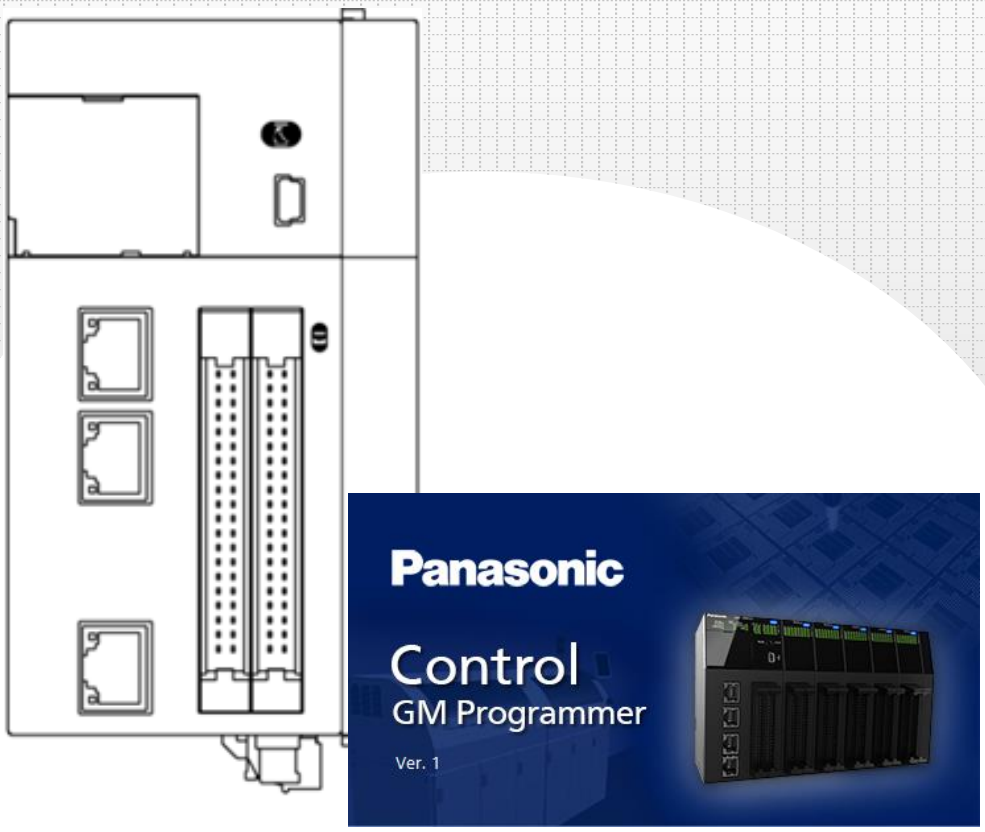
Panasonic

Hello! GM1 通信編



memo

著作権および商標に関する記述

・Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標です。
・Ethernet は富士ゼロックス株式会社および米国 Xerox Corporation の登録商標です。
・EtherCAT は、ドイツ Beckhoff Automation GmbH によりライセンスされた特許取得済み技術であり登録商標です。
・EtherNet/IP は、ODVA（Open DeviceNet vender Association）の商標登録です。
・SDHC、SD ロゴは、SD-3C、LLC の商標です。
・その他の会社および製品名は、各社の商標または商標登録です。

安全上の注意事項

人への危害、財産の損害を防止するため、必ずお守りいただくことを説明しています。
・誤った使い方をしたときに生じる危害や損害の程度を区分して説明しています。

⚠ 警告　「死亡や重傷を負うおそれがある内容」です。
⚠ 注意　「軽傷を負うことや、財産の損害が発生するおそれがある内容」です。

🚫 してはいけない内容です。
❗ 実行しなければならない内容です。

⚠ 警告

❗ ・本製品の故障や外部要因による異常が発生しても、システム全体が安全側に働くように本製品での外部で安全対策を行ってください。
🚫 ・燃焼性ガスの雰囲気中は使用しないでください。爆発の原因となります。
🚫 ・本製品を火中に投棄しないでください。電池や電子部品などが破裂する原因となります。

⚠ 注意

❗ ・異常発熱や発煙を防止するため、本製品の保証特性・性能の数値に対し余裕をもたせて使用してください。
🚫 ・分解、改造はしないでください。異常発熱や発煙の原因となります。
🚫 ・通電中は端子に触れないでください。
❗ ・非常停止、インターロック回路は外部で構成してください。
❗ ・電線やコネクタは確実に接続してください。接続不十分な場合は、異常発熱や発煙の原因となります。
🚫 ・電源を入れた状態では施工（接続、取り外しなど）しないでください。
❗ ・弊社が指定していない方法で使用すると、ユニットの保護機能が損なわれることがあります。
❗ ・本製品は、工場環境に使用する目的で開発／製造された製品です。

本テキストの記載内容と責任の範囲

本テキストは GM1 シリーズの立ち上げ手順と GM Programmer の操作方法について記載したものであり、安全に関する注意事項や、各機器の使用上の注意事項については記載していません。
必ず、本テキストで使用する機器のマニュアルや取扱説明書を入手し、安全に関する注意事項や使用上の注意事項についてご確認のうえ使用してください。
当社商品やソフトウェア、本テキストに関連して生じた損害について、当社は責任を負いません。

GM1　通信編

0　事前準備

ツールソフトのインストール
- ・GM Programmer
- ・PANATERM Lite for GM

■EtherNet 通信

Modbus TCP　マスタ／スレーブ

1　基本設定

1−1　動作イメージ
1−2　必要な機器の準備～配線
1−3　RTEX タイプ:マスタ　IP アドレス設定～ネットワークスキャン
1−4　EtherCAT タイプ:スレーブ　IP アドレスの設定～USB 追加

2　スレーブ側設定

2−1　デバイスの追加
2−2　構造体の宣言
2−3　グローバル変数の宣言
2−4　読み出し／書き込み変数の設定～ログイン

3　マスタ側設定～プログラム作成

3−1　デバイスの追加
3−2　読み出し／書き込み設定
3−3　書き込みプログラム作成(トリガ:立ち上がりエッジ)
3−4　書き込みプログラム作成(トリガ:アプリケーション)

4　通信動作の確認

EtherNet/IP　スキャナ／アダプタ

1　基本設定

1−1　動作イメージ
1−2　必要な機器の準備～配線
1−3　RTEX タイプ:スキャナ　IP アドレスの設定～ネットワークスキャン
1−4　EtherCAT タイプ:アダプタ　IP アドレスの設定～USB 追

2　スキャナ側設定

2−1　デバイスの追加
2−2　デバイスの設定
2−3　変数の登録

3　アダプタ側設定

3−1　デバイスの追加
3−2　モジュールの設定

4　通信動作の確認

0 事前準備

ツールソフトのインストール

以下 Web サイトより、GM Programmer のインストールをお願い致します。
GM Programmer　:https://industrial.panasonic.com/ac/j/motor/motion-controller/mc/gm1/index.jsp

INFO

GM Programmer をインストールすると、PANATERM Lite for GM と Gateway（CODESYS Gateway）、
CodeMeter アプリケーションも同時にインストールされます。

・GM Programmer：GM1 コントローラの設定ツールです。GM Programmer を使用することで、位置決めデータや各種位置決めパラメータの設定、各種モニタが可能です。
・PANATERM Lite for GM1（今回は使用しません）：パナソニック製サーボアンプ MINAS シリーズのセットアップ支援ツールです。
GM Programmer をインストールすると、同時に“PANATERM Lite for GM”がインストールされます。
パソコン画面上で、サーボアンプ内部のパラメータ設定や制御状態の監視あるいはセットアップ支援、機械の分析などが実行できるツールです。

PC にインストールする際は、PC に Administrator 権限にてログインしてください。
他のアプリケーションを起動している場合、インストールする前に必ずすべてのアプリケーションを終了してください。

本テキストは GM1RTEX タイプ、EtherCAT タイプ 1 台ずつ使用していますが、両方の通信仕様に違いはありません。
対応機種：AGM1CSRX16T、AGM1CSEC16T、AGM1CSEC16P

EtherNet 通信　Modbus TCP マスタ／スレーブ

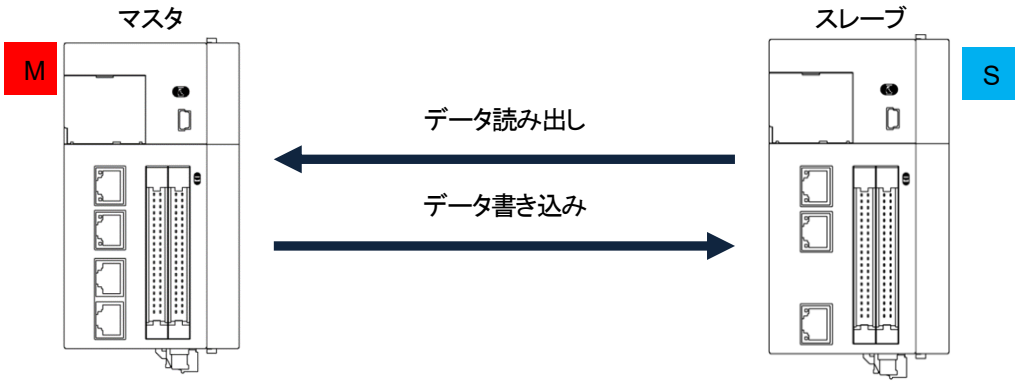
1 基本設定

ModbusTCP マスタ機能を使用して、スレーブ機器にコマンドを送信する方法は下記の 2 通りあります。
1)デバイスオブジェクト設定を使用する場合
●スレーブ初期化で送信する方法
●送信方法
・サイクリック
・立ち上がりエッジ
・アプリケーション(ModbusChannel ファンクションブロック)
2)デバイスオブジェクト設定を使用しない場合
●ユーザプログラム上(ModbusRequest ファンクションブロック)でコマンドを生成し、送信する方法

本テキストでは、1)デバイスオブジェクト設定を使用して、スレーブ機器にコマンドを送信します。

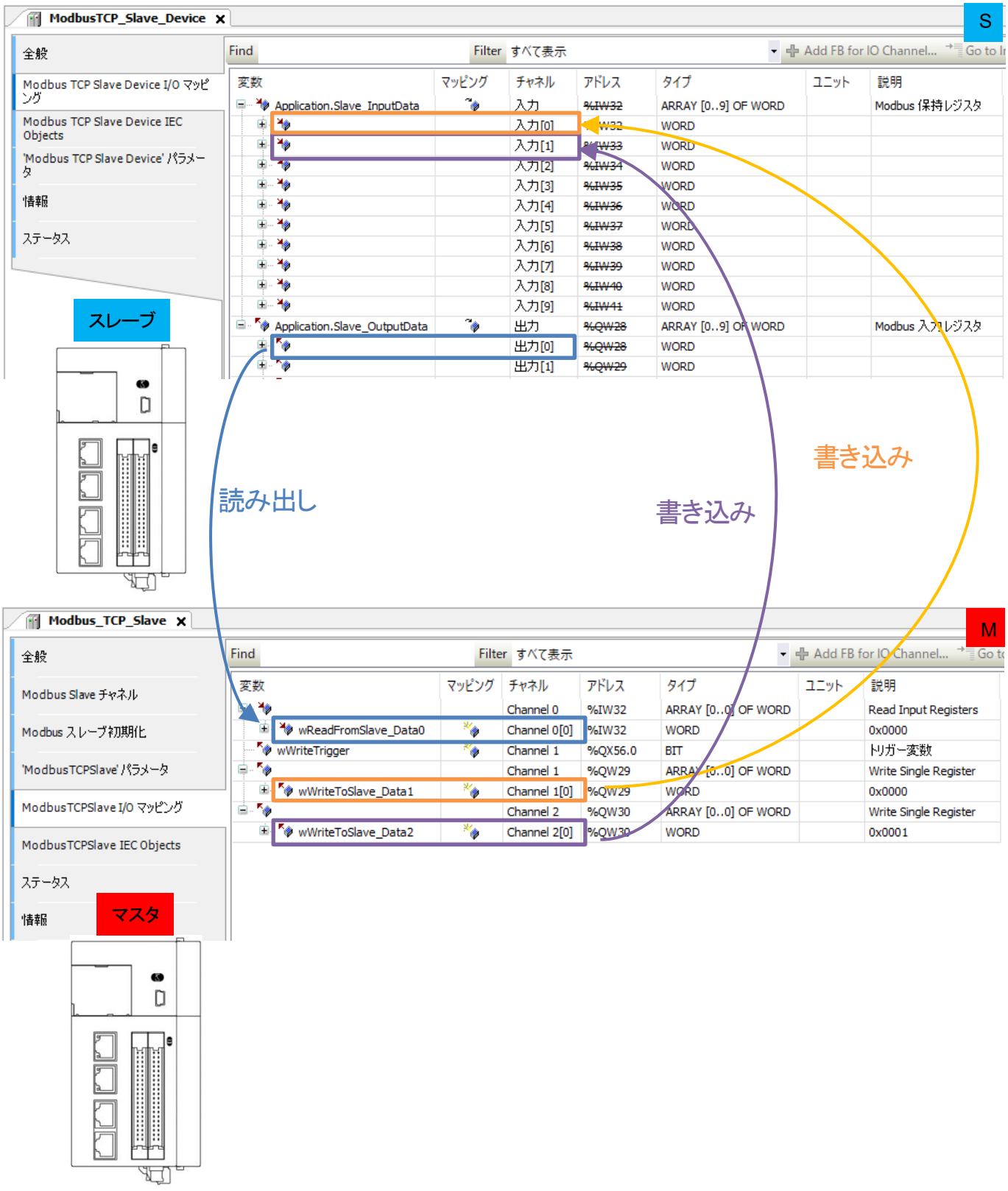
1－1 動作イメージ

Modbus TCP
2 台の GM1 コントローラをマスタ／スレーブで使用します。
お互いの LAN Port2 を使用して、Modbus TCP マスタ通信を行います。



※マスタとスレーブで 2 つの GM Programmer を立ち上げるため、分かりやすく見分けるために、
GM Programmer の画像右上に M と S を付けています。
M :マスタ
S :スレーブ

スレーブ側 GM1 の「ModbusTCP_Slave_Device」で設定する変数と、
マスタ側 GM1 の「Modbus_TCP_Slave」で設定する変数でデータの書き込み／読み出しを実施します。



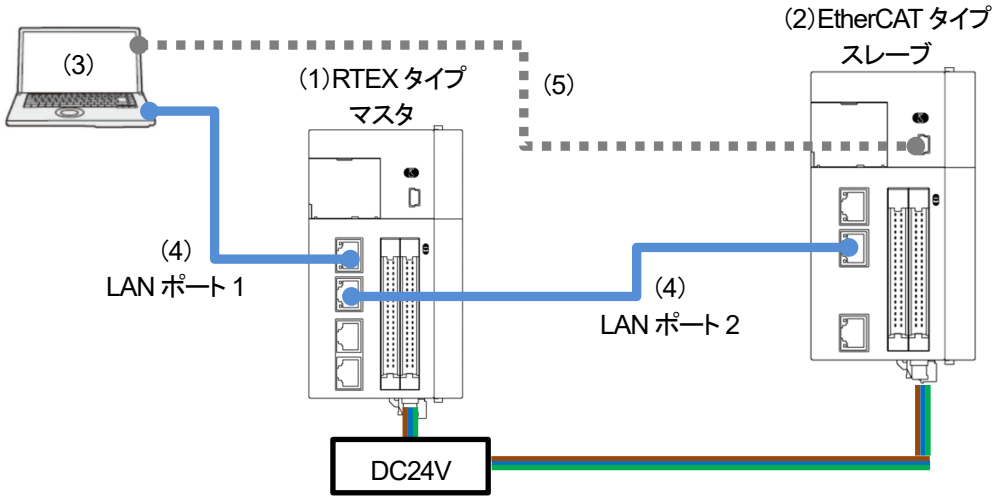
1－2　必要な機器の準備～配線

以下の機器を用意してください。

No.	名称	
(1)	GM1 コントローラ 1 台(RTEX タイプ):マスタ	(本テキストでは、RTEX タイプと EtherCAT タイプ 1 台ずつ使用)
(2)	GM1 コントローラ 1 台(EtherCAT タイプ):スレーブ	
(3)	PC(GM Programmer インストール済み)	
(4)	LAN ケーブル:2 本	
(5)	USB ケーブル(mini-b)	

※本テキストでは、RTEX タイプと EtherCAT タイプを 1 台ずつ使用していますが、双方の通信仕様に違いはありません。

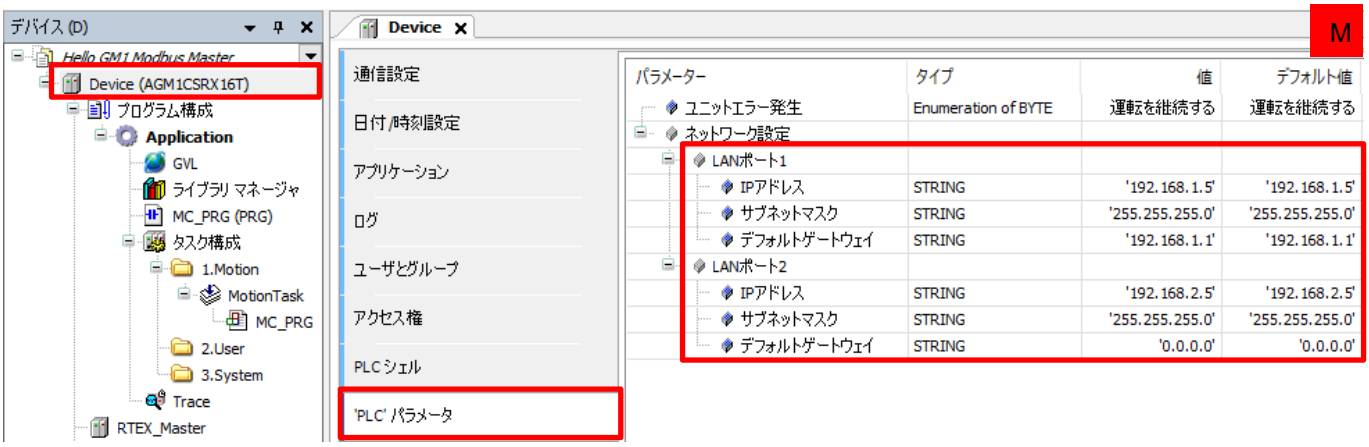
下図のように配線してください。



1－3　RTEX タイプ：マスタ　IP アドレスの設定～ネットワークスキャン

手順 1

GM Programmer を開き、「Device」をダブルクリックします。
「PLC パラメータ」を選択し、LAN ポート 1 と LAN ポート 2 の IP アドレスを確認します。



LAN ポート 1（初期値）

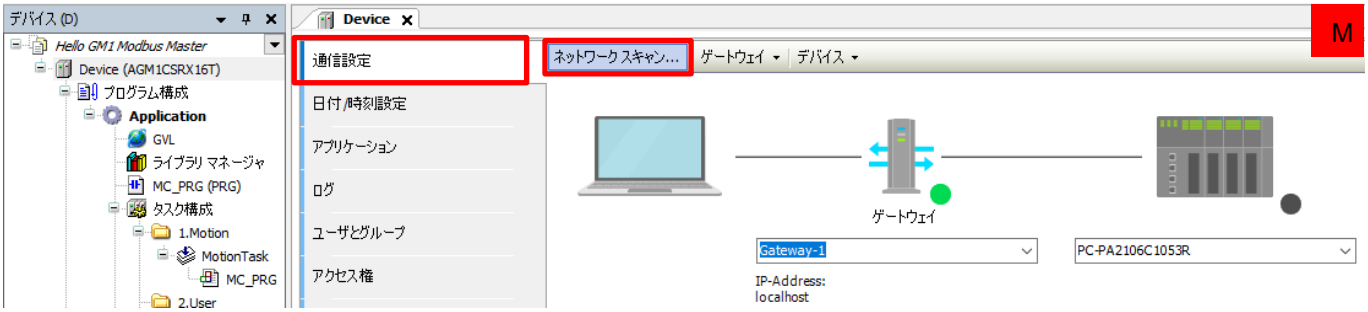
IP アドレス	192.168.1.5
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	192.168.1.1

LAN ポート 2（初期値）

IP アドレス	192.168.2.5
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	0.0.0.0

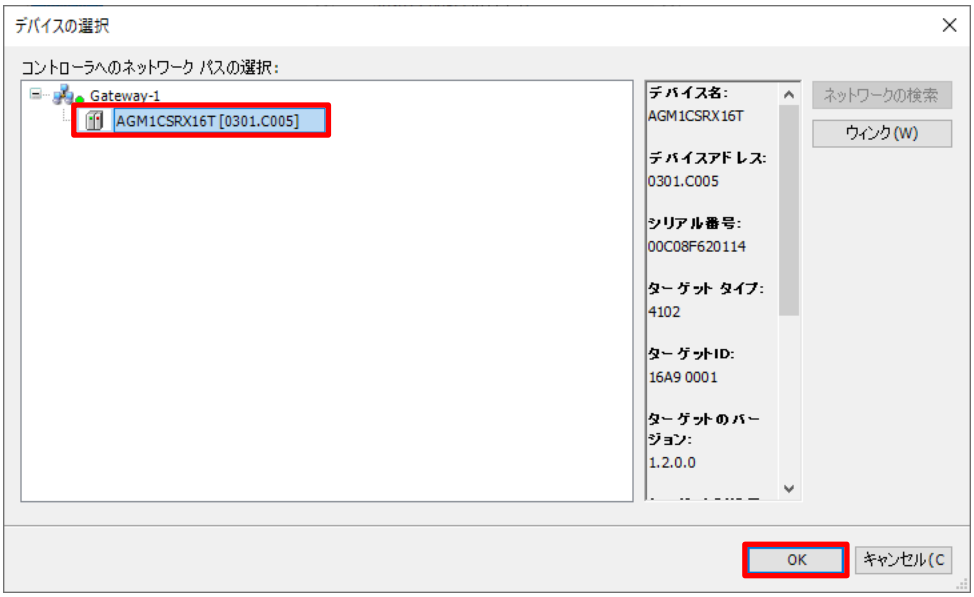
手順 2

「通信設定」を選択し、「ネットワークスキャン」をクリックします。



手順 3

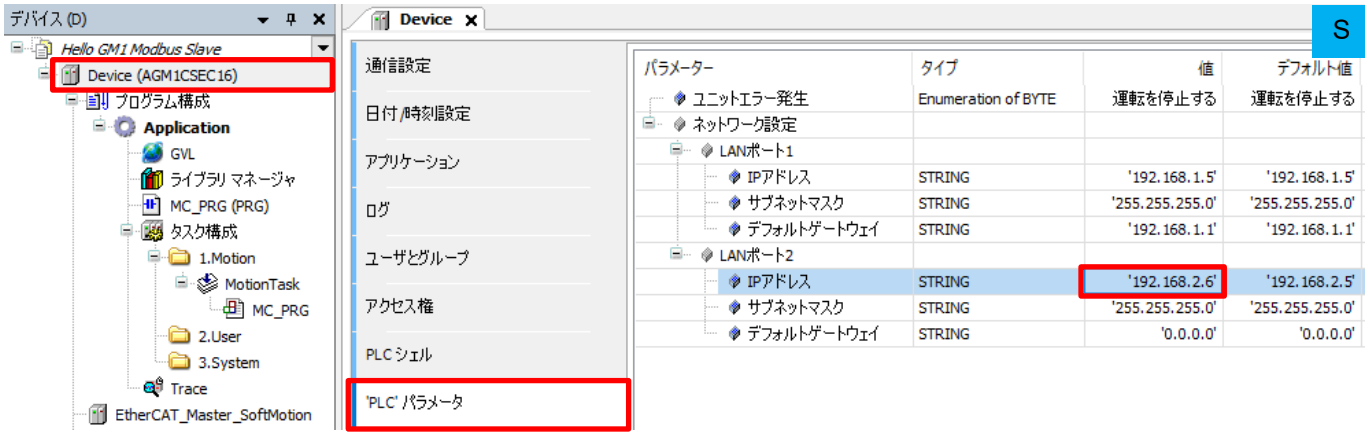
接続するデバイスを選択し「OK」をクリックします。



1－4　EtherCAT タイプ：スレーブ　IP アドレスの設定～USB 追加

手順 1

GM Programmer を開き、「Device」をダブルクリックします。
「PLC パラメータ」を選択し、LAN ポート 2 の IP アドレスを「192.168.2.6」に変更します。

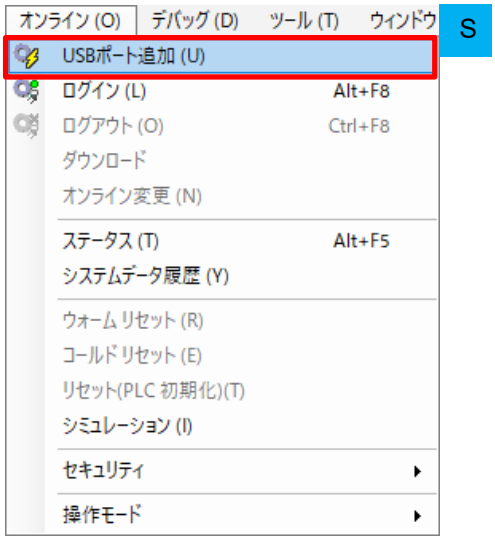


LAN ポート 2

IP アドレス	192.168.2.6
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	0.0.0.0

手順 2

メニューバーの**オンライン→USB ポート追加**をクリックします。



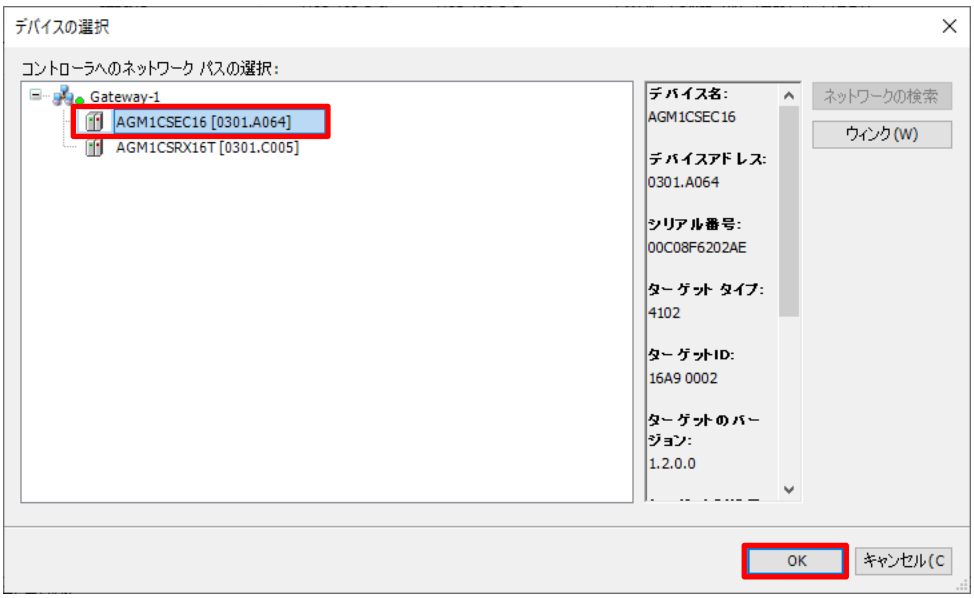
手順 3

「USB ポート追加」ダイアログが表示されます。デバイスと使用ポートを確認して「OK」をクリックします。

手順４

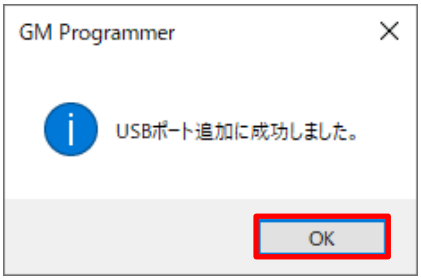
「デバイスの選択」ダイアログが表示されます。

接続するデバイスを選択し「OK」をクリックします。



手順５

接続が完了すると、PC と GM1 コントローラ間の通信インターフェイスに USB が追加されます。



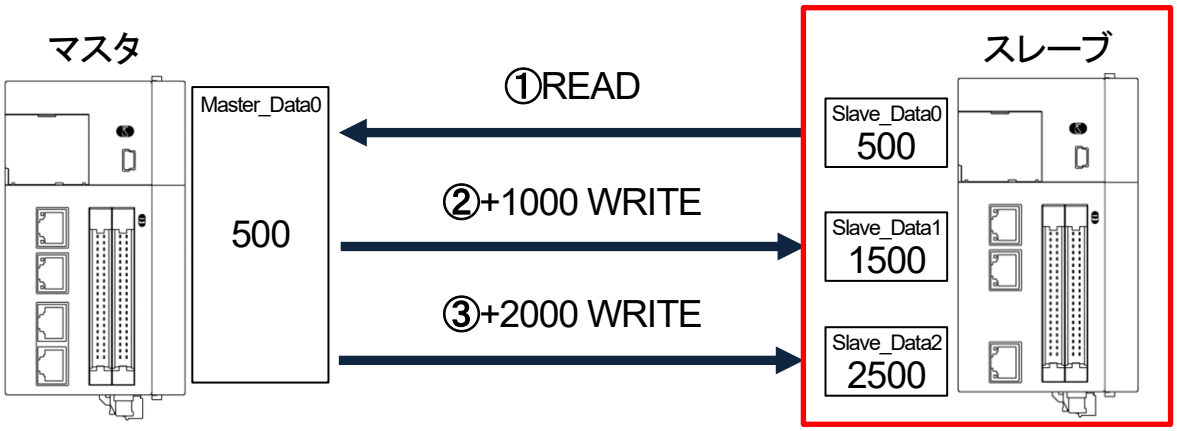
コラム①　GM1 コントローラ対応ファンクションコード

ファンクションコード	アクセスタイプ	内容	アドレス
1	Read Coils	コイル読み出し	%IX
2	Read Discrete Inputs	ディスクリート入力読み出し	%QX
3	Read Holding Registers	保持レジスタ読み出し	%IW
4	Read Input Registers	入力レジスタ読み出し	%QW
5	Write Single Coil	単一コイル書き込み	%IX
6	Write Single Register	単一レジスタ書き込み	%IW
15	Write Multiple Coils	複数コイル書き込み	%IX
16	Write Multiple Registers	複数レジスタ書き込み	%IW
23	Read/Write Multiple Registers	複数レジスタ読み出し／書き込み	%QW／%IW

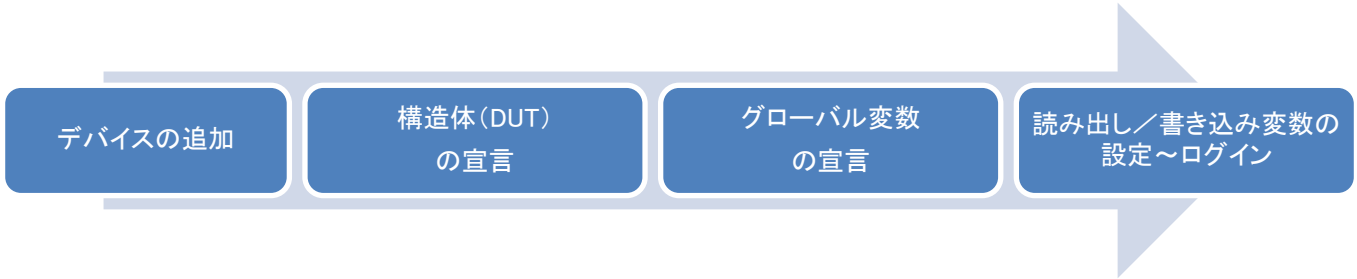
本テキストでは、ファンクションコード 4 とファンクションコード 6 を使用しています。

2 スレーブ側設定

①スレーブの Slave_Data0 の値を読み出します。(ファンクションコード:4)
②+1,000 した値をスレーブの Slave_Data1 に(トリガ:立ち上がりエッジで)書き込みます。(ファンクションコード:6)
③+2,000 した値をスレーブの Slave_Data2 に(トリガ:アプリケーションで)書き込みます。(ファンクションコード:6)



以下の順番で、スレーブ側の設定からしていきます。

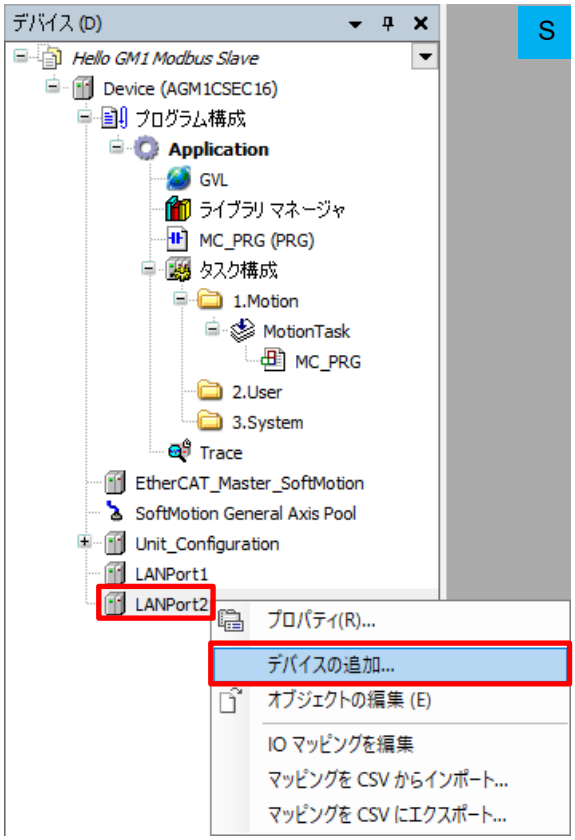


2－1　デバイスの追加

スレーブからのデータを読み出すマスタを追加します。

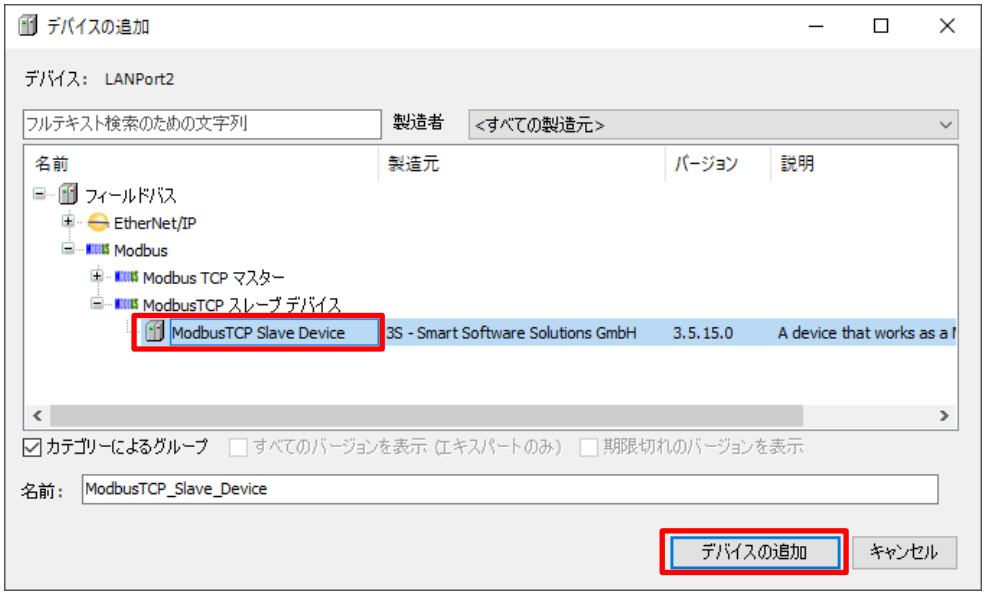
手順 1

LAN Port2 に Modbus TCP マスタのデバイスを追加します。
ナビゲータウィンドウの「LANPort2」を右クリックして、「デバイスの追加」をクリックします。



手順 2

表示されたダイアログの「Modbus」ー「Modbus TCP スレーブデバイス」ー「Modbus TCP Slave Device」を選択し、
「デバイスの追加」をクリックします。



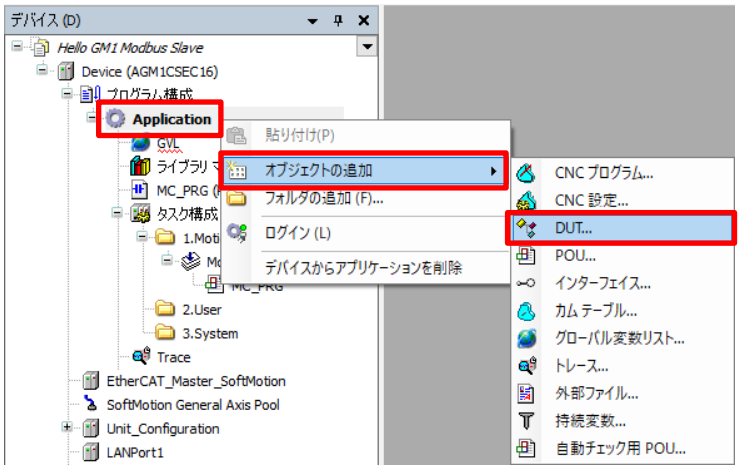
2－2　構造体（DUT）の宣言

INFO

DUT 構造体は Data Unit Type の略で、複数の異なるデータ型をもつ変数で構成されます。
構造体をまず定義し、その後、標準的なデータ型（BOOL、INT など）と同様にグローバル変数リストや POU ヘッダで使用します。

