# FreqGen - Portable Raspberry Pi RF Signal Generator

v.2019-01-14 – rgrokett WA4EFH


## Overview

Based on F5OEO rpitx https://github.com/F5OEO/rpitx

F50E0's impressive package, rpitx, converts a Raspberry Pi into a radio frequency transmitter all done in software.

I wanted to package the transmitter into a portable package to be used as a signal generator, without requiring a HDMI monitor and keyboard (or remote access via a laptop).

This project adds a small OLED screen, battery power and case to a $5 Raspberry Pi Zero to turn it into a **portable RF signal generator** for use with spectrum analyzers such as the Raspberry Pi Freq Show https://github.com/rgrokett/FreqShow or with RTL-SDR receivers or other radio receivers anywhere in the range of AM, FM, Shortwave or VHF/UHF radios.

THIS PROJECT IS **NOT NEEDED** if you just want to run F5OEO's rpitx software package.

Be sure to note F5OEO's **warning** about transmitting using a larger antenna or amplifier. The rpitx software generates a square wave signal with strong harmonics unless you add specific filters for the frequency band desired. But with a small antenna a foot or less long, the 10mw or so signal will be relatively short ranged under normal conditions. YOU MUST HAVE A FCC OR SIMILAR LICENSE to operate this at higher power, better antenna, or outside of the unlicensed radio bands (434mhz).

This version does not implement the record/play capability of the rpitx. (Future enhancement!)

It is suggested that you try out the Pi by installing the rpiTX software before attempting to build this system just to familiarize yourself with its operation. Instructions are on the F5OEO Github (Url above).

NOTE that if purchasing all of the other components as shown below, the costs are about $50 or more, though you will end up with many extra parts for future projects.

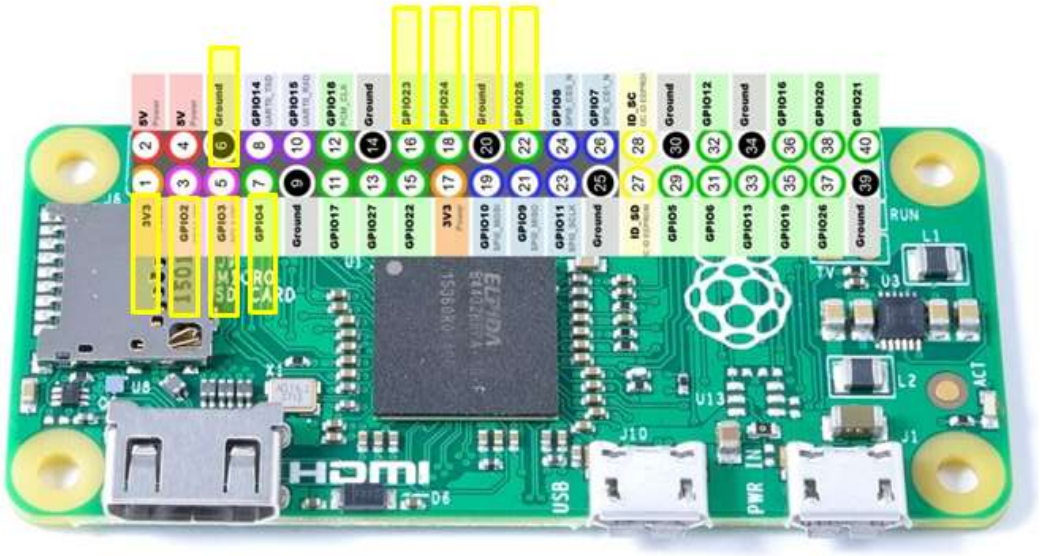
The Pi Zero is sufficient to run this project, but most Raspberry Pi's can be used if you prefer. You will need a larger power supply/battery if running the 3+ or similar Pi.

