Online Support Portal:
help.z3controls.com

# NetMeter-OMNI-8C

## 8 Channel Universal Commercial/Industrial Meter/Monitor with Integrated Networking

## MODBUS[1] IMPLEMENTATION MANUAL

DRAFT DOCUMENT

Copyright © 2013 Z3 Controls Inc.

[1] Modbus is a trademark of the Modbus Organization, www.modbus.org.

# CONTENTS

# 1 Important Notice

The NetMeter-OMNI should not be used with voltages above 28V (24V nominal).

In no event will Z3 Controls be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment. The examples and diagrams in this manual are included solely for illustrative purposes.

Because of the many possible use scenarios and requirements associated with any particular installation, Z3 Controls cannot assume responsibility or liability for actual use based on the examples and diagrams.

Any specific application of the NetMeter-OMNI system should be inspected and approved according to local standards and practices.

No patent liability is assumed by Z3 Controls with respect to use of information, circuits, equipment, or software described in this manual.

# 2 Purpose

This document describes the implementation of the Modbus protocol used in the Z3 Controls NetMeter-OMNI devices. It is intended to assist control system programmers to utilize NetMeter-OMNIs in their SCADA/HMI systems or in building automation systems.

This document assumes that the user understands the general use of Modbus. More information about the Modbus protocol is available at the Modbus Organization's web site: www.modbus.org

# 3 Introduction

The NetMeter-OMNI implements a subset of the TCP/IP variant of the Modbus specification.

The NetMeter-OMNI acts as a Modbus "slave" device and will only respond on the Ethernet network when queried by a Modbus "master". When queried appropriately by a Modbus master, the NetMeter-OMNI is capable of reporting a complete set of data representing the state of the inputs and counters etc.

Summary of Key NetMeter-OMNI Modbus Features:

- Modbus TCP server compliant with "Modbus Application Protocol Specification V1.1b" and "Modbus Messaging on TCP/IP Implementation Guide V1.0b"[2]
- Provides a complete set of data parameters
- Data available in both integer and floating point formats
- Supports 2 concurrent Modbus TCP connections
- Operates concurrent with the NetMeter-OMNI's advanced and user friendly web interface

[2] Available at www.modbus.org

- NetMeter-OMNI web interface provides an online Modbus Map to simplify configuration of PLC/SCADA/BAS systems

# 4 Data Formats

Data is available in three formats:

- 16 bit integers signed (S16) or unsigned (U16)
- 32 bit integers signed (S32) or unsigned (U32)
- 64 bit integers signed (S64) or unsigned (U64)
- 32 bit floating point

Most data is available in both integer and float formats. The real-time clock is the only exception. It is a 32 bit count of the number of seconds since midnight of January 1, 2010. It is transmitted only as a 32 bit unsigned integer.

Most integer numbers require scaling, and floating-point numbers do not.

For systems which cannot handle floating point numbers, integer values may be used. Each of these must be multiplied by a scale factor and, if required, have an offset added to convert the number to a value of the correct unit (Volts, Amps, CFM, ºC, etc.). See Chapter 5 (Modbus Map) on page 5 for more about how to scale integers.

Note that all data is transmitted as big endian. Consequently, Float and 32 bit integers are sent as bits 31:16 in the first (lowest) register address and 15:0 in the second (highest) register address. 64 bit integers are captured with reads to 4 consecutive Modbus registers starting with bits 63:48 (lowest register) followed by 47:32, 31:16, and finally 15:0.

## 4.1 16 Bit Integer Data Types

Since Modbus registers are naturally 16 bit, data is strait forward. However, some of the data types are signed values and should be interpreted as 2's compliment values.

## 4.2 32 Bit Integer Data Types

32 bit data must be read as 2 consecutive 16 bit registers. The register with the lowest address contains the most significant 16 bits of data. The higher address is then the least significant 16 bits of data. Each pair of 16 bits should be read using a single Modbus command in order to ensure that the data remains consistent for the upper and lower halves.

