EmonTX calibration process

for an EmonTX bought online from the shop

Alexandre CUER

Table of contents

Basic settings – RF node number, Mhz band, network group

Some parameters can be set without compiling the firmware.

The procedure is described as follow :

Enter '+++' during the POST (Power-On Self-Test)

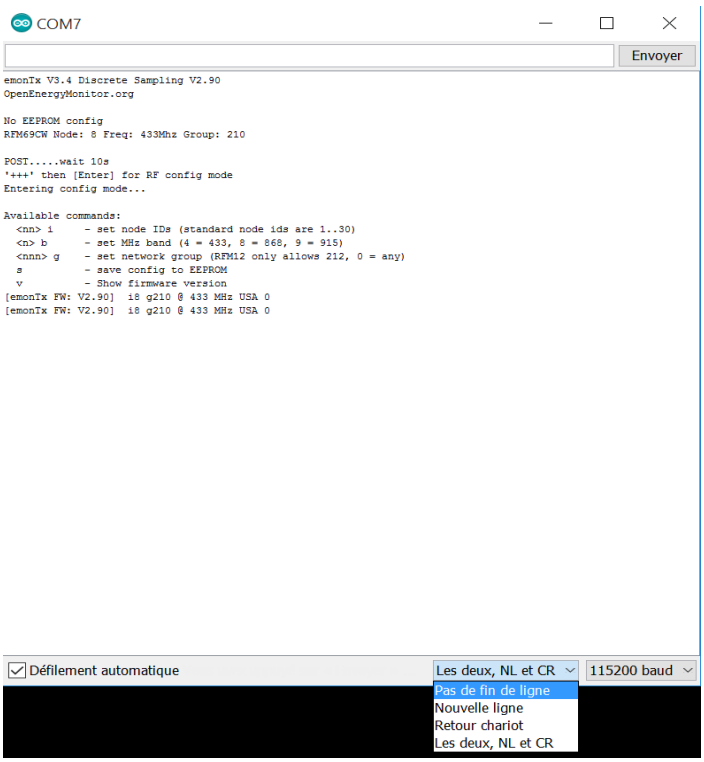
The serial monitor must be configured as follow :

- baud rate : 115200
- the "line ending" that will be sent to Arduino when you send commands must be « Both, NL & CR »

Once you've entered the config mode :
Send 1i to set the RF node to 1
Send 2i to set the RF node to 2
Send 3i to set the RF node to 3

type v to check ther version

type s to save



Advanced configuration – calibration

The calibration process requires to modify the firmware a couple of time, to compile it and to

upload it into the EmonTX microcontroller. It cannot be done via an EEPROM config.

The firmware can be downloaded on github
https://github.com/openenergymonitor/emontx3

The firmware consists of two files : src.ino and config.ino

Three libraries need to be included in the arduinoIDE, so you can compile and upload the firmware in the microcontroller :