WiFi is NOT needed for this project once completed, but you will have to temporarily connect your Zero to either WiFi, an Ethernet wired adapter, or a keyboard & HDMI monitor to setup and configure it. Once assembled and working, you will no longer need the keyboard, monitor or Ethernet connection. Only uses the mini OLED screen and push buttons to control.

## Parts List

- Raspberry Pi Zero or Zero W
- 8GB or larger micro SD card
- I2C Serial 128x64 SH1106 OLED LCD
  https://www.amazon.com/dp/B07BHHV844/ref=cm_sw_em_r_mt_dp_U_N9kpCbV14V130
- 3 – micro momentary push buttons
  https://www.amazon.com/dp/B01GN79QF8/ref=cm_sw_em_r_mt_dp_U_61ltCbHBHTSKH
- LiPo charger & 5V 1A Boost power module
  https://www.amazon.com/dp/B07D3SQYKJ/ref=cm_sw_em_r_mt_dp_U_CYltCb2R2H0BB
- 3.7v LiPo 2000mAh rechargeable battery pack
  https://www.amazon.com/dp/B07CZFMFB3/ref=cm_sw_em_r_mt_dp_U_rOltCbAWCAVS5
- BNC Female RF Jack
  https://www.amazon.com/dp/B01L6GTL10/ref=cm_sw_em_r_mt_dp_U_E6ltCbAAZJXNY
- Solder Perf board approx. 2"x2.5"
  https://www.amazon.com/dp/B01M7R5YIB/ref=cm_sw_em_r_mt_dp_U_o-ltCbMP8ZQDA
- Plastic box  or 3D printed case
- Wire, soldering equip, etc.

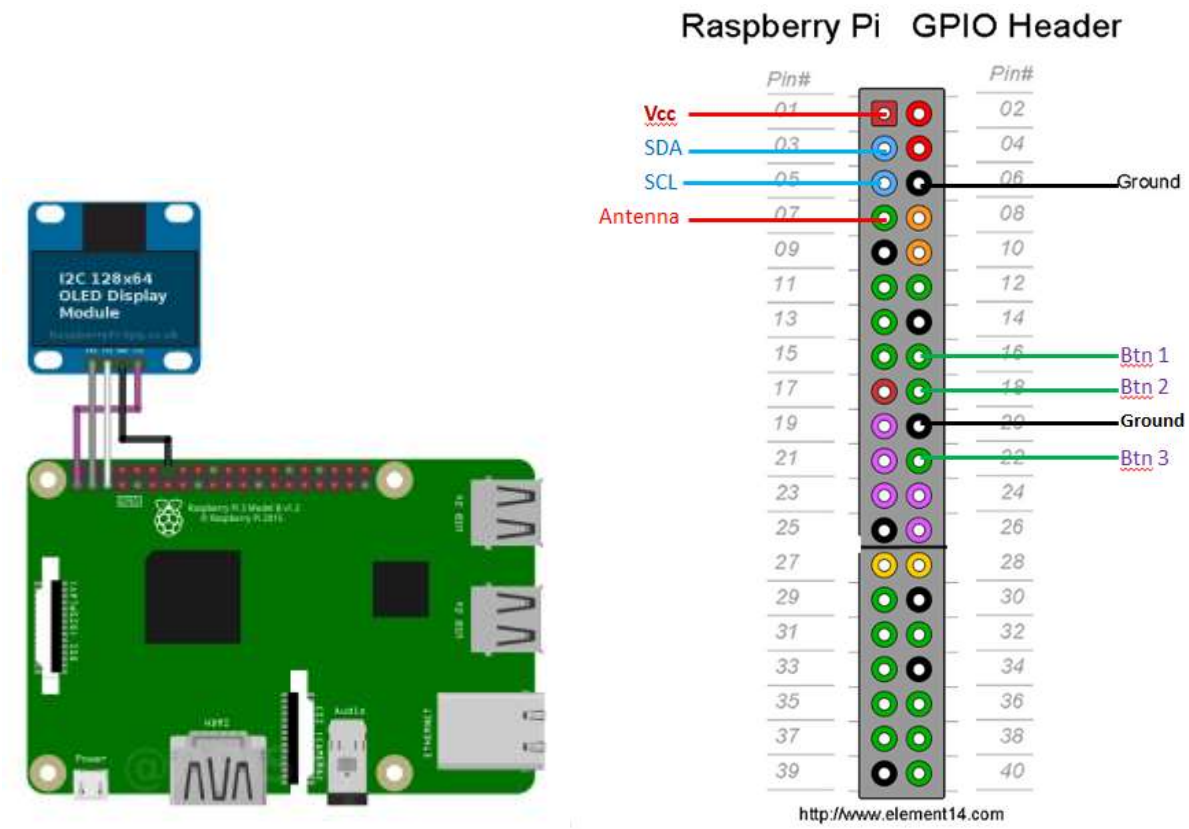
It is suggested that you use a breadboard to wire everything and pre-test it to verify the components and the wiring.




