Aurora Monitor – Quick Start Guide

I feel it important for students to play with kit such as this to quickly develop an understanding of the system.

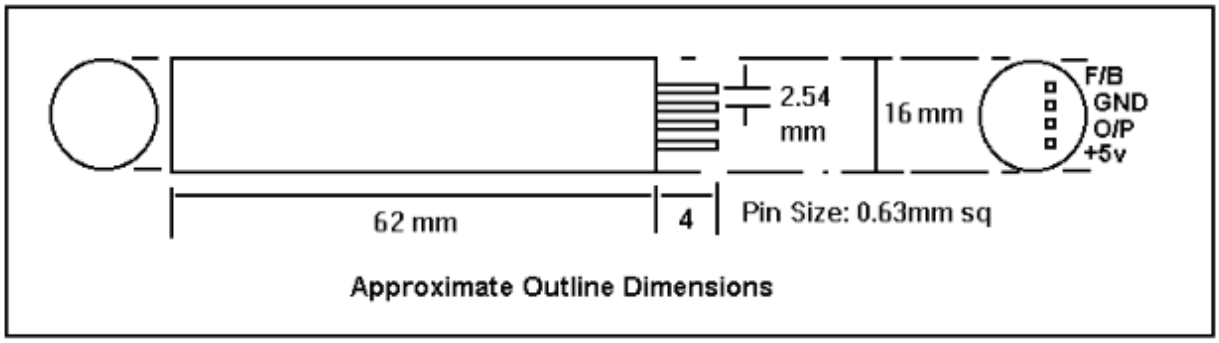
For a full system you will need..

- A Raspberry PI
- An Arduino Uno
- A powersupply – one can start with a lab psu

The FGM Sensor

I would start with becoming familiar with the sensor

- fg-3 sensor (fgsensors.com)
- Oscilloscope
- Lab P.S.U



The original Speake sensor sensor (above) has 4 pins – we only need ground (GND), output (O/P) and +5V. The newer fg-sensor has only GND, O/P and +5V

If you have a breadboard you should be able to plug the sensor into this for temporary connections. Failing that I would solder 3 leads of about 30cm to an 8 pin dip socket – having first checked that the sensor will fit, otherwise solder directly to the FGM pins taking care not to overheat it – one can use a metal clip or similar as a heat sink.

Then connect the gnd and +5V to a lab power supply. The sensor is not too fussy about input voltages 4.5-5.5V work. However the output does vary if the input voltage changes. For an aurora monitor you will need to construct a simple stabilised supply.

