XTWIDOS Driver Manual

Schneider Electric Telemecanique Twido PLC Serial Driver

CPKSoft Engineering

Process Monitoring and Industrial Automation Software

Copyright 1990-2009, CPKSoft Engineering. All rights reserved.

Index

1. Introduction

CPKSoft Engineering assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

This driver is included with all unlimited licenses of TAS-HMITalk. It is not sold separately. It requires the TAS-HMITalk ActiveX to work, therefore it cannot be used as a stand-alone driver.

If you use this driver in your applications, you need to include the xtwidos.tlk in the set of files that you distribute. This file must be located in the same folder where the hmitalk.ocx file is registered in order to be found by the activex when the applications are executed.

The source-code for the xtwidos.tlk driver is available in plain-C language for additional USD 399 if you own a license of TAS-HMITalk 8.04 or higher (this price is subject to change).

Refer to the following link to visit the xtwidos driver page at CPKSoft Engineering website: http://www.cpksoft.com/tabid/55/ProductID/120/PageIndex/1/Default.aspx.

Visit this link if you want to see a complete list of drivers that are currently available for TAS-HMITak: http://www.cpksoft.com/Drivers/tabid/55/Default.aspx.

Also, refer to this link if you are interested in purchasing a license of the most recent version of TAS-HMITalk: http://www.cpksoft.com/Products/tabid/54/Default.aspx.

We welcome your comments about this document. You can reach us by e-mail at contact @ cpksoft.com.

2. Driver details

2.1. Driver overview

XTWIDOS driver allows you to connect to Schneider Electric TWIDO PLCs using the Modbus RTU protocol through the TWIDO serial port.

This driver supports RS485 networking to connect multiple slave devices to a single computer running the TAS-HMITalk application.

This driver supports to communicate to the PLC serial port in an ethernet-to-serial configuration and can connect to a serial server such as Moxa or Exemys directly, without needing to install a COM port redirector or creating a virtual COM port.

The Twido port must be configured for Modbus. If using Port 1 of the Twido, the DPT signal must be tied to signal ground.

REGISTER TYPES SUPPORTED:

- %MW
- %M

SERIAL COMMUNICATION SETTINGS:

- Serial port I/F: RS232, RS485
- Data Bits: 7 or 8 (Must match the pc's port setting)
- Stop Bits: 1 or 2 (Must match the pc's port setting)
- Baud Rate: 9600,19200,38400,57600,115200 (Must match the pc's port setting)
- Parity: Even, Odd, None (Must match the pc's port setting)
- PLC station No.: 0-255 (Working as a slave device)
- PLC time out: 1500 to 5000 ms (adjust if longer timeout is required)

2.2. Supported devices

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

SCHNEIDER ELECTRIC TELEMECANIQUE TWIDO PLC

3. Command list

3.1. Read multiple %M status

Description of this command:

Obtains the current status (ON=1/OFF=0) in a group of consecutive %M. Use DriverNumPoints to specify the number of consecutive %M to be read. Use DriverP0 to specify the PLC station number in the RS485 network.

Type of data handled by this command:

Digital Input

Number of points accepted by this command:

1-1000

Meaning of the DriverP0 parameter:

Station number (0-255)

Meaning of the DriverP1 parameter:

Must be set to 1

Meaning of the DriverP2 parameter:

Indicates the first %M address to be read (%M0=0)

Values that are returned:

Value in PointValue (0) = First %M status (0=OFF, 1=ON)
Value in PointValue (1) = Second %M status (0=OFF, 1=ON)
...
Value in PointValue (n-1) = Last %M status (0=OFF, 1=ON)

3.2. Write single %M status

Description of this command:

Writes a single %M. Use DriverP0 to specify the PLC station number in the RS485 network.

Type of data handled by this command:

Digital Output

Number of points accepted by this command:

1

Meaning of the DriverP0 parameter:

Station Number (0-255). If the station number is 0, the command is sent as a broadcast message and no response is expected.

Meaning of the DriverP1 parameter:

Must be set to 5

Meaning of the DriverP2 parameter:

Indicates the %M address to be written (%M0=0).

Values that are sent:

Value in PointValue (0) = New %M status (0=OFF, 1=ON)

3.3. Write multiple %M status

Description of this command:

Writes a series of consecutive %M. Use DriverNumPoints to specify the number of %M to be written. Use DriverP0 to specify the PLC station number in the RS485 network.

Type of data handled by this command:

Digital Output

Number of points accepted by this command:

1-1000 (It is convenient that this number is a multiple of 8)

Meaning of the DriverP0 parameter:

Station Number (0-255). If the station number is 0, the command is sent as a broadcast message and no response is expected.