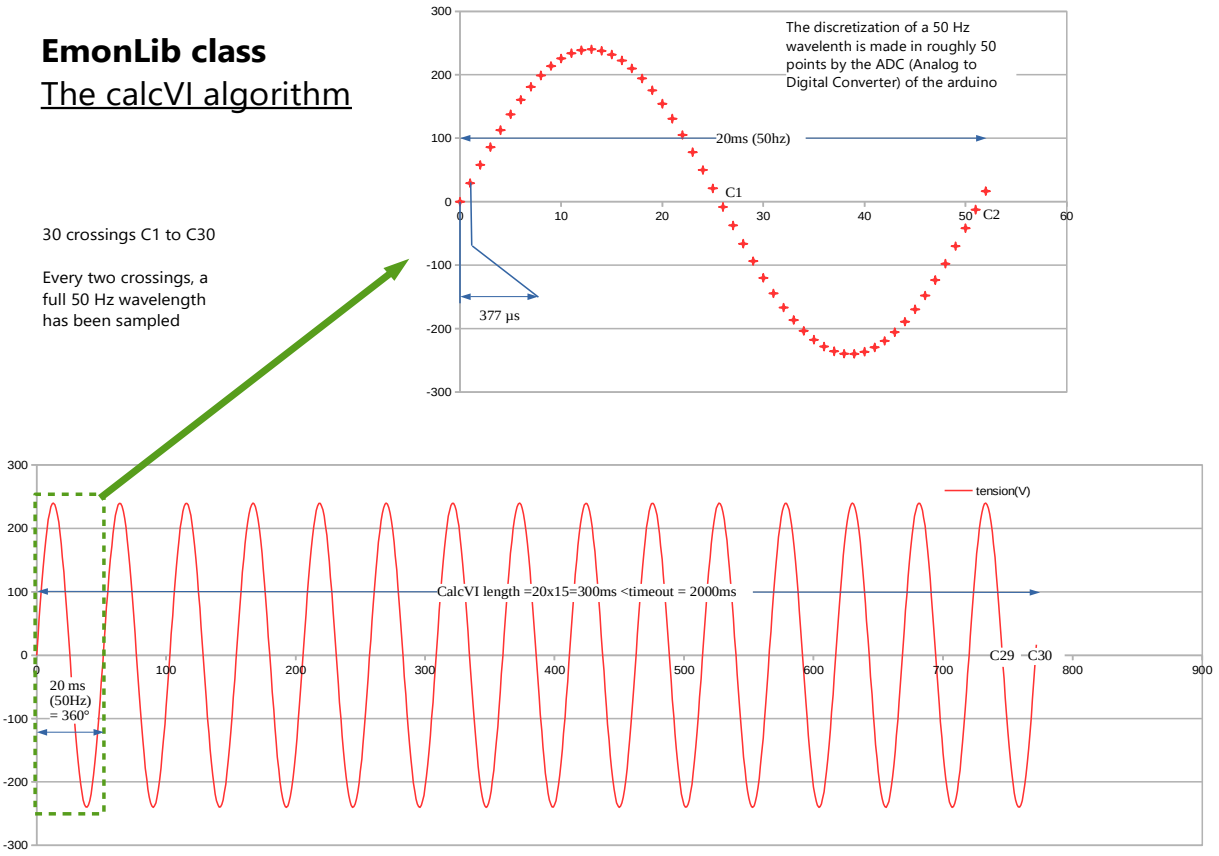
EmonLib	https://github.com/openenergymonitor/EmonLib
OneWire	http://www.pjrc.com/teensy/td_libs_OneWire.html
DallasTemperature	http://download.milesburton.com/Arduino/MaximTemperature/DallasTemperature_LATEST.zip

The src.ino file defines 4 EnergyMonitor objects, based on the EmonLib class, :

```
EnergyMonitor ct1, ct2, ct3, ct4;
```

For an EnergyMonitor object named ct1, the following methods are available :

ct1.voltage(unsigned int _inPinV, double _VCAL, double _PHASECAL)	Defines the input analog pin for voltage monitoring and the voltage and phase calibration constants cf building blocks for more details : https://learn.openenergymonitor.org/electricity-monitoring/ctac/ct-and-ac-power-adaptor-installation-and-calibration-theory and https://learn.openenergymonitor.org/electricity-monitoring/ctac/explanation-of-the-phase-correction-algorithm
ct1.current(unsigned int _inPinI, double _ICAL)	Defines the input analog pin for intensity monitoring and the intensity calibration constants
ct1.calcVI(no_of_half_wavelengths,timeout)	Calculates apparentPower, realPower, powerFactor Calculates the root of the mean of the voltage and current squared (rms) : Vrms and Irms
Ct1.serialprint()	Makes a serial print of real power, apparentPower, Vrms, Irms, powerfactor



We will use the serialprint() method in the calibration process….

In the loop section of the src ino, modify each CT statement by adding a ctx.serialprint() command, as following :

```
if (CT1) {
    if (ACAC) {
      ct1.calcVI(no_of_half_wavelengths,timeout); emontx.power1=ct1.realPower;
      emontx.Vrms=ct1.Vrms*100;
    } else {
      emontx.power1 = ct1.calcIrms(no_of_samples)*Vrms;
    }
    ct1.serialprint();
  }
```

Once the firmware is loaded you will be able to process to the calibration as described in the Learn section of the openenergymonitor web site : https://learn.openenergymonitor.org

A test report template can be used to guide you through the calibration process : emonTX_calibration.ods

Definition of a measurement standard

You can choose a multimeter as described in the learn section : https://learn.openenergymonitor.org/electricity-monitoring/ctac/how-good-is-your-multimeter

A possibility is to use an inline energy meter
The Voltcraft Energy Logger 4000F is one of those and can be found on
http://www.conrad.fr/ce/fr/product/125449/Compteur-de-consommation-VOLTCRAFT

The caracteristics of the Energy Logger 4000F are synthetised in the following table.