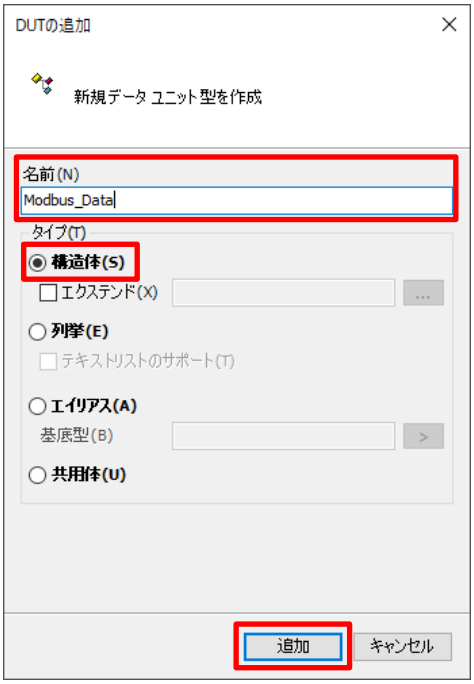
手順 1

Application で右クリックし、「オブジェクトの追加」－「DUT」をクリックします。

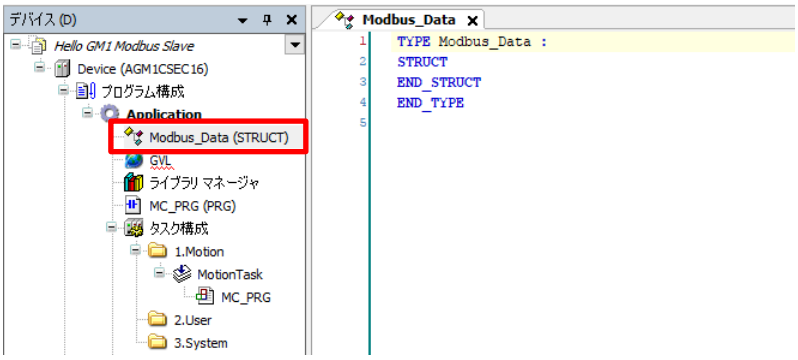


手順 2

「DUT の追加」ダイアログが表示されるので、任意の名前を付けます。
タイプ：構造体を選択し、「追加」をクリックします。



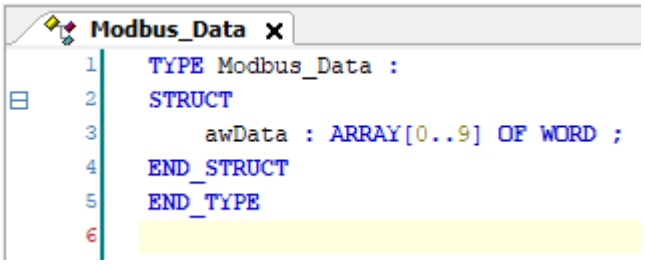
Modbus_Data（STRUCT）が追加されました。



手順３

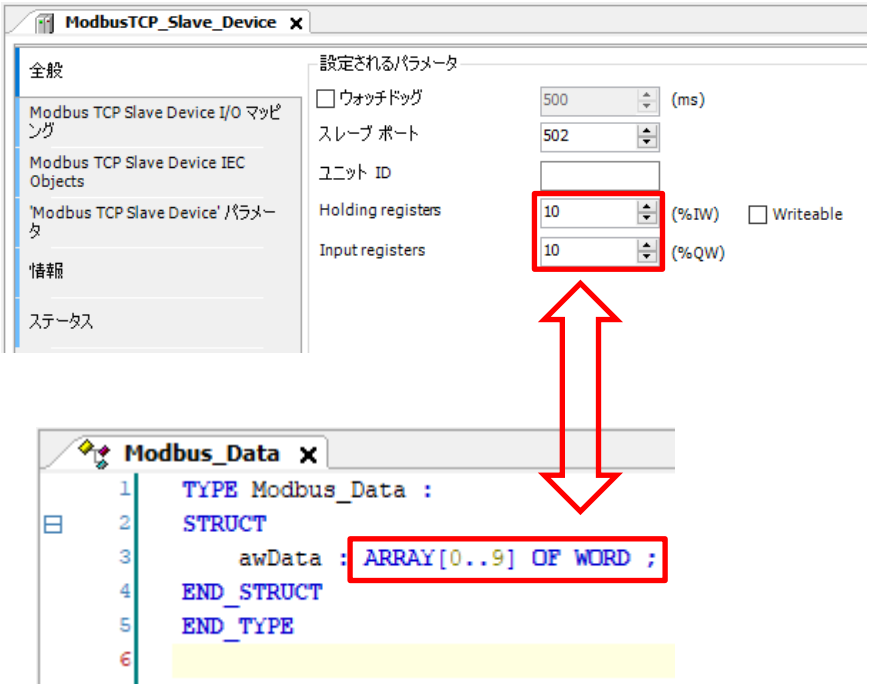
下図のように構造体を構成します。

変数名	データ型
awData	ARRAY[0..9]OF WORD



INFO

「2−4　読み出し／書き込み変数の設定～ログイン」で設定する「Holding registers」と「Input registers」の設定値を「10」WORD にします。それらの数と、構造体の中身の数を合わせる必要があります。



以上で、構造体の宣言は完了です。

コラム② レジスタとアドレスの関係

レジスタ	アドレス
保持レジスタ(WORD)	%IW
保持レジスタ(BOOL)	%IX
入力レジスタ(WORD)	%QW
入力レジスタ(BOOL)	%QX

変数	マッピング	チャネル	アドレス	タイプ
		入力[8]	%IW40	WORD
		入力[9]	%IW41	WORD
		Bit0	%IX82.0	BOOL
		Bit1	%IX82.1	BOOL
		Bit2	%IX82.2	BOOL
		Bit3	%IX82.3	BOOL
		Bit4	%IX82.4	BOOL
		Bit5	%IX82.5	BOOL
		Bit6	%IX82.6	BOOL
		Bit7	%IX82.7	BOOL
		Bit8	%IX83.0	BOOL
		Bit9	%IX83.1	BOOL
		Bit10	%IX83.2	BOOL
		Bit11	%IX83.3	BOOL
		Bit12	%IX83.4	BOOL
		Bit13	%IX83.5	BOOL
		Bit14	%IX83.6	BOOL
		Bit15	%IX83.7	BOOL
		出力	%QW28	ARRAY [0..9] OF WORD
		出力[0]	%QW28	WORD
		Bit0	%QX56.0	BOOL
		Bit1	%QX56.1	BOOL

「ModbusTCP_Slave_Device」の全般タブで、「Writeable」に ☑ を入れることで、書き込みも可能になり、アドレスも「%QW」に変更されます。

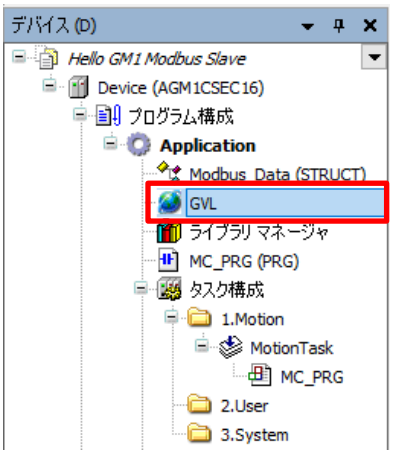
2−3 グローバル変数の宣言

INFO

グローバル変数は、プロジェクト全体にわたって使用できます。

手順 1

「GVL」をダブルクリックします。



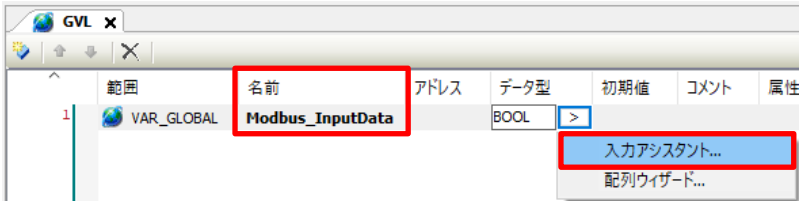
手順 2

グレーの個所で右クリックし、「挿入」をクリックします。

手順 3

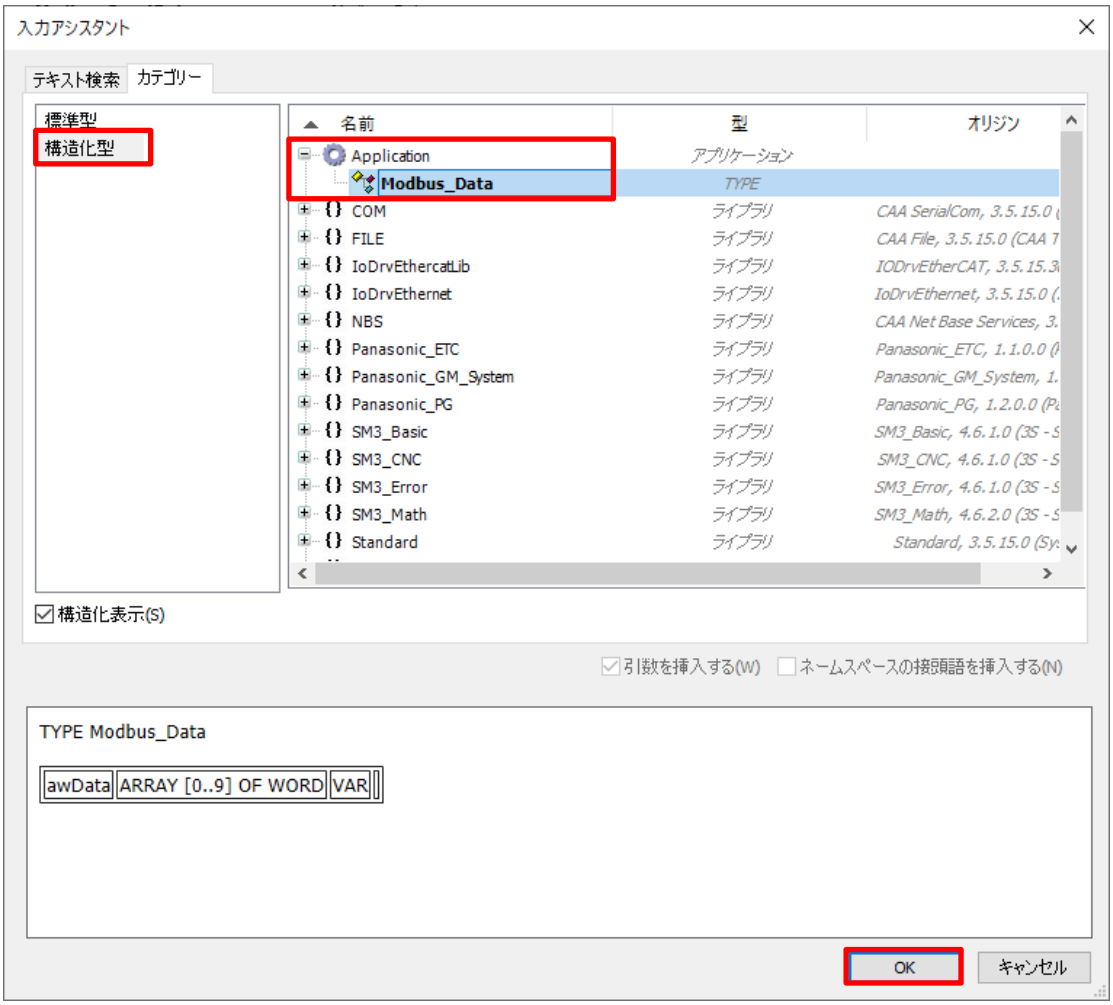
「名前」に「Modbus_InputData」を入力します。

「データ型」をダブルクリックし、「入力アシスタント」をクリックします。



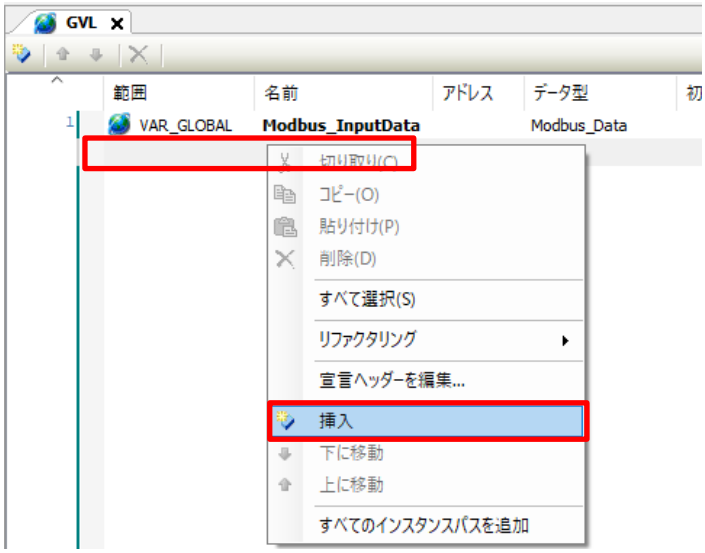
手順４

「入力アシスタント」ダイアログが表示されますので、「構造化型」を選択し、「Application」ー「Modbus_Data」を選択し、「OK」をクリックします。



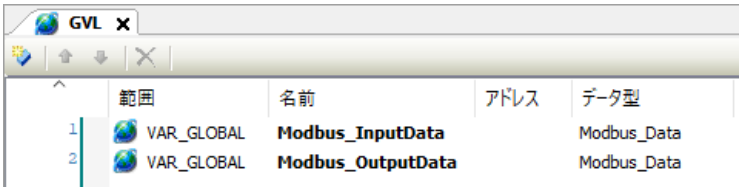
手順５

挿入された「Modbus_InputData」の下で右クリックし、「挿入」をクリックします。



手順６

「名前」に「Modbus_OutputData」と入力します。

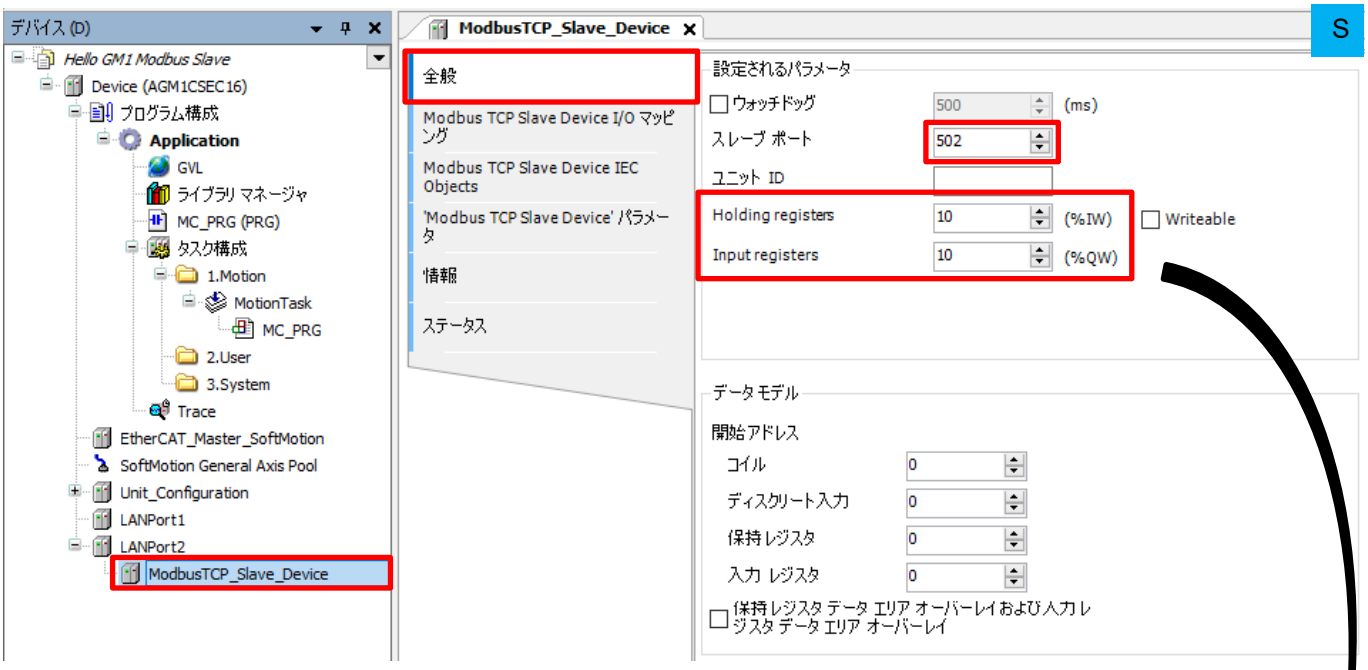


以上で、グローバル変数の宣言は完了です。

2−4　読み出し／書き込み変数の設定～ログイン

手順 1

追加された「ModbusTCP_Slave_Device」をダブルクリックし、「全般」を選択します。
「スレーブポート」を 502（初期値）に設定します。



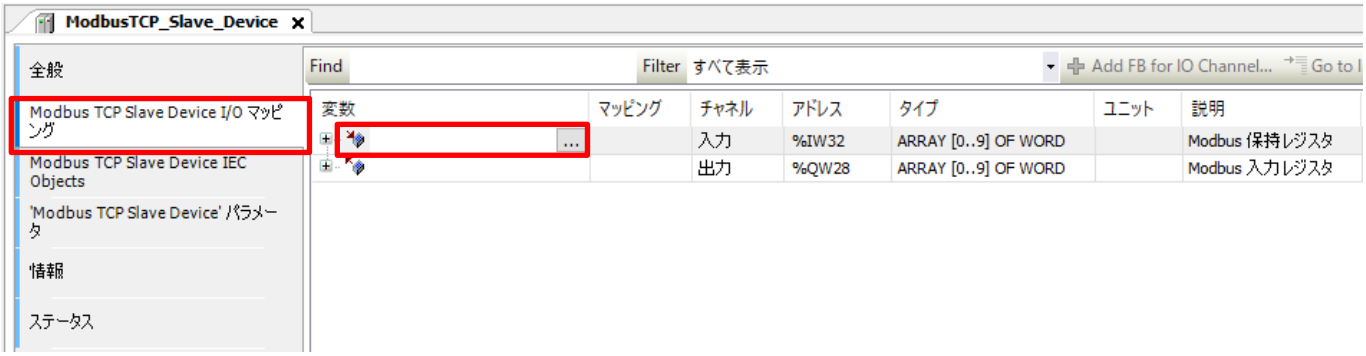
INFO

上記「ModbusTCP_Slave_Device」ー「全般」の「Holding registers」と「Input registers」が「10」になっています。
それは、下図の入力[0]～入力[9]の 10 個の保持レジスタと、出力[0]～出力[9]の 10 個の入力レジスタがエリアとして確保されます。

変数	マッピング	チャネル	アドレス	タイプ	ユニット	説明
		入力	%IW54	ARRAY [0..9] OF WORD		Modbus 保持レジスタ
		入力[0]	%IW54	WORD		
		入力[1]	%IW55	WORD		
		入力[2]	%IW56	WORD		
		入力[3]	%IW57	WORD		
		入力[4]	%IW58	WORD		
		入力[5]	%IW59	WORD		
		入力[6]	%IW60	WORD		
		入力[7]	%IW61	WORD		
		入力[8]	%IW62	WORD		
		入力[9]	%IW63	WORD		
		出力	%QW34	ARRAY [0..9] OF WORD		Modbus 入力レジスタ
		出力[0]	%QW34	WORD		
		出力[1]	%QW35	WORD		
		出力[2]	%QW36	WORD		
		出力[3]	%QW37	WORD		
		出力[4]	%QW38	WORD		
		出力[5]	%QW39	WORD		
		出力[6]	%QW40	WORD		
		出力[7]	%QW41	WORD		
		出力[8]	%QW42	WORD		
		出力[9]	%QW43	WORD		

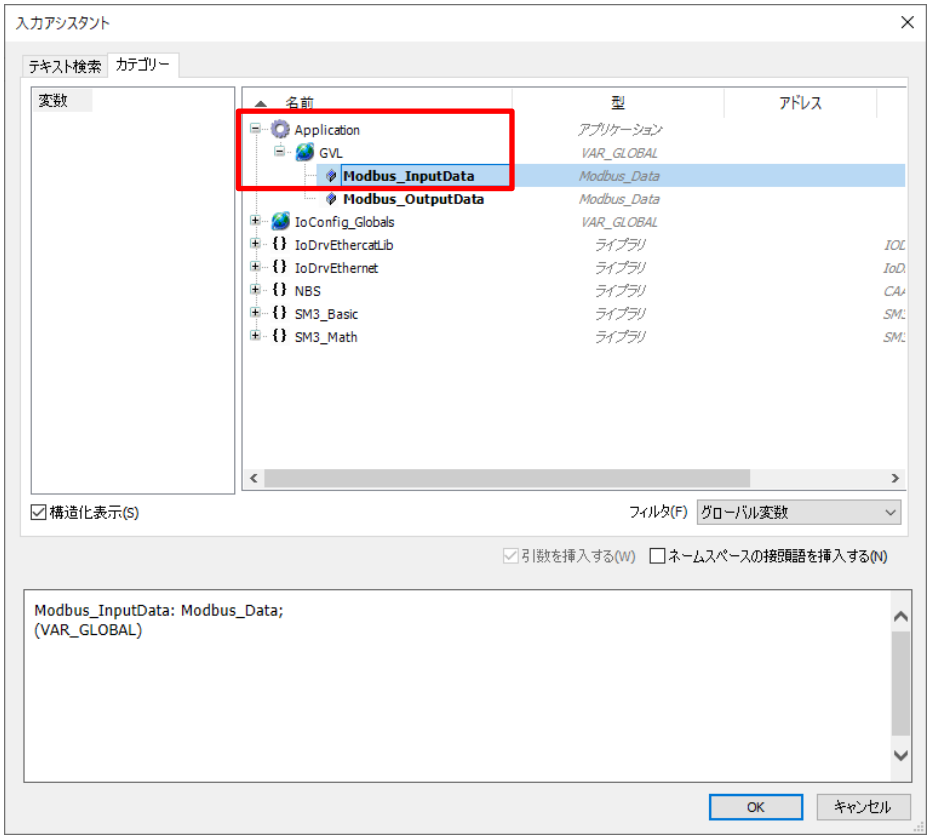
手順２

「ModbusTCP_Slave_Device I/O マッピング」を選択し、「変数」の赤枠内をダブルクリックし、 ... をクリックします。



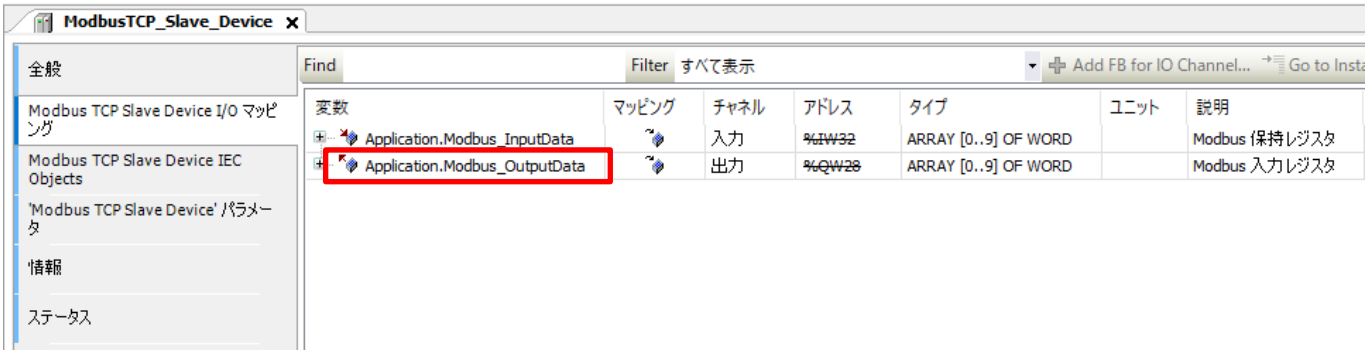
手順３

「入力アシスタント」ダイアログが表示されますので、「Application」ー「GVL」ー「Modbus_InputData」を選択し、「OK」をクリックします。



手順４

手順２ー手順３と同様にして、赤枠内に「Modbus_OutputData」を選択します。

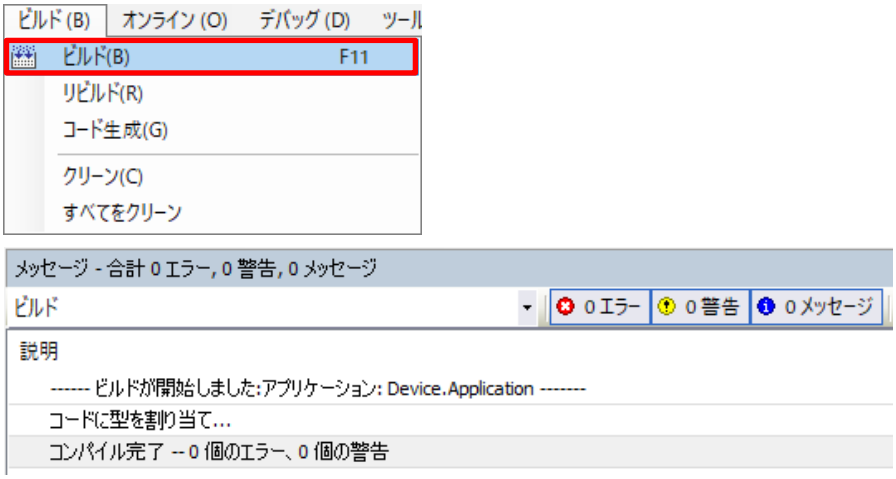


手順５

「変数を常に更新」の右側「親デバイス設定を使用」→「有効２(常にバス サイクル タスク)」に変更します。

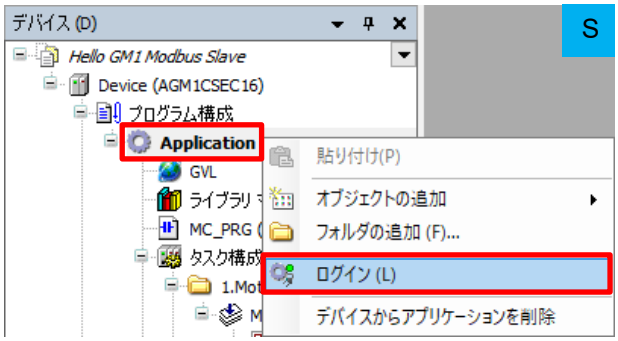
手順６

メニューバーの「ビルド」→「ビルド」をクリックし、エラーがないことを確認します。



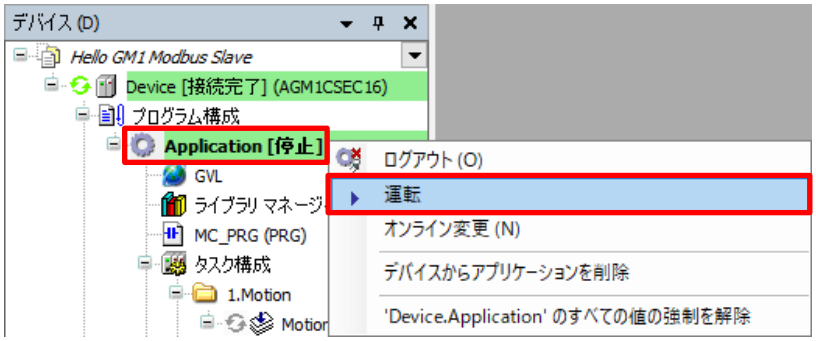
手順 7

「Application」を右クリックし、「ログイン」をクリックして、GM1 本体にダウンロードします。

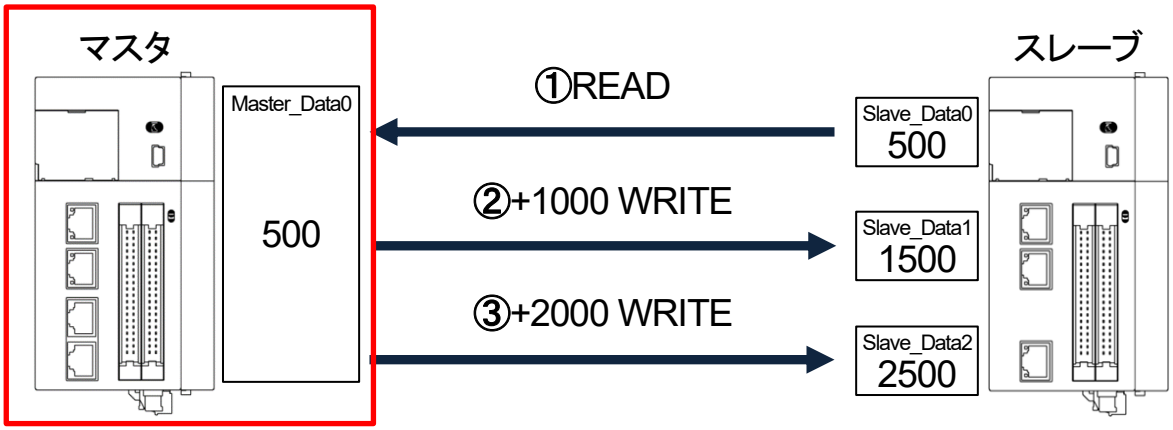


手順 8

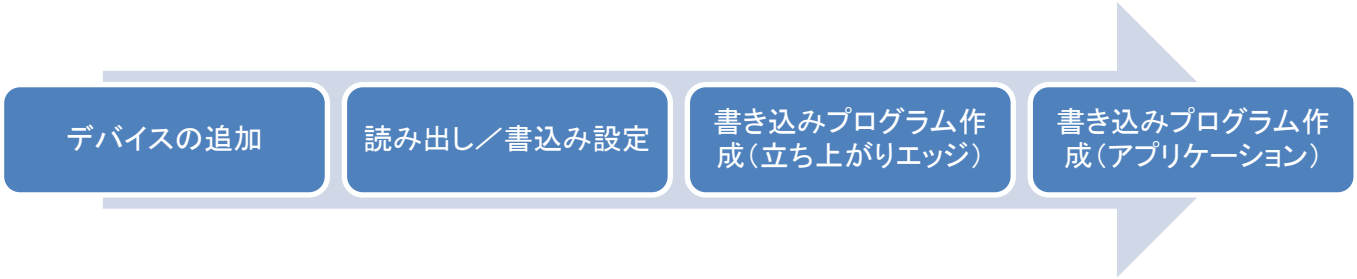
「Application[停止]」を右クリックし、「運転」をクリックして、停止→運転に切り替えます。



3 マスタ側設定～プログラム作成



以下の順番で、マスタ側の設定をしていきます。

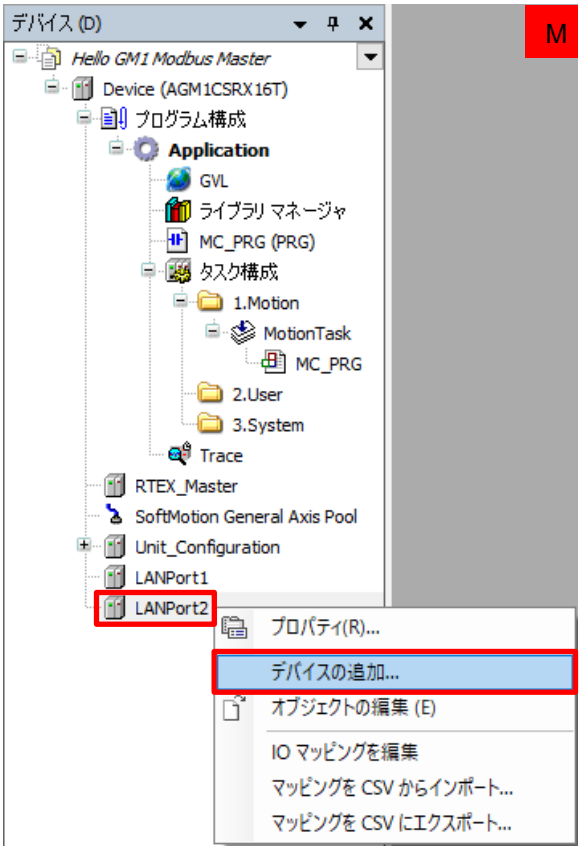


3－1　デバイスの追加

読み出すスレーブを追加します。

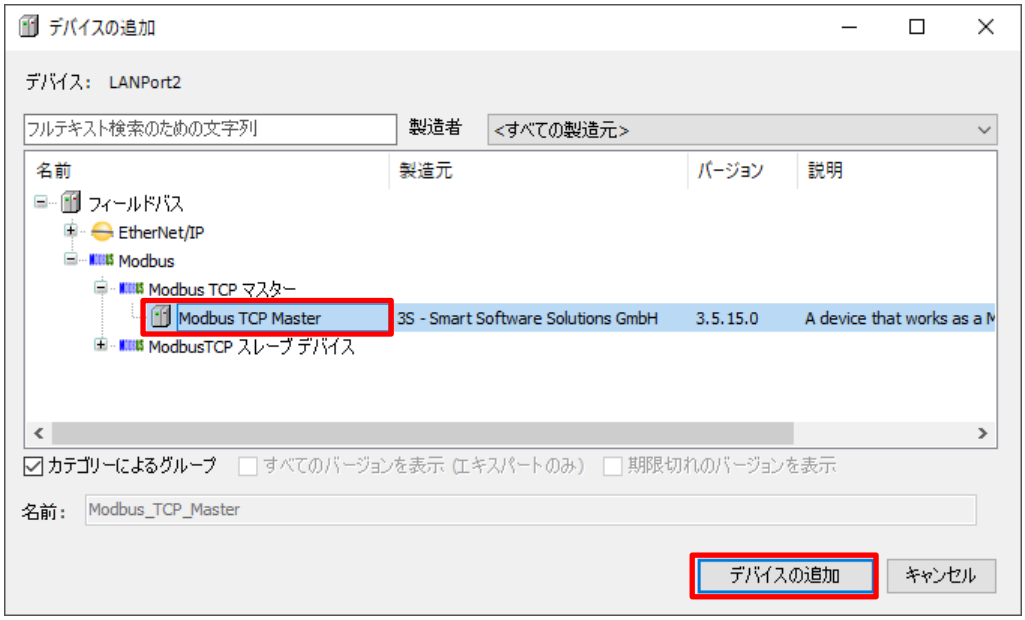
手順 1

LAN Port2 に Modbus TCP マスタのデバイスを追加します。
ナビゲータウィンドウの「LANPort2」を右クリックして、「デバイスの追加」をクリックします。



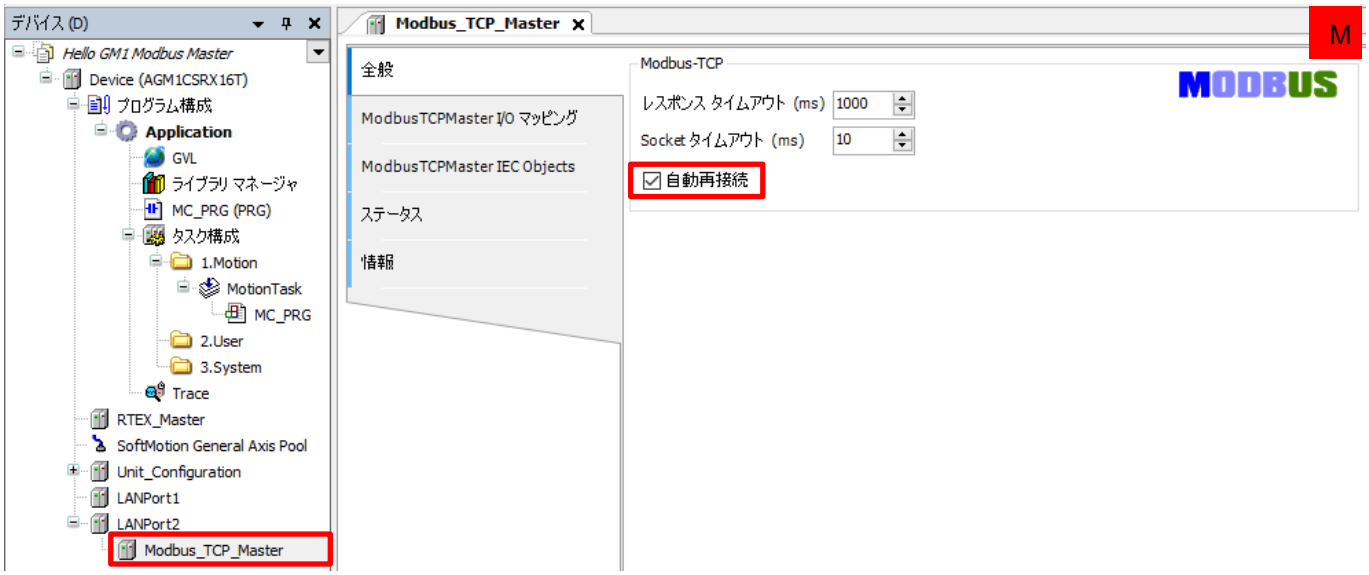
手順 2

表示されたダイアログの「Modbus」ー「Modbus TCP マスタ」ー「Modbus TCP Master」を選択し、
「デバイスの追加」をクリックします。



手順 3

追加された「ModbusTCP_TCP_Master」をダブルクリックし、「全般」を選択します。
「全般」タブの「自動再接続」に ☑ を入れてください。

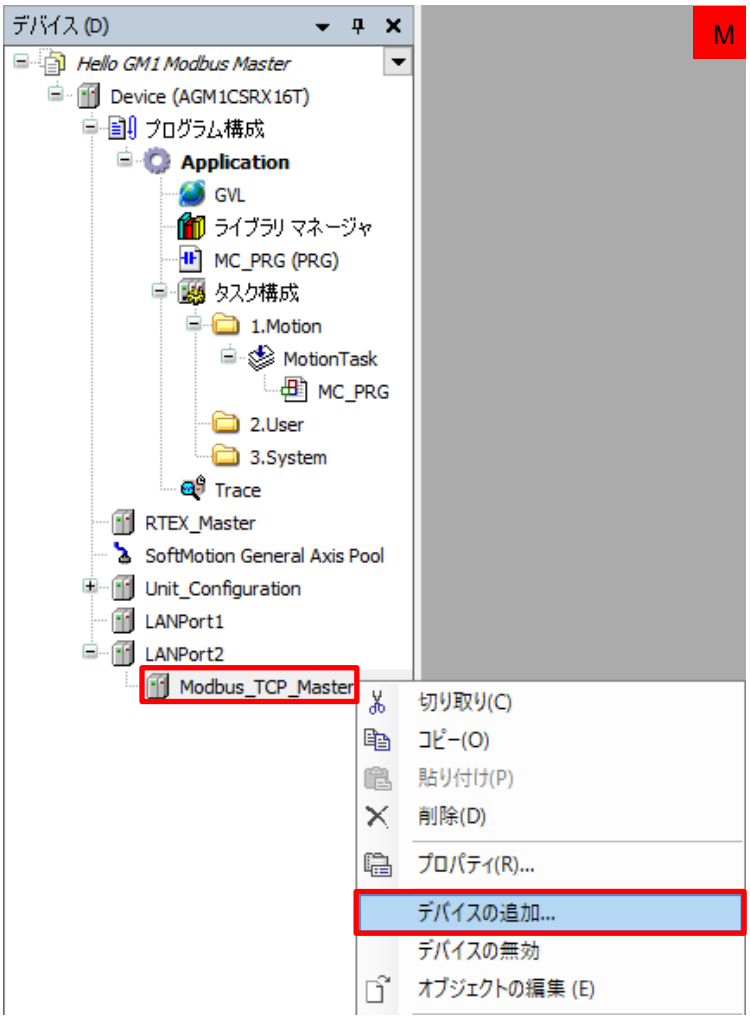


INFO

GM1 コントローラがマスタ時には、GM1 コントローラ側からコネクションオープンをかけますが、
チェックを入れない場合、タイムアウト等で接続に失敗した場合のリトライを行いません。

