

NJ/NX シリーズ ラダー言語プログラミングセミナー

セミナー資料

目次	
1. はじめに.....	3
1.1. セミナーの流れ.....	3
2. Sysmac Studio の起動.....	4
2.1. コントローラ NJ/NX/NY シリーズ用統合開発環境「Sysmac Studio」の起動.....	4
2.2. 新規プロジェクトを作成.....	4
2.3. プロジェクトのプロパティを入力.....	4
3. ユニット構成の設定.....	5
3.1. ユニット構成.....	5
3.2. CPU・増設ラック構成編集画面の起動.....	5
3.3. NX ユニットの登録.....	6
4. 入出力接点に変数を割付.....	10
4.1. I/O マップの表示.....	10
4.2. 入力接点に変数を割り付ける.....	10
4.3. 出力接点に変数を割り付ける.....	11
4.4. グローバル変数確認.....	11
5. プログラムの作成.....	12
5.1. POU 作成.....	12
5.2. プログラム作成.....	12
5.3. セクション追加.....	12
5.4. プログラム追加.....	13
5.5. プログラムのコンパイル.....	13
6. タスク割り付け.....	14
6.1. POU をタスクに割り付ける.....	14
7. オフラインデバッグ.....	15
7.1. シミュレーターの起動.....	15
7.2. ウォッチウィンドウを開く.....	15
7.3. ウォッチウィンドウに変数設定.....	16
7.4. 動作確認の実施.....	16
8. データトレース.....	17
8.1. データトレースとは.....	17
8.2. データトレースを開く.....	17
8.3. データトレース設定（トリガートレース）.....	17
8.4. データトレース開始.....	17
8.5. シミュレーター停止.....	18
9. プロジェクト保存.....	19
9.1. プロジェクトの保存.....	19
参考資料 1 NJ シリーズ CPU ユニット.....	20
参考資料 2 NX シリーズ CPU ユニット.....	21
参考資料 3 NX シリーズ デジタル I/O ユニットの特長.....	25
参考資料 4 DC 入力ユニット NX-ID5342 の仕様.....	26
参考資料 5 トランジスタ出力ユニット NX-OD5121 の仕様.....	27

参考資料 6	I/O 電源接続ユニットの仕様.....	29
参考資料 7	変数の種類	30
参考資料 8	ラダー回路	32
参考資料 9	タスク	41
参考資料 10	オフラインデバッグ.....	42
参考資料 11	データトレース	44
改訂履歴	46

FA パートナーズネットワーク株式会社

1. はじめに

本資料は、NJ/NX シリーズでラダー言語を使用したプログラミングセミナーの資料です。
別紙の課題とともに使用します。

1.1. セミナーの流れ

以下のようにセミナーを進めます。

2.	Sysmac Studio を起動する	統合開発環境「Sysmac Studio」を起動します。
3.	ユニット構成を設定する	16 点入力モジュール、16 点出力モジュールを設定します。
4.	入出力接点に変数を割り付ける	入力接点、出力接点に変数を割り付けます。
5.	プログラムを作成する	ラダー言語でプログラムを作成します。
6.	タスク割り付けを行う	作成したプログラムをタスクに割り付けます。
7.	オフラインデバッグを行う	シミュレーターを使用し、オフラインデバッグを行います。
8.	データトレースにより動作確認を行う	データトレースを使用し、変数が正しく変化していることを確認します。
9.	プロジェクトを保存する	作成したプログラム（プロジェクト）を保存します。

2. Sysmac Studio の起動

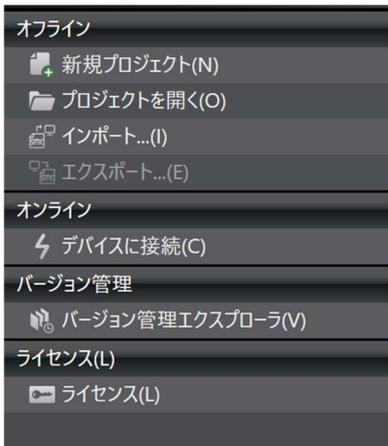
2.1. コントローラ NJ/NX/NY シリーズ用統合開発環境「Sysmac Studio」の起動

デスクトップにある「Sysmac Studio」アイコンをダブルクリックします。



2.2. 新規プロジェクトを作成

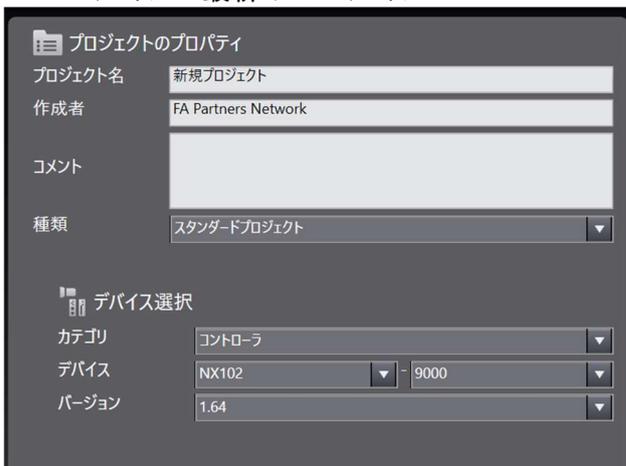
Sysmac Studio が起動し、スタートページが表示されるので、「新規プロジェクト」をクリックします。



2.3. プロジェクトのプロパティを入力

「プロジェクトのプロパティ」画面で、「プロジェクト名」を入力し、デバイス選択で以下を選択し、作成ボタンをクリックします。

- ・カテゴリ：コントローラ
- ・デバイス：NX102 – 9000
- ・バージョン：最新のバージョン



参考

参考資料 1 NJ シリーズ CPU ユニット

参考資料 2 NX シリーズ CPU ユニット

3. ユニット構成の設定

3.1. ユニット構成

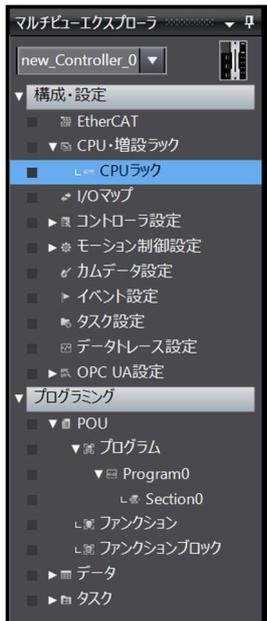
本課題では、PLC は NX102-9000 するものとします。

また、「試験用盤の配置図」の通り、16 点入力モジュール、16 点出力モジュールを使用するため、以下の NX ユニットを使用するものとします。

- ・ 16 点入力モジュール：ID-5342
- ・ 16 点出力モジュール：OD-5121

3.2. CPU・増設ラック構成編集画面の起動

マルチビューエクスプローラで「構成・設定」-「CPU・増設ラック」を選択し、「CPU ラック」をダブルクリックします。



エディットウィンドウに NX102 のイメージが表示されます。



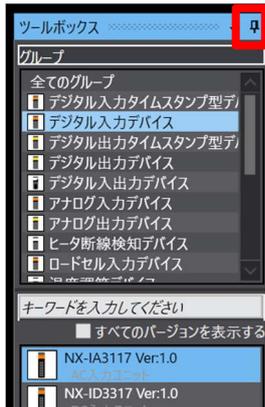
3.3. NX ユニットの登録

3.3.1. ID-5342 の登録

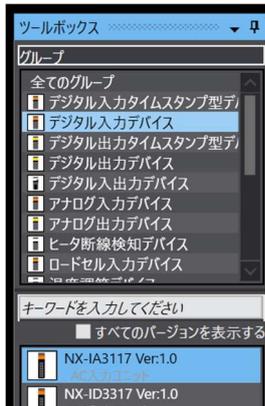
画面右側のツールボックスタブをクリックします。



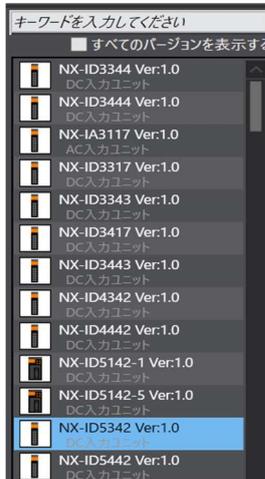
ツールボックスが表示されるので、押しピンをクリックし、ツールボックスが隠れないようにします。



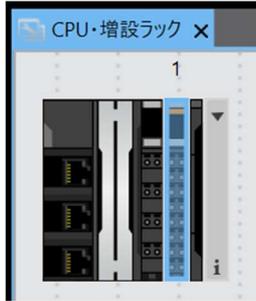
グループの中からデジタル入力デバイスをクリックします。



ツールボックスの下側にデジタル入力デバイスが表示されるので、「ID-5342 Ver1.0」をダブルクリックします。

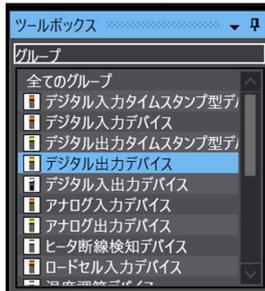


エディットウィンドウのイメージに ID-5342 が追加されます。

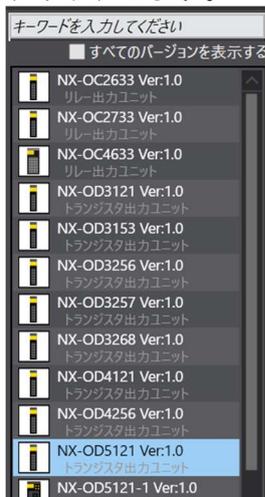


3.3.2. OD-5121 の登録

グループの中からデジタル出力デバイスをクリックします。



ツールボックスの下側にデジタル出力デバイスが表示されるので、「OD-5121 Ver1.0」をダブルクリックします。



エディットウィンドウのイメージに OD-5121 が追加されます。



参考

- 参考資料 3 NX シリーズ デジタル I/O ユニットの特長
- 参考資料 4 DC 入力ユニット NX-ID5342 の仕様
- 参考資料 5 トランジスタ出力ユニット NX-OD5121 の仕様
- 参考資料 6 I/O 電源接続ユニットの仕様

3.3.3. 電源接続ユニットの追加

グループの中からシステムユニットデバイスをクリックします。



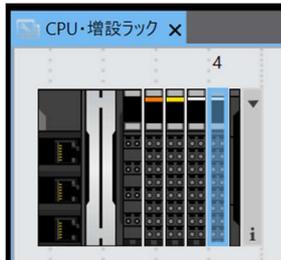
ツールボックスの下側にシステムユニットデバイスが表示されるので、「NX-PC0010 Ver1.0」をダブルクリックします。



