



Modbus TCP/IP Host Protocol

Table of Contents

1. Intent & Scope 1

2. Overview 1

3. Message Register Blocks 2

4. Command messages from PLC to DXI 3

5. Status messages from DXI to PLC 3

 5.1 Peer to peer mode 3

 5.2 Polled mode 4

6. Queue Register Blocks 5

7. Future Version Compatibility 5

8. Establishing a Connection 6

9. Synchronizing the PLC and DXI 6

 9.1 Command Buffering 6

10. Error Conditions 7

11. Configuration 7

 11.1 IP Address 7

 11.2 Host Port 7

 11.3 Master Station 9

12. Redundant DXI SAC computers 10

1. Intent & Scope

This document describes the interface from the DXI System to a PLC or HMI (Human-Machine Interface) Host via Modbus TCP/IP

Please note that low level communication details of the underlying protocol are beyond the scope of this document. This document is mainly concerned with the communication between the DXI and a PLC-based or HMI-based Host. Further references to a PLC-based Host apply equally to an HMI-based host.

2. Overview

The MicroComm DXI quickly and reliably exchanges command and status messages with external computer systems, which it refers to as “host computers”. This can be done with a variety of protocols, which are selected by the user in the configuration of the DXI system.

The Modbus TCP/IP driver allows the MicroComm DXI to act like another PLC on the network. It can read and write registers (and input and output points) on the real PLC(s), and the real PLC(s) can read and write virtual registers on the MicroComm DXI system. Since the DXI system looks like another PLC on the network even though it is not a real PLC, it is sometimes referred to as a “pseudo-PLC”.

The MicroComm DXI and the external PLC will exchange command and status information by reading and writing blocks of registers. These command and status messages are generally associated directly with a

particular master station in the DXI system. Therefore, the DXI uses a separate set of register blocks for communications with each master station. This results in increased system performance by allowing multiple commands to be transmitted simultaneously to different master stations through different register blocks.

For each master station controlled by the host, the user must allocate two blocks of five registers for that master to communicate with the PLC, plus one additional register for “handshaking”. One block of registers, called the “output registers”, carry status messages from the DXI system to the PLC. The DXI will write to these registers and the PLC will read them. The other block of registers, called the “input registers”, carry command messages from the PLC to the DXI system. The PLC will write these registers and the DXI will read them. Note, in this document the terms “input” and “output” are with respect to the DXI system.

The PLC sends all commands for a particular DXI master station through its input registers. These registers are always located on the DXI “pseudo-PLC”.

Two different methods can be used to send status messages from the DXI system to the PLC. The DXI configuration files determine which method will be used.

In the first method, called “peer to peer” mode, the DXI sends status messages to the PLC by writing directly to output registers contained within the PLC.

In this method, called “peer to peer” mode, the DXI sends status messages to the PLC by writing directly to output registers contained within the PLC.

In the other method, called “polled” mode, the output registers are located on the DXI “pseudo-PLC”, and the PLC or HMI must read them periodically, or poll, to determine if there is a new message from the DXI system.

When polled mode is used, the DXI never writes into the PLC. Instead, the PLC reads status messages from virtual output registers on the DXI “pseudo-PLC”, and writes commands to virtual input registers in the DXI “pseudo-PLC”.

To ensure that both the DXI and the PLC receive every message sent by the other, they must coordinate the message transfers. The handshake register and careful sequencing are used to ensure that the sending party does not over write any message before the receiving party reads it.

The DXI also supports an optional top of queue update feature. If this feature is used, the DXI will maintain a block of queue registers that show the top N entries on the alarm queue for each master station. In peer-to-peer mode these registers are maintained on the PLC.

