

マニュアル

**AI-HMI
外部拡張通信機器接続
(MODBUS・BarcodeReader 用)**

目次

第1章 本書について

第2章 各通信機器の通信仕様について

2-1 MODBUS 通信概要	2-1
2-2 バーコードリーダ通信概要	2-1

第3章 AI-HMI 上の設定について

3-1 MODBUS 機器通信設定	3-1
3-2 バーコードリーダ通信設定	3-9

第4章 外部拡張通信機器設定リファレンス

4-1 機能概要	4-1
4-2 ExtBusCommSetting	4-3
4-3 MODBUSSetting	4-6
4-4 BarcodeReaderSetting	4-10

第1章 本書について

本書はMODBUS機器またはバーコードリーダーとAI-HMIを接続する際の設定方法について説明したものです。AI-HMIの基本的な使用方法や設定については以下のマニュアルをご参照ください。

- ・ AI-HMI ユーザーズマニュアル
- ・ AI-HMI リファレンスマニュアル

本書および AI-HMI 関連のマニュアルでは変数型の表記として一般的なプログラミングで使用されている型名で記載しています。

これらの型名と一般的に PLC で使われているデータ型名の関連は以下のようになります。

図 1-1 データ型名対応表

マニュアル記載の型名	PLC 向け型名
BOOL	BIT (0 ~ 1)
INT	(対応データ型なし) (-32768 ~ 32767)
UINT	WORD (0 ~ 65535)
DINT	(対応データ型なし) (-2147483648 ~ 2147483647)
UDINT	DWORD (0 ~ 4294967295)
REAL	FLOAT (3.4E +/- 38)
LREAL	(対応データ型なし) (1.7E +/- 308)
STRING	文字列

第 2 章 各通信機器の通信仕様について

本章では各通信機器の通信仕様について説明します。

2-1 MODBUS 通信概要

AI-HMI では、MODBUS 通信機器とシリアルポートまたはネットワークポート経由で通信することが可能です。

Modbus Protocol の通信方式は、シングルマスター/マルチスレーブ方式です。マスターだけがクエリー（通信の開始）を発行することができます。スレーブは、このクエリーを見て、指定された機能を実行し、応答メッセージを返します。マスターは、指定のスレーブに対するクエリーまたはすべてのスレーブに対するブロードキャストクエリーのいずれかを発行することができます。ブロードキャストクエリーの場合には、スレーブは、指定の機能を実行するのみで応答メッセージは返しません。スレーブは、自分に対するクエリーのときにだけ応答メッセージを返します。

サポートしている MODBUS 通信は以下の 3 種類です。

- ◆MODBUS-RTU（シリアル通信）
- ◆MODBUS-ASCII（シリアル通信）
- ◆MODBUS-TCP（ネットワーク通信）

シリアル通信モードには ASCII（American Standard Code for Information Interchange）モードと RTU（Remote Terminal Unit）モードの 2 種類があり、選択することができます。ただし、1つのネットワーク上では、全てのデバイスが同一モードでなくてはなりません。ASCII モードでは、1バイト（8ビット）データを 2文字の ASCII コードに変換して伝送します。RTU モードでは、1バイト（8ビット）データをそのまま伝送します。従って ASCII モードより伝送効率が良いと言えます。

エラーチェックフィールドのチェックアルゴリズムは伝送モードによって異なります。ASCII モードの場合には、LRC（Longitudinal Redundancy Check）法が、また RTU モードの場合には、CRC（Cyclical Redundancy Check）法が採用されています。

MODBUS 通信には、TCP/IP を採用した MODBUS-TCP という通信方式があります。TCP/IP を使用していますのでインターネット環境でもメッセージのやり取りが可能になります。Modbus-TCP では、クライアント・サーバモデルで通信を行なうことになります。クライアントは従来の Modbus プロトコルでのマスターに相当し、サーバはスレーブに対応します。TCP/IP プロトコルを基本にしていますので、複数クライアント、複数サーバのサポートが可能になります。即ちマルチマスター、マルチスレーブのシステム構成が可能になります。

2-2 バーコードリーダ通信概要

AI-HMI では、バーコードリーダとシリアルポート経由で、1:1 で通信することが可能です。バーコードリーダの通信伝文は一般的に<プレフィックス><バーコードデータ><サフィックス>となっています。プレフィックスデータとバーコードデータ、サフィックスデータをそれぞれ、サイズと格納場所を指定してデータを受信することが出来ます。

第3章 AI-HMI 上の設定について

本章では AI-HMI と MODBUS 機器やバーコードリーダ等の外部機器との通信をするための設定について説明します。

3-1 MODBUS 機器通信設定

3-1-1 通信ポートの設定

- ① MODBUS 機器と通信を行うためには、通信設定を行う必要があります。

VisualStudio で AI-HMI プロジェクトを開き、「ソリューションエクスプローラー」から「MainWindow.xaml」をダブルクリックしてください。

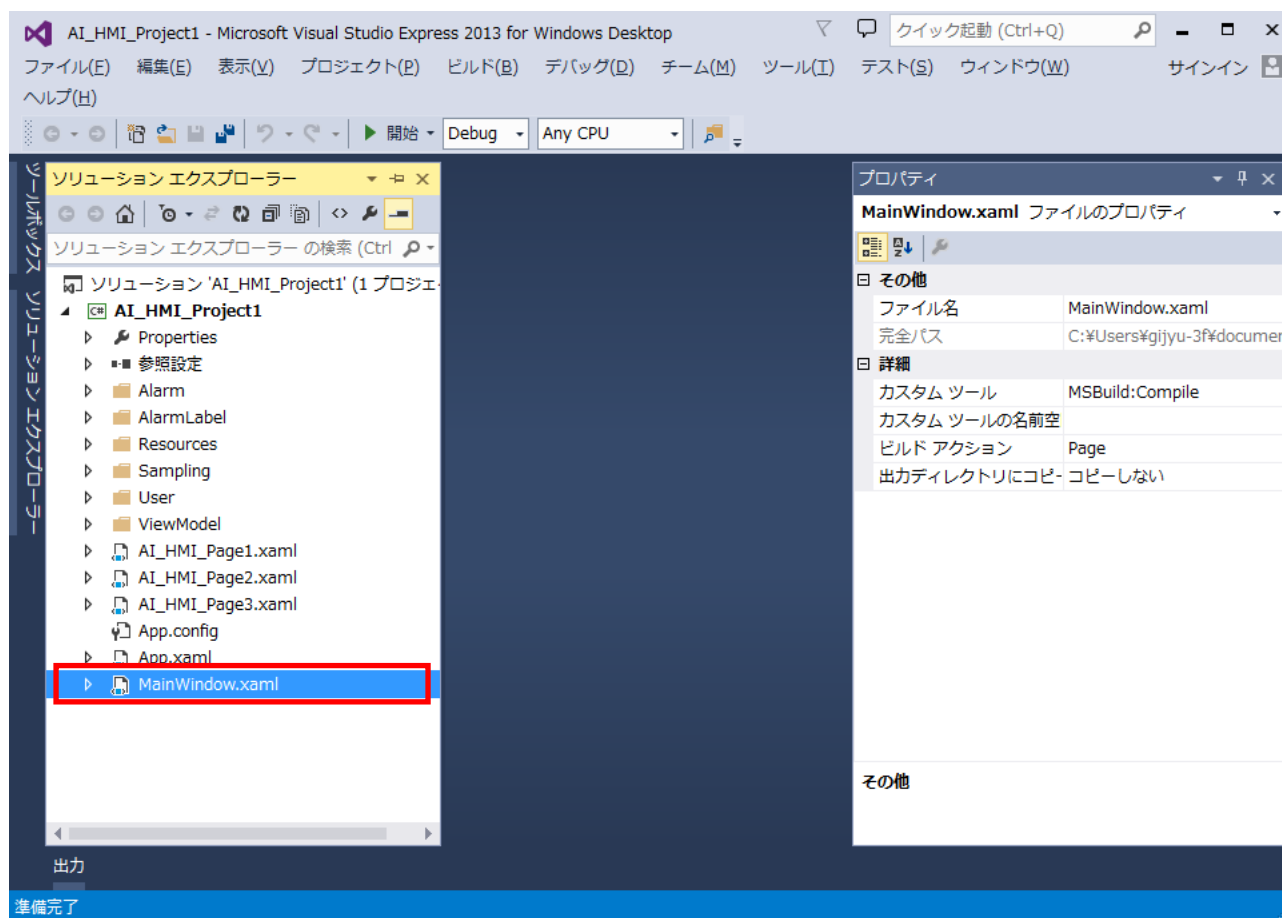


図 3-1-1-1 MainWindow.xaml の表示

- ② MainWindow 画面が開きます。
「ドキュメントアウトライン」から「AI_HMI_Window」を選択し、「プロパティ」の「AI HMI Ext Bus Comm Setting」を選択してください。

