EmonTx Phase Calibrator

This sketch (emonTx_Phase_Calibrator.ino) is based on the 3-phase PLL sketch and MartinR's PLL energy diverter. It is intended for use when setting up and calibrating an emonTx energy monitor. A programmer and IDE is required to upload, operate this sketch and read the calibration result; and after use, to re-load and calibrate the working sketch.

This document has the following main sections:

Key Properties

- Allows fine adjustment of the PHASECAL (phase error compensation) constant, which is the difference between the phase errors of the voltage and current transformers.
- Can also determine the amplitude calibration constants for emonLibCM.
- User-defined reporting interval.
- Suitable for use on all emonTx variants, at 50 or 60 Hz, and produces values that can be entered into the calibration of emonLibCM. The value found is not directly usable in emonLib (discrete sample version).
- Requires the Arduino IDE and a programmer.

Initial Configuration

Note: *This sketch communicates at 9600 baud, and no faster.*

The following parameters must be set in the sketch for correct operation. This is done using both *pre-processor directives* and normal variable definitions. There are three settings that must be checked and set prior to compiling and loading the sketch:

1. `#define EMONTX_V34`
   This sets the I/O pin numbers and default calibration for the board. You must use EMONTX_V22 or EMONTX25 or EMONTX_V32 or EMONTX_V34 or EMONTX_SHIELD as appropriate.
2. `#define SUPPLY_FREQUENCY 50`
   This defines the nominal line frequency, either 50 or 60 (Hz).
3. `#define SUPPLY_VOLTS 3.3`
   This defines the voltage that the board runs at. Use 5.0 for Arduino / emonTx Shield or 3.3 for everything else.

Sensor Calibration Constants

It is not necessary to calibrate the voltage and current readings if only the combined phase error is required.

These values are changed at run time. Note that none of these values are remembered by this sketch, therefore they must be recorded manually. If you are using this sketch to determine the voltage and current calibration constants for emonLibCM, you must keep each c.t. with its respective input, and transfer the constants to the working sketch after you have finished calibration and reloaded the working sketch.

VCAL

VCAL is the voltage that gives 1 V rms as measured by the ADC input. This is a function of the a.c. adapter / transformer voltage ratio and the resistor divider network. For the emonTx V3 using the 'standard' UK a.c. adapter, the calculated value is 240:11.6 x 13:1 = 268.97 For the emonTx Shield, the calculated value is 234.2

This value depends on tolerances both of the v.t. itself and of components within the emonTx and should be adjusted for best accuracy. The same constant is used by emonLibCM.

It is changed by entering the string

`v<x.xx>` where x.xx = a floating point number for the voltage calibration constant.

The current input pin is defined at run time. Only one power input (voltage transformer in combination with a current transformer) can be the active at any one time.

The c.t. input pin is changed by entering the string

`I<x>` `where x` = the Current input pin number: emonTx V2.2: 3, 0, 1, others: 1-4

ICAL

ICAL is the current that gives 1 V rms as measured by the ADC input. It is a function of the current transformer ratio and the burden resistor value (or the current transformer's current:voltage ratio for c.t's with an internal burden resistor).

For the emonTx V3 inputs 1-3 and the 'standard' current transformer, the calculated value is 100A:0.05A for the current transformer ÷ 22 Ω for the burden resistor = 90.91

For the emonTx Shield, the calculated value is 60.6 for each input.

This value depends on tolerances both of the c.t. itself and of components within the emonTx and should be adjusted for best accuracy. The same constant is used by emonLibCM.

It is changed by entering the string

`i<x.xx>` where x.xx = a floating point number for the current calibration constant.

Only one power input (voltage transformer in combination with a current transformer, selected by choosing the input pin), can be calibrated at a time.

Data reporting interval

This is the period over which the power factor is averaged, and thus the interval between values being presented.

It is changed by entering the string

`d<x.x>` where x.x = a floating point number for the data reporting interval in seconds.

You will probably want to start with a fairly short interval, say 2 s, and lengthen the interval when closing in on the final value.

PLL Settings

`LEDISLOCK`

The on-board LED lights to indicate that phase lock has been obtained, and flickers very briefly to indicate a data transmission (this might not be readily visible). Comment this line for the ‘standard’ LED behaviour where a flash indicates data transmission (not recommended).

PLLTIMERRANGE
Sets the limits for the PLL timer. 100 = ~ +/-0.5Hz. This should not normally be changed.
Default: 100

PLLLOCKRANGE
The allowable ADC range to enter locked state. This should not normally be changed.
Default: 40

PLLUNLOCKRANGE
The allowable ADC range to remain locked. This should not normally be changed.
Default: 80

PLLLOCKCOUNT
The number of cycles to determine if PLL is locked. This should not normally be changed.
Default: 100