3. Message Register Blocks

All command and status messages in the five-register blocks start with a code word followed by up to 4 parameter words. All codes and parameters are sent as binary words two bytes, or 16-bits, in size. As a result, the code and parameters are limited to the range 0-65535. The value 0 is used as a null message code to indicate that there is no valid message in the register block. The input and output register blocks are written with the message code word in the first (lowest addressed) register. The first parameter is in the second register, the second in the third register, etc. The mapping is:

| | | |
|----------|---|-------------------|
| Code | → | Base Register |
| Param #1 | → | Base Register + 1 |
| Param #2 | → | Base Register + 2 |
| Param #3 | → | Base Register + 3 |
| Param #4 | → | Base Register + 4 |

The DXI will process the command message in the input registers when the code register is written. Similarly, it may update the output registers with a new status message when the code register is read. Therefore, if the block of 5 registers can not be read or written in one atomic block transfer operation (i.e. if they are read or written one register at a time), then the registers should be accessed in reverse order (i.e. with the parameter registers being read or written first, and the code register being read or written last). Where possible, block register transfers should be used for better performance.

4. Command messages from PLC to DXI

The PLC sends command message to the DXI by writing to the input registers.

When the PLC is writing to the DXI input registers using block transfers, there is no need for handshaking since the DXI will queue each received message for processing and there is no possibility of overwriting the previous register contents.

However, when single register transfers are used the PLC must ensure that the parameter values are written before the code register since the DXI will process the message when the code register is written. If the code register were not written last, the DXI would process a command using the new command code and the parameters from the previous message. In many cases this could result in errors, in others it could simply result in unexpected call connections etc. Furthermore, **if the code register is not written to, the DXI will not process the command.** Hence, even if PLC is sending the same command with different parameters it must re-write the code register. If it does not, the command will not be processed and will appear to be “lost”.

5. Status messages from DXI to PLC

5.1 Peer to peer mode

The DXI will always write messages to the output registers, located on the PLC, using a block register transfer to update all 5 registers at the same time.

In peer-to-peer mode the DXI will write a message to the output registers and then wait for the PLC to read those registers before writing another message. Since the output registers are in the PLC in peer-to-peer mode, the DXI can not tell when the PLC has read them. Therefore, the PLC must write to the handshake register to acknowledge that it has read the current message from the output registers and that it is ready for the DXI to write the next message. The PLC can write any value into the handshake register, it is the write operation that is important, not the value written.

If a message is not acknowledged by writing the handshake register, the DXI system will not send further status messages to this master until a timeout period has expired. This will significantly slow the apparent response time of the system.

Note the handshake register is volatile. This means that the PLC will not read back the value previously written into it. The value returned when reading the handshake register indicates whether the DXI has placed a new message into the output registers.

Typically, the PLC software will look for a non-zero value in the code register to detect a new message. It will then process the message. When it is done it will clear the registers, or at least the code register, to zero to make it easy to detect the next message. Finally it will write to the handshake register to indicate it is ready for a new message. On startup, the PLC should initialize the output registers to zero so it can reliably detect the first message.

5.2 Polled mode

If the DXI configuration is set up to use polled mode; the PLC must poll the DXI output registers for each master to determine if new status information is available. If new status information is available, it must transfer the output register block to local registers within the PLC for processing.

In polled mode the DXI will update the output registers, located in the DXI, and put a non-zero value in the handshake register, to send a message. It cannot send another message until the PLC has read the output code register.

The handshake register should be used when the DXI is operating in polled mode and the PLC is using single register transfers. In this mode, without using the handshake register, the PLC would have to read all 5 output registers in reverse order to check for a new message. This would require five network transfers per master station per polling period. It can't just read the output code register to check for a new message because that read operation could cause the DXI to update the output registers, and change the parameter values. If the PLC were to read this way, it would end up reading the code for one message and the parameters for the next.

Instead the PLC should read the handshake register to check for a new message. Whenever the DXI updates the output registers it will cause the handshake register to return a non-zero value when read. Thus reading the handshake register allows the PLC to check for new messages with a single network operation. If a non-zero value is returned, the PLC must read the message from the output registers in reverse order (or at least read the code register last) as before.

When the output code register is read, the DXI will update the output and handshake registers. If it has no more messages to send, the output registers will be cleared to zero and the handshake register will return a zero until there is a new message. If there is another message to send, the message will be placed in the output registers and the handshake register will be updated with a non-zero value.

