



マシンオートメーションコントローラ NJシリーズ

汎用シリアル (RS-232C) 通信接続ガイド

オムロン株式会社

オートフォーカス・マルチコードリーダー

V320-F/V420-F/V430-Fシリーズ編

Network
Connection
Guide

SDNC-743C

著作権・商標について

スクリーンショットはマイクロソフトの許諾を得て使用しています。

Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標です。

Sysmac はオムロン株式会社製 FA 機器製品の日本およびその他の国における商標または登録商標です。

本資料に記載されている会社名・製品名は、それぞれ各社の商標または登録商標です。

目 次

1. 関連マニュアル	1
2. 用語と定義	2
3. 注意事項	3
4. 概要	4
5. 対象製品と対象ツール	5
5.1. 対象製品	5
5.2. デバイス構成	6
6. シリアル通信の設定内容	12
6.1. シリアル通信設定	12
6.2. ケーブル配線図	13
6.3. 通信接続確認例	15
7. 接続手順	16
7.1. 作業の流れ	16
7.2. コードリーダーの設定	17
7.3. コントローラの設定	22
7.4. シリアル通信の確認	41
8. 初期化方法	46
8.1. コントローラの初期化	46
8.2. コードリーダーの初期化	47
9. プログラム	48
9.1. 概要	48
9.2. 送受信シーケンス	52
9.3. 異常判断処理	53
9.4. 使用変数	54
9.5. ST 言語によるプログラム	57
9.6. プロトコルマクロデータ	62
9.7. タイムチャート	69
9.8. 異常処理	70
10. 改訂履歴	72

1. 関連マニュアル

システムを安全にご使用いただくため、システムを構成する機器・装置のマニュアルや取扱説明書などを必ず入手し、「安全上のご注意」「安全上の要点」など安全に関する注意事項を含め、内容を確認のうえ使用してください。

本資料に関連するオムロン株式会社（以下、オムロン）のマニュアルは以下のとおりです。

Man.No.	形式	マニュアル名称
SBCA-466	NJ シリーズ	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-467	NJ/NX シリーズ	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
SBCD-300	CJ シリーズ	CJ シリーズ シリアルコミュニケーションボード／ユニット ユーザーズマニュアル
SBCA-470	形 SYSMAC-SE2□□□	Sysmac Studio Version 1 オペレーションマニュアル
SBCA-468	NJ/NX シリーズ	NJ/NX シリーズ コマンドリファレンスマニュアル(基本編)
SDNC-745	V320-F/V330-F/V420-F/V430-F シリーズ	オートフォーカス・マルチコードリーダー MicroHAWK V320-F/V330-F/V420-F/V430-F シリ ーズ ユーザーズマニュアル
SDNC-735	V320-F/V330-F/V420-F/V430-F シリーズ	オートフォーカス・マルチコードリーダー MicroHAWK V320-F/V330-F/V420-F/V430-F シリ ーズ ユーザーズマニュアル 通信設定編

2. 用語と定義

用語	説明・定義
プロトコルマクロ	汎用外部機器とのデータ送受信手順（プロトコル）をシリアルコミュニケーションボードまたはユニットに記憶させ、CPU ユニットで PMCR 命令を実行することにより、汎用外部機器との送受信を可能とする機能です。
プロトコル	特定の汎用外部機器に対する独立した通信処理をデータ送受信手順としてまとめた単位を「プロトコル」と呼びます。「プロトコル」は、複数のシーケンスから構成されます。
シーケンス	ラダープログラム上の PMCR 命令から起動がかけられる独立した通信処理の単位です。起動がかけられたシーケンスは、その中のステップを順次実行します。
ステップ	メッセージの送信処理、受信処理、送信&受信処理、受信バッファクリア、ステップ待機のいずれかを実行する単位です。1つのシーケンスに最大 15 個まで設定が可能です。
送信メッセージ	通信先の汎用外部機器へ送り出す通信フレーム（コマンド）のことです。シーケンス中のステップから呼び出され、汎用外部機器に送信されます。
受信メッセージ	通信先の汎用外部機器から送られてくる通信フレーム（レスポンス）のことです。シーケンス中のステップから呼び出され、汎用外部機器から受信したデータと比較します。
受信マトリクス	汎用外部機器から送られてくる通信フレーム（レスポンス）が1つだけに特定できない場合に受信マトリクスを使用します。受信マトリクスには、複数の通信フレームを登録しておくことが可能です。
ケース	複数存在する通信フレーム（レスポンス）を受信マトリクスに登録する単位です。1つの通信フレームを1つのケースとして登録します。1つの受信マトリクスに最大15種類のケースが登録可能です。

3. 注意事項

- (1) 実際のシステム構築に際しては、システムを構成する各機器・装置の仕様をご確認のうえ、定格・性能に対し余裕を持った使い方をし、万一故障があっても危険を最小にする安全回路などの安全対策を講じてください。
- (2) システムを安全にご使用いただくため、システムを構成する各機器・装置のマニュアルや取扱説明書などを入手し、「安全上のご注意」「安全上の要点」など安全に関する注意事項を含め、内容を確認のうえ使用してください。
- (3) システムが適合すべき規格・法規または規制に関しては、お客様自身でご確認ください。
- (4) 本資料の一部または全部を、オムロン株式会社の許可なしに複写、複製、再配布することを禁じます。
- (5) 本資料の記載内容は、2023 年 2 月時点のものです。
本資料の記載内容は、改良のため予告なく変更されることがあります。

本資料で使われているマークには、次のような意味があります。

 警告	正しい取り扱いをしなければ、この危険のために、軽傷・中程度の傷害を負ったり万一の場合には重傷や死亡に至ったりする恐れがあります。また、同様に重大な物的損害をもたらす恐れがあります。
---	---

 注意	正しい取り扱いをしなければ、この危険のために、時に軽傷・中程度の傷害を負ったり、あるいは物的損害を受けたりする恐れがあります。
---	--



安全上の要点

製品を安全に使用するために実施または回避すべきことを示します。



使用上の注意

製品の動作不能、誤動作、または性能・機能への悪影響を予防するために実施または回避すべきことを示します。



参考

必要に応じて読んでいただきたい項目です。

知っておくと便利な情報や、使用するうえで参考となる内容について説明しています。

図記号の説明



●記号は、強制を意味しています。
具体的な内容は、●の中と文章で示します。
左図の場合は、「一般的な指示」を表します。

4. 概要

本資料は、オムロン株式会社（以下、オムロン）製コードリーダー（V320-F/V420-F/V430-F シリーズ）を、マシンオートメーションコントローラ NJ シリーズ（以下、コントローラ）とシリアルコミュニケーションユニット（以下、コントローラ）にシリアル通信で接続する手順とその確認方法をまとめたものです。

「6. シリアル通信の設定内容」と「7. 接続手順」で記載しているシリアル通信設定を通して、設定手順と設定時のポイントを理解することにより、シリアル通信接続を行うことができます。

あらかじめ準備された「Sysmac Studio プロジェクトファイル」内のユーザプログラムと「CX-Protocol プロジェクトファイル」内のプロトコルデータを使用し、コードリーダーに対する「読取りトリガ」コマンドにより、シリアル通信の接続確認を行います。

最新のプロジェクトファイルを事前に準備してください。ファイルの入手については、オムロンまでお問い合わせください。

名称	ファイル名	バージョン
Sysmac Studio プロジェクトファイル（拡張子：smc2）	OMRON_V320_V420_V430_NJ_PMC232C_V100.smc2	Ver.1.00
CX-Protocol プロジェクトファイル（拡張子：psw）	OMRON_V320_V420_V430_NJ_PMC232C_V100.psw	Ver.1.00

注意

本資料は、当該機器の通信接続に必要な配線方法および通信設定の内容とその設定手順について説明することを目的としており、本資料で使用するプログラムは、その接続作業が正しく行われたことを確認（接続確認）するために作成されたものです。恒常的に現場で利用されることを前提に作成されたプログラムではありませんので、機能面や性能面については十分に考慮されていません。実際のシステムを構築される場合には、本資料に掲載の配線方法や通信設定の内容と設定手順を参考にいただき、プログラムについてはお客様の目的に応じたものを新規に設計していただく必要があります。



5. 対象製品と対象ツール

5.1. 対象製品

接続を保証する対象機器は以下のとおりです。

メーカー	名称	形式	バージョン
オムロン	NJ シリーズ CPU ユニット	形 NJ501-1500 形 NJ501-1400 形 NJ501-1300 形 NJ301-□□□□	5.2. 項 で 記載したバージョン以降
オムロン	シリアルコミュニケーションユニット	形 CJ1W-SCU□1-V1 形 CJ1W-SCU□2	
オムロン	コードリーダー	形 V320-F□□□□□□□-□□□ 形 V420-F□□□□□□□-□□□ 形 V430-F□□□□□□□-□□□	



参考

本資料では、上記対象機器の中から 5.2.項に記載された機器を使用し接続確認を行っています。上記対象機器の中で、5.2.項に記載されていない機器を使用する場合は、本資料の内容を参考に接続確認を行ってください。



参考

本資料は機器の通信接続確立までの手順について記載したものであって、機器個別の操作や設置および配線方法に関しては記載しておりません。

上記製品（通信接続手順以外）の詳細に関しましては、対象製品の取扱説明書を参照するか、オムロンまでお問い合わせください。



使用上の注意

本資料の接続手順および接続確認では、上記対象機器の中から 5.2.項に記載された形式およびバージョンの機器を使用しています。

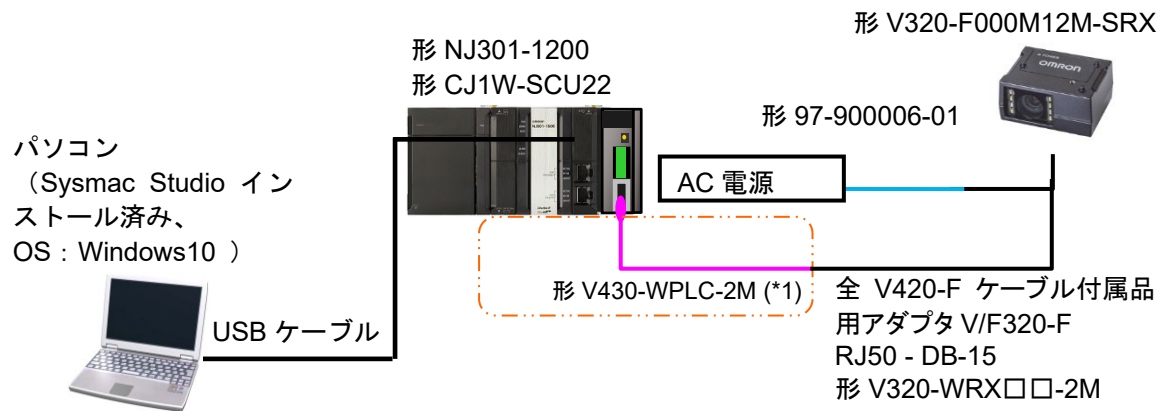
5.2 項に記載されたバージョンより古いバージョンの機器は使用できません。

上記対象機器の中から 5.2.項に記載されていない形式、あるいは 5.2.項に記載されているバージョンより新しいバージョンの機器を使用する場合は、取扱説明書などにより仕様上の差異を確認のうえ、作業を行ってください。

5.2. デバイス構成

本資料の接続手順を再現するための構成機器は以下のとおりです。

- ・ V320-F を使用した構成

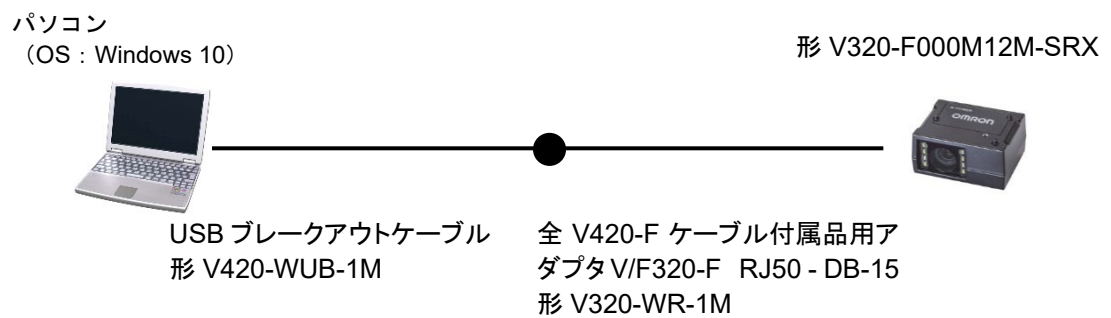


メーカー	名称	形式	バージョン
オムロン	NJ シリーズ CPU ユニット (内蔵 EtherNet/IP ポート)	形 NJ301-1200	Ver.1.19
オムロン	電源ユニット	形 NJ-PA3001	
オムロン	シリアルコミュニケーションユニット	形 CJ1W-SCU22	Ver.2.1
オムロン	Sysmac Studio	形 SYSMAC-SE2□□□	Ver.1.28
オムロン	Sysmac Studio プロジェクトファイル	OMRON_NJ_V320_V420_V430_PMCRC232C_V100.smc2	Ver.1.00
オムロン	CX-Protocol プロジェクトファイル	OMRON_NJ_V320_V420_V430_PMCRC_V100.psw	Ver.1.00
—	パソコン(OS : Windows 10)	—	
—	USB ケーブル (USB2.0 準拠 B コネクタ)	—	
オムロン	コードリーダー	形 V320-F000M12M-SRX	Ver.2.1.0
オムロン	全 V420-F ケーブル付属品用アダプタ V/F320-F RJ50 - DB-15	形 V320-WRX□□-2M	
オムロン	オムロン製プログラマブルコントローラ (CS/CJ/NJ) 接続用 RS-232C ケーブル	形 V430-WPLC-2M (*1)	
オムロン	電源	形 97-900006-01	
	AC 電源		

*1:RS-232C ケーブルを自作する場合、6.2 ケーブル配線図を参照の上作成してください。

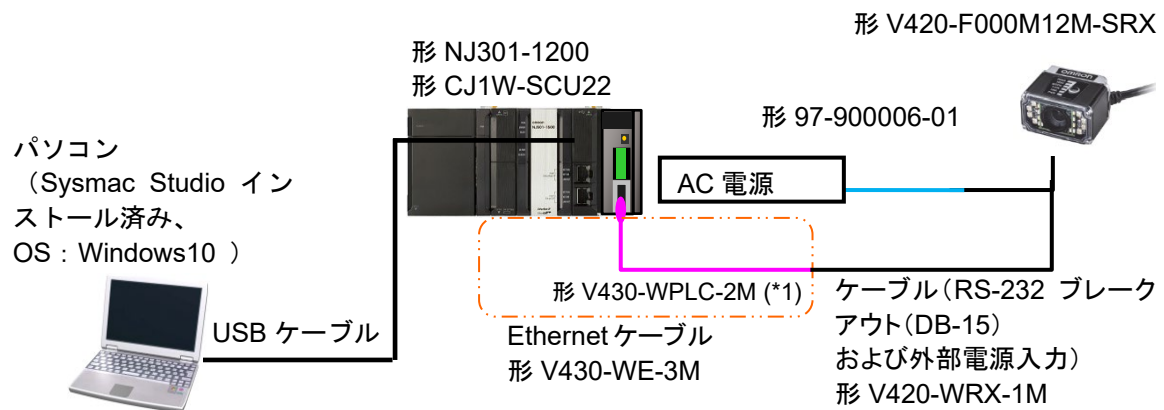
WebLink は RS-232C 接続では使用することができません。

WebLink を使用したい場合は、以下の USB 接続による構成機器が必要です。



メーカー	名称	形式	バージョン
	パソコン (OS : Windows 10)		
オムロン	コードリーダー	形 V320-F000M12M-SRX	Ver.2.1.0
オムロン	全 V420-F ケーブル付属品用アダプタ V/F320-F RJ50 - DB-15	形 V320-WR-1M	
オムロン	USB ブレークアウトケーブル	形 V420-WUB-1M	

・ V420-F を使用した構成

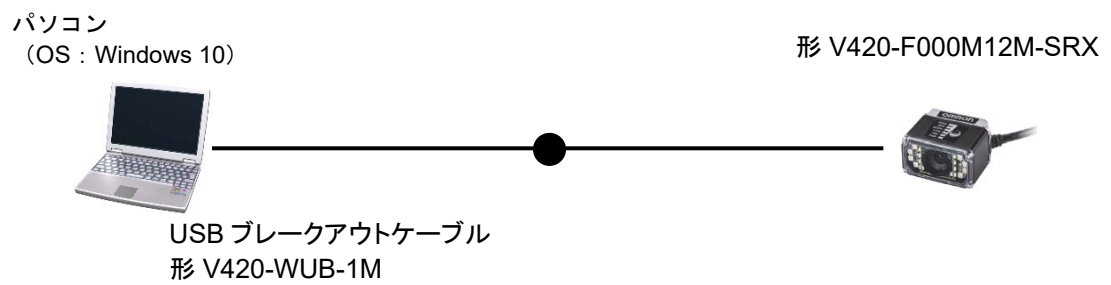


メーカー	名称	形式	バージョン
オムロン	NJ シリーズ CPU ユニット (内蔵 EtherNet/IP ポート)	形 NJ301-1200	Ver.1.19
オムロン	電源ユニット	形 NJ-PA3001	
オムロン	シリアルコミュニケーションユニット	形 CJ1W-SCU22	Ver.2.1
オムロン	Sysmac Studio	形 SYSMAC-SE2□□□	Ver.1.28
オムロン	Sysmac Studio プロジェクトファイル	OMRON_NJ_V320_V420_V430_PMCR232C_V100.smc2	Ver.1.00
オムロン	CX-Protocol プロジェクトファイル	OMRON_NJ_V320_V420_V430_PMCR_V100.psw	Ver.1.00
—	パソコン(OS : Windows 10)	—	
—	USB ケーブル (USB2.0 準拠 B コネクタ)	—	
オムロン	コードリーダー	形 V420-F000M12M-SRX	Ver.2.1.0
オムロン	ケーブル (RS-232C ブレークアウト (DB-15) および外部電源入力)	形 V420-WRX-1M	
オムロン	オムロン製プログラマブルコントローラ (CS/CJ/NJ) 接続用 RS-232C ケーブル	形 V430-WPLC-2M (*1)	
オムロン	電源	形 97-900006-01	
	AC 電源		

*1:RS-232C ケーブルを自作する場合、6.2 ケーブル配線図を参照の上作成してください。

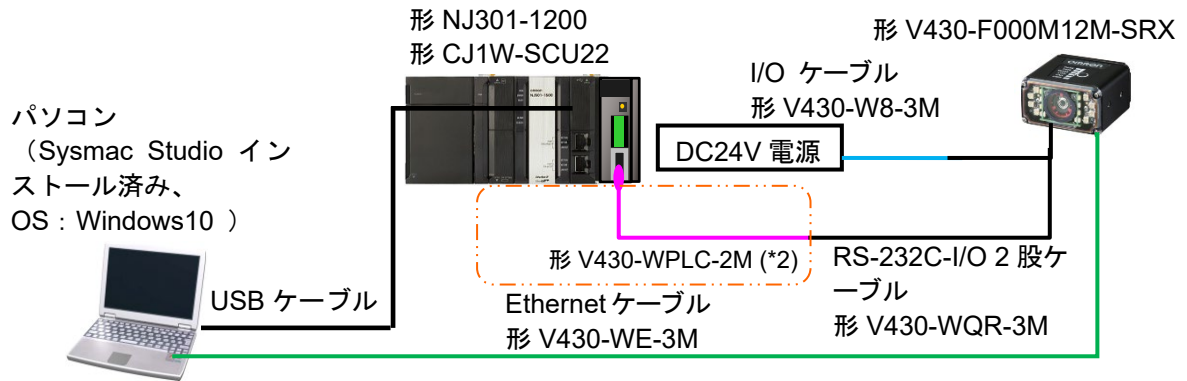
WebLink は RS-232C 接続では使用することができません。

WebLink を使用したい場合は、以下の USB 接続による構成機器が必要です。



メーカー	名称	形式	バージョン
	パソコン (OS : Windows 10)		
オムロン	コードリーダー	形 V420-F000M12M-SRX	Ver.2.1.0
オムロン	USB ブレークアウトケーブル	形 V420-WUB-1M	

- ・ V430-F を使用した構成



メーカー	名称	形式	バージョン
オムロン	NJ シリーズ CPU ユニット (内蔵 EtherNet/IP ポート)	形 NJ301-1200	Ver.1.19
オムロン	電源ユニット	形 NJ-PA3001	
オムロン	シリアルコミュニケーションユニット	形 CJ1W-SCU22	Ver.2.1
オムロン	Sysmac Studio	形 SYSMAC-SE2□□□	Ver.1.28
オムロン	Sysmac Studio プロジェクトファイル	OMRON_NJ_V320_V420_V430_PMCR232C_V100.smc2	Ver.1.00
オムロン	CX-Protocol プロジェクトファイル	OMRON_NJ_V320_V420_V430_PMCR_V100.psw	Ver.1.00
—	パソコン(OS : Windows 10)	—	
—	USB ケーブル (USB2.0 準拠 B コネクタ)	—	
オムロン	コードリーダー	形 V430-F000M12M-SRX	Ver.2.1.0
オムロン	RS-232C-I/O 2 股ケーブル	形 V430-WQR-3M	
オムロン	オムロン製プログラマブルコントローラ(CS/CJ/NJ) 接続用 RS-232C ケーブル	形 V430-WPLC-2M (*2)	
オムロン	I/O ケーブル	形 V430-W8-3M	
オムロン	Ethernet ケーブル(*1)	形 V430-WE-3M	
	DC24V 電源		

*1:Ethernet ケーブルは WebLink の接続のために使用しています。

*2:RS-232C ケーブルを自作する場合、6.2 ケーブル配線図を参照の上作成してください。



使用上の注意

オムロン株式会社より「Sysmac Studio プロジェクトファイル」の最新ファイルを事前に準備してください。

(ファイルの入手については、オムロン株式会社までお問い合わせください)



使用上の注意

Sysmac Studio と CX-Protocol を本項記載のバージョン以降に、オートアップデートしてください。なお、本項記載のバージョン以外を使用すると、7 章以降の手順に差異があることがあります。その場合は、「Sysmac Studio Version 1 オペレーションマニュアル」(SBCA-362) および「CX-Protocol オペレーションマニュアル」(SBCA-307)を参照して、手順と同等の処理を行ってください。

