

MODBUS Organization

MODBUS Master Series

MODBUS-TCP Client (Master) Driver

2023/10/12

バージョン

GX8 Design Studio

V1.4 以上

CONTENTS

当社(株)ミスミのTouch Operation Panel(ミスミ GX8) Seriesをご使用いただきまして誠にありがとうございます。本マニュアルを読み、「GX8 -外部機器」の接続方法と手順をご確認ください。

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、設定可能なシステムについて説明します。
本項を参照し、適切なシステムを選定してください。

2. 外部機器の選択

[3 ページ](#)

↓
GX8の機種と外部機器を選択します。

3. GX8通信設定

[4 ページ](#)

↓
GX8の通信設定の方法について説明します。
外部機器の設定が変更された場合、本項を参照し GX8通信設定と外部機器を同じ設定にしてください。

4. 外部機器の設定

[12 ページ](#)

↓
外部機器の通信設定方法について説明します。

5. サポートアドレス

[13 ページ](#)

↓
本項を参照して、外部機器との通信可能なアドレスを確認してください。

1. システム構成

本ドライバは、MODBUS OrganizationのMODBUS ProtocolのMODBUS-TCP Client (Master)です。

外部機器（MODBUS Slave Protocolサポート）に応じて、ドライバの「コマンドコード」、「プロトコルのフレーム形式」などを別途設定する必要があります。この場合、通信方式に応じた詳細な設定を外部機器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機器のシステム構成は以下の通りです。

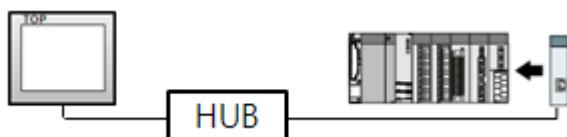
シリーズ	CPU	Link I/F	通信方式	システム設定	ケーブル
		MODBUS Slave/Server Device	Ethernet (TCP / UDP)	3. GX8通信設定 4. 外部機器の設定	ツイストペアケーブル ^{*注1)}

*注1) ツイストペアケーブル

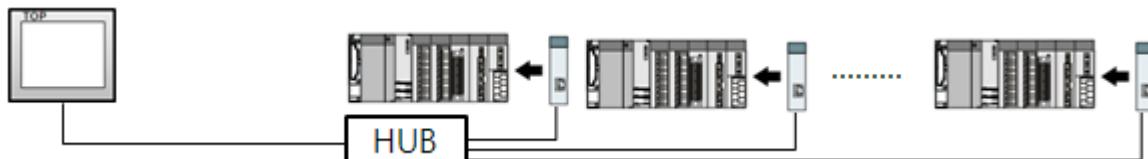
- STP（シールドツイストペアケーブル）、あるいはUTP（非シールドツイストペアケーブル）カテゴリー3, 4, 5を意味します。
- ネットワーク構成に応じて、ハブ、トランシーバなどの構成機器に接続可能であり、この場合、ダイレクトケーブルを使ってください。

