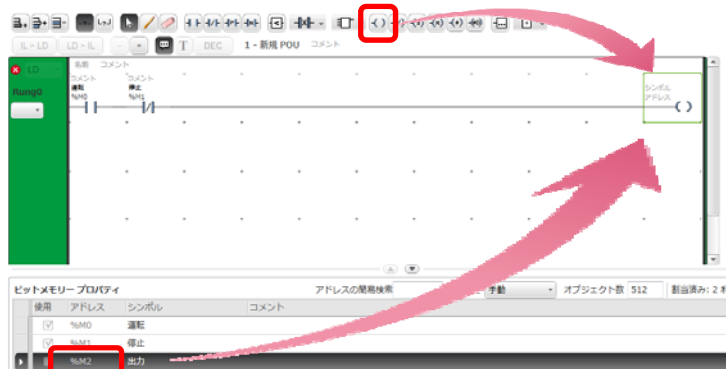
ビットメモリープロパティの「%M1」を  にドラッグし、割付けます。




- ⑥  をクリックします。

ラングに配置します。


「ビットメモリプロパティ」の「%M2」を  にドラッグし割付けます。

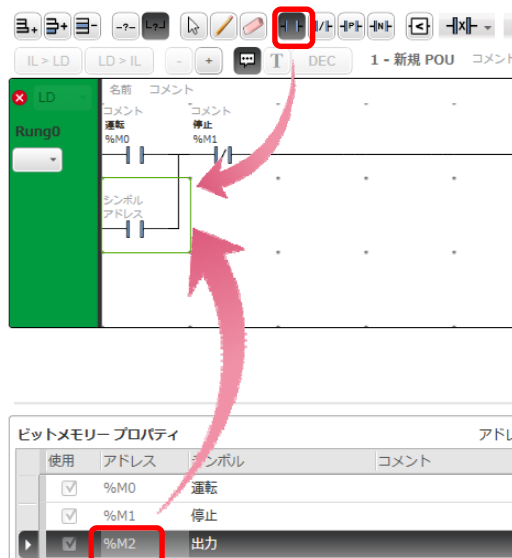
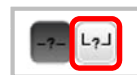


- ⑦ ツールバーで「通常モード(分岐なし)」を「分岐モード」に変更します。

 をクリックします。

ラングに配置します。

「ビットメモリプロパティ」の「%M0」を  にドラッグし、割付けます。



**CHECK!** 


設定が完了すれば、「通常モード(分岐なし)」に戻します。



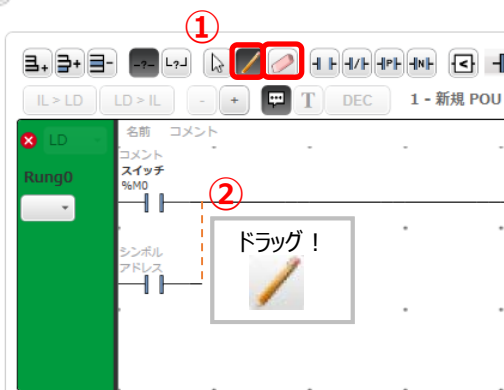
**CHECK!** 


### 便利な分岐の挿入方法

分岐は「通常モード(分岐なし)」でも、「ラインの描画」  を使用するとラングに分岐を挿入することができます。

- ①  をクリックします。

- ② 分岐箇所をドラッグしてラインを描画します。



ラインを消去する場合は、 をクリックし、消去箇所をドラッグします。

### (2)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、ビットアドレス「%M0」がONし、③ランプのビットアドレス「%M2」がONして回路が導通します。  
②スイッチをタッチすると、ビットアドレス「%M1」がONし、運転と出力間の回路が切れるため、③ランプ「%M2」はOFFします。  
つまり、「%M1」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（銘板:運転）  
・ビットアドレス：%M0  
・ビット動作：モーメンタリ

②スイッチ（銘板:停止）  
・ビットアドレス：%M1  
・ビット動作：モーメンタリ。

③ランプ（銘板:出力）  
・ビットアドレス：%M2

※上記以外の設定は任意です。

### (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。

#### CHECK!

シミュレーターには便利な  
ショートカットキーも用意しています。

**[Ctrl] + B** シミュレーター起動

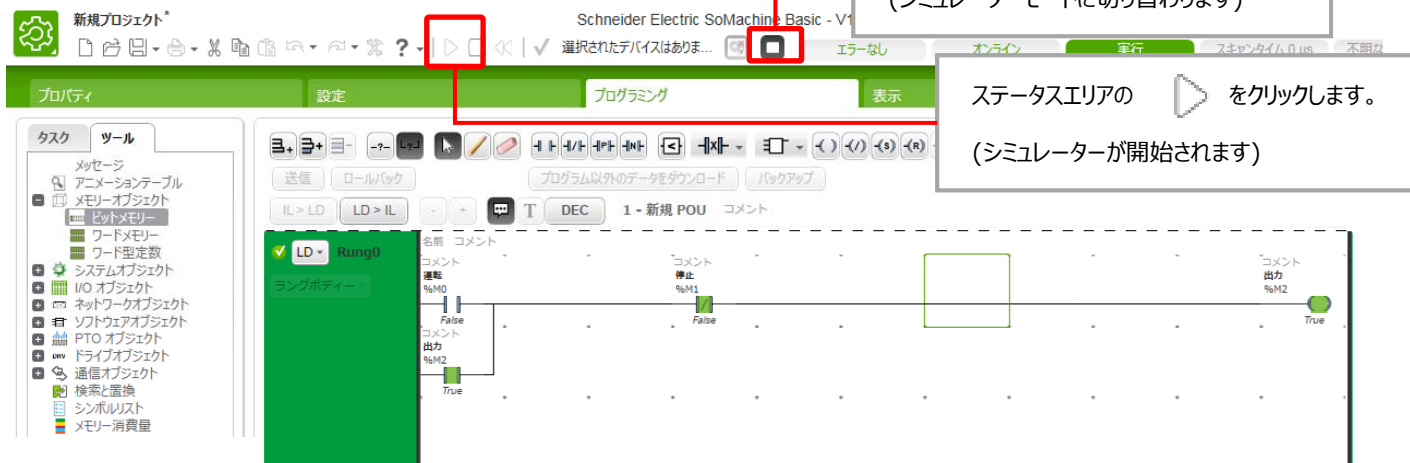
**[Ctrl] + W** シミュレーター停止

**[Ctrl] + L** コントローラー起動

**[Ctrl] + M** コントローラー停止

ステータスエリアの をクリックします。  
(シミュレーターモードに切り替わります)

ステータスエリアの をクリックします。  
(シミュレーターが開始されます)





Ⅲ.【理解しよう】基本回路 その2

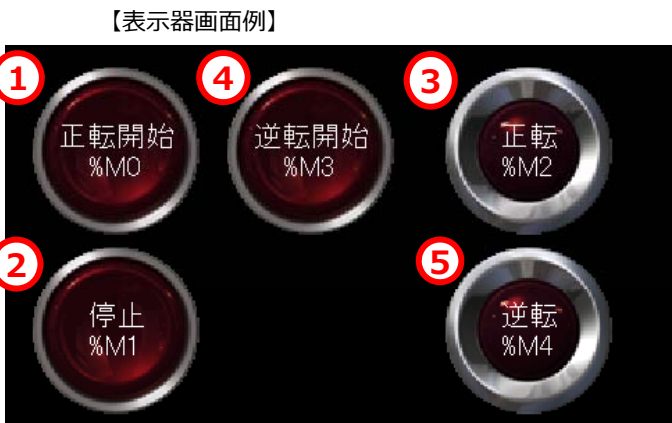
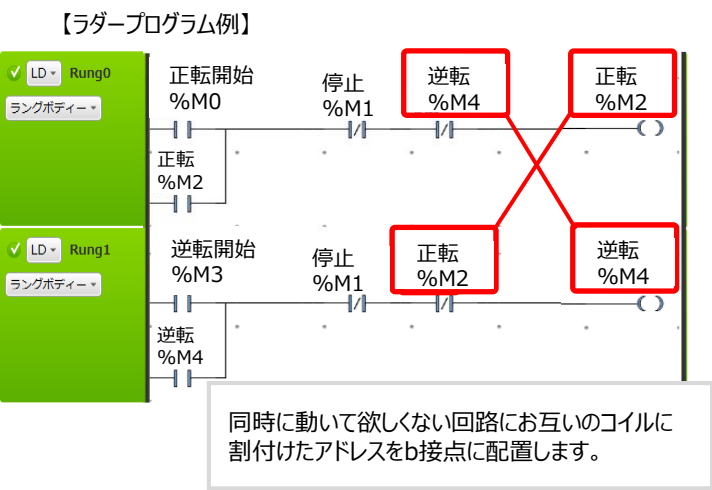
ここでは、SoMachine Basicのラダープログラムを使った「インターロック」「セット/リセット」回路の作成方法を解説します。SoMachine Basicでプログラムを作成し、シミュレーターで動作確認してください。あわせて、GP-Pro EXの画面作成方法もご参照ください。

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、機械が無理なく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、ビットアドレス「%M0」がONし、④ランプのビットアドレス「%M3」がONして回路が導通します。  
②スイッチをタッチしても、ビットアドレス「%M3」がONしているので、⑤ランプのビットアドレス「%M4」はONしません。  
逆も同様に、ビットアドレス「%M4」がONしているときは「%M3」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



【使用する命令】



【使用するアドレス例】

種類	アドレス	シンボル名
ビットメモリ	%M0	正転開始
ビットメモリ	%M1	停止
ビットメモリ	%M2	正転
ビットメモリ	%M3	逆転開始
ビットメモリ	%M4	逆転

- ①スイッチ（銘板:正転開始）  
・ビットアドレス： %M0  
・ビット動作：ビットモーメンタリ
- ②スイッチ（銘板:停止）  
・ビットアドレス： %M1  
・ビット動作：ビットモーメンタリ
- ③ランプ（銘板:正転）  
・ビットアドレス： %M2
- ④スイッチ（銘板:逆転開始）  
・ビットアドレス： %M3  
・ビット動作：ビットモーメンタリ
- ⑤ランプ（銘板:逆転）  
・ビットアドレス： %M4

※ 上記以外の設定は任意です。

## (2)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。




シミュレーターには便利な  
ショートカットキーも用意しています。


**[Ctrl] + B** シミュレーター起動

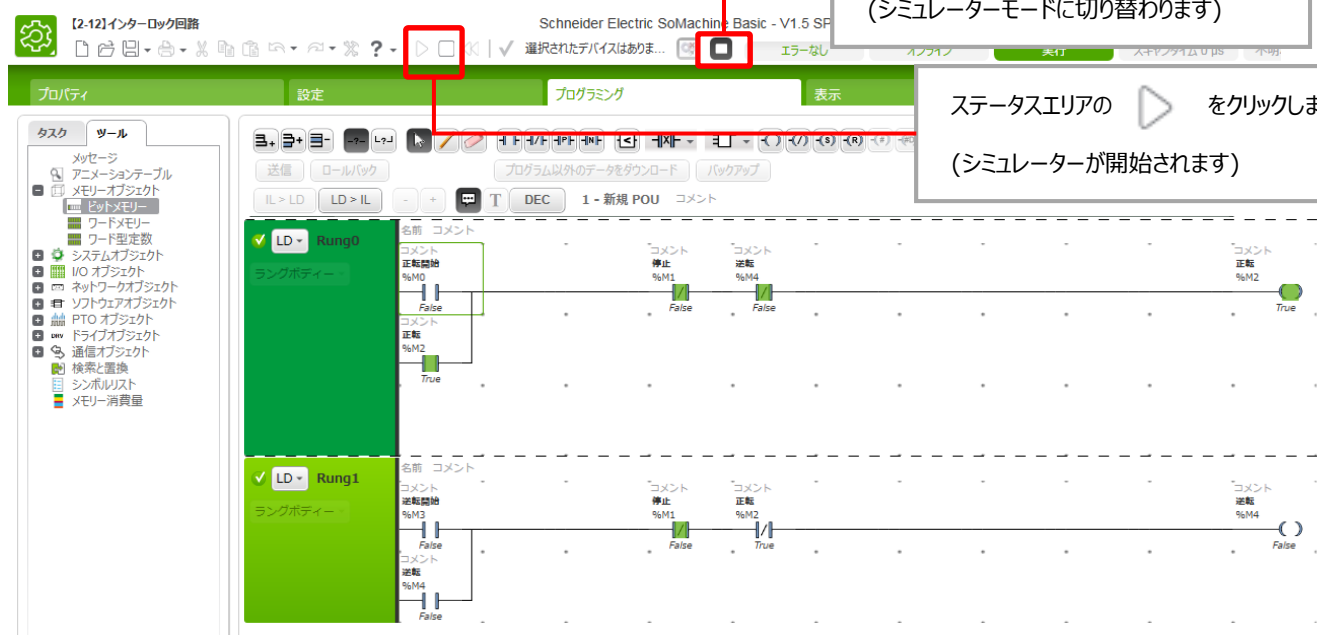
**[Ctrl] + W** シミュレーター停止

**[Ctrl] + L** コントローラー起動

**[Ctrl] + M** コントローラー停止

ステータスエリアの  をクリックします。  
(シミュレーターモードに切り替わります)