## 4.3 64 Bit Integer Data Types

64 bit data must be read as 4 consecutive 16 bit registers. The register with the lowest address contains the most significant 16 bits of data. The highest (4th) address is then the least significant 16 bits of data. Each set of four 16 bit registers should be read using a single Modbus command in order to ensure that the 64 bit data value remains consistent. Otherwise, it is possible for lower portions of the 64 bit data to change after the upper portion has been read.

### 4.4 Floating Point Data Types

Floating point values are in standard IEEE-754 32 bit format. Since Modbus provides only 16-bit registers, two registers must be read to obtain all 32 bits.

To eliminate the possibility that data may change between reading the two 16-bit halves of the 32 bit floating point value, each pair of 16 bits should be read using a single Modbus command.

# 5 Modbus Map

The Modbus map is given in Table 1. It has these main sections:

- 16 bit registers starting at Element 1 (Address 0x0000). These are the analog value of each of the channels: either voltage or current depending on how the NetMeter-OMNI is configured.
- 32 bit registers starting at Element 129 (Address 0x0080). Here is found the lower 32 bits of each counter and also the value for Time.
- 64 bit registers starting at Element 193 (Address 0x00C0). Here is found the full 48 bits of each counter. The upper 16 bits of the 64 bit counter register will always read as 0 thus making it effectively a 48 bit value.
- Floating point versions of the analog input values and 48 bit counters starting at Element 257 (Address 0x0100)

Table 1 shows 2 methods of Modbus addressing:

1. Data Element number: these start at 40001 for the Input Register space and start at 30001 for the Holding Register space. This follows the Modicon convention for point addressing.
2. Data Address: this is the physical address sent out in the Modbus packet and is zero based.

Some of the scale factors shown in Table 1 are variable based on how the sensor is set up. If user defined scale factors have been configured in the NetMeter-OMNI Sensor Configuration actual values, the NetMeter-OMNI web interface provides a complete table containing actual values for the specific NetMeter-OMNI setup. Using the NetMeter-OMNI's web interface, this information is found under:

**Setup** → **Modbus**

As firmware revisions change, the information in Table 1 may also change. Consequently, the Modbus Table available directly from the NetMeter-OMNI should always be regarded as definitive.

Modbus register locations not documented in Table 1 are undefined and may return any value.

Table 1: Modbus Register Map

| Data Element | Data Address | Bits | Register Name | Units | Format | Scale Factor[3] |
|---|---|---|---|---|---|---|
| 40001 | 0 (0x0000) | 15:0 | AIN(0) | V/mA[4] | S16 | 0.001 |
| 40002 | 1 (0x0001) | 15:0 | AIN(1) | V/mA | S16 | 0.001 |
| 40003 | 2 (0x0002) | 15:0 | AIN(2) | V/mA | S16 | 0.001 |
| 40004 | 3 (0x0003) | 15:0 | AIN(3) | V/mA | S16 | 0.001 |
| 40005 | 4 (0x0004) | 15:0 | AIN(4) | V/mA | S16 | 0.001 |
| 40006 | 5 (0x0005) | 15:0 | AIN(5) | V/mA | S16 | 0.001 |
| 40007 | 6 (0x0006) | 15:0 | AIN(6) | V/mA | S16 | 0.001 |
| 40008 | 7 (0x0007) | 15:0 | AIN(7) | V/mA | S16 | 0.001 |
| 40129 | 128 (0x0080) | 31:16 | COUNT(0) | Counts | U32 | 1 |
| 40130 | 128 (0x0081) | 15:0 | | | | |
| 40131 | 130 (0x0082) | 31:16 | COUNT(1) | Counts | U32 | 1 |
| 40132 | 130 (0x0083) | 15:0 | | | | |
| 40133 | 132 (0x0084) | 31:16 | COUNT(2) | Counts | U32 | 1 |
| 40134 | 132 (0x0085) | 15:0 | | | | |
| 40135 | 134 (0x0086) | 31:16 | COUNT(3) | Counts | U32 | 1 |
| 40136 | 134 (0x0087) | 15:0 | | | | |
| 40137 | 136 (0x0088) | 31:16 | COUNT(4) | Counts | U32 | 1 |
| 40138 | 136 (0x0089) | 15:0 | | | | |
| 40139 | 138 (0x008A) | 31:16 | COUNT(5) | Counts | U32 | 1 |
| 40140 | 138 (0x008B) | 15:0 | | | | |
| 40141 | 140 (0x008C) | 31:16 | COUNT(6) | Counts | U32 | 1 |
| 40142 | 140 (0x008D) | 15:0 | | | | |
| 40143 | 142 (0x008E) | 31:16 | COUNT(7) | Counts | U32 | 1 |
| 40144 | 142 (0x008F) | 15:0 | | | | |

