emonTx_V2_CT123_Voltage_Temp_Pulse_rfm69n.ino

This sketch should only be used when transmitting data to an emonPiCM that expects the radio data packet to use the “RFM69 Native” format. An emonPi or emonBase **not** using that format will be unable to receive the data from this sketch.
(Use emonTx_V2_CT123_Voltage_Temp_Pulse.ino instead.)

This sketch uses the Continuous Monitoring library.

Setting up

The emonTx V2 was originally fitted with a RFM12B radio module. That must be set in line 33 of the main sketch file:

#define RFM12B

unless of course it has been replaced by a RFM69CW. The range of r.f. power levels, and the number of discrete levels available in the RFM12B is very limited – the nearest available power corresponding with the desired setting will be used. If a RFM69CW has been fitted, ensure the correct band for the module fitted is chosen, and an efficient aerial is present, if the output power is turned up beyond the default value of 25. The RFM69CW can be damaged beyond repair if it is operated at or near full power without the proper aerial.

You must set the p.c.b. version appropriately in line 45.

Data is transmitted by radio approximately every 10 s. There is no longer a “standard” Node ID allocated to the emonTx V2, however, the data packet transmitted is the same as the emonTx3CM sketch, node ID 15, though there is of course no 4th c.t. input, which will always read zero.

Additional set-up options

At power-up, if you have a serial monitor connected via the FTDI connector and set to 115200 baud with “both NL & CR” selected, you will see a “Welcome” message giving the software version number, and a report of the configuration settings and calibration values.

```
emonTx V2 EmonLibCM Continuous Monitoring V0.00
OpenEnergyMonitor.org
Settings:
Group 210, Node 15, Band 433 MHz
Calibration:
vCal = 227.60
assumedV = 240.00
i1Cal = 111.10
i1Lead = 4.20
i2Cal = 111.10
i2Lead = 4.20
i3Cal = 111.10
i3Lead = 4.20
```

```
datalog = 9.85
pulses = 1
pulse period = 100
temp_enable = 1
Temperature Sensors found = 0 of 1
Temperature measurement is NOT enabled.

RF on
RFM12B Freq: 433MHz Group: 210 Node: 15

CT1 detected, i1Cal:111.10
CT2 detected, i2Cal:111.10
CT3 detected, i3Cal:111.10
AC missing
```

You will then see the standard serial data output, in a format compatible with the emonHub Serial Interfacer, which is sent to the FTDI port. It will be similar to this:

```
MSG:1,P1:284,P2:2242,P3:3829,E1:1,E2:6,E3:10,pulse:0
MSG:2,P1:21,P2:2237,P3:3853,E1:1,E2:12,E3:21,pulse:0
[etc]
```

At any time, you may enter the configuration and calibration mode by sending the characters **+++**[**Enter**]

If you successfully enter the access code, you will see an advisory and a menu of commands:

```
Entering Settings mode...

Available commands:
  l         - list the settings
  r         - restore sketch defaults
  s         - save settings to EEPROM
  v         - show firmware version
  z         - set the energy values and pulse count to zero
  x         - exit, lock and continue
  ?         - show this text again

  w<x>      - turn RFM Wireless data off: x = 0 or on: x = 1
  b<n>      - set r.f. band n = a single numeral: 4 = 433MHz, 8 = 868MHz,
                9 = 915MHz (may require hardware change)
  p<nn>     - set the r.f. power. nn - an integer 0 - 31 representing -18 dBm to +13 dBm.
                Default: 25 (+7 dBm)
  g<nnn>    - set Network Group  nnn - an integer (OEM default = 210)
  n<nn>     - set node ID n= an integer (standard node ids are 1..60)

  d<xx.x>   - xx.x = a floating point number for the datalogging period
  c<n>      - n = 0 for OFF, n = 1 for ON, enable voltage, current & power factor
                values to serial output for calibration.
  f<xx>     - xx = the line frequency in Hz: normally either 50 or 60
  k<x> <yy.y> <zz.z>
            - Calibrate an analogue input channel:
            - x = a single numeral: 0 = voltage calibration, 1 = ct1 calibration,
                  2 = ct2 calibration, etc
            - yy.y = a floating point number for the voltage/current calibration constant
            - zz.z = a floating point number for the phase calibration for this c.t.
                 (z is not needed, or ignored if supplied, when x = 0)
            -  e.g. k0 256.8
            -       k1 90.9 2.00
  a<xx.x>   - xx.x = a floating point number for the assumed voltage if no a.c.
                is detected
  m<x> <yy> - meter pulse counting:
               x = 0 for OFF, x = 1 for ON, <yy> = an integer for the pulse minimum
                period in ms. (y is not needed, or ignored when x = 0)
  t0 <y>    - turn temperature measurement on or off:
            - y = 0 for OFF, y = 1 for ON
  t<x> <yy> <yy> <yy> <yy> <yy> <yy> <yy> <yy>
            - change a temperature sensor's address or position:
            - x = a single numeral: the position of the sensor in the list (1-based)
            - yy = 8 hexadecimal bytes representing the sensor's address
               e.g.  28 81 43 31 07 00 00 D9
               N.B. Sensors CANNOT be added.
```

And the standard output will continue...

```
MSG:10,P1:31,P2:2224,P3:3829,E1:1,E2:61,E3:105,pulse:0
MSG:11,P1:35,P2:2236,P3:3850,E1:2,E2:67,E3:115,pulse:0
MSG:12,P1:34,P2:2234,P3:3847,E1:2,E2:73,E3:126,pulse:0
```

You can, and must, issue commands *between* the output messages, therefore you might find it convenient to set the datalogging period to a longer time, say 30 or 60 s, but you must remember to restore it before you save the settings and exit.

Exiting the configuration and calibration mode with **x** ensures that the access code will again be required, thereby reducing the risk of a change being made by accident.

If you exit *without* saving the settings, they will be used until the next restart, when it will revert to using the previous settings.

Note that Data Whitening must not be enabled in emonHub.

Data Output

The output by radio is, in order:

Message no.
Vrms
Powers 1-4
Energies 1-4
Temperatures 1-3
Pulse Count.

(Note: The values for Power 4 and Energy 4 will always be zero.)

The space-separated serial output is the same, but preceded by the NodeID.

The “key:value” pairs serial output for the EmonESP & EmonHubTx3eInterfacer sends the variables in the same order, but only the values in use are sent. The quantities are identified:

Message no.	MSG
Powers 1-4	P1 - P4
Energies 1-4	E1 - E4
Temperatures 1-3	T1 - T3
Pulse count	pulse