



eGauge Configuration Guide

(for firmware v4.5 or newer)

eGauge Systems LLC
1644 Conestoga St, Suite 2
Boulder, CO 80301
<http://egauge.net/>

September 11, 2023

Before continuing, we highly recommend checking out our Getting Started guide at <http://egauge.net/start>

This guide explains how to configure eGauge so it properly reflects the physical reality of an installation.

Please take a look through the Table of Contents on the following page.

The first section (1) has general information about connecting to and accessing the meter interface. Section 2 covers the different pages required to configure the meter and the available options for each.

The third section (3) contains wiring diagrams and configuration examples of common installation scenarios.

The final section, 4 section describes troubleshooting and possible correction methods through software that do not require an on-site visit.

Appendix A has an abridged list of formula register operators and functions. Appendix B has technical details about the different physical unit types registers support.

Please review the Getting Started guide at <http://eGauge.net/start> for a short introduction on installing and configuring the eGauge meter.

Contents

| | | |
|----------|---|-----------|
| 1 | Accessing the Meter | 4 |
| 1.1 | Classic and Modern interfaces | 4 |
| 1.2 | Users and Default Password | 4 |
| 1.3 | Settings Menu Locations | 5 |
| 2 | Installation Configuration Overview | 6 |
| 2.1 | Potential Transformer (PT) Configuration | 8 |
| 2.2 | Sensor Configuration | 8 |
| 2.2.1 | CT orientation | 8 |
| 2.2.2 | Classic CT configuration method | 9 |
| 2.2.3 | CTid® (EG4xxx only) | 10 |
| 2.2.4 | Multiplicative Scale | 12 |
| 2.2.5 | High-gain mode (EG4xxx only) | 12 |
| 2.2.6 | Rope CTs (Rogowski Coils) | 12 |
| 2.3 | Remote Devices | 13 |
| 2.4 | Register Configuration | 14 |
| 2.4.1 | Register Name | 14 |
| 2.4.2 | Register Type | 14 |
| 2.4.3 | Recording Power | 15 |
| 2.4.4 | Formula Registers | 18 |
| 2.5 | Totals and Virtual Registers Configuration | 19 |
| 2.5.1 | Totals | 20 |
| 2.5.2 | Defining Views | 21 |
| 3 | Configuration Examples | 23 |
| 3.1 | Standard Split-Phase back-fed | 24 |
| 3.2 | Direct-Feed Solar | 25 |
| 3.3 | Three-Phase 120/208 or 277/480 Wye w/ single-phase inverter | 26 |
| 3.4 | 3-phase inverter | 27 |
| 3.5 | Submetering single-phase units on a 120/208V service | 28 |
| 3.6 | Three-Phase High-leg Delta | 29 |
| 3.7 | Three-phase 347/600V Wye | 30 |
| 3.8 | Three-Phase Delta (L-L voltages $\leq 277V$) | 31 |
| 3.9 | Three-Phase Delta (L-L voltages $\geq 277V$) | 32 |
| 3.10 | Subpanels | 33 |
| 3.11 | Appliances | 34 |

| | |
|--|-----------|
| 4 Troubleshooting | 35 |
| 4.1 Identify correct phasing | 36 |
| 4.2 Fixing Leg/Phase Mixups | 37 |
| 4.3 Fixing CTs Pointing in the Wrong Direction | 37 |
| A Formula Operators and Functions | 38 |
| B Physical Unit | 41 |
| B.1 Rollover and maximum value example | 42 |

1 Accessing the Meter

Each meter connects to a TCP/IP network and contains an embedded webserver. Assuming a computer is connected to the same local network, the meter may be accessed at the local IP address displayed on the LCD screen. The LCD toggle switch may be used to switch screens until the “Name & IP” screen is displayed.

If the meter has Internet access, it will attempt to connect to the eGauge proxy server at <http://DEVNAME.d.egauge.net> or <http://DEVNAME.egaug.es> where *DEVNAME* is the meter name. The meter name may be found on the label on the back-side of the meter, opposite the sensor inputs. It may also be found on the LCD display. The LCD toggle switch may be used to switch screens until the “Name & IP” screen is displayed. The eGauge proxy server is a complimentary and best-effort service.

You may also visit <http://egauge.net/find> and enter the device name to obtain the proxy-server URL, if the meter is online and connected to the eGauge proxy server.

Visit the knowledgebase at <http://kb.egauge.net/> for LCD information, troubleshooting tips, and much more information.

1.1 Classic and Modern interfaces

The eGauge interface has remained relatively unchanged until firmware version 4.5 introduced the Modern interface. Newer meters will automatically use the Modern interface by default, while older meters will continue to use the Classic interface by default. If the meter is on firmware v4.5 or newer, you may visit the Modern interface through *View* → *Modern user-interface*.

The Modern interface is significantly more customizable and feature-rich and is therefore the recommended interface to use.

For more information on the Modern interface, visit <http://egauge.net/support/modern>.

1.2 Users and Default Password

With the exception of certain resellers or installers who have special default settings, the default username on meters is “owner”.

For meters sold after January 2021, the “owner” password is randomized and printed on a label on the back-side (side opposite of the sensor inputs), and “LAN Access” is not necessary.

The password on meters sold prior to January 2021 is “default”. To change settings with these meters, the computer must be on the same local network as the meter and connected directly via IP address. This can be accomplished by clicking “LAN Access” in the upper-right hand link menu from the Classic Interface.

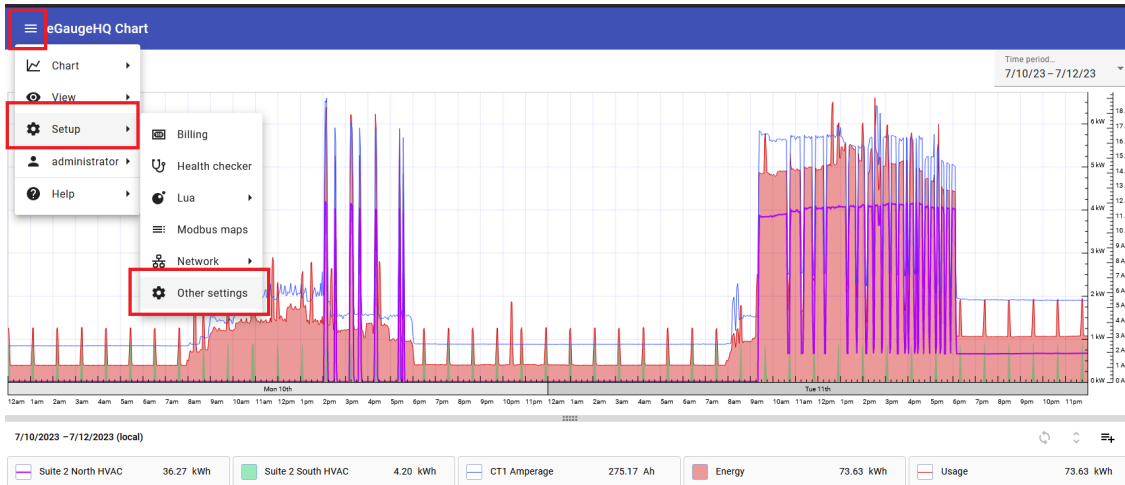
If you receive an error “Sorry, you are not authorized to make changes”, it means “LAN Access” was not used, or the meter is still detecting the system as a remote connection, which could happen through a VPN or other advanced network setup.

To change passwords or modify users, visit “Access Control” from the Settings menu.

1.3 Settings Menu Locations

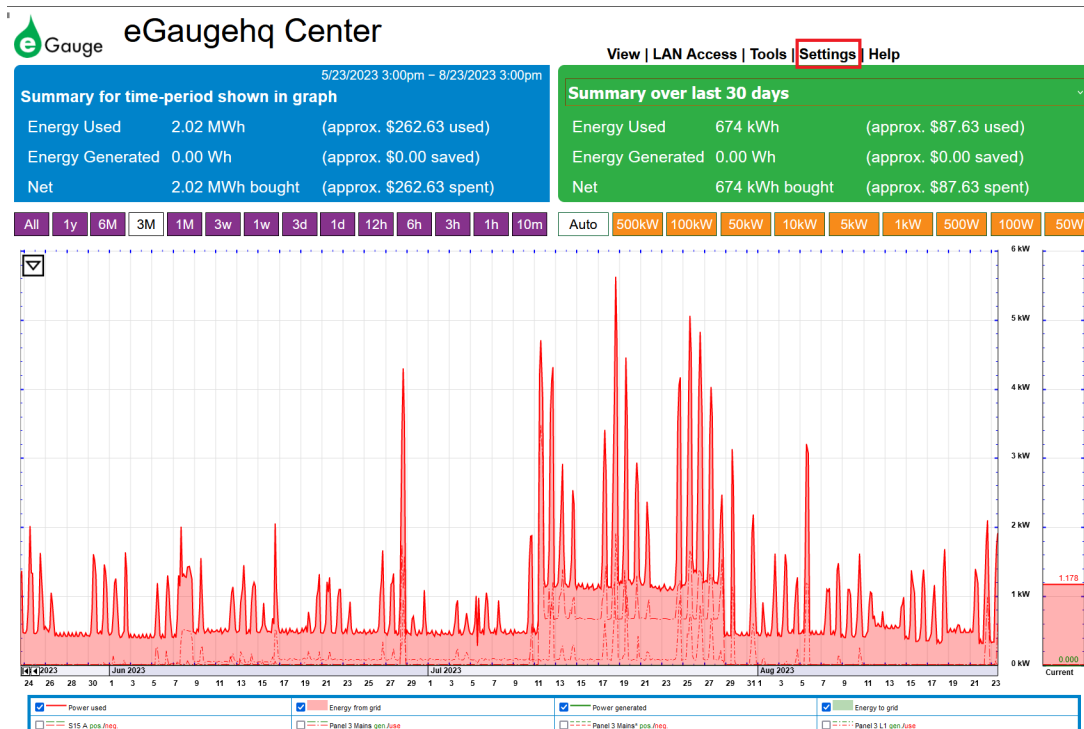
Modern Interface

The settings described in this Configuration Guide may be accessed through the Modern interface’s hamburger menu (3-line button) in the upper left-hand corner in *Setup* → *Other Settings*, as shown in the image below.



Classic Interface

The settings described in this Configuration Guide may be accessed through the Classic interface by clicking “Settings” in the upper right-hand link menu, as shown in the image below.



2 Installation Configuration Overview

As illustrated in Figure 1, the Installation page has five sections:

1. Potential Transformers (PTs)
2. Sensors
3. Remote Devices
4. Registers
5. Totals and Virtual Registers

We describe the purpose of each section in more detail below.

After any of the settings have been changed, they need to be saved by clicking on the “Save” button near the bottom of the page.

To restore the settings which are currently active on the device, click on the “Reset” button. This will revert any changes you have made to the page since the last save.

2.1 Potential Transformer (PT) Configuration

If measuring a system with line-to-line voltages greater than 480V and below 707V, the eGauge EV1000 high-voltage sensors should be used. See configuration examples 3.9 for systems without a neutral or example 3.7 for systems with a neutral for more information on using the EV1000. No configuration is needed in this section if using EV1000. More information about the EV1000 can be found at <http://egauge.net/support/m/ev1000>.

This section is used to inform the device of any potential-transformers (PTs) that may be installed. Usually, no PTs are installed and the settings here can be left at the default value of “direct (no PT)”. However, if PTs have been installed, the voltages that the device will measure at the L1, L2, or L3 pin will be lower than the actual line-voltage and the appropriate voltage-division ratios need to be selected in this section. For example, with PTs that transform 34,500V (34.5kV) down to 120V, the ratio “34.5kV:120V” should be selected.

WARNING: Potential transformers can introduce a phase shift and cause inaccurate measurements.

2.2 Sensor Configuration

This section defines what kind of sensor or current-transformer (CT) is connected to each of the CT positions. Depending on the eGauge model, there are between 12 and 30 inputs. A blank entry indicates that no CT or sensor is configured.

2.2.1 CT orientation

The direction or orientation of the CT matters and affects the polarity (positive or negative) of the power values. While power values are correct in either orientation, having negative values will cause the cumulative values (aka meter readings) to decrease over time.

It is imperative to have CTs oriented in a consistent manner, as power values are typically added together and inconsistent polarity will cause those values to be higher or lower than real. For example, if a three-phase service has 1 kW on each phase (3kW total consumption), but the CTs are oriented inconsistently, they may read 1kW, -1kW, and 1kW. When added together, the total will read 2kW, not 3kW, due to the reversed CT. Note, systems with back-fed generation such as from solar can have polarity reversals when power reverses direction and back-feeding occurs. See section 4 for information on using the Channel Checker to verify readings and power polarity. If a CT is placed in the incorrect direction, it may be reversed in the software by changing the sensor component of a register to the negated sensor component.

Typically, renewable energy systems should read positive when there is active generation, and battery systems should read positive while charging and negative while discharging.

Each CT has either a sticker (label) and/or directional arrow printed on it to indicate the polarity.