**参考**

構成機器、バージョンが異なる場合再現できないことがあります。構成、形式、バージョンを確認のうえ、お客様の構成と異なる場合は、オムロンまでお問い合わせください。

**参考**

本資料ではコントローラとの接続に USB を使用します。USB ドライバのインストールについては、「Sysmac Studio Version1.0 オペレーションマニュアル」(SBCA-470)の「付録 A-1 USB ケーブルで直接接続する場合のドライバのインストール方法」を参照してください。

**参考**

DC24V 電源（コードリーダー用）に使用可能な電源仕様は、『オートフォーカス・マルチコードリーダー MicroHAWK V320-F/V330-F/V420-F/V430-F シリーズ ユーザーズマニュアル (SDNC-745)』を参照してください。

6. シリアル通信の設定内容

本資料で設定する通信パラメータ、ケーブル配線などの仕様を示します。



参考

本章に記載している設定以外の設定で通信を行う場合には、プログラムの変更が必要となります。プログラムについては、「9. プログラム」を参照してください。

6.1. シリアル通信設定

シリアル通信の設定内容は、以下になります。

設定項目	SCU ユニット※	コードリーダー
ユニット番号	0	—
通信（接続）ポート	ポート 2 (RS-232C)	—
シリアル通信モード	プロトコルマクロ	—
データ長	8 ビット	8 ビット（初期値）
ストップビット	1 ビット	1 ビット（初期値）
パリティ	なし	なし（初期値）
伝送速度（ボーレート）	9,600bps（初期値）	9,600bps
プロトコルマクロ伝送方式	全二重	—
ヘッダ	—	なし（初期値）
フッタ	—	<CR+LF>（初期値）
ホストポートプロトコル	—	ポイントツーポイント（初期値）

※SCU ユニット：シリアルコミュニケーションユニット



使用上の注意

シリアルコミュニケーションユニットは「形 CJ1W-SCU22」を使用し、ユニット番号は「0」、通信（接続）ポートは「ポート 2」を使用することを前提としています。この条件以外で接続される場合は「9. プログラム」を参照のうえ、割付リレーエリアおよび PMCR 命令のコントロールデータを変更して、プログラムを作成してください。

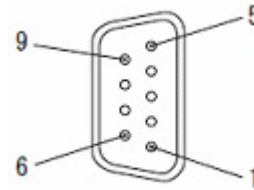
6.2. ケーブル配線図

ケーブル配線の詳細については「CJ シリーズ シリアルコミュニケーションボード／ユニット ユーザーズマニュアル」(SBCD-300)の「第 3 章 取付けと接続」を参照してください。
コネクタ形状および信号線（ピンアサイン）を確認してからケーブルを作成してください。

■コネクタ形状および信号線（ピンアサイン）

<形 CJ1W-SCU22> 適合コネクタ：D-SUB 9 ピン（CJ1W-SCU22 側コネクタ形状:メス）

ピン番号	略称	信号名称	入出力
1	FG	シールド	---
2	SD	送信データ	出力
3	RD	受信データ	入力
4	RS	送信要求	出力
5	CS	送信可	入力
6	5V	電源	---
7	DR	データセットレディ	入力
8	ER	データターミナルレディ	出力
9	SG	信号用接地	---
Shell	FG	シールド	---

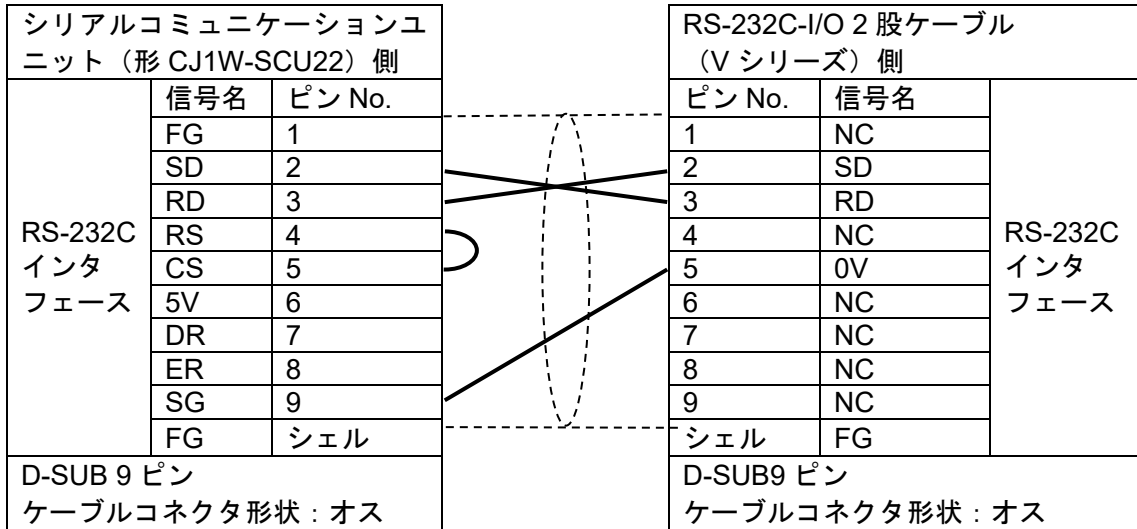


<V シリーズ RS-232C ケーブルコネクタ> 適合コネクタ：D-SUB 9 ピン（V シリーズ側コネクタ形状:メス）

ピン番号	信号名	ピン配置図
1	-	
2	SD	
3	RD	
4	-	
5	0V	
6	-	
7	-	
8	-	
9	-	

■RS-232C ケーブル／ピン配線図

以下の配線を参考に RS-232C ケーブルを作成してください。



<V430-F の場合>

形 NJ501-1500
形 CJ1W-SCU22



DC24V 電源

I/O ケーブル

形 V430-W8-3M

形 V430-F00M12M



RS-232C ケーブル

RS-232C-I/O 2 股ケーブル
形 V430-WQR-3M

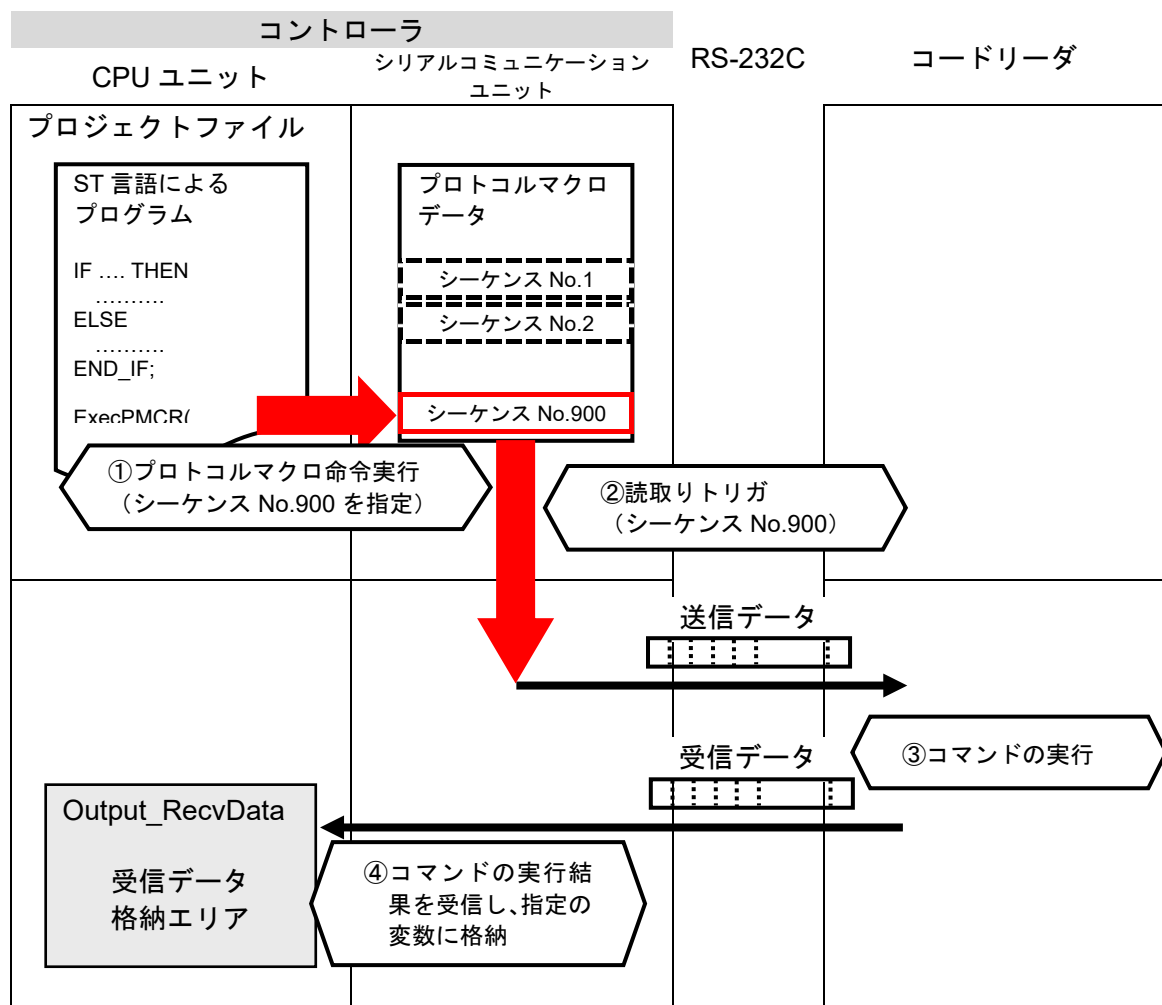
6.3. 通信接続確認例

本資料では、ストラクチャード・テキスト（以下、ST）言語によるプログラムとプロトコルマクロデータを使用し、コントローラからコードリーダーに対してメッセージを送受信する場合を例とします。

コントローラとコードリーダー間では、「読取りトリガ（シーケンス No.900）」のメッセージを送受信します。シーケンスの動作概要を以下に示します。

●シーケンス動作概要

- ①ST 言語によるプログラムで、プロトコルマクロデータのシーケンス No.900 を指定したプロトコルマクロ命令（ExecPMCR 命令）を実行
- ②プロトコルマクロデータからシーケンス No.900 の「読取りトリガ」コマンドが選択され、コードリーダーに送信
- ③コントローラからの送信データに応じ、コードリーダーにてコマンドを実行
- ④コードリーダーでのコマンド実行結果をコントローラで受信し、指定の変数に格納

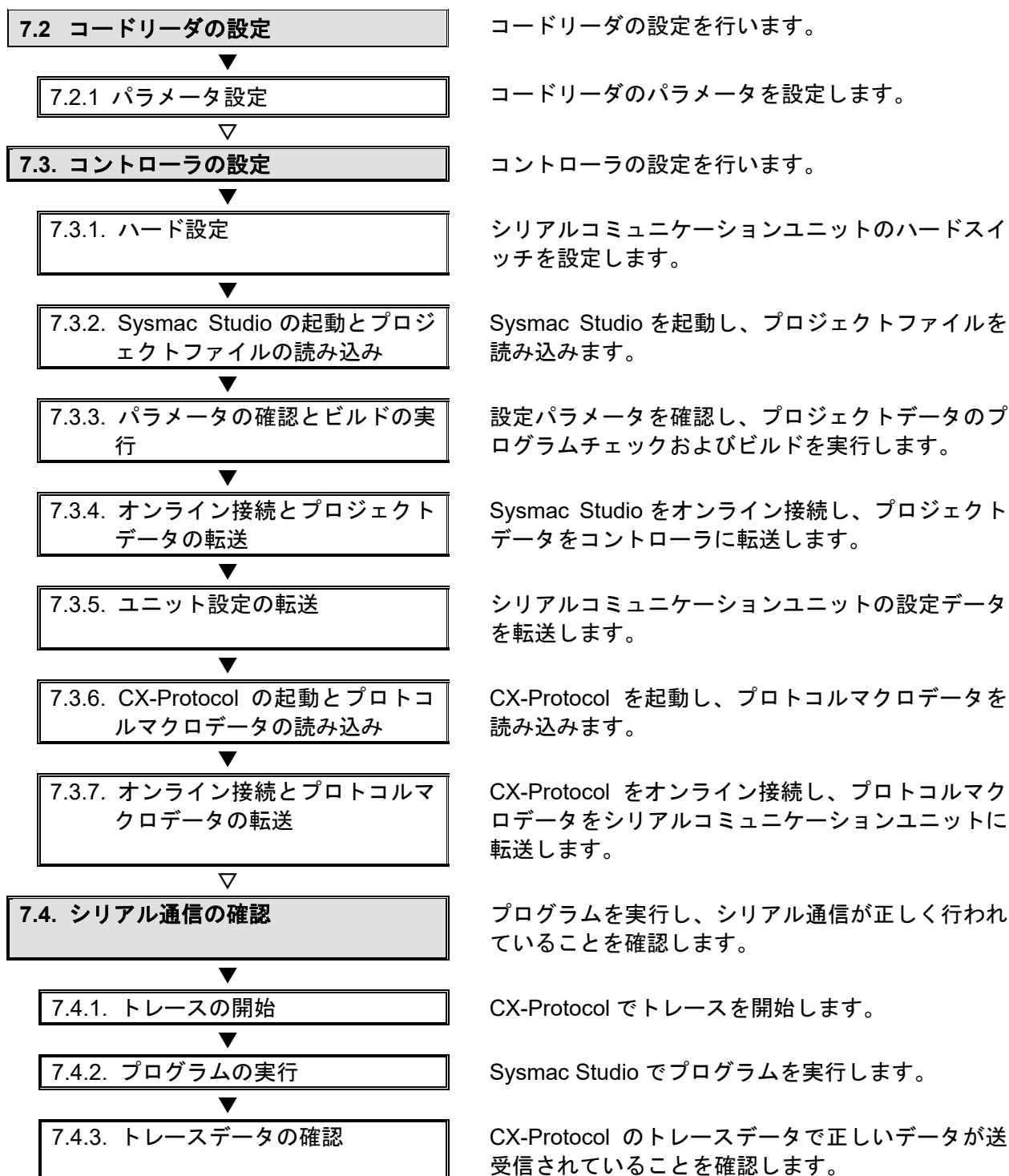


7. 接続手順

本章では、コードリーダをコントローラにシリアル通信で接続する手順について記載します。本資料では、コントローラおよびコードリーダが工場出荷時の初期設定状態であることを前提として説明します。機器の初期化については「8. 初期化方法」を参照してください。

7.1. 作業の流れ

コントローラをシリアル通信接続設定する手順は以下のとおりです。



7.4.4. 受信データの確認

Sysmac Studio でコントローラの変数に正しいデータが書き込まれていることを確認します。

7.2. コードリーダーの設定

コードリーダーの設定を行います。



使用上の注意

コードリーダーのパラメータを設定するためにパソコンを使用します。

パソコンの状態に応じてパソコンの設定を変更する必要がありますので、ご注意ください。

7.2.1. パラメータ設定

コードリーダーのパラメータを設定します。

使用するパソコンの IP アドレスを「192.168.188.100」、サブネットマスクを [255.255.0.0] に設定してください。

1 [V320-F,V420-F を使用する場合]

P.7 または P.9 の構成の通り、V320/V420 とパソコンの USB 接続を行ってください。接続ができれば、手順 4. から進めてください。

※V320,V420 を使用するにはドライバのインストールが必要です。ドライバのインストール方法については、『オートフォーカス・マルチコードリーダー MicroHAWK V320-F/V330-F/V420-F/V430-F シリーズ ユーザーズマニュアル (SDNC-745)』の「2-2-2 WebLink への接続」を参照してください。

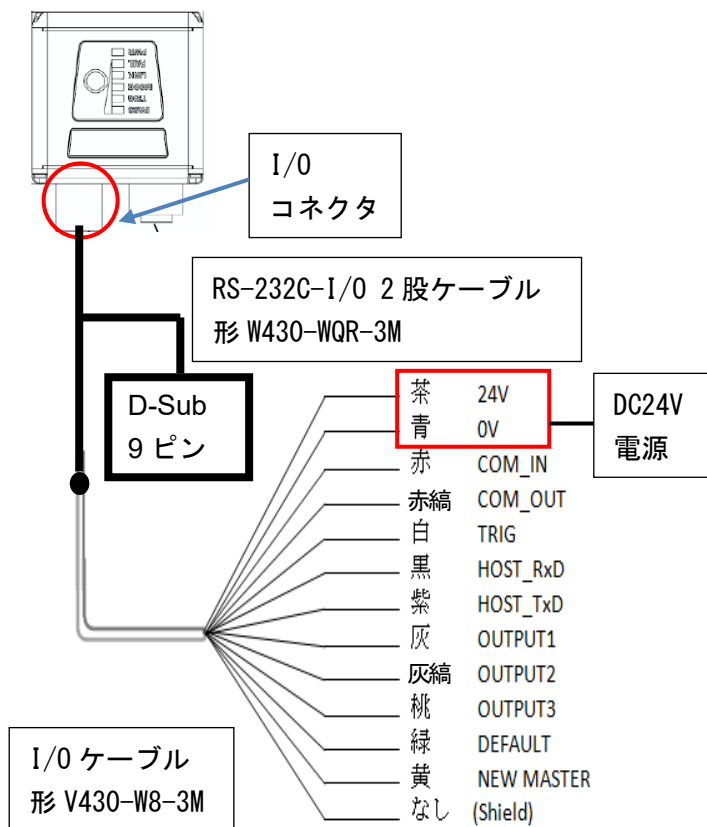
[V430-F を使用する場合]

[RS-232C-I/O 2 股ケーブル]
形 V430-WQR-3M を [I/O コネクタ]
に接続します。
さらに形 V430-WQR-3M の I/O
コネクタを I/O ケーブル形
V430-W8-3M に接続します。

形 W430-W8-3M の 24V 線と 0V
線を DC24V 電源に接続します。

※本資料では、I/O ケーブルの電
源線のみを接続して確認して
います。他の線を短絡しないよ
うに注意ください。

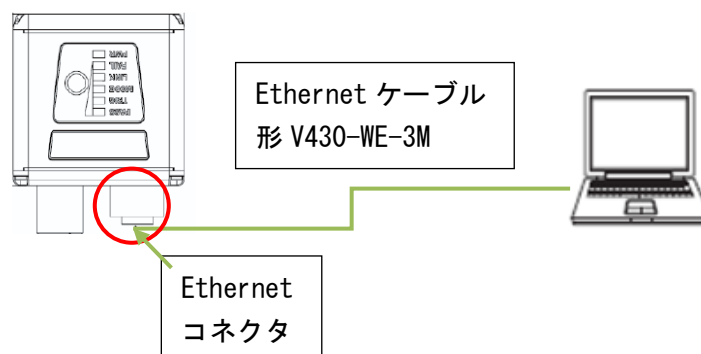
※シールド線は、必要に応じて接地
してください。接地方法につい
ては、『オートフォーカス・マルチ
コードリーダー MicroHAWK
V320-F/V330-F/V420-F/V430-F
シリーズ ユーザーズマニュアル
(SDNC-745)』の「付録」-「接
地と電源」を参照してください。



2 [V430-F シリーズのみ]

コードリーダーの Ethernet コネクタとパソコンを Ethernet ケーブル V430-WE-3M で接続します。

DC24V 電源を ON します。

**3** パソコンの IP アドレスの設定をします。

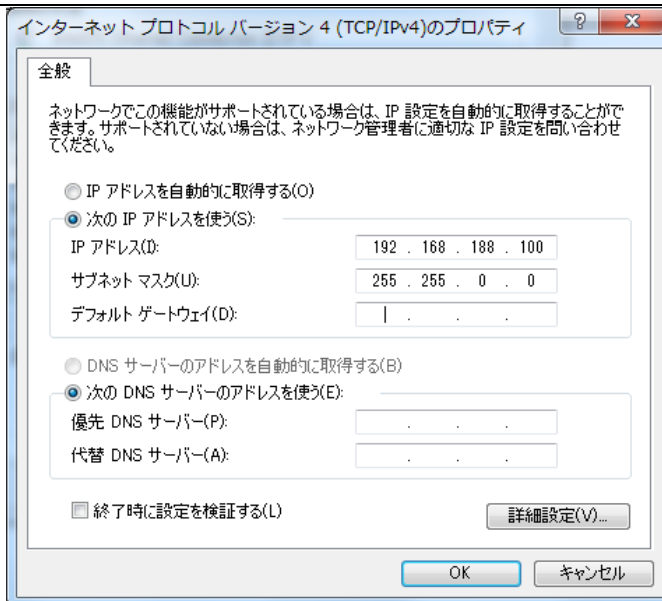
IP アドレスは

「192.168.188.100」、

サブネットマスクは

「255.255.0.0」を入力してください。

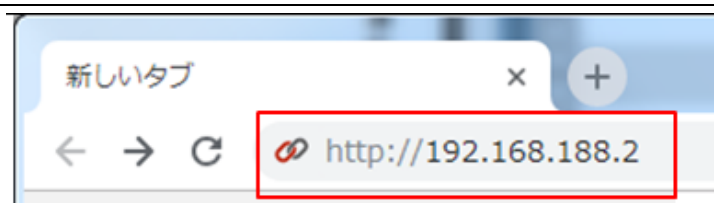
Windows 7 で右の画面を開く手順は、次の手順 4.を参照ください。



- 4**
1. [コントロールパネル] - [ネットワークと共有センター] に移動します。
 2. [ローカルエリア接続] ボタンをクリックします。[ステータス] ダイアログボックスで、[プロパティ] ボタンをクリックします。
 3. [ローカルエリア接続のプロパティ] ダイアログボックスで、[インターネットプロトコルバージョン 4 (TCP / IPv4)] を選択し、[プロパティ] ボタンをクリックします。
 4. [OK] ボタンをクリックします。

