

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	パ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 MODBUSSeting	4 - 6
4—4 BarcodeReaderSetting	4 – 1 0

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。

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

図 1-1 データ型名対応表

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 MDOBUS 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 という通信方式が有ります。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
- 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。
 Visual Studio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

 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

「AI HMI Ext Bus Comm Setting - ExtBusCommSe	tting」のボタンをクリックしてください。
	□ ▼ 3 <u>9/1992時 (Ctrl+Q) ● </u>
ビジョーションエク ▼ ヰ × MainWindow.xaml* +> ×	לםולדז ד אַ אַ
	□ 名前 <名前atal> 種類 Al_HMI_Window
	検索プロパティ ク 並べ替え: カテゴリマ ▲
J > □ AI_HMI_Pac J > □ AI_HMI_Pac	 ▶ ブラシ ▶ 外観
	 ▶ 共通 ▶ レイアウト
> □ AI_HMI_Pac > □ AI_HMI_Pac	 ▶ テキスト ▶ その他の指定
 ALHMLPac ALHMLPac ALHMLPac ALHMLPac 	AI HMI Alarm Setting AI HMI Beep Setting
 ALHMLPac ALHMLPac ALHMLPac 	▲ AI HMI Ext Bus Comm Setting BarcodeReaderSetting (コレクション)ロ
	ExtBusCommSetting (コレクション) … ロ MODBUSSetting (コレクション) … ロ
App.config App.xaml App.xaml MainWindc MainWindc	AI HMI Logging Setting
xmins:x= http://schem xmins:Components="cir" Title="MainWindow" He +	AI HMI Login Setting AI HMI PLC Setting
<u> </u>	Al HMI Sampling Setting

3 AI HMI Ext Bus Comm Settingの設定項目が表示されます。

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。

CommSettingReference コレクション エディター: ExtBusCommSetting X			
アイテム	プロパティ		
X 1 L CommSetting X 1810			
	OK ++77511		

図 3-1-1-4 CommSettingReferenceの変更

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。

改 止仮、	ION	小グノをク	999	· · · 。

[0] CommSettingReference ▲ Common COMMType NONE NetworkAddr □ PortNo 0 COMNo 1 Baudrate BPS_9600 DataLen BIT_8 ParityBit NONE StopBit BIT_1	የተテム	プロパティ		
COMMType NONE	[0] CommSettingReference	Common		
NetworkAddr □ PortNo 0 COMNo 1 Baudrate BPS_9600 • DataLen BIT_8 • ParityBit NONE • StopBit BIT_1 •		COMMType	NONE	•
PortNo 0 COMNo 1 Baudrate BPS_9600 DataLen BIT_8 ParityBit NONE StopBit BIT_1 ■		NetworkAddr		
COMNo 1 Baudrate BPS_9600 DataLen BIT_8 ParityBit NONE StopBit BIT_1		PortNo	0	
Baudrate BPS_9600 □ DataLen BIT_8 □ ParityBit NONE □ StopBit BIT_1 □		COMNo	1	
DataLen BIT_8 □ ParityBit NONE □ StopBit BIT_1 □		Baudrate	BPS_9600	•
ParityBit NONE □ StopBit BIT_1 □		DataLen	BIT_8	•
StopBit BIT_1 ・□		ParityBit	NONE	•
★ ↑ ↓ CommSetting > 追加		StopBit	BIT_1	•
	× ↑ ↓ CommSetting > 追加			

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 PTU	MODBUS 機器とシリアルポートの RTU モードで通信し
Соммтуре		ます。
NetworkAddross	t: 1 ×	COMMType が MODBUS_TCPIP の場合、通信する MODBUS
Ne LWOI KAULI ESS	なし※	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
	0%	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
COMINO	1	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 X	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。

