Solar Wind Monitor
construction notes

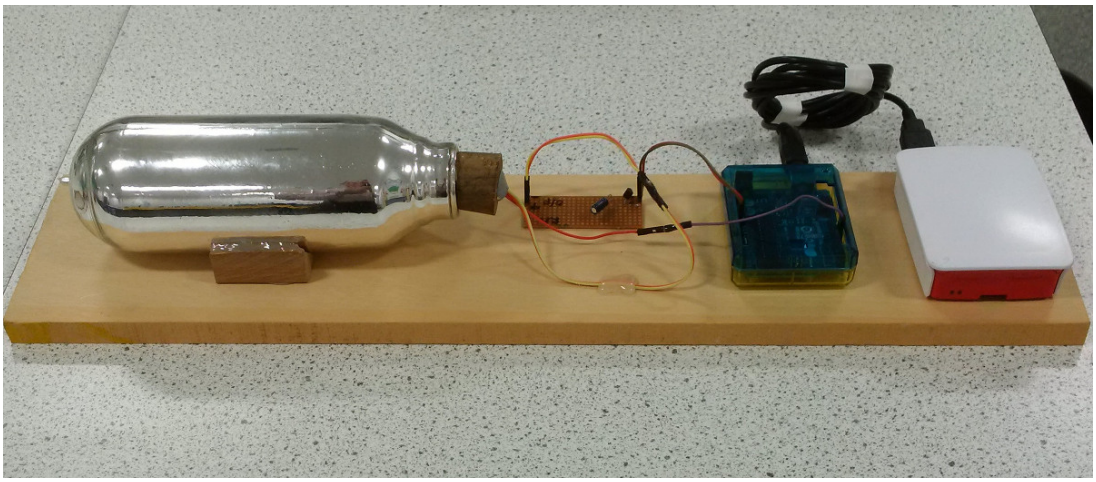
Ian Robinson

March 2, 2018

Abstract

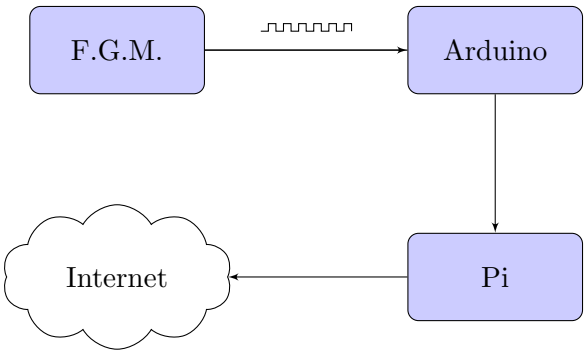
Described is the construction notes of an established geophysics project to build a solar wind monitor based on a nT resolution fluxgate magnetometer, these are detailed notes for the paper published in Physics Education (53,3) [1]. Low-cost and appropriate from school to university level it incorporates elements of astrophysics, geophysics, electronics, programming, computer networking and signal processing. The system monitors the earth's field in real-time uploading data and graphs to a website every few minutes. Modular design encourages construction and testing by teams of students as well as expansion and refinement. The system has been tested running unattended for months at a time. Both the hardware design and software is published open-source - [2].

1 Introduction



complete FGM rig with thermos housing

2 System Overview



3 Hardware

The system consists of a Speake or FG Fluxgate magnetometer sensor connected to an Arduino Uno R3 which counts the TTL pulses recieved in a set time. This frequency is output via USB to a PI (or PC) which logs the data in mseed format as well as producing plots which may be periodically uploaded to a website.

