Raspberry Pi Internet Radio

A Beginners Guide

A simple guide to making an Internet radio with the Raspberry Pi

Bob Rathbone Computer Consultancy

www.bobrathbone.com

27th of February 2022

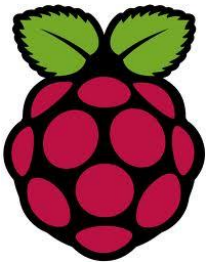
Radio Version 7.4

Contents

Figures

Tables

Introduction



This guide describes how to create one of the most popular Internet Radios using the Raspberry PI educational computer (RPi). This guide provides the very basics of how to construct a simple Internet radio and install the radio software.

The source and basic construction details are available from the following web site: https://bobrathbone.com/raspberrypi/pi_internet_radio.html

The following sections describe how to construct a simple Raspberry Pi Internet radio with an LCD display and a volume and tuning knob.

The information in this guide has been extracted from the **Raspberry Pi Internet Radio Constructors Manual** which is available on the above web site or from the **docs** directory at https://github.com/bobrathbone/piradio6.

The full constructors guide is very comprehensive document with many different designs and extra features such as support for an IR remote control, Spotify etc. It is over 370 pages long and may be quite overwhelming for a beginner at first. For this reason, a single design has been extracted from the constructor's manual to provide this simple guide for beginners.

Radio overview

This project is for the construction of a simple Internet radio with a 2 or 4-line LCD display connected with an Arduino PCF8574 I2C backpack and two rotary encoders for volume control and channel/track change. The software is provided by the **radiod** (Radio Daemon) program available as a Debian package from the Bob Rathbone web site. The **radiod** software interfaces to the Music Player Daemon (MPD) from https://www.musicpd.org.

Skills required

The project is designed such that anyone from young to old can create their own radio. However, some soldering skills are required which limits the project to older children. It is necessary to solder a 16-pin male header to the LCD display card and also to add one wire to two pins on the display card. All other connections are push fit using jumper/ribbon cables.

If you have limited or no soldering skills then perhaps ask for help from someone who has these skills. If this is not possible, all is not lost, as the main **Raspberry Pi Internet radio constructors manual** on the Bob Rathbone web site shows several other solutions which do not require any soldering skills at all.

Front photograph

The photograph on the front cover is of a Raspberry Pi Internet Radio built thanks to Joaquin Perez. He is using a pair of PC speakers to amplify the audio output from

Hardware overview

The principal hardware required to build the radio consists of the following components:

- A Raspberry Pi model 3B (other models can also be used)
- A 16GByte micro-SD card with Raspberry Pi OS (Bullseye)
- An Official micro-USB 5-Volt 2.5A power adapter
- An LCD HD44780U 2x16-character or 4x20-character display
- An Adafruit PCF8574 I2C backpack
- Two KY040 rotary encoders
- Female to female ribbon cable

Raspberry Pi computer

The **Raspberry Pi** is a credit-card-sized single-board computer developed in the United Kingdom by the Raspberry Pi Foundation

Figure 1 Raspberry PI Model 3B Computer

The HD44780U LCD display

Figure 2 The HD44780U LCD display

The HD44780U LCD interface is an industry standard interface for many LCD displays. These can come in various sizes.

This project can use either a 2x16 or 4x20 character LCD display and is software configurable during installation.

Official RPi USB power supply

Figure 3 Official USB power supply

The USB supply has 100-240v input, 5.1V 2.5A output. There are two types of USB connector:

- USB C for Raspberry Pi model 4B
- USB Micro for Raspberry Pi model 3B/2B

They can be ordered for various mains sockets, for example UK, USA, European or Australasia. Make sure that you order the correct one for your Raspberry Pi.
Note: Telephone chargers are inadequate.

Arduino PCF8574 I2C backpack

This type of backpack is popular with Arduino users. It interfaces the Raspberry Pi using the **I2C** protocol. **I2C** a communication protocol developed by Philips Semiconductors.