ツールボックスの下側にシステムユニットデバイスが表示されるので、「NX-PC0020 Ver1.0」をダブルクリックします。

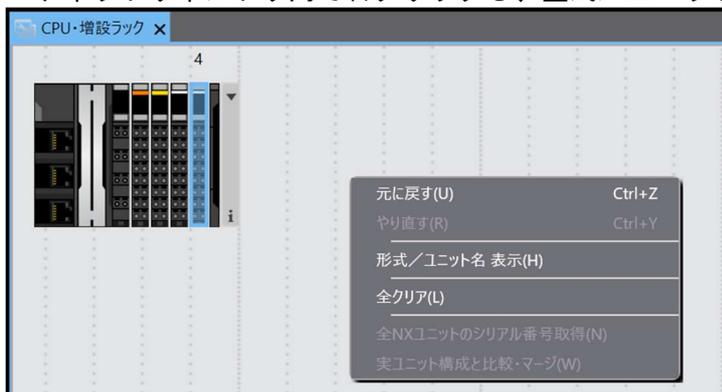


エディットウィンドウのイメージに PC0010 と PC0020 が追加されます。



3.3.4. 電源接続ユニットの装着位置変更

エディットウィンドウ内で右クリックし、型式／ユニット名 表示を選択します。



エディットウィンドウの画像イメージが型式／ユニット名に変わります。



NX-PC0010 をドラッグし、NX-ID5342 の左にドロップします。



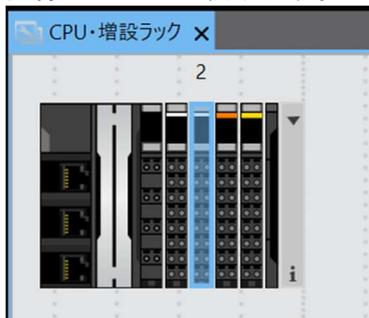
NX-PC0020 をドラッグし、NX-PC0010 の右にドロップします。



エディットウィンドウ内で右クリックし、型式／ユニット名を表示を選択しチェックを外します。



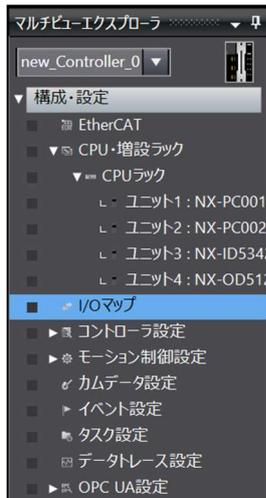
画像イメージに戻ります。



4. 入出力接点に変数を割付

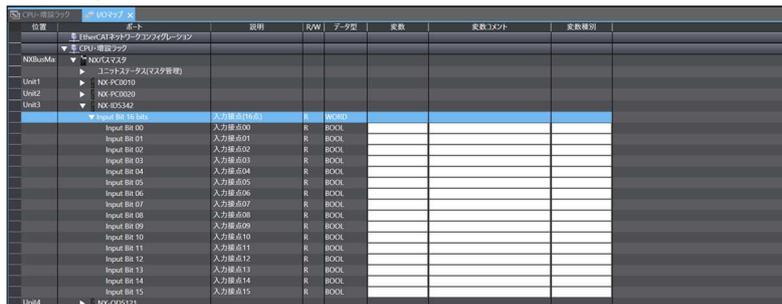
4.1. I/O マップの表示

マルチビューエクスプローラの構成・設定→I/O マップをダブルクリックします。



4.2. 入力接点に変数を割り付ける

I/O マップが表示されるので、NX バスマスタ→NX-ID5342 をクリックし、入力接点 16 個を表示します。

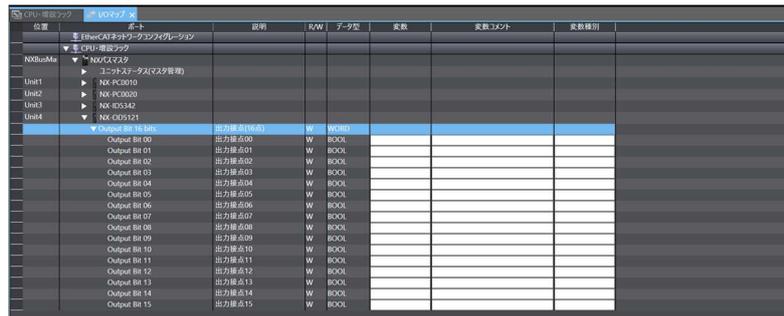


入力接点に変数を割り付けます。

ポート	変数	変数コメント
入力接点 00	LS1	コンベヤ右端リミットスイッチ
入力接点 01	LS2	コンベヤ左端リミットスイッチ
入力接点 02		
入力接点 03		
入力接点 04		
入力接点 05	PB1	自動モード：起動ボタン
入力接点 06	PB2	手動モード：左行ボタン
入力接点 07	PB3	手動モード：右行ボタン、自動モード：停止ボタン
入力接点 08	PB4	非常停止リセットボタン
入力接点 09	PB5	非常停止ボタン
入力接点 10		
入力接点 11	SS0	FALSE：手動モード、TRUE：自動モード
入力接点 12		
入力接点 13		
入力接点 14		
入力接点 15		

4.3. 出力接点に変数を割り付ける

I/O マップが表示されるので、NX バスマスタ→NX-OD5121 をクリックし、出力接点 16 個を表示します。



出力接点に変数を割り付けます。

ポート	変数	変数コメント
出力接点 00	RY1	コンベヤ左行指令
出力接点 01	RY2	コンベヤ右行指令
出力接点 02	PL1	サイクル動作中
出力接点 03	PL2	コンベヤ左行中
出力接点 04	PL3	コンベヤ右行中
出力接点 05	PL4	非常停止中
出力接点 06		
出力接点 07		
出力接点 08		
出力接点 09		
出力接点 10		
出力接点 11		
出力接点 12		
出力接点 13		
出力接点 14		
出力接点 15		

4.4. グローバル変数確認

プログラミング→データ→グローバル変数を選択し、I/O マップで入力した変数がグローバル変数に登録されていることを確認します。



参考

参考資料 7 変数の種類

5. プログラムの作成

5.1. POU 作成

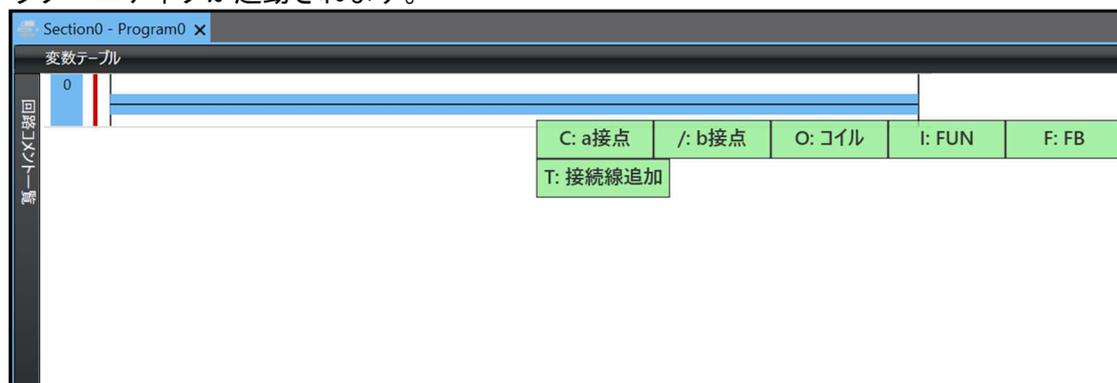
初期状態で Program0（ラダー言語用 POU）が作成されているので、**POU を追加する必要はありません。**

5.2. プログラム作成

POU→プログラム→Program0→Section0 をダブルクリックしてください。



ラダーエディタが起動されます。



5.3. セクション追加

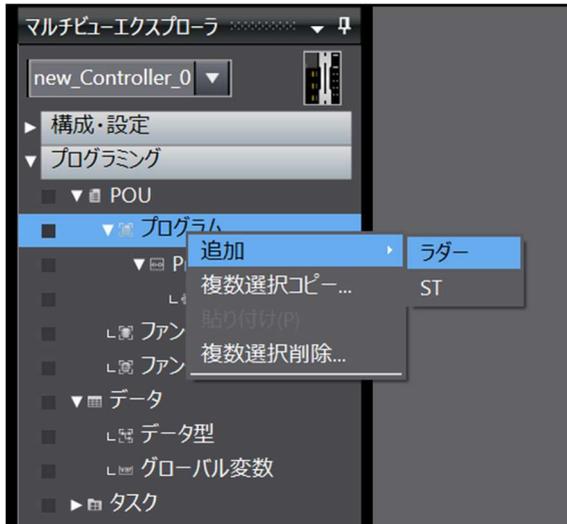
ラダープログラムは任意の管理単位に分割することができます。この分割単位を「セクション」と言います。

セクションを追加する場合、プログラミング→POU→プログラムでセクションを追加したいラダープログラムで右クリック、追加→セクションをクリックします。



5.4. プログラム追加

プログラム（POU : Program Organization Unit）を追加する場合、以下の手順で行ってください。
POU→プログラムで右クリックし、追加→ラダーをクリックします。



参考

参考資料 8 ラダー回路

5.5. プログラムのコンパイル

プログラム作成が完了したら、プログラムをビルド（コンパイル）します。
プロジェクト→リビルドを選択します。
ビルドエラーが発生したら修正を行います。

6. タスク割り付け

6.1. POU をタスクに割り付ける

6.1.1. タスク割り付け

ビルドエラーが無くなったら、タスク設定をダブルクリックし、プログラムをタスクに割り付けます。

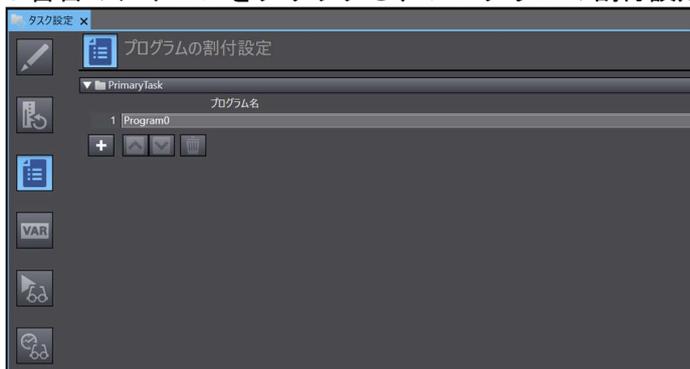


タスク設定が開きます。



6.1.2. プログラムをタスクに割り付け

今回は全てのプログラムをプライマリ定周期タスクに割り付けます。3番目のアイコンをクリックし、プログラムの割付設定を開きます。



プログラム名の下矢印をクリックし、割り付けたいプログラムを選択します。プログラムが複数ある場合、「+」をクリックしプログラムを割り付けます。



参考

参考資料 9 タスク

7. オフラインデバッグ

7.1. シミュレーターの起動

シミュレーション→実行をクリックし、オフラインでバックを行います。



線が緑色になったら、プログラムが動作しています。

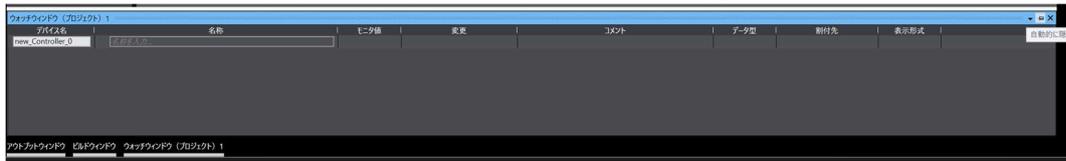


7.2. ウォッチウィンドウを開く

シミュレーション→実行をクリックし、ウォッチウィンドウを開き、デバッグを行います。



画面下に表示されるウォッチウィンドウをクリックしウォッチウィンドウを開き、右端のピンアイコンをクリックし、ウィンドウを固定表示します。（ウォッチウィンドウ以外が固定表示されたら、×アイコンをクリックしウィンドウを閉じてください。）



7.3. ウォッチウィンドウに変数設定

ウォッチウィンドウの名称に以下の変数を入力します。（手動モードの例）

変数名
PB2
PL2
RY1
PB3
PL3
RY2

7.4. 動作確認の実施

7.4.1. 手動モードの動作確認

入力変数（PB1、PB2）の変更欄の TRUE/FALSE を押下し、出力変数（PL2、PL3、RY1、RY2）が仕様通り出力されることを確認します。

デバイス名	名称	モニタ値	変更		コメント
new_Controller_0	PB2	True	TRUE	FALSE	手動モード：左行ボタン
new_Controller_0	PL2	True	TRUE	FALSE	コンベヤ左行中
new_Controller_0	RY1	True	TRUE	FALSE	コンベヤ左行指令
new_Controller_0	PB3	False	TRUE	FALSE	手動モード：右行ボタン、自動モード：停止ボタン
new_Controller_0	PL3	False	TRUE	FALSE	コンベヤ右行中
new_Controller_0	RY2	False	TRUE	FALSE	コンベヤ右行指令

