OMRON

産業用PCプラットフォーム

NYシリーズ

IPC マシンコントローラ
産業用パネル型PC／産業用ボックス型PC

ユーザーズマニュアル
内蔵EtherNet/IP™ポート編

形NY532-1500
形NY532-1400
形NY532-1300
形NY532-5400
形NY512-1500
形NY512-1400
形NY512-1300

産業用パネル型PC
産業用ボックス型PC



SYSMAC
always in control

SBCD-369H

おことわり

(1) 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。

(2) 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。

(3) 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、ドイツ Beckhoff Automation GmbH によりライセンスされ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。
- Intel、Intel Core は、米国およびその他の国における Intel Corporation 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。
- 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/nj_info_j/ をご覧ください。

はじめに

このたびは、NY シリーズ IPC マシンコントローラ 産業用パネル型 PC ／産業用ボックス型 PC をお買い上げいただきまして、ありがとうございます。
本マニュアルでは、対象となる製品の産業用パネル型 PC と産業用ボックス型 PC を総称して「NY シリーズ産業用 PC」と記載する場合があります。また、NY シリーズ産業用 PC のリアルタイム OS に組み込まれたコントローラ機能が直接制御する範囲を「コントローラ」と記載する場合があります。
本マニュアルは、NY シリーズコントローラを使用する上で、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 IEC 61131-3 もしくは国内標準規格 JIS B 3503 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。

- NY シリーズ IPC マシンコントローラ 産業用パネル型 PC
  - 形 NY532-15□□
  - 形 NY532-14□□
  - 形 NY532-13□□
  - 形 NY532-5400

- NY シリーズ IPC マシンコントローラ 産業用ボックス型 PC
  - 形 NY512-15□□
  - 形 NY512-14□□
  - 形 NY512-13□□

各製品の仕様の一部または制約事項が、他のマニュアルに記載されている場合があります。🕮「分冊構成 (P.2)」および🕮「関連マニュアル (P.23)」をご確認ください。

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										
	NY シリーズ IPC マシンコントローラ 産業用パネル型 PC ユーザーズマニュアル ハードウェア編	NY シリーズ IPC マシンコントローラ 産業用ボックス型 PC ユーザーズマニュアル ハードウェア編	NY シリーズ IPC マシンコントローラ 産業用パネル型 PC／産業用ボックス型 PC ユーザーズマニュアル セットアップ編	NY シリーズ IPC マシンコントローラ 産業用パネル型 PC／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編	NY シリーズ コマンドリファレンスマニュアル 基本編	NY シリーズ IPC マシンコントローラ 産業用パネル型 PC／産業用ボックス型 PC ユーザーズマニュアル モーション制御編	NY 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	NY シリーズ IPC マシンコントローラ 産業用パネル型 PC／産業用ボックス型 PC ユーザーズマニュアル 内蔵 EtherCAT ポート編	NY シリーズ IPC マシンコントローラ 産業用パネル型 PC／産業用ボックス型 PC ユーザーズマニュアル 内蔵 EtherNet/IP ポート編	NJ/NY シリーズ NC 統合コントローラ ユーザーズマニュアル	NY シリーズ トラブルシューティングマニュアル
NY シリーズパネル型 PC の概要を知る	○										
NY シリーズボックス型 PC 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									○		
セットアップを行う*1			○								
初期設定を行う場合											
コントローラの使用準備を行う場合											
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									○		
数値制御を行う場合										○	
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									○		
数値制御を行う場合										○	
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									○		
数値制御を行う場合										○	
異常管理機能やトラブル発生時の対処方法を知る*2										△	○
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									○		

*1. Windows 側のセットアップ方法やユーティリティなどの使い方については、📖『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル セットアップ編 (SBCA-441)』を参照してください。

*2. 異常管理の考え方と異常項目については、🕮『NY シリーズ トラブルシューティングマニュアル（SBCA-438）』を参照してください。ただし、△マークがついているマニュアルの製品に特有の異常の詳細については、△マークがついているマニュアルを参照してください。

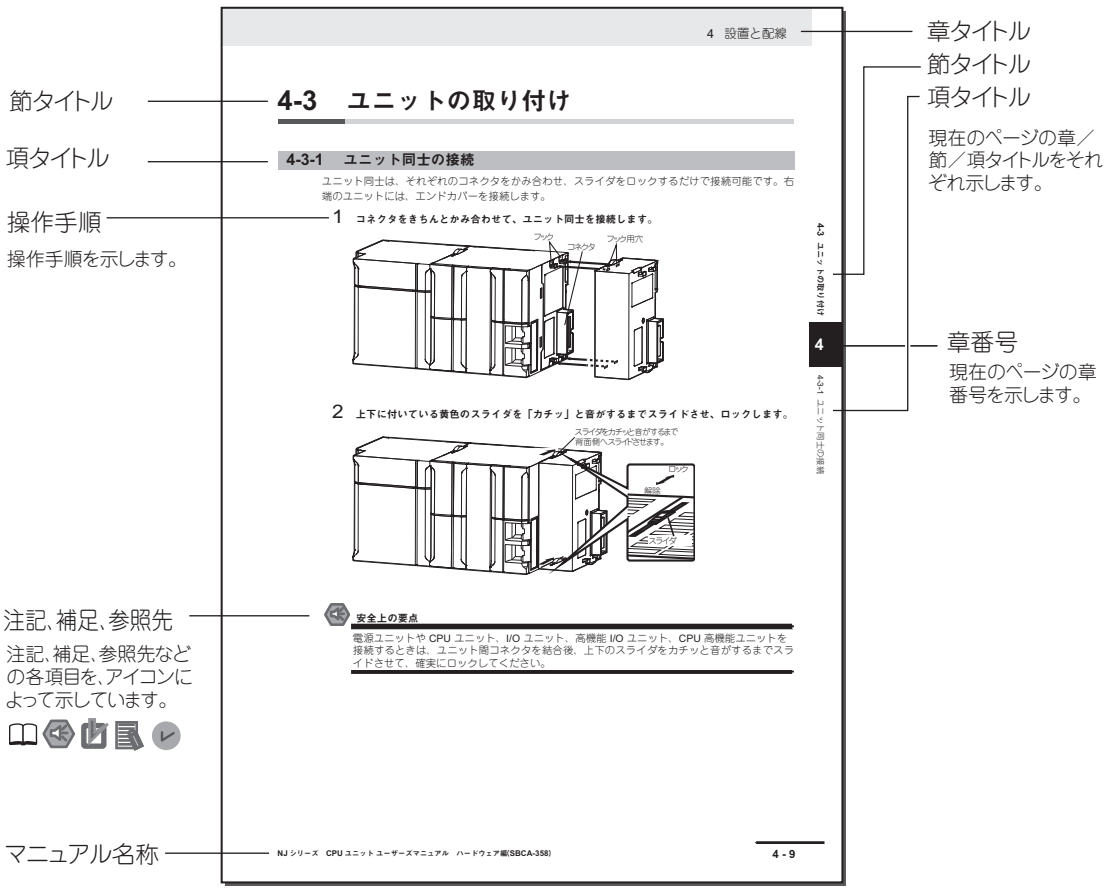
マニュアルの読み方

このマニュアルに記載している機能の説明は、NJ/NXシリーズと共通で書かれている部分があります。そのため、以下の点に注意してください。

- NJ/NX/NY シリーズの共通機能には、同じ機能名称を使用しています。「CPU ユニット名称機能」、「CPU ユニットへの書込プロテクト機能」など、機能名称に「CPU ユニット」を含む場合、NY シリーズでは、「コントローラ」のことを指します。
- 本マニュアルの機能仕様の一覧で、「CPU ユニット」と記載している場合も、NY シリーズでは、「コントローラ」のことを指します。

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

コントローラや 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 内蔵EtherNet/IPポート関連のシステム定義変数

4 IPアドレスを決定する方法

5 Sysmac Studioによる内蔵EtherNet/IPポート設定一覧

6 通信テスト

7 タグデータリンク機能

8 CIPメッセージ通信機能

9 ソケットサービス機能

10 FTPサーバ機能

11 FTPクライアント機能

12 SNMP（エージェント）機能

13 通信性能と通信負荷

14 ネットワークの通信ステータス確認と処置方法

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 5
表記についての留意事項 5
目次構成 **7**
ご承諾事項 **14**
安全上の注意 **16**
安全上の要点 **17**
使用上の注意 **18**
法規と規格 **19**
バージョン **20**
バージョンの確認方法 20
ユニットバージョンと周辺ツールのバージョン 22
関連マニュアル **23**
マニュアル改訂履歴 **25**

第 1 章　概要

1-1　概要 **1-2**
1-1-1　EtherNet/IP の特長 1-2
1-1-2　NY シリーズ産業用 PC の内蔵 EtherNet/IP ポートの特長 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　内蔵 EtherNet/IP ポート **1-8**
1-3-1　仕様 1-8
1-3-2　各部の名称と機能 1-10
1-4　通信サービスの概要 **1-12**
1-4-1　Windows との IP 通信サービス 1-12
1-4-2　CIP 通信（Common Industrial Protocol）サービス 1-12
1-4-3　IP ルーティング機能 1-14
1-4-4　BOOTP クライアント機能 1-17
1-4-5　FTP サーバ機能 1-17
1-4-6　FTP クライアント機能 1-18
1-4-7　ソケットサービス機能 1-19
1-4-8　ホスト名指定機能 1-20
1-4-9　SNMP（エージェント）機能 1-20
1-5　EtherNet/IP 通信の使用手順 **1-22**

第 2 章　Ethernet ネットワークの配線

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

2-2 ネットワークの布設 2-6
2-2-1 ネットワーク布設時の基本的注意事項 2-6
2-2-2 ネットワーク構成機器の推奨品 2-6
2-2-3 ツイストペアケーブル布設時の注意事項 2-6
2-2-4 スイッチングハブの設置・接続時の注意事項 2-10
2-3 ネットワークへの接続 2-11
2-3-1 Ethernet 接続コネクタ 2-11
2-3-2 ケーブルの接続方法 2-12

第 3 章 内蔵 EtherNet/IP ポート関連のシステム定義変数

3-1 内蔵 EtherNet/IP ポートのシステム定義変数 3-2
3-2 システム定義変数一覧 3-3
3-3 システム定義変数の個別仕様 3-13

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 4-4
4-2 コントローラの自 IP アドレス設定方法 4-5
4-2-1 IP アドレスを設定する方法 4-5
4-2-2 IP アドレスの設定手順 4-6
4-2-3 オンライン接続する 4-8
4-2-4 使用中の IP アドレスの確認方法 4-10
4-3 プライベート／グローバルアドレス 4-11
4-3-1 グローバルアドレスとプライベートアドレス 4-11
4-3-2 内蔵 EtherNet/IP ポートがプライベートアドレスの場合 4-12
4-3-3 内蔵 EtherNet/IP ポートがグローバルアドレスの場合 4-13

第 5 章 Sysmac Studio による 内蔵 EtherNet/IP ポート設定一覧

5-1 ［TCP/IP 設定］ダイアログ 5-2
5-2 ［LINK 設定］ダイアログ 5-6
5-3 ［FTP 設定］ダイアログ 5-7
5-4 ［SNMP 設定］ダイアログ 5-8
5-5 ［SNMP トラップ設定］ダイアログ 5-10
5-6 ［CIP 設定］ダイアログ 5-12

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 内蔵 EtherNet/IP ポートでの使用方法 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 7-3
7-1-3 タグデータリンクの機能・性能一覧 7-7
7-1-4 使用方法の概要 7-8
7-1-5 タグデータリンクの開始／停止 7-11
7-1-6 コントローラステータス 7-11
7-1-7 タグデータリンクするデータの同期性 7-13

7-2 タグデータリンクの設定 7-17
7-2-1 Network Configurator の起動 7-17
7-2-2 タグデータリンク設定手順 7-19
7-2-3 デバイスの登録 7-20
7-2-4 タグ／タグセットの作成 7-22
7-2-5 コネクション設定 7-35
7-2-6 ウィザードによるコネクションの作成 7-46
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 7-49
7-2-8 Network Configurator のネットワーク接続手順 7-52
7-2-9 タグデータリンクパラメータのダウンロード 7-57
7-2-10 タグデータリンクパラメータのアップロード 7-60
7-2-11 タグデータリンクパラメータの照合 7-63
7-2-12 タグデータリンクの開始／停止 7-66
7-2-13 デバイスパラメータのクリア 7-68
7-2-14 ネットワーク構成ファイルの保存 7-70
7-2-15 ネットワーク構成ファイルの読出 7-72
7-2-16 コネクションのチェック 7-74
7-2-17 デバイスの変更 7-75
7-2-18 デバイスステータス表示 7-76

7-3 タグデータリンクのラダープログラム 7-77
7-3-1 タグデータリンクでのラダープログラム 7-77
7-3-2 タグデータリンクに関連するステータス 7-81

7-4 他機種とのタグデータリンク 7-82

第 8 章　CIP メッセージ通信機能

8-1 CIP メッセージ通信機能の概要 8-3
8-1-1 CIP メッセージ通信サービス機能の概要 8-3
8-1-2 メッセージ通信サービスの機能・性能一覧 8-3

8-2 CIP メッセージ通信のクライアント機能 8-4
8-2-1 概要 8-4
8-2-2 CIP 通信命令の種類 8-5
8-2-3 CIP 通信命令の使用方法 8-6
8-2-4 ルートパス 8-7
8-2-5 リクエストパス（IOI） 8-10
8-2-6 サービスデータ／レスポンスデータ 8-14
8-2-7 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） 8-15
8-2-8 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） 8-20
8-2-9 動作タイミング 8-27
8-2-10 レスポンスコード 8-28

8-3 CIP メッセージ通信のサーバ機能 8-32
8-3-1 CIP オブジェクトにアクセスするための CIP メッセージ構造 8-32
8-3-2 変数にアクセスするための CIP メッセージ構造 8-33

8-4 リクエストパスの指定 8-34
8-4-1 CIP オブジェクトの指定例 8-34
8-4-2 変数の指定例 8-34
8-4-3 Logical Segment 8-35
8-4-4 Data Segment 8-35
8-4-5 リクエストパスでの変数名の指定方法 8-36

8-5 CIP オブジェクトサービス 8-40
8-5-1 内蔵 EtherNet/IP ポート宛 CIP オブジェクトの種類 8-40

8-5-2 Identity オブジェクト（クラス ID：01Hex）....8-41
8-5-3 TCP/IP Interface オブジェクト（クラス ID：F5Hex）....8-44
8-5-4 Ethernet Link オブジェクト（クラス ID：F6 Hex）....8-47
8-5-5 PLC オブジェクト（クラス ID：C4Hex）....8-51

8-6 変数への読出し／書込みサービス....8-52
8-6-1 変数読出しサービス....8-52
8-6-2 変数書込みサービス....8-53

8-7 変数のデータタイプ一覧....8-55
8-7-1 データタイプコード....8-55
8-7-2 共通フォーマット....8-55
8-7-3 基本型 (Elementary Data Types)....8-56
8-7-4 派生型 (Derived Data Types)....8-58

第 9 章　ソケットサービス機能

9-1 ソケット通信の基礎知識....9-2
9-1-1 ソケットとは....9-2
9-1-2 ソケットサービスで使用できるポート No.....9-2

9-2 プロトコルの基礎知識....9-3
9-2-1 TCP と UDP の相違点....9-3
9-2-2 送信データのフラグメント（分割）....9-5
9-2-3 データの受信処理....9-7
9-2-4 一斉同報送信....9-10

9-3 内蔵 EtherNet/IP ポートのソケットサービス概要....9-11
9-3-1 概要....9-11
9-3-2 使用手順....9-11

9-4 ソケットサービス機能に必要な設定....9-12

9-5 ソケットサービス用命令....9-13

9-6 ソケットサービスの利用の詳細....9-14
9-6-1 ソケットサービスの使用方法....9-14
9-6-2 ソケットサービスの使用手順....9-14
9-6-3 通信時の出力変数のタイミングチャート....9-17
9-6-4 UDP のサンプルプログラム....9-18
9-6-5 TCP のサンプルプログラム....9-23

9-7 ソケットサービス使用上の留意事項....9-29
9-7-1 UDP/TCP ソケットサービス共通の留意事項....9-29
9-7-2 UDP ソケットサービスのみの留意事項....9-29
9-7-3 TCP ソケットサービスのみの留意事項....9-30

第 10 章 FTP サーバ機能

10-1 FTP サーバ機能の概要....10-2
10-1-1 概要....10-2
10-1-2 仕様....10-3

10-2 FTP サーバ機能の詳細....10-4
10-2-1 対象ファイル....10-4
10-2-2 FTP サーバへの接続方法....10-4

10-3 FTP サーバ機能の使用手順....10-7
10-3-1 使用手順....10-7
10-3-2 FTP サーバ機能に必要な設定一覧....10-7

10-4 FTP サーバ機能の使用例....10-8

10-5 FTP コマンドの使用方法....10-10
10-5-1 コマンド一覧....10-10
10-5-2 コマンドの使用方法....10-11

10-6 SD メモリカード機能の使い方 10-17
10-6-1 ファイルの種類 10-18
10-6-2 変数データのフォーマット 10-19
10-7 ホストコンピュータからの使用例 10-20

第 11 章 FTP クライアント機能

11-1 FTP クライアント機能によるファイル転送 11-2
11-1-1 ファイル転送の実行方法 11-2
11-1-2 接続可能な FTP サーバ 11-3
11-1-3 ファイル転送のオプション機能 11-3
11-1-4 その他の機能 11-4
11-2 FTP クライアント通信命令 11-5
11-2-1 FTP クライアント通信命令の機能 11-5
11-2-2 FTP クライアント通信命令の制限 11-7
11-3 FTP クライアント機能の使用例 11-8

第 12 章 SNMP（エージェント）機能

12-1 SNMP（エージェント）機能の概要 12-2
12-1-1 SNMP（エージェント）機能とは 12-2
12-1-2 仕様 12-3
12-1-3 SNMP メッセージ 12-3
12-1-4 MIB の仕様 12-4
12-2 SNMP（エージェント）機能の使用手順 12-18
12-2-1 使用手順 12-18
12-2-2 SNMP（エージェント）機能に必要な設定一覧 12-18

第 13 章 通信性能と通信負荷

13-1 通信のしくみ 13-2
13-1-1 タグデータリンクの通信のしかた 13-2
13-1-2 コネクション数の計算方法 13-4
13-1-3 パケットインターバル（RPI）の精度 13-5
13-2 通信負荷の調整 13-6
13-2-1 タグデータリンクのデバイス帯域使用状況の確認 13-7
13-2-2 タグデータリンクの帯域使用率と RPI の関係 13-9
13-2-3 デバイス帯域使用状況の調整方法 13-10
13-2-4 RPI の変更方法 13-11
13-2-5 パケットインターバル（RPI）の設定例 13-16
13-3 タグデータリンクによる入出力応答時間 13-21
13-3-1 データ転送のタイミング 13-21
13-3-2 内蔵 EtherNet/IP ポートのデータ処理時間 13-22
13-3-3 タスク周期とパケットインターバル（RPI）の関係 13-23
13-3-4 タグデータリンク入出力最大応答時間 13-25
13-4 メッセージサービス機能伝送遅延時間 13-27

第 14 章 ネットワークの通信ステータス確認と処置方法

14-1 Network Configurator のデバイスモニタ機能 14-2
14-1-1 デバイスモニタ機能の起動方法 14-2
14-1-2 デバイスモニタ機能の各種ステータス表示 14-2
14-2 コネクションステータスのコード一覧と処置方法 14-9

付録

A-1 他シリーズとの EtherNet/IP ポートの機能比較 A-2

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-3
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-3
A-2-2 Sysmac Studio による EtherNet/IP コネクション設定の手順 A-4
A-2-3 EtherNet/IP コネクション設定機能 A-5
A-2-4 Sysmac Studio による EtherNet/IP コネクション設定 A-10
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-31
A-2-6 トラブルシューティング A-35

A-3 EDS ファイルの管理 A-38
A-3-1 EDS ファイルのインストール A-39
A-3-2 EDS ファイルの作成 A-39
A-3-3 EDS ファイルの削除 A-40
A-3-4 EDS ファイルの保存 A-40
A-3-5 EDS ファイルの検索 A-41
A-3-6 EDS ファイルのプロパティ表示 A-42
A-3-7 EDS インデックスファイルの再作成 A-42

A-4 Network Configurator を、Windows XP/Windows Vista/Windows 7 以降で使用する際の留意点 A-43
A-4-1 Windows ファイアウォールの設定の変更手順 A-43

A-5 変数のメモリ確保の方法 A-46
A-5-1 変数のメモリ確保の規則 A-46
A-5-2 留意するべきケース例 A-55

A-6 コントローラの外部出力にアクセスする場合の留意事項 A-58

A-7 TCP 状態遷移 A-59

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

安全上のご注意の内容については、以下のマニュアルを参照してください。

- 『NY シリーズ 産業用パネル型 PC ユーザーズマニュアル ハードウェア編（SBCA-435)』
- 『NY シリーズ 産業用ボックス型 PC ユーザーズマニュアル ハードウェア編（SBCA-434)』
- 『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編（SBCA-436)』

安全上の要点

安全上の要点の内容については、以下のマニュアルを参照してください。

- 『NY シリーズ 産業用パネル型 PC ユーザーズマニュアル ハードウェア編（SBCA-435)』
- 『NY シリーズ 産業用ボックス型 PC ユーザーズマニュアル ハードウェア編（SBCA-434)』
- 『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編（SBCA-436)』

使用上の注意

使用上の注意の内容については、以下のマニュアルを参照してください。

- 『NY シリーズ 産業用パネル型 PC ユーザーズマニュアル ハードウェア編（SBCA-435)』
- 『NY シリーズ 産業用ボックス型 PC ユーザーズマニュアル ハードウェア編（SBCA-434)』
- 『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編（SBCA-436)』

法規と規格

法規と規格の内容については、以下のマニュアルを参照してください。

- 『NY シリーズ 産業用パネル型 PC ユーザーズマニュアル ハードウェア編（SBCA-435）』
- 『NY シリーズ 産業用ボックス型 PC ユーザーズマニュアル ハードウェア編（SBCA-434）』

バージョン

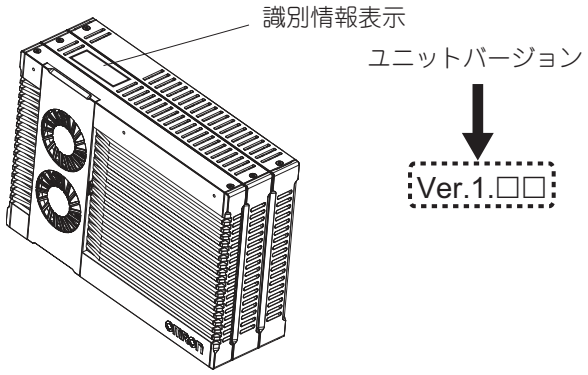
NY シリーズおよび各 EtherCAT スレーブは、ハードウェアやソフトウェアが、ハードウェアリビジョンやユニットバージョンという改変番号でバージョン管理されています。ハードウェアまたはソフトウェアに仕様変更があるごとに、ハードウェアリビジョンまたはユニットバージョンが更新されます。したがって、同じ形式のユニットや EtherCAT スレーブでも、ハードウェアリビジョンまたはユニットバージョンが異なると搭載機能や性能が異なります。

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

製品背面の識別情報表示で、バージョンが確認できます。
NY シリーズコントローラ形 NY5□2-□□□□ の識別情報表示は、下図のようになります。



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。

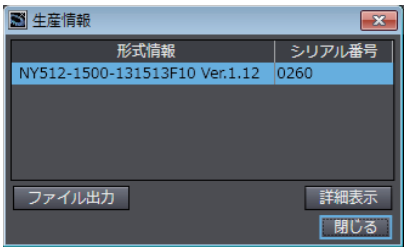
● NY シリーズコントローラでのバージョン確認方法

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、コントローラだけです。

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］–［CPU ラック］を右クリックし、［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、ハードウェアリビジョン、および各種の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

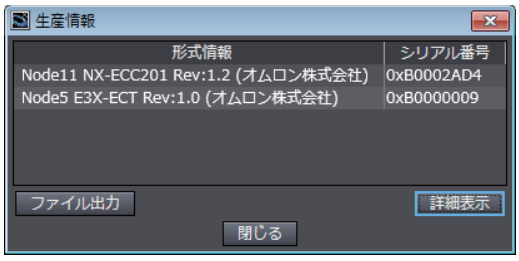
EtherCAT スレーブのバージョンは、オンライン状態の〔 生産情報 〕で確認することができます。確認方法は、以下のとおりです。

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。

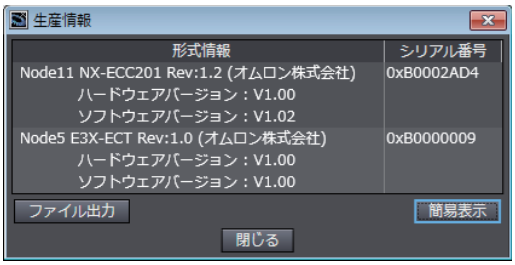
2 **EtherCAT 構成編集画面でマスタを右クリックして［生産情報表示］を選択します。**
生産情報ダイアログボックスが表示されます。
ユニットバージョンは、「Rev」を付けて表示されます。

● 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

ユニットバージョンと周辺ツールのバージョン

NY シリーズ コントローラ内蔵 Ethernet/IP ポートのタグデータリンク設定には、以下に記載したバージョンの Network Configurator および Sysmac Stuido を使用します。

コントローラ		Network Configurator for EtherNet/IP		Sysmac Studio	
形式	バージョン	Ver.3.59 以前	Ver.3.60	Ver.1.16 以前	Ver.1.17
形 NY5□2-□□□□	Ver.1.12 以降	×	○	×	○

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
NY シリーズ IPC マシンコントローラ 産業用パネル型 PC ユーザーズマニュアル ハードウェア編	SBCA-435	形 NY532-□□□□	NY シリーズ産業用パネル型 PC 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。	NY シリーズのシステム全体概要、および産業用パネル型 PC 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
NY シリーズ IPC マシンコントローラ 産業用ボックス型 PC ユーザーズマニュアル ハードウェア編	SBCA-434	形 NY512-□□□□	NY シリーズ産業用ボックス型 PC 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。	NY シリーズのシステム全体概要、および産業用ボックス型 PC 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
NY シリーズ IPC マシンコントローラ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル セットアップ編	SBCA-441	形 NY532-□□□□ 形 NY512-□□□□	NY シリーズ産業用 PC の初期設定やコントローラの使用準備について知りたいとき。	NY シリーズのシステム全体概要に関して、以下の内容を説明します。 • 2 つの OS のしくみ • 初期設定 • Industrial PC Support Utility • IPC Compolet • IPC .NET API • バックアップとリカバリ
NY シリーズ IPC マシンコントローラ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編	SBCA-436	形 NY532-□□□□ 形 NY512-□□□□	NY シリーズ産業用 PC のコントローラ機能のプログラミング／システムの立ち上げについて知りたいとき。	NY シリーズマシンオートメーションコントロールソフトウェアに関して、以下の内容を説明します。 • コントローラの動作 • コントローラの機能 • コントローラの設定 • IEC 61131-3ベースの言語仕様とプログラミング
NY シリーズ コマンドリファレンス マニュアル 基本編	SBCA-437	形 NY532-□□□□ 形 NY512-□□□□	NY シリーズ産業用 PC 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。
NY シリーズ IPC マシンコントローラ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル モーション制御編	SBCE-379	形 NY532-□□□□ 形 NY512-□□□□	NY シリーズ産業用 PC の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コントローラの設定や動作、プログラミングの考え方について説明します。
NY 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-380	形 NY532-□□□□ 形 NY512-□□□□	モーション命令仕様の詳細について知りたいとき。	各モーション制御命令の詳細を説明します。
NY シリーズ IPC マシンコントローラ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル 内蔵 EtherCAT® ポート編	SBCD-368	形 NY532-□□□□ 形 NY512-□□□□	NY シリーズ産業用 PC 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。

マニュアル名称	Man.No.	形式	用途	内容
NY シリーズ IPC マシンコントローラ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル 内蔵 EtherNet/IP™ ポート編	SBCD-369	形 NY532-□□□□ 形 NY512-□□□□	NY シリーズ産業用 PC 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。
NJ/NY シリーズ NC 統合コントローラ ユーザーズマニュアル	SBCE-428	形 NJ501-5300 形 NY532-5400	NJ/NY シリーズで数値制御を行うとき。	数値制御機能について説明します。プログラミング時に、コマンドリファレンスマニュアル G コード編 (SBCE-429) と併せて使用してください。
NJ/NY シリーズ コマンドリファレンス マニュアル G コード編	SBCE-429	形 NJ501-5300 形 NY532-5400	G コード／ M コード命令仕様の詳細について知りたいとき。	G コード／ M コード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 (SBCE-428) と併せて使用してください。
NY シリーズ トラブルシューティング マニュアル	SBCA-438	形 NY532-□□□□ 形 NY512-□□□□	NY シリーズ産業用 PC で検出する異常の詳細について知りたいとき。	NY 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。
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。
CNC Operator オペレーションマニュアル	SBCE-430	形 SYSMAC -RTNC0 □□□ D	CNC Operator の概要、使用方法について知りたいとき。	CNC Operator の概要、インストール方法、基本操作、接続操作、主な機能の操作方法などについて説明します。

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	2016 年 9 月	初版
B	2017 年 4 月	• コントローラ Ver1.14 以降のサポート機能に関する記述の追記 • 誤記修正
C	2017 年 10 月	誤記修正
D	2019 年 1 月	誤記修正
E	2019 年 7 月	• コントローラ Ver1.21 以降のサポート機能に関する記述の追記 • 誤記修正
F	2021 年 3 月	• コンフォーマンステストに関する記載の更新 • 接続可能な FTP サーバに Windows10/Windows Server2016（IIS10.0）を追加
G	2022 年 6 月	誤記修正
H	2024 年 3 月	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 NY シリーズ産業用 PC の内蔵 EtherNet/IP ポートの特長 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 内蔵 EtherNet/IP ポート 1-8
1-3-1 仕様 1-8
1-3-2 各部の名称と機能 1-10

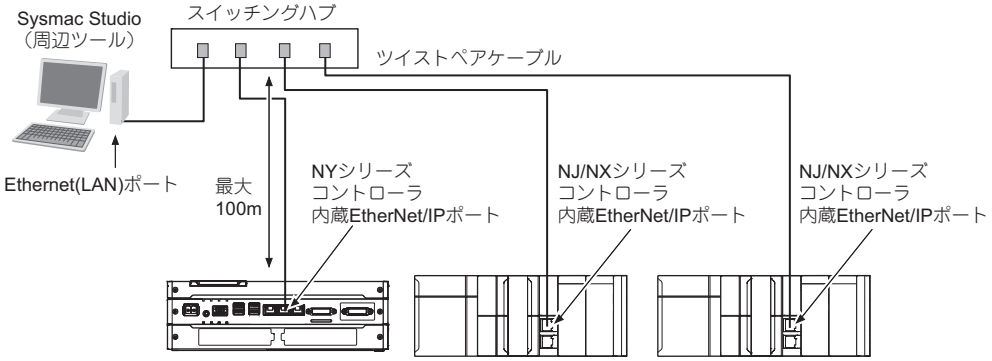
1-4 通信サービスの概要 1-12
1-4-1 Windows との IP 通信サービス 1-12
1-4-2 CIP 通信（Common Industrial Protocol）サービス 1-12
1-4-3 IP ルーティング機能 1-14
1-4-4 BOOTP クライアント機能 1-17
1-4-5 FTP サーバ機能 1-17
1-4-6 FTP クライアント機能 1-18
1-4-7 ソケットサービス機能 1-19
1-4-8 ホスト名指定機能 1-20
1-4-9 SNMP（エージェント）機能 1-20

1-5 EtherNet/IP 通信の使用手順 1-22

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　NY シリーズ産業用 PC の内蔵 EtherNet/IP ポートの特長

● 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、データを高速にデータ交換できます。

● CIP 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

NY シリーズの内蔵 EtherNet/IP ポートを 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。
複数の内蔵 EtherNet/IP ポートの 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。

● 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 クライアント機能）

FTP クライアント機能を内蔵しているため、コントローラからファイル単位で FTP サーバ機能を持つワークステーションやパソコン内のデータを読み出すことや、データを書き込むことができます。
FTP クライアント通信命令を用いて、FTP サーバとの間で複数のファイルを一度に転送できます。

● ソケットサービス機能

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。
プログラム上でソケット通信命令を順に実行することにより、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。

● ホスト名での指定 (DNS クライアント機能)

SNMP マネージャの IP アドレス、またはソケット命令や 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 クライアント機能または Hosts 設定)。
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内蔵 EtherNet/IP ポー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

● IP パケットを異なるネットワークセグメントにあるデバイスに転送が可能

内蔵 EtherNet/IP ポートには IP ルーティング機能を搭載しており、IP パケットを異なるネットワークセグメントにあるデバイスに転送することできます。
* ネットワーク構成に合わせて、ネットワーク上の各デバイスに IP ルータテーブルやデフォルトゲートウェイの設定が必要となります。設定項目の詳細は「5-1 ［TCP/IP 設定］ダイアログ」を参照してください。



参考

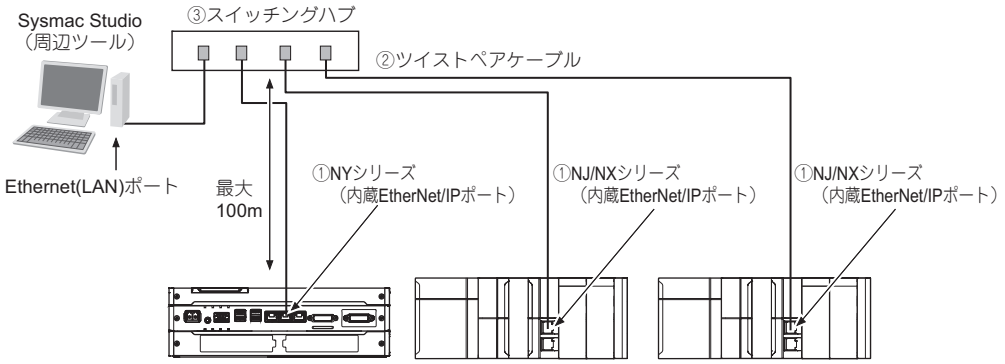
CIP（Common Industrial Protocol）とは
OSI アプリケーション層の産業用共通プロトコルです。EtherNet/IP、CompoNet、DeviceNet などで使用されています。
CIP をベースとしたネットワーク間では、ネットワーク間のルーチングを容易に行うことができます。そのため、フィールド機器から上位まで、透過的なネットワークを容易に構築できます。
CIP には、以下の特徴があります。

- 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 NY シリーズ (内蔵 EtherNet/IP ポート) （形 NY5□2-□□□□） NJ シリーズ（内蔵 EtherNet/IP ポート） （形 NJ501-□□□□/ 形 NJ301-□□□□/ 形 NJ101-□□□□） NX シリーズ（内蔵 EtherNet/IP ポート） （形 NX701-□□□□） その他オムロン製 コントローラ CJ2 （内蔵 EtherNet/IP ポート） （形 CJ2H-CPU□□ -EIP ／形 CJ2M-CPU3□） CJ シリーズ EtherNet/IP ユニット（*） （形 CJ1W-EIP21） CS シリーズ EtherNet/IP ユニット （形 CS1W-EIP21）	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、「2-1-1 ネットワーク構成機器の推奨品」を参照してください。

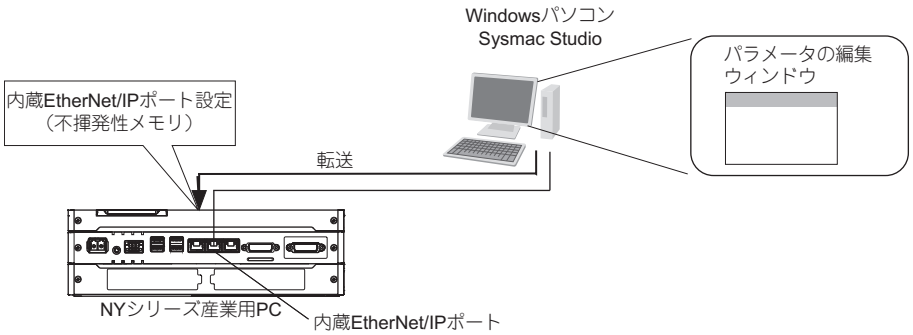
* NJ シリーズ CPU ユニットだけ装着可能です。ただし、CPU ユニットの Ver.1.01 以降、かつ Sysmac Studio の Ver1.02 以降で使用可能です。

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。
内蔵 EtherNet/IP ポートには、大きく分けて「Ethernet 設定」と「タグデータリンク設定」があり、いずれもコントローラ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。

● 内蔵 EtherNet/IP 設定：Sysmac Studio

内蔵 EtherNet/IP ポートの自 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。



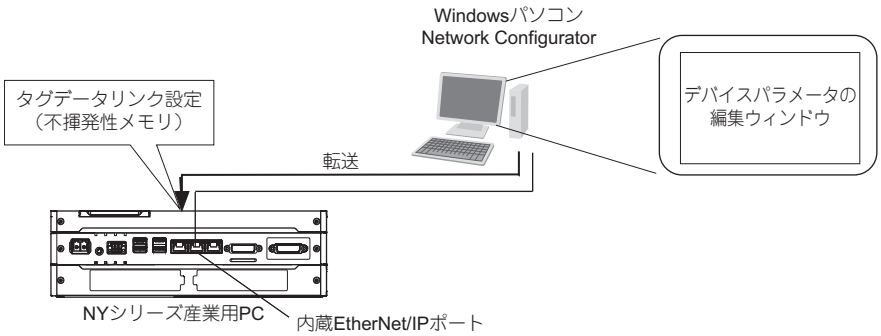
Sysmac Studio に関する詳細は 🕮『Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)』を参照してください。

● タグデータリンク設定：Network Configurator

内蔵 EtherNet/IP ポー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　内蔵 EtherNet/IP ポート

1-3-1　仕様

項　目		仕様 形 NY5□2-□□□□ Ver.1.12 以降
通信プロトコル		TCP/IP、UDP/IP
サポートサービス		Sysmac Studio 接続、タグデータリンク、CIP メッセージ通信、ソケットサービス、FTP サーバ、FTP 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）
ポート数		1（IP ルーティング機能付き）
物理層		100BASE-TX/10BASE-T/1000BASE-T （ただし、1000BASE-T、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	1000M ビット /s（1000BASE-T）
	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	128
	パケットインターバル（更新周期）	1 ～ 10000ms（1ms 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	20000pps*2
		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 は使用不可
	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	184832 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」を参照してください。
	登録可能なタグセット数	128（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
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：64（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	可 CIP ルーチング可能な相手ユニットは以下のとおり 形 NY5□2-□□□□ 形 NX701-□□□□ 形 NJ501-□□□□、形 NJ301-□□□□、形 NJ101-□□□□、 形 CS1W-EIP21、形 CJ1W-EIP21、 形 CJ2H-CPU□□-EIP、形 CJ2M-CPU3□ 上記いずれのユニットの組合せでも、最大 8 階層越えが可能。
SNMP	エージェント	SNMPv1、SNMPv2c
	MIB	MIB-II

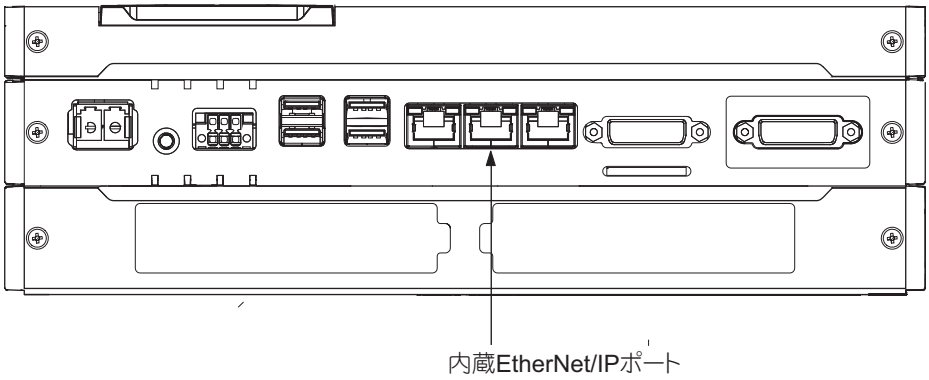
項　目	仕様 形 NY5□2-□□□□ Ver.1.12 以降
EtherNet/IP コンフォーマンステスト	CT16 準拠
Ethernet インタフェース	10BASE-T/100BASE-TX /1000BASE-T Auto Negotiation ／固定設定

*1. タグデータリンクをご使用の場合は、100BASE-TX、1000BASE-T 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/NX/NY 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、変更対象の内蔵 EtherNet/IP ポー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. 内蔵 EtherNet/IP ポートは、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。
*6. 内蔵 EtherNet/IP ポートは、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
タグデータリンク	UDP	2222	固定値
システムで使用	UDP	2223、2224	
	TCP	9610、9900	
CIP メッセージ	TCP	44818	
FTP クライアント（データ転送ポート）	TCP	20	
DNS クライアント	TCP/UDP	53	
BOOTP クライアント	UDP	68	
HTTP サーバ	TCP	80	
システム使用、その他	TCP/UDP	9600	Sysmac Studio のユニット設定で変更可能
FTP クライアント（制御ポート）	TCP	21	
SNMP エージェント	UDP	161	
SNMP トラップ	UDP	162	

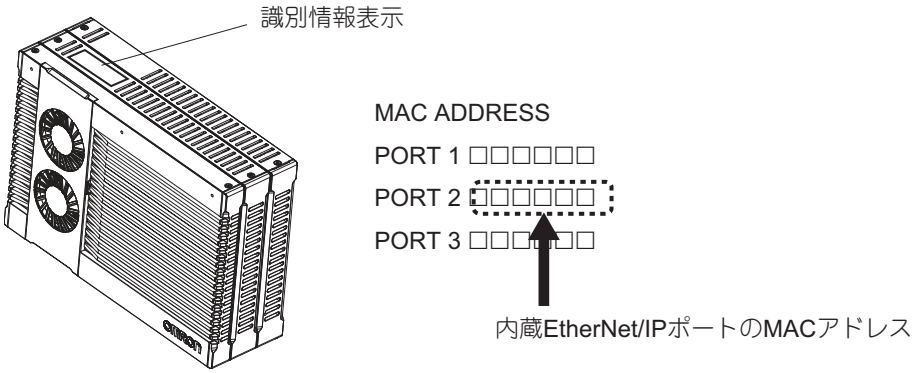
1-3-2　各部の名称と機能

各部の名称



MAC アドレスの表記

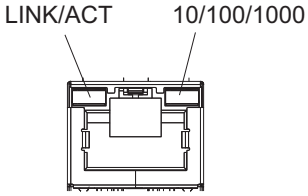
Ethernet に接続されるすべての通信機器には、固有の MAC アドレスが割り当てられています。下図のように製品背面の識別情報表示に、内蔵 EtherNet/IP ポートの MAC アドレスが 6 桁の 16 進数で表記されています。



表示部（LED）

NY シリーズの内蔵 EtherNet/IP ポートに関係する LED について説明します。

● LINK/ACT ／ 10/100/1000



・［LINK/ACT］LED：　Ethernet 通信の送受信状態を表示します。
・［10/100/1000］LED :伝送速度の状態を表示します。

表示	色	状態	動作状態
LINK/ACT	--	消灯	リンクが確立していません。 • ケーブル未接続の状態 • 電源 OFF 状態、または、リセット状態
	黄	点滅	リンクが確立し、データを送受信しています。
		点灯	リンクが確立しています。
10/100/1000	--	消灯	伝送速度が 10M ビット /s でリンクが確立しています。または、リンクが確立していません。
	緑	点灯	伝送速度が 100M ビット /s でリンクが確立しています。
	橙	点灯	伝送速度が 1000M ビット /s でリンクが確立しています。

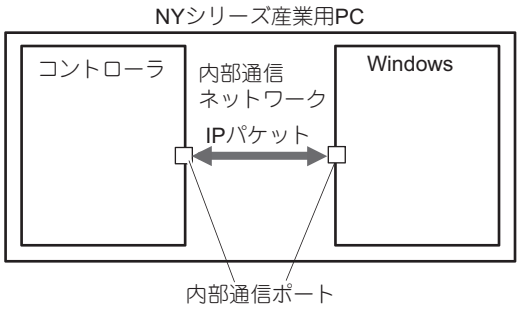
参考

内蔵 EtherNet/IP ポートの設定については、🕮「5-1 ［TCP/IP 設定］ダイアログ」を参照してください。

1-4　通信サービスの概要

1-4-1　Windows との IP 通信サービス

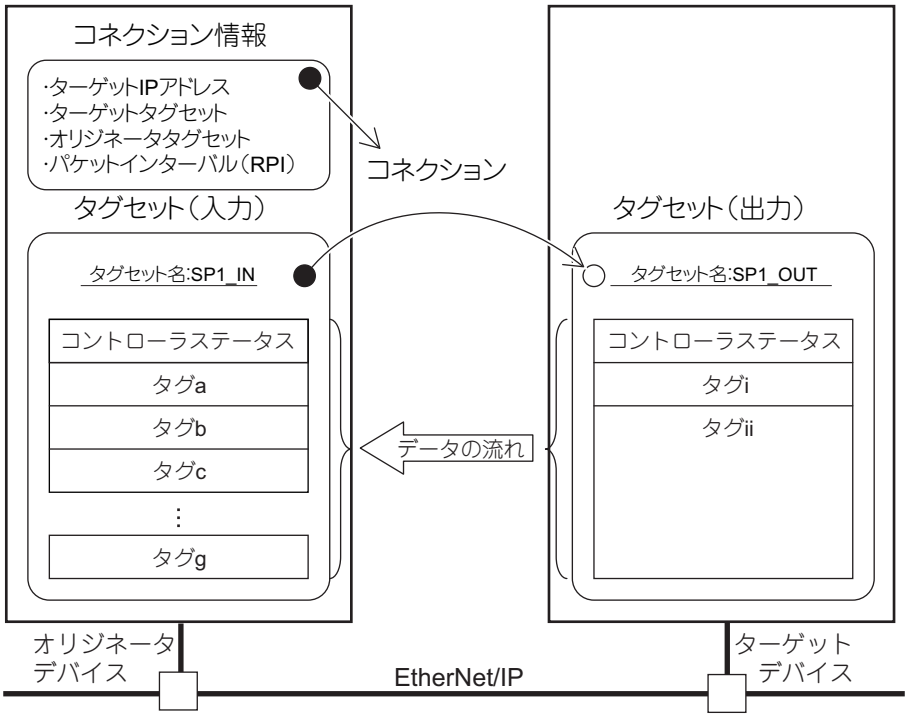
産業用 PC のコントローラと Windows はそれぞれ内部通信ポートを持っており、内部通信ポートを通じ、内部通信ネットワークを介して IP 通信を行い、IP パケットでやりとりすることができます。
Windows とコントローラのやりとりに関する詳細は、🕮『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル セットアップ編（SBCA-441)』を参照してください。



1-4-2　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）

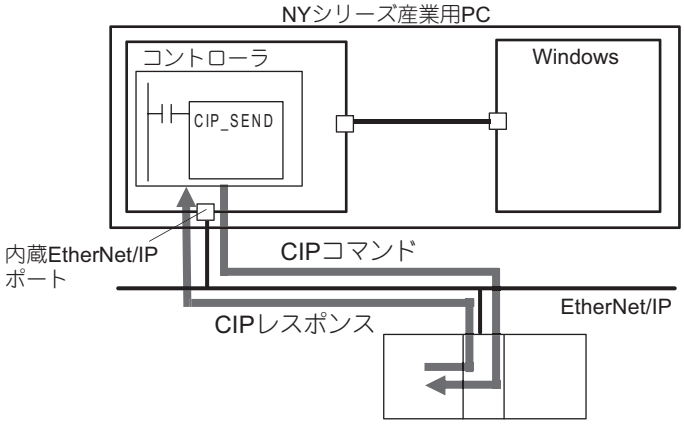
EtherNet/IP ネットワーク上のデバイスと、プログラムレスでサイクリックなデータ交換を行うことができます。
内蔵 EtherNet/IP ポートにおけるタグデータリンクでは、通常、Network Configurator で作成したタグをグループ化（タグセット）し、そのグループ単位でターゲットデバイスとのコネクションを開設し通信を行います。つまり、1 グループ(1 タグセット)で 1 コネクションを使用することになります。



※ 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

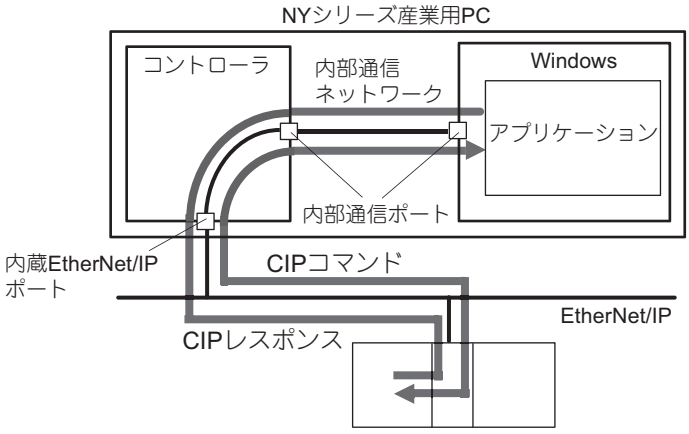
EtherNet/IP ネットワーク上のデバイスに対して、任意の CIP コマンドを送信することができます。NY シリーズコントローラのプログラム上で CIP 通信命令を実行することにより、データ読み書きなどの各種 CIP コマンドを発行し、レスポンスを受信します。



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、内蔵 EtherNet/IP ポートを経由して、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/NX/NY 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。ただし、IP ルーティングのスイッチングハブ（L3）による階層越えは、8 階層には含まれません。

また、NY シリーズ産業用 PC の Windows と EtherNet/IP ネットワーク内のデバイスの間で、コントローラの内部通信ポートおよび内蔵 EtherNet/IP ポートを経由し、CIP メッセージをルーチングすることが可能です。



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。

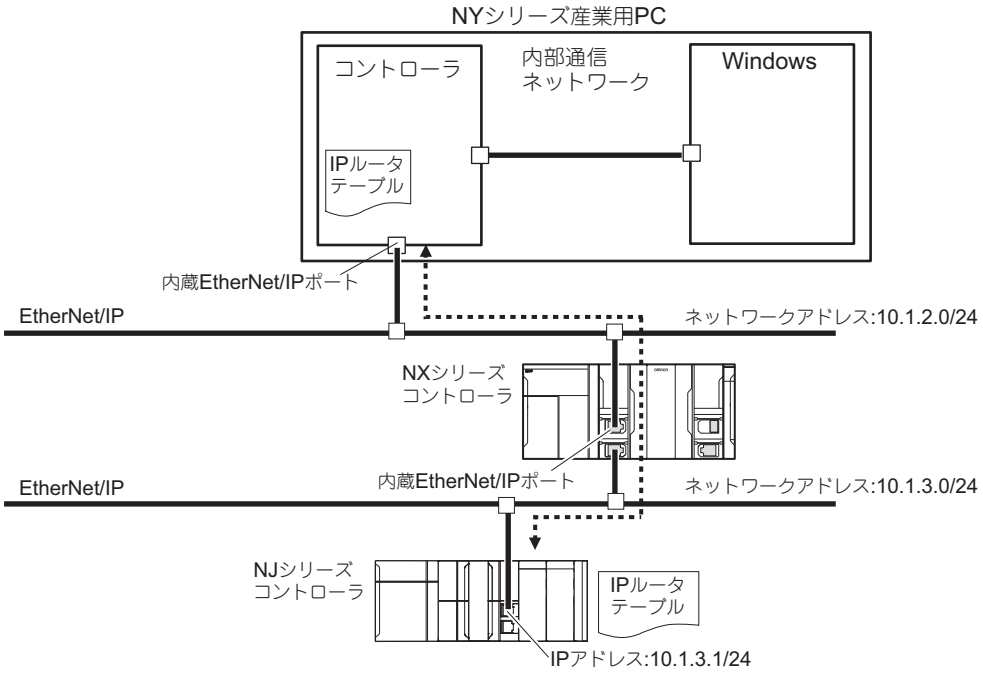
1-4-3　IP ルーティング機能

NY シリーズ産業用 PC では内蔵 EtherNet/IP ポートに IP ルーティング機能があります。IP ルーティング機能では、IP ルータテーブルに設定された経路情報に沿って、IP パケットを他のネットワークセグメントに送信します。
ネットワーク構成に合わせて、コントローラおよびネットワーク上の各デバイスに IP ルータテーブルやデフォルトゲートウェイを設定することにより、異なるネットワークセグメント上のデバイス間の通信が可能です。

使用上の注意

NY シリーズコントローラによる IP ルーティングを介した EtherNet/IP デバイス間のタグデータリンクは行えません。

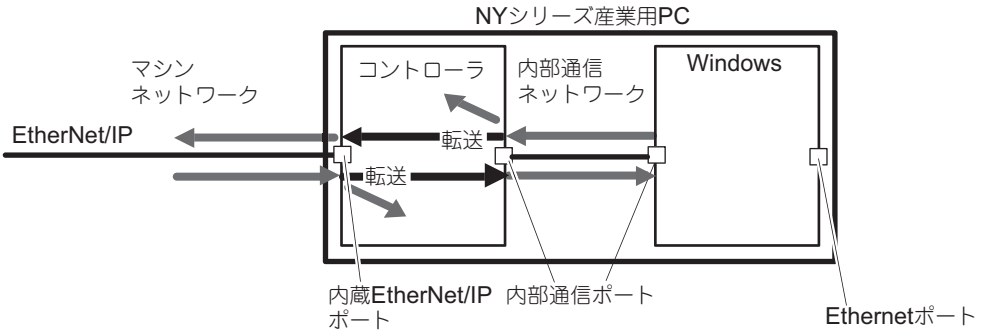
下図は、NY シリーズ産業用 PC のコントローラと異なるネットワークセグメント上の NJ シリーズコントローラと通信する例です。NY シリーズ産業用 PC のコントローラの IP ルータテーブルとデフォルトゲートウェイには、NX シリーズコントローラの内蔵 EtherNet/IP ポートを経由する経路情報の設定が必要です。



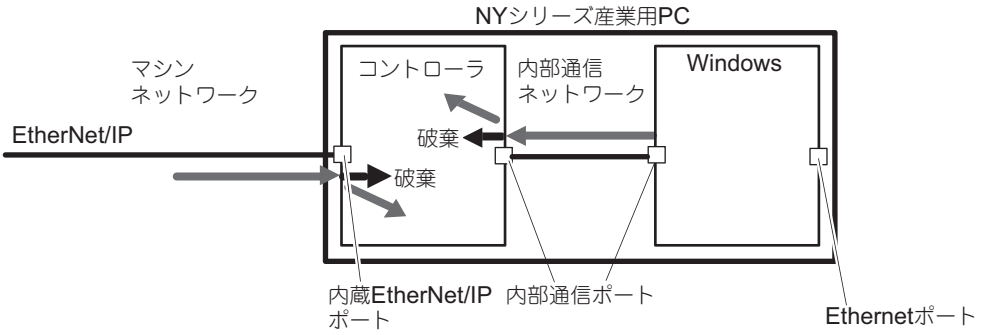
NY シリーズコントローラには、IP ルーティング機能に関する次の機能があります。
それぞれの機能の設定については、🕮「5-1 ［TCP/IP 設定］ダイアログ」を参照してください。

● IP Forward 機能

ネットワークから受信した IP パケットを別のネットワークに転送する機能です。
NY シリーズ産業用 PC のコントローラでは、IP パケットを内部通信ネットワークとマシンネットワーク間で中継することができます。異なるネットワーク間を中継するためにコントローラの内部通信ポートと内蔵 EtherNet/IP ポート間で IP パケットが転送されます。

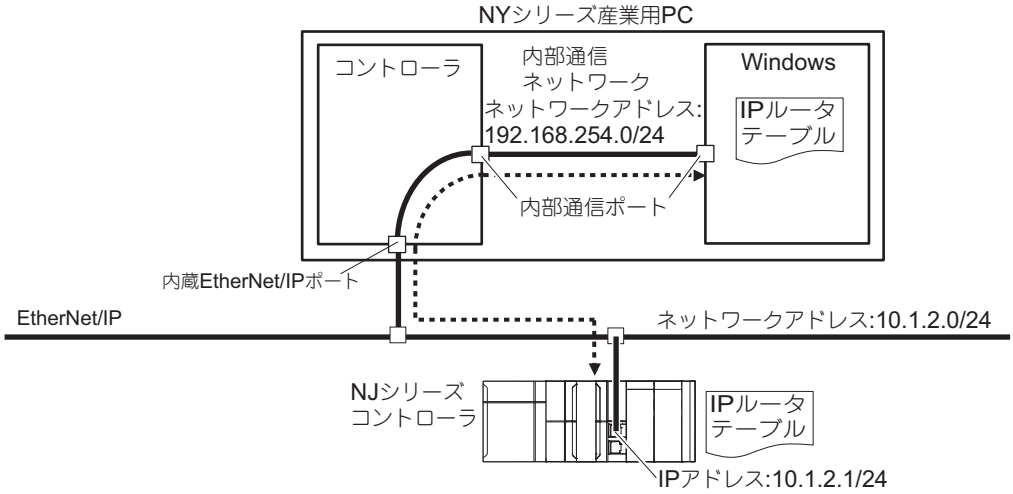


セキュリティ上、マシンネットワークと内部通信ネットワークを分断する必要がある場合、IP パケットを転送しない設定が可能です。IP パケットを転送しない設定の場合、コントローラあて以外の IP パケットは破棄されます。



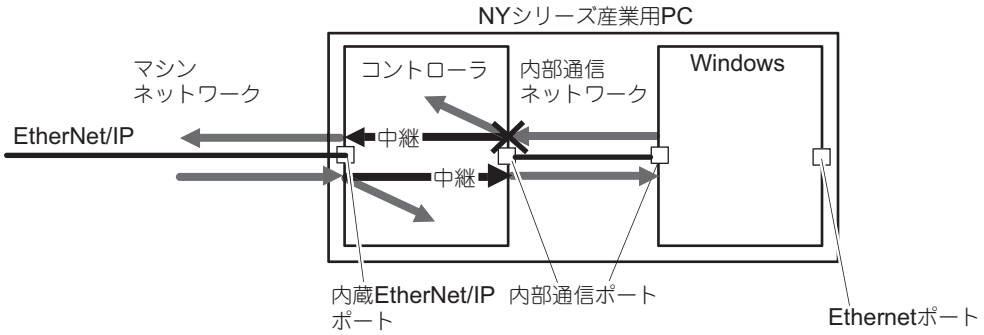
参考

NY シリーズ産業用 PC の Windows と異なるネットワークセグメント上の EtherNet/IP デバイスの間で IP ルーティングすることが可能です。
この場合、Windows に IP ルーティングする設定やデフォルトゲートウェイの設定が必要です。



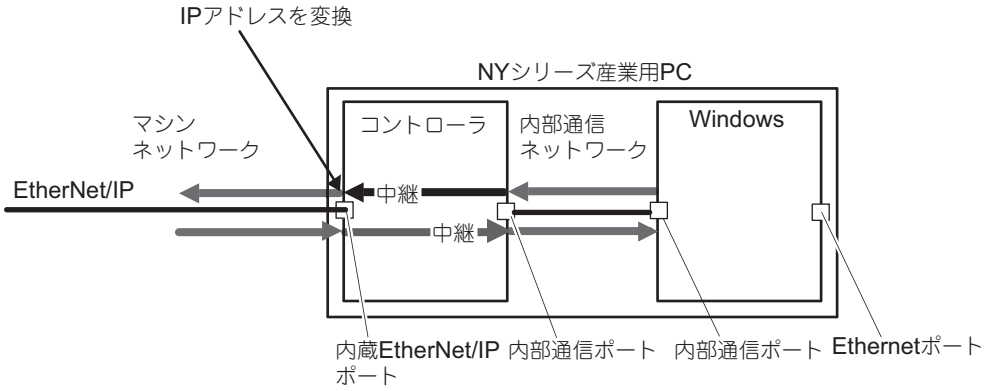
● PacketFilter 機能

ネットワークから受信した IP パケットを検査して、受信または別のネットワークに中継するか判断する機能です。
NY シリーズ産業用 PC のコントローラでは、内部通信ポートを経由した、コントローラを含むマシンネットワークへのアクセスを、あらかじめユーザが設定した条件で制限することができます。
コントローラの内部通信ポートで内部通信ネットワークから受信した IP パケットに対して検査されます。内部通信ポートの通過を許可するかをユーザが設定した条件で判断されます。
デフォルトでは、IP パケット内の送信元 IP アドレスが内部通信ポートの IP アドレスのネットワークだけ受信または別のネットワークに中継されます。



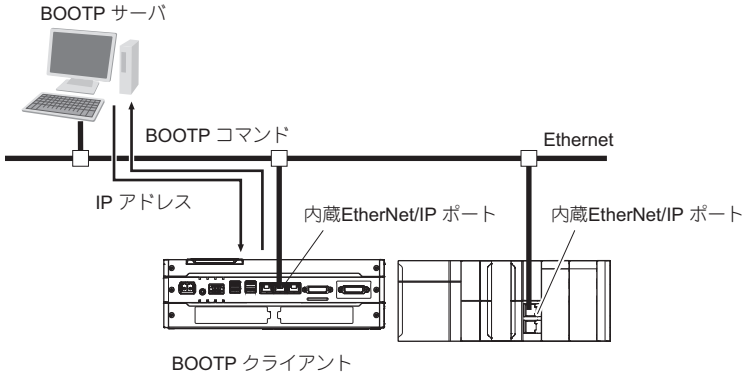
● NAT 機能

2 つのネットワーク間で IP パケットを中継する場合に、IP パケット内の送信元 IP アドレスを別の IP アドレスに変換する機能です。
NAT 機能を使用すると、自動的に内蔵 EtherNet/IP ポートで IP パケット内の送信元 IP アドレスを内蔵 EtherNet/IP ポートの IP アドレスに変換されます。
IP パケット内の送信元 IP アドレスを変換しないように NAT 機能を使用しない設定が可能です。



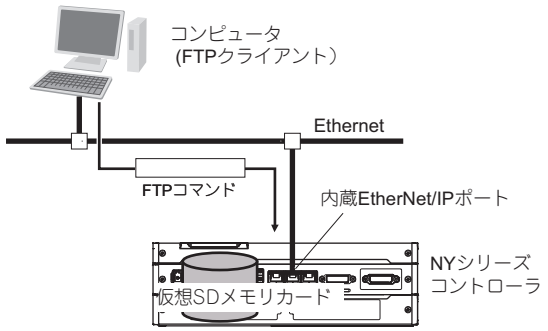
1-4-4　BOOTP クライアント機能

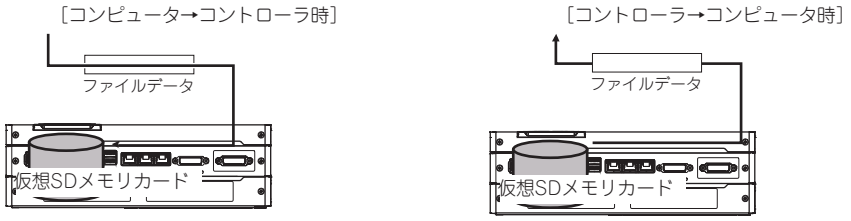
内蔵 EtherNet/IP ポートを BOOTP 設定にしておくことにより、内蔵 EtherNet/IP ポートの IP アドレスなどの設定を、BOOTP サーバから取得する機能です。



1-4-5　FTP サーバ機能

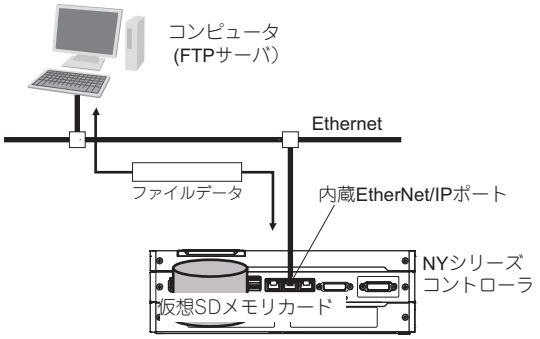
内蔵 EtherNet/IP ポートは FTP サーバ機能を内蔵しているため、Ethernet 上の他のコンピュータからコントローラの仮想 SD メモリカード内のファイルを読み書きできます。
これにより、コンピューターコントローラ間で、コンピュータが FTP クライアント、コントローラが FTP サーバとなった、ファイル単位のデータ交換が可能です。

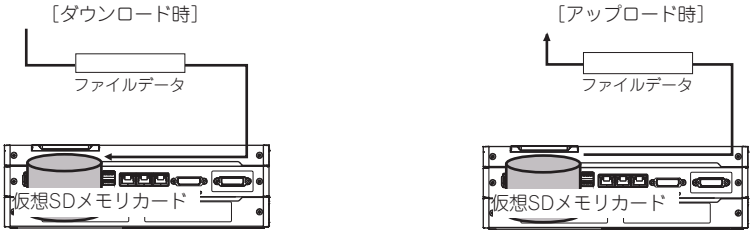




1-4-6　FTP クライアント機能

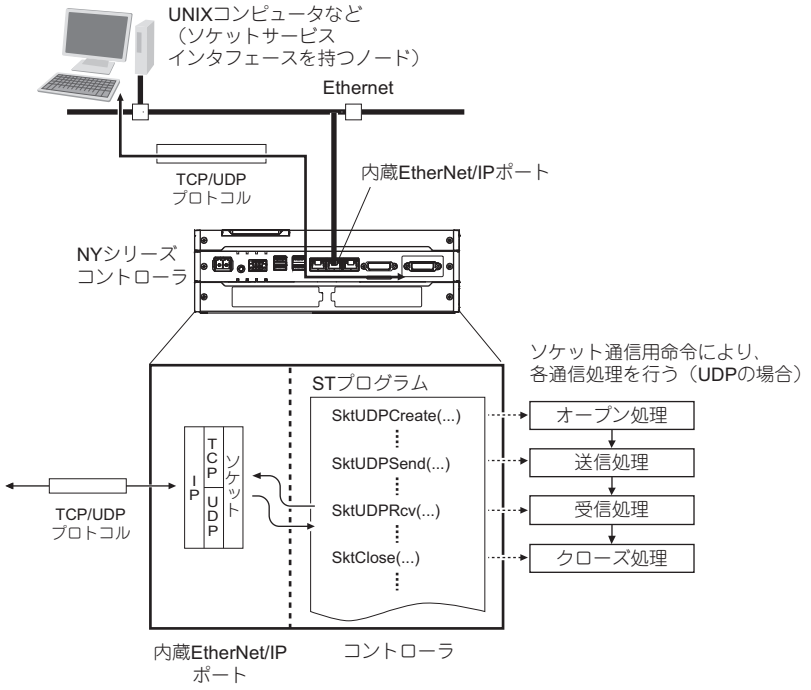
内蔵 EtherNet/IP ポートは FTP クライアント機能を内蔵しています。そのため、FTP クライアント通信命令を用いて、コントローラと Ethernet 上のコンピュータとの間でファイルを転送することができます。
これにより、コンピューターコントローラ間で、コントローラが FTP クライアント、コンピュータが FTP サーバとなった、ファイル単位のデータ交換が可能です。





1-4-7　ソケットサービス機能

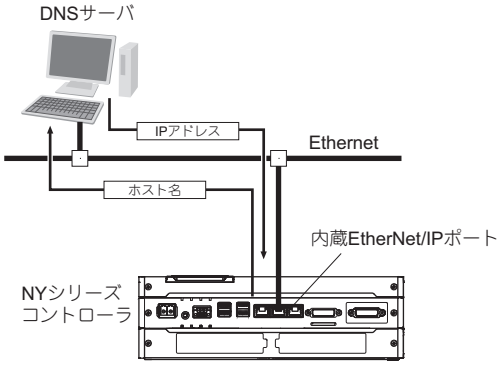
UDP または TCP プロトコルで、Ethernet 上の任意のノードと任意のデータの送受信が可能です。
ソケットサービスでのデータ送受信は、複数のソケット通信用命令を ST プログラム上で順に実行し、各通信処理を実行します。
オープン命令により通信相手機器とコネクションを開設した後、送信命令で指定した変数の値を送信し、受信命令で受信したデータを指定した変数に格納します。
クローズ命令によりコネクションを閉じて、通信を終了します。
また、TCP の場合はソケットの状態や受信データなどを読み出すこともできます。
TCP と UDP 合わせて計 30 ポートを使用できます。



1-4-8　ホスト名指定機能

SNMP マネージャの IP アドレス、またはソケット命令や 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 クライアント機能または Hosts 設定）。

例：DNS サーバでホスト名を設定する



使用上の注意

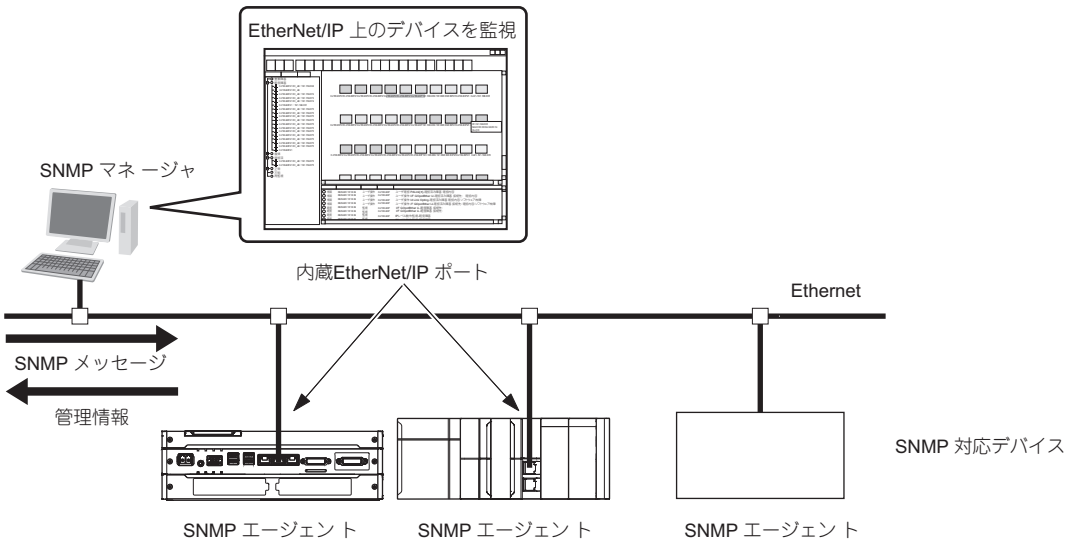
DNS クライアント機能による、サーバホスト名指定機能を利用するためには、DNS サーバが必要です。

1-4-9　SNMP（エージェント）機能

SNMP（エージェント）機能には以下のものがあります。

SNMP エージェント機能

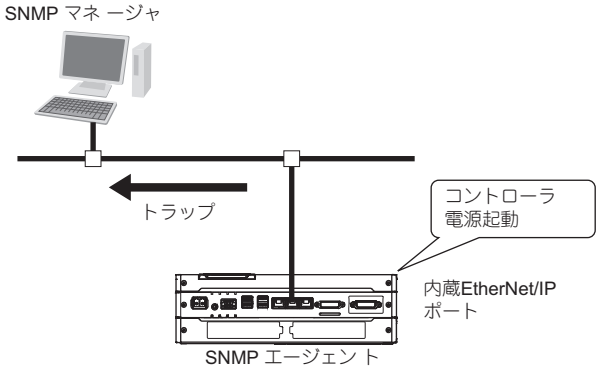
SNMP マネージャを使用したネットワーク管理ソフトウェアに、内蔵 EtherNet/IP ポートの内部状態の情報を提供します。



SNMP トラップ機能

特定の状況発生時、SNMP エージェントに設定した内蔵 EtherNet/IP ポートが、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，定期的に内蔵 EtherNet/IP ポー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



1-5　EtherNet/IP 通信の使用手順

● 基本操作

1　Ethernet ネットワーク（ツイストペアケーブル）の配線　　第 2 章 Ethernet ネットワークの配線

↓

2　Sysmac Studio で、内蔵 EtherNet/IP ポートの IP アドレスを設定　　第 4 章 IP アドレスを決定する方法

(1)　Sysmac Studio でプロジェクトを新規作成。

(2)　自 IP アドレスを以下から設定。

•デフォルト：

内蔵 EtherNet/IP ポート	: 192.168.250.1 （サブネットマスク= 255.255.255.0）
内部通信ポート	: 192.168.254.1 （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

↓

3　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

↓

4　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 内蔵 EtherNet/IP ポート設定一覧

•必要に応じて、TCP/IP 設定、Ethernet 設定などを任意に設定する。

● タグデータリンクを使用時

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

↓

4 タグデータリンク設定のダウンロード

↓

5 タグデータリンクの起動（電源 ON 時、自動で開始する）

↓

6 動作の確認

1-3-2「各部の名称と機能」
第 14 章 ネットワークの通信ステータス確認と処置方法

•内蔵 EtherNet/IP ポートの 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

● CIP メッセージ通信サービスを使用時

1 ユーザプログラム上で、CIP 通信命令を実行

第 8 章 CIP 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 14 章 ネットワークの通信ステータス確認と処置方法

•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

● ソケットサービスを使用時

1 ユーザプログラム上で、ソケットサービス命令の実行

第 9 章 ソケットサービス機能

↓

2 動作の確認

•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 10 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(NY シリーズコントローラ）への接続

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

使用上の注意

FTP サーバ機能を使用する場合、あらかじめ仮想 SD メモリカードの設定をする必要があります。仮想 SD メモリカードの設定方法は、🕮『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』を参照してください。

● FTP クライアント機能を使用時

1　ユーザプログラム上で、FTP クライアント通信命令の実行

第 11 章 FTP クライアント機能

↓

2　動作の確認

•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

使用上の注意

FTP クライアント機能を使用する場合、あらかじめ仮想 SD メモリカードの設定をする必要があります。仮想 SD メモリカードの設定方法は、🕮『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』を参照してください。

● 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 12 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 内蔵 EtherNet/IP ポート設定一覧

•BOOTP 設定

↓

2　動作の確認

•BOOTP 起動のイベントログを確認
•システム定義変数のオンラインを確認

2

Ethernet ネットワークの配線

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

2-2 ネットワークの布設 2-6
2-2-1 ネットワーク布設時の基本的注意事項 2-6
2-2-2 ネットワーク構成機器の推奨品 2-6
2-2-3 ツイストペアケーブル布設時の注意事項 2-6
2-2-4 スイッチングハブの設置・接続時の注意事項 2-10

2-3 ネットワークへの接続 2-11
2-3-1 Ethernet 接続コネクタ 2-11
2-3-2 ケーブルの接続方法 2-12

2-1　ネットワーク構成機器の選定

2-1-1　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	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） 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。
100BASE-TX/10BASE-T を使用する場合は、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。
1000BASE-T を使用する場合は、カテゴリ 5e 以上の STP ケーブル（アルミテープと編組の二重遮へい）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
1000BASE-T/100BASE-TX 用	サイズ・線心数（対数）：AWG24 × 4P *1	ケーブル	日立金属株式会社	NETSTAR-C5E SAB 0.5 × 4P CP
			倉茂電工株式会社	KETH-SB
			昭和電線ケーブルシステム株式会社	FAE-5004
			JMACS 株式会社	IETP-SB
		RJ45 コネクタ	パンドウイットコーポレーション	MPS588
100BASE-TX 用	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
			JMACS 株式会社	PNET/B
			昭和電線ケーブルシステム株式会社	FAE-5002
		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 *1	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。

2-1-2　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。

2-1-3　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。内蔵 EtherNet/IP ポート使用時は、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（オムロン製の内蔵 EtherNet/IP ポートは、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。

2-1-4　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🕮「13-2 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。

ネットワーク上で実行する通信速度に着目して選定するとき

● 100M ビット /s を超える伝送速度で、タグデータリンクをする場合

以下の条件で、タグデータリンクを行う場合は、マルチキャストフィルタ付き、または 1000M ビット /s の伝送速度に対応しているスイッチングハブを使用してください。
・マルチキャストを使用する
・100M ビット / s を超える伝送速度で通信する
もし、同一ネットワーク上に 100M ビット /s 以下で通信している Ethernet 機器があった場合、その Ethernet 機器がタグデータリンクと関係がなくても、タグデータリンク通信に影響を及ぼし、タグデータリンクが切断される可能性があります。

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。

2-2　ネットワークの布設

2-2-1　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、🕮『NY シリーズ 産業用ボックス型 PC ユーザーズマニュアル ハードウェア編（SBCA-434)』、または『NY シリーズ 産業用パネル型 PC ユーザーズマニュアル ハードウェア編（SBCA-435)』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。

2-2-2　ネットワーク構成機器の推奨品

内蔵 EtherNet/IP ポートを使用する場合のネットワーク構成機器については、🕮「2-1 ネットワーク構成機器の選定」を参照してください。

2-2-3　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**

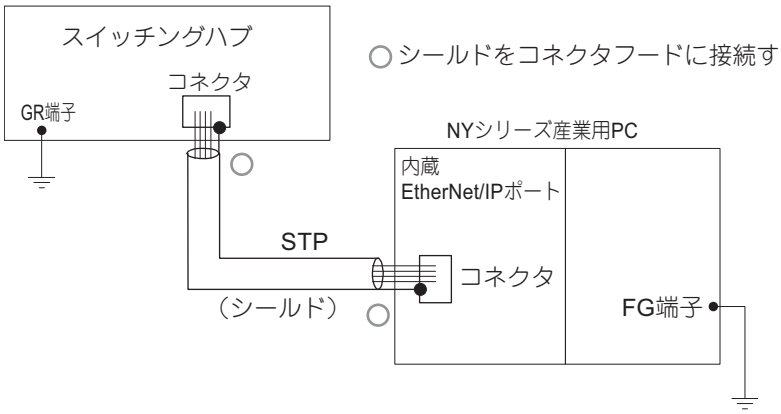
シールドのコネクタフードへの接続処理は、以下のようにしてください。

10BASE-T	100BASE-TX	1000BASE-T
• 両端接続 または • スイッチングハブ側のみ接続およびクランプコアをケーブルの EtherNet/IP ポート側に装着が必要		両端接続

- 10BASE-T または 100BASE-TX の場合
  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。

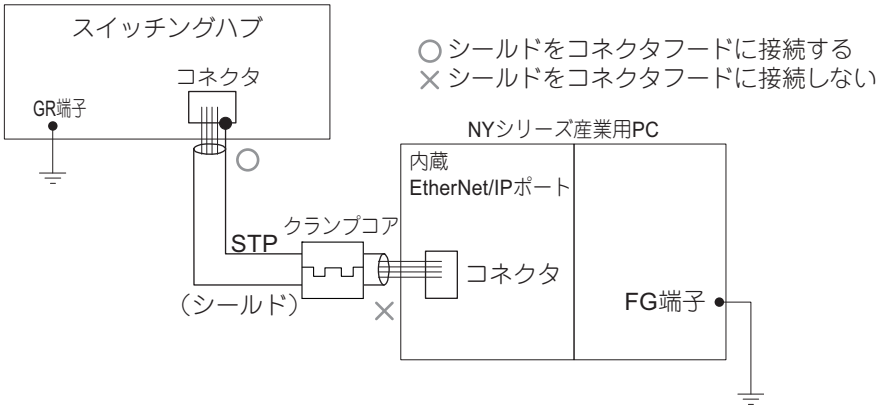
(a) 両端接続する

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

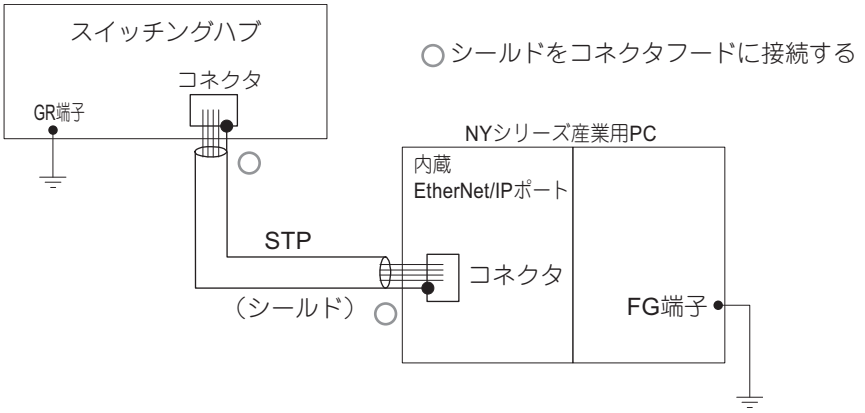
ケーブルの片端のスイッチングハブ側だけ、シールドをコネクタフードへ接続します。内蔵 EtherNet/IP ポート側のケーブルの付け根の位置に、クランプコアを装着してください。推奨するクランプコアおよびその装着方法は🕮「 推奨のクランプコアと装着方法」を参照してください。なお、EMC 規格に適合させる場合で、スイッチングハブ側だけでシールドをコネクタフードに接続するときは、クランプコアを装着することが必須です。



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。
伝送速度が 100Mbps 以下のときは、（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。

- 1000BASE-T の場合
  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、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てしてください。
  なお、1000BASE-T の場合、EMC 規格適合のために、この両端接続が必須です。



● 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内蔵EtherNet/IPポー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。

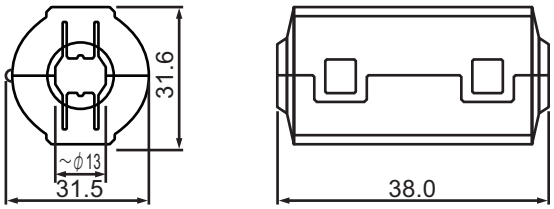
推奨のクランプコアと装着方法

スイッチングハブ側だけ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、内蔵 EtherNet/IP ポート側にクランプコアを装着する必要があります。
推奨するクランプコアとその装着方法は次のとおりです。

● 推奨クランプコア

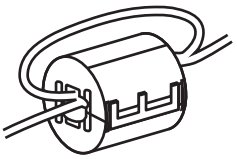
メーカ	品名	形式
NEC トーキン	クランプコア	ESD-SR-250

ESD-SR-250 の外形寸法



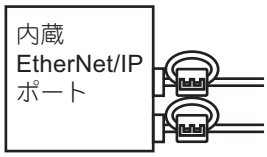
● 推奨装着方法

（1）クランプコアの通信ケーブルへの取り付け



図のように通信ケーブルを2ターンしてください。

（2）通信ケーブルの装着方法



図のように通信ケーブルの付け根に装着してください。

2-2-4　スイッチングハブの設置・接続時の注意事項

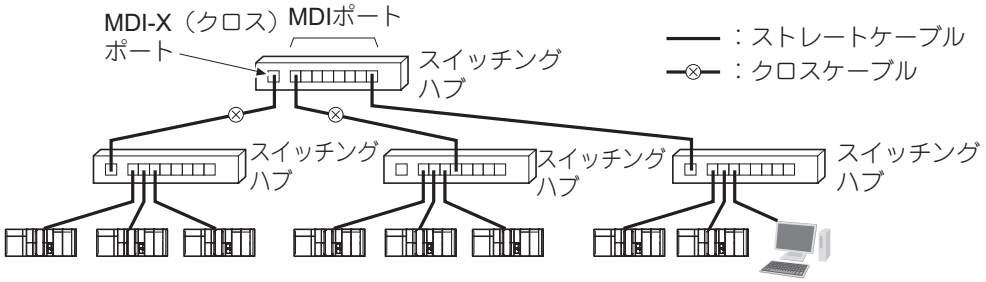
スイッチングハブ設置時の注意点

- 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

内蔵 EtherNet/IP ポートのリンク設定は、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		内蔵 EtherNet/IP ポート					
		AUTO-Nego	10M 固定		100M 固定		1000M 固定
			FULL	HALF	FULL	HALF	FULL
AUTO-Nego		◎	×	○	×	○	×
10M 固定	FULL	×	○	×	×	×	×
	HALF	○	×	○	×	×	×
100M 固定	FULL	×	×	×	○	×	×
	HALF	○	×	×	×	○	×
1000M 固定	FULL	×	×	×	×	×	◎

（◎：推奨、○：接続可能、×：接続不能）

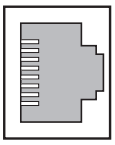
2-3　ネットワークへの接続

2-3-1　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。

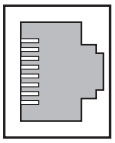
- 電気的特性：IEEE802.3 規格に準拠
- コネクタ機構：RJ45 の 8 ピン・モジュラ・コネクタ（ISO 8877 準拠）
- シールド線とコネクタフードの接続は、「2-1-2 スイッチングハブの種類」を参照してください。

10BASE-T/100BASE-TX の場合



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	－	－

1000BASE-T の場合



ピン番号	信号名称	略称	信号方向
1	送受信データ DA ＋	BI_DA ＋	入出力
2	送受信データ DA －	BI_DA －	入出力
3	送受信データ DB ＋	BI_DB ＋	入出力
4	送受信データ DC ＋	BI_DC ＋	入出力
5	送受信データ DC －	BI_DC －	入出力
6	送受信データ DB －	BI_DB －	入出力
7	送受信データ DD ＋	BI_DD ＋	入出力
8	送受信データ DD －	BI_DD －	入出力

2-3-2　ケーブルの接続方法

使用上の注意

- Ethernet通信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。
- Ethernet 通信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。
  コントローラに通信ケーブルの接続時の外形寸法は、🕮『NY シリーズ 産業用ボックス型 PC ユーザーズマニュアル ハードウェア編（SBCA-434)』、または『NY シリーズ 産業用パネル型 PC ユーザーズマニュアル ハードウェア編（SBCA-435)』を参照してください。ただし、使用する通信ケーブル、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先に確認してください。

(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内蔵 EtherNet/IP ポー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。

3

内蔵 EtherNet/IP ポート関連のシステム定義変数

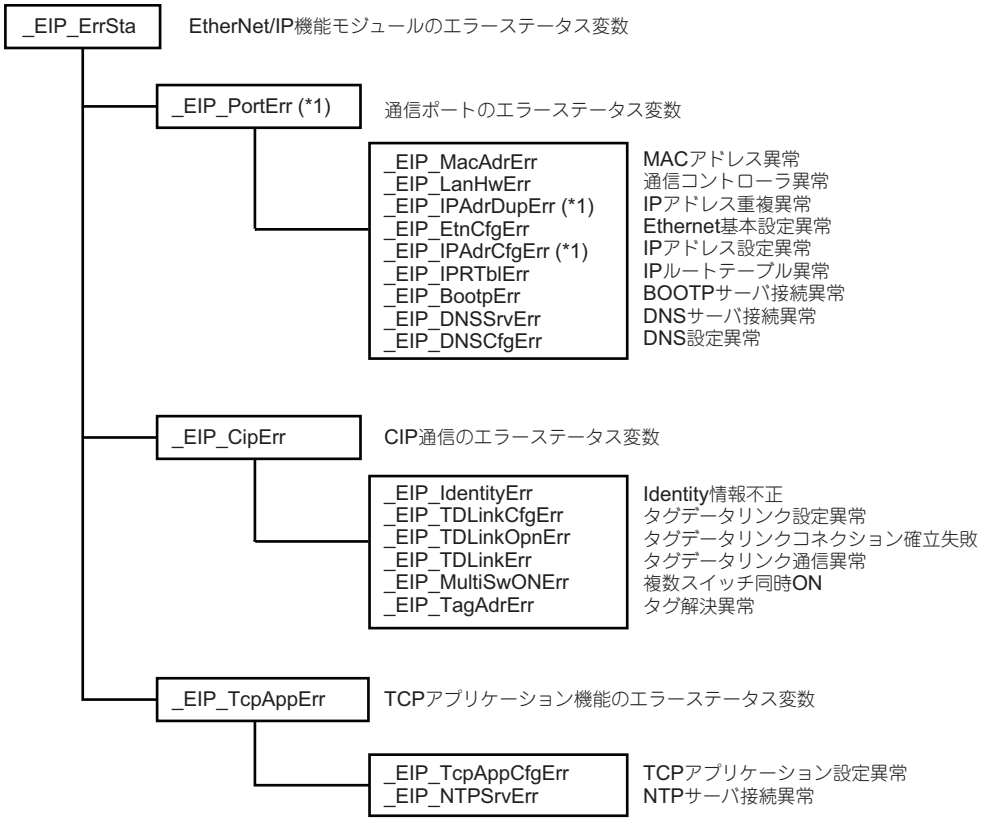
3-1 内蔵 EtherNet/IP ポートのシステム定義変数 3-2
3-2 システム定義変数一覧 3-3
3-3 システム定義変数の個別仕様 3-13

3-1　内蔵 EtherNet/IP ポートのシステム定義変数

内蔵 EtherNet/IP ポート用に用意されているシステム定義変数を使うと、内蔵 EtherNet/IP ポートのステータスをプログラム上で確認できます。

● 内蔵 EtherNet/IP ポート異常の確認

内蔵EtherNet/IPポートの異常、内部通信ポートの異常、Sysmac StudioやNetwork Configuratorの設定異常、TCP アプリケーション（FTP/SNMP）の異常などを確認できます。
以下のように階層構造になっています。各階層のエラーステータスは、その下位の階層のエラーステータスの情報を OR 論理により集約しています。



*1　NY シリーズでの異常関連のエラーステータス変数は、通信ポート 1 と内部通信ポート 1 ごとに分かれています。詳細は🕮「NY シリーズでの EtherNet/IP の異常関連システム定義変数の階層関係」を参照してください。

3-2　システム定義変数一覧

一覧表の見方は、以下のとおりです。

変数名	名称	機能	データ型	値の範囲	参照ページ
[システム定義変数名です。先頭にカテゴリ名が付きます]	[名称です]	[機能を説明します]	[変数のデータ型を示します]	[変数の値の範囲を示します]	[個別仕様表のページを示します]

● EtherNet/IP 通信の異常関連

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP_ErrSta	内蔵 EtherNet/IP 異常	内蔵 EtherNet/IP ポートの異常ステータス変数です。 NY シリーズコントローラの場合、以下の異常フラグを集約します。 • _EIP1_PortErr（通信ポート 1 異常） • _EIPIn1_PortErr（内部通信ポート 1 異常） • _EIP_CipErr（CIP 通信異常） • _EIP_TcpAppErr（TCP アプリ通信異常） （注） 各ビットの意味は後述の「異常ステータスの各ビットの意味」を参照してください。	WORD	16#0000 ～ 16#00F0	3-13
_EIP_PortErr	通信ポート異常	通信ポートの異常ステータス変数です。 NY シリーズコントローラの場合、以下の異常フラグを集約します。 • _EIP1_MacAdrErr（通信ポート 1 MAC アドレス異常） • _EIP1_LanHwErr（通信ポート 1 通信コントローラ異常） • _EIP1_EtnCfgErr（通信ポート 1Ethernet 基本設定異常） • _EIP1_IPAdrCfgErr（通信ポート 1IP アドレス設定異常） • _EIP1_IPAdrDupErr（通信ポート 1IP アドレス重複異常） • _EIP1_BootpErr（通信ポート 1BOOTP サーバ接続異常） • _EIP_DNSCfgErr(DNS 設定異常) • _EIP_DNSSrvErr（DNS サーバ接続異常） • _EIP_IPRTblErr（IP ルートテーブル異常） （注）「リンクオフ検知」または「内蔵 EtherNet/IP 処理異常」時にイベントログに登録して、本異常ビットを ON します。 各ビットの意味は後述の「異常ステータスの各ビットの意味」を参照してください。	WORD	16#0000 ～ 16#00F0	3-13

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP1_PortErr	通信ポート 1 異常	通信ポートの異常ステータス変数です。 以下の異常フラグを集約します。 • _EIP1_MacAdrErr（通信ポート 1 MAC アドレス異常） • _EIP1_LanHwErr（通信ポート 1 通信コントローラ異常） • _EIP1_EtnCfgErr（通信ポート 1Ethernet 基本設定異常） • _EIP1_IPAdrCfgErr（通信ポート 1IP アドレス設定異常） • _EIP1_IPAdrDupErr（通信ポート 1IP アドレス重複異常） • _EIP1_BootpErr（通信ポート 1BOOTP サーバ接続異常） • _EIP_DNSCfgErr(DNS 設定異常) • _EIP_DNSSrvErr (DNS サーバ接続異常) • _EIP_IPRTblErr (I P ルートテーブル異常) （注）「リンクオフ検知」または「内蔵 EtherNet/IP 処理異常」時にイベントログに登録して、本異常ビットを ON します。 各ビットの意味は後述の「異常ステータスの各ビットの意味」を参照してください。	WORD	16#0000 ～ 16#00F0	3-14
_EIPIn1_PortErr	内部通信ポート 1 異常	内部通信ポート 1 の異常ステータス変数です。 以下の異常フラグを集約します。 • _EIPIn1_IPAdrCfgErr（内部通信ポート 1IP アドレス設定異常） • _EIPIn1_IPAdrDupErr（内部通信ポート 1IP アドレス重複異常） • _EIP_DNSCfgErr(DNS 設定異常) • _EIP_DNSSrvErr（DNS サーバ接続異常） • _EIP_IPRTblErr（IP ルートテーブル異常） （注）「リンクオフ検知」または「内蔵 EtherNet/IP 処理異常」時にイベントログに登録して、本異常ビットを ON します。 各ビットの意味は後述の「異常ステータスの各ビットの意味」を参照してください。	WORD	16#0000 ～ 16#00F0	3-14
_EIP_CipErr	CIP 通信異常	CIP 通信の異常ステータス変数です。 以下の異常フラグを集約します。 • _EIP_IdentityErr（Identity 情報不正） • _EIP_TDLinkCfgErr（タグデータリンク設定異常） • _EIP_TDLinkOpnErr（タグデータリンクコネクション確立失敗） • _EIP_TDLinkErr（タグデータリンク通信異常） • _EIP_TagAdrErr（タグ解決異常） • _EIP_MultiSwONErr（複数スイッチ同時 ON 異常） （注）「タグ解決異常」時にイベントログに登録して、本異常を ON します。 各ビットの意味は後述の「異常ステータスの各ビットの意味」を参照してください。	WORD	16#0000 ～ 16#00F0	3-14

