MODBUS Organization, Inc.

MODBUS-TCP Client(Master) Driver

サポートバージョン	OS	V4.0 以上
	XDesignerPlus	4.0.0.0 以上

CONTENTS

㈱ミスミの“Touch Operation Panel(M2I GX7) Series”をご使用になるお客様に感謝申し上げます。本マニュアルを読んで“GX7-外部装置”の接続方法および手順を熟知してください。

1. システム構成 **2ページ**

接続に必要な機器、各機器の設定、ケーブル、構成可能なシステムについて説明します。
本節を参照して適切なシステムを選定してください。

2. GX7機種と外部装置選択 **3ページ**

GX7機種と外部装置を選択します。

3. システム設定例題 **4ページ**

本機器と該当外部端末の通信接続のための設定例題を説明します。
“1. システム構成”で選択したシステムに従って例題を選択してください。

4. 通信設定項目 **6ページ**

GX7の通信設定をする方法について説明します。
外部装置の設定が変わる場合、本章を参照してGX7の設定も外部装置と同じように設定してください。

5. サポートアドレス **8ページ**

本節を参照して外部装置と通信可能なアドレスを確認してください。

APPENDIX A. MODBUS Protocol **9ページ**

本機器の“MODBUS Serial Master Driver”がサポートするMODBUSプロトコルのコマンドおよびデバイスについて説明します。

1. システム構成

本ドライバは“MODBUS Organization, Inc.”の“MODBUS Protocol”のうち“MODBUS-TCP Client(Master)”です。

外部装置(MODBUS Slave Protocolサポート)によってドライバの“コマンドコード”、“プロトコルフレーム形式”などを別途に設定しなければなりません。この場合、通信方式による細部設定事項を外部装置側に合わせて設定してください。
本ドライバがサポートする外部装置とのシステム構成は下記のとおりです。

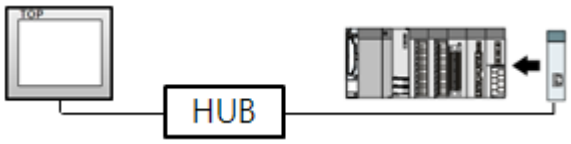
シリーズ	CPU	Link I/F	通信方式	システム設定	ケーブル
MODBUS Slave/Server Device			Ethernet (UDP)	3.1 設定例題 1 (ページ)	ツイストペアケーブル*注1)
			Ethernet (TCP)	3.2 設定例題 2 (ページ)	

*注1) ツイストペアケーブル

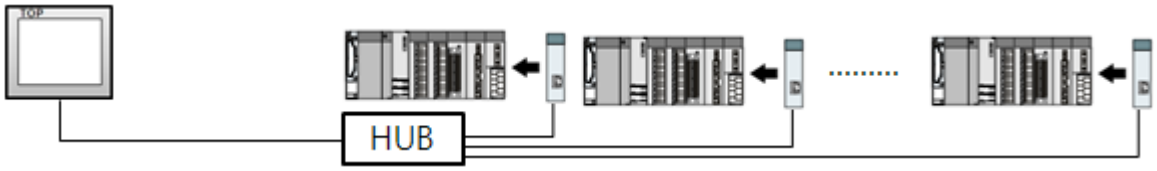
- STP(シールドツイストペアケーブル)またはUTP(非シールドツイストペアケーブル)カテゴリー3, 4, 5を意味します。
- ネットワーク構成によってハブ、トランシーバーなどの構成機器に接続可能で、この場合ダイレクトケーブルを使用してください。