If the PLC is using block register transfers it can poll for new messages by simply reading the entire output register block periodically. Each of these reads will also read the output code register and thereby indicate to the DXI that it can update the output registers. Therefore, the PLC must detect and process each message before polling for another new message to prevent messages from being lost.

The handshake register is not needed in polled mode if the PLC is using block register transfers. However, it can be used to increase the performance of the communications link. This can be accomplished by assigning all the master station's handshake registers at sequential register addresses, and then using a single block read of all the handshake registers to check for new messages on all the master stations. Then for each master station with a new message, the PLC will need to do another block read of the output registers to read the new message.

By using a block of handshake registers the PLC does a single network operation to poll all the masters plus a single network operation for each new message, rather than a network operation per master station to both check for and read the new message at the polling rate. Since the PLC must poll the DXI quite often, on the order of every quarter second, this can lead to a substantial reduction in network traffic. For example, a system with 30 master stations polled 4 times per second where a new message occurs every 2 seconds requires 120 network reads per second without using the handshake registers and only 19 network reads per second using a block read of the handshake registers.

In the case where the PLC is unable to poll the DXI output registers for any length of time, status messages which have not been read for longer than a preset period of time (thirty seconds) will be logged and then trashed.

6. Queue Register Blocks

The DXI can maintain a block of queue registers that show the top N entries on the alarm queue for each master station. This feature is optional and will only be activated if the queue registers are configured for a particular master station.

Each entry in this queue requires 2 registers, the first (lowest addressed) register in each pair will hold an alarm type code, and the second will hold the ID number of the device that generated the alarm. The length of the queue is controlled by the DXI configuration. Since each queue entry requires 2 registers the total number of registers in the queue register block will be 2 times the configured queue length.

The alarm codes are documented in the MicroComm DXI host port command and status message specification. The alarm code value of zero is used to indicate an unused or empty entry.

In peer-to-peer mode the queue registers are maintained on the PLC. The DXI can update these registers at any time, so it is important that they are used for display purposes only, and that they are redisplayed periodically to reflect any changes the DXI may have made to the registers.

In polled mode the queue registers are maintained on the DXI "pseudo-PLC". The PLC must poll the queue registers periodically and then use its local copy to update its queue display. The PLC can read these registers at any time.

7. Future Version Compatibility

All the command and status message codes are defined in the MicroComm DXI Host port command and status message specifications. However, the MicroComm DXI system may be enhanced by adding new command codes at any point in the future.

Therefore, it is important that the PLC not write any command codes other than those it is using. A command code that is ignored by the DXI today because it has an invalid command code may do something completely unexpected in a future version of the DXI software.

Similarly, the PLC must accept and ignore any status message sent by the DXI that it does not understand. When a message is ignored it must still be acknowledged so that the DXI can continue sending other status messages.

8. Establishing a Connection

The DXI will act as a server and wait for the client PLC to connect to it. Since the DXI is acting as a server it does not need to be configured with the IP address and port number of the PLC(s) it will be talking to. The DXI will obtain this information from the client when it connects to the DXI.

The PLC client however does need to know the IP address and port number of the DXI host port it will be talking to. This information can be obtained from the DXI configuration files.

The PLC can connect to the DXI by reading or writing any register on the DXI. When the DXI receives this message from the network it will extract the address and port number of the PLC. The DXI will then respond to PLC at that address.

In peer-to-peer mode the DXI will not write to the output registers until the PLC has connected to the DXI as described above. Therefore the PLC should write to the handshake register to open the connection. This will give the DXI the appropriate addressing information and tell the DXI that the PLC is ready for it to write to the output registers.

In polled mode the PLC should read the handshake register first. This will give the DXI the appropriate addressing information and let the PLC know if the DXI has a message ready for it to read from the output registers.

