

48x9A Addendum

B.1 INTRODUCTION

New Modbus commands have been added to the 48x9 series GPIB-to-Modbus Interfaces. This addendum describes these commands for users with 48x9 Instruction Manuals.

B.2 NEW PARAGRAPH 1.5.1

1.5.1 Modbus RTU Messages

Messages conform to the Modbus RTU format and include the device address, command, register number, data and CRC formatted as binary bytes. Supported Modbus commands are: 02, 03, 04, 05, 06, 07, 08, and 16 for integer values and commands 03 and 16 for floating point 32-bit values.

Integer range 16 bits or 0 to 65,536

Floating point IEEE-754

B.3 EXPANDED MODBUS COMMAND TABLE

The next three pages contain the 48x9A Modbus Command Table with the new commands.

B

.7 MODBUS COMMANDS

The following commands are used to Control Modbus slave devices. These Modbus Commands should not be mixed or concatenated with IEEE-488.2 or SCPI commands. Commands marked with an asterisk are new in the 48x9A series interfaces and are not available in the older 48x9 interfaces.

TABLE 3-4 MODBUS COMMANDS

Syntax	Meaning
C addr	Modbus Address Command. Sets Modbus slave device address for subsequent commands. Value for <i>addr</i> is 1 to 255. Default setting is 1.
RC[?] reg, ncoil *	Read Coil Status Command (code 0x02). Reads the status of coils in a remote device. User specifies starting coil address in register <i>reg</i> and number of coils to be read <i>ncoil</i> . The [?] is an optional symbol for smart programs. ⁽⁴⁾ Values for <i>reg</i> are 0 to 65535. Values for <i>ncoil</i> are 1 to 2000. Responses are returned as a packed binary value with 1 bit per coil. 1 = ON.
RI[?] reg, ninp *	Read Discrete Inputs Command (code 0x02). Reads discrete inputs. User specifies starting address in register <i>reg</i> and number of inputs to be read <i>ninp</i> . The [?] is an optional symbol for smart programs. ⁽⁴⁾ Values for <i>reg</i> are 0 to 65535. Values for <i>ninp</i> are 1 to 2000. Responses are returned as a packed binary value with 8 inputs per byte. 1 = ON.
R[?] reg, num	Read Register Command (code 0x03). Reads one or multiple Modbus device registers. User specifies starting register <i>reg</i> and number of registers to be read <i>num</i> . The [?] is an optional symbol for smart programs. ⁽⁴⁾ Values for <i>reg</i> are 0 to 65535. Values for <i>num</i> are 1 to 64. Responses are returned as 16-bit decimal or HEX values separated by commas. Output format selected with the Format command. i.e. R? 0,1 reads Watlow Model Number. Response is 5270 for Watlow Model F4 R? 0,3 reads three successive registers. Response is 5270,0,123 for the Watlow F4 Controller.

B

TABLE 3-4 MODBUS COMMANDS CONT'D

Syntax	Meaning
RR[?] reg, num *	Read Input Register Command (code 0x04). Reads one or multiple Modbus device registers. User specifies starting register <i>reg</i> and number of registers to be read <i>num</i> . The [?] is an optional symbol for smart programs. ⁽⁴⁾ Values for <i>reg</i> are 0 to 65535. Values for <i>num</i> are 1 to 64. Responses are returned as 16-bit decimal or HEX values separated by commas. Output format selected with the Format command. See the R? query above.
RE[?] *	Read Exception Status Query (code 0x07). Reads eight exception status outputs from a remote device. The [?] is an optional symbol for smart programs. ⁽⁴⁾ Responses are returned as a packed binary value with the eight status bits in one byte.
RF? reg	Read Floating Point Value Command (code 0x03). Reads two sequential registers as an IEEE-754 32-bit floating point value in low byte to high byte order. The specified register, <i>reg</i> , contains the lower two bytes and the next higher register contains the upper two bytes.
WC reg, b *	Write Coil Command* (code 0x05). Writes a ON/OFF value, <i>b</i> to a single Modbus device register, <i>reg</i> . Values for <i>reg</i> are 0 to 65535. Values for <i>b</i> are 0/OFF or 1/ON. An example is: WC 1000, ON
W reg, w	Write Register Command (code 0x06). Writes a 16-bit value, <i>w</i> to a single Modbus device register, <i>reg</i> . Values for <i>reg</i> are 0 to 65535. Values for <i>w</i> are 0 to 65535. An example is: W 100, 55 writes the decimal 55 to register 100.
WB reg, num, w(0),w(n)	Write Block Command (code 0x10). Writes multiple 16-bit words, <i>w(i)</i> to multiple registers or 32-bit values to two adjacent registers. Starting register, <i>reg</i> . Number, <i>num</i> specifies how many words are to be written. Values for <i>reg</i> are 0 to 65535. Values for <i>nu</i> are 1 to 64. Values for <i>w</i> are 0 to 65535. <i>w(i)</i> values are separated by commas.