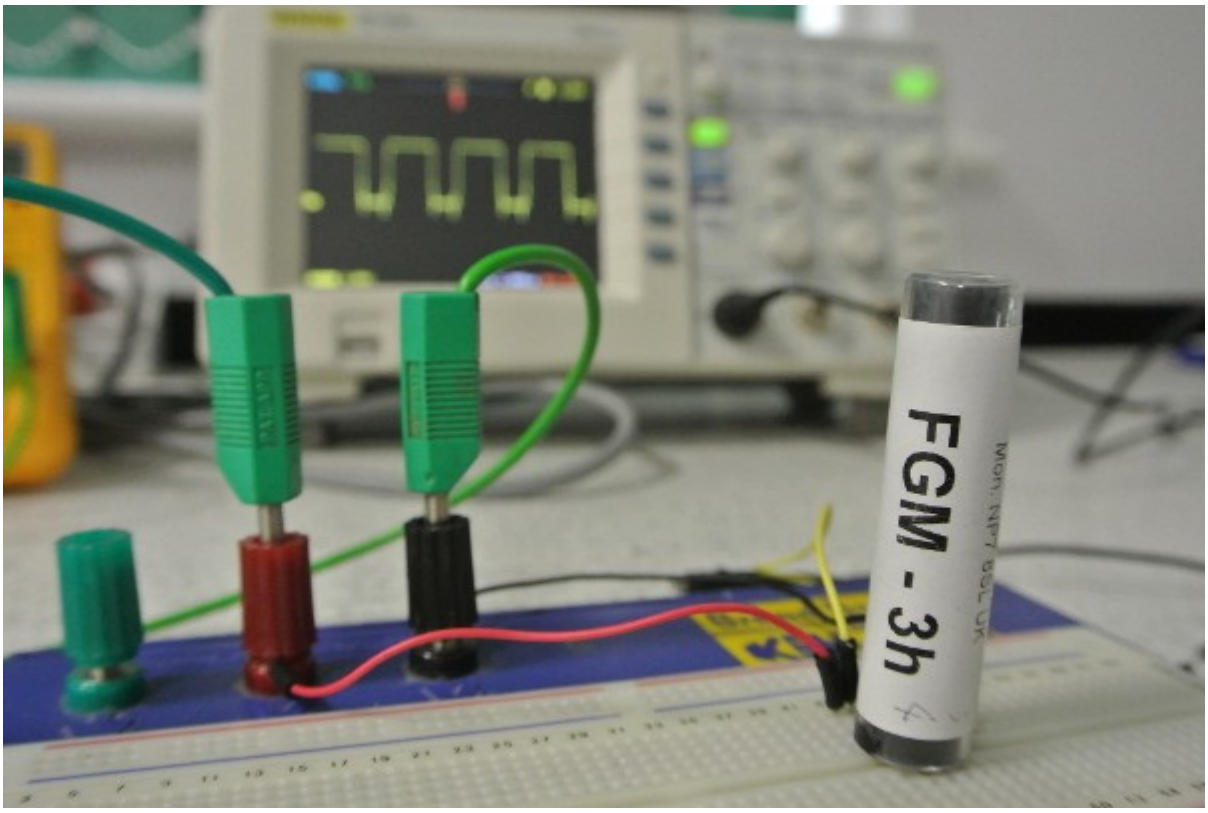
Now connect the input lead of an oscilloscope to the output – O/P pin.

Power up.

You should detect a roughly square wave TTL (0-5V) output at 20kHz →120kHz.

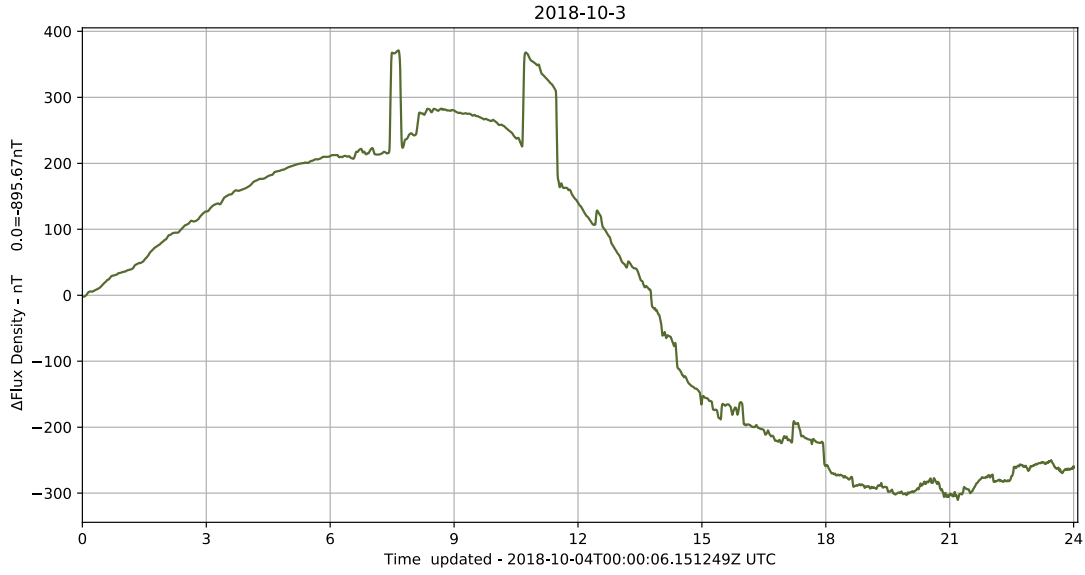
Orientate the sensor roughly East-West and you should be seeing an output of approx 70kHz.

Investigate (play). Moving a small piece of iron or even your own body in the vicinity of the sensor should cause an observable change in the output frequency.



Rotating the sensor we should see a dramatic change in output – probably saturating orientated North-South where the earth's field in strongest. Pointing the sensor north and changing the angle of dip one can also clearly see the declination of the earth's field.