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP_TcpAppErr	TCP アプリ通信異常	TCP アプリ通信の異常ステータス変数です。以下の異常フラグを集約します。 • _EIP_TcpAppCfgErr（TCP アプリケーション設定異常） • _EIP_NTPSrvErr（NTP サーバ接続異常） （注） 各ビットの意味は後述の「異常ステータスの各ビットの意味」を参照してください。	WORD	16#0000 ～ 16#00F0	3-15
_EIP_MacAdrErr	MAC アドレス異常	NY シリーズコントローラの場合、電源 ON 時に通信ポート 1 の MAC アドレスの読出異常が発生したことを示します。 TRUE：異常発生 FALSE：正常	BOOL	TRUE、FALSE	3-15
_EIP1_MacAdrErr	通信ポート 1MAC アドレス異常	電源 ON 時に通信ポート 1 の MAC アドレスの読出異常が発生したことを示します。 TRUE：異常発生 FALSE：正常	BOOL	TRUE、FALSE	3-15
_EIP_LanHwErr	通信コントローラ異常	NY シリーズコントローラの場合、通信ポート 1 で通信コントローラの故障が発生していることを示します。 TRUE： 故障発生 FALSE：正常	BOOL	TRUE、FALSE	3-15
_EIP1_LanHwErr	通信ポート 1 通信コントローラ異常	通信ポート 1 で通信コントローラの故障が発生していることを示します。 TRUE： 故障発生 FALSE：正常	BOOL	TRUE、FALSE	3-15
_EIP_EtnCfgErr	Ethernet 基本設定異常	NY シリーズコントローラの場合、通信ポート 1 の Ethernet 通信速度の設定（Speed ／ Duplex）が不正。または、読出に失敗したことを示します。 TRUE： 不正または読出失敗 FALSE：正常	BOOL	TRUE、FALSE	3-16
_EIP1_EtnCfgErr	通信ポート 1Ethernet 基本設定異常	通信ポート 1 の Ethernet 通信速度の設定（Speed ／ Duplex）が不正。または、読出に失敗したことを示します。 TRUE： 不正または読出失敗 FALSE：正常	BOOL	TRUE、FALSE	3-16
_EIP_IPAdrCfgErr	IP アドレス設定異常	NY シリーズコントローラの場合、通信ポート 1 の IP アドレス設定の異常を示します。 TRUE：• IP アドレスの設定値が不正 • 読出に失敗 • BOOTP サーバから取得した IP アドレスが不整合 FALSE：正常	BOOL	TRUE、FALSE	3-16
_EIP1_IPAdrCfgErr	通信ポート 1IP アドレス設定異常	通信ポート 1 の IP アドレス設定の異常を示します。 TRUE：• IP アドレスの設定値が不正 • 読出に失敗 • BOOTP サーバから取得した IP アドレスが不整合 FALSE：正常	BOOL	TRUE、FALSE	3-16
_EIPIn1_IPAdrCfgErr	内部通信ポート 1IP アドレス設定異常	内部通信ポート 1 の IP アドレス設定の異常を示します。 TRUE：• IP アドレスの設定値が不正 • 読出に失敗 • BOOTP サーバから取得した IP アドレスが不整合 FALSE：正常	BOOL	TRUE、FALSE	3-16
_EIP_IPAdrDupErr	IP アドレス重複異常	NY シリーズコントローラの場合、通信ポート 1 で IP アドレスの重複が発生していることを示します。 TRUE： 重複発生 FALSE：上記以外	BOOL	TRUE、FALSE	3-17

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP1_IPAdrDupErr	通信ポート 1IP アドレス重複異常	通信ポート 1 で IP アドレスの重複が発生していることを示します。 TRUE： 重複発生 FALSE：上記以外	BOOL	TRUE、FALSE	3-17
_EIPIn1_IPAdrDupErr	内部通信ポート 1IP アドレス重複異常	内部通信ポート 1 で IP アドレスの重複が発生していることを示します。 TRUE： 重複発生 FALSE：上記以外	BOOL	TRUE、FALSE	3-17
_EIP_DNSCfgErr	DNS 設定異常	DNS 設定値あるいは hosts 設定が不正。または、読出に失敗したことを示します。 TRUE： 不正または読出失敗 FALSE：正常	BOOL	TRUE、FALSE	3-17
_EIP_BootpErr	BOOTP サーバ接続異常	NY シリーズコントローラの場合、通信ポート 1 で BOOTP サーバの接続に失敗したことを示します。 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 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。	BOOL	TRUE、FALSE	3-17
_EIP1_BootpErr	通信ポート 1BOOTP サーバ接続異常	通信ポート 1 で BOOTP サーバの接続に失敗したことを示します。 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 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。	BOOL	TRUE、FALSE	3-18
_EIP_IPRTblErr	IP ルートテーブル異常	NY シリーズコントローラの場合、デフォルトゲートウェイの設定あるいは IP ルータテーブルの設定が不正。 または、読出に失敗したことを示します。 TRUE： 不正または読出失敗 FALSE：正常	BOOL	TRUE、FALSE	3-18
_EIP_IdentityErr	Identity 情報不正	NY シリーズコントローラの場合、CIP 通信 1 の Identity 情報（ユーザによる書換え不可）が不正。または、読出に失敗したことを示します。 TRUE： 不正または読出失敗 FALSE：正常	BOOL	TRUE、FALSE	3-18
_EIP_TDLinkCfgErr	タグデータリンク設定異常	NY シリーズコントローラの場合、CIP 通信 1 のタグデータリンクの設定値が不正。または、読出に失敗したことを示します。 TRUE： 不正または読出失敗 FALSE：正常	BOOL	TRUE、FALSE	3-18
_EIP_TDLinkOpnErr	タグデータリンクコネクション確立失敗	NY シリーズコントローラの場合、CIP 通信 1 のタグデータリンクコネクション確立が失敗したことを示します。 TRUE： タグデータリンクコネクション確立が、以下の原因により失敗しました。 • 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っている。 • 相手ノードからの応答がない。 FALSE：上記以外	BOOL	TRUE、FALSE	3-18
_EIP_TDLinkErr	タグデータリンク通信異常	NY シリーズコントローラの場合、CIP 通信 1 のタグデータリンクコネクションにタイムアウトが発生していることを示します。 TRUE： タイムアウト発生 FALSE：上記以外	BOOL	TRUE、FALSE	3-19

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP_TagAdrErr	タグ解決異常	NY シリーズコントローラの場合、CIP 通信 1 のタグ解決（タグ名からアドレスを特定すること）に失敗したことを示します。 TRUE： タグ解決（タグ名からアドレスを特定すること）に失敗しました。以下の要因があります。 • ネットワーク変数とタグ設定間で、サイズが異なる。 • タグデータリンクの設定入出力方向と、コントローラ内の変数の入出力方向が一致しない。 • コントローラ内に、タグ設定に該当するネットワーク変数が存在しない。 FALSE：上記以外	BOOL	TRUE、FALSE	3-19
_EIP_MultiSwONErr	複数スイッチ同時 ON 異常	NY シリーズコントローラの場合、CIP 通信 1 で複数のスイッチが同時に ON されたことを示します。 TRUE： 複数のデータリンク開始／データリンク停止スイッチが同時に TRUE とされました。 FALSE：上記以外	BOOL	TRUE、FALSE	3-19
_EIP_TcpAppCfgErr	TCP アプリケーション設定異常	TRUE： 少なくとも 1 つ以上、TCP アプリケーション機能（FTP、NTP、SNMP）の設定値に不正があります。または、読出に失敗しました。 FALSE：正常	BOOL	TRUE、FALSE	3-19
_EIP_NTPSrvErr	NTP サーバ接続異常	NY シリーズコントローラの場合、常に FALSE となります。	BOOL	TRUE、FALSE	3-19
_EIP_DNSSrvErr	DNS サーバ接続異常	TRUE： DNS 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：DNS の設定がされていない。または、DNS の設定がされていて接続に成功しました。	BOOL	TRUE、FALSE	3-20



参考

NY シリーズでの EtherNet/IP の異常関連システム定義変数の階層関係

EtherNet/IP の異常関連のシステム定義変数は、下表のように階層関係になっています。たとえば、第 2 階層の _EIP1_PortErr、_EIPIn1_PortErr、_EIP_CipErr、_EIP_TcpAppErr のいずれかの値が 0 でない場合に、その上位階層の _EIP_ErrSta の値が 0 でなくなります。つまり、上位の階層のシステム定義変数の値を参照することによって、それよりも下位の階層の異常が発生しているかどうかがわかります。

第 1 階層		第 2 階層		第 3 階層	
変数名	名称	変数名	名称	変数名	名称
_EIP_ErrSta	内蔵 EtherNet/IP 異常	_EIP1_PortErr	通信ポート 1 異常	_EIP1_MacAdrErr	通信ポート 1MAC アドレス異常
				_EIP1_LanHwErr	通信ポート 1 通信コントローラ異常
				_EIP1_EtnCfgErr	通信ポート 1Ethernet 基本設定異常
				_EIP1_IPAdrCfgErr	通信ポート 1IP アドレス設定異常
				_EIP1_IPAdrDupErr	通信ポート 1IP アドレス重複異常
				_EIP1_BootpErr	通信ポート 1BOOTP サーバ接続異常
				_EIP_DNSCfgErr	DNS 設定異常
				_EIP_DNSSrvErr	DNS サーバ接続異常
				_EIP_IPRTblErr	IP ルートテーブル異常
		_EIPIn1_PortErr	内部通信ポート 1 異常	_EIPIn1_IPAdrCfgErr	内部通信ポート 1IP アドレス設定異常
				_EIPIn1_IPAdrDupErr	内部通信ポート 1IP アドレス重複異常
				_EIP_DNSCfgErr	DNS 設定異常
				_EIP_DNSSrvErr	DNS サーバ接続異常
				_EIP_IPRTblErr	IP ルートテーブル異常
		_EIP_CipErr	CIP 通信異常	_EIP_IdentityErr	Identity 情報不正
				_EIP_TDLinkCfgErr	タグデータリンク設定異常
				_EIP_TDLinkOpnErr	タグデータリンクコネクション確立失敗
				_EIP_TDLinkErr	タグデータリンク通信異常
				_EIP_TagAdrErr	タグ解決異常
				_EIP_MultiSwONErr	複数スイッチ同時 ON
		_EIP_TcpAppErr	TCP アプリ通信異常	_EIP_TcpAppCfgErr	TCP アプリケーション設定異常
				_EIP_NTPSrvErr	NTP サーバ接続異常

（注）変数名が「_EIP1」ではじまるシステム定義変数は、変数名が「_EIP」ではじまる同名のシステム定義変数と同じ値を参照します。たとえば、_EIP1_PortErr（通信ポート 1 異常）は、_EIP_PortErr（通信ポート異常）と同じ値を参照します。

● 異常ステータスの各ビットの意味

ビット:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORD			－	－	－	－	－	－					－	－	－	－

ビット	内容
15	予約
14	スレーブ集約：イベント発生源（各機能モジュール）の下位階層（ユニット／スレーブ／軸／軸グループ等）において、コントローラ異常が発生しているか否かを示します。 TRUE:　いずれかの下位階層でコントローラ異常あり FALSE:　いずれかの下位階層でコントローラ異常なし （_MC_ErrSta /_EC_ErrSta で有効。）
13 ～ 8	予約
7	全停止フォールト (Major fault) レベルのコントローラ異常が発生中か否かを示します。 TRUE:　全停止フォールトレベルのコントローラ異常が発生中 FALSE:　全停止フォールトレベルのコントローラ異常が発生中でない
6	部分停止フォールト (Partial fault) レベルのコントローラ異常が発生中か否かを示します。 TRUE:　部分停止フォールトレベルのコントローラ異常が発生中 FALSE:　部分停止フォールトレベルのコントローラ異常が発生中でない
5	軽度フォールト (Minor fault) レベルのコントローラ異常が発生中か否かを示します。 TRUE:　軽度フォールトレベルのコントローラ異常が発生中 FALSE:　軽度フォールトレベルのコントローラ異常が発生中でない
4	監視情報 (Observation) レベルのコントローラ異常が発生中か否かを示します。 TRUE:　監視情報レベルのコントローラ異常が発生中 FALSE:　監視情報レベルのコントローラ異常が発生中でない
3 ～ 0	予約

（注） ビット 14、15 は、内蔵 EtherNet/IP の要因では TRUE になりません。

使用上の注意

「_ErrSta」（コントローラ異常ステータス）は、リアルタイム性、および各機能モジュールの異常要因を表すステータスとの同時性の観点から、ユーザプログラム上では使用できません。通信を介して外部から状態を参照するときにのみ使用します。

● EtherNet/IP 通信のステータス関連

内蔵 EtherNet/IP ポートのステータス（通信状態など）を確認できます。

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP_EtnOnlineSta	オンライン	NY シリーズコントローラの場合、内蔵 EtherNet/IP ポートの通信ポート 1 の通信機能が利用可能（リンク ON かつ、IP アドレス決定。かつ、異常が発生していないこと）になっていることを示します。 TRUE： 利用可能 FALSE：初期処理中、再起動中、異常発生、リンクオフにより内蔵 EtherNet/IP ポートの通信機能が利用できません。	BOOL	TRUE、FALSE	3-20
_EIP1_EtnOnlineSta	通信ポート 1 オンライン	内蔵 EtherNet/IP ポートの通信ポート 1 の通信機能が利用可能（リンク ON かつ、IP アドレス決定。かつ、異常が発生していないこと）になっていることを示します。 TRUE： 利用可能 FALSE：初期処理中、再起動中、異常発生、リンクオフにより内蔵 EtherNet/IP ポートの通信機能が利用できません。	BOOL	TRUE、FALSE	3-20
_EIPIn1_EtnOnlineSta	内部通信ポート 1 オンライン	内蔵 EtherNet/IP ポートの内部通信ポート 1 の通信機能が利用可能（リンク ON かつ、IP アドレス決定。かつ、異常が発生していないこと）になっていることを示します。 TRUE： 利用可能 FALSE：初期処理中、再起動中、異常発生、リンクオフにより内蔵 EtherNet/IP ポートの通信機能が利用できません。	BOOL	TRUE、FALSE	3-20
_EIP_TDLinkRunSta	タグデータリンク通信中	NY シリーズコントローラの場合、CIP 通信 1 で少なくとも 1 つ以上のコネクションが正常動作していることを示します。 TRUE： 正常動作 FALSE：上記以外	BOOL	TRUE、FALSE	3-20
_EIP_TDLinkAllRunSta	全タグデータリンク通信中	NY シリーズコントローラの場合、CIP 通信 1 の全タグデータリンクが通信中であることを示します。 TRUE： オリジネータとしてすべ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：少なくとも 1 つ以上のコネクションに異常が発生しています。	BOOL	TRUE、FALSE	3-21
_EIP_RegTargetSta[255]	登録ターゲットノード情報	NY シリーズコントローラの場合、内蔵 EtherNet/IP の CIP 通信 1 にコネクションが登録されているノードの一覧を表します。 本変数は内蔵 EtherNet/IP がオリジネータとなっている場合だけ有効となります。 配列 [x] が TRUE： ターゲットノード ID「x」のノードとのコネクションが登録されています。 配列 [x] が FALSE： ターゲットノード ID「x」のノードとのコネクションが登録されていません。	ARRAY [0..255] OF BOOL	TRUE、FALSE	3-21

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP_EstbTargetSta[255]	正常ターゲットノード情報	NY シリーズコントローラの場合、内蔵 EtherNet/IP の CIP 通信 1 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： ターゲットノード ID「x」のノードとのコネクションが正常に確立しています。 配列 [x] が FALSE： ターゲットノード ID「x」のノードとの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。	ARRAY [0..255] OF BOOL	TRUE、FALSE	3-21
_EIP_TargetPLCModeSta［255］	ターゲット PLC 動作モード	NY シリーズコントローラの場合、内蔵 EtherNet/IP がオリジネータとなっている CIP 通信 1 の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 のときだけ有効です。FALSE の場合は直前の値が保持されます。 配列 [x] が TRUE： ターゲットノード ID「x」のターゲットコントローラが運転している状態です。 配列 [x] が FALSE： 上記以外	ARRAY [0..255] OF BOOL	TRUE、FALSE	3-21
_EIP_TargetPLCErr［255］	ターゲット PLC 異常情報	NY シリーズコントローラの場合、内蔵 EtherNet/IP がオリジネータとなっている CIP 通信 1 の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本体の異常状態（運転停止、継続異常の論理和）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 のときだけ有効です。FALSE の場合は直前の値を保持します。 配列 [x] が TRUE： ターゲットノード ID「x」のターゲットコントローラが運転継続、または、停止異常が発生しています。 配列 [x] が FALSE： 上記以外	ARRAY [0..255] OF BOOL	TRUE、FALSE	3-22
_EIP_TargetNodeErr［255］	ターゲットノード異常情報	NY シリーズコントローラの場合、CIP 通信 1 の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 の場合だけ有効です。 配列 [x] が TRUE： ターゲットノード ID「x」のターゲットノードとのコネクションが正常に確立していない （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 FALSE）、 または、ターゲットノードとのコネクションが確立しているが、ターゲットコントローラに異常が発生しています。 配列 [x] が FALSE： ターゲットノード ID「x」のターゲットノードが登録されていない（登録ターゲットノード情報が FALSE）、または、ターゲットノードとのコネクションは確立していて、（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） ターゲットコントローラに異常が発生しています（ターゲット PLC 異常情報が TRUE）。	ARRAY [0..255] OF BOOL	TRUE、FALSE	3-22

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP_NTPResult	NTP の動作情報	－	_sNTP_RESULT		3-22
.ExecTime	NTP の最終動作時刻	NY シリーズコントローラの場合、初期値から変化しません。	DATE_AND_TIME	データ型に従う	3-22
.ExecNormal	NTP の動作結果	NY シリーズコントローラの場合、初期値から変化しません。	BOOL	TRUE、FALSE	3-22



参考

ターゲットノードとの通信状態

NY シリーズコントローラとターゲットノードとの通信状態は、以下の 4 つのシステム定義変数の値の組み合わせで、次の表のように決まります。

- _EIP_RegTargetSta（登録ターゲットノード情報）
- _EIP_EstbTargetSta（正常ターゲットノード情報）
- _EIP_TargetPLCErr（ターゲット PLC 異常情報）
- _EIP_TargetNodeErr（ターゲットノード異常情報）

_EIP_Reg TargetSta の値	_EIP_Estb TargetSta の値	_EIP_Target PLCErr の値	_EIP_Target NodeErr の値	ターゲットノードとの通信状態
TRUE	TRUE	FALSE	FALSE	ターゲットノードとのコネクションが正常に確立し、かつターゲット PLC に異常が発生していません。
		TRUE	TRUE	ターゲットノードとのコネクションは確立していますが、ターゲット PLC に異常が発生しています。
	FALSE	－	TRUE	ターゲットノードとのコネクションが正常に確立していません。
FALSE	－	－	－	ターゲットノードが登録されていないため、情報は無効です。

● EtherNet/IP 通信のスイッチ関連

タグデータリンクの開始、停止を実行できます。

変数名	名称	機能	データ型	値の範囲	参照ページ
_EIP_TDLinkStartCmd	タグデータリンク通信開始スイッチ	NY シリーズコントローラの場合、FALSE から TRUE にすると、CIP 通信 1 のタグデータリンクを開始します。タグデータリンク動作が開始すると、自動的に FALSE になります。 （注）自動的に FALSE になるまでは、このスイッチを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。	BOOL	TRUE、FALSE	3-23
_EIP_TDLinkStopCmd	タグデータリンク通信停止スイッチ	NY シリーズコントローラの場合、FALSE から TRUE にすると、CIP 通信 1 のタグデータリンクが停止します。タグデータリンク動作が停止すると、自動的に FALSE になります。 （注）自動的に FALSE になるまでは、このスイッチを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。	BOOL	TRUE、FALSE	3-23

3-3　システム定義変数の個別仕様

システム定義変数の個別仕様表の見方は、以下のとおりです。

変数名	［システム定義変数名です。先頭にカテゴリ名が付きます］			**メンバ（一部のみ）**	［構造体型の場合のみメンバ名を示します］
名称	［名称です］			**グローバル／ローカル**	［グローバル：グローバル変数、ローカル：ローカル変数］
機能	［機能を説明します］				
データ型	［変数のデータ型を示します］			**値の範囲**	［変数の値の範囲を示します］
R/W アクセス	［R：読出専用 RW：読書可能］	**保持**	［変数の保持属性を示します］	**ネットワーク公開**	［変数のネットワーク公開属性を示します］
ユーザプログラムでの使用可否	［ユーザプログラム上で直接指定して使用することが可能か否かを示します］	**関連命令**	［変数に関連する命令を示します。「ユーザプログラムでの使用」が不可の場合は、アクセスするための専用命令を示します］		

● 機能分類：EtherNet/IP 通信の異常関連

変数名	_EIP_ErrSta				
名称	内蔵 EtherNet/IP 異常			**グローバル／ローカル**	グローバル
機能	内蔵 EtherNet/IP ポートの異常ステータス変数です。 NY シリーズコントローラの場合、以下の異常フラグを集約します。 • _EIP1_PortErr（通信ポート 1 異常） • _EIPIn1_PortErr（内部通信ポート 1 異常） • _EIP_CipErr（CIP 通信異常） • _EIP_TcpAppErr（TCP アプリ通信異常） （注） 各ビットの意味は、📖 「異常ステータスの各ビットの意味」を参照してください。				
データ型	WORD			**値の範囲**	16#0000 ～ 16#00F0
R/W アクセス	R	**保持**	非保持	**ネットワーク公開**	公開
ユーザプログラムでの使用可否	可能	**関連命令**	ユーザプログラムからは以下の専用命令でアクセス可能です。 • GetEIPError		

変数名	_EIP_PortErr				
名称	通信ポート異常			**グローバル／ローカル**	グローバル
機能	通信ポートの異常ステータス変数です。 NY シリーズコントローラの場合、以下の異常フラグを集約します。 • _EIP1_MacAdrErr（通信ポート 1 MAC アドレス異常） • _EIP1_LanHwErr（通信ポート 1 通信コントローラ異常） • _EIP1_EtnCfgErr（通信ポート 1Ethernet 基本設定異常） • _EIP1_IPAdrCfgErr（通信ポート 1IP アドレス設定異常） • _EIP1_IPAdrDupErr（通信ポート 1IP アドレス重複異常） • _EIP1_BootpErr（通信ポート 1BOOTP サーバ接続異常） • _EIP_DNSCfgErr(DNS 設定異常) • _EIP_DNSSrvErr（DNS サーバ接続異常） • _EIP_IPRTblErr（IP ルートテーブル異常） （注）「リンクオフ検知」または「内蔵 EtherNet/IP 異常」時にイベントログに登録して、本異常ビットを ON します。各ビットの意味は、📖 「異常ステータスの各ビットの意味」を参照してください。				
データ型	WORD			**値の範囲**	16#0000 ～ 16#00F0
R/W アクセス	R	**保持**	非保持	**ネットワーク公開**	公開
ユーザプログラムでの使用可否	可能	**関連命令**	ユーザプログラムからは以下の専用命令でアクセス可能です。 • GetEIPError		

変数名	_EIP1_PortErr				
名称	通信ポート 1 異常			グローバル／ローカル	グローバル
機能	通信ポート 1 の異常ステータス変数です。 以下の異常フラグを集約します。 • _EIP1_MacAdrErr（通信ポート 1 MAC アドレス異常） • _EIP1_LanHwErr（通信ポート 1 通信コントローラ異常） • _EIP1_EtnCfgErr（通信ポート 1Ethernet 基本設定異常） • _EIP1_IPAdrCfgErr（通信ポート 1IP アドレス設定異常） • _EIP1_IPAdrDupErr（通信ポート 1IP アドレス重複異常） • _EIP1_BootpErr（通信ポート 1BOOTP サーバ接続異常） • _EIP_DNSCfgErr(DNS 設定異常) • _EIP_DNSSrvErr（DNS サーバ接続異常） • _EIP_IPRTblErr（IP ルートテーブル異常） （注）「リンクオフ検知」または「内蔵 EtherNet/IP 異常」時にイベントログに登録して、本異常ビットを ON します。各ビットの意味は、📖 「異常ステータスの各ビットの意味」を参照してください。				
データ型	WORD			値の範囲	16#0000 ～ 16#00F0
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	ユーザプログラムからは以下の専用命令でアクセス可能です。 • GetEIPError		

変数名	_EIPIn1_PortErr				
名称	内部通信ポート 1 異常			グローバル／ローカル	グローバル
機能	内部通信ポート 1 の異常ステータス変数です。 以下の異常フラグを集約します。 • _EIPIn1_IPAdrCfgErr（内部通信ポート 1IP アドレス設定異常） • _EIPIn1_IPAdrDupErr（内部通信ポート 1IP アドレス重複異常） • _EIP_DNSCfgErr(DNS 設定異常) • _EIP_DNSSrvErr（DNS サーバ接続異常） • _EIP_IPRTblErr（IP ルートテーブル異常） （注）「リンクオフ検知」または「内蔵 EtherNet/IP 異常」時にイベントログに登録して、本異常ビットを ON します。各ビットの意味は、📖 「異常ステータスの各ビットの意味」を参照してください。				
データ型	WORD			値の範囲	16#0000 ～ 16#00F0
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	ユーザプログラムからは以下の専用命令でアクセス可能です。 • GetEIPError		

変数名	_EIP_CipErr				
名称	CIP 通信異常			グローバル／ローカル	グローバル
機能	CIP 通信の異常ステータス変数です。 以下の異常フラグを集約します。 • _EIP_IdentityErr（Identity 情報不正） • _EIP_TDLinkCfgErr（タグデータリンク設定異常） • _EIP_TDLinkOpnErr（タグデータリンクコネクション確立失敗） • _EIP_TDLinkErr（タグデータリンク通信異常） • _EIP_TagAdrErr（タグ解決異常） • _EIP_MultiSwONErr（複数スイッチ同時 ON 異常） （注）「タグ解決異常」時にイベントログに登録して、本異常を ON します。各ビットの意味は、📖 「異常ステータスの各ビットの意味」を参照してください。				
データ型	WORD			値の範囲	16#0000 ～ 16#00F0
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	ユーザプログラムからは以下の専用命令でアクセス可能です。 • GetEIPError		

変数名	_EIP_TcpAppErr				
名称	TCP アプリ通信異常			グローバル／ローカル	グローバル
機能	TCP アプリ通信の異常ステータス変数です。 以下の異常フラグを集約します。 • _EIP_TcpAppCfgErr（TCP アプリケーション設定異常） • _EIP_NTPSrvErr（NTP サーバ接続異常） （注） 各ビットの意味は、📖 「異常ステータスの各ビットの意味」を参照してください。				
データ型	WORD			値の範囲	16#0000 ～ 16#00F0
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	ユーザプログラムからは以下の専用命令でアクセス可能です。 • GetEIPError		

変数名	_EIP_MacAdrErr				
名称	MAC アドレス異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、電源 ON 時に通信ポート 1 の MAC アドレスの読出異常が発生したことを示します。 TRUE：異常発生 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP1_MacAdrErr				
名称	通信ポート 1MAC アドレス異常			グローバル／ローカル	グローバル
機能	電源 ON 時に通信ポート 1 の MAC アドレスの読出異常が発生したことを示します。 TRUE：異常発生 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_LanHwErr				
名称	通信コントローラ異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、通信ポート 1 で通信コントローラの故障が発生していることを示します。 TRUE：故障発生 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP1_LanHwErr				
名称	通信ポート 1 通信コントローラ異常			グローバル／ローカル	グローバル
機能	通信ポート 1 で通信コントローラの故障が発生していることを示します。 TRUE：故障発生 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_EtnCfgErr				
名称	Ethernet 基本設定異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、通信ポート 1 の Ethernet 通信速度の設定（Speed ／ Duplex）が不正。または、読出に失敗したことを示します。 TRUE：不正または読出失敗 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP1_EtnCfgErr				
名称	通信ポート 1Ethernet 基本設定異常			グローバル／ローカル	グローバル
機能	通信ポート 1 の Ethernet 通信速度の設定（Speed ／ Duplex）が不正。または、読出に失敗したことを示します。 TRUE：不正または読出失敗 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_IPAdrCfgErr				
名称	IP アドレス設定異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、通信ポート 1 の IP アドレス設定の異常を示します。 TRUE：• IP アドレスの設定値が不正 • 読出に失敗 • BOOTP サーバから取得した IP アドレスが不整合 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP1_IPAdrCfgErr				
名称	通信ポート 1IP アドレス設定異常			グローバル／ローカル	グローバル
機能	通信ポート 1 の IP アドレス設定の異常を示します。 TRUE：• IP アドレスの設定値が不正 • 読出に失敗 • BOOTP サーバから取得した IP アドレスが不整合 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIPIn1_IPAdrCfgErr				
名称	内部通信ポート 1IP アドレス設定異常			グローバル／ローカル	グローバル
機能	内部通信ポート 1 の IP アドレス設定の異常を示します。 TRUE：• IP アドレスの設定値が不正 • 読出に失敗 • BOOTP サーバから取得した IP アドレスが不整合 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_IPAdrDupErr				
名称	IP アドレス重複異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、通信ポート 1 で IP アドレスの重複が発生していることを示します。 TRUE：重複発生 FALSE：上記以外				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP1_IPAdrDupErr				
名称	通信ポート 1IP アドレス重複異常			グローバル／ローカル	グローバル
機能	通信ポート 1 で IP アドレスの重複が発生していることを示します。 TRUE：重複発生 FALSE：上記以外				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIPIn1_IPAdrDupErr				
名称	内部通信ポート 1IP アドレス重複異常			グローバル／ローカル	グローバル
機能	内部通信ポート 1 で IP アドレスの重複が発生していることを示します。 TRUE：重複発生 FALSE：上記以外				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_DNSCfgErr				
名称	DNS 設定異常			グローバル／ローカル	グローバル
機能	DNS 設定値あるいは hosts 設定が不正。または、読出に失敗したことを示します。 TRUE：不正または読出失敗 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_BootpErr				
名称	BOOTP サーバ接続異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、通信ポート 1 で BOOTP サーバの接続に失敗したことを示します。 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 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。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP1_BootpErr				
名称	通信ポート 1BOOTP サーバ接続異常			グローバル／ローカル	グローバル
機能	通信ポート 1 で BOOTP サーバの接続に失敗したことを示します。 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 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。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_IPRTblErr				
名称	IP ルートテーブル異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、デフォルトゲートウェイの設定あるいは IP ルータテーブルの設定が不正。または、読出に失敗したことを示します。 TRUE：不正または読出失敗 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_IdentityErr				
名称	Identity 情報不正			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 の Identity 情報（ユーザによる書換え不可）が不正。または、読出に失敗したことを示します。 TRUE：不正または読出失敗 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TDLinkCfgErr				
名称	タグデータリンク設定異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 のタグデータリンクの設定値が不正。または、読出に失敗したことを示します。 TRUE：不正または読出失敗 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TDLinkOpnErr				
名称	タグデータリンクコネクション確立失敗			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 のタグデータリンクコネクション確立が失敗したことを示します。 TRUE：タグデータリンクコネクション確立が、以下の原因により失敗しました。 • 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っている。 • 相手ノードからの応答がない。 FALSE：上記以外				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TDLinkErr				
名称	タグデータリンク通信異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 のタグデータリンクコネクションにタイムアウトが発生していることを示します。 TRUE：タイムアウト発生 FALSE：上記以外				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TagAdrErr				
名称	タグ解決異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 のタグ解決（タグ名からアドレスを特定すること）に失敗したことを示します。 TRUE：タグ解決（タグ名からアドレスを特定すること）に失敗しました。以下の要因があります。 • ネットワーク変数とタグ設定間で、サイズが異なる。 • タグデータリンクの設定入出力方向と、コントローラ内の変数の入出力方向が一致しない。 • コントローラ内に、タグ設定に該当するネットワーク変数が存在しない。 FALSE：上記以外				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_MultiSwONErr				
名称	複数スイッチ同時 ON 異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 で複数のスイッチが同時に ON されたことを示します。 TRUE：複数のデータリンク開始／データリンク停止スイッチが同時に TRUE とされました。 FALSE：上記以外				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TcpAppCfgErr				
名称	TCP アプリケーション設定異常			グローバル／ローカル	グローバル
機能	TRUE：少なくとも 1 つ以上、TCP アプリケーション機能（FTP、NTP、SNMP）の設定値に不正があります。 または、読出に失敗しました。 FALSE：正常				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_NTPSrvErr				
名称	NTP サーバ接続異常			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、常に FALSE となります。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_DNSSrvErr				
名称	DNS サーバ接続異常			グローバル／ローカル	グローバル
機能	TRUE：DNS 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：DNS の設定がされていない。または、DNS の設定がされていて接続に成功しました。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

● 機能分類：EtherNet/IP 通信のステータス関連

変数名	_EIP_EtnOnlineSta				
名称	オンライン			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、内蔵 EtherNet/IP ポートの通信ポート 1 の通信機能が利用可能（リンク ON かつ、IP アドレス決定。かつ、異常が発生していないこと）になっていることを示します。 TRUE：利用可能 FALSE：初期処理中、再起動中、異常発生、リンクオフにより内蔵 EtherNet/IP ポートの通信機能が利用できません。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP1_EtnOnlineSta				
名称	通信ポート 1 オンライン			グローバル／ローカル	グローバル
機能	内蔵 EtherNet/IP ポートの通信ポート 1 の通信機能が利用可能（リンク ON かつ、IP アドレス決定。かつ、異常が発生していないこと）になっていることを示します。 TRUE：利用可能 FALSE：初期処理中、再起動中、異常発生、リンクオフにより内蔵 EtherNet/IP ポートの通信機能が利用できません。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIPIn1_EtnOnlineSta				
名称	内部通信ポート 1 オンライン			グローバル／ローカル	グローバル
機能	内蔵 EtherNet/IP ポートの内部通信ポート 1 の通信機能が利用可能（リンク ON かつ、IP アドレス決定。かつ、異常が発生していないこと）になっていることを示します。 TRUE：利用可能 FALSE：初期処理中、再起動中、異常発生、リンクオフにより内蔵 EtherNet/IP ポートの内部通信ポート 1 の通信機能が利用できません。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TDLinkRunSta				
名称	タグデータリンク通信中			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 で少なくとも 1 つ以上のコネクションが正常動作していることを示します。 TRUE：正常動作 FALSE：上記以外				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TDLinkAllRunSta				
名称	全タグデータリンク通信中			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 の全タグデータリンクが通信中であることを示します。 TRUE：オリジネータとしてすべ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：少なくとも 1 つ以上のコネクションに異常が発生しています。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_RegTargetSta［255］				
名称	登録ターゲットノード情報			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、内蔵 EtherNet/IP の CIP 通信 1 にコネクションが登録されているノードの一覧を表します。 本変数は内蔵 EtherNet/IP がオリジネータとなっている場合だけ有効となります。 配列 [x] が TRUE：ターゲットノード ID「x」のノードとのコネクションが登録されています。 配列 [x] が FALSE：ターゲットノード ID「x」のノードとのコネクションが登録されていません。				
データ型	ARRAY [0..255] OF BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_EstbTargetSta［255］				
名称	正常ターゲットノード情報			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、内蔵 EtherNet/IP の CIP 通信 1 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：ターゲットノード ID「x」のノードとのコネクションが正常に確立しています。 配列 [x] が FALSE：ターゲットノード ID「x」のノードとの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。				
データ型	ARRAY [0..255] OF BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TargetPLCModeSta［255］				
名称	ターゲット PLC 動作モード			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、内蔵 EtherNet/IP がオリジネータとなっている CIP 通信 1 の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 のときだけ有効です。FALSE の場合は直前の値が保持されます。 配列 [x] が TRUE：ターゲットノード ID「x」のターゲットコントローラが運転している状態です。 配列 [x] が FALSE：上記以外				
データ型	ARRAY [0..255] OF BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TargetPLCErr［255］				
名称	ターゲット PLC 異常情報			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、内蔵 EtherNet/IP がオリジネータとなっている CIP 通信 1 の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本体の異常状態（運転停止、継続異常の論理和）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 のときだけ有効です。FALSE の場合は直前の値を保持します。 配列 [x] が TRUE：ターゲットノード ID「x」のターゲットコントローラが運転継続、または、停止異常が発生しています。 配列 [x] が FALSE：上記以外				
データ型	ARRAY [0..255] OF BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TargetNodeErr［255］				
名称	ターゲットノード異常情報			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、CIP 通信 1 の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 の場合だけ有効です。 配列 [x] が TRUE：ターゲットノード ID「x」のターゲットノードとのコネクションが正常に確立していない（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 FALSE）、または、ターゲットノードとのコネクションが確立しているが、ターゲットコントローラに異常が発生しています。 配列 [x] が FALSE：ターゲットノード ID「x」のターゲットノードが登録されていない（登録ターゲットノード情報が FALSE）、または、ターゲットノードとのコネクションは確立していて、（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）ターゲットコントローラに異常が発生しています（ターゲット PLC 異常情報が TRUE）。				
データ型	ARRAY [0..255] OF BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_NTPResult			メンバ名	.ExecTime
名称	NTP の最終動作時刻			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、初期値から変化しません。				
データ型	構造体：_sNTP_RESULT メンバ：DATE_AND_TIME			値の範囲	データ型に従う
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	不可	関連命令	GetNTPStatus 命令で読出可能		

変数名	_EIP_NTPResult			メンバ名	.ExecNormal
名称	NTP の動作結果			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、初期値から変化しません。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	不可	関連命令	GetNTPStatus 命令で読出可能		

● 機能分類：EtherNet/IP 通信のスイッチ関連

変数名	_EIP_TDLinkStartCmd				
名称	タグデータリンク通信開始スイッチ			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、FALSE から TRUE にすると、CIP 通信 1 のタグデータリンクを開始します。 タグデータリンク動作が開始すると、自動的に FALSE になります。 （注） 自動的に FALSE になるまでは、このスイッチを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。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R/W	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

変数名	_EIP_TDLinkStopCmd				
名称	タグデータリンク通信停止スイッチ			グローバル／ローカル	グローバル
機能	NY シリーズコントローラの場合、FALSE から TRUE にすると、CIP 通信 1 のタグデータリンクが停止します。 タグデータリンク動作が停止すると、自動的に FALSE になります。 （注） 自動的に FALSE になるまでは、このスイッチを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。				
データ型	BOOL			値の範囲	TRUE、FALSE
R/W アクセス	R/W	保持	非保持	ネットワーク公開	公開
ユーザプログラムでの使用可否	可能	関連命令	－		

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 4-4

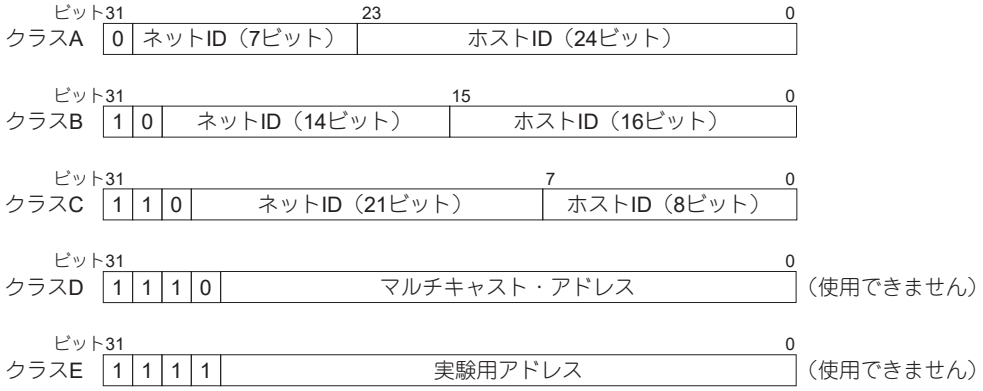
4-2 コントローラの自 IP アドレス設定方法 4-5
4-2-1 IP アドレスを設定する方法 4-5
4-2-2 IP アドレスの設定手順 4-6
4-2-3 オンライン接続する 4-8
4-2-4 使用中の IP アドレスの確認方法 4-10

4-3 プライベート／グローバルアドレス 4-11
4-3-1 グローバルアドレスとプライベートアドレス 4-11
4-3-2 内蔵 EtherNet/IP ポートがプライベートアドレスの場合 4-12
4-3-3 内蔵 EtherNet/IP ポートがグローバルアドレスの場合 4-13

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）

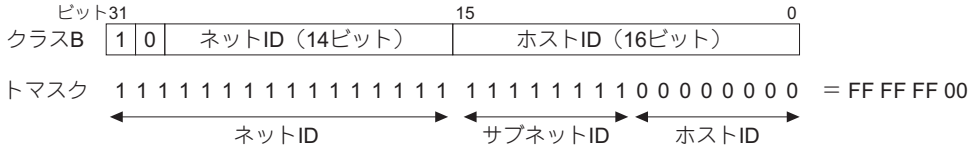
IP アドレスは、32 ビットのバイナリデータを 8 ビットずつの 4 つの組に分け、それぞれをドットで区切った 4 つの 10 進数で表記します。
（例）10000010　00111010　00010001　00100000　→　130.58.17.32

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 内蔵ポートでは、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

サブネットマスクを利用し、各ネットワークを識別するための情報にネットワークアドレスがあります。ネットワークアドレスによって、ある複数ノードが同一ネットワークに属するかどうかが分かります。ネットワークアドレスは、ノードの IP アドレスとサブネットマスクの論理積で算出されます。
次にネットワークアドレスの算出例を示します。
ノード 1 の IP アドレスに「192.168.250.20」、ノード 2 の IP アドレスに「192.168.245.30」、サブネットマスクに「255.255.240.0 」と設定すると、それぞれのノードのネットワークアドレスは下記の表のとおりです。
・ ノード 1 のネットワークアドレスの算出

項目	10 進表記	2 進数表記
IP アドレス	192.168.250.20	11000000.10101000.11111010.00010100
サブネットマスク	255.255.240.0	11111111.11111111.11110000.00000000
ネットワークアドレス	192.168.240.0	11000000.10101000.11110000.00000000

・ ノード 2 のネットワークアドレスの算出

項目	10 進表記	2 進数表記
IP アドレス	192.168.245.30	11000000.10101000.11111010.00010100
サブネットマスク	255.255.240.0	11111111.11111111.11110000.00000000
ネットワークアドレス	192.168.240.0	11000000.10101000.11110000.00000000

上記よりノード 1 とノード 2 のネットワークアドレスは同じため、同一ネットワークに属します。

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　コントローラの自 IP アドレス設定方法

4-2-1　IP アドレスを設定する方法

NY シリーズコントローラの内蔵 EtherNet/IP ポートの自 IP アドレスは、以下のいずれかの方法で設定します。内部通信ポートの自 IP アドレスは、任意の IP アドレスを設定する方法のみ可能です。

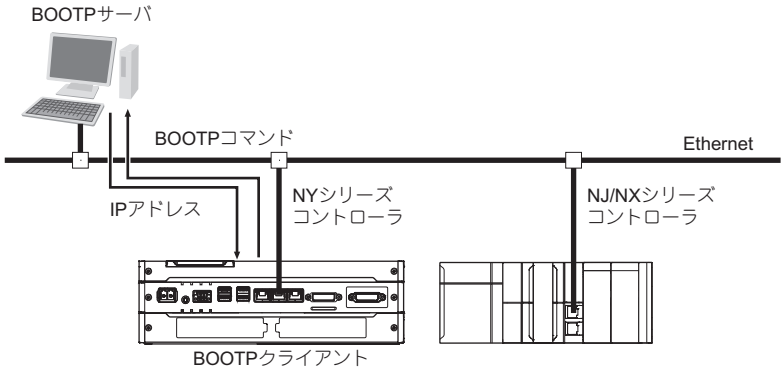
● 任意の IP アドレスを設定する

内蔵 EtherNet/IP ポートのデフォルトの IP アドレスを変更するとき、あるいは内蔵 EtherNet/IP ポートと他の 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

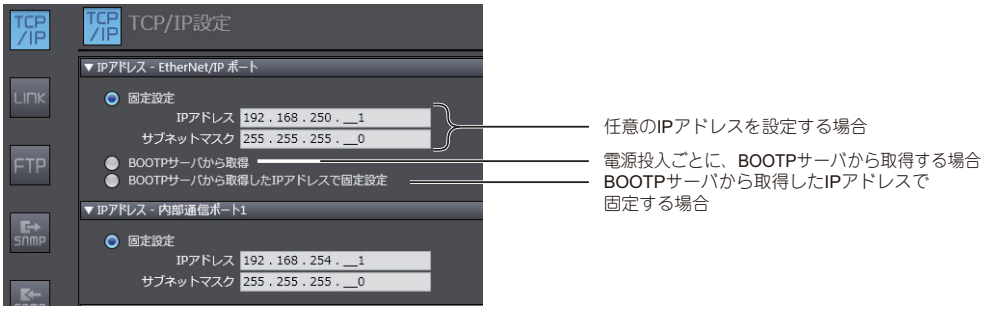
内蔵 EtherNet/IP ポートおよび内部通信ポートの自 IP アドレスは、Sysmac Studio で設定します。

参考

Sysmac Studio を使用せずに、Industrial PC Support Utility を使用して、自 IP アドレスの設定を行うことができます。
Industrial PC Support Utility による自 IP アドレスの設定方法については、🕮『NY シリーズ 産業用パネル型PC／産業用ボックス型PC ユーザーズマニュアル セットアップ編(SBCA-441)』を参照してください。

1　IP アドレスの設定方法を選択します。

コントローラ設定の内蔵 EtherNet/IP ポート設定の [TCP/IP] ダイアログから、自 IP アドレスの設定方法を、以下のように選択します。



使用上の注意

内蔵 EtherNet/IP ポートと内部通信ポートが同一ネットワークとなる IP アドレスは設定できません。

2　Sysmac Studio を、NY シリーズコントローラに Ethernet 経由で接続します。

3　Sysmac Studio を、NY シリーズコントローラ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NY シリーズコントローラ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　ダウンロード後、設定した IP アドレスが、以下のように NY シリーズコントローラに反映されます。

- **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コントローラに反映されます。
- **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 IP アドレスが自動的にコントローラに反映されます。
  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コントローラに反映されます。

- **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コントローラに反映され、「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、NY 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場合、[TCP/IP] ダイアログでは、[BOOTP サーバから取得した IP アドレスで固定設定] が選択された状態になっています。
  BOOTP サーバからの取得を止める場合、[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。
- 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、「BOOTP サーバから取得した IP アドレスで固定設定」のままとなります。
- 「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、内蔵 EtherNet/IP ポートの IP アドレス設定は自動的に「固定設定」となります。
  よって、Sysmac Studio でプログラムの照合を行うと、不一致と判断されます。

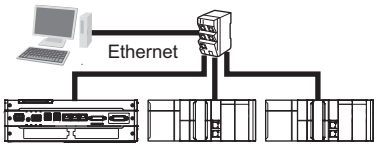
4-2-3　オンライン接続する

Sysmac Studio からコントローラにオンライン接続します。

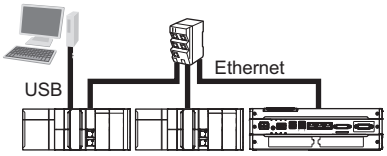
● コントローラと Sysmac Studio（パソコン）との接続形態

コントローラと Sysmac Studio（パソコン）は、Ethernet 経由で、以下のように接続します。

- 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 🕮「2-1-4 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。

- 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 USB ポートを持つ NJ/NX シリーズコントローラが必要です。 ※ 接続にはスイッチングハブが必要です。 詳細は 🕮「2-1-4 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。
- Sysmac Studio を、内蔵 EtherNet/IP ポート経由でオンライン接続した場合、接続している内蔵 EtherNet/IP ポートの IP アドレスを変更すると、Sysmac Studio でタイムアウトとなります。そのような場合は、Sysmac Studio で一旦オフライン状態にして、接続している内蔵 EtherNet/IP ポートの IP アドレスに変更し、再びオンラインに状態に戻せば、再接続できます。

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、内蔵 EtherNet/IP ポートは、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

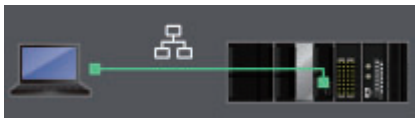
コントローラと Sysmac Studio（パソコン）を、Ethernet 経由で接続後、以下の手順を行います。

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

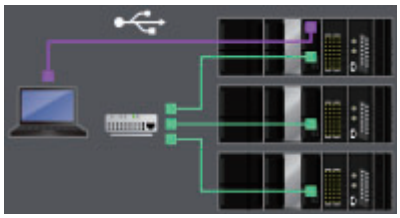
・Ethernet－直接接続



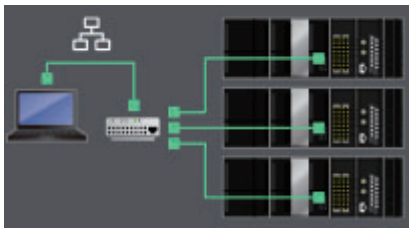
・1：N接続のとき

●Ethernet 接続

・USB リモート接続



・Ethernet－HUB 接続



参考

・設定された IP アドレスに誤りがある場合、イベントログに「IP アドレス設定異常」が記録されます。

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。Sysmac Studio を使用し、Ethernet －直接接続で IP アドレスを設定しなおしてください。
- 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
  - 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）

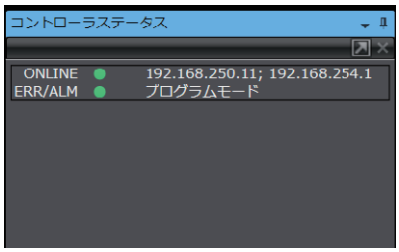
● 保存されたプロジェクトからの接続

設定した接続形態（EtherNet/IP 接続）はプロジェクトに保存されます。
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。

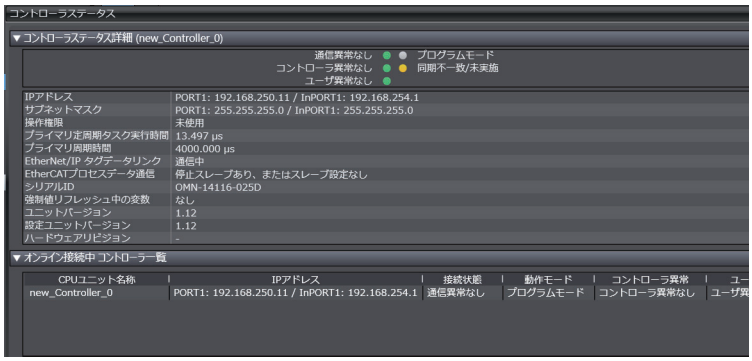
4-2-4　使用中の IP アドレスの確認方法

固定設定または BOOTP サーバから取得した、現在使用中の IP アドレスと Sysmac Studio のステータスウィンドウで確認できます。

- ステータスウィンドウ（基本画面ウィンドウ）



- ステータスウィンドウ（詳細画面ウィンドウ）



参考

- IP アドレスを BOOTP サーバから取得した場合は、コントローラの設定を Sysmac Studio で同期して、アップロードすることにより、BOOTP サーバから取得した IP アドレスを確認することができます。
- 内蔵 EtherNet/IP ポートの IP アドレスが、以下の場合により未登録の場合は”0.0.0.0”と表示されます。
  - BOOTP サーバから未取得

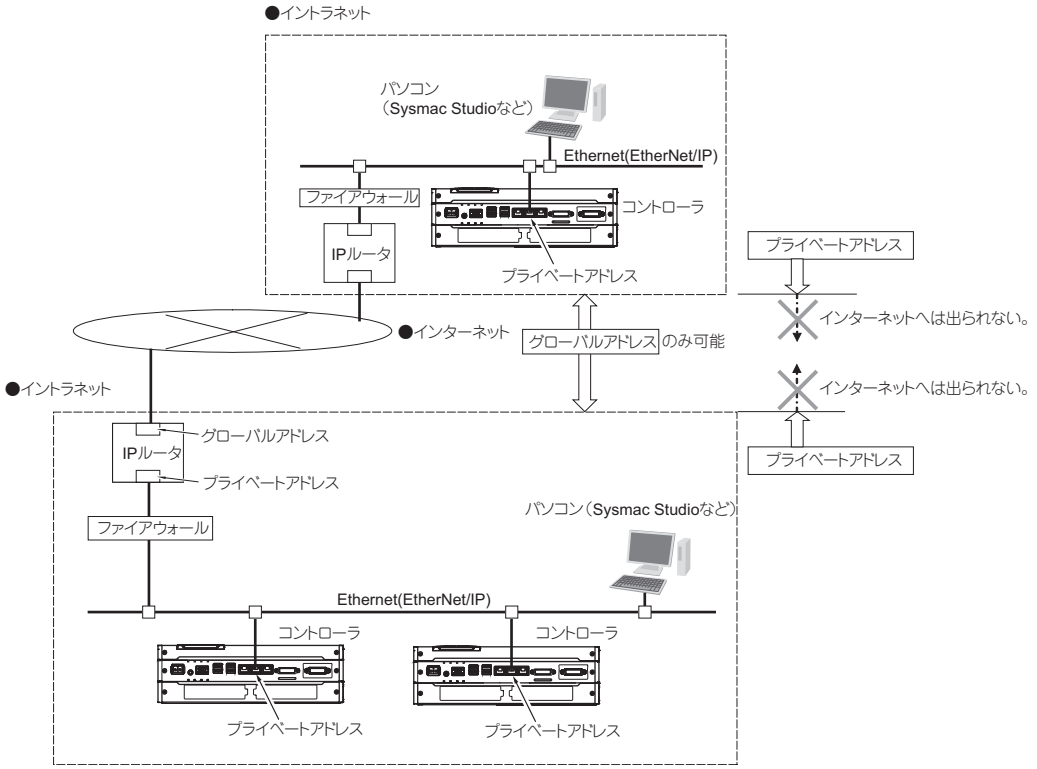
4-3　プライベート／グローバルアドレス

4-3-1　グローバルアドレスとプライベートアドレス

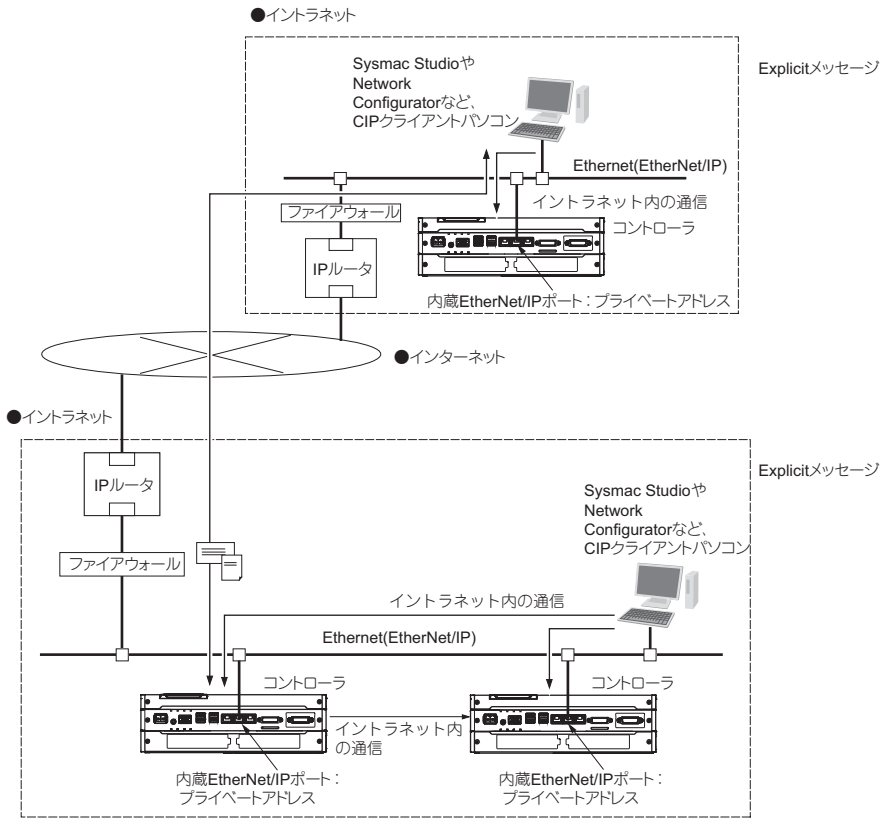
IP アドレスには、以下の 2 種類があります。

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、内蔵 EtherNet/IP ポー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　内蔵 EtherNet/IP ポー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

内蔵 EtherNet/IP ポー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。

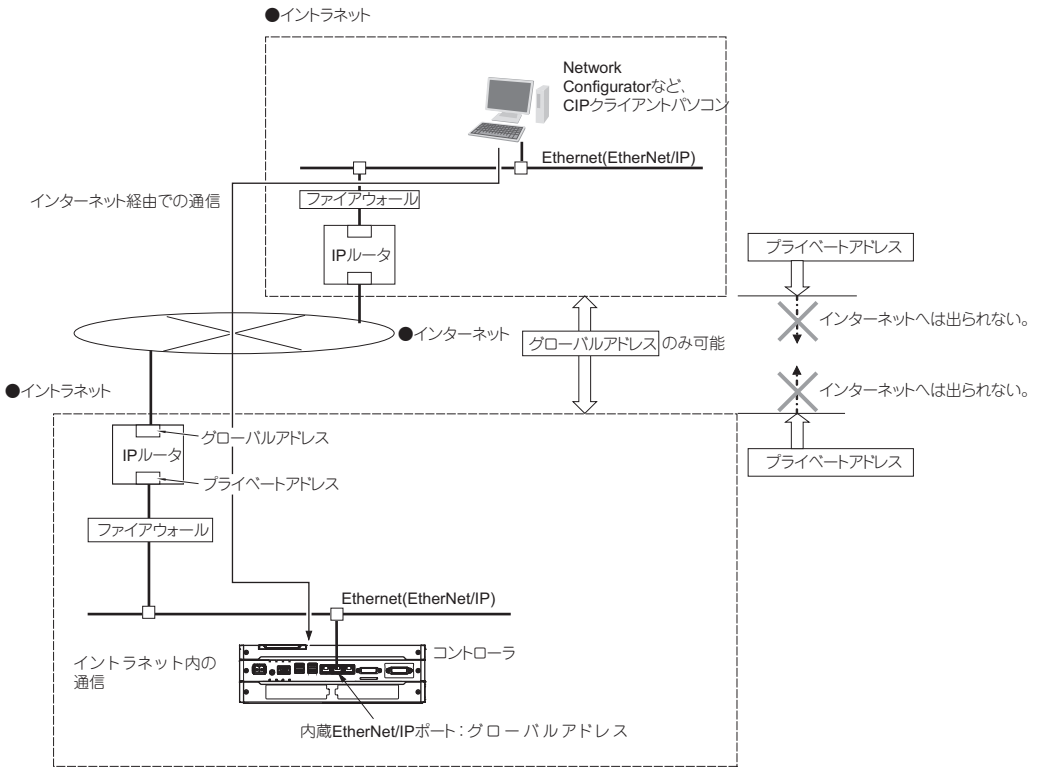
- イントラネット内のプライベートアドレスの内蔵EtherNet/IPポート間でのみ、Explicitメッセージ通信サービスが可能です。
- パソコン（Network Configurator を含む CIP アプリケーション）などから、インターネットを介して、プライベートアドレスの内蔵 EtherNet/IP ポートに対してオンライン接続して通信することはできません。
  また、インターネットを介して、プライベートアドレスの内蔵 EtherNet/IP ポート同士の Explicit メッセージ通信もできません。

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　内蔵 EtherNet/IP ポー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内蔵 EtherNet/IP ポー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

内蔵 EtherNet/IP ポートにグローバル 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 内蔵 EtherNet/IP ポート設定一覧

5-1 ［TCP/IP 設定］ダイアログ 5-2
5-2 ［LINK 設定］ダイアログ 5-6
5-3 ［FTP 設定］ダイアログ 5-7
5-4 ［SNMP 設定］ダイアログ 5-8
5-4 ［SNMP 設定］ダイアログ 5-8
5-5 ［SNMP トラップ設定］ダイアログ 5-10
5-6 ［CIP 設定］ダイアログ 5-12

5-1　[TCP/IP 設定] ダイアログ



● IP アドレス - EtherNet/IP ポート

項目名	内容	デフォルト値
IP アドレス設定方法	内蔵 EtherNet/IP ポートの IP アドレスの設定方法を次のうちから選択します。 • 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*1	内蔵 EtherNet/IP ポートの自 IP アドレスを設定します。*2	192.168.250.1
サブネットマスク *2	内蔵 EtherNet/IP ポートのサブネットマスクを設定します。	255.255.255.0

*1.「IP アドレス設定方法」を「固定設定」にした場合に、設定が必要です。
*2. IP アドレスの設定についての詳細は、「4-2 コントローラの自 IP アドレス設定方法」を参照してください。

● IP アドレス - 内部通信ポート

項目名	内容	デフォルト値
IP アドレス	内部通信ポートの自 IP アドレスを設定します。*1	192.168.254.1
サブネットマスク *1	内部通信ポートのサブネットマスクを設定します。	255.255.255.0

*1. IP アドレスの設定についての詳細は、「4-2 コントローラの自 IP アドレス設定方法」を参照してください。

使用上の注意

内蔵 EtherNet/IP ポートと内部通信ポートが同一ネットワークとなる IP アドレスは設定できません。

● 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	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	内蔵 EtherNet/IP ポートが属するドメインのドメイン名を設定します。（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。

● ホスト名 -IP アドレス

項目名	内容	デフォルト値
ホスト名	ホスト名で通信相手を指定するときに本設定に従いアドレス変換します。 DNS 設定の使用有無に関わらず、設定できます。 ホスト名は最大 6 つ設定できます。 （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
IP アドレス	登録したホスト名の IP アドレスを設定します。	なし

● KeepAlive

項目名	内容	デフォルト値
KeepAlive	ソケットサービス／ FTP サーバ機能／ Sysmac Studio ／ TCP などの接続先のサーバ、およびクライアントがもつ相手ノード生存確認機能を使用するか、しないかを設定します。 ここで「KeepAlive」の「使用する」を選択している場合、［設定］で設定した「KeepAlive 監視時間」を超えて相手ノードが無応答の場合、そのコネクション No. を切断します。 相手ノードが無断で電源 OFF したりすると、コネクションが開設状態のまま放置されることになるので、なるべく「使用する」としてください。 • 使用する • 使用しない	使用する
KeepAlive 監視時間	「KeepAlive」の「使用する」を選択している場合、［KeepAlive 監視設定］で設定した時間を超えて相手ノードが無応答の場合、コネクションを切断します。 設定範囲：1 ～ 65535［秒］	300
Linger オプション	ソケットサービス、TCP 接続時に、Linger オプションを指定するかしないかを設定します。 Linger 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. 解放までの時間（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実行されます。 • 指定する • 指定しない	指定しない

● デフォルトゲートウェイ

項目名	内容	デフォルト値
デフォルトゲートウェイ	内蔵 EtherNet/IP ポートのデフォルトゲートウェイの IP アドレスを設定します。*1 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. EtherNet/IP ポート、内部通信ポートともに「固定設定」の場合でも、設定できるデフォルトゲートウェイは 1 つだけです。

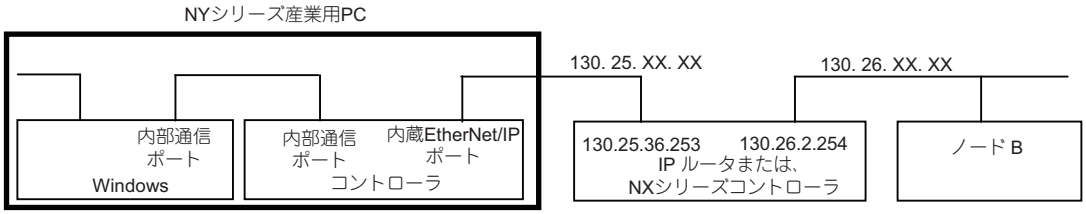
● IP ルータテーブル

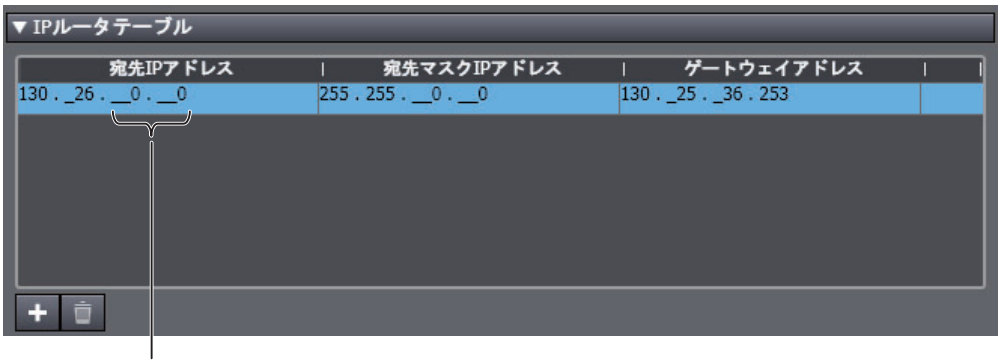
項目名	内容	デフォルト値
宛先アドレス	内蔵 EtherNet/IP ポートが IP ルータを介して他の IP ネットワークセグメント上のノードとタグデータリンクや CIP メッセージ通信をするときに、設定します。また、内蔵 EtherNet/IP ポートの IP ルーティング機能により IP ルータとして使用するときに設定します。 IP アドレスとゲートウェイアドレスの組み合わせは、最大 128 まで設定できます。 IP アドレスのホスト部には、0 を指定します。	なし
宛先マスク IP アドレス		なし
ゲートウェイアドレス		

参考

IP ルータテーブルの設定例

NY シリーズ産業用 PC のコントローラがノード B と、IP ルータを介して、CIP メッセージなどの Ethernet 通信を行う場合、コントローラの 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、コントローラからノード B 宛てに通信命令を実行した場合、コントローラはゲートウェイ IP アドレス（130.25.36.253）宛てにパケットを送信します。





なお、NY シリーズ産業用 PC の Windows がノード B と、NY シリーズ産業用 PC のコントローラおよび IP ルータを介して CIP メッセージなどの Ethernet 通信を行う場合、コントローラの IP ルータテーブルとは別に Windows 側に IP ルーティングするためのデフォルトゲートウェイなどの設定が必要です。

● ポート間転送

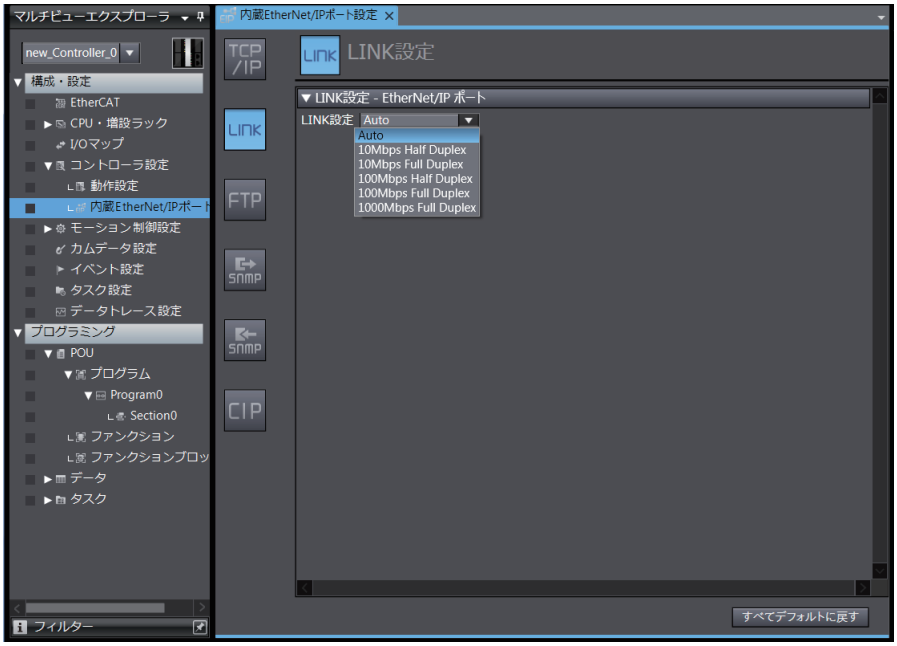
項目名	内容	デフォルト値
IP Forward	コントローラの内蔵 EtherNet/IP ポートと内部通信ポートの間で IP パケットを中継するかしないかを選択します。	する
NAT	Windows から内部通信ポート経由で内蔵 EtherNet/IP ポート側ネットワークにパケットを中継する場合に、送信元 IP アドレスを内蔵 EtherNet/IP ポートの IP アドレスに変換するかしないかを選択します。	する

● PacketFilter

項目名		内容	デフォルト値
PacketFilter		内部通信ポートで受信する IP パケットの条件を設定するかしないかを選択します。	する[*1]
Pass Frame		内部通信ポートで受信を許可する IP パケットの条件を、下記の項目で設定します。最大 32 個の受信を許可する IP パケットの条件を設定することができます。 本設定は、「PacketFilter」の設定を「する」に選択した場合だけ有効です。	-
	指定方法	受信を許可する IP パケットの指定方法を以下から選択します。 ・ IP アドレス指定[*2] ・ any ・ Interface Network[*3]	Interface Network
	IP アドレス[*4]	受信を許可する IP アドレスを設定します。	なし
	サブネットマスク[*4]	受信を許可する IP アドレスのサブネットマスクを設定します。	なし
	プロトコル	any、tcp、udp、igmp、icmp のいずれかを選択します。	any
	範囲指定	「プロトコル」を tcp または udp を選択した場合、受信を許可するポート番号を範囲指定するかどうかを選択します。 ・ チェックあり 　範囲指定する。「ポート A」と「ポート B」の項目で範囲指定する。 ・ チェックなし 　範囲指定しない。「ポート A」の項目で個別指定する。	チェックなし
	ポート A	受信を許可する開始ポート番号を設定します。	なし
	ポート B	受信を許可する終了ポート番号を設定します。「範囲指定」をチェックした場合だけ設定します。	なし

*1. デフォルトでは、内部通信ポートの IP アドレスのネットワーク（Windows 側）から送信された IP パケットの受信だけ許可します。
*2. IP アドレスの指定は、「IP アドレス」および「サブネットマスク」の項目で設定します。
*3. 内部通信ポートの IP アドレスのネットワークであることを示します。「IP アドレス指定」で設定した場合と異なり、内部通信ポートの IP アドレスを本設定後に変更しても、本設定を変更する必要がありません。
*4.「指定方法」が IP アドレス指定の場合だけ設定が必要です。

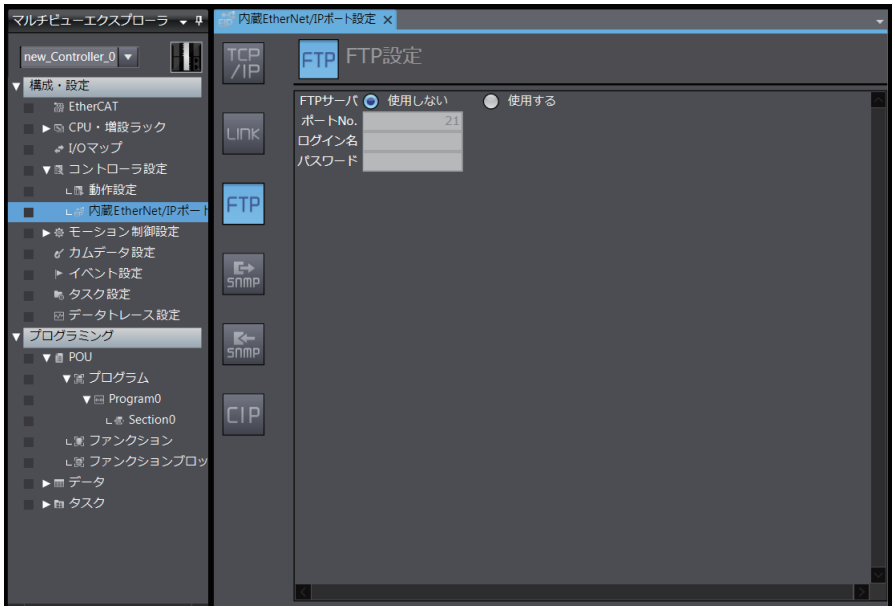
5-2　[LINK 設定] ダイアログ



● LINK 設定 - EtherNet/IP ポート

項目名	内容	デフォルト値
LINK 設定	内蔵 EtherNet/IP ポー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 • 1000Mbps Full Duplex	Auto

5-3　[FTP 設定] ダイアログ



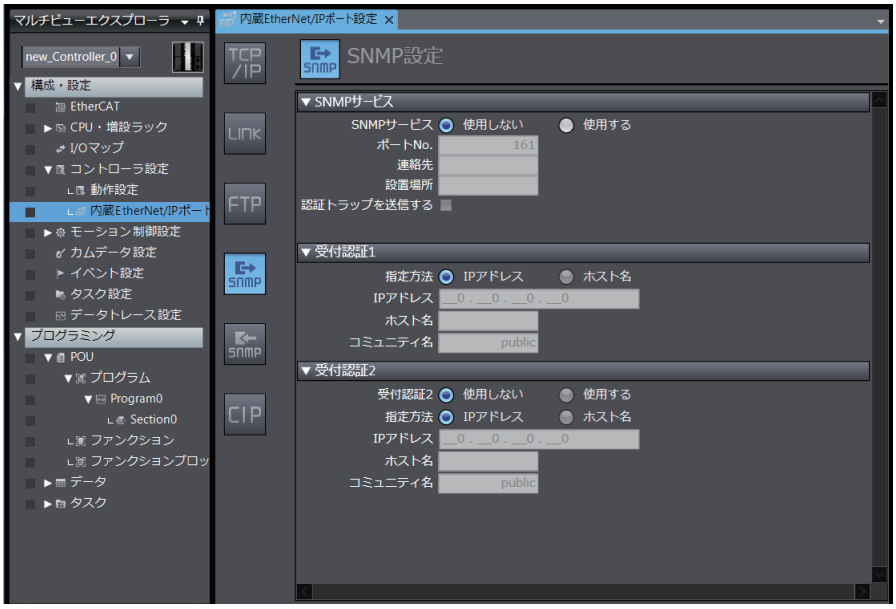
項目名	内容	デフォルト値
FTP サーバ	FTP サーバを使用するか、しないかを選択します。 「使用しない」を選択した場合は、外部からの FTP による接続はできません。	使用しない
ポート No*1,*2	内蔵 EtherNet/IP ポートの FTP ポート No. を設定します。通常、変更する必要はありません。 また、ここで設定できるのは FTP のコントロール用ポートです。データ転送ポートは、ポート No.20 固定となります。	21
ログイン名 *1	外部から内蔵 EtherNet/IP ポートに FTP で接続するためのログイン名を設定します。 (半角英数字 12 文字以下) *3	なし
パスワード *1	外部から内蔵 EtherNet/IP ポートに FTP で接続するためのパスワードを設定します。 (半角英数字 8 ～ 32 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. 20、23、25、80、110、9610、9900、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 10 章 FTP サーバ機能」を参照してください。

5-4　[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。*1 「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.*2	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。	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 255 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）

*1.「使用する」を選択した場合は、下記の受付認証 1 と受付認証 2 の設定が必要です。
*2. ポート No. 25、53、68、110、2222、2223、2224、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 12 章 SNMP（エージェント）機能」を参照してください。

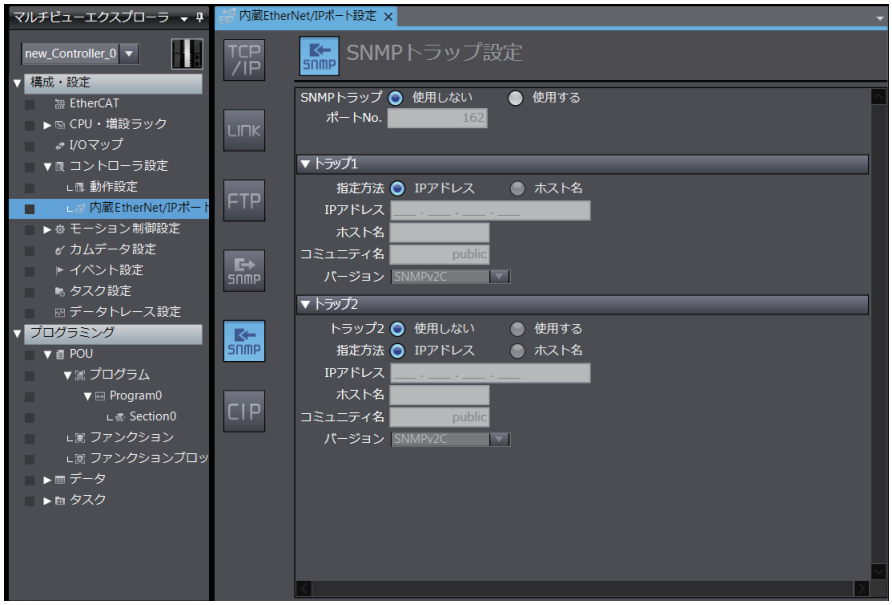
● 受付認証 1

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	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。）	なし
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、内蔵 EtherNet/IP ポー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：255 文字以内）	public

● 受付認証 2

項目名	内容	デフォルト値
受付認証 2	受付認証 2 を使用するかどうかを選択します。 • 使用する • 使用しない	使用しない
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	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。）	なし
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、内蔵 EtherNet/IP ポー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：255 文字以内）	public

5-5　[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。*1 「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.*2	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。	162

*1.「使用する」を選択した場合は、下記のトラップ 1 とトラップ 2 の設定が必要です。
*2. ポート No. 25、53、68、110、2222、2223、2224、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、「12-1-1 SNMP（エージェント）機能とは」を参照してください。

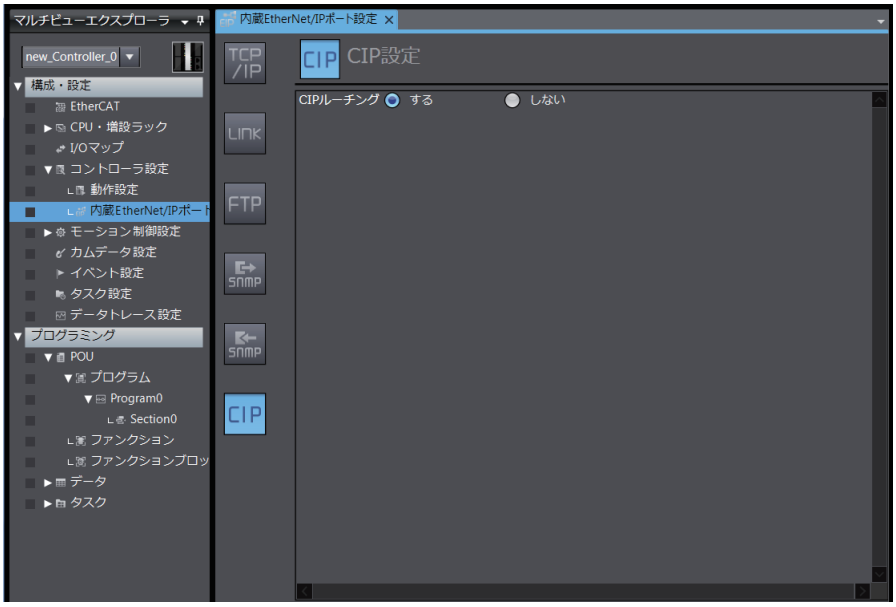
● トラップ 1

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	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。）	なし
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 255 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	SNMPv1

● トラップ 2

項目名	内容	デフォルト値
トラップ 2	トラップ 2 を使用するかどうかを選択します。 • 使用する • 使用しない	使用しない
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	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。）	なし
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 255 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	SNMPv1

5-6　[CIP 設定] ダイアログ



● CIP 設定

項目名	内容	デフォルト値
CIP ルーチング	内蔵 EthetNet/IP ポートまたは内部通信ポートで受信した CIP メッセージを、別のポートへルーチングするか、しないかを選択します。 「しない」を選択した場合は、受信した CIP メッセージは破棄されます。	する

6

通信テスト

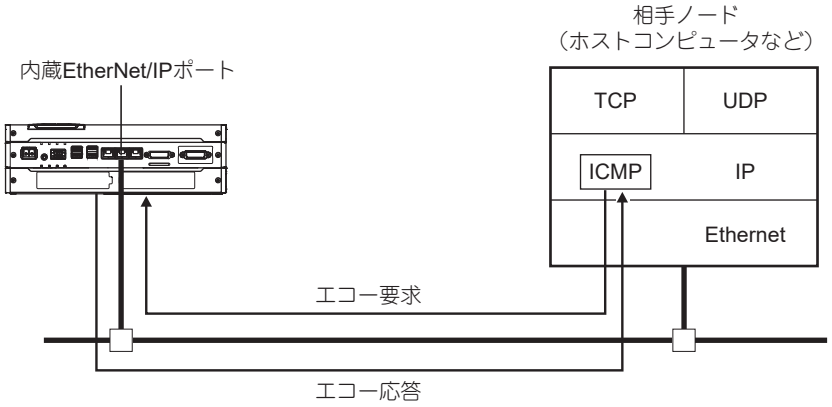
6-1 通信テスト 6-2
6-1-1 PING コマンドとは 6-2
6-1-2 内蔵 EtherNet/IP ポートでの使用方法 6-2
6-1-3 ホストコンピュータでの使用方法 6-3

6-1　通信テスト

内蔵 EtherNet/IP ポートに、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、内蔵 EtherNet/IP ポー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。内蔵 EtherNet/IP ポートは、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　内蔵 EtherNet/IP ポートでの使用方法

内蔵 EtherNet/IP ポートは、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内蔵 EtherNet/IP ポートに対して 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 7-3
7-1-3 タグデータリンクの機能・性能一覧 7-7
7-1-4 使用方法の概要 7-8
7-1-5 タグデータリンクの開始／停止 7-11
7-1-6 コントローラステータス 7-11
7-1-7 タグデータリンクするデータの同期性 7-13

7-2 タグデータリンクの設定 7-17
7-2-1 Network Configurator の起動 7-17
7-2-2 タグデータリンク設定手順 7-19
7-2-3 デバイスの登録 7-20
7-2-4 タグ／タグセットの作成 7-22
7-2-5 コネクション設定 7-35
7-2-6 ウィザードによるコネクションの作成 7-46
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 7-49
7-2-8 Network Configurator のネットワーク接続手順 7-52
7-2-9 タグデータリンクパラメータのダウンロード 7-57
7-2-10 タグデータリンクパラメータのアップロード 7-60
7-2-11 タグデータリンクパラメータの照合 7-63
7-2-12 タグデータリンクの開始／停止 7-66
7-2-13 デバイスパラメータのクリア 7-68
7-2-14 ネットワーク構成ファイルの保存 7-70
7-2-15 ネットワーク構成ファイルの読出 7-72
7-2-16 コネクションのチェック 7-74
7-2-17 デバイスの変更 7-75
7-2-18 デバイスステータス表示 7-76

7-3 タグデータリンクのラダープログラム 7-77
7-3-1 タグデータリンクでのラダープログラム 7-77
7-3-2 タグデータリンクに関連するステータス 7-81

7-4 他機種とのタグデータリンク 7-82

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です。タグには変数を割り当てます。

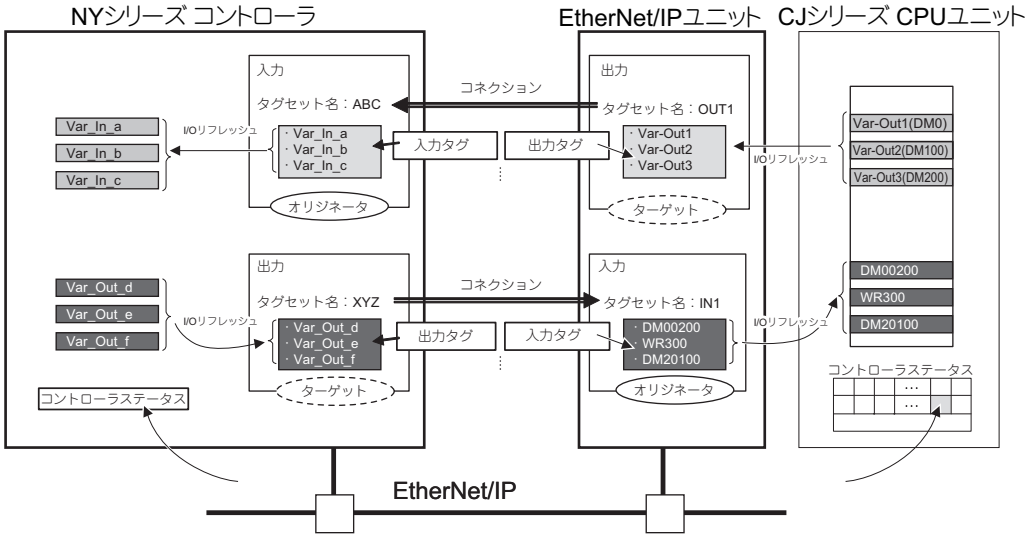
タグデータリンクの設定は Network Configurator で行います。設定方法の詳細については、🕮「7-2 タグデータリンクの設定」を参照してください。

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。



コントローラー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）側コントローラの内蔵 EtherNet/IP ポートに設定します。

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」となる内蔵 EtherNet/IP ポートに設定します。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」を参照ください。

タグセットとは

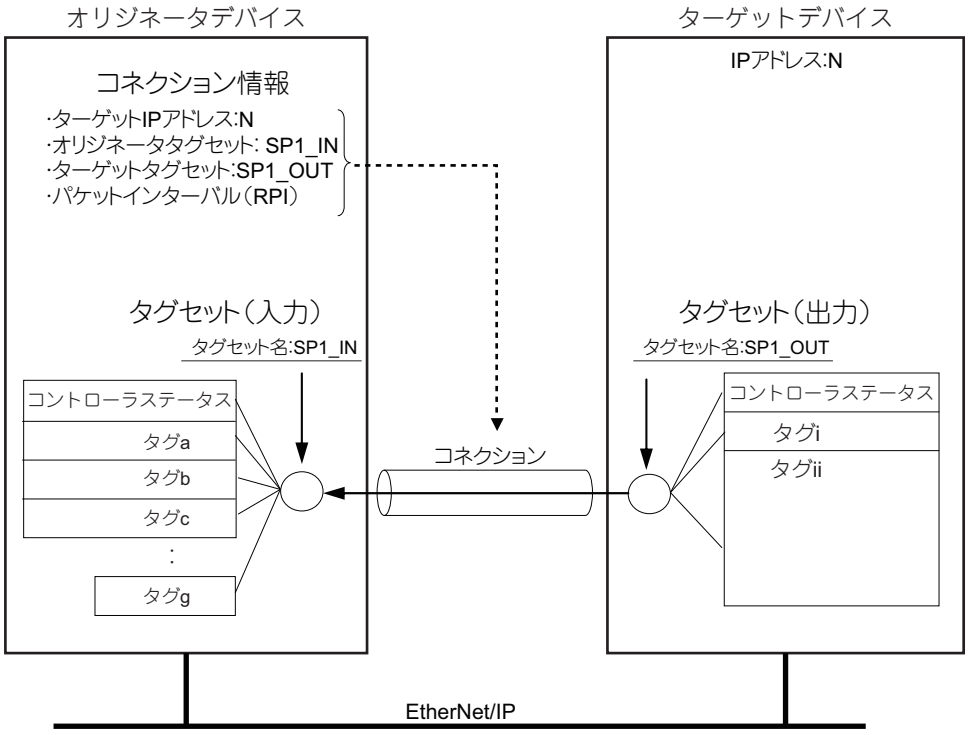
タグデータリンクでは、コネクションを張るときに、1 つまたは複数のタグ（コントローラステータスを含め、最大 8 個）をまとめて、タグのセットを構成します。これを「タグセット」と呼びます。
つまり、「タグセット」とは、タグデータリンクでコネクションを張る、データの単位となります。
タグデータリンクは、タグセットとタグセット間をコネクションで結びつけることで行います。タグセットには必ずタグセット名を設定します。
（注）コネクションとは、データの同時性を確保する、データ交換の単位です。
したがって、「タグセット」は、1 個または複数のタグのデータ交換時の、通信上のデータの同時性を確保する単位となりま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

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。

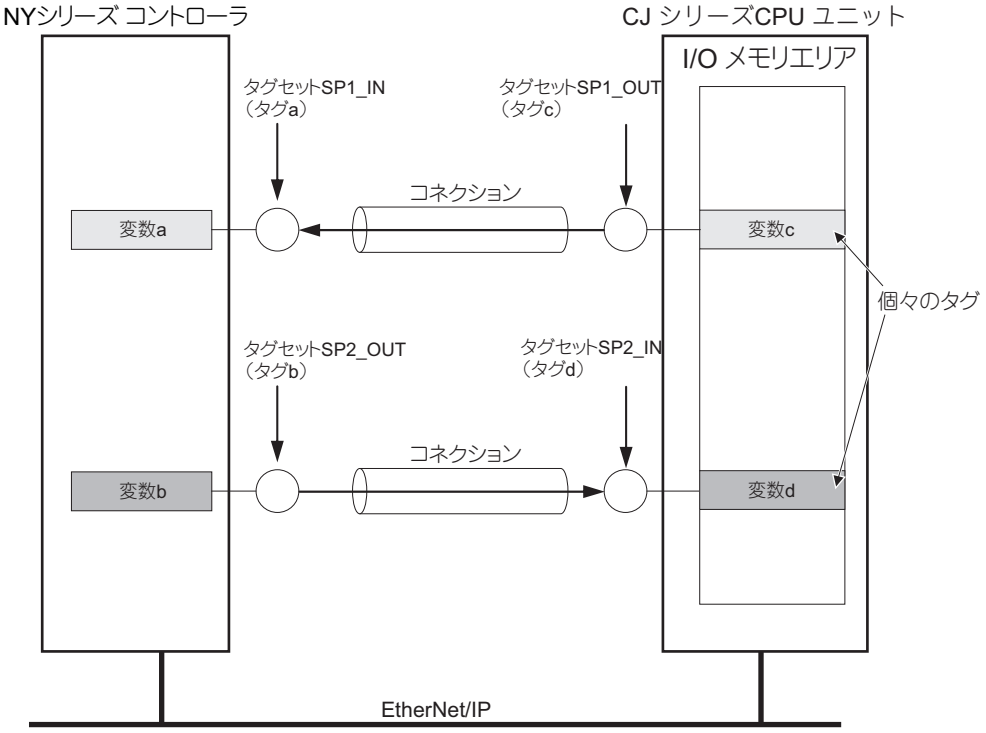
● タグセットには 1 つか複数のタグを使用

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。

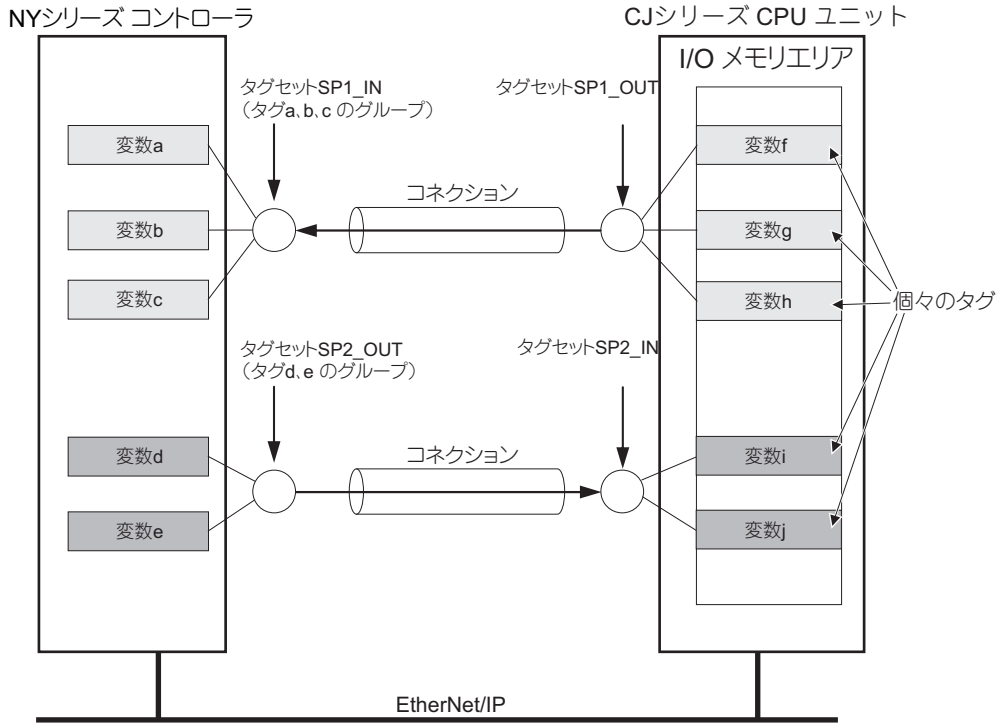
・1 個のタグが 1 個のタグセットの場合

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

NY シリーズコントローラ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 形 NY5□2-□□□□ ユニット Ver.1.12 以降
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、内蔵 EtherNet/IP ポー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	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。(*1)、(*2) • 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（*3）
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	128 個
コネクション		1 ユニットあたり最大コネクション数：128 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)		1 ～ 10000ms（1ms 単位） コネクションごとに設定可能。
ユニット許容通信帯域幅 (PPS)		20000pps 注：ハートビートを含む