7.4.2. 自動モード、非常停止の動作確認

自動モード、非常停止についても上記の方法で動作確認を行ってください。



参考

参考資料 10 オフラインデバッグ

8. データトレース

8.1. データトレースとは

自動モード、非常停止についても上記の方法で動作確認を行ってください。

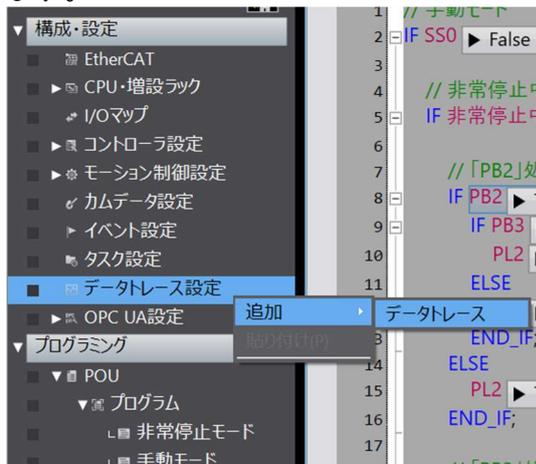
プログラムレスで指定した変数をサンプリングする機能です。

トリガ条件を設定し、条件成立前後のデータを記録するトリガトレース、およびトリガなしでサンプリングを連続実行し、結果をパソコン上のファイルに順次記録していく連続トレースの2つから選択できます。

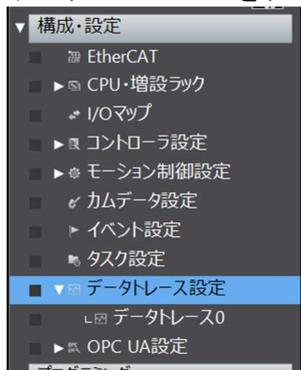
8.2. データトレースを開く

1秒経過、0.5秒での点滅をデータトレースにより確認します。

データトレース設定を右クリック→追加→データトレースを選択し、データトレース0を追加します。



データトレース0をダブルクリックします。



8.3. データトレース設定（トリガトレース）

エディットウィンドウで以下の設定を行います。

- ① トリガ条件有効をチェック。
- ② 右側のトリガ条件欄に PB1 を入力。
- ③ データトレース0タブの下側にある「+」をクリック。
- ④ 名前欄に下記の変数を入力。

PB1、PB、PL1、LS1、RY1、LS2、RY2

8.4. データトレース開始

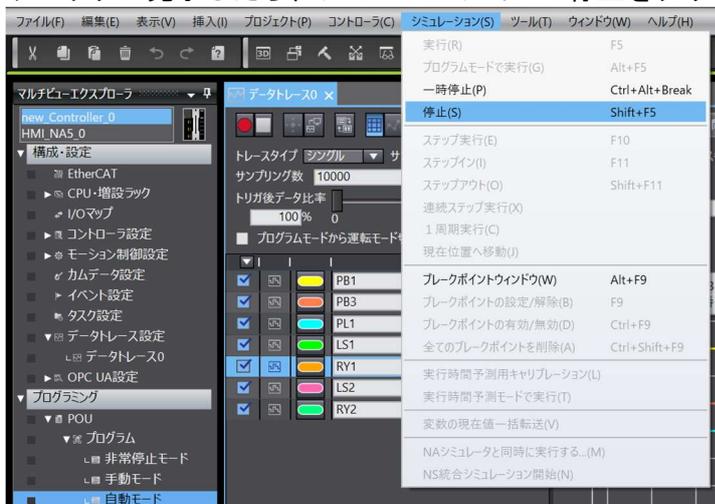
実行ボタンをクリックします。



自動モードでPB1 を ON するとデータトレースを開始します。
データトレースを終了するとグラフが表示されるので、1 秒、0.5 秒が正しく動作していることを確認します。

8.5. シミュレーター停止

デバッグが完了したら、シミュレーション→停止をクリックし、シミュレータを停止します。



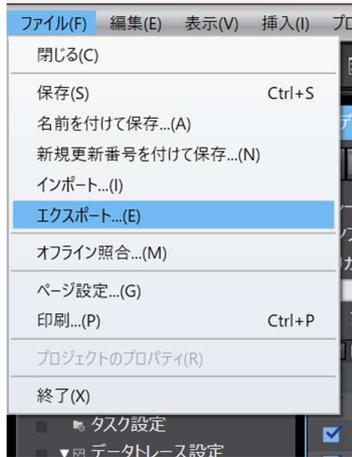
参考

参考資料 11 データトレース

9. プロジェクト保存

9.1. プロジェクトの保存

ファイル→エクスポートをクリックし、作成したプロジェクトを保存します。



プロジェクトを保存したい場所にファイル名を付けて保存します。

参考資料 1 NJ シリーズ CPU ユニット

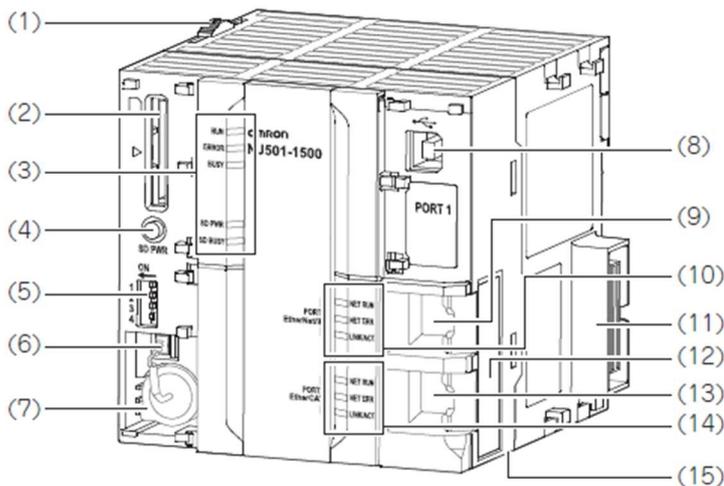
「NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-466)」より

●NJ シリーズ

3-1-1 形式と仕様

種別	形式	入出力点数/構成ユニット装着台数 (最大増設ラック数)	プログラム容量	変数容量	モーション軸数	データベース接続機能	GEM機能	オムロン製ロボット制御機能	NJ Robotics 機能でのロボット制御機能	数値制御機能		
NJ501 CPU ユニット	形 NJ501-1500	2560 点/40 台 (増設最大 3 ラック)	20MB	2MB:電断保持 4MB:電断非保持	64	なし	なし	なし	なし	なし		
	形 NJ501-1400				32							
	形 NJ501-1300				16							
	形 NJ501-R500				64						あり	あり
	形 NJ501-R400				32							
	形 NJ501-R300				16							
	形 NJ501-4500				64						なし	
	形 NJ501-4400				32							
	形 NJ501-4300				16							
	形 NJ501-4310				64							
	形 NJ501-1520				32						あり	なし
	形 NJ501-1420				32							
	形 NJ501-1320				16							
	形 NJ501-4320				なし						あり	なし
	形 NJ501-1340				なし						なし	あり
形 NJ501-5300	なし	なし	なし	あり								
NJ301 CPU ユニット	形 NJ301-1200		5MB	0.5MB:電断保持 2MB:電断非保持	8					あり なし		
	形 NJ301-1100				4							
NJ101 CPU ユニット	形 NJ101-1000		3MB		2							
	形 NJ101-9000				0							
	形 NJ101-1020				2						あり	
	形 NJ101-9020				0							

3-1-2 各部の名称と機能



参考資料 2 NX シリーズ CPU ユニット

「NX シリーズ 形□□ CPU ユニット ユーザーズマニュアル ハードウェア編」より

●NX1P2

3-1-1 形式と仕様

ここでは、CPU ユニットの概略仕様、電氣的、機械的仕様について説明します。その他の主な仕様については、「1-2 仕様 (P.1-9)」を参照してください。

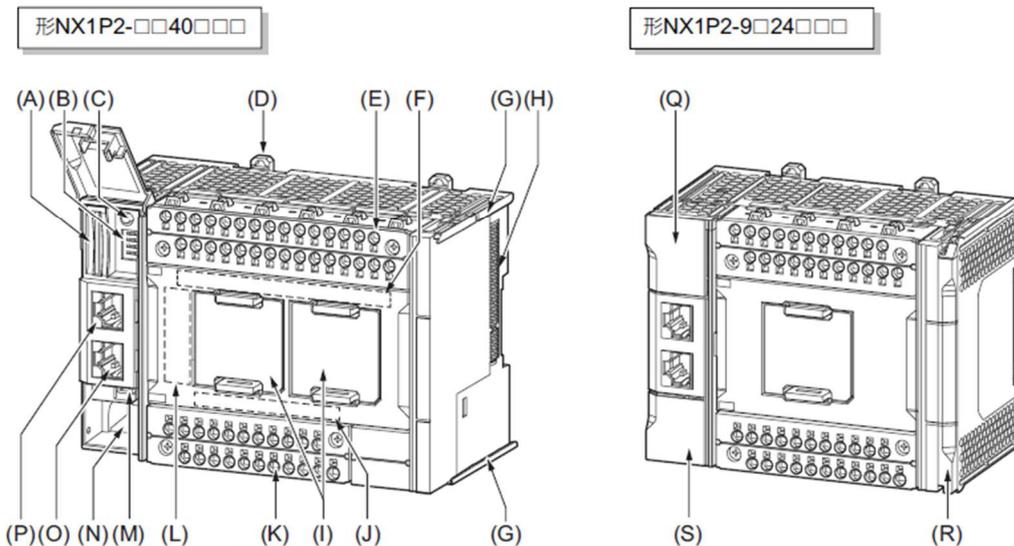
形式と概略仕様

形式と概略仕様を以下に示します。

形式	プログラム容量	変数容量	制御軸最大数	内蔵 I/O				
				入出力の合計点数	入力点数	出力点数		
形 NX1P2-1140DT	1.5MB	32kB (電断保持) /2MB (電断非保持)	12 軸	40 点	24 点	16 点.NPN トランジスタ		
形 NX1P2-1140DT1						16 点.PNP トランジスタ*1		
形 NX1P2-1040DT			10 軸			16 点.NPN トランジスタ		
形 NX1P2-1040DT1						16 点.PNP トランジスタ*1		
形 NX1P2-9024DT	1.0MB		4 軸	24 点	14 点	10 点.NPN トランジスタ		
形 NX1P2-9024DT1						10 点.PNP トランジスタ*1		
形 NX1P2-9B40DT			2 軸			40 点	24 点	16 点.NPN トランジスタ
形 NX1P2-9B40DT1								16 点.PNP トランジスタ*1
形 NX1P2-9B24DT	24 点	14 点		10 点.NPN トランジスタ				
形 NX1P2-9B24DT1				10 点.PNP トランジスタ*1				

3-1-2 各部の名称と機能

以下に示す 2 つの形式では、オプションボードスロット数と内蔵 I/O の点数は異なりますが、各部の名称と機能は共通です。CPU ユニットの形式と内蔵 I/O 点数などの概略仕様については、「3-1-1 形式と仕様 (P.3-2)」を参照してください。



