MODBUS Organization
MODBUS Master Series
MODBUS-TCP Client (Master) Driver

バージョン　GX8 Design Studio　V1.0 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

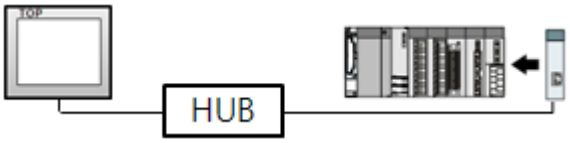
本ドライバは、MODBUS OrganizationのMODBUS ProtocolのMODBUS-TCP Client (Master)です。
外部機器(MODBUS Slave Protocolサポート)に応じて、ドライバの「コマンドコード」、「プロトコルのフレーム形式」などを別途設定する必要があります。この場合、通信方式に応じた詳細な設定を外部機器側に合わせて設定してください。
本ドライバがサポートする外部機器のシステム構成は以下の通りです。

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)

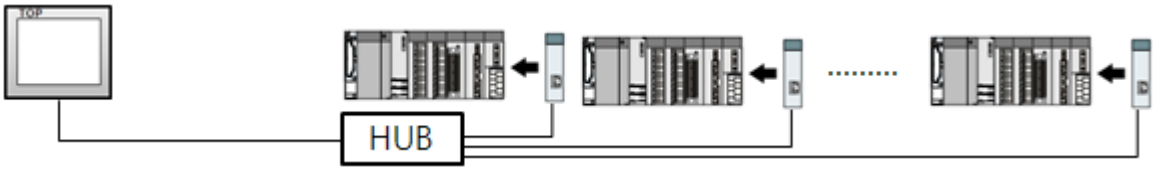
*注1) ツイストペアケーブル
- STP(シールドツイストペアケーブル)、あるいはUTP(非シールドツイストペアケーブル)カテゴリー3, 4, 5を意味します。
- ネットワーク構成に応じて、ハブ、トランシーバなどの構成機器に接続可能であり、この場合、ダイレクトケーブルを使ってください。

■接続構成

・1:1(GX8　1台と外部機器1台)接続

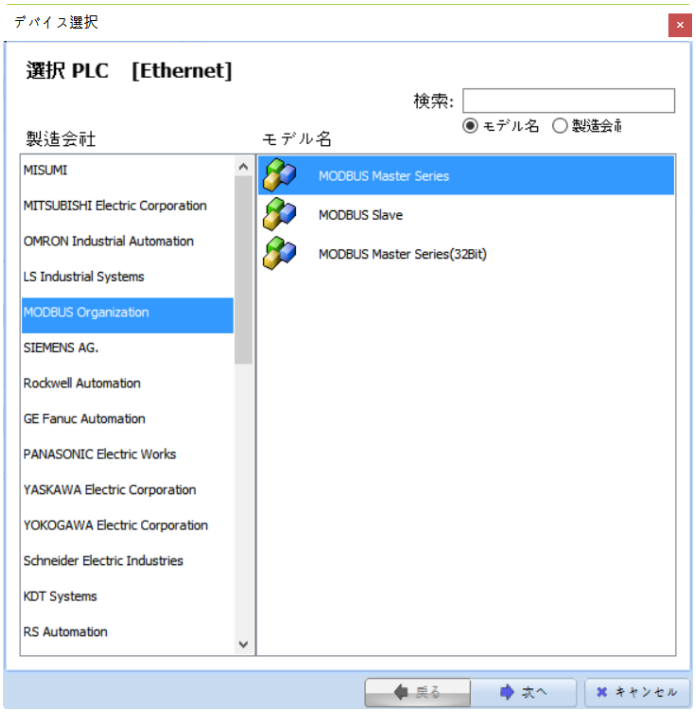


・1:N(GX8　1台と外部機器数台)接続



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を選択します。 「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。 モデル：MODBUS Master Series インタフェース：Ethernet プロトコル：MODBUS TCP 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項のシステム構成で確認してください。