■接続構成

- ・ 1 : 1 (GX8 1台と外部機器1台) 接続

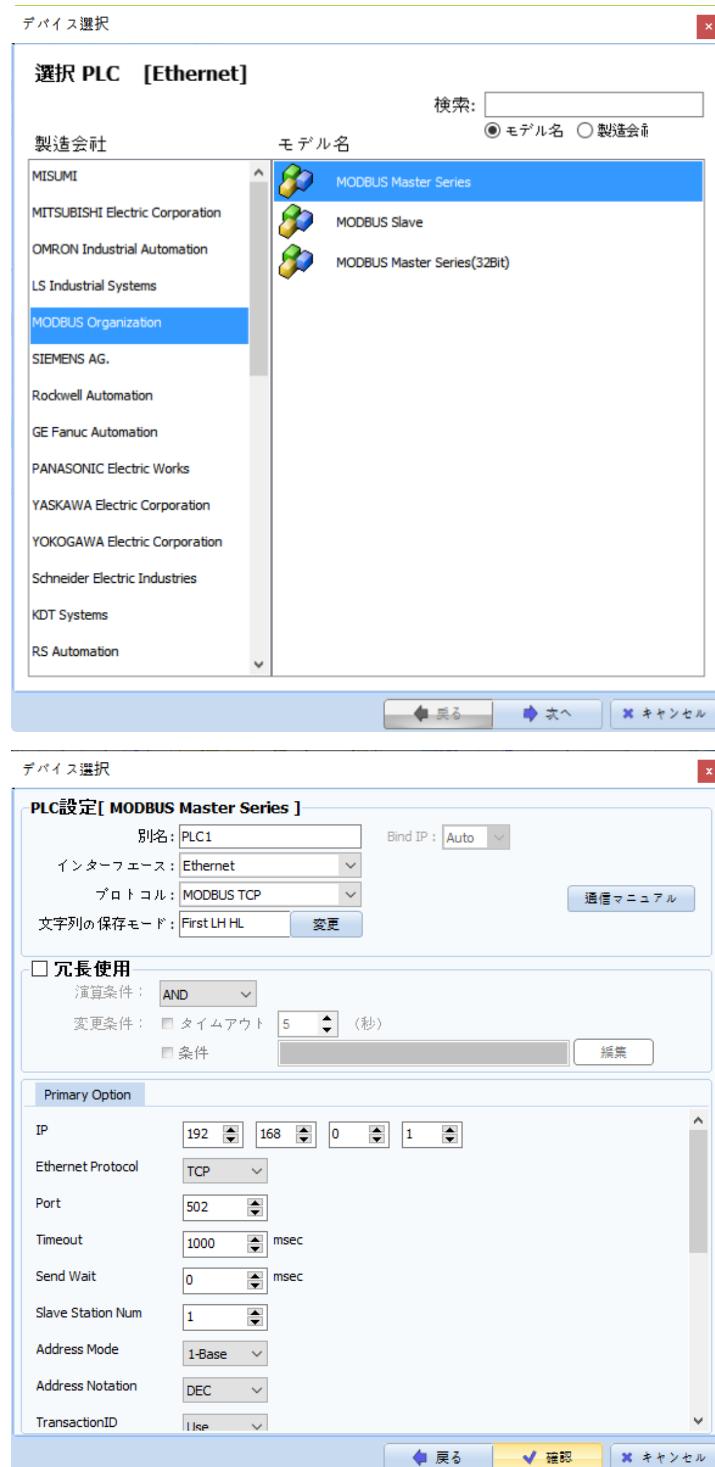


- ・ 1 : N (GX8 1台と外部機器数台) 接続



2. 外部機器の選択

■GX8 モデル及びポートを選択した後、外部機器を選択します。



設定事項		内容					
GX8	モデル	GX8 のディスプレイとプロセスを確認し、タッチモデルを選択します。					
外部機器	メーカー	GX8 と接続する外部機器のメーカー「MODBUS Organization」を選択します。					
	PLC	GX8と接続する外部機器を選択します。 <table border="1"><tr><th>モデル</th><th>インターフェース</th><th>プロトコル</th></tr><tr><td>MODBUS Master Series</td><td>Ethernet</td><td>MODBUS TCP</td></tr></table> <p>接続したい外部機器が、システムの構成可能な機種であることを1項目のシステム構成で確認してください。</p>	モデル	インターフェース	プロトコル	MODBUS Master Series	Ethernet
モデル	インターフェース	プロトコル					
MODBUS Master Series	Ethernet	MODBUS TCP					

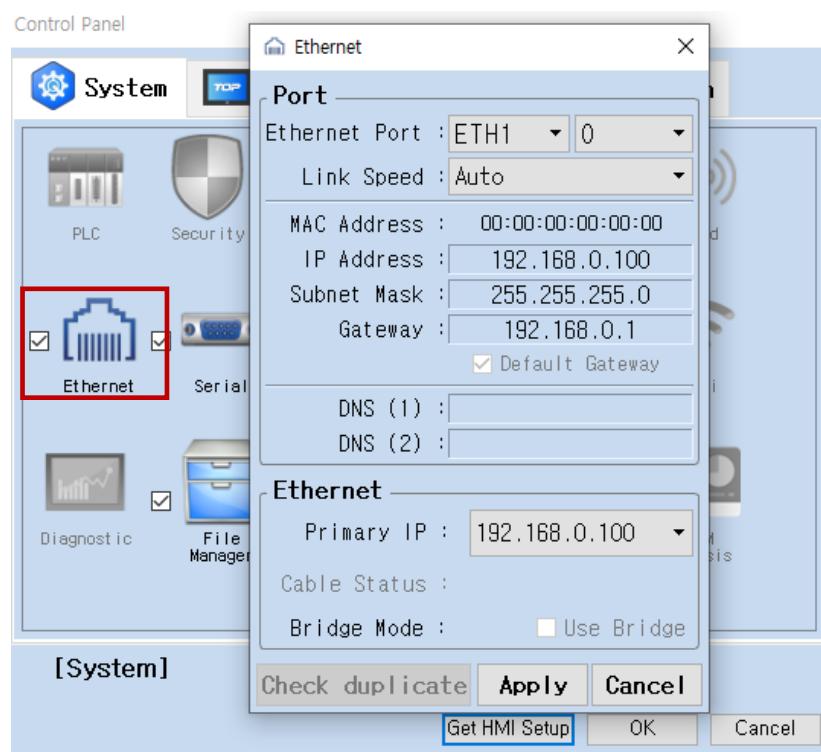
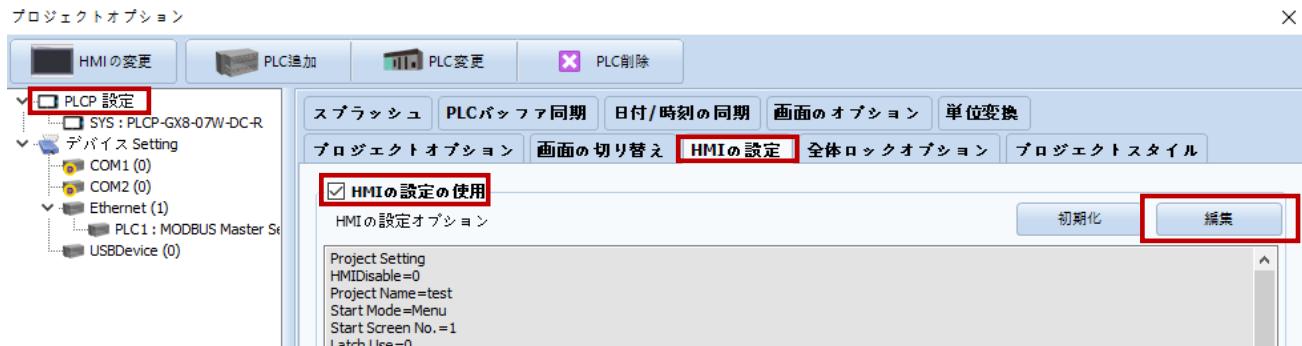
3. GX8通信設定