Energy Logger 4000F Technical data

Max. power/current	3500 W/15 A
Operating voltage	230 V/AC 50/60 Hz
Performance measurement display	0.1 -3500 W
Display energy use	0.000 - 9999 kWh
Accuracy	5 - 3500 W (± 1% + 1 count) 2 -5 W (± 5% + 1 count) < 2 W (±15% + 1 count)
Buffer battery 3 V	CR1620
Ambient conditions	10 - 50 °C/max. 90%rH (not condensing)
Operating altitude	max. 2000 m (above MSL)
Weight ca.	240 g
Dimensions (LxWxH)	164 x 82 x 83 (mm)
Overvoltage category	CAT II
Pollution degree	2

Voltage calibration

With CT1 clipped on the red or blue wire, proceed to 10 simultaneous readings of :

- the voltage calculted with the measurement standard you've decided to choose
- the voltage calcultated by the emonTX

Fill the test report template with the values and calculate the new Vcal constant

Fill the src ino file with the new value and upload the firmware on the microcontroller :

```
//float Vcal=268.97;
// (230V x 13) / (9V x 1.2) = 276.9 Calibration for UK AC-AC adapter 77DB-06-09
float Vcal=243.91;
```

Current calibration

You need a resistive load or loads (kettle, electric heater, etc) that draw a current close to but less than the maximum that your measurement standard can measure.

If you use an inline energy monitor, you can connect the load(s) on it.
If you use a multimeter, you have to connect it in in series with the load(s)

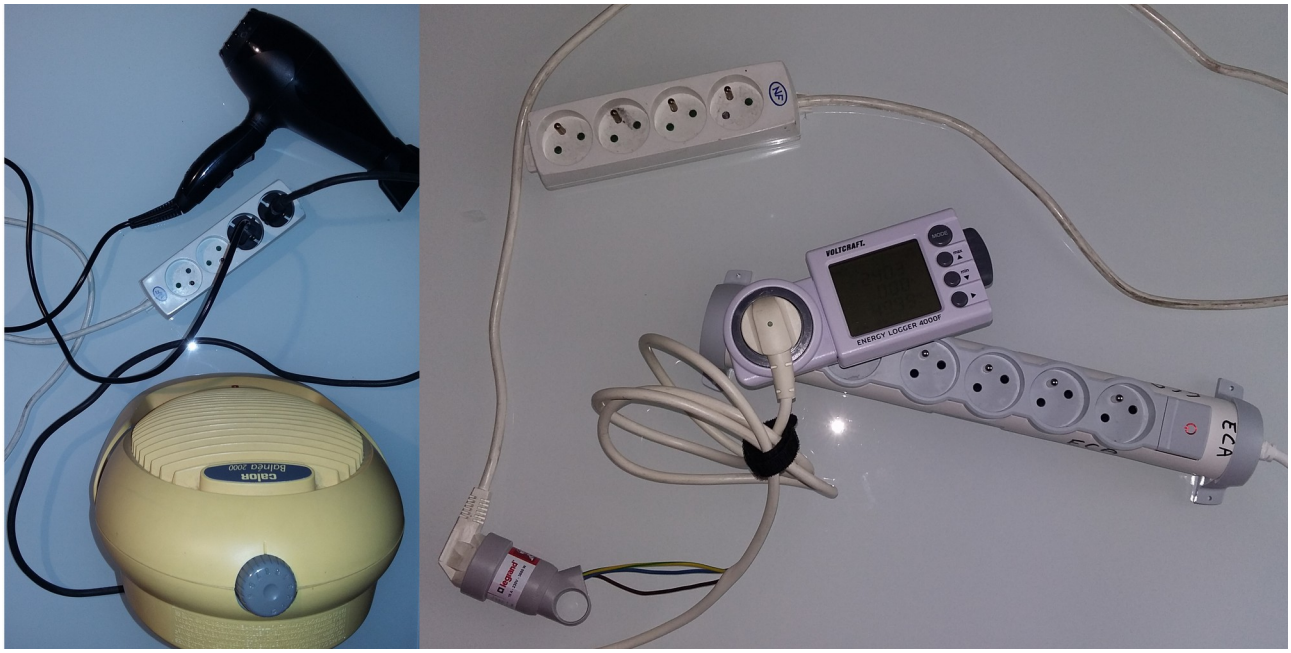


Illustration 1: picture of a calibration setup with two loads and an inline multimeter

Successively for each CT (current transformer), proceed to 10 simultaneaous readings of :

- the current measured by the measurement standard
- the current calculated by EmonTX via the CT sensor connected

Fill the test report template with the values and calculate the new Ical constant for each CT sensor connected

Fill the src ino file with the new value and upload the firmware on the microcontroller :

```
//const float ical1=90.9;
// (2000 turns / 22 Ohm burden) = 90.9
const float ical1=89.1;
const float ical2=89.6;
```

Phase calibration

You still need the load(s) for that part of the calibration
You can test different values of the phase correction constant, so that the power factor equals to 1 when using the load(s)
Try different average values of current intensity.

