

# Earthquake Pi

2016-06-17 R.Grokkett

version 1.3



## Overview

There are a number of earthquake detector projects for the raspberry pi. These are good for anyone living in an area prone to earthquakes or for those that want to try to detect distant quakes themselves.

This project is not focused on detecting quakes, but uses USGS earthquake data to make an interesting little alerting system. Sure, you can always go to your PC and browse the USGS maps for the latest data, but you are a Maker and you like to make things because you can!

## What does it do?

- It collects the USGS earthquake data from their website for the past 15 minutes.
- It displays location, magnitude and other data on a small LCD screen
- It rattles its box and flashes its lights to the magnitude of the quake (Optional)
- It plays earthquake sounds (Optional)

You set the minimum magnitude level you wish it to alarm on. But note, if you set to 1.0, it will probably go off every few hours! At 3.0, it will usually sound once every day or two.

Optionally, you can add motor rattles, audio sounds and LED lights. These are not needed, but really make the earthquakePi “alive”.

You can use other GPIO pins for controlling other devices. Even interface it with cloud Internet of Things (IOT) services such as IFTTT (<http://ifttt.com/>)

**NOTE:** With all these options active, this becomes a fairly elaborate wiring & software job and requires you to jump to other setup tutorials as noted in URL links. If you are not familiar with Linux command line and/or wiring GPIO pins, you may wish to try simpler projects first, or skip one or more of the optional features in this project.

## Hardware Parts

- Raspberry Pi running Raspian and python 2.  
Any Pi will do. Even a \$5 Pi Zero
- Ethernet connector or WiFi dongle for Raspberry.
- A I2C compatible LCD display **20 char x 4 line**  
such as Amazon <http://amzn.com/B00SMM6C64> 2004 I2C LCD or similar
- Breadboard, Pi Breakout cable/connector to breadboard, wire
- A wooden box (such as from craft stores like Michaels)
- 5v power supply for Raspi (2 amp if using all options)

## Options:

### *Vibration Motor:*

- A vibrating motor such as  
From an old battery toothbrush (see below)  
From a pager  
or buy one similar to <https://www.adafruit.com/products/1201>
- 220 ohm resistor
- 1N4001 or similar diode
- 2N2222 transistor

### *NeoPixel Bargraph:*

- Neopixels LED strip 8x5050 RGB LED NeoPixel Stick from Adafruit:  
<https://www.adafruit.com/products/1426>
- 1N4001 or similar diode

### *Audio:*

- External speaker (and amp) for Raspi audio.
- For Raspi Zero, add Adafruit I2S 3W Class D Amplifier  
<https://www.adafruit.com/products/3006> and speaker

## Hardware Installation

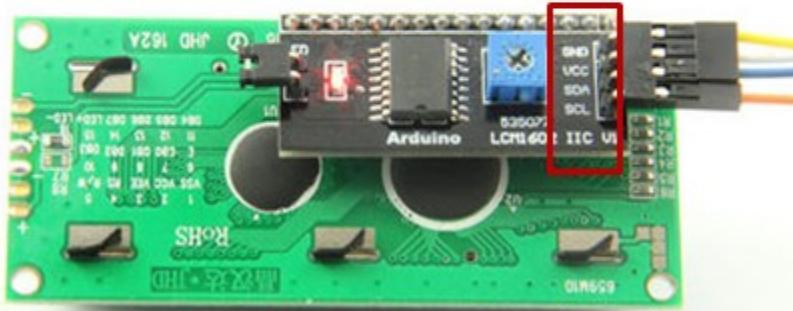
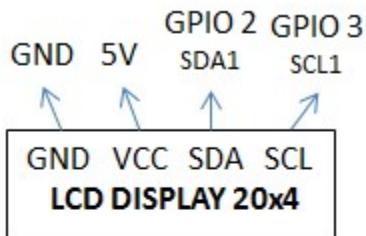
**NOTE:** You should initially set up your Raspberry Pi with Raspbian. I used Jessie version, but others should work. This project assumes you can already SSH to the Raspi and connect to the Internet before adding the features below. No HDTV monitor or keyboard is needed (headless operation).