●NX102

3-1-1 形式と仕様

ここでは、CPUユニットの概略仕様、電気的、機械的仕様について説明します。その他の主な仕様については、「1-2 仕様 (P.1-8)」を参照してください。

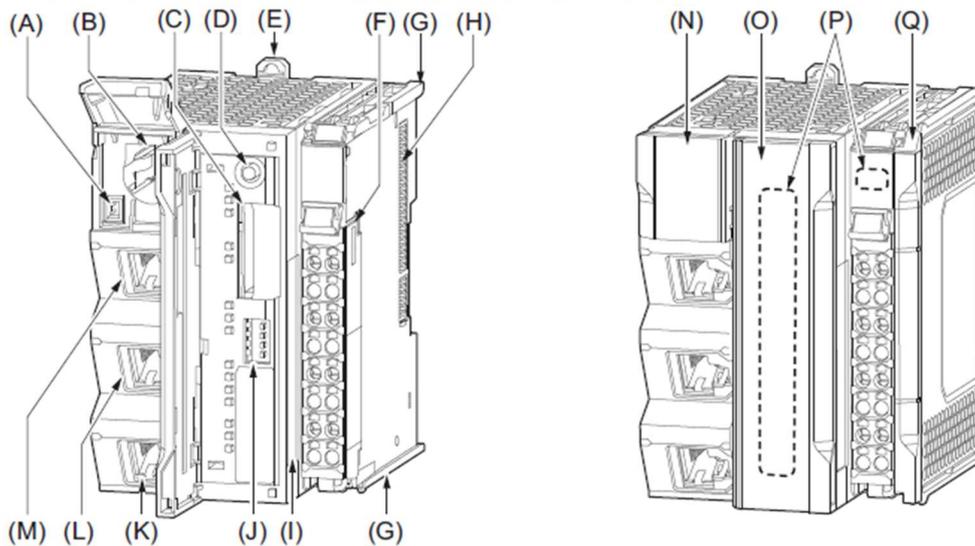
形式と概略仕様

形式と概略仕様を以下に示します。

形式	プログラム容量	変数容量	制御軸最大数 (使用実軸最大数)	データベース接続機能
形 NX102-1200	5MB	1.5MB (電断保持) /32MB (電断非保持)	15軸 (12軸)	なし
形 NX102-1100			15軸 (8軸)	
形 NX102-1000			15軸 (6軸)	
形 NX102-9000			4軸 (4軸)	
形 NX102-1220	あり		15軸 (12軸)	あり
形 NX102-1120			15軸 (8軸)	
形 NX102-1020			15軸 (6軸)	
形 NX102-9020			4軸 (4軸)	

3-1-2 各部の名称と機能

CPUユニットの各部の名称、およびそれらの機能を以下に示します。



●NX502

3-1-1 形式と仕様

ここでは、CPUユニットの概略仕様、電気的、機械的仕様について説明します。その他の主な仕様については、「1-2 仕様 (P.1-9)」を参照してください。

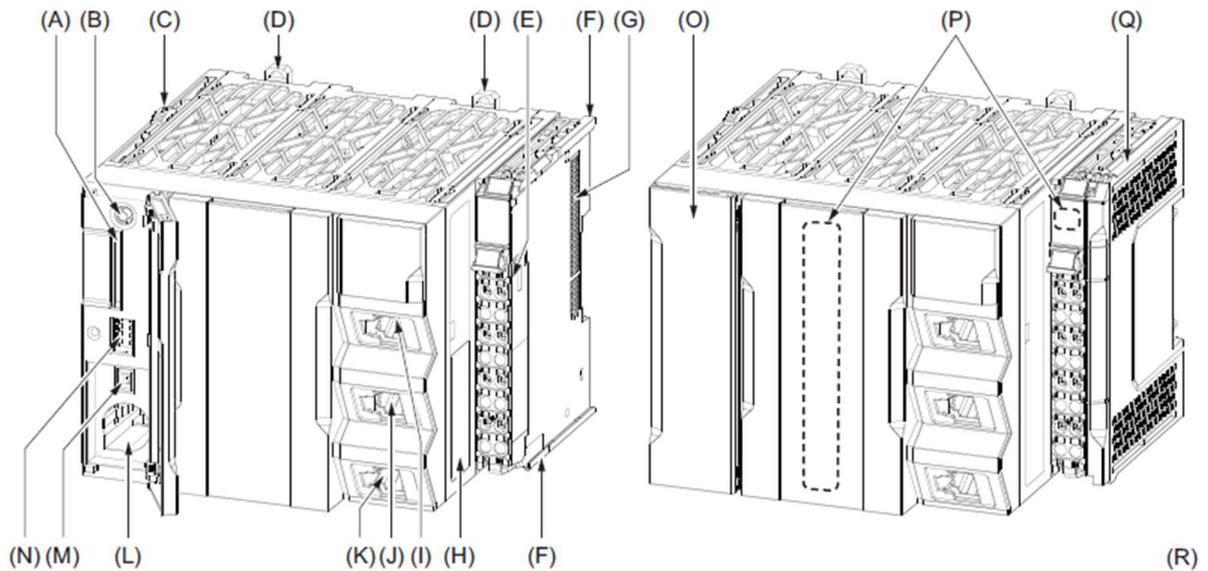
形式と概略仕様

形式と概略仕様を以下に示します。

形式	プログラム容量	変数容量	制御軸最大数 (使用実軸最大数)	データベース接続機能
形 NX502-1700	80MB	4MB (電断保持) /256MB (電断非保持)	256 軸 (256 軸)	あり
形 NX502-1600			128 軸 (128 軸)	
形 NX502-1500			128 軸 (64 軸)	
形 NX502-1400			64 軸 (32 軸)	
形 NX502-1300			32 軸 (16 軸)	

3-1-2 各部の名称と機能

CPUユニットの各部の名称、およびそれらの機能を以下に示します。

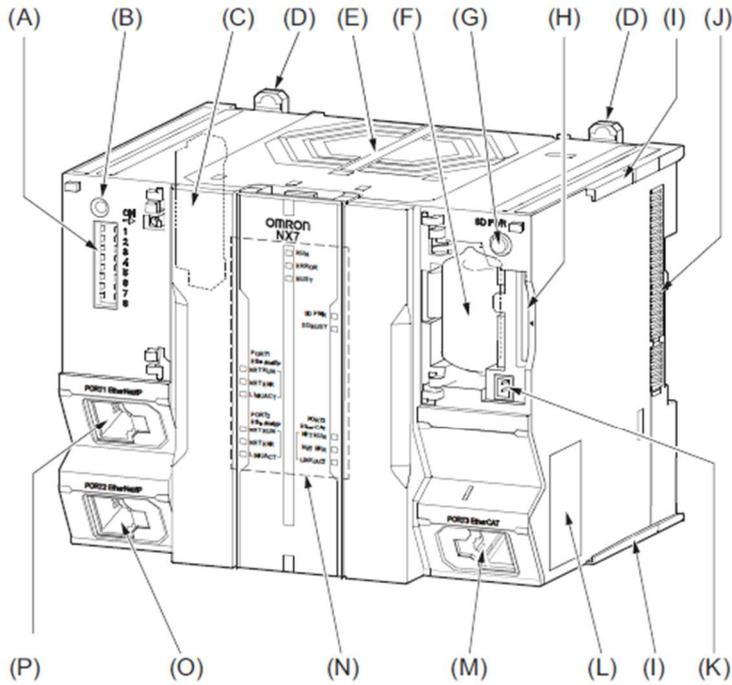


●NX701

3-1-1 形式と仕様

形式	プログラム容量	変数容量	モーション軸数	データベース接続機能
NX701-1700	80MB	4MB（電断保持） /256MB（電断非保持）	256軸	なし
NX701-1600			128軸	
NX701-1720	80MB	4MB（電断保持） /256MB（電断非保持）	256軸	あり
NX701-1620			128軸	

3-1-2 各部の名称と機能



参考資料 3 NX シリーズ デジタル I/O ユニットの特長

「NX シリーズ デジタル I/O ユニット ユーザーズマニュアル(SBCA-407)」より

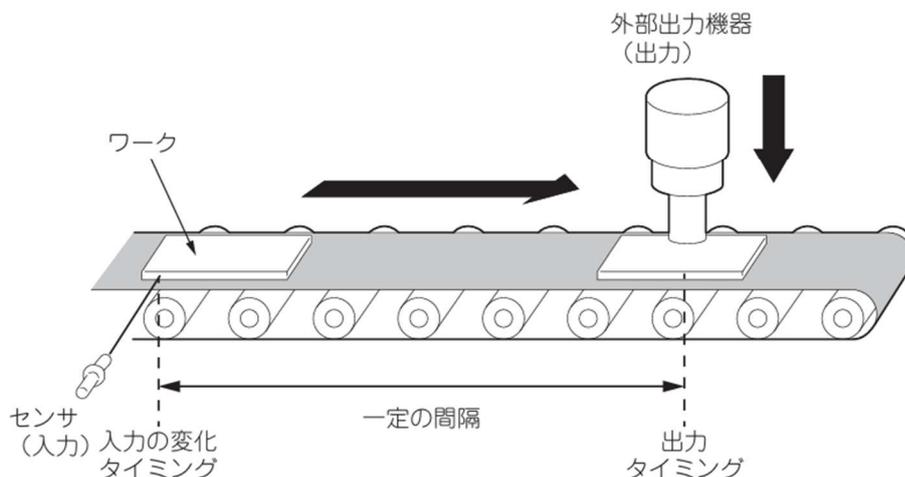
NX バスのリフレッシュ周期に同期した入出力同期

NX シリーズ CPU ユニットまたは EtherCAT カプラユニットと入出力同期リフレッシュ方式を持つ NX ユニットの組み合わせると、NX バスのリフレッシュ周期に同期したタイミングで、複数の NX ユニットの I/O 制御を同期することができます。

これにより、複数の NX ユニットの入出力タイミングのジッタの発生を抑制し、正確なタイミングで I/O 制御を行うことができます。

入力の変化タイミングから一定の間隔で出力する制御が可能

NX シリーズ CPU ユニットまたは EtherCAT カプラユニットと、変化時刻入力方式の NX ユニット、時刻指定出力方式の NX ユニットの組み合わせを使用して、あるセンサの入力の変化タイミングから一定の間隔で出力を行うといった制御ができます。



