rfmTxLib User Interface

The library should be saved in the user's library folder in the usual way.

This library is a stripped down version of the JeeLib library for the Hope RFM12B and RFM69CW radio modules. Only transmit functions are implemented. This library should be used where speed is of the essence, notably in the 'continuous monitoring' sketches.
It implements the “Classic” JeeLib RFM12 protocol. If the later “RFM69 Native” JeeLib RFM12 protocol is required, then “rfm69nTxLib” should be used instead. All credit for the major part of the code goes to Jean-Claude Wippler and his team, with a small section being the work of Martin Roberts.

To use the library, the following parameters must be defined in the sketch, using the preprocessor directives and variable declarations:

#define RFM69CW
Defines the type of Radio. Can be RFM12B or RFM69CW. You MUST use the one matching the module you have.

The following two definitions are required in the main sketch only if different pins are used to connect to the radio module:

#define RFMSELPIN 10
Defines the pin that enables the RFM Radio module. This is set by the pcb design:
10 for the emonTx V3.4; 5 or 10 for the emonTx Shield, according to the choice of jumper setting; 4 for the V3.2/RFμ_328 V1.2

#define RFMIRQPIN 2
Defines the pin that carries the interrupt from the RFM Radio module. This is set by the pcb design: 2 for the emonTx V3.4; 2 or 3 for the emonTx Shield, according to the choice of jumper setting; 3 for the V3.2/RFμ_328 V1.2

#include <rfmTxLib.h>
This includes the OEM library for the RFM12B & RFM69CW and must follow all of the above directives.

Initialising the library
The library is initialised, and the correct settings sent to the module, with the following function call, normally placed inside the `setup()` function:

void rfm_init(void);
There is no return value.

Sending data
The data to be sent must be a contiguous byte string. The length must not exceed 60 bytes. (The simplest way to organise several variables in this format is to set up the appropriate ***struct*** and assign values to the members of the structure in the normal way.) A pointer to the start of the data and the length, together with the group and node IDs are passed tby calling `rfm_send( )`. The full definition is:

bool rfm_send(const byte *data, const byte size, const byte group,
const byte node[, const uint8_t RF_freq[, const byte rf_power
[, const int threshold[, const byte timeout]]]]);

The minimum needed is:

```
rfm_send((byte *)&emontx, sizeof(emontx), networkGroup, nodeID);
```

where `emontx` is the name of the structure, and `networkGroup` and `nodeID` are as defined below.

`group` defines the wireless network group. This must be the same as the emonBase / emonPi and emonGLCD, if used. Normally, this variable will be a constant, but that is not a requirement. The group may be changed programmatically each time a message is sent. Refer to JeeLib documentation for more information about the usage of the network group. Specifically, all groups use the same radio band, therefore although receivers ignore traffic in a different group, mutual interference might still occur. The OEM default group is 210.

`node` is the Node ID for this sensor node. Normally, this variable will be a constant, but that is not a requirement. The Node ID may be changed programmatically each time a message is sent. The Node ID must be unique on the system. Allowed values are 1 – 30 in “Classic” format.

The remaining values `rf_freq, rf_power, threshold` and `timeout` are optional, and the default will be used if these are not specified.

`rf_freq` must be one of the constants RFM_433MHZ, RFM_868MHZ or RFM_915MHZ. The same frequency must be used throughout the installation. The default is RFM_433MHZ (the actual frequency is 434.000 MHz).

`rf_power` has a permissible range for the RFM69CW of 0 – 31, representing output powers of -18 dBm – +13 dBm. The range for the RFM12B is 7 – 25, representing output powers of -10.5 dBm – +7 dBm. The default value is 25, which gives an output power of +7 dBm for both modules. If `rf_power` is specified, `rf_freq` must also be specified.
Caution: The RFM69CW module might be destroyed if operated at or near maximum power without an efficient antenna for the frequency in use.

`threshold` is the received signal level detected on the radio band, below which the band is deemed “clear” and transmission may take place. The permissible range for the RFM69CW is -127 dBm – 0 dBm in 1 dB steps, and for the RFM12B is -103 dBm to -73 dBm in 6 dB steps. The default is -97 dBm. If `threshold` is specified, all the preceding optional parameters must also be specified.

`timeout` is the time in ms that transmission will wait while the band is “busy” - i.e. a signal above the threshold is detected *at the transmitter*, it does not necessarily indicate conditions at the receiver. The data is sent regardless when the timeout period elapses. If `timeout` = 0, there is no check for another signal. If `timeout` is specified, all the preceding optional parameters must also be specified.

The return value is `true` if the data was sent before timeout occurs, otherwise `false`. The sketch can use this information to cause the data to be sent again, if desired.

Note:
It is possible to lower the bit rate at which data is sent, which may help in difficult conditions, e.g. at the extreme limits of range. However, this setting needs to be changed

in all transmitters and receivers in the system, and is not readily available in JeeLib. The registers that need changing are identified in the source code. The default bit rate is approximately 49.2 kbps.

Version 1.0 27/6/2020 First release

Version 2.0 10/3/2021 Frequency setting moved into transmit function.