In this application, it transfers of data between the Raspberry Pi and the LCD display using just two signal wires. The device address is usually hex 0x27. Another manufacture may use hex 0x37. This is configurable in the radio configuration program.

Figure 4 Arduino PCF8574 I2C backpack

The wiring From top to bottom is:

1. GND (0 volts) – GPIO header pin 6
2. VCC +5 volts – GPIO header pin 2
3. SDA I2C Data – GPIO 2 (pin 3)
4. SCL I2C Clock – GPIO 3 (pin 5)

The blue potentiometer on the right is the contrast adjustment.

SD Card

Figure 5 16 MB micro SD card

The Raspberry Pi boots from a micro-SD card with the Raspberry Pi OS (Formerly Raspbian) burnt onto it. These come with or without a micro-SD adaptor. Various suppliers supply these with pre-loaded with the operating system.

A 16 MB micro-SD card is recommended for this project.

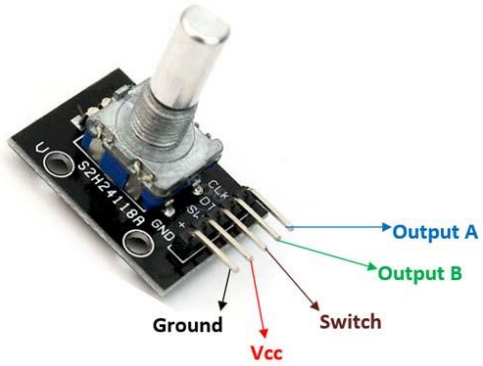


Figure 6 KY-040 Rotary encoders

These cost-effective Rotary Encoders from Handson Technology originally designed for use with Arduino are now being used more and more by constructors. The KY-040 Rotary Encoder specification shows that these are powered by +5V to the VCC pin.

However, the Raspberry Pi uses a +3.3V supply and cannot tolerate +5V on the GPIO's so only connect VCC to +3.3V. These encoders work fine with VCC connected to +3.3V with this project.

Figure 7 Radio knob

You will need two standard radio knobs for the above Rotary Encoders. There is a wide selection of these to be found on the Internet. Ones with a grub screw are better than the push-fit type.

Cables and connectors

Figure 8 Female to female connector cable

The female-to-female ribbon cable is used to connect the LCD display KY040 rotary encoders to the Raspberry Pi 40 pin GPIO header.
The cable can be separated into two smaller five-way ribbon cables for the rotary encoders and a four-way ribbon cable for the I2C backpack and LCD display.

This type of cable is available from various Raspberry Pi and Arduino suppliers.

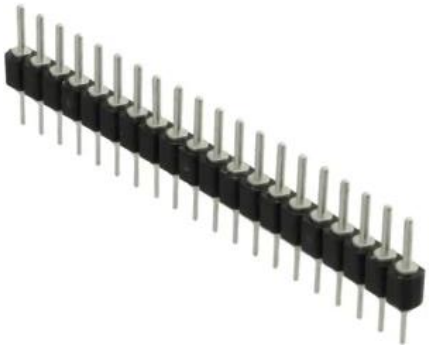


Figure 9 Single Inline Connectors (SIL)

This Single Inline Male Connector (SIL) is needs to be soldered onto the back of the LCD board. This will normally have 16 or 18 holes to take the connector. Only the first sixteen are used and then only eight of these are actually connected.

Buy the 'snap-off' type so that the correct length is easily attained.

Figure 10 Female SIL connector

The Single Inline Female Connector (SIL) is soldered onto the back of Arduino backpack so that it connects to LCD.

Again, buy the 'snap-off' type so that the correct length is easily attained.

Figure 11 PC loud-speakers

Any inexpensive PC speakers with either standard 3.5mm jack plug or USB input.

Figure 12 ABS enclosure

A suitable case to house the radio is required. Typically, an ABS enclosure is very suitable solution. These come in various sizes and are made of plastic and are easy to work.

However, any other solution may be used but make sure that the enclosure has adequate ventilation holes.