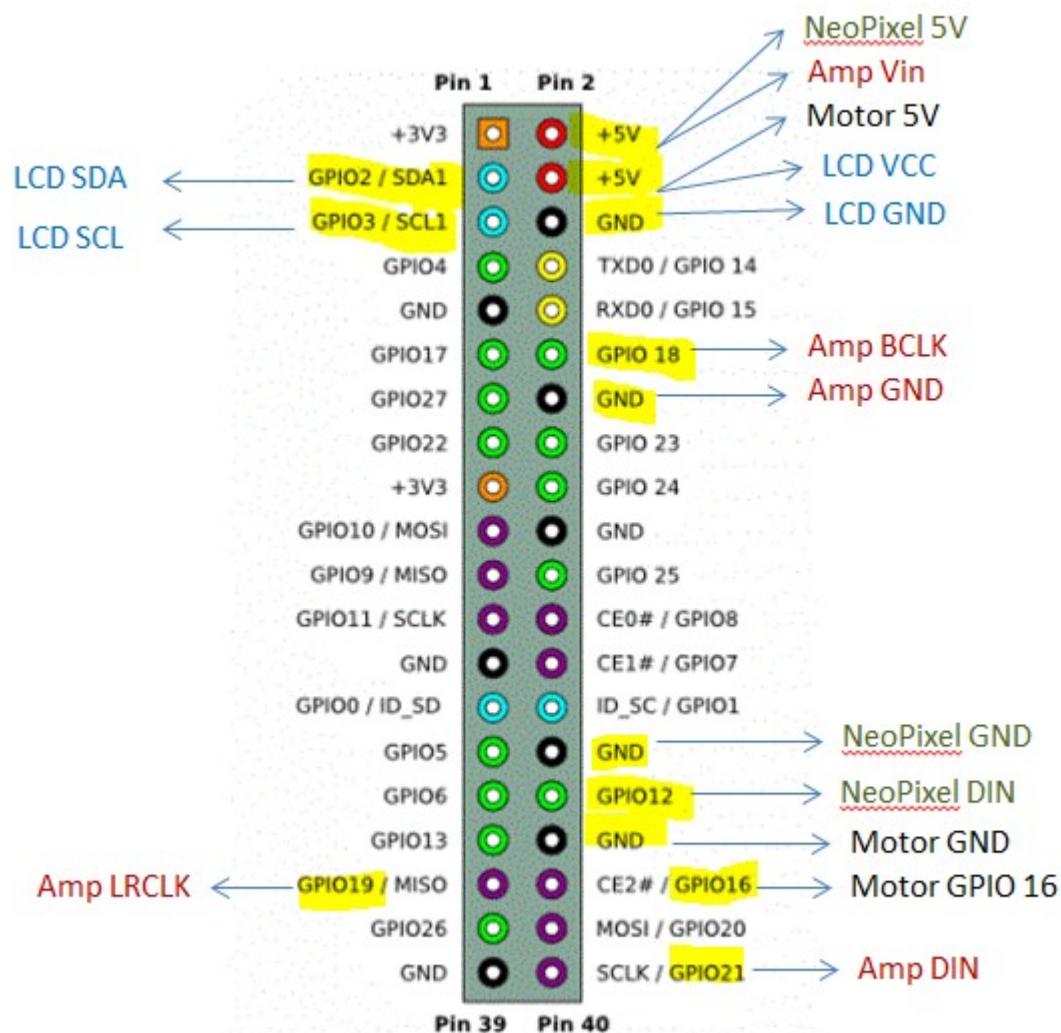
### Internet

As stated above, you need Internet connectivity, either via Ethernet or WiFi. Nothing special is needed for this project (no firewall rule changes, etc.) so just follow existing instructions elsewhere for setting up your Pi.

### LCD 20x4 Display

For the LCD 20x4 Display, wire the Pi up as shown below.





