

## RocketLaunchPi – Next Launch from Planet Earth

Version 2016-07-26 r.grokkett



### Overview

RocketLaunchPi displays the next space launch from planet Earth on an LCD screen using your Raspberry Pi (including a Pi Zero) including the vehicle name, location, status, and date/time of the launch in local time. If the launch is imminent, it even plays a count down and blast off sound effects.

**IMPORTANT NOTE:** *If you have already built an Earthquake Pi (See: <http://www.instructables.com/id/Earthquake-Pi-Shake-Rattle-Your-Desk/>) then you have all the hardware necessary for adding RocketLaunchPi. Just skip directly to the "SOFTWARE INSTALLATION" section, below.*

This project retrieves Launch data from the excellent site [www.launchlibrary.net](http://www.launchlibrary.net) via its JSON API interface. It then displays information for the next launch on the LCD screen. Audio effects are played if the launch is imminent.

### Hardware Installation

You will need a Raspberry Pi running Raspian Jessie or better, with Internet connection (via Ethernet or WiFi) along with the parts below. You should have SSH command line access to the Pi.

This project displays information on a 20 char x 4 line LCD screen. No video monitor or keyboard is needed. For audio, you can optionally add an amplifier and speaker.

## 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 box big enough for the Pi, audio speaker and LCD (such as from craft stores like Michaels)
- 5v power supply for Raspi.

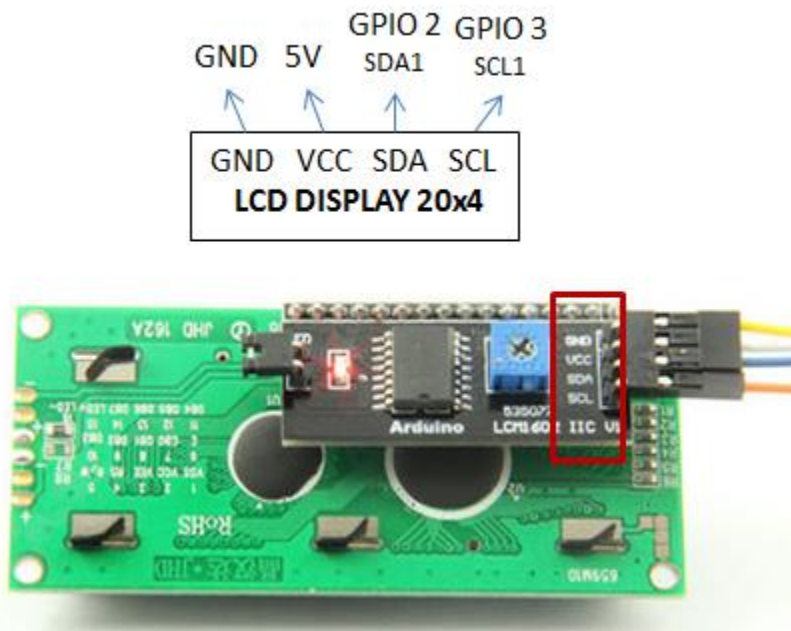
## Options:

### Audio:

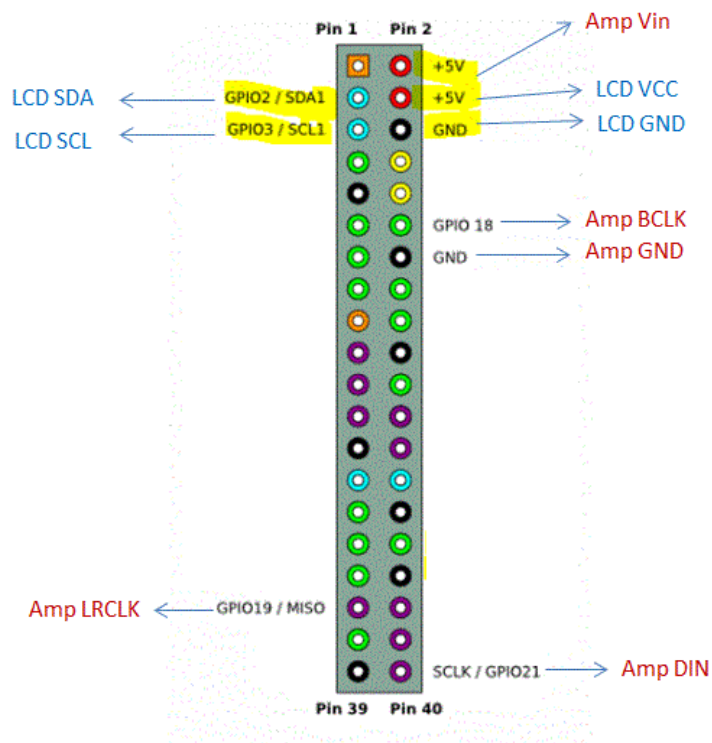
- External speaker (and amp) for Raspi audio.  
For Raspi Zero, add Adafruit I2S 3W Class D Amplifier  
<https://www.adafruit.com/products/3006> and speaker