## OLED Screen Wiring

The OLED Screen requires only 4 wires to connect it to the Raspberry Pi. The connections are the same for all Raspberries.


| OLED Pin | Pi Header Pin | Notes |
|---|---|---|
| Vcc | 1 | 3.3V |
| Gnd | 14 | Ground |
| SCL | 5 | I2C SCL |
| SDA | 3 | I2C SCA |



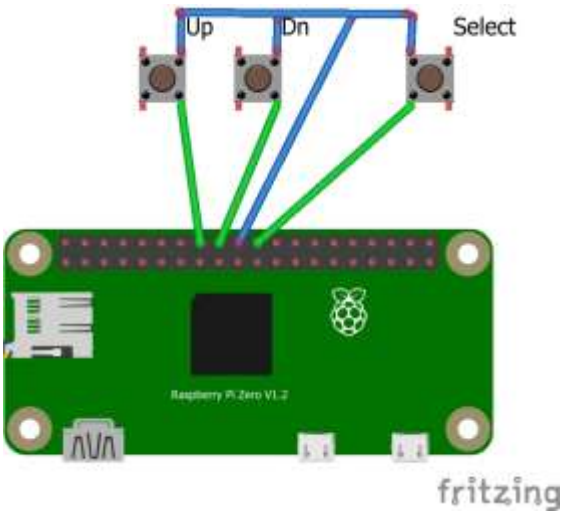

For more information, check out these sources:

- https://www.raspberrypi-spy.co.uk/2018/04/i2c-oled-display-module-with-raspberry-pi/
- https://github.com/rm-hull/luma.oled
- https://luma-oled.readthedocs.io/en/latest/intro.html#

## GPIO Button Wiring

There are three momentary push buttons needed to control the FreqGen OLED screen. Each is connected from a GPIO pin to Ground. When pushed, this will bring the GPIO pin to ground, which the Raspberry software will detect.

| Button | Pi Header Pin | GPIO |
|---|---|---|
| 1 | 16 | 23 |
| 2 | 18 | 24 |
| 3 | 22 | 25 |
| Gnd | 20 | Ground |




For more information on using Buttons on a Raspberry:

https://www.makeuseof.com/tag/add-button-raspberry-pi-project/

## Antenna Wiring

The Antenna should be connected to Pin 7 (shown in diagram above), also known as GPIO 4. The antenna should be a wire or antenna LESS THAN 1 foot (20cm) long. Just a few inches is sufficient. I added a RF connector to allow test leads or a very small antenna to be attached.

DO NOT connect a longer wire or outdoor antenna without a bandpass filter for the frequency to be used AND without a proper Amateur Radio License.

In my tests using a just a 6 inch antenna, I was able to receive the NFM signal on the 2meter ham band over a block away using just a handheld receiver. This is sufficient to cause radio interference with legitimate communications, if misused. And is illegal!