Wiring

Raspberry Pi 40-pin GPIO header

The following shows the pin outs for the GPIO for models 2B, 3B and 4B
See: http://elinux.org/RPi_Low-level_peripherals. For more details.

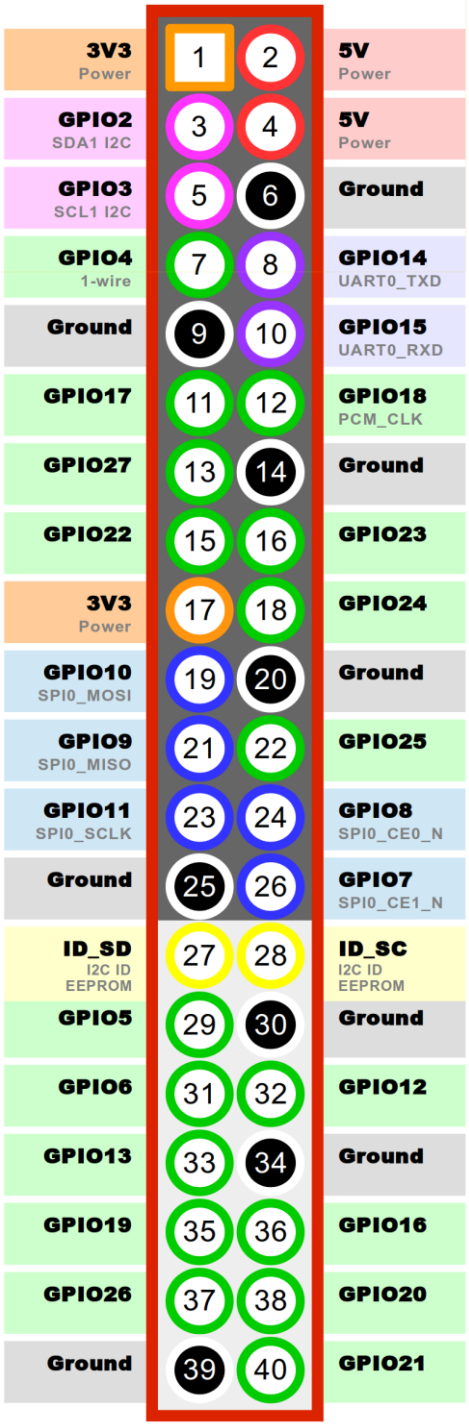


Figure 13 GPIO Numbers

The above diagram shows the GPIO 40 pin header viewed from above.

The following table shows how to wire up the rotary encoders.

Table 1 Radio Rotary Encoder Wiring

GPIO Pin	Description	Radio Function	Volume Rotary Encoder	Channel Rotary Encoder
1	3V3	+3.3V supply	+ (VCC 3.3V)	
6	GND	Zero volts	Common (GND)	
7	GPIO 4	Mute volume	Knob Switch (SW)	
8	GPIO 14	Volume up/down	Output A (CLK)	
10	GPIO 15	Volume up/down	Output B (DT)	
17	3V3	+3.3V supply		+ (VCC 3.3V)
9	GND	Zero volts		Common (GND)
11	GPIO 17	Menu switch		Knob Switch (SW)
16	GPIO 23	Channel up/down		Output A (CLK)
18	GPIO 24	Channel up/down		Output B (DT)

Table 2 I2C Backpack connections

Backpack	Label	Description	GPIO	Physical Pin
1	GND	Zero volts	-	14
2	VCC	+5V supply	-	4
3	SDA	I2C Data	GPIO 2	3
4	SCL	I2C Clock	GPIO 3	5