Г

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。

	AI_HMI_CTRL - Microsoft	Visual Studio	₽ ₹3	クイック起動 (Ctrl+Q)	₽ = ¤ ×
77	イル(<u>F) 編集(E)</u> 表示(<u>V</u>)	プロジェクト(<u>P)</u> ビルド(<u>B</u>) デバッグ(<u>D</u>) チーム	(<u>M</u>) デザイン(<u>G</u>)	書式(<u>O) ツール(T</u>)	テスト(<u>S</u>) サインイン 🎴
分析	f(<u>N)</u> ウィンドウ(<u>W</u>) ヘルプ	'(<u>H</u>)			
) - 🛛 🛅 - 😩 💾	▶ 🏓 🕶 🦿 🔹 🕨 🕨 開始 🕶 🖒 👻 Release 🔹	🎜 🖕 🖁 🔚 🖷	1월 2월 📕 14 14	2월 🖕
<u>ب</u>	ソリューション エク マ 甲 🗙	MainWindow.xaml* + ×	プロパティ		- ₽ ×
년	o o ☆ io - "	-	名前 <:	名前なし>	5 5
גלי	ソリューション エクスプロ 🔎 -	1000	□ 種類 AI	HMI Window	
Z	b □ ALHMI Pac ▲		検索プロパティ		م
L S	AI_HMI_Pac	1000	並べ替え: カテゴ!	J -	A
14	AL_HMI_Pag		▶ ブラシ		
문	P L ALHMI Pac D ALHMI Pac		▶ 外観		
ダイン	AI_HMI_Pag		▶ 共通		
90	🕨 🔓 Al_HMI_Pac		▶ レイアウト		
-9 V	ALHMIPac	0000	▶ テキスト		
r K	ALHMI_Pac	0000	▶ その他の指定		
	AI_HMI_Pac	0000	▶ AI HMI Alarn	n Setting	
	ALHMIPac	1000	▶ AI HMI Beep	Setting	
	P L ALHMIPAC ▶ L ALHMIPAC	0.000	AI HMI Ext B	us Comm Setting	
	AI_HMI_Pac		BarcodeRead	derSetting (コレクション)	
	AI_HMI_Pac		ExtBusComm	nSetting (コレクション)	
	App.config		MODBUSSet	ting (コレクション)	🗆
	App.xaml	<pre>Components:AI_HMI_Window +</pre>	AI HMI Logg	ing Setting	
	👂 🛄 MainWindc	xmins= http://schemas xmins:x="http://schem_	▶ AI HMI Logir	n Setting	
		xmlns:Components="cir Title="MainWindow" He •	♦ AI HMI PLC S	Setting	
	ソリューショ チームエクス	100 % - 4	AI HMI Samp	oling Setting	•
準備				AI HMI Sampling Set	tting プロパティ

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。

MODBUSCommReference コレクション エディター: MODBU	ISSetting X		
7474	プロパティ		
★ 1 ↓ MODBUSCommRŧ ✓ 追加	OK キャンセル		
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。

MODBUSCommReference コレクション エディター: MODBUSSetting				
ፖイテム	プロパティ			
[0] MODBUSCommReference	Common			
	COMMTypeNo	0		
	SlaveAddress	1		
	FanctionCode	FUNC_01H	•	
	DataTopAddr	0		
	DataSize	128	-	
	Address	150000		
	CycleTime	100		
× ↑ ↓ MODBUSCommRe × 追加				
	[OK	キャンセル	

図 3-1-2-3 MODBUSCommReference[0]の変更

項目	設定値	備考		
COMMTypeNe	0	CommSettingReference で登録した通信設定の		
COMMITYPENO	0	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。		
EquationCode		コイルのデータを読み出す場合、FUNC_01H を		
Fanctioncode		設定します。		
DeteTerAddress	0	読み出すコイルデータの先頭アドレスを指定		
Datalopaduress	0	します。		
		読み出すコイルデータサイズを指定します。こ		
Detecize	128	ちらの指定は bit 単位です。読み出しデータ		
Datasize		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		
Address	150000	頭アドレスを指定します。		
Qualatima	100	読み出す周期を10[ms]単位で指定します。100		
	100	だと1秒周期で実行します。		

<u>表 3-1-2-1</u> MODBUSCommReference [0]のプロパティ

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。

MODBUSCommReference コレクション エディター: MODBUSSetting				
アイテム	プロパティ			
[0] MODBUSCommReference	Common			
	COMMTypeNo	0		
	SlaveAddress	1	-	
	FanctionCode	FUNC_0FH	-	
	DataTopAddr	0	-	
	DataSize	128	-	
	Address	150300	-	
	CycleTime	100		
× ↑ ↓ MODBUSCommR∉ × 追加				
			ha hala la ll	
		OK	キャンセル	