通信設定はGX8 Design Studio、またはGX8 メインメニューから設定可能です。通信設定は外部機器と同一に設定する必要があります。

3.1 GX8 Design Studioでの通信設定

(1) 通信インターフェースの設定

- [プロジェクト>プロジェクトのプロパティ>GX8 設定]→[プロジェクトオプション>「HMIの設定使用」チェック>編集>イーサネット]
– GX8 通信インターフェースをGX8 Design Studioで設定します。



項目	GX8	外部機器	備考
IPアドレス*注1)注2)	192.168.0.100	192.168.0.51	
サブネットマスク	255.255.255.0	255.255.255.0	
ゲートウェイ	192.168.0.1	192.168.0.1	

*注1) GX8 と外部機器のネットワークアドレス（IPの前三桁192.168.0.0）は一致する必要があります。

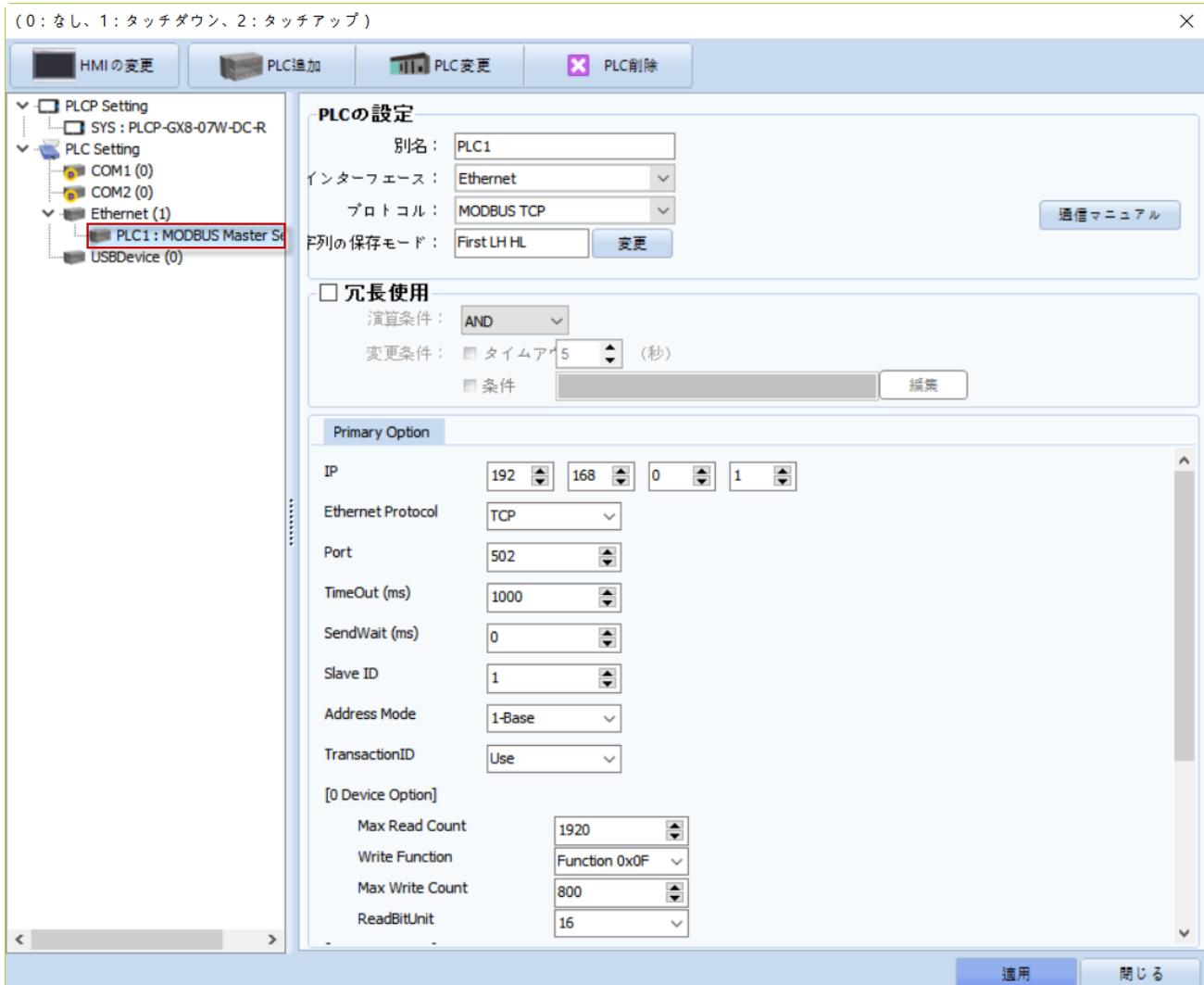
*注2) 同一ネットワーク上で、IPアドレスは重複して使用しないでください。

※上記の内容は、当社が推奨する設定例です。

項目	説明
IPアドレス	ネットワーク上でGX8 が使用する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入力します。
ゲートウェイ	ネットワークのゲートウェイを入力します。

(2) 通信オプションの設定

- [プロジェクト > プロジェクトのプロパティ > PLC設定 > ETHERNET(1) > PLC1 : MODBUS Master Series]
 - MODBUS-TCP Client (Master) 通信ドライバのオプションをGX8 Design Studioで設定します。