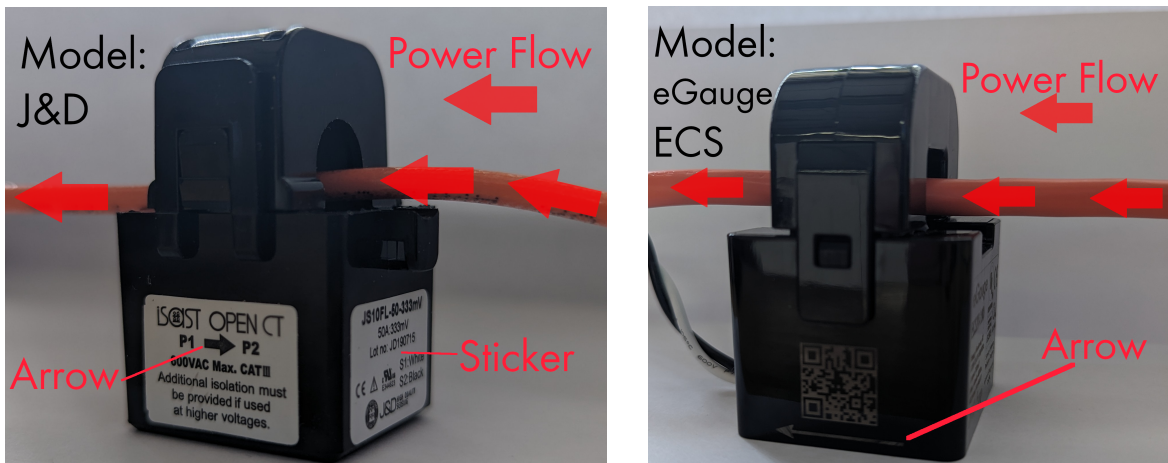
Table 1 below describes for each CT type the identifier (sticker or arrow) and what direction they should face to obtain positive power readings.

Example images of an eGauge ECS and J&D CT with arrow and sticker identifiers can be seen in figure 2 below.

Note, some CTs including ropes may have the arrow molded into the plastic enclosure or clasp rather than on a sticker.

| MANUFACTURER | CT TYPE | IDENTIFIER | DIRECTION |
|-----------------------------|-------------------------------|------------|--|
| AccuEnergy | Rope CT | Arrow | against direction of power flow |
| J&D | Split-core CT | Arrow | against direction of power flow |
| AccuEnergy | Split-core CT (80mm / 3") | Arrow | same direction as power flow |
| eGauge | Split-core CT (ECS series) | Arrow | same direction as power flow |
| Continental Control Systems | Split-core CT (AccuCT series) | Sticker | against direction of power flow |
| Magnelab | Split-core CT (50mm / 2") | Sticker | against direction of power flow |

Table 1: CT orientations for **positive** readings



(a) J&D CT

Note the arrow and sticker faces **against** power flow.
There is not a sticker on the opposite side

(b) eGauge ECS CT

Note arrow faces **same** direction as power flow

Figure 2: Different CT models have different orientation methods

2.2.2 Classic CT configuration method

When using a CTid enabled sensor, skip to section 2.2.3.

When a non-CTid enabled CT is connected to the device, simply select the appropriate CT selection from the pull-down list for the position the CT is plugged into. For example, if a 100A 0.94" J&D split-core CT is plugged into CT position 2, click on the pull-down list for CT2 and select "JD JS 24mm/0.94" 100A".

The format of the CT drop-down option is "[MFG] [MODEL] [SIZE_MM]/[SIZE_IN] [AMPERAGE]"

- [MFG] 2-letter manufacturer code (see table 2)
- [MODEL] CT Model (see table 3)
- [SIZE_MM] Inner-diameter in millimeters
- [SIZE_IN] Inner-diameter in inches
- [AMPERAGE] Amperage rating of CT (not breaker amperage)

| MFG code | Manufacturer |
|----------|-----------------------------|
| JD | J&D |
| CC | Continental Control Systems |
| ML | Magnelab |
| AE | AccuEnergy |
| CR | CR Magnetics |
| EG | eGauge Systems |
| AD | Aim Dynamics |
| HEYI | HEYI |

Table 2: CT manufacturer code description

| MODEL | Type |
|-------|--------------------------------------|
| JS | Split-core CT |
| JRFS | Rogowski coil (rope CT) |
| ACT | High accuracy split-core CT |
| SCT | Split-core CT |
| RCT | Rogowski coil (rope CT) |
| CR | Solid-core DC CT |
| CRS | Split-core DC CT |
| ESA | CTid-enabled Split-core CT |
| ERA | CTid-enabled Rogowski coil (rope CT) |
| KCT | Split-core CT |
| ASR | Rogowski coil (rope CT) |
| SCTE | CTid-enabled Split-core CT |

Table 3: CT model description


2.2.3 CTid® (EG4xxx only)

Available only on the EG4xxx models and with compatible sensors, CTid® allows for automatic detection and configuration of sensor inputs. CTid-enabled sensors also contain an LED that can be blinked through the CTid configuration page.

A selection of CTid enabled CTs and other sensors are sold by eGauge Systems. To configure these sensors, click the blue CTid® button above the CT drop-down menu:

Sensors:

The screenshot shows a configuration interface for sensors. At the top, there is a blue button labeled "CTid®". Below it is a checkbox labeled "Use high-gain mode" which is currently unchecked. At the bottom, there is a dropdown menu for selecting a sensor, with "S1" currently selected and a downward arrow visible.

Check the boxes for the sensor inputs to be scanned for CTid sensors and click the “Scan Checked Sensors” button above it. A green check-mark will display for successfully identified inputs, or a red cross-mark if no CTid sensor is detected. Specific information about the CT may be seen by clicking the “plus” button to the right of the sensor number. The LED can be blinked by pressing the circle in the “Blink LED” column, and a sensor can be deconfigured and removed by clicking the remove button  to the right of the LED button.

Back to Settings Scan Checked Sensors

| <input type="checkbox"/> | Sensor | Model | Last Scanned | Blink LED | | | | | | | | | | | | | | | | | | |
|---|--------------|---|------------------|-----------------------|-------------------|--|--|---------|--------------|------------|------|--------|--------|------|--------|--------|-------|--------|--------|-------|--------|--------|
| <input type="checkbox"/> | 1 | ✓ <input type="checkbox"/> eGauge ECS20 20mm/0.79" 200A SN 1590 AC CT, normal polarity, R _{source} 5Ω, R _{load} 1000000Ω, rated current 200A, voltage 0.333V (+135ppm/°C), bias 0.000mV, phase-shift +0.03° (+1.5m°/°C). | 08/26/23 12:04pm | <input type="radio"/> | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th colspan="3">Calibration Table</th> </tr> <tr> <th>Current</th> <th>Voltage adj.</th> <th>Phase adj.</th> </tr> </thead> <tbody> <tr> <td>1.5%</td> <td>+1.08%</td> <td>-0.20°</td> </tr> <tr> <td>5.0%</td> <td>+0.34%</td> <td>-0.06°</td> </tr> <tr> <td>15.0%</td> <td>+0.20%</td> <td>-0.06°</td> </tr> <tr> <td>50.0%</td> <td>+0.08%</td> <td>-0.02°</td> </tr> </tbody> </table> | | | | | Calibration Table | | | Current | Voltage adj. | Phase adj. | 1.5% | +1.08% | -0.20° | 5.0% | +0.34% | -0.06° | 15.0% | +0.20% | -0.06° | 50.0% | +0.08% | -0.02° |
| Calibration Table | | | | | | | | | | | | | | | | | | | | | | |
| Current | Voltage adj. | Phase adj. | | | | | | | | | | | | | | | | | | | | |
| 1.5% | +1.08% | -0.20° | | | | | | | | | | | | | | | | | | | | |
| 5.0% | +0.34% | -0.06° | | | | | | | | | | | | | | | | | | | | |
| 15.0% | +0.20% | -0.06° | | | | | | | | | | | | | | | | | | | | |
| 50.0% | +0.08% | -0.02° | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | 2 | n/a | | <input type="radio"/> | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | 3 | n/a | | <input type="radio"/> | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> | 4 | n/a | | <input type="radio"/> | | | | | | | | | | | | | | | | | | |

Back on the installation settings page, detected sensors will be displayed with a blue background.



For more information, see <http://egauge.net/support/ctid>

2.2.4 Multiplicative Scale

A multiplicative scale can be applied to the CT reading by entering a number box right of the CT option. This is most commonly used when a CT measures only a portion of the total current being carried on a phase. For example, suppose that a large commercial building receives its utility power through four equal sized conductors per phase. Rather than installing a large CT that encompasses all four conductors, it may be more economical to install a smaller CT that encompasses only a single conductor. Assuming the current on the four conductors is balanced, the smaller CT will capture a **fourth** of the total current per phase. To compensate for that, enter a value of 4 in the multiplier box.

WARNING: in many cases, parallel feeds are not perfectly balanced, and could have significant differences. Because of this, we suggest monitoring each conductor, either by using a CT for each individual conductor, or by using larger diameter CTs to encompass multiple parallel feeds (of the same phase).

2.2.5 High-gain mode (EG4xxx only)

Available only on the EG4xxx models, High-gain mode amplifies the signal to the eGauge sensor ports by 10 times. This allows, for example, a 100A CT to act as a 10A CT. High-gain mode affects **all sensor inputs**. Note, selecting high-gain mode immediately changes the CT drop-down option to its respective lower amperage-rated value. For example, enabling high-gain mode with a 100A CT will change the drop-down option to indicate a 10A CT. Likewise, a CT configured for 75A will appear as a 7.5A CT. **WARNING:** observe the upper-limit for the CT relative to the high-gain mode rating. For example, a 100A CT in high gain mode becomes 10A and cannot read over 10A.

2.2.6 Rope CTs (Rogowski Coils)

Self-powered rope CTs are wired directly to the eGauge with a green 2-pin power plug. Rope CTs are excellent for bus-bars and large switch-gear applications. Self-powered rope CTs are flexible and easy to install but will not be accurate at low amperages and ideally have a minimum of 20A or greater at all times. Rope CTs are configured like split-core CTs.

Some rope CTs may be CTid-enabled and are configured through the CTid interface as referenced in section 2.2.3. Non-CTid enabled ropes are configured the same way as split-core CTs. For example, the 106mm (4") 2775A AccuEnergy rope should be configured as `AE RCT 106mm/4.17" 2775A`.

As rope CTs are inaccurate at low amperages, they should not be used to monitor solar arrays, as solar produces a large range of amperages, often insufficient for accurate monitoring during morning and evening when there is low production. In addition, inverters have a very low amount of amperage and a high phase offset during standby and may display overnight generation due to the inaccuracy.

2.3 Remote Devices

This section is used to define remote devices whose data should be recorded. Examples of remote devices are other eGauges, PV inverters, or weather stations. If you are not planning on recording data from a remote device, skip this section.

For full information on using remote devices, please visit <http://egauge.net/support/remote-devices>.

For our running example, we will select “remote eGauge via UDP” as the protocol and “1708009” as the device’s address. After entering the address, click the “Done” button. For security-reasons, eGauge may prompt you for authentication before this step can be completed. Assuming proper authorization and successful connection, the protocol and address fields will be grayed out (Should you need to make changes, click the “Edit” button again.) the gray question mark will turn to a green check-mark, indicating the eGauge was able to communicate with the remote device and that it has fetched the list of available registers from that device.

The screenshot shows a configuration form with three main sections: "Device name:", "Protocol:", and "Device address:". The "Device name" field contains "Secondary". The "Protocol" dropdown menu is set to "remote eGauge via UDP". The "Device address" field contains "1708009". To the right of the address field is a green checkmark and an information icon (i). Below the form is an "Add Device" button. An "Edit" button is located to the right of the information icon.

If this step had failed for some reason, there would be a red cross-mark instead of the green check-mark, as shown below:

The screenshot shows the same configuration form as above, but the "Device address" field now has a red cross-mark instead of a green checkmark. The "Add Device" button is still present.

There are several reasons why you might get a red cross-mark after clicking “Done”:

- **Wrong address.** Verify that the address you entered is correct and has the proper format.
- **Wrong protocol.** Verify that you selected a protocol suitable for the remote device.
- **Remote device not working.** Verify that the remote device is up and running and has the requested protocol enabled.
- **Remote device not reachable.** Verify that the remote device can be reached from the eGauge. If the remote device is another eGauge, and both are using HomePlug, you can use the [Settings → HomePlug](#) screen to verify that the remote device can be reached with good signal strength.

For additional information on why it failed, you may click the info button (“i” in a blue circle) to the right of the red cross-mark.

You will only be able to define new registers for remote devices listed with a green check-mark. However, eGauge is careful to never delete already configured registers for remote devices, even if the remote devices may be (temporarily) inaccessible. Removing the remote register entry completely, however, will remove any registers using that remote device.

2.4 Register Configuration

This section defines what data gets recorded in the internal database of the eGauge device. Each register has a unique name that identifies what it is recording. There is a fixed upper limit on the number of registers a device can record. Depending on the generation of the hardware, the limit may be eight, twelve, sixteen or 64 registers. Devices can also be ordered with an optional alternative database, which increases the limit to 64, 128, or 256 registers. To make it easy to keep track of how many registers are in use and how many registers are available in total, the title of this section displays this information in the format “(*N* of *M* in use)”, where *N* is the number of registers in use and *M* is the total number of registers available.

The latest generation of hardware (EG4xxx) by default use a 64-register database.

With the standard database, register data is stored in minute-by-minute granularity. The following 10 years of data is stored in 15-minute granularity, and after that data is condensed to 1-day for the lifetime of the device. All data is retained on the device; the device does not copy the data to the internet.

In addition to the onboard storage, volatile memory stores a recent period of second-by-second or sub-second granular register data. This volatile memory is cleared upon reboot and configuration changes. Volatile memory data may be accessed the same way onboard storage data is (e.g., web interface or API access).