ステータスエリアの  をクリックします。  
(シミュレーターが開始されます)



2. SET/RSET回路

SET/RSET回路とは、入力条件がONすると、指定したアドレスを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、リセット命令を使用します。回路の動きは、前述した自己保持回路と同様です。

※SETとRSETは必ずペアで使ってください。



【セットコイル】  
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。



【リセットコイル】  
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、ビットアドレス「%M0」がONし、③ランプのビットアドレス「%M1」がONして回路が導通します。  
②スイッチをタッチすると、ビットアドレス「%M2」がONし、運転と出力間の回路が切れるため、③ランプ「%M1」はOFFします。  
つまり、「%M1」がリセット信号の役割を果たしています。

【SET/RSET回路】



【使用する命令】



a接点(NO)

【アドレス】

種類	アドレス(メモリ)	シンボル名
ビットメモリ	%M0	運転
ビットメモリ	%M1	出力
ビットメモリ	%M2	停止

【表示器画面例】



①スイッチ（銘板:運転）  
・ビットアドレス：%M0  
・ビット動作：ビットモーメンタリ

②ランプ（銘板:出力）  
・ビットアドレス：%M1

③スイッチ（銘板:停止）  
・ビットアドレス：%M2  
・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)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。



シミュレーターには便利な  
ショートカットキーも用意しています。

**Ctrl** + B シミュレーター起動

**Ctrl** + W シミュレーター停止

**Ctrl** + L コントローラー起動

**Ctrl** + M コントローラー停止

ステータスエリアの をクリックします。  
(シミュレーターモードに切り替わります)

ステータスエリアの をクリックします。  
(シミュレーターが開始されます)



メモ(以下は余白です。ご自由にお使いください。)

# 第3章

## 応用回路をマスターしよう

I . 応用回路 その1 .....	3-2
1. タイマー回路	
2. カウンター回路	
II . 応用回路 その2 .....	3-8
1. データ転送回路	
2. インクリメント/デクリメント演算	
3. 比較演算回路	



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 その1

ここでは、SoMachine Basicのラダープログラムを使った「タイマー」「カウンタ」回路の作成方法を解説します。SoMachine Basicでプログラムを作成し、シミュレーターで動作確認してください。あわせて、GP-Pro EXの画面作成方法もご参照ください。

1. タイマー回路

タイマー回路とは、設定時間後に接点が動作する回路のことです。SoMachine Basicでは、下記の3種類のタイマーのタイプを用意しています。

(1)作成手順

タイマーの設定方法 ※タイマーを使用するには、下記2つの設定が必要です。  
「タイマープロパティ」：タイマーアドレスで、タイプとプリセット値の設定を行います。  
「タイマー命令」：ラングにタイマー命令を配置し、タイマーアドレスを割付けます。



- ①タイマープロパティ
- ・「プログラムツリー」「ツール」の「ソフトウェアオブジェクト」「タイマー」をクリックします。
  - ・タイマープロパティの「タイプ」で3つのタイマタイプから指定します。
- 【オンディレイタイマー】(TON)  
【オフディレイタイマー】(TOF)  
【パルスタイマー】(TP)
- ・設定時間を指定します(時間ベース(単位) / プリセット)



タイマー プロパティ

使用	アドレス	シンボル	タイプ	保持	時間ベース	プリセット
<input checked="" type="checkbox"/>	%TM0	オンディレイタイマー	TON	<input type="checkbox"/>	1 ms	3000
<input checked="" type="checkbox"/>	%TM1	オフディレイタイマー	TON	<input type="checkbox"/>	1 ms	3000
<input checked="" type="checkbox"/>	%TM2	パルスタイマー	TOF	<input type="checkbox"/>	1 ms	3000
<input type="checkbox"/>	%TM3		TP	<input type="checkbox"/>	1 min	9999

アドレスの簡易検索

【タイマーのタイプ】 ※図は3000ms(=3秒)のタイマー動作の例です。



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



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



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

- ②タイマー命令
- ラングにタイマー命令を配置します。配置後は、タイマーアドレスを割付けます。



【タイマーアドレス(構造体アドレス)】

タイマー命令にはタイマーアドレスを割付けます。タイマーアドレスは、それぞれの役割を持った要素アドレスの集まり(構造体アドレス)です。要素アドレスの役割は下記になります。

要素アドレス名	アドレスタイプ	役割
%TM※.IN	ビットメモリ	タイマー計測時にON(タイマーアドレスに信号が導通するとONします)
%TM※.Q	ビットメモリ	タイマー計測完了時にON(計測が完了(%TM※.V = %TM※.P)するとONします)
%TM※.P	ワードメモリ	タイマーの設定値
%TM※.V	ワードメモリ	タイマーの現在値

(2)作成例

ここでは、「オンディレイタイマー」「オフディレイタイマー」「パルスタイマー」を使った回路について説明します。  
表示器画面上の各種スイッチをタッチすると、タイマー計測が開始されます。

【タイマー回路】



【表示器画面例】



※例ではタイマー設定値を3000(ms)に設定されていますが、タイマープロパティで変更可能です。

【使用する命令】



①スイッチ/ランプ (銘板：開始1) ・ビットアドレス：%M0 ・ビット動作：ビット反転	④ランプ (銘板：ランプ1) ・ビットアドレス：%M3
②スイッチ/ランプ (銘板：開始2) ・ビットアドレス：%M1 ・ビット動作：ビット反転	⑤ランプ (銘板：ランプ2) ・ビットアドレス：%M4
③スイッチ/ランプ (銘板：開始3) ・ビットアドレス：%M2 ・ビット動作：ビット反転	⑥ランプ (銘板：ランプ3) ・ビットアドレス：%M5

※上記以外の設定は任意です。

【使用するアドレス例】

種類	アドレス	シンボル名	タイプ	時間ベース	プリセット
ビットメモリ	%M0	開始1			
ビットメモリ	%M1	開始2			
ビットメモリ	%M2	開始3			
ビットメモリ	%M3	ランプ1			
ビットメモリ	%M4	ランプ2			
ビットメモリ	%M5	ランプ3			
タイマーメモリ	%TM0	オンディレイタイマー	TON	1ms	3000
タイマーメモリ	%TM1	オフディレイタイマー	TOF	1ms	3000
タイマーメモリ	%TM2	パルスタイマー	TP	1ms	3000



## (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。

ステータスエリアの をクリックします。  
(シミュレーターモードに切り替わります)

ステータスエリアの をクリックします。  
(シミュレーターが開始されます)

**CHECK!**

シミュレーターには便利な  
ショートカットキーも用意しています。

- Ctrl** + B シミュレーター起動
- Ctrl** + W シミュレーター停止
- Ctrl** + L コントローラー起動
- Ctrl** + M コントローラー停止

変名	コメント	タイプ	TB	プリセット	初期値
IN オンディレイタ...	%TM0	TON	1 ms	3000	0
IN オフディレイタ...	%TM1	TOF	1 ms	3000	0
IN パルスタイマー	%TM2	TP	1 ms	3000	0



メモ（以下は余白です。ご自由にお使いください。）

## 2. カウンター回路

カウンター回路とは、設定回数に到達すると接点が動作する回路です。SoMachine Basicでは、下記の2種類のカウンタを用意しています。一つの命令で、アップカウンターとダウンカウンターを切り替えて使うことができます。

### (1)作成手順

例)生産数のカウント  
目標値に対して何個通過したか、達したか など

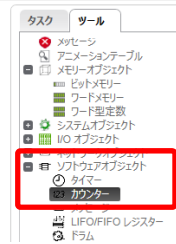


目標に達したら  
コンベアを止める

カウンターの設定方法 ※カウンタを使用するには、下記2つの設定が必要です。  
「カウンタープロパティ」：カウンターアドレスでプリセット値の設定を行います  
「カウンター命令」：ラングにカウンター命令を配置し、タイマーアドレスを割付けます。

#### ①カウンタープロパティ

- ・「プログラムツリー」の「ツール」の「ソフトウェアオブジェクト」の「カウンター」をクリックします。
- ・設定回数を指定します(プリセット)

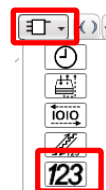


カウンター プロパティ				アドレスの簡易検索	割り当て
使用	アドレス	シンボル	プリセット	コメント	
<input checked="" type="checkbox"/>	%C0		3		
<input type="checkbox"/>	%C1		9999		
<input type="checkbox"/>	%C2		9999		
<input type="checkbox"/>	%C3		9999		

※図は3回のカウンタ動作の例です。

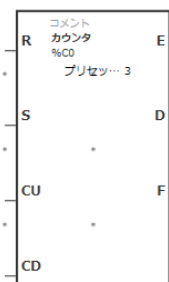
#### ②カウンター命令

ラングにカウンター命令を配置します。配置後は、カウンターアドレスを割付けます。



#### 【カウンターの種類】

※図は3回のカウンタ動作の例です。



#### 【加算カウンター】(CU)

CUに導通するたびに、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(%C※.DがON)します。

#### 【減算カウンター】(CD)

CDに導通するたびに、設定回数まで1ずつ減算し、0に到達すると出力(%C※.DがON)します。

#### 【カウンターアドレス(構造体アドレス)】

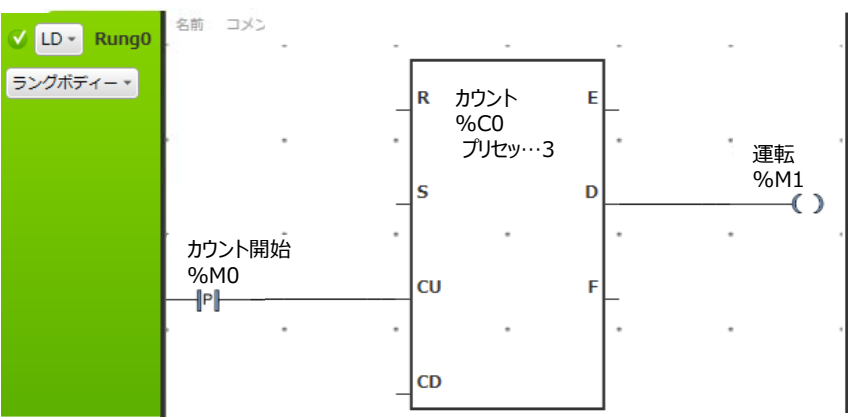
カウンター命令にはカウンターアドレスを割付けます。カウンターアドレスは、それぞれの役割を持った要素アドレスの集まり(構造体アドレス)です。要素アドレスの役割は下記になります。

要素アドレス名	アドレスタイプ	役割
%C※.CU	ビットメモリ	導通するたびに1ずつカウントアップします。
%C※.CD	ビットメモリ	導通するたびに1ずつカウントダウンします。
%C※.R	ビットメモリ	現在値リセット。
%C※.D	ビットメモリ	カウンター計測完了時にON(現在値が設定値に達するとONします)
%C※.P	ワードメモリ	カウンターの設定値
%C※.V	ワードメモリ	カウンターの現在値

(2)作成例

ここでは、アップカウンター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、カウントアップします。設定回数(例：3)に達すると「%M1」がONします。

【カウンタ回路】



【表示器画面例】



※例ではカウント設定値を3に設定されていますが、タイマープロパティで変更可能です。

【使用する命令】

立ち上がり検出接点

コイル出力

カウンター

- ①スイッチ（銘板：カウント開始）  
・ビットアドレス：%M0  
・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（銘板：運転）  
・ビットアドレス：%M1  
※上記以外の設定は任意です。

【使用するアドレス例】

種類	アドレス(メモリ)	シンボル名	プリセット
ビットメモリ	%M0	カウント開始	
ビットメモリ	%M1	運転	
カウンタ	%C0	カウンタ	3



メモ（以下は余白です。ご自由にお使いください。）

## (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。

ステータスエリアの をクリックします。  
(シミュレーターモードに切り替わります)

ステータスエリアの をクリックします。  
(シミュレーターが開始されます)

**CHECK!**

シミュレーターには便利な  
ショートカットキーも用意しています。

- Ctrl** + B シミュレーター起動
- Ctrl** + W シミュレーター停止
- Ctrl** + L コントローラー起動
- Ctrl** + M コントローラー停止



メモ（以下は余白です。ご自由にお使いください。）

II. 応用回路 その2

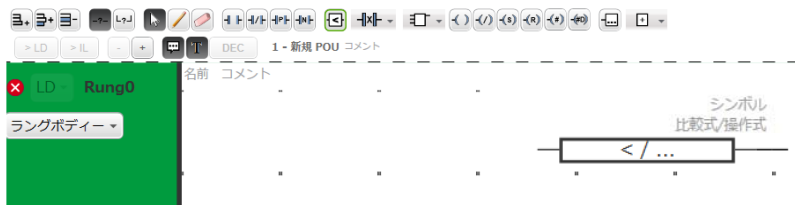
ここでは、SoMachine Basicのラダープログラムの「転送命令」「比較演算命令」を使った回路の作成方法を説明します。  
本テキストを参照しながら、SoMachine Basicでラダープログラムを作成し、シミュレーターで動作確認してください。あわせて、GP-Pro EXの画面作成方法もご参照ください。