*1 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。

*2 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	可
	メンバ	eee.yyy	可

*3 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

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。

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」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 または Sysmac Studio 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。
内蔵 EtherNet/IP ポー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。

参考

下記のタグデータリンクパラメータの設定で、設定可能な数や範囲の仕様は、コントローラのバージョンによって異なります。仕様は、🕮「1-3-1 仕様」を参照してください。

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
コントローラの変数を、入力（受信）タグ、出力（送信）タグとして作成します。
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、コントローラ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。
タグセットには、コントローラの動作状態（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。

● コネクションの設定パラメータ

上記の手順 4 のコネクションの設定では、次の設定パラメータがあります。

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 RPI で回線上のデータ交換が行われます。
内蔵 EtherNet/IP ポートではコネクションごとに、RPI を設定できます。

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。
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、🕮「13-2 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。
- 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。
- 100M ビット /s を超える伝送速度で、マルチキャストを使用したタグデータリンクをする場合は、1000M ビット /s の伝送速度に対応しているスイッチングハブを使用してください。
  もし、同一ネットワーク上に 100M ビット /s 以下で通信している Ethernet 機器があった場合、その Ethernet 機器がタグデータリンクと関係がなくても、タグデータリンク通信に影響を及ぼし、タグデータリンクが切断される可能性があります。

参考

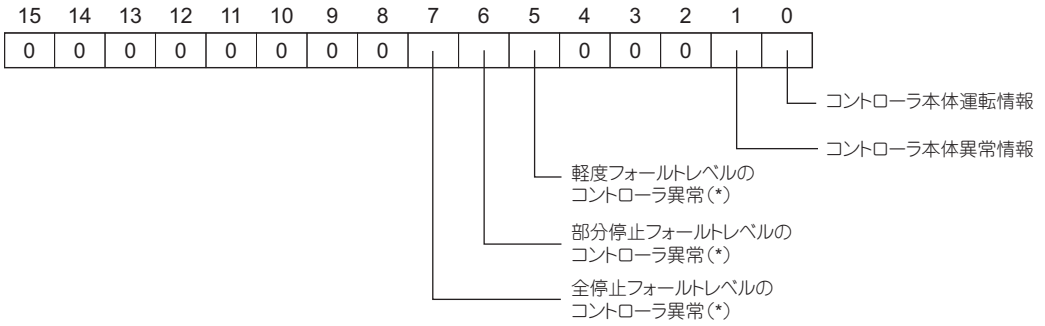
- コネクションタイプごとのコネクション数の数え方は、🕮「13-1-2　コネクション数の計算方法」を参照してください。
- 最大コネクション数を超えた場合、内蔵 EtherNet/IP ポートのコネクション数または、ノード数の見直しが必要です。

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、NY シリーズコントローラの電源が 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、システム定義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

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、コントローラの動作状態（運転情報、異常情報、コントローラ異常の重要度）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。）

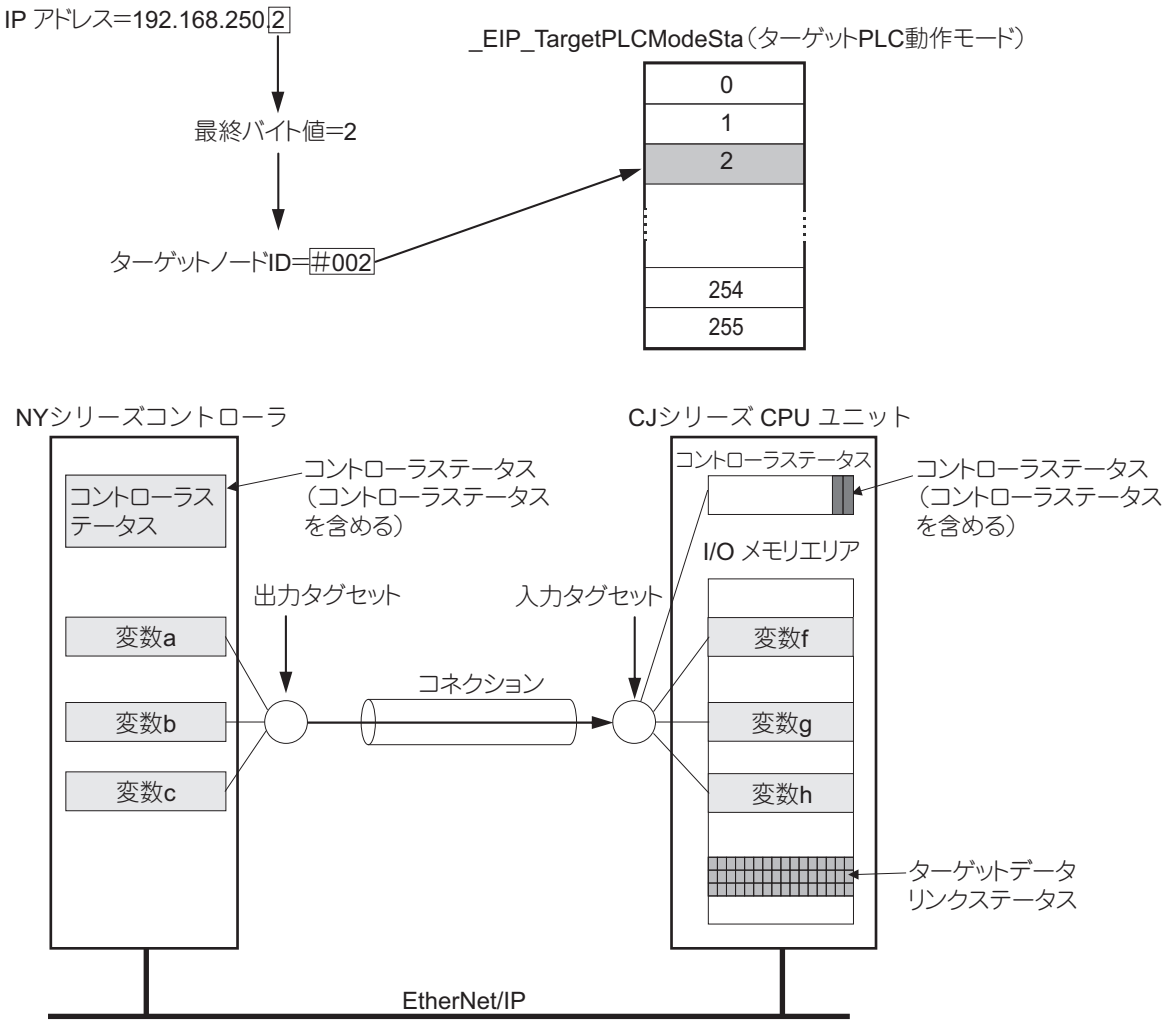


* 発生中の異常の重要度を示すフラグ（b7 ～ b5）は、一番高いレベルの異常のみが TRUE になります。
よって、軽度フォールドレベルと全停止フォールトレベルのコントローラ異常が同時に発生した場合、全停止フォールトレベルのコントローラ異常（b7）は True になりますが、軽度フォールトレベル（b5）は FALSE のままとなります。

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、次に示すシステム定義変数にコントローラステータスの内容が反映されます。

- ターゲット PLC 動作モード
  「_EIP_TargetPLCModeSta」
- ターゲット PLC 異常情報
  「_EIP_TargetPLCErr」

（例）NY シリーズコントローラで、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 PLC 動作モードを送信する場合



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
コントローラ本体運転情報	ターゲット PLC 動作モード：「_EIP_TargetPLCModeSta」	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。 （内蔵 EtherNet/IP ポー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。
コントローラ本体異常情報	ターゲット PLC 異常情報：「_EIP_TargetPLCErr」	該当ターゲットノードのコントローラ本体の異常状態（運転停止、継続異常の論理和）を表します。 （内蔵 EtherNet/IP ポートがオリジネータとなるコネクションの場合） 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、「ターゲット PLC 動作モード」「ターゲット PLC 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

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、🕮『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』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

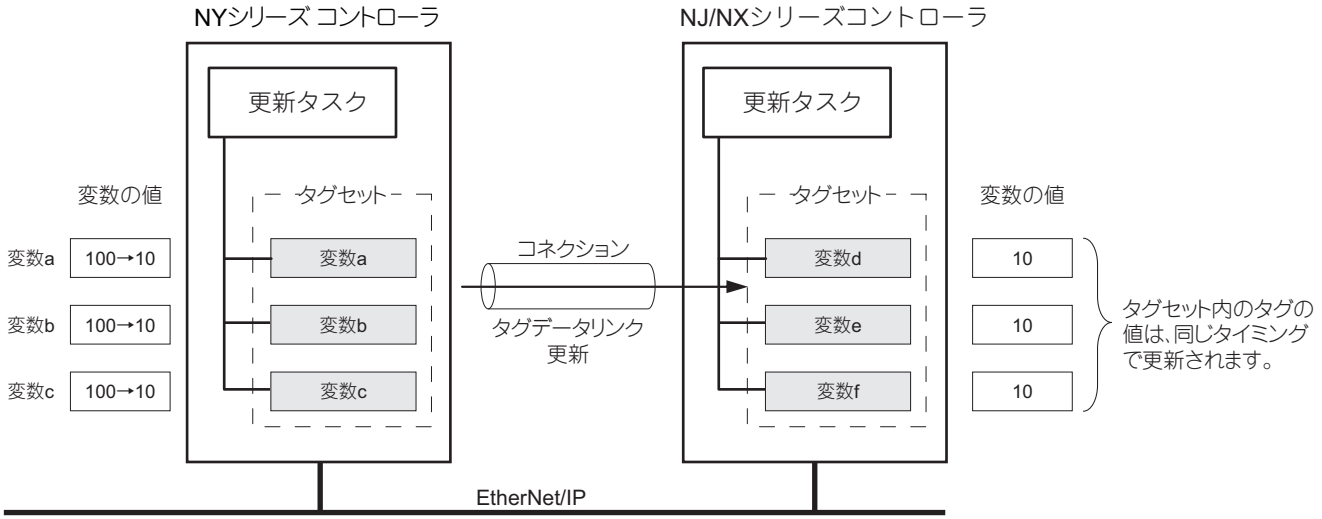
(2) タグに割り付けたネットワーク変数に、更新タスクを設定してあり、その更新タスクの種類が、タグセット内のタグ全て同じ種類であること。(*)

(3) 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

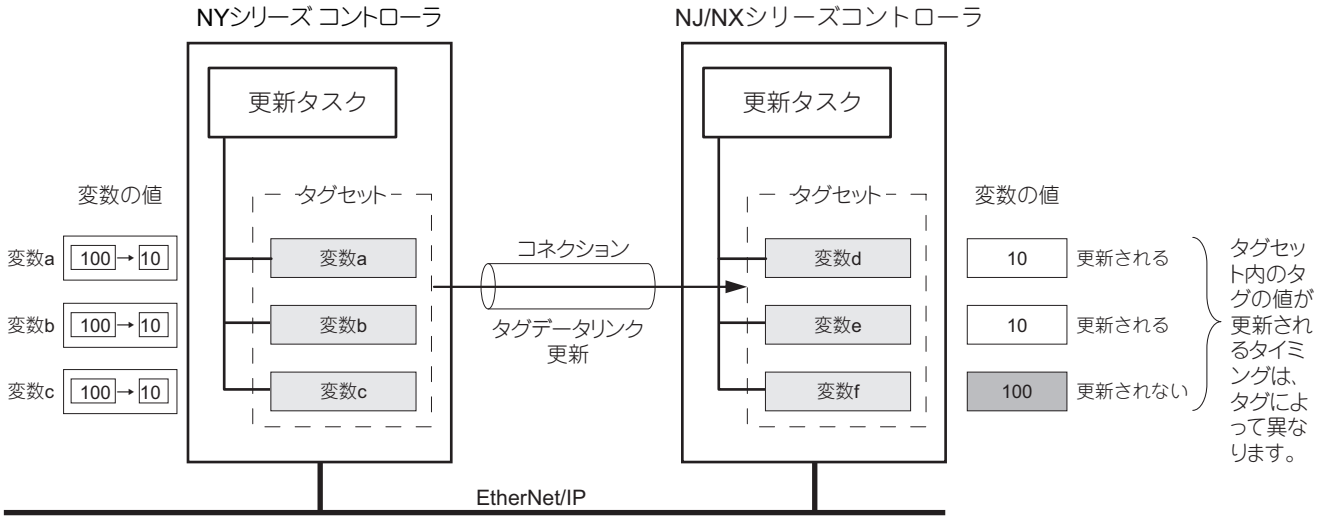
変数アクセス時間とデータ転送に必要な時間については、📖「13-3-3 タスク周期とパケットインターバル（RPI）の関係」を参照してください。

* 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。

- 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。



- 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。



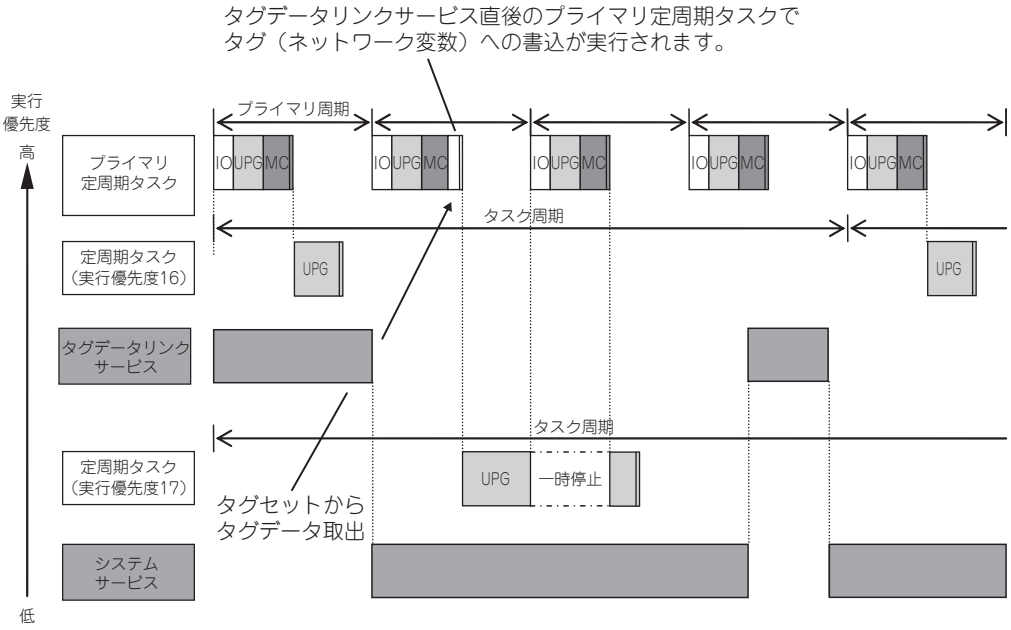
● **ネットワーク変数（タグ）の更新タイミングと、ユーザプログラムの実行周期を同期させる**

ユーザプログラム実行時、そのユーザプログラムがアクセスするネットワーク変数（タグ）の値が、最新のタグデータの値に都度更新されるようにするには、ネットワーク変数（タグ）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、ネットワーク変数（タグ）の更新タスクとして設定します。

ユーザプログラムと同じタスクを更新タスクとして設定したタグ（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）の動作の違いを以下に示します。

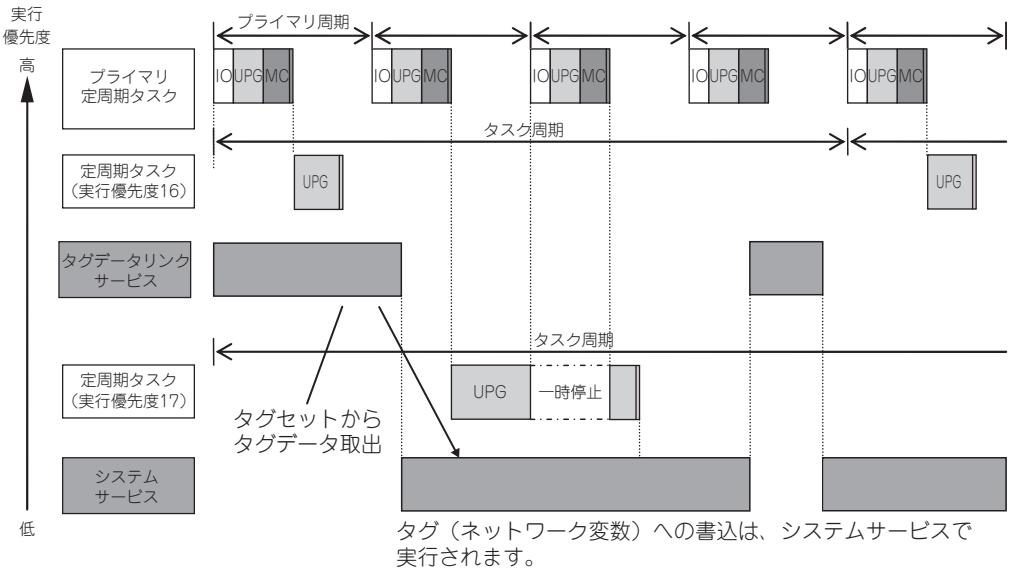
(1) 更新タスクを設定したタグ（ネットワーク変数）

プライマリ定周期タスクを更新タスクに設定している場合の例です。タグ（ネットワーク変数）への書込は、タグデータリンクサービス直後のプライマリ定周期タスクで実行されます。



(2) 更新タスクを設定しないタグ（ネットワーク変数）

タグ（ネットワーク変数）への書込は、システムサービスで実行されます。



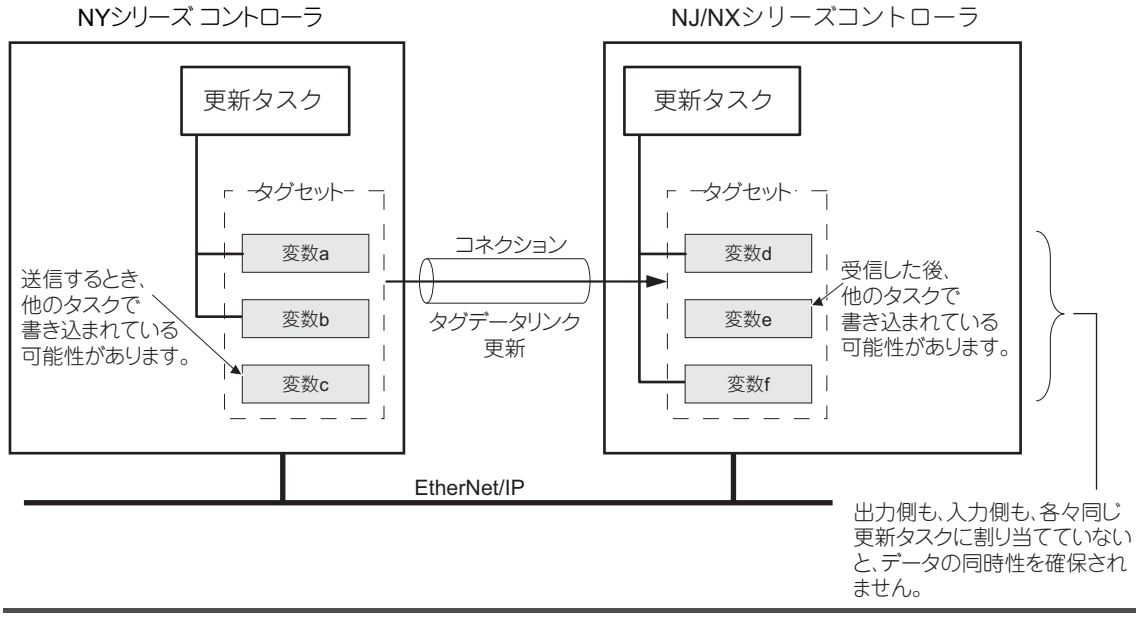
参考

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、出力側のコントローラにおいても、入力側のコントローラにおいても、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

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、🕮「13-3-3 タスク周期とパケットインターバル（RPI）の関係」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] ー [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、🕮『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』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

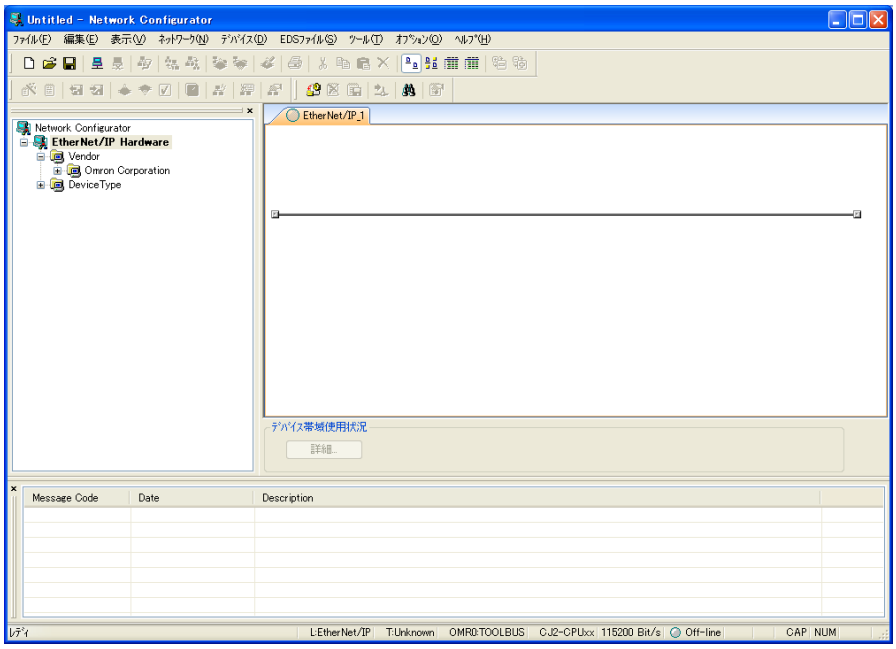
起動方法

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**

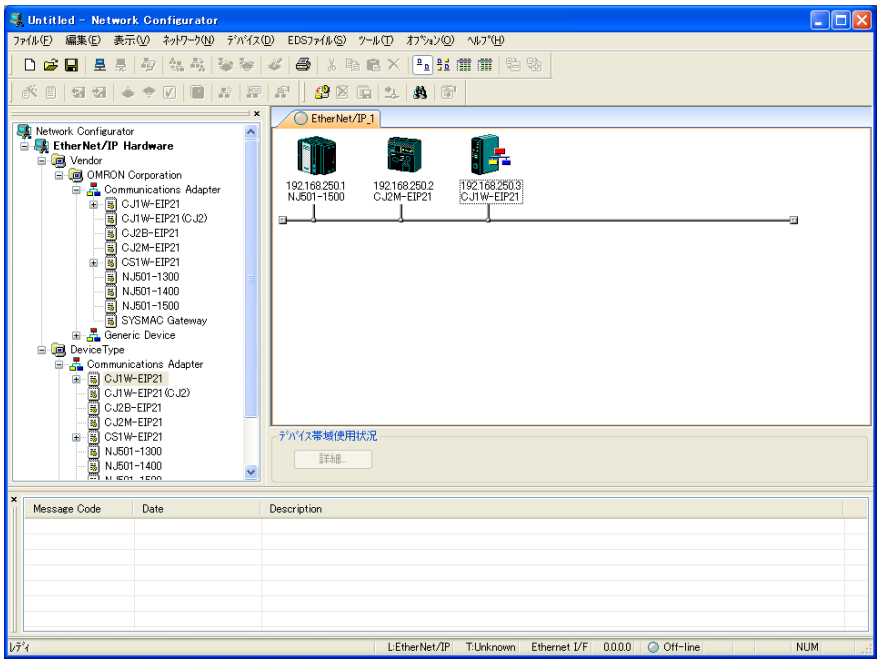
  ［OMRON］｜［Sysmac Studio］｜［Network Configurator for EtherNetIP］｜［Network Configurator］を選択します。

起動すると、以下に示す初期画面が表示されます。

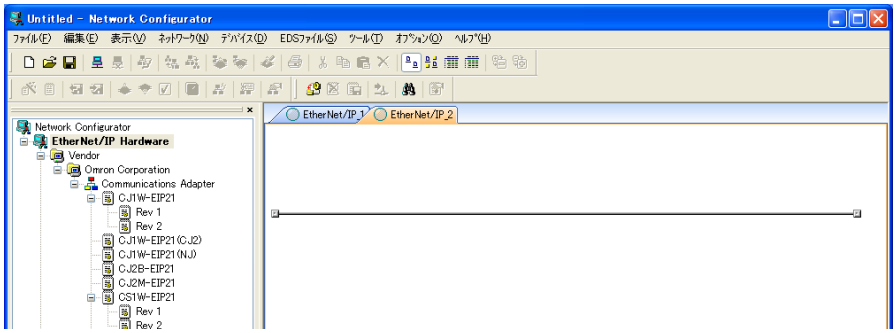


メインウィンドウ

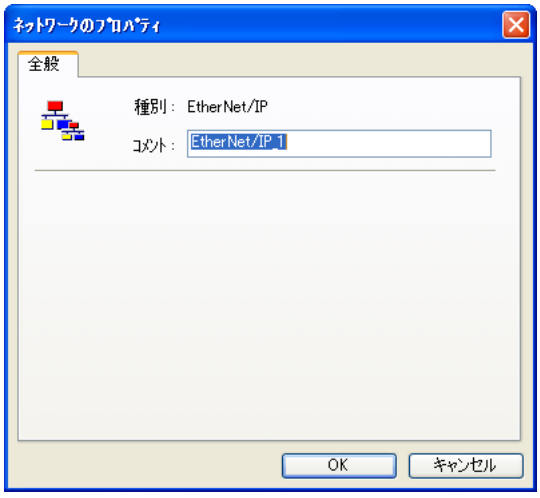
メインウィンドウは、以下のようにハードウェアリストとネットワーク構成ウィンドウで構成されています。



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内蔵 EtherNet/IP ポー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（内蔵 EtherNet/IP ポート）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　内蔵 EtherNet/IP ポートの LED（🕮『NY シリーズ トラブルシューティングマニュアル（SBCA-438）』参照）および Network Configurator のモニタ機能（🕮「14-1 Network Configurator のデバイスモニタ機能」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。

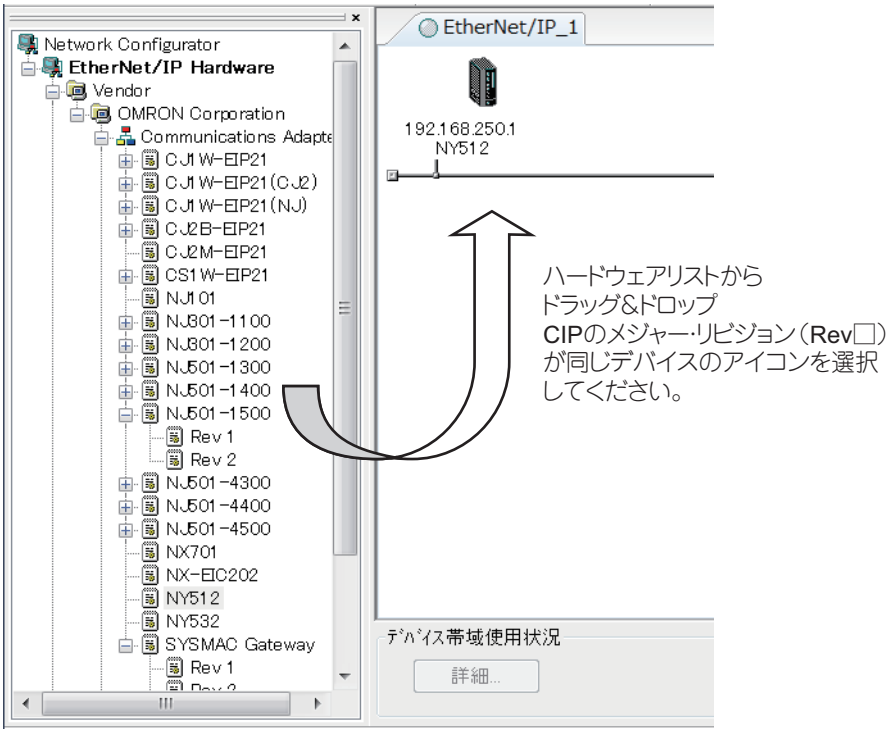
注：使用方法は 🕮『Sysmac Studio Version 1 オペレーションマニュアル（SBCA-470）』を参照してください。

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。
NY シリーズコントローラ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デバイス名	ユニットバージョン	CIP リビジョン	
		メジャー・リビジョン	ハードウェアリスト上のリビジョン名
NY512	ユニット Ver.1.12 以降	2	なし
NY532	ユニット Ver.1.12 以降	2	なし

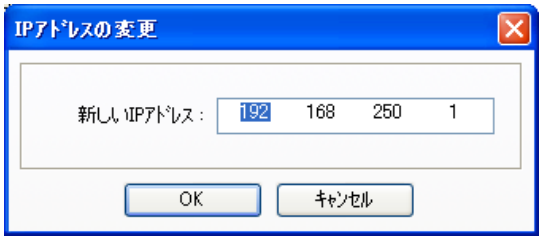
使用上の注意

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。

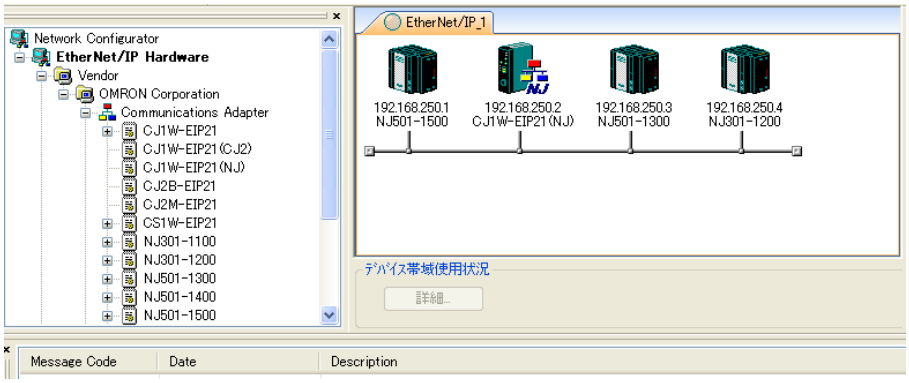
上記は、タグデータリンクパラメータのアップロードおよび照合する場合も同様です。
いずれの場合も、「7-2-17 デバイスの変更」を参照して、デバイスを変更してください。

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［IP 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。**



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

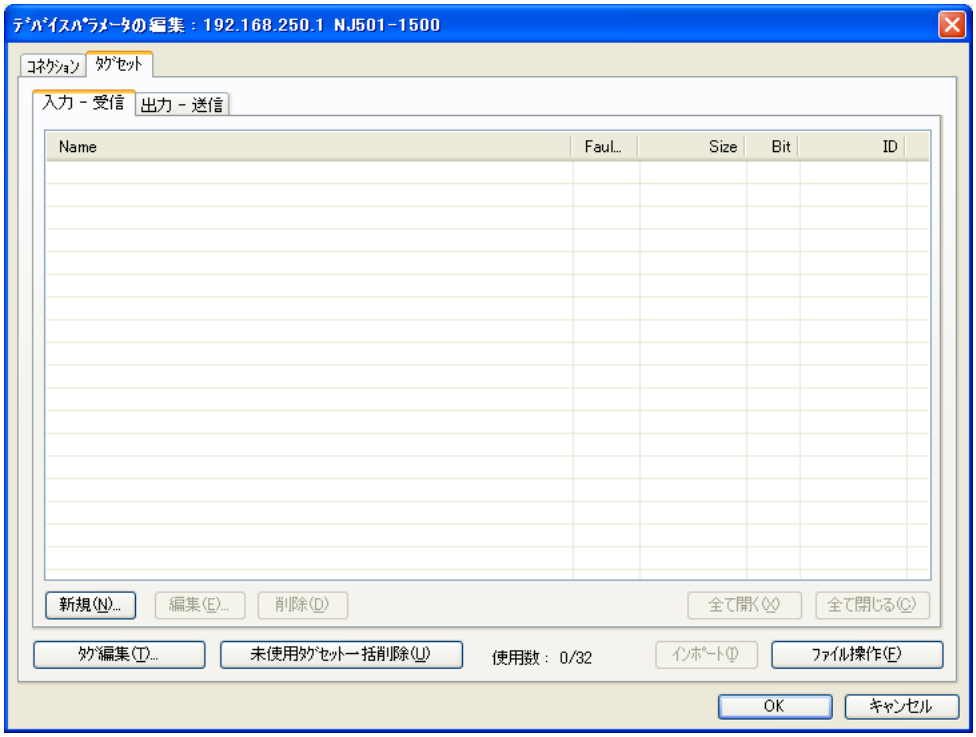
(1) Network Configurator の「デバイスパラメータの編集」機能を使って作成する方法

● 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、または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。

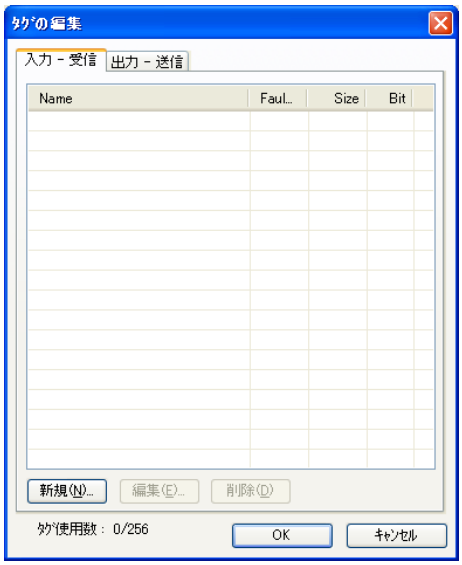


2 **［デバイスパラメータの編集］ダイアログボックスの［タグセット］タブをクリックします。**
タグセットは入力（受信）と出力（送信）、それぞれにタブがあります。

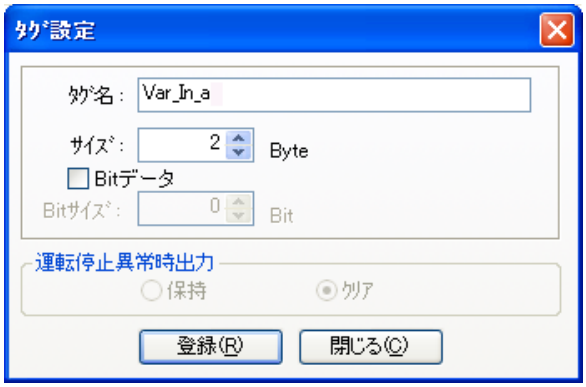


● タグの新規作成／追加

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 **［入力ー受信］タブを選択し［新規］ボタンをクリックします。**
［タグ設定］ダイアログボックスが開きます。



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や共用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）
  - 共用体のメンバ指定
    例：Union.member（メンバを ” .” で区切る）

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_

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。

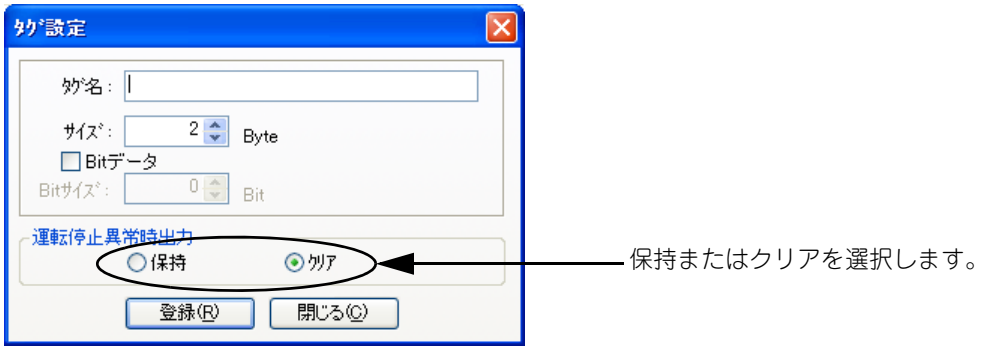
7 **［登録］ボタンをクリックし、タグを登録します。**
タグの登録が一通り終了した後、［閉じる］をクリックします。

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] ダイアログボックス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、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。

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。

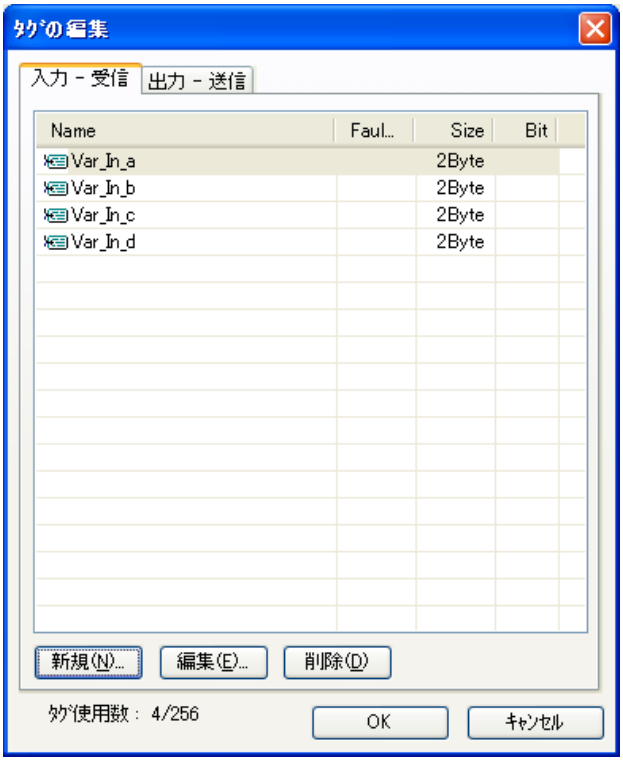


使用上の注意

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] ダイアログボックスの [OK] ボタンをクリックします。**

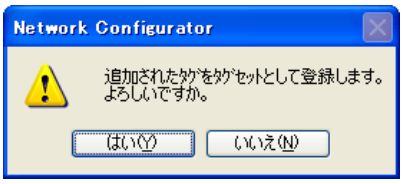


使用上の注意

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「更新タスク」として同じタスクを指定してください。

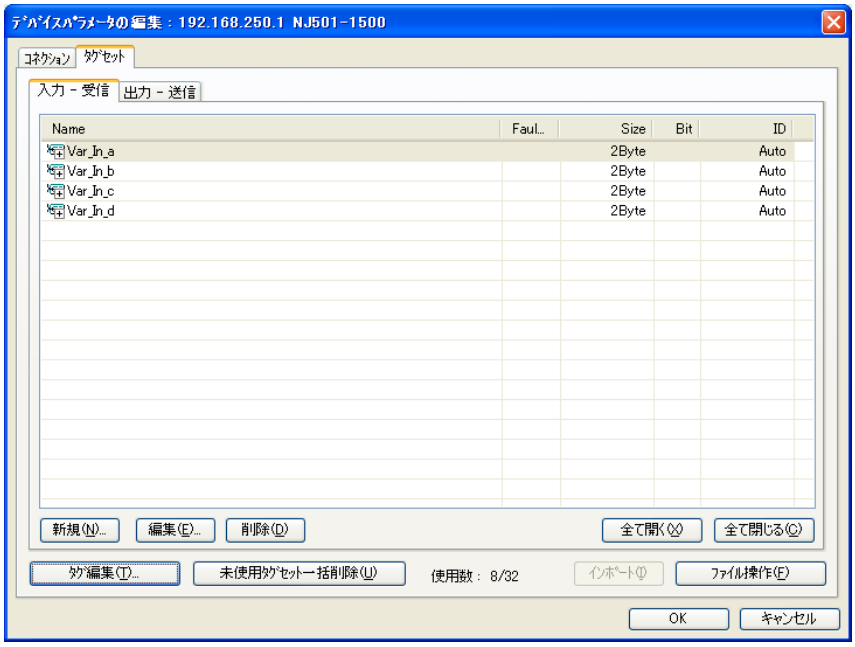
10 **このとき、登録したタグ名をそのままタグセット名として登録するか、確認メッセージが表示されます。タグセットは最大８個までのタグで構成できますが、タグ名をそのままタグセット名として登録した場合、１個のタグ＝１個のタグセットで登録されます。**
ここでは、[はい] をクリックします。



[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。

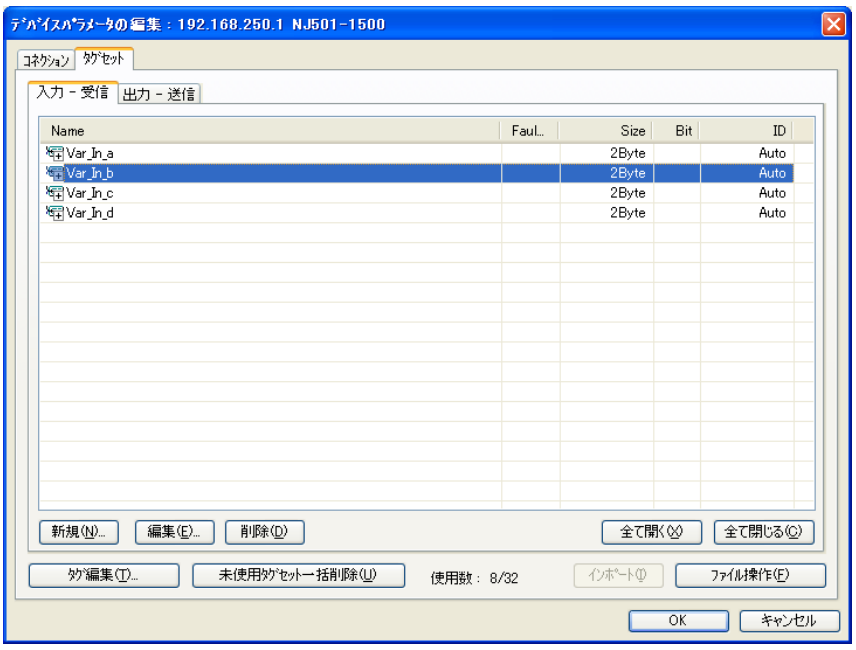
● タグセットの変更と登録

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。**

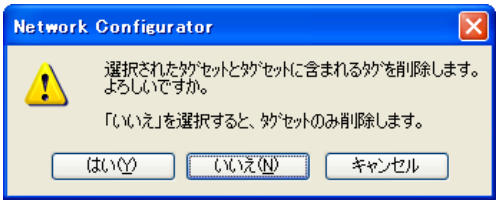

ﾃﾞﾊﾞｲｽﾊﾟﾗﾒｰﾀの編集：192.168.250.1 NJ501-1500
ｺﾈｸｼｮﾝ
ﾀｸﾞｾｯﾄ
入力 - 受信
出力 - 送信
Name
Faul...
Size
Bit
ID
Var_In_a 2Byte Auto
Var_In_b 2Byte Auto
Var_In_c 2Byte Auto
Var_In_d 2Byte Auto
新規(N)...
編集(E)...
削除(D)
全て開く(X)
全て閉じる(C)
ﾀｸﾞ編集(T)...
未使用ﾀｸﾞｾｯﾄ一括削除(U)
使用数： 8/32
ｲﾝﾎﾟｰﾄ(I)
ﾌｧｲﾙ操作(F)
OK
ｷｬﾝｾﾙ

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］ダイアログボックス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。）



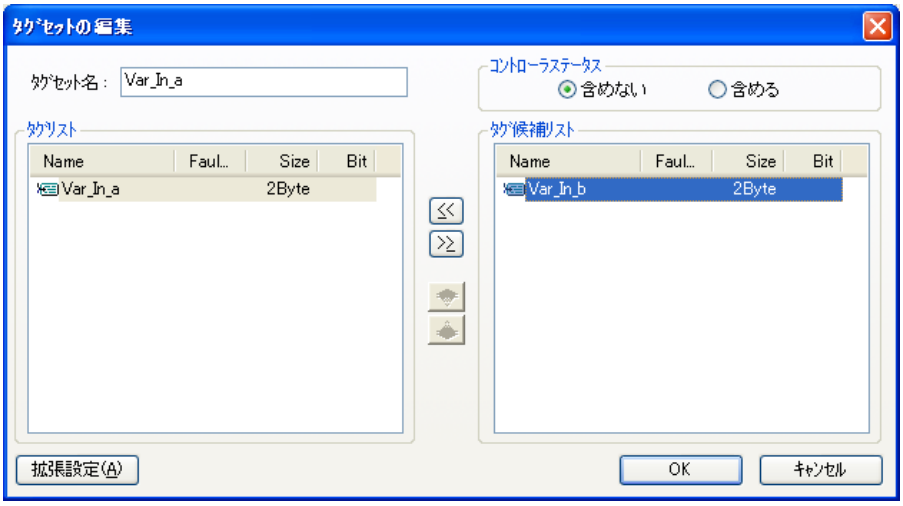
このとき、選択されたタグセットとタグセットに含まれるタグを削除するか確認されます。



［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］ダイアログボックスが開きます。



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。

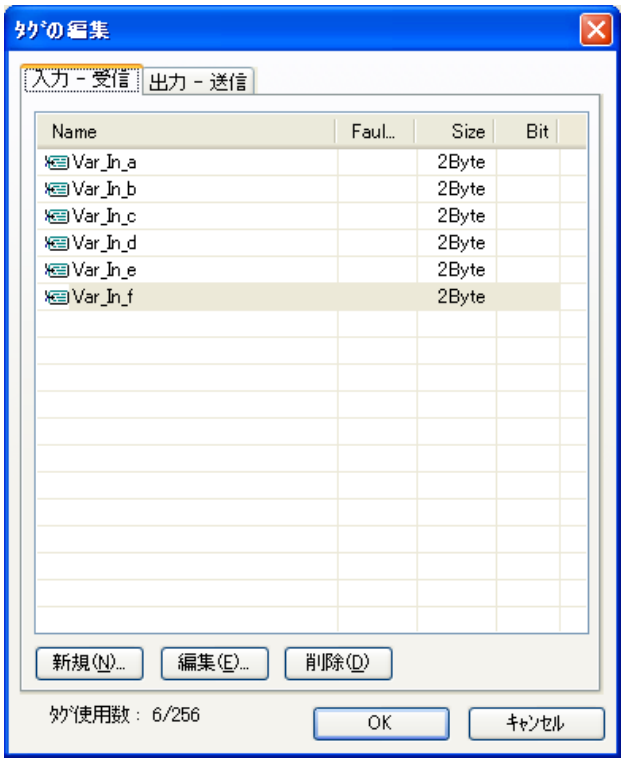
14 **タグセットにコントローラステータスを含める場合は、［タグセットの編集］ダイアログボックス右上の［コントローラステータス］を［含める］にします。**



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。**

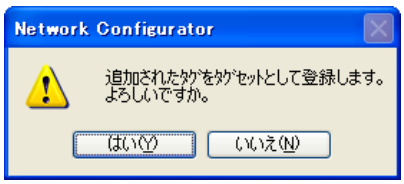
16 **［デバイスパラメータの編集］ダイアログボックスの［OK］ボタンをクリックします。**

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。**

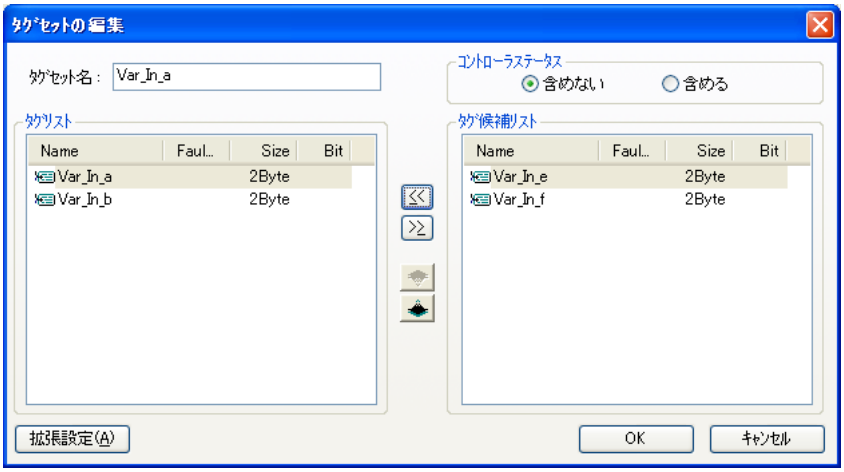


18 **必要なタグの登録が終了したら、［タグの編集］ダイアログボックスの［OK］ボタンをクリックします。**

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。**

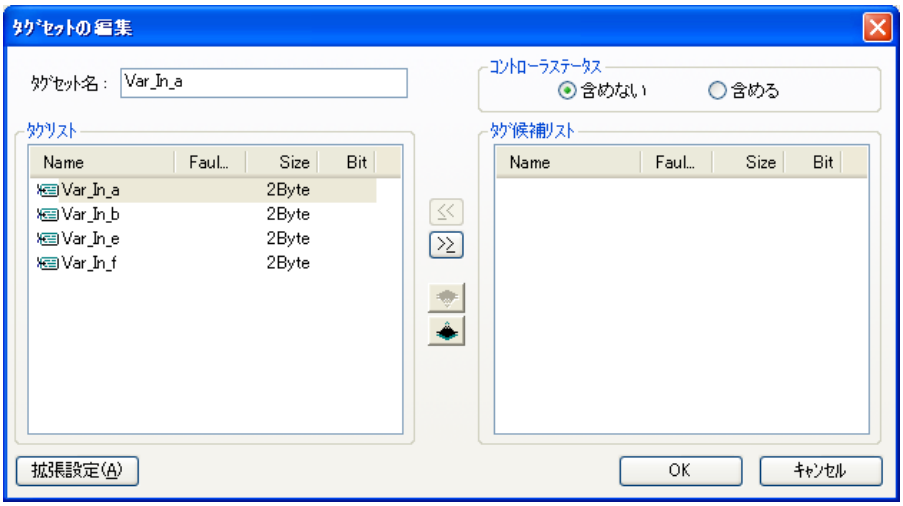


20 **新規に追加したタグをタグセットに登録するには、登録したいタグセットを選択し、ダブルクリックするか、[編集] ボタンをクリックしてください。**



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 のグローバル変数テーブルで、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。**

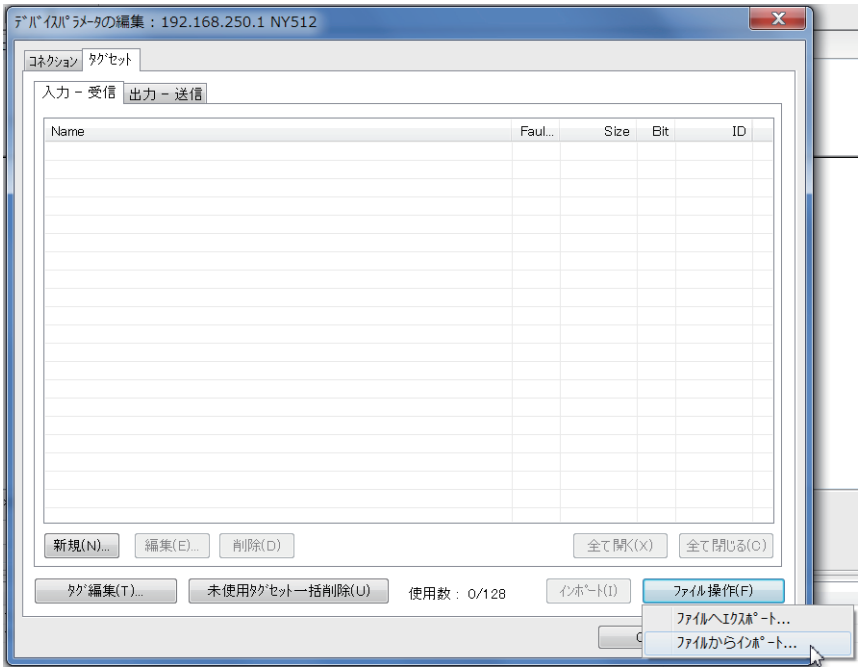
● Network Configurator へのインポート

使用上の注意

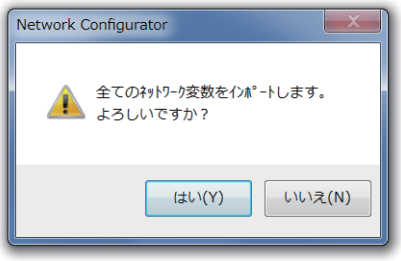
Sysmac Studio でネットワーク属性にした変数のうち、「0000」や「H0000」など、I/O メモリアドレスの表記と同じ変数名の変数は、CSV ファイルにエクスポートされません。
1) 半角数字のみからなる変数名 (例 :001)
2) 以下の半角英字（大文字 / 小文字とも）と、半角数字からなる変数名
- H（例：H30）
- W（例：W30）
- D（例：D100）
- E0_ ～　E18_（例：EA_100）

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｜ファイルからインポート］ボタンをクリックします。**



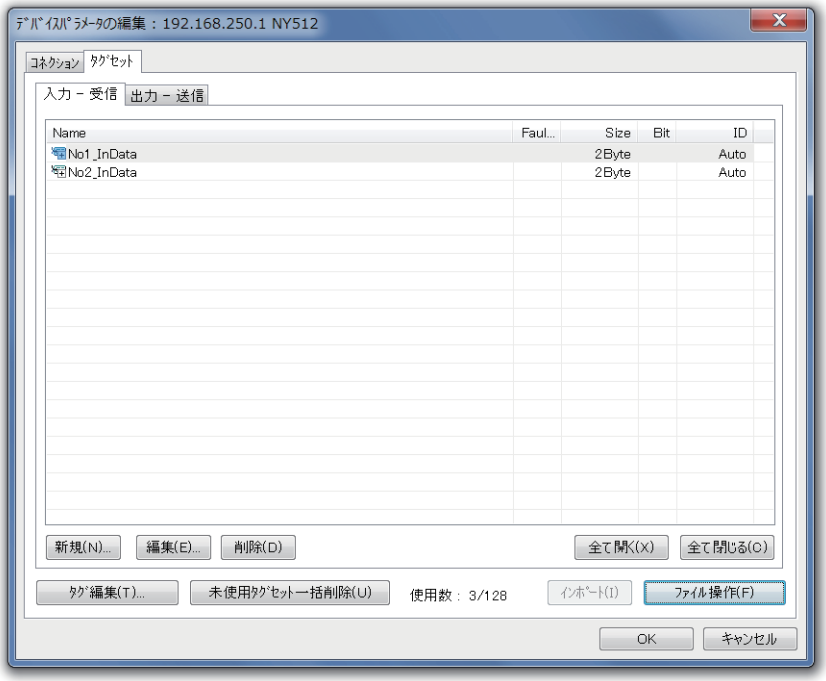
- ネットワーク公開属性にしたすべての変数をタグとしてインポートする場合は、［はい］ボタン、一部のみをインポートする場合は［いいえ］ボタンをクリックします。



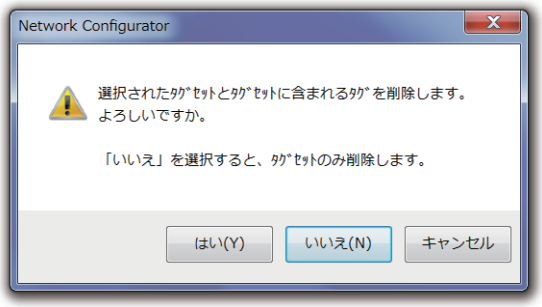
変数をタグにインポート後、タグセットまで自動的に作成する場合は、［はい］ボタン、タグセットは手動で設定する場合は、［いいえ］ボタンをクリックします。



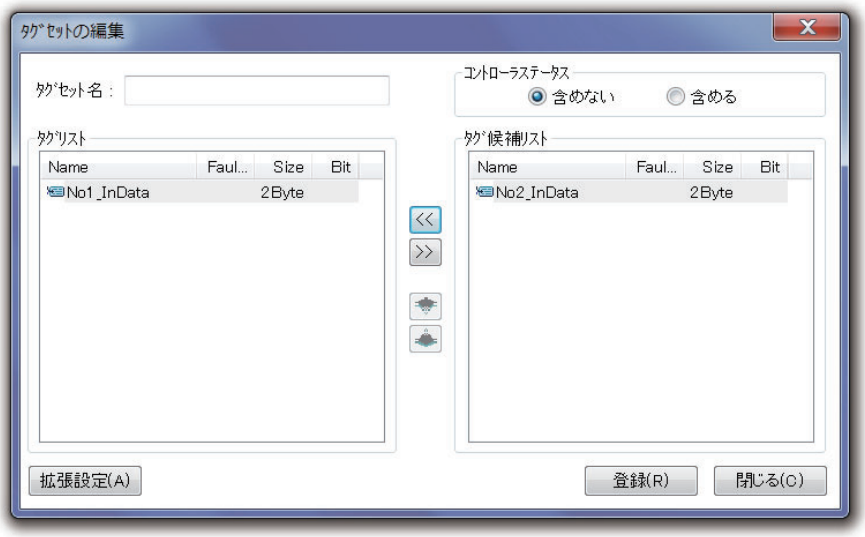
上記で［はい］を選択すると以下に示すように、［タグセット］タブで変数がインポートされ、1 変数＝ 1 タグセットとして、自動的にデバイスパラメータに登録されます（タグセット名は、変数名が割り付けられます）。



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

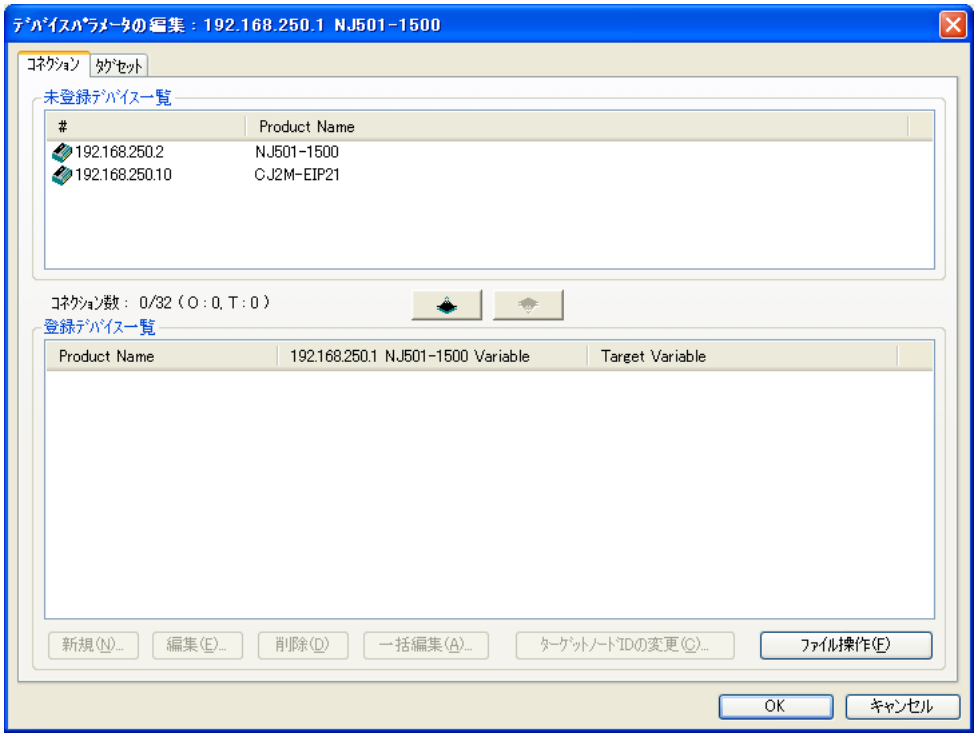
コネクション設定は、タグデータリンクを行う全デバイスのタグセットを作成してから行ってください。

コネクション設定（[コネクション] タブ）

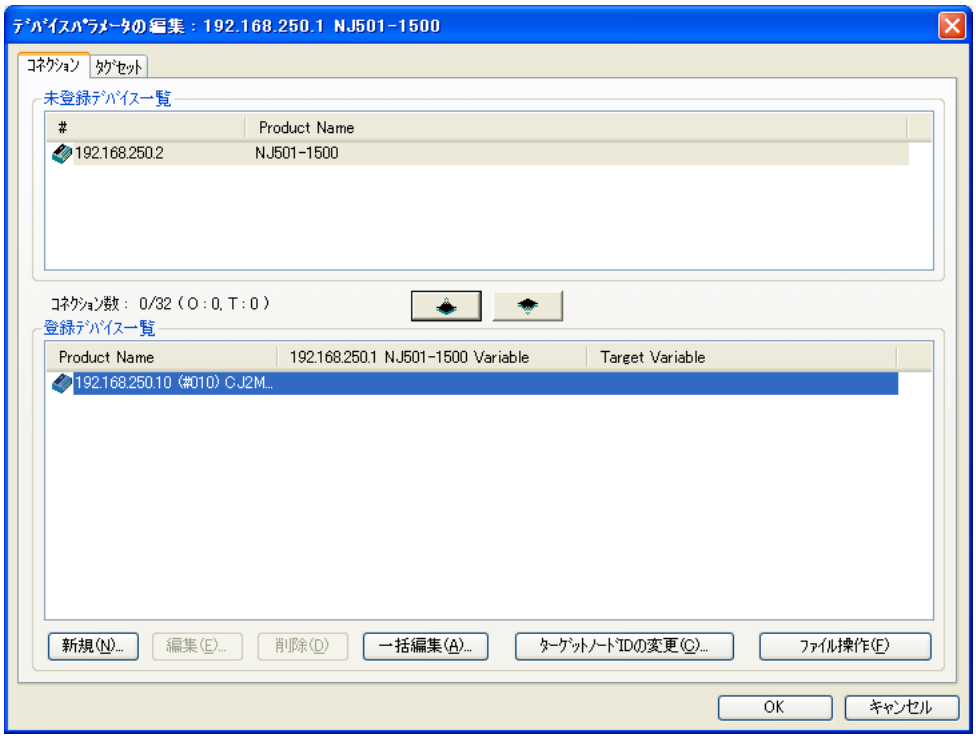
● 登録デバイス一覧への登録

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ターゲットノード PLC 動作モード、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、ターゲットノード ID 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ターゲットノード ID 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。

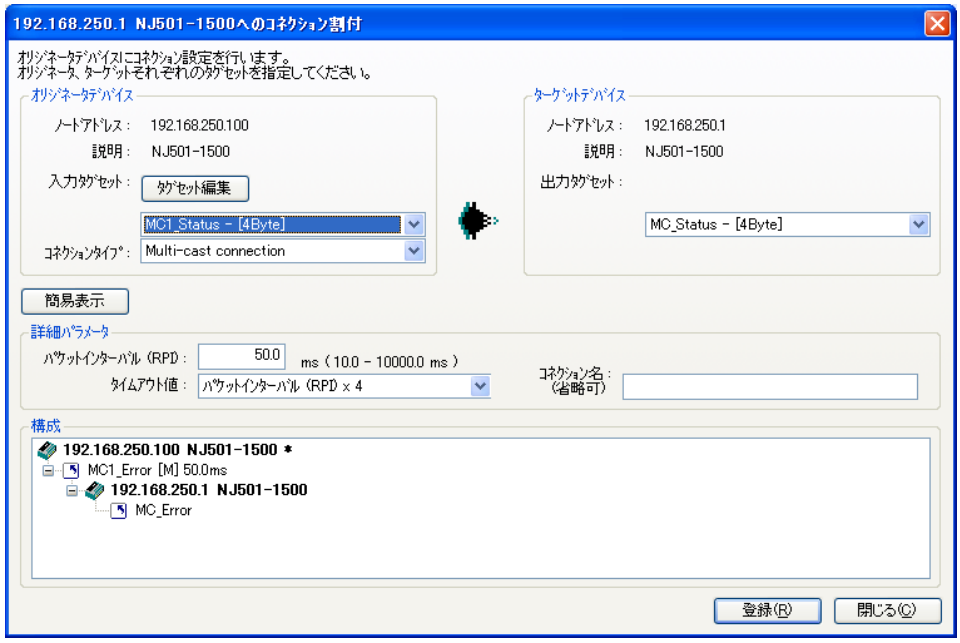


● コネクション設定（個別編集）

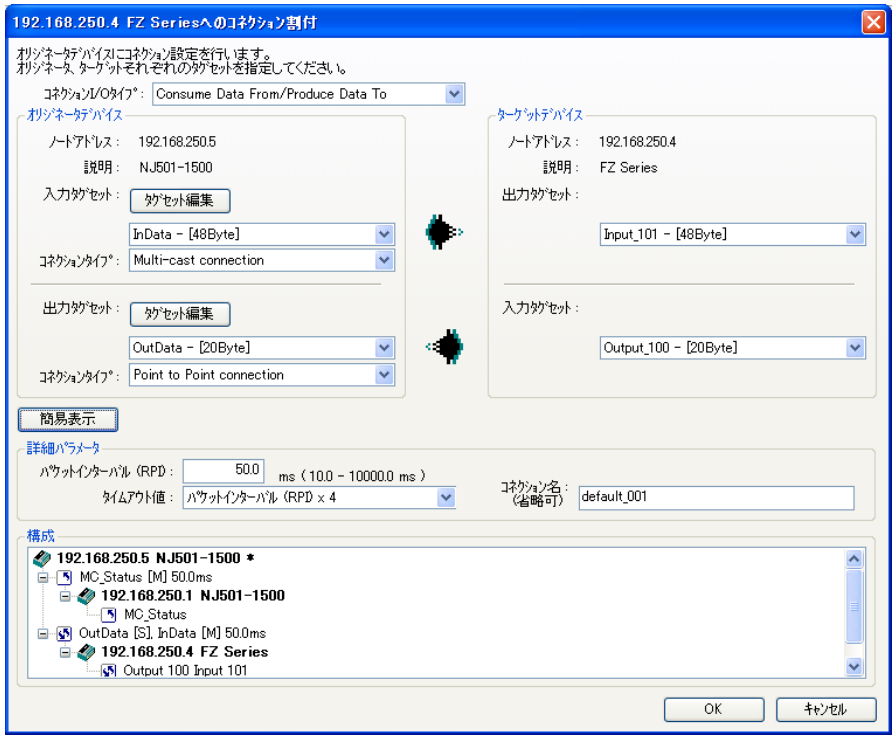
1 つ 1 つのコネクションを個別に編集します。
（注）テーブル形式で一括編集する方法については、次項を参照してください。

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。

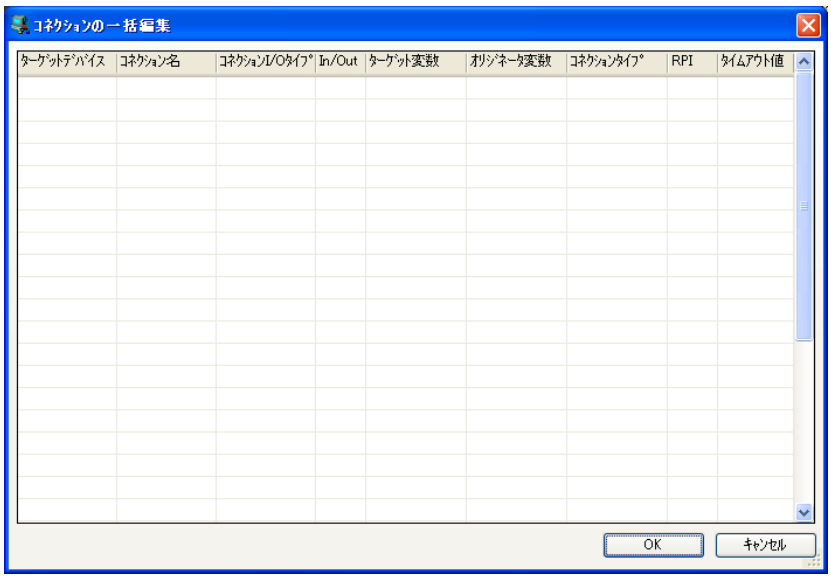
設定項目	内容
コネクション I/O タイプ	CS1W-EIP21、CJ1W-EIP21、CJ2B-EIP21、CJ2M-EIP21、CJ1W-EIP21(CJ2)、CJ1W-EIP21(NJ)、NX701、NJ501-□□□□、NJ301-□□□□、NJ101、NY5□2 とタグデータリンクする場合は、Input Only(Tag type) を選択します。 その他のターゲットデバイスとタグデータリンクする場合は、対象デバイスの EDS ファイルに指定されたコネクション I/O タイプを選択します。 「Input Only(ID type)」は、他社ノードがオリジネータで Tag type 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。 デフォルトは 50ms です（50ms に 1 回の更新になります）。 1 ～ 10000ms の範囲で 1ms 単位で設定できます。
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］ダイアログボックスが表示されます。



設定内容は、以下のとおりです。

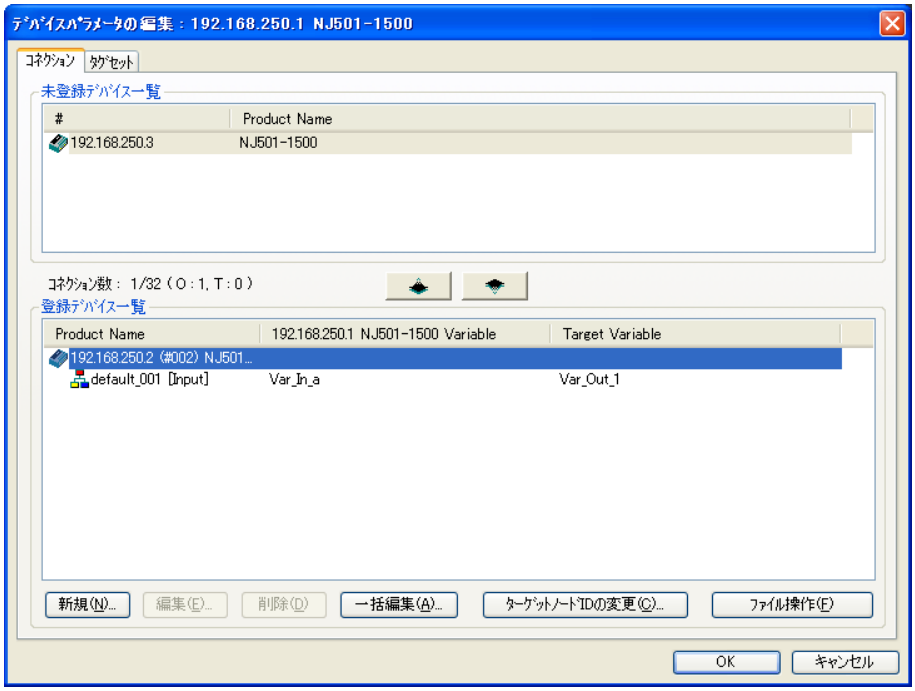
設定項目	内容
ターゲットデバイス	ターゲットデバイスを選択します。
コネクション名	コネクションに任意の名称をつけることができます。（最大半角 32 文字） 空欄にしておくとデフォルトの名称がつきます。 このコネクション名は、コメントとして使用します。
コネクション I/O タイプ	CS1W-EIP21、CJ1W-EIP21、CJ2B-EIP21、CJ2M-EIP21、CJ1W-EIP21(CJ2)、CJ1W-EIP21(NJ)、NX701、NJ501-□□□□、NJ301-□□□□、NJ101、NY5□2 とタグデータリンクする場合は、Input Only(Tag type) を選択します。 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。

設定項目	内容
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。 デフォルトは 50ms です（50ms に 1 回の更新になります）。 1 ～ 20000ms の範囲で 1ms 単位で設定できます。
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。

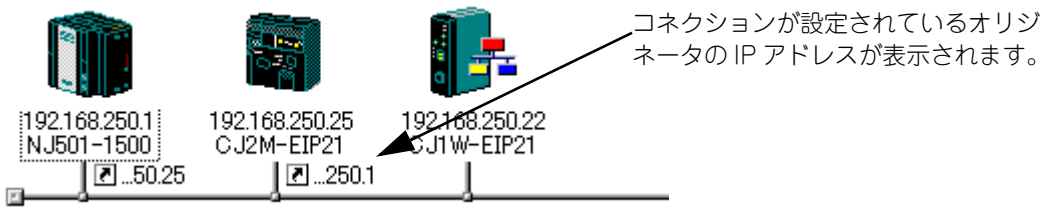
2　**設定終了後、[OK] ボタンをクリックします。**

● コネクション設定結果の確認

1　**登録デバイスー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。**



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。

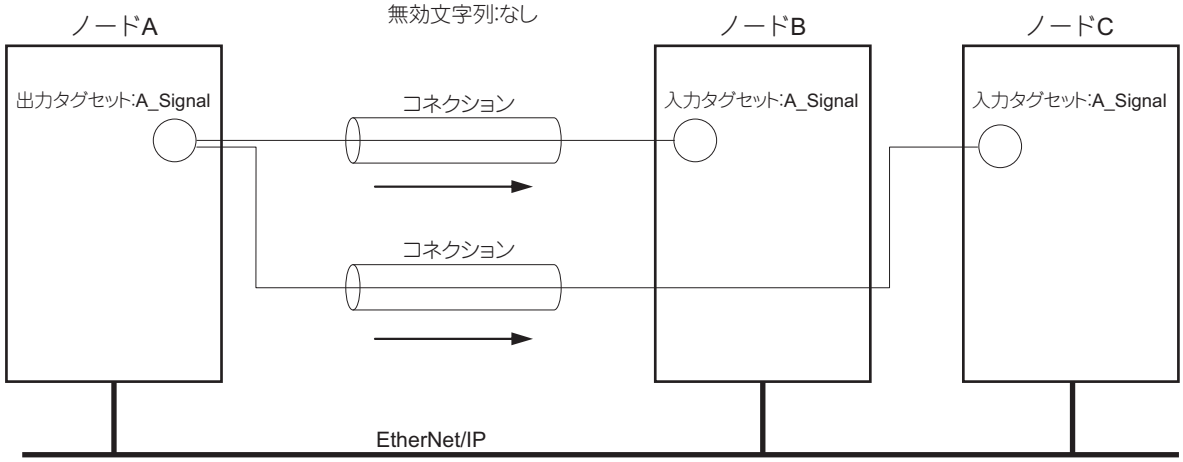
4 **コネクションを設定したのち、オリジネータかターゲットの一方でタグセットのサイズが変更されると、他方でサイズが合わなくなりパラメータ情報の不一致が発生します。**
そのため、コネクション設定の変更を行った場合は、必ず「コネクションのチェック」を実行してください。（詳細は 「7-2-16 コネクションのチェック」を参照してください。）

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]）

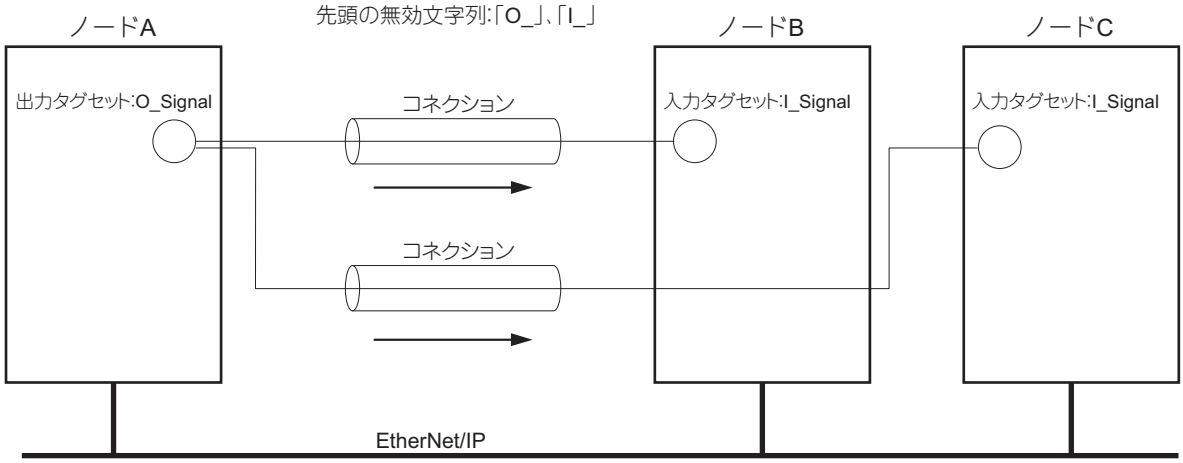
デバイスに設定されているタグセット名を自動識別させ、出力タグセットとタグセット名が同一（もしくは省略文字を除いて同一）の入力タグセットとのコネクション設定を自動で行うことができます。自動コネクション設定は以下の条件で実行されます。

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。

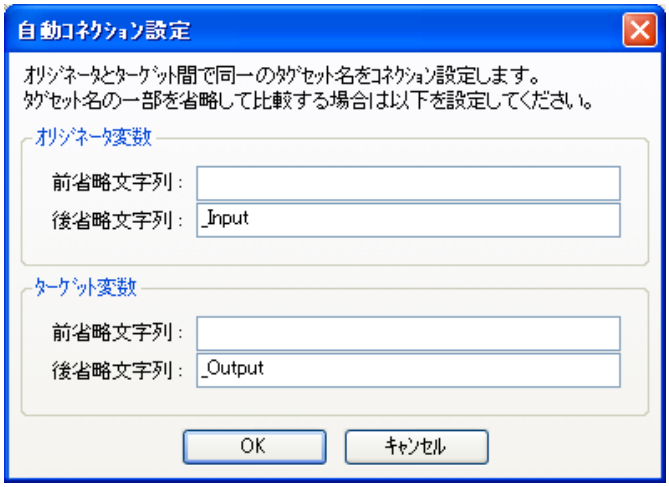


- 例 2：省略文字列を含んだ自動コネクション例
  ノード A のタグセット名「O_Signal」を出力（送信）、ノード B、C のタグセット名「I_Signal」を入力（受信）として、前省略文字列として、「O_」および「I_」を設定した場合、同一のタグセット名を「Signal」として、以下のコネクションが自動設定されます。



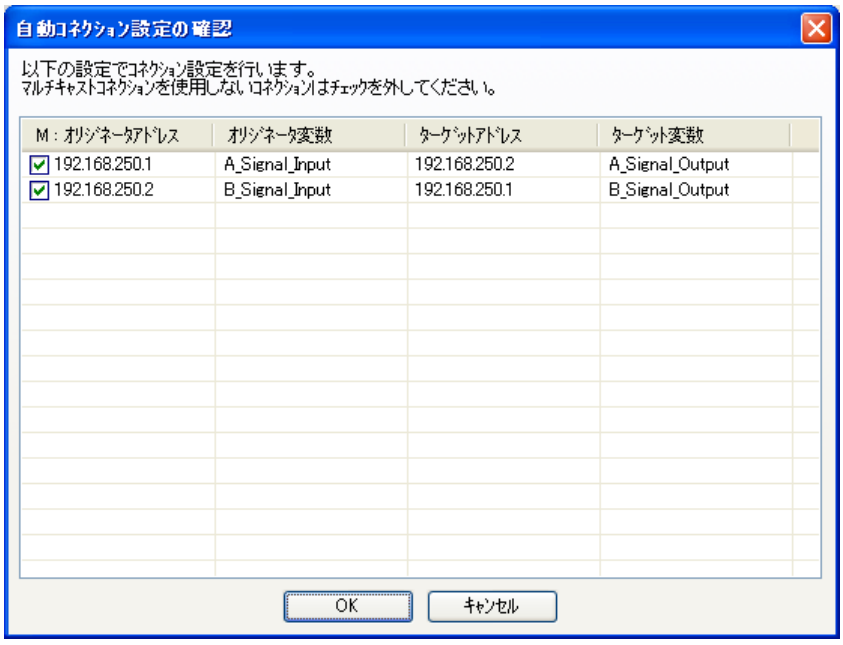
1 **コネクションの出力側と入力側のタグセット名を同じに設定します。前省略文字列、および後省略文字列を含めたタグセット名に設定することも可能です。**

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ダイアログボックスが表示されます。



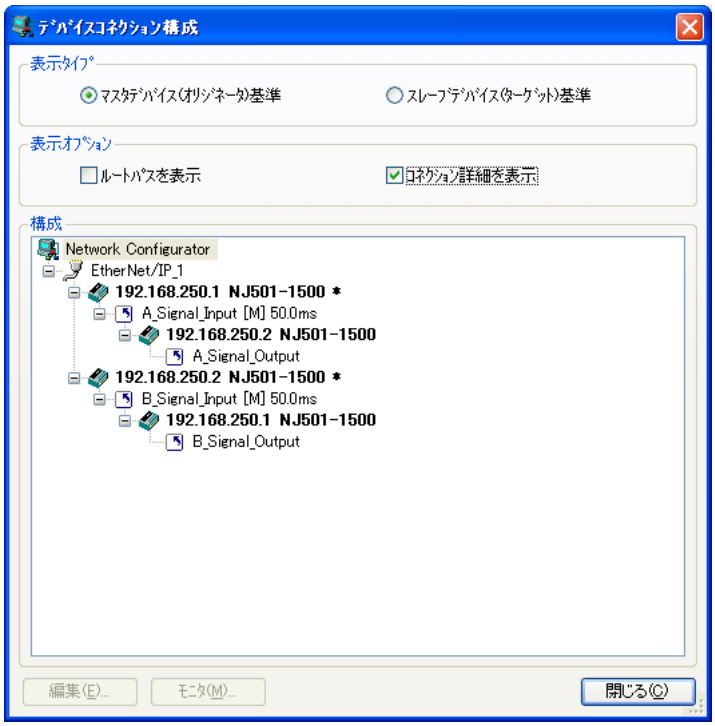
各々の条件を設定し［OK］ボタンをクリックすると、自動設定の処理が開始されます。

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。

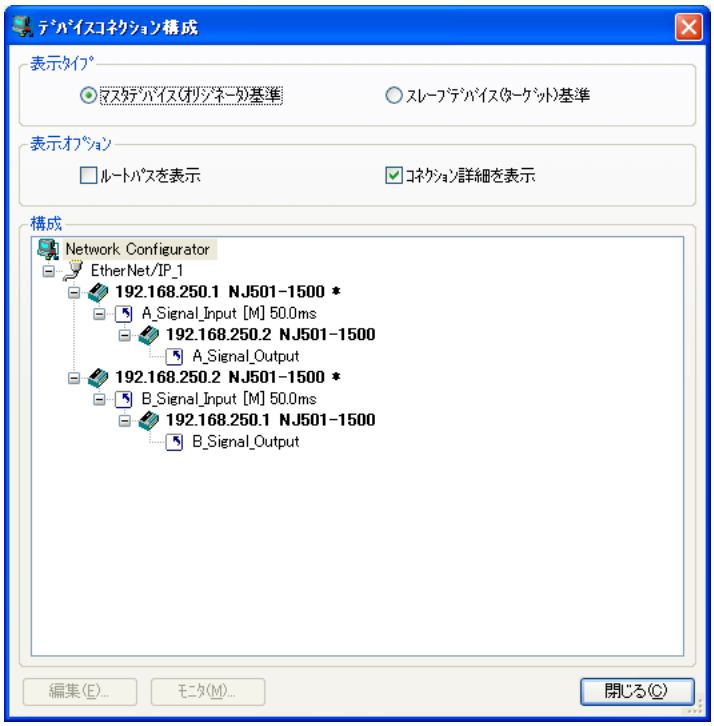
4　**自動設定実行後、コネクション構成表示で設定内容が表示されます。**



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。

デバイス名称	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
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
NX701	NX シリーズコントローラ内蔵の EtherNet/IP ポート
NJ501-□□□□	NJ シリーズコントローラ内蔵の EtherNet/IP ポート
NJ301-□□□□	
NJ101	
NY5□2	NY シリーズコントローラの内蔵 EtherNet/IP ポート

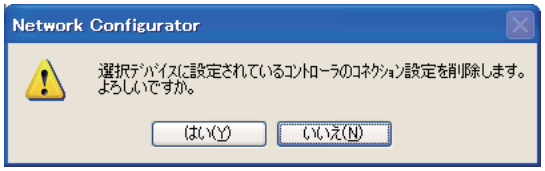
ウィザードを使用したコネクション（タグデータリンク）の作成方法は次の通りです。

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。

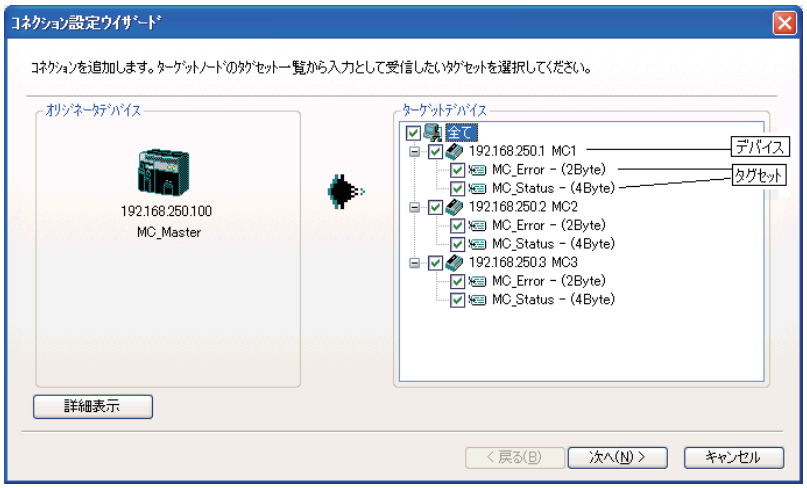


ウィザードが起動する前に次のダイアログボックスが表示されます。



［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。**



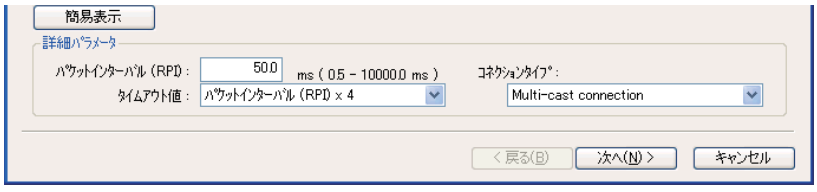
4 **ウィザード右側に、入力として受信可能なターゲットデバイスのタグセット一覧が表示されます。**

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

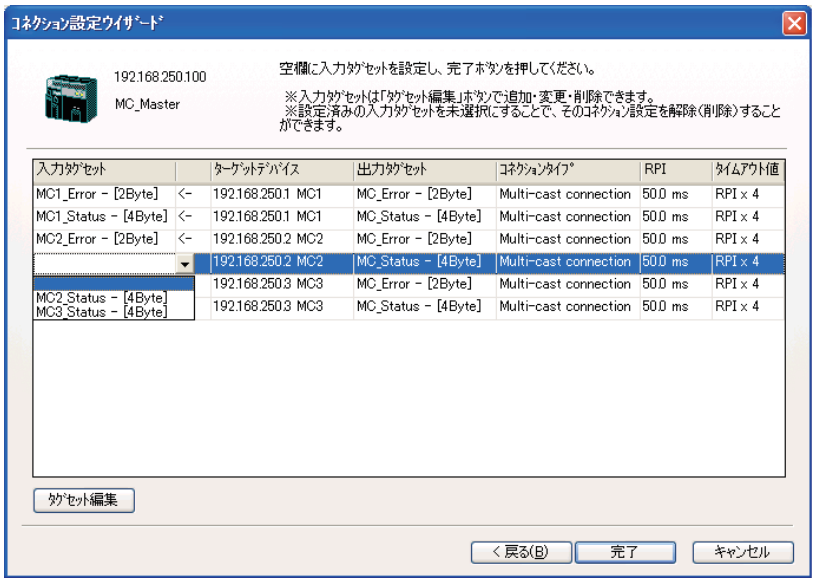
（注） 設定済みのコネクションで使用されているタグセットは表示されません。

［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。**

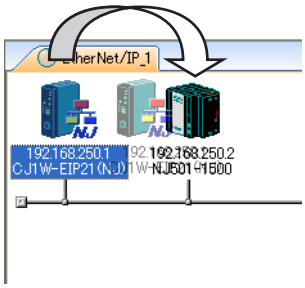
- 入力タグセットが空欄の行を含んでいても、ウィザードを完了することができます。この場合、空欄の行に対しては、コネクションが作成されません。
- 既に設定されていた入力タグセットを空欄にすることで、そのコネクションを解除（削除）することができます。

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。

デバイス名称	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
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
NX701	NX シリーズコントローラ内蔵の EtherNet/IP ポート
NJ501-□□□□ NJ301-□□□□ NJ101	NJ シリーズコントローラ内蔵の EtherNet/IP ポート
NY5□2	NY シリーズコントローラの内蔵 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。

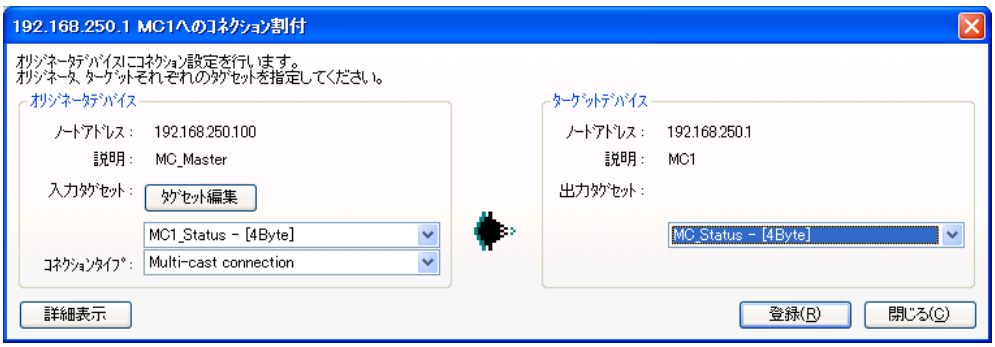
1　ターゲット（ドラッグする）デバイスに、タグ／タグセットを設定します。

(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。

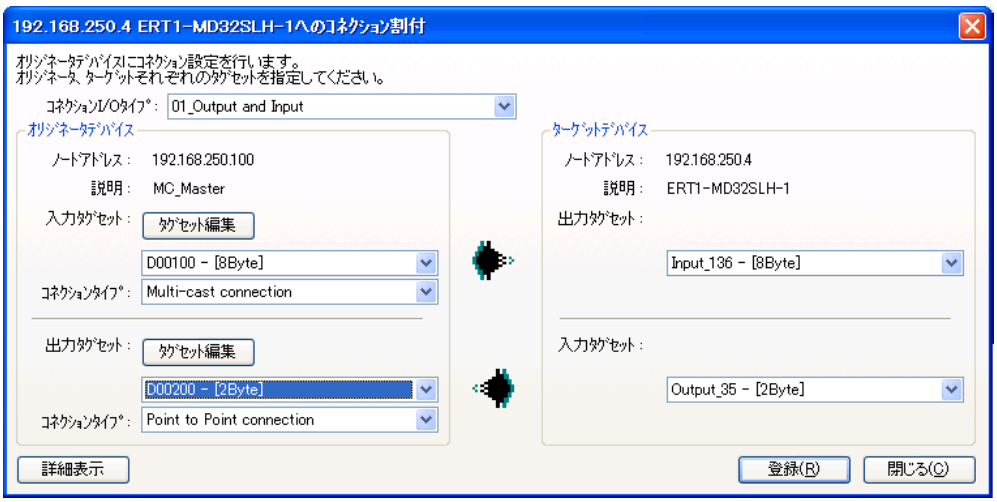
(1)　ターゲットが上記オムロン製 EtherNet/IP デバイスの場合



「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。

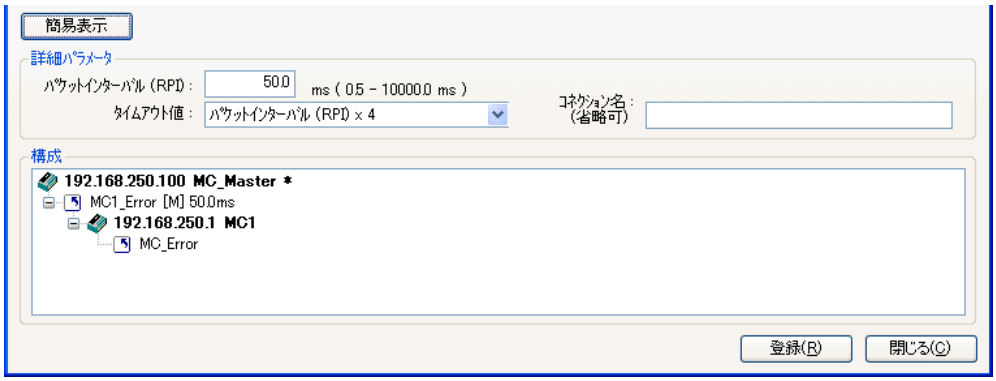
(2)　ターゲットがその他の EtherNet/IP デバイスの場合



「コネクション割付」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、次のダイアログボックスが表示されます。



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

Ethernet のスイッチングハブ経由でコントローラの内蔵 EtherNet/IP ポートへ接続します。

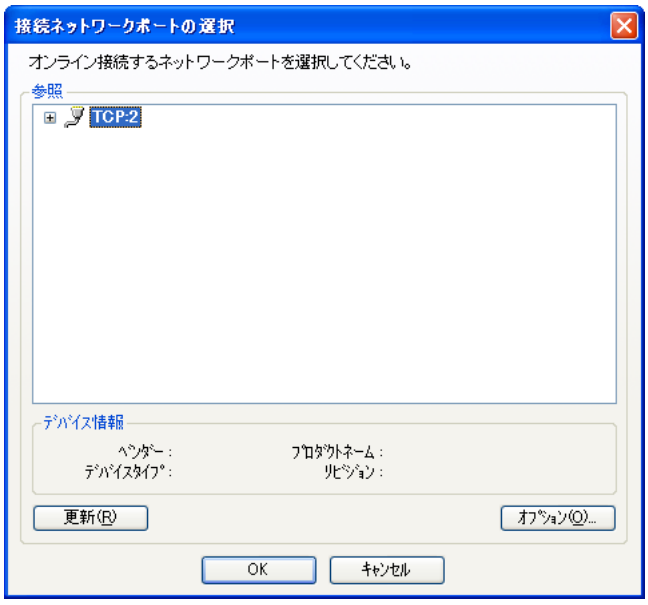
使用上の注意

Windows XP（SP2 以降）/Windows Vista/Windows 7 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、Windows XP/Windows Vista/Windows 7 以降で使用する際の留意点」を参照してください。

