OMRON

マシンオートメーションコントローラ

CJシリーズ

EtherNet/IP™ユニット

ユーザーズマニュアル
NJシリーズ接続編

形CJ1W-EIP21
形CJ1W-EIP21S

EtherNet/IPユニット



SBCD-355N

おことわり

- 本マニュアルの内容の一部または全部を無断で複写、複製、転載することを禁じます。
- 本マニュアルの内容に関しては、改良のため予告なしに仕様などを変更することがあります。
  あらかじめご了承ください。
- 本マニュアルの内容に関しては、万全を期しておりますが、万一ご不審な点や誤りなど、お気づきのことがありましたら、お手数ですが、弊社支店または営業所までご連絡ください。
  その際、巻末記載のMan.No.（マニュアルNo.）も併せてお知らせください。

商標

- Sysmacは、オムロン株式会社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。
- Microsoft、Windows、Excel、Visual Basic、Microsoft Edgeは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- EtherCAT® は、Beckhoff Automation GmbH（ドイツ）よりライセンスを受けた特許取得済み技術であり登録商標です。
- ODVA、CIP、CompoNet、DeviceNet、EtherNet/IPはODVAの商標です。
- SD、SDHCロゴは、SD-3C, LLCの商標です。

その他、本文中に掲載している会社名および製品名は、それぞれ各社の商標または登録商標です。

著作権

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。

はじめに

このたびは、CJ シリーズ EtherNet/IP ユニットをお買い上げいただきまして、ありがとうございます。
本マニュアルは、CJ シリーズ EtherNet/IP ユニットを使用する上で、必要な情報を記載しています。
お使いになる前に本マニュアルをよく読んで、機能・性能などを十分ご理解いただき、システムの構築にご活用ください。
また、お読みになった後も本マニュアルは大切に保管して、いつも手元においてお使いください。

対象となる読者の方々

本マニュアルは、次の方を対象に記述しています。
電気の知識（電気工事士あるいは同等の知識）を有する方で
- FA 機器の導入を担当される方
- FA システムを設計される方
- FA 機器を設置、接続される方
- FA 現場を管理される方

また、プログラミング言語に関して、国際標準規格 IEC61131-3 もしくは国内標準規格 JIS B3503 に規定される内容をご理解いただいている方を対象に記述しています。

対象となる製品

本マニュアルは以下の製品を対象にしています。
- CJ シリーズ EtherNet/IP ユニット
  - 形 CJ1W-EIP21
  - 形 CJ1W-EIP21S

本マニュアルは、形 CJ1W-EIP21/EIP21S を、NJ シリーズ CPU ユニットに接続して使用する上で、必要な情報を記載しています。
CJ シリーズ EtherNet/IP ユニットを NJ シリーズ CPU ユニットに接続するには、次のバージョンを使用してください。

EtherNet/IPユニット	NJシリーズCPUユニット	Sysmac Studio
形 CJ1W-EIP21　Ver.2.1 以降	Ver.1.01 以降	Ver.1.02 以降
形 CJ1W-EIP21　Ver.3.0 以降	Ver.1.01 以降	Ver.1.11 以降
形 CJ1W-EIP21S　Ver.1.0 ロット番号 241001口 以降 *1	Ver.1.67 以降	Ver.1.60 以降

*1. ロット番号 241001口 以降の場合は、ユニットの前面右下に「+NJ」または「+CJ/CP/NJ」と印字されています。前面右下の印字については、「1-3-2 各部の名称と機能」を参照してください。

分冊構成

本製品のマニュアルは、次表のとおりに分冊されています。目的に応じてお読みいただき、本製品を活用してください。
本製品の操作には、おもにオートメーションソフトウェア Sysmac Studio を使用します。Sysmac Studio については、🕮『Sysmac Studio Version 1 オペレーションマニュアル（SBCA-470）』を参照してください。

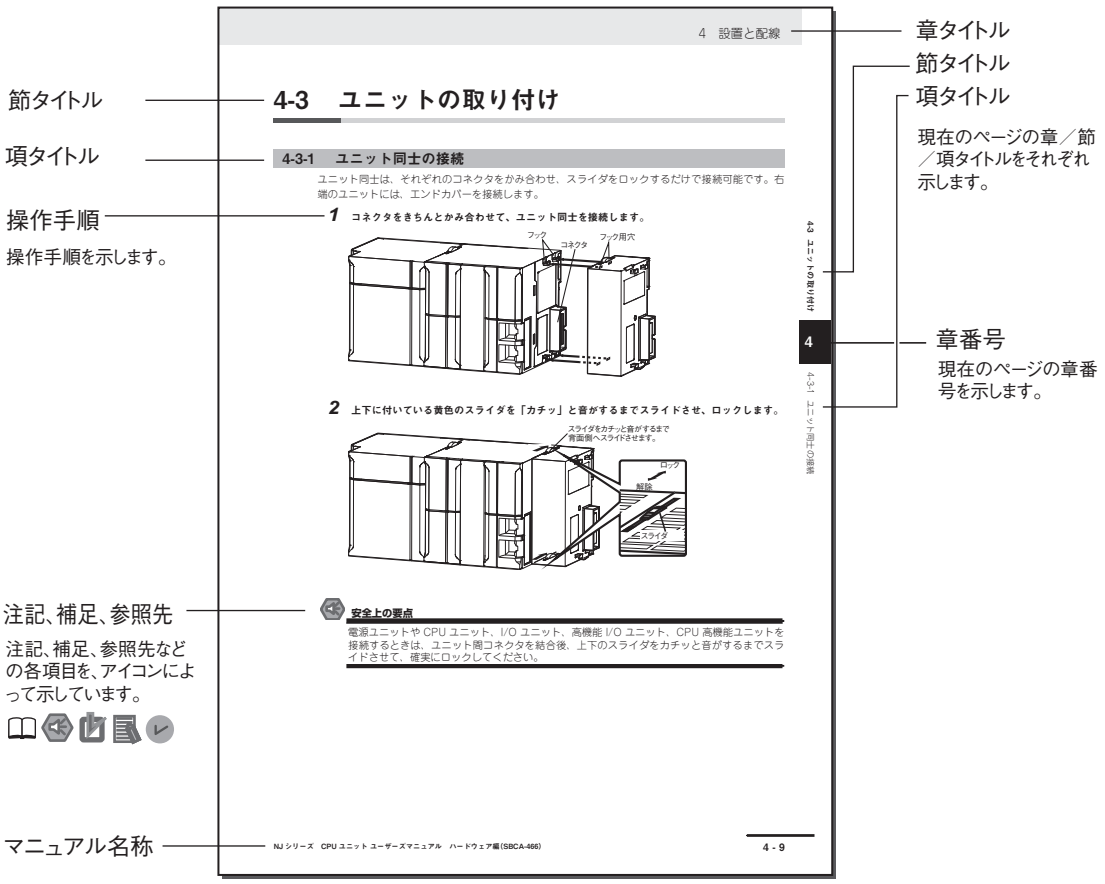
使用目的	マニュアル											
	基本情報											
	NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	NJ/NX シリーズ コマンドリファレンスマニュアル 基本編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	NJ/NX シリーズ コマンドリファレンスマニュアル モーション編	NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT ポート ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP ポートユーザーズマニュアル	NJ シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	NJ シリーズ SECS/GEM 搭載 CPU ユニット ユーザーズマニュアル	NJ/NX シリーズ トラブルシューティングマニュアル	CJ シリーズ各ユニット ユーザーズマニュアル NJ シリーズ接続編
NX シリーズの概要を知る	●											
NJ シリーズの概要を知る		●										
装着、設置、ハードウェア設定を行う												
モーション制御を行う場合	●				●							
EtherCAT を使用する場合							●					
EtherNet/IP を使用する場合		●						●				
データベース接続サービスを使用する場合									●			
GEM サービスを使用する場合										●		
CJ シリーズユニットを使用する場合												●
ソフトウェア設定を行う												
モーション制御を行う場合					●							
EtherCAT を使用する場合			●				●					
EtherNet/IP を使用する場合								●				
データベース接続サービスを使用する場合									●			
GEM サービスを使用する場合										●		
ユーザプログラムを作成する												
モーション制御を行う場合					●	●						
EtherCAT を使用する場合							●					
EtherNet/IP を使用する場合			●	●				●				
データベース接続サービスを使用する場合									●			
GEM サービスを使用する場合										●		
CJ シリーズユニットを使用する場合												●
異常処理を組み込む場合											●	
動作確認やデバッグを行う												
モーション制御を行う場合					●							
EtherCAT を使用する場合			●				●					
EtherNet/IP を使用する場合								●				
データベース接続サービスを使用する場合									●			
GEM サービスを使用する場合										●		
異常管理機能やトラブル発生時の対処方法を知る[*1]	△	△	△		△		△	△	△	△	●	△
保守作業について知る												
モーション制御を行う場合	●				●							
EtherCAT を使用する場合		●					●					
EtherNet/IP を使用する場合								●				
CJ シリーズユニットを使用する場合												●

*1. 異常管理の考え方と異常項目の概要については、🕮『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。異常の詳細については、異常の内容に応じて△マークのマニュアルを参照してください。

マニュアルの読み方

ページ構成

このマニュアルの各ページの構成は次のとおりです。



このページは、説明用の見本です。実際の内容とは異なります。

アイコン

このマニュアルで使われているアイコンには、次のような意味があります。

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、使用する上で参考となる内容について説明しています。

バージョン関連情報

CPU ユニットや Sysmac Studio のバージョンによる性能や機能の違いについて説明しています。

詳しい情報や、関連情報の記載されている場所を示します。

表記についての留意事項

本マニュアルでは、Sysmac Studio から実機コントローラにデータを転送することを「ダウンロード」、実機コントローラからデータを Sysmac Studio に転送することを「アップロード」と記述しています。

Sysmac Studio では、「ダウンロード」および「アップロード」は、いずれも「同期」機能を使用します。Sysmac Studio の「同期」機能は、パソコン上のデータと実機コントローラのデータを自動照合し、どちらの方向に転送するかをユーザが選択する機能です。

目次構成

1 概要

2 Ethernet ネットワークの配線

3 CJユニットデバイス変数への割付

4 IPアドレスを決定する方法

5 Sysmac StudioによるEtherNet/IPユニットのユニット設定一覧

6 通信テスト

7 タグデータリンク機能

8 メッセージ通信機能

9 FTPサーバ機能

10 時計情報自動調整機能

11 SNMP（エージェント）機能

12 セキュリティ機能

13 ソケットサービス機能

14 通信性能と通信負荷

15 異常とその処理

A 付録

I 索引

目次

はじめに **1**
対象となる読者の方々 1
対象となる製品 1
分冊構成 **2**
マニュアルの読み方 **4**
ページ構成 4
アイコン 4
表記についての留意事項 5
目次構成 **7**
ご承諾事項 **16**
安全上の注意 **18**
安全に使用していただくための表示と意味 18
図記号の説明 19
警告 20
注意 22
安全上の要点 **23**
使用上の注意 **30**
法規と規格 **33**
海外でのご使用 33
EMC および電気安全に関わる法規制への適合 33
KC 規格への適合 34
船級規格への適合 34
ソフトウェアのライセンスと著作権 34
バージョン **35**
バージョンの確認方法 35
バージョン変更時の手順 37
CIP リビジョンとの対応表 38
ユニットバージョンと周辺ツールのバージョンについて 38
ユニットバージョンと、接続可能な CPU ユニットの機種について 38
関連マニュアル **40**
マニュアル改訂履歴 **42**

第 1 章　概要

1-1　概要 **1-2**
1-1-1　EtherNet/IP の特長 1-2
1-1-2　EtherNet/IP ユニットの特長 1-3
1-2　システム構成と構成機器 **1-5**
1-2-1　ネットワーク構築に必要な機器 1-5
1-2-2　ネットワーク構築に必要な周辺ツール 1-6
1-3　EtherNet/IP ユニット **1-7**
1-3-1　仕様 1-7
1-3-2　各部の名称と機能 1-9
1-3-3　外形寸法 1-16
1-4　通信サービスの概要 **1-17**
1-4-1　CIP 通信（Common Industrial Protocol）サービス 1-17
1-4-2　BOOTP クライアント機能 1-19
1-4-3　FTP サーバ機能 1-19
1-4-4　時計自動調整機能 1-20
1-4-5　ホスト名指定機能 1-20
1-4-6　SNMP（エージェント）機能 1-21
1-4-7　ソケットサービス機能（形 CJ1W-EIP21S のみ） 1-21

1-5 EtherNet/IP 通信の使用手順 1-22
1-5-1 基本操作 1-22
1-5-2 タグデータリンクを使用時 1-24
1-5-3 EtherNet/IP ネットワークの各機能を使用する 1-24

第 2 章 Ethernet ネットワークの配線

2-1 ユニットの前面スイッチの設定 2-2
2-2 CPU ラックまたは増設ラックへの装着 2-3
2-3 ネットワーク構成機器の選定 2-4
2-3-1 ネットワーク構成機器の推奨品 2-4
2-3-2 スイッチングハブの種類 2-5
2-3-3 スイッチングハブの機能 2-5
2-3-4 スイッチングハブ選定時の留意事項 2-6
2-4 ネットワークの布設について 2-8
2-4-1 ネットワーク布設時の基本的注意事項 2-8
2-4-2 ネットワーク構成機器の推奨品 2-8
2-4-3 ツイストペアケーブル布設時の注意事項 2-8
2-4-4 スイッチングハブの設置・接続時の注意事項 2-10
2-5 ネットワークへの接続 2-11
2-5-1 Ethernet 接続コネクタ 2-11
2-5-2 ケーブルの接続方法 2-11
2-5-3 リレー接点出力ユニット使用時のノイズ対策 2-12

第 3 章 CJ ユニットデバイス変数への割付

3-1 CPU ユニットとのデータ交換 3-2
3-1-1 データの流れ 3-2
3-1-2 各種変数の指定／作成方法 3-5
3-2 CJ ユニットデバイス変数の詳細 3-6
3-2-1 CJ ユニットデバイス変数への割付 3-6
3-2-2 ソフトスイッチ用の CJ ユニットデバイス変数 3-6
3-2-3 ステータス用の CJ ユニットデバイス変数 3-8
3-2-4 システム設定用の CJ ユニットデバイス変数 3-15
3-2-5 ソケットサービス関連の CJ ユニットデバイス変数（形 CJ1W-EIP21S のみ） 3-17
3-3 ステータスエリアのユーザ定義設定 3-25
3-3-1 ユーザ定義設定の概要 3-25
3-3-2 ユーザ定義設定の設定方法 3-25
3-3-3 ユーザ設定のステータスエリアへのアクセス方法 3-26
3-3-4 ユーザ定義設定エリアに割り付けるステータス情報 3-26

第 4 章 IP アドレスを決定する方法

4-1 IP アドレスとは 4-2
4-1-1 IP アドレスの構成 4-2
4-1-2 IP アドレスの割り当て 4-3
4-1-3 サブネットマスク 4-3
4-1-4 CIDR 機能 4-3
4-2 EtherNet/IP ユニットの自 IP アドレス 設定方法4-4
4-2-1 IP アドレスを設定する方法について 4-4
4-2-2 IP アドレスの設定手順 4-5
4-2-3 オンライン接続する 4-7
4-3 プライベート／グローバルアドレス 4-10
4-3-1 グローバルアドレスとプライベートアドレス 4-10

4-3-2 EtherNet/IP ユニットがプライベートアドレスの場合 4-11
4-3-3 EtherNet/IP ユニットがグローバルアドレスの場合 4-12

第 5 章 Sysmac Studio による EtherNet/IP ユニットのユニット設定一覧

5-1 EtherNet/IP ユニットのユニット設定について 5-2
5-1-1 ユニット設定の反映について 5-2
5-1-2 ユニット設定をデフォルトに戻す 5-2
5-2 ［TCP/IP 設定］ダイアログ 5-3
5-3 ［LINK 設定］ダイアログ 5-8
5-4 ［FTP 設定］ダイアログ 5-9
5-5 ［SNTP 設定］ダイアログ 5-10
5-6 ［SNMP 設定］ダイアログ 5-11
5-7 ［SNMP トラップ設定］ダイアログ 5-13
5-8 ［ステータスエリア設定］ダイアログ 5-15

第 6 章 通信テスト

6-1 通信テスト 6-2
6-1-1 PING コマンドとは 6-2
6-1-2 EtherNet/IP ユニットでの使用方法 6-2
6-1-3 ホストコンピュータでの使用方法 6-3

第 7 章 タグデータリンク機能

7-1 タグデータリンクの概要 7-2
7-1-1 タグデータリンクとは 7-2
7-1-2 タグデータリンクのデータエリア 7-4
7-1-3 タグデータリンクの機能・性能一覧 7-8
7-1-4 使用方法の概要 7-9
7-1-5 タグデータリンクの開始／停止 7-12
7-1-6 コントローラステータスについて 7-13
7-1-7 タグデータリンクするデータの同期性について 7-15
7-2 タグデータリンクの設定 7-19
7-2-1 Network Configurator の起動 7-19
7-2-2 タグデータリンク設定手順 7-21
7-2-3 デバイスの登録 7-22
7-2-4 タグ／タグセットの作成 7-24
7-2-5 コネクション設定 7-38
7-2-6 ウィザードによるコネクションの作成 7-48
7-2-7 デバイスのドラッグ & ドロップによるコネクションの作成 7-51
7-2-8 Network Configurator のネットワーク接続手順 7-54
7-2-9 タグデータリンクパラメータのダウンロード 7-58
7-2-10 タグデータリンクパラメータのアップロード 7-61
7-2-11 タグデータリンクパラメータの照合 7-64
7-2-12 タグデータリンクの開始／停止 7-68
7-2-13 デバイスパラメータのクリア 7-70
7-2-14 ネットワーク構成ファイルの保存 7-71
7-2-15 ネットワーク構成ファイルの読出 7-73
7-2-16 コネクションのチェック 7-75
7-2-17 デバイスの変更 7-76
7-2-18 デバイスステータス表示 7-79
7-3 タグデータリンクのラダープログラム 7-80

7-3-1 タグデータリンクでのラダープログラム 7-80
7-3-2 タグデータリンクに関連するステータスについて 7-84
7-4 NJ CPU 以外の機種とタグデータリンクを行う場合について 7-85

第 8 章 メッセージ通信機能

8-1 CIP メッセージ通信機能の概要 8-2
8-1-1 CIP メッセージ通信サービス機能の概要 8-2
8-1-2 メッセージ通信サービスの機能・性能一覧 8-2
8-2 CIP メッセージ通信の使い方 8-3
8-2-1 概要 8-3
8-2-2 CIP 通信命令の種類 8-4
8-2-3 CIP 通信命令の使用方法 8-5
8-2-4 ルートパス 8-6
8-2-5 CIP 通信命令使用時の派生データ型の用意 8-12
8-2-6 サンプルプログラム（CIP コネクションレス型 (UCMM) メッセージで通信する場合） 8-14
8-2-7 サンプルプログラム（CIP コネクション型 (Class3) メッセージで通信する場合） 8-21
8-2-8 動作タイミング 8-28
8-2-9 レスポンスコード 8-29
8-3 EtherNet/IP ユニット宛 CIP オブジェクトについて 8-33
8-3-1 EtherNet/IP ユニット宛 CIP オブジェクトの種類 8-33
8-3-2 Identity オブジェクト（クラス ID：01Hex） 8-33
8-3-3 TCP/IP Interface オブジェクト（クラス ID：F5Hex) 8-36
8-3-4 Ethernet Link オブジェクト（クラス ID：F6 Hex) 8-38
8-3-5 PLC オブジェクト（クラス ID：C4Hex) 8-41

第 9 章 FTP サーバ機能

9-1 FTP サーバ機能の概要 9-2
9-1-1 概要 9-2
9-1-2 仕様 9-2
9-2 FTP サーバ機能の詳細 9-3
9-2-1 対象ファイル 9-3
9-2-2 FTP サーバへの接続方法 9-3
9-3 FTP サーバ機能の使用手順 9-5
9-3-1 使用手順 9-5
9-3-2 FTP サーバ機能に必要な設定一覧 9-5
9-4 FTP サーバ機能の使用例 9-6
9-5 FTP コマンドの使用方法 9-7
9-5-1 コマンド一覧 9-7
9-5-2 コマンドの使用方法 9-8
9-6 FTP サーバ機能ステータス 9-13
9-6-1 FTP ステータス 9-13
9-7 SD メモリカード機能の使い方 9-14
9-7-1 SD メモリカードの種類 9-14
9-7-2 ファイルの種類 9-15
9-7-3 SD メモリカードの初期化 9-15
9-7-4 変数データのフォーマット 9-16
9-8 FTP によるファイル転送時間 9-17
9-9 ホストコンピュータからの使用例 9-18

第 10 章 時計情報自動調整機能

10-1 時計情報自動調整機能の概要 10-2

10-1-1　時計情報自動調整機能とは 10-2
10-1-2　仕様 10-2
10-2 時計情報自動調整機能の使用手順 10-3
10-2-1　使用手順 10-3
10-2-2　時計情報自動調整機能に必要な設定一覧 10-3
10-2-3　時計情報の更新 10-4

第 11 章 SNMP（エージェント）機能

11-1 SNMP（エージェント）機能の概要 11-2
11-1-1　SNMP（エージェント）機能とは 11-2
11-1-2　仕様 11-3
11-1-3　SNMP メッセージ 11-3
11-1-4　MIB の仕様 11-4
11-2 SNMP（エージェント）機能の使用手順 11-17
11-2-1　使用手順 11-17
11-2-2　SNMP（エージェント）機能に必要な設定一覧 11-17

第 12 章 セキュリティ機能

12-1 セキュリティ機能の概要 12-2
12-1-1　セキュリティ機能の一覧 12-2
12-2 ポートのオープン／クローズ機能 12-3
12-2-1　機能の概要 12-3
12-2-2　機能の詳細 12-3
12-3 Packet Filter 機能 12-7
12-3-1　機能の概要 12-7
12-3-2　機能の詳細 12-7
12-4 セキュリティ機能の一般的なユースケース 12-11
12-4-1　ユースケース 12-11
12-4-2　ケース 1：特定のプロトコルからの受信を許可する 12-11
12-4-3　ケース 2：特定の送信元 IP アドレスからの受信を許可する 12-12
12-5 セキュリティ脅威からの保護対策 12-14

第 13 章 ソケットサービス機能

13-1 ソケット通信の基礎知識 13-3
13-1-1　ソケットとは 13-3
13-1-2　ソケットサービスのポート No. 13-3
13-2 プロトコルの基礎知識 13-4
13-2-1　TCP と UDP の相違点 13-4
13-2-2　TCP ソケットのオープン処理 13-5
13-2-3　送信データのフラグメント（分割）について 13-6
13-3 EtherNet/IP ユニット形 CJ1W-EIP21S のソケットサービス概要 13-9
13-3-1　概要 13-9
13-3-2　デバイス変数操作によるソケットサービスの概要 13-9
13-3-3　SendCmd によるソケットサービスの概要 13-11
13-3-4　EtherNet/IP ユニットのソケットサービス種類 13-11
13-4 ソケットサービス機能に必要な設定 13-12
13-5 ソケットサービス機能のステータス 13-13
13-5-1　UDP/TCP ソケット No.1 ～ No.8 ステータス（ユニット→ CPU ユニット） 13-13
13-5-2　TCP 受信データバイト数（ユニット→ CPU ユニット） 13-13
13-5-3　TCP コネクション状態（ユニット→ CPU ユニット） 13-13
13-6 デバイス変数操作によるソケットサービスの利用の詳細 13-15

13-6-1 使用手順 13-15
13-6-2 ソケットサービスとソケットステータスの使用手順 13-16
13-6-3 ソケットサービスパラメータエリア（CPU ユニット⇔ユニット） 13-17
13-6-4 パラメータ詳細 13-18
13-6-5 ソケットサービス要求スイッチ（CPU ユニット⇔ユニット） 13-21
13-6-6 終了コード 13-22
13-6-7 タイミングチャート 13-26
13-6-8 サンプルプログラム 13-27

13-7 SendCmd 命令によるソケットサービスの利用の詳細 13-28
13-7-1 ソケットサービスの使用方法 13-28
13-7-2 ソケットサービス要求 13-29
13-7-3 ソケットサービスとソケットステータスの使用手順 13-46
13-7-4 終了コード 13-47
13-7-5 通信時のタイミングチャート 13-47
13-7-6 ソケットサービスのタイミングチャート 13-48
13-7-7 サンプルプログラム 13-50

13-8 ソケットサービス使用上の留意事項 13-51
13-8-1 UDP/TCP ソケットサービス共通の留意事項 13-51
13-8-2 UDP ソケットサービスのみの留意事項 13-51
13-8-3 TCP ソケットサービスのみの留意事項 13-51
13-8-4 デバイス変数操作に関する留意事項 13-52
13-8-5 ソケットサービスの送信／受信に要する時間（ネットワーク上での遅延を無視した場合） 13-53

第 14 章 通信性能と通信負荷

14-1 通信のしくみ 14-2
14-1-1 タグデータリンクの通信のしかた 14-2
14-1-2 コネクション数の計算方法 14-5
14-1-3 パケットインターバル（RPI）の精度 14-6

14-2 通信負荷の調整 14-7
14-2-1 タグデータリンクのデバイス帯域使用状況の確認 14-8
14-2-2 タグデータリンクの帯域使用率と RPI の関係 14-10
14-2-3 デバイス帯域使用状況の調整方法 14-11
14-2-4 RPI の変更方法 14-12
14-2-5 パケットインターバル（RPI）の設定例 14-17

14-3 タグデータリンクによる入出力応答時間 14-22
14-3-1 データ転送のタイミング 14-22
14-3-2 EtherNet/IP ユニットのデータ処理時間 14-23
14-3-3 タグデータリンクによる、タスク周期への影響 14-26
14-3-4 タグデータリンク入出力最大応答時間 14-26

14-4 メッセージサービス機能伝送遅延時間 14-28

第 15 章 異常とその処理

15-1 Network Configurator によるステータスの確認方法 15-2
15-1-1 Network Configurator のデバイスモニタ機能 15-2

15-2 LED 表示の意味と異常時の処置 15-10
15-2-1 EtherNet/IP ユニットで発生する異常と処置 15-10

15-3 コネクションステータスのコード一覧と処置方法 15-21

15-4 異常履歴機能 15-28
15-4-1 異常履歴データ 15-28
15-4-2 異常履歴コード一覧 15-29

15-5 イベントログ機能の概要 15-33
15-5-1 イベントログ機能の概要 15-33
15-5-2 異常一覧 15-34
15-5-3 異常の内容 15-37

15-6 トラブルシューティング 15-48
15-6-1 EtherNet/IP ユニットの装着で CPU ユニットの ERROR が点灯／点滅する場合 15-48
15-6-2 Ethernet の一般的なトラブル 15-48
15-6-3 タグデータリンクがスタートしない場合 * 15-49
15-6-4 タグデータリンクのトラブル 15-50
15-6-5 メッセージタイムアウトのトラブル 15-51

15-7 清掃と点検 15-52
15-7-1 清掃方法 15-52
15-7-2 点検方法 15-52

15-8 装置交換時の留意点 15-54
15-8-1 ユニット交換時のお願い 15-54
15-8-2 ユニット交換後の設定 15-54
15-8-3 ユニットの交換方法 15-55

付録

A-1 NJ シリーズ CPU ユニットと他シリーズとの EtherNet/IP 機能の比較 A-3

A-2 Sysmac Studio でタグデータリンク（EtherNet/IP コネクション）を設定する A-4
A-2-1 Sysmac Studio によるタグデータリンク（EtherNet/IP コネクション）設定の概要 A-4
A-2-2 Sysmac Studio による EtherNet/IP コネクション設定の手順 A-5
A-2-3 EtherNet/IP コネクション設定機能について A-6
A-2-4 Sysmac Studio による EtherNet/IP コネクション設定 A-11
A-2-5 ネットワークの通信ステータス確認と処置方法 A-32
A-2-6 トラブルシューティング A-36

A-3 EDS ファイルの管理 A-41
A-3-1 EDS ファイルのインストール A-42
A-3-2 EDS ファイルの作成 A-42
A-3-3 EDS ファイルの削除 A-43
A-3-4 EDS ファイルの保存 A-43
A-3-5 EDS ファイルの検索 A-44
A-3-6 EDS ファイルのプロパティ表示 A-45
A-3-7 EDS インデックスファイルの再作成 A-45

A-4 Network Configurator を、WindowsXP/ WindowsVista/Windows7 で使用する際の留意点 A-46
A-4-1 Windows ファイアウォールの設定の変更手順 A-46

A-5 変数のメモリ確保の方法について A-49
A-5-1 変数のメモリ確保の規則 A-49
A-5-2 留意するべきケース例 A-58

A-6 CPU ユニットの外部出力にアクセスする場合の留意事項 A-61

A-7 接続する CPU ユニット（NJ/CJ シリーズ）による相違点 A-62
A-7-1 機能の相違点 A-62
A-7-2 ユーザプログラムからのアクセス方法に関する相違点 A-62

A-8 形 CJ1W-EIP21 から形 CJ1W-EIP21S を使用したシステムへの置き換え A-101
A-8-1 仕様や機能などの差異 A-101
A-8-2 置き換えフロー A-105
A-8-3 事前準備 A-107
A-8-4 置き換えの実行 A-114

A-9 ソケットサービス機能のサンプルプログラム A-117
A-9-1 システム構成 A-117
A-9-2 サンプルプログラムで必要な設定 A-118
A-9-3 デバイス変数操作による UDP 通信のサンプルプログラム A-120
A-9-4 デバイス変数操作による TCP 通信のサンプルプログラム A-124
A-9-5 SendCmd 命令による UDP 通信のサンプルプログラム A-129
A-9-6 SendCmd 命令による TCP 通信のサンプルプログラム A-137

A-10 Generic デバイスを用いたタグデータリンク設定（Network Configurator Ver.3.74 以降） A-145
A-10-1 Generic デバイスの作成手順 A-145

A-10-2　Generic デバイスのタグ／タグセットの作成.....A-146
A-11 バージョン関連情報 A-150

索引

ご承諾事項

「当社商品」について特別の合意がない場合には、お客様のご購入先にかかわらず、本ご承諾事項記載の条件を適用いたします。

● 定義

本ご承諾事項中の用語の定義は次のとおりです。

- 「当社商品」:「当社」の FA システム機器、汎用制御機器、センシング機器、電子・機構部品
- 「カタログ等」:「当社商品」に関する、ベスト制御機器オムロン、電子・機構部品総合カタログ、その他のカタログ、仕様書、取扱説明書、マニュアル等であって電磁的方法で提供されるものも含みます。
- 「利用条件等」:「カタログ等」に記載の、「当社商品」の利用条件、定格、性能、動作環境、取り扱い方法、利用上の注意、禁止事項その他
- 「お客様用途」:「当社商品」のお客様におけるご利用方法であって、お客様が製造する部品、電子基板、機器、設備またはシステム等への「当社商品」の組み込み、または利用を含みます。
- 「適合性等」:「お客様用途」での「当社商品」の (a) 適合性、(b) 動作、(c) 第三者の知的財産の非侵害、(d) 法令の遵守および (e) 各種規格の遵守

● 記載事項のご注意

「カタログ等」の記載内容については次の点をご理解ください。

- 定格値および性能値は、単独試験における各条件のもとで得られた値であり、各定格値および性能値の複合条件のもとで得られる値を保証するものではありません。
- 参考データはご参考として提供するもので、その範囲で常に正常に動作することを保証するものではありません。
- 利用事例はご参考ですので、「当社」は「適合性等」について保証いたしかねます。
- 「当社」は、改善や当社都合等により、「当社商品」の生産を中止し、または「当社商品」の仕様を変更することがあります。

● ご利用にあたってのご注意

ご採用およびご利用に際しては次の点をご理解ください。

- 定格・性能ほか「利用条件等」を遵守しご利用ください。
- お客様ご自身にて「適合性等」をご確認いただき、「当社商品」のご利用の可否をご判断ください。「当社」は「適合性等」を一切保証いたしかねます。
- 「当社商品」がお客様のシステム全体の中で意図した用途に対して、適切に配電・設置されていることをお客様ご自身で、必ず事前に確認してください。
- 「当社商品」をご使用の際には、(i) 定格および性能に対し余裕のある「当社商品」のご利用、冗長設計などの安全設計、(ii)「当社商品」が故障しても、「お客様用途」の危険を最小にする安全設計、(iii) 利用者に危険を知らせるための、安全対策のシステム全体としての構築、(iv)「当社商品」および「お客様用途」の定期的な保守、の各事項を実施してください。
- 「当社」は DDoS 攻撃（分散型 DoS 攻撃）、コンピュータウイルスその他の技術的な有害プログラム、不正アクセスにより、「当社商品」、インストールされたソフトウェア、またはすべてのコンピュータ機器、コンピュータプログラム、ネットワーク、データベースが感染したとしても、そのことにより直接または間接的に生じた損失、損害その他の費用について一切責任を負わないものとします。

お客様ご自身にて、①アンチウイルス保護、②データ入出力、③紛失データの復元、④「当社商品」またはインストールされたソフトウェアに対するコンピュータウイルス感染防止、⑤「当社商品」に対する不正アクセス防止についての十分な措置を講じてください。

- 「当社商品」は、一般工業製品向けの汎用品として設計製造されています。従いまして、次に掲げる用途での使用は意図しておらず、お客様が「当社商品」をこれらの用途に使用される際には、「当社」は「当社商品」に対して一切保証をいたしません。ただし、次に掲げる用途であっても「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は除きます。
  （a）高い安全性が必要とされる用途（例：原子力制御設備、燃焼設備、航空・宇宙設備、鉄道設備、昇降設備、娯楽設備、医用機器、安全装置、その他生命・身体に危険が及びうる用途）
  （b）高い信頼性が必要な用途（例：ガス・水道・電気等の供給システム、24 時間連続運転システム、決済システムほか権利・財産を取扱う用途など）
  （c）厳しい条件または環境での用途（例：屋外に設置する設備、化学的汚染を被る設備、電磁的妨害を被る設備、振動・衝撃を受ける設備など）
  （d）「カタログ等」に記載のない条件や環境での用途
- 上記の（a）から（d）に記載されている他、「本カタログ等記載の商品」は自動車（二輪車含む。以下同じ）向けではありません。自動車に搭載する用途には利用しないで下さい。自動車搭載用商品については当社営業担当者にご相談ください。

● 保証条件

「当社商品」の保証条件は次のとおりです。

- 保証期間　ご購入後 1 年間といたします。
  （ただし「カタログ等」に別途記載がある場合を除きます。）
- 保証内容　故障した「当社商品」について、以下のいずれかを「当社」の任意の判断で実施します。
  （a）当社保守サービス拠点における故障した「当社商品」の無償修理
  （ただし、電子・機構部品については、修理対応は行いません。）
  （b）故障した「当社商品」と同数の代替品の無償提供
- 保証対象外　故障の原因が次のいずれかに該当する場合は、保証いたしません。
  （a）「当社商品」本来の使い方以外のご利用
  （b）「利用条件等」から外れたご利用
  （c）本ご承諾事項「ご利用にあたってのご注意」に反するご利用
  （d）「当社」以外による改造、修理による場合
  （e）「当社」以外の者によるソフトウェアプログラムによる場合
  （f）「当社」からの出荷時の科学・技術の水準では予見できなかった原因
  （g）上記のほか「当社」または「当社商品」以外の原因（天災等の不可抗力を含む）

● 責任の制限

本ご承諾事項に記載の保証が、「当社商品」に関する保証のすべてです。
「当社商品」に関連して生じた損害について、「当社」および「当社商品」の販売店は責任を負いません。

● 輸出管理

「当社商品」または技術資料を、輸出または非居住者に提供する場合は、安全保障貿易管理に関する日本および関係各国の法令・規制を遵守ください。お客様が法令・規則に違反する場合には、「当社商品」または技術資料をご提供できない場合があります。

安全上の注意

安全に使用していただくための表示と意味

本マニュアルでは、CJ シリーズ EtherNet/IP ユニットを安全に使用していただくために、注意事項を次のような表示と図記号で示しています。ここで示した注意事項は、安全に関する重大な内容を記載しています。必ず守ってください。
表示と意味は次のとおりです。

⚠ 警告	正しい取扱をしなければ、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	正しい取扱をしなければ、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。

図記号の説明



⃠記号は、禁止を意味しています。
具体的な内容は、⃠の中と文章で示します。
左図の場合は、「分解禁止」を表します。



△記号は、注意（警告を含む）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、「感電注意」を表します。



△記号は、注意（警告を含む）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、「一般的な注意」を表します。



●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、「一般的な強制事項」を表します。



△記号は、注意（警告を含む）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、「高温注意」を表します。

警告

⚠ 警 告

通電中

通電中は、端子部に触れないでください。感電の恐れがあります。



本製品を分解しないでください。特に通電中や通電 OFF 直後は、電源ユニット内部には昇圧による電圧の高い部分があり、感電の恐れがあります。また、内部の鋭利な部品でけがをする可能性があります。



フェールセーフ対策

CPU ユニットおよびスレーブやユニットの故障や外部要因による異常が発生した場合でも、システム全体が安全側に働くように、外部で安全対策を施してください。異常動作により、重大な事故につながる恐れがあります。



非常停止回路、インタロック回路、リミット回路など、安全保護に関する回路は、必ず外部の制御回路で構成してください。



出力リレーの溶着や焼損、出力トランジスタの破壊などによって、出力が ON または OFF になったままになることがあります。このとき、システムが安全側に動作するように、外部で対策を施してください。



CPU ユニットは、以下のときに、基本出力ユニットの全出力を OFF にし、スレーブはスレーブ側の動作に従います。
- 電源部異常が発生したとき
- 電源接続不正が発生したとき
- CPU 異常（WDT 異常）または CPU リセットが発生したとき
- 全停止フォールトレベルのコントローラ異常が発生したとき
- 電源投入から運転モードに移行するまでの起動中

これらのとき、システムが安全側に動作するよう、外部で対策を施してください。



スレーブなどの外部電源が過負荷状態になる、または短絡状態になると電圧が低下し、出力が OFF になったり、入力が取り込めなくなったりする場合があります。このときシステムが安全側に動作するよう、必要によっては外部電源電圧を監視し、制御に取り込むなど外部で対策を施してください。



変数メモリ、CJ ユニット用メモリなどの異常時は、意図しない出力をすることがあります。このとき、システムが安全側に動作するよう、外部で対策を施してください。



データリンク通信やリモート I/O 通信において、通信異常や誤動作が発生しても、システム全体が安全側に動作するよう通信システムやユーザプログラムで対策を施してください。



リモート I/O において通信障害が発生した場合や全停止フォールトレベルの異常が発生した場合に、出力状態は使用される製品の仕様によります。
上記通信障害または全停止フォールトレベルの異常発生時にどのような動作仕様であるかを確認の上、安全対策を施してください。
スレーブやユニットでは各々の設定を正しく行ってください。



NJ シリーズは瞬時停電発生時でも一定の時間は正常動作を続けるため、瞬時停電により影響を受けた外部機器から誤った信号を受け付けてしまう可能性があります。
外部にてフェールセーフ対策を行い、かつ必要によっては外部機器側の電源電圧自体を監視し、インタロック条件として取り込むなどの対策を行ってください。



信号線の断線、瞬時停電による異常信号などに備えて、ご使用者側でフェールセーフ対策を施してください。
異常動作により重大な事故につながる恐れがあります。



電圧／電流入力

スレーブやユニットに入力する電圧／電流は定められた範囲で入力してください。
範囲外の電圧／電流を使用すると故障や火災の原因となります。



転送

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。



本運転時

作成したユーザプログラムおよび各種データ・設定値は、十分な動作確認を行った後、本運転に移行してください。



注意

⚠ 注 意

使用時

通電中や電源を切った直後は、装置に触れないでください。火傷の恐れがあります。



配線時

端子台のねじ、ケーブルのねじは、本マニュアルまたは参照マニュアルで指定したトルクで締め付けてください。ねじが緩むと発火および誤動作の恐れがあります。



オンラインエディット

入出力タイミングが乱れても影響のないことを確認してからオンラインエディットをしてください。オンラインエディットすると、タスク実行時間がタスク周期を上回り、外部との入出力が更新されず、入力信号を読み取れなかったり、出力タイミングが乱れたりする場合があります。



エラーメッセージ「命令が意図しない動作を起こす可能性があります」が表示された場合

Sysmac Studio のオプション項目「プログラムチェック」の「入出力変数を特定の命令の特定の引数に渡す場合にエラーにする」のチェックを外した場合、命令が意図しない動作を起こし、装置にも影響を与える可能性があります。
🕮『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)』に記載された、使用できる条件に合致していることを必ずご確認ください。



バージョン関連情報

上記のエラーメッセージの表示、およびオプション設定ができるのは、Sysmac Studio Ver.1.02 だけです。

安全上の要点

分解・落下

- 本製品を、分解して修理や改造はしないでください。故障や発火の恐れがあります。
- 製品を落下させたり、異常な振動・衝撃を与えたりしないでください。製品の故障、焼損の可能性があります。

装着時

- 電源ユニットや CPU ユニット、I/O ユニットなど、各ユニットを接続するときは、ユニット間コネクタを結合後、上下のスライダをカチッと音がするまでスライドさせて、確実にロックしてください。

設置時

- 据え付け工事の際には、必ず D 種接地（第 3 種接地）をしてください。特に電源ユニットの LG 端子と GR 端子を短絡した場合には、感電防止のため必ず D 種接地（第 3 種接地）をしてください。

配線時

- 本マニュアルに示すとおり正しく配線してください。
  配線、スイッチなどの設定を十分確認してから通電してください。
- 配線は、圧着端子を付けてください。
  より合わせただけの電線を直接、端子台に接続しないでください。
- ケーブルを無理に曲げたり、引っ張ったりしないでください。
  ケーブルのコード部に重いものを載せないでください。断線する恐れがあります。
- 端子台、コネクタは装着位置を十分確認してから、装着してください。
- 端子台、増設ケーブルなどロック機構のあるものは、必ずロックしていることを確認してからご使用ください。
- 出荷時からユニット上部に防塵ラベルが貼ってある場合は、通電前に必ずラベルを外してください。放熱できず、誤動作の可能性があります。
- パソコンと CPU ユニットを接続する際には、パソコンの電源プラグを AC コンセントから抜いておいてください。また、パソコンに FG 端子がある場合は、電源ユニットの GR 端子と同電位になるように接続してください。
  パソコンとコントローラとの間に電位差が発生し、故障や誤動作の原因になる可能性があります。
- 出力ユニットやスレーブで極性のある外部電源に逆電圧を加えないでください。コントローラの動作に関係なく逆電流が流れ、接続された機器を壊す恐れがあります。

電源設計時

- システム内で使用する電源ユニットは、🕮『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)』に記載されている電源ユニットの定格供給容量を超えない範囲で使用してください。
  運転停止や誤動作が発生したり、電源断時に必要なデータが正しくバックアップされなかったりする可能性があります。
  NJ シリーズ CPU ラックと増設ラックも含めて、NJ シリーズ電源ユニットを使用してください。

CJ シリーズ電源を使用したり、NJ シリーズ電源を CJ シリーズユニットと組み合わせたりすると動作しません。
- 出力ユニットやスレーブは、最大定格を超える電圧の印加および負荷接続をしないでください。
- 電源投入時には突入電流が発生します。外部回路のヒューズやブレーカを選定される際は、溶断・検知特性や上記内容をご考慮の上、余裕を持った設計を行ってください。突入電流仕様については、🕮『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）』を参照してください。
- 耐電圧試験時にスイッチで全電圧を印加または遮断すると、発生するインパルス電圧により、電源ユニットが破損することがあります。印加電圧は試験機のボリュームで徐々に変化させてください。
- 絶縁抵抗および耐電圧試験時は、電源ユニットの L1 端子、L2 端子と GR 端子間に電圧を印加してください。
- AC 電源にインバータなどからの矩形波出力を給電しないでください。内部温度上昇により発煙・焼損の恐れがあります。必ず、🕮『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）』に示す周波数の正弦波を入力してください。
- 外部配線の短絡に備えて、ブレーカなどの安全対策を施してください。

電源投入時

- 電源投入から運転モードに移行するまで 10 ～ 20 秒程度時間がかかります。この間の出力はスレーブやユニットの仕様に従います。電源ユニットの運転中出力を利用するなどして外部機器が誤動作しないようにフェールセーフ回路を組んでください。
- コントローラの電源を ON にした後に、制御部の電源を ON にするように、外部回路を構成してください。制御部の電源を ON した後に、コントローラの電源を ON にすると、コントローラの電源 ON 時に、出力ユニットの出力が一瞬誤動作し、制御部の出力が一時的に正常動作しない可能性があります。
- 電源投入時に SD メモリカードのバックアップファイルのデータをコントローラに転送するときは、転送するデータグループを適切に選択してください。意図しないデータグループのデータをコントローラに転送すると設備が予期しない動作をする可能性があります。
- 電源投入から運転モードに移行するまでの間は、コントローラの電源を OFF にしないでください。電源 OFF した場合、次回起動時に、「バッテリバックアップメモリチェックエラー」が発生する場合があります。「バッテリバックアップメモリチェックエラー」発生時は、保持変数は初期値になり、CJ ユニット用メモリの HR ／ DM ／ EM はゼロクリアとなります。運転を再開する場合は、必要により、保持変数や CJ ユニット用メモリへ、正しいデータを再設定してください。

電源 OFF 時

- BUSY LED が点滅しているときに、コントローラの電源を OFF にしないでください。BUSY LED 点滅中は、CPU ユニット内のユーザプログラム、および設定値を内蔵不揮発性メモリにバックアップしています。電源を OFF にすると、これらが正常にバックアップされません。また、次回起動時に全停止フォールトレベルのコントローラ異常となり、運転は停止します。
- SD メモリカードアクセス中（SD BUSY LED が点滅）に電源を OFF したり、SD メモリカードを抜いたりしないでください。データが壊れる可能性があり、そのデータを用いた場合に正しく動作しません。通電中の CPU ユニットから SD メモリカードを取り出す場合は、いったん SD メモリカード給電停止ボタンを押し、SD BUSY LED が消えた後に、SD メモリカードを抜いてください。
- ツールから各種データ、ユーザプログラムを転送中はケーブルを抜いたり、コントローラの電源を OFF したりしないでください。

- 次のことを行うときは、コントローラの電源を OFF にしてください。
  I/O ユニットや CPU ユニットを着脱するとき
  装置を組み立てるとき
  ディップスイッチやロータリスイッチを設定するとき
  ケーブルを接続、配線するとき
  コネクタを取り付けたり、取り外したりするとき
- 電源切断後、電源ユニットは数秒間コントローラに対して給電を継続する場合があり、この間 PWR LED が点灯します。上記の操作は、PWR LED の消灯を確認してから行ってください。

操作時

- 次の操作は設備に影響がないかを確認したうえで行ってください。
  CPU ユニットの動作モードの変更（電源投入時の動作モード設定含む）
  ユーザプログラム、設定の変更
  設定値 / 現在値の変更
  強制値リフレッシュ
- スレーブやユニットの設定を変更してリスタートするときは、接続先の対象機器の安全を十分に確認してから行ってください。
- CJ シリーズ基本ユニットと EtherCAT スレーブのように異なる機能モジュールを連携させて装置を制御する場合は、片方の機能モジュールが停止しても装置に影響を与えないように、ユーザプログラムおよび外部にて対策を行ってください。それぞれの機能モジュールに部分停止フォールトレベルの異常が発生した場合、該当する出力はスレーブやユニットの仕様に従います。
- EtherCAT マスタ機能モジュールに対して、部分停止フォールトレベル以上のコントローラ異常の解除をするときは、接続先の設備の安全を十分に確認してください。
  部分停止フォールトレベル以上のコントローラ異常によりオペレーショナル状態以外（出力無効）に遷移しているスレーブはオペレーショナル状態に遷移し、出力が有効になります。
  異常の全解除やスレーブのリスタートを行うときは、EtherCAT マスタ機能モジュールの部分停止フォールトレベルのコントローラ異常が発生していないか確認してください。
- CJ シリーズ高機能ユニットに対してコントローラ異常の解除をするときは、接続先の設備の安全を十分に確認してください。監視情報レベル以上のコントローラ異常が発生している CJ シリーズ高機能ユニットに対して、コントローラ異常の解除をすると、そのユニットはリスタートされます。
  異常の全解除を行うときは、CJ シリーズ高機能ユニットの監視情報レベル以上のコントローラ異常が発生していないか確認してください。「コントローラ異常」タブ画面では監視情報レベルのイベントが現れないため、意図せず CJ シリーズ高機能ユニットのリスタートを行ってしまう可能性があります。
  監視情報レベルのコントローラ異常の有無は、異常ステータス変数 _CJB_UnitErrSta[0,0] ～ _CJB_UnitErrSta[3,9] をウォッチウインドウでモニタすると確認できます。

バッテリ保持

- ユーザプログラムおよび変数の初期値は、CPU ユニット内部の不揮発性メモリに保存されていますが、保持属性の変数の現在値、および CJ ユニット用メモリの HR/DM/EM の値はバッテリによってデータを保持されています。
  バッテリが装着されていない状態、またはバッテリが消耗している状態になると、CPU ユニットがバッテリバックアップメモリチェックエラーを検出します。
  その場合は、保持属性を持つ変数は初期値になり、また CJ ユニット用メモリの HR/DM/EM はゼロクリアとなります。
  この場合でも、装置が安全に動作するように保持属性を持つ変数の初期値の設定および初期値での動作に関しては十分に検証を行ってください。

- 絶対値エンコーダ原点位置オフセットはバッテリで保持しています。電源投入時に CPU ユニットがバッテリ電圧低下 / バッテリ非実装を検知すると、絶対値エンコーダ原点位置オフセットはゼロクリアされ、「絶対値エンコーダ原点位置オフセット読み出し異常」を発生させます。異常解除を行い、原点復帰を実行し、原点を確定させてください。原点確定をしなかった場合、制御対象が予期しない動作をする恐れがあります。

デバッグ時

- 強制値リフレッシュはユーザプログラム実行結果に関わらず、設定された値で I/O リフレッシュする機能です。そのため、I/O リフレッシュの対象でない接点は本機能によっていったん設定された値になりますが、その後ユーザプログラムによって書き変わります。CJ シリーズの強制セット／リセット機能とは動作が異なりますので、ご注意ください。
- 強制値リフレッシュ情報は Sysmac Studio によるアップロード／ダウンロードはできません。
  したがって、強制値リフレッシュ対象が含まれているデータをダウンロードした後、必要に応じて、運転モードに変更後、Sysmac Studio から強制値リフレッシュ操作を行ってください。
  強制状態の相違によって、制御対象が予期しない動作をする恐れがあります。
- 複数の変数を、同じアドレスに AT（割付先）指定しないでください。
  ひとつの実体に複数の変数名でアクセスするため、ユーザプログラムがたいへん見にくくなり、プログラムミスの原因にもなります。

通信一般

- データリンク通信においては、_ErrSta（コントローラ異常ステータス）を参照して、データリンク元の装置に異常が発生していない場合のみ、受信データを使用するユーザプログラムを作成してください。
  異常発生時には、データリンクで受信したデータが不正な値になっている可能性があります。
- 不適切なデータリンクテーブルが設定されていると、設備が予期しない動作をする恐れがあります。また適切なデータリンクテーブルが設定されている場合でも、設備に影響がないことを確認してから、データリンクテーブルを転送してください。転送後に自動的にデータリンクを開始します。
- ルーチングテーブルを周辺ツールから CPU ユニットに転送すると、すべての CPU 高機能ユニットにリスタートがかかります。リスタートがかかっても設備に影響がないことを確認してから、転送を実行してください。
- 運転中にタグデータリンクパラメータを転送することは可能ですが、タグデータリンクパラメータの転送中は、関連するノード間でのタグデータリンクが停止します。設備に影響がないことを確認してから、転送を実行してください。

EtherNet/IP 通信

- 形 CJ1W-EIP21 ユニットの設定（IP アドレスやタグデータリンクの設定）を周辺ツールから転送すると、設定対象すべての EtherNet/IP（ノード）にリセットがかかります。リセットがかかっても設備に影響がないことを確認してから、転送を実行してください。
- リピータハブを使用して EtherNet/IP のタグデータリンク通信（サイクリック通信）を行うと、ネットワークの通信負荷が高まるため、コリジョン（衝突）が多数発生し、安定した通信ができなくなります。タグデータリンクを利用するネットワークでは、必ずスイッチングハブを使用してください。

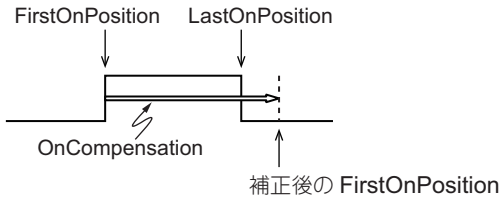
EtherCAT 通信

- EtherCAT は通信距離および接続台数／接続方法を仕様の範囲内でご使用ください。
  また、EtherNet/IP や一般的な構内 LAN など他のネットワークに接続しないでください。過負荷によるネットワークのダウンや誤動作の可能性があります。
- EtherCAT のマスタとスレーブのリビジョンの組み合わせによっては誤動作や意図しない動きをする可能性があります。ネットワーク設定のリビジョンチェックの機能を無効にして使用する場合は、Sysmac Studio を使ってマスタの設定上のスレーブリビジョンと実際のスレーブのリビジョンを確認し、スレーブのマニュアル等より機能の互換性を確認した上でご使用ください。なお、実際のスレーブのリビジョンは Sysmac Studio や銘板で確認することができます。
- ユーザプログラムを転送すると、CPU ユニットにリスタートが発生し、EtherCAT スレーブとの通信が途絶えます。その間のスレーブ出力は、スレーブの仕様に従います。EtherCAT ネットワーク構成により、通信が途絶える時間が異なります。
  ユーザプログラム転送時は装置に影響を与えないことを確認してから行ってください。
- 「縮退運転設定」を「停止する」と設定している場合、EtherCAT 通信異常がいずれかのスレーブで発生すれば、全スレーブとプロセスデータ通信を停止します。このときのサーボドライバの動作は、サーボドライバの仕様に従います。「縮退運転設定」については、装置の異常発生時に安全な挙動をとるように十分考慮してください。
- 電源投入直後、EtherCAT の通信が確立していない場合があります。システム定義変数を用いて通信が確立したことを確認してから、制御に使用するようにプログラミングしてください。
- ノイズが発生したときや、EtherCAT スレーブからネットワークケーブルを抜いたときに、通信中のフレームが消失することがあります。フレームが消失すると、スレーブの入出力データが伝わらなくなり、意図しない動作をする恐れがあります。スレーブの出力はスレーブの仕様に従います。詳細は、各スレーブのマニュアルを参照してください。ノイズ対策やスレーブの交換が必要な場合は、以下の処置を行ってください。
  - システム定義変数の「_EC_InDataInvalid」（入力データ無効）をインタロック条件としてユーザプログラム作成してください。
  - EtherCAT マスタの設定項目の「PDO 通信連続タイムアウト検出回数」を、必ず 2 回以上に設定してください。詳細は、🕮『NJ/NX シリーズ CPU ユニット内蔵 EtherCAT ポート ユーザーズマニュアル（SBCD-376）』を参照してください。
- EtherCAT スレーブを「離脱」や「無効」とした場合、該当スレーブ以降のスレーブとの通信も停止し、出力が制御できなくなります。設備に影響が無いことを確認してから「離脱」、「無効」の指示をしてください。
- 電源投入直後、NX バスの通信が確立していない場合があります。システム定義変数、および、EtherCAT カプラユニットのデバイス変数を用いて通信が確立したことを確認してから、制御に使用するようにプログラミングしてください。
- 汎用 Ethernet ハブ、リピータハブは使用しないでください。使用した場合、全停止フォールトレベルなどの異常が発生する恐れがあります。

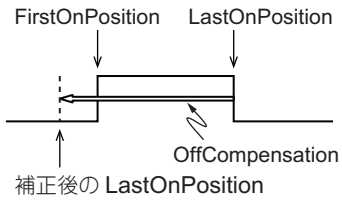
モーション制御

- MC 試運転を行う場合、軸番号をよく確認してから行ってください。
- MC 試運転中に Sysmac Studio-CPU ユニット間の通信が途絶えた場合は、モータを強制的に停止させます。PC と CPU ユニットをつなぐ通信ケーブルの抜けには注意し、設備への影響がないことを確認してから MC 試運転を実行してください。
- CPU ユニットのユーザプログラムや Sysmac Studio でカムデータの値を変更した場合、必ずカムテーブル保存命令を実行してください。行わない場合は、次の電源投入時にカムデータの値が変更前の状態に戻ってしまい、装置が意図しない動作をする可能性があります。

- サーボドライバの外部入力に割り付けられた正転側駆動禁止入力（POT）、逆転側駆動禁止入力（NOT）、原点近傍入力（DEC）をそれぞれ正転側限界入力、逆転側限界入力、原点近傍入力として使用します。これらの入力信号に関しては、必ず制御周期以上の信号幅を確保するようにしてください。入力信号が制御周期未満の幅となる場合、入力信号を検出できず、正常な動作ができなくなります。
- Sysmac Studio の [軸基本設定] 画面の「詳細設定」を変更した場合は、本運転の前に、装置や機械が意図どおりに動作するかを確認してください。
  モーション制御機能モジュールの機能と、軸に割り当てた EtherCAT スレーブのプロセスデータとの関係が正しくないと、装置や機械が想定外の動作をする恐れがあります。
- MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令は、必ず軸を一定速度で使用してください。
  カウントモードをロータリモードにして、OnCompensation（ON 時間補正）または OffCompensation（OFF 時間補正）を使用し、軸速度が急変すると、以下のような動作となります。
  - OnCompensation（ON 時間補正）または OffCompensation（OFF 時間補正）の値が、軸の半回転以上に相当する時間になる場合は、InOperation（作動中）が FALSE となります。
  - OnCompensation（ON 時間補正）の値が、LastOnPosition（ON 終了位置）を超える場合は、出力タイミングが不定となります。



  - OffCompensation（OFF 時間補正）の値が、FirstOnPosition（ON 開始位置）を超える場合は、出力タイミングが不定となります。



  - NX_AryDOutTimeStamp（タイムスタンプデジタル出力配列書込）命令は、MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令の InOperation（作動中）が TRUE になっていることを確認してから使用してください。

リストア時

- スレーブやユニットには、設定データのすべてまたは一部がバックアップ、リストア、照合の対象外であるものが存在します。また、無効設定されたスレーブやユニットも、バックアップ、リストア、照合の対象外です。リストア機能の使用時は、十分な動作確認を行った後に、本運転に移行してください。
- CPU ユニットは絶対値エンコーダの情報である「絶対値エンコーダ原点位置オフセット」をバッテリによって保持しています。以下のいずれかの条件を満たす場合は、絶対値エンコーダ原点位置オフセットをリストア対象からはずしてリストアを実施してください。その後、新たに絶対値エンコーダ原点設定を行ってください。絶対値エンコーダ原点設定をしなかった場合、制御対象が予期しない動作をするおそれがあります。
  - サーボモータとドライバの個体が、バックアップを作成したときの個体と異なる。
  - バックアップを作成した以降に、絶対値エンコーダのセットアップを実行した。
  - 絶対値エンコーダの絶対値データが失われている。

バッテリ交換時

- バッテリは液漏れ、破裂、発熱、発火などの恐れがありますので、＋、－の短絡、充電、分解、加熱、火への投入、強い衝撃を与えることなどは絶対に行わないでください。
- 床に落下させるなどして強い衝撃をあたえてしまったバッテリは、液漏れする恐れがありますので絶対に使用しないでください。
- バッテリの交換は熟練した技術者によって行われることが、UL 規格で定められています。交換作業は熟練した技術者がご担当ください。
- バッテリの交換前には 5 分以上通電し、電源 OFF 後 5 分以内（周囲温度 25 ℃時）に新しいバッテリと交換してください。5 分以上バッテリがない状態が続くと、保持されていたデータが消滅することがあります。
- バッテリの交換作業は、CPU ユニットの精密部品の静電気による損傷、誤動作を避けるため、無通電状態での実施を推奨いたします。電源を OFF にせず通電中にバッテリを交換することは可能ですが、その場合は、必ず接地された金属に触るなどして、人体の静電気を放電させてから交換作業に入ってください。
  バッテリ交換後、Sysmac Studio を接続して、「バッテリ電圧低下」を解除してください。

ユニット交換時

- CPU ユニットを交換するときは、運転再開に必要なユーザプログラム、「構成／設定」の設定データ、変数、CJ ユニット用メモリの値を、交換した CPU ユニット、および外部に接続される機器に転送してから、運転を開始してください。タグデータリンクの設定、ルーチングテーブル、その他 CPU 高機能ユニットのシステム設定も CPU ユニットに保存されているため転送を行ってください。
- CPU ユニットは絶対値エンコーダの情報である「絶対値エンコーダ原点位置オフセット」をバッテリによって保持しています。新規に装置を製作する、またはサーボモータを交換するなどのように、CPU ユニットとサーボモータの個体の組み合わせを変更する場合は、新たに絶対値エンコーダ原点設定を行ってください。

廃棄時

- 本体および不要になったバッテリの廃棄については、地方自治体により規制を受ける場合があります。それぞれの自治体規制に従って廃棄してください。



廃電池請回収

- リチウム一次電池（過塩素酸含有量が 6ppb 以上）を組み込んだ製品が米国・カリフォルニア州へ輸出・経由される場合、つぎの表示が義務化されています。
  Perchlorate Material - special handling may apply.
  See www.dtsc.ca.gov/hazardouswaste/perchlorate
- CPU ユニットにはリチウム一次電池（過塩素酸含有量が 6ppb 以上）を搭載しています。CPU ユニットが組み込まれた製品（最終完成品）を米国・カリフォルニア州へ輸出する場合は、個装箱、輸送箱などに上記表示を行ってください。

使用上の注意

保管時、装着時、配線時

- 次のような環境に設置や保管しないでください。運転停止、誤動作する可能性があります。
  日光が直接当たる場所
  周囲温度や相対湿度が仕様値の範囲を超える場所
  温度変化が急激で結露するような場所
  腐食性ガス、可燃性ガスのある場所
  ちり、ほこり、塩分、鉄粉が多い場所
  水、油、薬品などのひまつがかかる場所
  本体に直接振動や衝撃が伝わる場所
- 次のような場所で使用する際は、遮蔽対策を十分に行ってください。
  強い高周波ノイズを発生する機器の近く
  静電気などによるノイズが発生する場所
  強い電界や磁界が生じる場所
  放射線を被曝する恐れのある場所
  電源線や動力線が近くを通る場所
- 接地された金属に触るなどして人体の静電気を放電させてから、ユニットに触れてください。
- 発熱体の近傍への設置を避け、通風の確保などをして正しく設置してください。誤動作、運転停止、焼損の可能性があります。
- CS/CJ シリーズ用 I/O 接続ケーブルのコネクタが外れると、I/O バスチェックエラーとなり、運転を停止しますので、ご注意ください。
- ユニット開口部から異物を入れないでください。焼損、感電、故障の可能性があります。
- ユニット内に配線クズや切粉などが入らないようにしてください。焼損、故障、誤動作の原因となります。特に施工時は覆いを付けるなどの対策を行ってください。
- EtherCAT、EtherNet/IP の接続方法や使用するケーブルは、🕮『NJ/NX シリーズ CPU ユニット内蔵 EtherCAT ポート ユーザーズマニュアル（SBCD-376)』、🕮『NJ/NX シリーズ CPU ユニット内蔵 EtherNet/IP ポート ユーザーズマニュアル（SBCD-377)』の記載のとおりにしてください。通信不良になる恐れがあります。
- 電源ユニットは、定格電源電圧で使用してください。特に電源事情が悪い場所では、定格の電圧（や周波数）の電源が供給できるようにしてご使用ください。
- 電源の配線は、電線の電流容量に注意して行ってください。発熱の可能性があります。端子から端子へ渡り線で配線すると、それぞれの電線には加算された電流が流れます。渡り線で配線するときは、すべての電線の電流容量に注意してください。
- 電源を切った直後は電源ユニットの端子に触れないでください。残留電圧により感電する恐れがあります。
- AC 入力ユニットの入力接点としてリードスイッチを使用する場合は、許容電流 1A 以上のものをご使用ください。
  許容電流の小さいリードスイッチを使用した場合は、突入電流により接点が溶着することがあります。

異常処理

- 異常ステータス取得命令の検出結果を用いるアプリケーションでは、検出時に設備に与える影響を考慮のうえ、プログラミングしてください。たとえば、動作中にバッテリ交換を行った場合など軽微な異常を検出しても、ユーザプログラムしだいで動作に影響を与える可能性があります。
- コントローラ異常の重要度を変更した場合、変更によって異常発生時の出力状態が変わる場合があります。安全を確認の上、ご使用ください。

リストア時、自動転送時

・「リストアコマンドファイル」、「自動転送コマンドファイル」を編集する場合は、選択可能なデータグループの「yes/no」の記述だけにしてください。それ以外の記述を編集してリストアや自動転送すると、コントローラが意図しない動作をする場合があります。

スレーブやユニットの交換時

・ スレーブやユニットを交換した場合は、各スレーブまたは各ユニットのマニュアルを参照して、必要な設定をし直してください。

タスク設定

・「タスク周期オーバー」が発生した場合、タスクの周期内に収まるようにユーザプログラムを作成するか、またはタスクの周期を設定してください。

モーション制御

・ システム定義変数を用いて EtherCAT 通信が確立したことを確認したうえで、モーション制御命令の起動を行ってください。EtherCAT 通信が確立しない状態では、モーション制御命令を正常に起動できません。
・ システム定義変数を用いて、モーション制御機能モジュールが制御するスレーブに通信異常が発生していないかを監視してください。スレーブ通信異常が発生した状態ではモーション制御命令を正常に起動できません。
・ MC 試運転を開始する際には、動作パラメータが正しいことを確認してから MC 試運転動作を開始してください。
・ MC 試運転中にモーション設定のダウンロードは行わないでください。

EtherCAT 通信

・ 運転中に EtherCAT スレーブケーブルを抜く場合は、EtherCAT スレーブを離脱する、もしくは、後続する EtherCAT スレーブも含めて無効とした後で抜いてください。
・ コントローラとサーボドライバ間の EtherCAT 通信に異常が起こった場合に、動作を停止するようにサーボドライバを設定してください。
・ EtherCAT マスタ機能モジュールでの「ネットワーク構成照合異常」、「プロセスデータ通信異常」、「リンクオフ異常」を解除する場合は、復旧対象のすべてのスレーブが加入状態であることを事前に確認してください。加入していないスレーブがあると、EtherCAT マスタ機能モジュールが、指定と異なるノードアドレスのスレーブにアクセスしたり、正しく異常解除されなかったりする可能性があります。
・ 必ず指定されている EtherCAT スレーブケーブルを使用してください。指定外のケーブルを使用した場合、EtherCAT マスタまたは EtherCAT スレーブが異常を検出し、以下のような現象が発生する可能性があります。
  ・ プロセスデータ通信のリフレッシュが、継続的に実行されない。
  ・ プロセスデータ通信のリフレッシュが、設定周期内に完了しない。

バッテリ交換時

- バッテリのラベルを確認し、交換用には、製造年月が 2 年以内のものを使用してください。
- 長期間放置された CPU ユニットはバッテリを交換してから、電源を投入してください。
  バッテリ交換後、一度も電源投入しない状態で、再び放置しておくと、バッテリ寿命が短くなることがあります。
- バッテリ交換時はバッテリセット形 CJ1W-BAT01 をご使用ください。

SD メモリカード

- SD メモリカードは奥まで差し込んでください。
- SD メモリカードアクセス中にコントローラの電源を切らないでください。ファイルが壊れる場合があります。
  SD メモリカード内に壊れたファイルがあると、SD メモリカード修復機能によって、電源投入時に自動的にそのファイルが削除されます。

オンラインエディット

- CPU ユニット Ver.1.04 以降と Sysmac Studio Ver.1.05 以降の組み合わせでオンラインエディットをする場合、CPU ユニットはオンラインエディットで更新したプログラムを内蔵の不揮発性メモリにバックアップします。この間、Sysmac Studio はバックアップ中であることをメッセージで表示します。このメッセージが表示されている間はコントローラの電源を切らないでください。コントローラの電源を切ると、次回の電源投入時にコントローラが異常を発生します。

法規と規格

海外でのご使用

本製品のうち、外国為替および外国貿易管理法に定める輸出許可、承認対象貨物（または技術）に該当するものを輸出（または非居住者に提供）する場合は、同法に基づく輸出許可、承認（または役務取引許可）が必要です。

EMC および電気安全に関わる法規制への適合

適合の考え方

オムロンの商品は、各種機械、製造装置に組み込まれ使用される産業用電気機器であるため、組み込んだ機械や装置がより容易に EMC および電気安全に関わる法規制に適合できるように、商品自身の関連する規格への適合を図っています。

適合規格については、当社ホームページ（www.fa.omron.co.jp または www.ia.omron.com）、または当社営業担当者に確認してください。

● EMC 規制への適合

この商品は PLC システムまたはマシンオートメーションコントローラに組み込まれた状態で EMC 規制に適合しています。お客様の機械や装置を EMC 規制に適合させるにあたり、次の注意事項を必ず守ってください。

- この商品は盤内蔵型として定義されるため、必ず制御盤内に設置してください。
- この商品はエミッション規格に適合していますが、特に放射エミッションについては、使用する制御盤の構成や接続機器、配線方法により変化することがあります。したがって、EMC 規制に 適合した商品を使用する場合でも、お客様にて機械や装置全体の EMC 規制への適合を確認していただく必要があります。
- DC 電源ユニットおよび I/O ユニットに接続する DC 電源は、強化絶縁、または二重絶縁されたものを使用してください。

注意：
この商品は住宅環境での使用を意図していません。住宅環境で利用されると、電波妨害の原因となる可能性があります。

● 電気安全への適合

この商品は EU 指令、UKCA などの特定の法規制で求められる電気安全に適合しています。
形式ごとの注意事項については、製品同梱の取扱説明書を確認してください。

KC 規格への適合

本製品を韓国で使用する場合には、以下の注意事項を守ってください。

A 급 기기 (업무용 방송통신기자재)
이 기기는 업무용(A 급) 전자파적합기기로서 판매자
또는 사용자는 이 점을 주의하시기 바라며, 가정외의
지역에서 사용하는 것을 목적으로 합니다.

ClassA 機器（業務用放送通信機器）
この機器は、業務用電磁波発生機器（ClassA）であり、家庭以外の場所での使用を意図しています。販売者やユーザは、この点に注意してください。

船級規格への適合

本製品は、各種船級規格に適合しています。各船級規格への適用にあたっては使用条件が設定されており設置場所によっては使用できない場合もあるため、ご使用の際は必ず弊社営業へお問い合わせください。

各船級規格の使用条件（NK 、LR）

- 本製品は、必ず制御盤に設置してください。
- 制御盤の開閉口などの隙間は、ガスケットなどで完全に遮蔽してください。
- 電源ラインに以下のノイズフィルタを接続してください。

ノイズフィルタ

メーカ	形式
コーセル社	TAH-06-683

ソフトウェアのライセンスと著作権

本製品にはサードパーティ製のソフトウェアが組み込まれています。このソフトウェアに関連するライセンスと著作権については、http://www.fa.omron.co.jp/nx_info_j/ をご覧ください。

- OpenSSL
  This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit
  (http://www.openssl.org/).
  Copyright (C) 1998-2019 The OpenSSL Project. All rights reserved.
  Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.
  This product includes cryptographic software written by Eric Young
  (eay@cryptsoft.com)

バージョン

NJ/NX シリーズの各ユニットおよび各 EtherCAT スレーブは、ハードウェアやソフトウェアが、ユニットバージョンという改変番号でバージョン管理されています。ハードウェアおよびソフトウェアに仕様変更があるごとに、ユニットバージョンが更新されます。したがって、同じ形式のユニットや EtherCAT スレーブでも、ユニットバージョンが異なると搭載機能や性能が異なります。

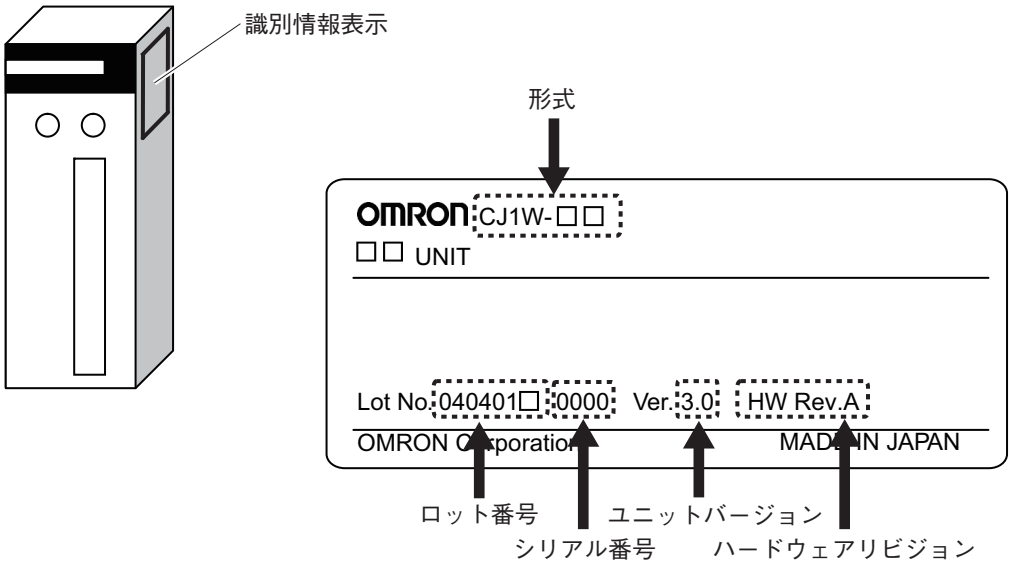
バージョンの確認方法

バージョンは、識別情報表示または Sysmac Studio で確認できます。

識別情報表示による確認

製品側面の識別情報表示で、バージョンが確認できます。

形 CJ1W-EIP21/EIP21S の場合：



識別情報表示には以下の情報が記載されています。

名称	機能
形式	本ユニットの形式を示します
ロット番号とシリアル番号	本ユニットのロット番号とシリアル番号を示します。 YYMMDD：ロット番号、□：弊社使用、xxxx：シリアル番号 YY: 西暦下二けた、MM: 月、DD: 日を表します。
ユニットバージョン	本ユニットのユニットバージョンを示します
ハードウェアリビジョン	本ユニットのハードウェアリビジョンを示します。*1

*1. ハードウェアリビジョンが「なし」のユニットでは、ハードウェアリビジョンの表示はありません。

Sysmac Studio による確認方法

Sysmac Studio で、バージョンが確認できます。ユニットと EtherCAT スレーブでは、確認方法が異なります。

● ユニットのバージョン確認方法

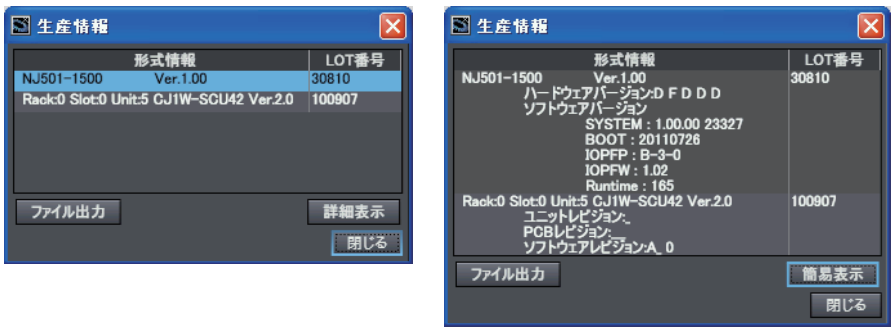
ユニットのバージョンは、オンライン状態の［生産情報］で確認できます。ただし、バージョンが確認できるユニットは、CPU ユニット、CJ シリーズの高機能 I/O ユニット、および CJ シリーズの CPU 高機能ユニットだけです。CJ ユニットの基本 I/O ユニットのバージョンは、Sysmac Studio では確認できません。
バージョンの確認方法は、以下のとおりです。

1 **マルチビューエクスプローラで［構成・設定］－［CPU・増設ラック］をダブルクリックします。または、［構成・設定］－［CPU・増設ラック］を右クリックして［編集］を選択します。**
ユニットエディタが表示されます。

2 **ユニットエディタの空欄上で右クリックして［生産情報表示］を選択します。**
［生産情報］ダイアログボックスが表示されます。

● 生産情報の表示内容の切り替え

1 **[生産情報] ダイアログボックスの右下の、[簡易表示] または [詳細表示] を選択します。**
[生産情報] の簡易表示と詳細表示が、切り替わります。



簡易表示　　詳細表示

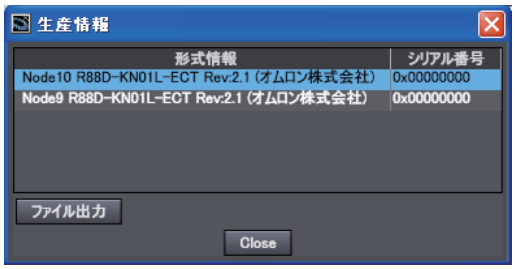
表示内容は、簡易表示と詳細表示とで異なります。詳細表示では、ユニットバージョン、ハードウェアバージョン、およびソフトウェアバージョンが表示されます。簡易表示では、ユニットバージョンだけが表示されます。

● EtherCAT スレーブのバージョン確認方法

EtherCAT スレーブのバージョンは、オンライン状態の [生産情報] で確認することができます。確認方法は、以下のとおりです。

1 **マルチビューエクスプローラで［構成・設定］－［EtherCAT］をダブルクリックします。または、［構成・設定］－［EtherCAT］を右クリックして［編集］を選択します。**
EtherCAT 構成編集画面が表示されます。

2 **EtherCAT 構成編集画面でマスタを右クリックして［生産情報表示］を選択します。**
生産情報ダイアログボックスが表示されます。
ユニットバージョンは、「Rev」を付けて表示されます。



バージョン変更時の手順

Sysmac Studio のプロジェクトファイルに設定した EtherNet/IP ユニット形 CJ1W-EIP21 のユニットバージョンの変更方法は次のとおりです。
以下の手順により、異なるバージョンの EtherNet/IP ユニットに置き換える場合に、以前のユニットの設定を引き継ぐことができます。

1 **Sysmac Studio で、形 CJ1W-EIP21 Ver2.1 の設定を含むコントローラプロジェクトを開きます。**

2 **CPU 増設ラックの編集画面で「CJ1W-EIP21 Ver2.1」を選択し、[高機能ユニット設定のエクスポート] メニューにより、設定をパラメータファイル（拡張子 .upf）に保存します。**

3 **以下の 2 つの場合によって、操作が異なります。**

- EtherNet/IP のコネクション設定（タグデータリンク設定）を SysmacStudio で行っていない場合

(1) Sysmac Studio で、I/O マップを開き、形 CJ1W-EIP21 の変数をコピーし、Excel のワークシートに貼り付けます。そのファイルを保存します。

(2) CPU 増設ラックの編集画面で「CJ1W-EIP21 Ver2.1」を選択し、[形式変更] メニューにより、「CJ1W-EIP21 Ver3.0」に変更します。

(3) Excel のワークシートに貼り付けた変数をコピーし、I/O マップを開いて、形 CJ1W-EIP21 の変数欄に貼り付けます。これにより復元します。

- EtherNet/IP のコネクション設定（タグデータリンク設定）を SysmacStudio で行っている場合、または上記コピー & ペーストによる作業を回避したい場合

(1) CPU 増設ラックの構成は同じだが形 CJ1W-EIP21 のみ Ver.3.0 のユニットに交換した実機コントローラを用意し、Sysmac Studio からオンライン接続します。

(2) CPU 増設ラックの編集画面で [比較・マージ] メニューにより、コントローラプロジェクトの形 CJ1W-EIP21 のバージョンを Ver.2.1 から Ver.3.0 に変更します。

4 **CPU 増設ラックの編集画面で「CJ1W-EIP21(NJ) Ver3.0」を選択し、[高機能ユニット設定のインポート] メニューにより、手順 2. で保存しておいたパラメータファイル（拡張子 .upf）をインポートして、設定を復元します。**

CIP リビジョンとの対応表

EtherNet/IP ユニットのユニットバージョンと CIP リビジョンとの対応表を以下に記載します。

● CJ1W-EIP21 の場合

ユニットバージョン	CIP リビジョン
Ver.2.1	Rev.2.04
Ver.3.0	Rev.3.01

● CJ1W-EIP21S の場合

ユニットバージョン	CIP リビジョン
Ver.1.0	Rev.4.01

ユニットバージョンと周辺ツールのバージョンについて

EtherNet/IP ユニットのユニット設定には、以下に記載したバージョンの Sysmac Studio や Network Configurator を使用します。

● CJ1W-EIP21 の場合

ユニットバージョン	Sysmac Studio		
	Ver.1.01 以前	Ver.1.02	Ver.1.11 以降
Ver.2.1	×	○	○
Ver.3.0	×	×	○

ユニットバージョン	Network Configurator for EtherNet/IP		
	Ver.3.40 以前	Ver.3.50	Ver.3.57 以降
Ver.2.1	×	○	○
Ver.3.0	×	×	○

● CJ1W-EIP21S の場合

ユニットバージョン	Sysmac Studio	
	Ver.1.59 以前	Ver.1.60 以降
Ver.1.0	×	○

ユニットバージョン	Network Configurator for EtherNet/IP	
	Ver.3.75 以前	Ver.3.76 以降
Ver.1.0	×	○

ユニットバージョンと、接続可能な CPU ユニットの機種について

EtherNet/IP ユニットが接続できる CPU ユニットの種類は、EtherNet/IP ユニットのユニットバージョンによって、以下のようになります。

● CJ1W-EIP21 の場合

ユニットバージョン	CPU ユニットの種類	
	CJ シリーズ CPU ユニット	NJ シリーズ CPU ユニット
Ver.2.0 以前	○	×
Ver.2.1	○	○ *
Ver.3.0 以降	○	○ *

* 接続する CPU ユニットは、Ver.1.01 以降、Sysmac Studio は Ver.1.02（ユニットバージョン Ver.2.1 の場合）または、Ver.1.11（ユニットバージョン V3.0 の場合）を使用してください。

● CJ1W-EIP21S の場合

ユニットバージョン	CPU ユニットの種類	
	CJ シリーズ CPU ユニット	NJ シリーズ CPU ユニット
Ver.1.0（ロット番号が 241001□ 未満）	○	×
Ver.1.0（ロット番号が 241001□ 以降）*1	○	○ *2

*1. ロット番号 241001□ 以降の場合は、ユニットの前面右下に「+NJ」または「+CJ/CP/NJ」と印字されています。前面右下の印字については、「1-3-2 各部の名称と機能」を参照してください。
*2. 接続する CPU ユニットは、Ver.1.67 以降、Sysmac Studio は Ver.1.60 以降を使用してください。

関連マニュアル

本マニュアルに関連するマニュアルは、下表のとおりです。あわせてご覧ください。

マニュアル名称	Man.No.	形式	用途	内容
NX 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-418	形 NX701-□□□□	NX シリーズ CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。おもにハードウェアに関する情報。	NX シリーズのシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検 ユーザーズマニュアル　ソフトウェア編（SBCA-467）と併せて使用してください。
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	形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ シリーズ CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。おもにハードウェアに関する情報。	NJ シリーズのシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検 ユーザーズマニュアル　ソフトウェア編（SBCA-467）と併せて使用してください。
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	形 NX701-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットのプログラミング／システムの立ち上げについて知りたいとき。おもにソフトウェアに関する情報。	NJ/NX シリーズ CPU ユニットに関して、以下の内容を説明します。 • CPU ユニットの動作 • CPU ユニットの機能 • 初期設定 • IEC61131-3 ベースの言語仕様とプログラミング ユーザーズマニュアル　ハードウェア編（SBCA-418、または SBCA-466）と併せて使用してください。
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	形 NX701-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズの基本命令仕様の詳細について知りたいとき。	各命令（IEC 61131-3 仕様）の詳細を説明します。 プログラミング時に、ユーザーズマニュアル ハードウェア編（SBCA-418、または SBCA-466）／ソフトウェア編（SBCA-467）と併せて使用してください。
NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	SBCE-433	形 NX701-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	モーション制御の設定やプログラミングの考え方について知りたいとき。	モーション制御のための CPU ユニットの設定や動作、プログラミングの考え方について説明します。 プログラミング時に、ユーザーズマニュアル ハードウェア編（SBCA-418、または SBCA-466）／ソフトウェア編（SBCA-467）と併せて使用してください。
NJ/NX シリーズ コマンドリファレンスマニュアル モーション編	SBCE-434	形 NX701-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	モーション命令仕様の詳細について知りたいとき。	各モーション命令の詳細を説明します。 プログラミング時に、ユーザーズマニュアル ハードウェア編（SBCA-418、または SBCA-466）／ソフトウェア編（SBCA-467）、ユーザーズマニュアル　モーション制御編（SBCE-433）と併せて使用してください。
NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT® ポート ユーザーズマニュアル	SBCD-376	形 NX701-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットの内蔵 EtherCAT ポートを使用するとき。	内蔵 EtherCAT ポートに関して説明します。概要、構成、機能、セットアップについて記述しています。 ユーザーズマニュアル　ハードウェア編（SBCA-418、または SBCA-466）／ソフトウェア編（SBCA-467）と併せて使用してください。

マニュアル名称	Man.No.	形式	用途	内容
NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP™ ポート ユーザーズマニュアル	SBCD-377	形 NX701-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットの内蔵 EtherNet/IP ポート を使用するとき。	内蔵 EtherNet/IP ポートに関して説明します。 基本設定、タグデータリンク、その他の機能について記述しています。 ユーザーズマニュアル　ハードウェア編（SBCA-418、または SBCA-466）／ソフトウェア編（SBCA-467）と併せて使用してください。
NJ シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	SBCA-411	形 NJ501-1□20	NJ シリーズでデータベース接続サービス機能を使用するとき。	データベース接続サービス機能について説明します。
NJ シリーズ SECS/GEM 搭載 CPU ユニット ユーザーズマニュアル	SBCA-412	形 NJ501-1340	NJ シリーズで GEM サービス機能を使用するとき。	GEM サービスの機能について説明します。
NJ/NX シリーズ トラブルシューティング マニュアル	SBCA-469	形 NX701-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズで検出する異常の詳細について知りたいとき。	NJ/NX シリーズ システムにて検出する異常管理の考え方と各異常項目について説明します。 ユーザーズマニュアル　ハードウェア編（SBCA-466）／ソフトウェア編（SBCA-467）と併せて使用してください。
Sysmac Studio Version 1 オペレーションマニュアル	SBCA-470	形 SYSMAC -SE2□□□	Sysmac Studio の操作方法、機能について知りたいとき	Sysmac Studio の操作方法について説明します。
CJ シリーズ EtherNet/IP™ ユニット ユーザーズマニュアル NJ シリーズ接続編	SBCD-355	形 CJ1W-EIP21 形 CJ1W-EIP21S	EtherNet/IP ユニットを使用するとき	EtherNet/IP ユニットを NJ CPU ユニットに接続して使用する場合について説明します。 基本設定、タグデータリンク、その他の機能について記述しています。 ユーザーズマニュアル　ハードウェア編（SBCA-466）／ソフトウェア編（SBCA-467）と併せて使用してください。

マニュアル改訂履歴

マニュアル改訂記号は、表表紙・裏表紙に記載されている Man.No. の後尾に付記されます。



改訂記号	改訂年月	改訂理由・改訂ページ
A	2012 年 3 月	初版
B	2012 年 5 月	• CPU ユニット Ver1.02 以降のサポート機能に関する記述の追記 • 誤記修正
C	2012 年 8 月	• CPU ユニット Ver.1.03 へのバージョンアップに伴う改訂
D	2013 年 2 月	• 誤記修正
E	2013 年 4 月	• 誤記修正
F	2014 年 11 月	• 許容通信帯域幅を 12,000pps に拡張
G	2015 年 4 月	• NJ シリーズの形 NJ101-□□□□を追加 • 誤記修正
H	2018 年 4 月	• 誤記修正
J	2020 年 1 月	• 誤記修正
K	2020 年 12 月	• 誤記修正
L	2023 年 5 月	• 誤記修正
M	2024 年 10 月	• CJ1W-EIP21S の追加に伴う改訂 • CPU ユニット Ver.1.67 へのバージョンアップに伴う改訂 • 誤記修正
N	2025 年 4 月	• ハードウェアリビジョンに関する記述の追加 • 誤記修正

1

概要

1-1 概要 1-2
1-1-1 EtherNet/IP の特長 1-2
1-1-2 EtherNet/IP ユニットの特長 1-3

1-2 システム構成と構成機器 1-5
1-2-1 ネットワーク構築に必要な機器 1-5
1-2-2 ネットワーク構築に必要な周辺ツール 1-6

1-3 EtherNet/IP ユニット 1-7
1-3-1 仕様 1-7
1-3-2 各部の名称と機能 1-9
1-3-3 外形寸法 1-16

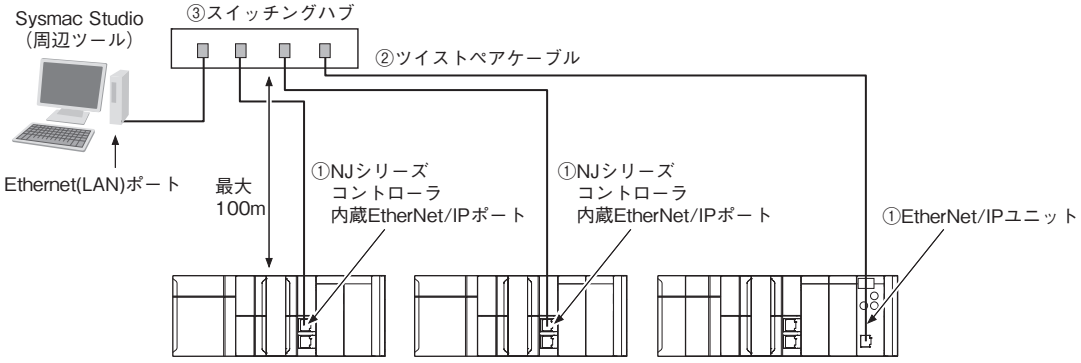
1-4 通信サービスの概要 1-17
1-4-1 CIP 通信（Common Industrial Protocol）サービス 1-17
1-4-2 BOOTP クライアント機能 1-19
1-4-3 FTP サーバ機能 1-19
1-4-4 時計自動調整機能 1-20
1-4-5 ホスト名指定機能 1-20
1-4-6 SNMP（エージェント）機能 1-21
1-4-7 ソケットサービス機能（形 CJ1W-EIP21S のみ） 1-21

1-5 EtherNet/IP 通信の使用手順 1-22
1-5-1 基本操作 1-22
1-5-2 タグデータリンクを使用時 1-24
1-5-3 EtherNet/IP ネットワークの各機能を使用する 1-24

1-1　概要

1-1-1　EtherNet/IP の特長

EtherNet/IP は、Ethernet を使用した産業用のマルチベンダネットワークです。
その仕様はオープンな規格として、DeviceNet と同様に ODVA（Open DeviceNet Vendor Association）にて管理されており、さまざまな産業用機器に採用されています。
EtherNet/IP は、コントローラ間のネットワークとしてだけでなく、フィールドネットワークとしても使用可能です。また、標準の Ethernet 技術が使用されているため、さまざまな汎用 Ethernet 機器とも混在させて使用することができます。



EtherNet/IPシステム構成例

● タグデータリンク（サイクリック通信）による高速・大容量なデータ交換

EtherNet/IP 規格標準の Implicit 通信をサポートしており、これにより EtherNet/IP 機器とサイクリック通信（本書ではタグデータリンクと呼びます）が可能です。

● ノード数に依存せず、アプリケーションごとに指定した周期でのタグデータリンク（サイクリック通信）が可能

コネクションごとに設定した更新周期で、回線上をデータ交換するため、ノード数が増加しても、通信上の更新周期は増加しません（コネクション内のデータの同時性は確保されます）。
また、コネクションごとに更新周期を設定できるため、アプリケーションごとに最適な更新周期で通信が可能です（例：工程間のインターロックは高速、生産指示や工程のステータスモニタは低速など）。

1-1-2　EtherNet/IP ユニットの特長

● タグデータリンク

EtherNet/IP ネットワーク上で、コントローラーコントローラ間、またはコントローラー他のデバイス間でのサイクリック通信が可能です。
タグデータリンクでは、最大 184832CH のデータを高速にデータ交換できます。

● メッセージ通信機能

EtherNet/IP ネットワーク上のデバイスに対して、プログラム上で CIP 通信命令を実行することにより、必要時に任意の CIP コマンドを発行することができます。
これにより、EtherNet/IP ネットワーク上のデバイスとデータの送受信することが可能となります。

● BOOTP クライアント機能

EtherNet/IP ユニットを BOOTP 設定にすると、コントローラの電源 ON 時に BOOTP クライアント機能が動作し、BOOTP サーバから IP アドレスを取得できます。
複数の EtherNet/IP ユニットの IP アドレスを一括して設定することが可能です。

● ホストコンピュータとの間でファイル単位のデータ転送（FTP サーバ機能）

FTP サーバ機能を内蔵しているため、FTP クライアント機能を持つワークステーションやパソコンから、ファイル単位でコントローラ内のデータを読み出すことや、コントローラ内へデータを書き込むことができます。
ラダープログラムを記述しなくても、クライアントから大容量のデータを一度に転送できます。

● コントローラ内蔵時計の自動調整

SNTP サーバの時計を基準にして、Ethernet に接続したコントローラの内蔵時計を自動調整することができます。システム全体で自動調整しておくと、生産設備に発生したさまざまな記録を、時計情報をもとに整理し分析することができるようになります。
* 時計の自動調整には、別途 SNTP サーバが必要となります。

● ホスト名での指定

SNTP サーバや SNMP マネージャの IP アドレス、または CIP 通信命令での送信先について、直接 IP アドレスを指定する以外に、ホスト名で指定することも可能です（DNS クライアント機能）。
これにより、システムの見直しなどで、各サーバの IP アドレスが変更になった場合にも、自動で IP アドレスを検索します。
* DNS によるホスト名でのサーバ指定には、別途 DNS サーバが必要となります。
* DNS サーバの IP アドレスは、直接 IP アドレスで指定します。

● SNMP マネージャによるネットワーク管理が可能

SNMP（エージェント）機能は、SNMP マネージャを使用したネットワーク管理ソフトウェアに EtherNet/IP ユニットの内部状態の情報を提供します。
* ネットワーク管理には、別途 SNMP マネージャが必要となります。

● 豊富なトラブル対処機能

迅速にトラブルに対処できる豊富な機能が用意されています。
・電源投入時の自己診断機能
・異常発生時の異常情報を記録するイベントログ機能

● ソケットサービス機能（形 CJ1W-EIP21S のみ）

汎用アプリケーションとコントローラ間のデータ送受信で、ソケットサービスが使用できます。このソケットを使用した通信サービスにより、ホストコンピュータとコントローラの間、またはコントローラとコントローラの間で、任意のデータを送受信することができます。
プログラム上でデバイス変数を操作、またはコマンド送信命令を使用することにより、ソケットサービスの各通信処理を実行していきます。
ソケットサービス機能は、UDP ソケットサービス、TCP ソケットサービスの 2 種類があります。

● セキュリティ機能（形 CJ1W-EIP21S のみ）

ポートのオープン／クローズ機能や Packet Filter 機能により外部からの侵入に対するセキュリティ性能を向上できます。

参考

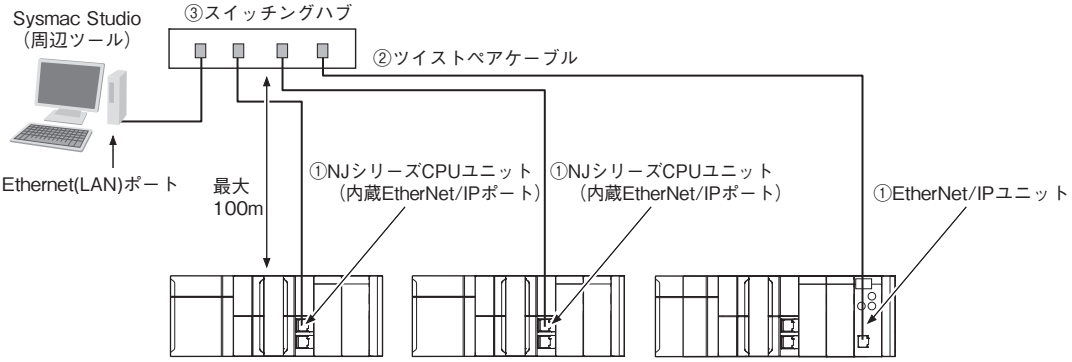
CIP（Common Industrial Protocol）とは
OSI アプリケーション層の産業用共通プロトコルです。EtherNet/IP、CompoNet、DeviceNet などで使用されています。
CIP をベースとしたネットワーク間では、ネットワーク間のルーチングを容易に行うことができます。そのため、フィールド機器から上位まで、透過的なネットワークを容易に構築できます。
CIP には、以下の特徴があります。

- 相手先ノードを、固定ルーチングテーブルではなく、相対パスで指定します。
- Producer/Consumer モデルを使用します。ネットワーク上のノードは、水平分散されていて、必要な機器と必要な通信をすることができます。
  Consumer は、Producer から生産（送信）されたデータを消費（受信）するように、必要な分を必要な性能で選択します。一方、Producer は、同じ性能の同じデータをマルチキャストで送信することも可能なため（マルチキャスト／ユニキャストの選択が可能）、この場合、転送の所要時間が Consumer（ノード）の数に関係なく一定となります。

1-2　システム構成と構成機器

1-2-1　ネットワーク構築に必要な機器

EtherNet/IP システムの基本構成は、スイッチングハブとそれにスター型で接続されるツイストペアケーブルおよび各ノードから成ります。



ネットワークを構成するには、以下に示す機器が別途必要となりますので、あらかじめ用意してください。

ネットワーク構成機器	内容
①各ノード NJ シリーズ CPU ユニット（内蔵 EtherNet/IP ポート） （形 NJ501- □□□□、形 NJ301- □□□□、形 NJ101- □□□□） CJ シリーズ EtherNet/IP ユニット （形 CJ1W-EIP21*1 ／形 CJ1W-EIP21S*2） その他オムロン製 コントローラ CJ2 CPU ユニット（内蔵 EtherNet/IP ポート） （形 CJ2H-CPU □□ -EIP ／形 CJ2M-CPU3 □） CS シリーズ EtherNet/IP ユニット （形 CS1W-EIP21 ／形 CS1W-EIP21S）	EtherNet/IP ネットワークに接続するためのユニットです。
②ツイストペアケーブル	RJ45 形式のモジュラコネクタを両端につけたツイストペアケーブル。 内蔵 EtherNet/IP ポート、または EtherNet/IP ユニットとスイッチングハブの間を接続します。 カテゴリ 5、5e 以上の STP（シールド付きツイストペア）ケーブルを使用します。
③スイッチングハブ	複数のノードをスター型 LAN で接続するための中継器です。 ネットワーク構成機器の推奨品は、「2-3-1 ネットワーク構成機器の推奨品」を参照してください。

*1. NJ シリーズ CPU ユニットの Ver.1.01 以降、かつ Sysmac Studio の Ver1.02 以降で使用可能です。
*2. NJ シリーズ CPU ユニットの Ver.1.67 以降、かつ Sysmac Studio の Ver.1.60 以降で使用可能です。

使用上の注意

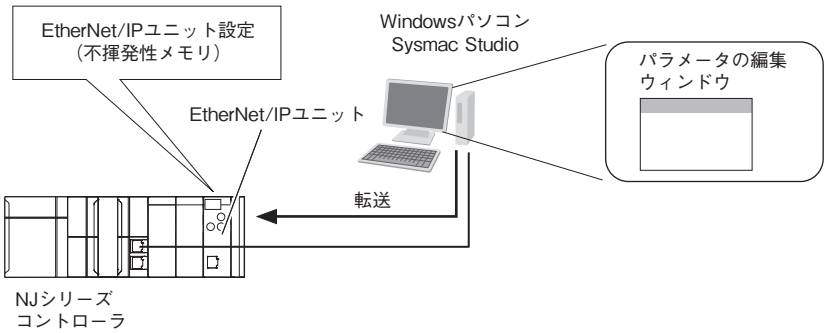
- Sysmac Studio は、本ユニットを経由して NJ シリーズ CPU ユニットにオンライン接続することはできません。
- 表示器を本ユニットを経由してコントローラに接続している場合は、表示器のトラブルシュータは使用できません。

1-2-2　ネットワーク構築に必要な周辺ツール

EtherNet/IP ネットワークの構築に必要な周辺ツールについて記述します。
EtherNet/IP ユニットには、大きく分けて「Ethernet 設定」と「タグデータリンク設定」があり、いずれも EtherNet/IP ユニット内の不揮発性メモリに保存されます。
各設定には、それぞれ専用のツールがあります。

● EtherNet/IP ユニット設定：Sysmac Studio

EtherNet/IP ユニットの自 IP アドレス、サブネットマスクなどの基本的な設定は、Sysmac Studio を使用して行います。
またタグデータリンクのデータ入出力が正常に行えているかの確認にも使用します。



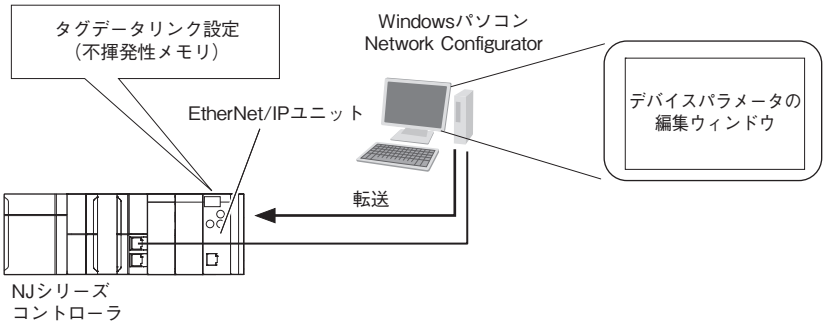
Sysmac Studio に関する詳細は 🕮『Sysmac Studio Version 1 オペレーションマニュアル』（SBCA-470）を参照してください。

● タグデータリンク設定：Network Configurator

EtherNet/IP ユニットのタグデータリンクの設定は、Network Configurator（Sysmac Studio スタンダードエディションに含まれる）を使用して行います。主な機能は次のとおりです。

1）タグデータリンク（コネクション）の設定・モニタ
ネットワークのデバイス構成やタグデータリンク（コネクション）の作成・編集を行います。また、ネットワークに接続し、デバイス構成やタグデータリンク設定を読み出し、モニタリングを行うこともできます。

2）マルチベンダ機器接続機能
他社製 EtherNet/IP デバイスを含むネットワークの構築・設定・管理が行えるよう、EDS ファイルのインストールや削除を行うことができます。また、EtherNet/IP デバイスの IP アドレスを変更することができます。



Network Configurator に関する詳細は🕮「第 7 章 タグデータリンク機能」を参照してください。

参考

タグデータリンクの設定は Sysmac Studio で行うこともできます。
Sysmac Studio での設定方法の詳細については、🕮「A-2 Sysmac Studio でタグデータリンク（EtherNet/IP コネクション）を設定する」を参照してください。

1-3　EtherNet/IP ユニット

1-3-1　仕様

項　目		仕様
品名		EtherNet/IP ユニット
ユニットの種類		CPU 高機能ユニット
装着可能位置		CPU ラックまたは増設ラック
装着可能台数		最大 4 台
通信プロトコル		TCP/IP、UDP/IP
サポートサービス		Sysmac Studio 接続、タグデータリング、CIP メッセージ通信、FTP サーバ、時計自動調整（SNTP クライアント）、SNMP（エージェント）、DNS（クライアント）、BOOTP（クライアント）
物理層		100BASE-TX/10BASE-T（ただし、100BASE-TX を推奨します。）*1
伝送仕様	媒体アクセス方式	CSMA/CD
	変調方式	ベースバンド
	伝送路形式	スター型
	伝送速度	100M ビット /s（100BASE-TX）
	伝送媒体	ツイストペアケーブル（シールド付：STP）： カテゴリ 5、5e 以上
	伝送距離	最大 100m（ハブとノード間の距離）
	カスケード接続数	スイッチングハブの使用において制限なし
CIP サービス タグデータリンク （サイクリック通信）	コネクション数	256
	パケットインターバル （更新周期）	0.5 ～ 10000ms（0.5ms 単位） コネクションごとに設定可能（ノード数に依存せず、設定した周期で回線上をデータ更新）
	ユニット許容通信帯域幅	6000 ～ 12000pps*2 （形 CJ1W-EIP21 のユニットバージョン Ver.2.1 以前の場合は、6000pps） 注：ハートビートを含む
	登録可能なタグ数	256
	タグ種別	ネットワーク変数 CIO/WR/HR/DM/EM
	1 コネクション（＝ 1 タグセット）あたりのタグ数	8（タグセットにコントローラステータスを含める場合は 7 ）
	1 ノード当たりの最大リンクデータサイズ	369664 バイト
	1 コネクションあたりの最大データサイズ	1444 バイト *3 コネクション内でのデータの同時性は確保されます。 同時性を確保するための方法は、「7-1-7 タグデータリンクするデータの同期性について」を参照してください。
	登録可能なタグセット数	256（1 コネクション= 1 タグセット）
	1 タグセットの最大サイズ	722CH （タグセットにコントローラステータスを含める場合は 1CH 分を使用）
	タグデータリンクパラメータ設定のコントローラの運転中変更	可 *4
	マルチキャストパケットフィルタ機能 *5	可

*1　タグデータリンクをご使用の場合は、100BASE-TX にてお使いください。
*2　Packet Per Second を意味し、1 秒間に処理可能な送受信パケット数を示します。
*3　データサイズとして 505byte 以上を使用するには、Large Forward Open（CIP オプション仕様）に対応している必要があります。CS/CJ/NJ シリーズ間では使用可能ですが、他社ノードとの接続の場合、使用される機器が Large Forward Open 仕様を対応しているかご確認ください。
*4　ただし、パラメータ変更時には、変更対象の EtherNet/IP ユニットはリスタートしますので、ご注意ください。また、変更対象と通信していた他ノードでは、変更対象との通信がいったんタイムアウト状態になり、リスタート後に自動復帰します。
*5　EthreNet/IP ユニットは、IGMP クライアント（Version 2）を実装しているため、IGMP Snooping 対応のスイッチングハブを使用することで不要なマルチキャストパケットのフィルタリングが行えます。

項　目		仕様
CIP メッセージサービス：Explicit メッセージ *6	Class3（コネクション数）	コネクション：128（クライアント + サーバ）
	UCMM（非コネクション型）	同時通信可能な最大クライアント数：32 同時通信可能な最大サーバ数：32
	CIP ルーチング（*7）	可 CIP ルーチング可能な相手ユニットは以下のとおり 形 NJ501- □□□□、形 NJ301- □□□□、 形 NJ101- □□□□、 形 CS1W-EIP21/EIP21S、形 CJ1W-EIP21/EIP21S、 形 CJ2H-CPU □□ -EIP、形 CJ2M-CPU3 □
SNMP	エージェント	SNMPv1、SNMPv2c
	MIB	MIB-II
EtherNet/IP コンフォーマンステスト		CT8 準拠
Ethernet インタフェース		10BASE-T/100BASE-TX Auto Negotiation ／固定設定
装着可能な CPU ユニット形式		形 NJ101- □□□□ CPU ユニット（Ver.1.10 以降） 形 NJ301- □□□□ CPU ユニット（Ver.1.01 以降） 形 NJ501- □□□□ CPU ユニット（Ver.1.01 以降）
消費電流（ユニット）		DC5V 410mA 以下
質量		94g 以下
外形寸法		31（W）×90（H）×65（D）mm

*6　EtherNet/IP ユニットは、下記に示す TCP/UDP ポート番号を使用します。
TCP 内、UDP 内で使用するポート番号が重複しないように設定してください。

サービス	種別	ポート番号	補　足
EIP データリンク	UDP	2222	固定値
CIP メッセージ	TCP/UDP	44818	
DNS クライアント	UDP	53	
BOOTP クライアント	UDP	68	
FTP サーバ（データ転送ポート）	TCP	20 ポート番号は、（制御ポート）- 1	Sysmac Studio のユニット設定で変更可能
FTP サーバ（制御ポート）	TCP	21	
SNTP クライアント	UDP	123	
SNMP エージェント	UDP	161	
SNMP トラップ	UDP	162	

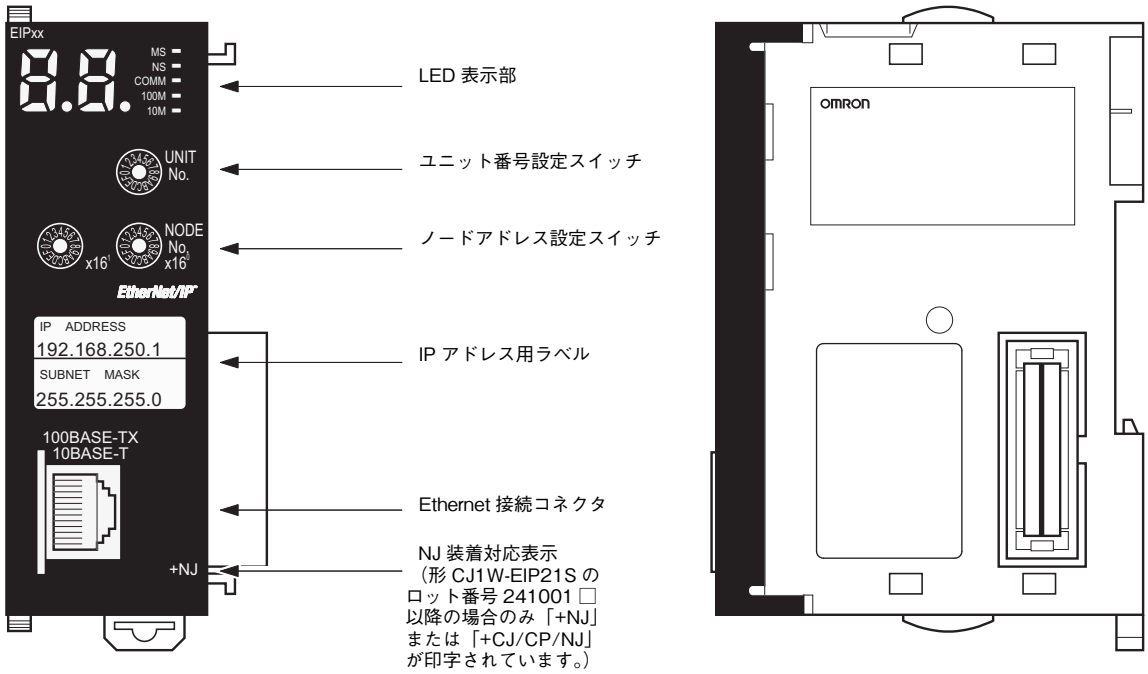
*7　CIP ルーチング機能は、CPU ユニットの Ver.1.01 以降、かつ Sysmac Studio Ver1.02 以降で、使用可能です。

形 CJ1W-EIP21S の取り付け制約事項

● ユニット起動時間が長いことへの対応

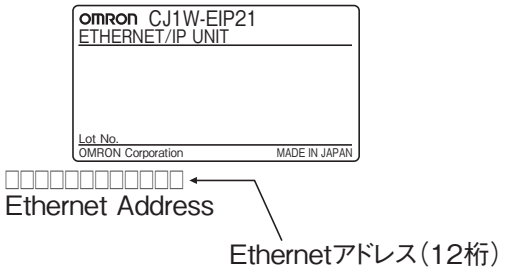
形 CJ1W-EIP21S のユニット起動時間は、形 CJ1W-EIP21 と比べて数秒程度長くなっています。CPU ユニットは電源を投入ののち、すべての高機能 I/O ユニットおよび CPU 高機能ユニットに対する認識を完了してから、運転開始可能となります。このため、形 CJ1W-EIP21S を含まないシステムと比べ、それを含むシステムでは CPU ユニットの運転開始時間が数秒遅くなります。これによる、システムの立ち上げ動作への影響を確認してください。

1-3-2　各部の名称と機能



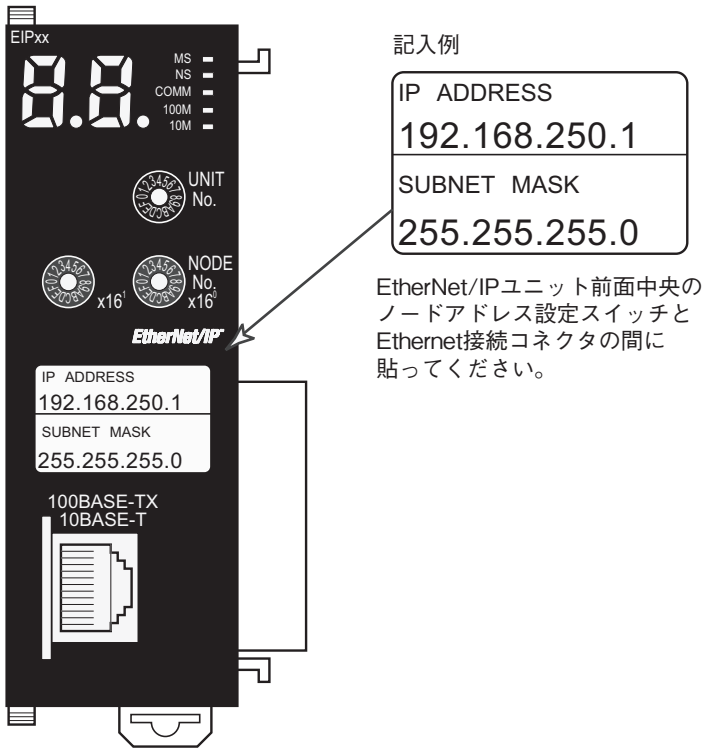
● Ethernet アドレスの表記

Ethernet に接続されるすべての通信機器には、固有の Ethernet アドレスが割り当てられています。EtherNet/IP ユニットでは、Ethernet アドレスはユニットの右側面に 12 桁の 16 進数で記載されています。



参考

ユーザが設定したIPアドレスとサブネットマスクの値を、ユーザが記入してユニット前面に貼ることができる「IP アドレス用ラベル」が標準添付されています。このラベルに記入して貼っておくことで、本ユニットのIPアドレスとサブネットマスクを容易に確認することができます。



表示部（LED）

EtherNet/IP ユニットでは、ユニット前面の以下の LED によって、ノードの状態やネットワークの状態を知ることができます。



● MS LED ／ NS LED ／ COMM LED ／ 100M LED ／ 10M LED

MS(Module Status)LED は、ユニットの状態を表示します。
NS(Network Status)LED は、ネットワークの状態を表示します。
COMM、100M、10M LED は、Ethernet 通信の状態を表示します。
MS、NS LED は、共に緑と赤の 2 色発光 LED です。COMM、100M、10M LED は黄の 1 色発光 LED です。これらの LED の色と点灯／点滅／消灯の意味は、次のとおりです。
なお、詳細については、🕮「第 15 章 異常とその処理」を参照してください。

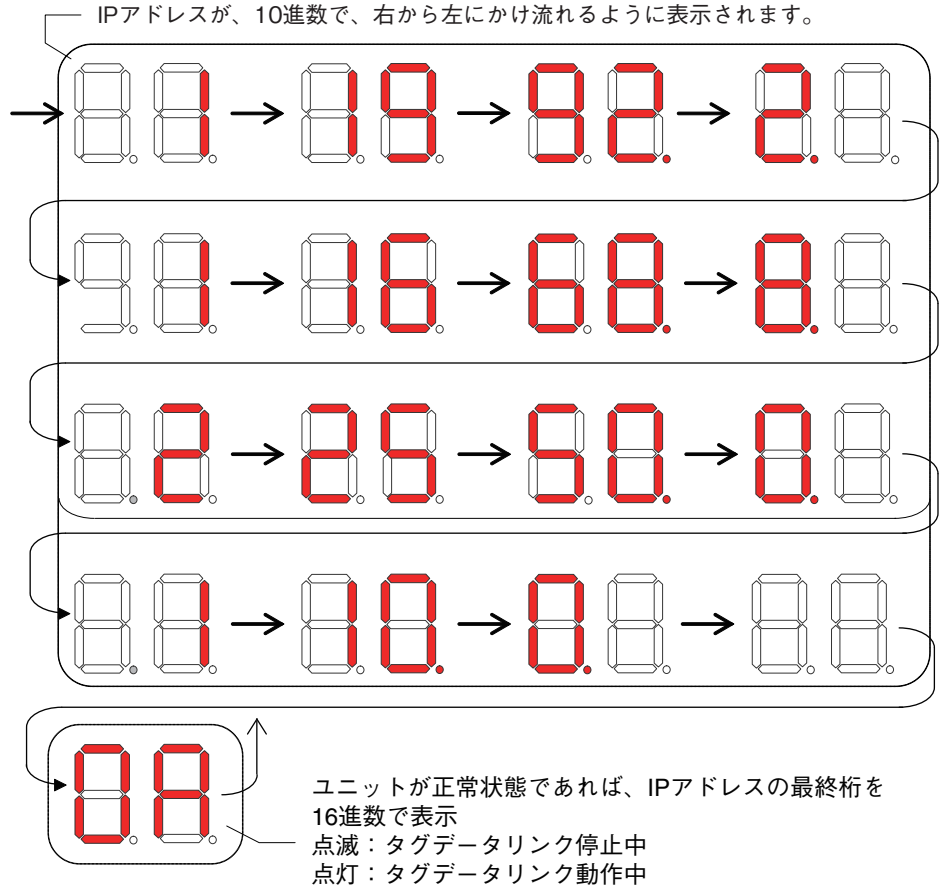
表示	名称	色	状態	動作状態
MS	Module Status	赤	点灯	致命的な故障
			点滅	回復可能な故障
		緑	点灯	正常状態
			点滅*1	IP アドレス未決定
		--	消灯	電源供給無し
NS	Network Status	赤	点灯	致命的な通信異常
			点滅	回復可能な通信異常
		緑	点灯	タグデータリンクおよびメッセージのコネクション確立あり
			点滅	タグデータリンクおよびメッセージのコネクション確立なし
		--	消灯	オフライン／電源供給無し
COMM	Communication	黄	点灯	データ送受信中
			消灯	送受信データなし
100M	100Mbps	黄	点灯	100BASE-TX のリンク確立
			消灯	100BASE-TX のリンク未確立
10M	10Mbps	黄	点灯	10BASE-T のリンク確立
			消灯	10BASE-T のリンク未確立

*1. 形 CJ1W-EIP21S の場合のみです。

● 7 セグメント LED 表示

電源 ON（リスタート）時に、全セグメントが 2 回点滅後、一度だけ、EtherNet/IP ユニットに設定されている IP アドレスを右から左にかけて流れるように表示します。その後、正常動作中は IP アドレスの下位 8 ビットを 16 進数で表示します。

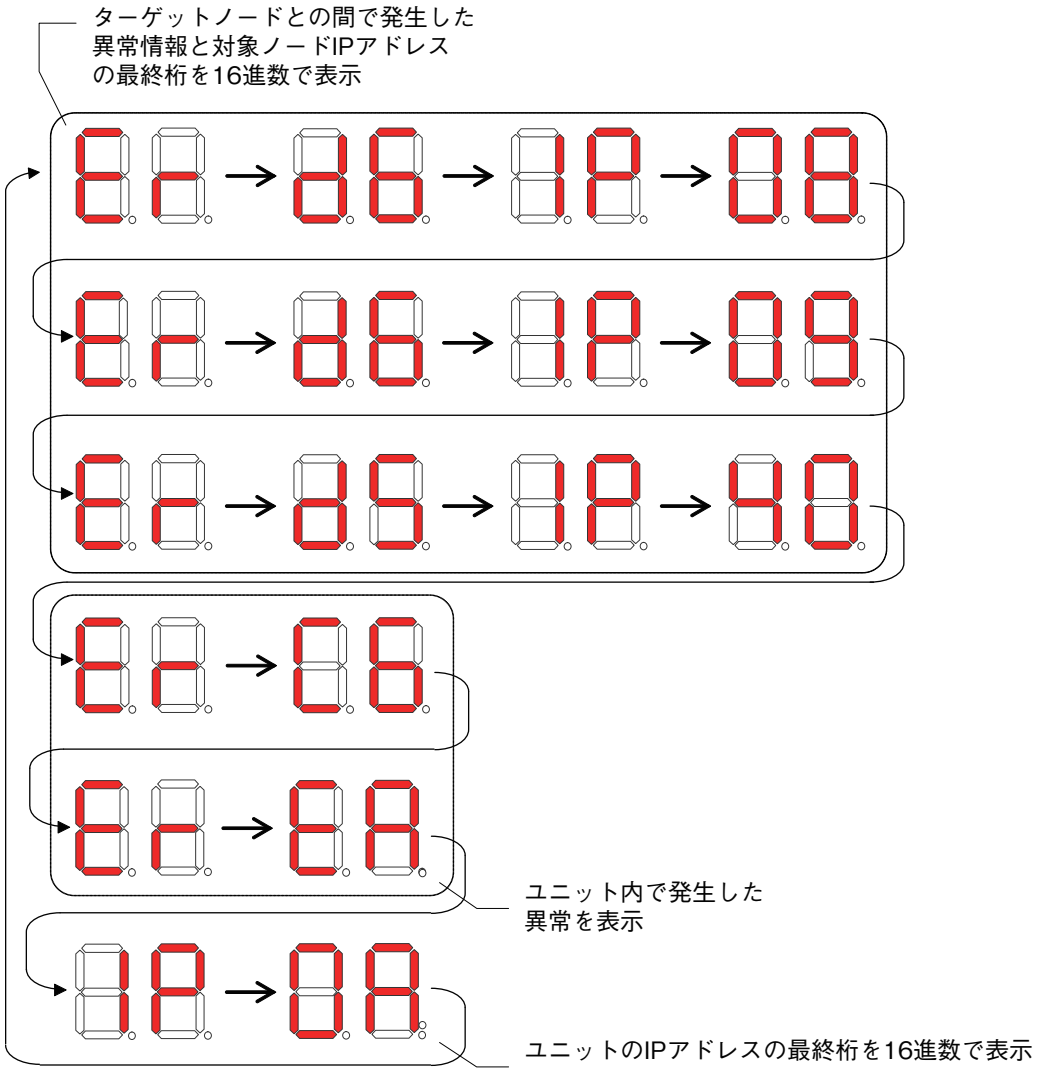
例）IP アドレスが 192.168.250.10 のとき。



エラーが発生したとき、エラーコード、異常デバイスの IP アドレスの下位 8 ビットが順次表示されます。エラーコードの詳細については、🕮「第 15 章 異常とその処理」を参照してください。

例）以下の複数の異常要因の表示例

- IP アドレス：192.168.250.8 との間で d6（コネクション確立失敗）が発生。
- IP アドレス：192.168.250.9 との間で d6（コネクション確立失敗）が発生。
- IP アドレス：192.168.250.64 との間で d5（照合異常（ターゲット不在））が発生。
- 自ユニット（192.168.250.10）で C6（複数スイッチ ON）と EA（EtherNet/IP 拡張設定異常）が発生。



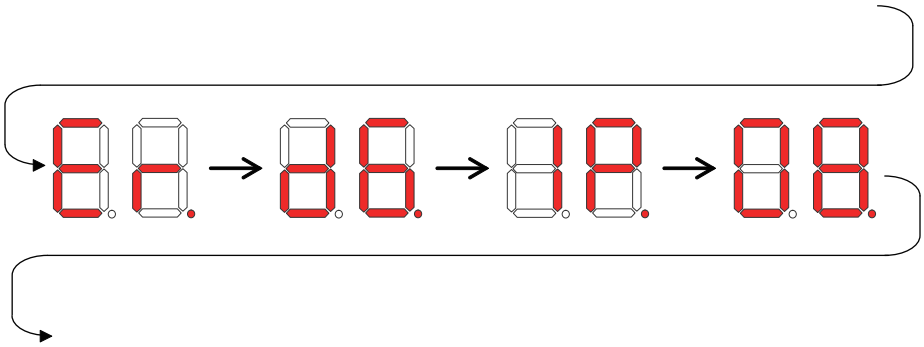
異常の表示には、特に優先順位はなく、発生している異常をすべて順番に繰り返します。

● 左右ドット LED

IP アドレス下位 8 ビットが同じ複数デバイスで異常が発生した場合は、該当ターゲットデバイス表示中に右側のドットを表示します。

例）以下の複数の異常要因の表示例

- IP アドレス：10.0.1.8 との間で d6（コネクション確立失敗）が発生。
- IP アドレス：10.0.2.8 との間で d6（コネクション確立失敗）が発生。



スイッチ設定

● ユニット番号設定スイッチ

ユニット番号設定スイッチでは、EtherNet/IP ユニットの CPU 高機能ユニットとしてのユニット番号を設定します。

設定方法	1 桁の 16 進数
設定範囲	0～F

（注）工場出荷時は 0 に設定されています。

同じ PLC に装着されている他の CPU 高機能ユニットとユニット番号が重複しない限り、範囲内で自由に設定することができます。

使用上の注意

- 小型のマイナスドライバを使用して、ロータリースイッチを傷つけないように設定してください。
- 設定は、必ずコントローラの電源が OFF の状態で行ってください。

参考

- 工場出荷時には 0 に設定されています。
- 同じコントローラに装着されている他の CPU 高機能ユニットとユニット番号が重複すると、CPU ユニットで No. 二重使用エラーが発生し、EtherNet/IP ユニットは起動できません。

● ノードアドレス設定スイッチ

ノードアドレス設定スイッチでは、EtherNet/IP ユニットのノードアドレスを設定します。



設定方法	2 桁の 16 進数
設定範囲	01 ～ FE

（注）工場出荷時は 01 に設定されています。
デフォルトでは、ここに設定した値が、以下のように、EtherNet/IP ユニットの自 IP アドレスの最下位桁となります。
デフォルト IP アドレス = 192.168.250. ノードアドレス
したがって、工場出荷時は、IP アドレス = 192.168.250.1 です。

ネットワーク内の他のノードとノードアドレスが重複しない限り、ノードアドレスは範囲内で自由に設定することができます。
ノードアドレスの設定については、🕮「2-1 ユニットの前面スイッチの設定」を参照してください。

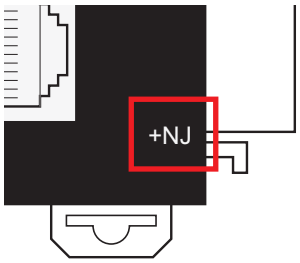
参考

- 動作中にノードアドレス設定スイッチを変更すると、MS LED が赤点滅します。

NJ 装着対応表示（形 CJ1W-EIP21S のロット番号 241001 □以降の場合のみ）

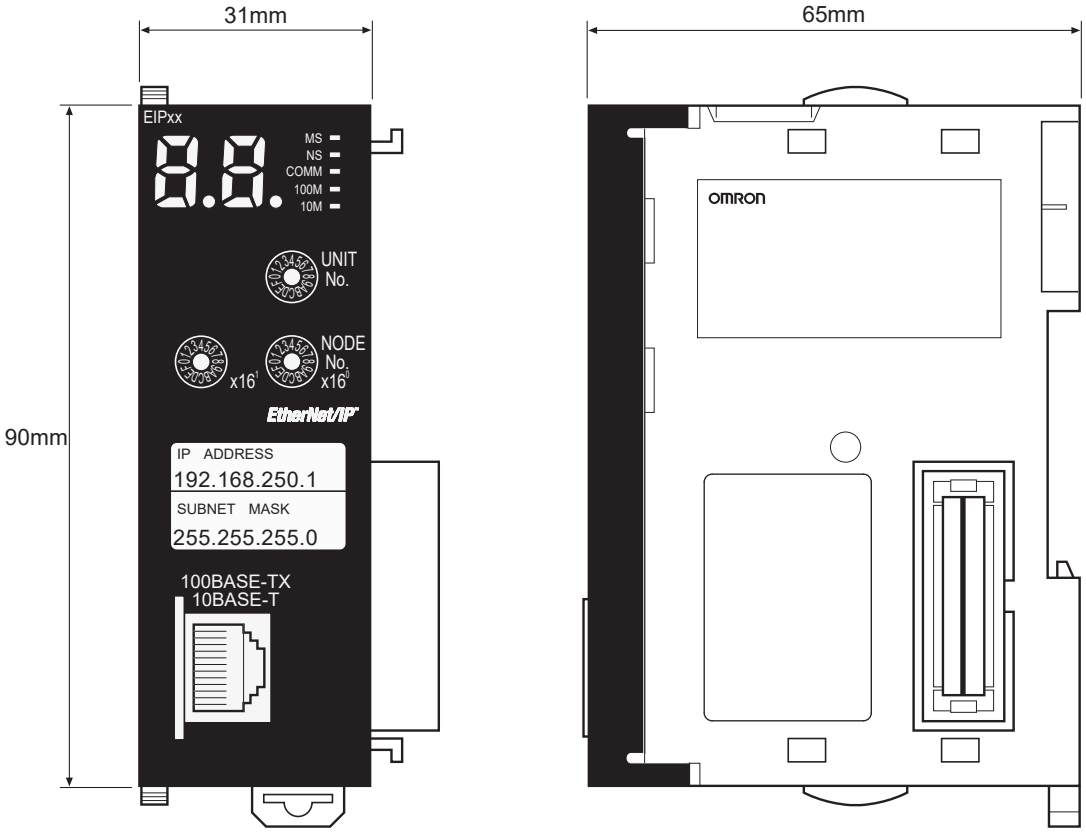
NJ シリーズ CPU ユニットに装着することができる形 CJ1W-EIP21S では、ユニット前面右下に「+NJ」または「+CJ/CP/NJ」と印字されています。形 CJ1W-EIP21S のロット番号 241001 □以降が該当します。

「+NJ」の場合の例）



1-3-3　外形寸法

• 形 CJ1W-EIP21/EIP21S



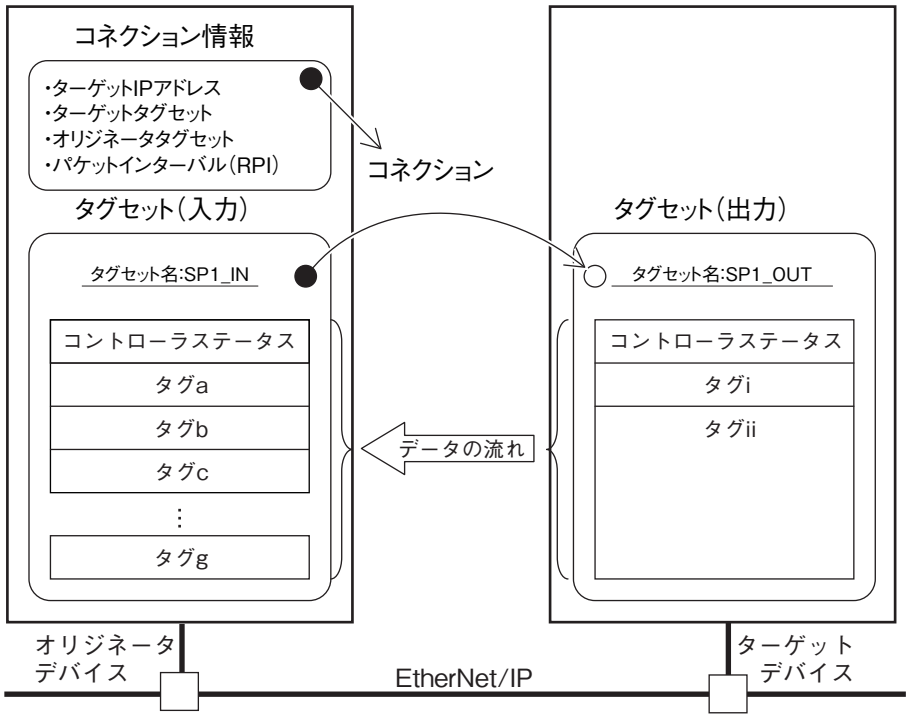
1-4　通信サービスの概要

1-4-1　CIP 通信（Common Industrial Protocol）サービス

タグデータリンク（サイクリック通信）

EtherNet/IP ネットワーク上のデバイスと、プログラムレスでサイクリックなデータ交換を行うことができます。
EtherNet/IP ユニットにおけるタグデータリンクでは、通常、Network Configurator で作成したタグをグループ化（タグセット）し、そのグループ単位でターゲットデバイスとのコネクションを開設し通信を行います。つまり、1 グループ（1 タグセット）で 1 コネクションを使用することになります。最大 256 コネクションまで登録できます。
EtherNet/IP ユニットのタグおよびタグセットの仕様は、以下のとおりです。

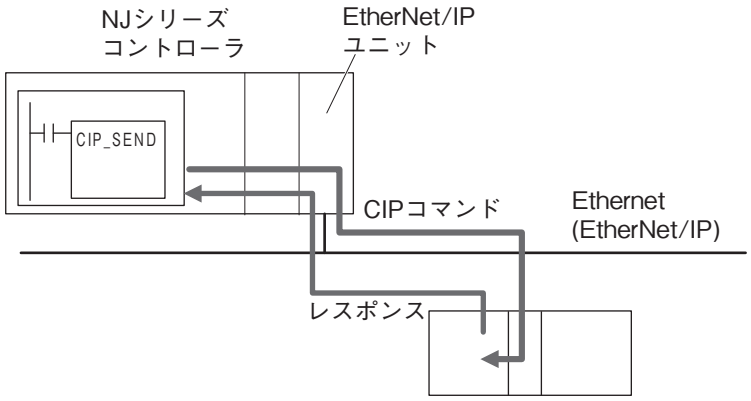
タグ	タグセット
全タグの合計サイズ≦ 184832CH	1 タグセットの最大サイズ≦ 722CH （タグセットにコントローラステータスを含める場合は≦ 721CH）
1 タグあたりの最大サイズ≦ 722CH （タグセットにコントローラステータスを含める場合≦ 721CH）	1 タグセットに含めるタグの数≦ 8 （タグセットにコントローラステータスを含める場合は 7） ※ 入力、出力の変数を混在させることはできません。
登録可能なタグの数≦ 256	登録可能なタグセットの数≦ 256



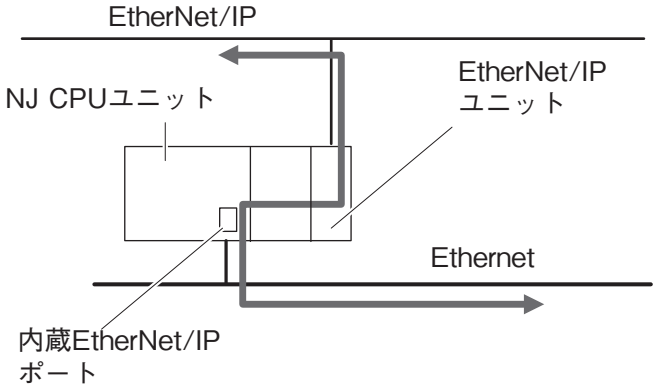
※ この例では、オリジネータのタグリスト a、b、c、…、g（ともに入力）を「SP1_IN」という名前でタグセットとし、ターゲットのタグリスト i、ii（ともに出力）を「SP1_OUT」という名前でタグセットにして、コネクションを設定しています。

CIP メッセージ通信

EtherNet/IP ネットワーク上のデバイスに対して、任意の CIP コマンドを送信することができます。
NJ シリーズ CPU ユニットのプログラム上で CIP 通信命令を実行することにより、データ読み書きなどの各種 CIP コマンドを発行し、レスポンスを受信します。



なお、ルートパスを指定することにより CIP メッセージ（CIP コマンド／レスポンス）を、EtherNet/IP ユニットを経由して、他の CIP ネットワークセグメントのデバイスに転送することも可能です（メッセージ通信の CIP ルーチング機能）。
CIP ルーチングのポート経由の階層越えは、CS/CJ/NJ シリーズのいずれの CPU の組合せでも、最大 8 階層越えです。



参考

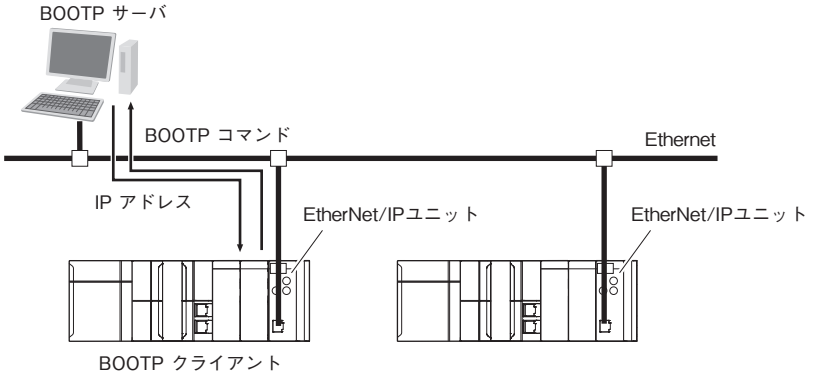
CIP ルーチングでは、ルーチングを行ったノード（ユニット）が、1 ホップ分のタイムアウトを減算し、ルート情報から自身のアドレスを削除して、次のノード（ユニット）にホップ（中継）します。
タイムアウトを指定する場合、ラストホップのときは、実際のリクエストサービス処理に対するタイムアウトを設定します。
中継ホップの場合は、リクエストに対するタイムアウトに加え、中継経路分のタイムアウトを加算する必要があります。
CIP 機能のあるオムロン製品では、1 ホップで 5 秒減算します。

バージョン関連情報

EtherNet/IP ユニット形 CJ1W-EIP21 は、CPU ユニットの Ver.1.01 以降、かつ Sysmac Studio Ver1.02 以降で使用可能です。
EtherNet/IP ユニット形 CJ1W-EIP21S は、CPU ユニットの Ver.1.67 以降、かつ Sysmac Studio Ver1.60 以降で使用可能です。

1-4-2　BOOTP クライアント機能

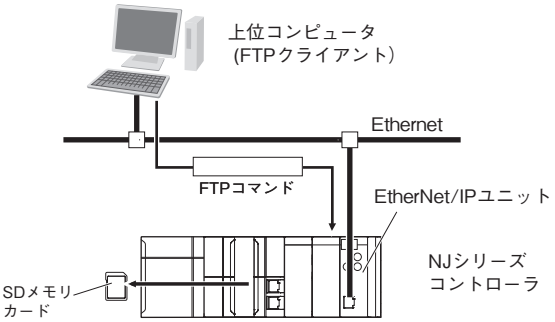
EtherNet/IP ユニットを BOOTP 設定にしておくことにより、EtherNet/IP ユニットの IP アドレスなどの設定を、BOOTP サーバから取得する機能です。

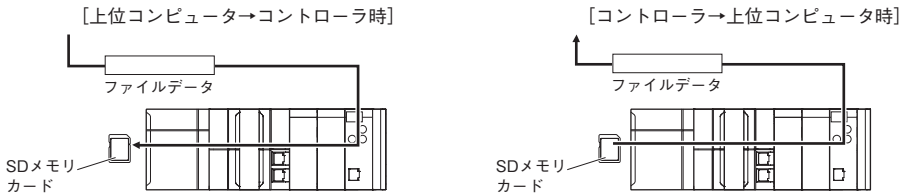


電源投入時、EtherNet/IPユニットのIPアドレスを、
BOOTPサーバから取得する

1-4-3　FTP サーバ機能

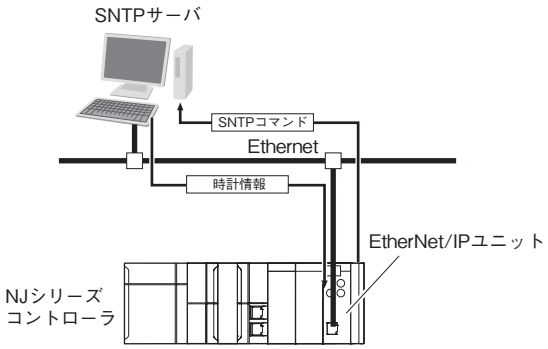
EtherNet/IP ユニットは FTP サーバ機能を内蔵しているため、Ethernet 上の他のコンピュータからコントローラの CPU ユニットの SD メモリカード内のファイルを読み書きできます。
これにより、上位コンピューターコントローラ間で、上位コンピュータが FTP クライアント、コントローラが FTP サーバとなった、ファイル単位のデータ交換が可能です。





1-4-4　時計自動調整機能

EtherNet/IP ユニットは、指定した時刻、または、時計情報自動調整実行スイッチの OFF → ON で、SNTP サーバから時間情報を取得し、CPU ユニットの内部時計情報を更新することができます。



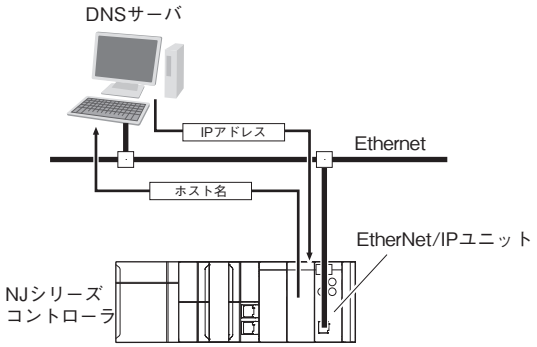
使用上の注意

時計自動調整機能を利用するためには、SNTP サーバが必要です。

1-4-5　ホスト名指定機能

SNTP サーバや SNMP マネージャの IP アドレス、CIP 通信命令での送信先について、IP アドレスではなく、ホスト名で指定することができます（DNS クライアント機能）。

例：DNS サーバでホスト名を設定する

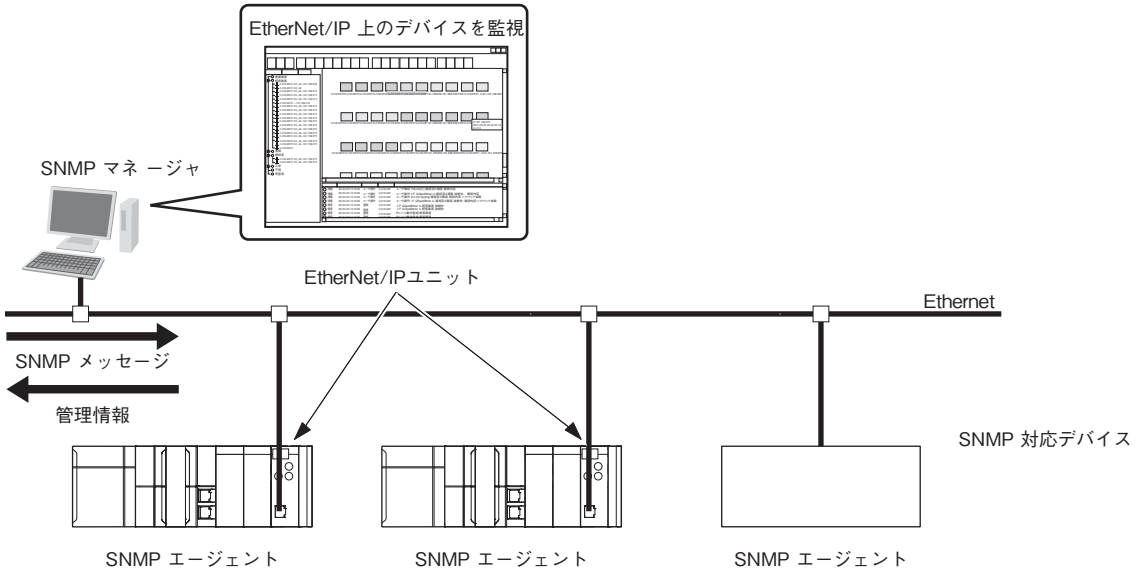


使用上の注意

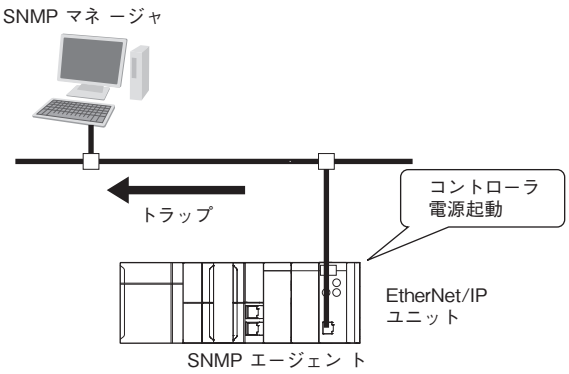
- DNS クライアント機能による、サーバホスト名指定機能を利用するためには、DNS サーバが必要です。

1-4-6　SNMP（エージェント）機能

SNMP マネージャを使用したネットワーク管理ソフトウェアに、EtherNet/IP ユニットの内部状態の情報を提供します。



- SNMP トラップ
  特定の状況発生時、SNMP エージェントに設定した EtherNet/IP ユニットが、状況通知レポートを SNMP マネージャに送信します。
  SNMP マネージャは，定期的に EtherNet/IP ユニットを監視しなくても、状況変化を知ることができます。
  状況通知レポートは、以下の状況のときに送信されます。
  - コントローラ電源投入時
  - リンク確立時
  - SNMP エージェントの承認失敗時



1-4-7　ソケットサービス機能（形 CJ1W-EIP21S のみ）

UDP または TCP プロトコルで、Ethernet 上の任意のノードと任意のデータの送受信が可能です。
ソケットサービスでのデータ送受信は、プログラム上でコマンド送信命令の実行、または形 CJ1W-EIP21S の I/O ポートに割り付けたデバイス変数の操作により、各通信処理を実行します。
通信相手機器とコネクションを開設した後、指定した変数の値を送信し、受信したデータを指定した変数に格納します。コネクションを閉じて通信を終了します。
また、TCP の場合はソケットの状態や受信データなどを読み出すこともできます。
TCP と UDP 合わせて計 8 ポートを使用できます。

1-5　EtherNet/IP 通信の使用手順

1-5-1　基本操作

EtherNet/IP ユニットの基本操作を示します。
プログラム作成およびユニットの設定には Sysmac Studio を使用します。
Sysmac Studio の操作については 🕮『Sysmac Studio Version 1 オペレーションマニュアル』（SBCA-470）を参照してください。

1　ユニット前面のロータリスイッチでユニット番号、ノードアドレスを設定
「2-1 ユニットの前面スイッチの設定」

↓

2　CPU ラックまたは増設ラックに装着
「2-2 CPU ラックまたは増設ラックへの装着」

↓

3　Ethernet ネットワーク（ツイストペアケーブル）の配線
「2-5 ネットワークへの接続」

↓

4　Sysmac Studio で、EtherNet/IP ユニットを登録
Sysmac Studio Version 1 オペレーションマニュアル（SBCA-470）

(1)　プロジェクトを新規作成。

(2)　ユニット構成に EtherNet/IP ユニットを登録。
［構成・設定］の［CPU・増設ラック］で、EtherNet/IP ユニットをユニット構成に登録。
このとき、登録するユニットのデバイス名とユニット番号を併せて設定します。

↓

5　Sysmac Studioで､I/OポートにCJデバイス変数を割り付けます。

- 既存の変数を選択して割り付け
- 新規に変数名を入力
- 「デバイス変数生成」で自動生成

↓

6　Sysmac Studio で､EtherNet/IP ユニットの初期設定データを作成
第 5 章 Sysmac Studio による EtherNet/IP ユニットのユニット設定一覧

- EtherNet/IP ユニットの［高機能ユニット設定の編集］でユニット設定をする。
- 必要に応じて、TCP/IP 設定、Ethernet 設定などを任意に設定する。

↓

7　Sysmac Studio で､EtherNet/IP ユニットの自 IP アドレスを設定
第 4 章 IP アドレスを決定する方法

自 IP アドレスを以下から設定。

- デフォルト：192.168.250.1（サブネットマスク= 255.255.255.0）
- 任意の IP アドレスを設定する
- BOOTP サーバから取得する

↓

8 コントローラの電源投入

コントローラの電源を投入します。
CPU ユニット内にユニット構成が存在し、それが実際のユニット構成と異なる場合、「I/O 設定チェックエラー」が発生します。この場合は手順 10 でコントローラをリセットするとエラーが解除されます。

↓

9 ユーザプログラムの転送

Sysmac Studio から CPU ユニットへユーザプログラム、ユニット構成・設定、変数情報を転送します。
この時点では高機能設定データは本ユニットに転送されていません。

↓

10 ユニット設定の反映

[高機能ユニット設定の編集] で設定されたシステム設定は、以下のいずれかの方法により、CPU ユニットから本ユニットへ転送されます。

- コントローラをリセットする
- EtherNet/IP ユニットをリスタートする
  - [高機能ユニット設定の編集] の [リスタート] ボタンをクリックする
  - CJ ユニットデバイス変数のユニットのリスタートフラグを、FALSE → TRUE する

リスタートフラグ

システム定義変数	型	R/W	名称	機能
_CJB_CBU** Restart	BOOL	RW	CPU 高機能ユニットリスタートフラグ	TRUE：CPU 高機能ユニットをリスタートする FALSE：リスタートしない 初期値：FALSE

（** にはユニット番号が入ります）

↓

11 パソコンから、PING コマンドを実行して、通信テスト

第 6 章 通信テスト

1-5-2　タグデータリンクを使用時

タグデータリンクを開始するまでの設定操作について示します。
タグデータリンクを行うためのタグデータリンクパラメータの設定は、Network Configurator を使用します。

1 **Sysmac　Studio で作成したタグに指定する変数の設定を Network Configurator にインポートする**
「7-2-4 タグ／タグセットの作成」

↓

2 **Network Configurator にてタグデータリンクテーブルの作成**
第 7 章 タグデータリンク機能

•ネットワーク構成の作成
•タグ、タグセット、コネクションの設定

↓

3 **Network Configurator のオンライン接続**
「7-2-8 Network Configurator のネットワーク接続手順」

↓

4 **タグデータリンク設定のダウンロード**
「7-2-9 タグデータリンクパラメータのダウンロード」

↓

5 **タグデータリンクの起動（電源 ON 時、自動で開始する）**
「7-2-12 タグデータリンクの開始／停止」

↓

6 **動作の確認**
1-3-2「各部の名称と機能」
第 15 章 異常とその処理

•EtherNet/IP ユニットの LED で確認
•Sysmac Studio を使用してシステム定義変数の全タグデータリンク通信中で通信状態を確認
•Network Configurator のモニタ機能でタグデータリンクが正常動作していることを確認

参考

タグデータリンクを行うターゲットノードのステータス情報について、
CJ ユニットデバイス変数とは別に、任意のユーザ定義変数に格納することができます。
📖3-5 項「ステータスエリアのユーザ定義設定」

1-5-3　EtherNet/IP ネットワークの各機能を使用する

● メッセージ通信サービスを使用時

• CIP 通信命令

1 **ユーザプログラム上で、CIP 通信命令を実行**
第 8 章 メッセージ通信機能

↓

2 **動作の確認**
1-3-2「各部の名称と機能」
第 15 章 異常とその処理

•Sysmac Studio を使用して命令の終了コード（Done, Err, ErrorID）で通信状態を確認

● FTP サーバ機能を使用時

1 Sysmac Studio で、EtherNet/IP 機能モジュールを初期設定

第 9 章 FTP サーバ機能

•FTP 設定（FTP 有効、ログイン名、パスワード）

↓

2 FTP クライアントソフトから FTP サーバ（NJ シリーズ CPU ユニット）への接続

•FTP のログイン名およびパスワードを入力して、EtherNet/IP ユニットにログイン
•FTP サーバ起動のイベントログを確認

● 時計情報自動調整機能を使用時

1 Sysmac Studio で、EtherNet/IP 機能モジュールを初期設定

第 10 章 時計情報自動調整機能

•SNTP 設定（SNTP 有効、実行条件）

↓

2 時刻情報自動調整の実行

•指定時刻で、自動調整が実行されます。

● SNMP（エージェント）機能を使用時

1 Sysmac Studio で、EtherNet/IP 機能モジュールを初期設定

第 11 章 SNMP（エージェント）機能

•SNMP 設定
•SNMP トラップ 設定

↓

2 動作の確認

•SNMP エージェント起動のイベントログを確認

● BOOTP 機能を使用時

1 Sysmac Studio で、EtherNet/IP 機能モジュールを初期設定

第 5 章 Sysmac Studio による EtherNet/IP ユニットのユニット設定一覧

•BOOTP 設定

↓

2 動作の確認

•BOOTP 起動のイベントログを確認
•CJ ユニットデバイス変数のオンラインを確認

● ソケットサービス機能を使用時

1　Sysmac Studio で、EtherNet/IP 機能モジュールを初期設定

第 5 章 Sysmac Studio による EtherNet/IP ユニットのユニット設定一覧

•ステータスエリア設定

↓

2　ユーザプログラムでソケットサービス機能を実行

第 13 章 ソケットサービス機能

•ソケットサービスパラメータの設定
•プログラム上でコマンド送信命令の実行、またはデバイス変数の操作を実行するように記述

↓

3　動作の確認

第 13 章 ソケットサービス機能 3-2-5「ソケットサービス関連の CJ ユニットデバイス変数（形 CJ1W-EIP21S のみ）」

•ソケットステータスを確認
•ソケットサービスパラメータエリアの終了コードを確認

2

Ethernet ネットワークの配線

2-1 ユニットの前面スイッチの設定 2-2

2-2 CPU ラックまたは増設ラックへの装着 2-3

2-3 ネットワーク構成機器の選定 2-4
2-3-1 ネットワーク構成機器の推奨品 2-4
2-3-2 スイッチングハブの種類 2-5
2-3-3 スイッチングハブの機能 2-5
2-3-4 スイッチングハブ選定時の留意事項 2-6

2-4 ネットワークの布設について 2-8
2-4-1 ネットワーク布設時の基本的注意事項 2-8
2-4-2 ネットワーク構成機器の推奨品 2-8
2-4-3 ツイストペアケーブル布設時の注意事項 2-8
2-4-4 スイッチングハブの設置・接続時の注意事項 2-10

2-5 ネットワークへの接続 2-11
2-5-1 Ethernet 接続コネクタ 2-11
2-5-2 ケーブルの接続方法 2-11
2-5-3 リレー接点出力ユニット使用時のノイズ対策 2-12

2-1　ユニットの前面スイッチの設定

● ユニット番号の設定

同一 CPU ユニットに装着されている他の CPU 高機能ユニットのユニット番号と重ならないように、ユニット番号を設定してください。
小型のドライバを使用して、ロータリスイッチを傷つけないように設定してください。工場出荷時には、0 に設定されています。



設定範囲
0 ～ F

使用上の注意

- 電源 OFF 状態で設定してください。
- Sysmac Studio のユニット構成の設定で、EtherNet/IP ユニットのユニット番号を設定した場合、ロータリスイッチ側の設定も合わせて設定を変更してください。
  ユニット構成の設定と、ロータリスイッチ側の設定が一致していない場合、「I/O 設定チェックエラー」が発生します。

● ノードアドレスの設定

ノードアドレスは、FINS 通信サービスで、個々の EtherNet/IP ユニット（ノード）を識別するときに使用します。
ノードアドレスは、同一 Ethernet ネットワークに接続されている他の EtherNet/IP ユニットや Ethernet ユニットのノードアドレスと重ならないように、ノードアドレス設定スイッチに 16 進数で設定します。
Ethernet ネットワーク上で FINS 通信サービスを使用しない場合は、他の EtherNet/IP ユニットと重複しても構いません。

ただし、ノードアドレス設定時は、以下の条件に合わせて設定してください。
条件に合わないと、エラー表示となり（MS LED ：赤点滅、7 セグメント LED：「H4」（ノードアドレス設定異常）を表示）、EtherNet/IP ユニットは動作を停止します。

- 01 ～ FE の範囲で設定してください。
- ロータリスイッチで設定したノードアドレスと、EtherNet/IP ユニットの自 IP アドレスの下位桁（ホスト ID 部）は、同じ値にしてください。

もし、ノードアドレスと EtherNet/IP ユニットの自 IP アドレスの下位桁を、同じ値で設定できない場合は、アドレス変換方式を「IP アドレステーブル方式」または「併用方式」に設定してください。詳しくは、🕮「第 5 章 Sysmac Studio による EtherNet/IP ユニットのユニット設定一覧」を参照してください。



設定範囲
01 ～ FE（10 進数 1 ～ 254）

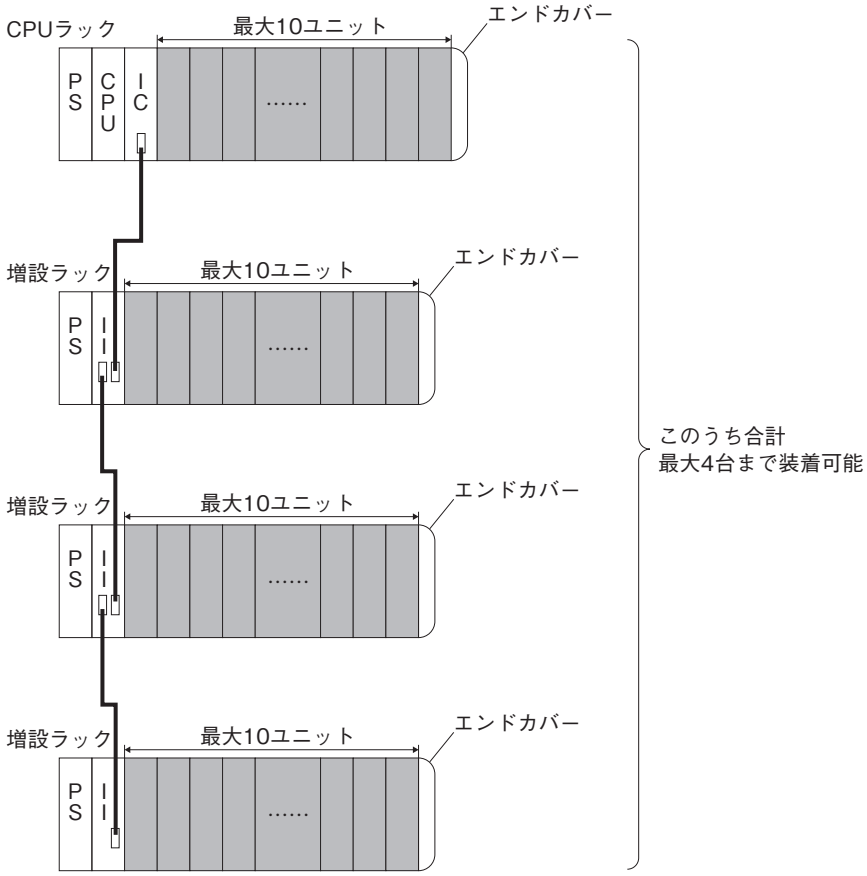
上のロータリスイッチで上位桁を、下のロータリスイッチで下位桁を設定します。
工場出荷時には、01 に設定されています。

使用上の注意

電源 OFF 状態で設定してください。

2-2　CPU ラックまたは増設ラックへの装着

CJ シリーズ EtherNet/IP ユニットは、NJ シリーズ CPU ラックと CJ シリーズ 増設ラックに接続できます。
以下のいずれかの位置にユニット上部および下部のスライダで固定してください。
1 台の PLC には最大 4 台までの EtherNet/IP ユニットが装着可能です。
なお、他の CPU 高機能ユニットと混在して装着するときは、CPU 高機能ユニットの最大台数 16 台を超えないように注意してください。



（注）PS ：電源ユニット
CPU：CPUユニット
IC ：I/Oコントロールユニット
II ：I/Oインタフェースユニット

参考

EtherNet/IP ユニットの最大消費電流は次のとおりです。
• 形 CJ1W-EIP21 の消費電流は、最大時 410mA です。
• 形 CJ1W-EIP21S の消費電流は、最大時 620mA です。
同じ CPU ラックまたは増設ラックに接続されているユニットの合計消費電流が、電源ユニットの出力容量の上限を超えないように注意してください。

2-3　ネットワーク構成機器の選定

2-3-1　ネットワーク構成機器の推奨品

EtherNet/IP を使用する場合のネットワーク構成機器には、以下の製品を推奨します。

● スイッチングハブ

メーカ	形式	機能
オムロン（株）	形 W4S1-03B 形 W4S1-05B 形 W4S1-05C 形 W4S1-05D	優先度制御（QoS）：EtherNet/IP の制御データ優先 故障検知：ブロードキャストストーム・LSI 異常検知、100BASE-TX/10BASE-T、Auto-Negotiation ポート数 （形 W4S1-03B：3、形 W4S1-05B：5、形 W4S1-05C：5、形 W4S1-05D：5） 故障検知機能（形 W4S1-05C のみ）
シスコシステムズ	メーカにお問い合わせください。 http://www.cisco.com/web/JP/index.html	
（株）コンテック	メーカにお問い合わせください。 http://www.contec.co.jp/	
フェニックスコンタクト（株）	メーカにお問い合わせください。 https://www.phoenixcontact.com/online/portal/jp	

● ツイストペアケーブル、コネクタ

使用する伝送速度により、使用可能なケーブルとコネクタが決まります。
カテゴリ 5 以上の STP（シールドツイストペア）ケーブルを使用します。ストレートケーブルまたはクロスケーブルのいずれも使用できます。
EtherNet/IP の通信ケーブルに使用する配線材料は下表のとおりです。
下表の商品名称で、EtherNet/IP の 100BASE-TX と記載しているときは 100BASE-TX と 10BASE-T のいずれも使用可能であることを示します。

商品名称		メーカ	形式
サイズ・線心数（対数）：AWG24 × 4P	ケーブル	東日京三電線株式会社	NETSTAR-C5E SAB 0.5 × 4P CP
		倉茂電工株式会社	KETH-SB
	RJ45 コネクタ	パンドウイットコーポレーション	MPS588
サイズ・線心数（対数）：AWG22 × 2P *1	ケーブル	倉茂電工株式会社	KETH-PSB-OMR
		日本電線工業株式会社	PNET/B
	RJ45 組立式コネクタ 	オムロン株式会社	形 XS6G-T421-1
サイズ・線心数（対数）：0.5mm × 4P	ケーブル	株式会社フジクラ	F-LINK-E 0.5mm×4P
	RJ45 コネクタ	パンドウイットコーポレーション	MPS588

*1. ケーブルおよびコネクタは、上記の組み合わせでのご使用を推奨します。

● ブーツ

メーカー	形式	お問い合わせ先電話番号
通信興業（株）	MK ブーツ (IV)LB	通信興業（株）

使用上の注意

- タグデータリンクを使用するネットワークでは、必ずスイッチングハブを使用してください。
- リピータハブを使用して EtherNet/IP のタグデータリンク（サイクリック通信）を行うと、ネットワークの通信負荷が高まるため、コリジョン（衝突）が多数発生し、安定した通信ができなくなります。

2-3-2　スイッチングハブの種類

● アンマネージドレイヤ 2（L2）スイッチングハブ

Ethernet アドレスによりスイッチングする機能をもつスイッチングハブ。一般的なスイッチングハブは、この機能を持ちます。スイッチングハブが持つ機能の切り替えや設定値の変更は行えません。

● マネージドレイヤ 2（L2）スイッチングハブ

Ethernet アドレスによりスイッチングする機能を持ち、ネットワーク上のスイッチングハブ専用ツールにより機能の切り替えや設定値の変更を行え、診断情報の収集などを行うことができます。アンマネージドレイヤ 2 スイッチングハブに比べ高機能なスイッチングハブになります。

2-3-3　スイッチングハブの機能

EtherNet/IP ネットワークで使用する際にポイントとなるスイッチングハブ機能の概要を以下に示します。

EtherNet/IP ユニット使用時は、以下の 2 つの機能の有無に分けて選定してください。

- マルチキャストフィルタ機能の有無
- TCP/UDP ポート番号 (L4)QoS(Quality of Service) 機能の有無

● マルチキャストフィルタ

マルチキャストパケットを特定のノードのみに転送する機能です。スイッチングハブに、IGMP Snooping または GMRP として実装されます。

ここで言う特定のノードとは、IGMP クライアント機能を持ちスイッチングハブに転送要求を依頼したノードを指します（オムロン製の EtherNet/IP ユニットは、IGMP クライアント機能をもちます）。

この機能がない場合、マルチキャストパケットは、ブロードキャストパケット同様、すべてのノードに転送されてしまい、ネットワーク上のトラフィックを増大させる原因になります。

この機能を有効にするには、スイッチングハブに対する設定が必要になります。マルチキャストフィルタ数は、使用するネットワークに対し十分である必要があります。

● TCP/UDP ポート番号 (L4) での QoS(Quality of Service)

特定の IP アドレス、TCP(UDP) ポート宛のパケットを優先的に転送するなど、パケット転送の優先制御を行う機能です。TCP、UDP プロトコルがトランスポート層プロトコルであることから、L4（レイヤ 4）QoS 機能と呼びます。

タグデータリンクとメッセージ通信を同一ネットワーク上で実行する場合、タグデータリンクパケットを優先的に転送することで、メッセージ通信のトラフィックによる転送遅延、バッファオーバーフローによるパケット破棄などの影響を避けることが可能となります。

この機能を有効にし、タグデータリンクパケットを優先的に転送させるには、スイッチングハブに対する設定が必要になります。

これらの機能と各スイッチングハブでのサポート状況を以下にまとめます。

スイッチングハブの種類	マルチキャストフィルタ	L4 QoS	備考
アンマネージド L2 スイッチングハブ	なし	なし	−
マネージド L2 スイッチングハブ	あり	あり	両機能ともツールによる設定が必要
オムロン製スイッチングハブ（W4S1 シリーズ）	なし	あり	L4 QoS の設定はスイッチ切替え実行。ツール不要。

参考

Network Configurator によるコネクション設定にてコネクションタイプとして「Multi-cast connection」を選択するとマルチキャストパケットが使われます。コネクションタイプとしてすべて「Point to Point connection」を選択するとマルチキャストパケットは使われません。

2-3-4　スイッチングハブ選定時の留意事項

スイッチングハブがサポートしている機能は、タグデータリンクの転送遅延やコントローラ構成・設定内容などに影響を及ぼします。
また、スイッチングハブが高度な機能をサポートしている場合には、その設定が必要になります。
スイッチングハブ選定の際には、ネットワーク上でどのような通信をどれくらいの負荷で使用するかにより、スイッチングハブを選定する必要があります。
以下の注意事項を参考にして、スイッチングハブを選定してください。
タグデータリンク通信負荷の見積り方法は 🕮「14-2 通信負荷の調整」を参照ください。

ネットワーク上で実行する通信種類に着目して選定するとき

● タグデータリンクのみ実行する場合

マルチキャストフィルタなしの L2 スイッチングハブ、または、マルチキャストフィルタ付きの L2 スイッチングハブの使用を推奨します。
マルチキャストフィルタ機能付きの場合、不必要なマルチキャストパケットによるトラフィックの増大を防げるため、より高速なタグデータリンクが可能となります。
ただし、以下のいずれかの条件に当てはまる場合、マルチキャストフィルタなし・付きによるトラフィックの差異はありません。

- ネットワーク上の全ノード間で同じデータを共有するタグデータリンク設定を行う場合（マルチキャストパケットがブロードキャスト同様に全ノードへ転送される）
- タグデータリンク設定がすべて 1 対 1（ユニキャスト）であり、マルチキャストパケットが使われない場合

マルチキャストフィルタ機能を使用する場合、スイッチングハブに対する設定が必要になります。
マルチキャストフィルタ数は、使用するネットワークに対し十分である必要があります。

● タグデータリンクとメッセージ通信を混在して実行する場合

マルチキャストフィルタ付き、かつ L4 QoS 付きの L2 スイッチングハブの使用を推奨します。
タグデータリンク用パケットを優先的に転送するように設定することにより、メッセージ通信のトラフィックによる転送遅延、およびバッファオーバーフローによるパケット破棄などの影響を避けることが可能となります。
マルチキャストフィルタ機能、および L4 QoS 機能を使用する場合、スイッチングハブに対する設定が必要になります。

スイッチングハブのサポート機能に着目して選定するとき

● マルチキャストフィルタ機能なしの L2 スイッチングハブを使用する場合

タグデータリンクのみ行い、以下のいずれかの条件に当てはまる場合に、推奨します。

- ネットワーク上の全ノード間で同じデータを共有するタグデータリンク設定を行う場合（マルチキャストパケットがブロードキャスト同様に全ノードへ転送される）
- タグデータリンク設定がすべて 1 対 1（ユニキャスト）であり、マルチキャストパケットが使われない場合
- タグデータリンクによるトラフィックが少ない場合

マルチキャストフィルタ機能なしの L2 スイッチングハブは、スイッチングハブに対する設定は不要です。

● マルチキャストフィルタ機能付きの L2 スイッチングハブを使用する場合

タグデータリンクのみ行い、以下の条件に当てはまる場合に推奨します。

- タグデータリンク設定で 1 対 N（N はネットワーク上の一部のノード）が多い、（つまり、マルチキャストパケットが多く使われる）、またはタグデータリンクによるトラフィックが多い場合

マルチキャストフィルタ機能付きの L2 スイッチングハブは、スイッチングハブに対する設定が必要です。マルチキャストフィルタ数は、使用するネットワークに対し十分である必要があります。

● マルチキャストフィルタ機能付き、かつL4 QoS機能付きのL3スイッチングハブを使用する場合

タグデータリンクおよびメッセージ通信を混在して実行する場合に、推奨します。
タグデータリンク用パケットを優先的に転送するように設定することにより、メッセージ通信のトラフィックによる転送遅延、バッファオーバーフローによるパケット破棄などの影響を避けることが可能となります。
マルチキャストフィルタ機能、L4 QoS 機能を使用する場合、スイッチングハブに対する設定が必要になります。マルチキャストフィルタ数は、使用するネットワークに対し十分である必要があります。

使用上の注意

- スイッチングハブに対する設定方法の詳細につきましては、スイッチングハブメーカにお問い合わせください。
- スイッチングハブの設置につきましては、耐環境性能を十分ご考慮の上、設置してください。耐環境性能の詳細につきましては、スイッチングハブメーカにお問い合わせください。

2-4　ネットワークの布設について

2-4-1　ネットワーク布設時の基本的注意事項

- Ethernet 布設工事を行うときは、十分な安全対策と規格に従った布設が必要です。Ethernet 布設には「JIS X5252」、布設工事には「電気設備技術基準」を参照してください。
- 布設工事は、安全対策、規格に詳しい専門業者に依頼されることをお勧めします。
- Ethernet ネットワーク機器は、ノイズを発生する機器の近くに布設しないでください。
  万が一、ノイズの多い環境に布設する場合は、各ネットワーク構成機器を金属ケースに収納したり、部分的に光ケーブルを使用するなどのノイズ対策を必ず施してください。
- ケーブルの両端ともにシールドをコネクタフードに接続されたシールドケーブルを使用する場合、機器の接地方法によってはグランドループができるため、ノイズ耐量が低下したり、機器が破壊したりすることがあります。機器の接地間の電位差によるグランドループを防止するため、機器間の基準電位を安定させる必要があります。機器間の接地線にノイズ電流が流れないように接地方法を検討してください。接地方法については、🕮『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）』および🕮『NX 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-418）』を参照してください。
- EtherNet/IP のケーブル布設資料の入手については、ODVA へお問い合わせください。
  ODVA のホームページ：http://www.odva.org
- 情報系ネットワークと混在して EtherNet/IP ネットワークを布設する際、タグデータリンクによって通信負荷が高くなることが想定される場合は、情報系ネットワークと別セグメントにするなどして影響を与えないネットワーク構成にすることをお勧めします。

2-4-2　ネットワーク構成機器の推奨品

EtherNet/IP ユニットを使用する場合のネットワーク構成機器については、🕮「2-3 ネットワーク構成機器の選定」を参照してください。

2-4-3　ツイストペアケーブル布設時の注意事項

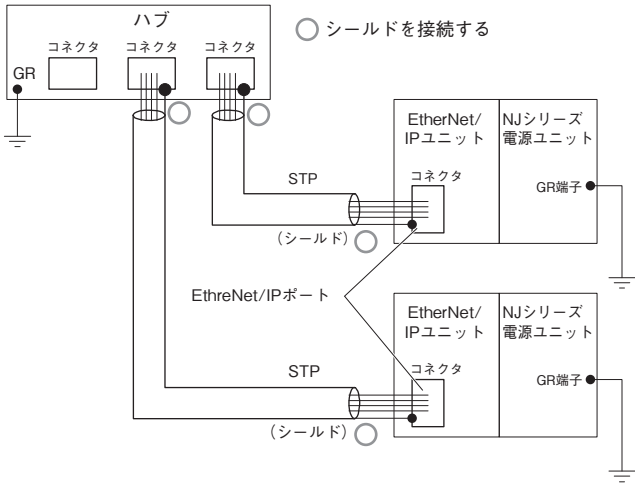
シールドのコネクタフードへの接続処理

● EtherNet/IP ポートースイッチングハブ間

ケーブルのシールドの各コネクタフードへの接続については、以下の（a）、（b）のいずれかとしてください。

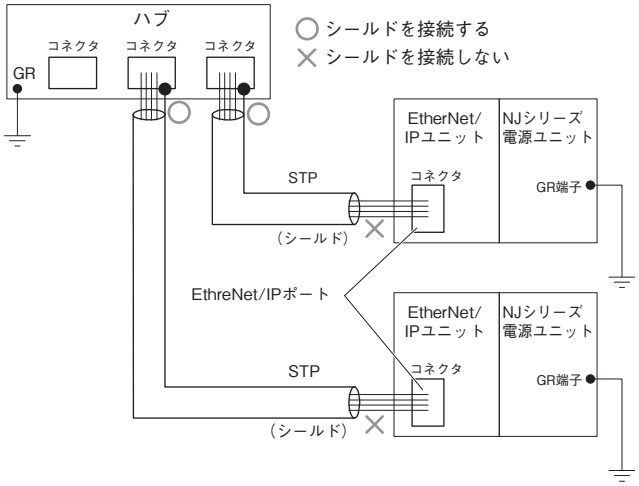
(a) 両端接続する

ケーブルの両端ともに、シールドを各コネクタフードへ接続します。



(b) スイッチングハブ側のみ接続する

ケーブルの片端のスイッチングハブ側のみ、シールドをコネクタフードへ接続します。



参考

シールドの処理や機器の接地方法によってはグランドループができるため、ノイズ耐量が低下したり、機器が破壊したりする場合があります。
(b) のスイッチングハブ側のみ接続をすることで、(a) の両端接続をするよりも、改善できる場合があります。

● スイッチングハブースイッチングハブ間

使用する伝送速度によらず、スイッチングハブースイッチングハブ間のネットワークの布設については、ケーブルのシールドを各コネクタフードへ接続するか否かを含め、スイッチングハブのメーカに確認してください。

ツイストペアケーブル布設時のその他の注意事項

- スイッチングハブおよび EtherNet/IP ユニットに接続するときは、ロックするまで確実にコネクタに差し込んでください。
- ツイストペアケーブルは、高圧電線と区別して、施設、配線してください。
- ノイズを発生させる機器の近くには、布設しないでください。
- 高温多湿の環境に布設しないでください。
- 粉塵、オイルミストなどのない場所で使用してください。

2-4-4　スイッチングハブの設置・接続時の注意事項

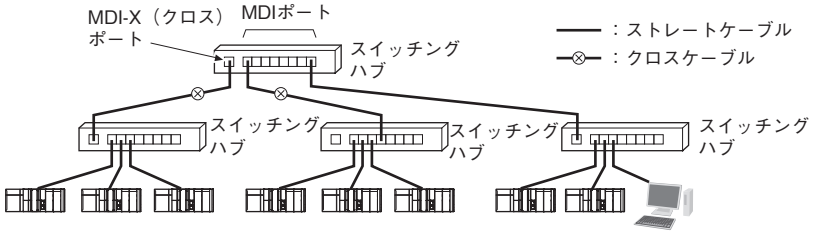
スイッチングハブ設置時の注意点

- スイッチングハブは、インバータなどの駆動系の機器と同一箇所にならないように接地してください。
- スイッチングハブへの電源は必ず専用の電源としてください。I/O 用電源、モータ用電源、および制御用電源などと、同一の電源を使用しないでください。
- スイッチングハブの耐環境性能仕様を十分ご考慮の上、使用環境に適合するスイッチングハブを設置してください。スイッチングハブの耐環境性能の詳細につきましては、スイッチングハブメーカにお問い合わせください。

スイッチングハブの接続方法

- ツイストペアケーブルによるスイッチングハブ間接続は、
  MDI ポートと MDI-X ポートをストレートケーブルで接続します。
  MDI ポート、MDI-X ポート同士は、クロスケーブルで接続します。

（注） クロスケーブルとストレートケーブルは、外観で区別することが難しく、誤って使用した場合、通信不良の原因となります。基本的にストレートケーブルで使用することをお勧めします。



- スイッチングハブによっては、自動で MDI/MDI-X を検出するものがあります。この場合はスイッチングハブ間もストレートケーブルが使用できます。

使用上の注意

EtherNet/IP ユニットのリンク設定は、接続するスイッチングハブの通信モード設定と合致させるようにしてください。合致していない場合、不安定なリンク状態となり、正常な通信ができません。スイッチングハブの通信モード別の設定表を下記に示します。
（AUTO-Nego：オートネゴシエーション、FULL：全二重、HALF：半二重を意味します）

スイッチングハブ		EtherNet/IP ユニット				
		AUTO-Nego	10M 固定		100M 固定	
			FULL	HALF	FULL	HALF
AUTO-Nego		◎	×	○	×	○
10M 固定	FULL	×	○	×	×	×
	HALF	○	×	○	×	×
100M 固定	FULL	×	×	×	◎	×
	HALF	○	×	×	×	○