項目	設定	備考
インターフェース	Ethernetを選択します。	「2. 外部機器の選択」参考
プロトコル	MODBUS TCPを選択します。	
文字列の保存モード	文字列の保存方法を選択します。	
冗長	冗長使用	冗長設定を使用の有無をチェックします。
	演算条件	変更条件に応じて、演算条件を設定します。 AND : チェックされた変更条件すべてが満たされた場合、Primary↔Secondaryに変更 OR : チェックされた変更条件のいずれかが満たされた場合、Primary↔Secondaryに変更
	変更条件	Primary ↔ Secondary 変更条件を設定します。
IP	外部機器のIPアドレスを入力します。	
Ethernet Protocol	GX8 -外部機器間のイーサネットプロトコルを選択します。	
Port	外部機器のイーサネット通信ポート番号を入力します。	
TimeOut (ms)	GX8 の外部機器からの応答を待つ時間を設定します。	
SendWait (ms)	GX8 の外部機器からの応答を受信し、次のコマンド要求の送信までの待機時間を設定します。	
Slave Station Num	外部機器(Slave)の局番を入力します。	
Address Mode	アドレスの入力方式を選択します。 「1-base」に設定した場合、通信時に作画に入力されたアドレスに1を引いた値に要求されます。 例) 400001を登録すると、Holding Registerの0000 (hex) を要求	
Address Notation	アドレス表記方式を選択します。	
Transaction ID	モードバスTCPパケットのTransaction ID適用の有無を設定します。 未使用時には0に固定	
[0 Device Option]	Coil (R/W)	
Max Read Count	Coilを読み取り要請時 1 回に要請できる最大数を設定します。	*注1) *注2)
Write Function	Coil書き込み要請命令語を設定します。 0x05 : Force Single Coil (1ビット単位で書き込み。ビット単位動作のみ可能) 0x0F : Force Multiple Coils (16ビット単位で書き込み) Auto : データ数によって0x05/0x0F要請	*注3)
Max Write Count	Coilの書き込み要請時最大要請数を設定します。	*注2)
Read Bit Unit	Coil読み取り要請時要請するビット数を設定します。 設定値が16で画面に続くアドレスが登録された場合 1 回に最大"Max Read Count"数分データを要請します。	
[1 Device Option]	Discrete Input (READ ONLY)	
Max Read Count	Discrete Input読み取り要請時 1 回に要請できる最大数を設定します。	*注1) *注2)
Read Bit Unit	Discrete Input読み取り要請時要請するビット数を設定します。 設定値が16で画面に続くアドレスが登録された場合一回に最大"Max Read Count"数分データを要請します。	
[3 Device Option]	Input Register(READ ONLY)	
Max Read Count	Input Register読み取り要請時 1 回に要請できる最大数を設定します。	*注1) *注2)
[4 Device Option]	Holding Register(R/W)	
Max Read Count	Holding Register読み取り要請時 1 回に要請できる最大数を設定します。	*注1)
Write Function	Holding Register書き込み要請命令語を設定します。 0x06 : Preset Single Register (1個書き込み) 0x10 : Preset Multiple Registers (N個書き込み) Auto : データ数によって0x06/0x10に要請	*注3)
Max Write Count	命令語を0x10にHolding Registerデータ書き込み要請時 1 回に要請できる最大数を設定します。	*注2)

*注1)

- 各デバイスのMax Read Countは画面に登録されたアドレスたちが連続していない通信を南海することではなく 1 回に要請するアドレス範囲としても使用されます。

例1)画面に数字オブジェクトで400001, 400002, 400003, 400004, 400005, 400120を登録し4デバイスのMax Read Countを120に設定する場合400001から400120までを連続したアドレスに仮定し400001から120ワードを一回の要請で読み取ります。

例2) 画面に数字オブジェクトで400001, 400002, 400003, 400004, 400005, 400120を登録し4デバイスのMax Read Countを3に設定す

る場合400001から400003まで3ワード、400004から400005まで2ワード、400120の1ワードとして3回の要請でデータを読み取ります。

例3) 画面に数字オブジェクトで400001, 400010, 400011, 400021, 400031, 400041を登録し4デバイスのMax Read Countを10に設定した場合400001から400010まで10ワード400011 1ワード、400021 1ワード、400031 1ワード、400041 1ワードとして5回の要請でデータを読み取ります。

*注2)

- 外部装置のマニュアルを参考し登録したアドレスか一回に何個のデータの読み取り/書き込みが可能か確認してください。

外部装置が支援している範囲より大きく設定する場合通信が正常に行割れられません。

例) 外部装置のHolding Register(4デバイス)が1回の通信に最大10ワードのみ応答可能な場合、GX8の通信設定の4デバイスMax Read Countを外部装置に合わせ10に設定してください。

*注3)

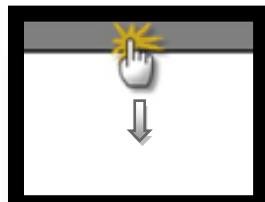
- 外部装置のマニュアルを参考し支援する書き込み命令語に合わせて設定してください。