<u>図 3-1-2-4 MODBUSCommReference [1]の変更</u>

項目	設定値	備考
COMMTypeNe	0	CommSettingReference で登録した通信設定の
commigpeno	0	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。
FanationCode		コイルのデータを書き込む場合、FUNC_OFH を
Fanctioncode		設定します。
DataTapAddroop	0	書き込むコイルデータの先頭アドレスを指定
DatatopAddress	0	します。
		書き込むコイルデータサイズを指定します。こ
DeteSize	128	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	150200	書き込みするデータを格納する共有メモリの
Address	150300	先頭アドレスを指定します。
QualaTima	100	読み出す周期を 10[ms]単位で指定します。100
Cycrerime	100	だと1秒周期で実行します。

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

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。

📧 AI HMI 共有火モ	リビューワー	↔	– 🗆 X
表示先頭アドレス 表示サイズ(byte)	150000 1000	✔ 更新 🔜 書込み	
表示データタイプ	UINT	◇ 表示データ形式 De	ecimal ×
アドレス	データ		書込みデータ
150000	0		0 ^
150002	0		0
150004	0		0
150006	0		0
150008	0		0
150010	0		0
150012	0		0
150014	0		0
150016	0		0
150018	0		0
150020	0		0
150022	0		0
150024	0		0
150026	0		0
150028	0		0
150030	0		0
150032	0		0
150034	0		0
150036	0		0
150038	0		0
150040	0		0
150042	0		0 ~

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

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。

AI_HMI_Project1 - Microsoft Visual Studio Express 2013 for Windows Desktop						\bigtriangledown	ワ クイッ	ク起動 (Ctrl+Q)	ç	- □	х		
7:	アイル(E) 編集(<u>E</u>) 表示(<u>∨</u>)	プロジェクト	(<u>P</u>) ビルド	<u>B) デバッグ(D</u>) チーム(<u>M</u>)	ツール(<u>T</u>)	テスト(<u>S</u>)	ウィンドウ(<u>W</u>)		サインイン	. 🖻
\sim	レプ(<u>H</u>)												
	G - 6) 🔁 😩	2 🔮 🦻	- 🤊 - 🕨 🖩	始 - Debug	 Any CPU 	• 🏓 📮						
<u>ب</u>	בעע	ーションエ	クスプローラ		×				プロパティ			- ₽	чх
いんだ	00	o 10 (- 2 Q 🗇	🔊 🗠 🖌 –	-				MainWind	ow.xaml ファイ	イルのプロパラ	Fr	•
50	בעע	ーションエ	クスプローラ	の検索 (Ctrl	<mark>ہ م</mark>)			
~	N	ノリューショ	> 'AI_HMI_F	Project1' (1 プロ	IST.				□ その他				
15	4 @	■ AI_HMI_	Project1						ファイル	名	MainWindow	.xaml	
L L	Þ	🔑 Proper	ties						完全パス		C:¥Users¥gij	jyu-3f¥doci	umer
W W	Þ	■■ 参照設	定						□ 詳細				
K		Alarm	- hal						カスタム	ツール	MSBuild:Con	npile	
5		Alarm	Label						カスタム	ツールの名前空	_		
Ľ		Resou	ing						ビルドア	クション	Page		
		- Jisor	ing						出力ディ	レクトリにコピー	コピーしない		
ļų	6	ViewM	odel										
	Þ		II Page1.xar	nl									
	⊳		II_Page2.xar	nl									
	⊳	AI_HM	II_Page3.xar	nl									
		App.co	onfig										
	Þ	🗅 App.xa	aml		_								
	►	📙 MainW	/indow.xaml										
									その他				
	•				•								
	出力												
準備	完了												