3. GX8通信設定

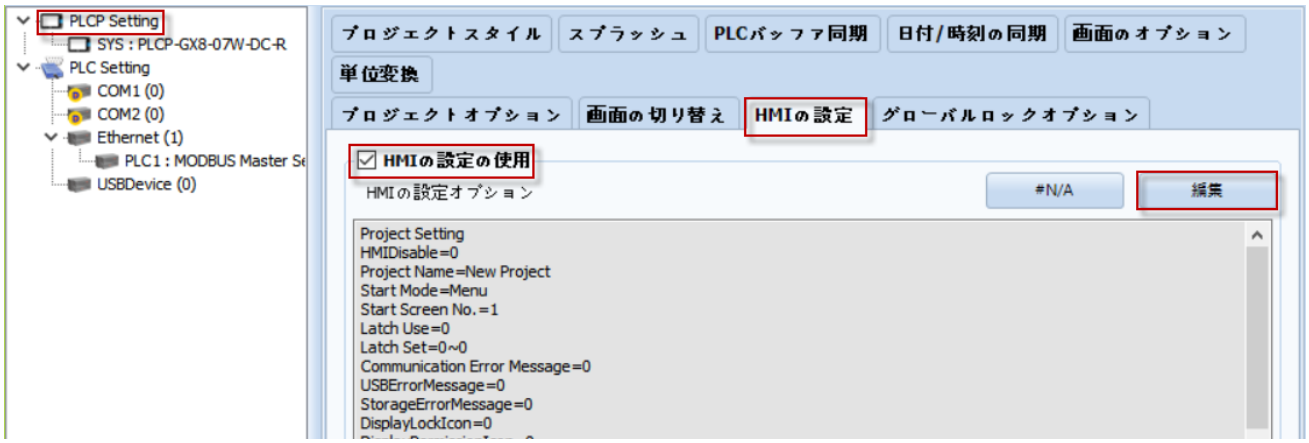
通信設定はTOP Design Studio、またはGX8 メインメニューから設定可能です。通信設定は外部機器と同一に設定する必要があります。

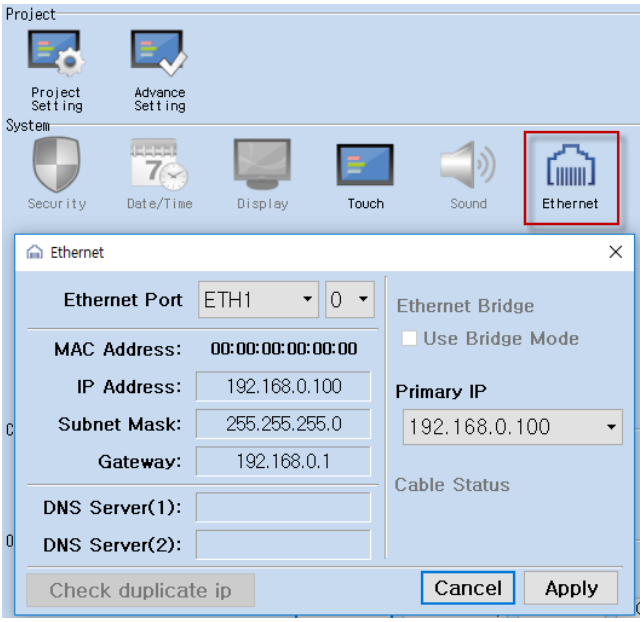
3.1 TOP Design Studioでの通信設定

(1) 通信インタフェースの設定

■[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 通信インタフェースをTOP 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	ユーザー設定
サブネットマスク	255.255.255.0	255.255.255.0	ユーザー設定
ゲートウェイ	192.168.0.1	192.168.0.1	ユーザー設定

