Raspberry PI Seismic Monitor Code

Sept 2020

Ian Robinson   www.starfishprime.co.uk

Key Features

- OpenSource.
- Easily configurable to store data from a variety of sensor inputs.
- Running unattended on multiple devices 24/7 for over 3 years with zero crashes.
- Produces regular data-plots for automatic uploading to a webpage.
- Creates subdirectories for plots and data
- Data is stored in miniSEED format permitting reading and analysis by standard seismic analysis software such as ObsPy.
- Hourly timestamps for accuracy.
- Recovers from power outages

Overview

This code evolved out of Raspberry PI projects to monitor solar wind, solar flares and infrasound. It runs successfully on RPi 3b and 4 and can be left unattended for months. A simple separate crontab script periodically (e.g. hourly) uploads the plots to a website. Should further analysis be required data-files from the Rpi are downloaded for detailed examination with standard seismic software such as ObsPy on a PC. An example of this ObsPy analysis code is also included.

The RPi code is written in python3 and invokes the ObsPy seismic data handling library.

Execution structure

1. Program startup – check for *Data* and *Plots* subdirectories – create if not present.
2. Record start time (UTC) of sample block.
3. Interrogate input for first data value.
4. Save reading in data arrays *HourlyReadings* and *DailyReadings*
5. Sleep for $^{1}/_{\text{(sampling frequency)}}$
6. Check whether hourly save required
7. Repeat

When hour changes (e.g. 21:00 → 22:00) the *save&plot* subroutine is invoked in multi-threading mode (to reduce interruption to sensor reads).

The *save&plot* subroutine is passed the old data array along with sampling start and end times. It calculates the actually sampling frequency over that one hour period saving the hourly datafile *<day>.mseed*. Appropriate SEED headers such as station ID, network etc. are written into the file-header with the data stored in binary format. Day, Month and Year changes also handled here.

The plot routine plots an obspy *dayplot in .svg* format. This is saved as *Today.svg* and copied to <date>.svg. When uploaded to a website this means that the page can display *Today.svg* as the most recent plot[1].

File Save Structure

```
Data/
    -----Year/ (2020)
         -----Month/ (03)
              ----Day/ (27)
                  <Hour1>.mseed, <Hour2>.mseed …..  (01.mseed,02.mseed, 03.mseed...)
```

```
Plots/
    -----Year/ (2020)
         ---<date>.svg , <date>.svg , <date>.svg
```

Notes

- By default readings stored as 32bit floats, change this at *st.write(dataFile,format='MSEED', encoding=4, reclen=4096)* (see SEED manual for explanation of *encoding*=)
- code plots both raw data values and acoustic power (latter can be commented out if not required)
- No signal conditioning is performed on raw data. It is assumed that this will be performed during detailed analysis on PC.
- Two sets of data are recorded. Data is temporarily stored in two arrays *DailyReadings* and *HourlyReadings.* A 24hr set is used for the plots with the start-time (typically midnight). The hourly array is the one saved to data-files with a new start-time for each block This increases the accuracy of the timings of the saved data by a mean factor of 12 over the dailyplots.
- An ObsPy program has been written to read in either single (hourl) file or an entire day's directory and demonstrate some simply plotting and analysis.

1 Example see https://www.starfishprime.co.uk/projects/infrasound/infrasound.html