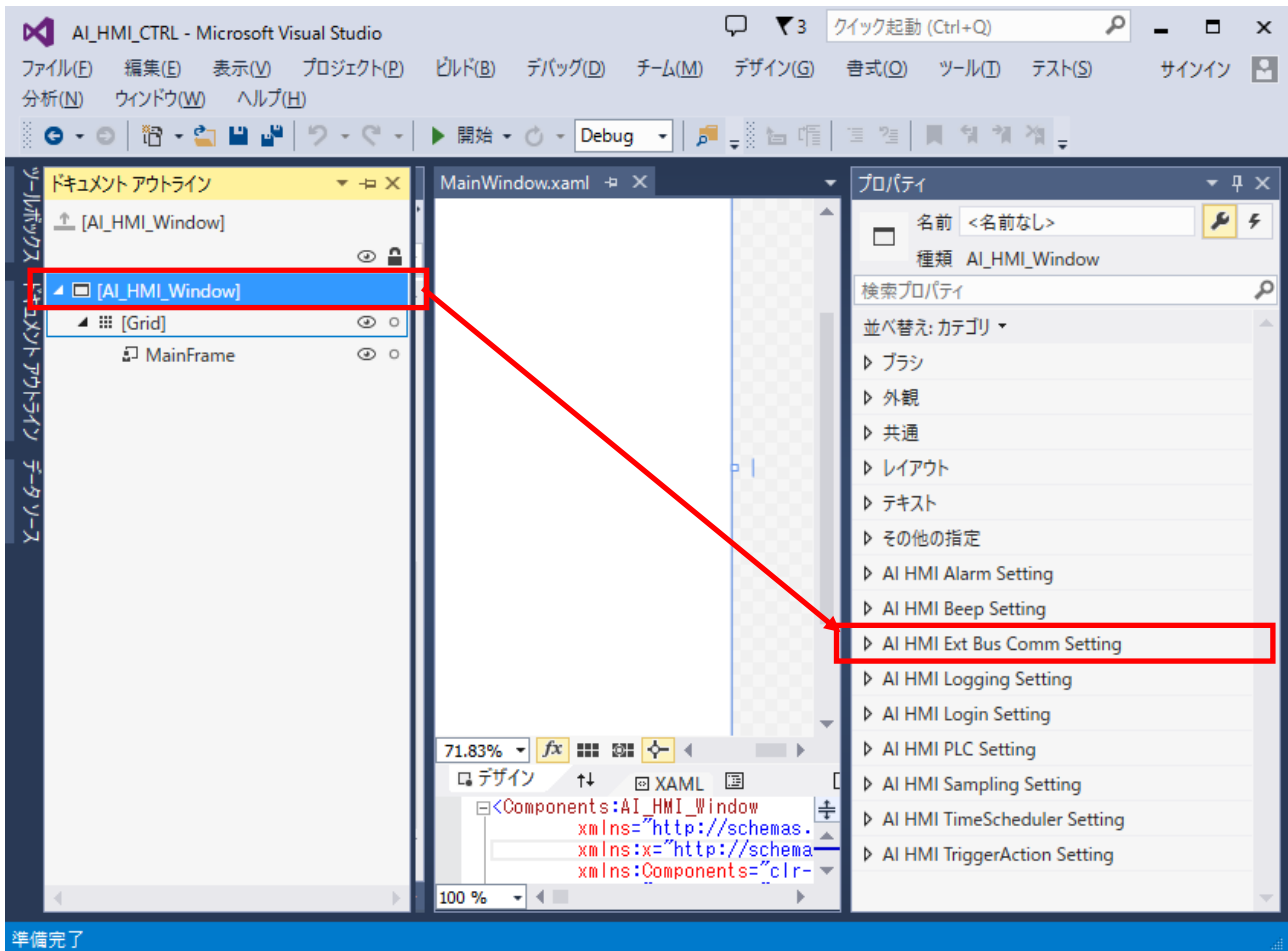


図 3-1-1-2 AI HMI Ext Bus Comm Setting の選択

- ③ AI HMI Ext Bus Comm Setting の設定項目が表示されます。
「AI HMI Ext Bus Comm Setting - ExtBusCommSetting」のボタンをクリックしてください。

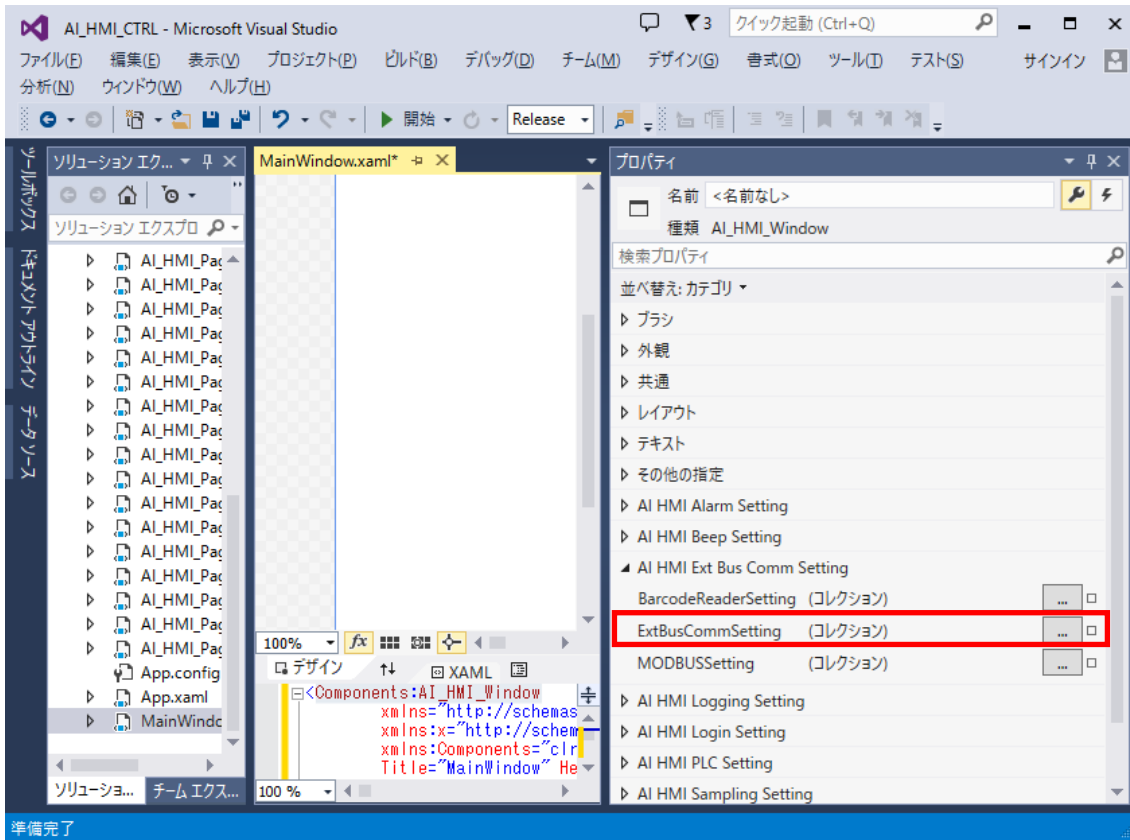


図 3-1-1-3 ExtBusCommSetting の設定

- ④ ExtBusCommSetting のコレクションエディタが開きます。「追加」ボタンをクリックしてください。

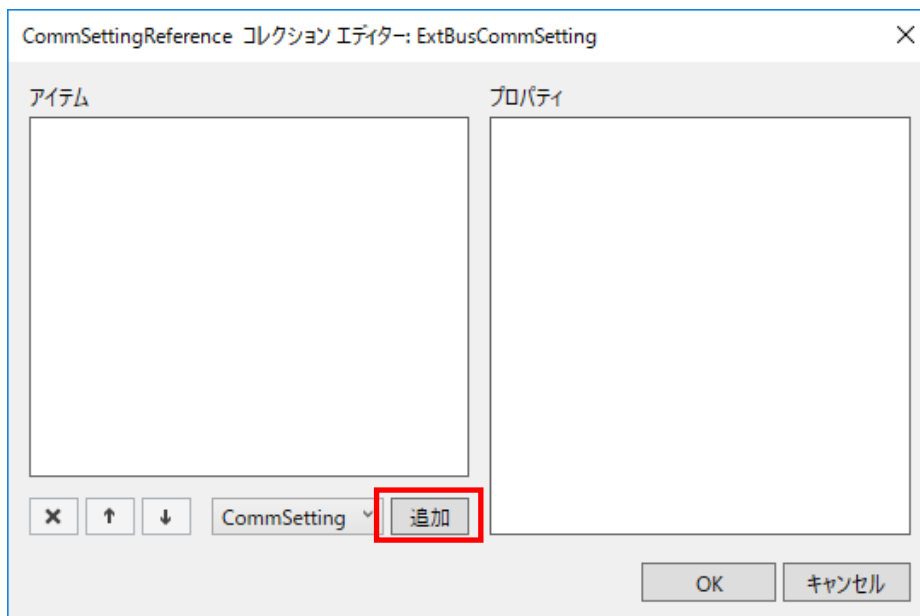


図 3-1-1-4 CommSettingReference の変更

- ⑤ CommSettingReference のプロパティが 1 項目追加されます。
表 3-1-1-1 のように設定してください。
設定後、「OK」ボタンをクリックしてください。

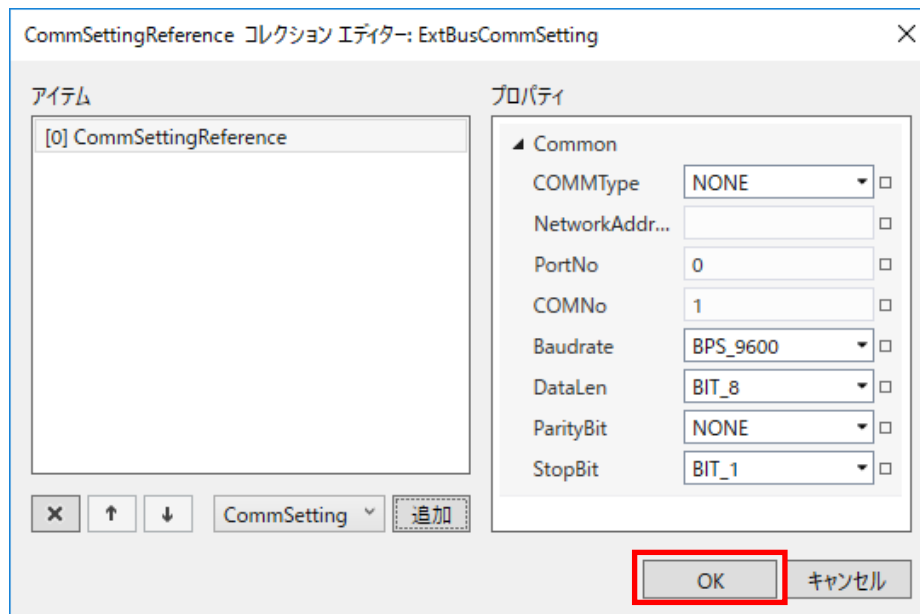


図 3-1-1-5 CommSettingReference の変更

表 3-1-1-1 CommSettingReference のプロパティ

項目	設定値	備考
COMMType	MODBUS_RTU	MODBUS 機器とシリアルポートの RTU モードで通信します。
NetworkAddress	なし※	COMMType が MODBUS_TCPIP の場合、通信する MODBUS 機器の IP アドレスを指定します。
PortNo	0※	COMMType が MODBUS_TCPIP の場合、ポート番号を設定します。
COMNo	1	MODBUS 機器と接続しているシリアルポート番号を指定します。COM1 の場合 1 と設定します。
Baudrate	BPS_9600※	シリアル通信ボーレートを設定します。
DataLen	BIT_8※	シリアル通信データ長を設定します。
ParityBit	NONE※	シリアル通信パリティビットの設定をします。
StopBit	BIT_1※	シリアル通信ストップビット長の設定をします。