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

AI HWI EXT DUS COUNN SELLING - EXTR	uscommiserting」のホタンをクリックしてくたさい。
Al_HMI_CTRL - Microsoft Visual Studio ファイル(E) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッ・ 分析(N) ウィンドウ(W) ヘルプ(H) ③ • ◎ 悶 • ③ ■ 』 ク • ○ • ▶ 開始 • ○ •	□ ▼ 3 クイック起動(Ctrl+Q)
้ ้ ัั่ 7/ปュ−ション I/ว ▼ 🕂 🗙 MainWindow.xaml* 🗢 🗙	 「プロパティ
Image: Second secon	A 前 <名前なL> 建類 AI_HMI_Window 検索プロパティ 並べ替え: カテゴリ マ > ブラジ > 外観 > 共通 > レイアウト > テキスト > その他の指定 > AI HMI Alarm Setting > AI HMI Beep Setting = AI HMI Ext Bus Comm Setting BarcodeReaderSetting (コレクション) … □ ExtBusCommSetting (コレクション) … □ MODBUSSetting (コレクション) … □ MODBUSSetting (コレクション) … □ MODBUSSetting (コレクション) … □
<u>ソリューショ</u> チームエクス <mark>100 % →</mark> ◀ ■	► AI HMI Sampling Setting ▼
洋備完了	ان الم

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。

CommSettingReference コレクション エディター: ExtBusCommSetting >					
717L	วื่อเ/เริ่า	_			
× ↑ ↓ CommSetting ど 追加					
	OK キャンセル				

図 3-2-1-4 CommSettingReferenceの変更

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。

改 止 仮 、	IOK]	ホダン	そう	リック	. < /26	· · · 。	

CommSettingReference コレクション エディター: ExtBusCommSetting					
アイテム	プロパティ				
[0] CommSettingReference	▲ Common				
	COMMType	NONE	•		
	NetworkAddr				
	PortNo	0			
	COMNo	1			
	Baudrate	BPS_9600	•		
	DataLen	BIT_8	•		
	ParityBit	NONE	• •		
	StopBit	BIT_1	•		
× ↑ ↓ CommSetting > 追加					
		OK :	キャンセル		

図 3-2-1-5 CommSettingReferenceの変更

<u>表 3-2-1-1 CommSettingReference のプロパティ</u>
--

項目	設定値	備考
СОММТуре	BarcodeReader	バーコードリーダとシリアルポートで通信します。
NetworkAddress	なし	バーコードリーダでは TCP/IP モードでは通信する
PortNo	0	ことができません。そのため、この設定は無効です。
COMNo	1	バーコードリーダと接続しているシリアルポート番
COMING		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。

×	AI_HMI_CTRL - Microsoft	Visual Studio	□ ▼3 クイック起動 (Ctrl+Q) □ ▼ ■ □ ×
ファイ 分析	(ル(E) 編集(E) 表示(V) f(N) ウィンドウ(W) ヘルス	フロジェクト(<u>P)</u> ビレド(<u>B</u>) デバック(<u>D</u>) チーム(が中)	ム(M) デザイン(G) 書式(Q) ツール(D) テスト(S) サインイン 🎦
8.) - O 🕅 - 🏠 💾 🚽	■ 9 • C • ▶ 開始 • O • Release •	# _ 8 告 晤 注 注 對 領 領 省 _
4		MainWindow xaml* + + ×	
-Ju#			名前 <名前 / · · · · · · · · · · · · · · · · · ·
דלע	ソリューション エクスプロ ク・	0000	■ 日本 C HANGOY 種類 AI HMI Window
푞	▷ □ AI_HMI_Pac ▲	10000	検索プロパティ ・
ΥĽ VYĽ	AI_HMI_Pac	0000	並べ替え: カテゴリ ▼
+ 7	▷ □ ALHMI_Pac	0000	▶ ブラシ
545	AL_HMI_Pag	0000	▶ 外観
N.	AI_HMI_Pag	0000	▶ 共通
Ť-S	P L ALHMIPac	0000	▶ レイアウト
7 7-	AI_HMI_Pac		▷ 7 ≠スト
R I	AI_HMI_Pag	0000	▶ その他の指定
	P L ALHMIPAC ▶ L ALHMIPAC	0000	AI HMI Alarm Setting
	AI_HMI_Pac	0000	AI HMI Beep Setting
	ALHMIPac	10000	BarcodeReaderSetting
	ALHMI_Pag	•	FxtBusCommSetting (コレクション)
	AI_HMI_Pag		MODBUSSetting (コレクション)
	Y App.config ▷ □ App.xaml	Components:AI_HMI_Window	
	MainWindc	xmins="http://schemas xmins:x="http://schem	AI HMI Login Setting
	▼ ↓	xmlns:Components="clr Title="MainWindow" He	 AI HMI PLC Setting
	ソリューショ チーム エクス		♦ AI HMI Sampling Setting
準備	完了		AI HMI Sampling Setting プロパティ