[3] The scale factors shown are the for the default units. A user scale factor and offset may be applied to convert V/mA/Counts into other units based on the gain and offset of the attached sensor/meter. If the user defined scale factors/offsets are defined in the Sensor Configuration, these values are advertised in the Setup→Modbus menu item of your NetMeter-OMNI.

[4] Each physical analog input channel is configured for either current mode (mA) or voltage mode (V) operation. The actual units (V or mA) is advertised in the Setup→Modbus menu item of your NetMeter-OMNI.

| Data Element | Data Address | Bits | Register Name | Units | Format | Scale Factor[3] |
|---|---|---|---|---|---|---|
| 40145 | 144 (0x0090) | 31:16 | Time | s | U32 | 1 |
| 40146 | 144 (0x0091) | 15:0 | | | | |
| 40193 | 192 (0x00C0) | 63:48 | COUNT(0) | Counts | U64 | 1 |
| 40194 | 192 (0x00C1) | 47:32 | | | | |
| 40195 | 192 (0x00C2) | 31:16 | | | | |
| 40196 | 192 (0x00C3) | 15:0 | | | | |
| 40197 | 196 (0x00C4) | 63:48 | COUNT(1) | Counts | U64 | 1 |
| 40198 | 196 (0x00C5) | 47:32 | | | | |
| 40199 | 196 (0x00C6) | 31:16 | | | | |
| 40200 | 196 (0x00C7) | 15:0 | | | | |
| 40201 | 200 (0x00C8) | 63:48 | COUNT(2) | Counts | U64 | 1 |
| 40202 | 200 (0x00C9) | 47:32 | | | | |
| 40203 | 200 (0x00CA) | 31:16 | | | | |
| 40204 | 200 (0x00CB) | 15:0 | | | | |
| 40205 | 204 (0x00CC) | 63:48 | COUNT(3) | Counts | U64 | 1 |
| 40206 | 204 (0x00CD) | 47:32 | | | | |
| 40207 | 204 (0x00CE) | 31:16 | | | | |
| 40208 | 204 (0x00CF) | 15:0 | | | | |
| 40209 | 208 (0x00D0) | 63:48 | COUNT(4) | Counts | U64 | 1 |
| 40210 | 208 (0x00D1) | 47:32 | | | | |
| 40211 | 208 (0x00D2) | 31:16 | | | | |
| 40212 | 208 (0x00D3) | 15:0 | | | | |
| 40213 | 212 (0x00D4) | 63:48 | COUNT(5) | Counts | U64 | 1 |
| 40214 | 212 (0x00D5) | 47:32 | | | | |
| 40215 | 212 (0x00D6) | 31:16 | | | | |
| 40216 | 212 (0x00D7) | 15:0 | | | | |
| 40217 | 216 (0x00D8) | 63:48 | COUNT(6) | Counts | U64 | 1 |
| 40218 | 216 (0x00D9) | 47:32 | | | | |
| 40219 | 216 (0x00DA) | 31:16 | | | | |
| 40220 | 216 (0x00DB) | 15:0 | | | | |

