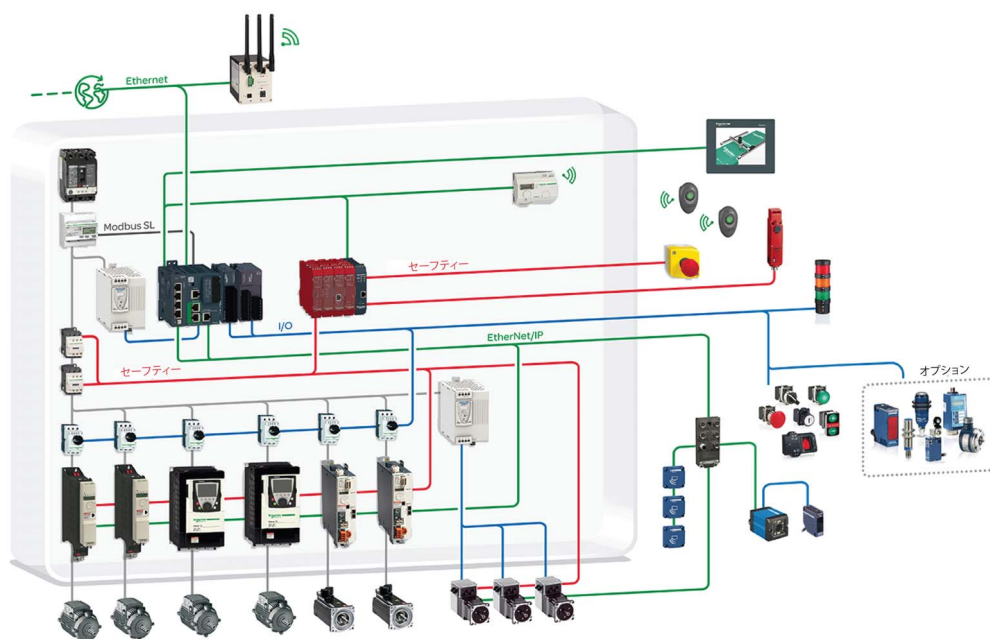


# SoMachine 産業用 Ethernet ユーザーガイド

04/2017



---

本文書内で提供される情報には、本文書内に記載された製品の性能に関する概要および/または技術的特性が含まれます。本文書はこれらの製品の特定の用途における適合性または信頼性の判断の代用となるものではなく、またこれらを判断するためのものではありません。製品に関連する特定の適用または使用の適切かつ完全なリスク分析および評価、試験の実施は、かかるユーザーまたはインテグレーターの義務となります。本文書に記載された情報の誤用について、Schneider Electric またはその関連会社、子会社は一切責任を負わないものとします。本文書に関する改善や修正について提案がある場合、または誤りに気付いた場合は、当社までご連絡ください。

Schneider Electric からの書面による許可がない限り、電子的または機械的を問わず（複写を含む）、どのような形式や手段であっても本文書のいかなる部分も複製してはなりません。

本製品を設置および使用する際は、該当する州/地域/地方の安全規定に従ってください。安全上の理由から、また立証済みのシステムデータに確実に準拠するため、コンポーネントの修理は必ずメーカーが行ってください。

本製品を技術的安全条件を伴う用途に使用する場合、関連する指示内容に従う必要があります。

Schneider Electric のハードウェア製品に当社製または当社認定ではないソフトウェアを使用すると、傷害または損害、不適切な操作結果につながる場合があります。

この情報を順守しないと、傷害または機器の損害につながる場合があります。

© 2017 Schneider Electric. All rights reserved.



	安全に関する使用上の注意	5
	本書について	7
<b>第 1 章</b>	<b>産業用 Ethernet の概要</b>	<b>11</b>
	概要	12
	アーキテクチャー	13
	原理	14
	ロジックコントローラー	16
	対応デバイス	18
	設定手順の概要	20
<b>第 2 章</b>	<b>デバイスネットワークの設定</b>	<b>21</b>
2.1	ネットワークの計画	22
	ネットワークの計画	22
2.2	IP アドレスの割り当て	24
	IP アドレスの割り当て	25
	IP アドレスの指定方式	27
	産業用 Ethernet マネージャーの設定	28
2.3	ネットワークデバイスの宣言	29
	ネットワークデバイスの宣言	29
2.4	ネットワーク設定とデバイス識別の適応	31
	ネットワーク設定とデバイス識別の適応	32
	EtherNet/IP Target (スレーブ) デバイスの設定	35
	Modbus TCP 設定	36
2.5	ネットワークデバイスの設定	37
	ネットワークデバイスの設定	37
2.6	ネットワークデバイスの交換	39
	FDR によるデバイス交換	40
	ユーザーパラメーターによるデバイス交換	41
2.7	サイクリックデータ交換の設定	43
	サイクリックデータ交換の概要	44
	EtherNet/IP サイクリックデータ交換の設定	45
	EtherNet/IP I/O マッピング	55
	Modbus TCP サイクリックデータ交換の設定	57
	Modbus TCP I/O マッピング	60
	産業用 Ethernet マネージャーの負荷検証	62
2.8	産業用 Ethernet でのプログラミング	64
	産業用 Ethernet でのプログラミング	64
<b>第 3 章</b>	<b>デバイスネットワークの試運転</b>	<b>67</b>
	試運転	68
	デバイスを認識させるための準備	70
	正しいデバイス設定の適用	72
<b>第 4 章</b>	<b>デバイスネットワークの運用</b>	<b>73</b>
	スレーブデバイスの動作モードの管理	74
	要求によるデータ交換	76
	カスタムサイクリックデータ交換	77
	起動時のスレーブデバイスの設定	78
	アウトオブプロセスのデータ交換	79
	産業用 Ethernet マネージャーの動作モード	81
	セキュリティ	84

<b>第 5 章</b>	<b>デバイスネットワーク診断</b> . . . . .	<b>85</b>
	ネットワークテスト . . . . .	86
	診断 : Web サーバー . . . . .	87
	診断 : SoMachine オンラインモード . . . . .	90
	トラブルシューティング . . . . .	93
<b>第 6 章</b>	<b>保守</b> . . . . .	<b>95</b>
	保守の概要 . . . . .	95
	<b>付録</b> . . . . .	<b>97</b>
<b>付録 A</b>	<b>Modbus Serial IOScanner ライブラリ</b> . . . . .	<b>99</b>
A.1	Modbus Serial IOScanner ファンクション . . . . .	100
	IOS_GETSTATE: Modbus Serial IOScanner の状態の読み込み . . . . .	101
	IOS_START: Modbus Serial IOScanner の起動 . . . . .	102
	IOS_GETHEALTH: ヘルスビット値の読み込み . . . . .	103
	IOS_STOP: Modbus Serial IOScanner の停止 . . . . .	104
	CONFIGURE_OTB: Advantys OTB のソフトウェア設定の送信 . . . . .	105
A.2	Modbus Serial IOScanner データ型 . . . . .	107
	IoStateCodes: Modbus Serial IOScanner の状態値 . . . . .	108
	CommunicationErrorCodes: 検出エラーコード . . . . .	109
	configurationOTBErrorCodes: OTB 設定の検出エラーコード . . . . .	110
<b>付録 B</b>	<b>EtherNet/IP Explicit メッセージ通信ライブラリ</b> . . . . .	<b>111</b>
B.1	EtherNet/IP Explicit メッセージ通信ファンクション . . . . .	112
	Get_Attribute_All: オブジェクトのすべての属性の取得 . . . . .	113
	Set_Attribute_All: インスタンスまたはクラスのすべての属性の設定 . . . . .	115
	Get_Attribute_Single: オブジェクトの属性の取得 . . . . .	117
	Set_Attribute_Single: オブジェクトの属性の設定 . . . . .	119
	EIPStartConnection: 接続の開始 . . . . .	121
	EIPStartAllConnection: すべての接続の開始 . . . . .	122
	EIPStopConnection: 接続の停止 . . . . .	123
	EIPStopAllConnections: すべての接続の停止 . . . . .	124
	EIPGetHealthBit: ヘルスビット値の取得 . . . . .	125
	デバイスのマニュアルでのオブジェクト情報の探し方 . . . . .	126
B.2	EIP Explicit メッセージ通信のデータ型 . . . . .	127
	CommunicationErrorCodes: 通信エラーコード . . . . .	128
	OperationErrorCodes: 動作エラーコード . . . . .	129
<b>付録 C</b>	<b>EtherNet/IP Scanner ライブラリ</b> . . . . .	<b>133</b>
C.1	EtherNet/IP Scanner ファンクション . . . . .	134
	EipControl: EtherNet/IP Scanner の制御 . . . . .	135
	EipGetHealth: ヘルスビットの読み込み . . . . .	136
	EipDataExch: Explicit メッセージの送信 . . . . .	137
C.2	EtherNet/IP Scanner データ型 . . . . .	140
	CommunicationErrorCodes: 通信エラーコード . . . . .	141
	OperationErrorCodes: 動作エラーコード . . . . .	142
	TCP_ADDR: TCP デバイスのアドレス . . . . .	143
<b>付録 D</b>	<b>モーション制御ライブラリ</b> . . . . .	<b>145</b>
	モーション制御ライブラリ . . . . .	145
<b>付録 E</b>	<b>汎用 TCP/UDP ライブラリ</b> . . . . .	<b>147</b>
	汎用 TCP/UDP ライブラリ . . . . .	147
<b>付録 F</b>	<b>ファンクションおよびファンクションブロックの表現</b> . . . . .	<b>149</b>
	ファンクションとファンクションブロックの相違 . . . . .	150
	IL 言語でのファンクションおよびファンクションブロックの使用方法 . . . . .	151
	ST 言語でのファンクションおよびファンクションブロックの使用方法 . . . . .	154
<b>用語集</b>	. . . . .	<b>157</b>
<b>索引</b>	. . . . .	<b>161</b>

# 安全に関する使用上の注意



## 重要情報

### お断り

本書をよくお読みいただき、装置の正しい取り扱いと機能を十分ご理解いただいた上で、設置、操作、保守を行ってください。本書および装置には以下の表示が使われています。これらは潜在的な危険を警告したり、手順を明確化あるいは簡素化する情報について注意を呼びかけるものです。



この記号が「危険」または「警告」安全ラベルに追加されると、電気的な危険が存在し、指示に従わないと人身傷害の危険があることを示します。



安全警告記号です。人的傷害の危険性があることを警告します。  
この記号の後に記載された安全に関する情報に従って、人的傷害や死亡の危険性を回避してください。

### ⚠ 危険

危険は、危険が生じる可能性のある状況を示します。回避しないと、死亡や重傷を招きます。

### ⚠ 警告

警告は、危険が生じる可能性のある状況を示します。回避しないと、死亡や重傷を招くおそれがあります。

### ⚠ 注意

注意は、危険が生じる可能性のある状況を示します。回避しないと、軽傷を招くおそれがあります。

### 注記

この表示は、指示に従わないと物的損害を負う可能性があることを示します。

### 注意

電子機器の設置、操作、整備は必ず資格のある人物が行ってください。Schneider Electric は、本資料の使用に起因するいかなる結果についても責任を負わないものとします。

資格のある人物とは、電子機器の構造、操作、設置に関する技術および知識を有し、かつ電子機器に伴う危険性を理解しこれを回避するための安全研修を受けた人物を指します。

---

# 本書について



## 概要

### 本書の適用範囲

本書は、次の用途にご使用ください。

- 産業用 Ethernet ネットワークの計画と構成
- 産業用 Ethernet ネットワークの設置と構築
- 産業用 Ethernet ネットワークの運用
- 産業用 Ethernet ネットワークの診断と保守

**注記：**コントローラーの取り付け、使用またはメンテナンスの前に、本書ならびに全ての関連マニュアルをよくお読みいただきご理解いただきますようお願いいたします。

### 有効性に関する注意

本書は、SoMachine V4.3 のリリース時に更新されました。

### 関連マニュアル

マニュアルタイトル	参照番号
Modicon M241 ロジックコントローラー - プログラミングガイド	<a href="#">EIO0000001432 (ENG)</a> <a href="#">EIO0000001433 (FRE)</a> <a href="#">EIO0000001434 (GER)</a> <a href="#">EIO0000001435 (SPA)</a> <a href="#">EIO0000001436 (ITA)</a> <a href="#">EIO0000001437 (CHS)</a>
Modicon M251 ロジックコントローラー - プログラミングガイド	<a href="#">EIO0000001462 (ENG)</a> <a href="#">EIO0000001463 (FRE)</a> <a href="#">EIO0000001464 (GER)</a> <a href="#">EIO0000001465 (SPA)</a> <a href="#">EIO0000001466 (ITA)</a> <a href="#">EIO0000001467 (CHS)</a>
Modicon TM4 Expansion Modules - Programming Guide	<a href="#">EIO0000001802 (ENG)</a> <a href="#">EIO0000001803 (FRE)</a> <a href="#">EIO0000001804 (GER)</a> <a href="#">EIO0000001805 (SPA)</a> <a href="#">EIO0000001806 (ITA)</a> <a href="#">EIO0000001807 (CHS)</a>
SoMachine - プログラミングガイド	<a href="#">EIO0000000067 (ENG)</a> <a href="#">EIO0000000069 (FRE)</a> <a href="#">EIO0000000068 (GER)</a> <a href="#">EIO0000000071 (SPA)</a> <a href="#">EIO0000000070 (ITA)</a> <a href="#">EIO0000000072 (CHS)</a>
Motion Control Library Guide	<a href="#">EIO0000002221 (ENG)</a> <a href="#">EIO0000002222 (GER)</a> <a href="#">EIO0000002223 (CHS)</a>
TcpUdpCommunication Library Guide	<a href="#">EIO0000002204 (ENG)</a> <a href="#">EIO0000002255 (FRE)</a> <a href="#">EIO0000002205 (GER)</a> <a href="#">EIO0000002257 (SPA)</a> <a href="#">EIO0000002256 (ITA)</a> <a href="#">EIO0000002258 (CHS)</a>
Distributed Modbus TCP Logic Controller M251 - System User Guide	<a href="#">EIO0000001680 (ENG)</a>
Compact EtherNet/IP Logic Controller M251 - System User Guide	<a href="#">EIO0000002183 (ENG)</a>

マニュアルや技術情報は（株）シュナイダーエレクトリックウェブサイトからダウンロードできます。  
<https://www.schneider-electric.com/en/download>

**⚠ 警告**

**制御不能**

- 制御手法の設計者は制御パスの障害モードが発生するおそれを考慮する必要があり、特定の重要制御機能については、バス障害の最中および終了後に安全な状態を実現するための方策を準備しておく必要があります。重要制御機能の例としては、緊急停止、オーバートラベル停止、停電、および再起動があります。
- 重要な制御機能に対しては、別のまたは冗長性のある制御バスを用意してください。
- システム制御バスには、データ通信が含まれることがあります。予期しないデータの転送遅れや障害について考慮する必要があります。
- あらゆる事故防止規制および地域の安全性ガイドライン<sup>1</sup>を遵守してください。
- 運用を開始する前に、各実装について、正しく動作するかどうかを個別に十分にテストする必要があります。

**上記の指示に従わないと、死亡、重傷、または物的損害を負う可能性があります。**

<sup>1</sup> 詳細は、NEMA ICS 1.1 (最新版)、“Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control”、および NEMA ICS 7.1 (最新版)、“Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems”、または該当地域での同等のガイドラインを参照してください。

**⚠ 警告**

**装置の意図しない動作**

- 本装置には、Schneider Electric 認定のソフトウェアのみ使用してください。
- ハードウェアの設定を変更した場合は、必ずアプリケーションプログラムも更新してください。

**上記の指示に従わないと、死亡、重傷、または物的損害を負う可能性があります。**

規格から派生した用語

技術用語、専門用語、シンボル、本書の記述、また本製品での表示は、国際規格用語および定義に由来しています。

安全機能システム、ドライブ、一般オートメーションにおいて、用語は、**安全性、安全機能、安全状態、異常、異常リセット、誤動作、障害、エラー、エラーメッセージ、危険**等を含みますが、それに限定されません。

特に以下の規格が含まれます。

規格	詳細
EN 61131-2: 2007	プログラマブルコントローラー、第 2 部：機器要件、および試験
ISO 13849-1: 2008	機械類の安全性：制御システムの安全関連部 設計の一般原則
EN 61496-1: 2013	機械類の安全性：電氣的検知保護装置 第 1 部：一般要件、および試験
ISO 12100: 2010	機械類の安全性 - 設計の一般原則 - リスク評価とリスク低減
EN 60204-1: 2006	機械類の安全性 - 機械の電気装置 - 第 1 部：一般要件
EN 1088: 2008 ISO 14119: 2013	機械類の安全性 - ガードと共同するインターロック装置 - 設計、および選択のための原則
ISO 13850: 2006	機械類の安全性 - 非常停止 - 設計原則
EN/IEC 62061: 2005	機械類の安全性 - 安全関連の電気・電子・プログラマブル電子制御システムの機能安全
IEC 61508-1: 2010	電気・電子・プログラマブル電子安全関連系の機能安全：一般要求事項
IEC 61508-2: 2010	電気・電子・プログラマブル電子安全関連系の機能安全：電気・電子・プログラマブル電子安全関連系に対する要求事項
IEC 61508-3: 2010	電気・電子・プログラマブル電子安全関連系の機能安全：ソフトウェア要求事項
IEC 61784-3: 2008	計測制御用デジタルデータ通信：機能安全フィールドバス



規格	詳細
2006/42/EC	機械指令
2014/30/EU	電磁両立性指令
2014/35/EU	低電圧指令

本書で使われている用語には下記の規格も含まれています。

規格	詳細
IEC 60034 シリーズ	回転電気機械
IEC 61800 シリーズ	可変速電気駆動システム
IEC 61158 シリーズ	計測制御用デジタルデータ通信 – 産業制御システム用のフィールドバス

*動作領域*は特定の危険性記述と併せて使われる場合があり、*機械指令 (2006/42/EC)* と *ISO 12100 : 2010 の危険区域* と同様に定義されています。

**注記**：前述の規格は、本書記載の特定の機器には適用されない場合があります。本書に記載されている製品の適用規格についての詳細は製品の特徴が記載された表を参照してください。



---

# 第 1 章

## 産業用 Ethernet の概要

---

### この章について

この章には次の項目が含まれています。

項目	参照ページ
概要	12
アーキテクチャー	13
原理	14
ロジックコントローラー	16
対応デバイス	18
設定手順の概要	20

## 概要

### 概要

産業用 Ethernet とは、標準 Ethernet の物理層を使用する産業用プロトコルを指します。

産業用 Ethernet ネットワークでは、以下を接続できます。

- 産業用デバイス (産業用プロトコル)
- 非産業用デバイス (その他の Ethernet プロトコル)

本書における産業用 Ethernet は以下が含まれます。

- EtherNet/IP
- Modbus TCP
- TCP/UDP

本書は、ロジックコントローラーのデバイスネットワークに接続された産業用 Ethernet デバイスに重点を置いています。

### 産業用 Ethernet の機能

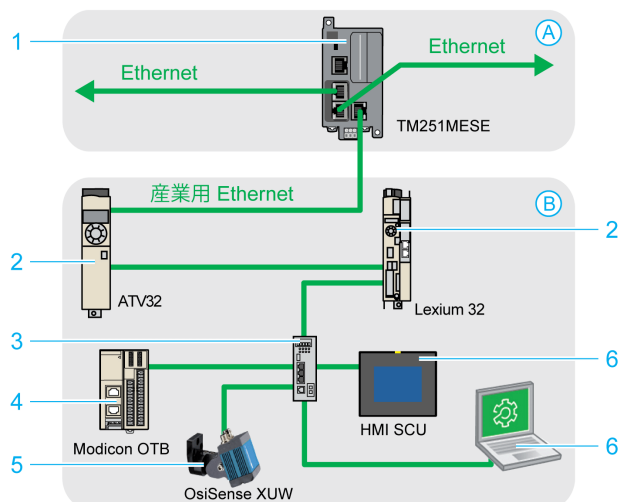
産業用 Ethernet の実装機能の概要を次の表に示します。

機能	産業用 Ethernet の性能
デバイス数およびネットワークの適応性	スター型構成 事実上無制限のインフラストラクチャー 優れた総ネットワーク距離 高度なネットワーク管理
データ速度、配線、距離	システム内で光ファイバーと銅線ケーブルの混在が可能。 10/100 Mbit/s、銅線ケーブルでは最長 100 m (328 ft)、光ファイバーケーブルでは最長 2000 m (6561 ft)。 ノイズ耐性
プロトコル	Ethernet を基にした多様なプロトコル

## アーキテクチャー

### 産業用 Ethernet のアーキテクチャー

この図は一般的な産業用 Ethernet のアーキテクチャーを示しています。



- A** 制御ネットワーク
- B** デバイスネットワーク
- 1 ロジックコントローラー (16 ページ参照)
- 2 デイジーチェーン接続されたデバイス
- 3 Ethernet スイッチ
- 4 リモート I/O (Modbus TCP)
- 5 視覚センサー (EtherNet/IP)
- 6 PC および HMI (TCP/UDP)
- 2、4、5 産業用 Ethernet デバイス (EtherNet/IP / Modbus TCP)

このアーキテクチャーは、SoMachine で設定可能です。

## 原理

### 概要

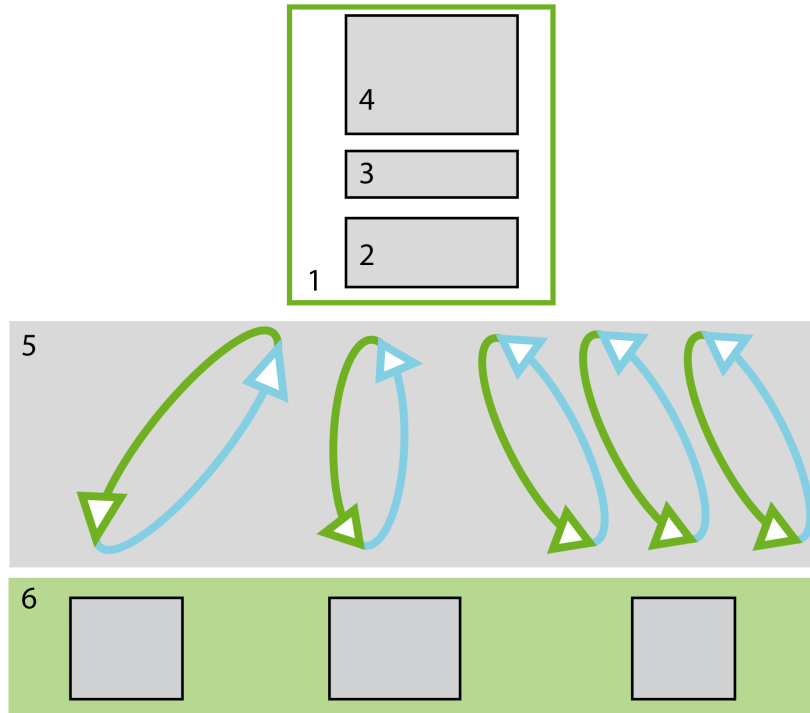
ロジックコントローラーは、産業用 Ethernet の動作モード管理を制御します。この管理は、安定した周期的なデータ交換（スキャナーサービス）を使用して行われます。

スキャナーサービスは、以下のプロトコルで使用できます。

- EtherNet/IP (15 ページ参照)
- Modbus TCP (15 ページ参照)

### スキャナーの原理

産業用 Ethernet スキャナーの原理



- 1 ロジックコントローラー (16 ページ参照)
- 2 I/O イメージ
- 3 アプリケーションインターフェイス
- 4 アプリケーション
- 5 Modbus チャンネルまたは EtherNet/IP 接続でのデータ交換
- 6 スレーブデバイス (18 ページ参照)

### データ交換

ロジックコントローラーは、各対応プロトコルごとに以下を管理します。

- サイクリックデータ交換
- 非サイクリックデータ交換

サイクリックデータ交換 (EtherNet/IP の Implicit メッセージ) は、以下のような一定周期でデータを交換する必要がある場合に使用します。

- I/O モジュールのスキャン
- インバーターの更新
- センサーの入力データの読み込み

非サイクリックデータ交換 (EtherNet/IP の Explicit メッセージ) は、対象のデバイスから、以下のようなオンデマンド情報を取得するために使用します。

- 設定
- 診断
- 収集データ

## EtherNet/IP の概要

EtherNet/IP は、標準 Ethernet の CIP プロトコルに実装されています。

EtherNet/IP プロトコルでは、Originator (マスター)/Target (スレーブ) アーキテクチャーを使用してデータを交換します。

**Originator (マスター)** は、ネットワーク上の Target (スレーブ) デバイスとのデータ交換を開始するデバイスです。これは、I/O 通信およびメッセージ通信サービスの両方に適用されます。これは、Modbus ネットワークのクライアントの役割と同じです。

**Target (スレーブ)** は Originator (マスター) で生成されたデータ要求に応答するデバイスです。これは、I/O 通信およびメッセージ通信サービスの両方に適用されます。これは、Modbus ネットワークのサーバーの役割と同じです。

**EtherNet/IP アダプター** は、EtherNet/IP ネットワークの終端デバイスです。I/O ブロックおよびドライブは、EtherNet/IP アダプターデバイスにもなります。

EtherNet/IP の Originator (マスター) と Target (スレーブ) の間の通信は、EtherNet/IP 接続 (45 ページ) を使用して行われます。

## Modbus TCP の概要

Modbus TCP プロトコルは、クライアント/サーバーアーキテクチャーを使用してデータを交換します。

Modbus TCP の Explicit (非サイクリック) データ交換は、アプリケーションで管理されます。

Modbus TCP の Implicit (サイクリック) データ交換は、Modbus Serial I/O Scanner で管理されます。

Modbus Serial I/O Scanner は、Ethernet を基にしたサービスで、データ、状態、および診断情報を交換するためにスレーブデバイスを継続的にポーリングします。この処理で入力を監視し、スレーブデバイスの出力を制御します。

**クライアント** は、ネットワーク上の他のデバイスとのデータ交換を開始するデバイスです。これは、I/O 通信およびメッセージ通信サービスの両方に適用されます。

**サーバー** は、クライアントで生成されたデータ要求を処理するデバイスです。これは、I/O 通信およびメッセージ通信サービスの両方に適用されます。

Modbus Serial I/O Scanner とスレーブデバイス間の通信は、Modbus TCP チャンネル (57 ページ) を使用して行われます。

## ロジックコントローラー

### ロジックコントローラー

産業用 Ethernet に対応しているロジックコントローラーを次の表に示します。

パラメーター		TM251MESE、TM241CE24*、TM241CE40*、TM241CEC24*
産業用 Ethernet	トポロジー	スイッチを介したデジチェーンおよびスター型
	帯域幅	10/100 Mbit/s
EtherNet/IP Scanner	性能	ロジックコントローラーで最大 16 台の EtherNet/IP Target (スレーブ) デバイスを管理。タイムスロット 10 ms 以内で監視
	接続数	0...16
	入力ワード数	0...1024
	出力ワード数	0...1024
	I/O 通信	EtherNet/IP Scanner サービス 設定および転送用ファンクションブロック Originator (マスター) / Target (スレーブ)
Modbus Serial IOScanner	性能	ロジックコントローラーで最大 64 台の Modbus TCP スレーブデバイスを管理。タイムスロット 64 ms 以内で監視
	チャンネル数	0...64
	入力ワード数	0...2048
	出力ワード数	0...2048
その他のサービス	I/O 通信	Modbus Serial IOScanner サービス データ転送用ファンクションブロック マスター / スレーブ
	FDT/DTM/EDS 管理	
	FDR (高速デバイス交換)	
	DHCP サーバー	
	セキュリティ管理 (セキュリティパラメーター およびファイアウォール設定を参照)	
	Modbus TCP サーバー	
	Modbus TCP クライアント	
	EtherNet/IP アダプター (EtherNet/IP の Target (スレーブ) としてのコントローラー)	
	EtherNet/IP Originator (マスター)	
	Modbus TCP サーバー (Modbus TCP のスレーブとしてのコントローラー)	
	Web サーバー	
	FTP サーバー	
	SNMP	
IEC VAR ACCESS		
その他の機能	最大 16 台の EtherNet/IP および Modbus TCP デバイスを混在可能。デバイスは、設定、監視、および管理するために直接アクセスが可能。 制御ネットワークとデバイスネットワーク間のネットワーク透過性 (ロジックコントローラーはゲートウェイとしても使用可)。 <b>注記:</b> ロジックコントローラーをゲートウェイとして使用する場合、ロジックコントローラーのパフォーマンスに影響を与える可能性があります。	

**注記:** スキャナーの入出力ワードの制限は、デバイスネットワーク上のデバイス数に影響します。例えば、TM251MESE は最大 4 台の OsiSense XUW デバイスしか接続できません。産業用 Ethernet マネージャーの負荷検証 (62 ページ) を参照してください。



## 産業用 Ethernet ポート

産業用 Ethernet ポートを設定するには、以下の手順に従います。

1. **デバイスツリー** の以下のノードをダブルクリックします。
  - TM241CE24・TM241CE40: **MyController** → **Ethernet\_1**
  - M251 ロジックコントローラ: **MyController** → **Ethernet\_2**
2. ネットワーク設定 (IP アドレス、サブネットマスク、ゲートウェイアドレス) を設定し、DHCP アドレス指定を使用している場合は DHCP サーバーを有効にします。

## 対応デバイス

### 対応デバイス

対応している産業用 Ethernet デバイスを次の表に示します。

デバイス名		対応プロトコル			TVDA	主要機能
		TCP/UDP	Modbus TCP	EtherNet/IP		
定義済みデバイス	Altivar 32	-	X	X	X	FDR、DTM、ライブラリ、定義済み接続、定義済みデータ交換
	Altivar 320	-	X	X	X	FDR、DTM、ライブラリ、定義済み接続、定義済みデータ交換
	Altivar 340	-	X	X	X	FDR、DTM、ライブラリ、定義済み接続、定義済みデータ交換
	Altivar 6**	-	X	X	X	FDR、DTM、ライブラリ、定義済み接続、定義済みデータ交換
	Altivar 71	-	X	X	X	FDR、DTM、ライブラリ、定義済み接続、定義済みデータ交換
	Altivar 9**	-	X	X	X	FDR、DTM、ライブラリ、定義済み接続、定義済みデータ交換
	Lexium 32 M	-	X	X	X	FDR、DTM、ライブラリ、定義済み接続、定義済みデータ交換
	Lexium ILA	-	X	X	X	FDR、ライブラリ、定義済み接続、定義済みデータ交換
	Lexium ILE	-	X	X	X	FDR、ライブラリ、定義済み接続、定義済みデータ交換
	Lexium ILS	-	X	X	X	FDR、ライブラリ、定義済み接続、定義済みデータ交換
	OsiSense XG	-	X	X	X	定義済み接続、定義済みデータ交換
	OsiSense XUW	-	-	X	X	定義済み接続、定義済みデータ交換
	OTB1EODM9LP	-	X	-	X	ライブラリ、定義済み接続、定義済みデータ交換
	XPSMCM	-	(1)	X	X	定義済み接続、定義済みデータ交換
Harmony XB4R/5R	-	X	-	-	DTM、ライブラリ、定義済み接続、定義済みデータ交換	
その他のデバイス	EDS ファイル付きデバイス <sup>(2)</sup>	-	-	X	-	ユーザーパラメーター、定義済み接続
	汎用スレーブデバイス <sup>(3)</sup>	X	X	X	-	ユーザーパラメーター (EtherNet/IP のみ)、ライブラリ

(1) デバイスを SoMachine に汎用スレーブデバイスとして追加する場合、Modbus TCP スレーブデバイスとして宣言できます。

(2) EDS ファイルは、特にデバイスネットワークの統合を容易にするための定義済み接続を提供します。

(3) SoMachine で汎用スレーブデバイスは、インバーター、センサー、またはその他のコントローラー (Modbus TCP、EtherNet/IP、または TCP/UDP デバイスなど) を追加するために使用します。