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。**

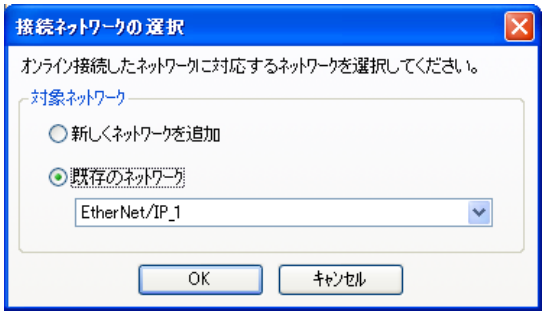
2　**［ネットワーク］｜［接続］を選択します。**

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。

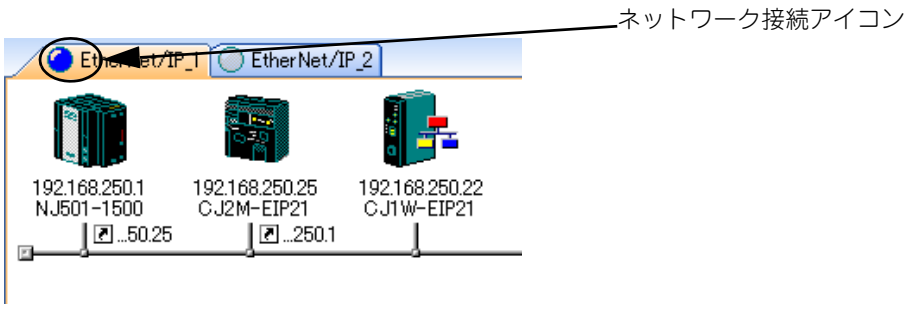


3 **[OK] ボタンをクリックします。**

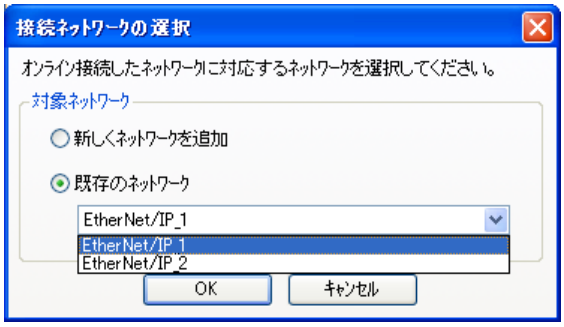
接続するネットワークを選択します。



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。

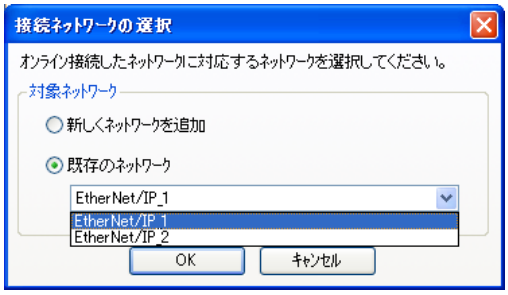


4 **以下のダイアログボックスが表示されます。**



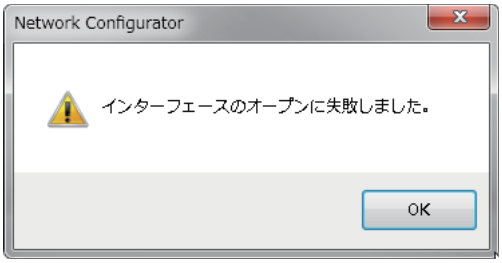
5 **[OK] ボタンをクリックします。**

接続するネットワークを選択します。



参考

NY 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ボックスが表示された場合、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。



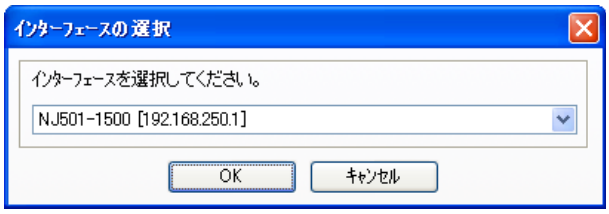
推定要因	処置対策
ケーブルが正しく接続されていない	ケーブルの断線、ゆるみがないか確認してください。
ファイアウォールの設定により、コントローラとの接続がブロックされている	ファイアウォールの設定でコントローラとの接続がブロックされている場合は解除してください。 ファイアウォールの設定は 🕮「A-4 Network Configurator を、Windows XP/Windows Vista/Windows 7 以降で使用する際の留意点」を参照してください。

内蔵 EtherNet/IP ポートへの Ethernet 直接接続

Ethernet から直接、内蔵 EtherNet/IP ポートへ接続します。
パソコン側の IP アドレスが設定されていなくても、内蔵 EtherNet/IP ネットワークに接続できます。

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/NX/NY Series Ethernet Direct I/F］を選択します。

2 **［ネットワーク］｜［接続］を選択し、［OK］ボタンをクリックします。**
接続可能なコントローラが複数ある場合、［インターフェースの設定］ダイアログボックスが表示されます。

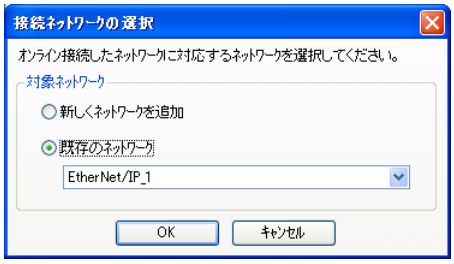


3 **接続するコントローラのインターフェースを選択し、［OK］ボタンをクリックします。**
［コントローラの形式（IP 番号）］で表示される中から選択すると、次に示すダイアログボックスが表示されます。



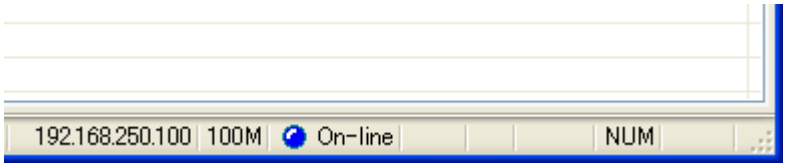
4 **「TCP：2」を選択し、[OK] ボタンをクリックします。**

次に示すダイアログボックスが表示されます。



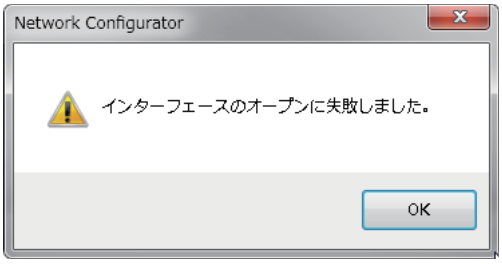
5 **接続するネットワークを選択します。**

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

NY 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ボックスが表示された場合、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、Windows XP/Windows Vista/Windows 7 以降で使用する際の留意点」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。

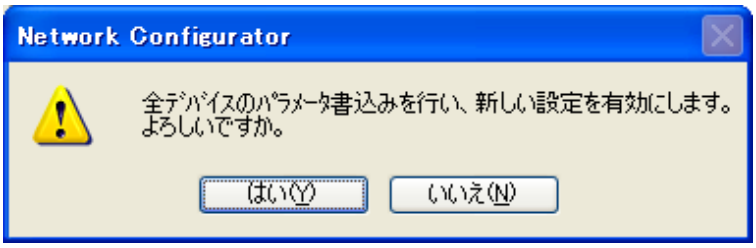
使用上の注意

- 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、これらの変数がコントローラ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。コントローラ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。「14-1 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、自動的に内蔵EtherNet/IPポー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NYシリーズコントローラの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。

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。

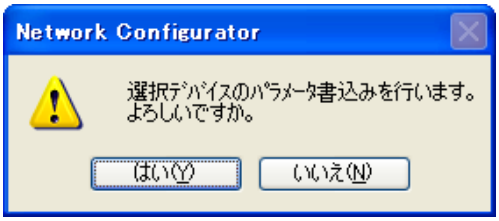
- ネットワークを一括してダウンロードする場合
  ［ネットワーク］メニューから［ダウンロード］を選択します。
  次のダイアログボックスが表示されます。



- 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。

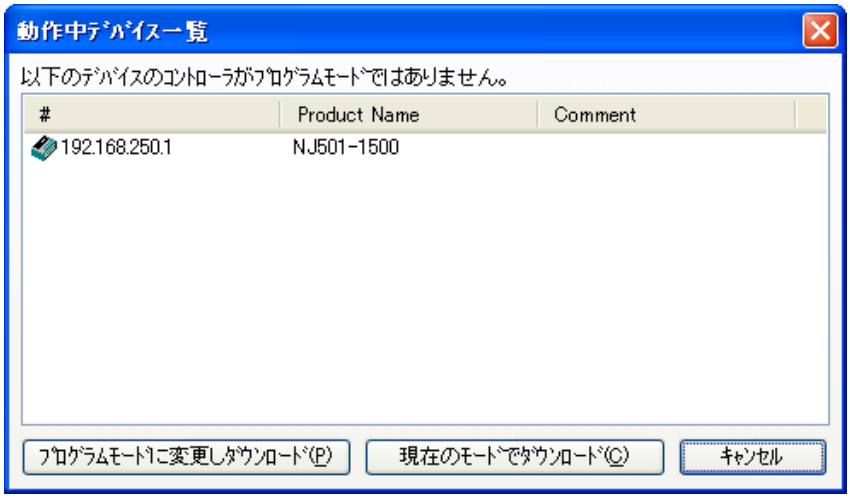


次のダイアログボックスが表示されます。



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。**

すべてのコントローラが、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。

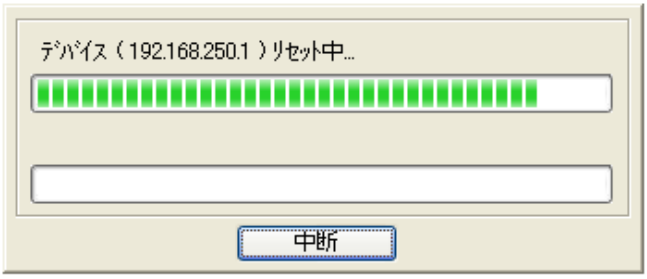


「プログラムモードに変更しダウンロード」をクリックした場合には、すべてのコントローラをいったんプログラムモードへ変更しダウンロードを行います。この場合は、制御装置をプログラムモードに変更しても影響がないことを確認してください。ダウンロード終了後にモードを元に戻すことができます。

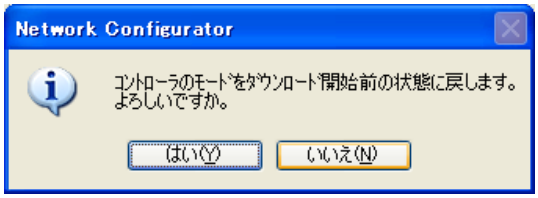
「現在のモードでダウンロード」をクリックした場合には、コントローラが運転モードでもダウンロードが可能です。

「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。



コントローラ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 デバイス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

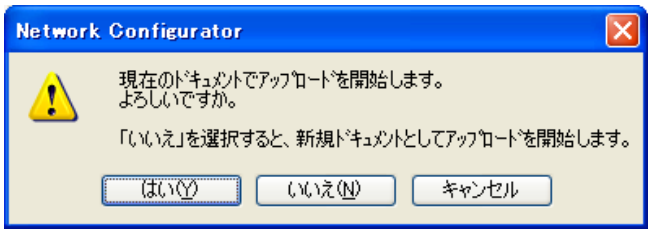
- Network Configuratorで登録したデバイスと使用するコントローラのCIP のメジャー･リビジョンは同じにしてください。CIP のメジャー・リビジョンが異なると、アップロードされない場合があります。アップロードの可否は、🕮「7-2-3 デバイスの登録」を参照してください。

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

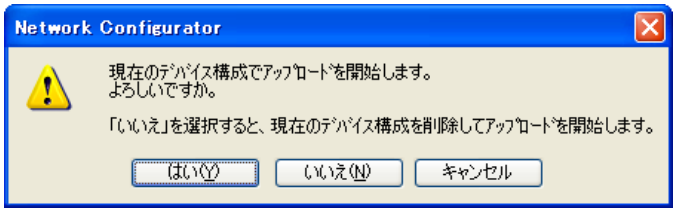
1　**Network Configurator をネットワークへ接続後、［ネットワーク］メニューから［アップロード］を選択します。**

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。

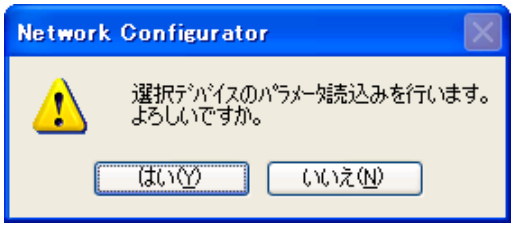


特定のデバイスから個別にアップロードする場合

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。

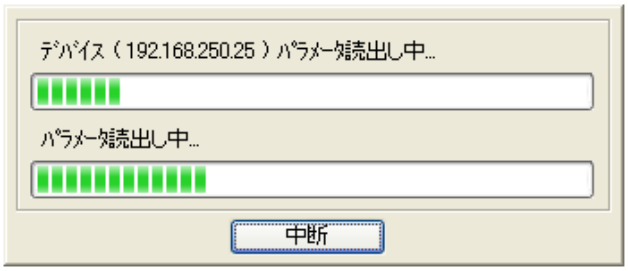


2 **次のダイアログボックスが表示されます。**



［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 ネットワーク上の内蔵 EtherNet/IP ポー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

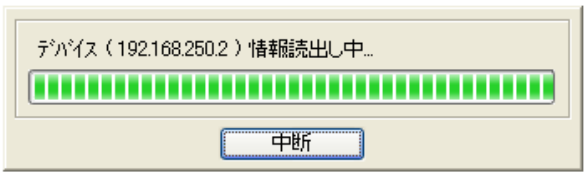
- Network Configuratorで登録したデバイスと使用するコントローラのCIP のメジャー･リビジョンは同じにしてください。CIP のメジャー・リビジョンが異なると、照合されない場合があります。照合の可否は、🕮「7-2-3 デバイスの登録」を参照してください。

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

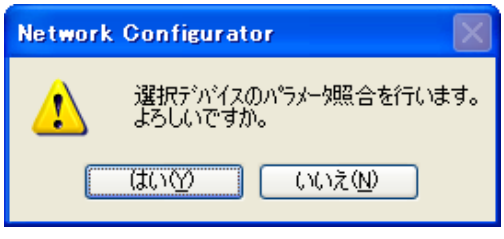
ネットワーク構成ウィンドウ上で選択されたデバイスと、EtherNet/IP ネットワーク上のデバイスとの間でデバイスパラメータを照合します。
ネットワーク構成の照合では、IP アドレスとデバイス種別、およびデバイスパラメータが比較されます。

1　Network Configurator をネットワークへ接続します。

2　照合したい内蔵EtherNet/IPポー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。



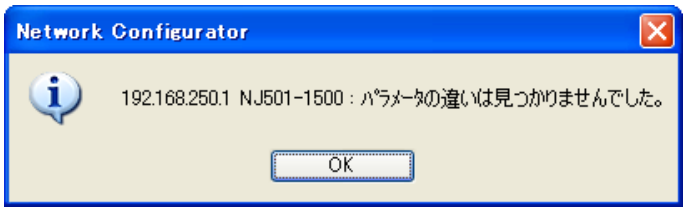
3　次のダイアログボックスが表示されます。



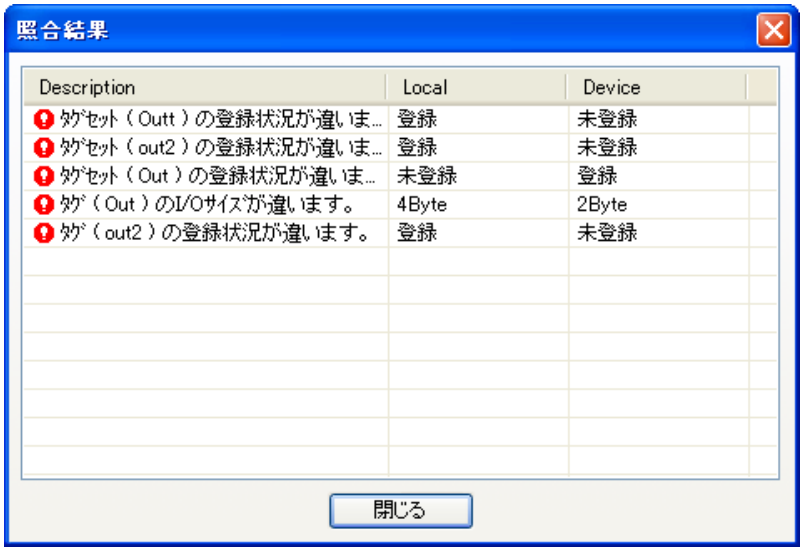
［はい］ボタン、または［いいえ］をクリックします。

4 **次のダイアログボックスが表示されます。**

- 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

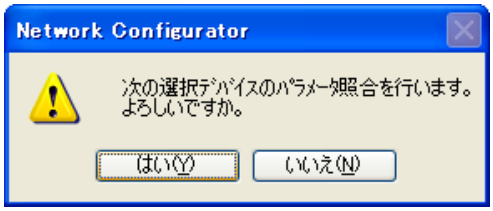


- デバイス種別に違いがあった場合



[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）



参考

オリジネータデバイスとして動作する場合、タグデータリンクの起動後、1 分間以内にターゲットデバイスとコネクションを確立できないと「タグデータリンクコネクション確立タイムアウト」異常が発生します。
この異常が発生した後も定期的に再コネクションを続行することにより、自動的に復旧します。
なお、ターゲットデバイスを後から起動するなど、ご利用の環境によってこの異常を無視できる場合は、イベント重要度変更機能を使用して監視情報に変更することができます。

ネットワーク全体のタグデータリンクの開始／停止

ネットワーク全体の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、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、Network Configurarot 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システム定義変数を 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 内蔵 EtherNet/IP ポート関連のシステム定義変数」を参照）

- 「_EIP_TDLinkStartCmd」（タグデータリンク開始スイッチ）
- 「_EIP_TDLinkStopCmd」（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、システム定義変数の「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後、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。

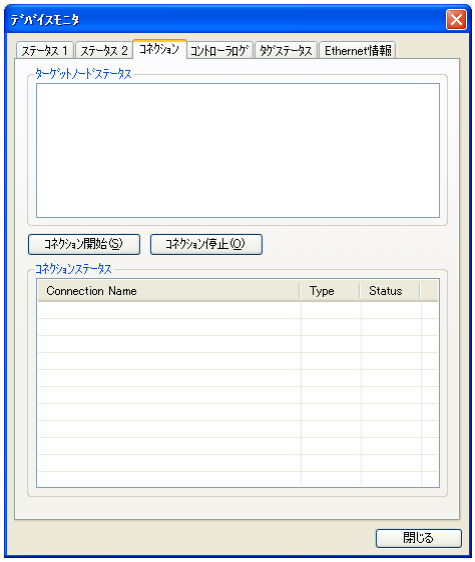
● Network Configurator からの操作

［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、ネットワーク全体の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

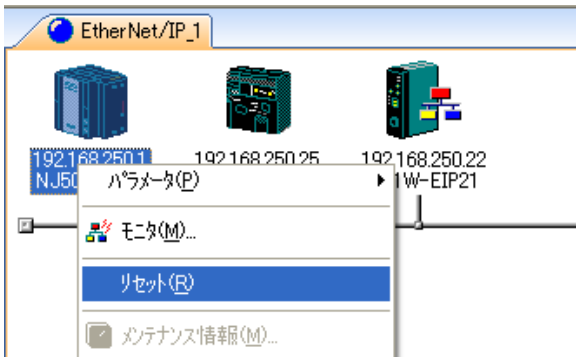
登録した EtherNet/IP のデバイスに保存されているタグデータリンク設定をクリアする（工場出荷状態に戻す）ことができます。
以下にタグデータリンクパラメータをクリアする手順を示します。なお、Network Configurator をネットワークへ接続する方法については 🕮「7-2-8 Network Configurator のネットワーク接続手順」を参照してください。

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デバイス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。

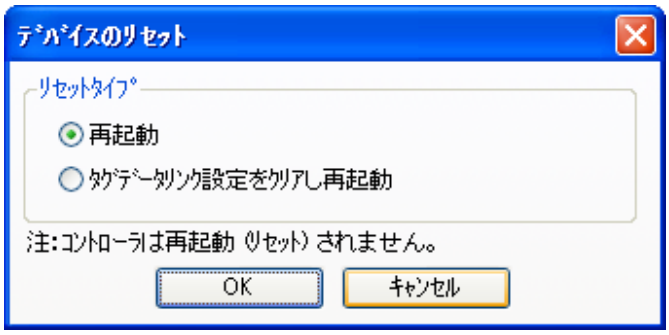


4 **次のダイアログボックスが表示されます。**



- ［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。内蔵 EtherNet/IP ポートおよび内部通信ポートをリセットします。

- ［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。

参考

コントローラの Identity Object の Reset サービスを実行することでタグデータリンク設定をクリアすることができます。Network Configurator 上の実行手順は次のとおりです。

(1)　Network Configurator をネットワークへ接続します。

(2)　メインウィンドウで、[ツール] - [汎用パラメータ設定] を選択します。

汎用パラメータ設定のダイアログボックスが表示されます。

(3)　設定対象デバイスと送信するメッセージを指定します。

- 設定対象ノードアドレス　：対象デバイスの IP アドレスを入力します。
- サービス　：「Reset」を選択します。
- クラス　：「01」を入力します。
- インスタンス　：「01」を入力します。
- アトリビュート　：「00」を入力します。
- データ　：「02」を入力します。

(4)　[送信] ボタンをクリックします。

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。**



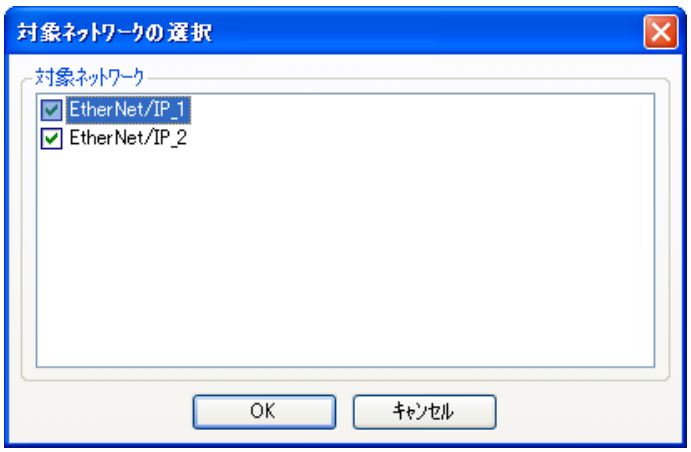
以上で、ネットワーク構成ファイルの保存が完了します。

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

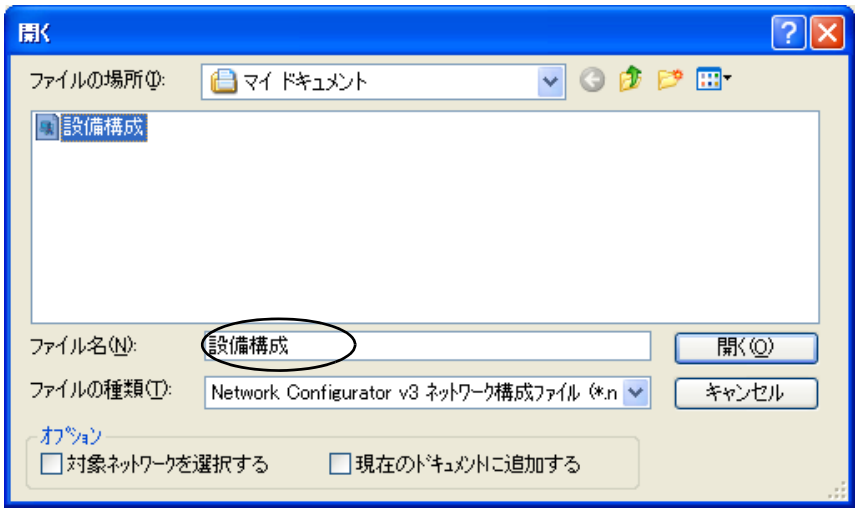
保存しておいたネットワーク構成ファイルを Network Configurator 読み出すことができます。

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。

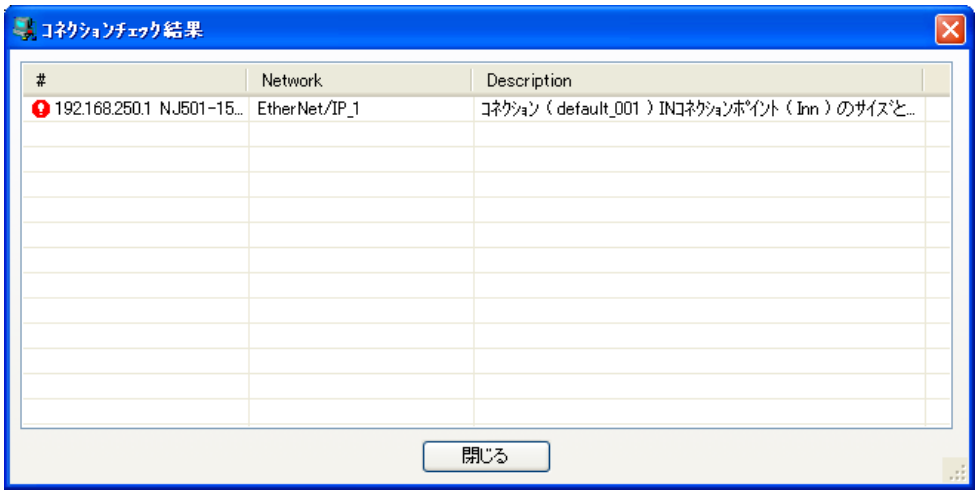
7-2-16　コネクションのチェック

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。

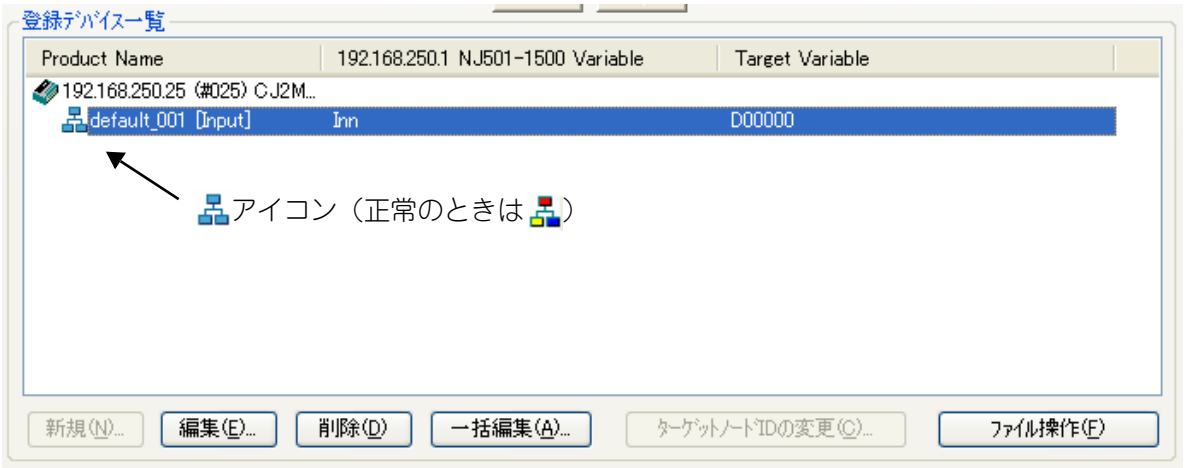
1　**［ネットワーク］メニューから［コネクションのチェック］を選択します。**
パラメータが正常の場合は、次のダイアログボックスが表示されます。



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。

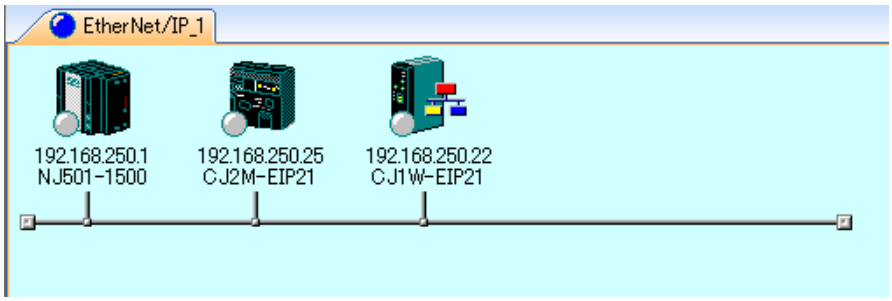
デバイス変更方向

変更後ユニット形式		CJ1W-EIP21 (NJ)		NJ501-□□□□ NJ301-□□□□ NJ101		NX701	NY5□2
変更前 ユニット形式	CIP Rev	Rev2	Rev3	Rev1	Rev2	Rev2	Rev2
CJ1W-EIP21 (NJ)	Rev2	-	○	△	△	○	△
	Rev3	○	-	△	△	○	△
NJ501-□□□□ NJ301-□□□□ NJ101	Rev1	○	○	-	○	×	×
	Rev2	○	○	○	-	○	○
NX701	Rev2	○	○	×	△	-	△
NY5□2	Rev2	○	○	×	△	○	-

○：変更できます。
×：変更できません。
△：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

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。

- **NY シリーズの内蔵 EtherNet/IP ポートがタグデータリンク可能状態になっている条件**
  - **(1) 内蔵 EtherNet/IP 異常（_EIP_ErrSta）の下記異常 bit が FALSE**
    （全停止フォールト：bit7）
    （部分停止フォールト：bit6）
    （軽度フォールト：bit5）が FALSE、かつ
  - **(2) オンライン（_EIP_EtnOnlineSta）（*1）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情報（_EIP_EstbTargetSta）のターゲットノードアドレスに対応するビットが TRUE**

- **ターゲットノードのコントローラの運転情報（運転、停止）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 PLC 動作モード（_EIP_TargetPLCModeSta）のターゲットノードアドレスに対応するビットが TRUE**

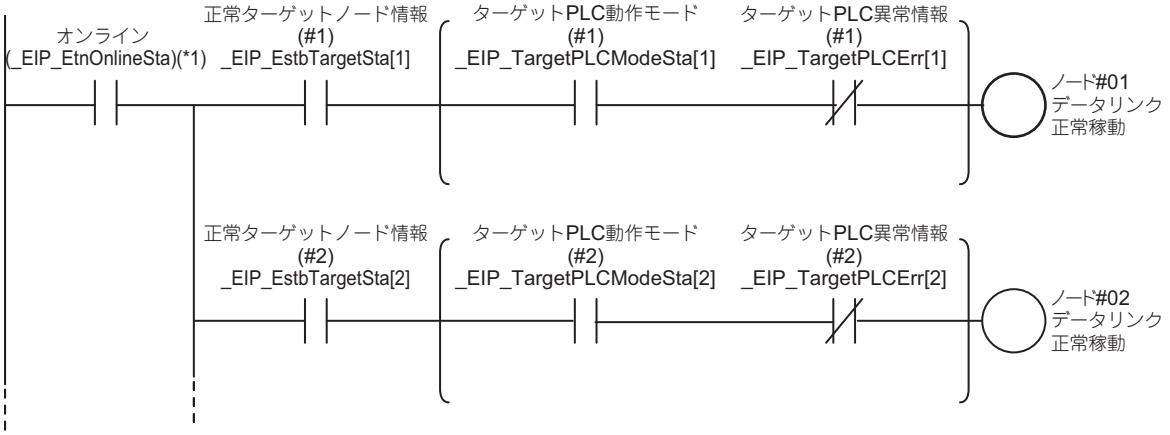
- **ターゲットノードの PLC 異常状態（停止異常、継続異常）による条件（オムロン製コントローラのみ有効）**
  - **(5) ターゲット PLC 異常情報（_EIP_TargetPLCErr）のターゲットノードアドレスに対応するビットが FALSE**
    ターゲットノードの PLC 本体運転･異常情報を使用したい場合は Network Configurator にてオリジネータ、ターゲット双方のタグセットでコントローラステータスを「含める」に設定しておく必要があります。

*1 通信ポートごとに以下となります。
内蔵 EtherNet/IP ポート：_EIP1_EtnOnlineSta
内部通信ポート：_EIPIn1_EtnOnlineSta

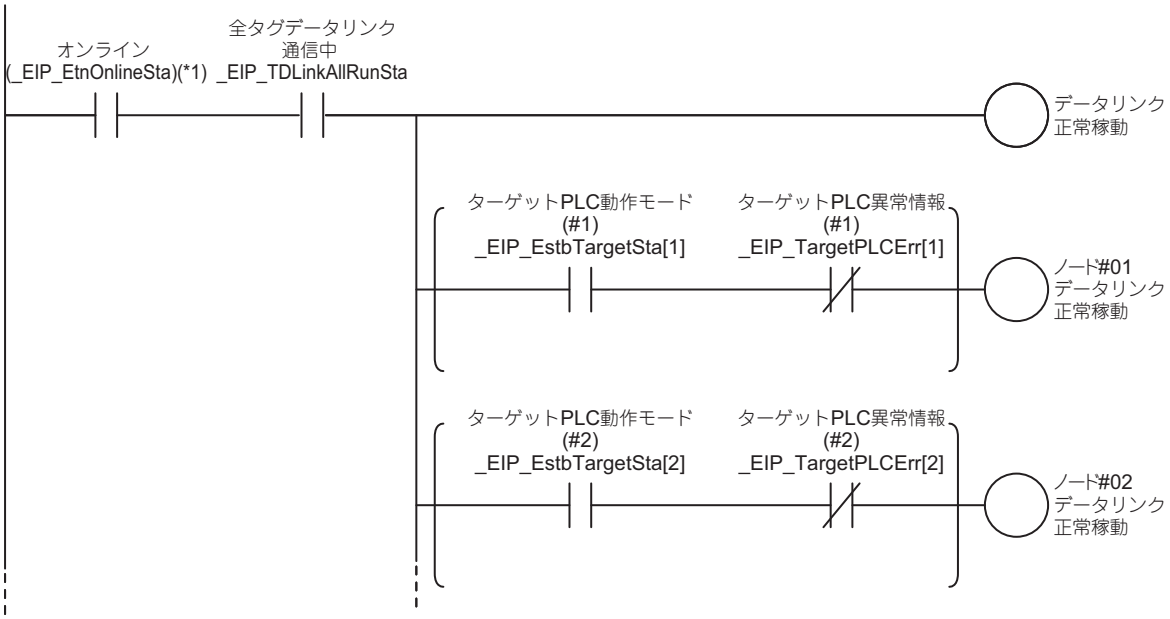
● 正常検知プログラム例

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



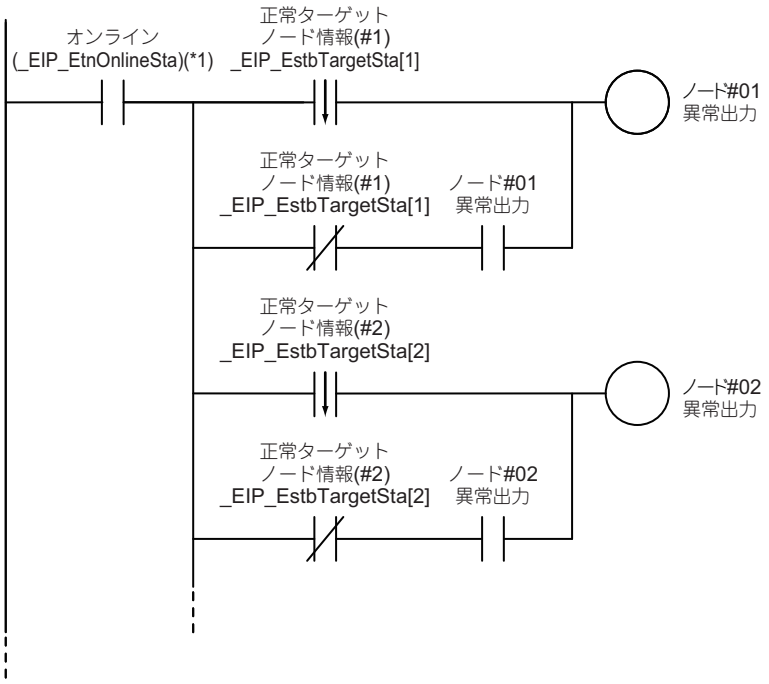
- 正常検知プログラム例 2



*1 通信ポートごとに以下となります。
内蔵 EtherNet/IP ポート：_EIP1_EtnOnlineSta
内部通信ポート：_EIPIn1_EtnOnlineSta

● 異常検知プログラム例

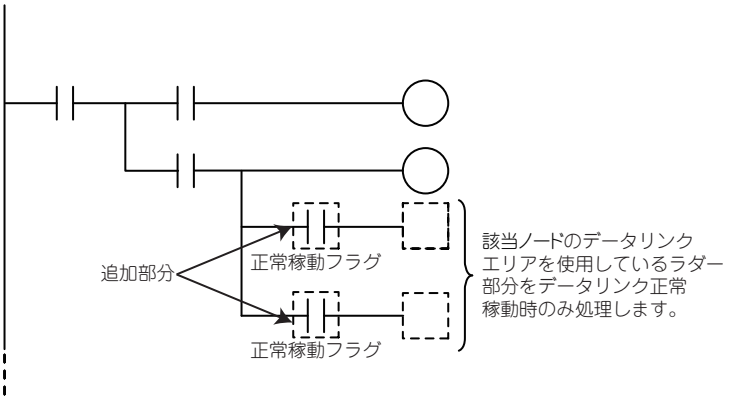
他ノード（ターゲットノード）ごとのタグデータリンク異常を確認するプログラムです。全ノードでデータリンクが正常に起動した後に新たに発生する異常を検知する場合に使用します。



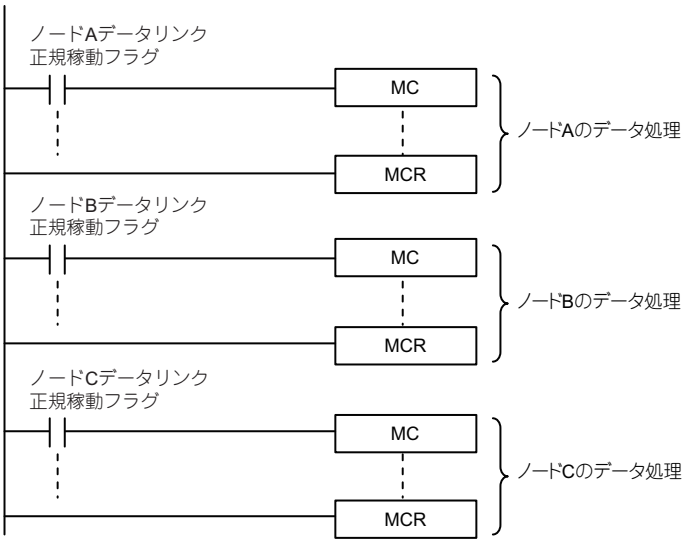
*1 通信ポートごとに以下となります。
内蔵 EtherNet/IP ポート：_EIP1_EtnOnlineSta
内部通信ポート：_EIPIn1_EtnOnlineSta

● 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、内蔵 EtherNet/IP 異常（_EIP_ErrSta）の下記の bit 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。
（全停止フォールト：bit7）
（部分停止フォールト：bit6）
（軽度フォールト：bit5）が ON の場合

7-3-2　タグデータリンクに関連するステータス

タグデータリンクのステータスは、以下に示すシステム定義変数に反映されます。

変数名	内容
_EIP_TargetPLCModeSta［255］（ターゲット PLC 動作モード） ※ コントローラステータスの「コントローラ本体運転情報」に該当します	内蔵 EtherNet/IP 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
_EIP_TargetNodeErr［255］（ターゲットノード異常情報） ※ コントローラステータスの「コントローラ本体異常情報」に該当します	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情報が 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 で、ターゲット PLC 異常情報が FALSE となっています。
_EIP_EstbTargetSta［255］（正常ターゲットノード情報 ） ※ コントローラステータスには該当しない	内蔵 EtherNet/IP 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　他機種とのタグデータリンク

タグデータリンクの性能は、コントローラや EtherNet/IP ユニットの機種によって、以下のように異なります。
NY シリーズコントローラの内蔵 EtherNet/IP ポートと、他のコントローラや EtherNet/IP 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		NY シリーズコントローラ	NX シリーズコントローラ	NJ シリーズコントローラ		形 CJ2M-CPU3 □ ユニットバージョン		形 CS1W-EIP21 形 CJ1W-EIP21 形 CJ2H-CPU6 □ -EIP
				Ver.1.00～1.02	Ver.1.03 以降	Ver.2.0	Ver.2.1 以降	
タグ	全タグの合計サイズ	92416CH	184832CH（2 ポート合計で 369664CH）	9600CH		640CH		184832CH
	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）		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）		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）	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）
	登録可能なタグの数	256（*3）	256（*3）（2 ポート合計で 512）	256（*3）		32		256
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）		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）		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）	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） 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	128	256（2 ポート合計で 512）	32		32		256
コネクション	コネクション数	128	256（2 ポート合計で 512）	32		32		256
	1 コネクションあたりの最大データサイズ	722CH（*2）（1 コネクション内でのデータの同期性は確保されます。）	722CH（*2）（1 コネクション内でのデータの同期性は確保されます。）	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」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	252CH または 722CH（*2）（1 コネクション内でのデータの同期性は確保されます。
パケットインターバル（RPI）		1～10000ms（1ms 単位）	0.5～10000ms（0.5ms 単位）	10～10000ms（1ms 単位）	1～10000ms（1ms 単位）	1～10000ms（0.5ms 単位）		0.5～10000ms（0.5ms 単位）
使用通信領域（PPS）（*1）		20000pps	40000pps	1000pps	3000pps	3000pps		12000pps

*1 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。
*2 データサイズとして 505 バイト以上を使用するには、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 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

● 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、使用するコントローラによっては、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。
コントローラ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

コントローラ	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
NY シリーズコントローラ	−	NY512 NY532	○	×
NX シリーズコントローラ	−	NX701	○	×
NJ シリーズコントローラ	−	NJ501- □□□□ NJ301- □□□□ NJ101	○	○（*）
	形 CJ1W-EIP21	CJ1W-EIP21(NJ)	○	○（*）
形 CJ2H-CPU6 □ -EIP	−	CJ2B-EIP21	○	○
	形 CJ1W-EIP21	CJ1W-EIP21(CJ2)	○	○
形 CJ2H-CPU6 □	形 CJ1W-EIP21	CJ1W-EIP21(CJ2)	×	○
形 CJ2M-CPU3 □	−	CJ2M-EIP21	○	○
	形 CJ1W-EIP21	CJ1W-EIP21(CJ2)	○	○
形 CJ2M-CPU1 □	形 CJ1W-EIP21	CJ1W-EIP21(CJ2)	×	○
CJ1 CPU ユニット	形 CJ1W-EIP21	CJ1W-EIP21	×	○
CS1 CPU ユニット	形 CS1W-EIP21	CS1W-EIP21	×	○

* 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。

8

CIP メッセージ通信機能

8-1 CIP メッセージ通信機能の概要 8-3
8-1-1 CIP メッセージ通信サービス機能の概要 8-3
8-1-2 メッセージ通信サービスの機能・性能一覧 8-3

8-2 CIP メッセージ通信のクライアント機能 8-4
8-2-1 概要 8-4
8-2-2 CIP 通信命令の種類 8-5
8-2-3 CIP 通信命令の使用方法 8-6
8-2-4 ルートパス 8-7
8-2-5 リクエストパス（IOI） 8-10
8-2-6 サービスデータ／レスポンスデータ 8-14
8-2-7 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） 8-15
8-2-8 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） 8-20
8-2-9 動作タイミング 8-27
8-2-10 レスポンスコード 8-28

8-3 CIP メッセージ通信のサーバ機能 8-32
8-3-1 CIP オブジェクトにアクセスするための CIP メッセージ構造 8-32
8-3-2 変数にアクセスするための CIP メッセージ構造 8-33

8-4 リクエストパスの指定 8-34
8-4-1 CIP オブジェクトの指定例 8-34
8-4-2 変数の指定例 8-34
8-4-3 Logical Segment 8-35
8-4-4 Data Segment 8-35
8-4-5 リクエストパスでの変数名の指定方法 8-36

8-5 CIP オブジェクトサービス 8-40
8-5-1 内蔵 EtherNet/IP ポート宛 CIP オブジェクトの種類 8-40
8-5-2 Identity オブジェクト（クラス ID：01Hex） 8-41
8-5-3 TCP/IP Interface オブジェクト（クラス ID：F5Hex） 8-44
8-5-4 Ethernet Link オブジェクト（クラス ID：F6 Hex） 8-47
8-5-5 PLC オブジェクト（クラス ID：C4Hex） 8-51

8-6 変数への読出し／書込みサービス 8-52
8-6-1 変数読出しサービス 8-52
8-6-2 変数書込みサービス 8-53

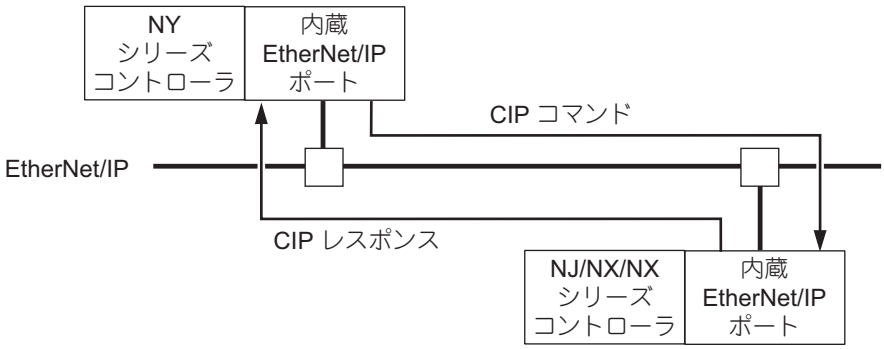
8-7 変数のデータタイプ一覧 8-55
8-7-1 データタイプコード 8-55
8-7-2 共通フォーマット 8-55

8-7-3　基本型 (Elementary Data Types) 8-56
8-7-4　派生型 (Derived Data Types) 8-58

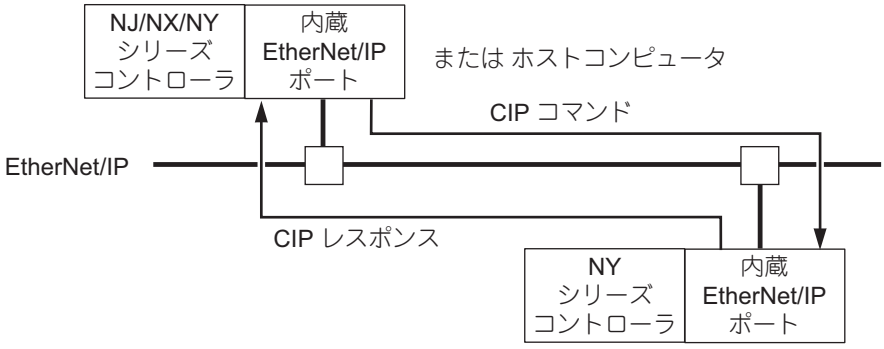
8-1　CIP メッセージ通信機能の概要

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。
NY シリーズコントローラのプログラム上で 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

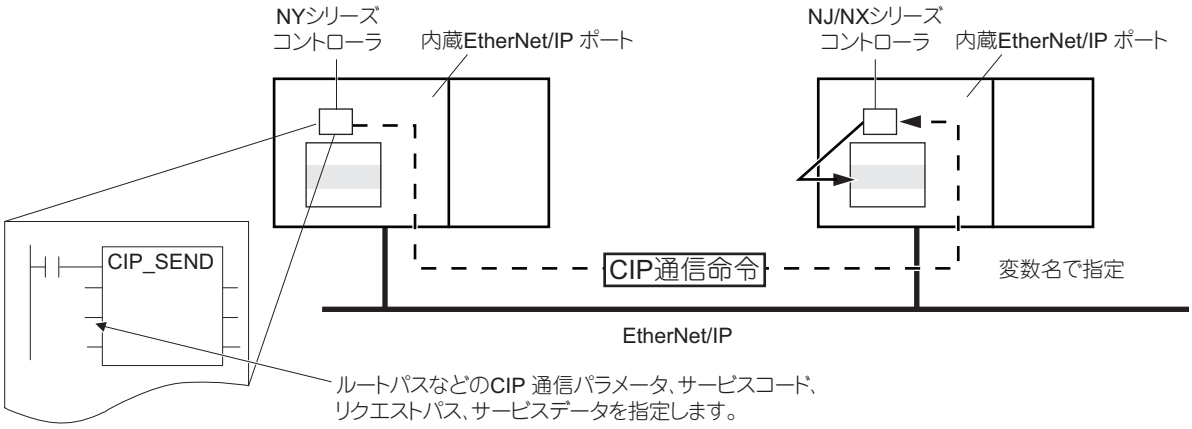
項目		内容
メッセージ種類		以下を選択可能 非 CIP コネクション型（UCMM） CIP コネクション型（Class3）
実行方法		CIPSend 命令（Class3：コネクション型）、または CIPUCMMSend 命令（UCMM：非コネクション型）による
データ内容		任意の CIP コマンドの発行、レスポンス受信
通信パラメータ		メッセージ種別、タイムアウト時間、ルートパス指定
1 コネクションあたりのデータ長	非コネクション型 (UCMM)	502 バイト
	コネクション型 (Class3)	・Forward_Open 使用時 502 バイト ・Large_Forward_Open 使用時 8192 バイト

8-2　CIP メッセージ通信のクライアント機能

8-2-1　概要

NY シリーズコントローラは、外部に対して CIP メッセージを発行し、CIP メッセージ通信サーバ機能をもつ機器の内蔵オブジェクトを指定してサービスを要求することができます。
この機能を、「CIP メッセージ通信のクライアント機能」と呼びます。

NY シリーズコントローラでは、ユーザプログラム上で CIP 通信命令を実行して、CIP メッセージを発行します。CIP メッセージにより、EtherNet/IP 上の NJ/NX シリーズコントローラの変数の読み書きなどを実行できます。



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 🕮『NY シリーズ　コマンドリファレンスマニュアル　基本編（SBCA-437）』を参照してください。

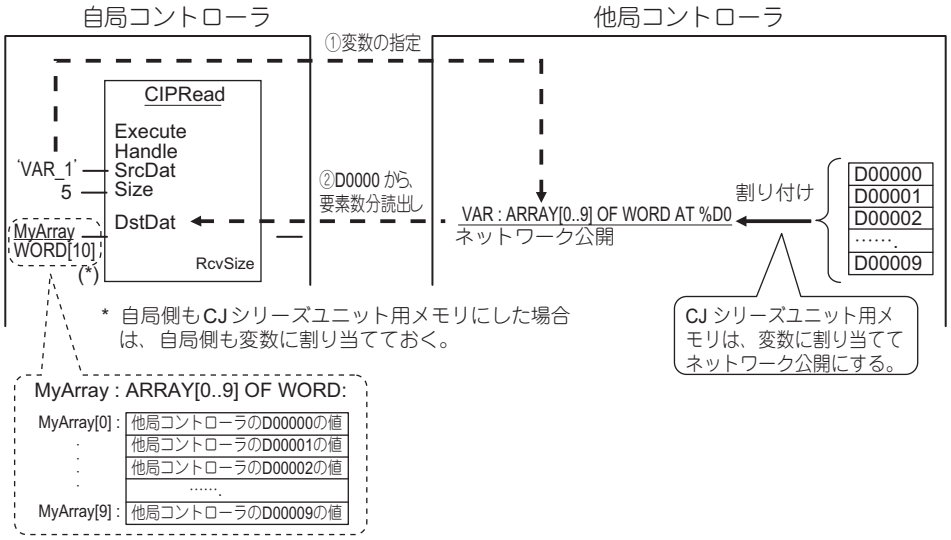
命令	命令名	内容	通信方式
CIPUCMM Read 命令	変数読出（UCMM Explicit メッセージ）	指定した CIP ネットワーク上の、他局コントローラのネットワーク公開属性の変数の値を読み出して、自局コントローラの変数に値を格納します。	CIP コネクションレス型（UCMM）
CIPUCMM Write 命令	変数書込（UCMM Explicit メッセージ）	指定した CIP ネットワーク上の、他局コントローラのネットワーク公開属性の変数に、自局コントローラの変数の値を書き込みます。	
CIPUCMM Send 命令	任意 Explicit メッセージ送信（UCMM）	指定した 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。 NY シリーズコントローラ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 🕮「8-2-10 レスポンスコード」、🕮「8-5 CIP オブジェクトサービス」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 Class3 コマンドを発行します。 NY シリーズコントローラ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、🕮「8-2-10 レスポンスコード」、🕮「8-5 CIP オブジェクトサービス」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 Class3 コネクションを切断します。	

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。 CIP Class3 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、同時取得できるコネクションのハンドル数は、32 個以下です。
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



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。
上記個数以上の命令が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

自コントローラから、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。

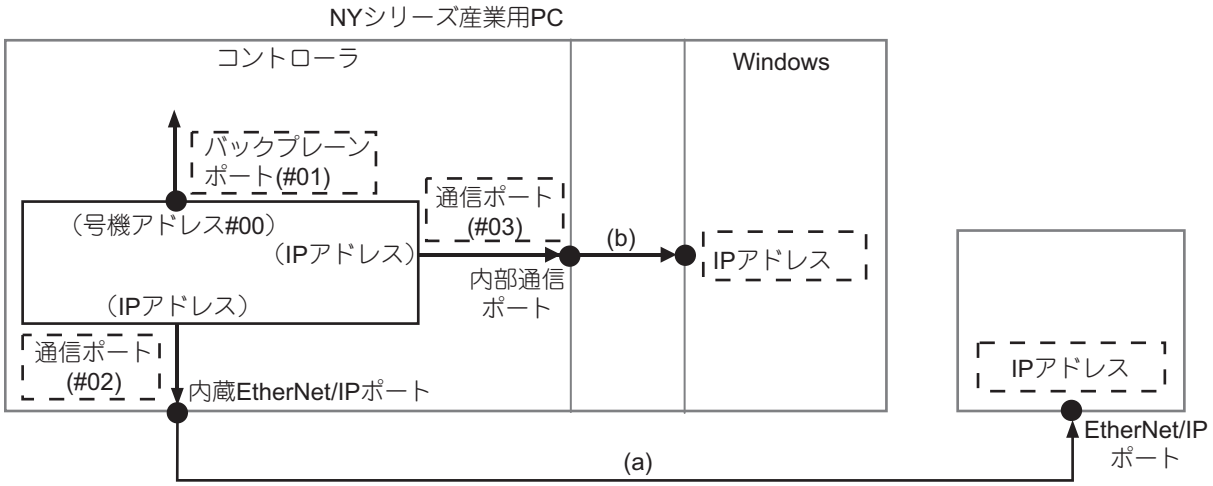
ルートパスの記述フォーマット

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

NY シリーズ産業用 PC のコントローラの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。

ネットワーク種別	ルートパス記述内容	
	ネットワーク種別番号 (16 進)	相手アドレス (16 進)
バックプレーンポート	#01	相手先ユニットの号機アドレス（下記の参考を参照）
内蔵 EtherNet/IP ポート	#02	IP アドレス
内部通信ポート	#03	



(a) 産業用 PC のコントローラから EtherNet/IP ネットワーク上のノードに出力する経路の場合
内蔵 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。

(b) 産業用 PC のコントローラから Windows に出力する経路の場合
内部通信ポートに出力して、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。
・ コントローラ：00Hex

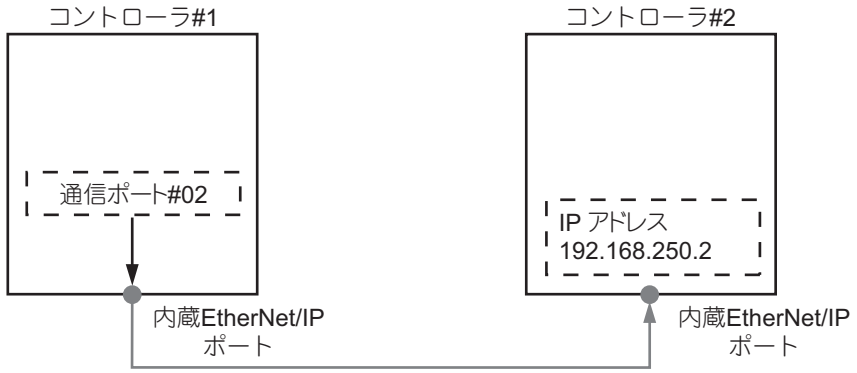
ルートパスの記述例

以下にルートパスの記述例を示します。

● 産業用 PC のコントローラから EtherNet/IP ネットワーク上のノードに出力

(1) 内蔵 EtherNet/IP ポート同士で通信する場合

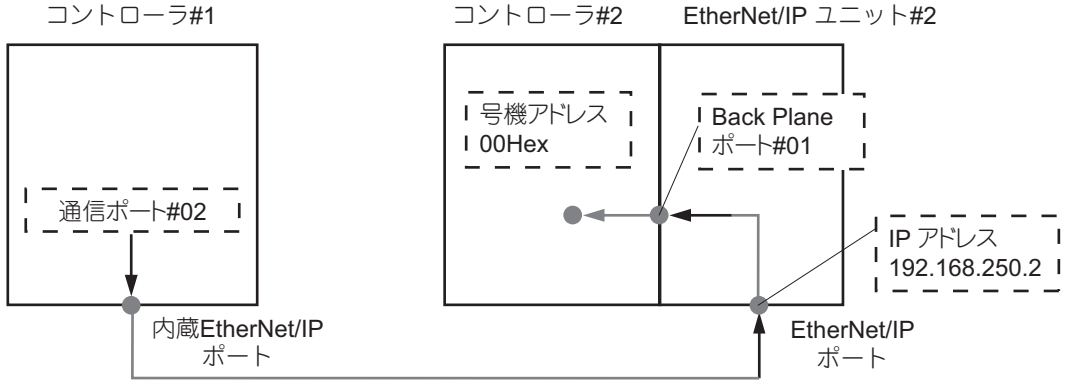
例）コントローラ #1(NY シリーズ) とコントローラ #2 の内蔵 EtherNet/IP ポート同士により通信する場合



- 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　：02\192.168.250.2

(2) 内蔵 EtherNet/IP ポートから、EtherNet/IP ユニットへ通信する場合

例）コントローラ #1(NY シリーズ) の内蔵 EtherNet/IP ポートから、コントローラ #2 に装着されている EtherNet/IP ユニット経由で、コントローラ #2 に通信する場合



①コントローラ #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

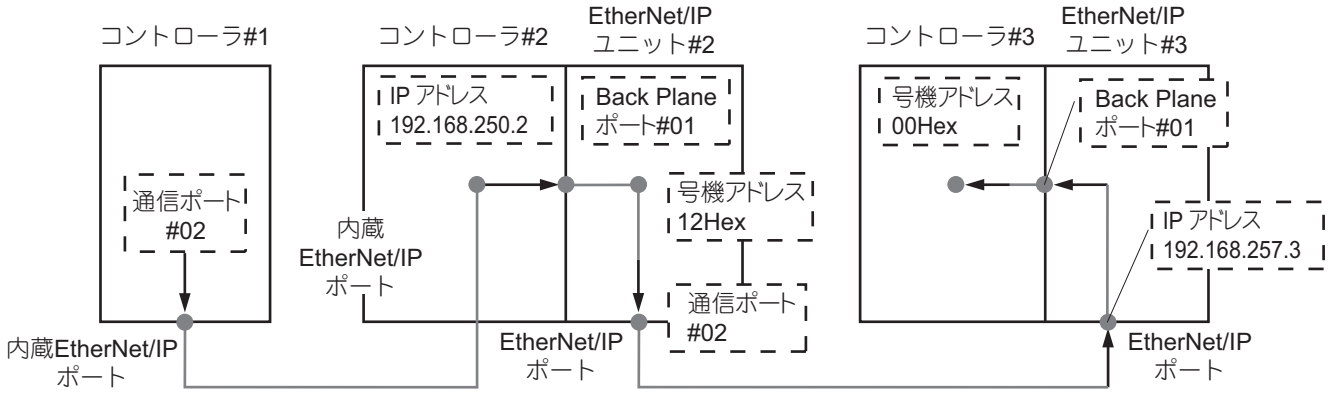
② EtherNet/IP ユニット #2 →コントローラ #2

- 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”（コントローラの号機アドレス）

ルートパス　：　02\192.168.250.2\01\#00
　　①　②

(3) 中継局を介してアクセスする場合

コントローラ #2 を介して、コントローラ #1(NY シリーズ) からコントローラ #3 に通信を行う場合



①コントローラ #1 →コントローラ #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

②コントローラ #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））

③ 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

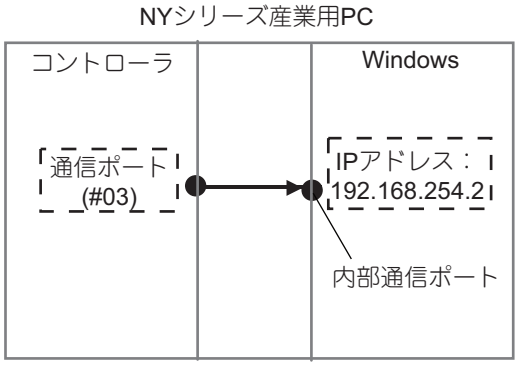
④ EtherNet/IP ユニット #3 →コントローラ #3

- 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”（コントローラの号機アドレス）

ルートパス　：　02\192.168.250.2\01\#12\02\192.168.257.3\01\#00

①　②　③　④

● 産業用 PC のコントローラから Windows に出力する経路の場合



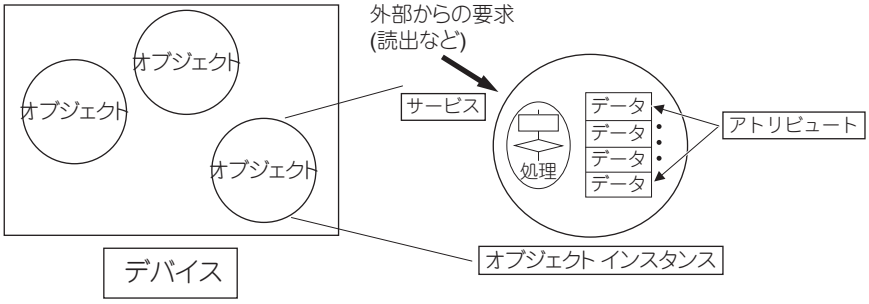
- ネットワーク種別番号　：“#03”（内部通信ポートによる出力）
- 相手アドレス　：相手 IP アドレスを指定
- ルートパス　：03\192.168.254.2

8-2-5　リクエストパス（IOI）

リクエストパスとは、ネットワーク上の他デバイスのオブジェクトを示したパスです。
CIP 通信命令では、このリクエストパスをもとに、他デバイスのオブジェクトにアクセスします。

リクエストパスの概要

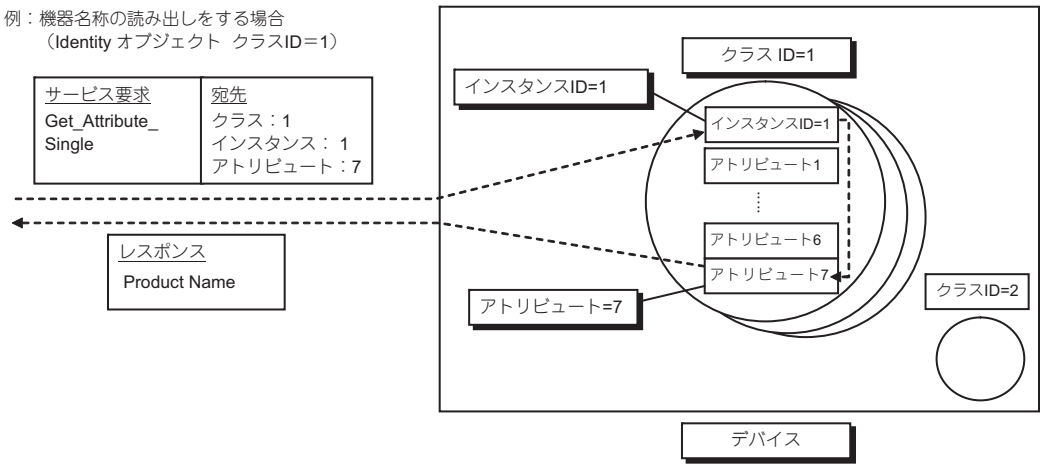
CIP の世界では、各デバイスは「オブジェクト」の集合としてモデル化されます。オブジェクトとは、デバイスの特定の構成要素を抽象的に表現したものです。



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）



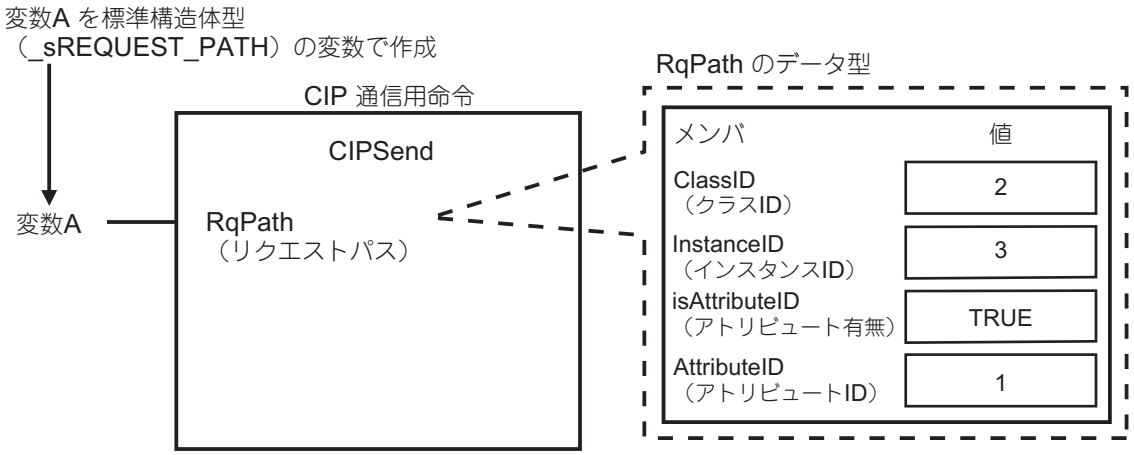
このように「クラス ID」、「インスタンス ID」、「アトリビュート ID」でデバイス内部の位置を特定するので「IOI (Internal Object Identifier)」と呼ばれます。
オブジェクトごとの「クラス ID」、「インスタンス ID」、「アトリビュート ID」、「サービスコード」については、📖「8-5 CIP オブジェクトサービス」を参照してください。

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、リクエストパスを格納するための変数を用意し、その変数にユーザプログラムでアクセスするオブジェクトを指定します。
リクエストパスの変数のデータ型には、オブジェクトの「クラス ID」、「インスタンス ID」、「アトリビュート ID」を指定する構造体型を用意しています。
構造体型には、標準構造体型（_sREQUEST_PATH）と拡張構造体型（_sREQUEST_PATH_EX）の 2 つがあります。拡張構造体型の場合、アクセスするオブジェクトの「クラス ID」、「インスタンス ID」、「アトリビュート ID」の値のサイズに合わせて指定することができます。標準構造体型の場合は 16bit 固定です。

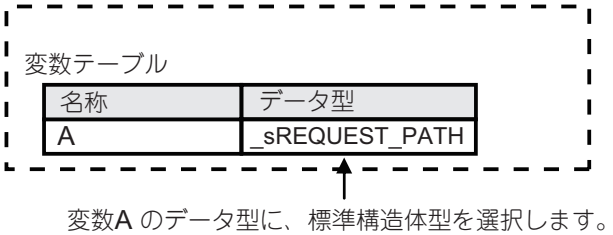
● 標準構造体型変数の場合

例）CIPSend 命令の RqPath（リクエストパス）を標準構造体型変数で値を入力する



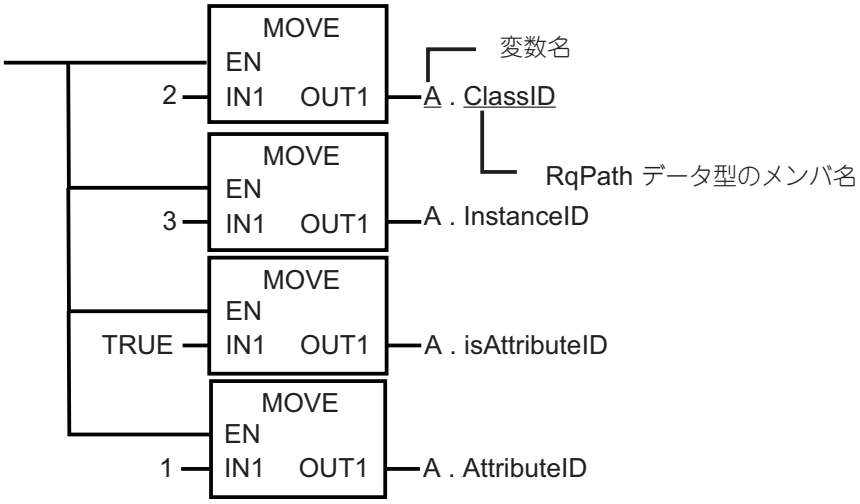
1 **標準構造体型変数を作成します。**

CIP 通信命令の RqPath（リクエストパス）に標準構造体型変数で値を入力するには、標準構造体型のユーザ変数を、あらかじめ作成しておく必要があります。
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 通信命令用の標準構造体型（_sREQUEST_PATH）を選択します。



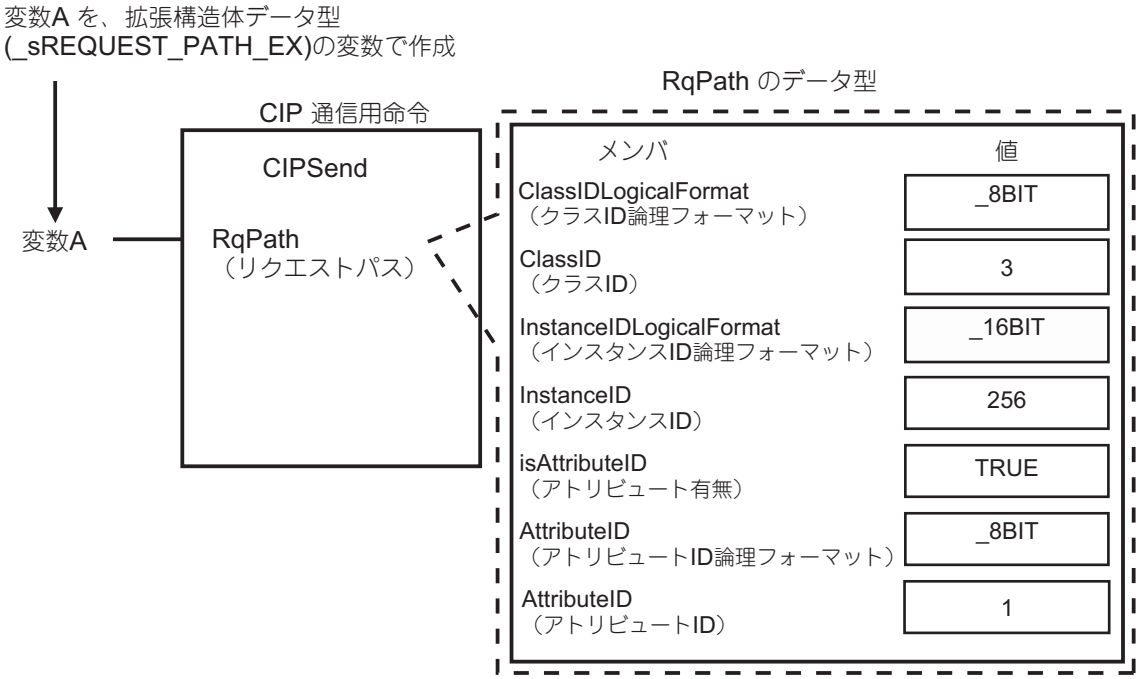
2 **標準構造体型変数の各メンバに値を入力します**

標準構造体型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。



● 拡張構造体型変数の場合

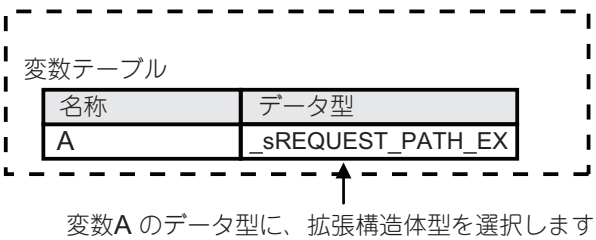
例）CIPSend 命令の RqPath（リクエストパス）を拡張構造体型変数で値を入力する



1 **拡張構造体型変数を作成します。**

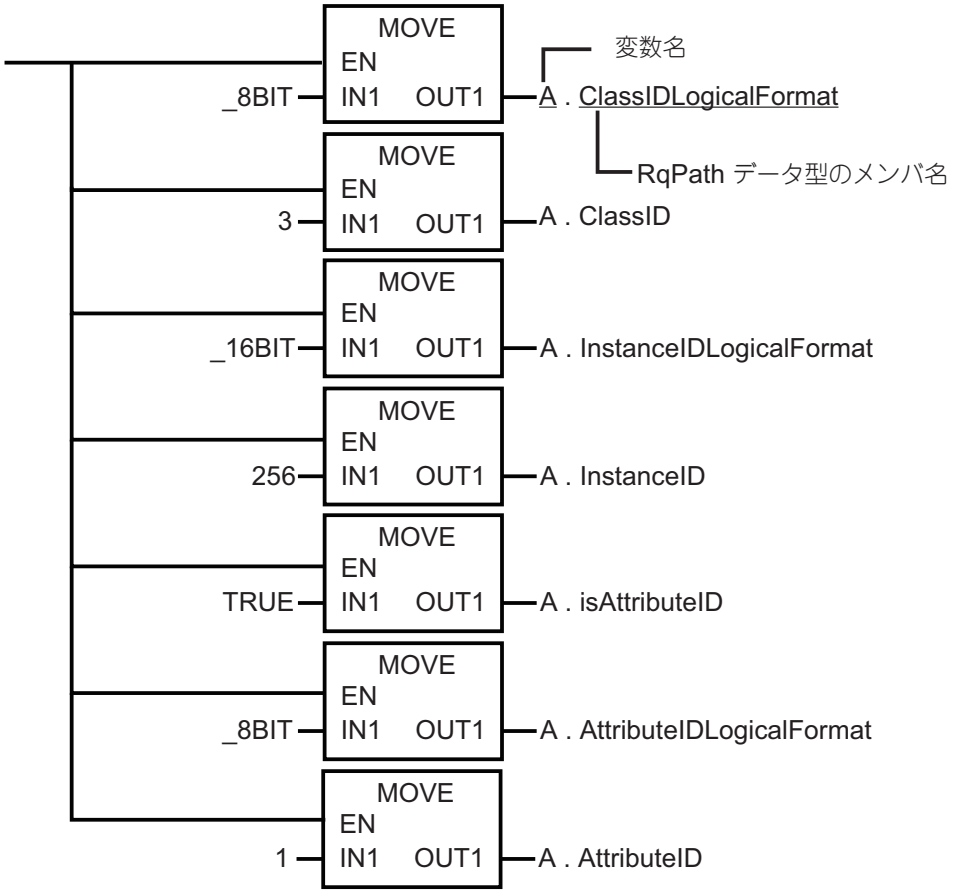
CIP 通信命令の RqPath（リクエストパス）に拡張構造体型変数で値を入力するには、拡張構造体型のユーザ変数を、あらかじめ作成しておく必要があります。

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 通信命令用の拡張構造体型（_sREQUEST_PATH_EX）を選択します。



2　拡張構造体型変数の各メンバに値を入力します

拡張構造体型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。



8-2-6　サービスデータ／レスポンスデータ

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。

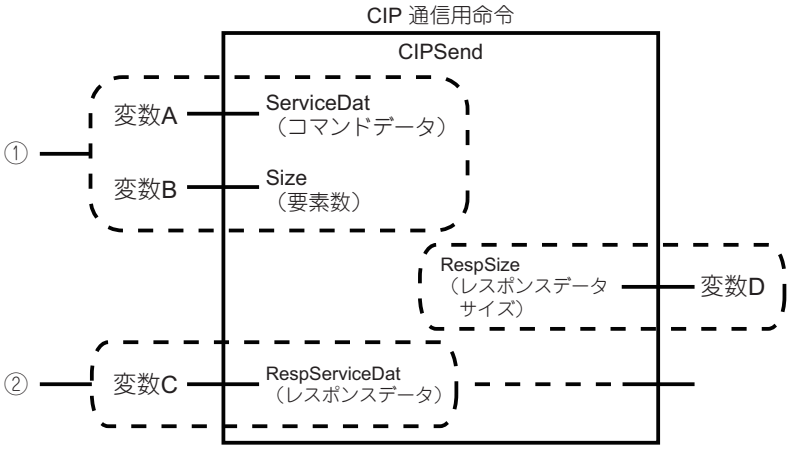
サービスデータ／レスポンスデータなどの入出力のための配列型変数の用意

CIP 通信命令で送受信するサービスデータおよびレスポンスデータを格納するための配列型変数について説明します。

● 配列型変数の作成

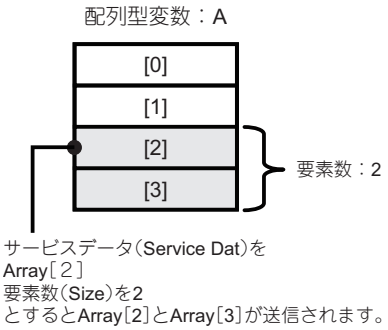
CIP 通信命令の配列型変数に値を入力するには、その配列型変数と同じ構造の変数を、あらかじめ作成しておく必要があります。

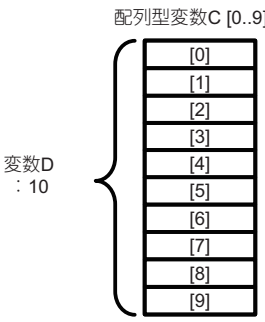
例）CIPSend 命令の配列型変数にデータを入力する変数を作成する



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）	−

8-2-7　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）

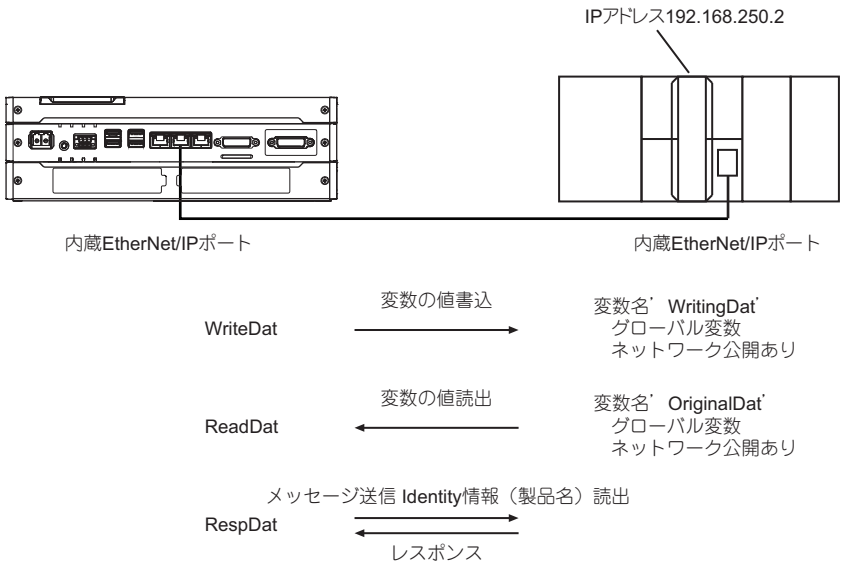
CIP の UCMM メッセージを使用して、変数書込、変数読出、メッセージ送信を行います。
コントローラどうしは、EtherNet/IP ネットワークで接続しており、相手ノードの IP アドレスは 192.168.250.2 とします。
処理の手順は、以下のようになります。

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。**

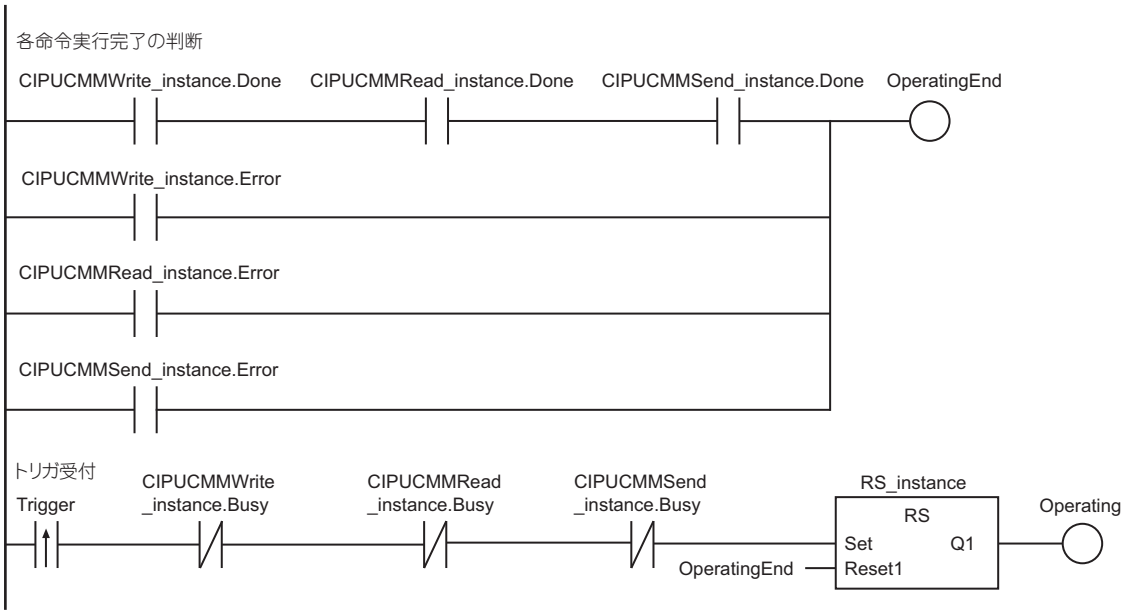
3 **CIPUCMMSend 命令を使用して、相手ノードに Explicit メッセージを送信します。メッセージの内容は、Identity 情報（製品名）読出です。その際のクラス ID、インスタンス ID、アトリビュート ID、サービスコードは、以下のとおりです。レスポンスデータは、変数 RespDat に格納します。**

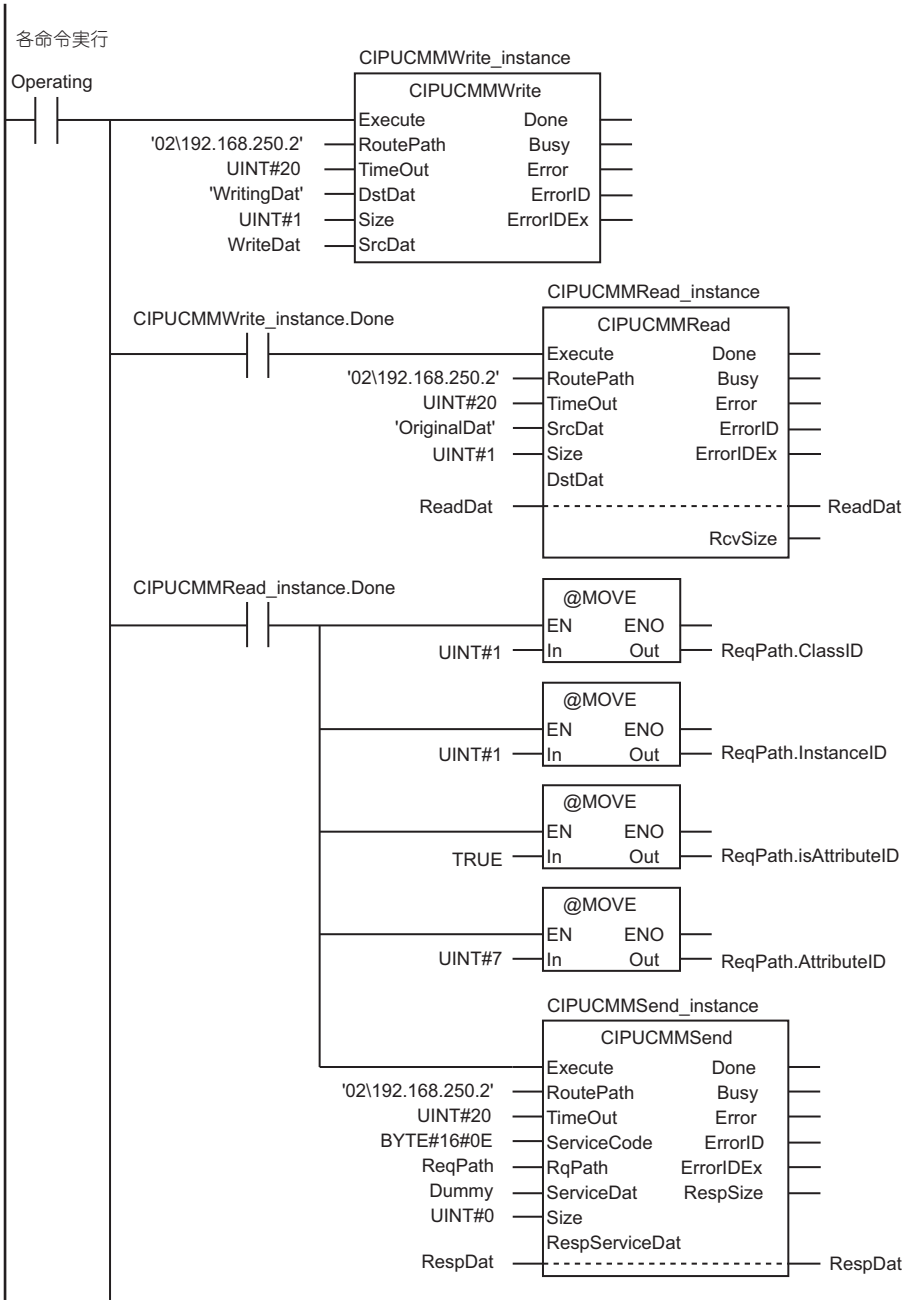
項目	値
クラス ID	1
インスタンス ID	1
アトリビュート ID	7
サービスコード	16#0E

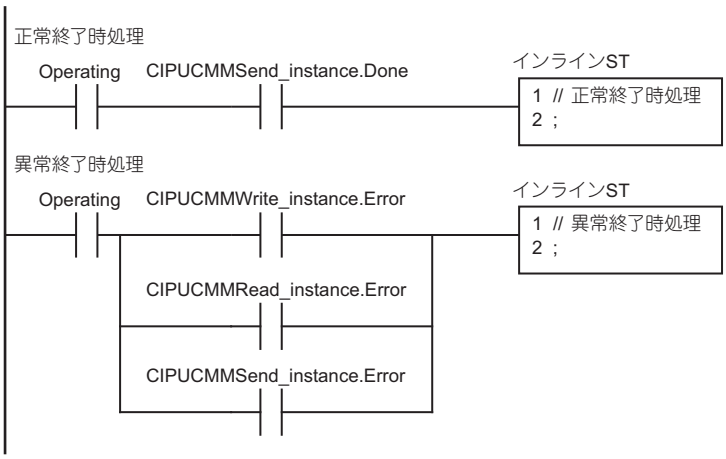


LD

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
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
'02\192.168.250.2'
RoutePath
Busy
UINT#20
TimeOut
Error
'WritingDat'
DstDat
ErrorID
UINT#1
Size
ErrorIDEx
WriteDat
SrcDat
CIPUCMMRead_instance
CIPUCMMWrite_instance.Done
CIPUCMMRead
Execute
Done
'02\192.168.250.2'
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
'02\192.168.250.2'
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
	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
	_EIP_EtnOnlineSta（*1）	BOOL	☑	オンライン

*1 使用する通信ポートによって、_EIP1_EtnOnlineSta（通信ポート 1 オンライン）、または _EIPIn1_EtnOnlineSta（内部通信ポート 1 オンライン）のいずれかに置き換えてください。

```
// Trigger の立ち上がりでシーケンス開始
IF ((Trigger=TRUE) AND (DoUCMMTrigger=FALSE) AND (_EIP_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   :='02\192.168.250.2',  // 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       :='02\192.168.250.2',  // 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       :='02\192.168.250.2',  // 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;
```

8-2-8 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(Large_Forward_Open) 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。**

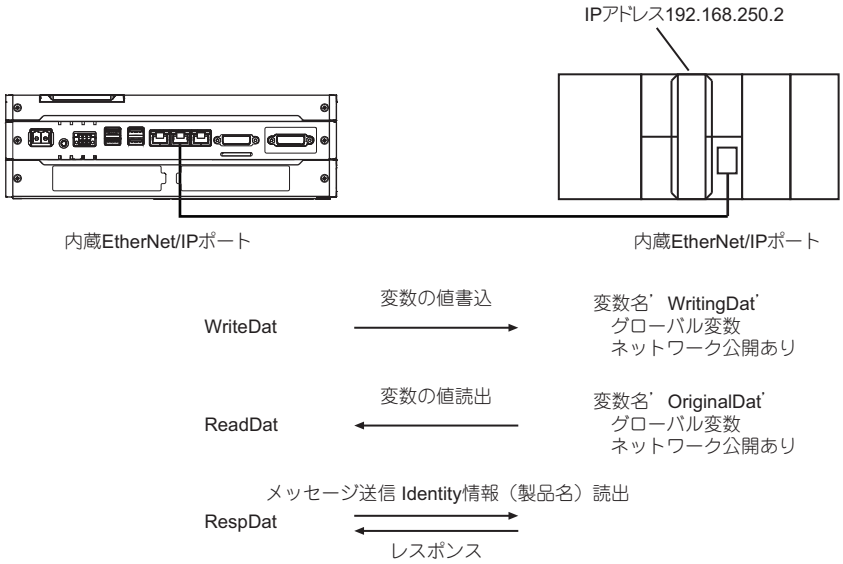
2 **CIPWrite 命令を使用して、相手ノードの変数に値を書き込みます。他局変数名は 'WritingDat'、書き込む値は、変数 WriteDat の中身とします。'WritingDat' は、相手ノードでグローバル変数として定義されており、かつ、ネットワーク公開されていることが必要です。**

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。**

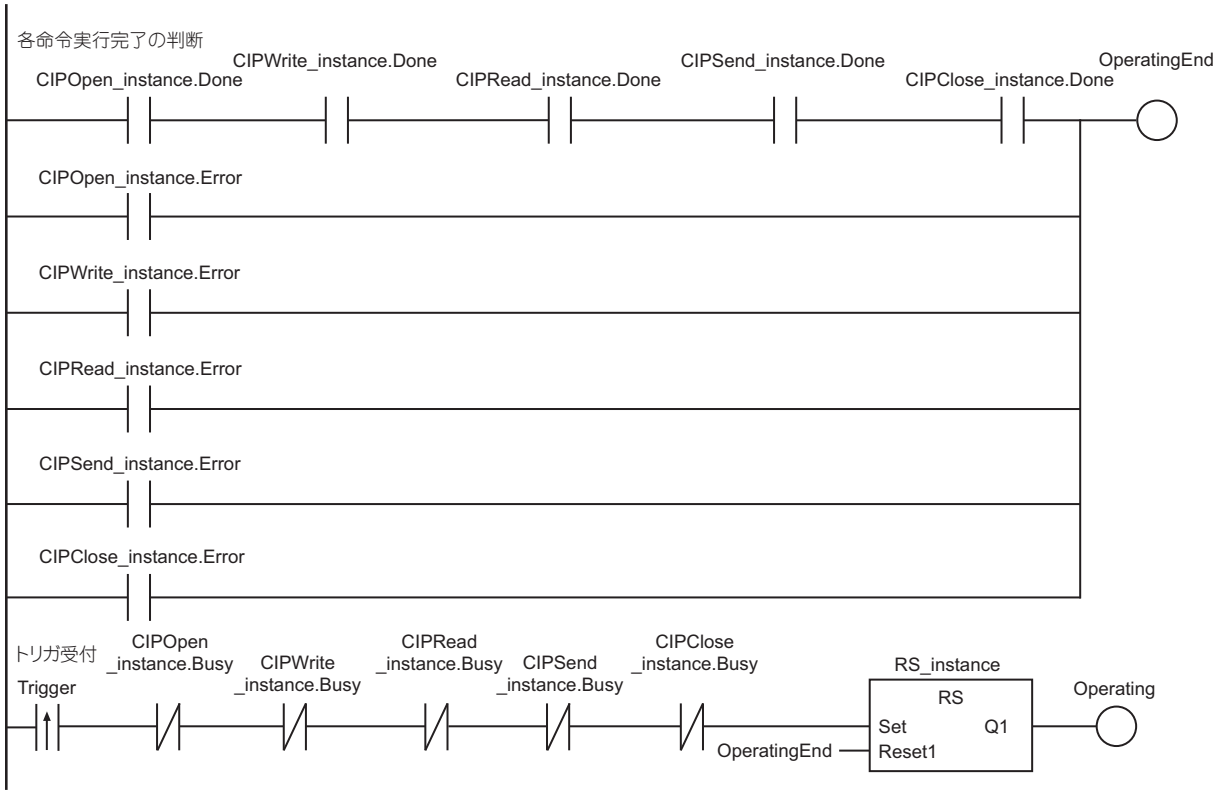
項目	値
クラス ID	1
インスタンス ID	1
アトリビュート ID	7
サービスコード	16#0E

