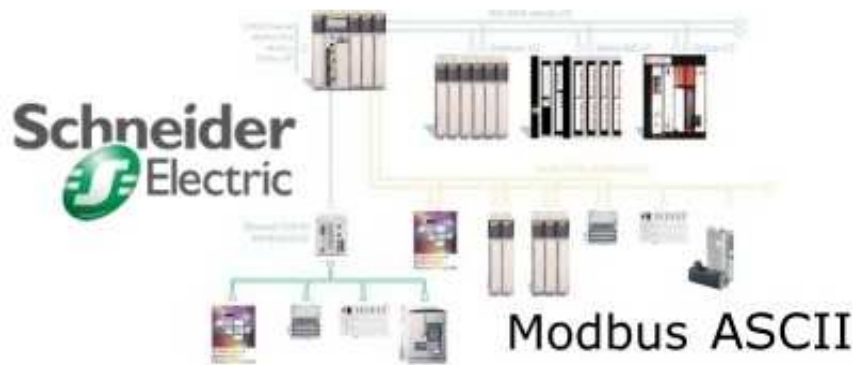


XMODBUS Driver Manual

Gould Modicon Modbus 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 Coil Status	5
3.2.	Read Input Status.....	5
3.3.	Read Holding Registers as Unsigned	6
3.4.	Read Input Registers as Unsigned	6
3.5.	Read Holding Registers as Signed.....	7
3.6.	Read Input Registers as Signed	7
3.7.	Force Single Coil.....	8
3.8.	Preset Single Register.....	8
3.9.	Force Multiple Coils.....	8
3.10.	Preset Multiple Registers.....	9
3.11.	Read Holding Registers reading 8 of each 16 bits.....	9
3.12.	Preset Multiple Registers as IEEE floats	10
3.13.	Read Holding Registers skipping 2 bytes.....	10
3.14.	Read Holding Registers as IEEE floats	11
3.15.	Read Holding Registers as IEEE floats skipping 2 bytes	11
3.16.	Read Holding Registers as inverted IEEE floats	12
3.17.	Read Holding Registers as inverted IEEE floats skipping 2 bytes	12
3.18.	Preset Single/Multiple Registers as IEEE floats.....	13
3.19.	Preset Single/Multiple Registers as IEEE floats skipping 2 bytes.....	13
3.20.	Preset Single/Multiple Registers as inverted IEEE floats.....	14
3.21.	Preset Single/Multiple Registers as inverted IEEE floats skipping 2 bytes.....	14
3.22.	Read Holding Registers as Signed 32 Bits Integer	15
3.23.	Read Holding Registers as Unsigned 32 Bits Integer.....	15
3.24.	Preset Multiple Registers as Signed 32 Bits Integer.....	16
3.25.	Preset Multiple Registers as Unsigned 32 Bits Integer.....	16
4.	Appendices	18
4.1.	Error messages	18
4.2.	Keywords list.....	18

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 xmodbusa.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 xmodbusa.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 xmodbusa driver page at CPKSoft Engineering website: <http://www.cpksoft.com/tabid/55/ProductID/68/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

XMODBUS driver allows you to connect to any equipment which uses the Gould Modicon MODBUS ASCII Protocol (Use the XMODBUS driver if you need the binary version).

2.2. Supported devices

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

- AEG MODICON A984-120 PLC
- AEG MODICON A984-130 PLC
- AEG MODICON A984-145 PLC
- ARCOM CONTROL SYSTEMS M1 Mini RTU
- CONTROL MICROSYSTEMS TeleSafe 16EX RTU
- CONTROL MICROSYSTEMS TeleSafe Micro16 RTU
- ITT BARTON 1140 Flow Meter

3. Command list

3.1. Read Coil Status

Description of this command:

Obtains current status (ON/OFF) in a group of logic coils.

Type of data handled by this command:

Digital Input

1-250. It is convenient to handle groups in multiples of 8 elements.

Meaning of the DriverP0 parameter:

Station Number (0-255). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

1

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.2. Read Input Status

Description of this command:

Obtains current status (ON/OFF) in a group of discrete inputs.

Type of data handled by this command:

Digital Input

1-250. It is convenient to handle groups in multiples of 8 elements.

Meaning of the DriverP0 parameter:

Station Number (0-255). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

2

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.3. Read Holding Registers as Unsigned

Description of this command:

Obtains current values in one or more holding registers as unsigned 2-bytes integer numbers from 0 to 65535.

Type of data handled by this command:

Analog Input / Digital Input

Number of points accepted by this command:

1-125

Meaning of the DriverP0 parameter:

Station Number (0-255). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

3

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.4. Read Input Registers as Unsigned

Description of this command:

Obtains current values in one or more input registers as unsigned 2-bytes integer numbers from 0 to 65535.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

4

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.5. Read Holding Registers as Signed

Description of this command:

Obtains current values in one or more holding registers as signed 2-bytes integers numbers from -32768 to 32767.