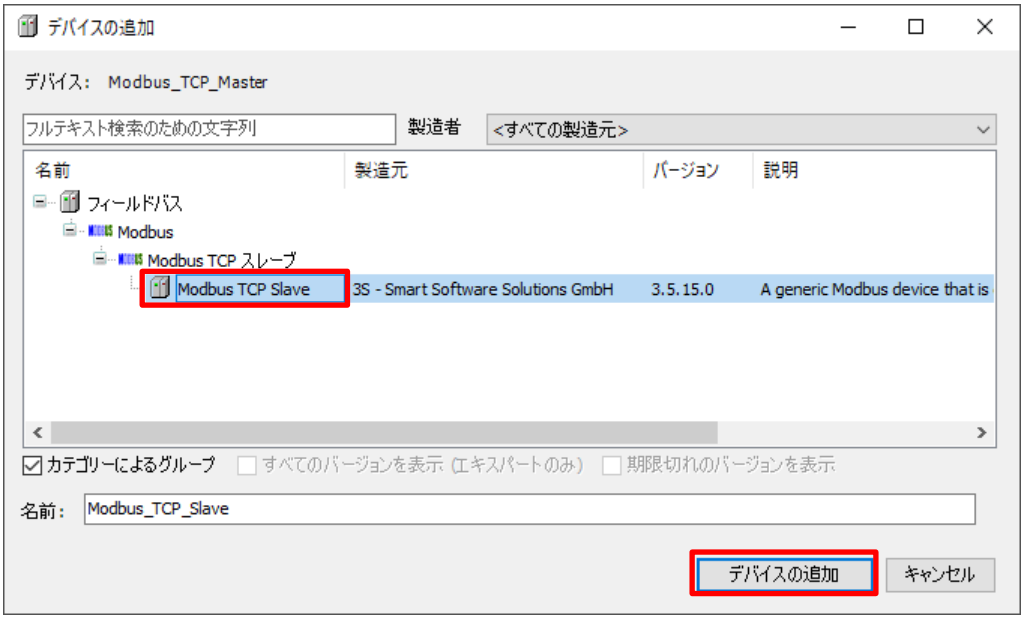
手順４

「Modbus TCP Master」を右クリックし、「デバイスの追加」をクリックします。



手順５

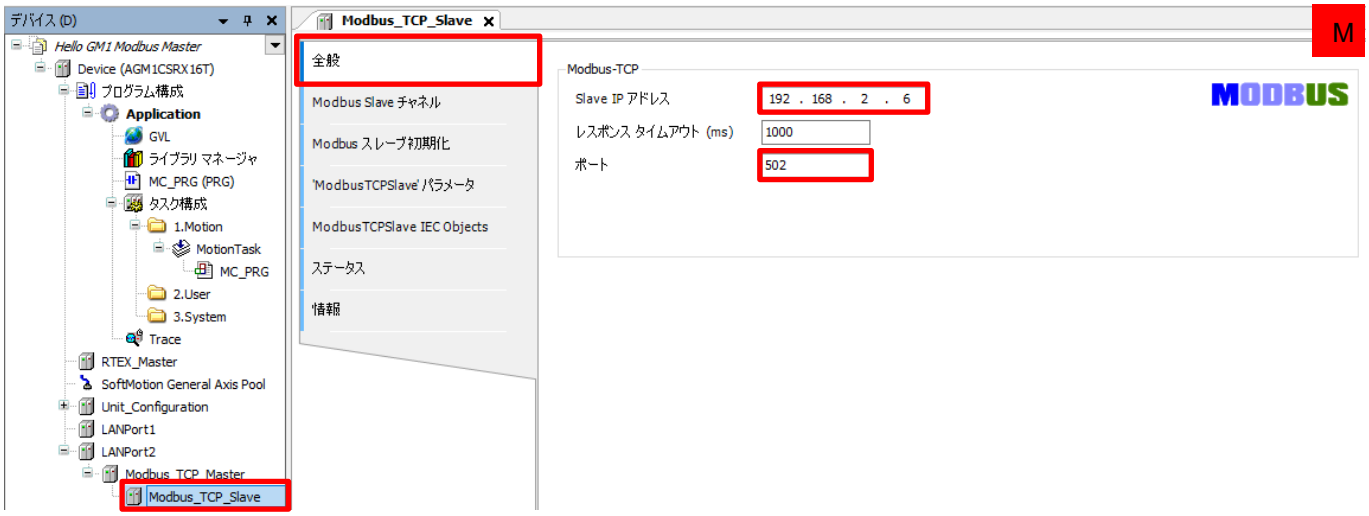
表示されたダイアログの「Modbus TCP スレーブ」ー「Modbus TCP Slave」を選択し、「デバイスの追加」をクリックします。



手順 6

「Modbus TCP Master」の下に「Modbus TCP Slave」が追加されました。
「Modbus TCP Slave」をダブルクリックし、「全般」タブを選択します。

IP アドレス／ポート No.を設定します。
スレーブ側 GM1 の IP アドレス：192.168.2.6、ポート：502 を使用するため、下図のように設定を行います。



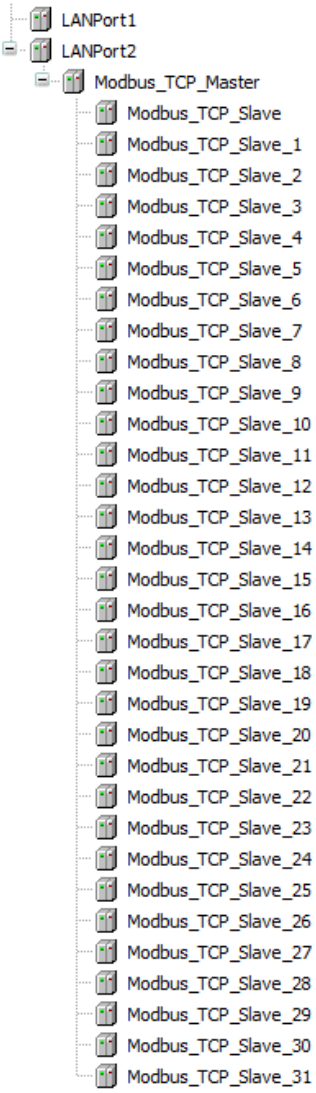
これで、スレーブ側 GM1 が登録されました。

コラム③　通信ポート数

LAN ポート 1 には通信ポートとして、16 コネクションの割り当てが、
LAN ポート 2 には通信ポートとして、32 コネクションの割り当てが可能です。
直前で追加した「Modbus TCP Slave」が 1 コネクション分に当たります。

LAN ポート 2 を例にすると、下図の様に「Modbus TCP Slave」を 32 個登録も可能です。
それぞれに異なる IP アドレスやポート番号を割り付けて異なるコネクションとして使用可能です。

33 個以上登録しようとした場合、コンテキストメニューの「デバイスの追加」がグレーアウトし追加はできません。



3－2　読み出し／書き込み設定

読み出し／書き込み設定ごとに、「Modbus TCP Slave」に「チャンネル」を追加していきます。

手順 1

「①Slave_Data0 の読み出し」設定を行います。

「Modbus TCP Slave」の「Modbus Slave　チャネル」タブを選択し、「チャネルの追加」をクリックします。

INFO

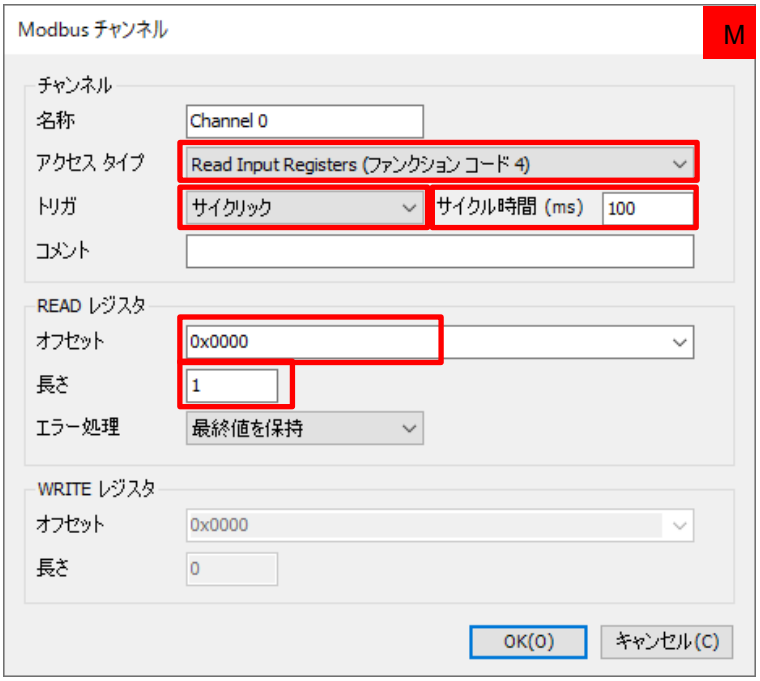
GM1 コントローラでは Modbus TCP　マスタ通信のコマンド送信開始に 3 つのトリガが用意されています。

サイクリック	「サイクル時間」に従って自動でコマンド送信を実行します。 スレーブ機器の値（経過値等）を周期的に読み出したい場合などに使用します。
立ち上がりエッジ	トリガとなるビットの立ち上がりでコマンド送信を実行します。 実行中や実行完了の各種フラグは用意されていません。
アプリケーション	「ModbusChannel 命令」の実行でコマンド送信を実行します。 命令ファンクションブロックの出力に「Busy」「Done」「Error」等の実行中や実行完了のフラグが用意されているためユーザプログラムでコマンドの実行状態を管理する場合はこの方法を使用します。

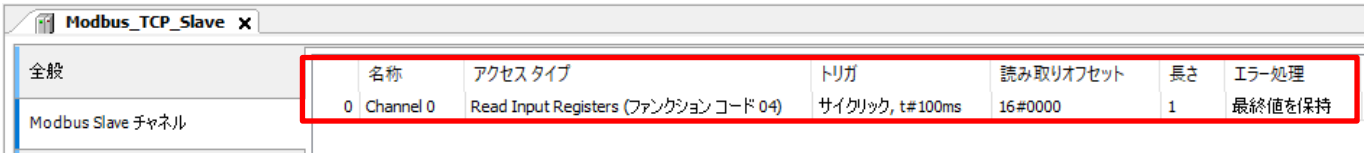
手順 2

「Modbus チャネル」の設定画面が開きますので、各項目は下記の様に設定します。
Modbus 入力レジスタの出力[0]を読み出します。
マスタからコマンドを 100ms 周期で自動送信する設定にします。

アクセスタイプ	Read Input Resisters（ファンクションコード 4）
トリガ	サイクリック
サイクル時間	100
オフセット	0×0000
長さ	1（Word）

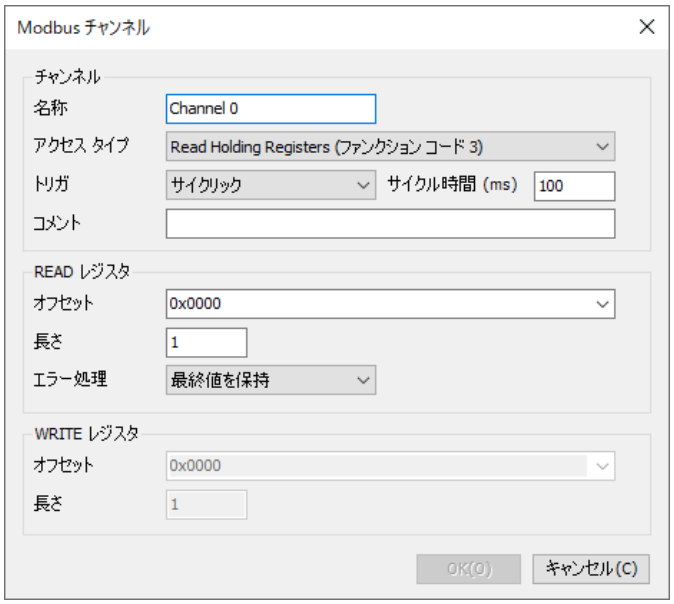


チャネルが追加されました。



コラム④ Modbus チャネル設定の説明

「Modbus チャンネル」では「アクセスタイプ（ファンクションコード）」「オフセット（送受信先アドレス）」「長さ（送受信データサイズ）」とコマンド送信タイミングを個別で設定していきます。
1 つのチャンネルにつき本設定は 1 つまでですが、1 つの Modbus TCP Slave の中にチャンネルは 100 個まで登録することができます。



チャンネル	
名称	各チャンネルの名称
アクセスタイプ	Modbus ファンクションコードの設定
トリガ	送信トリガ種別の選択
サイクル時間	トリガ条件が「サイクリック」の場合に有効
コメント	チャンネルに任意のコメントを記述

READ レジスタ	
オフセット	読み出す先の先頭アドレス
長さ	読み出すデータの長さ
エラー処理	エラー発生時の処理方法

WRITE レジスタ	
オフセット	書き込む先の先頭アドレス
長さ	書き込むデータの長さ

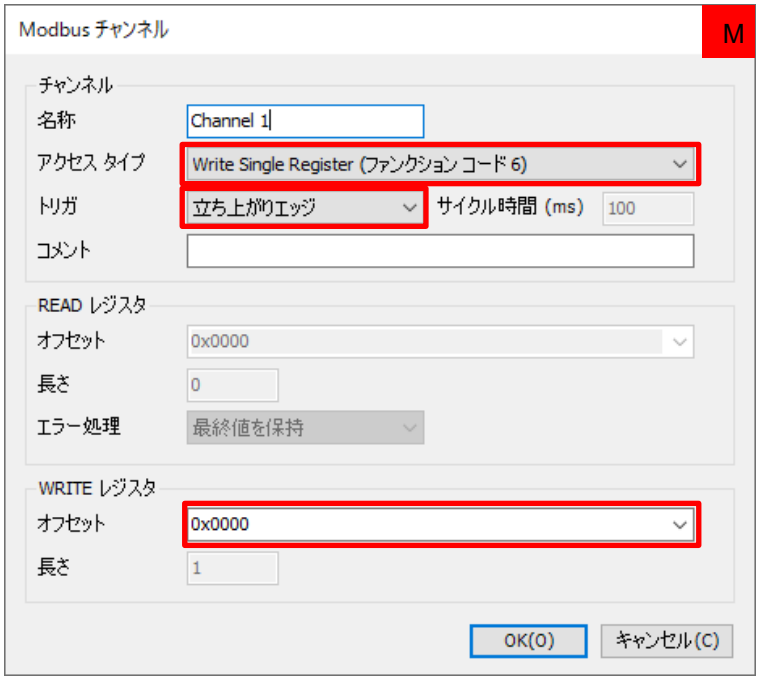
READ／WRITE レジスタは、アクセスタイプで選択したファンクションコードに依存していずれかの設定が有効になります。
上図は、ファンクションコード 03（保持レジスタ読み出し）のため READ が有効になっています。

手順３

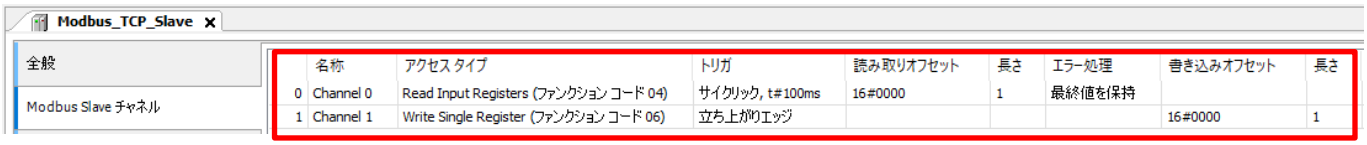
「②Slave_Data1 への書込み」設定を行います。

「チャネルの追加」をクリックし、各項目を下記の様に設定します。

アクセスタイプ	Write Single Resister（ファンクションコード 6）
トリガ	立ち上がりエッジ
オフセット	0×0000
長さ	1（ファンクションコード 6 は書き込み 1W のため、長さは 1 固定です。）



チャネルが追加されました。

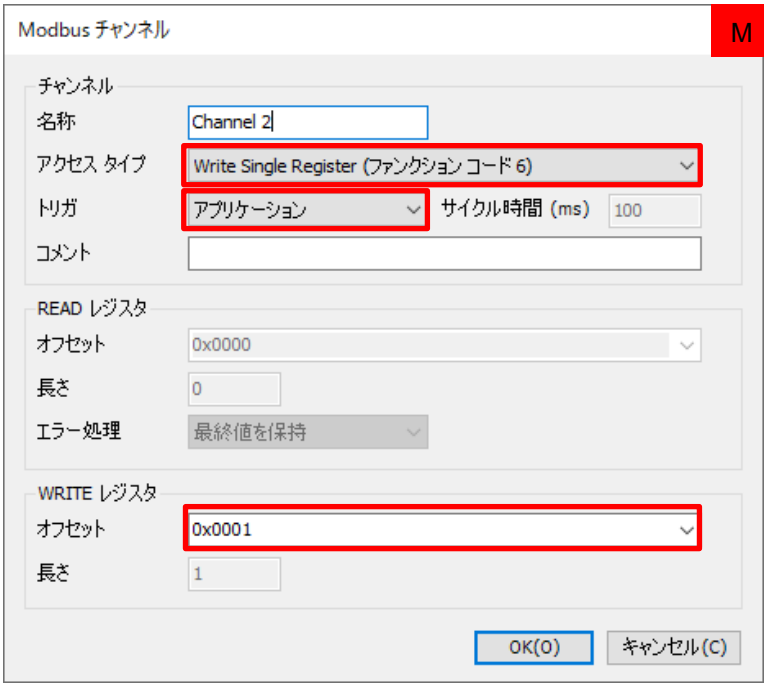


手順５

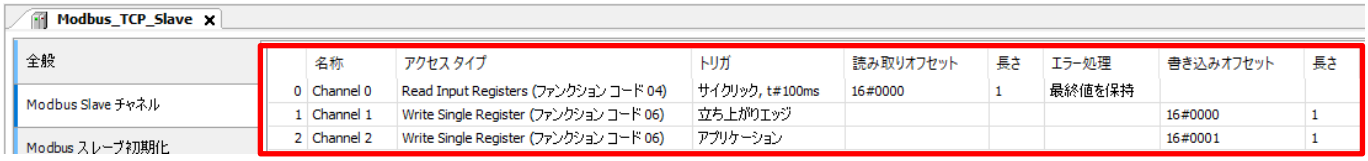
「③Slave_Data2 への書き込み」設定を行います。

「チャネルの追加」をクリックし、各項目を下記の様に設定します。

アクセスタイプ	Write Single Resister（ファンクションコード 6）
トリガ	アプリケーション
オフセット	0×0001
長さ	1（ファンクションコード 6 は書き込み 1W のため、長さは 1 固定です。）



チャネルが追加されました。



手順 6

設定した各チャネルのデータ領域に変数を割り当てます。

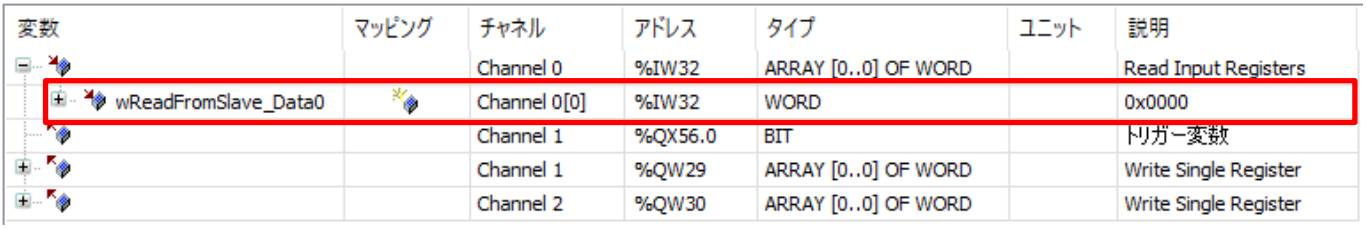
「Modbus TCP Slave I/O マッピング」タブを選択し、Channel0 左の ＋ をクリックします。

手順 7

Channel 0[0]の「変数」に使用変数の登録を行います。

「説明」列に読み出し先のオフセット値「0x0000」が記載されている場所が I/O マップ上での読み出し先になります。

Channel 0 は「Slave_Data0」に読み出しの設定ですので、「wReadFromSlave_Data0」と登録をします。



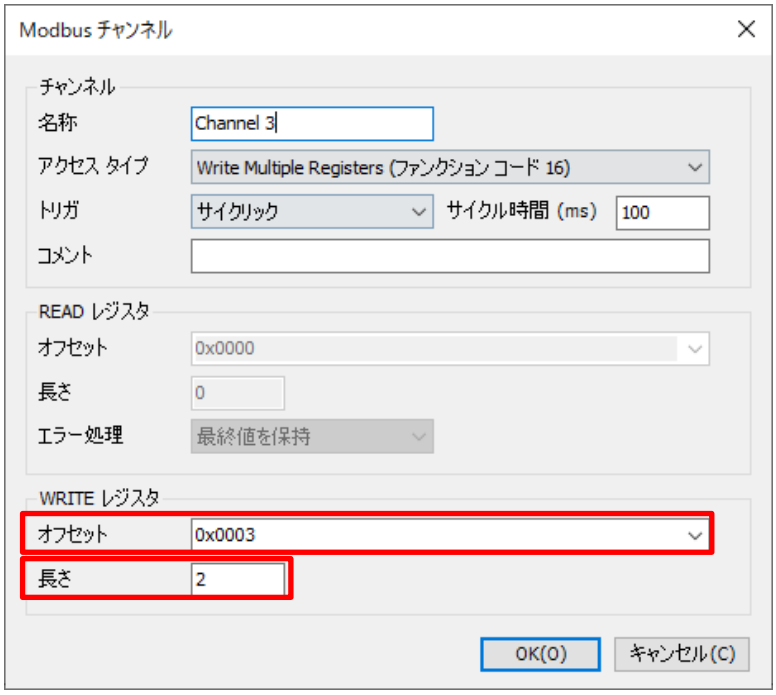
手順 8

Channel 1、Channel 2 左の ＋ をそれぞれクリックし、Channel1[0]と Channel2[0]の「変数」にも使用変数の登録を行います。

チャネル	変数
Channel1	xWriteTrigger
Channel1[0]	wWriteToSlave_Data1
Channel2[0]	wWriteToSlave_Data2

以上で、GM1 の Modbus マスタ通信の設定は完了です。

コラム⑤ Modbus チャンネルのオフセットと長さについて



オフセット「0×0003」なので、入力[3]になります。
長さ「2」なので入力[3]から 2WORD 分（入力[3]と入力[4]）になります。

変数	マッピング	チャネル	アドレス	タイプ	ユニット	説明
		入力	%IW32	ARRAY [0..9] OF WORD		Modbus 保持レジスタ
		入力[0]	%IW32	WORD		
		入力[1]	%IW33	WORD		
		入力[2]	%IW34	WORD		
		入力[3]	%IW35	WORD		
		入力[4]	%IW36	WORD		
		入力[5]	%IW37	WORD		
		入力[6]	%IW38	WORD		
		入力[7]	%IW39	WORD		
		入力[8]	%IW40	WORD		
		入力[9]	%IW41	WORD		

2WORD 以上の変数を使用する場合、構造体で、扱えるワード長を持つ変数を宣言することで扱うことができます。

DUT

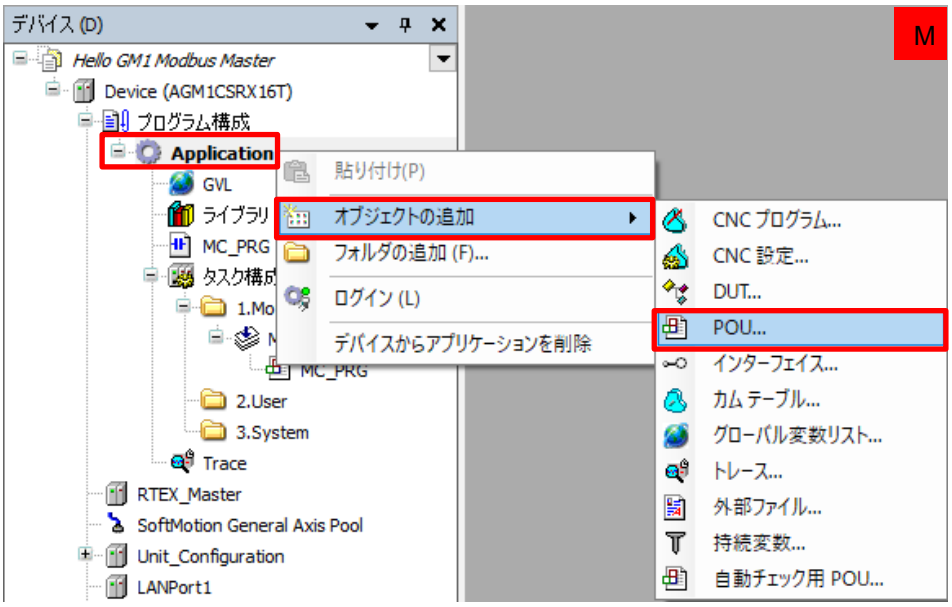
```
1  TYPE DUT :
2  STRUCT
3      wTest0  : WORD;
4      wTest1  : WORD;
5      wTest2  : WORD;
6      dwTest3 : DWORD;
7      awTest  : ARRAY[0..4] OF WORD;
8  END_STRUCT
9  END_TYPE
10
```

3－3　書き込みプログラム作成(トリガ：立ち上がりエッジ)

スレーブ GM1 の「Slave_Data0」から読み出した値に対して、+1000 した値を「Slave_Data1」に書き込むプログラムを作成します。

手順 1

「Application」を右クリックして、**オブジェクトの追加→POU** を選択して新規 POU を追加します。



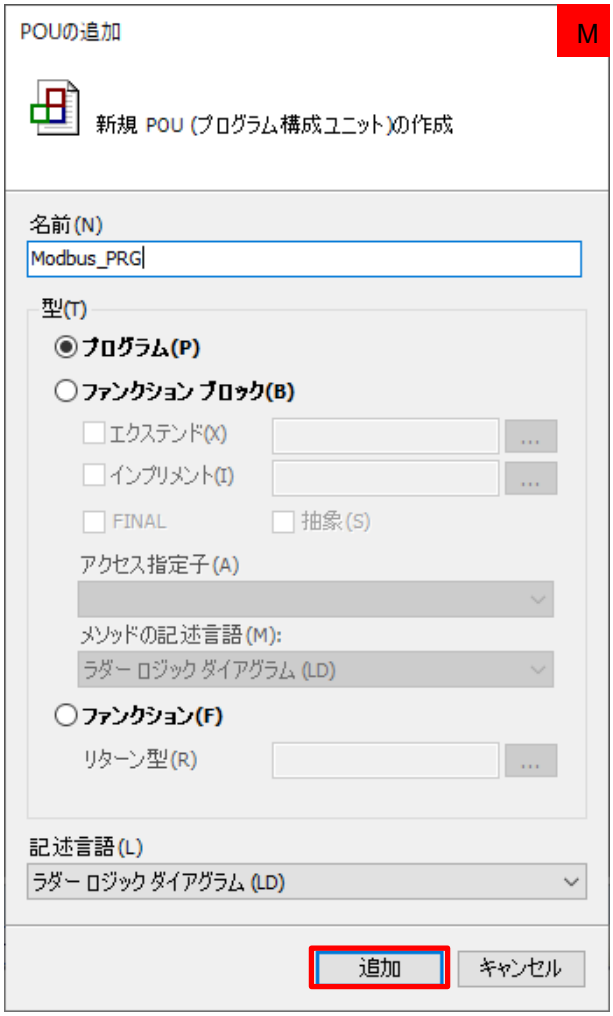
手順２

「POU の追加」ダイアログで、以下のように設定し「追加」をクリックします。

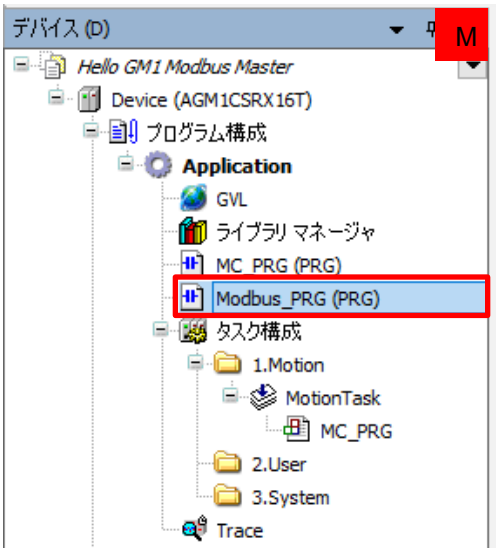
名前(N)　　:Modbus_PRG

型(T)　　:プログラム(P)

記述言語(L)　:ラダーロジックダイアグラム(LD)

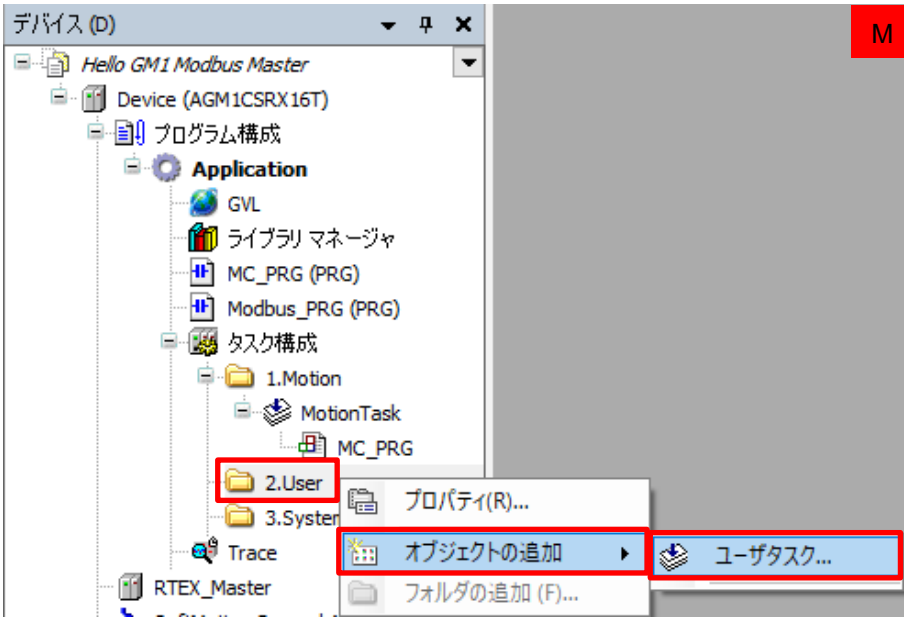


「Application」に「Modbus_PRG」が追加されました。



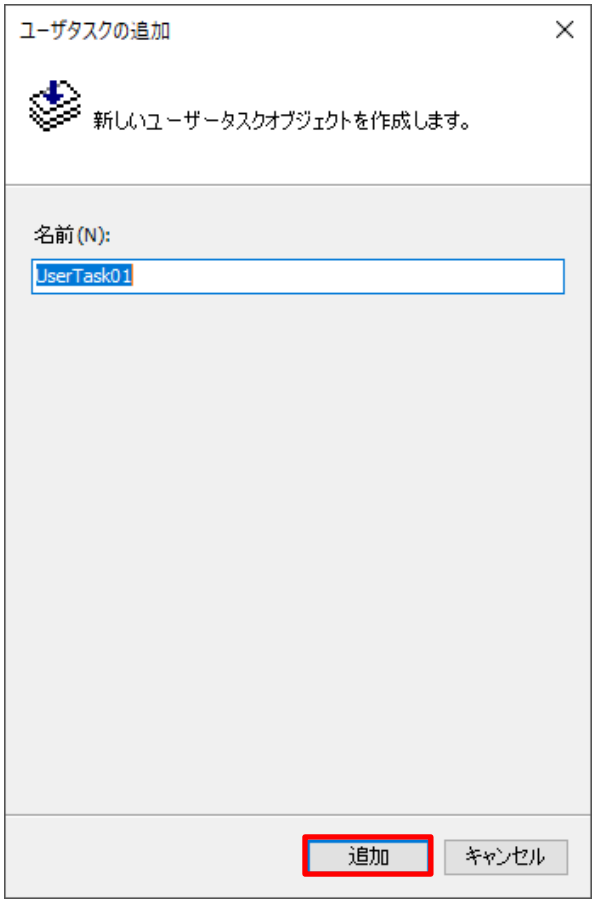
手順３

「2.User」で右クリックし、「オブジェクトの追加」ー「ユーザタスク」をクリックします。

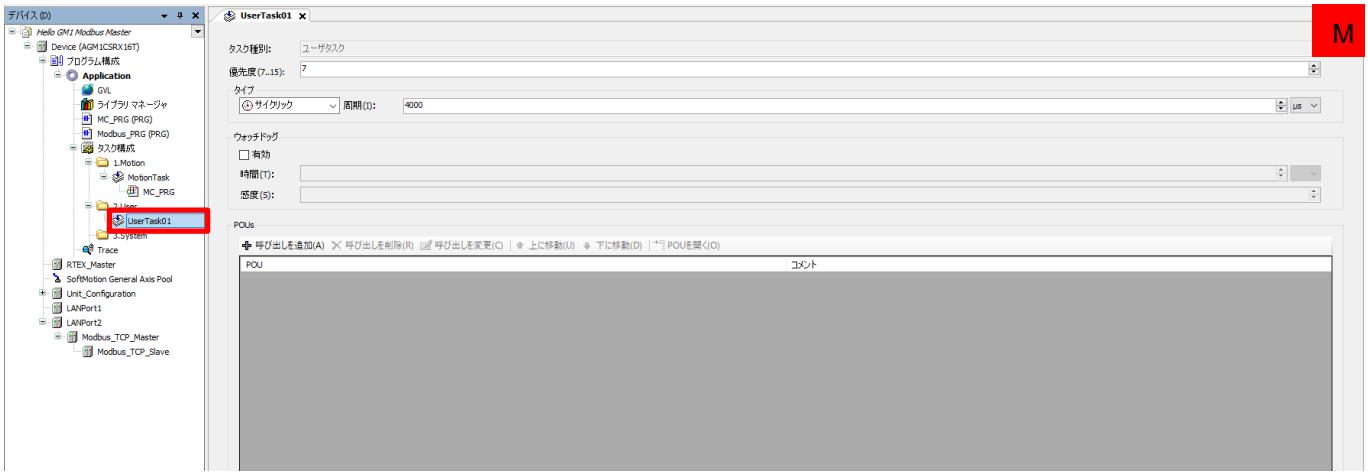


手順４

「ユーザタスクの追加」ダイアログが表示されますので、名前:UserTask01（初期値）で「追加」をクリックします。

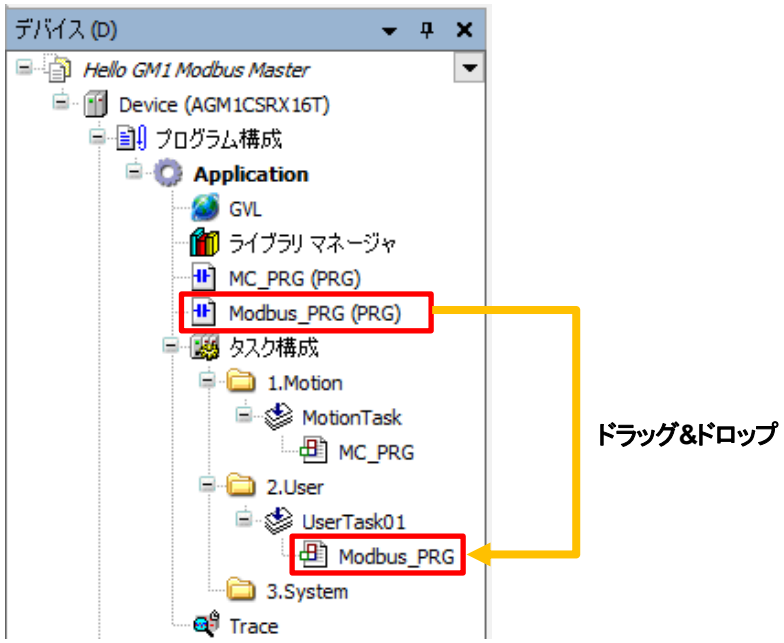


「UserTask01」が追加されました。



手順５

追加された「UserTask01」に「Modbus_PRG」をドラッグ＆ドロップし、タスクに追加します。



コラム⑥　タスクについて

タスク	内容
モーションタスク	モーション制御を行うためのユーザプログラム用タスクです。 優先度が最も高いです。プロジェクトに対してモーションタスクは、1 つのみとなります。
ユーザタスク	モーション制御以外の制御を行うためのユーザプログラム用タスクです。 ユーザが優先度を設定できます。1 つのプロジェクト内に最大 50 のタスクを登録することができます。
システムタスク	システムが使用するタスクで、ユーザプログラムは追加できないタスクです。 他のタスクの空き時間に処理されます。

