Infrasound Monitor – getting Started

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

We are going to use a Raspberry PI to record the frequency, 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*

CronTab

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
*/60 * * * * /home/pi/InfraSound/uploadHourly.sh 2> /home/pi/InfraSound/errorHourly.txt
05 0 * * * /home/pi/InfraSound/uploadDaily.sh  2> /home/pi/InfraSound/errorDaily.txt
@reboot python3 /home/pi/InfraSound/InfraSoundMonitor.py 2> /home/pi/InfraSound/errors.txt &
```

Replacing InfraSound/InfraSoundMonitor.py 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/

***explain about voltage level shifrt*