■ 接続可能な構成

・1：1接続(GX7 1台と外部装置1台)接続

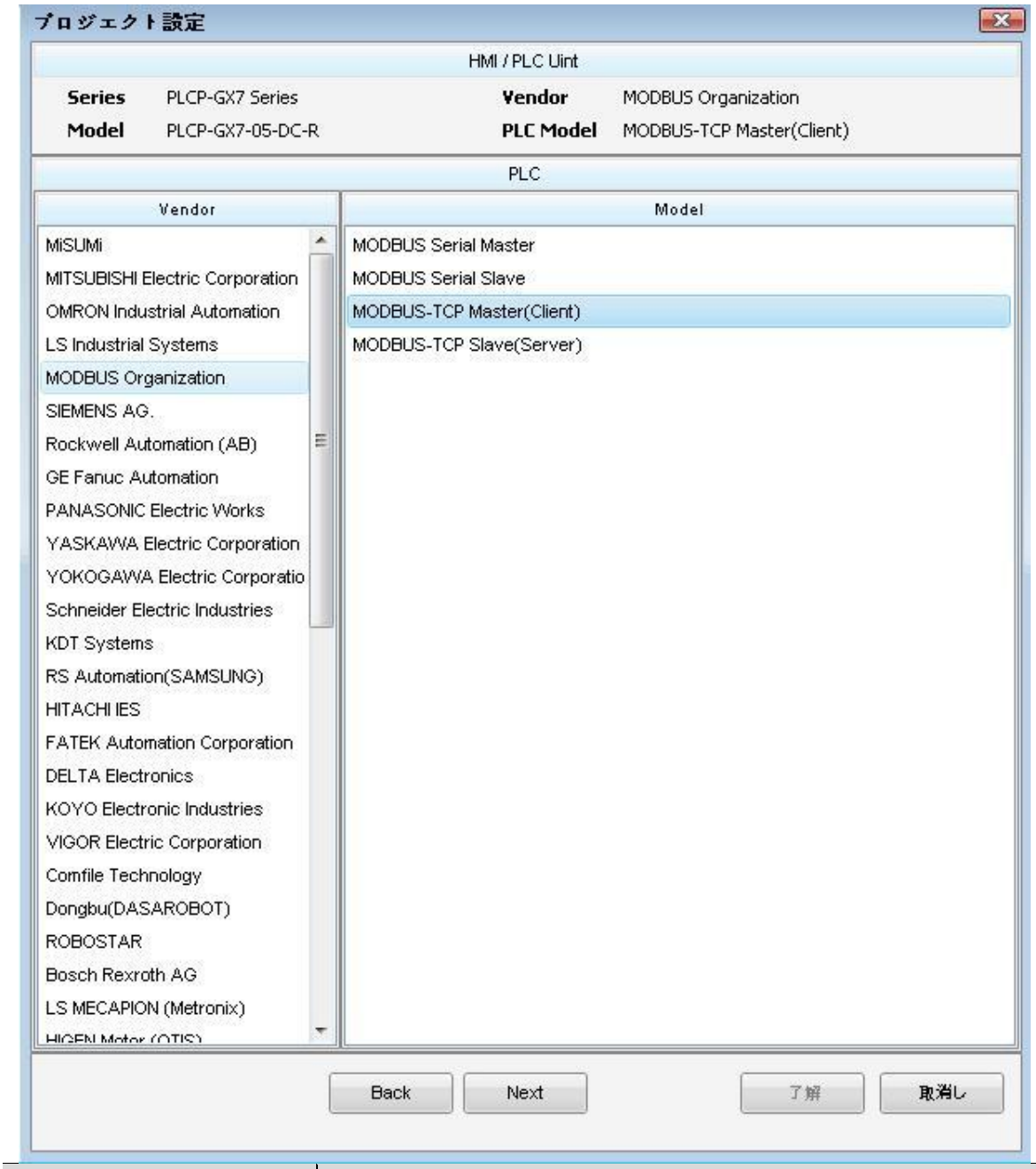


・1：N接続(GX7 1台と外部装置複数台)接続



2. GX7機種と外部装置選択

GX7と接続される外部装置を選択します。



設定事項		内容
GX7	Series	PLCと接続するGX7のシリーズ名称を選択します。 設定内容をDownloadする前にGX7のシリーズによって下の表に明示されたバージョンのOSをインストールしてください。 シリーズ：XGX7 / HGX7 バージョン名称：V4.0
	Name	GX7製品のモデル名を選択します。
外部装置	製造会社	GX7と接続する外部装置の製造会社を選択します。 “MODBUS Organization, Inc.”を選択してください。
	PLC	GX7に接続される外部装置のモデルシリーズを選択します。 “MODBUS TCP Client(Master)”を選択してください。 接続する外部装置がシステム構成可能な機種なのか1章のシステム構成で確認してください。

3. システム設定例題

GX7と"MODBUS TCP Slave Device"の通信インターフェース設定を下記のとおり推奨します。

3.1 設定例題 1

構成したシステムを下記のとおり設定します。

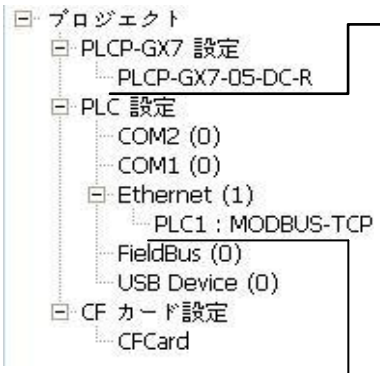
項目	GX7	Slave Device	備考
IP Address*注1)注2)	192.168.0.50	192.168.0.51	ユーザー設定
ポート	Don`t Care	502	ユーザー設定
プロトコル	UDP		ユーザー設定

*注1) GX7と外部装置のネットワークアドレス(IP前の3桁 192.168.000)は一致しなければならない。

*注2) 同じネットワーク上で重複したIPアドレスを使用しないでください。

(1) XDesignerPlus設定

[プロジェクト > プロジェクト設定]で下の内容を設定後、GX7機器に設定内容をダウンロードします。



■ [プロジェクト > プロジェクトプロパティ > プロジェクト > 設定 > GX7 Name].