Meaning of the DriverP1 parameter:

Must be set to 15

Meaning of the DriverP2 parameter:

Indicates the first %M address to be written (%M0=0).

Values that are sent:

Value in PointValue (0) = New status for first %M (0=OFF, 1=ON)
Value in PointValue (1) = New status for second %M (0=OFF, 1=ON)
...
Value in PointValue (n-1) = New status for last %M (0=OFF, 1=ON)

3.4. Read multiple %MW as unsigned words

Description of this command:

Obtains the current values in a group of consecutive registers of type %MW, returning the values as unsigned words ranging from 0 to 65535. Use DriverNumPoints to specify the

number of %MW registers to be read. Use DriverP0 to specify the PLC station number in the RS485 network.

Type of data handled by this command:

Analog Input

Number of points accepted by this command:

1-125

Meaning of the DriverP0 parameter:

Station number (0-255)

Meaning of the DriverP1 parameter:

Must be set to 3

Meaning of the DriverP2 parameter:

Indicates the first %MW register address to be read (%MW0=0).

Values that are returned:

Value in PointValue (0) = First %MW register value (0 to 65535)
Value in PointValue (1) = Second %MW register value (0 to 65535)
...
Value in PointValue (n-1) = Last %MW register value (0 to 65535)

3.5. Write multiple %MW as unsigned words

Description of this command:

Writes values to a group of consecutive registers of type %MW, treating the values as unsigned words ranging from 0 to 65535. Use DriverNumPoints to specify the number of %MW registers to be written. Use DriverP0 to specify the PLC station number in the RS485 network.

Type of data handled by this command:

Analog Output

Number of points accepted by this command:

1-125

Meaning of the DriverP0 parameter:

Station number (0-255) If the station number is 0, the command is sent as a broadcast message and no response is expected.

Meaning of the DriverP1 parameter:

Must be set to 16

Meaning of the DriverP2 parameter:

Indicates the first %MW register address to be written (%MW0=0)

- Use 40001 for %MW0

- Use 40002 for %MW1

...

- Use 40256 for %MW255

Values that are returned:

Value in PointValue (0) = First %MW register value (0 to 65535)
Value in PointValue (1) = Second %MW register value (0 to 65535)

...

Value in PointValue (n-1) = Last %MW register value (0 to 65535)

3.6. Read multiple %MW as signed integers

Description of this command:

Obtains the current values in a group of consecutive registers of type %MW, returning the values as unsigned words ranging from -32768 to 32767. Use DriverNumPoints to specify the number of %MW registers to be read. Use DriverP0 to specify the PLC station number in the RS485 network.

Type of data handled by this command:

Analog Input

Number of points accepted by this command:

1-125

Meaning of the DriverP0 parameter:

Station number (0-255)

Meaning of the DriverP1 parameter:

Must be set to 74

Meaning of the DriverP2 parameter:

Indicates the first %MW register address to be read (%MW0=0)

Values that are returned:

Value in PointValue (0) = First %MW register value (-32768 to 32767)
Value in PointValue (1) = Second %MW register value (-32768 to 32767)

...

Value in PointValue (n-1) = Last %MW register value (-32768 to 32767)

3.7. Write multiple %MW as signed integers

Description of this command:

Writes values to a group of consecutive registers of type %MW, treating the values as unsigned words ranging from -32768 to 32767. Use DriverNumPoints to specify the number of %MW registers to be written. Use DriverP0 to specify the PLC station number in the RS485 network.

Type of data handled by this command:

Analog Output

Number of points accepted by this command:

1-125

Meaning of the DriverP0 parameter:

Station number (0-255) If the station number is 0, the command is sent as a broadcast message and no response is expected.

Meaning of the DriverP1 parameter:

Must be set to 78

Meaning of the DriverP2 parameter:

Indicates the first %MW register address to be written (%MW0=0)

Values that are returned:

Value in PointValue (0) = First %MW register value (-32768 to 32767)
Value in PointValue (1) = Second %MW register value (-32768 to 32767)
...
Value in PointValue (n-1) = Last %MW register value (-32768 to 3276

4. Appendices

4.1. Error messages

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

This list does not include some error messages that can be returned by the activex component while attempting to establish a connection.

[1005] DRIVER (Internal): Invalid driver stage
[1300] PROTOCOL (Timeout): No answer

4.2. Keywords list

The following list shows a set of words directly related to this driver.

"ELECTRIC, PLC, SCHNEIDER, Serial, TELEMECANIQUE, TWIDO".