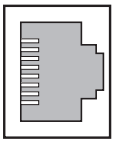
（◎：推奨、○：接続可能、×：接続不能）

2-5　ネットワークへの接続

2-5-1　Ethernet 接続コネクタ

Ethernet のツイストペアケーブルを接続するコネクタです。

- 電気的特性：IEEE802.3 規格に準拠
- コネクタ機構：RJ45 の 8 ピン・モジュラ・コネクタ（ISO 8877 準拠）

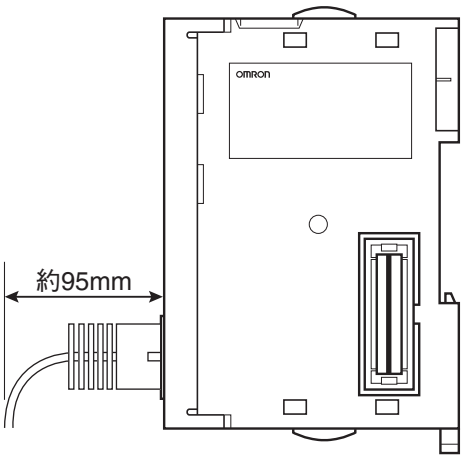


ピン番号	信号名称	略称	信号方向
1	送信データ＋	TD ＋	出力
2	送信データ－	TD －	出力
3	受信データ＋	RD ＋	入力
4	未使用	－	－
5	未使用	－	－
6	受信データ－	RD －	入力
7	未使用	－	－
8	未使用	－	－
コネクタフード	保安用接地	FG	－

2-5-2　ケーブルの接続方法

使用上の注意

- ツイストペアケーブルの接続や脱着は、コントローラの電源が OFF の状態で行ってください。
- ツイストペアケーブルの曲げ半径を確保するため、余裕を持ったスペースを取ってください。使用する通信ケーブルやコネクタにより必要なスペースが異なりますので、各メーカまたは購入元にお問い合わせください。



(1)　ツイストペアケーブルを布設します。

(2)　スイッチングハブにツイストペアケーブルを接続します。

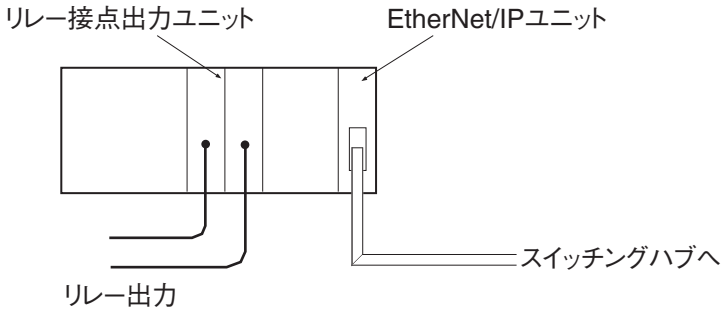
(3)　ツイストペアケーブルを EtherNet/IP ユニットのコネクタに取り付けます。
スイッチングハブおよび Ethernet に接続する際は、ロックするまで確実にコネクタを差し込んでください。

2-5-3　リレー接点出力ユニット使用時のノイズ対策

EtherNet/IP ユニットとリレー接点出力ユニットを、1 つの PLC に同時に装着すると、リレー接点出力ユニットのリレーが発するノイズで、通信エラーが発生することがあります。設置時に、次の対策をとってください。

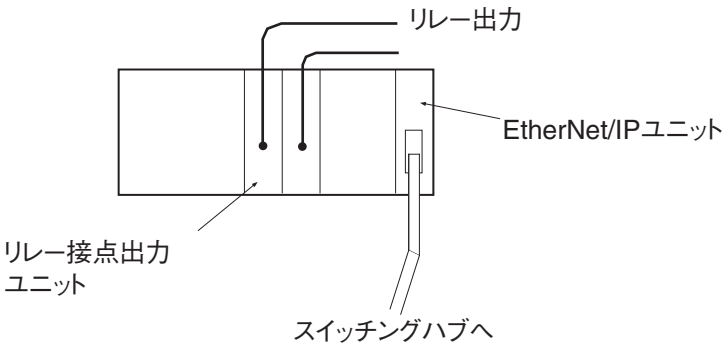
● ユニットの取付け位置を離す

EtherNet/IP ユニットとリレー接点出力ユニットは、できるだけ離して EtherNet/IP ユニットを接続してください。



● ケーブルを離して配線する

ツイストペアケーブルと、リレー接点出力ユニットの配線を分離すると効果があります。



3

CJユニットデバイス変数への割付

3-1 CPU ユニットとのデータ交換 3-2
3-1-1 データの流れ 3-2
3-1-2 各種変数の指定／作成方法 3-5
3-2 CJ ユニットデバイス変数の詳細 3-6
3-2-1 CJ ユニットデバイス変数への割付 3-6
3-2-2 ソフトスイッチ用の CJ ユニットデバイス変数 3-6
3-2-3 ステータス用の CJ ユニットデバイス変数 3-8
3-2-4 システム設定用の CJ ユニットデバイス変数 3-15
3-2-5 ソケットサービス関連の CJ ユニットデバイス変数（形 CJ1W-EIP21S のみ）. 3-17
3-3 ステータスエリアのユーザ定義設定 3-25
3-3-1 ユーザ定義設定の概要 3-25
3-3-2 ユーザ定義設定の設定方法 3-25
3-3-3 ユーザ設定のステータスエリアへのアクセス方法 3-26
3-3-4 ユーザ定義設定エリアに割り付けるステータス情報 3-26

3-1　CPU ユニットとのデータ交換

本ユニットと CPU ユニットとのデータ交換は、EtherNet/IP ユニット用に割り付けられる I/O ポート、または必要により CJ ユニット用メモリを使用して行います。
I/O ポートの詳細は、🕮「3-2 CJ ユニットデバイス変数の詳細」を参照してください。

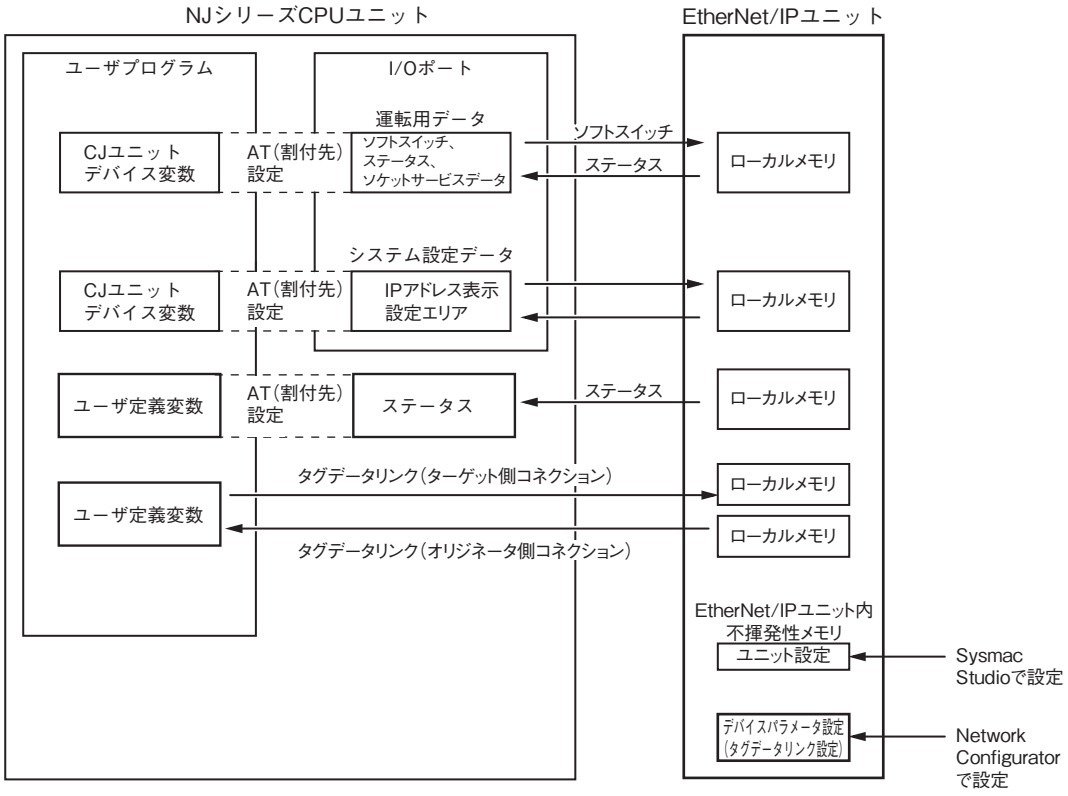
3-1-1　データの流れ

CPU ユニットと EtherNet/IP ユニットは、以下の表および図のとおりデータ交換を行います。

データ交換の種類

ユーザプログラムからのアクセス方法	AT（割付先）指定先	データ交換のタイミング	ユニットのデータ種類
CJ ユニットデバイス変数	I/O ポート（運転用データ）	I/O リフレッシュ時	ソフトスイッチ
			ステータス *1 ソケットサービスデータ *2
	I/O ポート（システム設定データ）	I/O リフレッシュ時	IP アドレス表示・設定エリア
ユーザ定義変数	CJ ユニット用メモリ	I/O リフレッシュ時	ステータス *3

*1 ステータスとして送信されるのは、ターゲットノード PLC 本体運転／異常情報、ユニットステータス、通信ステータス、登録／正常ターゲットノード、FINS/TCP 接続ステータスです。
*2 形 CJ1W-EIP21S、かつ、ステータスエリア設定のステータスレイアウトが［ユーザ設定］時の場合です。このとき、デフォルトで用意されているステータスのうち、送信されるステータスはユニットステータス、通信ステータスのみとなります。
*3 ステータスとして送信されるのは、ターゲットノード PLC 本体運転／異常情報、登録／正常ターゲットノードです。



CJ ユニットデバイス変数

CJ ユニットデバイス変数は、以下に説明する I/O ポートに AT（割付先）指定した変数です。
ユーザプログラムは、CJ ユニットデバイス変数を使って EtherNet/IP ユニットなどの構成ユニットにアクセスできます。
CJ ユニットデバイス変数の I/O ポートへの割付については、📖「CJ ユニットデバイス変数の指定方法」（P.3-5）を参照してください。

● I/O ポート

「I/O ポート」とは、CPU ユニットが EtherNet/IP ユニットなどの構成ユニットと、データを交換するための論理的なインタフェースです。
I/O ポートは、ユニット形式および機能ごとに固有名があらかじめ定義されています。
I/O ポートは Sysmac Studio 上でユニット構成を作成することで、自動的に生成されます。

EtherNet/IP ユニットに定義された I/O ポートの詳細は、📖「3-2 CJ ユニットデバイス変数の詳細」を参照してください。

EtherNet/IP ユニットでは、運転用データとシステム設定用データの 2 種類の I/O ポートがあります。

● 運転用データ

- ソフトスイッチ
  タグデータリンク開始スイッチなどの、出力用（CPU ユニット→本ユニット）のソフトスイッチデータがあります。
  ソフトスイッチデータは、EtherNet/IP ユニット用に割り付けられている運転用データの I/O ポートに格納されています。
  ユーザプログラムからは、CJ ユニットデバイス変数を介して参照やスイッチ操作を行います。
  更新は、I/O リフレッシュのタイミングで行われます。

- ステータス
  通信ステータスなどの、入力用（本ユニット→ CPU ユニット）のステータスデータがあります。
  ステータスデータは、EtherNet/IP ユニット用に割り付けられている運転用データの I/O ポートに格納されています。
  ユーザプログラムからは、CJ ユニットデバイス変数を介して参照します。
  更新は、I/O リフレッシュのタイミングで行われます。

- ソケットサービスデータ（形 CJ1W-EIP21S のみ）
  ステータスエリア設定のステータスレイアウトが［ユーザ設定］時

  各ソケットに対するソケットサービス要求スイッチや UDP/TCP ソケットステータスなどがあります。
  ユーザプログラムからは、CJ ユニットデバイス変数を介して参照やスイッチ操作を行います。

● システム設定用データ

- IP アドレス表示・設定エリア
  システム設定データは、通常 Sysmac Studio の、［高機能ユニット設定の編集］画面を使用して設定します。ユーザプログラムから、該当 I/O ポートに AT（割付先）指定された CJ ユニットデバイス変数（システム設定用）を用いても設定可能です。

ユーザ定義変数

NJ シリーズ CPU ユニットに EtherNet/IP ユニットを装着し、タグデータリンクなど通信を使用するために、ユーザ定義変数を使用します。ユーザ定義変数については、🕮『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）を参照してください。

ユーザプログラムからのアクセス方法

CPU ユニットと EtherNet/IP ユニットは、CPU ユニット内の CJ ユニット用メモリを介して、各種データを交換します。

CJ ユニット用メモリの詳細は、🕮『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)』を参照してください。

ユーザプログラムからは、以下を使用して、各種データを交換します。

データ種類	アクセス方法
ソフトスイッチ	CJ ユニットデバイス変数
ステータスデータ	
システム設定用データ	
ステータスデータ	ユーザ定義変数
タグデータリンクで更新するデータ	

3-1-2　各種変数の指定／作成方法

CJ ユニットデバイス変数の指定方法

Sysmac Studio の［I/O マップ］で、I/O ポートに CJ ユニットデバイス変数を割り付けます。
割り付ける変数名は以下の方法で指定します。

(1) 既存の変数を選択してきて割り付け

(2) 新規に変数名を入力

(3)「デバイス変数生成」で自動生成

以下は、(3) の自動生成で生成した変数名の構成です。

CJユニットデバイス変数名
*_ Unista1

- I/Oポート名です。
  ユニットの機能、ステータス、パラメータを表す名称となっています。
- ユニット構成に、本ユニットを登録するときのデバイス名です。
  （デフォルトではユニット構成への登録順に「J01」～となります。）
  個々のユニットを識別することができます。

CJ ユニットデバイス変数の詳細は、🕮「3-2 CJ ユニットデバイス変数の詳細」を参照してください。

これ以降の説明では、「*_TDLStartCmd」のように、CJ ユニットデバイス変数名として、自動生成した変数名を使用して説明します。

ステータスエリアを割り付けるユーザ定義変数について

ユーザ設定で、ステータスエリアをユーザ定義変数に割り付ける場合、ステータスエリアを割り付けるユーザ定義変数用の構造体のデータ型が、あらかじめ定義してあります。

このデータ型を使用するには、あらかじめ Sysmac Studio の［構成・設定］の［CPU・増設ラック］画面で、EtherNet/IP ユニット用の構造体データ型を登録しておきます。
［構成・設定］の［CPU・増設ラック］でユニットを選択した状態で右クリックして［ユニット用データ型登録］を選択すると、データ型が登録され使用可能となります。

EtherNet/IP ユニット用の構造体データ型の登録後は、ステータスエリアを割り付けるために、上記の構造体型を用いたユーザ定義変数を作成します。
ユーザプログラムは、このユーザ定義変数を使用して、ステータスエリアの情報を読み出すことができます。
ステータスエリアのユーザ設定については、🕮「3-3 ステータスエリアのユーザ定義設定」を参照してください。

- ユーザ定義ステータスエリア用のデータ型

構造体データ型名	メンバ	データ型
_sCJEIP21_User_StaArea	RegTargetSta （登録ターゲットノード）	_uCJEIP21_TargetSta
	EstbTargetSta （正常ターゲットノード）	_uCJEIP21_TargetSta
	TargetPLCMdSta （ターゲットノード PLC 本体運転情報）	_uCJEIP21_TargetSta
	TargetPLCErrSta （ターゲットノード PLC 本体異常情報）	_uCJEIP21_TargetSta

3-2　CJ ユニットデバイス変数の詳細

個々の CJ ユニットデバイス変数の機能を説明します。
以下の説明は、［I/O マップ］で自動生成した変数名で記述します。

3-2-1　CJ ユニットデバイス変数への割付

NJ シリーズでは、EtherNet/IP ユニットの運転用データおよびシステム設定用データの、以下の 3 種類のデータを、I/O ポートとして、あらかじめ定義しています。
I/O ポートに CJ ユニットデバイス変数を割り付けて使用します。

(1) ソフトスイッチ（CPU ユニットから EtherNet/IP ユニットへの各機能の実行指令）

(2) ステータス（EtherNet/IP ユニットの状態やエラー情報、ネットワークやタグデータリンクするターゲットのステータスを示す）

(3) ソケットサービス関連（ソケットサービス要求スイッチや TCP/UDP ソケットステータスなど）

(4) システム設定（EtherNet/IP ユニットに設定されている IP アドレスを格納します）

以降に、EtherNet/IP ユニット用の CJ ユニットデバイス変数を 3 種のデータごとに説明します。

なお、CJ ユニットデバイス変数名と運転用データおよびシステム設定用データのアドレスの対応は、🕮「A-7-2 ユーザプログラムからのアクセス方法に関する相違点」を参照してください。

3-2-2　ソフトスイッチ用の CJ ユニットデバイス変数

（CPU ユニット→ EtherNet/IP ユニット［出力］）

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_TDLStartCmd	BOOL	RW	タグデータリンク開始スイッチ	FALSE → TRUE の変化でタグデータリンクを開始 FALSE: タグデータリンク動作の開始時 初期値： FALSE
*_TDLStopCmd	BOOL	RW	タグデータリンク停止スイッチ	FALSE → TRUE の変化でタグデータリンクを停止 FALSE: タグデータリンク動作の停止時 初期値： FALSE
*_AdjTmCmd	BOOL	RW	時計情報自動調整実行スイッチ	FALSE → TRUE の変化で時計情報自動調整を実行 初期値： FALSE
*_SktForceCloseCmd （形 CJ1W-EIP21S のみ）	BOOL	RW	ソケット強制クローズスイッチ	FALSE → TRUE の変化でソケットの強制クローズ実行 初期値： FALSE

● タグデータリンク開始スイッチ（*_TDLStartCmd）

FALSE から TRUE にすることで、タグデータリンクを開始します。
すでにタグデータリンク動作中の場合は無視されます。
タグデータリンクは、Network Configurator でタグデータリンクパラメータ設定をダウンロード直後、電源再投入を含むコントローラリセット後、またはユニットリスタート後、自動的に起動します。
タグデータリンク停止スイッチ（*_TDLStopCmd）を FALSE から TRUE にすることにより、いったんタグデータリンクを停止した後は、本タグデータリンク開始スイッチ（*_TDLStartCmd）を FALSE から TRUE にすることにより起動します。
タグデータリンク開始後は、EtherNet/IP ユニットが自動的に FALSE に戻します。自動的に FALSE に戻るまでは、このスイッチを強制的に操作しないようにしてください。

● タグデータリンク停止スイッチ（*_TDLStopCmd）

FALSE から TRUE にすることで、タグデータリンクを停止します。
一度タグデータリンク動作を停止させると、ユニットをリスタートするか、タグデータリンク動作を再開させるまで（タグデータリンク開始スイッチを FALSE から TRUE にするまで）、タグデータリンクは停止のままとなります（Network Configurator からの設定ダウンロードでもタグデータリンク動作は再開します）。
すでにタグデータリンク停止中の場合は無視されます。
メッセージ通信はタグデータリンク停止中でも可能です。
タグデータリンク停止後は、EtherNet/IP ユニットが自動的に FALSE に戻します。
自動的に FALSE に戻るまでは、このスイッチを強制的に操作しないようにしてください。

● 時計情報自動調整実行スイッチ（*_AdjTmCmd）

FALSE から TRUE にすることで、時計情報自動調整を実行できます。時計情報自動調整に必要な SNTP サーバは、ユニット設定で設定します。
時計情報自動調整の実行完了後は、Ethernet/IP ユニットが自動的に FALSE に戻します。自動的に FALSE に戻るまでは、このスイッチを強制的に操作しないようにしてください。

● ソケット強制クローズスイッチ（*_SktForceCloseCmd）（形 CJ1W-EIP21S のみ）

FALSE から TRUE にすることで、ソケットサービス機能で使用するすべての UDP ソケット、TCP ソケットを強制的にクローズすることができ、エラー処理などで使用できます。
通信中のソケットを強制クローズするとエラーが発生しますので注意してください。
全ソケットの強制クローズが完了すると、EtherNet/IP ユニット形 CJ1W-EIP21S が自動的に FALSE にします。自動的に FALSE に戻るまでは、このスイッチを強制的に操作しないようにしてください。
EtherNet/IP ユニット形 CJ1W-EIP21S ユニットが固有に使用するポートはクローズされません。

3-2-3　ステータス用の CJ ユニットデバイス変数

EtherNet/IP ユニット→ CPU ユニット［入力］のデータ交換時に使用します。
ソケットサービスデータについても本項で説明します。

ターゲットノード CPU ユニット情報

ステータスエリア設定のステータスエリアレイアウトのレイアウトタイプがデフォルト時のみ使用可能です。
ステータスエリア設定のステータスエリアレイアウトのレイアウトタイプがユーザ設定の場合は、本変数ではなくステータスエリア設定の割付エリアで設定したユーザ定義変数を参照してください。詳細は、「3-3 ステータスエリアのユーザ定義設定」を参照してください。

本ユニットがオリジネータとなっているコネクションにおいて、該当ターゲットノードの CPU ユニットの情報をあらわします。

参考

- このステータスは、オリジネータとターゲットの双方で、通信データにコントローラステータスを含んでいる場合に有効となります。
- この CJ ユニットデバイス変数では、ノード 0 ～ 63 のみの情報を示しています。
- 64 以上のノードを表示させたい場合は、🕮「3-3 ステータスエリアのユーザ定義設定」を参照して下さい。

● ターゲットノード PLC 本体運転情報（ユニット→ CPU ユニット）（*_TargetPLCMdSta［0］～［63］）

本ユニットがオリジネータとなっているコネクションにおいて該当ターゲットノードの CPU ユニットの運転情報をあらわします。
本エリアの情報は、正常ターゲットノード情報が TRUE のときのみ有効となります。
FALSE の場合は直前の値が保持されます。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_TargetPLCMdSta［0］～［63］	ARRAY［0..63］OF BOOL	R	ターゲットノード PLC 本体運転情報	TRUE:　該当ターゲットノードの CPU ユニットが運転状態 FALSE:　上記以外 初期値：0000

● ターゲットノード PLC 本体異常情報（ユニット→ CPU ユニット）（*_TargetPLCErrSta［0］～［63］）

本ユニットがオリジネータとなっているコネクションにおいて該当ターゲットノードの CPU ユニットのコントローラ異常（全停止／部分停止／軽度フォールトの論理和）をあらわします。
本エリアの情報は、正常ターゲットノード情報が TRUE のときのみ有効となります。
FALSE の場合は直前の値が保持されます。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_TargetPLCErrSta［0］～［63］	ARRAY［0..63］OF BOOL	R	ターゲットノード PLC 本体異常情報	TRUE:　該当ターゲットノードの CPU ユニットに全停止／部分停止／軽度フォールトのいずれかが発生中 FALSE:　上記以外

ユニットステータス情報

EthreNet/IP ユニットの状態をあらわします。

● ユニットステータス 1 （ユニット→ CPU ユニット）（*_UnitSta1）

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_UnitSta1	WORD	R	ユニットステータス 1	ビット 00: ユニット異常発生中 ビット 01: ネットワーク異常発生中 ビット 04: ユニットメモリ異常 ビット 05: 通信コントローラ異常 ビット 06: IP アドレス重複異常 ビット 09: リンクオフ異常 ビット 14: ステータスエリア、レイアウト設定異常
*_UnitErr	BOOL	R	ユニット異常発生中	TRUE: EtherNet/IP ユニットとしての動作に関する異常が発生していることを示します。 以下のユニットステータス 1 の CJ ユニットデバイス変数のいずれかが TRUE となっているとき、TRUE になります。（OR 情報） *_NetEr *_UnitMemErr *_LANHwErr *_IPAdrDupErr *_LkOffErr *_UserStaAreaCfgErr FALSE: 上記異常が発生していないことを示します。または異常が解消されたことを示します。 初期値 : FALSE
*_NetErr	BOOL	R	ネットワーク異常発生中	TRUE: ネットワークに関する異常が 1 つ以上発生していることを示します。（通信ステータス 1，3 の異常系ビットの OR にて TRUE になります。） FALSE: 上記異常が発生していないことを示します。または異常が解消されたことを示します。 初期値 : FALSE
*_UnitMemErr	BOOL	R	ユニットメモリ異常	TRUE: ユニット内部の不揮発性メモリへのアクセスで異常が発生したことを示します。（素子の異常） FALSE: 上記異常が発生していないことを示します。この異常は 1 度発生すると正常には復帰しません。（TRUE のままとなります。） 初期値 : FALSE
*_LANHwErr	BOOL	R	通信コントローラ異常	TRUE: 通信コントローラに異常が発生したことを示します。 FALSE: 正常 いったん TRUE になると、電源再投入またはコントローラリセット以外では FALSE に戻りません。 初期値 : FALSE
*_IPAdrDupErr	BOOL	R	IP アドレス重複異常	TRUE: 設定された IP アドレスで ARP を送信し IP アドレス重複を確認します。 ARP 応答があればアドレス重複と判断します。 電源再投入を含むコントローラリセットまたはユニットリスタートで FALSE になります。 （Ethernet インタフェースは停止） FALSE: 上記以外 初期値 : FALSE
*_LkOffErr	BOOL	R	リンクオフ異常	TRUE: スイッチングハブとの間でリンク確立異常が発生していることを示します。 FALSE: スイッチングハブとの間でリンクが正常に確立していることを示します。 初期値 : FALSE

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_UserStaAreaCfgErr	BOOL	R	ステータスエリア レイアウト設定異常	TRUE: 割付けリレーエリアのレイアウト設定に異常が発生していることを示します。 この場合、割付けリレーエリアのレイアウトは「デフォルト」で動作します。 ただし、存在しないエリアを指定したとき、割付けリレーエリアのレイアウトは「ユーザ設定」で動作し、ユーザ設定エリアはリフレッシュされません。 FALSE: 割付けリレーエリアのレイアウト設定が問題なくされていることを示します。 初期値 : FALSE

● ユニットステータス 2（ユニット→ CPU ユニット）（*_UnitSta2）

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_UnitSta2	WORD	R	ユニットステータス 2	ビット 00: オンライン ビット 01: タグデータリンク動作中 ビット 02: 運転中の IP アドレス変更 ビット 11: ユーザ設定エリア有効 ビット 14: 複数スイッチ同時 ON 異常 ビット 15: 異常履歴あり
*_ETNOnlineSta	BOOL	R	オンライン	TRUE: オンライン状態（EtherNet/IP ユニット内部の通信処理が動作可能）であることを示します。 FALSE: オフライン状態であることを示します。 次の場合にオフライン状態となります。 •IP アドレス重複異常 •Ethernet 通信コントローラ異常（ハード異常） •BOOTP サーバ異常
*_TDLOprSta	BOOL	R	タグデータリンク動作中	TRUE: タグデータリンク動作中であることを示します。 次の場合に TRUE になります。 •ユニットにオリジネータとしての設定がされていて、電源再投入を含むコントローラリセットまたはユニットリスタートしたとき。 •ユニットにオリジネータとしての設定がされていて、タグデータリンク開始スイッチを TRUE にしたとき。 FALSE: タグデータリンク動作停止中であることを示します。 次の場合は通信を停止するため FALSE になります。 •タグデータリンク停止スイッチを TRUE にしたとき。 •ハード異常 •IP アドレス重複異常 •BOOTP サーバ異常 •Ethernet 基本設定異常 •メモリ異常（MAC アドレス異常）
*_IPAdrChgErr	BOOL	R	運転中の IP アドレス変更	TRUE: ノードアドレス設定スイッチを CPU ユニット運転中に変更した場合に TRUE になります。 FALSE: ノードアドレス設定スイッチが CPU ユニットの運転開始時と同じ状態を保持しています。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_UserStaAreaEnblSta	BOOL	R	ユーザ設定エリア有効	TRUE: 該当ユーザ設定エリアのデータが有効なのを示します。 割付けリレーエリアのレイアウトが「ユーザ設定」に指定された場合で、ユーザ設定エリアのリフレッシュを開始したとき TRUE になります。 FALSE: 該当ユーザ設定エリアのデータが無効なのを示します。 次の場合は通信を停止するため FALSE になります。 •割付けリレーエリアのレイアウトが「デフォルト」に指定された場合 •割付けリレーエリアのレイアウトが「ユーザ設定」に指定された場合で、レイアウト設定異常発生中のとき
*_MultiSwOnErr	BOOL	R	複数スイッチ同時 ON 異常	TRUE: 複数のソフトスイッチを同時に ON したときに TRUE になります。（未使用エリアは無視します。） FALSE: 次のスイッチ操作開始時に FALSE になります。
*_ErrLogStoreSta	BOOL	R	異常履歴あり	TRUE: 異常履歴が登録されていることを示します。 FALSE: 異常履歴に 1 つも異常が登録されていないことを示します。異常履歴クリアの要求を受けたときも FALSE となります。

通信ステータス情報

タグデータリンクや EtherNet/IP に関するステータスが、次のように反映されます。

● 通信ステータス 1（ユニット→ CPU ユニット）（*_CommSta1）

タグデータリンクに関するステータスが、次のように反映されます。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_CommSta1	WORD	R	通信ステータス 1	ビット 00: 照合異常 ビット 02: タグデータリンク異常 ビット 04: 通信パラメータ設定不正 ビット 05: タグリフレッシュ異常 ビット 06: タグデータベース異常 ビット 14: 全タグデータリンク通信中 ビット 15: タグデータリンク通信中
*_TDLOpnErr	BOOL	R	照合異常	TRUE: タグデータリンクパラメータに登録されている相手ノード情報と実際のノード情報が異なることを示します。 主な要因 •指定されたターゲットが存在しない場合 •変数名が不一致の場合 •コネクションサイズが異なる場合 •コネクションリソース不足 FALSE: 照合異常が発生していないことを示します。照合異常が解消されたとき FALSE になります。
*_TDLErr	BOOL	R	タグデータリンク異常	TRUE: オリジネータとしてのコネクションに 1 つ以上の異常があることを示します。 ただし、以下のケースについては、本ステータスの異常の対象外となります。 •ターゲットとしてのコネクションの異常 •スイッチングハブとのリンクオフ異常発生によるコネクションタイムアウト FALSE: 上記異常が発生していないことを示します。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_CommParamErr	BOOL	R	通信パラメータ設定不正	TRUE: 不揮発性メモリに保存されるタグデータリンクに関連するパラメータの妥当性確認で、異常と判断された場合、SUM 値異常が発生した場合 TRUE になります。（Ethernet の基本設定に関連するパラメータも含みます。）タグデータリンクは停止します。 FALSE: 不揮発性メモリに保存されるパラメータの妥当性が正しいことを示します。
*_TagRefreshErr	BOOL	R	タグリフレッシュ異常	TRUE: タグデータリンクでサポート外のエリア種別、アドレス範囲を指定したとき TRUE になります。 TALSE: タグデータリンクでサポート内のエリア種別、アドレス範囲を指定しているとき FALSE になります。
*_TagDbErr	BOOL	R	タグデータベース異常	TRUE: EtherNet/IP ユニットの設定で変数名（タグデータリンク、ステータスレイアウト割付けエリアなど）を使用した場合に、CPU ユニットでタグデータベース異常が発生したとき TRUE になります。（形 CJ2H-CPU6 □ -EIP、形 CJ2M-CPU3 □ のみ） FALSE: EtherNet/IP ユニットの設定で変数名を使用していない、タデータベース異常が発生していない、異常が解除されたときなどに FALSE になります。
*_TDLAllRunSta	BOOL	R	全タグデータリンク通信中	TRUE: オリジネータとして全てのコネクションでタグデータリンクが通信中であることを示します。 FALSE: オリジネータとして 1 つ以上のコネクションでタグデータリンクできていないことを示します。（一部で通信していても FALSE になります。）
*_TDLRunSta	BOOL	R	タグデータリンク通信中	TRUE: オリジネータとして 1 つ以上のコネクションでタグデータリンクが通信中であることを示します。 FALSE: オリジネータとして 1 つのコネクションもタグデータリンクできていないことを示します。（ターゲットとして通信していても FALSE になります。）

● 通信ステータス 2（ユニット→ CPU ユニット）（*_CommSta2）

Ethernet に関するステータスが、次のように反映されます。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_CommSta2	WORD	R	通信ステータス 2	ビット 00: FTP ステータス ビット 14: リンクステータス
*_FTPSta	BOOL	R	FTP ステータス	TRUE: FTP サーバ動作中です。（FTP クライアント接続中） FALSE: FTP サーバ待機中です。（クライアント接続待ち）
*_LkSta	BOOL	R	リンクステータス	TRUE: スイッチングハブとの間でリンクが確立しています。 FALSE: スイッチングハブとの間でリンクが停止しています。

● 通信ステータス 3（ユニット→ CPU ユニット）（*_CommSta3）

Ethernet の異常に関するステータスが、次のように反映されます。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_CommSta3	WORD	R	通信ステータス 3	ビット 02: Ethernet 基本設定異常 ビット 03: IP アドレステーブル異常 ビット 04: IP ルータテーブル異常 ビット 05: DNS サーバ異常 ビット 06: ルーチングテーブル異常 ビット 09: Ethernet 拡張設定異常 ビット 10: BOOTP サーバ異常 ビット 11: SNTP サーバ異常 ビット 14: アドレス不一致 ビット 15: 不揮発性メモリ異常
*_ETNBaseSetErr	BOOL	R	Ethernet 基本設定異常	TRUE: Ethernet に関する以下の設定が不正です。 •TCP/IP 設定 （IP アドレス、サブネットマスク、LINK 設定） FALSE: 正常です。
*_IPAdrTblErr	BOOL	R	IP アドレステーブル異常	TRUE: IP アドレステーブルの情報が正しくありません。 FALSE: IP アドレステーブルの情報が正常です。
*_IPRouterTblErr	BOOL	R	IP ルータテーブル異常	TRUE: IP ルータテーブルの情報が正しくありません。 FALSE: IP ルータテーブルの情報が正常です。
*_DNSSvrErr	BOOL	R	DNS サーバ異常	TRUE: DNS クライアントが DNS サーバの接続に失敗（タイムアウト）しました。 FALSE: DNS の設定がされていない。または、DNS サーバの情報がされていて接続に成功しました。
*_RTblErr	BOOL	R	ルーチングテーブル異常	TRUE: ルーチングテーブルの情報が正しくありません。 FALSE: ルーチングテーブルが正常です。
*_ETNAdvSetErr	BOOL	R	Ethernet 拡張設定異常	TRUE: Ethernet に関する以下の設定が不正です。 •FINS 設定 FALSE: 正常です。
*_BootpSvrErr	BOOL	R	BOOTP サーバ異常	TRUE: BOOTP サーバの接続に失敗（タイムアウト）しました FALSE: BOOTP 設定がされていないか、または、BOOTP サーバ設定がされていて正常に BOOTP サーバから IP アドレスを取得しました。
*_SNTPSvrErr	BOOL	R	SNTP サーバ異常	TRUE: SNTP クライアントがサーバの接続に失敗（タイムアウト）しました。 FALSE: SNTP 設定がされていないか、または、SNTP の設定がされていて接続に成功しました。
*_AdrMismatchErr	BOOL	R	アドレス不一致	TRUE: モード設定で相手 IP アドレス変換方式が「自動生成方式」に指定されているが、自 IP アドレスのホスト ID と FINS ノードアドレスが一致していない。 FALSE: 上記の設定状態以外
*_MemErr	BOOL	R	不揮発性メモリ異常	TRUE: ユニット内部の不揮発性メモリが異常です。 FALSE: ユニット内部の不揮発性メモリが正常です。

ターゲットノードテーブル

ステータスエリア設定のステータスエリアレイアウトのレイアウトタイプがデフォルト時のみ使用可能です。
ステータスエリア設定のステータスエリアレイアウトのレイアウトタイプがユーザ設定の場合は、本変数ではなくステータスエリア設定の割付エリアで設定したユーザ定義変数を参照してください。詳細は、「3-3 ステータスエリアのユーザ定義設定」を参照してください。

本ユニットがオリジネータとなっているコネクションにおいて該当ターゲットノードの情報をあらわします。



参考

本エリアではノード 0 ～ 63 のみの情報を示しています。64 以上のノードを表示させたい場合は、🕮「3-3 ステータスエリアのユーザ定義設定」を参照して下さい。

● 登録ターゲットノード（ユニット→ CPU ユニット）（*_RegTargetSta［0］～［63］）

本ユニットがオリジネータとなっているコネクションにおいて該当ターゲットノードのタグデータリンクの登録情報をあらわします。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_RegTargetSta［0］～［63］	ARRAY［0..63］OF BOOL	R	登録ターゲットノード	TRUE: タグデータリンクが登録されていることを示します。 FALSE: タグデータリンクが登録されていないことを示します。

● 正常ターゲットノード（ユニット→ CPU ユニット）（*_EstbTargetSta［0］～［63］）

本ユニットがオリジネータとなっているコネクションにおいて該当ターゲットノードとのコネクション確立情報をあらわします。
ターゲットデバイス単位で複数あるコネクションのすべてのデータを、CPU ユニットへリフレッシュしてから TRUE になります。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_EstbRegTargetSta［0］～［63］	ARRAY［0..63］OF BOOL	R	正常ターゲットノード	TRUE: コネクションを正常に確立しているノード一覧を表します。 FALSE: コネクションが正常に確立されていないノードを表します。

3-2-4　システム設定用の CJ ユニットデバイス変数

IP アドレス表示・設定エリア（*_IPAdrCfg）

自 IP アドレスの確認や設定を行う場合、次の手順で変数を定義してください。

1　I/O ポートの IP アドレス表示・設定エリア（*_IPAdrCfg）に、デバイス変数が割り付けられていないことを確認します。

割り付けられている場合は、割り付けを解除します。（該当するデバイス変数を削除する）

割り付けられていないことを確認

ポート	説明	R/W	データ型	変数
▼IPAdrCfg	IPアドレス表示・設定エリア	RW	DWORD	
IPAdr1Cfg	IPアドレス 1	RW	USINT	
IPAdr2Cfg	IPアドレス 2	RW	USINT	
IPAdr3Cfg	IPアドレス 3	RW	USINT	
IPAdr4Cfg	IPアドレス 4	RW	USINT	

2　以下の設定を行います。

- IP アドレス表示・設定エリアをグローバル変数テーブルに登録します。
- 登録した変数の「割付先」属性に、AT（割付先）指定するアドレスを設定します。
  アドレスは下記にて計算できます。
  IP アドレス表示・設定エリア = D30000 +ユニット番号 ×100 + 98,99 （ユニット番号：0 ～ 15）
- 登録した変数の「保持」属性にチェックを入れます。
- 外部機器からモニタする場合は「ネットワーク公開」を「公開のみ」とします。

例えば、変数名が「J01_IPAdr*Cfg」、EtherNet/IP ユニットのユニット番号が 0 の場合、IP アドレス表示・設定エリアのアドレスは D30098,D30099 となり、下記のようにグローバル変数に定義します。

グローバル変数

名称	データ型	初期値	割付先	保持	コンスタント	ネットワーク公開	
J01_IPAdrCfg	DWORD		%D30098	☑	☐	公開のみ	
J01_IPAdr1Cfg	USINT		%D30098.08	☑	☐	公開のみ	← EtherNet/IPユニットのIPアドレスの①
J01_IPAdr2Cfg	USINT		%D30098	☑	☐	公開のみ	← EtherNet/IPユニットのIPアドレスの②
J01_IPAdr3Cfg	USINT		%D30099.08	☑	☐	公開のみ	← EtherNet/IPユニットのIPアドレスの③
J01_IPAdr4Cfg	USINT		%D30099	☑	☐	公開のみ	← EtherNet/IPユニットのIPアドレスの④

自 IP アドレスが以下のように格納されます。

バイト	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
先頭	①	②
先頭+1	③	④

IPアドレス:①.②.③.④（16進数）

TCP/IP 設定の自 IP アドレスに 0.0.0.0 以外が設定されている場合、手順 2 で設定した変数は、自 IP アドレスの表示エリアとなり、TCP/IP 設定で設定されている自 IP アドレスが、電源 ON 時またはユニットリスタート時に読み出されて格納されます。

TCP/IP 設定の自 IP アドレスに 0.0.0.0 が設定されている場合、電源 ON 時またはユニットリスタート時に EtherNet/IP ユニットが手順 2 で設定した変数を読み出し、自 IP アドレスとして設定します。
手順 2 で設定した変数と、TCP/IP 設定の自 IP アドレスの両方が 0.0.0.0 の場合は、デフォルトの IP アドレス（デフォルト IP アドレス = 192.168.250. ノードアドレス）となります。
IP アドレスについての詳細は 🕮「第 4 章 IP アドレスを決定する方法」を参照ください。

使用上の注意

- TCP/IP 設定の自 IP アドレスに 0.0.0.0 以外が設定されている場合は、手順 2 で設定した変数に IP アドレスを設定しても、TCP/IP 設定内の自 IP アドレスの値に書き替えられます。
- 次のような IP アドレスは設定できません。
  - ネット ID 部の全ビットが 0 または 1
  - ホスト ID 部の全ビットが 0 または 1
  - サブネット ID 部の全ビットが 1
  - 127（7F Hex）で始まる IP アドレス（例 : 127.35.21.16）

3-2-5　ソケットサービス関連の CJ ユニットデバイス変数（形 CJ1W-EIP21S のみ）

ステータスエリア設定のステータスエリアレイアウトのレイアウトタイプが［ユーザ設定］時のみ使用可能です。
ステータスエリア設定のステータスエリアレイアウトのレイアウトタイプのユーザ設定については、「3-3 ステータスエリアのユーザ定義設定」を参照してください。

ソケットサービス要求スイッチ 1 ～ 8（CPU ユニット→ユニット）（*_Skt1Cmd ～ *_Skt8Cmd）

変数操作でソケットサービス要求を実行する場合に、以下の変数を操作します。
詳細は「第 13 章 ソケットサービス機能」を参照してください。

ソケットサービス要求スイッチ 1

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt1Cmd	BYTE	R/W	ソケットサービス要求スイッチ 1	ソケットサービス要求を実行します。
*_Skt1UdpOpenCmd	BOOL	R/W	UDP オープン要求スイッチ	FALSE → TRUE の変化で UDP オープン実行 FALSE: オープン完了時 初期値： FALSE
*_Skt1TcpPassiveCmd	BOOL	R/W	TCP オープン (Passive) 要求スイッチ	FALSE → TRUE の変化で TCP オープン実行 FALSE: オープン完了時（コネクション時） 初期値： FALSE
*_Skt1TcpActiveCmd	BOOL	R/W	TCP オープン (Active) 要求スイッチ	FALSE → TRUE の変化で TCP オープン実行 FALSE: オープン完了時（コネクション時） 初期値： FALSE
*_Skt1SendCmd	BOOL	R/W	送信要求スイッチ	FALSE → TRUE の変化で送信実行（UDP か TCP かはオープン時に決定） FALSE: 送信完了時 初期値： FALSE
*_Skt1RcvCmd	BOOL	R/W	受信要求スイッチ	FALSE → TRUE の変化で受信実行（UDP か TCP かはオープン時に決定） FALSE: 受信完了時 初期値： FALSE
*_Skt1CloseCmd	BOOL	R/W	クローズ要求スイッチ	FALSE → TRUE の変化でクローズ実行 FALSE: クローズ完了時 初期値： FALSE

以降、ソケットサービス要求スイッチ 2 ～ 8 も同様の項目となります。
（ソケットサービス要求スイッチ 2 ～ 7 については表の記載を省略します。）

ソケットサービス要求スイッチ 8

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt8Cmd	BYTE	R/W	ソケットサービス要求スイッチ 8	ソケットサービス要求を実行します。
*_Skt8UdpOpenCmd	BOOL	R/W	UDP オープン要求スイッチ	FALSE → TRUE の変化で UDP オープン実行 FALSE: オープン完了時 初期値： FALSE
*_Skt8TcpPassiveCmd	BOOL	R/W	TCP オープン (Passive) 要求スイッチ	FALSE → TRUE の変化で TCP オープン実行 FALSE: オープン完了時（コネクション時） 初期値： FALSE
*_Skt8TcpActiveCmd	BOOL	R/W	TCP オープン (Active) 要求スイッチ	FALSE → TRUE の変化で TCP オープン実行 FALSE: オープン完了時（コネクション時） 初期値： FALSE
*_Skt8SendCmd	BOOL	R/W	送信要求スイッチ	FALSE → TRUE の変化で送信実行（UDP か TCP かはオープン時に決定） FALSE: 送信完了時 初期値： FALSE

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt8RcvCmd	BOOL	R/W	受信要求スイッチ	FALSE → TRUE の変化で受信実行（UDP か TCP かはオープン時に決定） FALSE: 受信完了時 初期値： FALSE
*_Skt8CloseCmd	BOOL	R/W	クローズ要求スイッチ	FALSE → TRUE の変化でクローズ実行 FALSE: クローズ完了時 初期値： FALSE

UDP ソケット No.1 ～ No.8 ステータス（ユニット→ CPU ユニット）（*_Skt1UdpSta ～ *_Skt8UdpSta）

詳細は「第 13 章 ソケットサービス機能」を参照してください。

ソケット No.1

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt1UdpSta	BYTE	R	UDP ソケット No.1 ステータス	ソケットサービスのステータス情報を表します。
*_Skt1UdpOpening	BOOL	R	オープン処理中フラグ	TRUE: オープン処理中（オープン要求受付時必ず TRUE になる） FALSE: オープン処理完了時 初期値： FALSE
*_Skt1UdpRcving	BOOL	R	受信処理中フラグ	TRUE: 受信処理中（高速化オプション無効時に受信要求受付時必ず TRUE、有効時には FALSE のまま） FALSE: 受信処理完了時 初期値： FALSE
*_Skt1UdpSending	BOOL	R	送信処理中フラグ	TRUE: 送信処理中（高速化オプション無効時に送信要求受付時必ず TRUE、有効時には FALSE のまま） FALSE: 送信処理完了時 初期値： FALSE
*_Skt1UdpClosing	BOOL	R	クローズ処理中フラグ	TRUE: クローズ処理中（クローズ要求受付時必ず TRUE になる） FALSE: クローズ処理完了時 初期値： FALSE
*_Skt1UdpAreaErr	BOOL	R	結果格納異常フラグ	TRUE: SendCmd 命令を使用してソケットサービス機能を使用する際に、結果格納エリアの指定に誤りがあるとき （TRUE のタイミングは、オープン処理中フラグ、受信処理中フラグ、送信処理中フラグ、クローズ処理中フラグのいずれかが TRUE → FALSE（処理完了時）の時点） FALSE: 次の要求を受け付けたとき 初期値： FALSE
*_Skt1UdpOpened	BOOL	R	UDP オープン済みフラグ	TRUE: オープン処理完了後 FALSE: クローズ処理完了後 （オープン処理の異常完了は FALSE のまま） 初期値： FALSE

以降、ソケット No.2 ～ No.8 も同様の項目となります。
（ソケット No.2 ～ No.7 については表の記載を省略します。）

ソケット No.8

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt8UdpSta	BYTE	R	UDP ソケット No.8 ステータス	ソケットサービスのステータス情報を表します。
*_Skt8UdpOpening	BOOL	R	オープン処理中フラグ	TRUE: オープン処理中（オープン要求受付時必ず TRUE になる） FALSE: オープン処理完了時 初期値： FALSE

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt8UdpRcving	BOOL	R	受信処理中フラグ	TRUE: 受信処理中（高速化オプション無効時に受信要求受付時必ず TRUE、有効時には FALSE のまま） FALSE: 受信処理完了時 初期値： FALSE
*_Skt8UdpSending	BOOL	R	送信処理中フラグ	TRUE: 送信処理中（高速化オプション無効時に送信要求受付時必ず TRUE、有効時には FALSE のまま） FALSE: 送信処理完了時 初期値： FALSE
*_Skt8UdpClosing	BOOL	R	クローズ処理中フラグ	TRUE: クローズ処理中（クローズ要求受付時必ず TRUE になる） FALSE: クローズ処理完了時 初期値： FALSE
*_Skt8UdpAreaErr	BOOL	R	結果格納異常フラグ	TRUE: SendCmd 命令を使用してソケットサービス機能を使用する際に、結果格納エリアの指定に誤りがあるとき（TRUE のタイミングは、オープン処理中フラグ、受信処理中フラグ、送信処理中フラグ、クローズ処理中フラグのいずれかが TRUE → FALSE（処理完了時）の時点） FALSE: 次の要求を受け付けたとき 初期値： FALSE
*_Skt8UdpOpened	BOOL	R	UDP オープン済みフラグ	TRUE: オープン処理完了後 FALSE: クローズ処理完了後（オープン処理の異常完了は FALSE のまま） 初期値： FALSE

TCP ソケット No.1 ～ No.8 ステータス（ユニット→ CPU ユニット）（*_Skt1TcpSta ～ *_Skt8TcpSta）

詳細は「第 13 章 ソケットサービス機能」を参照してください。

ソケット No.1

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt1TcpSta	BYTE	R	TCP ソケット No.1 ステータス	ソケットサービスのステータス情報を表します。
*_Skt1TcpOpening	BOOL	R	オープン処理中フラグ	TRUE: オープン処理中（オープン要求受付時必ず TRUE になる） FALSE: オープン処理完了時 初期値： FALSE
*_Skt1TcpRcving	BOOL	R	受信処理中フラグ	TRUE: 受信処理中（高速化オプション無効時に受信要求受付時必ず TRUE、有効時には FALSE のまま） FALSE: 受信処理完了時 初期値： FALSE
*_Skt1TcpSending	BOOL	R	送信処理中フラグ	TRUE: 送信処理中（高速化オプション無効時に送信要求受付時必ず TRUE、有効時には FALSE のまま） FALSE: 送信処理完了時 初期値： FALSE
*_Skt1TcpClosing	BOOL	R	クローズ処理中フラグ	TRUE: クローズ処理中（クローズ要求受付時必ず TRUE になる） FALSE: クローズ処理完了時 初期値： FALSE
*_Skt1TcpDatRcved	BOOL	R	受信データあり／なしフラグ	TRUE: オープン済み TCP ソケットで、相手ノードからのデータを受信したとき FALSE: オープン済み TCP ソケットで、受信処理要求を実行したとき 初期値： FALSE

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt1TcpAreaErr	BOOL	R	結果格納異常フラグ	TRUE:　SendCmd 命令を使用してソケットサービス機能を使用する際に、結果格納エリアの指定に誤りがあるとき （TRUE のタイミングは、オープン処理中フラグ、受信処理中フラグ、送信処理中フラグ、クローズ処理中フラグのいずれかが TRUE → FALSE（処理完了時）の時点） FALSE:　次の要求を受け付けたとき 初期値：FALSE
*_Skt1TcpOpened	BOOL	R	TCP オープン済みフラグ	TRUE:　オープン処理完了後（コネクト中） FALSE:　クローズ処理完了後 （オープン処理の異常完了は FALSE のまま） 初期値：FALSE

以降、ソケット No.2 ～ No.8 も同様の項目となります。
（ソケット No.2 ～ No.7 については表の記載を省略します。）

ソケット No.8

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt8TcpSta	BYTE	R	TCP ソケット No.8 ステータス	ソケットサービスのステータス情報を表します。
*_Skt8TcpOpening	BOOL	R	オープン処理中フラグ	TRUE:　オープン処理中（オープン要求受付時必ず TRUE になる） FALSE:　オープン処理完了時 初期値：FALSE
*_Skt8TcpRcving	BOOL	R	受信処理中フラグ	TRUE:　受信処理中（高速化オプション無効時に受信要求受付時必ず TRUE、有効時には FALSE のまま） FALSE:　受信処理完了時 初期値：FALSE
*_Skt8TcpSending	BOOL	R	送信処理中フラグ	TRUE:　送信処理中（高速化オプション無効時に送信要求受付時必ず TRUE、有効時には FALSE のまま） FALSE:　送信処理完了時 初期値：FALSE
*_Skt8TcpClosing	BOOL	R	クローズ処理中フラグ	TRUE:　クローズ処理中（クローズ要求受付時必ず TRUE になる） FALSE:　クローズ処理完了時 初期値：FALSE
*_Skt8TcpDatRcved	BOOL	R	受信データあり／なしフラグ	TRUE:　オープン済み TCP ソケットで、相手ノードからのデータを受信したとき FALSE:　オープン済み TCP ソケットで、受信処理要求を実行したとき 初期値：FALSE
*_Skt8TcpAreaErr	BOOL	R	結果格納異常フラグ	TRUE:　SendCmd 命令を使用してソケットサービス機能を使用する際に、結果格納エリアの指定に誤りがあるとき （TRUE のタイミングは、オープン処理中フラグ、受信処理中フラグ、送信処理中フラグ、クローズ処理中フラグのいずれかが TRUE → FALSE（処理完了時）の時点） FALSE:　次の要求を受け付けたとき 初期値：FALSE
*_Skt8TcpOpened	BOOL	R	TCP オープン済みフラグ	TRUE:　オープン処理完了後（コネクト中） FALSE:　クローズ処理完了後 （オープン処理の異常完了は FALSE のまま） 初期値：FALSE

TCP ソケット No.1 ～ 8 受信データバイト数（ユニット→ CPU ユニット）（*_Skt1TcpRcvDatByte ～ *_Skt8TcpRcvDatByte）

この変数には、TCP ソケットごとに、受信バッファにある受信データのバイト数が格納されます。受信バッファには、最大 4,096 バイトまでデータが保持されますが、変数操作または SendCmd 命令の受信要求で設定できる最大値（1,984 バイト）までの範囲で値を格納します。

0000Hex: 0 バイト
07C0Hex: 1,984 バイト

この変数と連動して、UDP/TCP ソケット No.1 ～ 8 ステータスの「受信データあり / なしフラグ」を TRUE/FALSE にします。変数操作または SendCmd 命令により受信要求を実行した時点で、本変数はいったん 0000Hex になります。
受信要求処理を完了しても受信バッファにデータが残っている場合は、再び残存バイト数を格納し、「受信データあり / なしフラグ」を TRUE にします。
受信要求は、受信データ数に必要なデータが揃ったのを確認してから実行すると便利です。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt1TcpRcvDatByte	UINT	R	TCP ソケット No.1 受信データバイト数	TCP ソケット No.1 の受信データバイト数を表します。
*_Skt2TcpRcvDatByte	UINT	R	TCP ソケット No.2 受信データバイト数	TCP ソケット No.2 の受信データバイト数を表します。
*_Skt3TcpRcvDatByte	UINT	R	TCP ソケット No.3 受信データバイト数	TCP ソケット No.3 の受信データバイト数を表します。
*_Skt4TcpRcvDatByte	UINT	R	TCP ソケット No.4 受信データバイト数	TCP ソケット No.4 の受信データバイト数を表します。
*_Skt5TcpRcvDatByte	UINT	R	TCP ソケット No.5 受信データバイト数	TCP ソケット No.5 の受信データバイト数を表します。
*_Skt6TcpRcvDatByte	UINT	R	TCP ソケット No.6 受信データバイト数	TCP ソケット No.6 の受信データバイト数を表します。
*_Skt7TcpRcvDatByte	UINT	R	TCP ソケット No.7 受信データバイト数	TCP ソケット No.7 の受信データバイト数を表します。
*_Skt8TcpRcvDatByte	UINT	R	TCP ソケット No.8 受信データバイト数	TCP ソケット No.8 の受信データバイト数を表します。

TCP ソケット No.1 ～ 8 コネクション状態（ユニット→ CPU ユニット）（*_Skt1TcpConSta ～ *_Skt8TcpConSta）

この変数には、TCP ソケットごとのコネクション状態が、コードで格納されます。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt1TcpConSta	WORD	R	TCP ソケット No.1 コネクション状態	TCP ソケット No.1 のコネクション状態を表します。
*_Skt2TcpConSta	WORD	R	TCP ソケット No.2 コネクション状態	TCP ソケット No.2 のコネクション状態を表します。
*_Skt3TcpConSta	WORD	R	TCP ソケット No.3 コネクション状態	TCP ソケット No.3 のコネクション状態を表します。
*_Skt4TcpConSta	WORD	R	TCP ソケット No.4 コネクション状態	TCP ソケット No.4 のコネクション状態を表します。
*_Skt5TcpConSta	WORD	R	TCP ソケット No.5 コネクション状態	TCP ソケット No.5 のコネクション状態を表します。
*_Skt6TcpConSta	WORD	R	TCP ソケット No.6 コネクション状態	TCP ソケット No.6 のコネクション状態を表します。
*_Skt7TcpConSta	WORD	R	TCP ソケット No.7 コネクション状態	TCP ソケット No.7 のコネクション状態を表します。
*_Skt8TcpConSta	WORD	R	TCP ソケット No.8 コネクション状態	TCP ソケット No.8 のコネクション状態を表します。

コネクション状態のコードは以下の次のとおりです。

コード	状態	内容
0	CLOSED	クローズされている。
1	LISTEN	コネクションを待っている。
2	SYN SENT	活性状態でSYNを送信した。
3	SYN RECEIVED	SYNを送信し、受信した。
4	ESTABLISHED	開設されている。
5	CLOSE WAIT	FINを受信した後、終了待ちの状態である。
6	FIN WAIT1	終了して、FINを送信した。
7	CLOSING	終了して、FINを交換した。 FINの送達確認(ACK)を待っている。
8	LAST ACK	FINを送信して終了した。 FINの送達確認(ACK)を待っている
9	FIN WAIT2	FINの送達確認(ACK)を受信した。 FINを待っている。
A	TIME WAIT	終了した後、最大セグメント生存時間の2倍(2MSL)の沈黙待ちの状態である。

ソケットサービスパラメータエリア 1 ～ 8（ユニット→ CPU ユニット）(*_Skt1ParamArea ～ *_Skt8ParamArea)

変数操作によってソケットサービス要求を実行する場合は、あらかじめこのパラメータエリアに必要な設定を行う必要があります。
ただし、使用されるパラメータは、サービス要求によって異なります。
サービス要求ごとの使用パラメータエリアや、設定内容、使用方法については「第 13 章 ソケットサービス機能」を参照してください。

ソケットサービスパラメータエリア 1

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt1ParamArea	ARRAY [0..9] OF WORD	R/W	ソケットサービスパラメータエリア 1	ソケットのサービスのパラメータを設定します。
*_Skt1No	USINT	R/W	UDP/TCP ソケット No.	オープンする UDP または TCP のソケット No. を指定します。
*_Skt1Option	BYTE	R/W	ソケットオプション	TCP ソケットのオープン要求コマンドで、keep-alive 機能の仕様を設定します。 オープンする UDP または TCP のソケット No. を指定します。
*_Skt1KeepAlive	BOOL	R/W	Keep-alive 機能使用	TCP ソケットのオープン要求（PASSIVE/ACTIVE）コマンドで、keep-alive 機能の使用を指定します。 TRUE: 使用する FALSE: 使用しない
*_Skt1Linger	BOOL	R/W	Linger 機能使用	linger 機能の使用を指定します。 TRUE: 使用する FALSE: 使用しない
*_Skt1LocalPortNo	UINT	R/W	自 UDP/TCP ポート No.（0000 ～ FFFF Hex）	ソケットが送受信を行う UDP または TCP ポート No. を指定します。
*_Skt1RemoteIPAdr	DWORD	R/W	相手 IP アドレス（00000000 ～ FFFFFFFF Hex）	相手先の IP アドレスを指定します。
*_Skt1RemoteIPAdr1	USINT	R/W	相手 IP アドレスの 1 バイトめ（16#00 ～ 16#FF）	相手 IP アドレスの 1 バイトめを指定します。
*_Skt1RemoteIPAdr2	USINT	R/W	相手 IP アドレスの 2 バイトめ（16#00 ～ 16#FF）	相手 IP アドレスの 2 バイトめを指定します。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt1RemoteIPAdr3	USINT	R/W	相手 IP アドレスの 3 バイトめ (16#00 ～ 16#FF)	相手 IP アドレスの 3 バイトめを指定します。
*_Skt1RemoteIPAdr4	USINT	R/W	相手 IP アドレスの 4 バイトめ (16#00 ～ 16#FF)	相手 IP アドレスの 4 バイトめを指定します。
*_Skt1RemotePortNo	UINT	R/W	相手 UDP/TCP ポート No.（0000 ～ FFFF Hex）	相手先の UDP または TCP のポート No. を指定します。
*_Skt1SendRcvByte	UINT	R/W	送信／受信バイト数 (0000 ～ 07C0 Hex(1984))	送信要求または受信要求時の、送信または受信データのバイト数を指定します。
*_Skt1SendRcvAdr	DWORD	R/W	送信／受信データアドレス	送信要求時の送信データの格納先先頭アドレス、または受信要求時の受信データの格納先先頭アドレスを指定します。
*_Skt1SendRcvAdrType	BYTE	R/W	エリア種別 (#16#00 ～ #16#FF)	エリア種別を指定します。
*_Skt1SendRcvAdrCh1	BYTE	R/W	チャネルアドレス（上位 2 けた）	チャネルアドレスの上位 2 けたを指定します。
*_Skt1SendRcvAdrCh2	BYTE	R/W	チャネルアドレス（下位 2 けた）	チャネルアドレスの下位 2 けたを指定します。
*_Skt1Timeout	UINT	R/W	タイムアウト値 (0000 ～ FFFF Hex)	受信（TCP、UDT）、オープン（PASSIVE）時ソケットサービス要求スイッチを TRUE にしてから、完了するまでの制限時間を指定します。
*_Skt1RespCode	WORD	R	終了コード	ソケットサービス要求処理完了時に処理結果が格納されます。

以降、ソケットサービスパラメータエリア 2 ～ 8 も同様の項目となります。
（ソケットサービスパラメータエリア 2 ～ 7 については表の記載を省略します。）

ソケットサービスパラメータエリア 8

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt8ParamArea	ARRAY [0..9] OF WORD	R/W	ソケットサービスパラメータエリア 8	ソケットのサービスのパラメータを設定します。
*_Skt8No	USINT	R/W	UDP/TCP ソケット No.	オープンする UDP または TCP のソケット No. を指定します。
*_Skt8Option	BYTE	R/W	ソケットオプション	TCP ソケットのオープン要求コマンドで、keep-alive 機能の仕様を設定します。 オープンする UDP または TCP のソケット No. を指定します。
*_Skt8KeepAlive	BOOL	R/W	Keep-alive 機能使用	TCP ソケットのオープン要求（PASSIVE/ACTIVE）コマンドで、keep-alive 機能の使用を指定します。 TRUE: 使用する FALSE: 使用しない
*_Skt8Linger	BOOL	R/W	Linger 機能使用	linger 機能の使用を指定します。 TRUE: 使用する FALSE: 使用しない
*_Skt8LocalPortNo	UINT	R/W	自 UDP/TCP ポート No.（0000 ～ FFFF Hex）	ソケットが送受信を行う UDP または TCP ポート No. を指定します。
*_Skt8RemoteIPAdr	DWORD	R/W	相手 IP アドレス (00000000 ～ FFFFFFFF Hex)	相手先の IP アドレスを指定します。
*_Skt8RemoteIPAdr1	USINT	R/W	相手 IP アドレスの 1 バイトめ (16#00 ～ 16#FF)	相手 IP アドレスの 1 バイトめを指定します。
*_Skt8RemoteIPAdr2	USINT	R/W	相手 IP アドレスの 2 バイトめ (16#00 ～ 16#FF)	相手 IP アドレスの 2 バイトめを指定します。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_Skt8RemoteIPAdr3	USINT	R/W	相手 IP アドレスの 3 バイトめ (16#00 ～ 16#FF)	相手 IP アドレスの 3 バイトめを指定します。
*_Skt8RemoteIPAdr4	USINT	R/W	相手 IP アドレスの 4 バイトめ (16#00 ～ 16#FF)	相手 IP アドレスの 4 バイトめを指定します。
*_Skt8RemotePortNo	UINT	R/W	相手 UDP/TCP ポート No.（0000 ～ FFFF Hex）	相手先の UDP または TCP のポート No. を指定します。
*_Skt8SendRcvByte	UINT	R/W	送信／受信バイト数 (0000 ～ 07C0 Hex(1984))	送信要求または受信要求時の、送信または受信データのバイト数を指定します。
*_Skt8SendRcvAdr	DWORD	R/W	送信／受信データアドレス	送信要求時の送信データの格納先先頭アドレス、または受信要求時の受信データの格納先先頭アドレスを指定します。
*_Skt8SendRcvAdrType	BYTE	R/W	エリア種別 (#16#00 ～ #16#FF)	エリア種別を指定します。
*_Skt8SendRcvAdrCh1	BYTE	R/W	チャネルアドレス（上位 2 けた）	チャネルアドレスの上位 2 けたを指定します。
*_Skt8SendRcvAdrCh2	BYTE	R/W	チャネルアドレス（下位 2 けた）	チャネルアドレスの下位 2 けたを指定します。
*_Skt8Timeout	UINT	R/W	タイムアウト値 (0000 ～ FFFF Hex)	受信（TCP、UDT）、オープン（PASSIVE）時ソケットサービス要求スイッチを TRUE にしてから、完了するまでの制限時間を指定します。
*_Skt8RespCode	WORD	R	終了コード	ソケットサービス要求処理完了時に処理結果が格納されます。

ソケット強制クローズスイッチ

「3-2-2 ソフトスイッチ用の CJ ユニットデバイス変数」のソケット強制クローズスイッチ (*_SktForceCloseCmd) を参照してください。

3-3　ステータスエリアのユーザ定義設定

3-3-1　ユーザ定義設定の概要

本ユニットがオリジネータのコネクションにおいて、該当ターゲットノードのステータス情報を、デフォルトで決まっている CJ ユニットデバイス変数ではなく、任意のユーザ定義変数に格納することができます。
デフォルトの CJ ユニットデバイス変数では、ノード 0 ～ 63 のみの情報しか表示されませんが、ユーザ定義設定のエリアにすると、256 のノード情報を表示させることができます。

参考

本エリアの情報は、正常ターゲットノード情報が TRUE のときのみ有効となります。
FALSE の場合は直前の値が保持されます。

3-3-2　ユーザ定義設定の設定方法

1 **Sysmac Studio の［構成・設定］の［CPU・増設ラック］でユニットを選択した状態で右クリックし、［ユニット用データ型登録］を選択します。この操作によりデータ型：_sCJEIP21_User_StaArea が利用可能になります。**

2 **Sysmac Studio のマルチビューエクスプローラで［プログラミング］-［データ］-［グローバル変数］を選択します。**

3 **グローバル変数テーブル内で、名称：(任意)、データ型：_sCJEIP21_User_StaArea、ネットワーク公開：" 公開のみ " または " 出力 " のユーザ定義変数を設定します。**



4 **Sysmac Studio の［構成・設定］の高機能ユニット設定の［ステータスエリア設定］ダイアログのレイアウトタイプを「ユーザ設定」にします。**

5 **割付エリアに手順 3 で設定した変数を指定します。**



3-3-3　ユーザ設定のステータスエリアへのアクセス方法

ユーザプログラムからは、ステータスエリアに指定したユーザ定義変数を介して、ステータスエリアにアクセスします。

ステータスエリア用のデータ型について

● ステータスエリア用のデータ型（_sCJEIP21_User_StaArea）

ステータス情報の種類ごとにメンバに分かれています。
各メンバは、ターゲットノード情報が格納できるデータ型（_uCJEIP21_TargetSta）で構成されています。

構造体データ型名	メンバ	データ型
_sCJEIP21_User_StaArea	RegTargetSta （登録ターゲットノード）	_uCJEIP21_TargetSta
	EstbTargetSta （正常ターゲットノード）	_uCJEIP21_TargetSta
	TargetPLCMdSta （ターゲットノード PLC 本体運転情報）	_uCJEIP21_TargetSta
	TargetPLCErrSta （ターゲットノード PLC 本体異常情報）	_uCJEIP21_TargetSta

ターゲットノード情報格納用のデータ型（_uCJEIP21_TargetSta）
ノード 0 ～ 255 までの情報を格納できます。
共用体型の変数になっており、複数の異なるデータ型でアクセスできます。
アクセス可能な複数の異なるデータ型は、要素数 256 の BOOL 型の配列指定や 256 個の BOOL 型と、WORD 型となります

構造体データ型名	メンバ	データ型
_uCJEIP21_TargetSta	TargetSta	array[0..255] of BOOL
	TargetStaWd	array[0..15] of WORD

ステータスエリアを割り付ける任意のユーザ定義変数について

ステータスエリアを割り付ける任意のユーザ定義変数は、以下のように設定します。

1 **Sysmac Studio のマルチビューエクスプローラで［プログラミング］–［データ］–［グローバル変数］を選択します。**

2 **グローバル変数テーブル内で、名称：（任意）、データ型：_sCJEIP21_User_StaArea、ネットワーク公開：公開のみ（または出力）のユーザ定義変数を設定します。**

3-3-4　ユーザ定義設定エリアに割り付けるステータス情報

登録ターゲットノード（ユニット→ CPU ユニット）

本エリアでは、該当ターゲットノードの登録情報をあらわします。
本エリアは、本ユニットがオリジネータとなっている場合のみ有効となります。

x. RegTargetSta. TargetSta [0] ～ [255]
x. RegTargetSta. TargetStaWd [0] ～ [15]
x：ユーザ設定で、ステータスエリアに指定したユーザ定義変数名

ビット	名称	状態	操作する側	ユニット動作	参照
－	登録ターゲットノード	TRUE	ユニット	タグデータリンクが登録されていることを示す	「3-2-3 ステータス用の CJ ユニットデバイス変数」
		FALSE	ユニット	タグデータリンクが登録されていないことを示す	

正常ターゲットノード（ユニット→ CPU ユニット）

本エリアでは、該当ターゲットノードとのコネクション確立情報をあらわします。
ターゲットデバイス単位で複数あるコネクションのすべてのデータを、CPU ユニットへリフレッシュしてから TRUE されます。

x. EstbTargetSta. TargetSta [0] ～ [255]
x. EstbTargetSta. TargetStaWd [0] ～ [15]
x：ユーザ設定で、ステータスエリアに指定したユーザ定義変数名

ビット	名称	状態	操作する側	ユニット動作	参照
－	正常ターゲットノード	TRUE	ユニット	ターゲットデバイス単位で複数あるコネクションの全データを、CPU ユニットへリフレッシュしてから TRUE	「3-2-3 ステータス用の CJ ユニットデバイス変数」
		FALSE	ユニット	上記以外	

ターゲットノード PLC 本体運転情報（ユニット→ CPU ユニット）

本エリアでは、該当ターゲットノードの PLC 本体の運転情報をあらわします。
本エリアは、本ユニットがオリジネータとなっている場合のみ有効となります。
本エリアの情報は、正常ターゲットノード情報が TRUE のときのみ有効となります。FALSE の場合は直前の値が保持されます。

x. TargetPLCMdSta. TargetSta [0] ～ [255]
x. TargetPLCMdSta. TargetStaWd [0] ～ [15]
　x：ユーザ設定で、ステータスエリアに指定したユーザ定義変数名

ビット	名称	状態	操作する側	ユニット動作	参照
－	ターゲットノード PLC 本体運転情報	TRUE	ユニット	該当ターゲットノードの PLC 本体が運転状態	「3-2-3 ステータス用の CJ ユニットデバイス変数」
		FALSE	ユニット	上記以外	

ターゲットノード PLC 本体異常情報（ユニット→ CPU ユニット）

本エリアでは、該当ターゲットノードの CPU ユニットのコントローラ異常（全停止／部分停止／軽度フォールトの論理和）をあらわします。
本エリアは、本ユニットがオリジネータとなっている場合のみ有効となります。
本エリアの情報は、正常ターゲットノード情報が TRUE のときのみ有効となります。FALSE の場合は直前の値が保持されます。

x. TargetPLCErrSta. TargetSta [0] ～ [255]
x. TargetPLCErrSta. TargetStaWd [0] ～ [15]
　x：ユーザ設定で、ステータスエリアに指定したユーザ定義変数名

ビット	名称	状態	操作する側	ユニット動作	参照
－	ターゲットノード PLC 本体異常情報	TRUE	ユニット	該当ターゲットノードの PLC 本体に全停止／部分停止／軽度フォールトの論理和が発生中	「3-2-3 ステータス用の CJ ユニットデバイス変数」
		FALSE	ユニット	上記以外	

4

IP アドレスを決定する方法

4-1 IP アドレスとは 4-2
4-1-1 IP アドレスの構成 4-2
4-1-2 IP アドレスの割り当て 4-3
4-1-3 サブネットマスク 4-3
4-1-4 CIDR 機能 4-3

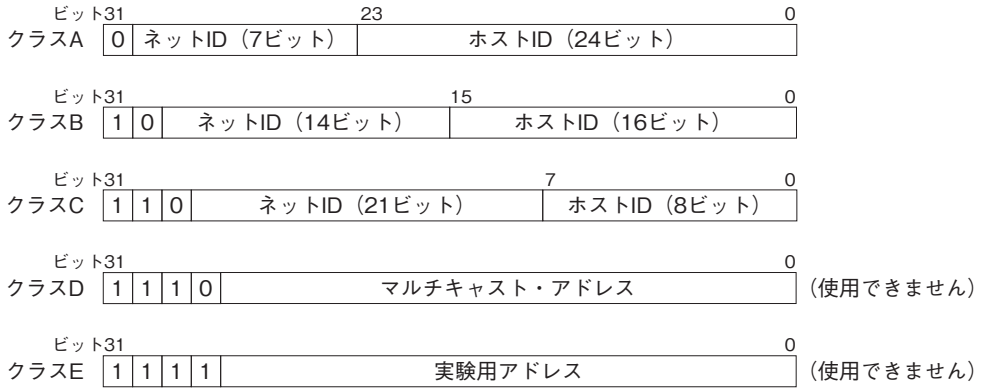
4-2 EtherNet/IP ユニットの自 IP アドレス 設定方法 4-4
4-2-1 IP アドレスを設定する方法について 4-4
4-2-2 IP アドレスの設定手順 4-5
4-2-3 オンライン接続する 4-7

4-3 プライベート／グローバルアドレス 4-10
4-3-1 グローバルアドレスとプライベートアドレス 4-10
4-3-2 EtherNet/IP ユニットがプライベートアドレスの場合 4-11
4-3-3 EtherNet/IP ユニットがグローバルアドレスの場合 4-12

4-1　IP アドレスとは

4-1-1　IP アドレスの構成

IP アドレスは 32 ビットのバイナリデータで、ネット ID とホスト ID から構成されています。ネット ID とはネットワークを識別するためのアドレスで、ホスト ID とはホスト（ノード）を識別するためのアドレスです。
IP アドレスは、ネットワークの規模に応じてアドレス体系が選択できるように、クラス A ～クラス C の 3 つのクラスに分かれています（クラス D とクラス E は使用できません）。



クラスにより、識別できるネットワーク数とホスト数が異なります。

クラス	ネットワーク数	ホスト数
クラス A	小	$2^{24} - 2$ 以下（16777214 以下）
クラス B	中	$2^{16} - 2$ 以下（65534 以下）
クラス C	大	$2^{8} - 2$ 以下（254 以下）

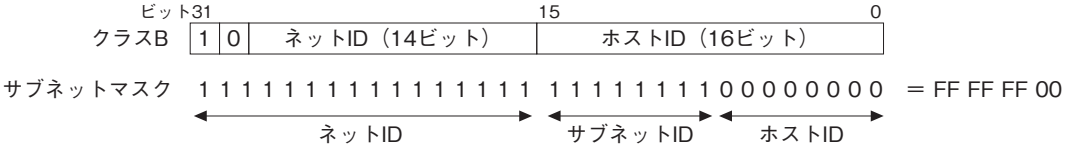
IP アドレスは、32 ビットのバイナリデータを 8 ビットずつの 4 つの組に分け、それぞれをドットで区切った 4 つの 10 進数で表記します。
（例）10000010　00111010　00010001　00100000　→　130.58.17.32

4-1-2　IP アドレスの割り当て

IP アドレスは、ネットワーク内（または複数のネットワーク間）で重複がないようにノードに割り当てる必要があります。

4-1-3　サブネットマスク

1 つのネットワークに数多くのノードを接続すると、ネットワークの運用や管理が大変になります。このような場合、1 つのネットワークを複数のサブネットワークに分けてシステムを構築すると便利です。内部的には複数のネットワークとして運用しますが、外部からは 1 つのネットワークとして見えるようになります。
このような運用を行うためには、IP アドレスのホスト ID 部をサブネット ID 部とホスト ID 部の 2 つに分けて使用します。
このサブネット ID 部を識別するための情報がサブネットマスクです。サブネットマスクは、ネット ID 部とサブネット ID 部を「1」に、ホスト ID 部を「0」にビットマスクしたものです。
（例）クラス B の IP アドレスで 8 ビットをサブネット ID として使用する場合のサブネットマスク



サブネットマスクは、そのサブネットワーク上のすべてのノードが共通のサブネットマスク値を設定してください。EtherNet/IP ユニットでは、CIDR（Classless Inter-Domain Routing）に対応しているためサブネットマスクには、192.0.0.0 ～ 255.255.255.252 まで設定が可能です。
サブネットワークを使用しない場合は、IP アドレスのクラス A ～クラス C によって、以下のサブネットマスク値を設定してください。

クラス	サブネットマスク値
クラス A	255.0.0.0
クラス B	255.255.0.0
クラス C	255.255.255.0

4-1-4　CIDR 機能

CIDR とは Classless Inter-Domain Routing の略で、クラスを使わない IP アドレスの割り当てを行う機能です。
クラスを使用した IP アドレスでは、ネット ID 部とホスト ID 部が決められたブロック単位で区切られるので、IP アドレス空間（ホスト数）の利用が効率良く割り当てられないデメリットがあります。CIDR ではクラスを使わないため、任意の IP アドレス空間で区切ることができ、IP アドレス空間（ホスト数）を効率良く使用することができます。
例えば、クラス C（192.168、･･･）を使用しても CIDR によるネットマスク設定をすることで、254 ノードを超える水平分散ネットワークを構築することが可能になります。

サブネットマスク範囲
192.0.0.0 ～ 255.255.255.252

4-2　EtherNet/IP ユニットの自 IP アドレス設定方法

4-2-1　IP アドレスを設定する方法について

EtherNet/IP ユニットの自 IP アドレスは、以下のいずれかの方法で設定します。

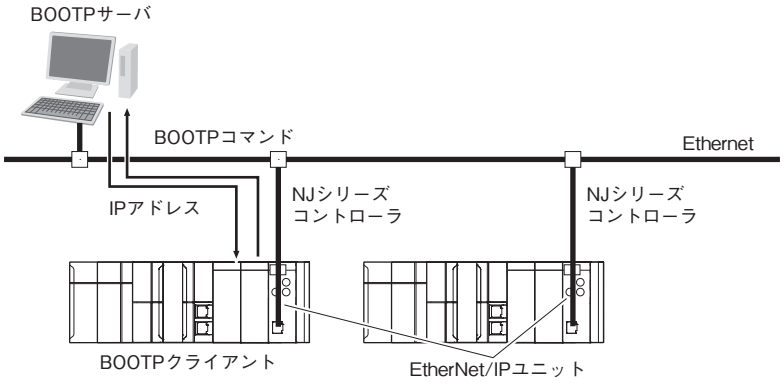
● 任意の IP アドレスを設定する

EtherNet/IP ユニットのデフォルトの IP アドレスを変更するとき、あるいは EtherNet/IP ユニットと他の EtherNet/IP ノードを使用するときに、IP アドレスを任意に設定します。

● BOOTP サーバから自動で IP アドレスを取得する

IP アドレスを自動で取得する方法は、以下の 2 種類から選択できます。

- 電源投入ごとに BOOTP サーバから取得する
- 一度電源投入して BOOTP サーバから取得した IP アドレスで固定する

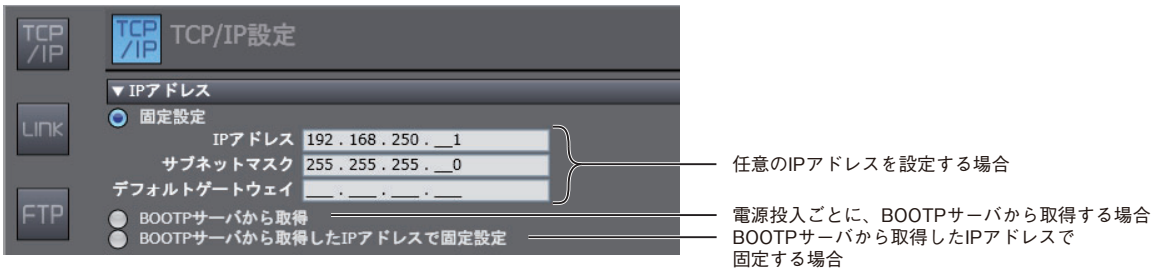


4-2-2　IP アドレスの設定手順

EtherNet/IP ユニットの自 IP アドレスは、Sysmac Studio で設定します。

1　IP アドレスの設定方法を選択します。

Sysmac Studio のユニット構成に登録した EtherNet/IP ユニットの高機能ユニット設定にて、[TCP/IP] 設定の IP アドレスを設定します。



ユニット設定の詳細は、📖「5-2 ［TCP/IP 設定］ダイアログ」を参照してください。

2　Sysmac Studio を、NJ CPU ユニットに USB 接続または Ethernet 経由で接続します。

3　Sysmac Studio を、NJ CPU ユニットにオンライン接続します。

オンライン接続の方法については、📖「4-2-3 オンライン接続する」を参照してください。

4　Sysmac Studio で設定した IP アドレス設定を、以下のいずれかの方法でオンライン接続した NJ CPU にダウンロードします。

- **コントローラとオンライン接続の上、[コントローラ] メニューから [同期] を選択します。**
  パソコン上のデータと実機コントローラのデータが自動照合されます。
- **[転送 [パソコン→コントローラ]] ボタンをクリックします。**

（注） 転送（ダウンロードおよびアップロード）には、Sysmac Studio の「同期機能」を使用します。

5　EtherNet/IP ユニットをリスタートさせます。

［高機能ユニット設定の編集］の［リスタート］ボタンをクリックします。

6　リスタート時、EtherNet/IP ユニットの 7 セグメント LED を確認します。

7 セグメント LED がいったん消灯した後、再び LED テスト、IP アドレス表示が行われれば、EtherNet/IP ユニットは、新たな TCP/IP 設定の内容（ここでは IP アドレス）を認識しています。

7　設定した IP アドレスは、以下のように EtherNet/IP ユニットに反映されます。

- **任意の IP アドレスを設定した場合**
  ダウンロード後、設定した IP アドレスが自動的に EtherNet/IP ユニットに反映されます。
- **電源投入ごとに BOOTP サーバから取得する場合**
  ダウンロード後、BOOTP サーバから取得した IP アドレスが自動的に EtherNet/IP ユニットに反映されます。
  また、電源投入ごとに BOOTP サーバから取得した IP アドレスが自動的に EtherNet/IP ユニットに反映されます。

参考

BOOTP サーバから取得したIPアドレスが不正な値の場合や、IP アドレス取得に失敗した場合は、［IP アドレス設定］で［固定設定］を選択し、手動で IP アドレス、サブネットマスク、デフォルトゲートウェイを設定してください。
BOOTP サーバへの接続に失敗した場合は、BOOTP サーバへの取得要求を続けます。

- **一度電源投入して BOOTP サーバから取得した IP アドレスで固定する場合**
  ダウンロード後、BOOTP サーバから取得した IP アドレスが自動的に EtherNet/IP ユニットに反映され、「固定設定」となります。

参考

- BOOTP サーバから正常に IP アドレスを取得しても、［TCP/IP］ダイアログの表示は更新されません。
  BOOTP サーバから取得した IP アドレスを確認したい場合は、NJ シリーズコントローラからプロジェクトをアップロードしてコントローラステータスで確認してください。
- BOOTP サーバからの IP アドレス取得に失敗した状態でコントローラ電源 OFF → ON した場合、「固定設定」のデフォルト値となります。
- 「BOOTP サーバから取得した IP アドレスで固定設定」を選択して、BOOTP サーバからの IP アドレスを取得後、EtherNet/IP ユニットの IP アドレス設定は自動的に「固定設定」となります。

4-2-3　オンライン接続する

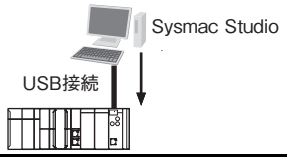
Sysmac Studio から NJ CPU ユニットにオンライン接続します。

● CPU ユニットと Sysmac Studio（パソコン）との接続形態について

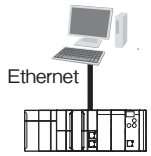
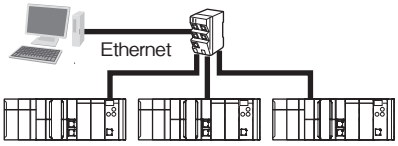
CPU ユニットと Sysmac Studio（パソコン）は、USB または Ethernet 経由で、以下のように接続します。

・ USB 接続する場合

USB －直接接続



・ Ethernet 接続する場合

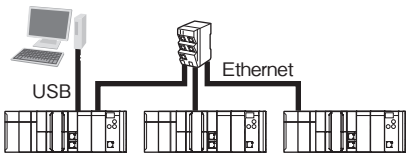
Ethernet －直接接続（AutoIP 機能による 1：1 接続）	Ethernet － HUB 接続（1：N 接続）
	
※ スイッチングハブ無しでも接続できます。 ※Ethernet ケーブルは、ストレートケーブル、クロスケーブルどちらでも接続できます。	※ 接続にはスイッチングハブが必要です。 詳細は 📖「2-3-4 スイッチングハブ選定時の留意事項」を参照してください。

使用上の注意

Sysmac Studio は、本ユニットを経由して NJ シリーズ CPU ユニットにオンライン接続することはできません。

・ USB → Ethernet 階層を越えて接続する場合

USB リモート接続（1：N 接続）



※ 接続にはスイッチングハブが必要です。
詳細は 📖「2-3-4 スイッチングハブ選定時の留意事項」を参照してください。

参考

AutoIP とは、Windows 98 以降の OS が搭載する IP アドレス自動割り当て機能のことで、AutoIP 用にあらかじめ予約された「169.254.0.0 ～ 169.254.255.255」のアドレス範囲から、他のノードと重複しない任意の IP アドレスを自動で割り当てます。

使用上の注意

EtherNet/IP ネットワーク上で、同じ IP アドレスが設定されているノードが複数存在する場合、EtherNet/IP ユニットは、先に検出したノードに対して接続します。
その場合、IP アドレス重複異常などは検出されませんので。ご注意ください。

● オンライン接続手順

CPU ユニットと Sysmac Studio（パソコン）を、USB または Ethernet 経由で接続後、以下の手順を行います。

1　Sysmac Studio のプロジェクトウィンドウで、「コントローラ」→「通信設定」を選択し［OK］ボタンをクリックします。

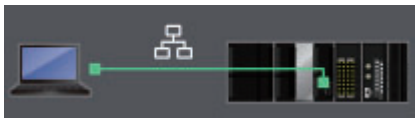
・1：1接続のとき

●直接接続

・USB－直接接続



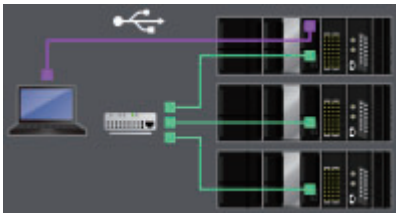
・Ethernet－直接接続



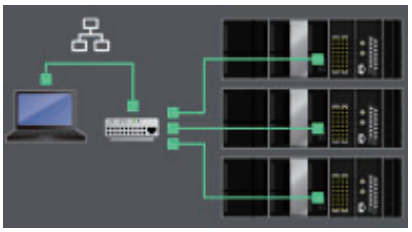
・1：N接続のとき

●Ethernet 接続

・USB リモート接続



・Ethernet－HUB 接続



使用上の注意

- IP アドレスが重複している、または正しく設定されていないと、EtherNet/IP 経由での通信が行えません。
- 以下に示す IP アドレスの範囲はシステムが使用するため、指定できません。
  169.254.0.0 ～ 169.254.255.255
  192.168.255.0 ～ 192.168.255.255
- 以下に示す IP アドレスは Ethernet の制約上、指定できません。
  - 全ビットが 0 または、1 の IP アドレス
  - 127（10 進数）または、0 または、255 で始まる IP アドレス
  - ホスト ID の全ビットが 0 または、1 の IP アドレス
  - ネット ID の全ビットが 0 または、1 の IP アドレス
  - クラス D の IP アドレス（224.0.0.0 ～ 239.255.255.255）
  - クラス E の IP アドレス（240.0.0.0 ～ 255.255.255.255）

● 保存されたプロジェクトからの接続

設定した接続形態（USB接続、EtherNet/IP接続）はプロジェクトに保存されます。（拡張子xxx.smc）保存しておいたプロジェクトを再び Sysmac Studio に読み込み、初めから設定することなくEtherNet/IP ネットワークシステムに接続できます。

● 使用中の IP アドレスの確認方法

EtherNet/IP ユニットに設定した自 IP アドレスは、以下の CJ ユニットデバイス変数により確認できます。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_IPAdrCfg	DWORD	RW	IP アドレス表示・設定エリア	初期値 : 00000000
*_IPAdr1Cfg	USINT	RW	IP アドレス 1	EtherNet/IP ユニットの IP アドレス①部分を格納
*_IPAdr2Cfg	USINT	RW	IP アドレス 2	EtherNet/IP ユニットの IP アドレス②部分を格納
*_IPAdr3Cfg	USINT	RW	IP アドレス 3	EtherNet/IP ユニットの IP アドレス③部分を格納
*_IPAdr4Cfg	USINT	RW	IP アドレス 4	EtherNet/IP ユニットの IP アドレス④部分を格納

自 IP アドレスが以下のように格納されます。

バイト	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
先頭	①	②
先頭+1	③	④

IPアドレス:①.②.③.④（16進数）

自 IP アドレスを読み出す場合の方法については、🕮「3-2-4 システム設定用の CJ ユニットデバイス変数」を参照してください。

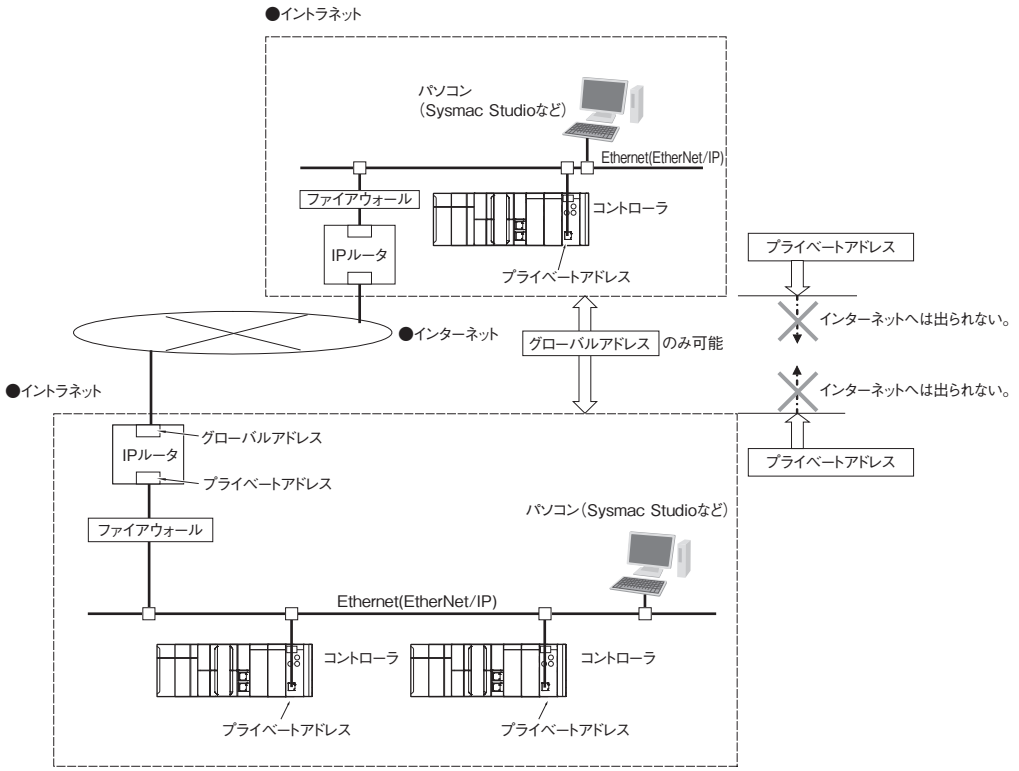
4-3　プライベート／グローバルアドレス

4-3-1　グローバルアドレスとプライベートアドレス

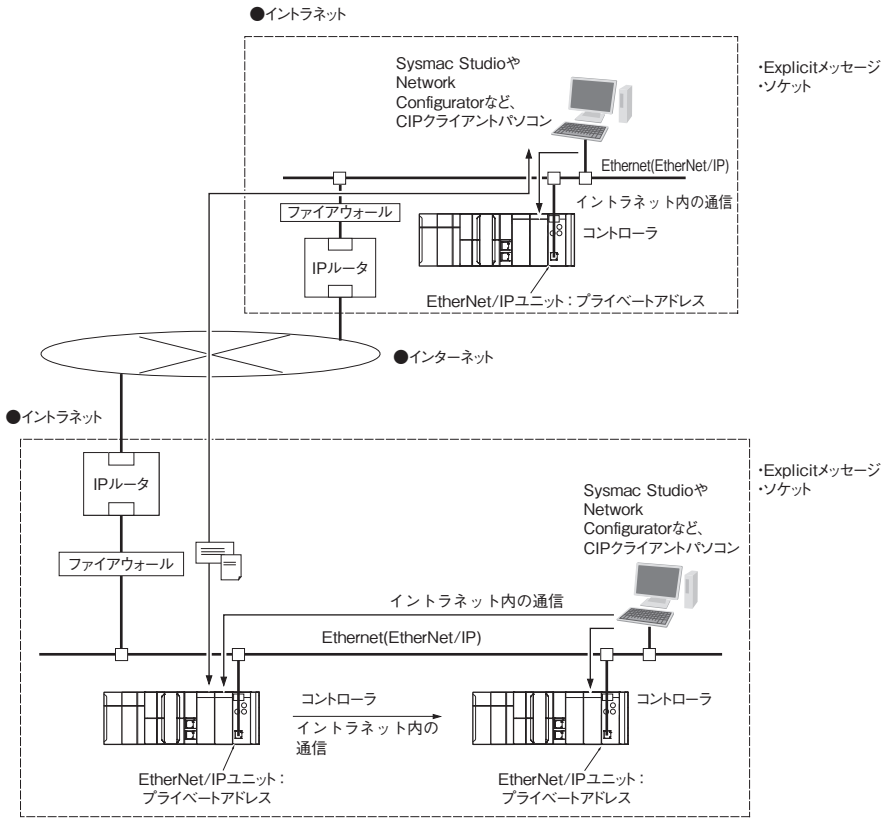
IP アドレスには、以下の 2 種類があります。

グローバルアドレス	インターネットに直接接続できる IP アドレスです。NIC（日本の場合は JPNIC）に申請して割り当てられる、世界中でユニークな IP アドレスで、世界で約 43 億個まで割り当てることができます。
プライベートアドレス	インターネットに直接接続できない、イントラネット内部（社内 LAN）用の IP アドレスです。この IP アドレスを含むフレームは、ルータによって外部に出ることが制限されていますので、インターネットに直接出て行くことはできません。

一般的に、以下のように、イントラネット内では、グローバルアドレスは、インターネットとの接続インタフェースにあたる IP ルータ（ブロードバンドルータなど）にだけ割り当てられます。それ以外のイントラネット内の、EtherNet/IP ユニットを含めたすべてのノードには、プライベートアドレスが割り当てられます。



4-3-2　EtherNet/IP ユニットがプライベートアドレスの場合



● 通信アプリケーションの使用条件

EtherNet/IP ユニットがプライベートアドレスの場合、以下の条件で、Explicit メッセージ通信サービスを使用できます。

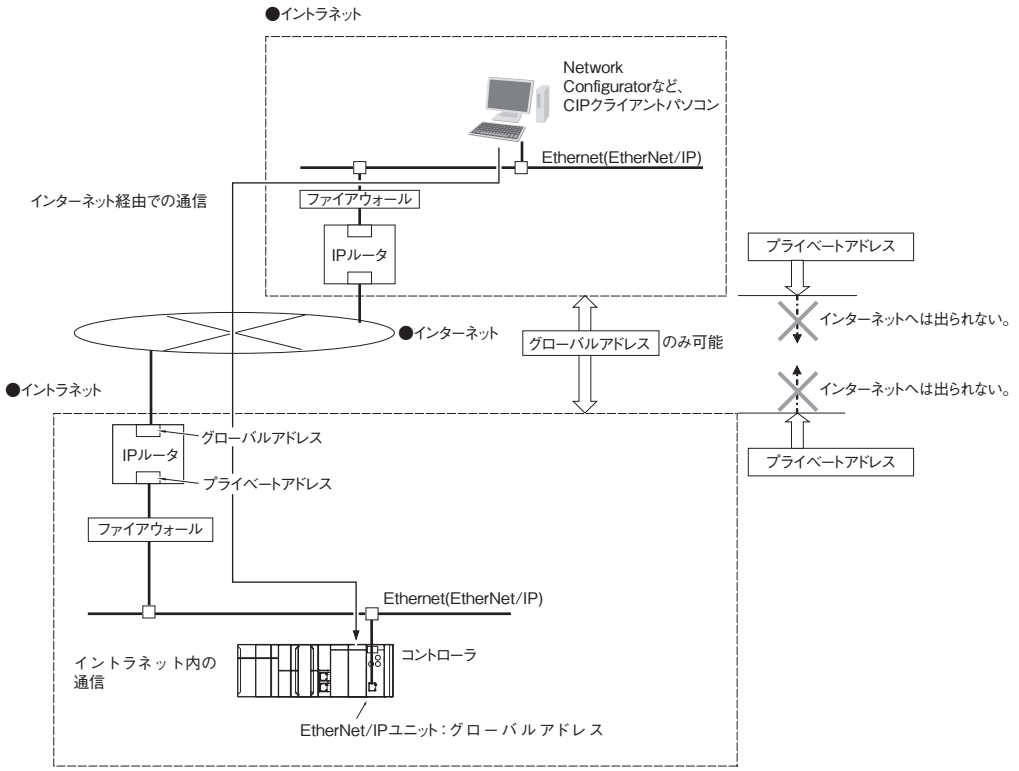
- イントラネット内のプライベートアドレスの EtherNet/IP ユニット間でのみ、Explicit メッセージ通信サービスが可能です。
- パソコン（Network Configurator を含む CIP アプリケーション）などから、インターネットを介して、プライベートアドレスの EtherNet/IP ユニットに対してオンライン接続して通信することはできません。
  また、インターネットを介して、プライベートアドレスの EtherNet/IP ユニット同士の Explicit メッセージ通信もできません。

使用上の注意

• ネットワークセキュリティとファイアウォール

グローバルアドレスを介してイントラネットを構築する場合は、ネットワークセキュリティの問題から、ネットワーク分野に精通したスペシャリストに十分相談のうえ、ファイアウォールの設置を検討してください。
逆に、通信事業者によるファイアウォールの設定から、使用できない通信アプリケーションがある場合があります。必ず、通信事業者に確認してください。

4-3-3　EtherNet/IP ユニットがグローバルアドレスの場合



● 通信アプリケーションの使用条件

インターネットを介した場合、以下の条件で、Explicit メッセージ通信サービスを使用できます。

- パソコン（Network Configurator を含む CIP アプリケーション）などから、インターネットを介して、グローバルアドレスの EtherNet/IP ユニットに対してオンライン接続して通信することができます。
- EtherNet/IP で使用する TCP ポート番号 (44818) または UDP ポート番号 (44818) が、通信経路の過程でファイアウォールにより使用禁止となっている場合は、使用できません。

使用上の注意

• ネットワークセキュリティとファイアウォール

EtherNet/IP ユニットにグローバル IP アドレスを設定する場合は、ネットワークセキュリティの問題から、ブロードバンドなどの一般回線ではなく、通信事業者との個別契約となる専用回線による利用をお勧めします。また、ネットワークセキュリティなどには、ファイアウォールなど、ネットワーク分野に精通したスペシャリストに十分相談のうえ、設置を検討するようにしてください。
逆に、通信事業者によるファイアウォールの設定から、使用できない通信アプリケーションがある場合があります。必ず、通信事業者に確認してください。

5

Sysmac Studio による EtherNet/IP ユニットのユニット設定一覧

5-1 EtherNet/IP ユニットのユニット設定について 5-2
5-1-1 ユニット設定の反映について 5-2
5-1-2 ユニット設定をデフォルトに戻す 5-2
5-2 ［TCP/IP 設定］ダイアログ 5-3
5-3 ［LINK 設定］ダイアログ 5-8
5-4 ［FTP 設定］ダイアログ 5-9
5-5 ［SNTP 設定］ダイアログ 5-10
5-6 ［SNMP 設定］ダイアログ 5-11
5-7 ［SNMP トラップ設定］ダイアログ 5-13
5-8 ［ステータスエリア設定］ダイアログ 5-15

5-1　EtherNet/IP ユニットのユニット設定について

ユニット設定は、Sysmac Studio の［高機能ユニット設定の編集］で設定します。

5-1-1　ユニット設定の反映について

ユニット設定を、EtherNet/IP ユニットへ反映させるには、Sysmac Studio から CPU ユニットにユニット設定を転送した後、以下のいずれかの方法を行ってください。
それにより、CPU ユニットから本ユニットへ転送されます。

● コントローラをリセットする

コントローラをリセットすると、CPU ユニット起動時に、ユニット設定が CPU ユニットから本ユニットへ転送されます。

● EtherNet/IP ユニットをリスタート

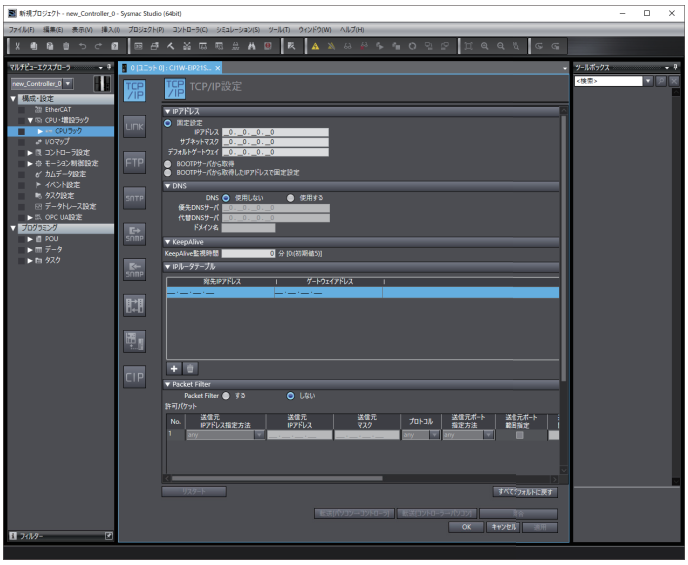
EtherNet/IP ユニットを以下の方法でリスタートさせると、ユニット設定が CPU ユニットから本ユニットへ転送されます。

- ［高機能ユニット設定の編集］の［ リスタート ］ボタンをクリックする
- システム定義変数のユニットのリスタートフラグ（_CJB_CBU**Restart）を、FALSE → TRUE する

5-1-2　ユニット設定をデフォルトに戻す

設定したユニット設定の内容を、デフォルトに戻すことができます。
［高機能ユニット設定の編集］の［リスタート］ボタンをクリックした後、ユニットのリスタートタイプを「出荷時設定に戻して再起動」を選択して、リスタートさせます。

5-2　[TCP/IP 設定] ダイアログ



● IP アドレス

項目名	内容	デフォルト値
IP アドレス設定方法	EtherNet/IP ユニットの IP アドレスの設定方法を次のうちから選択します。*1 • 固定設定 • BOOTP サーバから取得 • BOOTP サーバから取得した IP アドレスで固定設定	固定設定
IP アドレス *2	EtherNet/IP ユニットの自 IP アドレスを設定します。	なし *3
サブネットマスク *2	EtherNet/IP ユニットのサブネットマスクを設定します。	なし
デフォルトゲートウェイ *4	EtherNet/IP ユニットのデフォルトゲートウェイ IP アドレスを設定します。 デフォルトゲートウェイを使用しない場合は、設定する必要はありません。	なし

*1　IP アドレスの設定についての詳細は、「4-2 EtherNet/IP ユニットの自 IP アドレス 設定方法」を参照してください。
*2　「IP アドレス」を「固定設定」とした場合に、設定が必要です。
*3　デフォルト値は、EtherNet/IP ユニット前面のノードアドレスのロータリスイッチで設定されます。
*4　「IP アドレス設定方法」を「固定設定」にした場合に、本設定が有効です。

● DNS

項目名	内容	デフォルト値
DNS を使用する／ DNS を使用しない	CIP 通信命令やソケット命令、NTP サーバ設定などホスト名で通信相手を指定するときに、ホスト名の名前解決に DNS を使用する場合は、「DNS を使用する」を選択します。 DNS を使用するには、別途 DNS サーバが必要です。	DNS を使用しない
優先 DNS サーバ *1、*2	DNS サーバの IP アドレスを設定します。 IP アドレスは優先と代替の 2 個まで設定できます。	なし
代替 DNS サーバ		なし
ドメイン名 *1	EtherNet/IP ユニットが属するドメインのドメイン名を設定します。 EtherNet/IP ユニットが実際の通信でドメイン名を使用することはありません。 （半角英数字、ドット、ハイフン：48 文字以内）	なし

*1　「DNS を使用する／ DNS を使用しない」を「DNS を使用する」とした場合に、設定が必要です。
*2　以下に示す IP アドレスは Ethernet の制約上、指定できません。
- 127（10 進数）または、0 または、255 で始まる IP アドレス
- クラス D の IP アドレス（224.0.0.0 ～ 239.255.255.255）
- クラス E の IP アドレス（240.0.0.0 ～ 255.255.255.255）

● keep-alive（形 CJ1W-EIP21S のみ）

項目名	内容	デフォルト値
監視時間（分）	生存確認（keep-alive）監視時間を設定します。 FINS/TCP または TCP/IP でのソケットサービス使用時、相手ノード（サーバまたはクライアント）がここで設定した時間以上アイドリング（無応答）状態を継続すると、コネクションを切断します（FINS/TCP の場合または TCP/IP でのソケットサービスの場合のみ）。 設定範囲：0 ～ 65535 分（初期値は 0 で、そのときの監視時間は 5 分となります。） ［FINS/TCP］タブで設定する各コネクション No. ごとの keep-alive（相手ノード生存確認機能使用有無）に共通の設定です。	0（初期値 5 分）

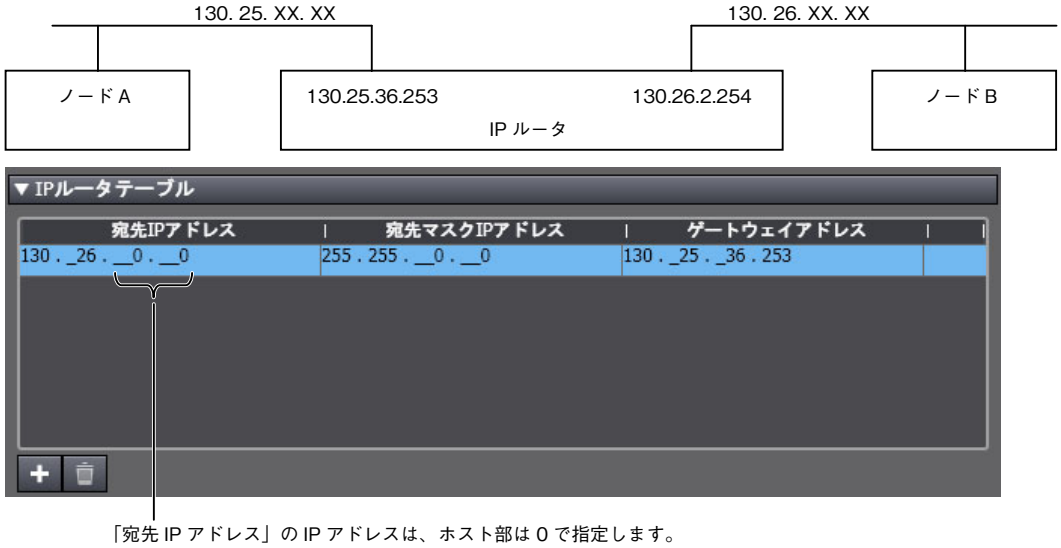
● IP ルータテーブル

項目名	内容	デフォルト値
宛先 IP アドレス	EtherNet/IP ユニットが IP ルータを介して他の IP ネットワークセグメント上のノードとタグデータリンクや CIP メッセージ通信をするときに、設定します。 IP アドレスとゲートウェイアドレスの組み合わせを最大 8 つ設定できます。 IP アドレスのホスト部には、0 を指定します。 IP ルータテーブルの設定例については、以下を参照ください。	なし
ゲートウェイアドレス		なし

参考

IP ルータテーブルの設定例

ノード A がノード B と、IP ルータを介して、タグデータリンクや CIP メッセージ通信を行う場合、ノード A には IP ルータテーブルを、下記のように設定します。
IP ルータテーブルを設定することにより、ノード A からノード B 宛てに通信命令を実行した場合、ノード A はゲートウェイ IP アドレス（130.25.36.253）宛てにパケットを送信します。



● Packet Filter（形 CJ1W-EIP21S のみ）

項目名	内容	デフォルト値
Packet Filter	Packet Filter を使用するか、しないかを選択します。	使用しない
許可パケット	許可するパケットを設定します。 最大 32 エントリを設定できます。	1 エントリ
送信元 IP アドレス指定方法	送信元 IP アドレスの指定方法を選択します。 • any：すべての IP アドレスを許可する。 • IP アドレス指定：指定された IP アドレスを許可する。	any
送信元 IP アドレス	送信元 IP アドレス指定方法が IP アドレス指定のときに、IP アドレスを設定します。 なお、［マスク］でマスクに設定したビット以降を「0」に設定してください。	なし
送信元マスク	送信元 IP アドレス指定方法が IP アドレス指定のときに、マスクを設定します。	なし
プロトコル	許可するプロトコルを選択します。 • any：すべてのプロトコルを許可します。 • TCP：TCP を許可する。 • UDP：UDP を許可する。 • igmp：igmp を許可する • icmp：icmp を許可する	any
送信元ポート指定方法	プロトコルが TCP または UDP のときに、許可する送信元ポートの指定方法を選択します。 • any；すべてのポートを許可する。 • ポート指定：指定されたポートを許可する。	any
送信元ポート範囲指定	送信元ポート指定方法がポート指定のときに、送信元ポートを範囲指定するかを選択します。 • OFF：範囲指定しない。 • ON：範囲指定する。	OFF
送信元ポート開始番号	送信元ポート指定方法がポート指定のときに、送信元ポートの開始番号を設定します。 1 ～ 65535 の範囲で指定できます。	なし
送信元ポート終了番号	送信元ポート指定方法がポート指定、かつ送信元ポート範囲指定が ON のときに、宛先ポートの終了番号を設定します。 1 ～ 65535 の範囲で指定できます。 本設定を空白にせずに［送信元ポート範囲指定］のチェックを ON から OFF にした場合でも、本設定は残ります。このとき、次の制約が発生します。 • ユニットパラメータ設定をユニットに転送した後に照合を実施すると照合不一致となります。 • その後、ユニットパラメータ設定をユニットから Sysamc Studio に転送すると、Sysmac Studio での本設定はクリアされます。	なし
宛先ポート指定方法	プロトコルが TCP または UDP のときに、許可する宛先ポートの指定方法を選択します。 • any；すべてのポートを許可する。 • ポート指定：指定されたポートを許可する。	any
宛先ポート範囲指定	宛先ポート指定方法がポート指定のときに、宛先ポートを範囲指定するかを選択します。 • OFF：範囲指定しない。 • ON：範囲指定する。	OFF
宛先ポート開始番号	宛先ポート指定方法がポート指定のときに、宛先ポートの開始番号を設定します。 1 ～ 65535 の範囲で指定できます。	なし
宛先ポート終了番号	宛先ポート指定方法がポート指定、かつ宛先ポート範囲指定が ON のときに、宛先ポートの終了番号を設定します。 1 ～ 65535 の範囲で指定できます。 本設定を空白にせずに［宛先元ポート範囲指定］のチェックを ON から OFF にした場合でも、本設定は残ります。このとき、次の制約が発生します。 • ユニットパラメータ設定をユニットに転送した後に照合を実施すると照合不一致となります。 • その後、ユニットパラメータ設定をユニットから Sysamc Studio に転送すると、Sysmac Studio での本設定はクリアされます。	なし

A. IP アドレスでフィルタする場合の動作説明

設定した IP アドレスのパケット受信を許可します。
［送信元 IP アドレス指定方法］を［Any］に設定した場合、すべての IP アドレスのパケット受信を許可します。
［送信元 IP アドレス指定方法］を［IP アドレス指定］に設定した場合、下位の設定項目の［送信元 IP アドレス］［送信元マスク］設定した IP アドレスのパケット受信を許可します。
なお、［IP アドレス］には［マスク］でマスクに設定したビット以降を「0」に設定してください。

設定の例
- 1 つの IP アドレスを許可する場合
  この場合は［送信元 IP アドレス］に受信許可する IP アドレス、［送信元マスク］に「255.255.255.255」を設定します。
  例えば、192.168.250.100 を受信許可したい場合、［送信元 IP アドレス］に「192.168.250.100」［送信元マスク］に「255.255.255.255」を設定します。
- 複数の IP アドレスを許可する場合
  受信許可する IP アドレスを［送信元 IP アドレス］と［送信元マスク］に入力します。
  例えば、192.168.***.*** を受信許可したい場合、［送信元 IP アドレス］に「192.168.0.0」、［送信元マスク］に「255.255.0.0」を設定します。

B. プロトコルでフィルタする場合の動作説明

設定したプロトコルおよびポートのパケット受信を許可します。
［プロトコル］に［any］を設定した場合、すべての TCP/UDP ポート、IGMP、ICMP のパケット受信を許可します。
［プロトコル］に［TCP］または［UDP］、［送信元ポート指定方法範囲指定］を［any］に設定した場合、ポート番号に関係なくすべての TCP、または、UDP のパケット受信を許可します。
［プロトコル］に［TCP］または［UDP］、［送信元ポート指定方法範囲指定］を［ポート指定］に設定した場合、指定した範囲の TCP ポート番号または、UDP ポート番号のパケット受信を許可します。

設定の例
- 1 つの TCP/UDP ポート番号を許可する場合
  ［送信元ポート範囲指定］のチェックを OFF、［送信元ポート開始番号］に受信を許可する TCP/UDP ポート番号を設定します。
- 複数の TCP/UDP ポート番号を許可する場合
  ［送信元ポート範囲指定］のチェックを ON、［送信元ポート開始番号］と［送信元ポート終了番号］に受信を許可する TCP/UDP ポート番号の範囲を設定します。
  ［送信元ポート開始番号］から［送信元ポート終了番号］までの範囲のポート番号のパケット受信を許可します。

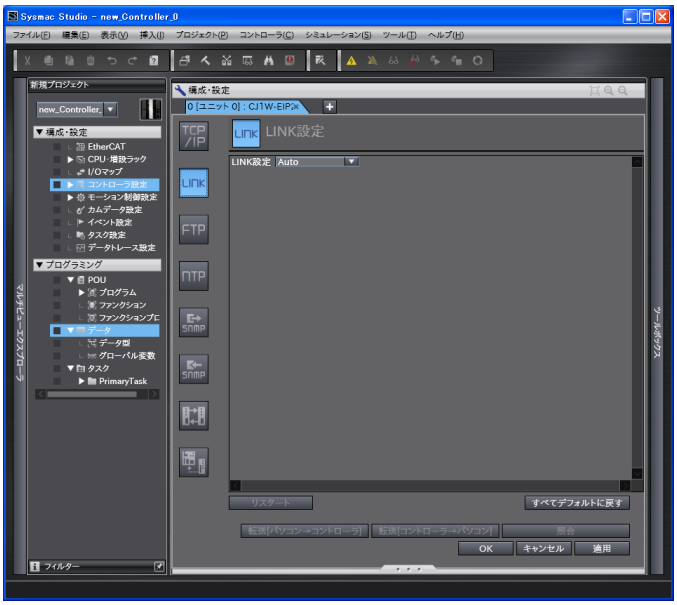
● 一斉同報（形 CJ1W-EIP21S のみ）

項目名	内容	デフォルト値
一斉同報	FINS/UDP での一斉同報時の IP アドレス指定方法を設定します。 • すべて 1(4.3BSD)：ホスト ID がすべて 1 で一斉同報する • すべて 0(4.2BSD)：ホスト ID がすべて 0 で一斉同報する 通常、デフォルトのまますべて 1(4.3BSD) で使用します。	すべて 1 (4.3BSD)

● ソケットサービス（形 CJ1W-EIP21S のみ）

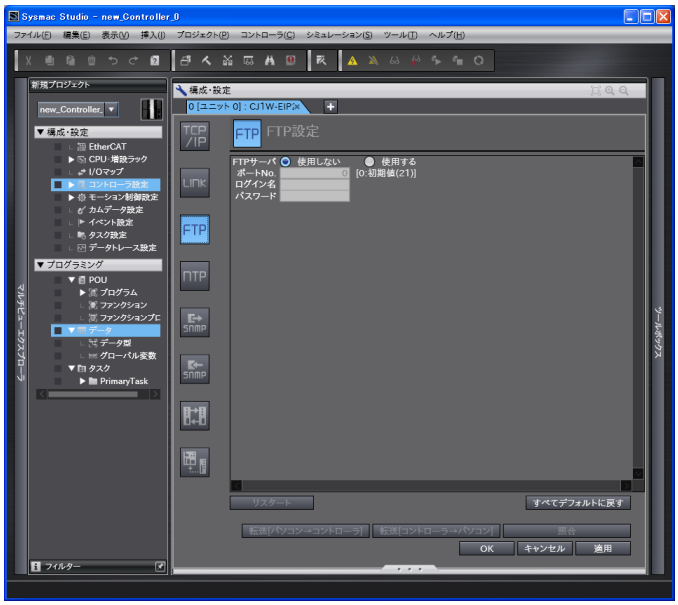
項目名	内容	デフォルト値
高速化する	［高速化する］オプションをチェックすると、ソケットサービスの変数操作による送受信処理性能が改善します。通信性能については、「13-8-5 ソケットサービスの送信／受信に要する時間（ネットワーク上での遅延を無視した場合）」を参照してください。なお、本オプションがチェックのときは、SendCmd 命令によるソケットサービスはエラーとなります。	チェックなし

5-3　[LINK 設定] ダイアログ



項目名	内容	デフォルト値
LINK 設定	EtherNet/IP ユニットの伝送速度を選択します。 • Auto • 10Mbps Half Duplex • 10Mbps Full Duplex • 100Mbps Half Duplex • 100Mbps Full Duplex	Auto

5-4　[FTP 設定] ダイアログ



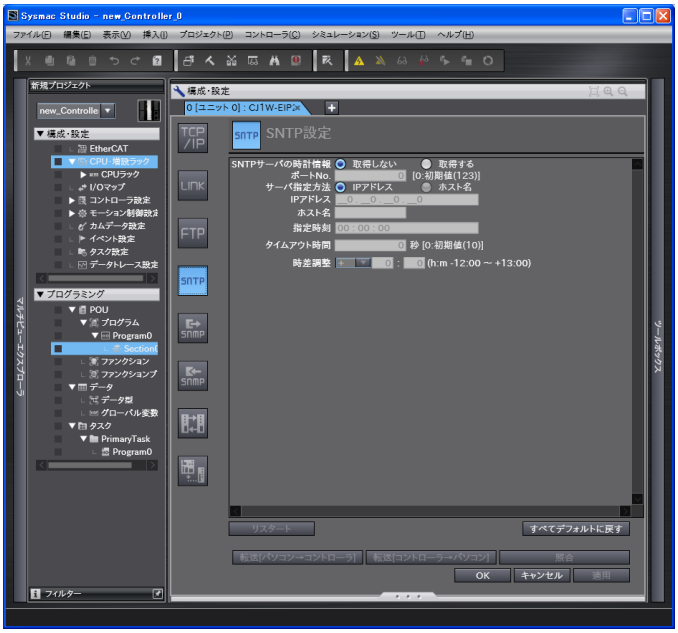
項目名	内容	デフォルト値
FTP サーバ	FTP サーバを使用するか、しないかを選択します。 「使用しない」を選択した場合は、外部からの FTP による接続はできません。	使用しない
ポート No.*1、*2	EtherNet/IP ユニットの FTP ポート No. を設定します。通常、変更する必要はありません。 また、ここで設定できるのは FTP のコントロール用ポートです。データ転送ポートは、ポート No.-1 を使用します。	0 (ポート No.21)
ログイン名 *1	外部から EtherNet/IP ユニットに FTP で接続するためのログイン名を設定します。 形 CJ1W-EIP21 の場合、半角英数字 1 ～ 12 文字 *3 形 CJ1W-EIP21S の場合、半角英数字 1 ～ 16 文字 *3	なし
パスワード *1	外部から EtherNet/IP ユニットに FTP で接続するためのパスワードを設定します。 形 CJ1W-EIP21 の場合、半角英数字 1 ～ 8 文字 *3 形 CJ1W-EIP21S の場合、半角英数字 8 ～ 16 文字 *3	なし

*1.「FTP サーバ」を「使用する」場合に、設定が必要です。
*2. ポート No. 9600、44818 はシステムで使用するため設定できません。
*3. 大文字と小文字は区別されます。

参考

FTP サーバ機能についての詳細は、🕮「第 9 章 FTP サーバ機能」を参照してください。

5-5　[SNTP 設定] ダイアログ



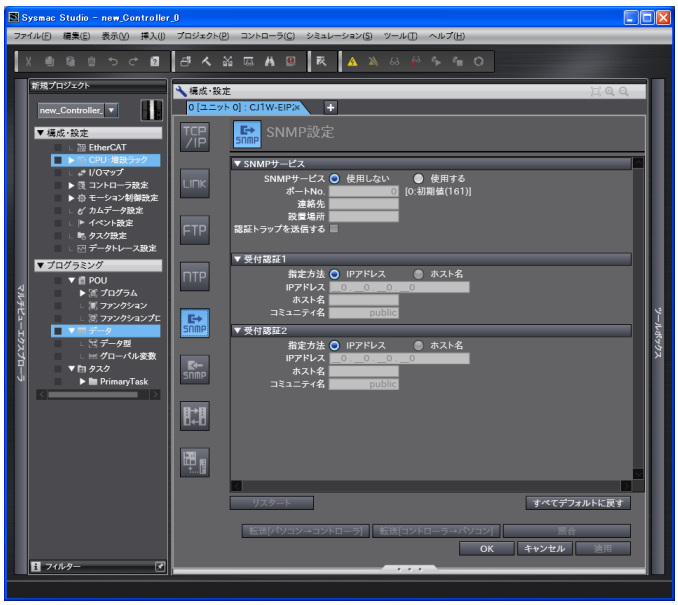
項目名	内容	デフォルト値
SNTP サーバの時計情報	SNTP サーバから時計情報を取得し、取得した時刻を CPU 内蔵時計に反映するか、しないかを選択します。	取得しない
ポート No.*1、*2	時計情報を取得する SNTP サーバとの接続に使うポート No. を設定します。通常は変更する必要はありません。	0 (ポート No.123)
サーバ指定方法 *1	時計情報を取得する SNTP サーバを、どの方法で指定するかを選択します。 • IP アドレス • ホスト名	IP アドレス
IP アドレス *3	SNTP サーバの IP アドレスを設定します。 (「サーバ設定方法」を、「IP アドレス」に指定した場合。)	なし
ホスト名	SNTP サーバのホスト名（ホストのドメイン名）を設定します。 (「サーバ設定方法」を、「ホスト名」に指定した場合。) (半角英数字、ドット、ハイフン：200 文字以内、 “.” と “.” の間は半角英数字 63 文字以内)	なし
指定時刻［時 : 分 : 秒］	SNTP サーバにアクセスして、時計を合わせる時刻を設定します。 (設定範囲：00:00:00 ～ 23:59:59)	00:00:00
タイムアウト時間［秒］*1	タイムアウトの検知時間を設定します。 (設定範囲：0 ～ 255 秒) リモートホストから応答がないときは、ここで設定した時間の間隔で、再送処理を 4 回まで行います。	0(10 秒)
時差調整	SNTP サーバから取得した時間データに時差を足したものを、CPU ユニットの時計情報に設定したい場合、設定します。 SNTP サーバから取得した時間情報をそのまま使用したい場合は、0 を入力します。	+0 : 0

*1 「SNTP サーバの時計情報」を「取得する」場合に、設定が必要です。
*2 ポート No. 53、68、161、162、2222、9600、44818 はシステムで使用するため設定できません。
*3 以下に示す IP アドレスは Ethernet の制約上、指定できません。
- 127（10 進数）または、0 または、255 で始まる IP アドレス
- クラス D の IP アドレス（224.0.0.0 ～ 239.255.255.255）
- クラス E の IP アドレス（240.0.0.0 ～ 255.255.255.255）

参考

SNTP サーバから時計情報の取得についての詳細は、「第 10 章 時計情報自動調整機能」を参照してください。

5-6　[SNMP 設定] ダイアログ



● SNMP サービス

項目名	内容	デフォルト値
SNMP サービス	SNMP（監視機能）サービスを使用するか、しないかを選択します。 「使用しない」を選択した場合は、外部からの SNMP マネージャによる接続はできません。	使用しない
ポート No.*	SNMP マネージャからの接続に使用する SNMP サーバとの接続に使うポート No. を設定します。 通常、変更する必要はありません。	0 （ポート No.161）
連絡先	通信機器の管理者名、および設置場所を文字情報で入力しておきます。必ずしもいずれの項目を入力する必要はありません。設定した内容は SNMP マネージャから読み出されます。 （それぞれ半角英数字 200 文字以内）	なし
設置場所		なし
認証トラップを送信する	認証トラップを送信するか、しないかを選択します。 「認証トラップを送信する」に設定した場合、受付認証 1、受付認証 2 に設定していない SNMP マネージャからアクセスがあった場合、SNMP マネージャに認証トラップを送信します。 「認証トラップを送信する」に設定した場合は、「SNMP トラップ」タブで、SNMP トラップの設定を行ってください。	チェックなし （使用しない）

* ポート No.53、68、123、162、2222、9600、44818 はシステムで使用するため設定できません。

参考

SNMP サービスについての詳細は、🕮「第 11 章 SNMP（エージェント）機能」を参照してください。

● 受付認証 1[*1]

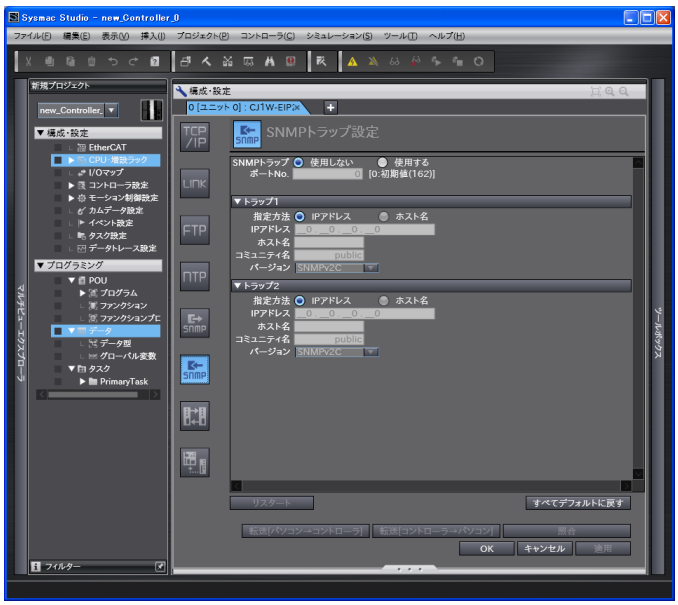
項目名	内容	デフォルト値
指定方法	アクセスを許可する SNMP マネージャの指定方法を選択します。 • IP アドレス • ホスト名 特定の SNMP マネージャのみアクセスを許可する場合は、この設定が必要です。 IP アドレスまたはホスト名が設定されていないと、受付認証が行われません。	IP アドレス
IP アドレス[*2]	SNMP マネージャの IP アドレスを設定します。 デフォルト（0.0.0.0）にした場合、すべての SNMP マネージャからのアクセスを確認します。 （受付認証 1 の「指定方法」を「IP アドレス」に指定した場合。）	0.0.0.0
ホスト名	SNMP マネージャのホスト名を設定します。 （受付認証 1 の「指定方法」を「ホスト名」に指定した場合。） （半角英数字、ドット、ハイフン 200 文字以内、 “.” と “.” の間は半角英数字 63 文字以内）	なし
コミュニティ名	SNMP マネージャが、EtherNet/IP ユニットから情報を取得する時に必要なコミュニティ名の設定を行います。 （半角英数字、ドット、ハイフン：200 文字以内）	public

● 受付認証 2[*1]

項目名	内容	デフォルト値
指定方法	アクセスを許可する SNMP マネージャの指定方法を選択します。 • IP アドレス • ホスト名 特定の SNMP マネージャのみアクセスを許可する場合は、この設定が必要です。 IP アドレスまたはホスト名が設定されていないと、受付認証が行われません。	IP アドレス
IP アドレス[*2]	SNMP マネージャの IP アドレスを設定します。 デフォルト（0.0.0.0）にした場合、すべての SNMP マネージャからのアクセスを確認します。 （受付認証 2 の「指定方法」を「IP アドレス」に指定した場合。）	0.0.0.0
ホスト名	SNMP マネージャのホスト名を設定します。 （受付認証 2 の「指定方法」を「ホスト名」に指定した場合。） （半角英数字、ドット、ハイフン：200 文字以内、 “.” と “.” の間は半角英数字 63 文字以内）	なし
コミュニティ名	SNMP マネージャが、EtherNet/IP ユニットから情報を取得する時に必要なコミュニティ名の設定を行います。 （半角英数字、ドット、ハイフン：200 文字以内）	public

*1 「SNMP サービス」を「使用する」場合に設定が必要です。
*2 以下に示す IP アドレスは Ethernet の制約上、指定できません。
- 127（10 進数）または、0 または、255 で始まる IP アドレス
- クラス D の IP アドレス（224.0.0.0 ～ 239.255.255.255）
- クラス E の IP アドレス（240.0.0.0 ～ 255.255.255.255）

5-7　[SNMP トラップ設定] ダイアログ



● SNMP トラップ

項目名	内容	デフォルト値
SNMP トラップ	SNMP トラップ（ネットワーク異常検出）機能を使用するか、しないかを選択します。 「使用しない」を選択した場合は、SNMP マネージャへの SNMP トラップは送信されません。	使用しない
ポート No.*	SNMP サーバとの接続に使うポート番号を選択します。 通常、変更する必要はありません。	0 (ポート No.162)

* ポート No.53、68、123、161、2222、9600、44818 はシステムで使用するため設定できません。

参考

SNMP トラップについての詳細は、📖「11-1-1 SNMP（エージェント）機能とは」を参照してください。

● トラップ 1[*1]

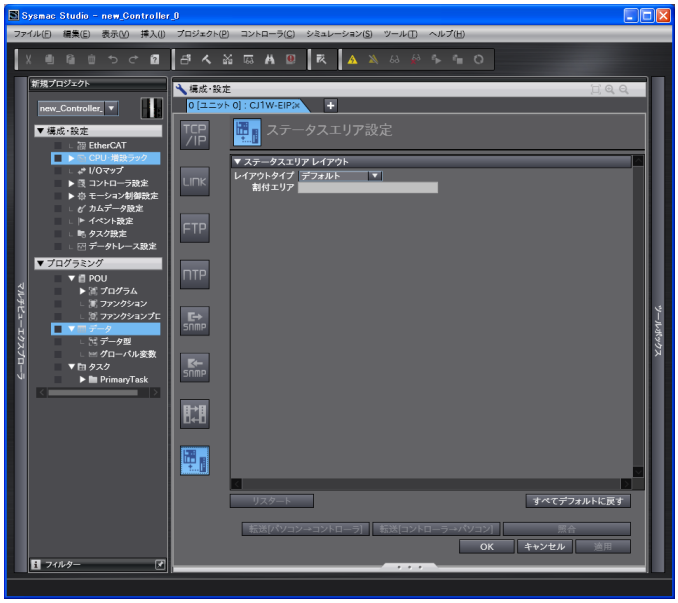
項目名	内容	デフォルト値
指定方法	SNMP トラップの送信先とする SNMP マネージャを、どの方法で指定するかを選択します。 • IP アドレス • ホスト名	IP アドレス
IP アドレス[*2]	SNMP マネージャの IP アドレスを設定します。 （トラップ 1 の「指定方法」を「IP アドレス」に指定した場合。）	0.0.0.0
ホスト名	SNMP マネージャのホスト名を設定します。 （トラップ 1 の「指定方法」を「ホスト名」に指定した場合。） （半角英数字、ドット、ハイフン：200 文字以内、 “.” と “.” の間は半角英数文字 63 文字以内）	なし
コミュニティ名	コミュニティ名を設定します。 （半角英数字 200 文字以内）	public
バージョン	SNMP マネージャが対応しているバージョンを選択します。 • SNMPv1 • SNMPv2C	SNMPv2C

● トラップ 2[*1]

項目名	内容	デフォルト値
指定方法	SNMP トラップの送信先とする SNMP マネージャを、どの方法で指定するかを選択します。 • IP アドレス • ホスト名	IP アドレス
IP アドレス[*2]	SNMP マネージャの IP アドレスを設定します。 （トラップ 2 の「指定方法」を「IP アドレス」に指定した場合。）	0.0.0.0
ホスト名	SNMP マネージャのホスト名を設定します。 （トラップ 2 の「指定方法」を「ホスト名」に指定した場合。） （半角英数字、ドット、ハイフン：200 文字以内、 “.” と “.” の間は半角英数文字 63 文字以内）	なし
コミュニティ名	コミュニティ名を設定します。 （半角英数字 200 文字以内）	public
バージョン	SNMP マネージャが対応しているバージョンを選択します。 • SNMPv1 • SNMPv2C	SNMPv2C

*1 「SNMP トラップ」を「使用する」場合に設定が必要です。
*2 以下に示す IP アドレスは Ethernet の制約上、指定できません。
- 127（10 進数）または、0 または、255 で始まる IP アドレス
- クラス D の IP アドレス（224.0.0.0 ～ 239.255.255.255）
- クラス E の IP アドレス（240.0.0.0 ～ 255.255.255.255）

5-8　［ステータスエリア設定］ダイアログ



● ステータスエリアレイアウト

項目名	内容	デフォルト値
レイアウトタイプ	本ユニットがオリジネータのコネクションにおいて、該当ターゲットノードのステータス情報を、格納する変数を選択します。 • デフォルト 以下の CJ 用デバイス変数に格納します。 •「*_TargetPLCMdSta」（ターゲットノード PLC 本体運転情報） •「*_TargetPLCErrSta」（ターゲットノード PLC 本体異常情報） •「*_RegTargetSta」（登録ターゲットノード） •「*_EstbTargetSta」（正常ターゲットノード） • ユーザ定義 任意のユーザ定義変数に格納することができます。	デフォルト
割付エリア	レイアウトタイプを「ユーザ設定」にした場合、ターゲットノードのステータス情報を格納するユーザ定義変数を入力します。 ユーザ定義変数の設定方法については、「3-3-2 ユーザ定義設定の設定方法」を参照してください。	なし

6

通信テスト

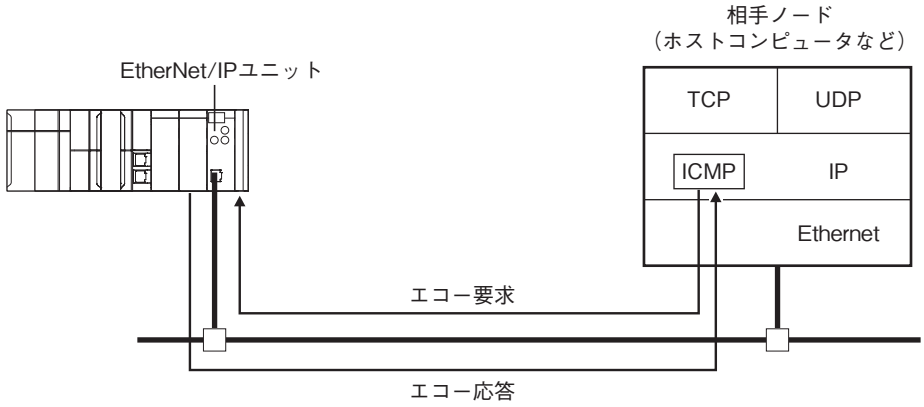
6-1 通信テスト 6-2
6-1-1 PING コマンドとは 6-2
6-1-2 EtherNet/IP ユニットでの使用方法 6-2
6-1-3 ホストコンピュータでの使用方法 6-3

6-1　通信テスト

EtherNet/IP ユニットに、基本的な設定（特に IP アドレスとサブネットマスク）を正しく設定できれば、EtherNet/IP 上のノードと通信することが可能となります。
ここでは、パソコンの PING コマンドを使用して、EtherNet/IP ユニットとの間での通信テスト方法を説明します。

6-1-1　PING コマンドとは

相手ノードに対して、エコーバックを要求するパケットを送出し、その応答パケットを受信することで、相手ノードと正常に通信できることを確認するための機能です。PING コマンドは、ICMP のエコー要求／応答機能を使用しています。エコー応答パケットは、ICMP のプロトコルで自動的に返信されます。通常、PING コマンドは、ネットワーク構築時などに相手ノードとの接続を確認するために使用します。EtherNet/IP ユニットは、ICMP の応答の機能を持っています。
PING コマンドにより、相手ノードから正常に返信を受けることができれば、物理的な接続や、Ethernet としてのノードの設定が正常に行われていることになります。



6-1-2　EtherNet/IP ユニットでの使用方法

EtherNet/IP ユニットは、他ノード（ホストコンピュータなど）から送信されてきた PING コマンドによるエコー要求に対して自動的にエコー応答を返信します。

6-1-3　ホストコンピュータでの使用方法

ホストコンピュータから EtherNet/IP ユニットに対して PING コマンドを実行することができます。
ホストコンピュータでの PING コマンドの使用方法は次のとおりです。

使用方法

ホストコンピュータのプロンプト（$）が表示されている状態で下記のように入力します。

$ ping　IP アドレス（ホスト名）

送信先は、IP アドレスまたはホスト名で指定します。

参考

PING コマンドは、ホストコンピュータの機種によってはサポートしていないことがありますのでご注意ください。

使用例

この例では、IP アドレス 130.25.36.8 のノードへ PING コマンドを送信します。
画面例の "$" は、ホストコンピュータのプロンプトを表します。

● 正常実行時の画面例

```
$ ping   130.25.36.8                                          ←PINGコマンドを起動します。
PING   130.25.36.8: 56   data   bytes
64   bytes   from   130.25.36.8:   icmp_seq=0.   time=0. ms
64   bytes   from   130.25.36.8:   icmp_seq=0.   time=0. ms
             :            :            :            :            :
64   bytes   from   130.25.36.8:   icmp_seq=0.   time=0. ms
                                                              ←中断するために[CTRL]＋Cキーを押します。
---- 130.25.36.8   PING   Statistics ----
9 packets   transmitted,   9 packets   received,   0% packets   loss
round-trip   (ms)       min/avg/max   = 0/1/16
$
```

● 異常発生時の画面例

```
$ ping   130.25.36.8                                          ←PINGコマンドを起動します。
PING   130.25.36.8: 56   data   bytes
                                                              ←中断するために[CTRL]＋Cキーを押します。
---- 130.25.36.8   PING   Statistics ----
9 packets   transmitted,   0 packets   received,   100% packets   loss
$
```

ホストコンピュータでの PING コマンド使用の詳細については、使用するホストコンピュータの OS のコマンドリファレンスを参照してください。

7

タグデータリンク機能

7-1 タグデータリンクの概要 7-2
7-1-1 タグデータリンクとは 7-2
7-1-2 タグデータリンクのデータエリア 7-4
7-1-3 タグデータリンクの機能・性能一覧 7-8
7-1-4 使用方法の概要 7-9
7-1-5 タグデータリンクの開始／停止 7-12
7-1-6 コントローラステータスについて 7-13
7-1-7 タグデータリンクするデータの同期性について 7-15

7-2 タグデータリンクの設定 7-19
7-2-1 Network Configurator の起動 7-19
7-2-2 タグデータリンク設定手順 7-21
7-2-3 デバイスの登録 7-22
7-2-4 タグ／タグセットの作成 7-24
7-2-5 コネクション設定 7-38
7-2-6 ウィザードによるコネクションの作成 7-48
7-2-7 デバイスのドラッグ＆ドロップによるコネクションの作成 7-51
7-2-8 Network Configurator のネットワーク接続手順 7-54
7-2-9 タグデータリンクパラメータのダウンロード 7-58
7-2-10 タグデータリンクパラメータのアップロード 7-61
7-2-11 タグデータリンクパラメータの照合 7-64
7-2-12 タグデータリンクの開始／停止 7-68
7-2-13 デバイスパラメータのクリア 7-70
7-2-14 ネットワーク構成ファイルの保存 7-71
7-2-15 ネットワーク構成ファイルの読出 7-73
7-2-16 コネクションのチェック 7-75
7-2-17 デバイスの変更 7-76
7-2-18 デバイスステータス表示 7-79

7-3 タグデータリンクのラダープログラム 7-80
7-3-1 タグデータリンクでのラダープログラム 7-80
7-3-2 タグデータリンクに関連するステータスについて 7-84

7-4 NJ CPU 以外の機種とタグデータリンクを行う場合について 7-85

7-1　タグデータリンクの概要

7-1-1　タグデータリンクとは

タグデータリンクとは、EtherNet/IP ネットワーク上で、コントローラ－コントローラ間、またはコントローラー他のデバイス間で、互いのタグをサイクリックにデータ交換する機能です。タグには変数を割り当てます。（タグに I/O メモリアドレスを割り当てることもできます。）

タグデータリンクの設定は Network Configurator で行います。設定方法の詳細については、🕮「7-2 タグデータリンクの設定」を参照してください。

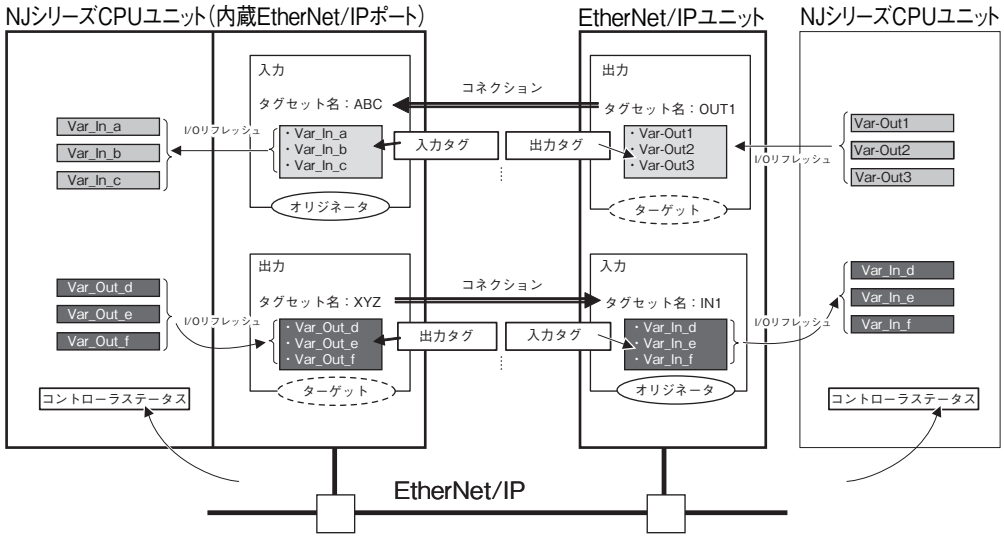
使用上の注意

［CIP メッセージサーバ］を［使用しない］に設定している場合は、タグデータリンク機能は使用できません。［CIP メッセージサーバ］を［使用する］に設定してください。CIP メッセージサーバの設定方法は、🕮「12-2 ポートのオープン／クローズ機能」の「● CIP メッセージサーバ」を参照してください。

参考

タグデータリンクの設定は Sysmac Studio で行うこともできます。
Sysmac Studio での設定方法の詳細については、🕮「A-2 Sysmac Studio でタグデータリンク（EtherNet/IP コネクション）を設定する」を参照してください。

タグデータリンクでは、一方のノードが、「コネクション」と呼ぶ通信回線の開設を要求（オープン）し、相手のノードとデータ交換します。
コネクションの開設を要求する側を「オリジネータ」、要求される側を「ターゲット」と呼びます。



コントローラ－コントローラ間通信の場合、コネクション情報は「オリジネータ」となる入力（受信）側コントローラの EtherNet/IP ユニットに設定します。



参考

コントローラー I/O 機器通信の場合、コネクション情報は「オリジネータ」となる EtherNet/IP ユニットに設定します。I/O 機器を使用する場合、コネクション情報が記述されている EDS ファイルを Network Configurator にインストールする必要があります。インストール方法は「A-3 EDS ファイルの管理」を参照してください。

コネクション情報にはデータ交換する各ノードの出力エリアと入力エリアを指定する必要があります。これらのエリアはそれぞれ出力タグセット、入力タグセットと呼び、タグセットには少なくとも 1 つ以上のタグを指定する必要があります。
データ交換するデータのサイズは、タグセットに含まれる各タグのサイズの総和となり、出力タグセットのサイズと入力タグセットのサイズは一致している必要があります。

7-1-2　タグデータリンクのデータエリア

タグとは

「タグ」とは、タグデータリンクでデータ交換をする対象の単位です。
「タグ」に指定した自身のネットワーク変数と、相手のネットワーク変数、または指定した I/O メモリエリア間でデータが交換されます。



使用上の注意

「タグ」に割り付けるネットワーク変数には、変数の値の同期性を確保するため、「更新タスク」を設定する必要があります。
詳細については、🕮「7-1-7 タグデータリンクするデータの同期性について」を参照ください。

タグセットとは

タグデータリンクでは、コネクションを張るときに、1 つまたは複数のタグ（コントローラステータスを含め、最大 8 個）をまとめて、タグのセットを構成します。これを「タグセット」と呼びます。
つまり、「タグセット」とは、タグデータリンクでコネクションを張る、データの単位となります。
タグデータリンクは、タグセットとタグセット間をコネクションで結びつけることで行います。タグセットには必ずタグセット名を設定します。
タグセットに含まれる各タグは、登録された順番にデータ交換されるため、入力タグセットと出力タグセットに含まれるタグの順番をあわせてください。
（注）コネクションとは、データの同時性を確保する、データ交換の単位です。
したがって、「タグセット」は、1 個または複数のタグのデータ交換時の、通信上のデータの同時性を確保する単位となります。

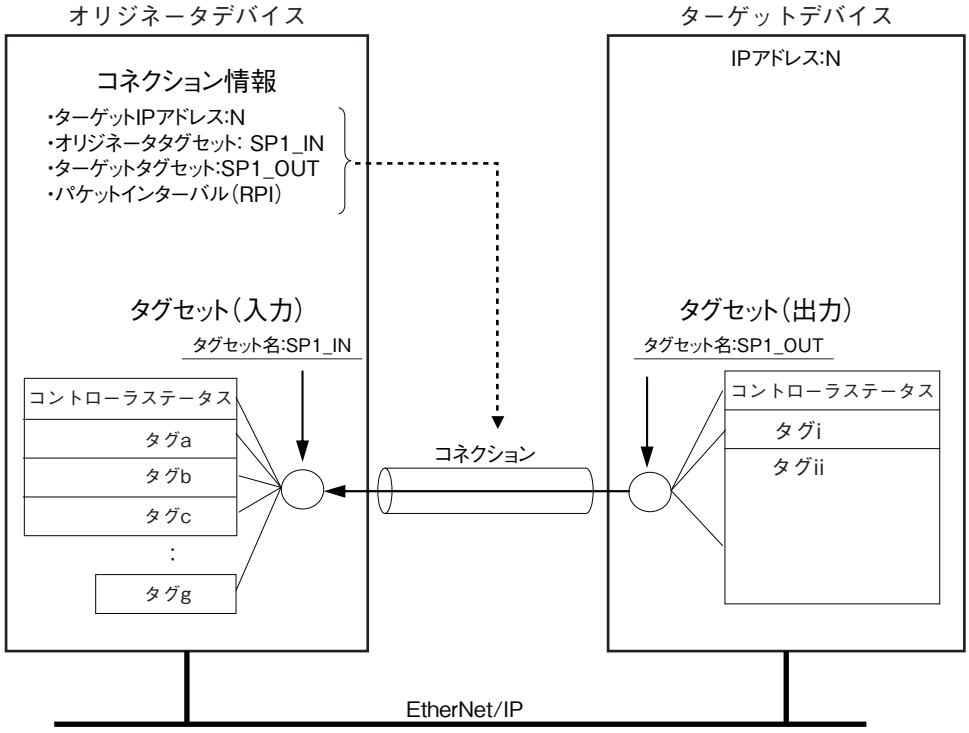


使用上の注意

タグセットに含まれる各タグは、登録された順番にデータ交換されるため、入力タグセットと出力タグセットに含まれるタグの登録順序をあわせてください。

● **例**

以下の例では、オリジネータのタグ a、b、c、…、g（ともに入力）を「SP1_IN」という名前でタグセットとし、ターゲットのタグ i、ii（ともに出力）を「SP1_OUT」という名前でタグセットにして、コネクションを設定しています。



タグセットには、入力（受信）タグセット、出力（送信）タグセットがあり、1 タグセット内に入力と出力のタグを混在させることはできません。また、1 個の入力のタグを複数の入力（受信）タグセットに重複して登録することはできません。

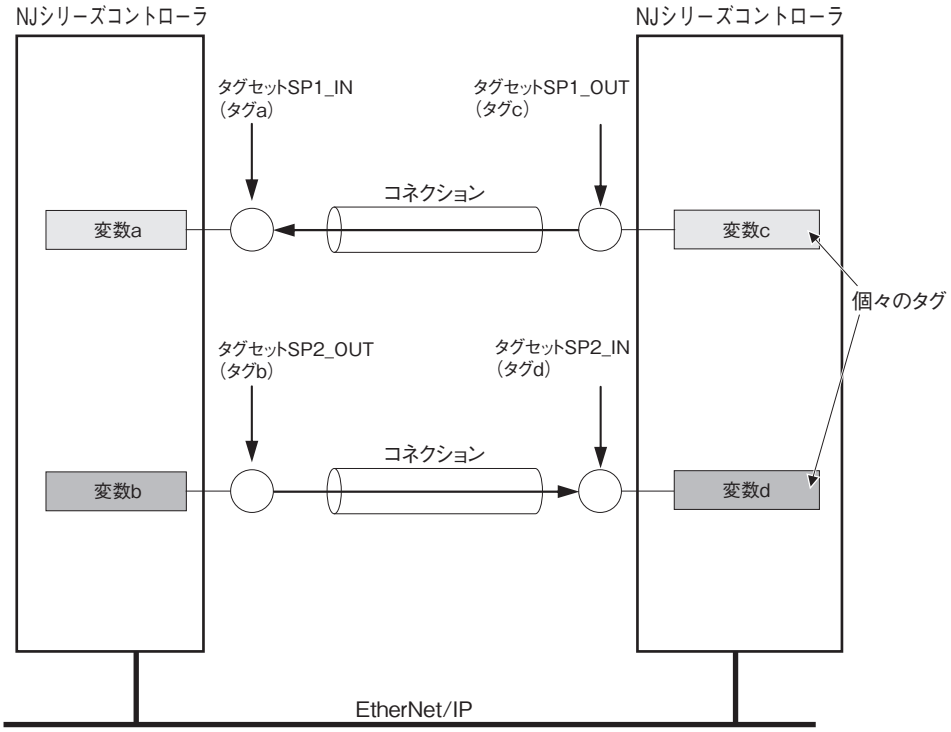
● タグセットには 1 つか複数のタグを使用

1 つのコネクションの入力／出力タグセットには、1 つのタグまたは複数のタグのタグセットを任意に設定できます。

例えば、入力タグセットには 1 つのタグのタグセットを、出力タグセットには複数のタグのタグセットを、それぞれ設定できます。

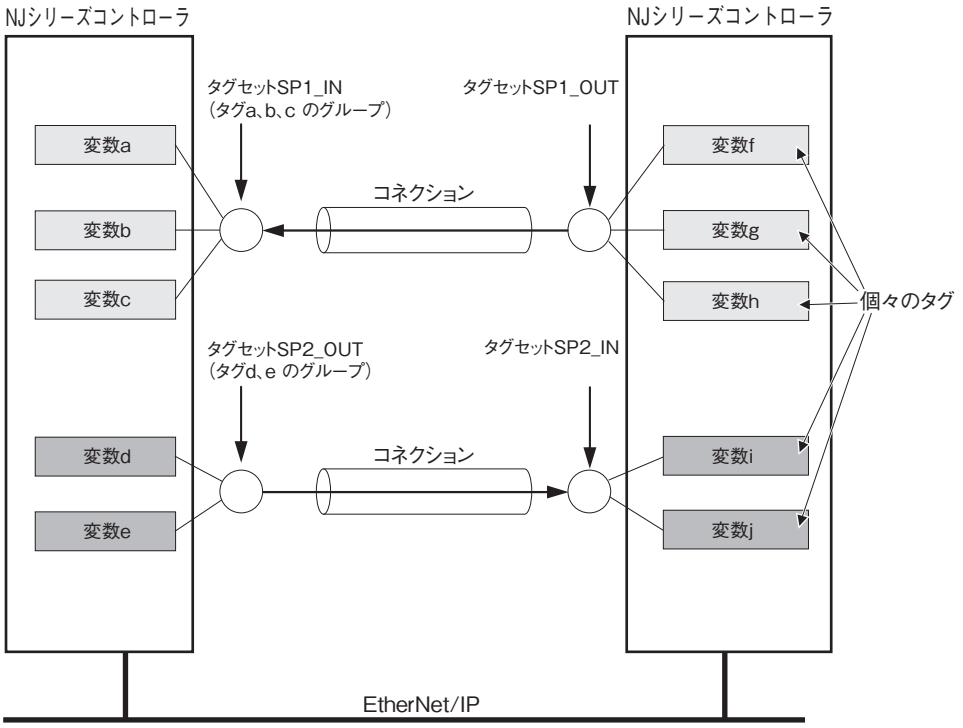
・1 個のタグが 1 個のタグセットの場合

Network Configurator の操作上では、タグセットは基本的に 1 個のタグから成ります。



・複数個のタグが 1 個のタグセットの場合

以下のように、複数のタグをグループ化することもできます。1 個のタグセットには、最大 8 個のタグ（合計最大 722CH）をまとめることができます。



（注）タグセットはコネクションの単位のため、入力のみ、または出力のみから構成します（入力、出力を混在させることはできません）。

7-1-3　タグデータリンクの機能・性能一覧

NJ CPU のタグデータリンク機能と性能仕様を次に示します。

項目		内容
通信種類		EtherNet/IP 標準仕様 Implicit 通信（コネクション型のサイクリック通信）
設定方法		Network Configurator でタグ／タグセットおよびコネクションの設定をしたあと、タグデータリンクパラメータを EtherNet/IP ネットワーク上の全デバイスにダウンロードします。 Sysmac Studio で作成したネットワーク変数は CSV ファイルにエクスポートできます。 これを Network Configurator でインポートすれば、ネットワーク変数をタグに割り付けることができます。 ダウンロード後、EtherNet/IP ユニットがリスタートされタグデータリンクが開始されます。
タグ*1	使用できる変数の種類	次のネットワーク変数をタグとして指定できます。*2、*3 • グローバル変数
	1 タグあたり最大 CH 数	722CH（1444 バイト）
	最大タグ数	256 個
タグセット	1 タグセットあたり最大タグ数	8 個（コントローラステータスを含める場合は 7 個）
	1 タグセットあたり最大 CH 数	722CH（1444 バイト）
	最大タグセット数	256 個
コネクション		1 ユニットあたり最大コネクション数：256 本
コネクションタイプ		コネクションごとに、1:1（ユニキャスト）または 1:N（マルチキャスト）の通信を選択することが可能。
パケットインターバル（RPI）		0.5 ～ 10000ms（0.5ms 単位） コネクションごとに設定可能。

*1. 特定の I/O メモリアドレスをタグに指定したい場合は、Sysmac Studio にて I/O メモリアドレスを AT 指定した変数を作成し、その AT 指定した変数をタグとして指定してください。
*2. Sysmac Studio で作成したネットワーク変数はタグとして Network Configurator にインポートできます。
ただし、Sysmac Studio でネットワーク属性にした変数のうち、「0000」や「H0000」など、I/O メモリアドレスの表記の変数は、CSV ファイルにエクスポートされません。
*3. タグとして指定できる変数の単位は以下のとおりです。

種類		例	指定可否
基本データ型変数		**aaa**	可
列挙型変数		**bbb**	可
配列指定の変数	**配列全体**	**ccc**	可
	要素	**ccc[2]**	可
構造体型変数	**構造体全体**	**ddd**	可
	メンバ	**ddd.xxx**	可
共用体型変数	**共用体全体**	**eee**	否
	メンバ	**eee.yyy**	可

7-1-4　使用方法の概要

本書では、設定されたコネクション情報をタグデータリンクパラメータと呼びます。
以下では Sysmac Studio と、Network Configurator を使用したタグデータリンクの設定方法を解説します。

ネットワーク変数の設定（Sysmac Studio）

Sysmac Studio にて、タグデータリンクさせる変数を、あらかじめネットワーク変数として登録します。

1　グローバル変数テーブルにて、タグデータリンクしたい（タグとして使用したい）変数の「ネットワーク公開属性」を「入力」または「出力」に設定にして、ネットワーク公開の属性にします。

2　タグセット内のタグ間のデータの同時性を確保したい場合は、同一タグセット内のタグ（「ネットワーク公開属性」ありの変数）を以下のように設定します。

タグデータリンクするデータについて、以下のような同期性を確保するため、ネットワーク公開の属性にした変数に「更新タスク」を設定します。

（注）タグに AT 指定した変数を指定する場合は、「更新タスク」の設定は不要です。
プライマリ定周期タスクで更新されます。

タグデータリンクするデータとの同期性についての詳細は、📖「7-1-7 タグデータリンクするデータの同期性について」を参照ください。

- タグセット内のタグ間データの同時性を確保する
- タグに割り付けたネットワーク変数の更新タイミングと、そのネットワーク変数にアクセスするユーザプログラムの実行周期を同期させる

使用上の注意

タグデータリンクさせる変数の変数名には、以下のように I/O メモリアドレスを表す表記を使用することはできません。

1) 半角数字のみからなる変数名（例 :001）
2) 以下の半角英字（大文字／小文字とも）と、半角数字からなる変数名
   - H（例：H30）
   - W（例：W30）
   - D（例：D100）
   - E0_ ～　E18_

タグデータリンクパラメータの設定／ダウンロード（Network Configurator）

Network Configurator で、以下に示すタグデータリンクパラメータ（コネクション情報）を作成し、EtherNet/IP ネットワーク上のすべてのオリジネータデバイスにダウンロードします。
EtherNet/IP ユニットでタグデータリンク機能を使用する場合、Network Configurator で以下の設定を行います。

1　構成情報の作成

タグデータリンクのコネクションを作成する EtherNet/IP ポート、ユニットを登録します。
詳細は、📖「7-2-3 デバイスの登録」を参照してください。

2　タグの設定

CPU ユニットの変数を、入力（受信）タグ、出力（送信）タグとして作成します。
EtherNet/IP ユニットでは最大 256 個のタグが作成できます。（1 タグ当たりのデータサイズは最大 1444 バイト (722CH)）
Sysmac Studio で作成したネットワーク変数は、CSV ファイルへのエクスポート／インポート機能を使うことにより、タグとして Network Configurator に登録することができます。
出力（送信）タグでは、CPU ユニットの運転停止異常発生時に出力データを 0 クリアするか、異常発生直前のデータを出力しつづけるかの定義ができます。

3　タグセットの設定

出力用・入力用それぞれにタグセットを作成し、タグを配置します（1 タグセット当たり最大 8 個）。
EtherNet/IP ユニットでは最大 256 個のタグセットが作成できます。（1 タグセット当たりのデータサイズは最大 1444 バイト (722CH)）
タグセットには、CPU ユニットの動作状態（運転情報、異常情報）を示すコントローラステータスを指定することが可能です。

4　コネクションの設定

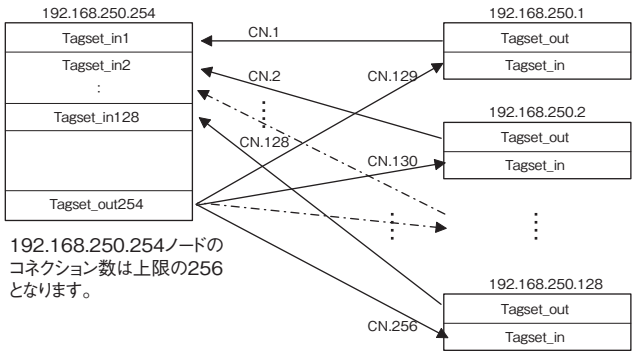
ターゲットデバイスの出力タグセットと、オリジネータデバイスの入力タグセットを、コネクションとして関連付けます。
EtherNet/IP ユニットでは、最大 32 本のコネクションが開設可能です。

参考

コネクション数の数え方

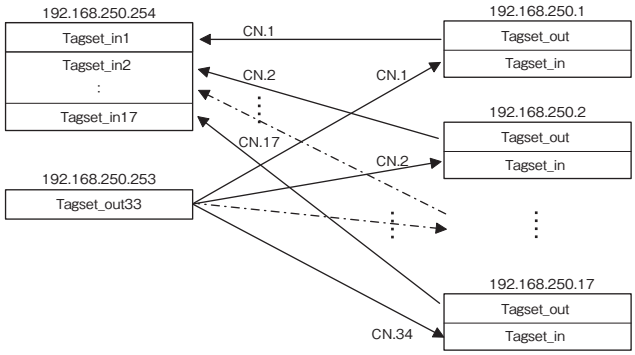
コネクション数は、データを受信している入力タグセットの数と、出力タグセットを送信しているノードの数の総和となります（下図参照）。
コネクションタイプとしてマルチキャスト（Multi-cast connection）、ユニキャスト（Point to Point connection）のいずれを指定した場合も、1 つのコネクション設定によりコネクションの消費数は、1 本となります。

コネクション数の計算例:
・IPアドレス192.168.250.254のEtherNet/IPユニットが、
128台のノードと双方向のコネクションを開設した場合



コネクション数が上限を超える場合、コントローラに EtherNet/IP ユニットを増設する必要があります。（下図参照）

コネクション数の計算例:
・IPアドレス192.168.250.254の内蔵EtherNet/IPポートが、
17台のノードと双方向のコネクションを開設する場合、1ユニット当たり
のコネクション数の上限32を超えてしまいます。
この場合、同一コントローラに例えばIPアドレス192.168.250.253のEtherNet/IP
ユニットを増設し、出力タグセットを増設側に配置してコネクション作成することで
17台以上のノードとも双方向通信が可能です。



バージョン関連情報

- EtherNet/IP ユニット形 CJ1W-EIP21 を NJ シリーズコントローラに装着して使用するのは、CPU ユニットの Ver.1.01 以降、かつ Sysmac Studio Ver1.02 以降で可能です。
- EtherNet/IP ユニット形 CJ1W-EIP21S を NJ シリーズコントローラに装着して使用するのは、CPU ユニットの Ver.1.67 以降、かつ Sysmac Studio Ver.1.60 以降で可能です。

パケットインターバル (RPI) の設定

パケットインターバル RPI(Requested Packet Interval) とは、タグデータリンク時の Ethernet 回線上の入出力データ更新周期で、コネクションごとに設定します。
EtherNet/IP ユニットでは 0.5 ～ 10000ms（0.5ms 単位）の範囲で設定できます。デフォルトは 50ms です。
EtherNet/IP では、ノード数に依存せず、コネクションごとに設定したパケットインターバル（RPI）で回線上のデータ交換が行われます。

コネクションタイプの設定

タグデータリンクのコネクション設定では、コネクションタイプとしてマルチキャスト (Multi-cast connection) またはユニキャスト (Point to Point connection) を選択できます。
マルチキャストは、1 パケットで 1 つの出力タグセットを複数のノードに送信し入力タグセットに割り付けることが可能です。
一方、ユニキャストは、1 つの出力タグセットを、各ノードに対して個別に送信するため、入力タグセット数と同じ数のパケットを送信します。
このため、1 つの出力タグセットを複数のノードに送信する場合、マルチキャストを使用したほうが、通信負荷を減らすことができます。
ただし、マルチキャストを使用する場合、ネットワーク上のすべてのノードがこれを受信する場合を除き、マルチキャストフィルタ機能を持つスイッチングハブを使用してください。
マルチキャストフィルタ機能を持たないスイッチングハブを使用した場合、マルチキャストパケットがネットワーク全体にブロードキャスト（一斉同報）されるため、無駄なパケットがこれを必要としないノードにも送信され、そのノードの通信負荷を増加させる原因となります。
なお、マルチキャストを使用し 1 パケットで 1 つの出力タグセットを複数のノードに送信するには、これを受信する各コネクションのコネクションタイプがマルチキャスト (Multi-cast connection) で、コネクション I/O タイプ、パケットインターバル (RPI)、タイムアウト値のすべてが同一の場合に限ります。

使用上の注意

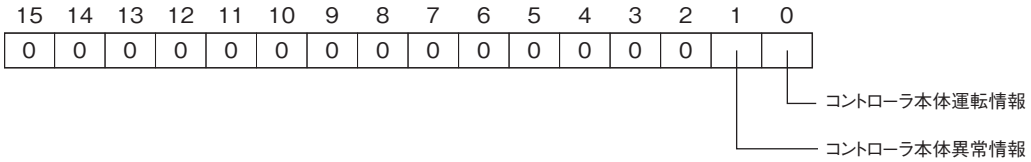
通信デバイスの性能は、製品仕様ごとに、ある程度の限界が定められています。このため、パケットインターバル (RPI) を無制限に設定することはできません。
詳細は、🕮「14-2 通信負荷の調整」を参照して、適正なパケットインターバル (RPI) を設定してください。

7-1-5　タグデータリンクの開始／停止

タグデータリンクは、Network Configurator でタグデータリンクパラメータをダウンロードし、NJ シリーズ CPU ユニットの電源が ON すると、自動的に起動します。
その後、タグデータリンクは、Network Configurator からの操作によって、ネットワーク全体またはデバイス単位（オリジネータ側のみ）に対して、開始／停止操作が可能です。
また、CJ ユニットデバイス変数でもネットワーク全体の開始／停止が可能です。詳細は、
🕮「7-2-12 タグデータリンクの開始／停止」を参照してください。

7-1-6　コントローラステータスについて

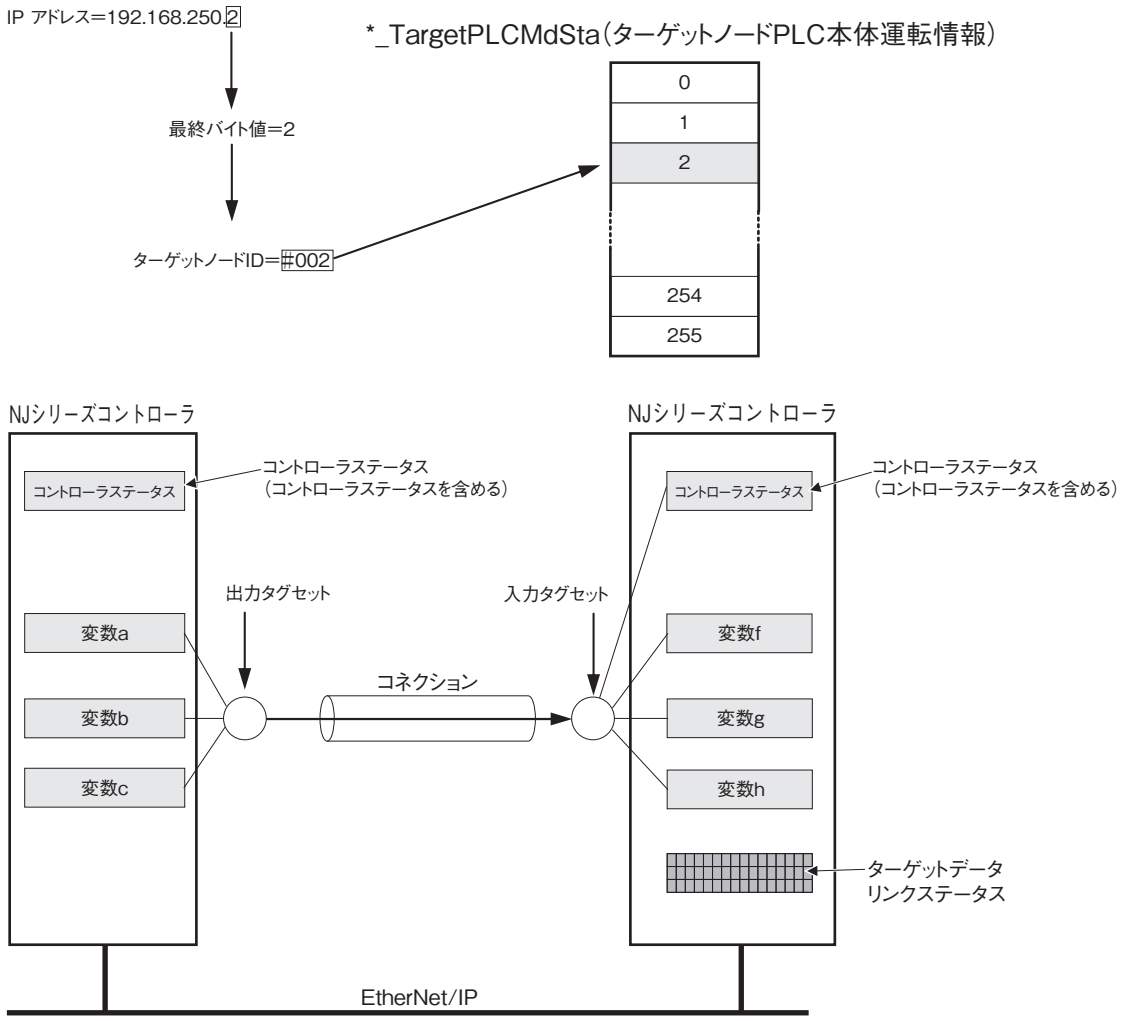
タグセットのメンバとして、コントローラステータスをタグデータリンクの送受信データに含めることができます。
コントローラステータスとは、EtherNet/IP ユニットが装着されている CPU ユニットの動作状態（運転情報、異常情報）を示すフラグです。
コントローラステータスを出力（送信）タグとして指定すると、コントローラステータスが、以下のフォーマットで、タグセットの先頭に加えられます。
（[タグセットの編集] ダイアログボックス右上の［コントローラステータス］を［含める］にします。）



コントローラステータスを受信するには、受信タグセットの編集ダイアログボックスでも、コントローラステータスを入力（受信）タグに指定します。
（[タグセットの編集] ダイアログボックス右上の［コントローラステータス］を［含める］にします。）
タグデータリンクを開始すると、次に示す CJ ユニットデバイス変数にコントローラステータスの内容が反映されます。

- 「*_TargetPLCMdSta」（ターゲットノード PLC 本体運転情報）
- 「*_TargetPLCErrSta」（ターゲットノード PLC 本体異常情報）

（例）ターゲットノードの IP アドレスが 192.168.250.2 のターゲットノード PLC 本体運転情報を送信する場合



参考

ターゲットノード ID は、ターゲットノードの IP アドレスにより重複することがあります。この場合は、Network Configurator でターゲットノード ID が重複しないように変更してください。

ターゲットノード ID の変更方法は、「7-2-5 コネクション設定」の、コネクション設定－登録デバイス一覧への登録－手順 4 を参照してください。

あるノードと複数のコネクションを使用して通信を行う場合に、すべてのコネクションでコントローラステータスを入力タグと出力タグとして指定すると、コントローラステータスの各情報が次に示す変数に反映されます。

コントローラステータスの情報	変数名称	動作内容
ターゲットノード PLC 本体運転情報	「*_TargetPLCMdSta」（ターゲットノード PLC 本体運転情報）	該当ターゲットノードのコントローラ本体の運転情報を表します。 （EtherNet/IP ユニットがオリジネータとなるコネクションの場合） 該当ターゲットノードとの全コネクションの情報がすべて運転状態になったとき、ターゲット側のターゲットノード ID に対応する配列が 1(TRUE) になります。 IP アドレスに対応するターゲットノード ID は、Network Configurator で変更することが可能です。 このステータスは、オリジネータとターゲットの双方で、通信データにコントローラステータスを含んでいる場合に有効となります。 この変数は随時更新されます。

参考

出力（送信）タグで「コントローラステータスを含める」とした場合でも、入力（受信）タグでも必ず「コントローラステータスを含める」と設定する必要はありません。
入力（受信）タグで「コントローラステータスを含める」と設定しない場合、「PLC ユニット運転情報」「ターゲットノード PLC ユニット異常情報」の変数にはコントローラステータスの内容が更新されませんが、入力（受信）タグには送信されています。
よって、入力（受信）タグで受け取ったコントローラステータスのデータを、受信データとして、使用することは可能です。

7-1-7　タグデータリンクするデータの同期性について

タグデータリンクでのデータの同期性を確保するため、タグを割り付けるネットワーク変数には、それぞれ更新タスクを設定する必要があります。

- タグセット内のタグ間データの同期性を確保する
- タグに割り付けたネットワーク変数の更新タイミングと、そのネットワーク変数にアクセスするユーザプログラムの実行周期を同期させる

参考

更新タスクとは、グローバル変数に対し、書き込み可能なタスクを 1 つだけ指定し、そのタスク以外からのグローバル変数に対する書き込みを禁止することで、参照する側のタスクからみたグローバル変数の値の同時性を確保する機能です。
更新タスクの仕組みについては、🕮『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）を参照してください。

● タグセット内のタグ間データの同期性を確保する

1 つのタグセット内にある複数タグの値について、同期性を確保させたい場合は、タグ（ネットワーク公開属性の変数）が、以下 4 つの条件を満たす必要があります。

(1) 同じタグセット（コネクション）に割り当てられていること

(2) タグに割り付けたネットワーク変数に、更新タスクを設定してあり、その更新タスクの種類が、タグセット内のタグ全て同じ種類であること。

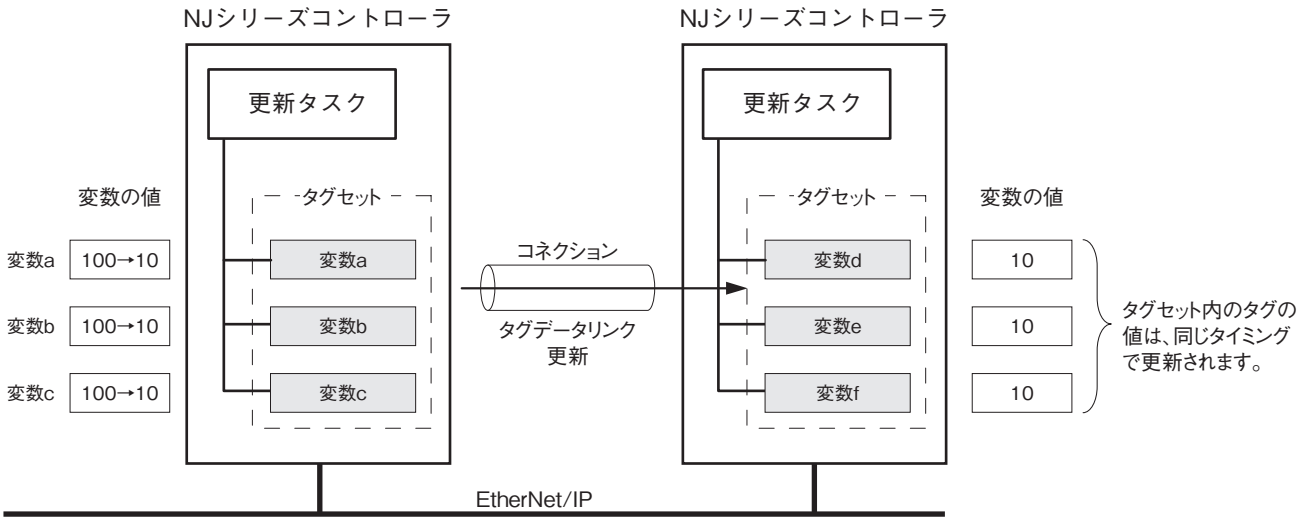
（注） ネットワーク変数に更新タスクを設定した場合、コントローラ外部からネットワーク変数へのアクセス時間を確保するための「変数アクセス時間」を設定する必要があります。

(3) AT（割付先）指定されているタグを、同一タグセット内に混在して設定しない

(4) タスクごとに設定する「変数アクセス時間」を、タグのデータ転送に必要な時間以上の時間で設定していること

変数アクセス時間とデータ転送に必要な時間については、🕮「14-3-3 タグデータリンクによる、タスク周期への影響」を参照してください。

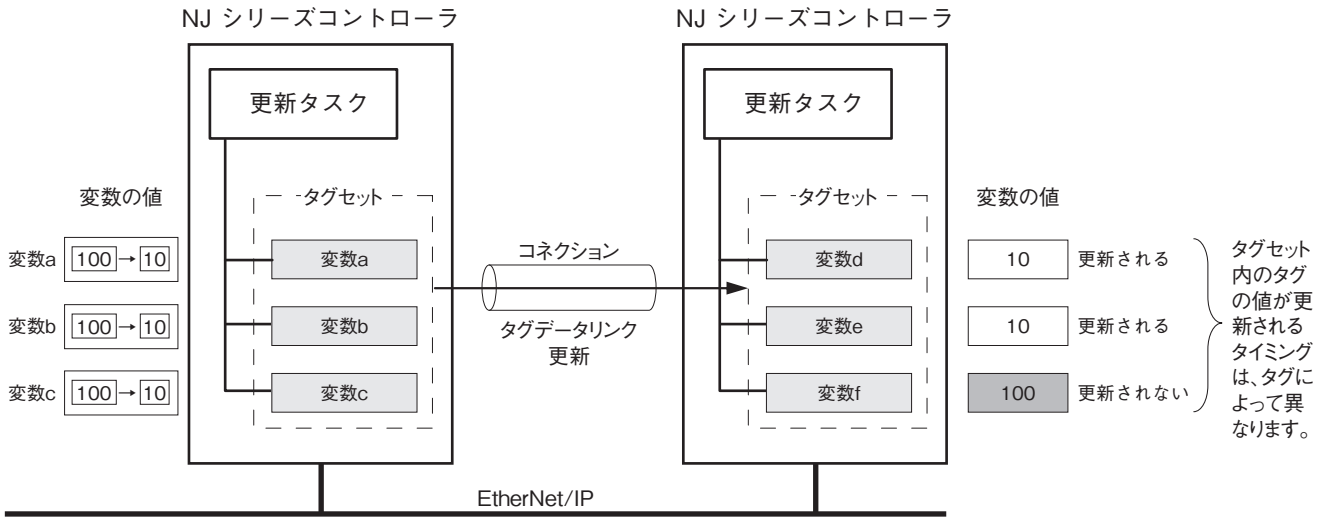
• タグ（ネットワーク変数）に更新タスクを設定した場合
　タグセット内のタグの同期性が確保されます。