2.4.1 Register Name

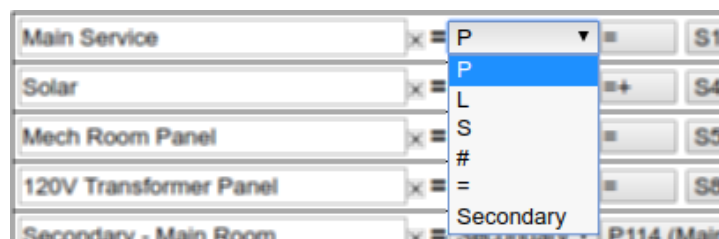
Register names may contain any ASCII/UTF-8 characters except for control characters such as backspace, delete, or newline. Most punctuation marks such as slash, backslash, quotes, etc., are all permissible. There are two exceptions:

- You may not enter a register name that ends with a plus (+), minus (-), vertical-bar (|), or asterisk (*)
- You may not enter a register name that consists entirely of digits

Through UTF-8, it is possible to use symbols from non-English character sets. However, note that non-English symbols encoded in UTF-8 occupy up to four bytes of memory. eGauge limits the length of register names to 31 bytes. With English characters, register-names can therefore be up to 31 characters long. With non-English characters, fewer symbols can be stored in a register-name, with the exact maximum length depending on the chosen symbols. As a rule of thumb, non-ASCII Latin characters (e.g., accented characters such as German Umlauts) consume two bytes per symbol and symbols from ideographic languages such as Chinese consume four bytes per symbol.

2.4.2 Register Type

Each register can be configured to record one of several types of data. The type can be selected from the drop-down menu to the right of a register-name. In our example, the drop-down menu looks as follows:



The entries in this popup-menu have the following meaning:


- P:** Records the **power** calculated from one or more current/voltage-pairs.
 - L:** From a **Line voltage** input, records either *normal value (AC)*, *DC-only (mean) value*, or *frequency* of the chosen Line input. Line input options include the L1, L2, and L3 terminals of the meter, as well as the Ldc 2-pin DC voltage terminal (EG4xxx only). D1, D2, and D3 are special virtual calculations typically used for troubleshooting with delta non-neutral installations.
 - S:** From a **Sensor** input, records either *normal value (AC)*, *DC-only (mean) value*, or *frequency* of the chosen Sensor input.
 - #:** Records numeric data. The only numeric data available at this time is a quantity called “.*excess*”. When there are remote devices, the quantity has a non-zero value only if a remote device was temporarily unavailable and eGauge is catching up with the remote device’s current data. The details of this register are arcane but a non-zero value can be used as an indicator that there were some (temporary) communication problems with remote devices and that the eGauge is now catching up to the current data, which may cause the temporary inflation of up to $\pm 10\%$ of the values read from remote registers.
 - =:** Formula register. Records the value obtained by evaluating the specified formula. See Section 2.4.4 for details.
- remote device name:** Selecting the name of a remote devices (in our example, *Secondary*), allows you to record a register from that device. Before adding a remote register, make sure that there is a green check-mark (✓) indicator to the right of the address field of the remote device. If the indicator is a question-mark (?), click on it to fetch the list of registers available from the remote device. If the indicator is a red cross-mark (✗), the eGauge was unable to communicate with the remote device. See the previous section for information on how to resolve the problem.

In the example of Figure 1, we see nine rows in the `Registers` section: the first four calculate power (P). We will discuss those in more detail in the next section.

The fifth row defines a remote register that records the data from the “P114 (Main Room)” register of remote device *Secondary*.

The sixth row is a register that records the AC current of the CT connected to port 1 (S → S1 → normal value). The seventh records the frequency of the CT connected to port 1 (S → S1 → frequency).

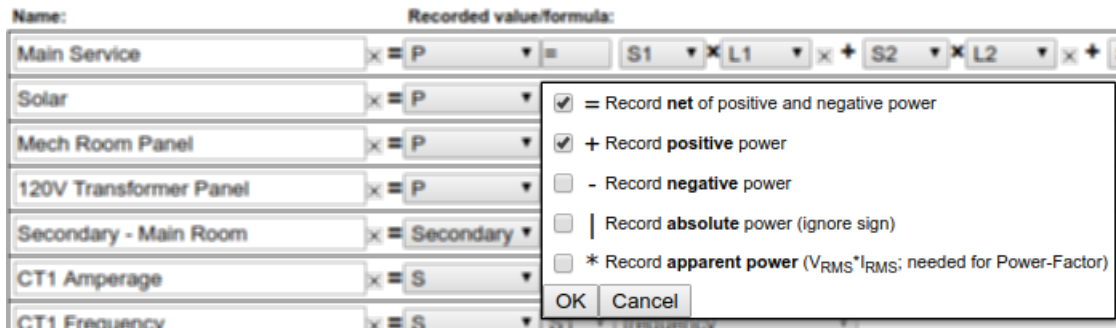
The eighth records the AC voltage of line input 1 (L → L1 → normal value). The last register records the frequency of line of line input 1 (L → L1 → frequency).

A register can be added by clicking on the “Add Register” button. An existing register can be deleted by clicking on the remove icon  to the right of the register name.

2.4.3 Recording Power

There are two parts to configuring a register to record power: first, the subtype(s) to be recorded need to be selected, then the power-components that should be used to calculate the power need to be specified.

Power Subtypes The subtypes to record for a power register are selected by clicking on the subtype-selector button to the right of the register-type button. This will pop up the dialog shown below:



As shown here, there are five subtypes which can be recorded by turning on the corresponding checkbox:

- = **(net)**: Checking this box ensures that the net flow of power is available as a register-value. The value of such a register increases when the calculated real power is positive and decreases when it is negative. This is equivalent to a utility's net-meter, for example.
- + **(positive-only)**: Checking this box ensures that a register-value is available that increases only when the calculated real power is positive. When the real power is negative, the register-value does not change. This can be used, for example, to record only the power generated by an inverter or only the power consumed from the utility.
- **(negative-only)**: Checking this box ensures that a register-value is available that increases only when the calculated real power is negative. When the real power is positive, the register-value does not change. This can be used, for example, to record only the power consumed by an inverter when there is no sun or only the power fed back into the grid. **NOTE:** While the power values being read are negative, the register value *increases*. For example, if a negative-only power register reads $-2000kWh$, and then another $-500kWh$, the register value will be $2500kWh$.
- | **(absolute)**: Checking this box ensures that a register-value is available that counts independent of the sign of the calculated real power. That is, the register-value increases by the absolute value of the real power regardless of whether it is positive or negative. This is not often used. **WARNING:** This calculates the total power of all components in the formula and then takes the absolute value. Given a formula of $CT1 * L1 + CT2 * L2$ where $CT1 * L1$ is $-500kW$ and $CT2 * L2$ is $+300W$, the resulting value will be $200kW$, not $800kW$.
- * **(apparent)**: Checking this box ensures that apparent power is recorded in a register. Apparent power is the sum of the product of the RMS voltage and current of each power-component. As such, apparent power always has a non-negative value. This can be used, for example, to track the quality of the power as the ratio of apparent power and absolute real power is the power-factor.

When you are done selecting subtypes, click on the `OK` button to save the selection or on the `Cancel` button to revert to the previously existing configuration.

Note that the values of the different register subtypes are dependent on each other, except for apparent power. For example, given the positive-only and the net register values, the negative-only register value can be calculated as the difference of the two. To conserve storage space, eGauge automatically exploits these dependencies and only records the minimum data needed. Because of this, eGauge never uses up more than three registers no matter what combination of subtypes are selected.

For example, the positive-only (+) register can be used to accurately calculate total power usage at a site with a photo-voltaic (PV) solar system. It can also be used to accurately track energy-flow of a utility meter, which can be helpful for cases where a utility charges a different rate for consumed power than it pays for power fed back into the grid. Using the absolute (|) subtype can simplify CT installation for, e.g., pure loads. By ignoring the sign of the calculated power, the measured power is always positive, no matter which way the CT is installed. Finally, the apparent power (*) subtype enables power-factor tracking as the ratio between apparent power and the absolute value of real power is the power-factor.

Registers other than net power automatically have a suffix the same as their power sub-type symbol appended to the register name. For example, a negative-only power register named “Returned Power” will be saved as “Returned Power-”, a positive-only register named “Demand Power” will be saved as “Demand Power+”, and a register named “Absolute Power” will be saved as “Absolute Power|”. The suffix is not displayed in the Installation Settings page where the register name is set.

Power Components The power components define which currents and voltages are to be combined to calculate a register’s power figure. In our example, the power components for the `Main Service` and `Solar` registers look as follows:

| Name: | Recorded value/formula: |
|--------------|--|
| Main Service | $S1 \times L1 + S2 \times L2 + S3 \times L3$ |
| Solar | $S4 \times L1 - S4 \times L2$ |


This shows that `Main Service` power is calculated as the product of the current measured by CT1 (S1) and the voltage measured on line L1 (e.g., black leg) plus the current measured by CT2 (S2) and the voltage measured on line L2 (e.g., red leg), plus the current measured by CT3 (S3) and the voltage measured on line L3 (e.g., blue leg).¹

In other words, the power coming from the three legs of the three-phase power feed from the utility are combined into a single register. It is possible to record the three legs in separate registers, but that would use up more of the available storage space and would add a level of detail that most users do not care about.

It is illustrative to look at the power-components for the `Solar` register since it uses $-S4$ in the second component. Naively, a 240Vac solar inverter would have to be measured by two CTs: one per leg. However, provided there is no current flowing on the Neutral wire (usually the case for a single-phase inverter), we know that at any given point in time, the current flowing out of leg 1 of the inverter has to flow back into leg 2 and vice versa (the flow of current reverses 60 times a second assuming a 60Hz power system). Thus, instead of measuring the second leg of the inverter with another CT, we can simply use the negated value of $S4$, i.e., $-S4$ to accurately calculate the power delivered from the inverter. This saves the cost of a CT and the time that would be required to install it. Note that the minus-sign in $-S4$ should not be taken too literally in the formula: it causes the polarity of the component to be reversed for real power calculations, but for apparent power calculations, the sign has no effect, since the RMS value of the current measured by $S4$ and $-S4$ is by definition always positive.







WARNING: Only pure single-phase loads may utilize this method. If a neutral is used, current may be asymmetrical and this calculation will yield invalid results. It is safest to use 2 separate CTs on any single-phase load. 3-phase loads always require 3 CTs (aside from certain deltas without a neutral, see diagrams in section 3 for examples).

¹Prior to firmware version 4.0, only line inputs are available for use as the secondary component in power calculations. Firmware v4.0 and newer make available all sensor inputs as the secondary component. This is to support voltage readings from supported sensors, such as the eGauge EV1000.

To add a new power-component to a register, click on the “Add Component” button. To delete a component, click on the remove icon  to the right of the component.

2.4.4 Formula Registers

Formula registers are a powerful tool to record quantities that can be derived from other, directly-measured registers. For example, suppose register `Main Service` measures the real power used at a site, then register `Main Service*` would measure the apparent power used at the same site. To record the power-factor for the site, we would have to divide the absolute value of `Main Service` by `Main Service*`. This can be accomplished by defining a formula register as shown below:




| Name: | Recorded value/formula: | |
|-------------------|---|---|
| Main Service |  | P  S1 \times L1  S2 \times L2  S3 \times L |
| Main Power Factor |  | $=$  Number with 3 decimals \downarrow <code>abs(\$"Main Service")/\$"Main Service"</code> |
| Add Register | | Angle [°] Number with 3 decimals |

In this example, we named the register “Main Power Factor” and selected the equal-sign (formula-register) as the register type (see Section 2.4.2). Next, we selected the type of physical quantity that is being recorded by the register. As described in more detail in Appendix B, various choices are available here: power, voltage, irradiance, and so on. In our case, power-factor is a unit-less number between 0 and 1, so we selected `Number with 3 decimals` from the drop-down box. Finally, we entered a formula of `abs($"Main Service")/$"Main Service"` for calculating the value of the register.

Here, `$"regname"` is used to obtain the instantaneous (second-by-second) value of register `regname`, so `$"Main Service"` will evaluate to the instantaneous site-usage (in Watts) and `$"Main Service*"` evaluates to the apparent power used by the site. As you might expect, the `abs()` function calculates the absolute value of the value passed to it (i.e., it returns the passed value after dropping its sign). Dividing the value returned by the `abs()` function by the apparent power yields a number between 0 and 1 (the power-factor).

A list of functions that are available for use in calculating the value of a formula register can be found in Appendix A. An online-version of this function-list can also be found in the `Help` section of the eGauge web-pages. Apart from basic mathematical functions such as square-root, exponential, or min and max functions, there are some time-related functions available to read the time-of-day, month of the year, and the weekday. Also, there are some convenience-functions available to calculate, for example, expected output from a solar PV array or to calculate reactive power given measured real and apparent power.

Below are two more examples of useful applications of formula registers.

| Name: | Recorded value/formula: | |
|-----------------|---|---|
| Solar Predicted |  | $=$  power [W] \downarrow <code>P_PV(\$"Irradiance", \$"Panel Temp", 27*225, -0.38)</code> |
| Heating Degrees |  | $=$  temp. [C] \downarrow <code>max(0, 18.333-\$"Ambient Temp")</code> |

The first (`Solar (predicted)`) shows how to calculate and record the predicted power of a solar PV array given solar irradiance (`$"Irradiance"`), the panel temperature (`$"Panel Temp"`), the rated power of the PV array (27 panels of 225W each), and the temperature-coefficient of the panels (-0.38%). Note that irradiance and panel temperature would have to be measured by a weather-station that is compatible with


eGauge, such as the IMT Weatherstation. Note that predicted PV-output is a physical quantity measured in Watts, so we selected “power [W]” as the physical quantity for this register.