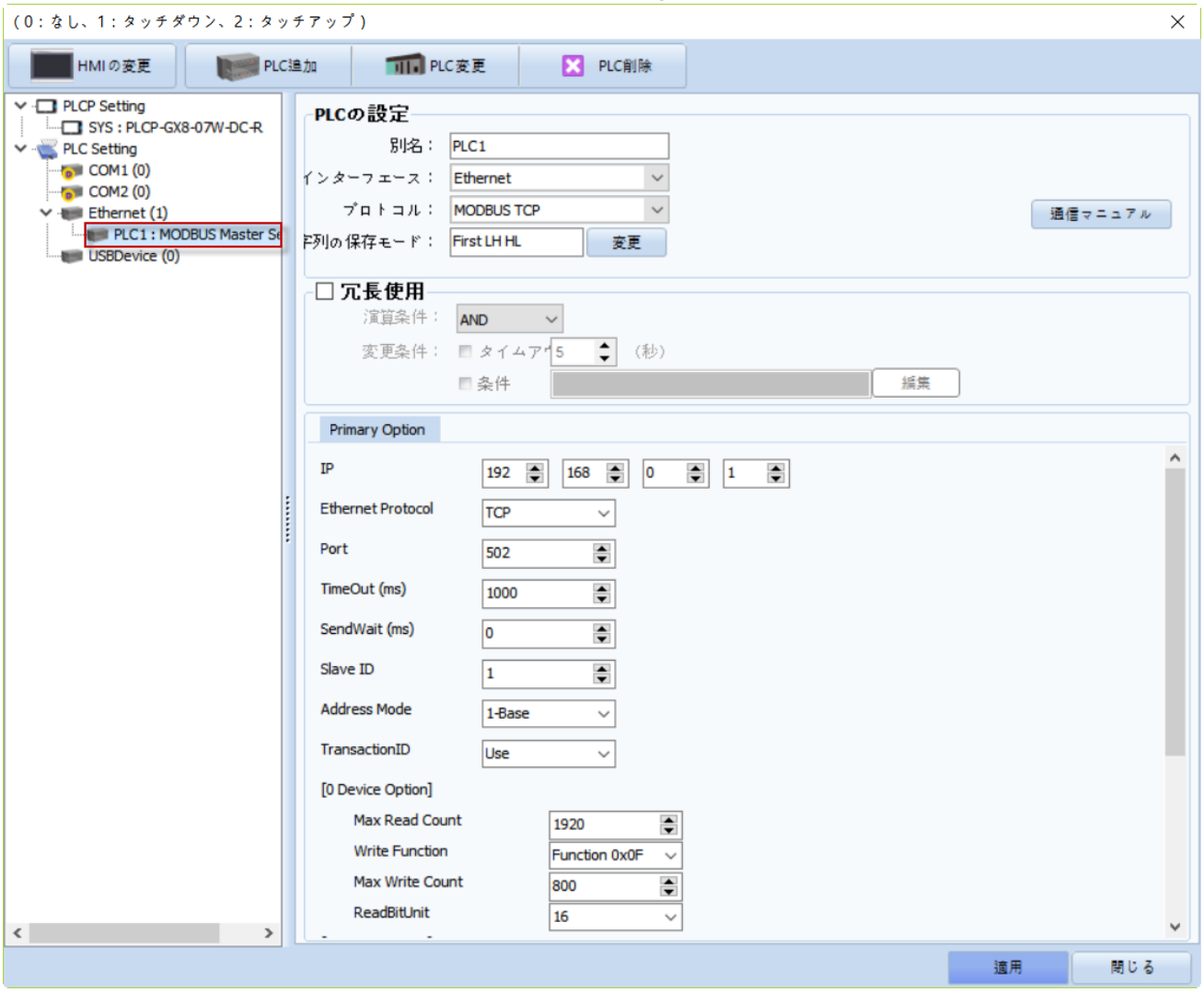
*注1) GX8 と外部機器のネットワークアドレス(IPの前三桁192.168.0.0)は一致する必要があります。
*注2) 同一ネットワーク上で、IPアドレスは重複して使用しないでください。
※上記の内容は、当社が推奨する設定例です。

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TOP Design Studioで設定します。



項目	設定	備考
インタフェース	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。	ユーザー設定
冗長　冗長使用	冗長設定を使用の有無をチェックします。	ユーザー設定
冗長　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	
冗長　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。	ユーザー設定
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。	ユーザー設定
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。	ユーザー設定
TimeOut (ms)	GX8 の外部機器からの応答を待つ時間を設定します。	ユーザー設定
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。	ユーザー設定
Slave Equipment Address No	外部機器(Slave)の局番を入力します。	ユーザー設定
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	ユーザー設定
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	ユーザー設定
[0 Device Option]	**Coil**	
Max Read Count	読み取り要求時、一度に要求できるCoilの数を設定します。	ユーザー設定 *注1）*注2）
Write Function	書き込み要求コマンドを設定します。 0x05：Force Single Coil（1ビット単位で書き込み。ビット単位の動作のみ使用可能） 0x0F：Force Multiple Coils（16ビット単位で書き込み） 外部機器のマニュアルを参照し、サポートするコマンドに応じて設定してください。	ユーザー設定 *注3）
Max Write Count	書き込み要求時、一度に要求できるCoilの数を設定します。	ユーザー設定 *注2）
Read Bit Unit	読み取り要求時、登録されたアドレスよりいくつのCoil（ビット単位）を要求するかを設定します。 「16」以外の値は「Max Read Count」の設定を無視し、「Read Bit Unit」の設定値ほど要求します。	ユーザー設定
[1 Device Option]	**Discrete Input**	
Max Read Count	読み取り要求時、一度に要求できるDiscrete Inputの数を設定します。	ユーザー設定 *注1）*注2）
Read Bit Unit	読み取り要求時、登録されたアドレスよりいくつのDiscrete Input（ビット単位）を要求するかを設定します。 「16」以外の値は「Max Read Count」の設定を無視し、「Read Bit Unit」の設定値ほど要求します。	ユーザー設定
[3 Device Option]	**Input Register**	
Max Read Count	読み取り要求時、一度に要求できるInput Registerの数を設定します。	ユーザー設定 *注1）*注2）
[4 Device Option]	**Holding Register**	
Max Read Count	読み取り要求時、一度に要求できるHolding Registerの数を設定します。	ユーザー設定 *注1）
Write Function	書き込み要求コマンドを設定します。 0x06：Preset Single Register（1個の書き込み） 0x10：Preset Multiple Registers（n個の書き込み））	ユーザー設定 *注3）
Max Write Count	書き込みコマンド0x10要求時、一度に要求できるHolding Registerの数を設定します。	ユーザー設定 *注2）