1. データ転送回路

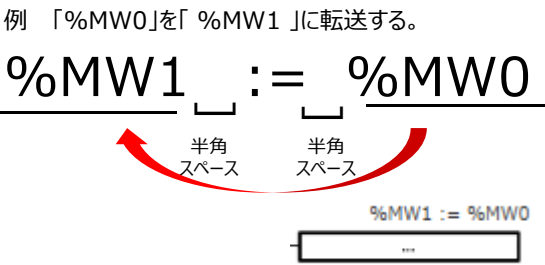
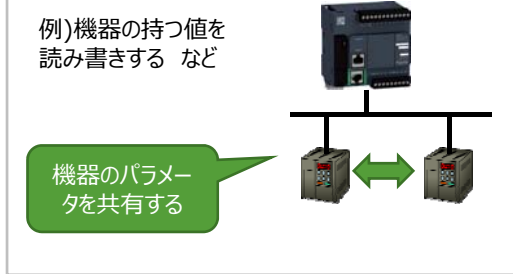
転送は、アドレスに格納されたデータを別のアドレスに、コピーする命令です。

(1)作成手順

データ転送を行うには、演算ブロックをラングに配置し、操作式を入力します。



【演算ブロック】  
演算ブロックを配置すると、操作式を入力することができます。  
転送(データコピー)や算術演算を入力すると、入力内容に従って、データを加工します。

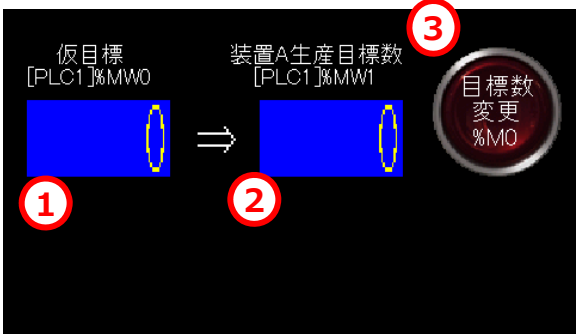
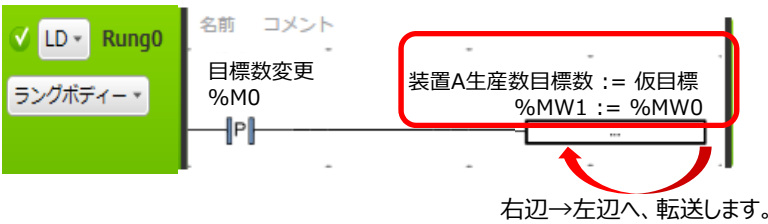


(2)作成例

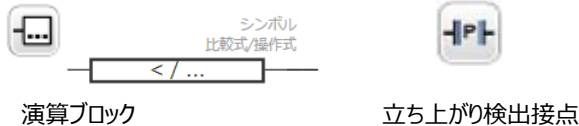
- 表示器画面上の
- ①データ表示器に任意の数値を入力するとワードアドレス「%MW0」の値が更新されます。
  - ②スイッチをタッチすると、ワードアドレス「%MW1」の値が整数アドレス「装置A生産目標数」にコピーされます。

【転送命令(MOV)を使用した基本回路】

【表示器画面例】



【使用する命令】



【使用するアドレス例】


種類	アドレス(メモリ)	シンボル名
ビットメモリ	%M0	目標数変更
ワードメモリ	%MW0	仮目標
ワードメモリ	%MW1	装置A生産目標数

- ①データ表示器 (仮目標 [PLC]%MW0)  
・ワードアドレス : %MW0
- ②データ表示器 (装置A生産目標数 [PLC]%MW1)  
・ワードアドレス : %MW1
- ③スイッチ (銘板 : 目標数変更 %M0)  
・ビットアドレス : %M0  
・ビット動作 : ビットモーメント

※上記以外の設定は任意です。


(3)動作確認


シミュレーターを使えば、パソコン上で動作確認できます。


ステータスエリアの  をクリックします。  
(シミュレーターモードに切り替わります)


CHECK!


シミュレーターには便利な  
ショートカットキーも用意しています。

 + B シミュレーター起動

 + W シミュレーター停止

 + L コントローラー起動

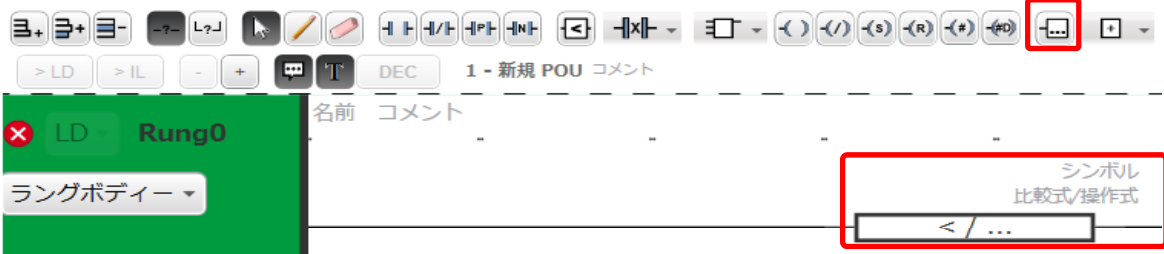
 + M コントローラー停止

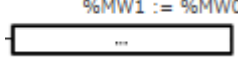
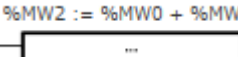
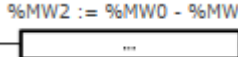
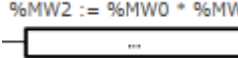
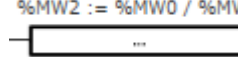

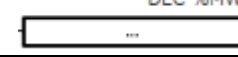
ステータスエリアの  をクリックします。  
(シミュレーターが開始されます)

CHECK!

演算ブロック



演算ブロックを使えば、ラダー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。



名称	式(サンプル)	図(サンプル)	機能
転送(コピー)	%MW1 := %MW0		「%MW0」を「%MW1」に転送します。
加算演算	%MW2 := %MW0 + %MW1		「%MW0」と「%MW1」が加算され、 「%MW2」に結果が格納されます。
減算演算	%MW2 := %MW0 - %MW1		「%MW0」から「%MW1」が減算され、 「%MW2」に結果が格納されます。
乗算演算	%MW2 := %MW0 * %MW1		「%MW0」と「%MW1」が乗算され、 「%MW2」に結果が格納されます。
除算演算	%MW2 := %MW0 / %MW1		「%MW0」から「%MW1」が除算され、 「%MW2」に結果が格納されます。
インクリメント	INC %MW0		「%MW0」を1ずつ増やします。
デクリメント	DEC %MW0		「%MW0」を1ずつ減らします。

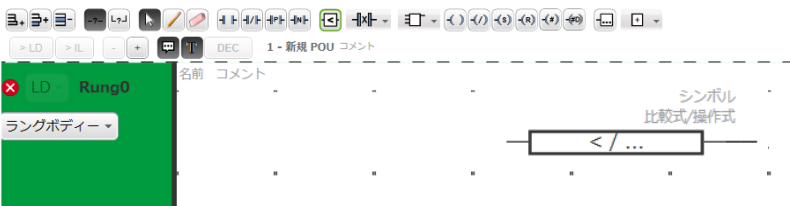
2. インクリメント/デクリメント演算

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。

命令名	呼称	ラダー記号	機能
INC	インクリメント演算		INC命令を実行すると、「%MW0」を1加算します。
DEC	デクリメント演算		DEC命令を実行すると、「%MW0」を1減算します。

(1)作成手順

データ転送を行うには、演算ブロックをラングに配置し、操作式を入力します。



【演算ブロック】  
演算ブロックを配置すると、操作式を入力することができます。  
ここでは、インクリメント演算を使用します。


例)数の確認(目標値なし)  
スイッチが何回押されたか数える など



インクリメントのための操作式は下記です。

INC %MW0

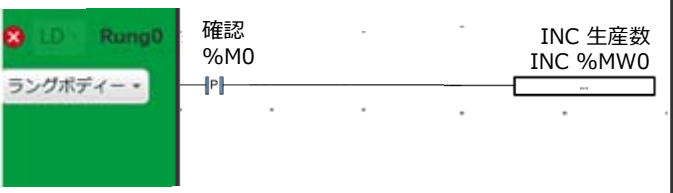
半角  
スペース



(2)作成例

表示器画面上の①確認をタッチすると、②生産数が1ずつ増加します。  
下図を参考にラダー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)

【ラダープログラム例】(インクリメント演算)



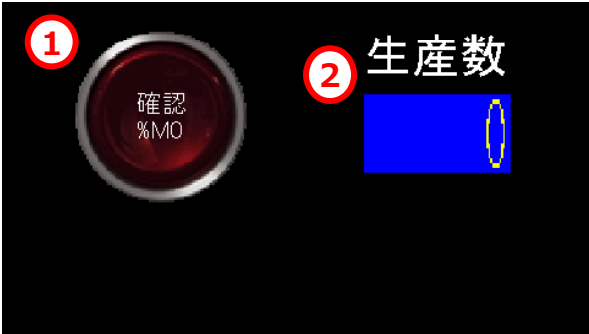
【使用する命令】



【使用するアドレス例】

種類	アドレス(メモリ)	シンボル名
ビットメモリ	%M0	確認
ワードメモリ	%MW0	生産数

【表示器画面例】



【スイッチ/ランプ設定例】

①スイッチ (銘板：確認)  
・ビットアドレス(変数)： %M0  
・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データ表示器 (生産数)  
・ワードアドレス： %MW0  
・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。

### (3)動作確認

シミュレーターを使えば、パソコン上で動作確認できます。



シミュレーターには便利な  
ショートカットキーも用意しています。

**Ctrl** + B シミュレーター起動


**Ctrl** + W シミュレーター停止


**Ctrl** + L コントローラー起動

**Ctrl** + M コントローラー停止



減算の場合は、INC  
命令をDEC命令に  
置き換えます。

ステータスエリアの  をクリックします。  
(シミュレーターモードに切り替わります)


ステータスエリアの  をクリックします。  
(シミュレーターが開始されます)



CHECK!

#### スマートコーディング

演算ブロックでは、「スマートコーディング」を利用することで、簡単に操作式を作成することができます。

①演算ブロックを配置後に、 をクリックします。



②ファンクションを選択し、挿入します。



※スマートコーディングを使った操作式の一覧は付-5をご参照ください。



3. 比較演算回路

比較演算命令は、2つのデータの大小比較(=、<>、>、<、>=、<=)を行い、条件が成立するとONになるa接点として動作します。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)作成手順

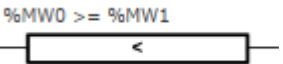
比較を行うには、比較ブロックをラングに配置し、操作式を入力します。



【比較ブロック】  
比較ブロックを配置すると、比較式を入力することができます。  
条件が成立するとONになるa接点として動作します。

例 「%MW0」と「%MW1」を比較する。

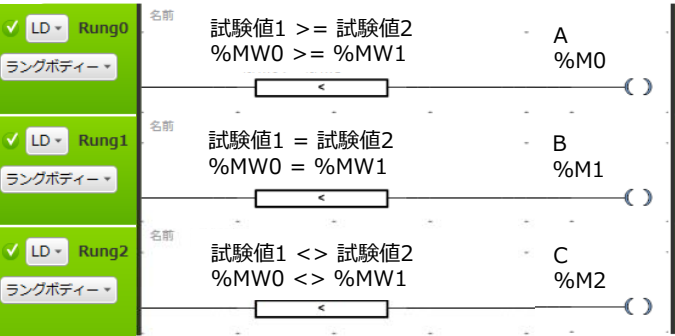
%MW0  $\frac{\text{半角スペース}}{\text{半角スペース}}$  >  $\frac{\text{半角スペース}}{\text{半角スペース}}$  %MW1



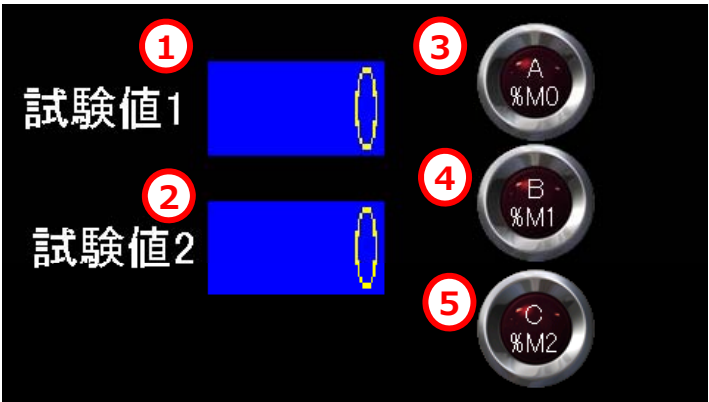
(2)作成例

- 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ワードアドレス「%MW0」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ワードアドレス「%MW1」の値が更新されます。
- 「%MW0」>=「%MW1」なら③ランプのビットアドレス「%M0」がONします。
- 「%MW0」=「%MW1」なら④ランプのビットアドレス「%M1」がONします。
- 「%MW0」≠「%MW1」なら⑤ランプのビットアドレス「%M2」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】