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。

BarcodeReaderReference コレクション エディター: Barcodel	ReaderSetting ×
アイテム	プロパティ
★ ↑ ↓ BarcodeReaderRet ¥ 追加	
	OK キャンセル
⊠ 3-2-2-2 BarcodeRea	aderReferenceの変更

3 – 1 3

 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。

BarcodeReaderReference コレクション エディター: BarcodeReaderSetting X				
アイテム	プロパティ			
[0] BarcodeReaderReference	▲ Common			
	COMMTypeNo	0		
	HeaderCode	0	-	
	TerminalCode	13	-	
	PrefixAddress	150500	-	
	PrefixDataLen	0	-	
	SuffixAddress	150600	-	
	SuffixDataLen	0	-	
	Address	150700	-	
	DataLen	20	-	
	CycleTime	100		
× ↑ ↓ BarcodeReaderRet × 追加				
		OK +t	ッンセル	

<u>図 3-2-2-3 MODBUSCommReference[0]の変更</u>

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。
HaadarCadaX	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
	0	は0を設定します。
TerminalCodeX	13	通信フォーマットの終端コードを設定します。
	10	「CR」の場合 10 進数で 13 を設定します。
Prefivaddress	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
	100000	ます。
PrefivDataSize*	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
	0	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
	100000	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
	I	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
	130700	す。
		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
DataLen	20	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。
CvcleTime	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
	100	秒周期で実行します。

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

※通信するパ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。

📧 AI HMI 共有火モ	リビューワー	↔	– 🗆 X
表示先頭アドレス 表示サイズ(byte)	150000 1000	✔ 更新 🔜 書込み	
表示データタイプ	UINT	◇ 表示データ形式 De	ecimal ×
アドレス	データ		書込みデータ
150000	0		0 ^
150002	0		0
150004	0		0
150006	0		0
150008	0		0
150010	0		0
150012	0		0
150014	0		0
150016	0		0
150018	0		0
150020	0		0
150022	0		0
150024	0		0
150026	0		0
150028	0		0
150030	0		0
150032	0		0
150034	0		0
150036	0		0
150038	0		0
150040	0		0
150042	0		0 ~

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
		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)
Ex+Buo Comm	CommSetting Reference	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 きに有効です。)
Setting		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き に有効です。)
		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。
		FanctionCode	FUNC_01H	実行するファンクションコードを設定します。
MODBUSSetting	MODBUSCommRefe	DataTopAddress	1	先頭アドレスを設定します。
mobboootting	rnece	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。

	項目		初期値	機能
		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
			2	の数値を設定します。デフォルトは STX です。
				通信ターミナルコードを指定します。ASCII コ
		TerminalCode	3	ードの数値を設定します。 デフォルトは ETX で
				す。
		ProfixAddross	0	プレフィックス文字列を格納する共有メモリ
	BarcodeReader Reference	FreitxAuuress	U	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。
BaraadaPaadar				HeaderCode を含んだサイズです。
Sotting		SuffixAddress	0	サフィックス文字列を格納する共有メモリ先
Setting				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。
		Addross	0	読み出したデータを格納する共有メモリの先
		Address	0	頭アドレスを指定します。
				読み出したデータを格納するバッファの最大
		Datal on	10	サイズを指定します。指定したサイズを超える
		DataLen	10	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
			100	指定します。100 では1 秒周期で実行します。

1) A	I HMI	Window —	ΑI	HMI	Ext	Bus	Comm	Setting	(2/2)
------	-------	----------	----	-----	-----	-----	------	---------	-------

4-2 ExtBusCommSetting

CommSettingReference コレクション エディター: ExtBusCommSetting				
アイテム	プロパティ			
[0] CommSettingReference	Common			
	COMMType	NONE		
	NetworkAddr			
	PortNo	0		
	COMNo	1		
	Baudrate	BPS_9600		
	DataLen	BIT_8	• •	
	ParityBit	NONE	• •	
X 1 L CommSettingRefe × Bth	StopBit	BIT_1	• •	
Commettinghere				
	[ОК ++>>t	211/	