参考

AT 指定している変数（タグ）は、プライマリ定周期タスクで更新されるので、更新タスクを設定する必要はありません。

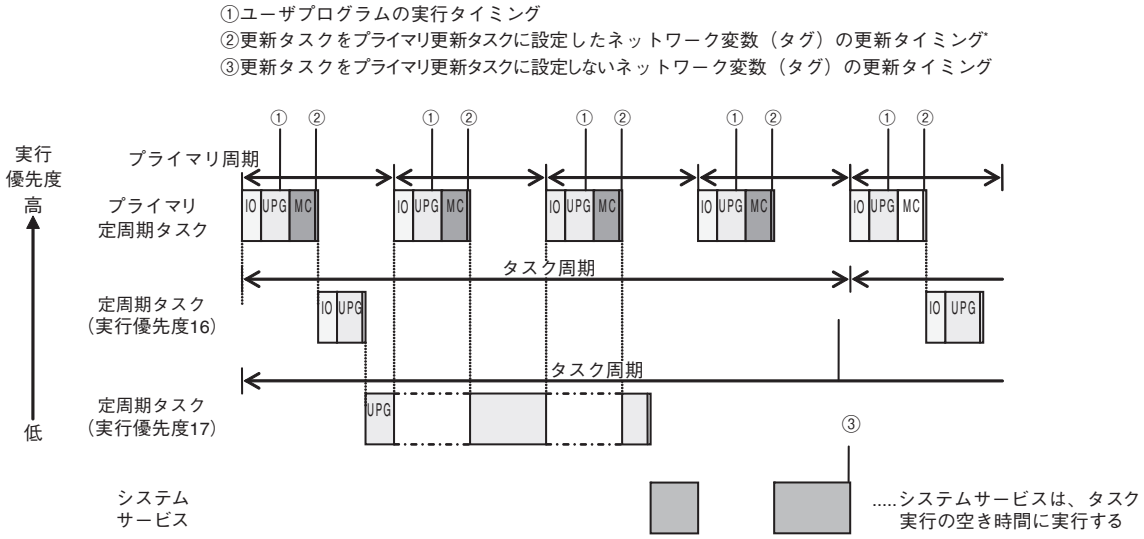
・タグ（ネットワーク変数）に更新タスクを設定しない場合
タグセット内のタグの同期性は確保されません。



● タグに割り付けたネットワーク変数の更新タイミングと、そのネットワーク変数にアクセスするユーザプログラムの実行周期を同期させる

タグに割り付けたネットワーク変数に、そのネットワーク変数にアクセスするユーザプログラムと同じ種類のタスクを、更新タスクとして設定します。

・更新タスクを設定したタグ（ネットワーク変数）と、更新タスクを設定しないタグ（ネットワーク変数）の動作の違い
ユーザプログラムと同じタスクを更新タスクに設定したタグ（ネットワーク変数）は、ユーザプログラムの実行タイミングに合わせて更新されます。
更新タスクを設定しないタグ（ネットワーク変数）が更新されるタイミングは、実行優先度の一番低いシステムサービスで処理されますので、ユーザプログラムの実行タイミングには同期しません。



*. タスク処理の「システム共通処理2」で更新処理されます。

参考

AT 指定のネットワーク変数にアクセスするユーザプログラムは、AT 指定したネットワーク変数の更新タイミングに合うように、プライマリ定周期タスクに設定してください。

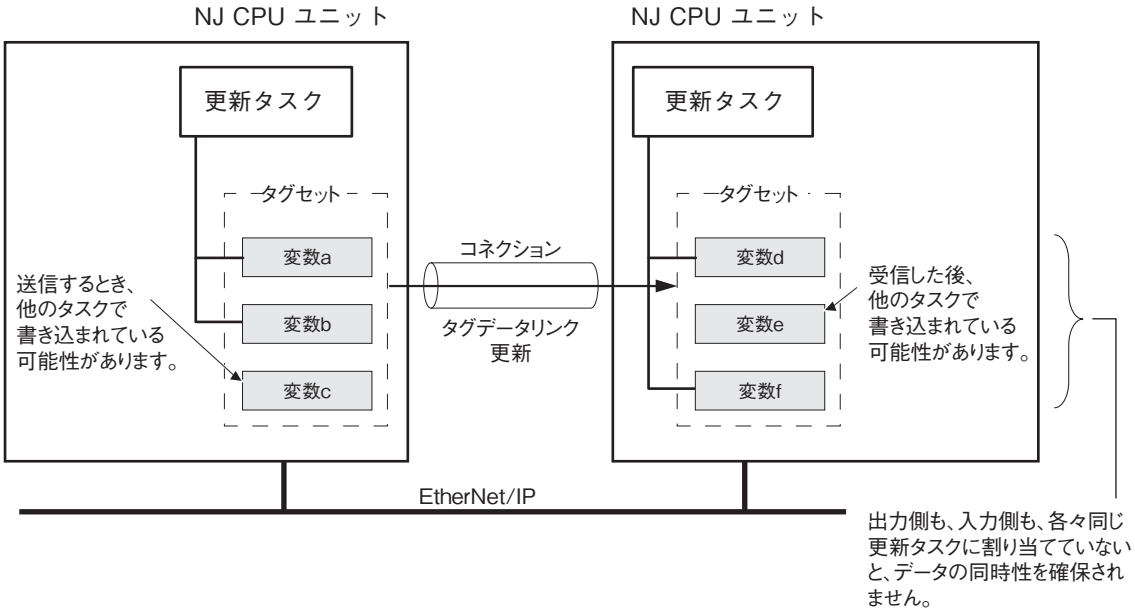
参考

更新タスク機能と、タグデータリンクでのデータの同期性確保の関係

タグデータリンクにおいて、更新タスクを指定していないグローバル変数の場合、

1. 出力側にてタグセット単位でデータを送信するとき、送信前に、タイミングによっては他のタスクによって値が書き込まれている可能性があります。
2. 入力側にてタグセット単位でデータを受信したあと、他のタスクによって値が書き込まれている可能性があります。

したがって、タグデータリンクでデータの同時性を確保したい場合、出力側の CPU ユニットにおいても、入力側の CPU ユニットにおいても、各々、同じ更新タスクを指定してください。



● 同期性を確保するために必要な処理時間について

データリンクのデータの同期性を確保するため、タグ（ネットワーク変数）に更新タスクを設定すると、更新タスクに指定したタスク自体の処理時間が増加します。タスクの処理時間が延びることにより、タグデータリンクのデータ更新が、コネクションごとに設定したパケットインターバル（RPI）の周期で更新されない可能性があります。

よって、パケットインターバル（RPI）の設定を、更新タスクに指定したタスクの周期に合わせて調整する必要が出てきます。

詳細は、🕮「14-3-3 タグデータリンクによる、タスク周期への影響」を参照してください。

● タスク設定の操作方法

(1) Sysmac Studio から、［タスク設定］－［変数のタスク間排他制御設定］にて、更新タスク指定をするグローバル変数、およびそれに対する更新タスク、参照タスクを設定します。

(2) 更新タスクごとに「変数アクセス時間」を設定します。

詳細は、🕮『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）を参照してください。

7-2　タグデータリンクの設定

参考

タグデータリンクの設定は Sysmac Studio で行うこともできます。
Sysmac Studio での設定方法の詳細については、「A-2 Sysmac Studio でタグデータリンク（EtherNet/IP コネクション）を設定する」を参照してください。

7-2-1　Network Configurator の起動

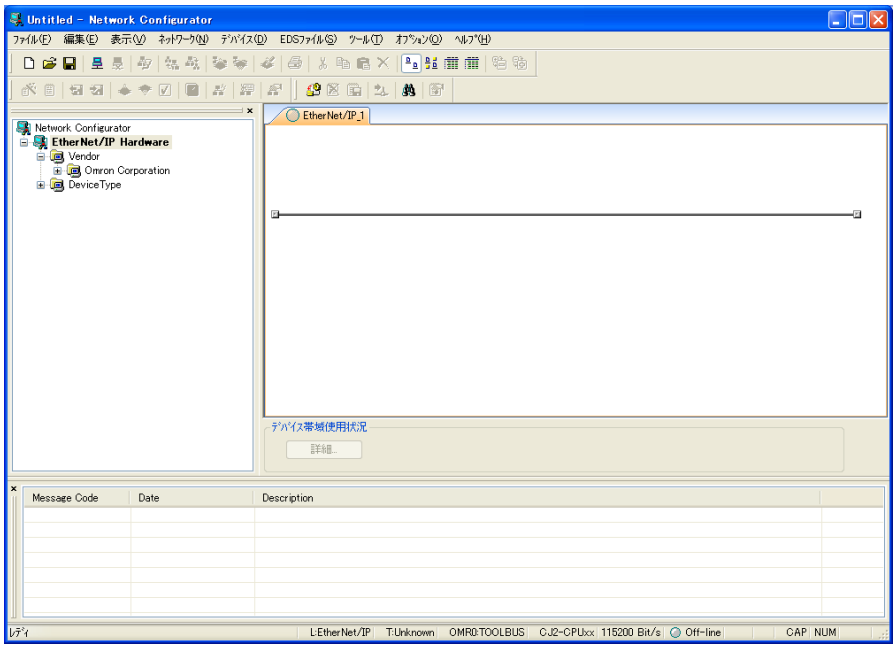
起動方法

タグデータリンクは Network Configurator を使って設定します。Network Configurator の起動方法は次のとおりです。

● Windows の［スタート］メニューから起動する

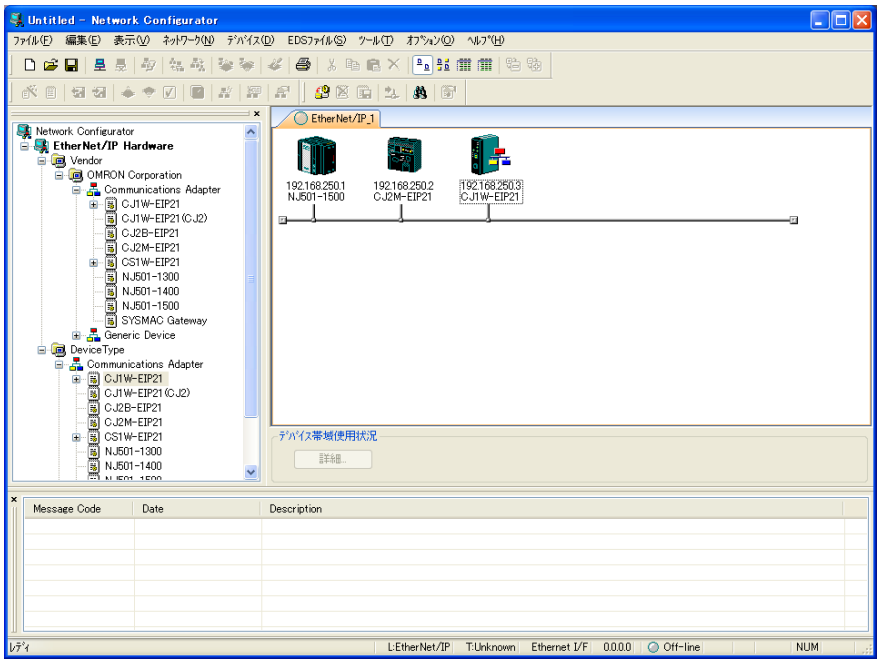
［OMRON］|［Sysmac Studio］|［Network Configurator for EtherNetIP］|［Network Configurator］を選択します。

起動すると、以下に示す初期画面が表示されます。

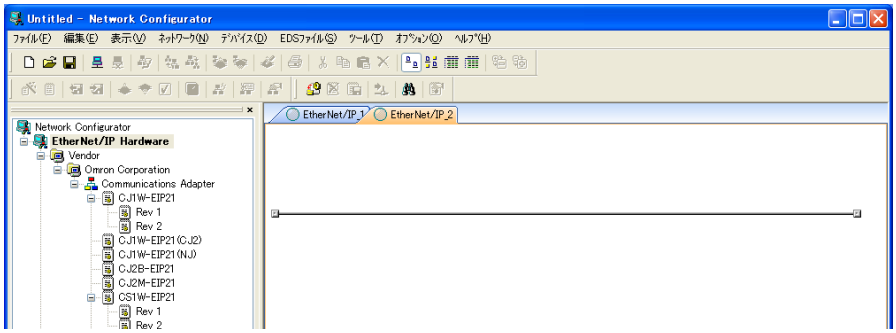


メインウィンドウ

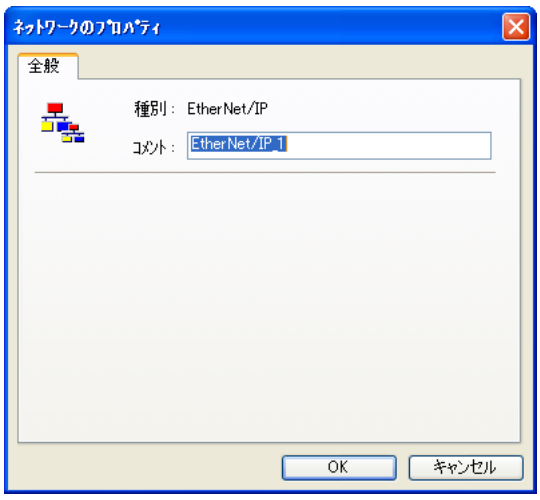
メインウィンドウは、以下のようにハードウェアリストとネットワーク構成ウィンドウで構成されています。



複数のネットワークを管理する場合は、［ネットワーク］｜［追加］を選択してください。新規のネットワーク構成ウィンドウを追加する事ができます。



ネットワークのタブに表示される名称を変更するには、［ネットワーク］｜［プロパティ］を選択してください。ネットワークのプロパティ画面のコメント欄に設定した名称に変更できます。



7-2-2　タグデータリンク設定手順

タグデータリンク（コネクション情報）の設定手順を以下に示します。
コントローラ－コントローラ間のデータリンクの場合、コネクション情報はオリジネータ側、つまり、データ入力として受信する側にのみ設定します。

1　ネットワーク構成の作成

(1) EtherNet/IP のネットワーク構成ウィンドウにコネクションを作成する EtherNet/IP ユニットをすべて登録します。（🕮「7-2-3 デバイスの登録」参照）

※ 既にシステムがある場合は EtherNet/IP ネットワークにオンライン接続しネットワークの構成をアップロードします。
（🕮「7-2-10 タグデータリンクパラメータのアップロード」参照）

↓

2　タグ・タグセット・コネクションの作成

(1) すべての登録デバイス（EtherNet/IP　ユニット）に対し、タグ／タグセットを作成する。（🕮「7-2-4 タグ／タグセットの作成」参照）

(2) オリジネータデバイス（入力としてデータを受信する登録デバイス）に対し、コネクションを作成する。（🕮「7-2-5 コネクション設定」参照）

↓

3　タグデータリンクパラメータのダウンロード（🕮「7-2-9 タグデータリンクパラメータのダウンロード」参照）

↓

4　EtherNet/IP ユニットの LED（🕮「15-2 LED 表示の意味と異常時の処置」参照）および Network Configurator のモニタ機能（🕮「15-1 Network Configurator によるステータスの確認方法」参照）を利用し、タグデータリンクが正常動作していることを確認します。

↓

5　Sysmac Studio のウォッチウィンドウを利用し、出力タグのデータが入力タグに反映されていることを確認します。

注：使用方法は 🕮『Sysmac Studio Version 1 オペレーションマニュアル』（SBCA-470）を参照してください。

参考

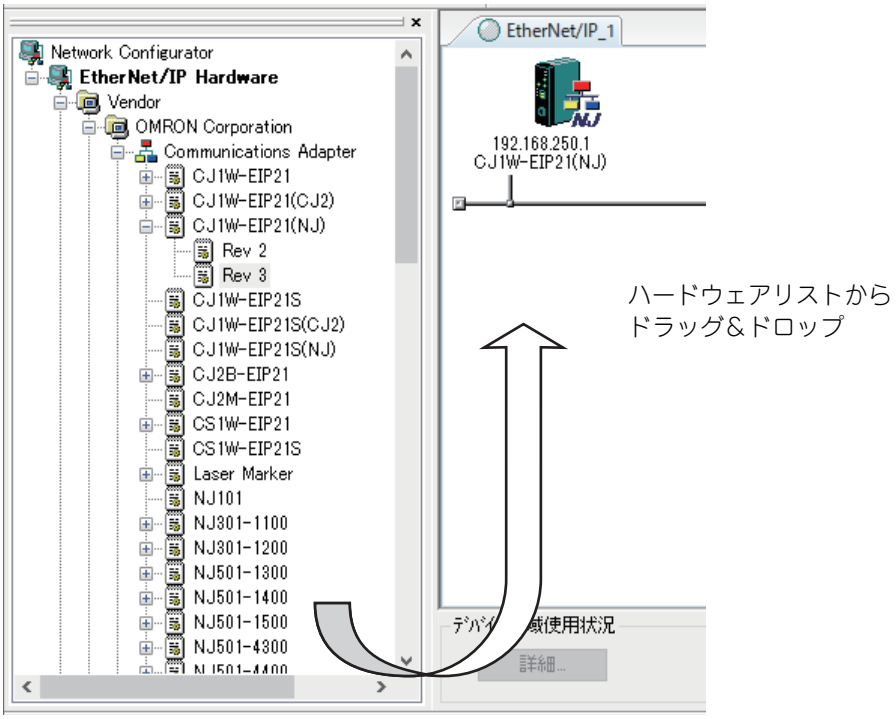
EDS ファイルを持たないデバイスとタグデータリンクを行う場合は、Generic デバイスを使用して設定を行います。
Generic デバイスを使用した設定方法は、「A-10 Generic デバイスを用いたタグデータリンク設定（Network Configurator Ver.3.74 以降）」を参照してください。

7-2-3　デバイスの登録

タグデータリンクを行う EtherNet/IP ユニットなど、設備に必要なすべてのデバイスをネットワーク構成として登録します。

1 **ハードウェアリストから、タグデータリンクを行うデバイスをドラッグ（マウスの左ボタンを押しながら移動）し、右側のネットワーク構成上にドロップ（マウスの左ボタンを解除）して登録します。**

ハードウェアリストのデバイスを選択し、ENTER キーを入力して登録することもできます。
以下のようにデバイスのアイコンがネットワーク構成上に表示されます。



ハードウェアリスト上には、デバイス名および、CIP のメジャー・リビジョンが「Rev □」と表示されます。
EtherNet/IP ユニットのデバイス名および CIP のメジャー・リビジョンは、以下のとおりです。

ハードウェアリスト上の名称	CIP リビジョン	ユニットバージョン
CJ1W-EIP21(NJ)	Rev.2	Ver.2.1
	Rev.3	Ver.3.0
CJ1W-EIP21S(NJ)	Rev.4	Ver.1.0

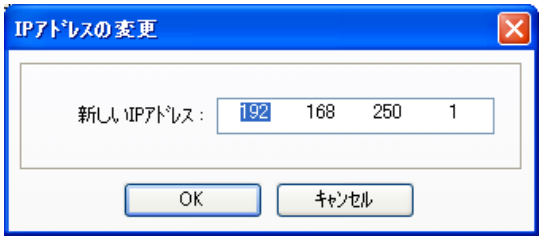
使用上の注意

登録するデバイスは、実際に使用するデバイスのデバイス名および CIP のメジャー・リビジョンと同じものを選択してください。いずれかが異なる場合に、Network Configurator でタグデータリンクパラメータをダウンロードした時の動作は以下のとおりです。

- デバイス名が異なる場合、「デバイスにアクセスできないか、デバイス種別が違います」と表示され、ダウンロードされません。
- CIP のメジャー・リビジョンが異なる場合、「ユニットリビジョンが違います」と表示され、ダウンロードされません。

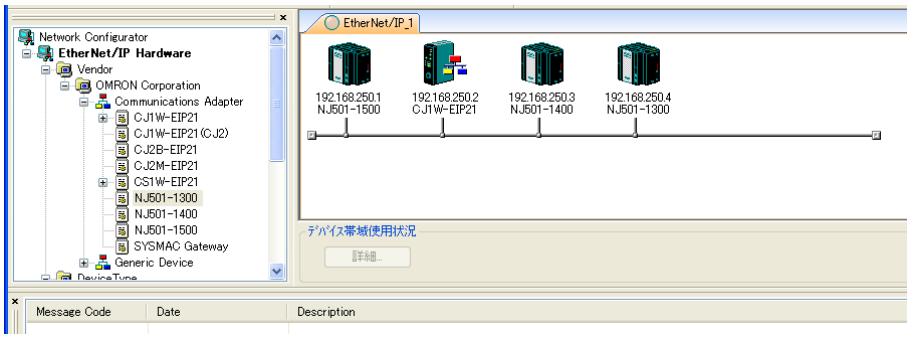
上記は、タグデータリンクパラメータのアップロードおよび照合する場合も同様です。
いずれの場合も、「7-2-17 デバイスの変更」を参照して、デバイスを変更してください。

2 **登録したデバイスのアイコン上でマウスを右クリックし、ポップアップメニューから［ノードアドレスの変更］を選択します。**



3 **実際に使用するデバイスのノードアドレス（IP アドレス）に合わせて設定したあと、［OK］ボタンをクリックします。**

4 **手順１～３の操作を繰り返して、タグデータリンクを行うすべてのデバイスを登録します。**



登録デバイスと使用するデバイスの CIP のメジャー・リビジョンの関係

登録したデバイスと実際に使用するデバイスの CIP のメジャー・レビジョンの組み合わせで、Network Configurator でタグデータリンクパラメータのダウンロード、アップロードおよび照合が可能かどうかが決まります。下記の表のとおりです。

○：可　×：不可

登録したデバイスの CIP メジャー・レビジョン	実際に使用するデバイスの CIP メジャー・レビジョン		
	2	3	4 以降
2	○	○	○
3	×	○	○
4	×	×	○

7-2-4　タグ／タグセットの作成

登録した内蔵 EtherNet/IP ポート／ EtherNet/IP ユニットに対しコネクションの作成に必要なタグセット、およびそのメンバとなるタグを作成します。タグに対しては制御プログラムで使用するネットワーク変数を設定することができます。
本章では、まずタグ／タグセットの基本的な作成方法として下記 (1) を説明します。

その後つづけて、Sysmac Studio からネットワーク公開属性の変数を、Network Configurator にインポートする機能を（2）で説明します。

(1)　Network Configurator の「デバイスパラメータの編集」機能を使って作成する方法

(2)　Sysmac Studio で作成したネットワーク公開属性の変数を、Network Configurator にインポートする方法

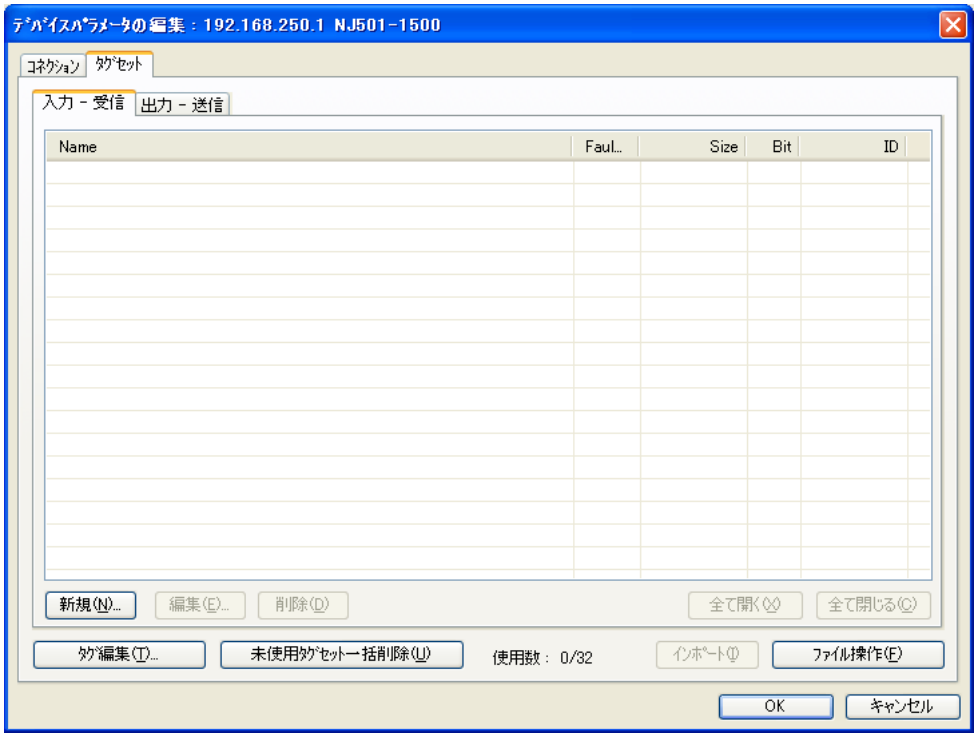
(1) Network Configurator の「デバイスパラメータの編集」機能を使って作成する方法

● **タグセットの作成**

1 **タグセットを作成するデバイスのアイコンをダブルクリックし、[デバイスパラメータの編集] ダイアログボックスを表示します。アイコンを右クリックし、ポップアップメニューから［パラメータ］｜［編集］を選択しても表示します。**

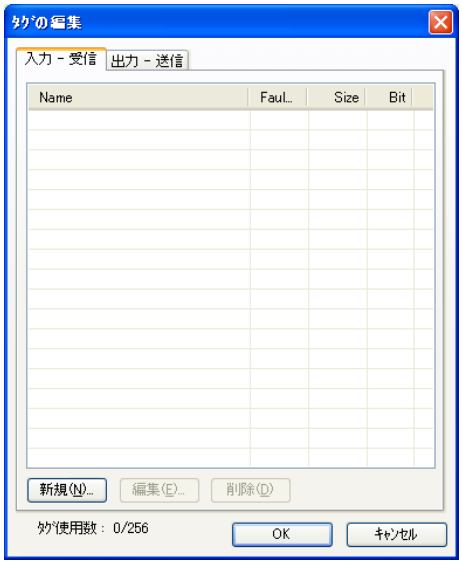


2 **［デバイスパラメータの編集］ダイアログボックスの［タグセット］タブをクリックします。**
タグセットは入力（受信）と出力（送信）、それぞれにタブがあります。

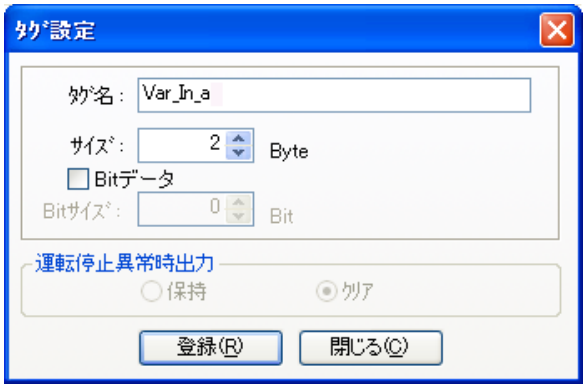


● タグの新規作成／追加

3 **［タグ編集］ボタンをクリックします。**
［タグの編集］ダイアログボックスが開きます。
タグは入力（受信）と出力（送信）を別に登録します。



4 **［入力－受信］タブを選択し ［新規］ボタンをクリックします。**
［タグ設定］ダイアログが開きます。



5 **「タグ名」の欄に、変数名を直接入力します。（例：Var_In_a）。**

参考

- タグ名には、以下の文字が使用できます。
  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）
- タグ名には、以下の文字が使用できません。
  「!」、「 “」、「#」、「$」、「&」、「 ‘」、「(」、「)」、「*」、「+」、「,」、「-」、「.」、「/」 、「:」、「;」、「<」、「=」、「>」、「?」、「@」、「[」、「] 」、「^」、「`」、「%」、「¦」、「￥」、「{」、「}」、空白、数字 (0 ～ 9) で始まる文字列
- タグ名の長さは、最大 255 バイトです。
- 変数に配列や構造体の属性がある場合、以下のように指定します。
  - 配列の要素指定
    例： array [2][3]（または array [2,3]）
    　　array [2][3][4]（または array [2,3,4]）
  - 構造体のメンバ指定
    例：Struct.member（メンバを ” .” で区切る）
- タグに特定の I/O メモリアドレスを割り付けたい場合は、Sysmac Studio にて、その I/O メモリアドレスを AT 指定した変数を作成してください。
  Network Configurator で、その AT 指定した変数をタグに指定すれば、AT 指定した I/O メモリアドレスに対してタグデータリンクさせることができます。

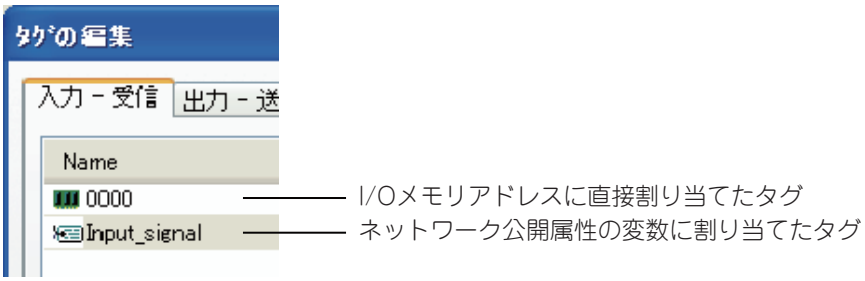
使用上の注意

タグには変数名を指定してください。
I/O メモリアドレスをタグに指定したい場合は、タグに I/O メモリアドレスを直接指定するのではなく、Sysmac Studio で I/O メモリアドレスを AT 指定した変数を作成し、その AT 指定した変数を、タグとして指定してください。

また、Network Configurator では、以下の I/O メモリアドレスを表す表記をタグ名として入力すると、そのタグは変数ではなく、I/O メモリアドレスに直接割り付けられますので、タグには変数名を指定するようにしてください。

1) 半角数字のみからなる変数名（ 0000 ～ 6143）
2) 以下の半角英字（大文字／小文字とも）と、半角数字からなる変数名
   - H（H000 ～ H511）
   - W（W000 ～ W511）
   - D（D00000 ～ D32767）
   - E0_ ～　E18_（E0_00000 ～ E0_32767...E18_00000 ～ E18_32767）

［タグの編集］ダイアログボックスでは、タグの割付け先をアイコン表示によって、以下のように確認することができます。



6 **「サイズ」の欄に、タグのサイズを、バイト単位で入力します。**

タグのサイズは変数のデータ型サイズと同じになるように入力します。
BOOL 型変数の場合、「Bit データ」にチェックを入れ、「Bit サイズ」に「1」を入力します。

使用上の注意

2012 年 10 月以降製造の EtherNet/IP ユニットは、奇数バイトのタグサイズが指定可能です。ご使用になる場合には、次の点に注意してください。

- CPU ユニット上のメモリは 2 バイト単位で消費します。
  そのため、NetworkConfigurator で指定したい奇数バイトのタグサイズを 1 バイト単位で設定し、SysmacStudio ではそのサイズに 1 バイト足したサイズの変数を作成し、タグとして指定してください。

7 **［登録］ボタンをクリックし、タグを登録します。**

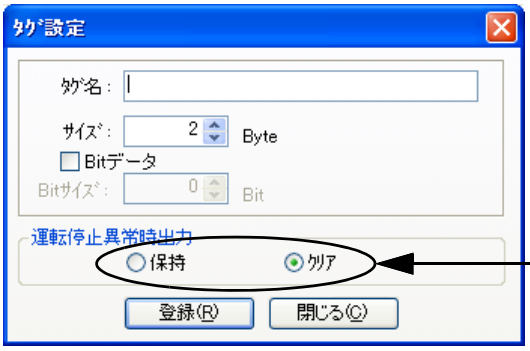
タグ名に I/O メモリアドレスを指定した場合、次のタグ編集用に、連続したアドレスをタグ名とした［タグ設定］ダイアログボックスが開きますので、続けてタグを登録することができます。
タグの登録が一通り終了した後、［閉じる］をクリックします。

8 **引き続き、［出力ー送信］タブを選択し ［新規］ボタンをクリックします。**

［タグ設定］ダイアログが開くので同様に入力していきます。
出力（送信）タグの「運転停止異常時出力」で、CPU ユニットに全停止フォールトレベルのコントローラ異常が発生した時に、出力データをクリアするか、全停止フォールト異常が発生する前のデータをそのまま送信し続けるかを選択します。

入力（受信）タグセットでは、「運転停止異常時出力」の設定は不要です。

•全停止フォールトレベルのコントローラ異常時に出力を保持する：［保持］（デフォルト）
　全停止フォールトレベルのコントローラ異常後も全停止フォールト異常が発生する前の出力データを送信し続けます。
•全停止フォールトレベルのコントローラ異常時に出力をクリアする：［クリア］
　全停止フォールトレベルのコントローラ異常発生時に出力データをクリア（0 に）して送信し続けます。



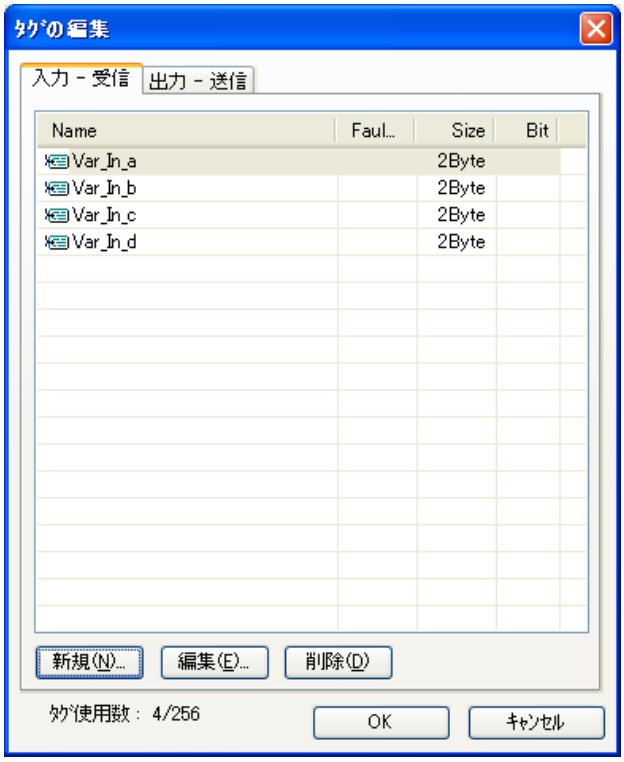
保持またはクリアを選択します。

使用上の注意

タグデータリンク中にオリジネータ側のコントローラで以下に示す異常要因が発生した場合は、コネクションが強制的に切断されます。

- 全停止フォールトレベルのコントローラ異常
- 部分停止フォールトレベルのコントローラ異常

9 **必要なタグの登録が終了したら、［タグの編集］ダイアログの［OK］ボタンをクリックします。**

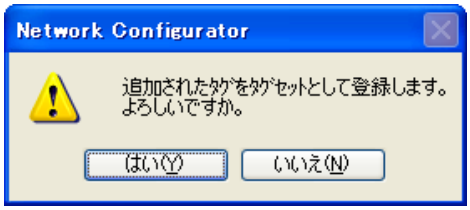


使用上の注意

同一タグセット内のタグのデータを同じタイミングで更新させたい場合、以下のように設定してください。

- 同一タグセット内のタグに指定する変数群は、あらかじめ Sysmac Studio にて、同じタスクを「更新タスク」として指定してください。
- I/O メモリを AT 指定したタグ（変数）と、AT 指定していないタグ（変数）は、同一タグセット内で混在させないでください。

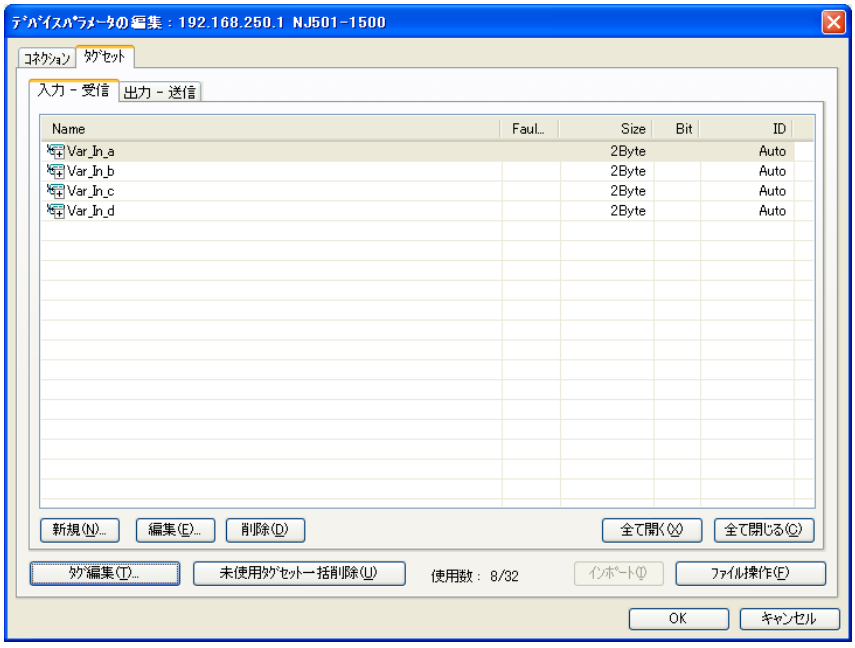
10 **このとき、登録したタグ名をそのままタグセット名として登録するか、確認メッセージが表示されます。タグセットは最大 8 個までのタグで構成できますが、タグ名をそのままタグセット名として登録した場合、1 個のタグ＝1 個のタグセットで登録されます。**
ここでは、［はい］をクリックします。



［いいえ］をクリックし、タグセットへタグを後から追加することも可能です。新規にタグのみを登録し、後からタグセットへタグを追加する手順は、手順 18. を参照してください。

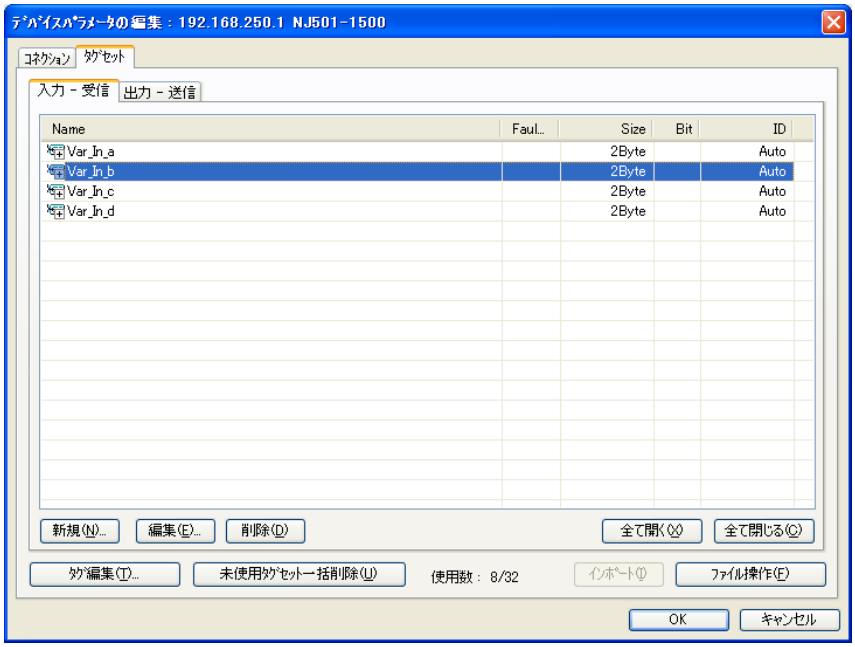
● タグセットの変更と登録

11 **［タグの編集］から、そのままタグセット名として登録した場合、以下のような表示になります。**

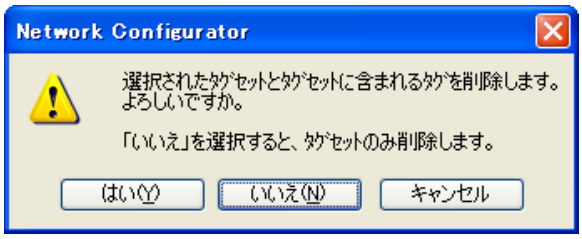


12 **すでに入力タグセットに登録されている入力タグを、他の入力タグセットへ登録変更するには、すでに登録されたタグの登録を解除する必要があります。**

［デバイスパラメータの編集］ダイアログボックスの［タグセット］タブで、登録変更したいタグを含むタグセットを選択し、［削除］ボタンをクリックして下さい。（すでにタグセットに登録されているタグが複数ある場合は、［タグセットの編集］ダイアログにより登録解除したいタグを選択し、>> ボタンで選択したタグの登録を解除する事が可能です。）

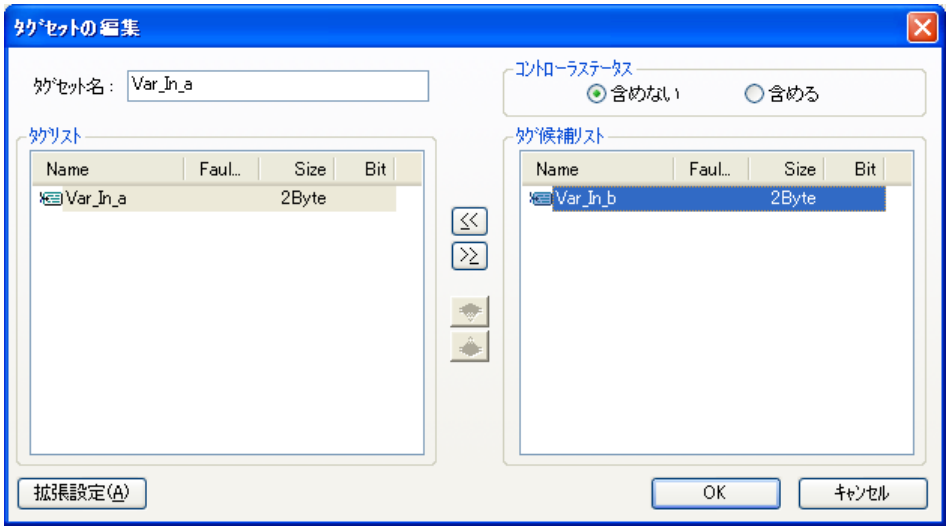


このとき、選択されたタグセットとタグセットに含まれるタグを削除するか確認されます。



［いいえ］をクリックすると、タグセットのみが削除されます。［いいえ］をクリックします。

13 **登録したタグセットを編集してタグを追加するには、編集したいタグセットを選択しダブルクリックするか、［編集］ボタンをクリックします。**
［タグセットの編集］ダイアログが開きます。



左側の［タグリスト］にすでにタグセットに登録されているタグが、右側の［タグリスト候補］にタグセットには登録されていないその他のタグ候補がリストアップされています。［タグリスト候補］の中から追加したいタグを選択し、  ボタンをクリックします。

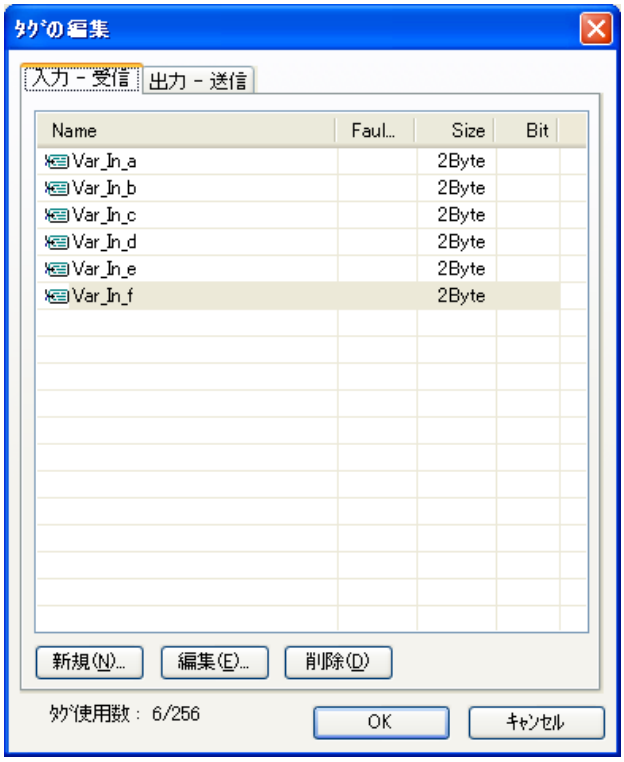
14 **タグセットにコントローラステータスを含める場合は、［タグセットの編集］ダイアログボックス右上の［コントローラステータス］を［含める］にします。**



15 **変更を確定するのに、［タグセットの編集］ダイアログボックスの [OK] ボタンをクリックします。**

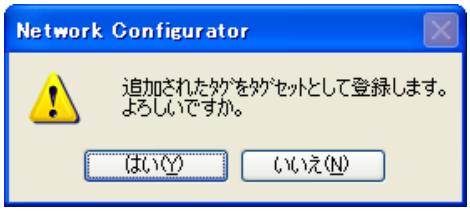
16 **［デバイスパラメータの編集］ダイアログボックスの [OK] ボタンをクリックします。**

17 新規にタグのみを追加しタグセットに登録したい場合は、まず手順 1 ～手順 9 でタグを登録します。ここでは、新規に Var_In_e、Var_In_f の入力タグを作成します。



18 必要なタグの登録が終了したら、［タグの編集］ダイアログの [OK] ボタンをクリックします。

19 このとき、登録したタグ名をそのままタグセット名として登録するか、確認されます。ここでは、新規にタグのみを追加するので、［いいえ］ボタンをクリックしてください。タグセットとして登録されずに、タグのみが登録されます。

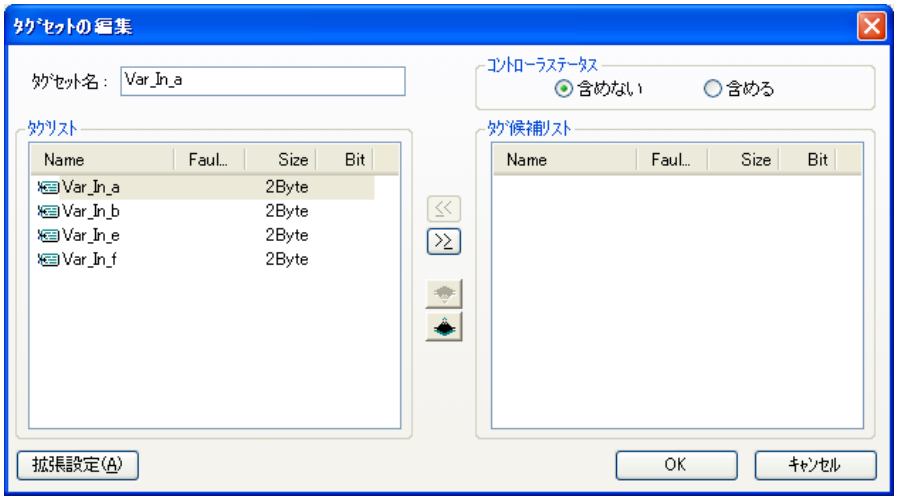


20 **新規に追加したタグをタグセットに登録するには、登録したいタグセットを選択し、ダブルクリックするか、[編集] ボタンをクリックしてください。**



左側にすでにタグセットに登録されているタグが、右側には、タグセットには登録されていないその他のタグ候補がリストアップされています。

21 **タグ候補の中から追加したいタグを選択し、<<ボタンをクリックします。**



タグセットに登録可能なタグの最大数は、8 個です。（タグセットにコントローラステータスを含める場合、タグの最大数は 7 個となり、サイズが 2 バイト追加されます。）
タグリストに表示される並びでタグセット内のデータが受け渡されます。並びを変更する場合は、変更したいタグを選択し、、ボタンでタグの位置を並び替えます。

22 **変更を確定するのに、[タグセットの編集] ダイアログボックスの [OK] ボタンをクリックします。**

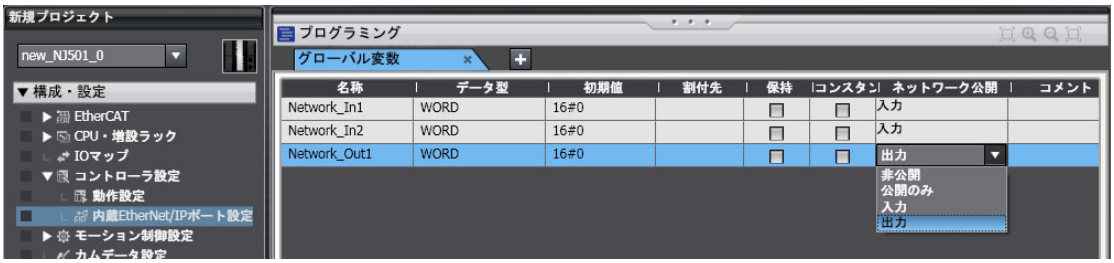
23 **[デバイスパラメータの編集] ダイアログボックスの [OK] ボタンをクリックします。**

(2) Sysmac Studio で作成したネットワーク公開属性の変数を、Network Configurator にインポートする方法

Sysmac Studio でネットワーク変数を作成し、これを Network Configurator にインポートして、タグ / タグセットを作成することができます。以下にその手順を示します。

● Sysmac Studio によるグローバル変数のエクスポート

1 **あらかじめ、Sysmac Studio のグローバル変数エディタで、グローバル変数を作成し、さらにグローバル変数の「ネットワーク公開」属性を「入力」または「出力」に設定します。**



2 **［ツール］｜［グローバル変数のエクスポート］｜［Network Configurator...］を選択します。「ネットワーク変数」属性を「入力」または「出力」に設定したグローバル変数のみが、後述の変数のインポート操作（● Network Configurator へのインポート）によって、csv ファイルがインポートされます。**

● Network Configurator へのインポート

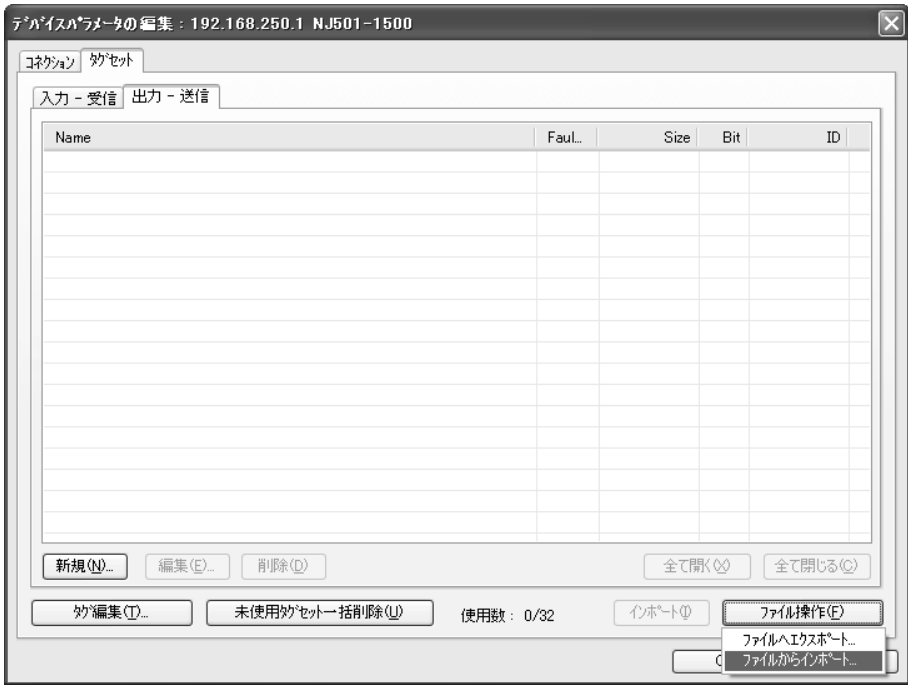
使用上の注意

Sysmac Studio でネットワーク属性にした変数のうち、「0000」や「H0000」など、I/O メモリアドレスの表記と同じ変数名の変数は、CSV ファイルにエクスポートされません。
1) 半角数字のみからなる変数名 (例 :001)
2) 以下の半角英字（大文字 / 小文字とも）と、半角数字からなる変数名
- H（例：H30）
- W（例：W30）
- D（例：D100）
- E0_ ～　E18_（例：EA_100）

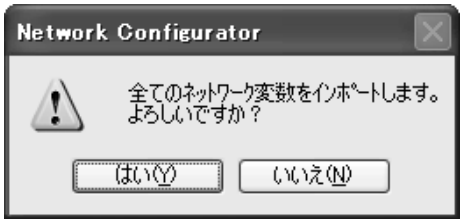
1　Network Configurator で登録しておいたデバイスから、ネットワーク公開属性にした変数をインポートさせたいアイコンをダブルクリックし、［デバイスパラメータの編集］ダイアログボックスを表示します。

アイコンを右クリックし、ポップアップメニューから［デバイス］｜［パラメータ］｜［編集］を選択しても表示します。

2　［デバイスパラメータの編集］ダイアログボックスの［タグセット］タブで、［ファイル操作｜ファイル操作のインポート］ボタンをクリックします。



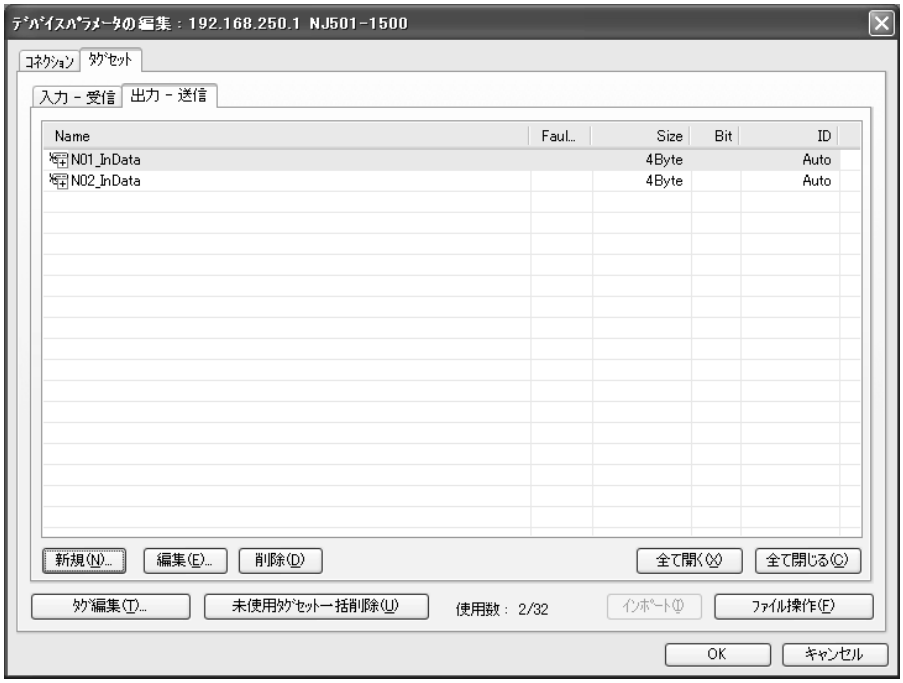
- ネットワーク公開属性にしたすべての変数をタグとしてインポートする場合は、［はい］ボタン、一部のみをインポートする場合は［いいえ］ボタンをクリックします。



変数をタグにインポート後、タグセットまで自動的に作成する場合は、［はい］ボタン、タグセットは手動で設定する場合は、［いいえ］ボタンをクリックします。



上記で［はい］を選択すると以下に示すように、［タグセット］タブで変数がインポートされ、1 変数＝ 1 タグセットとして、自動的にデバイスパラメータに登録されます（タグセット名は、変数名が割り付けられます）。



Sysmac Studio からインポートした複数の入力変数（入力タグ）を 1 つのタグセットにまとめる場合は、すでに入力タグセットに登録されている入力タグの登録を解除する必要があります。

タグセットにまとめる変数が含まれるタグセットを選択し、［削除］ボタンをクリックしてください。このとき、選択されたタグセットとタグセットに含まれるタグを削除するか確認されます。ここでは、タグセットのみを削除するので、［いいえ］をクリックします。



新規にタグセットを作成してタグをまとめる場合は、［新規］ボタンをクリックします。すでに登録されているタグセットに追加する場合は各タグセットをダブルクリックするか［編集］ボタンをクリックします。
［タグセットの編集］ダイアログボックスが表示され、インポートしたすべての変数のうち、その他のタグセットに登録されていないタグは、［タグセットの編集］ダイアログボックスの右側に［タグ候補リスト］としてリストアップされます。
ボタンでタグを個別に追加していきます。



3 **タグセット名を変更したい場合は、この画面で変更します。変更を確定するのに、［タグセットの編集］ダイアログボックスの［登録］ボタンをクリックします。**

4 **タグデータリンクを行うすべてのデバイスについて、手順１～手順３を実行し、変数のインポートとタグセットの作成をします。**

7-2-5　コネクション設定

タグセットの作成後、［デバイスパラメータの編集］ダイアログボックスの［コネクション］タブで、以下のコネクション情報を設定します。

- コネクションを開設するターゲットデバイス／タグセット
- コネクションタイプ（マルチキャスト／ユニキャスト）
- パケットインターバル（RPI）
- コネクション名（省略可）

コネクション設定は、オリジネータ側でのみ設定します。ターゲットデバイス側では、コネクション設定は不要です。

使用上の注意

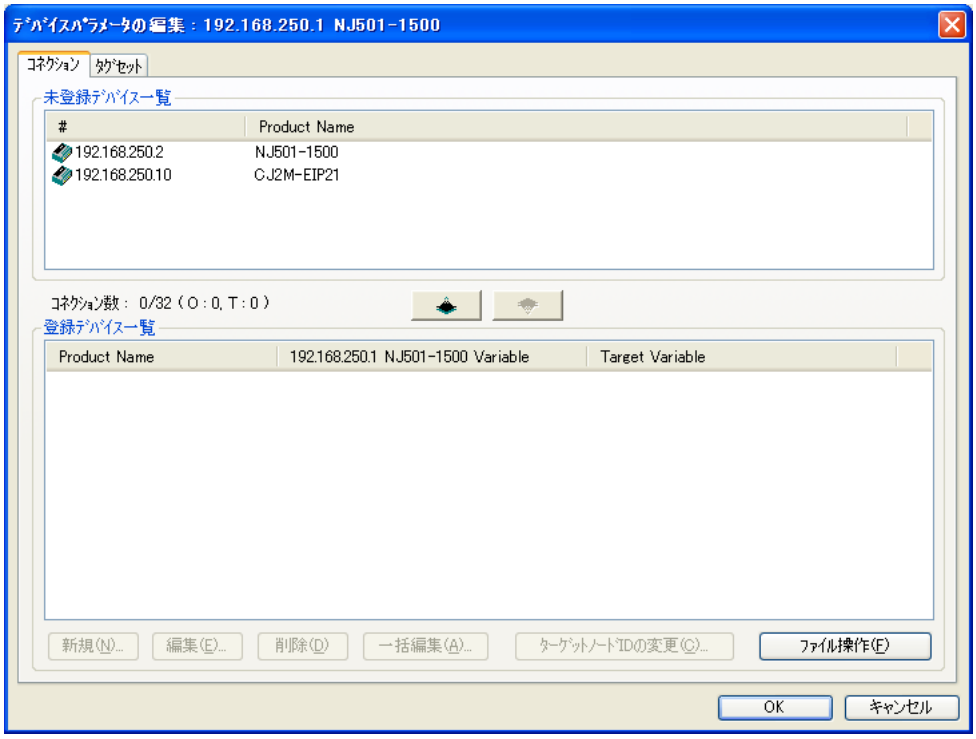
コネクション設定は、タグデータリンクを行う全デバイスのタグセットを作成してから行ってください。

コネクション設定（［コネクション］タブ）

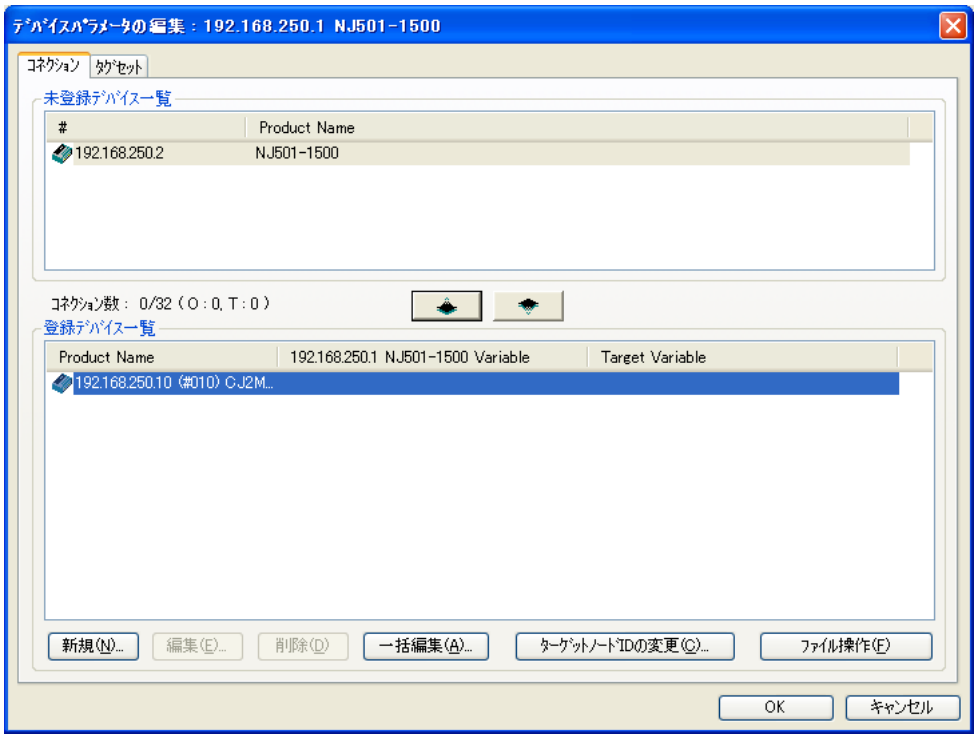
● 登録デバイス一覧への登録

1 **Network Configurator のネットワーク構成ウィンドウで、オリジネータ設定を行うデバイスのアイコンをダブルクリックし、［デバイスパラメータの編集］ダイアログボックスを表示します。**
アイコンを右クリックし、ポップアップメニューから［パラメータ］｜［編集］を選択しても表示します。

2 **［デバイスパラメータの編集］ダイアログボックスの［コネクション］タブをクリックします。**
ネットワークに登録されているすべてのデバイス（自ノード以外）が表示されます。



3 **「未登録デバイス一覧」で、コネクション設定を行うターゲットデバイスを選択し（デバイスをクリックすると選択ノードが灰色になる）、その後、ボタンをクリックします。**
以下に示すように、選択したターゲットデバイスが「登録デバイス一覧」に表示されていきます。



4 **「登録デバイス一覧」に登録されたデバイスには、ターゲットノード ID が付与されます。**
このターゲットノード ID は、オリジネータ側コントローラのユニット運転情報、ターゲットノード PLC 本体異常情報、ターゲットノード異常情報、登録ターゲットノード情報、正常ターゲットノード情報のビット配列位置になります。ターゲットノード ID はデフォルトで IP アドレスの下位 8 ビットが自動的に割り付けられます。上記の例ではターゲットデバイスの IP アドレスは 192.168.250.2 なので、デバイス No. は #002 になります。まれに、ターゲットノード ID が重複するようなケースでデバイス No. を変更したい場合は、「ターゲットノード ID の変更」ボタンをクリックし変更します。



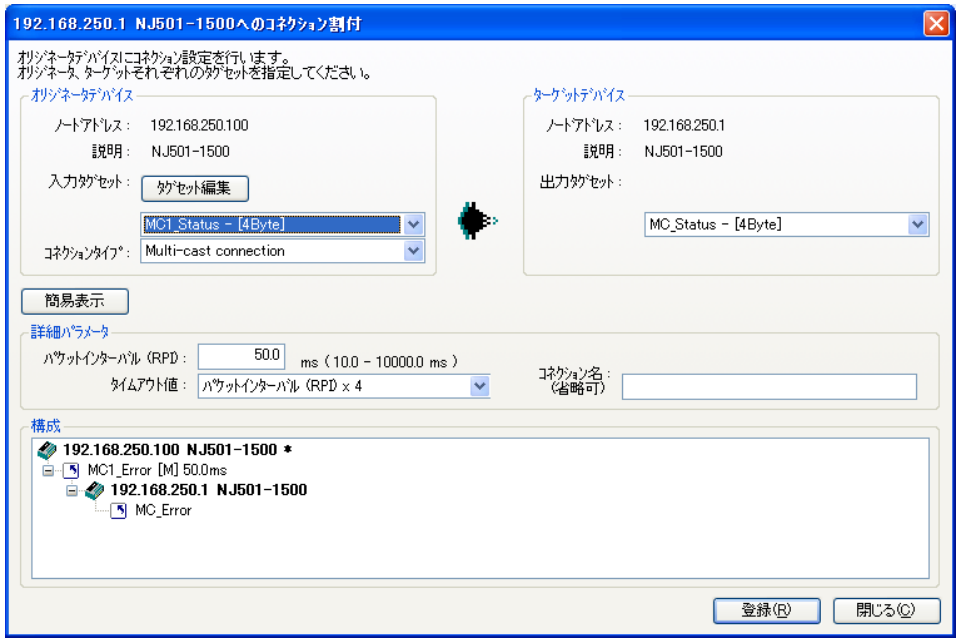
● コネクション設定（個別編集）

１つ１つのコネクションを個別に編集します。

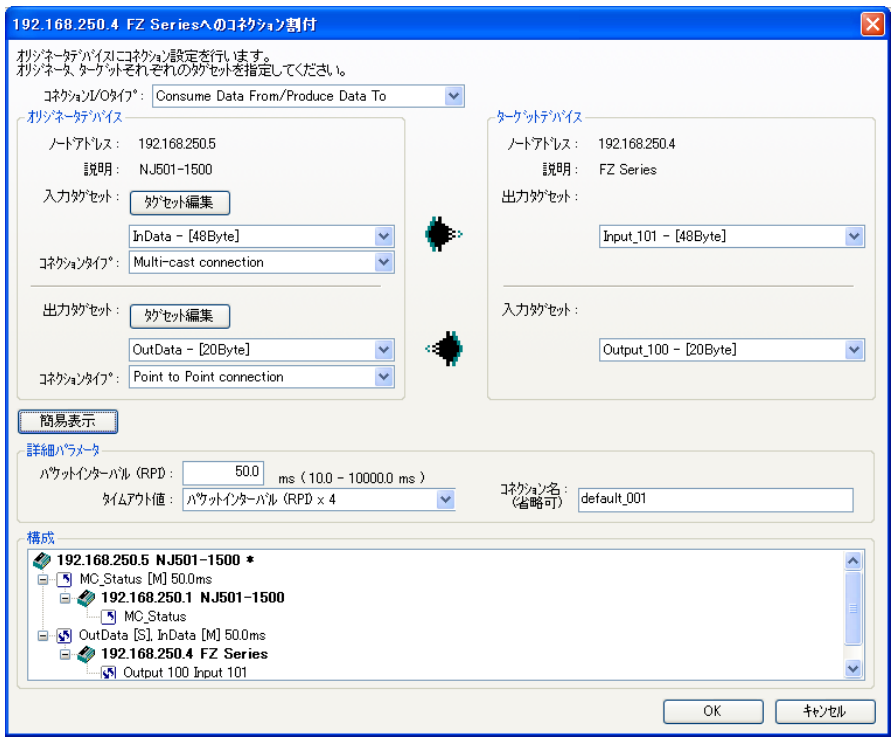
（注）テーブル形式で一括編集する方法については、次項を参照してください。

1　［コネクション］タブを選択して、［新規］ボタンをクリックします。
選択しているデバイスの種別に応じて、以下の［コネクション割付］ダイアログボックスが表示されます。

(A) ターゲットが内蔵 EtherNet/IP ポートの場合（Input Only の場合）



(B) ターゲットがその他の EtherNet/IP デバイスの場合（Input Only 以外の場合）



設定内容は以下のとおりです。

設定項目	内容
コネクション I/O タイプ	CS1W-EIP21、CS1W-EIP21S、CJ1W-EIP21、CJ2B-EIP21、CJ2M-EIP21、CJ1W-EIP21(CJ2)、CJ1W-EIP21(NJ)、CJ1W-EIP21S、CJ1W-EIP21S(CJ2)、CJ1W-EIP21S(NJ)、NX701、NJ501-□□□□、NJ301-□□□□、NJ101 でタグデータリンクする場合は、Input Only(Tag type) を選択します。 その他のターゲットデバイスとタグデータリンクする場合は、対象デバイスの EDS ファイルに指定されたコネクション I/O タイプを選択します。 「Input Only(ID type)」は、他社ノードがオリジネータで Tag type によるコネクション設定をサポートしていない場合に使用します。
コネクションタイプ	データをマルチキャスト／ユニキャスト、いずれで送信するかを選択します。デフォルトはマルチキャストです。 • Multi-cast connection：　同一データを複数ノードで共有するときに選択します。通常はこちらを選択します。 • Point to Point connection：同一データを複数ノードで共有しないときに選択します。ユニキャストで通信されるので、その他のノードに必要のない負荷を与えたくない場合に選択します。 ※ マルチキャストとユニキャストの使い方、コネクション数の数え方については、📖「7-1-4 使用方法の概要」を参照してください。
［簡易表示］ボタンを押した場合［構成］フィールドおよび以下の項目は表示されません。	
パケットインターバル (RPI)	オリジネータとターゲット間の、コネクションごとのデータ更新周期（パケットインターバル）を設定します。 0.5 ～ 10000ms の範囲で 0.5ms 単位（NJ シリーズコントローラ：10 ～ 10000ms の範囲で 1ms 単位）で設定できます。 デフォルトは 50ms です（50ms に 1 回の更新になります）。
タイムアウト値	コネクションタイムアウトを検知する時間を設定します。設定は、パケットインターバル (RPI) の 4、8、16、32、64、128、256、512 倍から選択可能です。 デフォルトは、パケットインターバル (RPI) の 4 倍です。
コネクション名	コネクションごとに名前を付けます。(最大半角 32 文字)

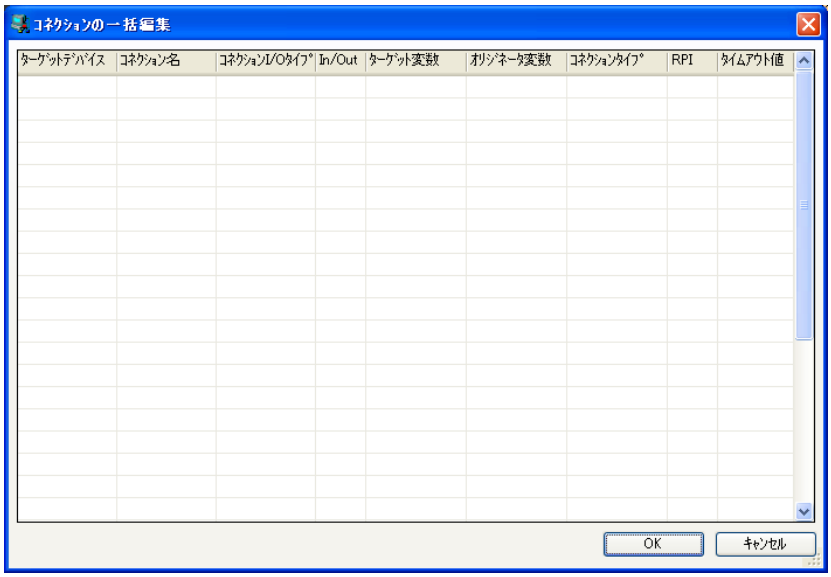
2　設定終了後、［登録］ボタンをクリックします。

● コネクション設定（一括編集）

オリジネータと、登録デバイス一覧で選択したすべてのターゲットデバイスとのコネクション設定を一覧表で一括編集します。

1　［コネクション］タブを選択して、［一括編集］ボタンをクリックします。

以下の［コネクションの一括編集］ウィンドウが表示されます。



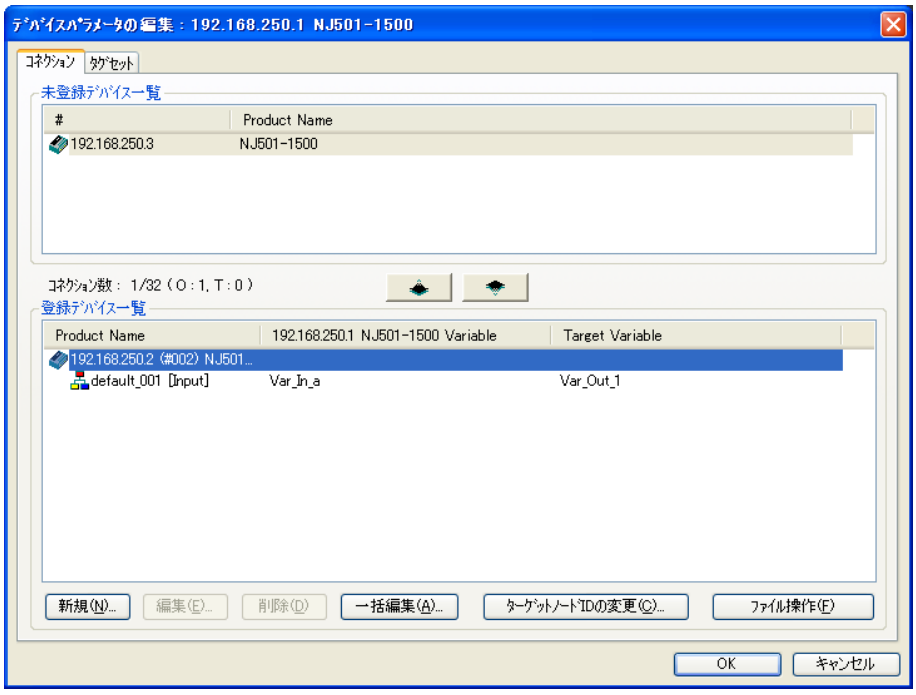
設定内容は、以下のとおりです。

設定項目	内容
ターゲットデバイス	ターゲットデバイスを選択します。
コネクション名	コネクションに任意の名称をつけることができます。(最大半角 32 文字) 空欄にしておくとデフォルトの名称がつきます。 このコネクション名は、コメントとして使用します。
コネクション I/O タイプ	CS1W-EIP21、CS1W-EIP21S、CJ1W-EIP21、CJ2B-EIP21、CJ2M-EIP21、CJ1W-EIP21(CJ2)、CJ1W-EIP21(NJ)、CJ1W-EIP21S、CJ1W-EIP21S(CJ2)、CJ1W-EIP21S(NJ)、NX701、NJ501-□□□□、NJ301-□□□□、NJ101 タグデータリンクする場合は、Input Only(Tag type)を選択します。 その他のターゲットデバイスとタグデータリンクする場合は、対象デバイスの EDS ファイルに指定されたコネクション I/O タイプを選択します。 「Input Only(ID type)」は、他社ノードがオリジネータで Tag type によるコネクション設定をサポートしていない場合に使用します。
In/Out	コネクションの選択にともない自動的に表示されます。 Input Only：In とだけ表示されます。
ターゲット変数	ターゲット側のタグセットを選択して、割り付けます。 • In：　ターゲットの出力（送信）タグセットを選択します。 • Out：ターゲットの入力（受信）タグセットを選択します。
オリジネータ変数	オリジネータ側のタグセットを選択して、割り付けます。 • In：　オリジネータの入力（受信）タグセットを選択します。 • Out：オリジネータの出力（送信）タグセットを選択します。
コネクションタイプ	データをマルチキャスト／ユニキャスト、いずれで送信するかを選択します。デフォルトはマルチキャストです。 • Multi-cast connection：　同一データを複数ノードで共有するときに選択します。通常はこちらを選択します。 • Point to Point connection：同一データを複数ノードで共有しないときに選択します。 ユニキャストで通信されるので、その他のノードに必要のない負荷を与えたくない場合に選択します。 ※ マルチキャストとユニキャストの使い方、コネクション数の数え方については、「7-1-4 使用方法の概要」を参照してください。
RPI	オリジネータとターゲット間の、コネクションごとのパケットインターバル (RPI) を設定します。 0.5 ～ 10000ms の範囲で 0.5ms 単位で設定できます。 デフォルトは 50ms です（50ms に 1 回の更新になります)。
タイムアウト値	コネクションタイムアウトを検知する時間を設定します。設定は、パケットインターバル (RPI) の 4、8、16、32、64、128、256、512 倍から選択可能です。 デフォルトは、パケットインターバル (RPI) の 4 倍です。

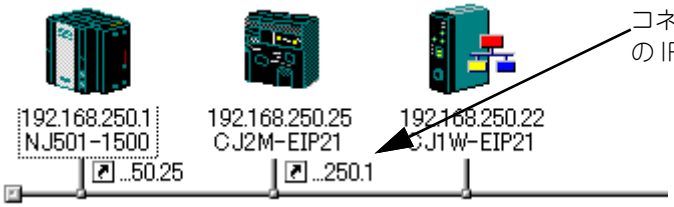
2　設定終了後、[OK] ボタンをクリックします。

● コネクション設定結果の確認

1　**登録デバイス一覧に設定したコネクションの概要が表示されます。**



2　**[OK] ボタンをクリックします。以下のように表示されます。**



コネクションが設定されているオリジネータの IP アドレスが表示されます。

3　**コネクション設定の手順を繰り返して、すべてのコネクションを設定します。**

使用上の注意

設定終了後、[デバイスパラメータの編集] ダイアログボックスを閉じて他の操作を行うときは、必ず、[OK] ボタンをクリックしてください。[キャンセル] ボタンをクリックして、ダイアログボックスを閉じると、設定内容が破棄されます。

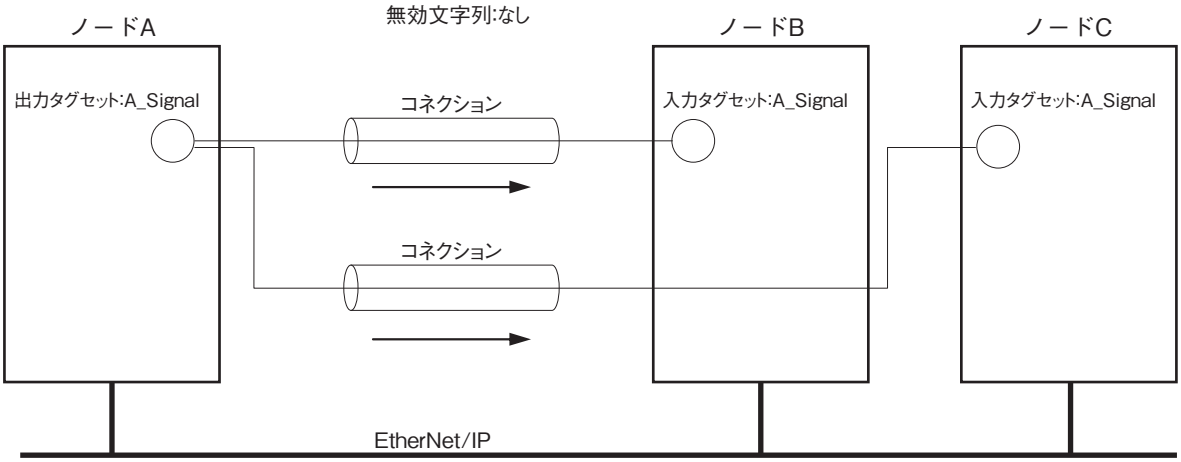
4　**コネクションを設定したのち、オリジネータかターゲットの一方でタグセットのサイズが変更されると、他方でサイズが合わなくなりパラメータ情報の不一致が発生します。**
そのため、コネクション設定の変更を行った場合は、必ず「コネクションのチェック」を実行してください。（詳細は 「7-2-16 コネクションのチェック」を参照してください。）

コネクションの自動設定（[ネットワーク] ー [自動コネクション設定]）

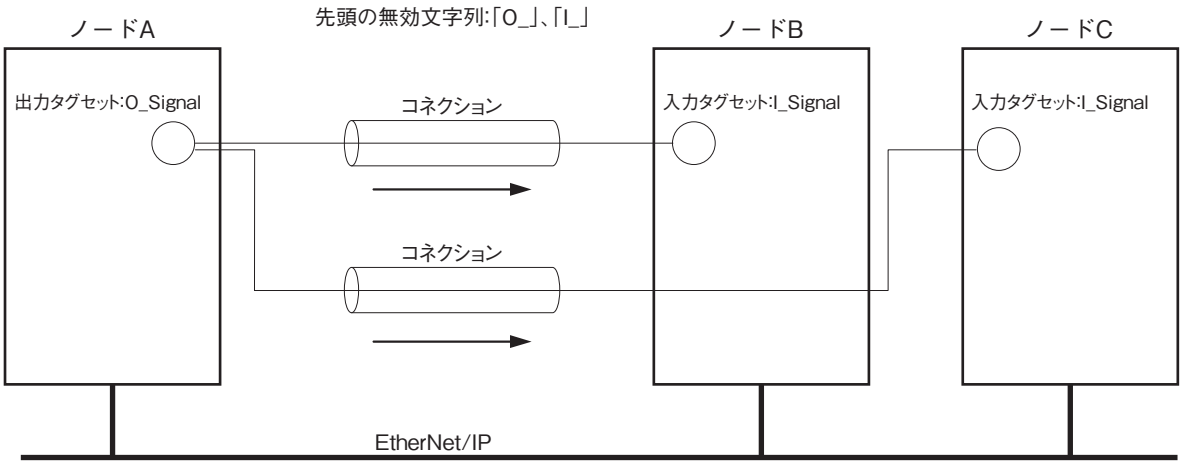
デバイスに設定されているタグセット名を自動識別させ、出力タグセットとタグセット名が同一（もしくは省略文字を除いて同一）の入力タグセットとのコネクション設定を自動で行うことができます。
自動コネクション設定は以下の条件で実行されます。

コネクション設定対象の出力側タグセット名	入力側タグセット名と同一（もしくは省略文字を除いて同一）であること。 文字列の先頭と終端に省略文字列を指定可能。
コネクション設定対象の入力側タグセット名	出力側タグセット名と同一（もしくは省略文字を除いて同一）であること。 文字列の先頭と終端に省略文字列を指定可能。
コネクションタイプ	Input Only 固定 マルチキャスト、ユニキャストは実行時指定可能
RPI	デフォルト値を使用
タイムアウト	デフォルト値を使用

- 例 1：同一のタグセット名の自動コネクション例
  ノード A のタグセット名「A_Signal」を出力（送信）、ノード B、C のタグセット名「A_Signal」を入力（受信）とした場合、同一のタグセット名を「A_Signal」として、以下のコネクションが自動設定されます。



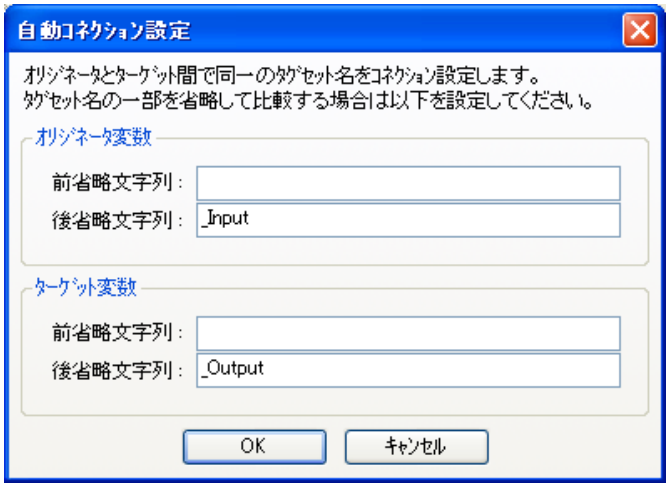
- 例 2：省略文字列を含んだ自動コネクション例
  ノード A のタグセット名「O_Signal」を出力（送信）、ノード B、C のタグセット名「I_Signal」を入力（受信）として、前省略文字列として、「O_」および「I_」を設定した場合、同一のタグセット名を「Signal」として、以下のコネクションが自動設定されます。



1 **コネクションの出力側と入力側のタグセット名を同じに設定します。前省略文字列、および後省略文字列を含めたタグセット名に設定することも可能です。**

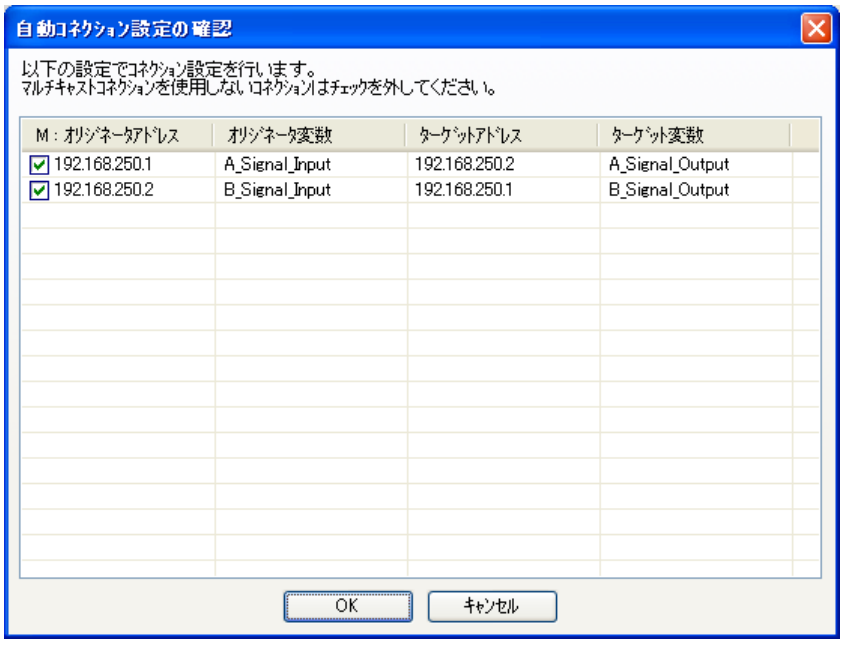
2 **［ネットワーク］メニューから［自動コネクション設定］を選択することで、コネクションが自動設定されます。**

自動設定処理開始時、入力タグセット名および出力タグセット名の、前省略文字列および後省略文字列の設定ウィンドウが表示されます。



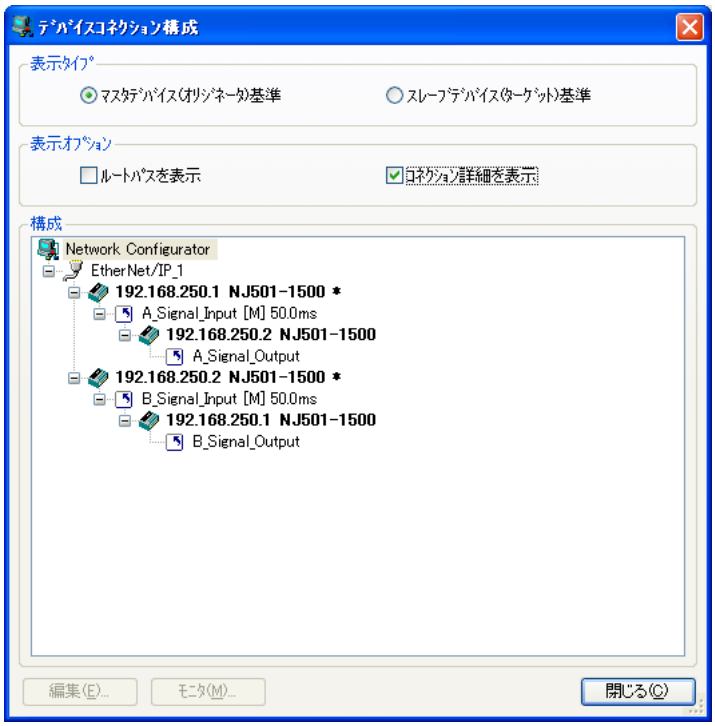
各々の条件を設定し［OK］ボタンをクリックすると、自動設定の処理が開始されます。

3 **条件に見合うタグセットが存在する場合は、その内容が表示されます。**



［OK］ボタンをクリックすると、自動設定の処理が開始されます。

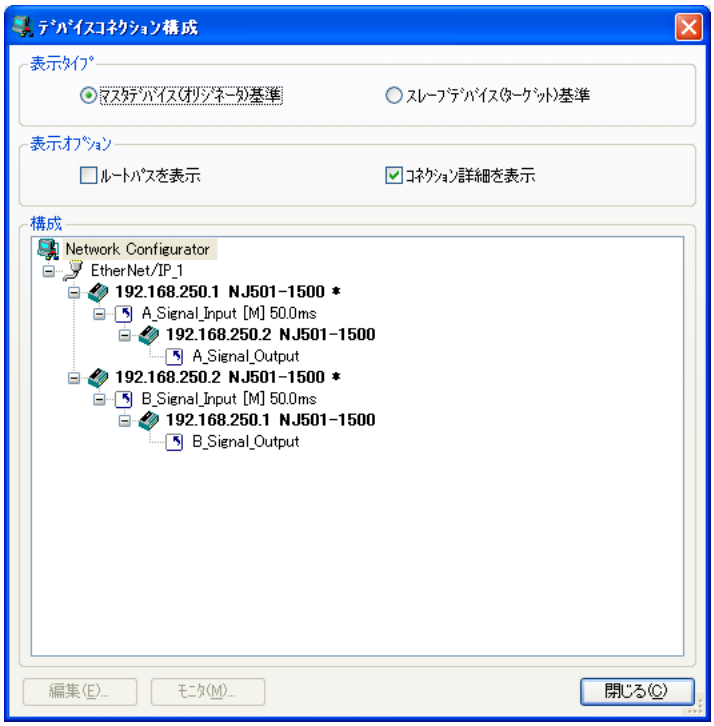
4　**自動設定実行後、コネクション構成表示で設定内容が表示されます。**



5　**(必要によりデバイスコネクションの構成表示などを使って、RPI ／タイムアウト値の調整を個別に行います。)**

デバイスコネクション構成表示

ネットワーク構成中のコネクション設定状況を一覧表示させることができます。
［ネットワーク］メニューから［デバイスのコネクション構成表示］を選択します。



- ［コネクション詳細を表示］オプションにより、タグデータリンクの送受信一覧をデバイス単位／コネクション単位で切り替え表示ができます。
- コネクション設定されているデバイス（オリジネータ）には、デバイス名の後ろに＊マークが付加されます。
- コネクションを選択して、［編集］ボタンをクリックすることにより、［デバイスパラメータの編集］画面が表示され、コネクションの編集を行うことができます。

7-2-6　ウィザードによるコネクションの作成

Network Configurator のウィザード機能を使用することにより、オムロン製 PLC 間のコネクションをガイドに従って簡単に作成することができます。

参考

ウィザード機能で扱うことのできる EtherNet/IP デバイスは、以下のオムロン製デバイスに限ります。

デバイス名	補足
CJ1W-EIP21(NJ)	NJ シリーズ CPU ユニットに、形 CJ1W-EIP21 を装着した場合
CJ1W-EIP21	CJ1 シリーズ PLC に形 CJ1W-EIP21 を装着した場合
CJ1W-EIP21(CJ2)	CJ2 シリーズ PLC に形 CJ1W-EIP21 を装着した場合
CJ1W-EIP21S(NJ)	NJ シリーズ CPU ユニットに、形 CJ1W-EIP21S を装着した場合
CJ1W-EIP21S	CJ1 シリーズ PLC に形 CJ1W-EIP21S を装着した場合
CJ1W-EIP21S(CJ2)	CJ2 シリーズ PLC に形 CJ1W-EIP21S を装着した場合
CJ2B-EIP21	CJ2H 内蔵の EtherNet/IP ポート
CJ2M-EIP21	CJ2M 内蔵の EtherNet/IP ポート
CS1W-EIP21	CS1 シリーズ PLC に形 CS1W-EIP21 を装着した場合
CS1W-EIP21S	CS1 シリーズ PLC に形 CS1W-EIP21S を装着した場合
NJ101	NJ シリーズ CPU ユニット内蔵の EtherNet/IP ポート
NJ301- □□□□	
NJ501- □□□□	
NX701	NX シリーズ CPU ユニット内蔵の EtherNet/IP ポート

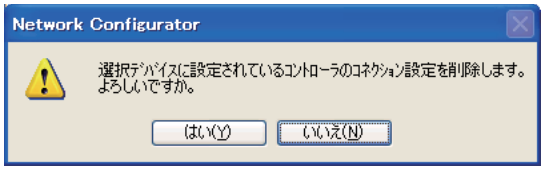
ウィザードを使用したコネクション（タグデータリンク）の作成方法は次の通りです。

1　**ウィザードを起動する前に全てのデバイスにタグ／タグセットを設定します。設定方法については、「7-2-4 タグ／タグセットの作成」を参照してください。**

2　**オムロン製 PLC 間のタグデータリンクでは、データを入力として受信する側の PLC （オリジネータデバイス）にコネクションを作成していきます。**
まず、Network Configurator のネットワーク構成画面でコネクションを作成したい登録デバイスを選択し、［デバイス］｜［パラメータ］｜［ウィザード］メニューを実行します。

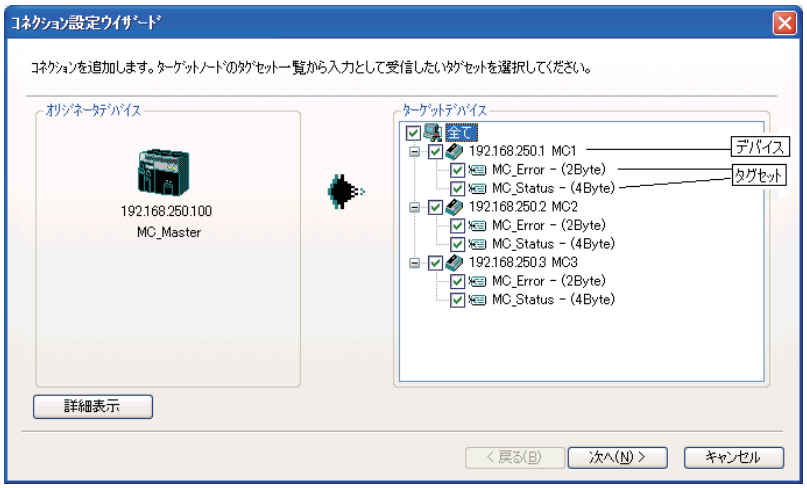


ウィザードが起動する前に次のダイアログが表示されます。



［はい］を選択すると、設定済みのオムロン製 PLC とのコネクションを削除してからウィザードが起動します。

3 **起動したウィザード画面（下図）のガイドに従い、コネクションを作成します。**



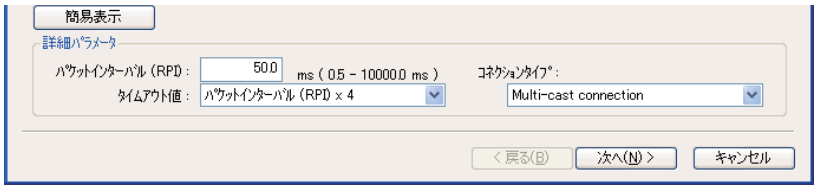
4 **ウィザード右側に、入力として受信可能なターゲットデバイスのタグセット一覧が表示されます。**

オリジネータデバイスで受信したいタグセットにチェックを入れます。
タグセット一覧で表示されるアイコン／チェックの意味は次のとおりです。

アイコン	表示位置	状態
☑	全て	全てのデバイスの全ての出力タグセットを選択中
	デバイス	対象デバイスの全ての出力タグセットを選択中
	タグセット	対象出力タグセットを選択中。本タグセットがコネクション設定対象となる。
☑	全て	一部のデバイスの全てあるいは一部の出力タグセットを選択中。
	デバイス	対象デバイスの一部の出力タグセットを選択中。
☐	全て	全てのデバイスの全ての出力タグセットを未選択中
	デバイス	対象デバイスの全ての出力タグセットを未選択中
	タグセット	対象出力タグセットを未選択中。本タグセットがコネクション削除対象となる
☐	デバイス	対象タグセット無し

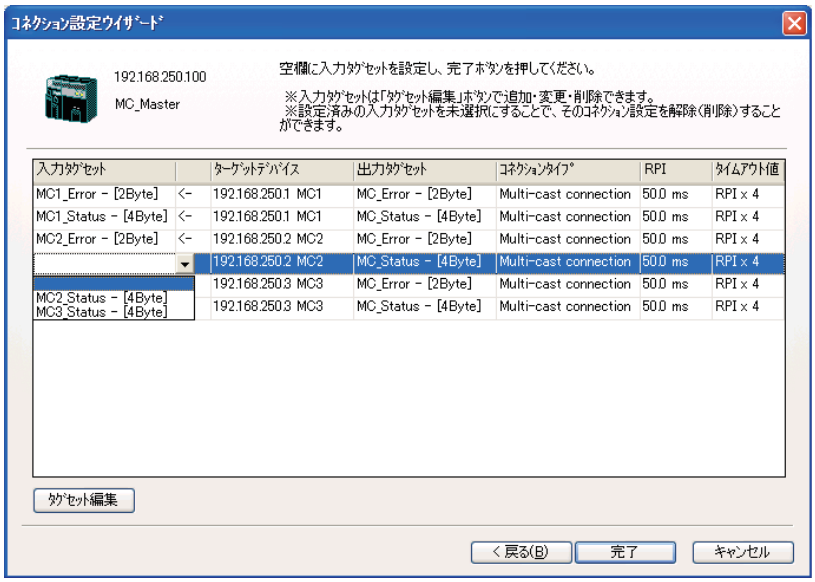
(注)設定済みのコネクションで使用されているタグセットは表示されません。

［詳細表示］ボタンをクリックすると下図のようになります。



詳細パラメータの規定値が表示されますので必要に応じて変更します。
コネクション名は設定できません。以下の規則で自動生成します。
default_N （※ N は 1 ～、3 桁表記（001，002，…））

5 **［次へ］ボタンを選択すると、表形式の以下のウィザード画面に切り替わります。ガイドに従って、ターゲットデバイスの出力タグセットを受信するオリジネータデバイスの入力タグセットを、リストボックスから選択して入力します。**



- 入力タグセットが空欄の行が新規作成しようとしているコネクションです。
- 入力タグセットが設定されている行は、既に設定済みのコネクションです。
- 使用済みの入力タグセットは、重複指定を防止するため、入力タグセットのリストボックスに表示されません。
- 適当な入力タグセットがない場合、［タグセット編集］［タグ編集］ボタンを使って、編集もしくは新規作成することができます。

6 **入力タグセットの設定が終わったら、［完了］ボタンをクリックします。設定したコネクションは［ネットワーク］｜［デバイスのコネクション構成表示］メニューで確認することができます。**

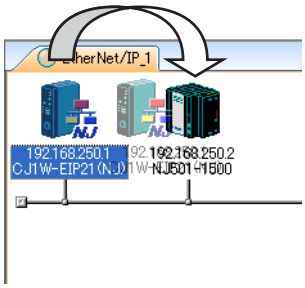
- 入力タグセットが空欄の行を含んでいても、ウィザードを完了することができます。この場合、空欄の行に対しては、コネクションが作成されません。
- 既に設定されていた入力タグセットを空欄にすることで、そのコネクションを解除（削除）することができます。

7-2-7　デバイスのドラッグ & ドロップによるコネクションの作成

ターゲットデバイスをドラッグし、オリジネータデバイスにドロップすることで、オリジネータへのコネクション作成を行うことができます。

例)192.168.250.1 のデバイス（ターゲット）をドラッグし、192.168.250.100 のデバイス（オリジネータ）にドロップする。

ドラッグ&ドロップ



参考

オリジネータとなる（コネクションを設定できる）EtherNet/IP デバイスは、以下のオムロン製 EtherNet/IP デバイスに限ります。

デバイス名	補足
CJ1W-EIP21(NJ)	NJ シリーズコントローラに、形 CJ1W-EIP21 を装着した場合
CJ1W-EIP21	CJ1 シリーズ PLC に形 CJ1W-EIP21 を装着した場合
CJ1W-EIP21(CJ2)	CJ2 シリーズ PLC に形 CJ1W-EIP21 を装着した場合
CJ1W-EIP21S(NJ)	NJ シリーズ CPU ユニットに、形 CJ1W-EIP21S を装着した場合
CJ1W-EIP21S	CJ1 シリーズ PLC に形 CJ1W-EIP21S を装着した場合
CJ1W-EIP21S(CJ2)	CJ2 シリーズ PLC に形 CJ1W-EIP21S を装着した場合
CJ2B-EIP21	CJ2H 内蔵の EtherNet/IP ポート
CJ2M-EIP21	CJ2M 内蔵の EtherNet/IP ポート
CS1W-EIP21	CS1 シリーズ PLC に形 CS1W-EIP21 を装着した場合
CS1W-EIP21S	CS1 シリーズ PLC に形 CS1W-EIP21S を装着した場合
NJ101 NJ301-□□□□ NJ501-□□□□	NJ シリーズ CPU ユニット内蔵の EtherNet/IP ポート
NX701	NX シリーズ CPU ユニット内蔵の EtherNet/IP ポート

ドラッグ&ドロップによるコネクション (タグデータリンク) の作成方法は次のとおりです。

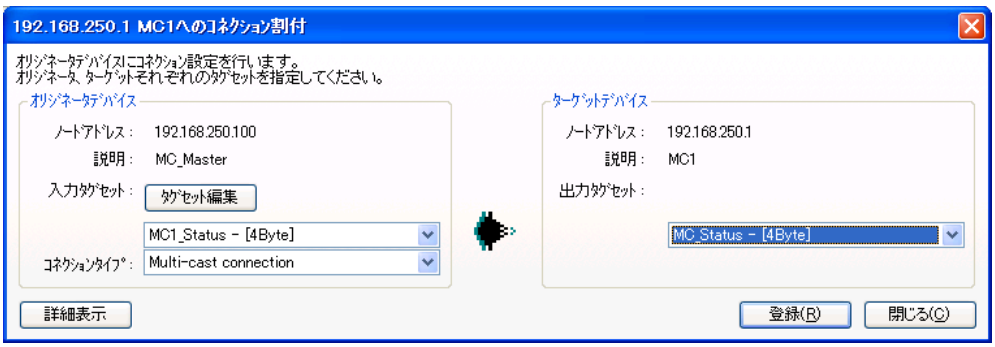
1　ターゲット（ドラッグする）デバイスに、タグ／タグセットを設定します。

(1)　ターゲットが上記オムロン製 EtherNet/IP デバイスの場合、設定方法は　「7-2-4 タグ／タグセットの作成」を参照してください。

(2)　ターゲットがその他の EtherNet/IP デバイスの場合、該当デバイスのマニュアルをご確認の上、必要に応じて設定してください。

2　ターゲットデバイスをドラッグし、オムロン製 EtherNet/IP デバイスにドロップすると下図のような、コネクション割付け用ダイアログボックスが表示されます。

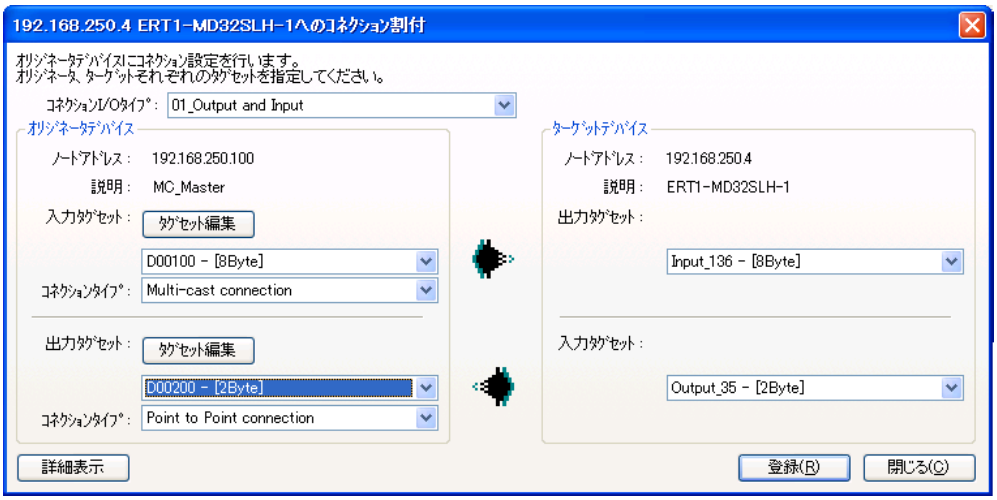
(1)　ターゲットが上記オムロン製 EtherNet/IP デバイスの場合



「コネクション設定」ダイアログボックスの右側、［ターゲットデバイス］から出力タグセットを選択し、続いて左側の「オリジネータデバイス」にて、これを受信する入力タグセットを選択します。

- 適当なオリジネータ側の入力タグセットがない場合、［タグセット編集］［タグ編集］ボタンを使って新規作成することができます。

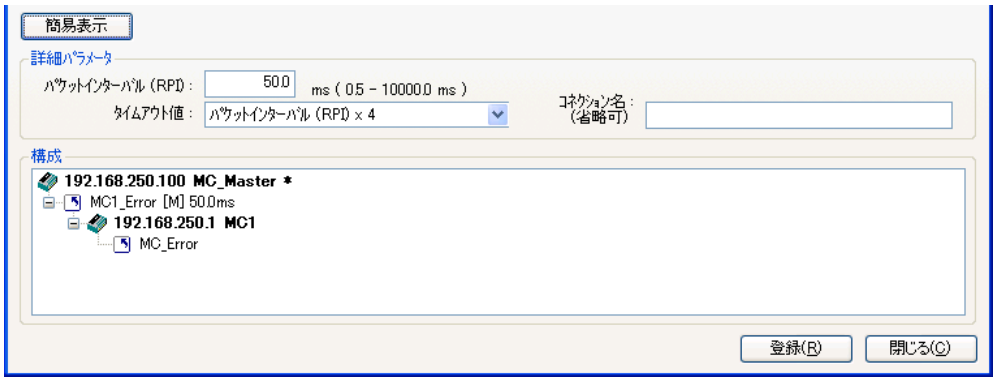
(2)　ターゲットがその他の EtherNet/IP デバイスの場合



「コネクション設定」ダイアログボックス上段にある［コネクション I/O タイプ］リストボックスに、選択可能なコネクション I/O タイプが表示されます。用途に応じたものを選択します。

- 選択可能なコネクション I/O タイプは、ターゲットデバイスで決められます。
- 選択したコネクション I/O タイプに応じて、設定可能な項目が有効になります。
- ターゲット側の出力（もしくは入力、両方の）タグセットを選択し、これに対応するオリジネータ側の入力（もしくは出力、両方の）タグセットを指定します。
- 適当なオリジネータ側のタグセットが存在しない場合、［タグセット編集］［タグ編集］ボタンを使って新規作成することができます。

［詳細表示］ボタンをクリックすると下図のようになります。



詳細パラメータの規定値が表示されますので必要に応じて変更します。
コネクション名は、以下の規則で自動生成します。
default_N （※ N は 1 ～、3 桁表記（001，002，…））

参考

I/Oデータが存在しないターゲットデバイスをドロップした場合、次のダイアログが表示されます。



該当デバイスのマニュアルなどを参考にコネクション作成に必要な I/O データ（出力タグセット）を作成した上で、再度ドロップしてください。

3　設定が完了しましたら［登録］ボタンをクリックしコネクションを作成します。作成が完了すると入力タグセット・出力タグセットが空欄になります。
続けてコネクション I/O タイプを選択しタグセットを設定することで、連続してコネクションの作成を行うことができます。

7-2-8　Network Configurator のネットワーク接続手順

Network Configurator のネットワーク接続手順を説明します。

Ethernet による接続方法

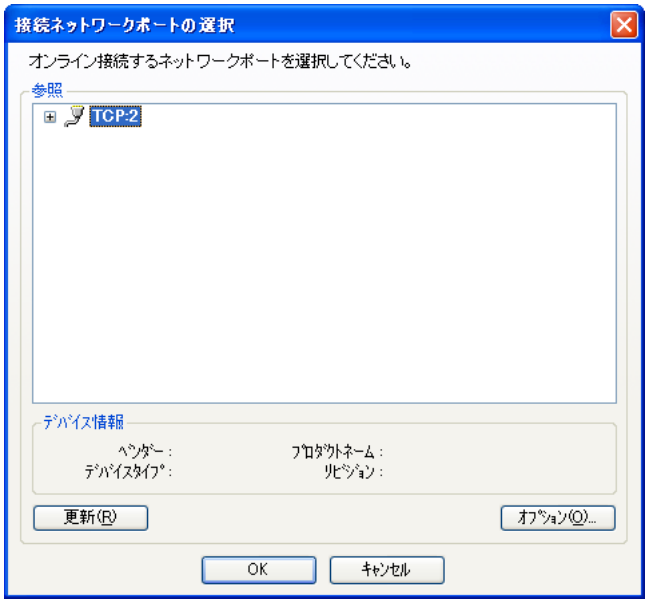
使用上の注意

WindowsXP(SP2以降)/WindowsVista/Windows7で初めて本接続を行う場合は、Windowsファイアウォールの設定の変更が必要です。
変更方法の詳細は、「A-4 Network Configurator を、WindowsXP/WindowsVista/Windows7 で使用する際の留意点」を参照してください。

Ethernet 経由で EtherNet/IP ユニットの Ethernet ポートへ接続します。

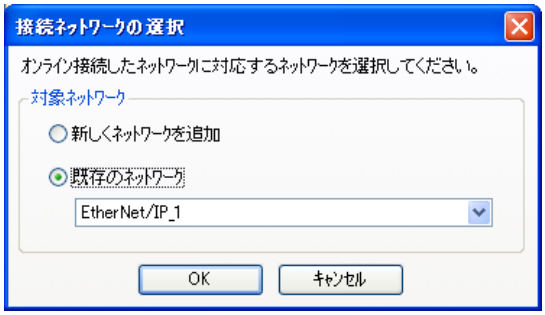
1　**［オプション］｜［インターフェースの選択］｜［Ethernet I/F］を選択します。**

2　**［ネットワーク］｜［接続］を選択します。**
コンピュータ上に Ethernet I/F が複数ある場合は、［インターフェースの選択］ダイアログボックスが表示されますので、接続するインタフェースを選び［OK］ボタンを押します。
以下のダイアログボックスが表示されます。

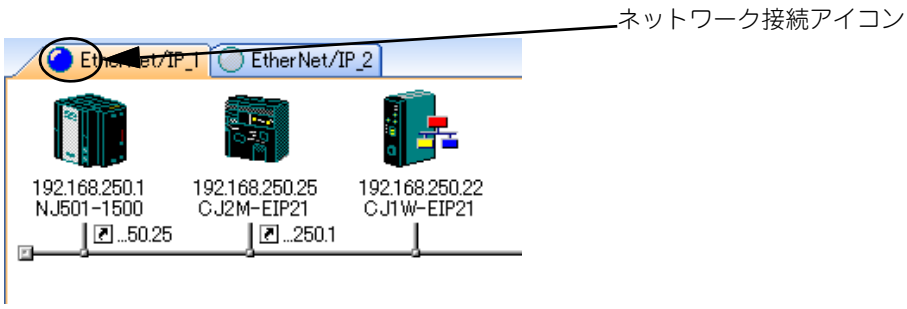


3 **[OK] ボタンをクリックします。**

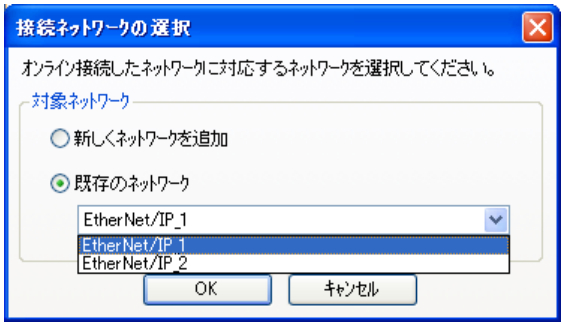
接続するネットワークを選択します。



Network Configurator が EtherNet/IP ネットワークに接続されます。正常にオンライン接続されると、画面下部のステータスバーに「On-line」と表示されます。
また、Network Configurator が接続しているネットワークタブのネットワーク接続アイコンが青色に表示されます。



接続するネットワークを切り替えるには、［ネットワーク］｜［接続ネットワークの変更］で切り替えます。

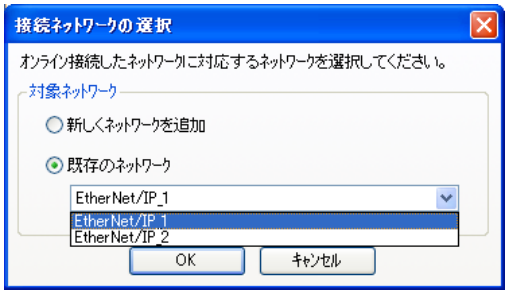


4 **以下のダイアログボックスが表示されます。**



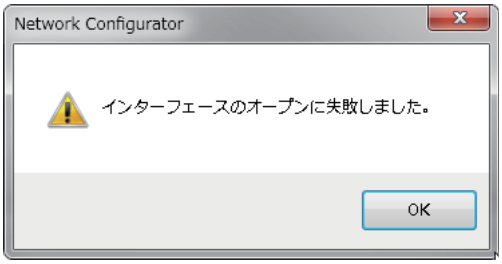
5 **[OK] ボタンをクリックします。**

接続するネットワークを選択します。



参考

NJ/NXシリーズコントローラと接続時にNetwork Configurator上に次のダイアログが表示された場合、下表の推定要因が考えられますので処置対策してください。



推定要因	処置対策
ケーブルが正しく接続されていない	ケーブルの断線、ゆるみがないか確認してください。
ファイアウォールの設定により、コントローラとの接続がブロックされている	ファイアウォールの設定でコントローラとの接続がブロックされている場合は解除してください。 ファイアウォールの設定は 📖「A-4 Network Configurator を、WindowsXP/WindowsVista/Windows7 で使用する際の留意点」を参照してください。

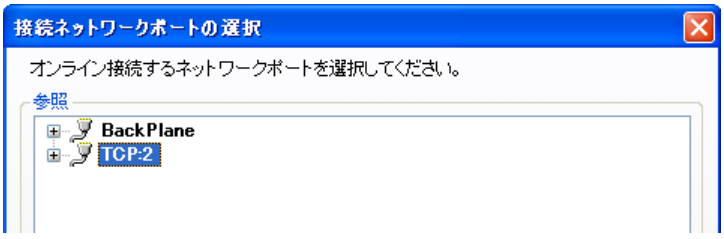
CPU ユニットの USB ポートによる接続方法（NJ CPU ユニット）

CPU ユニットの USB ポート経由で、EtherNet/IP ユニットへ接続します。

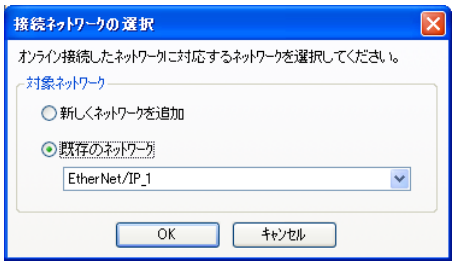
1 **通信インタフェースを選択します。**

［オプション］｜［インターフェースの選択］から、［NJ Series USB Port］を選択します。

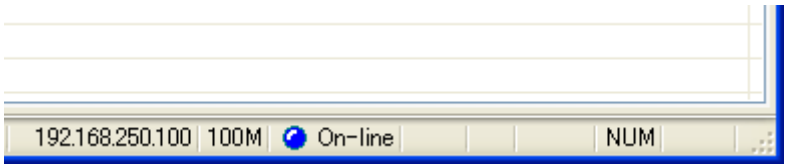
2 **「TCP：2」を選択し、[OK] ボタンをクリックします。**



3 **接続するネットワークを選択します。**

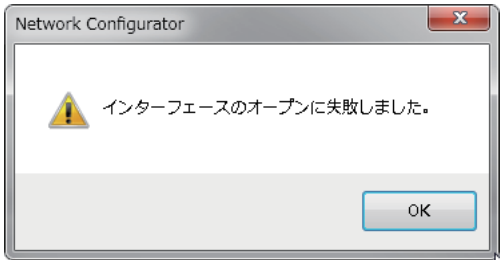


4 **Network Configurator が EtherNet/IP ネットワークに接続します。正常にオンライン接続されると、画面下部のステータスバーに「On-line」と表示します。**



参考

NJ/NXシリーズコントローラと接続時にNetwork Configurator上に次のダイアログが表示された場合、下表の推定要因が考えられますので処置対策してください。



推定要因	処置対策
ケーブルが正しく接続されていない	ケーブルの断線、ゆるみがないか確認してください。
ファイアウォールの設定により、コントローラとの接続がブロックされている	ファイアウォールの設定でコントローラとの接続がブロックされている場合は解除してください。 ファイアウォールの設定は 🕮「 A-4 Network Configurator を、WindowsXP/WindowsVista/Windows7 で使用する際の留意点 A-46」を参照してください。
USB ドライバが正しくインストールされていない	USB ドライバを正しくインストールしてください。 USB ドライバのインストール方法は、🕮『Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)』を参照してください。

7-2-9　タグデータリンクパラメータのダウンロード

タグデータリンクを行うには、タグセット設定や、コネクション設定などのタグデータリンクパラメータを、EtherNet/IP ネットワーク上のすべてのデバイスにダウンロードする必要があります。ダウンロードを実行すると、タグデータリンクパラメータは、設定対象となる EtherNet/IP デバイスへ転送されます。

以下に、タグデータリンクパラメータをダウンロードする手順を示します。
Network Configurator をネットワークへ接続する方法については 🕮「7-2-8 Network Configurator のネットワーク接続手順」を参照してください。

使用上の注意

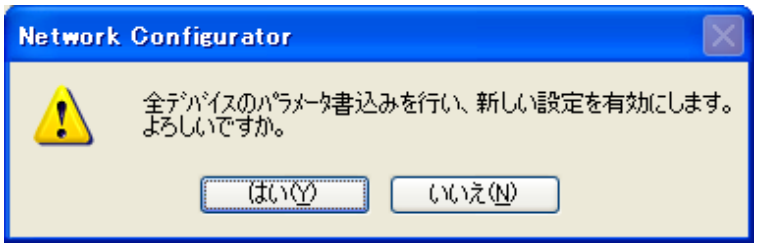
- 設定対象ノードアドレス（IP アドレス）が正しく設定されていないと、意図しないコントローラに接続して不正なデバイスパラメータを設定してしまうことがあるため、接続先コントローラを事前に確認してからダウンロードを実行してください。
- 不適切なタグデータリンクパラメータが設定されると、設備が予期しない動作をする恐れがあります。また適切なタグデータリンクパラメータが設定されている場合でも、設備に影響がないことを確認してから、転送を実行してください。
- タグの設定にネットワーク変数を用いる場合、これらの変数が CPU ユニットにも設定されていないと、コネクションがエラーとなります。CPU ユニットへネットワーク変数が設定されていることを確認してから、タグデータリンクパラメータのダウンロードをするようにしてください。「15-1-1 Network Configurator のデバイスモニタ機能」の［コネクション］タブ［タグステータス］タグにより、ネットワーク変数、タグ、コネクションの設定が正しく機能しているかを確認してください。
- 通信障害が発生した場合、出力状態は使用される製品の仕様によります。出力装備を伴う製品を使用される場合は、通信障害時にどのような動作仕様であるかを確認の上、安全対策を施してください。
- ダウンロードが完了すると、自動的に EtherNet/IP ユニットの通信ポートにリスタートがかかります。このリスタート動作は、設定したタグセットやコネクション情報を、有効にするために必要です。リスタートがかかっても設備に支障がないことを確認してから、ダウンロードを実行してください。
- Network Configurator で登録したデバイスと使用する EtherNet/IP ユニットの CIP のメジャー・リビジョンは同じにしてください。CIP のメジャー・リビジョンが異なると、ダウンロードされない場合があります。ダウンロードの可否は、🕮「7-2-3 デバイスの登録」を参照してください。
- ダウンロード中に Ethernet ケーブルを抜いたり、EtherNet/IP ユニットのリセットや電源 OFF をしないでください。
- ダウンロード中は、関連するノード間でのタグデータリンク（データ交換）を停止します。運転モードのままダウンロードする場合は、設備に影響がないことを確認してから転送してください。
  また、タグデータリンクを使用するラダープログラムでは、タグデータリンク停止中、およびタグデータリンク異常中のデータ処理をインターロックするなどしてください。
- EtherNet/IP ユニットが Rev.2 以降では、CPU ユニットが運転モードのままタグデータリンクパラメータをダウンロードすることが可能です（プログラムモードに変更してから転送することも可能です）。
  ダウンロード中は、関連するノード間でのタグデータリンク（データ交換）を停止します。運転モードのままダウンロードする場合は、設備に影響がないことを確認してから転送してください。
  また、タグデータリンクを使用するラダープログラムでは、タグデータリンク停止中、およびタグデータリンク異常中のデータ処理をインターロックするなどしてください。
- EtherNet/IP ユニット Rev.1 では、タグデータリンクパラメータをダウンロードするには、必ず CPU ユニットをプログラムモードにして行う必要があります。

・Rev.2 以降の場合でも、ネットワーク中に Rev.1 のユニットが混在する場合は、必ずすべての CPU ユニットをプログラムモードにして行う必要があります。

1　Network Configurator をネットワークへ接続します。

2　ダウンロードには 2 通りの方法があります。

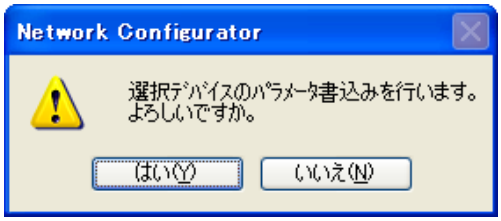
・ネットワークを一括してダウンロードする場合
［ネットワーク］メニューから［ダウンロード］を選択します。
次のダイアログボックスが表示されます。



・特定のデバイスへ個別にダウンロードする場合
ダウンロードしたい EtherNet/IP ユニットのアイコンをマウスでクリックして選択します。複数ノードを選択するときは、あらかじめ［Shift］キーまたは［Ctrl］キーを押しながらアイコンをマウスでクリックします。(以下の例では、192.168.250.1 と 192.168.250.2 の 2 ノードを選択しています。)
選択している状態から、マウスの右ボタンクリックで表示されるメニューから［パラメータ］｜［ダウンロード］を選択しクリックします。

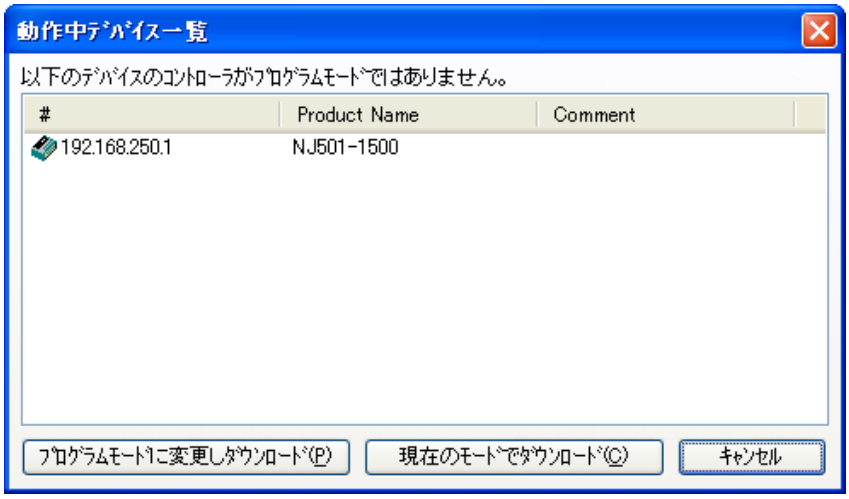


次のダイアログボックスが表示されます。



3 **［はい］ボタンをクリックし、EtherNet/IP デバイスに対しタグデータリンクパラメータ設定をダウンロードします。**

すべての CPU ユニットが、プログラムモードでない場合は以下のようなウィンドウが表示されます。

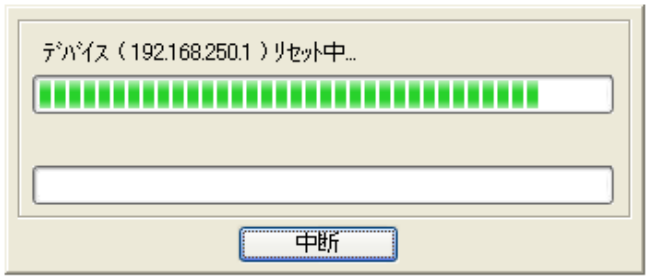


「プログラムモードに変更しダウンロード」をクリックした場合には、すべての CPU ユニットをいったんプログラムモードへ変更しダウンロードを行います。この場合は、制御装置をプログラムモードに変更しても影響がないことを確認してください。ダウンロード終了後にモードを元に戻すことができます。

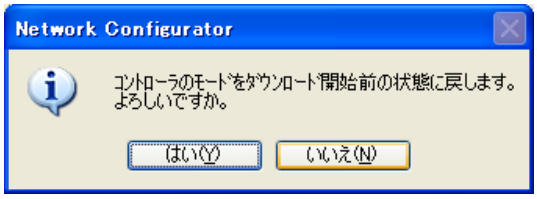
「現在のモードでダウンロード」をクリックした場合には、CPU ユニットが運転モードでもダウンロードが可能です。

「現在のモードでダウンロード」に対応していない EtherNet/IP ユニットの機種（CJ1W-EIP21/ CS1W-EIP21 の Rev.1 など）の場合、「現在のモードでダウンロード」ボタンは有効になりません。

ダウンロード中は、以下のようにインジケータが表示され、ダウンロード経過が分かります。



CPU ユニットのモードを変更してダウンロードした場合には、以下のように元に戻すか指定します。「いいえ」を選択した場合は、プログラムモードのままとなります。



4 **以下のダイアログボックスが表示され、ダウンロードが完了します。**



7-2-10　タグデータリンクパラメータのアップロード

EtherNet/IP ネットワーク上の EtherNet/IP ユニットからタグセット情報、コネクション情報などのタグデータリンクパラメータをアップロードすることができます。
以下にアップロードする手順を示します。なお、Network Configurator をネットワークへ接続する方法については 🕮「7-2-8 Network Configurator のネットワーク接続手順」を参照してください。

使用上の注意

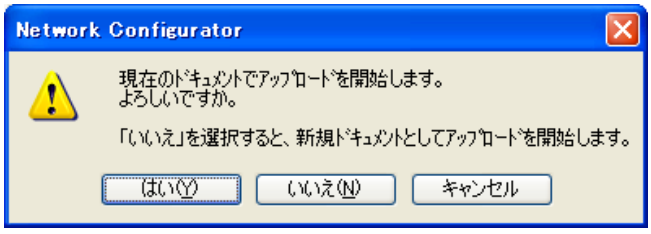
- Network Configurator で登録したデバイスと使用する EtherNet/IP ユニットの CIP のメジャー・リビジョンは同じにしてください。CIP のメジャー・リビジョンが異なると、アップロードされない場合があります。アップロードの可否は、🕮「7-2-3 デバイスの登録」を参照してください。

アップロードには 2 通りの方法があります。

- ネットワーク内の全デバイスのパラメータを一括してアップロードする場合

1 **Network Configurator をネットワークへ接続後、［ネットワーク］メニューから［アップロード］を選択します。**

2 **次のダイアログボックスが表示されます。**

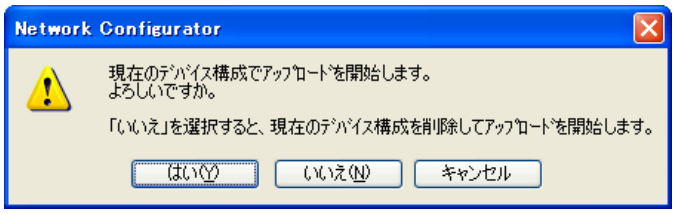


- ［はい］を選択した場合
  現在のプロジェクトでタグデータリンクパラメータをアップロードします。

- ［いいえ］を選択した場合
  新規プロジェクトを開いて、タグデータリンクパラメータをアップロードします。現在の開いているプロジェクトは閉じられます。

- ［キャンセル］ボタンをクリックした場合
  アップロードをキャンセルします。アップロードは行いません。

3　**手順 2 で、［はい］を選択した場合、次のダイアログボックスが表示されます。**



・［はい］を選択した場合
ネットワーク構成ウィンドウに登録されているデバイスからのみアップロードを行い、登録されていないデバイスからはアップロードを行いません。

・［いいえ］を選択した場合
ネットワーク一括アップロードのとき
新規にネットワーク上のすべてのデバイスからアップロードを行います。
現在のネットワーク構成情報は失われてしまいます。
以下のダイアログボックスが表示されますので、アップロードするデバイスを選択して［OK］ボタンをクリックしてください。



・［キャンセル］ボタンをクリックした場合
アップロードをキャンセルします。アップロードは行いません。

4 **手順 2 で、［いいえ］を選択した場合、次のダイアログボックスが表示されます。**
アップロードするデバイスを選択して［OK］ボタンをクリックしてください。

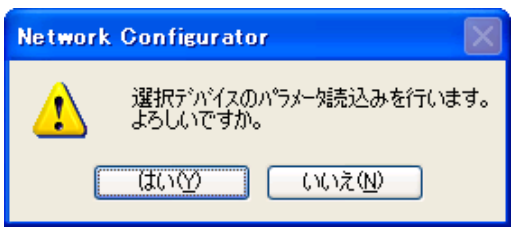


特定のデバイスから個別にアップロードする場合

1 **Network Configurator をネットワークへ接続後、アップロードしたい。**
マウスでクリックして選択します。複数ノードを選択するときは、あらかじめ［Shift］キーまたは［Ctrl］キーを押しながらアイコンをマウスでクリックします。（以下の例では、192.168.250.1 と 192.168.250.2 の 2 ノードを選択しています。）
選択している状態から、マウスの右ボタンクリックで表示されるメニューから［パラメータ］|［アップロード］を選択しクリックします。

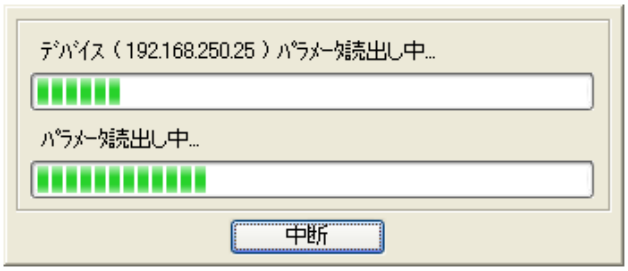


2 **次のダイアログボックスが表示されます。**



［はい］ボタンまたは［いいえ］ボタンをクリックします。

3 **アップロード中は、以下のようにインジケータが表示され、アップロード経過が分かります。**



4 **次のダイアログボックスが表示され、アップロードは完了します。**



7-2-11　タグデータリンクパラメータの照合

EtherNet/IP ネットワーク上の EtherNet/IP ユニットとタグセット情報、コネクション情報などのタグデータリンクパラメータ情報を照合することができます。
以下に照合する手順を示します。なお、Network Configurator をネットワークへ接続する方法については 🕮「7-2-8 Network Configurator のネットワーク接続手順」を参照してください。

使用上の注意

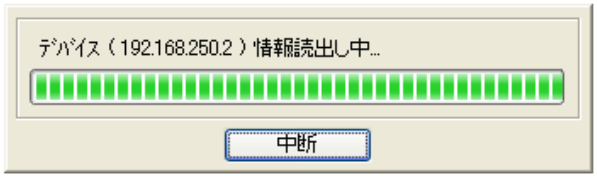
Network Configurator で登録したデバイスと使用する EtherNet/IP ユニットの CIP のメジャー・リビジョンは同じにしてください。CIP のメジャー・リビジョンが異なると、照合されない場合があります。照合の可否は、🕮「7-2-3 デバイスの登録」を参照してください。

ネットワーク構成の照合方法

ネットワーク構成ウィンドウに登録したデバイス一覧を EtherNet/IP ネットワーク上に接続されたデバイスと照合し、IP アドレスとデバイス種別を確認します。
本機能では、デバイスパラメータの照合は行われません。

1 **Network Configurator をネットワークへ接続します。**

2 **［ネットワーク］メニューから［構成の照合］を選択します。**
次のインジケータが表示され、ネットワークから情報を読み出し、照合します。



3 **ネットワーク構成ファイルとネットワークからの情報の照合結果によって、以下の表示となります。**

•照合結果に違いがなかった場合



•照合結果に違いがあった場合



•デバイス種別に違いがあった場合



[OK] ボタンもしくは［閉じる］ボタンをクリックします。

デバイスパラメータの照合方法

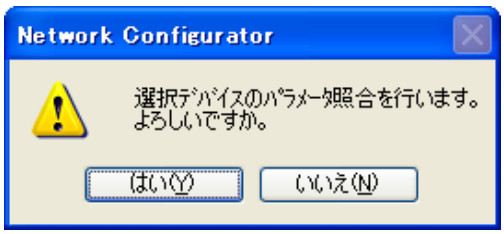
ネットワーク構成ウィンドウ上で選択されたデバイスと、EtherNet/IP ネットワーク上のデバイスとの間でデバイスパラメータを照合します。
ネットワーク構成の照合では、IP アドレスとデバイス種別、およびデバイスパラメータが比較されます。

1　**Network Configurator をネットワークへ接続します。**

2　**照合したい EtherNet/IP ユニットのアイコンをマウスでクリックして選択します。複数ノードを選択するときは、あらかじめ［Shift］キーまたは［Ctrl］キーを押しながらアイコンをマウスでクリックします。（以下の例では、192.168.250.1 と 192.168.250.2 の 2 ノードを選択しています。）**
選択している状態から、マウスの右ボタンクリックで表示されるメニューから［パラメータ］｜［照合］を選択しクリックします。



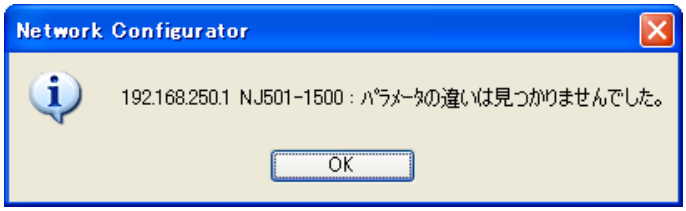
3　**次のダイアログが表示されます。**



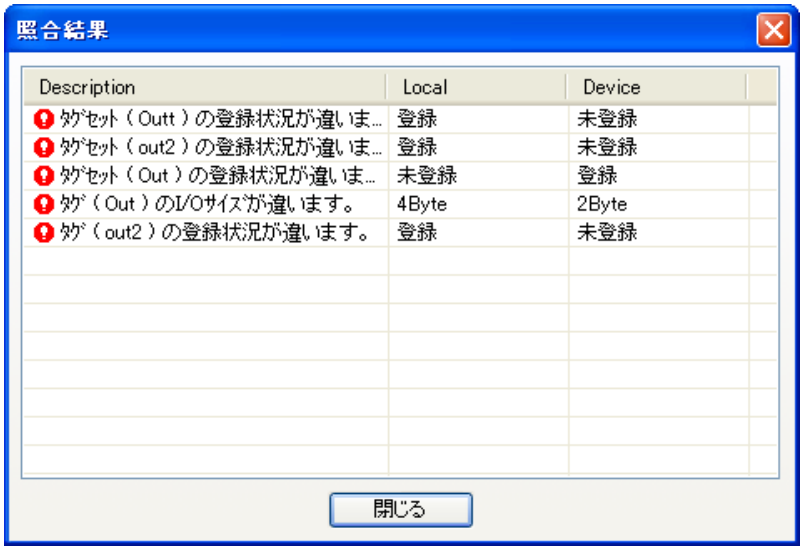
［はい］ボタン、または［いいえ］をクリックします。

4 **次のダイアログが表示されます。**

•照合結果に違いがなかった場合



•照合結果に違いがあった場合

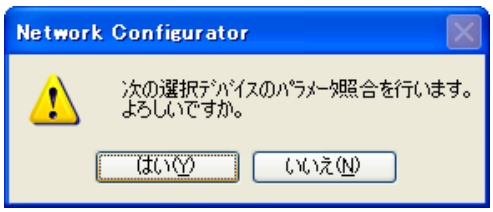


•デバイス種別に違いがあった場合



［OK］ボタンもしくは［閉じる］ボタンをクリックします。

5 **複数ノードを選択して照合していたときは、以下のメッセージが表示されるので［はい］ボタンをクリックします。**



選択したノード順に照合結果が表示されます。

7-2-12　タグデータリンクの開始／停止

タグデータリンクの自動起動

タグデータリンクは、Network Configurator でタグデータリンクパラメータをダウンロード直後にユニットが自動的にリスタートし起動します。
（コントローラの電源投入後、またはユニットリスタート後にも自動的に起動します）

ネットワーク全体のタグデータリンクの開始／停止

タグデータリンクの開始／停止は、ユーザプログラムからの操作と、Network Configurator からの操作が可能です。

使用上の注意

タグデータリンクの開始操作と停止操作は、同じ操作方法（ユーザプログラム、Network Configurator）で実行してください。
例えば、タグデータリンクを、CJ ユニットデバイス変数の「*_TDLStopCmd」（タグデータリンク停止スイッチ）で停止した場合、Network Configurator の操作では、タグデータリンクは再開しません。

● ユーザプログラムからの操作

ユーザプログラム上で次の CJ ユニットデバイス変数を FALSE → TRUE にすると、デバイス単位でタグデータリンクを開始／停止できます。（🕮「第 3 章 CJ ユニットデバイス変数への割付」を参照）

- 「*_TDLStartCmd」（タグデータリンク開始スイッチ）
- 「*_TDLStopCmd」（タグデータリンク停止スイッチ）

参考

- タグデータリンク通信開始スイッチは、タグデータリンク通信停止スイッチが FALSE のときに TRUE にしてください。
  タグデータリンク通信停止スイッチが TRUE のときに、タグデータリンク通信開始スイッチを TRUE にしても、タグデータリンクは開始されません。
  また、タグデータリンク通信開始／停止スイッチの両方が TRUE になると、エラーとなり、CJ ユニットデバイス変数の「複数スイッチ同時 ON 異常」が TRUE になり、イベントログに記録されます。
- タグデータリンク開始時、タグデータリンク通信開始スイッチが、自動的に FALSE に戻る前に、ユーザプログラムや Sysmac Studio などから、強制的に操作しないでください。

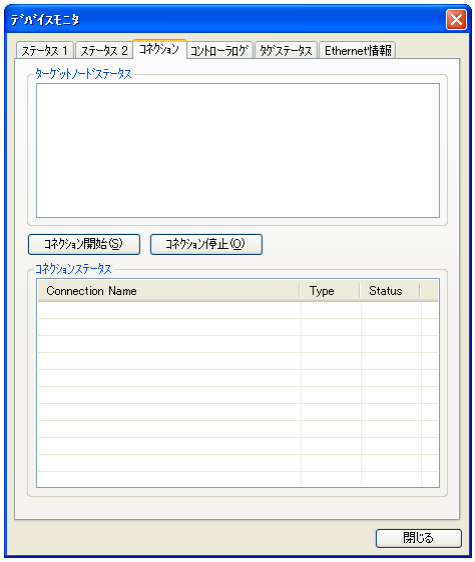
● Network Configurator からの操作

［ネットワーク］メニューから［I/O コネクション］｜［開始］／［停止］を選択すると、デバイス単位でタグデータリンクを開始／停止できます。

デバイス単位のタグデータリンクの開始／停止

● Network Configurator からの操作

［デバイス］メニューから［モニタ］を選択して、［デバイスモニタ］ダイアログボックスの［コネクション］タブにて、以下の操作をすると、デバイス単位（オリジネータ側のみ）でタグデータリンクを開始／停止できます。



［コネクション開始］ボタン：
自デバイスをオリジネータとしたすべてのコネクションを開始
［コネクション停止］ボタン：
自デバイスをオリジネータとしたすべてのコネクションを停止

7-2-13　デバイスパラメータのクリア

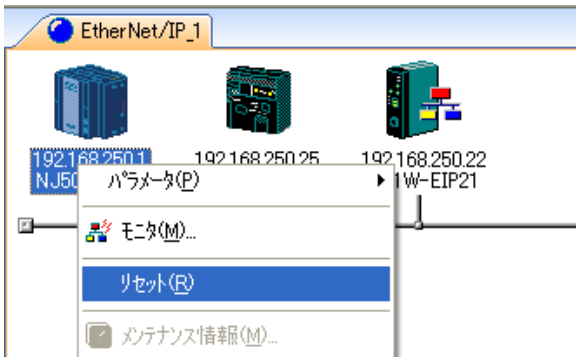
EtherNet/IP ネットワーク上の EtherNet/IP ユニットに保存されているタグデータリンクパラメータをクリアする（タグデータリンクのパラメータを工場出荷状態に戻す）ことができます。
以下にタグデータリンクパラメータをクリアする手順を示します。なお、Network Configurator をネットワークへ接続する方法については 🕮「7-2-8 Network Configurator のネットワーク接続手順」を参照してください。

1　**Network Configurator をネットワークへ接続します。**

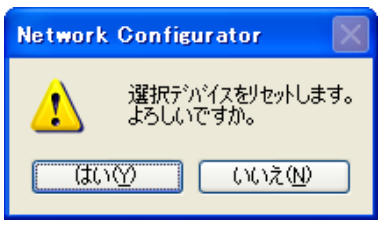
2　**デバイスパラメータをクリアしたい EtherNet/IP ユニットのアイコンをマウスでクリックして選択しておきます。**
以下の画面例では、192.168.250.1 と 192.168.250.2 の 2 ノードを選択しています。
複数ノードを選択するときは、[Shift] キーを押しながらアイコンをマウスでクリックします。



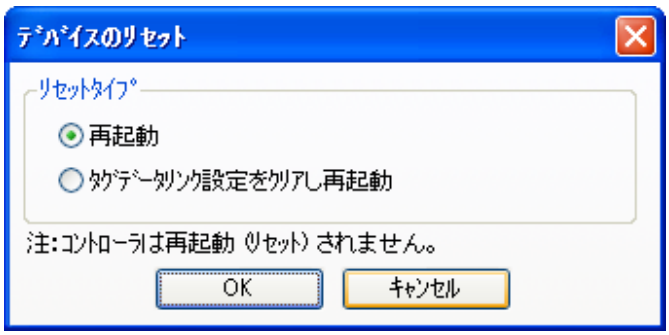
3　**［デバイス］メニューから［リセット］を選択します。**
右クリックし、ポップアップメニューから［リセット］をクリックして選択することもできます。



4　**次のダイアログボックスが表示されます。**



・［はい］を選択した場合
　次のダイアログボックスが表示されます。



［タグデータリンク設定をクリアし再起動］を選択し、［OK］ボタンをクリックします。

使用上の注意

コントローラ本体は再起動しません。EtherNet/IP ユニットのみ再起動します。

・［いいえ］ボタンをクリックした場合
　デバイスパラメータのクリアもリセットもされません。

7-2-14　ネットワーク構成ファイルの保存

Network Configurator で設定したデバイスパラメータや、ネットワークからアップロードしたデバイスパラメータをネットワーク構成ファイルとして保存することができます。

1　**［ファイル］メニューから［名前を付けて保存］を選択します。**
　次のダイアログボックスが表示されます。



デフォルトのファイル名として仮に Untitled と表示しています。

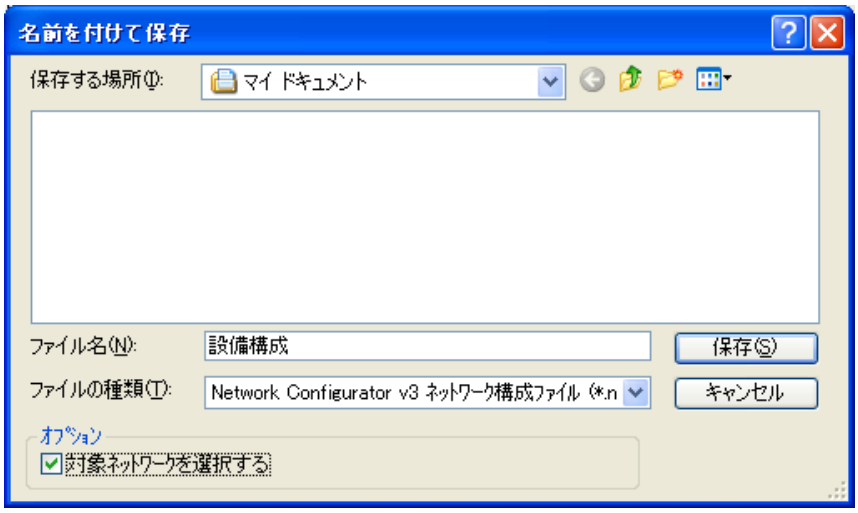
2　**ファイル名を入力し、[保存] ボタンをクリックします。**



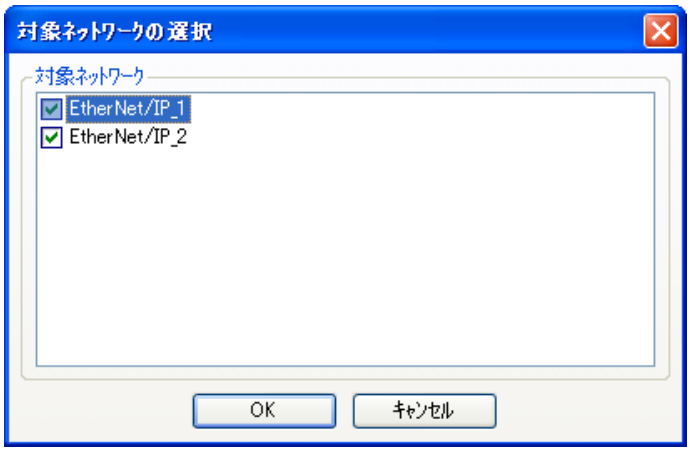
以上で、ネットワーク構成ファイルの保存が完了します。

3　**以降で、ネットワーク構成を変更した場合は、[ファイル] メニューから [上書き保存] か、 ボタンをクリックすることでネットワーク構成ファイルを上書き保存することができます。**

4　**「オプション」の「対象ネットワークを選択する」にチェックを入れて保存を選択すると、存在する複数のネットワーク構成から、必要な構成を選択して保存することが可能です。**



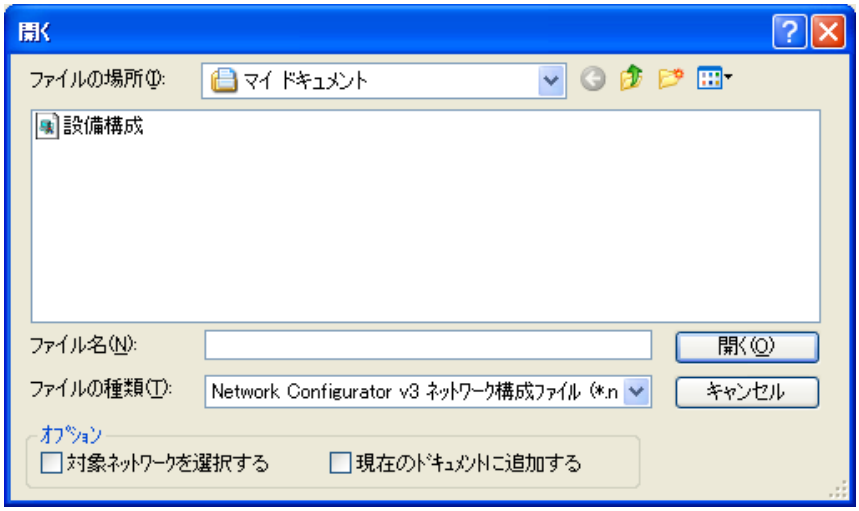
保存したいネットワークにのみチェックボックスをマークし〔OK〕をクリックします。



7-2-15　ネットワーク構成ファイルの読出

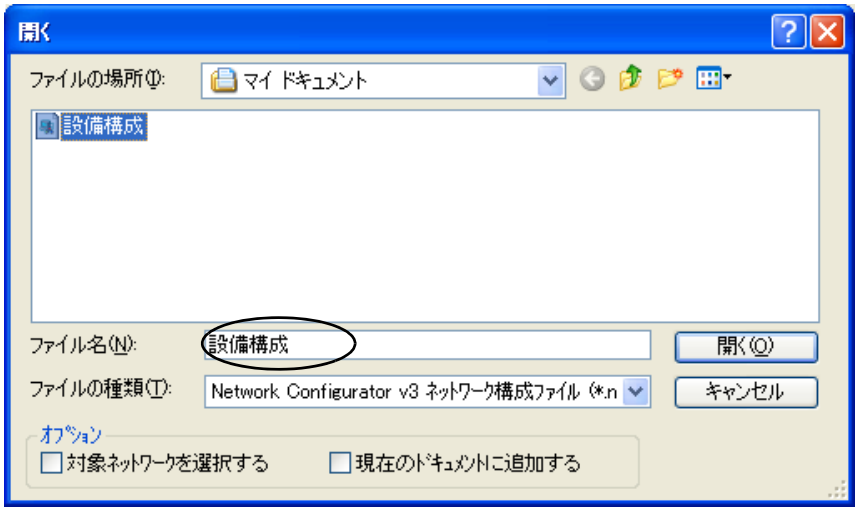
保存しておいたネットワーク構成ファイルを Network Configurator に読み出すことができます。

1　**［ファイル］メニューから［開く］を選択するか、ボタンをクリックします。**
次のダイアログボックスが表示されます。



読み出したいネットワーク構成ファイルが表示されていないときは、ファイルの場所を移動してください。

2 **読み出したいネットワーク構成ファイルをクリックすると、ファイル名に選択したファイルが表示します。**



3 **[開く] ボタンをクリックし、ネットワーク構成ファイルを読み出します。**

4 **Network Configurator のタイトルバーに、読み出したファイル名が表示されます。**



5 **「オプション」設定を利用する場合、各項目にチェックを入れてください。**
オプションの設定内容は以下のとおりです。

設定項目	内容
対象ネットワークを選択する	開こうとしているファイルに含まれる複数のネットワーク構成から、必要なものを選択して開くことができます。
現在のドキュメントに追加する	開こうとしているファイルに含まれるネットワーク構成を、今現在オープン中の構成ファイルに追加します。

参考

Network Configurator のバージョンによって、構成ファイルの保存形式が異なります。Network Configurator for EtherNet/IP（Ver.2 以前）に作成された構成ファイル（*.ncf）は、メニューの［ファイル］｜［外部データ］｜［インポート］からインポート（開く）することが可能です。

7-2-16　コネクションのチェック

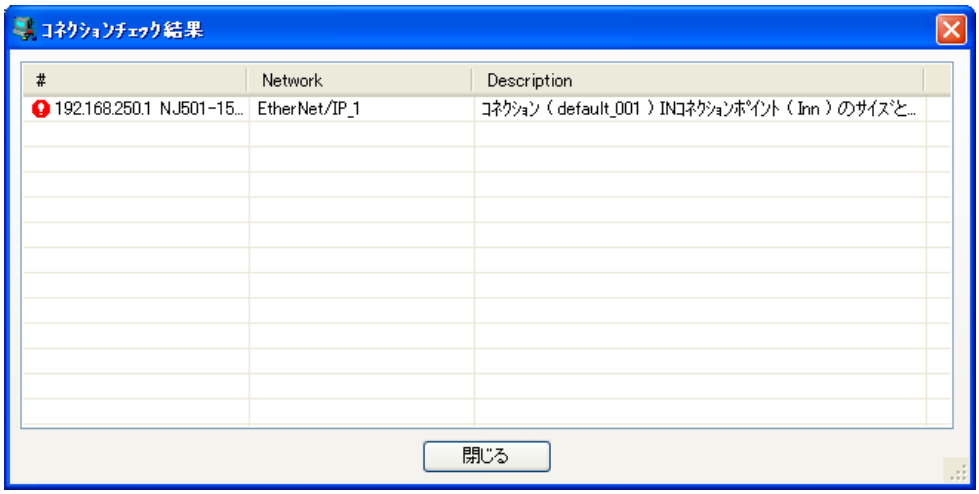
Network Configurator で設定したデバイスパラメータや、ネットワークからアップロードしたデバイスパラメータを、ネットワーク構成ファイルのコネクションパラメータとの整合性をチェックします。

1　**［ネットワーク］メニューから［コネクションのチェック］を選択します。**

パラメータが正常の場合は、次のダイアログボックスが表示されます。

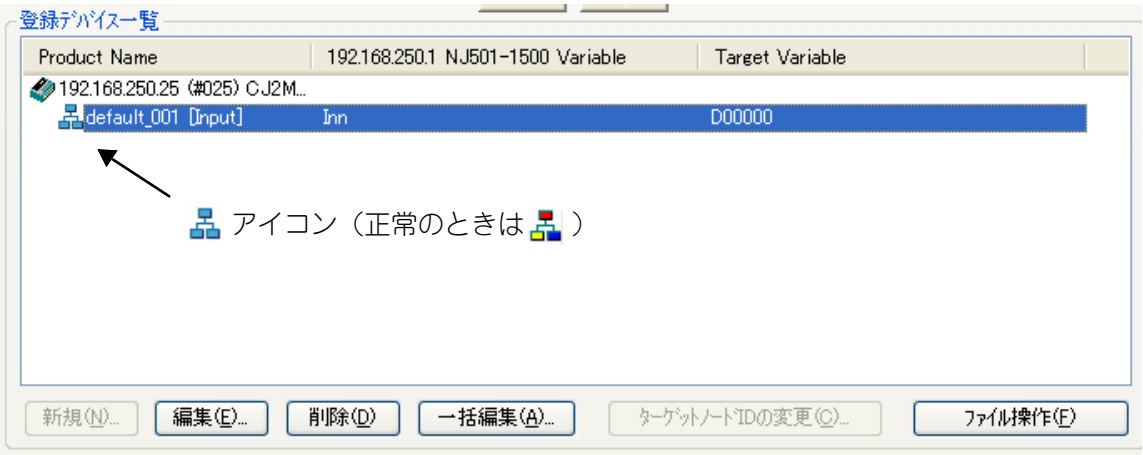


パラメータに異常が見つかった場合は、次のダイアログボックスが表示されますので、表示内容を参考に、設定を見直してください。



不一致が発生している場合、オリジネータデバイスの［デバイスパラメータの編集］ダイアログボックスを開き、［コネクション］タブを選択すると［登録デバイス一覧］のうち、パラメータ不一致のコネクションのアイコンが  となります（正常のときは  ）。

コネクション設定を変更し Target Variable を選択しなおす場合は、対象のコネクションを以下のように選択し、［編集］ボタンをクリックし修正します。



7-2-17　デバイスの変更

Network Configurator のネットワーク構成にいったん登録・設定したデバイスを後から変更することができます。
［デバイス］メニューから［デバイス変更］をクリックすることで、変更可能なデバイスのリストが示されるので、変更したいデバイスを選択することができます。
変更は、上位（もしくは完全）互換にあるデバイスへのみ可能になっているため、下位バージョンなどへの変更はできません。

デバイス変更方向

変更後ユニット形式		CS1W-EIP21	CJ1W-EIP21	CS1W-EIP21	CJ1W-EIP21	CJ1W-EIP21 (CJ2)	CJ2B-EIP21	CJ1W-EIP21 (NJ)	CJ2M	CS1W-EIP21	CJ1W-EIP21	CJ1W-EIP21 (CJ2)	CJ2B-EIP21	CJ1W-EIP21 (NJ)
変更前ユニット形式	Rev	1	1	2	2	2	2	2	2	3	3	3	3	3
CS1W-EIP21	1		○	○	○	○	○	○	△3	○	○	○	○	○
CJ1W-EIP21	1	○		○	○	○	○	○	△3	○	○	○	○	○
CS1W-EIP21	2	×	×		○	○	○	△5	△3	○	○	○	○	△5
CJ1W-EIP21	2	×	×	○		○	○	△5	△3	○	○	○	○	△5
CJ1W-EIP21 (CJ2)	2	×	×	△1	△1		○	△5	△3	△1	△1	○	○	△5
CJ2B-EIP21	2	×	×	△1	△1	○		△5	△3	△1	△1	○	○	△5
CJ1W-EIP21 (NJ)	2	×	×	△1 △2	△1 △2	△2	△2		△2 △6	△1 △2	△1 △2	△2	△2	○
CJ2M	2	×	×	△1	△1	○	○	△5		△1	△1	○	○	△5
CS1W-EIP21	3	×	×	○	○	○	○	△5	△3		○	○	○	△5
CJ1W-EIP21	3	×	×	○	○	○	○	△5	△3	○		○	○	△5
CJ1W-EIP21 (CJ2)	3	×	×	△1	△1	○	○	△5	△3	△1	△1		○	△5
CJ2B-EIP21	3	×	×	△1	△1	○	○	△5	△3	△1	△1	○		△5
CJ1W-EIP21 (NJ)	3	×	×	△1 △2	△1 △2	△2	△2	○	△2 △6	△1 △2	△1 △2	△2	△2	
CS1W-EIP21S	4	×	×	○	○	○	○	△5	△3	○	○	○	○	△5
CJ1W-EIP21S	4	×	×	○	○	○	○	△5	△3	○	○	○	○	△5
CJ1W-EIP21S (CJ2)	4	×	×	△1	△1	○	○	△5	△3	△1	△1	○	○	△5
CJ1W-EIP21S (NJ)	4	×	×	△1 △2	△1 △2	△2	△2	○	△2 △6	△1 △2	△1 △2	△2	△2	○
NJ501-□□□□ NJ301-□□□□ NJ101	1	×	×	△1 △2	△1 △2	△2	△2	○	△2 △6	△1 △2	△1 △2	△2	△2	○
	2	×	×	△1 △2	△1 △2	△2	△2	○	△2 △6	△1 △2	△1 △2	△2	△2	○
NX701	2	×	×	×	×	△2	△2	○	△2 △6	×	×	△2	△2	○
NX1P2	2	×	×	×	×	△2	△2	○	△2 △6	×	×	△2	△2	○
NX102	2	×	×	×	×	△2	△2	○	△2 △6	×	×	△2	△2	○
NY512/532	2	×	×	×	×	×	×	○	×	×	×	×	×	○
NX-CSG	2	×	×	×	×	×	×	×	×	×	×	×	×	×

変更後ユニット形式		CS1W-EIP21	CJ1W-EIP21	CS1W-EIP21	CJ1W-EIP21	CJ1W-EIP21 (CJ2)	CJ2B-EIP21	CJ1W-EIP21 (NJ)	CJ2M	CS1W-EIP21	CJ1W-EIP21	CJ1W-EIP21 (CJ2)	CJ2B-EIP21	CJ1W-EIP21 (NJ)
変更前ユニット形式	Rev	1	1	2	2	2	2	2	2	3	3	3	3	3
NX502	2	×	×	×	×	△2 △7	△2 △7	△2 △7	△2 △6 △7	×	×	△2 △7	△2 △7	△2 △7
NX-EIP201	2	×	×	×	×	△2 △6 △7	△2 △6 △7	△2 △6 △7	△2 △4 △7	×	×	△2 △6 △7	△2 △6 △7	△2 △4 △7

○：　変更できます。
×：　変更できません。
△1：タグにネットワーク変数が指定されている場合は変更できません。
△2：タグ名、タグセット名の最大サイズ（UTF-8 変換後サイズ）が、48 バイトを超えている場合は変更できません。
△3：次の項目が変更したデバイスの設定可能範囲を超えている場合は変更できません。
　　I/O コネクション数、タグ数、タグセット数、または 1 タグセットのサイズ
△4：次のいずれかの場合は変更できません。
　　• I/O コネクション数、タグ数、タグセット数、1 タグセットのサイズが変更したデバイスの設定可能範囲を超えている
　　• RPI が設定可能範囲を超えている、または 0.5ms 単位（例：10.5ms）で設定されている
△5：タグが物理アドレスで奇数サイズの場合は変更できません。
△6：タグ、タグセット、リフレッシュサイズなどが設定可能範囲を超えている場合は変更できません。
　　（リフレッシュサイズとは 1 ノードがデータ交換するサイズで複数あるタグセットの総和になり上限があります。）
△7：1 タグセットあたりのタグ最大数が設定可能範囲を超えている場合は変更できません。

（つづき）

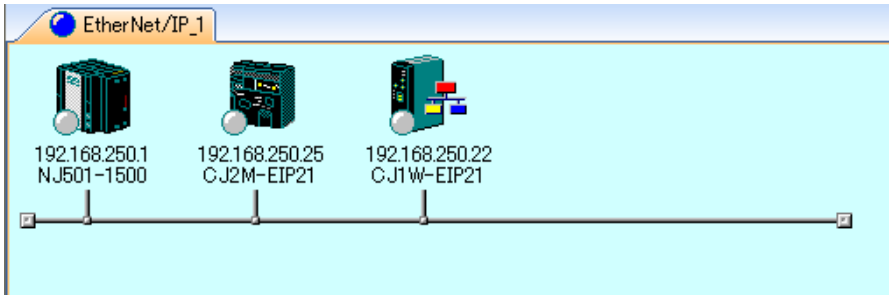
変更後ユニット形式		CS1W-EIP21S	CJ1W-EIP21S	CJ1W-EIP21S (CJ2)	CJ1W-EIP21S (NJ)	NJ501-□□□□ NJ301-□□□□ NJ101		NX701	NX1P2	NX102	NY512/532	NX-CSG	NX502	NX-EIP201
変更前ユニット形式	Rev	4	4	4	4	1	2	2	2	2	2	2	2	2
CS1W-EIP21	1	○	○	○	○	△4	△4	×	×	×	×	×	×	×
CJ1W-EIP21	1	○	○	○	○	△4	△4	×	×	×	×	×	×	×
CS1W-EIP21	2	○	○	○	△5	△4 △5	△4 △5	×	×	×	×	×	×	×
CJ1W-EIP21	2	○	○	○	△5	△4 △5	△4 △5	×	×	×	×	×	×	×
CJ1W-EIP21 (CJ2)	2	△1	△1	○	△5	△4 △5	△4 △5	△5	△4 △5	△4 △5	×	×	△4 △5	△5
CJ2B-EIP21	2	△1	△1	○	△5	△4 △5	△4 △5	△5	△4 △5	△4 △5	×	×	△4 △5	△5
CJ1W-EIP21 (NJ)	2	△1 △2	△1 △2	△2	○	△4	△4	○	△4	△4	△4	×	△4	○
CJ2M	2	△1	△1	○	△5	△4 △5	△4 △5	△5	△4 △5	△4 △5	×	×	△5	△5
CS1W-EIP21	3	○	○	○	△5	△4 △5	△4 △5	×	×	×	×	×	×	×
CJ1W-EIP21	3	○	○	○	△5	△4 △5	△4 △5	×	×	×	×	×	×	×
CJ1W-EIP21 (CJ2)	3	△1	△1	○	△5	△4 △5	△4 △5	△5	△4 △5	△4 △5	×	×	△4 △5	△5
CJ2B-EIP21	3	△1	△1	○	△5	△4 △5	△4 △5	△5	△4 △5	△4 △5	×	×	△4 △5	△5
CJ1W-EIP21 (NJ)	3	△1 △2	△1 △2	△2	○	△4	△4	○	△4	△4	△4	×	△4	○
CS1W-EIP21S	4		○	○	△5	△4 △5	△4 △5	×	×	×	×	×	×	×
CJ1W-EIP21S	4	○		○	△5	△4 △5	△4 △5	×	×	×	×	×	×	×
CJ1W-EIP21S (CJ2)	4	△1	△1		△5	△4 △5	△4 △5	△5	△4 △5	△4 △5	×	×	△4 △5	△5
CJ1W-EIP21S (NJ)	4	△1 △2	△1 △2	△2		△4	△4	○	△4	△4	△4	×	△4	○

変更後ユニット形式		CS1W-EIP21S	CJ1W-EIP21S	CJ1W-EIP21S (CJ2)	CJ1W-EIP21S (NJ)	NJ501-□□□□ NJ301-□□□□ NJ101		NX701	NX1P2	NX102	NY512/532	NX-CSG	NX502	NX-EIP201
変更前ユニット形式	Rev	4	4	4	4	1	2	2	2	2	2	2	2	2
NJ501-□□□□ NJ301-□□□□ NJ101	1	△1 △2	△1 △2	△2	○		○	×	×	×	×	×	×	×
	2	△1 △2	△1 △2	△2	○	○		○	△4	○	○	×	○	○
NX701	2	×	×	△2	○	×	△4		△4	△4	△4	×	△4	○
NX1P2	2	×	×	△2	○	×	○	○		○	△4	×	○	○
NX102	2	×	×	△2	○	×	○	○	△4		△4	×	○	○
NY512/532	2	×	×	×	○	×	△4	○	○	○		×	△4	○
NX-CSG	2	×	×	×	×	×	×	×	×	×	×		×	×
NX502	2	×	×	△2 △7	△2 △7	×	△4 △7	△7	△4 △7	△4 △7	△7	×		○
NX-EIP201	2	×	×	△2 △6 △7	△2 △4 △7	×	△4 △7	△6 △7	△4 △7	△4 △7	△4 △7	×	△4	

○：　変更できます。
×：　変更できません。
△1：　タグにネットワーク変数が指定されている場合は変更できません。
△2：　タグ名、タグセット名の最大サイズ（UTF-8 変換後サイズ）が、48 バイトを超えている場合は変更できません。
△3：　次の項目が変更したデバイスの設定可能範囲を超えている場合は変更できません。
　　I/O コネクション数、タグ数、タグセット数、または 1 タグセットのサイズ
△4：　次のいずれかの場合は変更できません。
　　• I/O コネクション数、タグ数、タグセット数、1 タグセットのサイズが変更したデバイスの設定可能範囲を超えている
　　• RPI が設定可能範囲を超えている、または 0.5ms 単位（例：10.5ms）で設定されている
△5：　タグが物理アドレスで奇数サイズの場合は変更できません。
△6：　タグ、タグセット、リフレッシュサイズなどが設定可能範囲を超えている場合は変更できません。
　　（リフレッシュサイズとは 1 ノードがデータ交換するサイズで複数あるタグセットの総和になり上限があります。）
△7：　1 タグセットあたりのタグ最大数が設定可能範囲を超えている場合は変更できません。

7-2-18　デバイスステータス表示

メンテナンスモード時に、デバイスのステータスが、以下のアイコンで表示されます。
メンテナンスモードは、［表示］メニューから［大きいアイコン－メンテナンスモード］を選択します。



アイコン	状態
表示色：白	オフライン状態
表示色：グレー	デフォルト状態（コントローラ構成・設定が未実装状態を含む）
表示色：緑	アイドル状態（コントローラが「プログラム」モードのときを含む）
表示色：青	正常通信状態（コントローラが「運転」のときを含む）
表示色：黄	警告状態（コントローラが「部分停止フォールトレベルのコントローラ異常」または「運転継続異常」のときを含む）
表示色：赤	アラーム状態（コントローラが「全停止フォールトレベルのコントローラ異常」または「運転停止異常」のときを含む）

7-3　タグデータリンクのラダープログラム

7-3-1　タグデータリンクでのラダープログラム

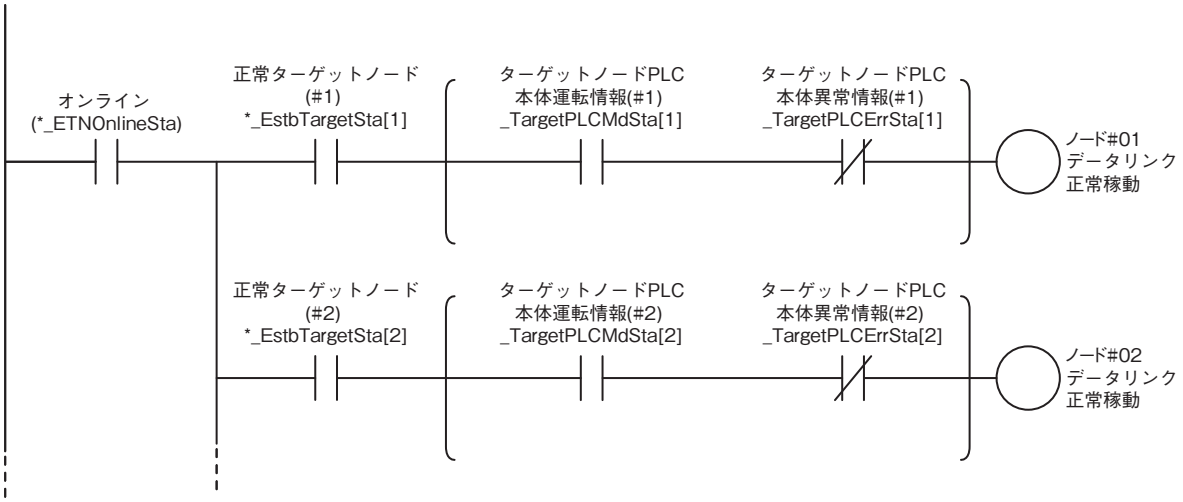
ラダープログラムでタグデータリンクしているデータを使用する場合には、以下の (1) ～ (3) の条件を用いてください。ターゲットノードのコントローラ情報を入力条件にしたい場合は、(4)、(5) の条件を追加してください。

- **EtherNet/IP ユニットがタグデータリンク可能状態になっている条件**
  - **(1) ユニット異常発生中（*_UnitErr）が FALSE**
  - **(2) オンライン（*_ETNOnlineSta）が TRUE**
- **ターゲットデバイスとのコネクションが確立しタグデータリンク中である条件**
  - **(3) 正常ターゲットノード（*_EstbTargetSta [0] ～ [63]）のターゲットノードアドレスに対応するビットが TRUE**
- **ターゲットノード PLC 本体運転情報（*_TargetPLCMdSta）による条件（オムロン製コントローラのみ有効）**
  - **(4) ターゲットノード PLC 本体運転情報の該当するターゲットノード PLC が TRUE**
- **ターゲットノード PLC 本体異常情報による条件（オムロン製コントローラのみ有効）**
  - **(5) ターゲットノード PLC 本体異常情報（*_TargetPLCErrSta）の該当するターゲットノード PLC 本体異常情報ビットが FALSE**
    ターゲットノードの PLC 本体運転・異常情報を使用したい場合は Network Configurator にてオリジネータ、ターゲット双方のタグセットでコントローラステータスを「含める」に設定しておく必要があります。

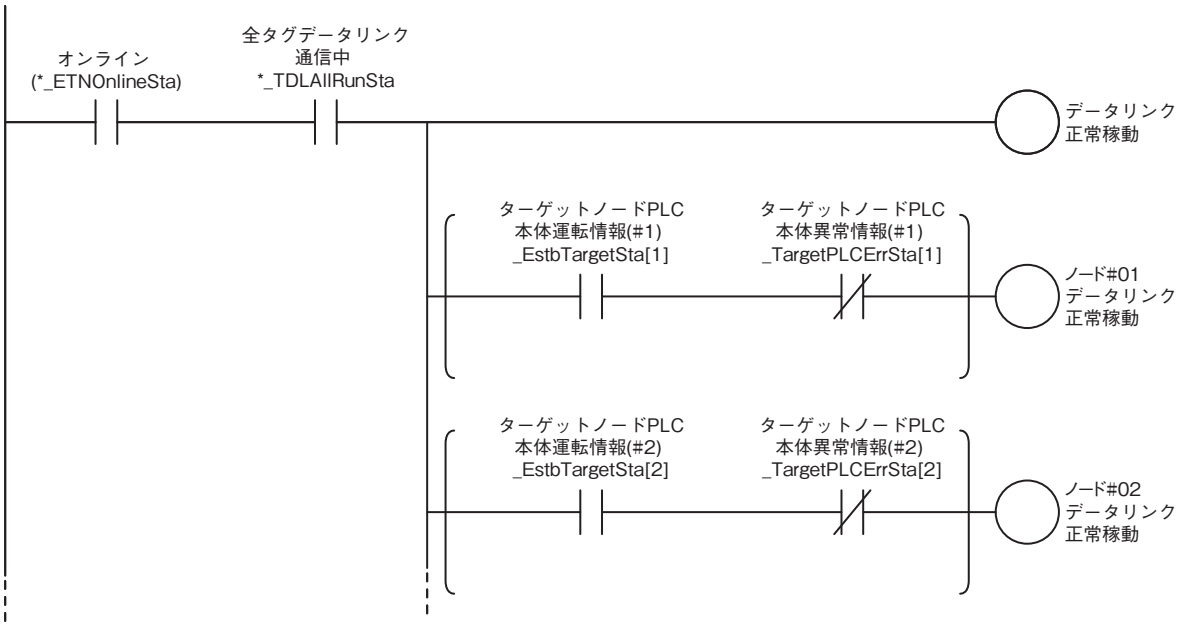
● 正常検知プログラム例

他ノード（ターゲットノード）ごとでタグデータリンク正常通信を確認するためのプログラムです。タグにコントローラステータスを使用するとターゲットノードごとのコントローラ状態も認識可能となります。

・正常検知プログラム例 1

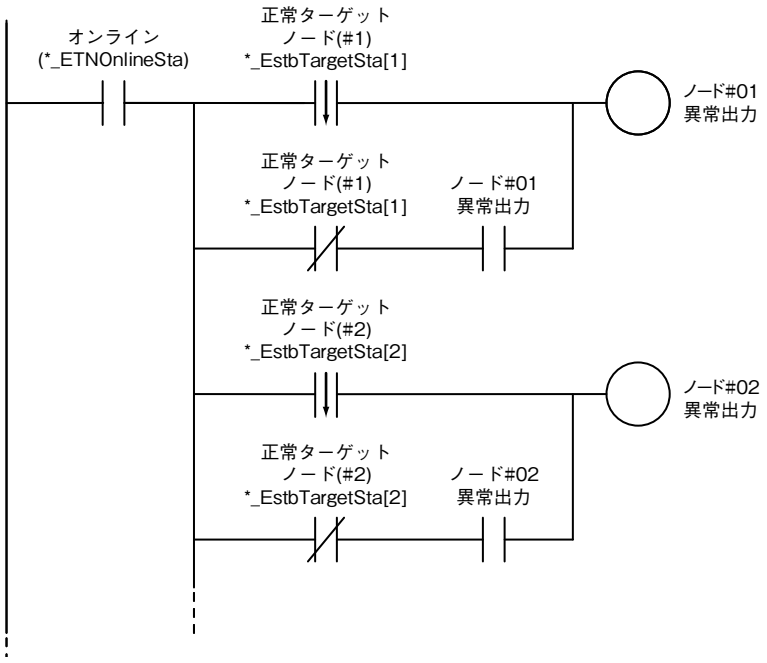


・正常検知プログラム例 2



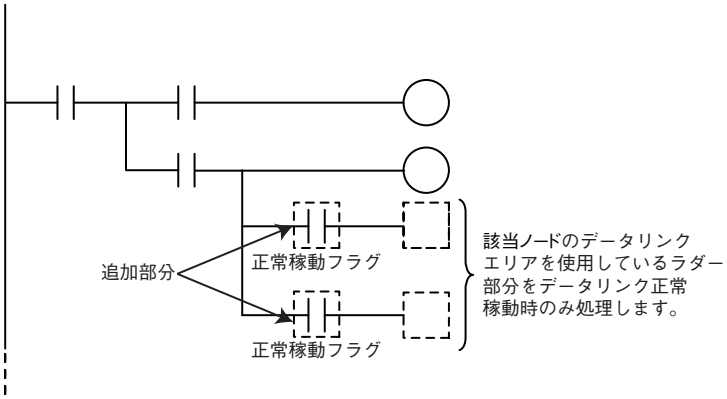
● 異常検知プログラム例

他ノード（ターゲットノード）ごとのタグデータリンク異常を確認するプログラムです。全ノードでデータリンクが正常に起動した後に新たに発生する異常を検知する場合に使用します。

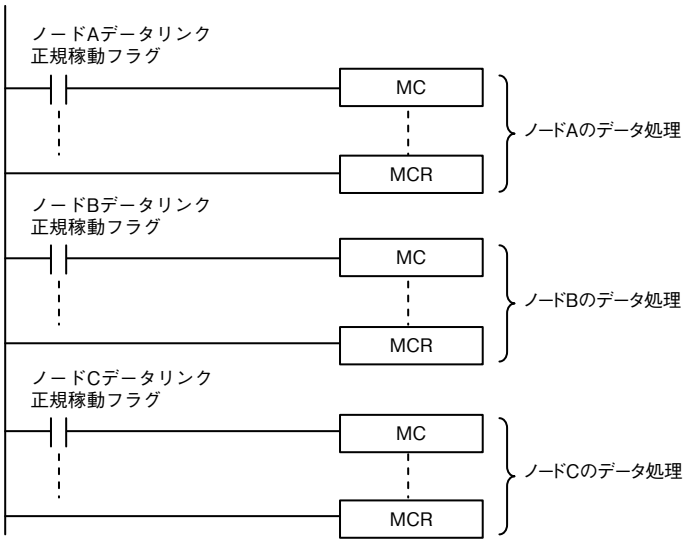


● データ処理プログラム例

• データリンク正常稼働時のみデータ処理を行う例



• MC − MCR 命令や JMP 命令を使用して正常時のみ処理を行う例



使用上の注意

ターゲットデバイスとの通信が異常となっても、ターゲットデバイスからの入力データは割り付けた自局エリアに保持されています。誤動作を防ぐため、ユニット異常発生中（*_UnitErr）が TRUE の場合、入力処理をしないように、ユーザプログラムを作成してください。

