MODERN ENERGY MONITORING

Modbus Register Map

EB – eTactica Current Bar

WE MAKE ENERGY
MONITORING SIMPLE

v1.5
20.06.2016

Revision history

Version	Action	Author	Date
1.0	Initial document	KP	25.08.2013
1.1	Document review, description and register update	GP	26.08.2013
1.2	Status bits, current noise floor	GP	29.08.2013
1.3	Using EG100 as a Modbus TCP/RTU bridge	GP	28.10.2013
1.4	Separate document for each device	GP	07.07.2014
1.5	Brand changed to eTactica, names of meters changed, the command register was updated.	ÁH	20.06.2016

Introduction

All eTactica hardware devices are interconnected using the Modbus/RTU protocol over half-duplex RS485 serial interface. For a complete eTactica installation, the eTactica gateway EG-200 is always the Modbus master and all other measurement devices are pre-configured as Modbus slaves.

The eTactica measurement devices are as follows:

- The eTactica Power Meter, EM-xxx
- The eTactica Current Bar, EB-2xx
- The eTactica Current Meter, ES-xxx

The eTactica measurement devices implement a register table with both configurable and read-only parameters. These parameter values are accessible via standard Modbus requests.

The eTactica measurement devices can be part of any 3rd party data collection installation that supports Modbus/RTU. This means that a 3rd party Modbus master can be used instead of the eTactica gateway EG-200. For this purpose, you can either connect directly via Modbus/RTU over RS485 or via Modbus/TCP using the EG-200 as a standalone Modbus TCP/RTU bridge.

To use 3rd party data collector, simply configure your Modbus Master for the eTactica devices as described in this document and note that:

- You must allow up to 500 msec for response
- The minimum read-request interval from the Modbus Master, on each device, must be set to 100 msec or longer

References

The Modbus protocol specification:
http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

RS485 Serial Settings

All eTactica hardware devices have default settings for the RS485 serial interface:

- 19200 baud rate
- 8 data bits
- Even parity
- 1 stop bit

These settings are editable by writing to given registers, limited to devices with firmware version 3.2 and above.

Modbus Supported Functions

All eTactica hardware devices support the following Modbus function codes:

- 0x03 – Read Holding Registers
- 0x10 – Write Multiple Registers

Data Format and Addressing

Unless otherwise noted, each register value is an unsigned 16-bit integer. Signed values are regular 2's Complement Signed.

Data Encoding

According to the Modbus protocol specification the Big-Endian representation of both data and addresses is used. This means that the most significant byte (MSB) is sent first.

Addressing

The first data entry in the Modbus table for every eTactica measurement device, is at data address 0x2000 (the vendor ID) or 8192. This is not 40001 or 0x9C41, as you might expect for accessing the first entry of analog holding registers. If you are using one of many Modbus software drivers (PC or PLC masters) that expect 40001 as the first entry for holding registers, you need to remember to add the offset of 8192.

See these pages for more of an explanation:

- http://www.csimn.com/CSI_pages/Modbus101.html#mb101_reg1
- http://www.simplymodbus.ca/faq.htm#Map

The Register Map below lists the data addresses to use when forming the Modbus request (ADU message format) to each of the eTactica measurement devices.

Examples

Byte and Register ordering

As specified in Section 4.2 of the Modbus Application Protocol Specification, all values are stored in Big Endian, MSB first order. All register addresses in this document are "PDU Addresses" as per Section 4.4. In other words, the first register (Vendor ID) is accessed at register address 0 (plus the offset of 8192 (0x2000)). Note that some Modbus applications refer to this first register as "Modbus Data Model" register 1, which is then at address 0.

Values marked as 32bit, are *also* stored in Big Endian, MSB first, as would be implied by a sensible reading of section 4.2. 64 bit values are also stored Big Endian, MSB first.

Example 16bit value

Read a 16-bit value, Line frequency on EM-xxx, data address 0x200F.

PDU	
Function code	0x03
Starting address	0x200F
Quantity of registers	1
PDU message	0x03 – 0x20 – 0x0F – 0x00 – 0x01

Final Value	**Value Stored**	**Register high byte**	**Register low byte**
50.42 Hz	50420 (Register stores mHz)	0xC4	0xF4

Example 32bit value

Read a 32 bit value, Current on Channel 0 on ES-xxx or EB-xxx, data address 0x2016.

Register 0 = 0x2016, Register 1 = 0x2017

PDU	
Function code	0x03
Starting address	0x2016
Quantity of registers	2
PDU message	0x03 – 0x20 – 0x16 – 0x00 – 0x02

Final Value	**Value Store**	**Register 0 high**	**Register 0 low**	**Register 1 high**	**Register 1 low**
320.123 Amps	320123 (Value in mA) (0x4E27B)	0x00	0x04	0xe2	0x7b

Modbus Register Map

Common Registers

Below you find the registers, common to all eTactica measurement devices.

Register Address	R/W	Description
0x2000	R	Vendor id (0x524d)
0x2001	R	Product id
0x2002	R	Firmware version
0x2003	R	Serial number bytes 0..1
0x2004	R	Serial number bytes 2..3
0x2005	R	Serial number bytes 4..5
0x2006	R/W	**Command**
0x2007	R	Total register count
0x2008	R/W	**Serial communication settings**
0x2009	R/W	**Modbus slave ID**
0x200A		Reserved
0x200B		Reserved
0x200C		Reserved
0x200D		Reserved

Modbus Registers – EB Specific

The Modbus register map for the family of eTactica Current Bars.
Note that for a 3 point device (i.e. EB-203), only the registers for current breaker 0 to 2 are valid, and likewise for 6 and 9 point devices.