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

- L CommSettingReference
 - ├ COMMType
 - ⊢ NetworkAddress
 - ⊢ PortNo

 - ⊢ Baudrate
 - ⊢ DataLen
 - ⊢ ParityBit
 - ${}^{\cup}$ StopBit

<u>図 4-2-2 CommSettingReferenceの階層構造</u>

項目	初期値	
		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。
	Nono	 None :外部機器と通信しません
COMMITYPE	NOTE	・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_TCPIP : 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。
		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。
NotworkAddross	如期値たし	
NetworkAddress	初期値なし	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_TCPIP を設定したときです。
		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。
PortNo	0	
FOLINO	U	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_TCPIP を設定したときです。
		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。
	1	
COMNo		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。
		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。
		・BPS_1200 :1200bps で通信
		・BPS_2400 :2400bps で通信
		・BPS_4800 :4800bps で通信
Baudrate	BPS_9600	・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
		本設定が有効なのは、COMMIype にシリアル通信を行うタイフを設定したときで
		シリアル通信のテータ長を設定します。
		入力はノルダリンメニューから行います。
D . I		ヘルできる旭は以下の通りです。
DataLen	R11 ⁸	・DII_0 : アーダ 長 XDI L じ 週 信 - DII フー・ データ 트 7ki + ボ ほ 信
		・DII_/ :アープ技 /DIL (週間) 本語学校方称なのけ、COMMType にとして世孫信ちにこちノゴナ語ウレキ したで
		☆政ルルカガなのは、いणणा ypt にンリアル週店を打つダイノを設正したとざじ す
DataLen	BIT_8	 BIT_8 : データ長 8bit で通信 BIT_7 : データ長 7bit で通信 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。

表 4-2-1 CommSettingReference 詳細 (1/2)

項目	初期値	内容
		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。
Devity (Dit	NONE	・NONE :パリティビットなし
Paritybit	NONE	・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。
		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。
Char D'h	DIT 1	・BIT_1 : ストップビット 1bit
StopBit	R11 ⁻ 1	・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。

表 4-2-1 CommSettingReference 詳細 (2/2)

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。

CommSettingReference コレクション エディター: ExtBusCommSetting				
アイテム	プロパティ			
[0] CommSettingReference [1] CommSettingReference	✓ Common COMMType	NONE	• •	
[2] commissioning weierence	NetworkAddr			
	PortNo	0		
	COMNo	1		
	Baudrate	BPS_9600	-	
	DataLen	BIT_8	<u> </u>	
	ParityBit	NONE		
× ↑ ↓ CommSettingRefe > 追加	StopBit	BIT_1	•	
		OK +77	セル	

図 4-2-3 CommSettingReferenceの複数登録

4-3 MODBUSSeting

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。

MODBUSCommReference コレクション エディター: MODBUSSetting				
ፖイテム	プロパティ			
[0] MODBUSCommReference	▲ Common			
	COMMTypeNo	0		
	SlaveAddress	0		
	FanctionCode	FUNC_01H	-	
	DataTopAddr	1		
	DataSize	1		
	Address	0		
	CycleTime	100		
× ↑ ↓ MODBUSCommR€ > 追加				
		ОК	キャンセル	

図 4-3-1 MODBUSCommReference のプロパティ

- L MODBUSCommReference
 - ├ COMMTypeNo
 - ├ SlaveAddress
 - FanctionCode
 - \vdash DataTopAddress
 - ⊢ DataSize
 - ⊢ Address
 - └ CycleTime

図 4-3-2 MODBUSCommReferenceの階層構造

項目	初期値	内容		
		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。		
	0	[COMMTypeNo]の番号の CommSettingReference に対応する外部機器に対して通		
COMINITYPENO	U	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。		
		MODBUS 通信のスレーブアドレスを設定します。		
SlavaAddroca	0	入力できる値は 8bit の整数値です。		
SlaveAuuress	U			
		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。		
		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。		
		FUNC_01H : Read Coil Status		
		FUNC_02H : Read Input Status		
		FUNC_03H : Read Holding Register		
FanctionCode	FUNC_01H	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		
		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。		
DataTonAddress	1	入力できる値は16bitの整数値です。		
		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なます。		
		詳細は MODBUS 通信機器のマニュアルを参照してください。		
		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。		
DataSize	1	入力できる値は16bitの整数値です。		
		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なます。		
		詳細は MODBUS 通信機器のマニュアルを参照してください。		