**Figure showing pinouts with all options**

### Optional Vibrating Motor

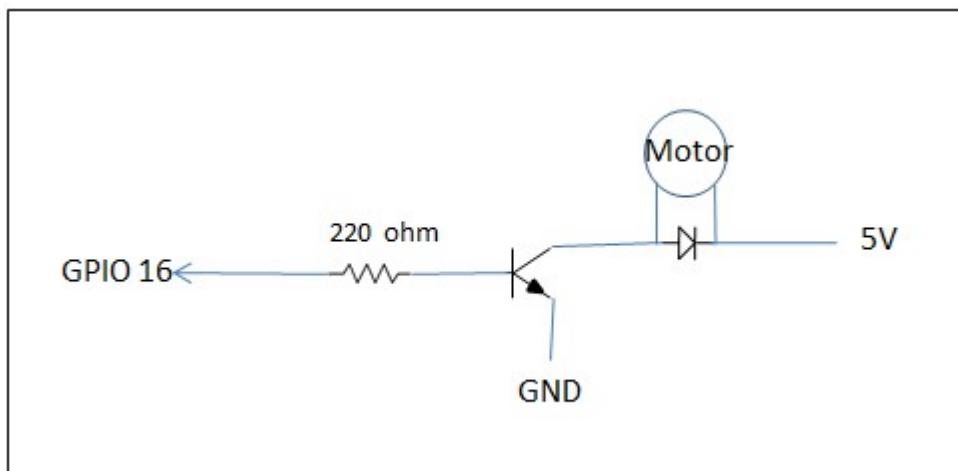
For the optional vibrating motor, I used an old toothbrush such as the Oral B battery powered brush. Open it up and you will find a tiny “vibrator” motor. Alternately, pick up a vibrating motor from numerous sources.



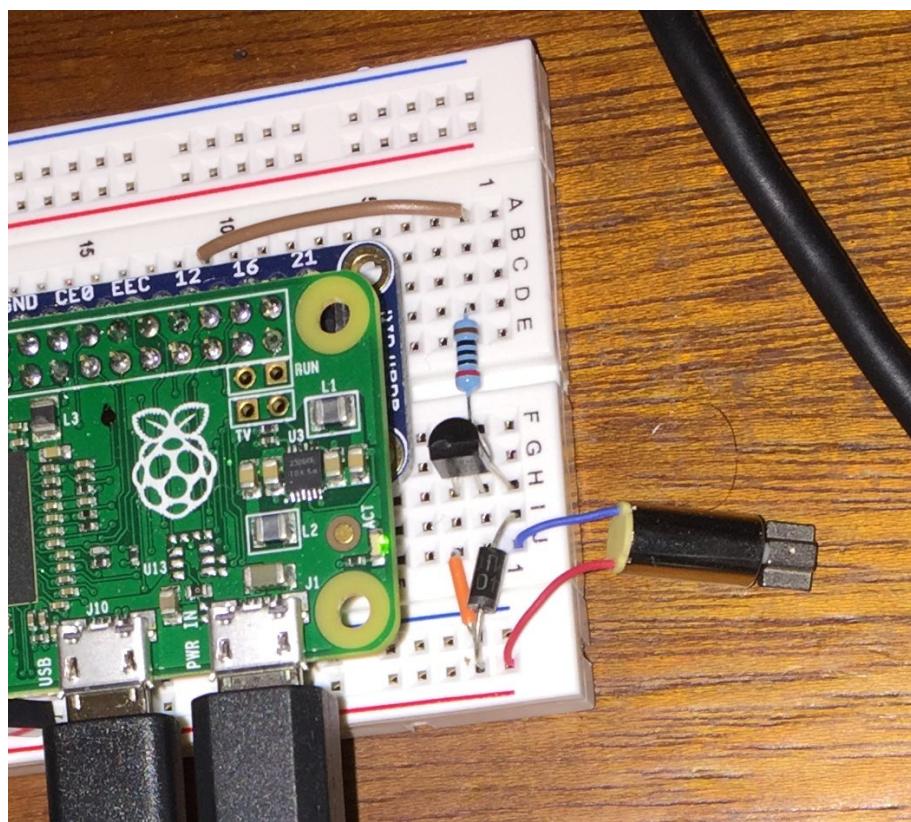
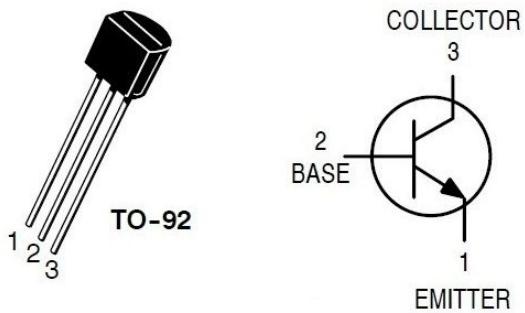
Note that you may need to adjust the settings in the python earthquake.py program to get the “best” vibration desired. The idea is to have it crescendo quickly and then fade over several seconds, with the volume and length determined by the magnitude of the quake.