5 任意のブラウザを起動し、「http://192.168.188.2」と入力します。

推奨ブラウザは、「Google Chrome」です。



6 WebLink の起動画面が表示されたら、手順 8.に進んでください。

WebLink で接続できない場合は、手順 7.を確認してください。

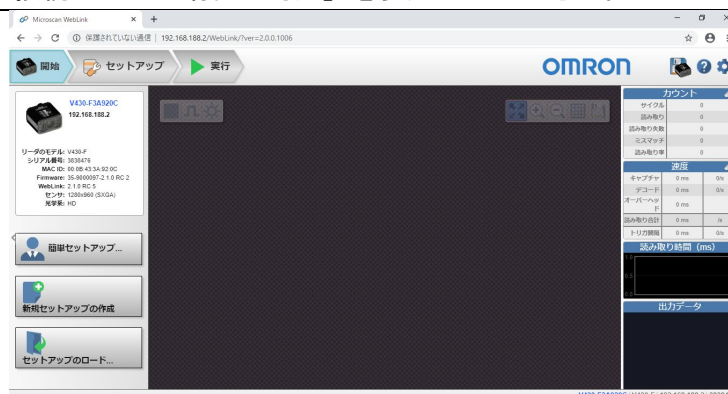


7 WebLink の起動画面が表示されない場合、コードリーダーとパソコン間の通信が確立していませんので、以下を確認してください。

- ・コードリーダーとパソコンが物理的に接続されていること。
→手順 1、手順 2. を参考に接続を確認してください。
- ・パソコンの IP アドレスとコードリーダーの IP アドレスの設定が正しいこと。
→手順 4.を参考にパソコンの IP アドレスを設定してください。

その他の解決方法は、『オートフォーカス・マルチコードリーダー MicroHAWK V320-F/V330-F/V420-F/V430-F シリーズ ユーザーズマニュアル (SDNC-745)』の「付録」 - 「よくあるご質問」 - 「WebLink で接続できない場合の対処」を参照してください。

8 WebLink の画面が表示されます。



9 [セットアップ] タブを押下し、[読み取りサイクルシーケンス] - [サイクル] を [単発読み取り] に設定します。



- 10** 画面右上の歯車マークを押下し、「詳細」を選択します。



- 11** 詳細設定画面が表示されます。
[通信設定] タブを選択して下記の設定を確認してください。

[RS232A]

ボーレート:9600

パリティ:なし

ストップビット:1

データ長:8

[ホストプロトコル]

プロトコル選択:ポイントツーポイント

[ヘッダ(プリアンブル)]

ヘッダ:無効

[フッタ(ポストアンブル)]

フッタ:有効

フッタ文字:<CR><LF>



- 12** 赤枠のアイコンを押下して、設定をコードリーダー本体に保存します。

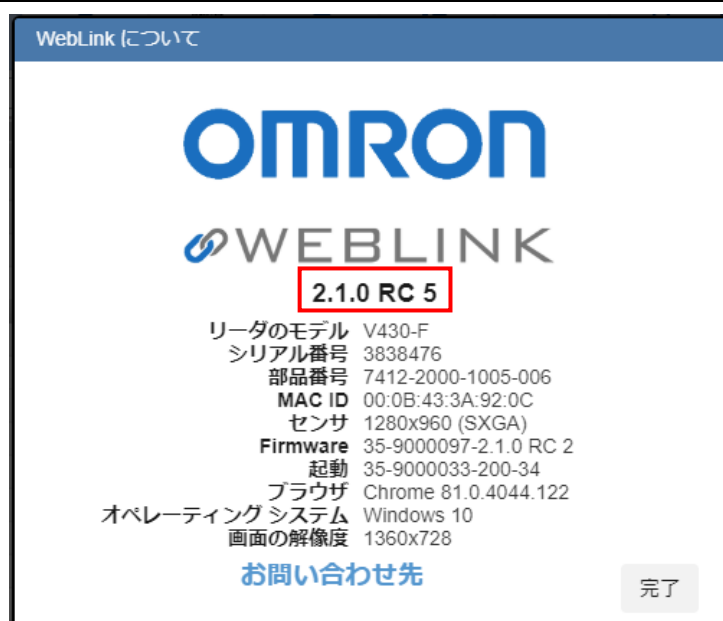


- 13** 最後にコードリーダーのバージョンを確認します。画面右上の歯車マークを押下し、[WebLinkについて...]を選択します。



- 14** [WebLinkについて]が表示されるため、コードリーダーの現在のバージョンを確認できます。

必要によりコードリーダーを最新バージョンにアップデートしてください。



7.3. コントローラの設定

コントローラの設定を行います。

7.3.1. ハード設定

シリアルコミュニケーションユニットのハードスイッチを設定します。



使用上の注意

電源 OFF 状態で設定してください。

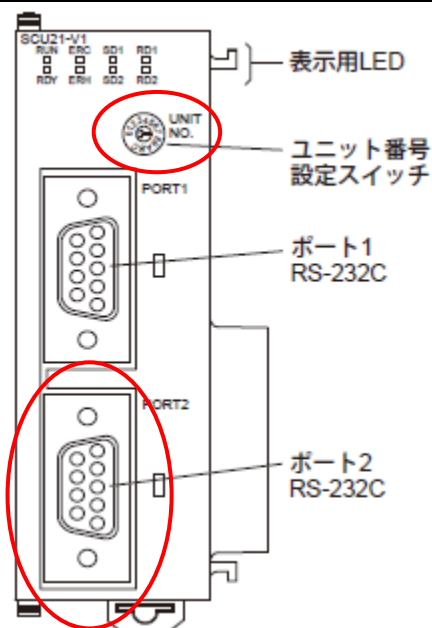
- 1 コントローラの電源がOFFの状態であることを確認します。

※電源が ON の状態であると、以降の操作を手順どおりに進めることができない場合があります。

シリアルコミュニケーションユニット前面のハードスイッチの位置を、右図をもとに確認します

「ポート 2」(RS-232C)コネクタの位置を、右図をもとに確認します。

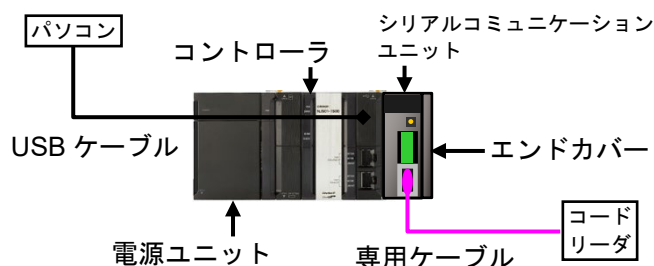
※シリアルコミュニケーションユニットの「ポート 2」の使用を前提とします。



- 2 ユニット番号設定スイッチを「0」に設定します。
(工場出荷時のユニット番号は「0」です)



- 3 右図のとおり、コントローラにシリアルコミュニケーションユニットを接続します。
シリアルコミュニケーションユニットにコードリーダーの専用ケーブルを接続します。
コントローラとパソコンの間をUSBケーブルで接続します。



7.3.2. Sysmac Studio の起動とプロジェクトファイルの読み込み

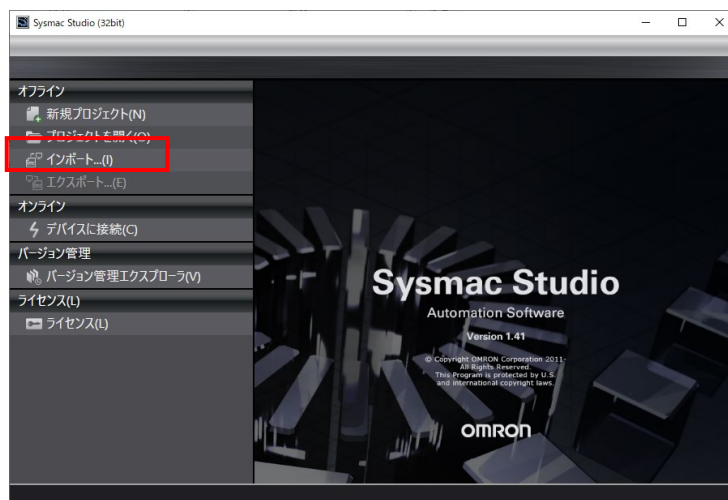
Sysmac Studio を起動し、プロジェクトファイルを読み込みます。

Sysmac Studio と USB ドライバをあらかじめパソコンにインストールしてください。

- 1** パソコンとコントローラが USB ケーブルで接続されていることを確認し、コントローラの電源を投入します。

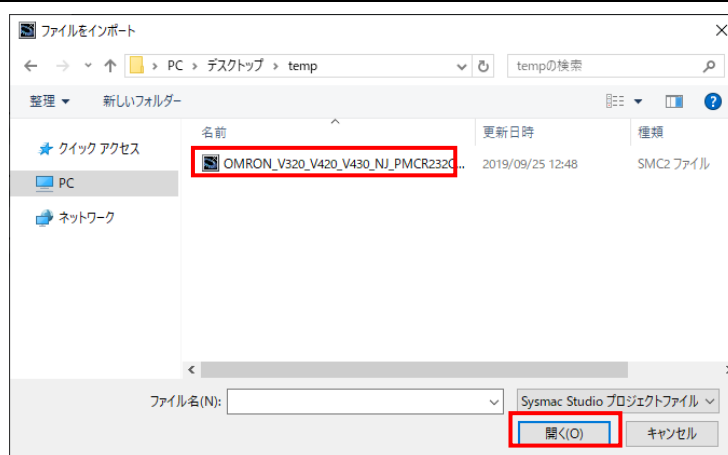
Sysmac Studio を起動し、[インポート] をクリックします。

※起動時に、アクセス権確認用のダイアログが表示される場合、起動する選択を行ってください。



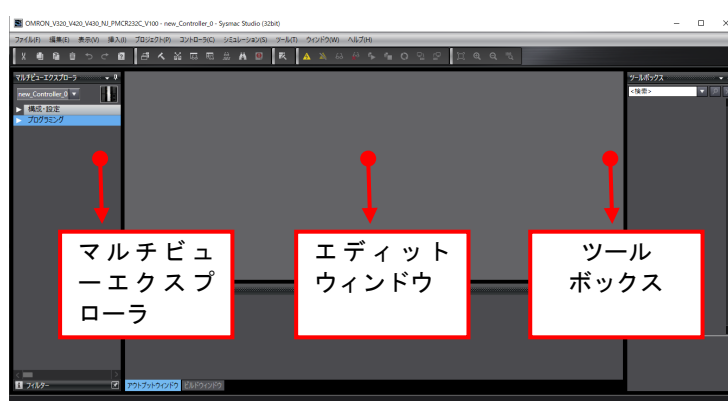
- 2** [ファイルをインポート] ダイアログが表示されますので、[OMRON_V320_V420_V430_NJ_PMCR232C_V100.smc2] を選択し、[開く] をクリックします。

※使用するプロジェクトファイルは、オムロンより入手してください。



- 3** [OMRON_V320_V420_V430_NJ_PMCR232C_V100.smc2] プロジェクト画面が表示されます。画面左側を「マルチビューエクスプローラ」、右側を「ツールボックス」、中央を「エディットウィンドウ」といいます。

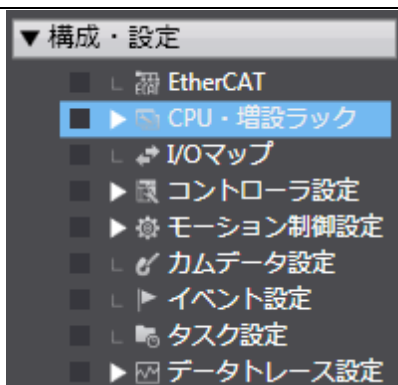
※「プロジェクトのバージョンが異なります」というエラーメッセージが表示された場合、Sysmac Studio のバージョンを「5.2.デバイス構成」以降に変更してください。



7.3.3. パラメータの確認とビルドの実行

設定パラメータを確認し、プロジェクトデータのプログラムチェックおよびビルドを実行します。

- 1 [マルチビューエクスプローラ]から、[構成・設定]－[CPU・増設ラック]をダブルクリックします。



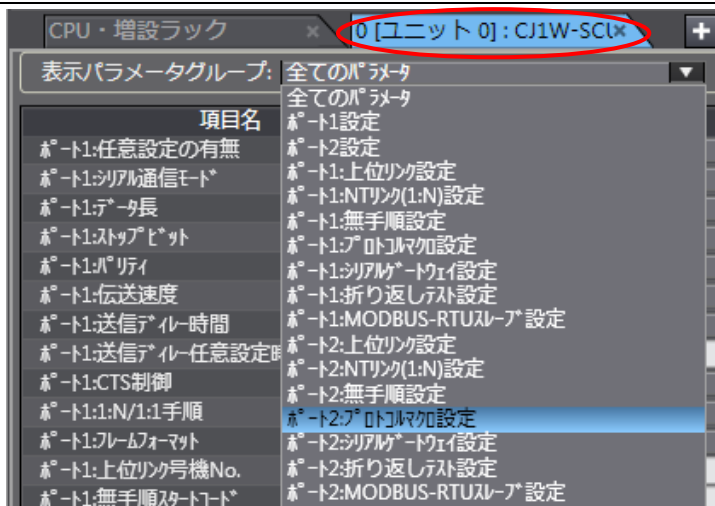
- 2 [エディットウィンドウ]に[CPU・増設ラック]タブが表示されます。
右図のように、画面上のシリアルコミュニケーションユニットを選択します。
[CJ1W-SCU22]の設定値が表示され、[デバイス名]が[J01]、[ユニット番号]が[0]であることを確認します。



※設定内容が異なる場合は、値を変更してください。

[高機能ユニット設定の編集]をクリックします。

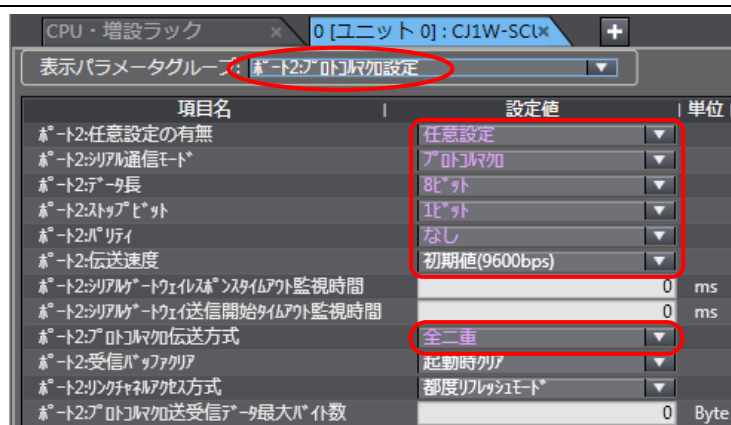
- 3 [0 [ユニット0]:]タブが表示されます。
[表示パラメータグループ]のプルダウンメニューから、[ポート2: プロトコルマクロ設定]を選択します。



- 4 [表示パラメータグループ] が、
[ポート 2: プロトコルマクロ設定]
] となります。

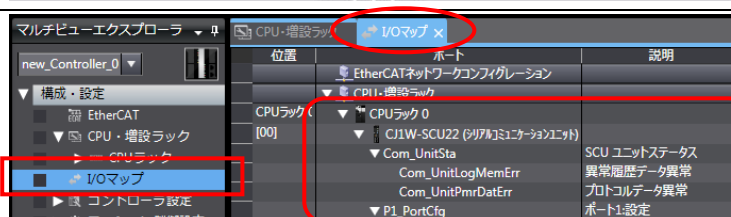
[ポート 2: プロトコルマクロ設定]
] の設定内容が表示されます。
[ポート 2: 任意設定の有無] が、
[任意設定] で、その他の項目
が、6.1.項の設定内容と同一であ
ることを確認します。

※設定内容が異なる場合は、プ
ルダウンメニューから値を変
更してください。値を変更し
た場合は、[適用] をクリック
してください。



- 5 [マルチビューエクスプロー
ラ] から、[構成・設定] - [I/O
マップ] をダブルクリックしま
す。

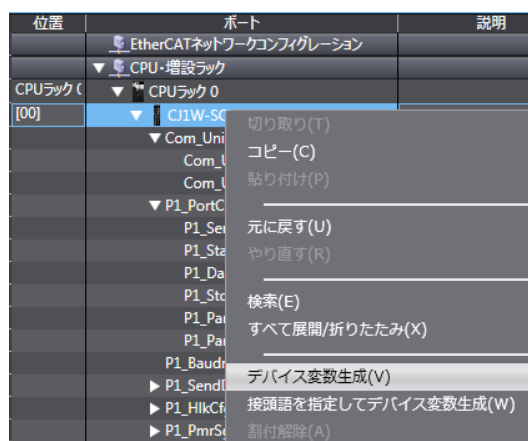
[I/O マップ] タブが表示され、
ユニットのパラメータが表示さ
れます。



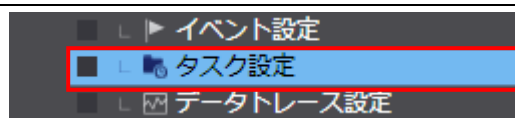
- 6 [I/O マップ] 画面の [変数] エ
リア内のデータの先頭が [J01]
で、[変数種別] エリアが [グロ
ーバル変数] であることを確認
します。

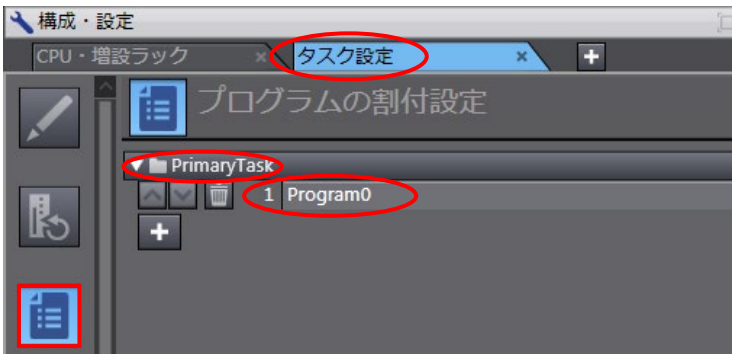
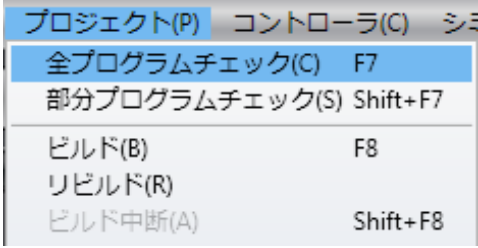
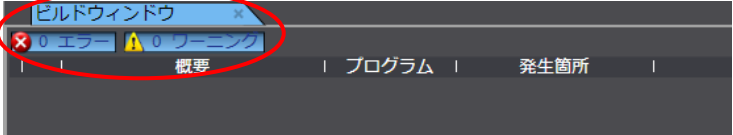
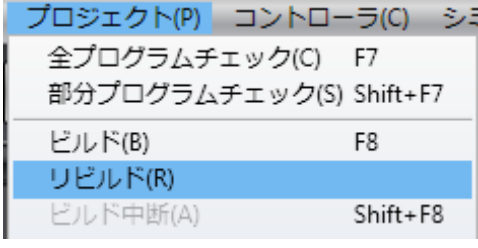
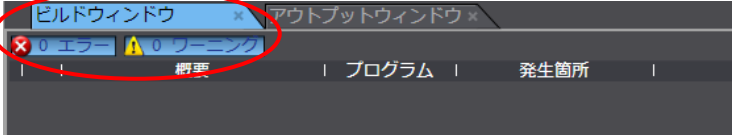
※設定内容が異なる場合は、右
の図のように [CJ1W-SCU22]
を選択した状態で、マウスの
右ボタンメニューから [デバ
イス変数生成] を選択してく
ださい。

ポート	説明	R/W	データ型	変数	変数コメント	変数種別
EtherCATネットワークコンフィグレーション						
CPUラック 0						
CJ1W-SCU22 (シリアル通信ユニット)						
Com_UnitSta	SCU ユニットステータス	R	WORD	J01_Com_UnitSta		グローバル変数
Com_UnitLogMemErr	異常履歴データ異常	R	BOOL	J01_Com_UnitLog		グローバル変数
Com_UnitPmrDatErr	プロトコルデータ異常	R	BOOL	J01_Com_UnitPmr		グローバル変数
P1_PortCfg	ポート1:設定	RW	WORD	J01_P1_PortCfg		グローバル変数
P1_SerSetCfg	ポート1:任意設定	RW	BOOL	J01_P1_SerSetCfg		グローバル変数



- 7 [マルチビューエクスプロー
ラ] から、[構成・設定] - [タ
スク設定] をダブルクリックし
ます。



- 8** [エディットウィンドウ] に、[タスク設定] タブが表示されますので、[プログラムの割付設定] を選択し、[PrimaryTask] に [Program0] が設定されていることを確認します。
- 
- 9** メニューバーから、[プロジェクト] - [全プログラムチェック] を選択します。
- 
- 10** [エディットウィンドウ] 下に、[ビルドウィンドウ] が表示されますので、エラーおよびワーニングが、ともに「0」であることを確認します。
- 
- 11** メニューバーから、[プロジェクト] - [リビルド] を選択します。
- 
- 変換中の画面が表示されます。
- 
- 12** [ビルドウィンドウ] 内のエラーおよびワーニングが、ともに「0」であることを確認します。
- 

7.3.4. オンライン接続とプロジェクトデータの転送

Sysmac Studio をオンライン接続し、プロジェクトデータをコントローラに転送します。

警告

Sysmac Studio からユーザプログラム、「構成／設定」のデータ、デバイス変数、CJ ユニット用メモリの値を転送するときは、転送先ノードの安全を確認してから行ってください。

CPU ユニットの動作モードにかかわらず、装置や機械が想定外の動作をする恐れがあります。

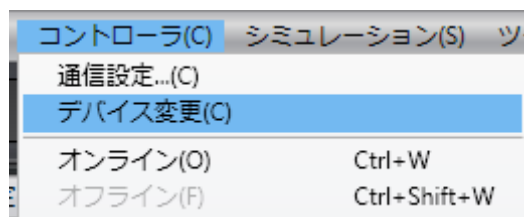


注意

コントローラやユニットをリセットするときには安全を確認してから行ってください。



- 1** メニューバーから、[コントローラ] - [デバイス変更] を選択します。



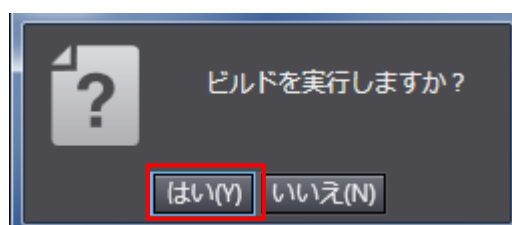
- 2** [デバイス変更] ダイアログが表示されます。
[デバイス] および [バージョン] が、右図のコントローラを使用する設定になっていることを確認し、[OK] をクリックします。