## Software Installation

1. If you haven't already, install Raspbian Lite version onto a 8GB or larger microSD card. You DO NOT need the GUI version, as this project does not use a monitor or keyboard.
   https://www.raspberrypi.org/downloads/raspbian/

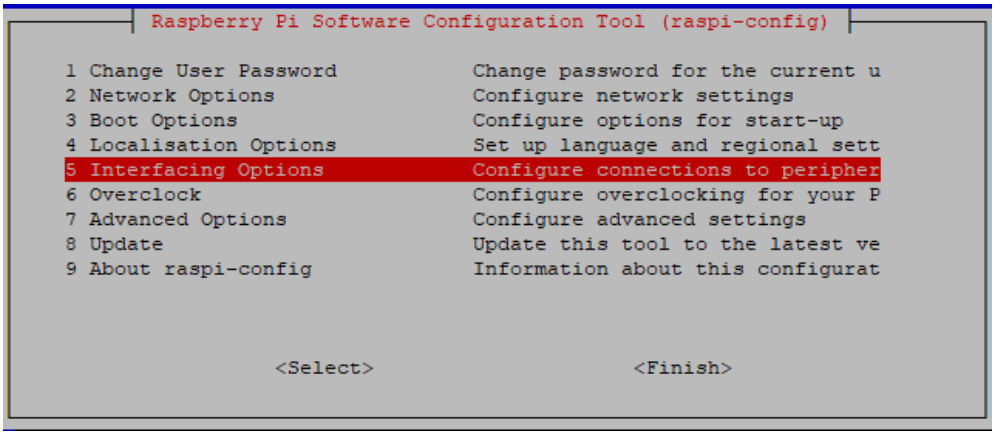
2. Plug in a keyboard & HDMI monitor temporarily.
   Alternately, use a USB Ethernet Adapter or WiFi card (or built-in WiFi, if available).
   This is only needed long enough to install the software and check things out.

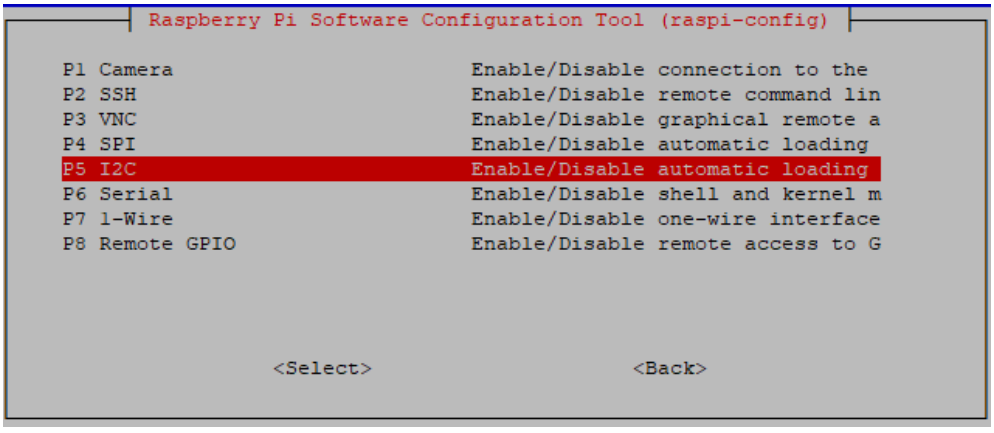
3. Insert the microSD card into the Pi and plug in the power now. It will take a few moments to boot. All commands will be given via the Linux command line. No GUI needed.

4. Log into your Pi. Default id/password is “pi/raspberry”
   Once logged in type:

**5.** Update your Raspbian:
   **`$ sudo apt update`**
   **`$ sudo apt upgrade`**

6. Configure the Raspberry, you need to change the password and turn on the I2C Interface.
   **`$ sudo raspi-config`**
   a. Change User Password
   b. Interfacing Options -> I2C -> Yes
      (Enable I2C interface. *See below screens*)




   c. Once enabled, TAB to “Finish” and answer “Yes” to reboot.

