

XMBENRON Driver Manual

Modbus Enron ASCII Protocol Driver



CPKSoft Engineering Process Monitoring and Industrial Automation Software

Copyright 1990-2008, CPKSoft Engineering. All rights reserved.

Index

1.	Introduction	3
2.	Driver details	4
2.1.	Driver overview	4
2.2.	Supported devices.....	4
3.	Command list	5
3.1.	Read Unit Status	5
3.2.	Read Boolean Variables	5
3.3.	Read Short Integer Variables.....	6
3.4.	Read Long Integer Variables	6
3.5.	Read Floating Point Variables.....	7
3.6.	Read Manual/Live Status of Short Integer Variables	7
3.7.	Read Manual/Live Status of Long Integer Variables	8
3.8.	Read Manual/Live Status of Floating Point Variables	8
3.9.	Set Single Boolean Variable.....	9
3.10.	Set Single Short Integer Variable	9
3.11.	Set Single Long Integer Variable	10
3.12.	Set Single Floating Point Variable	10
3.13.	Set Single Manual/Live Status of Short Integer Variable.....	11
3.14.	Set Single Manual/Live Status of Long Integer Variable.....	11
3.15.	Set Single Manual/Live Status of Floating Point Variable	12
3.16.	Set Multiple Boolean Variables.....	12
3.17.	Set Multiple Short Integer Variables	13
3.18.	Set Multiple Long Integer Variables	13
3.19.	Set Multiple Floating Point Variables	14
3.20.	Set Multiple Manual/Live Status of Short Integer Variables.....	14
3.21.	Set Multiple Manual/Live Status of Long Integer Variables.....	15
3.22.	Set Multiple Manual/Live Status of Floating Point Variables.....	15
4.	Appendices	16
4.1.	Error messages	16
4.2.	Keywords list.....	16

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 xmbenron.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 xmbenron.tlk driver is available in plain-C language for additional USD 199 if you own a license of TAS-HMITalk 8.04 or higher.

Refer to the following link to visit the xmbenron driver page at CPKSoft Engineering website: <http://www.cpksoft.com/tabid/55/ProductID/64/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](mailto:contact@cpksoft.com).

2. Driver details

2.1. Driver overview

XMBENRON driver was designed to communicate with any device that use the modified Gould Modicon MODBUS ENRON ASCII Protocol. Enron Modbus is just normal Modbus with special handling of 32-bit values as ONE register instead of TWO. Enron or Enron/Daniels Modbus is Standard Modbus with a few vendor extensions.

2.2. Supported devices

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

BRISTOL BABCOCK Computer Using Modbus Enron
DANIEL INDUSTRIES Enhanced Solar Flow Plus 2470
DANIEL INDUSTRIES Flow Computer 2500
DANIEL INDUSTRIES Flow Computer Ultra 3000
DANIEL INDUSTRIES Chromatograph Danalyzer

3. Command list

3.1. Read Unit Status

Description of this command:

This command allows the interrogation of eight Boolean unit status variables within the device. The state of the eight status variables is to be maintained by the device until they are read by the master computer. After the variables are read, if set, they are to be reset to zero.

Type of data handled by this command:

Digital Input

Number of points accepted by this command:

1-8

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

7

Values that are sent:

Value in PointValue (0) = Shutdown
Value in PointValue (1) = Unknown failure
Value in PointValue (2) = Power failure
Value in PointValue (3) = Unacknowledged alarms
Value in PointValue (4) = Starting
Value in PointValue (5) = Running
Value in PointValue (6) = Warm start
Value in PointValue (7) = Cold start

3.2. Read Boolean Variables

Description of this command:

This command allows the user to obtain the ON/OFF status of logic Booleans used to control discrete outputs from the addressed device. Up to 250 boolean variables can be obtained at each request; however, the device may have restrictions which reduce the maximum quantity.

Type of data handled by this command:

Digital Input

Number of points accepted by this command:

1-250

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

1

Meaning of the DriverP2 parameter:

Indicates the initial Boolean address to be read (starting at 1001 and up to 3000).

3.3. Read Short Integer Variables

Description of this command:

This command allows the user to obtain the contents of numeric variables of type short integer (16-bits) in the addressed device. The addressing allows up to 125 variables to be obtained at each request.

Type of data handled by this command:

Analog Input

Number of points accepted by this command:

1-124

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

3

Meaning of the DriverP2 parameter:

Indicates the initial short-integer address to be read (starting at 3001 and up to 5000).

3.4. Read Long Integer Variables

Description of this command:

This command allows the user to obtain the contents of numeric variables of type long integer (32-bits) in the addressed device. The addressing allows up to 64 variables to be obtained at each request.

