

# Modbus Output Communication

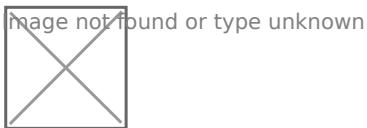
All eGauge models with firmware v3.2 or higher have Modbus TCP output capability. eGauge models with USB ports also have capability for Modbus RTU (serial via RS485) output capability, with use of the [eGauge USB485 converter](#). The Modbus server can be enabled in Settings -> Modbus Server. If the Modbus Server option is not available, upgrade your firmware in Tools -> Firmware Upgrade.

NOTICE: Prior to firmware v4.0, the Modbus map addresses are **liable to change** if the register name or contents is modified. Firmware v4.0 adds additional ranges that are tied to the register ID. The addresses used prior to v4.0 are still usable to provide backwards-compatibility, but are marked as deprecated and are not recommended to use as installation settings reconfiguration may change them. **Virtual registers** are in the order they appear on the settings page, meaning addresses can change if virtual registers are added or deleted.

---

## Enabling the Modbus server

In the eGauge interface, navigate to Settings -> Modbus Server. The following options are available:



Check the box for "Enable Modbus" for the appropriate service (Modbus TCP uses Ethernet, Modbus RTU uses the eGauge USB485 serial converter).

For Modbus TCP, choose the port to listen on (default: 502).

For Modbus RTU, click inside the serial device box and detected adapters will be listed to choose from as shown below. Choose the appropriate baud rate, parity and stop bits to match the Modbus master.



# Modbus Register Map

The exact Modbus map depends on the eGauge device configuration and is available on the Modbus settings screen. The map is available for viewing in the web browser or for download as a comma-separated value (CSV) file which can then be imported into any spreadsheet program.

Address range	Description
<b>Timestamps</b>	
30000 -- 30003	local timestamp (for use when reading from addresses <35000)
30004 -- 30007	THD timestamp indicates when data was acquired that is used by FFTarg(), FFTarg2(), FFTmag(), and THD() functions
30008 -- 30009	register timestamp (for use when reading from addresses ≥35000)
<b>Local data (channel checker)</b>	
30500 -- 30999	Line voltages (RMS)
31000 -- 31499	Line voltages (DC/mean)
31500 -- 31999	Line frequencies
32000 -- 32499	Sensor values (normal/RMS)
32500 -- 32999	Sensor values (mean/DC)
33000 -- 33949	Sensor frequencies
<b>Register Data [v4.0 and greater]</b>	
34000 -- 34999	Reduced precision cumulative physical register values (float)
35000 -- 35999	Cumulative virtual register values (signed 64-bit integers)
36000 -- 36999	Change of virtual register value (float)
37000 -- 37999	Reduced precision cumulative virtual register values (float)
38000 -- 38999	Cumulative physical register values (signed 64-bit integers)

39000 -- 39999	Change of physical register value (float)
----------------	---

	Register Data [PRIOR to firmware v4.0, now deprecated]
35000 -- 35999	Cumulative register values (signed 64-bit integers) [DEPRECATED]
36000 -- 36999	Change of register value (float) [DEPRECATED]
37000 -- 37999	Reduced precision cumulative register values (float) [DEPRECATED]

In **v4.0 or greater**, the Modbus addresses are tied to the eGauge register IDs. They can be calculated as:

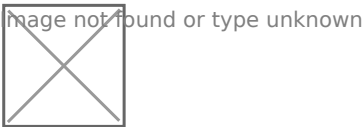
Physical float cumulative value = 34000+(ID\*2)

Physical s64 cumulative value = 38000+(ID\*4)

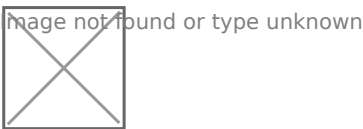
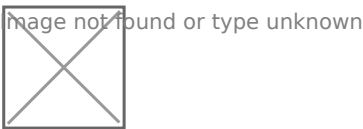
Physical float instant value = 39000+(ID\*2)

Virtual registers are in the order they appear on the installation settings page, and are therefore liable to shift during deletions or additions.

The register ID can be found by hovering over the register delete button in the Installation Settings page:



An example Modbus map, which can be found in Settings -> Modbus Server, is displayed below:



The t16 type Modbus registers are variable-length UTF-8 strings and therefore multiple registers may need to be read until a UTF-8 null character is reached.

# Example Modbus Transmissions:

## Example Modbus TCP request and response for L1 RMS voltage:

Request: [00 01] [00 00] [00 06] [01] [04]  
[01 f4] [00 02]

[00 01] = Transaction identifier

[00 00] = Protocol identifier (always 00 00)

[00 06] = Bytes following (0x0006 = 6 bytes)

[01] = Unit ID

[04] = Function code (0x04 is read input registers)

[01 f4] = Starting register address (500)

[00 02] = Number of words to return (2 words for float value)

Response: [00 01] [00 00] [00 07] [01]  
[04] [04] [42 f6 2a 06]

[00 01] = Transaction identifier

[00 00] = Protocol identifier (always 00 00)

[00 07] = Bytes following (0x0007 = 7 bytes)

[01] = Unit ID

[04] = Function code (0x04 is read input registers)

[42 f6 2a 06] = Contents of register 500 and 501 respectively

0x42f62a06 decoded as big-endian (ABCD) float is roughly 123.08, so L1 voltage is 123.08Vrms

## Example Modbus RTU request and response for L1 RMS voltage:

Request: [01] [04] [01 f4] [00 02] [31 c5]

[01] = Unit ID

[04] = Function code (0x04 is read input registers)

[01 f4] = Starting register address (500)

[31 c5] = CRC error check

Response: [01] [04] [04] [42 f6 2a 06] [90 ac]

[01] = Unit ID

[04] = Function code (0x04 is read input registers)

[04] = Data bytes to follow (not including CRC)

[42 f6 2a 06] = Contents of register 500 and 501 respectively

0x42f62a06 decoded as big-endian (ABCD) float is roughly 123.08, so L1 voltage is 123.08Vrms

---

## Additional technical information

The maximum Modbus frame size is 255 bytes. This translates to (61) 32-bit registers or (30) 64-bit registers. Most DAS and Modbus systems will automatically split any requests to limit the response size to not exceed the upper limit. If requesting more registers than can be returned in a single frame, the response may be truncated or an exception may be returned.

Additional help is available by clicking the [?] buttons on the Modbus Server settings page.

All data is provided as Modbus input registers and therefore their Modbus addresses start at 30000. The raw frame sent does not include the prefixed '30', this indicates the function code used.

For each register recorded by eGauge, the register's current cumulative value and the amount by which it changed since the previous one-second interval can be read. The cumulative value is available with full resolution as a signed 64-bit value. For convenience, the same value is also

available, at reduced resolution, as a 32-bit floating point number. The change-in-value is available only as a 32-bit floating point number. In addition to the register data, it is also possible to read locally measured data such as line-voltages, frequencies, and the value measured by each sensor-port.

The Modbus server guarantees that any data read with a single request is consistent. Timestamps are provided as a means to ensure consistency when data is read with multiple requests (e.g., to read multiple, discontinuous registers). This is accomplished by (i) reading the timestamp, (ii) performing the desired read requests, and (iii) reading the timestamp again. If the two timestamp values are the same, then reader can be certain that the data is consistent.

---

Please visit [kb.egauge.net](http://kb.egauge.net) for the most up-to-date documentation.