An example is given in the following table, with 4 current intensities tested

Average current consumption	9	9	9	8	6	6	6	6	4	4
phase_shift	1,4	1,7	1,35	1,4	1,35	1,4	1,45	1,5	1,5	1,4
power factor	1	1,01	1	1	0,99	0,99	0,99	0,99	1	1

As explained in the learn section, a phase shift of 1,3 will in theory correct the 2° error caused by the delay between sampling voltage and current

RF Payload and EmonHub configuration

Datas are packed using a C structure (cf https://en.wikipedia.org/wiki/Struct_%28C_programming_language%29)

This structure is defined as follow :

```
typedef struct {
  int power1, power2, power3, power4, Vrms, temp[MaxOnewire];
  unsigned long pulseCount;
} PayloadTX;      // create structure - a neat way of packaging data for RF comms
```

Each EmonTX will send the datas'payloads via RF to EmonHub

The [node] section of emonhub must be configured as follow, as far as the EmonTX transmitting on RF node 1 is concerned :

```
[[1]]
    nodename = emonTx3_Node1
    firmware =V2_3_emonTxV3_4_DiscreteSampling
    hardware = emonTx_(NodeID_DIP_Switch1:OFF)
    [[[rx]]]
        names = power1, power2, power3, power4, Vrms, temp1, temp2, temp3, temp4, temp5, temp6,
pulse
        datacodes = h,h,h,h,h,h,h,h,h,h,L
        scales = 1,1,1,1,0.01,0.1,0.1, 0.1,0.1,0.1,0.1,1
        units =W,W,W,W,V,C,C,C,C,C,C,p
```

If you do not use the RJ45 port to connect single wire temperature sensors, you can disable the single wire bus in the src.ino :

```
const byte MaxOnewire=                0;
```

This will imply to modify the configuration of the node in emonhub as follow :

```
[[1]]
    nodename = emonTx3_Node1
    firmware =V2_3_emonTxV3_4_DiscreteSampling
    hardware = emonTx_(NodeID_DIP_Switch1:OFF)
    [[[rx]]]
        names = power1, power2, power3, power4, Vrms, pulse
        datacodes = h,h,h,h,h,h,h,h,h,h,L
        scales = 1,1,1,1,0.01,0.1,0.1, 0.1,0.1,0.1,0.1,1
        units =W,W,W,W,V,C,C,C,C,C,C,p
```

RSSI (Received Signal Strength Indication) ranges to appreciate the quality of the 868 Mhz radio signal :

- -10 to -50 dBm : excellent quality
- from - 50 to -85 dBm : good quality
- under -85 dBm : poor quality

If calculated power is negative for all inputs of a single EmonTX, you can change the orientation of the ACAC adapter in the 220 plug.

If only a single input is concerned, change the direction of the CT on the wire.

SAFETY

Your safety is your responsibility. Clip-on current sensors are non-invasive and should not have direct (metallic) contact with the AC mains. However, installing the sensors will require working in close proximity to cables carrying high voltage. As a precaution, we recommend ensuring the cables are fully isolated, i.e. power is switched off prior to installing your sensors, and proceeding slowly and with care.

If you have any doubts, seek professional assistance.

Always connect the C.T. leads before clipping it round a current-carrying conductor, and always un-clip the C.T. from the current-carrying conductor before disconnecting its leads.

Never open-circuit a C.T. whilst it is on a current-carrying conductor. It is always safe to short-circuit a C.T.

Never attempt to fit a C.T. to a bare conductor unless you are absolutely certain it is designed for that. Apart from the obvious danger of electric shock, there are two factors to be considered: the strength of the C.T.'s insulation, and its ability to withstand the higher temperatures at which bare conductors normally operate.

A split-core CT, especially one that has a ferrite core (such as the ones made by YHDC) should never be "clamped" to the cable using any sort of packing material, because the brittle nature of the ferrite core means that it might easily be broken, thus destroying the CT. You should only clamp the CT to the cable or busbar if the housing is specifically designed to do so. Similarly, a ring-core CT should never be forced onto a cable that is too large to pass freely through the centre. The position and orientation of the cable within the CT aperture makes no practical difference to the output.