※ 通信する機器の通信設定に合わせる必要があります。

3-1-2 MODBUS データを共有メモリに展開する設定、共有メモリから書き込みデータを取得する設定

- ① 共有メモリ上に展開する MODBUS コマンドと、アドレス、データサイズを設定します。

先項の MainWindow.xaml の画面のまま設定を行います。

「AI HMI Ext Bus Comm Setting - MODBUSSetting」のボタンをクリックしてください。

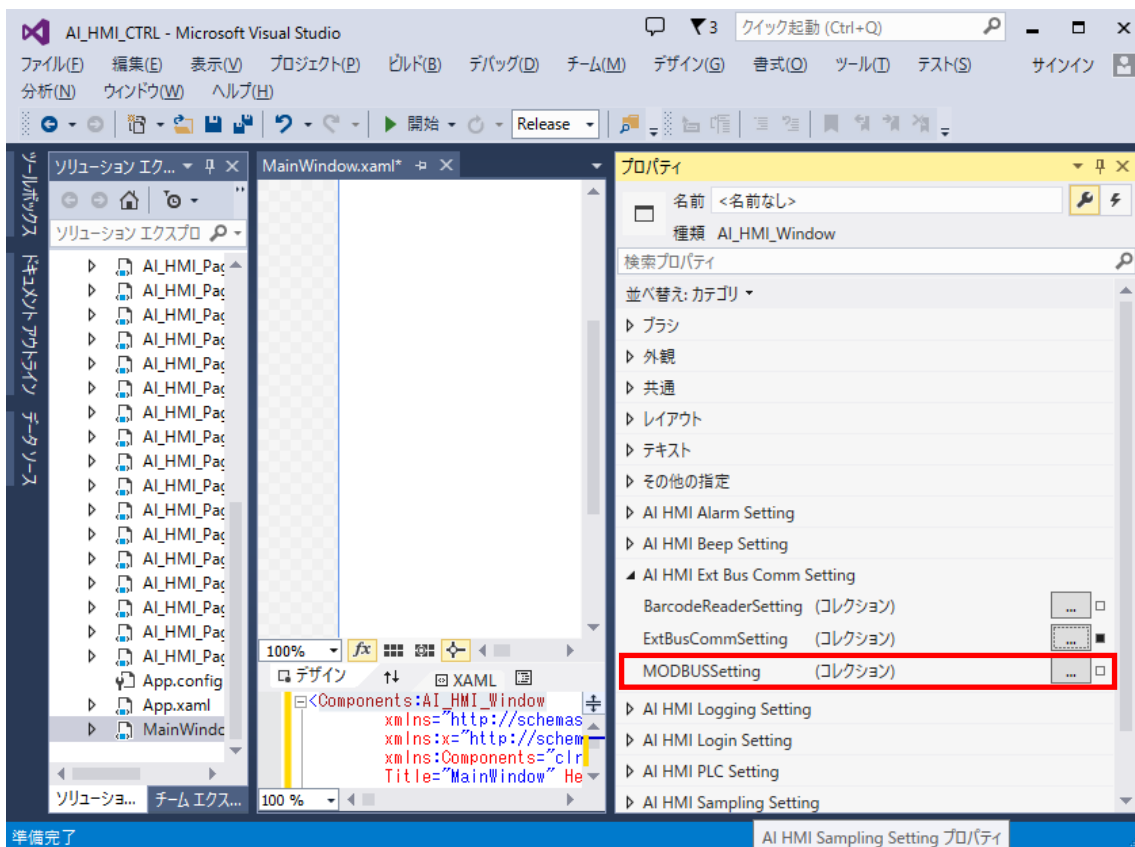


図 3-1-2-1 MODBUSSetting の設定

- ② MODBUSSetting のコレクションエディタが開きます。「追加」ボタンをクリックしてください。

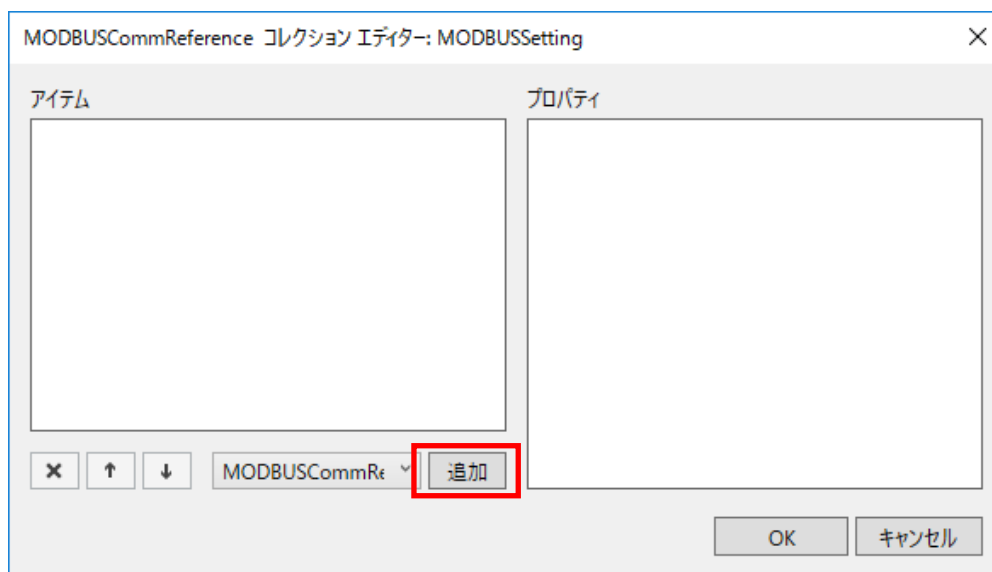


図 3-1-2-2 MODBUSCommReference の変更

- ③ MODBUSCommReference のプロパティが 1 項目追加されます。
表 3-1-2-1 のように設定してください。

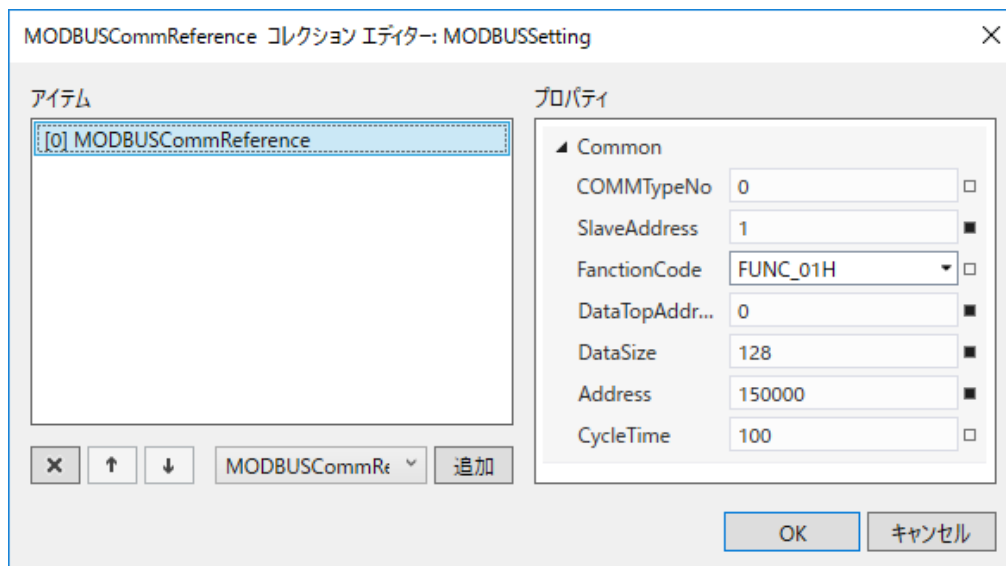


図 3-1-2-3 MODBUSCommReference[0]の変更

表 3-1-2-1 MODBUSCommReference [0]のプロパティ

項目	設定値	備考
COMMTypeNo	0	CommSettingReference で登録した通信設定の配列番号を指定します。
SlaveAddress	1	MODBUS 機器のスレーブ番号を設定します。
FunctionCode	FUNC_01H	コイルのデータを読み出す場合、FUNC_01H を設定します。
DataTopAddress	0	読み出すコイルデータの先頭アドレスを指定します。
DataSize	128	読み出すコイルデータサイズを指定します。こちらの指定は bit 単位です。読み出しデータは、DataSize / 8 + 1 (8 で割り切れない場合) バイトのデータとして読み出せます。
Address	150000	読み出したデータを格納する共有メモリの先頭アドレスを指定します。
CycleTime	100	読み出す周期を 10[ms]単位で指定します。100 だと 1 秒周期で実行します。

MODBUSCommReference の設定の詳細については『4-3 MODBUSCommReference』を参照してください。
共有メモリについての詳細は『AI-HMI リファレンスマニュアル』を参照してください。

- ④ 次に、共有メモリから MODBUS 機器にデータを書き込む設定を行います。「追加」ボタンをクリックしてください。

MODBUSCommReference のプロパティが更に 1 項目追加されます。

表 3-1-2-2 のように設定してください。

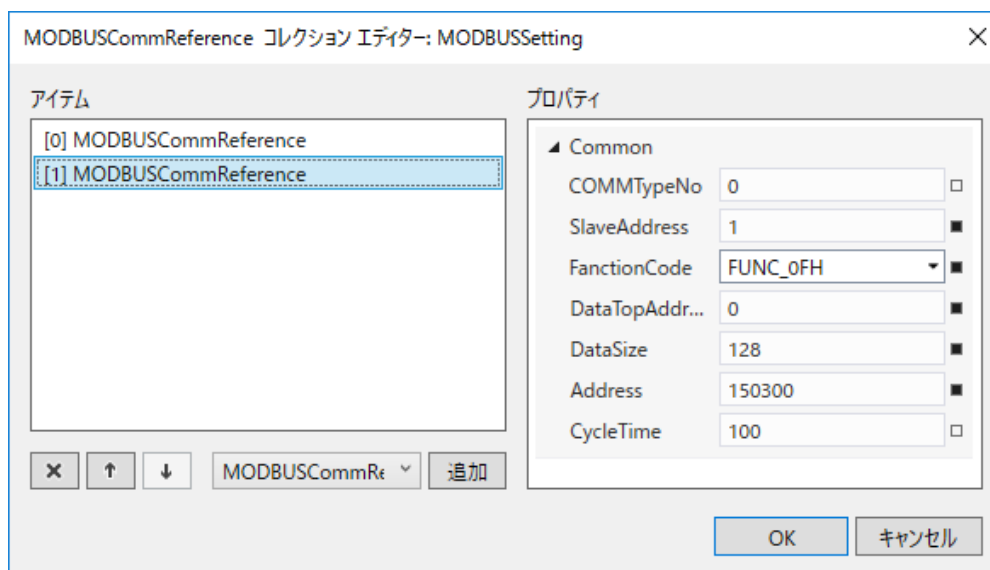


図 3-1-2-4 MODBUSCommReference [1]の変更

表 3-1-2-2 MODBUSCommReference [1]のプロパティ

項目	設定値	備考
COMMTypeNo	0	CommSettingReference で登録した通信設定の配列番号を指定します。
SlaveAddress	1	MODBUS 機器のスレーブ番号を設定します。
FunctionCode	FUNC_OFH	コイルのデータを書き込む場合、FUNC_OFH を設定します。
DataTopAddress	0	書き込むコイルデータの先頭アドレスを指定します。
DataSize	128	書き込むコイルデータサイズを指定します。こちらの指定は bit 単位です。書き込みデータは、DataSize / 8 + 1 (8 で割り切れない場合) バイトのデータとして書き込まれます。
Address	150300	書き込みするデータを格納する共有メモリの先頭アドレスを指定します。
CycleTime	100	読み出す周期を 10[ms]単位で指定します。100 だと 1 秒周期で実行します。