9. Synchronizing the PLC and DXI

If the PLC connects to the DXI after it has been running for some time, its status information will not be in sync with the DXI's. For example, the DXI may have call requests queued on master stations, or calls in progress that the PLC does not know about, and is not indicating on its displays. To obtain this information the PLC should send a status request command for each master station (see STAT in the message specification). This special command causes the DXI to send all the messages the PLC will need to synchronize its status with the DXI. The messages that are sent are exactly the same as if the call requests had just been queued, or the calls had just been made so that no additional processing, other than that required for normal operation, needs to be done (or programmed).

9.1 Command Buffering

The PLC should keep a one command buffer that is overwritten if new commands are generated by the operators faster than the PLC can send the commands to the DXI.

10. Error Conditions

If the PLC sends an invalid register read or write command to the DXI; that is, if it tries to access a register that is not defined in the DXI configuration, or attempts to write to a read only register, the DXI will ignore the command and respond with an error. These errors are logged by the DXI.

If the PLC issues command messages with invalid or disabled command codes the, DXI will ignore the command and respond with a failure message (see FAIL in the host port message specifications).

11. Configuration

11.1 IP Address

The host port IP address is determined by the IP address of the SAC computer which is providing the host port. This IP address is set when the system software is installed on the SAC computer.

11.2 Host Port

When configuring a card as a host port using Modbus TCP/IP protocol the following fields must be defined in the Maintenance menus, "Edit, Modify Hardware, "Card" database of the system.

The "Card ID" field is a unique identifier for this card. It will be used to direct messages from master stations to this card.

The "Exchange" field must contain the exchange ID of the SAC computer that the PLC will be connected to.

The "Type" field must be set to "MODBUS_TCPIP".

The "Master" field should be set to the ID number of the master station to which communications fault alarms generated by this host port should be sent.

| | | |
|------------------------|--------------------------------|-----------------------|
| ↓ / Enter: Next Field. | ↑: Previous field. | Home: Next Page |
| ←/→: Select | PgUp/PgDn: Next/previous entry | End: Quit |
| | F9: Delete F10: Add | F1: Help |
| Page 1 of 3 | | |
| Card ID: | 100 | |
| Name (English): | MODBUS_HOST | |
| Name (French): | ? | |
| Name (Spanish): | ? | |
| Exchange: | 1 | |
| Net A channel: | No Channel | |
| Net B channel: | No Channel | |
| Type: | MODBUS_TCP | |
| Password: | 0 | |
| Serial Port: | 1 | |
| Baud Rate: | 0 | (bits per second) |
| Data bits: | 0 | (7 or 8) |
| Stop bits: | 0 | (0 to 2) |
| Parity: | None | |
| Flow Control: | None | |
| Mother Card: | 0 | |
| Debounce: | 25 | (milliseconds) |
| Master: | 0 | (to report faults on) |
| Host Ack: | 3 | (0=Off 1=On) |
| Set Time: | 0 | (0=Off 1=On) |
| Log Level: | 5 | |

The "Host Ack" field selects the way that the DXI acknowledges host port commands. When this field is "0" the DXI will only respond to commands which fail for some reason (i.e. busy stations, invalid Ids, etc.) with a negative acknowledgement. When this field is "1" the DXI will acknowledge all commands from the PLC with either a positive or negative acknowledgement. When this field is "2" the DXI will acknowledge all commands as above, and it will also send all status messages resulting from commands from other sources in the form of positive acknowledgements. When this field is "3" the DXI will acknowledge all commands, and send all status messages resulting from commands without the positive acknowledgement prefix. See DONE, FAIL, and BUSY in the host port message specifications for additional information on the acknowledgment messages. If you are writing a new driver, you can set this to "3" to make writing the driver easier.

The following fields are on page 2 of the configuration.

The “DXI node” and “PLC node” fields are unused, but can be set to the Modbus node numbers for the DXI and PLC systems respectively.

The “PLC base reg” field is used to indicate the base address of the Modbus 4X holding registers (typically 40001 or 400001)

The “PLC routing path” field is unused.

The “IP In Port” field should contain the TCP port number that the DXI will monitor for connections from the PLC. This is normally set to 502 for most Modbus drivers' default settings.

The “IP Out Port” field is used for peer-to-peer mode only. This is normally set to 502 for most Modbus drivers' default settings.