3.1 The Fluxgate Magnetometer Sensor

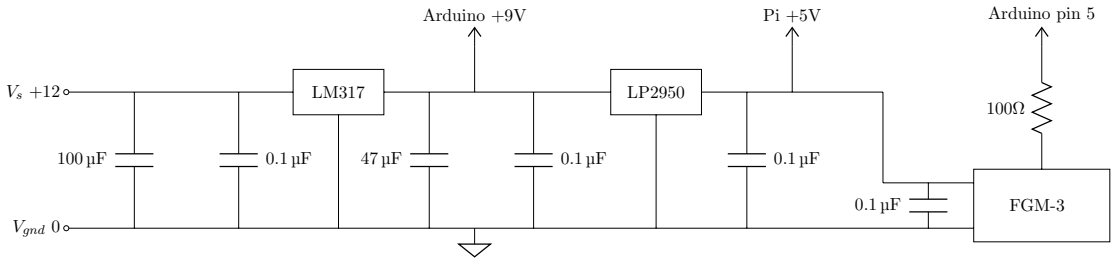
Developed in the 1930s fluxgate magnetometers work by alternately magnetising and demagnetising a core of high magnetitic susceptibility [3]. The presence of an external field produces a phase shift between the magnetising current and that induced in a pick-up coil. The F.G.M. is a vector sensor, detecting the field along its axis rather than the total field. This is useful here as the East-West component of the earth's field is the preferred indicator of aurora. They are widely used on spacecraft and in the geophysics community - in part due to their electronic simplicity and stability.

There are several suppliers of reasonably priced FGMs. I have built a number of systems using those sold for many years by Speake Sensors. Their application in earth field geomagnetism is well documented [4] will the sensors noted for high sensitivity at a low cost of £30-£40. Following the death of Bill Speake production has been continued by FGsensors in Slovenia [5]. Unlike the more expensive Mayer [6] devices which give a voltage output Speake sensors output a TTL square wave pulse train the frequency of which is proportional to the external magnetic field with a resolution of the order of nano-tesla. Such resolution with the Mayer devices would require an additional 16 bit resolution A.D.C.

The Speake/FG sensor has 4 pins, three of which are used gnd, +5V and TTL output. The output is connects to pin 5 of the arduino via a $\sim 100\Omega$ resistor. The resistor reduces the $\sim 5V$ output to a more suitable $\sim 3V$ for the Arduino.

3.2 Stabilised Power Supply

Initially one could simply used a stabilised lab p.s.u. to deliver approx 5V to the F.G.M. A better solution is building a small regulated supply to power the entire sysem. A simple double-regulated power supply constructed on veroboard is used to provide a stable voltage to the sensor as well as powering the Arduino and Raspberry Pi thereby ensuring a common ground.



4 Software

4.1 Arduino Frequency Counter: FreqCounter5.ino

The Speake FGM pulses at approx 20-120 kHz, too high a frequency to be reliably counted by the Pi. Rather an Arduino Uno runs a simple program to count the pulses over an interval of a few seconds - converting this to a frequency presented to the Pi/PC via a USB line.

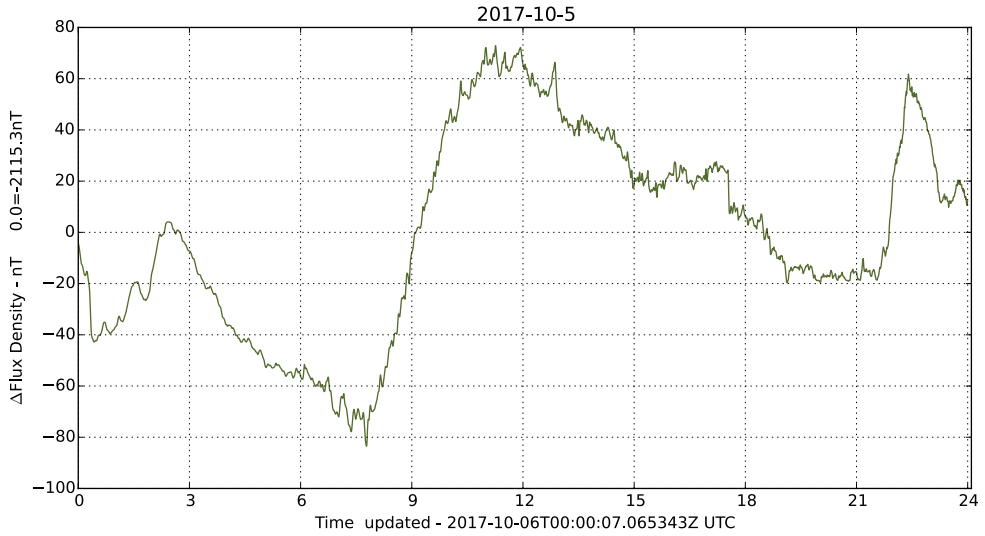
The code (github[2]) is simple to follow and employs the FreqCount.h library [7]. If you use the Arduino IDE freqcount.h may be automatically installed via the library manager. The arduino is connected to the PI via USB.