支援していない書き込み命令語を設定するとデータの書き込み動作が行われません。

3.2 GX8 での通信設定

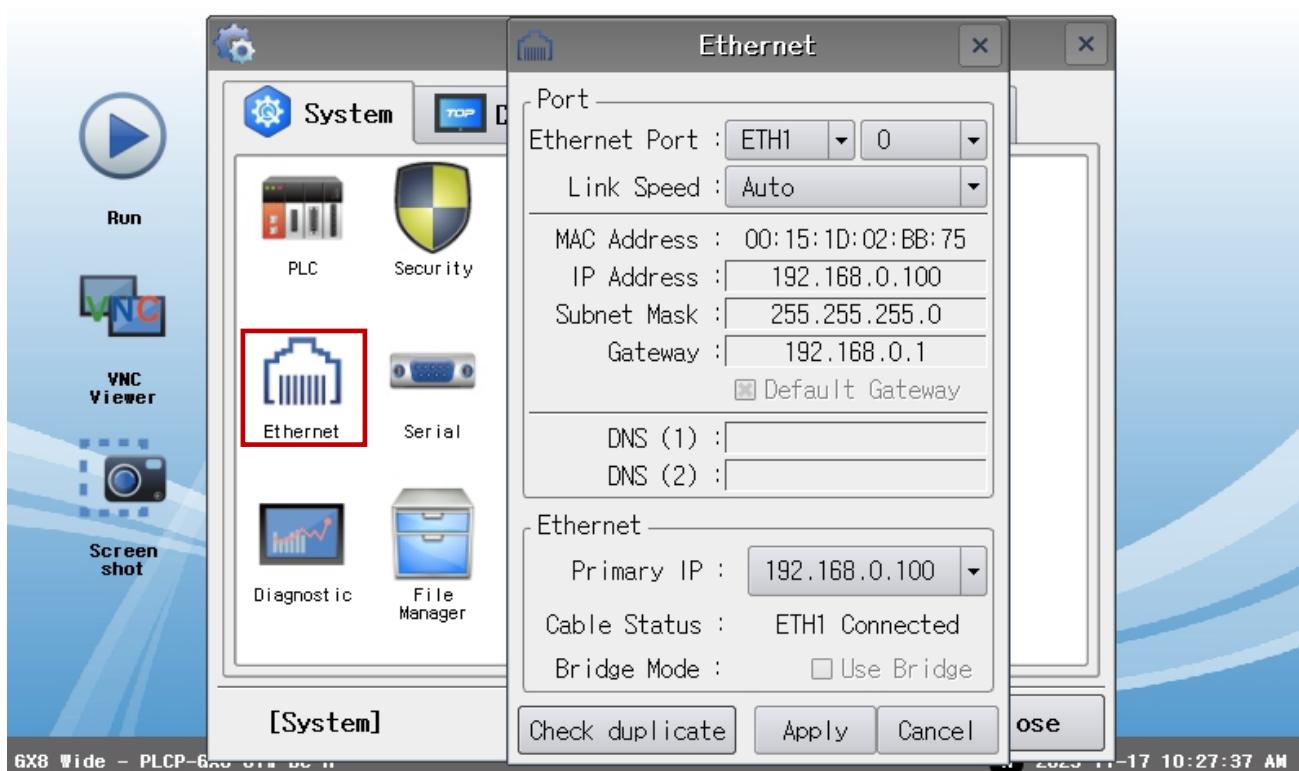
※「3.1 GX8 Design Studioでの通信設定」の「HMIの設定を使用する」をチェックしていない場合の設定方法です。

- GX8 画面上部をタッチして下にドラッグします。ポップアップウィンドウの「EXIT」をタッチして、メイン画面に移動します。



(1) 通信インターフェースの設定

- [メイン画面 > コントロールパネル > イーサネット]



項目	GX8	外部機器	備考
IPアドレス*注1)注2)	192.168.0.100	192.168.0.51	
サブネットマスク	255.255.255.0	255.255.255.0	
ゲートウェイ	192.168.0.1	192.168.0.1	