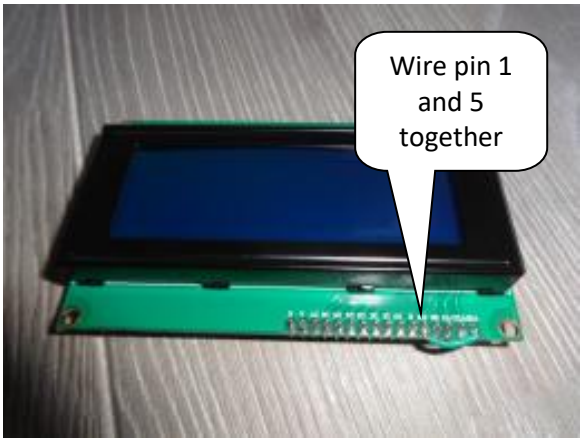


Figure 14 Wire LCD pin 1 (GND) and 5 (RW) together

The Read/Write (RW) pin 5 must be connected to pin 1 (0V). It is very important that this pin is grounded! If pin 5 is not grounded it will damage the Raspberry PI. Always wire LCD pin 5 and 1 directly together. Do not rely on grounding pin 5 with a GND wire on the GPIO connector. If this wire drops off then the LCD data lines will be put into write mode putting +5V on the GPIO pins which will probably cause irreparable damage to the Raspberry Pi.

Figure 15 The I2C backpack mounted on the back of the LCD

The diagram on the left shows the Arduino I2C backpack mounted on the back of a 2x16 LCD using the SIL connectors.
The four pins on the right are for the power supply and I2C connections.

KY040 Rotary Encoder wiring

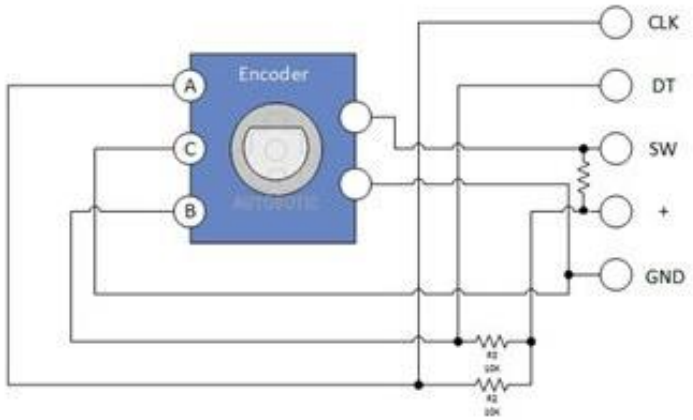


Figure 16 KY-040 Circuit Diagram

The specification shows the rotary encoders are labelled CLK(Clock), DT(Data) and + (VCC) however it is more usual to label these A, B and C. It has its own three 10K pull-up resistors.

Connect **+** to the +3.3V supply.
Do **not** connect to +5V despite the fact that the KY040 specification states this.

Parts list

The following table shows the parts list for a basic Raspberry PI Internet Radio. This list is for the version using the HD44780U LCD connected to the Raspberry Pi via an I2C backpack.

Table 3 Parts list

Qty	Part	Supplier
1	Raspberry Pi 3B Computer	PiHut
1	16 GByte Micro-SD Card	Any PC or Photographic supplier
1	Raspberry Pi Bullseye OS	Raspberry Pi foundation downloads
1	LCD HD44780U 2 x 16 Display *	PiHut
1	Arduino PCF8574 I2C backpack	PiHut
2	KY040 Rotary encoders	PiHut
1	Official Raspberry Pi +5V 2.5A power supply	PiHut
2	Radio Knobs	PiHut
1	Set of PC speakers (Either USB or 3.5mm jack plug	Any PC supplier

Construction

The main construction consists of soldering the SIL male header onto the LED board and wiring up the rotary encoders. There are numerous soldering tutorials on the Internet to help with this task.