| Data Element | Data Address | Bits | Register Name | Units | Format | Scale Factor[3] |
|---|---|---|---|---|---|---|
| 40221 | 220 (0x00DC) | 63:48 | COUNT(7) | Counts | U64 | 1 |
| 40222 | 220 (0x00DD) | 47:32 | | | | |
| 40223 | 220 (0x00DE) | 31:16 | | | | |
| 40224 | 220 (0x00DF) | 15:0 | | | | |
| 40257 | 256 (0x0100) | 31:16 | AIN(0) | V/mA | Float | 1 |
| 40258 | 256 (0x0101) | 15:0 | | | | |
| 40259 | 258 (0x0102) | 31:16 | AIN(1) | V/mA | Float | 1 |
| 40260 | 258 (0x0103) | 15:0 | | | | |
| 40261 | 260 (0x0104) | 31:16 | AIN(2) | V/mA | Float | 1 |
| 40262 | 260 (0x0105) | 15:0 | | | | |
| 40263 | 262 (0x0106) | 31:16 | AIN(3) | V/mA | Float | 1 |
| 40264 | 262 (0x0107) | 15:0 | | | | |
| 40265 | 264 (0x0108) | 31:16 | AIN(4) | V/mA | Float | 1 |
| 40266 | 264 (0x0109) | 15:0 | | | | |
| 40267 | 266 (0x010A) | 31:16 | AIN(5) | V/mA | Float | 1 |
| 40268 | 266 (0x010B) | 15:0 | | | | |
| 40269 | 268 (0x010C) | 31:16 | AIN(6) | V/mA | Float | 1 |
| 40270 | 268 (0x010D) | 15:0 | | | | |
| 40271 | 270 (0x010E) | 31:16 | AIN(7) | V/mA | Float | 1 |
| 40272 | 270 (0x010F) | 15:0 | | | | |
| 40385 | 384 (0x0180) | 31:16 | COUNT(0) | Counts | Float | 1 |
| 40386 | 384 (0x0181) | 15:0 | | | | |
| 40387 | 386 (0x0182) | 31:16 | COUNT(1) | Counts | Float | 1 |
| 40388 | 386 (0x0183) | 15:0 | | | | |
| 40389 | 388 (0x0184) | 31:16 | COUNT(2) | Counts | Float | 1 |
| 40390 | 388 (0x0185) | 15:0 | | | | |
| 40391 | 390 (0x0186) | 31:16 | COUNT(3) | Counts | Float | 1 |
| 40392 | 390 (0x0187) | 15:0 | | | | |
| 40393 | 392 (0x0188) | 31:16 | COUNT(4) | Counts | Float | 1 |
| 40394 | 392 (0x0189) | 15:0 | | | | |
| 40395 | 394 (0x018A) | 31:16 | COUNT(5) | Counts | Float | 1 |
| 40396 | 394 (0x018B) | 15:0 | | | | |

| Data Element | Data Address | Bits | Register Name | Units | Format | Scale Factor[3] |
|---|---|---|---|---|---|---|
| 40397 | 396 (0x018C) | 31:16 | COUNT(6) | Counts | Float | 1 |
| 40398 | 396 (0x018D) | 15:0 | | | | |
| 40399 | 398 (0x018E) | 31:16 | COUNT(7) | Counts | Float | 1 |
| 40400 | 398 (0x018F) | 15:0 | | | | |

# 6 Supported Modbus Functions

The NetMeter-OMNI is a Modbus read only device. That is, there are no control points present in the NetMeter-OMNI. However, the NetMeter-OMNI can act as a gateway for control devices on its ZCAN control network.