## LCD 20x4 Display Installation

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



**Fig. – LCD Schematic & Photo**



**Fig.3 - Showing pinouts with all options (Audio Amp and LCD)**

## LCD Software

Install packages:

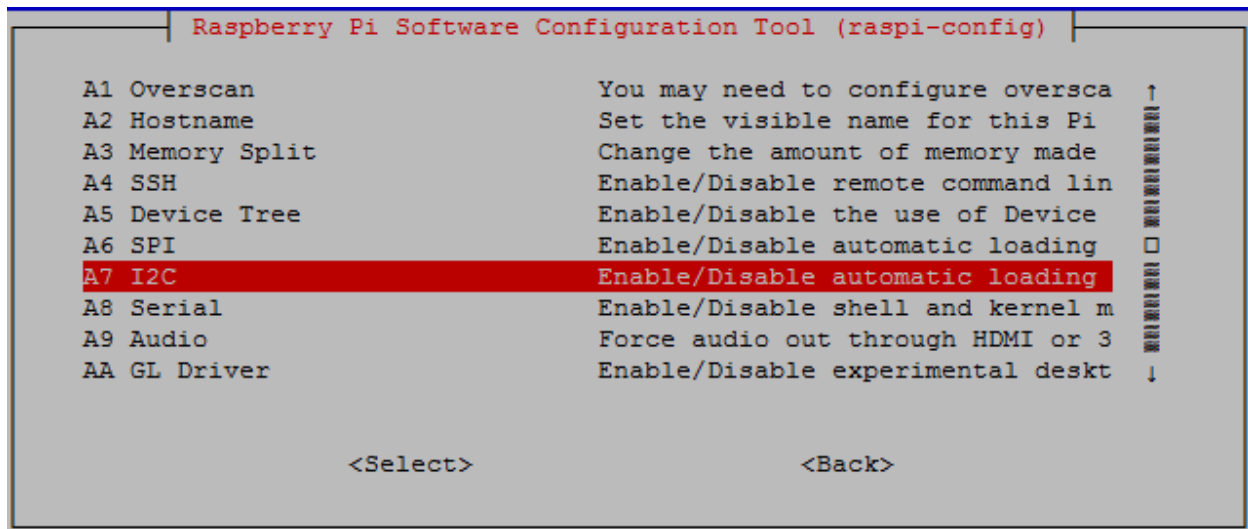
```
$ cd /home/pi
$ sudo apt-get update
$ sudo apt-get install build-essential git
$ sudo apt-get install python-dev python-smbus python-pip
$ sudo pip install RPi.GPIO
$ sudo apt-get install i2c-tools
```

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.