This device really is very sensitive. At the surface the earth's field varies from 25,000-65,000 nT (nano Tesla). This device will reliably detect a change of about 1-2 nT.



The step changes at ~8 and~11 were caused by the moving of a car approx 12m from the sensor.

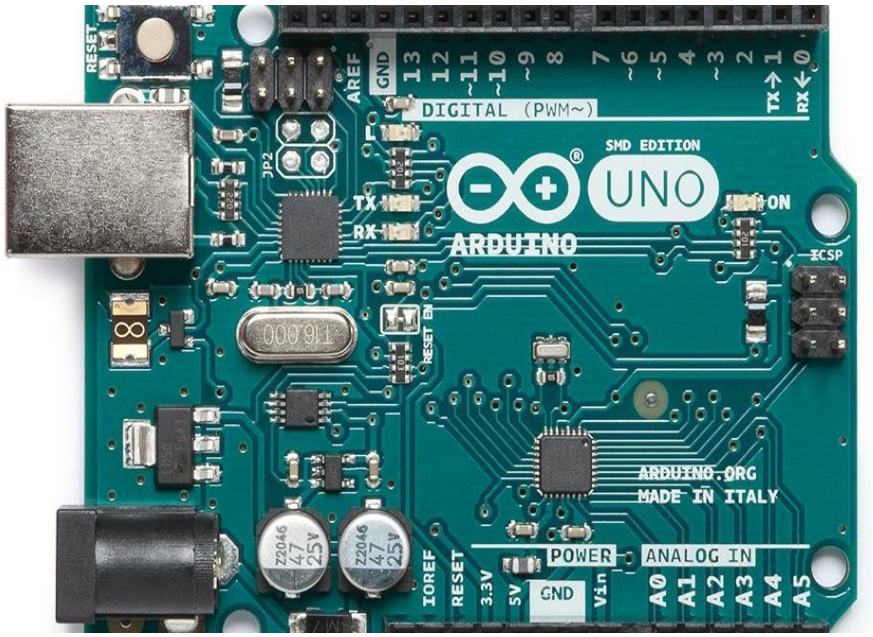
Next…

Measuring the Output Frequency

The sensor outputs a TTL (5V square wave) whose frequency varies approximately linearly with the external magnetic field. When orientated East-West, where the earth's field is lowest output is approximately 70kHz. This is too fast to be accurately counted by a raspberry Pi so we will use an Arduino (R3) to count the pulses and output the frequency via USB to a Pi.

Setting Up The Arduino

I would recommend using an Arduino Uno costing about £29. Cheaper options are available though.