7-3-2　タグデータリンクに関連するステータスについて

タグデータリンクのステータスは、以下に示す CJ ユニットデバイス変数に反映されます。

変数名	内容
*_TargetPLCMdSta［0］～［63］（ターゲットノード PLC 本体運転情報） ※ コントローラステータスの「運転情報」に該当	EtherNet/IP ユニットがオリジネータとなっているコネクションにおいて該当ターゲットノードのコントローラの運転情報を表します。 本エリアの情報は、正常ターゲットノード情報が TRUE のときのみ有効です。FALSE の場合は直前の値が保持されます。 配列 [x] が TRUE：ノードアドレスが x のターゲット PLC が運転状態 配列 [x] が FALSE：上記以外
*_TargetPLCErrSta［0］～［63］（ターゲットノード PLC 本体異常情報） ※ コントローラステータスの「PLC 本体異常情報」に該当	登録ターゲットノード情報の該当するコネクションが確立していないか、ターゲットのコントローラに異常が発生していることを示します。 本エリアは、登録ターゲットノード情報が TRUE の場合のみ有効です。※ 配列 [x] が TRUE：ノードアドレスが x の登録ターゲットノード情報が TRUE で、正常ターゲットノードが FALSE、または、ターゲット PLC 異常情報が TRUE となっています。 配列 [x] が FALSE：ノードアドレスが x の登録ターゲットノード情報が FALSE の場合、または、登録ターゲットノード情報が TRUE で、正常ターゲットノード情報がTRUE となります。 コントローラ異常情報が FALSE の場合です。
*_EstbTargetSta [0] ～［63］(正常ターゲットノード) ※ コントローラステータスには該当しない	EtherNet/IP ユニットが、コネクションを正常に確立しているノードの一覧を表します。 配列 [x] が TRUE：ノードアドレスが x のノードとすべてのコネクションを正常に確立しています。 配列 [x] が FALSE：コネクションに未確立または異常が発生しているものがあります。

7-4　NJ CPU 以外の機種とタグデータリンクを行う場合について

タグデータリンクの性能は、CPU ユニットや EtherNet/IP ユニットの機種によって、以下のように異なります。
EtherNet/IP ユニットと、他の EtherNet/IP ユニットや CPU ユニットとタグデータリンクを行う場合、タグデータリンクの設定を通信性能の低いユニットに合わせて設定してください。

● タグデータリンクの性能仕様の違い

項目		形 CS1W-EIP21/EIP21S 形 CJ1W-EIP21/EIP21S 形 CJ2H-CPU6 □ -EIP	形 CJ2M-CPU3 □ ユニットバージョン		NJ CPU ユニット	
			Ver.2.0	Ver.2.1 以降	Ver.1.02 以前	Ver.1.03 以降
タグ	全タグの合計サイズ	184832CH	640CH		9600CH	
	1 タグあたりの最大サイズ	722CH （タグセットにコントローラステータスを含める場合は≦ 721CH）	20CH （タグセットにコントローラステータスを含める場合は≦ 19CH）	640CH （タグセットにコントローラステータスを含める場合は≦ 639CH）	300CH （タグセットにコントローラステータスを含める場合は≦ 299CH）	
	登録可能なタグの数	256	32		256[*1]	
タグセット	1 タグセットの最大サイズ	722CH （タグセットにコントローラステータスを含める場合≦ 721CH）	20CH （タグセットにコントローラステータスを含める場合≦ 19CH）	640CH （タグセットにコントローラステータスを含める場合≦ 639CH）	300CH （タグセットにコントローラステータスを含める場合≦ 299CH）	
	1 タグセットに含めるタグの数	8 （タグセットにコントローラステータスを含める場合は 7） 注 . 入力、出力の変数を混在させることはできません。				
	登録可能なタグセットの数	256	32		32	
コネクション	コネクション数	256	32		32	
	1 コネクションあたりの最大データサイズ	252CH または 722CH[*2] （1 コネクション内でのデータの同期性を確保するための条件については、📖「7-1-7 タグデータリンクするデータの同期性について」を参照してください。	20CH （1 コネクション内でのデータの同期性は確保されます。）	640CH	300CH （1 コネクション内でのデータの同期性を確保するための条件については、📖「7-1-7 タグデータリンクするデータの同期性について」を参照してください。	
パケットインターバル（RPI）		0.5 ～ 10000ms（0.5ms 単位）	1 ～ 10000ms（0.5ms 単位）		10 ～ 10000ms（1ms 単位）	1 ～ 10000ms（1ms 単位）
ユニット許容通信帯域（PPS）		6000 ～ 12000pps[*3,*4]	3000pps[*3]		1000pps[*3]	3000pps[*3]

*1. 次の条件でタグを登録した場合のタグ数を示します。
・ 1 タグセットに含めるタグ数 8 をすべて使用
・ 登録可能なタグセット数 32 をすべて使用
*2. データサイズとして 505 ～ 1444byte を使用するには、LargeForwardOpen（CIP オプション仕様）に対応している必要があります。SYSMAC CS/CJ シリーズ間では使用可能ですが、他社ノードとの接続の場合、使用される機器 LargeForwardOpen 仕様を対応しているかご確認ください。
*3. Packet Per Second を意味し、1 秒間に処理可能な送受信パケット数を示します。
*4. ユニットバージョン Ver.2.1 以前の場合は、6000pps となります。

● タグの指定方法

タグの割り付け先を指定する方法には、変数と I/O メモリアドレスの 2 種類の方法がありますが、使用する CPU ユニットによっては、両方ともサポートしていない機種もあります。
ただし、それらの機器と通信する場合でも、通信する相手ノードのタグ設定は I/O メモリアドレス、
もしくはネットワーク変数名のどちらで指定されているかには依存しませんので、相互に通信が可能です。
CPU ユニットごとに可能なタグの指定方法は、以下のとおりとなります。

○：可能　×：不可

CPU ユニット	EtherNet/IP ユニット	Network Configurator ハードウェアリスト上の名称	ネットワーク変数名での指定	I/O メモリアドレスでの指定
NX シリーズ CPU ユニット	−	NX701	○	×
NJ シリーズ CPU ユニット	−	NJ501-□□□□ NJ301-□□□□ NJ101	○	○ *1
	形 CJ1W-EIP21	CJ1W-EIP21(NJ)	○	○ *1
	形 CJ1W-EIP21S	CJ1W-EIP21S(NJ)	○	○ *1
形 CJ2H-CPU6□-EIP	−	CJ2B-EIP21	○	○
	形 CJ1W-EIP21	CJ1W-EIP21(CJ2)	○	○
	形 CJ1W-EIP21S	CJ1W-EIP21S(CJ2)	○	○
形 CJ2H-CPU6□	形 CJ1W-EIP21	CJ1W-EIP21(CJ2)	○ *2	○
	形 CJ1W-EIP21S	CJ1W-EIP21S(CJ2)	○ *2	○
形 CJ2M-CPU3□	−	CJ2M-EIP21	○	○
	形 CJ1W-EIP21	CJ1W-EIP21(CJ2)	○	○
	形 CJ1W-EIP21S	CJ1W-EIP21S(CJ2)	○	○
形 CJ2M-CPU1□	形 CJ1W-EIP21	CJ1W-EIP21(CJ2)	○ *3	○
	形 CJ1W-EIP21S	CJ1W-EIP21S(CJ2)	○ *3	○
CJ1 CPU ユニット	形 CJ1W-EIP21	CJ1W-EIP21	×	○
	形 CJ1W-EIP21S	CJ1W-EIP21S	×	○
CS1 CPU ユニット	形 CS1W-EIP21	CS1W-EIP21	×	○
	形 CS1W-EIP21S	CS1W-EIP21S	×	○

*1. I/O メモリアドレスをタグに指定したい場合は、タグに I/O メモリアドレスを直接指定するのではなく、Sysmac Studio で I/O メモリアドレスを AT 指定した変数を作成し、その AT 指定した変数を、タグとして指定してください。
*2. 形 CJ2H-CPU6□ Ver.1.6 以降が対応しています。
*3. 形 CJ2M-CPU1□ Ver.2.2 以降が対応しています。

8

メッセージ通信機能

8-1 CIP メッセージ通信機能の概要 8-2
8-1-1 CIP メッセージ通信サービス機能の概要 8-2
8-1-2 メッセージ通信サービスの機能・性能一覧 8-2

8-2 CIP メッセージ通信の使い方 8-3
8-2-1 概要 8-3
8-2-2 CIP 通信命令の種類 8-4
8-2-3 CIP 通信命令の使用方法 8-5
8-2-4 ルートパス 8-6
8-2-5 CIP 通信命令使用時の派生データ型の用意 8-12
8-2-6 サンプルプログラム（CIP コネクションレス型 (UCMM) メッセージで通信する場合）..... 8-14
8-2-7 サンプルプログラム（CIP コネクション型 (Class3) メッセージで通信する場合）..... 8-21
8-2-8 動作タイミング 8-28
8-2-9 レスポンスコード 8-29

8-3 EtherNet/IP ユニット宛 CIP オブジェクトについて 8-33
8-3-1 EtherNet/IP ユニット宛 CIP オブジェクトの種類 8-33
8-3-2 Identity オブジェクト（クラス ID：01Hex）..... 8-33
8-3-3 TCP/IP Interface オブジェクト（クラス ID：F5Hex）..... 8-36
8-3-4 Ethernet Link オブジェクト（クラス ID：F6 Hex）..... 8-38
8-3-5 PLC オブジェクト（クラス ID：C4Hex）..... 8-41

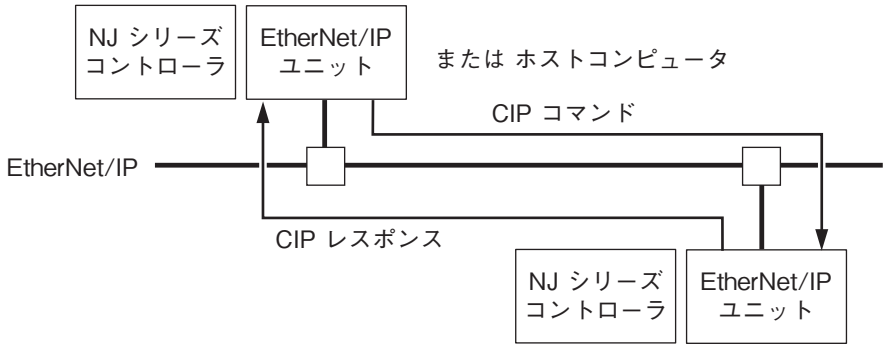
8-1　CIP メッセージ通信機能の概要

8-1-1　CIP メッセージ通信サービス機能の概要

EtherNet/IP ネットワーク上のデバイスに対して、必要時に任意の CIP コマンドを発行することができます。

NJ シリーズ CPU ユニットのプログラム上で CIPSend 命令を実行することによりデータ読み書きなどの各種 CIP コマンドを発行し、レスポンスを受信します。

本機能により、サーバ側コントローラでユーザプログラムを作成しなくても、クライアント側からサーバ側コントローラのメモリ読出／書込などを行うことができます。



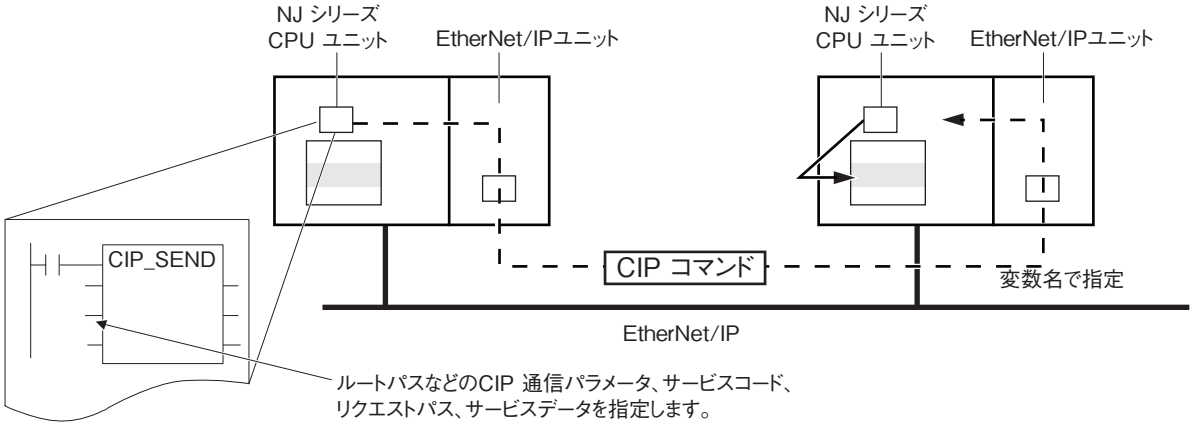
8-1-2　メッセージ通信サービスの機能・性能一覧

項目		内容
メッセージ種類		以下を選択可能 CIP コネクションレス型（UCMM） CIP コネクション型（Class3）
実行方法		CIPSend 命令（Class3：コネクション型）、または CIPUCMM（UCMM：非コネクション型）による
データ内容		任意の CIP コマンドの発行、レスポンス受信
通信パラメータ		メッセージ種別、タイムアウト時間、ルートパス指定
1 コネクションあたりのデータ長	非コネクション型（UCMM）	502 バイト
	コネクション型（Class3）	502 バイト（Forward_Open） 1994 バイト（Large_Forward_Open）

8-2　CIP メッセージ通信の使い方

8-2-1　概要

NJ シリーズ CPU ユニットのユーザプログラム上で、CIP 通信命令を実行することにより、EtherNet/IP 上の NJ CPU ユニット、または CJ2 CPU ユニットの変数の読み書き、任意の CIP コマンドを発行することができます。



8-2-2　CIP 通信命令の種類

CIP 通信命令は以下の種類があります。

CIP 通信命令については

🕮『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）を参照してください。

命令	命令名	内容	通信方式
CIPUCMM Read 命令	変数読出（UCMM Explicit メッセージ）	指定した CIP ネットワーク上の、他局コントローラのネットワーク公開属性の変数の値を読み出して、自局コントローラの変数に値を格納します。	CIP コネクションレス型（UCMM）
CIPUCMM Write 命令	変数書込（UCMM Explicit メッセージ）	指定した CIP ネットワーク上の、他局コントローラのネットワーク公開属性の変数に、自局コントローラの変数の値を書き込みます。	
CIPUCMM Send 命令	任意 Explicit メッセージ送信（UCMM）	指定した CIP ネットワーク上の他局コントローラに対して、任意の CIP コマンドを発行します。 NJ シリーズ CPU ユニットに対して使うことができるサービスコードやレスポンスコードなどの詳細については 🕮「8-2-9 レスポンスコード」、🕮「8-3 EtherNet/IP ユニット宛 CIP オブジェクトについて」を参照してください。	
CIPOpen 命令	CIP Class3（Large_Forward_Open）コネクション確立	指定した相手ノードに対して、CIP Class3（Large_Forward_Open）コネクションを確立します。	CIP コネクション型（Class3）
CIPOpen WithData Size 命令	CIP Class3 コネクション確立（サイズ指定あり）	指定した相手ノードに対して、指定したデータ長以下の Class3 Explicit メッセージを送受信可能な CIP の Class3 コネクションを確立します。	
CIPRead 命令	変数読出（Class3 Explicit メッセージ）	指定した CIP ネットワーク上の、他局コントローラのネットワーク公開属性の変数の値を読み出して、自局コントローラの変数に値を格納します。	
CIPWrite 命令	変数書込（Class3 Explicit メッセージ）	指定した CIP ネットワーク上の、他局コントローラのネットワーク公開属性の変数に、自局コントローラの変数の値を書き込みます。	
CIPSend 命令	任意 Explicit メッセージ送信（Class3）	指定した CIP ネットワーク上の他局コントローラに対して、任意の CIP コマンド（Class3）を発行します。 NJ シリーズ CPU ユニットに対して使うことができるサービスコードやレスポンスコードなどの詳細については、🕮「8-2-9 レスポンスコード」、🕮「8-3 EtherNet/IP ユニット宛 CIP オブジェクトについて」を参照してください。	
CIPClose 命令	CIP Class3 コネクション切断	指定したハンドルの CIP コネクション（Class3）を切断します。	

バージョン関連情報

CIPOpenWithDataSize 命令は、CPU ユニット Ver.1.06 以降、かつ Sysmac Studio Ver.1.07 以降で、使用可能です。

8-2-3　CIP 通信命令の使用方法

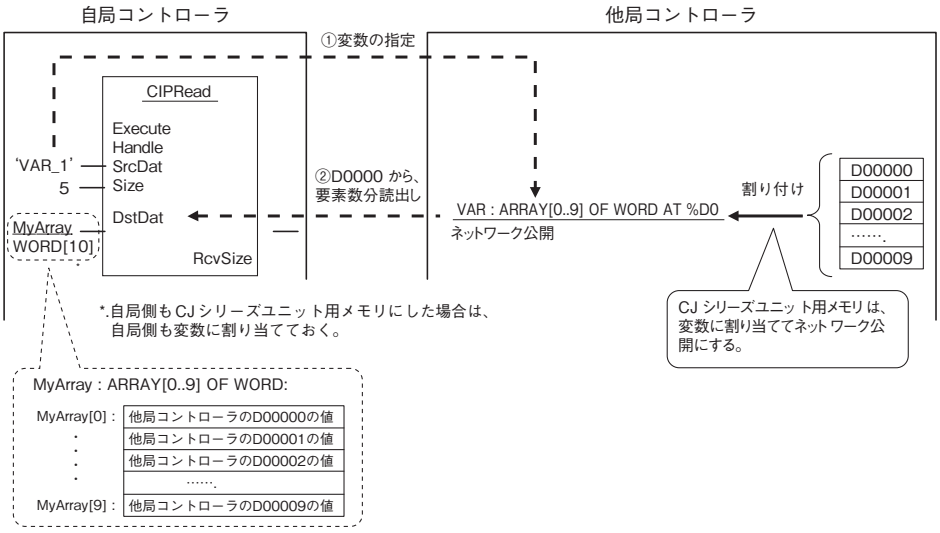
CIP メッセージ通信には、以下の処理があります。

通信方式が、CIP コネクション型（Class3）の場合、データの送受信を行う前後に、オープン処理とクローズ処理が必要となります。

処理	内容	命令
オープン処理 CIP Class3 コネクション型の場合のみ*1	CIP メッセージを行うときに、最初に実行する処理です。コネクションの確立までを行います。	CIPOpen CIPOpenWithDataSize
変数データの送受信処理*2	指定した公開属性の変数に対し、データの読出し／書込みを行います。	CIPUCMMRead CIPUCMMWrite CIPRead CIPWrite
CIP コマンド送信	任意の CIP コマンドを発行します。	CIPUCMMSend CIPSend
クローズ処理 CIP Class3 コネクション型の場合のみ	コネクションの切断を行います。	CIPClose

*1. オープン処理で取得できるコネクションのハンドルは、同時に 32 個までです。
ハンドルは、コネクションがタイムアウトなどで切断された場合も保持されますので、CIPClose 命令を実行してください。

*2. CJ シリーズユニット用メモリを直接指定（D0000 など）してアクセスすることはできません。
CJ シリーズユニット用メモリにアクセスする場合は、AT 指定した変数に対してアクセスすることが必要です。



使用上の注意

CIP 通信命令は、命令の種類に関わらず同時に使用できるのは、32 個までです。
33 個以上同時に実行されないように、ユーザプログラムで排他処理を行ってください。

8-2-4　ルートパス

自 CPU ユニットから、ネットワーク上の他コントローラまでの、経路を示したパスです。
CIP 通信命令では、このルートパスをもとに、ルーチングを行います。

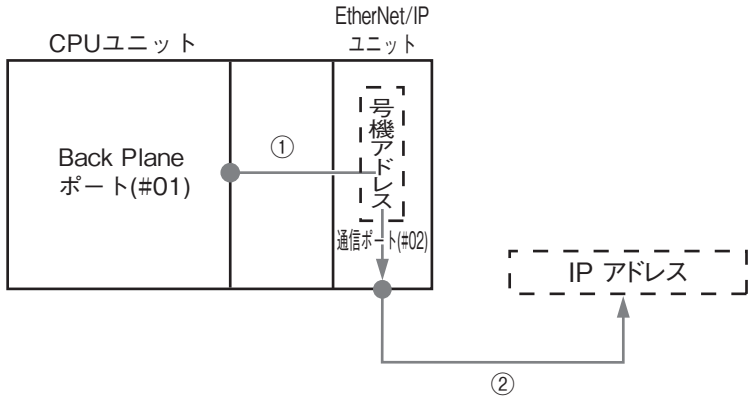
ルートパスの記述フォーマット

ルートパスの記述は「EPATH 型」で、基本のフォーマットは、以下のように記述します。

ネットワーク種別番号 ¥相手アドレス

経路の種類が、① CPU ラック上のユニットか、②通信ユニットの通信ポートかに応じて、ネットワーク種別番号および相手アドレスは以下のようになります。

経路	ネットワーク種別番号（16 進）	相手アドレス（16 進）
① CPU ラック上のユニットに出力	#01 （Back Plane ポート）	相手先ユニットの号機アドレス （下記の参考を参照）
②通信ユニットの通信ポートから出力	#02 （EtherNet/IP ユニット）	IP アドレス



① CPU ラック上のユニットに出力する経路の場合
CPU ラックとのネットワークとしての Back Plane ポートに出力し、相手先ユニットのアドレスとして、号機アドレスを指定します。
② 通信ユニットの通信ポートから出力する経路の場合
EtherNet/IP ポートに出力して、相手先ノードのアドレスとして、IP アドレスを指定します。

参考

号機アドレスとは
ネットワーク上の 1 ノード内に接続されている複数の機器を、識別するためのアドレスです。
以下のように設定されています。
- CPU ユニット　：00Hex、01Hex
- CPU 高機能ユニット（EtherNet/IP ユニット）：ユニット番号 +10Hex

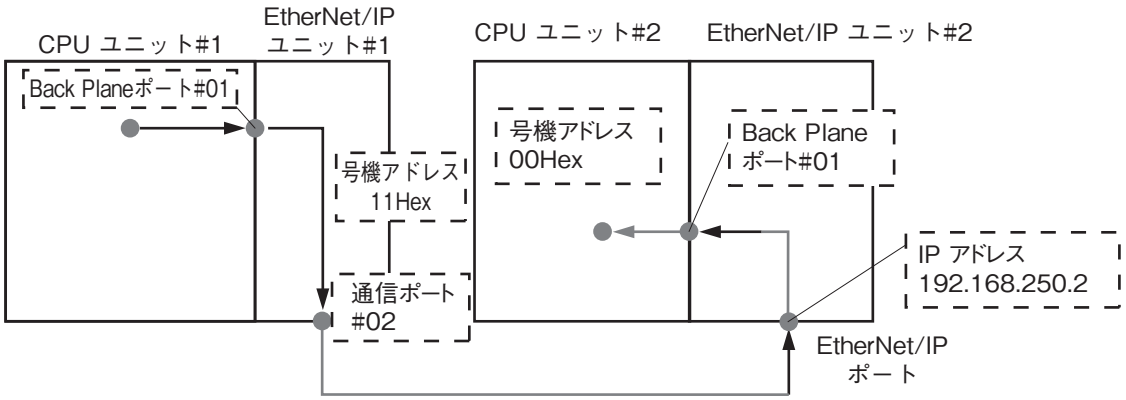
ルートパスの記述例

ルートパスの記述は、内蔵 EtherNet/IP ポートへ通信するか、EtherNet/IP ユニットへ通信するかにより異なります。

以下にルートパスの記述例を示します。

● EtherNet/IP ユニットへ通信する場合

例）CPU ユニット #1 に装着した EtherNet/IP ユニットから、CPU ユニット #2 に装着されている EtherNet/IP ユニット経由で、CPU ユニット #2 に通信する場合



① CPU ユニット #1 → EtherNet/IP ユニット #1
- •ネットワーク種別番号：“#01”（内部 Back Plane ポートによる出力）
- •相手アドレス：“#11Hex”（EtherNet/IP ユニットの号機アドレス（ユニット番号：1+10Hex））

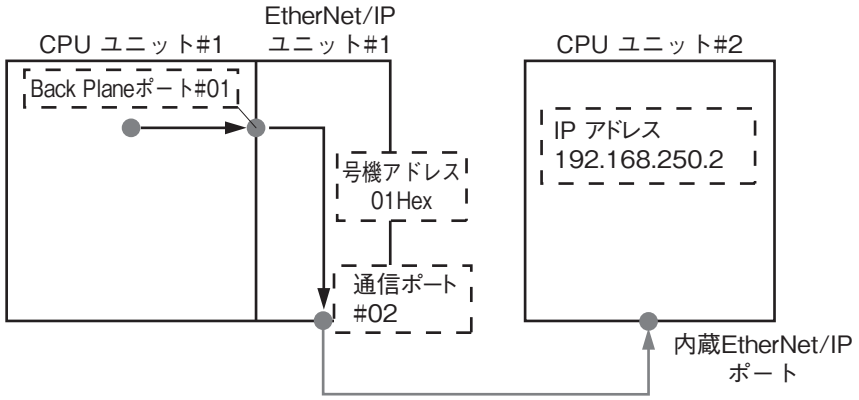
② EtherNet/IP ユニット #1 → EtherNet/IP ユニット #2
- •ネットワーク種別番号：“#02”（EtherNet/IP ポートによる出力）
- •相手アドレス：相手 IP アドレスを指定

③ EtherNet/IP ユニット #2 → CPU ユニット #2
- •ネットワーク種別番号：“#01”（内部 Back Plane ポートによる出力）
- •相手アドレス：“#00”（CPU ユニットの号機アドレス）

ルートパス　：　¥01¥#11¥02¥192.168.250.2¥01¥#00
①　②　③

● 内蔵 EtherNet/IP ポートへ通信する場合

例)CPUユニット #1に装着したEtherNet/IPユニットから、CPUユニット #2の内蔵EtherNet/IPポートに通信する場合



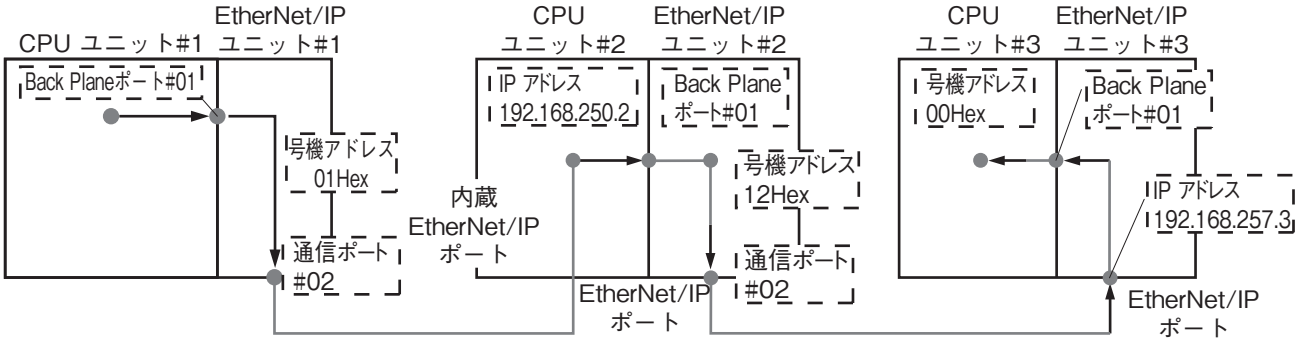
① CPU ユニット #1 → EtherNet/IP ユニット #1
- •ネットワーク種別番号：“#01”（内部 Back Plane ポートによる出力）
- •相手アドレス：“#01Hex”（EtherNet/IP ユニットの号機アドレス）

② EtherNet/IP ユニット #1 → CPU ユニット #2（内蔵 EtherNet/IP ポート）
- •ネットワーク種別番号：“#02”（EtherNet/IP ポートによる出力）
- •相手アドレス：相手 IP アドレスを指定

ルートパス　：　¥01¥#01¥02¥192.168.250.2

● 中継局を介してアクセスする場合

CPU ユニット #2 を介して、CPU ユニット #1 から CPU ユニット #3 に通信を行う場合



① CPU ユニット #1 → EtherNet/IP ユニット #1
- •ネットワーク種別番号：“#01”（内部 Back Plane ポートによる出力）
- •相手アドレス：“#01Hex”（EtherNet/IP ユニットの号機アドレス）

② EtherNet/IP ユニット #1 → CPU ユニット #2（内蔵 EtherNet/IP ポート）
- •ネットワーク種別番号：“#02”（EtherNet/IP ポートによる出力）
- •相手アドレス：相手 IP アドレスを指定

③ CPU ユニット #2 → EtherNet/IP ユニット #2
- •ネットワーク種別番号：“#01”（内部 Back Plane ポートによる出力）
- • 相手アドレス：“#12Hex”（EtherNet/IP　ユニットの号機アドレス（ユニット番号：2+10=12Hex））

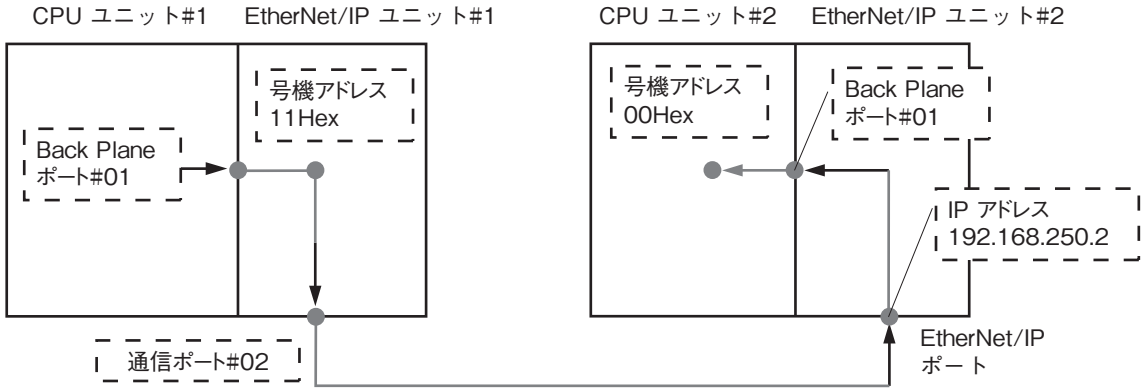
④ EtherNet/IP ユニット #2 → EtherNet/IP ユニット #3
- •ネットワーク種別番号：“#02”（EtherNet/IP ポートによる出力）
- •相手アドレス：相手 IP アドレスを指定

⑤ EtherNet/IP ユニット #3 → CPU ユニット #3
- •ネットワーク種別番号：“#01”（内部 Back Plane ポートによる出力）
- •相手アドレス：“#00”（CPU ユニットの号機アドレス）

ルートパス　：　¥01¥#01¥02¥192.168.250.2¥01¥#12¥02¥192.168.257.3¥01¥#00
①　②　③　④　⑤

● EtherNet/IP ユニット同士で通信する場合

例）CPU ユニット #1、#2 にそれぞれ装着されている EtherNet/IP ユニットにより通信する場合



① CPU ユニット #1 → EtherNet/IP ユニット #1
- ネットワーク種別番号：“#01”（内部 Back Plane ポートによる出力）
- 相手アドレス：“#11”（EtherNet/IP　　ユニットの号機アドレス（ユニット番号：1+10Hex））

② EtherNet/IP ユニット #1 → EtherNet/IP ユニット #2
- ネットワーク種別番号：“#02”（EtherNet/IP ポートによる出力）
- 相手アドレス：相手 IP アドレスを指定

③ EtherNet/IP ユニット #2 → CPU ユニット #2
- ネットワーク種別番号：“#01”（内部 Back Plane ポートによる出力）
- 相手アドレス：“#00”（CPU ユニットの号機アドレス）

ルートパス　：　¥01¥#11¥02¥192.168.250.2¥01¥#00
①　②　③

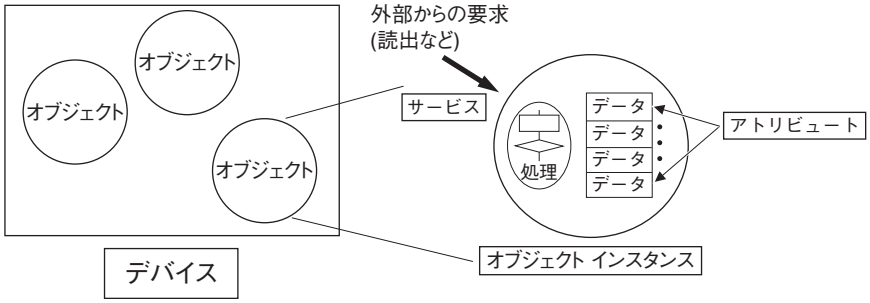
①自EtherNet/IP ユニットの号機アドレス（ユニット番号:1+10Hex）
②相手EtherNet/IP ユニットのIP アドレス
③相手CPU ユニットの号機アドレス

レ　バージョン関連情報

- EtherNet/IPユニット形CJ1W-EIP21をNJシリーズコントローラに装着して使用するのは、CPU ユニットの Ver.1.01 以降、かつ Sysmac Studio Ver1.02 以降で可能です。
- EtherNet/IP ユニット形 CJ1W-EIP21S を NJ シリーズコントローラに装着して使用するのは、CPU ユニットの Ver.1.67 以降、かつ Sysmac Studio Ver1.60 以降で可能です。

リクエストパス（IOI）について

CIP の世界では、各デバイスは「オブジェクト」の集合としてモデル化されます。オブジェクトとは、デバイスの特定の構成要素を抽象的に表現したものです。



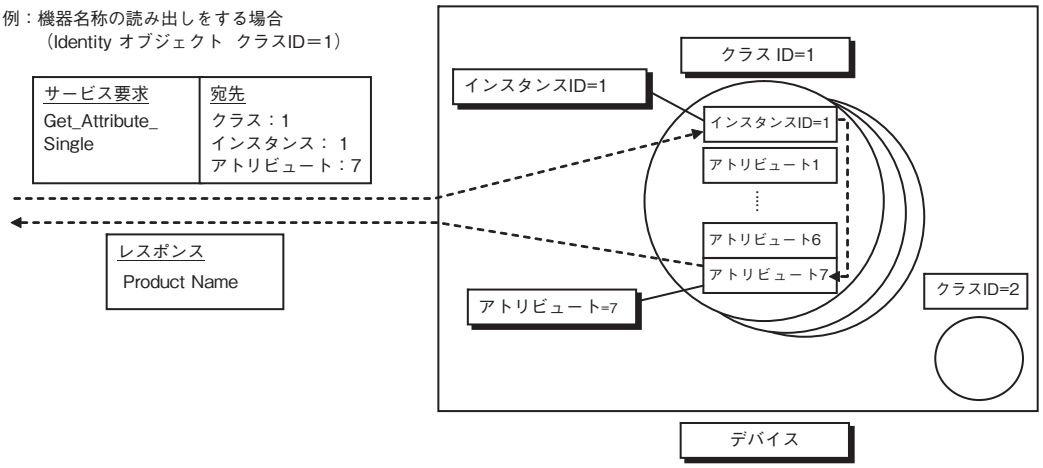
CIP Common Specification では「オブジェクト」、「クラス」、「インスタンス」、「アトリビュート」、「サービス」を次のように説明しています。（CIP Common Specification より）

用語	意味
オブジェクト	デバイス内の機能を抽象化して、処理とデータをまとめたもの
クラス	同種のシステム構成要素を表現するすべてのオブジェクトの集合で、オブジェクトを一般化したもの。
インスタンス	オブジェクトの具体的かつ実際の（物理的な）存在。
アトリビュート	外部から確認できるオブジェクトの特長や機能を記述したもの。
サービス	オブジェクト外からの要求（例えばデータの読み出しなど）。

各オブジェクトにアクセスするには、「クラス ID」、「インスタンス ID」、「アトリビュート ID」を使います。

デバイス内ではこの 3 つを指定することでアクセス先を特定することができます。

オブジェクト外から「サービス」を要求するときには、この「クラス ID」、「インスタンス ID」、「アトリビュート ID」を指定します。（サービスによってはインスタンス ID、アトリビュート ID を必要としないものもあります）



このように「クラス ID」、「インスタンス ID」、「アトリビュート ID」でデバイス内部の位置を特定するので「IOI (Internal Object Identifier)」と呼ばれます。

オブジェクトごとの「クラス ID」、「インスタンス ID」、「アトリビュート ID」、「サービスコード」については、🕮「8-3 EtherNet/IP ユニット宛 CIP オブジェクトについて」を参照してください。

8-2-5　CIP 通信命令使用時の派生データ型の用意

CIP 通信命令を使用するには、あらかじめ通信パラメータの値を入力するための派生データ型の変数を用意しておく必要があります。

本項で説明している変数について、詳細な仕様や設定方法については、🕮『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）を参照してください。

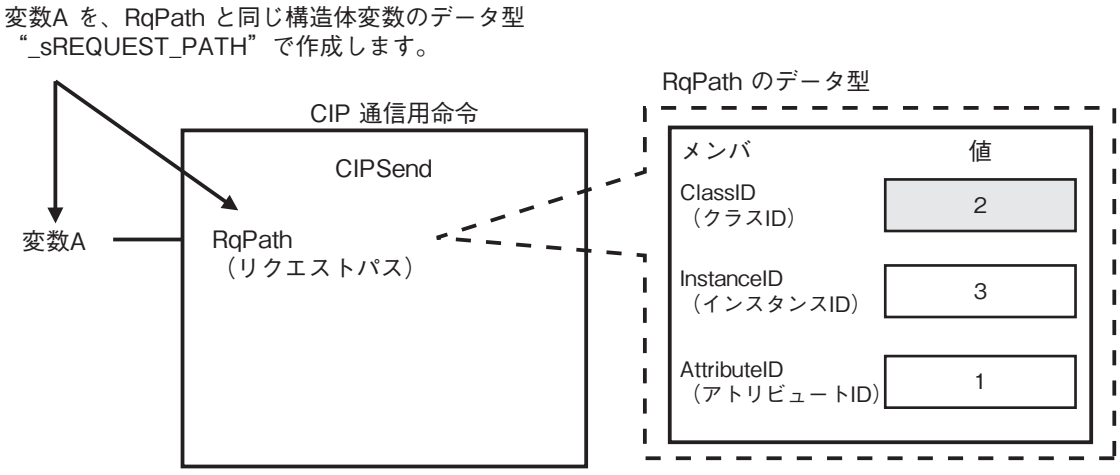
リクエストパス入力のための構造体型変数の用意

CIP 通信命令では、リクエストパスの各オブジェクトを、1 つの構造体型の変数にまとめています。

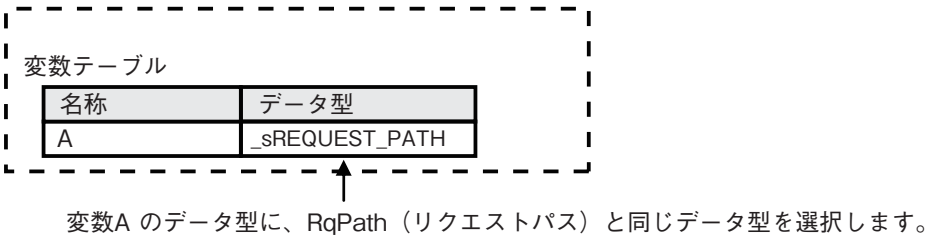
● 構造体型変数の作成

CIP 通信命令の構造体型変数に値を入力するには、同じ構造のユーザ変数を、あらかじめ作成しておく必要があります。

例）CIPSend 命令の配列型変数（RqPath）にデータを入力する変数を作成する

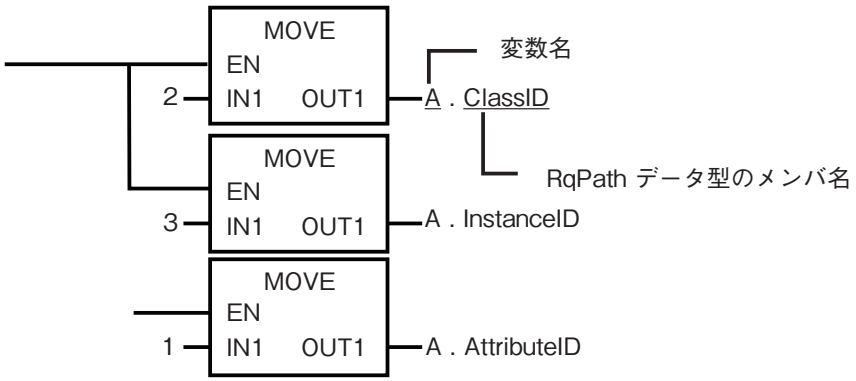


変数テーブルで変数作成時、変数のデータ型に、あらかじめ登録されている CIP 通信命令用のデータ型（_sREQUEST_PATH）を選択します。
これで、RqPath と同じ構造の構造体変数になります。
例）



● 構造体変数の各メンバに値を入力する

構造体変数にメンバとして登録されている各通信パラメータに、以下のように値を入力します。



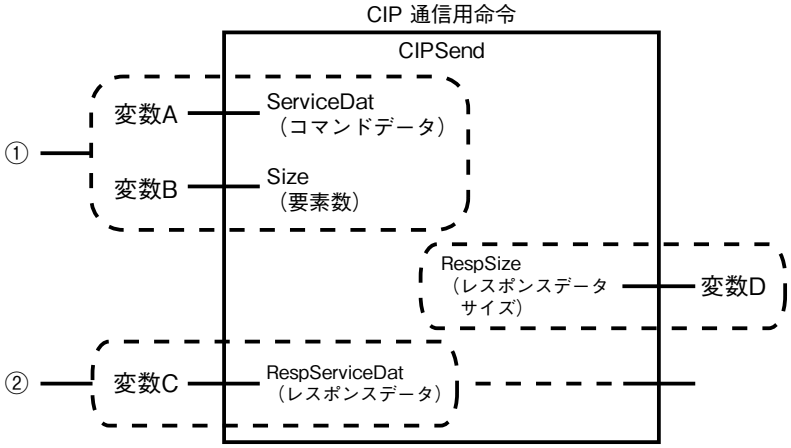
サービスデータ／レスポンスデータなどの入出力のための配列型変数の用意

CIP 通信命令では、配列型変数に格納したデータを送受信します。

● 配列型変数の作成

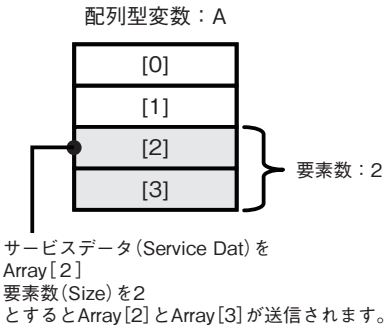
CIP 通信命令の配列型変数に値を入力するには、その配列型変数と同じ構造の変数を、あらかじめ作成しておく必要があります。

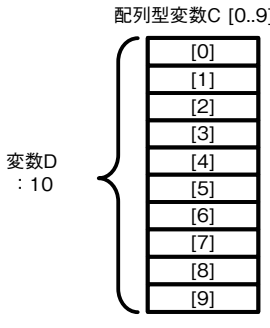
例）CIPSend 命令の配列型変数にデータを入力する変数を作成する



①送信するサービスデータの入力配列変数Aに送信データを格納します。配列変数 A の中で、特定の要素のみ指定する場合は、変数 B で要素数を指定します。

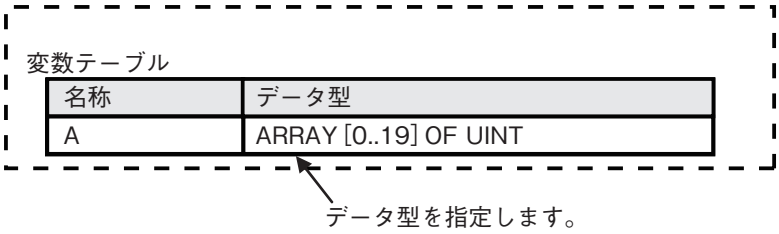
②受信したレスポンスデータを変数 C に格納します。また、実際に受信したレスポンスデータのバイトサイズを変数 D に格納します。





変数テーブルで以下のように変数を作成します。

要素の開始番号、要素の終了番号、データ型を指定します。
例）UINT 配列の場合



● **配列型変数を使用している CIP 通信命令**

命令名	構造体変数名		
	入力変数	入出力変数	出力変数
CIPRead	−	−	DstDat（読出しデータ）
CIPWrite	SrcDat（書込みデータ）	−	−
CIPSend	ServiceDat（コマンドデータ）	RespServiceDat（レスポンスデータ）	−

8-2-6　サンプルプログラム（CIP コネクションレス型 (UCMM) メッセージで通信する場合）

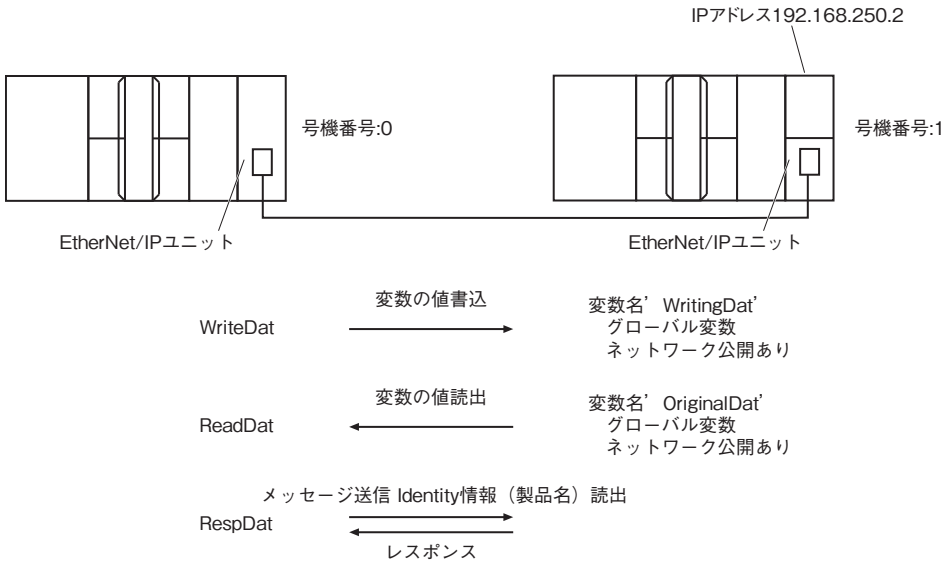
CIP の UCMM メッセージを使用して、変数書込、変数読出、メッセージ送信を行います。
コントローラどうしは、EtherNET/IP ネットワークで接続しており、相手ノードの IP アドレスは 192.168.250.2 とします。
処理の手順は、以下のようになります。

1 **CIPUCMMWrite 命令を使用して、相手ノードの変数に値を書き込みます。他局変数名は 'WritingDat'、書き込む値は、変数 WriteDat の中身とします。'WritingDat' は、相手ノードでグローバル変数として定義されており、かつ、ネットワーク公開されていることが必要です。**

2 **CIPUCMMRead 命令を使用して、相手ノードの変数の値を読み出します。他局変数名は 'OriginalDat'、読み出した値の格納先は、変数 ReadDat とします。'OriginalDat' は、相手ノードでグローバル変数として定義されており、かつ、ネットワーク公開されていることが必要です。**

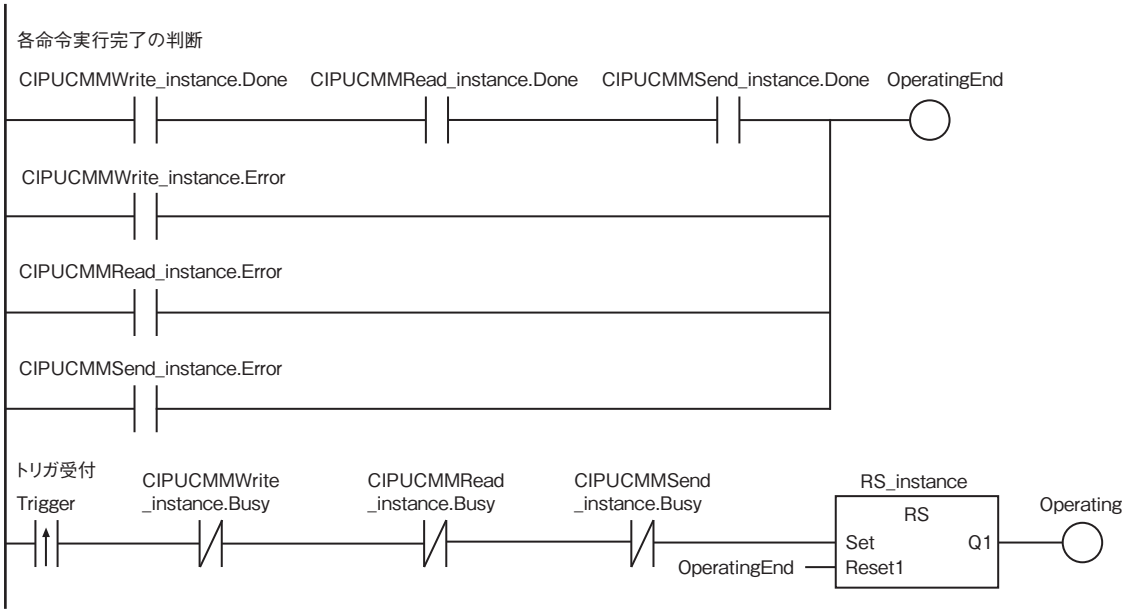
3 **CIPUCMMSend 命令を使用して、相手ノードに Explicit メッセージを送信します。メッセージの内容は、Identity 情報（製品名）読出です。その際のクラス ID、インスタンス ID、アトリビュート ID、サービスコードは、以下のとおりです。レスポンスデータは、変数 RespDat に格納します。**

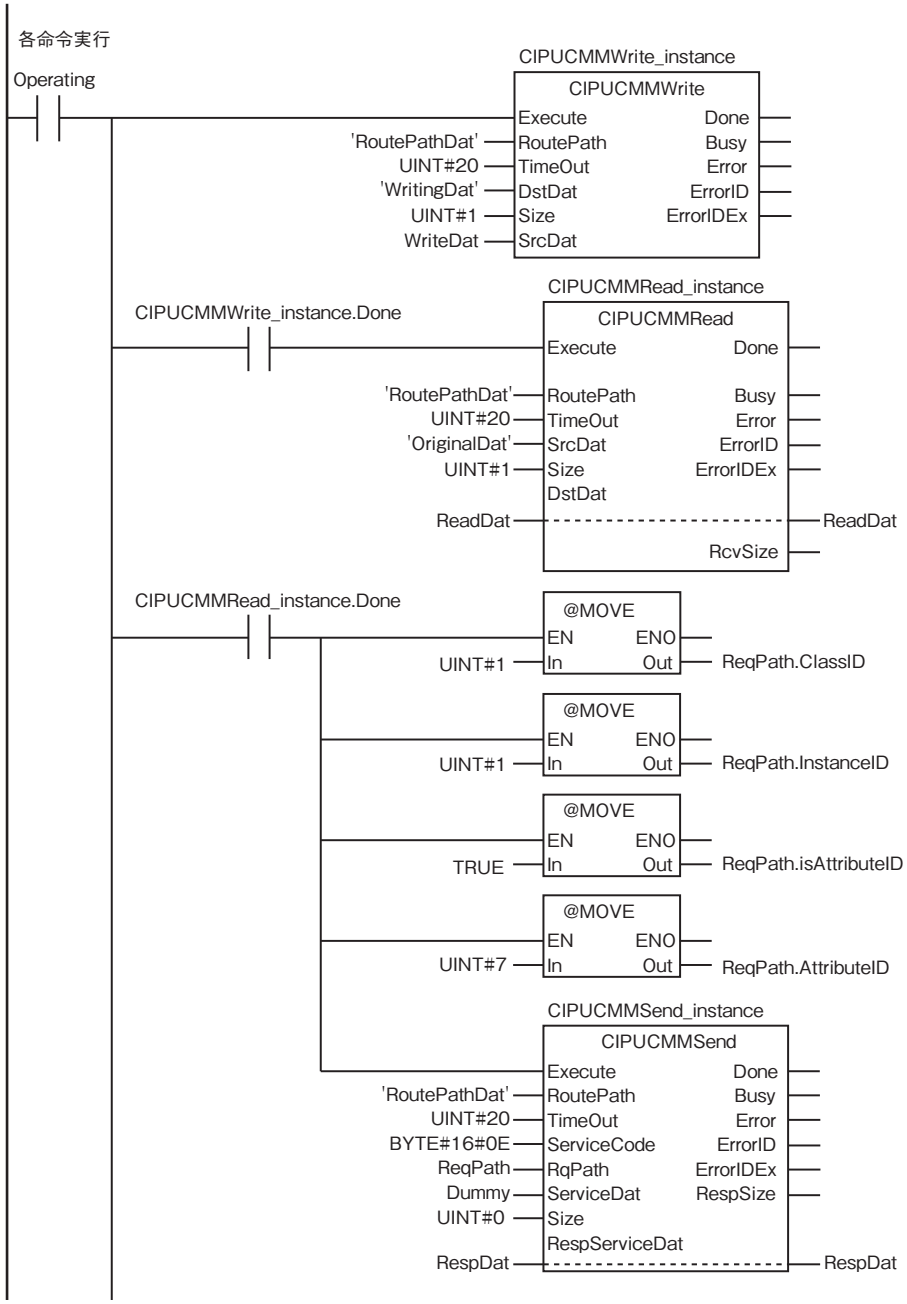
項目	値
クラス ID	1
インスタンス ID	1
アトリビュート ID	7
サービスコード	16#0E

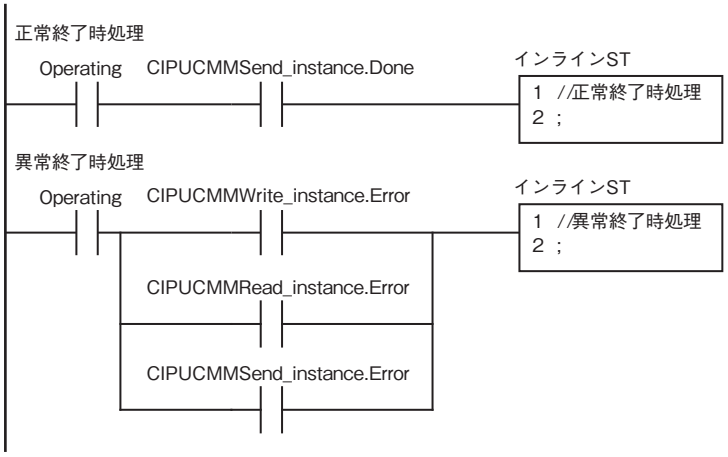

IPアドレス192.168.250.2
号機番号:0
号機番号:1
EtherNet/IPユニット
EtherNet/IPユニット
変数の値書込
WriteDat
変数名’WritingDat’
グローバル変数
ネットワーク公開あり
変数の値読出
ReadDat
変数名’OriginalDat’
グローバル変数
ネットワーク公開あり
メッセージ送信 Identity情報（製品名）読出
RespDat
レスポンス

LD

名称	データ型	初期値	コメント
OperatingEnd	BOOL	False	処理終了
Trigger	BOOL	False	実行条件
Operating	BOOL	False	処理中
WriteDat	INT	1234	書込データ
ReadDat	INT	0	読出データ
RoutePathDat	string [256]	'01\#10\02\192.168.250.2\01\#00'	ルートパス
ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	リクエストパス
RespDat	ARRAY[0..10] OF BYTE	[11(16#0)]	レスポンスデータ
Dummy	BYTE	16#0	ダミー
RS_instance	RS		
CIPUCMMWrite_instance	CIPUCMMWrite		
CIPUCMMRead_instance	CIPUCMMRead		
CIPUCMMSend_instance	CIPUCMMSend		




各命令実行
Operating
CIPUCMMWrite_instance
CIPUCMMWrite
Execute
Done
'RoutePathDat'
RoutePath
Busy
UINT#20
TimeOut
Error
'WritingDat'
DstDat
ErrorID
UINT#1
Size
ErrorIDEx
WriteDat
SrcDat
CIPUCMMRead_instance
CIPUCMMWrite_instance.Done
CIPUCMMRead
Execute
Done
'RoutePathDat'
RoutePath
Busy
UINT#20
TimeOut
Error
'OriginalDat'
SrcDat
ErrorID
UINT#1
Size
ErrorIDEx
DstDat
ReadDat
ReadDat
RcvSize
CIPUCMMRead_instance.Done
@MOVE
EN
ENO
UINT#1
In
Out
ReqPath.ClassID
@MOVE
EN
ENO
UINT#1
In
Out
ReqPath.InstanceID
@MOVE
EN
ENO
TRUE
In
Out
ReqPath.isAttributeID
@MOVE
EN
ENO
UINT#7
In
Out
ReqPath.AttributeID
CIPUCMMSend_instance
CIPUCMMSend
Execute
Done
'RoutePathDat'
RoutePath
Busy
UINT#20
TimeOut
Error
BYTE#16#0E
ServiceCode
ErrorID
ReqPath
RqPath
ErrorIDEx
Dummy
ServiceDat
RespSize
UINT#0
Size
RespServiceDat
RespDat
RespDat


正常終了時処理
Operating
CIPUCMMSend_instance.Done
インラインST
1 //正常終了時処理
2 ;
異常終了時処理
Operating
CIPUCMMWrite_instance.Error
インラインST
1 //異常終了時処理
2 ;
CIPUCMMRead_instance.Error
CIPUCMMSend_instance.Error

ST

内部変数	名称	データ型	初期値	コメント
	Trigger	BOOL	False	実行条件
	DoUCMMTrigger	BOOL	False	処理中
	Stage	INT	0	状態遷移
	WriteDat	INT	1234	書込データ
	ReadDat	INT	0	読出データ
	ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	リクエストパス
	RoutePathDat	string[256]	'01\#10\02\192.168.250.2\01\#00'	ルートパス
	RespDat	ARRAY[0..10] OF BYTE	[11(16#0)]	レスポンスデータ
	Dummy	BYTE	16#0	ダミー
	CIPUCMMWrite_instance	CIPUCMMWrite		
	CIPUCMMRead_instance	CIPUCMMRead		
	CIPUCMMSend_instance	CIPUCMMSend		

外部変数	名称	データ型	コンスタント	コメント
	J01_ETNOnlineSta	BOOL	☑	オンライン

```
// Trigger の立ち上がりでシーケンス開始
IF ((Trigger=TRUE) AND (DoUCMMTrigger=FALSE) AND (J01_ETNOnlineSta=TRUE))
THEN
    DoUCMMTrigger:=TRUE;
    Stage:=INT#1;
    CIPUCMMWrite_instance(
        Execute:=FALSE,        // インスタンス初期化
        SrcDat:=WriteDat);     // ダミー
    CIPUCMMRead_instance(// インスタンス初期化
        Execute:=FALSE,        // ダミー
        DstDat:=ReadDat);      // ダミー
    CIPUCMMSend_instance(
        Execute:=FALSE,        // インスタンス初期化
        ServiceDat:= Dummy,  // ダミー
        RespServiceDat:=RespDat);// ダミー
END_IF;

IF (DoUCMMTrigger=TRUE) THEN
    CASE Stage OF
    1 :                        // 変数書込要求
        CIPUCMMWrite_instance(
        Execute:=TRUE,
        RoutePath :='RoutePathDat',// ルートパス
        TimeOut:=UINT#20,    // タイムアウト時間
        DstDat:='WritingDat',  // 他局変数名
        Size:=UINT#1,          // 書込要素数
        SrcDat:=WriteDat);     // 書込データ

        IF (CIPUCMMWrite_instance.Done=TRUE) THEN
            Stage:=INT#2;      // 正常終了
        ELSIF (CIPUCMMWrite_instance.Error=TRUE) THEN
            Stage:=INT#10;     // 異常終了
        END_IF;
```

```
    2 :                         // 変数読出要求
        CIPUCMMRead_instance(
            Execute:=TRUE,
            RoutePath :='RoutePathDat',// ルートパス
            TimeOut:=UINT#20,// タイムアウト時間
            SrcDat:='OriginalDat',// 他局変数名
            Size:=UINT#1,       // 読出要素数
            DstDat:=ReadDat);// 読出データ

        IF (CIPUCMMRead_instance.Done=TRUE) THEN
            Stage:=INT#3;                   // 正常終了
        ELSIF (CIPUCMMRead_instance.Error=TRUE) THEN
            Stage:=INT#40;                  // 異常終了
        END_IF;

    3 :                         // 任意メッセージ送信
        ReqPath.ClassID:=UINT#01;
        ReqPath.InstanceID:=UINT#01;
        ReqPath.isAttributeID:=TRUE;
        ReqPath.AttributeID:=UINT#07;
        CIPUCMMSend_instance(
            Execute:=TRUE,
            RoutePath :=RoutePathDat,// ルートパス
            TimeOut:=UINT#20,               // タイムアウト値
            ServiceCode:=BYTE#16#0E,    // サービスコード
            RqPath:=ReqPath,                // リクエストパス
            ServiceDat:= Dummy,             // サービスデータ
            Size:=UINT#0,                   // 要素数
            RespServiceDat:=RespDat);       // レスポンスデータ

        IF (CIPUCMMSend_instance.Done=TRUE) THEN
            Stage:=INT#0;                   // 正常終了
        ELSIF (CIPUCMMSend_instance.Error=TRUE) THEN
            Stage:=INT#30;    // 異常終了
        END_IF;

    0:                          // 正常終了処理
        DoUCMMTrigger:=FALSE;
        Trigger:=FALSE;

    ELSE                        // 異常終了処理
        DoUCMMTrigger:=FALSE;
        Trigger:=FALSE;
    END_CASE;
END_IF;
```

8-2-7　サンプルプログラム（CIP コネクション型 (Class3) メッセージで通信する場合）

CIP の Class3 メッセージを使用して、変数書込、変数読出、メッセージ送信を行います。
コントローラどうしは、EtherNet/IP ネットワークで接続しており、相手ノードの IP アドレスは 192.168.250.2 とします。
処理の手順は、以下のようになります。

1 **CIPOpen 命令を使用して、Class3 コネクションを確立します。タイムアウト時間は、2s とします。**

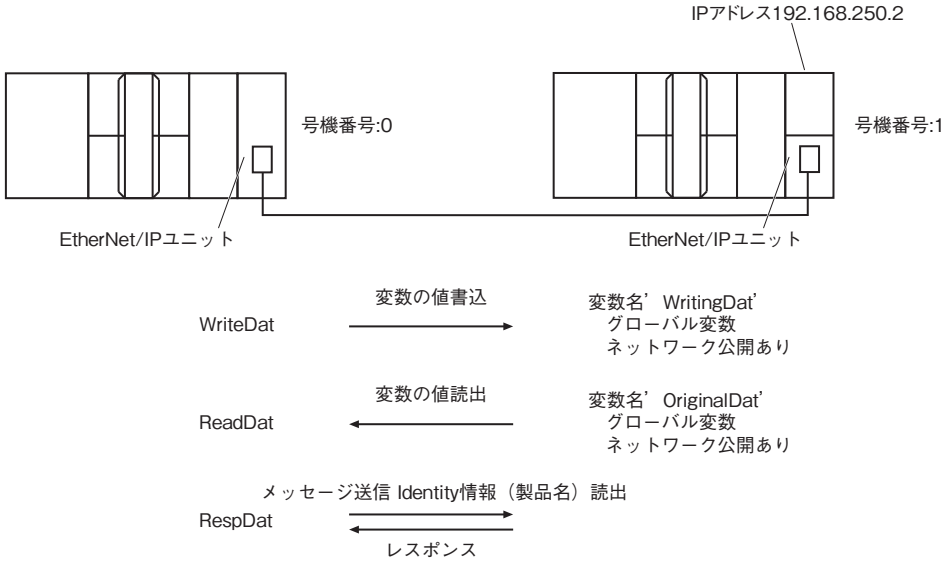
2 **CIPWrite 命令を使用して、相手ノードの変数に値を書き込みます。他局変数名は 'WritingDat'、書き込む値は、変数 WriteDat の中身とします。'WritingDat' は、相手ノードでグローバル変数として定義されており、かつ、ネットワーク公開されていることが必要です。**

3 **CIPRead 命令を使用して、相手ノードの変数の値を読み出します。他局変数名は 'OriginalDat'、読み出した値の格納先は、変数 ReadDat とします。'OriginalDat' は、相手ノードでグローバル変数として定義されており、かつ、ネットワーク公開されていることが必要です。**

4 **CIPSend 命令を使用して、相手ノードに Explicit メッセージを送信します。メッセージの内容は、Identity 情報（製品名）読出です。その際のクラス ID、インスタンス ID、アトリビュート ID、サービスコードは、以下のとおりです。レスポンスデータは、変数 RespDat に格納します。**

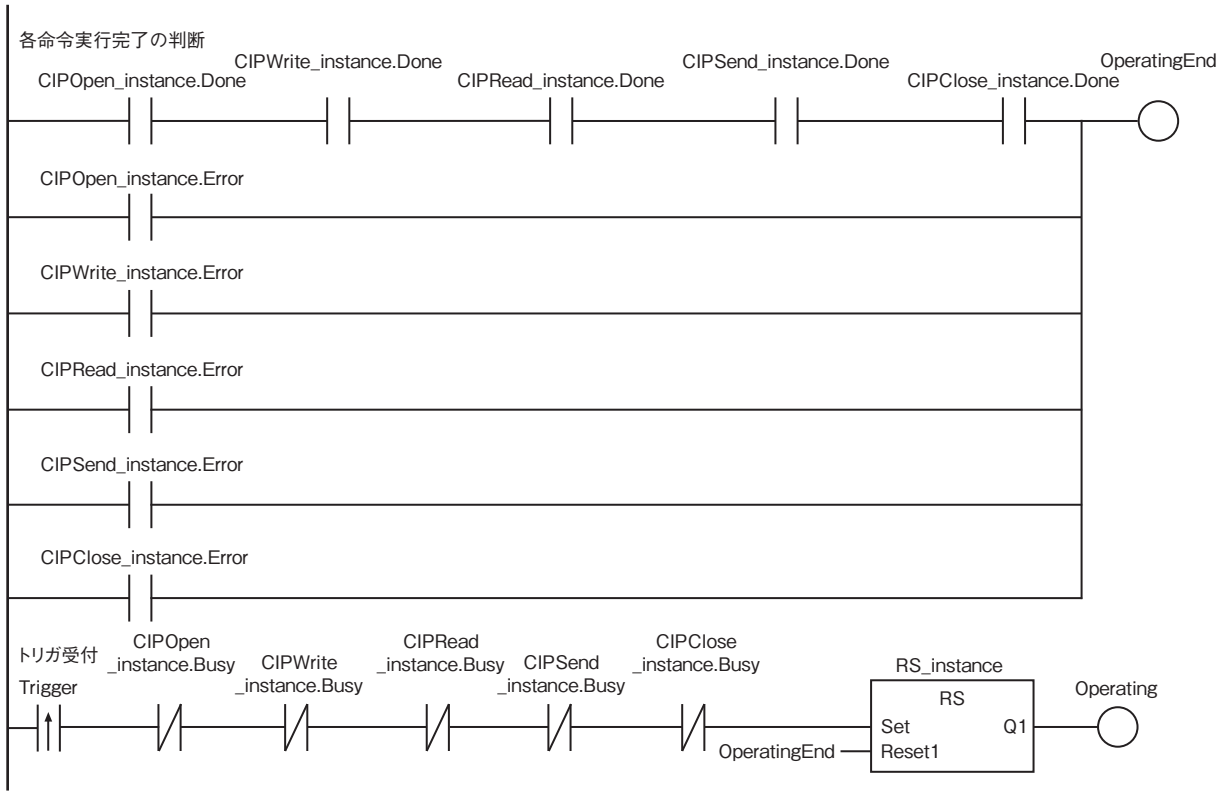
項目	値
クラス ID	1
インスタンス ID	1
アトリビュート ID	7
サービスコード	16#0E

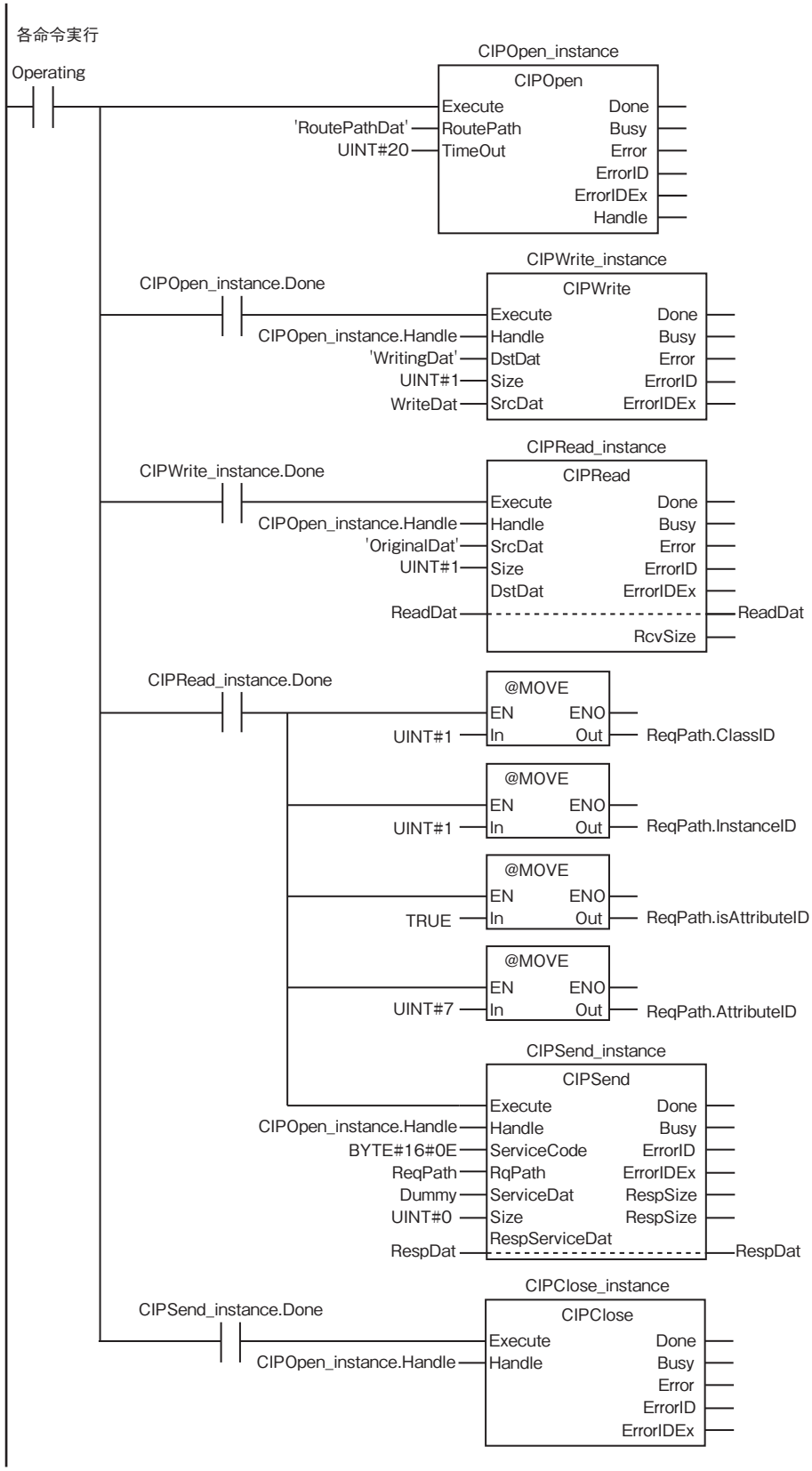
5 **CIPClose 命令を使用して、Class3 コネクションを切断します。**

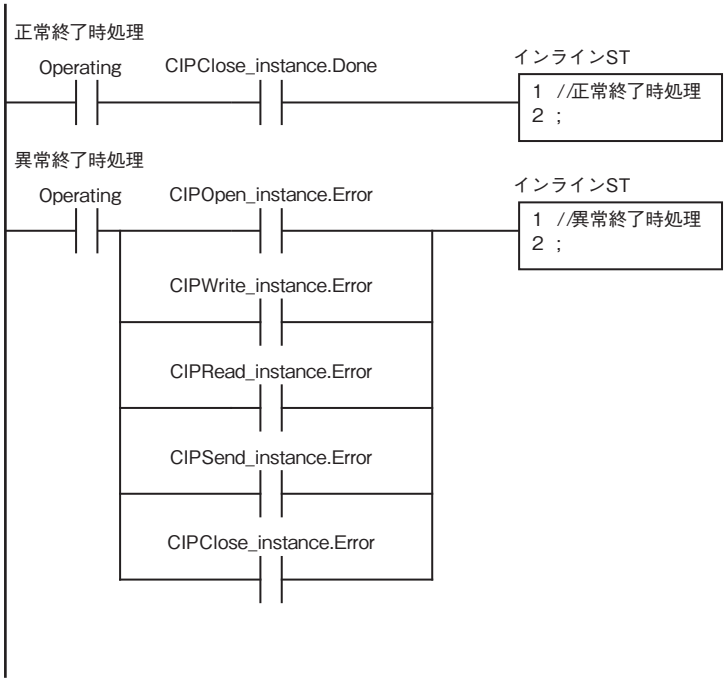


LD

名称	データ型	初期値	コメント
OperatingEnd	BOOL	False	処理終了
Trigger	BOOL	False	実行条件
Operating	BOOL	False	処理中
WriteDat	INT	1234	書込データ
ReadDat	INT	0	読出データ
ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	リクエストパス
RoutePathDat	string[256]	'01\#10\02\192.168.250.2\01\#00'	ルートパス
RespDat	ARRAY[0..10] OF BYTE	[11(16#0)]	レスポンスデータ
Dummy	BYTE	16#0	ダミー
RS_instance	RS		
CIPOpen_instance	CIPOpen		
CIPWrite_instance	CIPWrite		
CIPRead_instance	CIPRead		
CIPSend_instance	CIPSend		
CIPClose_instance	CIPClose		




各命令実行
Operating
CIPOpen_instance
CIPOpen
Execute
'RoutePathDat'
RoutePath
UINT#20
TimeOut
Done
Busy
Error
ErrorID
ErrorIDEx
Handle
CIPWrite_instance
CIPOpen_instance.Done
CIPWrite
Execute
CIPOpen_instance.Handle
Handle
'WritingDat'
DstDat
UINT#1
Size
WriteDat
SrcDat
Done
Busy
Error
ErrorID
ErrorIDEx
CIPRead_instance
CIPWrite_instance.Done
CIPRead
Execute
CIPOpen_instance.Handle
Handle
'OriginalDat'
SrcDat
UINT#1
Size
DstDat
ReadDat
ReadDat
RcvSize
Done
Busy
Error
ErrorID
ErrorIDEx
CIPRead_instance.Done
@MOVE
EN
ENO
UINT#1
In
Out
ReqPath.ClassID
@MOVE
UINT#1
ReqPath.InstanceID
@MOVE
TRUE
ReqPath.isAttributeID
@MOVE
UINT#7
ReqPath.AttributeID
CIPSend_instance
CIPSend
Execute
CIPOpen_instance.Handle
Handle
BYTE#16#0E
ServiceCode
ReqPath
RqPath
Dummy
ServiceDat
UINT#0
Size
RespServiceDat
RespDat
RespDat
Done
Busy
ErrorID
ErrorIDEx
RespSize
RespSize
CIPClose_instance
CIPSend_instance.Done
CIPClose
Execute
CIPOpen_instance.Handle
Handle
Done
Busy
Error
ErrorID
ErrorIDEx


正常終了時処理
Operating
CIPClose_instance.Done
インラインST
1 /正常終了時処理
2 ;
異常終了時処理
Operating
CIPOpen_instance.Error
インラインST
1 /異常終了時処理
2 ;
CIPWrite_instance.Error
CIPRead_instance.Error
CIPSend_instance.Error
CIPClose_instance.Error

ST

内部変数	名称	データ型	初期値	コメント
	Trigger	BOOL	False	実行条件
	DoCIPTrigger	BOOL	False	処理中
	Stage	INT	0	状態遷移
	WriteDat	INT	1234	書込データ
	ReadDat	INT	0	読出データ
	ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	リクエストパス
	RoutePathDat	string[256]	'01\#10\02\192.168.250.2\01\#00'	ルートパス
	RespDat	ARRAY[0..10] OF BYTE	[11(16#0)]	レスポンスデータ
	Dummy	BYTE	16#0	ダミー
	CIPOpen_instance	CIPOpen		
	CIPWrite_instance	CIPWrite		
	CIPRead_instance	CIPRead		
	CIPSend_instance	CIPSend		
	CIPClose_instance	CIPClose		

外部変数	名称	データ型	コンスタント	コメント
	J01_ETNOnlineSta	BOOL	☑	オンライン

```
// Trigger の立ち上がりでシーケンス開始
IF ((Trigger=TRUE) AND (DoCIPTrigger=FALSE) AND (J01_ETNOnlineSta=TRUE))THEN
    DoCIPTrigger:=TRUE;
    Stage:=INT#1;
    CIPOpen_instance(Execute:=FALSE);// インスタンス初期化
    CIPWrite_instance(
        Execute:=FALSE,        // インスタンス初期化
        SrcDat:=WriteDat);     // ダミー
    CIPRead_instance(      // インスタンス初期化
            Execute:=FALSE,    // ダミー
            DstDat:=ReadDat);// ダミー
    CIPSend_instance(
            Execute:=FALSE,    // インスタンス初期化
            ServiceDat:= Dummy,// ダミー
            RespServiceDat:=RespDat);// ダミー
    CIPClose_instance(Execute:=FALSE);// インスタンス初期化
END_IF;
```

```
IF (DoCIPTrigger=TRUE) THEN
    CASE Stage OF
        1 :                     // CIP Class3
                                                                コネクション確立
        CIPOpen_instance(
            Execute:=TRUE,
            TimeOut:=UINT#20,// タイムアウト時間 2.0s
            RoutePath :='RoutePathDat',// ルートパス

        IF (CIPOpen_instance.Done=TRUE) THEN
            Stage:=INT#2;       // 正常終了
        ELSIF (CIPOpen_instance.Error=TRUE) THEN
            Stage:=INT#10;      // 異常終了
        END_IF;

    2 :                         // 変数書込要求
        CIPWrite_instance(
            Execute:=TRUE,
            Handle:=CIPOpen_instance.Handle,// ハンドル
            DstDat:='WritingDat',// 他局変数名
            Size:=UINT#1,       // 書込要素数
            SrcDat:=WriteDat); // 書込データ

        IF (CIPWrite_instance.Done=TRUE) THEN
            Stage:=INT#3;       // 正常終了
        ELSIF (CIPWrite_instance.Error=TRUE) THEN
            Stage:=INT#20;      // 異常終了
        END_IF;

    3 :                         // 変数読出要求
        CIPRead_instance(
            Execute:=TRUE,
            Handle:=CIPOpen_instance.Handle,// ハンドル
            SrcDat:='OriginalDat',// 他局変数名
            Size:=UINT#1,       // 読出要素数
            DstDat:=ReadDat);// 読出データ

        IF (CIPRead_instance.Done=TRUE) THEN
            Stage:=INT#4;                   // 正常終了
        ELSIF (CIPRead_instance.Error=TRUE) THEN
            Stage:=INT#30;                  // 異常終了
        END_IF;
```

```
    4 :                         // 任意メッセージ送信
        ReqPath.ClassID:=UINT#01;
        ReqPath.InstanceID:=UINT#01;
        ReqPath.isAttributeID:=TRUE;
        ReqPath.AttributeID :=UINT#07;
        CIPSend_instance(
            Execute:=TRUE,
            Handle:=CIPOpen_instance.Handle,// ハンドル
            ServiceCode:=BYTE#16#0E,    // サービスコード
            RqPath:=ReqPath,            // リクエストパス
            ServiceDat:=Dummy,          // サービスデータ
            Size:=UINT#0,               // 要素数
            RespServiceDat:=RespDat);   // レスポンスデータ

        IF (CIPSend_instance.Done=TRUE) THEN
            Stage:=INT#5;       // 正常終了
        ELSIF (CIPSend_instance.Error=TRUE) THEN
            Stage:=INT#40;      // 異常終了
        END_IF;

    5 :                         // CIPClass3
                                                    コネクション切断要求
        CIPClose_instance(
            Execute:=TRUE,
            Handle:=CIPOpen_instance.Handle);// ハンドル

        IF (CIPClose_instance.Done=TRUE) THEN
            Stage:=INT#0;
        ELSIF (CIPClose_instance.Error=TRUE) THEN
            Stage:=INT#50;
        END_IF;

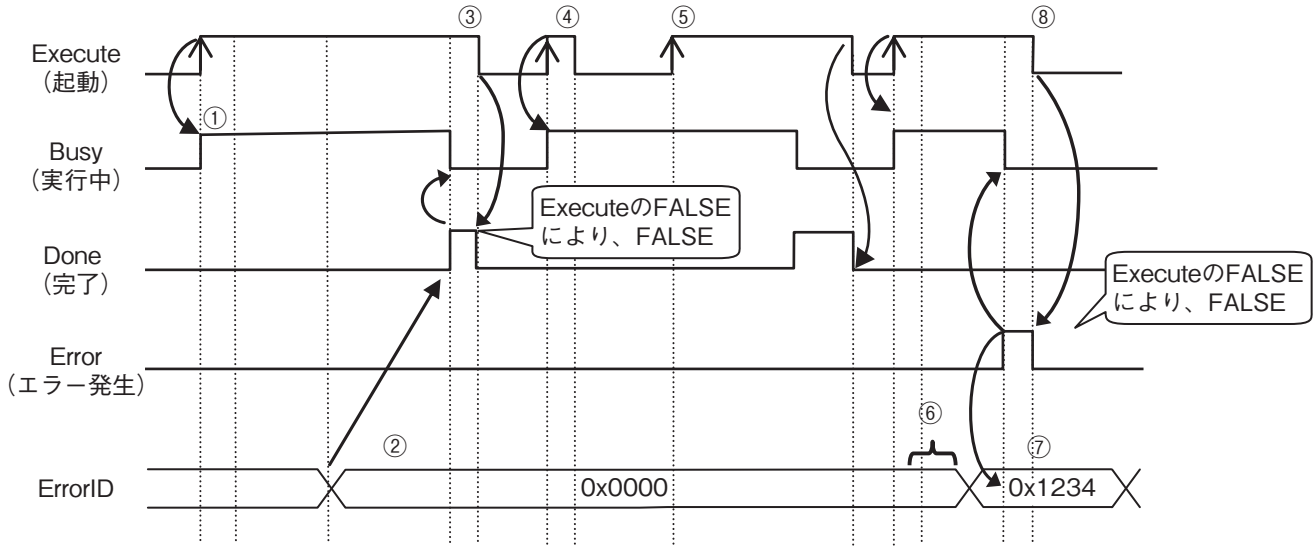
    0:                          // 正常終了処理
        DoCIPTrigger:=FALSE;
        Trigger:=FALSE;

    ELSE                        // 異常終了処理
        DoCIPTrigger:=FALSE;
        Trigger:=FALSE;
    END_CASE;
END_IF;
```

8-2-8　動作タイミング

出力変数の動作とタイミング

各命令の起動から実行完了までの状態は、出力変数の値を監視することで把握できます。
各出力変数の動作とタイミングチャートを以下に示します。

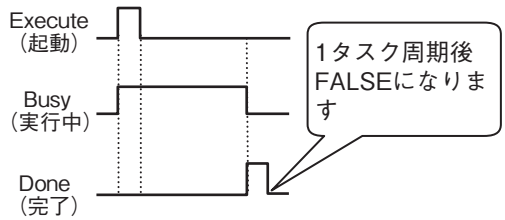


①Execute の立上りで、命令が実行され、Busy が TRUE になります。
②命令の実行結果が、出力変数に反映されると、Done が TRUE、Busy が FALSE になります。
③Execute を FALSE にすると、Done も FALSE に戻ります。
④再び Execute が立ち上がると、Busy が TRUE になります。
⑤命令実行中 (Busy = TRUE) に Execute が立ち上がった場合、無視されます。
⑥エラーが発生すると、内部ではリトライを数回実行します。
　リトライ中は、ErrorID にエラーが更新されません。
⑦通信エラーが発生した場合、Error が TRUE になり ErrorID が確定します。また、Busy と Done が FALSE になります。
⑧Execute を FALSE にすることにより、Error が FALSE になります。

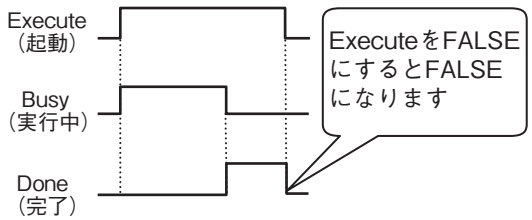
使用上の注意

Done が TRUE になる前に、Execute を FALSE に戻すと、Done は 1 タスク周期の間しか TRUE になりません。(例 1)
Done の TRUE 状態を、ユーザのタイミングによって確認したい場合は、Done が TRUE になるのを確認するまで、Execute を TRUE のまま保持させてください。
Done が TRUE になるまで Execute が TRUE になっていると、Done は Execute が FALSE になるまで、TRUE 状態を保持します。(例 2)

(例1)



(例2)

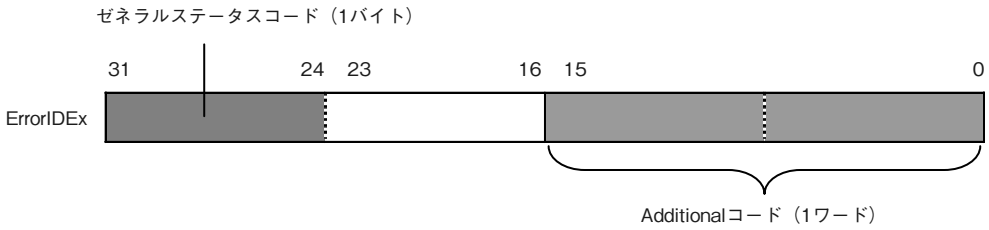


8-2-9　レスポンスコード

CIP メッセージ通信命令実行時にエラーが発生した場合、出力変数の ErrorIDEx に格納されるレスポンスコードについて説明します。

ゼネラルステータスコード

レスポンスコードとして、ゼネラルコードが CIP 通信命令実行終了後、出力変数の ErrorIDEx（DWORD）に格納されます。
また、Additional コードが付加されている場合は、Additional コードも格納されます。



General Status Code (Hex)	Status Name	Description of Status
00	Success	指定されたオブジェクトによるサービスの実行が成功した。
01	Connection failure	サービスに関連するコネクションがコネクションパスの途中で失敗した。
02	Resource unavailable	要求されたサービスは、必要なリソースに空きがなかったため実行できなかった。
03	Invalid parameter value	20Hex を参照。
04	Path segment error	処理ノードは、パスセグメント識別子またはセグメント構文を理解できなかった。Path segment error が発生すると、パス処理は停止する。
05	Path destination unknown	パスが、未知または処理ノードに含まれていないオブジェクトクラス、インスタンス、または構造体エレメントを参照している。Path destination unknown エラーが発生すると、パス処理は停止する。
06	Partial transfer	期待するデータの一部のみが転送された。
07	Connection lost	メッセージコネクションが失われた。
08	Service not supported	要求されたサービスは未サポートだった。または、要求されたサービスは、指定オブジェクトクラス／インスタンスでは未定義だった。
09	Invalid attribute value	要求されたサービスは、アトリビュートデータに異常があった。
0A	Attribute list error	Get_Attribute_List または Set_Attribute_List レスポンス内のアトリビュートが、ゼロ以外のステータスだった。
0B	Already in requested mode/state	指定オブジェクトは、要求されたモード／状態に遷移済みだった。
0C	Object state conflict	指定オブジェクトは、要求されたサービスを実行できる状態になっていなかった。
0D	Object already exists	作成要求されたオブジェクトのインスタンスがすでに存在していた。
0E	Attribute not settable	要求された設定サービスは、変更不可能なアトリビュートを指定した。
0F	Privilege violation	サービス要求元にアクセス権がなかった。
10	Device state conflict	指定デバイスは、要求されたサービスを実行できる状態になっていなかった。
11	Reply data too large	レスポンスデータ長は、処理可能なデータ長を超えていた。
12	Fragmentation of a primitive value	サービスがプリミティブなデータを分割する（例えば、REAL 型データの半分に分ける）処理を特定した。
13	Not enough data	要求されたサービスは、処理を実行するのに十分なデータを提供していなかった。
14	Attribute not supported	要求されたサービスは、未サポートのアトリビュートを指定した。

General Status Code (Hex)	Status Name	Description of Status
15	Too much data	要求されたサービスは、無効なデータまで含んでいた。
16	Object does not exist	要求されたサービスは、未実装オブジェクトを指定した。
17	Service fragmentation sequence not in progress	このサービスの分割がこのデータでは有効ではない。
18	No stored attribute data	このオブジェクトのアトリビュートデータは、このサービスが要求される以前に保存されていなかった。
19	Store operation failure	このオブジェクトのアトリビュートデータは、保存処理中に発生した障害のために保存されなかった。
1A	Routing failure (request packet too large)	サービス要求パケットが宛先までの中継ネットワークのサイズより大きい。中継ノードは強制的に中断した。
1B	Routing failure (response packet too large)	サービス応答パケットが宛先までの中継ネットワークのサイズより大きい。中継ノードは強制的に中断した。
1C	Missing attribute list entry data	サービスが、要求されたビヘイビアを実行するために必要なアトリビュートのリスト内のアトリビュートを供給しなかった。
1D	Invalid attribute value list	サービスは、無効であったアトリビュートに関するステータス情報が付加されたアトリビュートのリストを返す。
1E	Embedded service error	埋め込みサービスがエラーとなった。
1F	Vendor specific error	ベンダー固有のエラーが発生した。エラーレスポンスの Additional Code フィールドには、発生した特定のエラーが示される。このエラーコードを使用するのは、この表やオブジェクトクラス定義内に示されるエラーコードのどれもが、該当エラーに対応していない時に限られる。
20	Invalid parameter	要求されたサービスは、パラメータに異常があった。このコードが使用されるのは、パラメータが本仕様の要件や Application オブジェクト仕様に定義されている要件を満たさないときである。
21	Write-once value or medium already written	既に書き込まれているか、書込不可となっている、1 回のみ書込可能メディア (例えば、WORM ドライブ、PROM など) への書込を試みた。
22	Invalid Reply Received	不正な応答を受信した。(例えば、応答サービスコードが要求サービスコートと不一致である。応答メッセージが最小サイズより小さいなど)。このステータスコードは不正応答のその他の要因で使用される。
23-24		今後の拡張のために CIP によって予約されている。
25	Key Failure in path	パスの最初のセグメントのキーセグメントが宛先モジュールと一致しない。そのオブジェクトはどのキーチェックで失敗したかを示さなければならない。
26	Path Size Invalid	送信された要求サービスのパスのサイズがあるオブジェクトへ中継される要求に許されている大きさに満たないか、大きすぎる。
27	Unexpected attribute in list	この時点では Set 不能なアトリビュートをセットしようとした。
28	Invalid Member ID	要求されたサービスのメンバ ID は、未実装のクラス／インスタンス／アトリビュートを指定した。
29	Member not settable	要求された設定サービスは、変更不可能なメンバを指定した。
2A	Group 2 only server general failure	このエラーコードは、4K 以内のコードスペースしか持たない Group 2 Only サーバにより、Service not supported、Attribute not supported、および Attribute not settable の代替コードとしてのみ出力可能である。
2B-CF		今後の拡張のために CIP によって予約されている。

General Status Code (Hex)	Status Name	Description of Status
D0-FF	Reserved for Object Class and service errors	このエラーコードの範囲は、オブジェクトクラス固有のエラーを示すために使用される。この範囲のコードを使用するのは、この表に示されているエラーコードのどれもが、発生したエラーを正確に説明していないときに限られる。Additional Code フィールドを使用して、General Error Code をさらに詳しく説明することができる。

● General Status が 01 Hex のときの Additional Status の例（Connection Manager Object のステータス）

General Status (Hex)	Additional Status (Hex)	Explanation
01	0100	コネクション使用中、または重複 Forward_Open。
01	0103	サポートされていないトランスポートクラスとトリガの組み合わせ。
01	0106	所有権の不一致
01	0107	ターゲットアプリケーションでコネクションが見つからない。
01	0108	無効なコネクションタイプ。コネクションタイプまたはコネクションの優先順位のいずれかに問題がある。
01	0109	無効なコネクションサイズ。
01	0110	デバイスが設定されていない。
01	0111	RPI がサポートされていない。コネクションタイムアウト乗数、または Production Inhibit Time にも問題がある可能性がある。
01	0113	Connection Manager がこれ以上のコネクションをサポートできない。
01	0114	キーセグメント内のベンダ ID か製品コードのいずれかがデバイスと一致しなかった。
01	0115	キーセグメント内のデバイスタイプがデバイスと一致しなかった。
01	0116	キーセグメント内の Major Revision または Minor Revision
01	0117	無効なコネクションポイント
01	0118	無効な設定フォーマット
01	0119	現在オープンしている制御コネクションがないため、コネクションリクエストが失敗した。
01	011A	ターゲットアプリケーションがこれ以上のコネクションをサポートできない。
01	011B	Production Inhibit Time より RPI が小さい。
01	0203	コネクションがタイムアウトしたため、コネクションを終了できない。
01	0204	レスポンスを待つ間にUnconnected_Send サービスがタイムアウトになった。
01	0205	Unconnected_Send サービスでのパラメータエラー
01	0206	Unconnected メッセージサービスに対するメッセージが大きすぎる
01	0207	返信のない非コネクション型の ACK
01	0301	利用可能なバッファメモリがない
01	0302	データ送信に利用できるネットワーク帯域幅がない
01	0303	利用できるタグフィルタがない
01	0304	リアルタイムのデータを送るように設定されていない
01	0311	ポートセグメント内で指定されたポートが使用できない
01	0312	ポートセグメント内で指定された Link Address が使用できない
01	0315	パス内のセグメントタイプかセグメント値が無効である
01	0316	コネクションクローズ時にパスとコネクションが等しくない
01	0317	セグメントが存在しない。あるいはネットワークセグメントのエンコード値が無効である。
01	0318	自己への Link Address が無効である
01	0319	セカンダリのリソースが使用できない
01	031A	コネクションがすでに確立されている
01	031B	直接コネクションがすでに確立されている
01	031C	その他
01	031D	冗長コネクションが一致しない
01	031E	送信側モジュールにおいて使用できる受信用リソースがこれ以上ない
01	031F	ターゲットパス用のコネクションリソースが存在しない
01	0320 - 07FF	ベンダ固有

8-3　EtherNet/IP ユニット宛 CIP オブジェクトについて

8-3-1　EtherNet/IP ユニット宛 CIP オブジェクトの種類

EtherNet/IP ユニット宛 CIP オブジェクトは、以下の種類があります。

オブジェクト名称	機能内容	参照先
Identity オブジェクト	• CPU ユニットの識別情報の読出し • EtherNet/IP ユニットのリセット	P.8-33
TCP/IP Interface オブジェクト	• TCP/IP 設定を書込み／読出し	P.8-36
Ethernet Link オブジェクト	• Ethernet の設定と読出し • Ethernet スターテスの読出し	P.8-38
PLC オブジェクト	• コントローラのステータス取得 • コントローラの動作モード変更	P.8-41

8-3-2　Identity オブジェクト（クラス ID：01Hex）

CPU ユニットの識別情報の読出しや、EtherNet/IP ユニットをリセットするオブジェクトです。

● クラス ID

01 Hex を指定します。

● インスタンス ID

00 ～ 01 Hex を指定します。

● アトリビュート ID

読み出す情報を指定します。

クラス用アトリビュート ID

オブジェクト全体のアトリビュートを指定します。

アトリビュート ID	パラメータ名	内容	属性	読出しデータ	
				データ型	値
01 Hex	Revision	オブジェクトのリビジョン	読出	UINT	0001 Hex
02 Hex	Max Instance	最大のインスタンス番号	読出	UINT	0001 Hex

インスタンス用アトリビュート ID

インスタンスごとのアトリビュートを指定します。

アトリビュート ID	パラメータ名	内容	属性	読出しデータ	
				データ型	値
01 Hex	Vendor ID	ベンダー ID	読出	UINT	002F Hex
02 Hex	Device Type	デバイスタイプ	読出	UINT	000C Hex
03 Hex	Product Code	製品コード	読出	UINT	0668 Hex
04 Hex	Revision	機器のリビジョン	読出	Struct	－
	Max Instance	メジャーリビジョン	読出	USINT	詳細は下表の（1）を参照
	Revision	マイナーリビジョン	読出	USINT	
05 Hex	Status	EtherNet/IP ユニットのステータス	読出	WORD	詳細は下表の（2）を参照
06 Hex	Serial Number	シリアル番号	読出	UINT	設定値
07 Hex	Product Name	製品名	読出	STRING	設定値

(1) CIP のメジャー・リビジョン、マイナー・リビジョン

ユニットバージョン	CIP リビジョン	
	メジャー・リビジョン	マイナー・リビジョン
ユニット Ver.2.1	02 Hex	04 Hex
ユニット Ver.3.0	03 Hex	01 Hex

(2) EtherNet/IP ユニットのステータス詳細

ビット	名称	説明
0	Owned	EtherNet/IP ユニットがタグデータリンクのターゲットとしてコネクションを開設していることを示す
1	予約	常に FALSE
2	Configured	タグデータリンク設定あり
3	予約	常に FALSE
4 ～ 7	Extended Device Status	EtherNet/IP ユニットの状態を示す b7 / b6 / b5 / b4 0 / 1 / 0 / 1：Major Fault 発生中 0 / 0 / 1 / 0：ターゲットとなっているコネクションの 1 つ以上タイムアウトが発生している。 0 / 0 / 1 / 1：タグデータリンク設定がないことを示す。 0 / 1 / 1 / 0：1 つ以上のコネクションが正常に通信を行っている状態を示す。 0 / 1 / 1 / 1：上記以外
8	Minor Recoverable Fault	次のいずれかの異常が発生している場合に TRUE します • TCP/IP 拡張設定異常 • DNS サーバ接続異常 • タグデータリンク設定異常 • タグデータリンク異常 • タグデータリンクコネクション確立失敗 • FTP サーバ設定異常 • NTP クライアント設定異常 • SNMP 設定異常 • NTP サーバ接続異常 • タグ解決異常
9	Minor Unrecoverable Fault	次の異常が発生している場合 TRUE します • Identity 情報異常

ビット	名称	説明
10	Major Recoverable Fault	次のいずれかの異常が発生している場合 TRUE します • IP アドレス重複異常 • BOOTP サーバ接続異常 • Ethernet 基本設定異常 • TCP/IP 基本設定異常
11	Major Unrecoverable Fault	次のいずれかの異常が発生している場合 TRUE します • 通信コントローラ異常 • MAC アドレス異常
12 ～ 15	予約	常に FALSE

● サービスコード

実行させるサービスを、サービスコードで指定します。

サービスコード	パラメータ名	内容	サポートするサービス範囲	
			クラス単位	インスタンス単位
01 Hex	Get_Attribute_All	アトリビュートの値を読み出します。	○	○
0E Hex	Get_Attribute_Single	指定したアトリビュートの値を読み出します。	○	○
05 Hex	Reset	EtherNet/IP ユニットをリセットします。 IP アドレスの変更などパラメータの設定変更時、EtherNet/IP ユニットをリセットして、変更したパラメータを反映させるときなどに実行してください リセット方法によって、CIPSend 命令の「ServiceDat」に以下のいずれかの値を入力してください。 00 Hex： EtherNet/IP ユニットをリスタートします。 01 Hex： タグデータリンク設定をクリアしてリスタートします。	×	○

● オブジェクトを指定するリクエストパス（IOI）について

オブジェクトを指定するときは、サービスコードごとにリクエストパス（IOI）を以下のように指定します。

サービスコード		クラス ID	インスタンス ID	アトリビュート ID
01 Hex	Get_Attribute_All	01 Hex	• クラス単位でサービスを指定：00 Hex • インスタンス単位でサービスを指定：01 Hex（固定）	不要
0E Hex	Get_Attribute_Single			• クラスのアトリビュートを読み出す場合：01 ～ 02Hex • インスタンスのアトリビュートを読み出す場合：01 ～ 07Hex
05 Hex	Reset		01 Hex（固定）	不要

8-3-3　TCP/IP Interface オブジェクト（クラス ID：F5Hex)

IP アドレスやサブネットマスク、デフォルトゲートウェイなどの設定を書込み／読出しを行うオブジェクトです。

● クラス ID

F5 Hex を指定します。

● インスタンス ID

00 ～ 01 Hex を指定します。

● アトリビュート ID

読み出す情報を指定します。

クラス用アトリビュート ID

オブジェクト全体のアトリビュートを指定します。

アトリビュート ID	パラメータ名	内容	属性	読出しデータ	
				データ型	値
01 Hex	Revision	オブジェクトのリビジョン	読出	UINT	0001 Hex
02 Hex	Max Instance	最大のインスタンス番号	読出	UINT	0001 Hex

インスタンス用アトリビュート ID

インスタンスごとのアトリビュート（値）を指定します。

アトリビュート ID	パラメータ名	内容	属性	書込み／読出しデータ	
				データ型	値
01 Hex	Interface Configuration Status	EtherNet/IP ユニットの IP アドレス設定状態を示します	読出	DWORD	bit0 ～ 3： Interface Configuration Status 0=IP アドレス未設定（BOOTP 起動中含む）状態 1=IP アドレスが設定されている状態 bit4 ～ 31：予約（FALSE 固定）
02 Hex	Configuration Capability	EtherNet/IP ユニットに設定可能なコントローラ構成・設定を示します	読出	DWORD	bit0：BOOTP Client：TRUE(固定) bit1：DNS Client：TRUE（固定） bit2：DHCP Client：FALSE（固定） bit3：DHCP − DNS Update：FALSE（固定） bit4：Configuration Settable：TRUE（固定） bit5 ～ 31：予約（FALSE 固定）
03 Hex	Configuration Control	EtherNet/IP ユニットの起動時の IP アドレスの設定方法を設定します	書込	DWORD	bit0：固定 IP アドレス設定時 bit1：BOOTP 設定時
04 Hex	Physical Link Object	物理層リンクオブジェクトへのパス	読出	Struct	-
	Path size	パスのサイズ（WORD サイズ）		UINT	0002 Hex
	Path	物理層リンクオブジェクトのパス (固定)		EPATH	20 F6 24 01 Hex
05 Hex	Interface Configuration	EtherNet/IP ユニット設定	書込	Struct	-
	IP Address	IP アドレス		UDINT	設定値
	Network Mask	サブネットマスク		UDINT	設定値
	Gateway Address	デフォルトゲートウェイ		UDINT	設定値
	Nama Server	プライマリネームサーバ		UDINT	設定値
	Nama Server2	セカンダリネームサーバ		UDINT	設定値
	Domain Name	ドメイン名		STRING	設定値
06 Hex	Host Name	ホスト名称（予約）	書込	STRING	0000 Hex (固定)

● サービスコード

実行させるサービスを、サービスコードで指定します。

サービスコード	パラメータ名	内容	サポートするサービス範囲	
			クラス単位	インスタンス単位
01 Hex	Get_Attribute_All	アトリビュートの値を読み出します。	○	×
0E Hex	Get_Attribute_Single	指定したアトリビュートの値を読み出します。	○	○
10 Hex	Set_Attribute_Single	指定したアトリビュートに値を書込みます。 アトリビュートに値を書き込み後、EtherNet/IP ユニットは自動的にリスタートします。 リスタート処理がまだ完了していない時に、次の Set_Attribute_Single を実行した場合、一般ステータス “0C Hex” （Object State Conflict）を返信します。	×	○

● オブジェクトを指定するリクエストパス（IOI）について

オブジェクトを指定するときは、サービスコードごとにリクエストパス（IOI）を以下のように指定します。

サービスコード		クラス ID	インスタンス ID	アトリビュート ID
01 Hex	Get_Attribute_All	F5 Hex	• クラス単位でサービスを指定：00 Hex • インスタンス単位でサービスを指定：01 Hex	不要
0E Hex	Get_Attribute_Single			• クラスのアトリビュートを読み出す場合：01 ～ 02Hex • インスタンスのアトリビュートを読み書きする場合：01 ～ 06Hex
10 Hex	Set_Attribute_Single			

8-3-4　Ethernet Link オブジェクト（クラス ID：F6 Hex)

Ethernet 通信の設定・読出し、Ethernet 通信のスターテスを読み出すオブジェクトです。

● クラス ID

F6 Hex を指定します。

● インスタンス ID

00 ～ 01 Hex を指定します。

● アトリビュート ID

読み出す情報を指定します。

クラス用アトリビュート ID

オブジェクト全体のアトリビュートを指定します。

アトリビュート ID	パラメータ名	内容	属性	読出しデータ	
				データ型	値
01 Hex	Revision	オブジェクトのリビジョン	読出	UINT	0002 Hex
02 Hex	Max Instance	最大のインスタンス番号	読出	UINT	0001 Hex

インスタンス用アトリビュート ID

インスタンスごとのアトリビュート（値）を指定します。

アトリビュート ID	パラメータ名	内容	属性	書込み／読出しデータ	
				データ型	値
01 Hex	Interface Speed	EtherNet/IP ユニットの通信速度を示します。	読出	UDINT	現在値を読み出す
02 Hex	Interface Flags	EtherNet/IP ユニットのステータスを示します。	読出	DWORD	詳細は下表 (1) 参照
03 Hex	Physical Address	EtherNet/IP ユニットの MAC アドレスを示します。	読出	ARRAY [0...5] OF USINT	MAC アドレスの現在値を読み出す
04 Hex	Interface Counters	物理層リンクオブジェクトへのパス	読出	Struct	－
	In Octets	インタフェース上で受信したオクテット数 不要マルチキャストパケット、In Discards でカウントされる破棄パケットを含みます。		UDINT	現在値を読み出す
	In Unicast Packets	インタフェース上で受信したユニキャストパケット数。In Discards でカウントされる破棄パケットは含みません。		UDINT	現在値を読み出す
	In NonUnicast Packets	インタフェース上で受信したユニキャスト以外のパケット数 不要マルチキャストパケットを含み、In Discards でカウントされる破棄パケットは含みません。		UDINT	現在値を読み出す
	In Discards	インタフェース上で受信後、破棄された着信パケット数		UDINT	現在値を読み出す
	In Errors	エラーを含む着信パケット数。In Discards に含まれません。		UDINT	現在値を読み出す
	In Unknown Protos	不明のプロトコルを含む着信パケット数		UDINT	現在値を読み出す
	Out Octets	インタフェース上で送信したオクテット数		UDINT	現在値を読み出す
	Out Unicast Packets	インタフェース上で送信したユニキャストパケット数		UDINT	現在値を読み出す

アトリビュート ID	パラメータ名	内容	属性	書込み／読出しデータ	
				データ型	値
04 Hex	Out NonUnicast Packets	インタフェース上で送信したユニキャスト以外のパケット数	読出	UDINT	現在値を読み出す
	Out Discards	破棄された発信パケット数		UDINT	現在値を読み出す
	Out Errors	エラーを含む発信パケット数		UDINT	現在値を読み出す
05 Hex	Media Counters	EtherNet/IP ユニットのメディアカウンタ	読出	Struct	－
	Alignment Errors	長さがオクテット整数でない受信フレーム数		UDINT	現在値を読み出す
	FCS Errors	FCS チェックに合格しない受信フレーム数		UDINT	現在値を読み出す
	Single Collisions	コリジョンが 1 回だけあった送信成功フレーム数		UDINT	現在値を読み出す
	Multiple Collisions	コリジョンが 2 回以上あった送信成功フレーム数		UDINT	現在値を読み出す
	SQE Test Errors	SQE テストエラーメッセージが生成された回数		UDINT	現在値を読み出す
	Deferred Transmissions	媒体が BUSY だったため、最初の送信の試みが遅れたフレーム数		UDINT	現在値を読み出す
	Late Collisions	パケット送信において 512 ビットタイムより後にパケット送信に検出されたコリジョン数		UDINT	現在値を読み出す
	Excessive Collisions	コリジョン過多のため送信に失敗したフレーム数		UDINT	現在値を読み出す
	MAC Transmit Errors	内部の MAC 副層送信エラーのため送信に失敗したフレーム数		UDINT	現在値を読み出す
	Carrier Sense Errors	フレームを送信しようとしたときに、キャリアセンス条件を失った回数、またはアサートしなかった回数		UDINT	現在値を読み出す
	Frame Too Long	最大許容フレームサイズを超えた受信フレーム数		UDINT	現在値を読み出す
	MAC Receive Errors	内部の MAC 副層受信エラーのため、インタフェースで受信に失敗したフレーム数		UDINT	現在値を読み出す
06 Hex	Interface Control	EtherNet/IP ユニットのカウンタ	書込	Struct	－
	Control Bits	Ethernet 通信の Atuto Nego、全 2 重を指定します。		WORD	詳細は下表 (2) 参照
	Forced Interface Speed	Ethernet 通信速度の設定値を示します。		UINT	設定値を読み出す

(1) Interface Flags 詳細

ビット	名称	内容
0	LinkStatus	FALSE：リンクダウン中　TRUE：リンクアップ中
1	Half/FullDuplex	FALSE：半 2 重　　TRUE：全 2 重
2 ～ 4	Negotiation Status	00 Hex：オートネゴシエーション実行中 01 Hex：オートネゴシエーションとスピード検出に失敗 02 Hex：オートネゴシエーション失敗でスピード検出は成功 03 Hex：スピードと 2 重モードのネゴシエーションに成功 04 Hex：オートネゴシエーションを試みてない
5	Manual Setting Requires Speed	FALSE（固定）：変更の反映を自動的に実行可能
6	Local Hardware Fault	FALSE（固定）
7 ～ 31	予約	FALSE（固定）

(2) Control Bits 詳細

ビット	名称	内容
0	Auto-negotiate	FALSE：オートネゴシエーション無効（通信設定が 10Mbps 固定） TRUE：オートネゴシエーション有効（通信設定が 100/10Mbps 自動）
1	ForcedDuplex Mode	FALSE：半 2 重　　TRUE：全 2 重 （注） オートネゴシエーション有効 (bit0:TRUE) のときは、FALSE に固定してください。
2 ～ 16	予約	FALSE（固定）

● サービスコード

実行させるサービスを、サービスコードで指定します。

サービスコード	パラメータ名	内容	サポートするサービス範囲	
			クラス単位	インスタンス単位
0E Hex	Get_Attribute_Single	指定したアトリビュートの値を読み出します。	○	○
10 Hex	Set_Attribute_Single	指定したアトリビュートに値を書込みます。	○	○
4C Hex	Get_and_Clear	Attribute4、あるいは 5 を指定して、アトリビュートの値を 0 クリアします。	×	○

● オブジェクトを指定するリクエストパス（IOI）について

オブジェクトを指定するときは、サービスコードごとにリクエストパス（IOI）を以下のように指定します。

サービスコード		クラス ID	インスタンス ID	アトリビュート ID
0E Hex	Get_Attribute_Single	F6 Hex	• クラス単位でサービスを指定：00 Hex • インスタンス単位でサービスを指定：01 Hex（固定）	不要
10 Hex	Set_Attribute_Single			• クラスのアトリビュートを読み出す場合：01 ～ 02Hex • インスタンスのアトリビューを読み書きする場合：01 ～ 06Hex
4C Hex	Get_and_Clear			値を 0 クリアするアトリビュートを指定：01 ～ 05Hex

8-3-5　PLC オブジェクト（クラス ID：C4Hex)

コントローラのステータス取得や、コントローラの動作モード変更などを行うオブジェクトです。

● クラス ID

C4 Hex を指定します。

● インスタンス ID

00 Hex を指定します。

● アトリビュート ID

読み出す情報を指定します。

クラス用アトリビュート ID

オブジェクト全体のアトリビュート（値）を指定します。

アトリビュート ID	パラメータ名	内容	属性	書込み／読出しデータ	
				データ型	値
01 Hex	Revision	オブジェクトのリビジョン	読出	UINT	0002 Hex（固定）
02 Hex	Max Instance	最大のインスタンス番号	読出	UINT	0001 Hex（固定）
64 Hex	PLC Mode	コントローラ動作モードの読出し、変更が可能です。	書込	UINT	アトリビュート書込み時に指定します。 0000 Hex：プログラムモード時 0004 Hex：運転モード時
65 Hex	PLC Error Status	コントローラの異常発生を示します。運転停止異常／運転継続異常が発生している場合に ON します。	読出	UINT	0000Hex：コントローラ異常発生はしていない 0001 Hex：コントローラ異常発生時
66 Hex	PLC Model	コントローラ型式を示します。サイズ 2Byte + 名称 20Byte 固定で、使用していないエリアはスペースで埋まります。	読出	STRING	

インスタンス用アトリビュート ID

なし

● サービスコード

実行させるサービスを、サービスコードで指定します。

サービスコード	パラメータ名	内容	サポートするサービス範囲	
			クラス単位	インスタンス単位
0E Hex	Get_Attribute_Single	指定したアトリビュートの値を読み出します。	○	×
10 Hex	Set_Attribute_Single	指定したアトリビュートに値を書込みます。	○	×

● オブジェクトを指定するリクエストパス（IOI）について

オブジェクトを指定するときは、サービスコードごとにリクエストパス（IOI）を以下のように指定します。

サービスコード		クラス ID	インスタンス ID	アトリビュート ID
0E Hex	Get_Attribute_Single	C4 Hex	00 Hex	読出し／書込むクラスのアトリビュートを指定する ：01 ～ 02Hex、64 ～ 66Hex
10 Hex	Set_Attribute_Single			

9

FTP サーバ機能

9-1 FTP サーバ機能の概要 9-2
9-1-1 概要 9-2
9-1-2 仕様 9-2

9-2 FTP サーバ機能の詳細 9-3
9-2-1 対象ファイル 9-3
9-2-2 FTP サーバへの接続方法 9-3

9-3 FTP サーバ機能の使用手順 9-5
9-3-1 使用手順 9-5
9-3-2 FTP サーバ機能に必要な設定一覧 9-5

9-4 FTP サーバ機能の使用例 9-6

9-5 FTP コマンドの使用方法 9-7
9-5-1 コマンド一覧 9-7
9-5-2 コマンドの使用方法 9-8

9-6 FTP サーバ機能ステータス 9-13
9-6-1 FTP ステータス 9-13

9-7 SD メモリカード機能の使い方 9-14
9-7-1 SD メモリカードの種類 9-14
9-7-2 ファイルの種類 9-15
9-7-3 SD メモリカードの初期化 9-15
9-7-4 変数データのフォーマット 9-16

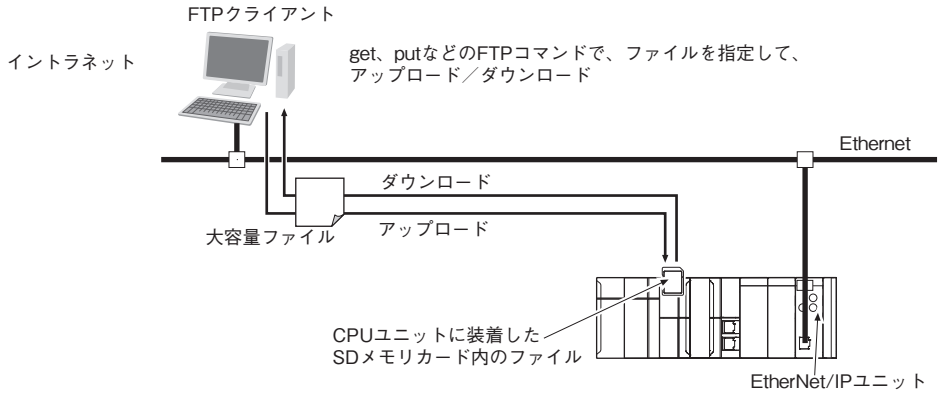
9-8 FTP によるファイル転送時間 9-17

9-9 ホストコンピュータからの使用例 9-18

9-1　FTP サーバ機能の概要

9-1-1　概要

EtherNet/IP ユニットは、FTP（ファイル転送プロトコル：File Transfer Protocol）サーバ機能を内蔵しているため、イントラネット内の Ethernet 上パソコンの FTP クライアントソフトから、FTP コマンドを使用して、SD メモリカード内の大容量ファイルを読み書き（アップロード／ダウンロード）することができます。



9-1-2　仕様

項目	仕様
実行できるコマンド	open　（指定されたホストの FTP サーバと接続する） user　（ユーザ名をリモート FTP サーバに指定する） ls　（リモートホストのファイル名を表示する） dir　（リモートホストのファイル名および詳細情報を表示する） rename　（ファイル名を変更する） mkdir　（リモートホストの作業ディレクトリにディレクトリを作成する） rmdir　（リモートホストの作業ディレクトリにあるディレクトリを削除する） cd　（リモートのホストの作業ディレクトリを指定のリモートディレクトリに変更する） pwd　（リモートのホストの作業ディレクトリを表示する） type　（ファイル転送のタイプを指定されたタイプに変更する） get　（指定されたリモートファイルをローカルホストに転送する） mget　（指定された複数のリモートファイルをローカルホストに転送する） put　（指定されたローカルファイルをリモートホストに転送する） mput　（指定された複数のローカルファイルをリモートホストに転送する） delete　（指定されたリモートホストのファイルを削除する） mdelete　（指定されたリモートホストの複数のファイルを削除する） close　（FTP サーバとの接続を切断する） bye　（FTP クライアントを終了する） quit　（FTP クライアントを終了する）
プロテクト	ログイン名 • 形 CJ1W-EIP21 の場合、半角英数字 1 ～ 12 文字 • 形 CJ1W-EIP21S の場合、半角英数字 1 ～ 16 文字 パスワード（8 文字以内） • 形 CJ1W-EIP21 の場合、半角英数字 1 ～ 8 文字 • 形 CJ1W-EIP21S の場合、半角英数字 8 ～ 16 文字
使用プロトコル	FTP（ポート番号：20/TCP, 21/TCP）
コネクション数	1

9-2　FTP サーバ機能の詳細

9-2-1　対象ファイル

EtherNet/IP ユニットが扱うコントローラ上のファイルシステムは、本体 CPU ユニットの SD メモリカード上に存在します。
以下にそのディレクトリ構造を示します。

/:ルート
└── MEMCARD:SD

接続時のディレクトリは「/」（ルート）ディレクトリです。

参考

- ls、dir、mkdir コマンドをルートディレクトリで実行したときに表示される MEMCARD の作成日付は、そのファイルシステムのボリュームラベルの作成日付です。
- MEMCARD のボリュームラベルが未作成の場合は、ログインした日付が表示されます。

9-2-2　FTP サーバへの接続方法

FTP のログイン名およびパスワードは、ユニット設定（CPU 高機能ユニットシステム設定）に設定されたものを使用します。

参考

汎用の FTP ソフトを使用すると、グラフィカルなユーザインタフェースでエクスプローラのような感覚でファイルの転送、読み出しができます。

● ログイン名とパスワードの設定

FTP のログイン名とパスワードはデフォルトでは設定されていません。
任意のログイン名とパスワードを EtherNet/IP ユニット設定で設定してください。

● ログイン時のメッセージ

状態	メッセージ
正常接続	220 xxx.xx.xx.xx yyyyyyyyyy FTP server （FTP Version z.zz）ready. xxx.xx.xx.xx：EtherNet/IP ユニットの IP アドレス yyyyyyyyyy ：EtherNet/IP ユニットの形式（例：CJ1W-EIP21） z.zz ：EtherNet/IP ユニットのファームウェアのバージョン
FTP サービス中	221 FTP server busy, Goodbye.

● 制約事項

設定条件としての制約事項を以下に示します。

- ログイン名とパスワードに使用できる文字は、半角英数字です。大文字と小文字は区別されます。
- ログイン名
  形 CJ1W-EIP21 の場合、半角英数字 1 ～ 12 文字
  形 CJ1W-EIP21S の場合、半角英数字 1 ～ 16 文字
- パスワード
  形 CJ1W-EIP21 の場合、半角英数字 1 ～ 8 文字
  形 CJ1W-EIP21S の場合、半角英数字 8 ～ 16 文字
- ログイン名を設定したときは必ずパスワードも設定してください。パスワードが設定されていないとログイン名の設定が有効となりません。
- ログイン名を設定しないときや、英数字以外の文字を指定したときは、ログイン名は無効となります。

● FTP のファイル転送モード

FTP のファイル転送モードには、ASCII モードとバイナリモードの 2 種類があります。ファイルの転送を開始する前に、type（データタイプ指定）コマンドを使用して、いずれかの適切なモードを選択してください。

- BIN 形式のファイルを転送する場合：　binary モードを選択
- ASCII 形式のファイルを転送する場合：ascii モードを選択

● 同一ファイルに対する多重アクセスについて

FTP のサーバ機能でアクセスするファイルは、別の FTP サーバの通信コマンドによるアクセスや命令語によって、複数からの同時アクセスが発生する可能性があります。
そのような場合は、排他制御が必要となります。
これは、書き込み途中のファイルに対する読出しや上書き、または読み出し途中のファイルに対する書込みなどを回避するためです。
CPU ユニットは、命令語と命令語の組み合わせの場合のみ、以下のように自動的に排他制御を行います。
それ以外の組み合わせの場合は、ファイル操作用（ファイル名変更、ファイルコピーなど）の命令または通信コマンドを使用して、排他制御を行ってください。

- 同一ファイルに対する SD メモリカードアクセス時の排他制御

<table>
<tr><th colspan="2" rowspan="2">先に実行中のアクセス
後から実行するアクセス</th><th colspan="2">命令語</th><th colspan="2">通信コマンド</th></tr>
<tr><th>読出</th><th>書込</th><th>読出</th><th>書込</th></tr>
<tr><td rowspan="2">命令語</td><td>読出</td><td colspan="2">自動的に排他制御が行われ、読出の命令語がエラーとなります（先勝ち）</td><td>（排他制御は不要です）</td><td rowspan="3">排他制御を行ってください</td></tr>
<tr><td>書込</td><td colspan="2">自動的に排他制御が行われ、書込の命令語がエラーとなります（先勝ち）</td><td>排他制御を行ってください</td></tr>
<tr><td rowspan="2">通信コマンド</td><td>読出</td><td>（排他制御は不要です）</td><td>排他制御を行ってください</td><td>（排他制御は不要です）</td></tr>
<tr><td>書込</td><td colspan="3">排他制御を行ってください</td></tr>
</table>

9-3　FTP サーバ機能の使用手順

9-3-1　使用手順

1 **基本的な設定を行います。**
基本的な操作の流れは、🕮「1-5 EtherNet/IP 通信の使用手順」を参照してください。

2 **Sysmac Studio で FTP サーバ機能の設定をします。(🕮「5-4 ［FTP 設定］ダイアログ」参照)**

3 **Sysmac Studio にて［コントローラ設定］|［EtherNet/IP ユニット設定］を選択します。「FTP」ダイアログにて、以下の設定をします。**
- FTP サーバ
- ポート No.
- ログイン名
- パスワード

4 **CPU ユニットにオンライン接続して、コントローラに設定データを転送します。**

5 **SD メモリカードを CPU ユニットに装着します。**

6 **FTP クライアントソフトから、EtherNet/IP ユニットに接続します。**

7 **EtherNet/IP ユニット設定に設定したログイン名、パスワードを入力して、EtherNet/IP ユニットにログインします。**

8 **コントローラに装着した SD メモリカード内のディレクトリ「MEMCARD」に対して、cd (Change Directory)、get（ファイルの取得）などの、ftp の各コマンドを使用できるようになります。**

9 **接続を解除します。**

9-3-2　FTP サーバ機能に必要な設定一覧

FTP サーバ機能を使用するときは、以下のユニット設定を設定することが必要です。

Sysmac Studio EtherNet/IP ユニット設定のタブ	項目名	設定が必要な条件	参照ページ
FTP	FTP サーバ	必須	🕮P.5-9
	ポート No.	任意 * （注）デフォルト値「21」から変更する場合は必要です。	
	ログイン名	必須 *	
	パスワード	必須 *	

*「FTP サーバ」を「使用しない」に指定した場合は、設定する必要がありません。

参考

FTP サーバ機能を使用する場合の設定は、［FTP 設定］ダイアログで行います。［FTP 設定］ダイアログについての詳細は、🕮「5-4 ［FTP 設定］ダイアログ」を参照してください。

9-4　FTP サーバ機能の使用例

FTP へ接続するときのログイン名を「user1」、パスワードを「password」とするときの FTP サーバ機能の使用例を示します。

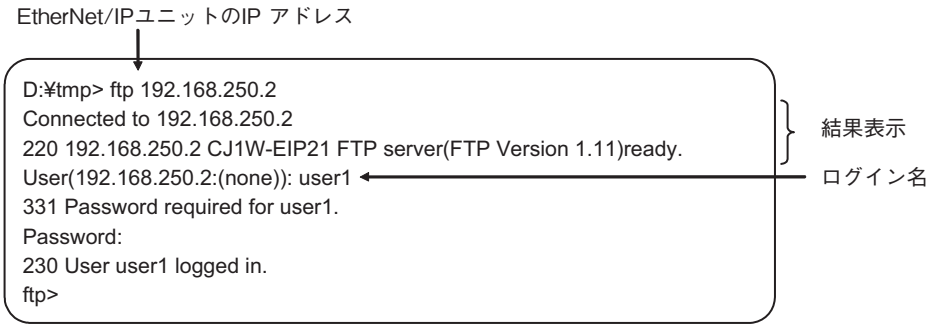
参考

汎用の FTP ソフトを使用すると、グラフィカルなユーザインタフェースでエクスプローラのような感覚でファイルの転送、読み出しができます。

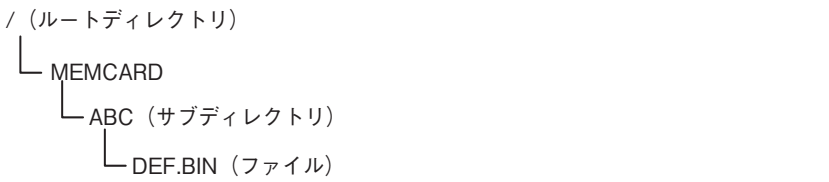
● **手順**

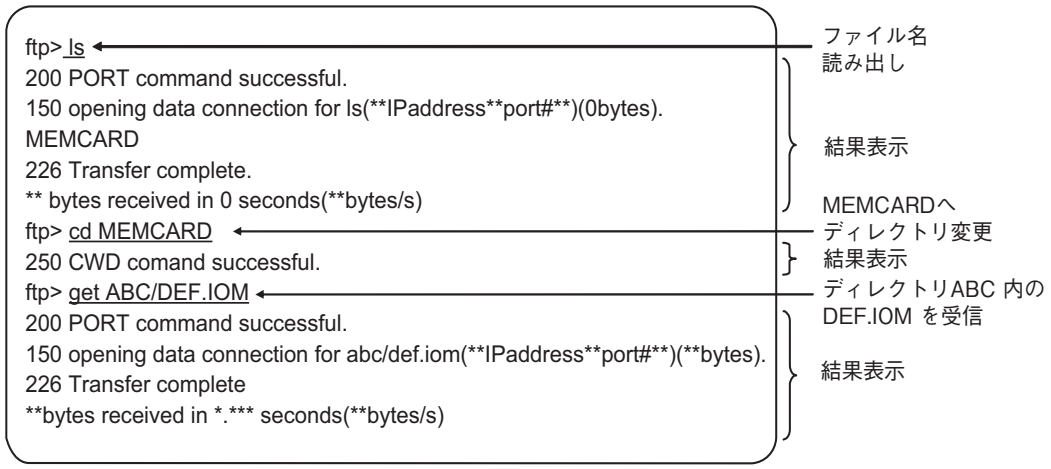
(1) SD メモリカードの装着を確認し、コントローラの電源を ON にします。

(2) 下図の下線部のように入力し、Ethernet 上のコンピュータから FTP サーバに接続します。



(3) ファイルの読み出し、ファイルの受信などの FTP コマンド（下線部）を入力します。ここでは、下図の構成を例に手順を示します。





9-5　FTP コマンドの使用方法

EtherNet/IP ユニットの FTP サーバ機能に対して、ホストコンピュータ（FTP クライアント）側から使用できる FTP コマンドの機能について説明します。
ただし、以下の説明は、ワークステーションの機種によって若干の相違が生じる場合もあります。その機種のマニュアルなどで確認してください。

9-5-1　コマンド一覧

EtherNet/IP ユニットに対して使用できる FTP コマンドは下表のとおりです。

コマンド名	機能
open	指定されたホストの FTP サーバと接続する。
user	ユーザ名をリモート FTP サーバに指定する。
ls	リモートホストのファイル名を表示する。
dir	リモートホストのファイル名および詳細情報を表示する。
rename	ファイル名を変更する。
mkdir	リモートホストの作業ディレクトリにディレクトリを作成する。
rmdir	リモートホストの作業ディレクトリにあるディレクトリを削除する。
cd	リモートのホストの作業ディレクトリを指定のリモートディレクトリに変更する。
pwd	リモートホストの作業ディレクトリを表示する。
type	ファイル転送のタイプを指定されたタイプに変更する
get	指定されたリモートファイルをローカルホストに転送する。
mget	指定された複数のリモートファイルをローカルホストに転送する。
put	指定されたローカルファイルをリモートホストに転送する。
mput	指定された複数のローカルファイルをリモートホストに転送する。
delete	指定されたリモートホストのファイルを削除する。
mdelete	指定されたリモートホストの複数のファイルを削除する。
close	FTP サーバとの接続を切断する。
bye	FTP（クライアント）を終了する。
quit	FTP（クライアント）を終了する。

（注）1 「リモートホスト」とは EtherNet/IP ユニットを示します。
2 「リモートファイル」とは、CPU ユニットの SD メモリカード上のファイルを示します。
3 「ローカルホスト」とは、ホストコンピュータ（FTP クライアント側）を示します。
4 「ローカルファイル」とは、ホストコンピュータ（FTP クライアント側）にあるファイルを示します。

9-5-2　コマンドの使用方法

open

● **書式**

open ［ 接続する FTP サーバの IP アドレスまたはホスト名 ］

● **機能詳細**

・FTP サーバに接続します。通常は、FTP クライアントを起動したときに、FTP サーバの IP アドレスを指定することにより自動実行されます。

user

● **書式**

user ［ ユーザ名 ］

● **機能詳細**

・ユーザ名を指定します。EtherNet/IP ユニットのシステム設定で指定した FTP ログイン名を指定します。

ls

● **書式**

ls ［-l][リモートファイル名 [ローカルファイル名]]

● **機能詳細**

・リモートホスト（SD メモリカード上）のファイル名を表示します。
・スイッチ「-l」を指定すると、ファイルの詳細情報（ファイル名、作成日時、容量）を表示します。省略すると、ファイル名のみを表示します。
・「リモートファイル名」には、SD メモリカード上のファイル名を指定します。
・「ローカルファイル名」を指定すると、ファイルの情報を指定ファイルに格納します。

dir

● **書式**

dir ［ リモートファイル名 [ローカルファイル名]]

● **機能詳細**

・リモートホスト（SD メモリカード上）のファイル名、作成日時、容量を表示します。
・コマンド「ls　-l」と同様の情報を表示します。
・「リモートファイル名」には、SD メモリカード上のファイル名を指定します。
・「ローカルファイル名」を指定すると、ファイルの情報を指定ファイルに格納します。

rename

● 書式

rename　変更元ファイル名　変更後ファイル名

● 機能詳細

- 指定された変更元ファイル名を、変更後ファイル名に変更します。
- 変更後のファイル名がリモートホスト（SD メモリカード上）で既に使用されているときは、ファイル名を変更したファイルで上書きします。
- rename は、ファイル名の変更のみを実行します。ファイルの位置（ディレクトリ）を変更することはできません。

mkdir

● 書式

mkdir　ディレクトリ名

● 機能詳細

- リモートホスト（SD メモリカード上）に、指定された名称のディレクトリを作成します。
- 作業ディレクトリに同名のファイルやディレクトリがすでに存在するときは、エラー終了します。

rmdir

● 書式

rmdir　ディレクトリ名

● 機能詳細

- リモートホスト（SD メモリカード上）にある、指定された名称のディレクトリを削除します。
- 指定されたディレクトリは空である必要があります。
- 指定されたディレクトリが存在しないときや、空でないときは、エラー終了します。

pwd

● 書式

pwd

● 機能詳細

- リモートホストの作業ディレクトリを表示します。

cd

● 書式

cd ［ディレクトリ名］

● 機能詳細

- リモートホストの作業ディレクトリを指定のリモートディレクトリに変更します。
- SD メモリカード上のファイルは、ルートディレクトリ（/）の下の MEMCARD ディレクトリにあります。
- EtherNet/IP ユニットにログインしたときのディレクトリは、ルートディレクトリ（/）です。CPU ユニットに SD メモリカードが装着されていないか、または CPU ユニットの SD メモリカード給電中 LED が消灯しているときは、ディレクトリ MEMCARD は存在しません。

type

● 書式

type　データタイプ

● 機能詳細

- ファイルのデータタイプを指定します。
- サポートしているデータタイプは、次のとおりです。
  ascii　　：ファイルのデータタイプが ASCII タイプとして転送されます。
  binary（image）：ファイルのデータタイプがバイナリデータとして転送されます。
  CPU ユニットでバイナリファイルを取り扱う場合、ファイルを読み出し / 書き込みする前に、必ず type コマンドで binary に指定してください。ASCII タイプのまま転送すると、ファイルの中身は保証されません。
- デフォルトは ascii です。

get

● 書式

get　ファイル名 ［受信ファイル名］

● 機能詳細

- 指定されたリモートファイル（SD メモリカード上のファイル）をローカルホストに転送します。
- 「受信ファイル名」を指定することにより、ローカルホストで受信するファイル名を指定することができます。

mget

● 書式

mget　ファイル名

● 機能詳細

・「ファイル名」にワイルドカード「*」を含めることにより、該当する複数のリモートファイル（SD メモリカード上のファイル）をローカルホストに転送します。

put

● 書式

put　ファイル名　［ 送信先ファイル名 ］

● 機能詳細

・ 指定されたローカルファイルをリモートホスト（SD メモリカード）に転送します。
・「送信先ファイル名」を指定すると、送信したファイルがその名前で保存されます。
・ リモートホスト（SD メモリカード上）に同名のファイルがすでに存在するときは上書きされます。
・ 転送途中でエラーが発生すると、転送中のファイルは消去され、エラー終了します。

mput

● 書式

mput　ファイル名

● 機能詳細

・「ファイル名」にワイルドカード「*」を含めることにより、該当する複数のローカルファイルをリモートホスト（SD メモリカード）に転送します。
・ リモートホスト（SD メモリカード上）に同名のファイルがすでに存在するときは上書きされます。
・ 転送途中でエラーが発生すると、転送中のファイルは消去され、そのファイルの転送処理はエラー終了します。しかし、mput 処理は継続され、残りのファイルが転送されます。

delete

● 書式

delete　ファイル名

● 機能詳細

・ 指定されたリモートファイル（SD メモリカード上のファイル）を削除します。

mdelete

● **書式**

mdelete　ファイル名

● **機能詳細**

・「ファイル名」にワイルドカード「*」を含めることで、該当する複数のリモートファイルを削除します。

close

● **書式**

close

● **機能詳細**

・ 接続している EtherNet/IP ユニットの FTP サーバとの接続を切断します。

bye

● **書式**

bye

● **機能詳細**

・ FTP を終了します。

quit

● **書式**

quit

● **機能詳細**

・ FTP を終了します。

9-6　FTP サーバ機能ステータス

9-6-1　FTP ステータス

FTP サーバが動作中かどうかのステータスを、以下の CJ ユニットデバイス変数により確認できます。

CJ ユニットデバイス変数名	型	R/W	名称	機能
*_CommSta2	WORD	R	通信ステータス 2	ビット 00: FTP ステータス ビット 14: リンクステータス
*_FTPSta	BOOL	R	FTP ステータス	TRUE:　FTP サーバ動作中です。 （FTP クライアント接続中） FALSE:　FTP サーバ待機中です。 （クライアント接続待ち）

使用上の注意

FTP が動作中のときは、SD メモリカード上のファイル操作が行われていることがあります。SD メモリカードを取り外したり、PLC 本体の電源を OFF にしたりしないでください。

参考

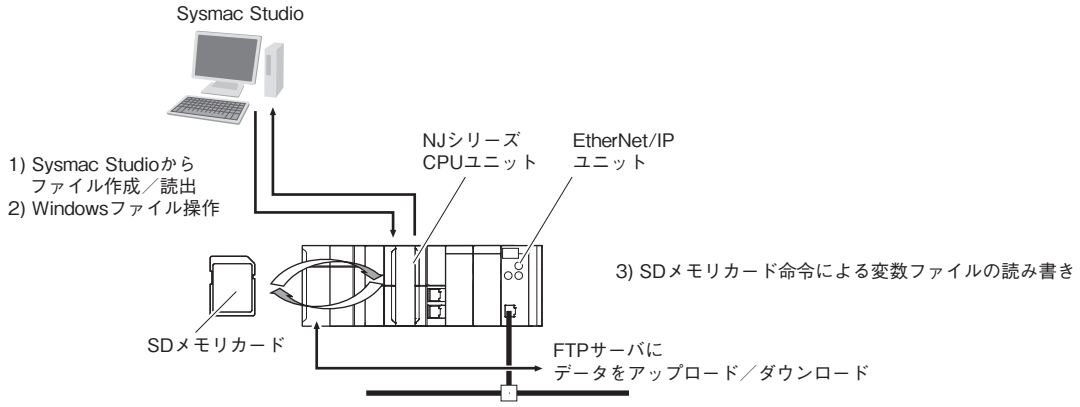
PLC 本体のプログラム中でファイル処理命令（データファイル読み出し / 書き込みなど）を実行する場合は、FTP ステータスを排他制御に使用し、同じ対象に同時にアクセスしないようにしてください。

9-7　SD メモリカード機能の使い方

EtherNet/IP ユニットでは、NJ シリーズ CPU ユニットに装着した SD メモリカードに格納された次のデータを FTP サーバにアップロードやダウンロードできます。

- 変数ファイル（バイナリ形式）

CPU は次の 3 つの方法で SD メモリカードにデータを格納したり読み出します。



9-7-1　SD メモリカードの種類

次に示す SD メモリカード、SDHC カードを使用してください。
それ以外の SD メモリカード、SDHC カードを使用した場合の動作は保証できません。

形式	カード種別	容量 [GB]	フォーマット	書き換え回数	質量
形 HMC-SD292	SD カード	2	FAT16	5 万回	2g 以下
形 HMC-SD492	SDHC カード	4	FAT32	10 万回	
形 HMC-SD1A2		16		10 万回	

9-7-2　ファイルの種類

ファイル名

ファイルにはファイル名と拡張子を付けることによって、識別します。
ファイル名には、以下の半角文字を使用できます。
指定するファイル名の大文字、小文字は問いません。（小文字にしても大文字になります。）
「A」～「Z」、「a」～「z」の英文字、「0」～「9」の数字、
記号「$」、「%」、「'」、「-」、「_」、「@ 」、「!」、「`」、「(」、「)」、「~」、「#」、「&」、「^」、「{」、「}」、「{」、「}」、「;」

ファイル名には、以下の文字は使用できません。
「/」、「\」、「?」,「*」、「¥」、「:」、「<」、「>」、「=」、「+」、「,」、「.」、空白、マルチバイト文字など

ファイル名の長さは、8 文字および、拡張子 3 文字までです。
文字列中、最初にあらわれる「.」が、ファイル名と拡張子の区切り文字として扱われます。
拡張子は、ファイルの種類により決められています。

ディレクトリ

SD メモリカード内のファイルの格納位置にディレクトリを指定して、ディレクトリを最大（ルートディレクトリを含めて）5 階層まで作成することが可能です。
ただし、ディレクトリ名の最大文字数は 65 文字です。

CPU ユニットで取り扱うファイル名

CPU ユニットで取り扱う（読み書き可能な）ファイル名は、以下のように設定されています。

ファイル種類	ファイル名	拡張子	内容	説明
変数ファイル（バイナリ形式）	🕮「9-7-2 ファイルの種類」参照	.bin	指定変数	指定した複数の変数（配列、構造体を含む）の値を、バイナリ形式（.bin）で記載したファイルです。

詳細は、🕮『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-377) を参照してください。

9-7-3　SD メモリカードの初期化

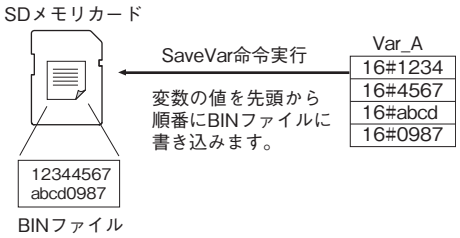
1 **CPU ユニットに SD メモリカードを装着**

2 **周辺ツールで SD メモリカードを初期化**

9-7-4　変数データのフォーマット

BIN 形式

CPU ユニットのラダー命令（変数ファイル読出 FileReadVar、変数ファイル保存 FileWriteVar）で指定するデータ形式（BIN）のファイルフォーマットです。
配列や構造体も読出、保存できます。
変数 Var_A のデータを BIN 形式の添付ファイルとした場合、以下の図のように、データを生成します。



参考

- NJ CPU でバイナリファイルを取り扱う場合、FTP でファイルの読み出し／書き込みする前に、必ず type コマンドでデータタイプを binary に指定してください。（🕮「9-5-2 コマンドの使用方法」参照）
- ラダーのファイル処理命令の使用方法については、🕮『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）を参照してください。

9-8　FTP によるファイル転送時間

FTP によるファイル転送は、ファイルの容量によっては数十分かかることもあります。
以下におおよその転送時間を示します。参考にしてください。

例）CPU ユニットの動作モード：運転モード、タスク周期：1ms での転送時間

処理の種類	転送サイズ	転送時間
put 処理転送長（バイト）	1k	2.0
	5k	10.1
	10k	19.6
	40k	79.7
get 処理転送長（バイト）	1k	10.3
	5k	51.4
	10k	107.0
	40k	409.7

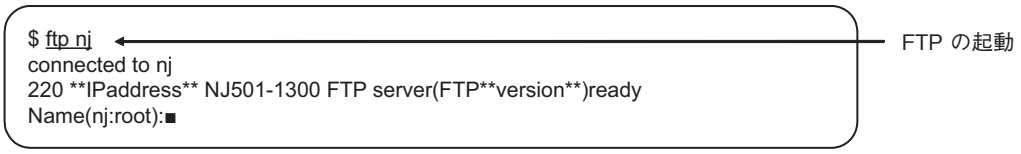
（単位：秒）

9-9　ホストコンピュータからの使用例

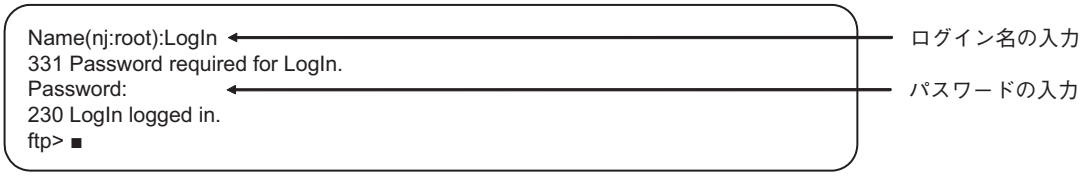
ホストコンピュータから次の手順で操作します。この例では次のように設定されているものとします。
- ホストコンピュータの hosts 内に EtherNet/IP ユニットの IP アドレスがホスト名「nj」として登録されていることを想定しています。
- FTP ログイン名は「LogIn」です。
- CPU ユニットの SD メモリカード上に製造結果のデータがファイル名「RESULT.BIN」で格納されています。
- ワークステーション上に製造指示データがファイル名「PLAN. BIN」で格納されています。

以下の操作では、CPU ユニットの SD メモリカード上の製造結果ファイル「RESULT.BIN」をワークステーションに転送し、次にワークステーション上の製造指示ファイル「PLAN.BIN」を CPU ユニットの SD メモリカードに転送します。
画面の下線部は、FTP クライアント側からのキー入力を示します。また、「$」はワークステーションのプロンプトを、「■」はカーソルを表します。

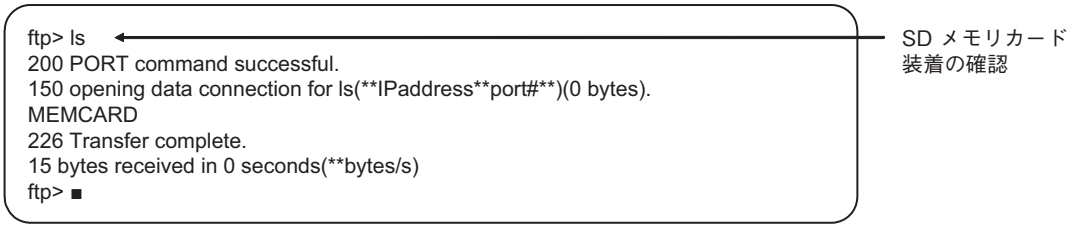
(1)　FTP を起動し、EtherNet/IP ユニットに接続します。



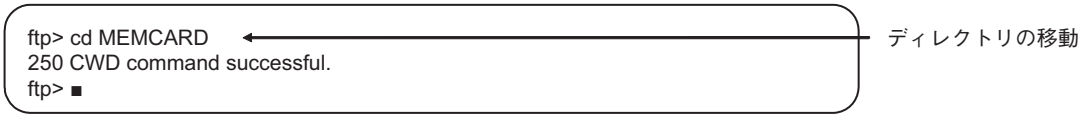
(2)　ログイン名を入力します。



(3)　SD メモリカードが正しく装着されていることを確認します。SD メモリカードの装着が認識されると、ディレクトリ MEMCARD が表示されます。



(4)　MEMARD1 ディレクトリに移動します。



(5)　ファイルのデータタイプをバイナリデータにします。

```
ftp> type binary
200 Type set to I.
ftp>■
```

バイナリデータの指定

(6)　ファイル名「RESULT.BIN」を読み出して、ワークステーションに転送します。

```
ftp> get RESULT.BIN
200 PORT command successful.
150 opening data connection for result.bin (**IPaddress**port#**) (**bytes).
226 Transfer complete.
** bytes received in *.*** seconds (**bytes/s)
ftp> ■
```

ファイルの読出し

(7)　ファイル名「PLAN.BIN」を SD メモリカードに書き込みます。

```
ftp> put PLAN.BIN
200 PORT command successful.
150 opening data connection for plan.bin (**IPaddress**port#**) .
226 Transfer complete.
** bytes received in *.** seconds (**bytes/s)
ftp> ■
```

ファイルの書込み

(8)　FTP を終了します。

```
ftp> bye
221 Goodbye.
$ ■
```

FTP の終了

10

時計情報自動調整機能

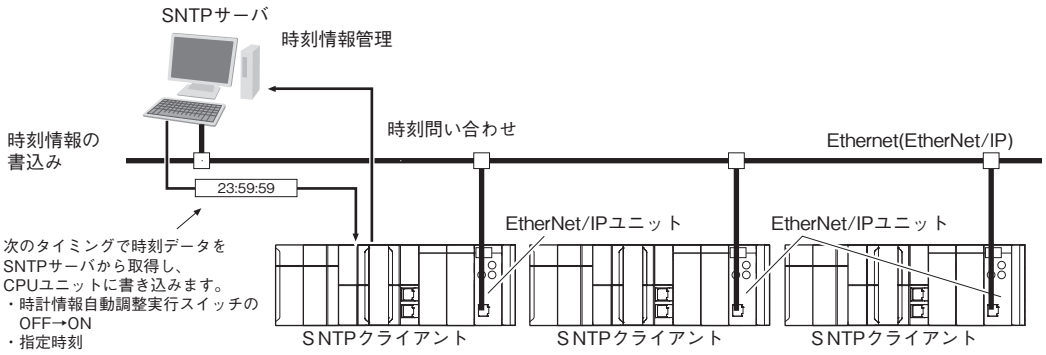
10-1 時計情報自動調整機能の概要 10-2
10-1-1 時計情報自動調整機能とは 10-2
10-1-2 仕様 10-2
10-2 時計情報自動調整機能の使用手順 10-3
10-2-1 使用手順 10-3
10-2-2 時計情報自動調整機能に必要な設定一覧 10-3
10-2-3 時計情報の更新 10-4

10-1 時計情報自動調整機能の概要

10-1-1 時計情報自動調整機能とは

EtherNet/IP ユニットは、指定した時刻、または、時計情報自動調整実行スイッチの OFF → ON で SNTP サーバから時間情報を取得し、自身が装着されている CPU ユニットの内部時計情報を更新することができます。

（注）SNTP(Simple Network Time Protocol) サーバは、LAN 上の時間を管理するサーバです。



- SNTP プロトコルの仕様上、2036 年 2 月 7 日以降は、自動調整できなくなります。
  EtherNet/IP ユニットでは、2036 年 2 月 7 日以降は、本機能は動作しなくなります（エラー情報として表示されません）。

10-1-2 仕様

項目	仕様	
プロトコル	SNTP	
ポート No.	123（UDP） ただし、Sysmac Studio の EtherNet/IP ユニット設定で変更できます。	
SNTP サーバへのアクセス	SNTP サーバ→自 CPU ユニットへ、時計情報書込	ネットワーク内に設置された SNTP サーバから時計情報を取得して、取得した情報を自 CPU ユニットへ反映する
SNTP 動作タイミング	SNTP 機能を使用すると、次のタイミングで時計情報が自動更新されます。 • 時計情報自動調整実行スイッチ OFF → ON 時 • 指定時刻	

10-2 時計情報自動調整機能の使用手順

10-2-1 使用手順

1 **基本的な設定を行います。**
基本的な操作の流れは 📖「1-5 EtherNet/IP 通信の使用手順」を参照してください。

2 **Sysmac Studio の［構成・設定］の高機能ユニット設定で、以下の設定をします。**
［SNTP 設定］ダイアログにて以下の設定をします。
- SNTP サーバの設定（必須）
- SNTP サーバへの指定時刻

3 **手動で時計情報自動調整機能を実行する場合は、時計情報自動調整実行スイッチ（*_AdjTmCmd）を FALSE → TRUE します。**

4 **［コントローラ］|［同期］メニューを選択します。EtherNet/IP ユニット設定が CPU ユニットに転送されます。**

10-2-2 時計情報自動調整機能に必要な設定一覧

時計情報自動調整機能を使用するときは、次に示す Sysmac Studio の EtherNet/IP ユニット設定を行います。

タブ	項目名		設定が必要な条件	参照ページ
SNTP	SNTP サーバの時計情報		必須	📖P.5-10
	ポート No.		任意 （注）　デフォルト値「123」から変更する場合は必要です。	
	サーバ指定方法		必須	
		IP アドレス	どちらか一方の設定が必須 （「サーバ指定方法」に従って設定します。）	
		ホスト名		
	指定時刻		必須	
	タイムアウト時間		任意 （注）　デフォルト値「10 秒」から変更するときには設定が必要です。	
	時差調整		任意 （注）　デフォルト値「+0：0」から変更するときには設定が必要です。	

参考

時計情報自動調整機能を使用する場合の設定は、［SNTP 設定］ダイアログで行います。［SNTP 設定］ダイアログについての詳細は、📖「5-5 ［SNTP 設定］ダイアログ」を参照してください。

10-2-3　時計情報の更新

ユーザプログラム上で、次のデバイス変数を FALSE → TRUE にすると、EtherNet/IP ユニットは、ネットワーク上の SNTP サーバから時計情報を取得し、取得した時計情報を自 CPU ユニットへ反映します。
反映終了後、自動的に TRUE → FALSE に戻ります。

- 「*_AdjTmCmd」（時計情報用自動調整実行スイッチ）

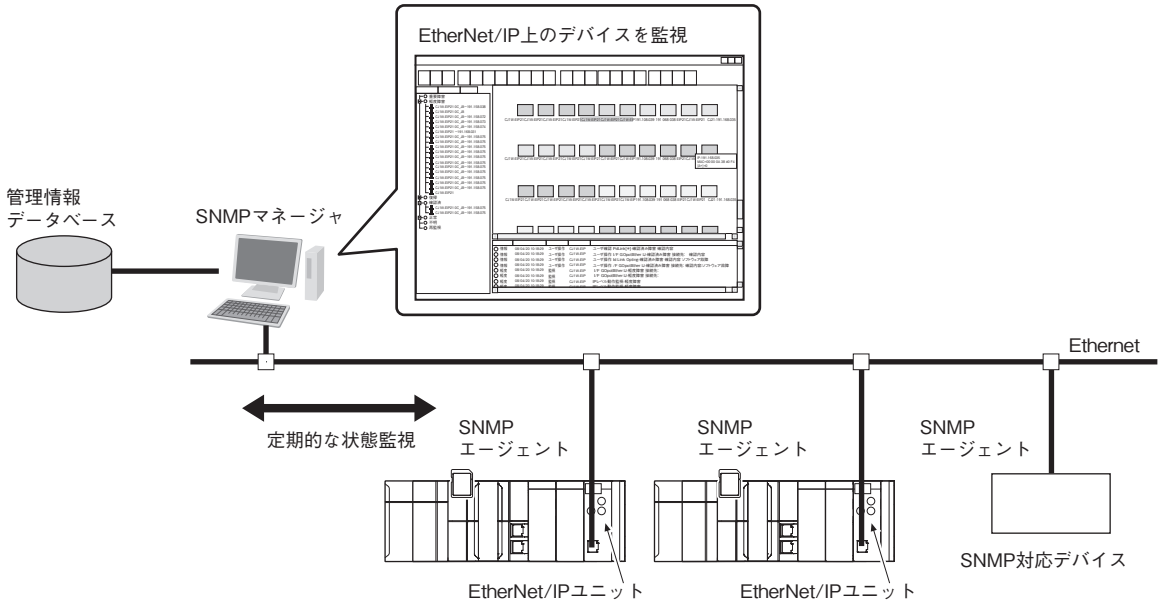
11

SNMP（エージェント）機能

11-1 SNMP（エージェント）機能の概要 11-2
11-1-1 SNMP（エージェント）機能とは 11-2
11-1-2 仕様 11-3
11-1-3 SNMP メッセージ 11-3
11-1-4 MIB の仕様 11-4
11-2 SNMP（エージェント）機能の使用手順 11-17
11-2-1 使用手順 11-17
11-2-2 SNMP（エージェント）機能に必要な設定一覧 11-17

11-1 SNMP（エージェント）機能の概要

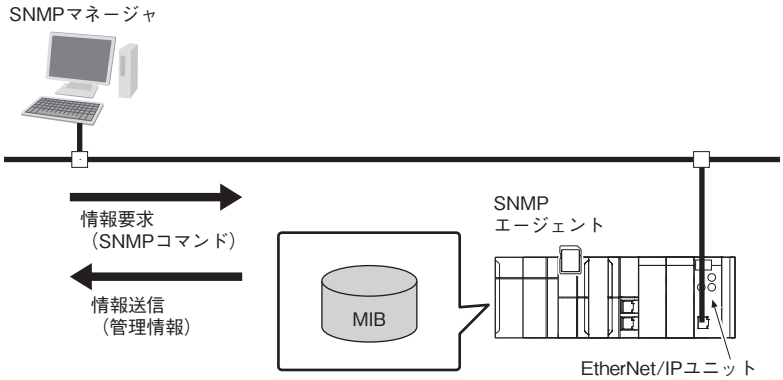
SNMP（simple network management protocol）は、ネットワーク管理プロトコルです。
SNMP をサポートしているネットワーク機器で構成されたネットワークの管理が可能です。
ネットワークを管理する側のサーバを SNMP マネージャ、管理される側のネットワーク機器を SNMP エージェントといいます



11-1-1 SNMP（エージェント）機能とは

SNMP エージェント機能

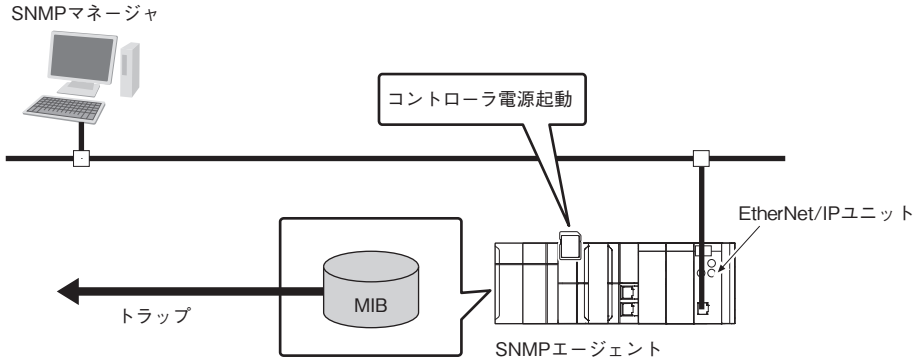
EtherNet/IP ユニットは、MIB（Management Information Base）という自身に関する管理情報を持ち、それを SNMP マネージャに提供することができます。
SNMP マネージャとは、SNMP ネットワークの各デバイス情報を収集・加工し、ネットワーク管理者に提供するソフトウェアです。
この SNMP マネージャにより、EtherNet/IP ユニットの監視を行うことができます。



SNMP マネージャには、MIB 情報を要求するための SNMP コマンドがあります。
EtherNet/IP ユニットの SNMP エージェント機能では、SNMPv1（RFC1157）、SNMPv2C（RFC1901）をサポートしています。
SNMP マネージャを使用し EtherNet/IP ユニットを管理する場合は、SNMPv1、SNMPv2C プロトコルで使用してください。なお、SNMPv1、SNMPv2C をそれぞれ同時に使用することもできます。

SNMP トラップ機能

障害などの特定の状況発生時に、トラップ（Trap）と呼ばれる状況通知レポートを送信します。
SNMP マネージャは、トラップを受信することで定期的に EtherNet/IP ユニットを監視しなくても、状況変化を知ることができます。
ただし，トラップは UDP を使用しているため、EtherNet/IP ポートからのトラップが、SNMP マネージャに受信されたかどうかの確認ができません。
そのため，ネットワークの状態によっては、トラップがSNMP マネージャに到達しない場合があります。

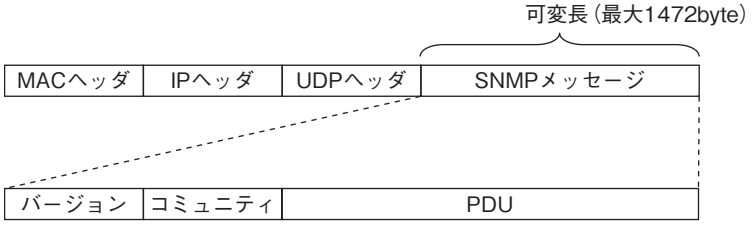


11-1-2　仕様

項目	仕様
プロトコル	SNMP
エージェント	SNMPv1、SNMPv2c
MIB	MIB-II
ポート No	SNMP エージェント：161（UDP） SNMP トラップ：162（UDP） ただし、Sysmac Studio の EtherNet/IP ユニット設定で変更できます。
SNMP トラップ動作タイミング	次のタイミングで状況通知レポートを SNMP マネージャに送信します。 • コントローラ電源投入時 • リンク確立時 • SNMP エージェントの承認失敗時
サポートする MIB コマンド	GetRequest/GetNextRequest

11-1-3　SNMP メッセージ

SNMP メッセージの構造は次のとおりです。



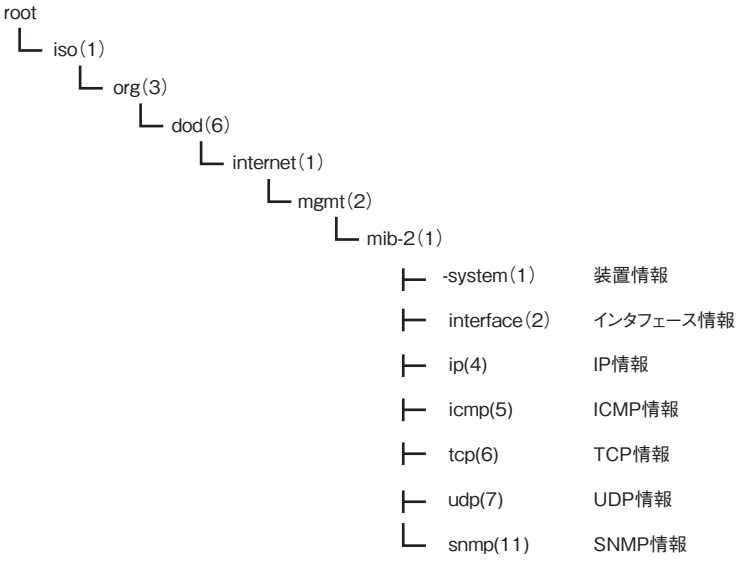
項目	設定値
バージョン	SNMP のバージョンを示す値。 SNMPv1　：0 SNMpv2c　：1
コミュニティ	認証用のコミュニティ名
PDU	PDU タイプによって異なる。

11-1-4　MIB の仕様

EtherNet/IP ユニットがサポートしている MIB の仕様を説明します。

MIB 体系図

EtherNet/IP ユニットの MIB の構造は、以下のようなツリー構造になっています。



MIB グループ一覧

MIB グループ			格納される情報
標準 MIB	system グループ		装置に関する情報の MIB です。
	interfaces グループ		インタフェースに関する情報の MIB です。
	ip グループ	ip	IP 情報の MIB です。
		ipAddrTable	IP アドレスに関するアドレッシングテーブル情報の MIB です。
		ipRouteTable	IP ルーティングテーブルに関する情報の MIB です。
		ipNetToMediaTable	IP アドレス変換テーブルに関する情報の MIB です。
		ipForward	IP フォワーディングテーブルに関する情報の MIB です。
	icmp グループ		ICMP 情報の MIB です。
	tcp グループ	tcp	TCP 情報の MIB です。
	udp グループ	udp	UDP 情報の MIB です。
	snmp グループ	snmp	SNMP 情報の MIB です。

MIB オブジェクトの詳細説明

● system グループ

サブツリー名	規格［(識別子) 属性］	サポート	実装仕様
sysDescr	[(1) RO] 機器の情報 (ハードウェア、OS、ソフトウェアの名称とバージョンを含む) ASCII 文字のみ使用可能	○	"OMRON Corporation " + 形名
sysObjectID	[(2) RO] ベンダの OID。 プライベート MIB のどこにこの機器の情報を割り当てたかを示す。	○	1.3.6.1.4.1.16838.1.1025.2
sysUpTime	[(3) RO] システムが起動してからの時間（1/100 秒単位）	○	規格の通り
sysContact	[(4) RW] 管理者への連絡方法とその人の情報。	○	ユーザ設定
sysName	[(5) RW] 管理用の名前。機器のフルドメイン名を設定する。	○	ユニット名称
sysLocation	[(6) RW] 物理的な場所。	○	ユーザ設定
sysServices	[(7) RO] 提供するサービスの値。	○	64

● interfaces グループ

サブツリー名	規格［(識別子) 属性］	サポート	実装仕様
ifNumber	[(1) RO] ネットワークインタフェースの数	○	1
ifTable	[(2) NA] インタフェースエンティティのテーブル	--	--
ifEntry	[(1) NA] インタフェース情報の行データ INDEX は「ifIndex」	--	--
ifIndex	[(1) RO] このインタフェースを識別するための番号	○	1
ifDescr	[(2) RO] インタフェースに関する情報（メーカー名、製品名、ハードウェアインタフェースのバージョンを含む）	○	10/100M Fast Ethernet Port
ifType	[(3) RO] プロトコルスタックのネットワーク層のすぐ下の物理／リンク層プロトコルに従って区別したインタフェースのタイプ	○	ethernet-csmacd(6)
ifMtu	[(4) RO] MTU 値 このインタフェースで送受信できるデータグラムの最大サイズ (オクテット)	○	1500
ifSpeed	[(5) RO] 推定帯域幅 帯域が変化しない、正確な値が得られない場合は公称値を設定する。	○	10000000

サブツリー名			規格［(識別子) 属性］	サポート	実装仕様
		ifPhysAddress	[(6) RO] MAC アドレス このインタフェースのネットワークレイヤ直下の物理アドレス	○	EtherNet/IP ポートの MAC アドレス
		ifAdminStatus	[(7) RW] インタフェースの望ましい状態 testing 状態では通常のパケットは送出することができない。 up(1) down(2) testing(3)	○	規格の通り
		ifOperStatus	[(8) RO] インタフェースの現在の状態 testing 状態では通常のパケットは送出することができない。 up(1) down(2) testing(3)	○	規格の通り
		ifLastChange	[(9) RO] このインタフェースの ifOperStatus が最後に変化した時の sysUpTime（1/100 秒単位）	○	規格の通り
		ifInOctets	[(10) RO] このインタフェースで受信したオクテットの数。フレーミングキャラクタも含む。	○	規格の通り
		ifInUcastPkts	[(11) RO] 上位プロトコルへ通知したユニキャスト・パケットの数	○	規格の通り
		ifInNUcastPkts	[(12) RO] 上位プロトコルへ通知した非ユニキャスト・パケット（ブロードキャスト，マルチキャストパケット）の数	○	規格の通り
		ifInDiscards	[(13) RO] パケット自身にはエラーはないが，上位プロトコルに渡すことのできなかったパケットの数（バッファ不足などで破棄された受信パケットの数）	○	規格の通り
		ifInErrors	[(14) RO] パケット中のエラーが含まれていることによって破棄されたパケットの数	○	規格の通り
		ifInUnknown Protos	[(15) RO] 不正なプロトコルか未サポートのプロトコルのパケットを受信し、破棄したパケットの数 例えば、Ethernetpacket の中の、上位のプロトコルを識別するフィールドが IP となって いない場合など。	○	規格の通り
		ifOutOctets	[(16) RO] このインタフェースで送信したパケットのオクテットの数 フレーミングキャラクタも含む。	○	規格の通り
		ifOutUcast Pkts	[(17) RO] 上位プロトコルが送信したユニキャスト・パケットの数 破棄パケット、未送信パケット数も含む。	○	規格の通り
		ifOutNUcast Pkts	[(18) RO] 上位プロトコルが送信した非ユニキャスト・パケットの数 破棄パケット、未送信パケット数も含む。	○	規格の通り

サブツリー名			規格［(識別子) 属性］	サポート	実装仕様
		ifOutDiscards	[(19) RO] パケット自身にエラーはないが，送信処理で破棄されたパケットの数（送信バッファ不足など)。	○	規格の通り
		ifOutErrors	[(20) RO] エラーが原因で送信できなかったパケットの数	○	規格の通り
		ifOutQLen	[(21) RO] 送信パケットキューのサイズ（パケット数で表す）	○	0 固定
		ifSpecific	[(22) RO] インタフェースのメディア特有の MIB の参照を表すオブジェクト ID 例としてイーサネットの場合にはイーサネットを定義した MIB のオブジェクト ID を設定する。もし情報がない場合は { 0.0 } を指定する。	○	0.0

● ip グループ（ip）

サブツリー名	規格［(識別子) 属性］	サポート	実装仕様
ipForwarding	[(1) RW] ゲートウェイとして動作するかどうか。 IP ゲートウェイは、データグラムの転送を行うが、IP ホストは、ソースルーティングを除いて転送を行わない。いくつかのノードはどちらかの値しかとらない。よって、マネージャからこのオブジェクトを変更しようとした場合、badValue エラーの応答を返す。 forwarding(1) not-forwarding(2)	○	not-forwarding(2)
IpDefaultTTL	[(2) RW] トランスポート層のプロトコルで TTL の値が与えられなかった場合に、IP ヘッダの TTL に設定されるデフォルト値。	○	64
IpInReceives	[(3) RO] エラーも含む、インタフェースに到着した全ての IP データグラムの数。	○	規格の通り
IpInHdrErrors	[(4) RO] IP ヘッダのエラーのために破棄された受信データグラムの数（チェックサムエラー、バージョン番号エラー、フォーマットエラー、TTL エラー、IP オプションエラーなど）	○	規格の通り
IpInAddrErrors	[(5) RO] IP ヘッダ中の宛先アドレスが有効ではないために破棄されたパケット数。	○	規格の通り
ipForwDatagrams	[(6) RO] 最終宛て先に転送された IP データグラムの数。もし、このノードが IP ゲートウェイとして動作しない場合には、この数はソースルーティングで転送に成功した数になる	○	規格の通り
ipInUnknownProtos	[(7) RO] 受信した自分宛て IP パケットで認識できないプロトコルかサポートしていないプロトコルであるため，破棄した IP データグラムの数。	○	規格の通り
ipInDiscards	[(8) RO] 以後の処理を続けるのに問題はないが、捨てられた IP データグラムの数 (例えば、バッファスペースが足りなかったりした時)	○	規格の通り

サブツリー名	規格［(識別子) 属性］	サポート	実装仕様
ipInDelivers	[(9) RO] IP ユーザプロトコル (ICMP を含む上位プロトコル) に配送されたデータグラムの数。	○	規格の通り
ipOutRequests	[(10) RO] ローカル IP ユーザプロトコル (ICMP を含む上位プロトコル) が IP データグラムの送信要求を実行した数。このカウンタは、ipForwDatagrams を含まない。	○	規格の通り
ipOutDiscards	[(11) RO] 送出するのに問題はないが、捨てられた (例えば、バッファスペースが足りなかったりした時)IP データグラムの数。	○	規格の通り
ipOutNoRoutes	[(12) RO] 送信経路がないため破棄された IP データグラムの数。このカウンタは、ipForwDatagrams で転送しようとした IP データグラムのうち no-route のため破棄されたデータグラムも含む。この値は、デフォルトゲートウェイがダウンしているために転送できないデータグラムの数になる。	○	規格の通り
ipReasmTimeout	[(13) RO] フラグメントされた IP データグラムを受信した場合、リアセンブリを行うため全 IP データグラムの受信を待つ時間の最大秒数。	○	60 sec
ipReasmReqds	[(14) RO] リアセンブリする必要のある受信 IP データグラムの数。IP ヘッダにフラグメントされているかどうかのフラグがあり、それで識別することが可能。	○	規格の通り
ipReasmOKs	[(15) RO] リアセンブリが成功した受信 IP データグラムの数。	○	規格の通り
ipReasmFails	[(16) RO] リアセンブリが失敗した受信 IP データグラムの数。	○	規格の通り
ipFragOKs	[(17) RO] フラグメントが成功した IP データグラムの数。	○	規格の通り
ipFragFails	[(18) RO] フラグメントを失敗した IP データグラムの数。(例えば、IP データグラムの Don'tFlagment フラグセットされているなど)	○	規格の通り
ipFragCreates	[(19) RO] フラグメンテーションの結果として生成された IP データグラムの数。	○	規格の通り
ipAddrTable	[(20) NA] IP アドレス別のアドレス情報テーブル	--	--
ipAddrEntry	[(1) NA] IP アドレス別のアドレス情報の行データ。 INDEX は「ipAdEntAddr」。	--	--
ipAdEntAddr	[(1) RO] IP アドレス。	○	規格の通り
ipAdEntIfIndex	[(2) RO] このエントリが適用するインタフェースのインデックス値。ifIndex と同じ値。	○	規格の通り
ipAdEntNetMask	[(3) RO] このエントリの IP アドレスに関するサブネットマスク。	○	規格の通り

サブツリー名	規格［(識別子) 属性］	サポート	実装仕様
ipAdEntBcastAddr	[(4) RO] IP ブロードキャスト送信時のアドレスの最下位ビットの値。インターネットスタンダードでは全ビット 1 のアドレスがブロードキャストとして使用される。その場合この値は 1 となる。	○	規格の通り
ipAdEntReasmMaxSize	[(5) RO] このインタフェースで受信した IP フラグメント分割された入力 IP データグラムからリアセンブリできる最大 IP パケットのサイズ。	○	規格の通り
ipRouteTable	[(21) NA] このエンティティの IP ルーティングテーブル。	--	--
ipRouteEntry	[(1) NA] 特定のあて先別の経路情報。INDEX は「ipRouteDest」。	--	--
ipRouteDest	[(1) RW] この経路のあて先 IP アドレス。このエントリーが 0.0.0.0 の場合は、デフォルトルートを示す。	○	規格の通り
ipRouteIfIndex	[(2) RW] このルートの次の宛て先ホストに送信するためのインタフェースの識別番号。この識別番号は、インタフェースを特定するために ifIndex と同じ番号。	○	規格の通り
ipRouteMetric1	[(3) RW] このルートのプライマリールーティングメトリック。この値は、ipRouteProto にて指定されるプロトコルに従って決められる。このメトリックが使用されない場合は ”-1” を設定する (ipRouteMetric2 ～ 4 も同様)	○	規格の通り
ipRouteMetric2	[(4) RW] この経路の代替ルーティングメトリック。	○	規格の通り
ipRouteMetric3	[(5) RW] この経路の代替ルーティングメトリック。	○	規格の通り
ipRouteMetric4	[(6) RW] この経路の代替ルーティングメトリック。	○	規格の通り
ipRouteNextHop	[(7) RW] この経路で次にホップする IP アドレス (ブロードキャスト・メディアで接続されたルートの場合、そのインタフェース上のエージェントアドレス・アドレスになる)。	○	規格の通り
ipRouteType	[(8) RW] 経路のタイプ。 other(1)：　以下のいずれでない場合 invalid(2)：　不正なルート direct(3)：　直接接続 indirect(4)：　間接接続 (LOCAL に接続されていない)	○	規格の通り

サブツリー名			規格［(識別子) 属性］	サポート	実装仕様
		ipRouteProto	[(9) RO] どのように経路を決定するかのルーティングメカニズム。ゲートウェイルーティングプロトコルに対応する値があるが、ホストがそれらのプロトコルをサポートすることを意味しているわけではないので注意。 other(1): 下記以外 local(2): ローカルマシン上で設定されたルート netmgmt(3): ネットワーク管理機能により設定された。 icmp(4): ICMP の Redirect 等により設定された。 egp(5): EGP 以下はゲートウェイプロトコル ggp(6): GGP hello(7): ハロー rip(8): RIP is-is(9) es-is(10) ciscoIgrp(11) bbnSpfIgp(12) ospf(13): OSPF bgp(14)	○	規格の通り
		ipRouteAge	[(10) RW] この経路が更新されてからの経過時間（秒単位）	○	規格の通り
		ipRouteMask	[(11) RW] ipRouteDest に関するサブネットマスク値。 任意のサブネットマスク値をサポートしていないシステムではこの値は、ipRouteDest フィールドのアドレスクラスにより決まる。 ipRouteDest が 0.0.0.0 の場合はこの値も 0.0.0.0 とする。	○	規格の通り
		ipRouteMetric5	[(12) RW] 代替ルーティングメトリック。	○	規格の通り
		ipRouteInfo	[(13) RO] このルートで使用するルーティングプロトコルについての MIB オブジェクト ID。定義されていなければ {0.0} を設定する。	○	0.0
ipNetToMediaTable			[(22) NA] IP アドレスから物理アドレスへのマッピングに使用される IP アドレス変換テーブル。	--	--
	ipNetToMediaEntry		[(1) NA] 変換テーブルの行データ。INDEX は「ipNetToMediaIfIndex」「ipNetToMediaNetAddress」	--	--
		ipNetToMediaIfIndex	[(1) RW] このエントリーのインタフェース識別番号。この値は、ifIndex の値を使用	○	規格の通り
		ipNetToMediaPhysAddress	[(2) RW] メディアに依存した物理アドレス。	○	規格の通り
		ipNetToMediaNetAddress	[(3) RW] メディアに依存した物理アドレスに対応する IP アドレス。	○	規格の通り

サブツリー名			規格［(識別子) 属性］	サポート	実装仕様
		ipNetToMediaType	[(4) RW] アドレス変換の方式。 other(1)：　下記以外の方式 invalid(2)：　無効な値 dynamic(3)：動的変換 static(4)：　静的変換	○	規格の通り
ipRoutingDiscards			[(23) RO] 有効だが放棄されたルーティングエントリーの数。例えば、他のルーティングエントリのためのバッファスペースが足りなくなった場合など。	○	規格の通り

● ip グループ（icmp）

名　前	規格［(識別子) 属性］	サポート	実装仕様
icmpInMsgs	[(1) RO] 受信した ICMP メッセージ総数 icmpInErrors でカウントされるものも含む	○	規格の通り
icmpInErrors	[(2) RO] 受信した ICMP メッセージエラーの数。(チェックサムエラー、フレーム長エラーなど)	○	規格の通り
icmpInDestUnreachs	[(3) RO] 受信した Destination Unreachable メッセージの数	○	規格の通り
icmpInTimeExcds	[(4) RO] 受信した Time Exceed メッセージの数	○	規格の通り
icmpInParmProbs	[(5) RO] 受信した Parameter Problem メッセージの数	○	規格の通り
icmpInSrcQuenchs	[(6) RO] 受信した Source Quench メッセージの数	○	規格の通り
icmpInRedirects	[(7) RO] 受信した Redirect メッセージの数	○	規格の通り
icmpInEchos	[(8) RO] 受信した Echo(request) メッセージの数	○	規格の通り
icmpInEchoReps	[(9) RO] 受信した Echo Reply メッセージの数	○	規格の通り
icmpInTimestamps	[(10) RO] 受信した Timestamp メッセージの数	○	規格の通り
icmpInTimestampReps	[(11) RO] 受信した Timestamp Reply メッセージの数	○	規格の通り
icmpInAddrMasks	[(12) RO] 受信した Address Mask Request メッセージの数	○	規格の通り
icmpInAddrMaskReps	[(13) RO] 受信した Address Mask Reply メッセージの数	○	規格の通り
icmpOutMsgs	[(14) RO] 送信した ICMP メッセージの総数。icmpOutErrors でカウントされるものも含む	○	規格の通り
icmpOutErrors	[(15) RO] エラーによって送信されなかった ICMP メッセージ数	○	規格の通り
icmpOutDestUnreachs	[(16) RO] 送信した Destination Unreachable メッセージの数	○	規格の通り
icmpOutTimeExcds	[(17) RO] 送信した Time Exceeded メッセージの数	○	規格の通り
icmpOutParmProbs	[(18) RO] 送信した Parameter Problem メッセージの数	○	規格の通り