手順 6

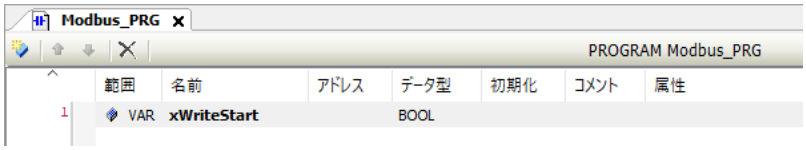
「Modbus_PRG」をダブルクリックしてプログラム画面を開きます。

手順 7

ローカル変数を一つ追加します。「名前」欄を選択した状態で、下記変数を登録してください。

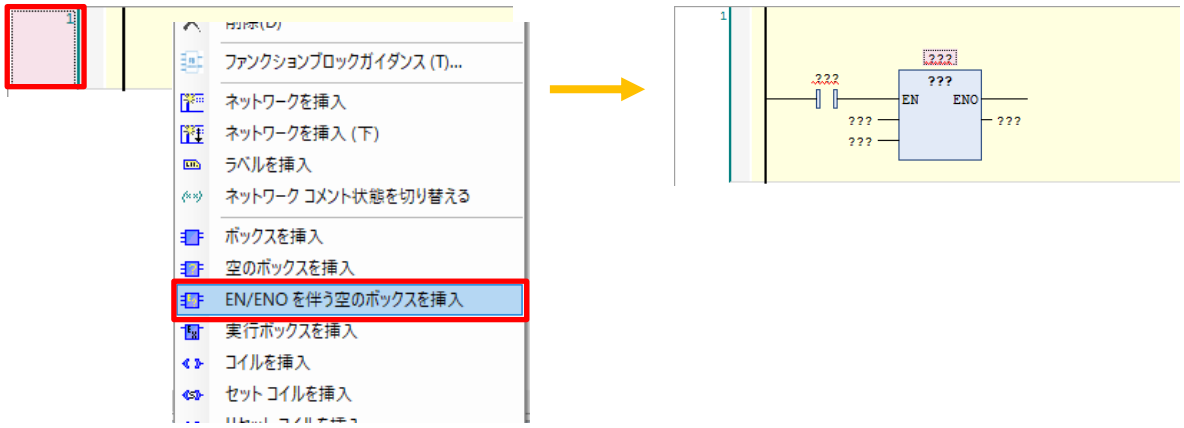
名前　　:xWriteStart

データ型　:BOOL



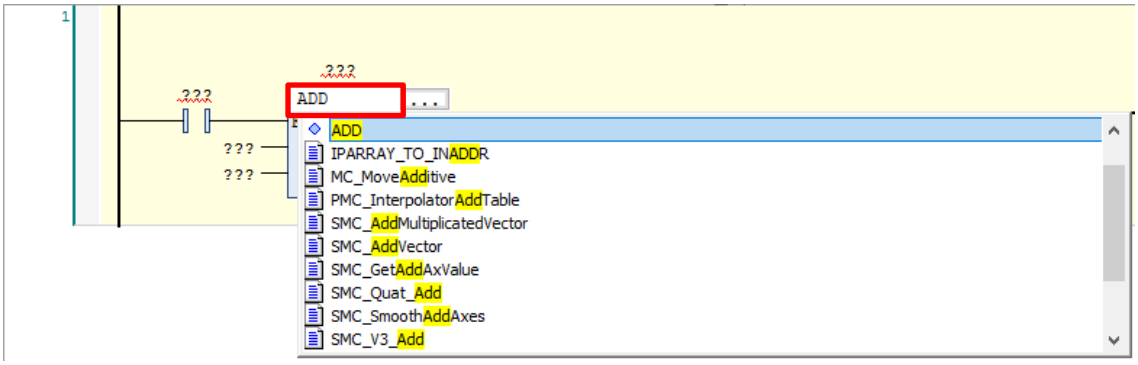
手順 8

ネットワークの一番左（下図の赤い箇所）で右クリックし、「EN/ENO を伴う空のボックスを挿入」を選択します。



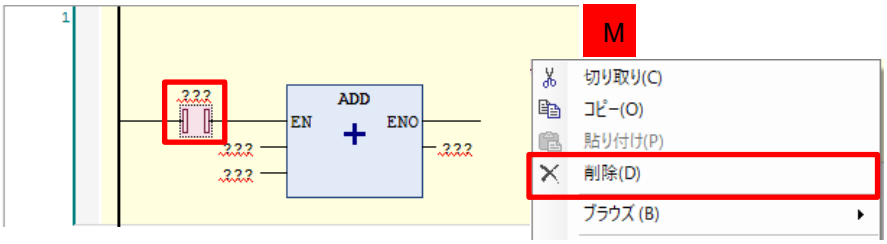
手順 9

ボックス内の「???」をクリックし、「ADD」と入力して Enter キーを押します。

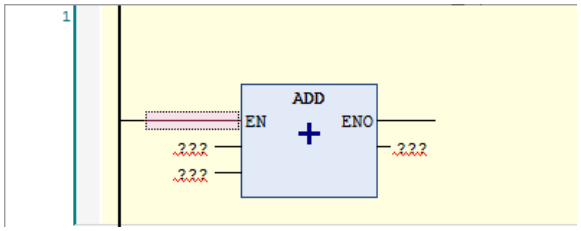


手順 10

入力された ADD 命令の A 接点を選択して右クリックし、「削除」を選択します。

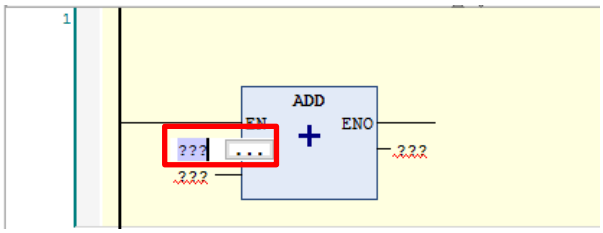


入力側の接点が削除されましたので、ADD 命令は常時実行状態になります。



手順 11

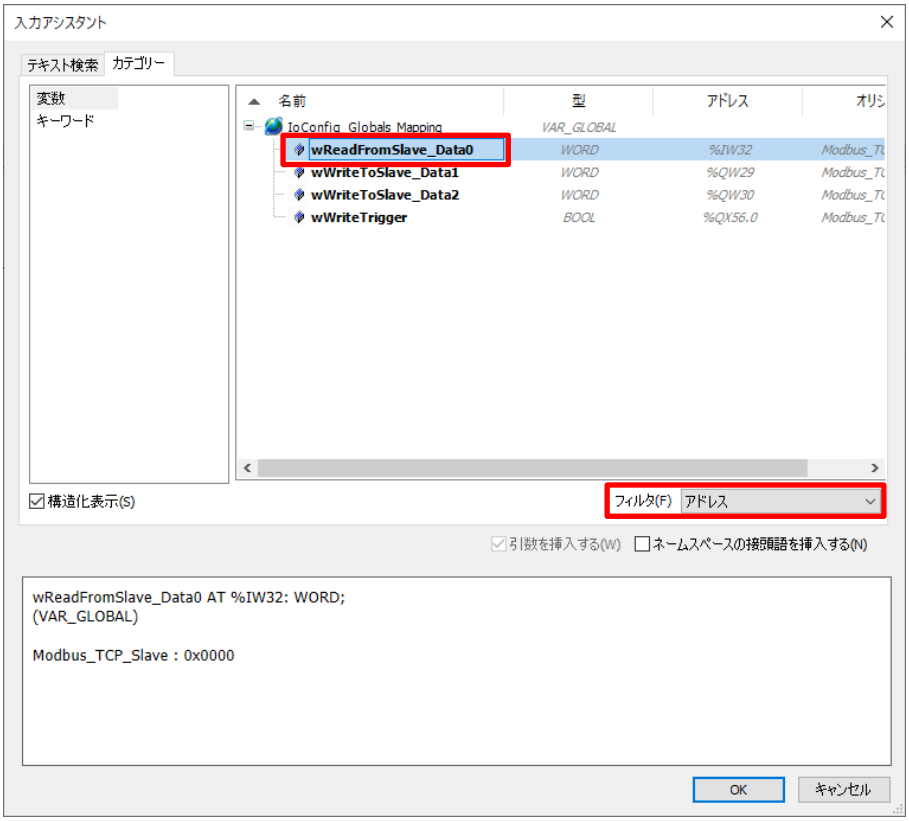
入力上側の「???」をクリックし、 ... をクリックして、入力アシスタントを開きます。



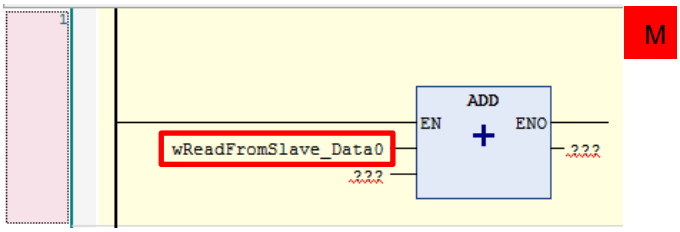
手順 12

「IoConfig_Globals_Mapping」ー「wReadFromSlave_Data0」を選択して、「OK」をクリックします。

フィルタで「アドレス」を選択することで、P.36 で「Modbus TCP Slave I/O マッピング」に登録した変数のみが表示されます。

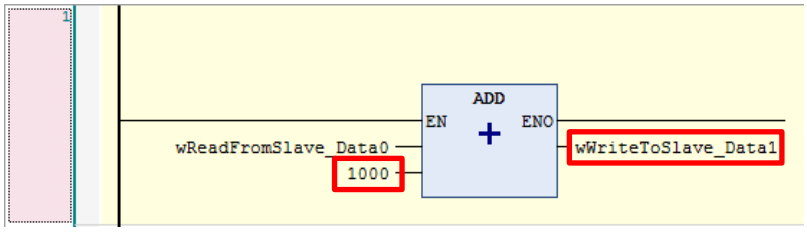


「wReadFromSlave_Data0」が挿入されました。



手順 13

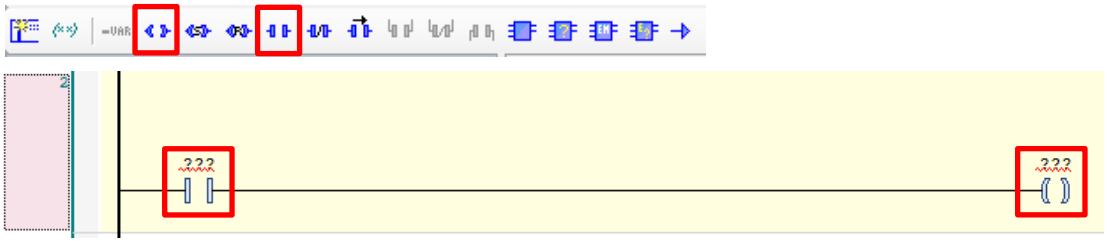
同様に、出力の「???」に入力アシスタントから「wWriteToSlave_Data1」を挿入します。
入力下側の「???」には固定値で「1000」を入力します。



スレーブの GM1 から読み取った数値にオフセット「1000」を行う簡易的な演算になります。

手順 14

新しくネットワークを挿入し、ツールバーから接点／コイルを挿入します。



手順 15

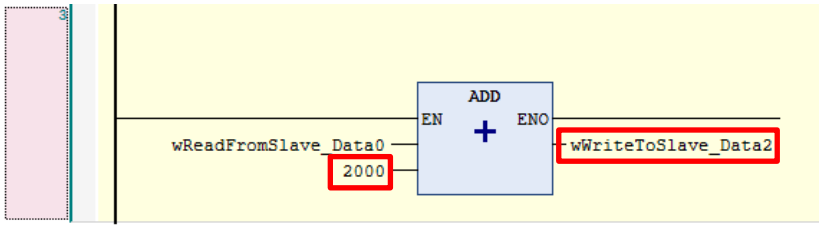
入力アシスタントから下記変数をそれぞれ選択します。
接点：wWriteStart
コイル：wWriteTrigger

3－4　書き込みプログラム作成（トリガ：アプリケーション）

まず、スレーブの GM1 から読み出した「wReadFromSlave_Data0」に＋2,000 するプログラムを作成します。

手順 1

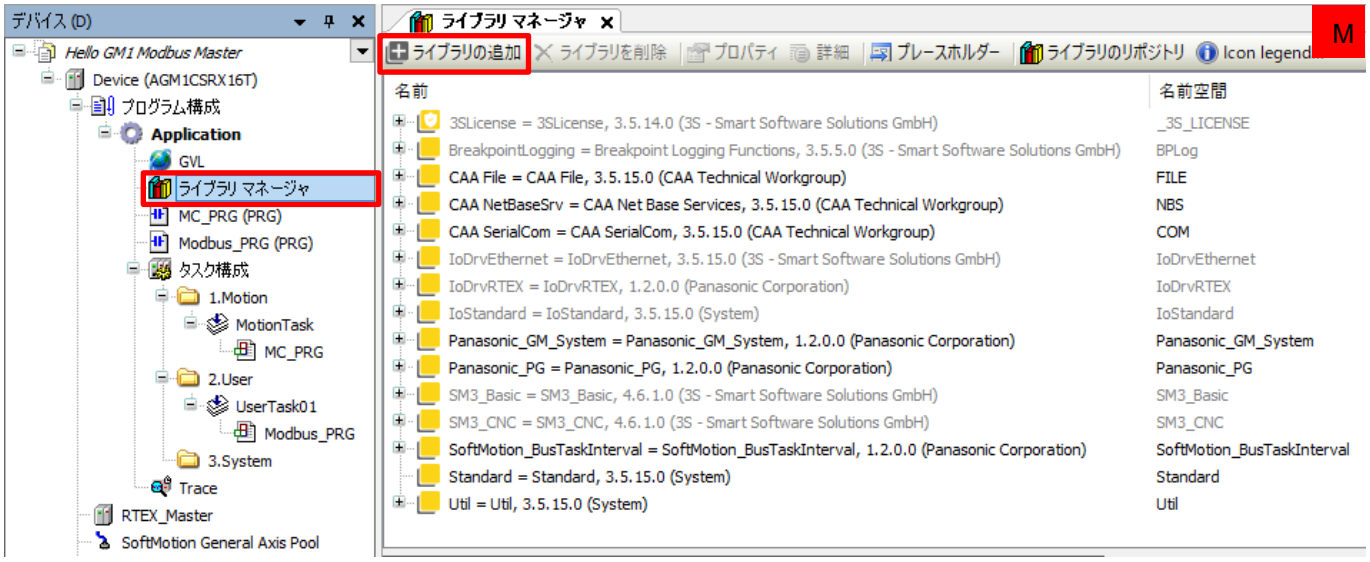
新しくネットワークを挿入し、下図のように ADD を追加します。



次にトリガ：アプリケーションでは「ModbusChannel」命令を使用します。

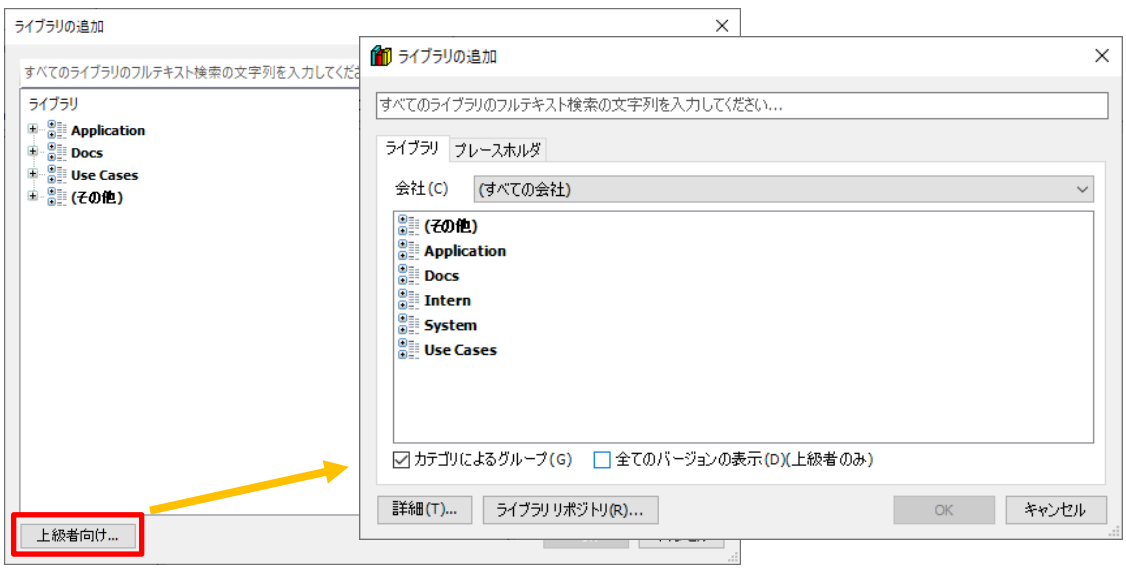
手順 1

ナビゲータウィンドウの「ライブラリマネージャ」をダブルクリックします。
表示された設定画面で「ライブラリの追加」をクリックします。



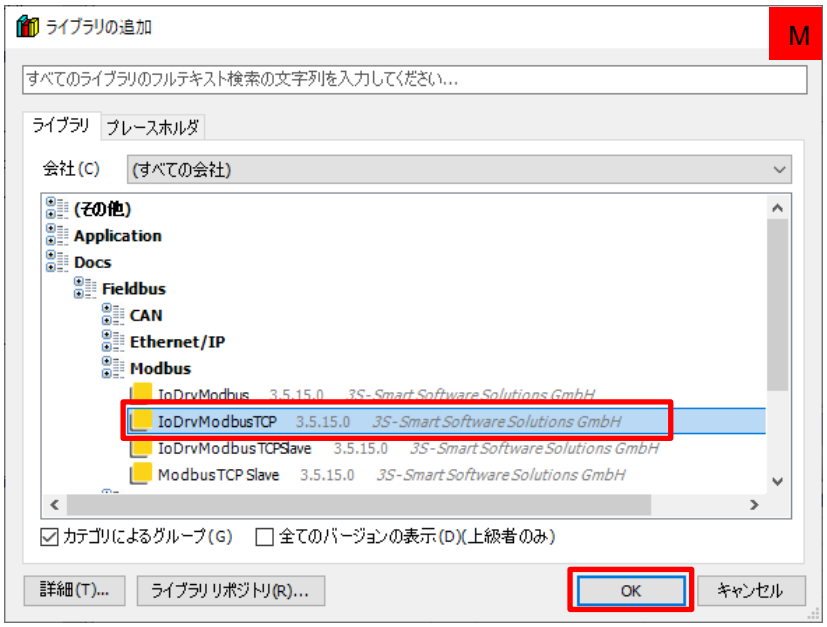
手順 2

表示された「ライブラリの追加」ダイアログの「上級者向け」をクリックし、ダイアログの表記を切り替えます。



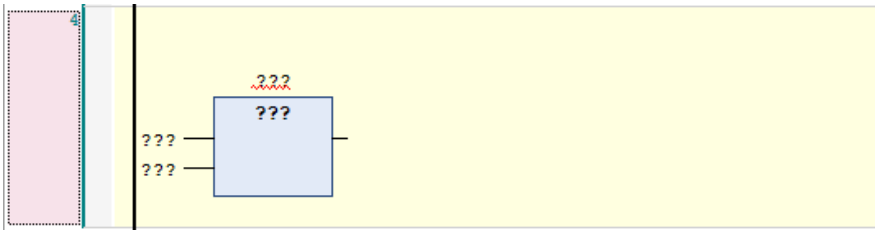
手順３

ダイアログ内のツリー**Docs→Fieldbus→Modbus→IoDrvModbusTCP** を選択し、「OK」をクリックします。



手順４

新しくネットワークを挿入し、右クリックで「空のボックスを挿入」を選択します。

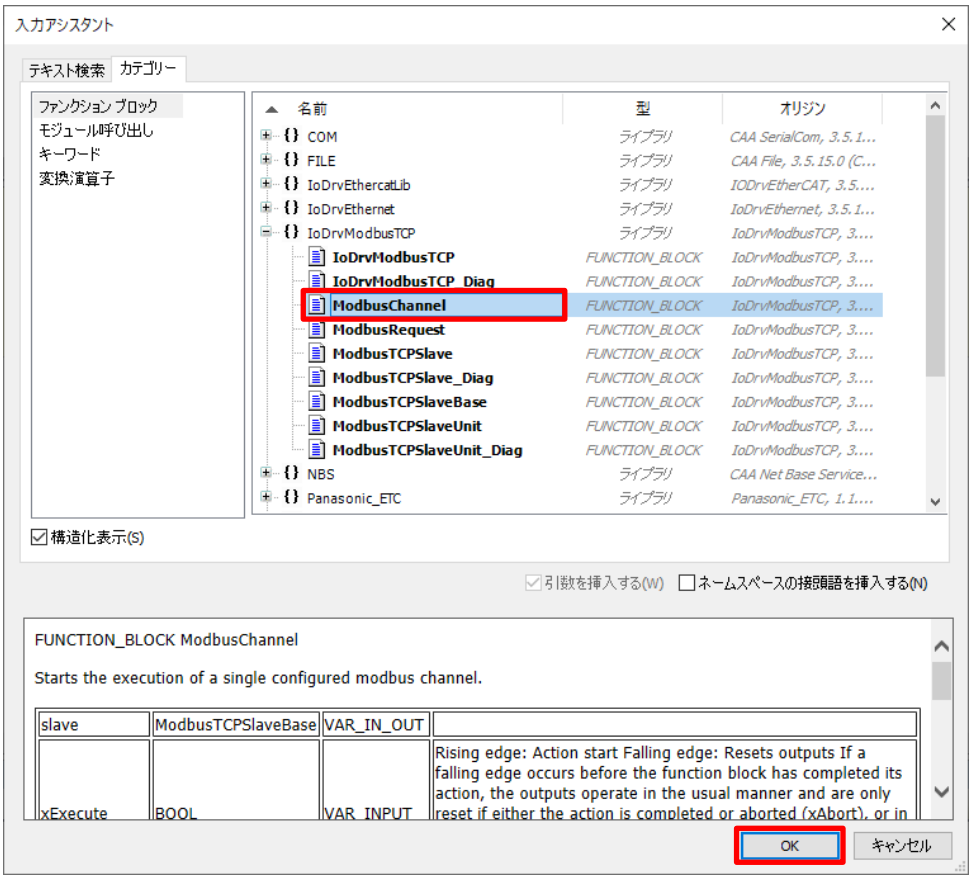


手順５

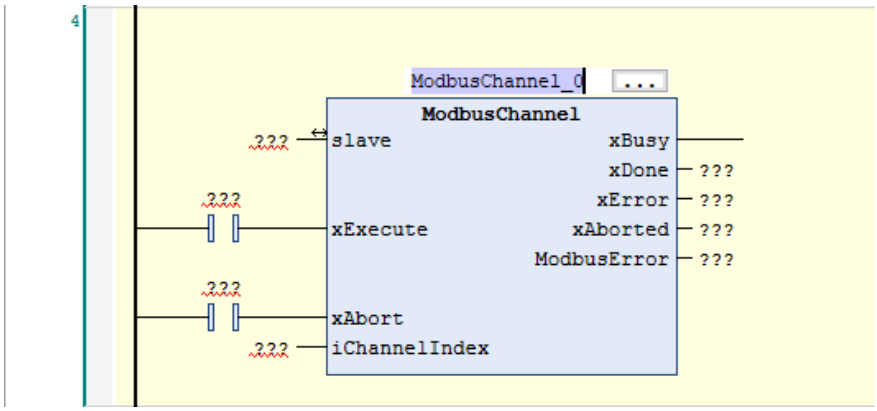
ボックスの「???」から「入力アシスタント」を表示します。

手順６

IoDrvModbusTCP→ModbusChannel を選択し、「OK」をクリックします。

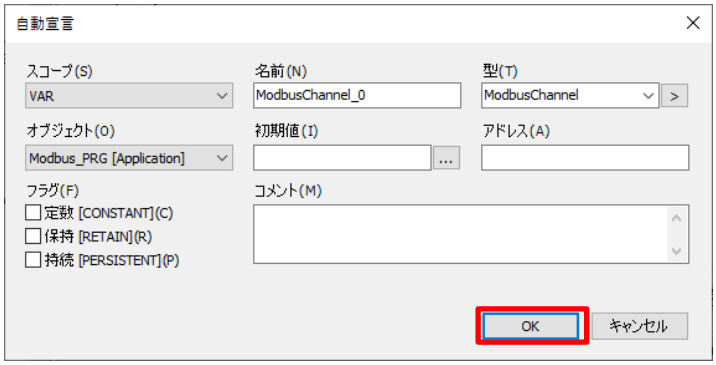


ネットワーク４にファンクションブロック「ModbusChannel」が挿入されます。

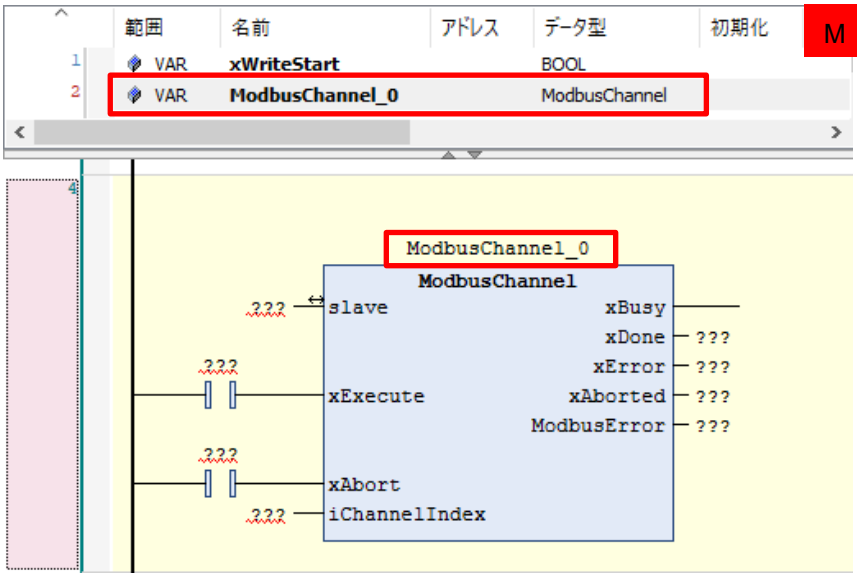


手順 7

そのまま PC の Enter キーを押すと、「自動宣言」ダイアログが表示されます。内容を確認して「OK」をクリックしてください。



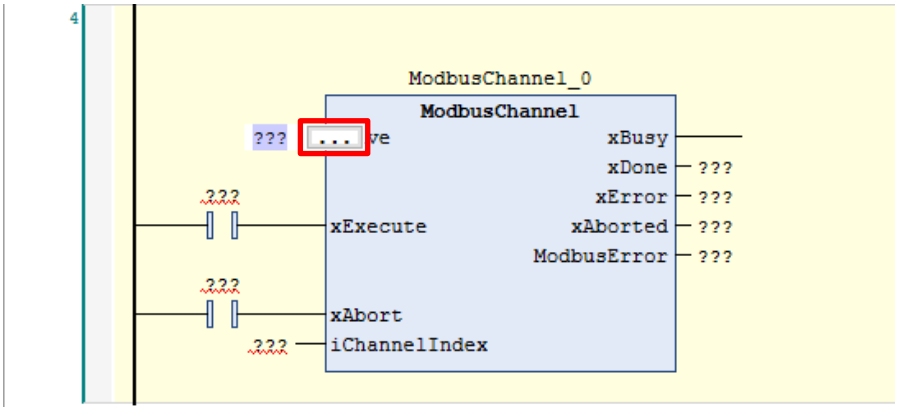
変数エディタに「ModbusChannnel_0」が自動挿入されました。



手順 8

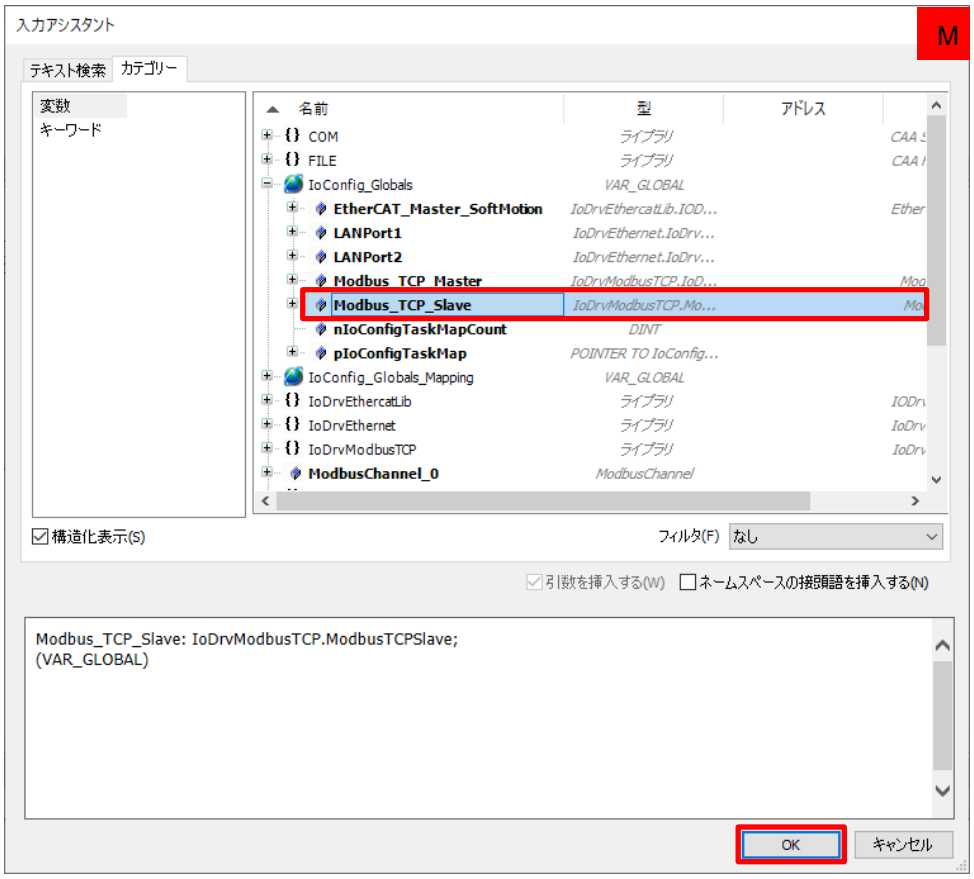
「ModbusChannel」命令の入出力を設定します。

入力「Slave」の「???」をクリックし、 ... をクリックします。

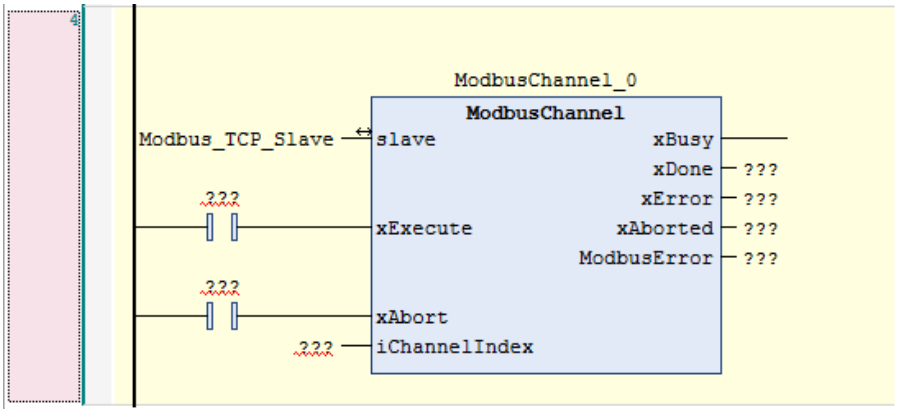


手順 9

入力アシスタントが表示されます。IoConfig_Globals→Modbus_TCP_Slave を選択し「OK」をクリックします。

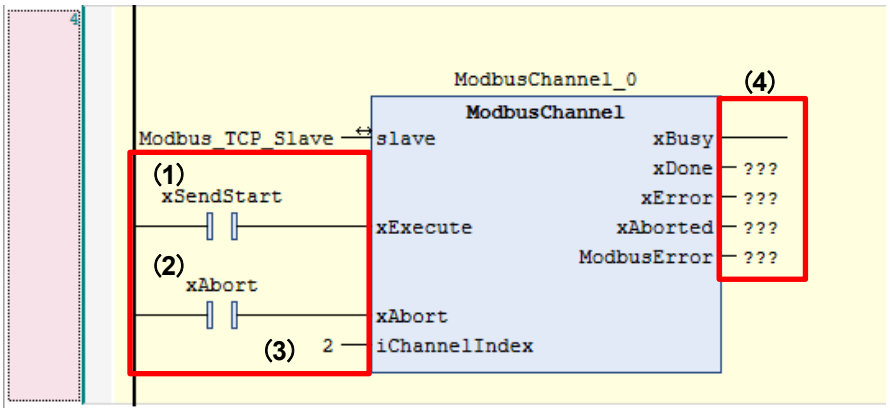


入力「slave」に「Modbus_TCP_Slave」が挿入されました。



手順 10

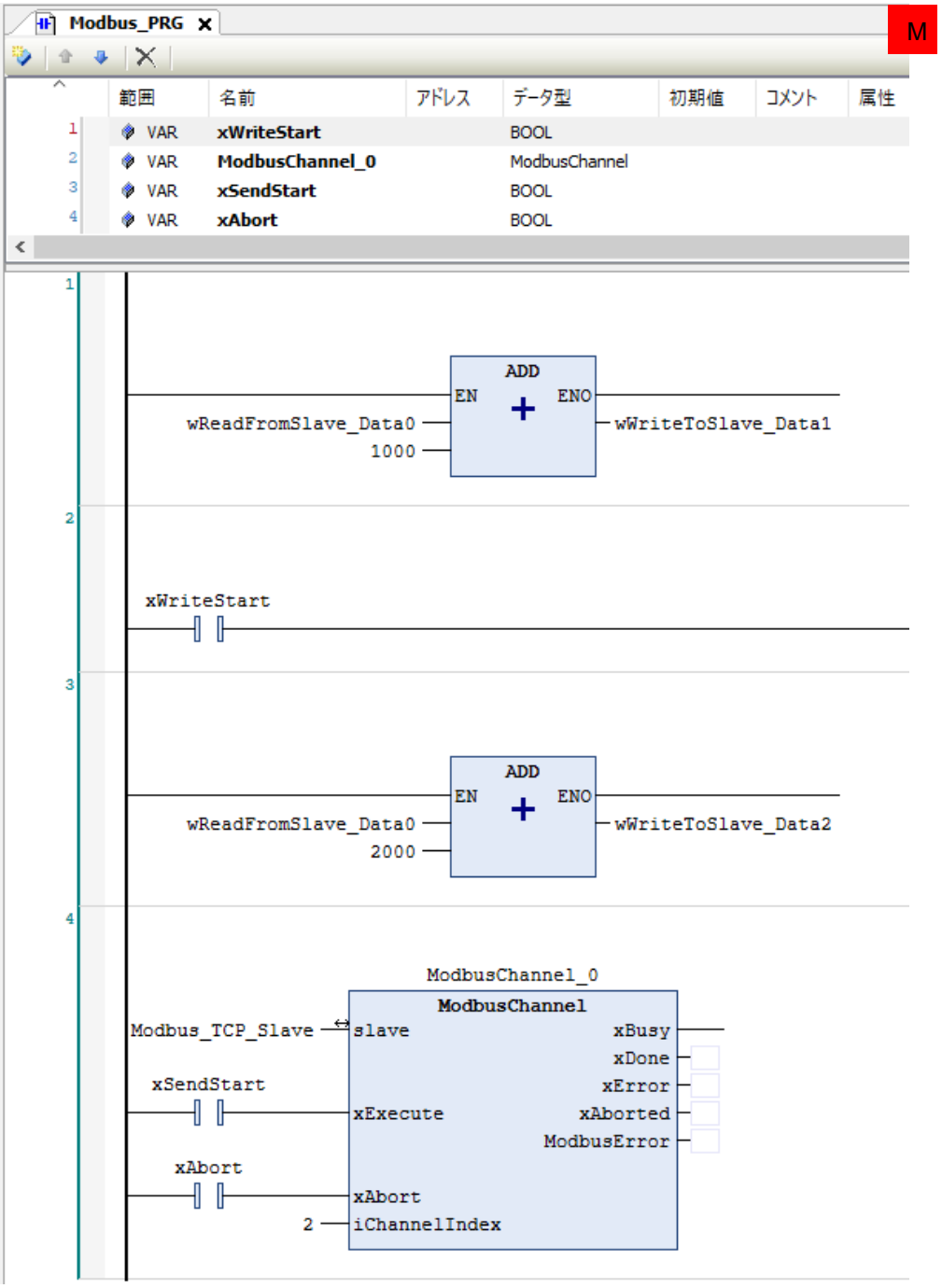
その他、下図のようにファンクションブロックを完成させます。



	種類	引数名	設定値	内容
(1)	入力	xExecute	xSendStart	立ち上がりエッジでコマンド送信開始
(2)		xAbort	xAbort	TRUE:実行を停止し、すべての出力をカット
(3)		iChannelIndex	2	送信するコマンドがセットされたチャネル番号
(4)	出力	xBusy	???を削除	TRUE:FB の処理が未完了
		xDone	???を削除	TRUE:処理完了
		xError	???を削除	TRUE:FB 内でエラー発生
		xAborted	???を削除	TRUE:ユーザの xAbort 入力により実行停止
		ModbusError	???を削除	エラーコード出力