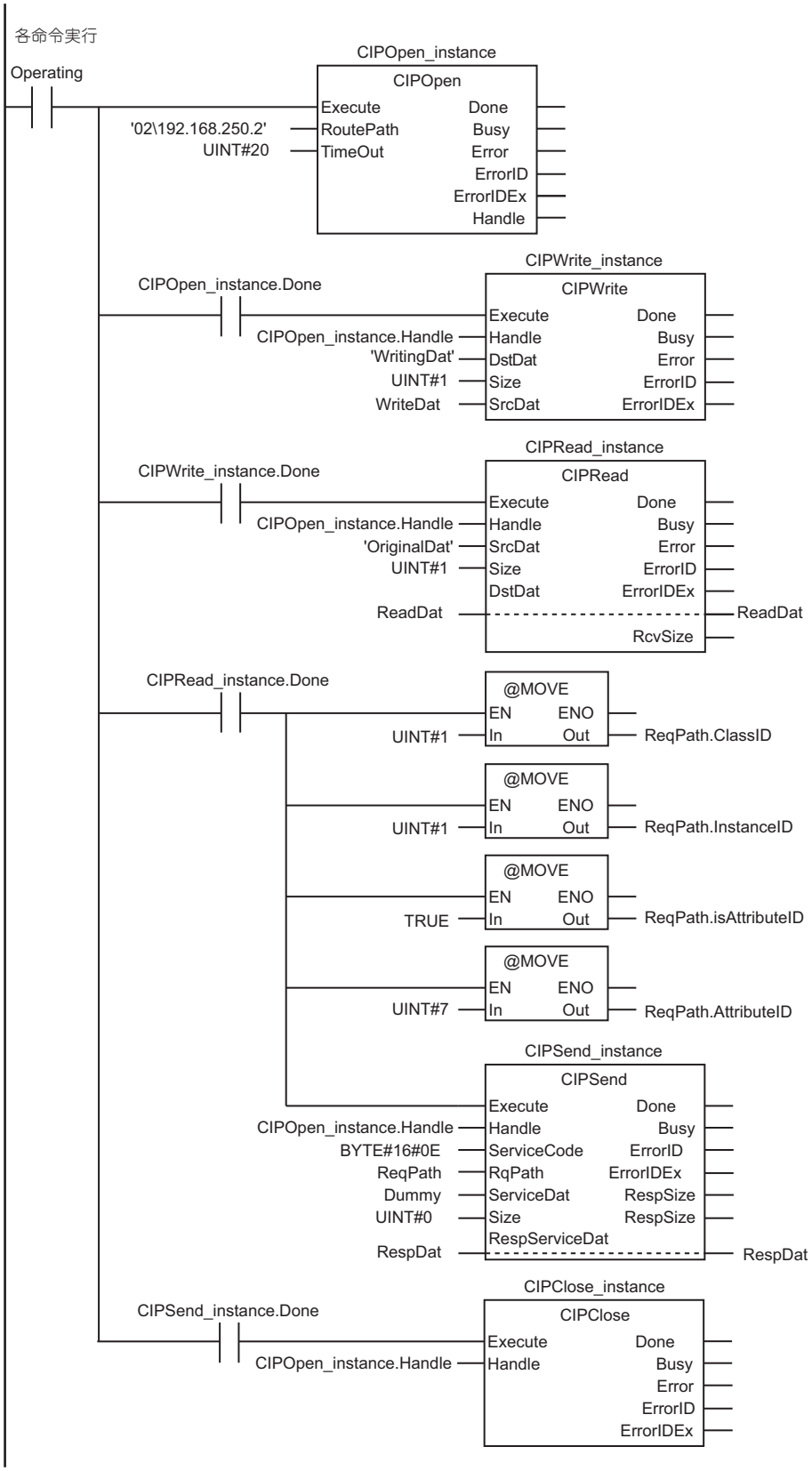
5 **CIPClose 命令を使用して、Class3 コネクションを切断します。**

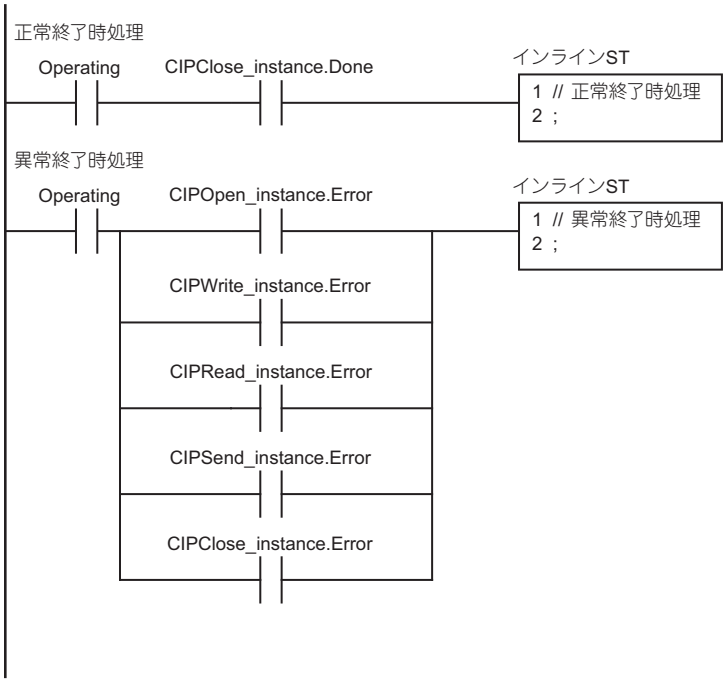


LD

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
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
RoutePath
TimeOut
Done
Busy
Error
ErrorID
ErrorIDEx
Handle
'02\192.168.250.2'
UINT#20
CIPWrite_instance
CIPWrite
CIPOpen_instance.Done
CIPOpen_instance.Handle
'WritingDat'
UINT#1
WriteDat
Execute
Handle
DstDat
Size
SrcDat
Done
Busy
Error
ErrorID
ErrorIDEx
CIPRead_instance
CIPRead
CIPWrite_instance.Done
CIPOpen_instance.Handle
'OriginalDat'
UINT#1
ReadDat
Execute
Handle
SrcDat
Size
DstDat
Done
Busy
Error
ErrorID
ErrorIDEx
ReadDat
RcvSize
CIPRead_instance.Done
@MOVE
EN
In
ENO
Out
UINT#1
ReqPath.ClassID
@MOVE
UINT#1
ReqPath.InstanceID
@MOVE
TRUE
ReqPath.isAttributeID
@MOVE
UINT#7
ReqPath.AttributeID
CIPSend_instance
CIPSend
CIPOpen_instance.Handle
BYTE#16#0E
ReqPath
Dummy
UINT#0
RespDat
Execute
Handle
ServiceCode
RqPath
ServiceDat
Size
RespServiceDat
Done
Busy
ErrorID
ErrorIDEx
RespSize
RespSize
RespDat
CIPClose_instance
CIPClose
CIPSend_instance.Done
CIPOpen_instance.Handle
Execute
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
	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
	_EIP_EtnOnlineSta（*1）	BOOL	☑	オンライン

*1 使用する通信ポートによって、_EIP1_EtnOnlineSta（通信ポート 1 オンライン）、または _EIPIn1_EtnOnlineSta（内部通信ポート 1 オンライン）のいずれかに置き換えてください。

```
// Trigger の立ち上がりでシーケンス開始
IF ((Trigger=TRUE) AND (DoCIPTrigger=FALSE) AND (_EIP_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(Large_Forward_Open) 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:='02\192.168.250.2');      // 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;

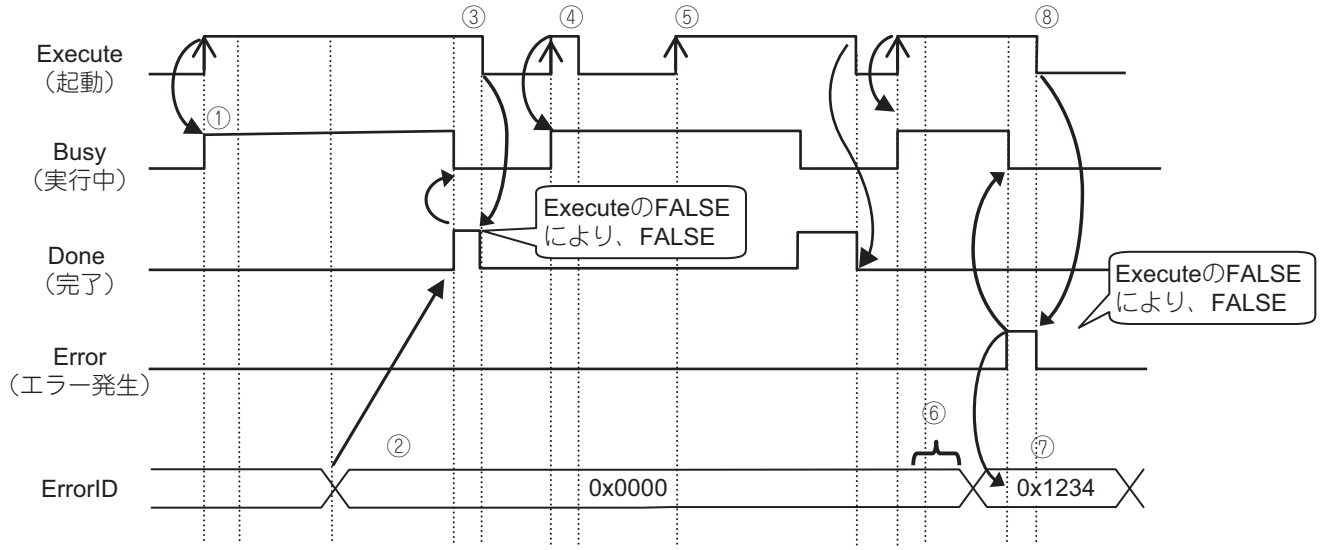
    0:                                                  // 正常終了処理
        DoCIPTrigger:=FALSE;
        Trigger        :=FALSE;

    ELSE                                                // 異常終了処理
        DoCIPTrigger:=FALSE;
        Trigger        :=FALSE;
    END_CASE;
END_IF;
```

8-2-9　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。

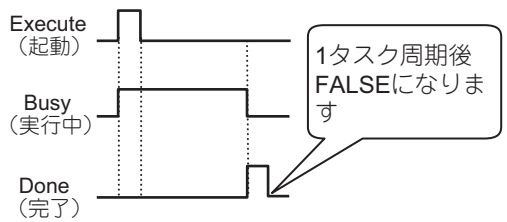


①Execute の立上りで、命令が実行され、Busy が TRUE になります。
②命令の実行結果が、出力変数に反映されると、Done が TRUE、Busy が FALSE になります。
③Execute を FALSE にすると、Done も FALSE に戻ります。
④再び Execute が立ち上がると、Busy が TRUE になります。
⑤命令実行中 (Busy = TRUE) に Execute が立ち上がった場合、無視されます。
⑥エラーが発生すると、内部ではリトライを数回実行します。
　リトライ中は、ErrorID にエラーが更新されません。
⑦通信エラーが発生した場合、Error 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。

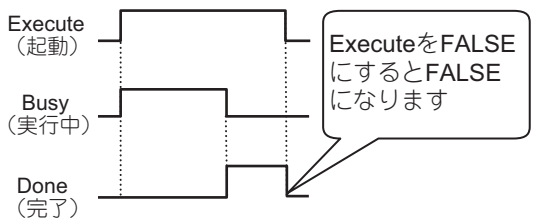
使用上の注意

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)

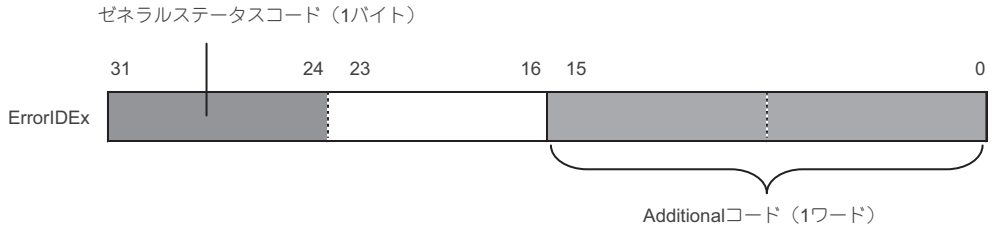


8-2-10　レスポンスコード

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）

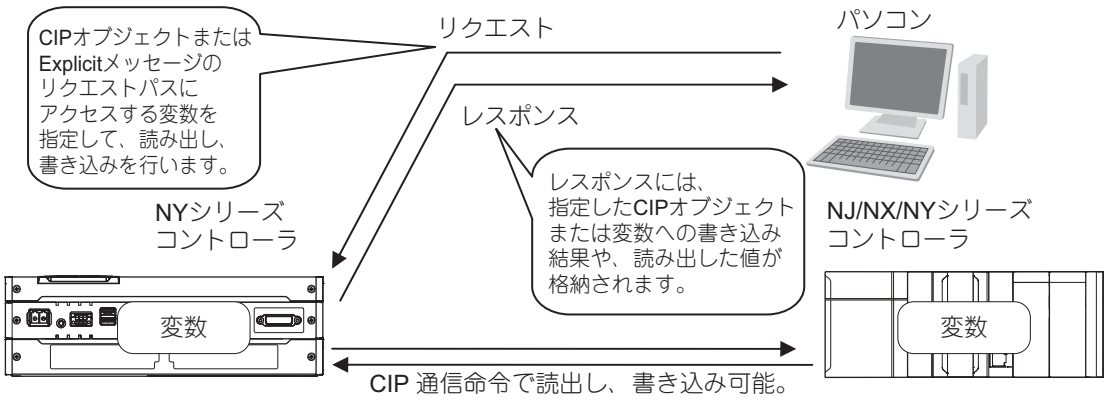
General Status (Hex)	Additional Status (Hex)	Explanation
01	0100	コネクション使用中、または重複 Forward_Open。
01	0103	サポートされていないトランスポートクラスとトリガの組み合わせ。
01	0106	所有権の不一致
01	0107	ターゲットアプリケーションでコネクションが見つからない。
01	0108	無効なコネクションタイプ。コネクションタイプまたはコネクションの優先順位のいずれかに問題がある。
01	0109	無効なコネクションサイズ。
01	0110	デバイスが設定されていない。
01	0111	RPI 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	0127	無効なコネクションサイズ（オリジネータ→ターゲットの向きのデータ）。
01	0128	無効なコネクションサイズ（ターゲット→オリジネータの向きのデータ）。
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
01	0813	ネットワークアドレスが異なるため、マルチキャストコネクションを使用できない。

8-3　CIP メッセージ通信のサーバ機能

NY シリーズコントローラは、外部から CIP メッセージを受信すると、コントローラが内蔵している指定オブジェクトのサービスを実行することができます。
この機能を、「CIP メッセージ通信のサーバ機能」と呼びます。

本章では、CIP メッセージ通信のサーバ機能を利用して、次のプログラムなどでコントローラに CIP オブジェクトの読出し、書き込みまたは変数の値を読出し、書き込みをさせるために発行する、CIP メッセージの記述や仕組みについて記載します。
・ EtherNet/IP ネットワーク上のパソコンのプログラム
・ NY シリーズ産業用 PC の Windows のプログラム

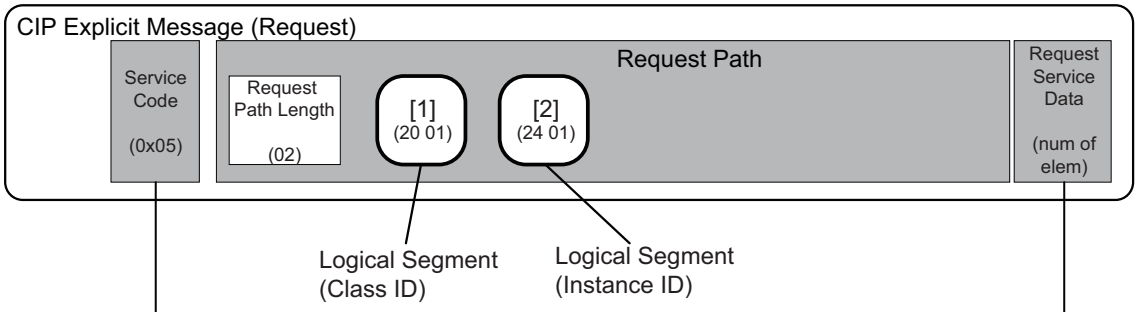
なお、NJ/NX/NY シリーズコントローラ間で、CIP オブジェクトの読出し、書き込みまたは変数の読出し、書き込みを行う場合は、CIP 通信命令で行ってください。
CIP 通信命令による CIP 通信命令についての詳細は🕮「8-2 CIP メッセージ通信のクライアント機能」を参照してください。



8-3-1　CIP オブジェクトにアクセスするための CIP メッセージ構造

CIP オブジェクトにアクセスするための指定方法を示します。
アクセス対象の CIP オブジェクトは、Explicit メッセージのリクエストパスのフィールド上に CIP Common Spec が定める各種セグメントを連結した形式で表現します。

例：　Identity オブジェクト（クラス ID: 01Hex）のインスタンス (01Hex) へサービス Reset(0x05) を実行します。



実行するサービスコードを指定します。
サービスコードについては、🕮「8-5 CIP オブジェクトサービス」を参照してください。

アクセスするクラス ID. インスタンス ID を指定します。

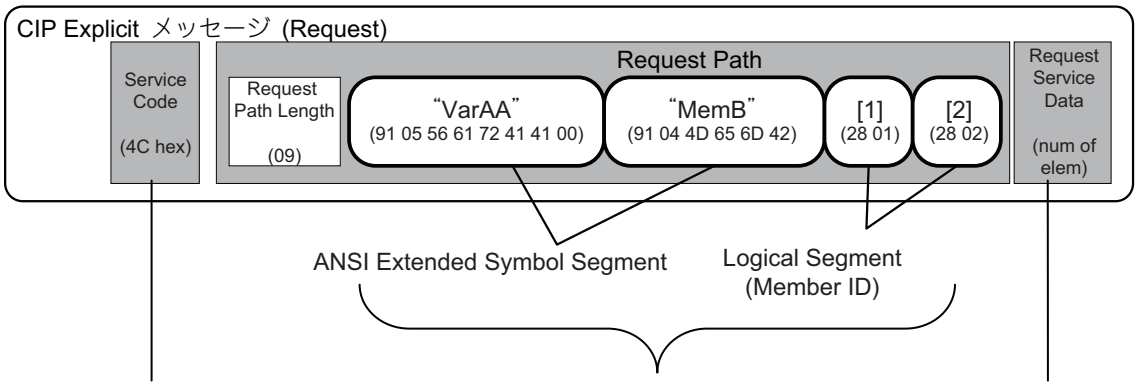
サービスに応じたリクエストデータを指定します。

8-3-2　変数にアクセスするための CIP メッセージ構造

変数にアクセスするための指定方法を示します。
アクセス対象の変数は、Explicit メッセージのリクエストパスのフィールド上に記述できるよう、CIP Common Spec が定める各種セグメントを連結した形式で表現します。

以下の要素の組み合わせで指定します。
対象変数の指定：CIP のセグメントに要素を格納、それを連結して表現

例 ： 変数 VarAA.MemB[1.2] (構造型変数内の 1 メンバ) の現在値読み出し
Variable Object の CIP Read Data Servcie を使う例



変数の読出し、書込みを実行するサービスコードを指定します。
サービスコードについては、📖「8-6 変数への読出し／書込みサービス」を参照ください。

アクセスする変数名、範囲を指定します。
変数名の指定方法については、📖「8-4-5 リクエストパスでの変数名の指定方法」を参照ください。

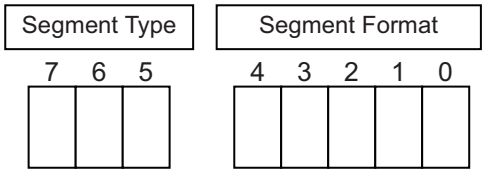
読出し、書込むデータの書式を指定します。
データ書式の指定方法については、📖「8-7 変数のデータタイプ一覧」を参照ください。

8-4　リクエストパスの指定

リクエストパスで、CIP オブジェクト、変数名、構造体メンバ名、配列のインデックスを指定します。
リクエストパスを表記するのに CIP では EPATH 型による表記を行います。
これは、リクエストパスをセグメントで区切りそこに値を記述していく方式です。
そのため、リクエストパスの表記はセグメントと呼ばれるデータをつなげて最終目的地を示します。
セグメントには、そのセグメントの種類を示す情報と、セグメントのデータが含まれます。

Segment 1	Segment 2	Segment 3	Segment 4	・・・・

あるセグメントの解釈方法はそのセグメントの最初の 1 バイトにあり、「Segment Type」3 ビットと「Segment Format」5 ビットの 2 つの部分から構成されています。



Segment Type の仕様は CIP 仕様で以下のように決められています。

Segment Type			説明
7	6	5	
0	0	0	Port Segment
0	0	1	Logical Segment
0	1	0	Network Segment
0	1	1	Symbolic Segment
1	0	0	Data Segment
1	0	1	Data Type
1	1	0	Data Type
1	1	1	Reserved

Segment Format の部分の仕様は Segment Type ごとに異なります。これらの Segment Format は任意のデバイスの任意のオブジェクトに対してサービスを要求するときに使用します。
以降では、CIP メッセージ通信にて変数を指定する上で必要となる Logical Segment、Data Segment について記述します。

8-4-1　CIP オブジェクトの指定例

リクエストパスでは、Logical Segment のセグメントをつなげて、以下のようにアクセスするオブジェクト機能を指定します。

Logical Segment（Class ID）	Logical Segment（Instance ID）	Logical Segment（Attribute ID）
クラスIDを指定	インスタンスIDを指定	アトリビュートIDを指定

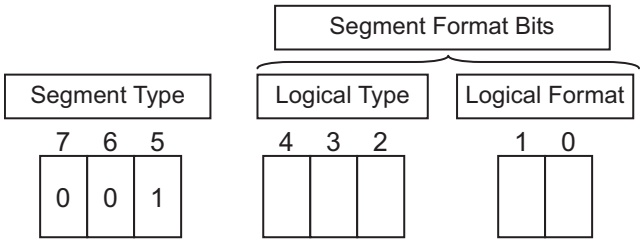
8-4-2　変数の指定例

リクエストパスでは、各種類のセグメントをつなげて、以下のようにアクセスする変数を指定します。

Data Segment（ANSI Extended Symbol Segment）	Logical Segment（Member ID）
変数名・メンバ名を指定	配列インデックスを指定

8-4-3　Logical Segment

リクエストパスにて、CIP オブジェクトまたは変数の範囲（配列）を記述するときなどに使用します。



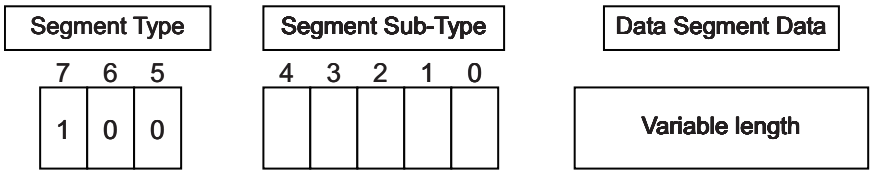
Logical Type			説　明
4	3	2	
0	0	0	Class ID
0	0	1	Instance ID
0	1	0	Member ID
0	1	1	Connection Point
1	0	0	Attribute ID
1	0	1	Special (Do not use the logical addressing definition for the Logical Format.)
1	1	0	Service ID (Do not use the logical addressing definition for the Logical Format.)
1	1	1	Reserved

Logical Format		説　明
1	0	
0	0	8 bit logical address
0	1	16 bit logical address
1	0	32 bit logical address
1	1	Reserved

Class ID, Attribute ID には 8-bit , 16-bit logical address が利用できます。
Instance ID には、8-bit , 16-bit , 32-bit logical address が利用できます。

8-4-4　Data Segment

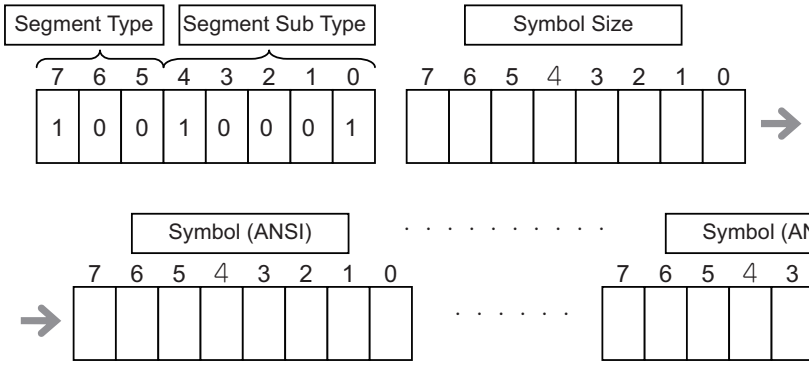
リクエストパスにて指定する変数名を指定するときなどに使用します。



Segment Sub-Type					説明
4	3	2	1	0	
0	0	0	0	0	Simple Data Segment
1	0	0	0	1	ANSI Extended Symbol Segment

Data Segment で利用するのは、主に ANSI Extended Symbol Segment です。
このタイプのセグメントを使用して変数データの読み書きを行います。

ANSI Extended Symbol Segment



8-4-5　リクエストパスでの変数名の指定方法

変数名

変数名は、Symbolic Segment (ANSI Extended Symbol Segment) にて指定します。

(変数名指定フォーマット)

BYTE	91 Hex	ANSI EXTENDED SYMBOL SEGMENT
BYTE	Length in BYTE	変数名長 (単位バイト)
Array of octet	: 変数名 :	変数名 (encoding は UTF-8)
Octet	(pad)	00 Hex。変数名が奇数長なら 1BYTE pad

変数名

UTF-8 で記述します。

構造体メンバ名

構造体メンバ名の指定方法は、変数名と同じです。
(ANSI Extended Symbol Segment に UTF-8 の文字列を格納します。)

配列インデックス

配列インデックスは、Logical Segment (Member ID) で指定します。
変数名の中に配列インデックス（”[x]”）を指定することも可能です。

（指定方法 1 ：インデックス長 8bit）

BYTE	28 Hex	LOGICAL SEGMENT (Member ID)
USINT	Index	配列インデックス (0 ～ 255)

（指定方法 2：インデックス長 16bit）

BYTE	29 Hex		LOGICAL SEGMENT (Member ID)
octet	00 Hex		Pad
UINT	Index	(L)	配列インデックス (0 ～ 65535)
		(H)	

Num of Element を使った範囲指定

「変数読出しサービス」と「変数書き込みサービス」は、そのリクエストデータ中に num of element フィールドを持ちます。これらのサービスでは、以下のように指定することで、配列の指定範囲にアクセスできます。

- 配列型変数内のアクセス範囲の先頭要素を対象として指定している
- アクセスする要素数を Num of Element フィールドに指定する

指定例

以下のような構造体型変数で、VarAA.MemB[1.2] を指定する場合

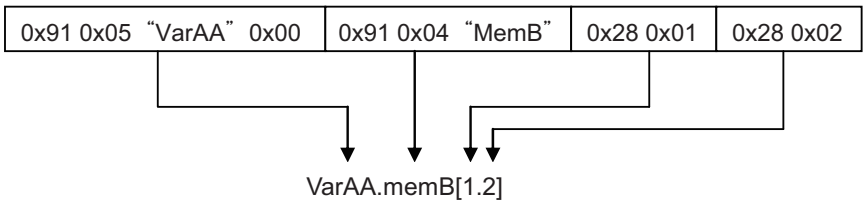
```
struct
{
   UINT       MemA;
   BOOL       MemB[10][10];
} VarAA;
```

(変数名指定フォーマット)

BYTE	91 Hex	ANSI EXTENDED SYMBOL SEGMENT
BYTE	05 Hex	変数名長 (単位：バイト)
Array of octet	'V'	変数名
	'a'	
	'r'	
	'A'	
	'A'	
Octet	00 Hex	Pad
BYTE	91 Hex	ANSI EXTENDED SYMBOL SEGMENT
BYTE	04 Hex	変数名長 (単位：バイト)
Array of octet	'M'	変数名
	'e'	
	'm'	
	'B'	
BYTE	28 Hex	LOGICAL SEGMENT (Member ID)
USINT	01 Hex	配列インデックス (1 次元目)
BYTE	28 Hex	LOGICAL SEGMENT (Member ID)
USINT	02 Hex	配列インデックス (2 次元目)

Symbolic Segment (ANSI Extended Symbol Segment) にて指定された変数名を、Communication スレッドに渡す場合には、変数名を文字列に変換する必要があります。変換ルールを以下に示します。

構造体メンバ・配列要素の指定例



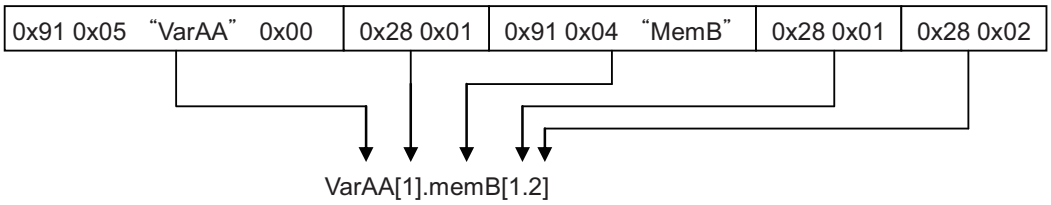
以下のような構造体型変数で、VarAA[1].MemB[1.2] を指定する場合

```
struct
{
   UINT        MemA;
   BOOL        MemB[10][10];
} VarAA[3];
```

(変数名指定フォーマット)

BYTE	91 Hex	ANSI EXTENDED SYMBOL SEGMENT
BYTE	05 Hex	変数名長 (単位バイト)
Array of octet	'V'	変数名
	'a'	
	'r'	
	'A'	
	'A'	
Octet	00 Hex	Pad
BYTE	28 Hex	LOGICAL SEGMENT (Member ID)
USINT	01 Hex	配列インデックス
BYTE	91 Hex	ANSI EXTENDED SYMBOL SEGMENT
BYTE	04 Hex	変数名長（単位バイト）
Array of octet	'M'	変数名
	'e'	
	'm'	
	'B'	
BYTE	28 Hex	LOGICAL SEGMENT (Member ID)
USINT	01 Hex	配列インデックス（１次元目）
BYTE	28 Hex	LOGICAL SEGMENT (Member ID)
USINT	02 Hex	配列インデックス（２次元目）

構造体配列の指定例



8-5　CIP オブジェクトサービス

リクエストパスに CIP オブジェクトを指定し、NY シリーズの CIP メッセージサーバ機能にアクセスするサービスを記載します。

8-5-1　内蔵 EtherNet/IP ポート宛 CIP オブジェクトの種類

内蔵 EtherNet/IP ポート宛 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	• コントローラの識別情報の読出し • 内蔵 EtherNet/IP ポートおよび内部通信ポートのリセット	P.8-41
TCP/IP Interface オブジェクト	• TCP/IP 設定を書込み／読出し	P.8-44
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.8-47
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.8-51

8-5-2　Identity オブジェクト（クラス ID：01Hex）

コントローラの識別情報の読出しや、内蔵 EtherNet/IP ポートおよび内部通信ポートをリセットするオブジェクトです。

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	内蔵 EtherNet/IP ポートおよび内部通信ポー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、ポートをリセットして、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：ポートをリセットします。 02 Hex：保存されているタグデータリンク設定をクリアして、ポートをリセットします。	×	○

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	詳細は下表の（1）を参照
04 Hex	Revision	機器のリビジョン	読出	Struct	－
	Major Revision	メジャー・リビジョン	読出	USINT	詳細は下表の（2）を参照
	Minor Revision	マイナー・リビジョン	読出	USINT	
05 Hex	Status	内蔵 EtherNet/IP ポートのステータス	読出	WORD	詳細は下表の（3）を参照
06 Hex	Serial Number	シリアル番号	読出	UDINT	設定値
07 Hex	Product Name	製品名	読出	STRING	設定値

(1) 形式ごとの製品コード

形式	製品コード
NY512-□□□□	0685 Hex
NY532-□□□□	0686 Hex

(2) ユニットバージョンごとの 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
Ver.1.12 ～ 1.19	02 Hex	05 Hex
Ver.1.21 以降		06 Hex

(3)　内蔵 EtherNet/IP ポートのステータス詳細

ビット	名称	説明
0	Owned	内蔵 EtherNet/IP ポー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	内蔵 EtherNet/IP ポートの状態を示す 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 • IP ルートテーブル設定異常 • DNS サーバ接続失敗 • タグデータリンク設定エラー • タグデータリンクタイムアウト • タグデータリンクコネクション確立タイムアウト • FTP サーバ設定エラー • SNMP 設定エラー • 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
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 サーバ接続失敗 • Ethernet 基本設定エラー • IP アドレス設定異常
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 • 通信コントローラ故障 • MAC アドレス異常
12 ～ 15	予約	常に FALSE

オブジェクトを指定するリクエストパス（IOI）

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

8-5-3　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。

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、内蔵 EtherNet/IP ポー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。	×	○

クラス ID

F5 Hex を指定します。

インスタンス ID

00 ～ 02 Hex を指定します。
00 : クラス指定
01 : 内蔵 EtherNet/IP ポート
02 : 内部通信ポート

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	0004 Hex
02 Hex	Max Instance	最大のインスタンス番号	読出	UINT	0002 Hex
03 Hex	Num of Instance	オブジェクトインスタンスの数	読出	UINT	0002 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	インタフェースの 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 起動中を含む）状態 1 ＝IP アドレスが設定されている状態 bit4 ～ 5：予約（FALSE 固定） bit6：AcdStatus： FALSE ＝ IP アドレス衝突未検知の状態 TRUE ＝ IP アドレス衝突検知の状態 bit7 ～ 31：予約（FALSE 固定）
02 Hex	Configuration Capability	インタフェース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： 内蔵 EtherNet/IP ポート =TRUE 内部通信ポート =FALSE 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：Hardware Configurable：FALSE（固定） bit6：Interface Configuration Change Requires Reset：FALSE（固定） bit7：ACD Capable：TRUE（固定） bit8 ～ 31：予約（FALSE 固定）
03 Hex	Configuration Control	インタフェースの起動時の IP アドレスの設定方法を設定します	読出／書込	DWORD	bit0 ～ 3：IP アドレス設定方法 0 ＝固定 IP アドレス設定 1 ＝BOOTP による設定 bit4：DNS 有効／無効設定 FALSE ＝ DNS 無効 TRUE ＝ DNS 有効 bit5 ～ 31：予約（FALSE 固定）
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	内蔵 EtherNet/IP ポート = 20 F6 24 01 Hex 内部通信ポート = 20 F6 24 02 Hex
05 Hex	Interface Cofiguration	インタフェースの設定	読出／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[*1]
06 Hex	Host Name	ホスト名称（予約）	読出／書込	STRING	設定値[*2]

*1. ドメイン名のサイズ（2Byte）＋ドメイン名（最大 48Byte）です。
*2. ホスト名称のサイズ（2Byte）＋ホスト名称（最大 64Byte）です。

オブジェクトを指定するリクエストパス（IOI）

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 ～ 02Hex	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			

8-5-4　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。

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。	×	○

クラス ID

F6 Hex を指定します。

インスタンス ID

00 ～ 02 Hex を指定します。
00 : クラス指定
01 : 内蔵 EtherNet/IP ポート
02 : 内部通信ポート

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	0004 Hex
02 Hex	Max Instance	最大のインスタンス番号	読出	UINT	0002 Hex
03 Hex	Num of Instance	オブジェクトインスタンスの数	読出	UINT	0002 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	インタフェース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	インタフェース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	インタフェースの 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

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	Interface Counters	インターフェース上で送受信したパケット数	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
	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		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	通信ポー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

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
06 Hex	Interface Control	インタフェースの制御設定	読出／書込	Struct	－
	Control Bits	Ethernet 通信の Auto 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
0C Hex	HC Interface Counters	大容量通信インターフェース上でのパケット受信に関連するカウンタ	読出	Struct	－
	HCInOctets	インタフェース上で受信したオクテット数。このカウンタは In Octets の 64 ビット版です。		ULINT	現在値を読み出す
	HCInUnicastPkts	インタフェース上で受信したユニキャストパケット数。このカウンタは Ucast Packets の 64 ビット版です。		ULINT	現在値を読み出す
	HCInMulticastPkts	インタフェース上で受信したマルチキャストのパケット数		ULINT	現在値を読み出す
	HCInBroadcastPkts	インタフェース上で受信したブロードキャストのパケット数		ULINT	現在値を読み出す
	HCOutOctets	インタフェース上で送信したオクテット数		ULINT	現在値を読み出す
	HCOutUnicastPkts	インタフェース上で送信したユニキャストパケット数。このカウンタは Out Octets の 64 ビット版です。		ULINT	現在値を読み出す
	HCOutMulticastPkts	インタフェース上で送信したマルチキャストパケット数		ULINT	現在値を読み出す
	HCOutBroadcastPkts	インタフェース上で送信したブロードキャストパケット数		ULINT	現在値を読み出す
0D Hex	HC Media Counters	通信ポートの大容量メディアカウンタ	読出	Struct	-
	HCStatsAlignmentErrors	長さがオクテット整数でない受信フレーム数。このカウンタは Alignment Errors の 64 ビット版です。		ULINT	現在値を読み出す
	HCStatsFCSErrors	FCS チェックに合格しない受信フレーム数。このカウンタは FCS Errors の 64 ビット版です。		ULINT	現在値を読み出す
	HCStatsInternalMacTransmitErrors	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。このカウンタは MAC Transmit Errors の 64 ビット版です。		ULINT	現在値を読み出す
	HCStatsFrameTooLongs	最大許容フレームサイズを超えた受信フレーム数。このカウンタは Frame Too Long の 64 ビット版です。		ULINT	現在値を読み出す
	HCStatsInternalMacReceiveErrors	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。このカウンタは MAC Receive Errors の 64 ビット版です。		ULINT	現在値を読み出す
	HCStatsMASymbolErrors	内部の MAC 副層シンボルエラーのため、インタフェースで受信に失敗したフレーム数		ULINT	現在値を読み出す

(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 TRUE：オートネゴシエーション有効
1	ForcedDuplex Mode	FALSE：半２重　　TRUE：全２重 * 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）

オブジェクトを指定するリクエストパス（IOI）

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 ～ 02Hex	• クラスのアトリビュートを読み出す場合：01 ～ 03Hex • インスタンスのアトリビューを読み書きする場合：01 ～ 06Hex、0CHex、0DHex
10 Hex	Set_Attribute_Single			
4C Hex	Get_and_Clear			値を 0 クリアするアトリビュートを指定：04Hex、05Hex、0CHex、0DHex

8-5-5　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。

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。	○	×
51 Hex	Reset_System_Alarm_All	コントローラの全ての異常を解除します。	○	×

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。

● 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。	読出／書込	UINT	アトリビュート書込み時に指定します。 0001 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

オブジェクトを指定するリクエストパス（IOI）

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			

8-6　変数への読出し／書込みサービス

リクエストパスに変数を指定し、NY シリーズの CIP メッセージサーバ機能にアクセスするサービスを記載します。

8-6-1　変数読出しサービス

サービスコードに 4C Hex を指定して、リクエストパスで指定した変数の値を読み出すことができます。

サービスコード：4C Hex

● **リクエストデータフォーマット**

Request Path Data

変数名指定

Request Service Data

+0	Num of Element	(L)	UINT	配列型変数に対し、読み出す要素数を指定する。
+1		(H)		

Response Service Data

Data Type	USINT	読み出した変数の型
AddInfo Length	USINT	付加情報（フィールドの長さ：単位はバイト）
(AddInfo)		付加情報（構造体の CRC 値）
実データ *		

* 実データはリトルエンディアンで格納されます。

Data Type	読み出した変数のデータ型コード。📖「8-7-1 データタイプコード」を参照。
AddInfoLength	構造体型変数にアクセスする場合のみ、AddInfo 領域のサイズが格納される。 構造体型の場合、02Hex となる。それ以外は 00Hex。
AddInfo	構造体型変数にアクセスする場合のみ、構造体定義の CRC コードが格納される。この場合、AddInfo のサイズは 2 バイト。
実データ	リトルエンディアンで指定する。 配列型で、0001 Hex を指定した場合は、その配列を構成する要素のデータ型の変数にアクセスした場合と同様のフォーマットで実データが格納される。

レスポンスコード

CIP ステータス	内容	Add ステータス	原因
00	SUCCESS	-	正常終了
02	RESOURCE_UNAVAILABLE	-	内部処理用バッファが現在利用できない
04	PATH_SEGMENT_ERROR	-	Request Path 指定が間違っている
05	PATH_DESTINATION_UNKNOWN	-	変数の宛先指定が誤っている。
0C	OBJECT_STATE_CONFLICT	8010	ダウンロード中、起動中
		8011	タグメモリ異常中
11	REPLY_DATA_TOO_LARGE	-	応答（レスポンス）が最大応答長を超えている
13	NOT_ENOUGH_DATA	-	指定サービスのデータ長が不足している
15	TOO_MUCH_DATA	-	指定サービスのデータ長が多すぎる

CIP ステータス	内容	Add ステータス	原因
1F	VENDOR_SPECIFIC_ERROR	0102,2104	読出し対象が、読出し不可の変数 I/O である
		0104,1103	セグメント領域を超えるアドレス、サイズが指定されている
		8001	内部異常
		8007	アクセス不可の変数が指定された
		8031	内部異常（メモリアロケーションエラー）
20	INVALID_PARAMETER	8009	セグメントタイプ異常
		800F	リクエストデータ中のデータ長情報矛盾
		8017	単体データに対して、1 要素以外を要求
		8018	配列データに対して、0 要素、又は範囲を超えてデータ要求
		8023	内部異常（コマンドフォーマット不正）
		8024	内部異常（コマンド長不正）
		8025	内部異常（パラメータ不正）
		8027	内部異常（パラメータ異常）
		8028	・サブレンジ指定された変数に対して範囲外の値を書き込んだ ・Enum 型変数に対して定義されていない値を書き込んだ

8-6-2 変数書込みサービス

サービスコードに 4D Hex を指定して、リクエストパスで指定した変数に値を書込むことができます。

変数の書き込みリクエストデータフォーマット

Request Path Data

変数名指定

Request Service Data

Data Type		USINT	書き込む変数の型
AddInfo Length		USINT	付加情報（フィールドの長さ：単位はバイト)
(AddInfo)			付加情報（構造体の CRC 値）
Num of Element	(L)	UINT	
	(H)		
実データ *			

Response Service Data

なし

* 書き込みデータ : 書き込むデータはリトルエンディアンで格納します。

Data Type	書き込む変数のデータ型コード。📖「8-7 変数のデータタイプ一覧」を参照。
AddInfoLength	構造体型変数にアクセスする場合のみ、AddInfo 領域のサイズを指定する。 構造体型の場合、02Hex を指定する。それ以外では 00 Hex を指定する。
AddInfo	構造体型変数にアクセスする場合のみ、構造体定義の CRC コードを指定する。 この場合、AddInfo のサイズは 2 バイト。
NumOfElement	配列の要素数を指定する。0000 Hex 指定はできない (エラーとなる)。 配列型以外では、必ず 0001 Hex を指定する。

実データ	リトルエンディアンで指定する。 配列型で、0001 Hex を指定した場合は、その配列を構成する要素のデータ型の変数にアクセスした場合と同様のフォーマットで実データを指定する。

レスポンスコード

CIP ステータス	内容	Add ステータス	原因
00	SUCCESS	-	正常終了
02	RESOURCE_UNAVAILABLE	-	内部処理用バッファが現在利用できない
04	PATH_SEGMENT_ERROR	-	Request Path 指定が間違っている
05	PATH_DESTINATION_UNKNOWN	-	リンク終端まで辿って対象が見つからなかった
0C	OBJECT_STATE_CONFLICT	8010	ダウンロード中、起動中
		8011	タグメモリ異常中
13	NOT_ENOUGH_DATA	-	指定サービスのデータ長が不足している
15	TOO_MUCH_DATA	-	指定サービスのデータ長が多すぎる
1F	VENDOR_SPECIFIC_ERROR	0102,2103	書き込み対象が定数 / 読出し専用である
		0104,1103	セグメント領域を超えるアドレス、サイズが指定されている
		8001	内部異常 (モジュール間 IF 部で情報矛盾検出)
		8007	アクセス不可の変数が指定された
		8029	SimpleDataSegment で一括アクセスできない領域が指定された
		8031	内部異常（メモリアロケーションエラー）
20	INVALID_PARAMETER	8009	セグメントタイプ異常
		800F	リクエストデータ中のデータ長情報矛盾
		8017	単体データに対して、1 要素以外を要求
		8018	配列データに対して、0 要素、又は範囲を超えてデータ要求
		8021	AddInfo エリアに 0, 2 以外の値が指定された
		8022	Request Service Data の「Data Type」が TAG 情報の型と一致しない Request Service Data の「AddInfo Length」が 0 でない
		8023	内部異常（コマンドフォーマット不正）
		8024	内部異常（コマンド長不正）
		8025	内部異常（パラメータ不正）
		8027	内部異常（パラメータ異常）
		8028	・サブレンジ指定された変数に対して範囲外の値を書き込んだ ・Enum 型変数に対して定義されていない値を書き込んだ

8-7　変数のデータタイプ一覧

CIP メッセージ通信で使用する、変数のデータタイプを説明します。

8-7-1　データタイプコード

変数のデータタイプのコードを以下に示します。

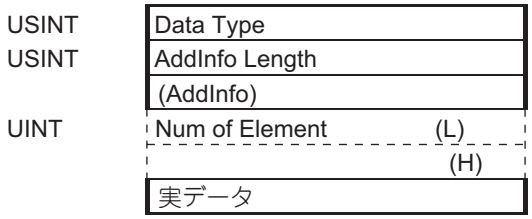
Data Type	Code(Hex)	Group*
BOOL（ビット）	C1	CIP Common
SINT（符号付き 1 バイト BIN）	C2	CIP Common
INT（符号付き 1 チャネル BIN）	C3	CIP Common
DINT（符号付き 2 チャネル BIN）	C4	CIP Common
LINT（符号付き 4 チャネル BIN）	C5	CIP Common
USINT（符号なし 1 バイト BIN）	C6	CIP Common
UINT（符号なし 1 チャネル BIN）	C7	CIP Common
UDINT（符号なし 2 チャネル BIN）	C8	CIP Common
ULINT（符号なし 4 チャネル BIN）	C9	CIP Common
REAL（浮動小数点 2 チャネル）	CA	CIP Common
LREAL（浮動小数点 4 チャネル）	CB	CIP Common
STRING	D0	CIP Common
BYTE（16 進 1 バイト）	D1	CIP Common
WORD（16 進 1 チャネル）	D2	CIP Common
DWORD（16 進 2 チャネル）	D3	CIP Common
TIME（8 バイトデータ ）	DB	CIP Common
LWORD（16 進 4 チャネル）	D4	CIP Common
Abbreviated STRUCT	A0	CIP Common
STRUCT	A2	CIP Common
ARRAY	A3	CIP Common
UINT BCD（符号なし 1 チャネル BCD）	04	Vendor Specific
UDINT BCD（符号なし 2 チャネル BCD）	05	Vendor Specific
ULINT BCD（符号なし 4 チャネル BCD）	06	Vendor Specific
ENUM	07	Vendor Specific
DATE_NSEC	08	Vendor Specific
TIME_NSEC	09	Vendor Specific
DATE_AND_TIME_NSEC	0A	Vendor Specific
TIME_OF_DAY_NSEC	0B	Vendor Specific
Union	0C	Vendor Specific

* CIP Common とは、CIP 共通の仕様で定義されたコードです。Vendor Specific とはオムロン独自に割り当てたコードです。

8-7-2　共通フォーマット

データの回線上表現の基本フォーマットは、以下になります。

（データフォーマット）



8-7-3　基本型 (Elementary Data Types)

バイト固定長データ

対象：BYTE,USINT,SINT

（データフォーマット）

USINT	Data Type		
USINT	00h		
UINT	Num of Elem	(L)	01 Hex
		(H)	00 Hex
USINT	データ		

2 バイト固定長データ

対象：INT, UINT, UINT BCD, WORD

（データフォーマット）

USINT	Data Type		
USINT	00h		
UINT	Num of Elem	(L)	01 Hex
		(H)	00 Hex
	データ	(L)	
		(H)	

4 バイト固定長データ

対象：DINT, UDINT, UDINT BCD, REAL, DWORD

（データフォーマット）

USINT	Data Type		
USINT	00h		
UINT	Num of Elem	(L)	01 Hex
		(H)	00 Hex
	データ	(LL)	
		(LH)	
		(HL)	
		(HH)	

8 バイト固定長データ

対象：LINT, ULINT, ULINT BCD, LREAL, LWORD

（データフォーマット）

USINT	Data Type		
USINT	00 Hex		
UINT	Num of Elem	(L)	01 Hex
		(H)	00 Hex
	データ	（最下位）	
		:	
		:	
		:	
		:	
		:	
		:	
		（最上位）	

BOOL 型

（データフォーマット）

USINT	Data Type		C1 Hex
USINT	00 Hex		
UINT	Num of Elem	(L)	01 Hex
		(H)	00 Hex
USINT	Status		01 Hex : TRUE / 00 Hex : FALSE
USINT	強制 Set/Reset 情報 *		01 Hex : 強制あり / 00 Hex : 強制なし

*　書き込み時は 0 を指定して下さい。

8-7-4　派生型 (Derived Data Types)

派生型として、配列および構造体の扱いを定義します。

メンバ 1 個へのアクセス

配列・構造体とも、構成要素の 1 メンバにアクセスする際のデータフォーマットは、対応する基本型と同じ仕様となります。

例：　UINT Var[10] で定義された変数に、Var[5] を指定してアクセスする場合は UINT 型アクセスと同じ仕様。

複数要素への一括アクセス

● 配列型

- 全体一括アクセス

  配列型変数を、要素指定なしにアクセスする場合、それは配列全体へのアクセス要求を意味します。

  この際のデータフォーマットは以下です。

（データフォーマット）

USINT	Data Type		構成要素の Data Type (A1 Hex は使わない)
USINT	00 Hex		
UINT	Num of Elem	(L)	構成要素数を格納する
		(H)	
	データ		構成要素の実データ部（単体指定と同じ書式）が並ぶ (*)
	:		
	データ		

* STRING 型のみ、出力フォーマットが以下の点で単体アクセス時と異ります。

- 文字列長のフィールドが存在せず、文字列本体 (NULL 含む) のみ存在する。
- 転送データ長は、文字列の長さでなくその文字列型変数に確保されたメモリサイズとなる。
- 上位バイト・下位バイトが反転した状態となる

（例）要素数 2 の STRING 型配列 s(要素辺りのデータ量 =4 バイトに設定) の出力は、
s[0]=”ab”, s[1]=”d” の場合、
単体 [0]：「D0 00 03 00 61 62 63」(Hex)
配列一括：「D0 00 62 61 ?? 00 00 64 ?? ??」(?? は不定値) (Hex)

• 多次元配列の扱い

多次元配列における要素の列挙順は、次元の深い要素から順に列挙します。
例えば、UINT Var[2][2] で定義された変数を、"Var" 指定で読み出す場合のフォーマットは以下です。

(データフォーマット)

USINT	C7 Hex	UINT 型の Data Type Code
USINT	00 Hex	
UINT	Var[0][0] の値 (L)	
	(H)	
UINT	Var[0][1] の値 (L)	
	(H)	
UINT	Var[1][0] の値 (L)	
	(H)	
UINT	Var[1][1] の値 (L)	
	(H)	

BOOL 配列 (例えば BOOL b[2][3] とする) の場合は以下のようになります。

(データフォーマット)

USINT	C1 Hex (BOOL 型の Data Type Code)							
USINT	00 Hex							
(WORD)	rsv	rsv	b[1][2]	b[1][1]	b[1][0]	b[0][2]	b[0][1]	b[0][0]
	rsv	rsv	rsv	rsv	rsv	rsv	rsv	rsv

• Num of element 指定時の例外

BOOL 型配列の Num of Element 指定（📖「8-4-5 リクエストパスでの変数名の指定方法」参照）による一部アクセス時は、報告フォーマットが以下のようになります。(BOOL 型変数 1 要素ごとの Status(TRUE/FALSE) を列挙する形式となる)

(データフォーマット)

USINT	Data Type	C1 Hex
USINT	00 Hex	
UINT	Num of Elem (L)	構成要素数を格納する
	(H)	
USINT	Status	01 Hex : TRUE / 00 Hex : FALSE
:	:	
USINT	Status	

● 構造体型

- 全体一括アクセス
  構造体型変数そのものを指定する場合、それは構造体全要素の一括アクセス要求として取り扱われます。

（データフォーマット）

USINT	Data Type		A0 Hex (Abbreviated STRUCT)
USINT	02 Hex		
UINT	CRC	(L)	構造体定義を表す CRC 値
		(H)	
UINT	Num of Elem	(L)	01 Hex
		(H)	00 Hex
	: データ :		

9

ソケットサービス機能

9-1 ソケット通信の基礎知識 9-2
9-1-1 ソケットとは 9-2
9-1-2 ソケットサービスで使用できるポート No. 9-2

9-2 プロトコルの基礎知識 9-3
9-2-1 TCP と UDP の相違点 9-3
9-2-2 送信データのフラグメント（分割） 9-5
9-2-3 データの受信処理 9-7
9-2-4 一斉同報送信 9-10

9-3 内蔵 EtherNet/IP ポートのソケットサービス概要 9-11
9-3-1 概要 9-11
9-3-2 使用手順 9-11

9-4 ソケットサービス機能に必要な設定 9-12

9-5 ソケットサービス用命令 9-13

9-6 ソケットサービスの利用の詳細 9-14
9-6-1 ソケットサービスの使用方法 9-14
9-6-2 ソケットサービスの使用手順 9-14
9-6-3 通信時の出力変数のタイミングチャート 9-17
9-6-4 UDP のサンプルプログラム 9-18
9-6-5 TCP のサンプルプログラム 9-23

9-7 ソケットサービス使用上の留意事項 9-29
9-7-1 UDP/TCP ソケットサービス共通の留意事項 9-29
9-7-2 UDP ソケットサービスのみの留意事項 9-29
9-7-3 TCP ソケットサービスのみの留意事項 9-30

9-1　ソケット通信の基礎知識

9-1-1　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。
内蔵 EtherNet/IP ポートの場合は、ユーザプログラムから、命令を実行することにより使用できます。
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。
内蔵 EtherNet/IP ポートは、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。

9-1-2　ソケットサービスで使用できるポート No.

TCP/IP で使用されるポート No.0 ～ 1023 は、well-known ポートとして予約されています。また、No.1024 ～ No.49151 は、Registered ポートとして使用するプロトコルが登録されています。
そのため、ソケットサービスでは登録されたプロトコル以外の用途で使用する場合は、No.49152 ～ No.65535 を使用することをお勧めします。
また、内蔵 EtherNet/IP ポートでは、ポート No.0 は指定できません。
さらに、内蔵 EtherNet/IP ポートでは、以下のポートが使用されていますので、重複して設定しないようにしてください。

アプリケーション	UDP	TCP
FTP サーバ	−	20、21（*）
DNS	53	−
BOOTP クライアント	68	−
HTTP サーバ	−	80
SNMP エージェント	161（*）	−
SNMP トラップ	162（*）	−
UDP	9600	−
TCP	−	9600（*）
システムで使用	2222、2223、2224	9610、9900
CIP メッセージ	44818	44818

* デフォルト設定されているポート No. です。設定によって変更できます。

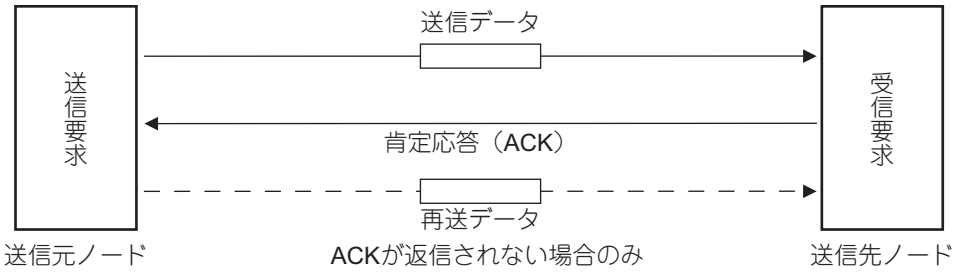
9-2　プロトコルの基礎知識

9-2-1　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

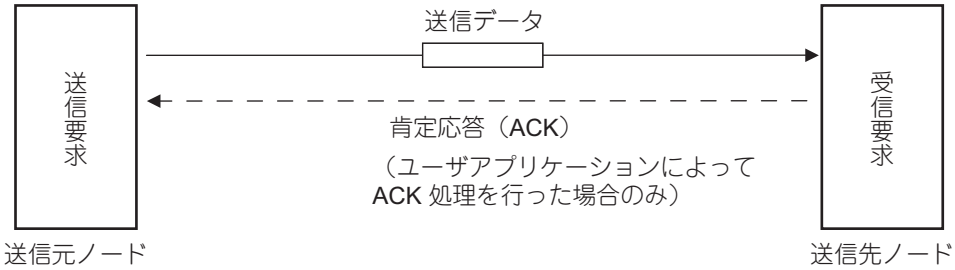
データの送信ごとに次のような動作を行うことによって、データが送信先ノードに確実に届くように処理されます。
送信先ノードでは、データを正常に受信すると肯定応答（ACK）を返信します。
送信元ノードでは、ACK が返信されてきたときは次のデータを送信し、一定時間以内に ACK が返信されないときは同じデータを再送します。



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するデータが格納されている変数を指定します。

UDP による通信

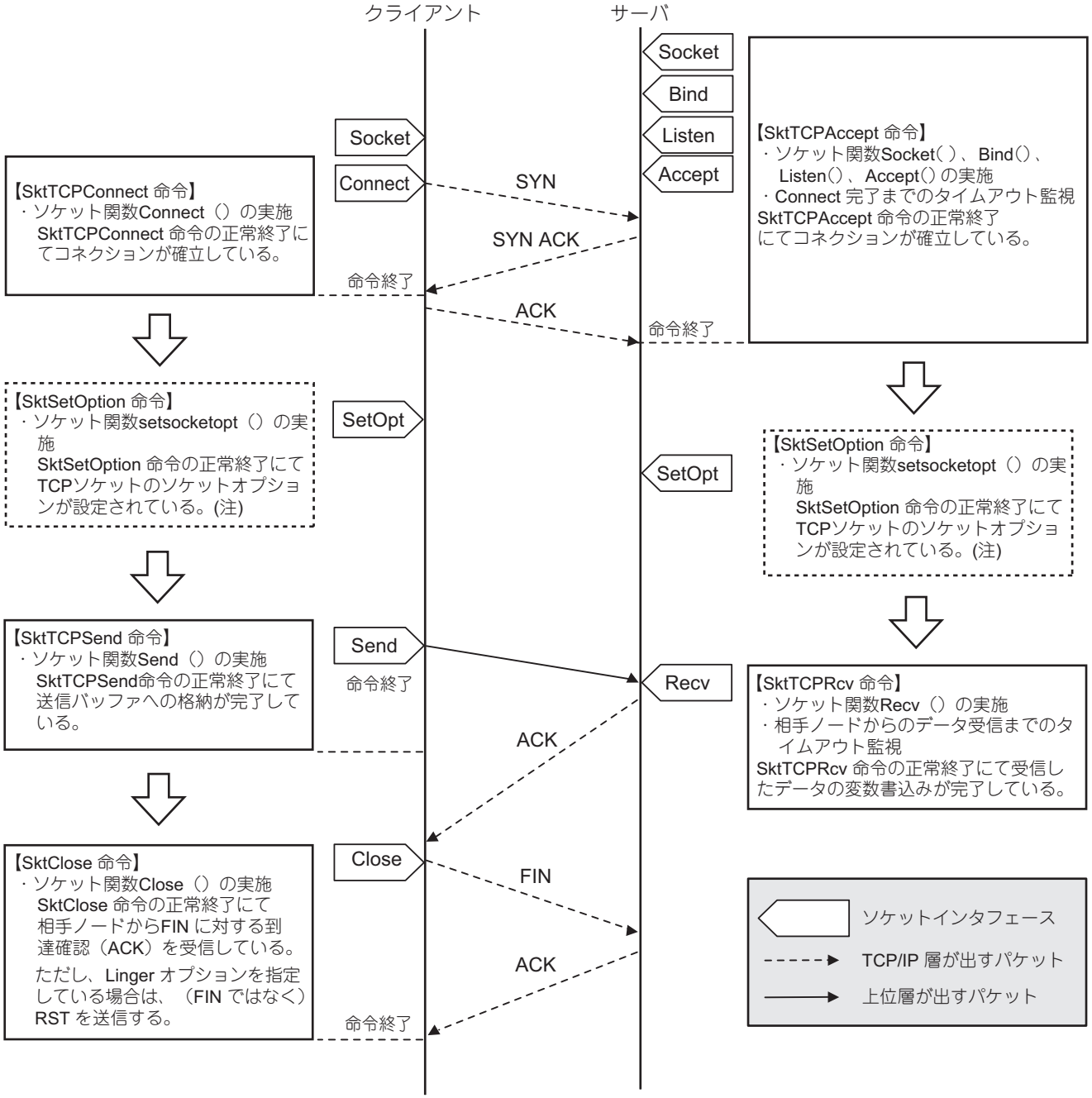
データは送信先ノードへ送信されるだけで、TCP のような受信確認・再送処理は行われません。
通信の信頼性を高めるためには、ユーザアプリケーションでデータの再送処理などを行う必要があります。



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.、送信するデータが格納されている変数を指定します。（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。

TCP による通信手段

内蔵 EtherNet/IP ポートを使った TCP 通信では、ソケット通信用の各命令を順次実行して、通信処理を行います。



（注）ソケットオプションの設定は、必要に応じて行います。ソケットオプションの内容は、🕮『NY シリーズ コマンドリファレンスマニュアル 基本編（SBCA-437)』を参照してください。

9-2-2　送信データのフラグメント（分割）

内蔵 EtherNet/IP ポートの受信バッファは、1 つのソケットハンドルで最大 9000 バイトです。9000 バイト以上のデータを受信した場合、そのデータは破棄されます。

1 回の受信要求で受信できるデータサイズは 2000 バイトで、その際以下のようにフラグメント（分割）して送信されます。

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。

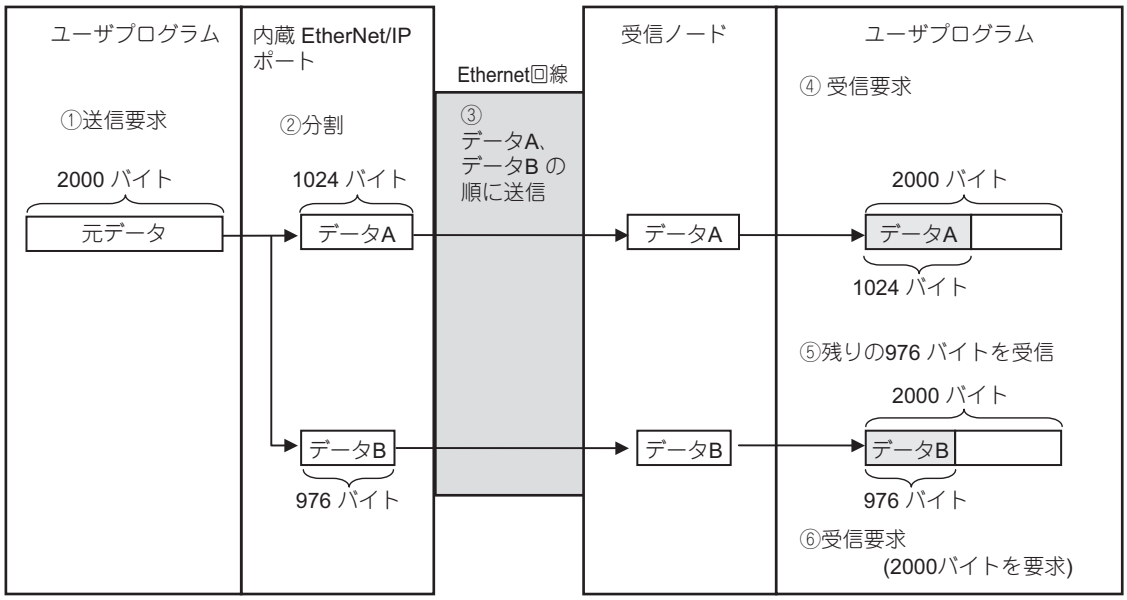
(1) 送信側のユーザプログラムで 2000 バイトのデータ長の変数を指定して送信要求を行います。

(2) 内蔵 EtherNet/IP ポートで、送信データが 1024 バイトのデータ A と、976 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) データ A の受信後、残りのデータ B が受信されます。

送信側（コントローラ）　　受信側（ホストコンピュータ）



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。

受信データのバイト数を確認し、すべてのデータが受信されたのを確認してから、次の受信要求を行う必要があります。（ソケット受信要求命令の出力変数（RecvDataSize）で受信したデータを確認できます。）

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

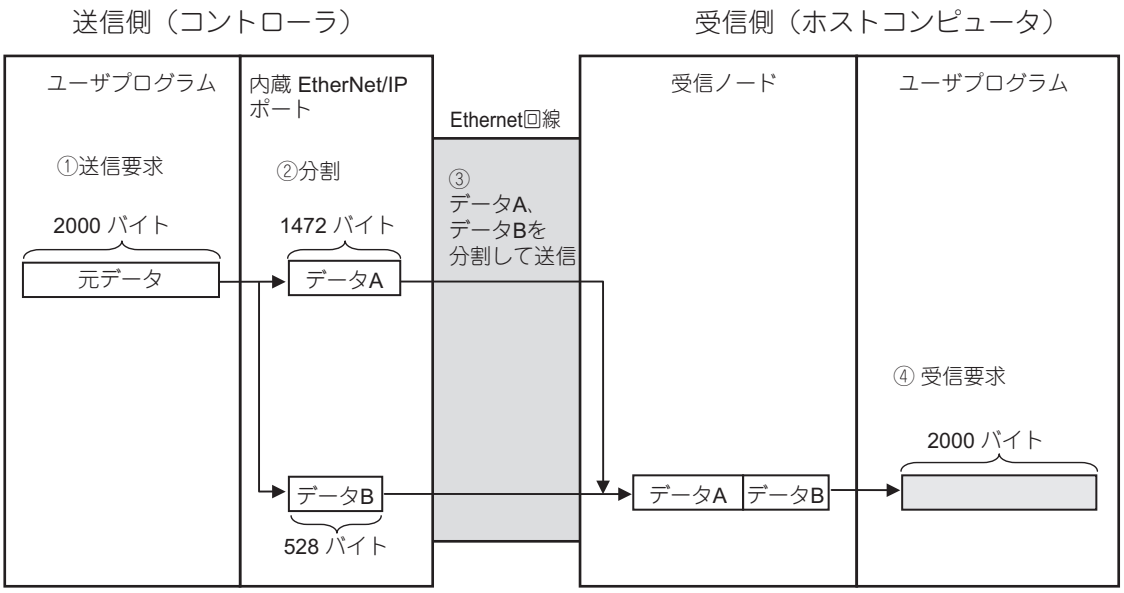
UDP の通信では、データが分割されて送信されるときに次図のような状態が発生します。

(1) 送信側のユーザプログラムで、2000 バイトのデータ長の変数を指定して送信要求を行います。

(2) 内蔵 EtherNet/IP ポートで、送信データが 1472 バイトのデータ A と、528 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) 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。

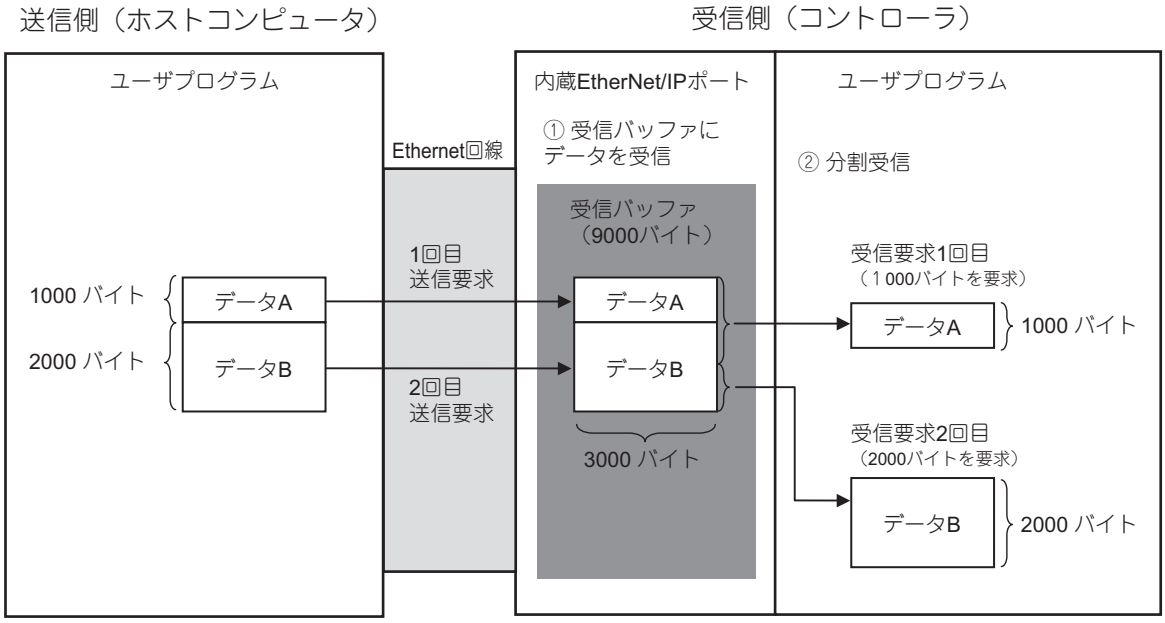
9-2-3　データの受信処理

TCP、UDP ごとのデータの受信処理について説明します。

● TCP の受信処理

TCP の通信では、受信バッファ（最大 9000 バイト）に格納された受信データを、分割受信できます。よって、1 回の受信要求で受信できるサイズ（2000 バイト）を超えるデータを受信した場合は、複数の受信要求によって、すべて受信することができます。
また、受信要求で指定した変数のサイズよりも、受信バッファのデータが少なかった場合は、受信データはすべて受信されます。

例）　3000 バイトの受信データを、分割受信する場合
① 送信側ノードから、2 回に分けて送信されたデータが、受信バッファに格納されます。
② 複数の受信要求を行って、送信データすべてを受信します。

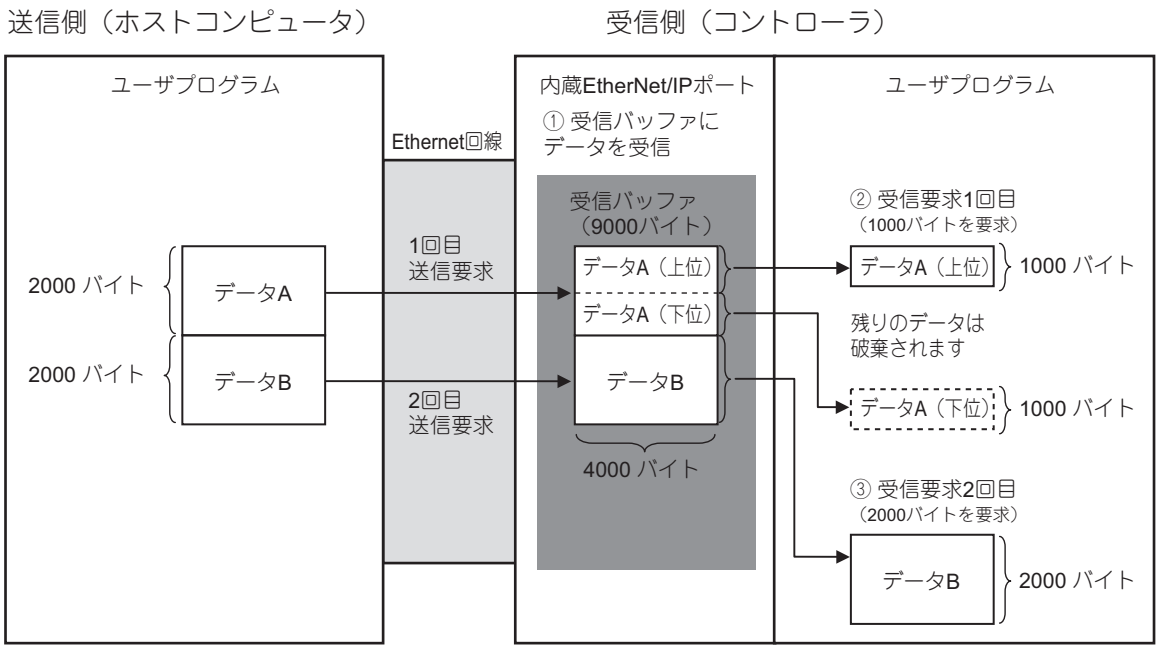


● UDP の受信処理

UDP の通信では、受信バッファ（最大 9000 バイト）に格納された受信データを、分割受信できません。
よって、1 回の送信要求で送信されたデータは、1 回の受信要求でデータを受信します。
受信側では以下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。

- 送信要求で送信されたデータのサイズに対して、受信要求で指定した変数のサイズが少ない場合
  受信要求で指定した変数のサイズを超える分の受信データは、破棄されます。
  また、受信要求で指定した変数のサイズよりも、受信バッファのデータが少なかった場合は、受信データはすべて受信されます。

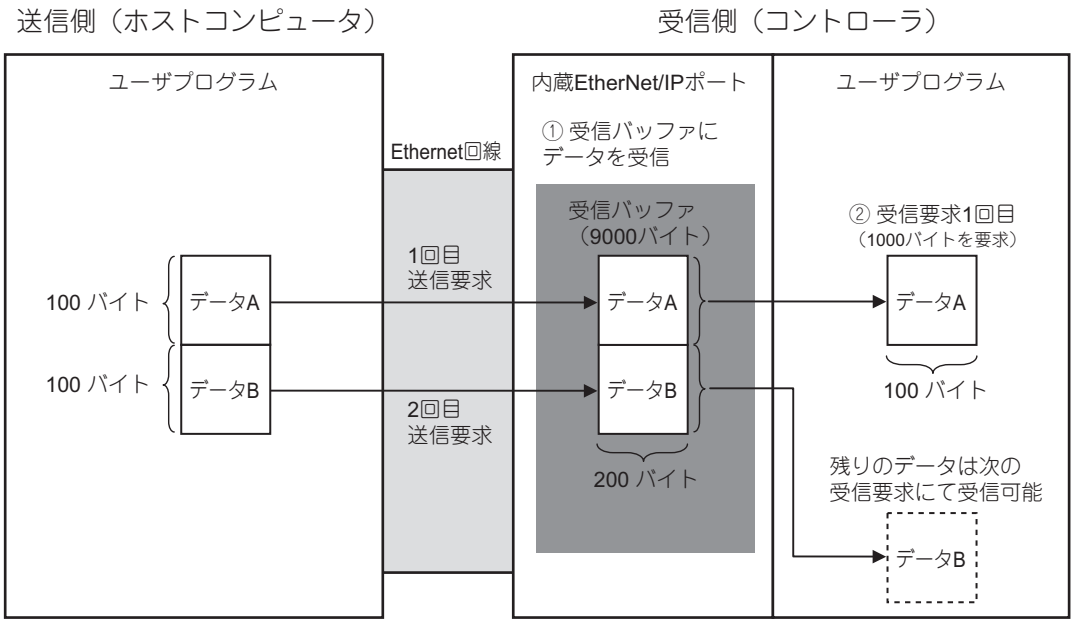
例 1）2000 バイトの受信データに対し、1000 バイトの受信要求をした場合
① 送信側ノードから、2 回に分けて送信されたデータが、受信バッファに格納されます。
② 1 回目の送信データに対して、1000 バイトの受信要求を行った場合、残りの 1000 バイトは破棄されます。
③ 次の受信要求を 2000 バイトで行った場合、2 回目の送信データは、すべて受信されます。



- 複数回の送信要求で送信されたデータを、1 回の受信要求で受信する場合
  データのサイズに関係なく、複数の送信要求で送信されたデータを、1 回の受信要求でまとめて受信することはできません。

例 2）2 回の送信要求で送信された 200 バイトの受信データに対し、1000 バイトの受信要求をした場合

① 送信側ノードから、2 回に分けて送信されたデータが、受信バッファに格納されます。
② 2000 バイトのデータサイズで受信要求しても、1 回目の送信要求分 100 バイトのみ受信されます。



9-2-4　一斉同報送信

UDP ソケットでは、送信先の IP アドレスにブロードキャストアドレスを指定することで、EtherNet/IP ポート自身が加入しているホストと同一ネットワーク内の全ノードに対して、一斉同報送信することができます。
ただし、ネットワーク上にルータがあった場合、ルータを越えての送信はできません。
送信できるデータは、最大 1472 バイトです。それを超えるデータは、一斉同報送信できません。

以下の 2 種類のブロードキャストアドレスを指定できます。

- ローカルブロードキャスト
  宛先の IP アドレスを指定しない場合、以下の IP アドレスが指定されます。
  ネットワーク部：　自身の IP アドレスのネットワーク部
  ホスト部：　　全ビットを 1 にしたアドレス

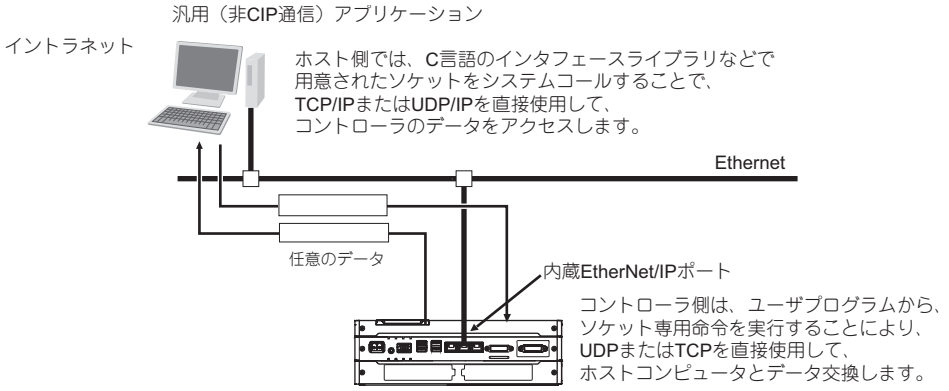
- グローバルブロードキャスト（制限ブロードキャスト）
  自ノードの IP アドレスや自ノードの所属するサブネットが不明な場合に指定します。
  以下のように、32 ビットすべてが 1 のアドレスとなります。

  255.255.255.255

9-3　内蔵 EtherNet/IP ポートのソケットサービス概要

9-3-1　概要

内蔵 EtherNet/IP ポートのソケットサービスは、CIP メッセージ通信に対応していない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、ユーザプログラムから、ソケットサービスを要求します。



ソケットサービス用命令によるソケットサービスの概要

ソケットサービス命令により、ソケットサービスを利用することができます。
UDP ソケット分と TCP ソケット分を合わせて、以下の個数までの利用が可能です。

UDP/TCP	ソケット数
	NY シリーズコントローラ
UDP ソケットサービス	合計 30 ソケットまで可能
TCP ソケットサービス	

9-3-2　使用手順

1.　ソケットサービスに必要な設定を行います。
「9-4 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.　ユーザプログラムからソケットサービス用の命令を実行します。
「9-5 ソケットサービス用命令」を参照してください。

9-4　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 ユニット設定タブ	設定項目	設定が必要な条件
設定	自 IP アドレス	必須
	サブネットマスク	必須
	TCP/IP KeepAlive	任意（デフォルト 5 分以外のときに必要）
	Linger オプション	任意



参考

設定は［TCP/IP］ダイアログで行います。［TCP/IP］ダイアログについての詳細は、🕮「5-1 ［TCP/IP 設定］ダイアログ」を参照してください。

9-5　ソケットサービス用命令

ソケットサービス用として、以下のソケットサービス用命令があります。
ソケットサービス用命令については、🕮『NY シリーズ コマンドリファレンスマニュアル 基本編 (SBCA-437)』を参照してください。

UDP/TCP	命令名	ソケットサービス種類
UDP ソケット	SktUDPCreate	UDP ソケットのクリエイト命令
	SktUDPRcv	UDP ソケットの受信命令
	SktUDPSend	UDP ソケットの送信命令
TCP ソケット	SktTCPAccept	TCP ソケットのアクセプト命令
	SktTCPConnect	TCP ソケットのコネクト命令
	SktTCPRcv	TCP ソケットの受信命令
	SktTCPSend	TCP ソケットの送信命令
	SktGetTCPStatus	TCP ソケットステータス読出命令
UDP/TCP ソケット共通	SktClose	UDP/TCP ソケットクローズ命令
	SktClearBuf	UDP/TCP バッファクリア命令
	SktSetOption	TCP ソケットオプション設定命令

使用上の注意

ソケットサービス用命令は、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。

9-6　ソケットサービスの利用の詳細

9-6-1　ソケットサービスの使用方法

内蔵 EtherNet/IP ポートでは、TCP、UDP 合計で以下の個数のソケットを持ちます。

UDP/TCP	ソケット数
	NY シリーズコントローラ
UDP ソケットサービス	合計 30 ソケットまで可能
TCP ソケットサービス	

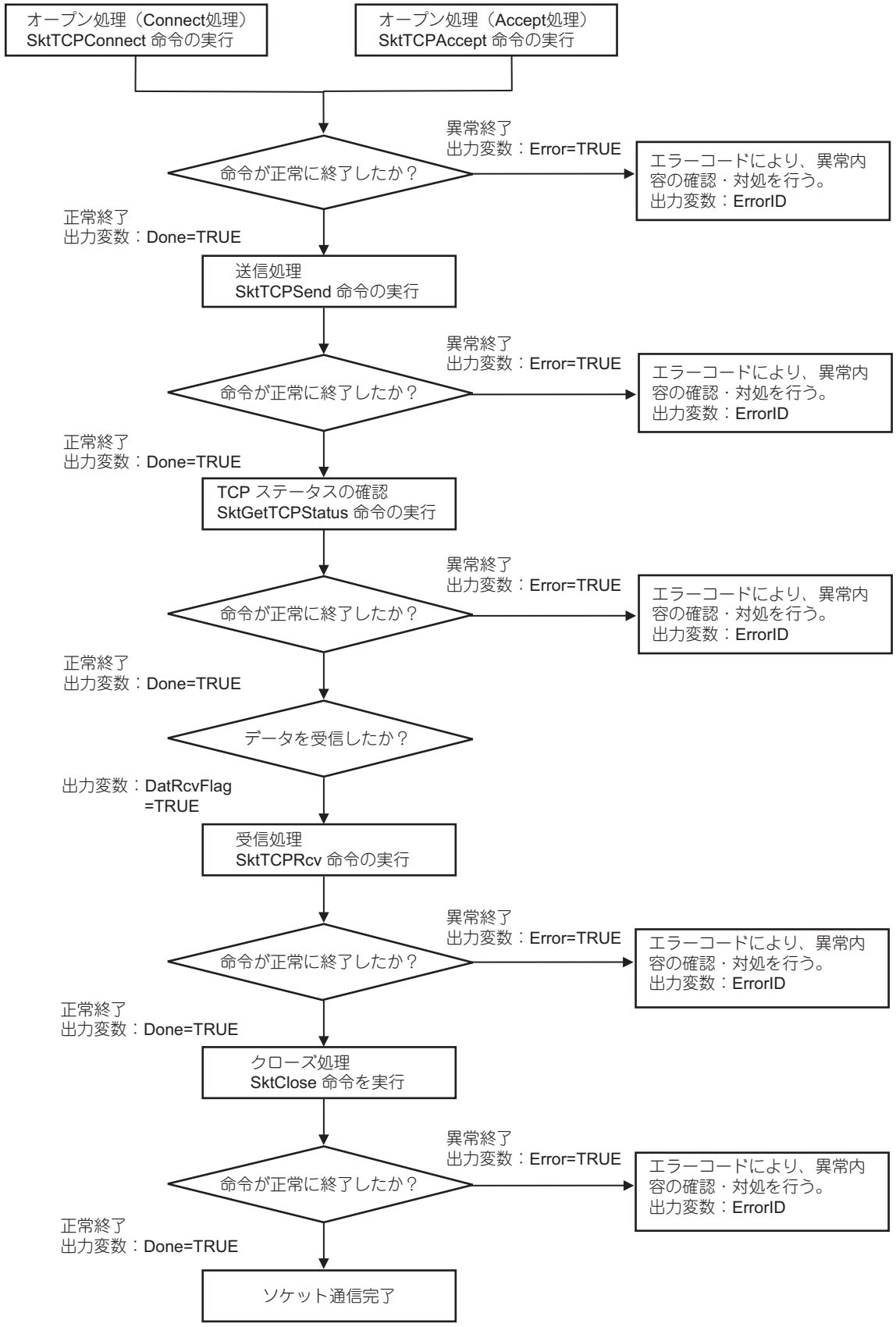
これらのソケットを使用して通信するため、ソケット専用の ST 命令を実行して、以下の処理を実行していきます。

オープン処理：	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：	ソケットからデータを送信します。
受信処理：	ソケットからデータを受信します。
クリア処理：	受信バッファ内の相手ノードから受信したデータをクリアします。

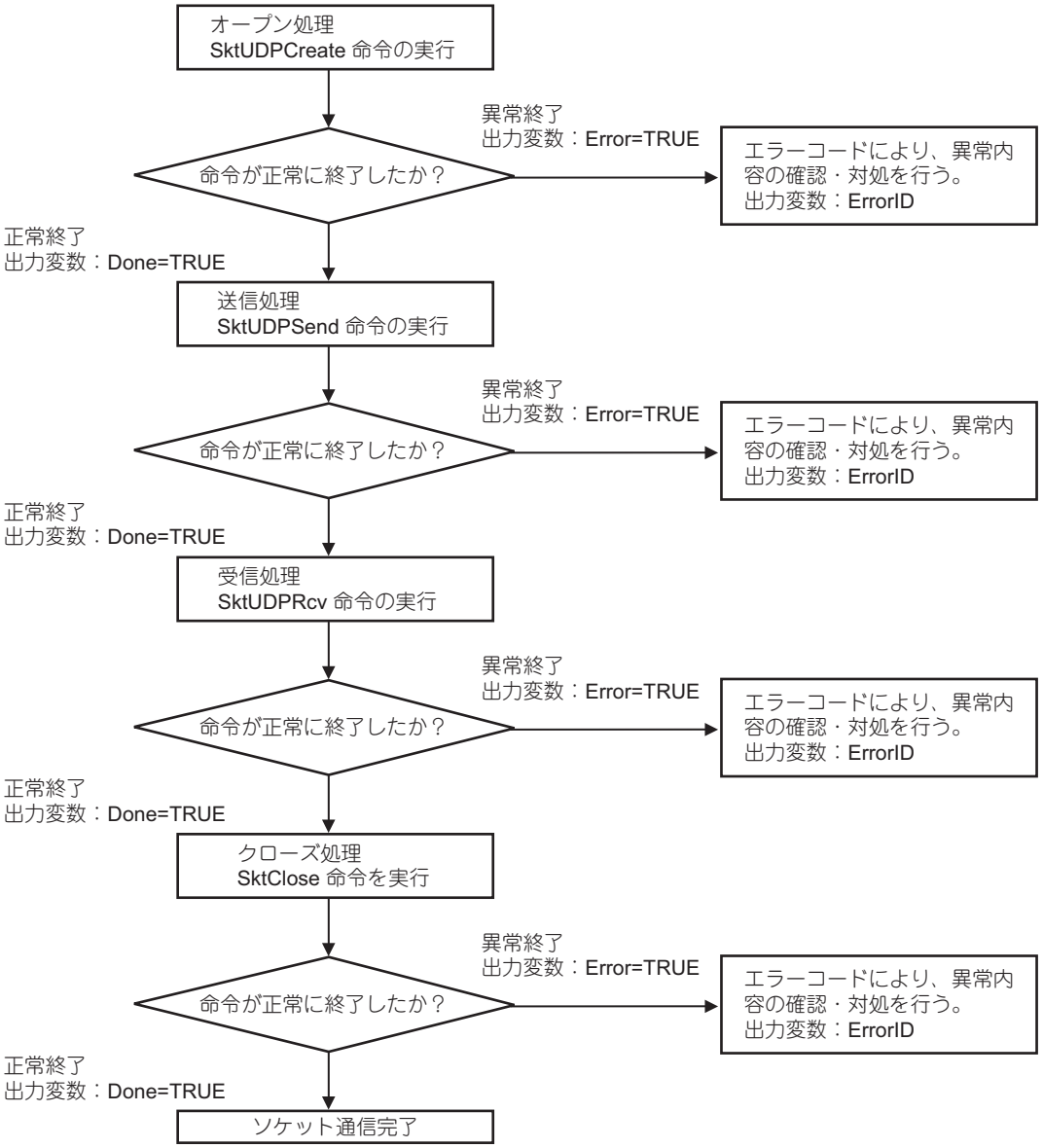
9-6-2　ソケットサービスの使用手順

ソケットサービスは、各ソケットの専用命令を順次実行して、以下のような流れで行います。
その際、各命令が正常に終了したかどうかを、各命令の出力変数の値で確認してください。

TCP の場合


オープン処理（Connect処理）
SktTCPConnect 命令の実行
オープン処理（Accept処理）
SktTCPAccept 命令の実行
命令が正常に終了したか？
異常終了
出力変数：Error=TRUE
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
送信処理
SktTCPSend 命令の実行
命令が正常に終了したか？
異常終了
出力変数：Error=TRUE
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
TCP ステータスの確認
SktGetTCPStatus 命令の実行
命令が正常に終了したか？
異常終了
出力変数：Error=TRUE
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
データを受信したか？
出力変数：DatRcvFlag
=TRUE
受信処理
SktTCPRcv 命令の実行
命令が正常に終了したか？
異常終了
出力変数：Error=TRUE
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
クローズ処理
SktClose 命令を実行
命令が正常に終了したか？
異常終了
出力変数：Error=TRUE
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
ソケット通信完了

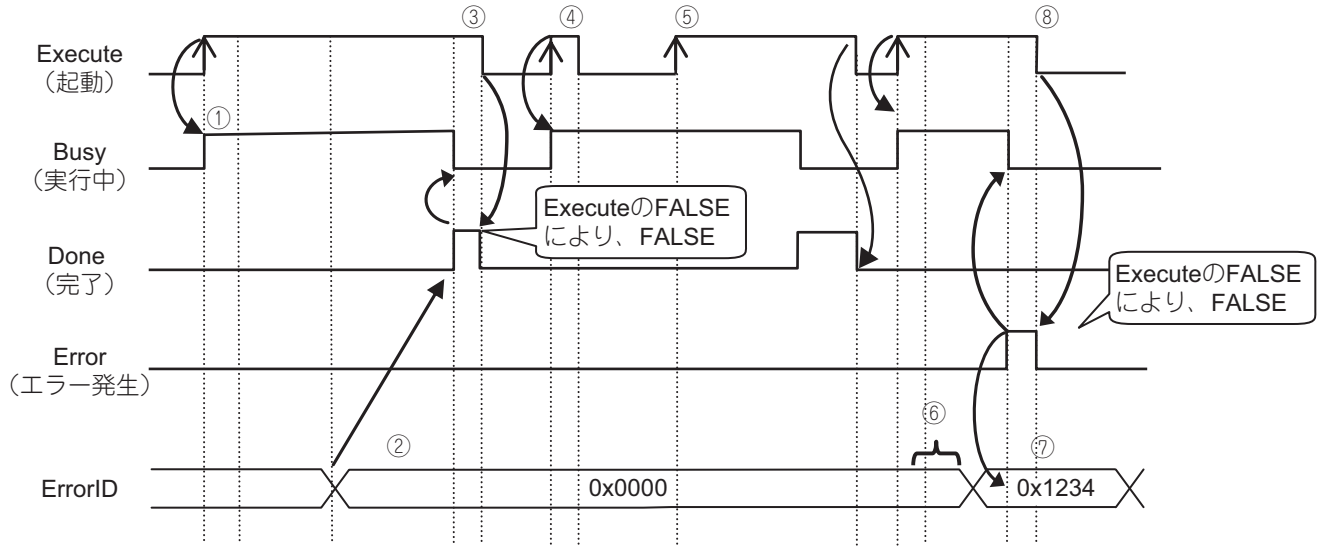
UDP の場合


オープン処理
SktUDPCreate 命令の実行
異常終了
出力変数：Error=TRUE
命令が正常に終了したか？
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
送信処理
SktUDPSend 命令の実行
異常終了
出力変数：Error=TRUE
命令が正常に終了したか？
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
受信処理
SktUDPRcv 命令の実行
異常終了
出力変数：Error=TRUE
命令が正常に終了したか？
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
クローズ処理
SktClose 命令を実行
異常終了
出力変数：Error=TRUE
命令が正常に終了したか？
エラーコードにより、異常内容の確認・対処を行う。
出力変数：ErrorID
正常終了
出力変数：Done=TRUE
ソケット通信完了