*注1) GX8 と外部機器のネットワークアドレス (IPの前三桁192.168.0.0) は一致する必要があります。

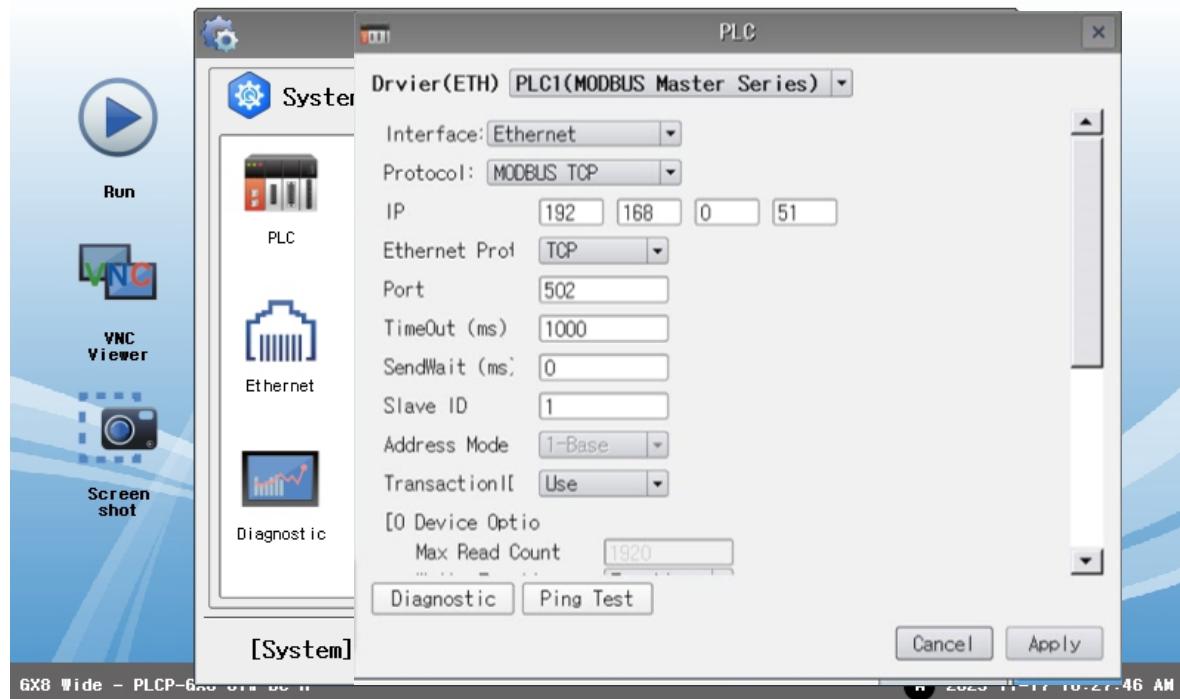
*注2) 同一ネットワーク上で、IPアドレスは重複して使用しないでください。

※上記の内容は、当社が推奨する設定例です。

項目	説明
IPアドレス	ネットワーク上でGX8 が使用する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入力します。
ゲートウェイ	ネットワークのゲートウェイを入力します。

(2) 通信オプションの設定

■ [メイン画面 > コントロールパネル > PLC]



項目	設定	備考
インターフェース	Ethernetを選択します。	「2. 外部機器の選択」参考
プロトコル	MODBUS TCPを選択します。	
文字列の保存モード	文字列の保存方法を選択します。	
冗長	冗長使用	冗長設定を使用の有無をチェックします。
	演算条件	変更条件に応じて、演算条件を設定します。 AND : チェックされた変更条件すべてが満たされた場合、Primary↔Secondaryに変更 OR : チェックされた変更条件のいずれかが満たされた場合、Primary↔Secondaryに変更
	変更条件	Primary ↔ Secondary 変更条件を設定します。
IP	外部機器のIPアドレスを入力します。	
Ethernet Protocol	GX8 -外部機器間のイーサネットプロトコルを選択します。	
Port	外部機器のイーサネット通信ポート番号を入力します。	
TimeOut (ms)	GX8 の外部機器からの応答を待つ時間を設定します。	
SendWait (ms)	GX8 の外部機器からの応答を受信し、次のコマンド要求の送信までの待機時間を設定します。	
Slave Station Num	外部機器(Slave)の局番を入力します。	
Address Mode	アドレスの入力方式を選択します。 「1-base」に設定した場合、通信時に作画に入力されたアドレスに1を引いた値に要求されます。 例) 400001を登録すると、Holding Registerの0000 (hex) を要求	
Address Notation	アドレス表記方式を選択します。	
Transaction ID	モードバスTCPパケットのTransaction ID適用の有無を設定します。 未使用時には0に固定	
[0 Device Option]	Coil (R/W)	
Max Read Count	Coilを読み取り要請時 1 回に要請できる最大数を設定します。	*注1) *注2)
Write Function	Coil書き込み要請命令語を設定します。 0x05 : Force Single Coil (1ビット単位で書き込み。ビット単位動作のみ可能) 0x0F : Force Multiple Coils (16ビット単位で書き込み) Auto : データ数によって0x05/0x0Fを要請	*注3)
Max Write Count	Coilの書き込み要請時最大要請数を設定します。	*注2)
Read Bit Unit	Coil読み取り要請時要請するビット数を設定します。 設定値が16で画面に続くアドレスが登録された場合 1 回に最大 "Max Read Count" 数分	

	データを要請します。	
[1 Device Option]	Discrete Input (READ ONLY)	
Max Read Count	Discrete Input読み取り要請時 1回に要請できる最大数を設定します。	*注1) *注2)
Read Bit Unit	Discrete Input読み取り要請時要請するビット数を設定します。 設定値が16で画面に続くアドレスが登録された場合一回に最大"Max Read Count"数分データを要請します。	
[3 Device Option]	Input Register(READ ONLY)	
Max Read Count	Input Register読み取り要請時 1回に要請できる最大数を設定します。	*注1) *注2)
[4 Device Option]	Holding Register(R/W)	
Max Read Count	Holding Register読み取り要請時 1回に要請できる最大数を設定します。	*注1)
Write Function	Holding Register書き込み要請命令語を設定します。 0x06 : Preset Single Register (1個書き込み) 0x10 : Preset Multiple Registers (N個書き込み) Auto : データ数によって0x06/0x10に要請	*注3)
Max Write Count	命令語を0x10にHolding Registerデータ書き込み要請時 1回に要請できる最大数を設定します。	*注2)

*注1)

- 各デバイスのMax Read Countは画面に登録されたアドレスたちが連続していない通信を南海することではなく1回に要請するアドレス範囲としても使用されます。

例1)画面に数字オブジェクトで400001, 400002, 400003, 400004, 400005, 400120を登録し4デバイスのMax Read Countを120に設定する場合400001から400120までを連続したアドレスに仮定し400001から120ワードを一回の要請で読み取ります。

例2) 画面に数字オブジェクトで400001, 400002, 400003, 400004, 400005, 400120を登録し4デバイスのMax Read Countを3に設定する場合400001から400003まで3ワード、400004から400005まで2ワード、400120の1ワードとして3回の要請でデータを読み取ります。

例3) 画面に数字オブジェクトで400001, 400010, 400011, 400021, 400031, 400041を登録し4デバイスのMax Read Countを10に設定した場合400001から400010まで10ワード400011 1ワード、400021 1ワード、400031 1ワード、400041 1ワードとして5回の要請でデータを読み取ります。

*注2)

- 外部装置のマニュアルを参考し登録したアドレスか一回に何個のデータの読み取り/書き込みが可能か確認してください。

外部装置が支援している範囲より大きく設定する場合通信が正常的に行割れられません。

例)外部装置のHolding Register(4デバイス)が1回の通信に最大10ワードのみ応答可能な場合、GX8の通信設定の4デバイスMax Read Countを外部装置に合わせ10に設定してください。