※設定内容が異なる場合は、プルダウンメニューから選択して、設定してください。

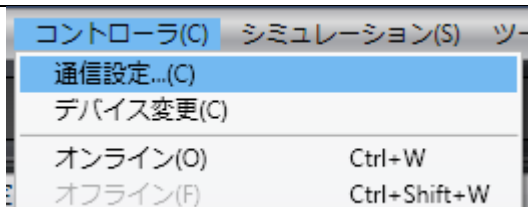


- 3** 2 項で設定を変更した場合、[ビルド] ダイアログが表示されますので、[はい] をクリックします。

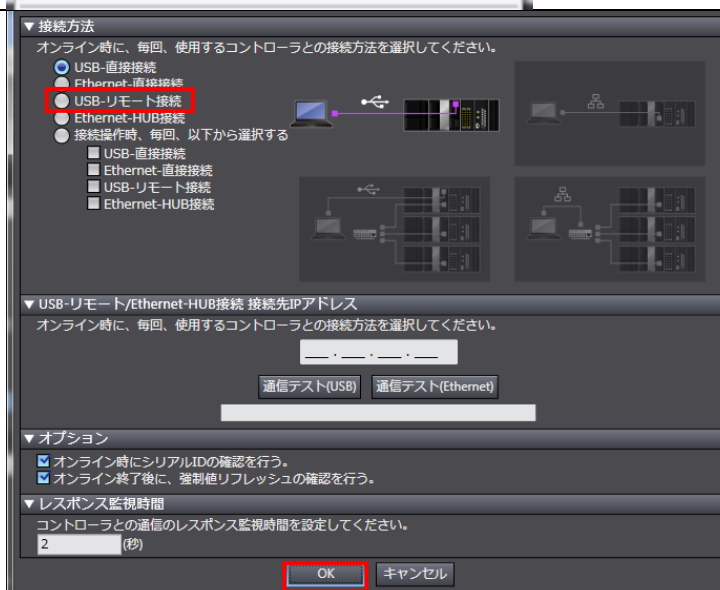
※変更のない場合には、このダイアログは表示されません。



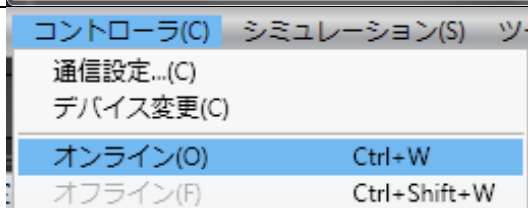
- 4 メニューバーから、[コントローラ] - [通信設定] を選択します。



- 5 [通信設定] ダイアログが表示されます。
[接続方法] から、[USB-直接接続] を選択します。

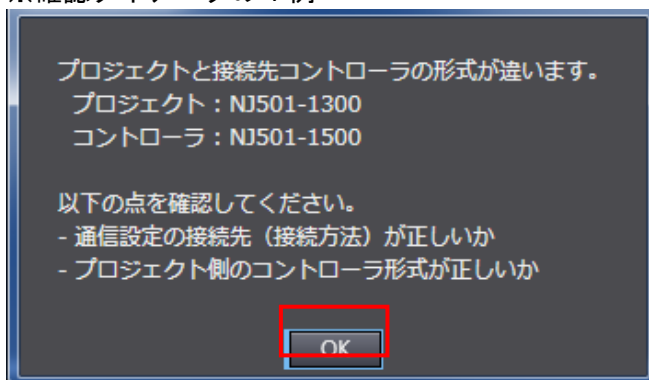


- 6 メニューバーから、[コントローラ] - [オンライン] を選択します。

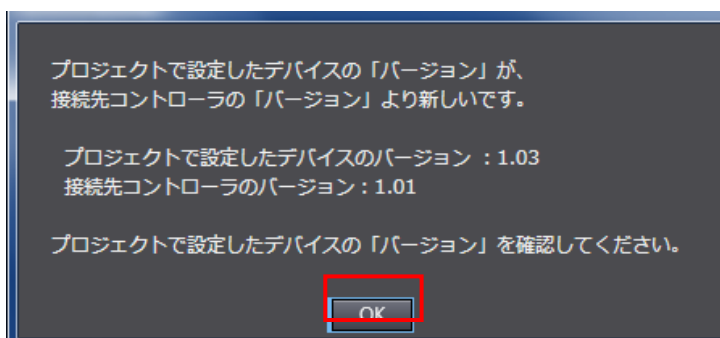


※右のような確認のダイアログが表示された場合、コントローラの形式あるいはバージョンがプロジェクトファイルのデバイス設定と異なっています。コントローラの形式とバージョンおよびプロジェクトファイルのデバイス設定を見直し、手順 1 に戻って、本項の手順を再実行してください。
ダイアログは、[OK] をクリックして、終了します。

※確認ダイアログの 1 例



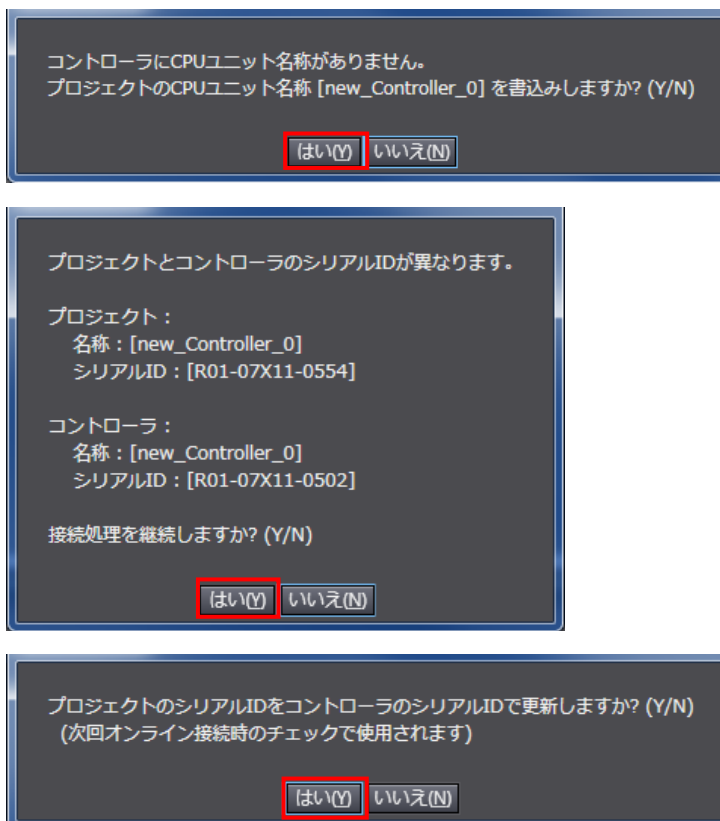
※確認ダイアログに表示される形式やバージョンは、ご使用のコントローラやプロジェクトファイルのデバイス設定により異なります。



- 7** 右図のように、確認のダイアログが表示されますので、[はい]をクリックします。

※使用するコントローラの状態により、表示されるダイアログが異なりますが、[はい]や[Yes]など処理を進める選択を行ってください。

※表示されるシリアルIDは機器により異なります。



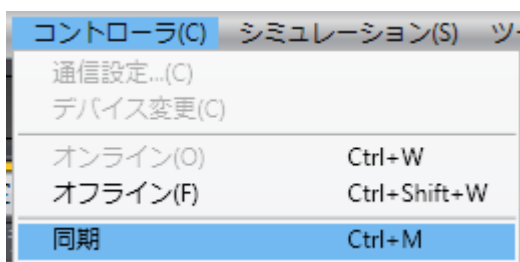
参考

コントローラとのオンライン接続に関する詳細については、「Sysmac Studio Version 1 オペレーションマニュアル」(SBCA-362)の「第5章 コントローラとの接続」を参照してください。

- 8** オンライン状態になると、[エディットウィンドウ]の上段に、黄色い枠が表示されます。

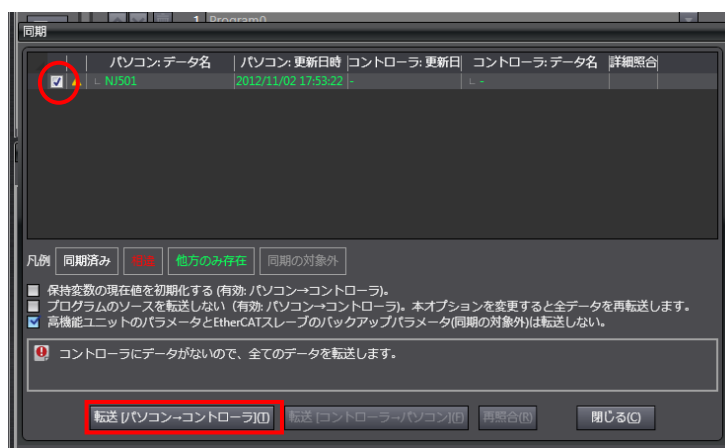


- 9** メニューバーから、[コントローラ] - [同期]を選択します。

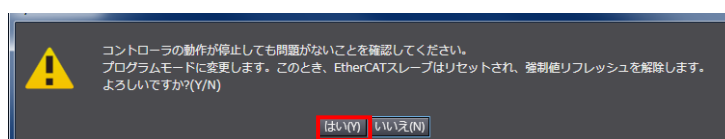


- 10** [同期] ダイアログが表示されます。
転送したいデータ（右図では、[NJ501]）にチェックが付いていることを確認して、[転送 [パソコン→コントローラ]] をクリックします。

※ [転送 [パソコン→コントローラ]] を実行すると、Sysmac Studio のプロジェクトデータがコントローラに転送され、データの照合が行われます。



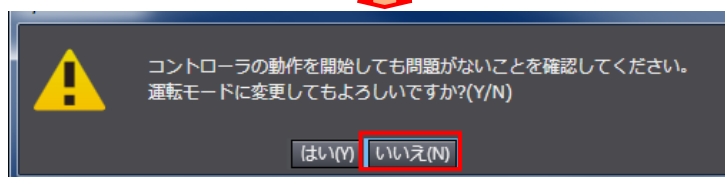
- 11** 確認ダイアログが表示されますので、[はい] をクリックします。



同期中の画面が表示されます。



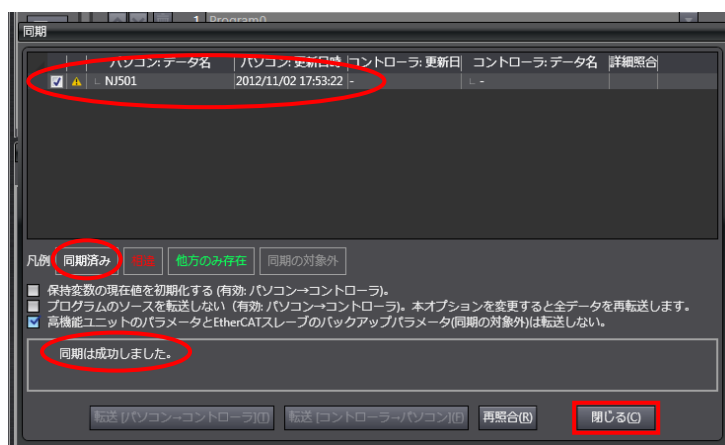
確認ダイアログが表示されますので、[いいえ] をクリックします。



※この後、「プログラムモード」での操作となりますので、ここでは[いいえ]を選択してください。

- 12** 同期したデータの文字色が[同期済み]色になり、「同期は成功しました。」と表示されていることを確認します。
問題がなければ、[閉じる] をクリックします。

※同期が失敗した場合は、配線を確認のうえ、本項の手順を再実行してください。

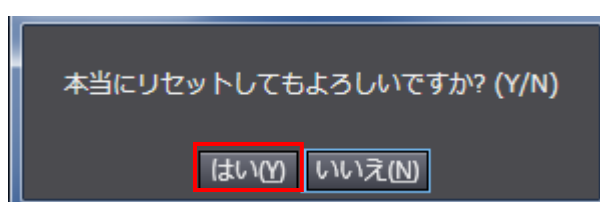
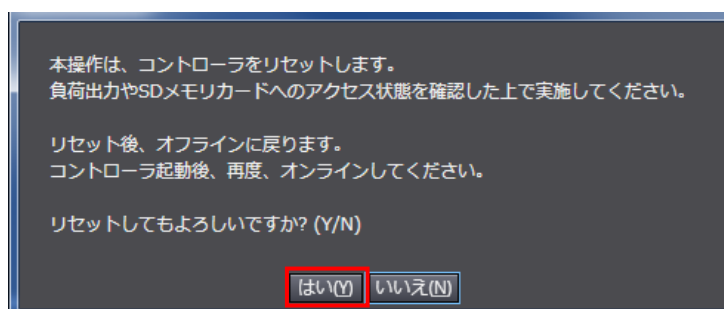


13 メニューバーから、[コントローラ] - [コントローラリセット] を選択します。

※ [コントローラリセット] が選択できない場合は、[動作モード] が [運転モード] になっています。メニューバーから、[コントローラ] - [動作モード] - [プログラムモード] を選択して、[プログラムモード] に変更してから、本項の手順を実行してください。



14 確認用のダイアログが何度か表示されますので、[はい] をクリックしてください。



15 コントローラがリセットされ、Sysmac Studio はオフライン状態になり、上段の黄色い枠が消えます。手順 6~8 を実行し、再度、オンライン状態にします。

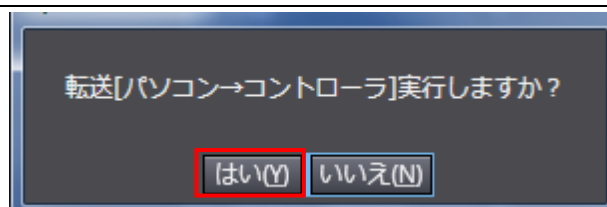


7.3.5. ユニット設定の転送

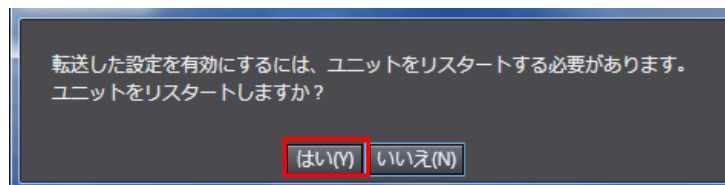
シリアルコミュニケーションユニットの設定データを転送します。

1	メニューバーから [コントローラ] - [動作モード] - [プログラムモード] を選択します。	
2	確認用のダイアログが表示されますので、[はい] をクリックします。	
3	[コントローラステータス] に、[プログラムモード] と表示されます。	
4	[マルチビューエクスプローラ] から、[構成・設定] - [CPU・増設ラック] をダブルクリックします。 画面上的シリアルコミュニケーションユニットを選択します。 [高機能ユニット設定] をクリックします。	
5	[0 [ユニット 0]:] タブが表示されます。 [転送 [パソコン→コントローラ]] をクリックします。	

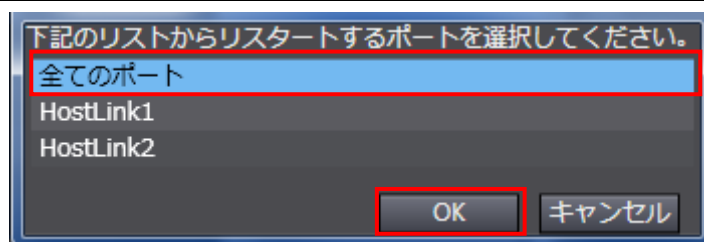
- 6** 確認用のダイアログが表示されます。
[はい] をクリックします。



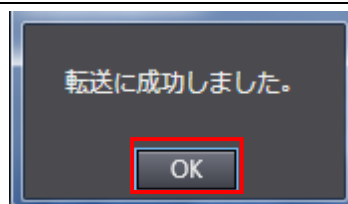
- 転送中のダイアログが表示された後、確認用のダイアログが表示されます。
[はい] をクリックします。



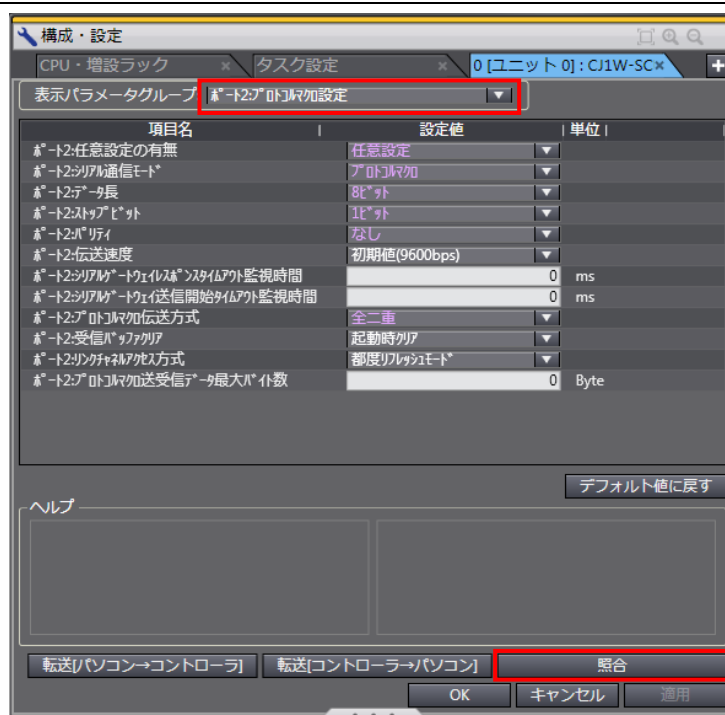
- 7** [ポート選択] ダイアログが表示されます。
[全てのポート] を選択し、
[OK] をクリックします。



- 8** 確認用のダイアログが表示されます。
[OK] をクリックします。



- 9** [表示パラメータグループ] のプルダウンメニューを開き、[ポート 2: プロトコルマクロ設定] を選択します。
[照合] をクリックします。



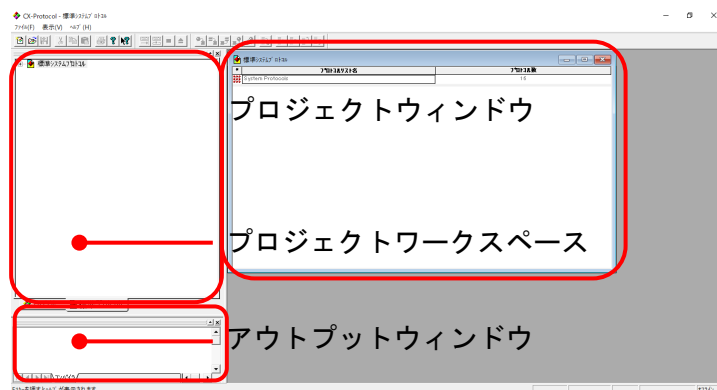
10 右図の赤枠部分に、「≠」（不一致）が発生していないことを確認します。

項目名	設定値	照合結果	単位
※-12:任意設定の有無	任意設定	任意設定	
※-12:シリアル通信モード	ブートローダー	ブートローダー	
※-12:データ長	8ビット	8ビット	
※-12:ストップビット	1ビット	1ビット	
※-12:パリティ	なし	なし	
※-12:伝送速度	初期値(9600bps)	初期値(9600bps)	
※-12:シリアルポートウェイトタイムアウト監視時間	0	0	ms
※-12:シリアルポート送信開始タイムアウト監視時間	0	0	ms
※-12:ブートローダー伝送方式	全二重	全二重	
※-12:受信バッファサイズ	起動時クリア	起動時クリア	
※-12:リフレッシュレート方式	都度リフレッシュ	都度リフレッシュ	
※-12:ブートローダー送信データ最大バイト数	0	0	Byte

7.3.6. CX-Protocol の起動とプロトコルマクロデータの読み込み

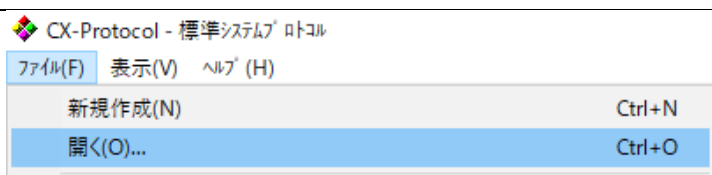
CX-Protocol を起動し、プロトコルマクロデータを読み込みます。

- 1** CX-Protocol を起動します。



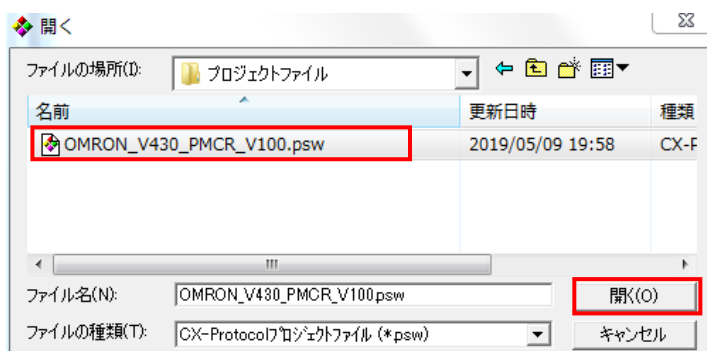
(CX-Protocol)

- 2** メニューバーから [ファイル] - [開く] を選択します。

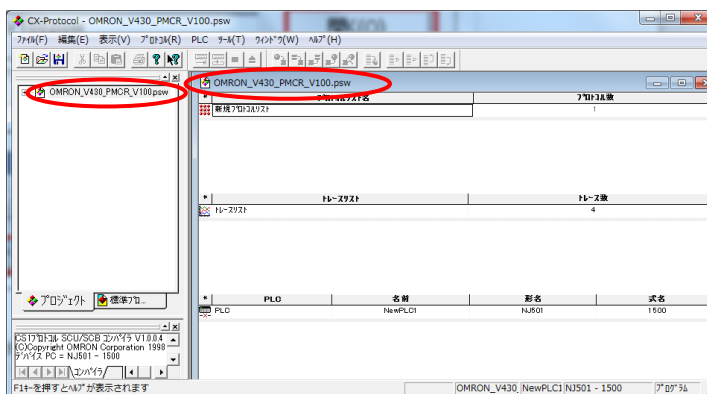


- 3** [開く] ダイアログが表示されますので、
[OMRON_V320_V420_V430_PMCR_V100.psw] を選択し、
[開く] をクリックします。

※使用するプロトコルマクロデータは、オムロンから入手してください。



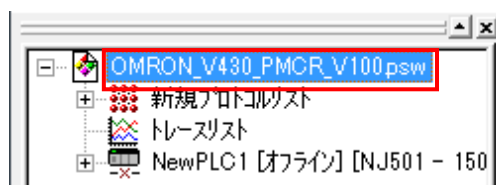
- 4** プロジェクトワークスペース、プロジェクトウィンドウにそれぞれ読み込んだプロトコルマクロデータが表示されます。