You can use either the 5v or 3.3v supply pins on the Pi to run the motor. Most will run ok either way. But you will need to add a simple driver as most motors take too much current to run directly off a GPIO pin, thus the 2N2222 transistor. The 1N4001 diode is needed to keep motor induction spikes off the Raspi.

Connect it to **GPIO pin 16** (default, but can be changed in the python program) as well as a **Ground pin** and a **5V pin** as shown.



## 2N2222



## Optional Audio Earthquake Sound Effect

For audio, plug an amplifier into the audio jack. The default on the Pi is set to send audio automatically to the analog output, but if you have an HDMI monitor plugged in, the sound shifts to the HDMI connector. Since this project doesn't need a monitor, unplug it and reboot. Alternately, use:

```
$ sudo raspi-config
```

Select Advanced Options -> Audio and select Analog out.

On a Pi Zero, you will need to add external audio, since it's not built-in. I used the low cost "Adafruit I2S 3W Amplifier breakout MAX98357A" <https://www.adafruit.com/products/3006>

This requires setup using Adafruit's excellent tutorial at:

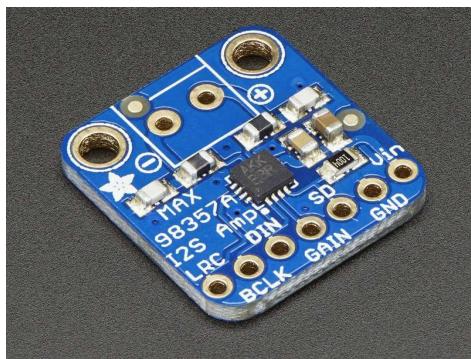
<https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/raspberry-pi-usage>

As shown in the tutorial, connect:

- **Amp Vin** to Raspi Zero Pi **5V**
- **Amp GND** to Raspi Zero Pi **GND**
- **Amp DIN** to Raspi Zero Pi **#21**
- **Amp BCLK** to Raspi Zero Pi **#18**
- **Amp LRCLK** to Raspi Zero Pi **#19**

Be sure to edit the files shown in the tutorial and test the audio.

Once you complete the setup tutorial including the audio tests, then sound should be ready.



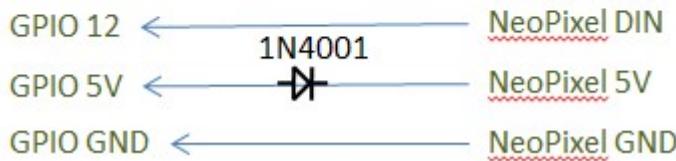
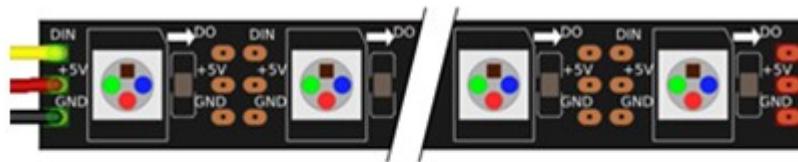
## Optional NeoPixel LED Bargraph

NOTE: The earthquakepi.py program is designed to only run the 8x5050 RGB LED NeoPixel Stick from Adafruit: <https://www.adafruit.com/products/1426>

Wire the Adafruit NeoPixel as described in the NeoPixels on Raspberry tutorial

<https://learn.adafruit.com/neopixels-on-raspberry-pi/wiring>

I was able to use the simpler “diode wiring” method, described in the tutorial, since I used only an 8 RGB LED strip.