*注 1)

－ 各機器のMax Read Countは、画面に登録されたアドレスが連続していない場合、通信を複数回行わず、1回に要求するアドレスの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機器のMax Read Countを120に設定した場合、400001から400120までの連続したアドレスとして仮定し、400001から120ワードにして一回の要求でデータを読み込みます。

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機器のMax Read Countを3に設定する場合、400001から400003までの3ワード、400004から400005までの2ワード、400120を1ワードとし、3回の要求でデータを読み込みます。

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機器のMax Read Countを10に設定した場合、400001から400010までの10ワード、400011を1ワード、400021を1ワード400031を1ワード、400041を1ワードとし、5回の要求でデータを読み込みます。

－ Max Read Count を 0 に設定した場合、連続したアドレスに限って、最大 120 ワード分を要求します。

*注 2)

－ 外部機器のマニュアルを参照し、登録されたアドレスより一度にいくつのデータが読み取り/書き込みが可能かを確認してください。
外部機器がサポートする範囲よりも大きく設定した場合、通信が正常に行われません。
例）外部機器の Holding Register（4 機器）が、1 回の通信に最大 10 個のワードが応答可能な場合、GX8 の通信設定のうち 4 機器の Max Read Count を外部機器の仕様に合わせて 10 に設定してください。

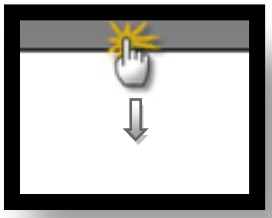
*注 3)

－ 外部機器のマニュアルを参照し、サポートする書き込みコマンドに応じて設定してください。
サポートしていない書き込みコマンドを設定すると、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 TOP 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]