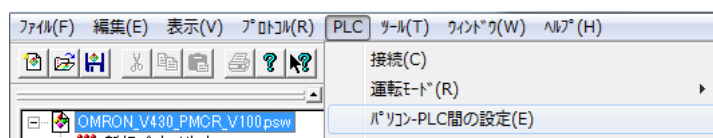
7.3.7. オンライン接続とプロトコルマクロデータの転送

CX-Protocol をオンライン接続し、プロトコルマクロデータをシリアルコミュニケーションユニットに転送します。

- 1** プロジェクトワークスペースの [OMRON_V320_V420_V430_P MCR_V100] をダブルクリックし、ツリーを開きます。

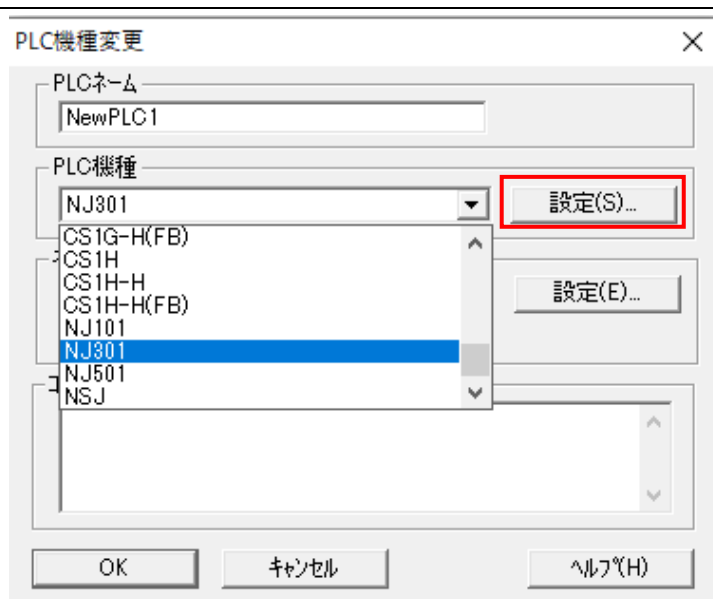


- 2** メニューバーから [PLC] - [パソコン-PLC 間の設定] を選択します。



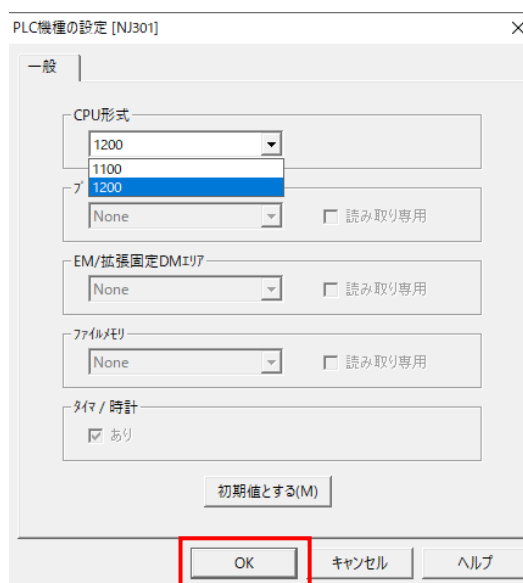
- 3** [PLC 機種変更] ダイアログが表示されますので、[PLC 機種] のプルダウンメニューから使用するコントローラの機種を選択し、[設定] をクリックします。

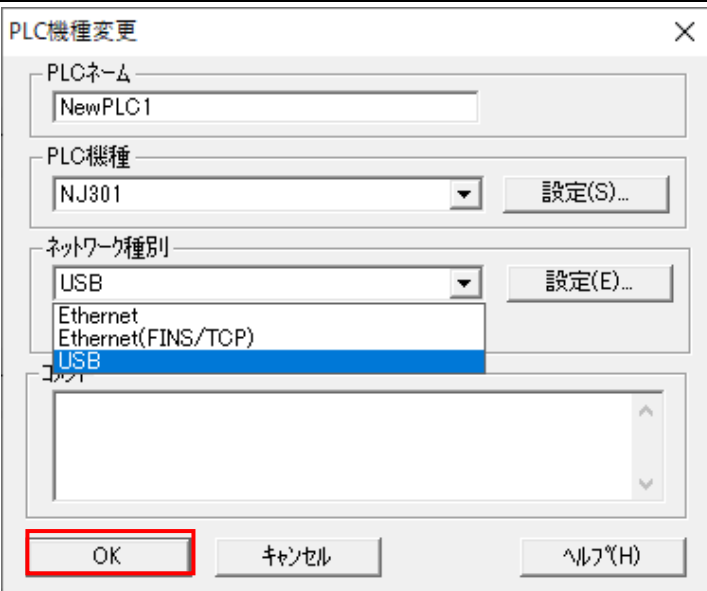
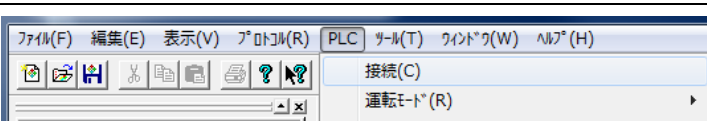
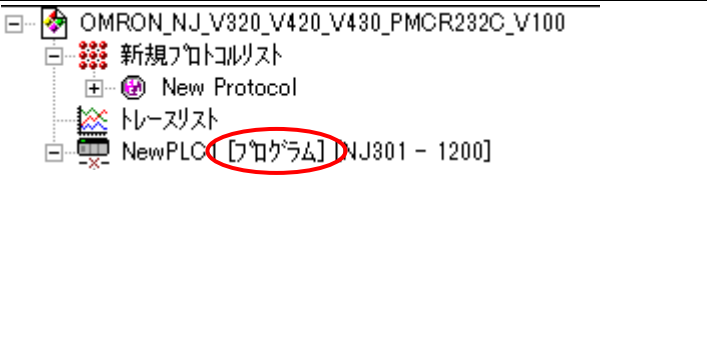
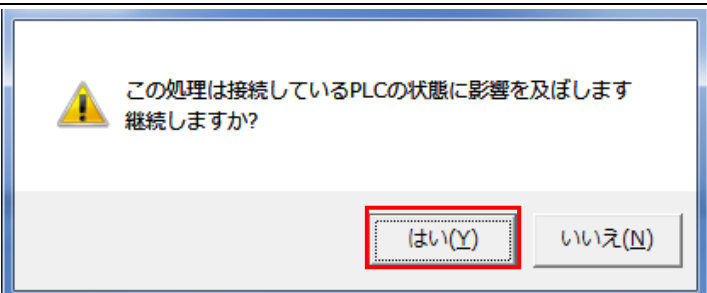
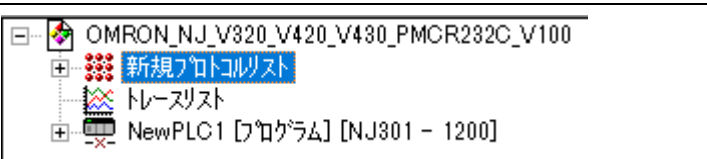
※本資料では [NJ301] を使用します。



- 4** [PLC 機種の設定] ダイアログが表示されますので、[CPU 形式] のプルダウンメニューから使用する CPU の形式を選択し、[OK] をクリックします。

※本資料では [1200] を使用します。



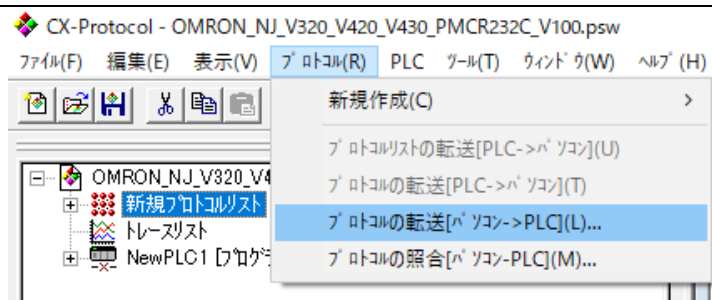
5	[PLC 機種変更] ダイアログの [ネットワーク種別] が [USB] になっていることを確認し、[OK] をクリックします。 ※ [ネットワーク種別] が [USB] になっていない場合には、プルダウンメニューから [USB] を選択してください。	
6	メニューバーから [PLC] - [接続] を選択します。	
7	プロジェクトワークスペースの PLC アイコンの表示が [オフライン] から [プログラム] に変わり、コントローラとオンライン接続できたことを確認します。 ※ [運転] と表示された場合は、以下の手順8～9を実行し、[プログラム] に変更します。	
8	コントローラの運転モードが [運転] である場合は、メニューバーから [PLC] - [運転モード] - [プログラム] を選択します。	
9	右図のダイアログが表示されますので、[はい] をクリックします。 手順7の表示のとおり、[プログラム] モードに変わったことを確認します。	
10	プロジェクトワークスペースの「新規プロトコルリスト」をダブルクリックし、ツリーを開きます。	

- 11** 右図のプロジェクトウィンドウが表示されますので、[ターゲット] に [SCU[0]] が入力されていることを確認します。

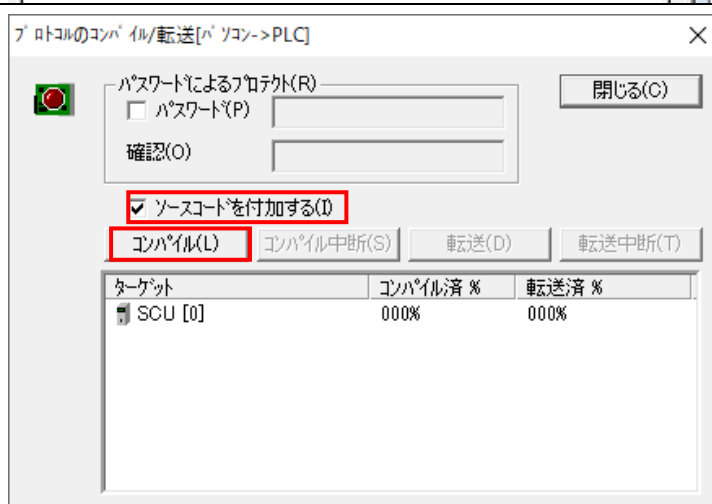
※ [SCU[0]] が入力されていない場合には、右図のように [SCU[0]] を選択してください。



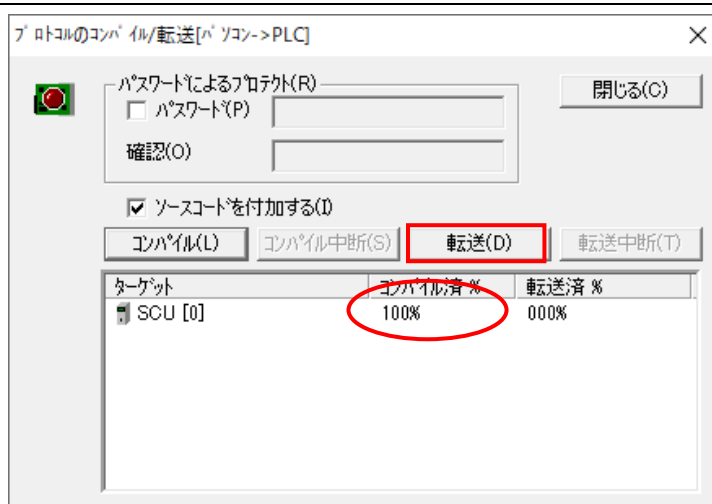
- 12** 「新規プロトコルリスト」を選択した状態で、メニューバーから [プロトコル] - [プロトコルの転送 [パソコン→PLC]] を選択します。



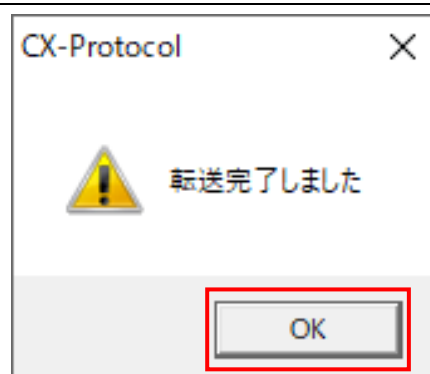
- 13** 右図のダイアログが表示されますので [ソースコードを付加する] にチェックを入れ、[コンパイル] をクリックします。



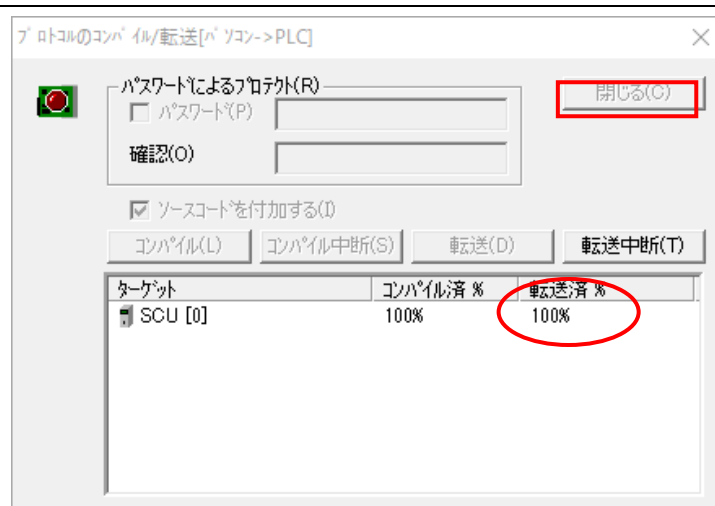
- 14** 右図のダイアログで [コンパイル済%] が [100%] となれば、コンパイルは完了です。コンパイルが完了したことを確認してから [転送] をクリックします。



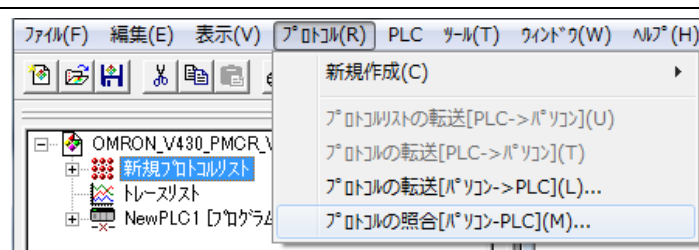
- 15** 右図のダイアログが表示されますので、[OK] をクリックします。



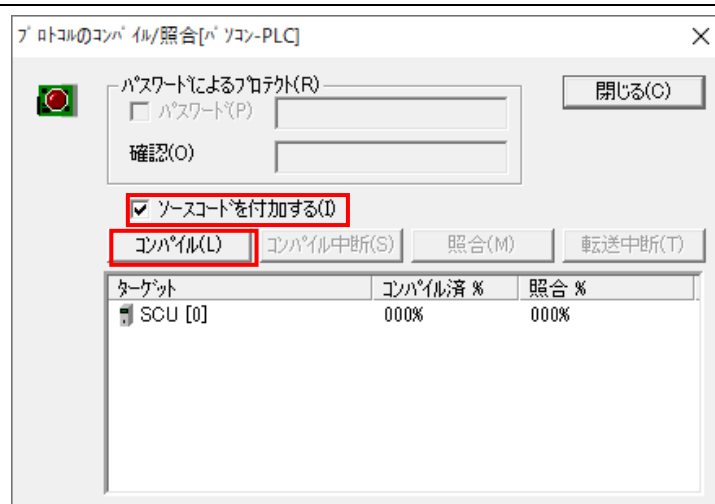
- 16** 右図のダイアログで [転送済%] が [100%] となったことを確認してから、[閉じる] をクリックします。



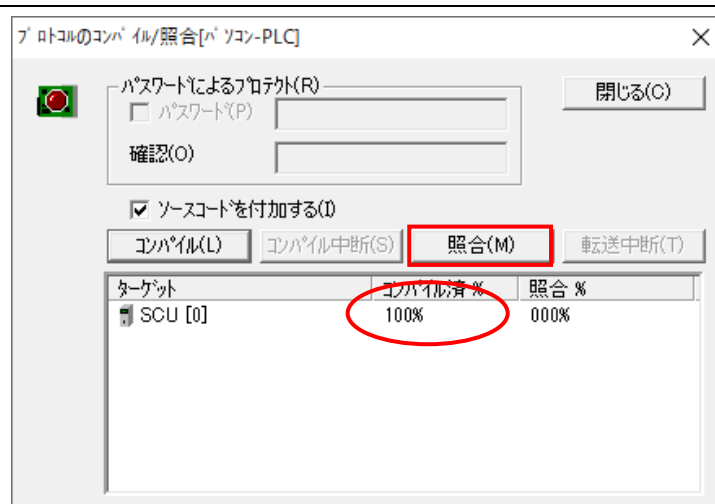
- 17** 「新規プロトコルリスト」を選択した状態で、メニューバーから [プロトコル] - [プロトコルの照合 [パソコン-PLC]] を選択します。



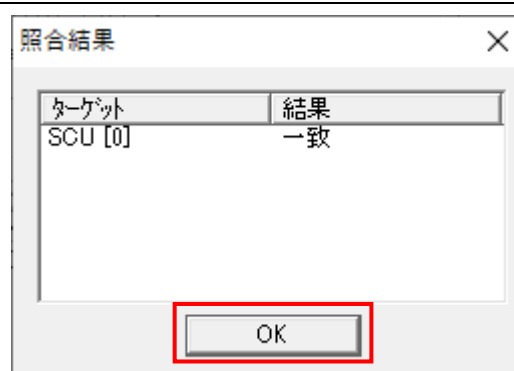
- 18** 右図のダイアログが表示されますので [ソースコードを付加する] にチェックを入れ、[コンパイル] をクリックします。



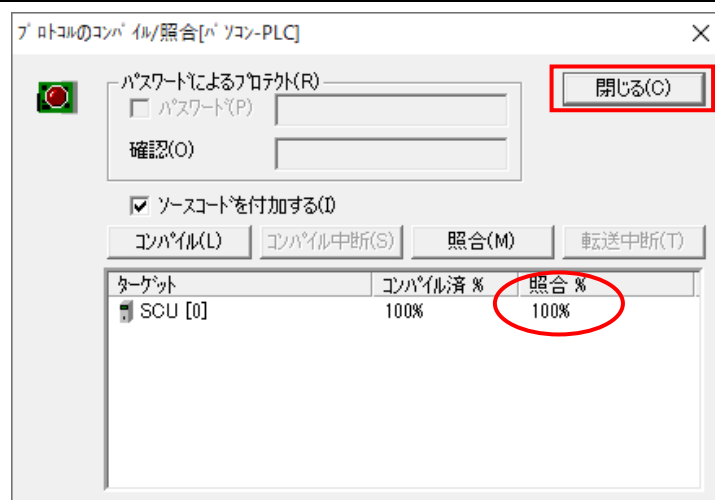
- 19** 右図のダイアログで「コンパイル済%」が「100%」となれば、コンパイルは完了です。コンパイルが完了したことを確認してから「照合」をクリックします。



- 20** 右図のダイアログが表示されますので「OK」をクリックします。



- 21** 右図のダイアログで「照合%」が「100%」となったことを確認してから、「閉じる」をクリックします。



7.4. シリアル通信の確認

プログラムを実行し、シリアル通信が正しく行われていることを確認します。

⚠ 注意

オンラインでウォッチウィンドウから変数の値を変更するときは、十分に安全を確認してから行ってください。誤操作を行うと、コントローラの動作モードにかかわらず出力ユニットに接続された機器が動作する恐れがあります。



使用上の注意

本資料では、コードリーダを工場出荷時の初期設定状態で使用します。コードリーダが工場出荷時の初期設定状態でない場合は、「8.2. コードリーダの初期化」を参照し、必ず初期化を行ってください。



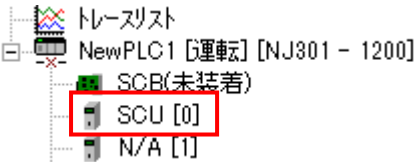
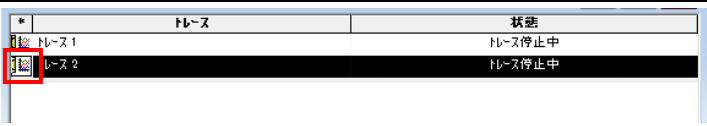
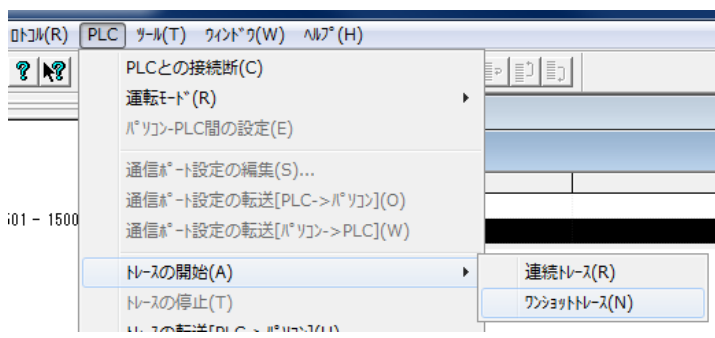
使用上の注意

以降の手順を実施する前に、シリアルケーブルが接続されていることを確認ください。接続されていない場合、各機器の電源を OFF にしてからシリアルケーブルを接続してください。

7.4.1. トレースの開始

CX-Protocol でトレースを開始します。

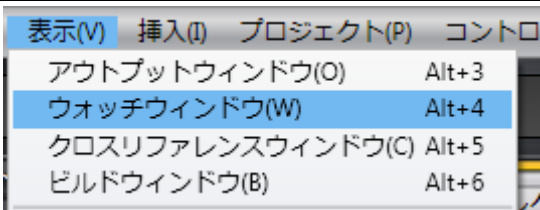
1 CX-Protocol のメニューバーから [PLC] – [運転モード] – [運転] を選択します。	
2 右図のダイアログが表示されますので、[はい] をクリックします。	
3 運転モードが [運転] モードに変わったことを確認し、 [NewPLC1] をダブルクリックします。	

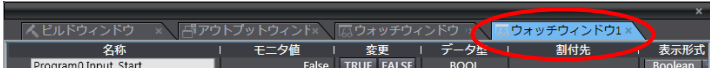
- 4** [-NewPLC1] の下のツリーが開きますので、シリアルコミュニケーションユニット（右図では [SCU[0]]）を選択します。
- 
- 5** プロジェクトウィンドウの[トレース 2] アイコン () を選択します。
右図のとおり、[トレース 2] が反転表示されることを確認します。
- 
- ※ [トレース 2] がシリアルコミュニケーションユニットの「ポート 2」に対応します。
- 6** メニューバーから [PLC] – [トレースの開始] – [ワンショットトレース] を選択します。
- 
- 7** プロジェクトウィンドウの[トレース 2] の状態が [ワンショットトレース実行中] になったことを確認します。
- 