- ⑤ 実際にアプリケーションを実行し、設定どおりに動作するかを確認します。
アプリケーションを実行するための方法はAI-HMI ユーザーズマニュアルの『第6章 作成したアプリの実行環境 PC での動作』を参照してください。
共有メモリへの展開が正常に動作しているかはAI-HMI デバッグツールを使用して確認します。
AI-HMI デバッグツールの使用方法についてはAI-HMI ユーザーズマニュアルの『第7章 AI-HMI デバッグツール』を参照してください。



図 3-1-2-5 AI-HMI デバッグツールでの確認

共有メモリの 150300 に書き込んだデータが、150000 のアドレスに反映されていることを確認してください。

3-2 バーコードリーダ通信設定

3-2-1 通信ポートの設定

- ⑥ バーコードリーダと通信を行うためには、通信設定を行う必要があります。

VisualStudioでAI-HMIプロジェクトを開き、「ソリューションエクスプローラー」から「MainWindow.xaml」をダブルクリックしてください。

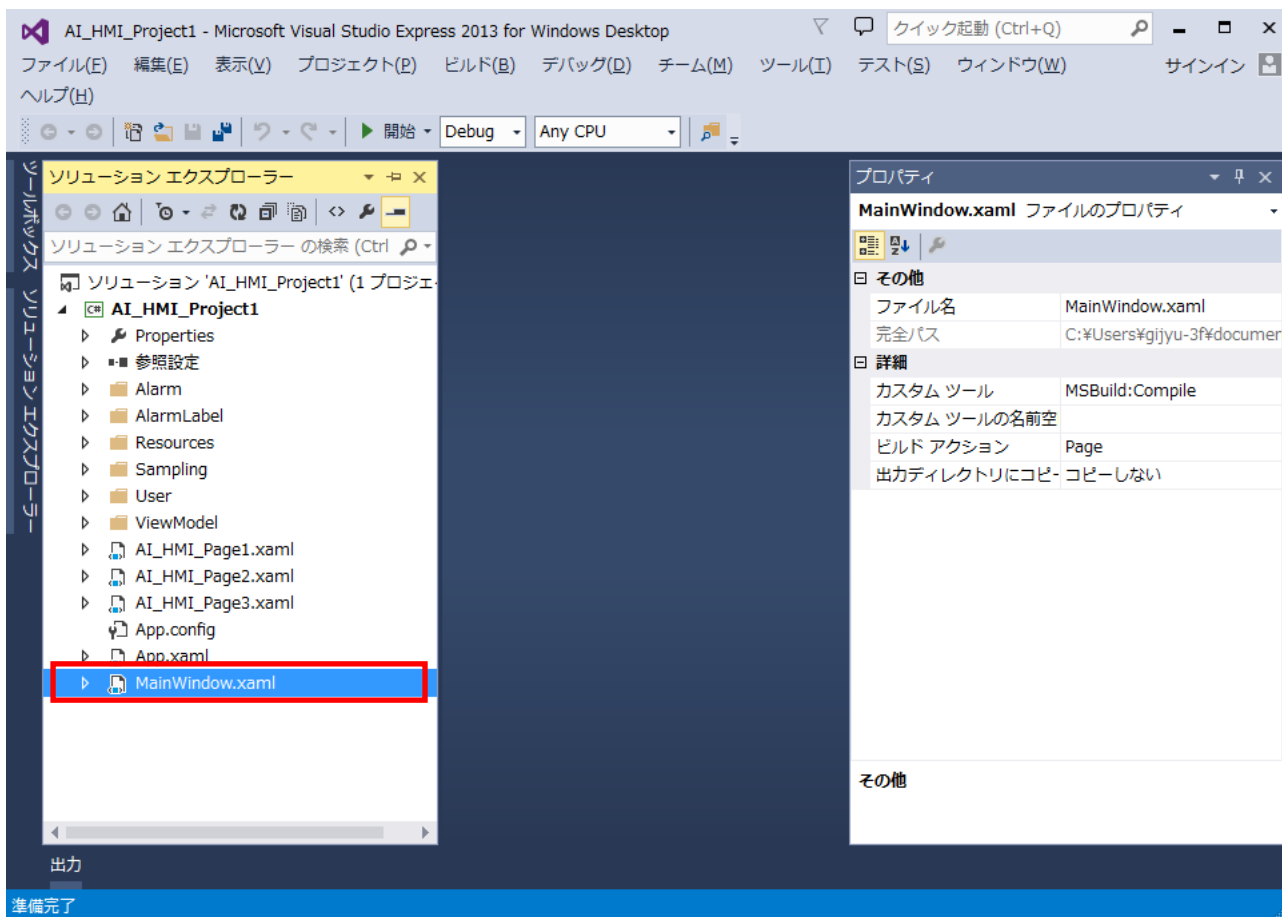


図 3-2-1-1 MainWindow.xaml の表示

- ⑦ MainWindow 画面が開きます。
「ドキュメントアウトライン」から「AI_HMI_Window」を選択し、「プロパティ」の「AI HMI Ext Bus Comm Setting」を選択してください。

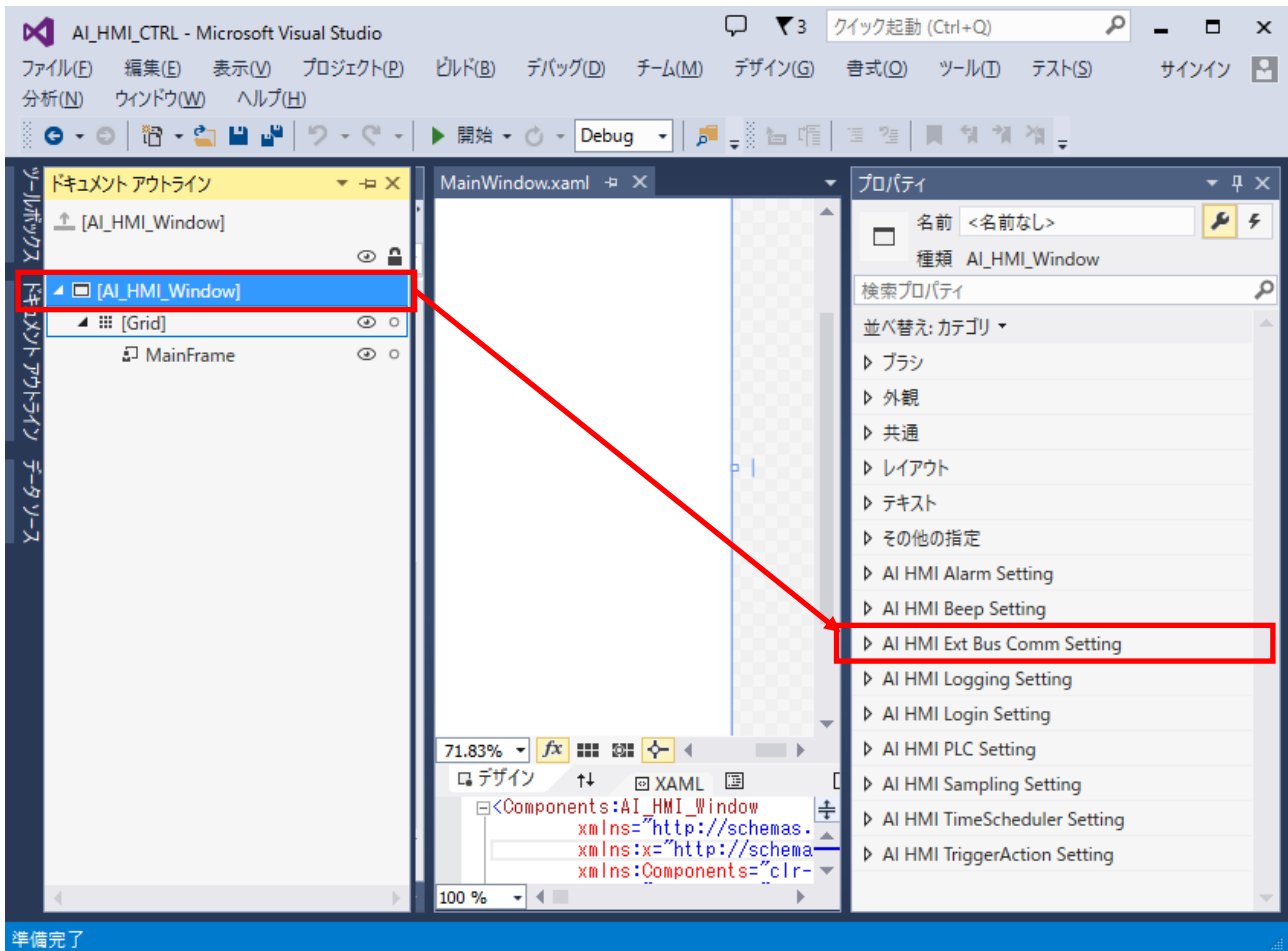


図 3-2-1-2 AI HMI Ext Bus Comm Setting の選択

- ⑧ AI HMI Ext Bus Comm Setting の設定項目が表示されます。
「AI HMI Ext Bus Comm Setting - ExtBusCommSetting」のボタンをクリックしてください。