Type of data handled by this command:

Analog Input

Number of points accepted by this command:

1-62

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

3

Meaning of the DriverP2 parameter:

Indicates the initial long-integer address to be read (starting at 5001 and up to 7000).

3.5. Read Floating Point Variables

Description of this command:

This command allows the user to obtain the contents of numeric variables of type floating point (32-bits) in the addressed device. The addressing allows up to 64 variables to be obtained at each request.

Type of data handled by this command:

Analog Input

Number of points accepted by this command:

1-62

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

3

Meaning of the DriverP2 parameter:

Indicates the initial floating point address to be read (starting at 7001 and up to 60000).

3.6. Read Manual/Live Status of Short Integer Variables

Description of this command:

This command allows the user to obtain the manual/live status of variables of type short integer (16-bits) in the addressed device. The addressing allows up to 125 variables to be obtained at each request.

Type of data handled by this command:

Digital Input

Number of points accepted by this command:

1-250

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

1

Meaning of the DriverP2 parameter:

Indicates the initial short-integer address whose manual/live status is to be read (starting at 3001 and up to 5000).

3.7. Read Manual/Live Status of Long Integer Variables

Description of this command:

This command allows the user to obtain the contents of numeric variables of type long integer (32-bits) in the addressed device. The addressing allows up to 64 variables to be obtained at each request.

Type of data handled by this command:

Digital Input

Number of points accepted by this command:

1-250

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

1

Meaning of the DriverP2 parameter:

Indicates the initial long-integer address whose manual/live status is to be read (starting at 5001 and up to 7000).

3.8. Read Manual/Live Status of Floating Point Variables

Description of this command:

This command allows the user to obtain the contents of numeric variables of type floating point (32-bits) in the addressed device. The addressing allows up to 64 variables to be obtained at each request.

Type of data handled by this command:

Digital Input

Number of points accepted by this command:

1-250

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

1

Meaning of the DriverP2 parameter:

Indicates the initial floating-point address whose manual/live status is to be read (starting at 7001 and up to 60000).

3.9. Set Single Boolean Variable

Description of this command:

This command forces a Boolean variable to either ON or OFF status. Boolean variables are numbered from 1001.

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)

Meaning of the DriverP1 parameter:

5

Meaning of the DriverP2 parameter:

Indicates the Boolean variable address to be set (1001 to 3000).

3.10. Set Single Short Integer Variable

Description of this command:

This command allows the user to change the contents of numeric variables of type short integer (16-bits) in the addressed device.

Type of data handled by this command:

Analog Output

Number of points accepted by this command:

1

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

6

Meaning of the DriverP2 parameter:

Indicates the short-integer address to be changed (3001 to 5000).

3.11. Set Single Long Integer Variable

Description of this command:

This command allows the user to change the contents of numeric variables of type long integer (32-bits) in the addressed device.

Type of data handled by this command:

Analog Output

Number of points accepted by this command:

1

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

6

Meaning of the DriverP2 parameter:

Indicates the long-integer address to be changed (5001 to 7000).

3.12. Set Single Floating Point Variable

Description of this command:

This command allows the user to change the contents of numeric variables of type floating point (32-bits) in the addressed device.

Type of data handled by this command:

Analog Output

Number of points accepted by this command:

1

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

6

Meaning of the DriverP2 parameter:

Indicates the floating point address to be changed (7001 to 60000).

3.13. Set Single Manual/Live Status of Short Integer Variable

Description of this command:

This command is used on a defined short-integer variable to switch from manual to live.

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)

Meaning of the DriverP1 parameter:

5

Meaning of the DriverP2 parameter:

Indicates the short-integer address whose manual/live status is to be set (3001 to 5000).

3.14. Set Single Manual/Live Status of Long Integer Variable

Description of this command:

This command is used on a defined long-integer variable to switch from manual to live.

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)

Meaning of the DriverP1 parameter:

5

Meaning of the DriverP2 parameter:

Indicates the long-integer address whose manual/live status is to be set (5001 to 7000).

3.15. Set Single Manual/Live Status of Floating Point Variable

Description of this command:

This command is used on a defined floating point variable to switch from manual to live.

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)

Meaning of the DriverP1 parameter:

5

Meaning of the DriverP2 parameter:

Indicates the floating-point address whose manual/live status is to be set (7001 to 60000).

3.16. Set Multiple Boolean Variables

Description of this command:

This command allows the user to set the ON/OFF status of multiple logic Booleans. Up to 250 boolean variables can be set at once; however, the device may have restrictions which reduce the maximum quantity.

Type of data handled by this command:

Digital Output

Number of points accepted by this command:

1-250

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

15

Meaning of the DriverP2 parameter:

Indicates the initial Boolean address to be set (starting at 1001 and up to 3000).