7.4.2. プログラムの実行

Sysmac Studio でプログラムを実行します。

- 1** Sysmac Studio のメニューバーから、[表示] - [ウォッチウィンドウ] を選択します。

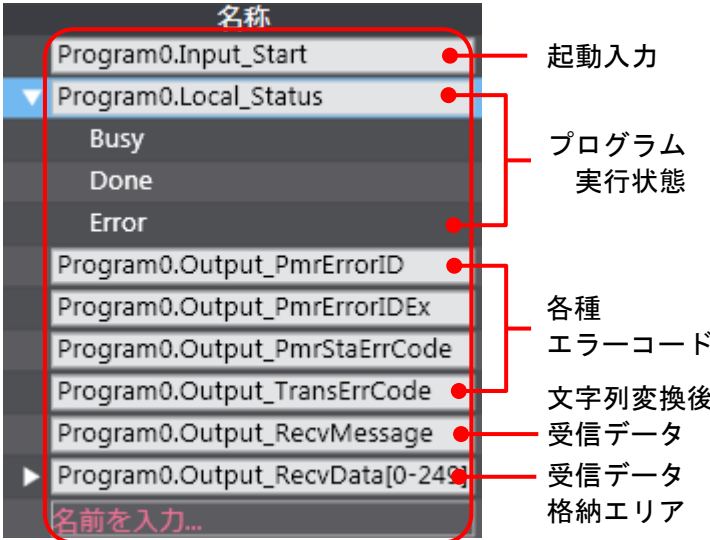

- 2** [エディットウィンドウ]の下段に、[ウォッチウィンドウ 1] タブが表示されます。


- 3** [名称]エリアに右図に示す変数が表示されていることを確認します。

※変数を追加する場合は、[名前を入力...]をクリックしてください。

※変数の左にある「▼」が「▶」表示の場合には、「▶」をクリックし、構造体変数の構成を展開してください。

※以降の説明では、[名称]の「Program0」は省略して説明します。

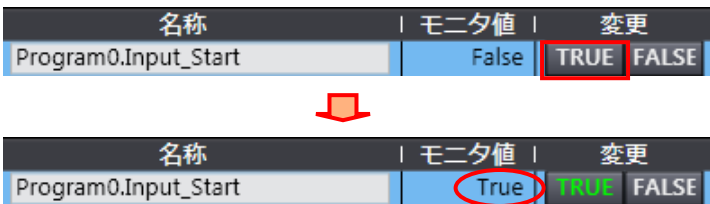

- 4** 読み取りの例として、右図のバーコードを使用します。コードリーダを右図のバーコードが読み取れる位置にセットします。


- 5** [Input_Start] の [変更] エリアの [TRUE] をクリックします。

[Input_Start] の [モニタ値] が [True] になります。

プログラムが動作し、コードリーダとシリアル通信が行われ、バーコードが読み取られます。

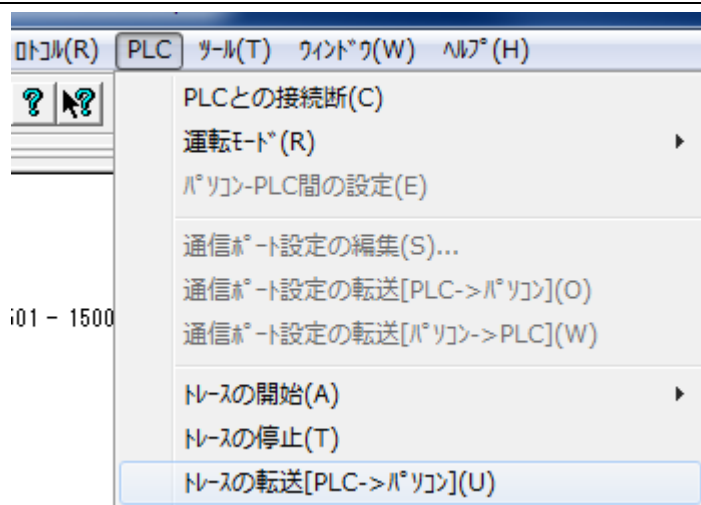
※コードリーダ本体 LED の PASS が緑点灯すれば読取は成功です。



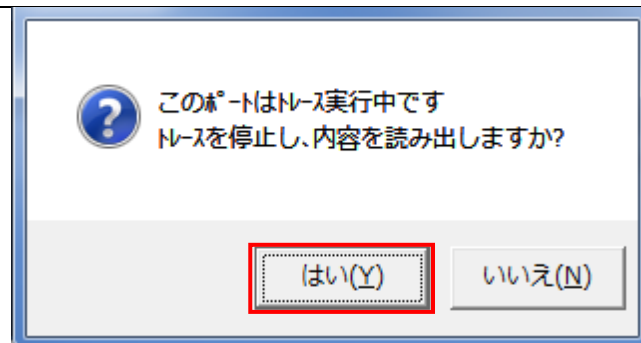
7.4.3. トレースデータの確認

CX-Protocol のトレースデータで正しいデータが送受信されていることを確認します。

- 1** CX-Protocol のメニューバーから、[PLC] - [トレースの転送 [PLC→パソコン]] を選択します。

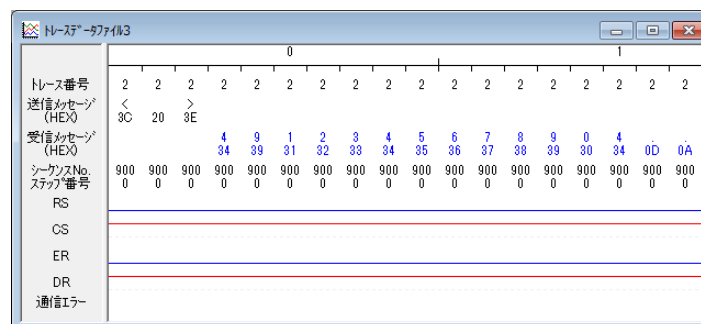


- 2** 右図のダイアログが表示されますので [はい] をクリックします。



- 3** 右図のトレースデータファイルで、送受信メッセージを確認します。

※右図の例では、コントローラから「読取りトリガ」コマンドを送信し、7.4.2.項の手順4のバーコードの内容を受信していることがわかります。



■送信データ

16 進数 : 3C 20 3E

文字列 : <>

■受信データ

16 進数 : 34 39 31 32 33 34 35 36 37 38 39 30 34 0D 0A

文字列 : "4912345678904" [CR] [LF]

7.4.4. 受信データの確認

Sysmac Studio でコントローラの変数に正しいデータが書き込まれていることを確認します。

1	プログラムの実行状態を示す [Local_Status.Done] の [モニタ値] が [True] になったことを確認します。 ※プログラムが正常終了したことがわかります。 ※異常終了した場合は、[Local_Status.Error] が [True] になりますので、手順 2 を参照し、対処してください。	【正常終了した場合】 <table border="1"> <thead> <tr> <th>名称</th><th>モニタ値</th><th>変更</th></tr> </thead> <tbody> <tr> <td>Program0.Input_Start</td><td>True</td><td>TRUE FALSE</td></tr> <tr> <td>▼ Program0.Local_Status</td><td></td><td></td></tr> <tr> <td> Busy</td><td>False</td><td>TRUE FALSE</td></tr> <tr> <td> Done</td><td>True</td><td>TRUE FALSE</td></tr> <tr> <td> Error</td><td>False</td><td>TRUE FALSE</td></tr> </tbody> </table> 【異常終了した場合】 <table border="1"> <thead> <tr> <th>名称</th><th>モニタ値</th><th>変更</th></tr> </thead> <tbody> <tr> <td>Program0.Input_Start</td><td>True</td><td>TRUE FALSE</td></tr> <tr> <td>▼ Program0.Local_Status</td><td></td><td></td></tr> <tr> <td> Busy</td><td>False</td><td>TRUE FALSE</td></tr> <tr> <td> Done</td><td>False</td><td>TRUE FALSE</td></tr> <tr> <td> Error</td><td>True</td><td>TRUE FALSE</td></tr> </tbody> </table>	名称	モニタ値	変更	Program0.Input_Start	True	TRUE FALSE	▼ Program0.Local_Status			Busy	False	TRUE FALSE	Done	True	TRUE FALSE	Error	False	TRUE FALSE	名称	モニタ値	変更	Program0.Input_Start	True	TRUE FALSE	▼ Program0.Local_Status			Busy	False	TRUE FALSE	Done	False	TRUE FALSE	Error	True	TRUE FALSE
名称	モニタ値	変更																																				
Program0.Input_Start	True	TRUE FALSE																																				
▼ Program0.Local_Status																																						
Busy	False	TRUE FALSE																																				
Done	True	TRUE FALSE																																				
Error	False	TRUE FALSE																																				
名称	モニタ値	変更																																				
Program0.Input_Start	True	TRUE FALSE																																				
▼ Program0.Local_Status																																						
Busy	False	TRUE FALSE																																				
Done	False	TRUE FALSE																																				
Error	True	TRUE FALSE																																				
2	プログラムが正常終了した場合は、各エラーコードが「0」であることを確認します。 ※プログラムが異常終了した場合は、発生した異常に応じて、エラーコードが格納されますので、「9.8. 異常処理」を参照し、対処してください。	【正常終了した場合】 <table border="1"> <tbody> <tr> <td>Program0.Output_PmrErrorID</td><td>0000</td><td></td></tr> <tr> <td>Program0.Output_PmrErrorIDEx</td><td>0000 0000</td><td></td></tr> <tr> <td>Program0.Output_PmrStaErrCode</td><td>0000</td><td></td></tr> <tr> <td>Program0.Output_TransErrCode</td><td>0000</td><td></td></tr> </tbody> </table> 【異常終了した場合】 <table border="1"> <tbody> <tr> <td>Program0.Output_PmrErrorID</td><td>0000</td><td></td></tr> <tr> <td>Program0.Output_PmrErrorIDEx</td><td>0000 0000</td><td></td></tr> <tr> <td>Program0.Output_PmrStaErrCode</td><td>0000</td><td></td></tr> <tr> <td>Program0.Output_TransErrCode</td><td>0010</td><td></td></tr> </tbody> </table>	Program0.Output_PmrErrorID	0000		Program0.Output_PmrErrorIDEx	0000 0000		Program0.Output_PmrStaErrCode	0000		Program0.Output_TransErrCode	0000		Program0.Output_PmrErrorID	0000		Program0.Output_PmrErrorIDEx	0000 0000		Program0.Output_PmrStaErrCode	0000		Program0.Output_TransErrCode	0010													
Program0.Output_PmrErrorID	0000																																					
Program0.Output_PmrErrorIDEx	0000 0000																																					
Program0.Output_PmrStaErrCode	0000																																					
Program0.Output_TransErrCode	0000																																					
Program0.Output_PmrErrorID	0000																																					
Program0.Output_PmrErrorIDEx	0000 0000																																					
Program0.Output_PmrStaErrCode	0000																																					
Program0.Output_TransErrCode	0010																																					
3	Sysmac Studio のウオッチウィンドウにて、受信した内容（読み取ったコード）を確認します。 ※右図の例では、Output_RecvMessage に格納されたデータは、「4912345678904」となり、7.4.3.項の手順 3 のトレースデータと同じ内容であることがわかります。 ※Output_RecvData [0] には、受信データ格納のために使用しているチャンネル数（16 進数で「0008」）が格納されます。読み取ったバーコードの内容は、Output_RecvData[1]~[7] に格納されますので、Output_RecvData[0]を含め、使用チャンネル数は 8 となります。	<table border="1"> <tbody> <tr> <td>Program0.Output_RecvMessage</td><td>4912345678904</td><td></td></tr> <tr> <td>▼ Program0.Output_RecvData[0-249]</td><td></td><td></td></tr> <tr> <td> Output_RecvData[0]</td><td>0008</td><td></td></tr> <tr> <td> Output_RecvData[1]</td><td>3439</td><td></td></tr> <tr> <td> Output_RecvData[2]</td><td>3132</td><td></td></tr> <tr> <td> Output_RecvData[3]</td><td>3334</td><td></td></tr> <tr> <td> Output_RecvData[4]</td><td>3536</td><td></td></tr> <tr> <td> Output_RecvData[5]</td><td>3738</td><td></td></tr> <tr> <td> Output_RecvData[6]</td><td>3930</td><td></td></tr> <tr> <td> Output_RecvData[7]</td><td>3400</td><td></td></tr> <tr> <td> Output_RecvData[8]</td><td>0000</td><td></td></tr> <tr> <td> Output_RecvData[9]</td><td>0000</td><td></td></tr> </tbody> </table> 受信内容（16 進数） - Output_RecvData [0] : "0008" - Output_RecvData[1] : "3439" - Output_RecvData[2] : "3132" - Output_RecvData[3] : "3334" - Output_RecvData[4] : "3536" - Output_RecvData[5] : "3738" - Output_RecvData[6] : "3930" - Output_RecvData[7] : "3400" 受信データ格納のために使用したチャンネル数 [0]~[7] の 8 チャンネル バーコードの値（文字列） Output_RecvMessage 「4912345678904」	Program0.Output_RecvMessage	4912345678904		▼ Program0.Output_RecvData[0-249]			Output_RecvData[0]	0008		Output_RecvData[1]	3439		Output_RecvData[2]	3132		Output_RecvData[3]	3334		Output_RecvData[4]	3536		Output_RecvData[5]	3738		Output_RecvData[6]	3930		Output_RecvData[7]	3400		Output_RecvData[8]	0000		Output_RecvData[9]	0000	
Program0.Output_RecvMessage	4912345678904																																					
▼ Program0.Output_RecvData[0-249]																																						
Output_RecvData[0]	0008																																					
Output_RecvData[1]	3439																																					
Output_RecvData[2]	3132																																					
Output_RecvData[3]	3334																																					
Output_RecvData[4]	3536																																					
Output_RecvData[5]	3738																																					
Output_RecvData[6]	3930																																					
Output_RecvData[7]	3400																																					
Output_RecvData[8]	0000																																					
Output_RecvData[9]	0000																																					

8. 初期化方法

本資料では、工場出荷時の初期設定状態であることを前提としています。

初期設定状態から変更された機器を利用される場合には、各種設定を手順どおりに進めることができない場合があります。

8.1. コントローラの初期化

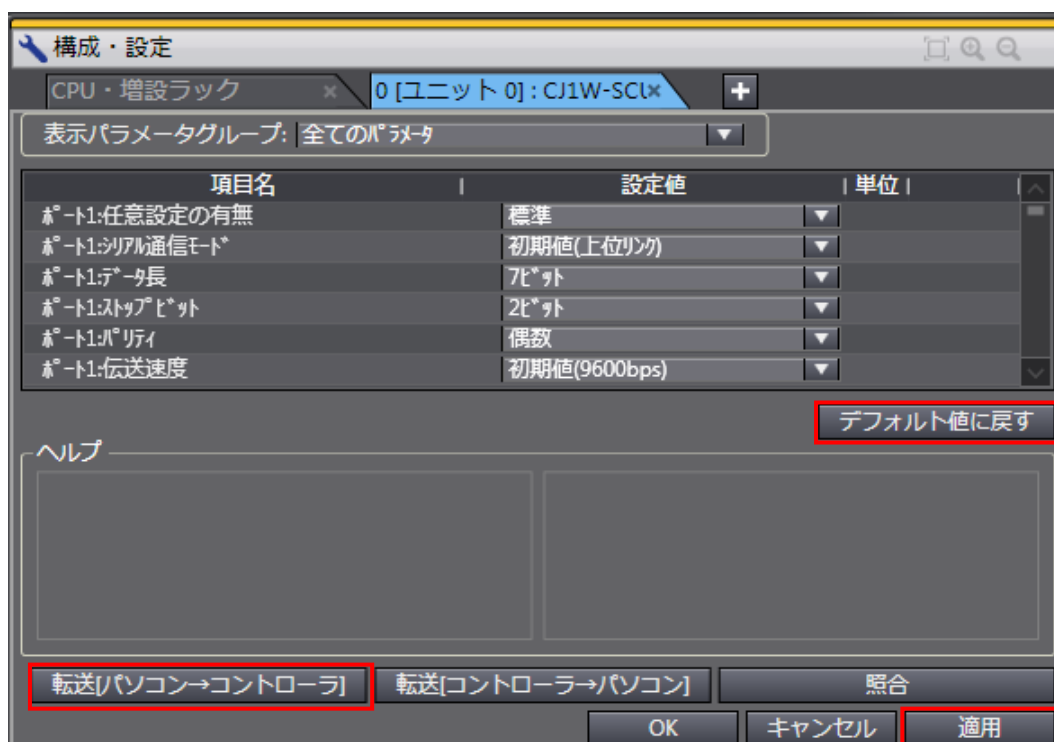
コントローラの初期化を行うためには、シリアルコミュニケーションユニットの初期化とCPUユニットの初期化が必要になります。初期化前にコントローラをプログラムモードにしてください。

8.1.1. シリアルコミュニケーションユニット

シリアルコミュニケーションユニットの設定を初期設定状態に戻すためには、「Sysmac Studio」にて、[CPU・増設ラック] 内の [CJ1W-SCU22] の [高性能ユニット設定の編集] を選択します。

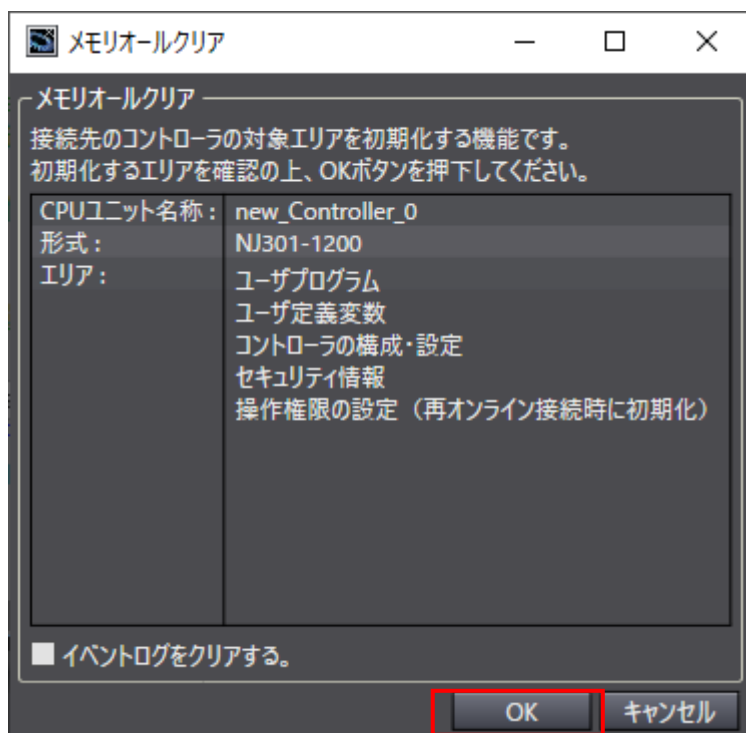


「[デフォルト値に戻す]」を選択し、「[適用]」を選択後、「[転送 (パソコン→コントローラ)]」を実行してください。



CPU ユニット

コントローラの設定を初期設定状態に戻すためには、Sysmac Studio のメニューバーから [コントローラ] - [メモリオールクリア] を選択します。[メモリオールクリア] ダイアログが表示されますので、[OK] をクリックします。



8.2. コードリーダーの初期化

コードリーダーの初期化方法については、『オートフォーカス・マルチコードリーダー MicroHAWK V320-F/V330-F/V420-F/V430-F シリーズ ユーザーズマニュアル (SDNC-745)』の「付録」の「よくあるご質問」-「設定を初期化する方法は？」を参照してください。

9. プログラム

本資料で使用するプログラムおよびプロトコルマクロデータの詳細を示します。

9.1. 概要

本章では、コードリーダー（以下、相手機器）とコントローラ（シリアルコミュニケーションユニット（以下、SCU ユニット））との接続を確認するために使用するプログラムおよびプロトコルマクロデータの仕様および機能について説明します。

本プログラムとプロトコルマクロデータは、SCU ユニットのプロトコルマクロ機能を使用し、相手機器に対して「読取りトリガ」コマンドの送受信を行い、正常／異常終了を判定します。プログラムの正常終了は、プロトコルマクロの送受信シーケンスの正常終了とします。また異常終了は、プロトコルマクロの送受信シーケンスの異常終了とします。

本章では、10 進データと 16 進データの区別が必要な場合には、10 進データの先頭に“10#”（省略可）、16 進データの先頭に“16#”を付けます。（たとえば、10 進データの場合「1000」あるいは「10#1000」、16 進データの場合「16#03E8」）
また、データ型を特定する場合には、先頭に“<データ型>#”を付けます。（たとえば、「WORD#16#03E8」など）

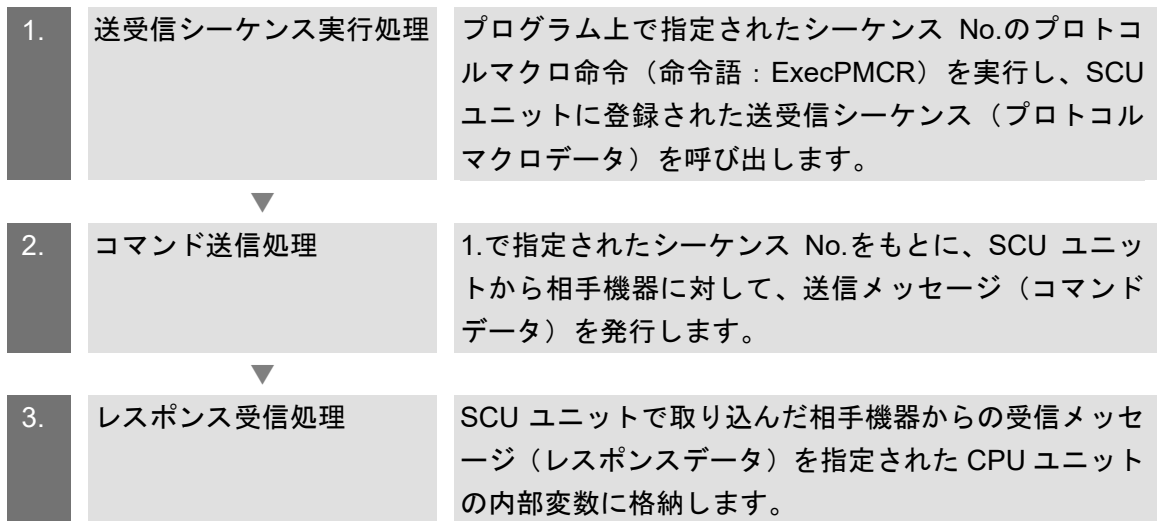


参考

本プログラムおよびプロトコルマクロデータは、当社の実施した試験構成、各商品バージョン、評価に使用した商品ロットにおいて通信が可能であることを確認しております。
電氣的ノイズ等の外乱下や機器自体の性能のばらつきにおいて、動作を保証するものではありません。