### 主要機能

主要機能を次の表に示します。

主要機能	説明
FDR	高速デバイス交換：デバイス設定がロジックコントローラーに保存されます。デバイスを交換すると、新しいデバイスに自動的に設定が読み込まれます。
DTM	DTM に対応しているデバイスの場合：FDT/DTM 技術により、SoMachine にネットワークデバイスを設定できます。Device Type Manager User Guide を参照してください。
ライブラリ	アプリケーションで使用するファンクション/ファンクションブロック (デバイス専用) が利用できます。

主要機能	説明
定義済み接続	サイクリックデータ交換を設定するために使用します。提示される関連情報を含む接続から 1 つ選択します。詳細については、サイクリックデータ交換 (45 ページ) を参照してください。
定義済みデータ交換	サイクリックデータ交換が自動的に設定されます。デバイスをプロジェクトに追加すると、定義済みの接続が 1 つ自動的に選択されます。
ユーザーパラメーター	電源投入時にデバイスに自動的に送信されるパラメーターです。このパラメーターは、FDR に対応していないデバイスを交換する場合に使用します。

## TVDA

以下の TVDA (Tested Validated Documented Architecture、動作検証済み、評価済み、文書化済みのアーキテクチャー) システムユーザーガイドは、産業用 Ethernet と関連があります。

- Distributed Modbus TCP Logic Controller M251
- Compact EtherNet/IP Logic Controller M251

対応している産業用 Ethernet デバイス (18 ページ) の一部には、SoMachine プロジェクトにインバーターまたはサーボドライブなどのデバイスを統合するためのアプリケーションコードテンプレート (デバイスモジュール) が付いています。デバイスモジュールはファンクションテンプレートで実行されます。SoMachine 内で定義済みアプリケーションプログラムの内容呼び出すメカニズムです。

各デバイスモジュールはフィールドデバイスを制御するために SoMachine アプリケーションの内容を埋め込み、その状態を監視し、検出されたエラーを処理します。それには、SoMachine オートメーションプロジェクト全体のデバイス機能にアクセスするためのインターフェイスを提供する個別のグローバル変数定義が含まれています。

詳細については、TVDA Device Module Library, Function Template Library Guide を参照してください。

## 設定手順の概要

### 概要

本書は、機器のライフサイクルのフェーズに従って構成しています。

以下の章には、ユースケースシステムを構築するための情報と手順が記述されています。

- デバイスネットワークの設定 (21 ページ参照)
- デバイスネットワークの試運転 (67 ページ参照)
- デバイスネットワークの運用 (73 ページ参照)
- デバイスネットワークの診断 (85 ページ参照)
- デバイスネットワークの保守 (95 ページ参照)

---

## 第 2 章

### デバイスネットワークの設定

---

#### 概要

この章では、デバイスネットワークの設定に必要な情報と手順について説明します。  
デバイスネットワークの設定は、SoMachine で設定します。  
このフェーズの最後に、デバイスネットワークの試運転 (67 ページ) を実行できます。

#### この章について

この章には次のセクションが含まれています。

セクション	項目	参照ページ
2.1	ネットワークの計画	22
2.2	IP アドレスの割り当て	24
2.3	ネットワークデバイスの宣言	29
2.4	ネットワーク設定とデバイス識別の適応	31
2.5	ネットワークデバイスの設定	37
2.6	ネットワークデバイスの交換	39
2.7	サイクリックデータ交換の設定	43
2.8	産業用 Ethernet でのプログラミング	64

## 2.1 ネットワークの計画

### ネットワークの計画

#### 目的

計画的なネットワークによって設置効率が向上し、設置時間とコストが削減されます。ネットワークを計画するために、事前にインターフェイス (スイッチ、ケーブル、ポート) を設計してください。

#### ネットワークの設計

産業用 Ethernet ネットワークを設計および計画するには、ODVA の *Media Planning and Installation Manual* などの対応ドキュメントを参照してください。このマニュアルは、[ODVA ウェブサイト](#) からダウンロードできます。

#### スイッチの種類

ネットワークの要件に応じて、適切なスイッチを使用します。

要件	使用するスイッチ
ネットワークの診断および動作情報	管理機能付きスイッチ
物理的な接続損失の際の通信確保	冗長化スイッチ
長距離ネットワーク (光ファイバー)	2 連 SC コネクタ付きスイッチ

ハブによって利用可能な帯域幅が減少する場合があります。そのために要求が損失したり、デバイスが管理されなくなる可能性があります。

### 注記

#### データの損失

ハブを使用して産業用 Ethernet ネットワークを構築しないでください。

**上記の指示に従わないと、物的損害を負う可能性があります。**

スイッチの詳細については、*Essential Guide: Networks, connectivity and Web servers* を参照してください。

#### ケーブルの種類

ネットワークで使用できるケーブルの型式を次の表に示します。

一般的な設置では、次のケーブルを使用できます。

型式	説明	詳細	長さ
490NTW000**	DTE 接続 用 Ethernet シールドケーブル	DTE 用両端 RJ45 コネクタ付き の通常ケーブル CE 準拠	2、5、12、40 または 80 m (6.56、16.4、39.37、 131.23 または 262.47 ft)
490NTW000**U		DTE 用両端 RJ45 コネクタ付き の通常ケーブル UL 準拠	2、5、12、40 または 80 m (6.56、16.4、39.37、 131.23 または 262.47 ft)
TCSECE3M3M**S4		両端 RJ45 コネクタ付きの過酷な 環境用ケーブル CE 準拠	1、2、3、5 または 10 m (3.28、6.56、9.84、 16.4 または 32.81 ft)
TCSECU3M3M**S4		両端 RJ45 コネクタ付きの過酷な 環境用ケーブル UL 準拠	1、2、3、5 または 10 m (3.28、6.56、9.84、 16.4 または 32.81 ft)
TCSECL1M1M**S2**		過酷な環境用ケーブル M12 コネクタ 2 個 CE 準拠	1、3、10、25 または 40 m (3.28、9.84、32.8、 82.02 または 131.23 ft)
TCSECL1M3M**S2**		過酷な環境用ケーブル M12 コネクタ 1 個 RJ45 コネクタ 1 個 CE 準拠	1、3、10、25 または 40 m (3.28、9.84、32.8、 82.02 または 131.23 ft)

光ファイバーネットワークでは、次のケーブルを使用できます。

型式	説明	詳細	長さ
490NOC00005	DTE 接続用ガラ ス製光ファイ バーケーブル	SC コネクタ 1 個 MT-RJ コネクタ 1 個	5 m (16.4 ft)
490NOT00005		ST コネクタ (BFOC) 1 個 MT-RJ コネクタ 1 個	5 m (16.4 ft)
490NOR00003		MT-RJ コネクタ 2 個	3 m (9.8 ft)
490NOR00005		MT-RJ コネクタ 2 個	5 m (16.4 ft)

## 2.2 IP アドレスの割り当て

---

### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
IP アドレスの割り当て	25
IP アドレスの指定方式	27
産業用 Ethernet マネージャーの設定	28



## IP アドレスの割り当て

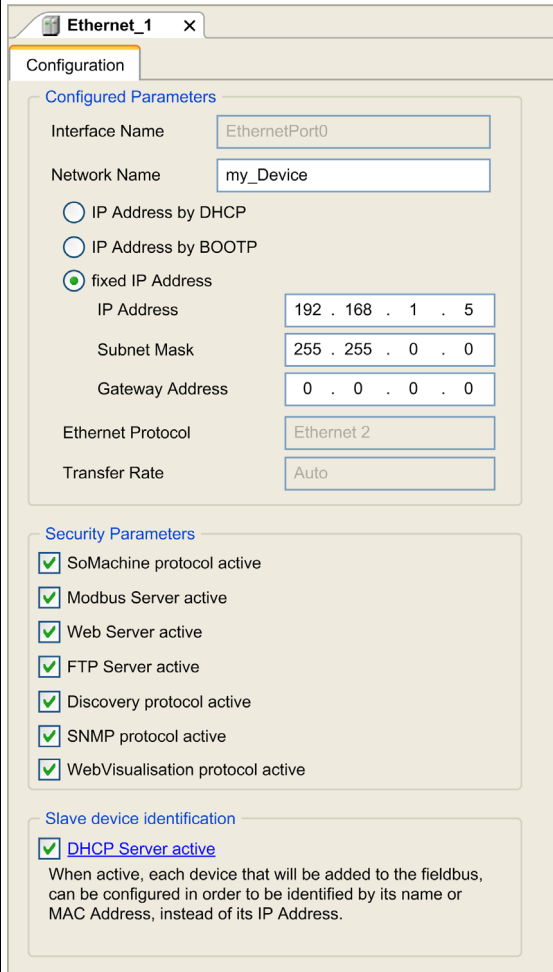
### 概要

このセクションでは、ネットワークデバイスに IP アドレスを割り当てる手順を説明します。

- コントローラーの産業用 Ethernet ポート (17 ページ) の設定
  - ネットワーク設定: IP アドレス、サブネットマスク、ゲートウェイアドレス
  - 使用する IP アドレスの指定方式 (27 ページ) の選択
- 産業用 Ethernet マネージャーの設定

### 産業用 Ethernet ポートの設定

産業用 Ethernet ポート (17 ページ) を設定するには、以下の手順で行います。

手順	手順内容
1	<p>デバイスツリーの産業用 Ethernet ポートノードをダブルクリックします。次のような設定タブが表示されます。</p> 
2	<b>固定 IP アドレス</b> を選択します。
3	<b>IP アドレス</b> を設定します。0.0.0.0 以外にしてください。 この IP アドレスは、ネットワークマネージャー (32 ページ) で使用します。
4	<b>サブネットマスク</b> を設定します。
5	<b>ゲートウェイアドレス</b> がデフォルトで 0.0.0.0 に設定されていることを確認します。 ゲートウェイアドレスによって、メッセージをローカルのネットワーク上にないデバイスへルーティングできます。 ゲートウェイがない場合、ゲートウェイアドレスは 0.0.0.0 です。
6	<p><b>セキュリティパラメーター</b> チェックボックスを選択します。</p> <ul style="list-style-type: none"> <li>● <b>Web サーバーが有効</b>: 設定および保守フェーズ時に使用します。</li> <li>● <b>FTP サーバーが有効</b>: FDR サービス (40 ページ) で使用します。</li> </ul>

手順	手順内容
7	DHCP サーバーを使用して IP アドレスを割り当てる場合は、 <b>DHCP サーバーが有効</b> チェックボックスを選択します。 詳細については、IP アドレスの指定方式 (27 ページ) を参照してください。

## IP アドレスの指定方式

### 概要

IP アドレスの指定方式を次の表に示します。

方式	説明	詳細
DHCP	DHCP サーバーは、デバイスの <b>DHCP デバイス名</b> を使用してデバイスの IP アドレスを送信します。 <b>DHCP デバイス名</b> は、FDR サービスでも使用されます。	新しいデバイスは、デフォルトで DHCP アドレス指定方式が使用されます。 DHCP を使用することで、FDR サービスを利用できます。 デバイスを交換するには次を行います。 <ul style="list-style-type: none"> <li>● 新しいデバイスを設置</li> <li>● デバイスに DHCP デバイス名を定義</li> <li>● デバイスの電源を入れ、アプリケーションを起動</li> </ul> 電源投入時に新しいデバイスが認識され、ロジックコントローラーによって以前に保存された設定が新しいデバイスに読み込まれます。
BOOTP	BOOTP サーバーは、デバイスの <b>MAC アドレス</b> を使用してデバイスの IP アドレスを送ります。	デバイスを交換するには次を行います。 <ul style="list-style-type: none"> <li>● 新しいデバイスを設置</li> <li>● SoMachine に、新しいデバイスの MAC アドレスを入力</li> <li>● アプリケーションを構築し、ロジックコントローラーに読み込み</li> <li>● デバイスのパラメーターを設定</li> <li>● デバイスの電源を入れ、アプリケーションを起動</li> </ul>
固定	アプリケーションで IP アドレスが固定されます。	デバイスを交換するには次を行います。 <ul style="list-style-type: none"> <li>● 新しいデバイスを設置</li> <li>● デバイスのネットワーク設定 (IP アドレス、サブネットマスク、ゲートウェイアドレス) を設定</li> <li>● 直接、または SoMachine を使用してデバイスのパラメーターを設定</li> <li>● デバイスの電源を入れ、アプリケーションを起動</li> </ul>

### DHCP サーバーの有効化

DHCP アドレス指定方式を使用すると、要求に応じて DHCP サーバーがデバイスに IP アドレスを割り当てます。

DHCP サーバーを有効にするには、以下の手順で行います。

手順	手順内容
1	<b>デバイスツリー</b> の産業用 Ethernet ポート ( <a href="#">17 ページ</a> ) ノードをダブルクリックします。
2	<b>DHCP サーバーが有効</b> チェックボックスを選択します。 有効にすると、フィールドバスに追加したデバイスを MAC アドレスまたは固定 IP アドレスの代わりに DHCP デバイス名で識別されるように設定できます。

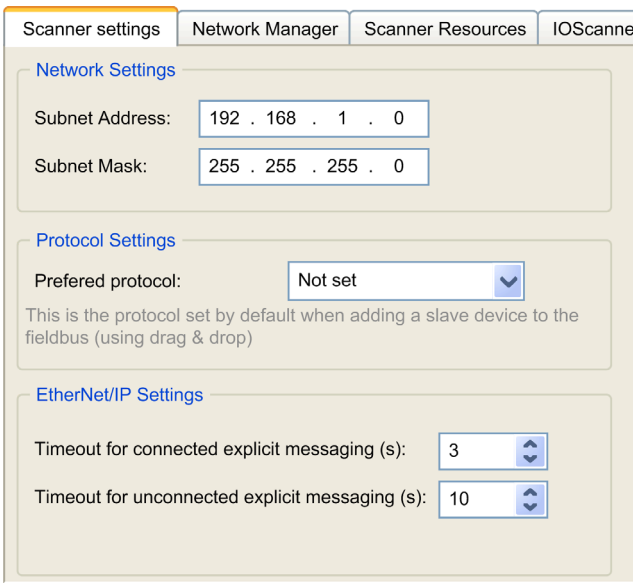
## 産業用 Ethernet マネージャーの設定

### 概要

ロジックコントローラーは、産業用 Ethernet マネージャーを使用してデバイスネットワークを管理します。

### 産業用 Ethernet マネージャー設定

産業用 Ethernet マネージャーを設定するには、以下の手順で行います。

手順	手順内容
1	<p>デバイスツリーの産業用 Ethernet ポート (17 ページ) ノードの上の <b>Industrial_Ethernet_Manager</b> をダブルクリックします。</p> <p><b>結果:</b></p>  <p><b>注記:</b> ネットワーク設定が、産業用 Ethernet ポートの ネットワーク設定 (25 ページ) に応じて自動的に生成されます。</p>
2	<p>優先プロトコルを選択します。</p> <ul style="list-style-type: none"> <li>● 未設定</li> <li>● EtherNet/IP (デフォルト)</li> <li>● Modbus TCP</li> </ul> <p>この選択肢が、各デバイス宣言 (29 ページ) においてデフォルトで設定されるデバイスプロトコルになります。</p>
3	<p><b>EtherNet/IP 設定</b>で、Explicit メッセージ通信のタイムアウトの値を指定します。</p>

**注記:** Modbus Serial IOScanner が設定されている場合、産業用 Ethernet ネットワーク用のポスト設定ファイルは無視されます。

## 2.3 ネットワークデバイスの宣言

### ネットワークデバイスの宣言

#### 概要

このセクションでは、**Industrial\_Ethernet\_Manager** ノードにデバイスを追加する方法について説明します。

使用できる Schneider Electric デバイス、および EDS ファイル付きデバイスは、**ハードウェアカタログ**に一覧表示されます。これらのデバイスには、定義済み接続設定 (18 ページ) が付いています。カタログに記載されていないその他のデバイスについては、**汎用スレーブデバイス**を使用してください。

#### 自動設定

各デバイスの宣言中、SoMachine は自動的に以下を行います。

- 産業用 Ethernet スキャナー設定に応じてネットワーク設定 (IP アドレス、サブネットマスク、ゲートウェイアドレス) を設定します。
- 固有の DHCP デバイス名を設定します。デバイス名は通常、デバイスの内部規則と互換性があります。(各 **DHCP デバイス名**は固有にしてください。)
- 定義済みデバイス用の定義済みデータ交換を作成します。

**注記**：設定された **DHCP デバイス名**がデバイスと互換性がない場合は、名前を変更できます。

#### デバイスの追加

**Industrial\_Ethernet\_Manager** ノードにデバイスを追加するには、**ハードウェアカタログ**で選択したデバイスを**デバイスツリー**へドラッグし、産業用 Ethernet ポート (17 ページ) ノード上でドロップします。

**注記**：デバイスを産業用 Ethernet ポートノードに追加すると、自動的に **Industrial\_Ethernet\_manager** ノードが追加されます。

デバイスが追加されると、**Industrial\_Ethernet\_Manager** の**ネットワークマネージャ**タブにデバイスが表示されます。ネットワーク設定とデバイス識別の適応 (32 ページ) を参照してください。

ドラッグ & ドロップで追加すると、可能である限り優先プロトコルでデバイスが定義されます。

プロジェクトへのデバイスの追加の詳細については、以下を参照してください。

- ドラッグ & ドロップの使用 (SoMachine, プログラミングガイド 参照)
- コンテキストメニューまたはプラスボタンの使用 (SoMachine, プログラミングガイド 参照)

#### 優先プロトコル以外のプロトコルでデバイスを追加

ドラッグ & ドロップで追加する場合

- デバイスを優先プロトコルで定義できない場合は、代わりにデバイスで対応しているデフォルトのプロトコルが使用されます。
- 優先プロトコルが設定されていない場合は、使用するプロトコルを選択するリストが表示されます。

優先プロトコル以外のプロトコルでスレーブデバイスを追加する場合は、コンテキストメニューまたはプラスボタンの使用 (SoMachine, プログラミングガイド 参照) を参照してください。

例えば、OTB1EODM9LP デバイスを追加すると、優先プロトコルが EtherNet/IP に設定されていても、Modbus TCP で設定されます。

#### テンプレートからデバイスを追加

主要機能はないが TVDA (18 ページ) に対応しているデバイスは、テンプレートを使用して宣言できません。これにより、プログラムの記述を容易にする追加要素がインポートされます。

OsiSense XGCS、XUW、および Preventa XPSMCM デバイスにはこの方法を使用します。

テンプレートから **Industrial\_Ethernet\_Manager** にデバイスを追加するには、以下の手順で行います。

手順	手順内容
1	ハードウェアカタログの <b>デバイス</b> テンプレートチェックボックスを選択します。
2	ハードウェアカタログで選択したデバイスを <b>デバイス</b> ツリーへドラッグし、産業用 Ethernet ポート (17 ページ) ノード上でドロップします。

プロジェクトへのデバイスの追加の詳細については、以下を参照してください。

- ドラッグ & ドロップの使用 (SoMachine, プログラミングガイド 参照)
- コンテキストメニューまたはプラスボタンの使用 (SoMachine, プログラミングガイド 参照)

### TCP/UDP デバイスの追加

**Industrial\_Ethernet\_Manager** ノードに TCP/UDP デバイスを追加するには、**ハードウェア**カタログで選択した**汎用 TCP/UDP 機器**を**デバイス**ツリーへドラッグし、産業用 Ethernet ポート (17 ページ) ノード上でドロップします。

### EDS ファイルからデバイスを追加

サードパーティのデバイスには、EDS ファイルが付属されています。

**Industrial\_Ethernet\_Manager** に EDS ファイル付きデバイスを追加するには、以下の手順で行います。

手順	手順内容
1	SoMachine メニューから、 <b>ツール</b> → <b>Device Repository</b> を選択します。
2	<b>インストール</b> をクリックして、 <b>Install Device Description</b> ダイアログボックスを開きます。
3	ファイル種類の一覧から <b>EDS and DCF files</b> を選択します。
4	EDS ファイルを選択します。
5	<b>OK</b> をクリックしてダイアログボックスを閉じます。
6	<b>閉じる</b> をクリックして、 <b>Install Device Description</b> ダイアログボックスを閉じます。
7	<b>Industrial_Ethernet_Manager</b> を選択して <b>プラス</b> ボタンをクリックします。新しく追加されたスレーブデバイスを選択して、 <b>デバイスの追加</b> をクリックします。詳細については、 <b>コンテキスト</b> メニューまたは <b>プラス</b> ボタンの使用 (SoMachine, プログラミングガイド 参照) を参照してください。

## 2.4

### ネットワーク設定とデバイス識別の適応

#### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
ネットワーク設定とデバイス識別の適応	32
EtherNet/IP Target (スレーブ) デバイスの設定	35
Modbus TCP 設定	36

## ネットワーク設定とデバイス識別の適応

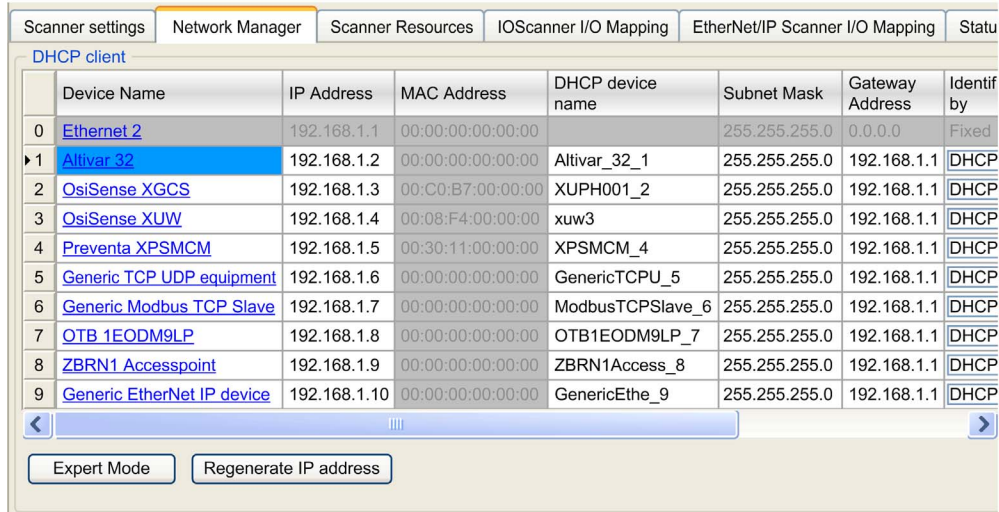
### 概要


Industrial\_Ethernet\_Manager にデバイスを追加したら、ネットワークマネージャータブを使用してネットワーク設定を編集します。

### ネットワークマネージャー

デバイスツリーの Industrial\_Ethernet\_Manager ノードをダブルクリックします。

ネットワークマネージャータブに、デバイスネットワークに定義されているデバイスが表示されます。



欄	用途	コメント
デバイス名	クリックするとデバイス設定が開きます	デバイス名 デフォルトの名前が付けられます。デバイス名を変更するには、名前ボックスに名前を入力してください。名前にはスペースを使用しないでください。名前の末尾にアンダースコア (_) を使用しないでください。プロジェクトの構成を容易にするために、デバイスに意味のある名前を付けてください。
IP アドレス	IP アドレスの変更	IP アドレスが、すでに同じプロトコルで DHCP アドレス割り当てを使用する別のデバイスに割り当てられている場合、無効として表示されます。 IP アドレスが無効な場合、アイコン  が表示されます。
MAC アドレス	MAC アドレスの入力	BOOTP を使用して IP アドレスを取得するために使用します。 各 IP アドレスは、特定のプロトコルおよび DHCP/BOOTP に対して固有にしてください。例えば、Modbus TCP および Ethernet/IP プロトコルの両方に同じデバイスを追加することはできませんが、一方のプロトコルの IP アドレスを BOOTP または DHCP を使用して取得する場合、他方のプロトコルにも固定 IP アドレスとして同じ IP アドレスを入力してください。
DHCP デバイス名	DHCP デバイス名の変更	DHCP を使用して IP アドレスを取得するためのデバイス名として使用します (最大 16 文字)。 DHCP デバイス名は、デバイスで定義した名前と同じにしてください。 各 DHCP デバイス名は固有にしてください。 デフォルトの DHCP デバイス名は、通常、デバイスの内部規則と互換性があります。 DHCP デバイス名の内部規則の詳細については、デバイスのマニュアルを参照してください。 <b>注記</b> ：設定された DHCP デバイス名がデバイスと互換性がない場合は、名前を変更できます。



欄	用途	コメント
サブネットマスク	サブネットマスクの変更	エキスパートモードをクリックすると、サブネットマスクの欄が表示 / 非表示されます。
ゲートウェイアドレス	ゲートウェイアドレスの変更	エキスパートモードをクリックすると、ゲートウェイアドレスの欄が表示 / 非表示されます。 操作の詳細については、アウトオブプロセスのデータ交換 (79 ページ) を参照してください。
識別方式	IP アドレスの指定方式の変更 ● DHCP ● BOOTP ● 固定	DHCP: DHCP デバイス名は、デバイスで定義した名前と同じにしてください。 FDR サービスでは、この方式が必須です。
		BOOTP: デバイスの MAC アドレスを入力してください。
		固定: IP アドレスは、デバイスで定義したアドレスと同じにしてください。
デバイスタイプ	-	デバイスのタイプ
プロトコル	-	使用するプロトコル

このタブで行った変更は、対応するデバイス設定タブに適用されます。

- EtherNet/IP Target (スレーブ) デバイスの設定 (35 ページ参照)
- Modbus TCP 設定 (36 ページ参照)

### IP アドレスの指定方式

デフォルトでは、追加したデバイスには DHCP が使用されます。

IP アドレスの指定方式を次の表に示します。

方式	説明	詳細
DHCP	DHCP サーバーは、デバイスの DHCP デバイス名を使用してデバイスの IP アドレスを送信します。 DHCP デバイス名は、FDR サービスでも使用されます。	DHCP を使用することで、FDR サービスを利用できます。 デバイスを交換するには、以下を行います。 ● 新しいデバイスを設置 ● デバイスに DHCP デバイス名を定義 ● デバイスの電源を入れ、アプリケーションを起動  電源投入時に新しいデバイスが認識され、ロジックコントローラーによって以前に保存された設定が新しいデバイスに読み込まれます。
BOOTP	BOOTP サーバーは、デバイスの MAC アドレスを使用してデバイスの IP アドレスを送信します。	デバイスを交換するには、以下を行います。 ● 新しいデバイスを設置 ● SoMachine に、新しいデバイスの MAC アドレスを入力 ● アプリケーションを構築し、ロジックコントローラーに読み込み ● デバイスのパラメーターを設定 ● デバイスの電源を入れ、アプリケーションを起動
固定	アプリケーションで IP アドレスが固定されます。	デバイスを交換するには、以下を行います。 ● 新しいデバイスを設置 ● デバイスのネットワーク設定 (IP アドレス、サブネットマスク、ゲートウェイアドレス) を設定 ● 直接、または SoMachine を使用してデバイスのパラメーターを設定 ● デバイスの電源を入れ、アプリケーションを起動

### IP アドレス設定の再初期化

Regenerate IP address をクリックすると、産業用 Ethernet ポート (17 ページ) に関連する IP アドレス設定が再初期化されます (例えば、産業用 Ethernet ポートの IP アドレスの変更後など)。

SoMachine は、産業用 Ethernet ポート (17 ページ) に設定されている IP アドレスを読み込み、使用できる次の IP アドレスをデバイスに割り当てます。例えば、産業用 Ethernet ポート (17 ページ) で設定された IP アドレスが 192.168.0.11 である場合、デバイスに指定される IP アドレスは 192.168.0.12、192.168.0.13、と続きます。

### アウトオブプロセスのデータ交換

制御ネットワークとデバイスネットワーク間のデータ交換は、アウトオブプロセスのデータ交換である場合があります。例えば、デバイスネットワーク上の対象のデバイスと通信するために、監視ソフトウェアまたはサードパーティの設定ツールを使用する場合があります。

操作の詳細については、アウトオブプロセスのデータ交換 (79 ページ) を参照してください。

アウトオブプロセスのデータ交換が必要な場合は、デバイスの正しいゲートウェイアドレスパラメータを設定します。

ロジックコントローラーの産業用 Ethernet ポート (17 ページ) の IP アドレスにネットワークデバイスのゲートウェイアドレスパラメータを設定してください。

パラメータを設定するために、設定ツールがネットワークデバイスと通信できるようにしてください。

設定ツールの状態	結果
制御ネットワークに接続されている	ネットワークデバイスのゲートウェイパラメータを更新してください (以下を参照)
デバイスネットワークに接続されている	ゲートウェイパラメータは使用しません
TCP/IP 以外のプロトコルを使用する	ゲートウェイパラメータは使用しません

ネットワークデバイスのゲートウェイパラメータを設定するには、デバイスのマニュアルを参照してください。

**注記：** DHCP サービスを使用してネットワークデバイスのアドレス指定をする場合、ゲートウェイパラメータはロジックコントローラーのネットワークマネージャータブ (32 ページ) で設定します。

## EtherNet/IP Target (スレーブ) デバイスの設定

### 概要

**Industrial\_Ethernet\_Manager** にデバイスを追加したら、**表示器設定**タブを使用してネットワーク設定を編集します。

### EtherNet/IP Target (スレーブ) デバイスの設定

デバイスツリーの EtherNet/IP デバイスノードをダブルクリックします。

アドレス設定の値は、**Industrial\_Ethernet\_Manager** で定義されている値と同じです。ネットワーク設定とデバイス識別の適応 (32 ページ) を参照してください。

### 電子キー

電子キー署名は、デバイスの識別に使用します。

電子キーは、デバイスのファームウェアに含まれている情報です (メーカーコード、製品コードなど)。スキャナーが起動すると、選択した電子キーの値とデバイスの対応する情報が比較されます。

デバイスの値がアプリケーションの値と同じでない場合、ロジックコントローラーはデバイスと通信しません。

設定済みデバイスの場合、電子キーの値は変更できません。

汎用 EtherNet/IP デバイスの場合、電子キーの値を変更できます。

電子キーの値については、デバイスのマニュアルにあるアイデンティティオブジェクト (F1 hex) の説明を参照してください。

## Modbus TCP 設定

### 概要

**Industrial\_Ethernet\_Manager** にデバイスを追加したら、**ネットワークマネージャー**タブを使用してネットワーク設定を編集します。

### Modbus TCP の設定

Modbus Serial IOScanner に追加した定義済みスレーブデバイスを設定するには、以下の手順で行います。

手順	手順内容
1	<p>デバイスツリーの Modbus TCP スレーブデバイスノードをダブルクリックします。  <b>結果</b>：設定ウィンドウが表示されます。</p> 
2	<p><b>スレーブ IP アドレス</b>の値を入力します。  <b>アドレス設定</b>の値は、<b>Industrial_Ethernet_Manager</b> (<a href="#">32ページ</a>) で定義されている値と同じです。</p>
3	<p><b>ヘルスタイムアウト (ms)</b> の値を入力します ( デフォルトでは 1000)。                  これは、Modbus Serial IOScanner の要求とスレーブからの応答の間の遅延 (ms) を表します。                  ヘルスタイムアウトの時間が切れると、関連付けられたヘルスビット値が 0 に変わります。                  ヘルスビット値は、<b>IOScanner I/O マッピング</b>タブ (<a href="#">60ページ</a>) または Web サーバーから可視化できます。                  ヘルスタイムアウトは、スレーブデバイスのチャンネルに適用されます。</p>
4	<p>高度な設定のできるデバイスの場合、一部の追加設定が必要な場合があります。</p> <ul style="list-style-type: none"> <li>● <b>繰り返し率 (ms)</b>: ms 単位で表される時間の値。これは、2 回の要求送信の間の遅延を表します。この値は、<b>ヘルスタイムアウト (ms)</b> より小さくしてください。</li> <li>● <b>ユニット ID</b>: Modbus TCP スレーブデバイスのユニット ID ( デフォルトでは 255)。</li> </ul> <p>Device Type Manager User Guide を参照してください。</p>