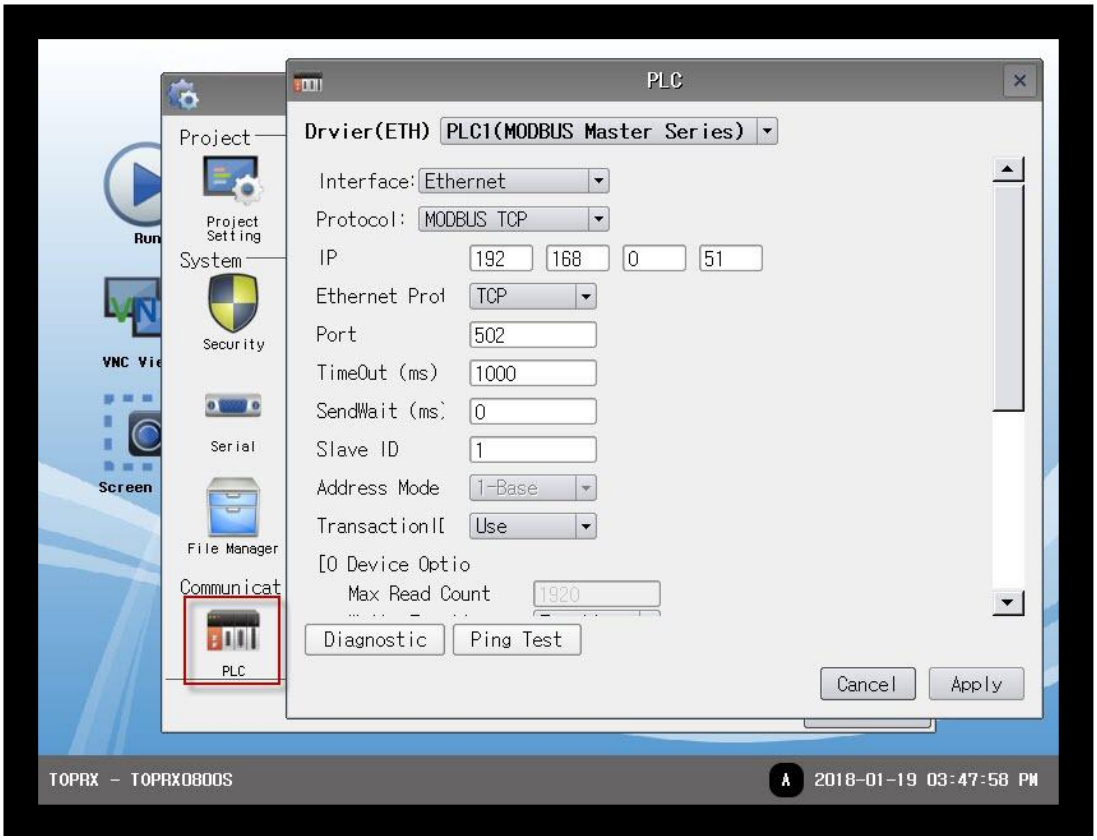
項目	GX8	外部機器	備考
IPアドレス*注1)注2)	192.168.0.100	192.168.0.51	ユーザー設定
サブネットマスク	255.255.255.0	255.255.255.0	ユーザー設定
ゲートウェイ	192.168.0.1	192.168.0.1	ユーザー設定

*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。	ユーザー設定
冗長	冗長使用	冗長設定を使用の有無をチェックします。	ユーザー設定
	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。	ユーザー設定
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。	ユーザー設定
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。	ユーザー設定
TimeOut (ms)		GX8 の外部機器からの応答を待つ時間を設定します。	ユーザー設定
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。	ユーザー設定
Slave Equipment Address No		外部機器(Slave)の局番を入力します。	ユーザー設定
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	ユーザー設定
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	ユーザー設定
[0 Device Option]		**Coil**	
Max Read Count		読み取り要求時、一度に要求できるCoilの数を設定します。	ユーザー設定 *注1) *注2)
Write Function		書き込み要求コマンドを設定します。 0x05:Force Single Coil(1ビット単位で書き込み。ビット単位の動作のみ使用可能) 0x0F:Force Multiple Coils(16ビット単位で書き込み) 外部機器のマニュアルを参照し、サポートするコマンドに応じて設定してください。	ユーザー設定 *注3)
Max Write Count		書き込み要求時、一度に要求できるCoilの数を設定します。	ユーザー設定 *注2)
Read Bit Unit		読み取り要求時、登録されたアドレスよりいくつのCoil(ビット単位)を要求するかを設定します。 「16」以外の値は「Max Read Count」の設定を無視し、「Read Bit Unit」の設定値ほど要求します。	ユーザー設定

[1 Device Option]	Discrete Input	
Max Read Count	読み取り要求時、一度に要求できるDiscrete Inputの数を設定します。	ユーザー設定 *注1) *注2)
Read Bit Unit	読み取り要求時、登録されたアドレスよりいくつのDiscrete Input(ビット単位)を要求するかを設定します。 「16」以外の値は「Max Read Count」の設定を無視し、「Read Bit Unit」の設定値ほど要求します。	ユーザー設定
[3 Device Option]	**Input Register**	
Max Read Count	読み取り要求時、一度に要求できるInput Registerの数を設定します。	ユーザー設定 *注1) *注2)
[4 Device Option]	**Holding Register**	
Max Read Count	読み取り要求時、一度に要求できるHolding Registerの数を設定します。	ユーザー設定 *注1)
Write Function	書き込み要求コマンドを設定します。 0x06:Preset Single Register(1個の書き込み) 0x10:Preset Multiple Registers(n個の書き込み))	ユーザー設定 *注3)
Max Write Count	書き込みコマンド0x10要求時、一度に要求できるHolding Registerの数を設定します。	ユーザー設定 *注2)