**Fig. – Raspi-Config**

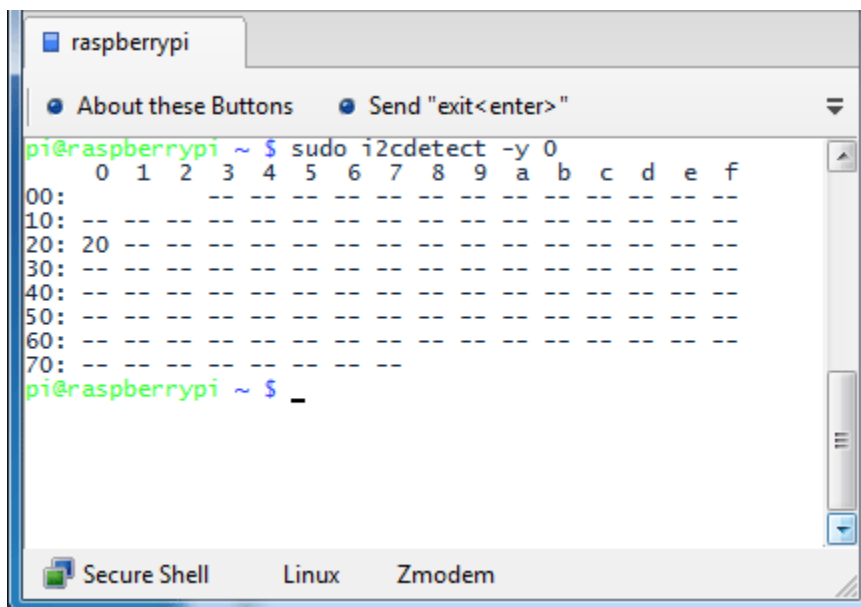
Select FINISH to Reboot your Pi, if asked, or execute:

```
$ 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.



**Fig. Raspi-I2C**

You will test the LCD later...

### Optional Audio Launch 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.

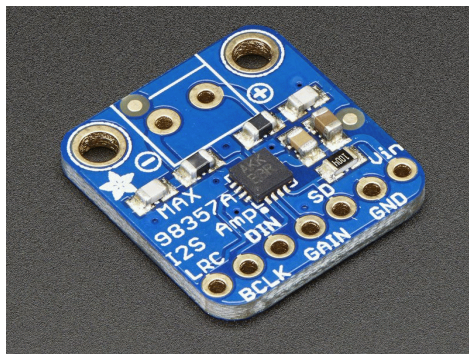


Fig.8 – Adafruit I2S Amplifier

## Software Installation

Available from GITHUB: <https://github.com/rgrokkett/RocketLaunchPi>

Install and test the rocketlaunch program:

1. `$ cd /home/pi`
2. `$ git clone https://github.com/rgrokkett/RocketLaunchPi.git`

**EARTHQUAKE PI USERS:** *If you already have EarthquakePi working, then all you need is the RocketLaunchPi python program. You can skip the LCD & Audio test below.*

## LCD Software & Testing

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

1. `$ cd ~/RocketLaunchPi`
2. `$ nano RPi_I2C_driver.py`  
Edit the LCD address to match what you found above.  
# LCD Address  
ADDRESS = 0x27 (or 0x20 for Adafruit version)
3. `$ python lcd.py`

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



Fig.14 – LCD Test Display

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/RocketLaunchPi/NASACountdown.wav
```

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

Now that the LCD and Audio are working, ready to try the rocketlaunchpi.py program:

**EARTHQUAKE PI USERS:** Can continue [HERE](#)

## Software Test rocketlaunchpi.py

Type:

```
$ sudo python /home/pi/RocketLaunchPi/rocketlaunchpi.py
```

The LCD display should blink several times and then show the information about the next launch.

## Operating RocketLaunchPi

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

```
$ cd /home/pi/RocketLaunchPi  
  
$ crontab -l > tmp  
$ cat pi.cron >> tmp  
$ crontab tmp
```

Example cron:

```
# Display message during bootup  
@reboot sudo python /home/pi/RocketLaunchPi/startup.py >/dev/null 2>&1
```

```
# RocketLaunchPi
5,20,35,50 08-22 * * * sudo python /home/pi/RocketLaunchPi/rocketlaunchpi.py
>/home/pi/RocketLaunchPi/rocket.log 2>&1
```

*Note that the @reboot line isn't needed if you are using RocketLaunchPi with EarthQuakePi. It can be left off or just ignored.*

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

```
$ sudo raspi-config
```

Select **Internationalization Options -> Change Timezone**

Finally, set up your RocketLaunchPi 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. The display will be blank otherwise.

NOTE: The launch countdown sound will only occur if it is within 15-30 minutes of a launch.

## Troubleshooting

Wiring errors or missing software packages are the most likely failure points.

Use the DEBUG = 1 option and manually run earthquake.py to display any error messages.

```
$ nano rocketlaunchpi.py
Set  DEBUG = 1
```

```
$ sudo python rocketlaunchpi.py
```

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

```
/home/pi/RocketLaunchPi/rocket.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 is 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.

5...4...3...2...1!