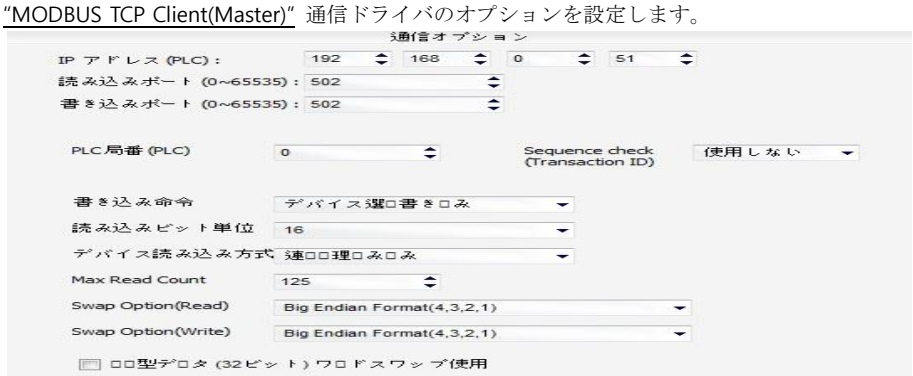
GX7機器の通信インターフェースを設定します。

– 右ウインドウで [HMI設定 > HMI設定使用チェック > 装置管理者]

– 右ウインドウで [HMI設定 > HMI設定使用チェック > PLC設定]

■ 外部装置設定

"MODBUS TCP Client(Master)" 通信ドライバのオプションを設定します。



– IPアドレス(PLC): 外部装置に割り当てたIP番号を記入します。

– 読み込みポート / 書き込みポート: 外部装置のイーサネット通信に使用するポート番号を選択します。

– PLC局番(PLC) : 外部装置設定局番

– Sequence check(Transaction ID) : 処理する過程をもう一度チェックします。

– 実数型データ(32ビット)ワードスワップ使用 : 実数型データのHigh/ Low Wordの順序をLow/High Wordの順に設定します。

(2) 外部装置設定

外部装置のユーザーマニュアルを参照して外部機器I/Fに"MODBUS Serial Slave Driver"を設定してください。

- 同じネットワーク上で重複したIPアドレスを使用しないでください。
- 外部装置側のアドレスマップの内容を確認し、その内容に従って通信アドレスを使ってください。

3.2 設定例題 2

構成したシステムを下記のとおり設定します。

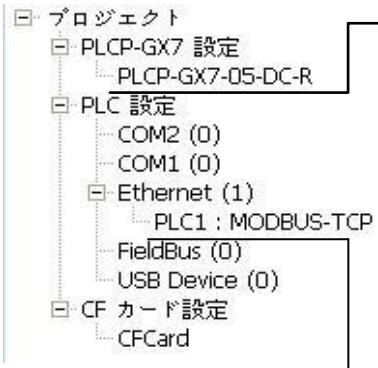
項目	GX7	Slave Device	備考
IP Address*注1)注2)	192.168.0.50	192.168.0.51	ユーザー設定
ポート	Don`t Care	502	ユーザー設定
プロトコル	TCP		ユーザー設定

*注1) GX7と外部装置のネットワークアドレス(IP前の3桁 192.168.000)は一致しなければならない。

*注2) 同じネットワーク上で重複したIPアドレスを使用しないでください。

(1) XDesignerPlus設定

[プロジェクト > プロジェクト設定]で下の内容を設定後、GX7機器に設定内容をダウンロードします。



■ [プロジェクト > プロジェクトプロパティ > プロジェクト > 設定 > GX7 Name].

