**Reader for Fine Offset type wireless weather stations using the FSK transmission protocol**

The original idea for this project was to connect the base station of my National Geographic NG265NC weather station to my newly acquired Raspberry Pi and use pywss or something similar to read the data through a USB connection. However, the USB connections of Fine Offset stations are legendary for their unreliability. So when I saw [Kevin Sangeelee’s blog post](Kevin%20Sangeelee's%20blog%20post) I went out and bought me a soldering iron, ordered the RFM01 and BMP085 modules on eBay, downloaded Kevin’s demo code and went to work.

The first frustration was that I had a good reception from the weather station, but couldn’t find any useful data. Well, as it turned out Kevin’s weather station uses 433MHz OOK transmission, while mine uses the 915MHz FSK protocol. That sorted out (with Kevin’s and many other’s help on the [Raspi forum](http://www.raspberrypi.org/phpBB3/viewtopic.php?f=37&t=14777)) I started to receive valid data streams from my station.

As I had to make major changes to Kevin’s program anyway, I played around a while with the wiringPI SPI library and finally settled with Mike McCauley’s bcm2835 library. You can download the latest version at [his web site](http://www.open.com.au/mikem/bcm2835/%5d). There are also instructions on how to install this library.

Note that this program is written for a 915 MHz RFM01 receiver module. You will have to change the rfm01\_init() function to reflect which frequency your weather station uses. It should also be very easy to make this program work with a RFM12 transceiver.

If you want to use a database and/or the BMP085 module, change the relevant lines at the top of the fo.h file. My first version only supported MySQL, but as it is probably a bit overkill for this application, I added support for Sqlite3. Also, I added a table/file for storing daily minimum/maximum values.

This latest version (using a Sqlite3 database) has been running 24/7 now for the last 2 weeks without a hitch, so I consider this project finished for me. Maybe, if I find myself with too much time at hand (highly unlikely), I will possibly add an .INI file parser and a FTP module. Also, I need the Raspi for a different project now …

I have written many million lines of code in the last 40+ years, starting with Assembler and Fortran, Cobol, Pascal … you name it. However, this is the very first program I’ve ever written in C (apart from the usual “Hello World” stuff). It was fun learning a new language along with some basic TX/RX stuff in this project.

To compile use:

* make DB=0 (output is stored in the files ‘fopi.dayfile’ and ‘fopi.details’)
* make DB=1 (output is stored in the ‘dayfile’ and ‘details’ tables of a MySQL database)
* make DB=2 (output is stored in the ‘dayfile’ and ‘details’ tables of a Sqlite3 database)

If you have any problems with this code or suggestions, feel free to send me a PM on this forum.

Cheers

Karl