For the NEOPIXELS LEDs option, you must install the `rpi_ws281x` Library as detailed in:  
<https://learn.adafruit.com/neopixels-on-raspberry-pi/software>

**Note!**: The default GPIO Pin 18 in the Neopixels example is DIFFERENT from the one used by `earthquake.py`!

Edit `~/rpi_ws281x/python/examples/strandtest.py` and change `LED_PIN = 18` to `LED_PIN = 12`

Wire the NeoPixel DIN to GPIO 12.

The GPIO 18 is needed for Raspi Zero sound.

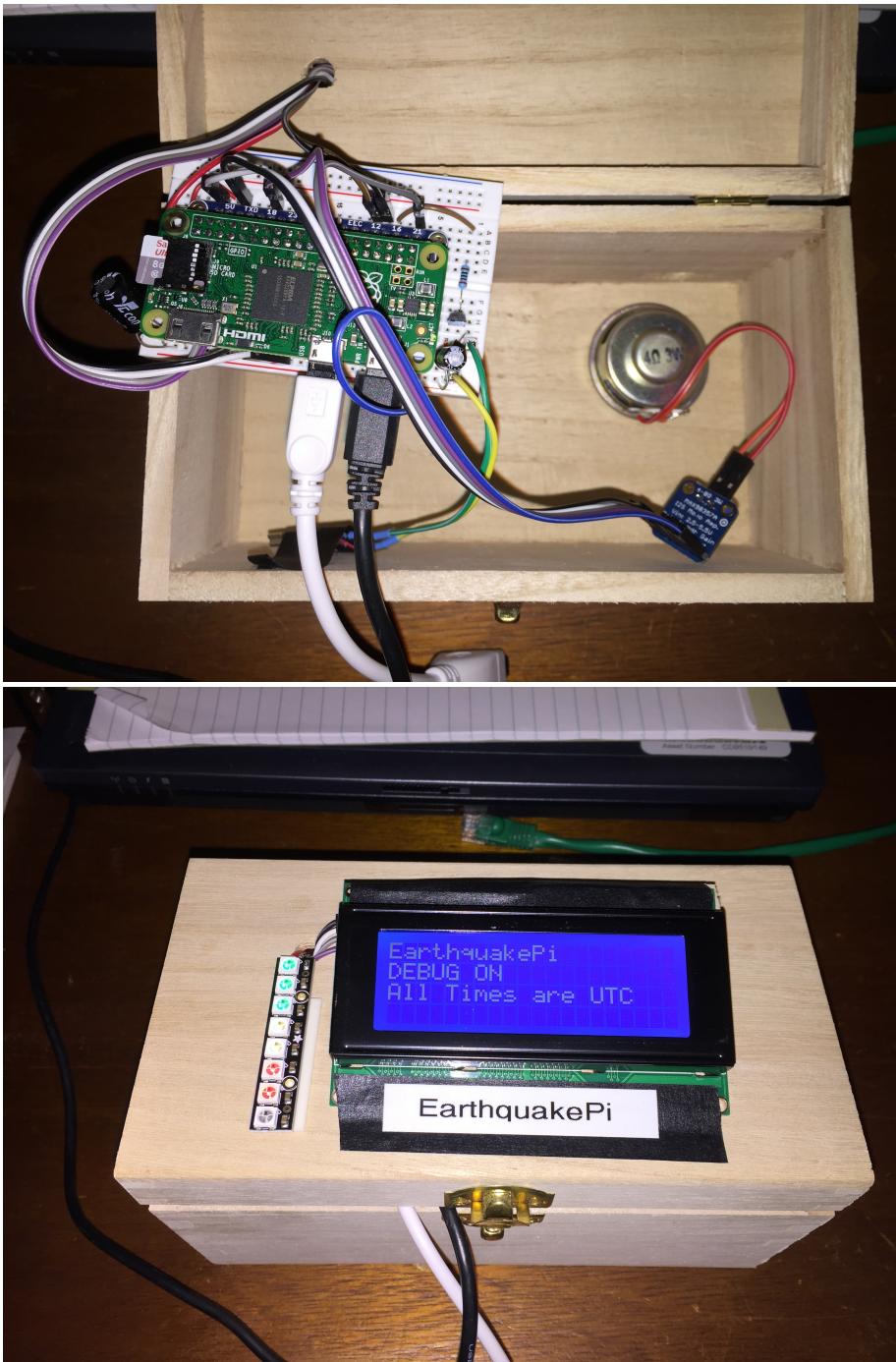
Be sure you can successfully run the “`strandtest.py`” example in the Neopixels tutorial before continuing with EarthquakePi.

