# EKM metering plugin

Document version 1, February 2015. By A-Lurker.

Plugin code version 0.51



## Background

This plugin is designed to suit the EKM range of meters. Refer to:

http://ekmmetering.com

Specifically the:

- EKM-OmniMeter Pulse v.4
- EKM-OmniMeter I v.3

For more information refer to the EKM forum:

http://forum.ekmmetering.com

## Older meters

In theory the plugin could possibly be altered to also handle older meters. The plugin can detect older meters but the capabilities/formats of the older meters are not known. Hence they are not (currently) catered for:

5A25EDS-N
23EDS-N
25EDS-N
15EDS-N
23EDS-N
25IDS-N
15IDS-N

**Connection to Vera**

The meter must be connected to Vera using a LAN to serial converter, such as EKM's own iSerial converter:

http://documents.ekmmetering.com/iSerial_Manual_2012.pdf

Alternatively an "ATC1000 TCP/IP Ethernet to Serial RS232 RS485 RS422" converter, manufactured by Shenzhen ATC Technology Co. Ltd can be used – refer ebay:

http://www.szatc.com/product/low-cost-tcpip-to-rs-232422485-converter

When using the above converter(s), no drivers, such as "VCOM", are required. Use the converter's web interface to set up a static ip address and to select the rather unusual serial settings:

TCP mode:

| | |
|---|---|
| Telnet Server/Client | Server |
| Port Number | 50000 |
| Remote Server IP Address | don't care |
| Client mode inactive timeout | 0 |
| Server mode protect timeout | 0 |

UDP mode:
Select disabled

UART:

| | | |
|---|---|---|
| Mode | RS485 | |
| Baudrate | 9600 (for v3 & v4 meters)<br>1200 (for v2 meters) | |
| Character Bits | 7 | note: 7 not 8 |
| Parity Type | Even | note: even not none |
| Stop Bit | 1 | |
| Hardware Flow Control | none | |
| Delimiter | Non selected | |

The converter appears to use the Telnet protocol to pass the serial data around in 7 bit mode rather than binary mode. However it doesn't appear to respond to Telnet commands.

*USB to serial converters are not catered for.*

## What the plugin does not do

The plugin is designed to **report data, not write it**. However, functional code exists in the plugin to read and write specific data from/to the meter but this is not made use of in any significant way. Refer to these example functions in the plugin:

specificDataRead()
specificDataWrite()  and  meterWrData (writes the time to the meter)

Note: if expanding on the above, make sure the polling is turned off before performing a read or write. See variable **PollEnable**.

You can use the **EKM Dash software** to write to the meter, alternatively rehash the actual plugin to suit your needs, based on the example code.

Additionally:
The relay outputs on the Ver 4 meter cannot be controlled but their status can be checked.
No Vera child devices, corresponding to the relay outputs or pulse counters, are created.

## Installation

This plugin has been tested using the Vera U15 interface and Firefox 35.0.1 only. The installation for UI5 is as follows:

Go to UI5-->APPS-->Develop Apps-->Luup files and upload the five files.

Select the "Create device" button on the same page. Enter "D_EKMmetering1.xml" into the "Upnp Device Filename" entry box and select the "Create device" button immediately below. Do a few reloads using the "Reload" button.

Once the device can be seen in the User Interface and after sufficient "Reloads" and browser refreshes (generally F5);  the "Advanced" tab will have entry boxes for the following:

**ip**

Enter the IP address for the EKM meter. If you need to change the port address, you can add that on to the end of the IP address using the usual colon notation. It defaults to 50000.

**MeterModel**

Enter the digit 3 (EKM-OmniMeter I v.3) or the digit 4 (EKM-OmniMeter Pulse v.4) to match your meter model. Variables will be created depending on the meter's capabilities.

**MeterSerialNumber**

Enter the meter's serial number. It's a twelve digit decimal number that should be evident on the meter's housing. Enter all 12 digits. If you don't have the serial number try using 999999999999 ie twelve nines.

Do reloads using the Reload button and browser refreshes till all appears in place.

## Variables

Depending on the meter, the meter's own internal variables, such as Volt, Amps, Watts, etc will be created automatically.

**Polling**
The default poll rate is one minute. The plugin will not allow a faster rate but a slower rate can be set in **PollInterval.** The units are in seconds.

**A second meter can be integrated:**
The plugin can use information from a second meter (ie from another plugin) to calculate these variables:

- totalForwardkWh
- totalReversekWh
- totalNetkWh

based on the setting of the **UseSecondMeter** and **secondMeterAkWh** variables:

*UseSecondMeter = 0*
No secondary meter is in use (default).