手順 11

以上で、Modbus マスタ通信の設定およびプログラム作成は完了です。
一度、プロジェクトデータを上書き保存してください。
GM Programmer 上でビルドを行い、エラーが発生していないことを確認してください。
GM1 コントローラにダウンロードを行い、運転モードを「RUN」にしてください。



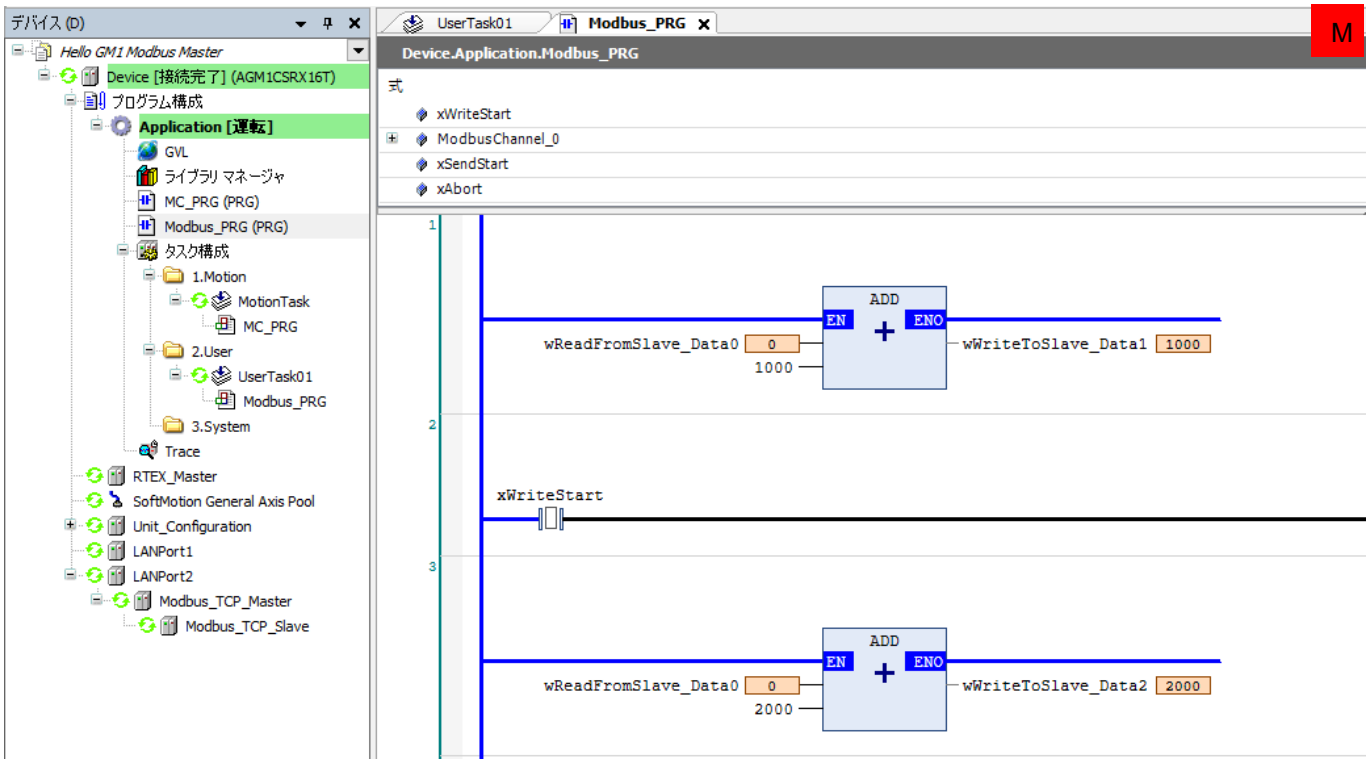
4 通信動作の確認

手順 1

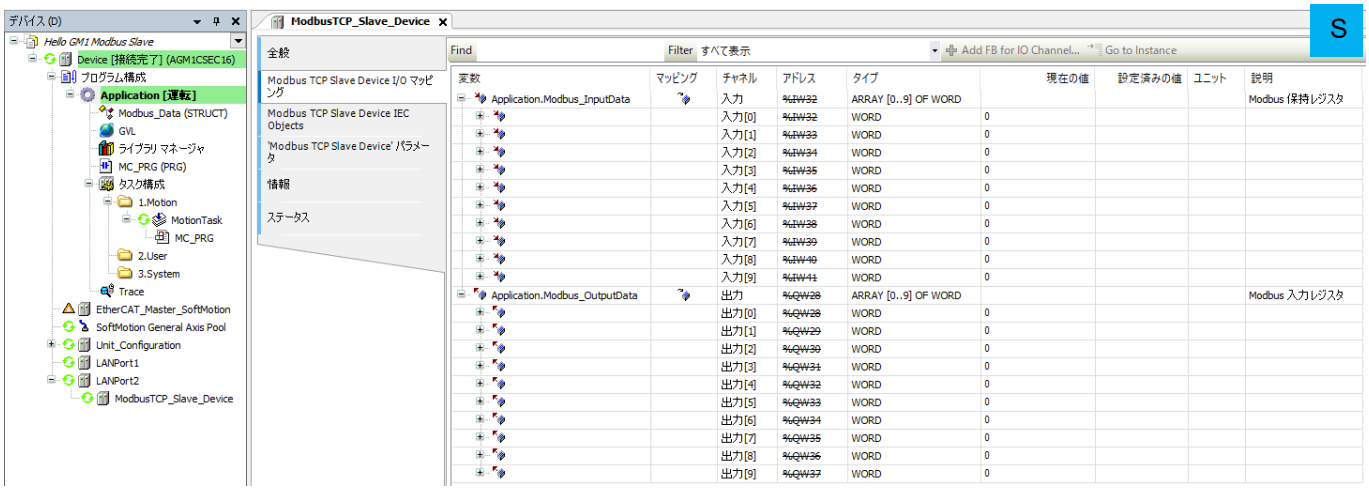
マスタ側の GM1 コントローラとスレーブ側の GM1 コントローラが共に「RUN」になっており、
2 つの GM Programmer が「ログイン」になっていることを確認してください。

マスタ側の GM Programmer で「POU：Modbus_PRG」を開き、
スレーブ側の GM Programmer で「ModbusTCP_Slave_Device：ModbusTCP_Slave_Device I/O マッピング」を開きます。

マスタ側 GM Programmer

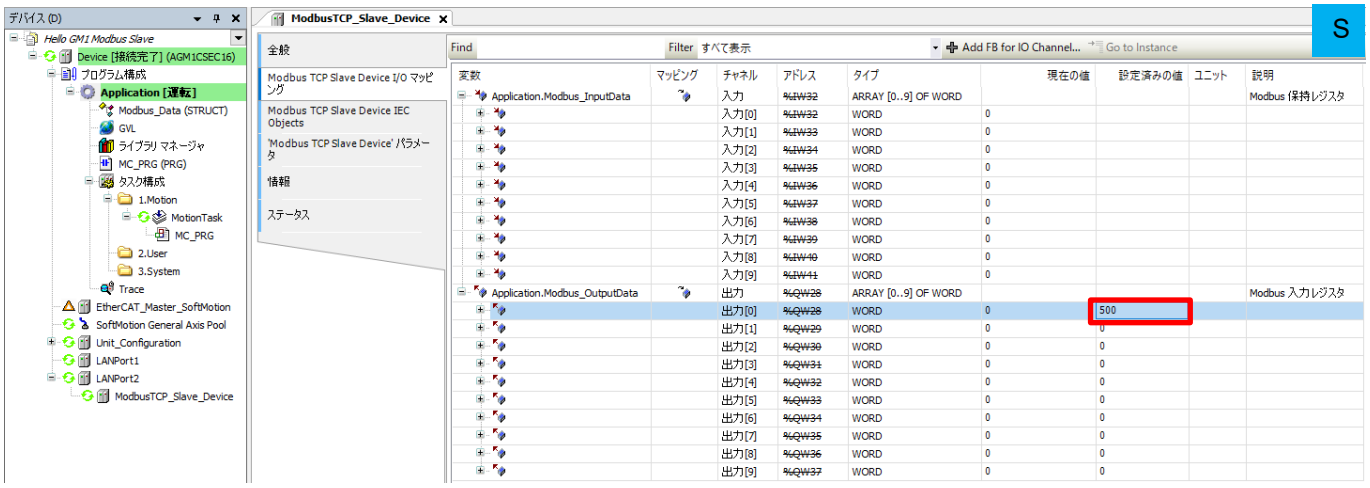


スレーブ側 GM Programmer



手順 2

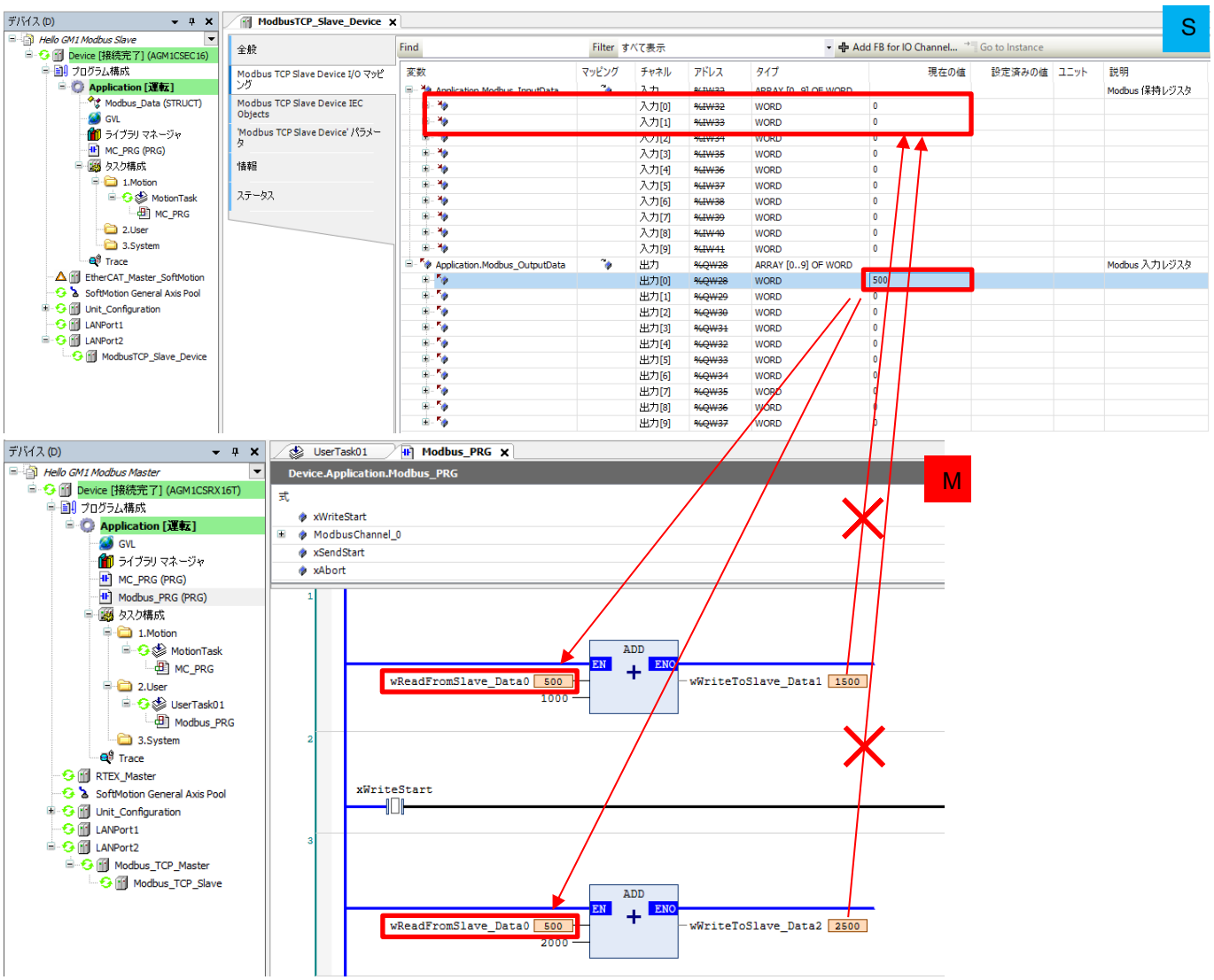
スレーブ側 GM Programmer の「ModbusTCP_Slave_DeviceI/O マッピング」で「出力[0]」の「設定済みの値」欄に、「500」を入力し Ctr+F7 を実行します。



手順 3

「現在の値」に書き込まれるのと同時に、スレーブ側 GM1 の「出力[0]」の値がマスタ側 GM1 の「wReadFromSlave_Data0」に読み出されています。

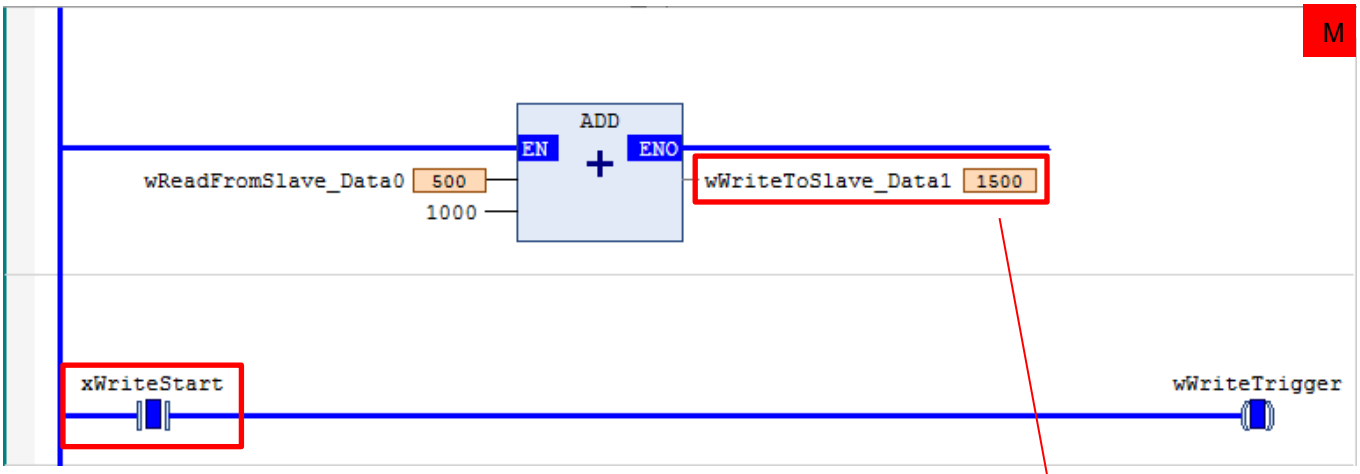
しかし、入力[0]と入力[1]には「wWriteToSlave_Data1」と「wWriteToSlave_Data2」の値は書き込まれていません。



Slave_Data0 の読み出しは「サイクリック」に設定されており、「サイクル時間:100ms」の周期で自動読み出しを行っていますが書き込みは「立ち上がりエッジ」と「アプリケーション」に設定されているため、書き込みは行われていません。

手順 4

「xWriteStart」を TRUE にすることで「xWriteTrigger」が TRUE(立ち上がり)になり Slave_Data1 への書き込み条件が成立し、「wWriteToSlave_Data1」の値が書き込まれます。

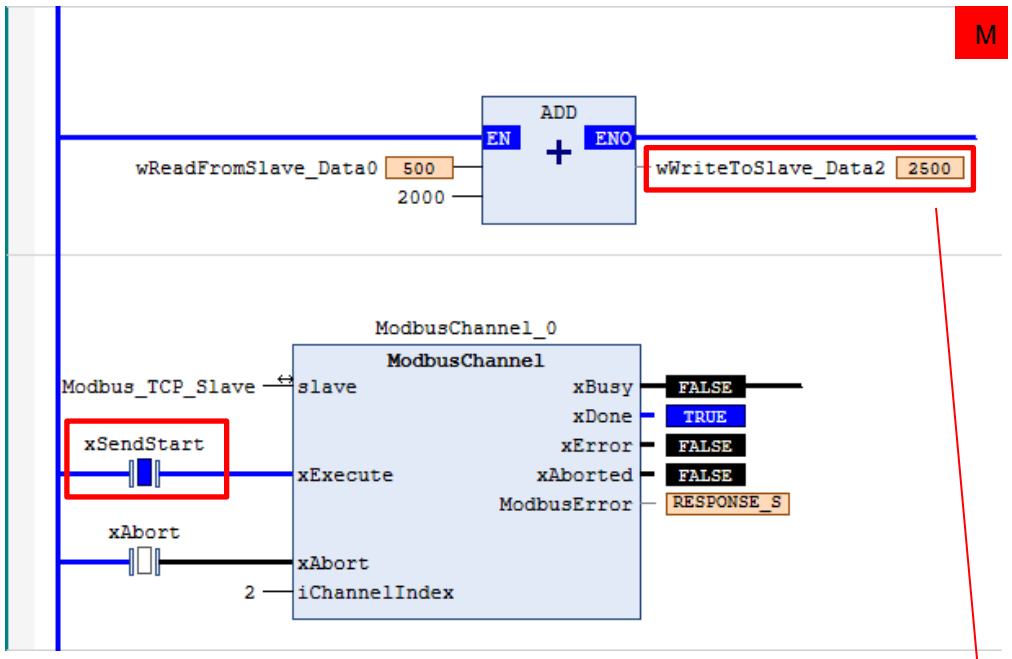


変数	マッピング	チャネル	アドレス	タイプ	現在の値
Application.Modbus_InputData		入力	%IW32	ARRAY [0..9] OF WORD	
		入力[0]	%IW32	WORD	1500
		入力[1]	%IW33	WORD	0

S

手順５

「xSendStart」を TRUE にすることで「ModbusChannel_0」が実行され、
Slave_Data2 へ「wWriteToSlave_Data2」の値が書き込まれます。



S

変数	マッピング	チャネル	アドレス	タイプ	現在の値
Application.Modbus_InputData		入力	%IW32	ARRAY [0..9] OF WORD	
		入力[0]	%IW32	WORD	1500
		入力[1]	%IW33	WORD	2500

INFO

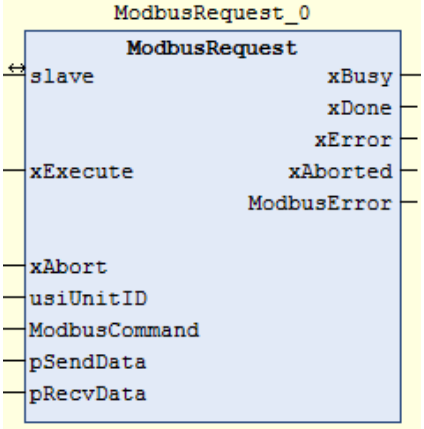
「ModbusChannnel」命令は出力に「xBusy」「xDone」を備えています。
命令の実行中は「xBusy」が TRUE となります。
命令の実行が完了すると「xBusy」は FALSE となり、「xDone」が TRUE となります。

「ModbusChannnel」命令も入力「xExecute」の立ち上がりエッジで実行されるため、
再度書き込みを行う場合は、「xSendStart」を FALSE にし、再度 TRUE にします。

コラム⑦ ModbusRequest ファンクションブロック

Modbus_TCP_Slave デバイスを使用せずに、I/O で指定した Modbus コマンドを処理するファンクションブロック

対応コマンド	
コマンド 1	複数点コイル状態読み出し
コマンド 2	複数点入力状態読み出し
コマンド 3	複数点保持レジスタ読み出し
コマンド 4	複数点入力レジスタ読み出し
コマンド 5	単点コイル書き込み
コマンド 6	単点保持レジスタ書き込み
コマンド 15	複数点コイル書き込み
コマンド 16	複数点保持レジスタ書き込み
コマンド 23	複数点保持レジスタ読み出し／書き込み



種類	引数名	型	内容
入力	Slave	ModbusTCPSlave	Modbus_TCP_Slave デバイスのハンドル
	xExecute	BOOL	立ち上がりエッジでコマンド送信開始
	xAbort	BOOL	TRUE:実行を停止し、すべての出力をリセット
	usiUnitID	USINT	スレーブアドレス 1～247
	ModbusCommand	ModbusCommand	発行するコマンドのパラメータを格納した構造体
	pSendData	POINTER TO BYTE	送信データバッファへのポインター
	pRecvData	POINTER TO BYTE	受信データバッファへのポインター
出力	xBusy	BOOL	TRUE:FB の処理が未完了
	xDone	BOOL	TRUE:処理完了
	xError	BOOL	TRUE:FB 内でエラー発生
	xAborted	BOOL	TRUE:ユーザの xAbort 入力により実行停止
	ModbusError	BYTE	エラーコード出力

EtherNet 通信　EtherNet/IP

1 基本設定

1－1 動作イメージ

EtherNet/IP
2 台の GM1 コントローラをスキャナ／アダプタで使用します。
お互いの LAN Port2 を使用して、EtherNet/IP 通信を行います。

※スキャナとアダプタで 2 つの GM Programmer を立ち上げるため、分かりやすく見分けるために、
GM Programmer の画像右上に S と A を付けています。
S :スキャナ
A :アダプタ

スキャナの Output をアダプタの Input へ
アダプタの Output をスキャナの Input へ

ウォッチ 1 — S

式	アプリケーション	タイプ	値	設定済みの値	実行点	アドレス
IoConfig_Globals_Mapping.wEIPScInput01	Device.Application	WORD	10		サイクリックモニタリング	%IW32
IoConfig_Globals_Mapping.wEIPScInput02	Device.Application	WORD	11		サイクリックモニタリング	%IW33
IoConfig_Globals_Mapping.dwEIPScInput03	Device.Application	DWORD	12		サイクリックモニタリング	%ID17
IoConfig_Globals_Mapping.wEIPScOutput01	Device.Application	WORD	1		サイクリックモニタリング	%QW28
IoConfig_Globals_Mapping.wEIPScOutput02	Device.Application	WORD	2		サイクリックモニタリング	%QW29
IoConfig_Globals_Mapping.dwEIPScOutput03	Device.Application	DWORD	3		サイクリックモニタリング	%QD15

ウォッチ 1 — A

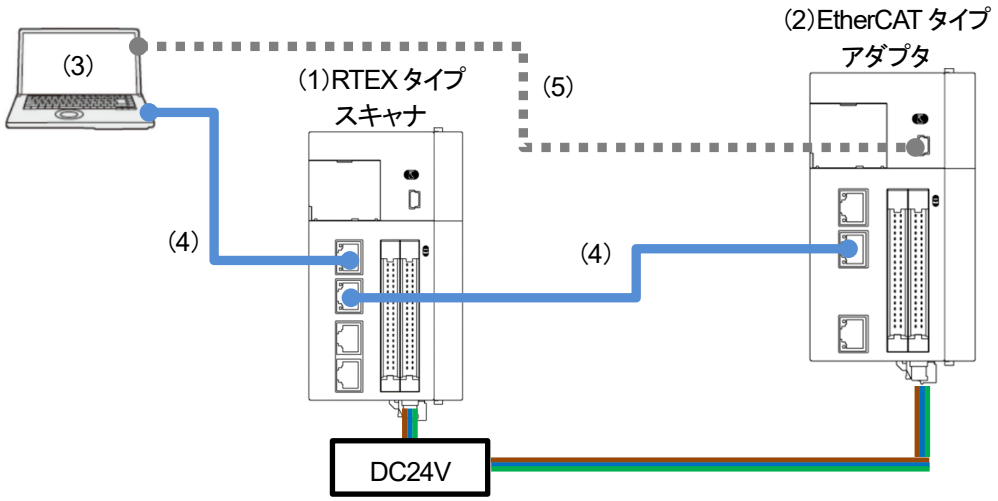
式	アプリケーション	タイプ	値	設定済みの値	実行点	アドレス
IoConfig_Globals_Mapping.wEIPAdInput01	Device.Application	WORD	1		サイクリックモニタリング	%IW32
IoConfig_Globals_Mapping.wEIPAdInput02	Device.Application	WORD	2		サイクリックモニタリング	%IW33
IoConfig_Globals_Mapping.dwEIPAdInput03	Device.Application	DWORD	3		サイクリックモニタリング	%ID17
IoConfig_Globals_Mapping.wEIPAdOutput01	Device.Application	WORD	10		サイクリックモニタリング	%QW28
IoConfig_Globals_Mapping.wEIPAdOutput02	Device.Application	WORD	11		サイクリックモニタリング	%QW29
IoConfig_Globals_Mapping.dwEIPAdOutput03	Device.Application	DWORD	12		サイクリックモニタリング	%QD15

1－2　必要な機器の準備～配線

以下の機器を用意してください。

No.	名称	
(1)	GM1 コントローラ 1 台(RTEX タイプ):スキャナ	(本テキストでは、RTEX タイプと EtherCAT タイプ 1 台ずつ使用)
(2)	GM1 コントローラ 1 台(EtherCAT タイプ):アダプタ	
(3)	PC(GM Programmer インストール済み)	
(4)	LAN ケーブル:2 本	
(5)	USB ケーブル(mini-b)	

下図のように配線してください。

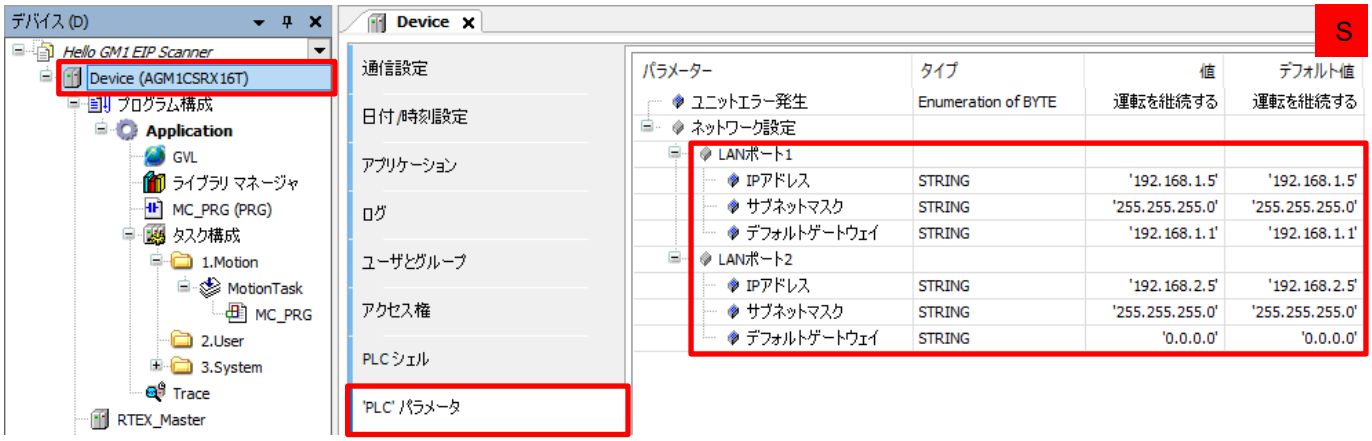


※EIP には EDS ファイルが必要になりますので、以下弊社 web ページよりダウンロードを行ってください。
https://www3.panasonic.biz/ac/j/motor/motion-controller/mc/gm1/index.jsp#Software

1－3　RTEX タイプ:スキャナ　IP アドレスの設定～ネットワークスキャン

手順 1

GM Programmer を開き、「Device」をダブルクリックします。
「PLC パラメータ」を選択し、LAN ポート 1 と LAN ポート 2 の IP アドレスを確認します。



LAN ポート 1(初期値)

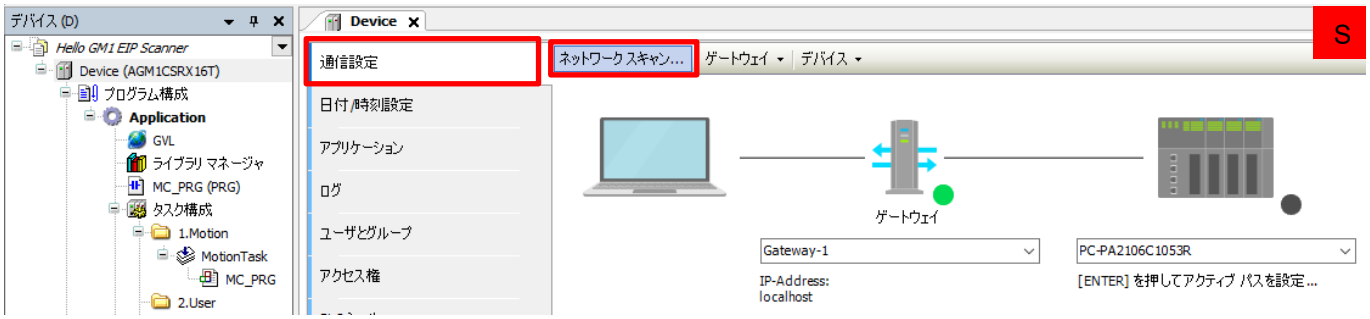
IP アドレス	192.168.1.5
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	192.168.1.1

LAN ポート 2(初期値)

IP アドレス	192.168.2.5
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	0.0.0.0

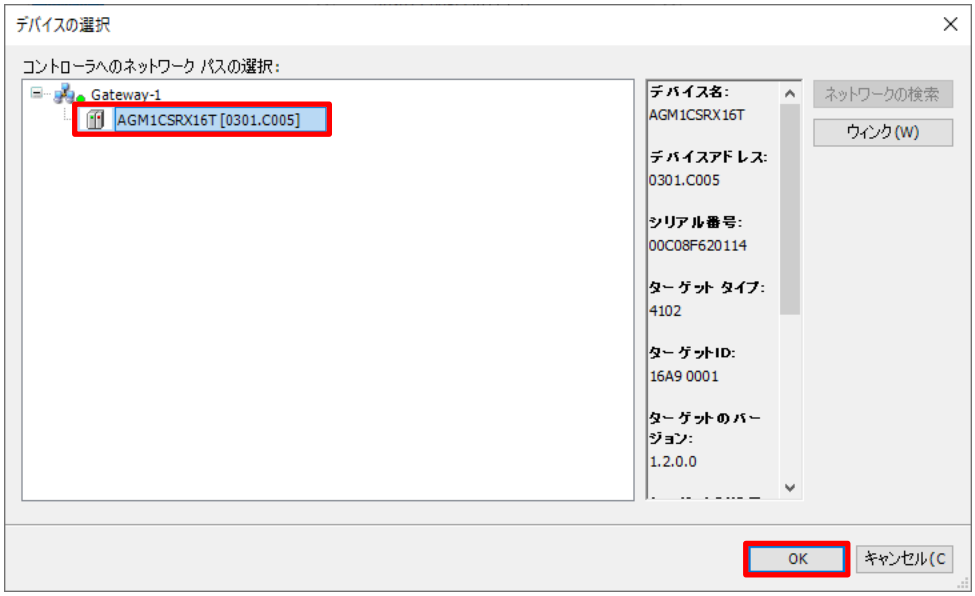
手順２

「通信設定」を選択し、「ネットワークスキャン」をクリックします。



手順３

接続するデバイスを選択し「OK」をクリックします。

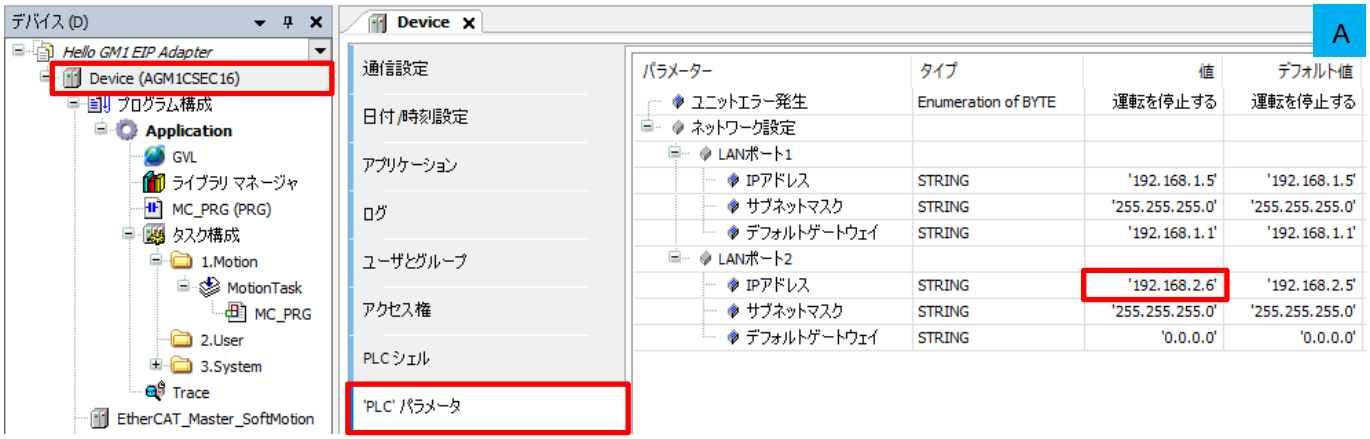


1－4　EtherCAT タイプ：アダプタ　IP アドレスの設定～USB 追加

手順１

GM Programmer を開き、「Device」をダブルクリックします。

「PLC パラメータ」を選択し、LAN ポート 2 の IP アドレスを「192.168.2.6」に変更します。

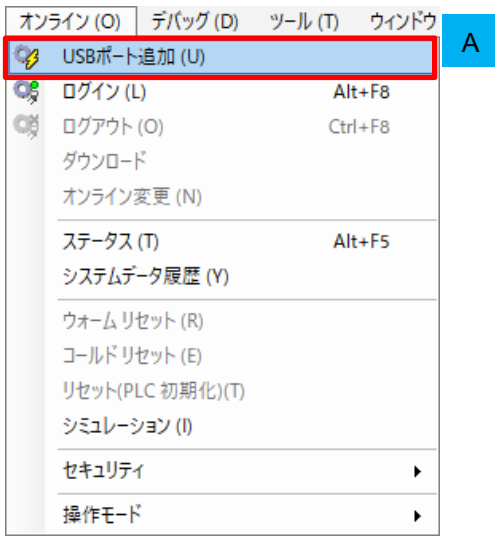


LAN ポート 2

IP アドレス	192.168.2.6
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	0.0.0.0