【使用するアドレス例】


種類	アドレス(メモリ)	シンボル名
ビットメモリ	%M0	A
ビットメモリ	%M1	B
ビットメモリ	%M2	C
ワードメモリ	%MW0	試験値1
ワードメモリ	%MW1	試験値2

- ①データ表示器（試験値 1）  
ワードアドレス： %MW0  
入力許可：チェックあり
- ②データ表示器（試験値 2）  
ワードアドレス： %MW1  
入力許可：チェックあり
- ③ランプ（銘板：A）  
・ビットアドレス： %M0
- ④ランプ（銘板：B）  
・ビットアドレス： %M1
- ⑤ランプ（銘板：C）  
・ビットアドレス： %M2

※上記以外の設定は任意です。





(3)動作確認


シミュレーターを使えば、パソコン上で動作確認できます。

ステータスエリアの  をクリックします。  
(シミュレーターモードに切り替わります)

**CHECK!**

シミュレーターには便利な  
ショートカットキーも用意しています。

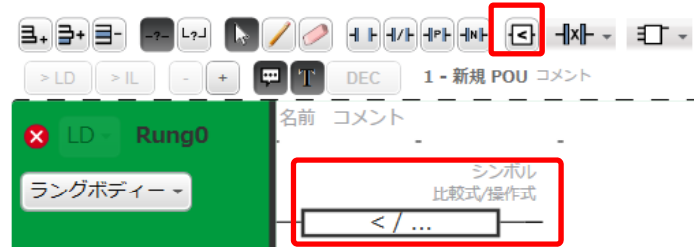
-  + B シミュレーター起動
-  + W シミュレーター停止
-  + L コントローラー起動
-  + M コントローラー停止

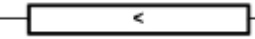

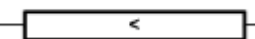
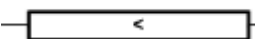
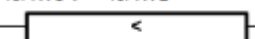
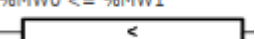
ステータスエリアの  をクリックします。  
(シミュレーターが開始されます)

**CHECK!**

比較ブロック

比較ブロックを使えば、ラダープログラム上で2つのデータの大小比較(=、<>、>、<、>=、<=)を行い、  
条件が成立するとONになるa接点として動作します。条件が成立しない場合はOFFとなり接点はOFFのままです。



比較演算子	式(サンプル)	図(サンプル)	機能
=	%MW0 = %MW1		「%MW0」と「%MW1」が比較され、比較の結果「%MW0」=「%MW1」なら導通します。
<>	%MW0 <> %MW1		「%MW0」と「%MW1」が比較され、比較の結果「%MW0」≠「%MW1」なら導通します。
>	%MW0 > %MW1		「%MW0」と「%MW1」が比較され、比較の結果「%MW0」>「%MW1」なら導通します。
<	%MW0 < %MW1		「%MW0」と「%MW1」が比較され、比較の結果「%MW0」<「%MW1」なら導通します。
>=	%MW0 >= %MW1		「%MW0」と「%MW1」が比較され、比較の結果「%MW0」>=「%MW1」なら導通します。
<=	%MW0 <= %MW1		「%MW0」と「%MW1」が比較され、比較の結果「%MW0」<=「%MW1」なら導通します。

# 第4章

## ライン操作盤をつくろう

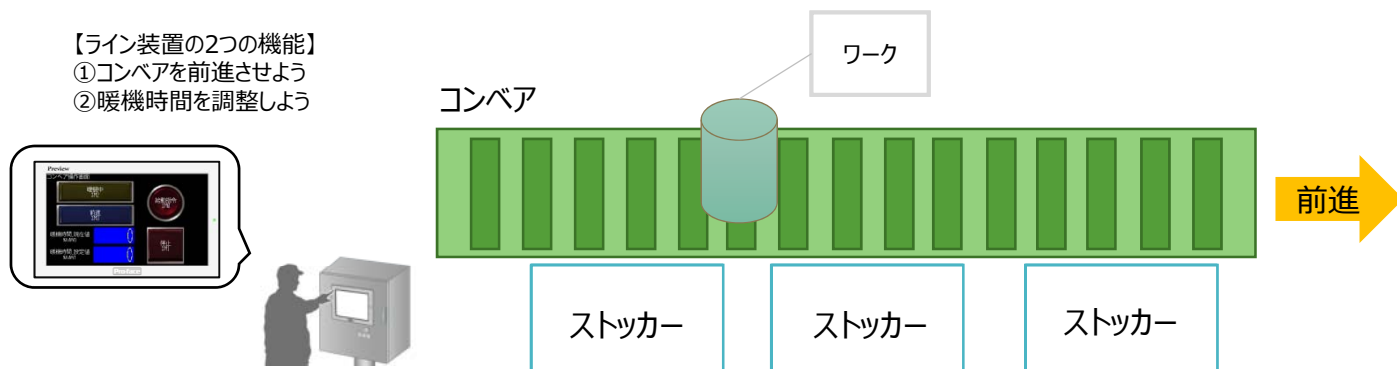
I . 実習について.....	4-2
1. ライン操作盤とは	
II . 実習の解説.....	4-3
1. 【設定】SoMachine Basic(M221シリーズ)	
2. 【プログラミング】SoMachine Basic(M221シリーズ)	
3. 【通信】SoMachine Basic(M221シリーズ)	
4. 【システム設定】GP-Pro EX(Pro-face製表示器)	
5. 【作画】GP-Pro EX(Pro-face製表示器)	
6. 【シミュレーション】GP-Pro EX(Pro-face製表示器)	
7. 【プロジェクト転送】GP-Pro EX(Pro-face製表示器)	
8. 動作確認しよう	



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、SoMachine Basic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2つに分け、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。



### 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

- ・「始動指令」スイッチをタッチします。
- ・「暖機中」ランプが3秒間点灯します（3秒間暖機を行います）
- ・「暖機中」ランプが消灯すると、「前進」ランプが点灯します。（コンベアが前進します）
- ・「停止」スイッチをタッチすると「暖機中」「前進」が消灯します（暖機または前進を即座に停止します）

##### ②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暖機時間(現在値)を表示します。
- ・表示器画面上のデータ表示器を使って暖機時間(設定値)を変更します。

#### (2)システム環境（実習環境）

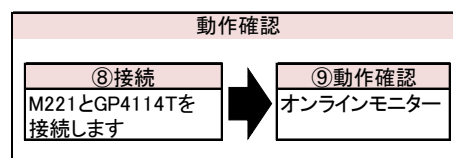
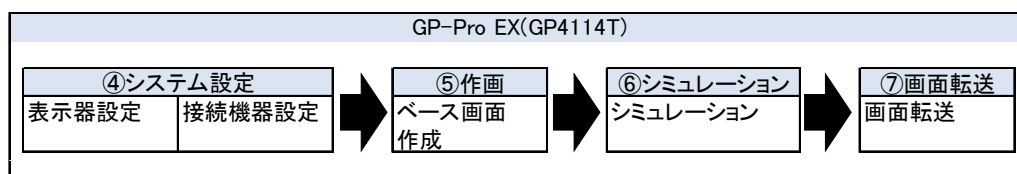
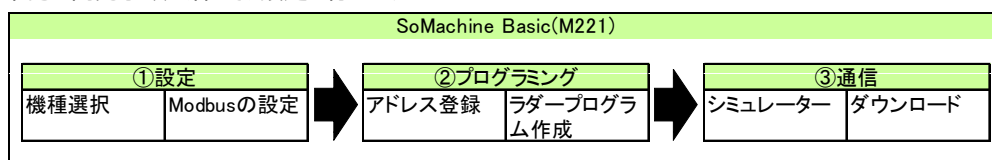
ここでは、以下の環境を例にシステム設定を行います。

【PLC】  
シリーズ：Modicon M221  
機種：TM221CE16R  
IPアドレス：192.168.1.20

【Pro-face製表示器】  
製品名：GP-4114T(PFXGP4114T2D)  
接続機器：[Schneider Electric SA]Modbus TCP マスタ  
IPアドレス：192.168.1.100

#### (3)開発手順

下記の開発手順に沿って、設定を行います。



## II. 実習の解説

## 1. 【設定】SoMachine Basic(M221シリーズ)

最初に、使用する機器や接続する機器等のハードウェアに関する設定を行います。

## (1)機種選択

【SoMachine Basicの起動】

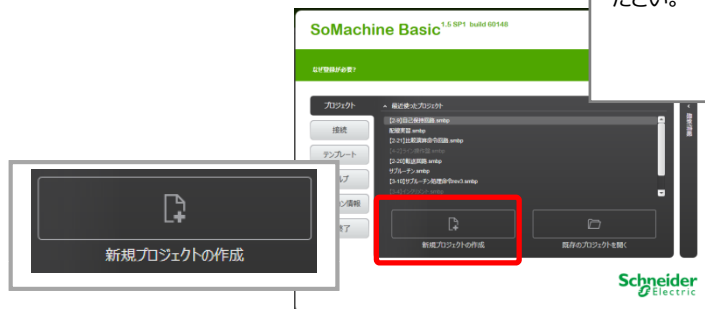
①デスクトップのSoMachine Basicのアイコンをダブルクリックします。



CHECK!

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新規プロジェクトの作成をクリックします。



【ハードウェアの設定】

③「設定」をクリックします。

【コントローラーの指定】

④右端の一覧から、TM221CE16Rをドラッグします。

コントローラー：TM221CE16R



⑤「確認」ウィンドウが表示されたら、「はい」をクリックします。



## (2)Modbusの設定

【IPアドレスの設定】

④左側から「ETH1」をクリックします。

⑤下記のようにM221自身のIPアドレスを指定します。

IPアドレス：192.168.1.20  
サブネットマスク：255.255.255.0

⑥「Modbusサーバーを有効」にチェックを入れます。

⑦「適用」をクリックします。



## 2. 【プログラミング】SoMachine Basic(M221シリーズ)

### (1) アドレス登録

ここでは、実習で使用するアドレスにシンボル名を付けます。

【使用するアドレス】

種類	アドレス	シンボル名	タイプ	時間ベース	プリセット
ビットメモリ	%M0	始動命令			
ビットメモリ	%M1	停止			
ビットメモリ	%M2	準備中			
ビットメモリ	%M3	前進			
ワードメモリ	%MW0	暖機時間_現在値			
ワードメモリ	%MW1	暖機時間_設定値			
タイマーメモリ	%TM0	準備タイマー	TON	1ms	3000

【プログラミングツリー】

①「プログラミング」をクリックします。

②「プログラミングツリー」の「ツール」をクリックします。



【ビットメモリ】

③「プログラミングツリー」の「メモリーオブジェクト」  
「ビットメモリ」をクリックします。

右図を参考に、シンボル名を登録します。  
右下の「適用」をクリックします。

ビットメモリ プロパティ

使用	アドレス	シンボル	コメント
<input type="checkbox"/>	%M0	始動指令	
<input type="checkbox"/>	%M1	停止	
<input type="checkbox"/>	%M2	準備中	
<input type="checkbox"/>	%M3	前進	
<input type="checkbox"/>	%M4		

【ワードメモリ】

④プログラミングツリーの「メモリーオブジェクト」  
「ワードメモリ」をクリックします。

右図を参考に、シンボル名を登録します。  
右下の「適用」をクリックします。

ワードメモリ プロパティ

使用	重複使用	アドレス	シンボル
<input type="checkbox"/>	<input type="checkbox"/>	%MW0	暖機時間_現在値
<input type="checkbox"/>	<input type="checkbox"/>	%MW1	暖機時間_設定値
<input type="checkbox"/>	<input type="checkbox"/>	%MW2	

【タイマー】

⑤プログラミングツリーの「ソフトウェアオブジェクト」  
「タイマー」をクリックします。

右図を参考に、各種設定を行います。  
右下の「適用」をクリックします。

タイマー プロパティ

使用	アドレス	シンボル	タイプ	保持	時間ベース	プリセット	コメント
<input type="checkbox"/>	%TM0	準備タイマー	TON	<input type="checkbox"/>	1 ms	3000	
<input type="checkbox"/>	%TM1		TON	<input type="checkbox"/>	1 ms	3000	