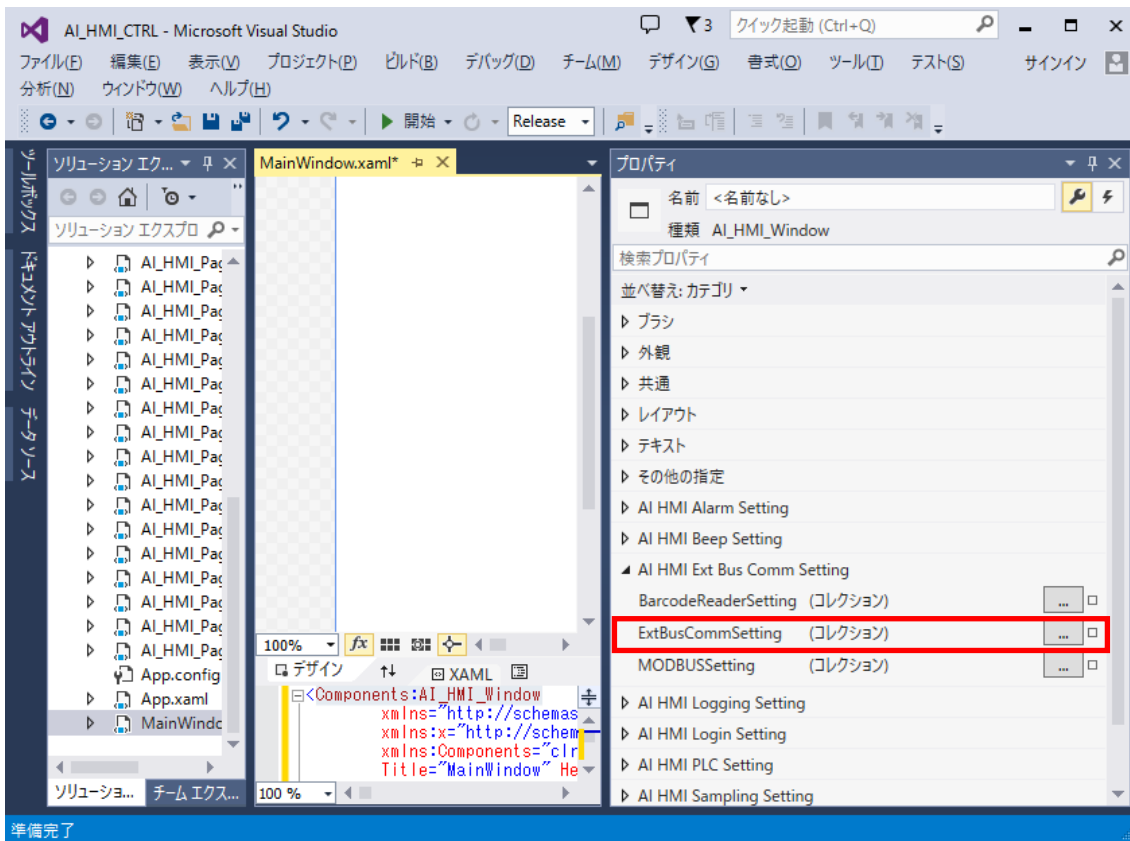


図 3-2-1-3 ExtBusCommSetting の設定

- ⑨ ExtBusCommSetting のコレクションエディタが開きます。「追加」ボタンをクリックしてください。

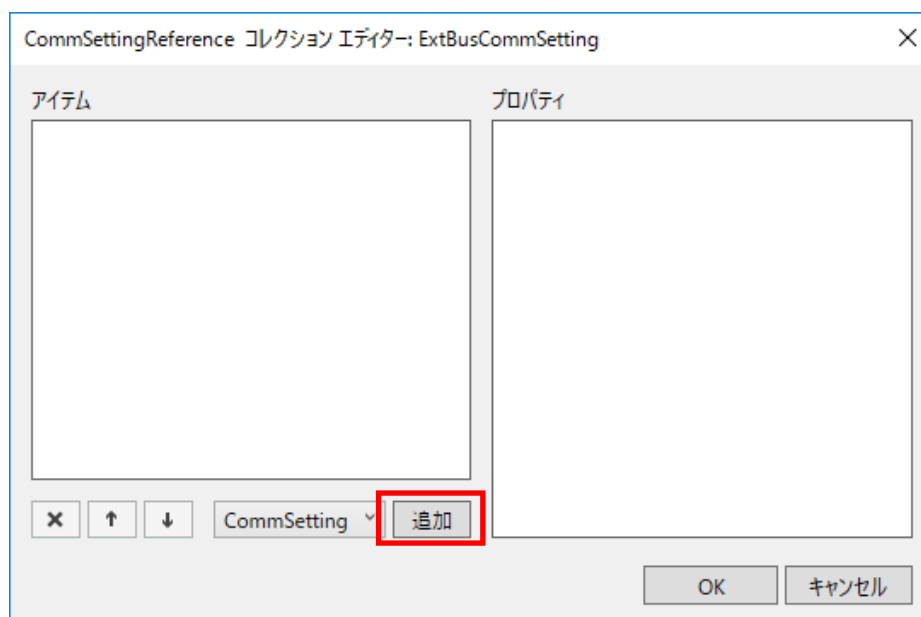


図 3-2-1-4 CommSettingReference の変更

- ⑩ CommSettingReference のプロパティが 1 項目追加されます。
表 3-2-1-1 のように設定してください。
設定後、「OK」ボタンをクリックしてください。

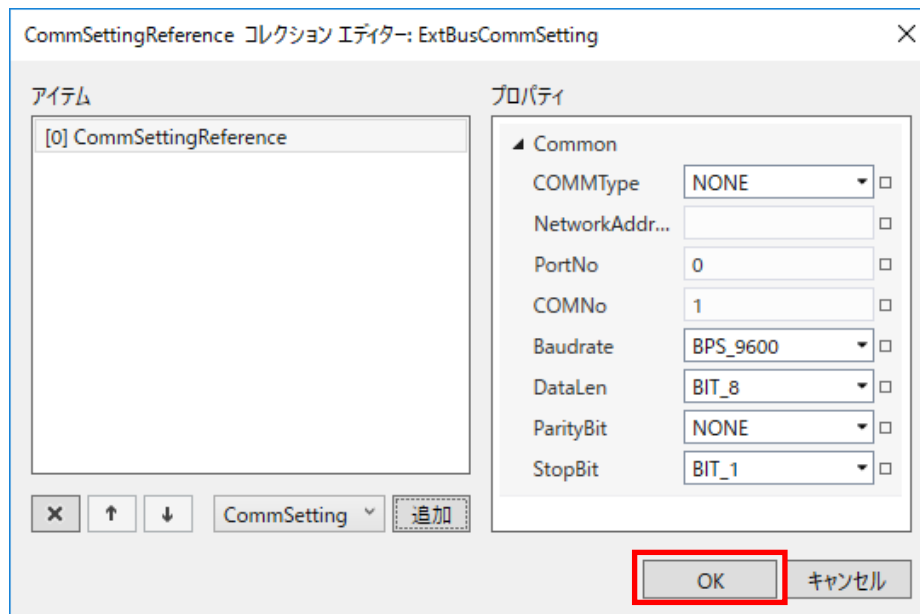


図 3-2-1-5 CommSettingReference の変更

表 3-2-1-1 CommSettingReference のプロパティ

項目	設定値	備考
COMMTYPE	BarcodeReader	バーコードリーダーとシリアルポートで通信します。
NetworkAddress	なし	バーコードリーダーでは TCP/IP モードでは通信することができません。そのため、この設定は無効です。
PortNo	0	
COMNo	1	バーコードリーダーと接続しているシリアルポート番号を指定します。COM1 の場合 1 と設定します。
Baudrate	BPS_9600※	シリアル通信ボーレートを設定します。
DataLen	BIT_8※	シリアル通信データ長を設定します。
ParityBit	NONE※	シリアル通信パリティビットの設定をします。
StopBit	BIT_1※	シリアル通信ストップビット長の設定をします。