この制御の例については「5-2-10 センサ入力変化から一定の時間で出力を ON にするときの例 (P.5-28)」を参照してください。

スクリーレスクランプ端子台による簡単な I/O 配線

端子台には、スクリーレスクランプ端子台を採用しています。

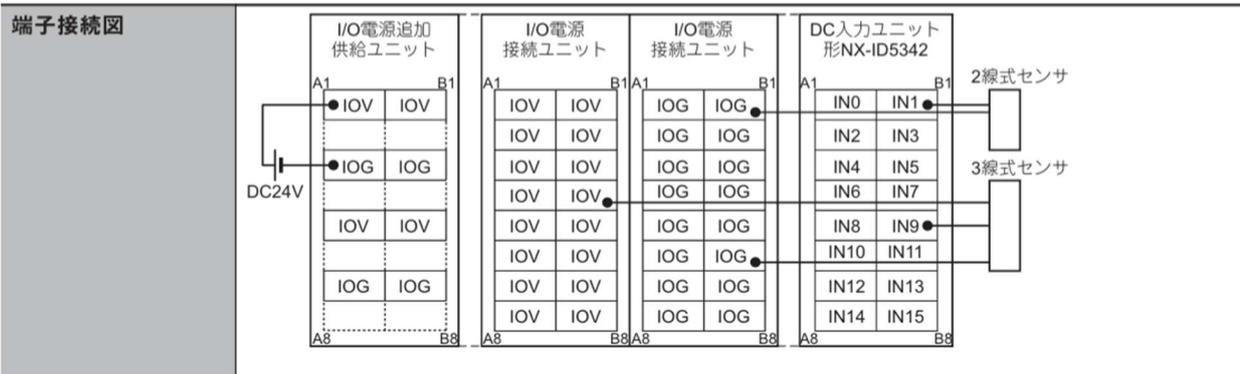
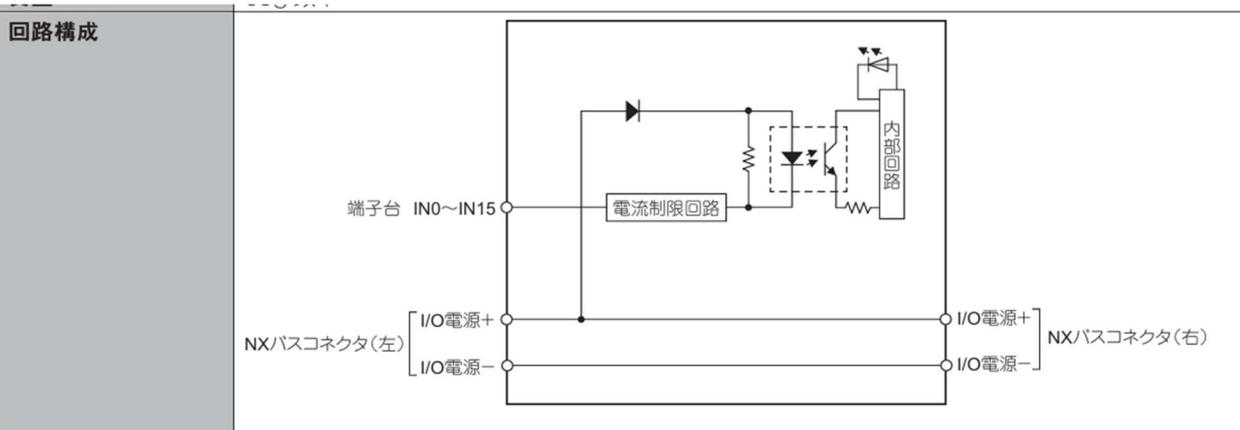
棒端子を使用することにより、差し込むだけで配線ができます。ねじ止めが不要で配線工数が大幅に削減できます。

参考資料 4 DC 入力ユニット NX-ID5342 の仕様

「NX シリーズ デジタル I/O ユニット ユーザーズマニュアル(SBCA-407)」より

DC 入力ユニット (スクルーレスクランプ端子台、12mm 幅)

形式	点数	内部 I/O コモン線処理	定格入力電圧	I/O リフレッシュ方式	ON/OFF 応答時間	参照ページ
形 NX-ID3317	4 点	NPN	DC12~24V	フリーランリフレッシュ、または入出力同期リフレッシュ切り替え	20 μ s 以下 / 400 μ s 以下	P.A-9
形 NX-ID3343			DC24V	変化時刻入力方式専用	100ns 以下 / 100ns 以下	P.A-11
形 NX-ID3344					P.A-13	
形 NX-ID3417		PNP	DC12~24V	フリーランリフレッシュ、または入出力同期リフレッシュ切り替え	20 μ s 以下 / 400 μ s 以下	P.A-15
形 NX-ID3443			DC24V	変化時刻入力方式専用	100ns 以下 / 100ns 以下	P.A-17
形 NX-ID3444					P.A-19	
形 NX-ID4342	8 点	NPN		フリーランリフレッシュ、または入出力同期リフレッシュ切り替え	20 μ s 以下 / 400 μ s 以下	P.A-21
形 NX-ID4442		PNP				P.A-23
形 NX-ID5342	16 点	NPN				P.A-25
形 NX-ID5442		PNP				P.A-27



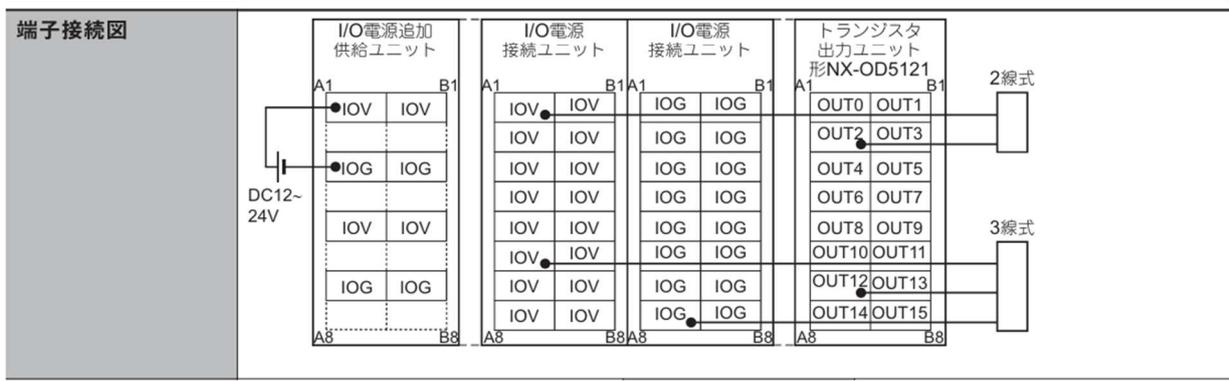
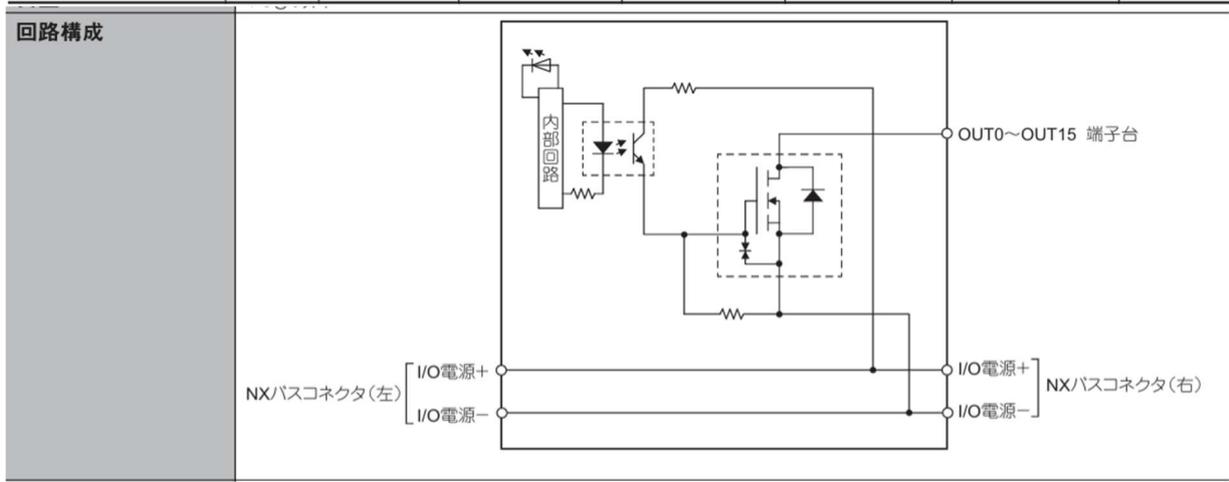
I/O 電源接続ユニットが必要。

参考資料5 トランジスタ出力ユニット NX-OD5121 の仕様

「NXシリーズ デジタル I/O ユニット ユーザーズマニュアル(SBCA-407)」より

トランジスタ出力ユニット (スクリーレスクランプ端子台、12mm幅)

形式	点数	内部 I/O コモン線処理	最大負荷電流	定格電圧	I/O リフレッシュ方式	ON/OFF 応答時間	参照ページ	
形 NX-OD2154	2 点	NPN	0.5A/点、1A/ユニット	DC24V	時刻指定出力方式専用	300ns 以下 / 300ns 以下	P.A-48	
形 NX-OD2258		PNP					P.A-50	
形 NX-OD3121	4 点	NPN	0.5A/点、2A/ユニット	DC12~24V	フリーランリフレッシュ、または入出力同期リフレッシュ切り替え	0.1ms 以下 / 0.8ms 以下	P.A-52	
形 NX-OD3153				DC24V			300ns 以下 / 300ns 以下	P.A-54
形 NX-OD3256								PNP
形 NX-OD3257		300ns 以下 / 300ns 以下		P.A-58				
形 NX-OD3268				2A/点、8A/ユニット		0.5ms 以下 / 1.0ms 以下	P.A-60	
形 NX-OD4121				8 点		NPN	DC12~24V	0.1ms 以下 / 0.8ms 以下
形 NX-OD4256	PNP	DC24V	0.5ms 以下 / 1.0ms 以下		P.A-64			
形 NX-OD5121	16 点	NPN		DC12~24V		0.1ms 以下 / 0.8ms 以下	P.A-66	
形 NX-OD5256		PNP		DC24V		0.5ms 以下 / 1.0ms 以下	P.A-68	



I/O 電源接続ユニットが必要。

●システムユニットの種類

1-1-2 システムユニットの種類

システムユニットの種類は、以下のとおりです。

種類	用途
システムユニット	CPU ラックまたはスレーブターミナルの構築上、必要に応じて使用するユニットです。
NX ユニット電源追加供給ユニット	NX ユニット電源が不足するときに使用するユニットです。
I/O 電源追加供給ユニット	I/O 電源供給が不足するときや、CPU ラックまたはスレーブターミナル内で I/O 電源を分離したいときに使用するユニットです。なお、NX シリーズ CPU ユニットの CPU ラックで使用する NX ユニットの I/O 電源供給方法が NX バスからの供給の場合、本ユニットを使用する必要があります。
I/O 電源接続ユニット	外部入出力機器の接続で I/O 電源端子が不足するときに使用するユニットです。
シールド接続ユニット	外部入出力機器の接続で複数のシールドを一括で接地するときに使用するユニットです。