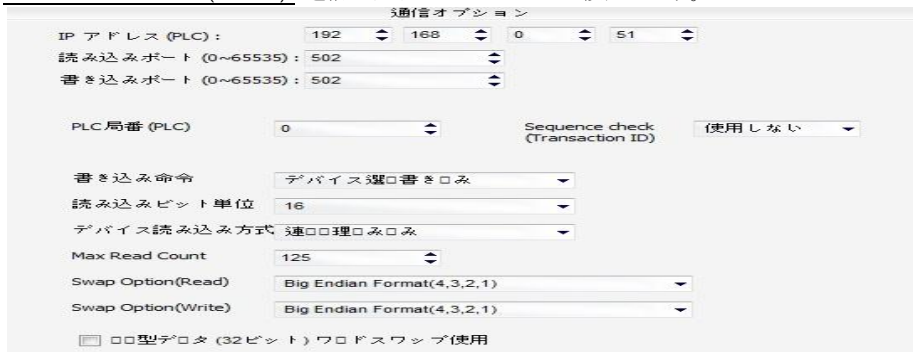
GX7機器の通信インターフェースを設定します。

– 右ウインドウで [HMI設定 > HMI設定使用チェック > 装置管理者]

– 右ウインドウで [HMI設定 > HMI設定使用チェック > PLC設定]

■ 外部装置設定

"MODBUS TCP Client(Master)" 通信ドライバのオプションを設定します。



– IPアドレス(PLC): 外部装置に割り当てたIP番号を記入します。

– 読み込みポート / 書き込みポート: 外部装置のイーサネット通信に使用するポート番号を選択します。

– PLC局番(PLC) : 外部装置設定局番

– Sequence check(Transaction ID) : 処理する過程をもう一度チェックします。

– 実数型データ(32ビット)ワードスワップ使用 : 実数型データのHigh/ Low Wordの順序をLow/High Wordの順に設定します。

(2) 外部装置設定

外部装置のユーザーマニュアルを参照して外部機器I/Fに"MODBUS Serial Slave Driver"を設定してください。

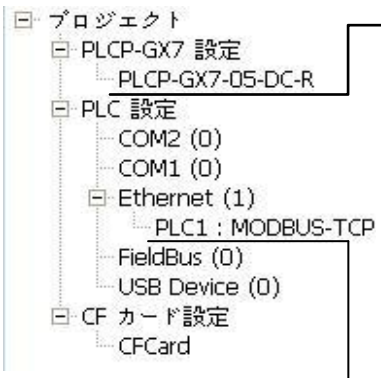
- 同じネットワーク上で重複したIPアドレスを使用しないでください。
- 外部装置側のアドレスマップの内容を確認し、その内容に従って通信アドレスを使ってください。

4. 通信設定項目

通信設定はXDesignerPlusもしくはGX7メインメニューで設定できます。通信設定は外部装置と同じように設定してください。

4.1 XDesignerPlus設定項目

下の画面の内容を示すために[プロジェクト > プロジェクトプロパティ]を選択してください。



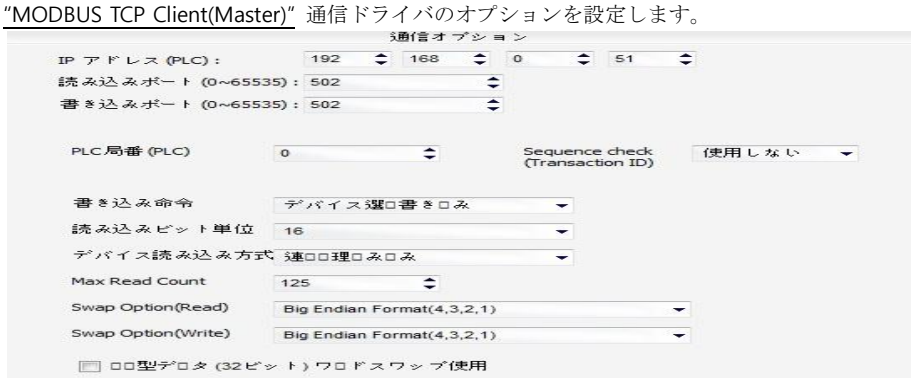
■ [プロジェクト > プロジェクトプロパティ > プロジェクト > 設定 > GX7 Name].
GX7機器の通信インターフェースを設定します。

– 右ウインドウで [HMI設定 > HMI設定使用チェック > 装置管理者]

– 右ウインドウで [HMI設定 > HMI設定使用チェック > PLC設定]