表 4-3-1 MODBUSCommReference 詳細(1/2)

項目	初期値	内容
Address	0	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。 [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。 例えば「FanctionCode」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。 ※ [Address]には OPC 変数は使用できません。
CycleTime	100	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。 設定された値 x 10 [ms] 周期でコマンドが実行されます。

表 4-3-1 MODBUSCommReference 詳細(続き)

[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 FanctionCode = 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 FanctionCode = 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	10	00
150002	10	D1
150004	10	02
150006	10	03
150008	10)4
150010	10)5

4-4 BarcodeReaderSetting

BarcodeReaderReference コレクション エディター: BarcodeReaderSetting				
アイテム	プロパティ			
[0] BarcodeReaderReference	▲ Common			
	COMMTypeNo	0		
	HeaderCode	2		
	TerminalCode	3		
	PrefixAddress	0		
	PrefixDataLen	1		
	SuffixAddress	0		
	SuffixDataLen	1		
	Address	0		
	DataLen	10		
× ↑ ↓ BarcodeReaderRei > 追加	CycleTime	100		
		OK ŧ	ャンセル	

図 4-4-1 BarcodeReaderSetting のプロパティ

- └ BarcodeReaderReference
 - ├ COMMTypeNo
 - ⊢ HeaderCode
 - ├ TerminalCode
 - ⊢ PrefixAddress
 - ⊢ PrefixDataLen
 - ⊢ SuffixAddress
 - ⊢ SuffixDataLen
 - ⊢ Address
 - . ⊢ DataLen
 - ${}^{\mbox{}}$ CycleTime

図 4-4-2 BarcodeReaderReference の階層構造

バーコードリーダは 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	~	р	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	プレフィックスデータ			データ		t	トフィックス	データ

図 4-4-3 バーコードリーダ電文構造

 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 Terminal Code 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。

項目	初期値	内容
		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。
	0	[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人力した場合、外部機器との通信を
		ハーコートリータ電文のヘッタコートを設定します。
	0	入力できる値は 8DIT の
HeaderGode	2	
		一般的には無し、または、STA(0X02)の場合が多いです。 悪しの場合は 0 を
		」
		ハーコートリータ电叉の於咖コートを設定しまり。 入力できる値は ghit の敷数値です
TerminalCode	3	ー般的には CR(0x0D) または FTX (0x03)の堤合が多いです 終端コードは
		かならず設定してください、終端コードの毎いバーコードリーダ電文は読み出
		すことができません。
		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	0	[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。
		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。
PrefixDataLen	1	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。
		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	U	[SuttixAddress]で設定した共有メモリのアトレスを先頭として
		[SuffixDataLen]分のサフィックスナーダを俗納します。 無効なスパースをきたした根本、サカノエリッの書たけなけたいません
		無効なアトレスを入力した場合、共有メモリへの書き込みは打いません。 ※「SuffixAddrass]には OPC 亦数は使用できません。
		※ [Juil I Made ess]には OFV 支数は使用 Cさません。
		シンコンシスティンショスを、Terminatoutを自めて設定しより。 入力できる値は 32hit の整数値です
		ー 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
SuffixDataLen	1	上の値を設定してください。
		│
		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(1/2)

項目	初期値	内容
		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。
Address	0	[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。 ※ [Address]には OPC 変数は使用できません。
DataLen	10	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。 [DataLen] 分のバッファを共有メモリに確保します。
CycleTime	100	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。 設定された値 x 10 [ms] 周期でコマンドが実行されます。

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

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付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。

7 7 A H 1 O O 1 3 A

2017年 3月 初版

✓▲ GD 株式会社アルゴシステム

本社

〒587-0021 大阪府堺市美原区小平尾6	56番地	TEL (O)7
------------------------	------	--------	----

TEL (072) 362-5067 FAX (072) 362-4856

ホームページ http://www.algosystem.co.jp/