*UseSecondMeter = 1*
A **to grid** secondary meter is in use:
The value loaded into secondMeterAkWh by the secondary meter is considered the totalReversekWh value for the calculations.

*UseSecondMeter = 2*
A **from grid** secondary meter is in use:
The value loaded into secondMeterAkWh by the secondary meter is considered the totalForwardkWh value for the calculations.

Note that all values are considered unsigned positive values and that's enforced by the plugin.

So you could write a scene that runs, say once an hour; that reads the secondary meter value and writes it to the EMK plugin, to keep the kWh readings updated.

**KWH HA variable**
In a similar fashion to UseSecondMeter, one of the following values:

- totalForwardkWh
- totalReversekWh
- totalNetkWh

can be directed to the Vera "KWH" HA variable, depending on the setting of the variable

**MeteringFunction**:

Values required for MeteringFunction are as follows:
*MeteringFunction = 0* then KWH = totalForwardkWh (default)
*MeteringFunction = 1* then KWH = totalReversekWh
*MeteringFunction = 2* then KWH = totalNetkWh

## Debugging

Debugging can be turned on and off using the variable **DebugEnabled**; off = 0, on = 1. It defaults to off.

## Meter time

The plugin contains functional software to decode and encode the meter time. If debugging is enabled (default is disabled), the meter time is written to the log file. However the meter time is not used. Any times recorded are from Vera.

## Encoded values

These variables are encoded. To make use of them, they will need decoding.

PulseInputHiLo
CurrentDirection
OutputsOnOffStatus
MaxDemandWattsTimePeriod
MaxDemandWattsAutoReset

Here are the mapping tables in hex. Variables available, depend on the meter type. Refer to the source code, to see what meter supports what:

| PulseInputHiLo | CurrentDirection | OutputsOnOffStatus | MaxDemandWattsAutoReset | MaxDemandWattsTimePeriod |
|---|---|---|---|---|
| 1,1,1 = 30 | F,F,F = 31 | off / off = 31 | off = 30 | 15 mins = 31 |
| 1,1,0 = 31 | F,F,R = 32 | off / on = 32 | monthly = 31 | 30 mins = 32 |
| 1,0,1 = 32 | F,R,F = 33 | on / off = 33 | weekly = 32 | 60 mins = 33 |
| 1,0,0 = 33 | R,F,F = 34 | on / on = 34 | daily = 33 | |
| 0,1,1 = 34 | F,R,R = 35 | | hourly = 34 | |
| 0,1,0 = 35 | R,F,R = 36 | | | |
| 0,0,1 = 36 | R,R,F = 37 | | | |
| 0,0,0 = 37 | R,R,R = 38 | | | |

## CRC

The unit uses a MODBUS CRC16. That's a traditional CRC16 that's been initiated with 0xFFFF instead of 0x0000. The CRC is calculated over all the bytes, excluding the first byte, which is always 0x02 (SOH start of header) and excluding the last two bytes, being the CRC itself.

The two CRC bytes are each and'ed with 0x7F to reset bit 7. This is done because the serial connection is 7 bits wide, not 8 bits and consequently the CRC check is a comparison of 7 bit values. The order of the two bytes is also important: lsb then msb.

Refer to:

http://www.lammertbies.nl/comm/info/crc-calculation.html

CRC info Modbus_over_serial_line_V1_02.pdf (search the net for a copy)

Also refer to the example test vectors in the plugin code.

**Protocol**

The protocol is extremely unwieldy. For example, to read one byte from the second data block (ver 4 meters); you have to read two blocks of 255 bytes each, to access the one only byte.

As there is no data delineation eg cr/lf or stx/etx, the "raw" protocol has to be used by Vera. That means each received character has to be processed individually, rather than complete lines.

It's also slow, with the following baud rates in use:
Baud Rate: 9600 (for v3 & v4 meters) or 1200 (for v2 meters)

The serial connection uses 7 bits, not 8 bits and even parity, not no parity.

**Oddities**

In the time variable: The day of the week is 1 to 7, not 0 to 6. There is also a spare 0 in the date time format.

In the reactive variables: Together with the cosine of the power factor is the letter: "L" (inductive), "C" (capacitive) or "space char" (resistive).

Version 4 of the meters can alter how many decimal points are in use, depending on the current transformers in use (the plugin handles this).

**Password**

It defaults to 00000000 ie eight zeroes. Probably best to leave it that way, as there is no factory reset method, to return it back to this default.