3.17. Set Multiple Short Integer Variables

Description of this command:

This command allows the user to set the contents of multiple numeric variables of type short integer (16-bits) in the addressed device. The addressing allows up to 125 variables to be set at once.

Type of data handled by this command:

Analog Output

Number of points accepted by this command:

1-124

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

16

Meaning of the DriverP2 parameter:

Indicates the initial short-integer address to be set (starting at 3001 and up to 5000).

3.18. Set Multiple Long Integer Variables

Description of this command:

This command allows the user to set the contents of multiple numeric variables of type long integer (32-bits) in the addressed device. The addressing allows up to 64 variables to be set at once.

Type of data handled by this command:

Analog Output

Number of points accepted by this command:

1-62

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

16

Meaning of the DriverP2 parameter:

Indicates the initial long-integer address to be set (starting at 5001 and up to 7000).

3.19. Set Multiple Floating Point Variables

Description of this command:

This command allows the user to set the contents of multiple numeric variables of type floating point (32-bits) in the addressed device. The addressing allows up to 64 variables to be set at once.

Type of data handled by this command:

Analog Output

Number of points accepted by this command:

1-62

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

16

Meaning of the DriverP2 parameter:

Indicates the initial floating point address to be set (starting at 7001 and up to 60000).

3.20. Set Multiple Manual/Live Status of Short Integer Variables

Description of this command:

This command allows the user to set the manual/live status of multiple variables of type short integer (16-bits) in the addressed device. The addressing allows up to 125 variables to be set at once.

Type of data handled by this command:

Digital Output

Number of points accepted by this command:

1-250

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

15

Meaning of the DriverP2 parameter:

Indicates the initial short-integer address whose manual/live status is to be set (starting at 3001 and up to 5000).

3.21. Set Multiple Manual/Live Status of Long Integer Variables

Description of this command:

This command allows the user to set the contents of multiple numeric variables of type long integer (32-bits) in the addressed device. The addressing allows up to 64 variables to be set at once.

Type of data handled by this command:

Digital Output

Number of points accepted by this command:

1-250

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

15

Meaning of the DriverP2 parameter:

Indicates the initial long-integer address whose manual/live status is to be set (starting at 5001 and up to 7000).

3.22. Set Multiple Manual/Live Status of Floating Point Variables

Description of this command:

This command allows the user to set the contents of multiple numeric variables of type floating point (32-bits) in the addressed device. The addressing allows up to 64 variables to be set at once.

Type of data handled by this command:

Digital Output

Number of points accepted by this command:

1-250

Meaning of the DriverP0 parameter:

Station Number (0-255)

Meaning of the DriverP1 parameter:

15

Meaning of the DriverP2 parameter:

Indicates the initial floating-point address whose manual/live status is to be set (starting at 7001 and up to 60000).

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
- [1408] PROTOCOL (Format): Invalid amount of data bytes received
- [1410] PROTOCOL (Format): Invalid device id in response
- [1421] PROTOCOL (Format): Negative acknowledge received from device
- [1433] PROTOCOL (Format): Validation error in device response
- [2184] CONFIG (NumValues): Too many values (max=124)
- [2229] CONFIG (NumValues): Too many values (max=62)
- [3014] CONFIG (P0): Invalid device address (0-255)
- [3022] CONFIG (P0): Invalid device address (1-255)
- [3521] CONFIG (P1): Invalid command (1 or 7)
- [3528] CONFIG (P1): Invalid command (3 only)
- [3533] CONFIG (P1): Invalid command (5 or 15 only)
- [3534] CONFIG (P1): Invalid command (6 or 16 only)
- [3545] CONFIG (P1): Invalid command or too many values
- [4010] CONFIG (P2): Invalid address (1001-60000)
- [4101] CONFIG (P2): Invalid starting address (3001 to 60000)
- [4110] CONFIG (P2): Invalid target address (1001 to 60000)
- [4111] CONFIG (P2): Invalid target address (3001 to 60000)
- [8013] CONFIG (Remote): Acknowledge
- [8034] CONFIG (Remote): Busy (rejected message)
- [8138] CONFIG (Remote): Failure in associated device
- [8168] CONFIG (Remote): Illegal data address
- [8170] CONFIG (Remote): Illegal data value
- [8172] CONFIG (Remote): Illegal function
- [8347] CONFIG (Remote): Unknown error

4.2. Keywords list

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

"2470, 2500, 3000, ASCII, BABCOCK, BRISTOL, Chromatograph, Danalyzer, DANIEL, Enhanced, Enron, Flow, INDUSTRIES, Modbus, Plus, Solar, Ultra".