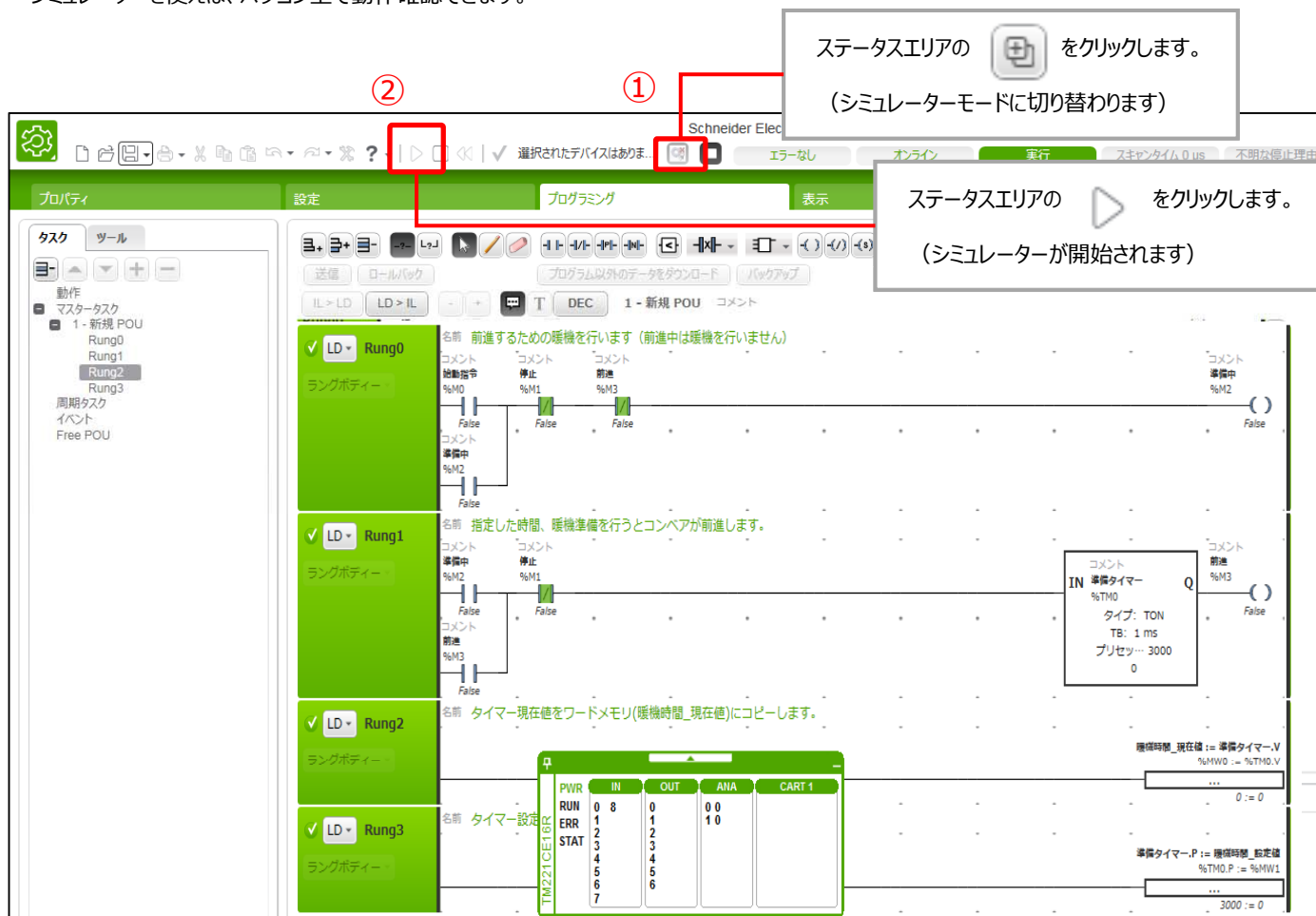






### 3. 【通信】SoMachine Basic(M221シリーズ)

#### (1) シミュレーター

シミュレーターを使えば、パソコン上で動作確認できます。



① ステータスエリアの  をクリックします。  
(シミュレーターモードに切り替わります)

② ステータスエリアの  をクリックします。  
(シミュレーターが開始されます。)

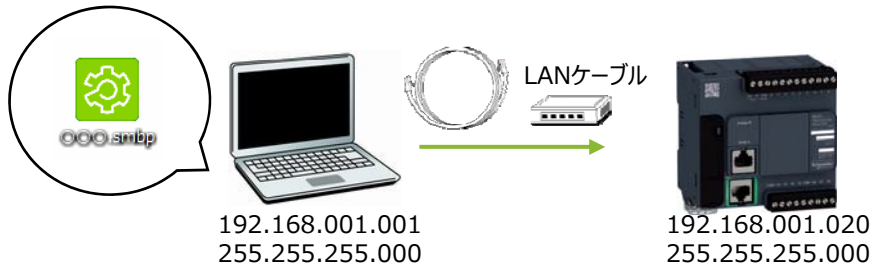
③ シミュレーターを使い、パソコン上で動作確認します。



## (2)プロジェクトのダウンロード

作成したプロジェクトをM221にダウンロードします。

※ここでは、LANケーブルを使ったダウンロード方法を解説します。

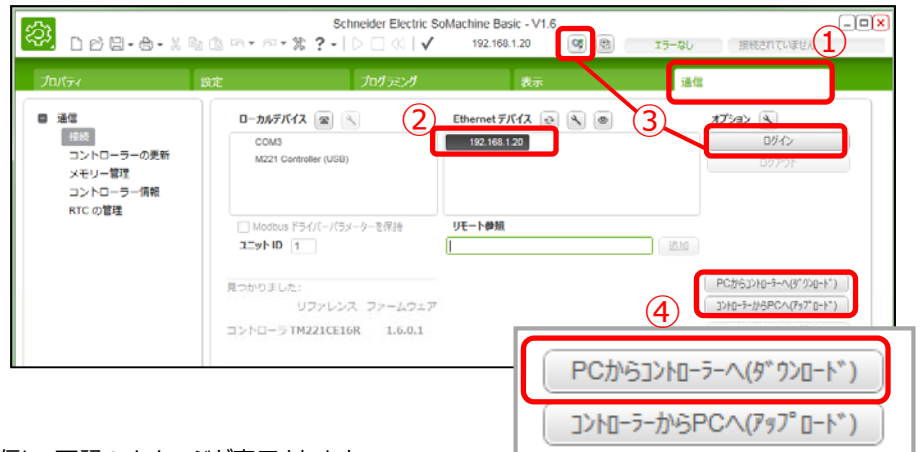


①「通信」をクリックします。

②「192.168.1.20」をクリックします。  
※M221とPCを接続するデバイス (方法)を指定します。

USBケーブルで転送する場合は、  
M221controller (USB)をクリックします。

③「ログイン」をクリックします。



※ログインすれば、M221とパソコンが通信し、下記のメッセージが表示されます。

⚠ PC とコントローラーのアプリケーションが違います  
PC とコントローラーのアプリケーションを比較

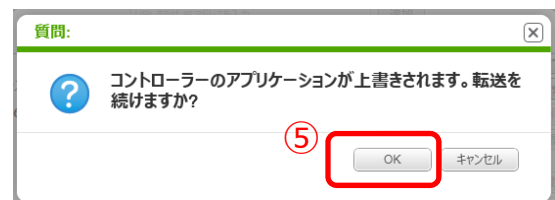
M221本体のラダープログラムとSoMachine Basicで設定しているラダープログラムに相違がある場合

✓ PC とコントローラー アプリケーションは同一です  
接続が確立されました

M221本体のラダープログラムとSoMachine Basicで設定しているラダープログラムが同じ場合

④「PCからコントローラへ」(ダウンロード)をクリックします。

⑤質問は「OK」をクリックします。



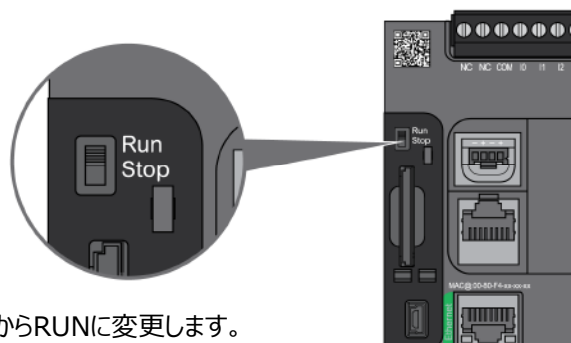
⑥正しく転送が完了すれば、下記のメッセージが表示されます。

✓ PC とコントローラー アプリケーションは同一です  
接続が確立されました



### M221本体でプログラムをRUNさせるには

M221シリーズには運転/ 停止ハードウェアスイッチが備わっており、コントローラを運転または停止状態にすることができます。



STOPからRUNに変更します。

## 4. 【システム設定】GP-Pro EX(Pro-face製表示器)

ここでは、表示器の画面を作成します。

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）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。



①コンベアを前進させよう

- ・「始動指令」スイッチをタッチします。
- ・「暖機中」ランプが3秒間点灯します（3秒間暖機を行います）
- ・「暖機中」ランプが消灯すると、「前進」ランプが点灯します（コンベアが前進します）
- ・「停止」スイッチをタッチすると「暖機中」「前進」が消灯します。（暖機または前進を即座に停止します）

②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暖機時間(現在値)を表示します。
- ・表示器画面上のデータ表示器を使って暖機時間(設定値)を変更します。

### (1)表示器設定

【GP-Pro EXの起動】

GP-Pro EXを起動し、プロジェクトファイルを作成します。

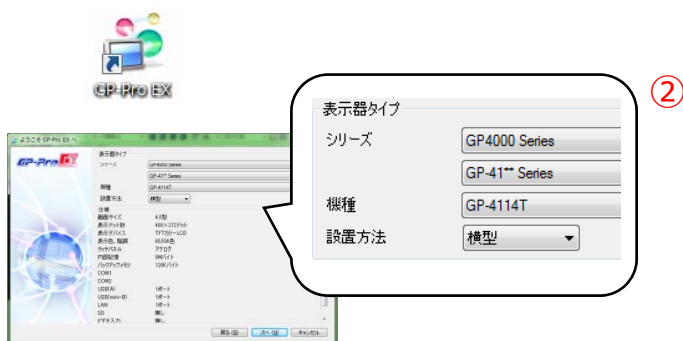
①デスクトップのGP-Pro EXのアイコンをダブルクリックします。

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。

【Pro-face製表示器】

製品名：GP-4114T(PFXGP4114T2D)



### (2)接続機器設定

③接続機器設定

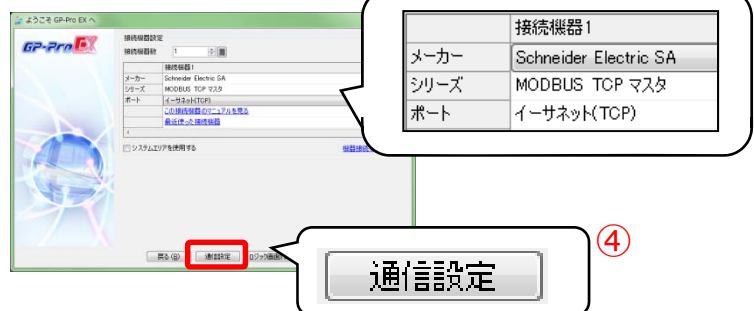
接続機器を指定します。

ここでは、M221と「Modbus-IDA」で接続します。

メーカー：[Schneider Electric SA]


シリーズ：Modbus TCP マスタ

ポート：イーサネット(TCP)



④「通信設定」をクリックします。

⑤「接続機器1」をクリックします。

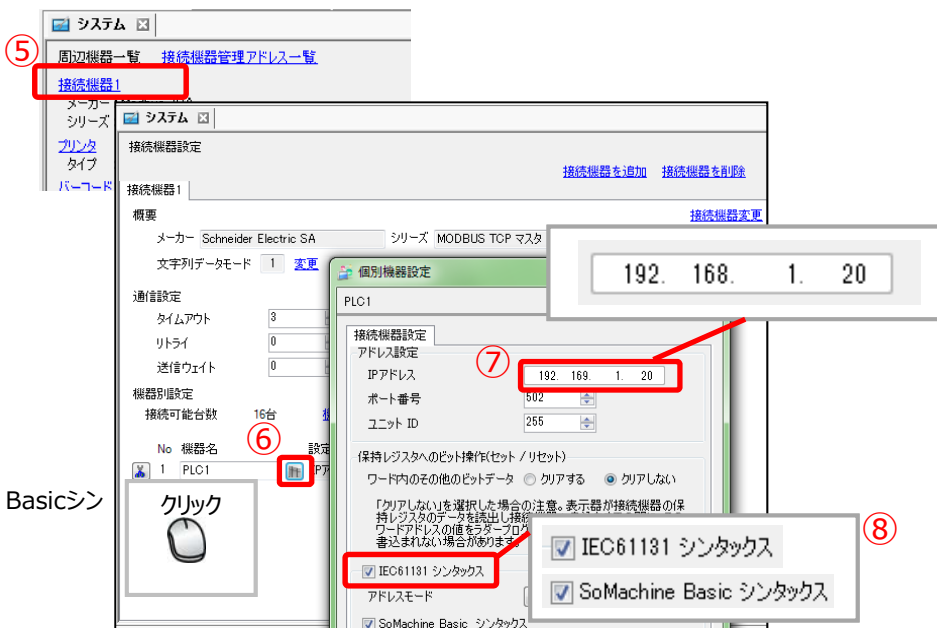
⑥  をクリックします。

⑦ IPアドレスを入力します。

※接続機器側(M221)のIPアドレスを入力します。

192. 168. 1. 20

⑧「IEC61131シンタックス」「SoMachine Basicシンタックス」にチェックを入れます。



## 5. 【作画】GP-Pro EX(Pro-face製表示器)

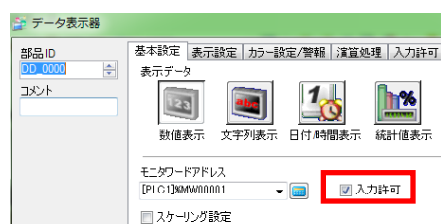
### (1)ベース画面作成

ここではコンベア操作画面で、操作のスイッチとランプ、データ表示器を使ってベース画面作成します。  
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】