## 2.5 ネットワークデバイスの設定

### ネットワークデバイスの設定

#### 概要

デバイスネットワーク上にネットワークデバイスを定義すると、以下を設定できます。

- ユーザーパラメーター
- DTM
- プラグイン
- サードパーティのツール

説明	利点
ユーザーパラメーター	EtherNet/IP デバイスで利用可能。 ユーザーパラメーターは、デバイスの交換に使用できます。 デバイスとの通信が開始されるたびに、ユーザーパラメーターがデバイスに書き込まれます。
DTM	複雑な設定を管理可能。
プラグイン	優れた透過性。 特別に SoMachine 用に設計。
サードパーティのツール	デバイス用に特別に設計されたツール。

#### ユーザーパラメーター

ユーザーパラメーター (41 ページ) を参照してください。

#### DTM 付きデバイス

一部のデバイスには DTM が付いています。対応デバイス (18 ページ) を参照してください。

DTM を使用してデバイスのパラメーターを変更できます。

DTM を使用してデバイスを設定するには、以下の手順で行います。

手順	手順内容
1	デバイスツリーのデバイスをダブルクリックします。
2	デバイスの設定タブを選択します。
3	OK をクリックします。 結果：DTM によって、タブの内容が更新されます。
4	デバイス設定を変更します。 詳細については、Device Type Manager User Guide を参照してください。

**注記：** DTM を使用する場合、ロジックコントローラーで特定のルーティングおよび IP 転送 (EcoStruxure Machine Expert, Device Type Manager (DTM), User Guide 参照) の設定が必要な場合があります。

#### プラグイン付きデバイス

プラグインによっては、ユーザーパラメーターが使用できない場合があります。その場合、デバイスの設定の管理はプラグインによります。

**例：** Advantys OTB1EODM9LP

SoMachine では Advantys OTB1EODM9LP は、ライブラリで対応しています。ファンクションブロックの 1 つが設定ファンクションブロックで、これによりデバイスに設定を送信できます。詳細については、Distributed Modbus TCP Logic Controller M251 System User Guide を参照してください。

OTB1EODM9LP を設定するには、以下の手順で行います。

手順	手順内容
1	デバイスツリーの OTB1EODM9LP ノードをダブルクリックします。
2	OTB I/O 設定タブで Advantys OTB デバイスの I/O を設定します。
3	OTB に取り付けられている TM2 拡張モジュールを追加および設定します。
4	前の手順で作成したデータで Advantys OTB 設定を更新するために、CONFIGURE_OTB ファンクションブロックを呼び出します。

**注記：**産業用 Ethernet スキャナーでは、カウンター、高速カウンター、パルス出力のような Advantys OTB の高度な機能は直接使用できません。

### サードパーティのツール

一部のデバイスは、SoMachine 外で設定します (特定のソフトウェア、キーパッド、Web サーバーなど)。詳細については、デバイスのマニュアルを参照してください。

### マスター IP アドレスパラメーター

一部のデバイスには**マスター IP アドレスパラメーター**があり、宣言された 1 台のマスターロジックコントローラーのみがデバイスにアクセスできます。

デバイスの状態	結果
産業用 Ethernet マネージャーを使用するように設定されている	デバイス内で <b>マスター IP アドレスパラメーター</b> を設定します (以下を参照)。
産業用 Ethernet マネージャーを使用するように設定されていない	0.0.0.0 をデバイスの <b>マスター IP アドレスパラメーター</b> として使用します。

デバイスの**マスター IP アドレスパラメーター**を、産業用 Ethernet マネージャーに対応しているロジックコントローラーの IP アドレスに設定してください (産業用 Ethernet ポート (17 ページ) で設定)。

デバイスにこのパラメーターを設定するには、デバイスのマニュアルを参照してください。

## 2.6 ネットワークデバイスの交換

### 概要

デバイスの交換は、次の方法で行います。

- FDR サービス
- ユーザーパラメーター

### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
FDR によるデバイス交換	40
ユーザーパラメーターによるデバイス交換	41

## FDR によるデバイス交換

### FDR の概要

一部のデバイスは、高速デバイス交換 (Fast Device Replacement : FDR) サービスに対応しています。

FDR サービスによって、ネットワーク上のデバイスのネットワークパラメーターおよび動作パラメーターが保存されます。デバイスを交換すると、交換したデバイスがこのサービスによって自動的に削除されたデバイスと同じパラメーターで設定されます。

このサービスをデバイスに設定するには、デバイスのマニュアルを参照してください。

FDR サーバーは、ロジックコントローラーに内蔵されている 2 つのアドバンストサービスに依存しています。

- デバイスのアドレス割り当て用 DHCP サーバー
- デバイスのパラメーターファイル用 FTP サーバー。このオプションのサービスは、パラメーターを含むデバイスでのみ使用されます。

DHCP サーバーによって、新しいデバイスを同じアドレス指定のパラメーターで設定できます。

パラメーターを含むデバイスは、FTP サーバーを使用してパラメーターファイルを保存します。

デバイスが交換されると、FTP サーバーにパラメーターファイルの復元を要求します。



## ユーザーパラメーターによるデバイス交換

### 概要

FDR サービスに対応していない EtherNet/IP デバイスの場合、デバイス交換を容易にするためのユーザーパラメーターを設定できます。このユーザーパラメーターは、以下の後、スキャナー接続が開始される直前にデバイスに送信されます。

- アプリケーションのダウンロード
- ウォームスタート/コールドスタートのリセット
- 手動による接続の開始

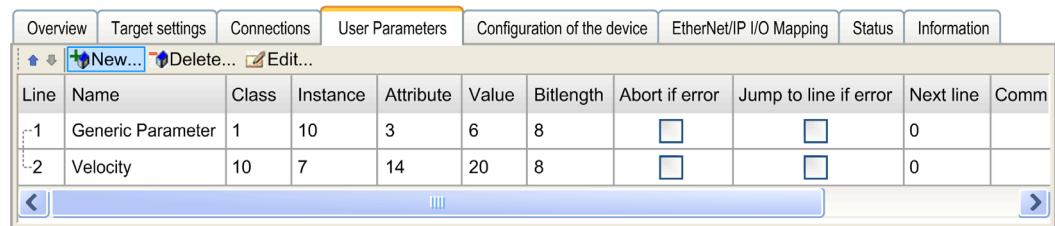
一部の EtherNet/IP デバイスには、あらかじめ定義されたユーザーパラメーターがあります。

ユーザーパラメータータブで、他のパラメーターを追加および管理できます。

管理の詳細については、正しいデバイス設定の適用 (72 ページ) を参照してください。

### ユーザーパラメーター

デバイスツリーの EtherNet/IP デバイスをダブルクリックし、ユーザーパラメータータブを選択します。



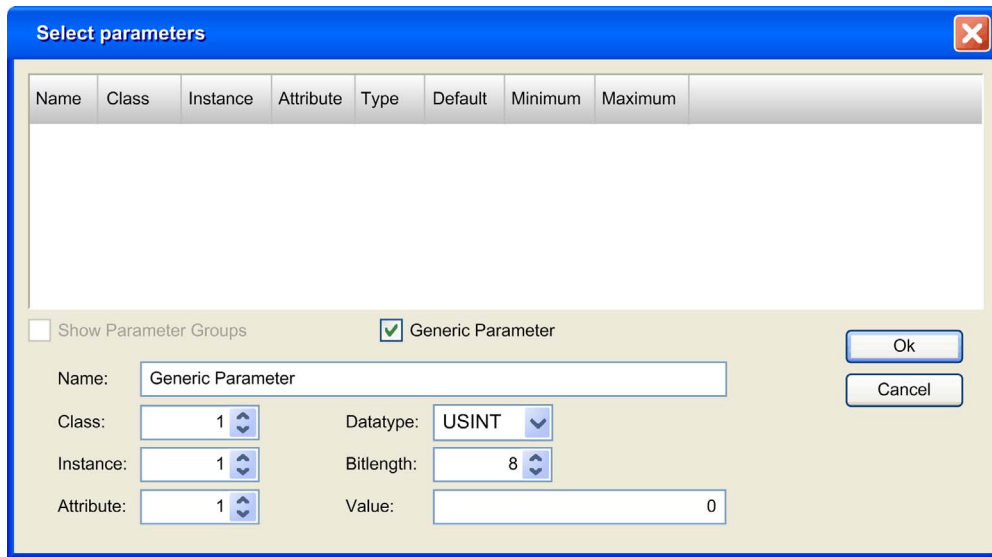
欄	説明
行	行番号。 デバイスに読み込まれたパラメーターの順序を示します。
名前	パラメーター名。
クラス	オブジェクトに対応するクラスのクラス ID <sup>(1)</sup> 。
インスタンス	オブジェクトに対応するインスタンスのインスタンス ID <sup>(1)</sup> 。
属性	オブジェクトに対応する属性の属性 ID <sup>(1)</sup> 。
値	パラメーターの値。 値を変更するには、ダブルクリックします。該当する場合、入力可能な値のリストが開きます。
ビット長	パラメーターのビット数。 選択したパラメーターのデータ型に応じて自動的に変更します。
エラーの場合は中止	選択すると、エラーが検出されたときにパラメーターの送信が中止されます。
エラーの場合は行にジャンプ	選択すると、エラーが検出されたときにプログラムが次の行で指定した行から再開します。初期化中にブロックをスキップしたり、リターンを定義できます。 <b>注記：</b> 特定のパラメーターの書き込みが不可能な場合、リターンは無限ループを引き起こす可能性があります。
次の行	ダブルクリックしてジャンプ先の行を入力します (エラーの場合は行にジャンプが選択されている場合)。
コメント	ダブルクリックしてコメントを入力します。

<sup>(1)</sup> クラス ID、インスタンス ID、および属性 ID は、デバイスマニュアルに記載されています。ユーザーパラメーター情報の探し方 (42 ページ) を参照してください。

アイコン	説明
上に移動	パラメーターリストで選択したパラメーターを上に移動させます。
下に移動	パラメーターリストで選択したパラメーターを下に移動させます。
新規	新しいパラメーターを作成します。
削除	選択したパラメーターを削除します。
編集	選択したパラメーターを編集します。

**ユーザーパラメーターの作成または設定**

新規をクリック、またはパラメーターを選択して編集をクリックします。



フィールド	説明
名前	パラメーター名。
クラス	オブジェクトタイプに対応するクラスのクラス ID <sup>(1)</sup> 。
インスタンス	クラスの実装に対応するインスタンスのインスタンス ID <sup>(1)</sup> 。
属性	インスタンスの特性に対応する属性の属性 ID <sup>(1)</sup> 。
データ型	使用可能なデータ型の一覧。
ビット長	パラメーターのビット数。 選択したの <b>データタイプ</b> に応じて自動的に変更します。
値	パラメーターの値。

<sup>(1)</sup> クラス ID、インスタンス ID、および属性 ID は、デバイスマニュアルに記載されています。ユーザーパラメーター情報の探し方 (42 ページ) を参照してください。

**ユーザーパラメーター情報の探し方**

設定可能なユーザーパラメーター情報は、デバイスのマニュアルに記載されています。これは通常、アプリケーションオブジェクト、Explicit メッセージ通信、または EtherNet/IP カテゴリー 3 に属するオブジェクトの説明の一部です。

ユーザーパラメーターの書き込みアクセスは、通常、ユーザーパラメーターが属するクラスまたはインスタンスに対して指定されます。書き込み処理は、一般的に Set\_Attribute\_Single または Write one attribute と呼ばれるサービスを使用して実行されます。代わりに、サービス識別子 0x10 (16 進数) または 16 (10 進数) に対応している場合があります。

ユーザーパラメーターには、必ず以下の数値プロパティがあります。

- クラスまたはクラス ID、通常 16 進数の値
- インスタンスまたはインスタンス ID、通常 16 進数の値
- 属性または属性 ID、通常 16 進数の値

また、ユーザーパラメーターには 3 組の 10 進数 (xx/yy/zz) または 3 組の 16 進数 (16#xx/yy/zz) 形式の識別子がある場合もあります。

## 2.7

### サイクリックデータ交換の設定

#### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
サイクリックデータ交換の概要	44
EtherNet/IP サイクリックデータ交換の設定	45
EtherNet/IP I/O マッピング	55
Modbus TCP サイクリックデータ交換の設定	57
Modbus TCP I/O マッピング	60
産業用 Ethernet マネージャーの負荷検証	62

## サイクリックデータ交換の概要

### 概要

産業用 Ethernet マネージャーは、ロジックコントローラーとスレーブデバイス間のサイクリックデータ交換 (Implicit メッセージ通信) に対応しています。

サイクリックデータ交換の要求は、以下によって行われます。

- EtherNet/IP では接続
- Modbus TCP ではチャンネル

あらかじめ定義されているデバイスには定義済みデータ交換があるため、サイクリックデータ交換が自動的に定義されます。EDS ファイル付きデバイスには定義済みの接続があります。アプリケーションで使用する接続またはチャンネルを選択してください。

必要に応じて、専用 DTM または適切なサードパーティツールを使用して、これらのデータ交換を設定してください。詳細については、デバイスのマニュアルを参照してください。

これらのデバイスおよび汎用スレーブデバイス用に新しい要求を追加、設定できます。

すべてのデータ交換において、プログラムで使用する変数をマップできます。

## EtherNet/IP サイクリックデータ交換の設定

### 接続概要

EtherNet/IP デバイスにアクセスするには、接続 (EtherNet/IP プロトコルレベルで使用されるグローバル名) を開始してください。

この接続によって、組み合わせられたデータをアセンブリ (45 ページ) に転送できます。

接続処理 (開始 / 停止) は、ロジックコントローラーで自動的に管理されます。

接続の制限については、ロジックコントローラープログラミングガイドを参照してください。

詳細については、産業用 Ethernet マネージャーの動作モード (81 ページ) を参照してください。

### アセンブリ

I/O データおよび設定データは、アセンブリオブジェクトに組み入れることができます。

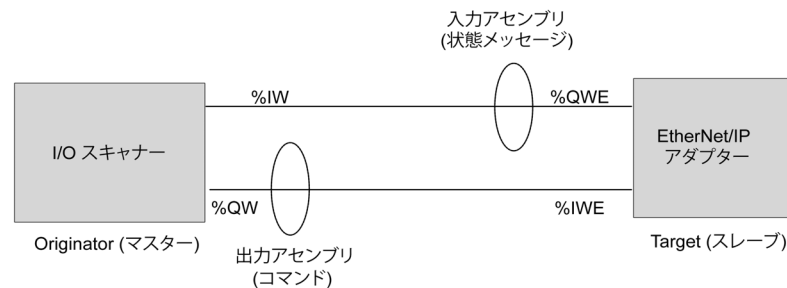
異なるオブジェクトのデータ (属性) を 1 つのオブジェクトに組み入れることで、単一の接続でデータを送受信できます。

アセンブリオブジェクトのインスタンスは、I/O 接続に関連する入力データおよびデータを統合するために使用します。

アセンブリオブジェクトは、クラス、インスタンス、および属性に構造化されています。

- クラスは、同じ種類のシステムコンポーネントを表すオブジェクトの集まりです。
- オブジェクトインスタンスは、クラス内の特定のオブジェクトを表します。各インスタンスには、それぞれ独自の属性値セットがあります。
- 属性は、オブジェクトおよびオブジェクトクラスの特徴です。一般的に属性は状態の情報を提供したり、オブジェクトの動作を定義します。

EtherNet/IP 通信での入力アセンブリ および出力アセンブリ の方向性を以下の図に示します。



EtherNet/IP 設定パラメーターは次のように定義されます。

- **インスタンス**: アセンブリを参照する番号
- **サイズ**: アセンブリのチャンネル数  
各チャンネルのメモリーサイズは 2 バイトで、%IWx オブジェクトまたは %QWx オブジェクトの値を格納します。x はチャンネル番号です。

例えば、**出力アセンブリのサイズ**が 20 の場合、%IWy...%IW(y+20-1) を扱う 20 個の入力チャンネル (IW0...IW19) があります。y はアセンブリに使用できる最初のチャンネルです。

### EtherNet/IP デバイス接続タブ

各 EtherNet/IP デバイスに接続があります。

デバイスツリーの EtherNet/IP デバイスをダブルクリックし、**接続タブ**を選択します。

Connection N°	Connection Name	RPI O-> T (ms)	RPI T-> O (ms)	O-> T size (byte)	T-> O size (byte)
257	CIP Basic Control	10	10	4	4

欄	コメント
接続 N°	接続番号は固有です。SoMachine によって自動的に割り当てられます。
接続名	接続名は、SoMachine によって自動的に割り当てられます。
RPI O --> T (ms)	Requested Packet Interval (要求パケット間隔): スキャナーによって要求されたサイクリックデータ送信の間隔。
RPI T --> O (ms)	
O->T サイズ (バイト)	Originator (O: マスター) と Target (T: スレーブ) の間で交換するバイト数。
T->O サイズ (バイト)	
Config#1 サイズ (バイト)	送信する設定パラメーターのバイト数。 接続に設定アセンブリ (49 ページ) が含まれている場合に表示されます。
Config#2 サイズ (バイト)	

接続を作成するには、**接続の追加**をクリックします。

接続を変更するには、接続を選択して**接続の編集**をクリックするか、接続をダブルクリックします。

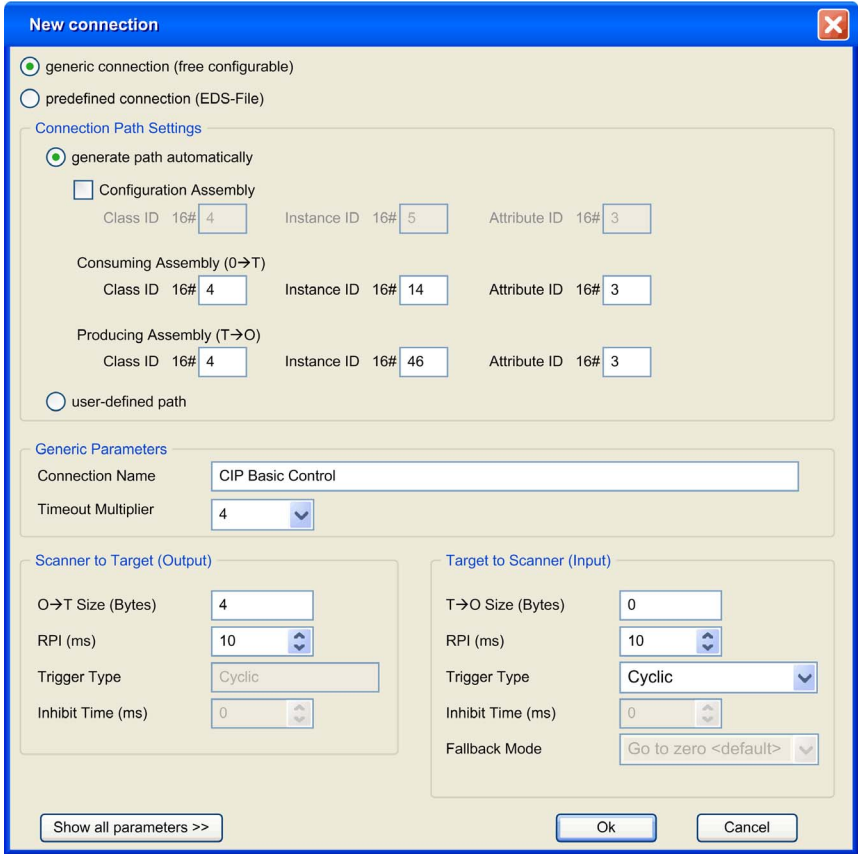
接続を削除するには、接続を選択して**接続の削除**をクリックします。

### EtherNet/IP 接続の追加

EtherNet/IP 接続を設定するには、以下の手順で行います。

手順	手順内容
1	デバイスツリーの EtherNet/IP デバイスをダブルクリックします。
2	接続タブを選択します。
3	接続の追加をクリックします。

(1) クラス ID、インスタンス ID、および属性 ID は、デバイスマニュアルに記載されています。アセンブリ情報の探し方 (54 ページ) を参照してください。

手順	手順内容
4	<p>汎用接続 (自由に設定可能) を選択します。</p> 
5	<p>設定アセンブリ (49 ページ) を選択します。</p>
6	<p>消費アセンブリ (O-&gt;T) を設定します。</p> <ul style="list-style-type: none"> <li>● クラス ID (デフォルトは 4): クラス識別子 <sup>(1)</sup></li> <li>● インスタンス ID: インスタンス識別子 <sup>(1)</sup></li> <li>● 属性 ID (デフォルトは 3): 属性識別子 <sup>(1)</sup></li> </ul>
7	<p>生成アセンブリ (T-&gt;O) を設定します。</p> <ul style="list-style-type: none"> <li>● クラス ID (デフォルトは 4): クラス識別子 <sup>(1)</sup></li> <li>● インスタンス ID: インスタンス識別子 <sup>(1)</sup></li> <li>● 属性 ID (デフォルトは 3): 属性識別子 <sup>(1)</sup></li> </ul>
8	<p>タイムアウト乗数を選択します。4 (デフォルト) / 8 / 16 / 32 / 64 / 128 / 256 / 512</p>
9	<p>スキャナーから表示器 (出力) を設定します。</p> <ul style="list-style-type: none"> <li>● O-&gt;T サイズ (バイト): 送信するバイト数: 最大 505</li> <li>● トリガータイプ: サイクリック</li> <li>● RPI (ms) (デフォルトは 10 ms): スキャナーによって要求されたサイクリックデータ送信の間隔。</li> </ul>
10	<p>表示器からスキャナー (入力) を設定します。</p> <ul style="list-style-type: none"> <li>● T-&gt;O サイズ (バイト): 送信するバイト数 (アセンブリのチャンネル数): 最大 509)</li> <li>● トリガータイプ: サイクリック / 状態の変更。状態の変更を選択した場合、インヒビット時間が有効になり、初期値の 2 ms に設定されます。</li> <li>● RPI (ms) (デフォルトは 10 ms): スキャナーによって要求されたサイクリックデータ送信の間隔。</li> <li>● インヒビット時間 (ms) (デフォルトは 2 ms): 2 つのデータ交換の間の最小時間。トリガータイプが状態の変更である場合にアクセス可能。値は 2 ms の倍数にしてください。最大値は Target to Scanner の RPI (ms) の値です。ただし、254 ms を超える設定はできません。</li> </ul>
11	<p>OK をクリックします。</p>
<p><sup>(1)</sup> クラス ID、インスタンス ID、および属性 ID は、デバイスマニュアルに記載されています。アセンブリ情報の探し方 (54 ページ) を参照してください。</p>	

対応しているアセンブリの詳細については、デバイスのマニュアルを参照してください。  
 アドバンスドパラメーターの詳細については、EtherNet/IP エキスパートモードでの接続プロパティ (51 ページ) を参照してください。

**注記:** O→T サイズ (バイト) と T→O サイズ (バイト) の制限、およびスキャナーの最大入力/出力ワード (1024) によるスキャナーリソースの過負荷 (62 ページ) を確認してください。

**定義済み接続の追加**

定義済み接続は以下で利用できます。

- 定義済みデバイス (18 ページ参照)
- DTM に対応しているデバイス
- EDS ファイル付きのデバイス

汎用スレーブドライブには定義済み接続はありません。

定義済み EtherNet/IP 接続を追加するには、以下の手順で行います。

手順	手順内容
1	デバイスツリーの EtherNet/IP デバイスをダブルクリックします。
2	接続タブを選択します。
3	接続の追加をクリックします。
4	定義済み接続 (EDS-File) を選択します。
4	定義済み接続から 1 つ選択します。
5	タイムアウト乗数を選択します。4 (デフォルト) / 8 / 16 / 32 / 64 / 128 / 256 / 512
6	スキャナーから表示器 (出力) を設定します。 <ul style="list-style-type: none"> <li>● O→T サイズ (バイト): 送信するバイト数</li> <li>● トリガータイプ: サイクリック</li> <li>● RPI (ms) (初期値は EDS で定義): スキャナーによって要求されたサイクリックデータ送信の間隔。</li> </ul>



手順	手順内容
7	<p>表示器からスキャナー (入力) を設定します。</p> <ul style="list-style-type: none"> <li>● <b>T → O サイズ (バイト)</b>: 送信するバイト数 (アセンブリのチャンネル数)</li> <li>● <b>トリガータイプ</b>: サイクリック / 状態の変更。<b>状態の変更</b>を選択した場合、<b>インヒビット時間</b>が有効になり、初期値の 2 ms に設定されます。</li> <li>● <b>RPI (ms)</b> (初期値は EDS で定義): スキャナーによって要求されたサイクリックデータ送信の間隔。</li> <li>● <b>インヒビット時間 (ms)</b> (デフォルトは 2 ms): 2 つのデータ交換の間の最小時間。<b>トリガータイプが状態の変更</b>である場合にアクセス可能。値は 2 ms の倍数にしてください。最大値は Target to Scanner の <b>RPI (ms)</b> の値です。ただし、254 ms を超える設定はできません。</li> </ul>
8	OK をクリックします。

### 設定アセンブリの設定

一部のデバイスは設定アセンブリに対応しています。

設定アセンブリは、スキャナーの起動時に送信され、単一の要求で設定パラメーターをデバイスに読み込む要求です。

設定アセンブリを設定するには、以下の手順で行います。

手順	手順内容
1	デバイスツリーの EtherNet/IP デバイスをダブルクリックします。
2	接続タブを選択します。
3	既存の接続を選択して <b>接続の編集</b> をクリックします。
4	<b>汎用接続 (自由に設定可能)</b> を選択します。
5	<b>設定アセンブリ</b> を選択します。
6	<p><b>設定アセンブリ</b>を設定します。</p> <ul style="list-style-type: none"> <li>● <b>クラス ID</b> (デフォルトは 4): クラス識別子<sup>(1)</sup></li> <li>● <b>インスタンス ID</b>: インスタンス識別子<sup>(1)</sup></li> <li>● <b>属性 ID</b> (デフォルトは 3): 属性識別子<sup>(1)</sup></li> </ul>
7	<b>すべてのパラメーターを表示 &gt;&gt;&gt;</b> をクリックします。
8	<p>スキャナーから表示器 (出力) を設定します。</p> <ul style="list-style-type: none"> <li>● <b>Config#1 サイズ (バイト)</b>: 設定パラメーターの最初のセットの番号</li> <li>● <b>Config#2 サイズ (バイト)</b>: 設定パラメーターの 2 番目のセットの番号</li> </ul>

手順	手順内容
9	<p><b>OK</b> をクリックします。  <b>結果</b>：接続タブに設定パラメーターが表示されます。</p>
10	<p>値の欄をダブルクリックして設定パラメーターの値を設定します。</p> <p>(1) クラス ID、インスタンス ID、および属性 ID は、デバイスマニュアルに記載されています。アセンブリ情報の探し方 (54 ページ) を参照してください。</p>

### EtherNet/IP 接続のプロパティ

アドバンストパラメータービューで接続を編集します。

### 接続設定

パラメーター	値	説明
パスの自動生成	はい / いいえ	アセンブリのパラメーターを設定できます。
設定アセンブリ	はい / いいえ	設定アセンブリ (49ページ) を設定できます。
クラス ID	2 バイト ( デフォルトは 04h)	クラス識別子 <sup>(1)</sup>
インスタンス ID	2 バイト ( デフォルトは 0)	インスタンス識別子 <sup>(1)</sup>
属性 ID	2 バイト ( デフォルトは 03h)	属性識別子 <sup>(1)</sup>
消費アセンブリ (O->T)		
クラス ID	2 バイト ( デフォルトは 04h)	クラス識別子 <sup>(1)</sup>
インスタンス ID	2 バイト ( デフォルトは 0)	インスタンス識別子 <sup>(1)</sup>
属性 ID	2 バイト ( デフォルトは 03h)	属性識別子 <sup>(1)</sup>
生成アセンブリ (T->O)		
クラス ID	2 バイト ( デフォルトは 04h)	クラス識別子 <sup>(1)</sup>
インスタンス ID	2 バイト ( デフォルトは 0)	インスタンス識別子 <sup>(1)</sup>
属性 ID	2 バイト ( デフォルトは 03h)	属性識別子 <sup>(1)</sup>

(1) クラス ID、インスタンス ID、および属性 ID は、デバイスマニュアルに記載されています。アセンブリ情報の探し方 (54 ページ) を参照してください。

パラメーター	値	説明
ユーザー定義パス	はい/いいえ	パスの自動生成が無効になり、接続パスフィールドが有効になります。
<p>(1) クラス ID、インスタンス ID、および属性 ID は、デバイスマニュアルに記載されています。アセンブリ情報の探し方 (54 ページ) を参照してください。</p>		

汎用パラメーター

パラメーター	値	説明
接続パス	バイト型配列	物理リンクオブジェクトのコード化したもの。
通信タイプ	<ul style="list-style-type: none"> <li>Exclusive Owner (独占オーナー)(デフォルト)</li> <li>Listen Only (リッスンオンリー)</li> <li>Input Only (入力オンリー)</li> </ul>	<p><b>Exclusive Owner (独占オーナー):</b> アセンブリのデータが1つのスキャナーでのみ制御できる出力接続ポイント (通常はアセンブリオブジェクト) への双方向接続です。入力アセンブリへの接続がある場合、データはスキャナーに送信されています。入力データ長が0の場合、ハートビート接続になります。</p> <p><b>Listen Only (リッスンオンリー):</b> スキャナーは Target (スレーブ) デバイスから入力データを受信し、Target (スレーブ) デバイスにハートビートを生成します。出力データはありません。Listen Only (リッスンオンリー) 接続は、既存の <b>Exclusive Owner (独占オーナー)</b> または <b>Input Only (入力オンリー)</b> 接続にのみ指定できます。この基本接続が停止すると、<b>リッスンのみ</b> 接続も停止またはタイムアウトします。</p> <p><b>Input Only (入力オンリー):</b> スキャナーは Target (スレーブ) デバイスから入力データを受信し、Target (スレーブ) デバイスにハートビートを生成します。出力データはありません。</p>
タイムアウト乗数	4 (デフォルト) / 8 / 16 / 32 / 64 / 128 / 256 / 512	スキャナータイムアウト (28 ページ) は、接続の RPI とタイムアウト乗数によって管理されます。

スキャナーから表示器 (出力)

パラメーター	値	説明
O->T サイズ (バイト)	0 ~ XX => デバイス固有	アセンブリのチャンネルのサイズ。各チャンネルのメモリーサイズは2バイトで、%IWx オブジェクトまたは %QWx オブジェクトの値を格納します。x はチャンネル番号です。
RPI (ms)	単位 ms (デフォルトは 10 ms)	<p>要求パケット間隔。スキャナーによって要求されたサイクリックデータ送信の間隔。デバイスは常に最小の RPI を提供しますが、コントローラーではシステムに過負荷を掛けないために最大の RPI にすることが目標です。EtherNet/IP フィールドバスにデバイスを追加するたび、または RPI の値を変更するたびにリソースを確認することを推奨します (スキャナーリソースチェッカー (62 ページ) を参照してください)。</p> <p>デバイスの RPI は、デバイスのマニュアルに指定されている場合があります。ただし、通常この情報はデバイスに付属されている EDS ファイル (30 ページ) の一部として提供されません。</p>
トリガータイプ	サイクリック	<b>サイクリック:</b> エンドポイントからあらかじめ定義された周期時間間隔でメッセージが送信されます。
<p><b>注記:</b> 通信フォーマットが <b>32 ビット 実行 / アイドル</b> に設定されている場合、要求内でスキャナーの状態が送信されます。スキャナーが IDLE 状態であるという情報を受信した場合、Target (スレーブ) デバイスは同様には応答しない場合があります。例えば、コントローラーが STOPPED または HALT のときは、他の Target (スレーブ) デバイスが入力を更新していても一部の Target (スレーブ) デバイスは更新しない可能性があります。</p>		