*注1)

- 各機器のMax Read Countは、画面に登録されたアドレスが連続していない場合、通信を複数回行わず、1回に要求するアドレスの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機器のMax Read Countを120に設定した場合、400001から400120までの連続したアドレスとして仮定し、400001から120ワードにして一回の要求でデータを読み込みます。

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機器のMax Read Countを3に設定する場合、400001から400003までの3ワード、400004から400005までの2ワード、400120を1ワードとし、3回の要求でデータを読み込みます。

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機器のMax Read Countを10に設定した場合、400001から400010までの10ワード、400011を1ワード、400021を1ワード400031を1ワード、400041を1ワードとし、5回の要求でデータを読み込みます。

- Max Read Count を0に設定した場合、連続したアドレスに限って、最大120ワード分を要求します。

*注2)

- 外部機器のマニュアルを参照し、登録されたアドレスより一度にいくつのデータが読み取り/書き込みが可能かを確認してください。

外部機器がサポートする範囲よりも大きく設定した場合、通信が正常に行われません。

例)外部機器の Holding Register(4機器)が、1回の通信に最大10個のワードが応答可能な場合、GX8 の通信設定のうち4機器の Max Read Count を外部機器の仕様に合わせて10に設定してください。

*注3)

- 外部機器のマニュアルを参照し、サポートする書き込みコマンドに応じて設定してください。

サポートしていない書き込みコマンドを設定すると、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	1. システム構成
	接続ケーブルの名称		OK	NG	
GX8	バージョン情報		OK	NG	2. 外部機器選択 3. 通信設定
	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	NG	
		通信診断	OK	NG	
	イーサネットポート設定	IPアドレス	OK	NG	
		サブネットマスク	OK	NG	
		ゲートウェイ	OK	NG	
外部機器	CPUの名称		OK	NG	4. 外部機器設定
	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	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	
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（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?

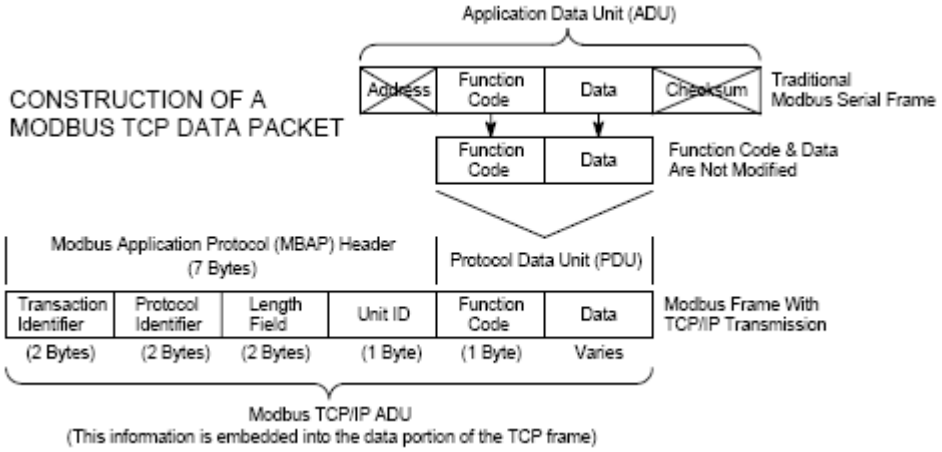
MODBUS TCP/IP (also MODBUS-TCP) is simply the MODBUS RTU protocol with a TCP interface that runs on Ethernet.
The MODBUS messaging structure is the application protocol that defines the rules for organizing and interpreting the data independent of the data transmission medium.

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 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	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭機器		機器点数	
	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 **0**00173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	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭機器		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	先頭機器		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	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭機器		機器点数	
	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	データ数 (byte)	データ(Inputs)		
	H	L	H	L	H	L				10204~10197	10212~10205	10218~10213
Hex	00	03	00	00	00	06	11	02	03	AC	DB	35

■ 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

(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	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭機器		機器点数 (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	データ数(byte)	データ 30009 Register	
	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番）の「**4**00108 – **4**00110 Register」データを読み取る例におり、「03」コマンドフレームを説明します。

（Master → Slave：要求フレーム）

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭機器		機器点数	
	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	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭機器		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	先頭機器		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)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave局番)	コマンド	先頭機器		Quantity of Register (Word Count)		データ数(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	先頭機器		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