参考資料 6 I/O 電源接続ユニットの仕様

「NX シリーズ システムユニット ユーザーズマニュアル(SBCA-409)」より

名称	機能																																																																																																												
I/O 電源接続ユニット	<p>デジタル I/O ユニットやアナログ I/O ユニットなどと外部接続機器を接続するとき、I/O 電源端子が足りない場合に、本ユニットを使用します。</p> <p>I/O 電源接続ユニットでは I/O 電源が分離されない</p> <p>NPNタイプの例</p> <p>I/O電源接続ユニット (IOV端子数: 16)</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>A1</td><td>IOV</td><td>IOV</td><td>B1</td></tr> <tr><td></td><td>IOV</td><td>IOV</td><td></td></tr> <tr><td></td><td>IOV</td><td>IOV</td><td></td></tr> <tr><td></td><td>IOV</td><td>IOV</td><td></td></tr> <tr><td></td><td>IOV</td><td>IOV</td><td></td></tr> <tr><td></td><td>IOV</td><td>IOV</td><td></td></tr> <tr><td></td><td>IOV</td><td>IOV</td><td></td></tr> <tr><td></td><td>IOV</td><td>IOV</td><td></td></tr> <tr><td>A8</td><td>IOV</td><td>IOV</td><td>B8</td></tr> </table> <p>I/O電源接続ユニット (IOG端子数: 16)</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>A1</td><td>IOG</td><td>IOG</td><td>B1</td></tr> <tr><td></td><td>IOG</td><td>IOG</td><td></td></tr> <tr><td></td><td>IOG</td><td>IOG</td><td></td></tr> <tr><td></td><td>IOG</td><td>IOG</td><td></td></tr> <tr><td></td><td>IOG</td><td>IOG</td><td></td></tr> <tr><td></td><td>IOG</td><td>IOG</td><td></td></tr> <tr><td></td><td>IOG</td><td>IOG</td><td></td></tr> <tr><td></td><td>IOG</td><td>IOG</td><td></td></tr> <tr><td>A8</td><td>IOG</td><td>IOG</td><td>B8</td></tr> </table> <p>DC入力ユニット (NPNタイプ)</p> <table border="1" style="display: inline-table;"> <tr><td>A1</td><td>0</td><td>1</td><td>B1</td></tr> <tr><td></td><td>2</td><td>3</td><td></td></tr> <tr><td></td><td>4</td><td>5</td><td></td></tr> <tr><td></td><td>6</td><td>7</td><td></td></tr> <tr><td></td><td>8</td><td>9</td><td></td></tr> <tr><td></td><td>10</td><td>11</td><td></td></tr> <tr><td></td><td>12</td><td>13</td><td></td></tr> <tr><td></td><td>14</td><td>15</td><td></td></tr> <tr><td>A8</td><td></td><td></td><td>B8</td></tr> </table> <p>2線式センサ* (リミットスイッチなど) 茶(白) 青(黒)</p> <p>NPN出力 3線式センサ* (光電センサ、近接センサなど) 黒(白) 茶(赤) 青(黒)</p> <p>* 光電センサと近接センサの JIS 規格の改訂に伴い、芯線の色が変更されています。 () 内は旧線色です。</p>	A1	IOV	IOV	B1		IOV	IOV		A8	IOV	IOV	B8	A1	IOG	IOG	B1		IOG	IOG		A8	IOG	IOG	B8	A1	0	1	B1		2	3			4	5			6	7			8	9			10	11			12	13			14	15		A8			B8																																																
A1	IOV	IOV	B1																																																																																																										
	IOV	IOV																																																																																																											
	IOV	IOV																																																																																																											
	IOV	IOV																																																																																																											
	IOV	IOV																																																																																																											
	IOV	IOV																																																																																																											
	IOV	IOV																																																																																																											
	IOV	IOV																																																																																																											
A8	IOV	IOV	B8																																																																																																										
A1	IOG	IOG	B1																																																																																																										
	IOG	IOG																																																																																																											
	IOG	IOG																																																																																																											
	IOG	IOG																																																																																																											
	IOG	IOG																																																																																																											
	IOG	IOG																																																																																																											
	IOG	IOG																																																																																																											
	IOG	IOG																																																																																																											
A8	IOG	IOG	B8																																																																																																										
A1	0	1	B1																																																																																																										
	2	3																																																																																																											
	4	5																																																																																																											
	6	7																																																																																																											
	8	9																																																																																																											
	10	11																																																																																																											
	12	13																																																																																																											
	14	15																																																																																																											
A8			B8																																																																																																										

参考資料 7 変数の種類

「NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)」より

2-3-1 変数の種類

NJ/NX シリーズ CPU ユニットでは、ユーザプログラムから、変数を介して I/O や CPU ユニット内部などにアクセスします。

変数は、ユーザによる属性の設定可否およびアクセス対象により以下のように分類されます。

変数		属性の設定可否	アクセス対象	
ユーザ定義変数		すべての属性が設定可能	CPU ユニット内部	
準ユーザ定義変数	デバイス変数	一部の属性が設定可能*1	EtherCAT スレーブデバイス変数	EtherCAT スレーブ*2*3
			NX ユニットデバイス変数*4	CPU ユニット上の NX ユニット
			X バスユニットデバイス変数*5	X バスユニット
			内蔵 I/O デバイス変数*6	内蔵 I/O
			オプションボードデバイス変数*6	アナログ入出力用オプションボード
			CJ ユニットデバイス変数*7	CJ シリーズ基本 I/O ユニット、CJ シリーズ高機能ユニット
	カムデータ変数		サーボドライバ、エンコーダ入力スレーブ、CPU ユニット内部	
システム定義変数	PLC システム変数	すべての属性が設定不可	CPU ユニット内部	
	モーション制御システム変数		MC 共通変数	サーボドライバ、エンコーダ入力スレーブ、CPU ユニット内部
			軸変数	
			軸グループ変数	
	EtherNet/IP システム変数		内蔵 EtherNet/IP ポート	
	EtherCAT マスタシステム変数		内蔵 EtherCAT マスタポート	
	X バスシステム変数*9		CPU ユニット内部	
NX バスシステム変数*8	CPU ユニット内部			

● 基本データ型の種類

基本データ型は、以下に分類できます。

分類	定義	
ブール型	値が 0 または 1 のいずれかのデータ型です。	BOOL
ビット列型	ビットの列で値を表現したデータ型です。	BYTE,WORD,DWORD,LWORD
整数型	値が整数値のデータ型です。	SINT,INT,DINT,LINT,USINT,UINT,UDINT,ULINT
実数型	値が実数値のデータ型です。	REAL,LREAL
持続時間型	値が時間 (日・時・分・秒・ミリ秒) のデータ型です。	TIME
時刻型	値が時刻 (時・分・秒) のデータ型です。	DATE
日付型	値が日付 (年・月・日) のデータ型です。	TIME_OF_DAY
日付時刻型	値が日付時刻 (年・月・日・時・分・秒・ミリ秒) のデータ型です。	DATE_AND_TIME
文字列型	値が文字列のデータ型です。	STRING

● 派生データ型の種類

派生データ型は、その仕様をユーザが定義するデータ型です。Sysmac Studio の「データ型」ビューに登録されます。派生データ型には、以下のものがあります。

種類	内容
構造体型	複数のデータ型を階層的にひとつにまとめたデータ型です。
共用体型	同じデータを、場合によって異なるデータ型として扱えるようにしたデータ型です。
列挙型	あらかじめ用意された名前リスト内のひとつを値とするデータ型です。

詳細は、後述の「6-3-6 派生データ型 (P.6-38)」を参照してください。

● データ型に対する指定

各データ型に対して、以下の配列指定、範囲指定が可能です。

種類	内容
配列指定	同じデータ型の要素をまとめて、先頭からの番号（添え字）で指定します。基本データ型および派生データ型に指定できます。
範囲指定	あらかじめ決められた範囲内の値をとることを明示した指定です。基本データ型の整数型に指定ができます。

● 変数名として使用可能な文字

- 0～9、A～Z、a～z
- 半角カナ
- _（アンダーバー）
- マルチバイト文字（例: 日本語）予約語は、後述の「予約語」を参照してください。

● 組み合わせて使用できない文字

- 数字（0～9）で始まる文字列
- P_で始まる文字列
- アンダーバーで始まる文字列
- アンダーバー（_）を2個以上連続した文字列
- アンダーバー（_）で終端する文字列
- 識別子を構成する文字列の接頭または、接尾に1文字以上の拡張空白類（全角空白など、Unicode まで拡張した空白系文字）を含む文字列

参考資料 8 ラダー回路

「Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)」より

●各回路部品の挿入操作手順

回路部品を挿入する基本的な操作方法は次の4通りです。

方法1: ツールボックスから回路部品をドラッグ&ドロップする。

方法2: 接続線を選択して、右クリックメニューから「回路部品の挿入」を選択する。

方法3: 接続線を選択して、回路部品の挿入のショートカットキーを押す。

方法4: 接続線を選択して、メインメニューの「挿入」-「回路部品」から各回路部品を選択する。

●接点の挿入 (AND 回路)

接点を AND 回路として挿入する操作方法は次の4通りです。

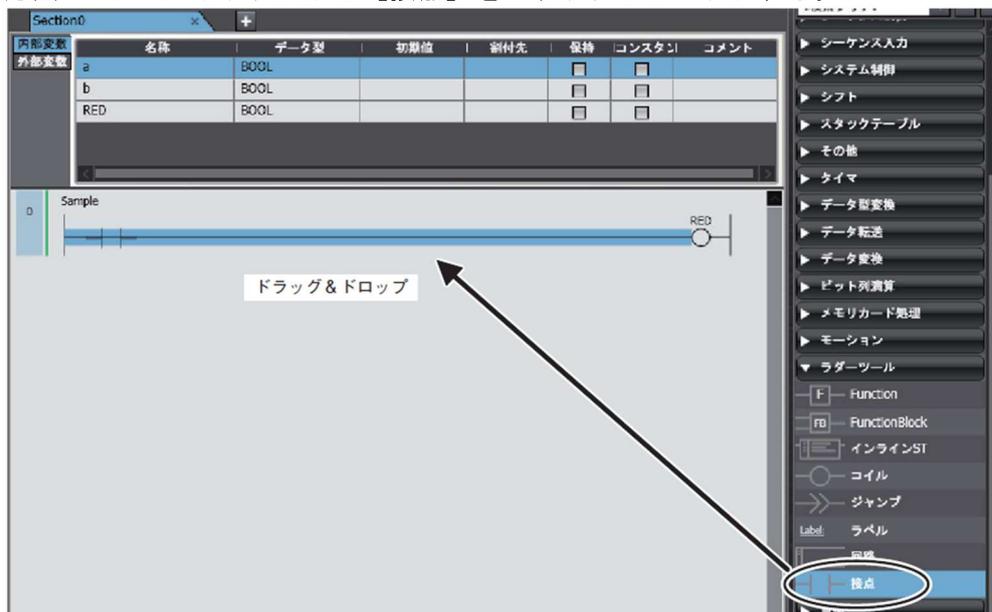
方法1: ツールボックスから「接点」をドラッグ&ドロップする。

方法2: 接続線を選択して、右クリックして「接点を挿入」を選択する。

方法3: 接続線を選択して、[C]キーを押す。

方法4: 挿入箇所を選択して、「挿入」-「回路部品」-「a 接点」を選択する。

方法1: ツールボックスから「接点」をドラッグ&ドロップする。



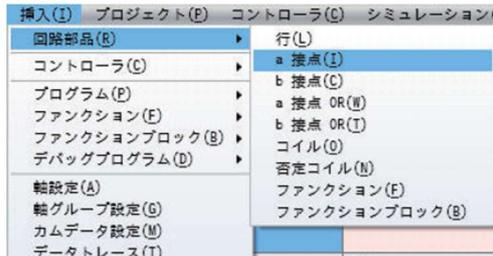
方法2: 接続線を選択して、右クリックして「接点を挿入」を選択する。



方法3: 接続線を選択して、[C]キーを押す。



方法 4: 挿入箇所を選択して、[挿入] - [回路部品] - [a 接点] を選択する。



●接点の挿入（OR 回路）

接点を OR 回路として挿入する操作方法は次の 5 通りの方法があります。

- 方法 1: ツールボックスから接点を挿入する上の接点に [接点] をドラッグ&ドロップする。
- 方法 2: OR 回路を挿入する接点を右クリックして [接点を挿入 (上側)] または [接点を挿入 (下側)] を選択する
- 方法 3: 接点を入力する始点の接続線から終点の接続線まで接続線をドラッグ&ドロップする。
- 方法 4: 接点を選択して、[W]キーを押す。否定の場合は、[X]キーを押す。
- 方法 5: 挿入箇所を選択して、[挿入] - [回路部品] - [a 接点 OR] を選択する。

●直列コイルの挿入（AND 回路）

直列コイルを挿入するに操作方法は次の 4 通りの方法があります。

- 方法 1: ツールボックスから [コイル] をドラッグ&ドロップで挿入する。
- 方法 2: 接続線を選択して、右クリックして [コイルを挿入] を選択する。
- 方法 3: 接続線を選択して、[O]キーを押す。否定の場合は、[Q]キーを押す。
- 方法 4: 挿入箇所を選択して、[挿入] - [回路部品] - [出力コイル] を選択する。