パラメーター	値	説明
インヒビット時間	0 ms	2つのデータ交換間の最小時間。
Config#1 サイズ (バイト)	0 ~ XX => デバイス固有	接続パスに設定アセンブリが含まれている場合にアクセスできます。
Config#2 サイズ (バイト)	0 ~ XX => デバイス固有	送信するパラメーター (1 バイト) の数。スキャナーの起動時に設定値がデバイスに送信されます。
接続タイプ	ポイントツーポイント	要求の接続タイプ
固定 / 変数	固定	要求の長さは固定です。
送信形式	<ul style="list-style-type: none"> <li>● 32 ビット 実行 / アイドル (デフォルト)</li> <li>● ピュアデータ</li> <li>● ハートビート</li> </ul>	要求の送信形式。詳細は、 <a href="#">ODVA website</a> を参照してください。
<p><b>注記:</b> 通信フォーマットが <b>32 ビット 実行 / アイドル</b> に設定されている場合、要求内でスキャナーの状態が送信されます。スキャナーが IDLE 状態であるという情報を受信した場合、Target (スレープ) デバイスは同様には応答しない場合があります。例えば、コントローラーが STOPPED または HALT のときは、他の Target (スレープ) デバイスが入力を更新していても一部の Target (スレープ) デバイスは更新しない可能性があります。</p>		

表示器からスキャナー (入力)

パラメーター	値	説明
T->O サイズ (バイト)	0 ~ XX => デバイス固有	アセンブリのチャンネルのサイズ。各チャンネルのメモリーサイズは 2 バイトで、%IWx オブジェクトまたは %QWx オブジェクトの値を格納します。x はチャンネル番号です。
RPI (ms)	単位 ms (デフォルトは 10 ms)	要求パケット間隔。スキャナーによって要求されたサイクリックデータ送信の間隔。デバイスは常に最小の RPI を提供しますが、コントローラーではシステムに過負荷を掛けないために最大の RPI にすることが目標です。EtherNet/IP フィールドバスにデバイスを追加するたび、または RPI の値を変更するたびにリソースを確認することを推奨します (スキャナーリソースチェッカー (62 ページ) を参照してください)。デバイスの RPI は、デバイスのマニュアルに指定されている場合があります。ただし、通常この情報はデバイスに付属されている EDS ファイル (30 ページ) の一部として提供されます。
トリガータイプ	<ul style="list-style-type: none"> <li>● サイクリック (デフォルト)</li> <li>● 状態の変更</li> </ul>	<p><b>サイクリック:</b> エンドポイントからあらかじめ定義された周期時間間隔でメッセージが送信されます。</p> <p><b>状態の変更:</b> 変更が発生したときに、状態が変更したエンドポイントからメッセージが送信されます。データは、変更がない場合に接続がタイムアウトしないようにバックグラウンドでも周期間隔 (RPI) で送信されます。</p>
インヒビット時間 (ms)	2 ms の倍数 (デフォルトは 2 ms)	2つのデータ交換間の最小時間。 トリガータイプが <b>状態の変更</b> である場合にアクセス可能。インヒビット時間の最大値は RPI です。また 254 ms までに制限されています。
フォールバックモード	ゼロに移動 < デフォルト >	エラー / 停止時に入力をリセットします。
接続タイプ	<ul style="list-style-type: none"> <li>● マルチキャスト (デフォルト)</li> <li>● ポイントツーポイント</li> </ul>	要求の接続タイプ
固定 / 変数	固定	要求の長さは固定です。

パラメーター	値	説明
送信形式	<ul style="list-style-type: none"> <li>● ピュアデータ (デフォルト)</li> <li>● ハートビート</li> </ul>	要求の送信形式。詳細は、 <a href="#">ODVA website</a> を参照してください。

### アセンブリ情報の探し方

アセンブリ情報は、デバイスのマニュアルに記載されています。通常、アセンブリオブジェクトの説明の一部です。

アセンブリを設定するには、以下の情報項目を指定します。

#### 1. クラス ID

「アセンブリオブジェクト」のクラス ID は 4 です。

#### 2. インスタンス ID

アプリケーションおよびデバイスの種類に応じて、アセンブリのインスタンスを選択します。アセンブリのインスタンスを選択すると、デバイスに専用のステートマシンが生成されます。

- **設定アセンブリ**：少数のデバイスでのみ対応しています。どのアセンブリのインスタンスに対応しているかはデバイスのマニュアルで確認してください。
- **消費アセンブリ**：デバイスのマニュアルでは「デバイス出力」(デバイス側からの観点)と呼ばれる場合もあります。
- **生成アセンブリ**：デバイスのマニュアルでは「デバイス入力」(デバイス側からの観点)と呼ばれる場合もあります。

#### 3. 属性 ID

読み込む属性を探します。これは、接続中に交換されるデータのバッファーに対応します。

属性のプロパティには、生成アセンブリの書き込みアクセス権および消費アセンブリの読み込みアクセス権が必要です。

属性 ID は 2 つのアセンブリで同じ 3 です。属性のアクセスが Get/Set の属性と一致します。名前は「データ」である場合が多く、データ型は「バイト型配列」です。

## EtherNet/IP I/O マッピング

### 概要

データ交換の設定をすると、プログラムで使用する変数をマップできます。

### EtherNet/IP Scanner I/O マッピングの設定

EtherNet/IP Scanner I/O マッピングを設定するには、以下の手順で行います。

手順	手順内容
1	<p>デバイスツリーの <b>Industrial_Ethernet_Manager</b> をダブルクリックします。  <b>結果</b>：設定ウィンドウが表示されます。</p>
2	<p><b>EtherNet/IP スキャナー I/O マッピング</b> タブを選択します。</p> 
3	<p>リストから<b>バス周期タスク</b>を選択します。</p> <ul style="list-style-type: none"> <li>● Use parent bus cycle setting (デフォルト)</li> <li>● MAST</li> <li>● プロジェクトの既存タスク</li> </ul> <p><b>注記</b>：産業用 Ethernet マネージャーを含むデバイスの I/O マッピングエディター内の<b>バス周期タスク</b>パラメーターが、I/O イメージ (%QW、%IW) を更新するタスクを定義します。これらの I/O イメージは、EtherNet/IP Target (スレーブ) デバイスに送信される EtherNet/IP 要求およびヘルスビットに対応しています。</p>

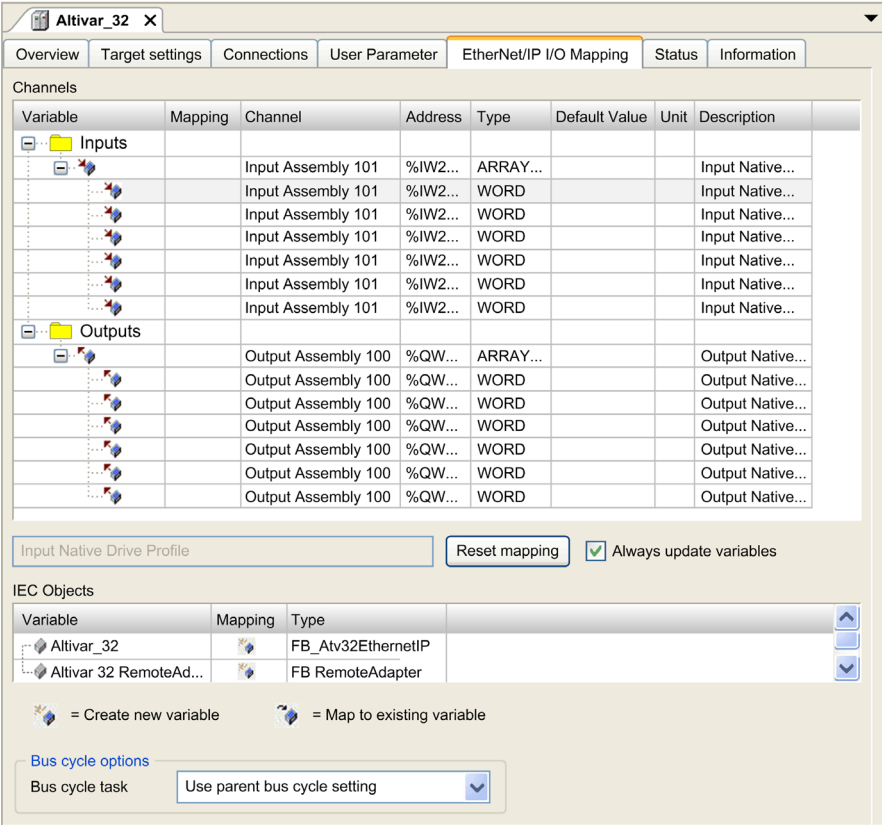
**注記**：産業用 Ethernet マネージャーが設定されている場合、デバイスネットワーク用のポスト設定ファイルは無視されます。

### EtherNet/IP Target (スレーブ) デバイスの I/O マッピングの設定

定義済み接続または新しい接続でデータ交換の設定をすると、プログラムで使用する変数をマップできます。

EtherNet/IP Target (スレーブ) デバイスの I/O マッピングを設定するには、以下の手順で行います。

手順	手順内容
1	<p>デバイスツリーの EtherNet/IP Target (スレーブ) デバイスをダブルクリックします。  <b>結果</b>：設定ウィンドウが表示されます。</p>

手順	手順内容
2	<p data-bbox="459 203 906 230">EtherNet/IP I/O マッピングタブを選択します。</p> 
3	<p data-bbox="459 1122 858 1149">リストから<b>バス周期タスク</b>を選択します。</p> <ul data-bbox="459 1149 922 1227" style="list-style-type: none"> <li>● <b>Use parent bus cycle setting</b> (デフォルト)</li> <li>● <b>MAST</b></li> <li>● プロジェクトの既存タスク</li> </ul> <p data-bbox="459 1238 1364 1361"><b>注記：</b>産業用 Ethernet マネージャーを含むデバイスの I/O マッピングエディター内の<b>バス周期タスク</b>パラメーターが、I/O イメージ (%QW、%IW) を更新するタスクを定義します。これらの I/O イメージは、EtherNet/IP Target (スレーブ) デバイスに送信される EtherNet/IP 要求およびヘルスビットに対応しています。</p>
4	<p data-bbox="459 1368 1364 1451"><b>変数</b>列のセルをダブルクリックしてテキストフィールドを開きます。変数名を入力するか、参照ボタン [...] をクリックして<b>入力アシスタント</b>から変数を選択します。</p>



## Modbus TCP サイクリックデータ交換の設定

### 概要

Modbus TCP サイクリックデータ交換を設定するには、以下を行います。

- 各 Modbus TCP スレーブデバイスに (チャンネル上の) データ交換要求および I/O マッピングを設定
- Modbus TCP スレーブデバイスに I/O スキャナーを設定

### Modbus TCP チャンネル


Modbus チャンネルが、マスターとスレーブ間で Modbus 要求を伝送します。

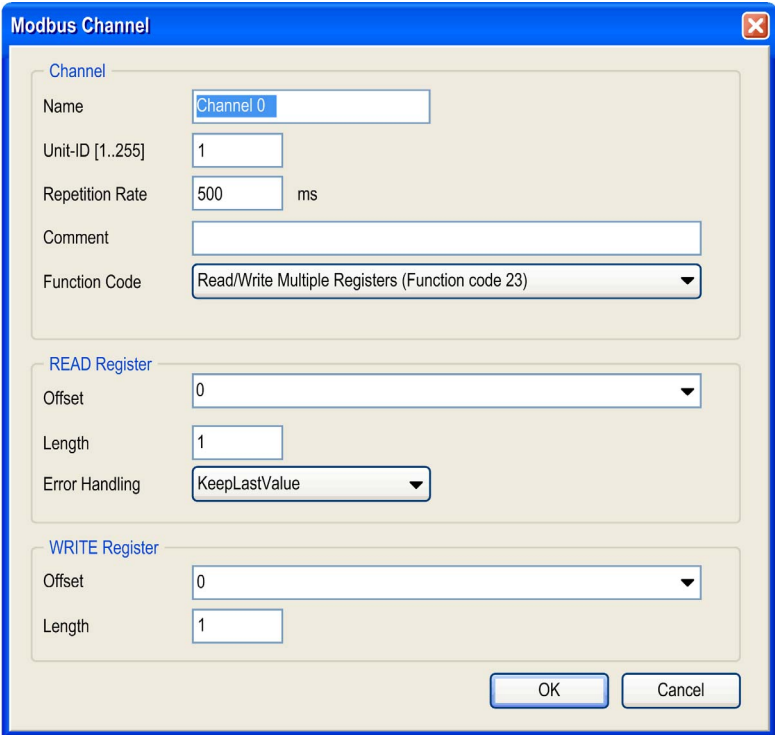
Advantys OTB および定義済みスレーブデバイスは、デバイスごとに 1 つのチャンネルを使用します。このチャンネルは、SoMachine ソフトウェアを使用して設定します。

汎用スレーブデバイスでは、複数のチャンネルを使用できます。デバイスに複数の異なる要求を送信するには、複数のチャンネルを作成します。

### Modbus TCP スレーブデバイスチャンネルの設定

Modbus TCP スレーブデバイスの (チャンネル上の) データ交換を設定するには、以下の手順で行います。

手順	手順内容
1	<p>デバイスツリーの Modbus TCP スレーブデバイスをダブルクリックします。  <b>結果:</b> 設定ウィンドウが表示されます。</p>
2	<p>Modbus TCP チャンネル設定タブをクリックします。</p> 
3	<p>未定義のチャンネルを削除するには、チャンネルを選択して<b>削除</b>をクリックします。</p>
4	<p>チャンネルのパラメーターを変更するには、チャンネルを選択して<b>編集</b>をクリックします。  <b>注記:</b> 定義済みチャンネルがあるデバイスの場合、<b>繰り返し率</b>の値のみ変更できます。</p>

手順	手順内容
5	<p>チャンネルを追加するには、<b>チャンネルを追加</b>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。</p> 
6	<p>チャンネルでは、以下を定義できます。</p> <ul style="list-style-type: none"> <li>● <b>名前</b>: チャンネルに名前を付けるためのオプションの文字列</li> <li>● <b>ユニット-ID [1..255]</b>: Modbus TCP スレーブデバイスのユニット ID <sup>(1)</sup> (デフォルトは 255)</li> <li>● <b>繰り返し率</b>: Modbus 要求のポーリング間隔 (デフォルトは 20 ms)</li> <li>● <b>コメント</b>: チャンネルの説明を記述するオプションのフィールド</li> <li>● <b>ファンクションコード</b>: Modbus 要求の種類             <ul style="list-style-type: none"> <li>○ 複数のレジスターの読み込み/書き込み (ファンクションコード 23) (デフォルト)</li> <li>○ 保持レジスターの読み込み (ファンクションコード 03)</li> <li>○ 複数のレジスターの書き込み (ファンクションコード 16)</li> </ul> </li> </ul> <p>レジスターの読み込みでは、以下を定義できます。</p> <ul style="list-style-type: none"> <li>● <b>オフセット</b>: 0 ~ 65535 の読み込み開始レジスター番号</li> <li>● <b>長さ</b>: 読み込むレジスターの数 (ファンクションコードによって異なる)</li> <li>● <b>エラー処理</b>: 通信が中断された場合のフォールバック値を定義             <ul style="list-style-type: none"> <li>○ 最後の値を保持 (デフォルト) は最後の有効な値を保持します</li> <li>○ ゼロに設定は値を 0 にリセットします</li> </ul> </li> </ul> <p>レジスターの書き込みでは、以下を定義できます。</p> <ul style="list-style-type: none"> <li>● <b>オフセット</b>: 0 ~ 65535 の書き込み開始レジスター番号</li> <li>● <b>長さ</b>: 書き込みレジスターの数 (ファンクションコードによって異なる)</li> </ul>
7	<p>OK をクリックして、チャンネルの設定を検証します。</p>
8	<p>手順 5 ~ 7 を繰り返して、デバイスの Modbus 通信を定義する他のチャンネルを作成します。各 Modbus 要求に対してチャンネルを作成してください。</p>

(1) 複数の Modbus デバイスで構成されている Modbus TCP デバイス、例えば Modbus TCP から Modbus RTU ゲートウェイなどではユニット識別子が使用されます。その場合、ユニット識別子によってゲートウェイの背後にあるデバイスのスレーブアドレスに到達することができます。デフォルトでは、Modbus/TCP 対応デバイスはユニット識別子パラメーターを無視します。

**レジスターの読み込み / 書き込みの長さ**

レジスターの読み込み / 書き込みの長さは、Modbus のファンクションコードによって異なります。

チャンネル 1 点に対してレジスタの読み込み / 書き込みの最大長を次の表示に示します。

Modbus ファンクションコード	最大長	
	レジスタの読み込み	レジスタの書き込み
複数のレジスタの読み込み / 書き込み (ファンクションコード 23)	125	121
レジスタの読み込み (ファンクションコード 03)	125	-
レジスタの書き込み (ファンクションコード 16)	-	123

**注記：**これらの制限およびスキャナーの最大入出力ワード (2048) によるスキャナーリソースの過負荷 (62 ページ参照) を確認してください。

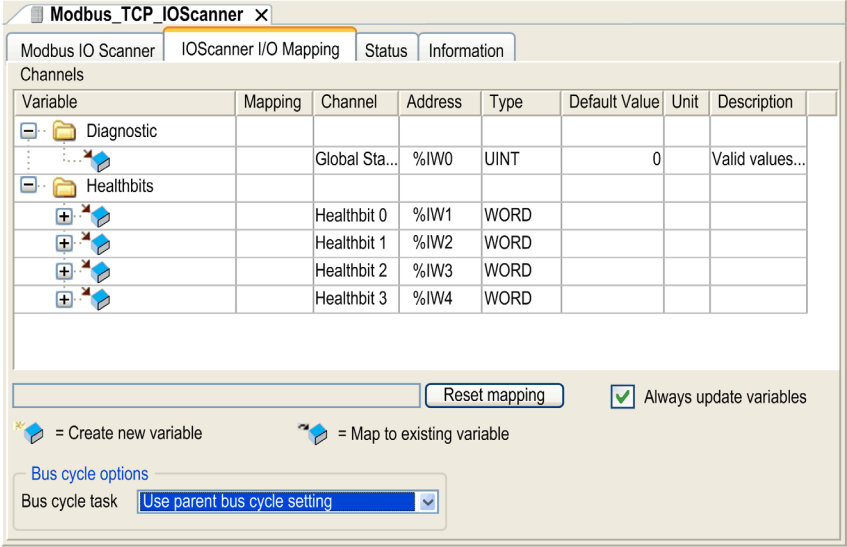
## Modbus TCP I/O マッピング

### 前提条件

Modbus TCP チャンネルが必要です。

### Modbus Serial IOScanner の設定

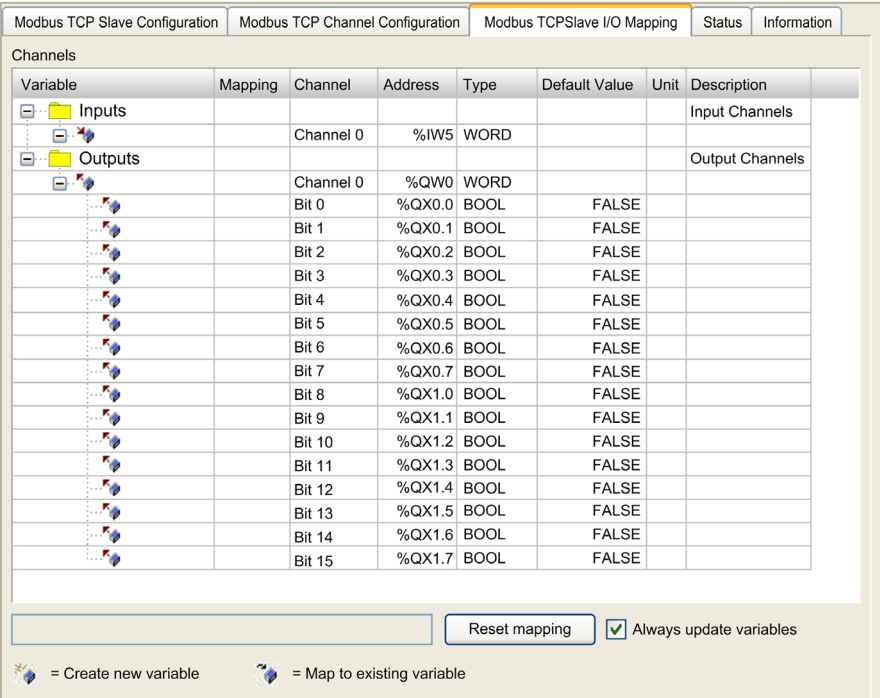
Modbus Serial IOScanner を設定するには、以下の手順で行います。

手順	手順内容
1	<p>デバイスツリーの <b>Industrial_Ethernet_Manager</b> をダブルクリックします。  <b>結果</b> : 設定ウィンドウが表示されます。</p>
2	<p><b>IOScanner I/O マッピング</b> タブを選択します。</p> 
3	<p>リストから<b>バス周期タスク</b>を選択します。</p> <ul style="list-style-type: none"> <li>● <b>Use parent bus cycle setting</b> (デフォルト)</li> <li>● <b>MAST</b></li> <li>● プロジェクトの既存タスク</li> </ul> <p><b>注記</b> : Modbus Serial IOScanner を含むデバイスの I/O マッピングエディター内の<b>バス周期タスク</b>パラメーターが、I/O イメージ (%QW、%IW) を更新するタスクを定義します。これらの I/O イメージは、Modbus スレーブおよびヘルスビットに送信される Modbus 要求に対応しています。</p>
4	<p>変数列のセルをダブルクリックしてテキストフィールドを開きます。                  変数名を入力するか、参照ボタン [...] をクリックして<b>入力アシスタント</b>から変数を選択します。</p>

### Modbus TCP スレーブデバイスの I/O マッピングの設定

Modbus TCP スレーブデバイスの I/O マッピングを設定するには、以下の手順で行います。

手順	手順内容
1	<p>デバイスツリーの Modbus TCP スレーブデバイスをダブルクリックします。  <b>結果</b> : 設定ウィンドウが表示されます。</p>

手順	手順内容
2	<p><b>ModbusTCPSlave I/O マッピングタブ</b>を選択します。</p> 
3	<p>リストから<b>バス周期タスク</b>を選択します。</p> <ul style="list-style-type: none"> <li>● Use parent bus cycle setting (デフォルト)</li> <li>● MAST</li> <li>● プロジェクトの既存タスク</li> </ul> <p><b>注記</b>：Modbus Serial IOScanner を含むデバイスの I/O マッピングエディター内の<b>バス周期タスク</b>パラメーターが、I/O イメージ (%QW、%IW) を更新するタスクを定義します。これらの I/O イメージは、Modbus スレーブおよびヘルスビットに送信される Modbus 要求に対応しています。</p>
4	<p><b>変数列</b>のセルをダブルクリックしてテキストフィールドを開きます。変数名を入力するか、参照ボタン [...] をクリックして<b>入力アシスタント</b>から変数を選択します。</p>

## 産業用 Ethernet マネージャーの負荷検証

### 目的

Industrial\_Ethernet\_manager の負荷が 100% を超えると、サイクリックデータ交換が設定した間隔で処理されない場合があります。

スキャナーリソースタブで Industrial\_Ethernet\_manager の負荷を推定できます。

機械を動作させる前にこの負荷を確認してください。

負荷に対応するには、次の負荷要因を調整します。

- スレーブの数
- EtherNet/IP の場合
  - (EtherNet/IP Scanner の ) 接続数
  - 接続の RPI
- Modbus TCP の場合
  - (Modbus Serial IOScanner の ) チャンネル数
  - チャンネルの繰り返し率

### 負荷の推定

この方程式によって、Industrial\_Ethernet\_manager が Modbus Serial IOScanner デバイスのみを管理する場合の負荷を推定できます。

$$\text{IOScanner の負荷 (\%)} = \sum_{\text{チャンネル}=1}^{\text{Nb チャンネル}} \frac{50}{\text{繰り返し率} \times \text{チャンネル}}$$

この方程式によって、Industrial\_Ethernet\_manager が 1 台以上の Ethernet/IP デバイスを管理する場合の負荷を推定できます。

$$\text{スキャナーの負荷 (\%)} = \sum_{\text{チャンネル}=1}^{\text{Nb チャンネル}} \frac{200}{\text{繰り返し率} \times \text{チャンネル}} + \sum_{\text{接続}=1}^{\text{Nb 接続}} \frac{\text{負荷}}{\text{RPI}_{\text{接続}}}$$

RPI 接続 < 5 の場合は負荷 = 100、それ以外の場合は負荷 = 62.5。

#### 注記：

この負荷の推定には、以下のようなアウトオブプロセスのデータ交換 (79 ページ) による負荷の増加は考慮されていません。

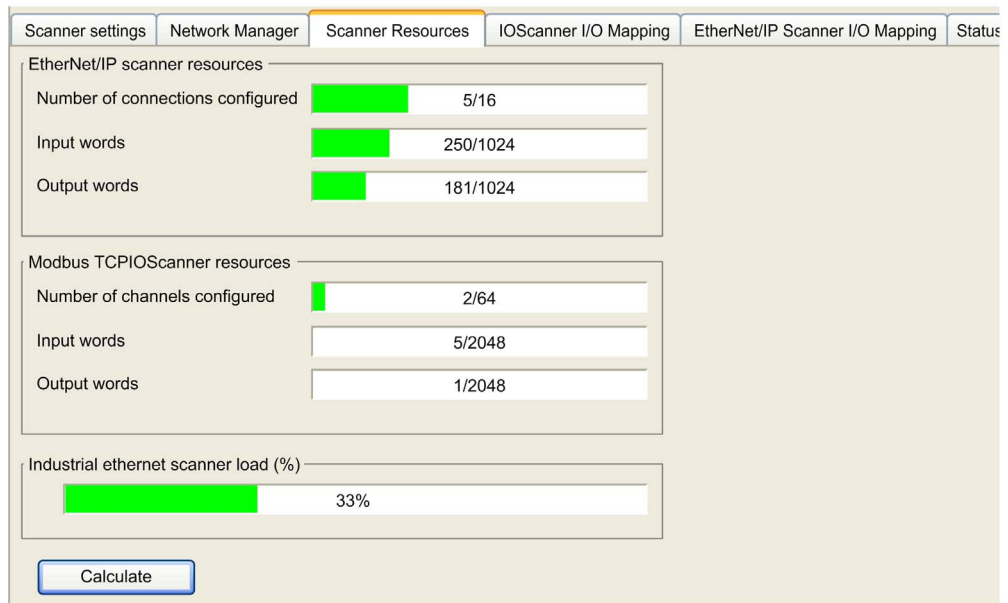
- DTM、Web サーバー、および Modbus TCP 要求
- フィールドバス通信 (DTM、PC がフィールドバス上にある場合の Web サーバー通信)
- TcpUdpCommunications ライブラリによって生成された TCP UDP 通信

SoMachine では、自動負荷計算が利用できます。

手順	手順内容
1	デバイスツリーの Industrial_Ethernet_manager ノードをダブルクリックします。
2	スキャナーリソースタブを選択します。
3	計算をクリックします。

詳細

スキャナリソースタブを次の図に示します。



## 2.8 産業用 Ethernet でのプログラミング

### 産業用 Ethernet でのプログラミング

#### 概要

**Industrial\_Ethernet\_manager** を追加すると、以下のライブラリが自動的にインスタンス化されます。

- Modbus Serial IOScanner
- EtherNet/IP Scanner

さらに、殆どの産業用 Ethernet スレーブデバイスにはファンクションおよびファンクションブロックを含む専用ライブラリがあります。

これらを使用することで、プログラムの記述が容易になります。

SoMachine には、使用可能な TVDA テンプレートが含まれています。

#### デバイスの動作モードの管理

Modbus Serial IOScanner ライブラリには以下のファンクションがあります。

- IOS\_GETSTATE: Modbus Serial IOScanner の状態の読み込み
- IOS\_START: Modbus Serial IOScanner の開始
- IOS\_GETHEALTH: ヘルスビット値の読み込み
- IOS\_STOP: Modbus Serial IOScanner の停止
- CONFIGURE\_OTB: Advantys OTB のソフトウェア設定の送信

詳細については、Modbus Serial IOScanner ライブラリ (99 ページ) を参照してください。

EtherNet/IP Scanner ライブラリには以下のファンクションがあります。

- EipControl: EtherNet/IP スキャナーの開始 / 停止
- EipGetHealth: ヘルスビット値の読み込み

詳細については、EtherNet/IP Scanner ライブラリ (133 ページ) を参照してください。

操作の詳細については、スレーブデバイスの動作モードの管理 (74 ページ) および 産業用 Ethernet のロジックコントローラー状態の影響 (81 ページ) を参照してください。

#### デバイスからのコマンドの送信と状態の読み込み

サイクリックデータ交換は、確定的なデータ交換が必要な汎用デバイスで使用します。サイクリックデータ交換は、産業用 Ethernet マネージャーで管理します。サイクリックデータ交換を設定するには、EtherNet/IP サイクリックデータ交換の設定 (45 ページ) を参照してください。プログラムでサイクリックデータを使用するには、EtherNet/IP I/O マッピング (55 ページ) を参照してください。

さらに、Explicit メッセージも送信できます。

EtherNet/IP デバイスでは、EtherNet/IP Explicit メッセージ通信ライブラリで以下を使用できます。

- Get\_Attribute\_All、Get\_Attribute\_All、オブジェクトのすべての属性の取得 (113 ページ) を参照してください
- Set\_Attribute\_All、Set\_Attribute\_All、インスタンスまたはクラスのすべての属性の設定 (115 ページ) を参照してください
- Get\_Attribute\_Single、Get\_Attribute\_Single、オブジェクトの属性の取得 (117 ページ) を参照してください
- Set\_Attribute\_Single、Set\_Attribute\_Single、オブジェクトの属性の設定 (119 ページ) を参照してください

EtherNet/IP デバイスでは、EtherNet/IP Explicit メッセージ通信ライブラリ (137 ページ) には実装されていない機能のために、EtherNet/IP Scanner ライブラリで EipDataExchange を使用できます。

Modbus TCP デバイスでは、READ\_VAR および WRITE\_VAR を使用できます。

操作の詳細については、開始時のスレーブデバイスの設定 (78 ページ) および要求によるデータ交換 (76 ページ) を参照してください。

#### TVDA テンプレートの使用

殆どの産業用 Ethernet スレーブデバイスが TVDA に含まれています。

SoMachine では、テンプレートからデバイスを追加 (29 ページ) することを推奨しています。



これにより、デバイスと共にすでにパラメーター化された複数のブロックおよびファンクションブロックが追加されます。



---

## 第 3 章

### デバイスネットワークの試運転

---

#### 概要

この章では、産業用 Ethernet ネットワークの試運転の方法について説明します。  
このフェーズは、デバイスネットワークの設定 (21 ページ) の次のフェーズです。  
このフェーズの最後に、アプリケーションの起動 (73 ページ) ができます。

#### この章について

この章には次の項目が含まれています。

項目	参照ページ
試運転	68
デバイスを認識させるための準備	70
正しいデバイス設定の適用	72

## 試運転

### 概要

試運転中に、以下を行ってください。

- 機械 ( ロジックコントローラーおよびスレーブデバイス ) に最初の電源投入
- ネットワークテストの実行
- 設定をネットワークデバイスにダウンロード
- ロジックコントローラーおよびネットワークデバイスの設定を調整 ( オンラインまたはデバイス上で直接 )
- 使用可能な各デバイスで FDR を完了
- アプリケーションのバックアップ

### 機械に最初の電源投入

初めての電源投入には、以下の手順に従います。

手順	手順内容
1	ロジックコントローラーにアプリケーションを転送します。 アプリケーションのダウンロード ( SoMachine, プログラミングガイド 参照 ) を参照してください。
2	ネットワーク設定 ( 32 ページ ) ( BOOTP、DHCP、固定 IP、ネットワーク名 ) を参照し、デバイスネットワーク上で認識されるように各デバイスを準備します。 詳細については、デバイスを認識させるための準備 ( 70 ページ ) を参照してください。
3	機械に電源を再投入します。一部のデバイスでは、正しいネットワーク設定を取得するために電源の再投入が必要な場合があります。
4	ネットワークテスト ( 86 ページ ) を実行します。

### 設定をネットワークデバイスにダウンロード

詳細については、正しいデバイス設定の適用 ( 72 ページ ) を参照してください。

### ロジックコントローラーとデバイスアプリケーションの調整

機械に最初の電源投入がされ、設定がデバイスにダウンロードされると、以下を使用してシステムを調整できます。

- ユーザーパラメーターのオンライン修正
- 次の内蔵 DTM のオンライン修正
  - パラメーターの調整
  - パフォーマンスとエネルギー効率のオートチューニング
  - 微調整用オシロスコープ
  - ...
- DTM のないデバイスの場合は、デバイスで直接手動で調整します。詳細は、デバイスのマニュアルを参照してください。

### FDR サービスの完了

システムを設定したら、FDR サービスを完了してください。この手順では、ロジックコントローラーの FTP サーバーにデバイス設定を保存します。

デバイスに応じて、以下のような複数のツールを使用できます。

- SoMachine
- サードパーティのツール ( 例 : SoMove )
- デバイス Web サーバー
- デバイスで直接 ( 内蔵 HMI 使用 )
- ...