※ 通信する機器の通信設定に合わせる必要があります。

3-2-2 バーコードリーダーから読み出すフォーマットの設定と展開する共有メモリアドレスの設定

- ① 共有メモリ上に展開するバーコードリーダーのフォーマットを設定します。

先項の MainWindow.xaml の画面のまま設定を行います。

「AI HMI Ext Bus Comm Setting - BarcodeReaderSetting」のボタンをクリックしてください。

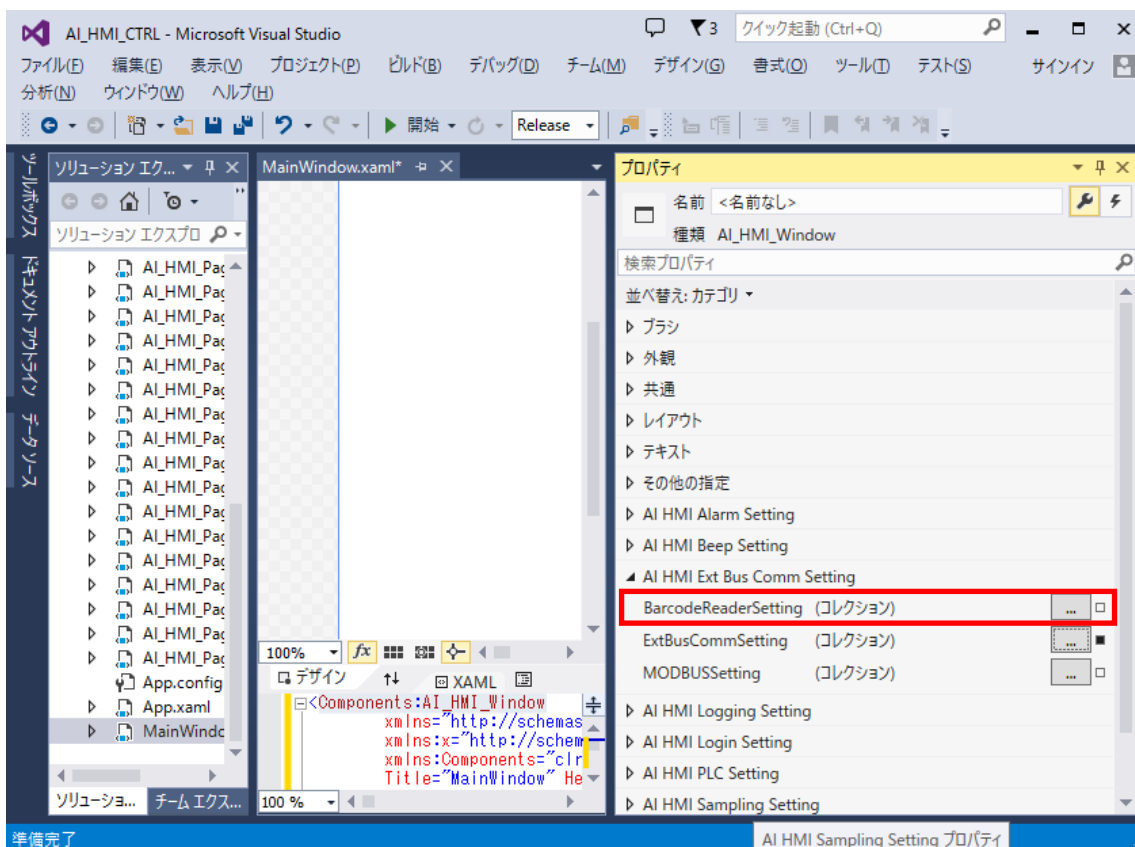


図 3-2-2-1 BarcodeReaderSetting の設定

- ② BarcodeReaderSetting のコレクションエディタが開きます。「追加」ボタンをクリックしてください。

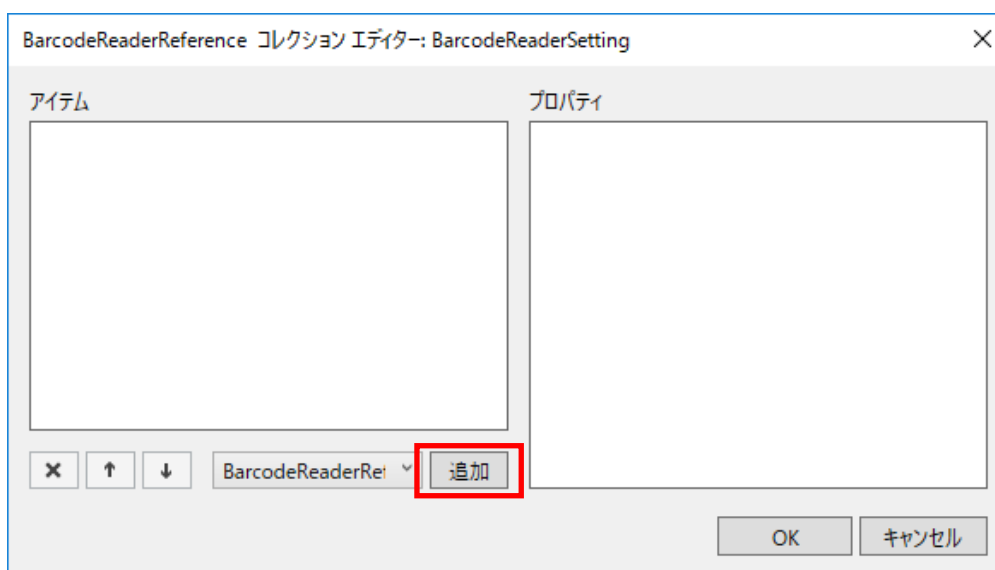


図 3-2-2-2 BarcodeReaderReference の変更

- ③ BarcodeReaderReference のプロパティが1項目追加されます。
表 3-2-2-1 のように設定してください。

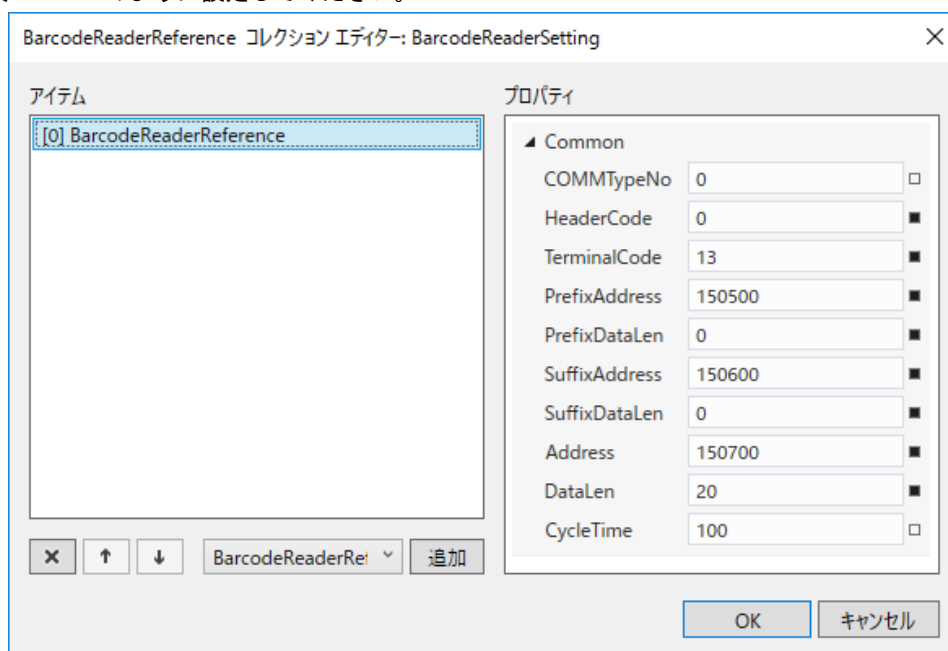


図 3-2-2-3 MODBUSCommReference [0] の変更

表 3-2-2-1 BarcodeReaderReference [0] のプロパティ

項目	設定値	備考
COMMTyNo	0	CommSettingReference で登録した通信設定の配列番号を指定します。
HeaderCode※	0	通信フォーマットの先頭コードを設定します。先頭コードがない場合は0を設定します。
TerminalCode※	13	通信フォーマットの終端コードを設定します。 「CR」の場合 10 進数で 13 を設定します。
PrefixAddress	150500	プレフィックス文字列を格納する共有メモリ先頭アドレスを指定します。
PrefixDataSize※	0	プレフィックスデータの文字数を指定します。HeaderCode を含んだサイズです。
SuffixAddress	150600	サフィックス文字列を格納する共有メモリ先頭アドレスを指定します。
SuffixDataSize※	1	サフィックスデータの文字数を指定します。TerminalCode を含んだサイズです。
Address	150700	読み出したデータを格納する共有メモリの先頭アドレスを指定します。
DataLen	20	読み出したデータを格納するバッファの最大サイズを指定します。指定したサイズを超えるデータを読み出しても、共有メモリに格納するのはここで指定したサイズだけです。
CycleTime	100	データがあるか確認する周期を 10[ms]単位で指定します。100 では 1 秒周期で実行します。

※通信するバーコードリーダの通信フォーマットに合わせます。

BarcodeReaderReference の設定の詳細については『4-4 BarcodeReaderReference』を参照してください。
共有メモリについての詳細は『AI-HMI リファレンスマニュアル』を参照してください。

- ⑤ 実際にアプリケーションを実行し、設定どおりに動作するかを確認します。
 アプリケーションを実行するための方法はAI-HMI ユーザーズマニュアルの『第6章 作成したアプリの実行環境 PC での動作』を参照してください。
 共有メモリへの展開が正常に動作しているかはAI-HMI デバッグツールを使用して確認します。
 AI-HMI デバッグツールの使用方法についてはAI-HMI ユーザーズマニュアルの『第7章 AI-HMI デバッグツール』を参照してください。



図 3-2-2-5 AI-HMI デバッグツールでの確認

バーコードを読み出して、共有メモリの 150500、150600、150700、のアドレスに反映されていることを確認してください。

第 4 章 外部拡張通信機器設定リファレンス

本章では外部拡張通信機器との接続にかかわる設定項目について説明します。

4-1 機能概要

1) AI HMI Window – AI HMI Ext Bus Comm Setting (1/2)

項目		初期値	機能
ExtBusComm Setting	CommSetting Reference	COMMType	None 通信する外部機器との通信タイプを設定します。
		NetworkAddress	初期値なし 通信する外部機器との IP アドレスを設定します。(COMMType が MODBUS_TCPIP のときのみ有効です)
		PortNo	0 通信する外部機器とのポート番号を設定します。(COMMType が MODBUS_TCPIP のときのみ有効です)
		COMNo	1 通信するシリアルポート番号を指定します。COM1 を使用する場合は 1 を指定します。(COMMType がシリアル通信用のデバイスのとときに有効です。)
		Baudrate	BPS_9600 シリアル通信のボーレートを設定します。(COMMType がシリアル通信用のデバイスのとときに有効です。)
		DataLen	BIT_8 シリアル通信のデータ長を設定します。(COMMType がシリアル通信用のデバイスのとときに有効です。)
		ParityBit	NONE シリアル通信のパリティビットを設定します。(COMMType がシリアル通信用のデバイスのとときに有効です。)
		StopBit	BIT_1 シリアル通信のストップビット長を設定します。(COMMType がシリアル通信用のデバイスのとときに有効です。)
MODBUSSetting	MODBUSCommReference	COMMTypeNo	0 CommSettingReference の配列番号です。どの通信設定を使用するかを指定します。
		SlaveAddress	0 MODBUS 通信のスレーブアドレスを設定します。
		FunctionCode	FUNC_01H 実行するファンクションコードを設定します。
		DataTopAddress	1 先頭アドレスを設定します。
		DataSize	1 データサイズを設定します。
		Address	0 読み書きを行う、共有メモリの先頭アドレスを設定します。
		CycleTime	100 通信周期を 10[ms] 単位で設定します。100 だと、1 秒周期にコマンドを実行します。

1) AI HMI Window – AI HMI Ext Bus Comm Setting (2/2)

項目		初期値	機能
BarcodeReader Setting	BarcodeReader Reference	COMMTyPeNo	0 CommSettingReference の配列番号です。どの通信設定を使用するかを指定します。
		HeaderCode	2 通信ヘッダコードを指定します。ASCII コードの数値を設定します。デフォルトは STX です。
		TerminalCode	3 通信ターミナルコードを指定します。ASCII コードの数値を設定します。デフォルトは ETX です。
		PrefixAddress	0 プレフィックス文字列を格納する共有メモリ先頭アドレスを指定します。
		PrefixDataLen	1 プレフィックスデータの文字数を指定します。HeaderCode を含んだサイズです。
		SuffixAddress	0 サフィックス文字列を格納する共有メモリ先頭アドレスを指定します。
		SuffixDataLen	1 サフィックスデータの文字数を指定します。TerminalCode を含んだサイズです。
		Address	0 読み出したデータを格納する共有メモリの先頭アドレスを指定します。
		DataLen	10 読み出したデータを格納するバッファの最大サイズを指定します。指定したサイズを超えるデータを読み出しても、共有メモリに格納するのはここで指定したサイズだけです。
		CycleTime	100 データがあるか確認する周期を 10[ms]単位で指定します。100 では 1 秒周期で実行します。

4-2 ExtBusCommSetting

外部機器と通信するための設定を行います。

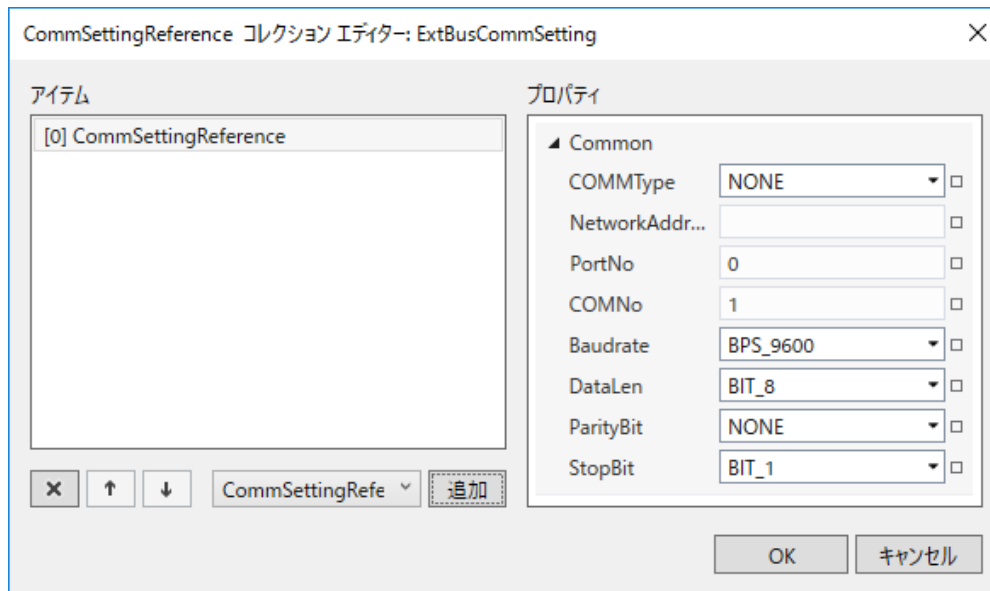


図 4-2-1 CommSettingReference のプロパティ

- └ CommSettingReference
 - └ COMMType
 - └ NetworkAddress
 - └ PortNo
 - └ COMMNo
 - └ Baudrate
 - └ DataLen
 - └ ParityBit
 - └ StopBit