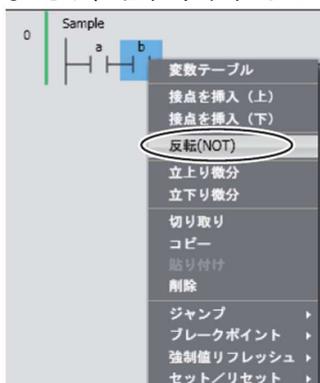
●コイルの挿入（OR 回路）

コイルを OR 回路として挿入する操作方法は次の 3 通りの方法があります。

- 方法 1: ツールボックスから [コイル] を OR 形に挿入されるコイルの上へドラッグ&ドロップで挿入する。
- 方法 2: コイルを入力する始点の接続線から終点の右母線まで、接続線をドラッグ&ドロップする。
- 方法 3: ラダーエディタ上の OR 形にしたいコイルを右クリックして [コイルの挿入] を選択する。

●接点、コイルの編集（接点、コイルの反転）

接点またはコイルを選択して、[I]キーを押します。
または、右クリックして [反転 (NOT)] を選択します。



●接点、コイルの編集（微分条件の設定）

立ち上がり微分を設定するには、接点、コイルを選択して、右クリックメニューから [立上り微分] を選択します。

立ち下がり微分を設定するには、[立下り微分] を選択します。



●接点、コイルの削除

削除する接点またはコイルを選択し、右クリックして [削除] を選択します。または、[Delete] キーを押します。

●回路部品への変数入力

挿入した接点、コイルに変数を入力するには、次の方法があります。

入力対象の型に一致した変数がドロップダウンリストに表示されます。

グローバル変数の場合

方法 1: あらかじめグローバル変数テーブルに登録された変数をラダーエディタ上でドロップダウンリストから選択する。

ローカル変数の場合

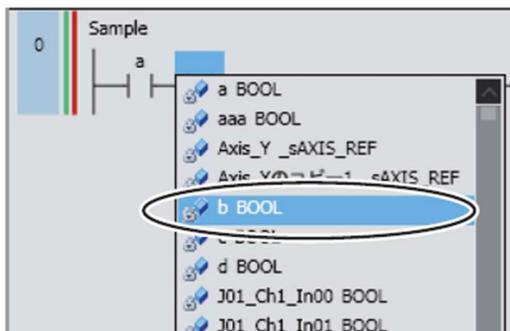
方法 1: あらかじめローカル変数テーブルに登録された変数をラダーエディタ上でドロップダウンリストから選択する。

方法 2: 同じ POU のローカル変数テーブルから変数をラダーエディタ上にドラッグ&ドロップする。

方法 3: ローカル変数テーブルに変数を登録する前に、変数名をラダーエディタ（または ST エディタ）上で入力する。

方法 1: ラダーエディタ上で変数を選択する

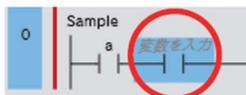
対象の接点を選択し、[変数を入力] 部分をクリックするか、[Ctrl]+[Space] キーを押して、ドロップダウンリストから変数を選択します。ドロップダウンリストには、入力対象の型に一致した変数が候補として表示されます。



方法 2: 同じ POU のローカル変数テーブルから変数をドラッグ&ドロップする
 ローカル変数テーブルから目的の変数をドラッグして、対象の接点にドロップします。



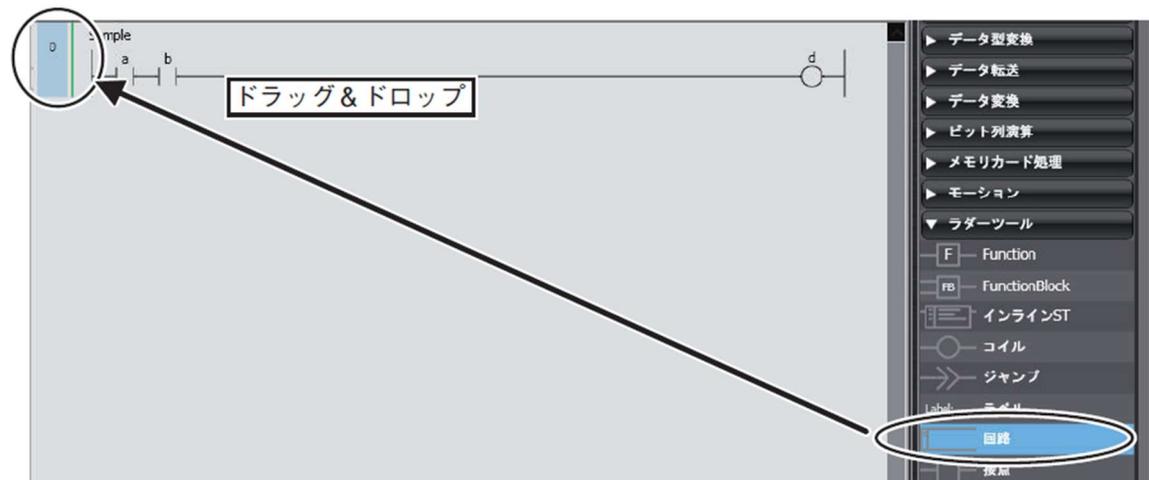
方法 3: ローカル変数テーブルに変数を登録する前に、変数名をラダーエディタ（または ST エディタ）上で入力する
 対象の接点を選択し、[変数を入力] 部分をクリックして、直接変数を入力します。



●回路の挿入

挿回路の挿入は、つぎの方法があります。

方法 1: ツールボックスの [ラダーツール] から [回路] を選択し、回路を挿入する箇所の直前の回路の先頭にドラッグ&ドロップ



方法 2: 回路の先頭を選択した状態で[R]キーを押す（回路の上に挿入する場合は[Shift]+[R]キーを押す）

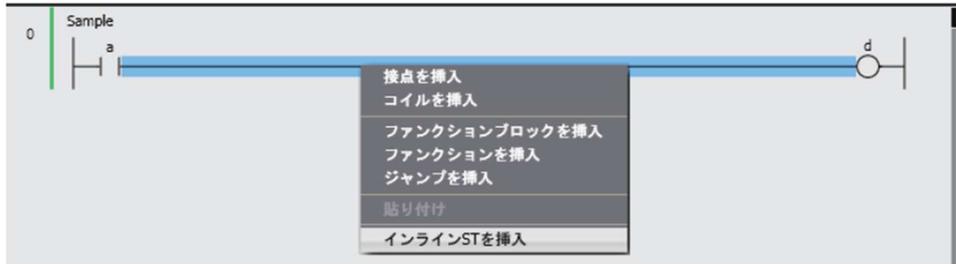
方法 3: 回路を右クリックし、[回路を挿入]（上）または [回路を挿入]（下）を選択



●インライン ST の挿入

インライン ST とは、ラダー図に ST 言語を記述できる回路部品のことです。ラダー図の中で ST 言語の処理アルゴリズムを記述することができます。

インライン ST を挿入する箇所の接続線を選択し、右クリックして [インライン ST を挿入] を選択します。または、ツールボックスの [ラダーツール] から [インライン ST] をドラッグ&ドロップします。



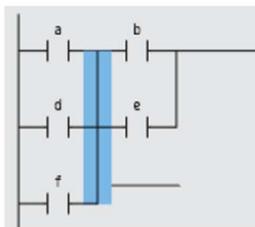
●インライン ST の削除

削除するインライン ST の回路部品を選択し、右クリックして [削除] を選択します。または、[Delete]キーを押します。



●接続線の接続先変更

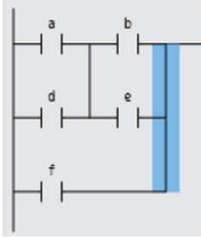
次の回路例で、接続先を変更したい接続線をドラッグします。



接続したい接続線の位置にドロップします。
ポップアップメニューが表示されます。



「接続先変更」を選択します。
ラダー回路が変更されます。



●ファンクションブロック型命令の挿入

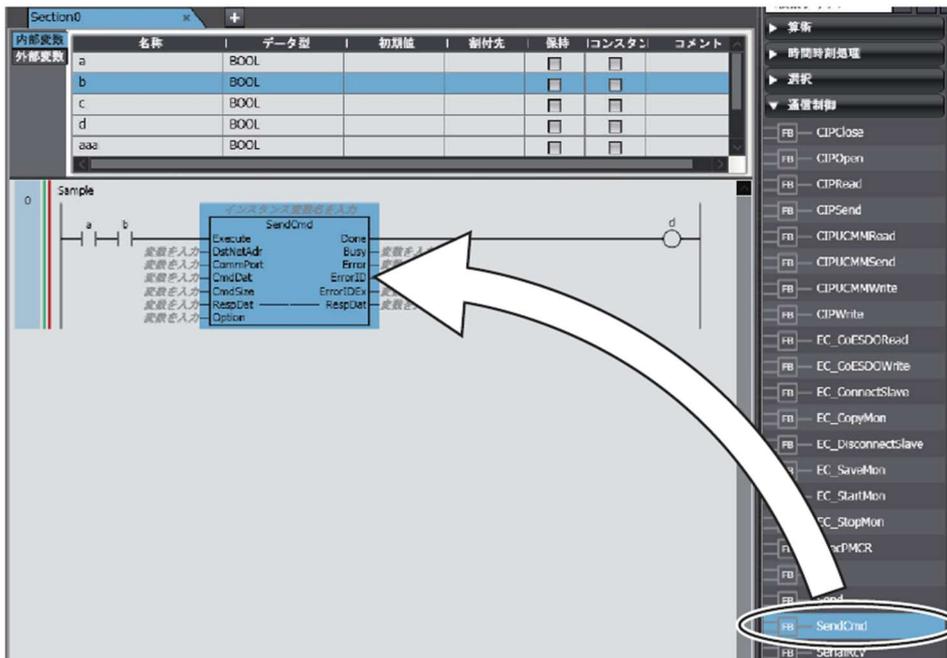
ファンクションブロック型命令、またはユーザ定義のファンクションブロック（以降、総称して「ファンクションブロック」）を挿入する方法には以下があります。

ファンクションブロック型命令の挿入方法

- 方法 1: ツールボックスからファンクションブロック型命令をドラッグ&ドロップで挿入
- ユーザ定義のファンクションブロックおよびファンクションブロック型命令共通の挿入方法
- 方法 2: ツールボックスからファンクションブロックをドラッグ&ドロップで挿入
- 方法 3: 右クリックして [ファンクションブロックを挿入] を選択
- 方法 4: 挿入位置へカーソルを移動し、[F]キーを押す

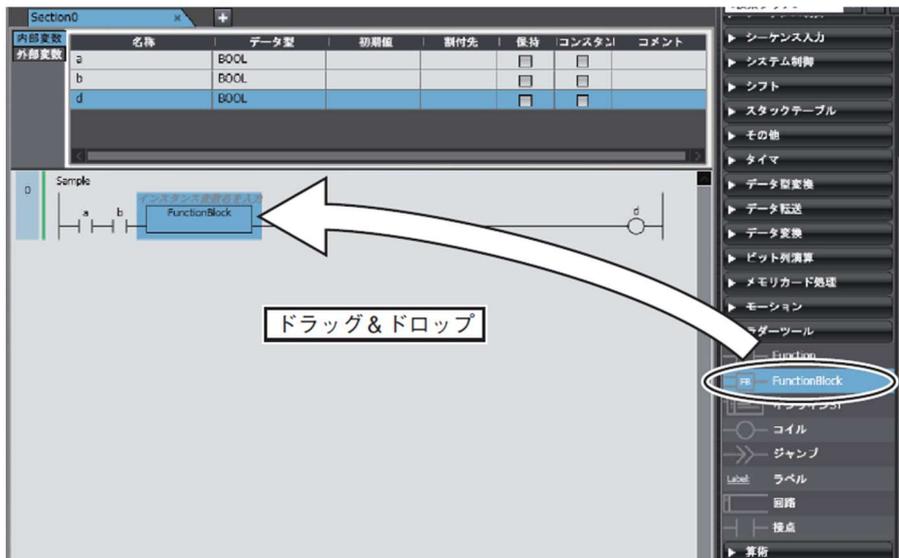
方法 1: ツールボックスからファンクションブロック型命令をドラッグ&ドロップで挿入