名　前	規格［(識別子) 属性］	サポート	実装仕様
icmpOutSrcQuenchs	[(19) RO] 送信した Source Quench メッセージの数	○	規格の通り
icmpOutRedirects	[(20) RO] 送信した Redirect メッセージの数	○	規格の通り
icmpOutEchos	[(21) RO] 送信した Echo(request) メッセージの数	○	規格の通り
icmpOutEchoReps	[(22) RO] 送信した Echo Reply メッセージの数	○	規格の通り
icmpOutTimestamps	[(23) RO] 送信した Timestamp メッセージの数	○	規格の通り
icmpOutTimestampReps	[(24) RO] 送信した Timestamp Reply メッセージの数	○	規格の通り
icmpOutAddrMasks	[(25) RO] 送信した Address Mask Request メッセージの数	○	規格の通り
icmpOutAddrMaskReps	[(26) RO] 送信した Address Mask Reply メッセージの数	○	規格の通り

● ip グループ（tcp）

名　前	規格［(識別子) 属性］	サポート	実装仕様
tcpRtoAlgorithm	[(1) RO] 再送に使用するタイムアウト時間を決定するアルゴリズム。 other(1): 下記以外 constant(2): 固定 RTO 値 rsre(3): MIL-STD-1778 で規定されているアルゴリズム vanj(4): VanJacobson のアルゴリズム	○	規格の通り
tcpRtoMin	[(2) RO] 再送タイムアウトの最小値（1/100 秒単位） この値は再送タイムアウトを決めるアルゴリズムに依存	○	規格の通り
tcpRtoMax	[(3) RO] 再送タイムアウトの最大値（1/100 秒単位） この値は再送タイムアウトを決めるアルゴリズムに依存	○	規格の通り
tcpMaxConn	[(4) RO] サポートできる TCP コネクションの総数。最大コネクション数が動的である場合、-1 とする。	○	規格の通り
tcpActiveOpens	[(5) RO] TCP コネクションが CLOSE 状態から SYN-SENT 状態に直接状態遷移した回数。（能動的なコネクション確立）	○	規格の通り
tcpPassiveOpens	[(6) RO] TCP コネクションが LISTEN 状態から SYN-RCVD 状態に直接状態遷移した回数。（受動的なコネクション確立）	○	規格の通り
tcpAttemptFails	[(7) RO] TCP コネクションが SYN-SENT 状態かまたは SYN-RCVD 状態から CLOSE 状態に直接状態遷移した回数と、SYN-RCVD 状態から LISTEN 状態に直接状態遷移した回数の合計。	○	規格の通り
tcpEstabResets	[(8) RO] TCP コネクションが ESTABLISHD 状態かまたは CLOSE-WAIT 状態から CLOSE 状態に直接状態遷移した回数。	○	規格の通り

名　前	規格［(識別子) 属性］	サポート	実装仕様
tcpCurrEstab	[(9) RO] 現在の状態が ESTABLISHD 状態か CLOSE-WAIT 状態である TCP コネクションの総数。	○	規格の通り
tcpInSegs	[(10) RO] 受信セグメントの総数。エラーセグメントの数も含まれる。	○	規格の通り
tcpOutSegs	[(11) RO] 送信セグメントの総数。現在の接続でのセグメント数は含むが、再送データのみのセグメントの数は含まれない。	○	規格の通り
tcpRetransSegs	[(12) RO] 再送信セグメントの総数	○	規格の通り
tcpConnTable	[(13) NA] TCP コネクション固有の情報テーブル	--	--
tcpConnEntry	[(1) NA] 特定の TCP コネクションに関するエントリ情報。コネクションが CLOSE 状態に推移するとこの値は削除される。INDEX は「tcpConnLocalAddress」「tcpConnLocalPort」「tcpConnRemAddress」「tcpConnRemPort」	--	--
tcpConnState	[(1) RW] TCP コネクションの状態。 closed(1) listen(2) synSent(3) synReceived(4) established(5) finWait1(6) finWait2(7) closeWait(8) lastAck(9) closing(10) timeWait(11)	○	規格の通り
tcpConnLocalAddress	[(2) RO] この TCP コネクションのローカル IP アドレス。ノードに関連するどんな IP インタフェースに対してもコネクションを受け取る LISTEN 状態のコネクションの 場合は 0.0.0.0 を使用	○	規格の通り
tcpConnLocalPort	[(3) RO] この TCP コネクションのローカルポート番号	○	規格の通り
tcpConnRemAddress	[(4) RO] この TCP コネクションのリモート IP アドレス	○	規格の通り
tcpConnRemPort	[(5) RO] この TCP コネクションのリモートポート番号	○	規格の通り
tcpInErrs	[(14) RO] 受信したエラーセグメントの総数（TCP チェックサムエラーなど）	○	規格の通り
tcpOutRsts	[(15) RO] RST フラグを持つセグメントの送信数（TCP コネクションをリセットした回数）	○	規格の通り

● ip グループ（udp）

名　前	規格［(識別子) 属性］	サポート	実装仕様
udpInDatagrams	[(1) RO] UDP のユーザに送られた UDP データグラムの総数（パケット数）	○	規格の通り
udpNoPorts	[(2) RO] 受け取ったが宛先ポートでアプリケーションが起動されていなかった UDP データグラム数	○	規格の通り
udpInErrors	[(3) RO] udpNoPorts 以外の理由で上位プロトコルに通知できなかった UDP データグラム数	○	規格の通り
udpOutDatagrams	[(4) RO] 送信した UDP データグラムの総数	○	規格の通り
udpTable	[(5) NA] UDP リスナーの情報テーブル。	--	--
udpEntry	[(1) NA] 特定の UDP リスナーに関するエントリ。INDEX は「udpLocalAddress」「udpLocalPort」	--	--
udpLocal Address	[(1) RO] この UDP リスナーのローカル IP アドレス。 ノードに関連するどんな IP インタフェースに対してもデータグラムを受け取る UDP リスナーの場合は 0.0.0.0 を使用	○	規格の通り
udpLocalPort	[(2) RO] この UDP リスナーのローカルポート番号	○	規格の通り

● ip グループ（snmp）

名　前	規格［(識別子) 属性］	サポート	実装仕様
snmpInPkts	[(1) RO] 受信した SNMP メッセージの総数	○	規格の通り
snmpOutPkts	[(2) RO] 送信した SNMP メッセージの総数	○	規格の通り
snmpInBadVersions	[(3) RO] 未サポートバージョンの受信メッセージの総数	○	規格の通り
snmpInBadCommunity Names	[(4) RO] 未登録コミュニティの受信メッセージの総数	○	規格の通り
snmpInBadCommunity Uses	[(5) RO] そのコミュニティでは許可されていないオペレーションが指定されている受信メッセージの総数	○	規格の通り
snmpInASNParseErrs	[(6) RO] デコード中に ASN.1 エラーもしくは BER エラーとなった受信メッセージの総数	○	規格の通り
snmpInTooBigs	[(8) RO] エラーステータスが tooBig の受信 PDU の総数	○	規格の通り
snmpInNoSuchNames	[(9) RO] エラーステータスが noSuchName の受信 PDU の総数	○	規格の通り
snmpInBadValues	[(10) RO] エラーステータスが badValue の受信 PDU の総数	○	規格の通り
snmpInReadOnlys	[(11) RO] エラーステータスが readOnly の受信 PDU の総数	○	規格の通り
snmpInGenErrs	[(12) RO] エラーステータスが genErr の受信 PDU の総数	○	規格の通り
snmpInTotalReqVars	[(13) RO] GetRequest、GetNextRequest を受信して正常に読み出された MIB オブジェクトの総数	○	規格の通り
snmpInTotalSetVars	[(14) RO] SetRequest を受信して正常に更新された MIB オブジェクトの総数	○	規格の通り
snmpInGetRequests	[(15) RO] 受信した GetRequestPDU の総数	○	規格の通り
snmpInGetNexts	[(16) RO] 受信した GetNextRequestPDU の総数	○	規格の通り
snmpInSetRequests	[(17) RO] 受信した SetRequestPDU の総数	○	規格の通り
snmpInGetResponses	[(18) RO] 受信した GetResponsePDU の総数	○	規格の通り
snmpInTraps	[(19) RO] 受信したトラップ PDU の総数	○	規格の通り
snmpOutTooBigs	[(20) RO] エラーステータスが tooBig の送信 PDU の総数	○	規格の通り
snmpOutNoSuchNames	[(21) RO] エラーステータスが noSuchName の送信 PDU の総数	○	規格の通り
snmpOutBadValues	[(22) RO] エラーステータスが badValue の送信 PDU の総数	○	規格の通り
snmpOutGenErrs	[(24) RO] エラーステータスが genErr の送信 PDU の総数	○	規格の通り
snmpOutGetRequests	[(25) RO] 送信した GetRequestPDU の総数	○	規格の通り
snmpOutGetNexts	[(26) RO] 送信した GetNextRequestPDU の総数	○	規格の通り

名　前	規格［(識別子) 属性］	サポート	実装仕様
snmpOutSetRequests	[(27) RO] 送信した SetRequestPDU の総数	○	規格の通り
snmpOutGetResponses	[(28) RO] 送信した GetResponsePDU の総数	○	規格の通り
snmpOutTraps	[(29) RO] 送信したトラップ PDU の総数	○	規格の通り
snmpEnableAuthen Traps	[(30) RW] エージェントが認証失敗トラップを生成するかどうか。 enabled(1) disabled(2)	○	規格の通り

11-2 SNMP(エージェント)機能の使用手順

11-2-1 使用手順

(1) 基本的な設定を行います。

基本的な操作の流れは 🕮「1-5 EtherNet/IP 通信の使用手順」を参照してください。

(2) Sysmac Studio の［構成・設定］の高機能ユニット設定で、以下の設定をします。

［SNMP］または［SNMP トラップ設定］ダイアログに以下の設定をします。

- SNMP サービス
- 受付認証 1
- 受付認証 2

(3) ［コントローラ］メニューから［転送［パソコン→コントローラ］］を選択し、［はい］ボタンをクリックします。EtherNet/IP ユニット設定が CPU ユニットに転送されます。

11-2-2 SNMP(エージェント) 機能に必要な設定一覧

SNMP（エージェント）機能を使用するときは、次に示す Sysmac Studio の EtherNet/IP ユニット設定を行います。

タブ	項目名		設定が必要な条件	参照ページ
SNMP 設定	SNMP サービス		必須	🕮 P.5-11
	ポート No.		任意 （注）デフォルト値「161」から変更する場合は必要です。	
	連絡先、設置場所		任意	
	認証トラップを送信する		任意 指定した SNMP マネージャ（受付認証 1、2）以外からアクセスがあった場合、認証トラップを送信する場合に設定します。	
	受付認証 1、受付認証 2		任意 特定の SNMP マネージャのみアクセスを許可する場合に設定します。	🕮 P.5-12
		IP アドレス		
		ホスト名		
		コミュニティ名		
SNMP トラップ設定	SNMP トラップ		必須	🕮 P.5-13
	ポート No.		任意 （注）デフォルト値「162」から変更する場合は必要です。	
	トラップ 1、トラップ 2			🕮 P.5-14
		IP アドレス	必須 SNMP トラップの送信先を IP アドレスかホスト名で設定します。	
		ホスト名		
		コミュニティ名	任意	
		バージョン	必須 SNMP マネージャが対応しているバージョンを選択します。	



参考

SNMP（エージェント）機能を使用する場合の設定は、［SNMP 設定 ］ダイアログおよび［SNMP トラップ設定 ］で行います。
［SNMP 設定 ］ダイアログについての詳細は、🕮「5-6 ［SNMP 設定］ダイアログ」を参照してください。また、［SNMP トラップ設定 ］ダイアログについての詳細は、🕮「5-7 ［SNMP トラップ設定］ダイアログ」を参照してください。

12

セキュリティ機能

ここでは、EtherNet/IP ユニット形 CJ1W-EIP21S のセキュリティ機能について説明します。本章で説明する機能は形 CJ1W-EIP21 にはありません。

12-1 セキュリティ機能の概要 12-2
12-1-1 セキュリティ機能の一覧 12-2
12-2 ポートのオープン／クローズ機能 12-3
12-2-1 機能の概要 12-3
12-2-2 機能の詳細 12-3
12-3 Packet Filter 機能 12-7
12-3-1 機能の概要 12-7
12-3-2 機能の詳細 12-7
12-4 セキュリティ機能の一般的なユースケース 12-11
12-4-1 ユースケース 12-11
12-4-2 ケース 1：特定のプロトコルからの受信を許可する 12-11
12-4-3 ケース 2：特定の送信元 IP アドレスからの受信を許可する 12-12
12-5 セキュリティ脅威からの保護対策 12-14

12-1　セキュリティ機能の概要

ここでは、EtherNet/IP ユニット形 CJ1W-EIP21S のセキュリティ機能の概要について説明します。

12-1-1　セキュリティ機能の一覧

EtherNet/IP ユニット形 CJ1W-EIP21S のセキュリティ機能は、ネットワークを経由するアクセスからユーザ資産を保護するための機能群です。
これらの機能により CPU ユニット内のユーザプログラムや各種データを保護します。

機能名	機能の概要	参照
ポートのオープン／クローズ機能	個々の通信機能に割り当てたTCP/UDPポートの通信を設定により遮断したり、通過させたりします。	「12-2 ポートのオープン／クローズ機能」
Packet Filter機能	あらかじめ設定した条件にしたがって、EtherNet/IPポートが受信するパケットを選択的に通過させます。	「12-3 Packet Filter機能」

また、ユースケースごとの構成例とその場合の設定について「12-4 セキュリティ機能の一般的なユースケース」で説明しています。

対応するサポートソフト

対応する周辺ツールは以下のとおりです。

周辺ツール	バージョン
Sysmac Studio	Ver.1.60以降

対応する CPU ユニット

対応する CPU ユニットは以下のとおりです。

CPUユニット	バージョン
NJシリーズCPUユニット	Ver.1.67以降

12-2 ポートのオープン／クローズ機能

ここでは、ポートのオープン / クローズ機能について説明します。

12-2-1 機能の概要

ポートのオープン／クローズ機能は、個々の通信機能に割り当てた TCP/UDP ポートの通信を設定により遮断したり、通過させたりする機能です。
使用しない通信機能を［使用しない］と設定することで、外部から攻撃できる入り口を減らしシステムのセキュリティ性能を向上することができます。

使用上の注意

使用するあるいは使用する可能性がある通信機能を全て［使用する］に設定してください。
この機能は、あるノードの通信機能を設定により有効にしたり無効にしたりする機能です。たとえば、CIP メッセージサーバを使用しない設定とすると、そのノードに対するタグデータリンクが動作しません。

12-2-2 機能の詳細

TCP/UDP 通信機能と、通信機能を使用するか使用しないかの選択可否は以下のとおりです。
ここでは形 CJ1W-EIP21S だけでサポートする機能について説明します。

設定項目

以下の通信機能について設定が可能です。

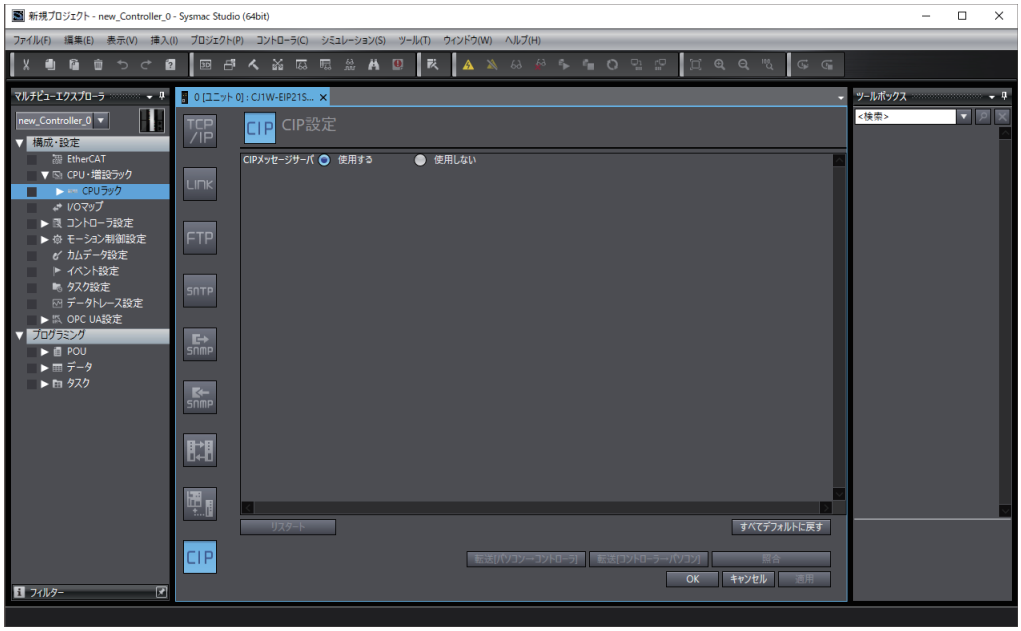
- CIP メッセージサーバ
- FINS/UDP
- FINS/TCP

設定をユニットへ転送完了したのち、ユニットをリスタートすると設定内容が反映され有効になります。

● CIP メッセージサーバ

Sysmac Studio の EtherNet/IP ユニット形 CJ1W-EIP21S の高機能ユニット設定画面の［CIP］-［CIP メッセージサーバ］で設定します。

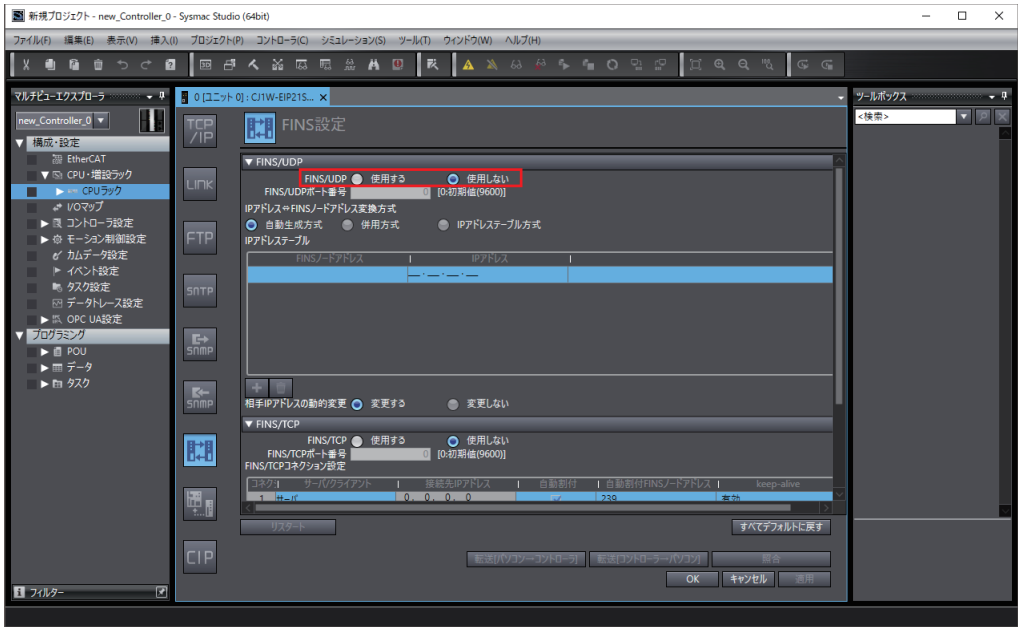
項目	内容	初期値
CIPメッセージサーバ	CIPメッセージサーバを使用するか使用しないかを選択します。 • 使用しない • 使用する	使用する



● FINS/UDP

Sysmac Studio の EtherNet/IP ユニット形 CJ1W-EIP21S の高機能ユニット設定画面の［FINS］-［FINS/UDP］で設定します。

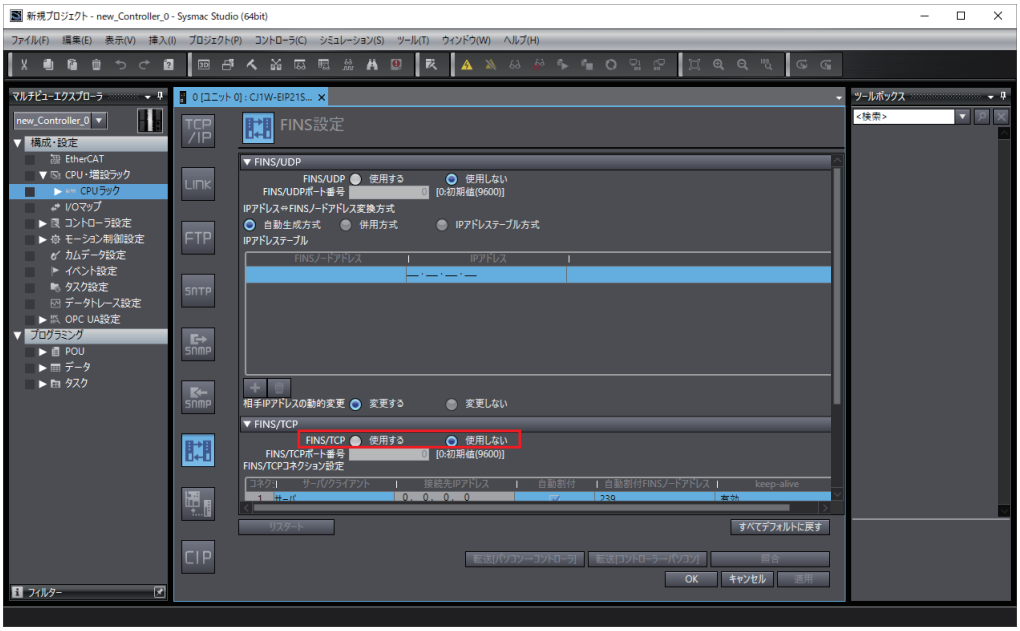
項目	内容	初期値
FINS/UDP	FINS/UDPを使用するか使用しないかを選択します。 • 使用しない • 使用する	使用しない



● FINS/TCP

Sysmac Studio の EtherNet/IP ユニット形 CJ1W-EIP21S の高機能ユニット設定画面の［FINS］-［FINS/TCP］で設定します。

項目	内容	初期値
FINS/TCP	FINS/TCPを使用するか使用しないかを選択します。 • 使用しない • 使用する	使用しない



仕様

● CIP メッセージサーバ

［CIP メッセージサーバ］を［使用する］に設定するとタグデータリンクや Explicit メッセージが使用できます。

項目	動作
使用しない	• タグデータリンクや Explicit メッセージが使用できない。 • Network Configurator を［Ethernet I/F］で接続できない。 • NS シリーズプログラマブルターミナルを［EtherNet/IP］で接続できない。
使用する *1	• タグデータリンクや Explicit メッセージが使用できる。 • Network Configurator を［Ethernet I/F］で接続できる。 • NS シリーズプログラマブルターミナルを［EtherNet/IP］で接続できる。

*1. 初期値

タグデータリンク設定があるとき、あるノードを［CIP メッセージサーバ］を［使用しない］に設定したときの振る舞いは以下のとおりです。

自ノードに設定		相手ノードに設定	
オリジネータ	ターゲット	オリジネータ	ターゲット
照合異常（ターゲット不在）が発生する。 • 7 セグ LED 表示 :d5 • 異常履歴コード : 記録しない。	エラー検出なし。	照合異常（ターゲット不在）が発生する。 • 7 セグ LED 表示 :d5 • 異常履歴コード : 記録しない。	エラー検出なし。

● FINS/UDP、FINS/TCP

サービス	設定	動作
FINS/UDP	使用しない *1	• FINS/UDP　サービスを使用できない。 • FINS/UDP、FINS/TCP とも使用しない設定の場合、SendCmd 命令を実行したら異常終了となる。*2
	使用する	FINS/UDPサービスを使用できる。
FINS/TCP	使用しない *1	• FINS/TCP　サービスを使用できない。 • FINS/UDP、FINS/TCP とも使用しない設定の場合、SendCmd 命令を実行したら異常終了となる。*2
	使用する	FINS/TCPサービスを使用できる。

*1. 初期値
*2. 命令のコントロールデータのうち「レスポンス要 / 不要」の設定により振る舞いが異なります。
レスポンス不要の場合、命令が正常終了します。
レスポンス要の場合、命令が異常終了します。終了コードは 0205Hex（タイムアウトエラー : 相手局から応答が返らず監視タイマがアップした）です。

12-3 Packet Filter 機能

ここでは、Packet Filter 機能について説明します。

12-3-1 機能の概要

Packet Filter 機能は、あらかじめ設定した条件にしたがって、EtherNet/IP ポートが受信するパケットを選択的に許可する機能です。これによりシステムのセキュリティ性能を向上することができます。

使用上の注意

Packet Filter には、受信を許可する機器や条件を全て登録してください。
Packet Filter の設定が該当システムの仕様と合致していない、あるいは設定の操作を誤っていた場合は、機器が通信できなかったり、サポートソフトと接続できなかったりする場合があります。そのため、「12-4 セキュリティ機能の一般的なユースケース」に説明するユースケースでだけ使用することを推奨します。
そうでない場合は、該当システムのネットワーク仕様を十分に把握していて Ethernet の知識を持つ方が、設定や管理を行ってください。
ネットワーク仕様とは、機器の IP アドレスや通信に使用する TCP/UDP ポート番号などを指します。

12-3-2 機能の詳細

Sysmac Studio の EtherNet/IP ユニット形 CJ1W-EIP21S の高機能ユニット設定画面の［TCP/IP］-［Packet Filter］で、機能を使用する／使用しないと、許可するパケットのフィルタ条件を設定します。

使用上の注意

- Packet Filter 設定を有効にした場合、フィルタ対象となる EtherNet/IP ポート経由で通信している機器と通信できなくなりますので、設定に注意してください。
- Sysmac Studio で［Packet Filter］を［使用する］から［使用しない］に変更したときに、［許可パケット］の設定内容を保持します。［許可パケット］の設定内容は、機能を使用する／使用しないに関係無く、EtherNet/IP ユニット形 CJ1W-EIP21S に転送されます。
- Packet Filter 機能はステートフルインスペクションに対応しています。したがって、EtherNet/IP ユニット形 CJ1W-EIP21S がクライアントとして相手機器にリクエストするとき、通信相手からのレスポンスを Packet Filter テーブルに追加することなく受信が可能です。
  ステートフルインスペクションに対応するプロトコルは TCP、UDP、ICMP です。

参考

以下の機能は Packet Filter 機能の対象外です。これは、以下の機能はクライアント機能で、使用するときにポートをオープンし、完了したらクローズするためです。
DNS/SNTP

設定項目

Packet Filter の設定画面と設定項目については、「Packet Filter」(P.5-5) を参照してください。

設定をユニットへ転送完了したのち、ユニットをリスタートすると設定内容が反映され有効になります。

利用シーン別の設定内容

Packet Filter を使用する場合の許可パケットの設定値を使用する機器や機能ごとに示します。

● サポートソフト接続時の設定値

サポートソフト使用時の設定値は、以下のとおりです。

サポートソフト	接続方法	使用プロトコル	［プロトコルフィルタ］の設定			
			プロトコル	［宛先ポート］の設定		
				範囲指定	開始ポート	終了ポート
Network Configurator	インタフェースに Ethernet I/Fを選択	CIP	TCP	使用する	44818	44818
		CIP	UDP		44818	44818
		ICMP	ICMP	使用しない	－	－

● EtherNet/IP 通信使用時の設定値

EtherNet/IP 通信使用時の設定値は、以下のとおりです。

通信名	使用プロトコル	条件	［プロトコルフィルタ］の設定			
			プロトコル	［宛先ポート］の設定		
				範囲指定	開始ポート	終了ポート
EtherNet/IPメッセージ	UCMM	サーバ	TCP	使用する	44818	44818
	Class3	サーバ	TCP		44818	44818
EtherNet/IPタグデータリンク	Class1	オリジネータ	IGMP *1	使用しない	－	－
		ターゲット	TCP	使用する	44818	44818

*1. マルチキャストを使用する場合に設定します。

参考

SYSMAC Gateway や CX-Compolet を使用する場合は、上記の設定に追加して ICMP も許可します。
設定内容は以下のとおりです。

［プロトコルフィルタ］の設定			
プロトコル	［宛先ポート］の設定		
	範囲指定	開始ポート	終了ポート
ICMP	使用しない	－	－

● オムロン製プログラマブルターミナル接続時の設定値

オムロン製プログラマブルターミナル接続時の設定値は、以下のとおりです。

・NA シリーズの場合

接続方法	使用プロトコル	［プロトコルフィルタ］の設定			
		プロトコル	［宛先ポート］の設定		
			範囲指定	開始ポート	終了ポート
「EtherNet/IP」で接続	CIP	TCP	使用する	44818	44818
「Ethernet」で接続	FINS	UDP		9600 *1	9600 *1

*1. FINS/UDP の設定で初期値以外のポート No. を設定した場合は、その値を設定します。

・NB シリーズの場合

接続方法	使用プロトコル	［プロトコルフィルタ］の設定			
		プロトコル	［宛先ポート］の設定		
			範囲指定	開始ポート	終了ポート
「Ethernet」で接続	FINS	UDP	使用する	9600 *1	9600 *1

*1. FINS/UDP の設定で初期値以外のポート No. を設定した場合は、その値を設定します。

・NS シリーズの場合

接続方法	使用プロトコル	［プロトコルフィルタ］の設定			
		プロトコル	［宛先ポート］の設定		
			範囲指定	開始ポート	終了ポート
「EtherNet/IP」で接続	CIP	TCP	使用する	44818	44818
「Ethernet」で接続	FINS	UDP		9600 *1	9600 *1

*1. FINS/UDP の設定で初期値以外のポート No. を設定した場合は、その値を設定します。

● FINS メッセージ通信機能使用時の設定値

FINS メッセージ通信機能使用時の設定値は、以下のとおりです。

接続方法	使用プロトコル	［プロトコルフィルタ］の設定			
		プロトコル	［宛先ポート］の設定		
			範囲指定	開始ポート	終了ポート
FINS/UDP	FINS	UDP	使用する	9600 *1	9600 *1
FINS/TCP	FINS	TCP		9600 *1	9600 *1

*1. FINS/UDP または FINS/TCP の設定で初期値以外のポート No. を設定した場合は、その値を設定します。

● SNMP 機能使用時の設定値

SNMP 機能使用時の［プロトコルフィルタ］［宛先ポート］の設定は、以下のとおりです。
［プロトコルフィルタ］［プロトコル］：UDP
［宛先ポート］
　範囲指定 : 使用する
　開始ポート：161 *1
　終了ポート：162 *1
*1SNMP/SNMP トラップのポート No. を設定している場合は、その値を設定します。

● FTP サーバ機能使用時の設定値

FTP サーバ機能使用時の［プロトコルフィルタ］［宛先ポート］の設定は、以下のとおりです。
［プロトコルフィルタ］［プロトコル］：TCP
［宛先ポート］
　範囲指定 : 使用する
　開始ポート :20 *1
　終了ポート :21 *1
*1*1:FTP のポート No. を設定している場合は、その値を設定します。

トラブルシュート

● EtherNet/IPユニット形CJ1W-EIP21SのPacket Filterの設定を忘れてしまった場合

該当のユニットを装着しているCPUユニットにSysmac StudioをオンラインにしてPacket Filterの設定を確認または変更します。

● 相手機器からアクセスできない場合

以下のステップで、原因を絞り込み対処します。

ステップ	項目	内容
1	Packet Filter設定が原因かを切り分ける	• Packet Filter を使用しないに設定して、相手機器がアクセス可能になるかを確認する。 • アクセスできる場合は Packet Filter 設定が原因なので、ステップ 2 を実施する。 • アクセスできない場合はPacket Filter設定以外の原因が考えられる。通信経路、接続先設定等々を見直す。
2	Packet Filter設定を見直す	• 形CJ1W-EIP21Sに接続する機器や周辺ツールのIPアドレスとポート番号を設定し直す。 • 回線データをモニタして、送信元 IP アドレス、送信元 Port、送信先 Port が Packet Filter 設定に登録されているかを確認する。（可能な場合のみ。）

12-4 セキュリティ機能の一般的なユースケース

ここでは、セキュリティ機能の一般的なユースケースとそれごとの構成例と設定について説明します。

12-4-1 ユースケース

以下のユースケースで説明します。

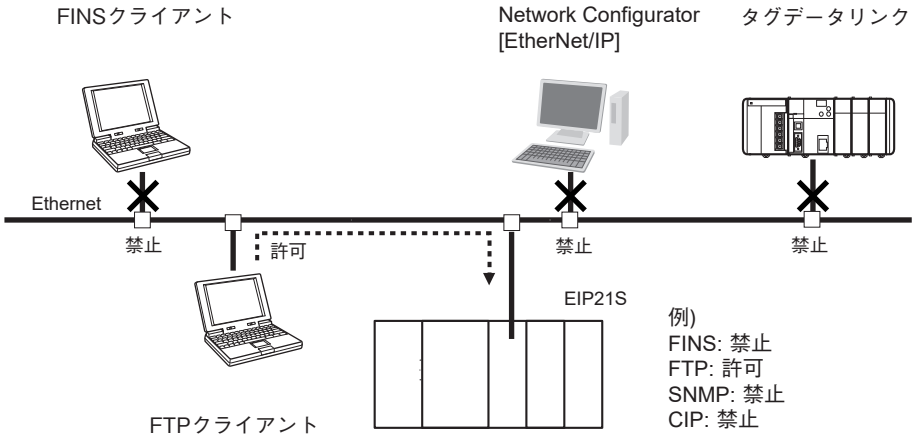
ケース	ユースケース	使用する機能	参照
1	特定のプロトコルの受信を許可する	ポートのオープン/クローズ機能	「12-2ポートのオープン／クローズ機能」
2	特定の送信元IPアドレスからの受信を許可する	Packet Filter機能	「12-3 Packet Filter機能」

12-4-2 ケース1：特定のプロトコルからの受信を許可する

特定のプロトコルのアクセスだけを許可するユースケースです。
使用しないプロトコルの通信を禁止したい場合に使用します。
このケースでは、ポートのオープン／クローズ機能を使用します。

構成例

この構成例では、FTP 以外のプロトコルの通信を禁止します。



構成例での設定

以下の表のとおり設定します。

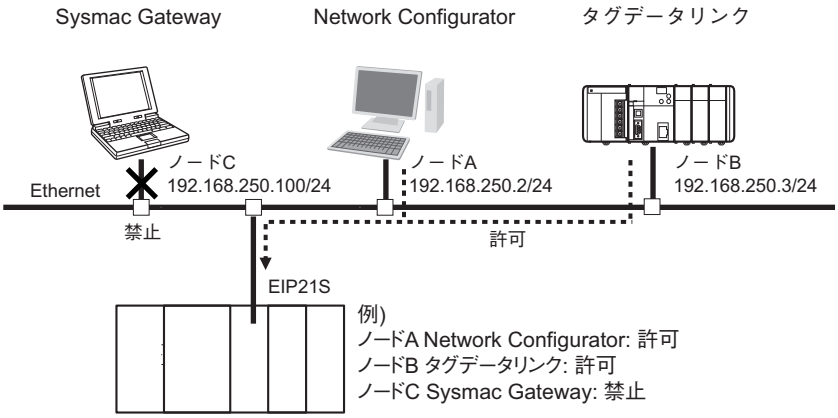
TCP/IP通信機能	設定	参照
CIPメッセージサーバ	使用しない	「12-2 ポートのオープン／クローズ機能」
FINS/UDP	使用しない	
FINS/TCP	使用しない	
FTPサーバ	使用する	「9-3 FTPサーバ機能の使用手順」
SNMP	使用しない	「11-2 SNMP（エージェント）機能の使用手順」

12-4-3　ケース 2：特定の送信元 IP アドレスからの受信を許可する

特定のノードからのアクセスだけを許可するユースケースです。
現場に未許可で持ち込んだ PC などのような、許可されていないノードからの接続を禁止したい場合に使用します。
このケースでは、Packet Filter 機能を使用します。

構成例

この構成例では、特定のパソコンと特定の外部機器からの通信だけを許可し、それ以外のクライアント機器からの通信を禁止します。



構成例での設定

以下の表のとおり、［Packet Filter］画面で設定します。
ノードA､BのIPアドレスを個別に設定します｡そのためマスクには｢255.255.255.255｣を設定します。

No.	IPフィルタ			プロトコルフィルタ						
	送信元設定			プロトコルフィルタ	送信元ポート			宛先ポート		
	指定方法	IPアドレス	マスク		範囲指定	開始ポート番号	終了ポート番号	範囲指定	開始ポート番号	終了ポート番号
1	IPアドレス指定	192.168.250.2	255.255.255.255	すべて	－	－	－	－	－	－
2	IPアドレス指定	192.168.250.3	255.255.255.255	すべて	－	－	－	－	－	－

使用上の注意

Packet Filter 設定に登録されない IP アドレスからの通信ができなくなるため、使用する全外部機器の IP アドレスを登録しておきます。

参考

マスクを 255.255.255.255 以外に設定すると、複数の IP アドレスの通信を許可することができます。
たとえば、以下のとおり設定すると、IP アドレスが 192.168.250.0 から 192.168.250.255 の機器からの通信を許可することができます。

No.	IPフィルタ			プロトコルフィルタ						
	送信元設定			プロトコルフィルタ	送信元ポート			宛先ポート		
	指定方法	IPアドレス	マスク		範囲指定	開始ポート番号	終了ポート番号	範囲指定	開始ポート番号	終了ポート番号
1	IPアドレス指定	192.168.250.0	255.255.255.0	すべて	－	－	－	－	－	－

12-5 セキュリティ脅威からの保護対策

EtherNet/IP ユニット形 CJ1W-EIP21S のセキュリティ機能を使用すると、セキュリティ脅威からのネットワークの保護に役立ちます。そのために、ユニットの機能を適切に使用することが必要です。ここでは、ユニットのセキュリティ機能を適切に使用するための運用方法への対策について説明します。

● 通信機器を信頼できるネットワークに配置する

EtherNet/IP ユニットを含む通信機器を信頼できるネットワーク内に配置してください。
そうでないネットワークを経由する場合は、VPN を使用するなどの対策を実施してください。
EtherNet/IP ユニット形 CJ1W-EIP21S が対応するサービスは「1-3-1 仕様」を参照してください。

● データを消去してからユニットを廃棄する

EtherNet/IPユニット形CJ1W-EIP21SおよびそれをCPUユニットを廃棄するときには、情報漏えいを防止するために以下の方法でユニット内部の情報を消去してください。

実施対象	実施内容	操作方法
CPUユニット	メモリオールクリアを実施する。	『Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)』を参照。
形CJ1W-EIP21S	工場出荷の初期状態にする。	Sysmac StudioのCJ1W-EIP21Sの［高機能ユニット設定の編集］画面の左下にある［リスタート］ボタンをクリックします。その後表示される［ユニットのリスタート］ダイアログで［出荷時設定に戻して再起動］を選択して［リスタート実行］ボタンでリスタートします。

● ユニットへの物理的なアクセスを制限する

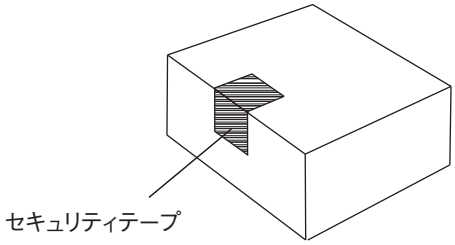
CPU ユニットと形 CJ1W-EIP21S を含むシステムを設置する場所へ関係者以外の人が入室することを制限する対策を実施してください。
該当場所への立ち入り者を制限することや、出入り口を施錠することなどです。

● ネットワーク構成ファイルへのアクセス制限と構成管理をする

Network Configurator で作成するネットワーク構成ファイルを、関係者以外の人が入手したり改ざんしたりすることができないように対策を実施してください。
ネットワーク情報ファイルをアクセス制限のある場所に保管することや、それの構成管理を行うことなどです。
ネットワーク情報ファイルについては『CS/CJ シリーズ Ethernet/IP ユニット ユーザーズマニュアル (SBCD-342)』を参照してください。

● 開梱されてない個体を使用する

EtherNet/IP ユニット形 CJ1W-EIP21S およびそれを接続する CPU ユニットは、購入時の個装箱のセキュリティテープがはがされていないものを使用してください。



使用上の注意

本商品の個装箱は商品の改ざん防止のため、開封したことが分かるテープで封止されています。開封されていないことを確認の上ご使用ください。

13

ソケットサービス機能

ここでは、EtherNet/IP ユニット形 CJ1W-EIP21S のソケットサービス機能について説明します。本章で説明する機能は形 CJ1W-EIP21 にはありません。

13-1 ソケット通信の基礎知識 13-3
13-1-1 ソケットとは 13-3
13-1-2 ソケットサービスのポート No. 13-3

13-2 プロトコルの基礎知識 13-4
13-2-1 TCP と UDP の相違点 13-4
13-2-2 TCP ソケットのオープン処理 13-5
13-2-3 送信データのフラグメント（分割）について 13-6

13-3 EtherNet/IP ユニット形 CJ1W-EIP21S のソケットサービス概要 13-9
13-3-1 概要 13-9
13-3-2 デバイス変数操作によるソケットサービスの概要 13-9
13-3-3 SendCmd によるソケットサービスの概要 13-11
13-3-4 EtherNet/IP ユニットのソケットサービス種類 13-11

13-4 ソケットサービス機能に必要な設定 13-12

13-5 ソケットサービス機能のステータス 13-13
13-5-1 UDP/TCP ソケット No.1 ～ No.8 ステータス（ユニット→ CPU ユニット） 13-13
13-5-2 TCP 受信データバイト数（ユニット→ CPU ユニット） 13-13
13-5-3 TCP コネクション状態（ユニット→ CPU ユニット） 13-13

13-6 デバイス変数操作によるソケットサービスの利用の詳細 13-15
13-6-1 使用手順 13-15
13-6-2 ソケットサービスとソケットステータスの使用手順 13-16
13-6-3 ソケットサービスパラメータエリア（CPU ユニット⇔ユニット） 13-17
13-6-4 パラメータ詳細 13-18
13-6-5 ソケットサービス要求スイッチ（CPU ユニット⇔ユニット） 13-21
13-6-6 終了コード 13-22
13-6-7 タイミングチャート 13-26
13-6-8 サンプルプログラム 13-27

13-7 SendCmd 命令によるソケットサービスの利用の詳細 13-28
13-7-1 ソケットサービスの使用方法 13-28
13-7-2 ソケットサービス要求 13-29
13-7-3 ソケットサービスとソケットステータスの使用手順 13-46
13-7-4 終了コード 13-47
13-7-5 通信時のタイミングチャート 13-47
13-7-6 ソケットサービスのタイミングチャート 13-48
13-7-7 サンプルプログラム 13-50

13-8 ソケットサービス使用上の留意事項 13-51
13-8-1 UDP/TCP ソケットサービス共通の留意事項 13-51
13-8-2 UDP ソケットサービスのみの留意事項 13-51
13-8-3 TCP ソケットサービスのみの留意事項 13-51
13-8-4 デバイス変数操作に関する留意事項 13-52
13-8-5 ソケットサービスの送信／受信に要する時間（ネットワーク上での遅延を無視した場合）. 13-53

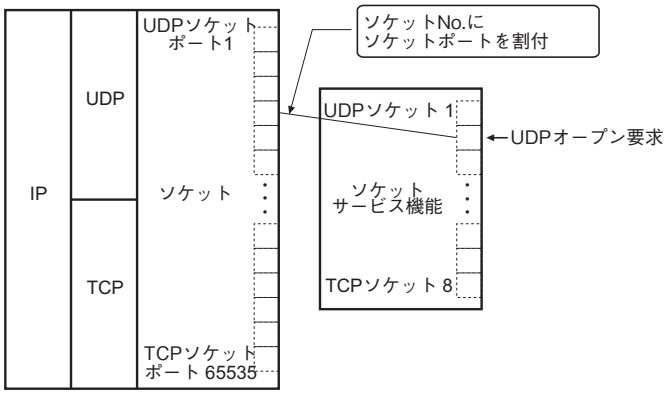
13-1 ソケット通信の基礎知識

13-1-1 ソケットとは

ソケットは､TCPまたはUDPの機能をユーザプログラムから直接利用するためのインタフェースです。
ホストコンピュータ（パソコンなど）の場合は、このソケットは C 言語のインタフェースライブラリなどの形で提供され、ライブラリを組込むと、ユーザプログラムから TCP や UDP を使用して通信が可能となります。
UNIX コンピュータの場合は、ソケットインタフェースがシステムコールの形で用意されています。
NJ シリーズコントローラの場合は、ユーザプログラムから、EtherNet/IP ユニットのデバイス変数を操作することでソケットサービスを使用できます。
このソケットを使用した通信サービスにより、ホストコンピューターコントローラ間、またはコントローラーコントローラ間で、任意のデータを相手ノードと送受信することができます。
本ユニットは、UDP ソケットサービス、TCP ソケットサービスの 2 つを提供します。

EtherNet/IP ユニットでのソケット利用

EtherNet/IP ユニットでは、ソケットサービス機能を介してソケットを利用します。
ソケットサービス機能では、UDP、TCP 各 8 ソケットを同時に使用でき、UDP、TCP それぞれ、No.1 ～ No.8 までのソケット No. が付けられています。
ラダープログラムからは、一つのソケット No. に対して一つのソケットポートを割り付けることでソケットを利用します。
割り付けるソケットポート No. は、ソケットのオープン時に指定します。



13-1-2 ソケットサービスのポート No.

UNIX ワークステーションでは、1023 以下のポート No. はスーパーユーザだけが使用できます。さらに、0 ～ 255 のポートは、well-known ポートとして予約されています。
したがって、ソケットサービスでは 1024 以上の値のポート No. を使用することをお勧めします。
また、EtherNet/IP ユニットは、ポート No.0 は持ちません。
さらに、特定のワークステーションでは、1024 以上のポートであっても予約済みの場合があります（例えば、X ウィンドウのサーバは 6000）。ポート No. を設定する場合は、重複して設定しないように注意してください。
UNIX ワークステーションのポート No. の設定状況については、/etc/services で確認することができます。

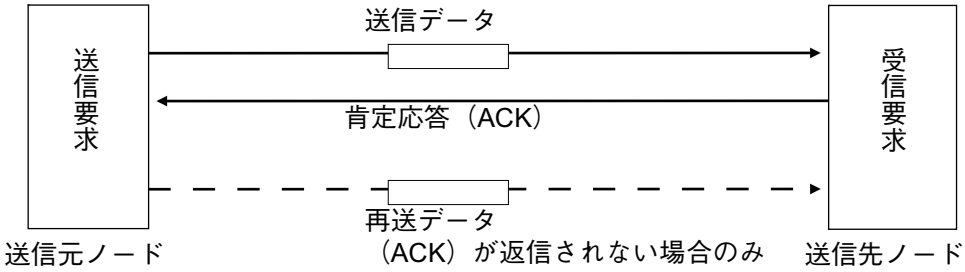
13-2 プロトコルの基礎知識

13-2-1 TCP と UDP の相違点

ソケットサービスで使用する TCP と UDP の機能は、次のように異なります。

TCP による通信

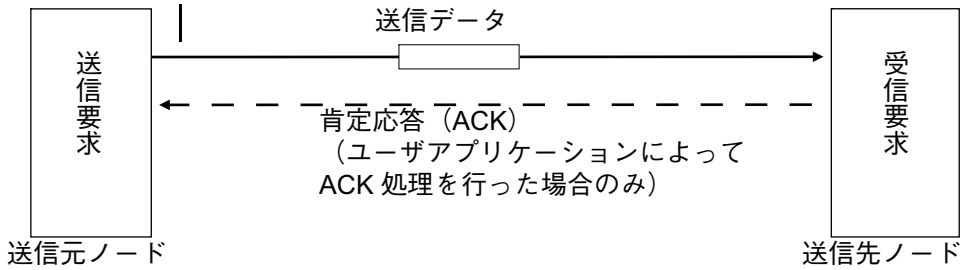
データの送信ごとに次のような動作を行うことによって、データが送信先ノードに確実に届くように処理されます。
送信先ノードでは、データを正常に受信すると肯定応答（ACK）を返信します。
送信元ノードでは、ACK が返信されてきたときは次のデータを送信し、一定時間以内に ACK が返信されないときは同じデータを再送します。



TCP では、ソケットのオープン要求時に相手 IP アドレスと相手 TCP ポート No. を指定し、送信要求時には送信バイト数と送信データ自身を指定します。
受信要求時には受信バイト数を指定します。
また TCP では、いったんソケットをオープンすると、クローズするまでは他の相手と通信することはできません。

UDP による通信

データは送信先ノードへ送信されるだけで、TCP のような受信確認・再送処理は行われません。
通信の信頼性を高めるためには、ユーザアプリケーションでデータの再送処理などを行う必要があります。



UDP では、ソケットのオープン要求時には相手 IP アドレスと相手 UDP ポート No. を指定せず、送信要求時に相手 IP アドレスと相手 UDP ポート No.、送信バイト数と送信データ自身を指定します。
受信要求時には、受信バイト数を指定します（受信したデータがどの IP アドレスからのどの UDP ポート No. を使用したレスポンスデータに反映されます）。
また UDP では、いったんソケットをオープンすると、クローズしなくとも他の相手と通信することができます。

13-2-2 TCP ソケットのオープン処理

TCP では、信頼性の高いデータ転送を行うために、2 ノード間に仮想的な通信回線を設定してからデータ転送を開始します。
この仮想的な通信回線のことを「コネクション」と呼びます。

Passive（受動的）オープンと Active（能動的）オープン

コネクション開設時、各ノードではオープン処理が実行されます。
このときノードがサーバになるか、クライアントになるかによって、オープンの仕方が異なります。
サーバとしてオープンする場合の処理を Passive（受動的）オープン、クライアントとしてオープンする場合の処理を Active（能動的）オープンと呼びます。



参考

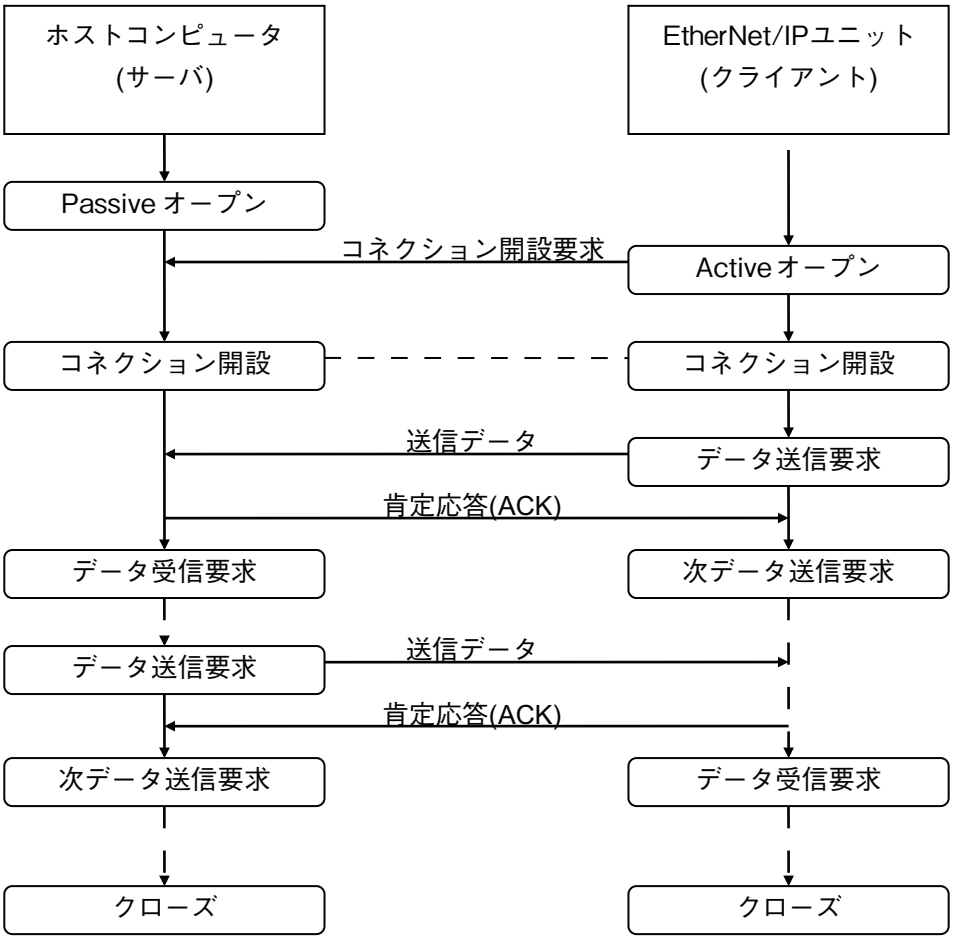
- コネクションを開設した TCP ソケット（サーバ側とクライアント側）では、クローズを実行しない限り、他の TCP ソケットと通信できません（TCP ソケットは 8 つまで用意されています）。
- UDP ソケットでは、複数の UDP ソケットと通信できます。

参考

クライアントとサーバ
2 ノード間でデータ転送処理を含んだアプリケーションが実行されるときに、処理のサービスを提供する側のノード上のプロセスを「サーバ」（店員）、サービスを要求する側のノード上のプロセスを「クライアント」（客）と呼びます。
サーバは先に起動されて、クライアントからのサービス要求を待ちます。
それに対して、クライアントはサーバに対してコネクションの開設要求を行ってから、データ転送を行います。
TCP を使用すると、これらの処理をアプリケーションのプログラム上で実行しなくても、プロトコルの段階で自動的に行うことができます。

TCP による通信手順

ホストコンピュータと EtherNet/IP ユニット間で TCP による通信を行い、EtherNet/IP ユニットがクライアント、ホストコンピュータがサーバとして働く場合は、次の図のような手順で処理が進行します。



13-2-3　送信データのフラグメント（分割）について

EtherNet/IP ユニットは、TCP の送信データについては 1,460 バイト、UDP の送信データについては 1,472 バイトにフラグメント（分割）して送信します。
そのため、受信側では次のような点に注意してデータを受信する必要があります。

TCP を使用する場合

TCP の通信では、データが分割されて送信されるときに次図のような状態が発生します。

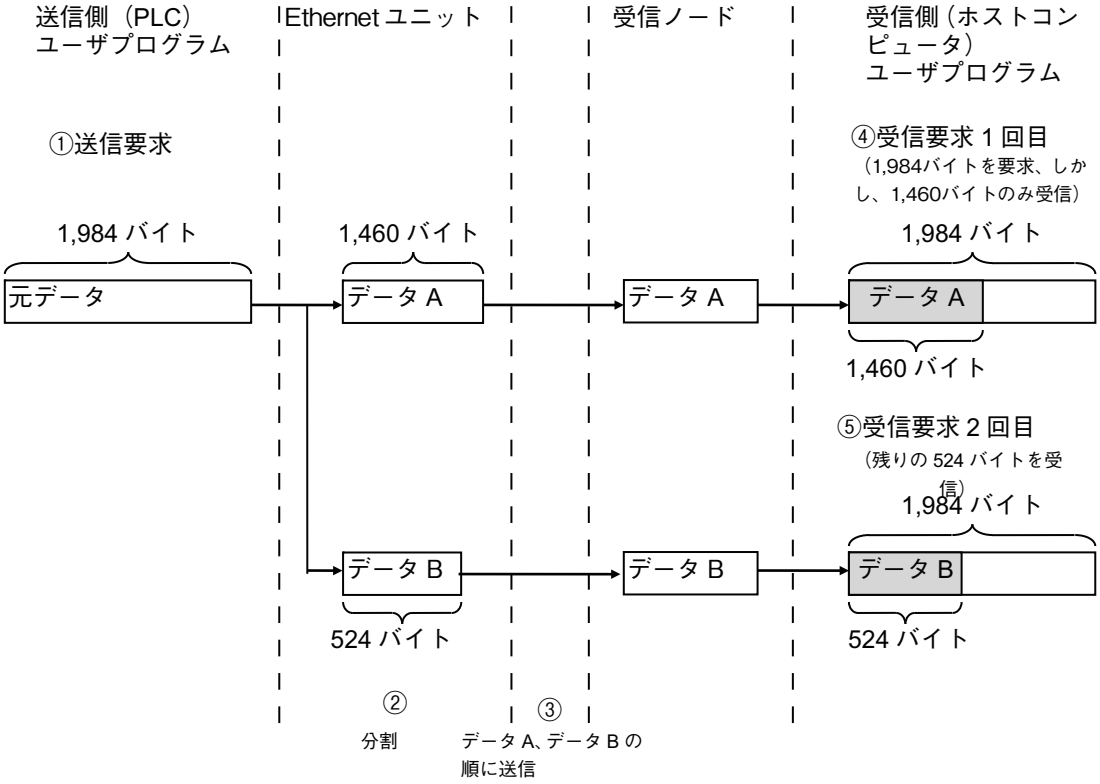
(1)　送信側のユーザプログラムで 1,984 バイトのデータの送信要求を行います。

(2)　EtherNet/IP ユニットで、送信データが 1,460 バイトのデータ A と、524 バイトのデータ B に分割されます。

(3)　送信側から、データ A とデータ B が順に送信されます。

(4)　受信側のユーザプログラムで、データ B が到着する前に 1,984 バイトのデータ受信要求を行っても、受信されるのは最初に送信されたデータ A のみで、データ B は受信されません。

(5)　再度受信要求を行うと、残りのデータ B が受信されます。



以上のように、TCP による通信ではデータが分割されたままの形でユーザプログラムに引き渡されます。
そのため、受信側のユーザプログラムでは送信データの終端を判断し、すべてのデータの受信が完了するまで、受信要求を繰返し行う必要があります。
上記の例では繰返し回数が 2 回ですが、通信経路にルータがあるとデータはさらに細分化されるので、受信要求の繰返し回数もさらに増やす必要があります。
ただし、受信要求時の受信データ長は、送信データ長と同じ値に指定する必要はありません。
たとえ送信データよりも小さい値を指定しても、受信要求を繰返すことにより、すべてのデータを受信することができるからです。

参考

異なるセグメントに対して、TCP で送信する場合は、536 バイトにフラグメントされます。

UDP を使用する場合

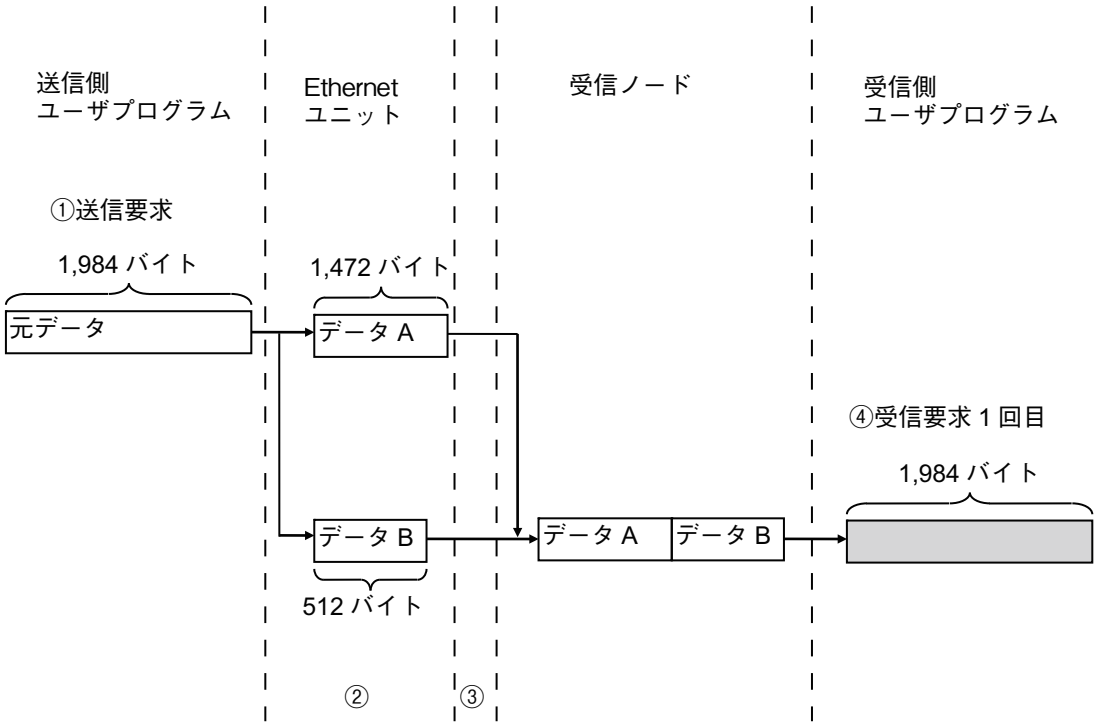
UDP の通信では、データが分割されて送信されるときに次図のような状態が発生します。

(1) 送信側のユーザプログラムで、1,984 バイトのデータの送信要求を行います。

(2) EtherNet/IP ユニットで、送信データが 1,472 バイトのデータ A と、512 バイトのデータ B に分割されます。

(3) 送信側から、データ A とデータ B が順に送信されます。

(4) 受信側のユーザプログラムが 1,984 バイトのデータ受信要求を行うと、データ A とデータ B が連結され、元の送信データに戻されて、ユーザプログラムに引き渡されます。



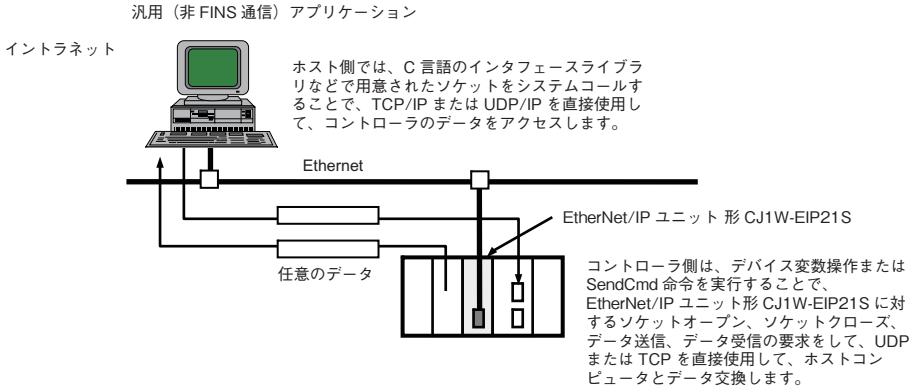
以上のように、UDP による通信はデータグラム単位で行われるため、送信データは元のデータの形に戻されてからユーザプログラムに引き渡されます。
そのため、ユーザプログラムでは送信データと同じデータ長を指定して受信要求を行えば、データ全体を一度の受信要求で受信することができます。
ただし、受信要求のデータ長指定を誤って送信データよりも短く指定すると、指定した受信データ長を超える部分の受信データは廃棄されます。

13-3 EtherNet/IP ユニット形 CJ1W-EIP21S のソケットサービス概要

ここでは、EtherNet/IP ユニット形 CJ1W-EIP21S のソケットサービス機能の概要について説明します。

13-3-1 概要

EtherNet/IP ユニット形 CJ1W-EIP21S のソケットサービスは、UDP または TCP を用いた汎用アプリケーションとコントローラ間でデータのやり取りをするときに使用します。
コントローラ側は、デバイス変数操作またはプログラム上の SendCmd 命令によって、ソケットサービスを要求します。



ソケットサービスを実行する方法には、以下の 2 つがあり、いずれかを選択できます。
- デバイス変数操作による方法：パラメータの設定とデバイス変数操作のみで、ソケットサービスを要求する
- SendCmd 命令による方法：EtherNet/IP ユニット形 CJ1W-EIP21S 宛サービス要求コマンドにより、ソケットサービスを要求する

なお、形 CJ1W-EIP21S でタグデータリンクを使用する場合に、形 CJ1W-EIP21S のソケットサービス機能を使用するときは、SendCmd 命令による方法を使用してください。デバイス変数による方法は使用しないでください。デバイス変数による方法を使用した場合、タグデータリンク応答性能が悪化する場合があります。

（注） 機能上の主な違いには、以下に示すサービス数の違いがあります。

UDP/TCP	デバイス変数による方法	SendCmd命令による方法
UDPソケットサービス	合計8ソケットまで可能	8ソケットまで可能
TCPソケットサービス		8ソケットまで可能

13-3-2 デバイス変数操作によるソケットサービスの概要

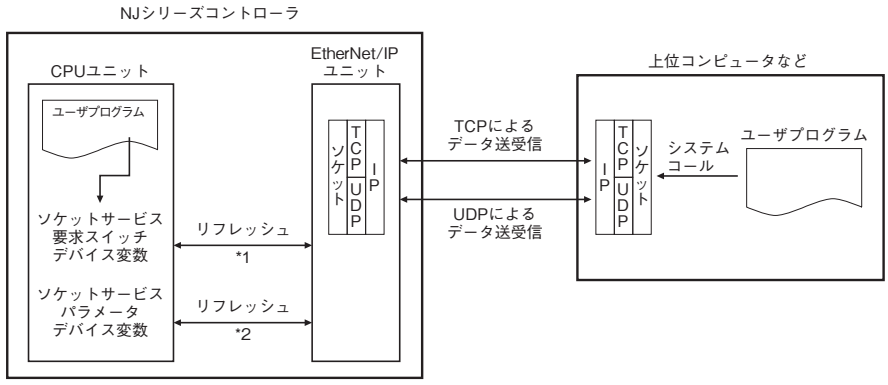
デバイス変数操作によるソケットサービスは、デバイス変数でのパラメータの設定と操作のみでソケットサービスを利用する方法です。ソケットサービスパラメータエリアのデバイス変数に必要なパラメータを格納後、ソケットサービス要求スイッチのデバイス変数を TRUE にすることで利用できます。
ただし、この方法で利用できるソケットは UDP と TCP 含めて 8 ソケットまでです。

また、UDP と TCP とで同じ No. のソケットを同時に使用することはできません（パラメータエリアはソケット No. 単位であり、UDP と TCP で共用のため）。

使用上の注意

形 CJ1W-EIP21S でタグデータリンクを使用する場合に、形 CJ1W-EIP21S のソケットサービス機能を使用するときは、SendCmd 命令による方法を使用してください。デバイス変数による方法は使用しないでください。デバイス変数による方法を使用した場合、タグデータリンク応答性能が悪化する場合があります。

使用方法のイメージ図を以下に示します。



*1：デバイス変数のソケットサービス要求スイッチによって、CPU→EtherNet/IPユニットへのサービスを要求
*2：デバイス変数のソケットサービスパラメータによって、CPU→EtherNet/IPユニットにサービス内容指示や、EtherNet/IPユニット→CPUに処理結果回答

ソケットサービスパラメータエリアのデバイス変数に必要なパラメータを設定してから、上記図中のソケットサービス要求スイッチのデバイス変数により、「UDP/TCP オープン」、「UDP/TCP 送信」、「UDP/TCP 受信」、「UDP/TCP クローズ」を実行します。
「UDP/TCP 送信」時には、ソケットサービスパラメータエリアの「送信 / 受信データアドレス」に設定した変数の送信データが送信されます。
「UDP/TCP 受信」時には、ソケットサービスパラメータエリアの「送信 / 受信データアドレス」に設定した変数に受信データが書き込まれます。

デバイス変数操作による方法

● 説明

デバイス変数でのパラメータの設定と操作のみで、EtherNet/IP ユニットのソケットサービスを使用する方法です。

● ポイント

EtherNet/IP ユニットのソケットサービスパラメータエリアのデバイス変数に必要なパラメータを格納後、ソケットサービス要求スイッチのデバイス変数を TRUE にすることで利用できます。

● メリット／デメリット

利用できるソケットは UDP と TCP 含めて 8 ソケットまでです。

13-3-3 SendCmd によるソケットサービスの概要

SendCmd 命令により EtherNet/IP ユニット宛サービス要求コマンドを発行することで、ソケットサービスを利用することができます。
SendCmd 命令を使用する場合は、UDP 8 ソケット分＋ TCP 8 ソケット分の計 16 ソケットの利用が可能です。

要求の発行はコマンドにより行い、ユニットが要求を受け付けるとレスポンスが返ってきます。
しかし、レスポンスが返ってきただけでは処理が完了したわけではなく、実際に要求した処理の完了はデバイス変数のソケットステータスに示される各動作中フラグの変化により判断する必要があります。
また、要求した処理が完了したときは、要求発行時（SendCmd 命令実行時）に指定した結果格納領域に処理結果が格納されます。

SendCmd 命令による方法

● **説明**

EtherNet/IP ユニット宛サービス要求コマンドにより、ソケットサービスを使用する方法です。

● **ポイント**

SendCmd 命令により EtherNet/IP ユニット宛サービス要求コマンドを発行することで、ソケットサービスを利用することができます。

● **メリット／デメリット**

- UDP 8 ソケット分＋ TCP 8 ソケット分の計 16 ソケットの利用が可能です。
- SendCmd 命令による方法は、デバイス変数操作による方法に比べて処理が低速です。

13-3-4 EtherNet/IP ユニットのソケットサービス種類

EtherNet/IP ユニットのソケットサービスの種類は、以下のとおりです。
デバイス変数操作、SendCmd 命令のどちらによっても実行できます。

UDP/TCP	ソケットサービス要求
UDPソケット	UDPソケットのオープン要求
	UDPソケットの受信要求
	UDPソケットの送信要求
	UDPソケットのクローズ要求
TCPソケット	TCPソケットのオープン要求(PASSIVE)
	TCPソケットのオープン要求(ACTIVE)
	TCPソケットの受信要求
	TCPソケットの送信要求
	TCPソケットのクローズ要求

13-4 ソケットサービス機能に必要な設定

ソケットサービス機能を使用するときは、以下のユニット設定を設定することが必要です。

Sysmac Studio設定画面	設定項目	設定が必要な条件	参照
TCP/IP	IPアドレス	任意	「5-2 ［TCP/IP設定］ダイアログ」
	サブネットマスク	任意	
	IPルータテーブル	任意（IPルータ越えのソケット通信を行う場合）	
	TCP/IP keep-alive	任意（デフォルト5分以外のとき）	
	高速化する（ソケットサービス高速化）	任意（ソケットサービスのデバイス変数操作の送受信処理性能改善する場合。ただし、高速化した場合は、SendCmd命令によるソケットサービスは使用できずエラーとなります。）	
ステータスエリア	レイアウトタイプ	必須 「ユーザ設定」に設定する。	「5-8［ステータスエリア設定］ダイアログ」
	割付エリア	必須 ユーザ設定エリアを割り付ける先頭アドレスを設定する。	

13-5 ソケットサービス機能のステータス

13-5-1 UDP/TCP ソケット No.1 ～ No.8 ステータス（ユニット→ CPU ユニット）

UDP ソケット、TCP ソケットの状態は、UDP/TCP ソケットステータスのデバイス変数に反映されます。デバイス変数は、「3-2-5 ソケットサービス関連の CJ ユニットデバイス変数（形 CJ1W-EIP21S のみ）」の以下を参照してください。
「UDP ソケット No.1 ～ No.8 ステータス（ユニット→ CPU ユニット）」
「TCP ソケット No.1 ～ No.8 ステータス（ユニット→ CPU ユニット）」

13-5-2 TCP 受信データバイト数（ユニット→ CPU ユニット）

TCP 受信データバイト数のデバイス変数には、その TCP ソケットの受信バッファにある、受信データのバイト数が格納されます。デバイス変数は、以下を参照してください。
「 TCP ソケット No.1 ～ 8 受信データバイト数（ユニット→ CPU ユニット）（*_Skt1TcpRcvDatByte ～ *_Skt8TcpRcvDatByte）」（P.3-21）

この変数と連動して、「受信データあり／なしフラグ」のデバイス変数を TRUE/FALSE にします。デバイス変数操作または SendCmd 命令により受信要求を実行した時点で、本デバイス変数はいったん 0000 Hex になります。
受信要求処理を完了しても受信バッファにデータが残っている場合は、再び残存バイト数を格納し、「受信データあり／なしフラグ」を TRUE にします。
データ受信のタイミングによっては、「受信データあり／なしフラグ」が TRUE になっても受信データ数が 0 の可能性があります。そのため、サンプルプログラム例を参考に、フラグ制御による受信処理を行ってください。
サンプルプログラム例は「A-9 ソケットサービス機能のサンプルプログラム」を参照してください。

受信要求は、受信データ数に、必要なデータが揃ったのを確認してから実行すると便利です。

受信バッファには、最大 4,096 バイトまでデータが保持されますが、デバイス変数操作または SendCmd 命令の受信要求で設定できる最大値（1,984 バイト）までの範囲で値を格納します。
　0000 Hex: 0 バイト
　07C0 Hex: 1984 バイト

13-5-3 TCP コネクション状態（ユニット→ CPU ユニット）

TCP コネクション状態には、その TCP ソケットを使用してオープンした、ポートの状態が反映されます。
また、ポートクローズ後も、再びそのソケットを使用してオープンするまで、前のポートの状態が反映されています。
ただし、ソケットステータスエリアとは同期していないため、変化のタイミングが多少異なります。
デバイス変数は、以下を参照してください。
「 TCP ソケット No.1 ～ 8 コネクション状態（ユニット→ CPU ユニット）（*_Skt1TcpConSta ～ *_Skt8TcpConSta）」（P.3-21）

値により、次のように状態を表します。

値	状態	内容
0	CLOSED	クローズされている。
1	LISTEN	コネクションを待っている。
2	SYN SENT	活性状態でSYNを送信した。
3	SYN RECEIVED	SYNを送信し、受信した。
4	ESTABLISHED	開設されている。
5	CLOSE WAIT	FINを受信した後、終了待ちの状態。
6	FIN WAIT1	終了して、FINを送信した。
7	CLOSING	終了して、FINを交換した。 FINの送達確認（ACK）を待っている。
8	LAST ACK	FINを送信して終了した。 FINの送達確認（ACK）を待っている。
9	FIN WAIT2	FINの送達確認（ACK）を受信した。 FINを待っている。
A	TIME WAIT	終了した後、最大セグメント生存時間の2倍（2MSL）の沈黙待ちの状態。

13-6 デバイス変数操作によるソケットサービスの利用の詳細

13-6-1 使用手順

手順

1 **ソケットサービス機能に必要な設定を行います。**
必要な設定項目については、「13-4 ソケットサービス機能に必要な設定」を参照してください。

2 **ソケットサービスパラメータエリアのデバイス変数にソケットサービスパラメータを設定します。**
ソケットサービスパラメータ 1 の場合のデバイス変数

CJ ユニットデバイス変数名	名称
*_Skt1ParamArea	ソケットサービスパラメータエリア 1
*_Skt1No	UDP/TCP ソケット No.
*_Skt1Option	ソケットオプション
*_Skt1KeepAlive	Keep-alive 機能使用
*_Skt1Linger	Linger 機能使用
*_Skt1LocalPortNo	自 UDP/TCP ポート No.
*_Skt1RemoteIPAdr	相手 IP アドレス
*_Skt1RemoteIPAdr1	相手 IP アドレスの 1 バイトめ
*_Skt1RemoteIPAdr2	相手 IP アドレスの 2 バイトめ
*_Skt1RemoteIPAdr3	相手 IP アドレスの 3 バイトめ
*_Skt1RemoteIPAdr4	相手 IP アドレスの 4 バイトめ
*_Skt1RemotePortNo	相手 UDP/TCP ポート No.
*_Skt1SendRcvByte	送信／受信バイト数
*_Skt1SendRcvAdr	送信／受信データアドレス
*_Skt1SendRcvAdrType	エリア種別
*_Skt1SendRcvAdrCh1	チャネルアドレス（上位 2 けた）
*_Skt1SendRcvAdrCh2	チャネルアドレス（下位 2 けた）
*_Skt1Timeout	タイムアウト値
*_Skt1RespCode	終了コード

3 **送信データまたは受信データを格納する変数を作成し、ソケットサービスパラメータエリアの送信／受信データアドレスに設定したメモリアドレスに AT（割付先）指定します。**
詳細な指定方法については、「送信／受信データアドレス」(P.13-20) を参照してください。

4 **ソケットサービス要求スイッチのデバイス変数を FALSE → TRUE に立ち上げます。**
ソケットサービス要求スイッチ 1 の場合のデバイス変数

CJ ユニットデバイス変数名	名称
*_Skt1Cmd	ソケットサービス要求スイッチ 1
*_Skt1UdpOpenCmd	UDP オープン要求スイッチ
*_Skt1TcpPassiveCmd	TCP オープン (Passive) 要求スイッチ
*_Skt1TcpActiveCmd	TCP オープン (Active) 要求スイッチ
*_Skt1SendCmd	送信要求スイッチ
*_Skt1RcvCmd	受信要求スイッチ
*_Skt1CloseCmd	クローズ要求スイッチ

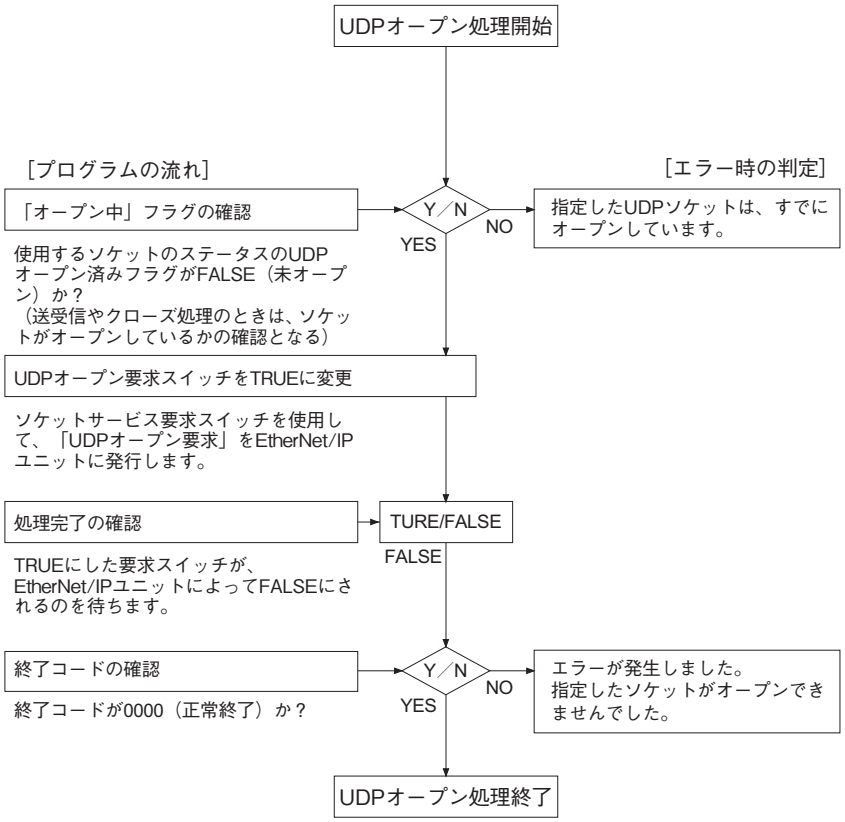
5 **送信要求または受信要求時には、ソケットサービスパラメータエリアの送信／受信データアドレスに AT（割付先）指定した変数のデータが自動的に送信または受信されます。**
すべての処理の結果を示す終了コードがソケットサービスパラメータの終了コードに自動的に格納されます。

留意事項

正常にオープンが完了したソケットのソケットサービスパラメータエリアは、クローズするまで別のソケット用のパラメータエリアとして使用することはできません。
各ソケットの TCP コネクション状態により、ソケットの状態を確認してください。デバイス変数の操作による TCP または UDP のソケットサービス機能にかぎり、オプション指定により送受信性能の改善を実施しています。また、TCP ソケットサービスに linger ソケットオプションを使用できます。このオプションを指定すると、ソケットクローズ後のポート No. 開放までの時間（約 1 分）を待たずに、即座に同じポート No. によるオープン処理が行えます。

13-6-2　ソケットサービスとソケットステータスの使用手順

ソケットサービスは、ソケットステータスエリアのフラグの変化やタイミングに注意しながら使用する必要があります。
下図は UDP オープン処理を例にしたときの流れを示しています。
その他のソケットサービスについても同様の手順となります。
UDP オープン処理以外のソケットサービスを使用する場合は、フラグ名称などを読み替えてください。



13-6-3　ソケットサービスパラメータエリア（CPU ユニット⇔ユニット）

ソケットサービス要求で使用するソケットサービス用のパラメータのデバイス変数は以下となっています。

ソケットサービスパラメータ 1 の場合のデバイス変数

CJ ユニットデバイス変数名	名称
*_Skt1ParamArea	ソケットサービスパラメータエリア 1
*_Skt1No	UDP/TCP ソケット No.
*_Skt1Option	ソケットオプション
*_Skt1KeepAlive	Keep-alive 機能使用
*_Skt1Linger	Linger 機能使用
*_Skt1LocalPortNo	自 UDP/TCP ポート No.
*_Skt1RemoteIPAdr	相手 IP アドレス
*_Skt1RemoteIPAdr1	相手 IP アドレスの 1 バイトめ
*_Skt1RemoteIPAdr2	相手 IP アドレスの 2 バイトめ
*_Skt1RemoteIPAdr3	相手 IP アドレスの 3 バイトめ
*_Skt1RemoteIPAdr4	相手 IP アドレスの 4 バイトめ
*_Skt1RemotePortNo	相手 UDP/TCP ポート No.
*_Skt1SendRcvByte	送信／受信バイト数
*_Skt1SendRcvAdr	送信／受信データアドレス
*_Skt1SendRcvAdrType	エリア種別
*_Skt1SendRcvAdrCh1	チャネルアドレス（上位 2 けた）
*_Skt1SendRcvAdrCh2	チャネルアドレス（下位 2 けた）
*_Skt1Timeout	タイムアウト値
*_Skt1RespCode	終了コード

～

ソケットサービスパラメータ 8 の場合のデバイス変数

CJ ユニットデバイス変数名	名称
*_Skt8ParamArea	ソケットサービスパラメータエリア 8
*_Skt8No	UDP/TCP ソケット No.
*_Skt8Option	ソケットオプション
*_Skt8KeepAlive	Keep-alive 機能使用
*_Skt8Linger	Linger 機能使用
*_Skt8LocalPortNo	自 UDP/TCP ポート No.
*_Skt8RemoteIPAdr	相手 IP アドレス
*_Skt8RemoteIPAdr1	相手 IP アドレスの 1 バイトめ
*_Skt8RemoteIPAdr2	相手 IP アドレスの 2 バイトめ
*_Skt8RemoteIPAdr3	相手 IP アドレスの 3 バイトめ
*_Skt8RemoteIPAdr4	相手 IP アドレスの 4 バイトめ
*_Skt8RemotePortNo	相手 UDP/TCP ポート No.
*_Skt8SendRcvByte	送信／受信バイト数
*_Skt8SendRcvAdr	送信／受信データアドレス
*_Skt8SendRcvAdrType	エリア種別
*_Skt8SendRcvAdrCh1	チャネルアドレス（上位 2 けた）
*_Skt8SendRcvAdrCh2	チャネルアドレス（下位 2 けた）
*_Skt8Timeout	タイムアウト値
*_Skt8RespCode	終了コード

ソケットサービス別の必須パラメーター覧

各サービスで必要なパラメータと、ソケットサービスの対応は、以下のとおりです。

必要なパラメータ	型	パラメータ範囲 ( )内は10進数	ソケットサービス								
			UDP オープン	UDP 受信	UDP 送信	UDP クローズ	TCP オープン PASSIVE	TCP オープン ACTIVE	TCP 受信	TCP 送信	TCP クローズ
UDP/TCPソケットNo.	USINT	1～8 Hex	W	W	W	W	W	W	W	W	W
ソケットオプション	BYTE	ビット指定	－	－	－	－	W	W	－	－	－
自UDP/TCPポートNo.	WORD	0～FFFF Hex (0～65535)	W	－	－	－	W	RW	－	－	－
相手IPアドレス	DWORD	0～FFFFFFFF Hex (0.0.0.0～255.255.255.255)	－	R	W	－	RW	W	－	－	－
相手UDP/TCPポートNo.	WORD	0～FFFF Hex (0～65535)	－	R	W	－	RW	W	－	－	－
送信／受信バイト数	WORD	0～07C0 Hex （0～1,984バイト）	－	RW	RW	－	－	－	RW	RW	－
送信／受信データアドレス	DWORD	「13-6-4 パラメータ詳細」を参照	－	W	W	－	－	－	W	W	－
タイムアウト値（単位：100ms）	WORD	0～FFFF Hex (0～65535) 0: 無制限 0.1秒～6553.5秒	－	W		－	W	－	W	－	－
終了コード	WORD		R	R	R	R	R	R	R	R	R

W : ユーザがパラメータを書き込むために必要
RW: 実行開始時にユーザがパラメータを書き込み、完了時に結果をユーザが読み出すために必要
R : 完了時に結果をユーザが読み出すために必要
－ : 必要なし

13-6-4　パラメータ詳細

● UDP/TCP ソケット No.

オープンする UDP または TCP のソケット No. を指定します。

● ソケットオプション

TCP ソケットのオープン要求（PASSIVE/ACTIVE）コマンドで、keep-alive 機能の使用を指定します。
keep-alive 機能を使用するには、*_KeepAlive を 1(TRUE) にします。
linger 機能を使用するには、*_Linger を 1(TRUE) にします。

● 自 UDP/TCP ポート No.

ソケットが送受信を行う UDP または TCP ポート No. を指定します。
ただし、以下の点に注意してください。

- UDPソケットのオープン要求のときは、FINS用UDPポートNo.（デフォルト値9600）およびCIP通信用 UDP ポート No.（デフォルト値 2222, 44818）として使用しているポートは指定しないでください。
- TCPソケットのオープン要求のときは、FTPサーバ用TCPポートNo.20と21、およびFINS用TCPポート No.（デフォルト値 9600）は指定しないでください。
- 通常は 1,024 以上を使用します。

TCP オープン（ACTIVE）の場合に 0 を指定すると、自動的に空いている TCP ポート No. が割り当てられます。このときに割り当てられた TCP ポート No. は、パラメータエリアの「自 UDP/TCP ポート No.」に格納されます（実際に割り当てられた No. が、ユーザが設定した値（0）に上書きされます）。

● 相手 IP アドレス

相手先の IP アドレスを指定します。
ただし、以下の点に注意してください。

- UDP ソケットの受信要求のときは、コマンド発行時にはこのパラメータは使用されませんが、レスポンス受信時のデータに、送信元 IP アドレスが格納されます。
- この場合、「相手 IP アドレス」のデバイス変数に反映されます。
- TCPソケットのオープン要求（PASSIVE）のときは、指定する相手IPアドレスと相手TCPポートNo. の組み合わせによって、その後の処理が下表のように変化します。

相手IPアドレス	相手TCPポートNo.	処理内容
0	0	すべてのコネクト要求を受け付ける
0	0以外	ポートNo.が一致するときだけ受け付ける
0以外	0	IPアドレスが一致するときだけ受け付ける
0以外	0以外	ポートNo.とIPアドレスの両方が一致するときだけ受け付ける。

相手 IP アドレスに「0」を設定すると、相手ノードを特定しないで、任意のノードから接続するのを待つようになります。
このときは接続してきた相手IPアドレスがパラメータエリアの「相手IPアドレス」に格納されます。
「0」以外を設定すると、指定した相手ノードが接続してくるのを待つようになります。
相手 TCP ポート No. に「0」を設定すると、相手ノードの TCP ポート No. を特定しないようになります。
「0」以外を設定すると、指定 TCP ポート No. での接続を待つようになります。

● 相手 UDP/TCP ポート No.

相手先の UDP または TCP のポート No. を指定します。
ただし、以下の点に注意してください。

- UDP ソケットの受信要求のときは、コマンド発行時にはこのパラメータは使用されませんが、要求完了時に送信元 UDP ポート No. がパラメータエリアの「相手 UDP/TCP ポート No.」に格納されます。
- TCPソケットのオープン要求（PASSIVE）のときは、指定する相手IPアドレスと相手TCPポートNo. の組合せによって、その後の処理が変化します。（上の「相手 IP アドレス」を参照）「0」を設定した場合は、相手 TCP ポート No. はパラメータエリア上の「相手 UDP/TCP ポート No.」に格納されます。

● 送信／受信バイト数

送信要求または受信要求時の、送信または受信データのバイト数を指定します。
完了時には、実際に送受信したデータサイズが格納されます。

● 送信／受信データアドレス

送信要求時の送信データ、または受信要求時の受信データを格納する変数に AT(割付先) 指定したメモリアドレスを、次のように指定します。

設定項目	設定内容
エリア種別	下表の「エリア種別」の値を設定
チャネルアドレス（上位2けた）	下表の「対応アドレス」の値の上位2けたを設定
チャネルアドレス（下位2けた）	下表の「対応アドレス」の値の下位2けたを設定

指定できるエリア範囲と設定値は、以下のとおりです。

変数エリア	データの種類		エリア種別	チャネル番号	対応アドレス
リレーエリア	1チャネルの現在値	CIO	B0	0000～6143	0000～17FF
		HR	B2	H000～H511	0000～01FF
DMエリア	1チャネルの現在値	DM	82	D00000～D32767	0000～7FFF
EMエリア	1チャネルの現在値	バンク0	A0	E0_E00000～E0_E32767	0000～7FFF
		バンク1	A1	E1_E00000～E1_E32767	0000～7FFF
		バンク2	A2	E2_E00000～E2_E32767	0000～7FFF
		バンク3	A3	E3_E00000～E3_E32767	0000～7FFF
		バンク4	A4	E4_E00000～E4_E32767	0000～7FFF
		バンク5	A5	E5_E00000～E5_E32767	0000～7FFF
		バンク6	A6	E6_E00000～E6_E32767	0000～7FFF
		バンク7	A7	E7_E00000～E7_E32767	0000～7FFF
		バンク8	A8	E8_E00000～E8_E32767	0000～7FFF
		バンク9	A9	E9_E00000～E9_E32767	0000～7FFF
		バンクA	AA	EA_E00000～EA_E32767	0000～7FFF
		バンクB	AB	EB_E00000～EB_E32767	0000～7FFF
		バンクC	AC	EC_E00000～EC_E32767	0000～7FFF

ソケットサービス機能では、CJ ユニット用メモリ上のデータをワード単位で扱います。したがって、AT 指定する変数の型には WORD などの 2 バイトのデータ型を指定してください。データ型については「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)」の「第 6 章 プログラミング」を参照してください。

例：DM エリアの D1000 を送受信アドレスとし、100 バイト（50 ワード）の WORD 型の送信／受信データを格納する変数名を SendRcvDat とする場合

- DM エリアのエリア種別は 82 Hex、チャネルアドレスは 1000(03E8 Hex) であるため、以下のように設定します。

設定項目	設定値
エリア種別	82
チャネルアドレス（上位2けた）	03
チャネルアドレス（下位2けた）	E8

- 100 バイト（50 ワード）の WORD 型のグローバル変数 SendRcvDat を作成し、チャネル番号 D1000 Hex を AT（割付先）指定します。



● **タイムアウト値**

受信（TCP、UDP）、オープン（PASSIVE）時ソケットサービス要求スイッチを TRUE にしてから、完了するまでの制限時間を 0.1 秒単位で指定します。

タイムアウトした場合は、終了コードに 0080 Hex（タイムアウト発生）が格納されます。

「0」が指定された場合は、ユニットは要求されたサービスのタイムアウトを監視しないようになります。

13-6-5　ソケットサービス要求スイッチ（CPU ユニット⇔ユニット）

デバイス変数操作によってソケットサービス要求を発行する場合は、下記のソケットサービス要求スイッチを操作します。

ソケット No.1

CJ ユニットデバイス変数名	名称	機能
*_Skt1Cmd	ソケットサービス要求スイッチ 1	ソケットサービス要求を実行します。
*_Skt1UdpOpenCmd	UDP オープン要求スイッチ	FALSE → TRUE の変化で UDP オープン実行 FALSE: オープン完了時 初期値： FALSE
*_Skt1TcpPassiveCmd	TCP オープン（Passive）要求スイッチ	FALSE → TRUE の変化で TCP オープン実行 FALSE: オープン完了時（コネクション時） 初期値： FALSE
*_Skt1TcpActiveCmd	TCP オープン（Active）要求スイッチ	FALSE → TRUE の変化で TCP オープン実行 FALSE: オープン完了時（コネクション時） 初期値： FALSE
*_Skt1SendCmd	送信要求スイッチ	FALSE → TRUE の変化で送信実行（UDP か TCP かはオープン時に決定） FALSE: 送信完了時 初期値： FALSE
*_Skt1RcvCmd	受信要求スイッチ	FALSE → TRUE の変化で受信実行（UDP か TCP かはオープン時に決定） FALSE: 受信完了時 初期値： FALSE
*_Skt1CloseCmd	クローズ要求スイッチ	FALSE → TRUE の変化でクローズ実行 FALSE: クローズ完了時 初期値： FALSE

～

ソケット No.8

CJ ユニットデバイス変数名	名称	機能
*_Skt8Cmd	ソケットサービス要求スイッチ 8	ソケットサービス要求を実行します。
*_Skt8UdpOpenCmd	UDP オープン要求スイッチ	FALSE → TRUE の変化で UDP オープン実行 FALSE: オープン完了時 初期値： FALSE
*_Skt8TcpPassiveCmd	TCP オープン（Passive）要求スイッチ	FALSE → TRUE の変化で TCP オープン実行 FALSE: オープン完了時（コネクション時） 初期値： FALSE
*_Skt8TcpActiveCmd	TCP オープン（Active）要求スイッチ	FALSE → TRUE の変化で TCP オープン実行 FALSE: オープン完了時（コネクション時） 初期値： FALSE

CJ ユニットデバイス変数名	名称	機能
*_Skt8SendCmd	送信要求スイッチ	FALSE → TRUE の変化で送信実行（UDP か TCP かはオープン時に決定） FALSE: 送信完了時 初期値： FALSE
*_Skt8RcvCmd	受信要求スイッチ	FALSE → TRUE の変化で受信実行（UDP か TCP かはオープン時に決定） FALSE: 受信完了時 初期値： FALSE
*_Skt8CloseCmd	クローズ要求スイッチ	FALSE → TRUE の変化でクローズ実行 FALSE: クローズ完了時 初期値： FALSE

上記のように、各ソケットサービス要求スイッチは、要求処理が完了すると EtherNet/IP ユニットによって TRUE → FALSE にされます。

参考

- ソケットサービス要求スイッチの他に、ソケットを制御できるスイッチとして、ユニットコントロールスイッチのソケット強制クローズスイッチ (*_SktForceCloseCmd) があります。このデバイス変数を FALSE → TRUE と操作することで、オープンしているすべてのソケットを強制的にクローズすることができます。詳細は「第 3 章 CJ ユニットデバイス変数への割付」を参照してください。
- デバイス変数操作によるソケットサービスを利用する場合は、各ソケットサービス要求スイッチが TRUE → FALSE となったときに、終了コードを確認するようにプログラムを組んでください。

13-6-6　終了コード

ソケットサービス要求処理完了時には、ソケットサービスパラメータエリア（デバイス変数操作時）に、処理結果が「終了コード」として格納されます。
以下に、サービス要求ごとの終了コードを示します。

● UDP ソケットのオープン要求

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	本体が異常のため実行できない
1100	「UDPソケットNo.」が1～8以外である 「自UDPポートNo.」が0である
110C	他の処理中に、要求スイッチをTRUEにした
220F	指定されたソケットがオープン済みである
2211	ユニットがビジーのためサービスを実行できない
2606	指定された「ソケットNo.」がTCPオープン済みのため、UDPオープンを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0049	UDPポートNo.が重複している(EADDRINUSE)
0081	指定されたソケットがオープン処理中にクローズされた

● UDP ソケットの受信要求

終了コード	内容
0000	正常終了
0302	本体が異常のため実行できない

終了コード	内容
1100	「受信バイト数」が許容範囲外である
1101	「送信/受信データアドレス」のエリア種別が許容範囲外である
1103	「送信/受信データアドレス」のビットアドレスが0でない
110C	他の処理中に要求スイッチをTRUEにした
220F	指定されたソケットが受信処理中である
2210	指定されたソケットが未オープンである
2211	ユニットがビジーのためサービスを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0066	内部処理用のメモリが確保できないためサービスを実行できない
0080	受信要求のタイムアウトが発生した
0081	指定されたソケットが受信処理中にクローズされた

● UDP ソケットの送信要求

終了コード	内容
0000	正常終了
0302	本体が異常のため実行できない
1100	「送信バイト数」が許容範囲外である 「送信先IPアドレス」が0である
1101	「送信/受信データアドレス」のエリア種別が許容範囲外である
1103	「送信/受信データアドレス」のビットアドレスが0でない
110C	他の処理中に、要求スイッチをTRUEにした
220F	指定されたソケットが送信処理中である
2210	指定されたソケットが未オープンである
2211	ユニットがビジーのためサービスを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042	送信先IPアドレスがブロードキャストアドレスで、かつ送信バイト数が1472バイトを超えている(EMSGSIZE)
004C	ネットIDが間違っている 送信先IPアドレスが間違っている(EADDRNOTAVAIL)
004E	ネットIDがIPルータテーブルにない ルータの指定が間違っている 送信先IPアドレスが間違っている(ENETUNREACH)
0051	ルータの指定が間違っている 送信先IPアドレスが間違っている(EHOSTUNREACH)
0081	指定されたソケットが送信処理中にクローズされた

● UDP ソケットのクローズ要求

終了コード	内容
0000	正常終了
0302	本体が異常のため実行できない
2210	指定されたソケットが未オープンである
2211	ユニットがビジーのためサービスを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である

● TCP ソケットのオープン要求 (PASSIVE)

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	本体が異常のため実行できない

終了コード	内容
1100	「TCPソケットNo.」が1～8以外である 「自TCPポートNo.」が0である
110C	他の処理中に要求スイッチをTRUEにした
220F	指定されたソケットがオープン（コネクト）済みか、またはオープン処理中である
2211	ユニットがビジーのためサービスを実行できない
2606	指定された「ソケットNo.」がUDPオープン済みのため、TCPオープンを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042 *1	エラーが発生した(EMSGSIZE)
0045	相手ノードとの通信エラーが発生した(ECONNABORTED)
0049	ポートNo.が重複している(EADDRINUSE)
004A *1	エラーが発生した(ECONNREFUSED)
004B *1	相手ノードとの通信エラーが発生した(ECONNRESET)
004E *1	相手先IPアドレスのパラメータエラーが発生した(ENETUNREACH)
0051 *1	相手先IPアドレスのパラメータエラーが発生した(EHOSTUNREACH)
0053	相手ノードとの通信エラーが発生した(ETIMEDOUT) 相手が存在しない
0066	内部処理用のメモリが確保できないためサービスを実行できない
0080	オープン要求のタイムアウトが発生した
0081	オープン処理中にクローズされた
0082	指定した相手先とコネクションが確立できなかった

*1. 多階層にわたる大規模なネットワークを構築しなければ発生しません。

● TCP ソケットのオープン要求 (ACTIVE)

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	本体が異常のため実行できない
1100	「TCPソケットNo.」が1～8以外である 「相手IPアドレス」が0である
110C	他の処理中に要求スイッチをTRUEにした
220F	指定されたソケットがオープン済みか、またはオープン処理中である
2211	ユニットがビジーのためサービスを実行できない
2606	指定された「ソケットNo.」がUDPオープン済みのため、TCPオープンを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である
000D	相手先IPアドレスのパラメータエラーが発生した(EACCES)
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042 *1	エラーが発生した(EMSGSIZE)
0044	ICMPのデータを受信した(ENOPROTOOPT)
0045	自ソケットがクローズされた(ECONNABORTED)
0049	ポートNo.が重複している(EADDRINUSE)
004A	エラーが発生した(ECONNREFUSED) 相手がPASSIVEオープンで待っていない
004B *1	相手ノードとの通信エラーが発生した(ECONNRESET)
004C	相手先IPアドレスのパラメータエラーが発生した(EADDRNOTAVAIL) パラメータの指定間違い 自ノードの自TCPポートに対してACTIVEオープンしようとした
004E	相手先IPアドレスのパラメータエラーが発生した(ENETUNREACH) ネットIDがIPルータテーブルにないか、ルータの設定間違い
0051	相手先IPアドレスのパラメータエラーが発生した(EHOSTUNREACH) ルータの設定間違い

終了コード	内容
0053	相手ノードとの通信エラーが発生した(ETIMEDOUT) 相手が存在しない
0081	オープン処理中にクローズされた

*1. 多階層にわたる大規模なネットワークを構築しなければ発生しません。

● TCP ソケットの受信要求

終了コード	内容
0000	正常終了
0302	本体が異常のため実行できない
1100	「受信バイト数」が許容範囲外である
1101	「送信/受信データアドレス」のエリア種別が許容範囲外である
1103	「送信/受信データアドレス」のビットアドレスが0でない
110C	他の処理中に要求スイッチをTRUEにした
220F	指定されたソケットが受信処理中である
2210	指定されたソケットのコネクションが確立していない
2211	ユニットがビジーのためサービスを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042 *1	ICMPのデータを受信した(EMSGSIZE)
0044 *1	ICMPのデータを受信した(ENOPROTOOPT)
0045 *1	相手ノードとの通信エラーが発生した(ECONNABORTED)
004B	相手ノードとの通信エラーが発生した(ECONNRESET)
004E *1	ICMPのデータを受信した(ENETUNREACH)
004F *1	ICMPのデータを受信した(EHOSTDOWN)
0051 *1	ICMPのデータを受信した(EHOSTUNREACH)
0053	相手ノードとの通信エラーが発生した(ETIMEDOUT)
0066	内部処理用のメモリが確保できないためサービスを実行できない
0080	受信要求のタイムアウトが発生した
0081	指定されたソケットが受信処理中にクローズされた

*1. 多階層にわたる大規模なネットワークを構築しなければ発生しません。

● TCP ソケットの送信要求

終了コード	内容
0000	正常終了
0302	本体が異常のため実行できない
1100	「送信バイト数」が許容範囲外である
1101	「送信/受信データアドレス」のエリア種別が許容範囲外である
1103	「送信/受信データアドレス」のビットアドレスが0でない
110C	他の処理中に要求スイッチをTRUEにした
220F	指定されたソケットが送信処理中である
2210	指定されたソケットのコネクションが確立されていない
2211	ユニットがビジーのためサービスを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である
0020	送信中、相手ソケットとのコネクションが切断された(EPIPE)
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042 *1	エラーが発生した(EMSGSIZE)
0044 *1	ICMPのデータを受信した(ENOPROTOOPT)
0045 *1	相手ノードとの通信エラーが発生した(ECONNABOTED)
004A	相手ノードとの通信エラーが発生した(ECONNREFUSED)

終了コード	内容
004B *1	相手ノードとの通信エラーが発生した(ECONNRESET)
004E *1	相手先IPアドレスのパラメータエラーが発生した(ENETUNREACH)
004F *1	ICMPのデータを受信した(EHOSTDOWN)
0051 *1	相手先IPアドレスのパラメータエラーが発生した(EHOSTUNREACH)
0053 *1	相手ノードとの通信エラーが発生した(ETIMEDOUT)
0081	指定されたソケットが送信処理中にクローズされた

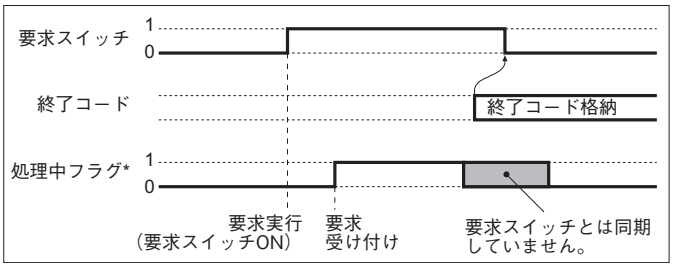
*1. 多階層にわたる大規模なネットワークを構築しなければ発生しません。

● TCP ソケットのクローズ要求

終了コード	内容
0000	正常終了
0302	本体が異常のため実行できない
2210	指定されたソケットのコネクションが確立していない
2211	ユニットがビジーのためサービスを実行できない
2607	指定された「ソケットサービスパラメータエリア」が、他のソケットにより使用中である

13-6-7　タイミングチャート

デバイス変数操作の場合の、各要求スイッチと、対応する処理中フラグの変化のタイミング、終了コードの格納の状態を下図に示します。



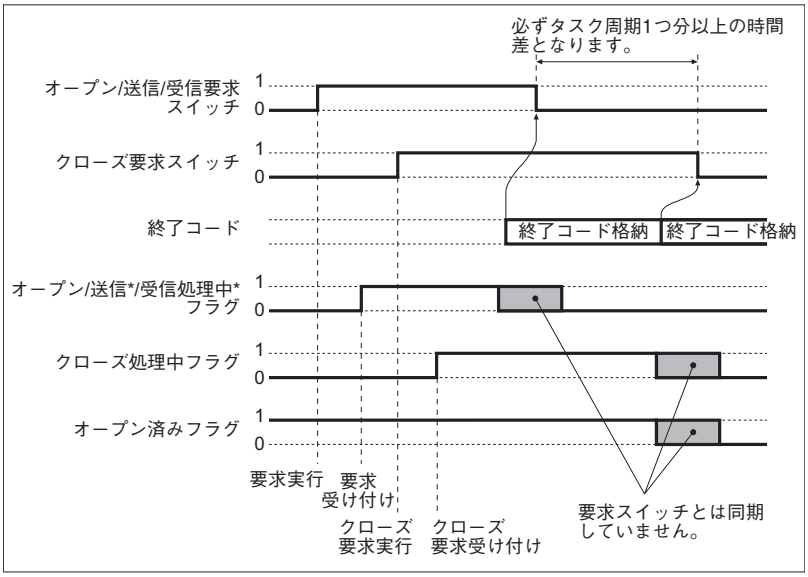
（注）ソケットサービス高速化オプションを有効にした場合、送信処理中および受信処理中フラグは TRUE しません。このため、送信／受信要求スイッチの TRUE → FALSE でのみプログラムを制御します。

要求中にクローズ処理を行ったとき

オープン／送信／受信要求中であっても、クローズ要求スイッチや、強制クローズスイッチによるクローズ処理を行うことができます（他の要求の処理中は、クローズ処理のみが可能です）。

● クローズ要求スイッチの場合

クローズ要求スイッチを使用した場合は、結果が終了コードとして格納されます。そのため、中断された処理の要求スイッチの FALSE と、クローズ要求スイッチの FALSE は、必ず 1 タスク周期分以上の時間差がとられ、それぞれの終了コードを読み出すことができるようになっています。



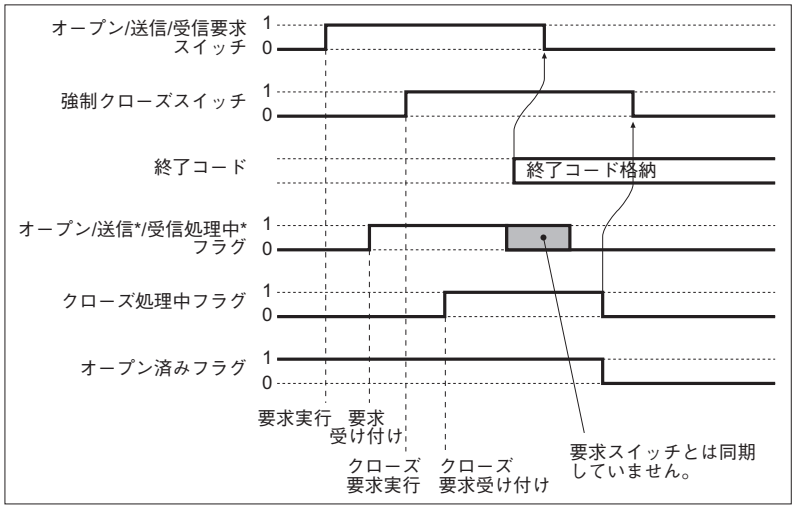
*ソケットサービス高速化オプションを有効にした場合、送信処理中および受信処理中フラグは TRUE にしません。

参考

オープン要求中にクローズ要求を行った場合は、オープン済みフラグは FALSE のままです。

● 強制クローズスイッチの場合

強制クローズスイッチを使用した場合は、要求中の処理が中断され、終了コードとして格納されます。



*ソケットサービス高速化オプションを有効にした場合、送信処理中および受信処理中フラグは TRUE にしません。

参考

オープン要求中に強制クローズを行った場合は、オープン済みフラグは FALSE のままです。

13-6-8　サンプルプログラム

デバイス変数操作によるソケットサービスのサンプルプログラムについては、「A-9 ソケットサービス機能のサンプルプログラム」を参照してください。

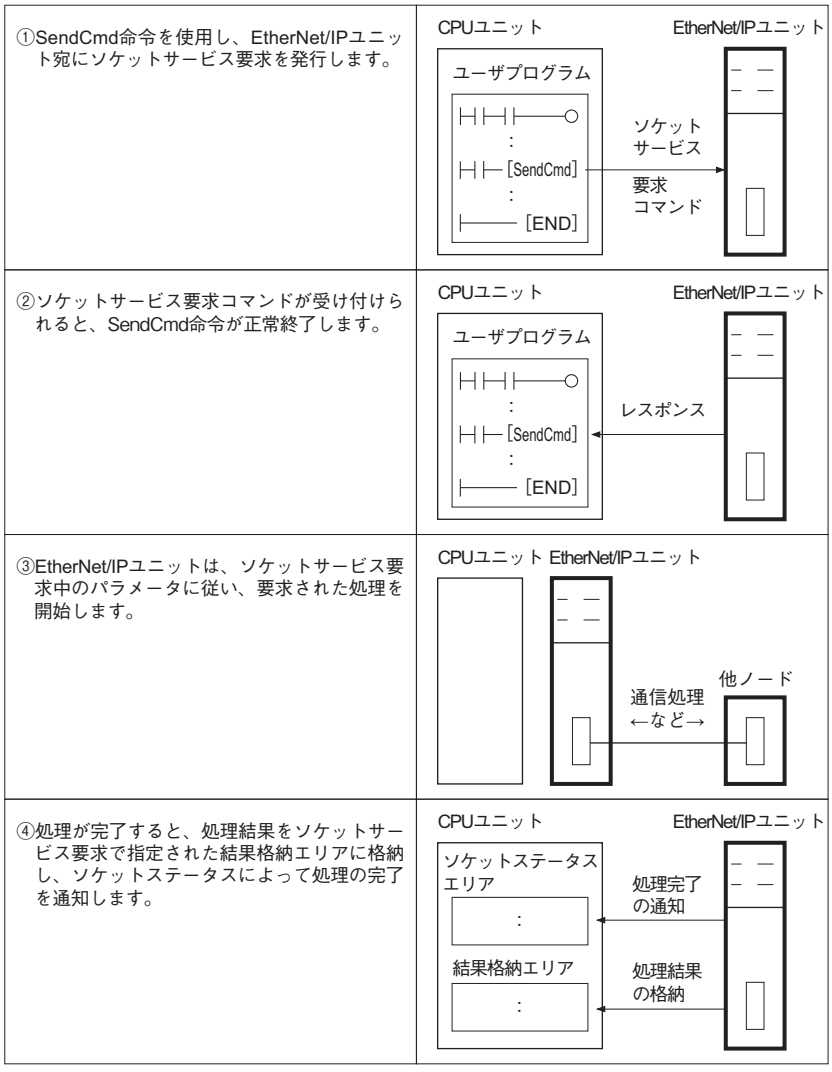
13-7 SendCmd 命令によるソケットサービスの利用の詳細

13-7-1 ソケットサービスの使用方法

EtherNet/IP ユニットでは、TCP、UDP 各々 8 つのソケットを持ちます。これらのソケットを使用して通信を行うために、オープン、クローズ、送信、受信の 4 つの処理が用意されています。

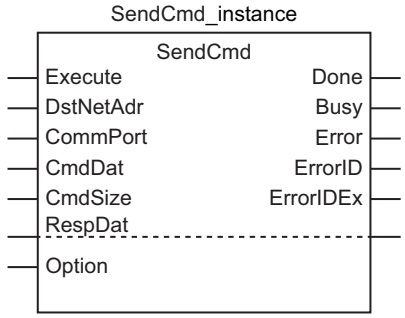
処理	説明
オープン処理	使用するソケットNo.を指定して、そのソケットを使用できる状態にします。ソケットサービスを行うときに、最初に実行する処理です。TCPでは、コネクションの確立までを行います。
クローズ処理	ソケットの使用を終了します。TCPでは、コネクションの切断を行います。
送信処理	オープン済みのソケットを指定し、そのソケットからデータを送信します。
受信処理	オープン済みのソケットを指定し、そのソケットからデータを受信します。

これらの処理を行うには、EtherNet/IP ユニット宛に SendCmd 命令を実行します。SendCmd 命令を開始してから完了するまでの処理の流れを次に示します。



13-7-2　ソケットサービス要求

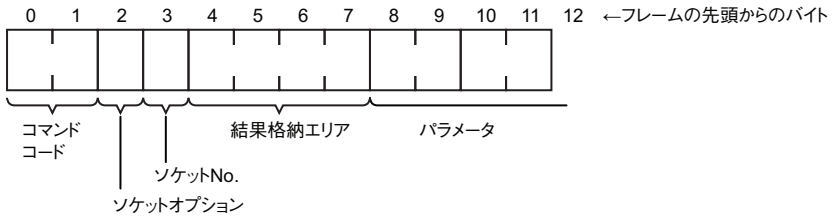
ソケットサービス要求時に、SendCmd 命令の入力変数に共通で設定する値は次のとおりです。「各サービスごと」と記載している箇所については、各ソケットサービス要求ごとに後述します。
各パラメータの詳細は『NJ/NX シリーズ コマンドリファレンスマニュアル 基本編 (SBCA-468)』を参照してください。



変数		内容	設定	備考
DstNetAdr		送信先ネットワークアドレス	−	
	NetNo	ネットワークアドレス	0	構造体_sDNET_ADRの初期値のまま利用可能です。
	NodeNo	ノードアドレス	0	構造体_sDNET_ADRの初期値のまま利用可能です。
	UnitNo	号機アドレス	ユニットのロータリスイッチで設定した"UNIT No."+16#10を指定してください。	例："UNIT No."を"A"に設定した場合、16#1Aを指定します。
CommPort		送信先シリアルポートの指定	_NONE	
ComdDat[]配列		送信するコマンド	各サービスごと	
CmdSize		コマンドデータサイズ	各サービスごと	
RespDat[]配列		レスポンス格納配列	各サービスごと	
Option		レスポンス	−	
	isNonResp	レスポンス不要	FALSE	構造体_sRESPONSEの初期値のまま利用可能です。
	TimeOut	タイムアウト時間	20	構造体_sRESPONSEの初期値のまま利用可能です。
	Retry	再送回数	0	構造体_sRESPONSEの初期値のまま利用可能です。

コマンドの基本フォーマット

コマンドの基本フォーマットは次のとおりです。



● コマンドコード

要求する処理のコードを指定します。

● ソケットオプション

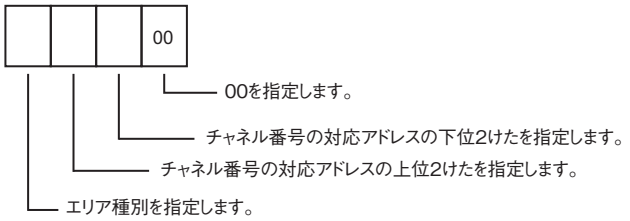
TCP ソケットのオープン要求 (PASSIVE/ACTIVE) コマンドで、keep-alive や linger 機能の使用を指定します。その他のコマンドには有効なオプションはなく、0 で使用します。

● ソケット No.

処理を要求するソケット No. を指定します。No. は 1 ～ 8 の範囲の値です。

● 結果格納エリア

要求した処理の結果を格納するエリアを指定します。
結果格納エリアの 4 バイトは、次のように構成されています。



エリア種別、チャネル番号の対応アドレスの指定方法は、「●送信／受信データアドレス」(P.13-20)と同様です。

● パラメータ

コマンドコードに定義されているパラメータを設定します。詳細は、「13-7 SendCmd 命令によるソケットサービスの利用の詳細」を参照してください。

UDP ソケットのオープン要求

UDP ソケットのオープン処理を要求します。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	UDPソケットのオープン要求のコマンドコードは16#2701です。	BYTE#16#27
CmdDat[1]			BYTE#16#01
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。 有効な設定はなく、0で指定してください。	BYTE#16#00

変数	項目	説明	値
CmdDat[3]	UDPソケットNo.	オープンするUDPソケットのNo.（1～8）を1バイトで指定します。	
CmdDat[4]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。 結果格納領域に格納される内容については後述します。	指定方法は、「 コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[5]			
CmdDat[6]			
CmdDat[7]			
CmdDat[8]	自UDPポートNo.	ソケットが送受信を行うUDPポートNo.を2バイト（0以外）で指定します。	
CmdDat[9]			
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#10

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#01
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1100	「UDPソケットNo.」が許容範囲外である 「自UDPポートNo.」が0である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
220F	指定されたソケットがオープン済み、またはクローズ処理中である
2211	ユニットの負荷が高いため、サービスを実行できない

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。
フォーマットは以下の通りです。



結果格納領域には終了コードが格納されます。

終了コード	内容
0000	正常終了
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0049	UDPポートNo.が重複している(EADDRINUSE)

UDP ソケットの受信要求

UDP ソケットの受信処理を要求します。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	UDPソケットの受信要求のコマンドコードは16#2702です。	BYTE#16#27
CmdDat[1]			BYTE#16#02
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。有効な設定はなく、0で指定してください。	BYTE#16#00
CmdDat[3]	UDPソケットNo.	受信するUDPソケットのNo.（1～8）を1バイトで指定します。	
CmdDat[4]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。 結果格納領域に格納される内容については後述します。	指定方法は、「コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[5]			
CmdDat[6]			
CmdDat[7]			
CmdDat[8]	受信バイト数	受信するデータのバイト数を指定します。最大で1984バイトまで指定できます。 結果格納領域には、受信したデータのバイト数が格納されます。	
CmdDat[9]			
CmdDat[10]	タイムアウト値	受信要求の受付から結果格納が行われるまでの制限時間を指定します。 制限時間になった場合、結果格納の終了コードが「タイムアウトエラー」になります。 単位は0.1秒です。0を指定すると「制限時間なし」になります。	
CmdDat[11]			
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#12

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

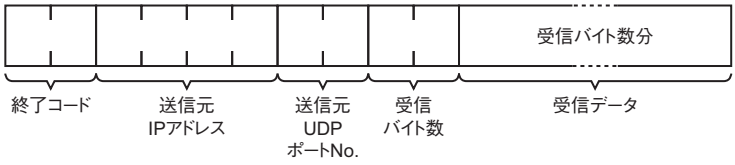
変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#02
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1100	「UDPソケットNo.」または「受信バイト数」が許容範囲外である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
220F	指定されたソケットが受信処理中である
2210	指定されたソケットが未オープンである
2211	ユニットの負荷が高いため、サービスを実行できない
2240	モードが異なるため、サービスが実行できない。（ソケットサービス高速化オプションを有効にしてSendCmd命令によるソケットサービスを使用した） または、割付リレーエリアのレイアウトタイプが［デフォルト］のときに、SendCmd命令によるソケットサービスを実行した。

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。
フォーマットは以下の通りです。



結果格納領域には、それぞれ以下が格納されます。

- 終了コード

終了コード	内容
0000	正常終了
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0066	内部処理用のメモリが確保できないためサービスを実行できない
0080	受信要求のタイムアウトが発生した
0081	指定されたソケットが受信処理中にクローズされた

- 送信元 IP アドレス
  送信元の IP アドレスが 16 進数で格納されます。

- 送信元 UDP ポート No.
  送信元の UDP ポート No. が格納されます。

- 受信バイト数
  受信したデータのバイト数が格納されます。

- 受信データ
  受信したデータが格納されます。

使用上の注意

コマンドの「受信バイト数」で指定した値より大きいバイト数のパケットを受信したときは、「受信バイト数」分だけソケットに格納し、残りのデータは破棄されます。

UDP ソケットの送信要求

UDP ソケットの送信処理を要求します。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	UDPソケットの送信要求のコマンドコードは16#2703です。	BYTE#16#27
CmdDat[1]			BYTE#16#03
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。有効な設定はなく、0で指定してください。	BYTE#16#00
CmdDat[3]	UDPソケットNo.	受信するUDPソケットのNo.（1～8）を1バイトで指定します。	

変数	項目	説明	値
CmdDat[4]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。 結果格納領域に格納される内容については後述します。	指定方法は、「コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[5]			
CmdDat[6]			
CmdDat[7]			
CmdDat[8]	送信先IPアドレス	送信先のIPアドレスを16進数で指定します。	
CmdDat[9]			
CmdDat[10]			
CmdDat[11]			
CmdDat[12]	送信先UDPポートNo.	送信先のUDPポートNo.を指定します。	
CmdDat[13]			
CmdDat[14]	送信バイト数	送信するデータのバイト数を指定します。最大で1984バイトまで指定できます。送信先IPアドレスにブロードキャスト（一斉同報）アドレスを指定した場合は、最大1472バイトです。 結果格納領域には、実際に送信されたデータのバイト数が格納されます。	
CmdDat[15]			
CmdDat[16] ～	送信データ	相手ノードに送信するデータを指定します。	
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#（CmdDatの配列サイズ）

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

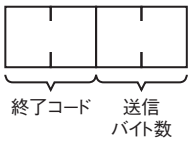
変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#03
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1003	「送信バイト数」と「送信データの長さ」が不一致である
1100	「UDPソケットNo.」または「受信バイト数」が許容範囲外である 「送信先IPアドレス」が0である 「送信先UDPポートNo.」が0である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
220F	指定されたソケットが送信処理中である
2210	指定されたソケットが未オープンである
2211	ユニットの負荷が高いため、サービスを実行できない
2240	モードが異なるため、サービスが実行できない。（ソケットサービス高速化オプションを有効にしてSendCmd命令によるソケットサービスを使用した） または、割付リレーエリアのレイアウトタイプが［デフォルト］のときに、SendCmd命令によるソケットサービスを実行した。

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。
フォーマットは以下の通りです。



結果格納領域には、それぞれ以下が格納されます。

- 終了コード

終了コード	内容
0000	正常終了
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042	送信先IPアドレスがブロードキャストアドレスで、かつ送信バイト数が1472バイトを超えている(EMSGSIZE)
004C	ネットIDが間違っている 送信先IPアドレスが間違っている(EADDRNOTAVAIL)
004E	ネットIDがIPルータテーブルにない ルータの指定が間違っている 送信先IPアドレスが間違っている(ENETUNREACH)
0051	ルータの指定が間違っている 送信先IPアドレスが間違っている(EHOSTUNREACH)

- 送信バイト数
  実際に送信されたデータのバイト数が格納されます。

UDP ソケットのクローズ要求

UDP ソケットのクローズ処理を要求します。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	UDPソケットのクローズ要求のコマンドコードは16#2704です。	BYTE#16#27
CmdDat[1]			BYTE#16#04
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。 有効な設定はなく、0で指定してください。	BYTE#16#00
CmdDat[3]	UDPソケットNo.	受信するUDPソケットのNo.（1～8）を1バイトで指定します。	
CmdDat[4]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。 結果格納領域に格納される内容については後述します。	指定方法は、「コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[5]			
CmdDat[6]			
CmdDat[7]			
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#8

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

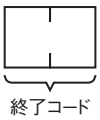
変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#04
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1100	「UDPソケットNo.」が許容範囲外である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
2210	指定されたソケットが未オープンである
2211	ユニットの負荷が高いため、サービスを実行できない
2240	モードが異なるため、サービスが実行できない。（ソケットサービス高速化オプションを有効にしてSendCmd命令によるソケットサービスを使用した） または、割付リレーエリアのレイアウトタイプが［デフォルト］のときに、SendCmd命令によるソケットサービスを実行した。

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。フォーマットは以下の通りです。



結果格納領域には終了コードが格納されます。

終了コード	内容
0000	正常終了

TCP ソケットのオープン要求 (PASSIVE)

TCP ソケットのオープン処理を要求します。ソケットは、他ノードからの接続待ちになります。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	TCPソケットのオープン要求(PASSIVE)のコマンドコードは16#2710です。	BYTE#16#27
CmdDat[1]			BYTE#16#10
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。 指定内容については後述します。	
CmdDat[3]	TCPソケットNo.	受信するTCPソケットのNo.（1～8）を1バイトで指定します。	

変数	項目	説明	値
CmdDat[4] CmdDat[5] CmdDat[6] CmdDat[7]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。 結果格納領域に格納される内容については後述します。	指定方法は、「 コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[8] CmdDat[9]	自TCPポートNo.	ソケットが送受信を行うTCPポートNo.（0以外）を2バイトで指定します。 以下のポートは使用できません。 • FTP サーバ用 TCP ポート No.(20, 21) • DNS サーバアクセス用 TCP ポート No.（デフォルト値 53）	
CmdDat[10] CmdDat[11]	タイムアウト値	オープン要求の受付から結果格納が行われるまでの制限時間を指定します。 制限時間になった場合、結果格納の終了コードが「タイムアウトエラー」になります。 単位は0.1秒です。0を指定すると「制限時間なし」になります。	
CmdDat[12] CmdDat[13] CmdDat[14] CmdDat[15]	相手IPアドレス	相手ノードのIPアドレスを16進数で指定します。 • 00 00 00 00：すべて 0 を設定すると、相手ノードを指定しません。任意のノードから接続するのを待ちます。 • 上記以外：指定した相手ノードが接続をしてくるのを待ちます。	
CmdDat[16] CmdDat[17]	相手TCPポートNo.	相手ノードのTCPポートNo.を指定します。 • 00 00 ：すべて 0 を設定すると、相手ノードの TCP ポート No. を指定しません。 • 上記以外:相手ノードの TCP ポート No. を指定します。	
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#18

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#10
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

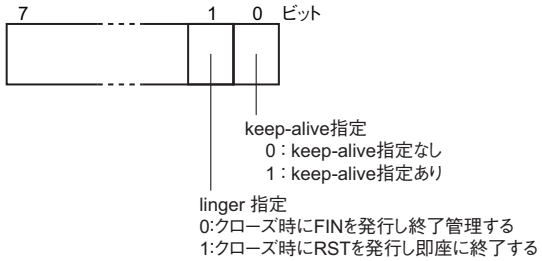
終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1100	「TCPソケットNo.」が許容範囲外である 「自TCPポートNo.」が0である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
220F	指定されたソケットがオープン（コネクト）済みか、オープン処理中またはクローズ処理中である

終了コード	内容
2211	ユニットの負荷が高いため、サービスを実行できない
2240	モードが異なるため、サービスが実行できない。（ソケットサービス高速化オプションを有効にしてSendCmd命令によるソケットサービスを使用した） または、割付リレーエリアのレイアウトタイプが［デフォルト］のときに、SendCmd命令によるソケットサービスを実行した。

● ソケットオプション

ソケットオプションを 1 バイトで指定します。

フォーマットは以下のとおりです。

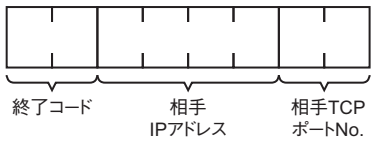


参考

- 通常、linger オプションを指定しない場合、TCP クローズ時に FIN データを発行し、その後の約 1 分間で相手ノードとの間で送達確認等の終了管理を行います。このため、同じポート No. の TCP ソケットを即座に使用できないことがあります。
- これに対し、linger オプションを指定すると、TCP クローズで RST データが送信され即座に終了処理がなされます。このことで、同じポート No. の TCP ソケットのオープンを即座に実行することができます。ただし、クローズ直前の送信データが相手ノードへの送達確認が行われません。liger オプションを指定する場合は、送信データの到達はアプリケーションで確保するようにしてください。

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。
フォーマットは以下の通りです。



結果格納領域には、それぞれ以下が格納されます。

- 終了コード

終了コード	内容
0000	正常終了
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042 *1	エラーが発生した(EMSGSIZE)
0045	相手ノードとの通信エラーが発生した(ECONNABORTED)
0049	ポートNo.が重複している(EADDRINUSE)
004A *1	エラーが発生した(ECONNREFUSED)
004B *1	相手ノードとの通信エラーが発生した(ECONNRESET)

終了コード	内容
004E *1	相手先IPアドレスのパラメータエラーが発生した(ENETUNREACH)
0051 *1	相手先IPアドレスのパラメータエラーが発生した(EHOSTUNREACH)
0053	相手ノードとの通信エラーが発生した(ETIMEDOUT) 相手が存在しない
0066	内部処理用のメモリが確保できないためサービスを実行できない
0080	オープン要求のタイムアウトが発生した
0081	オープン処理中にクローズされた
0082	指定した相手先とコネクションが確立できなかった

*1. 多階層にわたる大規模なネットワークを構築しなければ発生しません。

- 相手 IP アドレス
  接続した相手ノードの IP アドレスが格納されます。

- 相手 TCP ポート No.
  接続した相手ノードの TCP ポート No. が格納されます。

使用上の注意

指定する相手 IP アドレスと相手 TCP ポート No. の組合せによって、その後の処理が下表のように変化します。

相手IPアドレス	相手TCPポートNo.	処理内容
0	0	すべてのコネクト要求を受け付けます。
0	0以外	ポートNo.が一致するときだけ受け付けます。
0以外	0	IPアドレスが一致するときだけ受け付けます。
0以外	0以外	ポートNo.とIPアドレスの両方が一致するときだけ受け付けます。

TPC ソケットのオープン要求 (ACTIVE)

TCP ソケットのオープン処理を要求します。ソケットは他ノードへの接続を行います。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	TCPソケットのオープン要求(ACTIVE)のコマンドコードは16#2711です。	BYTE#16#27
CmdDat[1]			BYTE#16#11
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。指定内容については後述します。	
CmdDat[3]	TCPソケットNo.	受信するTCPソケットのNo.（1～8）を1バイトで指定します。	
CmdDat[4]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。 結果格納領域に格納される内容については後述します。	指定方法は、「コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[5]			
CmdDat[6]			
CmdDat[7]			
CmdDat[8]	自TCPポートNo.	ソケットが送受信を行うTCPポートNo.を2バイトで指定します。FTPサーバ用TCPポートNo.(21)は指定できません。0を指定すると自動的に空いているTCPポートNo.が割り当てられます。	
CmdDat[9]			

変数	項目	説明	値
CmdDat[10]	相手IPアドレス	相手ノードのIPアドレス（0以外）を16進数で指定します。	
CmdDat[11]			
CmdDat[12]			
CmdDat[13]			
CmdDat[14]	相手TCPポートNo.	相手ノードのTCPポートNo.を指定します（0以外）。	
CmdDat[15]			
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#16

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

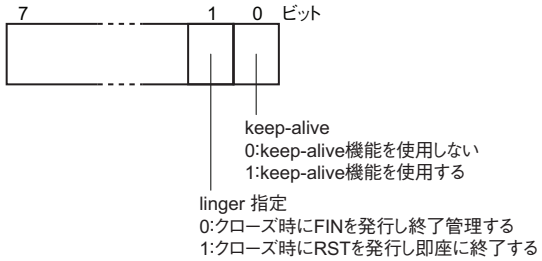
変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#11
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1100	「TCPソケットNo.」が許容範囲外である 「相手IPアドレス」または「相手TCPポートNo.」が0である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
220F	指定されたソケットがオープン済みか、オープン処理中またはクローズ処理中である
2211	ユニットの負荷が高いため、サービスを実行できない
2240	モードが異なるため、サービスが実行できない。（ソケットサービス高速化オプションを有効にしてSendCmd命令によるソケットサービスを使用した） または、割付リレーエリアのレイアウトタイプが［デフォルト］のときに、SendCmd命令によるソケットサービスを実行した。

● ソケットオプション

ソケットオプションを 1 バイトで指定します。
フォーマットは以下のとおりです。

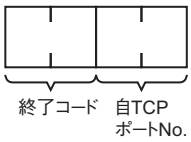


参考

- 通常、linger オプションを指定しない場合、TCP クローズ時に FIN データを発行し、その後の約 1 分間で相手ノードとの間で送達確認等の終了管理を行います。このため、同じポート No. の TCP ソケットを即座に使用できないことがあります。
- これに対し、linger オプションを指定すると、TCP クローズで RST データが送信され即座に終了処理がなされます。このことで、同じポート No. の TCP ソケットのオープンを即座に実行することができます。ただし、クローズ直前の送信データが相手ノードへの送達確認が行われません。liger オプションを指定する場合は、送信データの到達はアプリケーションで確保するようにしてください。

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。
フォーマットは以下の通りです。



結果格納領域には、それぞれ以下が格納されます。

- 終了コード

終了コード	内容
0000	正常終了
000D	相手先IPアドレスのパラメータエラーが発生した(EACCES)
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042 *1	エラーが発生した(EMSGSIZE)
0044	ICMPのデータを受信した(ENOPROTOOPT)
0045	自ソケットがクローズされた(ECONNABORTED)
0049	ポートNo.が重複している(EADDRINUSE)
004A	エラーが発生した(ECONNREFUSED) 相手がPASSIVEオープンで待っていない
004B *1	相手ノードとの通信エラーが発生した(ECONNRESET)
004C	相手先IPアドレスのパラメータエラーが発生した(EADDRNOTAVAIL) パラメータの指定間違い 自ノードの自TCPポートに対してACTIVEオープンしようとした
004E	相手先IPアドレスのパラメータエラーが発生した(ENETUNREACH) ネットID がIPルータテーブルにないか、ルータの設定間違い
0051	相手先IPアドレスのパラメータエラーが発生した(EHOSTUNREACH) ルータの設定間違い
0053	相手ノードとの通信エラーが発生した(ETIMEDOUT) 相手が存在しない
0081	オープン処理中にクローズされた

*1. 多階層にわたる大規模なネットワークを構築しなければ発生しません。

- 自 TCP ポート No.
  オープンしたソケットに割り当てられた TCP ポート No. が格納されます。

TCP ソケットの受信要求

TCP ソケットのデータ受信処理を要求します。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	TCPソケットの受信要求のコマンドコードは16#2712です。	BYTE#16#27
CmdDat[1]			BYTE#16#12
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。有効な設定はなく、0で指定してください。	BYTE#16#00
CmdDat[3]	TCPソケットNo.	受信するTCPソケットのNo.（1～8）を1バイトで指定します。	
CmdDat[4]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。 結果格納領域に格納される内容については後述します。	指定方法は、「コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[5]			
CmdDat[6]			
CmdDat[7]			
CmdDat[8]	受信バイト数	受信するデータのバイト数を指定します。最大で1984バイトまで指定できます。 結果格納領域には、受信したデータのバイト数が格納されます。	
CmdDat[9]			
CmdDat[10]	タイムアウト値	受信要求の受付から結果格納が行われるまでの制限時間を指定します。 制限時間になった場合、結果格納領域の終了コードが「タイムアウトエラー」になります。 単位は0.1秒です。0を指定すると「制限時間なし」になります。	
CmdDat[11]			
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#12

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

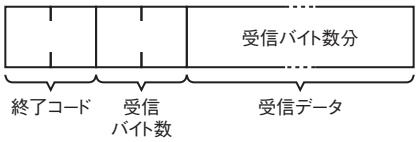
変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#12
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1100	「TCPソケットNo.」または「受信バイト数」が許容範囲外である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
220F	指定されたソケットが受信処理中である
2210	指定されたソケットのコネクションが確立していない
2211	ユニットの負荷が高いため、サービスを実行できない
2240	モードが異なるため、サービスが実行できない。（ソケットサービス高速化オプションを有効にしてSendCmd命令によるソケットサービスを使用した） または、割付リレーエリアのレイアウトタイプが［デフォルト］のときに、SendCmd命令によるソケットサービスを実行した。

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。
フォーマットは以下の通りです。



結果格納領域には、それぞれ以下が格納されます。

- 終了コード

終了コード	内容
0000	正常終了
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042 *1	ICMPのデータを受信した(EMSGSIZE)
0044 *1	ICMPのデータを受信した(ENOPROTOOPT)
0045 *1	相手ノードとの通信エラーが発生した(ECONNABORTED)
004A *1	エラーが発生した(ECONNREFUSED)
004B	相手ノードとの通信エラーが発生した(ECONNRESET)
004E *1	ICMPのデータを受信した(ENETUNREACH)
004F *1	ICMPのデータを受信した(EHOSTDOWN)
0051 *1	ICMPのデータを受信した(EHOSTUNREACH)
0053	相手ノードとの通信エラーが発生した(ETIMEDOUT)
0066	内部処理用のメモリが確保できないためサービスを実行できない
0080	受信要求のタイムアウトが発生した
0081	指定されたソケットが受信処理中にクローズされた

*1. 多階層にわたる大規模なネットワークを構築しなければ発生しません。

- 受信バイト数
  受信したデータのバイト数が格納されます。

- 受信データ
  受信したデータが格納されます。

TCP ソケットの送信要求

TCP ソケットのデータ送信処理を要求します。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	TCPソケットの受信要求のコマンドコードは16#2713です。	BYTE#16#27
CmdDat[1]			BYTE#16#13
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。有効な設定はなく、0で指定してください。	BYTE#16#00
CmdDat[3]	TCPソケットNo.	受信するTCPソケットのNo.（1～8）を1バイトで指定します。	

変数	項目	説明	値
CmdDat[4]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。 結果格納領域に格納される内容については後述します。	指定方法は、「コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[5]			
CmdDat[6]			
CmdDat[7]			
CmdDat[8]	送信バイト数	送信するデータのバイト数を指定します。最大で1984バイトまで指定できます。	
CmdDat[9]			
CmdDat[10] ～	送信データ	送信するデータを指定します。	
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#（CmdDatの配列サイズ）

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

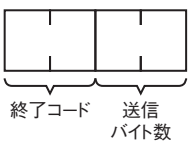
変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#13
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1003	「送信バイト数」と「データ数」が不一致である
1100	「TCPソケットNo.」または「受信バイト数」が許容範囲外である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
220F	指定されたソケットが送信処理中である
2210	指定されたソケットのコネクションが確立していない
2211	ユニットの負荷が高いため、サービスを実行できない
2240	モードが異なるため、サービスが実行できない。（ソケットサービス高速化オプションを有効にしてSendCmd命令によるソケットサービスを使用した） または、割付リレーエリアのレイアウトタイプが［デフォルト］のときに、SendCmd命令によるソケットサービスを実行した。

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。
フォーマットは以下の通りです。



結果格納領域には、それぞれ以下が格納されます。

• 終了コード

終了コード	内容
0000	正常終了
0020	送信中、相手ソケットとのコネクションが切断された(EPIPE)
003E	受信の負荷が高いため内部バッファが確保できない(ENOBUFS)
0042 *1	エラーが発生した(EMSGSIZE)
0044 *1	ICMPのデータを受信した(ENOPROTOOPT)
0045 *1	相手ノードとの通信エラーが発生した(ECONNABORTED)
004A*1	エラーが発生した(ECONNREFUSED)
004B *1	相手ノードとの通信エラーが発生した(ECONNRESET)
004E *1	相手先IPアドレスのパラメータエラーが発生した(ENETUNREACH)
004F *1	ICMPのデータを受信した(EHOSTDOWN)
0051 *1	相手先IPアドレスのパラメータエラーが発生した(EHOSTUNREACH)
0053 *1	相手ノードとの通信エラーが発生した(ETIMEDOUT)
0081	指定されたソケットが送信処理中にクローズされた

*1. 多階層にわたる大規模なネットワークを構築しなければ発生しません。

• 送信バイト数
送信したデータのバイト数が格納されます。

TCP ソケットのクローズ要求

TCP ソケットのクローズ処理を要求します。他の処理を実行中であった場合は、その処理を強制終了したことが結果格納領域に格納されます。

● 入力変数

SendCmd 命令の入力変数に設定する値は次のとおりです。

変数	項目	説明	値
CmdDat[0]	コマンドコード	TCPソケットの受信要求のコマンドコードは16#2714です。	BYTE#16#27
CmdDat[1]			BYTE#16#14
CmdDat[2]	ソケットオプション	ソケットオプションを1バイトで指定します。有効な設定はなく、0で指定してください。	BYTE#16#00
CmdDat[3]	TCPソケットNo.	受信するTCPソケットのNo.（1～8）を1バイトで指定します。	
CmdDat[4]	結果格納領域	EtherNet/IPユニットでのサービス処理結果を格納する領域を指定します。結果格納領域に格納される内容については後述します。	指定方法は、「コマンドの基本フォーマット」(P.13-30)を参照してください。
CmdDat[5]			
CmdDat[6]			
CmdDat[7]			
CmdSize	コマンドサイズ	CmdDatの有効データ長を指定します。	UINT#8

● 入出力変数

SendCmd 命令実行時は、入出力変数に値を設定する必要はありません。
SendCmd 命令実行完了後、入出力変数に設定される値は次のとおりです。

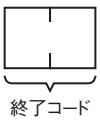
変数	項目	説明	値
RespDat[0]	コマンドコード	CmdDat[0]と同じ値が設定されます。	BYTE#16#27
RespDat[1]		CmdDat[1]と同じ値が設定されます。	BYTE#16#14
RespDat[2]	終了コード	SendCmd命令の終了コードが格納されます。	
RespDat[3]			

終了コードは以下のとおりです。

終了コード	内容
0000	正常終了
0105	自IPアドレスの設定エラーが発生している
0302	CPUユニットが異常のため実行できない
1001	コマンドサイズが大きい
1002	コマンドサイズが小さい
1100	「TCPソケットNo.」が許容範囲外である
1101	「結果格納領域」のエリア種別が許容範囲外である
1103	「結果格納領域」のビットアドレスが0でない
2210	指定されたソケットのコネクションが確立していない
2211	ユニットの負荷が高いため、サービスを実行できない
2240	モードが異なるため、サービスが実行できない。（ソケットサービス高速化オプションを有効にしてSendCmd命令によるソケットサービスを使用した） または、割付リレーエリアのレイアウトタイプが［デフォルト］のときに、SendCmd命令によるソケットサービスを実行した。

● 結果格納領域

指定した結果格納領域には、EtherNet/IP ユニットでのサービス処理結果が格納されます。フォーマットは以下の通りです。



結果格納領域には、終了コードが格納されます。

- 終了コード

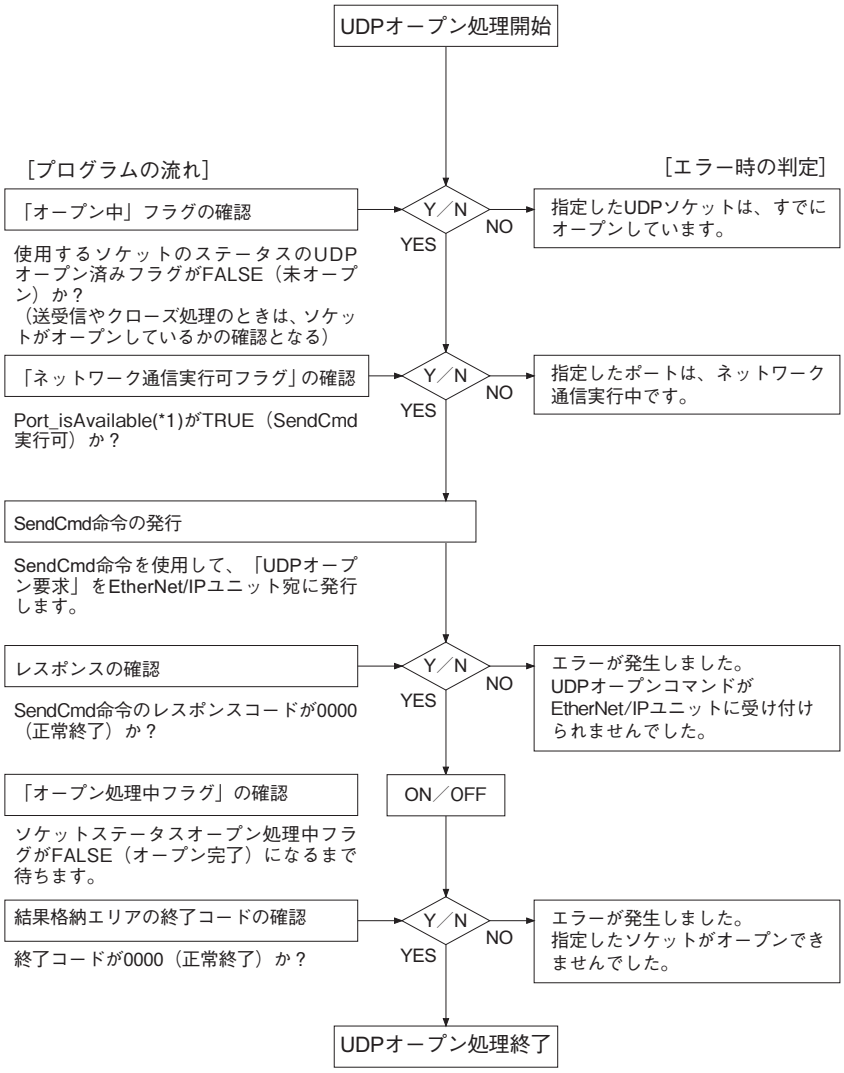
終了コード	内容
0000	正常終了

使用上の注意

データの送受信など、他の処理を実行中にこのコマンドを実行すると、実行中の処理は強制終了され、結果格納領域には強制終了されたことを示す終了コードが格納されます。

13-7-3　ソケットサービスとソケットステータスの使用手順

ソケットサービスは、ソケットステータスのフラグの変化やタイミングに注意しながら使用する必要があります。下図は UDP オープン処理を例にしたときの流れを示しています。
その他のソケットサービスについても同様の手順となります。UDP オープン処理以外のソケットサービスを使用する場合は、フラグ名称などを読み替えてください。



(*1)_Port_isAvailableはSendCmdの実行可否を判断するためのコントローラのシステム定義変数です。
詳細は「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)」を参照してください。

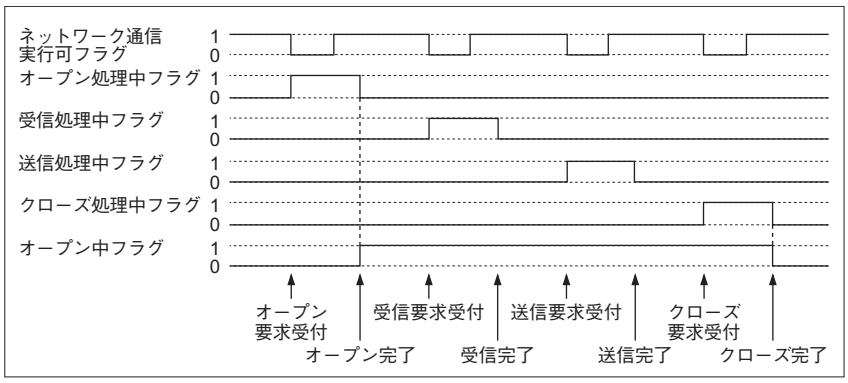
13-7-4　終了コード

SendCmd 命令の実行結果は、SendCmd 命令の出力変数（Done, Busy, Error, ErrorID, ErrorIDEx）に格納されます。詳細は『NJ/NX シリーズ コマンドリファレンスマニュアル 基本編 (SBA-468)』を参照してください。

SendCmd 命令が正常終了すると、EtherNet/IP ユニットはソケットサービス要求を処理し、処理結果が結果格納領域に「終了コード」として格納されます。詳細は「13-7-2 ソケットサービス要求」を参照してください。

13-7-5　通信時のタイミングチャート

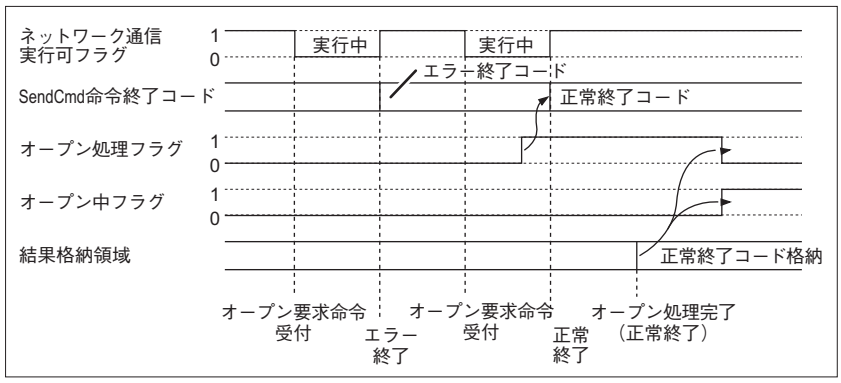
ソケットステータスの各フラグとネットワーク通信実行可フラグは、次のようなタイミングで変化します。



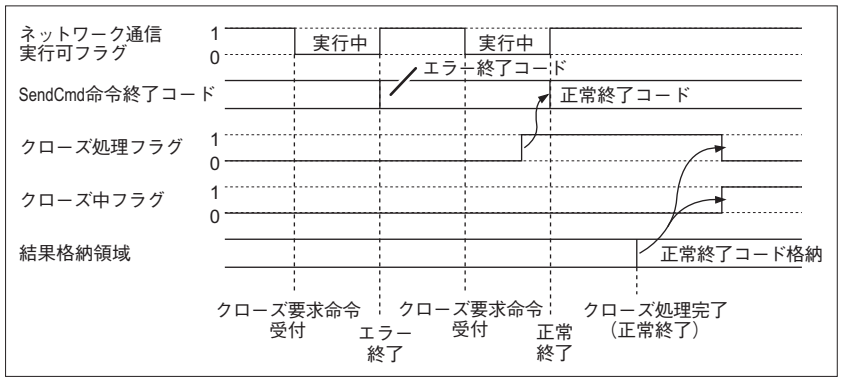
13-7-6　ソケットサービスのタイミングチャート

ソケットサービスのオープン／送信／受信／クローズの各要求コマンドは、次のようなタイミングで変化します。

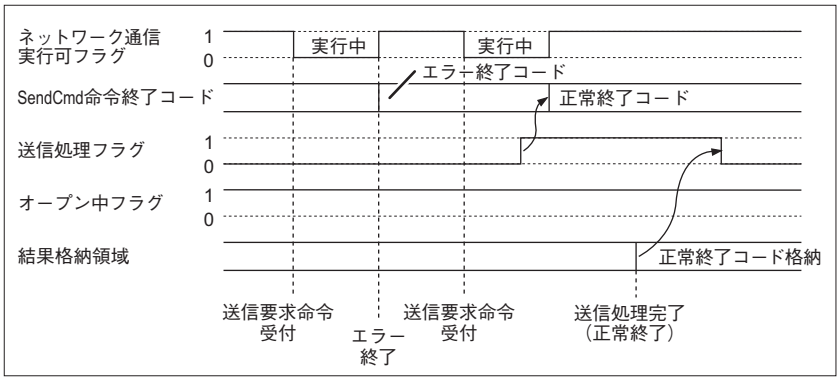
● オープン要求コマンド



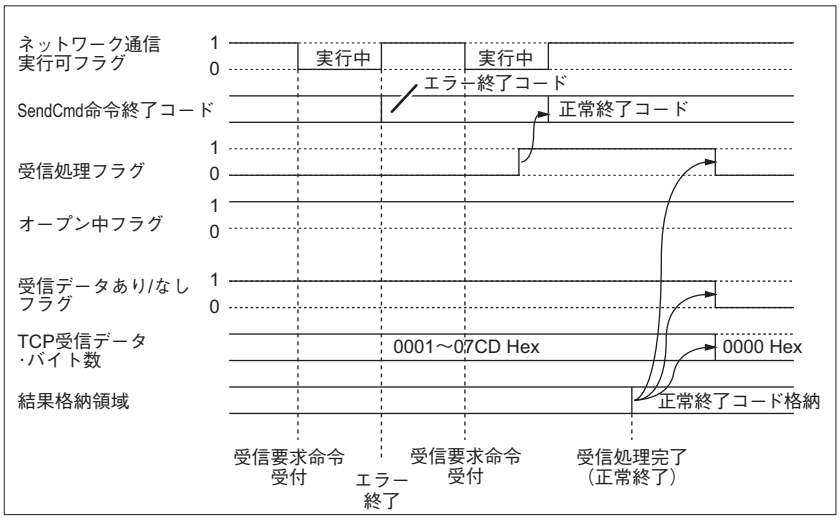
● クローズ要求コマンド



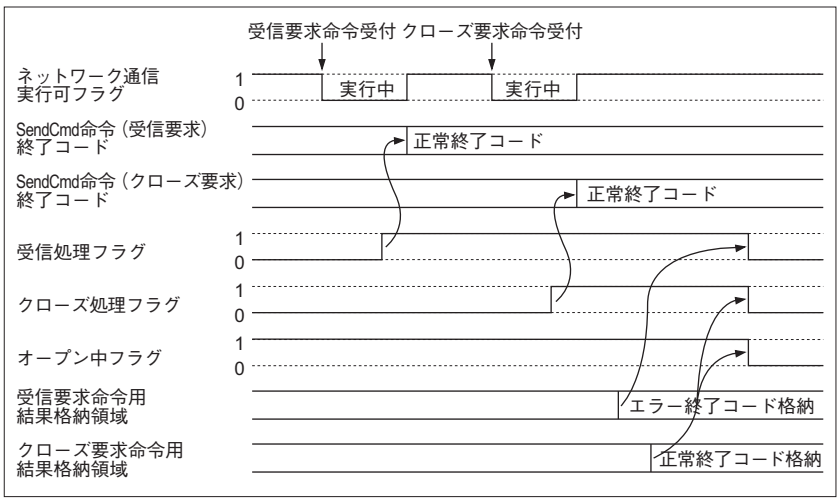
● 送信要求コマンド



● 受信要求コマンド



● 受信要求コマンド実行中にクローズ要求コマンドを実行した場合



参考

- 送信要求命令実行中にクローズ要求命令を実行した場合も、上図と同じタイミングになります。
- オープン要求命令実行中にクローズ要求命令を実行した場合は、オープン中フラグを除いて上図と同じタイミングになります。

13-7-7　サンプルプログラム

SendCmd 命令作によるソケットサービスのサンプルプログラムについては、「A-9 ソケットサービス機能のサンプルプログラム」を参照してください。

13-8 ソケットサービス使用上の留意事項

13-8-1 UDP/TCP ソケットサービス共通の留意事項

- SendCmd 命令のコントロールデータで指定するデータで、レスポンス監視時間が短く Ethernet ユニットの負荷が高いときには、レスポンスの終了コードがタイムアウトになっても結果が格納されることがあります。
  このような場合は、SendCmd 命令のコントロールデータで指定するレスポンス監視時間を長くしてください。
- ソケットステータスエリア（割付リレーエリア）は、コントローラ本体の動作モードを変更（例：プログラム→運転）すると 0 クリアされます。
  0 クリアされても、EtherNet/IP ユニットのソケットは、モード変更以前の状態にあります。
  これを回避する方法として、コントローラ本体を「IOM 保持」に設定する方法があります。設定方法はコントローラ本体のマニュアルを参照してください。
- ソケットステータス中の結果格納エリアの異常ビットが TRUE のときは、指定した格納エリアがコントローラ本体内にないことを示します。ユーザプログラムを見直してください。
- EtherNet/IP ユニットの諸機能を複数同時使用した場合や、ユーザプログラムの内容によっては、通信処理が遅くなることがあります。
- ネットワーク回線の通信負荷が高い場合、通信効率が低下することがあります。
- クローズ要求コマンドによるクローズ処理では、そのソケットの送受信バッファ中のデータはすべてフラッシュ（破棄）されます。
  たとえば、クローズ処理の直前に発行した送信要求コマンドの送信データが、送信できない場合があります。
- ソケットのオープンが完了すると、EtherNet/IP ユニットは、TCP1 ソケット当たり 4,096 バイト、UDP1 ソケット当たり 9,016 バイトの受信バッファを用意して、いつでも受信可能な状態となります。
  受信バッファがフルになると、そのソケットが受信したデータは破棄されてしまいますので、ユーザアプリケーションでは常に受信要求コマンドを発行して、この内部バッファがフルにならないようにしてください。

13-8-2 UDP ソケットサービスのみの留意事項

- UDP ソケットでは、送信先 IP アドレスにブロードキャストアドレスを設定することで、ネットワーク中の全ノードに対し一斉同報が行えます。
  ただしこの場合の送信データの最大長は、1,472 バイトになります。
  複数にフラグメントされるようなデータ長（UDP では 1,473 バイト以上）の送信はできません。
- UDP ソケットは、通信の信頼性を確保する送信データの送達確認などの制御を行いません。UDP ソケットを用いた通信で信頼性を高めるには、送達確認や再送処理をユーザプログラムで行ってください。

13-8-3 TCP ソケットサービスのみの留意事項

- 通信中に、相手ノードの TCP ソケットが一方的にクローズ（コネクションの切断）した場合、自ノードのソケットもクローズする必要があります。
  コネクションの切断は、TCP の送受信処理の結果格納により確認できます。
  相手ノード TCP ソケットのクローズを検知した場合は直ちに自ノードのソケットもクローズしてください。

TCP受信処理結果格納→　　終了コードが004B（相手ノードとの通信エラーが発生した）。
TCP送信処理結果格納→　　終了コードが0020（送信中、相手先ソケットとのコネクションが切断された）。

- 一方的に相手ノードの TCP ソケットがクローズした場合、送信すべきデータがまだ自ノードのバッファに残っていることがあります。
  残されたデータは、自ノードの TCP クローズ処理で廃棄されます。
  これを回避するには、送信側からクローズを許可するような送信データを送り、相手側で確認してからクローズするなど、アプリケーションによる工夫が必要です。
- コネクション開設中の TCP ソケットでは、先にクローズ処理を行った方のポートは、相手側ソケットのクローズ処理を行ってから 60 秒間再オープンすることができません。
  ただし、ACTIVE オープンで使用するポートを 0（ユニットによる自動割付）とし、ACTIVE オープンしたソケットからクローズした場合はあてはまりません。
- コネクションは、PASSIVE オープンしているソケットに対して、他方のソケットから ACTIVE オープンを行うことで開設されます。
  ACTIVE オープンを実行しているソケットに対して、他方のソケットから ACTIVE オープンを行っても、コネクションは開設されません。
  同じく、PASSIVE オープンを実行しているソケットに対し、他方のソケットから PASSIVE オープンを行ってもコネクションは開設されません。
  また、EtherNet/IP ユニットで PASSIVE オープンしている TCP ソケット 1 つに対し、他ノードからの複数の ACTIVE オープンによる複数コネクションの開設はできません。
- EtherNet/IP ユニットの TCP ソケットでは、ソケットオプションにより、キープアライブ（keep-alive）機能が使用できます。
  キープアライブ機能とは、コネクションが開設している通信ラインで、送受信が一定期間行われなかった場合に、コネクションが正常に確立されているかの確認を行う機能です。
  EtherNet/IP ユニット側で keep-alive 指定を行わなくとも、他ノードからの確認についてはレスポンスします。

13-8-4　デバイス変数操作に関する留意事項

- 特定ビット操作では、送信処理と受信処理を同時に実行することはできません。
  ソケットサービスパラメータエリアがソケット単位で共通のため、複数の設定を同時に行えないためです。
- たとえば、受信処理中に送信要求スイッチを TRUE にした場合、送信処理の終了コードは 110C Hex（他の処理中に要求スイッチを TRUE にした）となります。
  その後、受信処理の終了コードが上書きされます。
- 同時に複数の要求スイッチを TRUE にした場合も、終了コードは 110C Hex となり、すべての要求処理はエラー終了となります。
- ただし、クローズ処理中については、オープン処理中／送信処理中／受信処理中でも実行できます（緊急時にクローズ処理を実行可能とするためです。また、クローズ処理のパラメータは「ソケット No.」だけのため、他の処理の設定が残ったままでも実行できます）。

13-8-5　ソケットサービスの送信／受信に要する時間（ネットワーク上での遅延を無視した場合）

ソケットサービス機能の送信／受信に要する時間（伝送遅延時間）は、自ノードと相手ノードの通信処理時間の総和として求めます。

$$\text{伝送遅延時間} = \left(\begin{array}{c}\text{相手ノード}\\ \text{送信処理時間}\end{array}\right) + \left(\begin{array}{c}\text{自ノード}\\ \text{受信処理時間}\end{array}\right) + \left(\begin{array}{c}\text{自ノード}\\ \text{送信処理時間}\end{array}\right) + \left(\begin{array}{c}\text{相手ノード}\\ \text{受信処理時間}\end{array}\right)$$

EtherNet/IP ユニットにおける受信処理時間と送信処理時間は、下記の計算式より計算してください。ただし、下記の計算式で求めた遅延時間は、1 個のソケットサービスのみを使用した場合の目安値になります。複数のサービスを使用するなど、遅延時間は使用条件によりさらに長くなることがあります。なお、処理時間に対してネットワーク上での伝送遅延時間は、無視できる程度の時間として省略しています。

デバイス変数操作による UDP ソケットサービスの場合

- ソケット高速化オプション：有効　かつ　タスク周期≦ 1ms
  送信処理時間＝受信処理時間
  　＝タスク周期 ×5 ＋送受信バイト数 ×0.0002 + 1.45 (ms)
- ソケット高速化オプション：有効　かつ　2ms ≦タスク周期
  送信処理時間＝受信処理時間
  　＝タスク周期 ×6 (ms)
- ソケット高速化オプション：無効
  送信処理時間＝受信処理時間
  　＝タスク周期 ×7 (ms)

デバイス変数操作による TCP ソケットサービスの場合

- ソケット高速化オプション：有効　かつ　タスク周期≦ 1ms
  送信処理時間＝受信処理時間
  　＝タスク周期 ×5 ＋送受信バイト数 ×0.0002 + 1.45 (ms)
- ソケット高速化オプション：有効　かつ　2ms ≦タスク周期
  送信処理時間＝受信処理時間
  　＝タスク周期 ×6 (ms)
- ソケット高速化オプション：無効
  送信処理時間＝受信処理時間
  　＝タスク周期 ×7 (ms)

SendCmd 命令による UDP ソケットサービスの場合

- タスク周期< 20ms
  送信処理時間＝受信処理時間
  　＝タスク周期 ×6 + 70.0 (ms)
- 20ms ≦タスク周期
  送信処理時間＝受信処理時間
  　＝タスク周期 ×6 (ms)

SendCmd 命令による TCP ソケットサービスの場合

- タスク周期< 20ms
  送信処理時間＝受信処理時間
  　＝タスク周期 ×6 + 70.0 (ms)
- 20ms ≦タスク周期
  送信処理時間＝受信処理時間
  　＝タスク周期 ×6 (ms)

参考

- 上記の計算式で求めた値は、EtherNet/IP ユニット内で 1 つのソケットのみを使用した場合の伝送遅延時間の目安です。
  またユーザプログラム自体の実行時間は含まれていません。
- 相手ノード分の通信時間は、使用する機器により異なります。
  相手ノードが EtherNet/IP ユニット以外の場合は、使用される機器の説明書などにより計算してください。
- 伝送時間は、実際の使用環境によって計算値より長くなることがあります。
  長くなる理由としては、ネットワークの負荷率（回線の混み具合）、各ノードのウィンドウサイズ、また、EtherNet/IP ユニット自身の負荷（同時に複数のソケットを使用、またはソケットサービス機能を使用しているとき）、システムの構成などがあります。
- 上記値は、コントローラ設定での「周辺サービス時間一律設定」がデフォルト (4%) の場合の目安です。
- 「周辺サービス時間一律設定」の値を大きくすることにより、ソケットサービスの最大伝送遅延時間を短くすることができます。
- 送信要求をラダープログラムのタイマなどで定期的に処理する場合や、相手ノードからの連続したデータを受信要求処理する場合、EtherNet/IP ユニットの送信／受信処理能力よりも早くは処理できません。特に受信側はデータバッファが枯渇する場合もあります。このような場合は、処理能力（性能）よりも 1.5 倍程度遅くなる実負荷になるように送信タイミング、もしくは受信頻度（相手ノードからの送信タイミング）を調整してください。

例：2 台のコントローラ間でデバイス変数操作（ソケット高速化オプション：有効）による TCP ソケットサービスを使用し、512 バイトの送受信処理を双方向で行った場合、下記の条件では、最大伝送遅延時間の目安は下表のように計算します。

自ノード　タスク周期　　＝ 1ms
相手ノードタスク周期　　＝ 4ms

項目	計算式
自ノード受信処理時間	1×5+512×0.0002+1.45=6.552ms≒6.6
自ノード送信処理時間	1×5+512×0.0002+1.45=6.552ms≒6.6
相手ノード送信処理時間	4×6=24.0ms
相手ノード受信処理時間	4×6=24.0ms
伝送遅延時間	6.6+6.6+24.0+24.0=61.2ms

14

通信性能と通信負荷

14-1 通信のしくみ 14-2
14-1-1 タグデータリンクの通信のしかた 14-2
14-1-2 コネクション数の計算方法 14-5
14-1-3 パケットインターバル（RPI）の精度 14-6

14-2 通信負荷の調整 14-7
14-2-1 タグデータリンクのデバイス帯域使用状況の確認 14-8
14-2-2 タグデータリンクの帯域使用率と RPI の関係 14-10
14-2-3 デバイス帯域使用状況の調整方法 14-11
14-2-4 RPI の変更方法 14-12
14-2-5 パケットインターバル（RPI）の設定例 14-17

14-3 タグデータリンクによる入出力応答時間 14-22
14-3-1 データ転送のタイミング 14-22
14-3-2 EtherNet/IP ユニットのデータ処理時間 14-23
14-3-3 タグデータリンクによる、タスク周期への影響 14-26
14-3-4 タグデータリンク入出力最大応答時間 14-26

14-4 メッセージサービス機能伝送遅延時間 14-28

14-1 通信のしくみ

14-1-1 タグデータリンクの通信のしかた

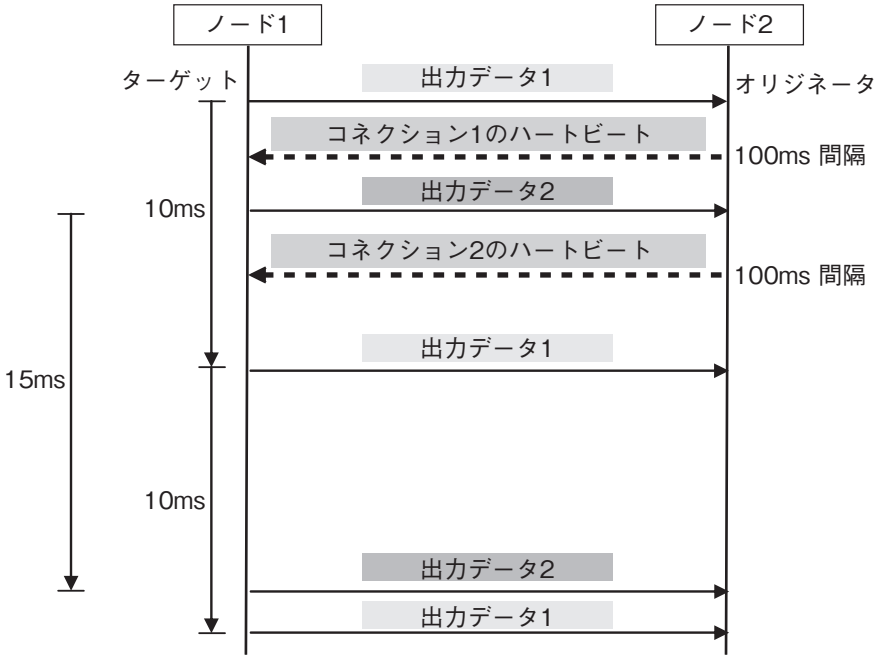
パケットインターバル（RPI）の設定

EtherNet/IP ユニットのタグデータリンクでは、データ送信周期を、コネクションごとに、パケットインターバル（RPI）として設定しています。
ターゲットデバイスは、ノード数に依存せず、指定したこのパケットインターバル（RPI）ごとに、データ送信（出力タグの送信）を行います。
また、オリジネータからターゲット宛にコネクションごとにハートビートフレームを送信しています。ハートビートはターゲットがオリジネータとのコネクションに異常が発生していないことを確認するために使用しています。ハートビートフレームの送信周期はパケットインターバル（RPI）の設定により決定します。

【ハートビートフレームの送信周期】

- パケットインターバル ＜ 100ms の場合
  →ハートビートフレーム送信周期は 100ms
- パケットインターバル ≧ 100ms の場合
  →ハートビートフレーム送信周期は RPI と同一

例）
ノード 2（オリジネータ）は、ノード 1（ターゲット）に対して、2 つのタグデータリンクコネクションを設定。
出力データ 1 を、パケットインターバル（RPI）10ms に設定。
出力データ 2 を、パケットインターバル（RPI）15ms に設定。
この場合、以下のようにノード 1 からノード 2 へ、出力データ 1 および 2 が、各々 10ms および 15ms の定周期で送信されます。
また、ノード 2（オリジネータ）からノード 1（ターゲット）に対して、コネクション 1 のハートビート 100ms 間隔、コネクション 2 のハートビート 100ms 間隔で送信されます。



パケットインターバル（RPI）と、使用通信帯域幅の関係

タグデータリンクにて、1 秒間に送受信するパケット数に逓倍化 (以下、N) したものを、「使用通信帯域幅」と呼びます。N はデータサイズに応じた重み付けの数値です。
使用通信帯域幅は、コネクションごとに、RPI、ハートビート、N から、以下のように算出されます。

コネクションで使用する使用通信帯域幅
＝（1000 ÷ RPI[ms] × N）+（1000 ÷ ハートビート送信周期 [ms] ）

N = タグデータリンク許容通信帯域幅
　　÷（ タグデータリンク許容通信帯域幅 + 係数 × 1 コネクションあたりのデータサイズ ）

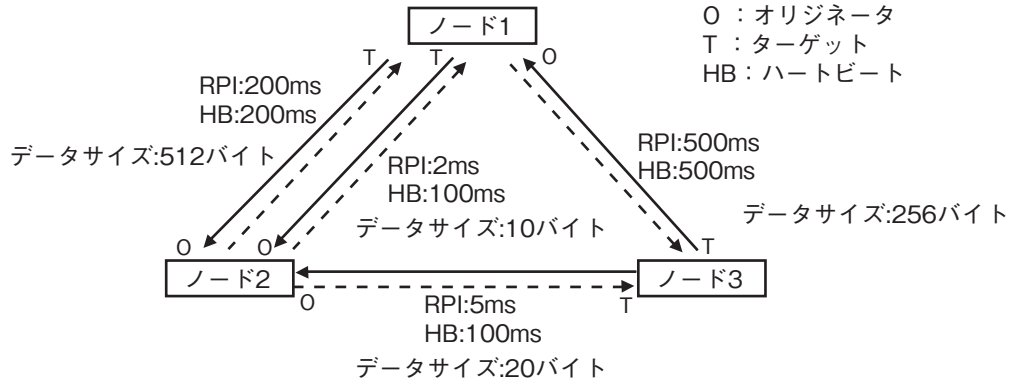
ユニットバージョン	タグデータリンク許容通信帯域幅	係数	N
V2.1 以前	6000	0	1
V3.0	12000	-4.155	1 ～ 2

また、各 EtherNet/IP ユニット（以下ユニット）にて、使用通信帯域幅の総数は、以下のように算出されます。
ユニットの総使用通信帯域幅
＝ オリジネータコネクションの総使用通信帯域幅＋ターゲットコネクションの総使用通信帯域幅
（注） ターゲットコネクションの総使用通信帯域幅 には、ターゲットとして設定されたコネクションも加算する必要があることに注意してください。

タグデータリンクにて、本ユニットの総使用通信帯域幅が上限値を超えない範囲で、コネクションの設定を行ってください。

ユニットバージョン	総使用通信帯域幅の上限値
V2.1 以前	6000
V3.0	12000

例）
ノード 1 は、オリジネータ、ターゲット両コネクションをもつ：送信用 RPI 200ms（データサイズ 512 バイト）と 2ms（データサイズ 10 バイト）、受信用 RPI 500ms（データサイズ 256 バイト）。
ノード 2 は、オリジネータコネクションだけをもつ：受信用 RPI 200ms（データサイズ 512 バイト）と 2ms（データサイズ 10 バイト）と 5ms（データサイズ 20 バイト）。
ノード 3 は、ターゲットコネクションだけをもつ：送信用 RPI 5ms（データサイズ 20 バイト）と 500ms（データサイズ 256 バイト）。



各ノードでの総使用通信帯域幅 は、以下のように算出されます。

● V2.1 以前の計算方法

・ノード 1 のユニットの総使用通信帯域幅
　＝ 1000 / 200 ms ×1 ＋ 1000 / 2 ms ×1 ＋ 1000 / 500 ms ×1 （データ用）
　　＋ 1000 / 200 ms ＋ 1000 / 100 ms ＋ 1000 / 500 ms （ハートビート用）
　＝ 524
・ノード 2 のユニットの総使用通信帯域幅
　＝ 1000 / 200ms ×1 ＋ 1000 / 2 ms ×1 ＋ 1000 / 5 ms ×1 （データ用）
　　＋ 1000 / 200ms ＋ 1000 / 100 ms ＋ 1000 / 100 ms （ハートビート用）
　＝ 730
・ノード 3 のユニットの総使用通信帯域幅
　＝ 1000 / 5 ms ×1 ＋ 1000 / 500 ms ×1 （データ用）
　　＋ 1000 / 100 ms ＋ 1000 / 500 ms （ハートビート用）
　＝ 214

いずれも、総使用通信帯域幅の上限値（6000）以下のため、送受信可能です。

● V3.0 の計算方法

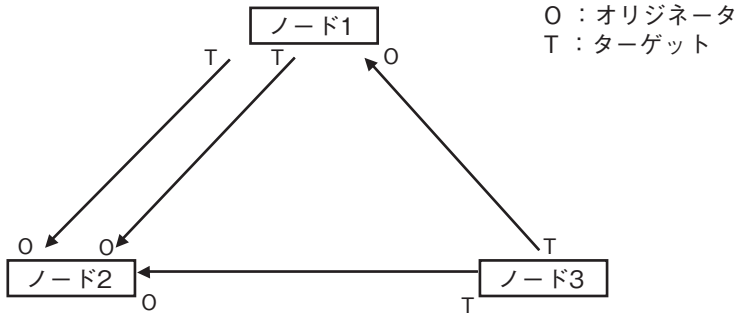
データサイズ [バイト]	N の値
10	12000/（12000-4.155×10）=1.003
20	12000/（12000-4.155×20）=1.007
256	12000/（12000-4.155×256）=1.097
512	12000/（12000-4.155×512）=1.215

・ノード 1 のユニットの総使用通信帯域幅
　＝ 1000 / 200 ms ×1.215 ＋ 1000 / 2 ms ×1.003 ＋ 1000 / 500 ms ×1.097（データ用）
　　＋ 1000 / 200 ms ＋ 1000 / 100 ms ＋ 1000 / 500 ms （ハートビート用）
　＝ 527
・ノード 2 のユニットの総使用通信帯域幅
　＝ 1000 / 200ms ×1.215 ＋ 1000 / 2 ms ×1.003 ＋ 1000 / 5 ms ×1.007（データ用）
　　＋ 1000 / 200ms ＋ 1000 / 100 ms ＋ 1000 / 100 ms （ハートビート用）
　＝ 734
・ノード 3 のユニットの総使用通信帯域幅
　＝ 1000 / 5 ms ×1.007 ＋ 1000 / 500 ms ×1.097（データ用）
　　＋ 1000 / 100 ms ＋ 1000 / 500 ms （ハートビート用）
　＝ 216

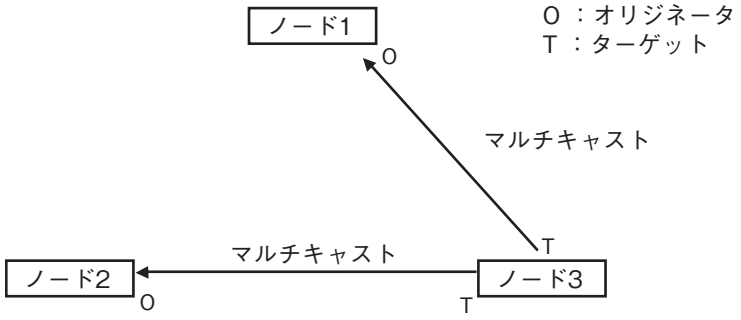
いずれも、総使用通信帯域幅の上限値（12000）以下のため、送受信可能です。

14-1-2　コネクション数の計算方法

EtherNet/IP ユニットのコネクション数は最大で 256 個です。
オリジネータとして自身が開設するコネクションと、ターゲットとしてオリジネータから開設されるコネクションの両方を合わせた数が 256 個以内で設定する必要があります。
例）
ノード 1 はノード 2 と 2 本（ターゲット）、ノード 3 と 1 本（オリジネータ）のコネクションを開設しています。よって、合計は 3 本となります。
ノード 2 はノード 1 と 2 本（オリジネータ）、ノード 3 と 1 本（オリジネータ）のコネクションを開設しています。よって、合計は 3 本となります。
ノード 3 はノード 1 と 1 本（ターゲット）、ノード 2 と 1 本（ターゲット）のコネクションを開設しています。よって、合計は 2 本となります。
いずれも、EtherNet/IP ユニットの最大コネクション数 256 個以内のため、コネクションの開設が可能です。



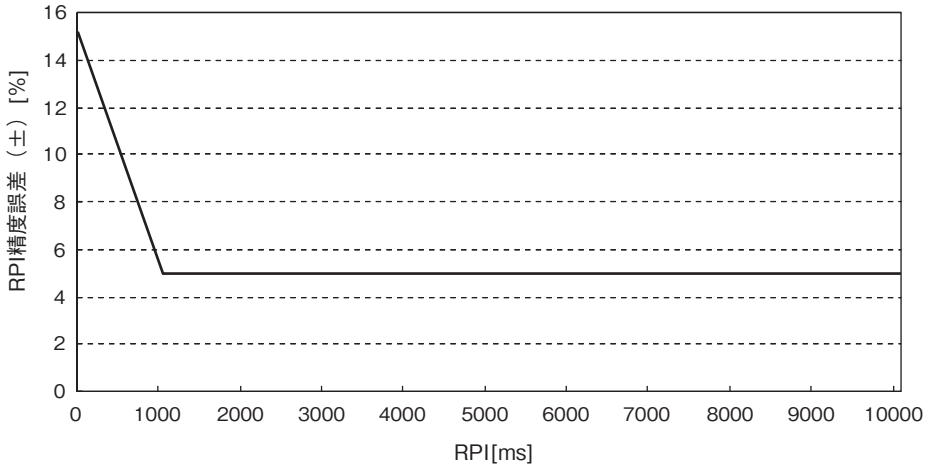
また、マルチキャストに設定した場合、送信するパケットは 1 つになりますが、コネクション数は消費されます。
例）
ノード 3 はノード 1 とノード 2 に対してマルチキャストパケットを 1 つ送信しています。このときのノード 3 はノード 1 と 1 本 (ターゲット)、ノード 2 と 1 本 (ターゲット)、計 2 本のコネクションを開設しています。
マルチキャストに設定してもユニキャストとコネクション消費数は変わらないので注意が必要です。