9.1.1. 通信データの流れ

コントローラ（SCU ユニット）から相手機器に対してシリアル通信によりコマンドデータを発行し、相手機器からレスポンスデータを受信するまでの流れです。



9.1.2. プロトコルマクロ実行用ファンクションブロックと送受信メッセージ

プロトコルマクロ実行用ファンクションブロック（以下、ExecPMCR 命令）と送受信メッセージの一般的な動きについての概要を説明します。



参考

詳しくは、「NJ/NX シリーズ コマンドリファレンスマニュアル 基本編」(SBCA-468)の「第2章 各命令の説明」－「シリアル通信命令 (ExecPMCR)」を参照してください。

●ExecPMCR 命令

本プログラムでは、以下の標準搭載命令を使用して、シリアル通信を実現しています。

名称	ファンクションブロック	説明
プロトコルマクロ	ExecPMCR	SCU ユニットに登録された送受信シーケンス（プロトコルデータ）の実行を依頼します。

●ExecPMCR 命令引数データ

命令	名称	FB/FUN	グラフィック表現	ST 表現
ExecPMCR	プロトコルマクロ	FB		<pre>ExecPMCR_instance(Execute, Port, SeqNo, SrcDat, DstDat, Done, Busy, Error, ErrorID, ErrorIDEx);</pre>

変数

	名称	入 / 出力	内容	有効範囲	単位	初期値
Port	相手先ポート指定	入力	相手先ポート指定	—	—	—
SeqNo	送受信シーケンス番号		送受信シーケンス番号	0 ~ 999		0
SrcDat[] 配列	送信データ配列		送信データ配列	データ型に従う		(*)
DstDat[] 配列	受信データ配列	入出力	受信データ配列	データ型に従う	—	—

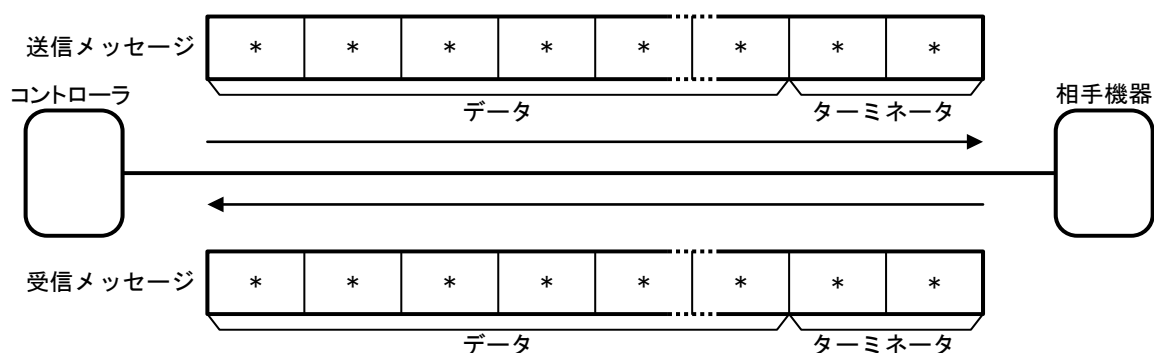
* 入力パラメータが省略された場合、初期値は適用されません。ビルド時に異常になります。

・相手先ポート指定「Port」のデータ型(_sPORT)

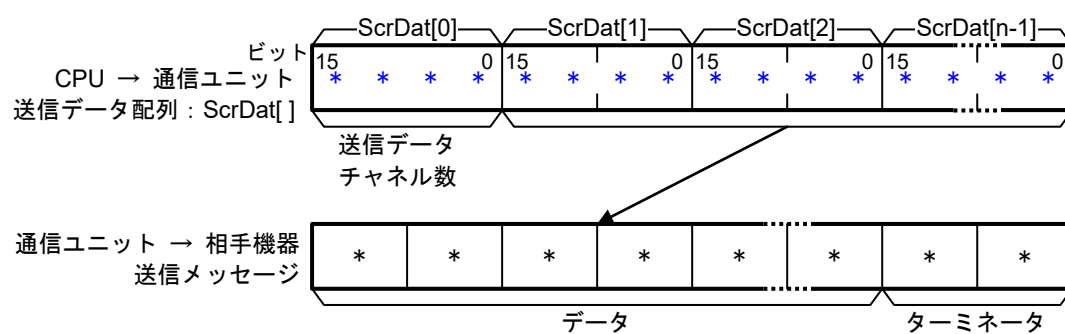
変数	名称	内容	データ型	有効範囲	単位	初期値
Port	相手先ポート指定	相手先ポート指定	_sPORT	—	—	—
UnitNo	ユニット番号	シリアルコミュニケーションユニットのユニット番号	_eUnitNo	_CBU_No00 ~ _CBU_No15	—	_CBU_No00
PhysicPort No	シリアルポート番号	シリアルコミュニケーションユニットの、シリアルポート番号	USINT	1,2	—	1

●送受信メッセージ

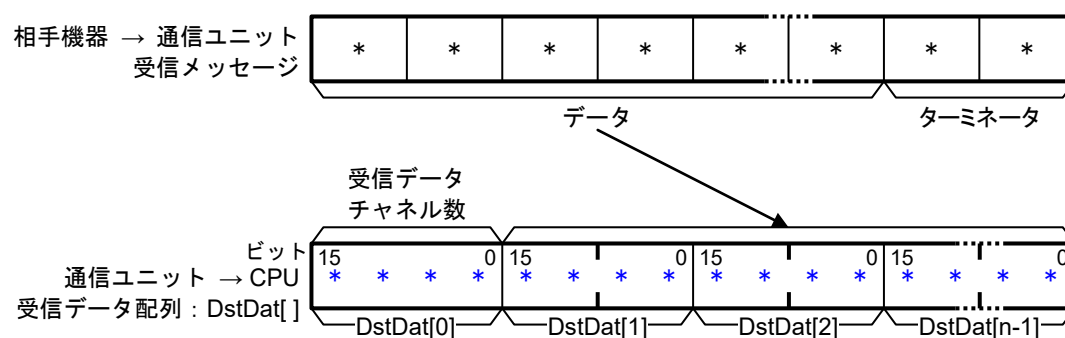
【送受信メッセージ概要】



【送信データ配列 : SrcDat[]と送信メッセージの関係】



【受信メッセージと受信データ配列 : DstDat[]との関係】



9.2. 送受信シーケンス

本プログラムの ExecPMCR 命令に使用できる送受信シーケンス（プロトコルマクロデータ）について説明します。

9.2.1. 送受信シーケンス No.

SCU ユニットに登録された送受信シーケンス（プロトコルマクロデータ）は送受信シーケンス No.によって識別され、ExecPMCR 命令で送受信シーケンス No.を指定することにより、相手機器において対応する相手機器コマンドが実行されます。

本プロトコルマクロデータには以下の送受信シーケンスが準備されています。

No.	コマンド名	内容
900	読取りトリガ	コードの読取りトリガを発行します。

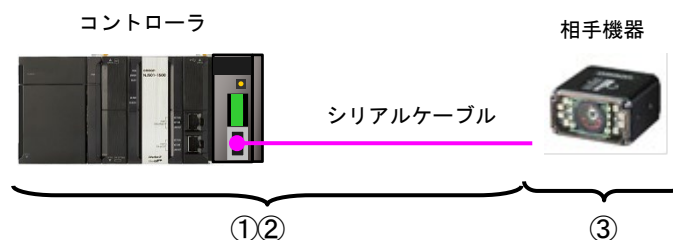
9.2.2. ExecPMCR 命令入力変数の設定

送受信シーケンス No.900「読取りトリガ」の ExecPMCR 命令入力変数の設定は、以下のとおりになっています。

変数	名称（データ型）	データ（説明）
Port	相手先ポート指定(_sPROT)	—
UnitNo	ユニット番号(_eUnitNo)	_eUnitNo#_CBU_No00 (SCU ユニット番号を指定)
PhysicPortNo	シリアルポート番号(USINT)	USINT#10#2 (シリアルポート No.2 を使用)
SeqNo	送受信シーケンス番号(UINT)	UINT#10#900 (読取りトリガを指定)
SrcDat[]配列	送信データ配列(WORD)	WORD#16#0000 (送信データチャネル数 0 : コマンドに必要なパラメータがないことを指定)
DstDat[]配列	受信データ配列(WORD)	(受信データのため、設定不要)

9.3. 異常判断処理

本プログラムでは、以下に示す①～③の 3 つの範囲に分け、異常判断処理を行っています。
エラーコードについては、「9.8. 異常処理」を参照してください。



①ExecPMCR 命令実行時の異常（ExecPMCR 命令異常）

ユニット本体の異常、通信設定の異常など、ExecPMCR 命令実行時の異常を「ExecPMCR 命令異常」として判定します。判定は、ExecPMCR 命令の「エラーコード(ErrorID)」および「拡張エラーコード(ErrorIDEx)」により行います。

②プロトコルマクロ実行時の異常（プロトコルマクロ異常）

プロトコルマクロ実行中に何らかの原因により異常終了した場合を「プロトコルマクロ異常」として判定します。判定はシーケンス Abort 終了フラグ(J01_P2_PmrSeqAbtSta)により行います。

③相手機器の異常（相手機器異常）

相手機器でのコマンド異常、パラメータ異常、データ異常、実行不可などの異常を「相手機器異常」として判定します。判定は、相手機器から返送されてくるレスポンスデータにより行います。本プログラムでは、異常時に相手機器からレスポンスが返送されないため、②のタイムアウト異常として検出されます。

上記①～②の異常発生の原因は関連することもありますので、いずれの異常発生時にも以下の異常コードをすべて格納します。

- ・ ExecPMCR 命令のエラーコード（ErrorID）
- ・ ExecPMCR 命令の拡張エラーコード（ErrorIDEx）
- ・ プロトコルマクロエラーコード（J01_P2_PmrSta の下位 4 ビット）
- ・ 伝送エラー発生状態（J01_P2_TransErrSta）

9.4. 使用変数

本プログラムで使用している変数です。

9.4.1. ユーザ定義変数一覧

本プログラムで使用するデータ型、外部変数（ユーザ定義のグローバル変数／CJユニットデバイス変数／システム定義変数）、内部変数の一覧です。

●データ型（構造体型）

【通信処理ステータスフラグ】

名称	データ型	説明
sStatus	STRUCT	通信処理ステータスフラグ列の構造体
Busy	BOOL	通信処理実行中フラグ TRUE : 実行中 / FALSE : 非実行
Done	BOOL	通信処理正常終了フラグ TRUE : 正常終了 / FALSE : 正常終了以外
Error	BOOL	通信処理異常終了フラグ TRUE : 異常終了 / FALSE : 異常終了以外

●外部変数

【ユーザ定義のグローバル変数】

変数名	データ型	説明
Input_Start	BOOL	通信開始スイッチ 「FALSE」→「TRUE」でプログラムが起動
Input_SendData	ARRAY[0..249] OF WORD	送信データ格納エリア（WORD 型 500 バイト）
Output_RecvData	ARRAY[0..249] OF WORD	受信データ格納エリア（WORD 型 500 バイト） ExecPMCR 命令の DstDat[] を格納するエリア
Output_RecvMessage	STRING[498]	文字列に変換された受信データ格納エリア（498 文字）
Output_PmrErrorID	WORD	ExecPMCR 命令のエラーコード格納エリア 正常終了：16#0000
Output_PmrErrorIDEx	DWORD	ExecPMCR 命令の拡張エラーコード格納エリア 正常終了：16#00000000
Output_PmrStaErrCode	WORD	プロトコルマクロエラーコード格納エリア 正常終了：16#0000
Output_TransErrCode	WORD	伝送エラー発生状態フラグ列格納エリア [J01_P2_TransErrSta]の格納エリア 正常終了：16#0000

※ExecPMCR 命令の DstDat[] を格納するエリアは、CJ ユニットメモリのアドレスに AT 指定が必要です。

【CJ ユニットデバイス変数】(SCU ユニット)

変数名	データ型	説明
J01_P2_PmrSta	WORD	プロトコルマクロ動作状態 ビット 03～00 : プロトコルマクロエラーコード
J01_P2_PmrExecSta	BOOL	プロトコルマクロ実行中フラグ
J01_P2_PmrSeqEndSta	BOOL	シーケンス End 終了フラグ
J01_P2_PmrSeqAbtSta	BOOL	シーケンス Abort 終了フラグ
J01_P2_TransErrSta	WORD	伝送エラー発生状態



参考

SCU ユニットの変数については、「CJ シリーズ シリアルコミュニケーションユニット ユーザーズマニュアル NJ シリーズ接続編」(SBCD-354)の「2-3 CJ ユニットデバイス変数の詳細」を参照してください。

【システム定義変数】

変数名	データ型	説明
_Port_isAvailable	BOOL	ネットワーク通信命令実行可フラグ TRUE : 使用可能ポートあり FALSE : 使用可能ポートなし



参考

ExecPMCR 命令使用時のシステム定義変数については、「NJ/NX シリーズ コマンドリファレンスマニュアル 基本編」(SBCA-468)の「第 2 章 各命令の説明」の「シリアル通信命令 (ExecPMCR)」の「関連するシステム定義変数」を参照してください。

●内部変数 (インスタンス変数)

ファンクションブロックをプログラム内で実行するための内部変数の一覧です。この内部変数は「インスタンス」と呼ばれ、使用するファンクションブロックの名称を変数のデータ型に指定します。

変数名	データ型	説明
ExecPMCR_instance	ExecPMCR	SCU ユニットに登録された送受信シーケンス (プロトコルデータ) の実行を依頼します。
F_TRIG_instance	F_TRIG	入力信号の立ち下り時、1 タスク周期だけ、TRUE を出力します。



参考

ExecPMCR 命令については、「NJ/NX シリーズ コマンドリファレンスマニュアル 基本編」(SBCA-468)の「第 2 章 各命令の説明」の「シリアル通信命令 (ExecPMCR)」を参照してください。



参考

F_TRIG 命令については、「NJ/NX シリーズ コマンドリファレンスマニュアル 基本編」(SBCA-468)の「第 2 章 各命令の説明」の「シーケンス入力命令 (F_TRIG)」を参照してください。

●内部変数

変数名	データ型	説明
Local_Status	sStatus	通信処理ステータスフラグ列 構造体型「sStatus」で定義
Local_State	DINT	状態処理番号
Local_ExecFlg	BOOL	通信命令起動フラグ
Local_EndExecPMCR	BOOL	プロトコルマクロ実行終了判定フラグ
Local_InPort	_sPort	使用ポート指定
Local_SeqNo	UINT	送受信シーケンス番号
Local_PmrStaCode	WORD	プロトコルマクロエラーコード編集エリア
Local_RecvWordData	ARRAY[0..248] OF WORD	受信データから文字列変換する部分を抜き出し格納するエ リア（498 バイト）
Local_RecvWordSize	UINT	文字列変換するデータのデータサイズ（単位：WORD）
Local_RecvByteData	ARRAY[0..497] OF BYTE	文字列変換するデータをバイト配列に変換したエリア（498 バイト）
Local_RecvByteSize	UINT	文字列変換するデータのデータサイズ（単位：BYTE）

9.5. ST 言語によるプログラム

9.5.1. プログラムの機能構成

本プログラムは ST 言語でプログラミングされています。その機能構成は、以下のとおりです。

大分類	小分類	内容
1. 通信処理	1.1. 通信処理開始 1.2. 通信処理ステータスフラグ列クリア 1.3. 通信実行中状態となり各状態処理へ遷移	通信処理を起動します。
2. 初期処理	2.1. 通信命令初期化 2.2. 通信命令起動フラグ初期化 2.3. エラーコード格納エリアの初期化 2.4. ExecPMCR 命令の入力変数設定 2.5. 受信データ格納エリアの初期化 2.6. 初期設定終了処理	通信命令（ExecPMCR 命令）のパラメータ設定と受信データ格納エリアのクリアを行います。
3. PMCR 通信処理	3.1. 通信処理状況の判定と起動フラグセット 3.2. 通信命令起動	ExecPMCR 命令を実行します。 実行結果の正常／異常を判断します。
4. 処理番号 異常処理	—	存在しない処理番号を検出した場合の異常処理を実行します。

9.5.2. プログラムリスト

本プログラムの内容を掲載します。

●プログラム：Program0（汎用シリアル通信 接続確認プログラム）

1. 通信処理

```
(* =====
  名称：NJシリーズ 汎用シリアル(RS-232C)通信接続確認プログラム（プロトコル加機能使用）
  対象機器：オムロン株式会社 コトリータ 形V430
  シリアル通信ユニット：CJ1W-SCU22（プロトコル加機能使用,ユニット番号：0 シリアルポート番号：2）
  バージョン情報：V1.00 2019/5 新規
  (C)Copyright OMRON Corporation 2013 All Rights Reserved.
  ===== *)
```

(* 1. 通信処理

通信開始スイッチ：Input_Start

通信処理ステータスフラグ列：Local_Status<STRUCT>

.Busy：通信実行中 .Done：通信正常終了 .Error：通信異常終了

状態処理番号：Local_State

10：初期処理 11：PMCR通信処理 *)

(* 1.1. 通信処理開始

通信処理ステータスフラグ列がクリアな状態で通信開始スイッチがONされた場合に通信処理を開始 *)

IF Input_Start AND

NOT (Local_Status.Busy OR Local_Status.Done OR Local_Status.Error) THEN

Local_Status.Busy:=TRUE;

Local_State:=10; //10:初期処理へ

END_IF;

(* 1.2. 通信処理ステータスフラグ列クリア

通信処理非実行状態で通信開始スイッチOFFにより通信処理ステータスフラグ列クリア *)

IF NOT Input_Start AND NOT Local_Status.Busy THEN

Local_Status.Done:=FALSE;

Local_Status.Error:=FALSE;

END_IF;

(* 1.3. 通信処理実行中状態となり各状態処理へ遷移 *)

IF Local_Status.Busy THEN

CASE Local_State OF

2. 初期処理

SCU ユニット番号、使用するポート番号、プロトコルマクロシーケンス番号、送信データサイズなどの入力値を変更する場合は、以下の赤枠の部分を変更します。

(* 2. 初期処理

- ・通信全体の各種初期化とパラメータ設定
- ・送信データの設定と受信データ格納エリアの初期化 *)

10:

(* 2.1. 通信命令の初期化 *)

```
ExecPMCR_instance(
    Execute:=FALSE,
    SrcDat:=Input_SendData[0],
    DstDat:=Output_RecvData[0]);
```

(* 2.2. 通信命令起動フラグ初期化 *)

```
Local_ExecFlg:=FALSE;
```

(* 2.3. I/Oポート格納エリアの初期化 *)

```
Clear(Output_PmrErrorID); //ExecPMCR命令I/Oポート格納エリア
Clear(Output_PmrErrorIDEx); //ExecPMCR命令拡張I/Oポート格納エリア
Clear(Output_PmrStaErrCode); //プロトコルマクロI/Oポート格納エリア
Clear(Output_TransErrCode); //伝送I/Oポート格納エリア
```

(* 2.4. ExecPMCR命令入力変数設定 *)

```
Local_InPort.UnitNo:=_eUnitNo#_CBU_No00; //SCUユニット番号
Local_InPort.PhysicPortNo:=USINT#2; //物理ポート番号
Local_SeqNo:=UINT#900; //プロトコルマクロシーケンス番号
Input_SendData[0]:=WORD#0; //送信データサイズ (単位: WORD)
```

(* 2.5. 受信データ格納エリアの初期化 *)

```
Clear(Output_RecvData); //ExecPMCR受信データ格納エリア
Clear(Local_RecvWordData); //文字列変換用WORDエリア
Clear(Local_RecvByteData); //文字列変換用BYTEエリア
Clear(Output_RecvMessage); //文字列格納エリア
```

(* 2.6. 初期設定終了処理 *)

```
Local_State:=11; //11:PMCR通信処理へ
```

3. PMCR 通信処理

(* 3. PMCR通信処理

・指定したシリアルポートから通信処理実行 *)

11:

(* 3.1. 通信処理状況の判定と起動フラグセット *)

(* 3.1.1. 7°ポート追加実行終了時の正常/異常判定処理 *)

F_TRIG_instance(J01_P2_PmrExecSta,Local_EndExecPMCR); //7°ポート追加実行終了判定

IF Local_EndExecPMCR THEN

IF NOT J01_P2_PmrSeqEndSta THEN //7°ポート追加異常終了

Output_TransErrCode:=J01_P2_TransErrSta;

Local_Status.Error:=TRUE;

ELSE

//正常終了処理

Output_TransErrCode:=WORD#16#0000;

Local_Status.Done:=TRUE;

//受信データの文字列変換

Local_RecvWordSize:=WORD_TO_UINT(Output_RecvData[0])-1;

MemCopy(Output_RecvData[1],Local_RecvWordData[0],Local_RecvWordSize);

ToAryByte(Local_RecvWordData,eBYTE_ORDER#_HIGH_LOW,Local_RecvByteData[0]);

Local_RecvByteSize:=Local_RecvWordSize*2;

Output_RecvMessage:=AryToString(Local_RecvByteData[0],Local_RecvByteSize);

END_IF;

Local_EndExecPMCR:=FALSE;

Local_Status.Busy:=FALSE;

Local_ExecFlg:=FALSE;

Local_State:=0; //0: 通信非実行状態へ

(* 3.1.2. 7°ポート追加実行中の異常判定処理 *)

ELSIF ExecPMCR_instance.Error THEN //ExecPMCR命令異常

Output_PmrErrorID:=ExecPMCR_instance.ErrorID;

Output_PmrErrorIDEx:=ExecPMCR_instance.ErrorIDEx;

MoveDigit(J01_P2_PmrSta,USINT#0,Local_PmrStaCode,USINT#0,USINT#1);