ツールボックスから使用したいファンクションブロックを選択し、挿入する箇所にドラッグ&ドロップします。



挿入されたファンクションブロック上部の「インスタンス変数名を入力」をクリックして、ファンクションブロックのインスタンス名を入力します。ローカル変数テーブルにファンクションブロックインスタンスが登録されている場合は、このインスタンスを選択することもできます。

方法 2: ツールボックスからファンクションブロックをドラッグ&ドロップで挿入
 ツールボックスの [ラダーツール] から、[Function Block] を選択し、挿入する箇所にドラッグ & ドロップします。

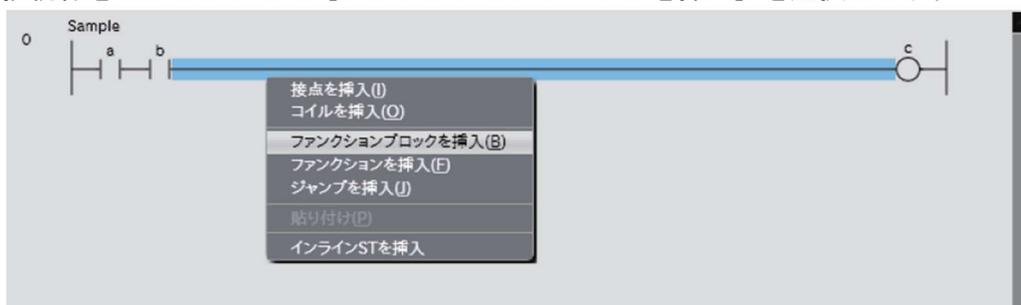


挿入されたファンクションブロックの [FunctionBlock] をクリックし、先頭の文字を入力するか、[Delete]キーを押してから[Ctrl]+[Space]キーを押します。ドロップダウンリストからファンクションブロックまたはファンクションブロック型命名を選択します。



挿入されたファンクションブロックの [インスタンス変数名を入力] をクリックして、ファンクションブロックのインスタンス名を入力します。ローカル変数テーブルにファンクションブロックインスタンスが登録されている場合は、このインスタンスを選択することもできます。

方法 3: 右クリックして [ファンクションブロックを挿入] を選択
 接続線を右クリックして [ファンクションブロックを挿入] を選択します。



挿入されたファンクションブロックの [ファンクションブロック定義名を入力] をクリックして、ファンクションブロックの定義名を入力し、[インスタンス変数名を入力] をクリックして、ファンクションブロックのインスタンス名を入力します。ローカル変数テーブルにファンクシ

ンブロックインスタンスが登録されている場合は、このインスタンスを選択することもできます。

方法 4: 挿入位置へカーソルを移動し、[F]キーを押す
挿入位置接続線を選択した状態で[F]キーを押します。
空のファンクションブロックが挿入されます。
以降の手順は方法 3 と同じです。

● ファンクション型命令の挿入

ファンクション型命令、またはユーザ定義のファンクション(以降、総称して「ファンクション」)を挿入する方法には以下があります。

ファンクション型命令の挿入方法

方法 1: ツールボックスからファンクション型命令をドラッグ&ドロップで挿入する

ユーザ定義のファンクションおよびファンクション型命令共通の挿入方法

方法 2: ツールボックスからファンクションの回路をドラッグ&ドロップで挿入する

方法 3: 右クリックして [ファンクションを挿入] を選択する

方法 4: 挿入位置へカーソルを移動し、[I]キーを押す

● ファンクションブロック、ファンクションへのパラメータ入力

挿入したファンクションブロックもしくはファンクションにパラメータを入力するには、次の方法があります。

入力対象の型に一致した変数がドロップダウンリストに表示されます。

グローバル変数の場合

方法 1: あらかじめグローバル変数テーブルに登録された変数をラダーエディタ上でドロップダウンリストから選択する

ローカル変数の場合

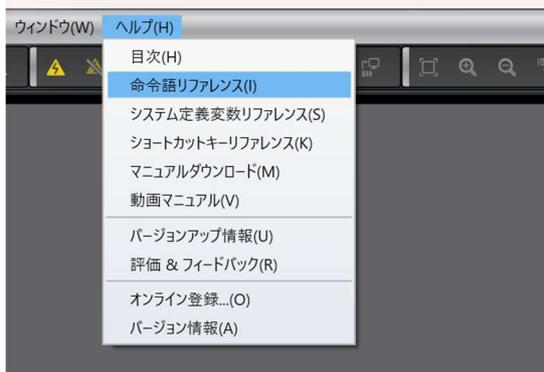
方法 1: あらかじめローカル変数テーブルに登録された変数をラダーエディタ上でドロップダウンリストから選択する

方法 2: 同じ POU のローカル変数テーブルから変数をラダーエディタ上にドラッグ&ドロップする

方法 3: ローカル変数テーブルに変数を登録する前に、変数名をラダーエディタ (または ST エディタ) 上で入力する。

●本課題で使用する命令（一部）

ヘルプ→命令語リファレンスでオンラインヘルプを開きます。



- 1秒経過で使用できる命令
 - 命令語・ST構文リファレンス→タイマ命令
 - ◇ TON（オンディレータイマ）
 - ◇ 1秒はT#1sと表します。
- 0.5秒ON→0.5秒OFFで使用できる命令
 - 命令語・ST構文リファレンス→その他命令
 - ◇ Get**Clk（クロックパルス取得グループ）

※上記以外にも実現方法はいろいろあります。

参考資料 9 タスク

「NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)」より

5-2-1 タスクとは

タスクとは、I/O リフレッシュやユーザプログラム実行などの一連の処理に、実行条件と実行順序を指定する機能です。

タスクには、実行条件および実行優先度によって、以下の 3 種類があります。

タスクの種類	タスク数	タスク実行優先度	定義	おもな処理内容
プライマリ定周期タスク	1	4	設定したタスク周期に 1 回実行します。どのタスクよりも最優先に実行します。プライマリ定周期タスクのモーション制御命令および EtherCAT の通信はプライマリ定周期タスクの周期で実行します。	I/O リフレッシュ、ユーザプログラム実行、モーション制御
定周期タスク	0~1	5*1	設定したタスク周期に 1 回実行します。プライマリ定周期タスクの次の優先度で実行します。実行優先度 5 の定周期タスクのモーション制御命令および EtherCAT の通信は実行優先度 5 の定周期タスクの周期で実行します。	I/O リフレッシュ、ユーザプログラム実行、モーション制御
	0~3	16*2、17、18	設定したタスク周期に 1 回実行します。実行優先度 16 の定周期タスクのモーション制御命令および EtherCAT の通信はプライマリ定周期タスクの周期で実行します。	タスク実行優先度により、処理内容が異なります。 実行優先度 16：I/O リフレッシュ、ユーザプログラム実行 実行優先度 17、18：ユーザプログラム実行
イベントタスク	0~32	8、48	指定した実行条件が成立したときに 1 回だけ実行します。	ユーザプログラム実行

参考資料 10 オフラインデバッグ

「NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)」より

●ロジック確認方法

必要であれば、以下の方法でロジック確認を行うことが可能です。

プログラムのロジックの確認

プログラムのロジックの正当性を確認するため、シミュレータの動作を止めたり、1ステップずつ実行したりすることができます。シミュレーションウィンドウのボタンから以下の操作ができます。

操作名	操作内容
ブレークポイント	プログラム上の指定した位置で、プログラム実行が一時停止されます。
ステップ実行	ST 言語のプログラムでは 1 行、ラダー図言語のプログラムでは 1 命令を実行すると、シミュレータが一時停止されます。
連続ステップ実行	一定間隔で連続してステップ実行されます。
一時停止	シミュレーションの実行が一時停止されます。
ステップイン実行	ファンクションブロックまたはファンクションのソース内をステップ実行されません。
ステップアウト実行	ステップイン実行中のファンクションブロック、またはファンクションが終了するまで実行されます。
1 周期実行	現在実行しているタスクが 1 周期分実行されます。次の周期のプログラム先頭で一時停止されます。
条件ブレークポイント	ブレークポイントに停止条件を設定し、条件成立時だけプログラム実行が一時停止されます。



●オンラインエディット

オンラインエディットとは、シミュレーターを停止せずに（シミュレーター動作中）にユーザプログラムの一部を変更・追加する機能です。

以下をオンラインエディットすることができます。

- ・ POU（プログラム、FB、FUN）
ラダー図言語で記述されたプログラムの場合はセクションを選択します。
- ・ グローバル変数

必要であれば、以下の方法でロジック確認を行うことが可能です。

- 1 オンラインエディット対象を選択します。
- 2 「プロジェクト」メニューから「オンラインエディット」-「開始」を選択します。
- 3 変更を行います。
- 4 「プロジェクト」メニューから「オンラインエディット」-「変更を転送」を選択します。
- 5 反映確認を行います。
- 6 オンラインエディット後のユーザプログラムで動作が開始されます。

参考資料 11 データトレース

「Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)」より

● グラフの切り替え

BOOL 型データ用のデジタルチャート、それ以外のデータ用のアナログチャート、モーション軸の動作確認用の 3D モーションモニタ表示の 3 種類のグラフの表示／非表示を切り替えます。また、デジタルチャートとアナログチャートを 1 つのチャートに重ねて表示することもできます。

操作手順

各ボタンをクリックします。



ボタンの詳細は次のとおりです。

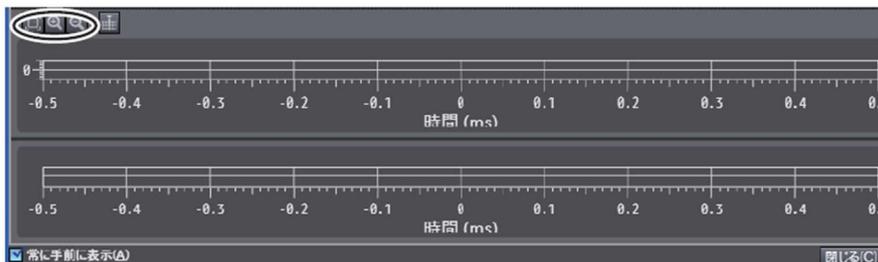
ボタン	機能
	BOOL 型データ用のデジタルチャートを表示／非表示します。
	BOOL 型以外のデータ用のアナログチャートを表示／非表示します。
	デジタルチャートとアナログチャートを 1 つのチャートに重ねて表示／非表示します。
	モーション軸の動作確認用の 3D モーションモニタ結果を表示／非表示します。

● グラフの拡大縮小

デジタルチャートおよびアナログチャートの表示の拡大・縮小を行います。デジタルチャートは X 軸方向のみ拡大・縮小可能です。

操作手順

- 1 【拡大】ボタン (🔍) / 【縮小】ボタン (🔍) をクリックします。



- ・各軸のスケールが変更されます。X 軸方向の拡大・縮小はデジタルチャート・アナログチャート間で連動します。

- [画面に合わせる] ボタン () をクリックすると、グラフ全体が表示されるスケールに自動設定されて表示されます。



参考

グラフの拡大縮小は次の操作で行うこともできます。

- 拡大・縮小を行う軸上にマウスカーソルを置き、ホイールを前後させると、それぞれの軸方向に拡大・縮小されます。
 - グラフ上にマウスカーソルを置き、ホイールを前後させると、カーソルの位置を中心に X 軸、Y 軸ともに拡大・縮小されます。
 - Ctrl キーを押しながら拡大したい範囲をマウスでドラッグして選択すると、選択した部分が拡大表示されます。
-

改訂履歴

改訂記号	改訂年月	改訂理由と改訂ページ
A	2025年2月17日	初版