The “Host Mode” setting of 0 (Peer) or 2 (Cimplicity) will cause the DXI to be in the “peer to peer” mode. The Cimplicity setting will also cause the DXI system to use the protocol variation of Modbus TCP/IP for Cimplicity/HMI. If the “Host Mode” setting 1 (Poll) is selected, the DXI system will use the “polled” mode for communicating with the PLC.

The “En/Dis host commands” field will bring up a list of the possible messages that the DXI can send to the PLC. Each of these messages can be enabled or disabled. When a message is disabled, the DXI will not send that message to, or accept that message from, the PLC.

After creating the host card, you will need to go to the Maintenance set of menus and use the “Card Swap” function to put it into service before you can use it.

```

↓ / Enter: Next Field.          ↑: Previous field.          Home: Next Page
←/→: Select                    PgUp/PgDn : Next/previous entry  End: Quit
F9: Delete                      F10: Add                       F1: Help

Card ID:          100          Page 2 of 3
Memory address:  0           (Hex: for some PC slot cards)
DXI node:        1           (for PLC host cards)
PLC node:        2           (for PLC host cards)
PLC base reg:    40001       (for PLC host cards)
PLC routing path:          (for PLC host cards)
IP In Port:      502        (typically 10004)
IP Out Port:     502        (typically 110XX)
Send NOOP every: 0         (seconds)
Inactivity Timeout: 0      (seconds)
2ndary ACB card ID: 0      (for redundant ACB cards)
Next ACB in loop: 0        (for ACB cards in CEPT loop)
CEPT loop number: 1      (for ACB cards in CEPT loop)
RNS card:        0         (for redundant card cages)
En/Dis host cmds:          (0=Peer 1=Poll 2=Cimplicity)
Host mode:       1
    
```

11.3 Master Station

The following fields must be defined in the Administration menus, “Edit, Modify Software, “Master” database of the system.

The “External host” field is used to specify which host port is to be used for communications to and from this master station, and how it will operate. It will be “None”, when the master does not use a host port. It will be “Parallel” or “Exclusive” when using a host port. In parallel mode, both the host port and other DXI command sources can actively control the master station at the same time (i.e. you can start a call with a host port command

| ↓ / Enter: Next Field. ←/→: Select | | ↑: Previous field. PgUp/PgDn : Next/previous entry | Home: Next Page End: Quit F1: Help |
|---------------------------------------|------------|---|--|
| | | Page 1 of 4 | |
| Master ID: | 1 | | |
| Name (English): | Master # 1 | | |
| Name (French): | ? | | |
| Name (Spanish): | ? | | |
| Password: | 1 | | |
| Secondary ID: | 0 | | |
| Enabled on startup: | 1 | (1=Enabled 0=Disabled) | |
| Call Priority: | 1 | | |
| External host: | Parallel | | |
| Ext host card: | 200 | | |
| Idle timeout: | 60 | (seconds) | |
| Keypress timeout: | 0 | (seconds) | |
| Login retries: | 10 | | |
| Login timeout: | 0 | (seconds) | |
| Unman timeout: | 0 | (seconds) | |
| Call announce: | 1 | | |
| Monitor announce: | 0 | | |
| Default announce: | 0 | | |
| Event remind rate: | 0 | (seconds) | |
| MCB menu type: | 1 | (0=Small 1=Full) | |
| #Alarm width: | 3 | | |

and end it by pressing a key on a DXI master station’s keyboard). In exclusive mode, the host port is the only active source of commands for the master station as long as the host port communications link is active. In exclusive mode the DXI will only accept master station keyboard commands after a host port communications link fault. It is recommended to set this to “Parallel” for most purposes.

The “Ext host card” field is used to specify which host port is to be used for communications to and from this master station. It contains the card ID number of a DXI card configured as a Modbus TCP/IP host.

The following fields appear on page 2 of the software configuration.

For all PLC register fields, the register number is the full Modbus holding register number (including the leading 4). For example, if you are using register numbers in the 4XXXX range the first Modbus holding register is 40001.