14-1-3 パケットインターバル（RPI）の精度

パケットインターバル（RPI）に基づきパケットを送信しようとした際、EtherNet/IP ユニット内部で送信処理遅延が発生します。
この遅延は、以下に示す RPI 精度誤差の範囲で変化するため、各 RPI における最大値が送信処理遅延時間となります。

パケットインターバル（RPI）	RPI 精度誤差（±）[%]
0.5ms ～ 1000ms	15 －（RPI[ms] ／ 100）
1000ms ～ 10000ms	RPI の 5%



14-2 通信負荷の調整

スイッチングハブを使用する Ethernet ネットワークでは、全ノードでネットワーク帯域幅を共有するのではなく、スイッチングハブを介した個々のノード間で、独立した伝送路が確立されます。
ノード間の通信では、スイッチングハブ内で専用の送受信バッファが確保され、他の伝送路とは非同期に全二重で（送信と受信が同時に）通信されます。さらに、他の伝送路での通信負荷の影響を受けないためコリジョン（パケットの衝突）が発生せず、高速で安定した通信が行われます。
次に示すスイッチングハブの機能は、タグデータリンク性能を決定する要素となります。

項目	説明
バッファ容量	スイッチングハブにパケットが集中した場合にバッファリングできるサイズ
マルチキャストフィルタ機能	マルチキャストパケットを特定のノードにのみ転送する機能
QoS 機能	パケット転送の優先制御を行う機能

EtherNet/IP ユニットでは、下記に示す仕様の範囲内でタグデータリンクの設定ができます。

項目	内容	値
ネットワーク帯域幅	Ethernet の物理的な伝送速度	100Mbps または 10Mbps
タグデータリンク許容通信帯域幅	1 秒間に処理可能なタグデータリンクの最大パケット数（pps：packet per second）	6000 ～ 12000pps
コネクションリソース	開設可能なコネクション数	最大 256
パケットインターバル（RPI：Requested Packet Interval）	タグデータの更新周期	0.5 ～ 10000ms （0.5ms 単位）

使用するスイッチングハブの性能を上回るタグデータリンクが設定されている場合は、パケットインターバル（RPI）の値を調整して（大きくして）再設定してください。
特に、マルチキャストフィルタ機能を持たないスイッチングハブを使用する場合、コネクション設定を行っていないノードに対してもマルチキャストパケットが送信されることを考慮して設定する必要があります。

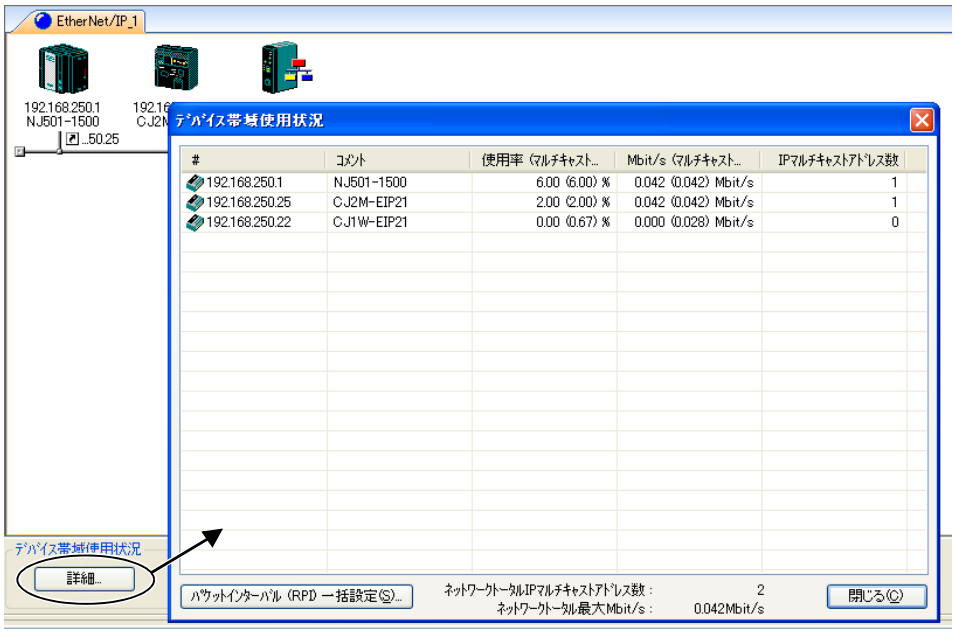
参考

Network Configurator によるコネクション設定にてコネクションタイプとして「Multi-cast connection」を選択するとマルチキャストパケットが使われます。コネクションタイプとして、すべて「Point to Point connection」を選択するとマルチキャストパケットは使われません。

また、必要とするタグデータリンク性能がこの仕様範囲内で実現できない場合は、スイッチングハブの再選定、またはネットワークを分割するなど、全体のネットワーク構成を再検討してください。
ここでは、設計したネットワークでタグデータリンクが使用するデバイス帯域使用状況の確認方法と、値を再設定する方法を説明します。

14-2-1　タグデータリンクのデバイス帯域使用状況の確認

Network Configurator では、ネットワーク構成上で設定したコネクションに応じて、タグデータリンクが実際に使用することになる帯域使用状況が、個々の EtherNet/IP ユニットごとに表示されます。タグデータリンクのデバイス帯域使用状況は、ネットワーク構成ウィンドウ下部の「デバイス帯域使用状況」内の［詳細］ボタンをクリックすることにより、確認できます。



項目	内容
#	該当デバイスの IP アドレス
コメント	該当デバイスに設定された説明。アイコン直下で表示されています。デフォルトで形式が表示されます。
使用率（マルチキャストフィルタなし）	該当デバイスのタグデータリンク許容通信帯域幅に対する使用率。「使用通信帯域幅」÷「タグデータリンク許容通信帯域幅」 （ ）外：マルチキャストフィルタありの場合を示します。 （ ）内：マルチキャストフィルタなしの場合を示します。
Mbit/s（マルチキャストフィルタなし）	100Mbit/s のネットワーク帯域のうち、該当デバイスに送受信されるタグデータリンクの帯域幅。 （ ）外：マルチキャストフィルタありの場合を示します。 （ ）内：マルチキャストフィルタなしの場合を示します。
IP マルチキャストアドレス数	該当デバイス上で、実際に送受信で使用される IP マルチキャストアドレスの数を示します。
ネットワークトータル IP マルチキャストアドレス数	ネットワーク全体で使用される IP マルチキャストアドレスの数を示します。スイッチにおけるマルチキャストフィルタ数の見積りに使用します。
ネットワークトータル最大 Mbit/s	ネットワーク全体で送受信されるタグデータリンクの帯域幅。100Mbit/s のネットワーク帯域を超えると、タグデータリンクが正常に動作しなくなります。

● タグデータリンク許容通信帯域幅に対する使用率、ネットワーク帯域の使用量の確認

EtherNet/IP ユニットごとの、タグデータリンク許容通信帯域幅に対する使用率「使用率（マルチキャストフィルタなし）」、およびネットワーク帯域の使用量「Mbit/s（マルチキャストフィルタなし）」が表示されます。

マルチキャストフィルタ機能を持たないスイッチングハブを使用した場合の使用率 / 使用量はそれぞれ（ ）内に表示されます。この場合、コネクション設定を行っていないノードに対してもマルチキャストパケットが送信されてくるため、これらのパケットも含めた使用率、使用量が表示されます。

これらの値は、「14-2-4 RPI の変更方法」に従い調整が可能です。

● ネットワークトータルマルチキャスト IP アドレス数の確認

マルチキャストフィルタ機能を持つスイッチングハブを使用する場合、マルチキャストフィルタ数が、使用するネットワークに対し十分である必要があります。Network Configurator では、コネクション設定の結果からネットワーク全体で使用するマルチキャスト IP アドレス数を表示します。

このネットワーク全体で使用するマルチキャスト IP アドレス数が、スイッチングハブがサポートするマルチキャストフィルタ数を超えないことを確認してください。超える場合は、スイッチングハブを十分なマルチキャストフィルタ数を持つものに変更するか、タグデータリンク許容通信帯域幅に対する使用率、ネットワーク帯域の使用量を、マルチキャストフィルタ機能を持たないスイッチングハブを使用した場合の値（( ) 内の値）で調整してください。これらの値は、「14-2-4 RPI の変更方法」に従い調整が可能です。

● ネットワークトータル最大 Mbit/s の確認

Network Configurator では、ネットワークトータルで使用する最大帯域使用量を表示します。この値は、スイッチングハブ間をカスケード接続した際に、そのカスケード接続した伝送路に流れる可能性のある最大帯域使用量を表します。この値が、実際のネットワークの各カスケード接続部の帯域（例えば 100Mbit/s）を超える場合、ネットワークの配線の仕方によっては一部の伝送路で帯域オーバーが発生しタグデータリンクが正常に動作しない可能性があります。

この場合、各伝送路で使用される帯域使用量を算出し、すべてのカスケード接続部の帯域を超えていないことを確認するか、ネットワークトータル最大 Mbit/s がすべてのカスケード接続部の帯域を越えないように調整してください。この値は、「14-2-4 RPI の変更方法」に従い調整が可能です。

14-2-2　タグデータリンクの帯域使用率と RPI の関係

タグデータリンク許容通信帯域幅に対する使用率「使用率（マルチキャストフィルタなし）」は、パケットインターバル（RPI）設定により調整することができます。
パケットインターバル（RPI）を短く設定すればするほど、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」が上がります。
逆に、パケットインターバル（RPI）を長く設定すればするほど、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」が下がります。

パケットインターバル（RPI）は、以下のいずれかの単位で設定できます。

- 全コネクション同一
- 特定デバイスのコネクション
- 特定コネクション

全コネクションに対するパケットインターバル（RPI）を同一に設定する場合は、設定パケットインターバル（RPI）を高速化すればするほど、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」は、基本的に、比例的に上がります。
例：全コネクションに対するパケットインターバル（RPI）設定が 50ms で、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」が 40% のとき、全コネクションに対するパケットインターバル（RPI）を 25ms に変更した場合、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」が 80% となる可能性があります。

使用上の注意

タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」が 80% を超え 100% 以下で使用した場合に、Network Configurator の操作（例：モニタなどネットワークに負荷がかかる操作）またはユーザアプリケーションによるメッセージ通信などで、一過性の負荷がかかり、その結果タイムアウトなどが発生することがあります。その場合、いずれかまたはすべてのパケットインターバル（RPI）設定を長い値に変更して、タグデータリンク許容通信帯域幅に対する使用率を下げてください。

14-2-3　デバイス帯域使用状況の調整方法

● スイッチングハブにマルチキャストフィルタ機能がないとき（スイッチングハブは100Mbit/s 対応）

- 全ノードの「Mbit/s（マルチキャストフィルタなし）」が、100Mbit/s 以下か？
  1 つでも超える場合：RPI 変更などコネクション設定の見直しを行ってください。
- 全ノードの「使用率（マルチキャストフィルタなし）」が、100% 以下か？
  1 つでも超える場合：RPI 変更などコネクション設定の見直しを行ってください。
- 「ネットワークトータル最大 Mbit/s」が、100Mbit/s 以下か？
  超える場合：ネットワーク配線（スイッチングハブをカスケード接続するなど）によっては一部の伝送路（スイッチングハブやメディアコンバータ）で帯域オーバーが発生しタグデータリンクが正常に動作しない可能性があります。各カスケード接続部の伝送路が帯域を超えていないことを確認してください。超えていた場合、ネットワーク配線の変更やスイッチングハブ間の帯域を拡大する（1Gbit/s 化など）ことにより対策してください。対策不可能な場合は、RPI 変更などコネクション設定の見直しを行いネットワークトータル最大 Mbit/s がすべてのカスケード接続部の帯域を超えないように調整してください。

● スイッチングハブにマルチキャストフィルタ機能があるとき（スイッチングハブは100Mbit/s 対応）

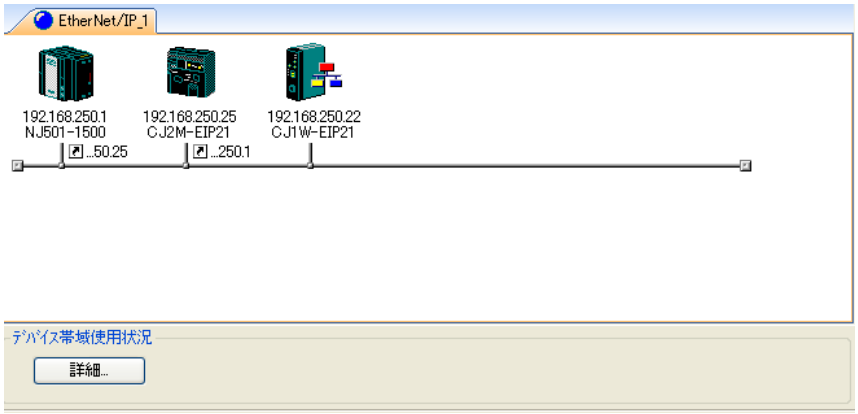
- 全ノードの「Mbit/s」が、100Mbps 以下か？
  1 つでも超える場合：RPI 変更などコネクション設定の見直しを行ってください。
- 全ノードの「使用率」が、100% 以下か？
  1 つでも超える場合：RPI 変更などコネクション設定の見直しを行ってください。
- 「ネットワークトータル最大 Mbit/s」が、100Mbit/s 以下か？
  超える場合：ネットワーク配線（スイッチングハブをカスケード接続するなど）によっては一部の伝送路（スイッチングハブやメディアコンバータ）で帯域オーバーが発生しタグデータリンクが正常に動作しない可能性があります。各カスケード接続部の伝送路が帯域を超えていないことを確認してください。超えていた場合、ネットワーク配線の変更やスイッチングハブ間の帯域を拡大する（1Gbit/s 化など）ことにより対策してください。対策不可能な場合は、RPI 変更などコネクション設定の見直しを行いネットワークトータル最大 Mbit/s がすべてのカスケード接続部の帯域を超えないように調整してください。
- 全ノードの「Mbit/s（マルチキャストフィルタなし）」が、100Mbit/s 以下か？
  または、全ノードの「使用率（マルチキャストフィルタなし）」が、100% 以下か？
  1 つでも超える場合：スイッチングハブのマルチキャストフィルタが正しく機能するか検証します。
  スイッチングハブのマルチキャストフィルタ数が［ネットワークトータル IP マルチキャストアドレス数］より少ない場合、ネットワーク配線（スイッチングハブをカスケード接続するなど）によっては一部の伝送路で帯域オーバーが発生しタグデータリンクが正常に動作しない可能性があります。ネットワーク上の各スイッチングハブで必要となるマルチキャストフィルタ数を算出し、その数がスイッチングハブのマルチキャストフィルタ数以下であることを確認してください。スイッチングハブのマルチキャストフィルタ数が不足している場合、十分なマルチキャストフィルタ数を持つスイッチを使用するか、RPI 変更などコネクション設定の見直しを行ってください。

14-2-4　RPI の変更方法

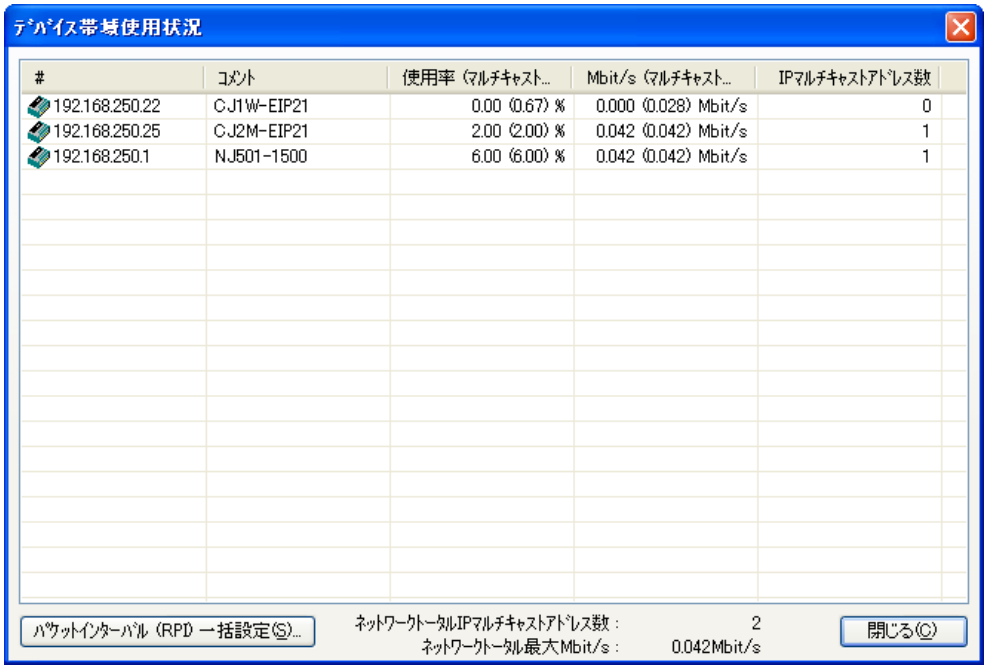
「14-2-1 タグデータリンクのデバイス帯域使用状況の確認」に従い、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」をオフラインで確認することができます。
タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」は、パケットインターバル（RPI）設定を変更することで調整が可能です。
ただし、調整の結果、必要な通信性能を確保できない場合、ネットワーク構成から再検討をしてください。

1　**Network Configurator のネットワーク構成上で、必要な設定を行います。**

2　**ネットワーク構成ウィンドウの下部にある「デバイス帯域使用状況」内の［詳細］ボタンをクリックします。**



［デバイス帯域使用状況］ダイアログボックスが表示されます。

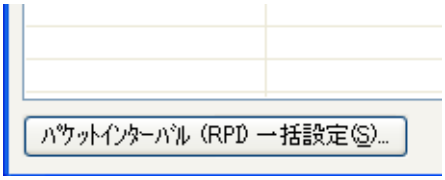


タグデータリンク許容通信帯域幅に対する使用率が「使用率（マルチキャストフィルタなし）」列に、ネットワーク帯域（伝送速度）での使用量が「Mbit/s（マルチキャストフィルタなし）」列に、表示されます。

3 **タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」を該当デバイスのパケットインターバル（RPI）設定を変更することで調整することが可能です。**
パケットインターバル（RPI）設定を変更するには、以下の 3 つの方法があります。

方法 1：すべてのコネクションを同一のパケットインターバル（RPI）設定とする場合：
すべてのコネクションのパケットインターバル（RPI）設定を一括して変更することでタグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」を調整することが可能です。

(1) ［デバイス帯域使用状況］ウィンドウの［パケットインターバル（RPI）一括設定］ボタンをクリックします。

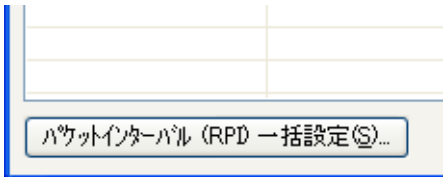


(2) ［パケットインターバル（RPI）一括設定］ウィンドウが表示されます。新たなパケットインターバル（RPI）値を入力し、［OK］ボタンをクリックしてください。



方法２：特定デバイスのパケットインターバル（RPI）設定を変更する場合：
特定デバイスのみ、すべてのコネクションのパケットインターバル（RPI）設定を一括して変更することでタグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」を調整することが可能です。
この場合、パケットインターバル（RPI）設定を変更したコネクションのターゲットデバイスでも、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」が変化することにご注意ください。

(1) ［デバイス帯域使用状況］ウィンドウの［パケットインターバル（RPI）一括設定］ボタンをクリックします。



(2) ［パケットインターバル（RPI）一括設定］ウィンドウが表示されます。［対象デバイス］フィールドにて、パケットインターバル（RPI）設定を変更しないデバイスのチェックボックスのチェックをはずしてください。

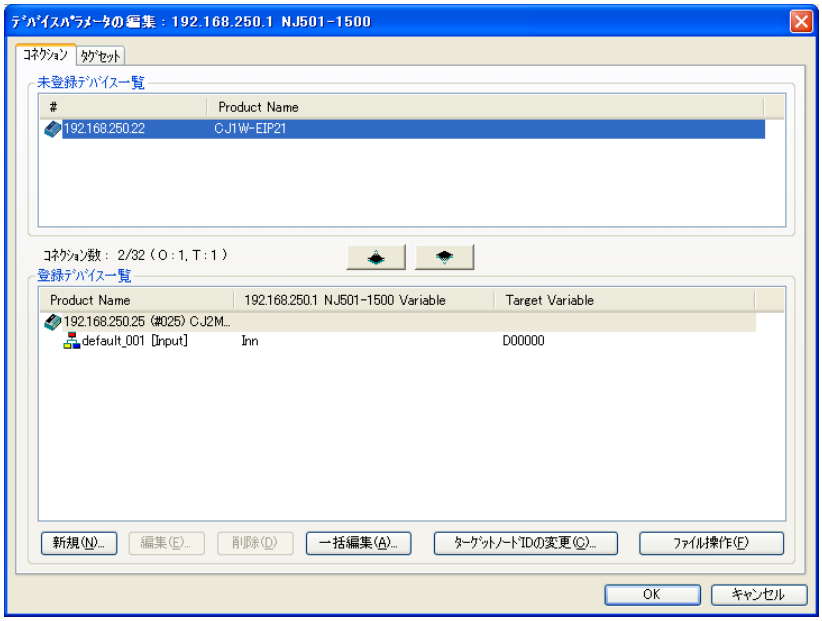


(3) 新たなパケットインターバル（RPI）値を入力し、［OK］ボタンをクリックしてください。

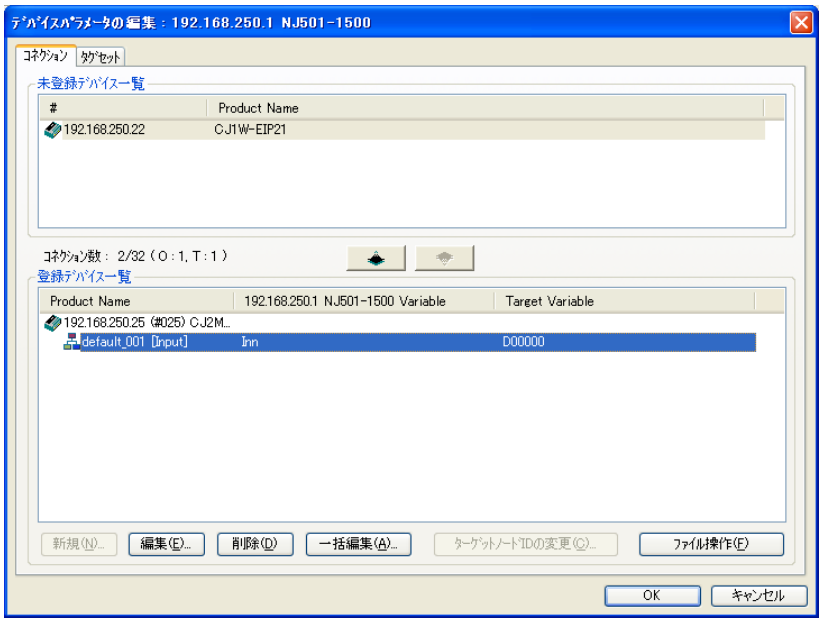
方法３：特定コネクションのパケットインターバル（RPI）設定を変更する場合：
特定コネクションのパケットインターバル（RPI）設定を個別に変更することでタグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」を調整することが可能です。
この場合、パケットインターバル（RPI）を変更したコネクションのターゲットデバイスでも、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」が変化することにご注意ください。

(1)　［デバイス帯域使用状況］ウィンドウの［閉じる］ボタンをクリックし、ウィンドウを閉じます。

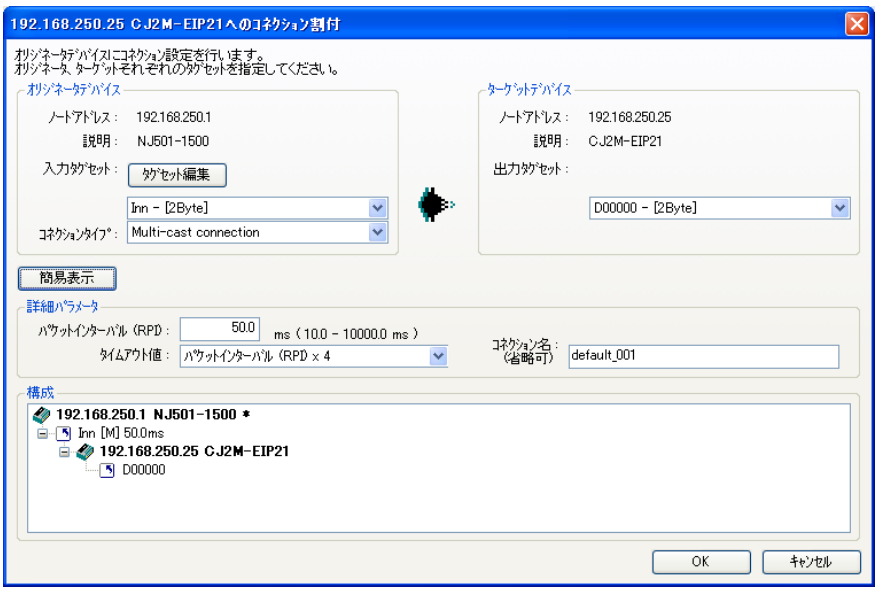
(2)　特定コネクションのオリジネータ設定を持つデバイスをダブルクリックします。［デバイスパラメータの編集］ウィンドウが表示されます。



(3)　［登録デバイスー覧］にて、パケットインターバル（RPI）を変更したいコネクションを選択し、［編集］ボタンをクリックします。



(4) 該当デバイスへのコネクション割付ウィンドウが表示されます。新たなパケットインターバル（RPI）値を入力し［OK］ボタンをクリックします。



4　上記設定を実行しても、希望する使用率に調整できないときは、再度、ネットワーク構成自体を、以下の観点で検討し直してください。（「14-2-3 デバイス帯域使用状況の調整方法」を参照）

- ノード数、コネクション数の削減
- ネットワークの分散

5　帯域使用率の再確認

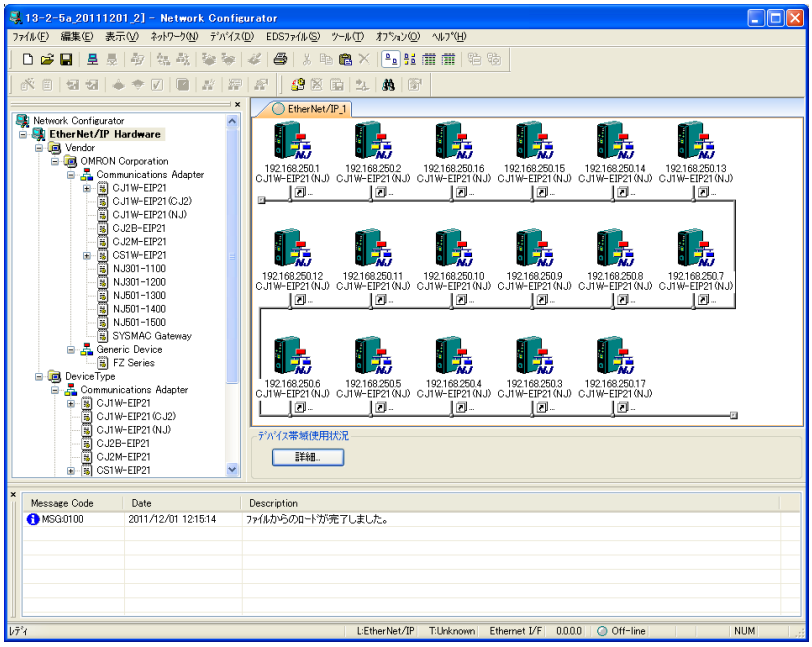
コネクション設定を変更した場合は、ネットワーク構成ウィンドウ下部の「デバイス帯域使用状況」内の［詳細］ボタンをクリックし、「14-2-1 タグデータリンクのデバイス帯域使用状況の確認」に従いチェックを行ってください。
とくに、パケットインターバル（RPI）一括設定を使用せずにコネクションごとに調整した場合、この確認が重要となります。

6　ユーザテストにより、設定した値に問題ないことを確認してください。

14-2-5　パケットインターバル（RPI）の設定例

次のネットワーク構成でパケットインターバル（RPI）の計算例を説明します。

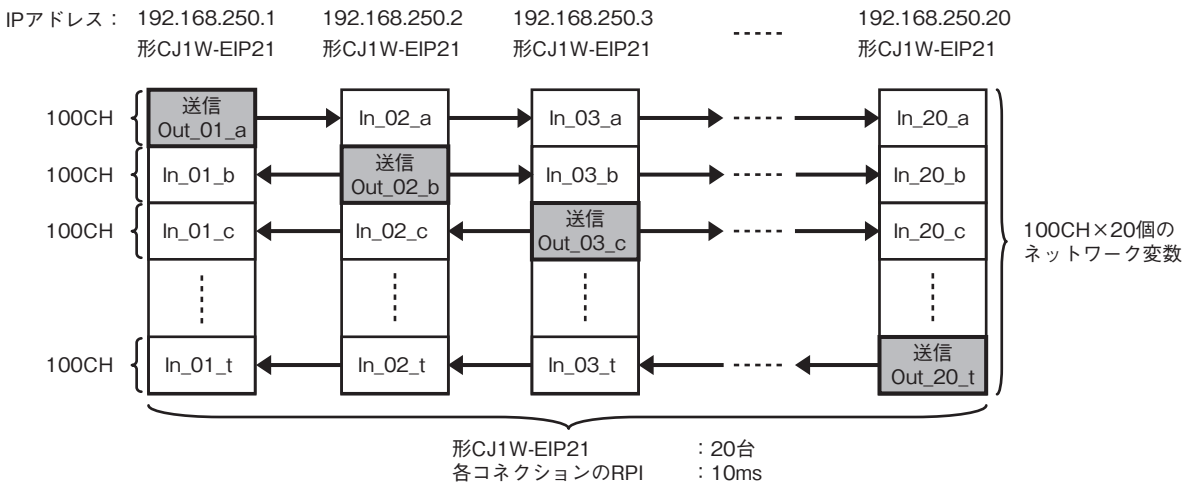
条件



以下の使用率の表記は、形 CJ1W-EIP21 のユニットバージョン V2.1 以前のものとなります。

- コネクション：

例）20 台の形 CJ1W-EIP21 がネットワークに接続されています。
各デバイスは、サイズ 100CH の送信用タグ 1 個と、サイズ 100CH の受信用タグを 19 個持ち、相互にデータ交換を行っています。
すべてのコネクションのパケットインターバル（RPI）は、10ms に設定します。
なお、各デバイスの IP アドレスは、192.168.250.1 ～ 192.168.250.20 とします。



デバイス帯域使用状況の確認

デバイス帯域使用状況の［詳細］ボタンをクリックすると、以下のように各デバイスのタグデータリンク許容通信帯域幅に対する使用率が 33.67%であることが分かります。



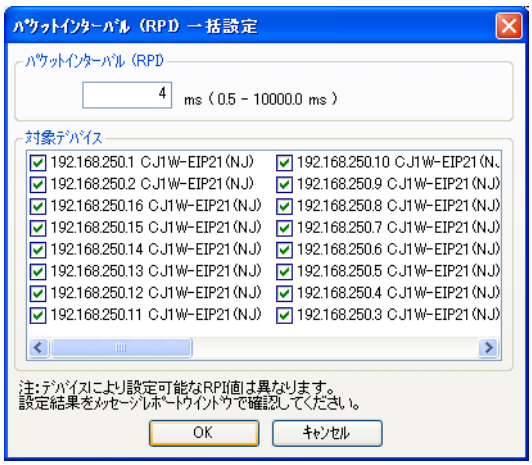
設定の変更

＜方法１＞すべてのコネクションを同一パケットインターバル（RPI）にする場合

すべてのコネクションのパケットインターバル（RPI）が 10ms の設定でタグデータリンク許容通信帯域幅に対する使用率が 33.67%でしたが、ここでは 80% 以内の許容帯域を目標に、パケットインターバル（RPI）を 4ms に変更します。

［デバイス帯域使用状況］ウィンドウの［パケットインターバル（RPI）一括設定］ボタンをクリックすると、［パケットインターバル（RPI）一括設定］ウィンドウが表示されます。

新たなパケットインターバル（RPI）値として 4ms を入力し、［OK］ボタンをクリックします。



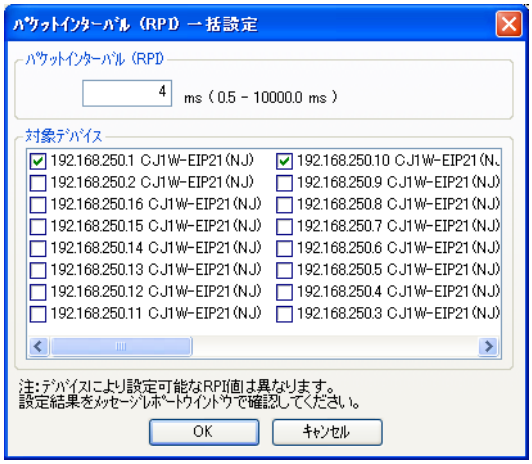
すべてのコネクションのパケットインターバル（RPI）を同一にした場合、タグデータリンク許容通信帯域幅に対する使用率が 76.17% になり、4ms が最も早い設定値であることが分かります。



＜方法 2 ＞特定デバイスのパケットインターバル（RPI）だけを変更する場合

デバイス 192.168.250.1 と 192.168.250.10 だけ、高速にタグデータリンクを行いたいとします。この場合［デバイス帯域使用状況］ウィンドウの［パケットインターバル（RPI）一括設定］ボタンをクリックし、［一括設定］ウィンドウを表示します。

［対象デバイス］にて、パケットインターバル（RPI）を変更しないデバイス（192.168.250.1 と 192.168.250.10 以外）のチェックをはずします。新たなパケットインターバル（RPI）値として 4ms を入力し、［OK］ボタンをクリックします。



デバイス 192.168.250.1 と 192.168.250.10 のタグデータリンク許容通信帯域幅に対する使用率が 77.83% になりこれらのデバイスのコネクションのパケットインターバル（RPI）が高速に設定されたことが分かります。

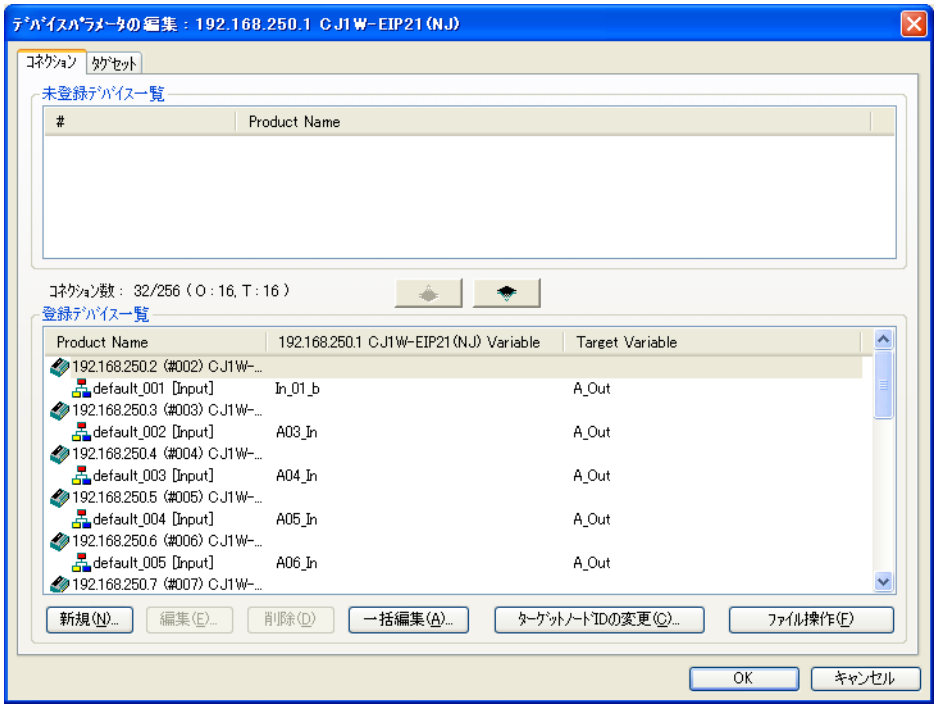
なお、この場合、デバイス 192.168.250.1 と 192.168.250.10 とコネクションを持つデバイスのタグデータリンク許容通信帯域幅に対する使用率も増えている（33.64%→ 37.83%）ことに注意してください。



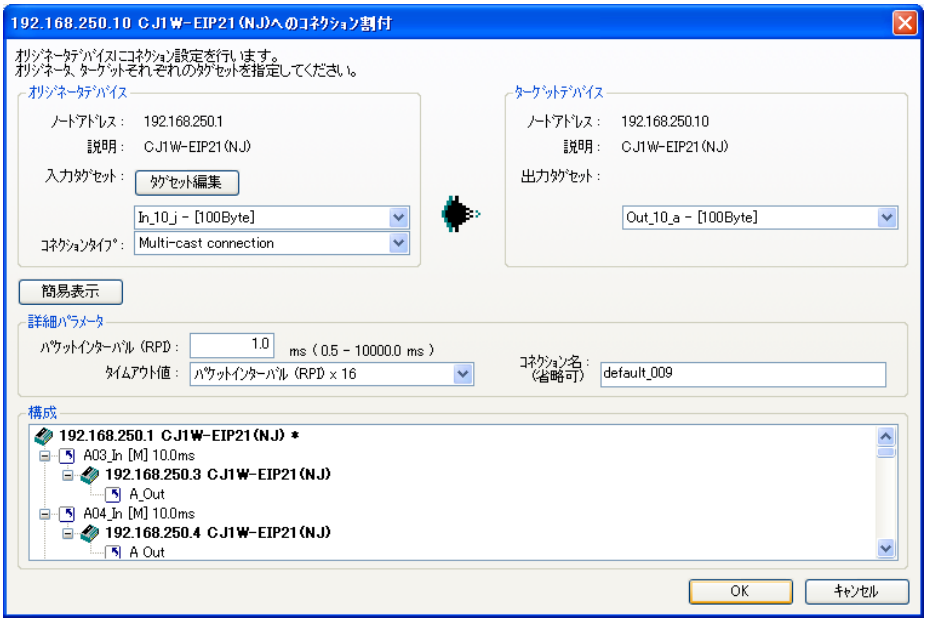
このケースでは、マルチキャストフィルタなしの場合に 104.50% となり、マルチキャストフィルタのないスイッチングハブの場合は、EtherNet/IP ユニットで通信負荷による通信障害が発生する可能性があります。

＜方法 3 ＞特定コネクションのパケットインターバル（RPI）のみ変更する場合

デバイス 192.168.250.1 の特定コネクションだけ、高速にタグデータリンクを行いたいとします。この場合、ネットワーク構成ウィンドウでデバイス 192.168.250.1 をダブルクリックします。



登録デバイス一覧にデバイス 192.168.250.10 とのコネクション情報が登録されていますので、これをダブルクリックします。



コネクション割付ウィンドウで新たなパケットインターバル（RPI）値として 1ms を入力し、［OK］ボタンをクリックします。デバイス 192.168.250.1 に対するタグデータリンク許容通信帯域幅に対する使用率が 48.67% になり特定のコネクションのパケットインターバル（RPI）が高速に設定されたことがわかります。



なお、この場合、デバイス 192.168.250.10 のタグデータリンク許容通信帯域幅に対する使用率も増えている（33.67%→ 50.33%）ことに注意してください。

14-3 タグデータリンクによる入出力応答時間

参考

ここでは、EtherNet/IP ユニットについて説明します。
NJ シリーズ CPU ユニット（形 NJ501- □□□□、形 NJ301- □□□□、形 NJ101- □□□□）の内蔵 EtherNet/IP ポートおよび CJ2H CPU ユニット（形 CJ2H-CPU6 □ -EIP）の内蔵 EtherNet/IP ポート、CJ2M CPU ユニット（形 CJ2M-CPU3 □）の内蔵 EtherNet/IP ポートについては、CPU ユニットとのデータ処理の性能が異なります。詳細は 🕮「7-4 NJ CPU 以外の機種とタグデータリンクを行う場合について」を参照してください。

🕮「7-1-7 タグデータリンクするデータの同期性について」の説明のとおり、更新タスクを設定したタグ（ネットワーク変数）は、設定した更新タスクのユーザプログラムの実行タイミングに合わせて更新されます。更新タスクを設定することで、システムサービスの影響を受けることなく、入出力応答時間を見積もることができます。

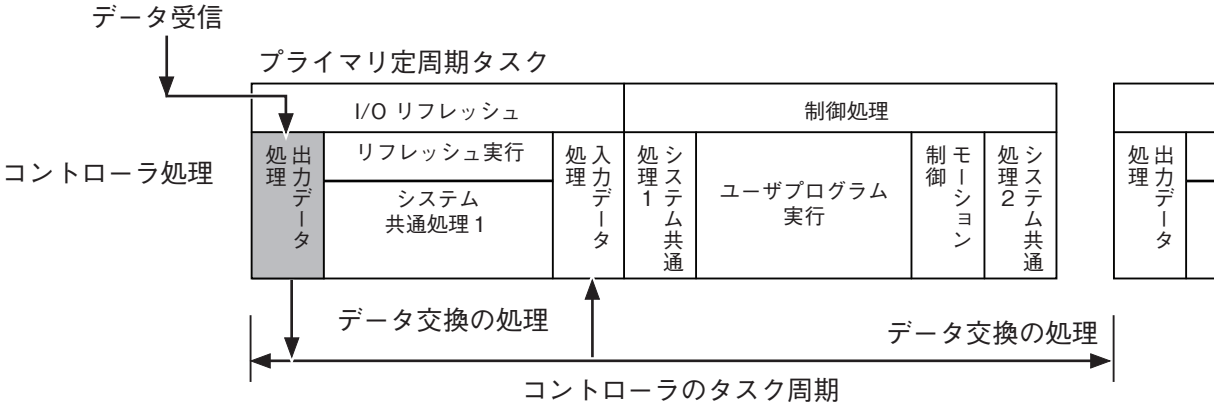
使用上の注意

必ずすべてのタグ（ネットワーク変数）について更新タスクを設定してください。更新タスクを設定したタグ（ネットワーク変数）と、更新タスクを設定しないタグ（ネットワーク変数）が両方とも存在する場合、システムサービスの影響を受けることがあり、本節に記載の入出力応答時間を得られないことがあります。

本節では、更新タスクを設定したときの入出力応答時間について説明します。

14-3-1 データ転送のタイミング

EtherNet/IP ユニットと CPU ユニットとのタグデータリンク時のデータ転送のタイミングについて説明します。
データ転送は、更新タスクに設定したタスクの「I/O リフレッシュ」の処理時に行われます。



更新タスクに設定が可能なタスクは、以下の 2 種類あります。

- プライマリ定周期タスク
  タスクの実行優先度が最も高く、高速・高精度に実行することができるタスク
- 定周期タスク
  プライマリ定周期タスクの実行の合間に実行するタスク

AT 指定したタグ（変数）は、更新タスクを設定する必要はなく、プライマリ定周期タスクでデータ転送されます。
タグ単位で、タグデータリンク処理を行うタスクを指定します。
更新タスクは、Sysmac Studio で、タグにする変数ごとに、更新タスクを設定します。
更新タスクの設定方法については、🕮『Sysmac Studio Version 1 オペレーションマニュアル』（SBCA-470）を参照してください。

14-3-2 EtherNet/IP ユニットのデータ処理時間

EtherNet/IP ユニットが、CPU ユニットとの間でデータ転送の処理に要する時間（データ処理時間）について説明します。

データ処理時間の概要

データ処理時間は、以下の①～④の要素があります。

① 変数へのアクセス時間

タグデータを転送するのに必要な時間を、変数へのアクセス時間として計算します。
タスク単位の計算となりますので、タスクに複数のタグセットが、更新タスクとして設定されている場合、複数のタグセットのタグの値で合計します。

変数へのアクセス時間は、次の式で算出します。
変数へのアクセス時間 [μs]= 変数の合計サイズ [バイト]×a+ 変数の数 ×b+ アクセス数 ×c+d
アクセス数：タグセットの数です。
a ～ d：定数です。値は次の表のとおりです。

CPU ユニットの形式	リフレッシュ方向	定数の値 (μs)			
		a	b	c	d
NJ501-□□□□	CPU ユニット→ EtherNet/IP ユニット	0.005	1.51*1	1.41	6.68
	EtherNet/IP ユニット→ CPU ユニット	0.009	2.95*2		
NJ301-□□□□	CPU ユニット→ EtherNet/IP ユニット	0.008	1.91*3	2.15	7.52
	EtherNet/IP ユニット→ CPU ユニット	0.010	3.52*4		
NJ101-□□□□	CPU ユニット→ EtherNet/IP ユニット	0.013	4.41	3.83	10.29
	EtherNet/IP ユニット→ CPU ユニット	0.016	5.48		

*1. CPU ユニット Ver.1.02 以前は「1.58」です。
*2. CPU ユニット Ver.1.02 以前は「3.18」です。
*3. CPU ユニット Ver.1.02 以前は「2.14」です。
*4. CPU ユニット Ver.1.02 以前は「4.08」です。

② データ転送回数

タグデータのデータ転送は、タスク処理の一部として実行されます。
データ転送の処理に要する時間が、「変数アクセス時間」を超えてしまう場合は、1 回のタスク周期ですべて送信されず、分割して転送されます。

データ転送回数＝（データ転送に必要な時間 *1 ÷ タスクに設定した「変数アクセス時間」*2）＋ 1*3

*1. ①で計算した変数へのアクセス時間です。
*2. 変数アクセス時間とは、変数へのアクセスにかかる処理時間の最大値のことです。
Sysmac Studio の［構成・設定］|［タスク設定］|［タスク設定］画面でタスクごとに設定します。
*3. CPU ユニットと EtherNet/IP ユニット間のデータ転送時の遅延時間

使用上の注意

EtherNet/IP ユニットの送受信できるタグデータリンクチャネル数は、最大で 9600CH です。タグデータリンクチャネル数が、CPU ユニットとの 1 回のデータ転送で処理できる量を超える場合も、複数回に分けてデータ転送が行われます。

③ I/O リフレッシュ時間

EtherNet/IP ユニットと CPU ユニット間で、実際にデータが転送される「I/O リフレッシュ」時の処理時間を計算します。
以下の 2 つの要素があります。

- 分割数
  タグデータリンクするデータサイズが、EtherNet/IP ユニットの 1 ユニット当たり送受信できる最大リンクデータサイズ以下（369664 バイト）でも、CPU ユニットとの 1 回のデータ転送で処理されるデータサイズが以下のデータサイズを越えた場合は、CPU ユニットと EtherNet/IP ユニット間でのデータ転送が複数回に分けて行われます。

  ［CPU ユニットとの 1 回のデータ転送で処理されるデータサイズ］
  出力／送信：最大約 14810 バイト
  入力／受信：最大約 14810 バイト

- I/O リフレッシュ処理時間
  実際の I/O リフレッシュ処理時間は、複数の条件により以下のようになります。
  （CPU 高機能ユニット装着台数 ×1ms）＋（I/O リフレッシュ時間 *）+1 タスク周期

*.I/O リフレッシュ時間の目安

総リンクデータサイズ / I/O リフレッシュ処理の種類	20 バイト	1444 バイト	2888 バイト	8664 バイト	12864 バイト以上
IN リフレッシュ	0.3ms	0.5ms	0.7ms	1.6ms	2.1ms
OUT リフレッシュ	0.1ms	0.3ms	0.6ms	1.2ms	2.3ms

④ 実際のデータ転送で要する時間

データ転送に要する実際の時間は、上記②と③で求めた数値を、以下のように計算して算出できます。

タスク周期 ×（データ転送回数（②）+ 分割数（③）− 1）＋ I/O リフレッシュ処理時間（③）

データ処理時間の計算例

タグデータリンクで、以下のようなタグデータのデータ転送を処理する場合の計算例を以下に示します。

- タグデータリンクする CPU の形式
  NJ501-□□□□
- コネクション方向
  自 EtherNet/IP ユニットは、オリジネータ側ノードとして設定
- 更新タスク
  プライマリ定周期タスク
  タスク周期：500 μs（変数アクセス時間 3%）
- タグセットの設定

タグセット	更新タスク	変数の数	変数の総サイズ
タグセット A	プライマリ定周期タスク	8 個	600 バイト
タグセット B	プライマリ定周期タスク	4 個	200 バイト
タグセット C	プライマリ定周期タスク	10 個	1000 バイト

1 **「変数へのアクセス時間」を以下のように計算します。**
[(8+4+10) 個 ×2.95 μs]+[(600+200+1000) バイト ×0.009 μs]+3×1.41 μs+6.68 μs
= 92.01 μs

2 **データ転送回数を計算します。**
[データ転送に必要な時間]　手順 1 の「変数へのアクセス時間」= 92.01 μs
[タスクに設定した変数アクセス時間]　500 μs×0.03 = 15 μs
[データ転送回数]　92.01 ÷ 15 μs + 1 = 7.13 回

データ転送回数は、約 8 回分必要となります。

3 **I/O リフレッシュ処理時間を計算します。**
1 台 ×1ms+0.5ms + 0.5ms=2ms (2000 μs)

4 **実際のデータ転送に要する時間を計算します。**
500 μs×（8 回 + 1 − 1）+ 2000 μs = 6000 μs

14-3-3　タグデータリンクによる、タスク周期への影響

タグデータのデータ転送は、タスク処理で実行されます。
よって、タスクデータの転送処理を、1 回のタスク周期できるようしたい場合は、タスク設定の「変数アクセス時間」と「タスク周期」の時間を調整して、タスク周期の時間を変更する必要がでてきます。

1　データ転送に必要な時間を、「変数へのアクセス時間」として計算します。

「変数へのアクセス時間」の計算式については、以下を参照してください。

🕮変数へのアクセス時間（P.14-23）

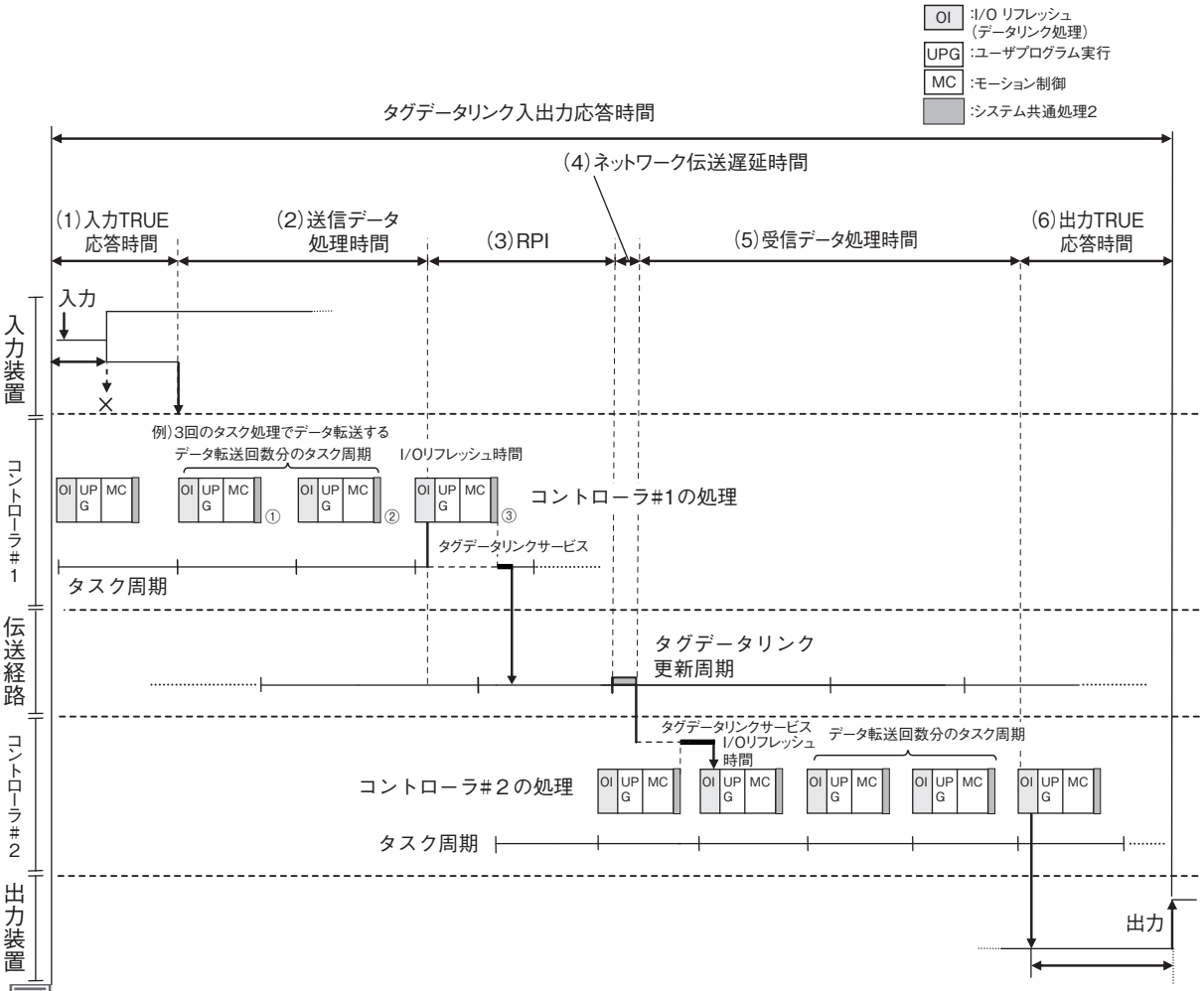
（注）変数アクセス時間は、タスクに複数のタグセットが更新タスクとして設定されている場合、複数のタグセットのタグの合計値となります。

2　タスク設定の「変数アクセス時間」を、手順 1 で算出した値以上になるように再設定します。

手順 1 を加算した時間で、タスク周期の時間を調整します。
「変数アクセス時間」やタスク周期の設定は、Sysmac Studio で設定します。
詳細は、🕮『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）を参照してください。

14-3-4　タグデータリンク入出力最大応答時間

入出力応答時間は、つぎの図に示す（1）～（6）の合計から求めることができます。



参考

- ユニット Ver,1,03 以降でのタグデータリンクの処理は、タグデータリンクサービスで実行します。
- ユニット Ver.1.00 ～ 1.02 でのタグデータリンクの処理はシステムサービスで実行します。タグデータリンクタイムアウトが発生する場合は、システムサービスの実行時間を見直してください。

(1) 入力 TRUE 応答時間

入力が実行されてから、実際にスイッチが TRUE となるまでの外部入力機器の遅延時間と、その入力データが CPU ユニットのメモリエリアに反映されるまでの時間です。入力スイッチの遅延時間は各機器の入力遅延を参照してください。
また、CPU ユニットのメモリエリアに反映されるまで、CPU のタスク周期 1 回分を要します。よって、入力 TRUE 応答時間は下記で求められます。

入力 TRUE 応答時間＝入力装置遅延時間 + タスク周期

(2) 送信データ処理時間

CPU ユニット上の変数を EtherNet/IP ユニットに転送するまでの時間です。

① データ転送は、タスク処理で実行されるため、送信データ処理時間は、タスク周期と同じになります。

また、1 回のタスク処理で送信できるデータ量（タスクの「変数アクセス時間」で設定）を超えてデータ転送する場合、複数のタスク周期にまたがってデータ転送が行われるため、タスク周期 × 転送回数の時間を加算します。

送信データ処理時間の求め方は、🕮「14-3-2 EtherNet/IP ユニットのデータ処理時間」を参照してください。

(3) パケットインターバル（RPI）

Network Configurator でコネクションごとに設定する通信の更新周期です。

(4) ネットワーク伝送遅延時間

Ethernet 回線上の伝送遅延は 50 μs 以下です。この遅延時間は無視できます。

(5) 受信データ処理時間

EtherNet/IP ユニットに受信したデータを CPU ユニット上の変数に転送する時間です。

① データ受信は、タスク処理で実行されるため、受信データ処理時間は、タスク周期と同じになります。

また、1 回のタスク処理で受信できるデータ量（タスクの「変数アクセス時間」で設定）を超えてデータ転送する場合、複数のタスク周期にまたがってデータ転送が行われるため、タスク周期 × 転送回数の時間を加算します。

受信データ処理時間の求め方は、🕮「14-3-2 EtherNet/IP ユニットのデータ処理時間」を参照してください。

さらに、データ転送は、1 タスク周期に 1 回行われるため、データ受信時に、たまたまそのタスク周期のデータ転送が終了していた場合、受信データの転送開始自体が、コントローラのタスク周期 1 回分だけ遅延します。

参考

他の複数のノードアドレスとコネクションがある場合、トータルで送受信するデータの量が増加するため、1 回の転送で処理できる量を超えるデータを送受信する場合があります。そうなった場合、データ転送回数は増加します。

(6) 出力 TRUE 応答時間

コントローラから出力が指示されてから、実際に出力が行われるまでの外部出力機器の遅延時間です。

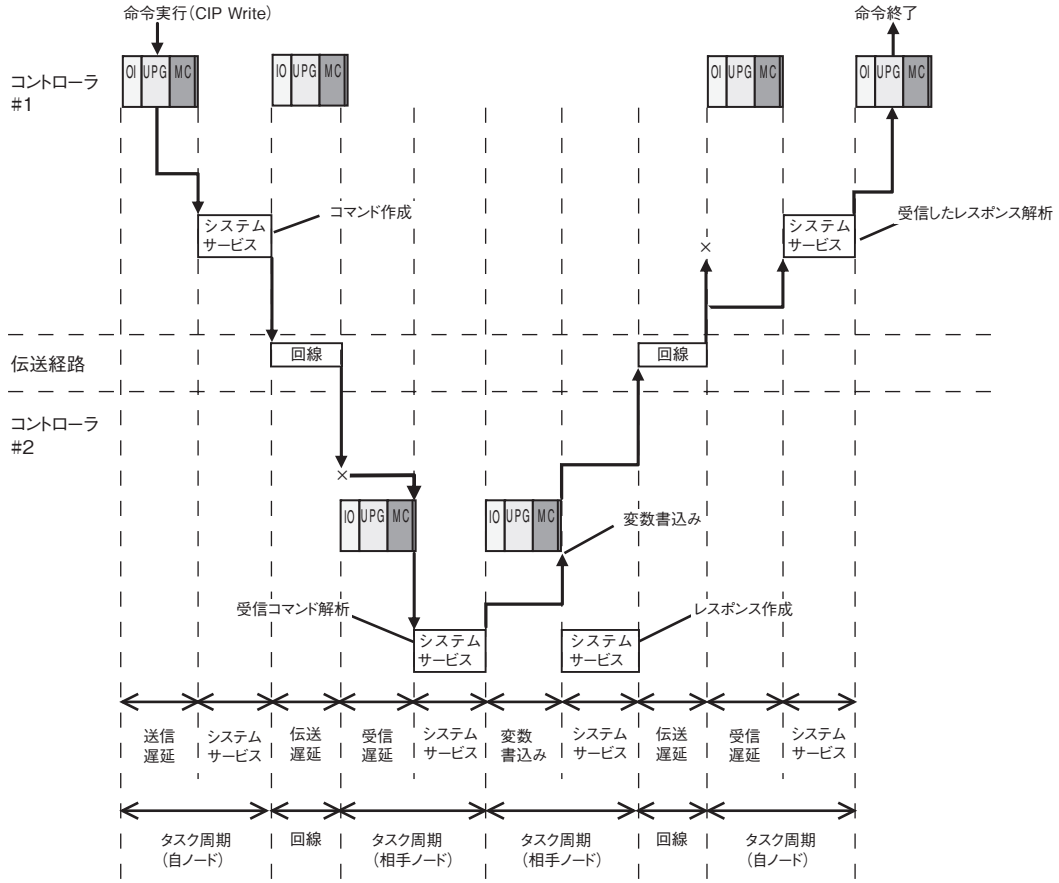
出力 TRUE 応答時間＝出力装置遅延時間 + CPU のタスク周期

参考

ノイズなどにより、入出力応答時間が延びることがあります。

14-4　メッセージサービス機能伝送遅延時間

CIP 通信命令（CIPWrite）のサービス処理で発生する遅延時間について記述します。



最大伝送遅延時間 ＝ 送信遅延+システムサービス+伝送遅延+受信遅延+システムサービス
（自ノードのタスク周期）　（相手ノードのタスク周期）
＋変数書込み+システムサービス+伝送遅延+受信遅延+システムサービス
（相手ノードのタスク周期）　（自ノードのタスク周期）

遅延時間が発生する各処理は、上図のように、それぞれ各ノードのタスク周期内で処理されます。
また、回線による遅延時間は、以下のようになります。

● 伝送遅延

Ethernet 回線上の伝送遅延は 50 μs 以下です。この遅延時間は無視できます。

参考

• 伝送時間は、実際の使用環境によって計算値より長くなることがあります。
長くなる要因としては、ネットワークの負荷率（回線の混み具合）、各ノードのウィンドウサイズ、または EtherNet/IP ユニット自身の負荷（同時にタグデータリンク機能などを使用しているとき）、システムの構成などがあります。
• CIP 通信処理は、システムサービスで実行します。
CIP 通信命令にて、タイムアウトが発生する場合は、システムサービスの実行時間を見直してください。

15

異常とその処理

15-1 Network Configurator によるステータスの確認方法 15-2
15-1-1 Network Configurator のデバイスモニタ機能 15-2

15-2 LED 表示の意味と異常時の処置 15-10
15-2-1 EtherNet/IP ユニットで発生する異常と処置 15-10

15-3 コネクションステータスのコード一覧と処置方法 15-21

15-4 異常履歴機能 15-28
15-4-1 異常履歴データ 15-28
15-4-2 異常履歴コード一覧 15-29

15-5 イベントログ機能の概要 15-33
15-5-1 イベントログ機能の概要 15-33
15-5-2 異常一覧 15-34
15-5-3 異常の内容 15-37

15-6 トラブルシューティング 15-48
15-6-1 EtherNet/IP ユニットの装着で CPU ユニットの ERROR が点灯／点滅する場合 15-48
15-6-2 Ethernet の一般的なトラブル 15-48
15-6-3 タグデータリンクがスタートしない場合 * 15-49
15-6-4 タグデータリンクのトラブル 15-50
15-6-5 メッセージタイムアウトのトラブル 15-51

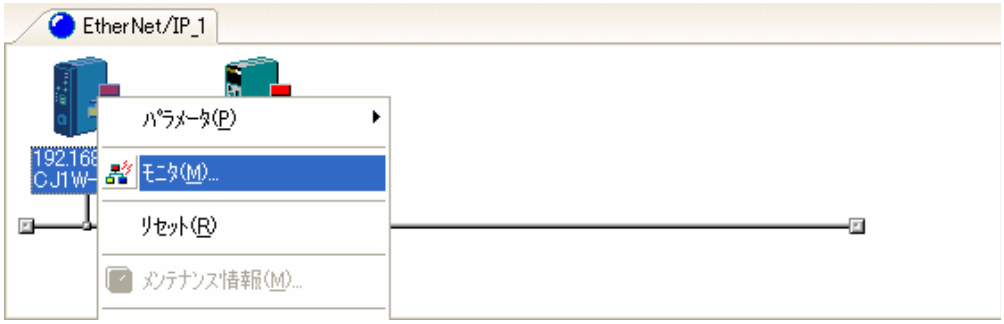
15-7 清掃と点検 15-52
15-7-1 清掃方法 15-52
15-7-2 点検方法 15-52

15-8 装置交換時の留意点 15-54
15-8-1 ユニット交換時のお願い 15-54
15-8-2 ユニット交換後の設定 15-54
15-8-3 ユニットの交換方法 15-55

15-1 Network Configurator によるステータスの確認方法

15-1-1 Network Configurator のデバイスモニタ機能

Network Configurator をオンライン接続状態にし、ステータスを確認したいデバイスを選択し、マウスの右ボタンクリックで表示されるメニューから［モニタ］を選択します。



［デバイスモニタ］ダイアログボックスが表示されます。

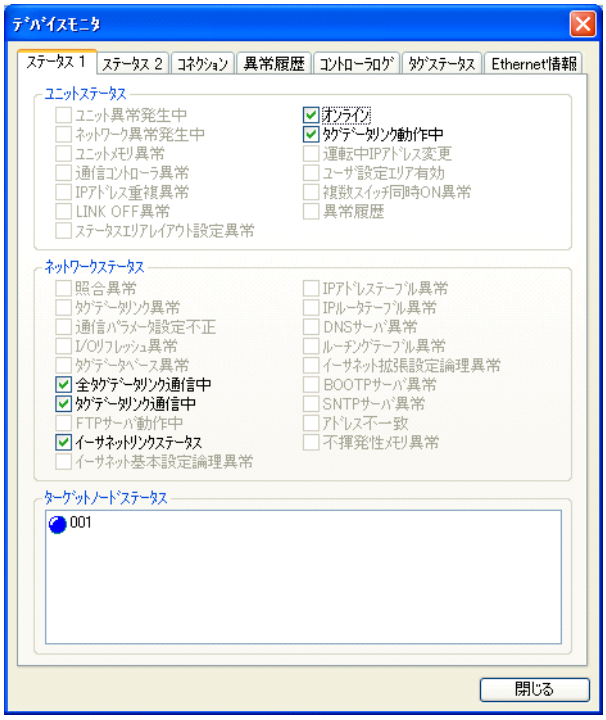
参考

モニタ中に通信異常が発生すると、最後に収集された情報が表示され続けます。
モニタを再開するには、一旦［デバイスモニタ］ダイアログボックスを閉じてから、再度［デバイスモニタ］を選択し直してください。

●［ステータス１］タブ

ステータス用の CJ ユニットデバイス変数でモニタできる各ステータス（ユニットステータス 1/2、通信ステータス 1/2/3）がチェックボックスで表示され、TRUE（1）のステータスが☑マークされます。

また、正常ターゲットノード情報のうち、タグデータリンクのオリジネータ設定されているノードの情報が表示されます。本ノードとのタグデータリンクコネクションが全て正常に確立している場合は青色で、1 つでも未確立のコネクションがある場合は赤色表示されます。



● [ステータス 2] タブ

ステータス用の CJ ユニットデバイス変数でモニタできるターゲットテーブル情報の正常ターゲットノード情報のうち、ターゲットノード PLC 本体運転情報とターゲットノード PLC 本体異常情報から、タグデータリンクのオリジネータ設定されているノードの情報が表示されます。正常の場合に青色、異常の場合は赤色で表示されます。
また、FINS/TCP コネクション接続ステータスがチェックボックスで表示され、接続中のコネクションのステータスが☑マークされます。

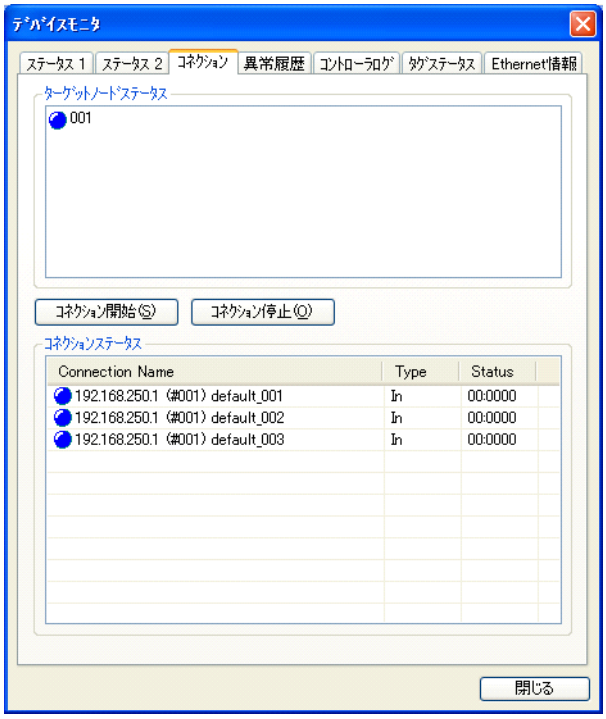


参考

ターゲットコントローラステータスは、オリジネータとターゲットの双方のコネクションすべてのタグセットに、コントローラステータスを「含める」にした場合に有効となります。
「含めない」の場合、ステータスはグレー表示となります。

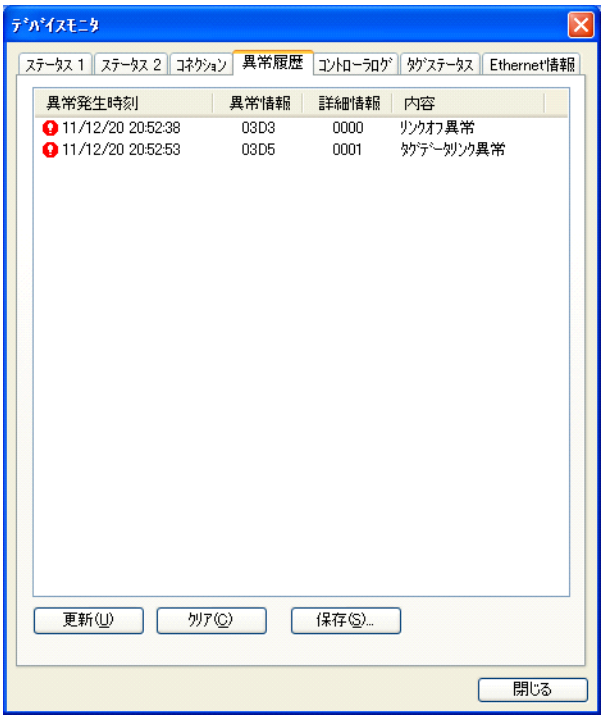
●［コネクション］タブ

正常ターゲットノード情報のうち、本ノードがオリジネータとなっているターゲットノードの情報を表示します。
本ノードとのタグデータリンクコネクションが全て正常に確立している場合は青色で、１つでも未確立のコネクションがある場合は赤色で表示されます。
ただし、本ノードのコネクションを停止した場合は灰色で表示されます。
また、コネクションステータスでは、オリジネータ設定されている各コネクションの現在状態が［Status］で表示されます。この情報をもとにして、タグデータリンクの異常要因を切り分けることができます。
コネクションステータス情報の詳細は、📖「15-3 コネクションステータスのコード一覧と処置方法」を参照してください。



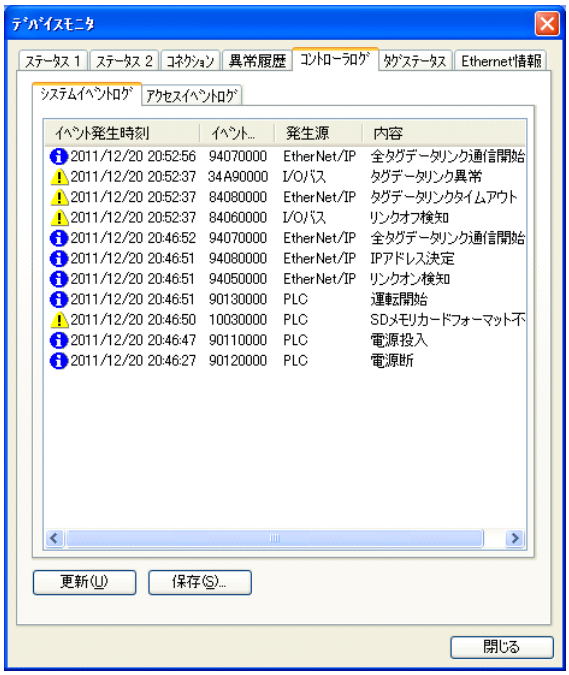
●［異常履歴］タブ

EtherNet/IP ユニットが保持している異常履歴を表示します。
過去に発生した異常が記録されており、必要により PC 上のファイルに保存することができます。
なお、異常履歴は電源 OFF で消去されるものと、保持する（消去されない）ものがあります。
異常情報の詳細は、「15-4 異常履歴機能」を参照してください。



●［コントローラログ］タブ

EtherNet/IP ユニットが取り付けられている PLC システムの CPU ユニットが保持しているイベント履歴を表示します。
過去に発生した異常が記録されており、必要に応じて PC 上のファイルに保存することができます。
イベントの詳細は、CPU ユニットのユーザーズマニュアルを参照してください。

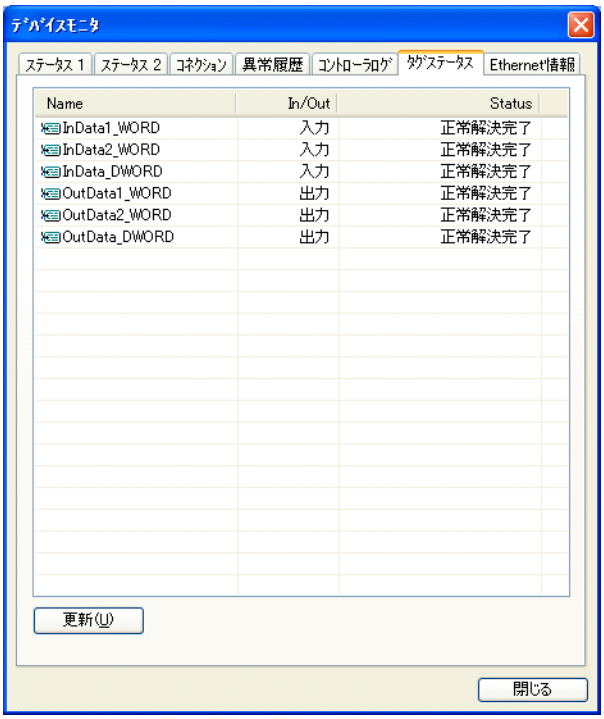


●［タグステータス］タブ

タグデータリンクの各タグが、CPU ユニットとの間で正しくデータ交換が行える設定になっているかを表示します。
設定されている状態により、以下の Status 表示になります。

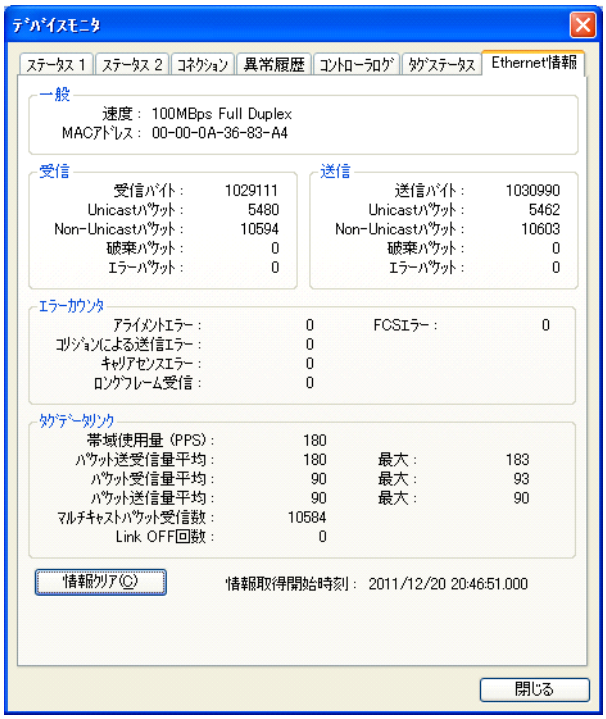
－正常解決完了：	正常なデータ交換が行えます。
－解決中：	タグが配置されている変数を解決中です。 この状態から正常解決完了に移行したとき、コネクション開設や正常なデータ交換が開始されます。
－サイズ不一致：	ネットワーク変数とタグ設定でサイズが異なります。 この異常が発生しているタグではコネクションは開設されません。
－タグなし：	タグ設定で指定したネットワーク変数が CPU ユニットの変数テーブルに設定されていません。または、共用体型変数のメンバで指定されておらず、共用体全体で指定されています。 この異常が発生しているタグではコネクションは開設されません。
－コントローラ I/F 異常：	CPU ユニット電源投入時のシステムサービスの実行時間が不足しています。［電源投入時の起動待ち時間］の設定を見直してください。［電源投入時の起動待ち時間］の設定については、『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）』を参照してください。 ［電源投入時の起動待ち時間］の設定を見直しても改善しない場合、CPU ユニットとのバス I/F に問題があります。LED 表示や異常履歴により原因を特定してください。
－属性異常：	Const 属性のため書込みできません。

正常解決完了以外の状態の場合は、タグデータリンク設定、もしくは NJ CPU ユニットの変数テーブルにおけるネットワーク変数の設定を見直してください。



●［Ethernet 情報］タブ

EtherNet/IP ユニットの通信ドライバレベルのステータスを表示します。
各種エラーカウンタ情報により、通信障害の発生状況を確認できます。
各種タグデータリンク情報では、帯域使用量（pps）などを確認することができます。



15-2 LED 表示の意味と異常時の処置

15-2-1 EtherNet/IP ユニットで発生する異常と処置

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント (*)					
CPU ユニットとのやり取りに関する異常							
赤点滅	消灯	H1	ユニット番号重複	ユニット番号が他のユニットと重複している	動作は停止します。	−	ユニット番号を正しく設定し直し、EtherNet/IP ユニットをリスタートしてください。
赤点滅	消灯	H2	CPU ユニットが不良	−	動作は停止します。	−	再起動しても再発する場合は、CPU ユニットを交換してください。
赤点灯	消灯	H3	EtherNet/IP ユニットが不良	−	動作は停止します。	−	再起動しても再発する場合は、EtherNet/IP ユニットを交換してください。
赤点滅	消灯	H4	ノードアドレス設定異常	ノードアドレス設定スイッチの設定に誤り（00 または FF）がある	動作は停止します。	−	ノードアドレスを正しく設定し直し、EtherNet/IP ユニットをリスタートしてください。
赤点滅	消灯	H6	CPU ユニットが不良	−	異常履歴を登録します。（時計情報はすべて 0） 動作は停止します。	000F	再起動しても再発する場合は、CPU ユニットを交換してください。
赤点滅	消灯	H7	I/O テーブル未登録	CPU ユニットの I/O テーブルに登録されていない	動作は停止します。	0006	I/O テーブルを作成してください。
赤点滅	—	H8	SD メモリカードバックアップ／ Sysmac Studio コントローラバックアップのリストアの失敗	何らかの理由により、バックアップのリストアに失敗した	EtherNet/IP ユニットの設定内容はすべてクリアします。ただし、該当ファイルなし、メモリ未装着、形式不一致のとき、設定内容はクリアされません。	—	再度、簡易バックアップ手順を行い再発する場合は、メモリカードもしくは EtherNet/IP ユニット（内蔵ポートの場合は CPU ユニット）を交換してください。
赤点滅	—	H9	I/O バス異常	CPU ユニットとのデータ交換時に異常が発生した	• タグデータリンクでオリジネータのコネクションは通信を停止します。 • タグデータリンクでターゲットのコネクションは通信を継続し、通信データにコントローラステータスを設定している場合はターゲットノード PLC 本体異常情報の該当ビットが TRUE になります。	000E	CPU ユニットの動作環境を見直してください。

*　7 セグメント LED はユニットのノードアドレスと異常を交互表示します。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
CPU ユニットとのやり取りに関する異常							
赤点滅	—	HA	CPU ユニットメモリ異常	CPU ユニットへのルーチングテーブルなどの読出しでパリティエラーが発生した	異常履歴を登録します。動作は継続します。ルーチングテーブル読出しエラーの場合は「ルーチングテーブルなし」として扱います。	0012	CPU ユニットにルーチングテーブルを再登録して、CPU ユニットを再起動してください。再発する場合は CPU ユニットを交換してください。
				CPU ユニットが保持するタグデータベースがメモリ異常などを発生した	• タグデータリンクやユニットステータスエリアで変数名（タグ名）が指定されている場合は、タグデータリンクユーザ指定ステータスエリアのリフレッシュを停止し、タグデータリンクは以下の動作となります。 • タグデータリンクでオリジネータのコネクションは通信を停止します。 • タグデータリンクでターゲットのコネクションは通信を継続し、通信データにコントローラステータスを設定している場合はターゲットノード PLC 本体異常情報の該当ビットが TRUE します。 注：この異常は復帰する可能性があるため、復帰時はタグデータリンクの開始処理を行い、正常状態に戻ります。	0017	エラーの解除は CPU ユニットへタグデータを再度ダウンロードしてください。再発する場合は CPU ユニットを交換してください。
赤点滅	消灯	Hb	CPU ユニットとのイベントサービスのタイムアウト	CPU ユニットへのルーチングテーブルなどの読み出しがタイムアウトした	動作は停止します。	0011	再起動しても再発する場合は、EtherNet/IP ユニットを交換してください。
赤点滅	—	HC	ルーチングテーブル異常	ルーチングテーブルの設定に論理的なエラーがある	ルーチングテーブルなし状態で動作を継続します。	021A	ルーチングテーブルを作成し直してください。

*　7 セグメント LED はユニットのノードアドレスと異常を交互表示します。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント (*)					
CPU ユニットとのやり取りに関する異常							
赤点滅	—	HE	CPU ユニットサービス監視異常	CPU ユニットからのサービスが一定時間内に完了しない 通常は 11s で監視	• タグデータリンクでオリジネータのコネクションは通信を停止します。 • タグデータリンクでターゲットのコネクションは通信を継続し、通信データにコントローラステータスを設定している場合はターゲットノード PLC 本体異常情報の該当ビットが TRUE になります。 注：この異常は復帰する可能性があるため、復帰時はタグデータリンクの開始処理を行い、正常状態に戻ります。	0002	CPU ユニットの動作環境を見直してください。
赤点滅	—	HF	CPU ユニット WDT 異常	CPU ユニットで異常が発生した	• タグデータリンクでオリジネータのコネクションは通信を停止します。 • タグデータリンクでターゲットのコネクションは通信を継続し、通信データにコントローラステータスを設定している場合はターゲットノード PLC 本体異常情報の該当ビットが TRUE します。	0001	CPU ユニットを交換してください。

*　7 セグメント LED はユニットのノードアドレスと異常を交互表示します。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
CPU ユニットとのやり取りに関する異常							
赤点滅	—	HH	CPU ユニット停止異常	CPU ユニットで停止異常が発生した	• タグデータリンクでオリジネータのコネクションは通信を停止します。 • タグデータリンクでターゲットのコネクションは通信を継続し、通信データにコントローラステータスを設定している場合はターゲットノード PLC 本体異常情報の該当ビットが TRUE します。 • タグデータリンクの送信データが負荷遮断動作の設定にしたがい 0 クリアデータ、もしくはそのままのデータで送信を継続します。	0015	CPU ユニットの異常要因を取り除いてください。 異常要因が取り除かれるとタグデータリンクは自動復旧します。
ソフトスイッチ操作での異常							
—	—	C6	複数スイッチ ON	複数のソフトスイッチが同時に ON された、またはひとつのソフトスイッチ処理が完了しないうちに、他のスイッチが ON された	• 7 セグメント LED へ異常コードを表示するだけです。（30 秒間） • 次回のソフトスイッチ操作が正常に完了したとき、7 セグメント LED の異常表示を消去します。[*_MultiSwOnErr（複数スイッチ同時 ON 異常）が TRUE]	—	1 つずつソフトスイッチ操作を再実行してください。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
—	—	d5	照合異常（ターゲット不在）	デバイスパラメータに登録されているターゲットが存在しない	ターゲットに対して定期的に再コネクション処理を続行します。[*_TDLOpnErr(照合異常) と *_UnitErr（ユニット異常発生中）と *_NetErr（ネットワーク異常発生中) が TRUE]	—	以下の項目を検討してください。 • コネクション登録されているノードの電源が入っているか • ケーブルが繋がっているか • ケーブルの断線・ゆるみがないか • ターゲットノードまたは、オリジネータの CIP メッセージ通信が停止していないか • ターゲットノードおよび、経路上の機器の Packet Filter 機能で CIP 通信が許可されているか • ノイズが多くないか

*　7 セグメント LED はユニットのノードアドレスと異常を交互表示します。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
タグデータリンクに関する異常							
—	—	d6	コネクション確立失敗	コネクション確立時、オリジネータとターゲット内でデバイスパラメータが不適合（変数名の不一致、サイズの不一致）、使用コネクション数の不足などのため、コネクションの確立に失敗した	ターゲットに対して定期的に再コネクション処理を続行します。[*_TDLOpnErr（照合異常）と *_UnitErr（ユニット異常発生中）と *_NetErr（ネットワーク異常発生中）がTRUE]	03D4	デバイスパラメータを正しいデータに修正した後、Network Configurator からデバイスパラメータをダウンロードしてください。
—	—	d9	タグデータリンク異常	タグデータリンクでタイムアウトが発生した（ターゲットからのタグデータがタイムアウト値で指定した時間を越えて受信できなかったとき）	タグデータリンク異常のターゲットに対して定期的に再コネクション処理を続行します。[*_TDLErr（タグデータリンク異常）と *_UnitErr（ユニット異常発生中）と *_NetErr（ネットワーク異常発生中がTRUE]	03D5	以下の項目を検討してください。 • コネクション登録されているノードの電源が入っているか • ケーブルが繋がっているか • ケーブルの断線・ゆるみがないか • ターゲットノードの CIP メッセージ通信が停止していないか • オリジネータおよび、経路上の機器の Packet Filter 機能で CIP 通信が許可されているか • ノイズが多くないか

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
赤点滅	—	E9	メモリアクセス異常	ユニット内部の不揮発性メモリに異常が発生した。次のタイミングで発生 1. 異常履歴の書き込みで異常 2. デバイスパラメータ設定の書き込みで異常 注：読み出し時のサムチェック異常は本異常には含まない	1. の場合： RAM 上にのみ異常履歴が残り、以降の不揮発性メモリへの書き込みはすべて無視されます。それ以外は通常の動作を続行します。（RAM への異常履歴の登録も続けます） 2. の場合： タグデータリンクおよびメッセージ通信の動作は続行します。[*_UnitErr（ユニット異常発生中）と *_UnitMemErr（ユニットメモリ異常）と *_MemErr（不揮発性メモリ異常）とが TRUE]	0602	Sysmac Studio からユニット設定を、Network Configurator からデバイスパラメータをダウンロードしてください。 再発する場合は、EtherNet/IP ユニットを交換してください。

*　7 セグメント LED はユニットのノードアドレスと異常を交互表示します。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
メモリアクセスに関する異常							
赤点滅	—	E8	デバイスパラメータ異常	デバイスパラメータで設定されたI/Oエリアが CPUユニットに存在しない パラメータサムチェック異常もしくは設定論理異常 ユニット設定後に異なるPLC（NJ⇔CJ2など）へ装着した。	ユニット内部の不揮発性メモリに保存されているパラメータ設定の内容に異常がありました（不揮発性メモリへの書込中の電源断などが原因）。 [*_UnitErr（ユニット異常発生中）と*_CommParamErr（通信パラメータ設定不正）がTRUE]	021A	Sysmac Studioから設定をダウンロードしなおしてください。 再発する場合は、EtherNet/IPユニットを交換してください。
赤点滅	—	EA	IP拡張設定異常		ユニット内部の不揮発性メモリに保存されているパラメータ設定の内容に異常がありました（不揮発性メモリへの書込中の電源断などが原因）。 [*_UnitErr（ユニット異常発生中）と*_CommParamErr（通信パラメータ設定不正）がTRUE]	03D1	異常履歴詳細情報から異常要因を特定し、正しい設定内容に修正した後、Sysmac Studioから設定をダウンロードしてください。
赤点滅	—	F2	Ethernet基本設定異常		—	03D0	Sysmac StudioのTCP/IPタブ、Ethernetタブの設定を、もしくはNetwork Configuratorからの TCP/IP設定をダウンロードしてください。
—	—	E1	Ethernetリンク未検出	スイッチングハブとの間でリンクが検出できていない 注：タグデータリンクが未設定の場合は、本異常は発生しません。	• ユニットはオフライン状態となり、通信動作はできません。すべての通信要求にエラーを返します。 • CPUユニットとのデータ交換動作（リフレッシュ）は続行します。 [*_UnitErr（ユニット異常発生中）と*_NetErr（ネットワーク異常発生中）と*_LkOffErr（リンクオフ異常）がTRUE、*_LkSta（リンクステータス）がFALSE]	03D3	以下の項目を検討してください。 • ケーブルが繋がっているか • ケーブルの断線・ゆるみがないか • ノイズが多くないか

*　7セグメントLEDはユニットのノードアドレスと異常を交互表示します。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
メモリアクセスに関する異常							
—	—	E3	サーバ接続異常	DNS サーバとの間で異常が発生した。	[*_DNSSvrErr（DNS サーバ異常）が TRUE]	履歴： 03C4 詳細： 00xx	以下のいずれかの処置を行ってください。 • DNS サーバの設定を正しく設定してください。 • 通信経路（EtherNet/IP ユニット、ケーブル接続、ハブ、ルータ、サーバ）を点検し、障害となっている不具合を取り除いてください。
ネットワークに関する異常							
—	—	E3	サーバ接続異常	BOOTP サーバとの間で異常が発生した。 1. BOOTP サーバからの応答がない 2. BOOTP サーバにより EtherNet/IP ユニットに設定されようとした IP アドレスが不正値	1. の場合： BOOTP サーバから応答があるまで BOOTP サーバへ要求しつづけます。 この間、ユニットはオフライン状態となり、通信動作はできません。すべての通信要求にエラーを返します。 CPU ユニットとのデータ交換動作（リフレッシュ）は続行します。 2. の場合： デフォルトの IP アドレス（(192.168.250.1）ノードアドレス）で動作します。 [*_UnitErr（ユニット異常発生中）と *_NetErr（ネットワーク異常発生中）と *_BootpSvrErr（BOOTP サーバ異常）が TRUE]	履歴： 03C4 詳細： 06xx	以下のいずれかの処置を行ってください。 • BOOTP サーバの設定を正しく設定してください。 • 通信経路（EtherNet/IP ユニット、ケーブル接続、ハブ、ルータ、サーバ）を点検し、障害となっている不具合を取り除いてください。

*　7 セグメント LED はユニットのノードアドレスと異常を交互表示します。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
ネットワークに関する異常							
—	—	E3	サーバ接続異常	SNTP サーバとの間で異常が発生した。	[*_SNTPSvrErr（SNTP サーバ異常）が TRUE]	履歴： 03C4 詳細： 03xx	以下のいずれかの処置を行ってください。 • SNTP サーバの設定を正しく設定してください。 • 通信経路（EtherNet/IP ユニット、ケーブル接続、ハブ、ルータ、サーバ）を点検し、障害となっている不具合を取り除いてください。
				SNMP トラップへの送信が異常となった。	—	履歴： 03C4 詳細： 07xx	以下のいずれかの処置を行ってください。 • SNMP トラップの設定を正しく設定してください。 • 通信経路（EtherNet/IP ユニット、ケーブル接続、ハブ、ルータ、サーバ）を点検し、障害となっている不具合を取り除いてください。
—	赤点灯	F0	IP アドレス重複	EtherNet/IP ユニットの IP アドレスが、他のノードの IP アドレスと重複している	• ユニットはオフライン状態となり、通信動作はできません。すべての通信要求にエラーを返します。 • CPU ユニットとのデータ交換動作（リフレッシュ）は続行します。 [*_UnitErr（ユニット異常発生中）と *_NetErr（ネットワーク異常発生中）と *_IPAdrDupErr（IP アドレス重複異常）が TRUE]	0211	他ノードの IP アドレスを確認してください。 重複しないように再設定後、EtherNet/IP ユニットをリスタートしてください。

*　7 セグメント LED はユニットのノードアドレスと異常を交互表示します。

LED			異常項目	要因	ユニットの動作	異常履歴（Hex）	対処
MS	NS	7セグメント(*)					
ネットワークに関する異常							
赤点滅	—	F3	アドレス不一致	IP アドレス変換方式が自動生成方式に設定されているが、自 IP アドレスの最終アドレス（1 バイト）と、ノードアドレス設定スイッチの値が一致していない	• 確定した IP アドレスの値を自 IP アドレスとして動作も継続します。[*_AdrMismatchErr（アドレス不一致）が TRUE]	—	IP アドレスまたはノードアドレス設定スイッチを見直してください。
赤点滅	消灯	F4	通信コントローラ異常	EtherNet/IP ユニット内部の通信コントローラで異常が発生した	• ユニットはオフライン状態となり、通信動作はできません。すべての通信要求にエラーを返します。 • CPU ユニットとのデータ交換動作（リフレッシュ）は続行します。[*_UnitErr（ユニット異常発生中）と *_NetErr（ネットワーク異常発生中）と *_LANHwErr（通信コントローラ異常）が TRUE]	020F	再起動しても再発する場合は EtherNet/IP ユニットを交換してください。
赤点滅	—	C8	運転中のノードアドレス設定変更	運転中にノードアドレス設定スイッチが変更された	動作は継続します。[*_IPAdrChgErr（運転中 IP アドレス変更）が TRUE]	—	正しいアドレスに設定したあとに、EtherNet/IP ユニットをリスタートしてください。
ユニットに関する異常							
赤点灯	消灯	—	高機能ユニット異常	高機能ユニット内部で異常が発生した	異常履歴を登録。ユニットは停止します。	0601	CPU ユニットを再起動してください。再発する場合は、EtherNet/IP ユニットを交換してください。

*　7 セグメント LED はユニットのノードアドレスと異常を交互表示します。

15-3 コネクションステータスのコード一覧と処置方法

タグデータリンクのコネクションステータス内容による異常の切り分けと処置方法を示します。コネクションステータスは、Network Configurator のデバイスモニタ機能の［コネクション］タブを使用して読出します。詳細は、📖「15-1 Network Configurator によるステータスの確認方法」を参照してください。

各コネクションステータスに対する、異常発生要因と、以下の構成ごとの対処方法を示します。

	オリジネータ	ターゲット
構成１	形 CS1W-EIP21/EIP21S、 形 CJ1W-EIP21/EIP21S、 形 CJ2H-CPU □□ -EIP、形 CJ2M-CPU3 □、 形 NJ501- □□□□、形 NJ301- □□□□、 形 NJ101- □□□□	形 CS1W-EIP21/EIP21S、 形 CJ1W-EIP21/EIP21S、 形 CJ2H-CPU □□ -EIP、形 CJ2M-CPU3 □、 形 NJ501- □□□□、形 NJ301- □□□□、 形 NJ101- □□□□
構成２	形 CS1W-EIP21/EIP21S、 形 CJ1W-EIP21/EIP21S、 形 CJ2H-CPU □□ -EIP、形 CJ2M-CPU3 □、 形 NJ501- □□□□、形 NJ301- □□□□、 形 NJ101- □□□□	他社製
構成３	他社製	形 CS1W-EIP21/EIP21S、 形 CJ1W-EIP21/EIP21S、 形 CJ2H-CPU □□ -EIP、形 CJ2M-CPU3 □、 形 NJ501- □□□□、形 NJ301- □□□□、 形 NJ101- □□□□

コネクション ステータス		異常発生要因	処置方法		
General Status (HEX)	Additional Status (HEX)		構成 1	構成 2	構成 3
00	0000	正常状態のコード： コネクションが開設されタグリンクが正常に通信している。	－	－	－
01	0100	ターゲット側が応答した異常コード： 同一のコネクションを多重に開設しようとした。	本異常は発生しません。	ターゲット側の動作仕様に依存します（本来は発生しえない異常のためターゲット側メーカにお問い合わせください）。	オリジネータ側の動作仕様に依存します（本来は発生しえない異常のためオリジネータ側メーカにお問い合わせください）。
01	0103	ターゲット側が応答した異常コード： サポート外トランスポートクラスでコネクション開設が要求された。	本異常は発生しません。	ターゲット側が Class1 をサポートしているか確認してください。	オリジネータ側が Class1 をサポートしているか確認してください。

コネクション ステータス		異常発生要因	処置方法		
General Status (HEX)	Additional Status (HEX)		構成 1	構成 2	構成 3
01	0106	コンシューマ側重複： 1 つのコンシューマ側データに複数コネクションを開設しようとした。	タグデータリンクの停止／起動を実行した場合、タイミングにより発生することがありますが、自動的に復旧します。	ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。	タグデータリンクの停止／起動を実行した場合、タイミングにより発生することがありますが、自動的に復旧します。
01	0107	ターゲット側が応答した異常コード： コネクションをクローズしようとしたが、すでに該当コネクションがクローズしている。	本異常は発生しません。	本異常は発生しません。	すでにコネクションはクローズされているため、異常ではありません。
01	0108	ターゲット側が応答した異常コード： サポート外コネクションタイプでコネクション開設が要求された。	本異常は発生しません。	ターゲット側で使用可能なコネクションタイプを確認してください（ターゲット側メーカにお問い合わせください）。 マルチキャスト、ポイントツーポイント以外の設定はできません。	オリジネータ側のコネクションタイプを確認してください（マルチキャスト、ポイントツーポイント以外が指定されている場合は、エラーとなります）。
01	0109	ターゲット側が応答した異常コード： コネクションのサイズがオリジネータとターゲットの設定で異なる。	オリジネータとターゲットのコネクション設定（サイズ）を確認してください。		
01	0110	ターゲット側が応答した異常コード： ターゲットが設定ダウンロード中などコネクション開設できない状態にある。	ターゲット側のタグデータリンクが停止状態となっていないか確認してください（ソフトスイッチのタグデータリンク通信開始／停止にて通信を再開します）。	ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。	ターゲット側のタグデータリンクが停止状態となっていないか確認してください（ソフトスイッチのタグデータリンク通信開始／停止にて通信を再開します）。
01	0111	ターゲット側が応答した異常コード： 仕様よりも大きな値で RPI が設定された。	本異常は発生しません。	ターゲット側の RPI 設定仕様を確認してください。	オリジネータ側の RPI 設定値を 10 秒以下に設定してください。

コネクション ステータス		異常発生要因	処置方法		
General Status（HEX）	Additional Status（HEX）		構成 1	構成 2	構成 3
01	0113	オリジネータ側で発生した、もしくはターゲット側が応答した異常コード： コネクション数（256 本）［CJ2M-EIP21：32 本］の仕様を超えて開設が要求された。	オリジネータとターゲットのコネクション設定（コネクション数）を確認してください。	オリジネータとターゲットのコネクション設定（コネクション数）を確認してください。 他社製品のコネクション数仕様を確認してください。	オリジネータとターゲットのコネクション設定（コネクション数）を確認してください。 他社製品のコネクション数仕様を確認してください。
01	0114	ターゲット側が応答した異常コード： 開設時の Vendor ID、Product Code 確認が不一致だった。	本異常は発生しません。	ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。 ターゲット側製品の EDS ファイルが正しいか確認してください。	オリジネータ側のコネクション設定を確認してください。
01	0115	ターゲット側が応答した異常コード： 開設時の Product Type 確認が不一致だった。	本異常は発生しません。	ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。 ターゲット側製品の EDS ファイルが正しいか確認してください。	オリジネータ側のコネクション設定を確認してください。
01	0116	ターゲット側が応答した異常コード： 開設時の Major/Minor Rev. 確認が不一致だった。	ターゲット製品とコネクションで設定した Major/Minor Rev. 情報を確認しください。必要により最新の EDS ファイル入手し設定しなおしてください。	ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。 ターゲット側製品の EDS ファイルが正しいか確認してください。	オリジネータ側のコネクション設定を確認してください。

コネクション ステータス		異常発生要因	処置方法		
General Status (HEX)	Additional Status (HEX)		構成 1	構成 2	構成 3
01	0117	ターゲット側が応答した異常コード： コネクションのターゲット変数で指定したタグセットがターゲット側に存在しない。	オリジネータ、ターゲットのタグセット、タグが正しく設定されているか確認してください。CPU ユニットの変数名を確認してください。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）。	オリジネータ側のコネクション設定を確認してください。ターゲットのタグセット、タグが正しく設定されているか確認してください。CPU ユニットの変数名を確認してください。
01	011A	オリジネータ側で発生する異常コード： 高負荷によるバッファ枯渇によりコネクション開設が失敗した。	想定外のネットワーク負荷を受けている可能性があります。Network Configurator デバイスモニタ・Ethernet タブによる帯域使用状況を確認し負荷を見直してください。ネットワーク接続形態でループ接続など、ブロードキャストストームが発生するような箇所があればこれを見直してください。	想定外のネットワーク負荷を受けている可能性があります。Network Configurator デバイスモニタ・Ethernet タブによる帯域使用状況を確認し負荷を見直してください。ネットワーク接続形態でループ接続など、ブロードキャストストームが発生するような箇所があればこれを見直してください。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）
01	011B	ターゲット側が応答した異常コード： 仕様よりも小さな値で RPI が設定された。	本異常発生しません。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）。	オリジネータ側の RPI 設定値を 0.5 ミリ秒以上に設定してください。
01	0203	オリジネータ側で発生する異常コード： コネクションタイムアウトが発生した。	ターゲットからのタグデータリンク通信がタイムアウトしました。ターゲットやスイッチを含む経路上機器の電源やケーブル配線状況を確認してください。高負荷による性能低下の場合は、タイムアウト値を大きくするか、RPI を大きくするなど性能設計を見直してください。 また、ターゲットの CIP メッセージ通信が停止していないか、オリジネータおよび、経路上の機器の Packet Filter 機能で CIP 通信が許可されているかを確認してください。		

コネクション ステータス		異常発生要因	処置方法		
General Status（HEX）	Additional Status（HEX）		構成 1	構成 2	構成 3
01	0204	オリジネータ側で発生する異常コード： コネクション開設処理でタイムアウトが発生した。	ターゲットからの応答がありません。ターゲットやスイッチを含む経路上機器の電源やケーブル配線状況を確認してください。 また、ターゲットまたは、オリジネータの CIP メッセージ通信が停止していないか、ターゲットおよび、経路上の機器の Packet Filter 機能で CIP 通信が許可されているかを確認してください。		
01	0205	ターゲット側が応答した異常コード： コネクション開設用フレームのパラメータ異常。	本異常は発生しません。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）。	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）。
01	0302	オリジネータで発生した、もしくはターゲット側が応答した異常コード： タグデータリンク許容通信帯域幅（pps）がオーバ。	オリジネータとターゲットのコネクション設定（コネクション数、RPI）を確認してください。	ターゲットのコネクション数、RPI を確認してください。オリジネータとターゲットのコネクション設定（コネクション数、RPI）を確認してください。	オリジネータとターゲットのコネクション設定（コネクション数、RPI）を確認してください。
01	0311	ターゲット側が応答した異常コード： コネクション開設用フレームのパラメータ異常。	本異常は発生しません。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）。	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）。
01	0312	ターゲット側が応答した異常コード： コネクション開設用フレームのパラメータ異常。	本異常は発生しません。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）。	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）。
01	0315	ターゲット側が応答した異常コード： コネクション開設用フレームのパラメータ異常。	本異常は発生しません。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）。	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）。
01	0316	ターゲット側が応答した異常コード： コネクションクローズ用フレームのパラメータ異常。	本異常発生しません。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）

コネクション ステータス		異常発生要因	処置方法		
General Status (HEX)	Additional Status (HEX)		構成 1	構成 2	構成 3
01	031C	オリジネータ側で発生するコード： その他の異常。	本異常発生しません。	オープン要求に対し、ターゲット側から規格外のレスポンスコードが応答された場合に、オリジネータがこの要因を発生します。	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）
08	−	ターゲット側が応答した異常コード： ターゲット機器にForward Open、Large Forward Open サービスがない。	本異常発生しません。	ターゲット側の動作仕様に依存する（ターゲット側メーカにお問い合わせください）。	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）
D0	0001	オリジネータ側で発生する異常コード： コネクション動作が停止中	タグデータリンク停止スイッチが ON により停止されたか、設定データのダウンロード中です。タグデータリンク開始スイッチを ON にするか設定データのダウンロードが完了するのを待ってください。 本体停止異常、ユニット故障、などの要因も含まれますので、これらの対処方法は 「15-2-1 EtherNet/IP ユニットで発生する異常と処置」を参照してください。	本エラーコードは各ベンダ独自に定義されるため、ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）
D0	0002	オリジネータ側で発生する異常コード： コネクションを開設（オープン処理）中	オープン処理が完了するまで待ってください。	本エラーコードは各ベンダ独自に定義されるため、ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。	オリジネータ側の動作仕様に依存する（オリジネータ側メーカにお問い合わせください）

コネクション ステータス		異常発生要因	処置方法		
General Status (HEX)	Additional Status (HEX)		構成 1	構成 2	構成 3
オムロン独自のエラーコード					
01	0810	ターゲット側が応答した異常コード： コネクション開設時に CPU ユニットから最新データが取得できない状態。 （コネクション開設リトライで自動復旧する）	コネクション開設時に CPU ユニットのサイクルタイムが長い場合、または PLC 上の何らかの障害でコントローラシステムが停止した場合に発生します。 タスク周期が長い場合は、自動復旧します。コントローラシステム停止の場合は、CPU ユニットの異常情報から異常原因を特定します。	本エラーコードは各ベンダ独自に定義されるため、ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。	本エラーコードは各ベンダ独自に定義されるため、オリジネータ側の動作仕様に依存します（オリジネータ側メーカにお問い合わせください）。
01	0811	オリジネータ側で発生する異常コード： コネクション開設時に CPU ユニットから最新データが取得できない状態。 （コネクション開設リトライで自動復旧する）	コネクション開設時に CPU ユニットのタスク周期が長い場合に発生します。 タスク周期が長い場合は、自動復旧します。	本エラーコードは各ベンダ独自に定義されるため、ターゲット側の動作仕様に依存します（ターゲット側メーカにお問い合わせください）。	本エラーコードは各ベンダ独自に定義されるため、オリジネータ側の動作仕様に依存します（オリジネータ側メーカにお問い合わせください）。

15-4 異常履歴機能

EtherNet/IP ユニットが検出した異常を、異常の発生時刻とともに記録しておく機能です。
記録された結果（異常履歴）は、Network Configurator から読み出したり、Sysmac Studio からクリアできます。

15-4-1 異常履歴データ

異常履歴データの構成

● 異常内容

発生した異常の内容です。

● 詳細情報

発生した異常内容それぞれに対するトラブルシューティングのための詳細情報です。
異常内容、詳細情報は次ページ以降の 🕮「15-4-2 異常履歴コード一覧」を参照してください。

● 時間情報

異常発生日時として、年（西暦下 2 桁）、月、日、時、分、秒が記録されます。

参考

EtherNet/IP ユニットは、CPU ユニットの時間情報を読み出して使用します。
CPU ユニットから時間情報を読み出すことができなかった場合は、異常履歴の時間情報がすべて 0 になります。
内蔵時計の時間設定が正しく行われていない場合は、正しい時間情報が記録されません。この異常履歴を読み出すと時間情報は不定になります。

異常履歴データの格納場所

● RAM の異常履歴データ

異常が発生すると、ユニット内の RAM の異常履歴テーブルに、異常 1 件につき 1 レコード、最大 64 レコード記録されます。

● EEP-ROM の異常履歴データ

とくに重要度の高い異常は、発生時に、（RAM の異常履歴テーブル以外に）ユニット内の EEP-ROM の異常履歴テーブルにも同時に記録されます。その場合、ユニットの電源 OFF やリセット後でも内容が保持されます（EEP-ROM の異常履歴テーブルは、コントローラリセット時に自動的に RAM の異常履歴テーブルに読み出されます）。EEP-ROM へは、異常 1 件につき 1 レコード、最大 32 レコード記録されます。格納データが最大値（RAM では 64 レコード、EEP-ROM では 32 レコード）に達すると、最古のレコードが消去され、最新のレコードが格納されます。

● 異常履歴データのクリア

Sysmac Studio のコントローライベントログ機能により、EtherNet/IP ユニットの RAM や EEP-ROM に格納された異常履歴データをクリアすることができます。
コントローライベントログの機能や操作方法については、🕮『Sysmac Studio Version 1 オペレーションマニュアル』（SBCA-470）を参照してください。

15-4-2 異常履歴コード一覧

異常コード（HEX）	異常内容	詳細情報		EEP-ROM
		1 バイト目	2 バイト目	
0001	CPU ユニット WDT エラー	00 Hex	00 Hex	○
0002	CPU ユニットサービス監視異常	監視時間（ms）		○
0006	その他の CPU ユニット異常	D11 ： 登録 I/O テーブルに該当号機なし その他のビットはシステム予約		○
000E	I/O バス異常	00 Hex	00 Hex	○
000F	CPU ユニット初期処理異常	00 Hex	00 Hex	○
0011	イベントタイムアウト	MRC（メインコマンド）	SRC（サブコマンド）	○
0012	CPU ユニットメモリ異常	01 Hex：読み出しエラー 02 Hex：書き込みエラー	03 Hex：ルーチングテーブル 05 Hex：高機能ユニットエリア（リレー／DM）	○
0015	CPU ユニット停止異常	00 Hex	00 Hex	○
0017	タグデータベース異常	00 Hex	00 Hex	○
0103	再送オーバにより送信不可	CIP フレームの場合：FFFF		×
0105	ノードアドレス設定異常のため送信不可			×
0107	相手局がネットワークに加入していないので送信不可			×
0108	号機アドレスに対応するユニットがないため送信不可			×
010B	CPU ユニット異常のため送信不可			×
010D	ルーチングテーブルに宛先アドレスが未設定のため送信不可			×
010E	ルーチングテーブル未登録のため送信不可			×
010F	ルーチングテーブル異常のため送信不可			×
0110	中継回数オーバのため送信不可			×
0111	コマンドが最大コマンド長を超えているため送信不可			×
0112	ヘッダ異常のため送信不可			×
0117	内部受信バッファフルのためパケット廃棄			×
0118	不正パケット廃棄			×
0119	自ノードビジーのため送信不可			×
0120	予期せぬルーチングエラー			×
0122	現在のモードではサービスを実行できないためパケット廃棄			×
0123	内部送信バッファフルのためパケット廃棄	CIP フレームの場合：FFFF		×
0124	最大フレーム長を超えているためルーチング不可			×
0125	レスポンスタイムアウトによりパケット廃棄			×
020F	通信コントローラ異常	00 Hex	01 Hex	○
0211	IP アドレス重複	ポート番号 （02 固定）	IP アドレスの下位 8 ビット	○

異常コード（HEX）	異常内容	詳細情報		EEP-ROM
		1 バイト目	2 バイト目	
021A	設定テーブル論理異常	00 Hex	02 Hex：ネットワークパラメータ 03 Hex：ルーチングテーブル 04 Hex：システム設定 0E Hex：ユニットネーム 12 Hex：ステータスエリアレイアウト設定異常 13 Hex：ステータスエリアレイアウト設定照合異常 15 Hex：ユニット設定後に異なるシリーズ（NJ ⇔ CJ2 など）の PLC へ装着した	○
0300	パラメータエラーによるパケット廃棄	CIP フレームの場合：FFFF		×
0347	I/O リフレッシュ異常	00Hex	00Hex	○
03C0	FINS/TCP 設定異常	01 Hex ～ 10 Hex：コネクション番号	01： 自動割付 FINS ノードアドレス重複 02： 接続先 IP アドレス異常 03： 接続先ポート No. 異常	×
03C1	サーバ設定異常	00 Hex： DNS 03 Hex： SNTP 04 Hex： FTP 06 Hex： BOOTP 07 Hex： SNMP 08 Hex： SNMP トラップ 09 Hex： FINS/UDP 0A Hex：FINS/TCP	01： IP アドレス 02： ホスト名 03： ポート No. 04： その他のパラメータ	×
03C2	FINS/TCP パケット廃棄	01 Hex ～ 10 Hex：コネクション番号	02 Hex：相手ノードがクローズのため再オープン中 03 Hex：受信異常のため再オープン中 04 Hex：送信異常のため再オープン中 05 Hex：相手ノードから RST を受信したため再オープン中 06 Hex：KeepAlive 応答なしのため再オープン中 07 Hex：FINS/TCP 手順の不正 08 Hex：サーバ処理でメモリ枯渇が発生 09 Hex：クライアント処理でメモリ枯渇が発生 0A Hex：ノード切替え時にメモリ枯渇が発生	×

異常コード（HEX）	異常内容	詳細情報		EEP-ROM
		1 バイト目	2 バイト目	
03C3	FINS/UDP パケット廃棄	00 Hex	01 Hex ～ FE Hex：送信元ノード No.	×
03C4	サーバ接続異常	00 Hex：DNS 03 Hex：SNTP 04 Hex：FTP 06 Hex：BOOTP 07 Hex：SNMP 08 Hex：SNMP トラップ	01 Hex：指定されたホストがない 02 Hex：指定されたホストでサービスなし 03 Hex：タイムアウト 06 Hex：ホスト名解決異常 07 Hex：送信異常 08 Hex：受信異常 09 Hex：その他の異常 0A Hex：取得 IP アドレス異常	×
03C6	時計情報書込みエラー	0001：CPU ユニット異常発生のため時計情報が更新できなかった。	CPU ユニットの異常を解除してください。	×
		0002：時計情報書込み不可の CPU ユニット・動作モードのため時計情報が更新できなかった。	「第 10 章 時計情報自動調整機能」を参考に、使用条件を確認してください。	
03D0	Ethernet 基本設定異常	01 Hex：Ethernet 設定異常	01Hex：サム値異常 11Hex：設定値に矛盾がある 12Hex：指定された通信速度未サポート	○
		02 Hex：TCP/IP 基本設定異常	01Hex：サム値異常 11Hex：IP アドレス不正 12Hex：サブネットマスク不正 13Hex：デフォルトゲートウェイアドレス不正 14Hex：プライマリネームサーバ 15Hex：セカンダリネームサーバ不正 16Hex：ドメインネーム不正 17Hex：ホストネーム不正	
03D1	Ethernet 拡張設定異常	02 Hex：ユニット設定異常	01Hex：サム値異常 10Hex：IP ルータテーブル不正 11Hex：FINS/UDP 設定不正 12Hex：FINS/TCP 設定不正 13Hex：FTP 設定不正 14Hex：SNTP 設定不正 15Hex：SNMP 設定不正 16Hex：SNMP トラップ設定不正	○
03D2	パケット廃棄	01 Hex	00 Hex	×

異常コード（HEX）	異常内容	詳細情報		EEP-ROM
		1 バイト目	2 バイト目	
03D3	リンクオフ異常	00 Hex	00 Hex	×
03D4	照合異常（タグデータリンクのみ） 注：照合異常の要因を特定したい場合、 📖「15-3 コネクションステータスのコード一覧と処置方法」を参照してください。 ターゲットノード不在時における本異常履歴は記録されません。	コネクションインスタンス No.（1 ～ 255）	IP アドレスの下位 8 ビット	×
03D5	タグデータリンク異常	00 Hex	IP アドレスの下位 8 ビット	×
0601	高機能ユニット異常	不定		○
0602	高機能ユニットメモリ異常	01 Hex：読み出しエラー 02 Hex：書き込みエラー	02 Hex：ネットワークパラメータ 06 Hex：異常履歴 09 Hex：Identity 情報 0E Hex：Unit Name 0FHex：Ethernet 基本設定 10Hex：Ethernet 拡張設定 11Hex：MAC アドレス 12 Hex：ステータスエリアレイアウト設定 14 Hex：タグアドレス解決記憶書込エラー	○（*）

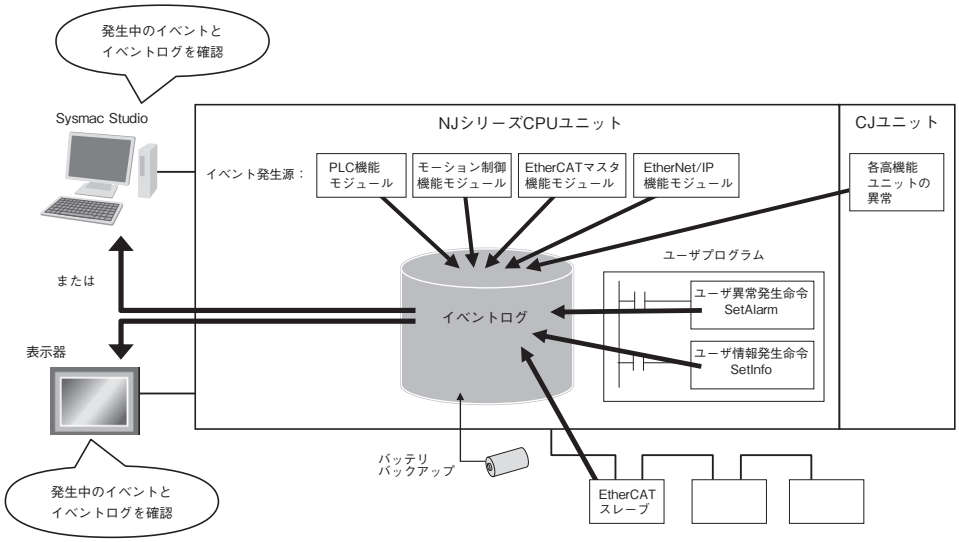
*　異常履歴エリア（EEP-ROM）がメモリ異常の場合は、EEP-ROM への異常登録を行わない。

15-5 イベントログ機能の概要

15-5-1 イベントログ機能の概要

NJ シリーズは、コントローラで発生したイベント（異常と情報）を共通の方法で一元的に管理します。Sysmac Studio または表示器にて、発生中のイベントと、発生したイベント履歴（以降「イベントログ」と呼びます）を確認することができます。
本ユニットで発生するコントローラ異常もNJシリーズCPUユニットにイベントとして通知されます。

NJ シリーズ CPU ユニットのイベントログ機能については、📖『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）、コントローラ異常の詳細、確認方法、対処方法については、📖『NJ/NX シリーズ トラブルシューティングマニュアル』（SBCA-469）を参照してください。



表示器でイベントを確認する場合は、表示器を　CPU　ユニットの内蔵　EtherNet/IP ポートで接続してください。

15-5-2 異常一覧

本ユニットで発生する異常の一覧を以下に示します。重要度の欄では以下のような略称と記号を使用しています。

略称	名称
全	全停止フォールトレベル
部	部分停止フォールトレベル
軽	軽度フォールトレベル
監	監視情報
般	一般情報

記号	意味
○	システムにより定義されている重要度
◎	ユーザが変更可能な重要度*1

*1　変更可能な重要度がある場合にだけ記載します。

イベントコード欄の（ ）内はそのイベントが追加された CPU ユニットのユニットバージョンを示します。

NJ シリーズのすべてのイベントコードについては、🕮『NJ/NX シリーズ トラブルシューティングマニュアル』（SBCA-469）を参照してください。

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
047A0000Hex	ユニットメモリ異常（素子の異常）	EtherNet/IP ユニット内部の不揮発性メモリへの異常履歴またはデバイスパラメータの書き込みで異常が発生しました	• 周囲にノイズ源がある • 不揮発性メモリの故障			○			🕮 P.15-38
047B0000Hex	不揮発性メモリ異常	不揮発性メモリに異常が発生しました	• 周囲にノイズ源がある • 不揮発性メモリの故障			○			🕮 P.15-39
047C0000Hex	通信コントローラ異常	通信コントローラに異常が発生しました	• ノイズ • 通信コントローラのハードウェア故障			○			🕮 P.15-39
14840000Hex	通信パラメータ設定不正	不揮発性メモリに保存されるタグデータリンクに関連するパラメータの妥当性確認で異常と判断されました	• ダウンロード中に通信異常が発生した • 不揮発性メモリの故障 • ダウンロード中に電断が発生した			○			🕮 P.15-40
14850000Hex	タグデータベース異常	タグデータリンク、ステータスレイアウトなどの変数の使用時に、CPU ユニットでタグデータベース異常が発生しました	• ダウンロード中に電断が発生した • ダウンロード中に通信異常が発生した			○			🕮 P.15-40
34A80000Hex	照合異常	タグデータリンクパラメータに登録されている相手ノード情報と実際のノード情報が異なります	• 指定されたターゲットが存在しない • 変数名が不一致 • コネクションサイズが異なる • コネクションリソース不足			○			🕮 P.15-41

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
34A90000Hex	タグデータリンク異常	オリジネータとしてのコネクションに1つ以上の異常があります。 ただし、以下は除きます • ターゲットとしてのコネクション • スイッチングハブとのリンクオフ異常発生によるコネクションタイムアウト	• ターゲットノードの電源OFF • ターゲットノードの通信停止 • EtherNet/IP の Ethernet ケーブルのコネクタのはずれ • EtherNet/IP の Ethernet ケーブルの断線 • ノイズ			○			P.15-41
34AA0000Hex	タグリフレッシュ異常	タグデータリンクでサポート外のエリア種別、アドレス範囲を指定しました	• タグデータリンクでサポート外のエリア種別、アドレス範囲を指定した			○			P.15-42
34AB0000Hex	Ethernet 基本設定異常	TCP/IP 設定が不正です	• ダウンロード中に電断が発生した • ダウンロード中に通信異常が発生した			○			P.15-42
34AC0000Hex	IP アドレステーブル異常	IP アドレステーブルの情報が不正です	• ダウンロード中に電断が発生した • ダウンロード中に通信異常が発生した			○			P.15-42
34AD0000Hex	IP ルータテーブル異常	IP ルータテーブルの情報が不正です	• ダウンロード中に電断が発生した • ダウンロード中に通信異常が発生した			○			P.15-43
34AE0000Hex	ルーチングテーブル異常	ルーチングテーブルの情報が不正です	• ダウンロード中に電断が発生した • ダウンロード中に通信異常が発生した			○			P.15-43
34AF0000Hex	Ethernet 拡張設定異常	FINS 設定が不正です	• ダウンロード中に電断が発生した • ダウンロード中に通信異常が発生した			○			P.15-44
34B00000Hex	アドレス不一致	自 IP アドレスのホスト ID と FINS ノードアドレスが一致していません。または、自 IP アドレスの最終アドレスと、ノードアドレス設定スイッチの値が一致していません	• IP アドレス変換方式が自動生成方式に設定されているが、自 IP アドレスのホスト ID と FINS ノードアドレスが一致していない、または、自 IP アドレスの最終アドレスと、ノードアドレス設定スイッチの値が一致していない			○			P.15-44
381C0000Hex	ステータスエリア レイアウト設定異常	EtherNet/IP ユニットのレイアウト設定に異常が発生しています	• EtherNet/IP ユニットのレイアウト設定が異常			○			P.15-45
54AE0000Hex	複数（ソフト）スイッチ同時 ON 異常	複数のソフトスイッチが同時に TRUE になりました	• 複数のソフトスイッチが同時に TRUE になった。または 1 つのソフトスイッチ処理が完了しないうちに、他のスイッチが TRUE にされた			○	◎		P.15-45

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
84E00000Hex	IP アドレス重複異常	IP アドレスが重複しています	• EtherNet/IP ポートの IP アドレスが他ノードの IP アドレスと重複している			○			P.15-46
84E10000Hex	BOOTP サーバ接続異常	BOOTP サーバとの接続に失敗しました	• サーバ設定の誤り（取得した IP アドレスが不正） • サーバダウン • 通信経路の異常発生			○			P.15-46
54AF0000Hex	変数範囲外アクセス検知	タグデータリンクに使用しているタグ変数に対して、範囲外の値のアクセスを検知しました	• 範囲指定した変数に対して、EtherNet/IP タグデータリンクにて範囲外の値を書き込んだ。または、列挙型の変数に対して、EtherNet/IP タグデータリンクにて列挙子以外の値を書き込んだ				○		P.15-47
84E20000Hex	リンクオフ異常	Ethernet のリンクステータスの立ち下りを検知しました	• Ethernet ケーブルが未接続 • Ethernet ケーブルの断線、ゆるみ • スイッチングハブの電源 OFF • 通信速度不一致 • ノイズ			◎	○		P.15-47

15-5-3　異常の内容

ここでは、各異常の内容を説明します。

コントローラ異常説明の見方

各異常の説明で使用する表の各項目の意味を［　］内に示します。

<table>
<tr><td>イベント名</td><td colspan="2">［異常の名称です］</td><td colspan="2">イベントコード</td><td colspan="2">［異常のコードです］</td></tr>
<tr><td>内容</td><td colspan="6">［異常の内容です］</td></tr>
<tr><td>発生源</td><td colspan="2">［異常が発生した箇所です］</td><td>発生源詳細</td><td>［発生源の詳細です］</td><td>検出タイミング</td><td>［異常検出のタイミングです］</td></tr>
<tr><td>異常の属性</td><td>重要度</td><td>［制御に対する影響の大きさです］（*1）</td><td>復旧方法</td><td>［復旧方法です］（*2）</td><td>ログカテゴリ</td><td>［保存されるログの種類です］（*3）</td></tr>
<tr><td>発生の影響</td><td>ユーザプログラム</td><td>［ユーザプログラムの実行状態です］（*4）</td><td>動作</td><td colspan="3">［異常（イベント）発生時の動作に関する特記事項です］</td></tr>
<tr><td rowspan="2">システム定義変数</td><td colspan="2">変数名</td><td colspan="2">データ型</td><td colspan="2">名称</td></tr>
<tr><td colspan="6">［異常を検知するシステム定義変数、異常の影響を受けるシステム定義変数、異常の原因となるシステム定義変数の変数名とデータ型、名称です］</td></tr>
<tr><td rowspan="2">発生要因とその処置</td><td colspan="2">発生要因（推定原因）</td><td colspan="2">処置対策</td><td colspan="2">再発防止</td></tr>
<tr><td colspan="6">［異常の発生要因とその処置対策、再発防止方法です］</td></tr>
<tr><td>付属情報</td><td colspan="6">［Sysmac Studio ／表示器で表示される付属情報内容です］</td></tr>
<tr><td>注意点／備考</td><td colspan="6">［その他の注意点、制約事項、補足説明などです。ユーザが選択可能な重要度がある場合は、選択可能な重要度、復旧方法、動作などの関連事項を記載します］</td></tr>
</table>

*1 次のいずれかです
　全停止フォールト：全停止フォールトレベル
　部分停止フォールト：部分停止フォールトレベル
　軽度フォールト：軽度フォールトレベル
　監視情報
　一般情報

*2 次のいずれかです
　自動復旧：要因が除去されたとき自動的に正常に戻ります
　異常解除：要因を除去後、異常解除を実行することで正常に戻ります
　電源再投入：要因を除去後、コントローラの電源再投入することで正常に戻ります
　コントローラリセット：要因を除去後、コントローラリセットすることで正常に戻ります
　発生要因依存：発生要因に依存します

*3 次のいずれかです
　システム：システムイベントログ
　アクセス：アクセスイベントログ

*4 次のいずれかです
　継続：ユーザプログラムの実行を継続します
　停止：ユーザプログラムの実行を停止します
　開始：ユーザプログラムの実行を開始します

異常の説明

イベント名	ユニットメモリ異常（素子の異常）			**イベントコード**	047A0000Hex	
内容	EtherNet/IP ユニット内部の不揮発性メモリへの異常履歴またはデバイスパラメータの書き込みで異常が発生しました					
発生源	PLC 機能モジュール		**発生源詳細**	CJ ユニット	**検出タイミング**	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	**重要度**	軽度フォールト	**復旧方法**	異常解除	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	不揮発性メモリの異常の種類により以下の動作となります。（EtherNet/IP ユニットの異常履歴、CIO ステータスで確認可能） 02 Hex：ネットワークパラメータ →タグデータリンク停止 06 Hex：異常履歴 →異常履歴を不揮発性メモリに保存をしない 09 Hex：Identity 情報 → EtherNet/IP 通信停止 0E Hex：Unit Name　→その他通信への影響無し 0F Hex：Ethernet 基本設定　→通信不可 10 Hex：Ethernet 拡張設定　→該当機能停止 11 Hex：MAC アドレス　→ Ethernet 通信停止 12 Hex：ステータスエリアレイアウト設定 →レイアウト設定は「デフォルト」で動作 14 Hex：タグアドレス解決記憶書込エラー →タグデータリンク通信停止		

システム定義変数	**変数名**	**データ型**	**名称**
	なし	−	−

発生要因とその処置	**発生要因（推定原因）**	**処置対策**	**再発防止**
	周囲にノイズ源がある	デバイスパラメータの書き込み失敗の場合は、Network Configurator や Sysmac Studio でデバイスパラメータまたはコネクション設定をダウンロードしてください。その後、電源再投入または EtherNet/IP ユニットのリスタートを行っても異常が解除されない場合は、周囲にノイズ源がないか確認し、ノイズ源がある場合にはノイズ源を除去してください	周囲にノイズ源がない環境で使用してください。 ノイズ対策を行ってください
	不揮発性メモリの故障	上記の処置を行っても、異常が再発する場合は、該当 EtherNet/IP ユニットを交換してください	なし

付属情報	なし
注意点／備考	なし

イベント名	不揮発性メモリ異常			イベントコード	047B0000Hex	
内容	不揮発性メモリに異常が発生しました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	常時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	電源投入時の自己チェックで検知した場合は、該当ユニット停止。運転中に発生した場合は、継続運転		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	周囲にノイズ源がある		周囲にノイズ源がある場合にはノイズ源を除去してください		周囲にノイズ源がない環境で使用してください。 ノイズ対策を行ってください	
	不揮発性メモリの故障		上記の要因でない場合は、該当ユニットを交換してください		なし	
付属情報	なし					
注意点／備考	なし					

イベント名	通信コントローラ異常			イベントコード	047C0000Hex	
内容	通信コントローラに異常が発生しました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	リンク確立以降
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットはオフライン状態となり、通信動作不可。すべての通信要求にエラーを返します。CPU ユニットとのデータ交換動作（リフレッシュ）は継続		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	ノイズ		コントローラを電源再投入してください。ノイズ対策を行ってください		周囲にノイズ源がない環境で使用してください。 ノイズ対策を行ってください	
	通信コントローラのハードウェア故障		上記の処置を行っても再発する場合は、該当 EtherNet/IP ユニットを交換してください		なし	
付属情報	なし					
注意点／備考	なし					

イベント名	通信パラメータ設定不正			イベントコード	14840000Hex	
内容	不揮発性メモリに保存されるタグデータリンクに関連するパラメータの妥当性確認で異常と判断されました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットのタグデータリンク機能使用不可		

システム定義変数	変数名	データ型	名称
	なし	−	−

発生要因とその処置	発生要因（推定原因）	処置対策	再発防止
	ダウンロード中に電断が発生した	Network Configurator や Sysmac Studio から再度デバイスパラメータまたはコネクション設定をダウンロードしてください	ダウンロード中に電断を行わないようにしてください
	ダウンロード中に通信異常が発生した	周囲にノイズ源がないか確認し、ノイズ源がある場合にはノイズ源を除去してください	周囲にノイズ源がない環境で使用してください。 ノイズ対策を行ってください
	不揮発性メモリの故障	上記の処置を行っても、異常が再発する場合は、該当 EtherNet/IP ユニットを交換してください	なし

付属情報	なし
注意点／備考	なし

イベント名	タグデータベース異常			イベントコード	14850000Hex	
内容	タグデータリンク、ステータスレイアウトなどの変数の使用時に、CPU ユニットでタグデータベース異常が発生しました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットのタグデータリンク機能使用不可		

システム定義変数	変数名	データ型	名称
	なし	−	−

発生要因とその処置	発生要因（推定原因）	処置対策	再発防止
	ダウンロード中に電断が発生した	Sysmac Studio から再度設定をダウンロードしてください	ダウンロード中に電断を行わないようにしてください
	ダウンロード中に通信異常が発生した	ノイズが多い場合は、ノイズ対策を行ってください	ノイズ対策を行ってください

付属情報	なし
注意点／備考	なし

イベント名	照合異常			イベントコード	34A80000Hex	
内容	タグデータリンクパラメータに登録されている相手ノード情報と実際のノード情報が異なります					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	リンク確立以降
異常の属性	重要度	軽度フォールト	復旧方法	自動復旧	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	ターゲットに対して定期的に再コネクション処理を続行		

システム定義変数	変数名	データ型	名称
	なし	−	−

発生要因とその処置	発生要因（推定原因）	処置対策	再発防止
	指定されたターゲットが存在しない	コネクション登録されているノードの電源が入っていない場合は電源を入れてください。ケーブルの接続、ケーブルの断線、ゆるみがある場合は修正してください。また、ノイズが多い場合はノイズ対策を行ってください	接続ノードの電源が入っていることを確認してください。ケーブルを確実に接続してください。また、ノイズが多い場合はノイズ対策を行ってください
	変数名が不一致 コネクションサイズが異なる	データリンクパラメータと実際のノード情報が一致するようにデータリンクパラメータを修正または接続するノードを変更してください	データリンクパラメータと実際のノード情報が一致するようにしてください
	コネクションリソース不足	Class3 メッセージの使用数を減らしてください	データリンク、Class3 メッセージの使用数を減らしてください

付属情報	なし
注意点／備考	異常履歴の詳細情報で、ターゲットノードを特定してください

イベント名	タグデータリンク異常			イベントコード	34A90000Hex	
内容	オリジネータとしてのコネクションに 1 つ以上の異常があります。 ただし、以下は除きます • ターゲットとしてのコネクション • スイッチングハブとのリンクオフ異常発生によるコネクションタイムアウト					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	リンク確立以降
異常の属性	重要度	軽度フォールト	復旧方法	自動復旧	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	対象となるデータリンクコネクションの停止。タグデータリンク異常のターゲットに対して定期的に再コネクション処理を続行		

システム定義変数	変数名	データ型	名称
	なし	−	−

発生要因とその処置	発生要因（推定原因）	処置対策	再発防止
	ターゲットノードの電源 OFF	ターゲットノードの状態を確認し、正常に起動してください	ターゲットノードが正常に起動していることを確認しておきます
	ターゲットノードの通信停止		
	EtherNet/IP の Ethernet ケーブルのコネクタのはずれ	コネクタが正しく嵌合しているかを確認し、再接続してください	コネクタを確実に接続します
	EtherNet/IP の Ethernet ケーブルの断線	Ethernet ケーブルを交換してください	なし
	ノイズ	ノイズが多い場合はノイズ対策を行ってください	ノイズが多い場合はノイズ対策を行ってください

付属情報	なし
注意点／備考	なし

イベント名	タグリフレッシュ異常			イベントコード	34AA0000Hex	
内容	タグデータリンクでサポート外のエリア種別、アドレス範囲を指定しました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	常時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットのタグデータリンク機能使用不可		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	タグデータリンクでサポート外のエリア種別、アドレス範囲を指定した		タグデータリンクで指定するエリア種別、アドレス範囲を指定可能範囲になるように修正してください		タグデータリンクで指定するエリア種別、アドレス範囲が指定可能範囲になるようにしてください	
付属情報	なし					
注意点／備考	なし					

イベント名	Ethernet 基本設定異常			イベントコード	34AB0000Hex	
内容	TCP/IP 設定が不正です					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットの Ethernet での通信不可		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	ダウンロード中に電断が発生した		Sysmac Studio から再度設定をダウンロードしてください		ダウンロード中に電断を行わないようにしてください	
	ダウンロード中に通信異常が発生した		ノイズが多い場合は、ノイズ対策を行ってください		ノイズ対策を行ってください	
付属情報	なし					
注意点／備考	なし					

イベント名	IP アドレステーブル異常			イベントコード	34AC0000Hex	
内容	IP アドレステーブルの情報が不正です					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットの Ethernet での通信不可		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	ダウンロード中に電断が発生した		Sysmac Studio から再度設定をダウンロードしてください		ダウンロード中に電断を行わないようにしてください	
	ダウンロード中に通信異常が発生した		ノイズが多い場合は、ノイズ対策を行ってください		ノイズ対策を行ってください	
付属情報	なし					
注意点／備考	なし					

イベント名	IP ルータテーブル異常			イベントコード	34AD0000Hex	
内容	IP ルータテーブルの情報が不正です					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットの Ethernet での通信不可		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	ダウンロード中に電断が発生した		Sysmac Studio から再度設定をダウンロードしてください		ダウンロード中に電断を行わないようにしてください	
	ダウンロード中に通信異常が発生した		ノイズが多い場合は、ノイズ対策を行ってください		ノイズ対策を行ってください	
付属情報	なし					
注意点／備考	なし					

イベント名	ルーチングテーブル異常			イベントコード	34AE0000Hex	
内容	ルーチングテーブルの情報が不正です					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットの Ethernet での通信不可		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	ダウンロード中に電断が発生した		Sysmac Studio から再度設定をダウンロードしてください		ダウンロード中に電断を行わないようにしてください	
	ダウンロード中に通信異常が発生した		ノイズが多い場合は、ノイズ対策を行ってください		ノイズ対策を行ってください	
付属情報	なし					
注意点／備考	なし					

イベント名	Ethernet 拡張設定異常			イベントコード	34AF0000Hex	
内容	FINS 設定が不正です					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当する機能停止		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	ダウンロード中に電断が発生した		EtherNet/IP ユニットの異常履歴詳細情報から異常要因を特定し、正しい設定内容に修正してください		ダウンロード中に電断を行わないようにしてください	
	ダウンロード中に通信異常が発生した		ノイズが多い場合は、ノイズ対策を行ってください		ノイズ対策を行ってください	
付属情報	なし					
注意点／備考	なし					

イベント名	アドレス不一致			イベントコード	34B00000Hex	
内容	自 IP アドレスのホスト ID と FINS ノードアドレスが一致していません。または、自 IP アドレスの最終アドレスと、ノードアドレス設定スイッチの値が一致していません					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	FINS 通信不可		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	IP アドレス変換方式が自動生成方式に設定されているが、自 IP アドレスのホスト ID と FINS ノードアドレスが一致していない、または、自 IP アドレスの最終アドレスと、ノードアドレス設定スイッチの値が一致していない		IP アドレスまたはノードアドレス設定を一致させてください		なし	
付属情報	なし					
注意点／備考	なし					

イベント名	ステータスエリア レイアウト設定異常			イベントコード	381C0000Hex	
内容	EtherNet/IP ユニットのレイアウト設定に異常が発生しています					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	電源投入時／コントローラリセット時／ユニットリスタート時
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	本異常発生中は、レイアウト設定は「デフォルト」で動作。 ただし、以下の場合はレイアウト設定は「ユーザ設定」で動作 • 存在しないエリアを指定したとき		

システム定義変数	変数名	データ型	名称
	なし	−	−

発生要因とその処置	発生要因（推定原因）	処置対策	再発防止
	EtherNet/IP ユニットのレイアウト設定が異常	EtherNet/IP ユニットのレイアウト設定を修正してください	EtherNet/IP ユニットのレイアウト設定に間違いがないことを確認してください

付属情報	なし
注意点／備考	なし

イベント名	複数（ソフト）スイッチ同時 ON 異常			イベントコード	54AE0000Hex	
内容	複数のソフトスイッチが同時に TRUE になりました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	ソフトスイッチ操作時
異常の属性	重要度	軽度フォールト	復旧方法	自動復旧	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	影響なし		

システム定義変数	変数名	データ型	名称
	なし	−	−

発生要因とその処置	発生要因（推定原因）	処置対策	再発防止
	複数のソフトスイッチが同時に TRUE になった。または 1 つのソフトスイッチ処理が完了しないうちに、他のスイッチが TRUE にされた	複数のソフトスイッチが同時に TRUE にならないようにしてください	複数のソフトスイッチを同時に TRUE にしないようにしてください

付属情報	なし
注意点／備考	重要度を監視情報に変更できます。監視情報に変更した場合、復旧は不要となります

イベント名	IP アドレス重複異常			イベントコード	84E00000Hex	
内容	IP アドレスが重複しています					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	IP アドレス確定以降
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	EtherNet/IP 通信動作不可。すべての通信要求にエラーを返します。 PLC 機能モジュールとのデータ交換動作（リフレッシュ）は継続		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	EtherNet/IP ポートの IP アドレスが他ノードの IP アドレスと重複している		次のいずれかの処置をした後、コントローラを電源再投入またはコントローラリセットしてください • 他ノードの IP アドレスを確認し、重複しないように IP アドレスの設定を修正する • IP アドレスが重複しているノードをネットワークから離脱させる		ネットワーク内で、各ノードの IP アドレスが互いに重複しないように割り当ててください	
付属情報	なし					
注意点／備考	なし					

イベント名	BOOTP サーバ接続異常			イベントコード	84E10000Hex	
内容	BOOTP サーバとの接続に失敗しました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	リンク確立以降
異常の属性	重要度	軽度フォールト	復旧方法	異常解除	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	BOOTP サーバから応答があるまで BOOTP サーバへ要求継続。この間、BOOTP を使用した通信動作不可。すべての通信要求にエラーを返します。 PLC 機能モジュールとのデータ交換動作（リフレッシュ）は継続。 BOOTP サーバにより EtherNet/IP ポートに設定されようとした IP アドレスは不正値デフォルトの IP アドレス（192.168.250. ノードアドレス）で動作		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	サーバ設定の誤り（取得した IP アドレスが不正）		サーバに設定した EtherNet/IP の IP アドレスが不正値の場合は、修正してください		サーバに設定した EtherNet/IP の IP アドレスが不正値にならないように設定してください	
	サーバダウン		接続先のサーバが正常に稼働しているか確認し、正常に稼働していない場合は正しく稼働させてください		接続先のサーバが正常に稼働していることを確認してください	
	通信経路の異常発生		サーバへの通信経路に問題がないかを調査して、問題がある場合には対策を行ってください		なし	
付属情報	なし					
注意点／備考	なし					

イベント名	変数範囲外アクセス検知			イベントコード	54AF0000Hex	
内容	タグデータリンクに使用しているタグ変数に対して、範囲外の値のアクセスを検知しました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	変数書込時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	範囲指定した変数に対して、EtherNet/IP タグデータリンクにて範囲外の値を書き込んだ。または、列挙型の変数に対して、EtherNet/IP タグデータリンクにて列挙子以外の値を書き込んだ		範囲指定した変数に対して、範囲内の値を書き込むように書き込み値を修正してください。または、列挙型の変数に対して、列挙子の値を書き込むように書き込み値を修正してください		範囲指定した変数に対して、範囲内の値を書き込むようにしてください。または、列挙型の変数に対して、列挙子の値を書き込むようにしてください	
付属情報	なし					
注意点／備考	• 範囲外の値、または列挙子以外の値の書き込みは正常終了していません • 範囲内の値、または列挙子の値の書き込みは正常終了しています					

イベント名	リンクオフ異常			イベントコード	84E20000Hex	
内容	Ethernet のリンクステータスの立ち下りを検知しました					
発生源	PLC 機能モジュール		発生源詳細	CJ ユニット	検出タイミング	リンク確立以降
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当 EtherNet/IP ユニットはオフライン状態となり、通信動作不可。すべての通信要求にエラーを返します。CPU ユニットとのデータ交換動作（リフレッシュ）は継続		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	Ethernet ケーブルが未接続		Ethernet ケーブルを接続してください		Ethernet ケーブルを確実に接続してください。また、使用するケーブルが断線していないことを確認してください	
	Ethernet ケーブルの断線、ゆるみ		Ethernet ケーブルを確実に接続してください。ケーブルが断線している場合にはケーブルを交換してください			
	スイッチングハブの電源 OFF		スイッチングハブの電源を ON にしてください。スイッチングハブが故障している場合には、スイッチングハブを交換してください		スイッチングハブの電源を切らないでください	
	通信速度不一致		通信相手ノードと同じ通信速度になるように設定を修正してください		通信相手ノードと同じ通信速度の設定を行ってください	
	ノイズ		ノイズが多い場合は、ノイズ対策を行ってください		ノイズ対策を行ってください	
付属情報	なし					
注意点／備考	重要度を軽度フォールトに変更できます。軽度フォールトに変更した場合の復旧方法は「自動復旧」となります					

15-6 トラブルシューティング

15-6-1 EtherNet/IP ユニットの装着で CPU ユニットの ERROR が点灯／点滅する場合

I/O 照合異常が発生している。	• ユニットが正しく装着されているか確認してください。 • ユニット構成と実機のユニット構成の照合を行い、修正してください。
CPU 高機能ユニット設定異常が発生している。	• ユニット構成に登録されているユニットの機種と、実装されているユニット構成の機種が異なっています。ユニット構成と実機のユニット構成の照合を行い、修正してください。
CPU 高機能異常が発生している。	• ユニットが正しく装着されているか確認してください。 • ユニットをリスタートしてください。 リスタートしても復旧しない場合は、ユニットを交換してください。
I/O バス異常が発生している。	• ユニットが正しく装着されているか確認してください。 • ユニットをリスタートしてください。 • リスタートしても復旧しない場合は、ユニットを交換してください。

詳細は、CPU ユニットのユーザーズマニュアルに従って処置してください。

15-6-2 Ethernet の一般的なトラブル

EtherNet/IP ユニットの 100M、10M LED がともに消灯している。	• 正しい規格のケーブルを使用しているか確認してください。 • ケーブルが正しくスイッチングハブと接続されているか、スイッチングハブの電源が ON されているか確認してください。（7 セグメント LED の異常コードが E1 となります。） • 設定機能があるスイッチングハブの場合、Ethernet のリンク設定が EtherNet/IP ユニットの設定と同じになっているか確認してください。（📖「2-4 ネットワークの布設について」を参照してください。）
EtherNet/IP ユニットの NS LED が赤点灯している。	• 他のノードとの IP アドレス重複を確認してください。（7 セグメント LED の異常コードが F0 となります。）

15-6-3　タグデータリンクがスタートしない場合 *

*　通信ステータス 1 の「データリンク通信中」が TRUE にならない場合を指します。

EtherNet/IP ユニットの LED がすべて消灯している。	• PLC の電源が投入されていることを確認してください。 • EtherNet/IP ユニットが CPU ラックに正しく装着されていることを確認してください。 • CPU ユニットで WDT 異常が発生している場合は、CPU ユニットのマニュアルを参照して処置してください。 • EtherNet/IP ユニットが CPU 高機能異常となった場合は、LED がすべて消灯します。 • EtherNet/IP ユニットをリスタートしてください。 リスタートしても復旧しない場合は、ユニットを交換してください。
EtherNet/IP ユニットの MS LED は緑色点灯したが、NS LED が消灯したまま変化しない。	• 7 セグメント LED に異常コードが表示される場合は、🕮「15-2 LED 表示の意味と異常時の処置」を参照してください。 • ケーブルが正しくスイッチングハブと接続されているか、スイッチングハブの電源が ON されているか確認してください。
EtherNet/IP ユニットの MS LED が緑色点灯した後、NS LED が緑色点滅のまま変化しない。	• 7 セグメント LED に異常コードが表示される場合は、🕮「15-2 LED 表示の意味と異常時の処置」を参照してください。 • ユニット内にタグデータリンク設定がない場合に NS LED が緑色点滅します。Network Configurator を用いて EtherNet/IP ユニットにタグデータリンクを設定後にリスタートしてください。
EtherNet/IP ユニットの MS LED が緑色点灯した後、NS LED が赤色点滅のまま変化しない。	• 7 セグメント LED の異常コードから 🕮「15-2 LED 表示の意味と異常時の処置」を参照して異常要因を取り除いてください。

15-6-4　タグデータリンクのトラブル

タグデータの同時性がない。	以下の点をご理解の上、アプリケーションプログラムを作成してください。 • CPU ユニットと EtherNet/IP ユニットの間では、コネクション単位のデータ同時性を保証しています。 タグデータリンクでのデータの同期性を確保するためには、タグを割り付けるネットワーク変数に、それぞれ更新タスクを設定してください。詳細は、🕮「7-1-7 タグデータリンクするデータの同期性について」を参照してください。 • 他社製品については、各メーカのマニュアルを参照してください。
立ち上げ時など、受信データが意図せず OFF となる。	• ラダープログラムで受信データを使用する場合、通信ステータス 1 の全タグデータリンク通信中フラグか、ターゲットノード PLC であるコントローラの本体運転情報の ON 条件で行ってください。PLC 本体運転情報を使用する場合は、送信受信側の両方のタグセットにコントローラステータスを使用する必要があります。 • 出力（送信）のタグ設定で負荷遮断を有効とした場合、CPU ユニット停止異常状態のときに出力（送信）データが OFF されます。出力（送信）側の PLC 状態を確認してください。
タグデータリンクが通信したり途切れたりする。	• 10Mbps の設定、もしくは 10M リピータハブ、100M リピータハブを使用していないか確認してください。タグデータリンク性能は、スイッチングハブの使用を前提に、オートネゴシエーションによる全二重 100Mbps で 12000pps[*1] の帯域性能を発揮する仕様です。 • 🕮「15-1 Network Configurator によるステータスの確認方法」を参照し、［Ethernet 情報］タブで各種エラーカウンタや廃棄パケットカウンタを参考にして、通信路でのノイズ、規格外ケーブルの使用、ケーブル／コネクタの破損、コネクタの緩み、また、想定外の異常な通信負荷、スイッチングハブ間を異常なループ配線としていないかなどを確認してください。 • 通信経路中のスイッチングハブの転送能力に問題がないかスイッチングハブメーカにお問合せください。スイッチングハブをカスケード接続していくと、途中のスイッチングハブで負荷が集中している場合があります。なお EtherNet/IP ユニット単体では、メッセージ通信よりも優先処理させているため、タグデータリンク性能のみで 12000pps の使用帯域を確保する仕様となっています。 • その他は、🕮「15-1 Network Configurator によるステータスの確認方法」を参照し、［コネクション］タブでコネクションステータスから、🕮「15-3 コネクションステータスのコード一覧と処置方法」を参照して異常要因を取り除いてください。

*1. ユニットバージョン Ver.2.1 以前の場合は、6000pps となります。

15-6-5　メッセージタイムアウトのトラブル

メッセージサービス（CIP UCMM、CIP クラス 3）で、タイムアウトエラーを頻発する。	• タグデータリンクが高負荷の場合、CPU ユニットのタスク周期が比較的大きい場合、複数ノードからメッセージが集中した場合などで、メッセージサービスの応答が遅くなったり、場合によってはメッセージが破棄されることがあります。 • この場合は、タグデータリンクの RPI を大きく（遅く）する、メッセージの負荷を少なくする（または、タイムアウト値を大きくする）などの通信負荷を低減する必要があります。 • タグデータリンク通信負荷は、🕮「15-1 Network Configurator によるステータスの確認方法」を参照し、［Ethernet 情報］タブで帯域使用量を確認することができます。 • 通信負荷の集中によるメッセージの破棄（メモリ枯渇）についての異常履歴コードは 0117、0119、0123、0125、03C2（詳細コード **08、**09、**0A）、03C3、03D2 があります。異常履歴は、🕮「15-1 Network Configurator によるステータスの確認方法」を参照し、［異常履歴］タブで異常履歴を読み出し確認します。

15-7 清掃と点検

ここでは、日常の機器のメンテナンスとして、清掃方法と点検方法について説明します。

15-7-1 清掃方法

EtherNet/IP ユニットを最良の状態で使用するために、次のように定期的に清掃を行ってください。

- 日常の清掃時には、乾いたやわらかい布で乾拭きしてください。
- 乾拭きでも汚れが落ちないときは、布を十分に薄めた中性洗剤（2%）で湿らせて、固く絞ってから拭いてください。
- ユニットにゴムや、ビニール製品、テープなどを長時間付着させておくと、シミが付くことがあります。付着している場合は、清掃時に取り除いてください。

使用上の注意

ベンジンや、シンナーなどの揮発性の溶剤や、化学雑巾などは、絶対に使用しないでください。

15-7-2 点検方法

最良の状態でご使用いただくためにも、定期点検を欠かさず行ってください。
点検は、通常は 6 カ月～ 1 年に 1 回の間隔で実施してください。ただし、極端に高温多湿の環境やほこりの多い環境などで使用する場合は、点検間隔を短くしてください。

点検に必要な器材

点検をする前に次の器材を準備します。

● 日常に必要なもの

- プラスドライバ、マイナスドライバ
- テスタ（またはデジタルボルトメータ）
- 工業用アルコールと純綿布

● 場合により必要なもの

- シンクロスコープ
- ペン書きオシロスコープ
- 温度計、湿度計

点検項目

次の項目について、判定基準から外れていないかどうかを点検します。判定基準からはずれているときは、基準内に入るように周囲の環境を改善するか、本体を調整してください。

点検項目	点検内容	判定基準	点検手段
環境状態	周囲および盤内温度は適当か	0 ～ 55 ℃	温度計
	周囲および盤内湿度は適当か	10 ～ 90%RH （結露・氷結のないこと）	湿度計
	ほこりが積もっていないか	ほこりのないこと	目視
取り付け状態	ユニットはしっかり固定されているか	ゆるみのないこと	
	Ethernet ケーブルのコネクタは完全に挿入されているか、ロックされているか	ゆるみのないこと	プラスドライバ

15-8 装置交換時の留意点

EtherNet/IP ユニットが故障した場合は、通信相手機器の動作に影響を及ぼすことがありますので、速やかに修復作業を行ってください。
機能の修復をできるだけ早く行うために、交換用に予備の EtherNet/IP ユニットを用意されることをお勧めします。

15-8-1 ユニット交換時のお願い

ユニットの交換を行うときは、次の点に注意してください。
- ユニットの交換は、必ずコントローラの電源が OFF の状態で行ってください。
- 交換後、新しいユニットにも異常がないかどうかを確認してください。
- 不良のユニットを修復のために発送していただく場合は、不良の内容についてできるだけ詳細に記載した用紙をユニットに添付して、巻末掲載の弊社支店または営業所へお送りください。

接触不良の場合は、接点をきれいな純綿布に工業用アルコールを染み込ませたもので拭いてください。その後、布のくずを取り除いてからユニットを装着してください。

使用上の注意

ユニット交換時は、感電防止のため、必ずネットワークを停止し、すべてのノードの電源を OFF にしてから作業を行ってください。

15-8-2 ユニット交換後の設定

EtherNet/IP ユニット交換後は、配線や以下の設定を交換前のユニットと同じ状態にしてください。
- ノードアドレス、ユニット番号
  ユニット前面のロータリスイッチの状態を、交換前のユニットに合わせてください。
- コンフィグレーションデータ（パラメータ設定）
  EtherNet/IP ユニット内部の不揮発性メモリに、記録されています。
  よって、各周辺ツールにより、交換前のデータをパソコンに転送しておくか、再設定してください。
  - ユニット設定（自 IP アドレスなど）
    Sysmac Studio を使って、転送・再設定します。
    🕮第 5 章 Sysmac Studio による EtherNet/IP ユニットのユニット設定一覧
  - パラメータ設定（タグデータリンクのパラメータなど）
    Network Configurator を使って、転送・再設定します。
    🕮第 7 章 タグデータリンク機能

使用上の注意

CPU ユニットを交換した場合は、Sysmac Studio で交換前のデータを転送してから、運転を開始してください。保持属性を持つ変数の状態とプログラムの関係によっては、思わぬ事故を招く場合があります。

15-8-3 ユニットの交換方法

1 **交換対象のEtherNet/IPユニットを装着するコントローラ、およびネットワーク接続している外部機器すべての電源を、OFF します。**

2 **交換対象の EtherNet/IP ユニットに接続されている Ethernet ケーブルと、ユニットを外します。**

3 **新しい EtherNet/IP ユニットの以下のハードウェアスイッチを、交換前と同じ内容に設定して、取り付けます。**

- ノードアドレス・ユニット番号設定スイッチ

4 **ユニットを交換した CPU ユニット、およびネットワーク接続している外部機器すべての電源を ON し、システムを起動します。**

5 **EtherNet/IP ユニットの LED 表示や、ステータス表示を確認し、システムが正しく動作しているか確認します。**

A

付録

A-1 NJ シリーズ CPU ユニットと他シリーズとの EtherNet/IP 機能の比較 A-3

A-2 Sysmac Studio でタグデータリンク（EtherNet/IP コネクション）を設定する ... A-4
A-2-1 Sysmac Studio によるタグデータリンク（EtherNet/IP コネクション）設定の概要 ... A-4
A-2-2 Sysmac Studio による EtherNet/IP コネクション設定の手順 A-5
A-2-3 EtherNet/IP コネクション設定機能について A-6
A-2-4 Sysmac Studio による EtherNet/IP コネクション設定 A-11
A-2-5 ネットワークの通信ステータス確認と処置方法 A-32
A-2-6 トラブルシューティング A-36

A-3 EDS ファイルの管理 A-41
A-3-1 EDS ファイルのインストール A-42
A-3-2 EDS ファイルの作成 A-42
A-3-3 EDS ファイルの削除 A-43
A-3-4 EDS ファイルの保存 A-43
A-3-5 EDS ファイルの検索 A-44
A-3-6 EDS ファイルのプロパティ表示 A-45
A-3-7 EDS インデックスファイルの再作成 A-45

A-4 Network Configurator を、WindowsXP/ WindowsVista/Windows7 で使用する際の留意点 A-46
A-4-1 Windows ファイアウォールの設定の変更手順 A-46

A-5 変数のメモリ確保の方法について A-49
A-5-1 変数のメモリ確保の規則 A-49
A-5-2 留意するべきケース例 A-58

A-6 CPU ユニットの外部出力にアクセスする場合の留意事項 A-61

A-7 接続する CPU ユニット（NJ/CJ シリーズ）による相違点 A-62
A-7-1 機能の相違点 A-62
A-7-2 ユーザプログラムからのアクセス方法に関する相違点 A-62

A-8 形 CJ1W-EIP21 から形 CJ1W-EIP21S を使用したシステムへの置き換え A-101
A-8-1 仕様や機能などの差異 A-101
A-8-2 置き換えフロー A-105
A-8-3 事前準備 A-107
A-8-4 置き換えの実行 A-114

A-9 ソケットサービス機能のサンプルプログラム A-117
A-9-1 システム構成 A-117
A-9-2 サンプルプログラムで必要な設定 A-118
A-9-3 デバイス変数操作による UDP 通信のサンプルプログラム A-120
A-9-4 デバイス変数操作による TCP 通信のサンプルプログラム A-124

A-9-5 SendCmd 命令による UDP 通信のサンプルプログラム A-129
A-9-6 SendCmd 命令による TCP 通信のサンプルプログラム A-137

A-10 Generic デバイスを用いたタグデータリンク設定（Network Configurator Ver.3.74 以降） A-145
A-10-1 Generic デバイスの作成手順 A-145
A-10-2 Generic デバイスのタグ／タグセットの作成 A-146

A-11 バージョン関連情報 A-150

A-1 NJ シリーズ CPU ユニットと他シリーズとの EtherNet/IP 機能の比較

○：サポート有り、×：サポート無し

項　目	EtherNet/IP ユニット 形 CS1W/CJ1W-EIP21S	形 CS1W/CJ1W-EIP21S 以外の EtherNet/IP ユニット（CJ2 CPU ユニット内蔵ポート）			NJ シリーズ CPU ユニット内蔵 EtherNet/IP ポート	CJ シリーズ Ethernet ユニット
	ユニット Ver.1.0 ロット番号 241001□ 以降	ユニット Ver.1.0	ユニット Ver.2.0	ユニット Ver.2.1 以降		
タグデータリンク通信サービス	○	○	○	○	○	×
CIP メッセージ通信サービス	○	○	○	○	○	×
ソケットサービス	○	×	×	×	○	○
ファイル転送（FTP）	○	×	○	○	○	○
メール送受信	×	×	×	×	×	○
WEB 機能	×	×	×	×	×	○
コントローラ内蔵時計の自動調整	○	×	○	○	○	○
異常履歴	○	○	○	○	○ *1	○
PING コマンド応答	○	○	○	○	○	○
SNMP/SNMP トラップ	○	×	○	○	○	×
IP アドレス CIDR 機能	○	×	○	○	○	×
CX-One からの EtherNet/IP によるオンライン接続	○	×	○	○	×	×
Network Configurator からの EtherNet/IP によるオンライン接続	○	○	○	○	○	×
NJ シリーズ CPU ユニットへの装着	○ *2 *3	×	×	○ *2 *3	-	×

*1 内蔵 EtherNet/IP ではイベントログを示します。
*2 形 CJ1W-EIP21/EIP21S のみ NJ シリーズ CPU ユニットへの装着が可能です。
*3 EtherNet/IP ユニット経由で CPU ユニットに接続する場合、以下の機能は使用できません。
- Sysmac Studio（Network Configurator 除く）からの CPU ユニットへのオンライン接続
- 表示器のトラブルシュータ機能

A-2 Sysmac Studio でタグデータリンク（EtherNet/IP コネクション）を設定する

A-2-1 Sysmac Studio によるタグデータリンク（EtherNet/IP コネクション）設定の概要

NJ シリーズコントローラでタグデータリンク（EtherNet/IP コネクション）*1 を行う場合、Sysmac Studio で設定が可能となります。

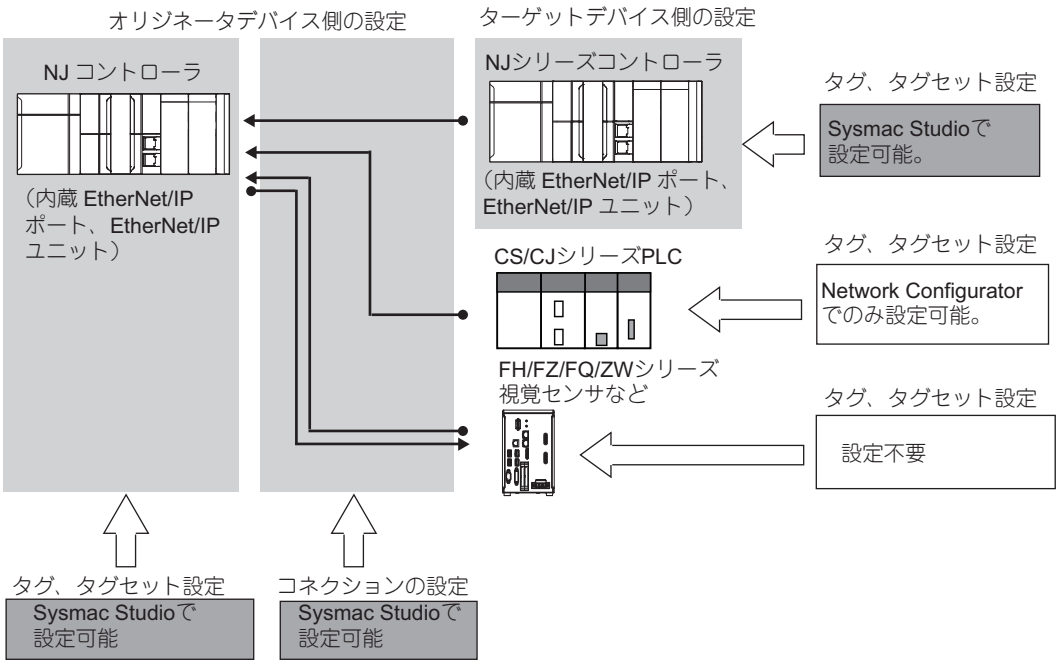
*1. EtherNet/IP ネットワーク上で、コントローラーコントローラ間、またはコントローラー他のデバイス間で、互いのタグをサイクリックにデータ交換する機能をタグデータリンク、もしくは EtherNet/IP コネクションと呼びます。本項では、これらを EtherNet/IP コネクションと表記します。

バージョン関連情報

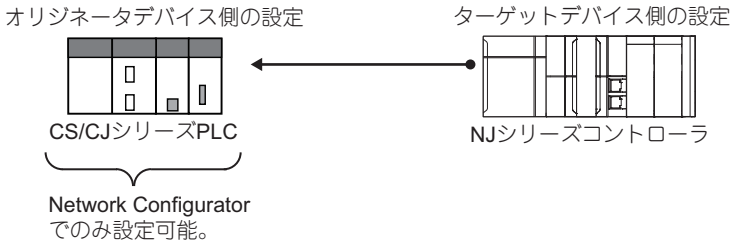
タグデータリンク（EtherNet/IP コネクション）設定は、Sysmac Studio Ver.1.10 以降で使用可能です。

Sysmac Studio で EtherNet/IP コネクション設定が可能なシステム構成について

オリジネータデバイスが NJ シリーズコントローラの場合は、EtherNet/IP コネクションのオリジネータデバイスの設定を Sysmac Studio を使って設定できます。
また、ターゲットデバイスも NJ シリーズコントローラの場合は、ターゲットデバイスのタグ・タグセットも、Sysmac Studio で設定することができます。



CS/CJ シリーズ PLC がオリジネータデバイスの場合は、Network Configurator で設定します。



A-2-2 Sysmac Studio による EtherNet/IP コネクション設定の手順

		設定画面
1 デバイスの登録 EtherNet/IP コネクションするデバイスをプロジェクトに登録します。		• メイン画面
2 ネットワーク変数を作成 (*1)	… 「ネットワーク変数の登録（オリジネータデバイス）」(P.A-12)	Sysmac Studio グローバル変数テーブル
3 タグ、タグセットの登録 手順 2 で設定したネットワーク変数をタグ、タグセットとして登録します。	… 「タグ・タグセットの登録」(P.A-14)	EtherNet/IP コネクション設定（タグセット画面）
4 コネクションの設定 EtherNet/IP コネクションで通信するデバイス（ターゲットデバイス、 オリジネータデバイス）、タグセットを指定	… 「コネクションの設定（オリジネータデバイス）」(P.A-17)	EtherNet/IP コネクション設定（コネクション画面）
5 Sysmac Studio のオンライン接続	… 「コネクション設定のデータ転送」(P.A-27)	• メイン画面
6 EtherNet/IP コネクション設定のダウンロード ※コネクションはダウンロード後自動で開始します。	… 「コネクション設定のデータ転送」(P.A-27)	• 同期画面 / 一括設定画面 • EtherNet/IP コネクション設定
7 動作の確認 コネクションの停止／開始	… 「A-2-5 ネットワークの通信ステータス確認と処置方法」(P.A-32)	EtherNet/IP コネクションモニタ画面

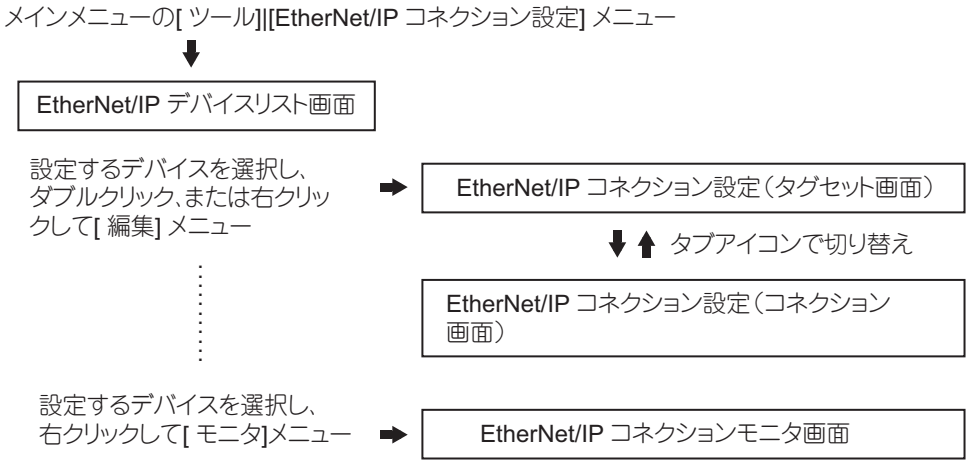
*1. ネットワーク変数とは、グローバル変数テーブルで [ネットワーク公開属性] が [出力] もしくは [入力] に設定された変数のことです。

A-2-3　EtherNet/IP コネクション設定機能について

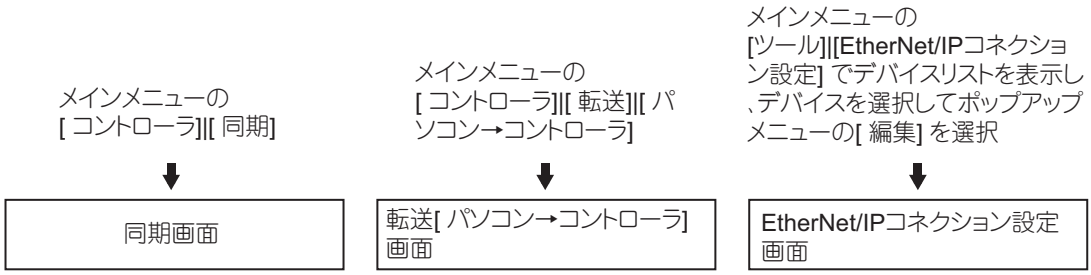
EtherNet/IP の EtherNet/IP コネクション設定機能の画面構成について説明します。

● EtherNet/IP コネクション設定機能の画面遷移

・ コネクション設定



・ コネクション設定の転送（パソコン→コントローラ）

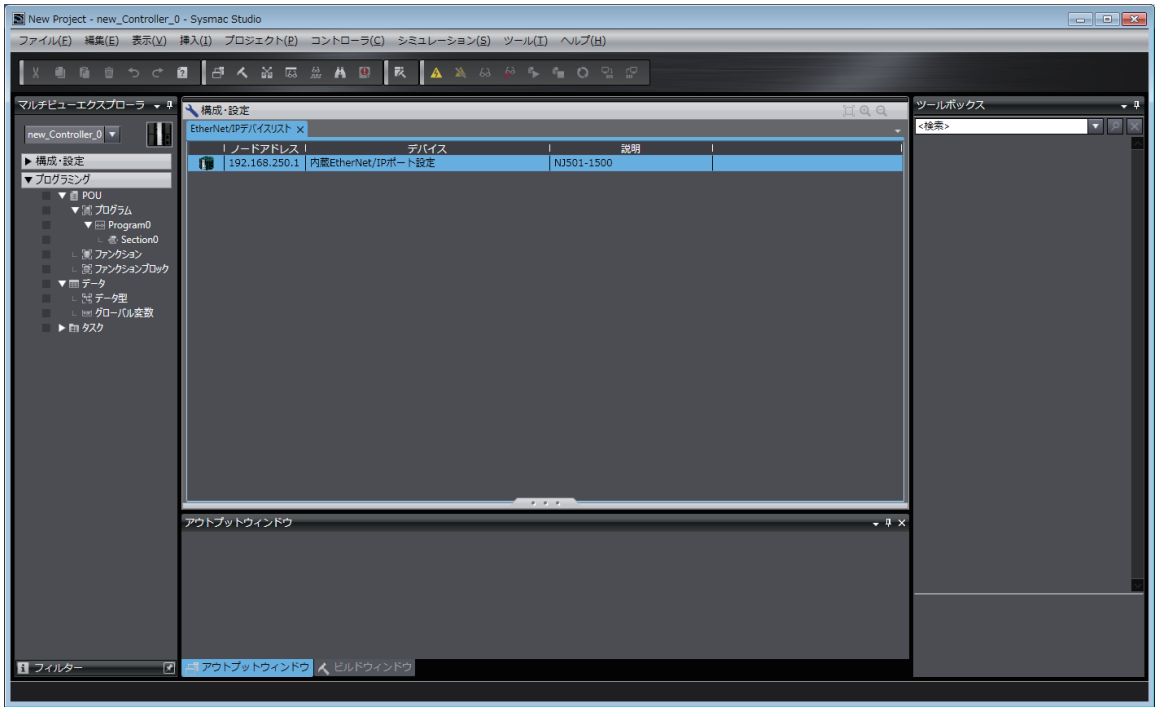


使用上の注意

コネクション設定のみを転送したい場合は、EtherNet/IP コネクション設定画面の転送機能をお使いください。
［転送［パソコン→コントローラ］］画面、または［転送［コントローラ→パソコン］］画面では、部分転送をしないため、パソコン側とコントローラ側のデータが一致している状態では、「コネクション設定を転送しない」オプションだけを OFF にしても、コネクション設定は転送されません。

● **EtherNet/IP デバイスリスト画面（「タグ・タグセットの登録」(P.A-14)）**

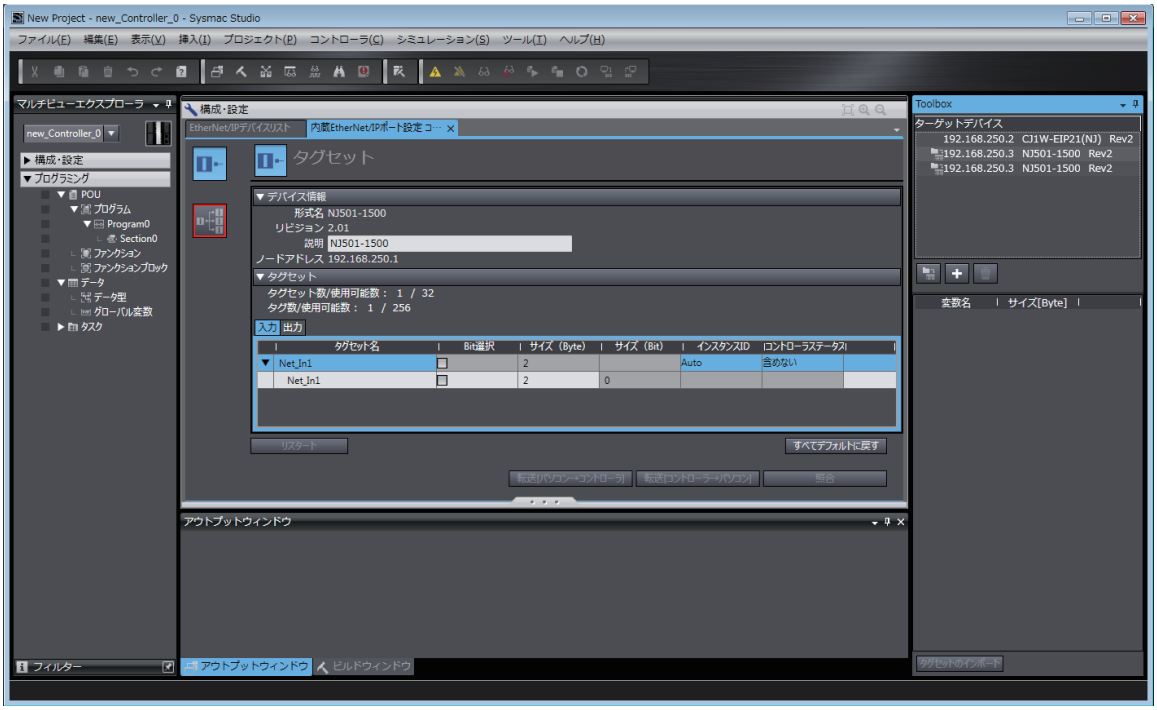
EtherNet/IP コネクションを設定できるデバイスが表示されます。



- **EtherNet/IP コネクション設定（タグセット画面）（「タグ・タグセットの登録」(P.A-14)）**

  コネクションに必要なタグセットを登録します。
  タグセットとは、コネクションによるデータ送受信の単位で、タグを最大 8 つまで登録できます。タグは、ネットワーク変数 *1 と同じ名前とサイズである必要があります。タグセットには、コントローラステータス情報をタグに含むかどうかを、出力タグには運転停止異常時のデータ出力動作を設定します。

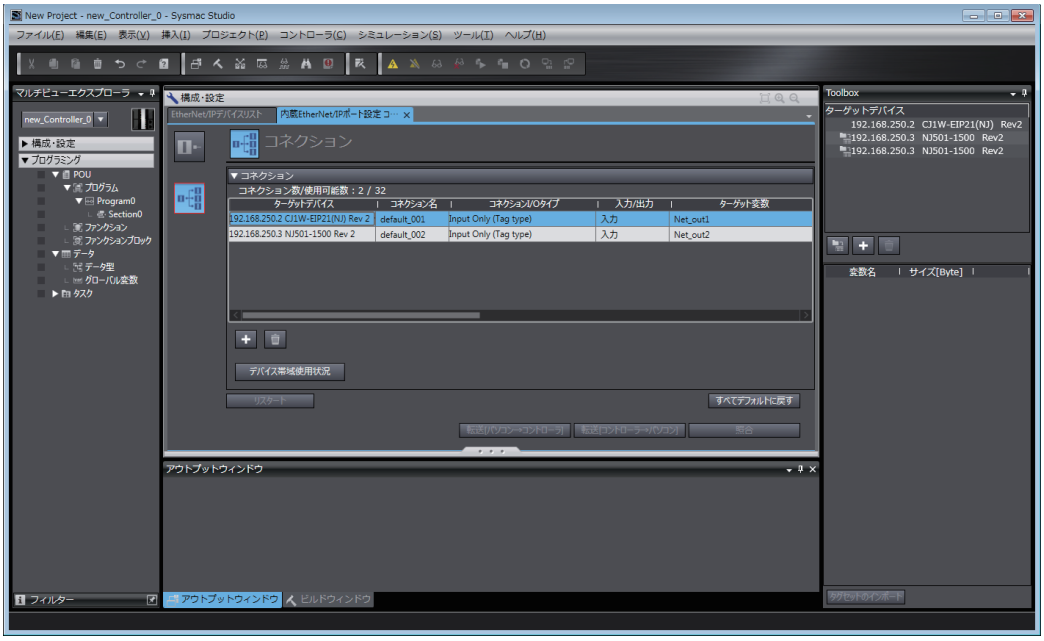
*1. ネットワーク変数とは、グローバル変数テーブルで［ネットワーク公開属性］が［出力］もしくは［入力］に設定された変数のことです。