4.2 Raspberry PI / PC Logger & Plotter: EFM.py

The PI requires more setting up than the Arduino. This code has run successfully on an old PI3 though I recommend the newer 3B to reduce stuttering as the graphs are plotted. One will need to install:-

- Python 3
- MatplotLib [sudo apt-get install python3-matplotlib]
- Numpy [sudo apt-get install python3-numpy]
- NTP [sudo apt-get install ntp]
- SciPy [sudo apt-get install python-numpy python-scipy python-matplotlib ipython ipython-notebook python-pandas python-sympy python-nose]
- Obspy [sudo apt-get install python-obspy python3-obspy]

Keypoints: The program *EFM.py* reads a frequency from the USB line, currently set to every two seconds. Periodically the current dataset is saved in *.mseed* format into a subdirectory /Data/..Year... The file is named with that day's date. At intervals the



day's data is plotted using matplotlib output as *Year-Month-Day.svg* and a copy as *Today.svg* - to ease display on a remote website.

The python code has been kept simple to allow easy understanding and adaptation by students. On reading the frequency from the Arduino it converts this to a field in nT, performs simple signal conditioning and saves the data. The PI is connected to a local network via Ethernet (wireless would also be possible though less reliable). Periodically a simple script uploads the data and graphs to a website for viewing and further analysis. The PI used here is cheap, rugged, lacks moving parts and consumes only about 10W.

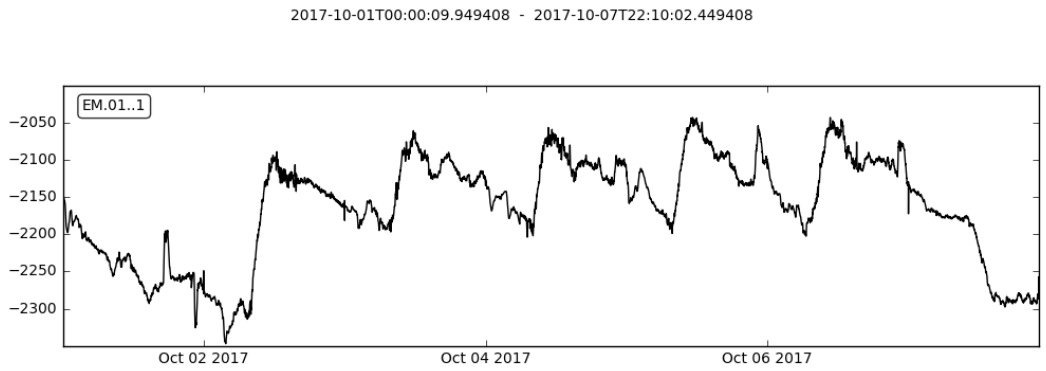
Datafiles may be read using Obs-Py [8], an opensource suite of programs used to plot and analyse seismic data. Obs-py is an excellent introduction to python programming, producing publication quality plots and signal processing techniques such as F.F.T.

4.3 Raspberry PI / PC Uploader

Two bash scripts *upload5min.sh and uploadDaily.sh* are setup to run using crontab, the former I run every 15min whilst the latter daily at 00:05. The *upload5min.sh* uploads only *Today.svg* to a remote website via ftp whil;st *uploadDaily.sh* uploads all datafiles. the latter needs improving so as to only upload changed files.

5 Siting the System

FGM sensors require siting in a thermally stable environment, preferably at least 20m from traffic. Nearby static metal is not a problem though proximity to strong e/m



5 day plot using Obspy

sources such as power limes should be minimised. Storage areas and loft space which may be suitable. A popular way of reducing thermal variation is to bury the sensor about 1m deep. I have constructed an enclosure using 2 inch water pipe - cement welded rather than pushfit [9]. This was successfully tested over 6 months in a buried location with 15m leads to the Arduino.

The sensor needs to be orientated magnetic E-W.

6 In Summary

For less than £100 one can have an ongoing 'real science' project delivering data 24/7 and displaying data via the web.