※上記以外の設定は任意です。



⑥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)



**CHECK!**

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。

**CHECK!**

### 「IEC61131 シンタックス」「SoMachine Basic シンタックス」

「IEC61131シンタックス」「SoMachine Basic シンタックス」にチェックを入れると、アドレスの表記がSoMachine Basicと同じ表記になり、設定がわかりやすくなります。

☐ IEC61131 シンタックス  
☐ SoMachine Basic シンタックス

☒ IEC61131 シンタックス  
☒ SoMachine Basic シンタックス



### 【 SoMachine Basicのアドレス表記例】

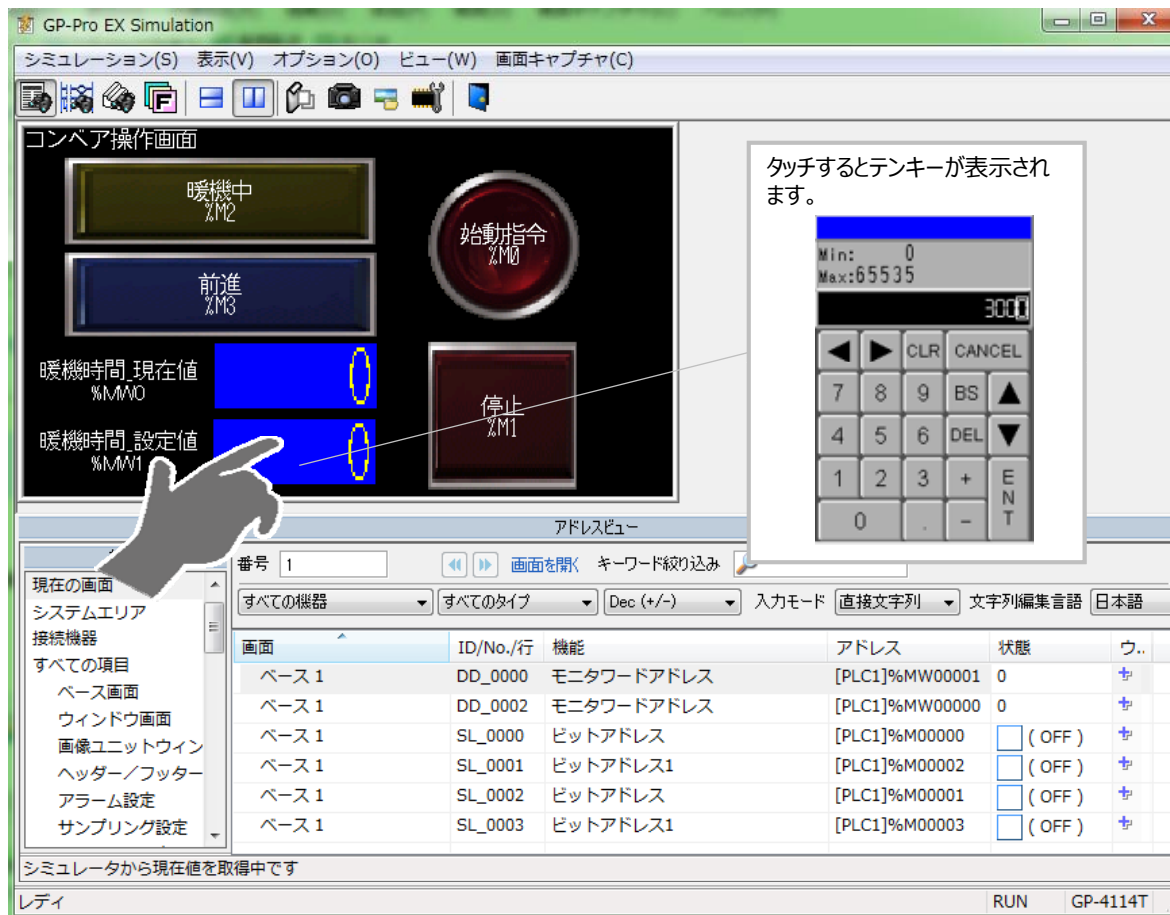
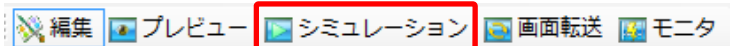
ビットメモリー プロパティ			
使用	アドレス	シンボル	コメント
<input type="checkbox"/>	%M0	ワードメモリー プロパティ	
<input type="checkbox"/>	%M1		
<input type="checkbox"/>	%M2		
<input type="checkbox"/>	%M3		
<input type="checkbox"/>	%M4		
<input type="checkbox"/>	%MW0		
<input type="checkbox"/>	%MW1		
<input type="checkbox"/>	%MW2		

## 6. 【シミュレーション】GP-Pro EX(Pro-face製表示器)

### (1)シミュレーションの起動

シミュレーションを使えば、Pro-face製表示器の動作をパソコン上で確認できます。  
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。）

## 7. 【プロジェクト転送】GP-Pro EX(Pro-face製表示器)

作成したプロジェクトをPro-face製表示器に送信（転送）します。

○○.prxファイル



### (1) 転送の種類

プロジェクトの転送方法は3つあります。



USBケーブル  
(A-MiniB)

オプションのUSBデータ転送ケーブル（ZC9USCBMB1）または、市販のUSBケーブル(Type A to mini 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。



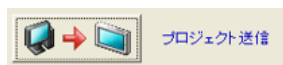
USBストレージ

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。

### (2) プロジェクト転送

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。



※必要に応じて、下記の転送設定を行ってください。

**CHECK!**

#### 転送設定

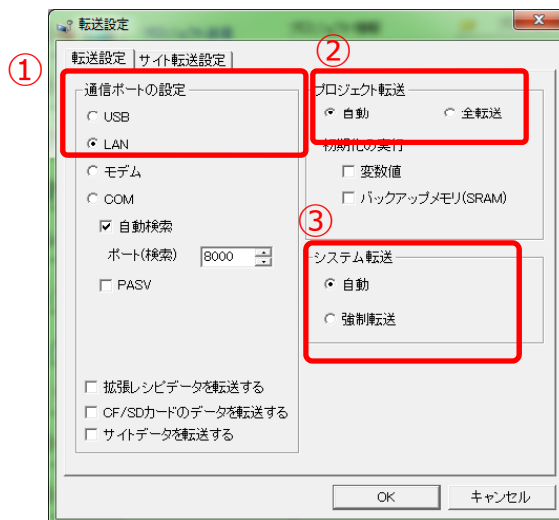
転送前に初期設定を行います（初期設定は、一度行くと記憶されます）

① 転送方法にあわせて、通信ポートを選択します。

USB : USBケーブルでの転送  
LAN : LANケーブルでの転送

② 自動的に変更画面を判別し転送するか、全部の画面を転送するかを選択します。  
※原則的には「自動」を選択してください。

③ システムを自動転送するか、強制転送するかを選択します。  
※原則的には「自動」を選択してください。





## 8. 動作確認しよう

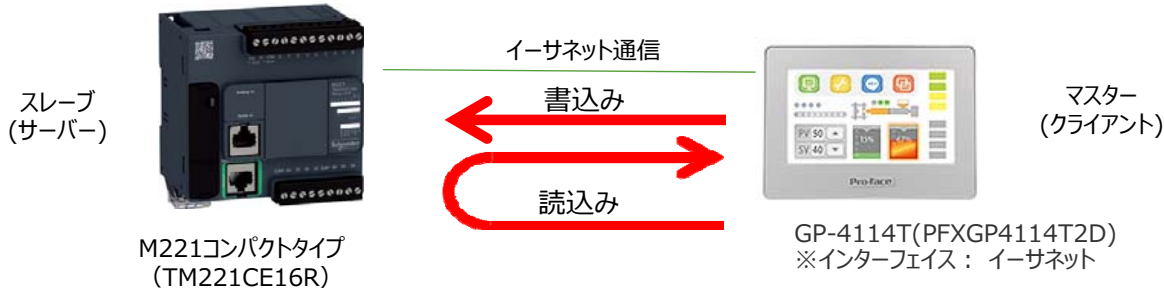
### (1) 接続

M221とPro-face製表示器をLANケーブルを使って接続します。

[Schneider Electric SA] のModbusプロトコルは、シリアル接続(Modbus)とイーサネット接続(Modbus TCP)があり、Pro-face製表示器とはどちらを使っても接続できます。ここでは、イーサネット接続(Modbus TCP)を使って接続します。

【(Modbus)イーサネット接続】

※本テキストでは、Pro-face製表示器をマスター(クライアント)、M221シリーズをスレーブ(サーバー)として説明します。



※イーサネット接続できるのは、M221のイーサネット組込み機種のみです

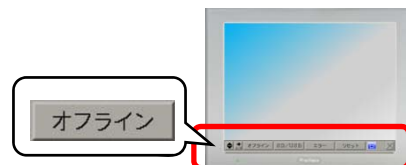


#### 表示器のイーサネット情報の設定(オフライン画面)

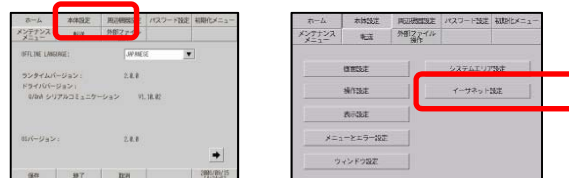
① 運転中の表示器画面の右上隅→左下隅 または 左上隅→右下隅(縦横40ドット以内)の順に0.5秒以内にタッチします。



② 画面にシステムメニューが表示されるので「オフライン」をタッチします。



③ オフラインのホーム画面が表示されるので「本体設定」「イーサネット設定」をタッチします。



④ IPアドレス入力枠をタッチすると、テンキーが表示されるので IPアドレスを入力します(サブネットマスクも同様に入力します)



## (2) オンラインモニター

### 【オンラインモニター】

運転中のM221シリーズ内部のラダープログラムをパソコン上でモニタすることができます。

デバイスアドレスのON/OFF状態や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。

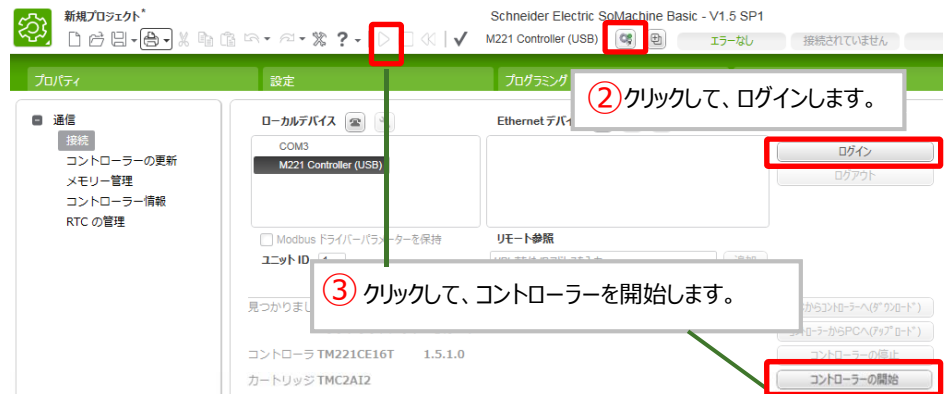


### 【オンラインモニター】

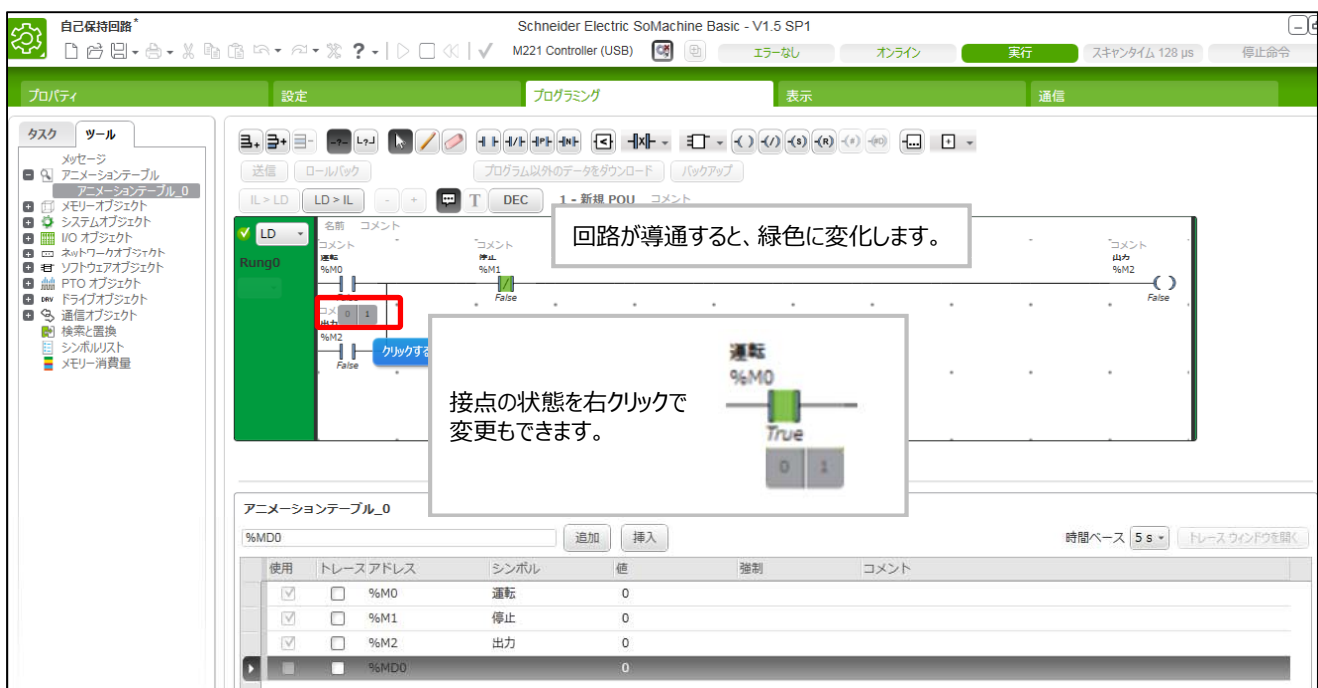
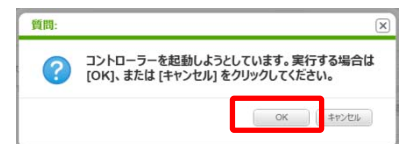
①「通信」をクリックします。

②ログインをクリックします。  
(ログインすることで、PCと接続されます)

③コントローラーの開始をクリックします。




④質問は、「OK」をクリックします。

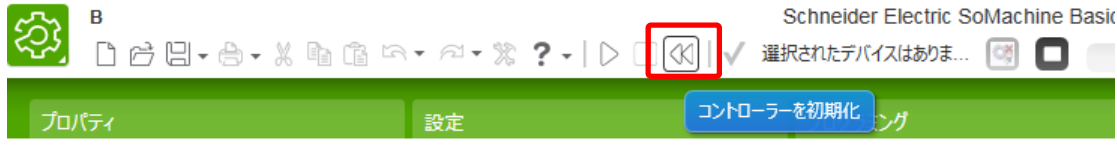




CHECK!