詳細については、デバイスのマニュアルを参照してください。

## アプリケーションのバックアップ

機械の試運転が完了したら運用フェーズの前に、後に使用できるようにプロジェクトをアップロードおよび保存します。

ロジックコントローラーによって、複数の方法があります。

- SoMachine: PC のハードディスクにアプリケーションプログラムをバックアップします。
- ロジックコントローラーの Web サーバー
- ロジックコントローラーのクローン機能 (SD カード)
- ...

詳細については、デバイスのマニュアルを参照してください。

## デバイスを認識させるための準備

### 概要

このステップの目的は、デバイスの IP アドレスの割り当て方式を、ネットワークマネージャー (32 ページ) で設定されている方式と一致するように設定することです。

これは、以下の間に行うことができます。

- 試運転フェーズ (67 ページ参照)
- デバイスの交換 (95 ページ参照)

デバイスに応じて、以下のような異なるツールを使用できます。

- ドライバー: ロータリースイッチやディップスイッチなどがあるデバイス (例: OTB)
- キーパッド (例: ATV)
- PC: 以下で設定する必要のあるデバイス
  - SoMachine
  - サードパーティのソフトウェア
  - その Web サーバー (例: OsiSense XGCS)

IP アドレスの割り当てに応じて、以下の異なる処理を実行できます。

- DHCP: デバイスに DHCP 名を設定
- BOOTP: BOOTP に設定されたデバイス (71 ページ) を参照
- 固定 IP: デバイスに IP アドレスを設定

EtherNet/IP および電子キー (71 ページ) を使用している場合、正しく設定されているか確認してください。

### メインデバイスの設定方式

ツール	IP アドレスの割り当て方式	説明
なし	DHCP	デバイスは、DHCP で正しい DHCP デバイス名であらかじめ設定されています。
ドライバー	DHCP	デバイス (ロータリースイッチ、ディップスイッチなど) にドライバーを使用して、DHCP デバイス名を設定します。 例: Advantys OTB。
	BOOTP	BOOTP のデバイス (ロータリースイッチ、ディップスイッチなど) にドライバーを使用します。 例: XPSMCM。
	固定 IP	デバイス (ロータリースイッチ、ディップスイッチなど) にドライバーを使用して、IP アドレスを設定します。
キーパッド	DHCP	デバイスのキーパッドを使用して、DHCP デバイス名を設定します。 例: ATV32。
	BOOTP	デバイスのキーパッドを使用して、BOOTP のデバイスを設定します。
	固定 IP	デバイスのキーパッドを使用して、IP アドレスを設定します。
PC、タブレットなど	DHCP BOOTP 固定 IP	PC またはタブレットを使用して <b>デバイスの Web サーバー</b> に接続し、ネットワークを設定します。 以下の接続方法を選択します。 <ul style="list-style-type: none"> <li>● PC をデバイスの Ethernet ポートに接続 デバイスの現在の IP アドレスが必要です。</li> <li>● WIFER TCSEGWB13FA0 をデバイスの Ethernet ポートに接続 PC を WIFER に接続します。</li> </ul>
パラメーターの変更を反映するために、デバイスの電源を再投入する必要がある場合があります。		

ツール	IP アドレスの割り当て方式	説明
PC	DHCP BOOTP 固定 IP	SoMachine (DTM を介して) を使用してネットワークを設定します。 PC をデバイスの専用通信ポートに接続します。 例 : ATV32 の Modbus シリアルラインポート 詳細については、DTM を使用した Modbus シリアルラインのデバイスの設定 ( <i>EcoStruxure Machine Expert, Device Type Manager (DTM), User Guide</i> 参照) を参照してください。
	DHCP BOOTP 固定 IP	<b>サードパーティソフトウェア</b> を使用してネットワークを設定します。 以下の接続方法を選択します。 <ul style="list-style-type: none"> <li>● PC をデバイスの Ethernet ポートに接続 デバイスの現在の IP アドレスが必要です。</li> <li>● PC をデバイスの専用通信ポートに接続します。</li> </ul>
パラメーターの変更を反映するために、デバイスの電源を再投入する必要がある場合があります。		

### BOOTP で設定されたデバイス

デバイスの IP アドレスの割り当てが BOOTP の場合、SoMachine を使用してください。

- ネットワークマネージャー (32 ページ参照) で新しいデバイスの MAC アドレスを設定します。
- ロジックコントローラーに新しいアプリケーションを読み込みます。

### EtherNet/IP の電子キー

**電子キー署名**は、デバイスの識別に使用します。

**電子キー**は、デバイスのファームウェア内に含まれているネットワークデバイスの情報です (メーカーコード、製品コードなど)。

スキャナーが起動すると、ネットワークデバイスの電子キーの値とアプリケーションに保存されている値が比較されます。

デバイスの値がアプリケーションの値と同じでない場合、ロジックコントローラーはデバイスと通信しません。

最初の試運転中およびデバイスの交換時に EtherNet/IP スキャナーで電子キーが検証された場合、SoMachine を使用して以下を行うことができます。

- 電子キー (35 ページ) の値の確認、および必要に応じて変更
- ロジックコントローラーに新しいアプリケーションの読み込み

## 正しいデバイス設定の適用

### 概要

デバイスがデバイスネットワーク上で認識されたら、デバイスを設定してください。

これは、以下の間に行うことができます。

- 試運転フェーズ (67 ページ参照)
- デバイスの交換 (95 ページ参照)

### 詳細

正しいデバイス設定を適用するために、デバイスによって異なる処理を行う必要があります。さらに、デバイスで設定情報を考慮させるにはデバイスの電源を再投入する必要がある場合があります。

手順内容	説明
手動による変更なし	<p>デバイスはあらかじめ設定されています。すべて自動化されています。</p> <p>アプリケーションが起動すると、ユーザーパラメーターがデバイスに送信されます。詳細については、ユーザーパラメーターによるデバイス交換 (41 ページ) を参照してください。</p> <p>Advantys OTB の場合、設定のダウンロードはプログラムのみで実行できません。詳細については、開始時の設定の提供 (78 ページ) を参照してください。</p>
SD カード、USB メモリーキー、キーパッドなど	<p>多くの場合、設定を保存するためのメディアは、処理用にすでに準備されています。ただし、新しいデバイスにメディアを挿入する場合は手動の処理が必要な場合があります。</p>
マルチローダー	<p>マルチローダーツールを使用して、以前に保存した設定ファイルをデバイスに読み込みます。</p>
FDR (キーパッドメニューより)	<p>場合によっては、デバイスに FDR サーバーから設定を取得するように Explicit に要求し、その後 FDR サービスを IDLE に戻す必要があります。詳細については、デバイスのマニュアルを参照してください。FDR の詳細については、FDR によるデバイス交換 (40 ページ) を参照してください。</p>
FDR (Web サーバー経由)	<p>Web ブラウザーの使用に対応している PC、スマートフォン、タブレットなどの外部ツールを使用して、デバイス交換に影響を与えます。場合によっては、デバイスに FDR サーバーから設定を取得するように Explicit に要求し、その後 FDR サービスを IDLE に戻す必要があります。</p>
デバイスの Web サーバー (パラメーターによるパラメーター)	<p>Web ブラウザーの使用に対応している PC、スマートフォン、タブレットなどの外部ツールを使用して、設定に影響を与えます。</p>
SoMachine	<p>SoMachine を使用してデバイスに設定をダウンロードします。DTM に対応しているデバイスについては、DTM を使用した Modbus シリアルラインのデバイスの設定 (EcoStruxure Machine Expert, Device Type Manager (DTM), User Guide 参照) を参照してください。</p>
サードパーティのソフトウェア	<p>サードパーティのソフトウェアを使用します。</p>
<p>パラメーターの変更を反映するために、デバイスの電源を再投入する必要がある場合があります。</p>	

デバイス設定の詳細については、デバイスのマニュアルを参照してください。



---

## 第 4 章

### デバイスネットワークの運用

---

#### 概要

この章では、機能、データ交換処理、および動作モードのセキュリティについて説明します。

#### この章について

この章には次の項目が含まれています。

項目	参照ページ
スレーブデバイスの動作モードの管理	74
要求によるデータ交換	76
カスタムサイクリックデータ交換	77
起動時のスレーブデバイスの設定	78
アウトオブプロセスのデータ交換	79
産業用 Ethernet マネージャーの動作モード	81
セキュリティ	84

## スレーブデバイスの動作モードの管理

### 概要

スレーブデバイスの動作モードは、以下のスキャナーと専用ライブラリを備えた産業用 Ethernet マネージャで管理します。

- Modbus Serial IOScanner: Modbus Serial IOScanner ライブラリ (99 ページ参照)
- EtherNet/IP Scanner:
  - EtherNet/IP Scanner ライブラリ (133 ページ参照)
  - EtherNet/IP Explicit メッセージ通信ライブラリ (111 ページ参照)

これらのライブラリによって、以下を行えます。

- Modbus Serial IOScanner の制御
- EtherNet/IP Scanner の制御
- サイクリックデータ交換 (Implicit メッセージ) の管理
- 状態変数の管理
- 非サイクリックデータ交換要求 (Explicit メッセージ) の送信

デバイスによって、その他のライブラリも使用できます。

### Modbus Serial IOScanner の状態変数

状態変数には次の 2 種類があります。

- **ヘルスビット**: チャンネルの通信状態を示す変数。チャンネルごとにヘルスビットが 1 つあります。
- **グローバルスキャナー状態**: Modbus Serial IOScanner の状態を示す変数。

ヘルスビット値を次の表に示します。

ヘルスビット値	チャンネルの通信状態
0	応答を受信せずにヘルスタイムアウトの期限が切れました。
1	エラー未検出。要求および応答が受信されます。

### EtherNet/IP Scanner の状態変数

EtherNet/IP Scanner のあらかじめ定義された状態変数はありません。

EtherNet/IP Target (スレーブ) デバイスのヘルスビットを可視化するには、以下を使用してください。

- EipGetHealth ファンクションブロック (136 ページ参照)
- EIPGetHealthBit ファンクションブロック (125 ページ参照)

### I/O イメージ変数

スキャナーは、デバイスからのデータ収集や、デバイスへのデータ書き込みをします。これらの変数で I/O イメージを構成します。

### 変数アドレス

各変数は、独自のアドレスを取得します。

変数	データ型	数
I/O イメージ変数	入力用は %IW 出力用は %QW	チャンネル / 接続ごとにワードの表が作成されます。
ヘルスビット	%IW	Modbus TCP 用は 4 つの連続したワード EtherNet/IP 用はなし
グローバルスキャナー状態	%IW	Modbus TCP 用はワード 1 つ EtherNet/IP 用はなし
EtherNet/IP については、EtherNet/IP Scanner の状態変数 (74 ページ) を参照してください。		

### Modbus Serial IOScanner を制御するファンクションブロック

Modbus Serial IOScanner ライブラリには、アプリケーションがロジックコントローラーおよび Modbus TCP スレーブとの通信に使用するファンクションブロックが含まれています。

- CONFIGURE\_OTB: Advantys OTB のソフトウェア設定の送信
- IOS\_GETSTATE: Modbus Serial IOScanner の状態の読み込み

- IOS\_START: Modbus Serial IOScanner の起動
- IOS\_GETHEALTH: ヘルスビット値の読み込み
- IOS\_STOP: Modbus Serial IOScanner の停止

詳細については、Modbus Serial IOScanner (99 ページ) を参照してください。

### EtherNet/IP Scanner を制御するファンクションブロック

EtherNet/IP Scanner ライブラリには、アプリケーションがロジックコントローラーおよび EtherNet/IP Target (スレーブ) デバイスとの通信に使用するファンクションブロックが含まれています。

- EipDataExchange: デバイスに Explicit メッセージを送信
- EipControl: EtherNet/IP Scanner の通信の開始 / 停止
- EipGetHealth: ヘルスビット値の読み込み

詳細については、EtherNet/IP Scanner (133 ページ) を参照してください。

### EtherNet/IP Explicit メッセージ通信用ファンクションブロック

EtherNet/IP Explicit メッセージ通信ライブラリには、アプリケーションが EtherNet/IP Explicit メッセージの送信に使用するファンクションブロックが含まれています。

- Get\_Attribute\_All: オブジェクトのすべての属性の取得
- Set\_Attribute\_All: インスタンスまたはクラスのすべての属性の設定
- Get\_Attribute\_Single: オブジェクトの属性の取得
- Set\_Attribute\_Single: クラスの属性の設定
- EIPStartConnection: 接続の開始
- EIPStartAllConnection: すべての接続の開始
- EIPStopConnection: 接続の停止
- EIPStopAllConnections: すべての接続の停止
- EipGetHealth: ヘルスビット値の読み込み

詳細については、EtherNet/IP Explicit メッセージ通信ライブラリ (111 ページ) を参照してください。

### ATV および Lexium デバイスを制御するファンクションブロック

ATV と Lexium デバイスを制御するには PLC Open およびドライブ専用の他のファンクションブロックを使用してください。これらのファンクションブロックは、GMC 独立 PLCopen MC ライブラリ、GMC 独立 Altivar ライブラリ、および GMC 独立 Lexium ライブラリからアクセスできます。詳細については、Motion Control Library Guide を参照してください。

### バス周期タスク

産業用 Ethernet スキャナーとスレーブデバイスは、アプリケーションタスクの周期ごとにデータを交換します。

バス周期タスクのパラメーターによって、スキャナーを管理するアプリケーションタスクを選択できます。

- **Use parent bus cycle setting:** スキャナーをコントローラーを管理するアプリケーションタスクに関連付けます。
- **MAST:** スキャナーを MAST タスクに関連付けます。
- その他の既存タスク: 既存タスクを選択し、スキャナーに関連付けできます。

アプリケーションタスクの詳細については、SoMachine プログラミングガイド (SoMachine, プログラミングガイド 参照) を参照してください。

## 要求によるデータ交換

### 詳細

サイクリック (Implicit) データ交換は、選択した産業用 Ethernet マネージャーで管理します。

要求に応じてデータ交換をするには、Explicit メッセージを使用してください。

Explicit メッセージは、アプリケーションで次のファンクションブロックを使用することで開始されます。

- EtherNet/IP デバイスの場合、EtherNet/IP Explicit メッセージ通信ライブラリ (111 ページ) のファンクションブロックを使用できます。
- EtherNet/IP デバイスの場合、EtherNet/IP Scanner ライブラリの汎用 EipDataExchange ファンクションブロック (137 ページ) も使用できます。
- Modbus TCP デバイスの場合、READ\_VAR および WRITE\_VAR ファンクションブロックを使用できます。
- TCP/UDP デバイスの場合、ファンクションブロック (147 ページ) を使用できます。

## カスタムサイクリックデータ交換

### 詳細

プロジェクトに定義済みのデバイスが追加されると、自動的にサイクリックデータ交換が作成されます。

さらに、追加で各スレーブデバイスにサイクリックデータ交換 ([43 ページ](#)) を作成できます。

## 起動時のスレーブデバイスの設定

### 詳細

デバイスの保守を容易にするために、スレーブデバイスに設定データを送信できます。

アプリケーションの起動時に、以下によって自動的にデバイス設定を送信できます。

- ユーザーパラメーター (41 ページ) (アプリケーションが接続を開始したとき)
- 設定アセンブリ (49 ページ) (この機能に対応しているデバイスの場合)

さらに、CONFIGURE\_OTB ファンクションブロック (105 ページ) を使用したアプリケーションからの要求によって Advantys OTB デバイスの設定を送信することもできます。

## アウトオブプロセスのデータ交換

### 概要

制御ネットワークとデバイスネットワーク間のデータ交換は、アウトオブプロセスのデータ交換である場合があります。例えば、デバイスネットワーク上の対象のデバイスと通信するために、監視ソフトウェアまたはサードパーティの設定ツールを使用する場合です。

産業用 Ethernet ネットワークでは、アウトオブプロセスのデータ交換を許可しています。

アウトオブプロセスのデータ交換を有効にするには、以下を行います。

- デバイスのゲートウェイアドレス (34 ページ) を設定
- IP 転送サービス が有効であることを確認
- PC のルーティングを確認 (以下を参照)

#### 注記：

以下のソースからのアウトオブプロセスデータ交換は、ロジックコントローラーのパフォーマンスに影響を与える可能性があります。

- DTM、Web サーバー、および Modbus TCP 要求
- ネットワーク通信 (DTM、PC がネットワーク上にある場合の Web サーバー通信)
- TcpUdpCommunications ライブラリによって生成された TCP UDP 通信

ネットワークを使用して DTM をデバイスに接続すると、DTM は実行中のアプリケーションと同時に通信します。システムの全体のパフォーマンスに影響を与え、ネットワークに過負荷が掛かる場合があるため、制御中のデバイス間のデータの一貫性に影響がでる可能性があります。

### 警告

#### 装置の意図しない動作

DTM がパフォーマンスに悪影響を与える場合、実行中のアプリケーションでデバイスネットワークを介して通信する DTM は接続しないでください。

上記の指示に従わないと、死亡、重傷、または物的損害を負う可能性があります。

### PC のルーティング

監視ソフトウェアまたは設定ツールに対応している PC は、スレーブデバイスと通信するための設定をしてください。PC は、コントローラーの Ethernet ポートの 1 つと同じサブネットマスク上にある必要があります。

スレーブデバイスの設定	結果
FDT/DTM によってあらかじめ定義されたスレーブとして設定されている	PC の特定のパラメーター化は必要ありません。 <b>注記：</b> PC の設定は変更されません。
その他のツールを使用して設定されている	PC がスレーブデバイスと同じサブネットにない場合は、PC のルーティングテーブルを更新してください (以下参照)。

PC のルーティングテーブルを更新するには、PC からコントローラーおよびその他のデバイスへの接続をすべて停止します。その後、Windows のコマンドプロンプトで次のコマンドを実行します。

```
route ADD 設定先 MASK サブネットマスク ゲートウェイ
```

パラメーター	値
設定先	産業用 Ethernet ネットワークの IP アドレス
サブネットマスク	産業用 Ethernet ネットワークのサブネットマスク
ゲートウェイ	制御ネットワークに接続されたコントローラーのポートの IP アドレス

例えば、TM251MESE が以下である場合

- PC の IP アドレス : 192.168.0.2
- PC のサブネットマスク : 255.255.0.0
- 産業用 Ethernet ネットワークの IP アドレス : 10.10.0.0
- 産業用 Ethernet ネットワークのサブネットマスク : 255.255.252.0

- 制御ネットワークのポート“Ethernet\_1”のIPアドレス：192.168.0.5
- 制御ネットワークのポート“Ethernet\_1”のサブネットマスク：255.255.0.0

これに対応するコマンドは以下のようになります。

```
route ADD 10.10.0.0 MASK 255.255.252.0 192.168.0.5
```

パラメーターを検証するには、以下のコマンドを実行します。

```
route PRINT
```

PC からルートを削除するには、以下のコマンドを実行します。

```
route DELETE 設定先
```

設定先は、以前に入力した産業用 Ethernet ネットワークの IP アドレスです。



## 産業用 Ethernet マネージャーの動作モード

### 産業用 Ethernet マネージャーの状態

デバイスの動作モードを管理するために、産業用 Ethernet マネージャーは以下で構成されています。

- Modbus Serial IOScanner
- EtherNet/IP Scanner

産業用 Ethernet マネージャーの状態によって、デバイスネットワークの様々なデバイスの動作が定義されています。各状態において、監視情報 (ヘルスビット、通信状態など) が特定されます。

スキャナーの状態は、ロジックコントローラーの状態によって異なります。

ロジックコントローラーの状態	Modbus Serial IOScanner の状態	EtherNet/IP Scanner の状態
EMPTY	IDLE	IDLE
CONFIGURED	STOPPED	STOPPED
STOPPED	STOPPED	OPERATIONAL
HALT	STOPPED	特定の動作で OPERATIONAL
RUNNING	OPERATIONAL	OPERATIONAL
RUNNING with breakpoint	特定の動作で OPERATIONAL	特定の動作で OPERATIONAL

### ロジックコントローラーの EMPTY 状態

TCP/IP 接続が閉じています。

デバイスの状態は、個別の動作モードに応じて管理されます。

Modbus Serial IOScanner および EtherNet/IP Scanner は作成されません (IDLE 状態)。ヘルスビットおよび I/O イメージはありません。

### ロジックコントローラーの CONFIGURED 状態

TCP/IP 接続が閉じています。

ロジックコントローラーは、以下の後に CONFIGURED 状態に入ります。

- アプリケーションの読み込み
- SoMachine によって送信されたりセット (コールド / ウォーム)

Modbus Serial IOScanner が STOPPED 状態の場合、Modbus TCP スレーブデバイスのすべてのチャンネルがハーフサイドモードで閉じています。

EtherNet/IP Scanner が STOPPED 状態の場合、発信先デバイスのすべての接続が閉じています。

### ロジックコントローラーの STOPPED 状態

Modbus Serial IOScanner は STOPPED 状態です。Modbus TCP スレーブデバイスのすべてのチャンネルがハーフサイドモードで閉じています。

スレーブデバイスは、個別の動作モードに応じて管理されます。

Modbus Serial IOScanner 用の SoMachine 変数を次の表に示します。

変数	値	コメント
ヘルスビット値	0	-
入力イメージ	0 または最後の読み込み値	入力値は <b>エラー処理</b> のパラメーターによって異なります。入力値は、ロジックコントローラーが STOPPED 状態に入ったときの値です。その後の入力の実際の状態を反映していない可能性があります。
出力イメージ	0 または最後の書き込み値	出力値は、 <b>Behavior for outputs in Stop</b> のパラメーターによって異なります。出力値は、その後の出力の実際の状態を反映していない可能性があります。

EtherNet/IP Scanner は、OPERATIONAL 状態のままです。すべての Originator (マスター) / Target (スレーブ) の接続もアクティブなままです。Target (スレーブ) デバイスとスキャナーの間のデータ交換は継続されます。

EtherNet/IP Scanner 用の SoMachine 変数を次の表に示します。

変数	値	コメント
入力イメージ	読み込み値	値は、EtherNet/IP Scanner を動作させるタスクと同期して更新されます。
出力イメージ	最後の書き込み値 または初期値	出力は、初期値に設定されるか、または現在値が維持されます (Behavior for outputs in Stop のパラメーターによる)。出力値は、その後の出力の実際の状態を反映していない可能性があります。接続の送信形式 (57 ページ) を参照してください。

**警告**

**メモリーの出力値が物理的状态と異なる可能性**  
 コントローラーが RUNNING 状態でないときは、物理的出力の状態がメモリーの値とは限りません。  
 上記の指示に従わないと、死亡、重傷、または物的損害を負う可能性があります。

**ロジックコントローラーの HALT 状態**

Modbus Serial IOScanner の場合は、ロジックコントローラーの STOPPED 状態と同じ動作です。

HALT 状態のタスクが EtherNet/IP バス周期タスク ( デフォルトは MAST) の場合の EtherNet/IP Scanner 用の SoMachine 変数を次の表に示します。

変数	値	コメント
入力イメージ	最後の読み込み値	入力値は、ロジックコントローラーが HALT 状態に入ったときの値です。その後の入力の実際の状態を反映していない可能性があります。
出力イメージ	最後の書き込み値 または初期値	出力は、初期値に設定されるか、または現在値が維持されます (Behavior for outputs in Stop のパラメーターによる)。出力値は、その後の出力の実際の状態を反映していない可能性があります。

HALT 状態のタスクがその他のタスクである場合の EtherNet/IP Scanner 用の SoMachine 変数を次の表に示します。

変数	値	コメント
入力イメージ	最後の読み込み値	値は、EtherNet/IP Scanner を動作させるタスクと同期して更新されます。
出力イメージ	最後の書き込み値 または初期値	出力は、初期値に設定されるか、または現在値が維持されます (Behavior for outputs in Stop のパラメーターによる)。出力は各サイクルごとに書き込まれます。出力値は、その後の出力の実際の状態を反映していない可能性があります。接続の送信形式 (57 ページ) を参照してください。出力へのオンライン修正は利用できません。

**警告**

**メモリーの出力値が物理的状态と異なる可能性**  
 コントローラーが RUNNING 状態でないときは、物理的出力の状態がメモリーの値とは限りません。  
 上記の指示に従わないと、死亡、重傷、または物的損害を負う可能性があります。

### ロジックコントローラーの RUNNING 状態

TCP/IP 接続は開いています。

スレーブデバイスは、ロジックコントローラーによって管理されます。

SoMachine の変数を次の表に示します。

変数	値	コメント
ヘルスビット値	0...1	0: タイムアウトする前にデバイスからの応答がない 1: タイムアウトする前に要求の送信と応答がある
入力イメージ	最後の読み込み値	値はスキャナーを動作させるタスクと同期して更新されます。
出力イメージ	最後の書き込み値	値はアプリケーションによって管理されます。

### ロジックコントローラーのブレイクポイント付き RUNNING 状態

TCP/IP 接続は開いています。

スレーブデバイスは、ロジックコントローラーによって管理されます。

ブレイクポイント付きで RUNNING 状態のタスクが EtherNet/IP バス周期タスク ( デフォルトは MAST) の場合の EtherNet/IP Scanner 用の SoMachine 変数を次の表に示します。

変数	値	コメント
入力イメージ	最後の読み込み値	入力値は、ロジックコントローラーがブレイクポイント付きで RUNNING 状態に入ったときの値です。その後の入力の実際の状態を反映していない可能性があります。
出力イメージ	最後の書き込み値 または初期値	出力は、現在値に維持されます。 出力値は、その後の出力の実際の状態を反映していない可能性があります。

ブレイクポイント付きで RUNNING 状態のタスクがその他のタスクである場合の EtherNet/IP Scanner 用の SoMachine 変数を次の表に示します。

変数	値	コメント
入力イメージ	最後の読み込み値	入力値は、ロジックコントローラーがブレイクポイント付きで RUNNING 状態に入ったときの値です。その後の入力の実際の状態を反映していない可能性があります。
出力イメージ	最後の書き込み値 または初期値	出力は、現在値に維持されます。 出力値は、その後の出力の実際の状態を反映していない可能性があります。 接続の送信形式 (57 ページ) を参照してください。

## 警告

### メモリーの出力値が物理的状态と異なる可能性

コントローラーが RUNNING 状態でないときは、物理的出力の状態がメモリーの値とは限りません。

上記の指示に従わないと、死亡、重傷、または物的損害を負う可能性があります。

## セキュリティ

### 概要

一部の特定の機能によって、デバイスの交換におけるシステムのセキュリティレベルを上げることができます。

- マスター IP アドレス
- 電子キー

### マスター IP アドレスの説明

一部のデバイスには**マスター IP アドレス**パラメーターがあり、宣言された 1 台のマスターロジックコントローラーのみがデバイスにアクセスできます。

詳細については、マスター IP アドレスパラメーター ([38 ページ](#)) を参照してください。

### 電子キーの説明

**電子キー署名**は、デバイスの識別に使用します。

**電子キー**は、デバイスのファームウェア内に含まれているネットワークデバイスの情報です (メーカーコード、製品コードなど)。

スキャナーが起動すると、デバイスの電子キーの値とアプリケーションに保存されている値が比較されます。

デバイスの値がアプリケーションの値と同じでない場合、ロジックコントローラーはデバイスと通信しません。

詳細については、EtherNet/IP の電子キ ([35 ページ](#)) を参照してください。

---

## 第 5 章

### デバイスネットワーク診断

---

#### 概要

この章では、トラブルシューティングの情報について説明します。

#### この章について

この章には次の項目が含まれています。

項目	参照ページ
ネットワークテスト	86
診断 : Web サーバー	87
診断 : SoMachine オンラインモード	90
トラブルシューティング	93

## ネットワークテスト

### 目的

産業用 Ethernet マネージャーを操作する前に、ネットワークをテストしてください。

以下を確認します。

- 各デバイスのアドレス設定が、ネットワークの計画と一致している。
- 各デバイスが正しく配線されている。

標準的なテスト方法を以下に示します。

### ステータス LED

デバイスに応じて、ステータス LED に配線が正しいことが表示されているか確認します。

### PC を使用した検証

PC を使用して、各ネットワークデバイスが接続およびアドレス指定されていることを確認します。

手順	手順内容
1	PC を産業用 Ethernet ネットワークに接続します。
2	コマンドプロンプトにアクセスします。
3	ping xxx.xxx.xxx.xxx コマンドを使用して、各ネットワークデバイスに到達します。 xxx.xxx.xxx.xxx はテストするデバイスの IP アドレスです。 <b>注記：</b> ping -h コマンドで、ping コマンドのヘルプが表示されます。

### Web サーバーを使用した検証

ロジックコントローラーの Web サーバーを使用して、ロジックコントローラーが各ネットワークデバイスと通信できることを確認します。

手順	手順内容
1	ロジックコントローラーの Web サーバーにアクセスします。
2	<b>Ethernet 診断</b> ページを開きます。
3	各デバイスの <b>Remote ping</b> サービス を使用します。

## 診断 : Web サーバー

### 概要

ロジックコントローラーの Web サーバーには診断タブがあります。



このタブから産業用 Ethernet 診断ページにアクセスできます。

- **Ethernet** 診断ページ (87 ページ参照)
- **Modbus TCP** 診断ページ (88 ページ参照)
- **EtherNet/IP** 診断ページ (89 ページ参照)

### Ethernet のページ

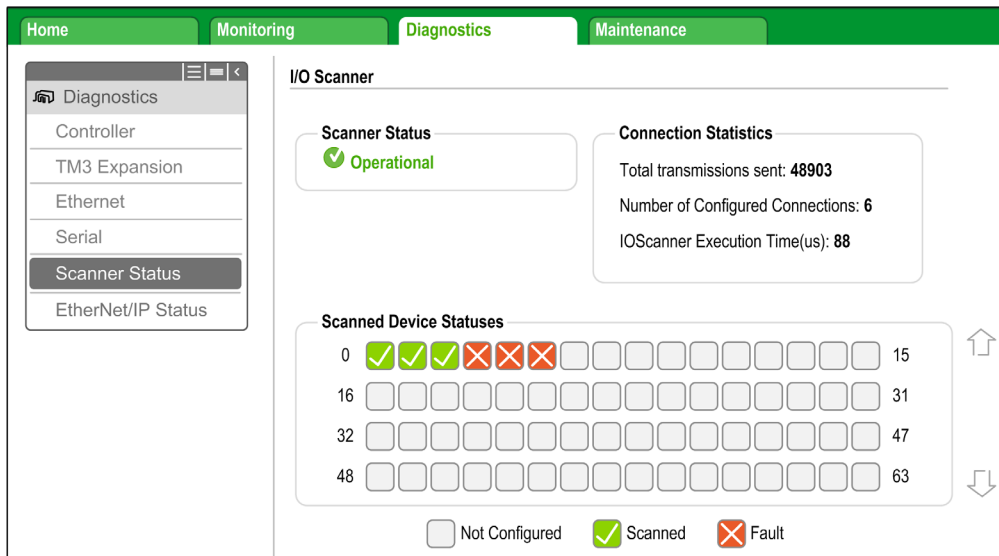
**Ethernet** をクリックすると、ロジックコントローラーの Ethernet 情報が表示され、特定の IP アドレスへの通信をテストできます。