手順２

メニューバーの**オンライン→USB ポート追加**をクリックします。



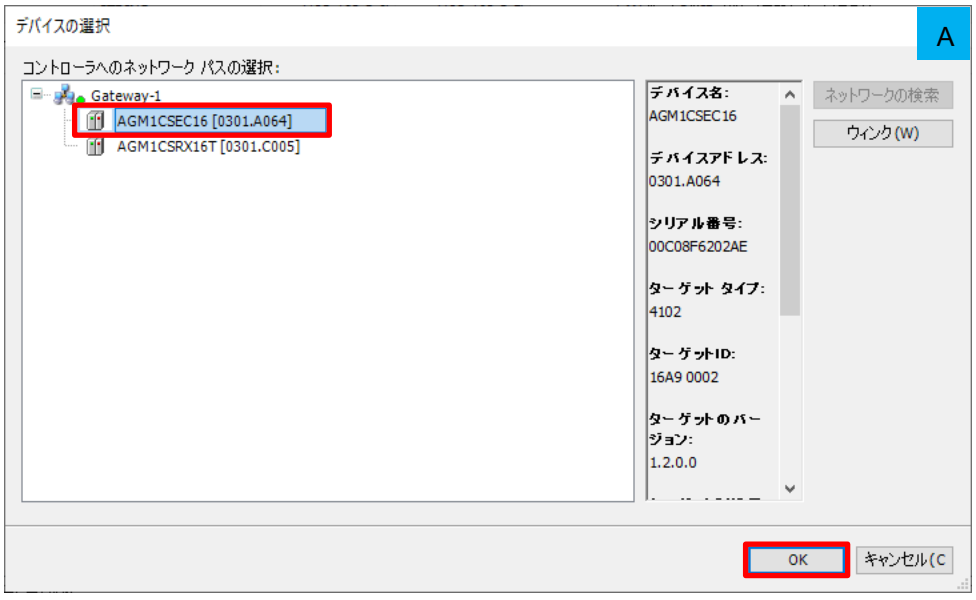
手順３

「USB ポート追加」ダイアログが表示されます。デバイスと使用ポートを確認して「OK」をクリックします。

手順４

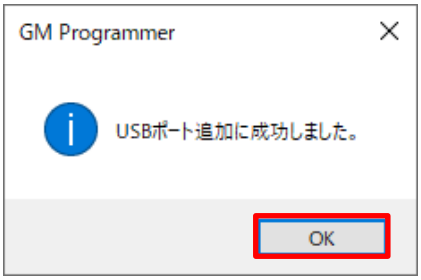
「デバイスの選択」ダイアログが表示されます。

接続するデバイスを選択し「OK」をクリックします。



手順５

接続が完了すると、PC と GM1 コントローラ間の通信インターフェイスに USB が追加されます。



2 スキャナ側設定

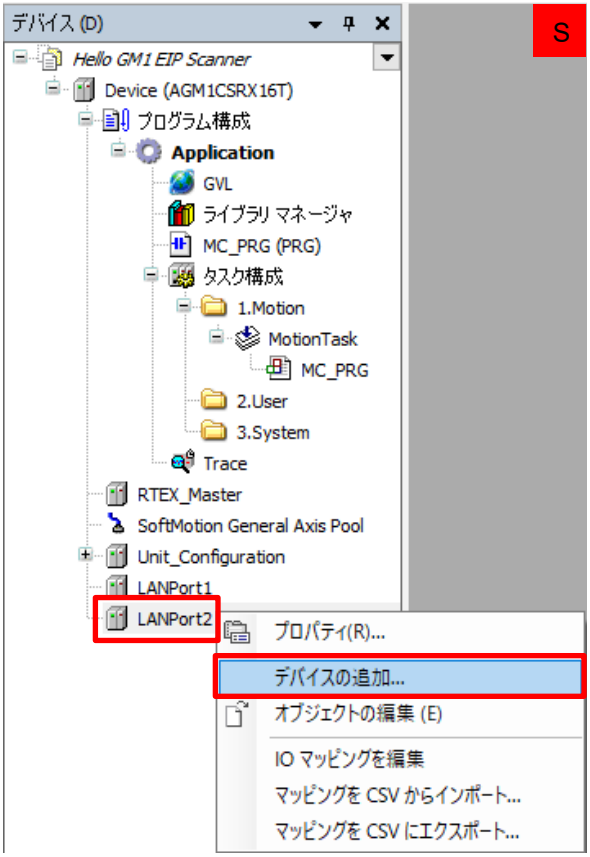
以下の順番で、スキャナ側の設定からしていきます。

2－1 デバイスの追加

接続するアダプタを登録します。

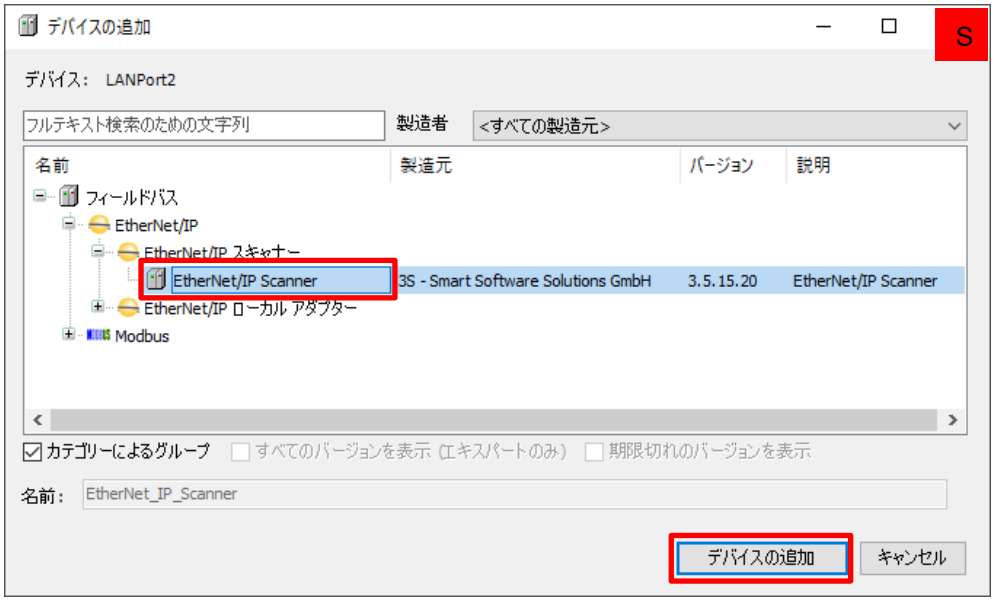
手順 1

LAN Port2 に EtherNet/IP スキャナのデバイスを追加します。
ナビゲータウィンドウの「LANPort2」を右クリックして、「デバイスの追加」をクリックします。

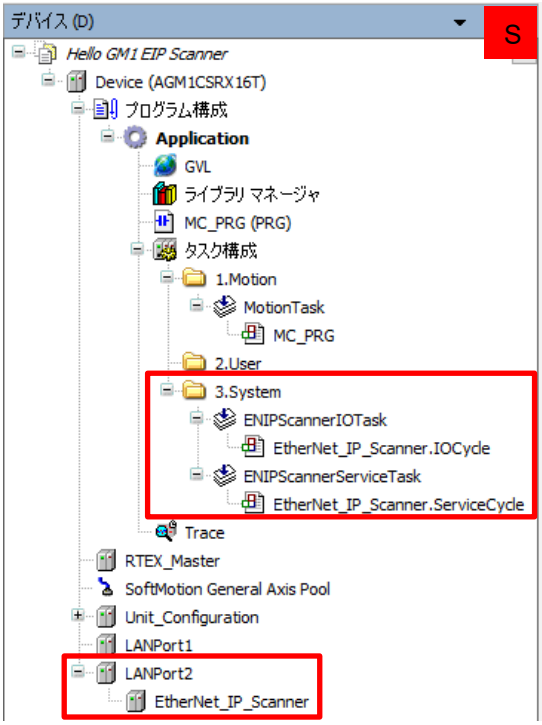


手順２

「EtherNet/IP」ー「EtherNet/IP スキャナ」ー「EtherNet/IP Scanner」を選択し、「デバイスの追加」をクリックします。

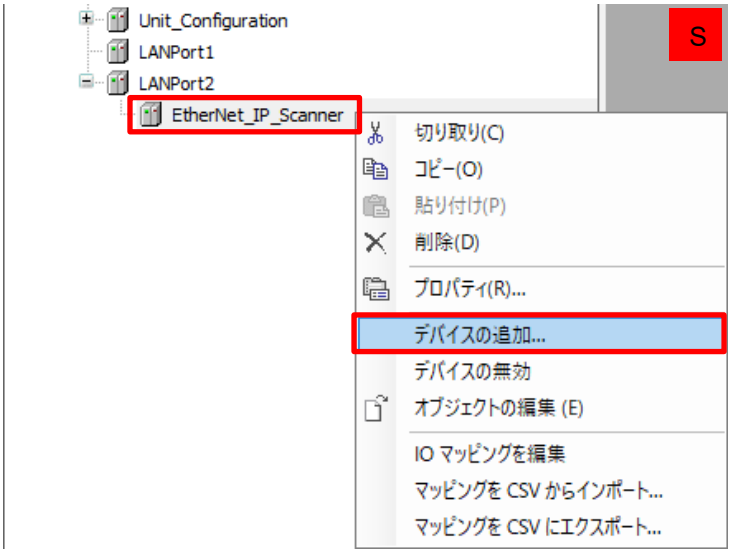


「タスク構成」ー「3.System」に「ENIPScannerIOTask」と「ENIPScannerServiceTask」が追加されます。
「LANPort2」に「EtherNet_IP_Scanner」が追加されます。



手順３

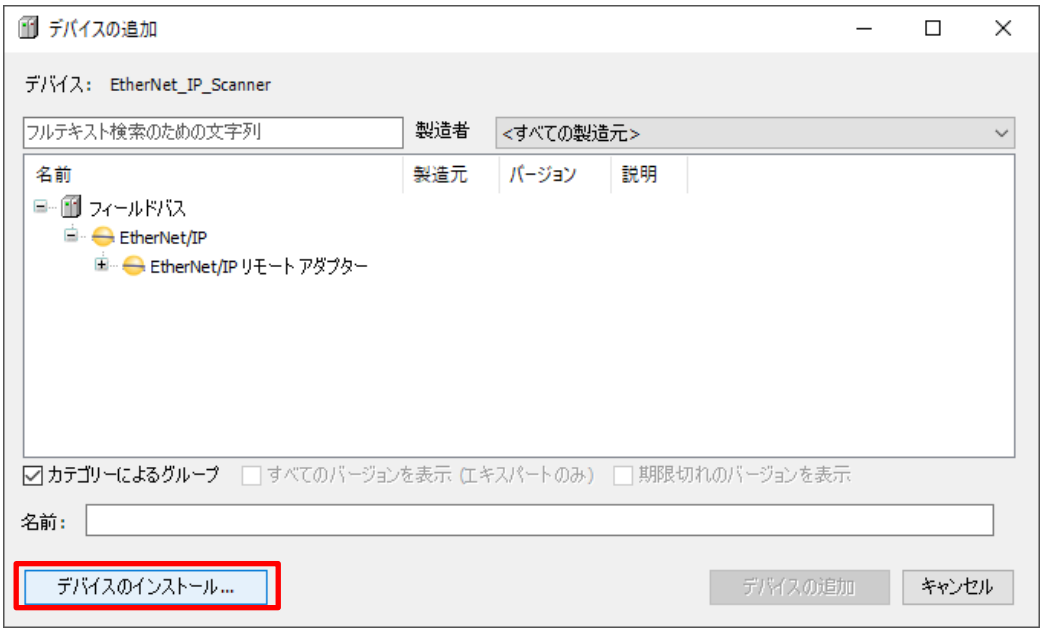
「EtherNet_IP_Scanner」を右クリックして、「デバイスの追加」をクリックします。



手順４

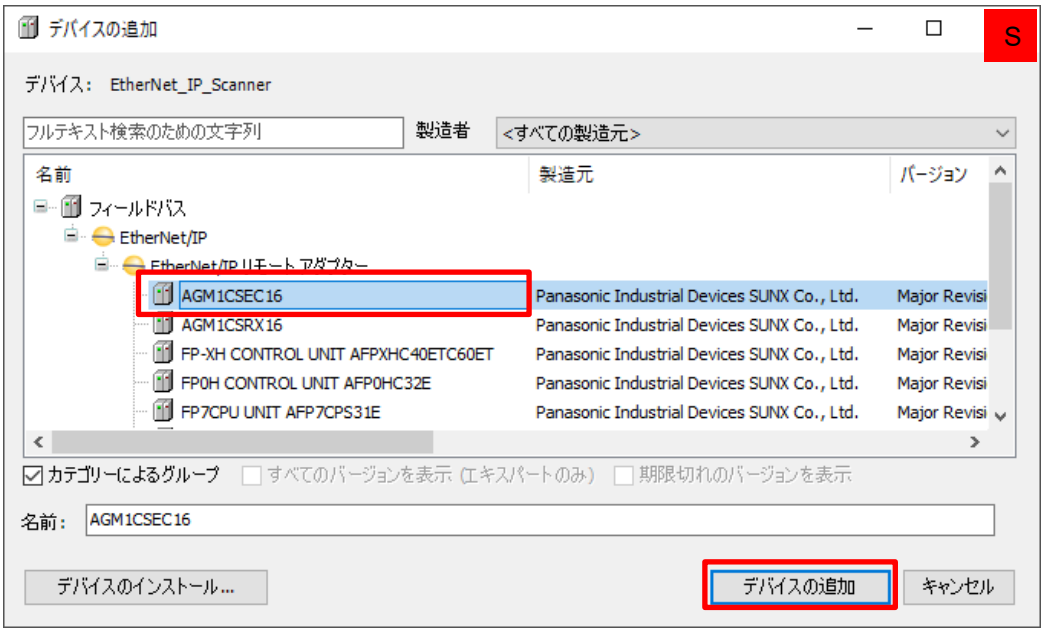
表示されたダイアログの「デバイスのインストール」をクリックします。

あらかじめダウンロードした EDS ファイルを選択します。



手順５

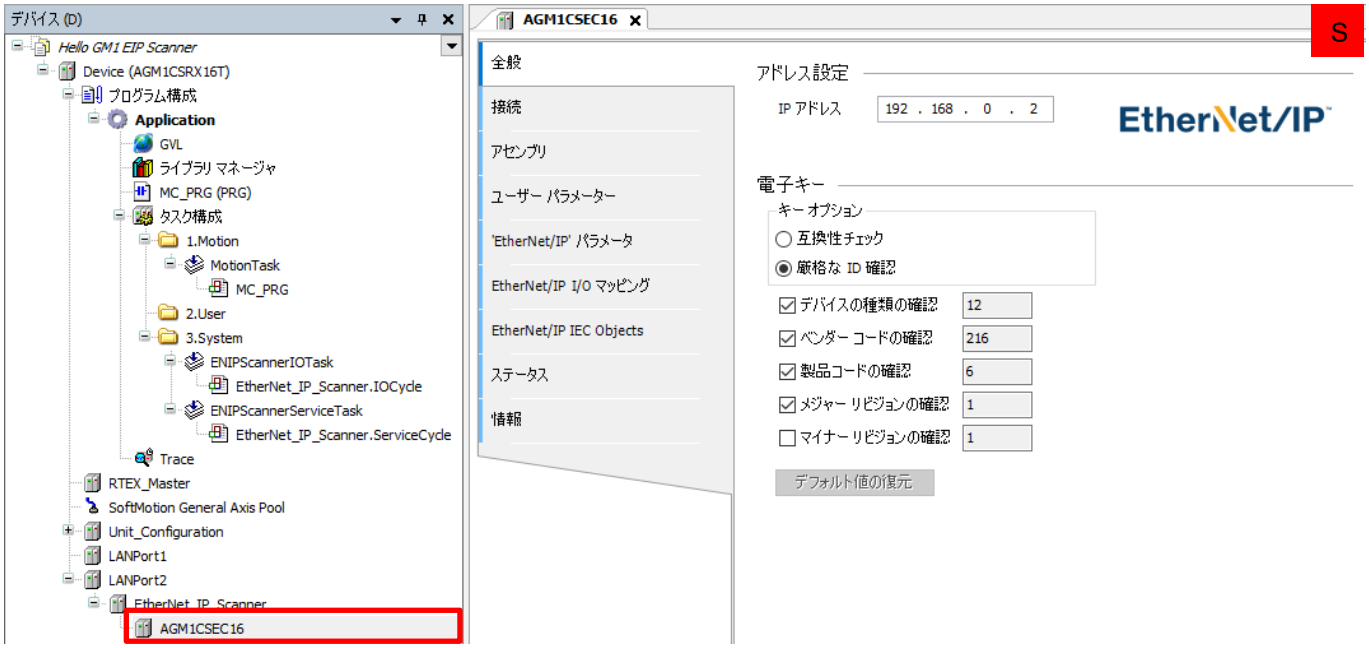
「AGM1CSEC16」と「AGM1CSRX16」が追加されますので、「AGM1CSEC16」を選択して「デバイスの追加」をクリックします。



手順６

「EtherNet_IP_Scaner」に「AGM1CSEC16」が追加されました。

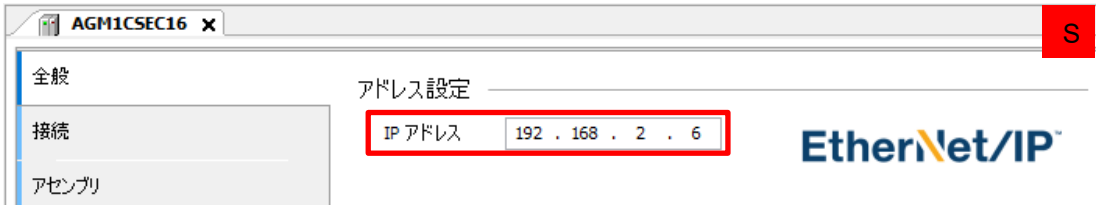
「AGM1CSEC16」をダブルクリックしてウインドウを開きます。



2−2 デバイスの設定

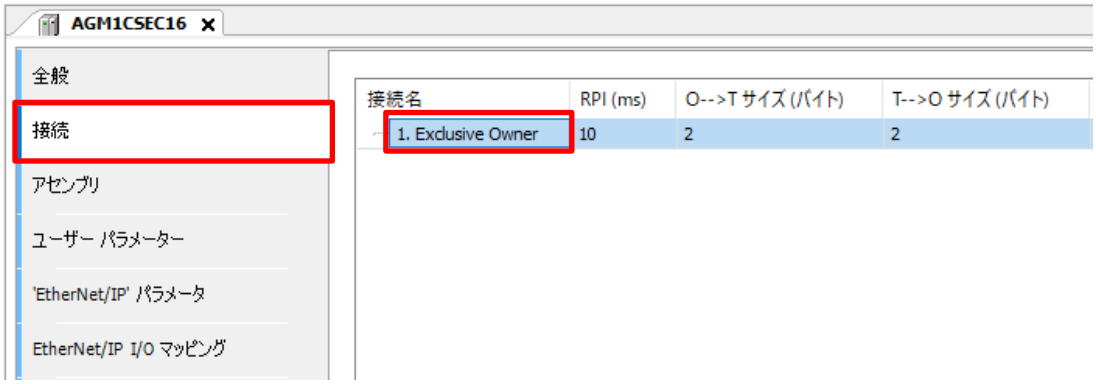
手順 1

「全般」タブで Scanner 機器の IP アドレスを設定します。
本テキストでのスキャナ側 GM1 の IP アドレスは「192.168.2.6」を使用するため、下図の様に設定を行います。



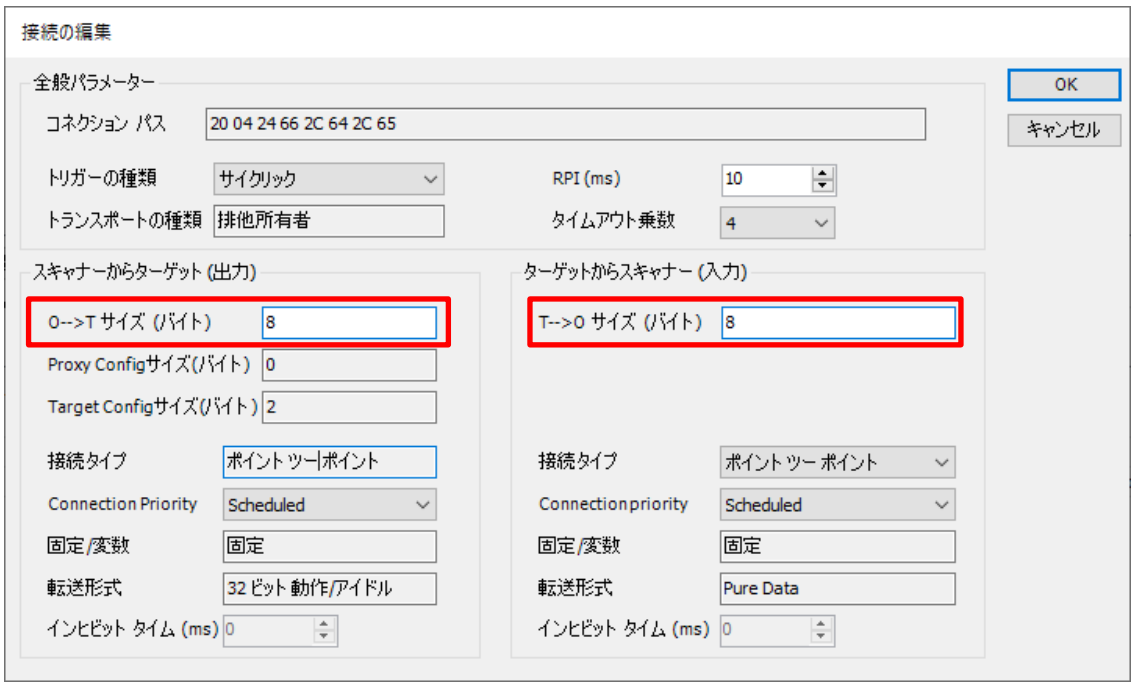
手順 2

「接続」タブを選択し、「1.Exclusive Owner」をダブルクリックします。



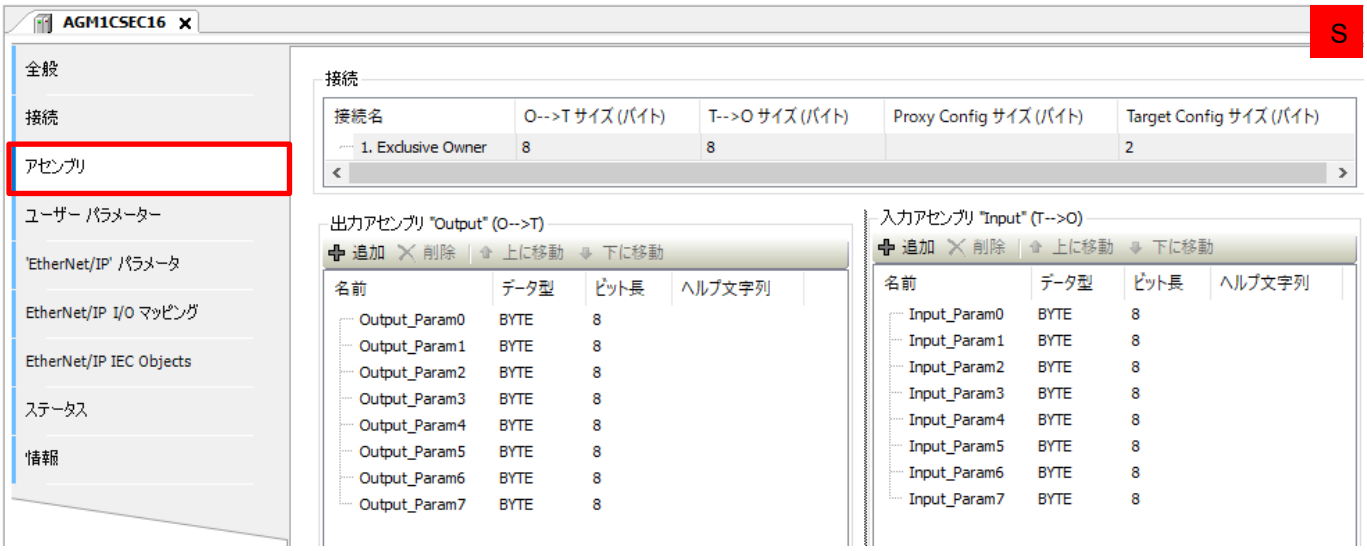
手順 3

「接続の編集」ダイアログが表示されますので、通信するデータの内容を設定します。
・スキャナからターゲット（出力）
O ---＞T サイズ（バイト）:8
・ターゲットからスキャナ（入力）
T ---＞O サイズ（バイト）:8



手順４

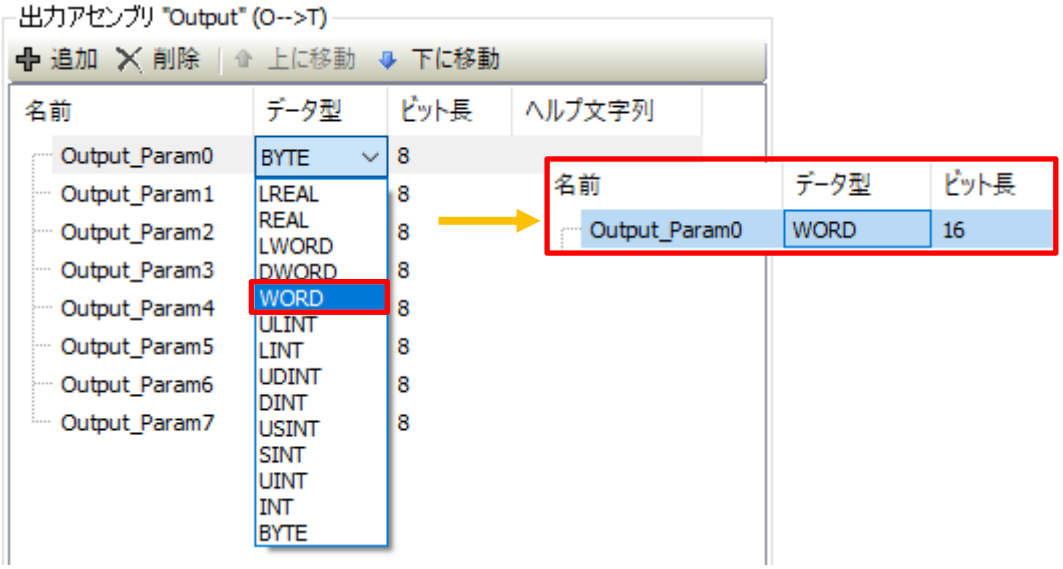
「アセンブリ」を選択します。



「出力アセンブリ」と「入力アセンブリ」で先ほど設定した８バイト分のデータの割り付けを設定します。
デフォルトではすべて１バイトのデータで設定されています。

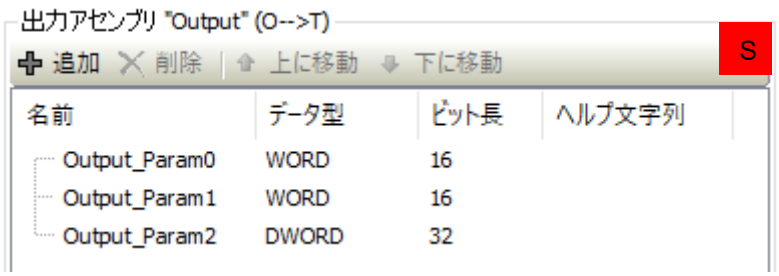
手順５

「Output_Param0」の「データ型」欄を選択して「WORD」を選択すると、設定のデータ型が変更されます。



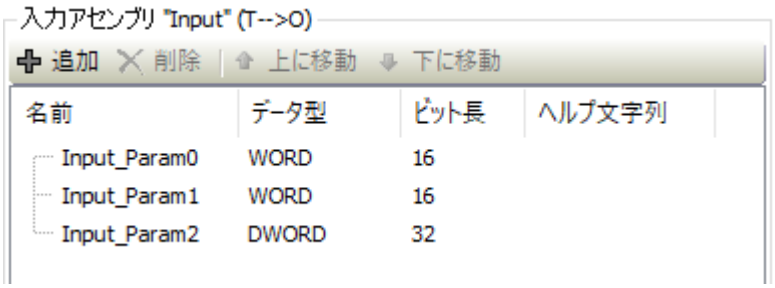
手順６

同様に「Output_Param1」「Output_Param2」をそれぞれ「WORD」「DWORD」に変更します。
「Output_Param3」以下は不要なため削除します。



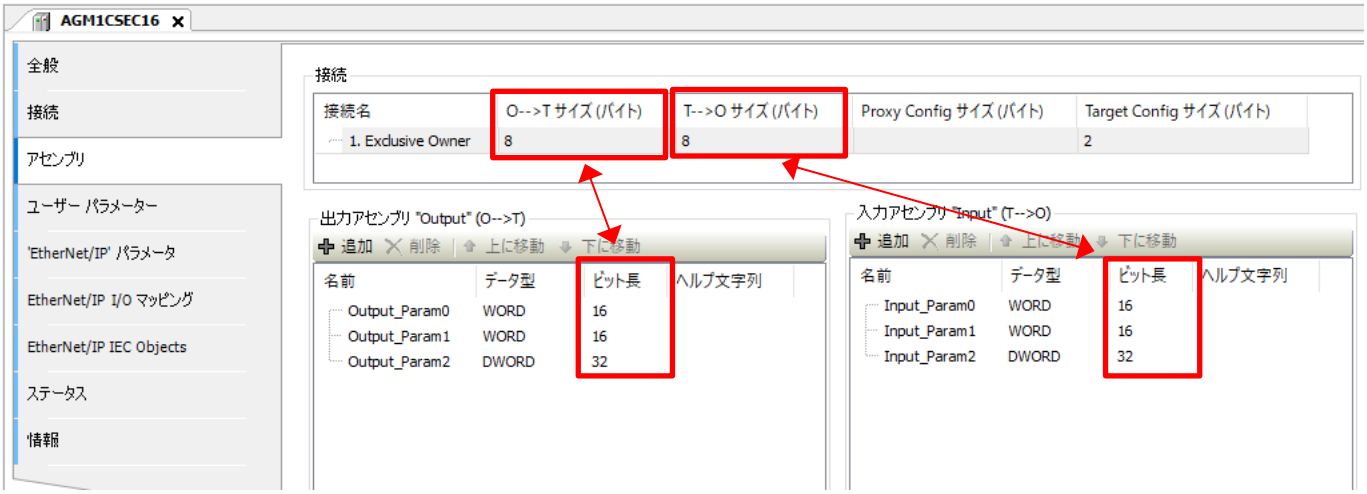
手順７

「入力アセンブリ」も「出力アセンブリ」と同様に設定します。



コラム

削除をしないとデータサイズが登録した「8 バイト」となりません。
スキャナ側 GM1 でもデータサイズを「8 バイト」と設定するため、スキャナーアダプタ間で設定されたデータサイズに
差異が生じ、通信設定異常となってしまいます。



2－3　変数の登録

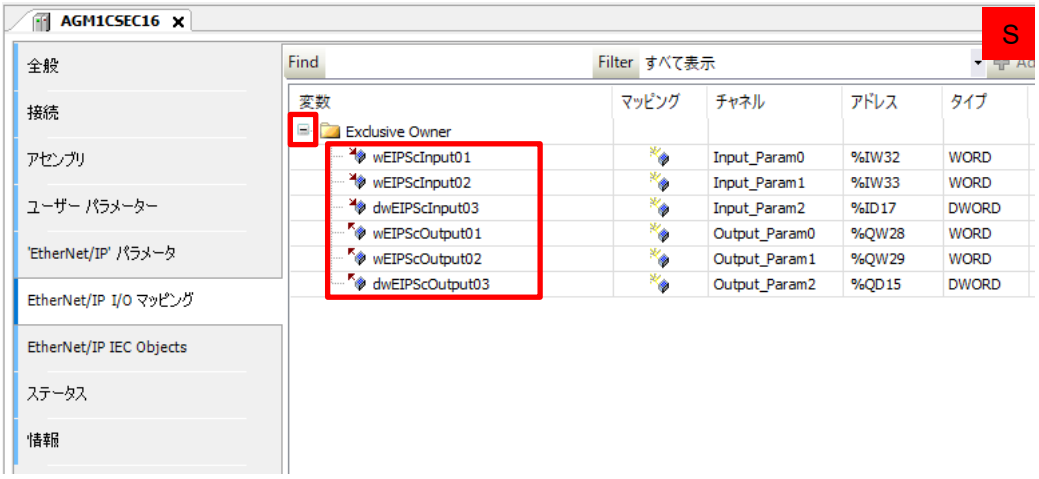
設定した各データに変数を登録します。

手順 1

「EtherNet/IP I/O マッピング」タブを選択します。

「Exclusive Owner」を展開し、変数を登録します。

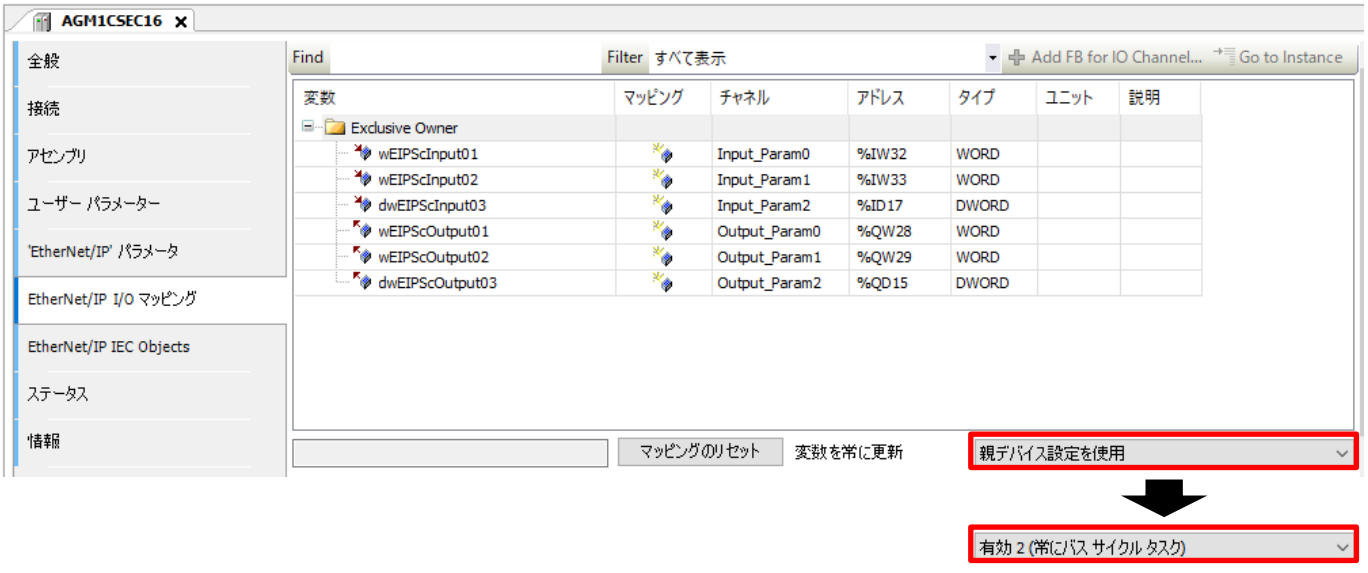
変数	チャネル
wEIPScInput01	Input_Param0
wEIPScInput02	Input_Param1
dwEIPScInput03	Input_Param2
wEIPScOutput01	Output_Param0
wEIPScOutput02	Output_Param1
dwEIPScOutput02	Output_Param2
dwEIPScOutput03	Output_Param2



手順 2

「変数を常に更新」の右側「親デバイス設定を使用」→「有効 2（常にバス サイクル タスク）」に変更します。

※設定内容詳細はマニュアルを参照ください。



以上でスキャナ側 GM1 コントローラの設定は完了です。

3 アダプタ側設定

以下の順番で、アダプタ側の設定をしていきます。

コラム⑧ EIP アダプタ機能 モジュールについて

GM1 コントローラがアダプタの場合は、入出力データごとに「モジュール」を設定します。
モジュールには下記の 10 種類から選択します。

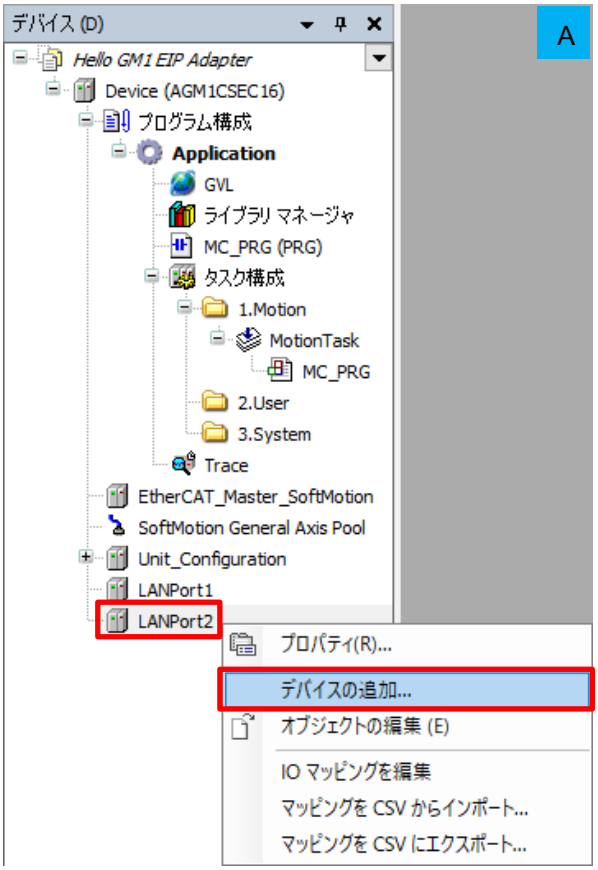
モジュール種別	サイズ	方向
Byte Input	1byte	O → T
Byte Output	1byte	T → O
Word Input	1word（2byte）	O → T
Word Output	1word（2byte）	T → O
DWord Input	1dword（4byte）	O → T
DWord Output	1dword（4byte）	T → O
Real Input 1	単精度実数（4byte）	O → T
Real Output 1	単精度実数（4byte）	T → O
Big Input	509byte	O → T
Big Output	505byte	T → O

複数のデータを扱う場合は、複数のモジュールの設定が必要になります。
例）O → T 1word データを二つ使用する場合：モジュール「Word Input」を 2 つ登録します。

3－1　デバイスの追加

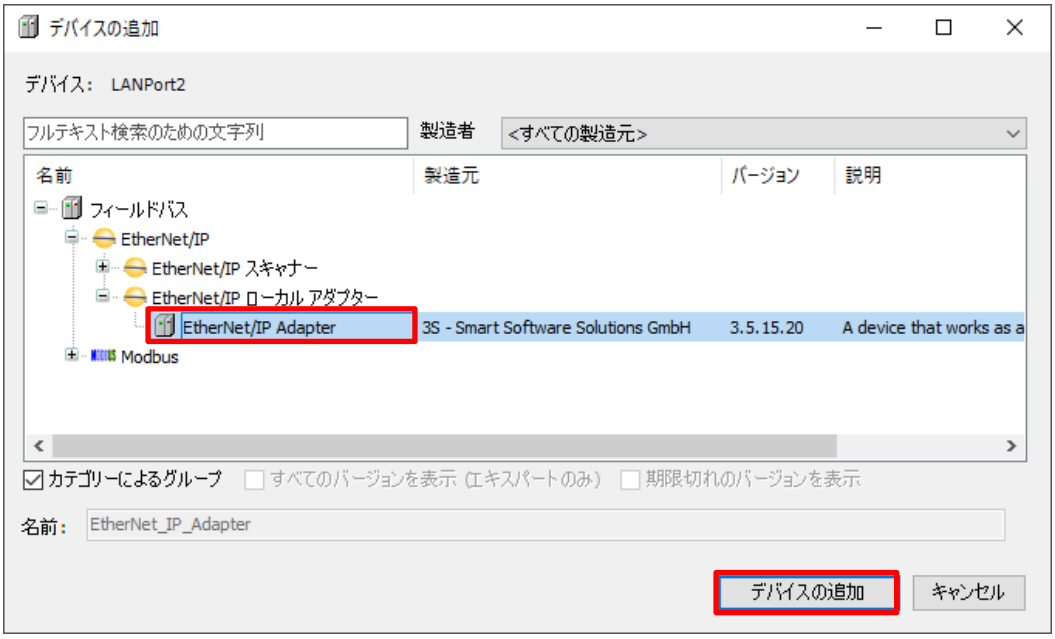
手順 1

LAN Port2 に EtherNet/IP アダプタのデバイスを追加します。
ナビゲータウィンドウの「LANPort2」を右クリックして、「デバイスの追加」をクリックします。