9-6-3　通信時の出力変数のタイミングチャート

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。

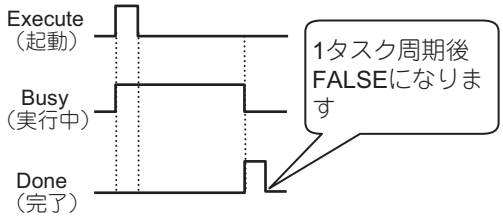


①Execute の立上りで、命令が実行され、Busy が TRUE になります。
②命令の実行結果が、出力変数に反映されると、Done が TRUE、Busy が FALSE になります。
③Execute を FALSE にすると、Done も FALSE に戻ります。
④再び Execute が立ち上がると、Busy が TRUE になります。
⑤命令実行中 (Busy = TRUE) に Execute が立ち上がった場合、無視されます。
⑥エラーが発生すると、内部ではリトライを数回実行します。
　リトライ中は、ErrorID にエラーが更新されません。
⑦通信エラーが発生した場合、Error 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。

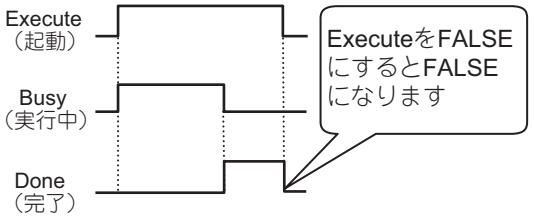
使用上の注意

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)

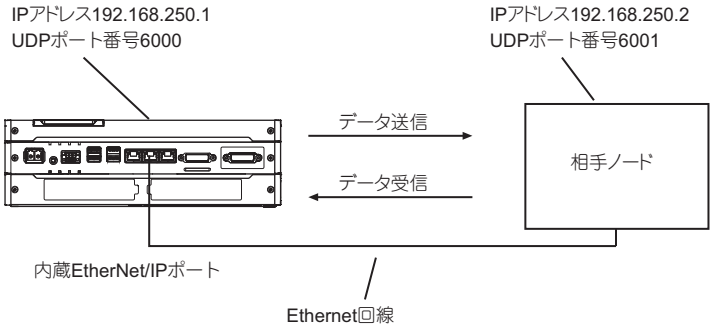


(例2)



9-6-4　UDP のサンプルプログラム

NY シリーズコントローラと、相手ノードとの間で、ソケットサービス機能 (UDP) を使用して、データの送受信を行います。



処理の手順は、以下のとおりです。

1. **SktUDPCreate 命令で、UDP ソケットのクリエイト要求を行います。**
2. **SktUDPSend 命令で、送信要求を行います。送信するデータは、SendSocketDat[] の中身とします。**
3. **SktUDPRcv 命令で、受信要求を行います。受信したデータは、RcvSocketDat[] に格納します。**
4. **SktClose 命令で、ソケットをクローズします。**

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
	DoSendAndRcv	BOOL	False	処理中
	Stage	INT	0	状態遷移
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	受信データ
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0, IpAdr:=''), DstAdr:=(PortNo:=0, IpAdr:=''))	ソケット
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	送信データ
	SktUDPCreate_instance	SktUDPCreate		
	SktUDPSend_instance	SktUDPSend		
	SktUDPRcv_instance	SktUDPRcv		
	SktClose_instance	SktClose		

外部変数	名称	データ型	コンスタント	コメント
	_EIP_EtnOnlineSta*1	BOOL	☑	オンライン

*1. 使用する通信ポートによって、_EIP1_EtnOnlineSta（通信ポート 1 オンライン）、または _EIPIn1_EtnOnlineSta（内部通信ポート 1 オンライン）のいずれかに置き換えてください。

```
// Trigger の立ち上がりでシーケンス開始
IF((Trigger=TRUE) AND(DoSendAndRcv=FALSE) AND (_EIP_EtnOnlineSta=TRUE))THEN
    DoSendAndRcv :=TRUE;
    Stage            :=INT#1;
    SktUDPCreate_instance(Execute:=FALSE);    // インスタンス初期化
    SktUDPSend_instance(                      // インスタンス初期化
        Execute      :=FALSE,
        SendDat      :=SendSocketDat[0]);     // ダミー
    SktUDPRcv_instance(                       // インスタンス初期化
        Execute      :=FALSE,
        RcvDat       :=RcvSocketDat[0]);      // ダミー
    SktClose_instance(Execute:=FALSE);        // インスタンス初期化
END_IF;

IF (DoSendAndRcv=TRUE) THEN
    CASE Stage OF
    1 :                 // クリエイト要求
        SktUDPCreate_instance(
            Execute   :=TRUE,
            SrcUdpPort:=UINT#6000,            // 自 UDP ポート番号
            Socket    =>WkSocket);            // ソケット

        IF (SktUDPCreate_instance.Done=TRUE) THEN
            Stage     :=INT#2;                // 正常終了
        ELSIF (SktUDPCreate_instance.Error=TRUE) THEN
            Stage     :=INT#10;               // 異常終了
        END_IF;

    2 :                 // 送信要求
        WkSocket.DstAdr.PortNo :=UINT#6001;
        WkSocket.DstAdr.IpAdr:='192.168.250.2';
        SktUDPSend_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket,             // ソケット
            SendDat   :=SendSocketDat[0],     // 送信データ
            Size      :=UINT#2000);           // 送信データサイズ

        IF (SktUDPSend_instance.Done=TRUE) THEN
            Stage     :=INT#3;                // 正常終了
        ELSIF (SktUDPSend_instance.Error=TRUE) THEN
            Stage     :=INT#20;               // 異常終了
        END_IF;
```

```
    3 :                 // 受信要求
        SktUDPRcv_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket,                  // ソケット
            TimeOut   :=UINT#0,                    // タイムアウト時間
            Size      :=UINT#2000,                 // 受信データサイズ
            RcvDat    :=RcvSocketDat[0]);          // 受信データ

        IF (SktUDPRcv_instance.Done=TRUE) THEN
            Stage     :=INT#4;                     // 正常終了
        ELSIF (SktUDPRcv_instance.Error=TRUE) THEN
            Stage     :=INT#30;                    // 異常終了
        END_IF;

    4 :                 // クローズ要求
        SktClose_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket);                 // ソケット

        IF (SktClose_instance.Done=TRUE) THEN
            Stage     :=INT#0;                     // 正常終了
        ELSIF (SktClose_instance.Error=TRUE) THEN
            Stage     :=INT#40;                    // 異常終了
        END_IF;

    0 :                 // 正常終了
        DoSendAndRcv:=FALSE;
        Trigger       :=FALSE;

    ELSE                // 異常による中断
        DoSendAndRcv:=FALSE;
        Trigger       :=FALSE;
    END_CASE;

END_IF;
```

● 相手ノードのユーザプログラム

この例では、相手ノードにもユーザプログラムが必要です。上記の処理手順とは、送信と受信の順序が逆になります。

1 **SktUDPCreate 命令で、UDP ソケットのクリエイト要求を行います**

2 **SktUDPRcv 命令で、受信要求を行います。受信したデータは、RcvSocketDat[] に格納します。**

3 **SktUDPSend 命令で、送信要求を行います。送信するデータは、SendSocketDat[] の中身とします。**

4 **SktClose 命令で、ソケットをクローズします。**

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
	DoSendAndRcv	BOOL	False	処理中
	Stage	INT	0	状態遷移
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	受信データ
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0, IpAdr:=''), DstAdr:=(PortNo:=0, IpAdr:=''))	ソケット
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	送信データ
	SktUDPCreate_instance	SktUDPCreate		
	SktUDPSend_instance	SktUDPSend		
	SktUDPRcv_instance	SktUDPRcv		
	SktClose_instance	SktClose		

外部変数	名称	データ型	コンスタンス	コメント
	_EIP_EtnOnlineSta*1	BOOL	☑	オンライン

*1. 使用する通信ポートによって、_EIP1_EtnOnlineSta（通信ポート 1 オンライン）、または _EIPIn1_EtnOnlineSta（内部通信ポート 1 オンライン）のいずれかに置き換えてください。

```
// Trigger の立ち上がりでシーケンス開始
IF((Trigger=TRUE) AND (DoSendAndRcv=FALSE) AND (_EIP_EtnOnlineSta=TRUE))THEN
    DoSendAndRcv :=TRUE;
    Stage        :=INT#1;
    SktUDPCreate_instance(Execute:=FALSE);    // インスタンス初期化
    SktUDPSend_instance(                      // インスタンス初期化
        Execute      :=FALSE,
        SendDat:=SendSocketDat[0]);           // ダミー
    SktUDPRcv_instance(                       // インスタンス初期化
        Execute      :=FALSE,
        RcvDat       :=RcvSocketDat[0]);      // ダミー
    SktClose_instance(Execute:=FALSE);        // インスタンス初期化
END_IF;

IF (DoSendAndRcv=TRUE) THEN
    CASE Stage OF
    1 :                       // クリエイト要求
        SktUDPCreate_instance(
            Execute    :=TRUE,
            SrcUdpPort:=UINT#6001,            // 自 UDP ポート番号
            Socket     =>WkSocket);           // ソケット

        IF (SktUDPCreate_instance.Done=TRUE) THEN
            Stage      :=INT#2;               // 正常終了
        ELSIF (SktUDPCreate_instance.Error=TRUE) THEN
            Stage      :=INT#10;              // 異常終了
        END_IF;
```

```
    2 :                         // 受信要求
        SktUDPRcv_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket,                   // ソケット
            TimeOut   :=UINT#0,                     // タイムアウト時間
            Size      :=UINT#2000,                  // 受信データサイズ
            RcvDat    :=RcvSocketDat[0]);           // 受信データ

        IF (SktUDPRcv_instance.Done=TRUE) THEN
            Stage     :=INT#3;                      // 正常終了
        ELSIF (SktUDPRcv_instance.Error=TRUE) THEN
            Stage     :=INT#20;                     // 異常終了
        END_IF;

    3 :                         // 送信要求
        WkSocket.DstAdr.PortNo:=UINT#6000;
        WkSocket.DstAdr.IpAdr :='192.168.250.1';
        SktUDPSend_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket,                   // ソケット
            SendDat   :=SendSocketDat[0],           // 送信データ
            Size      :=UINT#2000);                 // 送信データサイズ

        IF (SktUDPSend_instance.Done=TRUE) THEN
            Stage     :=INT#4;                      // 正常終了
        ELSIF (SktUDPSend_instance.Error=TRUE) THEN
            Stage     :=INT#30;                     // 異常終了
        END_IF;

    4 :                         // クローズ要求
        SktClose_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket);                  // ソケット

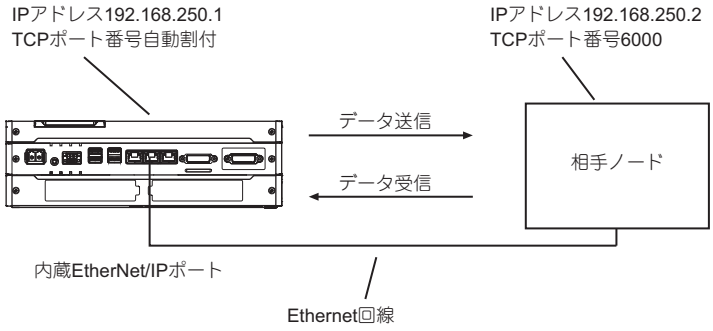
        IF (SktClose_instance.Done=TRUE) THEN
            Stage     :=INT#0;                      // 正常終了
        ELSIF (SktClose_instance.Error=TRUE) THEN
            Stage     :=INT#40;                     // 異常終了
        END_IF;

    0 :                         // 正常終了
        DoSendAndRcv:=FALSE;
        Trigger      :=FALSE;
    ELSE                        // 異常による中断
        DoSendAndRcv:=FALSE;
        Trigger      :=FALSE;
    END_CASE;

END_IF;
```

9-6-5　TCP のサンプルプログラム

NY シリーズコントローラと、相手ノードとの間で、ソケットサービス機能 (TCP) を使用して、データの送受信を行います。



処理の手順は、以下のとおりです。

1. **SktTCPConnect 命令で、相手ノードの TCP ポートにコネクト要求を行います。**
2. **SktClearBuf 命令で、TCP ソケットの受信バッファをクリアします。**
3. **SktGetTCPStatus 命令で、TCP ソケットのステータスを読み出します。**
4. **SktTCPSend 命令で、送信要求を行います。送信するデータは、SendSocketDat[] の中身とします。**
5. **SktTCPRcv 命令で、受信要求を行います。受信したデータは、RcvSocketDat[] に格納します。**
6. **SktClose 命令で、ソケットをクローズします。**

ST

内部変数	名称	データ型 0	初期値	コメント
	Trigger	BOOL	False	実行条件
	DoTCP	BOOL	False	処理中
	Stage	INT	0	状態遷移
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	受信データ
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0,IpAdr:=''), DstAdr:=(PortNo:=0, IpAdr:=''))	ソケット
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	送信データ
	SktTCPConnect_instance	SktTCPConnect		
	SktClearBuf_instance	SktClearBuf		
	SktGetTCPStatus_instance	SktGetTCPStatus		
	SktTCPSend_instance	SktTCPSend		
	SktTCPRcv_instance	SktTCPRcv		
	SktClose_instance	SktClose		

外部変数	名称	データ型	コンスタント	コメント
	_EIP_EtnOnlineSta*1	BOOL	☑	オンライン

*1. 使用する通信ポートによって、_EIP1_EtnOnlineSta（通信ポート 1 オンライン）、または _EIPIn1_EtnOnlineSta（内部通信ポート 1 オンライン）のいずれかに置き換えてください。

```
// Trigger の立ち上がりでシーケンス開始
IF ((Trigger=TRUE) AND (DoTCP=FALSE) AND (_EIP_EtnOnlineSta=TRUE)) THEN
    DoTCP           :=TRUE;
    Stage           :=INT#1;
    SktTCPConnect_instance(Execute:=FALSE);  // インスタンス初期化
    SktClearBuf_instance(Execute:=FALSE);    // インスタンス初期化
    SktGetTCPStatus_instance(Execute:=FALSE);// インスタンス初期化
    SktTCPSend_instance(                     // インスタンス初期化
        Execute     :=FALSE,
        SendDat     :=SendSocketDat[0]);     // ダミー
    SktTCPRcv_instance(                      // インスタンス初期化
        Execute     :=FALSE,
        RcvDat      :=RcvSocketDat[0]);      // ダミー
    SktClose_instance(Execute:=FALSE);       // インスタンス初期化
END_IF;

IF (DoTCP=TRUE) THEN
    CASE Stage OF
    1 :                      // コネクト要求
        SktTCPConnect_instance(
            Execute   :=TRUE,
            SrcTcpPort:=UINT#0,              // 自 TCP ポート番号　自動割付
            DstAdr    :='192.168.250.2',     // 相手 IP アドレス
            DstTcpPort:=UINT#6000,           // 相手 TCP ポート番号
            Socket    =>WkSocket);           // ソケット

        IF (SktTCPConnect_instance.Done=TRUE) THEN
            Stage     :=INT#2;               // 正常終了
        ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
            Stage     :=INT#10;              // 異常終了
        END_IF;

    2 :                      // 受信バッファクリア
        SktClearBuf_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket);           // ソケット

        IF (SktClearBuf_instance.Done=TRUE) THEN
            Stage     :=INT#3;               // 正常終了
        ELSIF (SktClearBuf_instance.Error=TRUE) THEN
            Stage     :=INT#20;              // 異常終了
        END_IF;
```

```
3 :                         // ステータス読出要求
   SktGetTCPStatus_instance(
      Execute   :=TRUE,
      Socket    :=WkSocket);                  // ソケット

   IF (SktGetTCPStatus_instance.Done=TRUE) THEN
      Stage     :=INT#4;                      // 正常終了
   ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
      Stage     :=INT#30;                     // 異常終了
   END_IF;

4 :                         // 送信要求
   SktTCPSend_instance(
      Execute   :=TRUE,
      Socket    :=WkSocket,                   // ソケット
      SendDat   :=SendSocketDat[0],           // 送信データ
      Size      :=UINT#2000);                 // 送信データサイズ

   IF (SktTCPSend_instance.Done=TRUE) THEN
      Stage     :=INT#5;                      // 正常終了
   ELSIF (SktTCPSend_instance.Error=TRUE) THEN
      Stage     :=INT#40;                     // 異常終了
   END_IF;

5 :                         // 受信要求
   SktTCPRcv_instance(
      Execute   :=TRUE,
      Socket    :=WkSocket,                   // ソケット
      TimeOut   :=UINT#0,                     // タイムアウト時間
      Size      :=UINT#2000,                  // 受信データサイズ
      RcvDat    :=RcvSocketDat[0]);           // 受信データ

   IF (SktTCPRcv_instance.Done=TRUE) THEN
      Stage     :=INT#6;                      // 正常終了
   ELSIF (SktTCPRcv_instance.Error=TRUE) THEN
      Stage     :=INT#50;                     // 異常終了
   END_IF;

6 :                         // クローズ要求
   SktClose_instance(
      Execute   :=TRUE,
      Socket    :=WkSocket);                  // ソケット

   IF (SktClose_instance.Done=TRUE) THEN
      Stage     :=INT#0;                      // 正常終了
   ELSIF (SktClose_instance.Error=TRUE) THEN
      Stage     :=INT#60;                     // 異常終了
   END_IF;
```

```
    0 :                     // 正常終了
        DoTCP       :=FALSE;
        Trigger     :=FALSE;

    ELSE                    // 異常による中断
        DoTCP       :=FALSE;
        Trigger     :=FALSE;
    END_CASE;

END_IF;
```

● 相手ノードのユーザプログラム

この例では、相手ノードにもユーザプログラムが必要です。上記の処理手順とは、送信と受信の順序が逆になります。

1 **SktTCPAccept 命令で、TCP ソケットのアクセプト要求を行います。**

2 **SktTCPRcv 命令で、受信要求を行います。受信したデータは、RcvSocketDat[] に格納します。**

3 **SktTCPSend 命令で、送信要求を行います。送信するデータは、SendSocketDat[] の中身とします。**

4 **SktClose 命令で、ソケットをクローズします。**

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
	DoTCP	BOOL	False	処理中
	Stage	INT	0	状態遷移
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	受信データ
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0,IpAdr:=''), DstAdr:=(PortNo:=0,IpAdr:=''))	ソケット
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	送信データ
	SktTCPAccept_instance	SktTCPAccept		
	SktTCPSend_instance	SktTCPSend		
	SktTCPRcv_instance	SktTCPRcv		
	SktClose_instance	SktClose		

外部変数	名称	データ型	定数	コメント
	_EIP_EtnOnlineSta*1	BOOL	☑	オンライン

*1. 使用する通信ポートによって、_EIP1_EtnOnlineSta（通信ポート 1 オンライン）、または _EIPIn1_EtnOnlineSta（内部通信ポート 1 オンライン）のいずれかに置き換えてください。

```
// Trigger の立ち上がりでシーケンス開始
IF ((Trigger=TRUE) AND (DoTCP=FALSE) AND (_EIP_EtnOnlineSta=TRUE)) THEN
    DoTCP                   :=TRUE;
    Stage                   :=INT#1;
    SktTCPAccept_instance(Execute:=FALSE);    // インスタンス初期化
    SktTCPSend_instance(                      // インスタンス初期化
        Execute     :=FALSE,
        SendDat     :=SendSocketDat[0]);      // ダミー
    SktTCPRcv_instance(                       // インスタンス初期化
        Execute     :=FALSE,
        RcvDat      :=RcvSocketDat[0]);       // ダミー
    SktClose_instance(Execute:=FALSE);        // インスタンス初期化
END_IF;

IF (DoTCP=TRUE) THEN
    CASE Stage OF
    1 :                     // アクセプト要求
        SktTCPAccept_instance(
            Execute   :=TRUE,
            SrcTcpPort:=UINT#6000,            // 自 TCP ポート番号
            TimeOut   :=UINT#0,               // タイムアウト時間
            Socket    =>WkSocket);            // ソケット

        IF (SktTCPAccept_instance.Done=TRUE) THEN
            Stage     :=INT#2;                // 正常終了
        ELSIF (SktTCPAccept_instance.Error=TRUE) THEN
            Stage     :=INT#10;               // 異常終了
        END_IF;

    2 :                     // 受信要求
        SktTCPRcv_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket,             // ソケット
            TimeOut   :=UINT#0,               // タイムアウト時間
            Size      :=UINT#2000,            // 受信データサイズ
            RcvDat    :=RcvSocketDat[0]);     // 受信データ

        IF (SktTCPRcv_instance.Done=TRUE) THEN
            Stage     :=INT#3;                // 正常終了
        ELSIF (SktTCPRcv_instance.Error=TRUE) THEN
            Stage     :=INT#20;               // 異常終了
        END_IF;
```

```
    3 :                         // 送信要求
        SendSocketDat:=RcvSocketDat;
        SktTCPSend_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket,                   // ソケット
            SendDat   :=SendSocketDat[0],           // 送信データ
            Size      :=UINT#2000);                 // 送信データサイズ

        IF (SktTCPSend_instance.Done=TRUE) THEN
            Stage     :=INT#4;                      // 正常終了
        ELSIF (SktTCPSend_instance.Error=TRUE) THEN
            Stage     :=INT#30;                     // 異常終了
        END_IF;

    4 :                         // クローズ要求
        SktClose_instance(
            Execute   :=TRUE,
            Socket    :=WkSocket);                  // ソケット

        IF (SktClose_instance.Done=TRUE) THEN
            Stage     :=INT#0;                      // 正常終了
        ELSIF (SktClose_instance.Error=TRUE) THEN
            Stage     :=INT#40;                     // 異常終了
        END_IF;

    0 :                         // 正常終了
        DoTCP     :=FALSE;
        Trigger   :=FALSE;
    ELSE                        // 異常による中断
        DoTCP     :=FALSE;
        Trigger   :=FALSE;
    END_CASE;

END_IF;
```

9-7　ソケットサービス使用上の留意事項

9-7-1　UDP/TCP ソケットサービス共通の留意事項

- 内蔵 EtherNet/IP ポー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。

- クローズ要求命令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命令の送信データが、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、内蔵 EtherNet/IP ポートは、TCP 1 ソケット当たり 9000 バイト、UDP 1 ソケット当たり 9000 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。

9-7-2　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 アドレスには、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、1472 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 1473 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。

9-7-3　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 ソケットのステータス読出し命令（SktGetTCPstatus）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。

- 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、SktTCPConnect 命令で使用するポートを 0（ユニットによる自動割付）とした場合はあてはまりません。

- コネクションは、Accept 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 Connect オープンを行うことで開設されます。
  Connect 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 Connect 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、Accept 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 Accept 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、内蔵 EtherNet/IP ポートで Accept 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 Connect オープンによる複数コネクションの開設はできません。

- 内蔵 EtherNet/IP ポートの TCP ソケットでは、キープアライブ（KeepAlive）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。
  内蔵 EtherNet/IP ポート側で KeepAlive 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。

- TCP ソケットでは、送信データに対し相手ノードから肯定応答（ACK）が無い場合、送信データを最大 12 回再送します。再送する間隔は最小 1 秒から最大 64 秒の間で、再送するごとに増えます。

- TCP ソケットでは、Connect オープンを行うことでコネクション確立要求（SYN）を送信します。相手ノードからコネクション確立の確認応答（SYN + ACK）が無い場合、SYN を最大 4 回再送します。
  オープン処理を行ってから 75 秒経過しても SYN ACK の受信が無い場合、エラーとなります。

10

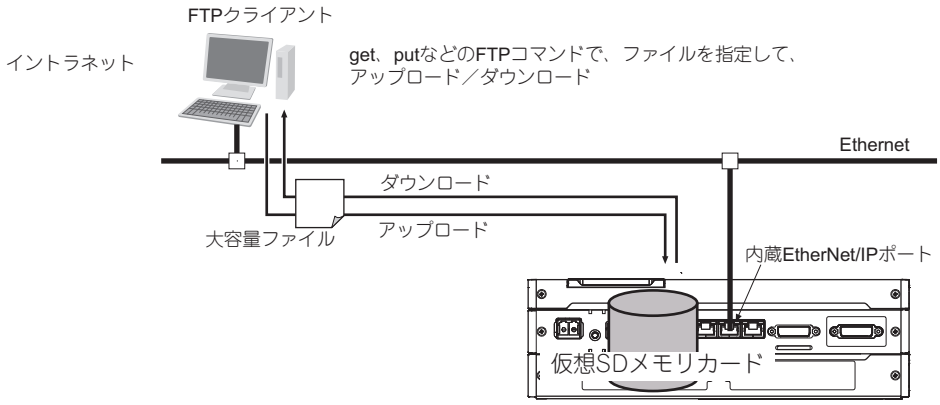
FTP サーバ機能

10-1 FTP サーバ機能の概要 10-2
10-1-1 概要 10-2
10-1-2 仕様 10-3
10-2 FTP サーバ機能の詳細 10-4
10-2-1 対象ファイル 10-4
10-2-2 FTP サーバへの接続方法 10-4
10-3 FTP サーバ機能の使用手順 10-7
10-3-1 使用手順 10-7
10-3-2 FTP サーバ機能に必要な設定一覧 10-7
10-4 FTP サーバ機能の使用例 10-8
10-5 FTP コマンドの使用方法 10-10
10-5-1 コマンド一覧 10-10
10-5-2 コマンドの使用方法 10-11
10-6 SD メモリカード機能の使い方 10-17
10-6-1 ファイルの種類 10-18
10-6-2 変数データのフォーマット 10-19
10-7 ホストコンピュータからの使用例 10-20

10-1 FTP サーバ機能の概要

10-1-1 概要

内蔵 EtherNet/IP ポートは、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、仮想 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。



使用上の注意

仮想 SD メモリカードを使用するためには、コントローラの SD メモリカード機能の設定が必要です。SD メモリカード機能の設定は、🕮『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』を参照してください。

10-1-2 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） mls　（複数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） mdir　（複数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） append　（現在指定しているファイルのデータタイプを使って、ローカルファイルをリモートホストに追加する） 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（12 文字以内） パスワード（8 ～ 32 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	6

10-2 FTP サーバ機能の詳細

10-2-1 対象ファイル

内蔵 EtherNet/IP ポートが扱うコントローラ上のファイルシステムは、NY シリーズコントローラの の仮想 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

- MEMCARD1:SD
- usr
- bin
- dev

（usr、bin、dev：システムで使用します）

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 MEMCARD1 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。
- MEMCARD1 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。

10-2-2 FTP サーバへの接続方法

FTP クライアントソフトから、FTP のログイン名およびパスワードを入力して、内蔵 EtherNet/IP ポートにログインします。FTP のログイン名およびパスワードは、Sysmac Studio の内蔵 EtherNet/IP ポート設定で設定します。

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内蔵 EtherNet/IP ポー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 FTP server ready. xxx.xx.xx.xx：コントローラの形式（例：NY512-1500-131513F10）
最大コネクション数 (6) で接続中	530 FTP server busy, Goodbye.

● ログイン名とパスワードの設定の制約事項

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。
- ログイン名の長さは 12 文字までです。
- パスワードの長さは 8 ～ 32 文字です。
- 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

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。
コントローラは、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仮想 SD メモリカードアクセス時の排他制御

<table>
<tr><th colspan="3" rowspan="3"></th><th colspan="6">先に実行中のアクセス</th></tr>
<tr><th colspan="2">命令*1</th><th colspan="2">Sysmac Studio からのファイル操作</th><th colspan="2">FTP サーバ機能</th></tr>
<tr><th>読出</th><th>書込</th><th>読出</th><th>書込</th><th>読出</th><th>書込</th></tr>
<tr><td rowspan="6">後から実行するアクセス</td><td rowspan="2">命令</td><td>読出</td><td colspan="2" rowspan="2">自動的に排他制御され、後から実行する命令がエラーとなります。</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><td>排他制御を行ってください。</td></tr>
<tr><td>書込</td><td colspan="2">排他制御を行ってください。</td><td colspan="2">排他制御を行ってください。</td></tr>
<tr><td rowspan="2">Sysmac Studio からのファイル操作</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><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><td>排他制御を行ってください。</td></tr>
<tr><td>書込</td><td colspan="2">排他制御を行ってください。</td><td colspan="2">排他制御を行ってください。</td><td colspan="2">排他制御を行ってください。</td></tr>
<tr><td rowspan="2">FTP サーバ機能</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><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><td>排他制御を行ってください。</td></tr>
<tr><td>書込</td><td colspan="2">排他制御を行ってください。</td><td colspan="2">排他制御を行ってください</td><td colspan="2">排他制御を行ってください</td></tr>
</table>

*1. 命令には、SD メモリカード操作命令と FTP クライアント通信命令が含まれます。

● FTP サーバへの接続の制限

FTP サーバへの接続と切断を短時間に頻繁に繰り返すと、システム保護のため一時的に接続を制限することがあります。FTP サーバへの接続ができなくなった場合、10 分以上たってから再度接続してください。

10-3 FTP サーバ機能の使用手順

10-3-1 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。(🕮「5-3 ［FTP 設定］ダイアログ」参照)**

3 **Sysmac Studio にて［コントローラ設定］|［内蔵 EtherNet/IP ポート設定］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 **コントローラ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 **FTP クライアントソフトから、内蔵 EtherNet/IP ポートに接続します。**

6 **内蔵 EtherNet/IP ポート設定に設定したログイン名、パスワードを入力して、内蔵 EtherNet/IP ポートにログインします。**

7 **コントローラの仮想 SD メモリカード内のディレクトリ「MEMCARD1」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。**

8 **接続を解除します。**

10-3-2 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 内蔵 EtherNet/IP ポート設定のタブ	項目名	設定が必要な条件	参照ページ
FTP	FTP サーバ	必須	🕮P.5-7
	ポート No.	任意（*） * 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、🕮「5-3 ［FTP 設定］ダイアログ」を参照してください。

10-4 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。

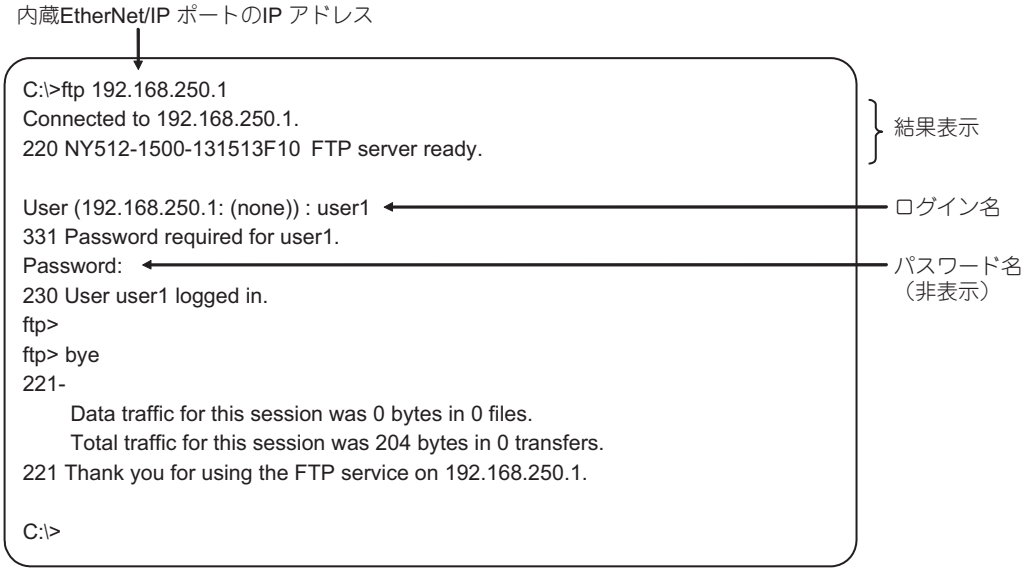
参考

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**

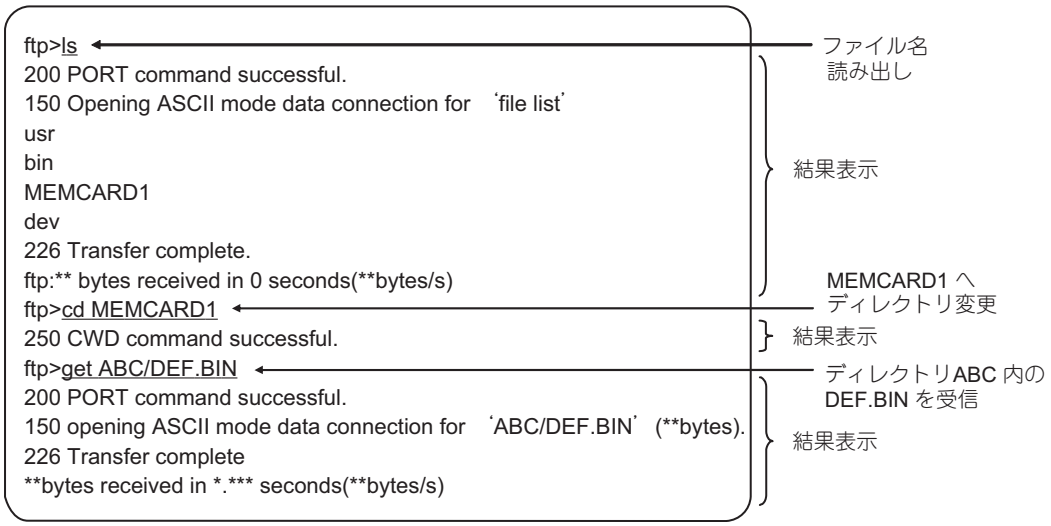
(1) 仮想SDメモリカードの設定をしていることを確認し、コントローラの電源をONにします。

(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。

/（ルートディレクトリ）
└ MEMCARD1
　└ ABC（サブディレクトリ）
　　└ DEF.BIN（ファイル）



10-5 FTP コマンドの使用方法

内蔵 EtherNet/IP ポートの 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。

10-5-1 コマンド一覧

内蔵 EtherNet/IP ポー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。
mls	複数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。
mdir	複数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。
append	type コマンドで指定したファイルのデータタイプを使って、ローカルファイルをリモートホストに追加する。
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」とは内蔵 EtherNet/IP ポートを示します。
2 「リモートファイル」とは、仮想 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。

10-5-2 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。内蔵 EtherNet/IP ポー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。
- FTP サーバへ接続した直後は、自動的にユーザ名を質問してきます。

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（仮想 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、仮想 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。

mls

● **書式**

mls リモートファイル名 ローカルファイル名

● **機能詳細**

- 該当する複数のリモートホスト（仮想 SD メモリカード上）のファイル名の一覧を表示します。
- 「リモートファイル名」には一覧を表示する SD メモリカード上のディレクトリ、またはファイル名を指定します。現在、作業中のディレクトリを一覧表示する場合は、アスタリスク (*) を入力します。
- 「ローカルファイル名」を指定するとファイル情報を指定ファイルに格納します。ファイル名の一覧を格納せず、リモートホストの一覧表示をする場合は、ハイフン (-) を入力します。

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（仮想 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、仮想 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。

mdir

● 書式

mdir リモートファイル名 ローカルファイル名

● 機能詳細

- 該当する複数のリモートホスト（仮想 SD メモリカード上）のファイル名とサブディレクトリの一覧、作成日時、容量を表示します。
- 「リモートファイル名」には一覧を表示する仮想 SD メモリカード上のディレクトリ、またはファイル名を指定します。現在、作業中のディレクトリを一覧表示する場合は、ハイフン (-) を入力します。
- 「ローカルファイル名」を指定するとファイル情報を指定ファイルに格納します。ファイル情報を格納せず、リモートホストの一覧表示をする場合は、ハイフン (-) を入力します。

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（仮想 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（仮想 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（仮想 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。

append

● **書式**

append ローカルファイル名 [リモートファイル名]

● **機能詳細**

- type コマンドで指定したファイルのデータタイプを使って、ローカルファイルをリモートホスト（仮想 SD メモリカード上）に追加します。

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。
- 仮想 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 MEMCARD1 ディレクトリにあります。
- 内蔵 EtherNet/IP ポー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。コントローラに仮想 SD メモリカードの設定がされていないときは、ディレクトリ MEMCARD1 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。
  コントローラ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(仮想 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（仮想 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（仮想 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(仮想 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。

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（仮想 SD メモリカード）に転送します。
- リモートホスト(仮想 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。

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（仮想 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**

- 接続している内蔵 EtherNet/IP ポートの 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**

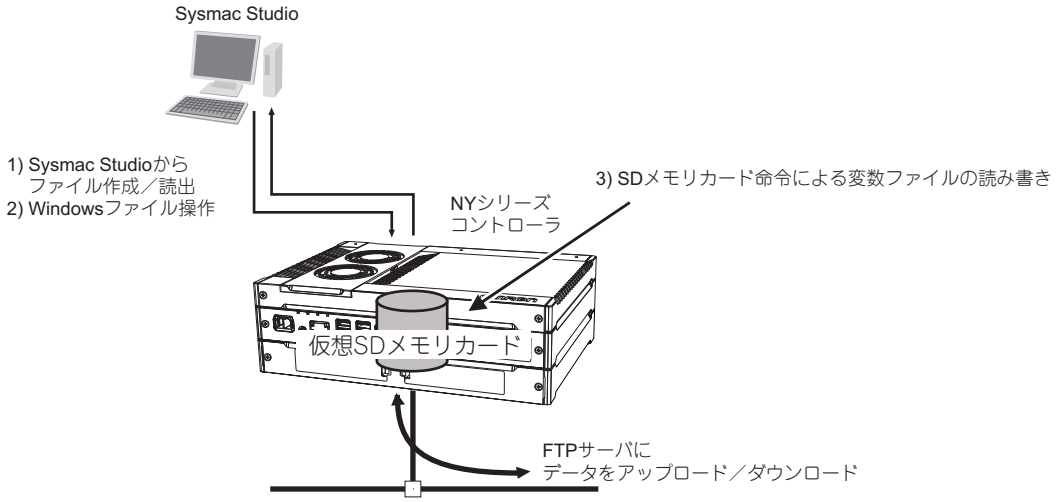
- FTP を終了します。

10-6 SD メモリカード機能の使い方

内蔵 EtherNet/IP ポートでは、NY シリーズコントローラの仮想 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）

コントローラは次の 3 つの方法で仮想 SD メモリカードにデータを格納したり読み出します。



使用上の注意

仮想 SD メモリカードを使用するためには、コントローラの SD メモリカード機能の設定が必要です。SD メモリカード機能の設定は、『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』を参照してください。

10-6-1　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。
「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、拡張子を含めて 65 文字です。
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

仮想 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。

コントローラで取り扱うファイル名

コントローラで取り扱う（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。

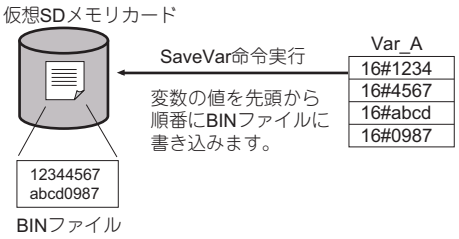
ファイル種類	ファイル名	拡張子	内容	説明
変数ファイル（バイナリ形式）	🕮「10-6-1 ファイルの種類」参照	.bin	指定変数	指定した複数の変数（配列、構造体を含む）の値を、バイナリ形式（.bin）で記載したファイルです。

詳細は、🕮『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』を参照してください。

10-6-2 変数データのフォーマット

BIN 形式

コントローラのラダー命令（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

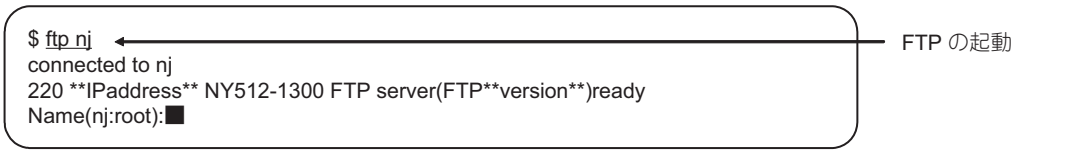
- NY シリーズコントローラでバイナリファイルを取り扱う場合、FTP でファイルの読み出し／書き込みする前に、必ず type コマンドでデータタイプを binary に指定してください。（🕮「10-5-2 コマンドの使用方法」参照）
- ラダーのファイル処理命令の使用方法については、🕮『NY シリーズ コマンドリファレンスマニュアル 基本編（SBCA-437）』を参照してください。

10-7 ホストコンピュータからの使用例

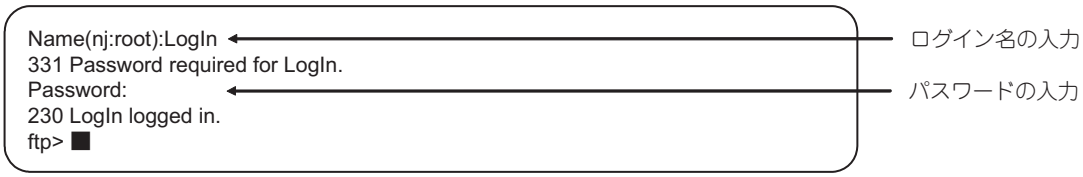
ホストコンピュータから次の手順で操作します。この例では次のように設定されているものとします。
- ホストコンピュータの hosts 内に内蔵 EtherNet/IP ポートの IP アドレスがホスト名「nj」として登録されていることを想定しています。
- FTP ログイン名は「LogIn」です。
- NY シリーズコントローラの仮想 SD メモリカード上に製造結果のデータがファイル名「RESULT.BIN」で格納されています。
- ワークステーション上に製造指示データがファイル名「PLAN. BIN」で格納されています。

以下の操作では、仮想 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」を仮想 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。

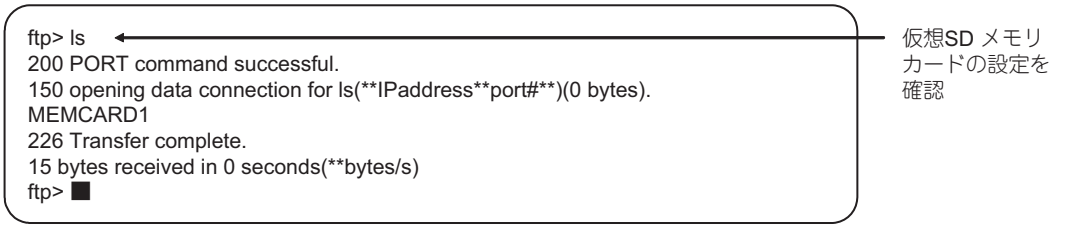
(1) FTP を起動し、内蔵 EtherNet/IP ポートに接続します。



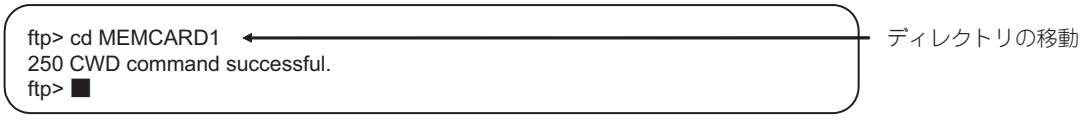
(2) ログイン名を入力します。



(3) 仮想 SD メモリカードの設定が正しいことを確認します。仮想 SD メモリカードの設定が正しい場合、ディレクトリ MEMCARD1 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」を仮想 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

11

FTP クライアント機能

11-1 FTP クライアント機能によるファイル転送 11-2
11-1-1 ファイル転送の実行方法 11-2
11-1-2 接続可能な FTP サーバ 11-3
11-1-3 ファイル転送のオプション機能 11-3
11-1-4 その他の機能 11-4
11-2 FTP クライアント通信命令 11-5
11-2-1 FTP クライアント通信命令の機能 11-5
11-2-2 FTP クライアント通信命令の制限 11-7
11-3 FTP クライアント機能の使用例 11-8

11-1 FTP クライアント機能によるファイル転送

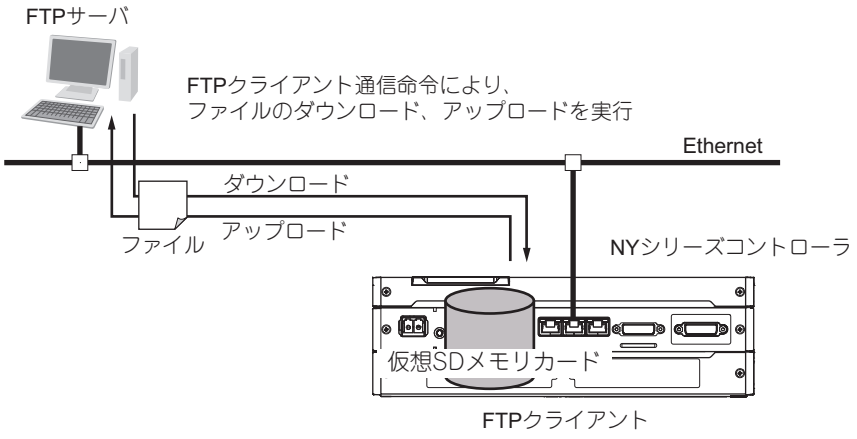
FTP クライアント機能を使って、FTP サーバと FTP クライアント間で、ファイル転送を行うことができます。ファイル転送の方向は、FTP サーバから FTP クライアント（ダウンロード）、FTP クライアントから FTP サーバ（アップロード）の、両方が実行可能です。

11-1-1 ファイル転送の実行方法

FTP クライアント機能によるファイル転送は、すべてユーザプログラム内の FTP クライアント通信命令を用いて行います。ファイル転送に関する設定も、すべて FTP クライアント通信命令のパラメータ指定で行います。したがって、Sysmac Studio の設定は必要ありません。
FTP クライアント通信命令の種類と機能は、以下のとおりです。FTP クライアント通信命令の同時実行数は、最大 3 個です。

命令	機能
FTPGetFileList	FTP サーバ内のファイルのリストを取得します。
FTPGetFile	FTP サーバからファイルをダウンロードします。
FTPPutFile	FTP サーバへファイルをアップロードします。
FTPRemoveFile	FTP サーバのファイルを削除します。
FTPRemoveDir	FTP サーバのディレクトリを削除します。

ダウンロードしたファイルは、仮想 SD メモリカードに格納されます。また、アップロードする場合も、仮想 SD メモリカードに格納されているファイルが FTP サーバに転送されます。したがって、ダウンロードする場合も、アップロードする場合も、NY シリーズコントローラに仮想 SD メモリカードの設定をしておく必要があります。



使用上の注意

仮想 SD メモリカードを使用するためには、コントローラの SD メモリカード機能の設定が必要です。SD メモリカード機能の設定は、🕮『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』を参照してください。

11-1-2 接続可能な FTP サーバ

NY シリーズコントローラが接続可能な FTP サーバは、以下のとおりです。FTP サーバの設定や使用方法の詳細については、それぞれのマニュアルをご確認ください。

- NY シリーズコントローラ内蔵 EtherNet/IP ポート
- NJ/NX シリーズコントローラ内蔵 EtherNet/IP ポート
- CJ シリーズ EtherNet/IP ユニット（Ver. 2.0 以降）
- CJ シリーズ CJ2（EtherNet/IP 機能付き) CPU ユニット
- CJ シリーズ CJ1M（Ethernet 機能付き) CPU ユニット
- CJ シリーズ Ethernet ユニット
- Windows 7/Windows Server2008 R2(Internet Information Services(IIS)7.5)
- Windows8/Windows Server2012(IIS8.0)
- Windows10/Windows Server2016(IIS10.0)
- Linux

11-1-3 ファイル転送のオプション機能

ファイル転送の際のオプション機能として、以下のものが使用できます。いずれも、FTP クライアント通信命令のパラメータとして指定します。

- ファイル転送モード
- データコネクションのオープンモード
- 転送後ファイル削除の有無
- 上書きの有無

以下に、それぞれのオプション機能について説明します。

● ファイル転送モード

ファイル転送には、テキストデータの改行コードの処理の違いによって、ASCII モードと Binary モードの、2 種類のモードがあります。それぞれの違いを、以下に示します。

転送モード	テキストデータの改行コードの処理
ASCII モード	UNIX、Windows など、転送先のシステムの改行コードに変換して転送します。
Binary モード	変換せず、そのまま転送します。

● データコネクションのオープンモード

ファイル転送を行う際には、FTP サーバと FTP クライアントの間で、TCP コネクションをオープンする必要があります。TCP コネクションには、通信制御のためのコントロールコネクションとデータ転送のためのデータコネクションがあります。このうち、データコネクションのオープンモードには、接続要求を出す側の違いによって、Active モードと Passive モードの 2 種類があります。それぞれの違いを、以下に示します。

オープンモード	データコネクションの接続要求
Active モード	FTP サーバが接続要求を出します。
Passive モード	FTP クライアントが接続要求を出します。

たとえば FTP サーバがインターネットの外に存在する場合、データコネクションのオープンモードが Active モードだと、FTP サーバからの接続要求がセキュリティポリシー上許可されない可能性があります。このような場合は、データコネクションのオープンモードを Passive モードにして、FTP クライアントから接続要求を出す必要があります。

● 転送後ファイル削除の有無

ファイルを転送した後に、元のファイルを削除するかどうかを指定することができます。何らかの原因でファイル転送に失敗した場合は、転送後ファイル削除の指定をしていても、元のファイルは削除されません。

● 上書きの有無

ファイルの転送先に、転送ファイルと同じ名前のファイルが存在する場合に、上書きするかしないかを指定することができます。上書きしない指定をしていて、転送先に転送ファイルと同じ名前のファイルが存在する場合は、ファイル転送が行われません。

11-1-4　その他の機能

ファイル転送のその他の機能として、以下の二つがあります。

- FTP サーバとの接続処理のリトライ
- 転送ファイルのワイルドカード指定

以下にそれぞれの機能を説明します。

● FTP サーバとの接続処理のリトライ

FTP サーバとの接続処理に失敗した場合、自動的に最大 3 回まで、接続処理をリトライします。接続失敗を判断するまでのタイムアウト時間、リトライする回数、リトライの間隔は、変更可能です。

● 転送ファイルのワイルドカード指定

転送するファイルのファイル名を、ワイルドカード指定することが可能です。それにより、複数のファイルを一度に転送することができます。

11-2 FTP クライアント通信命令

FTP クライアント機能によるファイル転送は、すべて FTP クライアント通信命令を用いて行います。
FTP クライアント通信命令の種類と機能は、以下のとおりです。

命令	機能
FTPGetFileList	FTP サーバ内のファイルのリストを取得します。
FTPGetFile	FTP サーバからファイルをダウンロードします。
FTPPutFile	FTP サーバへファイルをアップロードします。
FTPRemoveFile	FTP サーバのファイルを削除します。
FTPRemoveDir	FTP サーバのディレクトリを削除します。

FTP クライアント通信命令の使用の詳細は、🕮『NY シリーズ コマンドリファレンスマニュアル 基本編（SBCA-437)』を参照してください。

11-2-1 FTP クライアント通信命令の機能

FTP クライアント通信命令の各機能について説明します。

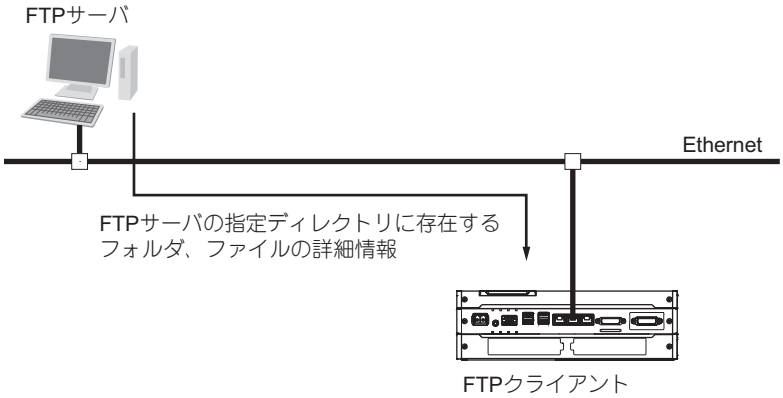
FTPGetFileList 命令

FTPGetFileList 命令は、FTP サーバ内の指定ディレクトリに存在するフォルダおよびファイルの、リストを取得する命令です。
取得する情報は、以下のとおりです。

- 指定ディレクトリに存在するフォルダおよびファイルの数
- 各フォルダおよびファイルの名前
- 各フォルダおよびファイルの更新日時
- 各ファイルのサイズ
- 各フォルダおよびファイルの読取専用属性

以下のオプション指定が可能です。

- データコネクションのオープンモード



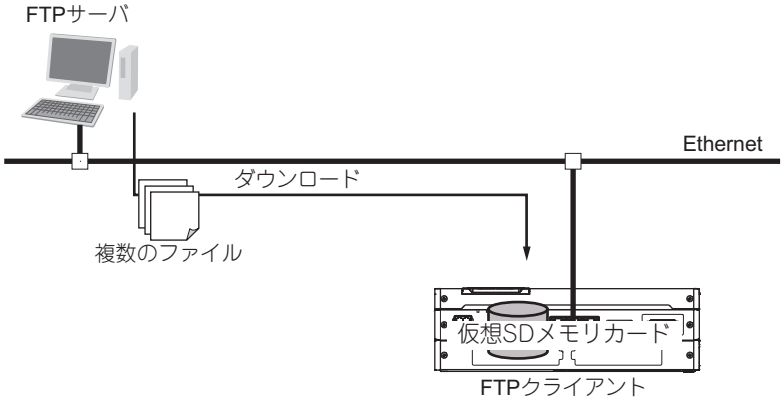
参考

CPU ユニット Ver.1.14 以降では、AM/PM 12 時台のファイル更新日時について改善しています。

FTPGetFile 命令

FTPGetFile 命令は、FTP サーバの指定ディレクトリの指定ファイルを、仮想 SD メモリカードの指定ディレクトリにダウンロードする命令です。
ファイル名をワイルドカードで指定すれば、複数のファイルを一度にダウンロードすることが可能です。
仮想 SD メモリカードにダウンロード先の指定ディレクトリが存在しない場合は、ディレクトリを作成したうえでダウンロードが実行されます。
以下のオプション指定が可能です。

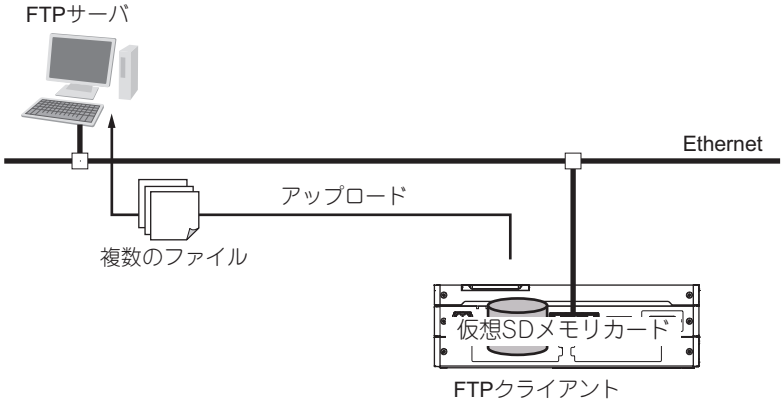
- 転送モード
- データコネクションのオープンモード
- 転送後ファイル削除の有無
- 上書きの有無



FTPPutFile 命令

FTPPutFile 命令は、仮想 SD メモリカードの指定ディレクトリの指定ファイルを、FTP サーバの指定ディレクトリにアップロードする命令です。
ファイル名をワイルドカードで指定すれば、複数のファイルを一度にアップロードすることが可能です。
FTP サーバにアップロード先の指定ディレクトリが存在しない場合は、ディレクトリを作成したうえでアップロードが実行されます。
以下のオプション指定が可能です。

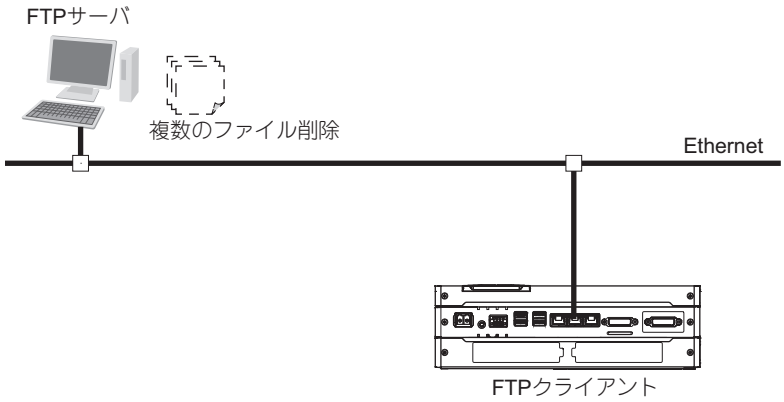
- 転送モード
- データコネクションのオープンモード
- 転送後ファイル削除の有無
- 上書きの有無



FTPRemoveFile 命令

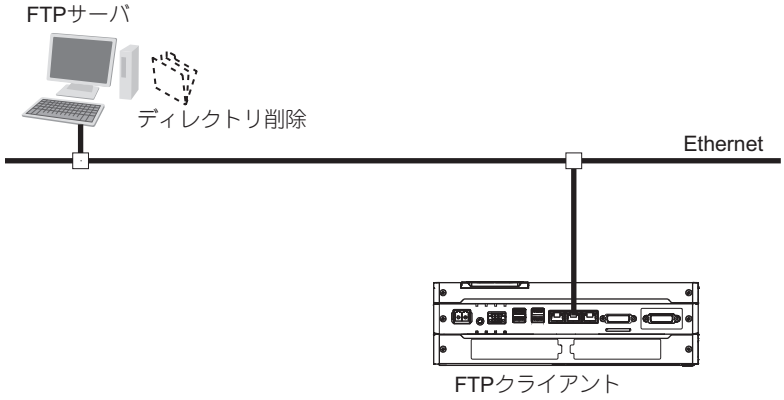
FTPRemoveFile 命令は、FTP サーバ内の指定ディレクトリの指定ファイルを、削除する命令です。
ファイル名をワイルドカードで指定すれば、複数のファイルを一度に削除することが可能です。
以下のオプション指定が可能です。

- データコネクションのオープンモード



FTPRemoveDir 命令

FTPRemoveDir 命令は、FTP サーバ内の指定ディレクトリを、削除する命令です。



11-2-2 FTP クライアント通信命令の制限

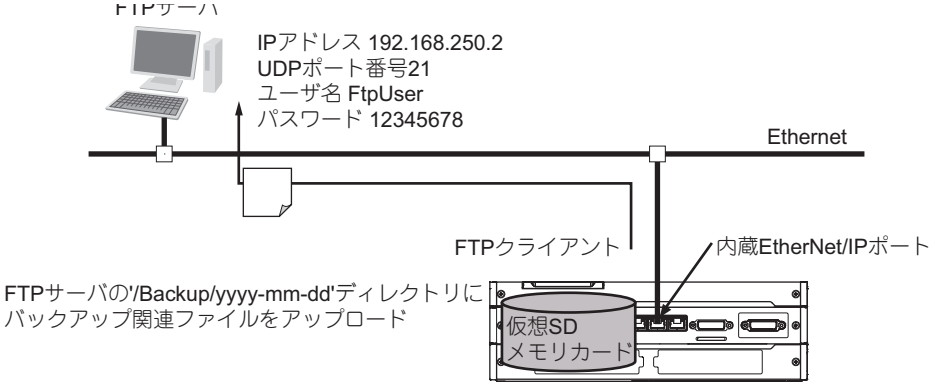
FTP クライアント通信命令には、以下のような制限があります。ユーザプログラム作成時には、注意してください。

- 仮想SDメモリカードを読み書きするFTPクライアント通信命令を複数同時に実行すると、書込途中のファイルを読み出すなど、意図しない処理が行われる可能性があります。ユーザプログラムの中で、命令の排他処理を行ってください。
- 仮想SDメモリカードを読み書きするFTPクライアント通信命令と、仮想SDメモリカードを読み書きする他の機能とを同時に実行する場合も、意図しない処理が行われる可能性があります。ユーザプログラムの中で、命令の排他処理を行って下さい。仮想 SD メモリカード内のファイルアクセスの排他制御の詳細は、🕮『NYシリーズ 産業用パネル型PC／産業用ボックス型PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』を参照してください。

11-3 FTP クライアント機能の使用例

FTP クライアント機能は、すべて FTP クライアント通信命令を用いて実行します。以下に、FTP クライアント通信命令を使用したサンプルプログラムを示します。

SD メモリカードバックアップを実行し、すべてのバックアップ関連ファイルを、FTP サーバの '/Backup/yyyy-mm-dd' ディレクトリにアップロードします。



コントローラと FTP サーバは、EtherNet/IP ネットワークで接続しています。FTP サーバと接続するための設定パラメータの値は、以下のとおりとします。

設定パラメータ	値
IP アドレス	192.168.250.2
UDP ポート番号	21
ユーザ名	FtpUser
パスワード	12345678

処理の手順は、以下のようになります。

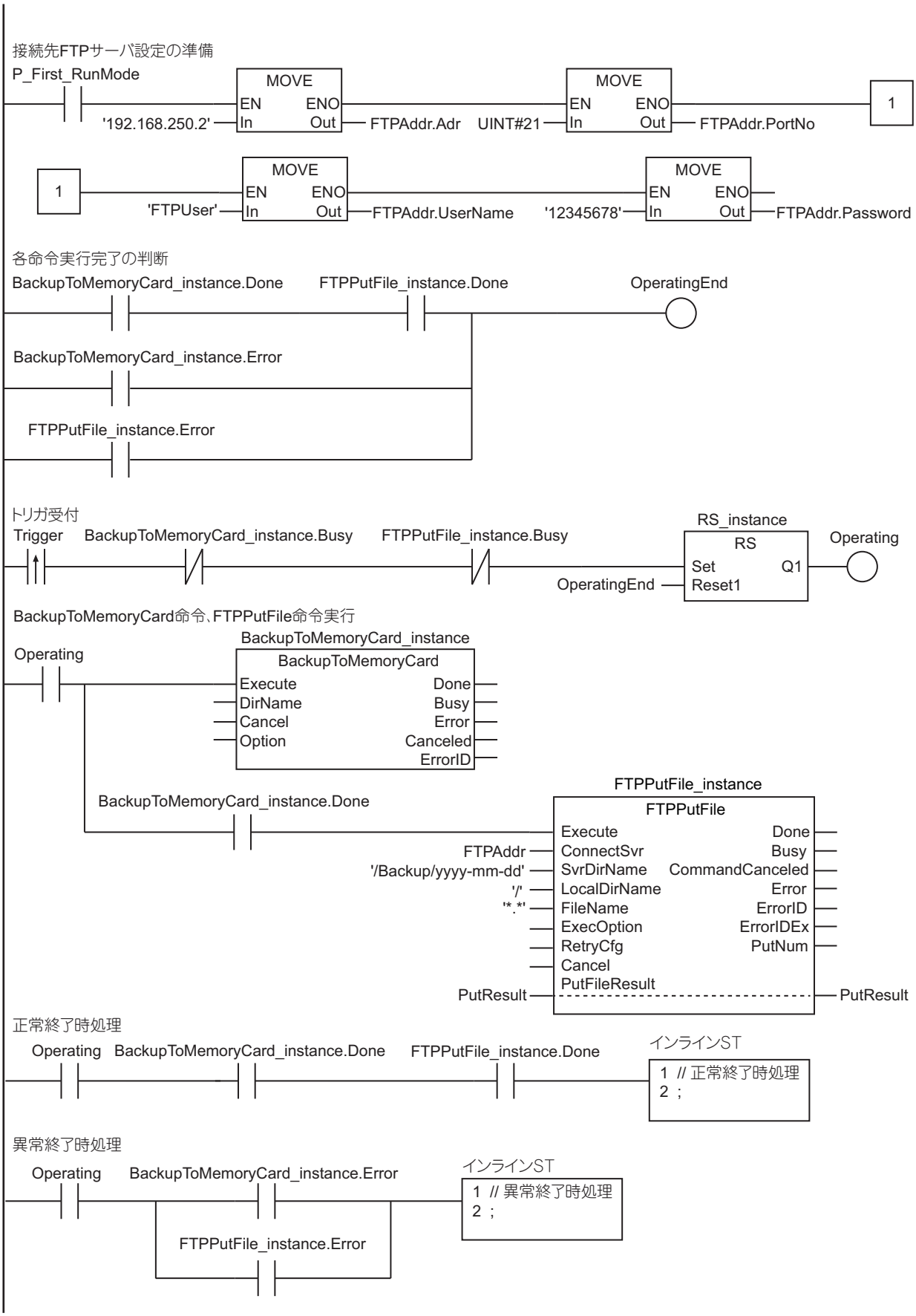
1 **BackupToMemoryCard 命令を使用して、NY シリーズ コントローラのバックアップ関連ファイルを、仮想 SD メモリカードのルートディレクトリに保存します。**

2 **FTPPutFile 命令を使用して、バックアップ関連ファイルを FTP サーバの '/Backup/yyyy-mm-dd' ディレクトリにアップロードします。転送ファイル名は、ワイルドカード指定で '*.*' とします。**

3 **すべての処理が正常に終了した場合は正常終了時処理を実行します。異常が発生した場合は異常終了時処理を実行します。**

LD

内部変数	名称	データ型	初期値	コメント
	FTPPutFile_instance	FTPPutFile		FTPPutFile 命令のインスタンス
	FTPAddr	_sFTP_CONNECT_SVR	(Adr := '', PortNo := 0, UserName := '', Password := '')	接続先 FTP サーバ設定
	PutResult	ARRAY[0..0] OF _sFTP_FILE_RESULT	[(Name := '', TxError := False, RemoveError := False, Reserved := [4(16#0)])]	アップロード済みファイル結果
	RS_instance	RS		RS 命令のインスタンス
	OperatingEnd	BOOL	FALSE	処理終了
	Trigger	BOOL	FALSE	実行条件
	Operating	BOOL	FALSE	処理中
	BackupToMemoryCard_instance	BackupToMemoryCard		BackupToMemoryCard 命令のインスタンス


接続先FTPサーバ設定の準備
P_First_RunMode
MOVE
EN
ENO
'192.168.250.2'
In
Out
FTPAddr.Adr
MOVE
UINT#21
FTPAddr.PortNo
1
'FTPUser'
FTPAddr.UserName
'12345678'
FTPAddr.Password
各命令実行完了の判断
BackupToMemoryCard_instance.Done
FTPPutFile_instance.Done
OperatingEnd
BackupToMemoryCard_instance.Error
FTPPutFile_instance.Error
トリガ受付
Trigger
BackupToMemoryCard_instance.Busy
FTPPutFile_instance.Busy
RS_instance
RS
Set
Q1
Reset1
Operating
BackupToMemoryCard命令、FTPPutFile命令実行
BackupToMemoryCard_instance
BackupToMemoryCard
Execute
DirName
Cancel
Option
Done
Busy
Error
Canceled
ErrorID
FTPPutFile_instance
FTPPutFile
FTPAddr
ConnectSvr
'/Backup/yyyy-mm-dd'
SvrDirName
'/'
LocalDirName
'*.*'
FileName
ExecOption
RetryCfg
Cancel
PutFileResult
CommandCanceled
ErrorIDEx
PutNum
PutResult
正常終了時処理
インラインST
1 // 正常終了時処理
2 ;
異常終了時処理
1 // 異常終了時処理
2 ;

ST

内部変数	名称	データ型	初期値	コメント
	R_TRIG_instance	R_TRIG		R_TRIG 命令のインスタンス
	UP_Q	BOOL	FALSE	トリガ出力
	FTPPutFile_instance	FTPPutFile		FTPPutFile 命令のインスタンス
	DoFTPTrigger	BOOL	FALSE	BackupToMemoryCard、FTPPutFIle 実行条件
	FTPAddr	_sFTP_CONNECT_SVR	(Adr := '', PortNo := 0, UserName := '', Password := '')	接続先 FTP サーバ設定
	PutResult	ARRAY[0..0] OF _sFTP_FILE_RESULT	[(Name := '', TxError := False, RemoveError := False, Reserved := [4(16#0)])]	アップロード済みファイル結果
	Stage	UINT	0	命令実行手順
	Trigger	BOOL	FALSE	実行条件
	BackupToMemoryCard_instance	BackupToMemoryCard		BackupToMemoryCard 命令のインスタンス

```
// 接続先 FTP サーバ設定の準備
IF P_First_RunMode THEN
  FTPAddr.Adr      := '192.168.250.2';      // アドレス
  FTPAddr.PortNo   := UINT#21;              // ポート番号
  FTPAddr.UserName := 'FtpUser';            // ユーザー名
  FTPAddr.Password := '12345678';           // パスワード
END_IF;

// トリガ受付
R_TRIG_instance(Trigger, UP_Q);
IF ( (UP_Q = TRUE) AND (BackupToMemoryCard_instance.Busy = FALSE) AND
    (FTPPutFile_instance.Busy = FALSE) ) THEN
  DoFTPTrigger := TRUE;
  Stage := INT#1;
  BackupToMemoryCard_instance(          // インスタンス初期化
    Execute := FALSE) ;
  FTPPutFile_instance(                  // インスタンス初期化
    Execute       := FALSE,
    ConnectSvr    := FTPAddr,
    SvrDirName    := '/Backup/yyyy-mm-dd',
    LocalDirName  := '/',
    FileName      := '*.*',
    PutFileResult := PutResult) ;
END_IF;
```

```
IF (DoFTPTrigger = TRUE) THEN
  CASE Stage OF
    1 : // BackupToMemoryCard 命令実行
      BackupToMemoryCard_instance(
        Execute := TRUE) ;                  // 起動
      IF (BackupToMemoryCard_instance.Done = TRUE) THEN
        Stage := INT#2;                     // 次へ
      ELSIF (BackupToMemoryCard_instance.Error = TRUE) THEN
        Stage := INT#10;                    // 異常終了
      END_IF;
    2 : // FTPPutFile 命令実行
      FTPPutFile_instance(
        Execute       := TRUE,              // 起動
        ConnectSvr    := FTPAddr,           // 接続先 FTP サーバ
        SvrDirName    := '/Backup/yyyy-mm-dd',// FTP サーバディレクトリ名
        LocalDirName  := '/',               // ローカルディレクトリ名
        FileName      := '*.*',             // ファイル名
        PutFileResult := PutResult) ;       // アップロードファイル結果
      IF (FTPPutFile_instance.Done = TRUE) THEN
        Stage := INT#0;                     // 正常終了
      ELSIF (FTPPutFile_instance.Error = TRUE) THEN
        Stage := INT#20;                    // 異常終了
      END_IF;
    0 : // 正常終了処理
      DoFTPTrigger:=FALSE;
      Trigger     :=FALSE;
    ELSE // 異常終了処理
      DoFTPTrigger:=FALSE;
      Trigger     :=FALSE;
  END_CASE;
END_IF;
```

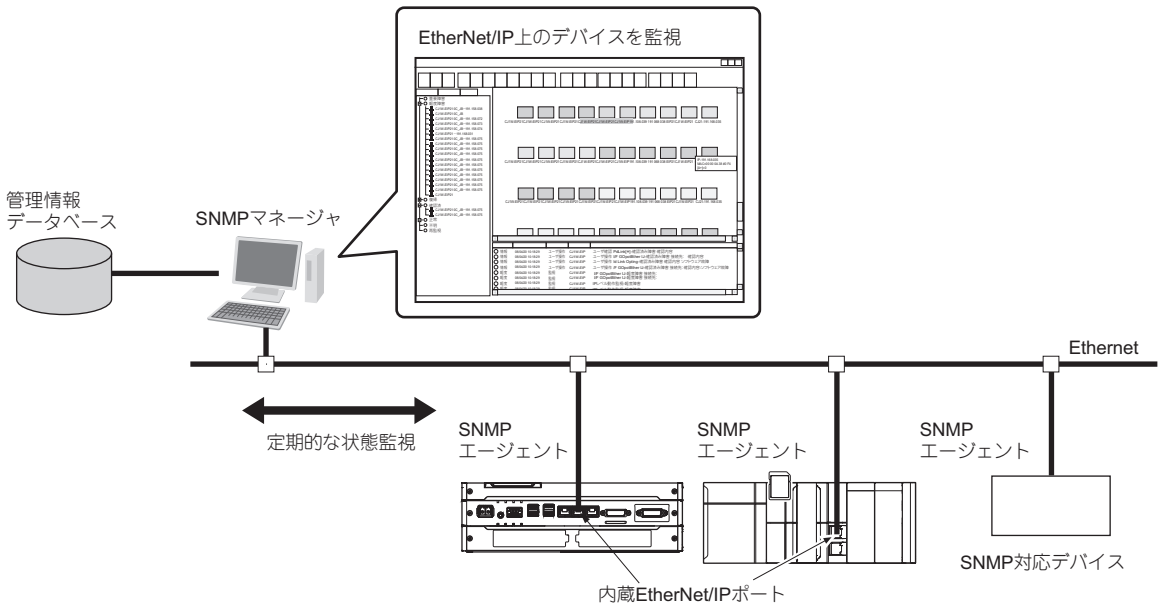
12

SNMP（エージェント）機能

12-1 SNMP（エージェント）機能の概要 12-2
12-1-1 SNMP（エージェント）機能とは 12-2
12-1-2 仕様 12-3
12-1-3 SNMP メッセージ 12-3
12-1-4 MIB の仕様 12-4
12-2 SNMP（エージェント）機能の使用手順 12-18
12-2-1 使用手順 12-18
12-2-2 SNMP（エージェント）機能に必要な設定一覧 12-18

12-1 SNMP（エージェント）機能の概要

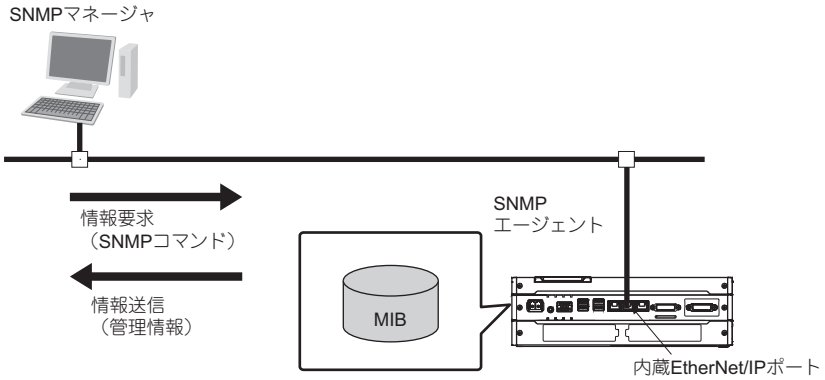
SNMP（simple network management protocol）は、ネットワーク管理プロトコルです。
SNMP をサポートしているネットワーク機器で構成されたネットワークの管理が可能です。
ネットワークを管理する側のサーバを SNMP マネージャ、管理される側のネットワーク機器を SNMP エージェントといいます



12-1-1 SNMP（エージェント）機能とは

SNMP エージェント機能

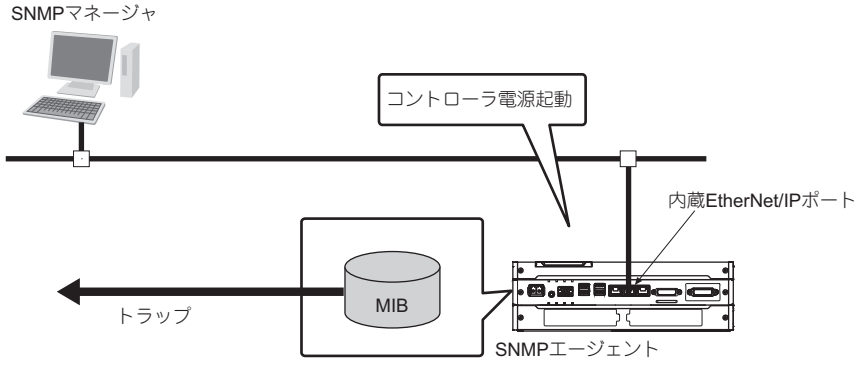
内蔵 EtherNet/IP ポートは、MIB（Management Information Base）という自身に関する管理情報を持ち、それを SNMP マネージャに提供することができます。
SNMP マネージャとは、SNMP ネットワークの各デバイス情報を収集・加工し、ネットワーク管理者に提供するソフトウェアです。
この SNMP マネージャにより、内蔵 EtherNet/IP ポートの監視を行うことができます。



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。
内蔵 EtherNet/IP ポートの 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内蔵 EtherNet/IP ポー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内蔵 EtherNet/IP ポー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。

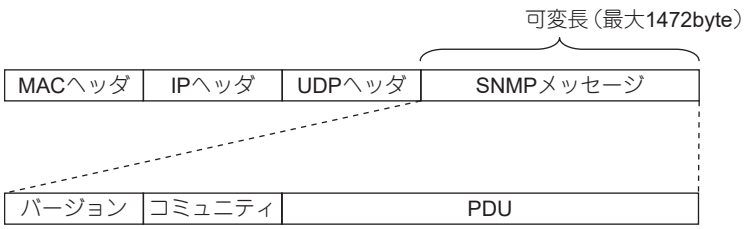


12-1-2　仕様

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 の内蔵 EtherNet/IP ポー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

12-1-3　SNMP メッセージ

SNMP メッセージの構造は次のとおりです。



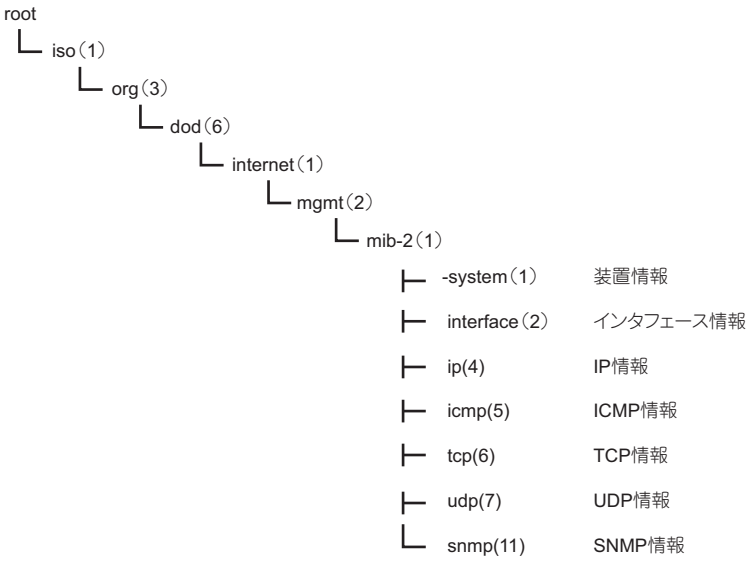
項目	設定値
バージョン	SNMP のバージョンを示す値。 SNMPv1　：0 SNMpv2c　：1
コミュニティ	認証用のコミュニティ名
PDU	PDU タイプによって異なる。

12-1-4　MIB の仕様

内蔵 EtherNet/IP ポートがサポートしている MIB の仕様を説明します。

MIB 体系図

内蔵 EtherNet/IP ポートの 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 " + CPU 形名 + CPU バージョン • CPU 形名：（例）NY512-1300 • CPU バージョン：（例）Ver1.12
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。	○	1.3.6.1.4.1.16838.1.1025.6
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。	○	CPU 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	○	3
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 ～ 3
ifDescr	[(2) RO] インタフェースに関する情報（メーカ名、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）	○	1. Loop Back 2. 10/100/1000M Gigabit Ethernet Port 3. Internal Network 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/ 100000000/ 1000000000
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

サブツリー名	規格［(識別子) 属性］	サポート	実装仕様
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
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

サブツリー名			規格［(識別子) 属性］	サポート	実装仕様
		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)	○	• forwarding(1) • not-forwarding(2) 設定に依存
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
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

サブツリー名	規格［(識別子) 属性］	サポート	実装仕様
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
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。	--	--

サブツリー名			規格［(識別子) 属性］	サポート	実装仕様
	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
		ipRouteNext Hop	[(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) ciscolgrp(11) bbnSpflgp(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）	○	0 固定
		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
tcpConnLocal Address	[(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
tcpConnLocal Port	[(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
tcpConnRem Address	[(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
tcpConnRem Port	[(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

12-2 SNMP(エージェント)機能の使用手順

12-2-1 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 にて［コントローラ設定］|［内蔵 EtherNet/IP ポート設定］を選択します。

［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。内蔵 EtherNet/IP ポート設定がコントローラに転送されます。

12-2-2 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 の内蔵 EtherNet/IP ポート設定を行います。

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.5-8
	ポート No.		任意 ＊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.5-9
		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.5-10
	ポート No.		任意 ＊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.5-11
		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、「5-4 ［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、「5-5［SNMP トラップ設定］ダイアログ」を参照してください。

13

通信性能と通信負荷

13-1 通信のしくみ 13-2
13-1-1 タグデータリンクの通信のしかた 13-2
13-1-2 コネクション数の計算方法 13-4
13-1-3 パケットインターバル（RPI）の精度 13-5

13-2 通信負荷の調整 13-6
13-2-1 タグデータリンクのデバイス帯域使用状況の確認 13-7
13-2-2 タグデータリンクの帯域使用率と RPI の関係 13-9
13-2-3 デバイス帯域使用状況の調整方法 13-10
13-2-4 RPI の変更方法 13-11
13-2-5 パケットインターバル（RPI）の設定例 13-16

13-3 タグデータリンクによる入出力応答時間 13-21
13-3-1 データ転送のタイミング 13-21
13-3-2 内蔵 EtherNet/IP ポートのデータ処理時間 13-22
13-3-3 タスク周期とパケットインターバル（RPI）の関係 13-23
13-3-4 タグデータリンク入出力最大応答時間 13-25

13-4 メッセージサービス機能伝送遅延時間 13-27

13-1 通信のしくみ

13-1-1 タグデータリンクの通信のしかた

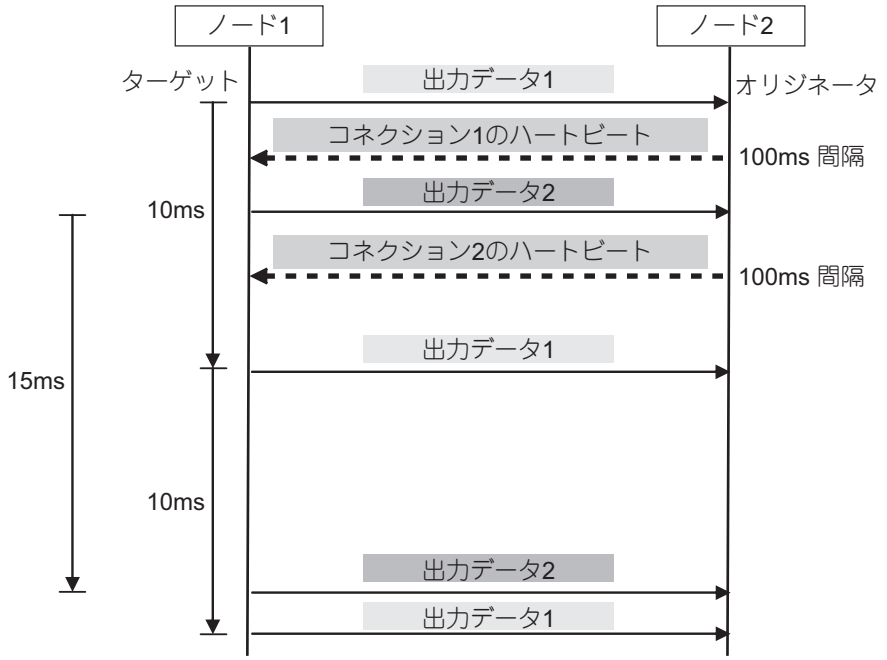
パケットインターバル（RPI）の設定

内蔵 EtherNet/IP ポー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（PPS）の関係

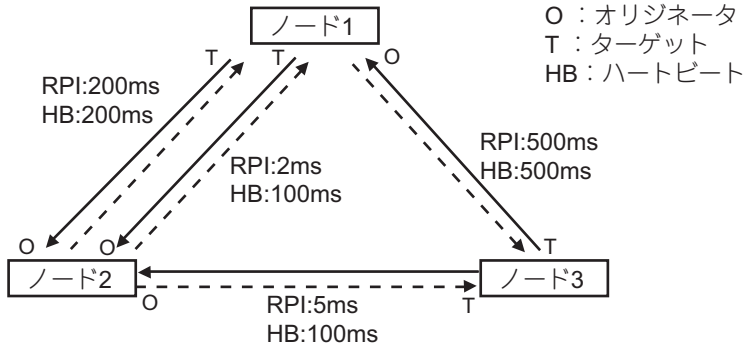
タグデータリンクにて、1 秒間に送受信するパケット数を、「使用通信帯域幅」（以下 PPS（packet per second））と呼びます。
PPS は、コネクションごとに、RPI とハートビートから、以下のように算出されます。

コネクションで使用する PPS 値 [pps]
＝（1000 ÷RPI [ms]）+（1000 ÷ ハートビート送信周期 [ms] ）

また、各内蔵 EtherNet/IP ポート（以下ユニット）にて、1 秒間に送受信するパケットの総数は、以下のように算出されます。
内蔵 EtherNet/IP ポートの総 PPS ＝ オリジネータコネクションの総 PPS
＋ ターゲットコネクションの総 PPS（*）
* 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、内蔵 EtherNet/IP ポートが 1 秒間に送受信可能な最大パケット数（つまり、「ユニット許容通信帯域」）は、以下です。したがって、コネクションの設定は、この範囲で行ってください。
- NY シリーズコントローラ：20000pps

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 と 2ms、受信用 RPI 500ms。
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 と 2ms と 5ms。
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 と 1ms。



各ノードでの総 PPS は、以下のように算出されます。
- ノード 1 のユニットの総 PPS
  ＝ 1000 / 200 ms ＋ 1000 / 2 ms ＋ 1000 / 500 ms　（データ用）
  ＋ 1000 / 200 ms ＋ 1000 / 100 ms ＋ 1000 / 500 ms（ハートビート用）
  ＝ 524 pps
- ノード 2 のユニットの総 PPS
  ＝ 1000 / 200ms ＋ 1000 / 2 ms ＋ 1000 / 5 ms　（データ用）
  ＋ 1000 / 200ms ＋ 1000 / 100 ms ＋ 1000 / 100 ms（ハートビート用）
  ＝ 730 pps
- ノード 3 のユニットの総 PPS
  ＝ 1000 / 5 ms ＋ 1000 / 500 ms　（データ用）
  ＋ 1000 / 100 ms ＋ 1000 / 500 ms　（ハートビート用）
  ＝ 214pps

いずれも、ユニット許容通信帯域の使用範囲内のため、送受信可能です。

13-1-2　コネクション数の計算方法

内蔵 EtherNet/IP のコネクション数は最大以下です。

- NY シリーズコントローラ：128 本

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設定可能なコネクション最大数以内で設定する必要があります。

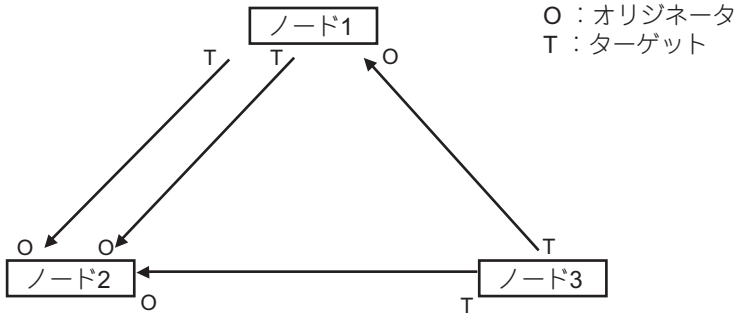
例）

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、内蔵 EtherNet/IP で設定可能なコネクション最大数以内のため、コネクションの開設が可能です。

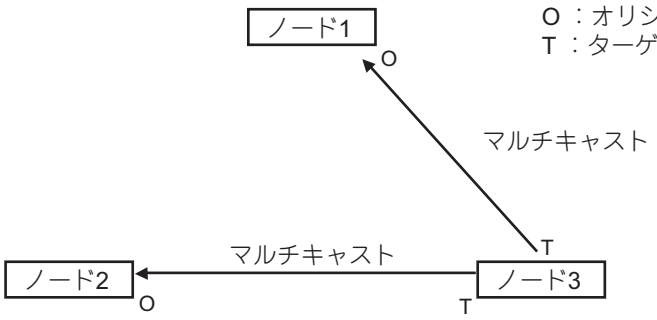


また、マルチキャストに設定した場合、送信するパケットは 1 つになりますが、コネクション数は消費されます。

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。

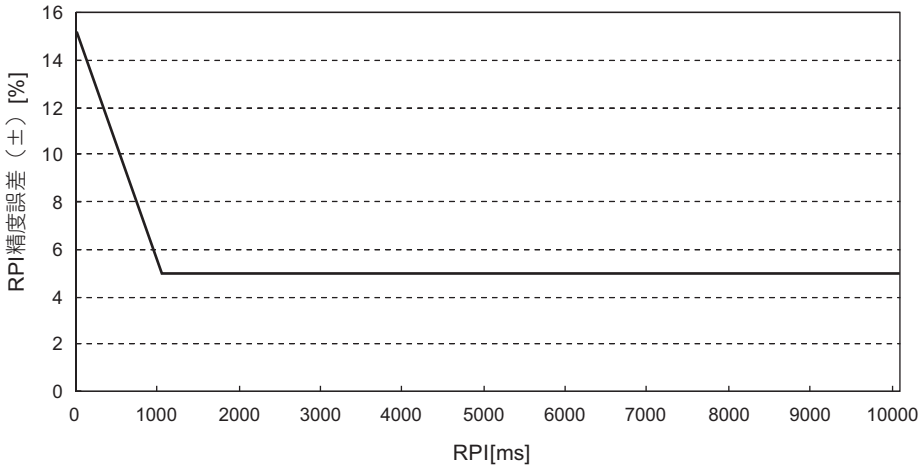
マルチキャストに設定してもユニキャストとコネクション消費数は変わらないので注意が必要です。



13-1-3 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、内蔵 EtherNet/IP ポー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（±）[%]
1ms ～ 1000ms	15 －（RPI[ms] ／ 100）
1000ms ～ 10000ms	RPI の 5%



13-2 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

内蔵 EtherNet/IP ポートでは、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	NY シリーズコントローラ
ネットワーク帯域幅	Ethernet の物理的な伝送速度	1000Mbps
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）	最大 20000pps
コネクションリソース	開設可能なコネクション数	最大 128
パケットインターバル（RPI：Requested Packet Interval）	タグデータの更新周期	1 ～ 10000ms（1ms 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。

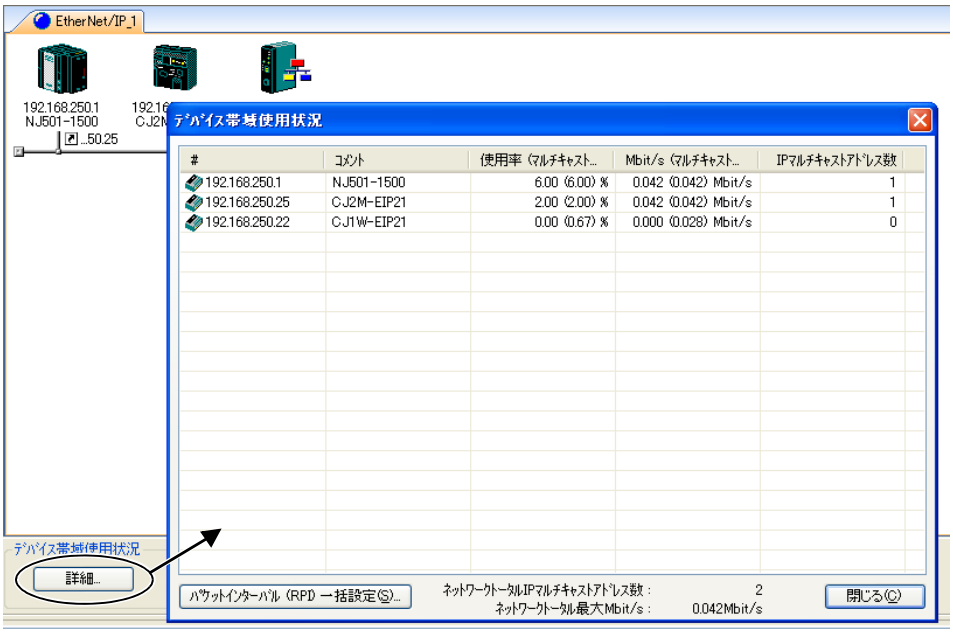
参考

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。

13-2-1　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、個々の内蔵 EtherNet/IP ポー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）	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。 設定可能な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

内蔵 EtherNet/IP ポートごとの、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、「13-2-4 RPI の変更方法 (P.13-11)」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、「13-2-4 RPI の変更方法 (P.13-11)」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 1000Mbit/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、「13-2-4 RPI の変更方法 (P.13-11)」に従い調整が可能です。

13-2-2　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。

13-2-3　デバイス帯域使用状況の調整方法

タグデータリンクのデバイス帯域使用状況を調整する方法を記載します。

使用上の注意

スイッチングハブが、コントローラに設定できるネットワーク帯域幅の最大値に対応していることとします。NY シリーズコントローラのネットワーク帯域幅の最大値は 1000Mbps です。

● 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

- 全ノードの「Mbit/s（マルチキャストフィルタなし）」が、設定できるネットワーク帯域幅以下か？
  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、設定できるネットワーク帯域幅以下か？
  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