TABLE 3-4 MODBUS COMMANDS CONT'D

Syntax	Meaning														
WF? reg, num	Write Floating Point Value Command (0x16). Writes an IEEE 754 single precision 32-bit value to two registers in low word to high word order. <i>reg</i> specifies the low word. <i>reg +1</i> is the high word. <i>num</i> is determined by the parameter being controlled and can range from 2^{127} to 2^{-127}														
L[?] w	Loopback Command (code 0x08). Writes a 16-bit word, <i>w</i> , out to a Modbus device and returns a single response word to the GPIB bus. The question mark is optional for smart programs. ⁽⁴⁾ Value for <i>w</i> is 0 to 65535.														
D time	Timeout Command. Sets timeout value of Modbus response message in milliseconds. Timeout is the total time for the message to be received by the 48x9A. Value for <i>time</i> is 1 to 65,535 milliseconds. The default value is 100.														
D?	Queries the current timeout setting.														
E?	<p>Read Error Command. Reads and clears the Modbus Error Register and bit 6 in the Event Status Register. Returns a error code whose value is 0 to 255. Current error values are:</p> <table data-bbox="441 938 989 1198"> <tr> <td>0</td> <td>No errors present</td> </tr> <tr> <td>1</td> <td>Exception Code 1</td> </tr> <tr> <td>2</td> <td>Exception Code 2</td> </tr> <tr> <td>3</td> <td>Exception Code 3</td> </tr> <tr> <td>100</td> <td>CRC Error</td> </tr> <tr> <td>101</td> <td>Timeout Error indicates no characters received in the response message.</td> </tr> <tr> <td>2nn</td> <td>Partial or corrupted message received. where nn is the number of received bytes.</td> </tr> </table>	0	No errors present	1	Exception Code 1	2	Exception Code 2	3	Exception Code 3	100	CRC Error	101	Timeout Error indicates no characters received in the response message.	2nn	Partial or corrupted message received. where nn is the number of received bytes.
0	No errors present														
1	Exception Code 1														
2	Exception Code 2														
3	Exception Code 3														
100	CRC Error														
101	Timeout Error indicates no characters received in the response message.														
2nn	Partial or corrupted message received. where nn is the number of received bytes.														

Notes:

1. All values are in decimal. To enter HEX values, the value must be preceded with a #h . i.e. 100 decimal = #h64. Integer range is 0 to 65535.
2. Response parameter format set by SCPI FORMat command. Default is ASCii
3. Do not combine the Modbus commands in Table 3-4 with IEEE-488.2 commands or SCPI commands to avoid query errors or otherwise confusing the GPIB<->Modbus Interfaces.
4. The [?] is an optional symbol for smart programs like ICS's GPIBKybd program. These programs can recognize the command as a query and automatically read the response.

B.3 NEW PROGRAMMING EXAMPLES

The following paragraphs were added to Section 3.8

3.8.8 Floating Point Variables

Some new Modbus devices like Watlow's EZ PM series controllers use two consecutive register to control a value or to read back a process variable. The two registers hold an IEEE-754 32-bit floating point word. The registers are read and written to in the low word-upper word order.

3.8.9 Floating Point Write

The WF command writes the num value in floating point format to two consecutive registers starting with the low word register.

WF 2160, 75 'writes to registers 2160 and 2161

3.8.10 Floating Point Read

The RF? query reads a 32-bit floating point value from two sequential register in low word-upper word order. The RF? does not require the number of register to read since it is fixed at two registers.

RF? 360 'reads registers 360 and 361