### POWER REQUIREMENTS:

All the options combined can use close to the limit of available current from the Raspberry so if you use different hardware, be mindful to keep total current under 500ma from the Raspi 5V lines.

### Wooden Box

Install the system into a box. I used a small 4x7x3 wooden box from Michael’s Arts & Crafts. Cut holes for the leads. Hot melt glue the motor down. I used electrical tape with a small piece of wood to angle the LCD display and hold it down on the top of the box. I used double stick foam tape for the RGB LED strip beside the LCD display.



## Software

Raspian Linux with Python 2.X

Software from GITHUB: <https://github.com/rgrokett/earthquakepi>

## Software Installation

This assumes you already have your Pi set up with Raspian, python and connected to the Internet.

Install packages:

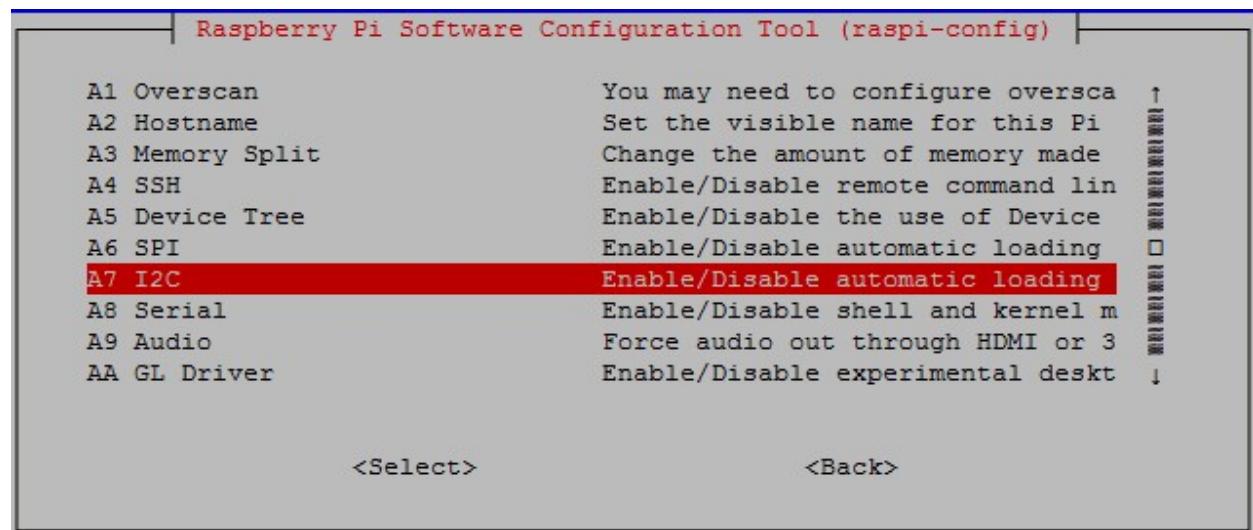
```
$ cd /home/pi  
$ sudo apt-get update  
$ sudo apt-get install build-essential git  
$ sudo apt-get install python-dev python-smbus python-pip  
$ sudo pip install RPi.GPIO  
$ sudo apt-get install i2c-tools
```

Configure the SPI and I2C bus devices by using:

```
$ sudo raspi-config
```

Select Advanced Options -> **I2C enable**.