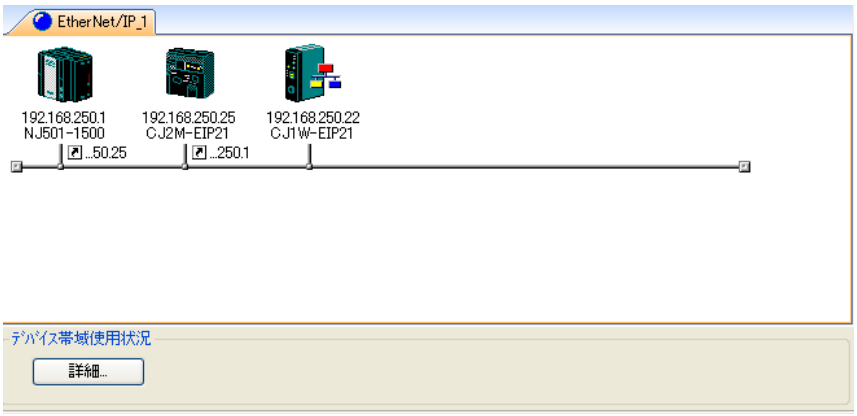
- 全ノードの「Mbit/s」が、設定できるネットワーク帯域幅以下か？
  1 つでも超える場合：RPI 変更などコネクション設定の見直しを行ってください。
- 全ノードの「使用率」が、100% 以下か？
  1 つでも超える場合：RPI 変更などコネクション設定の見直しを行ってください。
- 「ネットワークトータル最大 Mbit/s」が、設定できるネットワーク帯域幅以下か？
  超える場合：ネットワーク配線（スイッチングハブをカスケード接続するなど）によっては一部の伝送路（スイッチングハブやメディアコンバータ）で帯域オーバーが発生しタグデータリンクが正常に動作しない可能性があります。各カスケ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、設定できるネットワーク帯域幅以下か？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。

13-2-4　RPI の変更方法

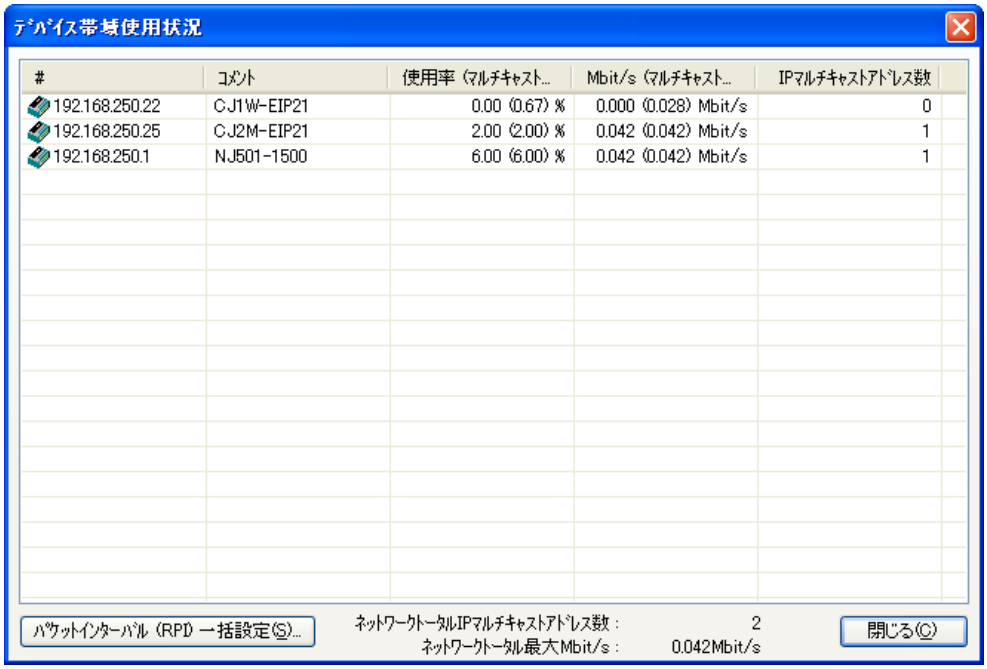
「13-2-1 タグデータリンクのデバイス帯域使用状況の確認 (P.13-7)」に従い、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし)」をオフラインで確認することができます。
タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし)」は、パケットインターバル（RPI）設定を変更することで調整が可能です。
ただし、調整の結果、必要な通信性能を確保できない場合、ネットワーク構成から再検討をしてください。

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。

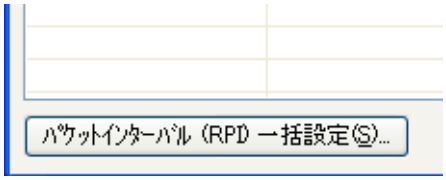


タグデータリンク許容通信帯域幅に対する使用率が「使用率（マルチキャストフィルタなし)」列に、ネットワーク帯域（伝送速度）での使用量が「Mbit/s（マルチキャストフィルタなし)」列に、表示されます。

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。

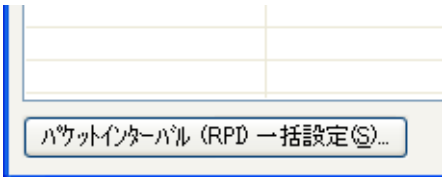


(2) ［パケットインターバル（RPI）一括設定］ウィンドウが表示されます。新たなパケットインターバル（RPI）値を入力し、［OK］ボタンをクリックしてください。



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。

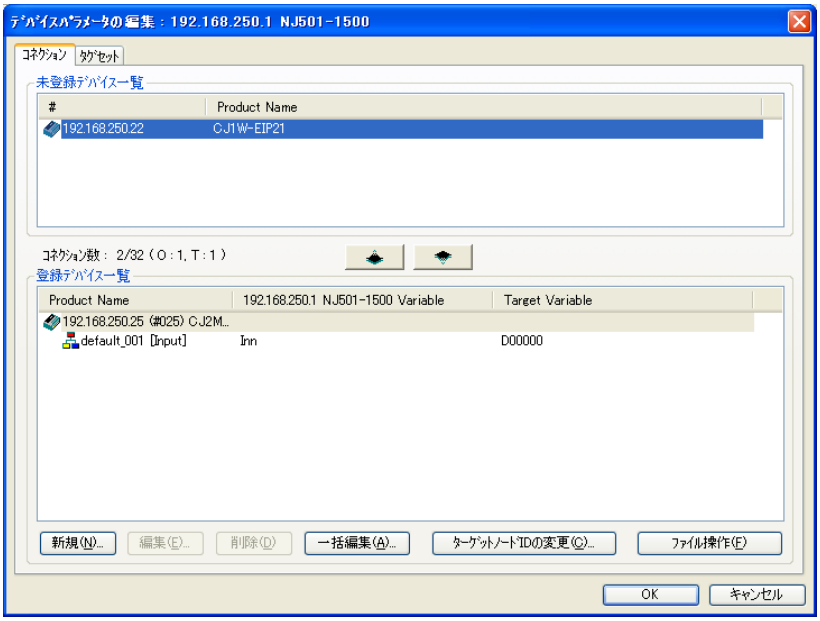


(3) 新たなパケットインターバル（RPI）値を入力し、［OK］ボタンをクリックしてください。

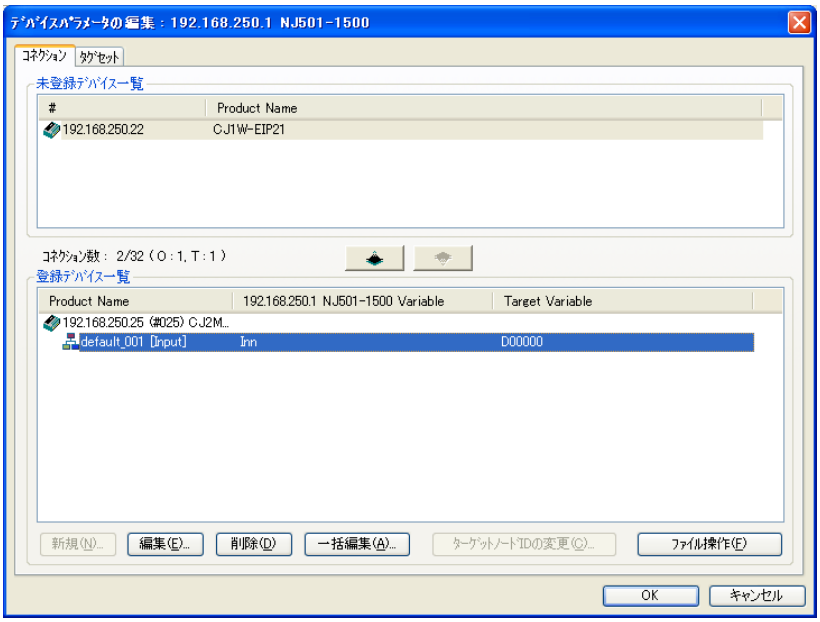
方法３：特定コネクションのパケットインターバル（RPI）設定を変更する場合：
特定コネクションのパケットインターバル（RPI）設定を個別に変更することでタグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」を調整することが可能です。
この場合、パケットインターバル（RPI）を変更したコネクションのターゲットデバイスでも、タグデータリンク許容通信帯域幅に対する「使用率（マルチキャストフィルタなし）」が変化することにご注意ください。

(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。

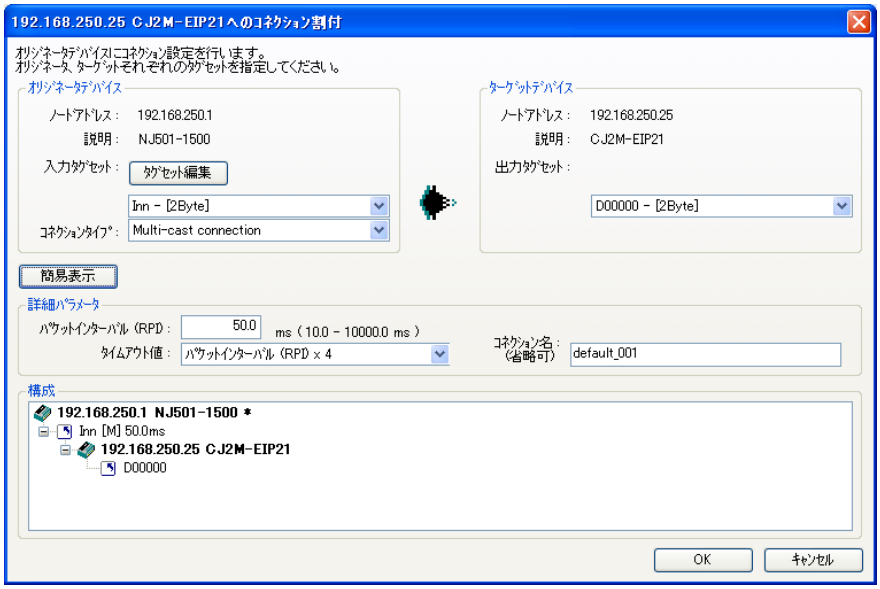
(2) 特定コネクションのオリジネータ設定を持つデバイスをダブルクリックします。［デバイスパラメータの編集］ウィンドウが表示されます。



(3) ［登録デバイス一覧］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。（📖「13-2-3 デバイス帯域使用状況の調整方法 (P.13-10)」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**

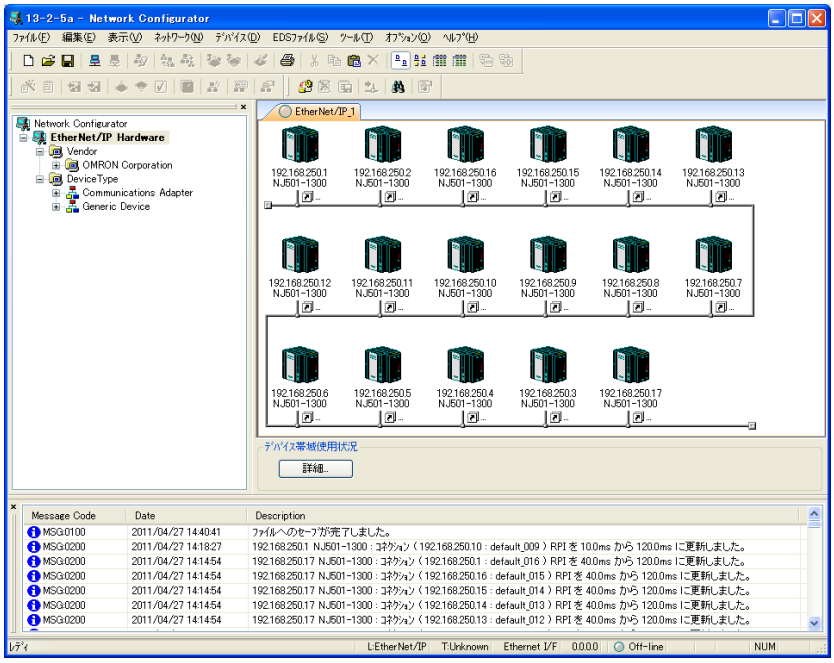
コネクション設定を変更した場合は、ネットワーク構成ウィンドウ下部の「デバイス帯域使用状況」内の［詳細］ボタンをクリックし、📖「13-2-1 タグデータリンクのデバイス帯域使用状況の確認 (P.13-7)」に従いチェックを行ってください。
とくに、パケットインターバル（RPI）一括設定を使用せずにコネクションごとに調整した場合、この確認が重要となります。

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。**

13-2-5　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



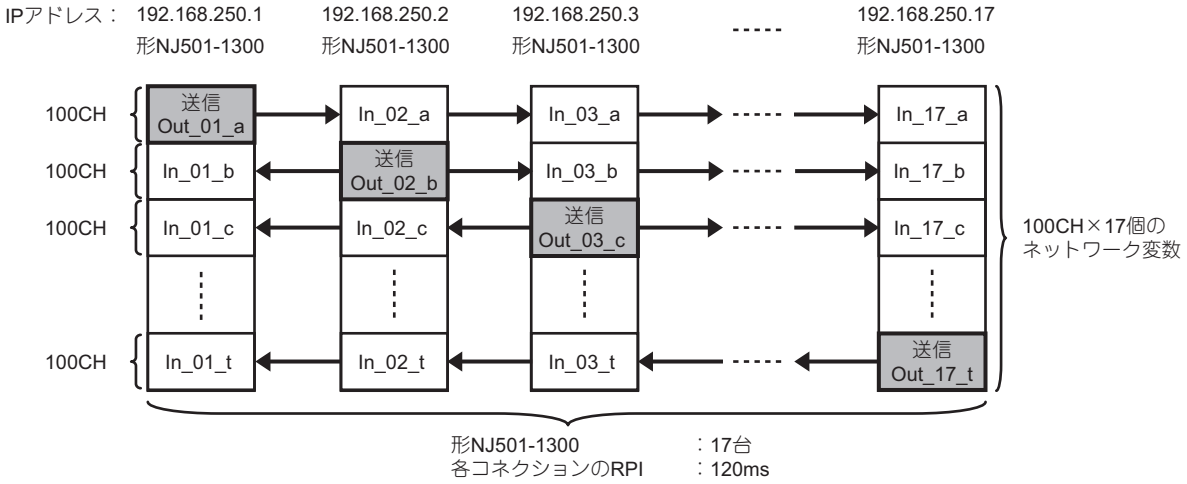
- コネクション：

  例）17 台の形 NJ501-1300 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 16 個持ち、相互にデータ交換を行っています。

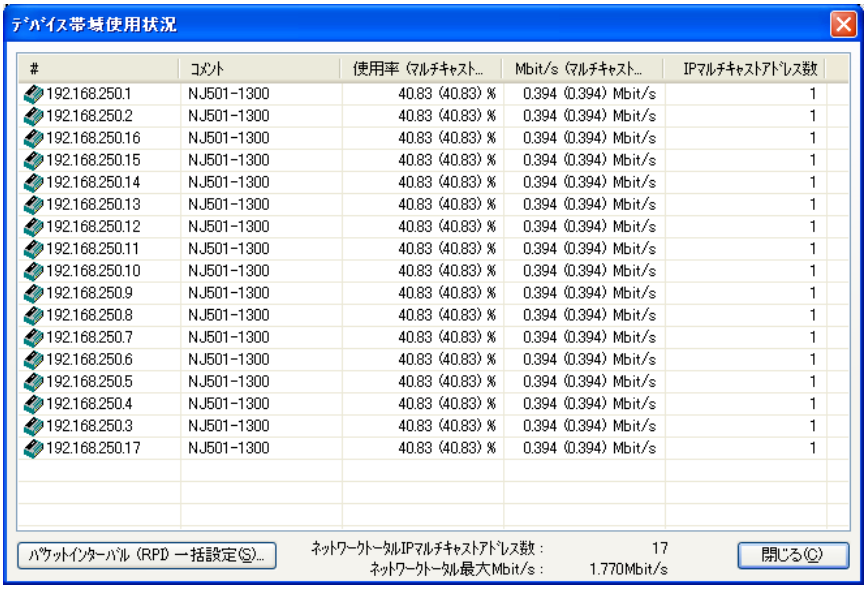
  すべてのコネクションのパケットインターバル（RPI）は、120ms に設定します。

  なお、各デバイスの IP アドレスは、192.168.250.1 ～ 192.168.250.17 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 40.83%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

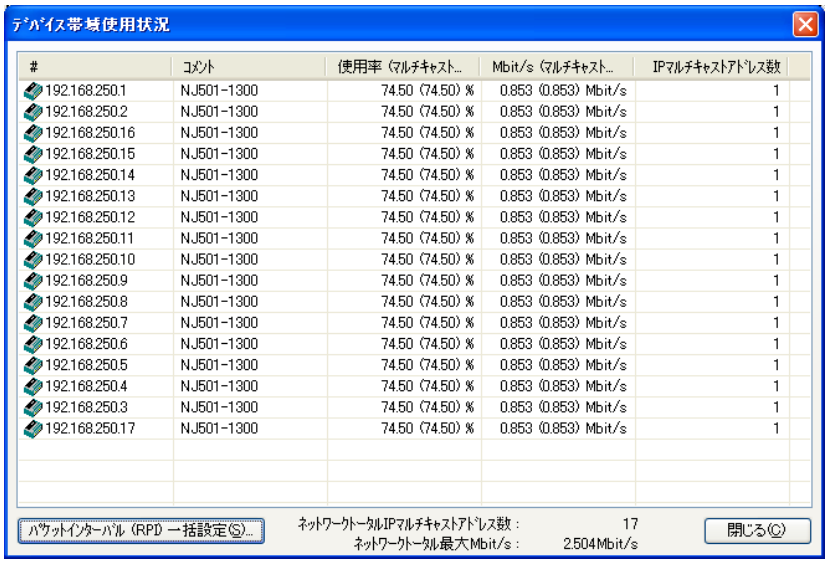
すべてのコネクションのパケットインターバル（RPI）が 120ms の設定でタグデータリンク許容通信帯域幅に対する使用率が 40.83%でしたが、ここでは 80% 以内の許容帯域を目標に、パケットインターバル（RPI）を 40ms に変更します。

［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 40ms 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 74.50% になり、40ms 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

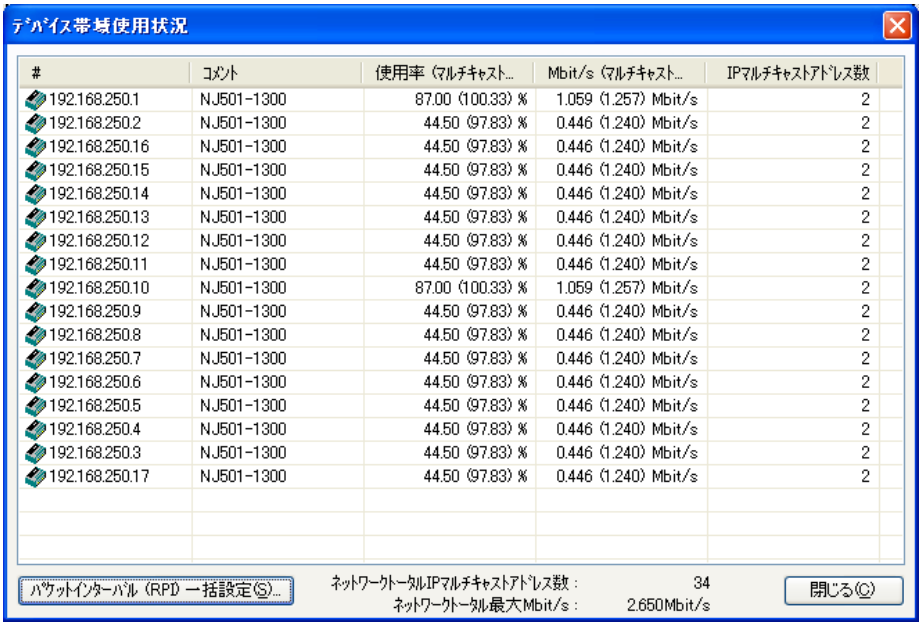
デバイス 192.168.250.1 と 192.168.250.10 だけ、高速にタグデータリンクを行いたいとします。この場合［デバイス帯域使用状況］ウィンドウの［パケットインターバル（RPI）一括設定］ボタンをクリックし、［一括設定］ウィンドウを表示します。

［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 30ms 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 87.00% 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。

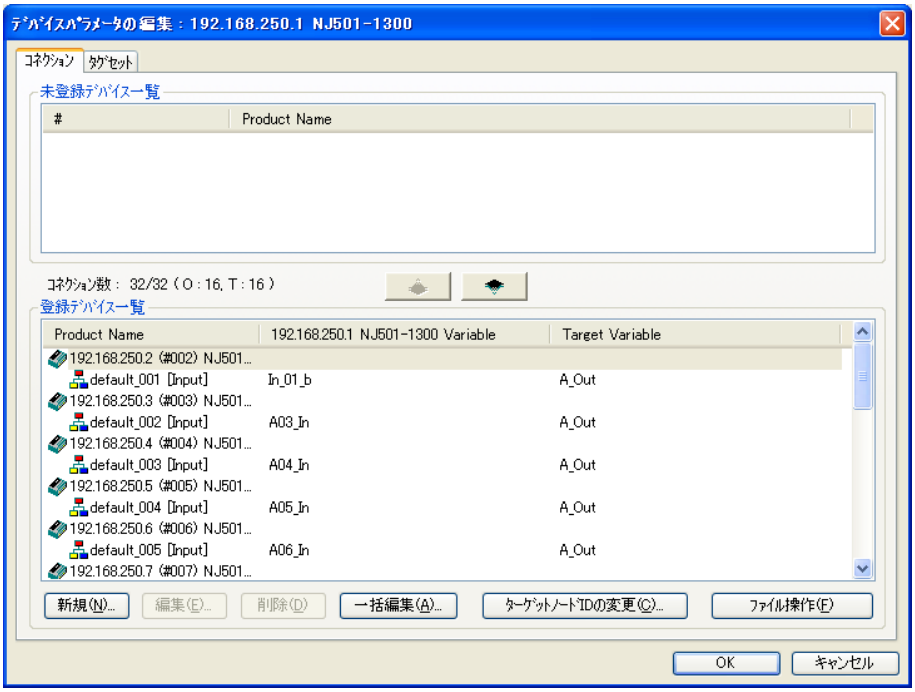
なお、この場合、デバイス 192.168.250.1 と 192.168.250.10 とコネクションを持つデバイスのタグデータリンク許容通信帯域幅に対する使用率も増えている（40.83%→ 44.67%）ことに注意してください。



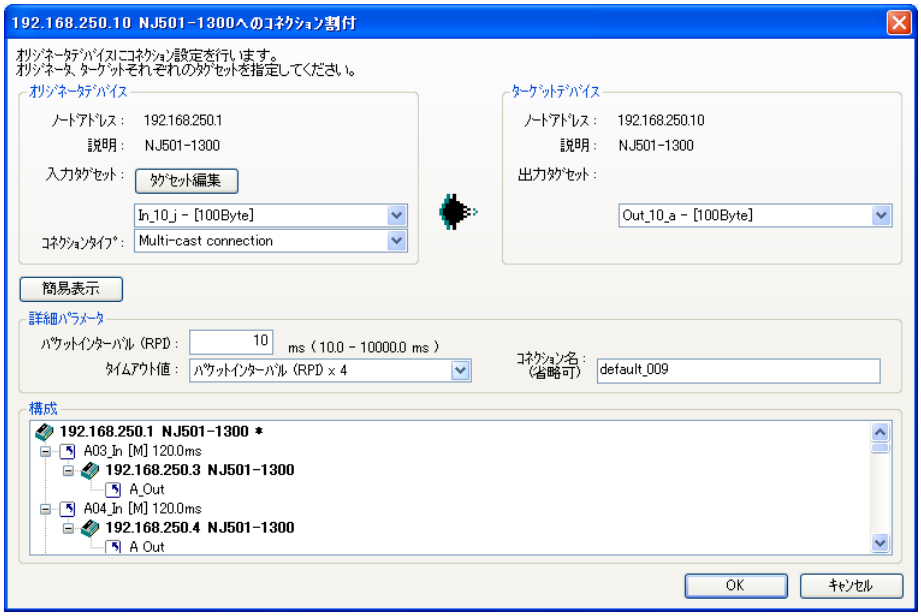
このケースでは、マルチキャストフィルタなしの場合に 100.33% となり、マルチキャストフィルタのないスイッチングハブの場合は、内蔵 EtherNet/IP ポートで通信負荷による通信障害が発生する可能性があります。

＜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

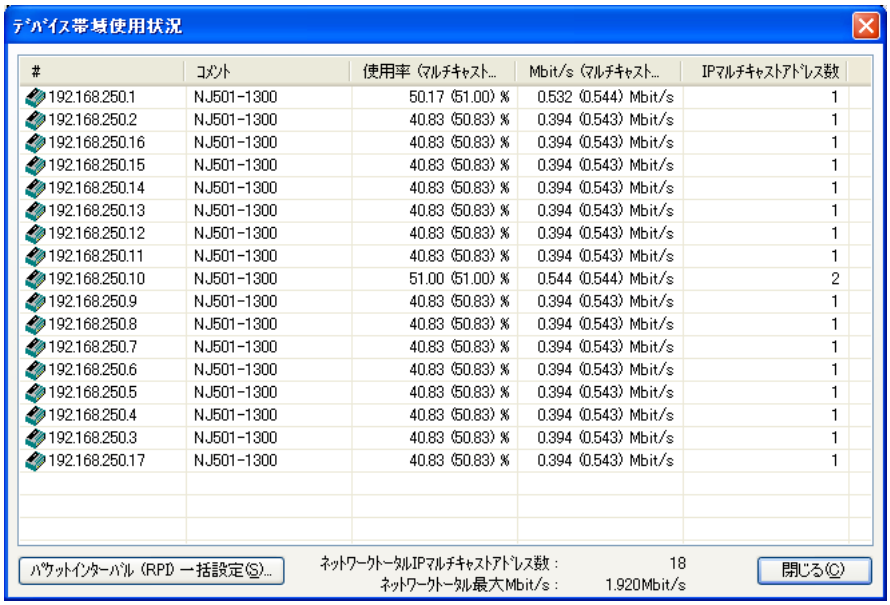
デバイス 192.168.250.1 の特定コネクションだけ、高速にタグデータリンクを行いたいとします。この場合、ネットワーク構成ウィンドウでデバイス 192.168.250.1 をダブルクリックします。



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 10ms 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 50.17% 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（40.83%→51.00%）ことに注意してください。

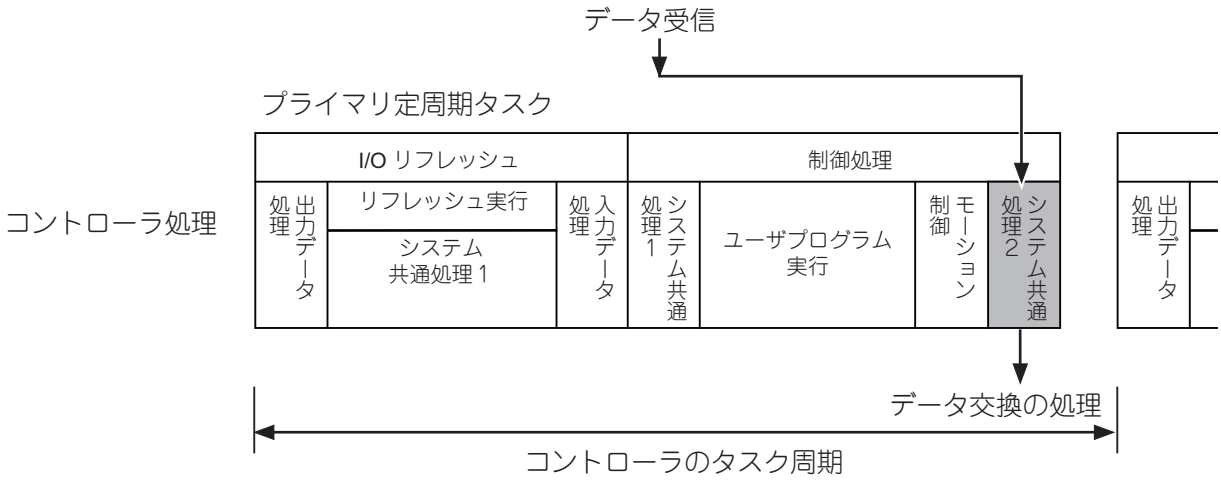
13-3 タグデータリンクによる入出力応答時間

参考

ここでは、NY シリーズコントローラの内蔵 EtherNet/IP ポートについて説明します。
EtherNet/IP ユニットおよび NJ/NX シリーズコントローラの内蔵 EhterNet/IP ポート、CJ2H CPU ユニット（形 CJ2H-CPU6□-EIP）の内蔵 EhterNet/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、コントローラとのデータ処理の性能が異なります。詳細は 「7-4 他機種とのタグデータリンク (P.7-82)」を参照してください。

13-3-1 データ転送のタイミング

内蔵 EtherNet/IP ポートとコントローラ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「システム共通処理 2」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

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。

13-3-2　内蔵 EtherNet/IP ポートのデータ処理時間

内蔵 EtherNet/IP ポートが、コントローラ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①～③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。

コントローラの形式	定数の値 (μs)			
	a	b	c	d
NY5□2-□□□□	0.0006	0.100	1.40	6.60

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」(*2) 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 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。

使用上の注意

内蔵 EtherNet/IP ポートの送受信できるタグデータリンクチャネル数は、最大 92416CH です。タグデータリンクチャネル数が、コントローラとの 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、更新タスクのタスク周期と、②で求めた「データ転送回数」によって計算できます。

タスク周期 × データ転送回数

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コントローラの形式
  形 NY5□2-□□□□
- 更新タスク
  プライマリ定周期タスク
  タスク周期：500μs（変数アクセス時間 2%）
- 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。**
[(600+200+1000) バイト ×0.0006μs]+[(8+4+10) 個 ×0.100μs]+3×1.40μs
+6.60μs = 11.28μs

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」= 11.28μs
[タスクに設定した変数アクセス時間]	500μs×0.02 = 10μs
[データ転送回数]	11.28μs÷10μs = 1.128 回

データ転送回数は、約 2 回分必要となります。

3 **実際のデータ転送に要する時間を計算します。**
500μs×2 回= 1000μs

13-3-3　タスク周期とパケットインターバル（RPI）の関係

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、タグの更新タスクに設定されたタスクは、タスク処理にタグデータのデータ転送処理が加わるため、タスク設定の「変数アクセス時間」と「タスク周期」の時間を、1 回のタスク周期で処理できるように、調整する必要がでてきます。

1　データ転送に必要な時間を計算し、「変数アクセス時間」(*) とします。

データ転送に必要な時間の計算式は、📖「 データ処理時間の概要（P.13-22)」を参照してください。

* 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、📖『NY シリーズ 産業用パネル型 PC ／産業用ボックス型 PC ユーザーズマニュアル ソフトウェア編 (SBCA-436)』を参照してください。

タスク周期によるパケットインターバル（RPI）の調整

パケットインターバル（RPI）設定に関わらず、タグデータの転送は、実際のデータ転送で要する時間（タスク周期 × データ転送回数）で実行します。

よって、パケットインターバル（RPI）設定は、以下を目安に設定してください。

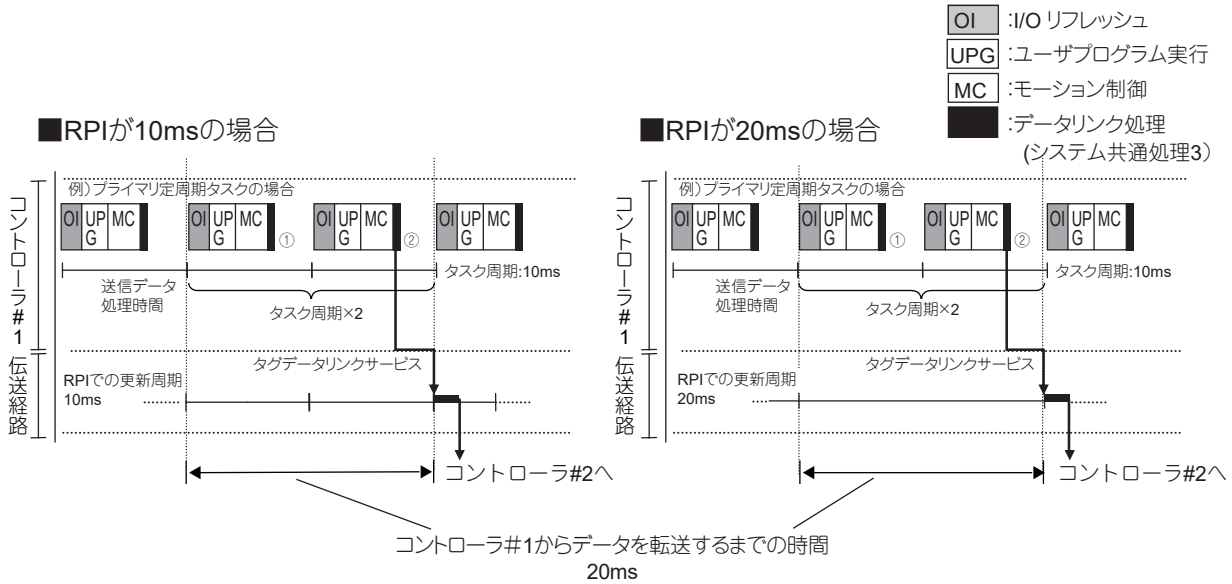
実際のデータ転送で要する時間（タスク周期 × データ転送回数）< RPI

「実際のデータ転送で要する時間」については、📖「13-3-2 内蔵 EtherNet/IP ポートのデータ処理時間 (P.13-22)」を参照してください。

例） RPI 設定とデータ転送される時間の関係

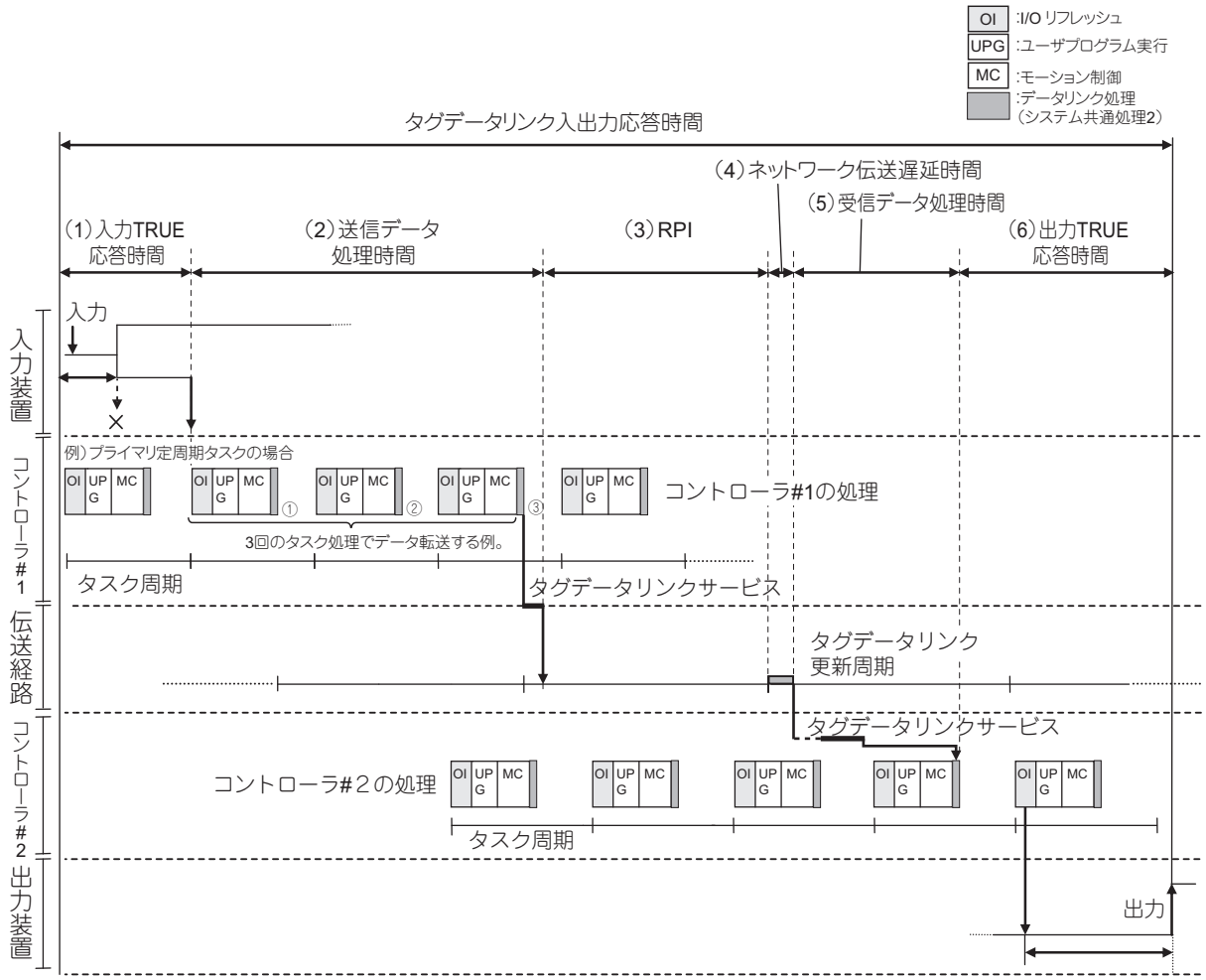
- タスク周期：10ms
- データ転送回数：2 回
- 実際のデータ転送で要する時間：10ms×2 回＝ 20ms

RPI の値に関わらず、データ転送される時間は 20ms となります。



13-3-4　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。



(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、その入力データがコントローラ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、コントローラのメモリエリアに反映されるまで、コントローラのタスク周期 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 + コントローラのタスク周期

(2) 送信データ処理時間

コントローラ上の変数を内蔵 EtherNet/IP ポートに転送するまでの時間です。
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、🕮「13-3-2 内蔵 EtherNet/IP ポートのデータ処理時間 (P.13-22)」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

内蔵 EtherNet/IP ポートに受信したデータをコントローラ上の変数に転送する時間です。
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、🕮「13-3-2 内蔵 EtherNet/IP ポートのデータ処理時間 (P.13-22)」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

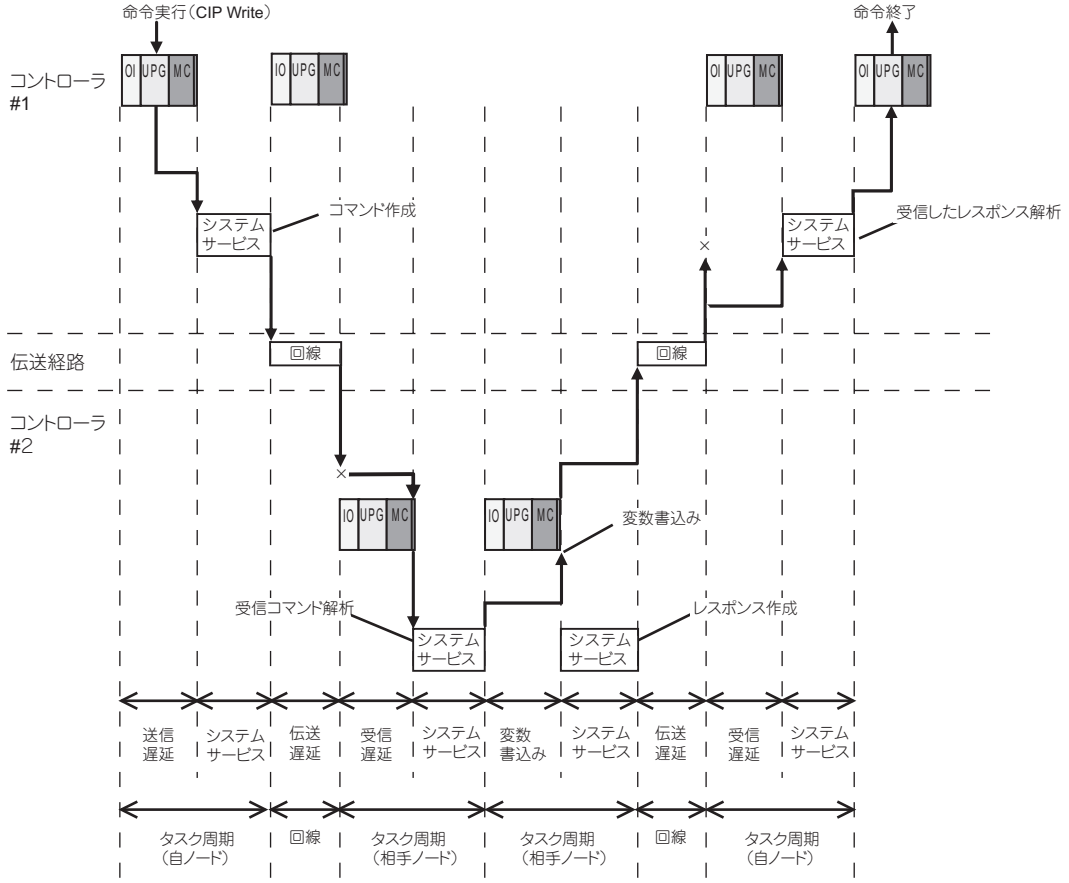
コントローラから出力が指示されてから、実際に出力が行われるまでの外部出力機器の遅延時間です。
出力 TRUE 応答時間＝出力装置遅延時間 + コントローラのタスク周期

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。

13-4 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、または内蔵 EtherNet/IP ポート自身の負荷（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。

14

ネットワークの通信ステータス確認と処置方法

EtherNet/IP ネットワークの通信ステータスを確認する Network Configurator の機能説明と、タグデータリンクのコネクションステータスによる異常の切り分けと処置方法を説明します。

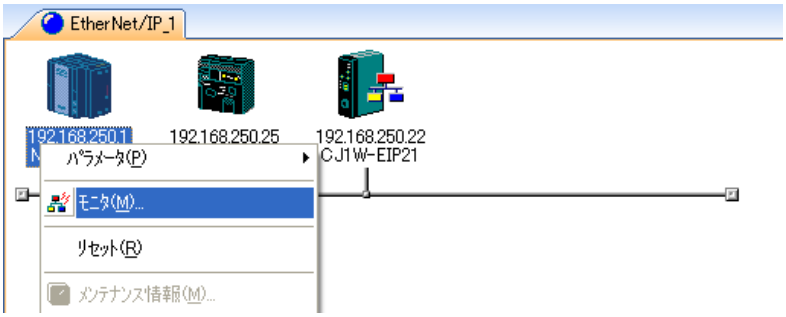
14-1 Network Configurator のデバイスモニタ機能 14-2
14-1　Network Configurator のデバイスモニタ機能 14-2
14-2　コネクションステータスのコード一覧と処置方法 14-9
14-2 コネクションステータスのコード一覧と処置方法 14-9

14-1　Network Configurator のデバイスモニタ機能

Network Configurator のデバイスモニタ機能の起動方法と表示する各種ステータス内容を説明をします。

14-1-1　デバイスモニタ機能の起動方法

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。

14-1-2　デバイスモニタ機能の各種ステータス表示

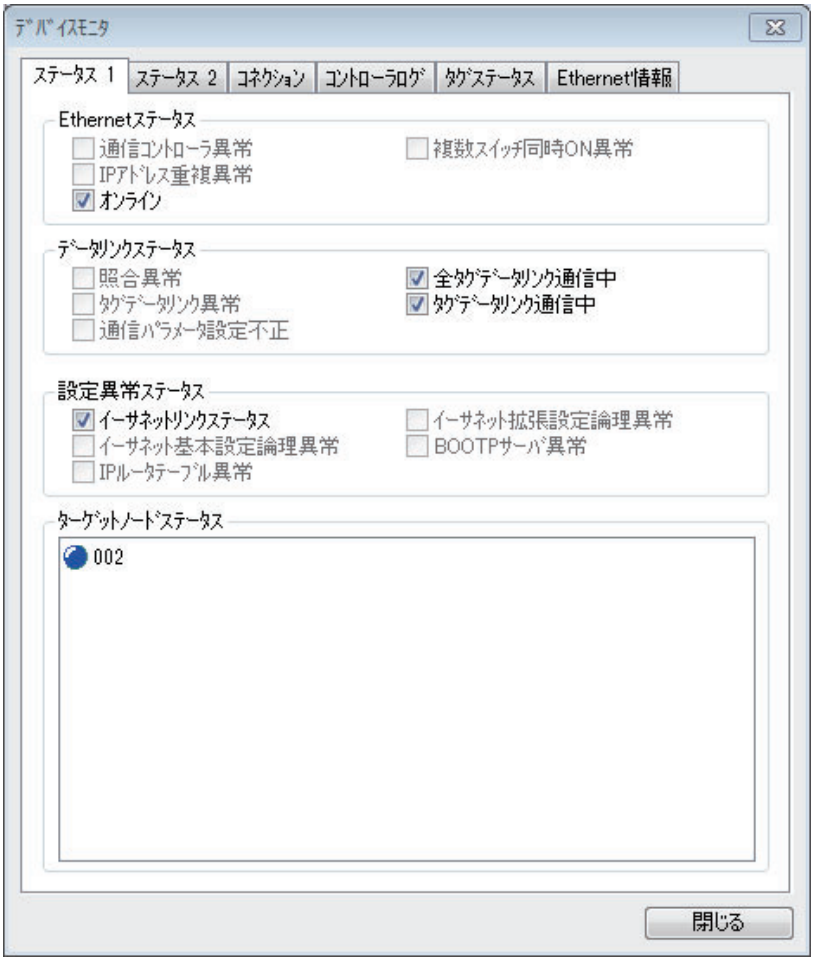
●［ステータス 1］タブ

次の各ステータス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。

分類	項目	対応するシステム定義変数
Ethernet ステータス	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たことを示す
	IP アドレス重複異常	IP アドレスの重複が発生したことを示す
	オンライン	オンライン状態（EtherNet/IP の通信処理が動作可能）であることを示す
	複数スイッチ同時 ON 異常	複数のデータリンク開始／データリンク停止スイッチを同時に TRUE したことを示す
データリンクステータス	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ることを示す 主な要因　• 指定されたターゲットが存在しない • 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

分類	項目	対応するシステム定義変数
データリンクステータス	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 つ以上の異常があることを示す
	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た場合
	全タグデータリンク通信中	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す
	タグデータリンク通信中	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す
設定異常ステータス	イーサネットリンクステータス	スイッチングハブとの間でリンクが確立しているとき TRUE
	イーサネット基本設定論理異常	以下の設定が不正のとき TRUE • TCP/IP 設定（IP アドレス／サブネットマスク／ LINK 設定）
	IP ルータテーブル異常	IP ルータテーブルの情報が不正のとき TRUE
	イーサネット拡張設定論理異常	0 固定（TRUE することはありません）
	BOOTP サーバ異常	Bootp サーバの使用で以下のエラーのとき TRUE • サーバから受信した IP アドレスが不正 • サーバーとの通信でタイムアウトが発生

正常ターゲットノードのうち、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。



● [ステータス 2] タブ

システム定義変数の「_EIP_EstbTargetSta」（正常ターゲットノード情報）のうち、「_EIP_TargetPLCModeSta」（ターゲット PLC 動作モード）と「_EIP_TargetPLCErr」（ターゲット PLC 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。正常の場合には青色、異常の場合は赤色で表示されます。

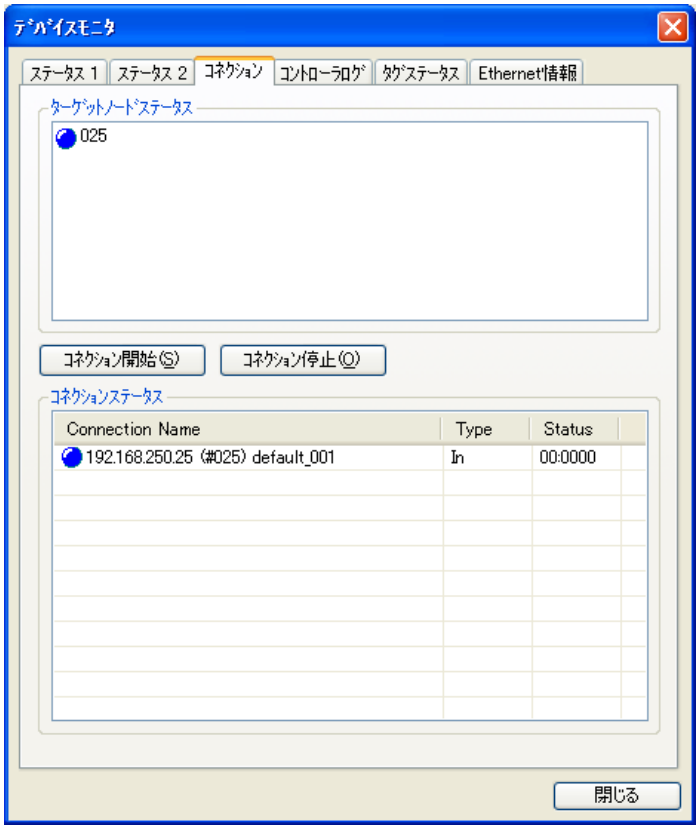


参考

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

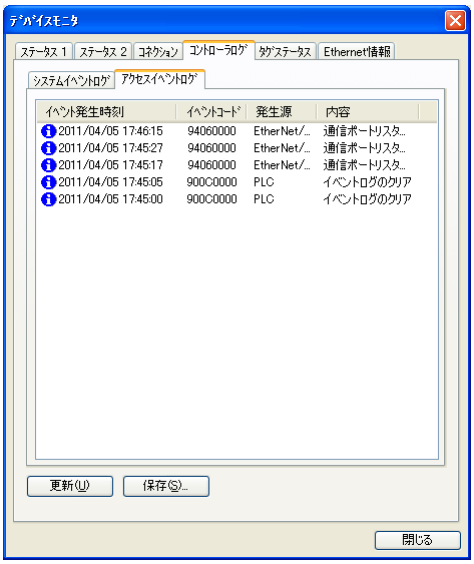
正常ターゲットノード情報のうち、本ノードがオリジネータとなっているターゲットノードの情報を表示します。
本ノードとのタグデータリンクコネクションが全て正常に確立している場合は青色で、１つでも未確立のコネクションがある場合は赤色で表示されます。
ただし、本ノードのコネクションを停止した場合は灰色で表示されます。
また、コネクションステータスでは、オリジネータ設定されている各コネクションの現在状態が［Status］で表示されます。この情報をもとにして、タグデータリンクの異常要因を切り分けることができます。
コネクションステータス情報の詳細は、📖「14-2 コネクションステータスのコード一覧と処置方法 (P.14-9)」を参照してください。



●［コントローラログ］タブ

コントローラが保持しているコントローラのイベント履歴を表示します。
過去に発生したイベントが記録されており、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、コントローラのユーザーズマニュアルを参照してください。





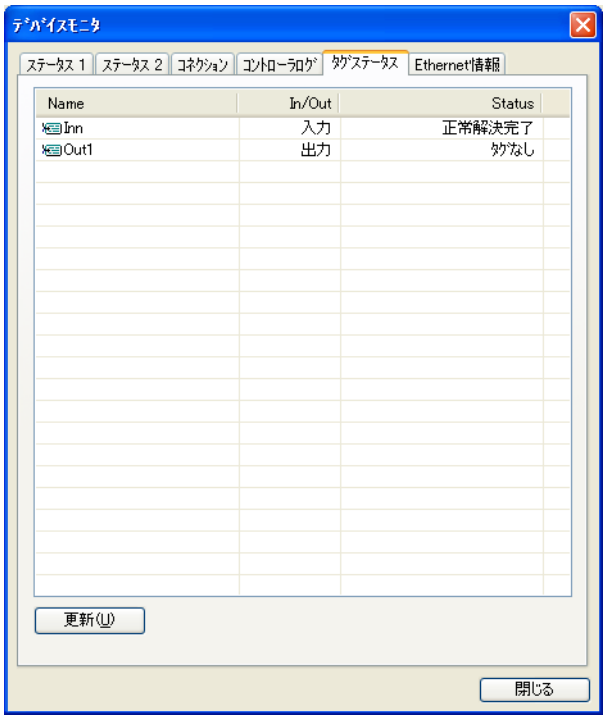
●［タグステータス］タブ

タグデータリンクの各タグが、コントローラ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コントローラの変数テーブルに設定されていません。
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：
1. Const 属性のため書込みできません。
2. タグデータリンク設定の入出力方向と、コントローラの変数の入出力方向が一致していません。
コントローラの変数の「ネットワーク公開属性」設定に誤りがあります。

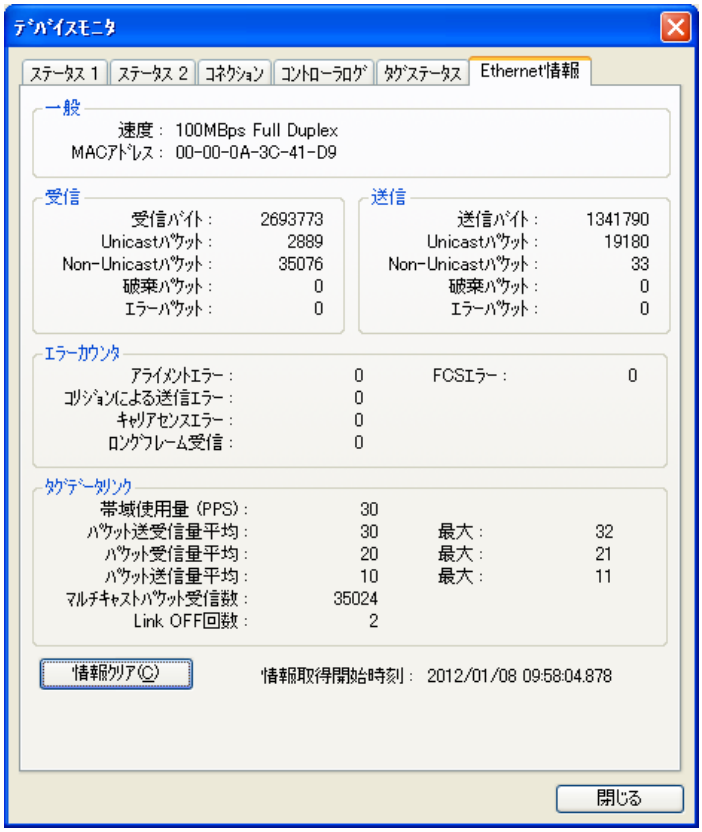
この異常が発生しているタグではコネクションは開設されません。

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 NY シリーズコントローラ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

内蔵 EtherNet/IP 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。
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。



14-2 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、「14-1 Network Configurator のデバイスモニタ機能 (P.14-2)」を参照してください。

参考

コネクションステータスは、CIP 仕様書に定義されている Connection Manager のエラーレスポンスコード General ／ Additional を意味します。

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、形 CJ1W-EIP21、 形 CJ2H-CPU□□-EIP、形 CJ2M-CPU3□、 形 NJ501-□□□□、形 NJ301-□□□□、 形 NJ101-□□□□、形 NX701-□□□□、 形 NY5□2-□□□□	形 CS1W-EIP21、形 CJ1W-EIP21、 形 CJ2H-CPU□□-EIP、形 CJ2M-CPU3□、 形 NJ501-□□□□、形 NJ301-□□□□、 形 NJ101-□□□□、形 NX701-□□□□、 形 NY5□2-□□□□
構成 2	形 CS1W-EIP21、形 CJ1W-EIP21、 形 CJ2H-CPU□□-EIP、形 CJ2M-CPU3□、 形 NJ501-□□□□、形 NJ301-□□□□、 形 NJ101-□□□□、形 NX701-□□□□、 形 NY5□2-□□□□	他社製
構成 3	他社製	形 CS1W-EIP21、形 CJ1W-EIP21、 形 CJ2H-CPU□□-EIP、形 CJ2M-CPU3 □、 形 NJ501-□□□□、形 NJ301-□□□□、 形 NJ101-□□□□、形 NX701-□□□□、 形 NY5□2-□□□□

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。
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。		

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	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。
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（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。

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	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。
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。	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。
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 1ms 以上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	0127	ターゲット側が応答した異常コード： コネクションのデータ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→ターゲットの向きのデータ）	本異常発生しません。	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。（オリジネータ→ターゲットの向きのデータ） 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。	本異常発生しません。
01	0128	ターゲット側が応答した異常コード： コネクションのデータサイズがオリジネータとターゲットの設定で異なる。（ターゲット→オリジネータの向きのデータ）	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。（ターゲット→オリジネータの向きのデータ）		
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。 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。		
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）。

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	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）
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）
01	0813	ターゲット側が応答した異常コード：異なるネットワークからマルチキャストコネクション開設が要求された。	コネクションタイプをマルチキャストからユニキャストに変更してください。もしくは、オリジネータとターゲットが同一ネットワークになるよう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
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 🕮『NY シリーズ トラブルシューティングマニュアル (SBCA-438)』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）

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
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）
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コントローラ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コントローラのタスク周期が長い場合、またはコントローラ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、コントローラ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コントローラ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コントローラのタスク周期が長い場合に発生します。 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）。

A

付録

A-1 他シリーズとの EtherNet/IP ポートの機能比較 A-2

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-3

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-3
A-2-2 Sysmac Studio による EtherNet/IP コネクション設定の手順 A-4
A-2-3 EtherNet/IP コネクション設定機能 A-5
A-2-4 Sysmac Studio による EtherNet/IP コネクション設定 A-10
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-31
A-2-6 トラブルシューティング A-35

A-3 EDS ファイルの管理 A-38

A-3-1 EDS ファイルのインストール A-39
A-3-2 EDS ファイルの作成 A-39
A-3-3 EDS ファイルの削除 A-40
A-3-4 EDS ファイルの保存 A-40
A-3-5 EDS ファイルの検索 A-41
A-3-6 EDS ファイルのプロパティ表示 A-42
A-3-7 EDS インデックスファイルの再作成 A-42

A-4 Network Configurator を、Windows XP/Windows Vista/Windows 7 以降で使用する際の留意点 A-43

A-4-1 Windows ファイアウォールの設定の変更手順 A-43

A-5 変数のメモリ確保の方法 A-46

A-5-1 変数のメモリ確保の規則 A-46
A-5-2 留意するべきケース例 A-55

A-6 コントローラの外部出力にアクセスする場合の留意事項 A-58

A-7 TCP 状態遷移 A-59

A-1　他シリーズとの EtherNet/IP ポートの機能比較

○：サポート有り、×：サポート無し

項　目	NY シリーズ内蔵 EtherNet/IP ポート	NX シリーズ内蔵 EtherNet/IP ポート	NJ シリーズ内蔵 EtherNet/IP ポート	CJ シリーズ Ethernet ユニット	EtherNet/IP ユニット（CJ2 CPU ユニット内蔵ポート）		
					ユニット Ver.1.0	ユニット Ver.2.0	ユニット Ver.2.1
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	○	○	○	×	○	○	○
IP ルーティング機能	○	○	×	×	×	×	×
ソケットサービス	○	○	○	○	×	×	×
FTP サーバ機能	○	○	○	○	×	○	○
FTP クライアント機能	○	○	○	×	×	×	×
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	○*1	○*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	○	○	○	×	○	○	○
Sysmac Studio によるコネクション設定	○	○	○	×	×	×	○

*1. 内蔵 EtherNet/IP ではイベントログを示します。

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

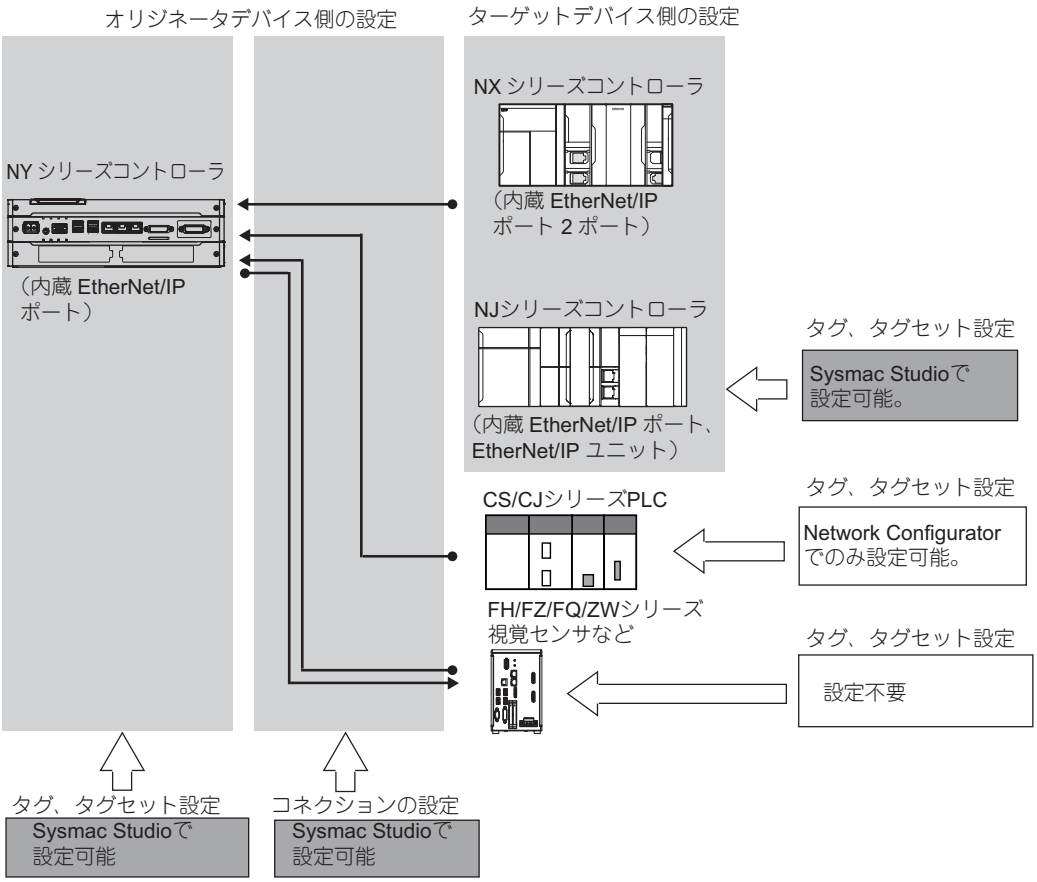
A-2-1 Sysmac Studio によるタグデータリンク（EtherNet/IP コネクション）設定の概要

NY 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。

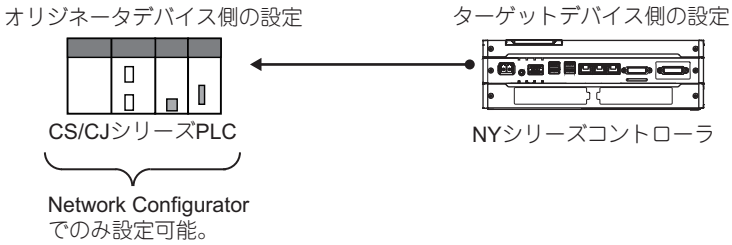
*1. EtherNet/IP ネットワーク上で、コントローラーコントローラ間、またはコントローラー他のデバイス間で、互いのタグをサイクリックにデータ交換する機能をタグデータリンク、もしくは EtherNet/IP コネクションと呼びます。本項では、これらを EtherNet/IP コネクションと表記します。

Sysmac Studio で EtherNet/IP コネクション設定が可能なシステム構成

オリジネータデバイスが NY 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 NY 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

1 デバイスの登録

・メイン画面

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。

		設定画面
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-11)	Sysmac Studio グローバル変数テーブル
3 タグ、タグセットの登録	… 「タグ・タグセットの登録」(P.A-13)	EtherNet/IP コネクション設定（タグセット画面）
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。		
4 コネクションの設定	… 「コネクションの設定（オリジネータデバイス）」(P.A-16)	EtherNet/IP コネクション設定（コネクション画面）
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		
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-26)	・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-26)	・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-31)	EtherNet/IP コネクションモニタ画面

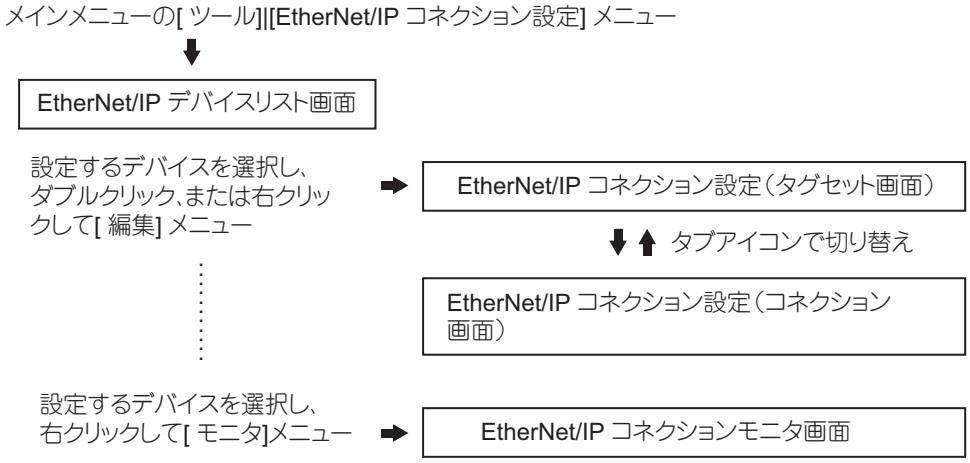
*1. ネットワーク変数とは、グローバル変数テーブルで [ネットワーク公開属性] が [出力] もしくは [入力] に設定された変数のことです。

A-2-3　EtherNet/IP コネクション設定機能

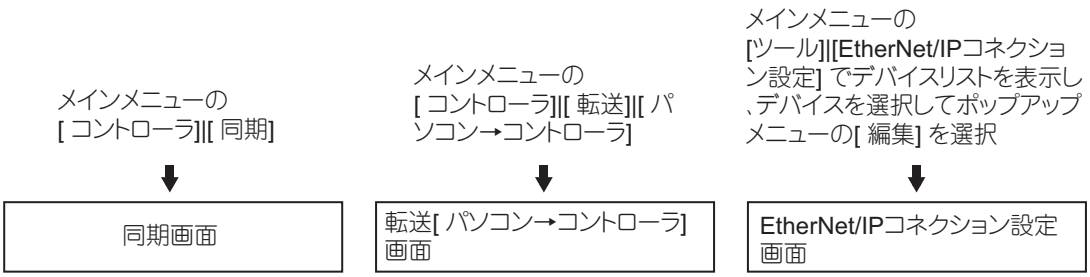
EtherNet/IP の EtherNet/IP コネクション設定機能の画面構成について説明します。

● 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）

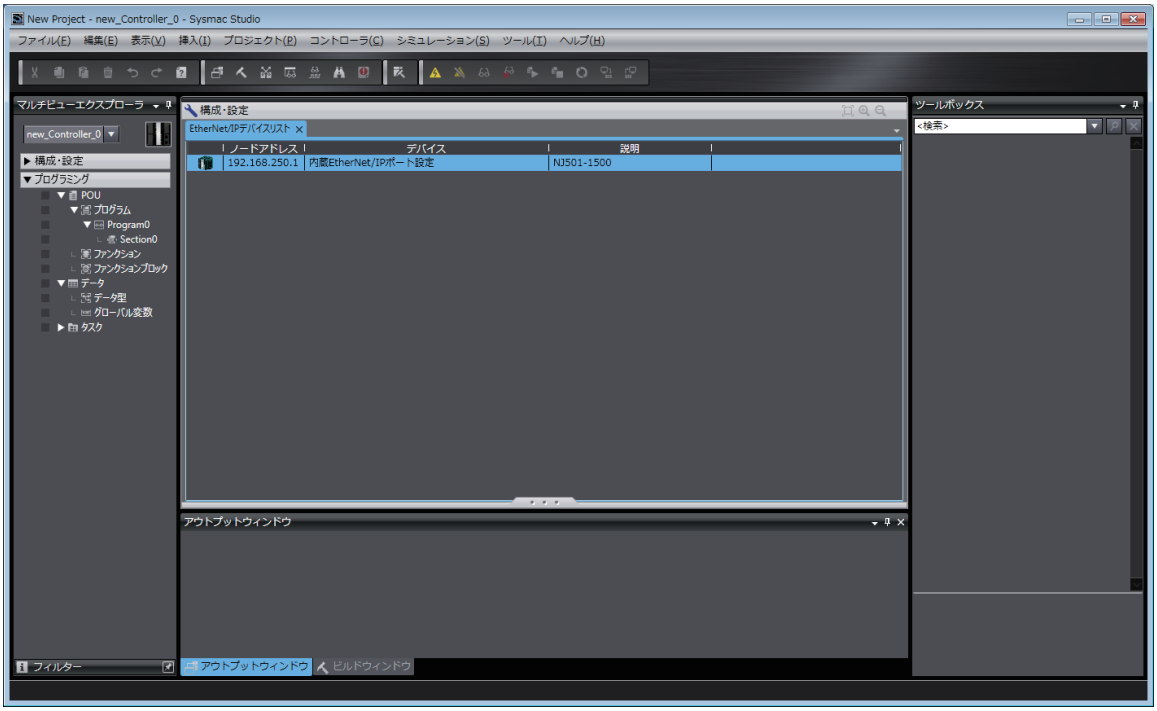


使用上の注意

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使用してください。
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 デバイスリスト画面（「タグ・タグセットの登録」(P.A-13)）**

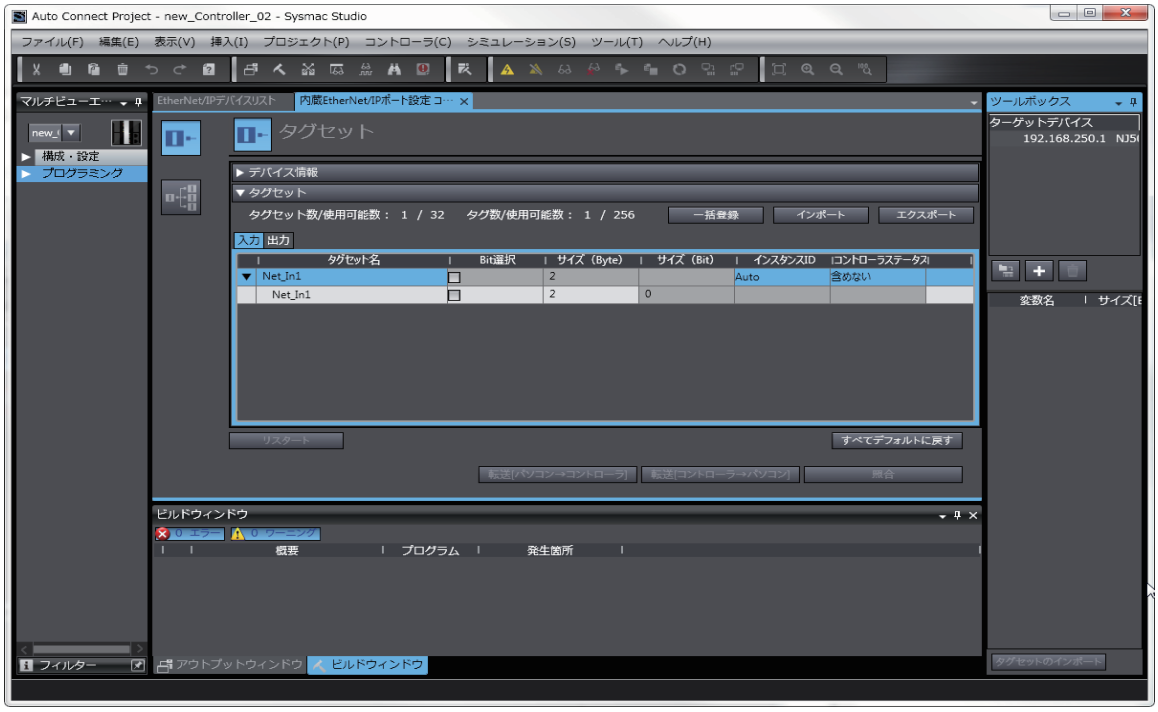
  EtherNet/IP コネクションを設定できるデバイスが表示されます。



● **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-13)）**

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。

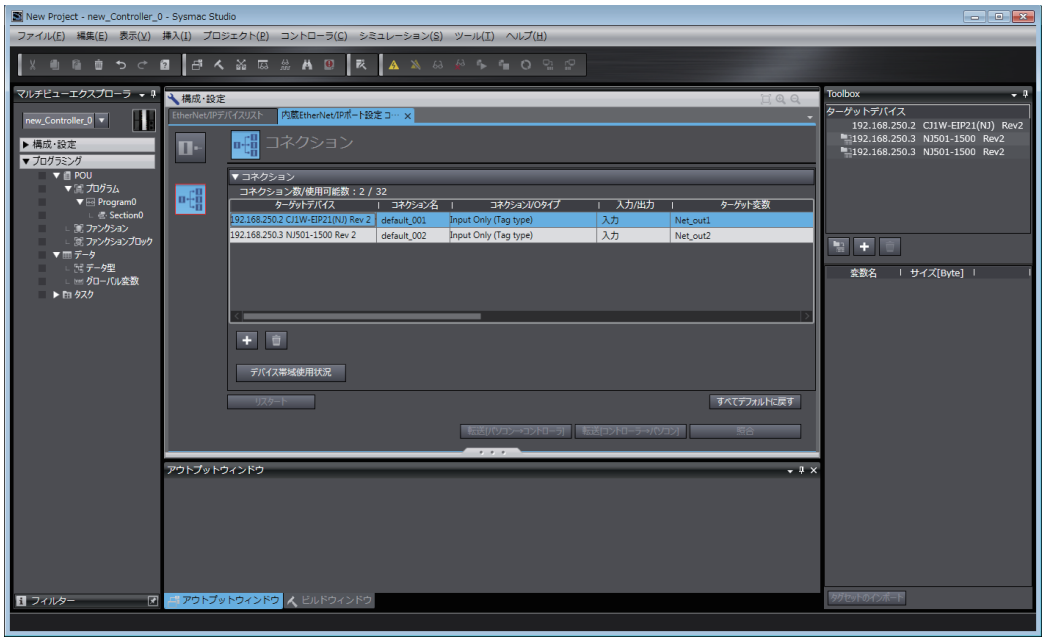


- **EtherNet/IP コネクション設定（コネクション画面）（📖「 コネクションの設定（オリジネータデバイス）」(P.A-16)）**

  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。

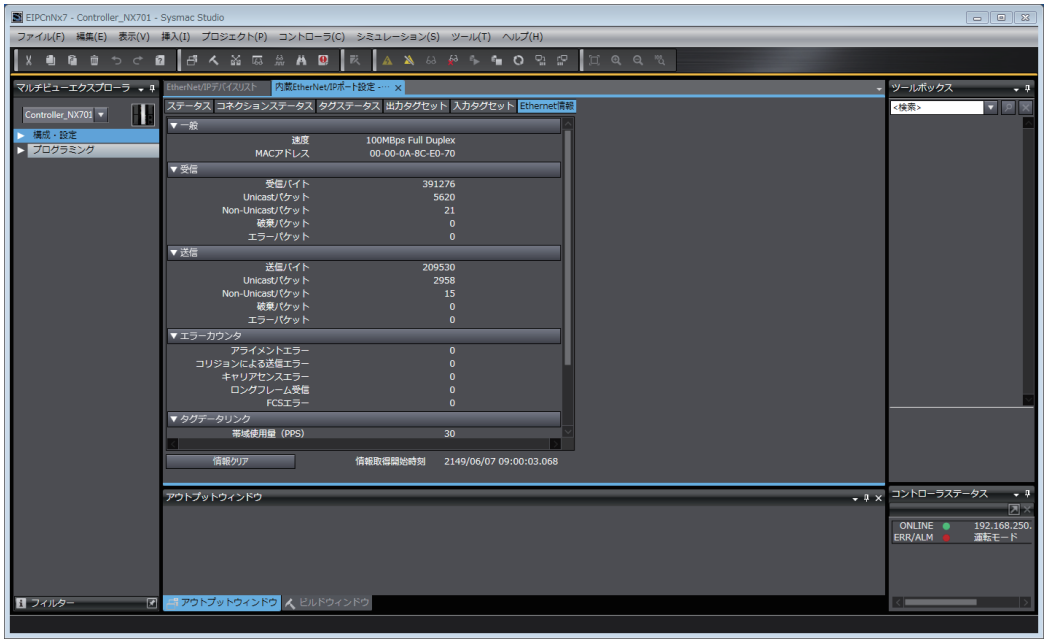
使用上の注意

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、コネクション設定を一括で変更してください。



- **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-31)）**

  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。

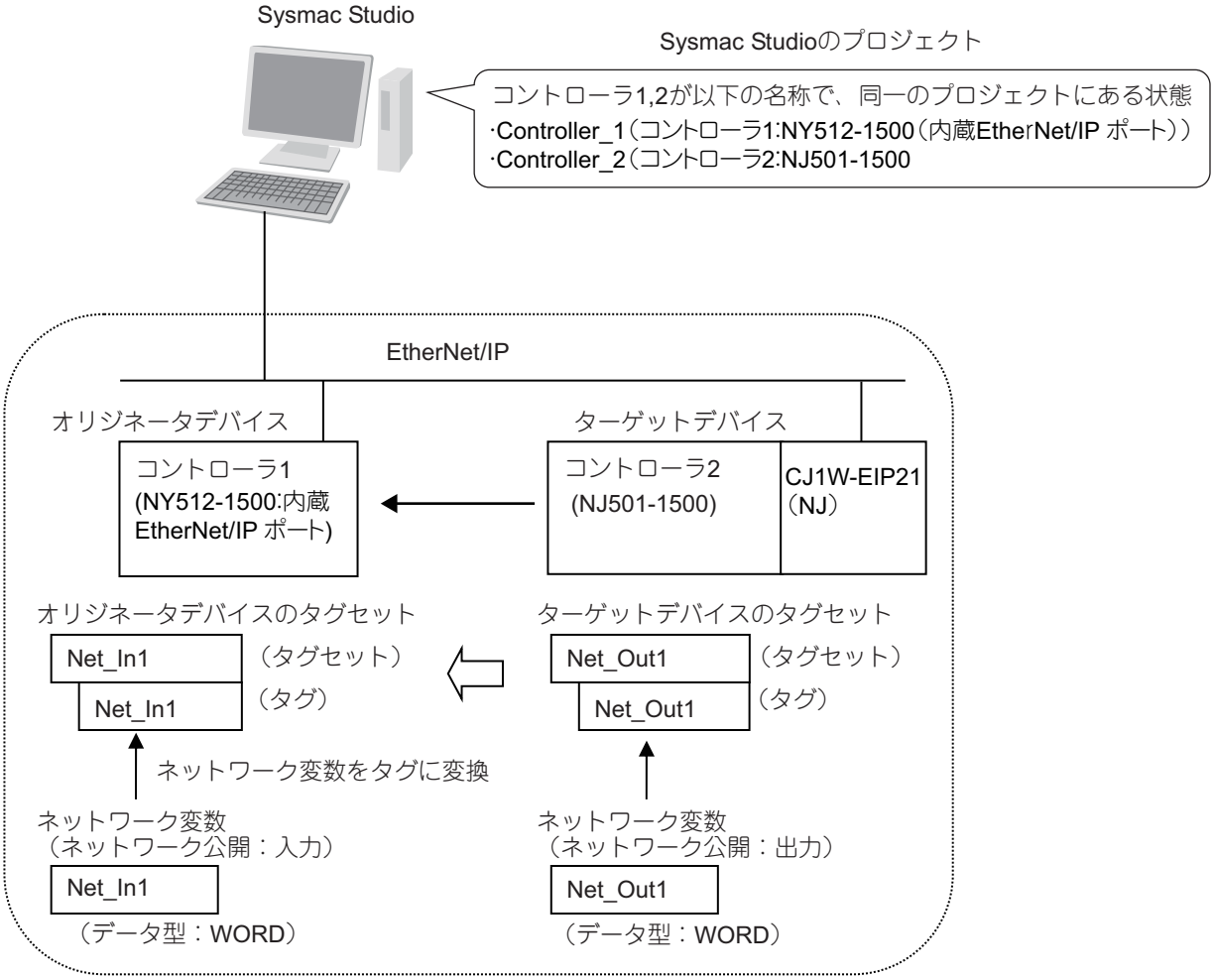


A-2-4 Sysmac Studio による EtherNet/IP コネクション設定

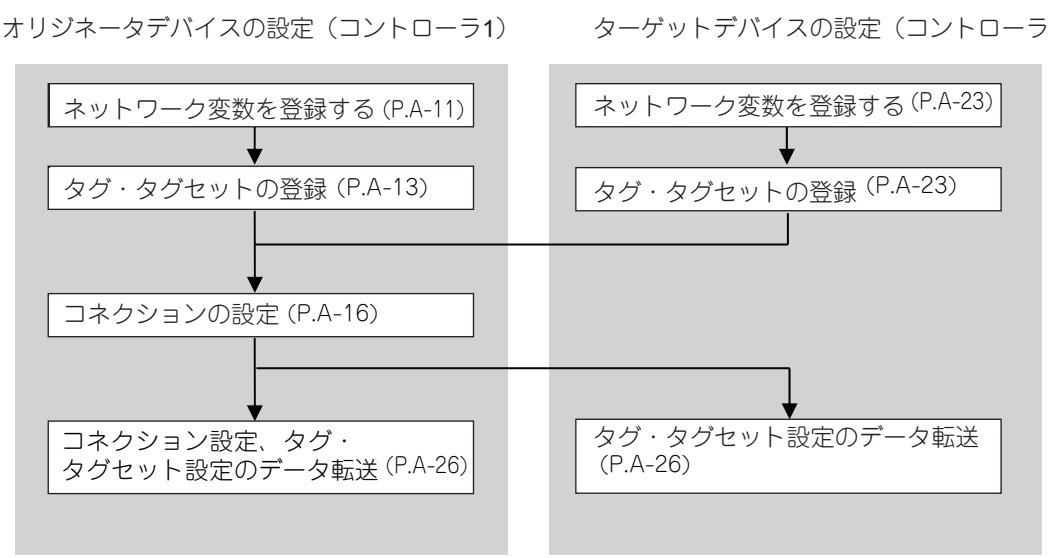
Sysmac Studio による EtherNet/IP コネクションの設定手順を説明します。
ここでは、以下のようなシステム構成で、EtherNet/IP コネクションを設定する場合の設定操作について説明します。

例）コントローラ 1 の内蔵 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

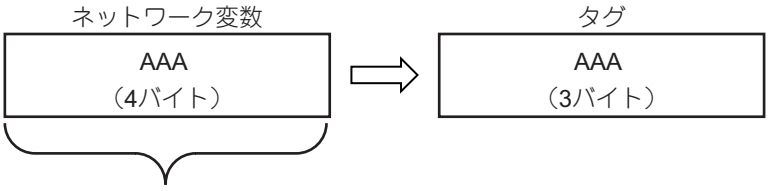
・ネットワーク変数名
EtherNet/IP コネクションの設定のタグ名には、I/O メモリアドレス名称は使用できません。
よって、タグの元となるネットワーク変数の名称にも、I/O メモリアドレス名を使用しないようにしてください。
I/O メモリアドレス名として認識されるのは、以下の文字列です。

(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）

・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。



コントローラ上のメモリは2バイト単位で消費するため、奇数バイトのタグに対応するネットワーク変数は、1バイト足した偶数サイズとする必要があります。

・データの同期性
「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」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。

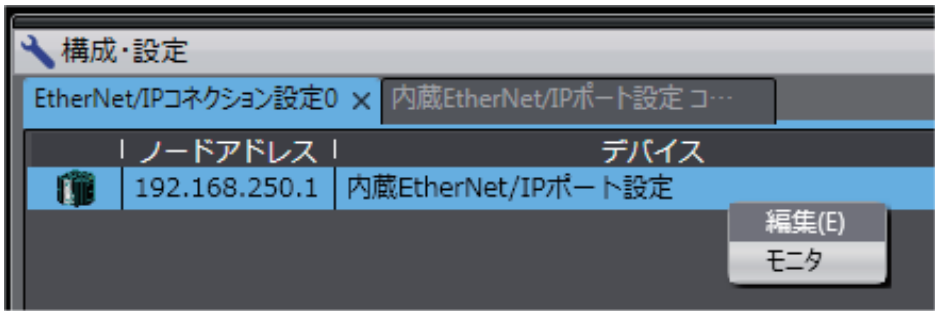
使用上の注意

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。

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［内蔵 EtherNet/IP ポート設定］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。

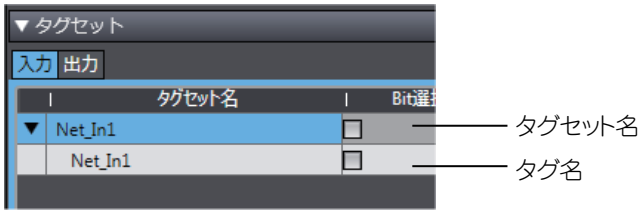
5 **タグセットとタグの個別登録を行います。**

(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。

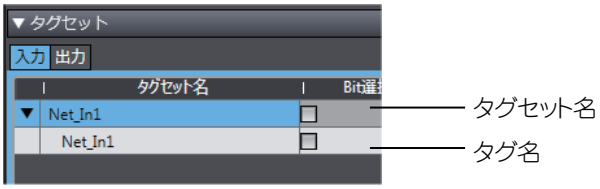
(1) EtherNet/IP コネクション設定画面の入力タブ内で右クリックし [タグセットの一括登録] を選択、または［タグセット一括登録］ボタンをクリックし、[タグセット一括登録設定] 画面を表示します。

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に登録されており、かつ［ネットワーク公開］が［入力］または［出力］に設定されている変数が一覧表示されます。

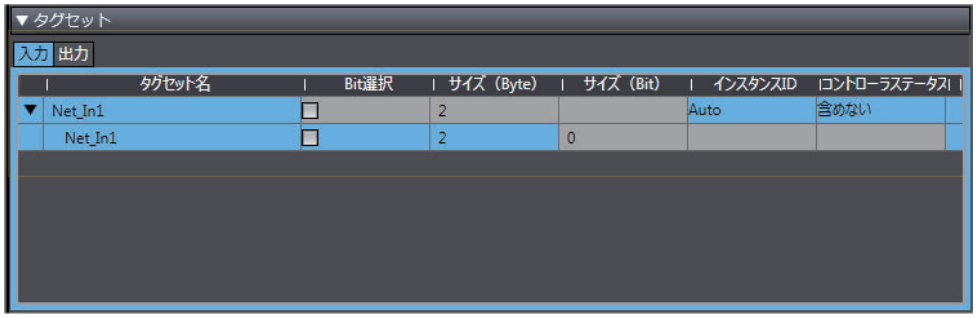


(2) タグとして登録したい変数にチェックが入っている状態にして、[登録] ボタンをクリックします。

(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。

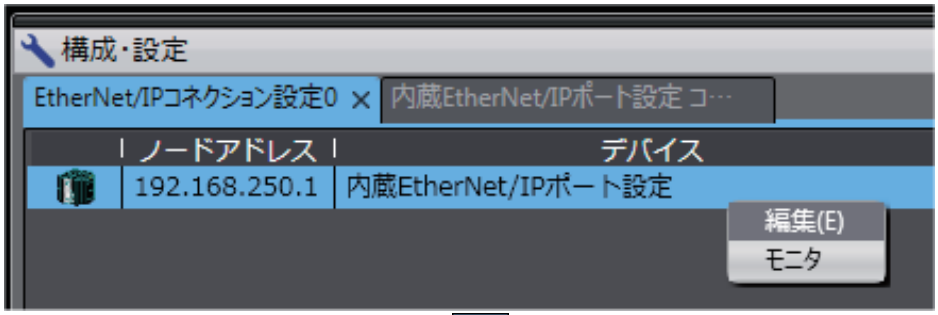
本例によるコネクション設定をする前に、ターゲットデバイスとなるコントローラ 2 について、あらかじめ、タグ・タグセットの登録をしておいてください。
タグ・タグセットの登録操作については、📖「 タグ・タグセットの登録（ターゲットデバイス）」(P.A-23) を参照してください。

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」(P.A-21) 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であるコントローラ 1 の［内蔵 EtherNet/IP ポート設定］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。

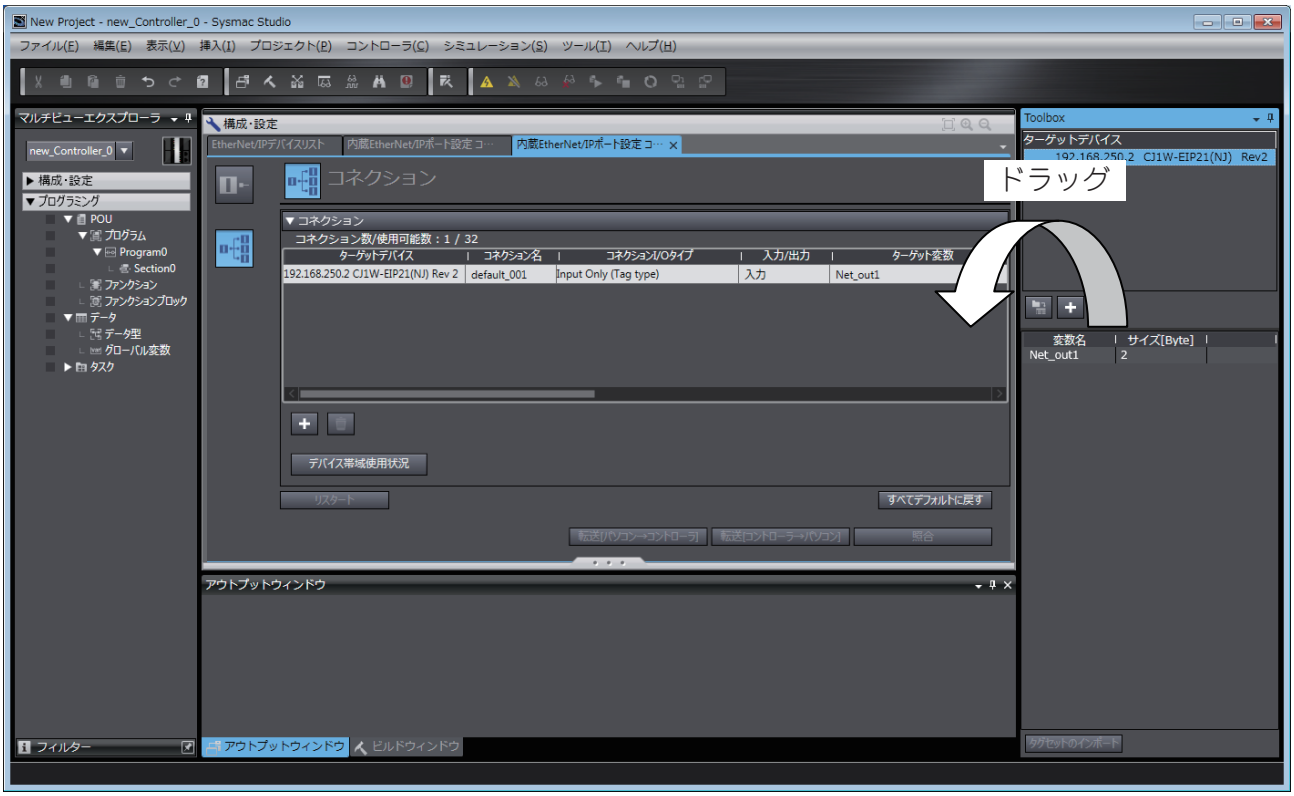


3 **EtherNet/IP コネクション設定画面の（コネクション編集画面表示）アイコンをクリックします。**

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、CJ1W-EIP21、CJ2B-EIP21、CJ2M-EIP21、CJ1W-EIP21(CJ2)、CJ1W-EIP21(NJ)、NX701、NJ501-□□□□、NJ301-□□□□、NJ101、NY5□2 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/NX/NY 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。

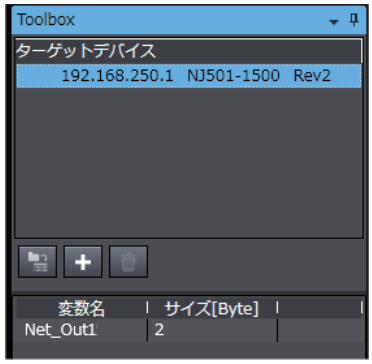
参考

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。

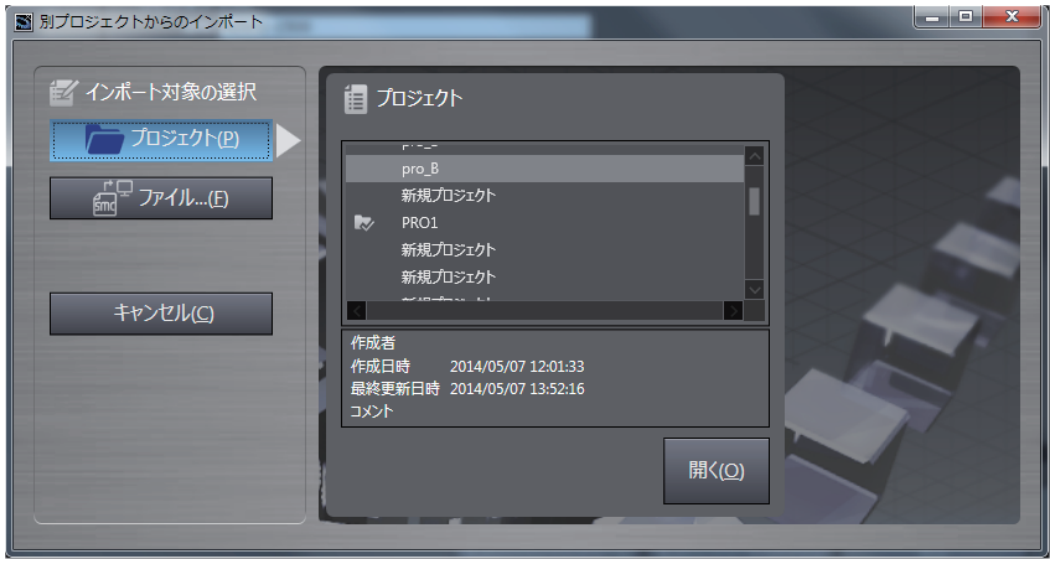
🕮「EDS ファイルの追加」（P.A-20）を参照してください。

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。



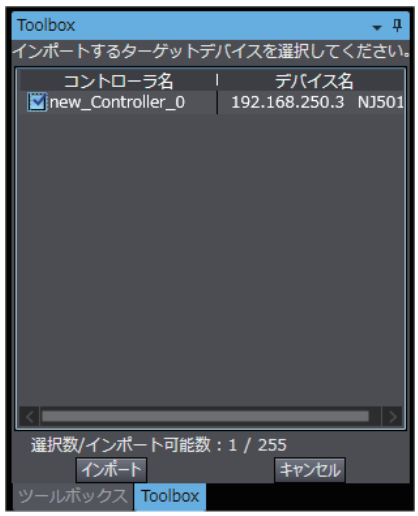
(2) 他プロジェクトのインポート選択画面が表示されますので、[プロジェクト] ボタンをクリックして、インポートするプロジェクトを選択して [開く] ボタンをクリックします。