*注3)

- 外部装置のマニュアルを参考し支援する書き込み命令語に合わせて設定してください。

支援していない書き込み命令語を設定するとデータの書き込み動作が行われません。

3.3通信診断

■GX8 - 外部機器間のインターフェースの設定の状態を確認

- GX8 画面上部をタッチして下にドラッグ。ポップアップウィンドウの「EXIT」をタッチして、メイン画面に移動する
- [コントロールパネル>イーサネット]で使用するポート（ETH1/ETH2）の設定が、外部装置の設定内容と同じであることを確認する

■ポートの通信異常の有無の診断

- [コントロールパネル>PLC]で「通信診断」をタッチする。
- 画面上にDiagnosticsダイアログボックスがポップアップし、診断の状態を表示する。

OK	通信設定の正常
Time Out Error	通信設定の異常 -ケーブルと、外部機器の設定状態を確認する。（参照：通信診断シート）

■通信診断シート

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

項目	内容	確認	参考
システム構成	システムの接続方法	OK	NG
	接続ケーブルの名称	OK	NG
GX8	バージョン情報	OK	NG
	使用ポート	OK	NG
	ドライバ名称	OK	NG
	その他の詳細設定情報	OK	NG
	相手局番	プロジェクトの設定	OK
		通信診断	NG
		IPアドレス	OK
	イーサネットポート設定	サブネットマスク	OK
		ゲートウェイ	OK
		NG	
外部機器	CPUの名称	OK	NG
	通信ポートの名称（モジュール名）	OK	NG
	プロトコル（モード）	OK	NG
	設定局番	OK	NG
	その他の詳細設定情報	OK	NG
	イーサネットポート設定	IPアドレス	OK
		サブネットマスク	OK
		ゲートウェイ	OK
	アドレス範囲チェック	OK	NG
			5. サポートアドレス (詳細については、PLCメーカーのマニュアルを参照してください。)

4. 外部機器の設定

外部機器のユーザーマニュアルを参照し、外部機器I / Fに「MODBUS-TCP Slave (Server) Driver」を設定してください。



- 同一ネットワーク上で、IPアドレスを重複して使用しないでください。
- 外部機器側のアドレスマップの内容を確認し、その内容に応じて通信アドレスを使用します。

5. サポートアドレス

GX8 で使用可能な機器は、以下の通りです。

CPUモジュールシリーズ/タイプに応じて、機器の範囲（アドレス）の差があることがあります。GX8 シリーズは、外部機器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。使用する機器がサポートしているアドレス範囲を超えないように各CPUモジュールユーザーマニュアルを参照して/注意してください。

	Bit Address	Word Address	32 bits	Remarks
Coil	000001 – 065536	000001 – 065521	L/H	*注1)
Discrete Input	100001 – 165536	100001 – 165521		
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（読み取り専用）

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

本機の「MODBUS TCP Client (Master) Driver」がサポートしているMODBUSプロトコルのコマンド及び機器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.

[次ページに続く](#)

WHAT IS MODBUS TCP/IP?

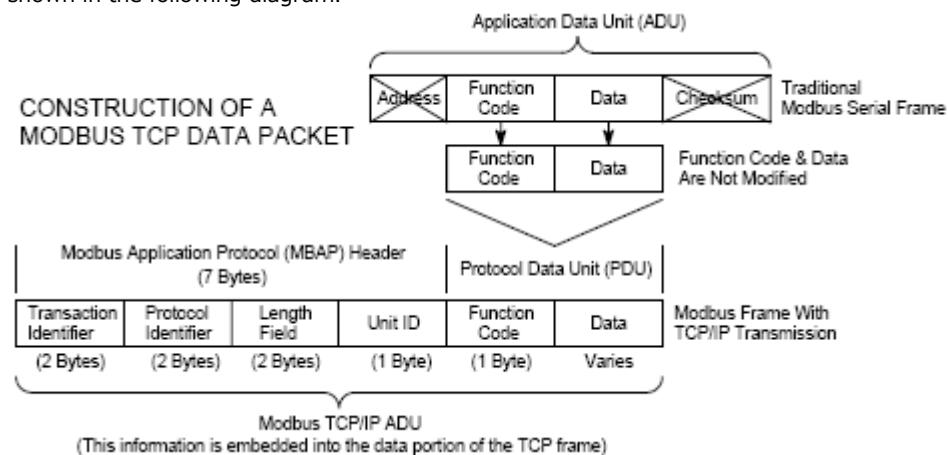
MODBUS TCP/IP (also MODBUS-TCP) is simply the MODBUS RTU protocol with a TCP interface that runs on Ethernet. The MODBUS messaging structure is the application protocol that defines the rules for organizing and interpreting the data independent of the data transmission medium.

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 guarantee 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 familiar with the operation of MODBUS TCP/IP.