On-line calibration & adjustments

On startup, in the IDE monitor window, you will see a short menu, and the sketch will start taking readings every 2 s. Here is the output when running on an emonTx V3.4:

```
OpenEnergyMonitor.org
emonTx V3.4 PLL Phase Calibrator V1.0

Available commands:
  l        - list the settings
  r        - restart using sketch default values
  F        - show firmware version
  ?        - show this text again

  d<x.x>   - x.x = a floating point number for the data reporting interval in seconds
  v<x.xx>  - x.xx = a floating point number for the voltage calibration constant
  I<x>     - x = the Current input pin number: emonTx V2.2: 3,0,1, others: 1-4
  i<x.xx>  - x.xx = a floating point number for the current calibration constant
             [These calibration constants are only needed if you require to
              measure the phase difference at a known voltage and current.]
  p<x.xx>  - x.xx = a floating point number for the phase lead (degrees)
  l        - list the config values

  PROCEDURE:
  Set the emon board type and recompile & reload if necessary.
  Set a Current pin, ensure "PLL is locked" Adjust phase lead (p) until "pf:" shows
  less than 0.002
  WRITE DOWN the value for phase lead, repeat for each c.t. in use,
  then manually transfer those to the other sketch.

  See the documentation for full details.

Vrms: 238.10 I: 23.36 f: 50.049  PLL is unlocked
Vrms: 238.23 I: 23.35 f: 50.056  PLL is locked  pf: -0.0150 phase: 0.000
Vrms: 238.23 I: 23.34 f: 50.055  PLL is locked  pf: -0.0151 phase: 0.000
```

`[etc]`

When you change one or more of the settings, the change will take effect immediately.

Setting up for Calibration & Measurement

Calibration beyond the first three settings above will not be necessary unless you need an accurate measurement of voltage and current at which the phase error is determined. If you don't need that, you don't need a multimeter and you don't adjust the voltage and current calibration values; but you still need to set up a resistive load, the a.c adapter and the c.t's.

Before calibrating the sketch, read (but do not do) the calibration instructions in Learn > Electricity Monitoring > Current & Voltage. Those instructions contain the general procedure and safety warnings, which you must be familiar with. The detailed instructions that follow apply only to this sketch. Follow *these* instructions for the order in which to make the adjustments and how to apply the values in the sketch, but follow the *general instructions* for how to proceed with the measurements.

1. Obtain a resistive load that will draw a current that is close to but less than the maximum that your multimeter can read and less than the rating of your c.t. If you wish to measure the phase error at a higher current, you can pass the wire several times through the c.t. The current that the c.t. sees will be the actual current multiplied by the number of passes through the c.t. The entire calibration and measurement procedure can be done on the test bench.
2. Plug in the a.c. adapter. Using your multimeter (or a reliable voltage measurement), adjust VCAL so that the voltage measured is the same as your meter indicates.
3. Set a c.t. (or all c.t's that will be used) on the cable to your test load, the direction the c.t. faces is not important.
4. Insert your multimeter in series with the test load, to measure current. Adjust the current calibration factor ICAL to read the same current as your meter, or the appropriate multiple.

Measuring the phase error

The procedure is, you adjust the timing inside the sketch by giving it a new value for `phase` so that the power factor shown is as close to zero as is *reasonable*. Due to minor fluctuations in line voltage and frequency and because the measurement technique is so sensitive, it is not necessary to aim for a power factor lower than 0.002 (ignoring the sign), even though the value displayed is shown to 4 decimal places.

The phase adjustment starts at 0.0°. Note the power factor reading, and change the phase adjustment to 1.0°.

You do this by entering the string

p<`x.xx`> where x.xx = a floating point number for the phase lead (degrees). You may enter a negative value.

This is likely to move the power factor closer to zero. Adjust the value so that the power factor is acceptably close to zero. Bear in mind that the "standard" c.t. and v.t. have phase errors that change by around 1.5° and 3° respectively over their possible working ranges, though the *difference,* which is what is measured here, will vary much less. A p.f. of 0.002 represents a phase error of a little more than 0.1°

Make a note of the final phase adjustment value. This value belongs to the combination of this c.t. and the v.t. Later, you will need to enter it to calibrate your working sketch.

Change the pin number to another c.t. and repeat the procedure for each c.t. On the emonTx V3, c.t's 1 – 3 should show roughly the same phase difference, c.t. 4 will be significantly different, due to the different burden value.

Data Output Format

Data output is to the serial monitor only.

```
Vrms: 238.10 I: 23.36 f: 50.049  PLL is unlocked
Vrms: 238.23 I: 23.35 f: 50.056  PLL is locked  pf: -0.0150 phase: 0.000
```

The first line shows the output when the PLL is not locked. Lock must be achieved in order for the sketch to work.

The second line shows the normal output. "**pf**" shows the measured power factor, and "**phase**" the latest value you entered for the phase difference.

Explanation of operation of the sketch

General Principle

The sketch measures samples of voltage and current 36 times in each cycle of mains electricity. The voltage sample is delayed by 9 samples (90°) so that a pure resistive load will appear to be a pure capacitive reactance, which absorbs zero real power averaged over the cycle.

The sketch uses a Phase-Locked Loop (PLL) to ensure that the readings always happen at the same place on each phase on each mains cycle, even though the mains frequency can change.

Pairs of readings (voltage and current) are multiplied together and added in an accumulator. At the end of the 'data reporting interval', the average power is calculated. The rms averages for the voltage and current are calculated in a similar way, and multiplied together to give the apparent power. Finally, real power is divided by apparent power to give the power factor. The actual amplitudes of voltage and current do not directly affect the power factor, though they will affect the phase errors of the individual transformers.

REQUIRED LIBRARY

This library is required to support the sketch:

SPI [Arduino standard library]

The sketch comprises two files:

emonTx_Phase_Calibrator.ino
emonTx_Phase_Calibrator_config.ino

Acknowledgements

Martin Roberts (MartinR) for the original PLL energy diverter and RFM12B interface.
Jörg Becker for background work on interrupts and the ADC and all those who contributed to the 3-phase PLL sketch.

Documentation Version 1.0 10 April 2021
for Sketch Version 1.0