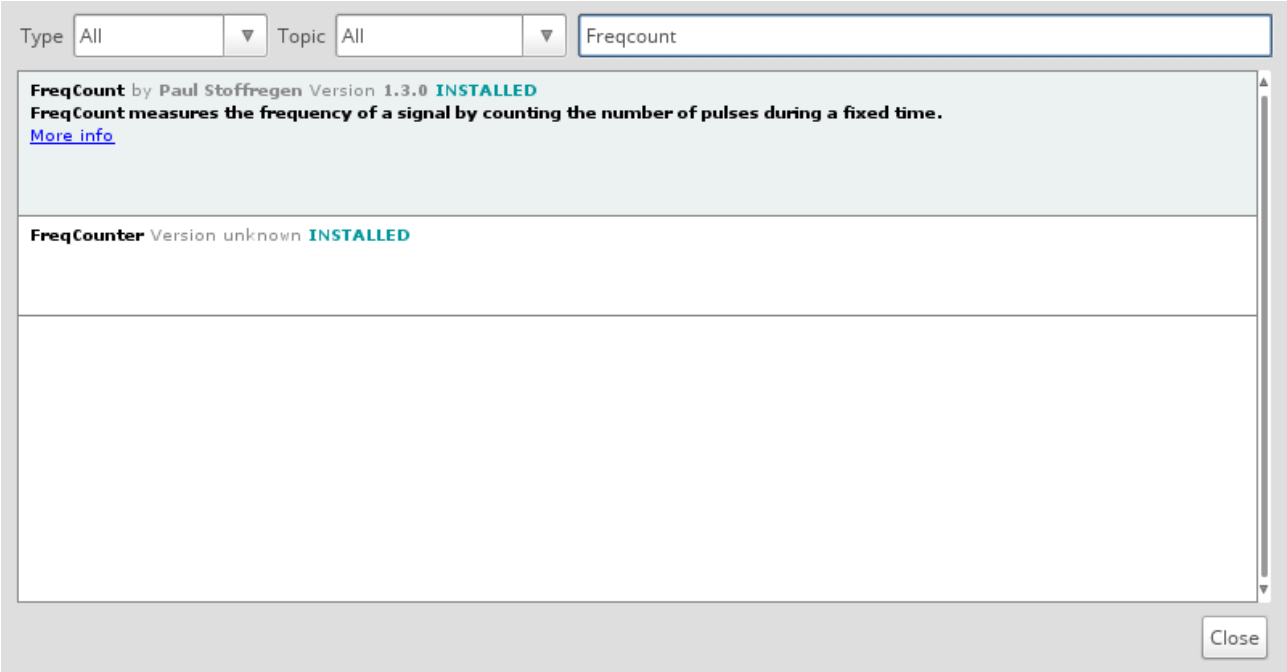


The Arduino is a 'bare metal processor'. Here it will run one simple program automatically when started up.

You will need a PC (Linux or Windows) with the Arduino IDE installed from https://www.arduino.cc/en/main/software

Firing up the IDE choose

Sketch → Include Library → Manage Libraries ->



Ensure that the Freqcount library is installed - the ide can pull it from the net if not.

Then do a File ..new and paste the following code in.

```
/* FreqCount - Example with serial output
 * http://www.pjrc.com/teensy/td_libs_FreqCount.html
 * This example code is in the public domain.
 */
#include <FreqCount.h>
void setup() {
  Serial.begin(57600);
  FreqCount.begin(1000);/* gatetime in mSeconds*/
}
void loop() {
  if (FreqCount.available()) {
    unsigned long count = FreqCount.read();
    Serial.println(count);
  }
}
```

This repeatedly counts the number of pulses over 1 sec (1000 msecs) and presents this frequency to the seial output (USB line) . This we can read with the PI.

Connect the Arduino to a USB slot on the PC and within the Arduino IDE select Sketch .. Upload to compile the code and upload to the Arduino where it should start running immediately. If all has worked (probably hasn't) selecting Tools .. Serial Monitor should show the output from the Arduino.

Setting Up The Pi

Right – this can be involved but we learn a lot about computing here.

We are going to use a Raspberry PI to save data, plot graphs and upload to your website.

There are two ways we can 'talk' to the PI – either using a screen, keyboard and mouse – like a conventional computer or from a PC over a network – a.k.a 'headless mode'. A monitor and keyboard is handy in a schools lab but for a proper network install we are better with headless.

So the first thing to do is get the PI running. I will assume that you have connected to a hdmi monitor or TV and have a keyboard and mouse attached.

Firstly you will need to install the operating system. This will be Raspian – a version of Linux.

Raspbian Stretch Lite from

https://www.raspberrypi.org/downloads/raspbian/

headed (i.e. with monitor and keyboard) installation instructions at

https://www.raspberrypi.org/documentation/installation/installing-images/README.md

Headless installation instructions at

https://hackernoon.com/raspberry-pi-headless-install-462ccabd75d0

Hopefully you now have a working PI.

Next we need to install Obspy, a suite of seismic analysis software

Installing ObsPy

To use the MiniSeed data format format, the best way is to use a library made for this: ObsPy. So we must first install it. You can use a notepad editor in root, e.g. from terminal, as long as you have an Internet connection on your Raspberry Pi.

sudo nano /etc/apt/sources.list

Add the following to the end of this sources file (the repository to the ObsPy Libraries)

deb http://deb.obspy.org stretch main

Installing Required Software

Using a terminal run each of the following commands

```
sudo raspi-config
```

(enable i2c)

```
sudo apt-get install python3
```

```
wget --quiet -O - https://raw.github.com/obspy/obspy/master/misc/debian/public.key | sudo apt-key add -
```

```
sudo apt-get update
```

```
sudo apt-get install python3-obspy
```

```
sudo apt-get install python3-smbus
```

```
sudo apt-get install python3-serial
```

```
sudo apt-get install python3-matplotlib
```