Register Address	R/W	Description
0x200E	R	Temperature
0x200F		Reserved
0x2010	R/W	total number of breakers
0x2011	R/W	disabled breakers (bitmap)
0x2012		Reserved
0x2013		Reserved
0x2014		Reserved
0x2015		Reserved
0x2016	R	current breaker 0 (mA) – (31..16)
0x2017	R	current breaker 0 (mA) – (15..0)
0x2018	R	current breaker 1 (mA) – (31..16)
0x2019	R	current breaker 1 (mA) – (15..0)
0x201A	R	current breaker 2 (mA) – (31..16)
0x201B	R	current breaker 2 (mA) – (15..0)
0x201C	R	current breaker 3 (mA) – (31..16)
0x201D	R	current breaker 3 (mA) – (15..0)
0x201E	R	current breaker 4 (mA) – (31..16)
0x201F	R	current breaker 4 (mA) – (15..0)
0x2020	R	current breaker 5 (mA) – (31..16)
0x2021	R	current breaker 5 (mA) – (15..0)
0x2022	R	current breaker 6 (mA) – (31..16)
0x2023	R	current breaker 6 (mA) – (15..0)

Register Address	R/W	Description
0x2024	R	current breaker 7 (mA) – (31..16)
0x2025	R	current breaker 7 (mA) – (15..0)
0x2026	R	current breaker 8 (mA) – (31..16)
0x2027	R	current breaker 8 (mA) – (15..0)
0x2028	R	current breaker 9 (mA) – (31..16)
0x2029	R	current breaker 9 (mA) – (15..0)
0x202A	R	current breaker 10 (mA) – (31..16)
0x202B	R	current breaker 10 (mA) – (15..0)
0x202C	R	current breaker 11 (mA) – (31..16)
0x202D	R	current breaker 11 (mA) – (15..0)
0x202E		Reserved
0x202F		Reserved
0x2030		Reserved
0x2031		Reserved
0x2032		Reserved
0x2033		Reserved
0x2034		Reserved
0x2035		Reserved
0x2036		Reserved

Detailed Register Descriptions

Register 0x2006 – Command

The command register is a 16 bit value. You use this register to permanently store new configuration settings in EEPROM or reload factory default. The meaning of each bit and bit combination is described in the table below.

Bit #	Description
15	Reserved
14..6	unused
5	"Frequency mode", Enable this bit to have a 60 Hz digital filter, or leave it at 0 to keep the 50 Hz digital filter.(just for EB and ES with firmware version 3.14 or higher.)
4	Reserved
3	LED Control State (1 == LED on, 0 == LED off)
2	LED Control State Valid (1 == bit 3 is valid, 0 == bit 3 is ignored)
1	Enable this bit to reload default device configuration to RAM (use in conjunction with bit 0 to reset EEPROM to factory defaults)
0	Enable this bit to store current configuration to EEPROM and restart device

Register 0x2008 - Serial communication settings

The serial communication register is a 16 bit value. It allows you to edit the protocol settings for the RS485 serial interface. Take care modifying these settings. It can be tedious to rediscover what the settings are, for an unknown device.

Default settings for all devices is: 19200 - 8 - Even - 1

Firmware Limitation

The editable feature is only available for devices with firmware version 3.2 or above.

After writing a value to this register, you must write to the **Command** register (0x2006) to store settings in EEPROM and reinitialize the device. This will make the new settings take effect.

Bit #	Description
15..12	Stop Bits (normally 1, 2 is also allowed) 0001 (0x01) : 1 stop bit 0010 (0x02) : 2 stop bits
11..8	Parity (0: None, 1: Odd, 2: Even) 0000 (0x00) : Parity none 0001 (0x01) : Parity odd 0010 (0x02) : Parity even
7..0	Baud rate value (See table below)

Baud Rate Table

Values to write as the lowest byte in this register that represent pre-defined baud rates.

Lowest Byte of 0x2008	Baud Rate
0000 (0x00)	default (19200 at present)
0001 (0x01)	600
0010 (0x02)	1200
0011 (0x03)	2400
0100 (0x04)	4800
0101 (0x05)	9600
0110 (0x06)	19200
0111 (0x07)	38400
1000 (0x08)	57600
1001 (0x09)	115200

Examples

Contents of register 0x2008	Description
0x1200	Factory Default, 1 Stop bit, Even Parity, Default Baud Rate (19200)
0x1005	1 Stop Bit, No Parity, 9600 Baud
0x1209	1 Stop Bit, Even Parity, 115200 Baud
0x1101	1 Stop bit, Odd Parity, 600 Baud
0x0044	Don't do this! (Unexpected values will be converted to 1 Stop bit, No Parity, 115200)

Register 0x2009 - Modbus slave ID

The Modbus slave ID register, is a 16 bit value. It is a configurable register where you can modify the default slave ID for your device. Only the lower byte for this 16 bit value is valid for the slave ID. Take care to preserve the upper byte as is.

According to the Modbus protocol, it is only allowed to use addresses from 1 - 247.

Bit #	Description
15..8	Reserved, do not modify contents
7..0	Modbus slave ID (values from 1 to 247)