The second example (Heating Degrees) shows how to record heating degrees given a register called “Ambient Temp,” which records the ambient (outside) temperature at a site. Heating degrees is defined to be the average temperature by which a room would have to be heated to keep it at a comfortable temperature of 65° Fahrenheit (18.333° Celsius). If the outside temperature is above 65F (18.333C), no heating is required and the formula is defined to return 0 in that case (the maximum value of 0 and a negative number is 0). If the ambient temperature is any lower than that, the formula returns the difference as 18.333 minus a number smaller than that will be a positive number. Note that we selected “temp. [Celsius]” as the physical quantity for this register, as heating degrees is a temperature. This also illustrates the point that formula registers always record physical quantities using the metric system. Any conversions to user-preferred units have to happen when the recorded values are being presented to the user, e.g., through the built-in web-pages.


Several caveats apply to using formula registers:

- You might wonder what would happen if a formula attempted to use the value of another formula register. The answer is that this is an undefined operation. It could return the current value of that register, the previous second’s value of the register, or any other value. Thus, formulas should refer only to measured (physical) registers.
- Calculating formulas takes time. The more complicated a formula is, the more time it takes to calculate its value. In extreme cases, it might be possible to overload a device by defining lots of complicated calculations, especially if the device is already heavily loaded, e.g., due to lots of power calculations. If this were to happen, the device-response could become sluggish, even to the point where it might seem unresponsive.
- All calculations are performed using single-precision IEEE-754 floating-point numbers. The final value of a formula register is then rounded to a 64-bit signed integer number.

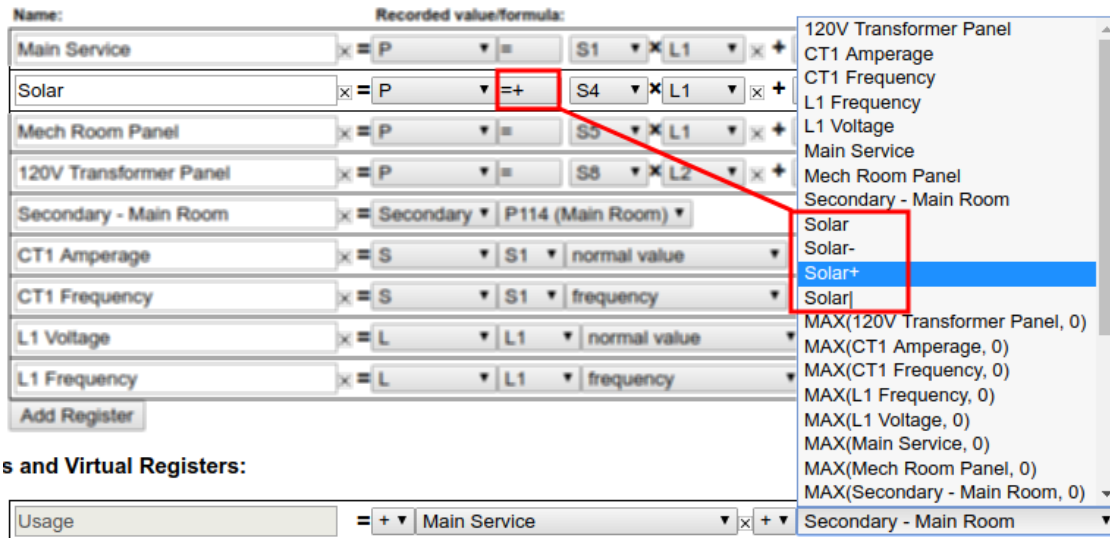
2.5 Totals and Virtual Registers Configuration

Virtual registers are calculated as a sum or difference of real (physical) registers. As such, virtual registers do not consume any storage space and there is no a priori limit on how many of them you can define. A virtual register can be added by clicking on the “Add Virtual Register” button. To remove one, click on the remove icon  to the right of the register name. In our example, this section looks as follows:

| | | | | |
|--------------------------|----------------------------------|-------------------------------|---|--------------|
| Usage | = + ▾ Main Service ▾ | x + ▾ Secondary - Main Room ▾ | x | Add Register |
| Generation | = + ▾ Solar ▾ | x | | Add Register |
| Battery | = | | | Add Register |
| Battery left | = | | | Add Register |
| 120V Loads and Main Room | x = + ▾ 120V Transformer Panel ▾ | x + ▾ Secondary - Main Room ▾ | x | Add Register |
| Add Virtual Register | | | | |

For each virtual register, you can define an arbitrary number of registers which are added or subtracted together when calculating the virtual register’s value. A register is added by clicking on the Add Register button. A register can be removed by clicking on the remove icon  to the right of the register drop-down menu. You can select whether there register should be added to or subtracted from the virtual registers value by selecting + or – from the drop-down menu to the left of the register name, respectively. The

register to be added or subtracted can also be selected from a drop-down menu. In our example, this menu looks as follows:



Each power register will show up in several variations in the menu, namely with one variant for each available subtype. If you look back at the register configuration for `Main Service`, only the net (=) subtype is being recorded, so it shows up only once as `Main Service` (since net metering is the default for eGauge, the equal-sign is omitted from the name, unlike for any other subtype). However, for `Solar`, both net (=) and positive-only (+) subtypes are being recorded. Because of this, all subtype variants except for apparent power appear in the menu: `Solar` (net), `Solar+` (positive-only), `Solar-` (negative-only), and `Solar|` (absolute). The last two are available because they can be calculated from net and positive-only.

2.5.1 Totals

There are two special virtual registers called `Usage` and `Generation`. These are called *totals* and their value is used to represent *total site usage* and *total site generation*. In the dashboard graph, total usage is represented by a thick red line, and total generation is represented as a thick green line. The totals are always present (cannot be deleted), but if a site does not measure usage or if there is no site-local power generation, the respective total can be left blank.

Two more special virtual registers have been added in firmware v4.0 called `Battery` and `Battery left`. These are intended to be filled by remote device data from a battery backup system, and used with the Modern Interface Dashboard introduced in firmware v4.5.

2.5.2 Defining Views

Views perform a few different functions depending on whether it is used in the Classic or Modern interface. The Classic interface is being deprecated in favor of the Modern interface. See <http://egauge.net/support/modern> for more information.

Classic Interface

1. In the Classic interface, a view creates a new chart layout that displays only the registers defined for that view. Views may be changed by clicking the upside-down triangle in the upper left-hand corner of the Classic interface chart. For example, if a meter has a high number of registers, different views can contain only a subset of these registers which can avoid clutter. For example, one view may contain only amperage registers, another view only temperature and environmental readings, addend another view containing only the power readings.
2. If the meter has a site-wide password enabled, a non-administrative user can be “locked” to that particular view’s chart layout by naming the view the same as the username, and setting that user’s permissions to “Restricted to username view”. The restricted user will not have access to any registers not defined in a virtual register for their view.

Modern Interface

Similar to the second function in the Classic Interface mentioned above, a restricted user using the Modern interface will have only access to the registers defined in their view. They may modify their chart and dashboard locally, but only with access to the registers defined in their view.

Example and Configuration

For example, an apartment manager may be monitoring multiple units with a single eGauge meter and wishes to grant access to individual tenants, but only for their own units consumption information. Let’s say we have tenants Dave, Sam, and Alex in units 203, 204, and 205 respectively and would like to have each tenant have a log-in to view their particular usage, as well as line voltages.

The register configuration may look like this:

| Name: | Recorded value/formula: |
|------------|---|
| Unit 203 | <input type="checkbox"/> = P <input type="checkbox"/> = <input type="checkbox"/> S1 <input type="checkbox"/> L1 <input type="checkbox"/> + <input type="checkbox"/> S2 <input type="checkbox"/> L2 <input type="checkbox"/> |
| Unit 204 | <input type="checkbox"/> = P <input type="checkbox"/> = <input type="checkbox"/> S3 <input type="checkbox"/> L3 <input type="checkbox"/> + <input type="checkbox"/> S4 <input type="checkbox"/> L1 <input type="checkbox"/> |
| Unit 205 | <input type="checkbox"/> = P <input type="checkbox"/> = <input type="checkbox"/> S5 <input type="checkbox"/> L2 <input type="checkbox"/> + <input type="checkbox"/> S6 <input type="checkbox"/> L3 <input type="checkbox"/> |
| Voltage L1 | <input type="checkbox"/> = L <input type="checkbox"/> L1 <input type="checkbox"/> normal value <input type="checkbox"/> |
| Voltage L2 | <input type="checkbox"/> = L <input type="checkbox"/> L2 <input type="checkbox"/> normal value <input type="checkbox"/> |
| Voltage L3 | <input type="checkbox"/> = L <input type="checkbox"/> L3 <input type="checkbox"/> normal value <input type="checkbox"/> |

To create views for these tenants, a virtual register whose name is created in the format of *viewname.registername* where *viewname* is the name of the view (which must be the same as the username that should have access to this view), and *registername* is the register name that will be displayed for that view.

For our example, the virtual registers may look like this:

| | | | | | | | | | | |
|----------------|---|---|----------|------------|---|----------|---|---|----------|---|
| Usage | = | + | Unit 203 | x | + | Unit 204 | x | + | Unit 205 | x |
| dave.Usage | x | = | + | Unit 203 | x | | | | | |
| dave.Voltage 1 | x | = | + | Voltage L1 | x | | | | | |
| dave.Voltage 2 | x | = | + | Voltage L2 | x | | | | | |
| sam.Usage | x | = | + | Unit 204 | x | | | | | |
| sam.Voltage 1 | x | = | + | Voltage L3 | x | | | | | |
| sam.Voltage 2 | x | = | + | Voltage L1 | x | | | | | |
| alex.Usage | x | = | + | Unit 205 | x | | | | | |
| alex.Voltage 1 | x | = | + | Voltage L2 | x | | | | | |
| alex.Voltage 2 | x | = | + | Voltage L3 | x | | | | | |

Note, a single register such as 'Voltage L1' may be used on multiple views. As with all virtual registers, additional physical registers may be added to a virtual register.

The user configuration section should contain users 'dave', 'sam', and 'alex' all with privileges "Restricted to username view". At least one other user with save privileges must exist on the meter:

| | |
|---|--|
| Password-protect entire site ? <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Username ? owner dave sam alex | Privileges ? Allowed to view all data and change settings from anywhere Restricted to username view Restricted to username view Restricted to username view |

Now, when any of the restricted users log in, they will see a chart with only the registers defined in their view. The 'owner' account will be able to see all registers, as well as be able to display the individual views through the chart menu.

3 Configuration Examples

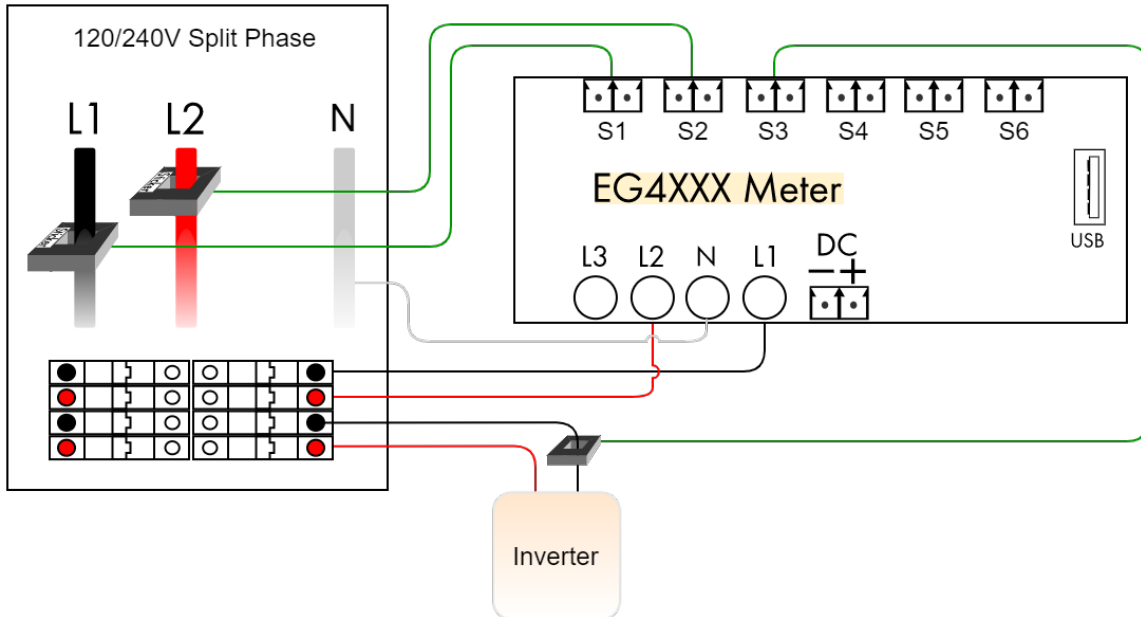
The following pages illustrate various common installation scenarios and matching configurations.

- Pay attention to phasing: CTs must be configured for the correct phase they are on. E.g., if a breaker feeding the eGauge L1 is on the black phase, CTs around any black phase must be applied to L1. The first available breaker slot may not be the actual panel L1 and may cause an offset between the eGauge phasing and panel phasing. Visit <http://egauge.net/support/phasing> for more information on phasing and how to validate consistent phasing.
- CT directionality. CTs may face one of two directions, and will give a positive or negative power value depending on which direction the power flows. Positive or negative values are arbitrary, but care must be taken to orient all CTs consistently.
Bidirectional power will result in the opposite power polarity when the power direction changes.
See section 2.2.1 for more information on CT orientation.
- EG4xxx meter diagrams and other hardware are simplified for clarity.
- Ground wiring is not shown (note: eGauge meters and equipment do not have a ground connection).
- Most examples are partial configurations. For a complete system, the configurations from various examples may have to be combined.
- All Sensor inputs are Current Transformer inputs, so “CT1” and “S1” may be used interchangeably, similarly “CT3” and “S3” mean the same thing.