## コントローラの初期化

M221シリーズのデバイスアドレスの状態を初期するには、ステータスバーの  をクリックします。特にコントローラの初期化を行うと、強制ON/OFFしたデバイスの状態も初期状態に戻ります。



CHECK!

## アニメーションテーブル

アニメーションテーブルを使用すれば、デバッグ(シミュレーター/オンラインモニタ)時に、アドレスの現在値を変更し、動作確認を行うことができます。

① プログラミングツリーの「ツール」から新規アニメーションテーブルを追加します。

② アニメーションテーブルで、アドレスを追加します。

③ アドレスの値を変更することができます。

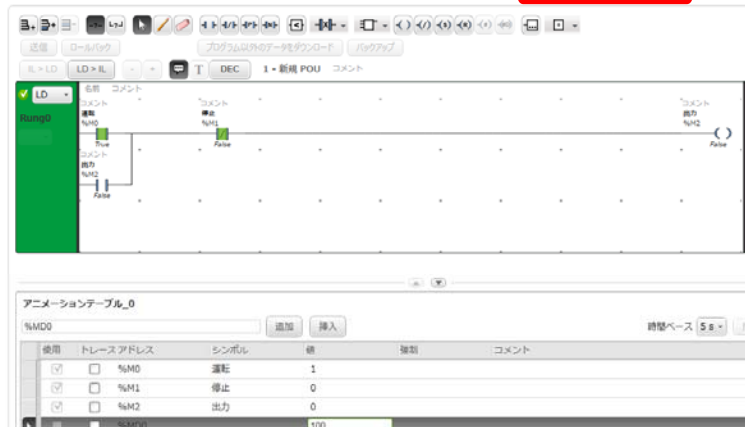


②



CHECK!

アニメーションテーブルは、次項のオンラインモニターでも活用できます。





メモ（以下は余白です。ご自由にお使いください。）

# 付録

## さらに便利に使おう

### I . SoMachine Basicのプログラミングツール

(1)オペレーションツール.....	付-2
(2)命令一覧.....	付-3
(3)スマートコーディング.....	付-5
(4)PID .....	付-7
(5)スケジュールブロック.....	付-8
(6)データロギング.....	付-9



メモ(以下は余白です。ご自由にお使いください。)

I . SoMachine Basicのプログラミングツール

SoMachine Basicには、効率よく作業を行うための様々なツールや工夫がされています。付録では、その中の一部をご紹介します。

(1)オペレーションツール

ラダープログラミングは、プログラミングワークスペース上部にあるアイコンを使用すると効率よく行うことができます。



【オペレーション用のアイコン】

図	名称	図	名称	図	名称
	新規ラングの追加		通常モード(分岐なし)		選択モード
	新規ラングの挿入		分岐モード		ラインの描画
	ラングの削除				ラインの消去
図	名称	図	名称	図	名称
	ILからラダーに変換		列を削除		コメントの表示/非表示
	ラダーからILに変換		列を追加		(ラダーエディターでは使用されていません)



PLCプログラミング言語の グローバルスタンダード「IEC 61131-3」

「IEC 61131-3」とは、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。

この規格では、日本で普及しているラダーダイアグラムを含む5種類の言語が規定されており、使用者のスキルや目的に合った言語が使うことができます。

SoMachine Basicは、「IEC 61131-3」に準拠していることにより、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)という特長があります。ラダー方式だけでなく、IL(Instruction List)やSFC( Sequential Function Chart )にも対応しています。

SoMachine Basicでは、LDとILを1クリックで切り替えることができます。



(2)命令一覧

SoMachine Basicのラダープログラムの命令はすべて、ツールバーから挿入します。



【接点】

図	名称	記号
	A接点	LD
	B接点	LDN
	立ち上がり接点	LDR
	立ち下がり接点	LDF

【出力(コイル)】

図	名前	記号
	コイル	ST
	反転コイル	STN
	セットコイル	S
	リセットコイル	R

【オペレーション(転送/演算)】

図	名称	図	名称	式(サンプル)
	演算ブロック		転送(コピー)	%MW1 := %MW0
			加算演算	%MW2 := %MW0 + %MW1
			減算演算	%MW2 := %MW0 - %MW1
			乗算演算	%MW2 := %MW0 * %MW1
			除算演算	%MW2 := %MW0 / %MW1
			インクリメント	INC %MW0
			デクリメント	DEC %MW0

【比較】

図	名称	図	比較式
	比較ブロック		=
			>
			<
			<>
			<=
			>=

【ブール操作】

図	名称	図	演算子
	XOR 命令		XOR
			XORN
			XORR
			XORF















【グラフセ命令】

図	名前	記号
	グラフセステップ有効化	#
	グラフセステップ無効化	#D

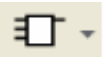






【その他の命令】

図	名称	図	名称
	他のラダー項目		立上がり接点 ファンクション
			立下がり接点 ファンクション
			Not
			開く
			Short
			プログラムの終端
			CNの終了
			ジャンプ
			ジャンプCN
			IF
			ELSE
			ENDIF
			FOR
			ENDFOR


【ファンクション】

図	名称	図	名称
	ファンクション		タイマー
			LIFO/FIFOレジスター
			ビットレジスターをシフト
			ステップカウンター
			カウンター
			高速カウンター(FC)
			高速カウンター(HSC)
			ドラム
			RTC
			パルス
			PLS FREQGEN
			パルス幅変調
			データロギング
			メッセージ

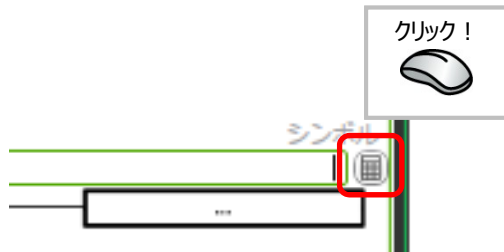
【ファンクション】

図	名称	図	名称	図	名称
	ファンクション		PTO		Motion
					MC_MotionTask_PTO
					MC_Power_PTO
					MC_MoveVel_PTO
					MC_MoveRel_PTO
					MC_MoveAbs_PTO
					MC_Home_PTO
					MC_SetPos_PTO
					MC_Stop_PTO
					MC_Halt_PTO
					Administrative
					MC_ReadActVel_PTO
					MC_ReadActPos_PTO
					MC_ReadSts_PTO
					MC_ReadMotionState_PTO
					MC_ReadAxisError_PTO
					MC_Reset_PTO
					MC_TouchProbe_PTO
					MC_AbortTrigger_PTO
					MC_ReadPar_PTO
					MC_WritePar_PTO
			ドライブオブジェクト		MC_Power_ATV
					MC_Jog_ATV
					MC_MoveVel_ATV
					MC_Stop_ATV
					MC_ReadStatus_ATV
					MC_ReadMotionState_ATV
			通信		MC_Reset_ATV
					Read Var
					Write Var
					Write Read Var
			ユーザー定義 ファンクションブロック		Send Receive Message
					Send Receive SMS

### (3)スマートコーディング

スマートコーディングは、オペレーションブロック(演算ブロック)の操作式に、表示される  をクリックすると表示されます。スマートコーディングを使用すると、命令(関数)を簡単に挿入することができます。

また、スマートコーディングは、命令をカテゴリに分けて、登録されています。



【スマートコーディングの命令(関数)】

カテゴリ	名称	内容
ASCII	ASCII_TO_FLOAT	ASCII文字列を浮動小数点に変換
	ASCII_TO_INT	ASCII文字列を整数値に変換
	FLOAT_TO_ASCII	浮動小数点をASCII文字列に変換
	INT_TO_ASCII	整数をASCII文字列に変換
	ROUND	ASCII文字列(1≤n≤8)に格納されている浮動小数点を四捨五入

カテゴリ	名称	内容
数値処理	ABS	オペランドの絶対値
	BTI	BCDから2進数に変換
	CONCATW	2つのワードを連結(ダブルワード)
	DEC	デクリメントオペランド
	DWORD	16ビットワードを32ビットのダブルワードに変換
	HW	ダブルワードのMSBをワードへ書出し
	INC	インクリメントオペランド
	ITB	2進数からBCDへの変換
	LW	ダブルワードのLSBをワードに抽出
	NOT	オペランドの論理補数(ビット単位)
	ROL	左へn位置ローテートシフト1≤n≤16(ワード)または1≤n≤32(ダブル)
	ROR	右へn位置ローテートシフト1≤n≤16(ワード)または1≤n≤32(ダブル)
	SHL	左へn位置論理シフト1≤n≤16(ワード)または1≤n≤32(ダブル)
	SHR	右へn位置論理シフト1≤n≤16(ワード)または1≤n≤32(ダブル)
	SQRT	オペランドの平方根

カテゴリ	名称	内容
テーブル	EQUAL_ARR	2テーブルの要素ごとの比較を行います。相違がある場合、最初の相違要素の順位がワードで返されます。それ以外の場合の戻り値は-1(テーブルの最大長:225)
	FIND_EQR	最初の要素が指定された値と等しいダブルまたはフロートワードテーブルの位置の検索
	FIND_GTR	最初の要素が指定された値より大きいダブルまたはフロートワードテーブルの位置の検索
	FIND_LTR	最初の要素が指定された値より小さいダブルまたはフロートワードテーブルの位置の検索
	LKUP	LKUPファンクションは与えられたX軸に対して、X対Yの浮動小数点データを補間
	MAX_ARR	ダブルワードまたはフロートワードテーブルの最大値の検索
	MEAN	MEANファンクションは浮動小数点テーブルの値から平均または平均値を計算
	MIN_ARR	ダブルワードまたはフロートワードテーブルの最小値の検索
	OCCUR_ARR	ダブルワードまたはフロートワードテーブルから、与えられた値と等しい要素を検索
	ROL_ARR	ダブルワードまたはフロートワードのテーブルの要素をn位置に、上から下にローテートシフトを実行
	ROR_ARR	ダブルワードまたはフロートワードのテーブルの要素をn位置に、下から上にローテートシフトを実行
	SOFT_ARR	ダブルワードまたはフロートワードのテーブルの要素順序を昇順または降順でソートして、同じテーブルに結果を保存
	SUM_ARR	オブジェクトテーブルのすべての要素を追加(テーブルの最大長:255)



カテゴリ	名称	内容
浮動小数点	ABS	オペランドの絶対値
	ACOS	アークコサイン(0と $\pi$ の間)
	ASIN	アークコサイン( $-\pi/2$ と $\pi/2$ の間)
	ATAN	アークタンジェント( $-\pi/2$ と $\pi/2$ の間)
	COS	ラジアン表記の角度のコサイン
	DEG_TO_RAD	度からラジアンへの変換。結果は0と $2\pi$ の間
	DINT_TO_REAL	ダブルワード(整数)からフロートへの変換
	EXP	自然指数
	EXPT	整数のリアル乗
	INT_TO_REAL	整数ワードからフロートへの変換
	LN	自然対数
	LOG	10の指数
	RAD_TO_DEG	ラジアン表記の角度の変換結果はから360度の間
	REAL_TO_DINT	浮動小数点を整数ダブルワードに変換(結果に最も近い整数値に四捨五入されます)
	REAL_TO_INT	浮動小数点を整数ワードに変換(結果に最も近い整数値に四捨五入されます)
	SIN	ラジアン表記の角度のサイン
	SQRT	オペランドの平方根
	TAN	ラジアン表記の角度のタンジェント
	TRUNC	全体の浮動小数点の値

