Sample EZ-ZONE® PM 32-bit Modbus RTU Packet

Sent to PM - Read (32-bit) Analog Input 1 value

Binary	Hex	Decimal	Purpose
00000001	0x01	1	Controller Address
00000011	0x03	3	Function Code - Read Holding Registers
00000001	0x01	1	Read Starting at register High Byte (Analog Input 1 value is contained in registers 360 & 361)
11010000	0x68	104	Read Starting at register Low Byte (Analog Input 1 value is contained in registers 360 & 361)
00000000	0x00	0	Read number of consecutive registers - High Byte (Always 0)
00000010	0x02	2	Read number of consecutive registers - Low Byte (1 to 125)
00101011	0x2B	43	Low byte of CRC
01000100	0x44	68	High byte of CRC

The CRC (also a 16 bit wide value) is sent in reverse order, low byte then high byte.

Received from PM - Read Analog Input 1 value of 78.295 °F (32-bit)

Binary	Hex	Decimal	Purpose
00000001	0x01	1	Controller Address
00000011	0x03	3	Function Code - Read Holding Registers
00000100	0x04	4	Number of data bytes returned
10010111	0x97	151	Data High Byte of 1st register Read - MSB of LSW
01111101	0x7D	125	Data Low Byte of 1st register Read - LSB of LSW consecutive registers
01000010	0x42	66	Data High Byte of 2nd register Read - MSB of MSW
10011100	0x9C	156	Data Low Byte of 2nd register Read - LSB of MSW consecutive registers
01110110	0x76	118	Low byte of CRC
10010110	0x96	150	High byte of CRC

Description of example above:

Analog Input 1 value is contained in two 16-bit registers. Register 360 contains the two lower bytes (least significant word, LSW) while register 361 contains the two higher bytes (most significant word, MSW). Register 360 is 0x0168. The 32-bit answer is an IEEE 754, 32-bit float data type.

The packet described is assembled and sent to the PM as one continuous stream of bits per the Modbus standard. The packet returned from the PM is decoded per the Modbus standard.

In this example, we extract 0x977D 0x429C from the packet for the answer. Changing the Modbus Word Order to High Word, Low Word we see the answer is 0x429C 0x977D.

Converting the raw data 0X429C977D to a 32-bit floating point value shows the analog input is reading +78.295 degrees.

1241 Bundy Blvd
Winona, MN 55987
Telephone (507) 494-5656

General steps to read registers are:
Assemble a packet to send the controller:

1. Determine controller address to read. Example: Address 0x01
2. Determine function code for read. Example: Function Code (0x03) – Read Holding Registers or Function Code (0x04) – Read Input Registers. The PM responds to both command with the same information.
3. Determine Modbus relative registers to read (360 & 361 decimal for Analog Input 1 value using Data Map 2)
4. Convert register numbers to hexadecimal. Example: 360 decimal = 0x0168
5. Determine number of registers to read.
6. Enter 0x00 for number of registers to read high byte
7. Enter number of registers to read low byte from previous step into packet. As many as 125 registers may be read with one read command. Example: Use 0x02 registers to retrieve one 32-bit value. Use 0x04 to retrieve four consecutive registers which might contain two 32-bit values or four 16-bit values or other combinations.
8. Calculate the CRC on the packet. Use the Internet to find free programs to demonstrate how CRCs are calculated.
9. Enter the Low Byte of CRC calculation into packet
10. Enter the High Byte of CRC calculation into packet
11. Send packet as one continuous stream
12. Wait for response from controller
13. If no response or exception code, enter into error routine per standard

Process the packet received based on these steps:

14. Process packet for accuracy by comparing CRC to calculated value
15. If CRC is correct, proceed else enter into error routine per standard
16. Parse answer from packet based on number of bytes returned
17. Convert answers to appropriate data type. Some data is 32-bit floating point values while enumerated data is 16-bit unsigned integer values. In very few registers, the data type is 32-bit unsigned integer values. A column in the PM User's Guide provides the relative register address, the data type and whether the register is read only or read/write capable.

Please note that the difference between a Modbus RTU packet and a Modbus TCP packet is that Modbus TCP uses the same Modbus RTU packet without the CRC shown above and encapsulates the packet into an Ethernet packet using the Modbus TCP protocol.

MSB = Most significant byte
LSB = Least significant byte
MSW = Most significant word
LSW = Least significant word
CRC = Cyclic Redundancy Check

1241 Bundy Blvd
Winona, MN 55987
Telephone (507) 494-5656

Sample EZ-ZONE® PM 32-bit Modbus RTU Packet

Sent to PM - Write (32-bit) Set Point 1 of 75.0 °F

Binary	Hex	Decimal	Purpose
00000001	0x01	1	Controller Address
00010000	0x10	16	Function Code – Write Multiple Registers
00001010	0x0A	10	Write Starting at register High Byte (Set Point 1 is register 2640 & 2641)
01010000	0x50	80	Write Starting at register Low Byte (Set Point 1 is register 2640 & 2641)
00000000	0x00	0	Write number of consecutive registers - High Byte (Always 0)
00000010	0x02	2	Write number of consecutive registers - Low Byte (1 to 123)
00000100	0x04	4	Number of Bytes to Write (always 2 x write number)
01000010	0x42	66	Data High Byte of 1st register Write - MSB of LSW
10010110	0x96	150	Data Low Byte of 1st register Write - LSB of LSW consecutive registers
01000010	0x00	0	Data High Byte of 2nd register Write - MSB of MSW
10010110	0x00	0	Data Low Byte of 2nd register Write - LSB of MSW consecutive registers
01100111	0x67	103	Low byte of CRC
01111100	0x7C	124	High byte of CRC

The CRC (also a 16 bit wide value) is sent in reverse order, low byte then high byte.