Please review the Owner’s Manual for detailed installation requirements. For safety, the eGauge device always must be wired to a breaker and the breaker should be labeled “eGauge Disconnect.”

3.1 Standard Split-Phase back-fed

Basic installation measuring power coming from a power utility (grid) and from a single-phase solar-system inverter.



Registers

$$\begin{aligned} \text{Grid} &= [\text{P}] [=] \text{S1} \cdot \text{L1} + \text{S2} \cdot \text{L2} \\ \text{Solar} &= [\text{P}] [=+] \text{S3} \cdot \text{L1} + -\text{S3} \cdot \text{L2} \end{aligned}$$

Totals and Virtual Registers

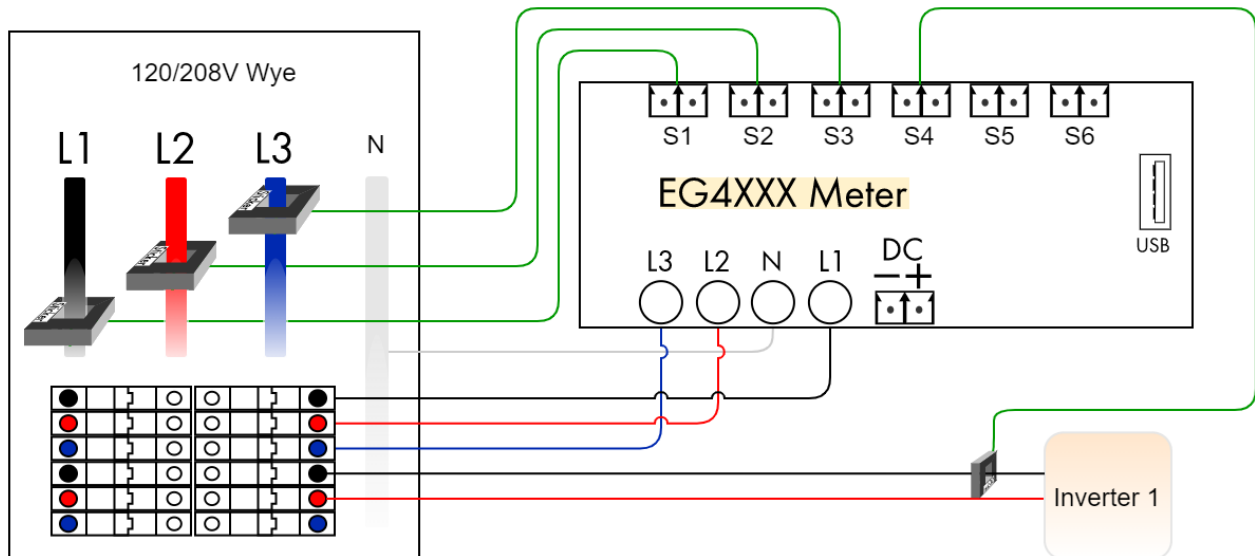
$$\begin{aligned} \text{Usage} &= \text{Grid} + \text{Solar+} \\ \text{Generation} &= \text{Solar} \end{aligned}$$

Notes:

- With a single-phase inverter, the current flowing at any given instant on leg L2 is the negative of the current flowing on leg L1. Hence, rather than using 2 CTs on the inverter, simply multiply the negated value of CT3 with L2 to calculate the power on that leg.
- The totaling rules indicate that total usage (consumption) is calculated as the sum of the power reported for register `Grid` and, the positive-only register `Solar+`. If the power reported for `Solar` is negative (indicating that the inverter is consuming power, e.g., during the night), then total usage is simply equal to the power reported for register `Grid`. It would be (slightly) wrong to define the `Usage` totaling rule as `Grid + Solar` because in this case, the inverter's consumption would be canceled out of the usage, giving a lower than real consumption figure.

3.3 Three-Phase 120/208 or 277/480 Wye w/ single-phase inverter

Standard three-phase installation measuring power coming from a power utility (grid) and from a single-phase solar-system inverter. The color coding shows 120/208V, but applies to 277/480V as well. This diagram is for a Wye system with a neutral. Refer to delta diagrams for systems without neutral.



Registers

```
Grid    = [P] [= ] S1*L1 + S2*L2 + S3*L3
Solar   = [P] [=+] S4*L1 + -S4*L2
```

Totals and Virtual Registers

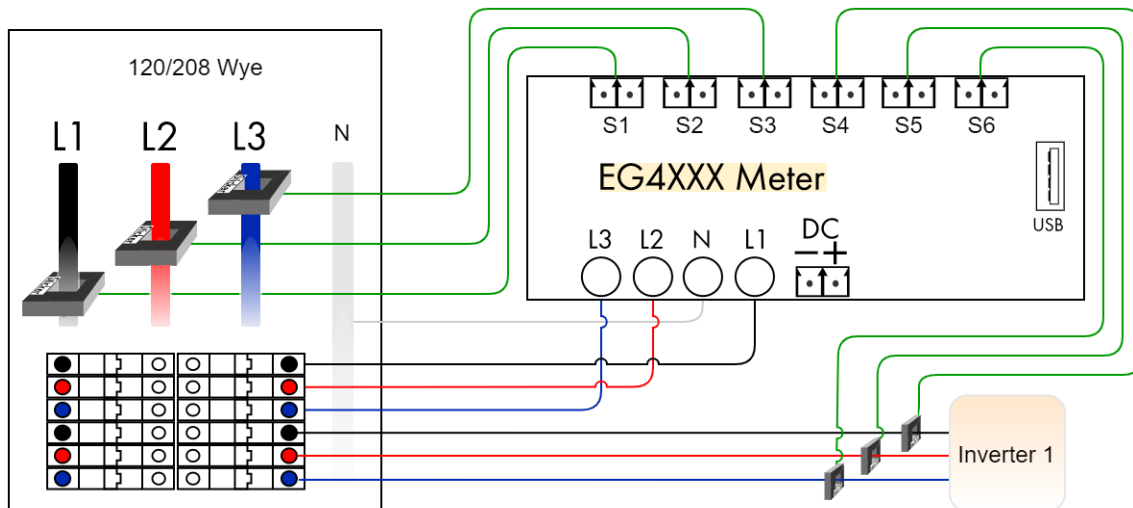
```
Usage      = Grid + Solar+
Generation = Solar
```

Notes:

- Three-phase installations are set up the same as split-phase systems, except that a third voltage-tap (L3) and a third Grid CT is required to measure power flow on phase 3.
- With multiple inverters, add one Solar CT per inverter and define a separate register for each inverter (e.g., `Solar 1` and `Solar 2`). Adjust the register definition according to the CT that is measuring the current and the phases that the inverter feeds onto. For example, CT5 measuring current onto L2 and also feeding onto L3: `Solar 2 = S5*L2 + -S5*L3`.
- If there are more than three single-phase inverters, it is more economical to measure the total solar output with one CT per phase.
- If the solar is a line-side tap rather than back-fed, the `Usage` register should use `Solar-` rather than `Solar+`. See configuration example 3.2, Direct-Feed Solar, for more information.

3.4 3-phase inverter

Standard three-phase installation measuring power coming from a power utility (grid) and from a three-phase solar-system inverter. The color coding shows 120/208V, but applies to 277/480V as well. This diagram is for a Wye system with a neutral. Refer to delta diagrams for systems without neutral.



Registers

```
Grid    = [P] [= ] S1*L1 + S2*L2 + S3*L3
Solar   = [P] [=+] S4*L1 + S5*L2 + S6*L3
```

Totals and Virtual Registers

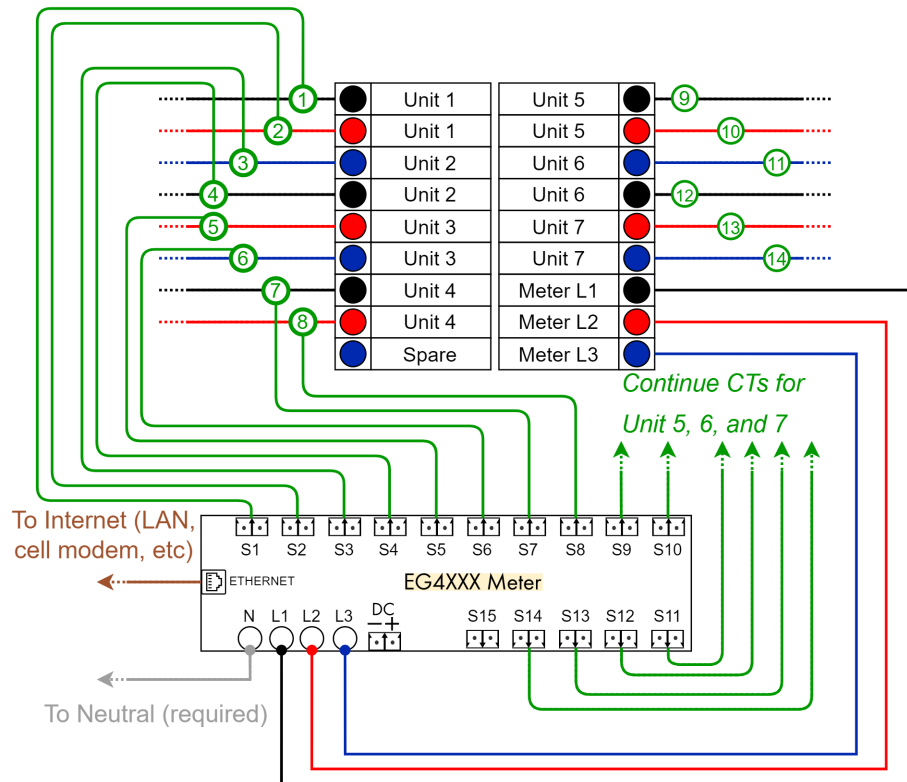
```
Usage      = Grid + Solar+
Generation = Solar
```

Notes:

- Three-phase installations are set up the same as split-phase systems, except that a third voltage-tap (L3) and a third Grid CT is required to measure power flow on phase 3.
- Three-phase inverters require a CT on each phase. Sometimes during low production they will not output evenly on all 3 legs, and during standby they do not often draw the same amount of current on each leg. Further, in standby, they may output power on certain legs and draw power on others, which can lead to seeing erroneous overnight generation if a single CT is used.
- If the solar is a line-side tap rather than back-fed, the Usage register should use Solar- rather than Solar+. See configuration example 3.2, Direct-Feed Solar, for more information.

3.5 Submetering single-phase units on a 120/208V service

The EG4xxx series meters are capable of monitoring up to 7 single-phase units (EG4015) or 15 single-phase units (EG4030) with a single meter. Single-phase units require 2 CTs each, one for each phase.



Registers

- Unit 1 = [P] [=] S1*L1 + S2*L2
- Unit 2 = [P] [=] S3*L3 + S4*L1
- Unit 3 = [P] [=] S5*L2 + S6*L3
- Unit 4 = [P] [=] S7*L1 + S8*L2
- Unit 5 = [P] [=] S9*L1 + S10*L2
- Unit 6 = [P] [=] S11*L3 + S12*L1
- Unit 7 = [P] [=] S13*L2 + S14*L3

Totals and Virtual Registers

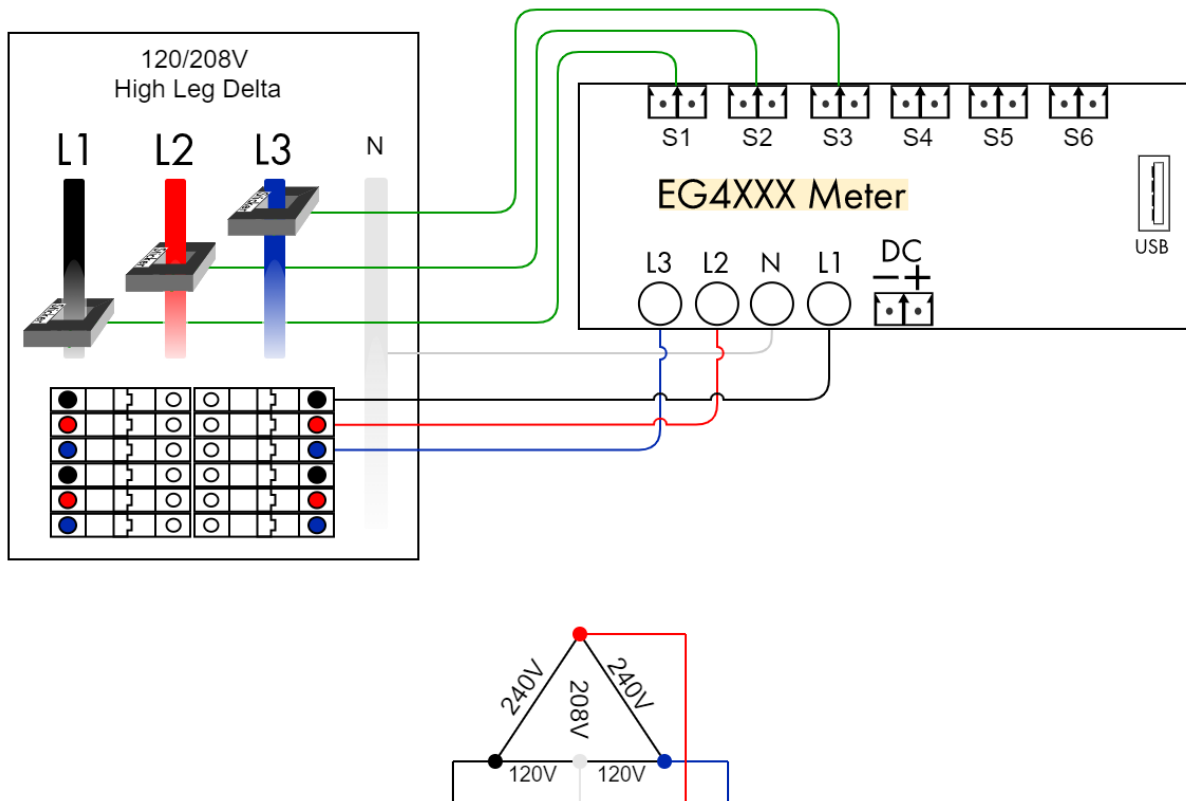
$$\text{Usage} = \text{Unit 1} + \text{Unit 2} + \text{Unit 3} + \text{Unit 4} + \text{Unit 5} + \text{Unit 6} + \text{Unit 7}$$

Notes:

- While the phasing pattern for units is typically staggered as L1 and L2, then L3 and L1, then L2 and L3, and so on, in this example it changes with Unit 5 because the breakers for Unit 5, 6, and 7 begin on the right-hand side of the panel.