■ 外部装置設定

"MODBUS TCP Client(Master)" 通信ドライバのオプションを設定します。

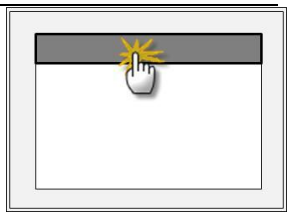


■ 通信インターフェース設定

項目	内容
IPアドレス	ネットワーク上でGX7に付与するIPアドレスを設定します。
サブネットマスク	ネットワークのサブネットマスクを記入します。
ゲートウェイ	ネットワークのサブネットマスクを記入します。
PLC IPアドレス	外部装置に割り当てたIP番号を記入します。
読み込みポート / 書き込みポート	外部装置のイーサネット通信に使用するポート番号を選択します。
GX7ポート	外部装置とイーサネット通信する場合、ポート番号は自動設定されます。
PLC局番 [0~65535]	相手機器の局番です。[0 – 65535]までの値を選択します。
イーサネットタイムアウト	GX7が外部装置からの応答を待機する時間を[0 – 99] x 100 mSecに設定します。
送信前の遅延時間 [x1 mSec]	GX7が外部装置からの応答受信 – 次のコマンド要請転送の間に待機する時間を[0 – 5000] x 1 mSecに設定します。
プロトコル	外部装置と設定ポート番号によって許容されたプロトコル方式を選択します。

4.2 GX7メインメニュー設定項目

- 電源をリセット中にブザー音が鳴るとき、LCD上端1点をタッチして"GX7管理メイン"画面に移動します。
- GX7でドライバインターフェース設定は下記の**Step1 → Step2**の内容に従って設定します。
  (**Step 1.**で"GX7イーサネット設定"を押せば、**Step2.**で設定を変えることができます。)



Step 1. [PLC設定] – ドライバインターフェースを設定します。

PLC設定	
PLC IP : 192 . 168 . 0 . 51 プロトコル : UDP PLC読み込みポート : 502 PLC書き込みポート : 502 GX7ポート : 1024 PLC局番 : 0 タイムアウト : 1000 [mSec] 送信前の遅延時間 : 0[mSec] GX7 IP : 192 . 168 . 0 . 50 GX7イーサネット設定 通信診断	通信インターフェース設定

Step 1–Reference.

項目	内容
PLC IP	外部装置に割り当てたIP番号です。
プロトコル	外部装置と設定ポート番号によって許容されたプロトコル方式を選択します。
PLC読み込みポート	外部装置のイーサネット通信に使用するポート番号です。
PLC書き込みポート	外部装置のイーサネット通信に使用するポート番号です。
GX7ポート	外部装置とイーサネット通信する場合、ポート番号は自動設定されます。
PLC局番[0~65535]	相手機器の局番です。[0 – 65535]までの値を選択します。
タイムアウト [x1 mSec]	GX7が外部装置からの応答を待機する時間を[0 – 5000] x 1 mSecに設定します。
送信前の遅延時間 [x1 mSec]	GX7が外部装置からの応答受信 – 次のコマンド要請転送の間に待機する時間を[0 – 5000] x 1 mSecに設定します。
GX7 IP	ネットワーク上でGX7に付与するIPアドレスを設定します。

Step 2. [PLC設定] >[GX7イーサネット設定] – 該当ポートのシリアルパラメータを設定します。

ポート設定	
* イーサネット通信 + ネットワーク設定 - MAC : 00 – 15 – ID – 00 – 30 – 52 (機器によって異なる固有アドレス) - IPアドレス : 192 . 168 . 0 . 50 - サブネットマスク : 255 . 255 . 255 . 0 - ゲートウェイ : 192 . 168 . 0 . 1	イーサネットポート 通信インターフェース設定

Step 2-Reference.

項目	内容
MAC	ネットワーク上の物理的な固有アドレスです。
IPアドレス	ネットワーク上でGX7に付与するIPアドレスを設定します。
サブネットマスク	IPアドレスに対するネットワークIDとホストIDを区分するアドレスです。
ゲートウェイ	ネットワークと他のネットワークが接続するアドレスです。

4.3 通信診断

■ GX7 – 外部装置間のインターフェース設定状態を確認

- GX7の電源をリセットし、LCD画面の上端をクリックしてメニュー画面に移動する。
- [メインメニュー >通信設定] 20~24番の内容が"■設定例題1"の設定内容と同じか確認する。
- PLC設定 > GX7イーサネット"通信診断"のボタンをクリックする。
- 画面上にDiagnosticsダイアログボックスがポップアップされ、ボックスの3番項に表示された内容によって診断状態を判断する。

OK!	**通信設定　正常**
Time Out Error!	通信設定　非正常 - ケーブルおよびGX7/外部装置の設定状態エラー **(参照 : 通信診断シート)**

