

## Configuration of RF Module & on-line calibration

The user has the opportunity during the Power-On-Self-Test (POST) procedure to turn the RF module on or off, to change the RF power and frequency, the Network Group and the Node ID.

When running, the sensor calibration and various other settings can be adjusted.

To enter configuration mode at start-up, using the serial monitor part of the Arduino IDE, enter “+++” followed by the [Enter] key when prompted. You must respond within 10 seconds. If you do nothing, the start-up procedure continues normally after the timeout has expired.

You will then see a short menu:

Available commands for config during start-up:

```
b<n>      - set r.f. band n = a single numeral:
           4 = 433MHZ, 8 = 868MHZ, 9 = 915MHZ
           (may require hardware change)

f<x>      - the line frequency in Hz
           x = an integer value, normally either 50 or 60

g<nnn>    - set Network Group nnn - an integer (OEM default = 210)
           (g - reports the network group)

k         - turn on-line calibration on. Default on startup or on reset is OFF.

i<nn>     - set node ID (standard node ids are 1..30)
           (i - reports the node ID)

p<nn>     - set RF Power nn = an integer, 0 - 31. 0 = -18 dBm, 31 = +13 dBm,
           default = 19 (+7 dBm)
           DO NOT USE a value > 19 unless an antenna is fitted.

r         - restore sketch defaults

s         - save config to EEPROM

v         - Show firmware version

w<x>      - turn RFM Wireless data on or off:
           x = 0 for OFF, x = 1 for ON, x = 2 for ON with whitening

x         - exit and continue

?         - show this text again
```

Available commands when running:

```
c<n>      - enable voltage, current & power factor values to serial output for
           calibration. n = 0 for OFF, n = 1 for ON,

k<x> <yy.y> <zz.z>
           - sensor calibration
             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

l         - list the config values

m<x> <yy>
           - meter pulse counting
             x = 0 for OFF, x = 1 for ON,
             y = an integer for the pulse minimum period in ms.
             y is not needed, or ignored, when x = 0
```

```

p<xx.x> - datalogging period
          xx.x = a floating point number for the datalogging period in seconds
s        - save config to EEPROM
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>
          - 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
?        - show this text again

```

*Note: The sketch might not make use of all of the options listed here.*

On-line calibration is disabled by default. It must be turned on at start-up when required. When RFM wireless data is turned off, the serial output in a format suitable for the ESP8266 WiFi module is automatically turned on, and on-line calibration & configuration is therefore not available.

For the configuration commands changed during start-up, if you change one or more of the settings, the change will take effect when you exit and continue (option 'x'). Take care that the correct RF frequency is selected to match your hardware. Operating the transmitter at high power on the wrong frequency, or without an effective antenna, can destroy the RFM module.

For the settings you change when running, the command will be acknowledged and you will see the displayed values change. You will also see confirmation when you save the changes. When you change one or more of the settings, the change will take effect immediately.

Option ('s') will save all the changes. If you do not do this, the settings will revert to the previous values at the next restart. After you save ('s') the changes, the new settings will be used forever, or until changed again.

If you restore the sketch default values ('r' during start-up), all the EEPROM data is ignored and the sketch restarts immediately, using the values set in the sketch. There is then no means of recovering the EEPROM data.

If the EEPROM has been used previously and has had non-compliant values written, the EEPROM content will be ignored and the sketch will start using its own default values. Saving the configuration will format the EEPROM and the sketch's set values will be stored.

## Using the emonTx with the ESP8266

The ESP8266 accepts serial data from the emonTx to send via WiFi. It also appears to send unwanted characters back to the emonTx, which conflict with the on-line calibration facility detailed above. For that reason, the **k** command is included in the start-up options. In order to perform on-line calibration, the ESP8266 must be disconnected and a programmer connected to allow the user to interact with the emonTx via the Arduino IDE's monitor screen. During the start-up phase, calibration must be specifically enabled (issue the command **k**), and when calibration is complete and the settings saved, the emonTx

must be restarted, when on-line calibration is disabled automatically. If this is not done, the ESP8266 can interfere with correct operation of the emonTx in an unpredictable manner. Do not confuse this with the calibration command *when running* for adjusting the voltage and current inputs.