3.6 Three-Phase High-leg Delta

Three-phase high-leg delta installation measuring power coming from a power utility (grid).



Registers

$$\text{Grid} = [\text{P}] [=] \text{S1} \cdot \text{L1} + \text{S2} \cdot \text{L2} + \text{S3} \cdot \text{L3}$$

Totals and Virtual Registers

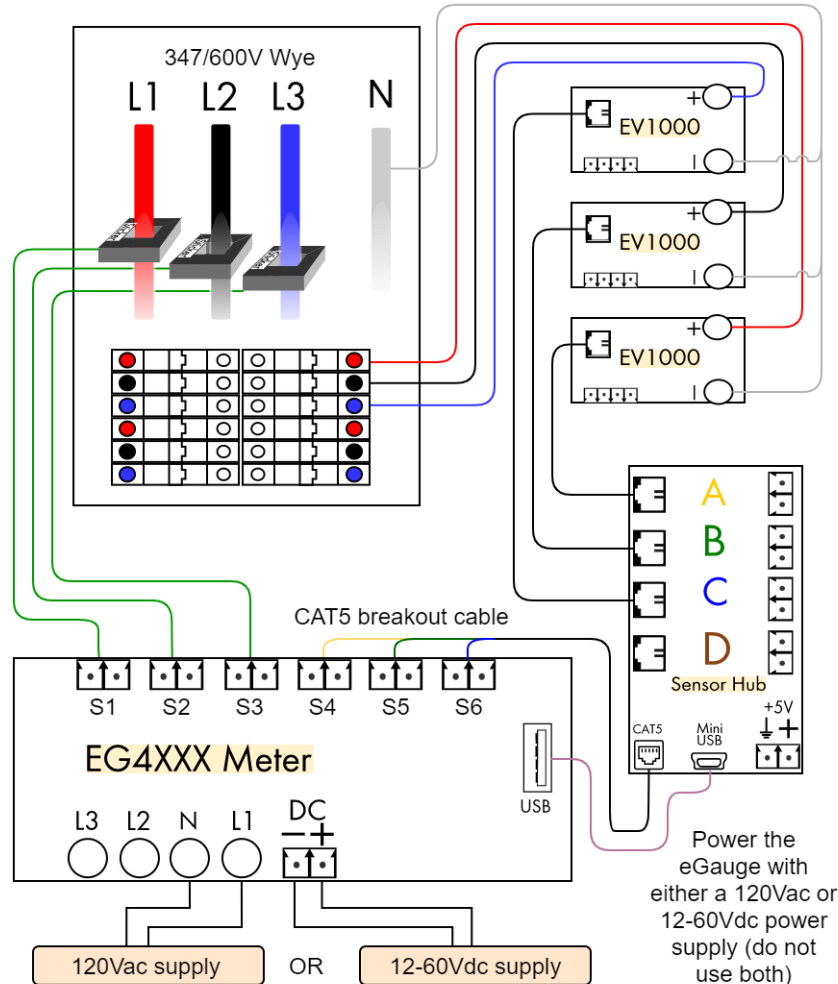
$$\text{Usage} = \text{Grid}$$

Notes:

- Wiring and configuration is identical to a standard three-phase site.
- It does not matter whether the high-leg is wired to L1, L2, or L3, as long as all three phases are measured and line-to-line voltage does not exceed 480V (or line-to-neutral over 277V).

3.7 Three-phase 347/600V Wye

Three-phase 347/600V Wye installation measuring power coming from a power utility (grid).



Registers

$$\text{Grid} = [\text{P}] [=] \text{S1} * \text{S4} + \text{S2} * \text{S5} + \text{S3} * \text{S6}$$

Totals and Virtual Registers

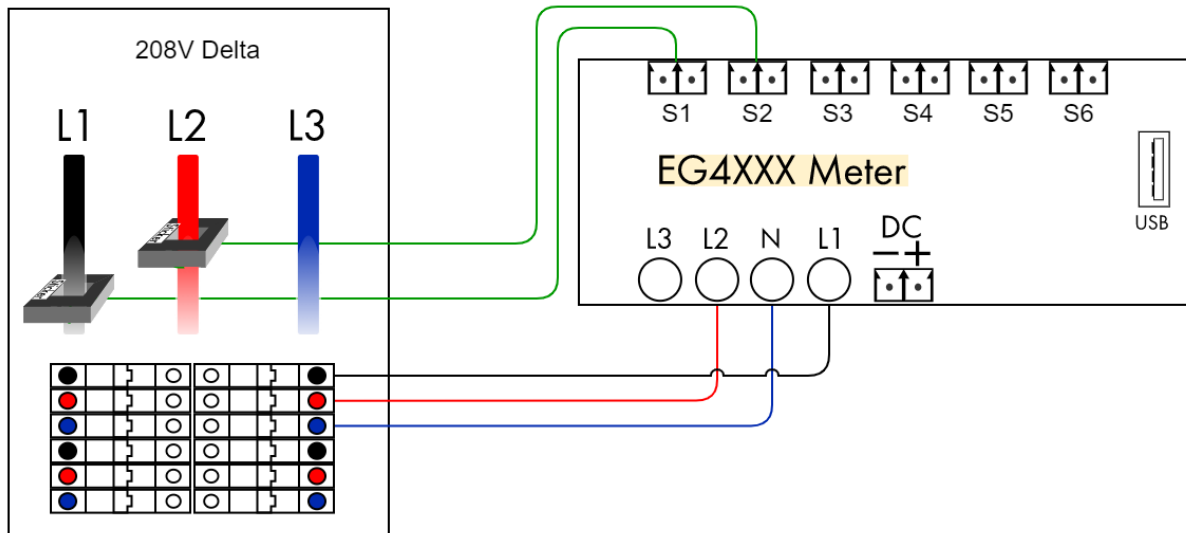
$$\text{Usage} = \text{Grid}$$

Notes:

- Requires the eGauge Sensor Hub, 3x EV1000 voltage sensors and an EG4xxx series meter.
- Please visit <http://egauge.net/support/m/ev1000> for more information on using the EV1000 high voltage sensors sensors and Sensor Hub.

3.8 Three-Phase Delta (L-L voltages $\leq 277V$)

Three-phase delta installation measuring power coming from a power utility (grid). The color diagram below shows a 208V delta, but this also applies to a 120V delta or 240V delta.



Registers

$$\text{Grid} = [P] [=] S1 * L1 + S2 * L2$$

Totals and Virtual Registers

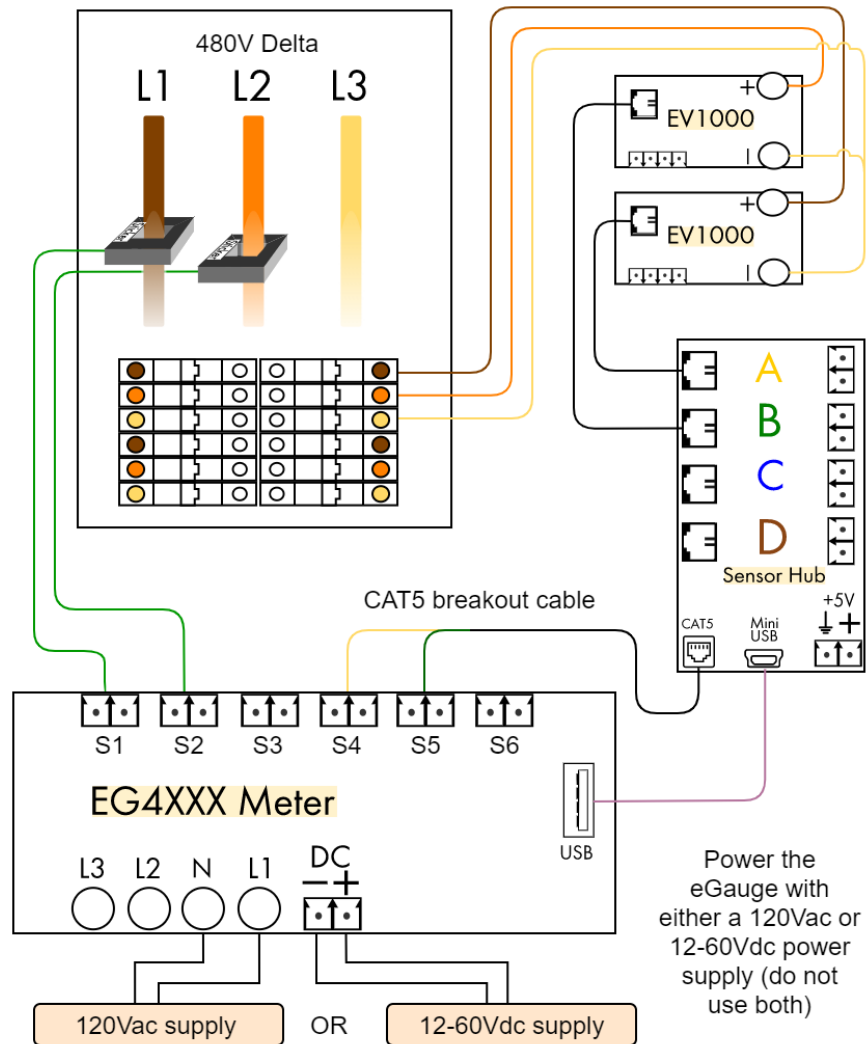
$$\text{Usage} = \text{Grid}$$

Notes:

- **CAUTION:** Since there is no neutral and none of the phases may be grounded, all conductive/metallic parts, including CT-plug screws and CT wiring should be considered live and hazardous!
- If one of the phases is grounded, wire that phase to the Neutral (N) pin on the eGauge.
- The eGauge inputs can measure up to 277V L-N terminal. A 600V delta must be measured the same way as a 480V Delta.

3.9 Three-Phase Delta (L-L voltages $\geq 277V$)

High-voltage (480V) three-phase delta installation measuring power coming from a power utility (grid). This same method may apply to a 600V delta.



Registers

$$\text{Grid} = [P] [=] S1*S4 + S2*S5$$

Totals and Virtual Registers

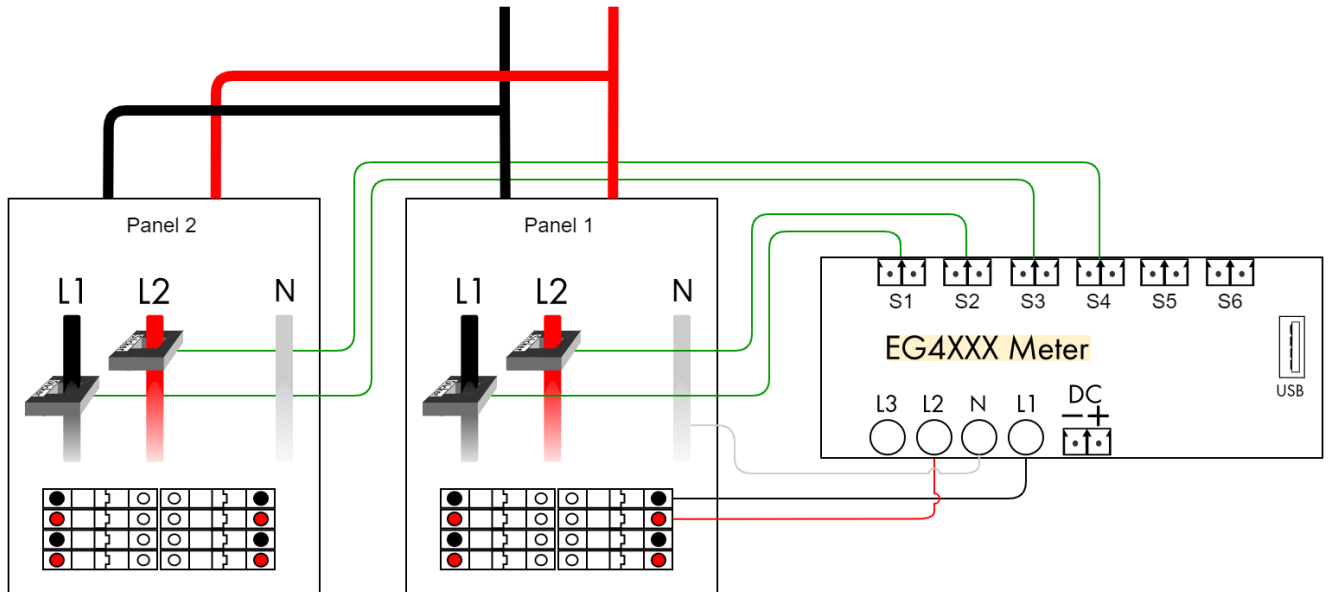
$$\text{Usage} = \text{Grid}$$

Notes:

- **CAUTION:** Since there is no neutral and none of the phases may be grounded, all conductive/metallic parts, including CT-plug screws and CT wiring should be considered live and hazardous!
- Please visit <http://egauge.net/support/m/ev1000> for more information on using the EV1000 high voltage sensors and Sensor Hub.