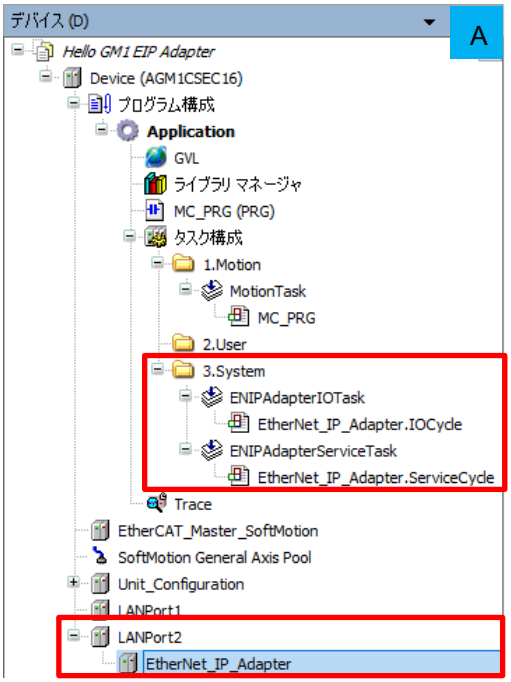


手順 2

「EtherNet/IP ローカルアダプタ」ー「EtehrNet/IP Adapter」を選択し、「デバイスの追加」をクリックします。

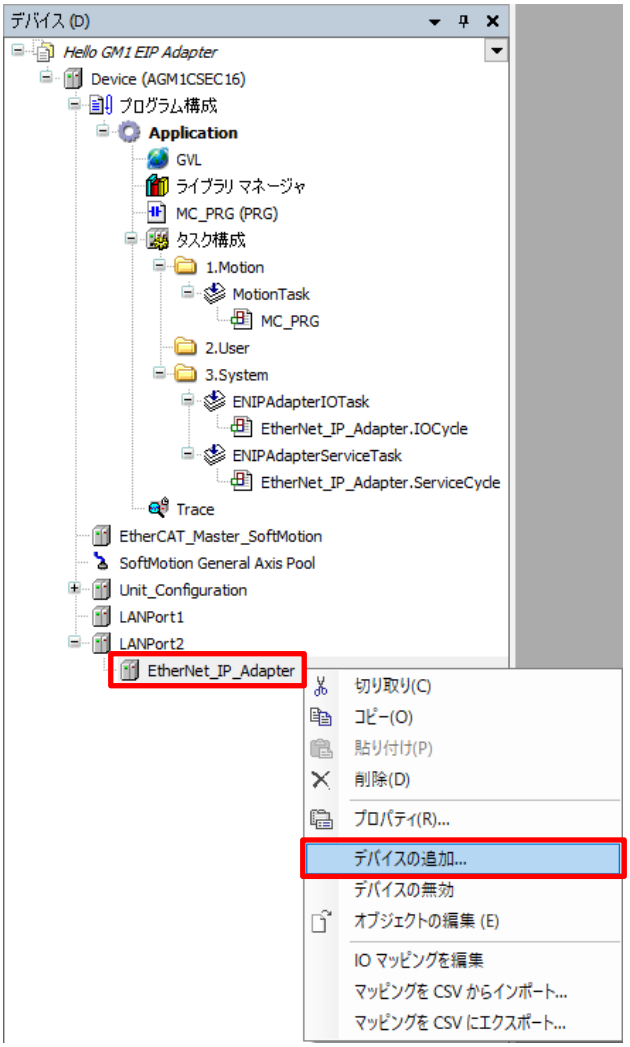


「タスク構成」ー「3.System」に「ENIPAdapterIOTask」と「ENIPAdapterServiceTask」が追加されます。
「LANPort2」に「EtherNet_IP_Adapter」が追加されます。



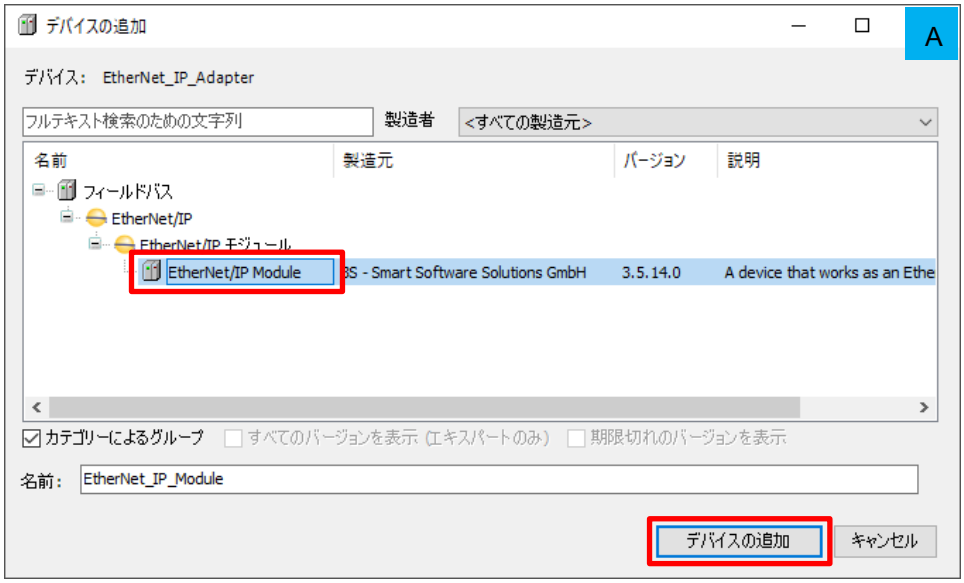
手順３

追加された「EtherNet/IP Adapter」を右クリックして、「デバイスの追加」をクリックします。



手順４

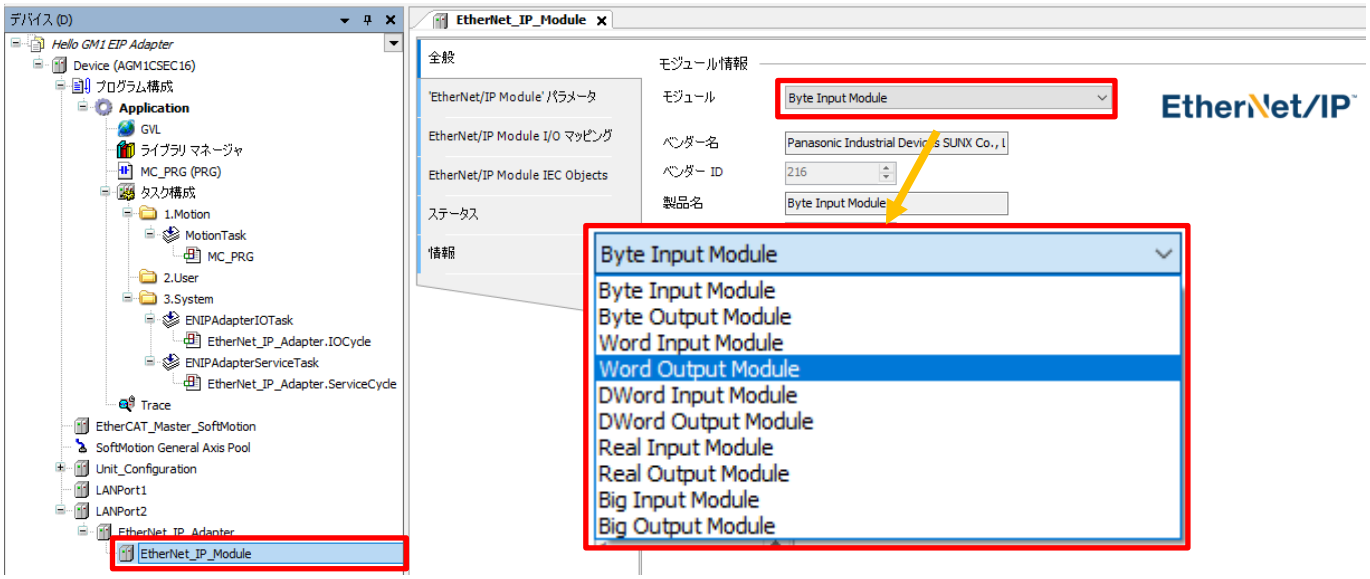
「EtherNet/IP モジュール」ー「EtehrNet/IP Module」を選択し、名前:EtherNet_IP_Module（初期値）のまま、
「デバイスの追加」をクリックします。



3ー2 モジュールの設定

手順１

「EtherNet_IP_Module」をダブルクリックして、「全般」タブを選択します。
「モジュール」がデフォルトで「Byte Input Module」となっているので、
枠内をクリックしてドロップダウンリスト内から「Word Output Module」を選択します。

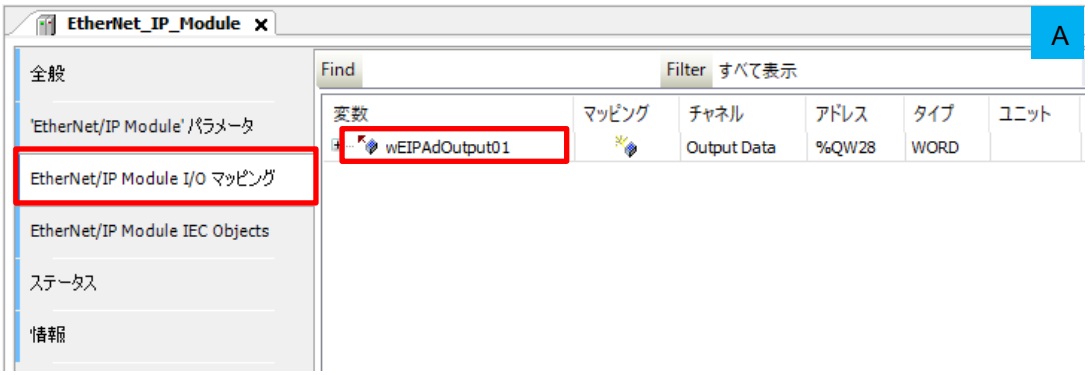


手順２

この 1word 分のデータに変数を設定します。

※変数の設定は後述する「ウォッチ」ウインドウでのモニタに登録するために必要となります。

「EtherNet/IP Module I/O マッピング」タブを選択し、変数名「wEIPAdOutput01」を下図の様に設定します。



手順３

同様の手順で「EtehrNet/IP Module」を追加し、「モジュール」の選択と変数の設定を行います。

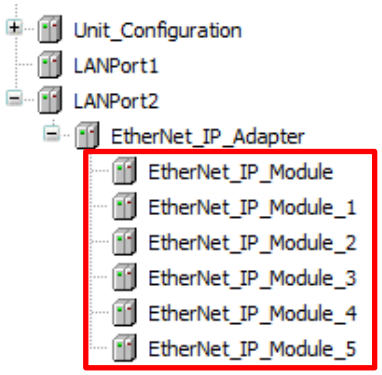
・Output（アダプタ側 GM1 コントローラ ＞ スキャナ側 GM1 コントローラ）データの設定

「EtehrNet/IP Module」名称	モジュール	登録変数名	データ内容
EtherNet_IP_Module	Word Output Mobule	wEipAdOutput01	1word データ　※登録済
EtherNet_IP_Module_1	Word Output Mobule	wEipAdOutput02	1word データ
EtherNet_IP_Module_2	DWord Output Mobule	dwEipAdOutput03	2word データ

・Input（アダプタ側 GM1 コントローラ ＜ スキャナ側 GM1 コントローラ）データの設定

「EtehrNet/IP Module」名称	モジュール	登録変数名	データ内容
EtherNet_IP_Module_3	Word Input Mobule	wEipAdInput01	1word データ
EtherNet_IP_Module_4	Word Input Mobule	wEipAdInput02	1word データ
EtherNet_IP_Module_5	DWord Input Mobule	dwEipAdInput03	2word データ

「LANPort2」ー「EtherNet_IP_Adapter」の下に各項目が設定されたことを確認してください。

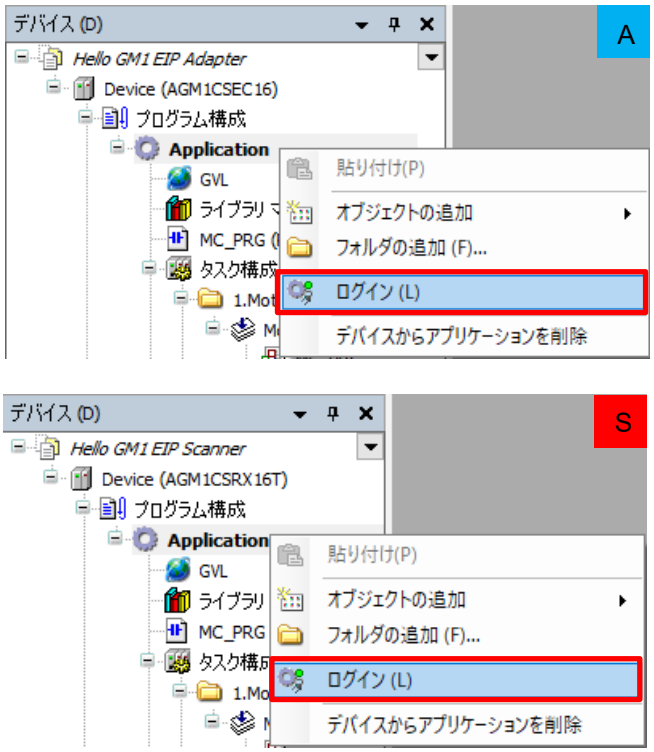


以上でアダプタ側 GM1 コントローラの EtherNet/IP 設定は完了です。

4 通信動作の確認

手順 1

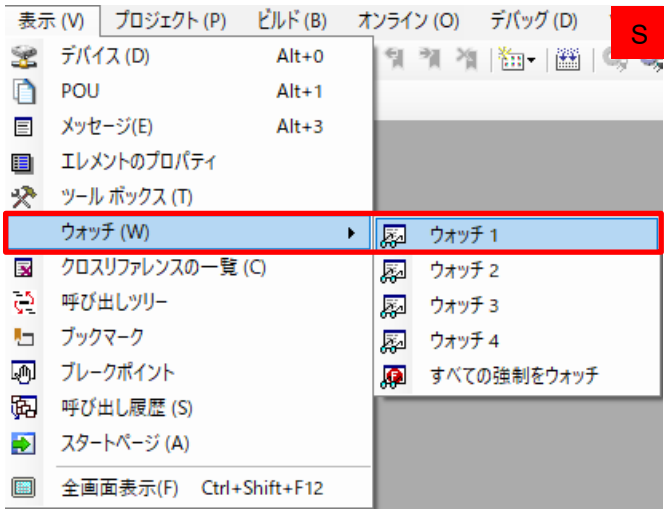
スキャナ側、アダプタ側両方の GM Programmer で「Application」を右クリックし、「ログイン」をクリックします。



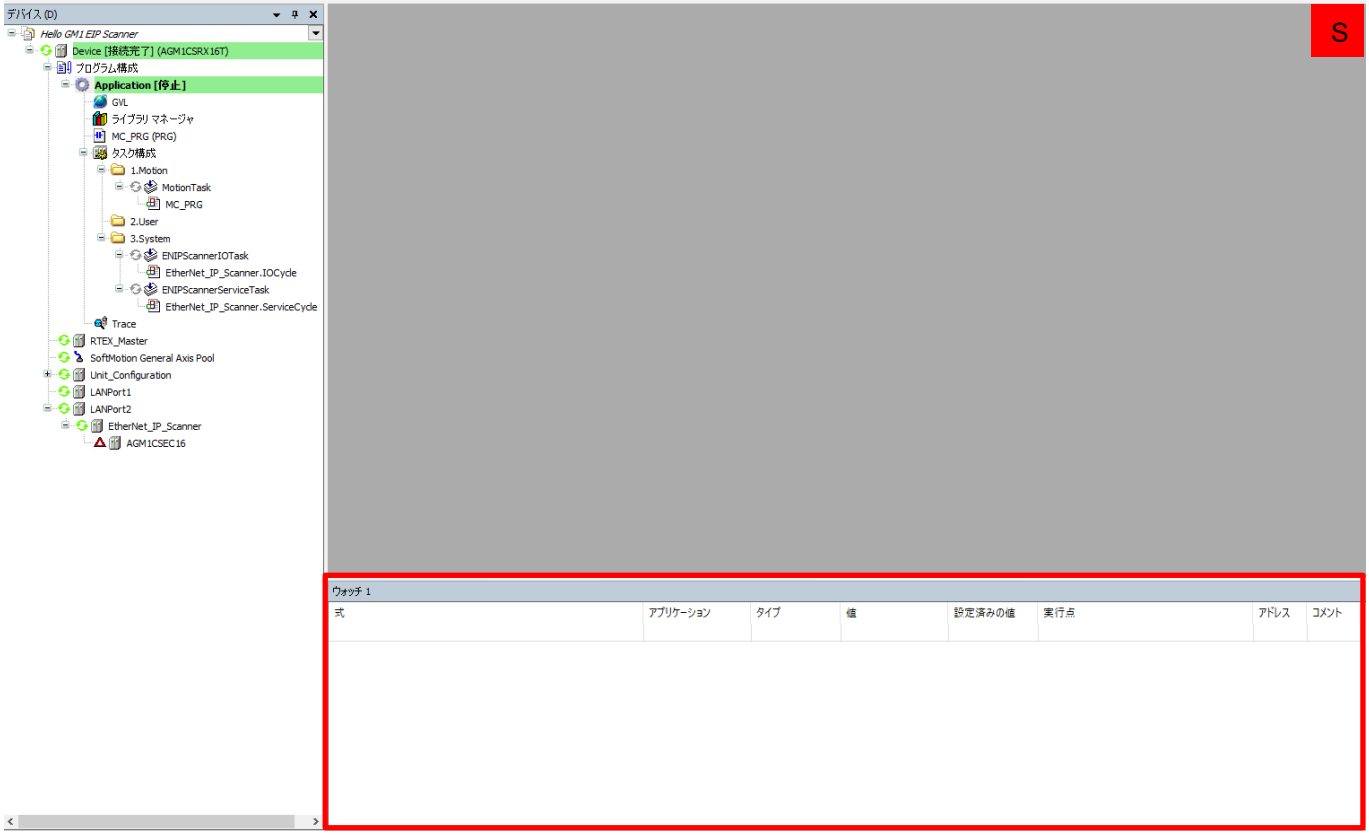
手順 2

まずは、スキャナ側の GM Programmer を開きます。

メニューバーの「表示」ー「ウォッチ」ー「ウォッチ 1」をクリックします。

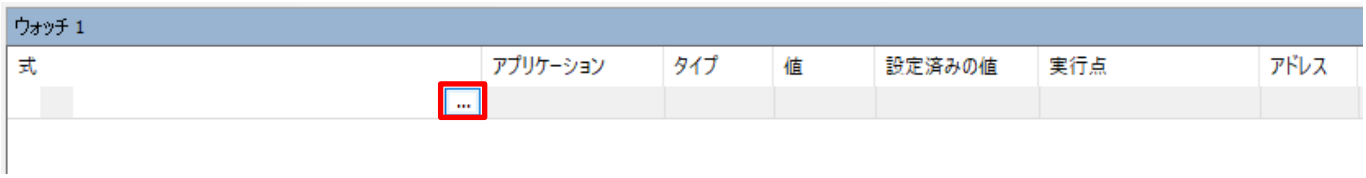


画面下部に「ウォッチ 1」が表示されます。



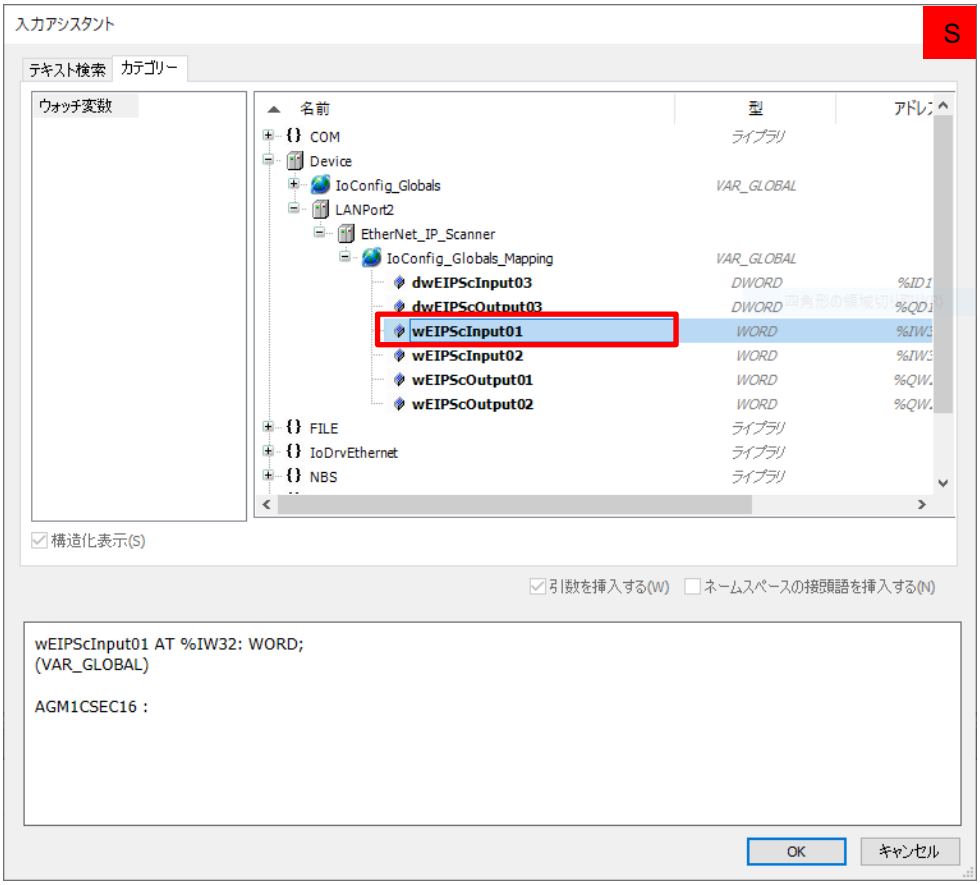
手順３

「ウォッチ 1」の式欄をクリックし ... をクリックします。



手順４

「Device」ー「LANPort2」ー「EtherNet_IP_Scanner」ー「IoConfig_Globals_Mapping」の順で ⊞ :クリックし、展開します。
「wEIPScInput01」を選択し「OK」をクリックします。



同様の手順で、「wEIPScInput02」「dwEIPScInput03」「wEIPScOutput01」「wEIPScOutput02」「dwEIPScOutput03」を
ウォッチ１に登録します。

ウォッチ 1 S

式	アプリケーション	タイプ	値	設定済みの値	実行点	アドレス	コメント
IoConfig_Globals_Mapping.wEIPScInput01	Device.Application	WORD	0		サイクリックモニタリング	%IW32	AGM1CSEC16 :
IoConfig_Globals_Mapping.wEIPScInput02	Device.Application	WORD	0		サイクリックモニタリング	%IW33	AGM1CSEC16 :
IoConfig_Globals_Mapping.dwEIPScInput03	Device.Application	DWORD	0		サイクリックモニタリング	%ID17	AGM1CSEC16 :
IoConfig_Globals_Mapping.wEIPScOutput01	Device.Application	WORD	0		サイクリックモニタリング	%QW28	AGM1CSEC16 :
IoConfig_Globals_Mapping.wEIPScOutput02	Device.Application	WORD	0		サイクリックモニタリング	%QW29	AGM1CSEC16 :
IoConfig_Globals_Mapping.dwEIPScOutput03	Device.Application	DWORD	0		サイクリックモニタリング	%QD15	AGM1CSEC16 :

手順５

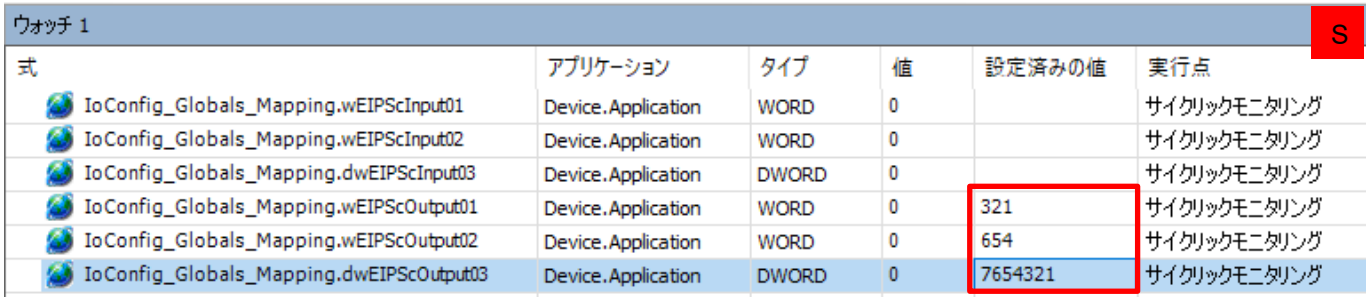
次にアダプタ側の GM Programmer を開きます。
スキャナ側の GM Programmer と同じ手順で「ウォッチ 1」を表示させ、以下変数を登録してください。
「wEIPAdInput01」「wEIPAdInput02」「dwEIPAdInput03」「wEIPAdOutput01」「wEIPAdOutput02」「dwEIPAdOutput03」

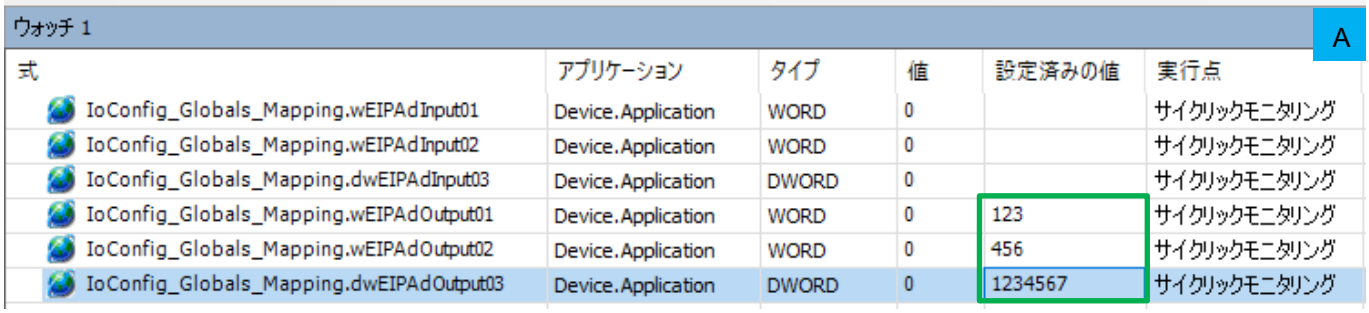
ウォッチ 1 A

式	アプリケーション	タイプ	値	設定済みの値	実行点	アドレス	コメント
IoConfig_Globals_Mapping.wEIPAdInput01	Device.Application	WORD	0		サイクリックモニタリング	%IW32	EtherNet_IP_Module_4 :
IoConfig_Globals_Mapping.wEIPAdInput02	Device.Application	WORD	0		サイクリックモニタリング	%IW33	
IoConfig_Globals_Mapping.dwEIPAdInput03	Device.Application	DWORD	0		サイクリックモニタリング	%ID17	EtherNet_IP_Module_5 :
IoConfig_Globals_Mapping.wEIPAdOutput01	Device.Application	WORD	0		サイクリックモニタリング	%QW28	EtherNet_IP_Module :
IoConfig_Globals_Mapping.wEIPAdOutput02	Device.Application	WORD	0		サイクリックモニタリング	%QW29	EtherNet_IP_Module_1 :
IoConfig_Globals_Mapping.dwEIPAdOutput03	Device.Application	DWORD	0		サイクリックモニタリング	%QD15	EtherNet_IP_Module_2 :

手順 6

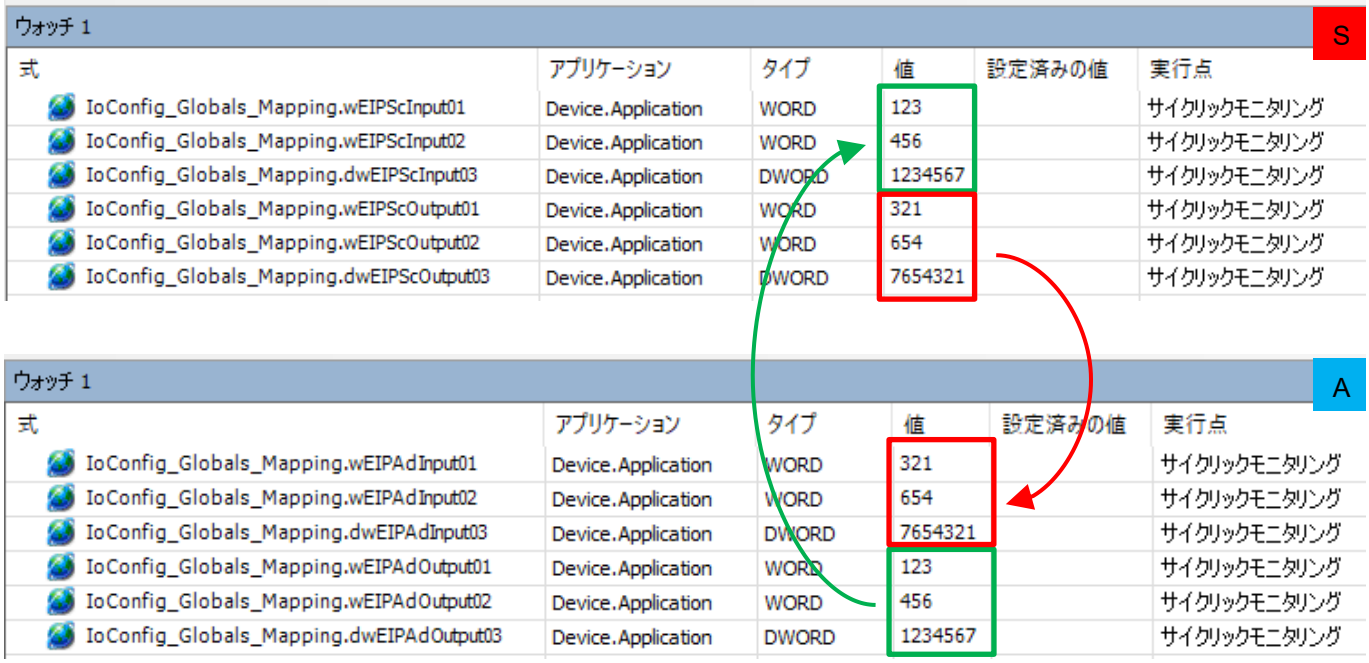
スキャナ側の「wEIPScOutput01」「wEIPScOutput02」「dwEIPScOutput03」
アダプタ側の「wEIPAdOutput01」「wEIPAdOutput02」「dwEIPAdOutput03」の「設定済みの値」に任意の値を入れます。





手順 7

Ctr+F7 を実行して、値を書き込みます。



以上で、GM1 コントローラの EtherNet/IP 通信の動作確認は完了となります。

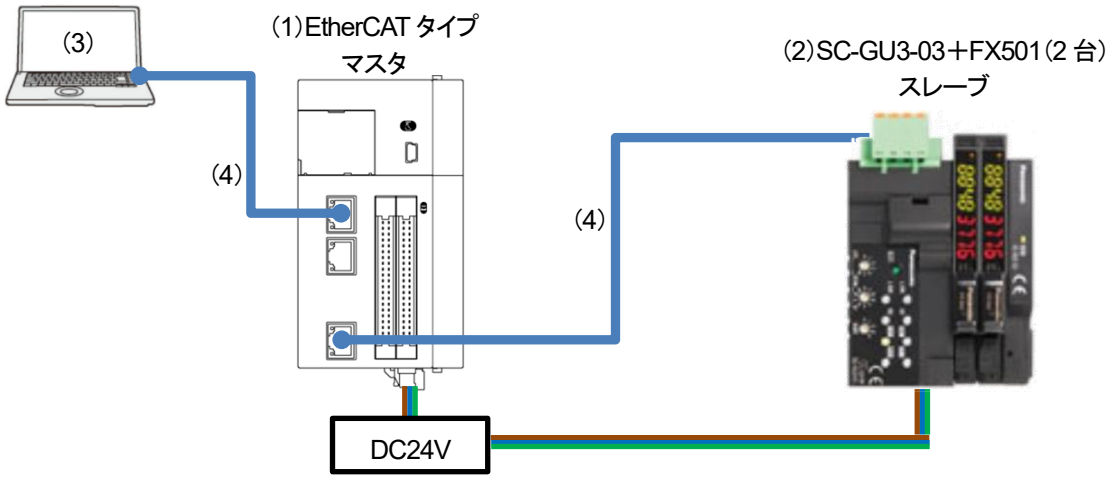
EtherNet 通信　CAT スレーブ

1 基本設定

1－1 必要な機器の準備と配線

以下の機器を用意してください。

No.	名称
(1)	GM1 コントローラ 1 台(EtherCAT タイプ):マスタ
(2)	SC-GU3-03＋FX501(ファイバセンサ)2 台
(3)	PC(GM Programmer インストール済み)
(4)	LAN ケーブル:2 本



1－2　ESI ファイルのインストール

手順 1

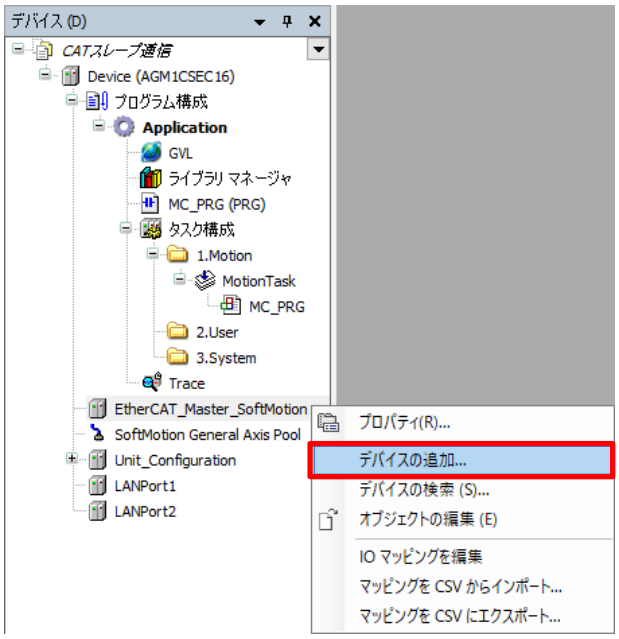
以下 HP より、ESI ファイルをダウンロードします。

https://www3.panasonic.biz/ac/j/dl/software/index.jsp?series_cd=1454

	タイトル	言語
TOOL ツール	SC-GU3-03用ESIファイル EtherCAT規格のModular Device Profile(MDP)規格(ETG.5001.1)に対応したファイルになります。	JP
	SC-GU3-03用ESIファイル 2020年4月以前生産分のSC-GU3-03用です。	JP

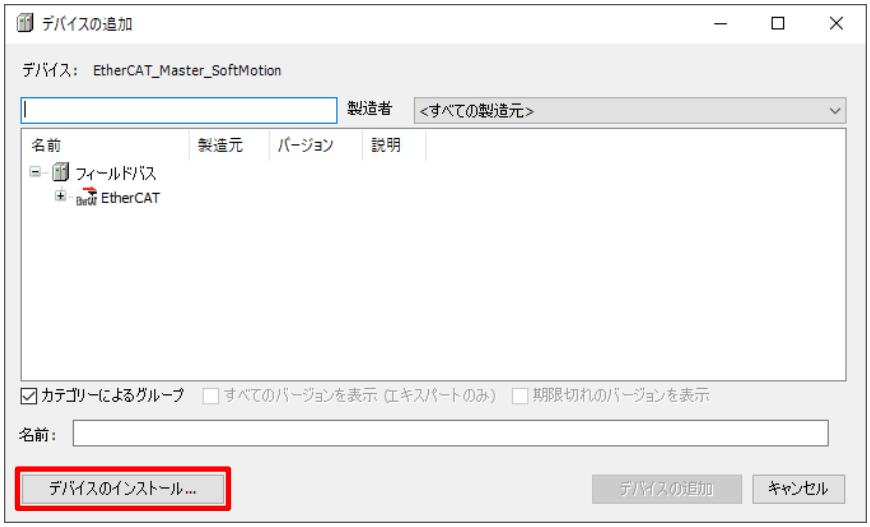
手順 2

「EtherCAT_Master_SoftMotion」で右クリックし、「デバイスの追加」をクリックします。



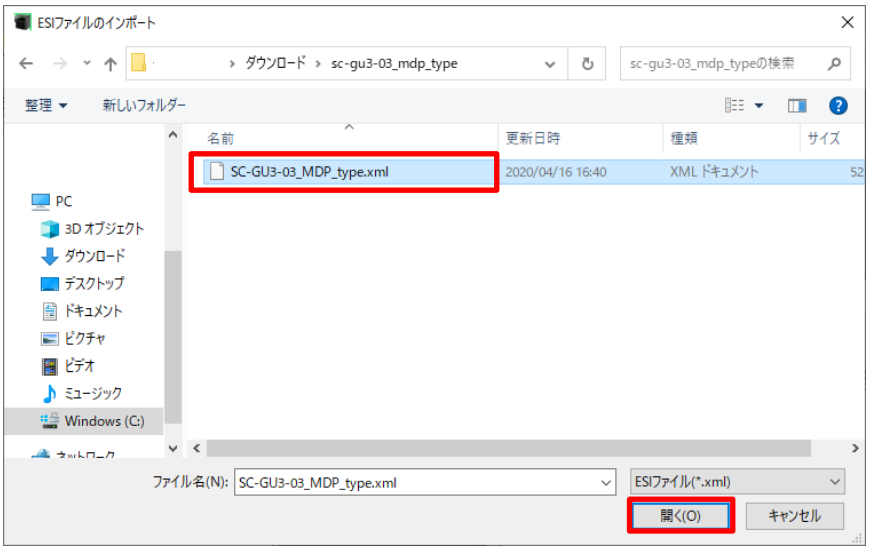
手順 3

「デバイスの追加」ダイアログが表示されたら、「デバイスのインストール…」をクリックします。



手順 4

先ほどダウンロードしたファイルを選択し、「開く」をクリックします。

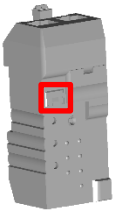


これで SC-GU3-03 がインストールできました。

1－3　デバイス（SC-GU3-03）の追加

手順 1

赤枠内のカバーを開け、DIP スイッチの設定を行います。



1	ON
2	ON
3	OFF

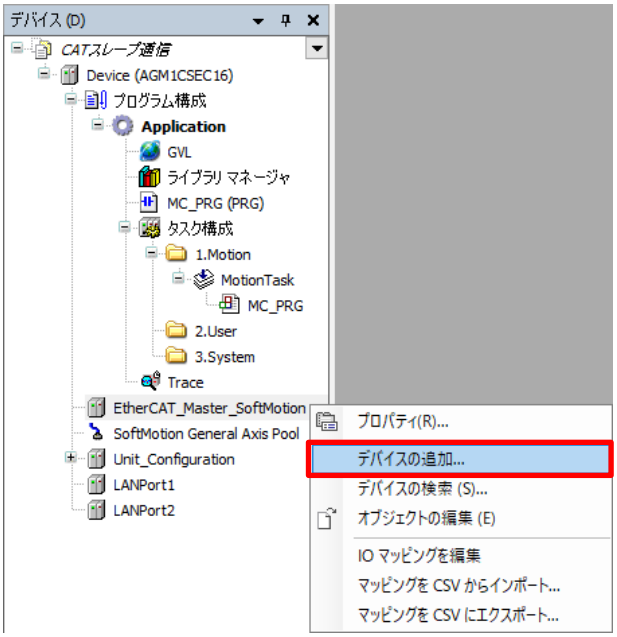
INFO

I/O 通信の動作モードは 3 モードあります。動作モードは、DIP スイッチにより切り替えることができます。
工場出荷状態ではフルモード設定になっています。

		bit321	T×PDO	R×PDO
mode1	I/O モード	001	2byte	0byte
mode2	チェックモード	010	4byte	0byte
mode3	フルモード	011	44byte	10byte

