Industrial communication solutions for Windows

XSQUARED Driver Manual

Square-D SY/MAX Point-to-Point Communications Protocol Driver

Contents

XSQUARED technical specifications

General information

XSQUARED driver utilizes the SY/MAX Protocol for Point-to-Point Data Transfer Communications. SY/MAX Protocol is used in all SY/MAX programmable controllers and SY/MAX devices that support serial communications. SY/MAX is based on RS-485.

This driver provides the following commands for transferring data from device to device through standard serial communications or via the SY/NET Local Area Network.

- Read Consecutive Registers as Unsigned Integers
- Read Consecutive Registers as Signed Integers
- Read Consecutive Registers as Bits
- Write Consecutive Registers as Unsigned Integers
- Write Consecutive Registers as Signed Integers
- Write Register Individual Bit

Important note:

If you are using a RS-232 to RS-422/485 converter, it might be necessary to set RTS while transmitting and receiving.

Communication settings must be:

- BaudRate = Same than meter
- Parity = Even
- DataBits = 8
- StopBits = 1

<About Registers> Please refer to your device's manual to find information about the meaning of the different registers in your device's memory. Because this information may vary from one device to other while SY/MAX remains the same, it makes no sense to write down in this help all the possible register meanings for all the devices that support SY/MAX protocol. Besides, this information is included in the users manual supplied with all Square-D devices. For PowerLogic in particular, this information is located in Appendix-E (Abbreviated Register Listing) of the PowerLogic Circuit Monitor Operation Manual.

<About Routes Configuration> This section explains how users should configure HMITalk1.DriverP6 to access personalized routing information.

If you don't need to supply any special information for the ID field and ROUTE, you should leave the HMITalk1.DriverP6 property empty.

When HMITalk1.DriverP6 is empty, the following internal values are assumed:
ID = 18 (12 hex) ROUTE = 3 (03 hex) + 103 (67 hex) + station number The syntax to change these values is:

Meaning of the DriverP6 parameter:

ID,Drop1,Drop2 (Drop3 is always the station address previously defined in the HMITalk1.DriverP0 parameter) For example: Suppose that you want an ID of 17 (11 hex), Drop1 = 4 (04 hex), and Drop2 = 11 (0B hex). You must enter: HMITalk1.DriverP6 = 17,4,11 (if you omit a parameter, its default value is assumed).

Command list

Read Consecutive Registers as Unsigned Integers

Description of this command:

Reads HMITalk1.DriverNumPoints registers starting at a specified register number, returning register values as 2-byte unsigned integers. Returned values can range from 0 to 65535.

Methods used to run this command:

Analog Input

Number of points accepted by this command:

1-128

Meaning of the DriverP0 parameter:

Indicates the meter station address in the RS-485 network (range is: 1-199).

Meaning of the DriverP1 parameter:
Indicates the starting register number (1-65535).
Meaning of the DriverP2 parameter:
0
Meaning of the DriverP3 parameter:
1
Meaning of the DriverP4 parameter:
Number of registers used to form each value returned (1-4). (The total number of registers cannot be more than 128).
Meaning of the DriverP6 parameter:
Indicates advanced header information (See note about routes configuration in driver overview).

Read Consecutive Registers as Signed Integers

Description of this command:
Reads HMITalk1.DriverNumPoints registers starting at a specified register number, returning register values as 2-byte signed integers. Returned values can range from -32767 to 32767.
Methods used to run this command:
Analog Input
Number of points accepted by this command:
1-128
Meaning of the DriverP0 parameter:
Indicates the meter station address in the RS-485 network (range is: 1-199).
Meaning of the DriverP1 parameter:
Indicates the starting register number (1-65535).
Meaning of the DriverP2 parameter:
0
Meaning of the DriverP3 parameter:
0
Meaning of the DriverP4 parameter:
Number of registers used to form each value returned (1-4). (The total number of registers cannot be more than 128).
Meaning of the DriverP6 parameter:
Indicates advanced header information (See note about routes configuration in driver overview).

Read Consecutive Registers as Bits

Description of this command:
Reads HMITalk1.DriverNumPoints bits starting at a specified bit of a specified register number.
Methods used to run this command:
Digital Input
Number of points accepted by this command:
1-128
Meaning of the DriverP0 parameter:
Indicates the meter station address in the RS-485 network (range is: 1-199).
Meaning of the DriverP1 parameter:
Indicates the starting register number (1-65535).
Meaning of the DriverP2 parameter:
0
Meaning of the DriverP3 parameter:
1
Meaning of the DriverP4 parameter:
Indicates the starting bit in the first register (0-15).
Meaning of the DriverP6 parameter:
Indicates advanced header information (See note about routes configuration in driver overview).