3.10 Subpanels

This example illustrates how to measure subpanels.



Registers

$$\begin{aligned} \text{Subpanel 1} &= [\text{P}] [=] \text{S1} \cdot \text{L1} + \text{S2} \cdot \text{L2} \\ \text{Subpanel 2} &= [\text{P}] [=] \text{S3} \cdot \text{L1} + \text{S4} \cdot \text{L2} \end{aligned}$$

Totals and Virtual Registers

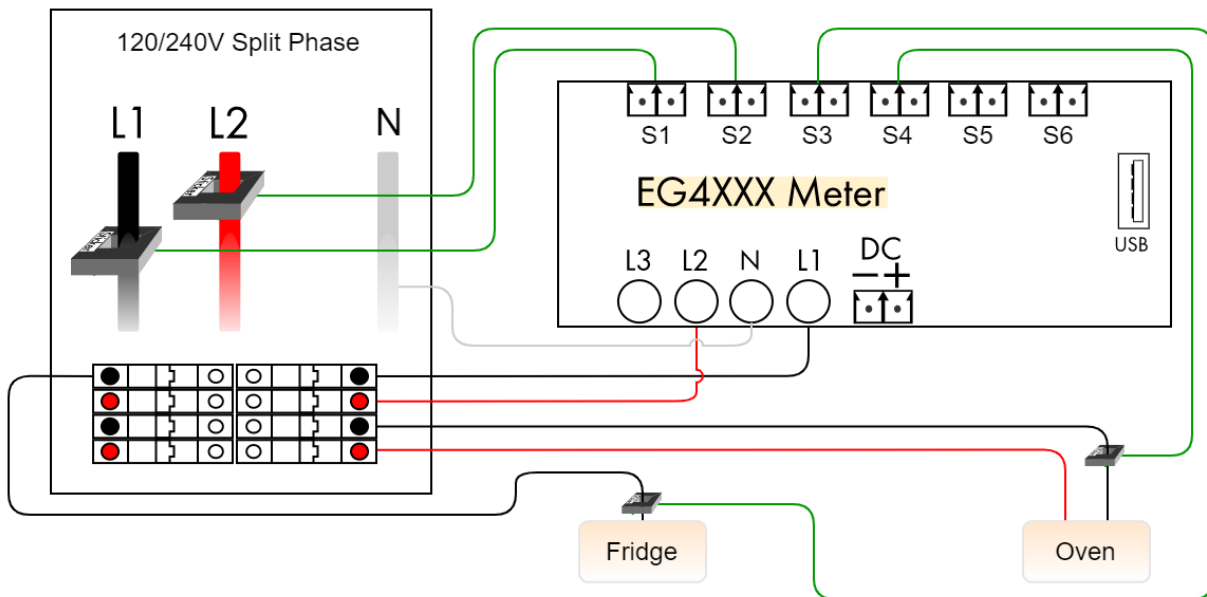
$$\text{Usage} = + \text{Subpanel 1} + \text{Subpanel 2}$$

Notes:

- It does not matter which subpanel the meter obtains voltage from, but care must be taken to identify correct phasing. It is possible for phases to be reversed or inconsistent between panels.

3.11 Appliances

This example illustrates how to configure 120Vac (single pole) and 240Vac (two pole) appliances.



Registers

```

Grid    = [P] [= ] S1*L1 + S2*L2
Fridge  = [P] [= ] S3*L1
Oven    = [P] [= ] S4*L1 + -S4*L2
  
```

Totals and Virtual Registers

Notes:

- Registers defined for appliances do not appear in the totaling rules. Just defining the registers will make them appear in the appropriate places (e.g., in the legend of the dashboard graph).
- A single CT is sufficient both for 120Vac and pure 240Vac appliances. 240Vac appliances with asymmetric currents on the two legs (i.e., with a non-zero current on Neutral) require separate CTs per leg/phase.

4 Troubleshooting

If an installation does not work as expected, we recommend using the channel checker tool to troubleshoot the problem. This tool is available through “Tools” link in the top right menu of the eGauge web pages then click on “Channel Checker” in the left menu. Figure 3 provides an example of what this tool looks like. Note, the two channel checker tables are displayed vertically one next to the other in the device interface. It is displayed horizontally on this page for readability.

As of Thu 25 Jan 2018 09:54:32am:

| Channel | AC+DC (RMS) | AC (RMS) | DC (Mean) | Frequency |
|---------|-------------|-----------|-----------|-----------|
| L1 | 121.570 V | 121.570 V | -0.045 V | 60.01 Hz |
| L2 | 121.650 V | 121.650 V | -0.204 V | 60.01 Hz |
| CT1 | 3.470 A | 3.470 A | -0.047 A | 60.08 Hz |
| CT2 | 7.407 A | 7.407 A | -0.048 A | 60.07 Hz |
| CT3 | 0.129 A | 0.128 A | -0.019 A | 275.24 Hz |
| CT4 | 7.774 A | 7.774 A | -0.013 A | 60.01 Hz |
| CT5 | 0.500 A | 0.496 A | -0.061 A | 321.91 Hz |
| CT6 | 0.236 A | 0.235 A | -0.026 A | 322.52 Hz |
| CT7 | 0.640 A | 0.640 A | -0.017 A | 60.02 Hz |
| CT8 | 0.243 A | 0.243 A | -0.012 A | 111.06 Hz |
| CT9 | 0.594 A | 0.591 A | -0.062 A | 229.26 Hz |
| CT10 | 3.764 A | 3.764 A | -0.040 A | 60.00 Hz |

| Register Name | Value | Power Factor |
|---------------------------------|-----------|--------------|
| Grid (L1*CT1) | 336.40 W | 0.797 |
| Grid (L2*CT2) | 870.10 W | 0.966 |
| Solar SPR (L1*CT3) | -2.60 W | 0.166 |
| Solar SPR (L2*CT3) | -2.60 W | 0.166 |
| Solar Tigo+SPR (L1*CT4) | 921.20 W | 0.975 |
| Solar Tigo+SPR (L2*CT4) | 922.40 W | 0.975 |
| EV Charging (L1*CT5) | -0.30 W | 0.005 |
| EV Charging (L2*CT5) | -0.30 W | 0.005 |
| Oven (L1*CT6) | -0.60 W | 0.021 |
| Oven (L2*CT6) | -0.60 W | 0.021 |
| Comp Closet #1 (L1*CT7) | -56.30 W | 0.724 |
| Garage+Guest Bath+PHEV (L2*CT8) | -18.50 W | 0.626 |
| Cooktop+Mech Room (L1*CT9) | -28.30 W | 0.392 |
| Cooktop+Mech Room (L2*CT10) | -393.60 W | 0.860 |

Figure 3: Example of eGauge Channel Checker Tool

The channel checker reports the real-time voltages measured for each active voltage input and currents for each active CT. If a voltage or a CT is not used by any register, it is deactivated and not used in any register configuration.

RMS (AC) and mean (DC) values for both CTs and voltage inputs are calculated. Frequencies for each

channel input are also calculated and displayed here.

In addition to the currents and voltages, the page also reports power figure for each component of all registers. The power figure is accompanied by a power-factor, which is a value between 0 and 1. A value of 1 indicates a purely resistive load whereas a value of 0 indicates either a purely inductive or purely capacitive load. An unexpected power factor may be caused by a phase mixup, in which a CT amperage is applied to an incorrect phase it is not on.

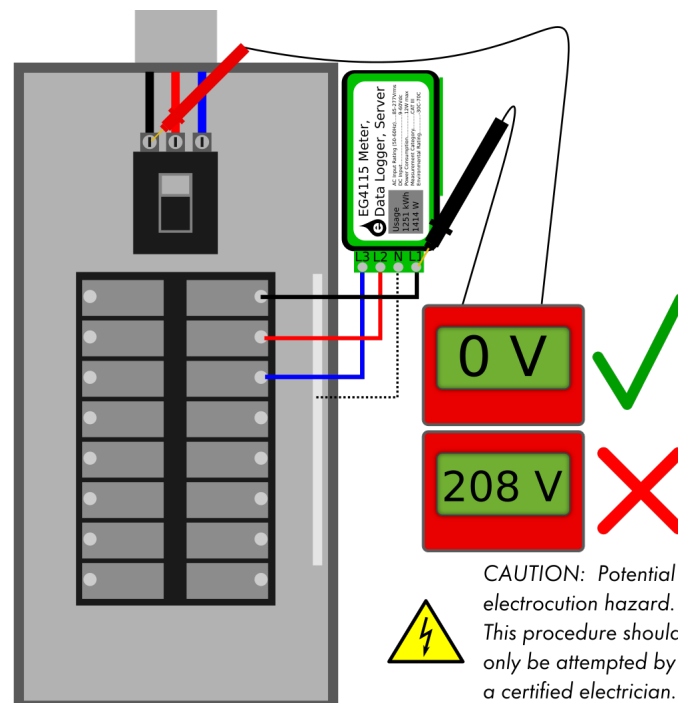
4.1 Identify correct phasing

Visit <http://egauge.net/support/phasing> for additional information.

To identify a phase mixup or adequately confirm correct phasing, use a handheld voltmeter set to AC voltage for the electrical system the eGauge is monitoring. Connect one probe to the L1 screw terminal of the eGauge 5-pin power plug, and connect the other probe to the conductor a CT is monitoring. **DO NOT** make contact with the 2-pin CT plug, only to the high voltage conductor a CT is monitoring.

If the voltmeter shows phase-to-phase voltage, the probes are on two different phases and the conductor is not on L1. If the voltmeter shows 0V differential, the probes are on the same phase and the conductor is on L1.

This process should be repeated for all lines and conductors monitored.



Note: This example shows how to use a hand-held voltmeter to phase check a 120/208V service. CTs and other wiring omitted for clarity. This procedure works with all service types.

4.2 Fixing Leg/Phase Mixups

For installations involving multiple subpanels, it is sometimes not easy to track down which conductor belongs to what leg or phase. If a CT is thought to have been installed on a conductor wired to L1 but actually was on a conductor wired to L2, the reported power will be wrong. In a split-phase system, the power will have wrong polarity (negative when it should be positive and vice versa). In a three-phase system, the power-factor may be significantly lower than expected. Rather than moving the CTs or the CT wiring, it may be easier to adjust the configuration to match reality. This can be done by adjusting the register definitions such that the incorrectly installed CT(s) are multiplied by the voltages that they're actually connected to. If the polarity of the reported power is correct and the power-factor is greater than .7, that usually can be taken as a sign that the correct phases are being used. It is usually a good idea to temporarily turn off any renewable energy systems to confirm that the polarities and power-factors make sense even when there is no renewable power.

WARNING: A high power factor is not a guarantee of correct phasing. Capacitive and inductive loads can have power factors significantly lower than .7. The only way to verify phasing is with a voltmeter as described in section 4.1.

4.3 Fixing CTs Pointing in the Wrong Direction

If a CT accidentally was installed pointing in the wrong direction, its polarity will be reversed (i.e., the power will read negative when it should be positive and vice versa). Since fixing the installation may be time-consuming, it is usually easier to adjust the configuration. This can be done easily by adjusting all register definitions which use the incorrectly installed CT to use the CT value of opposite polarity. For example, if CT3 was installed incorrectly, change all appearances of CT3 to -CT3 and vice versa.

A Formula Operators and Functions

The following operators are available in register formulas (in decreasing order of precedence):

| Operator: | Precedence: | Description: |
|-------------------|-------------|--|
| $\$“reg”$ | 6 | The value of register <i>reg</i> . |
| (e) | 5 | The value of parenthesized sub-expression <i>e</i> . |
| $-e$ | 4 | The negative of <i>e</i> . |
| $e_1 * e_2$ | 3 | The product $e_1 \cdot e_2$. |
| e_1 / e_2 | 3 | The quotient e_1 / e_2 . |
| $e_1 + e_2$ | 2 | The sum $e_1 + e_2$. |
| $e_1 - e_2$ | 2 | The difference $e_1 - e_2$. |
| $e_1 < e_2$ | 1 | 1 if $e_1 < e_2$, 0 otherwise. |
| $e_1 > e_2$ | 1 | 1 if $e_1 > e_2$, 0 otherwise. |
| $e_1 \leq e_2$ | 1 | 1 if $e_1 \leq e_2$, 0 otherwise. |
| $e_1 \geq e_2$ | 1 | 1 if $e_1 \geq e_2$, 0 otherwise. |
| $e_1 = e_2$ | 1 | 1 if $e_1 = e_2$, 0 otherwise. |
| $e_1 ? e_2 : e_3$ | 0 | e_2 if e_1 is not 0, e_3 otherwise. |

Caveat: Care needs to be taken when comparing floating-point values for equality. Except when comparing the result of a conditional operator, it is usually better to replace $e_1 = e_2$ with a test of the form $\text{abs}(e_1 - e_2) < \epsilon$, where ϵ is the largest expected error.