A.1 “0” Device (Coil)

(1) Read Single Coil : 01

MASTER機器でSlave機器側（局番：17番）の「000020-000056 Coil」データを読み取る例により、「01」コマンドフレームを説明します。

■ RTU Mode

(Master → Slave : 要求フレーム)

Comment	Hex	Coils												
		Coils				Coils								
機器ID(8bit)								H L						
25								H L						
H 00								H L						
13								H L						
01								H L						
00								H L						
11								H L						

(Slave → Master : 応答フレーム)

Comment	Hex	Coils												
		Coils				Coils								
機器ID(8bit)								H L						
1B								H L						
データ								H L						
Coils								H L						
35~28								H L						
L - - - -								H L						
C 6 B2 0E								H L						
D B								H L						
05								H L						
01								H L						
00								H L						
11								H L						
01								H L						
00								H L						
08								H L						
01								H L						
05								H L						
01								H L						
00								H L						
11								H L						

■ Coils データの状態

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 000173をFORCE「ON」にした例により、「05」コマンドフレームを説明します。

■ RTU Mode

(Master → Slave : 要求フレーム)

Comment	Hex	Force data												
		Force data				Force data								
先頭機器								H L						
FF FF								H L						
00 00								H L						
00 00								H L						
05								H L						
01								H L						
00								H L						
11								H L						

■ 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	Hex	Force data												
		Force data				Force data								
先頭機器								H L						
FF FF								H L						
00 00								H L						
00 00								H L						
05								H L						
01								H L						
00								H L						
11								H L						

A.2 “1” Device (Discrete Input)

(1) Read Input Status : 02

MASTER機器でSlave機器側（局番：17番）の「100197-100218 Input」データを読み取る例により、「02」コマンドフレームを説明します。

(Master → Slave : 要求フレーム)											
Comment											
	Protocol Identifier		Length Field		Unit ID (Slave 地址)		Length Field		Unit ID (Slave 地址)		機器台数
Hex	H 00	L 03	H 00	L 00	H 00	L 06	11	02	H 00	L C4	H 00 L 16
Comment	Transaction Identifier	Protocol Identifier	Length Field	Unit ID (Slave 地址)	Length Field	Unit ID (Slave 地址)	Length Field	Unit ID (Slave 地址)	Length Field	Unit ID (Slave 地址)	機器台数

(Slave → Master : 応答フレーム)											
Comment											
	Protocol Identifier		Length Field		Unit ID (Slave 地址)		Length Field		Unit ID (Slave 地址)		データ(Inputs)
Hex	H 00	L 03	H 00	L 00	H 00	L 06	11	02	03	A C	D B
Comment	Transaction Identifier	Protocol Identifier	Length Field	Unit ID (Slave 地址)	Length Field	Unit ID (Slave 地址)	Length Field	Unit ID (Slave 地址)	Length Field	Unit ID (Slave 地址)	データ(Inputs)

■ Coils データの状態											
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

A.3 “3” Device (Input Register)

(1) Read Input Registers : 04

MASTER機器でSlave機器側（局番：17番）の「300009 Register」データを読み取る例により、「02」コマンドフレームを説明します。

(Master → Slave : 要求フレーム)									
Comment									機器ID(Word Count)
Hex	H	L	H	L	H	L	H	L	
	00	04	00	00	00	06	11	04	01

(Slave → Master : 応答フレーム)									
Comment									データ Register
Hex	H	L	H	L	H	L	H	L	
	00	04	00	00	00	05	11	04	0A

(1) Read Holding Registers : 03

MASTER機器でSlave機器側（局番：17番）の「400108 – 400110 Register」データを読み取る例により、「03」コマンドフレームを説明します。

(Master → Slave : 要求フレーム)									
Comment	Hex	Protocol Identifier	Length Field	Unit ID (Slave 地址)	Register	Register	機器台数	先頭機器	データ
	H 00 L 05	H 00 L 00	H 00 L 06	11	03	H 00 L 6B	H 00 L 00	H 00 L 03	40110 64

(Slave → Master : 応答フレーム)									
Comment	Hex	Protocol Identifier	Length Field	Unit ID (Slave 地址)	Register	Register	機器台数	先頭機器	データ
	H 00 L 05	H 00 L 00	H 00 L 09	11	03	H 02 L 2B	H 00 L 00	H 00 L 03	40109 06

(2) Preset Single Register : 06

Slave機器側の400002 Registerに0003 (hex) のデータを入力する例により、「06」コマンドフレームを説明します。

(Master → Slave : 要求フレーム)									
Comment	Hex	Protocol Identifier	Length Field	Unit ID (Slave 地址)	Register	Register	機器台数	先頭機器	Preset data
	H 00 L 06	H 00 L 00	H 00 L 06	11	06	H 00 L 01	H 00 L 00	H 00 L 03	

(Slave → Master : 応答フレーム)									
Comment	Hex	Protocol Identifier	Length Field	Unit ID (Slave 地址)	Register	Register	機器台数	先頭機器	Preset data
	H 00 L 06	H 00 L 00	H 00 L 06	11	06	H 00 L 01	H 00 L 00	H 00 L 03	



(3) Preset Multiple Register : 10

Slave機器側の400002 Registerに「000A (hex)」、「0102 (hex)」の連続した二つのデータを入力する例により、「10」コマンドフレームを説明します。 (Error Code : 90H)

(Master → Slave : 要求フレーム)		データ	
Register	40003	Register	40002
1h→2w(Byte)	04	H 00 L 0A	H 01 L 02

(Slave → Master : 応答フレーム)		データ	
Quantity of Register (Word Count)	H 00 L 02	Quantity of Register (Word Count)	H 00 L 02
先頭機器	H 00 L 01	先頭機器	H 00 L 01
11h→10h	10	11h→10h	10
Unit ID (Slave回線)	11	Unit ID (Slave回線)	11
Length Field	H 00 L 0B	Length Field	H 00 L 06
Protocol Identifier	H 00 L 00	Protocol Identifier	H 00 L 00
Transaction Identifier	H 00 L 07	Transaction Identifier	H 00 L 07