**Ethernet** ページのテスト結果を次の表に示します。

アイコン	意味
	通信テストが成功しました。
	ロジックコントローラーは、指定された IP アドレスと通信できません。

Modbus TCP 状態のページ

Scanner Status をクリックすると、Modbus Serial IOScanner の状態 (IDLE、STOPPED、OPERATIONAL) および最大 64 台の Modbus TCP スレーブデバイスのヘルスビットが表示されます。



0...63 は、チャンネル ID に対応しています。

Scanner Status のページに表示される各チャンネルの状態を次の表に示します。

アイコン	ヘルスビット値	意味	スキャナーの状態
	1	要求および応答は、時間通りに実行しています。	OPERATIONAL
	0	エラーが検出されました。通信が閉じています。	OPERATIONAL
	-	この ID は設定されたチャンネルに対応していません。	OPERATIONAL
	0	通信が閉じています。	STOPPED

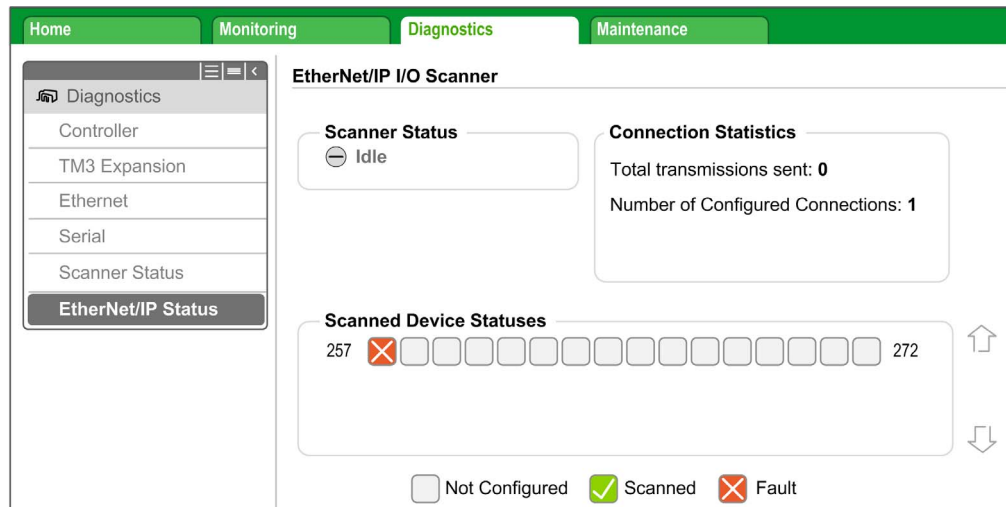
注記：アイコンをクリックすると、デバイスの Web サーバーが開きます (Web サーバーがある場合)。Web サーバーにアクセスするには、コンピューターをデバイスと通信させてください。詳細については、PC ルーティング (79 ページ) を参照してください。

Modbus Serial IOScanner の状態が IDLE の場合、アイコンは表示されません。No scanned device reported と表示されます。



**EtherNet/IP 状態のページ**

**EtherNet/IP Status** をクリックすると、EtherNet/IP Scanner の状態 (IDLE、STOPPED、OPERATIONAL) および最大 16 台の EtherNet/IP Target (スレーブ) デバイスのヘルスビットが表示されます。



257...272 は、接続 ID に対応しています。

**EtherNet/IP Status** のページに表示される各接続の状態を次の表に示します。

アイコン	ヘルスビット値	意味	スキャナーの状態
	1	通信は時間通りに実行しています。	STOPPED または OPERATIONAL
	0	エラーが検出され、通信が閉じています。	STOPPED または OPERATIONAL
	-	この ID は設定された接続に対応していません。	STOPPED または OPERATIONAL

**注記:** アイコンをクリックすると、ネットワークデバイスの Web サーバーが開きます (Web サーバーがある場合)。Web サーバーにアクセスするには、コンピューターをデバイスと通信させてください。詳細については、PC ルーティング (79 ページ) を参照してください。

EtherNet/IP Scanner の状態が IDLE の場合、アイコンは表示されません。**No scanned device reported** と表示されます。

## 診断 : SoMachine オンラインモード



### 概要

オンラインモードでは、以下を使用して SoMachine の産業用 Ethernet マネージャーを監視できます。

- デバイスツリーのアイコン
- 産業用 Ethernet マネージャーおよびデバイスの状態タブ
- Modbus Serial IOScanner の産業用 Ethernet マネージャーの **IOScanner I/O マッピング** タブ
- EtherNet/IP Target (スレーブ) デバイスのヘルスビット変数のビジュアライゼーション
- デバイスの I/O マッピングタブ
- 産業用 Ethernet マネージャーのリソースタブ

### デバイスツリー

産業用 Ethernet マネージャーおよびデバイスの通信状態は、**デバイスツリー**にアイコンで表示されま  
す。

アイコン	意味
	デバイスとの通信は正常です。 <b>注記</b> : 産業用 Ethernet マネージャーには、常にこのアイコンが表示されます。
	ロジックコントローラーがデバイスと通信できません。 <b>注記</b> : 産業用 Ethernet マネージャーが STOPPED 状態のときは、すべてのデバイスにこのア イコンが表示されます。

### 産業用 Ethernet マネージャーの I/O マッピング

産業用 Ethernet マネージャーの **IOScanner I/O マッピング** タブで、Modbus Serial IOScanner の状態お  
よび Modbus TCP スレーブデバイスのヘルスビットを監視できます。

IOScanner I/O Mapping								
Channels								
Variable	Mapping	Channel	Address	Type	Defaul...	Curren...	Prepar...	
Diagnostic								
Global St...		Global St...	%I...	UINT	0	2		
Healthbits								
Healthbit...		Healthbit...	%I...	WORD		63		
Healthbits_OTB1EODM9LP		Bit 0	%IX...	BOOL	FALSE	TRUE		
Healthbits_Altivar32		Bit 1	%IX...	BOOL	FALSE	TRUE		
Healthbits_Lexium32M		Bit 2	%IX...	BOOL	FALSE	TRUE		
Healthbits_Generic_Slave_channel3		Bit 3	%IX...	BOOL	FALSE	TRUE		
Healthbits_Generic_Slave_channel4		Bit 4	%IX...	BOOL	FALSE	TRUE		
Healthbits_Generic_Slave_channel5		Bit 5	%IX...	BOOL	FALSE	TRUE		
		Bit 6	%IX...	BOOL	FALSE	FALSE		
		Bit 7	%IX...	BOOL	FALSE	FALSE		
		Bit 8	%IX...	BOOL	FALSE	FALSE		
		Bit 9	%IX...	BOOL	FALSE	FALSE		
		Bit 10	%IX...	BOOL	FALSE	FALSE		
		Bit 11	%IX...	BOOL	FALSE	FALSE		
		Bit 12	%IX...	BOOL	FALSE	FALSE		
		Bit 13	%IX...	BOOL	FALSE	FALSE		
		Bit 14	%IX...	BOOL	FALSE	FALSE		
		Bit 15	%IX...	BOOL	FALSE	FALSE		
		Healthbit...	%I...	WORD		0		
		Healthbit...	%I...	WORD		0		
		Healthbit...	%I...	WORD		0		

欄		用途	コメント
変数	診断	グローバルスキャナー状態変数に名前を割り当てます。	-
	ヘルスビット	各ヘルスビットに名前を割り当てます。 例えば、ヘルスビットに関連するデバイス名を付けます。	ヘルスビットは、16 ビットの 4 つのサブフォルダーにグループ化されます。
アドレス		各変数のアドレスを取得します。	設定を変更すると、アドレスも変更される場合があります。
現在値		Modbus TCP デバイスを監視します。	ブール値 (ヘルスビット) の場合 ● TRUE = 1 ● FALSE = 0

### EtherNet/IP Target (スレーブ) デバイスのヘルスビット

EtherNet/IP Target (スレーブ) デバイスのヘルスビットを監視するには、以下を行います。

- アプリケーションでビジュアライゼーションを作成
- ビジュアライゼーションに以下のヘルスビット変数を追加
  - EipGetHealth ファンクションブロック (136 ページ参照)
  - EIPGetHealthBit ファンクションブロック (125 ページ参照)

### スレーブデバイスのマッピング

産業用 Ethernet デバイスには、デバイスの I/O を表示する I/O マッピングタブがあります。

注記：汎用 TCP/UDP には、I/O マッピングタブはありません。

Advantys OTB スレーブデバイスの I/O マッピングタブの例を次の図に示します。

Variable	Mapping	Channel	Address	Type	Default Value	Current Value
<b>Inputs</b>						
iwOTB1EODM9LP_Read_Inputs	[Icon]	Read Inputs	%IW18	WORD		2049
		Bit 0	%IX3...	BOOL	FALSE	TRUE
		Bit 1	%IX3...	BOOL	FALSE	FALSE
		Bit 2	%IX3...	BOOL	FALSE	FALSE
		Bit 3	%IX3...	BOOL	FALSE	FALSE
		Bit 4	%IX3...	BOOL	FALSE	FALSE
		Bit 5	%IX3...	BOOL	FALSE	FALSE
		Bit 6	%IX3...	BOOL	FALSE	FALSE
		Bit 7	%IX3...	BOOL	FALSE	FALSE
		Bit 8	%IX3...	BOOL	FALSE	FALSE
		Bit 9	%IX3...	BOOL	FALSE	FALSE
		Bit 10	%IX3...	BOOL	FALSE	FALSE
Bit 11	%IX3...	BOOL	FALSE	TRUE		
<b>Outputs</b>						
qwOTB1EODM9LP_Output_commands	[Icon]	Output co...	%QW1...	WORD		255
		Bit 0	%QX2.0	BOOL	FALSE	TRUE
		Bit 1	%QX2.1	BOOL	FALSE	TRUE
		Bit 2	%QX2.2	BOOL	FALSE	TRUE
		Bit 3	%QX2.3	BOOL	FALSE	TRUE
		Bit 4	%QX2.4	BOOL	FALSE	TRUE
		Bit 5	%QX2.5	BOOL	FALSE	TRUE
		Bit 6	%QX2.6	BOOL	FALSE	TRUE
Bit 7	%QX2.7	BOOL	FALSE	TRUE		

欄		用途	コメント
変数	入力	デバイスの各入力に名前を割り当てます。	各ビットもマッピングできます。
	出力	デバイスの各出力に名前を割り当てます。	
アドレス		各変数のアドレスを取得します。	設定を変更すると、アドレスも変更される場合があります。
現在値		デバイス入力のリアルタイム値に従います。出力値は動的に変更されます。	ブール値 (各ビット) の場合 <ul style="list-style-type: none"> <li>● TRUE = 1</li> <li>● FALSE = 0</li> </ul>

## トラブルシューティング

### 主な問題

症状	考えられる原因	解決策
デバイスツリーの産業用 Ethernet マネージャーに赤い三角が表示されている。	設定がロジックコントローラーのバージョンに準拠していません。	<ul style="list-style-type: none"> <li>● <b>ビルド→すべてクリーン</b></li> <li>● <b>ビルド→すべて再ビルド</b></li> <li>● ロジックコントローラーに最新バージョンのファームウェアがあることを確認します。</li> </ul>
デバイスツリーのデバイスに赤い三角が表示されている。	ロジックコントローラーがデバイスと通信できません。	<ul style="list-style-type: none"> <li>● デバイスの配線および電源を確認します。</li> <li>● デバイスの IP アドレスを確認します ( デバイスの IP アドレスに Remote ping service を使用 )。</li> <li>● デバイスが読み込み / 書き込み要求に対応しているかを確認します。</li> <li>● アクセスされたレジスターがこのデバイスに関連しているかを確認します。</li> <li>● アクセスされたレジスターが書き込み禁止されていないかを確認します。</li> <li>● FDR (高速デバイス交換) サービスがデバイスで正しく設定されていることを確認します。</li> <li>● <b>マスター IP アドレス</b>のパラメーターがデバイスで正しく設定されていることを確認します。</li> <li>● <b>電子キー</b>のパラメーターがデバイスで正しく設定されていることを確認します。</li> </ul>
デバイス / チャンネルが一時的に赤色で表示されている。	配線が不安定です。	配線を確認します。
	設定の調整が必要です。	<ul style="list-style-type: none"> <li>● ヘルスタイムアウトの値を増やします。</li> <li>● 繰返し率の値を増やします。</li> </ul>
	産業用 Ethernet マネージャーの負荷が大き過ぎます。	<b>スキャナーリソースタブ (62 ページ)</b> を確認します。
デバイスの状態の一部がアプリケーションに表示されていない。	Modbus TCP スレーブデバイスの場合、繰返し率が遅過ぎます ( 値が大き過ぎます )。	デバイスに関連付けられたチャンネルの繰返し率を減らします。
	EtherNet/IP Target ( スレーブ ) デバイスの場合、RPI 値が遅過ぎます ( 値が大き過ぎます )。	デバイスに関連付けられた接続の RPI 値を減らします。
	バス周期タスクの速さが十分でない。	<ul style="list-style-type: none"> <li>● スキャナーを別のタスク (Modbus Serial IOScanner または EtherNet/IP Scanner) に関連付けます。</li> <li>● 関連付けられたタスクのサイクル値を減らします。</li> </ul>



---

## 第 6 章

### 保守

---

#### 保守の概要

##### 主な手順

デバイスを交換する場合の主な手順は次のとおりです。

- 機械および影響を受ける機械の部分の電源を切断
- デバイスの取り外し
- 新しいデバイスの取り付け
- 新しいデバイスの電源の投入
- システムで認識されるようにデバイスを準備 (70 ページ参照)
- 正しいデバイス設定の適用 (72 ページ参照)
- デバイス交換の確認 (アプリケーションによって異なります)







## 付録について

付録には次の章が含まれています。

章	章タイトル	参照ページ
A	Modbus Serial IOScanner ライブラリ	99
B	EtherNet/IP Explicit メッセージ通信ライブラリ	111
C	EtherNet/IP Scanner ライブラリ	133
D	モーション制御ライブラリ	145
E	汎用 TCP/UDP ライブラリ	147
F	ファンクションおよびファンクションブロックの表現	149



---

# 付録 A

## Modbus Serial IOScanner ライブラリ

---

### 概要

この章では、ModbusTCPIOScanner ライブラリについて説明します。

### この章について

この章には次のセクションが含まれています。

セクション	項目	参照ページ
A.1	Modbus Serial IOScanner ファンクション	100
A.2	Modbus Serial IOScanner データ型	107

# A.1

## Modbus Serial IOScanner ファンクション

### 概要

このセクションでは、ModbusTCPIOScanner ライブラリに含まれているファンクションについて説明します。

### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
IOS_GETSTATE: Modbus Serial IOScanner の状態の読み込み	101
IOS_START: Modbus Serial IOScanner の起動	102
IOS_GETHEALTH: ヘルスビット値の読み込み	103
IOS_STOP: Modbus Serial IOScanner の停止	104
CONFIGURE_OTB: Advantys OTB のソフトウェア設定の送信	105

## IOS\_GETSTATE: Modbus Serial IOScanner の状態の読み込み

### ファンクションの説明

このファンクションは、Modbus Serial IOScanner の状態に対応する値を返します。

図



### IL および ST による表記

IL または ST 言語での一般的な表記については、ファンクションおよびファンクションブロックの表記 (149 ページ) を参照してください。

### I/O 変数の説明

出力変数を次の表に示します。

出力	データ型	コメント
IOS_GETSTATE	IosStateCodes (108 ページ参照)	戻り値: IosStateCodes enum

例

以下は、このファンクションの呼び出しの例です。

```
mystate := IOS_GETSTATE () ; (* 0=NOT CONFIGURED 2=OPERATIONAL or 3=STOPPED. *)
```

## IOS\_START: Modbus Serial IOScanner の起動

### ファンクションの説明

このファンクションは、Modbus Serial IOScanner を起動させます。

これにより、Modbus Serial IOScanner の実行をランタイムで制御できます。デフォルトでは、アプリケーションの開始時に自動的に Modbus Serial IOScanner が起動します。

このファンクションの呼び出しは、Modbus Serial IOScanner が物理的に起動するのを最長で 5 ms 間待ちます。

Modbus Serial IOScanner がすでに起動している場合は何もおきません。

図



### IL および ST による表記

IL または ST 言語での一般的な表記については、ファンクションおよびファンクションブロックの表記 (149 ページ) を参照してください。

### I/O 変数の説明

出力変数を次の表に示します。

出力	データ型	コメント
IOS_START	UDINT	<ul style="list-style-type: none"> <li>● 0 = 正常に起動</li> <li>● その他の値 = 起動に失敗</li> </ul>

例

以下は、このファンクションの呼び出しの例です。

```
rc := IOS_START () ;
```

```
IF rc <> 0 THEN (* Abnormal situation to be processed at application level *)
```

## IOS\_GETHEALTH: ヘルスビット値の読み込み

### ファンクションの説明

このファンクションは、特定のチャンネルのヘルスビット値を返します。



### IL および ST による表記

IL または ST 言語での一般的な表記については、ファンクションおよびファンクションブロックの表記 (149 ページ) を参照してください。

### I/O 変数の説明

入力変数を次の表に示します。

入力	データ型	コメント
channelID	UINT	監視するチャンネルのチャンネル ID

出力変数を次の表に示します。

出力	データ型	コメント
IOS_GETHEALTH	UINT	<ul style="list-style-type: none"> <li>0: チャンネルの I/O 値は更新されていません。</li> <li>1: チャンネルの I/O 値が更新されています。</li> </ul>

### 例

以下は、このファンクションの呼び出しの例です。

```
chID := 1 ;
```

```
channelHealth := IOS_GETHEALTH(chID) (* Get the health value (1=OK, 0=Not OK) of the channel number chID. The channel ID is displayed in the configuration editor of the device *)
```

## IOS\_STOP: Modbus Serial IOScanner の停止

### ファンクションの説明

このファンクションは、Modbus Serial IOScanner を停止させます。

これにより、Modbus Serial IOScanner の実行をランタイムで制御できます。デフォルトでは、コントローラーが STOPPED 状態になると、Modbus Serial IOScanner は停止します。

最初の電源投入からすべてのネットワークデバイスが動作可能になるまで、Modbus Serial IOScanner が停止している必要があります。

このファンクションの呼び出しは、Modbus Serial IOScanner が物理的に停止するまで待つため 5 ms 掛かる場合があります。

すでに停止している Modbus Serial IOScanner を停止しても何も起きません。



### IL および ST による表記

IL または ST 言語での一般的な表記については、ファンクションおよびファンクションブロックの表記 (149 ページ) を参照してください。

### I/O 変数の説明

出力変数を次の表に示します。

出力	データ型	コメント
IOS_STOP	UDINT	<ul style="list-style-type: none"> <li>● 0 = 正常に停止</li> <li>● その他の値 = 停止に失敗</li> </ul>



以下は、このファンクションの呼び出しの例です。

```
rc := IOS_STOP () ;
```

```
IF rc <> 0 THEN (* Abnormal situation to be processed at application level *)
```



## CONFIGURE\_OTB: Advantys OTB のソフトウェア設定の送信

### ファンクションブロックの説明

このファンクションブロックは、Modbus TCP を介して Advantys OTB の SoMachine 設定データを物理デバイスに送ります。

サードパーティのソフトウェアを使用せずにリモート I/O の設定パラメータを更新できます。

このファンクションを呼び出す前に、Modbus Serial IOScanner を停止してください。

このファンクションブロックの実行は非同期です。設定の完了を確認するために、Done、Busy、および Error の出力フラグをアプリケーションサイクルごとにテストしてください。



### IL および ST による表記

IL または ST 言語での一般的な表記については、ファンクションおよびファンクションブロックの表記 (149 ページ) を参照してください。

### I/O 変数の説明

入力変数を次の表に示します。

入力	データ型	コメント
Execute	BOOL	開始入力。立上がりで設定を開始します。
sAddr	STRING	OTB IP アドレス。文字列の形式は、3 { xx . xx . xx . xx } です。

出力変数を次の表に示します。

出力	データ型	コメント
Done	BOOL	設定が正常に完了すると TRUE に設定されます。
Busy	BOOL	設定の実行中に TRUE に設定されます。
Error	BOOL	エラーが検出されて設定が終了すると TRUE に設定されます。
ConfError	configurationOTBErrorCodes (110 ページ参照)	戻り値: configurationOTBErrorCodes
CommError	CommunicationErrorCodes (109 ページ参照)	戻り値: CommunicationErrorCodes

### 例

以下は、このファンクションの呼び出しの例です。

```
VAR
```

```
(*Function Block to configure OTB , need to stop the IOscanner before the execution of the FB*)
```

```
configure_OTB1 : CONFIGURE_OTB ;
```

```
(*init value different than 16#00000000 , IO_start_done=0 when we have a successful start*)
```

```
IO_start_done : UDINT := 1000 ;
```

```
(*init value different than 16#FFFFFFFF , IO_start_done=16#FFFFFFFF when we have a successful stop*)
```

```
IO_stop_done : UDINT := 1000 ;
```

```
(*Configure_OTB_done= true when we configure with success the OTB, then we can start the IO scanner*)
```

```
Configure_OTB_done : BOOL ;
myBusy : BOOL ;
myError : BOOL ;
myConfError : configurationOTBErrorCodes ;
myCommError : UINT ;
myExecute : BOOL ;
END_VAR

(* First, stop the IOScanner, before configuring OTB *)
IF NOT myExecute THEN
IO_stop_done := IOS_STOP () ;
END_IF

(* Send the configuration data to OTB, at IP address 95.15.3.1, when myExecute is TRUE *)
configure_OTB1 (
Execute := myExecute ,
sAddr := ' 3 { 95 . 15 . 3 . 1 }' ,
Done => Configure_OTB_done ,
Busy => myBusy ,
Error => myError ,
ConfError => myConfError ,
CommError => myCommError ) ;

(* After OTB is successfully configured, start the IOScanner *)
IF Configure_OTB_done THEN
IO_start_done := IOS_START () ;
END_IF
```

## A.2 Modbus Serial IOScanner データ型

### 概要

このセクションでは、ModbusTCPIOScanner ライブラリのデータ型について説明します。

### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
IoStateCodes: Modbus Serial IOScanner の状態値	<a href="#">108</a>
CommunicationErrorCodes: 検出エラーコード	<a href="#">109</a>
configurationOTBErrorCodes: OTB 設定の検出エラーコード	<a href="#">110</a>

## iosStateCodes: Modbus Serial IOScanner の状態値

### 列挙型の説明

iosStateCodes 列挙型には次の値が含まれます。

列挙子	値	コメント
iosErr	0	Modbus Serial IOScanner はエラー状態です。
iosIdle	1	Modbus Serial IOScanner は IDLE 状態です。設定が空であるか、準備していません。
iosOperational	2	Modbus Serial IOScanner は OPERATIONAL 状態です。
iosStopped	3	Modbus Serial IOScanner は STOPPED 状態です。

## CommunicationErrorCodes: 検出エラーコード

### 列挙型の説明

CommunicationErrorCodes 列挙型には次の値が含まれます。

列挙子	値	コメント
CommunicationOK	16 進数 00	通信は正常です。
TimedOut	16 進数 01	タイムアウトのため通信が停止しました。
Canceled	16 進数 02	ユーザーの要求により通信が停止しました。
BadAddress	16 進数 03	アドレスの形式が正しくありません。
BadRemoteAddr	16 進数 04	リモートアドレスが正しくありません。
BadMgtTable	16 進数 05	管理テーブルの形式が正しくありません。
BadParameters	16 進数 06	特定のパラメーターが正しくありません。
ProblemSendingRq	16 進数 07	要求を指定先へ送信中にエラーが検出されました。
RecvBufferTooSmall	16 進数 09	受信バッファのサイズが小さ過ぎます。
SendBufferTooSmall	16 進数 0A	送信バッファのサイズが小さ過ぎます。
SystemResourceMissing	16 進数 0B	システムリソースがありません。
BadTransactionNb	16 進数 0C	トランザクション番号が正しくありません。
BadLength	16 進数 0E	長さが正しくありません。
ProtocolSpecificError	16 進数 FE	検出された動作エラーにプロトコル固有のコードが含まれています。
Refused	16 進数 FF	トランザクションが拒否されました。

## configurationOTBErrorCodes: OTB 設定の検出エラーコード

### 列挙型の説明

configurationOTBErrorCodes 列挙型には次の値が含まれます。

列挙子	値	コメント
ConfigurationOK	16 進数 00	OTB 設定が正常に完了しました。
IPAddrErr	16 進数 01	sAddr 入力パラメーターが正しくありません。
ChannelNbErr	16 進数 02	この IP アドレスの OTB チャンネル初期値がありません。
ChannelInitValueErr	16 進数 03	OTB チャンネルの初期値を取得できません。
CommunicationErr	16 進数 04	エラーが検出されたため OTB 設定が停止しました。
IoStateErr	16 進数 05	Modbus Serial IOScanner は実行中です。 CONFIGURE_OTB ファンクションを実行する前に、 Modbus Serial IOScanner を停止してください。

---

# 付録 B

## EtherNet/IP Explicit メッセージ通信ライブラリ

---

### この章について

この章には次のセクションが含まれています。

セクション	項目	参照ページ
B.1	EtherNet/IP Explicit メッセージ通信ファンクション	112
B.2	EIP Explicit メッセージ通信のデータ型	127

## B.1

### EtherNet/IP Explicit メッセージ通信ファンクション

#### このセクションについて

このセクションには次の項目が含まれています。

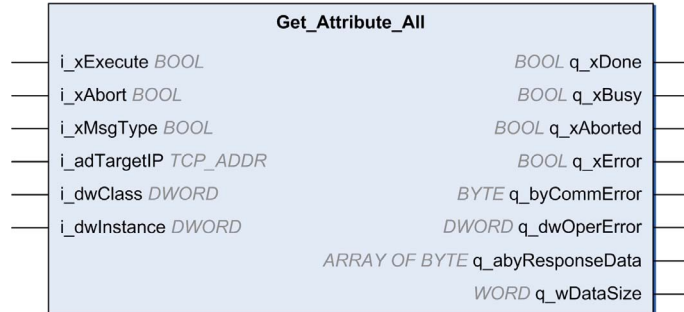
項目	参照ページ
Get_Attribute_All: オブジェクトのすべての属性の取得	113
Set_Attribute_All: インスタンスまたはクラスのすべての属性の設定	115
Get_Attribute_Single: オブジェクトの属性の取得	117
Set_Attribute_Single: オブジェクトの属性の設定	119
EIPStartConnection: 接続の開始	121
EIPStartAllConnection: すべての接続の開始	122
EIPStopConnection: 接続の停止	123
EIPStopAllConnections: すべての接続の停止	124
EIPGetHealthBit: ヘルスビット値の取得	125
デバイスのマニュアルでのオブジェクト情報の探し方	126



## Get\_Attribute\_All: オブジェクトのすべての属性の取得

### ファンクションブロックの説明

このファンクションブロックは、オブジェクトのすべての属性の内容を返します。



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
i_xExecute	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できます。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 <ul style="list-style-type: none"> <li>● FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。</li> <li>● TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。</li> </ul>
i_xAbort	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行は中止されていません。</li> <li>● TRUE: 実行は別のファンクションブロックによって中止されました。</li> </ul>
i_xMsgType	BOOL	<ul style="list-style-type: none"> <li>● FALSE: UCCM</li> <li>● TRUE: 接続された (クラス 3) メッセージ</li> </ul>
i_adTargetIP	TCP_ADDR	対象のデバイスの IP アドレス。
i_dwClass	DWORD	対象のデバイスのクラス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 クラスが要求の一部ではない場合は、0xFFFFFFFF にしてください。
i_dwInstance	DWORD	対象のデバイスのインスタンス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 対象のデバイスがクラスインスタンスの場合は、0 にできません。インスタンスが要求の一部ではない場合は、0xFFFFFFFF にしてください。

## 出力

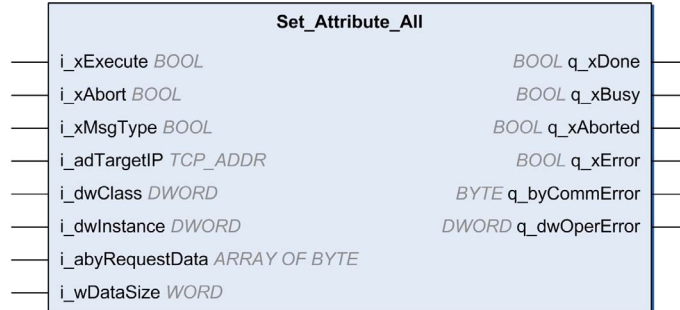
出力変数を次の表に示します。

出力	データ型	コメント
q_xDone	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>● TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
q_xBusy	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>● TRUE: ファンクションブロックが実行中です。</li> </ul>
q_xAborted	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行は中止されていません。</li> <li>● TRUE: 実行は Abort 入力によって中止されました。</li> </ul>
q_xError	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>● TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>
q_byCommError	BYTE	検出されたエラーに関する情報を提供します。
q_dwOperError	DWORD	検出されたエラーに関する情報を提供します。
q_abyResponseData	ARRAY OF BYTE 0...MAX_EIP_REQUEST_DATA_SIZE	成功した場合の応答データ。 <sup>1</sup>
q_wDataSize	WORD	応答データのサイズ (バイト)。
<sup>1</sup> Get_Attribute_All ファンクションは、ODVA 仕様に従ってフォーマットされたバッファを返します。CIP の Get_Attribute_All 応答を参照してください。		

## Set\_Attribute\_All: インスタンスまたはクラスのすべての属性の設定

### ファンクションブロックの説明

このファンクションブロックは、インスタンスまたはクラスのすべての属性を設定します。



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
i_xExecute	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できます。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 <ul style="list-style-type: none"> <li>● FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。</li> <li>● TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。</li> </ul>
i_xAbort	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行は中止されていません。</li> <li>● TRUE: 実行は別のファンクションブロックによって中止されました。</li> </ul>
i_xMsgType	BOOL	<ul style="list-style-type: none"> <li>● FALSE: UCCM</li> <li>● TRUE: 接続された (クラス 3) メッセージ</li> </ul>
i_adTargetIP	TCP_ADDR	対象のデバイスの IP アドレス。
i_dwClass	DWORD	対象のデバイスのクラス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 クラスが要求の一部ではない場合は、0xFFFFFFFF にしてください。
i_dwInstance	DWORD	対象のデバイスのインスタンス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 対象のデバイスがクラスインスタンスの場合は、0 にできます。インスタンスが要求の一部ではない場合は、0xFFFFFFFF にしてください。
<sup>1</sup> 入力データバッファも同様にフォーマットしてください。ODVA EtherNet/IP 仕様ボリューム 1 の Set_Attribute_All 応答データを参照してください。		

入力	データ型	コメント
i_abyRequestData	ARRAY OF BYTE 0...MAX_EIP_REQU EST_DATA_SIZE	データを対象のデバイスに送信する必要があります。 送信しない場合は、wDataSize を 0 にしてください。 <sup>1</sup>
q_wDataSize	WORD	abyRequestData の実際のサイズ。 <sup>1</sup>
<sup>1</sup> 入力データバッファも同様にフォーマットしてください。ODVA EtherNet/IP 仕様ボリューム 1 の Set_Attribute_All 応答データを参照してください。		