7 Addendum

As of 2nd march 2018 there exists a package problem with obspy running on a PI. This is being worked on, however if you get an error message of the form

```
    from .headers import (ENCODINGS, ENDIAN,
        FIXED_HEADER_ACTIVITY_FLAGS,
  File "/usr/lib/python3/dist-packages/obspy/io/mseed/headers.
    py", line 26, in <module>
    sizeoff_off_t = C.c_int.in_dll(__clibmseed, "
        LM_SIZEOF_OFF_T").value
ValueError: /usr/lib/python3/dist-packages/obspy/core/util
  /../../lib/libmseed_Linux_32bit_py34.cpython-34m.so:
  undefined symbol: LM_SIZEOF_OFF_T
```

you need to rename all the files in the

`/usr/lib/python3/dist-packages/obspy/lib/`

i.e.

```
sudo cp /usr/lib/python3/dist-packages/obspy/lib/
    libmseed_Linux_32bit_py34.cpython-34m-arm-linux-gnueabihf.so
     /usr/lib/python3/dist-packages/obspy/lib/
    libmseed_Linux_32bit_py34.cpython-34m.so
sudo cp /usr/lib/python3/dist-packages/obspy/lib/
    libtau_Linux_32bit_py34.cpython-34m-arm-linux-gnueabihf.so
    libtau_Linux_32bit_py34.cpython-34m.so
sudo cp /usr/lib/python3/dist-packages/obspy/lib/
    libtau_Linux_32bit_py34.cpython-34m-arm-linux-gnueabihf.so /
    usr/lib/python3/dist-packages/obspy/lib/
    libtau_Linux_32bit_py34.cpython-34m.so
sudo cp /usr/lib/python3/dist-packages/obspy/lib/
    libevresp_Linux_32bit_py34.cpython-34m-arm-linux-gnueabihf.
    so /usr/lib/python3/dist-packages/obspy/lib/
    libevresp_Linux_32bit_py34.cpython-34m.so
sudo cp /usr/lib/python3/dist-packages/obspy/lib/
    libsignal_Linux_32bit_py34.cpython-34m-arm-linux-gnueabihf.
    so /usr/lib/python3/dist-packages/obspy/lib/
    libsignal_Linux_32bit_py34.cpython-34m.so
sudo cp /usr/lib/python3/dist-packages/obspy/lib/
    libsegy_Linux_32bit_py34.cpython-34m-arm-linux-gnueabihf.so
    /usr/lib/python3/dist-packages/obspy/lib/
    libsegy_Linux_32bit_py34.cpython-34m.so
sudo cp /usr/lib/python3/dist-packages/obspy/lib/
    libevresp_Linux_32bit_py34.cpython-34m/usr/lib/python3/dist-
    packages/obspy/lib/libevr
sudo cp /usr/lib/python3/dist-packages/obspy/lib/
    libevresp_Linux_32bit_py34.cpython-34m /usr/lib/python3/dist
    -packages/obspy/lib/libevresp_Linux_32bit_py34.cpython-34m.
    so
```

References

[1] I. Robinson, "Solar wind monitor — a school geophysics project." `https://doi.org/10.1088/1361-6552/aaacb2`.

[2] I. Robinson, "Github: starfishprime101/aurora-monitor." `https://github.com/starfishprime101/Aurora-Monitor`.

[3] P. Ripka, "Review of fluxgate sensors," *Sensors and Actuators A: Physical*, vol. 33, no. 3, pp. 129 – 141, 1992.

[4] W. Reeve, "Fgm-3 and fgm-3h magnetometer sensors." http://www.reeve.com/FGMSensors.htm.

[5] FGSensors, "Commercial site." https://www.fgsensors.com/.

[6] S. Mayer, "Commercial site." http://www.stefan-mayer.com/en/products/magnetometers-and-sensors/fluxgate-sensor-fl1-100.html.

[7] P. Stoffregen, "Commercial site." https://www.pjrc.com/teensy/td_libs_FreqCount.html.

[8] obspy developers, "Obspy - an open source framework for seismology." https://github.com/obspy.

[9] I. Robinson, "Video - subsurface housing." https://www.youtube.com/watch?v=cAbXet-sfUU.