Type of data handled by this command:

Analog Input / Digital Input

Number of points accepted by this command:

1-125

Meaning of the DriverP0 parameter:

Station Number (0-255). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

74

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.6. Read Input Registers as Signed

Description of this command:

Obtains current values in one or more input registers as signed 2-bytes integers numbers from -32768 to 32767.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

79

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.7. Force Single Coil

Description of this command:

Forces a single logic coil to a state ON or OFF.

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 is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

5

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.8. Preset Single Register

Description of this command:

Places a specified value into a holding register as an unsigned 2-bytes integer value.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

6

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.9. Force Multiple Coils

Description of this command:

Forces a series of consecutive logic coils to selected ON or OFF states.

Type of data handled by this command:

Digital Output

1-125. It is convenient for the groups to be multiple of 8 elements.

Meaning of the DriverP0 parameter:

Station Number (0-255). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

15

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.10. Preset Multiple Registers

Description of this command:

Places specified values into a series of consecutive holding registers, as unsigned 2-bytes integer values.

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 is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

16

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.11. Read Holding Registers reading 8 of each 16 bits

Description of this command:

Obtains current values in one or more holding registers reading 8 bits out of every 16 bits, as unsigned 2-bytes integer values.

Type of data handled by this command:

Digital Input

1-250. It is convenient to handle groups in multiples of 8 elements.

Meaning of the DriverP0 parameter:

Station Number (0-255). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

-3

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.12. Preset Multiple Registers as IEEE floats

Description of this command:

Places specified values into a series of consecutive holding registers, as 4-bytes IEEE floating point values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

-16

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.13. Read Holding Registers skipping 2 bytes

Description of this command:

Obtains current values in one or more holding registers skipping 2 bytes between every data read (starts skipping 2), as unsigned 2-bytes integer values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

65

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.14. Read Holding Registers as IEEE floats

Description of this command:

Obtains current values in one or more holding registers, as 4-bytes IEEE floating point values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

66

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.15. Read Holding Registers as IEEE floats skipping 2 bytes

Description of this command:

Obtains current binary value in one or more holding registers skipping 2 bytes between every data read (starts skipping 2), as 4-bytes IEEE floating point values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

67

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.16. Read Holding Registers as inverted IEEE floats

Description of this command:

Obtains current values in one or more holding registers as 4-bytes inverted IEEE floating point values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

68

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.17. Read Holding Registers as inverted IEEE floats skipping 2 bytes

Description of this command:

Obtains current values in one or more holding registers as 4-bytes inverted IEEE floating point values and skipping 2 bytes between every data read (starts skipping 2).

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

69

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.18. Preset Single/Multiple Registers as IEEE floats

Description of this command:

Places specified values into a series of consecutive holding registers as 4-bytes IEEE floating point values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

70

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.19. Preset Single/Multiple Registers as IEEE floats skipping 2 bytes

Description of this command:

Places specified values into a series of consecutive holding registers as 4-bytes IEEE floating point values and skipping 2 bytes between every data read (starts skipping 2).

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

71

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.20. Preset Single/Multiple Registers as inverted IEEE floats

Description of this command:

Places specified values in inverse mode into a series of consecutive holding registers as 4-bytes inverted IEEE floating point values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

72

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.21. Preset Single/Multiple Registers as inverted IEEE floats skipping 2 bytes

Description of this command:

Places specified values in inverse mode into a series of consecutive holding registers as 4-bytes inverted IEEE floating point values and skipping 2 bytes between every data read (starts skipping 2).

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

73

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.22. Read Holding Registers as Signed 32 Bits Integer

Description of this command:

Obtains current values in one or more holding registers as signed 4-bytes integer numbers.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

75

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.23. Read Holding Registers as Unsigned 32 Bits Integer

Description of this command:

Obtains current values in one or more holding registers as unsigned 4-bytes integer numbers.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

76

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.24. Preset Multiple Registers as Signed 32 Bits Integer

Description of this command:

Places specified values into a series of consecutive holding registers, as signed 4-bytes integer values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

77

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

3.25. Preset Multiple Registers as Unsigned 32 Bits Integer

Description of this command:

Places specified values into a series of consecutive holding registers, as unsigned 4-bytes integer values.

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). If the station is 0 this implies a broadcasting message, only valid to send outputs.

Meaning of the DriverP1 parameter:

78

Meaning of the DriverP2 parameter:

Indicates the memory address of the selected element.

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
- [2147] CONFIG (NumValues): Only one value can be read or written
- [2185] CONFIG (NumValues): Too many values (max=125)
- [2229] CONFIG (NumValues): Too many values (max=62)
- [2274] CONFIG (NumValues): Too many values to write (max=600)
- [3014] CONFIG (P0): Invalid device address (0-255)
- [3022] CONFIG (P0): Invalid device address (1-255)
- [3508] CONFIG (P1): Invalid command
- [4001] CONFIG (P2): Invalid address
- [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.

"1140, 16EX, A984-120, A984-130, A984-145, AEG, ARCOM, ASCII, BARTON, CONTROL, Flow, Gould, ITT, Meter, Micro16, MICROSYSTEMS, Mini, Modbus, MODICON, PLC, RTU, SYSTEMS, TeleSafe".