## 出力

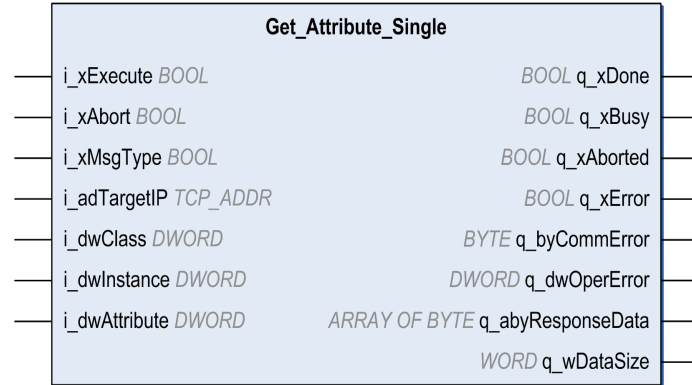
出力変数を次の表に示します。

出力	データ型	コメント
q_xDone	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>● TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
q_xBusy	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>● TRUE: ファンクションブロックが実行中です。</li> </ul>
q_xAborted	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行は中止されていません。</li> <li>● TRUE: 実行は Abort 入力によって中止されました。</li> </ul>
q_xError	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>● TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>
q_byCommError	BYTE	検出されたエラーに関する情報を提供します。
q_dwOperError	DWORD	検出されたエラーに関する情報を提供します。

## Get\_Attribute\_Single: オブジェクトの属性の取得

### ファンクションブロックの説明

このファンクションブロックは、オブジェクトインスタンスの特定の属性の内容を返します。



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
i_xExecute	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できません。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 <ul style="list-style-type: none"> <li>FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。</li> <li>TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。</li> </ul>
i_xAbort	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>FALSE: 実行は中止されていません。</li> <li>TRUE: 実行は別のファンクションブロックによって中止されました。</li> </ul>
i_xMsgType	BOOL	<ul style="list-style-type: none"> <li>FALSE: UCCM</li> <li>TRUE: 接続された (クラス 3) メッセージ</li> </ul>
i_adTargetIP	TCP_ADDR	対象のデバイスの IP アドレス。
i_dwClass	DWORD	対象のデバイスのクラス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 クラスが要求の一部ではない場合は、0xFFFFFFFF にしてください。
i_dwInstance	DWORD	対象のデバイスのインスタンス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 対象のデバイスがクラスインスタンスの場合は、0 にできません。インスタンスが要求の一部ではない場合は、0xFFFFFFFF にしてください。

入力	データ型	コメント
i_dwAttribute	DWORD	対象のデバイスの属性。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 属性が要求の一部ではない場合は、0xFFFFFFFF にして ください。

## 出力

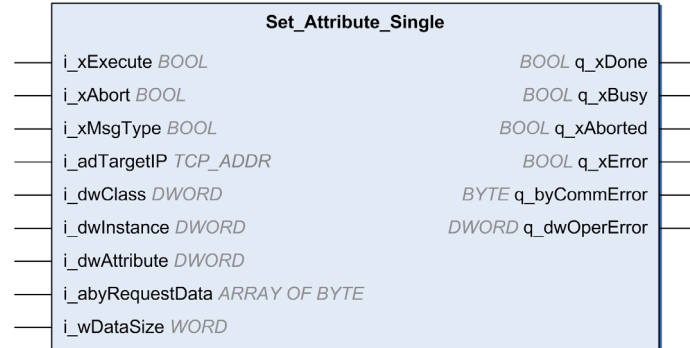
出力変数を次の表に示します。

出力	データ型	コメント
q_xDone	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>● TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
q_xBusy	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>● TRUE: ファンクションブロックが実行中です。</li> </ul>
q_xAborted	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行は中止されていません。</li> <li>● TRUE: 実行は Abort 入力によって中止されました。</li> </ul>
q_xError	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>● TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>
q_byCommError	BYTE	検出されたエラーに関する情報を提供します。
q_dwOperError	DWORD	検出されたエラーに関する情報を提供します。
q_abyResponseData	ARRAY OF BYTE 0...MAX_EIP_REQU EST_DATA_SIZE	成功した場合の応答データ。
q_wDataSize	WORD	応答データのサイズ (バイト)。

## Set\_Attribute\_Single: オブジェクトの属性の設定

### ファンクションブロックの説明

このファンクションブロックは、オブジェクトインスタンスの特定の属性を設定します。



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
i_xExecute	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できます。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 <ul style="list-style-type: none"> <li>● FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。</li> <li>● TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。</li> </ul>
i_xAbort	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行は中止されていません。</li> <li>● TRUE: 実行は別のファンクションブロックによって中止されました。</li> </ul>
i_xMsgType	BOOL	<ul style="list-style-type: none"> <li>● FALSE: UCCM</li> <li>● TRUE: 接続された (クラス 3) メッセージ</li> </ul>
i_adTargetIP	TCP_ADDR	対象のデバイスの IP アドレス。
i_dwClass	DWORD	対象のデバイスのクラス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 クラスが要求の一部ではない場合は、0xFFFFFFFF にしてください。
i_dwInstance	DWORD	対象のデバイスのインスタンス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 対象のデバイスがクラスインスタンスの場合は、0 にできます。インスタンスが要求の一部ではない場合は、0xFFFFFFFF にしてください。
<sup>1</sup> 入力データバッファも同様にフォーマットしてください。ODVA EtherNet/IP 仕様ボリューム 1 の Set_Attribute_Single 応答データを参照してください。		

入力	データ型	コメント
i_dwAttribute	DWORD	対象のデバイスの属性。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 属性が要求の一部ではない場合は、0xFFFFFFFF にしてください。
i_abyRequestData	ARRAY OF BYTE 0...MAX_EIP_REQUEST_DATA_SIZE	データを対象のデバイスに送信する必要があります。送信しない場合は、wDataSize を 0 にしてください。 <sup>1</sup>
q_wDataSize	WORD	abyRequestData の実際のサイズ。 <sup>1</sup>
<sup>1</sup> 入力データバッファも同様にフォーマットしてください。ODVA EtherNet/IP 仕様ボリューム 1 の Set_Attribute_Single 応答データを参照してください。		

## 出力

出力変数を次の表に示します。

出力	データ型	コメント
q_xDone	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>● TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
q_xBusy	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>● TRUE: ファンクションブロックが実行中です。</li> </ul>
q_xAborted	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行は中止されていません。</li> <li>● TRUE: 実行は Abort 入力によって中止されました。</li> </ul>
q_xError	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>● TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>
q_byCommError	BYTE	検出されたエラーに関する情報を提供します。
q_dwOperError	DWORD	検出されたエラーに関する情報を提供します。



## EIPStartConnection: 接続の開始

### ファンクションブロックの説明

このファンクションブロックは、対応する制御ビットにアクセスすることで指定した接続を開始し、接続が開始されると完了を返します。

図



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
<i>i_xExecute</i>	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できます。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 <ul style="list-style-type: none"> <li>● FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。</li> <li>● TRUE: Enable が TRUE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。</li> </ul>
<i>i_uiConnId</i>	UINT	接続 ID。

### 出力

出力変数を次の表に示します。

出力	データ型	コメント
<i>q_xDone</i>	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>● TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
<i>q_xBusy</i>	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>● TRUE: ファンクションブロックが実行中です。</li> </ul>
<i>q_xError</i>	BOOL	値の範囲 : FALSE、TRUE 初期値 : FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>● TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>

## EIPStartAllConnection: すべての接続の開始

### ファンクションブロックの説明

このファンクションブロックは、対応する制御ビットにアクセスすることですべての接続を開始し、接続が開始されると完了を返します。

図



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
i_xExecute	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できます。対象の値は、立上がりが発生した時点で新しい値に書き換えられます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 <ul style="list-style-type: none"> <li>FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。</li> <li>TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。</li> </ul>

### 出力

出力変数を次の表に示します。

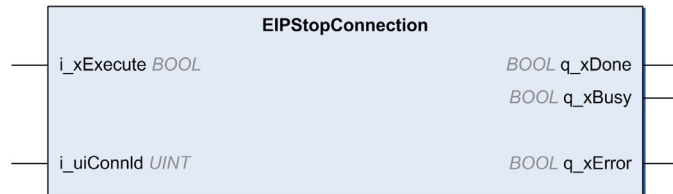
出力	データ型	コメント
q_xDone	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
q_xBusy	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>TRUE: ファンクションブロックが実行中です。</li> </ul>
q_xError	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>

## EIPStopConnection: 接続の停止

### ファンクションブロックの説明

このファンクションブロックは、対応する制御ビットにアクセスすることで指定した接続を停止し、接続が停止すると完了を返します。

**注記：** 接続を停止することができても、システムは接続を再開しようとします。通信を停止するには、関連するリモートアダプターを無効にしてください。  
<DeviceName>. DisableRemoteAdapter (TRUE);



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
i_xExecute	BOOL	値の範囲：FALSE、TRUE 初期値：FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できます。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 <ul style="list-style-type: none"> <li>● FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。</li> <li>● TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。</li> </ul>
i_uiConnId	UINT	接続 ID。

### 出力

出力変数を次の表に示します。

出力	データ型	コメント
q_xDone	BOOL	値の範囲：FALSE、TRUE 初期値：FALSE <ul style="list-style-type: none"> <li>● FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>● TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
q_xBusy	BOOL	値の範囲：FALSE、TRUE 初期値：FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>● TRUE: ファンクションブロックが実行中です。</li> </ul>
q_xError	BOOL	値の範囲：FALSE、TRUE 初期値：FALSE <ul style="list-style-type: none"> <li>● FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>● TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>

## EIPStopAllConnections: すべての接続の停止

### ファンクションブロックの説明

このファンクションブロックは、対応する制御ビットにアクセスすることですべての接続を停止し、接続が停止すると完了を返します。

**注記：** 接続を停止することができても、システムは接続を再開しようとします。通信を停止するには、関連するリモートアダプターを無効にしてください。  
 <DeviceName>. DisableRemoteAdapter (TRUE);



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
i_xExecute	BOOL	値の範囲：FALSE、TRUE 初期値：FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できません。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 ● FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。 ● TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。

### 出力

出力変数を次の表に示します。

出力	データ型	コメント
q_xDone	BOOL	値の範囲：FALSE、TRUE 初期値：FALSE ● FALSE: 実行が開始されていない、またはエラーが検出されました。 ● TRUE: エラーは検出されずに実行が終了しました。
q_xBusy	BOOL	値の範囲：FALSE、TRUE 初期値：FALSE ● FALSE: ファンクションブロックの実行が開始されていない、または終了していません。 ● TRUE: ファンクションブロックが実行中です。
q_xError	BOOL	値の範囲：FALSE、TRUE 初期値：FALSE ● FALSE: ファンクションブロックの実行中です。エラーは検出されていません。 ● TRUE: ファンクションブロックの実行中にエラーが検出されました。

## EIPGetHealthBit: ヘルスビット値の取得

### ファンクションブロックの説明

このファンクションブロックは、指定したヘルスビットを返します。



### 入力

入力変数を次の表に示します。

入力	データ型	コメント
i_xExecute	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できます。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されます。 <ul style="list-style-type: none"> <li>FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。</li> <li>TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。</li> </ul>
i_uiConnId	UINT	接続 ID。

### 出力

出力変数を次の表に示します。

出力	データ型	コメント
q_xDone	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
q_xBusy	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>TRUE: ファンクションブロックが実行中です。</li> </ul>
q_xError	BOOL	値の範囲: FALSE、TRUE 初期値: FALSE <ul style="list-style-type: none"> <li>FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>
q_byCommError	BYTE	検出されたエラーに関する情報を提供します。
q_byOperError	BYTE	検出されたエラーに関する情報を提供します。
q_HealthValue	UINT	ヘルス値を返します。

## デバイスのマニュアルでのオブジェクト情報の探し方

### 概要

デバイスのマニュアルには、アクセスするデータに対応するオブジェクトの説明があります。それらは通常、アプリケーションオブジェクトと呼ばれ、Explicit メッセージ通信でアクセスが可能、またはカテゴリ 3 に属すると記述されています。

オブジェクトは、ソフトウェアプログラミングの辞書と似ています。辞書には、自動的に順序付けされる、または異なる検索メカニズムをもっている、などいくつかの種類があります。例えば、SortedDictionary が 1 つのクラスであり、UnsortedDictionary は別のクラスです。オブジェクトがこれらのクラスの 1 つを使用して構築されている場合、クラスの識別子はそれぞれ SortedDictionary および UnsortedDictionary です。

そのようなオブジェクトを変数名 myDictionary でインスタンス化することは、メモリー内の予約領域に、例えば instance と呼ばれるこの辞書が割り当てられることを意味します。その識別子は myDictionary です。

値は、辞書内で構造体 ( キー、値 ) に格納されます。辞書には、例えば attribute と呼ばれるキーの一覧を取得するためのメソッドがあります。その識別子は GetKeys です。この辞書には、値の一覧を取得するメソッドもあります。このメソッドは、別の attribute で、その識別子は Values です。2 つの属性の識別子は両方のクラスに共通なため、「クラス」の「属性」と呼ばれます。実際には、SortedDictionary には専用の属性があり、その識別子は GetSortedKeys です。このような場合は、「インスタンス属性」と呼ばれます。

属性は一部のサービスにも対応しています。属性 GetKeys はサービス **Get\_Attribute\_Single** (読み込みアクセス) に対応しているのに対して、属性 Values はサービス **Get\_Attribute\_Single** または **Set\_Attribute\_Single** (読み込みまたは書き込みアクセス) に対応しています。対応しているサービスの識別子は、Get\_Attribute\_Single または Set\_Attribute\_Single です。

使用するファンクションブロックに応じた、対応する情報は以下のとおりです。

- `i_byService`: データにアクセスするために使用するサービスの識別子。例えば、「対応しているクラス属性サービス」または「対応しているインスタンス属性サービス」を検索することで探すことができます。
- `i_dwClass`: アクセスするオブジェクトを記述するクラスの識別子。「クラス ID」は数値プロパティで、殆どの場合 16 進数の値で表されます。
- `i_dwInstance`: アクセスするオブジェクトを記述するインスタンスの識別子。「インスタンス ID」は数値プロパティで、殆どの場合 16 進数の値で表されます。
- `i_dwAttribute`: アクセスする属性の識別子で、特定のデータです。これは同じクラスのすべてのインスタンスに共通なクラス属性、またはただのインスタンス属性です。「属性 ID」は数値プロパティで、殆どの場合 16 進数の値で表されます。
- `i_dwMember`: オブジェクトをグループのメンバーとして識別します。ただし、殆ど使用されません。

## B.2

### EIP Explicit メッセージ通信のデータ型

#### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
CommunicationErrorCodes: 通信エラーコード	128
OperationErrorCodes: 動作エラーコード	129

## CommunicationErrorCodes: 通信エラーコード

### 列挙型の説明

CommunicationErrorCodes 列挙型には、中断および検出されたエラーなどの通信診断に関する情報が含まれています。含まれる値は次のとおりです。

列挙子	値 (16 進数)	説明
CommunicationOK	00	通信は有効です。
TimedOut	01	タイムアウトしたときに通信が停止しました。
Canceled	02	ユーザーの要求により通信が停止しました (Abort コマンド)。
BadAddress	03	アドレスの形式が正しくありません。
BadRemoteAddr	04	リモートアドレスが正しくありません。
BadMgtTable	05	管理テーブルの形式が正しくありません。
BadParameters	06	特定のパラメーターが正しくありません。
ProblemSendingRq	07	要求を送信先へ送信中に問題が発生しました。
RecvBufferTooSmall	09	受信バッファサイズが不十分です。
SendBufferTooSmall	0A	送信バッファサイズが不十分です。
SystemResourceMissing	0B	システムリソースがありません。
BadTransactionNb	0C	トランザクション番号が正しくありません。
BadLength	0E	長さが正しくありません。
ProtocolSpecificError	FE	動作エラーコードにプロトコル固有のコードが含まれています。
Refused	FF	メッセージが拒否されました。



## OperationErrorCodes: 動作エラーコード

### 列挙型の説明

OperationErrorCodes 列挙型には、検出されたエラーに対応するコードが含まれています。

#### 00

CommunicationErrorCodes が 16 進数 00 (正しいトランザクション) の場合、OperationErrorCodes 列挙型は以下の値を返します。

列挙子	値 (16 進数)	説明
OperationOK	00	通信は有効です。
NotProcessed_or_TargetResourceMissing	01	要求が処理されていません。
BadResponse	02	受信した応答が正しくありません。

#### FF

CommunicationErrorCodes が 16 進数 FF (メッセージが拒否された) の場合、OperationErrorCodes 列挙型は以下の値を返します。

列挙子	値 (16 進数)	説明
NotProcessed_or_TargetResourceMissing	01	対象のデバイスシステムのリソースに通信できません。
BadLength	05	長さが正しくありません。
CommChannelErr	06	通信チャンネルが、検出されたエラーと関連しています。
BadAddr	07	アドレスが正しくありません。
SystemResourceMissing	0B	システムリソースがありません。
TargetCommInactive	0C	対象のデバイスの通信機能が無効です。
TargetMissing	0D	対象のデバイスに通信できません。
ChannelNotConfigured	0F	チャンネルが設定されていません。

#### FE

CommunicationErrorCodes が 16 進数 FE の場合、OperationErrorCodes 列挙型は以下の値を返します。

状態名	値 (16 進数)	説明
成功	0x00	サービスは指定されたオブジェクトによって正常に実行されました。
接続障害	0x01	接続に関連するサービスは、接続パスに伴い正常に実行しません。
リソースがない	0x02	オブジェクトが要求されたサービスを実行するために必要なリソースがありません。
無効なパラメータ値	0x03	状態コード 0x20 を参照してください。この状態で使用するのに適した値です。
パスセグメントエラー	0x04	パスセグメント識別子またはセグメント構文が、処理ノードで認識されませんでした。パスセグメントエラーが発生すると、パス処理は停止します。
パスの指定先不明	0x05	正しくない、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、または構造体要素をパスが参照しています。このエラーが発生すると、パス処理は停止します。
部分的な送信	0x06	予定されたデータの一部分のみが送信されませんでした。
接続切断	0x07	メッセージ通信接続が切断されました。
対応していないサービス	0x08	要求されたサービスが実装されていないか、このオブジェクトクラス/インスタンスに定義されていません。
無効な属性値	0x09	属性データが無効です。

状態名	値 (16 進数)	説明
属性リストエラー	0x0A	Get_Attribute_List または Set_Attribute_List 応答の属性が 0 以外の状態です。
すでに要求されたモード / 状態	0x0B	オブジェクトはすでにサービスによって要求されているモード / 状態です。
オブジェクト状態の競合	0x0C	オブジェクトは、現在のモード / 状態では要求されたサービスを実行できません。
すでに存在するオブジェクト	0x0D	作成するために要求されたオブジェクトのインスタンスはすでに存在します。
設定できない属性	0x0E	変更できない属性を変更する要求が受信されました。
権限違反	0x0F	許可 / 権限のチェックに失敗しました。
デバイス状態の競合	0x10	デバイスのモード / 状態が要求されたサービスの実行を許可しません。
返信データが大き過ぎ	0x11	応答バッファで送信されるデータが、割り当てられた応答バッファよりも大きいです。
プリミティブ値のフラグメント化	0x12	サービスには、プリミティブデータ値、つまり REAL データ型の半分をフラグメント化する処理が指定されています。
不十分なデータ	0x13	指定された処理を実行するための十分なデータがサービスから提供されませんでした。
対応していない属性	0x14	要求で指定された属性には対応していません。
多過ぎるデータ	0x15	サービスから予定以上のデータが提供されました。
存在しないオブジェクト	0x16	指定されたオブジェクトはデバイスに存在しません。
サービスのフラグメント化シーケンスが実行していない	0x17	このサービスのフラグメント化シーケンスは、このデータに対して有効ではありません。
保存された属性データがない	0x18	このオブジェクトの属性データは、要求されたサービス以前に保存されていません。
保存処理の失敗	0x19	このオブジェクトの属性データは保存されませんでした。
ルーティング障害、要求パケットが大き過ぎ	0x1A	サービスの要求パケットが、ネットワーク上のバスの送信先へ送信するには大き過ぎました。ルーティングデバイスはサービスの中止を強制されました。
ルーティング障害、応答パケットが大き過ぎ	0x1B	サービスの応答パケットが、ネットワーク上のバスの送信先へ送信するには大き過ぎました。ルーティングデバイスはサービスの終了を強制されました。
属性リストの項目データの不足	0x1C	サービスは、要求された動作を実行するためにサービスに必要な属性リストに属性を提供しませんでした。
無効な属性値のリスト	0x1D	サービスは、無効な属性の状態情報が入った属性リストを返しています。
内蔵サービスのエラー	0x1E	内蔵サービスでエラーが発生しました。
メーカー固有のエラー	0x1F	メーカー固有のエラーが検出されました。エラー応答の追加のコードフィールドで、発生した特定のエラーが定義されます。この一般的なエラーコードは、この表のエラーコードまたはオブジェクトクラス定義内に正確にエラーが示されていない場合のみ使用されません。
無効なパラメーター	0x20	要求に関連するパラメーターが無効です。このコードは、パラメーターがこの仕様の要件またはアプリケーションオブジェクトの仕様で定義された要件を満たしていない場合に使用されます。
既にかき込まれている追記型の値またはメディア	0x21	既にかき込まれている追記型メディア (例えば、WORM ドライブ、PROM) にかき込もうとしたか、一度設定されたら変更できない値を変更しようとした。
無効な応答の受信	0x22	無効な応答が受信されました (例えば、応答サービスコードが要求サービスコードと一致していない、または応答メッセージが予定された最小応答サイズより短いなど)。この状態コードは、その他の原因による無効な応答にも使用されます。
バッファのオーバーフロー	0x23	受信したメッセージが、受信バッファが処理できる大きさより大きいです。メッセージ全体が破棄されます。
メッセージフォーマットエラー	0x24	受信したメッセージのフォーマットがサーバーで対応していません。

状態名	値 (16 進数)	説明
パスのキーエラー	0x25	パスの最初のセグメントとして含まれているキーセグメントが送信先のモジュールと一致していません。オブジェクト固有の状態は、キーチェックでどの部分が失敗したかを示します。
無効なパスサイズ	0x26	サービス要求と共に送信されるパスのサイズが、要求をオブジェクトにルーティングするのに十分でないか、含まれているルーティングデータが多過ぎます。
リストの予定外の属性	0x27	現時点では設定できない属性を設定しようとしてしました。
無効なメンバー ID	0x28	要求で指定されたメンバー ID が、指定されたクラス / インスタンス / 属性に存在しません。
設定できないメンバー	0x29	変更できないメンバーを変更する要求が受信されました。
グループ 2 のみの一般的なサーバー障害	0x2A	このエラーコードは、4K 以下のコードスペースしかなく、対応していないサービスや対応していない属性、設定できない属性の代わりであるグループ 2 のみのサーバーによってのみ報告される場合があります。
不明な Modbus エラー	0x2B	CIP から Modbus への変換器で無効な Modbus 例外コードを受信しました。
取得できない属性	0x2C	読み込み不可の属性を読み込む要求が受信されました。
削除できないインスタンス	0x2D	要求されたオブジェクトインスタンスは削除できません。
サービスは指定されたパス 1 には対応していません。	0x2E	オブジェクトはサービスに対応していますが、指定されたアプリケーションパス (例えば、属性など) には対応していません。 <b>注記:</b> 設定サービスには使用しないでください (代わりに一般の状態コード 0x0E または 0x29 を使用してください)。
タイムアウト	0xFF	対象のデバイスからの応答がありません。



---

# 付録 C

## EtherNet/IP Scanner ライブラリ

---

### 概要

この章では、EtherNet/IP Scanner ライブラリについて説明します。

### この章について

この章には次のセクションが含まれています。

セクション	項目	参照ページ
C.1	EtherNet/IP Scanner ファンクション	134
C.2	EtherNet/IP Scanner データ型	140

## C.1 EtherNet/IP Scanner ファンクション

### 概要

このセクションでは、EtherNet/IP Scanner ライブラリに含まれているファンクションについて説明します。

### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
EipControl: EtherNet/IP Scanner の制御	<a href="#">135</a>
EipGetHealth: ヘルスビットの読み込み	<a href="#">136</a>
EipDataExch: Explicit メッセージの送信	<a href="#">137</a>

## EipControl: EtherNet/IP Scanner の制御

### ファンクションの説明

このファンクションは、1 つまたは複数の EtherNet/IP 接続を開始または停止させます。  
アプリケーションは、直接制御ビットを操作しません。EipControl ファンクションを使用してください。  
各 EtherNet/IP Target (スレーブ) デバイスの接続 ID は、**接続 タブ (45 ページ)** にあります。

図



### IL および ST による表記

IL または ST 言語での一般的な表記については、ファンクションおよびファンクションブロックの表記 (149 ページ) を参照してください。

### I/O 変数の説明

入力変数を次の表に示します。

入力	データ型	コメント
i_uiConnId	UINT	監視する接続の接続 ID (45 ページ参照)。
i_uiControl	UINT	<ul style="list-style-type: none"> <li>● 0 = 指定した接続の開始</li> <li>● 1 = 指定した接続の停止</li> <li>● 2 = すべての接続の開始</li> <li>● 3 = すべての接続の停止</li> </ul>

出力変数を次の表に示します。

出力	データ型	コメント
EipControl	UDINT	<ul style="list-style-type: none"> <li>● 0 = 正常に開始または停止</li> <li>● -1 = 接続 ID が正しくありません</li> </ul>

例

以下は、このファンクションの呼び出しの例です。

```
rc := EipControl (0, 257) ;(* opens the connection No 116 *)
```

```
IF rc <> 0 THEN (* Abnormal situation to be processed at application level *)
```

## EipGetHealth: ヘルスビットの読み込み

### ファンクションの説明

このファンクションは、特定の EtherNet/IP 接続のヘルスビット値を返します。  
 各 EtherNet/IP Target (スレーブ) デバイスの接続 ID は、**接続 タブ** (45 ページ) にあります。

図



### IL および ST による表記

IL または ST 言語での一般的な表記については、ファンクションおよびファンクションブロックの表記 (149 ページ) を参照してください。

### I/O 変数の説明

入力変数を次の表に示します。

入力	データ型	コメント
i_uiconnId	UINT	監視する接続の接続 ID (45 ページ参照)。

出力変数を次の表に示します。

出力	データ型	コメント
EipGetHealth	UINT	<ul style="list-style-type: none"> <li>● 0: 接続が確立されていません</li> <li>● 1: 接続が確立されました</li> </ul>

例

以下は、このファンクションの呼び出しの例です。

```
connID := 257 ;
```

```
channelHealth := EipGetHealth(connID) (* Get the health value (1=OK, 0=Not OK) of the connection number connID. The connection ID is displayed in the configuration editor of the device *)
```



## EipDataExch: Explicit メッセージの送信

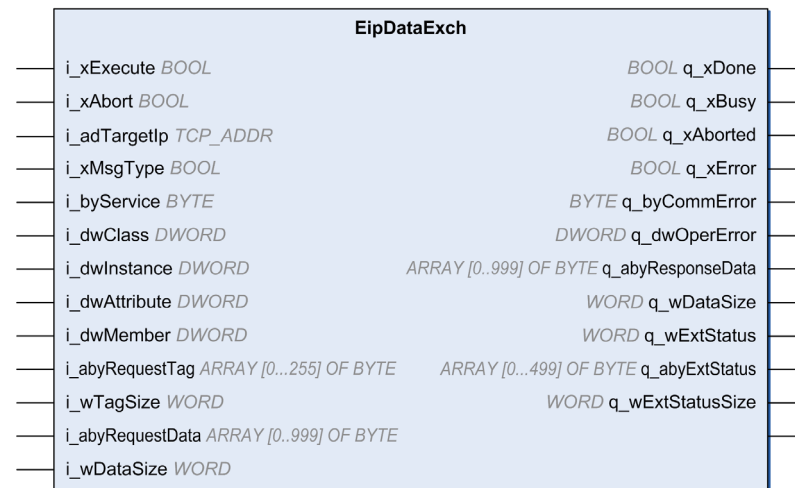
### ファンクションブロックの説明

このファンクションブロックは、Explicit メッセージを送信します。

処理を実行する時間は、産業用 Ethernet マネージャー (28 ページ) で設定できます。

接続されたメッセージのタイムアウト値と接続されていないメッセージのタイムアウト値があります。

この汎用ファンクションブロックは、EtherNet/IP Explicit メッセージ通信ライブラリに実装されていない機能に使用できます。



### IL および ST による表記

IL または ST 言語での一般的な表記については、ファンクションおよびファンクションブロックの表記 (149 ページ) を参照してください。

### I/O 変数の説明

入力変数を次の表に示します。

入力	データ型	継承元	コメント
i_xExecute	BOOL	BASE	初期値: FALSE 入力 Execute の立上がりでファンクションブロックが開始します。ファンクションブロックは実行を継続し、出力 Busy は TRUE に設定されます。動作をトリガーするファンクションブロックが実行されている間に、それらを再度開始できます。対象の値は、立上がりが発生した時点で新しい値に上書きされます。ファンクションブロックの実行中は、入力 Execute での立上がりは無視されません。 ● FALSE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は 1 サイクルの間だけ TRUE になります。 ● TRUE: Enable が FALSE に設定されている場合、出力 Done、Error、または CommandAborted は TRUE のまま維持されます。
i_xAbort	BOOL	BASE	初期値: FALSE ● FALSE: 実行は中止されていません。 ● TRUE: 実行は別のファンクションブロックによって中止されました。
i_xMsgType	BOOL	-	● FALSE: UCCM ● TRUE: 接続された (クラス 3) メッセージ

入力	データ型	継承元	コメント
i_adTargetIP	TCP_ADDRES (143 ページ参照)	-	対象のデバイスの IP アドレス。
i_byService	BYTE	-	実行するサービス (上記のサービスコードを参照)
i_dwClass	DWORD	-	対象のデバイスのクラス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 クラスを要求の一部にはいけない場合は、0xFFFFFFFF にしてください。
i_dwInstance	DWORD	-	対象のデバイスのインスタンス。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 対象のデバイスがクラスインスタンスの場合は、0 にできます。インスタンスを要求の一部にはいけない場合は、0xFFFFFFFF にしてください。
i_dwAttribute	DWORD	-	対象のデバイスの属性。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 属性を要求の一部にはいけない場合は、0xFFFFFFFF にしてください。
i_dwMember	DWORD	-	対象のデバイスのメンバー。 デバイスのマニュアルでのオブジェクト情報の探し方 (126 ページ) を参照してください。 メンバーを要求の一部にはいけない場合は、0xFFFFFFFF にしてください。
i_abyRequestTag	ARRAY OF [0...250] BYTE	-	対象のデバイスの拡張シンボルセグメント。使用しない場合は、i_wTagSize を 0 にしてください。
i_wTagSize	WORD	-	i_abyRequestTag の実際のサイズ。
i_abyRequestData	ARRAY OF [0...999] BYTE	-	対象のデバイスに送信するデータ。使用しない場合は、i_wDataSize を 0 にしてください。
i_wDataSize	WORD	-	i_abyRequestData の実際のサイズ。

出力変数を次の表に示します。

出力	データ型	継承元	コメント
q_xDone	BOOL	BASE	初期値 : FALSE <ul style="list-style-type: none"> <li>FALSE: 実行が開始されていない、またはエラーが検出されました。</li> <li>TRUE: エラーは検出されずに実行が終了しました。</li> </ul>
q_xBusy	BOOL	BASE	初期値 : FALSE <ul style="list-style-type: none"> <li>FALSE: ファンクションブロックの実行が開始されていない、または終了していません。</li> <li>TRUE: ファンクションブロックが実行中です。</li> </ul>
q_xAborted	BOOL	BASE	初期値 : FALSE <ul style="list-style-type: none"> <li>FALSE: 実行は中止されていません。</li> <li>TRUE: 実行は Abort 入力によって中止されました。</li> </ul>
q_xError	BOOL	BASE	初期値 : FALSE <ul style="list-style-type: none"> <li>FALSE: ファンクションブロックの実行中です。エラーは検出されていません。</li> <li>TRUE: ファンクションブロックの実行中にエラーが検出されました。</li> </ul>
q_byCommError	CommunicationError Codes (141 ページ参照)	BASE	通信エラーコード
q_dwOperError	OperationErrorCodes (142 ページ参照)	BASE	動作エラーコード