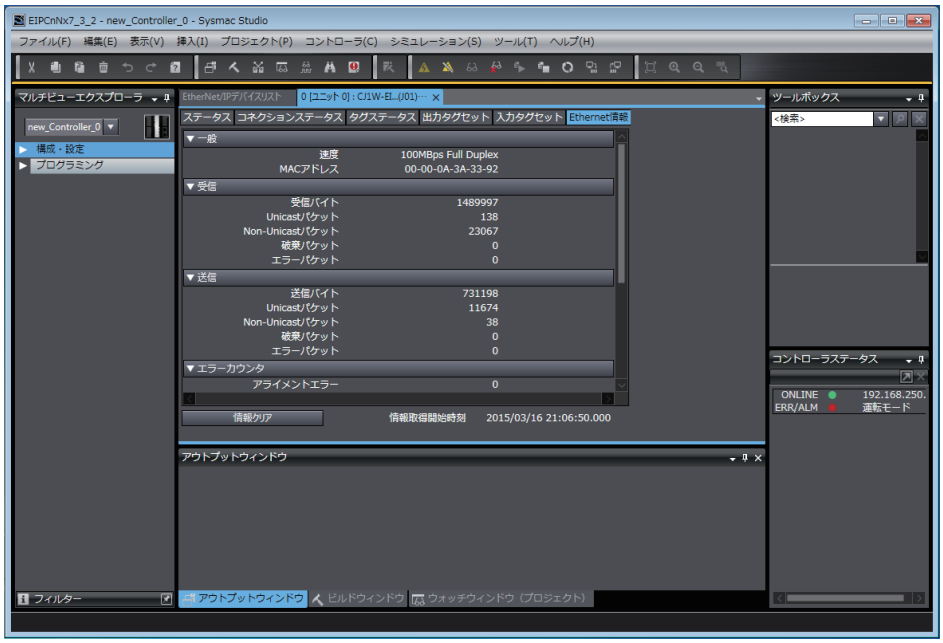
- **EtherNet/IP コネクション設定（コネクション画面）（「コネクションの設定（オリジネータデバイス）」(P.A-17)）**
  ターゲットデバイスを指定し、コネクションを設定します。
  各コネクションには、コネクション名、コネクション IO タイプ、入出力、ターゲットデバイスのタグセット（ターゲット変数）、オリジネータデバイスのタグセット（オリジネータ変数）、及び、RPI 値とタイムアウト値を設定します。

使用上の注意

コネクション設定後、ターゲットデバイスの IP アドレスまたはターゲットデバイス（形式・リビジョンを含む）を変更した場合は、次のようにしてください。
Sysmac Studio Ver.1.11 以降の場合は、コネクション設定を一括で変更してください。
Sysmac Studio Ver.1.10 の場合は、コネクションを再度作成してください。



- **EtherNet/IP コネクションモニタ画面（「A-2-5 ネットワークの通信ステータス確認と処置方法」(P.A-32)）**
  EtherNet/IP コネクションの設定状態（オフライン）や通信状態（オンライン）を確認できます。オンラインでは、コネクションの起動と停止も可能です。

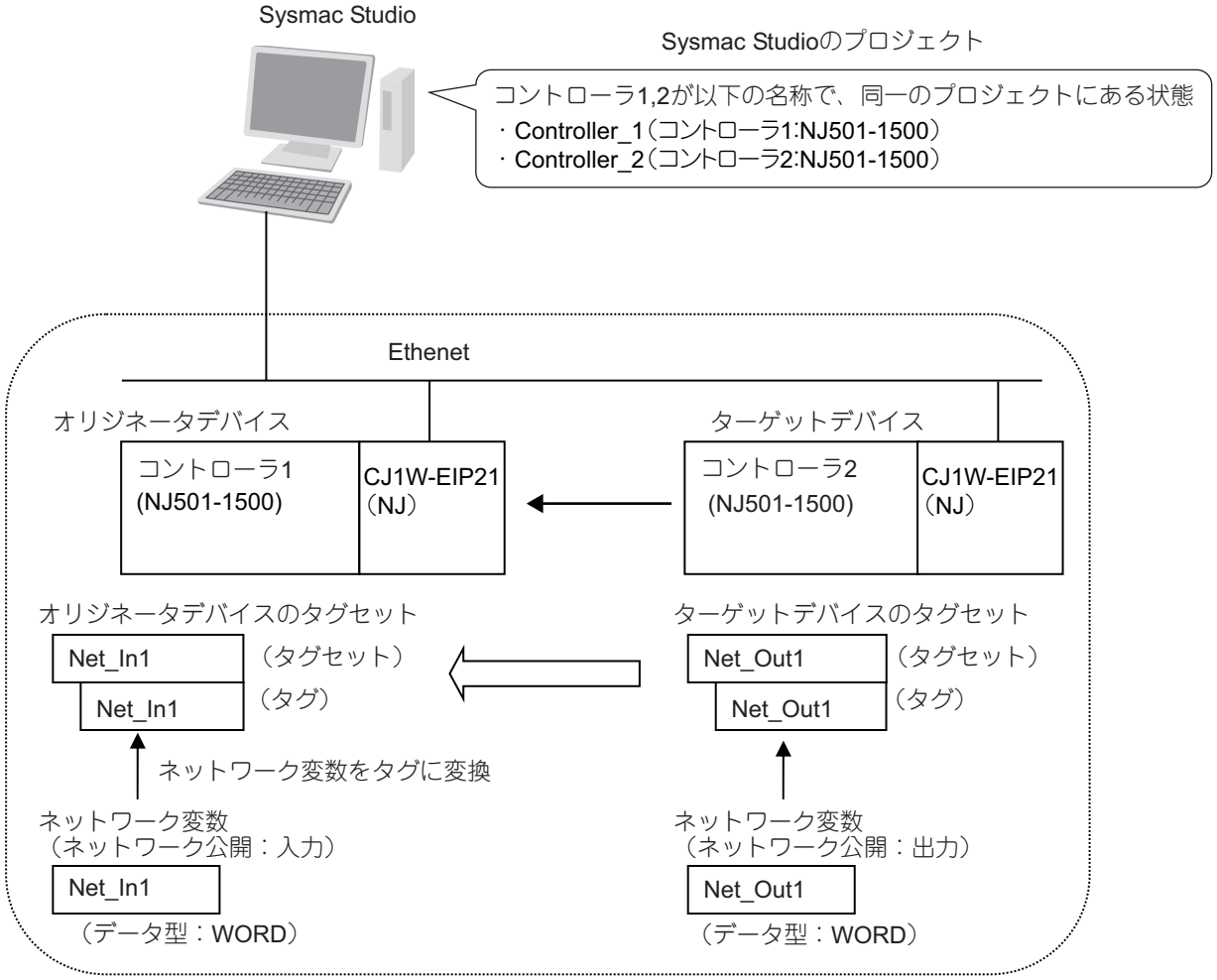


A-2-4　Sysmac Studio による EtherNet/IP コネクション設定

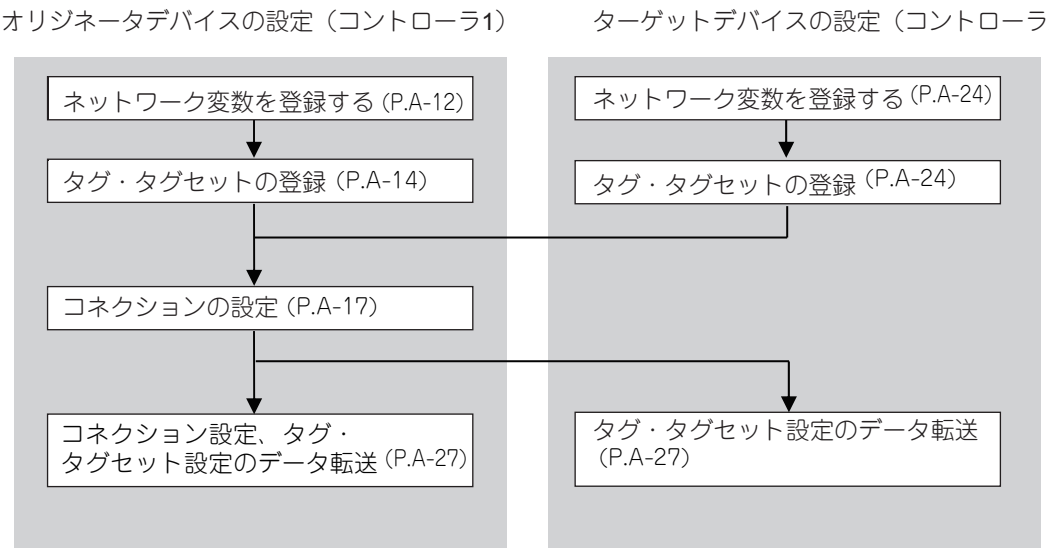
Sysmac Studio による EtherNet/IP コネクションの設定手順を説明します。
ここでは、以下のようなシステム構成で、EtherNet/IP コネクションを設定する場合の設定操作について説明します。

例）コントローラ 1 の EtherNet/IP ポートとコントローラ 2 の EtherNet/IP ポートが Ethernet で接続されているシステム

- コントローラ 2 のネットワーク変数 (Net_Out1) の値をコントローラ 1 のネットワーク変数 (Net_In1) に、50ms の周期 (RPI) で送信する設定をおこないます。
- コントローラ 1 とコントローラ 2 のプログラムは同一のプロジェクトに登録されているとします。



EtherNet/IP コネクションを行うコントローラ 1 とコントローラ 2 に、各設定を以下の順に行っていきます。
オリジネータデバイスとターゲットデバイスごとに、設定が必要な項目は、以下のようになります。



ネットワーク変数の登録（オリジネータデバイス）

EtherNet/IP コネクションでデータを送受信するネットワーク変数を登録します。
変数の登録操作については、『Sysmac Studio Version 1 オペレーションマニュアル』(SBCA-470) を参照してください。

1 **オリジネータデバイスとなるコントローラ 1 に、EtherNet/IP コネクションのタグで使用するネットワーク変数を設定します。**
この変数は、ターゲットデバイスであるコントローラ 2 からのデータを受信するネットワーク変数となります。

グローバル変数テーブルで、グローバル変数の設定「ネットワーク公開」を、「入力」または「出力」に設定することで、EtherNet/IP コネクションで使用できるネットワーク変数となります。
本例ではコントローラ 1 のネットワーク変数を、以下のように設定します。



- 変数名：Net_In1
- データ型：WORD
- ネットワーク公開：入力

● EtherNet/IP コネクションで使用するネットワーク変数について

・ ネットワーク変数名
EtherNet/IP コネクションの設定のタグ名には、I/O メモリアドレス名称は使用できません。
よって、タグの元となるネットワーク変数の名称にも、I/O メモリアドレス名を使用しないようにしてください。
I/O メモリアドレス名として認識されるのは、以下の文字列です。

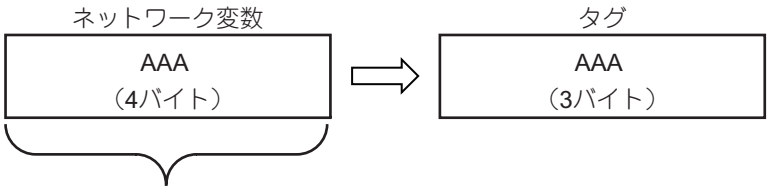
(1) 半角数字のみからなる変数名（0000 ～ 6143）

(2) 以下の半角英字（大文字／小文字とも）と、半角数字からなる変数名

- H（H000 ～ H511）
- W（W000 ～ W511）
- D（D00000 ～ D32767）
- E0_ ～ E18_（E0_00000... E0_32767 ～ E18_00000... E18_32767）

なお、I/O メモリアドレスをタグに指定したい場合は、タグに I/O メモリアドレスを直接指定するのではなく、Sysmac Studio で I/O メモリアドレスを AT 指定した変数を作成し、その AT 指定した変数を、タグとして指定してください。

・ 変数のサイズ
EtherNet/IP ユニットを EtherNet/IP デバイスとして使用する場合、EtherNet/IP コネクションで使用するネットワーク変数のサイズは、タグのサイズが奇数バイトであっても、偶数バイトで設定してください。



・ データの同期性について
「タグ」に割り付けるネットワーク変数には、変数の値の同期性を確保するため、「更新タスク」を設定する必要があります。
詳細については、📖「7-1-7 タグデータリンクするデータの同期性について」を参照ください。

タグ・タグセットの登録

EtherNet/IP コネクションでデータ転送に必要なタグ・タグセットを登録します。
タグ・タグセットの登録は、EtherNet/IP コネクション設定画面で設定します。

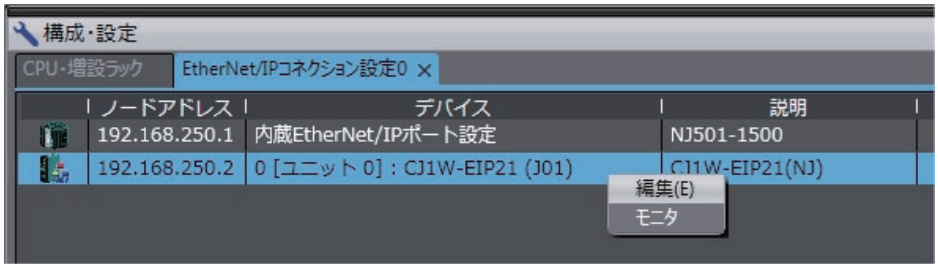
使用上の注意

同一タグセット内のタグのデータを同じタイミングで更新させたい場合、以下のように設定してください。

- 同一タグセット内のタグに指定する変数群は、あらかじめ Sysmac Studio にて「更新タスク」として同じタスクを指定してください。
- I/O メモリを AT 指定したタグ（変数）と、AT 指定していないタグ（変数）は、同一タグセット内で混在させないでください。

1 **メインメニューの［ツール］｜［EtherNet/IP コネクション設定］を選択します。**
EtherNet/IP デバイスリスト画面を表示します。

2 **本例ではオリジネータデバイスとなるコントローラ 1 に装着されている EtherNet/IP ユニットの [CJ1W-EIP21] を選択し、右クリックして［編集］を選択し、EtherNet/IP コネクション設定画面を開きます。**



3 **EtherNet/IP コネクション設定画面の（タグセット画面表示）アイコンをクリックします。**

4 **［入力］タブをクリックし、［入力］タブリストの画面を切り替えて、タグセットとタグを登録していきます。**
タグセットとタグは、以下のいずれかの方法で登録できます。

- 個別登録 ： コントローラに登録されているネットワーク変数を、手動でタグとして登録します。
- 一括登録 ： コントローラに登録されているネットワーク変数を、一括してタグとして登録します。

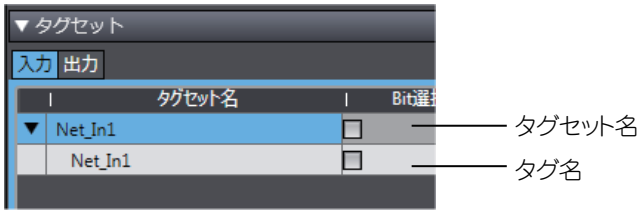
5 **タグセットとタグの個別登録を行います。**

(1) EtherNet/IP コネクション設定画面の入力タブ内で右クリックして、［タグセットの新規登録］を選択します。

(2) ［入力］タブのリストに、タグセット名称を「Net_In1」と直接入力します。

(3) 入力タブ内で右クリックして、［タグの新規登録］を選択します。

(4) タグ名称を「Net_In1」と入力します。



使用上の注意

タグセット名は任意で構いませんが、タグ名は、実際にコントローラに登録されているネットワーク変数と同じ名称で設定してください。
文字を入力する（もしくは Ctrl ＋スペースキーを押す）と Sysmac Studio の入力支援機能によりコントローラに登録されている変数名がリスト表示されますので、そのリストの中から選択することができます。

参考

タグセットには、最大 8 個までのタグを登録できます。
その場合、以下のように設定してください。
例）

	タグセット名	
▼	Network_Input_Value	....（タグセット名）
	Net_In1	....（タグ名）
	Net_In2	....（タグ名）

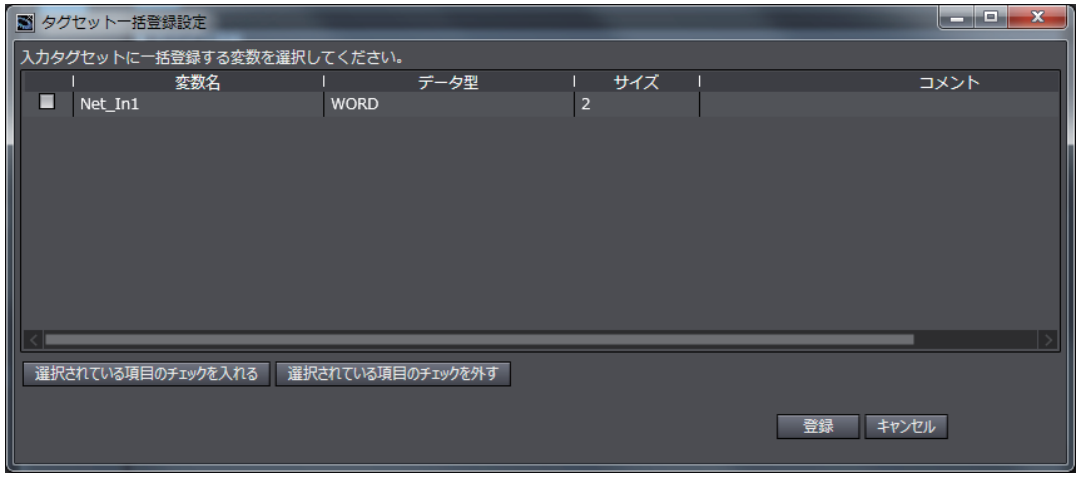
6 タグセットとタグの一括登録を行います。

(1) EtherNet/IP コネクション設定画面の入力タブ内で右クリックし、[タグセットの一括登録] を選択して、[タグセット一括登録設定] 画面を表示します。

タグセット一括登録設定画面では、グローバル変数テーブル登録されている以下の変数が一覧表示されます。

［入力］タブでのタグセット一括登録　：［ネットワーク公開］が［入力］に設定されている変数が一覧表示されます。

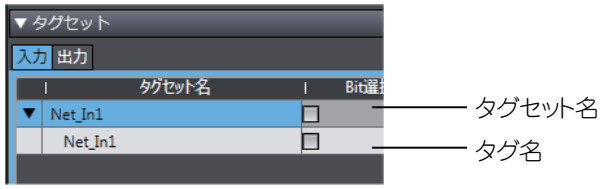
［出力］タブでのタグセット一括登録　：［ネットワーク公開］が［出力］に設定されている変数が一覧表示されます。



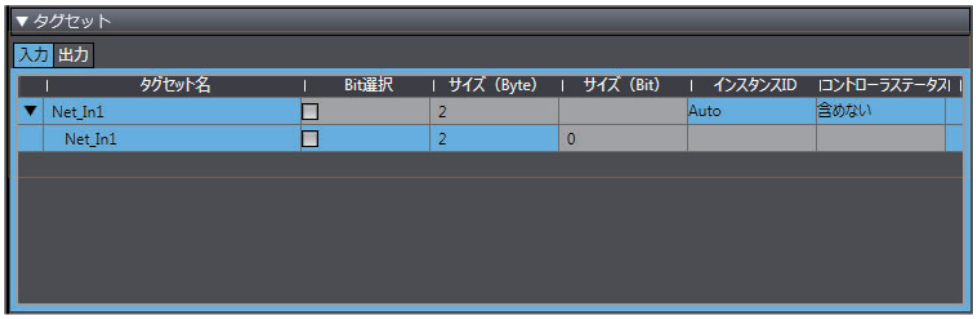
(2) タグとして登録したい変数にチェックが入っている状態にして、[登録] ボタンをクリックします。

(3) EtherNet/IP コネクション設定画面で、自動登録されたタグがリストに追加されます。

自動登録で登録した場合、タグセットとタグは同じ名称となり、1 タグセット内に 1 つのタグという構成で登録されます。



7 **登録したタグ・タグセットに対して、以下の設定を行います。**



・タグセットの設定

項目名	設定項目
タグセット名	タグセットの名を入力します。 タグセット名は任意に変更できます。
サイズ（Byte）	タグの合計サイズ（バイト）を表示します。
インスタンス ID	インスタンス ID を表示します。 • Auto • IN_{min}…IN_{max} {min} は対象デバイスの EDS ファイルに記載されてある Produced Assembly ID の最小値。 {max} は対象デバイスの EDS ファイルに記載されてある Produced Assembly ID の最大値。
コントローラステータス	タグセットにコントローラステータスを含めるかどうかを選択します。

・タグの設定

項目名	設定項目
タグ名	タグ名を入力します。 タグ名は、実際にコントローラに登録されているネットワーク変数と同じ名称で設定してください。
Bit 選択	タグのデータサイズを、ビットで設定するかどうかを選択します。 チェックあり：Bit サイズにて設定します。 チェックなし：Byte サイズにて設定します。
サイズ（Byte）	タグのサイズ（バイト）を表示します。
サイズ（Bit）	タグのサイズ（ビット）を表示します。
運転停止異常時出力	コントローラに全停止フォールトレベルのコントローラ異常が発生した時に、出力データをクリアするか、全停止フォールト異常が発生する前のデータをそのまま送信し続けるかを選択します。 • 保持 • クリア

コネクションの設定（オリジネータデバイス）

タグセット登録後、EtherNet/IP コネクションでデータ転送するためのコネクション設定を行います。
コネクション設定は、オリジネータデバイス（この例ではコントローラ 1）でのみ行います。

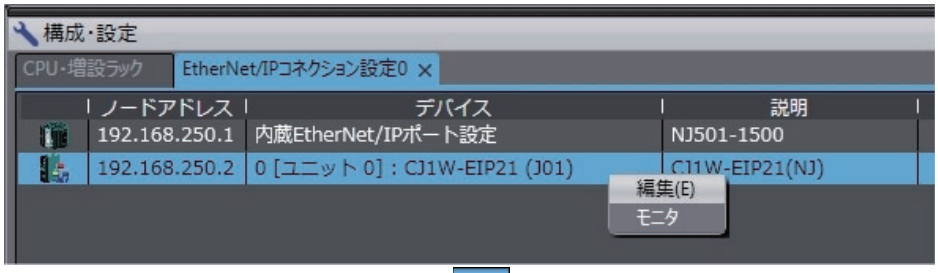
本例によるコネクション設定をする前に、オリジネータデバイスとなるコントローラ 2 について、あらかじめ、タグ・タグセットの登録をしておいてください。
タグ・タグセットの登録操作については、📖「 タグ・タグセットの登録（ターゲットデバイス）」(P.A-24) を参照してください。

使用上の注意

コネクション設定後、ターゲットデバイスの IP アドレスまたはターゲットデバイスのリビジョンを含む形式を変更した場合は、コネクション設定に含まれているターゲットデバイスの設定も同様に変更する必要があります。
コネクション設定に含まれるターゲットデバイスの設定の変更方法については、📖「コネクション設定後に、ターゲットデバイスの設定を変更した場合について」(P.A-22) を参照してください。

1 **メインメニューの［ツール］｜［EtherNet/IP コネクション設定］を選択して EtherNet/IP デバイスリスト画面を表示します。**

2 **本例ではオリジネータデバイスとなるコントローラ 1 に装着されている EtherNet/IP ユニットの [CJ1W-EIP21] を選択し、右クリックして［編集］を選択し、EtherNet/IP コネクション設定画面を開きます。**
EtherNet/IP コネクション設定画面が開きます。

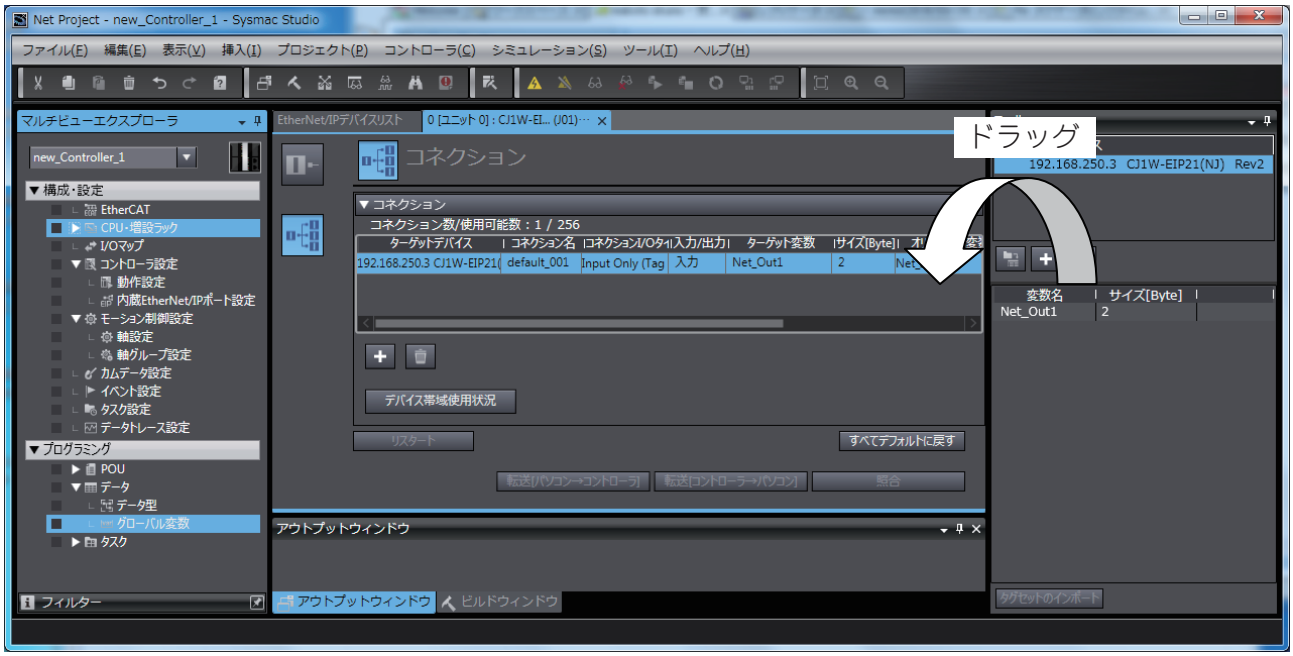


3 **EtherNet/IP コネクション設定画面の（コネクション編集画面表示）アイコンをクリックします。**

4 **EtherNet/IP コネクション設定画面右のツールボックスの［ターゲットデバイス］から、[CJ1W-EIP21(NJ)] を選択します。**
[CJ1W-EIP21(NJ)] を選択すると、ツールボックスの［変数］に、コントローラ 2 で設定したターゲットデバイスのタグセット (Net_Out1) が表示されます。

5 **ツールボックスの［変数］に表示されたターゲットデバイスのタグセット (Net_Out1) を選択して、コネクション一覧にドラッグします。**

コネクションのターゲット変数は、文字を入力する（もしくは Ctrl ＋スペースキーを押す）と設定可能な値がリスト表示されますので、そのリストの中から選択することもできます。



6 **ターゲットデバイスのタグセットをドラッグしたコネクション行で、［オリジネータ変数］と［オリジネータ変数サイズ [Byte]］を選択します。**

ここでは、［オリジネータ変数］を［Net_In1］に、［サイズ [Byte]］を［2］と設定します。

それ以外の項目で、設定変更が必要な項目があれば、設定内容を変更します。
コネクション設定は、以下の設定項目があります。

項目名	設定方式
ターゲットデバイス	ターゲットデバイスを選択します。
コネクション名	コネクションに任意の名称をつけることができます。（最大 32 文字）
コネクション I/O タイプ	CS1W-EIP21、CS1W-EIP21S、CJ1W-EIP21、CJ2B-EIP21、CJ2M-EIP21、CJ1W-EIP21(CJ2)、CJ1W-EIP21(NJ)、CJ1W-EIP21S、CJ1W-EIP21S(CJ2)、CJ1W-EIP21S(NJ)、NX701-□□□□、NJ501-□□□□、NJ301-□□□□、NJ101-□□□□で EtherNet/IP コネクションを行う場合は、Input Only(Tag type) が選択されます。 その他のターゲットデバイスと EtherNet/IP コネクションを行う場合は、対象デバイスの EDS ファイルに指定されたコネクション I/O タイプを選択します。 「Input Only(ID type)」は、他社ノードがオリジネータで Tag type によるコネクション設定をサポートしていない場合に使用します
入力／出力	コネクションの選択にともない自動的に表示されます。 Input Only：［入力］とだけ表示されます。
ターゲット変数	ターゲット側のタグセットを選択して、割り付けます。 • ［入力／出力］が入力の場合：ターゲットの出力（送信）タグセットを選択します。 • ［入力／出力］が出力の場合：ターゲットの入力（受信）タグセットを選択します。
サイズ [Byte]	ターゲット変数のデータサイズが表示されます。

項目名	設定方式
オリジネータ変数	オリジネータ側のタグセットを選択して、割り付けます。 • ［入力／出力］が入力の場合：オリジネータの入力（受信）タグセットを選択します。 • ［入力／出力］が出力の場合：オリジネータの出力（送信）タグセットを選択します。
サイズ［Byte］	オリジネータ変数のデータサイズを入力します。
コネクションタイプ	データをマルチキャスト／ユニキャスト、いずれで送信するかを選択します。デフォルトはマルチキャストです。 • Multi-cast connection：同一データを複数ノードで共有するときに選択します。通常はこちらを選択します。 • Point to Point connection：同一データを複数ノードで共有しないときに選択します。 ユニキャストで通信されるので、その他のノードに必要のない負荷を与えたくない場合に選択します。 （注）マルチキャストとユニキャストの使い方、コネクション数の数え方については、「7-1-4 使用方法の概要」を参照してください。
RPI［ms］	オリジネータとターゲット間の、コネクションごとのデータ更新周期（パケットインターバル）を設定します。 デフォルトは 50ms です（50ms に 1 回の更新になります）。
タイムアウト値	コネクションタイムアウトを検知する時間を設定します。 設定は、パケットインターバル（RPI）の 4、8、16、32、64、128、256、512 倍から選択可能です。 デフォルトは、［RPI×4］です。 タイムアウト値は 10ms 以上となるように設定する必要があります。

7 ツールボックスに表示されるターゲットデバイスは、オリジネータデバイスと同じ Sysmac Studio のプロジェクトに登録されているデバイスが表示されます。

オリジネータデバイスと同じ Sysmac Studio のプロジェクトに登録されてないデバイスを、ターゲットデバイスの一覧に追加するには、以下のいずれかの方法で追加できます。

- 他プロジェクトで登録されているデバイスをインポートする
  他のプロジェクトデータに登録されている NJ シリーズコントローラをインポートして、デバイス一覧に追加できます。
- 任意の設定でデバイスを登録する
  ターゲットデバイスを手動でデバイス一覧に追加できます。

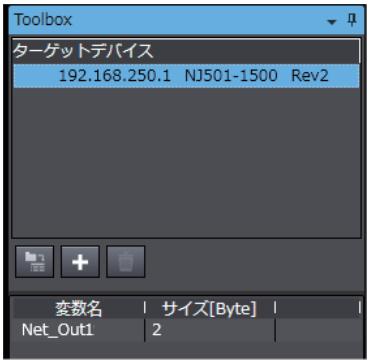
参考

コネクション情報が記述されている EDS ファイルを Sysmac Studio にインストールして、ターゲットデバイスを Sysmac Studio のプロジェクトに登録することにより、デバイスリストにデバイスを追加することもできます。

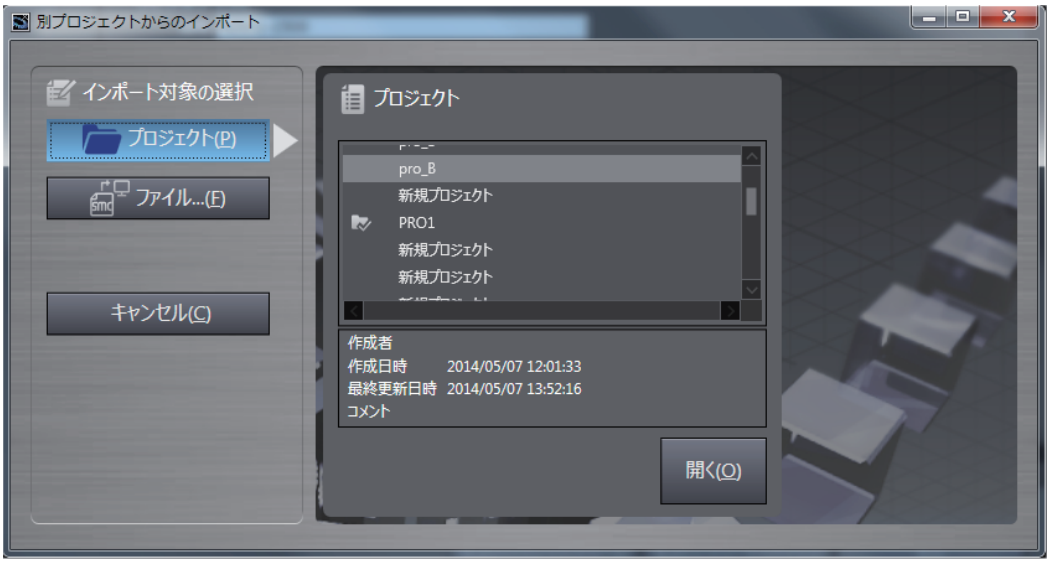
📖「EDS ファイルの追加」（P.A-21）を参照してください。

8 **他プロジェクトで登録されているデバイスをインポートします。**

(1) EtherNet/IP コネクション設定画面右のツールボックスの（インポート）ボタンをクリックします。



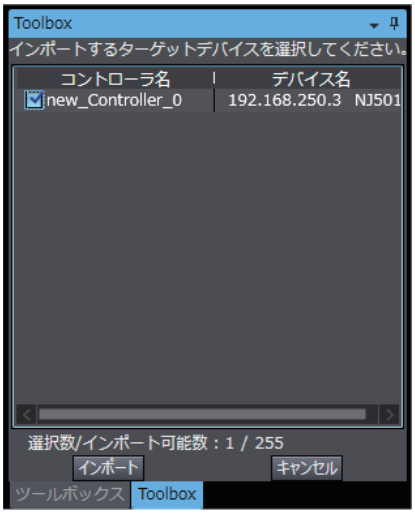
(2) 他プロジェクトのインポート選択画面が表示されますので、[プロジェクト] ボタンをクリックして、インポートするプロジェクトを選択して [開く] ボタンをクリックします。



(3) 選択したプロジェクトに登録されている EtherNet/IP デバイスリストが表示されますので、インポートするターゲットデバイスをチェックして、[インポート] ボタンをクリックします。

(注)EtherNet/IP コネクション設定を含んだプロジェクトに限ります。

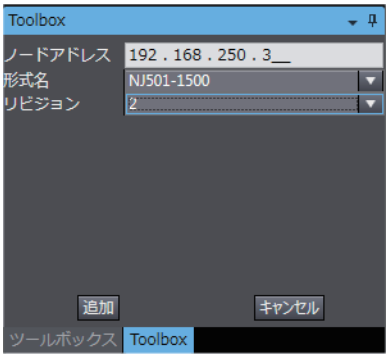
ツールボックスのターゲットデバイス一覧に、インポートした EtherNet/IP デバイスが追加されます。



9 **任意にデバイスを登録します。**

(1) ツールボックスのターゲットデバイス一覧下の [+] ボタンをクリックします。
ターゲットデバイス追加画面が表示されます。

(2) 追加するターゲットデバイスについて、項目を入力します。



メニュー名	設定する内容
ノードアドレス	ターゲットデバイスの IP アドレスを入力します。
形式名	ターゲットデバイスの形式を選択します。
リビジョン	ターゲットデバイスのリビジョンを選択します。

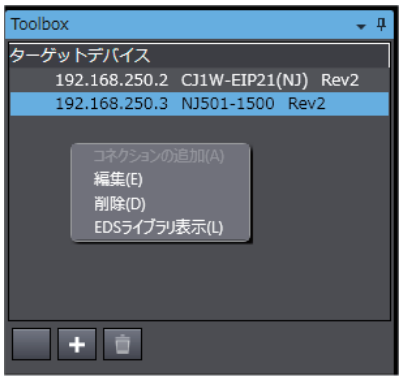
(3) ここでは、コントローラ 3 を、以下のように設定して、[追加] をクリックします。
ツールボックスに表示されたターゲットデバイスに、ターゲットデバイスが追加されます。
[ノードアドレス]：192.168.250.3
[コントローラ]：NJ501-1500
[リビジョン]：2

(4) [タグセットのインポート] ボタンをクリックすることにより、Network Configurator で設定したタグセットを、ターゲットデバイスにインポートすることができます。
インポートするファイル（CSV ファイル）は、Network Configurator の［デバイスパラメータの編集］ダイアログボックスの［タグセット］タブで、［ファイル操作｜ファイル操作へエクスポート］ボタンをクリックして作成します。

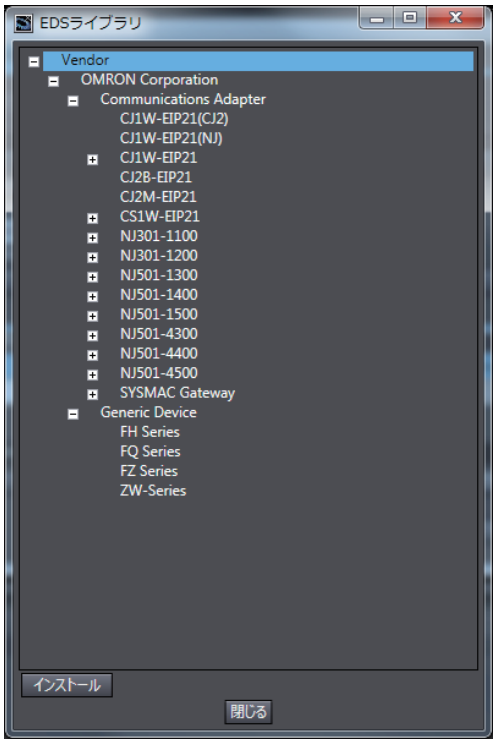
● EDS ファイルの追加

（注） ModularEDS デバイスは、Sysmac Studio Ver.1.11 以降で対応しています。

1 **EtherNet/IP コネクション設定画面のツールボックスのターゲットデバイスリストで、右クリックして、[EDS ライブラリ表示] メニューを選択します。**



2 **EDS ライブラリ画面が表示されますので、［インストール］ボタンをクリックします。**



3 **追加する EDS ファイルを選択し［開く］ボタンをクリックします。**
EDS ファイルが追加されます。

4 **EDS ファイルをインストールした EtherNet/IP デバイスが、EDS ライブラリに追加されます。**
EDS ライブラリに表示されるデバイスは、EtherNet/IP コネクション設定画面のツールボックスで、ターゲットデバイスの追加を行う場合の形式名の選択候補として一覧表示されます。

● コネクション設定後に、ターゲットデバイスの設定を変更した場合について

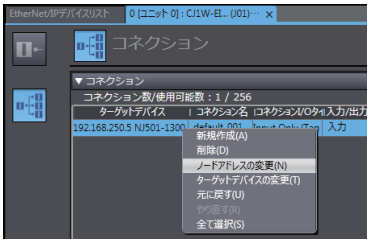
コネクション設定後、ターゲットデバイスの IP アドレスまたはターゲットデバイスのリビジョンを含む形式を変更した場合は、コネクション設定に含まれているターゲットデバイスの設定も同様に変更する必要があります。
ターゲットデバイスの変更は、一括で行うことができます。

使用上の注意

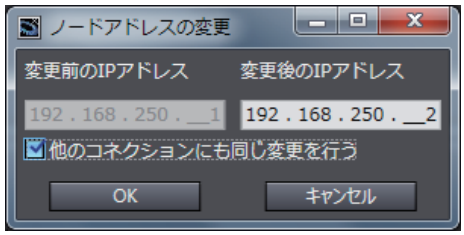
Ver.1.10 以前の Sysmac Studio を使用している場合で、コネクション設定後にターゲットデバイスを変更したときは、コネクションを再度作成し直してください。

ターゲットデバイスの IP アドレスを一括変更する場合

1 **コネクション行を右クリックし、［ノードアドレスの変更］を選択します。**



2 **［ノードアドレスの変更］ダイアログボックスが表示されますので、「変更後の IP アドレス」を入力します。**
他のコネクションにも同じ変更を行う場合、「他のコネクションにも同じ変更を行う」をチェックします。



3 **他のコネクションにも同じ変更を行う場合、「他のコネクションにも同じ変更を行う」をチェックします。**

4 **[OK] ボタンをクリックします。**

ターゲットデバイス（形式・リビジョンを含む）を一括変更する場合：

1 **コネクション行を右クリックし、［ターゲットデバイスの変更］を選択します。**

2 **［ターゲットデバイスの変更］ダイアログボックスが表示されますので、「変更後」のターゲットデバイスを選択します。**



使用上の注意

- 変更可能なターゲットデバイスは Vender ID が OMRON、Device Type が Communications Adapter の EDS デバイスに限ります。
- 選択可能な変更後のターゲットデバイスはツールボックスのターゲットデバイスに登録されている必要があります。

3 **他のコネクションにも同じ変更を行う場合、「他のコネクションにも同じ変更を行う」をチェックします。**

4 **[OK] ボタンをクリックします。**

ネットワーク変数の登録（ターゲットデバイス）

1 **ターゲットデバイスとなるコントローラ 2 に、EtherNet/IP コネクションのタグで使用するネットワーク変数を設定します。**
この変数は、オリジネータデバイスであるコントローラ 1 へ送信するデータを格納するネットワーク変数となります。

グローバル変数テーブルで、グローバル変数の設定「ネットワーク公開」を、「入力」または「出力」に設定することで、EtherNet/IP コネクションで使用できるネットワーク変数となります。
本例ではコントローラ 1 のネットワーク変数を、以下のように設定します。



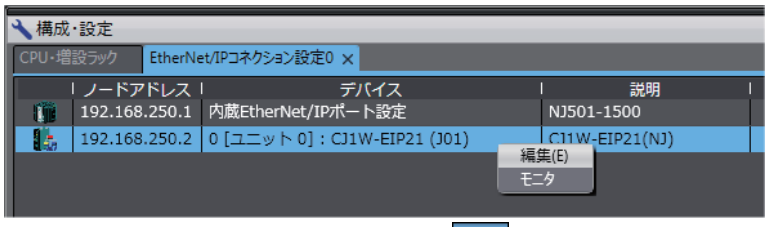
- 変数名：Net_Out1
- データ型：WORD
- ネットワーク公開：出力

タグ・タグセットの登録（ターゲットデバイス）

ターゲットデバイスのタグ・タグセットを設定します。

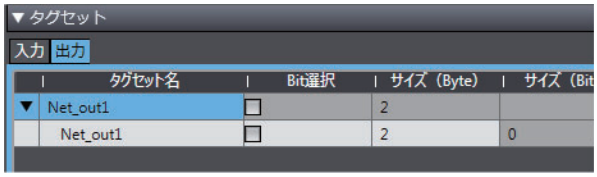
1 **メインメニューの［ツール｜EtherNet/IP コネクション設定］を選択します。**
EtherNet/IP デバイスリスト画面を表示します。

2 **本例ではオリジネータデバイスとなるコントローラ 2 に装着されている EtherNet/IP ユニットの [CJ1W-EIP21] を選択し、右クリックして［編集］を選択します。**
EtherNet/IP コネクション設定画面が開きます。



3 **EtherNet/IP コネクション設定画面の  （タグセット画面表示）アイコンをクリックします。**

4 **［出力］タブをクリックし、［出力］タブリストの画面を切り替えて、以下のタグ・タグセットを登録します。**
タグ・タグセットの登録方法は、ターゲットデバイスのタグ・タグセットの登録と同じになります。（🕮「タグ・タグセットの登録 (P.A-14)」参照）



デバイス帯域使用状況の確認

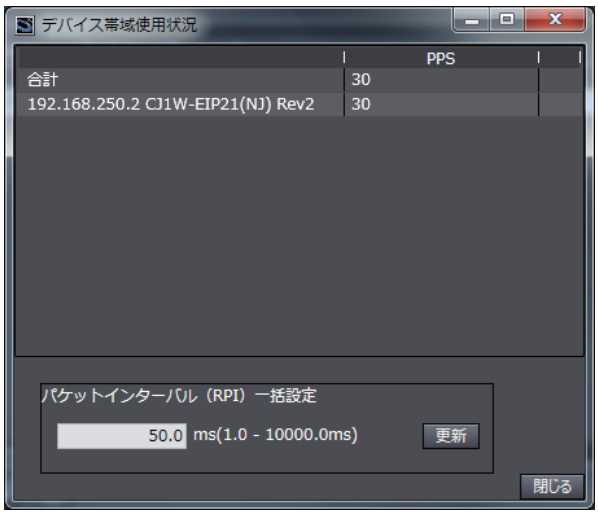
EtherNet/IP コネクションの設定から、PPS を表示できます。
表示された PPS は、マルチキャストフィルタありの場合です。

使用上の注意

デバイス帯域使用状況画面で確認できるのは、1 つのオリジネータからターゲットデバイスへの EtherNet/IP コネクションで使用する PPS のみです。
実際の EtherNet/IP ネットワーク上で使用される PPS については、他デバイスの EtherNet/IP コネクションで使用する PPS も含めて計算する必要があります。

● 操作方法

EtherNet/IP コネクション設定画面（ターゲットデバイス画面）の［デバイス帯域使用状況］ボタンをクリックします。



メニュー名	設定する内容
PPS	ターゲットデバイスごとの PPS と、その合計値を表示します。
パケットインターバル (RPI) 一括設定	すべてのターゲットデバイスのパケットインターバル (RPI) の値を一括で変更します。

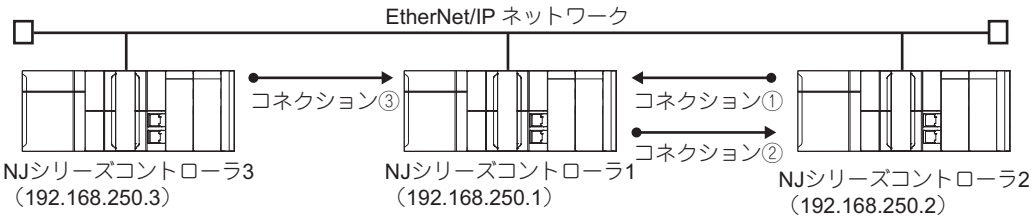
参考

［パケットインターバル (RPI) 一括設定］で値を入力して、［更新］ボタンをクリックすると、ターゲットデバイスのコネクション設定のパケットインターバル (RPI) の値を、一括で変更できます。

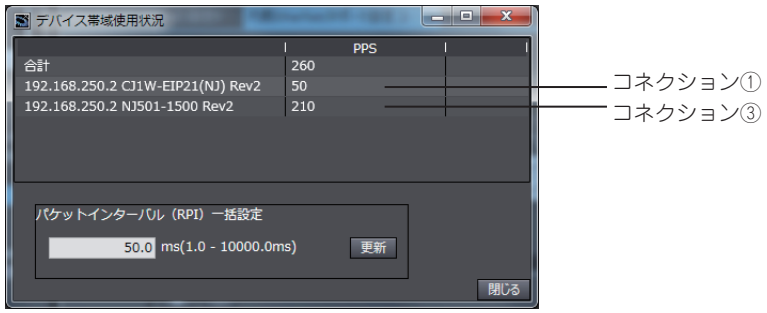
● EtherNet/IP コネクションによる各デバイスの PPS 計算例

EtherNet/IP ネットワークにおいて、コントローラ 1 ～ 3 の間で以下 3 つの EtherNet/IP コネクションを設定した場合

コネクション種類	EtherNet/IP コネクションの対象デバイス	デバイスの PSS
コネクション①	NJ シリーズコントローラ 1（オリジネータデバイス） ← NJ シリーズコントローラ 2（ターゲットデバイス）	50 PPS
コネクション②	NJ シリーズコントローラ 1（ターゲットデバイス） → NJ シリーズコントローラ 2（オリジネータデバイス）	10 PPS
コネクション③	NJ シリーズコントローラ 1（オリジネータデバイス） ← NJ シリーズコントローラ 3（ターゲットデバイス）	210 PPS



各 EtherNet/IP デバイスの PPS は、以下となります。



コントローラ 1 の EtherNet/IP コネクション設定



コントローラ 2 の EtherNet/IP コネクション設定

この例では、コネクション①の PPS が 50、コネクション②の PPS が 10、コネクション③の PPS が 210 となりますので、各 EtherNet/IP デバイスの PPS は、以下となります。

192.168.250.1：270pps ＝ ①の 50+ ②の 10+ ③の 210
192.168.250.2：60pps ＝ ①の 50+ ②の 10
192.168.250.3：210pps ＝ ③の 210

● **調整方法**

デバイス帯域使用状況画面での計算結果の値が、EtherNet/IP コネクションで使用するデバイスの仕様を超える場合は、スイッチングハブの再選定、またはネットワークを分割するなど、全体のネットワーク構成を再検討してください。

また、EtherNet/IP コネクションの PPS は、パケットインターバル (RPI) 設定を大きくすればするほど小さくすることができます。

本画面では、すべてのターゲットデバイスへのコネクション設定のパケットインターバル (RPI) の値を、［パケットインターバル (RPI) 一括設定］で変更できます。

デバイスの PPS と、パケットインターバル (RPI) の関係については🕮「14-2-2 タグデータリンクの帯域使用率と RPI の関係」(P.14-10) を参照してください。

コネクション設定のデータ転送

使用上の注意

- 設定対象ノードアドレス（IP アドレス）が正しく設定されていないと、意図しないコントローラに接続して不正なデバイスパラメータを設定してしまうことがあるため、接続先コントローラを事前に確認してからダウンロードを実行してください。
- 不適切なコネクション設定が設定されると、設備が予期しない動作をする恐れがあります。また適切なコネクション設定が設定されている場合でも、設備に影響がないことを確認してから、転送を実行してください。
- タグの設定に用いたネットワーク変数が、コントローラに設定されていないと、コネクションがエラーとなります。タグの設定で使用したネットワーク変数が、コントローラに設定されていることを確認してから、コネクション設定をダウンロードするようにしてください。
- 通信障害が発生した場合、出力状態は使用される製品の仕様によります。出力装備を伴う製品を使用される場合は、通信障害時にどのような動作仕様であるかを確認の上、安全対策を施してください。
- ダウンロードが完了すると、自動的に内蔵 EtherNet/IP ポートや EtherNet/IP ユニットの通信ポートにリスタートがかかります。このリスタート動作は、設定したタグセットやコネクション情報を有効にするために必要です。リスタートがかかっても設備に支障がないことを確認してから、ダウンロードを実行してください。
- ダウンロード中に Ethernet ケーブルを抜いたり、EtherNet/IP ユニットのリセットや電源 OFF をしないでください。
- ダウンロード中は、関連するノード間での EtherNet/IP コネクションを停止します。運転モードのままダウンロードする場合は、設備に影響がないことを確認してから転送してください。

  また、EtherNet/IP コネクションを使用するラダープログラムでは、EtherNet/IP コネクション停止中、および EtherNet/IP コネクション異常中のデータ処理をインターロックするなどしてください。
- EtherNet/IP ネットワークで、デバイスの PPS がユニット許容通信帯域幅 (PPS) を超える場合は、設定どおりに EtherNet/IP コネクションが動作しない場合があります。

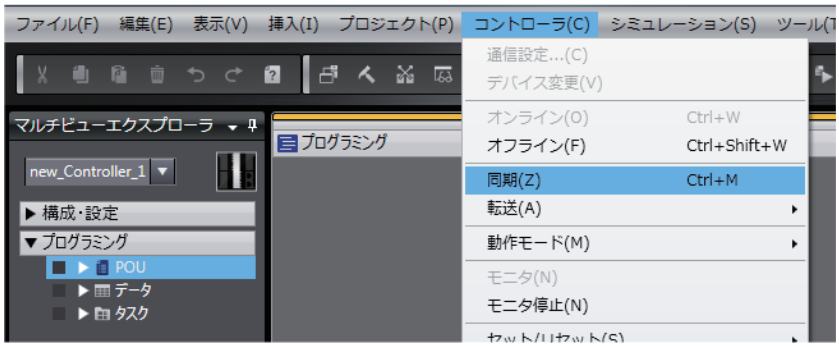
  その場合は、コネクション設定の RPI 値を大きくすることで、動作が改善できる場合があります。

● プロジェクト単位での同期・転送

EtherNet/IP コネクション設定を、プログラムデータと一緒に同期・転送／一括・転送します。

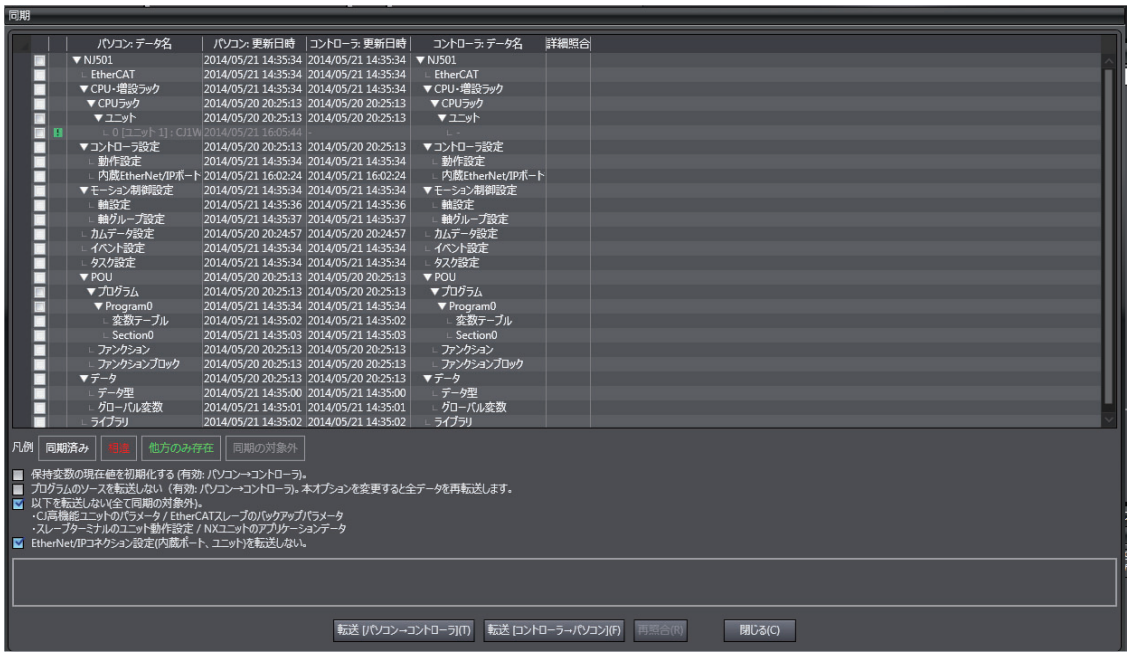
・ データを同期・転送する

1 **パソコンとコントローラをオンライン接続し、メインメニューから［コントローラ］｜［同期］を選択します。（または、ツールバーの ボタンをクリックします。）**



同期ウィンドウが表示され、Sysmac Studio とコントローラ内のプログラム、および設定データの照合が開始されます。

2 **自動照合後、以下のようにデータの転送（ダウンロード／アップロード）の画面が表示されます。**



3 **[EtherNet/IP コネクション設定（内蔵ポート、ユニット）を転送しない］のチェックを外して、［転送［パソコン→コントローラ］］ボタンをクリックしてください。**

EtherNet/IP コネクション設定が同期不一致のデータと共に転送されます。
ただし、EtherNet/IP コネクション設定がない場合は転送されません。

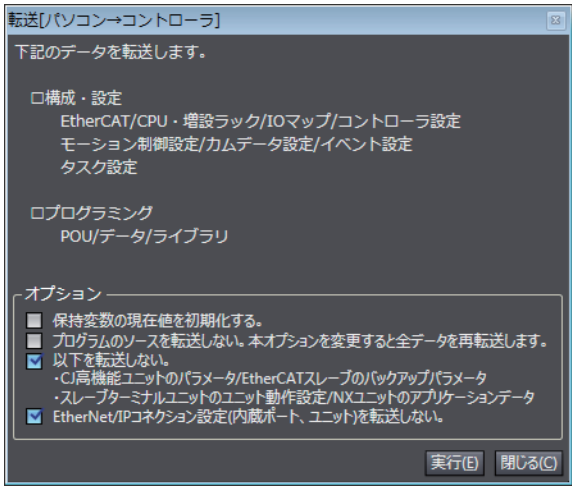
・データを一括・転送する

1 **パソコンとコントローラをオンライン接続し、メインメニューから［コントローラ］｜［転送］｜［パソコン→コントローラ］を選択します。（または、ツールバーのボタンをクリックします。）**

2 **［転送［パソコン→コントローラ］］ダイアログボックスが表示されます。**
［EtherNet/IP コネクション設定（内蔵ポート、ユニット）を転送しない］のチェックを外して、［実行］ボタンをクリックしてください。

使用上の注意

コネクション設定のみを転送したい場合は、EtherNet/IP コネクション設定画面の転送機能をお使いください。
同期画面、［転送［パソコン→コントローラ］］画面、または［転送［コントローラ→パソコン］］画面では、部分転送をしないため、パソコン側とコントローラ側のデータが一致している状態では、「コネクション設定を転送しない」オプションだけを OFF にしても、コネクション設定は転送されません。



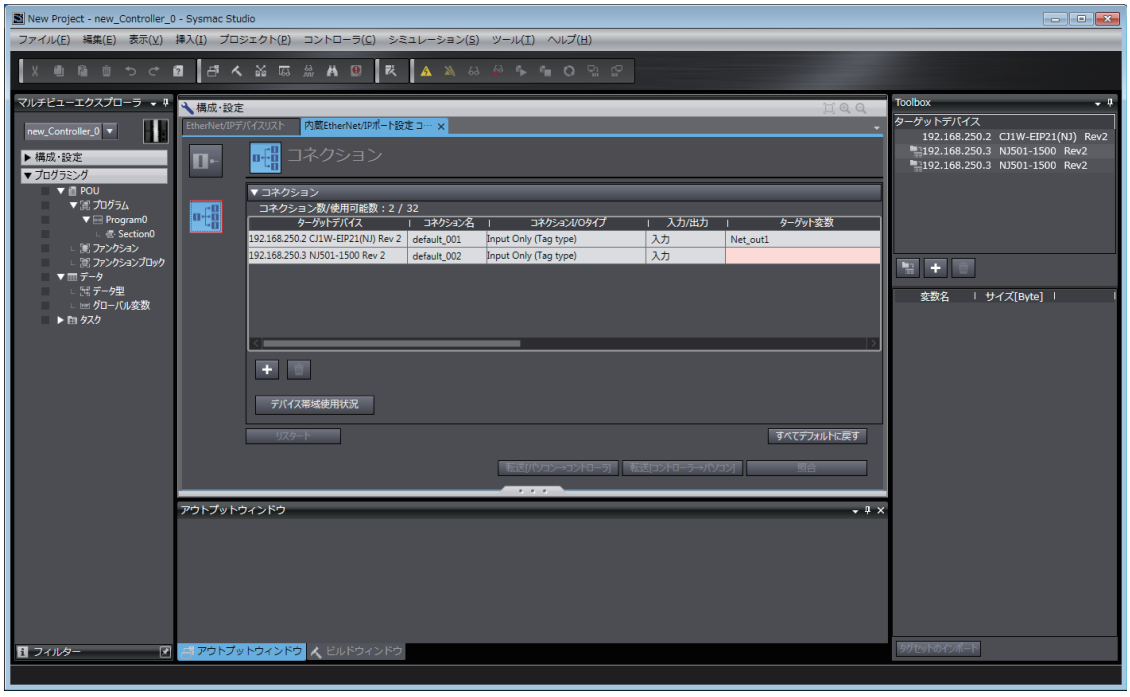
● EtherNet/IP コネクション設定のみ転送する

タグセットとコネクションを、EtherNet/IP デバイスに転送します。

1 **パソコンとコントローラをオンライン接続します。**

2 **EtherNet/IP コネクション設定画面で、［転送［パソコン→デバイス］］、または［転送［デバイス→パソコン］］ボタンをクリックします。**
その時点で設定されているタグ設定とコネクション設定がオンライン接続しているコントローラに対して転送されます。

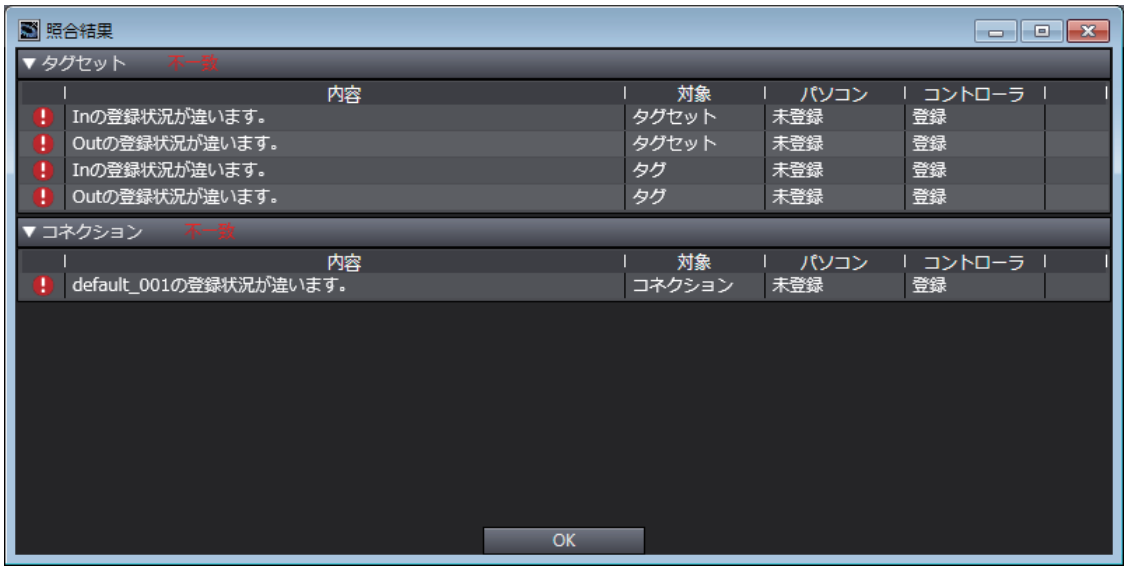
3 **オンライン接続先のコントローラが運転モードの場合、プログラムモードに変更してから転送するかの確認メッセージが表示されます。**



● 照合機能

タグセットとコネクションについて、プロジェクト側設定と、EtherNet/IP デバイス側設定の差分を表示できます。

1 **EtherNet/IP コネクション設定画面の「照合」ボタンをクリックします。**



EtherNet/IP コネクションの開始／停止

● EtherNet/IP コネクションの自動起動

EtherNet/IP コネクションは、Sysmac Studio でコネクション設定をダウンロード直後に、EtherNet/IP デバイスが自動的にリスタートし起動します。

使用上の注意

EtherNet/IP コネクション中に、オリジネータ側のコントローラで以下に示す異常要因が発生した場合は、コネクションが強制的に切断されますので、ご注意ください。
- 全停止フォールトレベルのコントローラ異常
- 部分停止フォールトレベルのコントローラ異常

● ネットワーク全体の EtherNet/IP コネクションの開始／停止

EtherNet/IP コネクションの開始／停止は、ユーザプログラムからの操作と、Sysmac Studio からの操作が可能です。

使用上の注意

EtherNet/IP コネクションの開始操作と停止操作は、同じ操作方法（ユーザプログラム、ツールソフトウエア）で実行してください。
例えば、EtherNet/IP コネクションを、システム定義変数の「_EIP_TDLinkStopCmd」（タグデータリンク通信停止スイッチ）で停止した場合、Sysmac Studio や Network Configurator の操作では、タグデータリンクは再開しません。

A-2-5　ネットワークの通信ステータス確認と処置方法

設定した EtherNet/IP コネクションの通信状態についてのモニタと、異常発生時の異常内容の確認ができます。

使用上の注意

モニタ機能は、Sysmac Studio 側とコントローラ側のコネクション設定が一致している状態でご使用ください。コネクション設定が一致しているかどうかは、「照合機能」（P.A-30）を使って確認することができます。

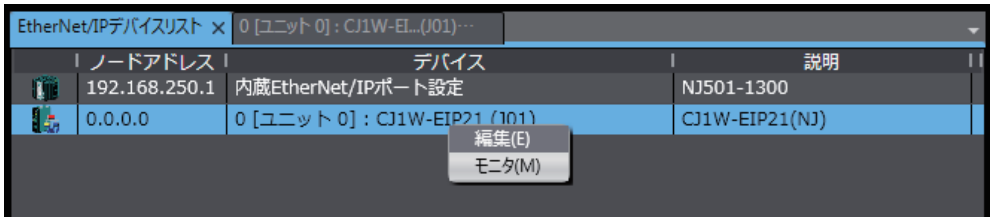
ネットワークの通信ステータス確認

EtherNet/IP コネクションモニタ画面により、EtherNet/IP コネクションの通信状態を確認できます。

1　メインメニューの［ツール］｜［EtherNet/IP コネクション設定］を選択して EtherNet/IP デバイスリスト画面を表示します。

2　通信状態を確認したいコントローラを選択し、右クリックして [モニタ] を選択します。

EtherNet/IP コネクションモニタ画面が開きます。EtherNet/IP コネクションモニタ画面では、通信状態を 6 つのタブで表示します。

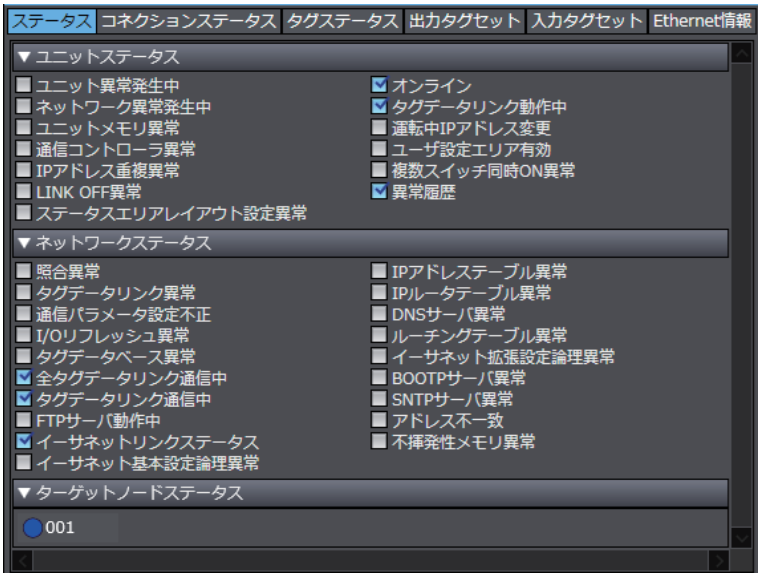


3　確認する通信状態をタブで選択します。

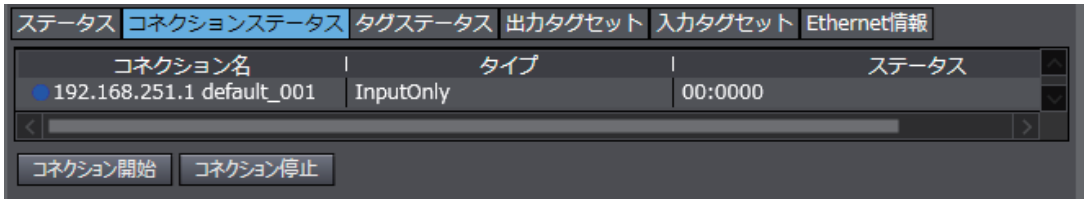
- ［ステータス］タブ

  データリンクの異常や通信状態についてのステータス状態をモニタするシステム定義変数の状態が、チェックボックスで表示され、ON 状態のステータスが ☑ マークされます。

  各ステータス項目についての詳細は、「15-1-1 Network Configurator のデバイスモニタ機能」（P.15-2）を参照ください。

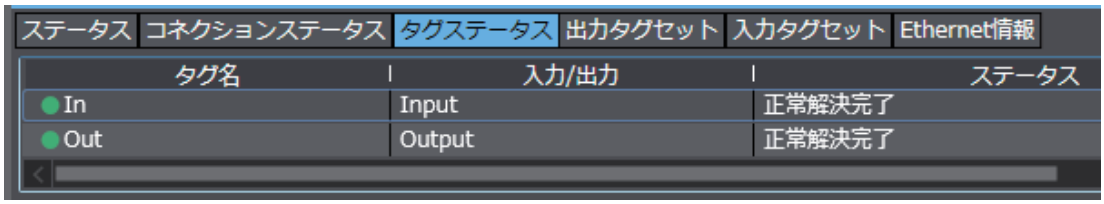


・［コネクションステータス］タブ
各コネクションの現在の状態を表示します。



項目名	説明
コネクション名	各コネクションの現在の状態を色で表示します。 緑：正常 赤：１つでも未確立のコネクションがある 灰色：コネクションがない、またはコネクションが停止中
タイプ	コネクションタイプを表示します。
ステータス	各コネクションの現在状態が、コードで表示されます。 • 正常時：“00:0000” • 異常時：異常コードが表示されます。 この情報をもとにして、EtherNet/IP コネクションの異常要因を切り分けることができます。コネクションステータス情報の詳細は、 「15-3 コネクションステータスのコード一覧と処置方法」(P.15-21) を参照してください。

・［タグステータス］タブ
EtherNet/IP コネクションの各タグが、ターゲットデバイスとの間で正しくデータ交換が行える設定になっているかを表示します。



項目名	説明
タグ名	タグ名を表示します。
入力／出力	タグの種類を表示します。
ステータス	設定されている状態により、以下のステータス表示になります。 • 正常解決完了：正常なデータ交換が行えます。 • サイズ不一致：ネットワーク変数とタグ設定でサイズが異なります。この異常が発生しているタグではコネクションは開設されません。 • タグなし：タグ設定で指定したネットワーク変数が CPU ユニットの変数テーブルに設定されていません。または、共用体型変数のメンバで指定されておらず、共用体全体で指定されています。この異常が発生しているタグではコネクションは開設されません。 • 属性異常：Const 属性のため書込みできません。

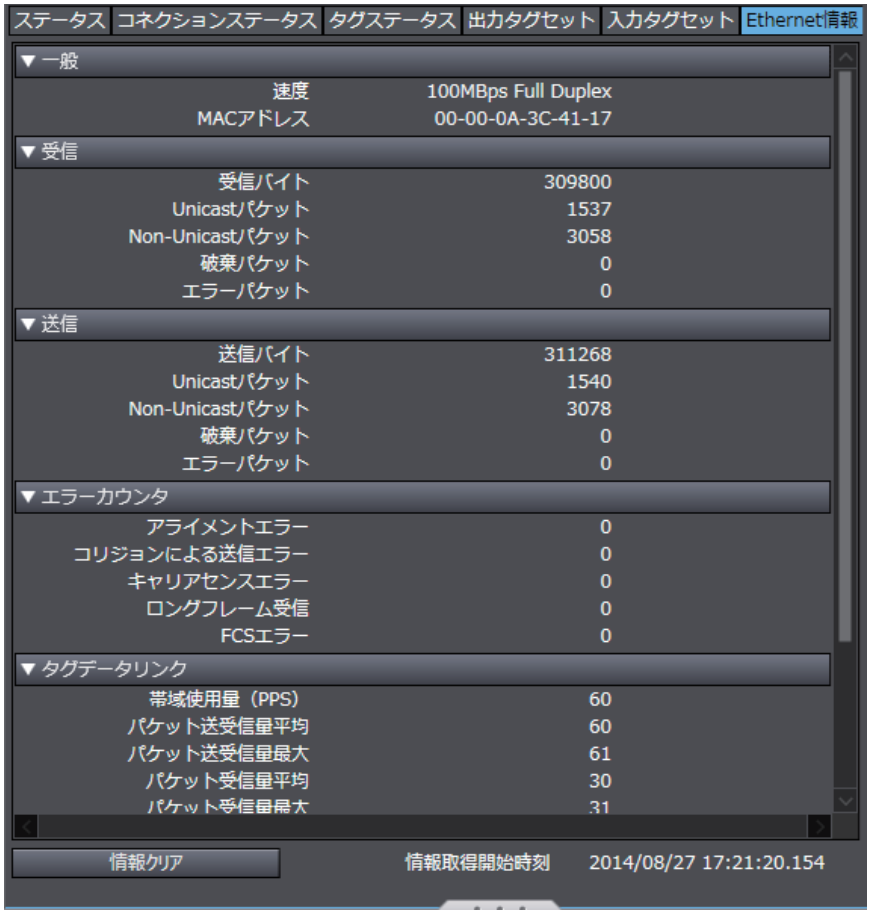
- ［出力タグセット］タブ、［入力タグセット］タブ
  EtherNet/IP コネクションで使用中の出力タグセット、および、入力タグセットの状態をそれぞれモニタ表示します。
  （注）本機能は、NJシリーズコントローラVer.1.08以前の内蔵EtherNet/IPポートでは、使用できません。
  各タグの▼をクリックして、詳細を表示させます。



項目名	説明
タグセット名	接続状態を表示します。 接続異常時は「未接続」または「異常発生中」と表示されます。
タグセットサイズ	タグセットサイズ（byte）を表示します。
接続時間	対象タグの積算接続時間（ms）を表示します。
未接続時間	対象タグの積算未接続時間（ms）を表示します。
コネクション数（［出力タグセット］タブの場合）	コネクション数を表示します。
接続オリジネータ数（［出力タグセット］タブの場合）	対象タグに接続しているオリジネータ数を表示します。
オリジネータリスト（［出力タグセット］タブの場合）、ターゲットリスト（［入力タグセット］タブの場合）	接続オリジネータの詳細情報を表示します。
オリジネータ名（［出力タグセット］タブの場合）、プロデュースタグ名（［入力タグセット］タブの場合）	何も表示されません。
IP アドレス（［出力タグセット］タブの場合）、リモート IP アドレス（［入力タグセット］タブの場合）	オリジネータデバイスの IP アドレスを表示します。
接続時間（［出力タグセット］タブの場合）	このオリジネータとの積算接続時間（ms）を表示します。
未接続時間（［出力タグセット］タブの場合）	このオリジネータとの積算未接続時間（ms）を表示します。
ディスティネーション IP アドレス（［出力タグセット］タブの場合）	ディスティネーション IP アドレスを表示します。マルチキャストコネクションの場合は、自マルチキャストアドレスを表示します。
O->T RPI	オリジネータデバイスからターゲットデバイスへのコネクションの RPI（パケットインターバル）（ms）を表示します。
T->O RPI	ターゲットデバイスからオリジネーターゲットデバイスへのコネクションの RPI（パケットインターバル）（ms）を表示します。

項目名	説明
O->T Timeout	オリジネータデバイスからターゲットデバイスへのコネクションのタイムアウト（ms）を表示します。
T->O Timeout	ターゲットデバイスからオリジネータデバイスへのコネクションのタイムアウト（ms）を表示します。
O->T API	オリジネータデバイスからターゲットデバイスへのコネクションの API（実際のパケットインターバル）（ms）を表示します。
T->O API	ターゲットデバイスデバイスからオリジネータへのコネクションの API（実際のパケットインターバル）（ms）を表示します。
O->T Connection ID	オリジネータデバイスからターゲットデバイスへのコネクションのコネクション ID（16 進数）を表示します。
T->O Connection ID	ターゲットデバイスからオリジネータデバイスへのコネクションのコネクション ID（16 進数）を表示します。

・〔Ethernet 情報 〕タブ
EtherNet/IP ユニット（内蔵ポート）の通信ドライバレベルのステータスを表示します。
各種エラーカウンタ情報により、通信障害の発生状況を確認できます。
タグデータリンク欄では、デバイスの帯域使用量（PPS）などを確認することができます。



A-2-6　トラブルシューティング

EtherNet/IP のネットワークの設定不備、通信異常が発生した場合、Sysmac Studio のトラブルシュートダイアログボックスで、異常内容が表示されます。

異常内容の確認方法と、エラー内容については、「15-5 イベントログ機能の概要」(P.15-33) を参照してください。

EtherNet/IP コネクション設定の転送やモニタができない場合の対処方法（Sysmac Studio Ver1.10 以降について）

Windows のファイアウォール機能が有効になっていると、Sysmac Studio Ver.1.10 以降で、初めてコントローラとオンライン接続したときに、接続を確認するダイアログが表示される場合があります。その場合は、以下を選択してください。

- Windows Xp/Vista：「ブロックを解除する」
- Windows7：「アクセスを許可する」

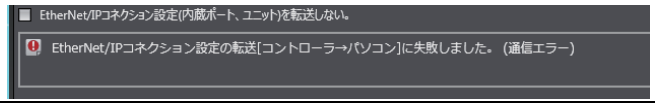
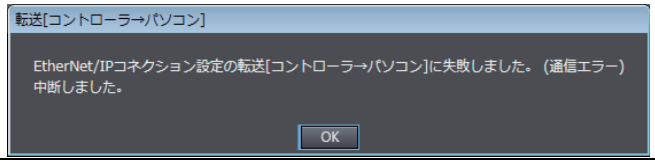
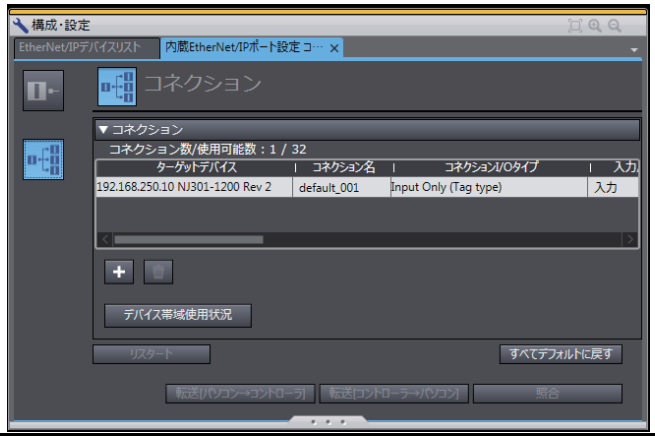
もし、上記以外を選択した場合は、Sysmac Studio Ver1.10 以降でコントローラにオンライン接続できても、EtherNet/IP コネクション設定の転送やモニタが、正常にできない現象が発生する場合があります。

そのような場合、下記の対策 1 または対策 2 を行ってください。

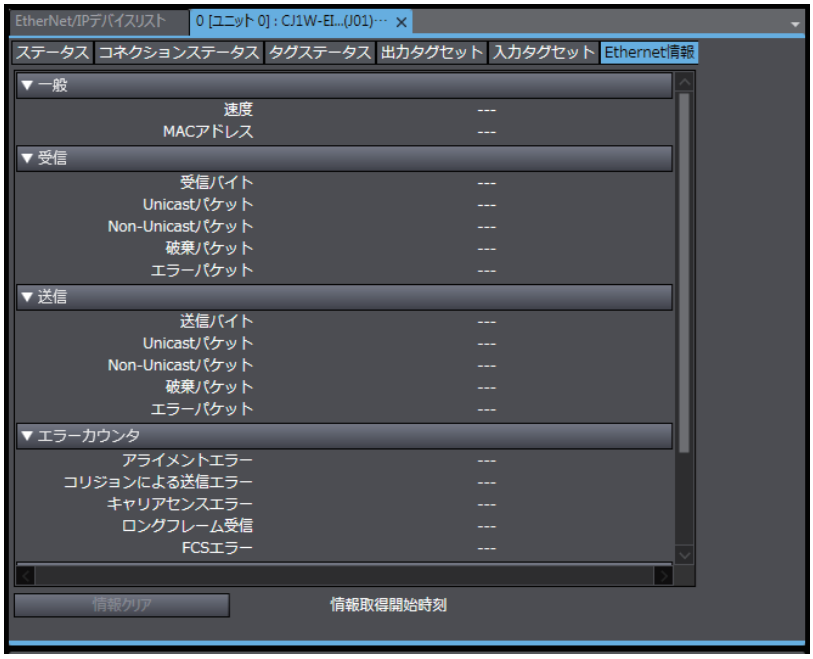
- 「対策 1（Windows のファイアウォール機能を解除する）」(P.A-37) を参照
- 「対策 2（CIP メッセージサーバ設定を［使用する］にする）」(P.A-39) を参照

● 現象

- コネクション設定のデータ転送ができない

データ転送画面	現象
同期画面	以下のエラーメッセージが表示され転送できない 
転送［パソコン→コントローラ］画面	以下のエラーダイアログが表示され転送できない。 
EtherNet/IP コネクション設定画面	［転送［パソコン→デバイス］］、または［転送［デバイス→パソコン］］ボタンがクリックできず、転送／照合できない。 

・ モニタができない
EtherNet/IP コネクションモニタ画面で、モニタ情報が「---」のまま表示されない。



● 対策 1（Windows のファイアウォール機能を解除する）

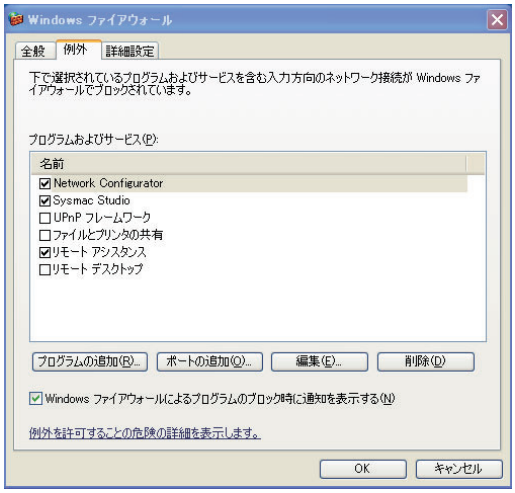
使用上の注意

ファイアウォール機能は、主に外部（インターネットなど）からの不正アクセスを防ぐためのものです。
本手順のファイアウォールの設定変更は、Sysmac Studio と NJ シリーズコントローラとの接続を可能にするためのものですが、パソコンを社内のネットワークなどで使用する場合は、設定を変更する前にセキュリティへの影響がないかご確認ください。

・ Windows Xp の場合

1 **［スタート］メニューから、［コントロールパネル］ | ［Windows ファイアウォール］を選択します。**
［Windows ファイアウォール］ダイアログボックスが表示されます。

2 **Windows ファイアフォールの［例外］タブをクリックし、［プログラムおよびサービス］の一覧にある [SysmacStudio] にチェックを入れてください。**

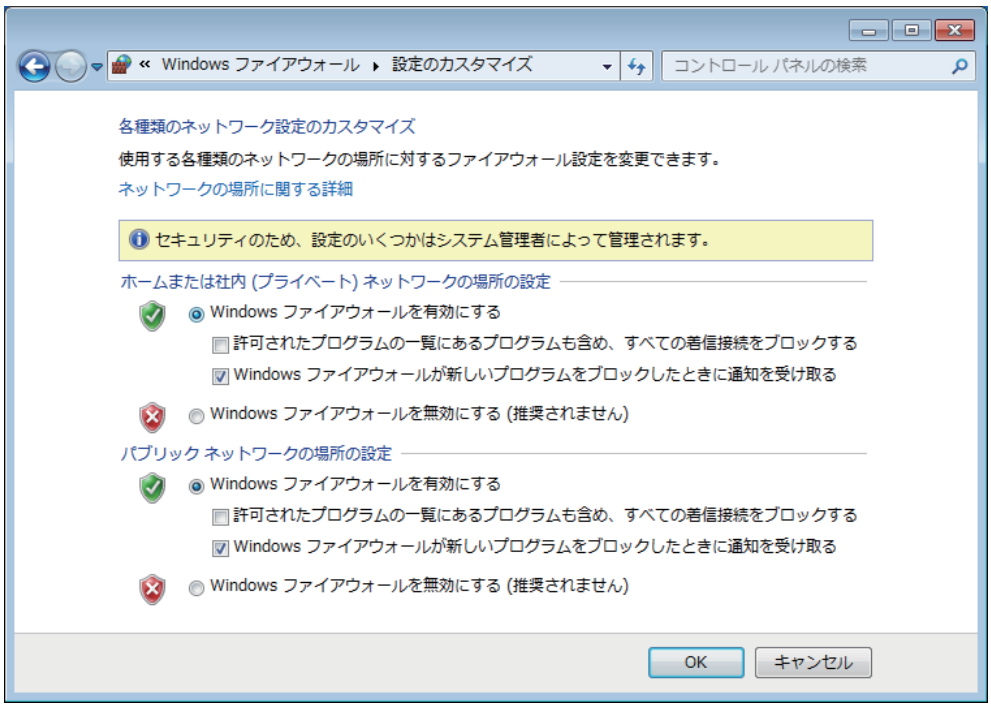


・ Windows Vista/7 以降の場合

1 **［スタート］メニューから、［コントロールパネル］ | ［Windows ファイアウォール］を選択します。**
［Windows ファイアウォール］ダイアログボックスが表示されます

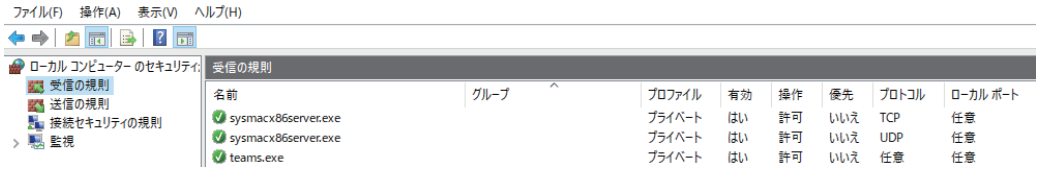
2 **［Windows ファイアウォールの有効化または無効化］を選択します。**
「設定のカスタマイズ」が表示されます。

3 **「許可されたプログラムの一覧にあるプログラムも含めすべての着信接続をブロックする」のチェックを全て外して、[OK] ボタンをクリックします。**



4 **Windows ファイアウォールの「詳細設定」をクリックします。**
「セキュリティが強化された Windows ファイアウォール」が表示されます。

5 **「受信の規則」をクリックし、「受信の規則」リストで、Sysmac Studio の Ver.1.31 以降の場合は、「sysmacx86server」をダブルクリックします。Sysmac Studio の Ver.1.31 未満の場合は、「SysmacStudio」をダブルクリックします。**
「sysmacx86server」をダブルクリックした場合は、［sysmacx86server プロパティ］画面が表示されます。「SysmacStudio」をダブルクリックした場合は、［SysmacStudio プロパティ］画面が表示されます。



6 Sysmac Studio の Ver.1.31 以降の場合は、［sysmacx86server プロパティ］画面で以下の設定を行います。Sysmac Studio の Ver.1.31 未満の場合は、［SysmacStudio プロパティ］画面で以下の設定を行います。

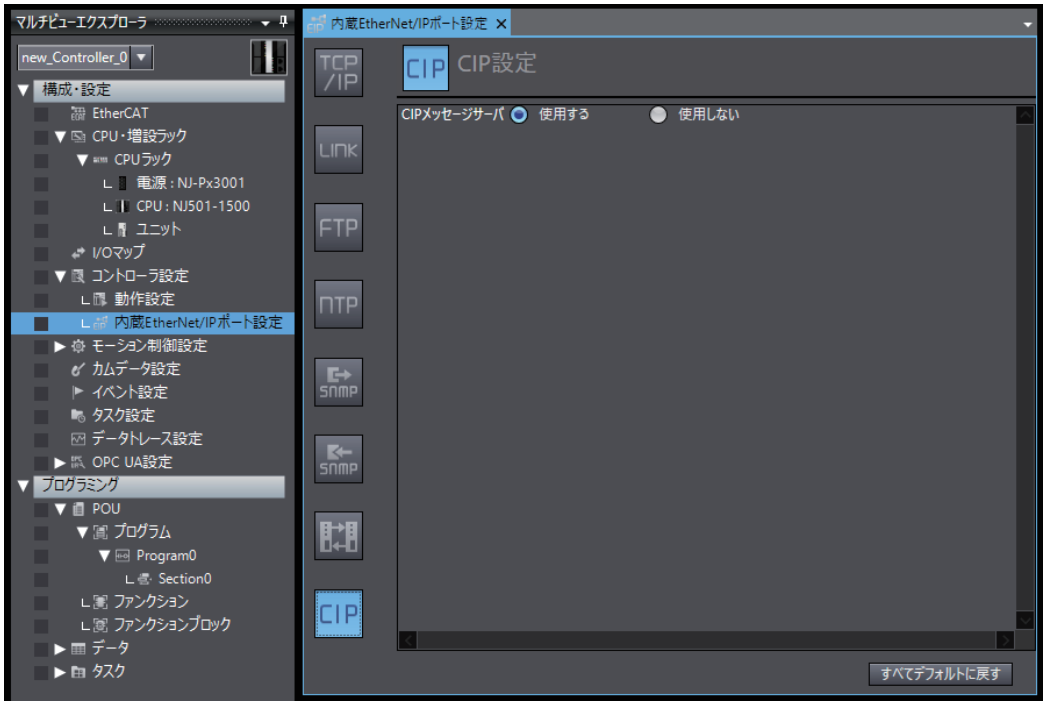
- ［詳細設定］タブの［プロファイル］:［パブリック］にチェックが入っていない場合、チェックを入れる
- ［全般］タブの［全般］:［有効］にチェックが入っていない場合、チェックを入れる
- ［全般］タブの［操作］:［接続を許可する］を選択する

● 対策 2（CIP メッセージサーバ設定を［使用する］にする）

1 Sysmac Studio をコントローラに接続します。

2 ［構成・設定］-［コントローラ設定］-［内蔵 EtherNet/IP ポート設定］-［CIP 設定］を選択します。

3 ［CIP メッセージサーバ］を［使用する］に変更します。



● 対策 3（Packet Filter 設定で、Sysmac Studio の EtherNet/IP コネクション設定が使用するパケットを許可する）

1 Sysmac Studio をコントローラに接続します。

2 ［構成・設定］-［コントローラ設定］-［内蔵 EtherNet/IP ポート設定］-［TCP/IP 設定］を選択します。

3 ［Packet Filter］で、Sysmac Studio の EtherNet/IP コネクション設定が使用するパケットを許可します。設定の詳細は、📖『NJ/NX シリーズ CPU ユニット内蔵 EtherNet/IP ポート ユーザーズマニュアル（SBCD-377）』で、「［TCP/IP 設定］ダイアログ」の「Packet Filter」を参照してください。

● 対策 4（コントローラの電源を OFF/ON する）

NJ シリーズコントローラの電源を OFF/ON し、EtherNet/IP コネクション設定の転送やモニタ操作を再度行ってください。

（注） 内蔵 EtherNet/IP ポートの IP アドレスを変更するために転送（パソコン→コントローラ）操作を行った場合に必要となる場合があります。

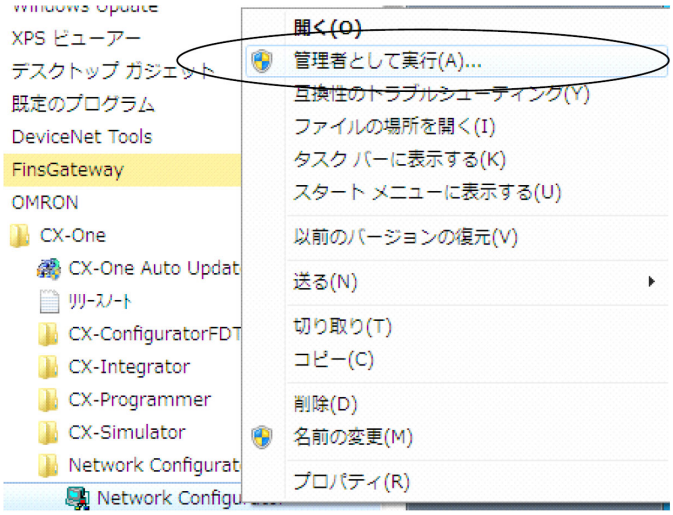
A-3 EDS ファイルの管理

ここでは、ネットワークコンフィグレータで使用する EDS ファイルの管理機能について説明します。

使用上の注意

Windows Vista /Windows7 上でご使用になる場合：
Network Configurator で、EDS ファイルについての操作を行う場合は、Network Configurator を「管理者として実行」から起動することを推奨します。
「管理者からの実行」で起動しないと、Windows セキュリティのユーザ管理の影響により、以下のようになります。
［EDS ファイル］｜［インストール］／［作成］／［削除］／［EDS インデックスファイル作成］：異なるユーザアカウントでログインしたときにこれらの操作は全て無効となりますので、再実行が必要となります。
Network Configurator を「管理者として実行」する方法は次のとおりです。

1. スタートメニューで Network Configurator を選択し、右ボタンをクリックする。
2. 表示されるポップアップメニューで、「管理者として実行」を選択する。



A-3-1 EDS ファイルのインストール

［EDS ファイル］｜［インストール］

EDS ファイルのインストールを行うことで、ネットワークコンフィグレータで新しいデバイスをサポートすることができます。
EDS ファイルをインストールするには、次の手順で行います。

1 **［EDS ファイル］｜［インストール］を選択します。**
EDS ファイルのインストールダイアログボックスが表示されます。

2 **インストールする EDS ファイルを選択し、［開く］ボタンをクリックします。**
その後、アイコンファイル（＊.ico）を選択すると、ハードウェアリストウィンドウに新しいハードウェアとして追加されます。
同一ハードウェアが既に存在する場合には、最新のものに更新されます。
また、ハードウェアのバージョンが異なる場合には、バージョンごとにハードウェアリストに追加されます。

A-3-2 EDS ファイルの作成

［EDS ファイル］｜［作成］

ネットワークコンフィグレータでネットワーク構成を作成するには、EDS ファイルは必要不可欠なものです。EDS ファイルの作成を行うには、次の手順で行います。

1 **［EDS ファイル］｜［作成］を選択します。**

2 **デバイス情報を設定します。**
オンラインの場合は、ネットワーク上のデバイスからデバイス情報を取得することができます。

3 **EDS ファイルのインストールと同様に、ハードウェアリストウィンドウに新しいデバイスとして追加されます。**



参考

ネットワークコンフィグレータの「EDS ファイル作成」機能では、デバイスパラメータの設定部分を作成することができません。デバイスパラメータの設定を行うには、デバイス製造メーカから EDS ファイルを入手してください。

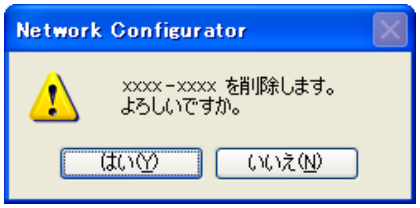
A-3-3　EDS ファイルの削除

[EDS ファイル] | [削除]

EDS ファイルを削除するには、次の手順で行います。

1 **ハードウェアリストウィンドウで対象とするハードウェア（デバイス）を選択します。**

2 **[EDS ファイル] | [削除] を選択します。**
次のような、確認ウィンドウが表示されます。



3 **[はい] ボタンをクリックします。**
ハードウェアリストウィンドウから、対象デバイスが削除され、EDS ファイルも削除されます。

A-3-4　EDS ファイルの保存

[EDS ファイル] | [保存]

EDS ファイルを保存するには、次の手順で行います。

1 **ハードウェアリストウィンドウで対象とするハードウェア（デバイス）を選択し、[EDS ファイル] | [保存] を選択します。**

2 **EDS ファイルの保存ダイアログボックスが表示されます。**

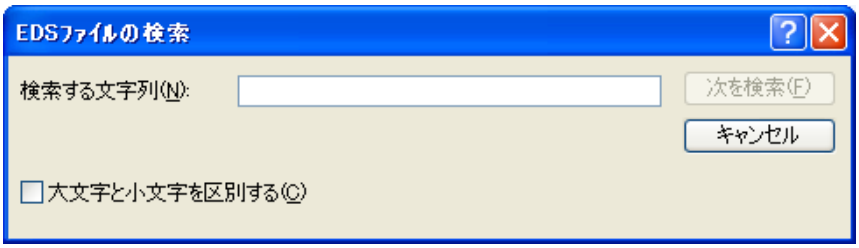
3 **フォルダ・ファイル名を指定して、[保存] ボタンをクリックします。**
ファイルへの保存が実行されます。

A-3-5　EDS ファイルの検索

［EDS ファイル］｜［検索］

ハードウェアリストウィンドウに表示されているデバイス（EDS ファイル）を検索するには、次の手順で行います。

1　**［EDS ファイル］｜［検索］を選択します。**
次のようなウィンドウが表示されます。



2　**検索文字列を設定して、［次を検索］ボタンをクリックします。**

3　**一致するデバイスがある場合は、そのデバイス位置にカーソルが移動します。**

4　**検索を終了するには、［キャンセル］ボタンをクリックします。**

参考

- 検索対象となるデバイスは、ハードウェアリストウィンドウの現在カーソル位置以降のデバイスです。
- すべてのデバイスの中から検索する場合には、ハードウェアリストウィンドウの“Hardware”を選択してから検索実行します。

A-3-6　EDS ファイルのプロパティ表示

[EDS ファイル] | [プロパティ]

EDS ファイルのプロパティを表示するには、次の手順で行います。

1 **ハードウェアリストウィンドウで対象とするハードウェア（デバイス）を選択します。**

2 **[EDS ファイル] | [プロパティ] を選択します。**
次のような、ウィンドウが表示されます。



EDS ファイルの作成日時や、デバイス情報が表示されます。

A-3-7　EDS インデックスファイルの再作成

[EDS ファイル] | [EDS インデックスファイル作成]

EDS ファイルを手動で追加した場合や、ハードウェアリストにデバイスが正しく表示されない場合、次の手順を実行して、EDS のインデックスファイルを作成し直します。
(Network Configurator V3.30 以降)

1 **[EDS ファイル] | [EDS インデックスファイル作成] を選択します。**

2 **Network Configurator を再起動します。**

A-4 Network Configurator を、WindowsXP/ WindowsVista/Windows7 で使用する際の留意点

Windows XP（SP2 以降）/WindowsVista/ Windows7 ではファイアウォールセキュリティの強化にともない、データ送受信の制限が強化されています。そのため、Network Configurator と NJ シリーズ CPU ユニットを接続して、以下の操作で通信する前に、本項に記載した方法で、Windows ファイアウォールの設定変更が必要となる場合があります。

- ［オプション］｜［インターフェースの選択］｜［Ethernet I/F］を選択するとき
- ［オプション］｜［インターフェースの選択］｜［NJ/NX Series Ethernet Direct I/F］を選択するとき
- ［オプション］｜［インターフェースの選択］｜［NJ/NX Series USB Port］を選択するとき

使用上の注意

ファイアウォール機能は、主に外部（インターネットなど）からの不正アクセスを防ぐためのものです。本手順のファイアウォールの設定変更は、Network Configurator と NJ シリーズ CPU ユニットとの接続を可能にするためのものですが、パソコンを社内のネットワークなどで使用する場合は、設定を変更する前にセキュリティへの影響がないかご確認ください。

A-4-1 Windows ファイアウォールの設定の変更手順

Windows XP の場合

1 **Network Configurator で、NJ シリーズ CPU ユニットに接続すると、[Windows セキュリティの重要な警告] ダイアログボックスが表示されます。**

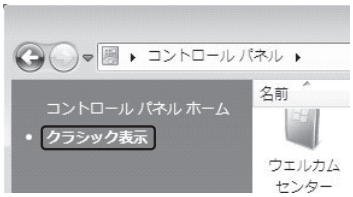
2 **［ブロックを解除する］を選択します。**

Network Configurator に USB 接続や EtherNet/IP 接続が許可された状態になり、以後、Network Configurator による接続が可能になります。

Windows Vista/Windows 7 の場合

次の手順に従って設定を行います。
オンライン接続できない場合は必ず手順 1 ～ 6 を実施してください。途中、［ユーザアカウント制御］ダイアログボックスが表示されることがあります。［続行］を選択して設定を続けてください。

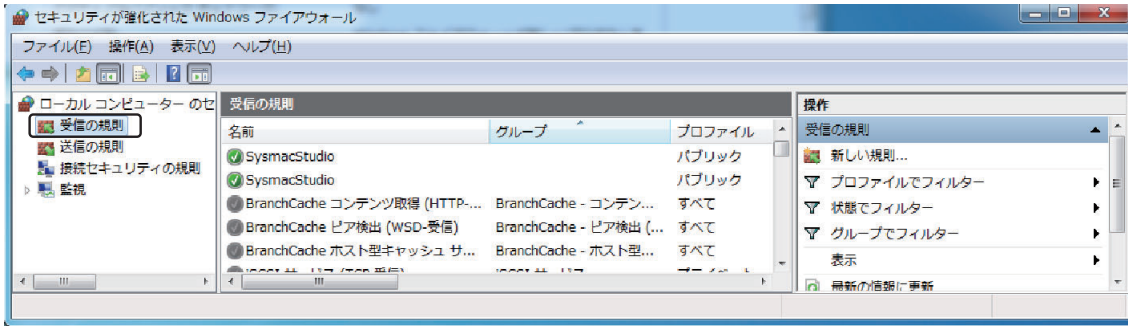
1 **スタートメニューから、［コントロールパネル］を選択し、「クラシック表示」を選択します。**



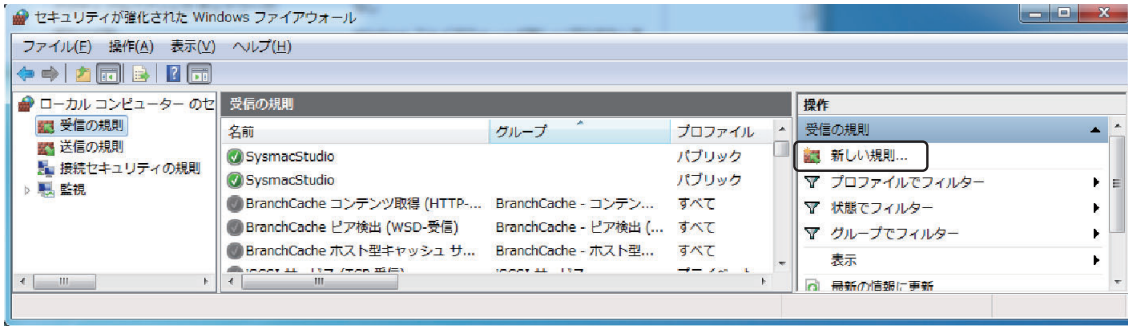
2 **「管理ツール」選択し、表示されるダイアログボックスから、「セキュリティが強化された Windows ファイアウォール」を選択します。**



3 **［セキュリティが強化された Windows ファイアウォール］ダイアログボックスの左側にある［ローカルコンピュータのセキュリティが強化された Windows ファイアウォール］ー［受信の規則］をクリックします。**



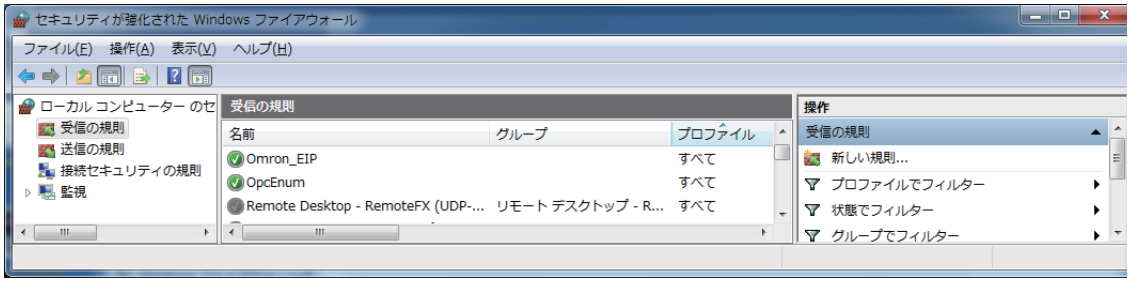
4 **ダイアログボックスの右側にある［操作］ー［受信の規則］ー［新しい規則］をクリックします。**



5 **［新規の受信の規則ウィザード］ダイアログボックスで、ステップにしたがい、各項目の設定を行います。各項目で次のように選択して［次へ］ボタンをクリックします。**

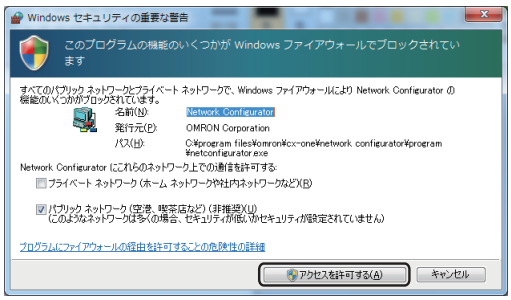
「規則の種類」	［カスタム］を選択
「プログラム」	［すべてのプログラム］を選択
「プロトコルおよびサポート」	プロトコルの種類として「ICMPv4」を選択 
「スコープ」	すべて［任意の IP アドレス］を選択
「操作」	［接続を許可する］を選択
「プロファイル」	［ドメイン］［プライベート］［パブリック］のすべてを選択
「名前」	任意の名前（例：Omron_EIP）を入力

6 **名前を入力して、［完了］ボタンをクリックすると、「受信の規則」に、作成した規則（例：Omron_EIP）が登録されます。**
［セキュリティが強化された Windows ファイアウォール］ダイアログボックスを閉じます。



7 **Network Configurator で、NJ シリーズ CPU ユニットに接続すると、[Windows セキュリティの重要な警告］ダイアログボックスが表示されます。**

8 **［ブロックを解除する］を選択します。**



Network Configurator に USB 接続や EtherNet/IP 接続が許可された状態になり、以後、Network Configurator による接続が可能になります。

A-5 変数のメモリ確保の方法について

変数のメモリ確保の方法は、構造体型や共用体型変数のメンバのメモリ上の位置を、他の機器と合わせる場合に意識する必要があります。主には、構造体型や共用体型変数を使って、他の機器と以下のような通信を行う場合に調整が必要です。

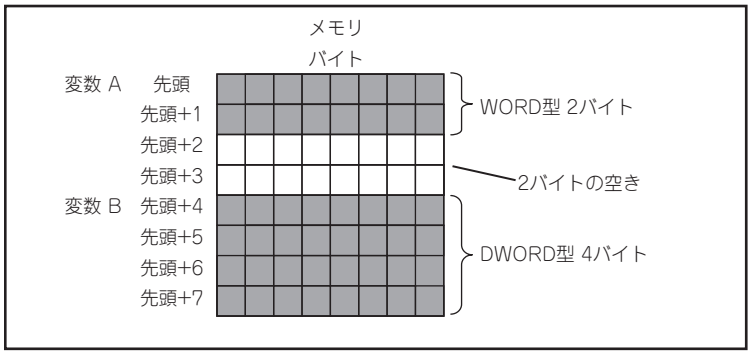
- NJ シリーズ CPU ユニットと他 CPU ユニット間で、EtherNet/IP のタグデータリンクや、CIP メッセージによる変数アクセスを行うとき
- ID タグなどの CPU ユニット以外の機器と、構造体型や共用体型変数でデータ交換を行うとき

A-5-1 変数のメモリ確保の規則

変数のために確保されるメモリの量とメモリ上の位置は、その変数のデータ型によって変わります。また、配列の要素、構造体のメンバ、共用体のメンバのために確保されるメモリの量とメモリ上の配置は、それらのデータ型に加えて、配列、構造体、共用体の宣言内容によって変わります。

データ型のアライメントと、確保されるメモリの量

データ型は、それぞれデータサイズが決められています。データサイズとは、そのデータ型の値を格納するために最低限必要なメモリの量を意味します。
一方、変数に対応するメモリは、最も効率よくアクセスできるようにコントローラにより自動的に配置されます。このため、変数のために確保されるメモリの総量と、それらの変数のデータサイズの合計とは必ずしも一致しません。たとえば、WORD 型と DWORD 型の変数を宣言した場合、データサイズの合計は 6 バイトですが、メモリは下図のように 8 バイト確保されます。



変数テーブル

名称	データ型
A	WORD
B	DWORD

このような、メモリ上の変数の配置が決められるための情報が、アライメントです。アライメントはデータ型ごとに決まっており、その変数のために確保されるメモリの量とメモリ上の位置は、次のとおりです。

項目	仕様
確保されるメモリの量	アライメントの整数倍。ただし、データサイズ以上。
メモリ上の位置	メモリ先頭から、アライメントの整数倍の位置。

基本データ型と列挙型の、アライメントと確保されるメモリの量は、以下のとおりです。

データ型	アライメント [バイト]	確保されるメモリの量 [バイト]
BOOL	2	2
BYTE,USINT,SINT	1	1
WORD,UINT,INT	2	2
DWORD,UDINT,DINT	4	4
LWORD,ULINT,LINT	8	8
REAL	4	4
LREAL	8	8
TIME,DATE,TIME_OF_DAT,DATE_ANT_TIME	8	8
STRING[N+1]*1	1	N+1
列挙型	4	4

*1. N は、取り扱う最大文字数を表します。たとえば、半角英数字の最大文字数が 10 の場合、文字列終端の NULL を加えた、11 文字分のメモリを確保する必要があります。

配列、構造体型、共用体型は、それらの各要素や各メンバも、それぞれが最も効率よくアクセスできるようにメモリ上に配置されます。配列、構造体型、共用体型のアライメントと、確保されるメモリの量は、その宣言によって以下のように決まります。

データ型	アライメント	確保されるメモリの量
配列	要素のデータ型のアライメントと同じ	（要素のデータ型のために確保されるメモリの量）× 要素数*1
構造体型	すべてのメンバのアライメントのうち、最大の値	各メンバを、そのメンバのデータ型のアライメントの整数倍の位置に、順番に配置した際に確保されるメモリの総量以上、かつアライメントの整数倍
共用体型	すべてのメンバのアライメントのうち、最大の値	すべてのメンバの、確保されるメモリの量のうち、最大の値

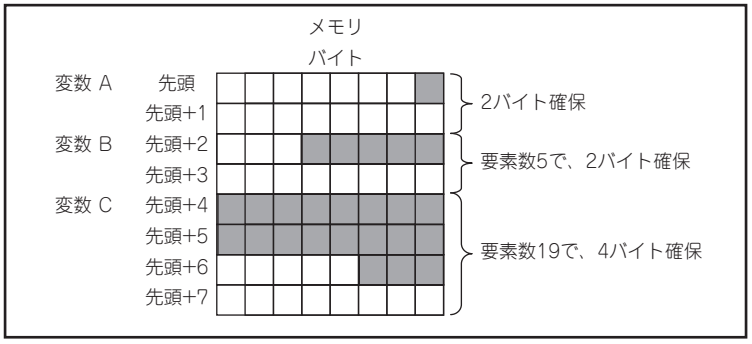
*1. BOOL 型の配列の場合は、例外です。BOOL 型の配列のために確保されるメモリの量については、後の使用上の注意を参照してください。

使用上の注意

BOOL 型の配列のために確保されるメモリの量

BOOL 型の、単独の変数、構造体のメンバ、共用体のメンバのために確保されるメモリの量は、2 バイトです。

しかし、BOOL 型の配列は、1 要素に 2 バイトずつのメモリが確保されるわけではありません。各要素に対応するメモリが 1 ビットずつ並んで確保されます。そして、配列全体としては、空きも含めて 2 バイトの整数倍のメモリが確保されます。



変数テーブル

名称	データ型
A	BOOL
B	ARRAY [1..5] OF BOOL
C	ARRAY [0..18] OF BOOL

したがって、BOOL 型の配列のために確保されるメモリの量は、以下の式で求められます。要素数が 1 ～ 16 の場合は 2 バイト、17 ～ 32 の場合は 4 バイトのメモリが確保されます。

$$\text{メモリの量} = 2\left[\frac{\text{要素数}-1}{16}\right]+2$$

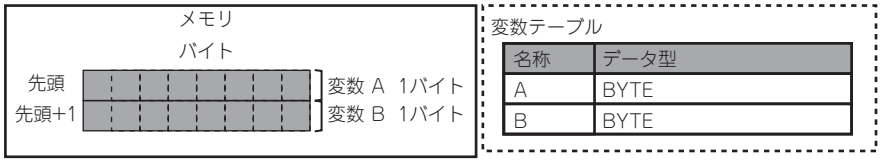
括弧内の計算結果の小数点以下は切り捨て

以下、各データ型の変数のメモリ確保の規則を、具体例を用いて説明します。

基本データ型

● BYTE 型などのアライメントが 1 バイトの変数

1 バイト単位のアライメントで、1 バイトのメモリが確保されます。
例）BYTE 型の変数が、2 個連続している場合

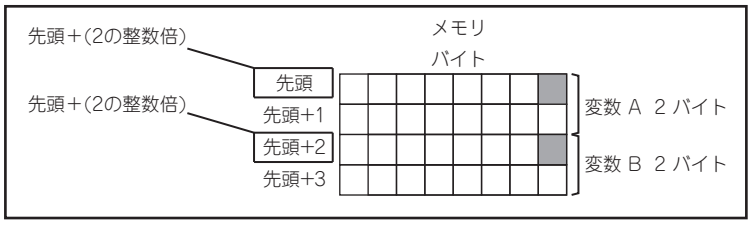


変数テーブル

名称	データ型
A	BYTE
B	BYTE

● BOOL 型、WORD 型などの、アライメントが 2 バイトの変数

2 バイト単位のアライメントで、2 バイトのメモリが確保されます。
例）BOOL 型の変数が、2 個連続している場合

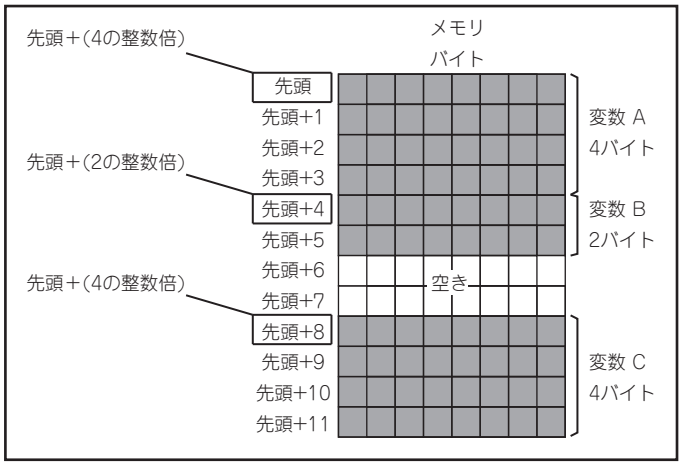


変数テーブル

名称	データ型
A	BOOL
B	BOOL

● DWORD 型などの、アライメントが 4 バイトの変数

4 バイト単位のアライメントで、4 バイトのメモリが確保されます。
データがメモリに配置される先頭位置が、4 バイトの整数倍になるので、WORD 型などのアライメントが 2 バイトの変数が挿入されると、2 バイト分空きが発生します。
例）DWORD 型、WORD 型、DWORD 型の順で、変数が連続している場合

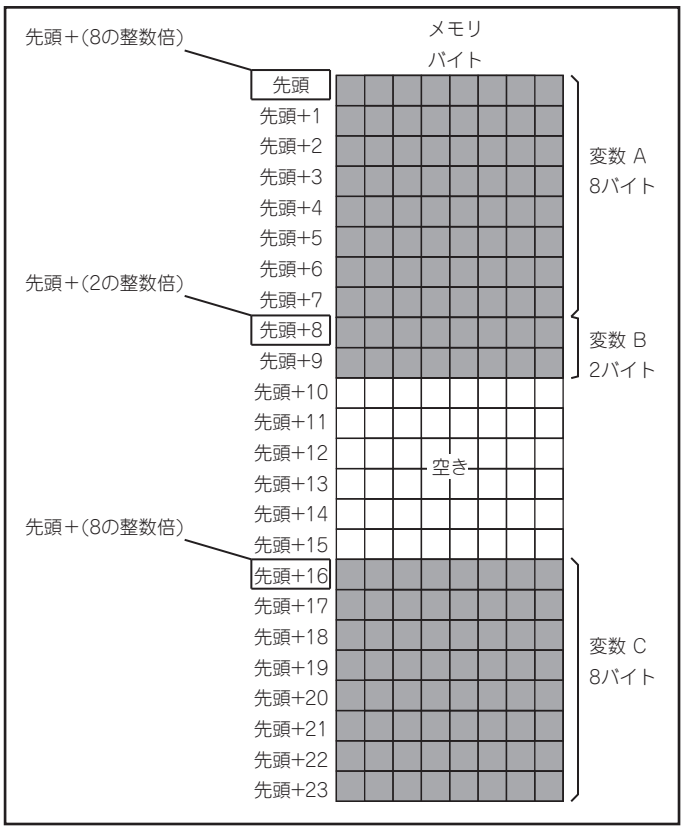


変数テーブル

名称	データ型
A	DWORD
B	WORD
C	DWORD

● LWORD 型などの、アライメントが 8 バイトの変数

8 バイト単位のアライメントで、8 バイトのメモリが確保されます。
データがメモリに配置される先頭位置が、8 バイトの整数倍になるので、WORD 型などのアライメントが 2 バイトの変数が挿入されると 6 バイト分、DWORD 型などのアライメントが 4 バイトの変数が挿入されていると 4 バイト分の、空きが発生します。
例）LWORD 型、WORD 型、LWORD 型の順で、変数が連続している場合



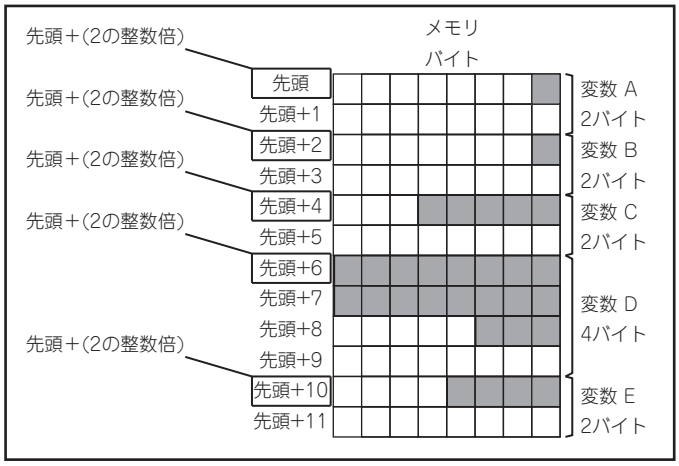
変数テーブル

名称	データ型
A	LWORD
B	WORD
C	LWORD

配列

配列変数は、配列要素のデータ型のデータサイズを単位として、要素数分だけ連続してメモリが確保されます。配列のアライメントは、要素のデータ型のアライメントと同じです。

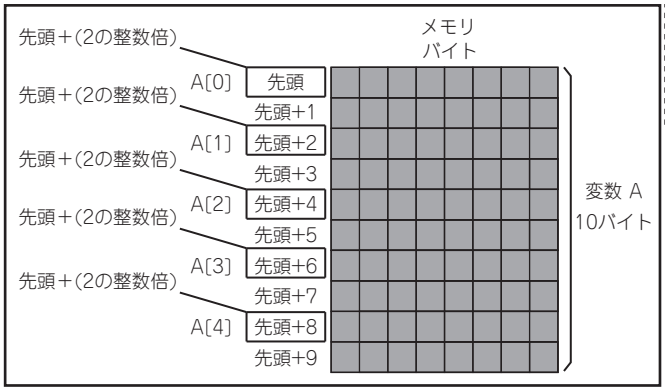
例）BOOL 型変数が 2 個、要素数 5 の BOOL 型配列、要素数 19 の BOOL 型配列、要素数 4 の BOOL 型配列の順で、変数が連続している場合



変数テーブル

名称	データ型
A	BOOL
B	BOOL
C	ARRAY [1..5] OF BOOL
D	ARRAY [0..18] OF BOOL
E	ARRAY [5..8] OF BOOL

例）要素数 5 の INT 型配列の場合



変数テーブル

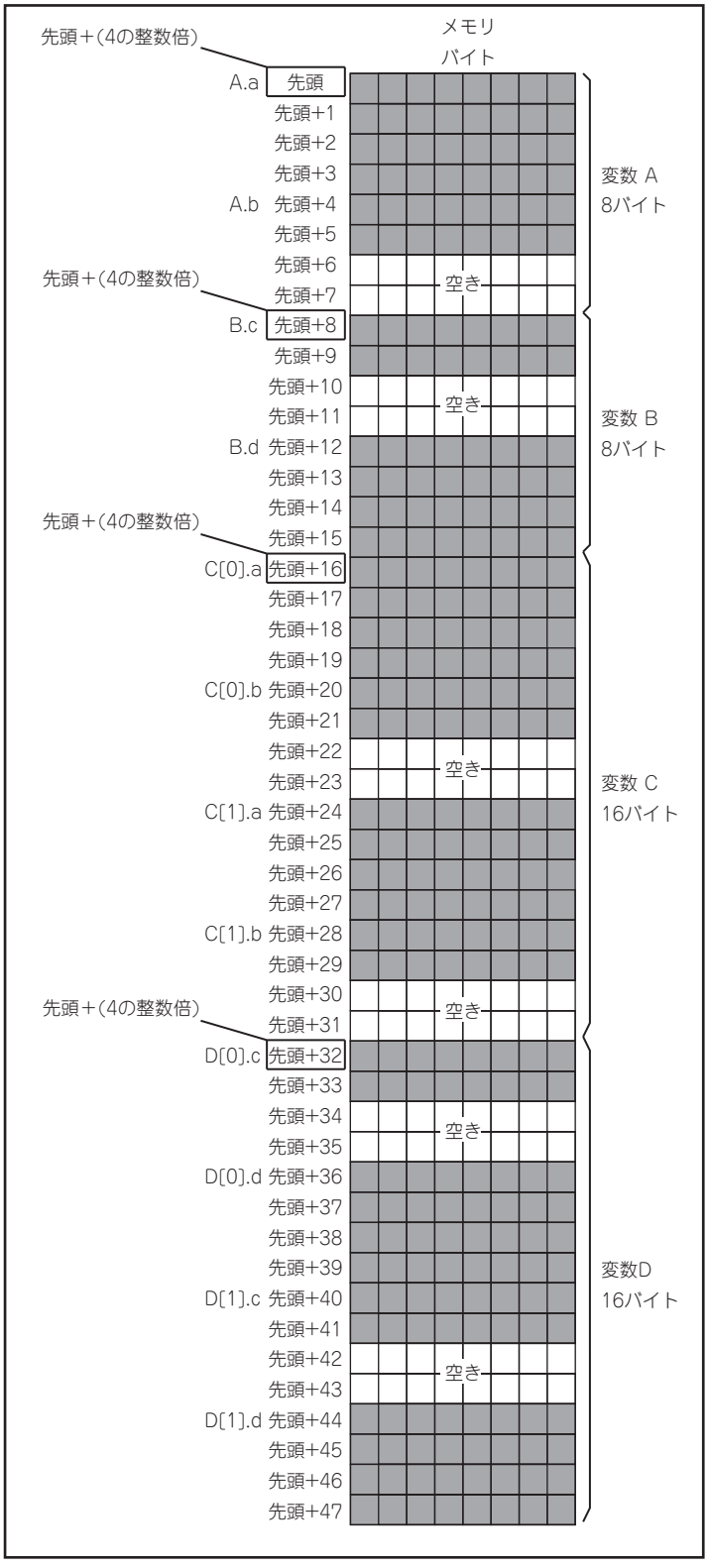
名称	データ型
変数 A	ARRAY[0..4] OF INT

構造体型

構造体変数は、各メンバが宣言された順番にメモリに配置されます。また、各メンバが、そのメンバのデータ型のアライメントの整数倍の位置に配置されます。したがって、メンバ間やメンバの終端にメモリの空きが挿入される場合があります。構造体型のアライメントは、すべてのメンバのアライメントのうちの最大の値です。確保されるメモリの量は、各メンバを、そのメンバのデータ型のアライメントの整数倍の位置に順番に配置した際に確保されるメモリの総量以上で、かつアライメントの整数倍になります。

例）下図の 4 つの変数宣言の場合、それぞれのアライメントと、確保されるメモリの量は、以下の値になります。

変数	アライメント [バイト]	確保されるメモリの量 [バイト]
A	4	8
B	4	8
C	4	16
D	4	16



データ型定義

名称	データ型
構造体 STR_A	STRUCT
a	DINT
b	INT

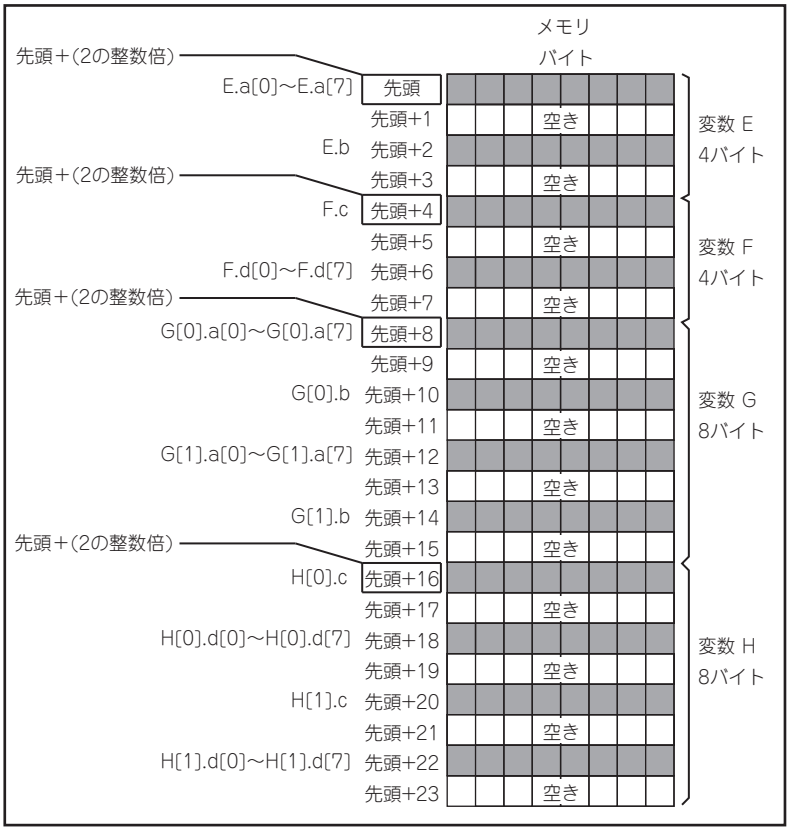
名称	データ型
構造体 STR_B	STRUCT
c	INT
d	DINT

変数テーブル

名称	データ型
変数 A	構造体 STR_A
変数 B	構造体 STR_B
変数 C	ARRAY[0..1] OF STR_A
変数 D	ARRAY[0..1] OF STR_B

例）下図の 4 つの変数宣言の場合、それぞれのアライメントと、確保されるメモリの量は、以下の値になります。

変数	アライメント [バイト]	確保されるメモリの量 [バイト]
E	2	4
F	2	4
G	2	8
H	2	8



データ型定義

名称	データ型
構造体 STR_C	STRUCT
a	ARRAY[0..7] OF BOOL
b	BYTE

名称	データ型
構造体 STR_D	STRUCT
c	BYTE
d	ARRAY[0..7] OF BOOL

変数テーブル

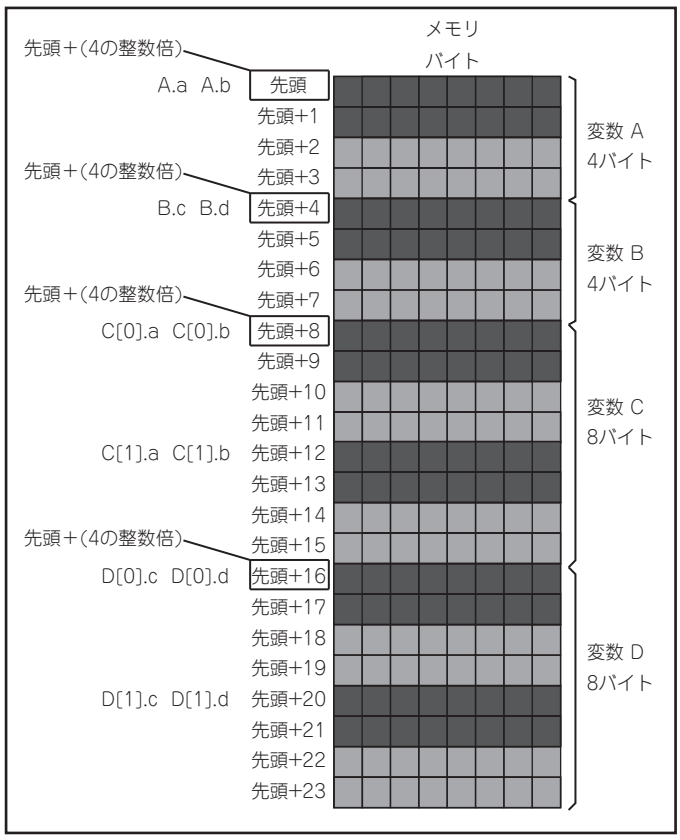
名称	データ型
変数 E	構造体 STR_C
変数 F	構造体 STR_D
変数 G	ARRAY[0..1] OF STR_C
変数 H	ARRAY[0..1] OF STR_D

共用体型

共用体型変数は、各メンバがメモリ上の同じ位置に重なって配置されます。
共用体型のアライメントは、すべてのメンバのアライメントのうちの最大の値です。確保されるメモリの量は、すべてのメンバの、確保されるメモリの量のうちの最大の値です。

例）下図の 4 つの変数宣言の場合、それぞれのアライメントと、確保されるメモリの量は、以下の値になります。

変数	アライメント [バイト]	確保されるメモリの量 [バイト]
A	4	4
B	4	4
C	4	8
D	4	8



データ型定義

名称	データ型
共用体 UNI_A	UNION
a	DWORD
b	WORD

名称	データ型
共用体 UNI_B	UNION
c	WORD
d	DWORD

変数テーブル

名称	データ型
変数 A	共用体 UNI_A
変数 B	共用体 UNI_B
変数 C	ARRAY[0..1] OF UNI_A
変数 D	ARRAY[0..1] OF UNI_B

A-5-2　留意するべきケース例

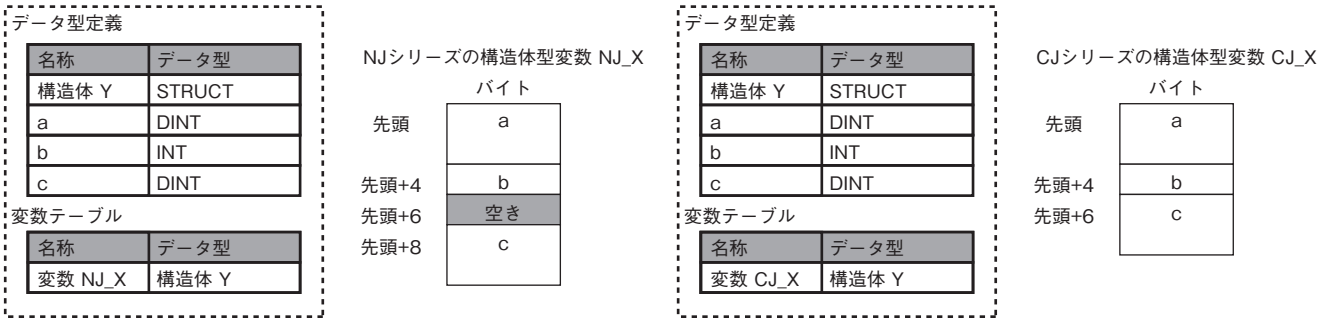
NJ シリーズ CPU ユニットの構造体型変数で、他の機器との通信などでデータ交換するときには、構造体型変数のメンバのメモリ配置を、通信する相手機器と一致させる必要があります。
その場合は、NJ シリーズ CPU ユニット側、または通信相手の機器側か、いずれかで本項目で説明する対策を行ってください。

参考

NJ シリーズ CPU ユニット同士でデータ交換する場合は、特にメンバのメモリ配置を意識する必要はありません。

通信する相手機器とメモリ配置を一致させる方法

通信する相手機器とメモリ配置を一致させる方法には、2 つの方法があります。
例えば、NJ シリーズ CPU ユニットと CJ シリーズの CPU では、構造体型変数のメモリ構成が、以下のように異なります。
本項では、そのメモリ配置を一致させる方法を例に説明します。



- **対策 1：NJ シリーズ CPU ユニットの構造体型変数のメンバのメモリ配置を変更する**

  NJ シリーズ CPU ユニットでは、メンバのオフセット指定によって、構造体型変数ごとにメンバのメモリ配置を設定できます。NJ シリーズ CPU ユニット側で、構造体型変数のメンバのメモリ配置を、通信する相手機器にあわせて変更して調整します。
  構造体型変数のメンバのオフセット指定は、構造体型の登録時に行います。
  通信する相手機器が CJ シリーズ CPU ユニットの場合、オフセット種別を「CJ」と選択することで自動的に CJ のメモリ配置にすることができます。
  また、オフセット種別を「任意」と選択することでオフセットを自由に決めることができます。

バージョン関連情報

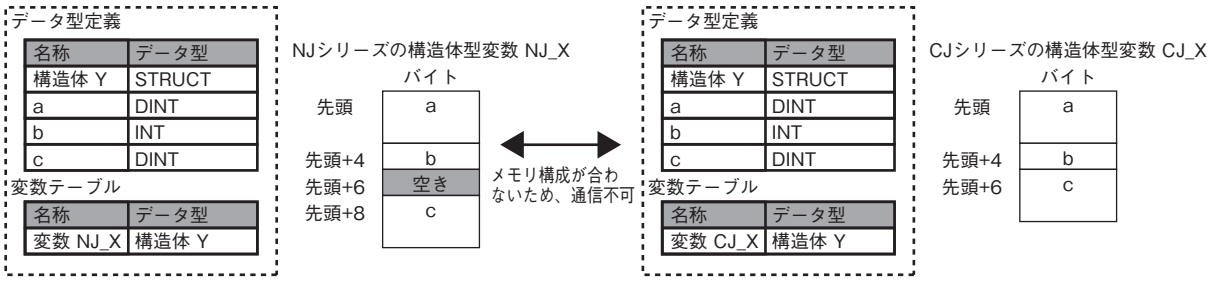
メンバのオフセット指定の使用可否は、CPU ユニットのユニットバージョンと Sysmac Studio のバージョンの組み合わせで、以下のように変わります。

CPU ユニットのユニットバージョン	Sysmac Studio のバージョン		
	Ver.1.01 以前	Ver.1.02	Ver.1.03 以降
Ver.1.01 以降	不可	可 *1	可
Ver.1.00	不可	不可	不可

*1 オフセット種別の選択機能はありません。メンバのオフセットは、任意に設定します。

本設定で構造体型変数のメモリ配置を調整した場合は、ネットワーク上にある他の NJ シリーズ CPU ユニットにも同様の設定変更を行ってください。
構造体型変数のメモリ配置の操作方法については、🕮『Sysmac Studio Version 1 オペレーションマニュアル（SBCA-470）』を参照してください。

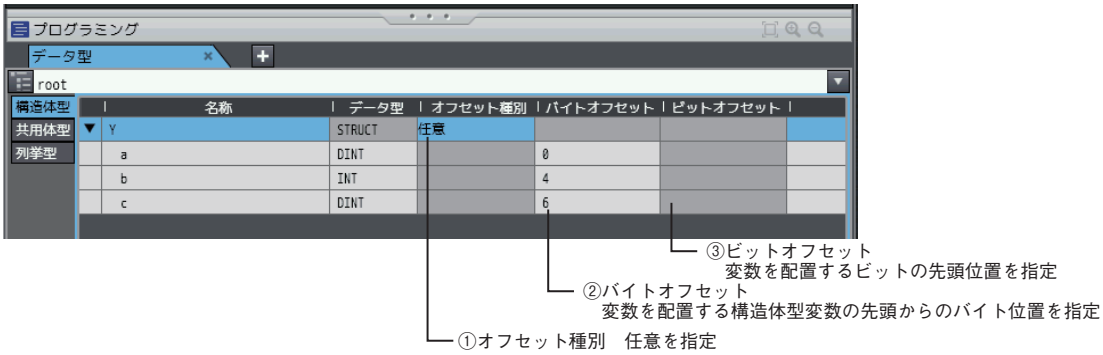
例）NJ シリーズ CPU ユニットの構造体型変数のメンバのメモリ配置を、CJ シリーズ CPU ユニットの構造体型変数のメンバにあわせて変更する

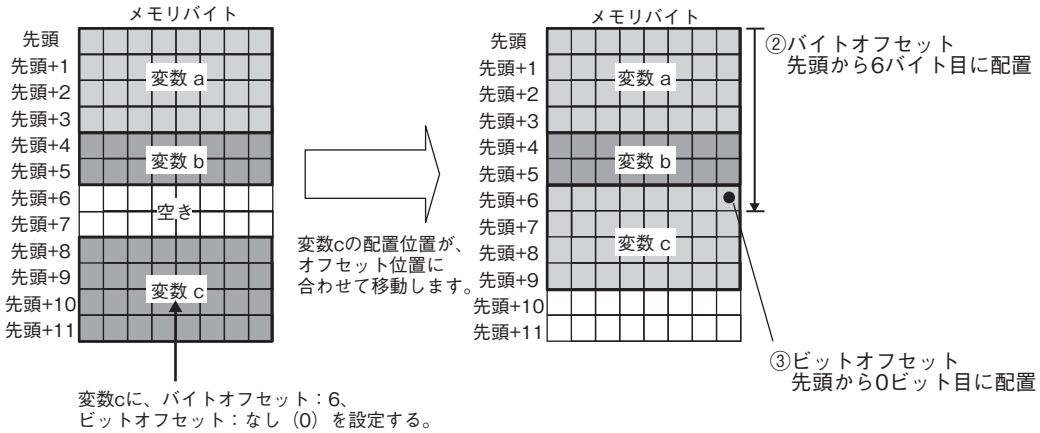


NJシリーズCPUユニットと、CJシリーズCPUユニットとのメモリ構成を合わすため、
Sysmac Studioでオフセット指定します。

構造体型変数 NJ_Xのデータ型Yのメンバcに、以下のオフセット値を指定します。



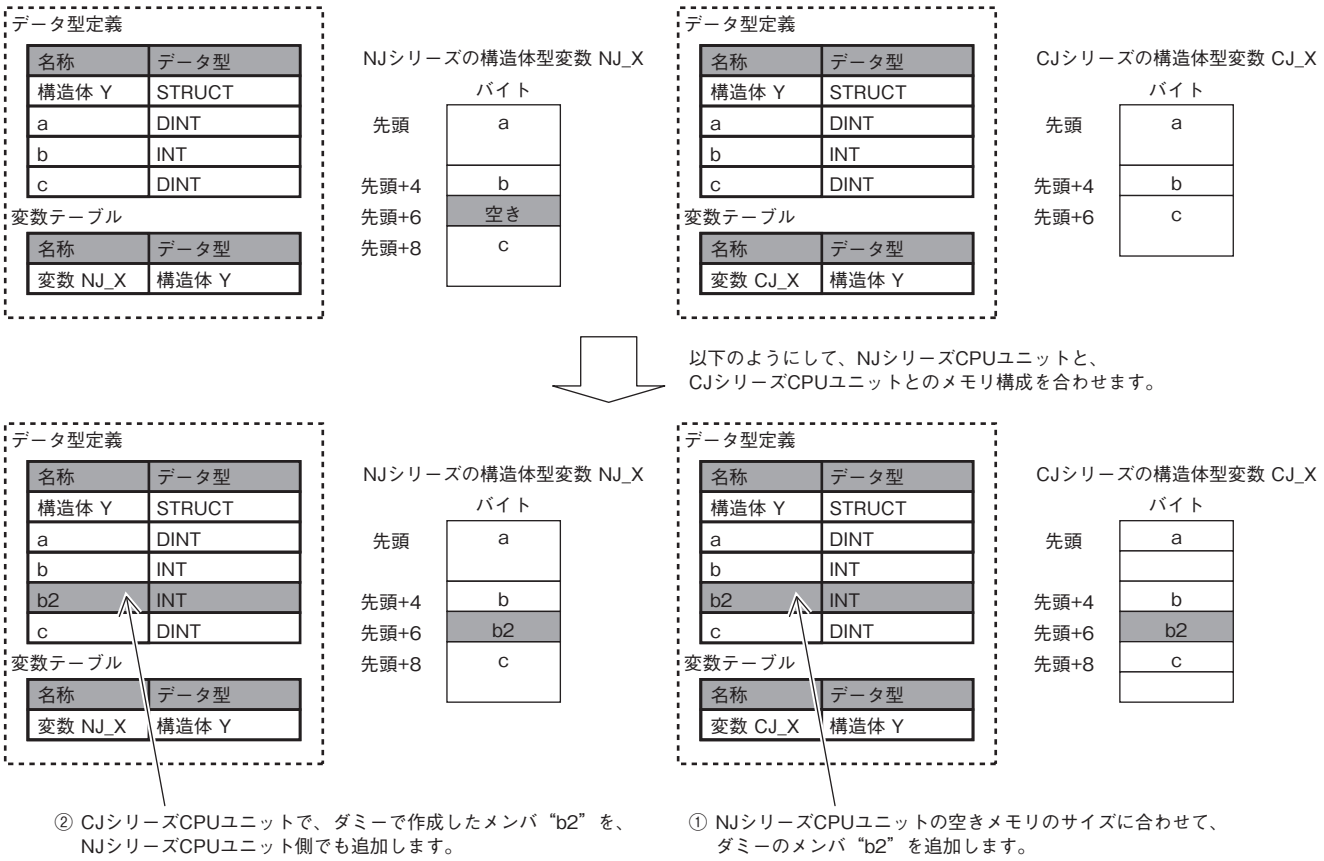




● 対策 2：通信する相手機器の構造体型変数のメモリ配置を変更する

NJ シリーズ CPU ユニットの構造体型変数のメモリ配置にあうように、通信する相手機器の構造体型変数のメモリ配置を、配置調整用のメンバなどを挿入して調整します。
また、メモリ配置だけでなく、データ型も一致させる必要があるので、配置調整用のメンバは、通信する相手機器、NJ シリーズ CPU ユニットともに作成する必要があります。

例）CJ シリーズ CPU ユニットの構造体型変数のメモリ配置を、NJ シリーズ CPU ユニットの構造体型変数のメモリ配置にあわせて変更する



A-6 CPU ユニットの外部出力にアクセスする場合の留意事項

NJ シリーズ CPU ユニットの外部出力に割り付けられている、変数、I/O メモリアドレスにアクセスする場合、以下の留意事項がありますので、ご注意ください。

- 外部出力に割り付けられている変数に対して、外部から書き込んだ場合の注意
  NJ シリーズ CPU ユニットの外部出力に割り付けられている変数に対して、タグデータリンクや通信命令などでアクセスして状態を変更しても、変更した値は、ユーザプログラムの実行結果で上書きされます。
  よって、タグデータリンクや通信命令などからのアクセスにより変更した変数の状態で、外部に出力
  することはできません。

外部出力に割り付けられている変数には、以下の種類があります。

- CJ シリーズ基本出力ユニットの I/O ポートに割り付けたデバイス変数（グローバル変数）
- EtherCAT 出力スレーブの I/O ポートに割り付けたデバイス変数（グローバル変数）
- CJシリーズ基本出力ユニットが割り付けられる出力リレーにAT指定しているグローバル変数

- CJ シリーズ基本出力ユニットの出力リレーに相当する I/O メモリアドレスに対して、直接外部から書き込んだ場合の注意
  タグデータリンクにより、CJ シリーズ基本出力ユニットが割り付けられる出力リレーに相当する I/O メモリアドレスに直接アクセスし、I/O メモリアドレスの状態を変更しても、変更した値はユーザプログラムの実行結果で上書きされます。
  よって、タグデータリンクによって、直接 I/O メモリアドレスに書き込んだ値で、外部に出力することはできません。

A-7 接続する CPU ユニット（NJ/CJ シリーズ）による相違点

本ユニットは、NJ シリーズ CPU ユニットに接続して使用する場合と、CJ シリーズ CPU ユニットに接続して使用する場合とで、一部仕様が異なります。以下に、それらの相違点を示します。

A-7-1 機能の相違点

本ユニットを NJ シリーズ CPU ユニットに接続して使用する場合、CJ シリーズ CPU ユニットに接続して使用する場合に対して、使用できなくなる、機能名称が異なる、仕様が異なるなどの相違がある機能があります。

使用できなくなる機能

以下の機能は、NJ シリーズ CPU ユニットに接続する場合には、使用できません。
- Network Configurator の EtherNet/IP データリンクツール機能
- サポートソフト（Sysmac Studio）による EtherNet/IP ユニット経由での CPU オンライン接続（Nerwork configurator によるオンライン接続は可能です）

名称が異なる機能

以下の機能は、NJ シリーズ CPU ユニットに接続して使用する場合と、CJ シリーズ CPU ユニットに接続して使用する場合とで、機能名称が異なります。対応する機能名称は、以下のとおりです。

項目	CJ シリーズ CPU ユニットに接続	NJ シリーズ CPU ユニットに接続
バックアップ機能	簡易バックアップ機能	SD メモリカードバックアップ機能 *1
	PLC バックアップツールによるバックアップ	Sysmac Studio コントローラバックアップ機能 *1

*1 NJ シリーズ CPU ユニット Ver.1.03 以降と、Sysmac Studio Ver.1.04 以降の組み合わせで使用可能です。

使用上の注意

NJ シリーズ CPU ユニットには、次の Ethernet/IP ユニットと Sysmac Studio を使用してください。

EtherNet/IP ユニット	NJ シリーズ CPU ユニット	Sysmac Studio
形 CJ1W-EIP21 Ver.2.1 以降	Ver.1.01 以降	Ver.1.02 以降
形 CJ1W-EIP21S Ver.1.0 ロット番号 241001□ 以降	Ver.1.67 以降	Ver.1.60 以降

A-7-2 ユーザプログラムからのアクセス方法に関する相違点

本ユニットを NJ シリーズで使用する場合、ユーザプログラムからは CJ ユニット用メモリを AT（割付先）指定する CJ ユニットデバイス変数を使用して EtherNet/IP ユニットの各種機能を使用します。CJ シリーズ CPU ユニットの I/O メモリ、および NJ シリーズ CPU ユニットの CJ ユニット用メモリの、CH アドレス、ビット位置と、CJ ユニットデバイス変数の対応を下記に示します。

割付リレーエリアの先頭アドレス：n ＝ 1500 ＋号機 No.×25（号機 No.：0 ～ 15）
割付 DM エリアの先頭アドレス：m ＝ D30000 ＋号機 No.×100（号機 No.：0 ～ 15）

CPU 高機能ユニット用割付リレーエリア

● nCH（ソフトスイッチ）

nCH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
nCH	0 ～ 15	*_SoftSwCmd	ソフトスイッチ（nCH のビット 0 ～ 15 の機能が右記 CJ ユニットデバイス変数に対応しています。）

nCH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
nCH	0 ～ 1	－	（未使用）
	2	*_TDLStartCmd	タグデータリンク開始スイッチ
	3	－	（未使用）
	4	*_TDLStopCmd	タグデータリンク停止スイッチ
	5	*_AdjTmCmd	時計情報自動調整実行スイッチ
	6 ～ 9	－	(未使用)
	10	*_SktForceCloseCmd	ソケット強制クローズスイッチ *1
	11 ～ 15	－	(未使用)

*1. 形 CJ1W-EIP21S だけにあるスイッチです。

● n+2 ～ n+5CH（ターゲットノード PLC 本体運転情報）

n+2 ～ n+5CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+2CH ～ n+5CH	各チャネルの 0 ～ 15	*_TargetPLCMdSta	ターゲットノード PLC 本体運転情報 (各チャネルのビット 0 ～ 15 の機能と右記 CJ ユニットデバイス変数の各ビットとの対応は以下です。 • n+2CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 0 ～ 15 • n+3CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 16 ～ 31 • n+4CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 32 ～ 47 • n+5CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 48 ～ 63

n+2 ～ n+5CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+2CH	0	*_TargetPLCMdSta[0]	ノードアドレス 0 のターゲットノード PLC ユニットの運転情報
	1	*_TargetPLCMdSta[1]	ノードアドレス 1 のターゲットノード PLC ユニットの運転情報
	2	*_TargetPLCMdSta[2]	ノードアドレス 2 のターゲットノード PLC ユニットの運転情報
	3	*_TargetPLCMdSta[3]	ノードアドレス 3 のターゲットノード PLC ユニットの運転情報
	4	*_TargetPLCMdSta[4]	ノードアドレス 4 のターゲットノード PLC ユニットの運転情報
	5	*_TargetPLCMdSta[5]	ノードアドレス 5 のターゲットノード PLC ユニットの運転情報
	6	*_TargetPLCMdSta[6]	ノードアドレス 6 のターゲットノード PLC ユニットの運転情報
	7	*_TargetPLCMdSta[7]	ノードアドレス 7 のターゲットノード PLC ユニットの運転情報
	8	*_TargetPLCMdSta[8]	ノードアドレス 8 のターゲットノード PLC ユニットの運転情報
	9	*_TargetPLCMdSta[9]	ノードアドレス 9 のターゲットノード PLC ユニットの運転情報
	10	*_TargetPLCMdSta[10]	ノードアドレス 10 のターゲットノード PLC ユニットの運転情報
n+2CH	11	*_TargetPLCMdSta[11]	ノードアドレス 11 のターゲットノード PLC ユニットの運転情報
	12	*_TargetPLCMdSta[12]	ノードアドレス 12 のターゲットノード PLC ユニットの運転情報
	13	*_TargetPLCMdSta[13]	ノードアドレス 13 のターゲットノード PLC ユニットの運転情報
	14	*_TargetPLCMdSta[14]	ノードアドレス 14 のターゲットノード PLC ユニットの運転情報
	15	*_TargetPLCMdSta[15]	ノードアドレス 15 のターゲットノード PLC ユニットの運転情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+3CH	0	*_TargetPLCMdSta[16]	ノードアドレス 16 のターゲットノード PLC ユニットの運転情報
	1	*_TargetPLCMdSta[17]	ノードアドレス 17 のターゲットノード PLC ユニットの運転情報
	2	*_TargetPLCMdSta[18]	ノードアドレス 18 のターゲットノード PLC ユニットの運転情報
	3	*_TargetPLCMdSta[19]	ノードアドレス 19 のターゲットノード PLC ユニットの運転情報
	4	*_TargetPLCMdSta[20]	ノードアドレス 20 のターゲットノード PLC ユニットの運転情報
	5	*_TargetPLCMdSta[21]	ノードアドレス 21 のターゲットノード PLC ユニットの運転情報
	6	*_TargetPLCMdSta[22]	ノードアドレス 22 のターゲットノード PLC ユニットの運転情報
	7	*_TargetPLCMdSta[23]	ノードアドレス 23 のターゲットノード PLC ユニットの運転情報
	8	*_TargetPLCMdSta[24]	ノードアドレス 24 のターゲットノード PLC ユニットの運転情報
	9	*_TargetPLCMdSta[25]	ノードアドレス 25 のターゲットノード PLC ユニットの運転情報
	10	*_TargetPLCMdSta[26]	ノードアドレス 26 のターゲットノード PLC ユニットの運転情報
	11	*_TargetPLCMdSta[27]	ノードアドレス 27 のターゲットノード PLC ユニットの運転情報
	12	*_TargetPLCMdSta[28]	ノードアドレス 28 のターゲットノード PLC ユニットの運転情報
	13	*_TargetPLCMdSta[29]	ノードアドレス 29 のターゲットノード PLC ユニットの運転情報
	14	*_TargetPLCMdSta[30]	ノードアドレス 30 のターゲットノード PLC ユニットの運転情報
	15	*_TargetPLCMdSta[31]	ノードアドレス 31 のターゲットノード PLC ユニットの運転情報
n+4CH	0	*_TargetPLCMdSta[32]	ノードアドレス 32 のターゲットノード PLC ユニットの運転情報
	1	*_TargetPLCMdSta[33]	ノードアドレス 33 のターゲットノード PLC ユニットの運転情報
	2	*_TargetPLCMdSta[34]	ノードアドレス 34 のターゲットノード PLC ユニットの運転情報
	3	*_TargetPLCMdSta[35]	ノードアドレス 35 のターゲットノード PLC ユニットの運転情報
	4	*_TargetPLCMdSta[36]	ノードアドレス 36 のターゲットノード PLC ユニットの運転情報
	5	*_TargetPLCMdSta[37]	ノードアドレス 37 のターゲットノード PLC ユニットの運転情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+4CH	6	*_TargetPLCMdSta[38]	ノードアドレス 38 のターゲットノード PLC ユニットの運転情報
	7	*_TargetPLCMdSta[39]	ノードアドレス 39 のターゲットノード PLC ユニットの運転情報
	8	*_TargetPLCMdSta[40]	ノードアドレス 40 のターゲットノード PLC ユニットの運転情報
	9	*_TargetPLCMdSta[41]	ノードアドレス 41 のターゲットノード PLC ユニットの運転情報
	10	*_TargetPLCMdSta[42]	ノードアドレス 42 のターゲットノード PLC ユニットの運転情報
	11	*_TargetPLCMdSta[43]	ノードアドレス 43 のターゲットノード PLC ユニットの運転情報
	12	*_TargetPLCMdSta[44]	ノードアドレス 44 のターゲットノード PLC ユニットの運転情報
	13	*_TargetPLCMdSta[45]	ノードアドレス 45 のターゲットノード PLC ユニットの運転情報
	14	*_TargetPLCMdSta[46]	ノードアドレス 46 のターゲットノード PLC ユニットの運転情報
	15	*_TargetPLCMdSta[47]	ノードアドレス 47 のターゲットノード PLC ユニットの運転情報
n+5CH	0	*_TargetPLCMdSta[48]	ノードアドレス 48 のターゲットノード PLC ユニットの運転情報
	1	*_TargetPLCMdSta[49]	ノードアドレス 49 のターゲットノード PLC ユニットの運転情報
	2	*_TargetPLCMdSta[50]	ノードアドレス 50 のターゲットノード PLC ユニットの運転情報
	3	*_TargetPLCMdSta[51]	ノードアドレス 51 のターゲットノード PLC ユニットの運転情報
	4	*_TargetPLCMdSta[52]	ノードアドレス 52 のターゲットノード PLC ユニットの運転情報
	5	*_TargetPLCMdSta[53]	ノードアドレス 53 のターゲットノード PLC ユニットの運転情報
	6	*_TargetPLCMdSta[54]	ノードアドレス 54 のターゲットノード PLC ユニットの運転情報
	7	*_TargetPLCMdSta[55]	ノードアドレス 55 のターゲットノード PLC ユニットの運転情報
	8	*_TargetPLCMdSta[56]	ノードアドレス 56 のターゲットノード PLC ユニットの運転情報
	9	*_TargetPLCMdSta[57]	ノードアドレス 57 のターゲットノード PLC ユニットの運転情報
	10	*_TargetPLCMdSta[58]	ノードアドレス 58 のターゲットノード PLC ユニットの運転情報
	11	*_TargetPLCMdSta[59]	ノードアドレス 59 のターゲットノード PLC ユニットの運転情報
	12	*_TargetPLCMdSta[60]	ノードアドレス 60 のターゲットノード PLC ユニットの運転情報
	13	*_TargetPLCMdSta[61]	ノードアドレス 61 のターゲットノード PLC ユニットの運転情報
	14	*_TargetPLCMdSta[62]	ノードアドレス 62 のターゲットノード PLC ユニットの運転情報
	15	*_TargetPLCMdSta[63]	ノードアドレス 63 のターゲットノード PLC ユニットの運転情報

● n+6 ～ n+9CH（ターゲットノード PLC ユニット異常情報テーブル）

n+6 ～ n+9CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+6CH ～ n+9CH	各チャネルの 0 ～ 15	*_TargetPLCErrSta	ターゲットノード PLC 本体異常情報 (各チャネルのビット 0 ～ 15 の機能と右記 CJ ユニットデバイス変数の各ビットとの対応は以下です。 • n+6CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 0 ～ 15 • n+7CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 16 ～ 31 • n+8CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 32 ～ 47 • n+9CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 48 ～ 63

n+6 ～ n+9CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+6CH	0	*_TargetPLCErrSta[0]	ノードアドレス 0 のターゲットノード PLC 本体異常情報
	1	*_TargetPLCErrSta[1]	ノードアドレス 1 のターゲットノード PLC 本体異常情報
	2	*_TargetPLCErrSta[2]	ノードアドレス 2 のターゲットノード PLC 本体異常情報
	3	*_TargetPLCErrSta[3]	ノードアドレス 3 のターゲットノード PLC 本体異常情報
	4	*_TargetPLCErrSta[4]	ノードアドレス 4 のターゲットノード PLC 本体異常情報
	5	*_TargetPLCErrSta[5]	ノードアドレス 5 のターゲットノード PLC 本体異常情報
	6	*_TargetPLCErrSta[6]	ノードアドレス 6 のターゲットノード PLC 本体異常情報
	7	*_TargetPLCErrSta[7]	ノードアドレス 7 のターゲットノード PLC 本体異常情報
	8	*_TargetPLCErrSta[8]	ノードアドレス 8 のターゲットノード PLC 本体異常情報
	9	*_TargetPLCErrSta[9]	ノードアドレス 9 のターゲットノード PLC 本体異常情報
	10	*_TargetPLCErrSta[10]	ノードアドレス 10 のターゲットノード PLC 本体異常情報
	11	*_TargetPLCErrSta[11]	ノードアドレス 11 のターゲットノード PLC 本体異常情報
	12	*_TargetPLCErrSta[12]	ノードアドレス 12 のターゲットノード PLC 本体異常情報
	13	*_TargetPLCErrSta[13]	ノードアドレス 13 のターゲットノード PLC 本体異常情報
	14	*_TargetPLCErrSta[14]	ノードアドレス 14 のターゲットノード PLC 本体異常情報
	15	*_TargetPLCErrSta[15]	ノードアドレス 15 のターゲットノード PLC 本体異常情報
n+7CH	0	*_TargetPLCErrSta[16]	ノードアドレス 16 のターゲットノード PLC 本体異常情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+7CH	1	*_TargetPLCErrSta[17]	ノードアドレス 17 のターゲットノード PLC 本体異常情報
	2	*_TargetPLCErrSta[18]	ノードアドレス 18 のターゲットノード PLC 本体異常情報
	3	*_TargetPLCErrSta[19]	ノードアドレス 19 のターゲットノード PLC 本体異常情報
	4	*_TargetPLCErrSta[20]	ノードアドレス 20 のターゲットノード PLC 本体異常情報
	5	*_TargetPLCErrSta[21]	ノードアドレス 21 のターゲットノード PLC 本体異常情報
	6	*_TargetPLCErrSta[22]	ノードアドレス 22 のターゲットノード PLC 本体異常情報
	7	*_TargetPLCErrSta[23]	ノードアドレス 23 のターゲットノード PLC 本体異常情報
	8	*_TargetPLCErrSta[24]	ノードアドレス 24 のターゲットノード PLC 本体異常情報
	9	*_TargetPLCErrSta[25]	ノードアドレス 25 のターゲットノード PLC 本体異常情報
	10	*_TargetPLCErrSta[26]	ノードアドレス 26 のターゲットノード PLC 本体異常情報
	11	*_TargetPLCErrSta[27]	ノードアドレス 27 のターゲットノード PLC 本体異常情報
	12	*_TargetPLCErrSta[28]	ノードアドレス 28 のターゲットノード PLC 本体異常情報
	13	*_TargetPLCErrSta[29]	ノードアドレス 29 のターゲットノード PLC 本体異常情報
	14	*_TargetPLCErrSta[30]	ノードアドレス 30 のターゲットノード PLC 本体異常情報
	15	*_TargetPLCErrSta[31]	ノードアドレス 31 のターゲットノード PLC 本体異常情報
n+8CH	0	*_TargetPLCErrSta[32]	ノードアドレス 32 のターゲットノード PLC 本体異常情報
	1	*_TargetPLCErrSta[33]	ノードアドレス 33 のターゲットノード PLC 本体異常情報
	2	*_TargetPLCErrSta[34]	ノードアドレス 34 のターゲットノード PLC 本体異常情報
	3	*_TargetPLCErrSta[35]	ノードアドレス 35 のターゲットノード PLC 本体異常情報
	4	*_TargetPLCErrSta[36]	ノードアドレス 36 のターゲットノード PLC 本体異常情報
	5	*_TargetPLCErrSta[37]	ノードアドレス 37 のターゲットノード PLC 本体異常情報
	6	*_TargetPLCErrSta[38]	ノードアドレス 38 のターゲットノード PLC 本体異常情報
	7	*_TargetPLCErrSta[39]	ノードアドレス 39 のターゲットノード PLC 本体異常情報
	8	*_TargetPLCErrSta[40]	ノードアドレス 40 のターゲットノード PLC 本体異常情報
	9	*_TargetPLCErrSta[41]	ノードアドレス 41 のターゲットノード PLC 本体異常情報
	10	*_TargetPLCErrSta[42]	ノードアドレス 42 のターゲットノード PLC 本体異常情報
	11	*_TargetPLCErrSta[43]	ノードアドレス 43 のターゲットノード PLC 本体異常情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+8CH	12	*_TargetPLCErrSta[44]	ノードアドレス 44 のターゲットノード PLC 本体異常情報
	13	*_TargetPLCErrSta[45]	ノードアドレス 45 のターゲットノード PLC 本体異常情報
	14	*_TargetPLCErrSta[46]	ノードアドレス 46 のターゲットノード PLC 本体異常情報
	15	*_TargetPLCErrSta[47]	ノードアドレス 47 のターゲットノード PLC 本体異常情報
n+9CH	0	*_TargetPLCErrSta[48]	ノードアドレス 48 のターゲットノード PLC 本体異常情報
	1	*_TargetPLCErrSta[49]	ノードアドレス 49 のターゲットノード PLC 本体異常情報
	2	*_TargetPLCErrSta[50]	ノードアドレス 50 のターゲットノード PLC 本体異常情報
	3	*_TargetPLCErrSta[51]	ノードアドレス 51 のターゲットノード PLC 本体異常情報
	4	*_TargetPLCErrSta[52]	ノードアドレス 52 のターゲットノード PLC 本体異常情報
	5	*_TargetPLCErrSta[53]	ノードアドレス 53 のターゲットノード PLC 本体異常情報
	6	*_TargetPLCErrSta[54]	ノードアドレス 54 のターゲットノード PLC 本体異常情報
	7	*_TargetPLCErrSta[55]	ノードアドレス 55 のターゲットノード PLC 本体異常情報
	8	*_TargetPLCErrSta[56]	ノードアドレス 56 のターゲットノード PLC 本体異常情報
	9	*_TargetPLCErrSta[57]	ノードアドレス 57 のターゲットノード PLC 本体異常情報
	10	*_TargetPLCErrSta[58]	ノードアドレス 58 のターゲットノード PLC 本体異常情報
	11	*_TargetPLCErrSta[59]	ノードアドレス 59 のターゲットノード PLC 本体異常情報
	12	*_TargetPLCErrSta[60]	ノードアドレス 60 のターゲットノード PLC 本体異常情報
	13	*_TargetPLCErrSta[61]	ノードアドレス 61 のターゲットノード PLC 本体異常情報
	14	*_TargetPLCErrSta[62]	ノードアドレス 62 のターゲットノード PLC 本体異常情報
	15	*_TargetPLCErrSta[63]	ノードアドレス 63 のターゲットノード PLC 本体異常情報

● n+10CH（ユニットステータス 1）

n+10CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+10CH	0 ～ 15	*_UnitSta1	ユニットステータス 1 (n+10CH のビット 0 ～ 15 の機能が右記 CJ ユニットデバイス変数のビット 0 ～ 15 の機能に対応しています。)

n+10CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+10CH	0	*_UnitErr	ユニット異常発生中
	1	*_NetErr	ネットワーク異常発生中
	2 ～ 3	－	（未使用）
	4	*_UnitMemErr	ユニットメモリ異常
	5	*_LANHwErr	通信コントローラ異常
	6	*_IPAdrDupErr	IP アドレス重複異常
	7 ～ 8	－	（未使用）
	9	*_LkOffErr	リンクオフ異常
	10 ～ 13	－	（未使用）
	14	*_UserStaAreaCfgErr	ステータスエリアレイアウト設定異常
	15	－	（未使用）

● n+11CH（ユニットステータス 2）

n+11CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+11CH	0 ～ 15	*_UnitSta2	ユニットステータス 2 （n+11CH のビット 0 ～ 15 の機能が右記 CJ ユニットデバイス変数のビット 0 ～ 15 の機能に対応しています。）

n+11CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+11CH	0	*_ETNOnlineSta	オンライン
	1	*_TDLOprSta	タグデータリンク動作中
	2	*_IPAdrChgErr	運転中の IP アドレス変更
	3 ～ 10	－	（未使用）
	11	*_UserStaAreaEnblSta	ユーザ設定エリア有効
	12 ～ 13	－	（未使用）
	14	*_MultiSwOnErr	複数スイッチ同時 ON 異常
	15	*_ErrLogStoreSta	異常履歴あり

● n+12CH（通信ステータス 1）

n+12CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+12CH	0 ～ 15	*_CommSta1	通信ステータス 1 （n+12CH のビット 0 ～ 15 の機能が右記 CJ ユニットデバイス変数のビット 0 ～ 15 の機能に対応しています。）

n+12CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+12CH	0	*_TDLOpnErr	照合異常
	1	－	（未使用）
	2	*_TDLErr	タグデータリンク異常
	3	－	（未使用）
	4	*_CommParamErr	通信パラメータ設定不正
	5	*_TagRefreshErr	タグリフレッシュ異常
	6	*_TagDbErr	タグデータベース異常
	7 ～ 13	－	（未使用）
	14	*_TDLAllRunSta	全タグデータリンク通信中
	15	*_TDLRunSta	タグデータリンク通信中

● n+13CH（通信ステータス 2）

n+13CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+13CH	0 ～ 15	*_CommSta2	通信ステータス 2 (n+13CH のビット 0 ～ 15 の機能が右記 CJ ユニットデバイス変数のビット 0 ～ 15 の機能に対応しています。)

n+13CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+13CH	0	*_FTPSta	FTP ステータス
	1 ～ 13	－	（未使用）
	14	*_LkSta	リンクステータス
	15	－	（未使用）

● n+14CH（通信ステータス 3）

n+14CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+14CH	0 ～ 15	*_CommSta3	通信ステータス 3 (n+14CH のビット 0 ～ 15 の機能が右記 CJ ユニットデバイス変数のビット 0 ～ 15 の機能に対応しています。)

n+14CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+14CH	0 ～ 1	－	（未使用）
	2	*_ETNBaseSetErr	Ethernet 基本設定異常
	3	*_IPAdrTblErr	IP アドレステーブル異常
	4	*_IPRouterTblErr	IP ルータテーブル異常
	5	*_DNSSvrErr	DNS サーバ異常
	6	*_RTblErr	ルーチングテーブル異常
	7 ～ 8	－	（未使用）
	9	*_ETNAdvSetErr	Ethernet 拡張設定異常
	10	*_BootpSvrErr	BOOTP サーバ異常
	11	*_SNTPSvrErr	SNTP サーバ異常
	12 ～ 13	－	（未使用）
	14	*_AdrMismatchErr	アドレス不一致
	15	*_MemErr	不揮発性メモリ異常

● n+16 ～ n+19CH（登録ターゲットノードテーブル）

n + 16 ～ n + 19CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+16CH ～ n+19CH	各チャネルの 0 ～ 15	*_RegTargetSta	ターゲットノードの登録情報 （各チャネルのビット 0 ～ 15 の機能と右記 CJ ユニットデバイス変数の各ビットとの対応は以下です。 • n+16CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 0 ～ 15 • n+17CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 16 ～ 31 • n+18CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 32 ～ 47 • n+19CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 48 ～ 63

n+16 ～ n+19CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+16CH	0	*_RegTargetSta[0]	ノードアドレス 0 のターゲットノードの登録情報
	1	*_RegTargetSta[1]	ノードアドレス 1 のターゲットノードの登録情報
	2	*_RegTargetSta[2]	ノードアドレス 2 のターゲットノードの登録情報
	3	*_RegTargetSta[3]	ノードアドレス 3 のターゲットノードの登録情報
	4	*_RegTargetSta[4]	ノードアドレス 4 のターゲットノードの登録情報
	5	*_RegTargetSta[5]	ノードアドレス 5 のターゲットノードの登録情報
	6	*_RegTargetSta[6]	ノードアドレス 6 のターゲットノードの登録情報
	7	*_RegTargetSta[7]	ノードアドレス 7 のターゲットノードの登録情報
	8	*_RegTargetSta[8]	ノードアドレス 8 のターゲットノードの登録情報
	9	*_RegTargetSta[9]	ノードアドレス 9 のターゲットノードの登録情報
	10	*_RegTargetSta[10]	ノードアドレス 10 のターゲットノードの登録情報
	11	*_RegTargetSta[11]	ノードアドレス 11 のターゲットノードの登録情報
	12	*_RegTargetSta[12]	ノードアドレス 12 のターゲットノードの登録情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+16CH	13	*_RegTargetSta[13]	ノードアドレス 13 のターゲットノードの登録情報
	14	*_RegTargetSta[14]	ノードアドレス 14 のターゲットノードの登録情報
	15	*_RegTargetSta[15]	ノードアドレス 15 のターゲットノードの登録情報
n+17CH	0	*_RegTargetSta[16]	ノードアドレス 16 のターゲットノードの登録情報
	1	*_RegTargetSta[17]	ノードアドレス 17 のターゲットノードの登録情報
	2	*_RegTargetSta[18]	ノードアドレス 18 のターゲットノードの登録情報
	3	*_RegTargetSta[19]	ノードアドレス 19 のターゲットノードの登録情報
	4	*_RegTargetSta[20]	ノードアドレス 20 のターゲットノードの登録情報
	5	*_RegTargetSta[21]	ノードアドレス 21 のターゲットノードの登録情報
	6	*_RegTargetSta[22]	ノードアドレス 22 のターゲットノードの登録情報
	7	*_RegTargetSta[23]	ノードアドレス 23 のターゲットノードの登録情報
	8	*_RegTargetSta[24]	ノードアドレス 24 のターゲットノードの登録情報
	9	*_RegTargetSta[25]	ノードアドレス 25 のターゲットノードの登録情報
	10	*_RegTargetSta[26]	ノードアドレス 26 のターゲットノードの登録情報
	11	*_RegTargetSta[27]	ノードアドレス 27 のターゲットノードの登録情報
	12	*_RegTargetSta[28]	ノードアドレス 28 のターゲットノードの登録情報
	13	*_RegTargetSta[29]	ノードアドレス 29 のターゲットノードの登録情報
	14	*_RegTargetSta[30]	ノードアドレス 30 のターゲットノードの登録情報
	15	*_RegTargetSta[31]	ノードアドレス 31 のターゲットノードの登録情報
n+18CH	0	*_RegTargetSta[32]	ノードアドレス 32 のターゲットノードの登録情報
	1	*_RegTargetSta[33]	ノードアドレス 33 のターゲットノードの登録情報
	2	*_RegTargetSta[34]	ノードアドレス 34 のターゲットノードの登録情報
	3	*_RegTargetSta[35]	ノードアドレス 35 のターゲットノードの登録情報
	4	*_RegTargetSta[36]	ノードアドレス 36 のターゲットノードの登録情報
	5	*_RegTargetSta[37]	ノードアドレス 37 のターゲットノードの登録情報
	6	*_RegTargetSta[38]	ノードアドレス 38 のターゲットノードの登録情報
	7	*_RegTargetSta[39]	ノードアドレス 39 のターゲットノードの登録情報
	8	*_RegTargetSta[40]	ノードアドレス 40 のターゲットノードの登録情報
	9	*_RegTargetSta[41]	ノードアドレス 41 のターゲットノードの登録情報
	10	*_RegTargetSta[42]	ノードアドレス 42 のターゲットノードの登録情報
	11	*_RegTargetSta[43]	ノードアドレス 43 のターゲットノードの登録情報
	12	*_RegTargetSta[44]	ノードアドレス 44 のターゲットノードの登録情報
	13	*_RegTargetSta[45]	ノードアドレス 45 のターゲットノードの登録情報
	14	*_RegTargetSta[46]	ノードアドレス 46 のターゲットノードの登録情報
	15	*_RegTargetSta[47]	ノードアドレス 47 のターゲットノードの登録情報
n+19CH	0	*_RegTargetSta[48]	ノードアドレス 48 のターゲットノードの登録情報
	1	*_RegTargetSta[49]	ノードアドレス 49 のターゲットノードの登録情報
	2	*_RegTargetSta[50]	ノードアドレス 50 のターゲットノードの登録情報
	3	*_RegTargetSta[51]	ノードアドレス 51 のターゲットノードの登録情報
	4	*_RegTargetSta[52]	ノードアドレス 52 のターゲットノードの登録情報
	5	*_RegTargetSta[53]	ノードアドレス 53 のターゲットノードの登録情報
	6	*_RegTargetSta[54]	ノードアドレス 54 のターゲットノードの登録情報
	7	*_RegTargetSta[55]	ノードアドレス 55 のターゲットノードの登録情報
	8	*_RegTargetSta[56]	ノードアドレス 56 のターゲットノードの登録情報
	9	*_RegTargetSta[57]	ノードアドレス 57 のターゲットノードの登録情報
	10	*_RegTargetSta[58]	ノードアドレス 58 のターゲットノードの登録情報
	11	*_RegTargetSta[59]	ノードアドレス 59 のターゲットノードの登録情報
	12	*_RegTargetSta[60]	ノードアドレス 60 のターゲットノードの登録情報
	13	*_RegTargetSta[61]	ノードアドレス 61 のターゲットノードの登録情報
	14	*_RegTargetSta[62]	ノードアドレス 62 のターゲットノードの登録情報
	15	*_RegTargetSta[63]	ノードアドレス 63 のターゲットノードの登録情報

● n+20 ～ n+23CH（正常ターゲットノードテーブル）

n+20 ～ n+23CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+20CH ～ n+23CH	各チャネルの 0 ～ 15	*_EstbRegTargetSta	ターゲットノードとのコネクション確立情報 （各チャネルのビット 0 ～ 15 の機能と右記 CJ ユニットデバイス変数の各ビットとの対応は以下です。 • n+20CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 0 ～ 15 • n+21CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 16 ～ 31 • n+22CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 32 ～ 47 • n+23CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 48 ～ 63

n+20 ～ n+23CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+20CH	0	*_EstbRegTargetSta[0]	ノードアドレス 0 のターゲットノードとの確立情報
	1	*_EstbRegTargetSta[1]	ノードアドレス 1 のターゲットノードとの確立情報
	2	*_EstbRegTargetSta[2]	ノードアドレス 2 のターゲットノードとの確立情報
	3	*_EstbRegTargetSta[3]	ノードアドレス 3 のターゲットノードとの確立情報
	4	*_EstbRegTargetSta[4]	ノードアドレス 4 のターゲットノードとの確立情報
	5	*_EstbRegTargetSta[5]	ノードアドレス 5 のターゲットノードとの確立情報
	6	*_EstbRegTargetSta[6]	ノードアドレス 6 のターゲットノードとの確立情報
	7	*_EstbRegTargetSta[7]	ノードアドレス 7 のターゲットノードとの確立情報
	8	*_EstbRegTargetSta[8]	ノードアドレス 8 のターゲットノードとの確立情報
	9	*_EstbRegTargetSta[9]	ノードアドレス 9 のターゲットノードとの確立情報
	10	*_EstbRegTargetSta[10]	ノードアドレス 10 のターゲットノードとの確立情報
	11	*_EstbRegTargetSta[11]	ノードアドレス 11 のターゲットノードとの確立情報
	12	*_EstbRegTargetSta[12]	ノードアドレス 12 のターゲットノードとの確立情報
	13	*_EstbRegTargetSta[13]	ノードアドレス 13 のターゲットノードとの確立情報
	14	*_EstbRegTargetSta[14]	ノードアドレス 14 のターゲットノードとの確立情報
	15	*_EstbRegTargetSta[15]	ノードアドレス 15 のターゲットノードとの確立情報
n+21CH	0	*_EstbRegTargetSta[16]	ノードアドレス 16 のターゲットノードとの確立情報
	1	*_EstbRegTargetSta[17]	ノードアドレス 17 のターゲットノードとの確立情報
	2	*_EstbRegTargetSta[18]	ノードアドレス 18 のターゲットノードとの確立情報
	3	*_EstbRegTargetSta[19]	ノードアドレス 19 のターゲットノードとの確立情報
	4	*_EstbRegTargetSta[20]	ノードアドレス 20 のターゲットノードとの確立情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+21CH	5	*_EstbRegTargetSta[21]	ノードアドレス 21 のターゲットノードとの確立情報
	6	*_EstbRegTargetSta[22]	ノードアドレス 22 のターゲットノードとの確立情報
	7	*_EstbRegTargetSta[23]	ノードアドレス 23 のターゲットノードとの確立情報
	8	*_EstbRegTargetSta[24]	ノードアドレス 24 のターゲットノードとの確立情報
	9	*_EstbRegTargetSta[25]	ノードアドレス 25 のターゲットノードとの確立情報
	10	*_EstbRegTargetSta[26]	ノードアドレス 26 のターゲットノードとの確立情報
	11	*_EstbRegTargetSta[27]	ノードアドレス 27 のターゲットノードとの確立情報
	12	*_EstbRegTargetSta[28]	ノードアドレス 28 のターゲットノードとの確立情報
	13	*_EstbRegTargetSta[29]	ノードアドレス 29 のターゲットノードとの確立情報
	14	*_EstbRegTargetSta[30]	ノードアドレス 30 のターゲットノードとの確立情報
	15	*_EstbRegTargetSta[31]	ノードアドレス 31 のターゲットノードとの確立情報
n+22CH	0	*_EstbRegTargetSta[32]	ノードアドレス 32 のターゲットノードとの確立情報
	1	*_EstbRegTargetSta[33]	ノードアドレス 33 のターゲットノードとの確立情報
	2	*_EstbRegTargetSta[34]	ノードアドレス 34 のターゲットノードとの確立情報
	3	*_EstbRegTargetSta[35]	ノードアドレス 35 のターゲットノードとの確立情報
	4	*_EstbRegTargetSta[36]	ノードアドレス 36 のターゲットノードとの確立情報
	5	*_EstbRegTargetSta[37]	ノードアドレス 37 のターゲットノードとの確立情報
	6	*_EstbRegTargetSta[38]	ノードアドレス 38 のターゲットノードとの確立情報
	7	*_EstbRegTargetSta[39]	ノードアドレス 39 のターゲットノードとの確立情報
	8	*_EstbRegTargetSta[40]	ノードアドレス 40 のターゲットノードとの確立情報
	9	*_EstbRegTargetSta[41]	ノードアドレス 41 のターゲットノードとの確立情報
	10	*_EstbRegTargetSta[42]	ノードアドレス 42 のターゲットノードとの確立情報
	11	*_EstbRegTargetSta[43]	ノードアドレス 43 のターゲットノードとの確立情報
	12	*_EstbRegTargetSta[44]	ノードアドレス 44 のターゲットノードとの確立情報
	13	*_EstbRegTargetSta[45]	ノードアドレス 45 のターゲットノードとの確立情報
	14	*_EstbRegTargetSta[46]	ノードアドレス 46 のターゲットノードとの確立情報
	15	*_EstbRegTargetSta[47]	ノードアドレス 47 のターゲットノードとの確立情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+23CH	0	*_EstbRegTargetSta[48]	ノードアドレス 48 のターゲットノードとの確立情報
	1	*_EstbRegTargetSta[49]	ノードアドレス 49 のターゲットノードとの確立情報
	2	*_EstbRegTargetSta[50]	ノードアドレス 50 のターゲットノードとの確立情報
	3	*_EstbRegTargetSta[51]	ノードアドレス 51 のターゲットノードとの確立情報
	4	*_EstbRegTargetSta[52]	ノードアドレス 52 のターゲットノードとの確立情報
	5	*_EstbRegTargetSta[53]	ノードアドレス 53 のターゲットノードとの確立情報
	6	*_EstbRegTargetSta[54]	ノードアドレス 54 のターゲットノードとの確立情報
	7	*_EstbRegTargetSta[55]	ノードアドレス 55 のターゲットノードとの確立情報
	8	*_EstbRegTargetSta[56]	ノードアドレス 56 のターゲットノードとの確立情報
	9	*_EstbRegTargetSta[57]	ノードアドレス 57 のターゲットノードとの確立情報
	10	*_EstbRegTargetSta[58]	ノードアドレス 58 のターゲットノードとの確立情報
	11	*_EstbRegTargetSta[59]	ノードアドレス 59 のターゲットノードとの確立情報
	12	*_EstbRegTargetSta[60]	ノードアドレス 60 のターゲットノードとの確立情報
	13	*_EstbRegTargetSta[61]	ノードアドレス 61 のターゲットノードとの確立情報
	14	*_EstbRegTargetSta[62]	ノードアドレス 62 のターゲットノードとの確立情報
	15	*_EstbRegTargetSta[63]	ノードアドレス 63 のターゲットノードとの確立情報