The “PLC Out register” field is used to specify the base address of a block of 5 output registers for the master station. This address will be in the DXI system if using polled mode, or in the PLC if using peer-to-peer or Cimplicity peer-to-peer mode.

| ↓ / Enter: Next Field. ←/→: Select | | ↑: Previous field. PgUp/PgDn: Next/previous entry | Home: Next Page End: Quit F1: Help |
|---------------------------------------|-------|--|--|
| | | Page 2 of 4 | |
| Master ID: | 1 | | |
| Annunciate alarms: | 0 | (0=AtSecondary 1=AtPrimary) | |
| External buzzer: | 0 | (0=NONE 1=Relay1 2=Relay2) | |
| Ext call Q light: | 0 | (0=NONE 1=Relay1 2=Relay2) | |
| Station call list: | 1-100 | | |
| Music zone list: | | | |
| Master enable list: | 2 | | |
| Card list: | | | |
| Auxiliary fns: | | | |
| Event priorities: | 0 | | |
| Monitor ID: | 0 | | |
| Monitor rate: | 10 | (seconds) | |
| Stn Monitor list: | 0 | | |
| PLC Out Register: | 40101 | (decimal) | |
| PLC In register: | 40001 | (decimal) | |
| PLC Queue reg: | 0 | (decimal) | |
| PLC Queue length: | 10 | (decimal) | |
| Radio Monitor list: | | | |
| Panel LED in Call: | | | |
| Can enable self: | 1 | (1=Yes 0=No) | |
| Can disable self: | 1 | (1=Yes 0=NO) | |

The “PLC In register” field is used to specify the base address of a block of 5 input registers for the master station. This address will be in the DXI.

The following fields are optional, and are only needed if the top of queue display feature is being used. If they are not used they should be left at zero.

The “PLC Queue reg” field is used to specify the base address of a block of queue registers for the master station. This address will be in DXI system if using polled mode, or in the PLC if using peer-to-peer or Cimplicity peer-to-peer mode.

The “PLC Queue length” field is used to specify the length of the queue register block for the master station. Each entry on the queue requires 2 registers so a value of 10 here indicates 5 queue entries will be used.

The following fields appear on page 3 of the software configuration.

The “PLC Hshake register” field is used to specify the address of a master station’s handshake register in the DXI “pseudo-PLC”.

The “PLC Out file num”, “PLC In file num”, “PLC Queue file num”, and “PLC HS file num” fields are unused.

```

↓ / Enter: Next Field.           ↑: Previous field.           Home: Next Page
← / →: Select                   PgUp/PgDn: Next/previous entry End: Quit
                                     F1: Help
                                     Page 3 of 4
Master ID:           1
Initial call vol:   2
Initial music vol:  4
Initial page vol:   5
Initial monitor vol: 0
Initial buzzer vol: 9
Initial sensitivity: 3
Initial backlight:  4
Caller ID list:     0           (0=off 1=on)
Page call list:    1-3
Conf call list:
Signal call list:
ALA set list:
Master call list:  1-3
Radio call list:
Visiting call list:
PLC Out file num:           (decimal)
PLC In file num:            (decimal)
PLC Queue file num:        (decimal)
Call Req timeout:          (decimal)
PLC Hshake register:       (decimal)
PLC HS file num:           (decimal)
    
```

12. Redundant DXI SAC computers

When connecting to a SAC computer, you can test whether that SAC computer is the active one by sending an **ActS** command to any master on that SAC computer. If the SAC responds with **ActS 1**, then it is active, if it responds with **ActS 0**, it is inactive.

In a redundant system, you should try to connect to each computer and send this **ActS** command.

Once you have determined which SAC computer is active, you should send all commands to the active computer only (except for **NOOP** which you can send to the inactive computer as well). All commands sent to an inactive computer will be **IGNORED** except for an **Actv** command which will tell this computer to become the active computer.

The inactive computer will not send any status response messages except for a periodic **NOOP** command (if configured to do so). The active computer, of course, will send all of the status response messages for this exchange (and also sends **NOOP** commands if configured).