Select **YES** to enable and **YES** to load by default for both of these.



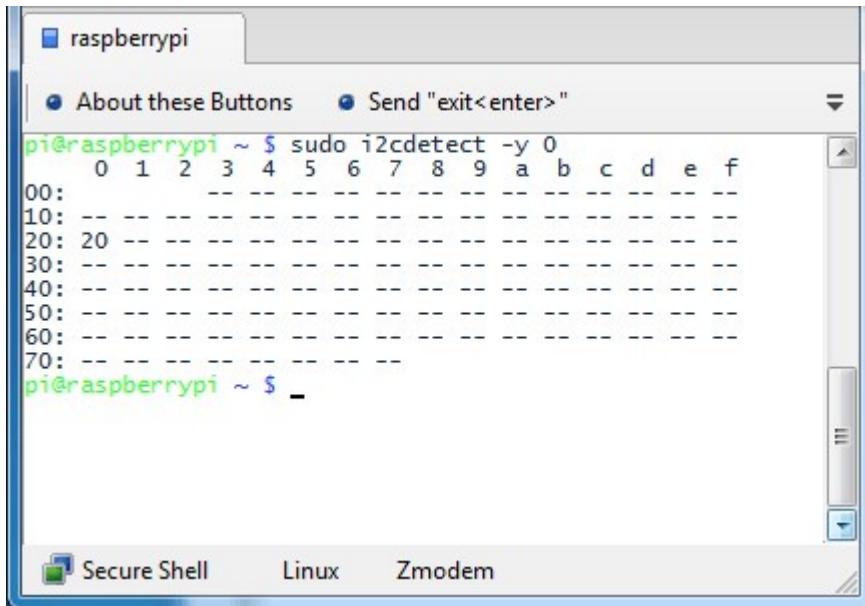
Select FINISH to Reboot your Pi, if asked, or execute:

```
$ sudo reboot
```

Verify the LCD is detected using:

```
$ sudo i2cdetect -y 1
```

You should see a screen similar to below. If the LCD display is detected, you will see an entry such as **0x20** for Adafruit LCD or **0X27** for Keyestudio LCD.



```
pi@raspberrypi ~ $ sudo i2cdetect -y 0
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -----
10: --
20: 20 --
30: -----
40: -----
50: -----
60: -----
70: -----
```

Next install and test the earthquake program:

1. \$ cd /home/pi
2. \$ git clone https://github.com/rgrokett/earthquakepi.git

Test your LCD Display first. This works with 16x2 or 20x4 LCD displays, but the Earthquake program expects 20x4 only, as it needs the extra lines of text.

1. \$ cd ~/earthquakepi
2. \$ nano RPi\_I2C\_driver.py  
Edit the LCD address to match what you found above.  
# LCD Address  
ADDRESS = 0x27
3. \$ python lcd.py

The test program will turn on the LCD backlight and then display a series of text and graphics:



If you have a problem, check your wiring, including adjusting the tiny potentiometer on the back of the LCD display for brightness.

### Audio Software & Test

If you use audio, set the maximum volume on your speaker using:

```
$ sudo amixer sset PCM,0 95%
$ aplay /home/pi/earthquakepi/earthquake.wav
```

*NOTE: Pi Zero & Adafruit MAX DAC doesn't use **amixer** volume controls and can only handle stereo WAV files (Earthquake.wav has been converted to stereo for compatibility.)*

Adjust your speaker/amplifier to make this LOUD as this is equivalent to a magnitude 9.0 earthquake! The program will range from barely audible 1.0 quakes upwards to your maximum volume you set.

### Test it!

Once you get the expected results, next is to try out the earthquake program.