出力	データ型	継承元	コメント
q_abyResponseData	ARRAY OF [0...999] BYTE	-	成功した場合の応答データ
q_wDataSize	WORD	-	応答データのサイズ (バイト)
q_abyExtStatus	ARRAY OF [0...499] BYTE	-	エラー応答の場合の拡張状態データ
q_wExtStatusSize	WORD	-	拡張状態データのサイズ (16 ビットのワード)
q_wExtStatus	WORD	-	拡張状態ワード

## 例

以下は、この関数の呼び出しの例です。

```
MyEipDataExch (
    i_xExecute := Execute ,
    i_xAbort := Abort ,
    q_xDone => Done ,
    q_xBusy => Busy ,
    q_xAborted => Aborted ,
    q_xError => Err ,
    q_byCommError => CommError ,
    q_dwOperError => OperError ,
    i_adTargetIp := IpAddr ,
    i_xMsgType := MsgType ,
    i_byService := Service ,
    i_dwClass := Class ,
    i_dwInstance := Instance ,
    i_dwAttribute := Attribute ,
    i_dwMember := Member ,
    i_abyRequestTag := RequestTag ,
    i_wTagSize := TagSize ,
    i_abyRequestData := RequestData ,
    i_wDataSize := ReqDataSize ,
    q_abyResponseData => ResponseData ,
    q_wDataSize => ResDataSize ,
    q_abyExtStatus => ExtStatusArray ,
    q_wExtStatusSize => ExtStatusSize ,
    q_wExtStatus => ExtStatus );
```

## C.2

### EtherNet/IP Scanner データ型

#### 概要

このセクションでは、EtherNet/IP Scanner ライブラリのデータ型について説明します。

#### このセクションについて

このセクションには次の項目が含まれています。

項目	参照ページ
CommunicationErrorCodes: 通信エラーコード	<a href="#">141</a>
OperationErrorCodes: 動作エラーコード	<a href="#">142</a>
TCP_ADDR: TCP デバイスのアドレス	<a href="#">143</a>

## CommunicationErrorCodes: 通信エラーコード

### 列挙型の説明

CommunicationErrorCodes 列挙型には、中断および検出されたエラーなどの通信診断に関する情報が含まれています。含まれる値は次のとおりです。

列挙子	値 (16 進数)	詳細
CommunicationOK	00	通信は有効です。
TimedOut	01	タイムアウトしたときに通信が停止しました。
Canceled	02	ユーザーの要求により通信が停止しました (Abort コマンド)。
BadAddress	03	アドレスの形式が正しくありません。
BadRemoteAddr	04	リモートアドレスが正しくありません。
BadMgtTable	05	管理テーブルの形式が正しくありません。
BadParameters	06	特定のパラメーターが正しくありません。
ProblemSendingRq	07	要求を送信先へ送信中に問題が発生しました。
RecvBufferTooSmall	09	受信バッファサイズが不十分です。
SendBufferTooSmall	0A	送信バッファサイズが不十分です。
SystemResourceMissing	0B	システムリソースがありません。
BadTransactionNb	0C	トランザクション番号が正しくありません。
BadLength	0E	長さが正しくありません。
ProtocolSpecificError	FE	動作エラーコードにプロトコル固有のコードが含まれています。
Refused	FF	メッセージが拒否されました。

## OperationErrorCodes: 動作エラーコード

### 列挙型の説明

OperationErrorCodes 列挙型には、検出されたエラーに対応するコードが含まれています。

#### 00

CommunicationErrorCodes が 16 進数 00 (正しいトランザクション) の場合、OperationErrorCodes 列挙型は以下の値を返します。

列挙子	値 (16 進数)	詳細
OperationOK	00	通信は有効です。
NotProcessed_or_TargetResourceMissing	01	要求が処理されていません。
BadResponse	02	受信した応答が正しくありません。

#### FF

CommunicationErrorCodes が 16 進数 FF (メッセージが拒否された) の場合、OperationErrorCodes 列挙型は以下の値を返します。

列挙子	値 (16 進数)	詳細
NotProcessed_or_TargetResourceMissing	01	対象のデバイスシステムのリソースに通信できません。
BadLength	05	長さが正しくありません。
CommChannelErr	06	通信チャンネルが、検出されたエラーと関連しています。
BadAddr	07	アドレスが正しくありません。
SystemResourceMissing	0B	システムリソースがありません。
TargetCommInactive	0C	対象のデバイスの通信機能が無効です。
TargetMissing	0D	対象のデバイスに通信できません。
ChannelNotConfigured	0F	チャンネルが設定されていません。

#### FE

通信エラーコードが 16 進数 FE の場合、OperationErrorCodes 列挙型には、プロトコル固有のエラー検出コードが含まれています。(特定のプロトコルのエラー検出コードを参照してください。)

## TCP\_ADDR: TCP デバイスのアドレス

### 構造体の説明

TCP\_ADDR 構造体データ型には TCP デバイスのアドレスが含まれます。含まれる変数は次のとおりです。

変数	データ型	説明
A	BYTE	IP アドレス A.B.C.D の 1 番目の値
B	BYTE	IP アドレス A.B.C.D の 2 番目の値
C	BYTE	IP アドレス A.B.C.D の 3 番目の値
D	BYTE	IP アドレス A.B.C.D の 4 番目の値
port	WORD	TCP ポート番号 (Modbus のデフォルトは 502)





---

# 付録 D

## モーション制御ライブラリ

---

### モーション制御ライブラリ

#### 概要

本書では、SoMachine ソフトウェア環境下でフィールドバスの ATV32、ATV320、ATV340 ドライブ、ATV6\*\*、ATV71、ATV9\*\*、LXM32M、ILA、ILE、ILS ドライブを制御するために使用するファンクションブロックについて説明しています。

詳細については、Motion Control Library Guide を参照してください。



---

# 付録 E

## 汎用 TCP/UDP ライブラリ

---

### 汎用 TCP/UDP ライブラリ

#### 概要

TcpUdpCommunication ライブラリは IPv4 を使用して TCP および UDP を実装します。

ライブラリは、TCP (クライアントおよびサーバー) または UDP (プラットフォームで対応している場合はブロードキャストおよびマルチキャストを含む) を使用して、ソケットによるネットワーク通信プロトコルを実装するための主要機能を提供します。IPv4 による通信にのみ対応しています。

リモート側 (バーコードスキャナー、視覚カメラ、産業用ロボット、データベースサーバーのようなソフトウェアが実行されているコンピュータシステムなどのハードウェア) で使用されるアプリケーションプロトコルは、このライブラリを使用して実装してください。これには、ソケットによる通信と使用するプロトコルの広い知識が必要ですが、TcpUdpCommunication ライブラリを使用することでアプリケーション層に集中できます。

詳細については、TcpUdpCommunication Library Guide を参照してください。



---

# 付録 F

## ファンクションおよびファンクションブロックの表現

---

### 概要

各ファンクションは次の言語で表現されます。

- IL: 命令リスト
- ST: 構造化テキスト
- LD: ラダー図
- FBD: ファンクション ブロック ダイアグラム
- CFC: コンティニューアスファンクションチャート

この章では、ファンクションおよびファンクションブロックの表現例および IL 言語と ST 言語での使用方法を説明します。

### この章について

この章には次の項目が含まれています。

項目	参照ページ
ファンクションとファンクションブロックの相違	150
IL 言語でのファンクションおよびファンクションブロックの使用方法	151
ST 言語でのファンクションおよびファンクションブロックの使用方法	154

## ファンクションとファンクションブロックの相違

### ファンクション

ファンクションは、

- 即時に 1 つの結果を返す POU (Program Organization Unit プログラム構成単位) です。
- インスタンス経由ではなく、ファンクションの名前で直接呼び出されます。
- 1 つの呼び出しから別の呼び出しへの状態を保持しません。
- 他の式のオペランドとして使用できます。

例：ブール演算子 (AND)、計算式、変換 (BYTE\_TO\_INT)

### ファンクションブロック

ファンクションブロックは、

- 1 つ以上の出力を返す POU (Program Organization Unit プログラム構成単位) です。
- インスタンスによる呼び出しをしてください (ファンクションブロックの専用の名前と変数を含むコピー)。
- 各インスタンスはファンクションブロックまたはプログラムからの、1 つの呼び出しから別の呼び出しへの状態 (出力および内部変数) を保持をします。

例：タイマー、カウンター

次の例では、Timer\_ON はファンクションブロック TON のインスタンスです。

```

1  PROGRAM MyProgram_ST
2  VAR
3      Timer_ON: TON; // Function Block Instance
4      Timer_RunCd: BOOL;
5      Timer_PresetValue: TIME := T#5S;
6      Timer_Output: BOOL;
7      Timer_ElapsedTime: TIME;
8  END_VAR

```

---

```

1  Timer_ON(
2      IN:=Timer_RunCd,
3      PT:=Timer_PresetValue,
4      Q=>Timer_Output,
5      ET=>Timer_ElapsedTime);

```

## IL 言語でのファンクションおよびファンクションブロックの使用方法

### 一般情報

ここでは、IL 言語でのファンクションおよびファンクションブロックの実装方法の説明をします。  
ファンクション IsFirstMastCycle、SetRTCDrift、およびファンクションブロック TON を使用して、実装例を示します。

### IL 言語におけるファンクションの使用

次の手順は、IL 言語にファンクションを挿入する方法を示します。

ステップ	手順内容
1	インストラクションリスト言語で POU を新規作成または開きます。 <b>注記</b> ：ここでは POU の作成手順を省略しています。詳細については、POU の追加および呼び出し (SoMachine, プログラミングガイド参照) を参照してください。
2	ファンクションに必要な変数を作成します。
3	ファンクションに 1 つ以上の入力がある場合、LD 命令を使用して 1 番目の入力から読み込みを始めます。
4	下に新しいラインを挿入し、次のいずれかの操作を行います。 <ul style="list-style-type: none"> <li>演算子の欄 (フィールド左側) にファンクション名を入力します。</li> <li>入力アシスタントを使用して、ファンクションを選択します (コンテキストメニューからボックスの挿入を選択します)。</li> </ul>
5	ファンクションに 1 つ以上の入力があり、入力アシスタントが使用中の場合は、必要なライン数が右側のフィールドに ??? を付けて自動生成されます。??? を入力の順序に対応する適切な値または変数に置き換えます。
6	適切な変数にファンクションの結果を格納するための新しいラインを挿入します。演算子の欄 (フィールド左側) に ST 命令を入力し、フィールド右側に変数名を入力します。

手順を説明するために、ファンクション IsFirstMastCycle (入力パラメーターなし) および SetRTCDrift (入力パラメーター付き) の図を次に示します。

ファンクション	図
入力パラメーターなし: IsFirstMastCycle	
入力パラメーター付き: SetRTCDrift	

IL 言語では、ファンクション名が演算子の欄に直接使用されます。

ファンクション	POU IL エディターでの表現
入力パラメーターなし ファンクションの IL 例: IsFirstMastCycle	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3      FirstCycle: BOOL; 4  END_VAR                     </pre> <hr/> <pre> 1  IsFirstMastCycle    ST          FirstCycle                     </pre>
入力パラメーター付き ファンクションの IL 例: SetRTCDrift	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3      myDrift: SINT (-29..29) := 5; 4      myDay: DAY_OF_WEEK := SUNDAY; 5      myHour: HOUR := 12; 6      myMinute: MINUTE; 7      myDiag: RTCSETDRIFT_ERROR; 8  END_VAR 9                     </pre> <hr/> <pre> 1  LD          myDrift    SetRTCDrift myDay            myHour            myMinute    ST          myDiag                     </pre>

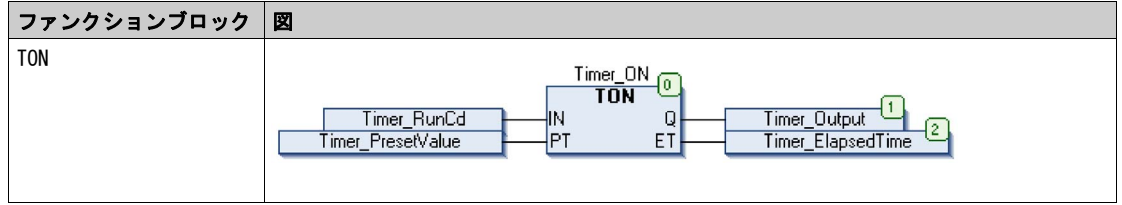
### IL 言語でのファンクションブロックの使用

次の手順は、IL 言語にファンクションブロックを挿入する方法を示します。

ステップ	手順内容
1	インストラクションリスト言語で POU を新規作成または開きます。 <b>注記：</b> ここでは POU の作成手順を省略しています。詳細については、POU の追加および呼び出し ( <i>SoMachine</i> , <i>プログラミングガイド</i> 参照) を参照してください。
2	ファンクションブロックに必要な変数を作成します。
3	ファンクションブロックは次のいずれかの CAL 命令を使用して呼び出されます。 <ul style="list-style-type: none"> <li>● <b>入力アシスタント</b>を使用して、FB を選択します (右クリックをし、コンテキストメニューから<b>ボックスの挿入</b>を選択します)。</li> <li>● 自動的に CAL 命令と必要な I/O が作成されます。</li> </ul> 各パラメーター (I/O) の命令は次のいずれかです。 <ul style="list-style-type: none"> <li>● 入力の値は ":" で設定されます。</li> <li>● 出力の値は "=&gt;" で設定されます。</li> </ul>
4	CAL 右側フィールドの ??? をインスタンス名に置き換えます。
5	他の ??? を適切な変数または即値に置き換えます。



手順を説明するために、TON ファンクションブロックを使用した例の図を次に示します。



IL 言語では、ファンクションブロック名が演算子の欄に直接使用されます。

ファンクションブロック	POU IL エディターでの表現
TON	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3  Timer_ON: TON; // Function Block instance declaration 4  Timer_RunCd: BOOL; 5  Timer_PresetValue: TIME := T#5S; 6  Timer_Output: BOOL; 7  Timer_ElapsedTime: TIME; 8  END_VAR 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 </pre>

## ST 言語でのファンクションおよびファンクションブロックの使用方法

### 一般情報

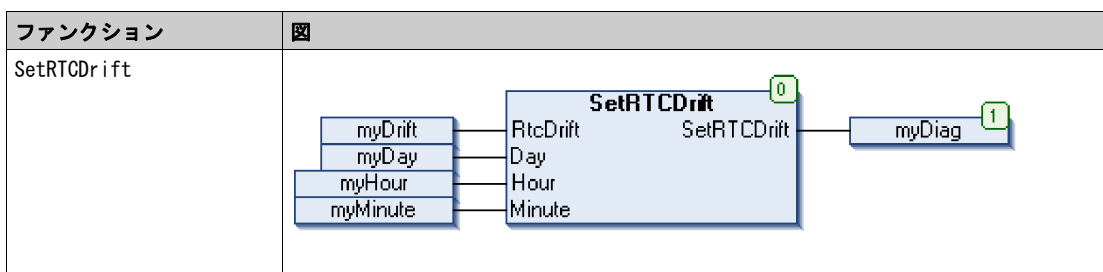
ST 言語でのファンクションおよびファンクションブロックの実装方法の説明をします。  
 ファンクション SetRTCDrift およびファンクションブロック TON を使用して、実装例を示します。

### ST 言語でのファンクションの使用

次の手順は、ST 言語にファンクションを挿入する方法を示します。

ステップ	手順内容
1	ストラクチャードテキスト言語で POU を新規作成または開きます。 <b>注記：</b> ここでは POU の作成手順を省略しています。詳細については、POU の追加および呼び出し ( <i>SoMachine</i> , <i>プログラミングガイド参照</i> ) を参照してください。
2	ファンクションに必要な変数を作成します。
3	ST 言語のファンクションには、 <b>POU ST エディター</b> で一般構文を使用します。一般構文を次に示します。 FunctionResult := FunctionName (Var Input1, Var Input2, ... Var Inputx);

手順を説明するために、ファンクション SetRTCDrift の図を次に示します。



このファンクションの ST 言語は次の通りです。

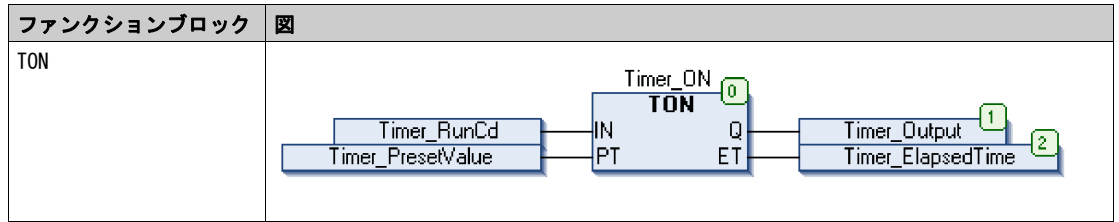
ファンクション	POU ST エディターでの表現
SetRTCDrift	<pre>PROGRAM MyProgram_ST VAR myDrift:SINT(-29..29) := 5; myDay:DAY_OF_WEEK := SUNDAY; myHour:HOUR := 12; myMinute:MINUTE; myRTCAjust:RTCDRIFT_ERROR; END_VAR myRTCAjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);</pre>

### ST 言語でのファンクションブロックの使用

次の手順は、ST 言語にファンクションブロックを挿入する方法を示します。

ステップ	手順内容
1	ストラクチャードテキスト言語で POU を新規作成または開きます。 <b>注記：</b> ここでは POU の作成手順を省略しています。POU の追加、宣言、および呼び出しの詳細については、関連ドキュメント ( <i>SoMachine</i> , <i>プログラミングガイド参照</i> ) を参照してください。
2	入力および出力変数とファンクションブロックに必要なインスタンスを作成します。 <ul style="list-style-type: none"> <li>● 入力変数は、ファンクションブロックに必要な入力パラメーターです。</li> <li>● 出力変数は、ファンクションブロックから返された値を受け取ります。</li> </ul>
3	ST 言語のファンクションブロックには、 <b>POU ST エディター</b> で一般構文を使用します。一般構文を次に示します。 FunctionBlock_ InstanceName ( Input1 := Var Input1 , Input2 := Var Input2, ... Ouput1 => VarOutput1 , Ouput2 => VarOutput2, ... );

手順を説明するために、TON ファンクションブロックを使用した例の図を次に示します。



この表は ST 言語でファンクションブロックの呼び出しの例を示します。

ファンクションブロック	POU ST エディター凡例
TON	<pre> 1  PROGRAM MyProgram_ST 2  VAR 3      Timer_ON: TON; // Function Block Instance 4      Timer_RunCd: BOOL; 5      Timer_PresetValue: TIME := T#5S; 6      Timer_Output: BOOL; 7      Timer_ElapsedTime: TIME; 8  END_VAR  1  Timer_ON( 2      IN:=Timer_RunCd, 3      PT:=Timer_PresetValue, 4      Q=&gt;Timer_Output, 5      ET=&gt;Timer_ElapsedTime);                 </pre>





## %IW

IEC 規格により、%IW は入力ワードレジスター (例えば、アナログ IN タイプの言語オブジェクト) を表します。

## %QW

IEC 規格により、%QW は出力ワードレジスター (例えば、アナログ OUT タイプの言語オブジェクト) を表します。

## アプリケーション

設定データ、シンボル、ドキュメントを含むプログラム。

## コントローラー

産業プロセスを自動化する (プログラマブルロジックコントローラーまたはプログラマブルコントローラーとして知られる)。

## デバイスネットワーク

ロジックコントローラーの特定の通信ポートに接続されたデバイスを含むネットワーク。このコントローラーは、デバイス側から見てマスターとみなされます。

## バイト

8 ビット形式でエンコードされたタイプ。範囲は 16 進数 00 から FF です。

## ファンクションブロックダイアグラム

制御システム用標準規格 IEC 61131-3 で対応しているロジックまたは制御用の 5 言語の 1 つです。ファンクションブロックダイアグラムは、グラフィカルなプログラミング言語です。論理式または算術式、ファンクションブロックの呼び出し、ジャンプ、またはリターン命令のいずれかを表すボックスおよび接続ラインのグラフィックで構造をつなげて組み合わせることで動作します。

## プログラム

アプリケーションのコンポーネント。ロジックコントローラーのメモリーにインストールできるコンパイルされたソースコードです。

## ヘルスタイムアウト

Modbus IO スキャナーの要求とスレーブの応答の間の最大時間 (ms) を表します。

## ヘルスビット

チャンネルの通信状態を示す変数。

## ポスト設定

(*ポスト設定*) アプリケーションを変更することなく、アプリケーションのパラメーターを変更できるオプション。ポスト設定パラメーターは、コントローラーに保存されているファイルで定義されます。それらは、アプリケーションの設定パラメーターを多重定義しています。

## 制御ネットワーク

ロジックコントローラー、SCADA システム、PC、HMI、スイッチなどを含むネットワーク。

2 種類のトロポジーに対応しています。

- フラット: このネットワーク内のすべてのモジュールとデバイスは同じサブネットに属します。
- 2 レベル: このネットワークはオペレーションネットワークとコントローラー間ネットワークに分割されています。

これら 2 つのネットワークは物理的に独立することができますが、通常はルーティングデバイスによってリンクされています。

## 変数

プログラムによって、変更およびアドレス指定されるメモリーユニット。

## 拡張バス

拡張 I/O モジュールとコントローラー間の電子通信バス。

## 繰り返し率

送信される Modbus 要求のポーリング間隔。

## 設定

システム内のハードウェアコンポーネントの配置と接続、およびシステムの動作特性を決めるハードウェアおよびソフトウェアパラメーターの設定。

**ATV**

Altivar ドライブのモデル接頭辞 (例えば、ATV312 は Altivar 312 インバーターを意味します)。

**CFC**

(コンティニューアスファンクションチャート) フローチャートのように機能するファンクション・ブロック・ダイアグラム言語 (FBD 言語) に基づくグラフィカルプログラミング言語 (標準規格 IEC 61131-3 の拡張版)。ネットワークは使用せず、グラフィック要素の自由な位置決めが可能なためフィードバックループが利用できます。各ブロックの入力は左側にあり、出力は右側にあります。ブロック出力を他のブロックの入力にリンクして、複雑な式を作成することができます。

**CIP**

(*common industrial protocol*, 共通産業プロトコル) CIP がネットワークのアプリケーション層に実装されると、プロトコルに関係なく他の CIP ベースのネットワークと途切れなく通信できます。例えば、Ethernet TCP/IP ネットワークのアプリケーション層に CIP を実装すると、EtherNet/IP 環境が作成されます。同様に、CAN ネットワークのアプリケーション層の CIP によって、DeviceNet 環境が作成されます。この場合、EtherNet/IP ネットワークのデバイスは、CIP ブリッジまたはルーターを介した DeviceNet ネットワークのデバイスと通信できます。

**DHCP**

(*dynamic host configuration protocol*) BOOTP を拡張したもの。DHCP はより高度ですが、DHCP および BOOTP は両者とも一般に普及しています。(DHCP は、BOOTP クライアントの要求を処理できます。)

**DTM**

(*device type manager*) 2 つのカテゴリに分類されます。

- デバイス DTM は、フィールドデバイス設定コンポーネントに接続します。
- CommDTM は、ソフトウェア通信コンポーネントに接続します。

DTM は、デバイスパラメーターにアクセスし、デバイスの設定、操作、および診断するための統一された構造体です。DTM は、デバイスパラメーターを設定するための単純なグラフィックユーザーインターフェイスから、診断および保守用のリアルタイムで複雑な計算ができる高度なアプリケーションにまで及びます。

**EDS**

(*electronic data sheet*, 電子データシート) フィールドバスデバイスの説明ファイル。パラメーターおよび設定などのデバイスのプロパティの記載があります。

**FB**

(*Function Block*, ファンクションブロック) 特定の正規化されたアクションの実行ため、プログラミング命令のグループを統合する便利なプログラミングメカニズムです。速度制御、インターバル制御、カウントなどがあります。ファンクションブロックは設定データ、内部または外部操作パラメーターのセット、および 1 つ以上の入出力データが含まれます。

**FDR**

(*fast device replacement*, 高速デバイス交換): デバイスに対応されいるサービス。動作しない機器の交換を容易にします。

**I/O**

(入力/出力)

**IL**

(*instruction list*, インストラクションリスト) コントローラーにより順に実行される一連のテキストベースの命令で書かれたプログラム。各命令は、ライン番号、命令コードおよびオペランドを含む (IEC 61131-3 を参照してください)。

**Input Assembly**

アセンブリは、ネットワークデバイスとロジックコントローラーの間で交換されるデータのブロック。入力アセンブリは、スレーブまたはターゲットデバイスから、マスターまたはオリジネーターが読み取ったステータス情報を含む。

**INT**

(*Integer*, 整数) 16 ビットでエンコードされた整数。

**LD**

(ラダーダイアグラム) コントローラープログラムの命令を表す図。コントローラーで順次実行される一連のラングにある接点、コイル、およびブロックのシンボルを含む。(IEC 61131-3 参照)

**MAC アドレス**

(メディアアクセス制御アドレス) 特定のハードウェアに関連付けられた固有の 48 ビット番号。MAC アドレスは、製造時に各ネットワークカードまたはデバイスにプログラムされます。

**ODVA**

(open DeviceNet vendors association) CIP (EtherNet/IP、DeviceNet および CompoNet) に構築されるネットワーク技術の一つです。

**Originator (マスター)**

EtherNet/IP では、Implicit または Explicit メッセージ通信の CIP 接続を開始するデバイス、または未接続の Explicit メッセージ通信のメッセージ要求を開始するデバイス。

Target (スレーブ) も参照。

**Output Assembly**

アセンブリは、ネットワークデバイスとロジックコントローラーの間で交換されるデータのブロック。一般的に出力アセンブリは、マスターまたはオリジネーターからスレーブまたはターゲットデバイスに送られるコマンドを含みます。

**POU**

(program organization unit、プログラムオーガニゼーションユニット) ソースコード内の変数宣言、および対応する命令セット。POUs はソフトウェアプログラム、ファンクション、およびファンクションブロックのモジュラー化した再利用を容易にします。一度宣言すると POUs がもう一方でも利用可能となります。

**RJ45**

Ethernet 用ネットワークケーブルの 8 ピンコネクタ (標準タイプ)。

**RPI**

(requested packet interval、要求パケット間隔) スキャナーによって要求されたサイクリックデータ交換の時間間隔。EtherNet/IP デバイスは、スキャナーによってデバイスに割り当てられた RPI で指定された速度でデータを発行し、RPI と同じ間隔でスキャナーからのメッセージ要求を受け取ります。

**ST**

(構造化テキスト) 複雑なステートメントとネストされた命令 (反復ループ、条件付き実行、関数など) を含む言語。ST は IEC 61131-3 に準拠しています。

**Target (スレーブ)**

EtherNet/IP では、デバイスが Implicit または Explicit メッセージ通信の接続要求の受信先である場合、デバイスは Target (スレーブ) であると見なされます。

Originator (マスター) も参照。

**TVDA**

(tested validated documented architectures、動作検証済み、評価済み、文書化済みのアーキテクチャー) Schneider Electric のコンポーネントをベースにした制御システムの提案。TVDA は幅広い種類の機器に対応し、機器のパフォーマンス要件、設置条件およびデバイスの費用も考慮します。実装への労力を最小にするために、各 TVDA には、システムのコンポーネントを制御するコントローラーおよび HMI アプリケーションだけでなく詳細なコンポーネント一覧、配線図、試運転ガイドも付属しています。

**UL**

(underwriters laboratories、米国保険業者安全試験所) 製品の試験および安全認証を行う米国の組織。







- Advantys OTB
  - CONFIGURE\_OTB, 105
- Advantys OTB の設定
  - CONFIGURE\_OTB, 105
- CommunicationErrorCodes
  - データ型, 109, 128, 141
- configurationOTBErrorCodes
  - データ型, 110
- CONFIGURE\_OTB
  - Advantys OTB のソフトウェア設定の送信, 105
- DHCP サーバー, 27
- EipControl
  - EtherNet/IP Scanner の開始または停止, 135
- EipDataExch
  - Explicit メッセージの送信, 137
- EipGetHealth
  - EtherNet/IP 接続のヘルスビットの読み込み, 136
- EIPGetHealthBit
  - EtherNet/IP のヘルスビット値の取得, 125
- EIPStartAllConnection
  - すべての EtherNet/IP 接続の開始, 122
- EIPStartConnection
  - 接続の開始, 121
- EIPStopAllConnection
  - すべての EtherNet/IP 接続の開始, 124
- EIPStopConnection
  - EtherNet/IP 接続の停止, 123
- EtherNet/IP
  - EipDataExch, 137
- EtherNet/IP Explicit メッセージ通信
  - EipDataExch で送信, 137
  - EIPGetHealthBit, 125
  - EIPStartAllConnection, 122
  - EIPStartConnection, 121
  - EIPStopAllConnection, 124
  - EIPStopConnection, 123
  - Get\_Attribute\_All, 113
  - Get\_Attribute\_Single, 117
  - Set\_Attribute\_All, 115
  - Set\_Attribute\_Single, 119
- EtherNet/IP Scanner
  - EipControl, 135
  - EipDataExch, 137
  - EipGetHealth, 136
- FDR サービス, 27
- Get\_Attribute\_All
  - オブジェクトのすべての属性の取得, 113
- Get\_Attribute\_Single
  - オブジェクトの属性の取得, 117
- IOS\_GETHEALTH
  - チャンネルのヘルスビット値の取得, 103
- IOS\_GETSTATE
  - Modbus Serial IOScanner の状態の読み込み, 101
- IOS\_START
  - Modbus Serial IOScanner の起動, 102
- IOS\_STOP
  - Modbus Serial IOScanner の停止, 104
- IoStateCodes
  - データ型, 108
- IP アドレスの指定方式, 27
- M251 Web サーバー
  - 産業用 Ethernet マネージャー, 87
- Modbus Serial IOScanner
  - CONFIGURE\_OTB, 105
  - IOS\_GETHEALTH, 103
  - IOS\_GETSTATE, 101
  - IOS\_START, 102
  - IOS\_STOP, 104
- OperationErrorCodes
  - データ型, 129, 142
- Set\_Attribute\_All
  - インスタンスまたはクラスの属性の設定, 115
- Set\_Attribute\_Single
  - オブジェクトの属性の設定, 119
- SoMachine を介した監視
  - 産業用 Ethernet マネージャー, 90
- TCP\_ADDR
  - データ型, 143
- アウトオブプロセスのデータ交換, 79
- インスタンス
  - Set\_Attribute\_All, 115
- オブジェクト
  - Get\_Attribute\_All, 113
- クラス
  - Set\_Attribute\_All, 115
- データ交換、アウトオブプロセス, 79
- データ型
  - CommunicationErrorCodes, 109, 128, 141
  - configurationOTBErrorCodes, 110
  - IoStateCodes, 108
  - OperationErrorCodes, 129, 142
  - TCP\_ADDR, 143
- トラブルシューティング
  - 産業用 Ethernet マネージャー, 93
- バス周期タスク
  - Modbus TCP IOScanner, 75
- ファンクション
  - IL 言語でのファンクションおよびファンクションブロックの使用法, 151
  - ST 言語でのファンクションおよびファンクションブロックの使用法, 154
  - ファンクションとファンクションブロックの相違, 150
- ヘルスビット
  - EipGetHealth, 136
  - EIPGetHealthBit, 125
  - IOS\_GETHEALTH, 103
- 動作モード
  - 産業用 Ethernet マネージャー, 81
- 属性
  - Get\_Attribute\_Single, 117
  - Set\_Attribute\_Single, 119
- 状態
  - 産業用 Ethernet マネージャー, 81
- 産業用 Ethernet マネージャー
  - M251 Web サーバー, 87
  - SoMachine を介した監視, 90
  - トラブルシューティング, 93
  - 動作モード, 81
  - 状態, 81

設定ツール, 79