■ 通信診断シート

- 外部端末と通信接続に問題がある場合、下記のシートの設定内容を確認してください。

項目	内容					確認	
GX7	バージョン情報		xDesignerPlus :	O.S :			
	ドライバ名称					OK	NG
	外部装置情報 (xDesignerPlusのプロジェクト設定)	IP Address				OK	NG
		サブネットマスク				OK	NG
		ゲートウェイ				OK	NG
	GX7情報 (本体メニュー設定)	プロトコル	UDP/IP	TCP/IP		OK	NG
		IP Address				OK	NG
		サブネットマスク				OK	NG
		ゲートウェイ				OK	NG
	その他細部設定事項					OK	NG
システム構成	システム接続方法		1:1	1:N	N:1	OK	NG
	ケーブル名称(ハブ使用有無)		ダイレクト(ハブ使用)	クロス(ハブ未使用)		OK	NG
外部装置	CPU名称					OK	NG
	通信モジュール名称					OK	NG
	プロトコル(モード)					OK	NG
	その他細部設定事項					OK	NG
	IP Address		(Local)	(Destination)		OK	NG
	ポート番号		(Local)	(Destination)		OK	NG
	サブネットマスク					OK	NG
	ゲートウェイ					OK	NG
	アドレス範囲確認(別途資料)					OK	NG

5. サポートアドレス

GX7で使用可能なデバイスは下記のとおりです。

CPUモジュールシリーズ/タイプによってデバイス範囲(アドレス)に差があることがあります。GX7シリーズは外部装置シリーズが使用する最大アドレス範囲をサポートします。使用する装置がサポートするアドレス範囲から出ないように各CPUモジュールユーザーマニュアルを参照/注意してください。

	Bit Address	Word Address	32 bits	Remarks
Coil	000001 – 065536	000001 – 065521	L/H	
Discrete Input	100001 – 165536	100001 – 165521		*注1)
Input Register	300001.00 – 365536.15	300001 – 365536		*注1)
Holding Register	400001.00 – 465536.15	400001 – 465536		

***注1)** 書き込み不可能(読み込み専用)

Appendix A. MODBUS TCP/IP ADU Frame(Data Frame)

本機器の"MODBUS TCP Client(Master) Driver"がサポートするMODBUSプロトコルコマンドおよびデバイスについて説明します。

WHAT IS MODBUS?

The MODBUS protocol was developed in 1979 by Modicon, Incorporated, for industrial automation systems and Modicon programmable controllers. It has since become an industry standard method for the transfer of discrete/analog I/O information and register data between industrial control and monitoring devices. MODBUS is now a widely-accepted, open, public-domain protocol that requires a license, but does not require royalty payment to its owner.

MODBUS devices communicate using a master-slave (client-server) technique in which only one device (the Client(Master)) can initiate

transactions (called queries). The other devices (slaves/servers) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device (I/O transducer, valve, network drive, or other measuring device) which processes information and sends its output to the master using MODBUS. The Acromag I/O Modules form slave/server devices, while a typical master device is a host computer running appropriate application software. Other devices may function as both clients (masters) and servers (slaves).

Masters can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually, but do not respond to broadcast queries. Slaves do not initiate messages on their own, they only respond to queries from the master.

A master's query will consist of a slave address (or broadcast address), a function code defining the requested action, any required data, and an error checking field. A slave's response consists of fields confirming the action taken, any data to be returned, and an error checking field. Note that the query and response both include a device address, a function code, plus applicable data, and an error checking field. If no error occurs, the slave's response contains the data as requested. If an error occurs in the query received, or if the slave is unable to perform the action requested, the slave will return an exception message as its response (see MODBUS Exceptions). The error check field of the slave's message frame allows the master to confirm that the contents of the message are valid. Traditional MODBUS messages are transmitted serially and parity checking is also applied to each transmitted character in its data frame.

At this point, It's important to make the distinction that MODBUS itself is an application protocol, as it defines rules for organizing and interpreting data, but remains simply a messaging structure, independent of the underlying physical layer. As it happens to be easy to understand, freely available, and accessible to anyone, it is thus widely supported by many manufacturers.

次のページへ続きます。

WHAT IS MODBUS TCP/IP?

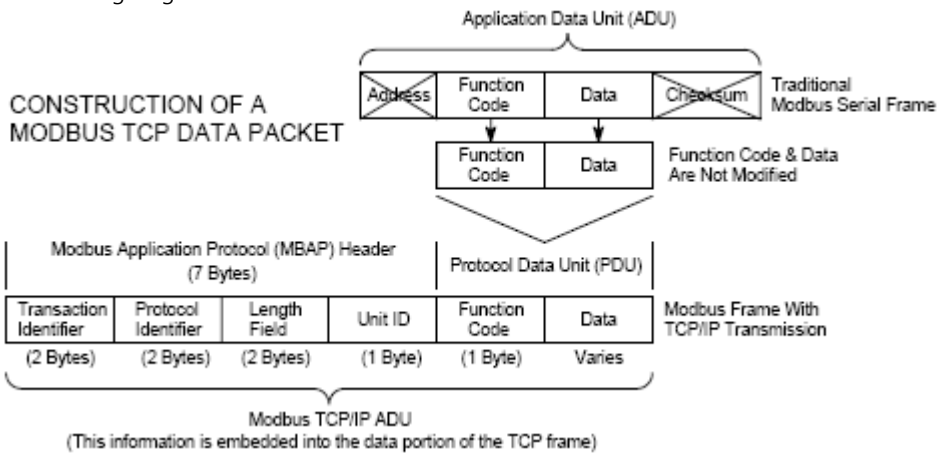
MODBUS TCP/IP (also MODBUS-TCP) is simply the MODBUS RTU protocol with a TCP interface that runs on Ethernet.
The MODBUS messaging structure is the application protocol that defines the rules for organizing and interpreting the data independent of the data transmission medium.