図 4-2-2 CommSettingReference の階層構造

表 4-2-1 CommSettingReference 詳細 (1/2)

項目	初期値	内容
COMMType	None	<p>通信する外部機器の通信タイプを設定します。 入力はプルダウンメニューから行います。 入力できる値は以下の通りです。</p> <ul style="list-style-type: none"> ・ None : 外部機器と通信しません ・ MODBUS_ASCII : MODBUS 機器とシリアルポートで、ASCII モードで通信します。 ・ MODBUS_RTU : MODBUS 機器とシリアルポートで、RTU モードで通信します。 ・ MODBUS_TCP/IP : MODBUS 機器と TCP/IP で通信します。 ・ BarcodeReader : バーコードリーダとシリアルポートで通信します。
NetworkAddress	初期値なし	<p>通信する外部機器の IP アドレスを設定します。 IP アドレス形式の値を入力できます。</p> <p>ここで入力した IP アドレスの PLC に対して通信を行うようになります。 無効な IP アドレスを入力した場合、外部機器との通信を行いません。 本設定が有効なのは、COMMType に MODBUS_TCP/IP を設定したときです。</p>
PortNo	0	<p>通信する外部機器のポート番号を設定します。 入力できる値は 16bit の整数値です。</p> <p>ここで入力したポート番号に対して通信を行うようになります。 無効なポート番号を入力した場合、外部機器との通信を行いません。 本設定が有効なのは、COMMType に MODBUS_TCP/IP を設定したときです。</p>
COMNo	1	<p>シリアル通信の COM ポート番号を指定します。 入力できる値は 8bit の整数値です。</p> <p>ここで入力した COM ポートを使ってシリアル通信を行います。 無効な COM ポートを指定した場合、通信を行いません。 本設定が有効なのは、COMMType にシリアル通信を行うタイプを設定したときです。</p>
Baudrate	BPS_9600	<p>シリアル通信のボーレートを設定します。 入力はプルダウンメニューから行います。 入力できる値は以下の通りです。</p> <ul style="list-style-type: none"> ・ BPS_1200 : 1200bps で通信 ・ BPS_2400 : 2400bps で通信 ・ BPS_4800 : 4800bps で通信 ・ BPS_9600 : 9600bps で通信 ・ BPS_19200 : 19200bps で通信 ・ BPS_38400 : 38400bps で通信 ・ BPS_56800 : 56800bps で通信 ・ BPS_115200 : 115200bps で通信 <p>本設定が有効なのは、COMMType にシリアル通信を行うタイプを設定したときです。</p>
DataLen	BIT_8	<p>シリアル通信のデータ長を設定します。 入力はプルダウンメニューから行います。 入力できる値は以下の通りです。</p> <ul style="list-style-type: none"> ・ BIT_8 : データ長 8bit で通信 ・ BIT_7 : データ長 7bit で通信 <p>本設定が有効なのは、COMMType にシリアル通信を行うタイプを設定したときです。</p>

表 4-2-1 CommSettingReference 詳細 (2/2)

項目	初期値	内容
ParityBit	NONE	シリアル通信のパリティビットを設定します。 入力はプルダウンメニューから行います。 入力できる値は以下の通りです。 ・NONE : パリティビットなし ・ODD : 奇数パリティ付加 ・EVEN : 偶数パリティ付加 本設定が有効なのは、COMMType にシリアル通信を行うタイプを設定したときです。
StopBit	BIT_1	シリアル通信のストップビットを設定します。 入力はプルダウンメニューから行います。 入力できる値は以下の通りです。 ・BIT_1 : ストップビット 1bit ・BIT_15 : ストップビット 1.5bit ・BIT_2 : ストップビット 2bit 本設定が有効なのは、COMMType にシリアル通信を行うタイプを設定したときです。

CommSettingReference を複数登録した場合、CommSettingReference コレクションエディターには上から [0]、[1]、[2] … という形で CommSettingReference が登録されます。(図 4-2-3)

このとき、[]内の数値は MODBUSSetting と、BarcodeReaderSetting の「COMMTypeNo」の項目で設定する値に紐付けられます。

詳細は『4-3 MODBUSSetting』および『4-4 BarcodeReaderSetting』を参照してください。

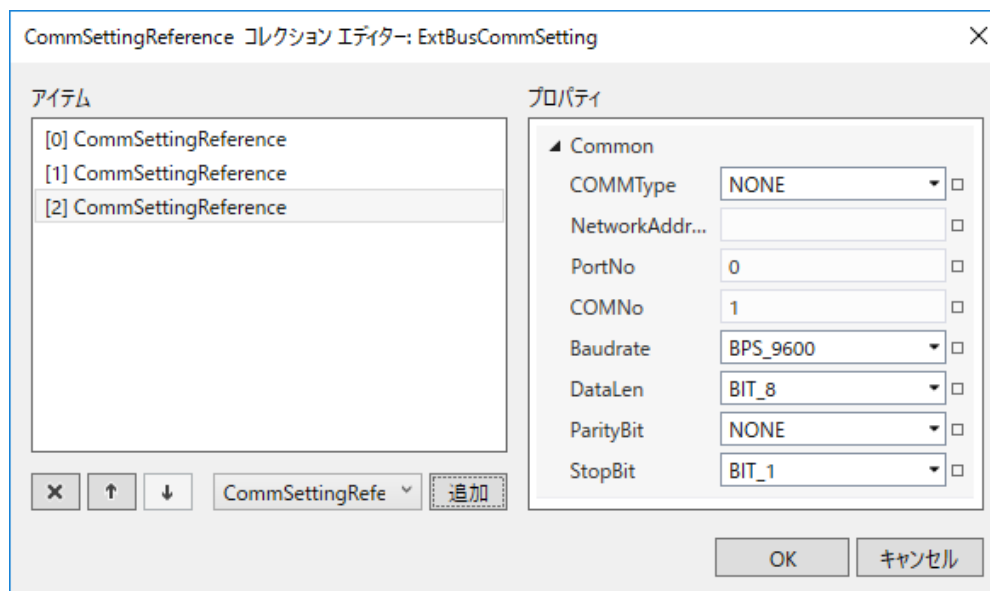


図 4-2-3 CommSettingReference の複数登録

4-3 MODBUSSetting

MODBUS 機器のデータを共有メモリからやり取りする設定を行います。

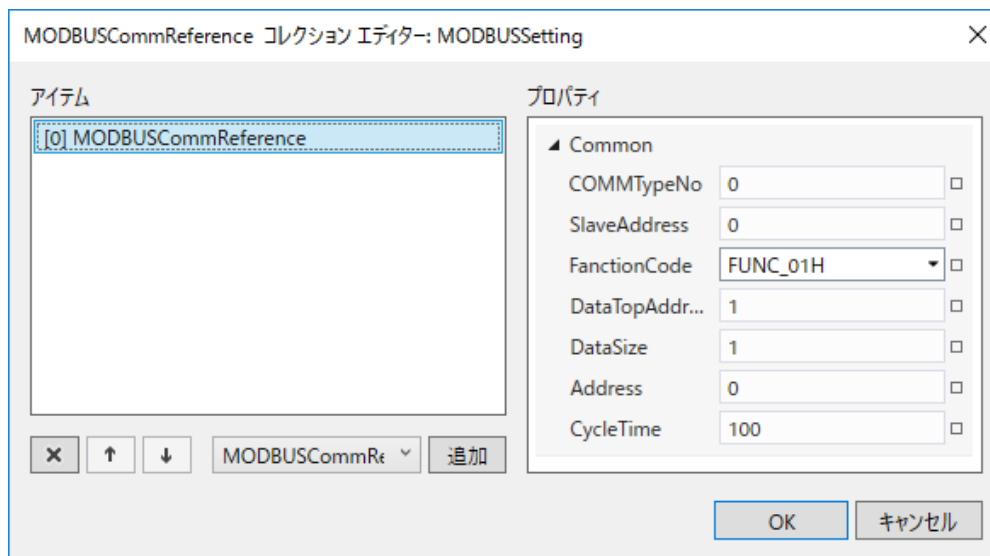


図 4-3-1 MODBUSCommReference のプロパティ

- └ MODBUSCommReference
 - └ COMMTypeNo
 - └ SlaveAddress
 - └ FuctionCode
 - └ DataTopAddress
 - └ DataSize
 - └ Address
 - └ CycleTime

図 4-3-2 MODBUSCommReference の階層構造

表 4-3-1 MODBUSCommReference 詳細(1/2)

項目	初期値	内容
COMMTypeNo	0	<p>CommSettingReference コレクション番号を設定します。 入力できる値は 16bit の整数値です。</p> <p>[COMMTypeNo]の番号の CommSettingReference に対応する外部機器に対して通信を行うようになります。 無効な CommSettingReference コレクション番号や存在しない CommSettingReference コレクション番号を入力した場合、外部機器との通信を行いません。</p>
SlaveAddress	0	<p>MODBUS 通信のスレーブアドレスを設定します。 入力できる値は 8bit の整数値です。</p> <p>SlaveAddress の詳細は MODBUS 通信機器のマニュアルを参照してください。</p>
FanctionCode	FUNC_01H	<p>MODBUS 通信のファンクションコードを設定します。 入力はプルダウンメニューから行います。 入力できる値は以下の 10 種類です。</p> <ul style="list-style-type: none"> ・ FUNC_01H : Read Coil Status ・ FUNC_02H : Read Input Status ・ FUNC_03H : Read Holding Register ・ FUNC_04H : Read Input Register ・ FUNC_05H : Force Single Coil ・ FUNC_06H : Preset Single Register ・ FUNC_0BH : Fetch Communication Event Counter ・ FUNC_0CH : Fetch Communication Event Log ・ FUNC_0FH : Force Multiple Coils ・ FUNC_10H : Preset Multiple Registers
DataTopAddress	1	<p>MODBUS 通信のファンクションコード毎のフォーマットである開始アドレスを設定します。 入力できる値は 16bit の整数値です。</p> <p>ファンクションコードによってどのように使用されるかは異なります。 詳細は MODBUS 通信機器のマニュアルを参照してください。</p>
DataSize	1	<p>MODBUS 通信のファンクションコード毎のフォーマットであるサイズを設定します。 入力できる値は 16bit の整数値です。</p> <p>ファンクションコードによってどのように使用されるかは異なります。 詳細は MODBUS 通信機器のマニュアルを参照してください。</p>