The following table describes the list of functions that are available in formula registers using up-to-date firmware. The descriptions are abridged and some functions may not exist on older firmware. Please refer to the function documentation on your specific device for the most up-to-date listing of available functions and full descriptions. This may be found at <http://DEV-URL/fundoc.html>, where DEV-URL is the URL of your eGauge device.

| Function: | Description: |
|--|--|
| FFTarg(<i>f</i> , <i>c</i>) | Returns phase-angle of the spectral component at frequency <i>f</i> of the signal at input channel <i>c</i> . The value returned is in degrees and may range from 0 to 360 degrees. |
| FFTarg2(<i>f</i> , <i>c</i> , <i>r</i>) | Returns phase-angle of the spectral component at frequency <i>f</i> of the signal at input channel <i>c</i> relative to the reference channel <i>r</i> . The value returned is in degrees and may range from -180 to 180 degrees. |
| FFTmag(<i>f</i> , <i>c</i>) | Returns magnitude of spectral component at frequency <i>f</i> of input channel <i>c</i> . |
| P_PV(<i>i</i> , <i>t</i> , <i>p</i> , <i>c</i>) | Given irradiance <i>i</i> , panel temperature <i>t</i> , STC-rated power <i>p</i> , and power-temperature-coefficient <i>c</i> , return expected PV output. |
| P_SunPower(<i>i</i> , <i>t</i> , <i>p</i>) | Calculates expected power of SunPower array given irradiance <i>i</i> , panel temperature <i>t</i> and rated power <i>p</i> . |
| P_est(<i>J</i> , <i>F</i> , <i>M</i> ,...) | Returns current estimated average production in Watts given expected monthly production values (in kWh). <i>J</i> is January, <i>F</i> is February, <i>M</i> is March, and so on for each of the 12 months. |
| THD(<i>f</i> , <i>c</i>) | Returns total harmonic distortion of channel <i>c</i> for fundamental frequency <i>f</i> . |
| VAr(<i>r</i> , <i>a</i>) | Given real power <i>r</i> and apparent power <i>a</i> , return reactive power. |
| __eq(<i>l</i> , <i>r</i>) | Return 1 if <i>l</i> is equal to <i>r</i> or NaN if either argument is NaN. |
| __ge(<i>l</i> , <i>r</i>) | Return 1 if <i>l</i> is greater than or equal to <i>r</i> or NaN if either argument is NaN. |
| __gt(<i>l</i> , <i>r</i>) | Return 1 if <i>l</i> is greater than <i>r</i> or NaN if either argument is NaN. |
| __le(<i>l</i> , <i>r</i>) | Return 1 if <i>l</i> is less than or equal to <i>r</i> or NaN if either argument is NaN. |
| __lt(<i>l</i> , <i>r</i>) | Return 1 if <i>l</i> is less than <i>r</i> or NaN if either argument is NaN. |
| __r(<i>s</i>) | Returns the value of register name <i>s</i> . |
| __rv(<i>n</i>) | Returns the value of register with index <i>n</i> . |
| abs(<i>x</i>) | value of <i>x</i> . |
| acos(<i>n</i>) | Returns the arc cosine of <i>n</i> |
| ago(<i>n</i> , <i>s</i> , <i>t</i> ?) | Returns the Unix timestamp that is <i>n</i> count periods earlier than time <i>t</i> . period <i>s</i> may be any of the periods that can be specified for function start_of(). |
| asin(<i>n</i>) | Returns the arc sine of <i>n</i> |
| atan2(<i>y</i> , <i>x</i>) | Returns the arc tangent of <i>y/x</i> . |
| battery_sim(<i>p</i> , <i>mc</i> , <i>md</i> , <i>cap</i> , <i>ch</i>) | Simulates a battery that can store up to <i>cap</i> Ws of energy, can be charged with up to <i>mc</i> W, and discharged with up to <i>md</i> W. <i>ch</i> is the current charge of the battery in Ws. <i>p</i> is the amount of power desired or available from the battery. |
| bit(<i>n</i> , <i>x</i>) | Returns bit <i>n</i> of number <i>x</i> . |
| chan(<i>s</i>) | Returns channel number for name <i>s</i> . <i>s</i> may be one of "L1" through "L3" for voltage inputs, or "CT1" through "CT12" for current inputs. |
| conn_qual() | Returns connection quality as percentage. |
| conn_tech() | Returns the connection technology as an integer. |
| cos(<i>n</i>) | Returns cosine of <i>n</i> , where <i>n</i> is an angle in degrees. |
| cumul(<i>s</i>) | Returns the current cumulative value of register named <i>s</i> |

| | |
|------------------------------|--|
| epoch() | Returns the time when recording started (epoch) as a Unix timestamp. |
| exp(x) | Returns 10 raised to the power of x (10^x) |
| floor(x) | Largest integral value not greater than x. |
| heat_power(r, tin, tout, m?) | Returns the power that is required to increase the temperature of medium <i>m</i> flowing at a rate of <i>r</i> cubic meters per second from <i>tin</i> to <i>tout</i> . |
| humidity_internal() | Returns the device-internal relative humidity in percent or NaN (Not-a-Number) if unavailable. NOTE: Humidity is not available on all meters. |
| isinf(x) | Returns 1 if <i>x</i> is positive infinity or -1 if <i>x</i> is negative infinity, or 0 otherwise. |
| isnan(x) | Returns non-zero value if <i>x</i> is Not-a-Number. |
| log(x) | Common (base-10) logarithm of <i>x</i> . |
| m3_per(u) | Returns the factor by which a quantity in the specified unit <i>u</i> needs to be multiplied by to get the same quantity in cubic meters (m^3). |
| map420(n, min, max) | Linearly map number <i>n</i> in the range from 4 to 20 to a number in the range from <i>min</i> to <i>max</i> . |
| map420s(n) | Linearly map number <i>n</i> in the range from 4 to 20 to a signed number in the range from -1 to +1. |
| map420u(n) | Linearly map number <i>n</i> in the range from 4 to 20 to an unsigned number in the range from 0 to +1. |
| max(x,y) | The greater value of <i>x</i> and <i>y</i> . |
| mday() | Returns the day of the month (1...31). |
| min(x,y) | The smaller value of <i>x</i> and <i>y</i> . |
| mod(x,y) | The remainder of dividing <i>x</i> by <i>y</i> . |
| month() | Number of months since January (0...11). |
| pref(p) | Returns value of preference-setting named <i>p</i> (e.g., "global.production.rated"). |
| rand() | Returns random number in the range from 0 to 1 (inclusive). |
| ravg(r, n) | Calculates the running average of register <i>r</i> over the most recent <i>n</i> minute interval. |
| round(n) | Round <i>n</i> to the nearest integer. |
| savg(r, n) | Calculates the set average of register <i>r</i> over the previous <i>n</i> minute interval. |
| sin(n) | Returns sine of <i>n</i> , where <i>n</i> is an angle in degrees. |
| sqr(x) | x^2 . |
| sqrt(x) | \sqrt{x} . |
| start_of(p, t?) | Returns the Unix timestamp of the start of the period <i>p</i> in which time <i>t</i> falls. If <i>t</i> is not specified, the current time is used. |
| stdap(p, a) | Converts the absolute atmospheric pressure <i>p</i> (in Pascals) measured at altitude <i>a</i> (in meters) to the equivalent pressure at sea level. |
| tan(n) | Returns tangent of <i>n</i> , where <i>n</i> is an angle in degrees. |
| temp_internal() | Returns device-internal temperature in degrees centigrade. |
| time(t?) | Without argument, returns current local time in hours since midnight. With an argument, returns the hours since midnight for the Unix timestamp <i>t</i> . |
| utime() | Returns current time as a Unix timestamp. |
| wday() | Number of days since Monday (0...6). |
| year(t?) | Without an argument, returns the local time's year (Common Era). With an argument, returns the year of the Unix timestamp <i>t</i> . |

B Physical Unit

The table below lists the various physical units supported by eGauge. In this table, column **Name** lists the name as it appears in the Web configuration interface (see Section 2.4.4). Column **Records** shows the physical quantity the register records and column **Unit** shows the physical unit that the data is being recorded with. eGauge records data as discrete quantities. Column **Res** shows the smallest value that can be recorded by the register and column **Max @ 30 yr** shows the maximum value that can be recorded without causing an overflow for a period of at least 30 years. For an example of how this is calculated, see section B.1.

The final column, **Code**, lists the type-code used for the register in XML and WebAPI JSON data (see <http://egauge.net/support/webapi> for the new JSON API or <http://egauge.net/support/xmlapi> for the older XML API).

| Name | Records | Unit | Res | Max @ 30 yr | Code |
|--|---------------------------------|------------------------|---------------------|------------------------|------|
| Air Quality Index (0=good, 500=bad) | Unit-less quantity | <i>n/a</i> | 1 | 19.4G | aq |
| Angle [°] | Angle (e.g., wind-direction) | ° | 1m° | 360° | a |
| Apparent Power [VA] | Apparent power | <i>VA</i> | 1VA | 19.4GVA | S |
| Charge [Ah] | Charge (e.g., battery) | <i>Ah</i> | 1mAh | 19.4MAh | Qe |
| Current (mean) [A] | DC current | <i>A</i> | 1mA | 19.4MA | Idc |
| Current [A] | RMS current | <i>A</i> | 1mA | 19.4MA | I |
| Discrete number | Unit-less number | <i>n/a</i> | 1 | 19.4G | d |
| Frequency | Frequency | <i>Hz</i> | 1mHz | 19.4MHz | F |
| Humidity | Relative humidity | % | 0.001% | 19.4M% | h |
| Irradiance | (Solar) Irradiance | <i>W/m²</i> | 1W/m ² | 19.4GW/m ² | Ee |
| Mass [g] | Mass | <i>g</i> | 1mg | 19.4Mg | m |
| Mass flow [g/s] | Mass flow | <i>kg/s</i> | 1g/s | 19.4Gg/s | Q |
| Monetary [<i>{currency}</i> /s] | Money | <i>variable</i> | 1.86 <i>n</i> | 36.2 | \$ |
| Number with 3 decimals | Number with 3 decimals | <i>n/a</i> | 0.001 | 19.4M | #3 |
| Parts per million [ppm] | Parts-per-million | <i>n/a</i> | 0.001 | 19.4M | ppm |
| Percentage [%] | Percentage | % | 0.001% | 19.4M% | % |
| Power [W] | Real power | <i>W</i> | 1W | 19.4GW | P |
| Pressure [Pa] | Pressure (e.g., atmospheric) | <i>Pa</i> | 1Pa | 19.4GPa | Pa |
| Reactive power [var] | Reactive power | <i>var</i> | 1var | 19.4Gvar | PQ |
| Resistance [Ω] | Resistance | Ω | 1Ω | 19.4GΩ | R |
| Speed [m/s] | Speed (e.g., rainfall rate) | <i>m/s</i> | 1mm/s | 19.4Mm/s | v |
| THD [%] | Total Harmonic Distortion | % | 0.001% | 19.4M% | THD |
| Temperature [°C] | Temperature | °C | 1m°C | 19.4M°C | T |
| Temperature difference [°C] | Temperature difference | °C | 1m°C | 19.4M°C | dT |
| Voltage (mean) [V] | DC voltage | <i>mV</i> | 1mV | 19.4MV | Vdc |
| Voltage [V] | RMS voltage | <i>mV</i> | 1mV | 19.4MV | V |
| Volumetric flow [m ³ /s] | Volumetric flow | <i>m³/s</i> | 1mm ³ /s | 19.4Mm ³ /s | Code |
| Whole number | Whole number | <i>n/a</i> | 1 | 19.4G | # |

Physical unit abbreviations and SI-multiplier prefixes used in the above table:

| Abbreviation | Physical unit name | Prefix | Multiplier |
|--------------|----------------------|----------|------------|
| <i>V</i> | Volt | <i>n</i> | 10^{-9} |
| <i>A</i> | Ampère | μ | 10^{-6} |
| <i>VA</i> | Volt-Ampère | <i>m</i> | 10^{-3} |
| <i>W</i> | Watt | <i>k</i> | 10^3 |
| <i>var</i> | Volt-Ampère reactive | <i>M</i> | 10^6 |
| <i>Hz</i> | Hertz | <i>G</i> | 10^9 |
| <i>g</i> | gram | | |
| <i>m</i> | meter | | |
| $^{\circ}$ | degrees (angle) | | |
| $^{\circ}C$ | degrees Centigrade | | |
| <i>Pa</i> | Pascal (kg/ms^2) | | |
| Ω | Ohm | | |

Note: Monetary registers do not have a pre-defined unit. Instead, such values are displayed using the currency-symbol selected via one of the preference settings, for example \$. The resolution for this type is very fine because the cost of electricity per second is generally extremely low. For example, if the cost of electricity were $\$0.40/kWh$, the cost per Watt-second is only $\$0.11\mu$.

B.1 Rollover and maximum value example

A Power register can record a maximum value of 19.4 gigawatts at all times for 30 years before data loss. In other words, a power register can record up to about 5 trillion kWh before a full overflow of the 64-bit signed integer will occur: 19.4GW for 30 years is 5,101,812,000,000 kWh ($19.4 * 10^9 \text{watts} * 30 \text{years} * 8766 \text{hours} / 1000 \approx 5 * 10^{12} kWh$). The range of signed 64-bit integers is $-9,223,372,036,854,775,808$ to $9,223,372,036,854,775,807$. A single wrap-around can occur (positive to negative or negative to positive), so the maximum value that can be stored is approximately $1.845 * 10^{19}$ watt-seconds ($(|s64max| + |s64min|)$), which is approximately 5 trillion kilowatt hours ($5 * 10^{12} kWh * 3.6 * 10^6 Ws \approx 1.8 * 10^{19}$)