TCP/IP refers to the Transmission Control Protocol and Internet Protocol, which provides the transmission medium for MODBUS TCP/IP messaging.

Simply stated, TCP/IP allows blocks of binary data to be exchanged between computers. It is also a world-wide standard that serves as the foundation for the World Wide Web. The primary function of TCP is to ensure that all packets of data are received correctly, while IP makes sure that messages are correctly addressed and routed. Note that the TCP/IP combination is merely a transport protocol, and does not define what the data means or how the data is to be interpreted (this is the job of the application protocol, MODBUS in this case).

So in summary, MODBUS TCP/IP uses TCP/IP and Ethernet to carry the data of the MODBUS message structure between compatible devices. That is, MODBUS TCP/IP combines a physical network (Ethernet), with a networking standard (TCP/IP), and a standard method of representing data (MODBUS as the application protocol). Essentially, the MODBUS TCP/IP message is simply a MODBUS communication encapsulated in an Ethernet TCP/IP wrapper.

In practice, MODBUS TCP embeds a standard MODBUS data frame into a TCP frame, without the MODBUS checksum, as shown in the following diagram.



The MODBUS commands and user data are themselves encapsulated into the data container of a TCP/IP telegram without being modified in any way. However, the MODBUS error checking field (checksum) is not used, as the standard Ethernet TCP/IP link layer checksum methods are instead used to guaranty data integrity. Further, the MODBUS frame address field is supplanted by the unit identifier in MODBUS TCP/IP, and becomes part of the MODBUS Application Protocol (MBAP) header (more on this later).

From the figure, we see that the function code and data fields are absorbed in their original form. Thus, a Modbus TCP/IP Application Data Unit (ADU) takes the form of a 7 byte header (transaction identifier + protocol identifier + length field + unit identifier), and the protocol data unit (function code + data). The MBAP header is 7 bytes long and includes the following fields:

- **Transaction/invocation Identifier (2 Bytes):** This identification field is used for transaction pairing when multiple messages are sent along the same TCP connection by a client without waiting for a prior response.
- **Protocol Identifier (2 bytes):** This field is always 0 for MODBUS services and other values are reserved for future extensions.
- **Length (2 bytes):** This field is a byte count of the remaining fields and includes the unit identifier byte, function code byte, and the data fields.
- **Unit Identifier (1 byte):** This field is used to identify a remote server located on a non TCP/IP network (for serial bridging). In a typical MODBUS TCP/IP server application, the unit ID is set to 00 or FF, ignored by the server, and simply echoed back in the response.

The complete MODBUS TCP/IP Application Data Unit is embedded into the data field of a standard TCP frame and sent via TCP to well-known system port 502, which is specifically reserved for MODBUS applications. MODBUS TCP/IP clients and servers listen and receive MODBUS data via port 502.
We can see that the operation of MODBUS over Ethernet is nearly transparent to the MODBUS register/command structure. Thus, if you are already familiar with the operation of traditional MODBUS, then you are already very with the operation of MODBUS TCP/IP.

(1) Read Single Coil : 01

MASTER機器でSlave機器側(局番:17番)の“**0**00020–**0**00056 Coil”データを読み込む例題を通じて“01”コマンドフレームを説明します。

■ RTU Mode

(Master → Slave : 要請フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		デバイス点数	
	H	L	H	L	H	L			H	L	H	L
Hex	00	01	00	00	00	06	11	01	00	13	00	25

(Slave → Master : 応答フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	データ個数(byte)	データ				
										Coils 27~20	Coils 35~28	Coils 43~36	Coils 51~44	Coils 56~52
	H	L	H	L	H	L				L	–	–	–	H
Hex	00	01	00	00	00	08	11	01	05	CD	6B	B2	0E	1B