**Table 2: Modbus Command Support**

| Function Code | Command | Description |
|---|---|---|
| 1 | Read Coil Status | Reserved for future use. Will generate exception code 02 (Illegal Data Address) |
| 2 | Read Input Status | Reserved for future use. Will generate exception code 02 (Illegal Data Address) |
| 3 | Read Holding Registers | Reserved for future use. For the current version of the firmware this operates the same as the Read Input Registers command. However, this behaviour should not be relied upon for compatibility with future firmware releases. |
| 4 | Read Input Registers | Used by the NetMeter-OMNI to return electrical data (see Table 1). For registers that are un-implemented, the value of 0x0000 will result. |
| 5 | Write Single Coil | Reserved for future use. Will generate exception code 02 (Illegal Data Address) |
| 6 | Write Single Register | Reserved for future use. Will generate exception code 02 (Illegal Data Address) |
| 15 | Write Multiple Coils | Reserved for future use. Will generate exception code 02 (Illegal Data Address) |
| 16 | Write Multiple Registers | Reserved for future use. Will generate exception code 02 (Illegal Data Address) |
| 17 | Report Slave ID | Returns byte count of 2: Slave ID=0, RUN indicator status (0 = OFF, FF = ON) |
| All others | | Will generate exception code 01 (Illegal Function Code) |

# 7 Transactions for Supported Modbus Functions

This section describes the Request/Response packets for each of the supported Modbus Functions. This section assumes basic knowledge of the Modbus Protocol as documented in:

- MODBUS Application Protocol Specification V1.1b
- MODBUS Messaging on TCP/IP Implementation Guide V1.0b

These documents are available at the Modbus web site: www.modbus.org

The NetMeter-OMNI supports 2 concurrent Modbus TCP connections. That is, 2 Modbus client devices (masters) can simultaneously remain connected to the NetMeter-OMNI. If Modbus clients disconnect from the socket between each query (or a sequence of queries) rather than keeping the socket connected, then even more clients may poll the NetMeter-OMNI.

The number of total Modbus TCP sockets and the number of sockets currently available is reported in the Modbus Setup page of the NetMeter-OMNI web interface.

## *7.1 Modbus Unit Identifier*

The Modbus Unit Identifier is used to address multiple physical or virtual Modbus devices that may be connected through a single IP address on a TCP/IP network and is like a "slave address" for Modbus-RTU.

The NetMeter-OMNI is polled using a Modbus Unit Identifier (UID) value of 1 (0x01) in order to obtain the data documented in Table 1. A UID value of 1 should be used if possible. Other UIDs up to 250 (0xFA) will also respond as above. However, future ZCAN modules will be mapped into UID 0x02 and above so UID=0x01 should be used when possible for future compatibility.

The Modbus Unit Identifier value of 0 (Broadcast) will be received by the NetMeter-OMNI and no response will be transmitted back by the NetMeter-OMNI. All broadcast requests to the NetMeter-OMNI are currently ignored but broadcast features may be added to the NetMeter-OMNI in the future.

Modbus Unit Identifier values greater than 250 (0xFA) are reserved for future use and will return an exception code of 0x0B (GATEWAY TARGET DEVICE FAILED TO RESPOND)

## *7.2 Modbus Transaction by Function*

Each of the following request/response tables are color-coded according to the following code:

| | |
|---|---|
| Purple | MBAP Header (MODBUS Application Protocol) |
| Green | PDU (Protocol Data Unit) |

See the document "MODBUS Messaging on TCP/IP Implementation Guide V1.0b" for more information on this terminology.

### 7.2.1 Function 17 (0x11): Report Slave ID

**Table 3: Report Slave ID Request**

| Offset | Size | Field | Value | Description |
|---|---|---|---|---|
| 0 | 2 | Transaction Identifier | TID | The value is chosen by the Modbus Client(master) device and will be sent back in the response by the NetMeter-OMNI |
| 2 | 2 | Protocol Identifier | 0x0000 | 0=MODBUS protocol, the same value will be sent back in the response by the NetMeter-OMNI |
| 4 | 2 | Length | 3 | Number of following bytes |
| 6 | 1 | Unit Identifier | UID=1 | See Section 7.1 |
| 7 | 1 | Function | 0x11 | Report Slave ID function code |