Edit the earthquakepi.py file and change any optional features you have installed:

```
DEBUG = 1
MINMAG = 1.0
AUDIO = 0 or 1
MOTOR = 0 or 1
NEOPIXEL = 0 or 1
```

Then type:

```
$ sudo python earthquake.py
```

An initial DEBUG mode test of the display along with optional LED bargraph, motor and audio should occur. If there is an error or missing software package, its information should print out on your SSH terminal.

## Operating Earthquake Pi

The program variable MINMAG defaults to magnitude 1.0 or greater which occurs many times per day. You can edit the earthquake.py program to change this higher.

Set MINMAG to 4.0 to limit to larger earthquakes only, which is typically every day or two.

```
##### USER VARIABLES
DEBUG      = 1          # Debug 0 off, 1 on
LOG        = 1          # Log Earthquake data for past 15 min
MINMAG     = 1.0        # Minimum magnitude to alert on
AUDIO      = 1          # Sound 0 off, 1 on
MOTOR      = 1          # Vibrate Motor 0 off, 1 on
MOTORPIN   = 16         # GPIO Pin for PWM motor control
NEOPIXEL   = 1          # 1 use Neopixel, 0 don't use Neopixel
NEO_BRIGHTNESS = 64    # Set to 0 for darkest and 255 for brightest
## OTHER SETTINGS
PAUSE      = 60         # Display each Earthquake for X seconds
WAV = "/home/pi/earthquakepi/earthquake.wav"  # Path to Sound file
DISPLAY    = 0          # 0 Turn off LCD at exit, 1 Leave LCD on after exit
##### END OF USER VARIABLES
```

Once tuned, turn off DEBUG mode by editing the program:

```
DEBUG = 0      debug off
MINMAG = 2.0 or higher  quake every few hours.
```

LOG = 1 prints USGS earthquake data (if any) for the past 15 minutes.

The log will be written to /home/pi/earthquakepi/earth.log by the cron, below.

Load CRON entries to run the program every 15 minutes between 8am and 11pm daily. This way it will only run during waking hours (adjust as desired!).

```
$ crontab pi.cron
```

The @reboot cron just displays a message for a few seconds anytime Raspi is rebooted.

```
@reboot sudo python /home/pi/earthquakepi/startup.py >/dev/null 2>&1
```

```
0,15,30,45 08-22 * * * sudo python  
/home/pi/earthquakepi/earthquakepi.py >/home/pi/earthquakepi/earth.log  
2>&1
```

Be sure your Raspi is set to LOCAL time for the cron to work as expected:

```
$ sudo raspi-config
```

Select Internationalization Options -> Change Timezone

**NOTE:** the LCD time display for earthquakes is always in **UTC** only, unaffected by local Timezone.

Finally, set up your EarthquakePi on your desk and reboot it. An initial display showing the IP address for the Pi will be shown for a few seconds.

Every 15 minutes you should see the LCD display (but no sounds) even if there are no earthquakes. The display will be blank otherwise.

When a quake above your preset minimum magnitude occurs, the box will rattle, the LED will flash, the LCD display will show the details and the earthquake sound effect will occur. (Assuming you added all of the various options!)

## Troubleshooting

Obviously, if all the options are active, there are potentials for wiring errors or missing software packages. Use the DEBUG = 1 option and manually run earthquake.py to display any error messages.

```
$ sudo python earthquake.py
```

In normal operation, if you have LOG = 1 (default), you can look in the log file for the last information.

```
/home/pi/earthquakepi/earth.log
```

Typically, issues would be missing software packages ( \$ sudo apt-get install {package}) or a wiring problem, particularly if you had to substitute different components from the above. You may need to cut/paste the error message into Google search to find assistance!

If you have intermittent crashes/reboots/hangs, most likely an insufficient power supply. You MUST have a high quality 5 volt power supply. You CANNOT run this from a PC USB port. A separate supply is needed with at least 1.5 amp or greater. I added 220uf 12v and 22uf 12v capacitors across the incoming power just to help filter out any noise generated by the motor and LEDs.

**NOTE! All the options combined use close to the limit of available current from the Raspberry so if you use different hardware, be mindful to keep total current under 500ma from the Raspi 5V lines.**

SHAKE & RATTLE!