Output_PmrStaErrCode:=Local_PmrStaCode;

Output_TransErrCode:=J01_P2_TransErrSta;

Local_Status.Busy:=FALSE;

Local_Status.Error:=TRUE;

Local_ExecFlg:=FALSE;

Local_State:=0; //0: 通信非実行状態へ

(* 3.1.3. 通信命令起動フラグセット *)

ELSIF _Port_isAvailable AND

NOT J01_P2_PmrExecSta AND

NOT ExecPMCR_instance.Busy THEN

Local_ExecFlg:=TRUE;

END_IF;

(* 3.2. 通信命令実行 *)

ExecPMCR_instance(

Execute:=Local_ExecFlg,

Port:=Local_InPort,

SeqNo:=Local_SeqNo,

SrcDat:=Input_SendData[0],

DstDat:=Output_RecvData[0]);

```
4. 処理番号異常処理
(* 4. 処理番号異常処理
  ・ 存在しない処理番号の異常処理 *)
99:
  Output_TransErrCode:=WORD#16#0010;
  Local_Status.Busy:=FALSE;
  Local_Status.Error:=TRUE;
  Local_State:=0; //0: 通信非実行状態へ

ELSE
  Local_State:=99; //99: 処理番号異常処理へ

END_CASE;

END_IF;
```

9.6. プロトコルマクロデータ

プロトコルマクロデータは、「シーケンス」、「ステップ」、「送受信メッセージ」、「受信マトリクス」という構成要素をもち、以下のような構成になります。

- ステップ（1回の送受信）に対して受信メッセージのフォーマットが1種類だけの場合
 - ・ステップに対して受信メッセージと送信メッセージを1つずつ設定

シーケンス No.900	ステップ No.00	送信メッセージ 00	受信メッセージ 00
⋮			
シーケンス No.xxx	ステップ No.yy	送信メッセージ yy	受信メッセージ yy

xxx : 最大 999 まで / yy : 最大 15 まで

- ステップ（1回の送受信）に対して受信メッセージのフォーマットが数種類存在する場合
 - ・ステップに対して送信メッセージと受信マトリクスを設定
 - ・受信マトリクスに対して数種類の「ケース」（受信メッセージ）を設定

シーケンス No.900	ステップ No.00	送信メッセージ 00	<受信マトリクス>	
⋮			ケース No.00	受信メッセージ 00
⋮	ステップ No.yy			
⋮			ケース No.zz	受信メッセージ zz
シーケンス No.xxx	yy : 最大 15 まで	zz : 最大 14 まで	ケース No.15	Other

xxx : 最大 999 まで ケース No.15 は自動設定

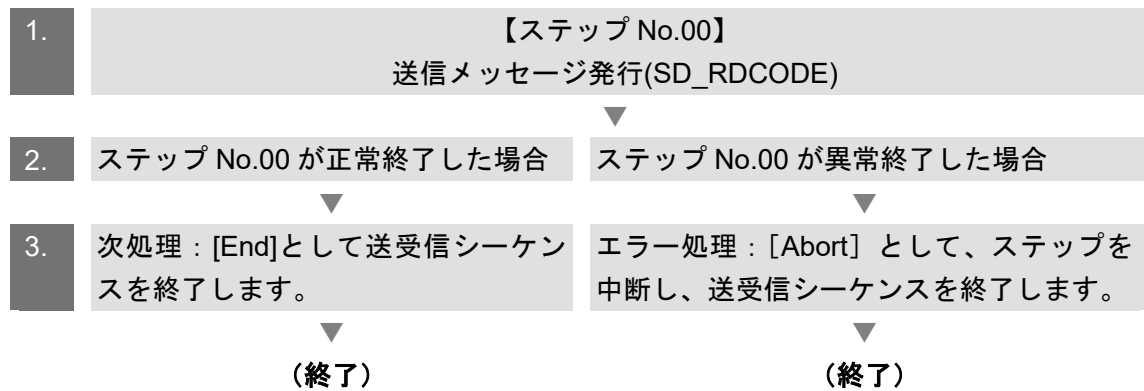
9.6.1. プロトコルマクロデータの構成

本プロトコルマクロデータでは、送信メッセージ (SD_RDCODE) に対し、1種類の受信メッセージ (RV_RDCODE) のみですので、受信マトリクスは使用せず、以下のような構成としています。詳しくは、「9.6.6. 受信メッセージの設定」を参照してください。

シーケンス No.900	ステップ No.00	SD_RDCODE	RV_RDCODE
--------------	------------	-----------	-----------

9.6.2. プロトコルマクロの処理手順

プロトコルマクロの処理手順です。



9.6.3. シーケンスの設定

本プロトコルマクロデータでは、送受信シーケンス No.900 を使用し、「読取りトリガ」（コードの読み取り）を行います。送受信シーケンスに対する設定項目に「タイムアウト時間」があります。



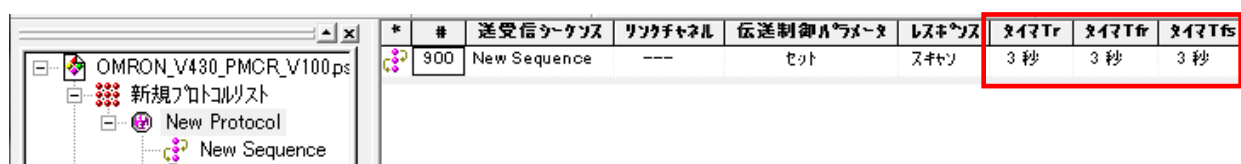
参考

シーケンスの設定内容の詳細については、「CX-Protocol オペレーションマニュアル」(SBCA-307)の「4-2 シーケンス単位（ステップ共通）の設定内容」を参照してください。

●タイムアウト時間の設定

シーケンスに対する「タイムアウト時間（タイマ Tr、Tfr、Tfs）」の設定内容です。

【送受信シーケンスの設定画面】



#	送受信シーケンス	リンクチャネル	伝送制御パラメータ	レスポンス	タイマ Tr	タイマ Tfr	タイマ Tfs
900	New Sequence	---	セット	スキャン	3 秒	3 秒	3 秒

【設定内容】

項目	内容	説明
タイマ Tr	受信待ち監視時間	シーケンス内のステップにおいて、受信待ち状態になってから最初のデータ（ヘッダ）を受信するまでの時間を監視します。本プロトコルマクロデータでは [3 秒] に設定しています。
タイマ Tfr	受信完了監視時間	シーケンス内のステップにおいて、最初のデータを受信してから受信が完了するまでの時間を監視します。本プロトコルマクロデータでは [3 秒] に設定しています。
タイマ Tfs	送信完了監視時間	ヘッダを送信してから最後のデータを送信するまでの時間を監視します。本プロトコルマクロデータでは [3 秒] に設定しています。



参考

監視時間の計算方法については、「CX-Protocol オペレーションマニュアル」(SBCA-307)の「5-5 監視時間の計算方法」を参照してください。

9.6.4. ステップの設定

送受信シーケンス No.900 に対する「ステップ」の設定内容です。ステップに対する設定項目に「リトライ回数」、「送受信メッセージ（メッセージ名）」、「次処理」と「エラー処理」があります。本プロトコルマクロデータのシーケンスは、「ステップ No.00」のみの構成となっています。



参考

ステップの設定内容の詳細については、「CX-Protocol オペレーションマニュアル」(SBCA-307)の「4-3 ステップ単位の設定内容」を参照してください。

●リトライ回数の設定

ステップに対する「リトライ回数」の設定内容です。エラーが発生した場合に、指定した回数分（0～9 回）ステップをリトライします。その結果、さらにエラーが発生する場合は、「エラー処理」に遷移します。

リトライ回数は、Send&Receive コマンド時のみ有効です。

【ステップの設定画面】

* ステップ*	リセットカウンタ	コマンド*	リトライ回数	送信ウェイト時間	送信メッセージ*	受信メッセージ*	レスポンス書込有無	次処理	エラー処理
00	リセット/001	Send & Receive	3	---	SD_RDCODE	RV_RDCODE	あり	End	Abort

【設定内容】

ステップ No.	リトライ回数
00	3

●送受信メッセージ（メッセージ名）の設定

ステップに対する「送受信メッセージ」の設定内容です。別途登録されている送信メッセージ名および受信メッセージ名を設定しています。

【ステップの設定画面】

* ステップ*	リセットカウンタ	コマンド*	リトライ回数	送信ウェイト時間	送信メッセージ*	受信メッセージ*	レスポンス書込有無	次処理	エラー処理
00	リセット/001	Send & Receive	3	---	SD_RDCODE	RV_RDCODE	あり	End	Abort

【設定内容】

ステップ No.	送信メッセージ	受信メッセージ
00	SD_RDCODE	RV_RDCODE

●次処理とエラー処理の設定

ステップに対する「次処理」と「エラー処理」の設定内容です。ステップの実行が正常に終了した場合に「次処理」の設定内容が実行され、通信異常が発生した場合に「エラー処理」の設定内容が実行されます。

【ステップの設定画面】



ステップ*	リセットカウンタ	コマンド*	リトライ回数	送信ウェイト時間	送信メッセージ*	受信メッセージ*	レスポンス書込有無	次処理	エラー処理
00	リセット/001	Send & Receive	3	---	SD,RDCODE	RV,RDCODE	あり	End	Abort

【設定内容】

ステップ No.	次処理	エラー処理
00	End	Abort

【処理項目一覧】

処理.	内容
End	送受信シーケンスを終了します
Next	次のステップ No.に遷移します
Abort	ステップを中断し、送受信シーケンスを終了します
Goto	指定のステップ No.に遷移します
受信マトリクス	「受信マトリクス」の設定に従います

9.6.5. 送信メッセージの設定

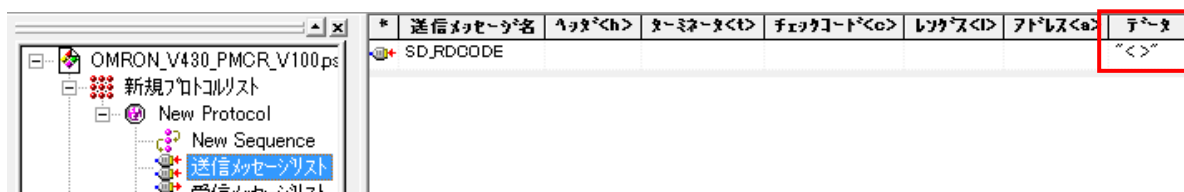
送信メッセージの設定内容です。



参考

送信メッセージの設定内容の詳細については、「CX-Protocol オペレーションマニュアル」(SBCA-307)の「4-4 送受信メッセージの設定内容」を参照してください。

【送信メッセージの設定画面】



●送信メッセージ「SD_RDCODE」の設定

【設定内容】

<>
①

No.	コード	内容
②	"< >"	定数 ASCII

※ヘッダ<h>、フッタ<f>ともに [なし] に設定しています。

【送信メッセージのコマンドフォーマット】

「SD_RDCODE」の設定に従って、SCU ユニットから相手機器に送信されるメッセージのコマンドフォーマットです。

"< >"

コマンド	バイト数	備考
"< >"	1	固定 : "< >" (16#3C203E) (相手機器コマンド「読取りトリガ」)

9.6.6. 受信メッセージの設定

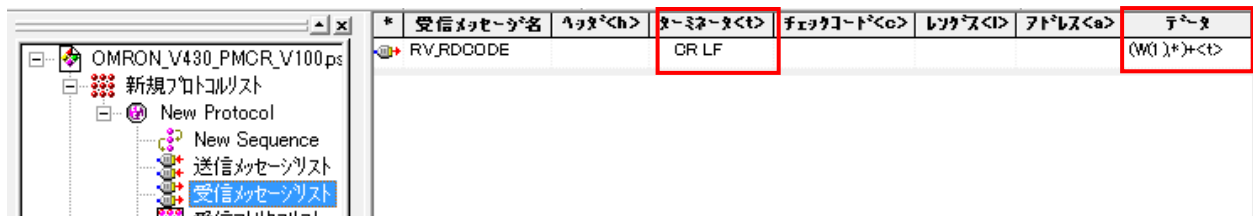
受信メッセージの設定内容です。



参考

受信メッセージの設定内容の詳細については、「CX-Protocol オペレーションマニュアル」(SBCA-307)の「4-4 送受信メッセージの設定内容」を参照してください。

【受信メッセージの設定画面】



●受信メッセージ「RV_RDCODE」の設定

【設定内容】

(W(1),*)+<t>

① ②

No.	コード	内容
①	(W(1),*)	メッセージデータ：変数 可変長データを全部取り込み、ExecPMCR 命令の「DstDat」に指定した配列変数[n]+[1]番目の配列要素以降に格納します（本プログラムでは、Output_RecvData[1]～）
②	<t>（ターミネータ）	メッセージデータ：定数 Hex、タイプ：Code、データ：CR LF

※ヘッダ<h>は「なし」に設定しています。

【受信メッセージのレスポンスフォーマット】

「RV_RDCODE」の設定に従って SCU ユニットで受信する相手機器からの「受信メッセージ」のレスポンスフォーマットです。

コード読み取り値	<CR><LF>
----------	----------

コマンド	バイト数	備考
データ	可変長	可変：コード読み取り値
ターミネータ	1	固定：<CR> <LF>(16#0D0A)

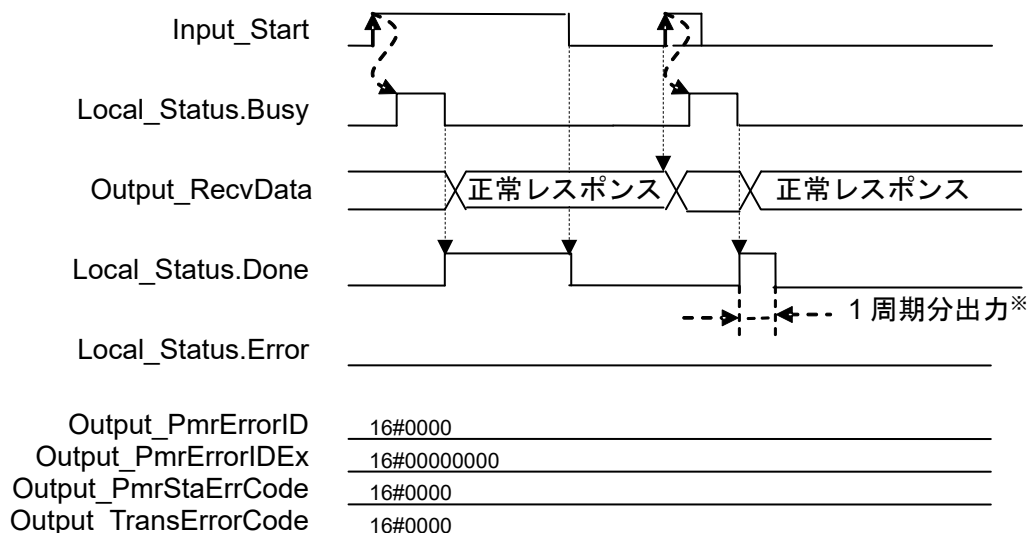
9.7. タイムチャート

プログラムのタイムチャートです。

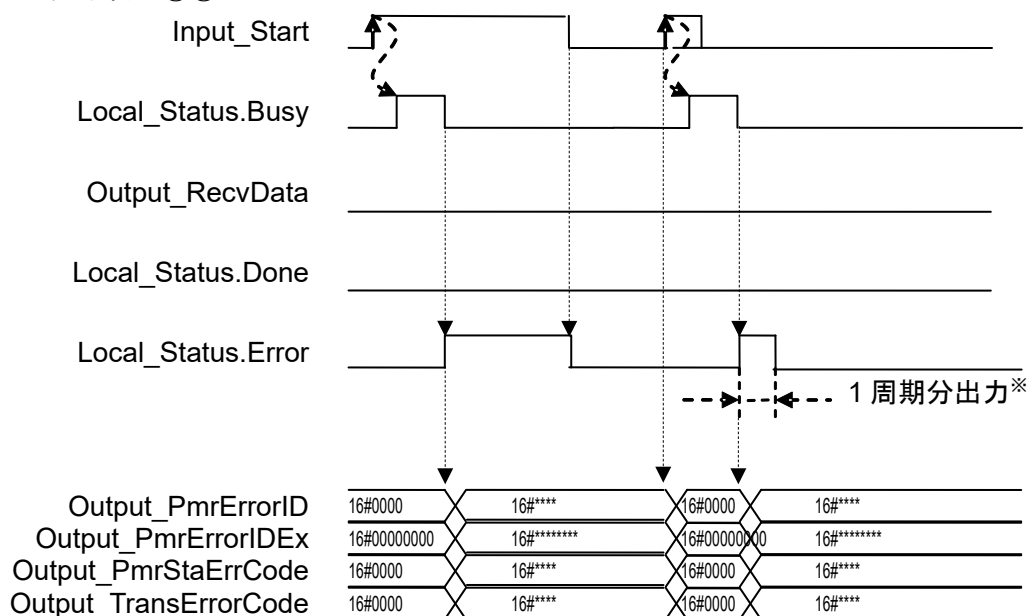
タイムチャートパターンの定義は以下のとおりです。

パターン	正常終了	異常終了① ExecPMCR 命令異常	異常終了② プロトコルマクロ異常
コマンド	正常	異常	異常
相手機器	正常	正常 or 異常	正常 or 異常
レスポンス	あり	なし	なし

●正常終了



●異常終了①②



※実行途中(Busy=TRUE)に[Input_Start]を「TRUE」から「FALSE」に変更した場合、処理を完了(Busy=FALSE)してから正常または異常を 1 周期出力します。

9.8. 異常処理

本プログラムを実行して発生する異常内容を以下に示します。

9.8.1. ExecPMCR 命令異常

ExecPMCR 命令が異常終了したときに発生する異常コードです。

●Output_PmrErrorID

ExecPMCR 実行後、[ExecPMCR_instance.ErrorID]の内容がセットされます。

コード	内容
16#0000	正常終了
16#0400	命令の入力パラメータが入力変数の範囲を超えています。
16#0406	命令で指定されたデータのメモリアドレスやデータサイズが不適切です。
16#0407	命令の演算結果が、出力パラメータのデータ領域の範囲を超えています。
16#0800	FINS コマンドの送受信で異常が発生しました。
16#0801	FINS ポートが使用中です。



参考

ErrorID の詳細については、「NJ/NX シリーズ コマンドリファレンスマニュアル 基本編」(SBCA-468)の「付録」の「A-1 エラーコードの概要」「A-2 エラーコードの詳細」を参照してください。

●Output_PmrErrorIDEx

[ExecPMCR_instance.ErrorID]の値が「16#0800」の場合に、
[ExecPMCR_instance.ErrorIDEx]の内容がセットされます。

コード	内容
16#00000000	正常終了
16#00001106	「SeqNo」の値が、登録されていない送受信シーケンス番号です。
16#00002201	既に、本命令実行中です。「Busy」および「J01_P2_PmrExecSta」の値が、TRUE です。
16#00002202	プロトコル切り替え中のため、実行できません。
16#00002401	プロトコルマクロデータが、サム値異常か、またはデータ転送中です。



参考

ErrorIDEx の詳細については、「NJ/NX シリーズ コマンドリファレンスマニュアル 基本編」(SBCA-468)の「第2章 各命令の説明 通信命令 (ExecPMCR)」を参照してください。

9.8.2. プロトコルマクロエラーコード

プロトコルマクロに起因するエラーが発生した場合のプロトコルマクロエラーコードです。
[Output_PmrStaErrCode]にセットされます。

●Output_PmrStaErrCode

[J01_P2_PmrSta]の下位 4 ビットの内容がセットされます。

コード	内容
16#0000	エラーなし
16#0002	シーケンス番号エラー
16#0003	データ読み出し／書き込み領域超えエラー
16#0004	プロトコルデータ文法エラー



参考

プロトコルマクロエラーコード (PmrStaErrCode) の詳細については、「CJ シリーズ シリアルコミュニケーションユニット ユーザーズマニュアル NJ シリーズ接続編」(SB CD-354)の「4-2-6 プロトコルステータス」を参照してください。

9.8.3. 伝送エラー発生状態

伝送エラーが発生した場合のエラーフラグ列です。
[Output_TransErrCode]にセットされます。

●Output_TransErrCode

[J01_P2_TransErrSta]の内容がセットされます。

ビット	内容
15	1 : 伝送エラー発生 0 : 伝送エラー発生なし
14	1 : 送信完了監視時間オーバー 0 : 正常
13	1 : 受信完了監視時間オーバー 0 : 正常
12	1 : 受信待ち監視時間オーバー 0 : 正常
8～11	リトライ回数
7	1 : FCS チェックエラー 0 : 正常
6	1 : コマンドエラー 0 : 正常
5	1 : タイムアウト 0 : 正常
4	1 : オーバーランエラー発生 0 : 正常
3	1 : フレーミングエラー発生 0 : 正常
2	1 : パリティエラー発生 0 : 正常
0、1	(未使用)



参考

伝送エラー発生状態 (TransErrCode) の詳細については、「CJ シリーズ シリアルコミュニケーションユニット ユーザーズマニュアル NJ シリーズ接続編」(SB CD-354)の「4-2-5 ステータス参照用の CJ ユニットデバイス変数」を参照してください。

10. 改訂履歴

改訂記号	改訂年月日	改訂理由・改訂ページ
A	2019 年 10 月	初版
B	2021 年 3 月	V320-F/V420-F シリーズを追加
C	2023 年 2 月	誤記修正

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

製品に関するお問い合わせ先

お客様
相談室

フリー
通話

0120-919-066

携帯電話・IP電話などではご利用いただけ
ませんので、右記の電話番号へおかけください。

055-982-5015
(通話料がかかります)

受付時間：9:00～19:00 (12/31～1/3を除く)

オムロンFAクイックチャット

www.fa.omron.co.jp/contact/tech/chat/

技術相談員にチャットでお問い合わせいただけます。(I-Webメンバーズ限定)

受付時間：平日9:00～12:00 / 13:00～17:00 (土日祝日・年末年始・当社休業日を除く)

※受付時間、営業日は変更の可能性があります。最新情報はリンク先をご確認ください。

その他のお問い合わせ：納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。



オムロン制御機器の最新情報をご覧ください。緊急時のご購入にもご利用ください。

www.fa.omron.co.jp

- 本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。
- 本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物(又は技術)に該当するものを輸出(又は非居住者に提供)する場合は同法に基づく輸出許可、承認(又は役務取引許可)が必要です。

オムロン商品のご用命は