**Table 4: Report Slave ID Normal Response**

| Offset | Size | Field | Value | Description |
|---|---|---|---|---|
| 0 | 2 | Transaction Identifier | TID | Value is copied from the request |
| 2 | 2 | Protocol Identifier | PID | Value is copied from the request, should normally be 0x0000. |
| 4 | 2 | Length | N+5 | Number of following bytes |
| 6 | 1 | Unit Identifier | UID=1 | Value is copied from the request |
| 7 | 1 | Function | 0x11 | Report Slave ID function code |
| 8 | 1 | Byte Count | N+2 | Number of following bytes |
| 9 | 1 | Slave ID | 0 | NetMeter-OMNI Slave ID=0 |
| 10 | 1 | Run Status | 0x00 or 0xFF | Run Indicator Status 0x00=OFF, 0xFF=ON (NetMeter-OMNI is enabled for proper data capture) |
| 11 | N | ID String | See Text | An ASCII string is transmitted in little endian order. It contains a list of comma separated fields:<br>NetMeter-OMNI MAC Address, NetMeter-OMNI Sensor Model#, Firmware Version, Firmware Build Number<br>Additional fields may be added in future releases. |

The Slave ID Exception Response (Table 5) is generated when the Unit Identifier is 2 or higher.

**Table 5: Report Slave ID Exception Response**

| Offset | Size | Field | Value | Description |
|---|---|---|---|---|
| 0 | 2 | Transaction Identifier | TID | Value is copied from the request |
| 2 | 2 | Protocol Identifier | PID | Value is copied from the request |
| 4 | 2 | Length | 3 | Number of following bytes |
| 6 | 1 | Unit Identifier | UID=1 | Value is copied from the request |
| 7 | 1 | Function | 0x91 | Report Slave ID exception function code |
| 8 | 1 | Exception Code | 0x04 | SLAVE_DEVICE_FAILURE |

### 7.2.2 Function 3 (0x03): Read Holding Registers

The Function 3 packets are the same as the Function 4 packets except that the value of the “Function” (offset 7) is 0x03 instead of 0x04. See Section 7.2.3.

### 7.2.3 Function 4 (0x04): Read Input Registers

**Table 6: Read Input Registers Request**

| Offset | Size | Field | Value | Description |
|---|---|---|---|---|
| 0 | 2 | Transaction Identifier | TID | The value is chosen by the Modbus Client device and will be sent back in the response by the NetMeter-OMNI |
| 2 | 2 | Protocol Identifier | 0x0000 | 0 = MODBUS protocol, the same value will be sent back in the response by the NetMeter-OMNI |
| 4 | 2 | Length | 6 | Number of following bytes |
| 6 | 1 | Unit Identifier | UID = 1 | See Section 7.1 |
| 7 | 1 | Function | 0x04 | Read Input Registers function code |
| 8 | 2 | Starting Address | | Address of the first Input Register to be loaded. See the Modbus Map on page 5 |
| 10 | 2 | Quantity of Registers | N = 1 to 125 | The number of 16 bit registers to be loaded. |

**Table 7: Read Input Registers Normal Response**

| Offset | Size | Field | Value | Description |
|---|---|---|---|---|
| 0 | 2 | Transaction Identifier | TID | Value is copied from the request |
| 2 | 2 | Protocol Identifier | PID | Value is copied from the request |
| 4 | 2 | Length | 2×N + 3 | Number of following bytes |
| 6 | 1 | Unit Identifier | UID = 1 | Value is copied from the request |
| 7 | 1 | Function | 0x04 | Read Input Registers function code |
| 8 | 1 | Byte Count | 2×N | Number of following bytes |
| 9 | 2×N | Register Values | | Values for the requested Input Register sequence |

Z3 Controls

www.z3controls.com

support@z3controls.com

Phone: 1-877-454-4436

4261-A14 Highway #7 East, Unit 290
Markham, ON L3R 9W6
Canada

Copyright © 2012 Z3 Controls Inc.