7. After rebooting, log back in and verify the I2C interface is working and can see the OLED screen:
   Install the i2c utilities and then try a test.
   **$ sudo apt install -y i2c-tools**
   **$ sudo i2cdetect -y 1**

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

8. You should see a “3C” listed, which is the default address of your OLED screen.
   If so, you are good to continue. Otherwise verify your wiring. You will test the actual display once the software drivers are installed, below.

## Install the OLED Driver Software:

```
sudo apt install git pip
sudo apt install –y python-dev python-pip
sudo apt install –y libfreetype6-dev libjpeg-dev build-essential
sudo apt install –y python-gpiozero
sudo apt install -y python-imaging python-smbus
sudo -H pip install --upgrade luma.oled
```

## Install the RpiTX Software:

```
cd /home/pi
git clone https://github.com/F5OEO/rpitx
cd rpitx
./install.sh
```

Takes 5-10minutes. Then answer “Y” to the question that should come up, below:

```
In order to run properly, rpitx need to modify /boot/config.txt. Are you sure
(y/n) y
Set GPU to 250Mhz in order to be stable gpu_freq=250
```

Once finished, reboot your Pi:

```
sudo reboot
```

*For more information, see:*

F5OEO rpitx https://github.com/F5OEO/rpitx

## Install FreqGen Software:

```
cd /home/pi
git clone https://github.com/rgrokett/FreqGen

cd FreqGen
```

Now you can test the OLED screen and the Buttons:

```
$ python testOLED.py
```

You should see a simple “Hello World” test appear.

```
$ python testButtons.py
```

Press each button to be sure it can be detected. A message will show which one is pressed.

Last, lets run the FreqGen software from the command line just to verify all the software components are installed properly:

```
$ python freqgen.py
```

After a few moments, (The zero is a bit slow), the FreqGen splash screen should appear and then the Main menu. You can exit the program using the buttons to navigate to the Quit option.

If all tests pass, then you are ready to do the final quick install which will make the Pi boot directly into the FreqGen software each time it is powered up:

```
./install.sh
```

Now Shut down the Pi and unplug the power and remove the keyboard & monitor and/or the Ethernet adapter. They are no longer needed.

## Operation

To use the RF generator, plug in a short antenna of any type (less than about 30cm or 1ft in length). DO NOT EXCEED this length. Even as is, the Pi can be detected from nearly a block away (in clear area). Power the Raspberry Pi up by plugging in the 5v power. After about a minute, the software will boot automatically and the OLED splash screen and menu will appear.

Use the momentary push buttons to navigate the menu:


Up/Down and Select button

**FREQ**: On the Frequency menu, use the Left/Right buttons to increment/decrement the digits. Press Enter button to move to the Next digit. Press Enter twice to end entry and accept that frequency, returning to the Main Menu.

You can enter a range of approx. 0.005 mhz to 1500.0 mhz. For the decimal point, you must click the Left/Up button pass the "9" digit until you see a period ".".

Note that the exact frequency of the RF signal can vary slightly with temperature and variations in the Pi, but are typically quite close even on the Zero.

**MODE**: On the Mode menu, you can select one of the several transmission modes available to the rpiTX software. Not all modes are available, as some are not useful for RF signal generation. If you use a 2m FM receiver, select the NFM (Narrow FM) mode for audio.

**STAT**: Selecting the Status option will turn the transmitter on/off (Idle/Active).

Use a radio receiver or RTL-SDR receiver or similar tuned to the selected frequency.

**QUIT**: Gracefully shutdown the Pi OS. After a few seconds, you should see the Pi's green power LED go OFF. Then it is safe to remove 5v power.

***Be SURE to check out the F5OEO rpitx documentation showing samples of the different Modes that are available. The doc also provides in-depth details of the operation*:**
**https://github.com/F5OEO/rpitx**