■ Coilsデータ状態

Coils	27	26	25	24	23	22	21	20
on/off	1	1	0	0	1	1	0	1
Coils	35	34	33	32	31	30	29	28
on/off	0	1	1	0	1	0	1	1
Coils	43	42	41	40	39	38	37	36
on/off	1	0	1	1	0	0	1	0
Coils	51	50	49	48	47	46	45	44
on/off	0	0	0	0	1	1	1	0
Coils	59	58	57	56	55	54	53	52
on/off	–	–	–	1	1	0	1	1

0: OFF /

(2) Force Single Coil : 05

MASTER機器でSlave機器側のCoil **0**00173にFORCE “ON”する例題を通じて“05”コマンドフレームを説明します。

■ RTU Mode

(Master → Slave : 要請フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		Force data	
	H	L	H	L	H	L			H	L	H	L
Hex	00	02	00	00	00	06	11	05	00	AC	FF	00

■ Force Data

	High	Low
Force ON	FF_H	00_H
Force OFF	00_H	00_H

(Slave → Master : 応答フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		Force data	
	H	L	H	L	H	L			H	L	H	L
Hex	00	02	00	00	00	06	11	05	00	AC	FF	00

(1) Read Input Status : 02

MASTER機器でSlave機器側(局番:17番)の”**1**00197–**1**00218 Input”データを読み込む例題を通じて”02” コマンドフレームを説明します。

(Master → Slave : 要請フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		デバイス点数	
	H	L	H	L	H	L			H	L	H	L
Hex	00	03	00	00	00	06	11	02	00	C4	00	16

(Slave → Master : 応答フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	データ個数(byte)	データ(Inputs)		
										10204~10197	10212~10205	10218~10213
	H	L	H	L	H	L						
Hex	00	03	00	00	00	06	11	02	03	AC	DB	35

■ Coilsデータ状態

Coils	204	203	202	201	200	199	198	197
on/off	1	0	1	0	1	1	0	0
Coils	212	211	210	209	208	207	206	205
on/off	1	1	0	1	1	0	1	1
Coils	220	219	218	217	216	215	214	213
on/off	–	–	1	1	0	1	0	1

0: OFF / 1:ON

(1) Read Input Registers : 04

MASTER機器でSlave機器側(局番:17番)の”**3**00009 Register”データを読み込む例題を通じて“03”コマンドフレームを説明します。

(Master → Slave : 要請フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		デバイス点数 (Word Count)	
	H	L	H	L	H	L			H	L	H	L
Hex	00	04	00	00	00	06	11	04	00	08	00	01

(Slave → Master : 応答フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	データ個数(byte)	データ Register 30009	
	H	L	H	L	H	L				H	L
Hex	00	04	00	00	00	05	11	04	02	00	0A

(1) Read Holding Registers : 03

MASTER機器でSlave機器側(局番:17)の“**4**00108 – **4**00110 Register“データを読み込む例題を通じて“03”コマンドフレームを説明します。

(Master → Slave : 要請フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		デバイス点数	
	H	L	H	L	H	L			H	L	H	L
Hex	00	05	00	00	00	06	11	03	00	6B	00	03

(Slave → Master : 応答フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	データ個数(byte)	データ					
										Register 40108		Register 40109		Register 40110	
	H	L	H	L	H	L				H	L	H	L	H	L
Hex	00	05	00	00	00	09	11	03	06	02	2B	00	00	00	64

(2) Preset Single Register : 06

Slave機器側の**4**00002 Registerに00 03 (hex)データを入力する例題を通じて“06”コマンドフレームを説明します。

(Master → Slave : 要請フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		Preset data	
	H	L	H	L	H	L			H	L	H	L
Hex	00	06	00	00	00	06	11	06	00	01	00	03

(Slave → Master : 応答フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		Preset data	
	H	L	H	L	H	L			H	L	H	L
Hex	00	06	00	00	00	06	11	06	00	01	00	03

(3) Preset Multiple Register : 10

Slave機器側の**4**00002 Registerに“00 0A (hex)”, “01 02 (hex)”連続した2つのデータを入力する例題を通じて“10”コマンドフレームを説明します。(Error Code : 90_H)

(Master → Slave : 要請フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		Quantity of Register (Word Count)		データ個数(Byte)	データ			
														Register 40002		Register 40003	
	H	L	H	L	H	L			H	L	H	L		H	L	H	L
Hex	00	07	00	00	00	0B	11	10	00	01	00	02	04	00	0A	01	02

(Slave → Master : 応答フレーム)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭デバイス		Quantity of Register (Word Count)	
	H	L	H	L	H	L			H	L	H	L
Hex	00	07	00	00	00	06	11	10	00	01	00	02