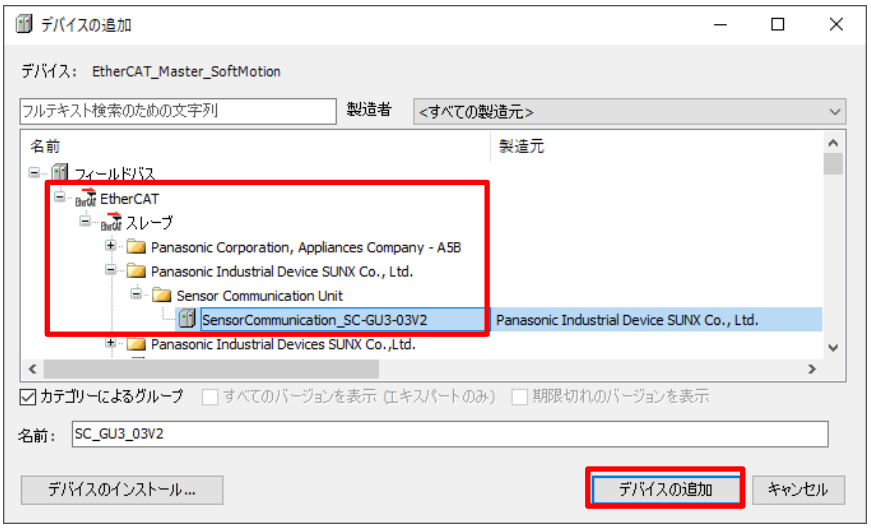
手順２

「EtherCAT_Master_SoftMotion」で右クリックし、「デバイスの追加」をクリックします。



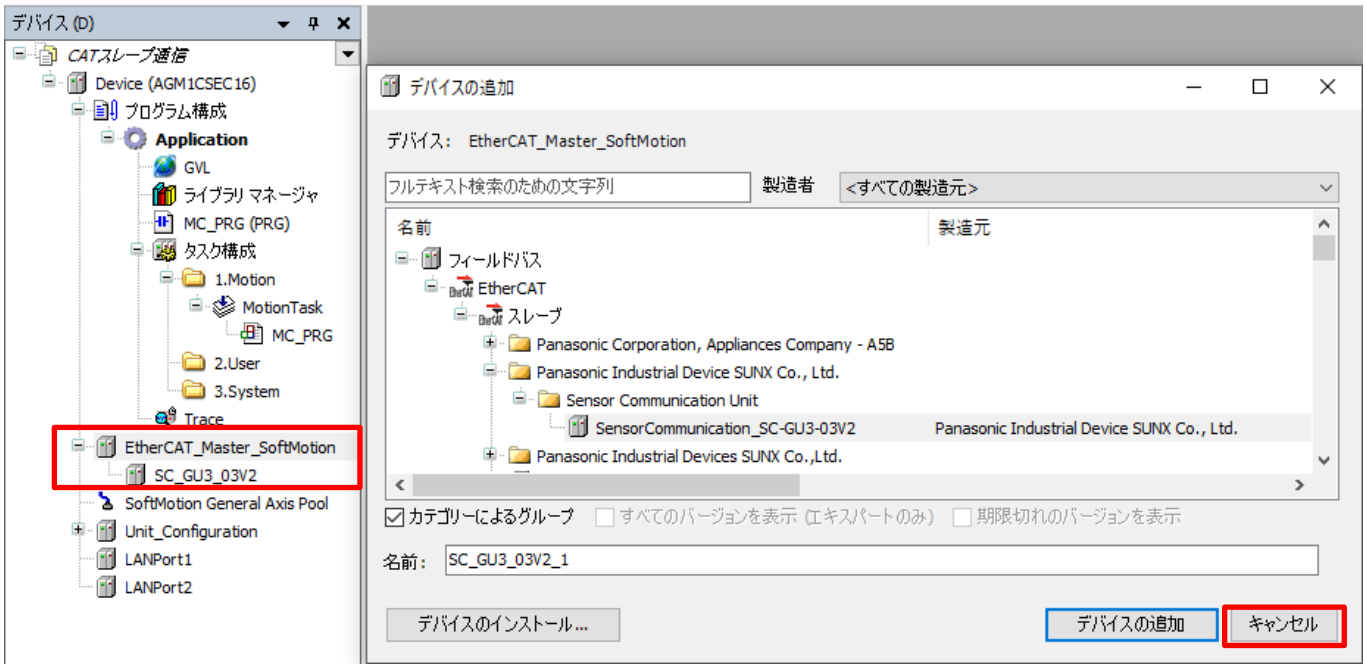
手順３

「EtherCAT」ー「Slave」ー「Panasonic Industrial Device SUNX Co.,Ltd.」ー「SensorCommunication_SC_GU3_03V2」を選択します。「デバイスの追加」をクリックします。



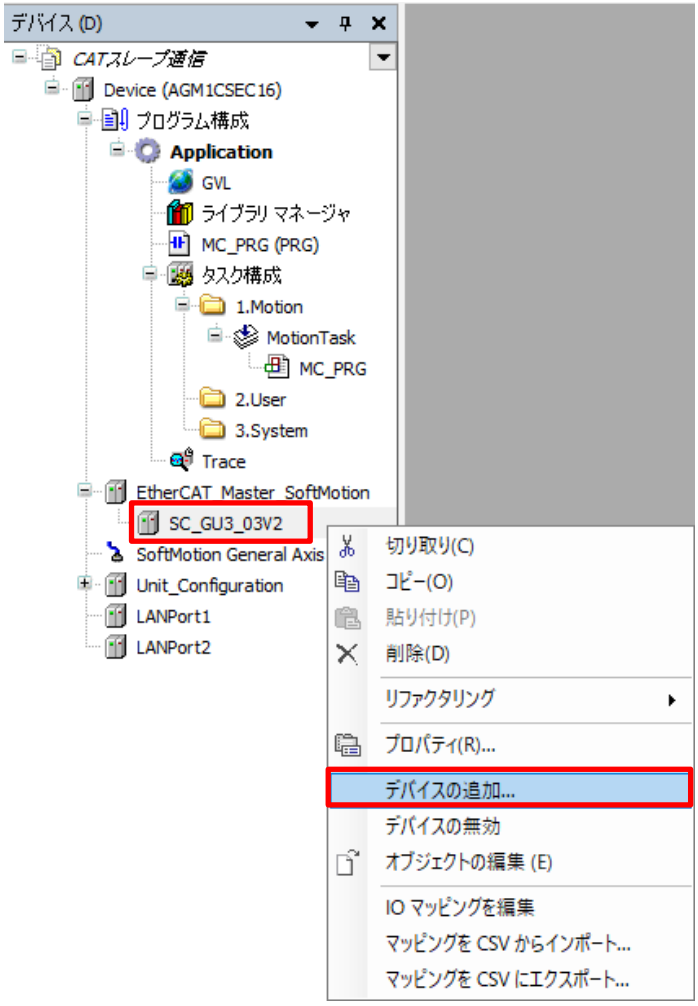
手順４

追加されたことを確認し、「キャンセル」をクリックします。



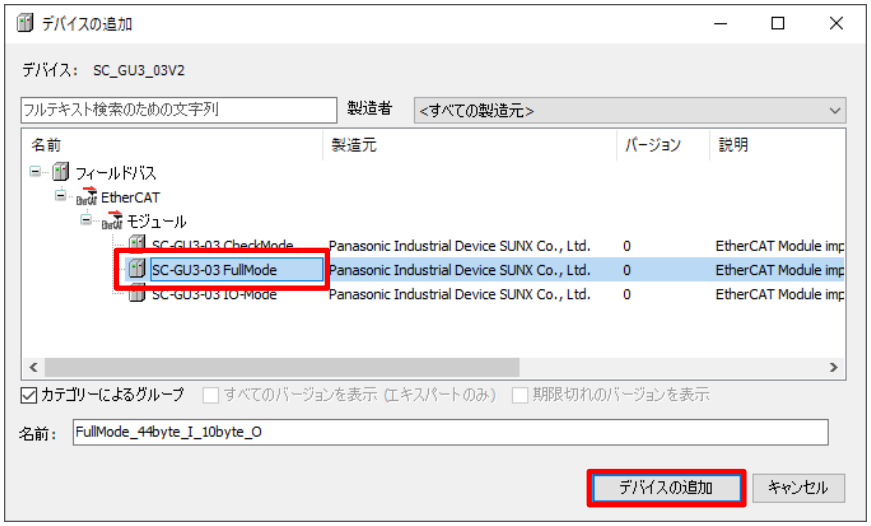
手順５

「SC-GU3-03V2」で右クリックし、「デバイスの追加」をクリックします。

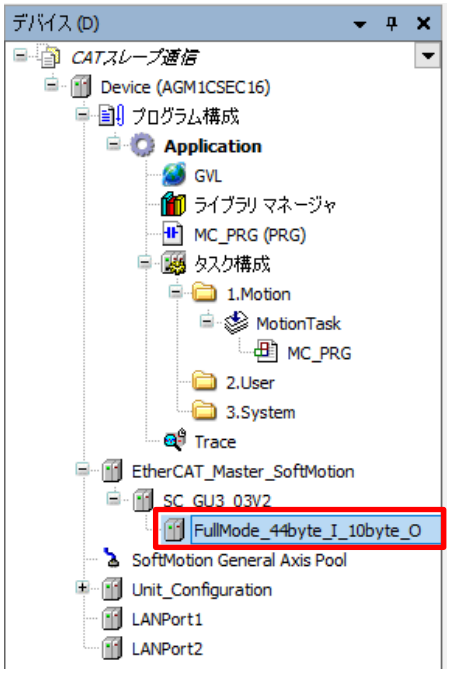


手順 6

手順 1 で DIP スイッチを「フルモード」に設定したので、「SC-GU3-03 FullMode」を選択し、「デバイスの追加」をクリックします。



「FullMode_44byte_I_10byte_O」が追加されました。



コラム⑨ Station ID の設定

SC-GU3-03 本体の前面に配置されているロータリ SW を使用する方法を説明します。
SDO 通信で 2001h／00h に「0」を書き込みます。

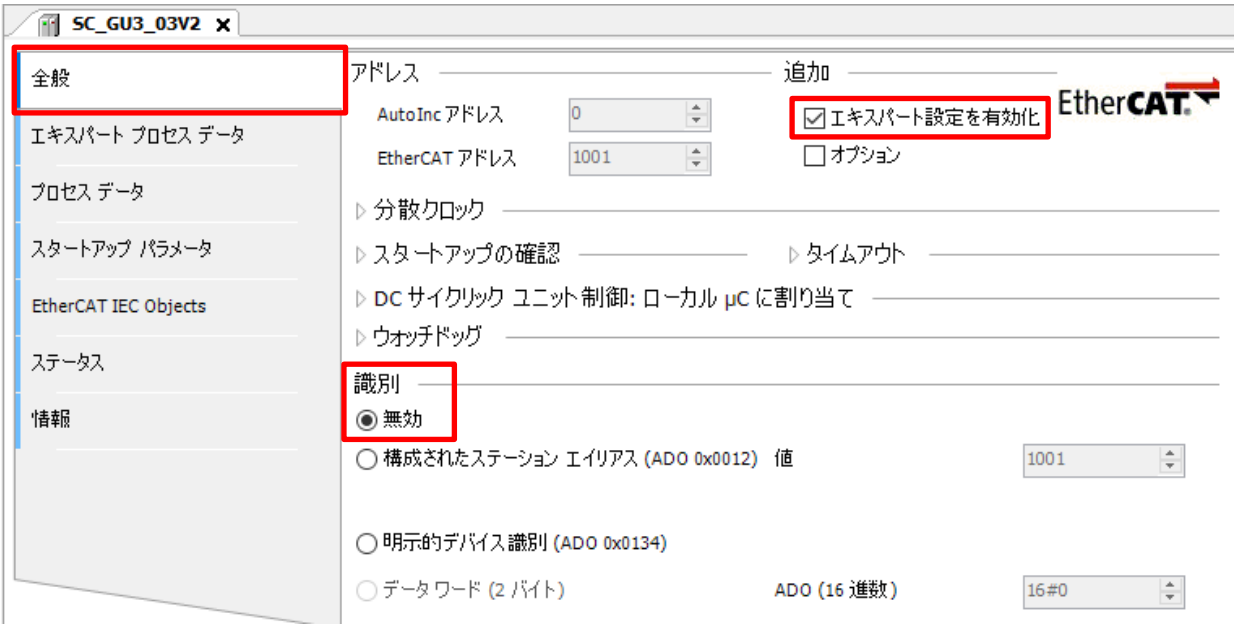
	Index	SubIndex	Name	Flag	機能
Manufacturer Specific Area	2001h	00h	R×PDO Station Alias setup（Hi）	RW	ステーションエイリアスの設定方法を選択します。（工場出荷状態 0001h） 0000h：ロータリスイッチ+「R×PDO Station Alias setup（Hi）」設定 0001h：SII EEPROM 設定 「1.6 ESC レジスタ 0012h～0013h（Configured Station Alias）設定」参照。

手順 1

GM1 コントローラと SC-GU3-03 を 1 対 1 で接続します。

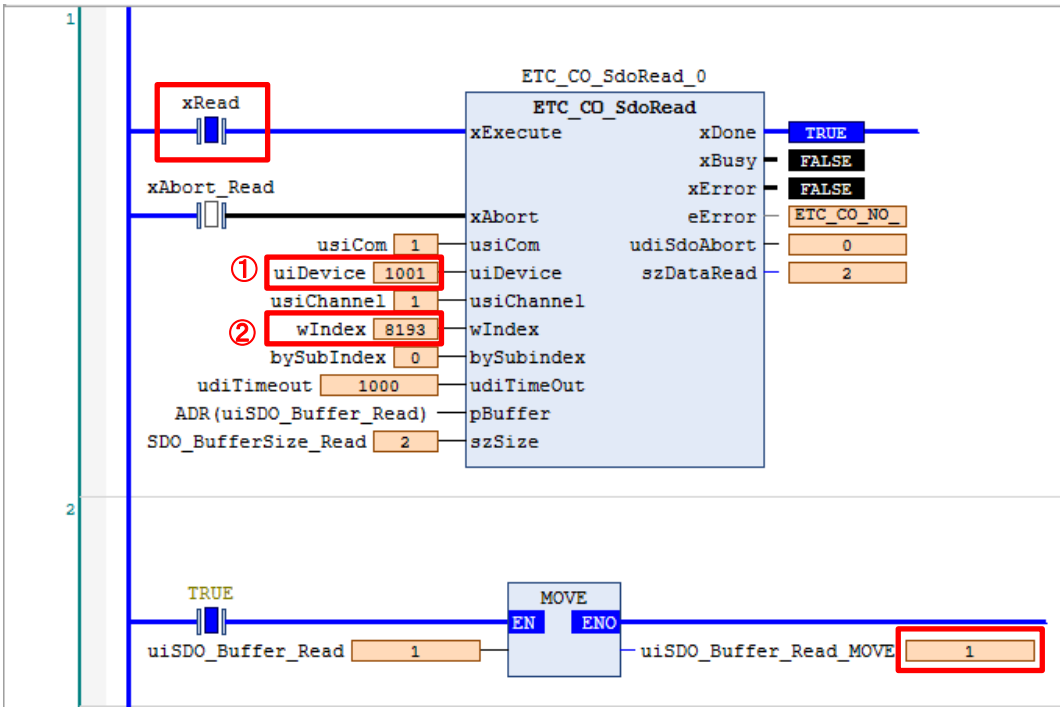
手順 2

SC-GU3-03 をダブルクリックし、「全般」タブを開きます。
「エキスパート設定を有効化」に☑を入れ、「識別」を「無効」にします。

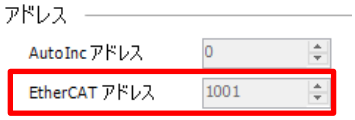


手順３

Sdo_Read で現在の値を読み出します。
下図のようにプログラムを作成し、「xRead」を TRUE に書き換えます。
「xDone」が TRUE になれば読み出し完了です。
「uiSDO_Buffer_Read_MOVE」が「1」になっていることを確認してください。



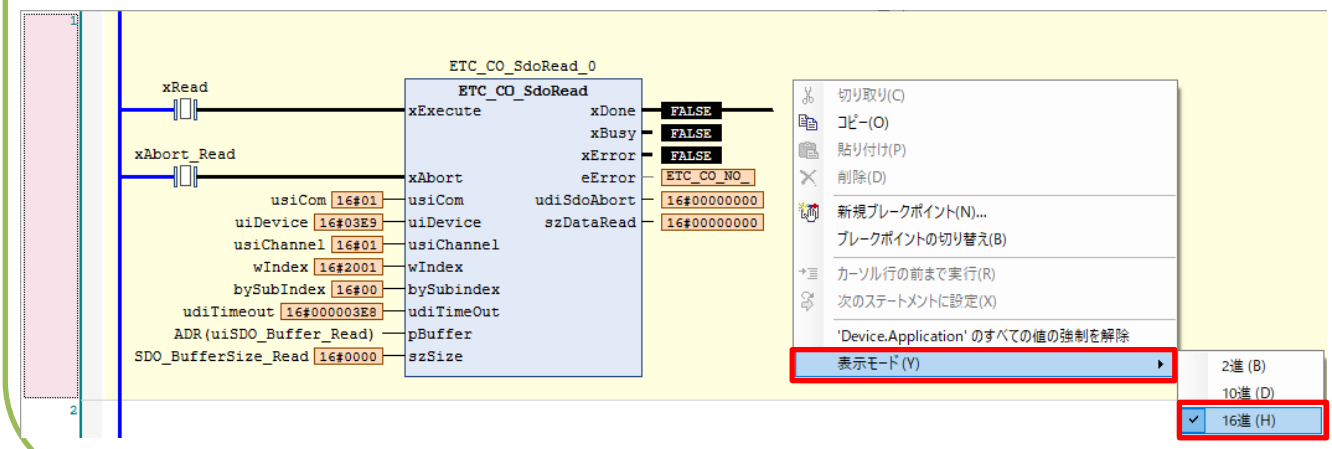
①「SC-GU3-03V2」をダブルクリックし、「全般」の「アドレス」－「EtherCAT アドレス」を入力します。
接続順に 1001、1002、1003 というように連番になります。



②R×PDO Station Alias setup（Hi）の Index 番号を入力します。
今回は「2001h」なので、16 進数で「8193」になります。

INFO

画面上で右クリックし、「表示モード」を選択すると、「2 進数／10 進数／16 進数」の表示を切り替えることができます。



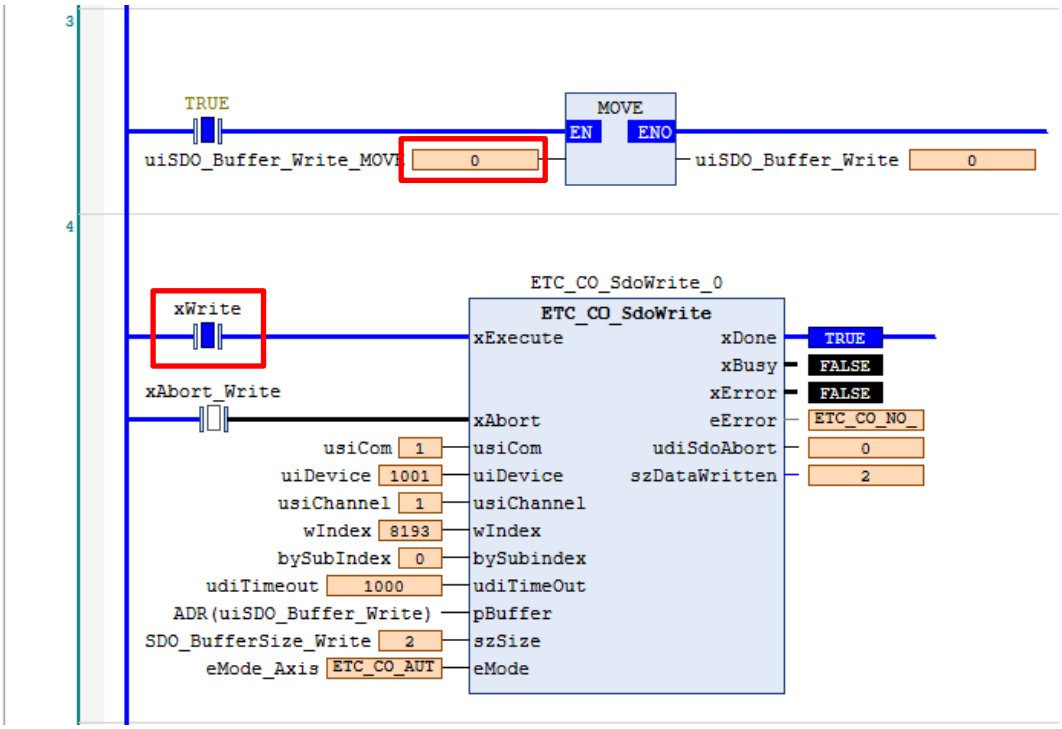
手順４

Sdo_Write でデータを書き込みます。

下図のようにプログラムを作成し、「uiSDO_Buffer_Write_MOVE」に書き込みたいデータを入力します。

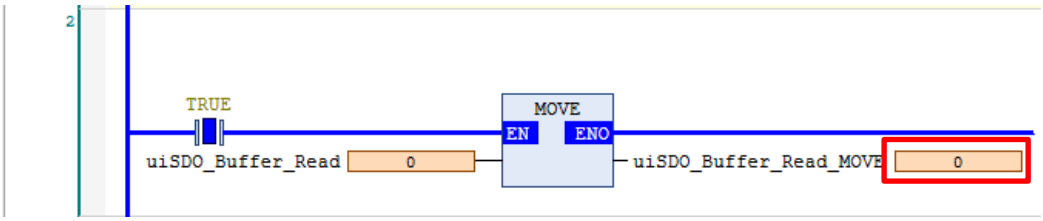
今回は「0」を入力し、「xWrite」を TRUE に書き換えます。

「xDone」が TRUE になれば、書き込み完了です。



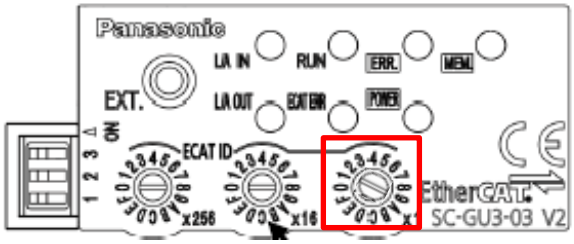
手順５

再度、手順３を実行し「uiSDO_Buffer_Read_MOVE」が「0」になっていることを確認してください。



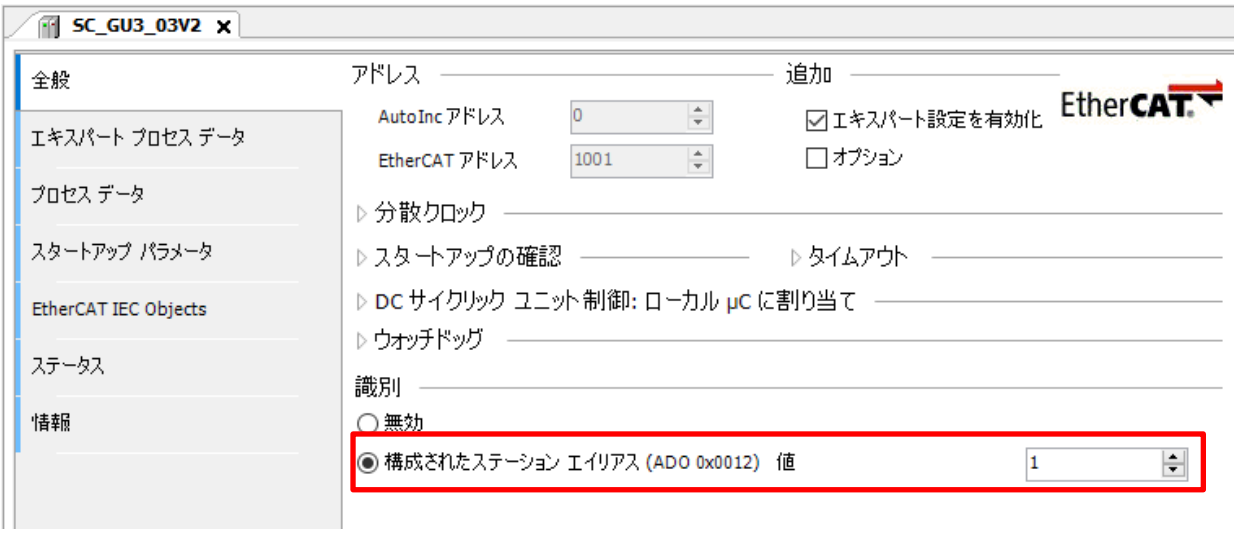
手順６

SC-GU3-03 のロータリ SW のアドレスを「1」に設定し、電源を再起動します。



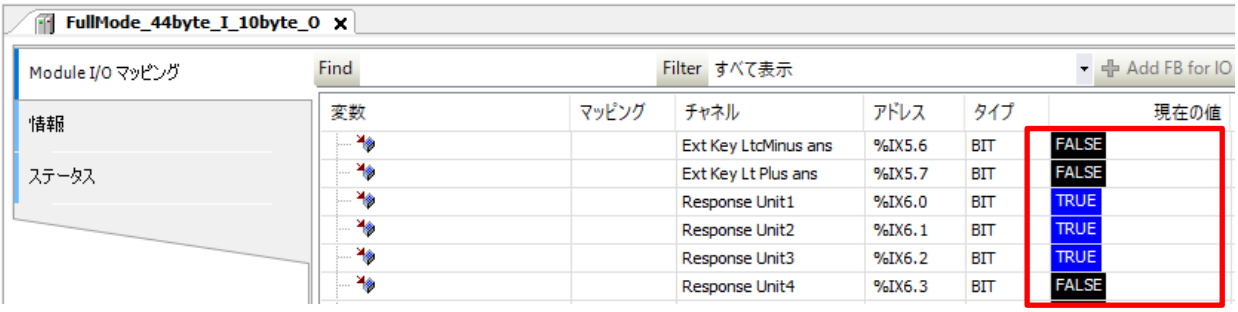
手順 7

SC-GU3-03 をダブルクリックし、「全般」タブを開きます。
「識別」を「構成されたステーションエイリアス（ADO 0×0012）値」を「1」にします。
GM1 コントローラにダウンロードします。



手順 8

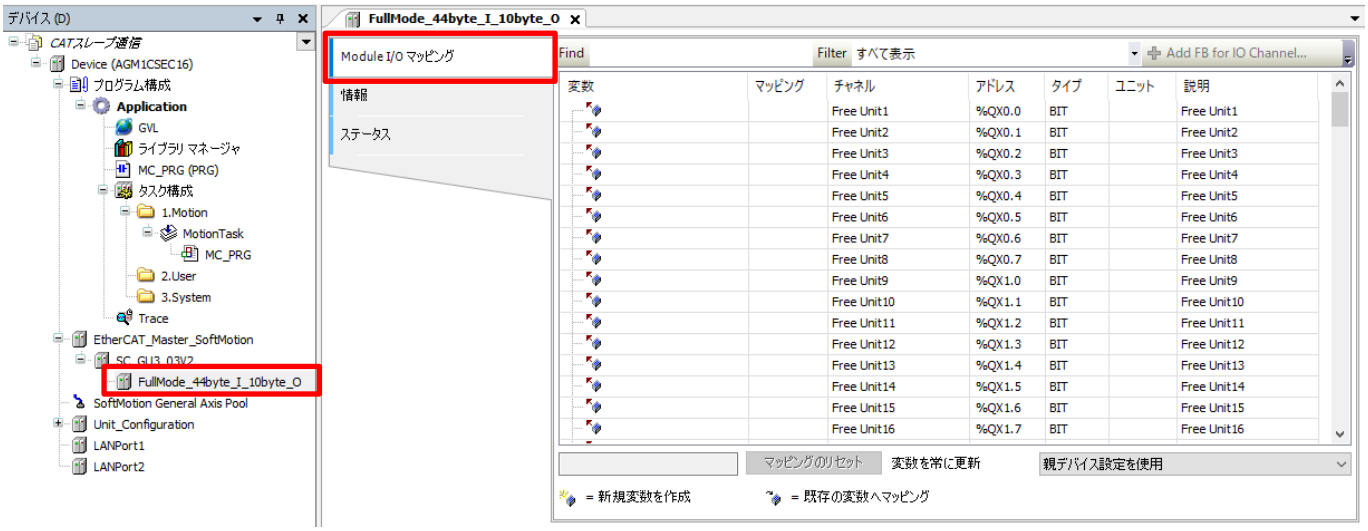
下図のように接続状態・出力確認ができれば、完了です。
（この例ではファイバセンサのアンプを 3 つ接続しているので Unit1～Unit3 が TRUE になっています）



2 GM1 設定

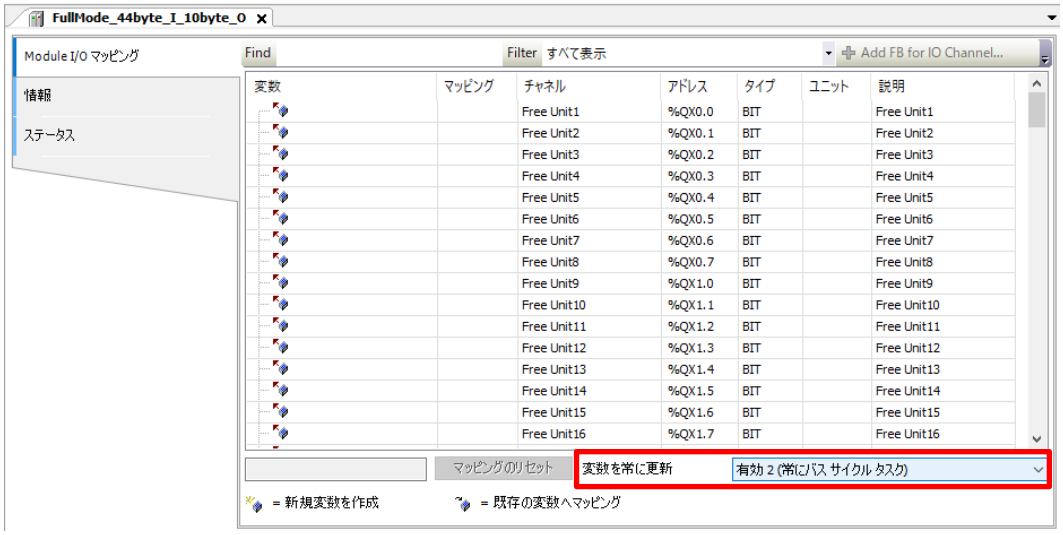
手順 1

追加された「FullMode_44byte_I_10byte_O」をダブルクリックします。
「ModuleI/O マッピング」を開きます。



手順 2

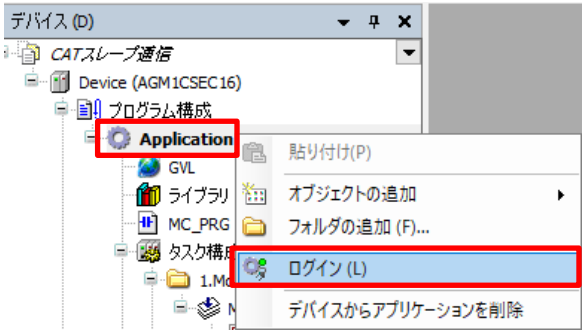
「変数を常に更新」を「有効 2(常にバス サイクル タスク)」に変更します。



3 接続確認

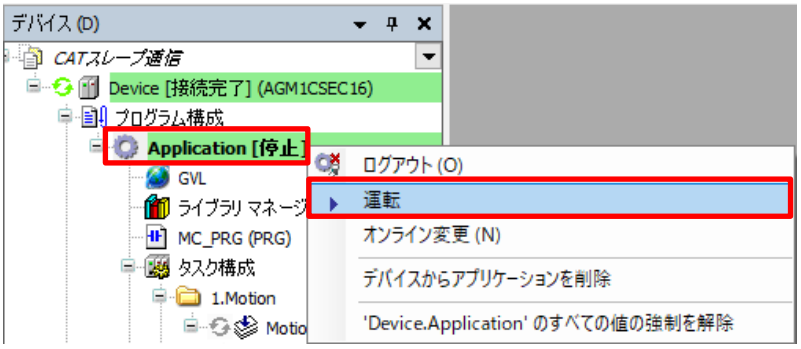
手順 1

「Application」を右クリックし、「ログイン」をクリックして、GM1 本体にダウンロードします。



手順 2

「Application[停止]」を右クリックし、「運転」をクリックして、停止→運転に切り替えます。

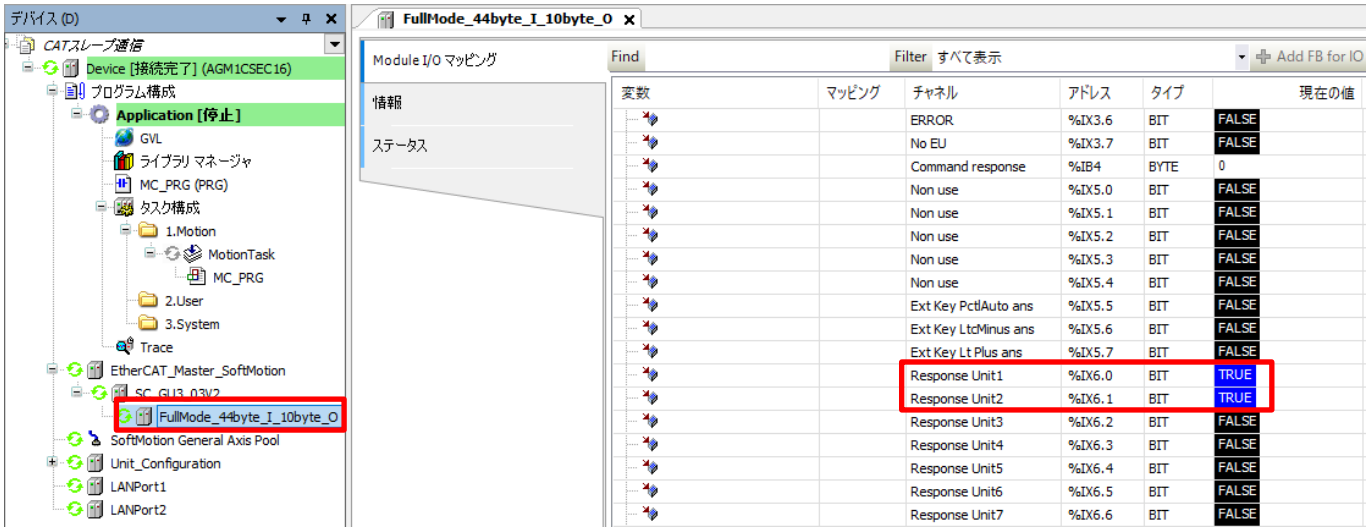


手順 3

「FullMode_44byte_I_10byte_O」をダブルクリックし、「ModuleI/O マッピング」を開きます。

テキストの例では、FX501 を 2 台接続しているので「Response Unit1」と「Response Unit2」が TRUE になっています。

これで接続できていることが確認できます。



Memo

改訂履歴

発行日付	マニュアル番号	改定内容
2022 年 4 月	AIM0010_01	初版

パナソニック インダストリー株式会社

〒574-0044 大阪府大東市諸福 7 丁目 1 番 1 号

このマニュアル記載内容は 2022 年 4 月現在のものです。