表 4-3-1 MODBUSCommReference 詳細(続き)

項目	初期値	内容
Address	0	<p>展開先の先頭共有メモリアドレスを設定します。 入力できる値は 32bit の整数値です。</p> <p>[Address] で設定した共有メモリのアドレスを先頭として [DataSize] 分の MODBUS データを読み書きします。読み書きの方向はファンクションコードにより決定されます。読み書きされるメモリサイズは、各ファンクションコードのレスポンスデータがそのまま格納されます。</p> <p>例えば「FactionCode」に「FUNC_01H : ReadCoilStatus」を設定した場合、「DataSize」設定値/8 バイトのデータが共有メモリにコピーされます。 無効なアドレスを入力した場合、共有メモリへの読み書きは行いません。</p> <p>※ [Address] には OPC 変数は使用できません。</p>
CycleTime	100	<p>通信する周期を設定します。 入力できる値は 32bit 整数値です。</p> <p>設定された値 x 10 [ms] 周期でコマンドが実行されます。</p>

[例]

MODBUS スレーブ 1 から、100~150 のコイルデータを取得して、150000 番地にデータを 100[ms]周期で反映させる場合は以下のように設定します。

SlaveAddress = 1
 FactionCode = FUNC_01H
 DataTopAddress = 100
 DataSize = 51
 Address = 150000
 CycleTime = 10

上記の設定をすると共有メモリの 150000 番地からは下記のようなデータの並びで展開されます。

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
150000	107	106	105	104	103	102	101	100
150001	115	114	113	112	111	110	109	108
150002	123	122	121	120	119	118	117	116
150003	131	130	129	128	127	126	125	124
150004	139	138	137	136	135	134	133	132
150005	147	146	145	144	143	142	141	140
150006	-	-	-	-	-	150	149	148

[例]

MODBUS スレーブ 1 から、100~105 の保持レジスタデータを取得して、150000 番地にデータを 100[ms]周期で反映させる場合は以下のように設定します。

```
SlaveAddress = 1  
FunctionCode = FUNC_03H  
DataTopAddress = 100  
DataSize = 6  
Address = 150000  
CycleTime = 10
```

上記の設定をすると共有メモリの 150000 番地からは下記のようなデータの並びで展開されます。

Address	上位バイト	下位バイト
150000		100
150002		101
150004		102
150006		103
150008		104
150010		105

4-4 BarcodeReaderSetting

データを取得するバーコードリーダーのデバイスを指定します。

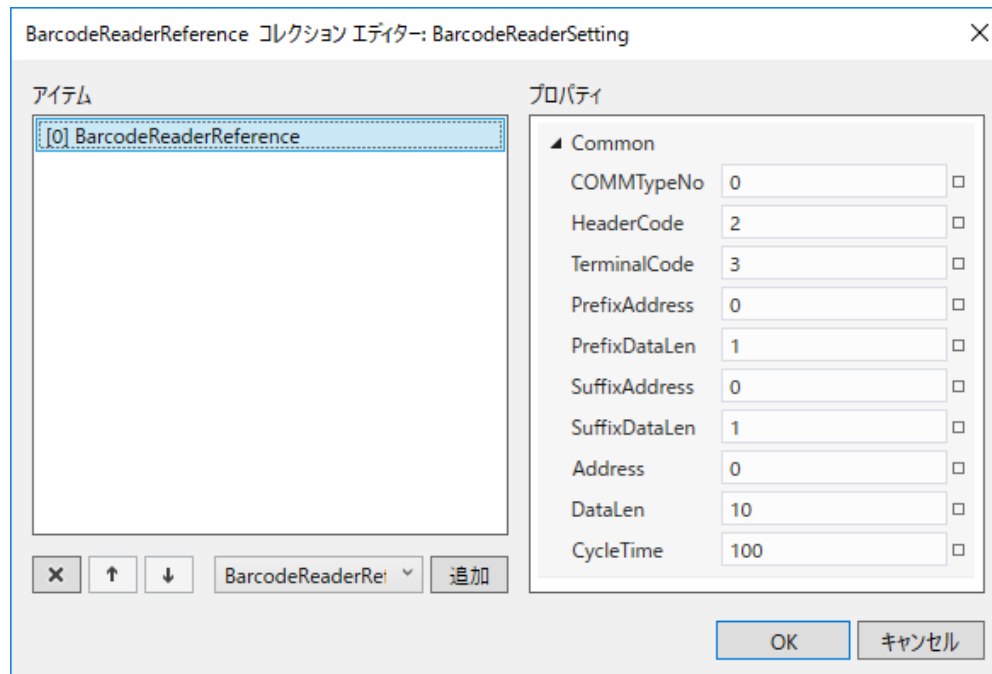


図 4-4-1 BarcodeReaderSetting のプロパティ

- └ BarcodeReaderReference
 - └ COMMTypeNo
 - └ HeaderCode
 - └ TerminalCode
 - └ PrefixAddress
 - └ PrefixDataLen
 - └ SuffixAddress
 - └ SuffixDataLen
 - └ Address
 - └ DataLen
 - └ CycleTime

図 4-4-2 BarcodeReaderReference の階層構造

バーコードリーダーは MODBUS のように共通のフォーマットが決まっているわけではありません。基本的には、図 4-4-3 のように大きく 3 つの部に別れます。

0	～	p	p+1	～	p+n	p+n+1	～	p+n+s
HeaderCode	ID 等	その他	バーコードデータ			BCC 等	その他	TerminalCode
プレフィックスデータ			データ			サフィックスデータ		

図 4-4-3 バーコードリーダー電文構造

1: プレフィックスデータ

先頭の HeaderCode を含め、スキャナの ID や、固定文字列を含んだデータです。初期値としては、なしの場合が多いです。

2: データ

読み出した、バーコードデータになります。これには、付加されるデータはありません。純粋に読み出したデータになります。

3: サフィックスデータ

終端の TerminalCode を含んだ、データです。終端コードの後に BCC 等のチェックサム値が付加される場合もありますが、本 AI-HMI では、あくまで、終端コードまでのデータを取得します。

プロパティに設定するのは、HeaderCode とプレフィックスデータサイズ、TerminalCode とサフィックスデータサイズとなります。TerminalCode の設定は必須です。

TerminalCode を検出したら、プレフィックスデータサイズだけ、プレフィックスデータとして、共有メモリに格納し、その後、残データサイズがサフィックスデータサイズと一致するまでをバーコードデータとして共有メモリに格納します。最後に、サフィックスデータを共有メモリに格納して完了となります。

ID 等の確認や BCC 等のチェックは行いません。

表 4-4-1 BarcodeReaderReference 詳細 (1/2)

項目	初期値	内容
COMMTypeNo	0	CommSettingReference コレクション番号を設定します。 入力できる値は 16bit の整数値です。 [COMMTypeNo]の番号の CommSettingReference に対応する外部機器に対して通信を行うようになります。 無効な CommSettingReference コレクション番号や存在しない CommSettingReference コレクション番号を入力した場合、外部機器との通信を行いません。
HeaderCode	2	バーコードリーダ電文のヘッダコードを設定します。 入力できる値は 8bit の整数値です。 一般的には無し、または、STX (0x02) の場合が多いです。無しの場合は 0 を設定します。
TerminalCode	3	バーコードリーダ電文の終端コードを設定します。 入力できる値は 8bit の整数値です。 一般的には CR (0x0D)、または、ETX (0x03) の場合が多いです。終端コードはかならず設定してください。終端コードの無いバーコードリーダ電文は読み出すことができません。
PrefixAddress	0	プレフィックスデータ展開先の先頭共有メモリアドレスを設定します。 入力できる値は 32bit の整数値です。 [PrefixAddress]で設定した共有メモリのアドレスを先頭として [PrefixDataLen]分のプレフィックスデータを格納します。 無効なアドレスを入力した場合、共有メモリへの書き込みは行いません。 ※ [PrefixAddress]には OPC 変数は使用できません。
PrefixDataLen	1	プレフィックスデータサイズを、HeaderCode を含めて設定します。 入力できる値は 32bit の整数値です。 プレフィックスデータが無い場合は 0 を設定してください。 プレフィックスデータサイズが読み出すバーコードによって可変の場合は対応できません。固定化できる部分のサイズを設定して、残りは、バーコードデータに含める形で設定してください。
SuffixAddress	0	サフィックスデータ展開先の先頭共有メモリアドレスを設定します。 入力できる値は 32bit の整数値です。 [SuffixAddress]で設定した共有メモリのアドレスを先頭として [SuffixDataLen]分のサフィックスデータを格納します。 無効なアドレスを入力した場合、共有メモリへの書き込みは行いません。 ※ [SuffixAddress]には OPC 変数は使用できません。
SuffixDataLen	1	サフィックスデータサイズを、TerminalCode を含めて設定します。 入力できる値は 32bit の整数値です。 サフィックスデータは少なくとも Terminal コードが必須になりますので 1 以上の値を設定してください。 サフィックスデータサイズが読み出すバーコードによって可変の場合は対応できません。固定化できる部分のサイズを設定して、残りは、バーコードデータに含める形で設定してください。

表 4-4-1 BarcodeReaderReference 詳細 (2/2)

項目	初期値	内容
Address	0	<p>バーコードデータ展開先の先頭共有メモリアドレスを設定します。 入力できる値は 32bit の整数値です。</p> <p>[Address] で設定した共有メモリのアドレスを先頭として読み出したバーコードデータを格納します。最大格納データサイズが [DataLen] 分のデータを格納します。[DataLen] を超えるデータを受信した場合、切り捨てとなります。無効なアドレスを入力した場合、共有メモリへの書き込みは行いません。</p> <p>※ [Address]には OPC 変数は使用できません。</p>
DataLen	10	<p>バーコードデータ最大格納サイズを設定します。 入力できる値は 32bit の整数値です。</p> <p>[DataLen] 分のバッファを共有メモリに確保します。</p>
CycleTime	100	<p>通信する周期を設定します。 入力できる値は 32bit 整数値です。</p> <p>設定された値 x 10 [ms] 周期でコマンドが実行されます。</p>

このマニュアルについて

- (1) 本書の内容の一部または全部を当社からの事前の承諾を得ることなく、無断で複写、複製、掲載することは固くお断りします。
- (2) 本書の内容に関しては、製品改良のためお断りなく、仕様などを変更することがありますのでご了承下さい。
- (3) 本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがございましたらお手数ですが巻末記載の弊社までご連絡下さい。その際、巻末記載の書籍番号も併せてお知らせ下さい。