● n+24CH（FINS/TCP コネクション接続ステータス）

n+24CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+24CH	0 ～ 15	*_FINSTCPConnSta	FINS/TCP コネクション接続ステータス (n+24CH のビット 0 ～ 15 の機能が右記 CJ ユニットデバイス変数のビット 0 ～ 15 の機能に対応しています。)

n+24CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+24CH	0	*_FINSTCPConnSta1	FINS/TCP コネクション No.1 とのコネクション確立情報
	1	*_FINSTCPConnSta2	FINS/TCP コネクション No.2 とのコネクション確立情報
	2	*_FINSTCPConnSta3	FINS/TCP コネクション No.3 とのコネクション確立情報
	3	*_FINSTCPConnSta4	FINS/TCP コネクション No.4 とのコネクション確立情報
	4	*_FINSTCPConnSta5	FINS/TCP コネクション No.5 とのコネクション確立情報
	5	*_FINSTCPConnSta6	FINS/TCP コネクション No.6 とのコネクション確立情報
	6	*_FINSTCPConnSta7	FINS/TCP コネクション No.7 とのコネクション確立情報
	7	*_FINSTCPConnSta8	FINS/TCP コネクション No.8 とのコネクション確立情報
	8	*_FINSTCPConnSta9	FINS/TCP コネクション No.9 とのコネクション確立情報
	9	*_FINSTCPConnSta10	FINS/TCP コネクション No.10 とのコネクション確立情報
	10	*_FINSTCPConnSta11	FINS/TCP コネクション No.11 とのコネクション確立情報
	11	*_FINSTCPConnSta12	FINS/TCP コネクション No.12 とのコネクション確立情報
	12	*_FINSTCPConnSta13	FINS/TCP コネクション No.13 とのコネクション確立情報
	13	*_FINSTCPConnSta14	FINS/TCP コネクション No.14 とのコネクション確立情報
	14	*_FINSTCPConnSta15	FINS/TCP コネクション No.15 とのコネクション確立情報
	15	*_FINSTCPConnSta16	FINS/TCP コネクション No.16 とのコネクション確立情報

● n+1CH ～ n+4CH（ソケットサービス要求スイッチ）（形 CJ1W-EIP21S のみ）

割付リレーエリアのステータスレイアウトが［ユーザ設定］時

n+1CH ～ n+4CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+1CH	0/8	*_Skt1UdpOpenCmd/ *_Skt2UdpOpenCmd	UDP オープン要求スイッチ（ソケット No.1）/ UDP オープン要求スイッチ（ソケット No.2）
	1/9	*_Skt1TcpPassiveCmd/ *_Skt2TcpPassiveCmd	TCP オープン (Passive) 要求スイッチ（ソケット No.1）/ TCP オープン (Passive) 要求スイッチ（ソケット No.2）
	2/10	*_Skt1TcpActiveCmd/ *_Skt2TcpActiveCmd	TCP オープン (Active) 要求スイッチ（ソケット No.1）/ TCP オープン (Active) 要求スイッチ（ソケット No.2）
	3/11	*_Skt1SendCmd/ *_Skt2SendCmd	送信要求スイッチ（ソケット No.1）/ 送信要求スイッチ（ソケット No.2）
	4/12	*_Skt1RcvCmd/ *_Skt2RcvCmd	受信要求スイッチ（ソケット No.1）/ 受信要求スイッチ（ソケット No.2）
	5/13	*_Skt1CloseCmd/ *_Skt2CloseCmd	クローズ要求スイッチ（ソケット No.1）/ クローズ要求スイッチ（ソケット No.2）
	6/14	—	—
	7/15	—	—
n+2CH	0/8	*_Skt3UdpOpenCmd/ *_Skt4UdpOpenCmd	UDP オープン要求スイッチ（ソケット No.3）/ UDP オープン要求スイッチ（ソケット No.4）
	1/9	*_Skt3TcpPassiveCmd/ *_Skt4TcpPassiveCmd	TCP オープン (Passive) 要求スイッチ（ソケット No.3）/ TCP オープン (Passive) 要求スイッチ（ソケット No.4）
	2/10	*_Skt3TcpActiveCmd/ *_Skt4TcpActiveCmd	TCP オープン (Active) 要求スイッチ（ソケット No.3）/ TCP オープン (Active) 要求スイッチ（ソケット No.4）
	3/11	*_Skt3SendCmd/ *_Skt4SendCmd	送信要求スイッチ（ソケット No.3）/ 送信要求スイッチ（ソケット No.4）
	4/12	*_Skt3RcvCmd/ *_Skt4RcvCmd	受信要求スイッチ（ソケット No.3）/ 受信要求スイッチ（ソケット No.4）
	5/13	*_Skt3CloseCmd/ *_Skt4CloseCmd	クローズ要求スイッチ（ソケット No.3）/ クローズ要求スイッチ（ソケット No.4）
	6/14	—	—
	7/15	—	—

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+3CH	0/8	*_Skt5UdpOpenCmd/ *_Skt6UdpOpenCmd	UDP オープン要求スイッチ（ソケット No.5）/ UDP オープン要求スイッチ（ソケット No.6）
	1/9	*_Skt5TcpPassiveCmd/ *_Skt6TcpPassiveCmd	TCP オープン (Passive) 要求スイッチ（ソケット No.5）/ TCP オープン (Passive) 要求スイッチ（ソケット No.6）
	2/10	*_Skt5TcpActiveCmd/ *_Skt6TcpActiveCmd	TCP オープン (Active) 要求スイッチ（ソケット No.5）/ TCP オープン (Active) 要求スイッチ（ソケット No.6）
	3/11	*_Skt5SendCmd/ *_Skt6SendCmd	送信要求スイッチ（ソケット No.5）/ 送信要求スイッチ（ソケット No.6）
	4/12	*_Skt5RcvCmd/ *_Skt6RcvCmd	受信要求スイッチ（ソケット No.5）/ 受信要求スイッチ（ソケット No.6）
	5/13	*_Skt5CloseCmd/ *_Skt6CloseCmd	クローズ要求スイッチ（ソケット No.5）/ クローズ要求スイッチ（ソケット No.6）
	6/14	—	—
	7/15	—	—
n+4CH	0/8	*_Skt7UdpOpenCmd/ *_Skt8UdpOpenCmd	UDP オープン要求スイッチ（ソケット No.7）/ UDP オープン要求スイッチ（ソケット No.8）
	1/9	*_Skt7TcpPassiveCmd/ *_Skt8TcpPassiveCmd	TCP オープン (Passive) 要求スイッチ（ソケット No.7）/ TCP オープン (Passive) 要求スイッチ（ソケット No.8）
	2/10	*_Skt7TcpActiveCmd/ *_Skt8TcpActiveCmd	TCP オープン (Active) 要求スイッチ（ソケット No.7）/ TCP オープン (Active) 要求スイッチ（ソケット No.8）
	3/11	*_Skt7SendCmd/ *_Skt8SendCmd	送信要求スイッチ（ソケット No.7）/ 送信要求スイッチ（ソケット No.8）
	4/12	*_Skt7RcvCmd/ *_Skt8RcvCmd	受信要求スイッチ（ソケット No.7）/ 受信要求スイッチ（ソケット No.8）
	5/13	*_Skt7CloseCmd/ *_Skt8CloseCmd	クローズ要求スイッチ（ソケット No.7）/ クローズ要求スイッチ（ソケット No.8）
	6/14	—	—
	7/15	—	—

● n+16CH ～ n+23CH（UDP/TCP ソケット No.1 ～ No.8 ステータス）（形 CJ1W-EIP21S のみ）

割付リレーエリアのステータスレイアウトが［ユーザ設定］時

n+16CH ～ n+23CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+16CH	0/8	*_Skt1UdpOpening/ *_Skt2UdpOpening	オープン処理中フラグ（ソケット No.1）/ オープン処理中フラグ（ソケット No.2）
	1/9	*_Skt1UdpRcving/ *_Skt2UdpRcving	受信処理中フラグ（ソケット No.1）/ 受信処理中フラグ（ソケット No.2）
	2/10	*_Skt1UdpSending/ *_Skt2UdpSending	送信処理中フラグ（ソケット No.1）/ 送信処理中フラグ（ソケット No.2）
	3/11	*_Skt1UdpClosing/ *_Skt2UdpClosing	クローズ処理中フラグ（ソケット No.1）/ クローズ処理中フラグ（ソケット No.2）
	4/12	—	—
	5/13	*_Skt1UdpAreaErr/ *_Skt2UdpAreaErr	結果格納異常フラグ（ソケット No.1）/ 結果格納異常フラグ（ソケット No.2）
	6/14	*_Skt1UdpOpened/ *_Skt2UdpOpened	UDP オープン済みフラグ（ソケット No.1）/ UDP オープン済みフラグ（ソケット No.2）
	7/15	—	—
n+17CH	0/8	*_Skt3UdpOpening/ *_Skt4UdpOpening	オープン処理中フラグ（ソケット No.3）/ オープン処理中フラグ（ソケット No.4）
	1/9	*_Skt3UdpRcving/ *_Skt4UdpRcving	受信処理中フラグ（ソケット No.3）/ 受信処理中フラグ（ソケット No.4）
	2/10	*_Skt3UdpSending/ *_Skt4UdpSending	送信処理中フラグ（ソケット No.3）/ 送信処理中フラグ（ソケット No.4）
	3/11	*_Skt3UdpClosing/ *_Skt4UdpClosing	クローズ処理中フラグ（ソケット No.3）/ クローズ処理中フラグ（ソケット No.4）
	4/12	—	—
	5/13	*_Skt3UdpAreaErr/ *_Skt4UdpAreaErr	結果格納異常フラグ（ソケット No.3）/ 結果格納異常フラグ（ソケット No.4）
	6/14	*_Skt3UdpOpened/ *_Skt4UdpOpened	UDP オープン済みフラグ（ソケット No.3）/ UDP オープン済みフラグ（ソケット No.4）
	7/15	—	—
n+18CH	0/8	*_Skt5UdpOpening/ *_Skt6UdpOpening	オープン処理中フラグ（ソケット No.5）/ オープン処理中フラグ（ソケット No.6）
	1/9	*_Skt5UdpRcving/ *_Skt6UdpRcving	受信処理中フラグ（ソケット No.5）/ 受信処理中フラグ（ソケット No.6）
	2/10	*_Skt5UdpSending/ *_Skt6UdpSending	送信処理中フラグ（ソケット No.5）/ 送信処理中フラグ（ソケット No.6）
	3/11	*_Skt5UdpClosing/ *_Skt6UdpClosing	クローズ処理中フラグ（ソケット No.5）/ クローズ処理中フラグ（ソケット No.6）
	4/12	—	—
	5/13	*_Skt5UdpAreaErr/ *_Skt6UdpAreaErr	結果格納異常フラグ（ソケット No.5）/ 結果格納異常フラグ（ソケット No.6）
	6/14	*_Skt5UdpOpened/ *_Skt6UdpOpened	UDP オープン済みフラグ（ソケット No.5）/ UDP オープン済みフラグ（ソケット No.6）
	7/15	—	—

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+19CH	0/8	*_Skt7UdpOpening/ *_Skt8UdpOpening	オープン処理中フラグ（ソケット No.7）/ オープン処理中フラグ（ソケット No.8）
	1/9	*_Skt7UdpRcving/ *_Skt8UdpRcving	受信処理中フラグ（ソケット No.7）/ 受信処理中フラグ（ソケット No.8）
	2/10	*_Skt7UdpSending/ *_Skt8UdpSending	送信処理中フラグ（ソケット No.7）/ 送信処理中フラグ（ソケット No.8）
	3/11	*_Skt7UdpClosing/ *_Skt8UdpClosing	クローズ処理中フラグ（ソケット No.7）/ クローズ処理中フラグ（ソケット No.8）
	4/12	—	—
	5/13	*_Skt7UdpAreaErr/ *_Skt8UdpAreaErr	結果格納異常フラグ（ソケット No.7）/ 結果格納異常フラグ（ソケット No.8）
	6/14	*_Skt7UdpOpened/ *_Skt8UdpOpened	UDP オープン済みフラグ（ソケット No.7）/ UDP オープン済みフラグ（ソケット No.8）
	7/15	—	—
n+20CH	0/8	*_Skt1TcpOpening/ *_Skt2TcpOpening	オープン処理中フラグ（ソケット No.1）/ オープン処理中フラグ（ソケット No.2）
	1/9	*_Skt1TcpRcving/ *_Skt2TcpRcving	受信処理中フラグ（ソケット No.1）/ 受信処理中フラグ（ソケット No.2）
	2/10	*_Skt1TcpSending/ *_Skt2TcpSending	送信処理中フラグ（ソケット No.1）/ 送信処理中フラグ（ソケット No.2）
	3/11	*_Skt1TcpClosing/ *_Skt2TcpClosing	クローズ処理中フラグ（ソケット No.1）/ クローズ処理中フラグ（ソケット No.2）
	4/12	*_Skt1TcpDatRcved/ *_Skt2TcpDatRcved	受信データあり／なしフラグ（ソケット No.1）/ 受信データあり／なしフラグ（ソケット No.2）
	5/13	*_Skt1TcpAreaErr/ *_Skt2TcpAreaErr	結果格納異常フラグ（ソケット No.1）/ 結果格納異常フラグ（ソケット No.2）
	6/14	*_Skt1TcpOpened/ *_Skt2TcpOpened	TCP オープン済みフラグ（ソケット No.1）/ TCP オープン済みフラグ（ソケット No.2）
	7/15	—	—
n+21CH	0/8	*_Skt3TcpOpening/ *_Skt4TcpOpening	オープン処理中フラグ（ソケット No.3）/ オープン処理中フラグ（ソケット No.4）
	1/9	*_Skt3TcpRcving/ *_Skt4TcpRcving	受信処理中フラグ（ソケット No.3）/ 受信処理中フラグ（ソケット No.4）
	2/10	*_Skt3TcpSending/ *_Skt4TcpSending	送信処理中フラグ（ソケット No.3）/ 送信処理中フラグ（ソケット No.4）
	3/11	*_Skt3TcpClosing/ *_Skt4TcpClosing	クローズ処理中フラグ（ソケット No.3）/ クローズ処理中フラグ（ソケット No.4）
	4/12	*_Skt3TcpDatRcved/ *_Skt4TcpDatRcved	受信データあり／なしフラグ（ソケット No.3）/ 受信データあり／なしフラグ（ソケット No.4）
	5/13	*_Skt3TcpAreaErr/ *_Skt4TcpAreaErr	結果格納異常フラグ（ソケット No.3）/ 結果格納異常フラグ（ソケット No.4）
	6/14	*_Skt3TcpOpened/ *_Skt4TcpOpened	TCP オープン済みフラグ（ソケット No.3）/ TCP オープン済みフラグ（ソケット No.4）
	7/15	—	—

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
n+22CH	0/8	*_Skt5TcpOpening/ *_Skt6TcpOpening	オープン処理中フラグ（ソケット No.5）/ オープン処理中フラグ（ソケット No.6）
	1/9	*_Skt5TcpRcving/ *_Skt6TcpRcving	受信処理中フラグ（ソケット No.5）/ 受信処理中フラグ（ソケット No.6）
	2/10	*_Skt5TcpSending/ *_Skt6TcpSending	送信処理中フラグ（ソケット No.5）/ 送信処理中フラグ（ソケット No.6）
	3/11	*_Skt5TcpClosing/ *_Skt6TcpClosing	クローズ処理中フラグ（ソケット No.5）/ クローズ処理中フラグ（ソケット No.6）
	4/12	*_Skt5TcpDatRcved/ *_Skt6TcpDatRcved	受信データあり／なしフラグ（ソケット No.5）/ 受信データあり／なしフラグ（ソケット No.6）
	5/13	*_Skt5TcpAreaErr/ *_Skt6TcpAreaErr	結果格納異常フラグ（ソケット No.5）/ 結果格納異常フラグ（ソケット No.6）
	6/14	*_Skt5TcpOpened/ *_Skt6TcpOpened	TCP オープン済みフラグ（ソケット No.5）/ TCP オープン済みフラグ（ソケット No.6）
	7/15	－	－
n+23CH	0/8	*_Skt7TcpOpening/ *_Skt8TcpOpening	オープン処理中フラグ（ソケット No.7）/ オープン処理中フラグ（ソケット No.8）
	1/9	*_Skt7TcpRcving/ *_Skt8TcpRcving	受信処理中フラグ（ソケット No.7）/ 受信処理中フラグ（ソケット No.8）
	2/10	*_Skt7TcpSending/ *_Skt8TcpSending	送信処理中フラグ（ソケット No.7）/ 送信処理中フラグ（ソケット No.8）
	3/11	*_Skt7TcpClosing/ *_Skt8TcpClosing	クローズ処理中フラグ（ソケット No.7）/ クローズ処理中フラグ（ソケット No.8）
	4/12	*_Skt7TcpDatRcved/ *_Skt8TcpDatRcved	受信データあり／なしフラグ（ソケット No.7）/ 受信データあり／なしフラグ（ソケット No.8）
	5/13	*_Skt7TcpAreaErr/ *_Skt8TcpAreaErr	結果格納異常フラグ（ソケット No.7）/ 結果格納異常フラグ（ソケット No.8）
	6/14	*_Skt7TcpOpened/ *_Skt8TcpOpened	TCP オープン済みフラグ（ソケット No.7）/ TCP オープン済みフラグ（ソケット No.8）
	7/15	－	－

CPU 高機能ユニット用割付 DM エリア

● m+98CH ～ m+99CH（IP アドレス表示・設定エリア）

m+98CH ～ m+99CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
m+98CH ～ m+99CH	0 ～ 15	*_IPAdrCfg	IP アドレス表示・設定エリア

m+98CH ～ m+99CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
m+98CH	0 ～ 3	*_IPAdr2Cfg	IP アドレス 2（以下②の部分）
	4 ～ 7		
	8 ～ 11	*_IPAdr1Cfg	IP アドレス 1（以下①の部分）
	12 ～ 15		

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
m+99CH	0 ～ 3	*_IPAdr4Cfg	IP アドレス 4（以下④の部分）
	4 ～ 7		
	8 ～ 11	*_IPAdr3Cfg	IP アドレス 3（以下③の部分）
	12 ～ 15		

IP アドレス：①.②.③.④（16 進数）

● m+1CH ～ m+8CH（TCP ソケット No.1 ～ 8 受信データバイト数）（形 CJ1W-EIP21S のみ）

m+1CH ～ m+8CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
m+1CH	0 ～ 15	*_Skt1TcpRcvDatByte	TCP ソケット No.1 受信データバイト数 (0000 ～ 07C0 Hex)
m+2CH	0 ～ 15	*_Skt2TcpRcvDatByte	TCP ソケット No.2 受信データバイト数 (0000 ～ 07C0 Hex)
m+3CH	0 ～ 15	*_Skt3TcpRcvDatByte	TCP ソケット No.3 受信データバイト数 (0000 ～ 07C0 Hex)
m+4CH	0 ～ 15	*_Skt4TcpRcvDatByte	TCP ソケット No.4 受信データバイト数 (0000 ～ 07C0 Hex)
m+5CH	0 ～ 15	*_Skt5TcpRcvDatByte	TCP ソケット No.5 受信データバイト数 (0000 ～ 07C0 Hex)
m+6CH	0 ～ 15	*_Skt6TcpRcvDatByte	TCP ソケット No.6 受信データバイト数 (0000 ～ 07C0 Hex)
m+7CH	0 ～ 15	*_Skt7TcpRcvDatByte	TCP ソケット No.7 受信データバイト数 (0000 ～ 07C0 Hex)
m+8CH	0 ～ 15	*_Skt8TcpRcvDatByte	TCP ソケット No.8 受信データバイト数 (0000 ～ 07C0 Hex)

● m+9CH～m+16CH（TCP ソケット No.1～8 コネクション状態）（形 CJ1W-EIP21S のみ）

m+9CH ～ m+16CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
m+9CH	0 ～ 3	*_Skt1TcpConSta	TCP ソケット No.1 コネクション状態
m+10CH	0 ～ 3	*_Skt2TcpConSta	TCP ソケット No.2 コネクション状態
m+11CH	0 ～ 3	*_Skt3TcpConSta	TCP ソケット No.3 コネクション状態
m+12CH	0 ～ 3	*_Skt4TcpConSta	TCP ソケット No.4 コネクション状態
m+13CH	0 ～ 3	*_Skt5TcpConSta	TCP ソケット No.5 コネクション状態
m+14CH	0 ～ 3	*_Skt6TcpConSta	TCP ソケット No.6 コネクション状態
m+15CH	0 ～ 3	*_Skt7TcpConSta	TCP ソケット No.7 コネクション状態
m+16CH	0 ～ 3	*_Skt8TcpConSta	TCP ソケット No.8 コネクション状態

● m+17CH ～ m+96CH（ソケットサービスパラメータエリア 1 ～ 8）（形 CJ1W-EIP21S のみ）

m+17CH ～ m+96CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

ソケット No.1

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
m+17CH	0 ～ 7	*_Skt1No	UDP/TCP ソケット No.
	8	*_Skt1KeepAlive	Keep-alive 機能使用
	9	*_Skt1Linger	Linger 機能使用
	10 ～ 15	－	－
m+18CH	0 ～ 15	*_Skt1LocalPortNo	自 UDP/TCP ポート No.(0000 ～ FFFF Hex)
m+19CH	0 ～ 7	*_Skt1RemoteIPAdr2	相手 IP アドレスの 2 バイトめ (00 ～ FF Hex)
	8 ～ 15	*_Skt1RemoteIPAdr1	相手 IP アドレスの 1 バイトめ (00 ～ FF Hex)
m+20CH	0 ～ 7	*_Skt1RemoteIPAdr4	相手 IP アドレスの 4 バイトめ (00 ～ FF Hex)
	8 ～ 15	*_Skt1RemoteIPAdr3	相手 IP アドレスの 3 バイトめ (00 ～ FF Hex)
m+21CH	0 ～ 15	*_Skt1RemotePortNo	相手 UDP/TCP ポート No.(0000 ～ FFFF Hex)
m+22CH	0 ～ 15	*_Skt1SendRcvByte	送信／受信バイト数 (0000 ～ 07C0 Hex(1984))
m+23CH	0 ～ 7	*_Skt1SendRcvAdrCh1	送信／受信データアドレス（チャネルアドレスの上位 2 けた）
	8 ～ 15	*_Skt1SendRcvAdrType	送信／受信データアドレス（エリア種別）
m+24CH	0 ～ 7	－	－
	8 ～ 15	*_Skt1SendRcvAdrCh2	送信／受信データアドレス（チャネルアドレスの下位 2 けた）
m+25CH	0 ～ 15	*_Skt1Timeout	タイムアウト値 (0000 ～ FFFF Hex)
m+26CH	0 ～ 15	*_Skt1RespCode	終了コード

以降、ソケットNo.2～No.8も同様の項目となります。開始のCH番号は順次+10ずつ加算されます。
（ソケット No.2 ～ No.7 については表の記載を省略します。）

ソケット No.8

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
m+87CH	0 ～ 7	*_Skt8No	UDP/TCP ソケット No.
	8	*_Skt8KeepAlive	Keep-alive 機能使用
	9	*_Skt8Linger	Linger 機能使用
	10 ～ 15	－	－
m+88CH	0 ～ 15	*_Skt8LocalPortNo	自 UDP/TCP ポート No.(0000 ～ FFFF Hex)
m+89CH	0 ～ 7	*_Skt8RemoteIPAdr2	相手 IP アドレスの 2 バイトめ (00 ～ FF Hex)
	8 ～ 15	*_Skt8RemoteIPAdr1	相手 IP アドレスの 1 バイトめ (00 ～ FF Hex)
m+90CH	0 ～ 7	*_Skt8RemoteIPAdr4	相手 IP アドレスの 4 バイトめ (00 ～ FF Hex)
	8 ～ 15	*_Skt8RemoteIPAdr3	相手 IP アドレスの 3 バイトめ (00 ～ FF Hex)
m+91CH	0 ～ 15	*_Skt8RemotePortNo	相手 UDP/TCP ポート No.(0000 ～ FFFF Hex)
m+92CH	0 ～ 15	*_Skt8SendRcvByte	送信／受信バイト数 (0000 ～ 07C0 Hex(1984))
m+93CH	0 ～ 7	*_Skt8SendRcvAdrCh1	送信／受信データアドレス（チャネルアドレスの上位 2 けた）
	8 ～ 15	*_Skt8SendRcvAdrType	送信／受信データアドレス（エリア種別）
m+94CH	0 ～ 7	－	－
	8 ～ 15	*_Skt8SendRcvAdrCh2	送信／受信データアドレス（チャネルアドレスの下位 2 けた）

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
m+95CH	0 ～ 15	*_Skt8Timeout	タイムアウト値 (0000 ～ FFFF Hex)
m+96CH	0 ～ 15	*_Skt8RespCode	終了コード

ユーザ設定エリア

x：割付リレーエリアに指定した I/O メモリエリア

● x ～ x+15CH（登録ターゲットノードテーブル）

x ～ x + 15CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
xCH ～ x+15CH	各チャネルの 0 ～ 15	*_x.RegTargetSta.TargetStaWd	ターゲットノードの登録情報 (各チャネルのビット 0 ～ 15 の機能と右記 CJ ユニットデバイス変数の各ビットとの対応は以下です。 • xCH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 0 ～ 15 • x+1CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 16 ～ 31 • x+2CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 32 ～ 47 • x+3CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 48 ～ 63 • x+4CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 64 ～ 79 • x+5CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 80 ～ 95 • x+6CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 96 ～ 111 • x+7CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 112 ～ 127 • x+8CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 128 ～ 143 • x+9CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 144 ～ 159 • x+10CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 160 ～ 175 • x+11CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 176 ～ 191 • x+12CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 192 ～ 207 • x+13CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 208 ～ 223 • x+14CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 224 ～ 239 • x+15CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 240 ～ 255

x ～ x+15CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
xCH	0	*_x.RegTargetSta.TargetSta[0]	ノードアドレス 0 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[15]	ノードアドレス 15 のターゲットノードの登録情報
x+1CH	0	*_x.RegTargetSta.TargetSta[16]	ノードアドレス 16 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[31]	ノードアドレス 31 のターゲットノードの登録情報
x+2CH	0	*_x.RegTargetSta.TargetSta[32]	ノードアドレス 32 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[47]	ノードアドレス 47 のターゲットノードの登録情報
x+3CH	0	*_x.RegTargetSta.TargetSta[48]	ノードアドレス 48 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[63]	ノードアドレス 63 のターゲットノードの登録情報
x+4CH	0	*_x.RegTargetSta.TargetSta[64]	ノードアドレス 64 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[79]	ノードアドレス 79 のターゲットノードの登録情報
x+5CH	0	*_x.RegTargetSta.TargetSta[80]	ノードアドレス 80 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[95]	ノードアドレス 95 のターゲットノードの登録情報
x+6CH	0	*_x.RegTargetSta.TargetSta[96]	ノードアドレス 96 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[111]	ノードアドレス 111 のターゲットノードの登録情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+7CH	0	*_x.RegTargetSta.TargetSta[112]	ノードアドレス 112 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[127]	ノードアドレス 127 のターゲットノードの登録情報
x+8CH	0	*_x.RegTargetSta.TargetSta[128]	ノードアドレス 128 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[143]	ノードアドレス 143 のターゲットノードの登録情報
x+9CH	0	*_x.RegTargetSta.TargetSta[144]	ノードアドレス 144 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[159]	ノードアドレス 159 のターゲットノードの登録情報
x+10CH	0	*_x.RegTargetSta.TargetSta[160]	ノードアドレス 160 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[175]	ノードアドレス 175 のターゲットノードの登録情報
x+11CH	0	*_x.RegTargetSta.TargetSta[176]	ノードアドレス 176 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[191]	ノードアドレス 191 のターゲットノードの登録情報
x+12CH	0	*_x.RegTargetSta.TargetSta[192]	ノードアドレス 192 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[207]	ノードアドレス 207 のターゲットノードの登録情報
x+13CH	0	*_x.RegTargetSta.TargetSta[208]	ノードアドレス 208 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[223]	ノードアドレス 223 のターゲットノードの登録情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+14CH	0	*_x.RegTargetSta.TargetSta[224]	ノードアドレス 224 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[239]	ノードアドレス 239 のターゲットノードの登録情報
x+15CH	0	*_x.RegTargetSta.TargetSta[240]	ノードアドレス 240 のターゲットノードの登録情報
	. . .	. . .	. . .
	15	*_x.RegTargetSta.TargetSta[255]	ノードアドレス 255 のターゲットノードの登録情報

● x+16 ～ x+31CH（正常ターゲットノードテーブル）

x+16 ～ x + 31CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+16CH ～ x+31CH	各チャネルの 0 ～ 15	*_x.EstbTargetSta.TargetStaWd	正常ターゲットノードとのコネクション確立情報（各チャネルのビット 0 ～ 15 の機能と右記 CJ ユニットデバイス変数の各ビットとの対応は以下です。 • x+16CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 0 ～ 15 • x+17CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 16 ～ 31 • x+18CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 32 ～ 47 • x+19CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 48 ～ 63 • x+20CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 64 ～ 79 • x+21CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 80 ～ 95 • x+22CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 96 ～ 111 • x+23CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 112 ～ 127 • x+24CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 128 ～ 143 • x+25CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 144 ～ 159 • x+26CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 160 ～ 175 • x+27CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 176 ～ 191 • x+28CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 192 ～ 207 • x+29CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 208 ～ 223 • x+30CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 224 ～ 239 • x+31CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 240 ～ 255

x+16 ～ x+31CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		**NJ シリーズの CJ ユニットデバイス変数**	
CH 番号	**ビット番号**	**変数名**	**名称**
x+16CH	0	*_x.EstbTargetSta.TargetSta[0]	ノードアドレス 0 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[15]	ノードアドレス 15 のターゲットノードとの確立情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+17CH	0	*_x.EstbTargetSta.TargetSta[16]	ノードアドレス 16 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[31]	ノードアドレス 31 のターゲットノードとの確立情報
x+18CH	0	*_x.EstbTargetSta.TargetSta[32]	ノードアドレス 32 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[47]	ノードアドレス 47 のターゲットノードとの確立情報
x+19CH	0	*_x.EstbTargetSta.TargetSta[48]	ノードアドレス 48 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[63]	ノードアドレス 63 のターゲットノードとの確立情報
x+20CH	0	*_x.EstbTargetSta.TargetSta[64]	ノードアドレス 64 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[79]	ノードアドレス 79 のターゲットノードとの確立情報
x+21CH	0	*_x.EstbTargetSta.TargetSta[80]	ノードアドレス 80 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[95]	ノードアドレス 95 のターゲットノードとの確立情報
x+22CH	0	*_x.EstbTargetSta.TargetSta[96]	ノードアドレス 96 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[111]	ノードアドレス 111 のターゲットノードとの確立情報
x+23CH	0	*_x.EstbTargetSta.TargetSta[112]	ノードアドレス 112 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[127]	ノードアドレス 127 のターゲットノードとの確立情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+24CH	0	*_x.EstbTargetSta.TargetSta[128]	ノードアドレス 128 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[143]	ノードアドレス 143 のターゲットノードとの確立情報
x+25CH	0	*_x.EstbTargetSta.TargetSta[144]	ノードアドレス 144 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[159]	ノードアドレス 159 のターゲットノードとの確立情報
x+26CH	0	*_x.EstbTargetSta.TargetSta[160]	ノードアドレス 160 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[175]	ノードアドレス 175 のターゲットノードとの確立情報
x+27CH	0	*_x.EstbTargetSta.TargetSta[176]	ノードアドレス 176 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[191]	ノードアドレス 191 のターゲットノードとの確立情報
x+28CH	0	*_x.EstbTargetSta.TargetSta[192]	ノードアドレス 192 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[207]	ノードアドレス 207 のターゲットノードとの確立情報
x+29CH	0	*_x.EstbTargetSta.TargetSta[208]	ノードアドレス 208 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[223]	ノードアドレス 223 のターゲットノードとの確立情報
x+30CH	0	*_x.EstbTargetSta.TargetSta[224]	ノードアドレス 224 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[239]	ノードアドレス 239 のターゲットノードとの確立情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+31CH	0	*_x.EstbTargetSta.TargetSta[240]	ノードアドレス 240 のターゲットノードとの確立情報
	. . .	. . .	. . .
	15	*_x.EstbTargetSta.TargetSta[255]	ノードアドレス 255 のターゲットノードとの確立情報

● x+32 ～ x+47CH（ターゲットノード PLC 本体運転情報）

x+32 ～ x + 47CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+32CH ～ x+47CH	各チャネルの 0 ～ 15	*_x.TargetPLCMdSta.TargetStaWd	ターゲットノード PLC 本体運転情報 (各チャネルのビット 0 ～ 15 の機能と右記 CJ ユニットデバイス変数の各ビットとの対応は以下です。 • x+32CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 0 ～ 15 • x+33CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 16 ～ 31 • x+34CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 32 ～ 47 • x+35CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 48 ～ 63 • x+36CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 64 ～ 79 • x+37CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 80 ～ 95 • x+38CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 96 ～ 111 • x+39CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 112 ～ 127 • x+40CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 128 ～ 143 • x+41CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 144 ～ 159 • x+42CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 160 ～ 175 • x+43CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 176 ～ 191 • x+44CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 192 ～ 207 • x+45CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 208 ～ 223 • x+46CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 224 ～ 239 • x+47CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 240 ～ 255

x+32 ～ x+47CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+32CH	0	*_x.TargetPLCMdSta.TargetSta[0]	ノードアドレス 0 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[15]	ノードアドレス 15 の PLC 本体運転情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+33CH	0	*_x.TargetPLCMdSta.TargetSta[16]	ノードアドレス 16 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[31]	ノードアドレス 31 の PLC 本体運転情報
x+34CH	0	*_x.TargetPLCMdSta.TargetSta[32]	ノードアドレス 32 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[47]	ノードアドレス 47 の PLC 本体運転情報
x+35CH	0	*_x.TargetPLCMdSta.TargetSta[48]	ノードアドレス 48 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[63]	ノードアドレス 63 の PLC 本体運転情報
x+36CH	0	*_x.TargetPLCMdSta.TargetSta[64]	ノードアドレス 64 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[79]	ノードアドレス 79 の PLC 本体運転情報
x+37CH	0	*_x.TargetPLCMdSta.TargetSta[80]	ノードアドレス 80 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[95]	ノードアドレス 95 の PLC 本体運転情報
x+38CH	0	*_x.TargetPLCMdSta.TargetSta[96]	ノードアドレス 96 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[111]	ノードアドレス 111 の PLC 本体運転情報
x+39CH	0	*_x.TargetPLCMdSta.TargetSta[112]	ノードアドレス 112 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[127]	ノードアドレス 127 の PLC 本体運転情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+40CH	0	*_x.TargetPLCMdSta.TargetSta[128]	ノードアドレス 128 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[143]	ノードアドレス 143 の PLC 本体運転情報
x+41CH	0	*_x.TargetPLCMdSta.TargetSta[144]	ノードアドレス 144 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[159]	ノードアドレス 159 の PLC 本体運転情報
x+42CH	0	*_x.TargetPLCMdSta.TargetSta[160]	ノードアドレス 160 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[175]	ノードアドレス 175 の PLC 本体運転情報
x+43CH	0	*_x.TargetPLCMdSta.TargetSta[176]	ノードアドレス 176 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[191]	ノードアドレス 191 の PLC 本体運転情報
x+44CH	0	*_x.TargetPLCMdSta.TargetSta[192]	ノードアドレス 192 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[207]	ノードアドレス 207 の PLC 本体運転情報
x+45CH	0	*_x.TargetPLCMdSta.TargetSta[208]	ノードアドレス 208 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[223]	ノードアドレス 223 の PLC 本体運転情報
x+46CH	0	*_x.TargetPLCMdSta.TargetSta[224]	ノードアドレス 224 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[239]	ノードアドレス 239 の PLC 本体運転情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+47CH	0	*_x.TargetPLCMdSta.TargetSta[240]	ノードアドレス 240 の PLC 本体運転情報
	. . .	. . .	. . .
	15	*_x.TargetPLCMdSta.TargetSta[255]	ノードアドレス 255 の PLC 本体運転情報

● x+48 ～ x+63CH（ターゲットノード PLC 本体異常情報）

x+48 ～ x+63CH の全ビットに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+48CH ～ x+63CH	各チャネルの 0 ～ 15	*_x.TargetPLCErrSta.TargetStaWd	ターゲットノード PLC 本体異常情報 (各チャネルのビット 0 ～ 15 の機能と右記 CJ ユニットデバイス変数の各ビットとの対応は以下です。 • x+48CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 0 ～ 15 • x+49CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 16 ～ 31 • x+50CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 32 ～ 47 • x+51CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 48 ～ 63 • x+52CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 64 ～ 79 • x+53CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 80 ～ 95 • x+54CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 96 ～ 111 • x+55CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 112 ～ 127 • x+56CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 128 ～ 143 • x+57CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 144 ～ 159 • x+58CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 160 ～ 175 • x+59CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 176 ～ 191 • x+60CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 192 ～ 207 • x+61CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 208 ～ 223 • x+62CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 224 ～ 239 • x+63CH のビット 0 ～ 15 と右記 CJ ユニットデバイス変数のビット 240 ～ 255

x+48 ～ x+63CH のビット 0 ～ 15 それぞれに対応する CJ ユニットデバイス変数は以下です。

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+48CH	0	*_x.TargetPLCErrSta.TargetSta[0]	ノードアドレス 0 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[15]	ノードアドレス 15 の PLC 本体異常情報
x+49CH	0	*_x.TargetPLCErrSta.TargetSta[16]	ノードアドレス 16 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[31]	ノードアドレス 31 の PLC 本体異常情報
x+50CH	0	*_x.TargetPLCErrSta.TargetSta[32]	ノードアドレス 32 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[47]	ノードアドレス 47 の PLC 本体異常情報
x+51CH	0	*_x.TargetPLCErrSta.TargetSta[48]	ノードアドレス 48 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[63]	ノードアドレス 63 の PLC 本体異常情報
x+52CH	0	*_x.TargetPLCErrSta.TargetSta[64]	ノードアドレス 64 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[79]	ノードアドレス 79 の PLC 本体異常情報
x+53CH	0	*_x.TargetPLCErrSta.TargetSta[80]	ノードアドレス 80 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[95]	ノードアドレス 95 の PLC 本体異常情報
x+54CH	0	*_x.TargetPLCErrSta.TargetSta[96]	ノードアドレス 96 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[111]	ノードアドレス 111 の PLC 本体異常情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+55CH	0	*_x.TargetPLCErrSta.TargetSta[112]	ノードアドレス 112 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[127]	ノードアドレス 127 の PLC 本体異常情報
x+56CH	0	*_x.TargetPLCErrSta.TargetSta[128]	ノードアドレス 128 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[143]	ノードアドレス 143 の PLC 本体異常情報
x+57CH	0	*_x.TargetPLCErrSta.TargetSta[144]	ノードアドレス 144 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[159]	ノードアドレス 159 の PLC 本体異常情報
x+58CH	0	*_x.TargetPLCErrSta.TargetSta[160]	ノードアドレス 160 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[175]	ノードアドレス 175 の PLC 本体異常情報
x+59CH	0	*_x.TargetPLCErrSta.TargetSta[176]	ノードアドレス 176 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[191]	ノードアドレス 191 の PLC 本体異常情報
x+60CH	0	*_x.TargetPLCErrSta.TargetSta[192]	ノードアドレス 192 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[207]	ノードアドレス 207 の PLC 本体異常情報
x+61CH	0	*_x.TargetPLCErrSta.TargetSta[208]	ノードアドレス 208 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[223]	ノードアドレス 223 の PLC 本体異常情報

CJ シリーズの I/O メモリアドレス		NJ シリーズの CJ ユニットデバイス変数	
CH 番号	ビット番号	変数名	名称
x+62CH	0	*_x.TargetPLCErrSta.TargetSta[224]	ノードアドレス 224 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[239]	ノードアドレス 239 の PLC 本体異常情報
x+63CH	0	*_x.TargetPLCErrSta.TargetSta[240]	ノードアドレス 240 の PLC 本体異常情報
	. . .	. . .	. . .
	15	*_x.TargetPLCErrSta.TargetSta[255]	ノードアドレス 255 の PLC 本体異常情報

A-8 形 CJ1W-EIP21 から形 CJ1W-EIP21S を使用したシステムへの置き換え

A-8-1 仕様や機能などの差異

ここでは、形 CJ1W-EIP21 の形 CJ1W-EIP21S への置き換えによる仕様や機能などの差異、および制約事項を示します。形 CJ1W-EIP21 と形 CJ1W-EIP21S の仕様や機能などの差異、および制約事項を確認し、置き換え元のシステムに対するハードウェア、ユーザプログラム、およびアプリケーションに対する影響を確認します。
以下の項目ごとに説明します。

項目	参照先
ユニット機能と仕様の差異	「 ユニット機能と仕様の差異」(P.A-101)
メモリ割付の差異	「 メモリ割付の差異」(P.A-104)
制約事項	「 制約事項」(P.A-105)

参考

ロット番号 241001□ 以降の形 CJ1W-EIP21S が NJ シリーズ CPU ユニットに装着できます。これに該当するユニットは前面右下に「+NJ」または「+CJ/CP/NJ」と印字されています。

ユニット機能と仕様の差異

差異がある機能と仕様および影響を以下に示します。

項目		置き換え元	置き換え先	置き換えの影響
		形 CJ1W-EIP21	形 CJ1W-EIP21S ロット番号 241001□ 以降	
質量		94g 以下	91g 以下	質量が 3g 減少します。影響ありません。
消費電流		DC5V 0.41A 以下	DC5V 0.65A 以下	0.24A 増加します。電源ユニットの供給が不足する場合は、増設ラックを使用したシステム構成への変更が必要となります。
適用 CPU ユニット		NJ シリーズ CPU ユニット Ver.1.01 以降	NJ シリーズ CPU ユニット Ver.1.67 以降	影響ありません。
CPU ユニットでの占有エリア	CPU 高機能ユニット割付リレーエリア	ソケットサービス用の機能なし	ソケットサービス用の機能あり	占有エリアのサイズに差異はありません。ただし、置き換え元の予約エリアや未使用エリアに、ソケットサービス用の機能が追加されています。割付や影響の詳細については、「 メモリ割付の差異」(P.A-104) を参照してください。
	CPU 高機能ユニット割付 DM エリア			

項目		置き換え元	置き換え先	置き換えの影響
		形 CJ1W-EIP21	形 CJ1W-EIP21S ロット番号 241001□ 以降	
CIP 通信サービス機能：タグデータリンク（サイクリック通信）	ユニット許容通信帯域	• 12000pps（ユニット Ver.3.0 以降） • 6000pps（ユニット Ver.2.1 以前）	12000pps	影響ありません。
	CPU ユニット 1 タスク周期あたりにリフレッシュ可能なタグ最大数	出力／送信（CPU → EtherNet/IP）時：19 入力／受信（EtherNet/IP → CPU）時：20 （ただし、割付リレーエリアのレイアウトをユーザ設定で使用時は、入力／受信は 19 個）	出力／送信（CPU → EtherNet/IP）時：19 入力／受信（EtherNet/IP → CPU）時：20 （ただし、割付リレーエリアのレイアウトをユーザ設定で使用時は、入力／受信、出力／送信ともに 18 個）	割付リレーエリアのレイアウトをユーザ設定で使用時は、入力／受信データのタグ最大数が減ります。最大数を超える場合は、CPU ユニットとのリフレッシュは複数周期にまたがります。通信性能に対する影響を確認してください。
	CPU ユニット 1 タスク周期あたりにリフレッシュ可能なデータサイズ	出力／送信（CPU → EtherNet/IP）時：7469CH 入力／受信（EtherNet/IP → CPU）時：7469CH （ただし、割付リレーエリアのレイアウトをユーザ設定で使用時は、入出力ともに 7405CH）	出力／送信（CPU → EtherNet/IP）時：7469CH 入力／受信（EtherNet/IP → CPU）時：7469CH （ただし、割付リレーエリアのレイアウトをユーザ設定で使用時は、出力は 7321CH、入力は 7385CH）	割付リレーエリアのレイアウトをユーザ設定で使用時は、データサイズが減ります。最大数を超える場合は、CPU ユニットとのリフレッシュは複数周期にまたがります。通信性能に対する影響を確認してください。
FINS 通信サービス機能	FINS/UDP を使用する／使用しない設定とデフォルト	• 設定機能：なし • デフォルト：使用する	• 設定機能：あり デフォルト：使用しない	FINS 機能を使用する場合は、［FINS 設定］ダイアログの［FINS/UDP］を［使用する］に設定する必要があります。
	FINS/TCP を使用する／使用しない設定とデフォルト	• 設定機能：なし • デフォルト：使用する	• 設定機能：あり デフォルト：使用しない	FINS 機能を使用する場合は、［FINS 設定］ダイアログの［FINS/TCP］を［使用する］に設定する必要があります。
	Keep-alive 監視時間の設定	Keep-alive 機能有効時、5 分に固定	［TCP/IP］タブで時間の設定可能。デフォルトは 5 分。	影響ありません。
FTP サーバ機能		あり（ユニット Ver.2.0 以降）	あり	影響ありません。

項目		置き換え元	置き換え先	置き換えの影響
		形 CJ1W-EIP21	形 CJ1W-EIP21S ロット番号 241001□ 以降	
	FTP サーバを使用する／使用しないのデフォルト	デフォルト：使用する	デフォルト：使用しない	FTP 機能を使用する場合は、［FTP 設定］ダイアログの［FTP サーバ］を［使用する］に設定する必要があります。Sysmac Studio の設定の引継ぎ支援機能を使用した場合は、置き換え元の設定がそのまま置き換え先に引き継がれるため手動での設定は不要です。
	ポート No.	22, 443 使用可	22, 443 使用不可	置き換え元で 22 または 443 を使用している場合は、22, 443 以外の値に変更してください。
	ユーザ名／パスワード	• ユーザ名 半角英数字 1 ～ 12 文字 デフォルト：CONFIDENTIAL • パスワード 半角英数字 1 ～ 8 文字 デフォルト：－（設定なし） 入力された文字を表示	• ユーザ名 半角英数字 1 ～ 16 文字 デフォルト：－（設定なし） • パスワード 形半角英数字 8 ～ 16 文字 デフォルト：－（設定なし） 入力された文字を伏字で表示	FTP 機能を使用する場合は、［FTP 設定］ダイアログの［ポート No.］を［使用する］に設定する必要があります。ユーザ名（ログイン名）／パスワードの再設定が必要です。Sysmac Studio の設定の引継ぎ支援機能を使用した場合は、置き換え元のユーザ名はそのまま置き換え先に引き継がれるため手動での設定は不要です。パスワードは手動で設定してください。
時計情報自動調整機能		あり（ユニット Ver.2.0 以降）	あり	影響ありません。
BOOTP クライアント機能	電源の投入またはリスタートするごとに、IP アドレスを BOOTP サーバから取得する	なし	あり	影響ありません。
バックアップ機能	バックアップのデータ容量	459K バイト（簡易バックアップ機能）	364K バイト（SD メモリカードバックアップ機能、Sysmac Studio コントローラバックアップ機能）	影響ありません。
SNMP/SNMP トラップ		あり（ユニット Ver.2.0 以降） コミュニティ名：入力された文字を表示	あり コミュニティ名：入力された文字を伏字で表示	影響ありません。
IP アドレス CIDR 機能		あり（ユニット Ver.2.0 以降）	あり	影響ありません。
CIP メッセージサーバを使用する／使用しないの設定		• 設定機能：なし • デフォルト：使用する	• 設定機能：あり デフォルト：使用する	影響ありません。

項目		置き換え元	置き換え先	置き換えの影響
		形 CJ1W-EIP21	形 CJ1W-EIP21S ロット番号 241001□ 以降	
セキュリティ機能	Packet Filter	なし	あり	置き換え後、セキュリティリスクの低減を行うために、お客さまのシステム運用に合わせて、本機能を使用するしないを判断ください。
	ポートのオープン／クローズ機能*1	以下のサービスに対するポートのオープン／クローズ機能なし。 • FINS/UDP サービス • FINS/TCP サービス • CIP メッセージサーバ	以下のサービスに対するポートのオープン／クローズ機能あり。 • FINS/UDP サービス • FINS/TCP サービス • CIP メッセージサーバ	置き換え前に使用しているサービスのポートを使用できるように設定する必要があります。
ソケットサービス機能		なし	あり	影響ありません。

*1. 以下のサービスに対するポートのオープン／クローズ機能は形 CJ1W-EIP21 と形 CJ1W-EIP21S の両方でサポートしています。
FTP サーバ、SNMP

デバイス変数の差異

項目	置き換え元	置き換え先	置き換えの影響
	形 CJ1W-EIP21	形 CJ1W-EIP21S ロット番号 241001□ 以降	
ソケットサービス用のデバイス変数	なし	あり	影響ありません。
ステータスレイアウトを［ユーザ設定］にしたときのタグデータリンクのステータスを示す以下の変数 • TargetPLCMdSta（ターゲットノード PLC 本体運転情報） • TargetPLCErrSta（ターゲットノード PLC 本体異常情報） • RegTargetSta（登録ターゲットノード） • EstbTargetSta（正常ターゲットノード）	（予約）	不定	形 CJ1W-EIP21S でソケットサービスを使用する場合は、ステータスレイアウトを［ユーザ設定］で使用してください。 このとき、以下のステータスを示す変数は、デフォルトの変数ではなく、ステータスエリア設定の割付エリアで設定したユーザ定義変数を参照してください。 • TargetPLCMdSta（ターゲットノード PLC 本体運転情報） • TargetPLCErrSta（ターゲットノード PLC 本体異常情報） • RegTargetSta（登録ターゲットノード） EstbTargetSta（正常ターゲットノード）

メモリ割付の差異

形 CJ1W-EIP21 と形 CJ1W-EIP21S のメモリ割り付けで差異がある機能を以下に示します。差異がある機能の詳細については、『CS/CJ シリーズ EtherNet/IP ユニット ユーザーズマニュアル (SBCD-342)』を参照してください。

項目			置き換え元	置き換え先	置き換えの影響
			形CJ1W-EIP21	形CJ1W-EIP21S ロット番号241001□以降	
CPU高機能ユニット割付リレーエリア					
	ユーザ設定のエリア	n+1～n+4CH	予約	ソケットサービス要求スイッチ（CPUユニット→ユニット）	割付リレーエリアのレイアウトをユーザ設定で使用時は、置き換え元の予約エリアにソケットサービス用の機能が追加されています。ユーザプログラムで予約エリアを操作していないことを確認してください。
		n+16～n+23CH	予約	UDPソケットNo.□ステータスおよびTCPソケットNo.□ステータス（ユニット→CPUユニット）	
	ソフトスイッチ（CPUユニット→ユニット）（nCH）	ビット10	未使用	ソケット強制クローズスイッチ（CPUユニット→ユニット）	置き換え先では、ソケットサービス用の機能が追加されています。ユーザプログラムで予約ビットを操作していないことを確認してください。
	ユニットステータス2（ユニット→CPUユニット）（n+11CH）	ビット10	未使用	予約	
		ビット12、ビット13	未使用	予約	
CPU高機能ユニット割付DMエリア					
	m+1～m+8CH		未使用	TCPソケットNo.□受信データバイト数（ユニット→CPUユニット）	置き換え先では、ソケットサービス用の機能が追加されています。ユーザプログラムで該当エリアを使用している場合は、データの意味が変わってしまいます。そのため、置き換え前に使用していたデータを別の空いているDMエリアに変更するなどの検討を行ってください。
	m+9～m+16CH		未使用	TCPソケットNo.□コネクション状態（ユニット→CPUユニット）	
	m+17～m+96CH		未使用	ソケットサービスパラメータエリア□（ユニット→CPUユニットまたはCPUユニット→ユニット）	

制約事項

置き換えに際しての制約事項を以下に示します。

項目	内容	置き換えによる影響
バックアップ／リストア	形 CJ1W-EIP21S ではセキュリティ機能の追加に伴い、バックアップする対象データが形 CJ1W-EIP21 よりも増えています。また形 CJ1W-EIP21 と同じ設定項目でも、デフォルト値が異なっている設定があります。	Sysmac Studio でバックアップした形 CJ1W-EIP21 のバックアップデータを、形 CJ1W-EIP21S にリストアできません。Sysmac Studio で形 CJ1W-EIP21 と同じように設定し直す必要があります。設定方法については、「形 CJ1W-EIP21S の設定」(P.A-107) を参照してください。
ユニットの起動時間	形 CJ1W-EIP21 に比べ、形 CJ1W-EIP21S は起動時間が長くなります。これにより、置き換え前のシステムと比べ、CPU ユニットの運転開始時間が数秒遅くなります。	システムの立ち上げ動作への影響の確認が必要となります。

A-8-2　置き換えフロー

形 CJ1W-EIP21 を使用したシステムの形 CJ1W-EIP21S を使用したシステムへの置き換えフローを以下に示します。事前準備のフローと置き換えの実行フローを説明します。

事前準備のフロー

置き換え前の準備の内容を以下に示します。このフローで置き換えの検討や置き換え後のユニット構成に合わせた各データを準備します。
置き換え先のユニットは、置き換えを実行する前までに入手してください。

手順	項目	内容	参照先
1	仕様や機能などの差異の確認	置き換え元と置き換え先機種の仕様や機能などの差異、および制約事項を確認し、置き換え元のシステムに対するハードウェア、ユーザプログラム、およびアプリケーションに対する影響を確認します。	「仕様や機能などの差異の確認」(P.A-107)
2	サポートソフトの入手	事前準備や置き換えの実行で使用するサポートソフトを入手します。	「サポートソフトの入手」(P.A-107)
3	データの読み出しと保存	置き換え元のシステムから以下のデータを読み出して、保存します。 本作業で読み出したデータを使用し、以降の作業で置き換え先のシステムに合わせてデータの変更や設定を行います。 • コントローラのプロジェクトデータ • タグデータリンクパラメータ	「データの読み出しと保存」(P.A-107)
4	形 CJ1W-EIP21S の設定	置き換え後のユニット構成で、形 CJ1W-EIP21S の設定を行います。	「形 CJ1W-EIP21S の設定」(P.A-107)
5	デバイス変数の設定	置き換え後のユニット構成で、形 CJ1W-EIP21S のデバイス変数の設定を行います。	「 デバイス変数の設定」(P.A-109)
6	ユーザプログラムの変更	置き換えによる仕様の差異や制約事項に対応するために、ユーザプログラムを変更します。	「ユーザプログラムの修正」(P.A-114)
7	タグデータリンクの設定	置き換え後のユニット構成に合わせて、タグデータリンクの設定を行います。	「タグデータリンクの設定」(P.A-114)

置き換えの実行フロー

事前準備の完了後に、実際に置き換えを行う内容を示します。置き換え後は必要に応じてセキュリティ施策を実施します。

手順	項目	内容	参照先
8	ユニットの置き換えと配線	置き換え後のユニット構成に合わせて、ユニットを置き換えます。置き換えたユニットの配線を行います。	「ユニットの置き換えと配線」(P.A-114)
9	電源の投入	コントローラに電源を投入します。	「電源の投入」(P.A-114)
10	各データの転送	置き換え後の実機に、事前に準備した以下のデータを CPU ユニットに直接接続したサポートソフトで転送します。 • コントローラのプロジェクトデータ • タグデータリンクパラメータ	「各データの転送」(P.A-114)
11	動作確認	意図どおりにシステムが動作することを確認します。	「動作確認」(P.A-115)
12	セキュリティ施策の実施 *1	セキュリティリスクを低減するためのセキュリティ機能を、形 CJ1W-EIP21S および Sysmac Studio に搭載しています。システム運用に合わせて、セキュリティ機能を使用して、セキュリティリスクの低減を行います。	「セキュリティ施策の実施」(P.A-115)
13	セキュリティ施策の実施後の動作確認	セキュリティ施策を実施した場合は、セキュリティ機能の設定に対し、動作確認を行います。	「セキュリティ施策の実施後の動作確認」(P.A-116)

*1. 必要により実施してください。

A-8-3　事前準備

ここでは、置き換えの実行前に準備を行う内容を説明します。

仕様や機能などの差異の確認

形 CJ1W-EIP21 と形 CJ1W-EIP21S の仕様や機能などの差異、および制約事項を確認し、置き換え元のシステムに対するハードウェア、ユーザプログラム、およびアプリケーションに対する影響を確認します。
詳細については、「A-8-1 仕様や機能などの差異」(P.A-101) を参照してください。

サポートソフトの入手

事前準備や置き換えの実行で使用するサポートソフトを入手します。
「ユニットバージョンと周辺ツールのバージョンについて」(P.38) に示したバージョンのサポートソフトを入手してください。

データの読み出しと保存

置き換え元のシステムから以下のデータを読み出して、パソコン等の任意のフォルダに保存します。
本作業で読み出したデータを使用し、以降の作業で置き換え先のシステムに合わせてデータの変更や設定を行います。

- コントローラのプロジェクトデータ
- タグデータリンクパラメータ

本作業は置き換え元システムのデータの復元を可能にするために実施してください。置き換え元のシステムのデータを所持している場合でも、実際の実機と差異があるとトラブルになるリスクが考えられますので、読み出して、保存しておくことを推奨します。

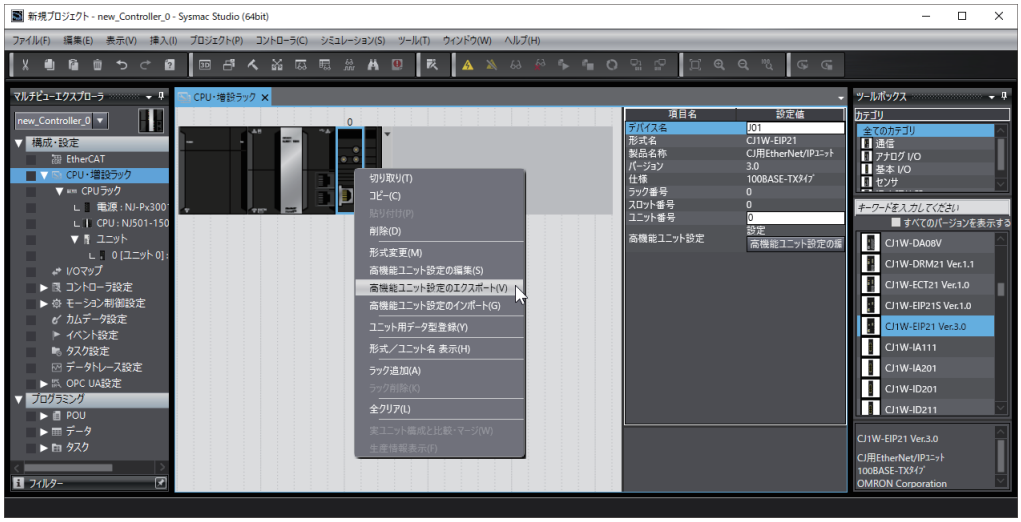
形 CJ1W-EIP21S の設定

置き換え後のユニット構成で、形 CJ1W-EIP21S の設定を行います。
Sysmac　Studio を使用し、形 CJ1W-EIP21S のパラメータを設定します。置き換え元の形 CJ1W-EIP21 の設定に合わせて、設定します。

なお、Sysmac Studio の設定の引継ぎ支援機能を使用することで形 CJ1W-EIP21 の設定を形 CJ1W-EIP21Sに簡単に引き継ぐことができます。この場合、形CJ1W-EIP21になく形CJ1W-EIP21Sにだけある設定項目については形 CJ1W-EIP21S のデフォルト値が設定されます。
引継ぎ支援機能による設定手順は以下のとおりです。

1 **Sysmac Studio で置き換え元のプロジェクトを開き、マルチビューエクスプローラで［構成・設定］–［CPU・増設ラック］をダブルクリックします。**
エディットウィンドウにユニットエディタが表示されます。

2 **形 CJ1W-EIP21 上でマウスを右クリックし、表示されたポップアップメニューから［高機能ユニット設定のエクスポート］を選択します。**

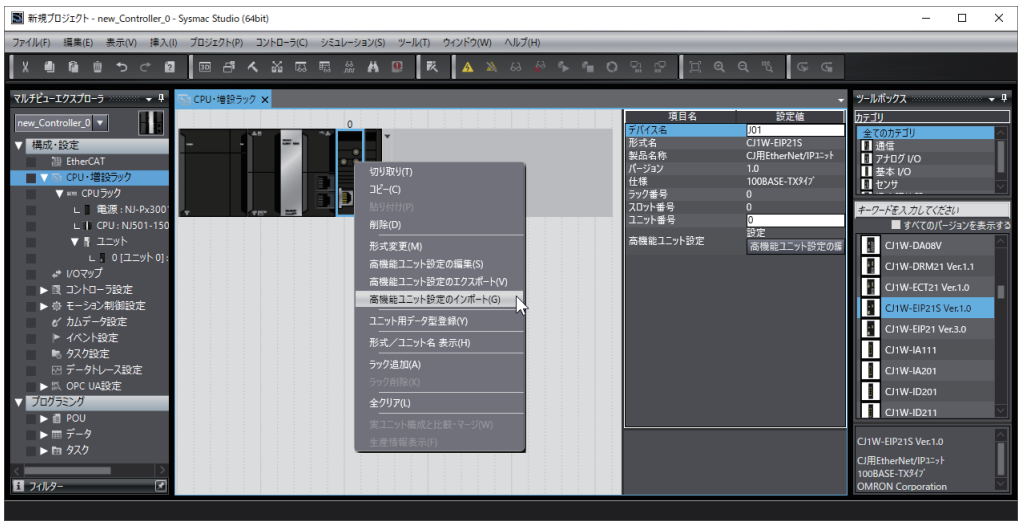


ファイル保存ダイアログが開きます。

3 **ファイル保存ダイアログで任意のファイル名を付けて［保存］ボタンをクリック。**
設定がファイルに保存されます。ファイルの拡張子は upf または upf2 となります。

4 **Sysmac Studio で、形 CJ1W-EIP21S が構成に含まれる置き換え先のプロジェクトを開き、マルチビューエクスプローラで［構成・設定］–［CPU・増設ラック］をダブルクリックします。**

5 **形 CJ1W-EIP21S 上でマウスを右クリックし、表示されたポップアップメニューから［高機能ユニット設定のインポート］を選択します。**



［ファイルを開く］ダイアログが開きます。

6 **手順 3 で保存したファイルを選択して［開く］ボタンをクリックします。**
ファイルに保存されていた設定が形 CJ1W-EIP21S の設定に反映されます。

7 **形 CJ1W-EIP21 にない設定項目で形 CJ1W-EIP21S にある設定項目についてはデフォルト値が設定されていますので、必要により設定します。**

使用上の注意

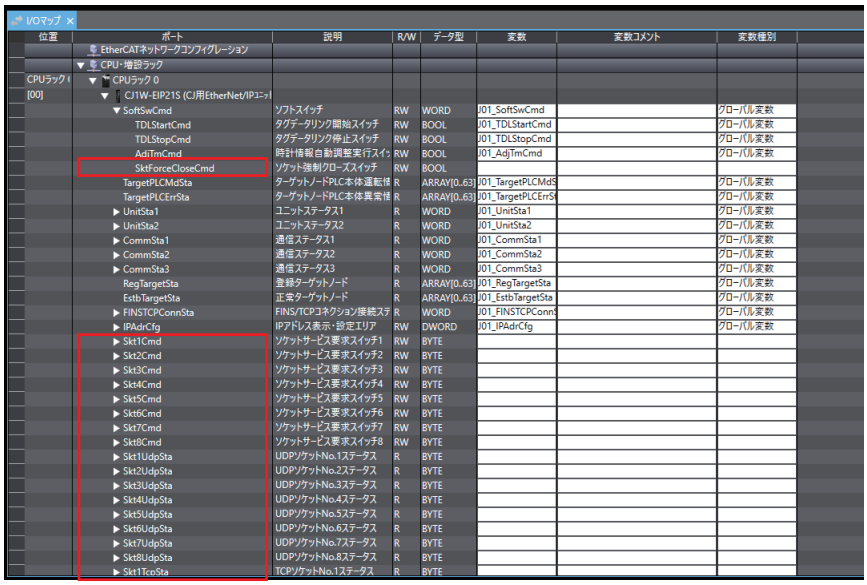
- 形CJ1W-EIP21と形CJ1W-EIP21SではFTPのパスワード長の設定可能範囲が異なるため、形 CJ1W-EIP21S の設定に反映されません。形 CJ1W-EIP21S に新たなパスワードを設定してください。FTP のパスワード長の設定可能範囲は以下のとおりです。
  形 CJ1W-EIP21: 半角英数字 1 ～ 8 文字
  形 CJ1W-EIP21S: 半角英数字 8 ～ 16 文字
- 形 CJ1W-EIP21 で FTP のポート番号に 22 または 443 が指定されている場合は、形 CJ1W-EIP21S のユニット設定でエラーとなります。形 CJ1W-EIP21S での FTP のポート番号を 22, 443 以外に変更してください。

デバイス変数の設定

置き換え後のユニット構成で、形 CJ1W-EIP21S のデバイス変数の設定を行います。
Sysmac Studio を使用し、置き換え元の形 CJ1W-EIP21 のデバイス変数で設定していた変数に合わせて設定します。

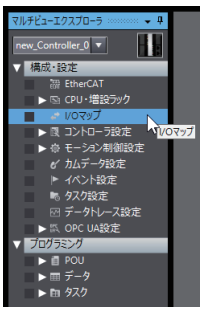
デバイス変数は I/O ポートに対して作成します。形 CJ1W-EIP21 と形 CJ1W-EIP21S の I/O ポートの違いは次のとおりです。

- 形 CJ1W-EIP21S では、［AdjTmCmd］の下に［SktForceCloseCmd］が追加されています。
- 形 CJ1W-EIP21S では、［IPAdrCfg］の下にソケット関連の I/O ポートが追加されています。

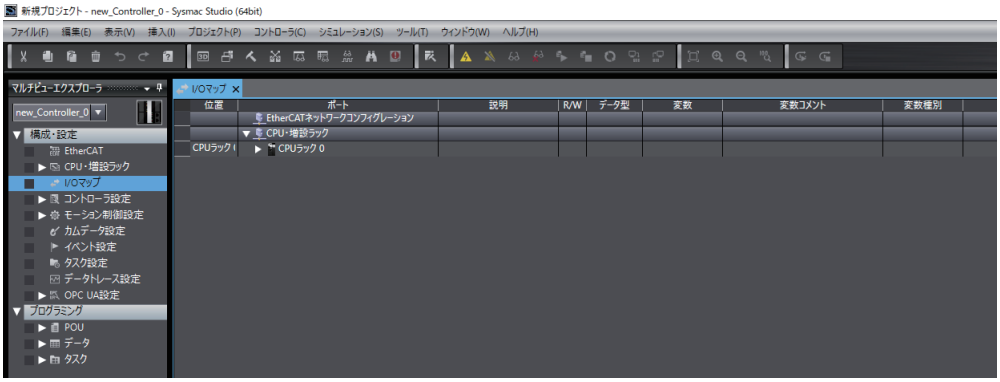


以下の手順で形 CJ1W-EIP21 のデバイス変数を形 CJ1W-EIP21S に移行します。
形 CJ1W-EIP21 から形 CJ1W-EIP21S の置き換えのすべての組で実施してください。

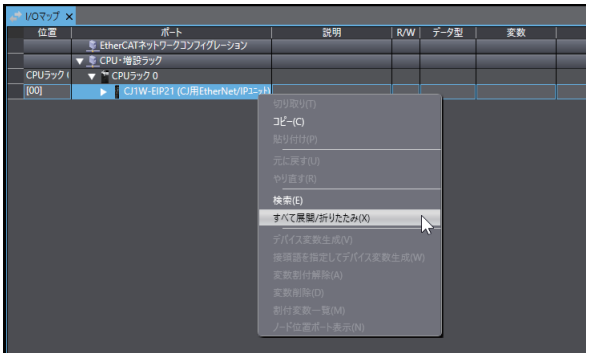
1 **Sysmac Studio で置き換え元のプロジェクトを開き、マルチビューエクスプローラで［構成・設定］－［I/O マップ］をダブルクリックします。**



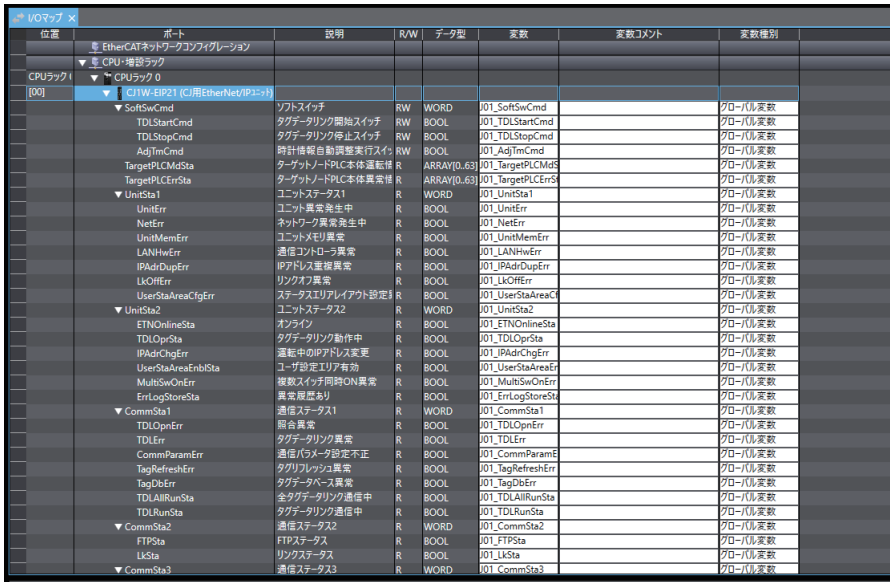
エディットウィンドウに I/O マップが表示されます。



2 **I/O マップの置き換え元の形 CJ1W-EIP21 上で右クリックして［すべて展開／折りたたみ］を選択します。**

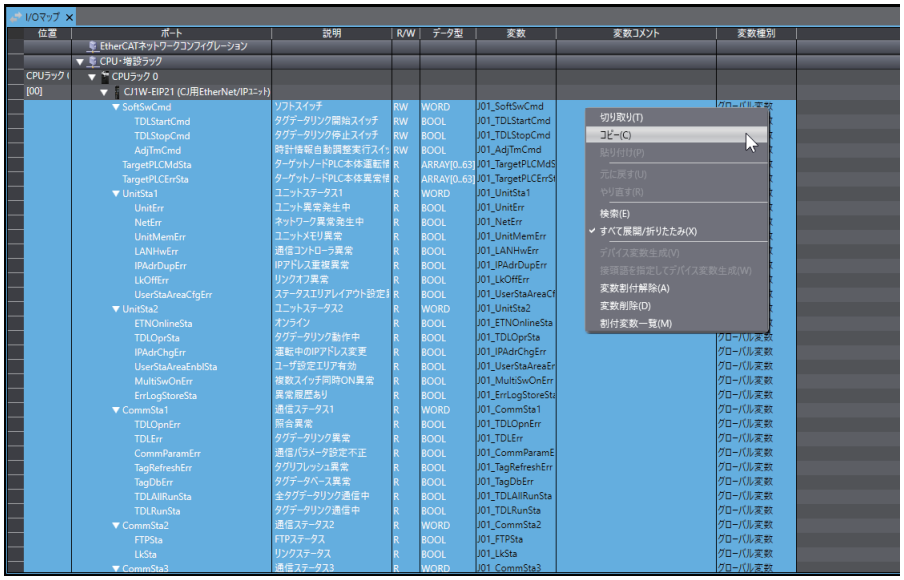


置き換え元の形 CJ1W-EIP21 の I/O ポートが展開されます。



位置	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・増設ラック						
CPUラック(	CPUラック 0						
[00]	CJ1W-EIP21 (CJ用EtherNet/IPユニット)						
	SoftSwCmd	ソフトスイッチ	RW	WORD	J01_SoftSwCmd		グローバル変数
	TDLStartCmd	タグデータリンク開始スイッチ	RW	BOOL	J01_TDLStartCmd		グローバル変数
	TDLStopCmd	タグデータリンク停止スイッチ	RW	BOOL	J01_TDLStopCmd		グローバル変数
	AdjTmCmd	時計情報自動調整実行スイッ	RW	BOOL	J01_AdjTmCmd		グローバル変数
	TargetPLCMdSta	ターゲットノードPLC本体運転情	R	ARRAY[0..63]	J01_TargetPLCMdS		グローバル変数
	TargetPLCErrSta	ターゲットノードPLC本体異常情	R	ARRAY[0..63]	J01_TargetPLCErrS		グローバル変数
	UnitSta1	ユニットステータス1	R	WORD	J01_UnitSta1		グローバル変数
	UnitErr	ユニット異常発生中	R	BOOL	J01_UnitErr		グローバル変数
	NetErr	ネットワーク異常発生中	R	BOOL	J01_NetErr		グローバル変数
	UnitMemErr	ユニットメモリ異常	R	BOOL	J01_UnitMemErr		グローバル変数
	LANHwErr	通信コントローラ異常	R	BOOL	J01_LANHwErr		グローバル変数
	IPAdrDupErr	IPアドレス重複異常	R	BOOL	J01_IPAdrDupErr		グローバル変数
	LkOffErr	リンクオフ異常	R	BOOL	J01_LkOffErr		グローバル変数
	UserStaAreaCfgErr	ステータスエリアレイアウト設定	R	BOOL	J01_UserStaAreaCf		グローバル変数
	UnitSta2	ユニットステータス2	R	WORD	J01_UnitSta2		グローバル変数
	ETNOnlineSta	オンライン	R	BOOL	J01_ETNOnlineSta		グローバル変数
	TDLOprSta	タグデータリンク動作中	R	BOOL	J01_TDLOprSta		グローバル変数
	IPAdrChgErr	運転中のIPアドレス変更	R	BOOL	J01_IPAdrChgErr		グローバル変数
	UserStaAreaEnblSta	ユーザ設定エリア有効	R	BOOL	J01_UserStaAreaEr		グローバル変数
	MultiSwOnErr	複数スイッチ同時ON異常	R	BOOL	J01_MultiSwOnErr		グローバル変数
	ErrLogStoreSta	異常履歴あり	R	BOOL	J01_ErrLogStoreSt		グローバル変数
	CommSta1	通信ステータス1	R	WORD	J01_CommSta1		グローバル変数
	TDLOpnErr	照合異常	R	BOOL	J01_TDLOpnErr		グローバル変数
	TDLErr	タグデータリンク異常	R	BOOL	J01_TDLErr		グローバル変数
	CommParamErr	通信パラメータ設定不正	R	BOOL	J01_CommParamE		グローバル変数
	TagRefreshErr	タグリフレッシュ異常	R	BOOL	J01_TagRefreshErr		グローバル変数
	TagDbErr	タグデータベース異常	R	BOOL	J01_TagDbErr		グローバル変数
	TDLAllRunSta	全タグデータリンク通信中	R	BOOL	J01_TDLAllRunSta		グローバル変数
	TDLRunSta	タグデータリンク通信中	R	BOOL	J01_TDLRunSta		グローバル変数
	CommSta2	通信ステータス2	R	WORD	J01_CommSta2		グローバル変数
	FTPSta	FTPステータス	R	BOOL	J01_FTPSta		グローバル変数
	LkSta	リンクステータス	R	BOOL	J01_LkSta		グローバル変数
	CommSta3	通信ステータス3	R	WORD	J01_CommSta3		グローバル変数

3 置き換え元の形 CJ1W-EIP21 の全ての I/O ポートを選択してコピーします。



4 コピーしたデータを表計算ソフトに貼り付けます。

	A	B	C	D	E	F	G	H	I	J	K
1	位置	ポート	説明	R/W	データ型	変数	変数コメント	変数種別			
2		SoftSwCmd	ソフトスイッチ	RW	WORD	J01_SoftSwCmd		グローバル変数			
3		TDLStartCmd	タグデータリンク開始スイッチ	RW	BOOL	J01_TDLStartCmd		グローバル変数			
4		TDLStopCmd	タグデータリンク停止スイッチ	RW	BOOL	J01_TDLStopCmd		グローバル変数			
5		AdjTmCmd	時計情報自動調整実行スイッチ	RW	BOOL	J01_AdjTmCmd		グローバル変数			
6		TargetPLCMdSta	ターゲットノード PLC 本体運転情報	R	ARRAY[0..63] OF BOOL	J01_TargetPLCMdSta		グローバル変数			
7		TargetPLCErrSta	ターゲットノード PLC 本体異常情報	R	ARRAY[0..63] OF BOOL	J01_TargetPLCErrSta		グローバル変数			
8		UnitSta1	ユニットステータス 1	R	WORD	J01_UnitSta1		グローバル変数			
9		UnitErr	ユニット異常発生中	R	BOOL	J01_UnitErr		グローバル変数			
10		NetErr	ネットワーク異常発生中	R	BOOL	J01_NetErr		グローバル変数			
11		UnitMemErr	ユニットメモリ異常	R	BOOL	J01_UnitMemErr		グローバル変数			
12		LANHwErr	通信コントローラ異常	R	BOOL	J01_LANHwErr		グローバル変数			
13		IPAdrDupErr	IP アドレス重複異常	R	BOOL	J01_IPAdrDupErr		グローバル変数			
14		LkOffErr	リンクオフ異常	R	BOOL	J01_LkOffErr		グローバル変数			
15		UserStaAreaCfgErr	ステータスエリアレイアウト設定異常	R	BOOL	J01_UserStaAreaCfgErr		グローバル変数			
16		UnitSta2	ユニットステータス 2	R	WORD	J01_UnitSta2		グローバル変数			
17		ETNOnlineSta	オンライン	R	BOOL	J01_ETNOnlineSta		グローバル変数			
18		TDLOprSta	タグデータリンク動作中	R	BOOL	J01_TDLOprSta		グローバル変数			
19		IPAdrChgErr	運転中の IP アドレス変更	R	BOOL	J01_IPAdrChgErr		グローバル変数			
20		UserStaAreaEnblSta	ユーザ設定エリア有効	R	BOOL	J01_UserStaAreaEnblSta		グローバル変数			
21		MultiSwOnErr	複数スイッチ同時 ON 異常	R	BOOL	J01_MultiSwOnErr		グローバル変数			
22		ErrLogStoreSta	異常履歴あり	R	BOOL	J01_ErrLogStoreSta		グローバル変数			
23		CommSta1	通信ステータス 1	R	WORD	J01_CommSta1		グローバル変数			
24		TDLOpnErr	照合異常	R	BOOL	J01_TDLOpnErr		グローバル変数			
25		TDLErr	タグデータリンク異常	R	BOOL	J01_TDLErr		グローバル変数			
26		CommParamErr	通信パラメータ設定不正	R	BOOL	J01_CommParamErr		グローバル変数			
27		TagRefreshErr	タグリフレッシュ異常	R	BOOL	J01_TagRefreshErr		グローバル変数			
28		TagDbErr	タグデータベース異常	R	BOOL	J01_TagDbErr		グローバル変数			
29		TDLAllRunSta	全タグデータリンク通信中	R	BOOL	J01_TDLAllRunSta		グローバル変数			
30		TDLRunSta	タグデータリンク通信中	R	BOOL	J01_TDLRunSta		グローバル変数			

5 **表計算ソフトで AdjTmCmd と TargetPLCMdSta の間に空行を 1 行追加します。**

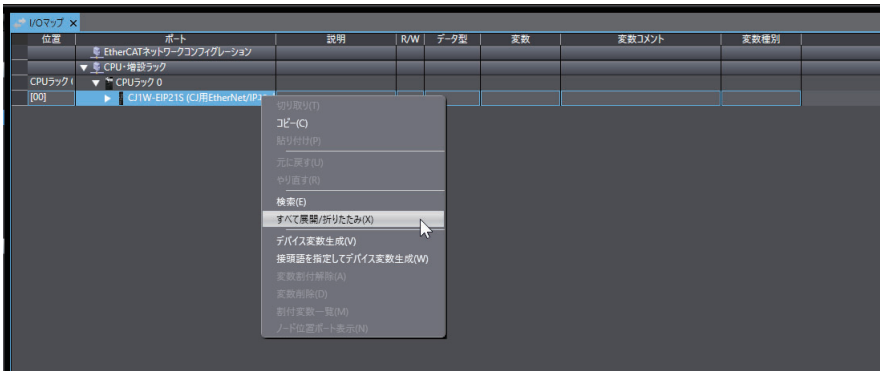
	A	B	C	D	E	F	G	H	I	J	K
1	位置	ポート	説明	R/W	データ型	変数	変数コメント	変数種別			
2		SoftSwCmd	ソフトスイッチ	RW	WORD	J01_SoftSwCmd		グローバル変数			
3		TDLStartCmd	タグデータリンク開始スイッチ	RW	BOOL	J01_TDLStartCmd		グローバル変数			
4		TDLStopCmd	タグデータリンク停止スイッチ	RW	BOOL	J01_TDLStopCmd		グローバル変数			
5		AdjTmCmd	時計情報自動調整実行スイッチ	RW	BOOL	J01_AdjTmCmd		グローバル変数			
6											
7		TargetPLCMdSta	ターゲットノード PLC 本体運転情報	R	ARRAY[0..63] OF BOOL	J01_TargetPLCMdSta		グローバル変数			
8		TargetPLCErrSta	ターゲットノード PLC 本体異常情報	R	ARRAY[0..63] OF BOOL	J01_TargetPLCErrSta		グローバル変数			
9		UnitSta1	ユニットステータス 1	R	WORD	J01_UnitSta1		グローバル変数			
10		UnitErr	ユニット異常発生中	R	BOOL	J01_UnitErr		グローバル変数			
11		NetErr	ネットワーク異常発生中	R	BOOL	J01_NetErr		グローバル変数			
12		UnitMemErr	ユニットメモリ異常	R	BOOL	J01_UnitMemErr		グローバル変数			
13		LANHwErr	通信コントローラ異常	R	BOOL	J01_LANHwErr		グローバル変数			
14		IPAdrDupErr	IP アドレス重複異常	R	BOOL	J01_IPAdrDupErr		グローバル変数			
15		LkOffErr	リンクオフ異常	R	BOOL	J01_LkOffErr		グローバル変数			

6 **表計算ソフトで変数、変数コメント、変数種別の列のデータをすべてコピーします。**

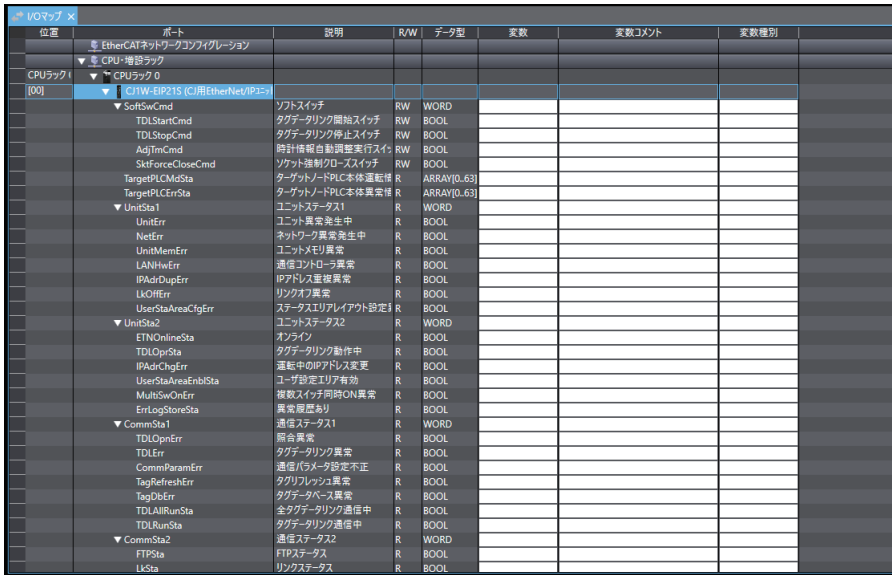
	A	B	C	D	E	F	G	H	I	J	K
1	位置	ポート	説明	R/W	データ型	変数	変数コメント	変数種別			
2		SoftSwCmd	ソフトスイッチ	RW	WORD	J01_SoftSwCmd		グローバル変数			
3		TDLStartCmd	タグデータリンク開始スイッチ	RW	BOOL	J01_TDLStartCmd		グローバル変数			
4		TDLStopCmd	タグデータリンク停止スイッチ	RW	BOOL	J01_TDLStopCmd		グローバル変数			
5		AdjTmCmd	時計情報自動調整実行スイッチ	RW	BOOL	J01_AdjTmCmd		グローバル変数			
6											
7		TargetPLCMdSta	ターゲットノード PLC 本体運転情報	R	ARRAY[0..63] OF BOOL	J01_TargetPLCMdSta		グローバル変数			
8		TargetPLCErrSta	ターゲットノード PLC 本体異常情報	R	ARRAY[0..63] OF BOOL	J01_TargetPLCErrSta		グローバル変数			
9		UnitSta1	ユニットステータス 1	R	WORD	J01_UnitSta1		グローバル変数			
10		UnitErr	ユニット異常発生中	R	BOOL	J01_UnitErr		グローバル変数			
11		NetErr	ネットワーク異常発生中	R	BOOL	J01_NetErr		グローバル変数			
12		UnitMemErr	ユニットメモリ異常	R	BOOL	J01_UnitMemErr		グローバル変数			
13		LANHwErr	通信コントローラ異常	R	BOOL	J01_LANHwErr		グローバル変数			
14		IPAdrDupErr	IP アドレス重複異常	R	BOOL	J01_IPAdrDupErr		グローバル変数			
15		LkOffErr	リンクオフ異常	R	BOOL	J01_LkOffErr		グローバル変数			

7 **Sysmac Studio で置き換え先のプロジェクトを開き、マルチビューエクスプローラで［構成・設定］–［I/O マップ］をダブルクリックします。**
エディットウィンドウに I/O マップが表示されます。

8 **I/O マップの置き換え先の形 CJ1W-EIP21S 上で右クリックして［すべて展開／折りたたみ］を選択します。**



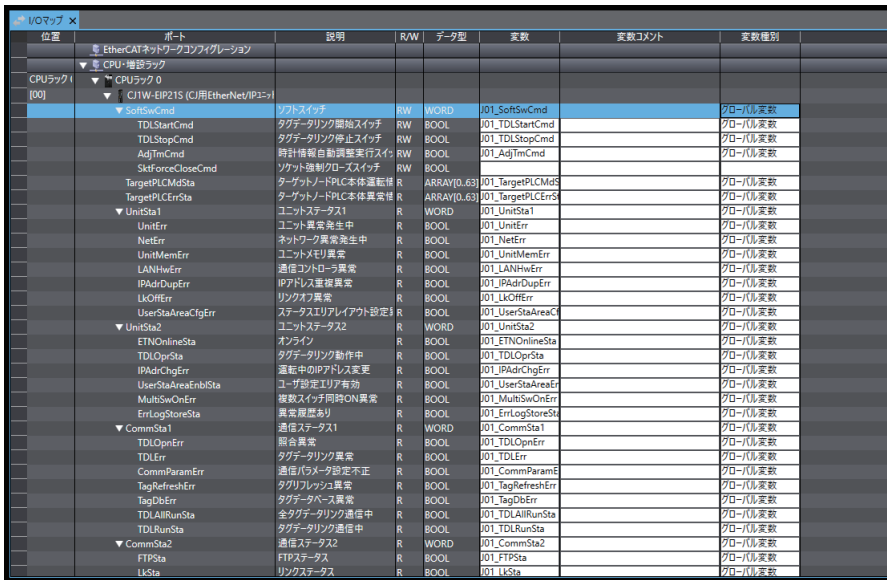
置き換え先の形 CJ1W-EIP21S の I/O ポートが展開されます。



9 置き換え先の形 CJ1W-EIP21S の I/O ポートの最初の行を選択して表計算ソフトからコピーしたデータを貼り付けます。



置き換え先の形 CJ1W-EIP21S の I/O ポートの変数、コメント、変数種別が設定されます。



ユーザプログラムの修正

置き換えによる仕様の差異や制約事項に対応するために、ユーザプログラムを修正します。置き換え元の機種との差異でユーザプログラムに影響する差異については、「A-8-1 仕様や機能などの差異」(P.A-101) を参照してください。
ユーザプログラムの修正後、コントローラのプロジェクトデータを保存します。

タグデータリンクの設定

置き換え後のユニット構成に合わせて、タグデータリンクの設定を行います。一から設定し直す必要はありません。Network Configurator を使用して、置き換え元のネットワーク構成ファイルに対し、デバイス変更の機能を使用して、置き換え元の形 CJ1W-EIP21 を形 CJ1W-EIP21S に変更します。

A-8-4　置き換えの実行

ここでは、実際に置き換えを行う内容を説明します。

ユニットの置き換えと配線

各ユニットの置き換えおよび配線を行います。各ユニットのハードウェア設定、設置方法、および配線については、置き換え先のユニットのユーザーズマニュアルを参照ください。
なお、置き換えが成功しなかったリスクを想定して、データのバックアップを実施してください。

● ユニットの置き換え

電源 OFF の状態で、各ユニットの置き換えを行います。
形 CJ1W-EIP21S のユニット番号とノードアドレス（デフォルト IP アドレス）のハードウェア設定を、置き換え元の形 CJ1W-EIP21 に合わせて設定します。
設定後、置き換えます。

形 CJ1W-EIP21S 以外に置き換えるユニットがある場合は、それらを置き換えます。

● ユニットの配線

置き換え後、配線を行います。
ただし形 CJ1W-EIP21S への Ethernet ケーブルはこの時点では外した状態にしておきます。

電源の投入

置き換え後のコントローラに電源を投入します。

使用上の注意

形 CJ1W-EIP21S はユニットの起動に時間がかかります。これにより、置き換え前のシステムと比べ、CPU ユニットの運転開始時間が数秒遅くなります。

各データの転送

置き換え後の実機に、事前に準備した以下のデータを CPU ユニットに直接接続したサポートソフトで転送します。

- コントローラのプロジェクトデータ
- タグデータリンクパラメータ

使用上の注意

形 CJ1W-EIP21S の EtherNet/IP ポートを経由した Sysmac Studio のオンライン接続はできません。

実機への転送フローを以下に示します。以下のフローに沿って、各データの転送を行います。

手順	項目	内容	参照先
1	コントローラのプロジェクトデータのダウンロード	置き換え後の各コントローラにプロジェクトデータを転送します。	「コントローラのプロジェクトデータのダウンロード」(P.A-115)
2	通信ケーブルの接続	各コントローラの電源を OFF にします。その後に置き換え先の形 CJ1W-EIP21S に通信ケーブルを接続します。	−
3	コントローラの電源 ON	装置の安全を確認して、各コントローラの電源を ON します。	−
4	タグデータリンクパラメータのダウンロード	タグデータリンクパラメータを EtherNet/IP ネットワークごとに一括でダウンロードします。	「タグデータリンクパラメータのダウンロード」(P.A-115)

● コントローラのプロジェクトデータのダウンロード

事前準備にて、置き換え後のユニット構成に合わせて作成したコントローラのプロジェクトデータを実機に転送します。Sysmac Studio を CPU ユニットに直接接続して、オンライン接続し、コントローラに転送します。

使用上の注意

EtherNet/IP ユニットのデータを転送するときは以下に注意してください。
メインメニューの［コントローラ］-［転送］を使用した転送では、デフォルトでは高機能ユニットのデータが転送されません。転送時のオプションで［以下を転送しない（すべて同期の対象外）］のチェックを外してから転送してください。
または［CPU・増設ラック］で編集画面で EtherNet/IP ユニットの設定画面を開き、［転送（パソコン→コントローラ）］ボタンを使用して転送してください。
転送方法の詳細については『Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)』を参照してください。

● タグデータリンクパラメータのダウンロード

置き換え元でタグデータリンクの機能を使用している場合が対象です。事前準備にて、置き換え後のユニット構成に合わせて作成したタグデータリンクの設定を、Network Configurator で実機にダウンロードします。

動作確認

各データの転送後、意図どおりにシステムが動作しているかを Sysmac Studio で確認します。また異常が発生してないかなどをユニットの異常の確認方法に従って、確認します。
各ユニットの動作確認方法やトラブルシューティングについては、使用するユニットのユーザーズマニュアルおよび『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

セキュリティ施策の実施

必要に応じて、置き換え後のシステム運用開始までに、外部からの攻撃によるシステムのセキュリティリスクの低減を行います。

セキュリティリスクを低減するためのセキュリティ機能を、形 CJ1W-EIP21S および Sysmac Studio に搭載しています。システム運用に合わせて、セキュリティ機能を使用して、セキュリティリスクの低減を行います。
詳細については、「第 12 章 セキュリティ機能」(P.12-1) を参照してください。

セキュリティ施策の実施後の動作確認

セキュリティ施策を実施した場合は、セキュリティ機能の設定に対し、動作確認を行います。

A-9 ソケットサービス機能のサンプルプログラム

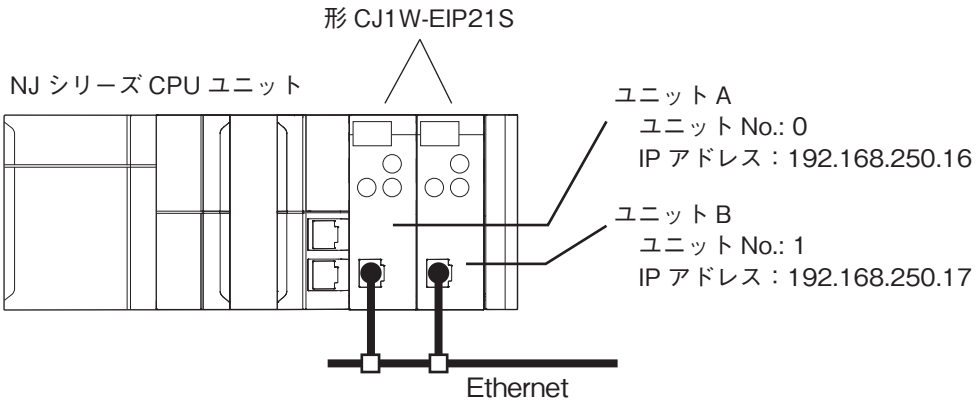
形 CJ1W-EIP21S では、ソケットサービスを利用できます。ここでは 4 つのケースについてのサンプルプログラムを記載します。

共通の内容については「A-9-1 システム構成」および「A-9-2 サンプルプログラムで必要な設定」を参照してください。

4 つのケースのそれぞれのサンプルプログラムについては以下を参照してください。
「A-9-3 デバイス変数操作による UDP 通信のサンプルプログラム」
「A-9-4 デバイス変数操作による TCP 通信のサンプルプログラム」
「A-9-5 SendCmd 命令による UDP 通信のサンプルプログラム」
「A-9-6 SendCmd 命令による TCP 通信のサンプルプログラム」

A-9-1 システム構成

プログラム例が前提とするシステム構成は以下のとおりです。



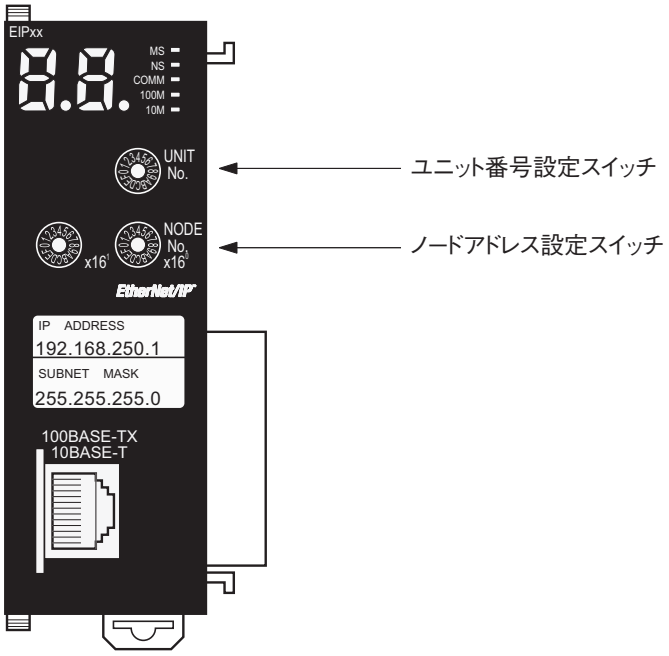
- 1 台の NJ シリーズ CPU ユニットに 2 台の形 CJ1W-EIP21S（以降、EtherNet/IP ユニットと記述します）を装着します。
- 2 台の EtherNet/IP ユニットを Ethernet ケーブルで接続します。
- ユニット A を UDP/TCP クライアント、ユニット B を UDP/TCP サーバとします。

A-9-2　サンプルプログラムで必要な設定

ここではサンプルプログラムを動作させるのに必要な設定内容を記載します。

ロータリスイッチの設定

ロータリスイッチを以下のとおりに設定します。

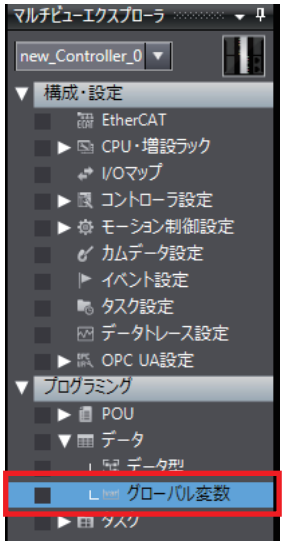


ロータリスイッチ	ユニットA	ユニットB
ユニット番号設定スイッチ	0	1
ノードアドレス設定スイッチ（×16^1）	1	1
ノードアドレス設定スイッチ（×16^0）	0	1

Sysmac Studio の設定

● グローバル変数の作成

［グローバル変数］画面でグローバル変数を作成します。



全サンプルプログラムで必要なグローバル変数を以下に示します。

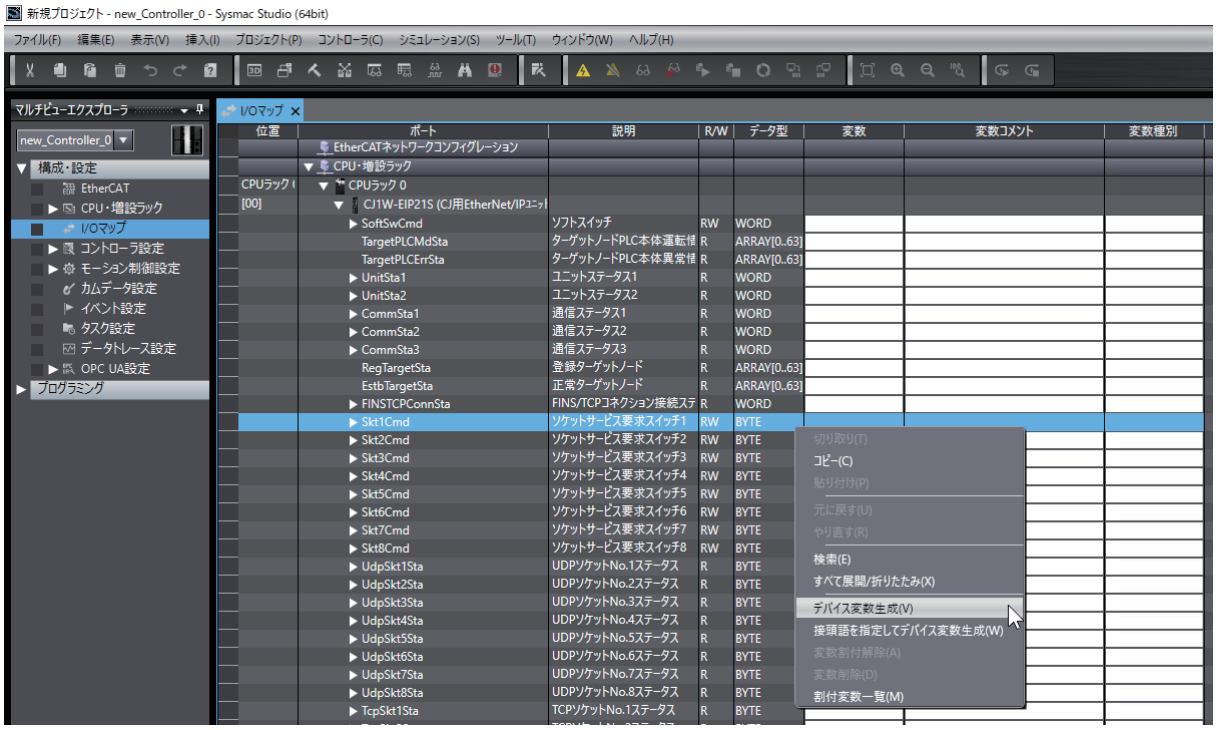
名称	データ型	割付先	保持	ネットワーク公開	コメント
J01_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J02_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用

なお、データ型 _sCJEIP21_User_StaArea を使用するには、このデータ型をあらかじめ登録しておく必要があります。［構成・設定］の［CPU・増設ラック］で形 CJ1W-EIP21S を選択した状態で右クリックして［ユニット用データ型登録］を選択して登録します。

各サンプルプログラムで必要なグローバル変数は、各サンプルプログラムの説明を参照してください。

● デバイス変数の作成

［I/O マップ］画面で該当するポートのデバイス変数を作成します。



必要なデバイス変数は、各サンプルプログラムの説明を参照してください。

● ユニットの設定

［CPU・増設ラック］の編集画面から、該当する EtherNet/IP ユニットの［高機能ユニット設定の編集］画面を開いて以下の項目を設定します。設定の詳細については「13-4 ソケットサービス機能に必要な設定」を参照してください。

設定項目	ユニット A	ユニット B
IP アドレス	192.168.250.16	192.168.250.17
サブネットマスク	255.255.255.0	255.255.255.0
レイアウトタイプ	ユーザ設定	ユーザ設定
割付エリア	J01_StaArea	J02_StaArea

A-9-3　デバイス変数操作による UDP 通信のサンプルプログラム

ユニット A からユニット B に UDP 通信を使用して 100 バイト（50 ワード）のデータを送信します。
このサンプルプログラムでは、UDP ソケット No.1 と UDP ポート No.4096 を使用します。

動作説明

(1)　ユニット B（UDP サーバ）のプログラムの内部変数「EnableServer」を TRUE にすると、ユニット B は UDP データの受信待ち状態となります。

(2)　ユニット A（UDP クライアント）のプログラムの内部変数「EnableClient」を TRUE にすると、ユニット A は UDP データの送信起動待ち状態となります。

(3)　ユニット A のプログラムの内部変数「SendTrigger」を TRUE にすると、ユニット A はユニット B に対して、J01_Skt1UdpSendDat に格納されたデータを 100 バイト（50 ワード）送信します。

(4)　ユニット B 側では、受信したデータが J02_Skt1UdpRcvDat に格納されます。

ユニット A のユーザプログラム

● グローバル変数

名称	データ型	割付先	保持	ネットワーク公開	コメント
J01_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J01_Skt1UdpSendDat	ARRAY[0..49] OF WORD	%D0	TRUE	非公開	送信データ格納用

● デバイス変数

ポート	名称	ネットワーク公開	コメント
Skt1UdpOpenCmd	J01_Skt1UdpOpenCmd	非公開	
Skt1SendCmd	J01_Skt1SendCmd	非公開	
Skt1CloseCmd	J01_Skt1CloseCmd	非公開	
Skt1UdpOpened	J01_Skt1UdpOpened	非公開	
Skt1No	J01_Skt1No	非公開	
Skt1LocalPortNo	J01_Skt1LocalPortNo	非公開	
Skt1RemoteIPAdr1	J01_Skt1RemoteIPAdr1	非公開	
Skt1RemoteIPAdr2	J01_Skt1RemoteIPAdr2	非公開	
Skt1RemoteIPAdr3	J01_Skt1RemoteIPAdr3	非公開	
Skt1RemoteIPAdr4	J01_Skt1RemoteIPAdr4	非公開	
Skt1RemotePortNo	J01_Skt1RemotePortNo	非公開	
Skt1SendRcvByte	J01_Skt1SendRcvByte	非公開	
Skt1SendRcvAdrType	J01_Skt1SendRcvAdrType	非公開	送信データ格納変数の割付先アドレスのエリア種別
Skt1SendRcvAdrCh1	J01_Skt1SendRcvAdrCh1	非公開	送信データ格納変数の割付先アドレスの上位 2 けた
Skt1SendRcvAdrCh2	J01_Skt1SendRcvAdrCh2	非公開	送信データ格納変数の割付先アドレスの下位 2 けた
Skt1RespCode	J01_Skt1RespCode	非公開	

● 内部変数

名称	データ型	コメント
EnableClient	BOOL	
ClientState	INT	
SendTrigger	BOOL	

● プログラムコード

```
IF (EnableClient=TRUE) THEN
    IF (ClientState=0) THEN
        ClientState:=1; // Next state is "Open".
    END_IF;
ELSE
    IF ((J01_Skt1UdpOpened=TRUE) AND (ClientState<5)) THEN
        ClientState:=5;  // Close opened socket.
    END_IF;
END_IF;

CASE ClientState OF
    0:  // Idle
        // Nothing to do.

    1: // Open
        J01_Skt1No:=USINT#1;
        J01_Skt1LocalPortNo:=UINT#4096;
        J01_Skt1UdpOpenCmd:=TRUE; // Open request
        ClientState:=2; // Next state is "Opening".

    2: // Opening
        IF (J01_Skt1UdpOpenCmd=FALSE) THEN // Open request completed
            IF (J01_Skt1RespCode=WORD#0000) THEN // Success
                ClientState:=3;  // Next state is "Opened".
            ELSE // Error
                EnableClient:=FALSE;
                ClientState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    3: // Opened
        IF (SendTrigger=TRUE) THEN
            SendTrigger:=FALSE;
            J01_Skt1No              :=USINT#1;
            J01_Skt1RemoteIPAdr1    :=USINT#192;
            J01_Skt1RemoteIPAdr2    :=USINT#168;
            J01_Skt1RemoteIPAdr3    :=USINT#250;
            J01_Skt1RemoteIPAdr4    :=USINT#17;
            J01_Skt1RemotePortNo    :=UINT#4096;
            J01_Skt1SendRcvByte     :=UINT#100;
            J01_Skt1SendRcvAdrType  :=BYTE#16#82;
            J01_Skt1SendRcvAdrCh1   :=BYTE#16#00;
            J01_Skt1SendRcvAdrCh2   :=BYTE#16#00;
            J01_Skt1SendCmd:=TRUE; // Send request
            ClientState:=4; // Next state is "Sending".
        END_IF;

    4: // Sending
        IF (J01_Skt1SendCmd=FALSE) THEN // Send request completed
            IF (J01_Skt1RespCode=WORD#0000) THEN // Success
                ClientState:=3; // Next state is "Opened".
            ELSE // Error
                ClientState:=5; // Next state is "Close".
            END_IF;
        END_IF;

    5: // Close
        IF (J01_Skt1UdpOpened=TRUE) THEN
```

```
            J01_Skt1No:=USINT#1;
            J01_Skt1CloseCmd:=TRUE; // Close request
            ClientState:=6; // Next state is "Closing".
        ELSE
            EnableClient:=FALSE;
            ClientState:=0; // Next state is "Idle".
        END_IF;

    6: // Closing
        IF (J01_Skt1CloseCmd=FALSE) THEN // Close request completed
            EnableClient:=FALSE;
            ClientState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

ユニットBのユーザプログラム

● グローバル変数

名称	データ型	割付先	保持	ネットワーク公開	コメント
J02_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J02_Skt1UdpRcvDat	ARRAY[0..49] OF WORD	%D1000	TRUE	非公開	受信データ格納用

● デバイス変数

ポート	名称	ネットワーク公開	コメント
Skt1UdpOpenCmd	J02_Skt1UdpOpenCmd	非公開	
Skt1RcvCmd	J02_Skt1RcvCmd	非公開	
Skt1CloseCmd	J02_Skt1CloseCmd	非公開	
Skt1UdpOpened	J02_Skt1UdpOpened	非公開	
Skt1No	J02_Skt1No	非公開	
Skt1LocalPortNo	J02_Skt1LocalPortNo	非公開	
Skt1SendRcvByte	J02_Skt1SendRcvByte	非公開	
Skt1SendRcvAdrType	J02_Skt1SendRcvAdrType	非公開	受信データ格納変数の割付先アドレスのエリア種別
Skt1SendRcvAdrCh1	J02_Skt1SendRcvAdrCh1	非公開	受信データ格納変数の割付先アドレスの上位2けた
Skt1SendRcvAdrCh2	J02_Skt1SendRcvAdrCh2	非公開	受信データ格納変数の割付先アドレスの下位2けた
Skt1Timeout	J02_Skt1Timeout	非公開	
Skt1RespCode	J02_Skt1RespCode	非公開	

● 内部変数

名称	データ型	コメント
EnableServer	BOOL	
ServerState	INT	

● プログラムコード

```
IF (EnableServer=TRUE) THEN
    IF (ServerState=0) THEN
        ServerState:=1; // Next state is "Open".
    END_IF;
ELSE
    IF ((J02_Skt1UdpOpened=TRUE) AND (ServerState<5)) THEN
        ServerState:=5;  // Close opened socket.
    END_IF;
```

```
END_IF;

CASE ServerState OF
    0: // Idle
        // Nothing to do.

    1: // Open
        J02_Skt1No          :=USINT#1;
        J02_Skt1LocalPortNo :=UINT#4096;
        J02_Skt1UdpOpenCmd  :=TRUE; // Open request
        ServerState:=2; // Next state is "Opening".

    2: // Opening
        IF (J02_Skt1UdpOpenCmd=FALSE) THEN // Open request completed
            IF (J02_Skt1RespCode=WORD#0000) THEN // Success
                ServerState:=3;  // Next state is "Opened".
            ELSE    // Error
                EnableServer:=FALSE;
                ServerState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    3: // Opened
        J02_Skt1No              :=USINT#1;
        J02_Skt1SendRcvByte     :=UINT#100;
        J02_Skt1SendRcvAdrType  :=BYTE#16#82;
        J02_Skt1SendRcvAdrCh1   :=BYTE#16#03;
        J02_Skt1SendRcvAdrCh2   :=BYTE#16#E8;
        J02_Skt1Timeout         :=0;
        J02_Skt1RcvCmd          :=TRUE; // Receive request

        ServerState:=4; // Next state is "Receiving".

    4: // Receiving
        IF (J02_Skt1RcvCmd=FALSE) THEN // Receive request completed
            IF (J02_Skt1RespCode=WORD#0000) THEN // Success
                ServerState:=3; // Next state is "Opened".
            ELSE // Error
                ServerState:=5; // Next state is "Close".
            END_IF;
        END_IF;

    5: // Close
        IF (J02_Skt1UdpOpened=TRUE) THEN
            J02_Skt1No      :=USINT#1;
            J02_Skt1CloseCmd:=TRUE; // Close request
            ServerState:=6; // Next state is "Closing".
        ELSE
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle".
        END_IF;

    6: // Closing
        IF (J02_Skt1CloseCmd=FALSE) THEN // Close request completed
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

A-9-4　デバイス変数操作による TCP 通信のサンプルプログラム

ユニット A からユニット B に TCP 通信を使用して 100 バイト（50 ワード）のデータを送信します。
このサンプルプログラムでは、TCP ソケット No.2 と TCP ポート No.4097 を使用します。

動作説明

(1) ユニット B（TCP サーバ）のプログラムの内部変数「EnableServer」を TRUE にすると、ユニット B は TCP のコネクション待ち状態となります。

(2) ユニット A（TCP クライアント）のプログラムの内部変数「EnableClient」を TRUE にすると、ユニット A はユニット B との間の TCP コネクションを開設、確立します。

(3) コネクション開設後、ユニット B は TCP データの受信待ち状態となります。ユニット A は TCP データの送信起動待ち状態となります。

(4) ユニット A のプログラムの内部変数「SendTrigger」を TRUE にすると、ユニット A はユニット B に対して、J01_Skt2TcpSendDat に格納されたデータを 100 バイト（50 ワード）送信します。

(5) ユニット B 側では、受信したデータが J02_Skt2TcpRcvDat に格納されます。

ユニット A のユーザプログラム

● グローバル変数

名称	データ型	割付先	保持	ネットワーク公開	コメント
J01_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J01_Skt2TcpSendDat	ARRAY[0..49] OF WORD	%D100	TRUE	非公開	送信データ格納用

● デバイス変数

ポート	名称	ネットワーク公開	コメント
Skt2TcpActiveCmd	J01_Skt2TcpActiveCmd	非公開	
Skt2SendCmd	J01_Skt2SendCmd	非公開	
Skt2CloseCmd	J01_Skt2CloseCmd	非公開	
Skt2TcpOpened	J01_Skt2TcpOpened	非公開	
Skt2No	J01_Skt2No	非公開	
Skt2RemoteIPAdr1	J01_Skt2RemoteIPAdr1	非公開	
Skt2RemoteIPAdr2	J01_Skt2RemoteIPAdr2	非公開	
Skt2RemoteIPAdr3	J01_Skt2RemoteIPAdr3	非公開	
Skt2RemoteIPAdr4	J01_Skt2RemoteIPAdr4	非公開	
Skt2RemotePortNo	J01_Skt2RemotePortNo	非公開	
Skt2SendRcvByte	J01_Skt2SendRcvByte	非公開	
Skt2SendRcvAdrType	J01_Skt2SendRcvAdrType	非公開	送信データ格納変数の割付先アドレスのエリア種別
Skt2SendRcvAdrCh1	J01_Skt2SendRcvAdrCh1	非公開	送信データ格納変数の割付先アドレスの上位 2 けた
Skt2SendRcvAdrCh2	J01_Skt2SendRcvAdrCh2	非公開	送信データ格納変数の割付先アドレスの下位 2 けた
Skt2Timeout	J01_Skt2Timeout	非公開	
Skt2RespCode	J01_Skt2RespCode	非公開	

● 内部変数

名称	データ型	コメント
EnableClient	BOOL	

名称	データ型	コメント
ClientState	INT	
SendTrigger	BOOL	

● プログラムコード

```
IF (EnableClient=TRUE) THEN
    IF (ClientState=0) THEN
        ClientState:=1; // Next state is "Open".
    END_IF;
ELSE
    IF ((J01_Skt2TcpOpened=TRUE) AND (ClientState<5)) THEN
            ClientState:=5;  // Close opened socket.
    END_IF;
END_IF;

CASE ClientState OF
    0:  // Idle
        // Nothing to do.

    1: // Open
            J01_Skt2No              :=USINT#2;
            J01_Skt2RemoteIPAdr1    :=USINT#192;
            J01_Skt2RemoteIPAdr2    :=USINT#168;
            J01_Skt2RemoteIPAdr3    :=USINT#250;
            J01_Skt2RemoteIPAdr4    :=USINT#17;
            J01_Skt2RemotePortNo    :=UINT#4097;
            J01_Skt2Timeout         :=0;
            J01_Skt2TcpActiveCmd    :=TRUE; // Active Open request
            ClientState:=2; // Next state is "Opening".

    2: // Opening
        IF (J01_Skt2TcpActiveCmd=FALSE) THEN // Active Open request completed
            IF (J01_Skt2RespCode=WORD#0000) THEN // Success
                ClientState:=3;  // Next state is "Opened".
            ELSE // Error
                EnableClient:=FALSE;
                ClientState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    3: // Opened
        IF (SendTrigger=TRUE) THEN
            SendTrigger:=FALSE;
            J01_Skt2No              :=USINT#2;
            J01_Skt2SendRcvByte     :=UINT#100;
            J01_Skt2SendRcvAdrType  :=BYTE#16#82;
            J01_Skt2SendRcvAdrCh1   :=BYTE#16#00;
            J01_Skt2SendRcvAdrCh2   :=BYTE#16#64;
            J01_Skt2SendCmd         :=TRUE; // Send request

            ClientState:=4; // Next state is "Sending".
        END_IF;

    4: // Sending
        IF (J01_Skt2SendCmd=FALSE) THEN // Send request completed
            IF (J01_Skt2RespCode=WORD#0000) THEN // Success
                ClientState:=3; // Next state is "Opened".
            ELSE // Error
                ClientState:=5; // Next state is "Close".
```

```
            END_IF;
        END_IF;

    5: // Close
        IF (J01_Skt2TcpOpened=TRUE) THEN
            J01_Skt2No      :=USINT#2;
            J01_Skt2CloseCmd:=TRUE; // Close request
            ClientState:=6; // Next state is "Closing".
        ELSE
            EnableClient:=FALSE;
            ClientState:=0; // Next state is "Idle".
        END_IF;

    6: // Closing
        IF (J01_Skt2CloseCmd=FALSE) THEN // Close request completed
            EnableClient:=FALSE;
            ClientState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

ユニットBのユーザプログラム

● グローバル変数

名称	データ型	割付先	保持	ネットワーク公開	コメント
J02_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J02_Skt2TcpRcvDat	ARRAY[0..49] OF WORD	%D1100	TRUE	非公開	受信データ格納用

● デバイス変数

ポート	名称	ネットワーク公開	コメント
Skt2TcpPassiveCmd	J02_Skt2TcpPassiveCmd	非公開	
Skt2RcvCmd	J02_Skt2RcvCmd	非公開	
Skt2CloseCmd	J02_Skt2CloseCmd	非公開	
Skt2TcpDatRcved	J02_Skt2TcpDatRcved	非公開	
Skt2TcpOpened	J02_Skt2TcpOpened	非公開	
Skt2No	J02_Skt2No	非公開	
Skt2LocalPortNo	J02_Skt2LocalPortNo	非公開	
Skt2TcpRcvDatByte	J02_Skt2TcpRcvDatByte	非公開	
Skt2SendRcvByte	J02_Skt2SendRcvByte	非公開	
Skt2SendRcvAdrType	J02_Skt2SendRcvAdrType	非公開	受信データ格納変数の割付先アドレスのエリア種別
Skt2SendRcvAdrCh1	J02_Skt2SendRcvAdrCh1	非公開	受信データ格納変数の割付先アドレスの上位2けた
Skt2SendRcvAdrCh2	J02_Skt2SendRcvAdrCh2	非公開	受信データ格納変数の割付先アドレスの下位2けた
Skt2Timeout	J02_Skt2Timeout	非公開	
Skt2RespCode	J02_Skt2RespCode	非公開	

● 内部変数

名称	データ型	コメント
EnableServer	BOOL	
ServerState	INT	

● プログラムコード

```
IF (EnableServer=TRUE) THEN
```

```
    IF (ServerState=0) THEN
        ServerState:=1; // Next state is "Open".
    END_IF;
ELSE
    IF ((J02_Skt2TcpOpened=TRUE) AND (ServerState<5)) THEN
            ServerState:=5;  // Close opened socket.
    END_IF;

END_IF;


CASE ServerState OF
    0:  // Idle
        // Nothing to do.

    1: // Open
            J02_Skt2No              :=USINT#2;
            J02_Skt2LocalPortNo     :=UINT#4097;
            J02_Skt2Timeout         :=0;
            J02_Skt2TcpPassiveCmd   :=TRUE; // Passive Open request
            ServerState:=2; // Next state is "Opening".

    2: // Opening
        IF (J02_Skt2TcpPassiveCmd=FALSE) THEN // Passive Open request completed
            IF (J02_Skt2RespCode=WORD#0000) THEN // Success
                ServerState:=3;  // Next state is "Opened".
            ELSE    // Error
                EnableServer:=FALSE;
                ServerState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    3: // Opened
        IF ((J02_Skt2TcpDatRcved=TRUE) AND (J02_Skt2TcpRcvDatByte>0)) THEN
            J02_Skt2No              :=USINT#2;
            J02_Skt2SendRcvByte     :=UINT#100;
            J02_Skt2SendRcvAdrType  :=BYTE#16#82;
            J02_Skt2SendRcvAdrCh1   :=BYTE#16#04;
            J02_Skt2SendRcvAdrCh2   :=BYTE#16#4C;
            J02_Skt2Timeout         :=0;
            J02_Skt2RcvCmd          :=TRUE; // Receive request

            ServerState:=4; // Next state is "Receiving".
        END_IF;

    4: // Receiving
        IF (J02_Skt2RcvCmd=FALSE) THEN // Receive request completed
            IF (J02_Skt2RespCode=WORD#0000) THEN // Success
                ServerState:=3; // Next state is "Opened".
            ELSE // Error
                ServerState:=5; // Next state is "Close".
            END_IF;
        END_IF;

    5: // Close
        IF (J02_Skt2TcpOpened=TRUE) THEN
            J02_Skt2No      :=USINT#2;
            J02_Skt2CloseCmd:=TRUE; // Close request
            ServerState:=6; // Next state is "Closing".
```

```
        ELSE
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle".
        END_IF;

    6: // Closing
        IF (J02_Skt2CloseCmd=FALSE) THEN // Close request completed
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

A-9-5　SendCmd 命令による UDP 通信のサンプルプログラム

ユニット A からユニット B に UDP 通信を使用して 100 バイト（50 ワード）のデータを送信します。
このサンプルプログラムでは、UDP ソケット No.3 と UDP ポート No.4098 を使用します。

動作説明

(1) ユニット B（UDP サーバ）のプログラムの内部変数「EnableServer」を TRUE にすると、ユニット B は UDP データの受信待ち状態となります。

(2) ユニット A（UDP クライアント）のプログラムの内部変数「EnableClient」を TRUE にすると、ユニット A は UDP データの送信起動待ち状態となります。

(3) ユニット A のプログラムの内部変数「SendTrigger」を TRUE にすると、ユニット A はユニット B に対して、プログラムの内部変数「SendDat」に格納されたデータを 100 バイト（50 ワード）送信します。

(4) ユニット B 側では、受信したデータがプログラムの内部変数「RcvDat」に格納されます。

ユニット A のユーザプログラム

● グローバル変数

名称	データ型	割付先	保持	ネットワーク公開	コメント
J01_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J01_Skt3UdpOpenResult	WORD	%D2200	TRUE	非公開	ソケットオープン要求結果格納用
J01_Skt3UdpSendResult	ARRAY[0..1] OF WORD	%D2220	TRUE	非公開	送信要求結果格納用
J01_Skt3UdpCloseResult	WORD	%D2210	TRUE	非公開	ソケットクローズ要求結果格納用

● デバイス変数

ポート	名称	ネットワーク公開	コメント
Skt3UdpOpening	J01_Skt3UdpOpening	非公開	
Skt3UdpSending	J01_Skt3UdpSending	非公開	
Skt3UdpClosing	J01_Skt3UdpClosing	非公開	
Skt3UdpOpened	J01_Skt3UdpOpened	非公開	

● 内部変数

名称	データ型	コメント
EnableClient	BOOL	
ClientState	INT	
SendTrigger	BOOL	
SendCmd_instance	SendCmd	
CmdDst	_sDNET_ADR	
CmdReq	ARRAY[0..255] OF BYTE	
CmdResp	ARRAY[0..3] OF BYTE	
CmdOption	_sRESPONSE	
PortUsing	BOOL	
SendDat	ARRAY[0..49] OF WORD	送信データ格納用

● プログラムコード

```
// State
//  0: Idle
```

```
// 10: Prepare Open,        11: Open,       12: Opening,        13: Opened
// 20: Prepare Send,        21: Send,       22: Sending
// 40: Prepare Close,       41: Close,      42: Closing

IF (EnableClient=TRUE) THEN
    IF (ClientState=0) THEN
        ClientState:=10; // Next state is "Prepare Open".
    END_IF;
ELSE
    IF (J01_Skt3UdpOpened=TRUE) AND (ClientState<40) THEN
        ClientState:=40; // Close opened socket. Next state is "Prepare Close".
    END_IF;
END_IF;

CASE ClientState OF
    0:  // Idle
        // Nothing to do.

    10: // Prepare Open
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0]:=BYTE#16#27; // Command
        CmdReq[1]:=BYTE#16#01; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#03; // Socket number(=3)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D2200/J01_Skt3UpdOpenResult)
        CmdReq[5]:=BYTE#16#08; //  same as above
        CmdReq[6]:=BYTE#16#98; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above
        CmdReq[8]:=BYTE#16#10; // Local port number(=4098)
        CmdReq[9]:=BYTE#16#02; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=11; // Next state is "Open".

    11: // Open
        IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#10,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF (SendCmd_instance.Done=TRUE) THEN
                ClientState:=12; // Next state is "Opening".
            ELSIF (SendCmd_instance.Error=TRUE) THEN
                EnableClient:=FALSE;
                ClientState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    12: // Opening
        IF (J01_Skt3UdpOpening=FALSE) THEN
```

```
        IF (J01_Skt3UdpAreaErr=FALSE) AND (J01_Skt3UdpOpenResult=WORD#16#0000) THEN
           // Successfully opened
           ClientState:=13; // Next state is "Opened".
        ELSE // Fail to open
           EnableClient:=FALSE;
           ClientState:=0;  // Next state is "Idle".
        END_IF;
    END_IF;

13: // Opened
    IF (SendTrigger=TRUE) THEN
        SendTrigger:=FALSE;
        ClientState:=20; // Next state is "Prepare Send"
    END_IF;

20: // Prepare Send
    CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
    CmdReq[0] :=BYTE#16#27; // Command
    CmdReq[1] :=BYTE#16#03; //  same as above
    CmdReq[2] :=BYTE#16#00; // Socket option
    CmdReq[3] :=BYTE#16#03; // Socket number(=3)
    CmdReq[4] :=BYTE#16#82; // Result storage area(=D2220/J01_Skt3UdpSendResult)
    CmdReq[5] :=BYTE#16#08; //  same as above
    CmdReq[6] :=BYTE#16#AC; //  same as above
    CmdReq[7] :=BYTE#16#00; //  same as above
    CmdReq[8] :=BYTE#16#C0; // Remote IP address(=192.168.250.17)
    CmdReq[9] :=BYTE#16#A8; //  same as above
    CmdReq[10]:=BYTE#16#FA; //  same as above
    CmdReq[11]:=BYTE#16#11; //  same as above
    CmdReq[12]:=BYTE#16#10; // Remote port number(=4098)
    CmdReq[13]:=BYTE#16#02; //  same as above
    CmdReq[14]:=BYTE#16#00; // Number of bytes sent(=100)
    CmdReq[15]:=BYTE#16#64; //  same as above
    // CmdReqDat[16..115] is data.
   // SendDat(WORD Type) and CmdReq(BYTE Type) have different endiannesses, so swap each bytes.
    ToAryByte(SendDat, _HIGH_LOW ,CmdReq[16]);

    SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
    PortUsing:=FALSE;
    ClientState:=21; // Next state is "Send".

21: // Send
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#116,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF (SendCmd_instance.Done=TRUE) THEN
                ClientState:=22; // Next state is "Sending".
            ELSIF (SendCmd_instance.Error=TRUE) THEN
                ClientState:=40; // Next state is "Prepare Close".
```

```
            END_IF;
        END_IF;

22: // Sending
    IF (J01_Skt3UdpSending=FALSE) THEN
        IF (J01_Skt3UdpAreaErr=FALSE) AND (J01_Skt3UdpSendResult[0]=WORD#16#0000) THEN
            // Successfully sent.
            ClientState:=13; // Next state is "Opened".
        ELSE // Fail to send.
            ClientState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

40: // Prepare Close
    IF (J01_Skt3UdpOpened=TRUE) THEN
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0]:=BYTE#16#27; // Command
        CmdReq[1]:=BYTE#16#04; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#03; // Socket number(=3)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D2210/J01_Skt3UdpCloseResult)
        CmdReq[5]:=BYTE#16#08; //  same as above
        CmdReq[6]:=BYTE#16#A2; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=41; // Next state is "Close".
    ELSE
        EnableClient:=FALSE;
        ClientState:=0;  // Next state is "Idle".
    END_IF;

41: // Close
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#8,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

            IF (SendCmd_instance.Done=TRUE) THEN
                ClientState:=42; // Next state is "Closing".
            ELSIF (SendCmd_instance.Error=TRUE) THEN
                EnableClient:=FALSE;
                ClientState:=0;  // Next state is "Idle".
            END_IF;
    END_IF;

42:  // Closing
    IF (J01_Skt3UdpClosing=FALSE) THEN
        EnableClient:=FALSE;
        ClientState:=0; // Next state is "Idle".
```

```
        END_IF;
END_CASE;
```

ユニット B のユーザプログラム

● グローバル変数

名称	データ型	割付先	保持	ネットワーク公開	コメント
J02_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J02_Skt3UdpOpenResult	WORD	%D3200	TRUE	非公開	ソケットオープン要求結果格納用
J02_Skt3UdpRcvResult	ARRAY[0..54] OF WORD	%D3230	TRUE	非公開	受信要求結果および受信データ格納用
J02_Skt3UdpCloseResult	WORD	%D3210	TRUE	非公開	ソケットクローズ要求結果格納用

● デバイス変数

ポート	名称	ネットワーク公開	コメント
Skt3UdpOpening	J02_Skt3UdpOpening	非公開	
Skt3UdpRcving	J02_Skt3UdpRcving	非公開	
Skt3UdpClosing	J02_Skt3UdpClosing	非公開	
Skt3UdpOpened	J02_Skt3UdpOpened	非公開	

● 内部変数

名称	データ型	コメント
EnableServer	BOOL	
ServerState	INT	
SendCmd_instance	SendCmd	
CmdDst	_sDNET_ADR	
CmdReq	ARRAY[0..16] OF BYTE	
CmdResp	ARRAY[0..3] OF BYTE	
CmdOption	_sRESPONSE	
PortUsing	BOOL	
RcvDat	ARRAY[0..49] OF WORD	受信データ格納用

● プログラムコード

```
// State
//  0: Idle
// 10: Prepare Open,        11: Open,       12: Opening,        13: Opened
// 30: Prepare Receive,     31: Receive,    32: Receiving
// 40: Prepare Close,       41: Close,      42: Closing

IF (EnableServer=TRUE) THEN
    IF (ServerState=0) THEN
        ServerState:=10; // Next state is "Prepare Open".
    END_IF;
ELSE
    IF ((J02_Skt3UdpOpened=TRUE) AND (ServerState<40)) THEN
        ServerState:=40; // Close opened socket. Next state is "Prepare Close".
    END_IF;
END_IF;

CASE ServerState OF
    0:  // Idle
        // Nothing to do.
```

```
10: // Prepare Open
    CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
    CmdReq[0]:=BYTE#16#27; // Command
    CmdReq[1]:=BYTE#16#01; //  same as above
    CmdReq[2]:=BYTE#16#00; // Socket option
    CmdReq[3]:=BYTE#16#03; // Socket number(=3)
    CmdReq[4]:=BYTE#16#82; // Result storage area(=D3200/J02_Skt3UpdOpenResult)
    CmdReq[5]:=BYTE#16#0C; //  same as above
    CmdReq[6]:=BYTE#16#80; //  same as above
    CmdReq[7]:=BYTE#16#00; //  same as above
    CmdReq[8]:=BYTE#16#10; //  Local port number(=4098)
    CmdReq[9]:=BYTE#16#02; //  same as above

    SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
    PortUsing:=FALSE;
    ServerState:=11; // Next state is "Open".

11: // Open
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#10,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF (SendCmd_instance.Done=TRUE) THEN
            ServerState:=12; // Next state is "Opening".
        ELSIF (SendCmd_instance.Error=TRUE) THEN
            ServerState:=0;  // Next state is "Idle".
        END_IF;
    END_IF;

12: // Opening
    IF (J02_Skt3UdpOpening=FALSE) THEN
        IF (J02_Skt3UdpAreaErr=FALSE) AND (J02_Skt3UdpOpenResult=WORD#16#0000) THEN
            // Successfully opened
            ServerState:=13; // Next state is "Opened".
        ELSE // Fail to open
            ServerState:=0;  // Next state is "Idle".
        END_IF;

    END_IF;

13: // Opened
     ServerState:=30; // Next state is "Prepare Receive".

30: // Prepare Receive
    CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
    CmdReq[0] :=BYTE#16#27; // Command
    CmdReq[1] :=BYTE#16#02; //  same as above
    CmdReq[2] :=BYTE#16#00; // Socket option
    CmdReq[3] :=BYTE#16#03; // Socket number(=3)
```

```
    CmdReq[4] :=BYTE#16#82; // Result storage area(=D3230/J02_SKt3UdpRcvResult)
    CmdReq[5] :=BYTE#16#0C; //  same as above
    CmdReq[6] :=BYTE#16#9E; //  same as above
    CmdReq[7] :=BYTE#16#00; //  same as above
    CmdReq[8] :=BYTE#16#00; // Number of reception bytes(=100)
    CmdReq[9] :=BYTE#16#64; //  same as above
    CmdReq[10]:=BYTE#16#00; // Timeout value(=0:unlimited)
    CmdReq[11]:=BYTE#16#00; //  same as above

    SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
    PortUsing:=FALSE;
    ServerState:=31; // Next state is "Receive".

31: // Receive
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#12,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF (SendCmd_instance.Done=TRUE) THEN
            ServerState:=32; // Next state is "Receiving".
        ELSIF (SendCmd_instance.Error=TRUE) THEN
            ServerState:=40; // Next state is "Prepare Close".
        END_IF;
    END_IF;

32: // Receiving
    IF (J02_Skt3UdpRcving=FALSE) THEN
        IF (J02_Skt3UdpAreaErr=FALSE) AND (J02_Skt3UdpRcvResult[0]=WORD#16#0000) THEN
            // Successfully received. You can process received data here.
            AryMove(J02_Skt3UdpRcvResult[5], RcvDat[0], UINT#50);
            ServerState:=13; // Next state is "Opened".
        ELSE // Fail to receive.
            ServerState:=40; // Next state is "Prepare Close".
        END_IF;

    END_IF;

40: // Prepare Close
    IF (J02_Skt3UdpOpened=TRUE) THEN
        CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
        CmdReq[0]:=BYTE#16#27; // Command
        CmdReq[1]:=BYTE#16#04; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#03; // Socket number(=3)
        CmdReq[4]:=BYTE#16#82; // Result storage area( =D3210/J02_Skt3UdpCloseResult)
        CmdReq[5]:=BYTE#16#0C; //  same as above
        CmdReq[6]:=BYTE#16#8A; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
```

```
            PortUsing:=FALSE;
            ServerState:=41; // Next state is "Close".
        ELSE
            ServerState:=0;  // Next state is "Idle".
        END_IF;

    41: // Close
        IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#8,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

                IF (SendCmd_instance.Done=TRUE) THEN
                    ServerState:=42; // Next state is "Closing".
                ELSIF (SendCmd_instance.Error=TRUE) THEN
                    ServerState:=0;  // Next state is "Idle".
                END_IF;
        END_IF;

    42:  // Closing
        IF (J02_Skt3UdpClosing=FALSE) THEN
             ServerState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

A-9-6　SendCmd 命令による TCP 通信のサンプルプログラム

ユニット A からユニット B に TCP 通信を使用して 100 バイト（50 ワード）のデータを送信します。
このサンプルプログラムでは、TCP ソケット No.4 と TCP ポート No.4099 を使用します。

動作説明

(1) ユニット B（TCP サーバ）のプログラムの内部変数「EnableServer」を TRUE にすると、ユニット B は TCP のコネクション待ち状態となります。

(2) ユニット A（TCP クライアント）のプログラムの内部変数「EnableClient」を TRUE にすると、ユニット A はユニット B との間の TCP コネクションを開設、確立します。

(3) コネクション開設後、ユニット B は TCP データの受信待ち状態となります。ユニット A は TCP データの送信起動待ち状態となります。

(4) ユニット A のプログラムの内部変数「SendTrigger」を TRUE にすると、ユニット A はユニット B に対して、プログラムの内部変数「SendDat」に格納されたデータを 100 バイト（50 ワード）送信します。

(5) ユニット B 側では、受信したデータがプログラムの内部変数「RcvDat」に格納されます。

ユニット A のユーザプログラム

● グローバル変数

名称	データ型	割付先	保持	ネットワーク公開	コメント
J01_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J01_Skt4TcpActOpenResult	ARRAY[0..1] OF WORD	%D2300	TRUE	非公開	ソケットオープン要求結果格納用
J01_Skt4TcpSendResult	ARRAY[0..1] OF WORD	%D2320	TRUE	非公開	送信要求結果格納用
J01_Skt4TcpCloseResult	WORD	%D2310	TRUE	非公開	ソケットクローズ要求結果格納用

● デバイス変数

ポート	名称	ネットワーク公開	コメント
Skt4TcpOpening	J01_Skt4TcpOpening	非公開	
Skt4TcpSending	J01_Skt4TcpSending	非公開	
Skt4TcpClosing	J01_Skt4TcpClosing	非公開	
Skt4TcpOpened	J01_Skt4TcpOpened	非公開	

● 内部変数

名称	データ型	コメント
EnableClient	BOOL	
ClientState	INT	
CmdDst	_sDNET_ADR	
CmdReq	ARRAY[0..255] OF BYTE	
CmdResp	ARRAY[0..3] OF BYTE	
SendCmd_instance	SendCmd	
PortUsing	BOOL	
CmdOption	_sRESPONSE	
SendTrigger	BOOL	
SendDat	ARRAY[0..49] OF WORD	

● プログラムコード

```
// State
//  0: Idle
// 10: Prepare Open,         11: Open,        12: Opening,         13: Opened
// 20: Prepare Send,         21: Send,        22: Sending
// 40: Prepare Close,        41: Close,       42: Closing

IF (EnableClient=TRUE) THEN
    IF (ClientState=0) THEN
        ClientState:=10; // Next state is "Prepare Open".
    END_IF;
ELSE
    IF (J01_Skt4TcpOpened=TRUE) AND (ClientState < 40) THEN
        ClientState:=40; // Close opened socket. Next state is "Prepare Close".
    END_IF;
END_IF;

CASE ClientState OF
    0: //Idle
        //Nothing to do

    10: //Prepare Open
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0] :=BYTE#16#27; // Command(Tcp Socket Open Request)
        CmdReq[1] :=BYTE#16#11; //  same as above
        CmdReq[2] :=BYTE#16#00; // Socket option
        CmdReq[3] :=BYTE#16#04; // Socket number(=4)
        CmdReq[4] :=BYTE#16#82; // Result storage area(=D2300/J01_Skt4TcpActOpenResult)
        CmdReq[5] :=BYTE#16#08; //  same as above
        CmdReq[6] :=BYTE#16#FC; //  same as above
        CmdReq[7] :=BYTE#16#00; //  same as above
        CmdReq[8] :=BYTE#16#10; //  Local port number(=4099)
        CmdReq[9] :=BYTE#16#03; //  same as above
        CmdReq[10]:=BYTE#16#C0; // Remote IP Addr(=192.168.250.17)
        CmdReq[11]:=BYTE#16#A8; // same as above
        CmdReq[12]:=BYTE#16#FA; // same as above
        CmdReq[13]:=BYTE#16#11; // same as above
        CmdReq[14]:=BYTE#16#10; // Remote TCP Port(=4099)
        CmdReq[15]:=BYTE#16#03; // same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=11; // Next state is "Open"

    11: // Open
        IF (_Port_isAvailable=TRUE) AND (PortUsing=FALSE) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#16,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF SendCmd_instance.Done=TRUE THEN
                ClientState:=12; // Next state is "Opening"
```

```
        ELSIF SendCmd_instance.Error=TRUE THEN
            EnableClient:=FALSE;
            ClientState:=0;  // Next state is "Idle"
        END_IF;
    END_IF;

12: // Opening
    IF (J01_Skt4TcpOpening=FALSE) THEN
        IF (J01_Skt4TcpAreaErr=FALSE) AND (J01_Skt4TcpActOpenResult[0]=WORD#16#0000) THEN
            // Successfully opened
            ClientState:=13; // Next state is "Opened"
        ELSE // Fail to open
            EnableClient:=FALSE;
            ClientState:=0;  // Next state is "Idle"
        END_IF;
    END_IF;

13: // Opened
    IF (SendTrigger=TRUE) THEN
        ClientState:=20; // Next state is "Prepare Send"
        SendTrigger:=FALSE;
    END_IF;

20: // Prepare Send
    CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
    CmdReq[0]:=BYTE#16#27; // Command(TCP Socket Send Request)
    CmdReq[1]:=BYTE#16#13; //  same as above
    CmdReq[2]:=BYTE#16#00; // Socket option
    CmdReq[3]:=BYTE#16#04; // Socket number(=4)
    CmdReq[4]:=BYTE#16#82; // Result storage area(=D2320/J01_Skt4TcpSendResult)
    CmdReq[5]:=BYTE#16#09; //  same as above
    CmdReq[6]:=BYTE#16#10; //  same as above
    CmdReq[7]:=BYTE#16#00; //  same as above
    CmdReq[8]:=BYTE#16#00; //  Number of bytes sent(=100)
    CmdReq[9]:=BYTE#16#64; //  same as above
    // CmdReq[10..109] is data.
   // SendDat(WORD Type) and CmdReq(BYTE Type) have different endiannesses, so swap each bytes.
    ToAryByte(SendDat, _HIGH_LOW ,CmdReq[10]);

    SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
    PortUsing:=FALSE;
    ClientState:=21; // Next state is "Send"

21: // Send
    IF (_Port_isAvailable=TRUE) AND (PortUsing=FALSE) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#110,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF SendCmd_instance.Done=TRUE THEN
            ClientState:=22; // Next state is "Sending"
```

```
        ELSIF SendCmd_instance.Error=TRUE THEN
            ClientState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

22: // Sending
    IF (J01_Skt4TcpSending=FALSE) THEN
        IF (J01_Skt4TcpAreaErr=FALSE) AND (J01_Skt4TcpSendResult[0]=WORD#16#0000) THEN
            // Successfully sent
            ClientState:=13; // Next state is "Opened"
        ELSE // Fail to send
            ClientState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

40: // Prepare Close
    IF (J01_Skt4TcpOpened=TRUE) THEN
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0]:=BYTE#16#27; // Command(TCP Socket Close Request)
        CmdReq[1]:=BYTE#16#14; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#04; // Socket number(=4)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D2310/J01_Skt4TcpCloseResult)
        CmdReq[5]:=BYTE#16#09; //  same as above
        CmdReq[6]:=BYTE#16#06; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=41; // Next state is "Close"
    ELSE
        ClientState:=0;  // Next state is "Idle".
    END_IF;

41: // Close
    IF (_Port_isAvailable=TRUE) AND (PortUsing=FALSE) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#8,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF SendCmd_instance.Done=TRUE THEN
            ClientState:=42; // Next state is "Closing".
        ELSIF SendCmd_instance.Error=TRUE THEN
            EnableClient:=FALSE;
            ClientState:=0;  // Next state is "Idle".
        END_IF;
    END_IF;

42: // Closing
    IF (J01_Skt4TcpClosing=FALSE) THEN
        EnableClient:=FALSE;
```

```
            ClientState:=0; // Next state is "Idle"
        END_IF;
END_CASE;
```

ユニット B のユーザプログラム

● グローバル変数

名称	データ型	割付先	保持	ネットワーク公開	コメント
J02_StaArea	_sCJEIP21_User_StaArea		FALSE	公開のみ	ステータスエリアレイアウト変更用
J02_Skt4TcpPsvOpenResult	ARRAY[0..3] OF WORD	%D3300	TRUE	非公開	ソケットオープン要求結果格納用
J02_Skt4TcpRcvResult	ARRAY[0..51] OF WORD	%D3330	TRUE	非公開	受信要求結果および受信データ格納用
J02_Skt4TcpCloseResult	WORD	%D3310	TRUE	非公開	ソケットクローズ要求結果格納用

● デバイス変数

ポート	名称	ネットワーク公開	コメント
Skt4TcpOpening	J02_Skt4TcpOpening	非公開	
Skt4TcpRcving	J02_Skt4TcpRcving	非公開	
Skt4TcpClosing	J02_Skt4TcpClosing	非公開	
Skt4TcpDatRcved	J02_Skt4TcpDatRcved	非公開	
Skt4TcpOpened	J02_Skt4TcpOpened	非公開	
Skt4TcpRcvDatByte	J02_Skt4TcpRcvDatByte	非公開	

● 内部変数

名称	データ型	コメント
EnableServer	BOOL	
ServerState	INT	
CmdDst	_sDNET_ADR	
CmdReq	ARRAY[0..255] OF BYTE	
CmdResp	ARRAY[0..3] OF BYTE	
PortUsing	BOOL	
CmdOption	_sRESPONSE	
SendCmd_instance	SendCmd	
RcvDat	ARRAY[0..49] OF WORD	受信データ格納用

● プログラムコード

```
// State
//  0: Idle
// 10: Prepare Open,          11: Open,          12: Opening,          13: Opened
// 30: Prepare Receive,       31: Receive        32: Receiving
// 40: Prepare Close,         41: Close,         42: Closing

IF (EnableServer=TRUE) THEN
    IF (ServerState=0) THEN
        ServerState:=10; // Next state is "Prepare Open".
    END_IF;
ELSE
    IF (J02_Skt4TcpOpened=TRUE) AND (ServerState < 40) THEN
        ServerState:=40; // Close opened socket. Next state is "Prepare Close".
    END_IF;
END_IF;
```

```
CASE ServerState OF
    0: // Idle
        // Nothing to do

    10: //Prepare Open
        CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
        CmdReq[0] :=BYTE#16#27; // Command(Tcp Socket Passive Open Request)
        CmdReq[1] :=BYTE#16#10; //  same as above
        CmdReq[2] :=BYTE#16#00; // Socket option
        CmdReq[3] :=BYTE#16#04; // Socket number(=4)
        CmdReq[4] :=BYTE#16#82; // Result storage area(=D3300/J02_Skt4TcpPcvOpenResult)
        CmdReq[5] :=BYTE#16#0C; //  same as above
        CmdReq[6] :=BYTE#16#E4; //  same as above
        CmdReq[7] :=BYTE#16#00; //  same as above
        CmdReq[8] :=BYTE#16#10; //  Local port number(=4099)
        CmdReq[9] :=BYTE#16#03; //  same as above
        CmdReq[10]:=BYTE#16#00; // Timeout(= Infinite)
        CmdReq[11]:=BYTE#16#00; // same as above
        CmdReq[12]:=BYTE#16#C0; // Remote IP Addr(=192.168.250.17)
        CmdReq[13]:=BYTE#16#A8; // same as above
        CmdReq[14]:=BYTE#16#FA; // same as above
        CmdReq[15]:=BYTE#16#10; // same as above
        CmdReq[16]:=BYTE#16#10; // Remote TCP Port(=4099)
        CmdReq[17]:=BYTE#16#03; // same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ServerState:=11; // Next state is "Open"

    11: // Open
        IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#18,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF SendCmd_instance.Done=TRUE THEN
                ServerState:=12; // Next state is "Opening"
            ELSIF SendCmd_instance.Error=TRUE THEN
                EnableServer:=FALSE;
                ServerState:=0;  // Next state is "Idle"
            END_IF;
        END_IF;

    12: // Opening
        IF (J02_Skt4TcpOpening=FALSE) THEN
            IF (J02_Skt4TcpAreaErr=FALSE) AND (J02_Skt4TcpPsvOpenResult[0]=WORD#16#0000) THEN
                // Successfully opened
                ServerState:=13; // Next state is "Opened"
            ELSE // Fail to open
                EnableServer:=FALSE;
                ServerState:=0;  // Next state is "Idle"
```

```
        END_IF;
    END_IF;

13: // Opened
    IF (J02_Skt4TcpDatRcved=TRUE) AND (J02_Skt4TcpRcvDatByte > 0) THEN
        ServerState:=30; // Next state is "Prepare Receive"
    END_IF;

30: // Prepare Receive
    CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
    CmdReq[0] :=BYTE#16#27; // Command(TCP Socket Receive Request)
    CmdReq[1] :=BYTE#16#12; //  same as above
    CmdReq[2] :=BYTE#16#00; // Socket option
    CmdReq[3] :=BYTE#16#04; // Socket number(=4)
    CmdReq[4] :=BYTE#16#82; // Result storage area(=D3330/J02_Skt4TcpRcvResult)
    CmdReq[5] :=BYTE#16#0D; //  same as above
    CmdReq[6] :=BYTE#16#02; //  same as above
    CmdReq[7] :=BYTE#16#00; //  same as above
    CmdReq[8] :=BYTE#16#00; //  Number of bytes received(=100)
    CmdReq[9] :=BYTE#16#64; //  same as above
    CmdReq[10]:=BYTE#16#00; // Timeout(= Infinity)
    CmdReq[11]:=BYTE#16#00; // same as above

    SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
    PortUsing:=FALSE;
    ServerState:=31; // Next state is "Receive"

31: // Receive
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#12,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF SendCmd_instance.Done=TRUE THEN
            ServerState:=32; // Next state is "Receiving"
        ELSIF SendCmd_instance.Error=TRUE THEN
            ServerState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

32: // Receiving
    IF (J02_Skt4TcpRcving=FALSE) THEN
        IF (J02_Skt4TcpAreaErr=FALSE) AND (J02_Skt4TcpRcvResult[0]=WORD#16#0000) THEN
            // Successfully received. You can process received data here.
            AryMove(J02_Skt4TcpRcvResult[2], RcvDat[0], UINT#50);
            ServerState:=13; // Next state is "Opened"
        ELSE // Fail to receive
            ServerState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;
```

```
    40: // Prepare Close
        IF (J02_Skt4TcpOpened=TRUE) THEN
            CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
            CmdReq[0]:=BYTE#16#27; // Command(TCP Socket Close Request)
            CmdReq[1]:=BYTE#16#14; //  same as above
            CmdReq[2]:=BYTE#16#00; // Socket option
            CmdReq[3]:=BYTE#16#04; // Socket number(=4)
            CmdReq[4]:=BYTE#16#82; // Result storage area(=D3310/J02_Skt4TcpCloseResult)
            CmdReq[5]:=BYTE#16#0C; //  same as above
            CmdReq[6]:=BYTE#16#EE; //  same as above
            CmdReq[7]:=BYTE#16#00; //  same as above

            SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
            PortUsing:=FALSE;
            ServerState:=41; // Next state is "Close"
        ELSE
            ServerState:=0;  // Next state is "Idle".
        END_IF;

    41: // Close
        IF (_Port_isAvailable=TRUE) AND (PortUsing=FALSE) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#8,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF SendCmd_instance.Done=TRUE THEN
                ServerState:=42; // Next state is "Closing".
            ELSIF SendCmd_instance.Error=TRUE THEN
                EnableServer:=FALSE;
                ServerState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    42: // Closing
        IF (J02_Skt4TcpClosing=FALSE) THEN
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle"
        END_IF;
END_CASE;
```

A-10 Generic デバイスを用いたタグデータリンク設定（Network Configurator Ver.3.74 以降）

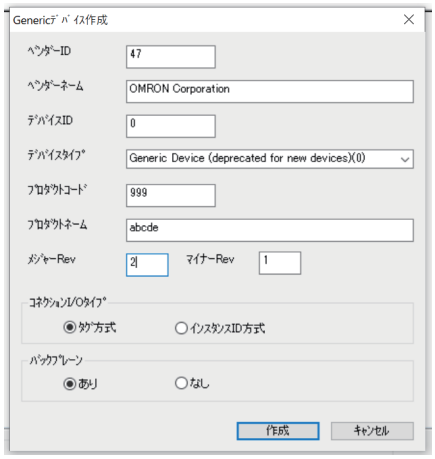
EDS ファイルを持たないデバイスとタグデータリンクを行いたい場合は、Generic デバイスを使用します。
Generic デバイスを使用するには、Network Configurator で Generic デバイスを作成します。
以下に Generic デバイスの作成手順およびタグ / タグセットの作成手順を示します。

参考

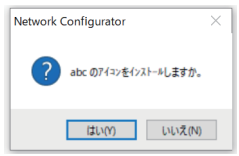
タグ／タグセットの作成以降の手順は、EDS ファイルを持つデバイスと同じです。「7-2 タグデータリンクの設定」を参照してください。

A-10-1 Generic デバイスの作成手順

1 **［EDS ファイル］メニューから［Generic デバイス作成］を選択します。**
Generic デバイス作成画面が表示されます。

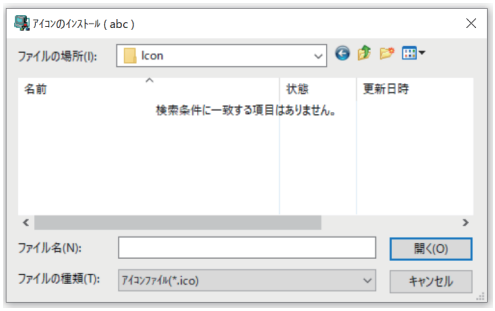


2 **デバイスに関する情報を設定し、［作成］ボタンをクリックします。**
アイコンのインストール確認ダイアログが表示されます。



3 **デバイスのアイコンを設定します。**
- ［はい］を選択した場合

［アイコンのインストール（EDS ファイル名）］のダイアログが表示されます。

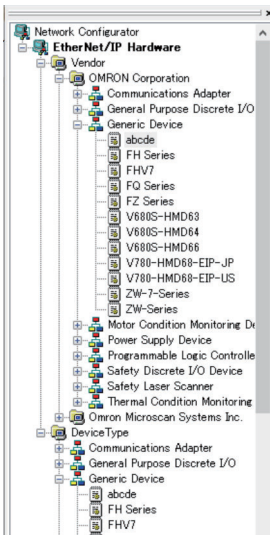


- ［いいえ］を選択した場合

Network Configurator のデフォルトのアイコンが設定されます。

4 **EDS ファイルに設定するアイコンファイル（*.ico）を選択し、［開く］ボタンをクリックします。**

作成した Generic デバイスがハードウェアリストに追加されます。



A-10-2　Generic デバイスのタグ／タグセットの作成

タグ／タグセットの作成方法として、タグ方式とインスタンス ID 方式の 2 種類あり、それぞれ作成手順を記載します。

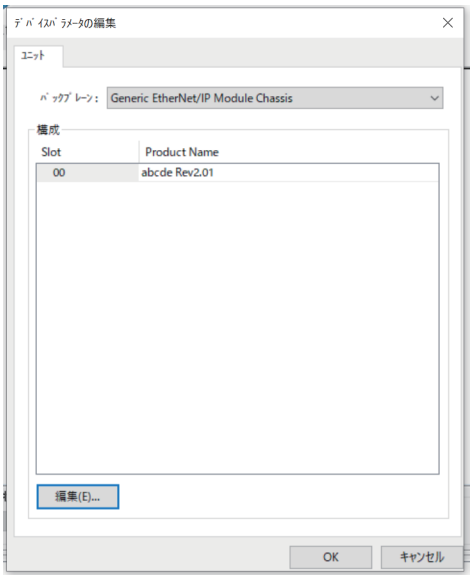
Generic デバイスの作成時に［コネクション I/O タイプ］で選択した方式となります。

タグ方式の場合の作成手順

1 **作成した Generic デバイスをネットワークウィンドウに追加します。**

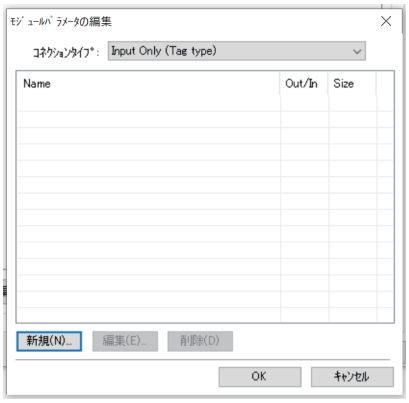
2 **デバイスのアイコンをダブルクリックします。**

［デバイスパラメータの編集］ダイアログボックスが表示されます。



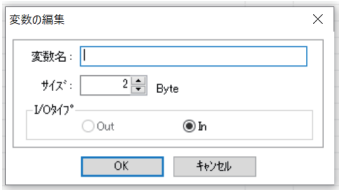
3 **［ユニット］タブから［構成］の Slot 番号［00］をクリックして選択し、［編集］ボタンをクリックします。**

［モジュールパラメータの編集］ダイアログボックスが表示されます。



4 **［コネクションタイプ］から［Input Only (Tag type)］または［Input & Output (Tag type)］を選択し、［新規］ボタンをクリックします。**

［変数の編集］ダイアログボックスが表示されます。



5 **変数の下記のパラメータを設定します。**

- ［変数名］

ネットワーク変数の名称を入力します。(例：VarInputOnly)

- ［サイズ］

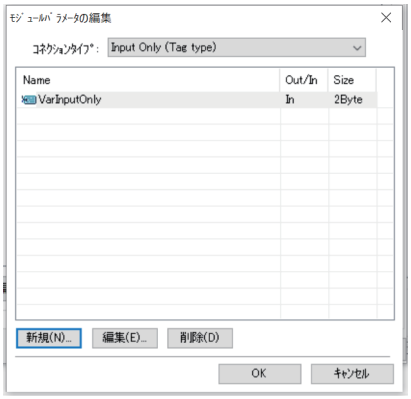
タグのサイズを、バイト単位で入力します。

- ［I/O タイプ］

［コネクションタイプ］が［Input & Output (Tag type)］の場合、［Out］または［In］を選択します。

6 **［OK］ボタンをクリックします。**

［モジュールパラメータの編集］ダイアログボックスが表示され、追加した変数が表示されます。

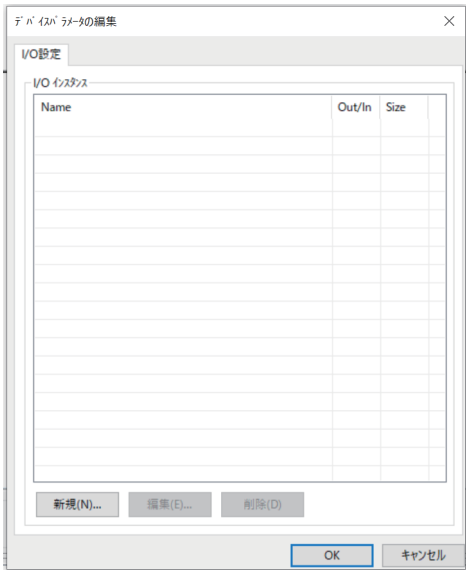


7 **手順 4 ～ 6 を繰り返して、必要な変数を登録します。登録が完了したら［OK］ボタンをクリックします。**

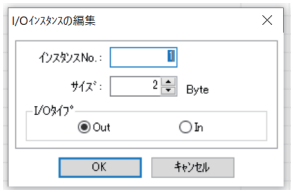
インスタンス ID 方式の場合の作成手順

1 **作成した Generic デバイスをネットワークウィンドウに追加します。**

2 **デバイスのアイコンをダブルクリックします。**
［デバイスパラメータの編集］ダイアログボックスが表示されます。



3 **［I/O 設定］タブから、［新規］ボタンをクリックします。**
［I/O インスタンスの編集］画面が表示されます。



4 **下記のパラメータを入力します。**

- ［インスタンス No］

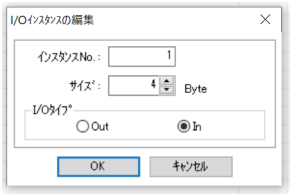
I/O インスタンスのインスタンス ID を入力します。

- ［サイズ］

タグのサイズで、バイト単位で入力します。

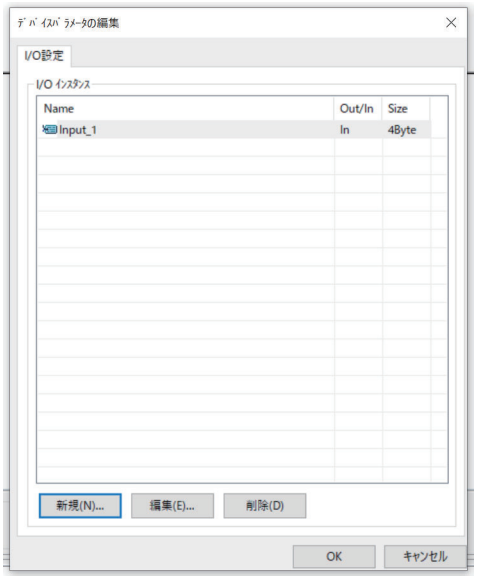
- ［I/O タイプ］

Out または In を選択します。



5 **［OK］ボタンをクリックします。**

［デバイスパラメータの編集］ダイアログボックスが表示され、追加した I/O インスタンスが［I/O 設定］に表示されます。



6 **手順 3 ～ 5 を繰り返して、必要な I/O インスタンスを登録します。登録が完了したら［OK］ボタンをクリックします。**

A-11 バージョン関連情報

CPU ユニットのユニットバージョンの更新により、追加または変更されたサポート機能を紹介します。

● 機能仕様の追加と変更

追加または変更された機能仕様ごとに、CPU ユニットのユニットバージョンと Sysmac Studio のバージョンを示します。

種別		追加／変更	参照先	ユニットバージョン	Sysmac Studio のバージョン
構造体オフセット機能	CJ	追加	A-58	Ver.1.02	Ver.1.03
形 CJ1W-EIP21S の装着		追加	−	Ver.1.67	V.er.1.60

I

索引

索引

Numerics

100M LED 1-11
10M LED 1-11
7 セグメント LED 表示 1-12

A

*_AdjTmCmd（時計情報自動調整実行スイッチ）.....3-7
*_AdrMismatchErr（アドレス不一致）.....3-13

B

BIN 形式 9-16
*_BootpSvrErr（BOOTP サーバ異常）.....3-13
BOOTP クライアント 1-19
bye 9-12

C

cd 9-10
CIDR 機能 4-3
CIPClose 命令 8-4
CIPOpen 命令 8-4
CIPRead 命令 8-4
CIPSend 命令 8-4
CIPUCMMRead 命令 8-4
CIPUCMMSend 命令 8-4
CIPUCMMWrite 命令 8-4
CIPWrite 命令 8-4
CIP 通信 1-17
CIP 通信命令の種類 8-4
CIP 通信命令の使用方法 8-5
CIP メッセージ通信 1-18
CIP メッセージ通信サービス機能の概要 8-2
CIP メッセージ通信サービスの機能・性能一覧 8-2
CIP メッセージ通信の使い方 8-3
CIP ルーチング機能 1-18
CJ ユニットデバイス変数 3-3
close 9-12
COMM LED 1-11
*_CommParamErr（通信パラメータ設定不正）.....3-12
*_CommSta1（通信ステータス 1）.....3-11
*_CommSta2（通信ステータス 2）.....3-12
*_CommSta3（通信ステータス 3）.....3-13
CPU ユニットとのデータ交換 3-2

D

delete 9-11
dir 9-8
*_DNSSvrErr（DNS サーバ異常）.....3-13
DNS 設定 5-3

E

EDS ファイルの管理 A-41
*_ErrLogStoreSta（異常履歴あり）.....3-11
*_EstbTargetSta（正常ターゲットノード）.....3-14
x. EstbTargetSta. TargetSta（正常ターゲットノード）.
.....3-27
x. EstbTargetSta. TargetStaWd（正常ターゲットノード）
.....3-27
Ethenet Link オブジェクト 8-38
EtherNet/IP ユニット仕様 1-7
EtherNet/IP ユニット設定一覧 5-1
Ethernet 接続コネクタ 2-10, 2-11
*_ETNAdvSetErr（Ethernet 拡張設定異常）.....3-13
*_ETNBaseSetErr（Ethernet 基本設定異常）.....3-13
*_ETNOnlineSta（オンライン）.....3-10

F

*_FTPSta（FTP ステータス）.....3-12
FTP サーバ 1-19, 5-9
FTP サーバ機能の概要 9-2
FTP サーバ機能の使用手順 9-5
FTP サーバ機能の使用例 9-6
FTP 設定ダイアログ 5-9

G

General Status 8-32
get 9-10

I

Indentity オブジェクト 8-33
*_IPAdrChgErr（運転中の IP アドレス変更）.....3-10
*_IPAdrDupErr（IP アドレス重複異常）.....3-9
*_IPAdrTblErr(IP アドレステーブル異常).....3-13
*_IPRouterTblErr（IP ルータテーブル異常）.....3-13
IP アドレス 5-3, 5-10, 5-12, 5-14
IP アドレス設定方法 5-3
IP アドレスの構成 4-2
IP アドレスの設定手順 4-5
IP アドレスの割り当て 4-3
IP アドレス表示・設定エリア 3-15
IP ルータテーブルの設定例 5-4

L

*_LANHwErr（通信コントローラ異常）.....3-9
LED 表示の意味 15-10
LINK 設定 5-8
*_LkOffErr（リンクオフ異常）.....3-9
*_LkSta（リンクステータス）.....3-12
ls 9-8

M

mdelete 9-12
*_MemErr（不揮発性メモリ異常） 3-13
mget 9-11
MIB オブジェクトの詳細説明 11-5
MIB グループ一覧 11-4
MIB 体系図 11-4
mkdir 9-9
mput 9-11
MS LED 1-11
*_MultiSwOnErr（複数スイッチ同時 ON 異常） 3-11

N

*_NetErr（ネットワーク異常発生中） 3-9
Network Configurator 1-6
　CPU ユニットの USB ポートによる接続方法（NJ CPU ユニット） 7-56
　Ethernet による接続方法 7-54
NS LED 1-11

O

open 9-8

P

PING コマンド 6-2
PLC オブジェクト 8-41
put 9-11
pwd 9-9

Q

quit 9-12

R

*_RegTargetSta（登録ターゲットノード） 3-14
x. RegTargetSta. TargetSta（登録ターゲットノード） 3-26
x. RegTargetSta. TargetStaWd（登録ターゲットノード） 3-26
rename 9-9
rmdir 9-9
RPI 7-42
RPI の変更方法 14-12
*_RTblErr（ルーチングテーブル異常） 3-13

S

_sCJEIP21_User_StaArea（ステータスエリア用のデータ型） 3-26
SD メモリカード機能
　種類 9-14
　初期化 9-15
　ファイルの種類 9-15
　変数データのフォーマット 9-16
*_Skt1Cmd（ソケットサービス要求スイッチ 1）～ _Skt8Cmd（ソケットサービス要求スイッチ 8） 3-17
*_Skt1ParamArea（ソケットサービスパラメータエリア 1）～ *_Skt8ParamArea ソケットサービスパラメータエリア 8 3-22
*_Skt1TcpConSta（TCP ソケット No.1 コネクション状態）～ *_Skt8TcpConSta（TCP ソケット No.8 コネクション状態） 3-21
*_Skt1TcpRcvDatByte（TCP ソケット No.1 受信データバイト数）～ *_Skt8TcpRcvDatByte（TCP ソケット No.1 受信データバイト数） 3-21
*_Skt1TcpSta（TCP ソケット No.1 ステータス）～ *_Skt8TcpSta（TCP ソケット No.8 ステータス） 3-19
*_Skt1UdpSta（UDP ソケット No.1 ステータス）～ *_Skt8UdpSta（UDP ソケット No.8 ステータス） 3-18
*_SktForceCloseCmd（ソケット強制クローズスイッチ） 3-7
SNMP（エージェント） 1-21
SNMP（エージェント）機能に必要な設定一覧 11-17
SNMP（エージェント）機能の概要 11-2
SNMP（エージェント）機能の使用手順 11-17
SNMP エージェント機能 11-2
SNMP サービス 5-11
SNMP 仕様 11-3
SNMP 設定ダイアログ 5-11
SNMP トラップ 5-13
SNMP トラップ機能 11-3
SNMP トラップ設定ダイアログ 5-13
SNMP メッセージ 11-3
*_SNTPSvrErr（SNTP サーバ異常） 3-13
SNTP サーバの時計情報 5-10
SNTP 設定ダイアログ 5-10
Sysmac Studio 1-6

T

*_TagDbErr（タグデータベース異常） 3-12
*_TagRefreshErr（タグリフレッシュ異常） 3-12
*_TargetPLCErrSta（ターゲットノード PLC 本体異常情報） 3-8
x. TargetPLCErrSta. TargetSta（ターゲットノード PLC 本体異常情報） 3-28
x. TargetPLCErrSta. TargetStaWd（ターゲットノード PLC 本体異常情報） 3-28
*_TargetPLCMdSta（ターゲットノード PLC 本体運転情報） 3-8
x. TargetPLCMdSta. TargetSta（ターゲットノード PLC 本体運転情報） 3-28
x. TargetPLCMdSta. TargetStaWd（ターゲットノード PLC 本体運転情報） 3-28
TCP/IP Interface オブジェクト 8-36
TCP/IP 設定ダイアログ 5-3
TCP ソケット No.1 ～ 8 コネクション状態 3-21
TCP ソケット No.1 ～ 8 受信データバイト数 3-21
TCP ソケット No.1 ～ No.8 ステータス 3-19

*_TDLAllRunSta（全タグデータリンク通信中）.....3-12
*_TDLErr（タグデータリンク異常）.....3-11
*_TDLOpnErr（照合異常）.....3-11
*_TDLOprSta（タグデータリンク動作中）.....3-10
*_TDLRunSta（タグデータリンク通信中）.....3-12
*_TDLStartCmd（タグデータリンク開始スイッチ）...3-7
*_TDLStopCmd（タグデータリンク停止スイッチ）...3-7
type 9-10

U

_uCJEIP21_TargetSta（正常ターゲットノード）.....3-26
_uCJEIP21_TargetSta（登録ターゲットノード）.....3-26
_uCJEIP21_TargetSta（
ターゲットノード PLC 本体異常情報）.....3-26
_uCJEIP21_TargetSta
（ターゲットノード PLC 本体運転情報）.....3-26
UDP ソケット No.1 ～ No.8 ステータス 3-18
*_UnitErr（ユニット異常発生中）.....3-9
*_UnitMemErr（ユニットメモリ異常）.....3-9
*_UnitSta1（ユニットステータス 1）.....3-9
*_UnitSta2（ユニットステータス 2）.....3-10
user 9-8
*_UserStaAreaCfgErr
（ステータスエリアレイアウト設定異常）.....3-10
*_UserStaAreaEnblSta（ユーザ設定エリア有効）....3-11

W

Windows ファイアウォールの設定の変更手順 A-46

あ

宛先アドレス 5-4

い

異常一覧 15-34
異常の内容 15-37
異常履歴 15-28
異常履歴コード一覧 15-29
イベントコード 15-34
イベント名 15-34
イベントログ 15-33

う

受付認証 1 5-12
受付認証 2 5-12

お

オリジネータ変数 7-42

か

外形寸法 1-16
外部出力にアクセスする場合の留意事項 A-61
各部の名称と機能 1-9

き

機能
　他シリーズとの機能比較 A-3

く

グローバルアドレス 4-10

け

ゲートウェイアドレス 5-4
ケーブルの接続方法 2-11

こ

コネクション I/O タイプ 7-41, 7-42
コネクション数の計算方法 14-5
コネクションステータス 15-21
コネクション設定
　一括編集 7-41
　個別編集 7-40
　登録デバイス一覧 7-38
コネクションタイプ 7-41, 7-42
コネクションの自動設定 7-44
コネクションのチェック 7-75
コネクション名 7-41, 7-42
コマンド一覧 9-7
コミュニティ名 5-12, 5-14
コントローラステータス 7-13

さ

サーバ指定方法 5-10
サービスデータ／レスポンスデータなどの入出力のための
　配列型変数の用意 8-13
サブネットマスク 4-3, 5-3
サンプルプログラム
　CIP メッセージ通信機能 8-14
　タグデータリンクでのラダープログラム 7-80

し

自 IP アドレス 4-4
指定時刻 5-10
指定方法 5-12, 5-14
自動コネクション設定 7-45
受信データ処理時間 14-27
出力 TRUE 応答時間 14-27
出力変数の動作とタイミング 8-28

す

スイッチングハブ 1-5
　機能 2-5
　種類 2-5

選定時の留意事項 2-6
ステータスエリア設定ダイアログ 5-15
ステータスエリアのユーザ定義設定 3-25

せ

正常ターゲットノード 3-14
設置場所 5-11
ゼネラルステータスコード 8-29

そ

送信データ処理時間 14-27
ソケットサービス 1-21
ソケットサービス強制クローズスイッチ 3-24
ソケットサービスパラメータエリア 1 ～ 8 3-22
ソケットサービス要求スイッチ 1 ～ 8 3-17
ソフトスイッチ 3-6

た

ターゲットデバイス 7-42
ターゲットノード PLC 本体異常情報 3-8
ターゲットノード PLC 本体運転情報 3-8
ターゲット変数 7-42
代替 DNS サーバ 5-3
タイムアウト時間 5-10
タイムアウト値 7-41, 7-42
タグ 7-4
タグ／タグセットの作成 7-24
タグセット 7-4
タグデータリンク
NJ CPU 以外の機種とタグデータリンクを行う場合について 7-85
概要 7-2
機能・性能一覧 7-8
設定 7-19
データエリア 7-4
データの同期性について 7-15
タグデータリンク（サイクリック通信） 1-17
タグデータリンク入出力最大応答時間 14-26
タグデータリンクによる、タスク周期への影響 14-26
タグデータリンクの開始／停止 7-12
タグデータリンクの自動起動 7-68
タグデータリンクの帯域使用率と RPI の関係 14-10
タグデータリンクの通信のしかた 14-2
タグデータリンクのデバイス帯域使用状況の確認 14-8
タグデータリンクパラメータ
ダウンロード 7-58
タグデータリンクパラメータのアップロード
一括してアップロードする 7-61
特定のデバイスから個別にアップロードする 7-63
タグデータリンクパラメータの照合 7-64
タグデータリンクパラメータの設定／ダウンロード . 7-10
タスク周期とパケットインターバル（RPI）の関係 14-26

つ

ツイストペアケーブル 1-5
接続方法 2-10
設置時の注意点 2-9, 2-10
布設時の注意点 2-8
通信ステータス 1 3-11
通信ステータス 2 3-12
通信ステータス 3 3-13
通信テスト 6-2
通信負荷の調整 14-7

て

データ処理時間の概要 14-23
データ処理時間の計算例 14-25
データ転送のタイミング 14-22
デバイスコネクション構成表示 7-47
デバイスステータス表示 7-79
デバイス帯域使用状況の調整方法 14-11
デバイス単位のタグデータリンクの開始／停止 7-68
デバイスの登録 7-22
デバイスの変更 7-76
デバイスパラメータのクリア 7-70
デバイスパラメータの照合方法 7-66
デフォルトゲートウェイ 5-3

と

登録ターゲットノード 3-14
時計自動調整機能 1-20
時計情報自動調整機能
仕様 10-2
使用手順 10-3
必要な設定一覧 10-3
時計情報自動調整機能の概要 10-2
ドメイン名 5-3
トラップ 1 5-14
トラップ 2 5-14
トラブルシューティング 15-48

に

入力 TRUE 応答時間 14-27
認証トラップを送信する 5-11

ね

ネットワーク構成の照合方法 7-64
ネットワーク構成ファイルの保存 7-71
ネットワーク構成ファイルの読出 7-73
ネットワーク伝送遅延時間 14-27
ネットワーク変数 7-9
Network Configurator にインポートする 7-34

の

ノードアドレス設定スイッチ 1-15
ノードアドレスの設定 2-2

は

バージョン 5-14
パケットインターバル（RPI）.....7-11, 7-41, 14-27
パケットインターバル（RPI）と、使用通信帯域幅の関係 14-3
パケットインターバル（RPI）の精度 14-6
パケットインターバル（RPI）の設定 14-2
パスワード 5-9

ひ

表示部（LED）.....1-11

ふ

プライベートアドレス 4-10

へ

変数のメモリ確保
　留意するべきケース例 A-58

ほ

ポート No. 5-9, 5-10, 5-11
ホストコンピュータからの使用例 9-17, 9-18
ホスト名 5-10, 5-12, 5-14
ホスト名指定 1-20

ま

マルチキャストとユニキャスト 7-12
マルチキャストフィルタ 2-5

め

メッセージサービス機能伝送遅延時間 14-28
メンテナンス 15-52

ゆ

ユーザ定義設定 3-25
優先 DNS サーバ 5-3
ユニット交換 15-54
ユニットステータス 1 3-9
ユニットステータス 2 3-10
ユニット設定 5-2
ユニットの装着 2-3
ユニット番号設定スイッチ 1-14
ユニット番号の設定 2-2

り

リクエストパス（IOI）について 8-11
リクエストパス入力のための構造体型変数の用意 8-12

る

ルートパス 8-6

れ

レスポンスコード 8-29
連絡先 5-11

ろ

ログイン名 5-9

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**（クイック　オムロン）

携帯電話の場合、

055-982-5015（有料）をご利用ください。

受付時間：9:00～17:00（土・日・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ございます。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いただけます。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

Man. No. SBCD-355N　2025年4月現在

©OMRON Corporation 2012 - 2025 All Rights Reserved.
お断りなく仕様などを変更することがありますのでご了承ください