Ensure that the PI knows the correct time

Install ntp time

```
sudo apt-get install ntpdate
sudo timedatectl set-ntp True
```

If you set the Time Zone in raspi-config the Raspberry Pi will automatically update the time on boot if connected to the internet.

```
sudo raspi-config
Select Internationalisation (Localisation) Options
Select I2 Change Timezone
Select your Geographical Area
Select your nearest City
Select Finish
Select Yes to reboot now
```

The Monitoring Programs

There are 3 python programs used

- aurora_monitor.py
- geostationModules.py
- globalvariablesModule.py

These should be copied into a folder in your home-area … I use /home/pi/EFM

The 3rd file, globalvariablesModule.py may need editing for your system. Especially Line 5 - home_directory = '/home/GeoPhysics/EFM'.

lines 14 and 26 may want minor editing.

Set Up CronTab to automatically start the Aurora Monitor on reboot

crontab -e (If given a choice of editors I would select 2- nano)

copy the following to the bottom of the file

```
# m h  dom mon dow   command
03 0 * * * /home/pi/EFM/uploadDaily.sh 2> /home/pi/EFM/uploadDailyerrors.txt
*/15 * * * * /home/pi/EFM/uploadHourly.sh 2> /home/pi/EFM/uploadHourlyerrors.txt

@reboot python3 /home/pi/EFM/aurora_monitor.py 2> /home/pi/EFM/errors.txt &
```

Replacing /EFM/with the name of the directory containing the monitor program

Exit with *CRTL o* then *CTRL x*

Install FTP to upload plots to your web-server

sudo apt-get install ftp

Adding a real Time Clock

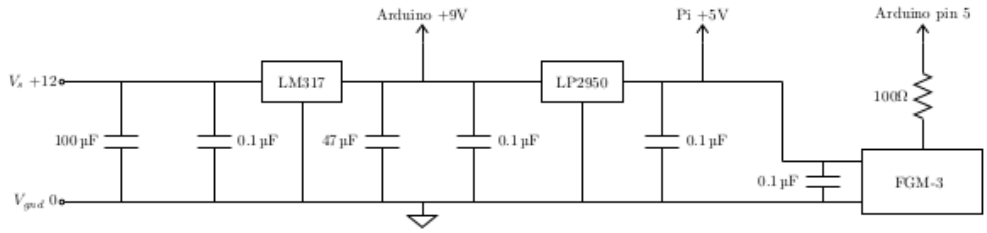
In normal use the PI gets its time signal from Internet. It lacks an internal clock so cannot add correct time to a trace if it is not connected to the internet.

Adding a precise clock module is thus desirable only if you intend to use the sensor away from an Internet connection.

I use the DS3231 Precision R.T.C. from AdaFruit following instructions at https://pimylifeup.com/raspberry-pi-rtc/

A Precision Power Supply

A simple double-regulated power supply constructed on veroboard may be used to provide a stable voltage to the sensor as well as powering the Arduino and Raspberry Pi thereby ensuring a common ground.



Uploading to a Website

Grab the file uploadhourly.sh from the github site or create a file with the same name in the aurora monitor directory on the PI (assumed to be /home/pi/EFM)

```
#!/bin/bash
HOST='yourwebsite.co.uk'
USER='your_webserver_ftp_username'
PASSWD='yourPassword'
ftp -p -n -v $HOST << EOT
ascii
user $USER $PASSWD
prompt
cd  StarFishPrime/projects/EFM/plots remote directory for files
ls -la
```

put /home/pi/EFM/Plots/Today.png Today.png transfer the current day's plot

bye

EOT

you should also create file uploaddaily.sh

HOST='yourwebsite.co.uk'

USER='your_webserver_ftp_username'

PASSWD='yourPassword'

ftp -p -n -v $HOST << EOT

ascii

ftp -p -n -v $HOST << EOT

ascii

user $USER $PASSWD

prompt

cd StarFishPrime/projects/EFM/plots

lcd /home/pi/EFM/Plots

ls -la

mput *.*

bye

EOT