カテゴリ	名称	内容
I/Oオブジェクト	READ_IMM	タスクの実行中に関数ブロックのパラメータを読み込み、同じサイクル中に入力イメージを更新
	READ_IMM_IN	タスクの実行中に内蔵のデジタル入力を読み込み、入力イメージをすぐに更新
	WRITE_IMM	ファンクションブロックパラメータをタスクの実行中に書き込みます。出力イメージを同じサイクル中に更新
	WRITE_IMM_OUT	内蔵されたデジタル出力にすぐに書き込まれます。値は出力イメージから読み込み

カテゴリ	名称	内容
通信	EXCH1	交換命令はロジックコントローラを使った送受信を、ASCIIまたはModbusデバイスと実行(SL1)
	EXCH2	交換命令はロジックコントローラを使った送受信を、ASCIIまたはModbusデバイスと実行(SL2)
	EXCH3	交換命令はロジックコントローラを使った送受信を、ASCIIまたはModbusデバイスと実行(ETH)

カテゴリ	名称	内容
PID	PID	PIDファンクションはアナログ計測とセットポイントを利用して「Idcorrection」を実行し、同じ形式のアナログ命令またはデジタル出力のPWMのいずれかを生成

カテゴリ	名称	内容
ユーザー定義 ファンクション		ユーザーが作成したファンクションを挿入


#### (4)PID

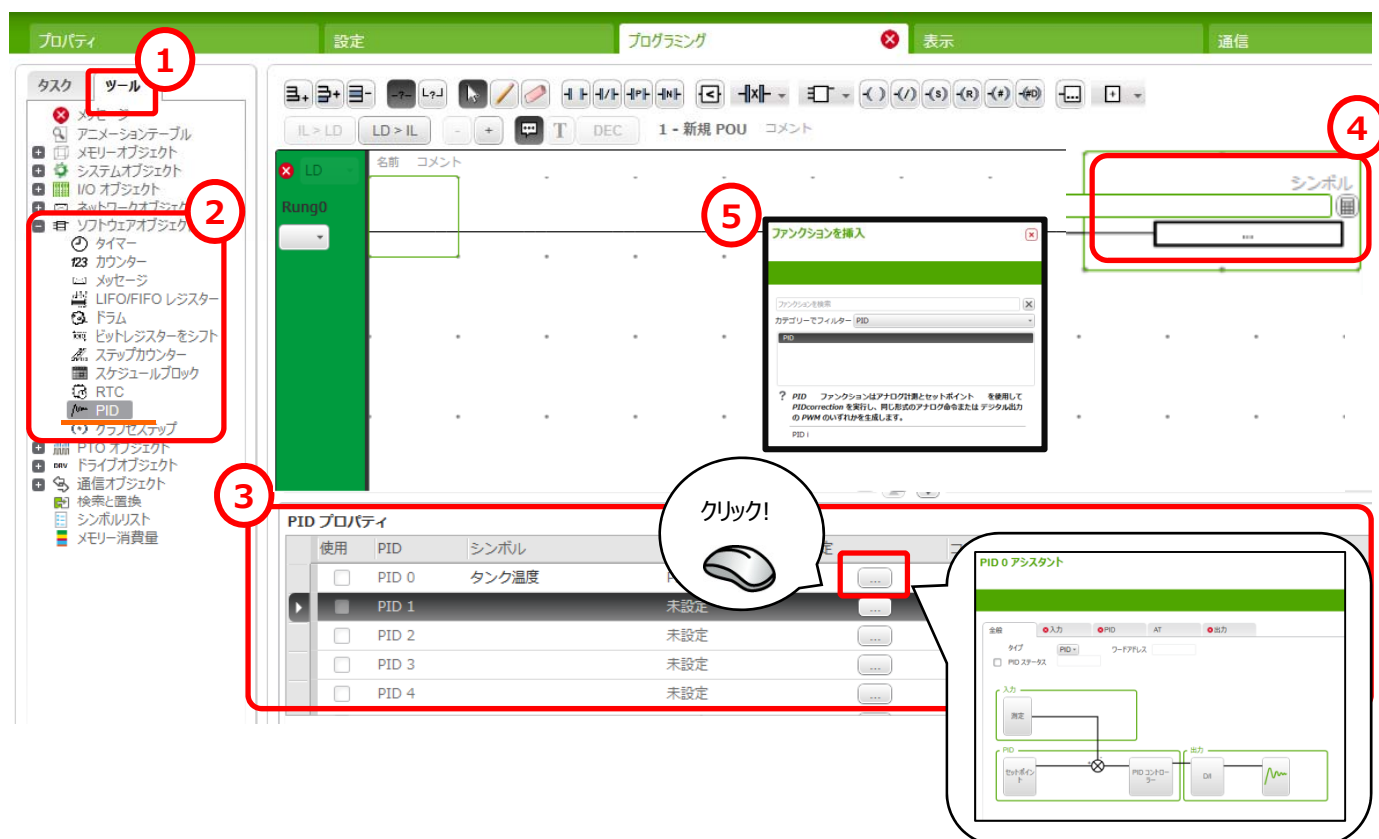
温度制御をはじめとしたフィードバック制御の一種であるPID制御を行うことができます。設定できるPIDのタイプは「PID」「AT+PID」「AT」「ワードアドレス」の4つです。

### 【設定方法】PIDのプロパティの設定

- ① 「プログラミングツリー」の「ツール」をクリックします。
- ② 「ソフトウェアオブジェクト」から「PID」をクリックします。
- ③ 「PIDプロパティ」で、各種設定を行います。  
「PIDアシスタント」での設定も行います。

【設定方法】ラダープログラムでPIDを使用するには

- ④ ラング内に演算ブロックを配置し、 をクリックします。
- ⑤ 「スマートアシスタント」から「PID」を挿入します。
- ⑥ 関連付けるPID番号(PIDプロパティ参照)を入力します。



タイプ	内容
PID	<p>PIDコントローラの起動時にシンプルPIDコントローラをデフォルトで起動します。プロセスを正常に制御するには、PIDタブで指定するゲイン値Kp、Ti、Tdを事前に把握しておく必要があります。</p> <p>PIDアシスタント画面のPIDタブでコントローラ（PIDまたはPI）の補正タイプを選ぶことができます。PI補正タイプが選択されると、「微分時間Tdフィールド」は無効になります。PIDモードを使用すると、オートチューニング機能は無効になり、アシスタント設定画面のATタブは使用できなくなります。</p>
AT + PID	<p>このモードでは、PIDコントローラの起動時にオートチューニング機能が有効になります。オートチューニング関数は、ゲイン値Kp、Ti、TdとPIDアクションのタイプを計算します。オートチューニングによって計算されたパラメータを使用し、オートチューニングシーケンスの終了時に、コントローラは調整された設定値でPIDモードに切り替わります。</p>
AT	<p>このモードでは、PIDコントローラの起動時にオートチューニング機能がアクティブになり、自動的にゲイン値Kp、Ti、TdとPIDアクションのタイプを計算します。</p>
ワードアドレス	<p>このPIDモードは、指定したワードアドレスに下記の値を割り当てることによってモードが選択されます。</p> <hr/> <div style="display: flex; justify-content: space-between;"> <span>0：コントローラは無効です。</span> <span>1：コントローラはシンプルPIDモードで動作します。</span> </div> <div style="display: flex; justify-content: space-between;"> <span>2：コントローラはAT + PIDモードで動作します。</span> <span>3：コントローラはATモードでのみ動作します。</span> </div> <div style="display: flex; justify-content: space-between;"> <span>4：コントローラはPI補正タイプのシンプルPIDモードで動作します。</span> </div>

## (5)スケジュールブロック

スケジュールブロックとは、設定した月/日/時間で、制御処理を行う機能です。

### 【設定方法】

- ①「プログラミングツリー」の「ツール」をクリックします。
- ②「ソフトウェアオブジェクト」から「スケジュールブロック」をクリックします。
- ③「スケジュールブロックプロパティ」で、各種設定を行います。  
設定内容の条件を満たすと、「出力ビット」がONします。

The screenshot shows the 'プログラミング' (Programming) window. On the left, the 'ツール' (Tools) menu is expanded, and 'スケジュールブロック' (Schedule Block) is selected. The 'スケジュールブロック プロパティ' (Schedule Block Properties) dialog is open, showing a table of schedule blocks.

使用	アドレス	設定済み	出力ビット	開始日	開始月	終了日	終了月	開始時間	終了時間	月	火	水	木	金	土	日	コメント
<input checked="" type="checkbox"/>	%SCH0	<input checked="" type="checkbox"/>	%M0	18	11月	25	12月	00:00	00:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	%SCH1	<input type="checkbox"/>		1	1月	1	1月	00:00	00:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	%SCH2	<input type="checkbox"/>		1	1月	1	1月	00:00	00:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	%SCH3	<input type="checkbox"/>		1	1月	1	1月	00:00	00:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	%SCH4	<input type="checkbox"/>		1	1月	1	1月	00:00	00:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	



メモ(以下は余白です。ご自由にお使いください。)

## (6) データロギング

SoMachine Basicでは、SDカードにデータを収集/保存(データロギング)を行うことができます。  
データロギングを使用すれば、ヒストリカルデータまたはイベントログをSDカード内にCSV形式で保存することができます。

ヒストリカルデータ：一度に最大32ワードのデータを収集することができます。

イベントログ：文字列データを収集することができます。RFID等の読み込み履歴の記録に便利です。



### 【作成手順】

データロギングの設定方法 ※データロギングを使用するには、下記2つの設定が必要です。

「データロギングプロパティ」：DATALOGアシスタントを使用し、収集するデータを指定します。

「データロギング命令」：ラングにデータロギング命令を配置します。

#### ① データロギングプロパティ

・「プログラムツリー」の「ツール」の「ソフトウェアオブジェクト」の「データロギング」をクリックします。

・設定 [...] をクリックし、DATALOGアシスタントで収集するデータを決めます。

**タスク ツール**

メッセージ

アニメーションテーブル

メモリオブジェクト

システムオブジェクト

I/O オブジェクト

ネットワークオブジェクト

ソフトウェアオブジェクト

タイマー

カウンター

メッセージ

LIFO/FIFO レジスタ

ドラム

ビットレジスタをシフト

ステップカウンター

スケジュールブロック

RTC

PID

**データロギング**

グラフセテップ

➡

**データロギング プロパティ**

使用	アドレス	シンボル	設定済み	設定	コメント
<input checked="" type="checkbox"/>	%DATALOG0		<input checked="" type="checkbox"/>	<div style="border: 2px solid red; padding: 2px;">...</div>	
<input type="checkbox"/>	%DATALOG1		<input type="checkbox"/>	...	
<input type="checkbox"/>	%DATALOG2		<input type="checkbox"/>	...	
<input type="checkbox"/>	%DATALOG3		<input type="checkbox"/>	...	

➡

**DATALOG0 アシスタント**

データログファイル: DATALOG0.csv    最大ファイルサイズ: 100 kB

☒ ヒストリカルデータ    ☐ イベントログ    ☒ タイムスタンプを追加

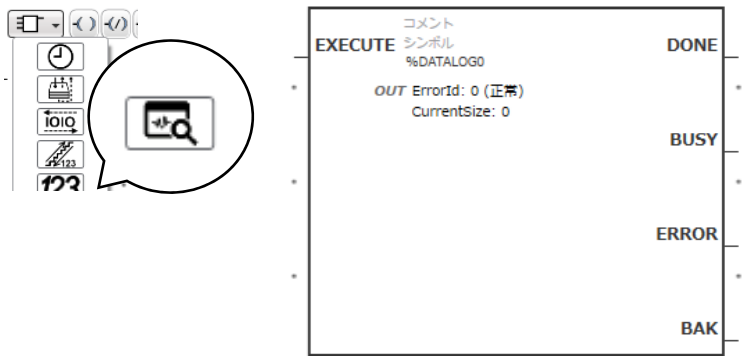
インデックス	オブジェクト	シンボル	コメント
0	%MW0	収集データ1	
1	%MW1	収集データ2	
2	%MW2	収集データ3	
3	%MW3	収集データ4	
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

適用    キャンセル

クリック!

収集するデータを指定します。

②データロギング命令  
ラングにデータロギング命令を配置します。



【データロギングアドレス(構造体アドレス)】  
データロギングアドレスは、それぞれの役割を持った要素アドレスの集まり(構造体アドレス)です。要素アドレスの役割は下記になります。

要素アドレス名	アドレスタイプ	役割
%DATALOG※.EXECUTE	ビットメモリ	データロギングのトリガーです。立上り時に一度データ収集を行います。
%DATALOG※.DONE	ビットメモリ	データ収集が正常に終了すれば、ONします。
%DATALOG※.BUSY	ビットメモリ	データ収集中、ONします。
%DATALOG※.ERROW	ビットメモリ	実行が正常に終了しなければ、ONします。
%DATALOG※.BAK	ビットメモリ	バックアップファイルがあれば、ONします。



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。)



発行 : 2018年9月(1.04)  
発行元 : Pro-faceトレーニングセンター  
(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)