Write Consecutive Registers as Unsigned Integers

Description of this command:
Writes HMITalk1.DriverNumPoints registers starting at a specified register number, converting corresponding HMITalk1.PointValue() data values to 2-byte unsigned integers ranging from 0 to 65535. HMITalk1.PointValue() data values will be forced to remain inside the valid range. A value lower than 0 will be treated as 0 and a value greater than 65535 will be treated as 65535.

Methods used to run this command:
Analog Output
Number of points accepted by this command:
1-125
Meaning of the DriverP0 parameter:
Indicates the meter station address in the RS-485 network (range is: 1-199).
Meaning of the DriverP1 parameter:
Indicates the starting register number (1-65535).
Meaning of the DriverP2 parameter:
2
Meaning of the DriverP3 parameter:
1
Meaning of the DriverP6 parameter:
Indicates advanced header information (See note about routes configuration in driver overview).

Write Consecutive Registers as Signed Integers

Description of this command:
Writes HMITalk1.DriverNumPoints registers starting at a specified register number, converting corresponding HMITalk1.PointValue() data values to 2-byte signed integers ranging from -32767 to 32767. HMITalk1.PointValue() data values will be forced to remain inside the valid range. A value lower than -32767 will be treated as -32767 and a value greater than 32767 will be treated as 32767.
Methods used to run this command:
Analog Output
Number of points accepted by this command:
1-125
Meaning of the DriverP0 parameter:
Indicates the meter station address in the RS-485 network (range is: 1-199).
Meaning of the DriverP1 parameter:
Indicates the starting register number (1-65535).
Meaning of the DriverP2 parameter:
2
Meaning of the DriverP3 parameter:
0
Meaning of the DriverP6 parameter:
Indicates advanced header information (See note about routes configuration in driver overview).

Write Register Individual Bit

Description of this command:
Writes a single bit at a specified register number.
Methods used to run this command:
Digital Output
Number of points accepted by this command:
1
Meaning of the DriverP0 parameter:
Indicates the meter station address in the RS-485 network (range is: 1-199).
Meaning of the DriverP1 parameter:
Indicates the register number (1-65535).
Meaning of the DriverP2 parameter:
2
Meaning of the DriverP3 parameter:
1
Meaning of the DriverP4 parameter:
Indicates the bit in the register (0-15).
Meaning of the DriverP6 parameter:
Indicates advanced header information (See note about routes configuration in driver overview).

Error messages

The following list shows the possible error messages that can be returned by the driver during a failed communication in the 'Status' property.

[1005] DRIVER (Internal): Invalid driver stage
[1300] PROTOCOL (Timeout): No answer
[1425] PROTOCOL (Format): Response is too short
[1433] PROTOCOL (Format): Validation error in device response
[2147] CONFIG (NumValues): Only one value can be read or written
[2203] CONFIG (NumValues): Too many values (max=250)
[2294] CONFIG (NumValues): Too many registers (max=128)
[3011] CONFIG (P0): Invalid device address (0-199)
[3575] CONFIG (P1): Invalid starting register (1-65535)
[4046] CONFIG (P2): Invalid command (must be 0)
[4047] CONFIG (P2): Invalid command (must be 2)
[4534] CONFIG (P3): Invalid data type (0=signed/1=unsigned)
[4535] CONFIG (P3): Invalid data type (must be 1)
[5010] CONFIG (P4): Invalid first bit (Range:0-15)
[5041] CONFIG (P4): Invalid number of registers (1-4)
[6015] CONFIG (P6): Invalid drop1 (range is 1-255)
[6016] CONFIG (P6): Invalid drop2 (range is 1-255)
[6022] CONFIG (P6): Invalid id (range is 1-255)
[8200] CONFIG (Remote): Invalid station address
[8352] CONFIG (Remote): Unknown message response
[8402] CONFIG (Remote): Invalid opcode in response

Supported devices

This driver can communicate with these devices, but is not necessarily limited to this list:

SQUARE-D PowerLogic CM-2050 Circuit Monitor
SQUARE-D PowerLogic CM-2150 Circuit Monitor
SQUARE-D PowerLogic CM-2250 Circuit Monitor
SQUARE-D PowerLogic CM-2350 Circuit Monitor
SQUARE-D PowerLogic CM-2450 Circuit Monitor
SQUARE-D SY/MAX Devices