(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。

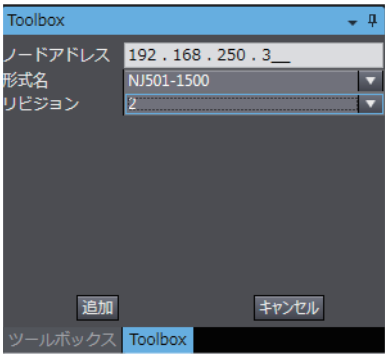
ツールボックスのターゲットデバイス一覧に、インポートした EtherNet/IP デバイスが追加されます。



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。



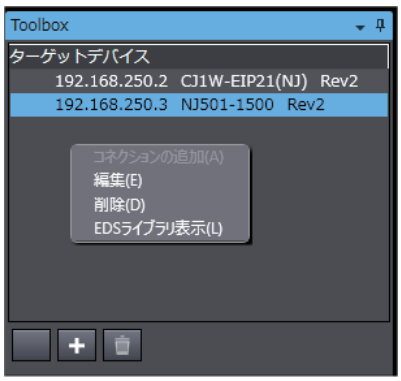
メニュー名	設定する内容
ノードアドレス	ターゲットデバイスの IP アドレスを入力します。
形式名	ターゲットデバイスの形式を選択します。
リビジョン	ターゲットデバイスのリビジョンを選択します。

(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
[形式名]：NJ501-1500
[リビジョン]：2

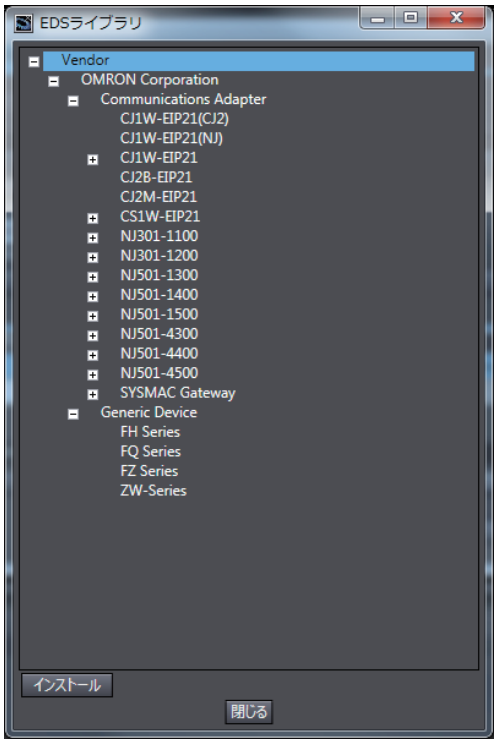
(4) [タグセットのインポート] ボタンをクリックすることにより、Network Configurator で設定したタグセットを、ターゲットデバイスにインポートすることができます。
インポートするファイル（CSV ファイル）は、Network Configurator の［デバイスパラメータの編集］ダイアログボックスの［タグセット］タブで、［ファイル操作｜ファイル操作へエクスポート］ボタンをクリックして作成します。

● EDS ファイルの追加

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。

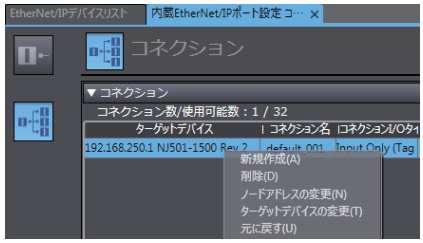
4 **EDS ファイルをインストールした EtherNet/IP デバイスが、EDS ライブラリに追加されます。**
EDS ライブラリに表示されるデバイスは、EtherNet/IP コネクション設定画面のツールボックスで、ターゲットデバイスの追加を行う場合の形式名の選択候補として一覧表示されます。

● 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

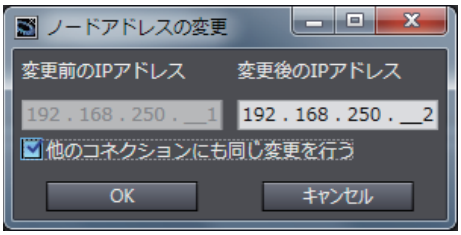
コネクション設定後、ターゲットデバイスの IP アドレスまたはターゲットデバイスのリビジョンを含む形式を変更した場合は、コネクション設定に含まれているターゲットデバイスの設定も同様に変更する必要があります。
ターゲットデバイスの変更は、一括で行うことができます。

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。



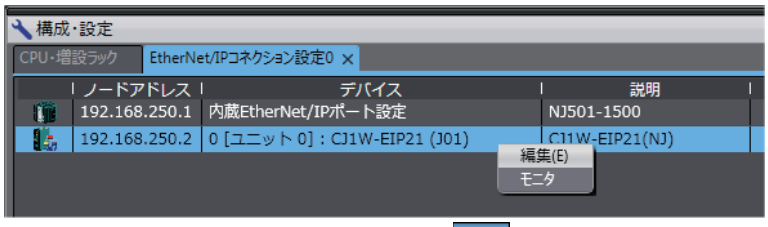
- 変数名：Net_Out1
- データ型：WORD
- ネットワーク公開：出力

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。

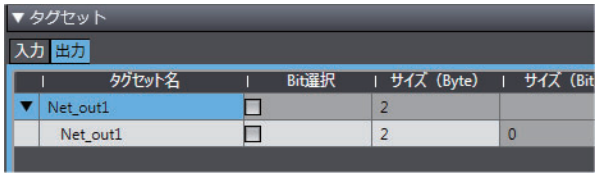
1 **メインメニューの［ツール｜EtherNet/IP コネクション設定］を選択します。**
EtherNet/IP デバイスリスト画面を表示します。

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-13)」参照）



デバイス帯域使用状況の確認

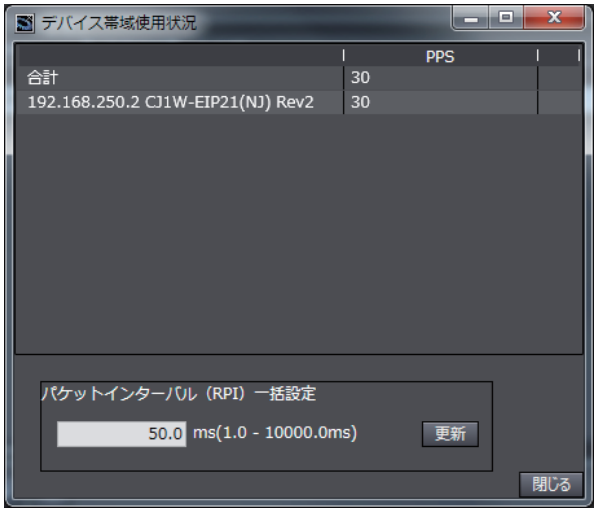
EtherNet/IP コネクションの設定から、デバイスの使用通信帯域幅（PPS）を表示できます。
表示された通信帯域幅 (PPS) は、マルチキャストフィルタありの場合です。

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通信帯域幅のみです。
実際の EtherNet/IP ネットワーク上で使用される通信帯域幅については、他デバイスの EtherNet/IP コネクションで使用する通信帯域幅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使用通信帯域幅と、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。

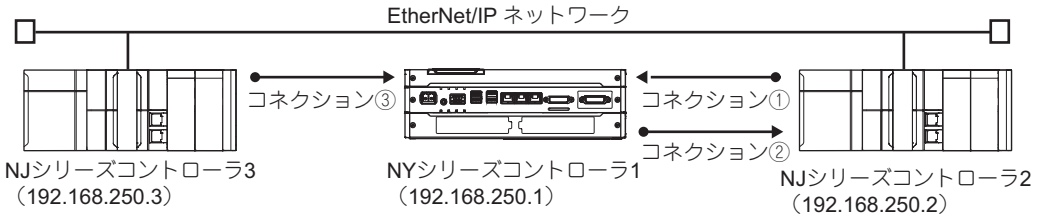
参考

［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。

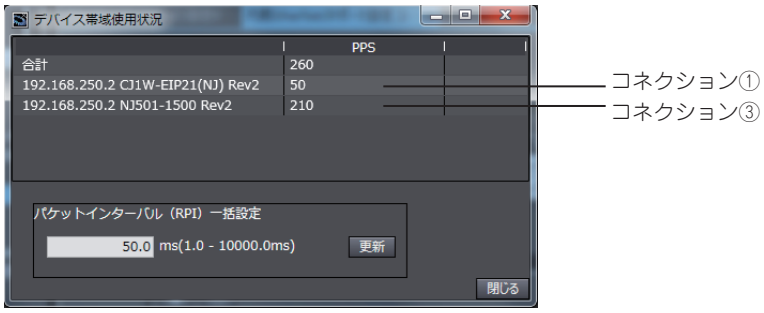
● EtherNet/IP コネクションによる各デバイスの使用通信帯域幅 (PPS) 計算例

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	デバイスの使用通信帯域幅 (PSS)
コネクション①	NY 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	NY 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	NY 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 デバイスの使用通信帯域幅 (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 デバイスの使用通信帯域幅（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 コネクションの使用通信帯域幅 (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。
デバイスの使用通信帯域幅 (PPS) と、パケットインターバル (RPI) の関係については🕮「13-2-2 タグデータリンクの帯域使用率と RPI の関係」(P.13-9) 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

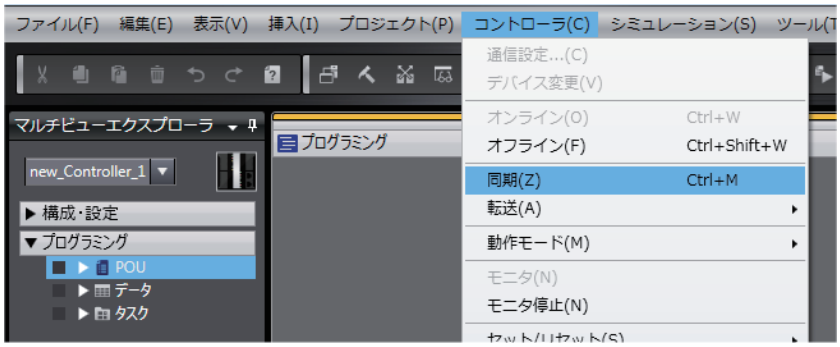
- 設定対象ノードアドレス（IP アドレス）が正しく設定されていないと、意図しないコントローラに接続して不正なデバイスパラメータを設定してしまうことがあるため、接続先コントローラを事前に確認してからダウンロードを実行してください。
- 不適切なコネクション設定が設定されると、設備が予期しない動作をする恐れがあります。また適切なコネクション設定が設定されている場合でも、設備に影響がないことを確認してから、転送を実行してください。
- タグの設定に用いたネットワーク変数が、コントローラに設定されていないと、コネクションがエラーとなります。タグの設定で使用したネットワーク変数が、コントローラに設定さ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、デバイスの使用通信帯域幅 (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。

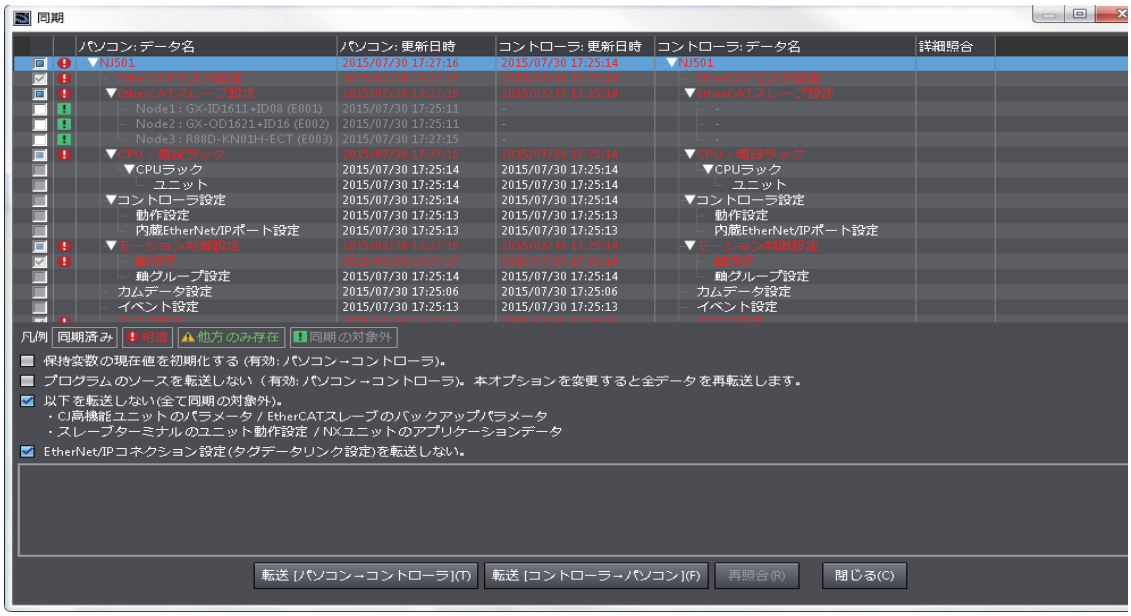
・データを同期・転送する

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（タグデータリンク設定）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。

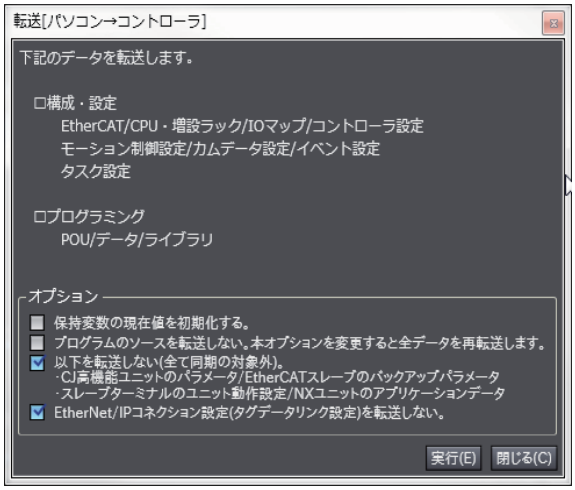
・データを一括・転送する

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（タグデータリンク設定）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使用してください。
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。



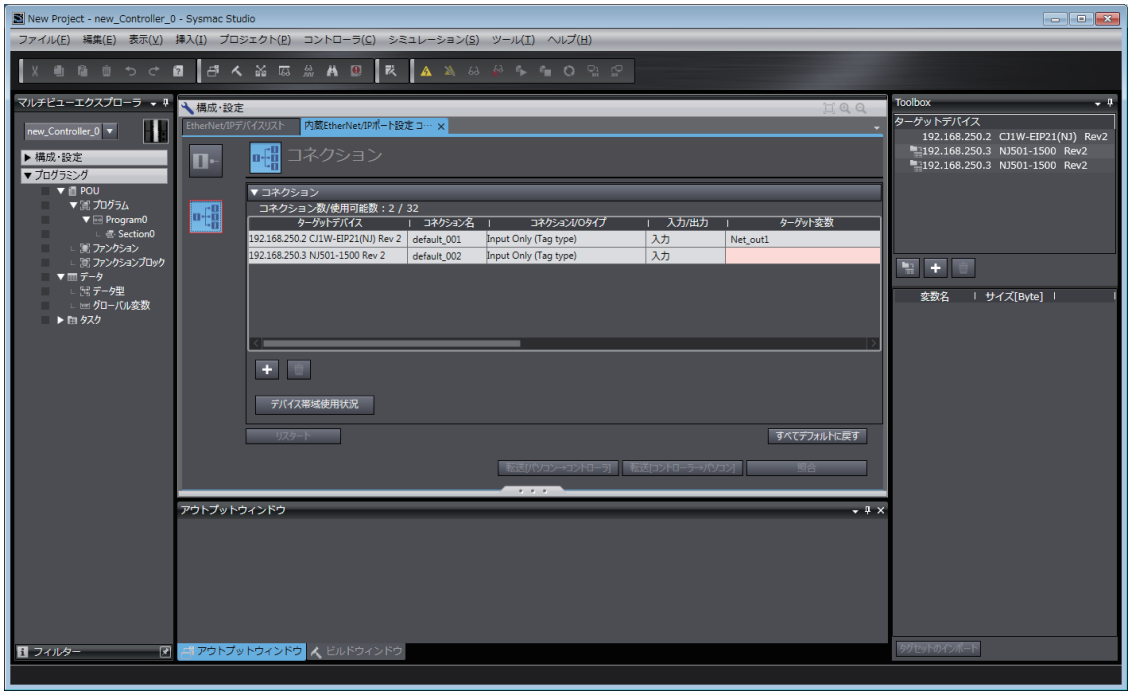
● EtherNet/IP コネクション設定のみ転送する

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。

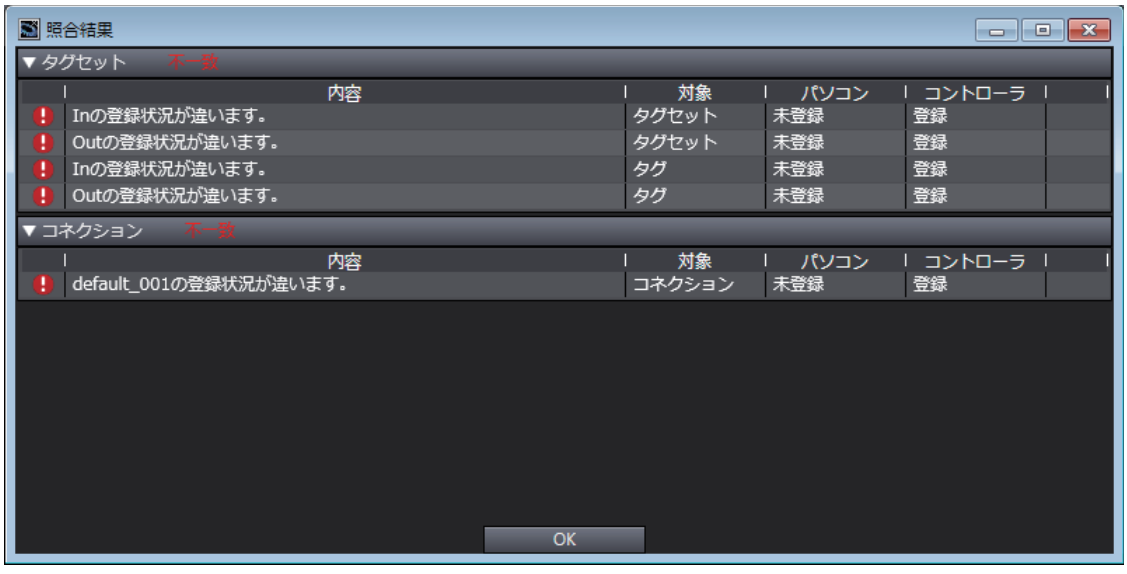
3 **オンライン接続先のコントローラが運転モードの場合、プログラムモードに変更してから転送するかの確認メッセージが表示されます。**



● 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-29)を使って確認することができます。

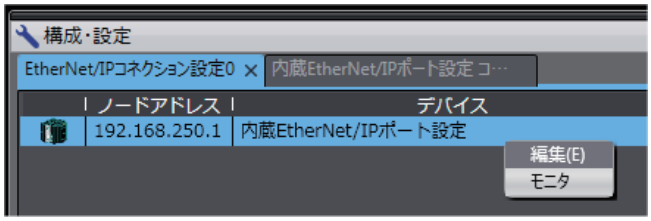
ネットワークの通信ステータス確認

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。

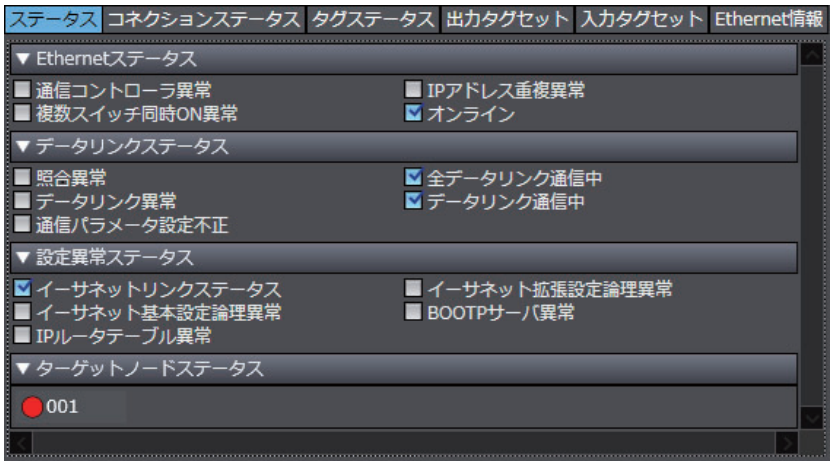


3　確認する通信状態をタブで選択します。

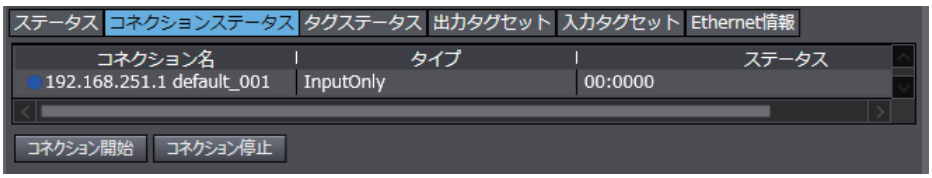
・［ステータス］タブ

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、「14-2 コネクションステータスのコード一覧と処置方法」(P.14-9) を参照ください。

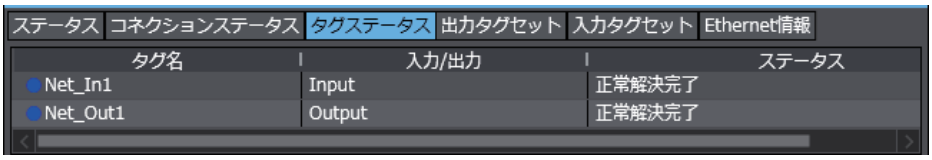


・［コネクションステータス］タブ
各コネクションの現在の状態を表示します。



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。 青：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、 「14-2 コネクションステータスのコード一覧と処置方法」(P.14-9) 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	各タグの現在の状態を色で表示します。 赤：タグ解決異常 青：タグ解決正常 灰色：転送前（デバイスに情報なし）
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コントローラの変数テーブルに設定されていません。 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：以下の 2 つの異常要因があります。 1. Const 属性のため書込みできません。 2. タグデータリンク設定の入出力方向と、コントローラの変数の入出力方向が一致していません。コントローラの変数の「ネットワーク公開属性」設定に誤りがあります。 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。

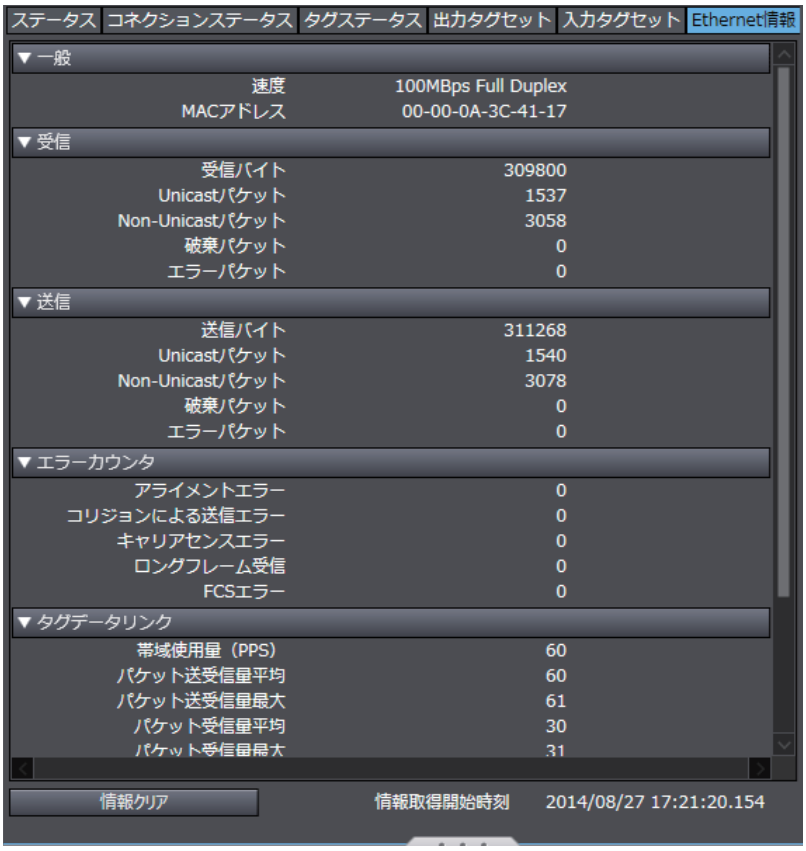
・［出力タグセット］タブ、［入力タグセット］タブ
EtherNet/IP コネクションで使用中の出力タグセット、および、入力タグセットの状態をそれぞれモニタ表示します。
（注）本機能は、NJシリーズコントローラVer.1.08以前の内蔵EtherNet/IPポートでは、使用できません。
各タグの▼をクリックして、詳細を表示させます。



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（パケットインターバル）	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 ハートビート送信周期［ms］	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ハートビート送信周期（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（実際のパケットインターバル）	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 実際のハートビート送信周期 [ms]	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実際のハートビート送信周期（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 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、デバイスの使用通信帯域幅（帯域使用量：PPS）などを確認することができます。



NY シリーズコントローラの画面例

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、『NY シリーズ トラブルシューティングマニュアル（SBCA-438）』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

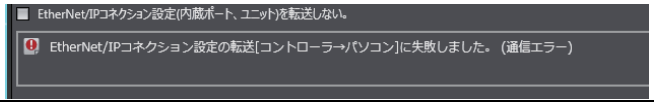
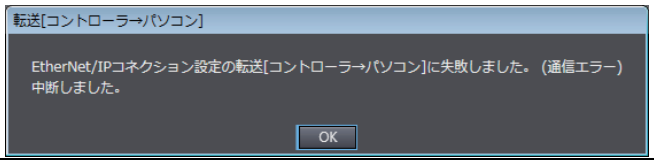
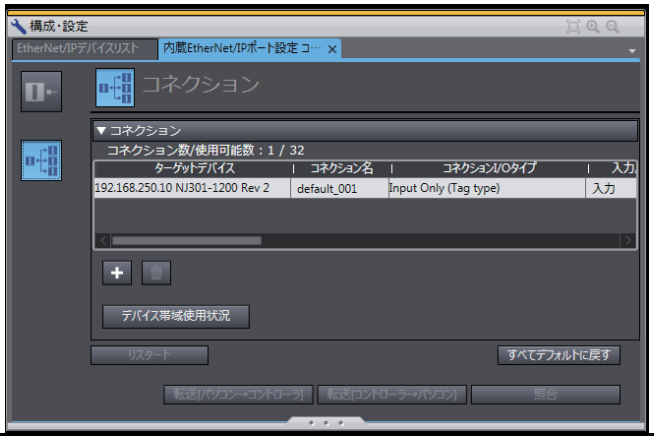
Windows のファイアウォール機能が有効になっていると、初めてコントローラとオンライン接続したときに、接続を確認するダイアログが表示される場合があります。その場合は、以下を選択してください。

- Windows Xp/Vista：「ブロックを解除する」
- Windows 7 以降：「アクセスを許可する」

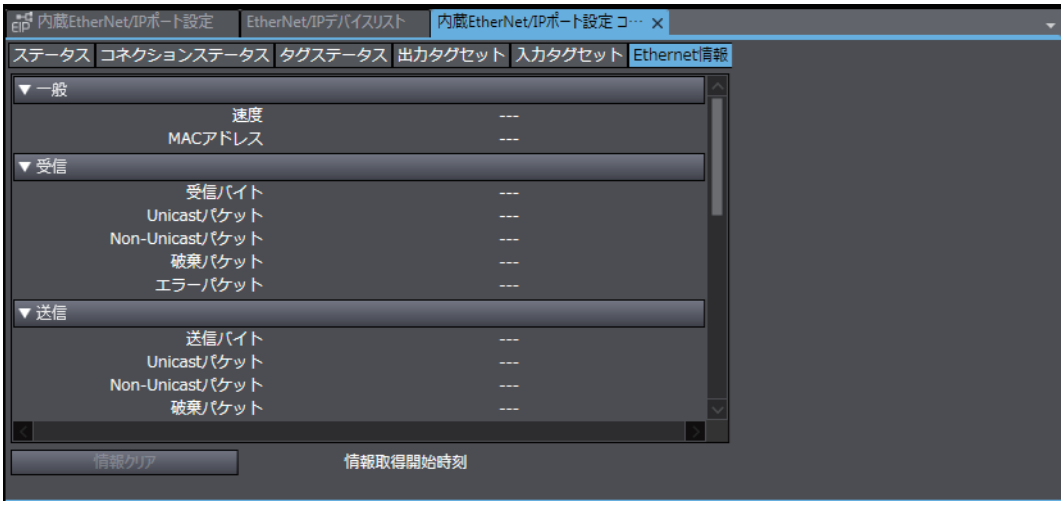
もし、上記以外を選択した場合は、コントローラにオンライン接続できても、EtherNet/IP コネクション設定の転送やモニタが、正常にできない現象が発生する場合があります。
そのような場合、下記の対策 1 または対策 2 を行ってください。

● 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）

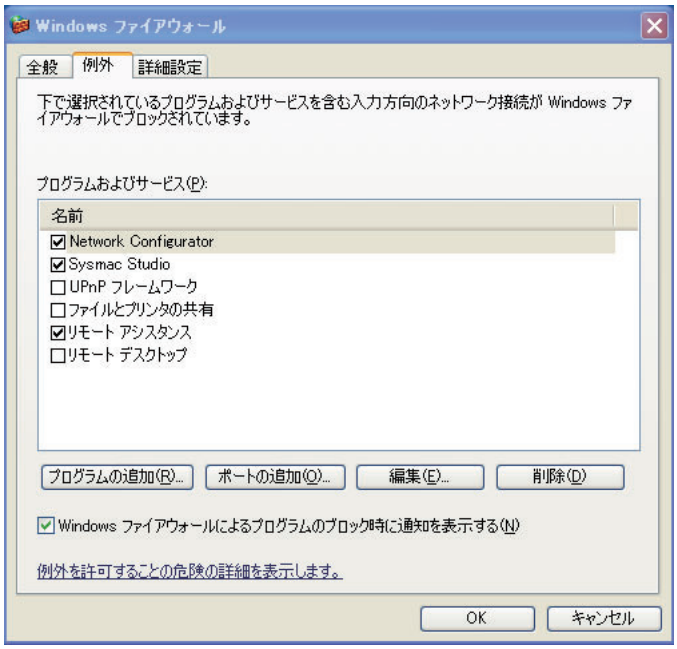
使用上の注意

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 NY 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。**

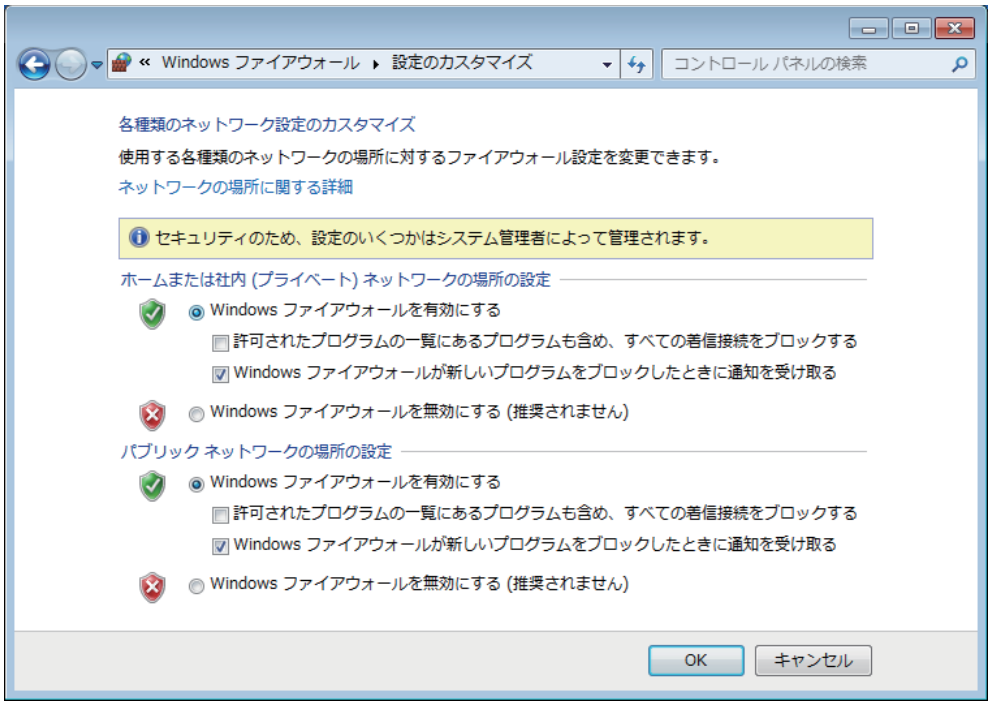


- Windows Vista/7 以降の場合

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

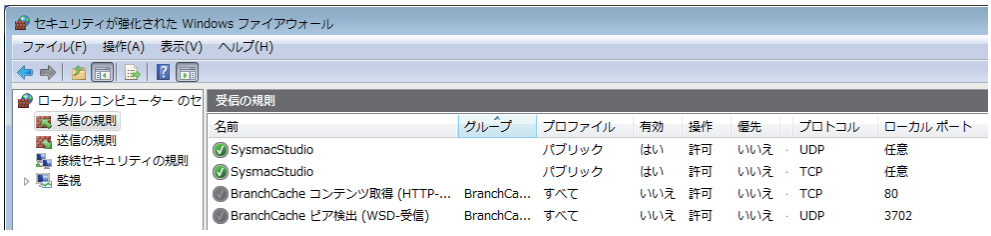
2 **［Windows ファイアウォールの有効化または無効化］を選択します。**
「設定のカスタマイズ」が表示されます。

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」リストの「SysmacStudio」をダブルクリックします。**
［SysmacStudio のプロパティ］画面が表示されます。



6 **［SyamacStudio のプロパティ］画面の［全般］タブで以下の設定を行います。**
［全般］:「有効」にチェックを入れる
［操作］:「接続を許可する」を選択

● 対策 2（コントローラの電源を OFF/ON する）

NY 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。

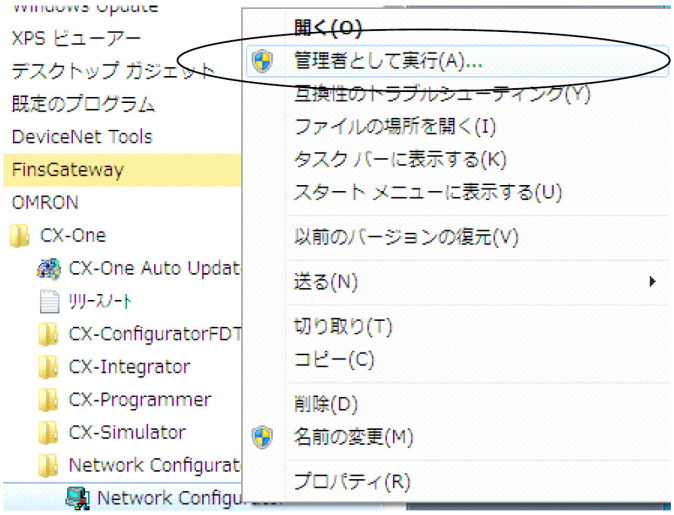
A-3 EDS ファイルの管理

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 /Windows 7 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。

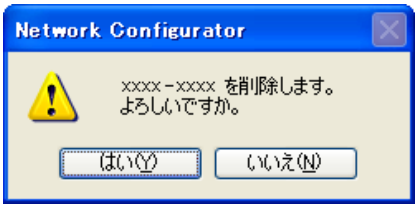
A-3-3　EDS ファイルの削除

［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。**

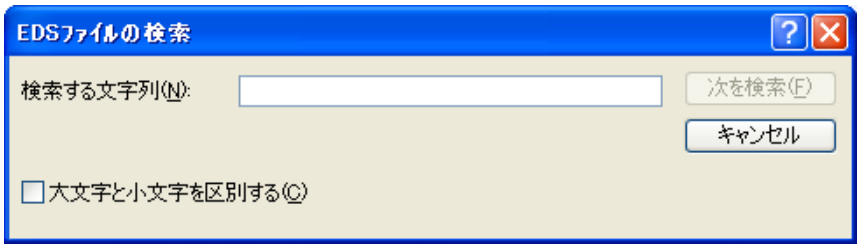
3 **フォルダ・ファイル名を指定して、［保存］ボタンをクリックします。**
ファイルへの保存が実行されます。

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、Windows XP/Windows Vista/Windows 7 以降で使用する際の留意点

Windows XP（SP2 以降）/Windows Vista/ Windows 7 以降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 NY シリーズコントローラを接続して、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/NY Series Ethernet Direct I/F］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 NY 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。

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、NY シリーズコントローラに接続すると、[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。**

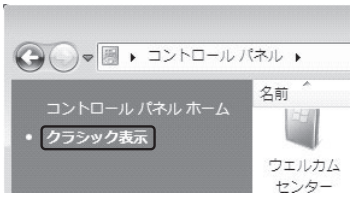
2 **[ブロックを解除する] を選択します。**

Network Configurator に 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 以降の場合

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。

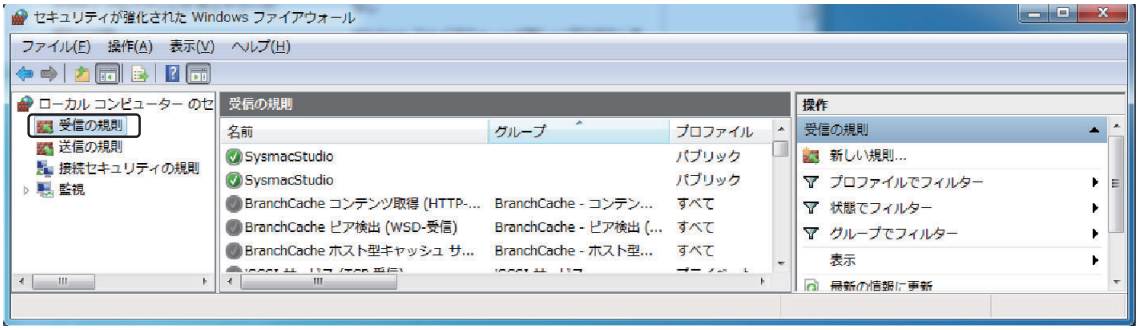
1 **スタートメニューから、［コントロールパネル］を選択し、「クラシック表示」を選択します。**



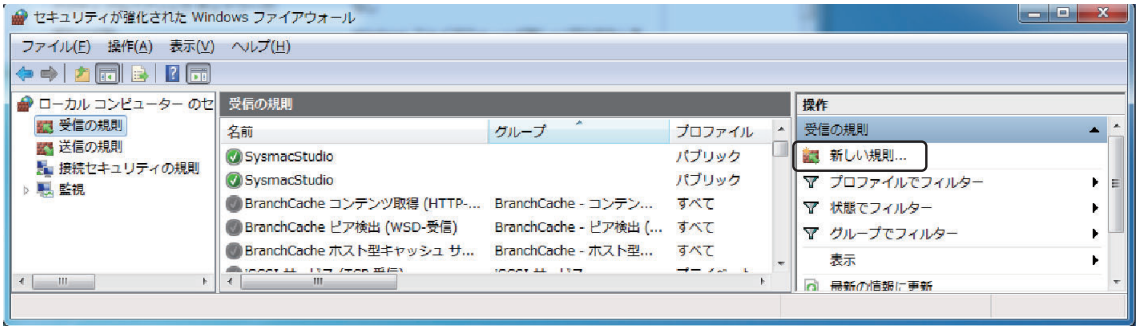
2 **「管理ツール」選択し、表示されるダイアログボックスから、「セキュリティが強化された Windows ファイアウォール」を選択します。**



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。**

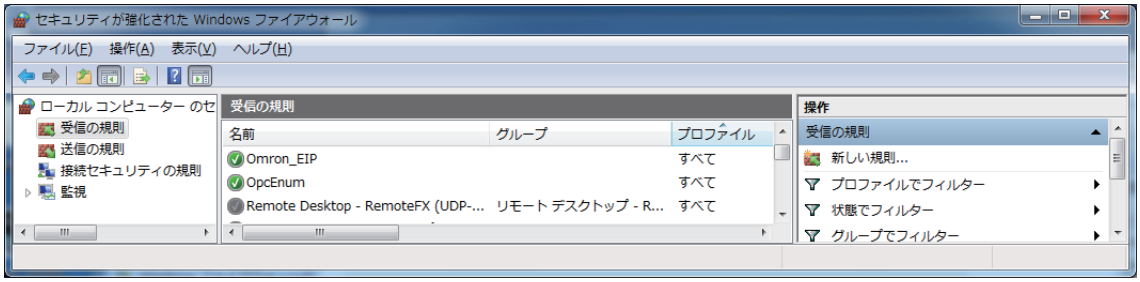


5 **［新規の受信の規則ウィザード］ダイアログボックスで、ステップにしたがい、各項目の設定を行います。各項目で次のように選択して［次へ］ボタンをクリックします。**

「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

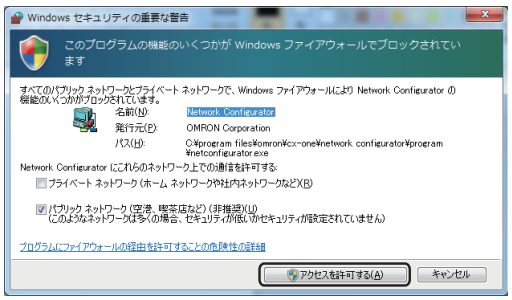
6 **名前を入力して、［完了］ボタンをクリックすると、「受信の規則」に、作成した規則（例：Omron_EIP）が登録されます。**

［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、NY シリーズコントローラに接続すると、[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 **［アクセスを許可する］を選択します。**



（Windows 7 の場合）

Network Configurator に 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

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。

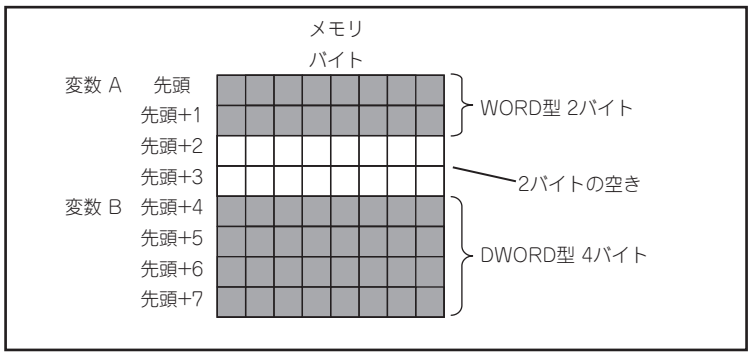
- NY シリーズコントローラと他コントローラ間で、EtherNet/IP のタグデータリンクや、CIP メッセージによる変数アクセスを行うとき
- ID タグなどのコントローラ以外の機器と、構造体型や共用体型変数でデータ交換を行うとき

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_DAY,DATE_AND_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

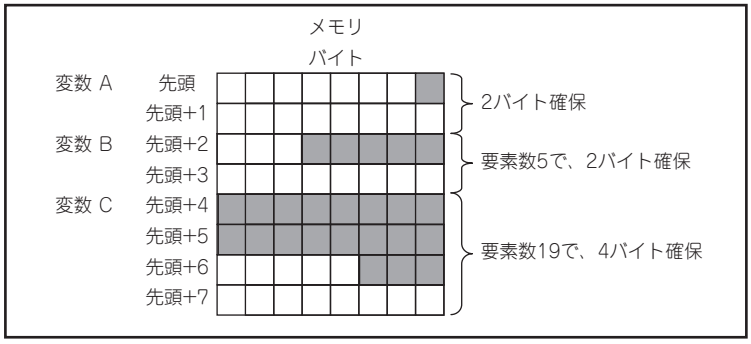
*1. BOOL 型の配列の場合は、例外です。BOOL 型の配列のために確保されるメモリの量については、後の使用上の注意を参照してください。

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。

$$\text{メモリの量}=2\left[\frac{\text{要素数}-1}{16}\right]+2$$

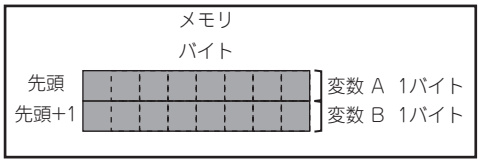
括弧内の計算結果の小数点以下は切り捨て

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

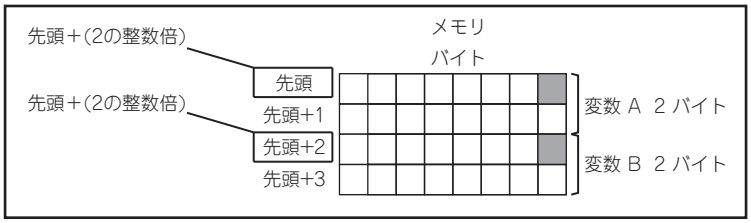


変数テーブル

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

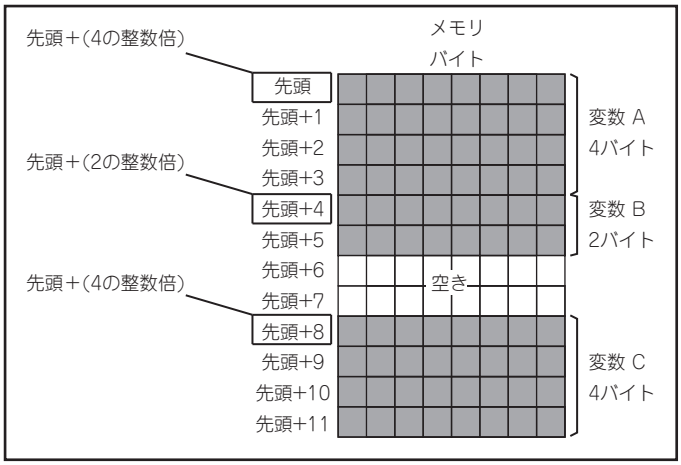


変数テーブル

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

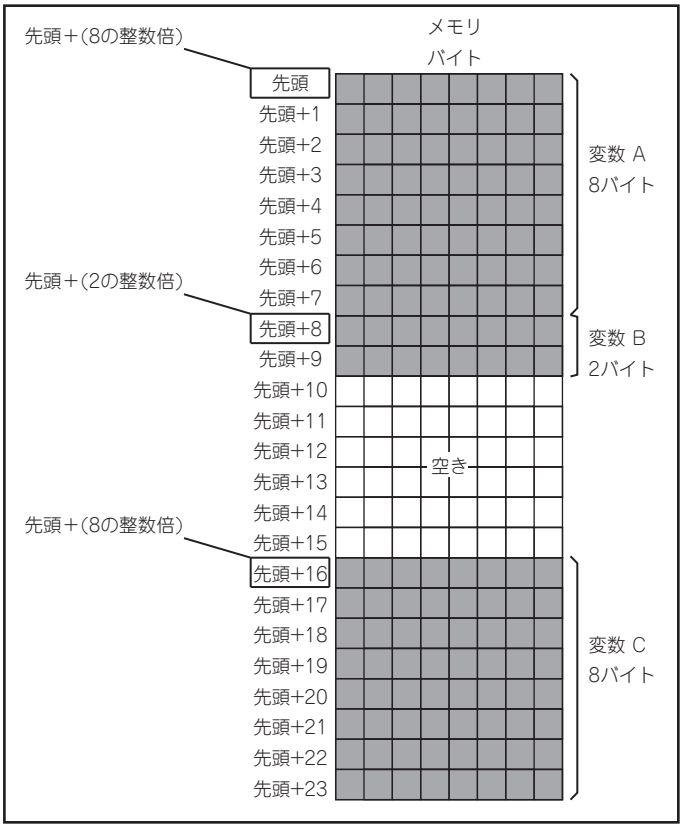


変数テーブル

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



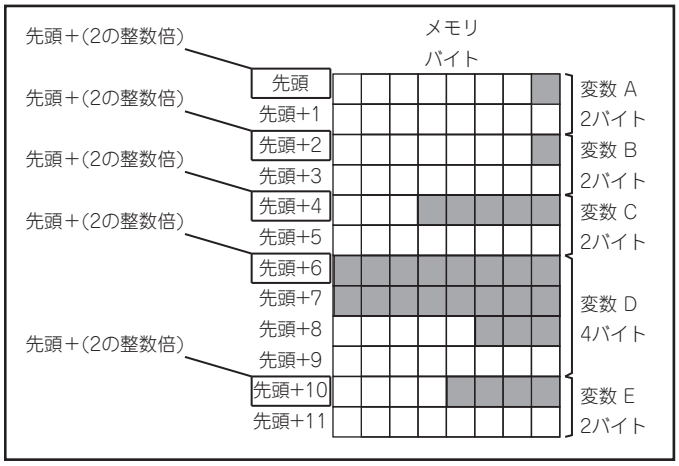
変数テーブル

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。

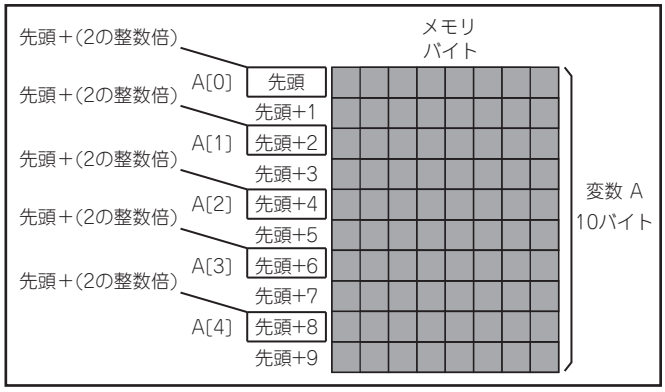
例）BOOL 型変数が 2 個、要素数 5 の BOOL 型配列、要素数 19 の BOOL 型配列、要素数 4 の BOOL 型配列の順で、変数が連続している場合



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

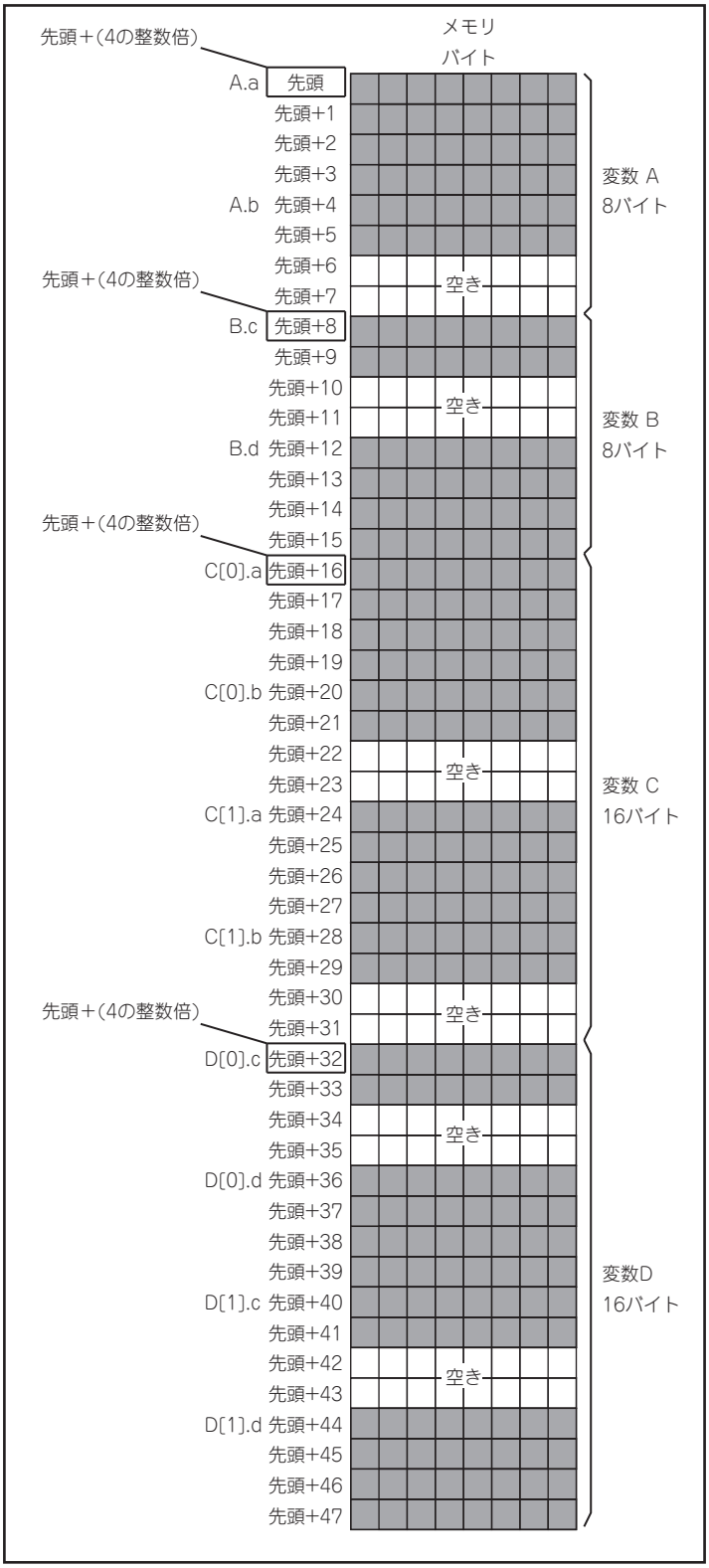
名称	データ型
変数 A	ARRAY[0..4] OF INT

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

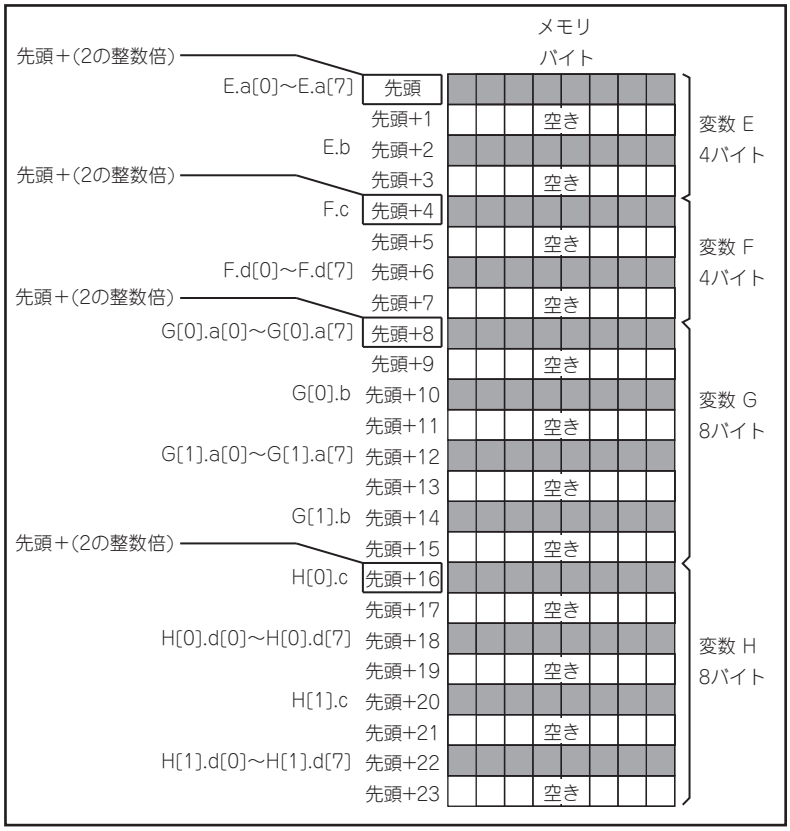
名称	データ型
構造体 STR_B	STRUCT
c	INT
d	DINT

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

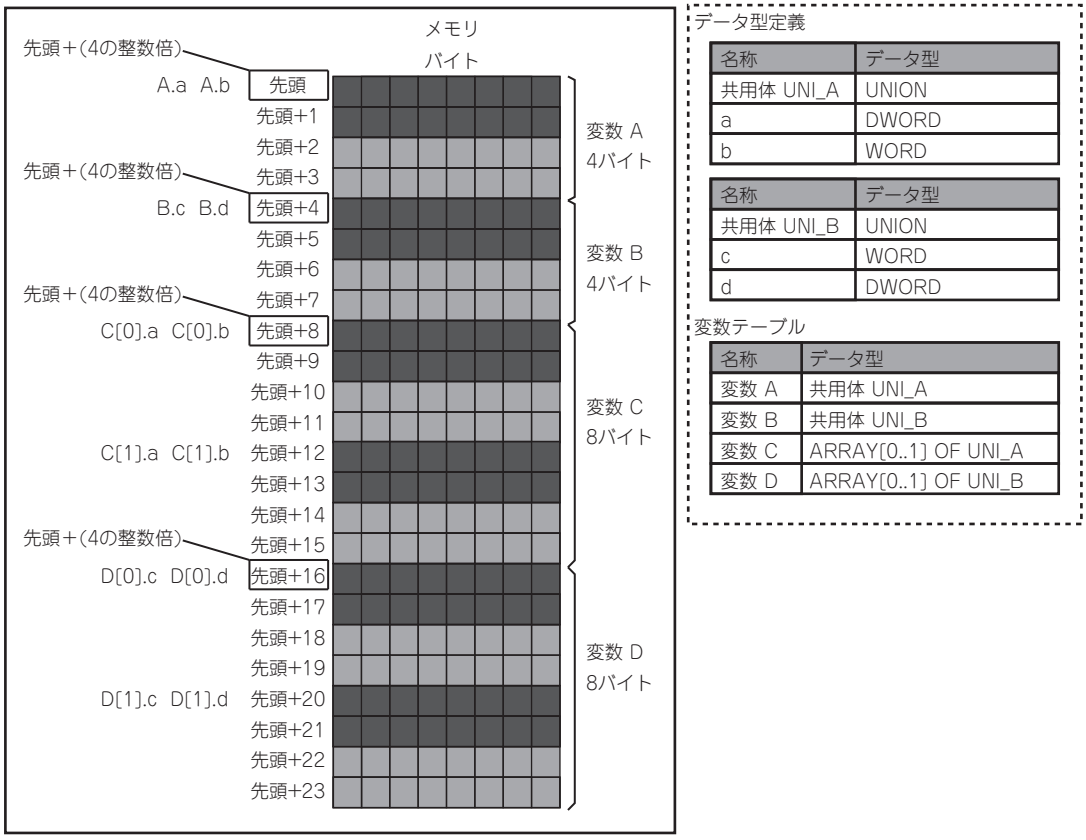
名称	データ型
変数 E	構造体 STR_C
変数 F	構造体 STR_D
変数 G	ARRAY[0..1] OF STR_C
変数 H	ARRAY[0..1] OF STR_D

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



A-5-2　留意するべきケース例

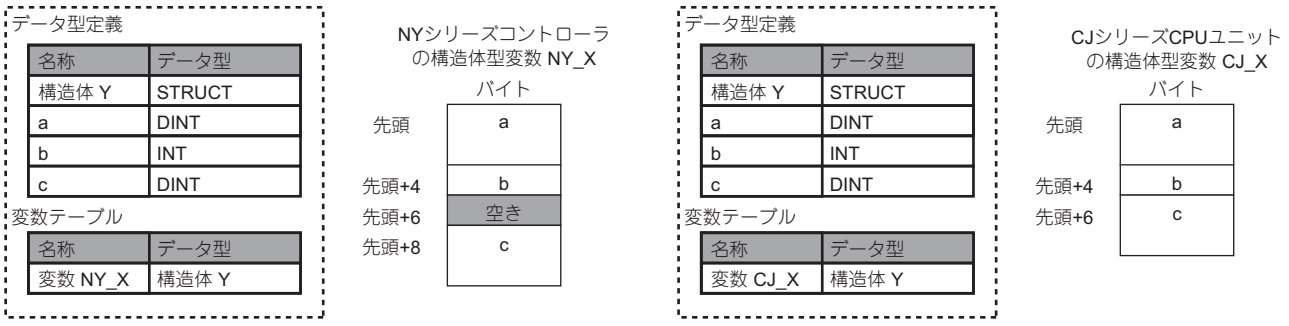
NY シリーズコントローラの構造体型変数で、他の機器との通信などでデータ交換するときには、構造体型変数のメンバのメモリ配置を、通信する相手機器と一致させる必要があります。
その場合は、NY シリーズコントローラ側、または通信相手の機器側か、いずれかで本項目で説明する対策を行ってください。

参考

NY シリーズコントローラ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、NY シリーズコントローラと CJ シリーズ CPU ユニットでは、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。

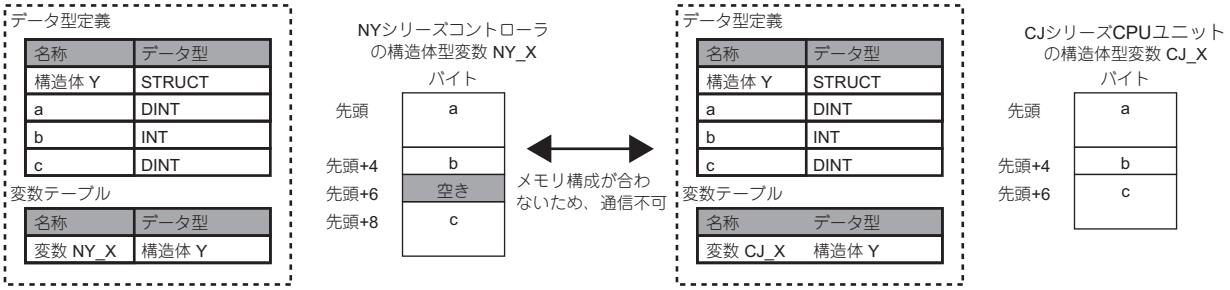


● 対策1:NYシリーズコントローラの構造体型変数のメンバのメモリ配置を変更する

NY シリーズコントローラでは、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。NY シリーズコントローラ側で、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。

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 NY シリーズコントローラ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。

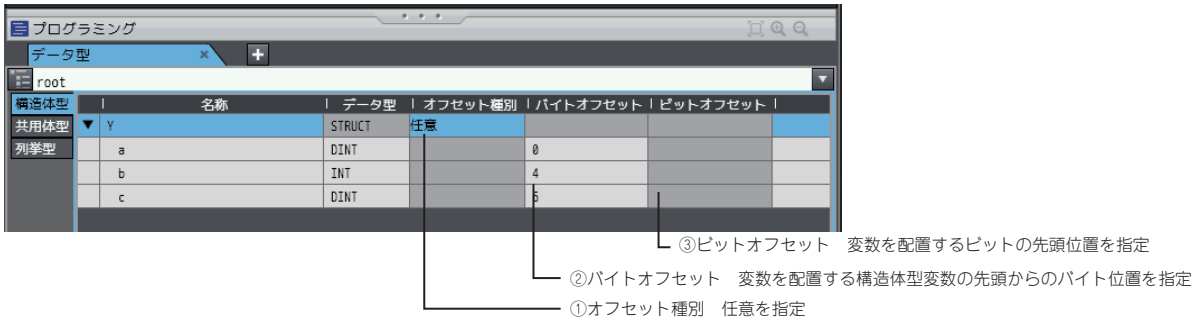
例）NY シリーズコントローラの構造体型変数のメンバのメモリ配置を、CJ シリーズ CPU ユニットの構造体型変数のメンバにあわせて変更する

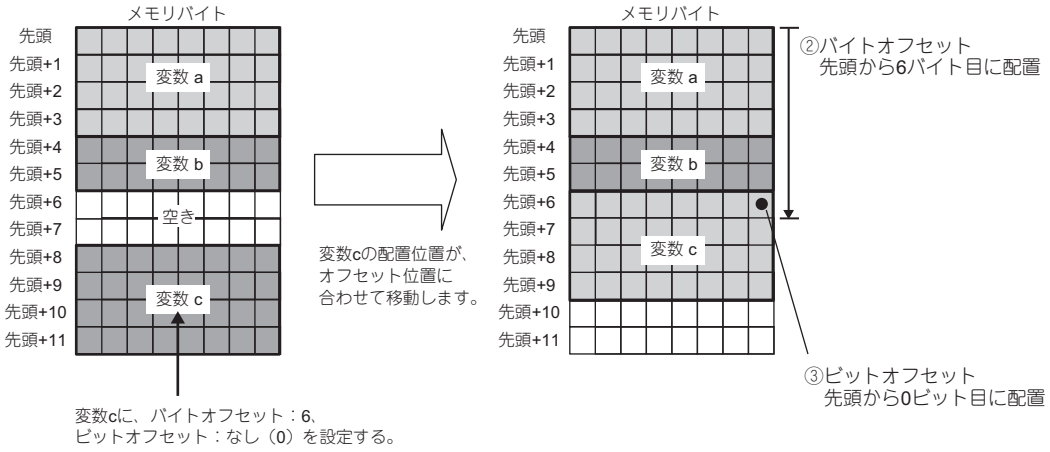


NYシリーズコントローラと、CJシリーズCPUユニットとのメモリ構成を合わすため、
Sysmac Studioでオフセット指定します。

構造体型変数 NY_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。



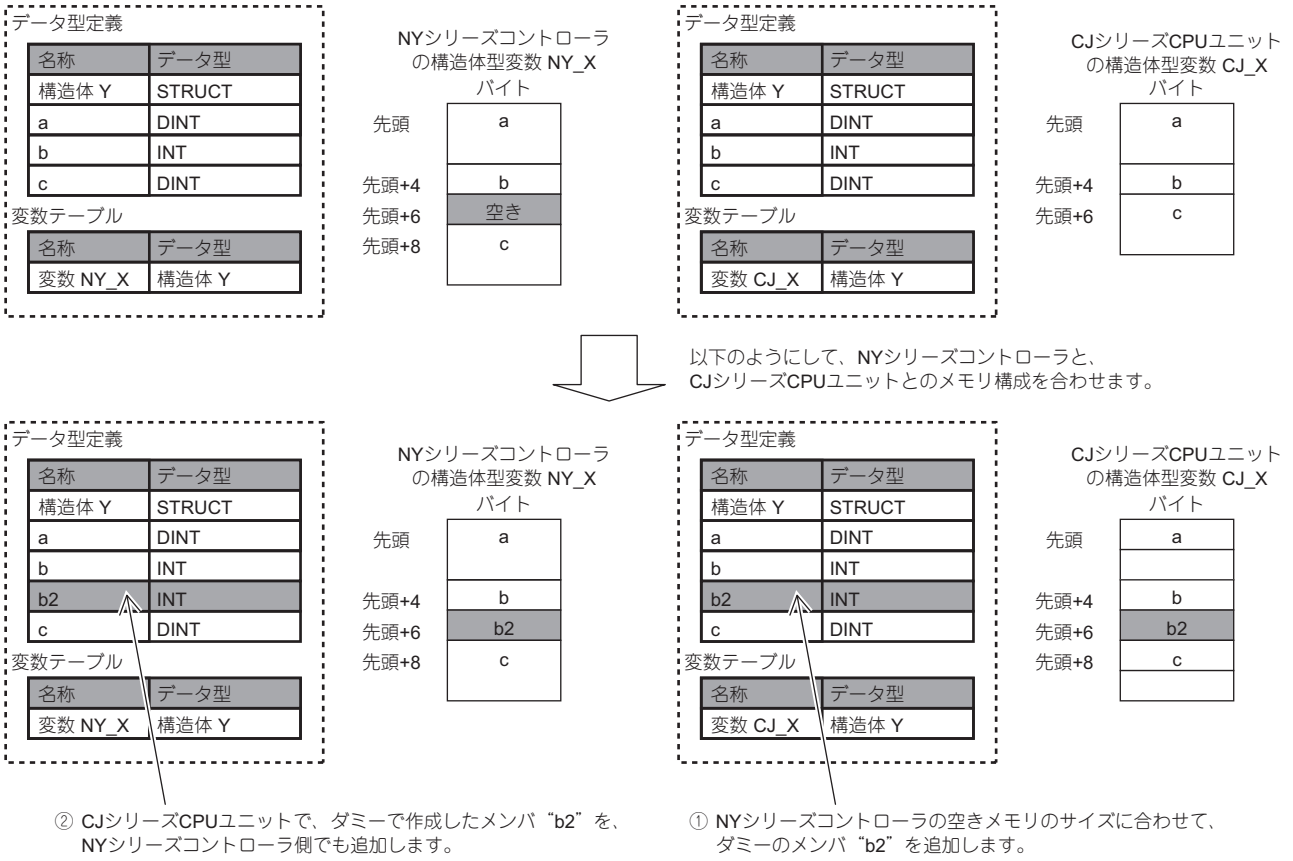




● 対策 2：通信する相手機器の構造体型変数のメモリ配置を変更する

NY シリーズコントローラ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、NY シリーズコントローラ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、NY シリーズコントローラの構造体型変数のメモリ配置にあわせて変更する



A-6 コントローラの外部出力にアクセスする場合の留意事項

NY シリーズコントローラの外部出力に割り付けられている、変数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

NY シリーズコントローラ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。

- 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）

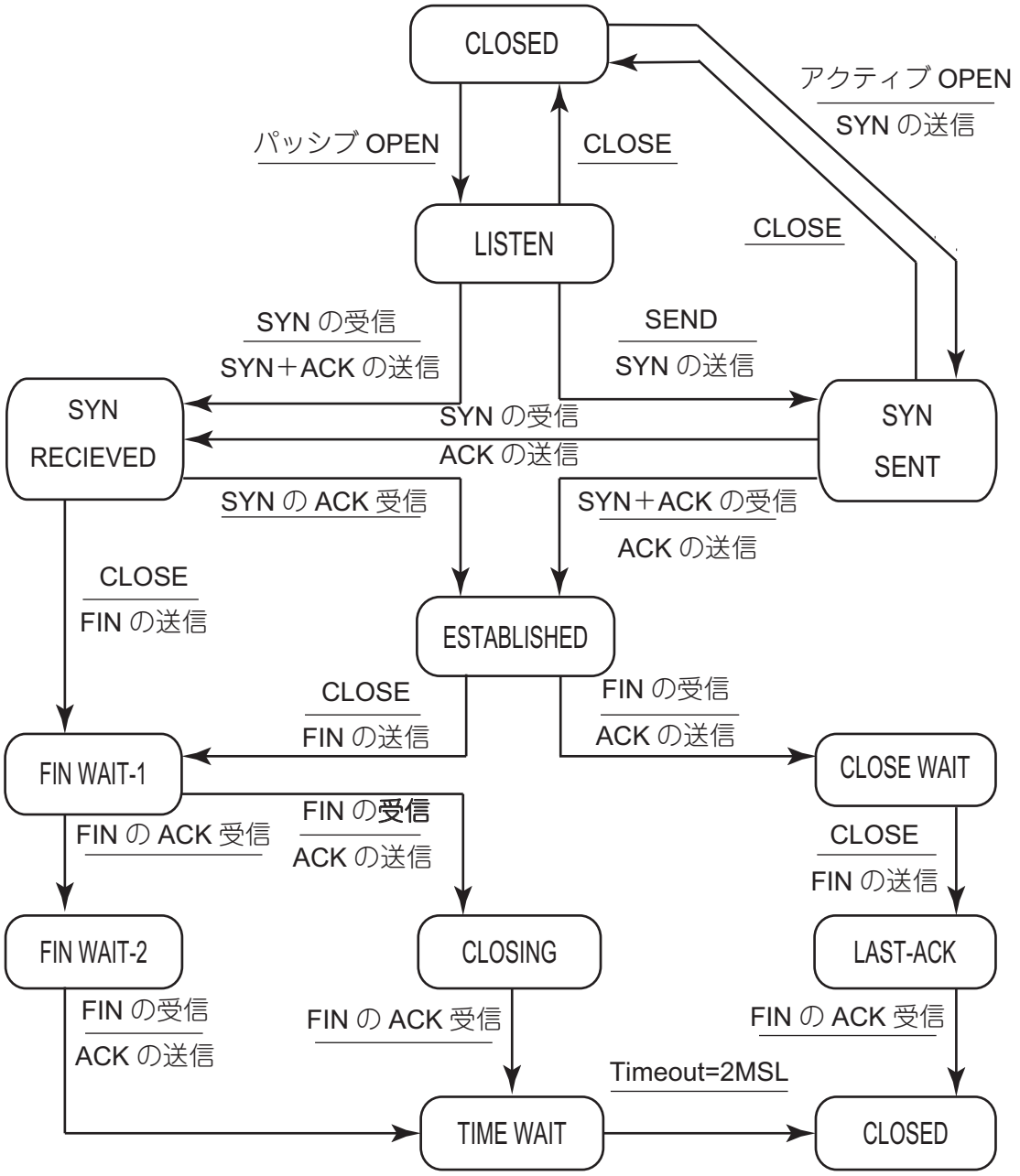
A-7 TCP 状態遷移

TCP は、コネクションの状態を 11 種類持っています。
ソケットサービス機能の TCP の状態は、TCP ソケットステータス読出命令 (SktGetTCPStatus) の出力の TCP コネクションステータスで、確認することができます。
TCP の状態とその意味は下記のとおりです。

TCP の状態	意味
CLOSED	コネクションがクローズされている。
LISTEN	サーバ側が、パッシブオープンでコネクション確立要求 (SYN) を待っている。
SYN SENT	クライアント側が、アクティブオープンでコネクション確立要求 (SYN) を送信し、それに対する確認応答 (SYN+ACK) を待っている。
SYN RECEIVED	サーバ側が、コネクション確立要求 (SYN) に対する確認応答 (SYN+ACK) を送信し、それに対する確認応答 (ACK) を待っている。
ESTABLISHED	コネクションが確立されている。
CLOSE WAIT	サーバ側が、コネクション切断要求 (FIN) に対する確認応答 (ACK) を送信し、サーバのアプリケーションが切断可能になるのを待っている。
FIN WAIT-1	クライアント側が、コネクション切断要求 (FIN) を送信し、それに対する確認応答 (ACK) を待っている。
CLOSING	クライアント側とサーバ側が、同時にコネクション切断要求 (FIN) 受信し、それに対する確認応答 (ACK) を待っている。
LAST-ACK	サーバ側が、コネクション切断要求 (FIN) を送信し、それに対する確認応答 (ACK) を待っている。
FIN WAIT-2	クライアント側が、コネクション切断要求 (FIN) を待っている。
TIME WAIT	クライアント側が、コネクション切断要求 (FIN) に対する確認応答 (ACK) を受信し、サーバに届いて処理されるのを待っている。

また TCP の状態は、相手ノードからの要求 / 応答やユーザプログラムからの TCP ソケットコネクトや TCP ソケットクローズなどの命令で、状態遷移します。
状態遷移時は、コネクションの確立 (SYN)/ 切断 (FIN) 要求や要求に対する確認応答 (ACK) を相手ノードとの間で送受信を行います。

下図は、TCP の状態の遷移を表しています。
図中の□内は TCP の状態を示し、各状態間の矢印に記述されている事柄は、上段が状態遷移が発生する条件を示し下段が状態遷移が発生したときの動作 (未記載の場合は動作なし) を示しています。
例 :「SYN SENT」状態で SYN と ACK を受信すると、ACK を送信して「ESTABLISHED」の状態に遷移する。



I

索引

索引

A

append 10-13

B

BIN 形式 10-19
BOOTP クライアント 1-17
BOOTP サーバ接続異常 3-6, 3-17
bye 10-15

C

cd 10-13
CIDR 機能 4-4
CIPClose 命令 8-5
CIPOpenWithDataSize 命令 8-5
CIPOpen 命令 8-5
CIPRead 命令 8-5
CIPSend 命令 8-5
CIPUCMMRead 命令 8-5
CIPUCMMSend 命令 8-5
CIPUCMMWrite 命令 8-5
CIPWrite 命令 8-5
CIP 通信 1-12
CIP 通信異常 3-4, 3-14
CIP 通信命令の種類 8-5
CIP 通信命令の使用方法 8-6
CIP メッセージ通信サービス機能の概要 8-3
CIP メッセージ通信サービスの機能・性能一覧 8-3
CIP メッセージ通信のクライアント機能 8-4
close 10-15

D

delete 10-15
dir 10-12
DNS サーバ接続異常 3-7, 3-20
DNS 設定 5-3
DNS 設定異常 3-6, 3-17

E

EDS ファイルの管理 A-38
_EIP1_BootpErr
（通信ポート 1BOOTP サーバ接続異常）..... 3-6, 3-18
_EIP1_EtnCfgErr
（通信ポート 1Ethernet 基本設定異常）..... 3-5, 3-16
_EIP1_EtnOnlineSta
（通信ポート 1 オンライン）..... 3-10, 3-20
_EIP1_IPAdrCfgErr
（通信ポート 1IP アドレス設定異常）..... 3-5, 3-16
_EIP1_IPAdrDupErr
（通信ポート 1IP アドレス重複異常）..... 3-6, 3-17
_EIP1_LanHwErr
（通信ポート 1 通信コントローラ異常）..... 3-5, 3-15
_EIP1_MacAdrErr
（通信ポート 1MAC アドレス異常）..... 3-5, 3-15
_EIP1_PortErr（通信ポート 1 異常）..... 3-4, 3-14
_EIP_BootpErr（BOOTP サーバ接続異常）..... 3-6, 3-17
_EIP_CipErr（CIP 通信異常）..... 3-4, 3-14
_EIP_DNSCfgErr（DNS 設定異常）..... 3-6, 3-17
_EIP_DNSSrvErr（DNS サーバ接続異常）..... 3-7, 3-20
_EIP_ErrSta（内蔵 EtherNet/IP 異常）..... 3-3, 3-13
_EIP_EstbTargetSta
（正常ターゲットノード情報）..... 3-11, 3-21, 14-4
_EIP_EtnCfgErr（Ethernet 基本設定異常）..... 3-5, 3-16
_EIP_EtnOnlineSta（オンライン）..... 3-10, 3-20
_EIP_IdentityErr（Identity 情報不正）..... 3-6, 3-18
_EIPIn1_EtnOnlineSta
（内部通信ポート 1 オンライン）..... 3-10, 3-20
_EIPIn1_IPAdrCfgErr
（内部通信ポート 1IP アドレス設定異常）..... 3-5, 3-16
_EIPIn1_IPAdrDupErr
（内部通信ポート 1IP アドレス重複異常）..... 3-6, 3-17
_EIPIn1_PortErr（内部通信ポート 1 異常）..... 3-4, 3-14
_EIP_IPAdrCfgErr（IP アドレス設定異常）..... 3-5, 3-16
_EIP_IPAdrDupErr（IP アドレス重複異常）..... 3-5, 3-17
_EIP_IPRTblErr（IP ルートテーブル異常）..... 3-6, 3-18
_EIP_LanHwErr（通信コントローラ異常）..... 3-5, 3-15
_EIP_MacAdrErr（MAC アドレス異常）..... 3-5, 3-15
_EIP_MultiSwONErr
（複数スイッチ同時 ON 異常）..... 3-7, 3-19
_EIP_NTPResult（NTP の動作情報）..... 3-12
_EIP_NTPResult.ExecNormal
（NTP の動作結果）..... 3-12, 3-22
_EIP_NTPResult.ExecTime
（NTP の最終動作時刻）..... 3-12, 3-22
_EIP_NTPSrvErr（NTP サーバ接続異常）..... 3-7, 3-19
_EIP_PortErr（通信ポート異常）..... 3-3, 3-13
_EIP_RegTargetSta
（登録ターゲットノード情報）..... 3-10, 3-21
_EIP_TagAdrErr（タグ解決異常）..... 3-7, 3-19
_EIP_TargetNodeErr
（ターゲットノード異常情報）..... 3-11, 3-22
_EIP_TargetPLCErr
（ターゲット PLC 異常情報）.... 3-11, 3-22, 7-11, 14-4
_EIP_TargetPLCModeSta
（ターゲット PLC 動作モード）. 3-11, 3-21, 7-11, 14-4
_EIP_TcpAppCfgErr（TCP/IP 設定異常）..... 3-7
_EIP_TcpAppCfgErr
（TCP アプリケーション設定異常）..... 3-19
_EIP_TcpAppErr（TCP アプリ通信異常）..... 3-5, 3-15
_EIP_TDLinkAllRunSta
（全タグデータリンク通信中）..... 3-10, 3-21
_EIP_TDLinkCfgErr
（タグデータリンク設定異常）..... 3-6, 3-18
_EIP_TDLinkErr（タグデータリンク通信異常）3-6, 3-19
_EIP_TDLinkOpnErr

（タグデータリンクコネクション確立失敗）..... 3-6, 3-18
_EIP_TDLinkRunSta
（タグデータリンク通信中）.....3-10, 3-20
_EIP_TDLinkStartCmd（タグデータリンク
通信開始スイッチ）.....3-12, 3-23, 7-66
_EIP_TDLinkStopCmd（タグデータリンク通信停止
スイッチ）.....3-12, 3-23, 7-66
Ethenet Link オブジェクト 8-47
EtherNet/IP 通信のスイッチ関連 3-12
EtherNet/IP 通信のステータス関連 3-10
Ethernet 基本設定異常 3-5, 3-16
Ethernet 情報タブ 14-8
Ethernet 接続コネクタ 2-10, 2-11

F

FTP クライアント 1-18
FTP サーバ 1-17, 5-7
FTP サーバ機能の概要 10-2
FTP サーバ機能の使用手順 10-7
FTP サーバ機能の使用例 10-8
FTP 設定ダイアログ 5-7

G

General Status 8-31
get 10-14

I

Identity 情報不正 3-6, 3-18
Indentity オブジェクト 8-41
IP アドレス 5-3, 5-9, 5-11
IP アドレス設定異常 3-5, 3-16
IP アドレス設定方法 5-2
IP アドレス重複異常 3-5, 3-17
IP アドレスの確認方法 4-10
IP アドレスの構成 4-2
IP アドレスの設定手順 4-6
IP アドレスの割り当て 4-3
IP 通信 1-12
IP ルータテーブルの設定例 5-4
IP ルーティング 1-14
IP ルートテーブル異常 3-6, 3-18

K

KeepAlive 5-3
KeepAlive 監視時間 5-3

L

Linger オプション 5-3
LINK/ACT 1-11
LINK 設定 5-6
ls 10-11

M

MAC アドレス 1-10
MAC アドレス異常 3-5, 3-15
mdelete 10-15
mdir 10-12
mget 10-14
MIB オブジェクトの詳細説明 12-5
MIB グループ一覧 12-4
MIB 体系図 12-4
mkdir 10-12
mls 10-11
mput 10-15

N

NET RUN 1-11
Network Configurator 1-6
　Ethernet による接続方法（NJ CPU ユニット）..7-55
　Ethernet による接続方法 7-52
NTP サーバ接続異常 3-7, 3-19
NTP の最終動作時刻 3-12, 3-22
NTP の動作結果 3-12, 3-22
NTP の動作情報 3-12

O

open 10-11

P

PING コマンド 6-2
PLC オブジェクト 8-51
put 10-14
pwd 10-13

Q

quit 10-16

R

rename 10-12
rmdir 10-13
RPI 7-40
RPI の変更方法 13-11

S

SD メモリカード機能
　ファイルの種類 10-18
　変数データのフォーマット 10-19
SktClearBuf 9-13
SktClose 9-13
SktGetTCPStatus 9-13
SktSetOption 9-13
SktTCPAccept 9-13
SktTCPConnect 9-13

SktTCPRcv 9-13
SktTCPSend 9-13
SktUDPCreate 9-13
SktUDPRcv 9-13
SktUDPSend 9-13
SNMP（エージェント）.....1-20
SNMP（エージェント）機能に必要な設定一覧 12-18
SNMP（エージェント）機能の概要 12-2
SNMP（エージェント）機能の使用手順 12-18
SNMP エージェント機能 12-2
SNMP サービス 5-8
SNMP 仕様 12-3
SNMP 設定ダイアログ 5-8
SNMP トラップ 5-10, 5-12
SNMP トラップ機能 12-3
SNMP トラップ設定ダイアログ 5-10, 5-12
SNMP メッセージ 12-3
Sysmac Studio 1-6

T

TCP/IP Interface オブジェクト 8-44
TCP/IP 設定異常 3-7
TCP/IP 設定ダイアログ 5-2
TCP アプリケーション設定異常 3-19
TCP アプリ通信異常 3-5, 3-15
type 10-14

U

user 10-11

W

Windows ファイアウォールの設定の変更手順 A-43

あ

宛先アドレス 5-4
宛先マスク IP アドレス 5-4

い

異常ステータス 3-9
一斉同報送信 9-10

う

受付認証 1 5-9
受付認証 2 5-9

お

オリジネータ変数 7-39
オンライン 3-10, 3-20

か

外部出力にアクセスする場合の留意事項 A-58

き

機能
　他シリーズとの機能比較 A-2

く

グローバルアドレス 4-11
グローバルブロードキャスト 9-10

け

ゲートウェイアドレス 5-4

こ

コネクション I/O タイプ 7-38, 7-39
コネクション数の計算方法 13-4
コネクション設定
　一括編集 7-39
　個別編集 7-37
　登録デバイス一覧 7-35
コネクションタイプ 7-38, 7-39
コネクションタブ 14-5
コネクションの自動設定 7-42
コネクションのチェック 7-74
コネクション名 7-38, 7-39
コマンド一覧 10-10
コミュニティ名 5-9, 5-11
コントローラステータス 7-11
コントローラのイベント履歴タブ 14-6

さ

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 8-14
サブネットマスク 4-3, 5-2
サンプルプログラム
　CIP メッセージ通信機能 8-15
　ソケットサービス機能 9-18, 9-23
　タグデータリンクでのラダープログラム 7-77

し

システム定義変数 3-2
指定方法 5-9, 5-11
自動コネクション設定 7-43
受信データ処理時間 13-26
出力 TRUE 応答時間 13-26
出力変数の動作とタイミング 8-27, 9-17

す

スイッチングハブ 1-5

機能 2-3
種類 2-3
選定時の留意事項 2-4
ステータス 1 タブ 14-2
ステータス 2 タブ 14-4

せ

正常ターゲットノード情報 3-11, 3-21, 14-4
設置場所 5-8
ゼネラルステータスコード 8-28
全タグデータリンク通信中 3-10, 3-21

そ

送信データ処理時間 13-26
ソケットサービス 1-19
ソケットサービス機能に必要な設定 9-12
ソケットサービス使用上の留意事項 9-29
ソケットサービス通信
TCP による通信手段 9-4
TCP による通信 9-3
UDP による通信 9-3
送信データのフラグメント（分割） 9-5
データの受信処理 9-7
ソケットサービスで使用できるポート No. 9-2
ソケットサービスの使用手順 9-14
ソケットサービス用命令 9-13
ソケットとは 9-2

た

ターゲット PLC 異常情報 3-11, 3-22, 7-11, 14-4
ターゲット PLC 動作モード 3-11, 3-21, 7-11, 14-4
ターゲットデバイス 7-39
ターゲットノード異常情報 3-11, 3-22
ターゲット変数 7-39
代替 DNS サーバ 5-3
タイムアウト値 7-38, 7-40
タグ 1-10, 7-3
タグ／タグセットの作成 7-22
タグ解決異常 3-7, 3-19
タグステータスタブ 14-7
タグセット 7-3
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 7-82
概要 7-2
機能・性能一覧 7-7
設定 7-17
データエリア 7-3
データの同期性 7-13
タグデータリンク（サイクリック通信） 1-12
タグデータリンクコネクション確立失敗 3-18, 3-6
タグデータリンク設定異常 3-6, 3-18
タグデータリンク通信異常 3-6, 3-19
タグデータリンク通信開始スイッチ 3-12, 3-23, 7-66
タグデータリンク通信中 3-10, 3-20
タグデータリンク通信停止スイッチ 3-12, 3-23, 7-66
タグデータリンク入出力最大応答時間 13-25
タグデータリンクによる、タスク周期への影響 13-23
タグデータリンクの開始／停止 7-11
タグデータリンクの自動起動 7-66
タグデータリンクの帯域使用率と RPI の関係 13-9
タグデータリンクの通信のしかた 13-2
タグデータリンクのデバイス帯域使用状況の確認 13-7
タグデータリンクパラメータ
ダウンロード 7-57
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 7-60
特定のデバイスから個別にアップロードする 7-62
タグデータリンクパラメータの照合 7-63
タグデータリンクパラメータの設定／ダウンロード 7-9
タスク周期とパケットインターバル（RPI）の関係 13-23
タスク周期によるパケットインターバル（RPI）の調整 13-24

つ

ツイストペアケーブル 1-5
接続方法 2-10, 2-11
設置時の注意点 2-8, 2-10
布設時の注意点 2-6, 2-9
通信コントローラ異常 3-5, 3-15
通信負荷の調整 13-6
通信ポート 1BOOTP サーバ接続異常 3-6, 3-18
通信ポート 1Ethernet 基本設定異常 3-5, 3-16
通信ポート 1IP アドレス設定異常 3-5, 3-16
通信ポート 1IP アドレス重複異常 3-6, 3-17
通信ポート 1MAC アドレス異常 3-5, 3-15
通信ポート 1 異常 3-4, 3-14
通信ポート 1 オンライン 3-10, 3-20
通信ポート 1 通信コントローラ異常 3-5, 3-15
通信ポート異常 3-3, 3-13

て

データ処理時間の概要 13-22
データ処理時間の計算例 13-23
データ転送のタイミング 13-21
デバイスコネクション構成表示 7-45
デバイスステータス表示 7-76
デバイス帯域使用状況の調整方法 13-10
デバイス単位のタグデータリンクの開始／停止 7-66
デバイスの登録 7-20
デバイスの変更 7-75
デバイスパラメータのクリア 7-68
デバイスパラメータの照合方法 7-64
デフォルトゲートウェイ 5-3

と

登録ターゲットノード情報 3-10, 3-21
ドメイン名 5-3
トラップ 1 5-11
トラップ 2 5-11

な

内蔵 EtherNet/IP 異常 3-3, 3-13
内蔵 EtherNet/IP ポート仕様 1-8
内蔵 EtherNet/IP ポート設定一覧 5-1
内蔵 EtherNet/IP ポートのソケットサービス概要 9-11
内部通信ポート 1IP アドレス設定異常 3-5, 3-16
内部通信ポート 1IP アドレス重複異常 3-6, 3-17
内部通信ポート 1 異常 3-4, 3-14
内部通信ポート 1 オンライン 3-10, 3-20

に

入力 TRUE 応答時間 13-26
認証トラップを送信する 5-8

ね

ネットワーク構成の照合方法 7-63
ネットワーク構成ファイルの保存 7-70
ネットワーク構成ファイルの読出 7-72
ネットワーク伝送遅延時間 13-26
ネットワーク変数 7-8
　Network Configurator にインポートする 7-31

は

バージョン 5-11
パケットインターバル（RPI）.....7-9, 7-38, 13-26
パケットインターバル（RPI）と、
　使用通信帯域幅（PPS）の関係 13-3
パケットインターバル（RPI）の精度 13-5
パケットインターバル（RPI）の設定 13-2
パスワード 5-7

ひ

表示部（LED）.....1-11

ふ

複数スイッチ同時 ON 異常 3-7, 3-19
プライベートアドレス 4-11

へ

変数のメモリ確保
　留意するべきケース例 A-55

ほ

ポート No. 5-7, 5-8, 5-10
ホストコンピュータからの使用例 10-20
ホスト名 5-3, 5-9, 5-11
ホスト名指定 1-20

ま

マルチキャストとユニキャスト 7-10
マルチキャストフィルタ 2-3

め

メッセージサービス機能伝送遅延時間 13-27

ゆ

優先 DNS サーバ 5-3

り

リクエストパス入力のための構造体型変数の用意 8-11

る

ルートパス 8-7

れ

レスポンスコード 8-28
連絡先 5-8

ろ

ローカルブロードキャスト 9-10

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・IP電話などではご利用いただけませんので、右記の電話番号へおかけください。 **055-982-5015**（通話料がかかります）

受付時間：9:00～19:00（12/31～1/3を除く）

オムロンFAクイックチャット
www.fa.omron.co.jp/contact/tech/chat/
技術相談員にチャットでお問い合わせいただけます。（I-Webメンバーズ限定）
受付時間：平日9:00～12:00 / 13:00～17:00（土日祝日・年末年始・当社休業日を除く）
※受付時間、営業日は変更の可能性がございます。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-369H　2024年3月現在

©OMRON Corporation 2016-2024 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