Received from PM - Writing Set Point 1 of 75.0 °F

Binary	Hex	Decimal	Purpose
00000001	0x01	1	Controller Address
00010000	0x10	16	Function Code – Write Multiple Registers
00001010	0x0A	10	High Byte of Register 2640 decimal – Start writing at register
01010000	0x50	80	Low Byte of Register 2640 decimal – Start writing at register
00000000	0x00	0	High Byte – number of registers written
00000010	0x02	2	Low Byte – number of registers written
00000001	0x01	1	Low byte of CRC
01000010	0x42	66	High byte of CRC

Description of example above:

The Set Point 1 of the PM is contained in two 16-bit registers. Register 2640 contains the two lower bytes (least significant word, LSW) while register 2641 contains the two higher bytes (most significant word, MSW). Register 2640 is 0x0A50. The 32-bit answer is an IEEE 754, 32-bit float data type. Floating point writes must always be accomplished using a Function Code - Multiple Write Registers command.

The packet described is assembled and sent to the PM as one continuous stream of bits per the Modbus standard. The packet returned from the PM is decoded per the Modbus standard.

In this example, we write a set point of 75.0.

0x42960000 = 75.0 degrees when read/written as a 32-bit float data type
0000 4296 is in Low Word (LSW), High Word (MSW) Order.
Register 2640 is written with LSW of 0x0000
Register 2641 is written with MSW of 0x4296

1241 Bundy Blvd
Winona, MN 55987
Telephone (507) 494-5656

General steps to write registers are:
Assemble a packet to send the controller:

1. Determine controller address to write. Example: Address 0x01
2. Determine function code for write. Example: Function Code (0x10) – Write Multiple Registers or Function Code (0x06) – Write Single Register. The PM uses Write Multiple Registers for all 32-bit values.
3. Determine starting Modbus relative registers to write (2640 decimal for Set Point 1)
4. Convert register number to hexadecimal. Example: 2640 decimal = 0x0A50
5. Enter 0x00 for number of consecutive registers to write high byte
6. Enter 0x02 for number of consecutive registers to write low byte
7. Enter 0x04 for the number of bytes to write – 4 bytes is for a 32-bit value
8. Calculate the CRC on the packet.
9. Enter the Low Byte of CRC calculation into packet
10. Enter the High Byte of CRC calculation into packet
11. Send packet as one continuous stream
12. Wait for response from controller

Process the packet received based on these steps:

13. Process packet for accuracy by comparing CRC to calculated value
14. If CRC is correct, proceed else enter into error routine per standard
15. Validate response matches what was sent.
16. If response does not match, enter into error routine per standard

1241 Bundy Blvd
Winona, MN 55987
Telephone (507) 494-5656

Additional details –

Some process values may be rounded off to fit into the four-character display of the PM.

Full floating point process values are readable via Modbus. The displayed units of measurement are independent of the units of measurement sent via communications.

- **Example: The controller may be set to display in °C on the PM LED display but utilize °F in communication exchanged values. For Modbus RTU settings, see 'Setup Page', 'Communications Menu' to configure Modbus Address, Baud Rate, Parity, Display Units, Modbus Word Order and Data Map settings. We suggest using Data Map 2 for all new applications. Data Map 1 is provided for legacy applications and contains less registers to access. The Display Units in the Network settings affects the communications exchanged and the Display Units in the 'Setup Page', 'Global Menu' affect the values on the PM LED display.**

All temperature parameters exchanged via communications are in °F through Modbus by default. Modbus Word Order is Low High by default. Baud Rate is 9600 with no parity by default. See the PM User's Guide for Modbus register assignment and additional information. To prevent unintended programming changes never write values until you have read the desired register and validated it as the correct register assignment.

By default the low register number contains the two lower bytes (least significant word); high register numbers contain the two higher bytes (most significant word) of the four-bytes for 32-bit floating-point values.

- **To change the word order, set parameter Modbus Word Order 'Low High' to 'High Low' in 'Setup Page', 'Communications Menu' using the PM keypad.**

The only function codes supported in the PM are –
Function Code (0x03) – Read Input Registers
Function Code (0x04) – Read Holding Registers
Function Code (0x06) – Write Single Register
Function Code (0x10) – Write Multiple Registers

Visit http://www.modbus.org for a free download of the Modbus RTU and Modbus TCP implementation specifications.

Visit http://www.watlow.com/literature/software.cfm and locate ModbusTest for a free sample program to test communication with Modbus RTU.

Visit http://www.watlow.com/literature/software.cfm and locate ModbusTCPTest for a free sample program to test communication with Modbus TCP (Ethernet).

Visit http://www.modbusdriver.com/shop/product_info.php?products_id=66 for a third party Modbus software driver. The software may be purchased from ModbusDRIVER.com and is an excellent buy to get your software quickly talking to Modbus devices when writing a .NET application. The software driver include their technical support assistance. See the documentation for converting between relative versus absolute Modbus addressing for a given driver.

Visit https://www.wireshark.org/download.html for a free data capture program when working with Ethernet. This is an excellent program to capture data transfer between Ethernet devices for analysis.

1241 Bundy Blvd
Winona, MN 55987
Telephone (507) 494-5656