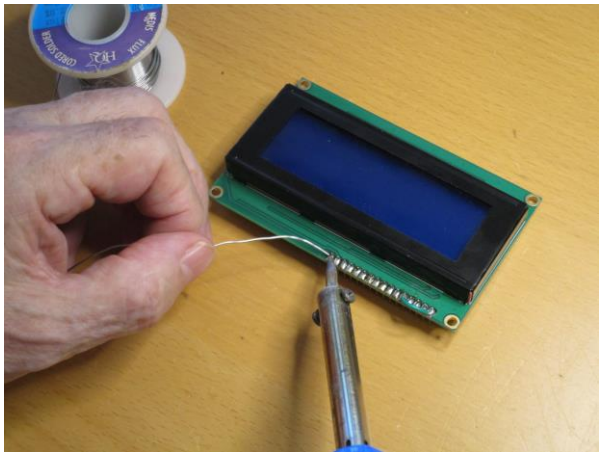


Figure 17 Soldering the LCD display header

The 16/18 pin SIL male header has to be soldered onto the LCD board so that the longer pins protrude out of the back of the LCD card.

Use a fine tipped soldering iron. Press the hot soldering iron onto the base of the pin to be soldered and then apply the solder. Once sufficient solder has melted onto the joint take away the solder wire but leave the soldering iron for two to three seconds to allow it to properly flow around the joint.

Do not apply too much solder to the joint

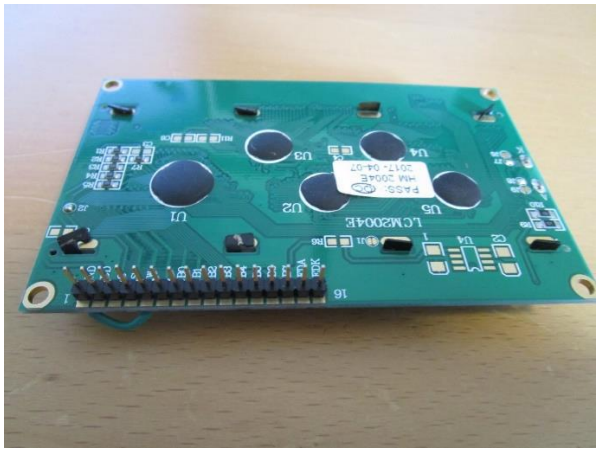


Figure 18 LCD display header rear view

The illustration on the left shows the 16-pin header mounted on the LCD card. Some LCDs have 18 pin headers to support a RGB backlight. Refer to the main Raspberry Pi Internet Radio Constructors manual for more information on these types of display.

Note the very important green jumper wire between pin 1 and 5 to hold the RW (Read/Write) line low. Do not connect the LCD without this wire in place or you will damage the Raspberry Pi.

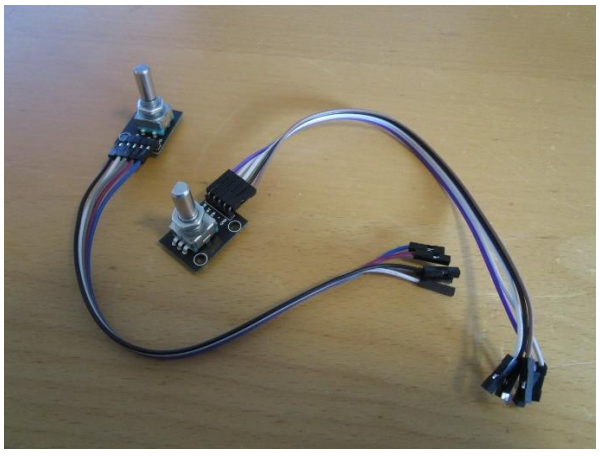


Figure 19 Wiring the rotary encoders

To wire up the KY040 rotary encoders, tear off five wires from the female-to-female ribbon cable.

Then gently push the cable one wire at a time onto the pins on the KY040 rotary encoder. They are quite stiff so take care not to damage the KY040 pins.

System Software Installation

Conventions used in this tutorial

Installation of the radio program requires you to enter lines at the command line prompt. This requires you to log into the Raspberry PI as user '**pi**'. The default password is **raspberry**.

Note: Don't carry out any of the following commands just yet. They are just examples.

```
Raspberrypi login: pi
Password: raspberry
Last login: Tue May  4 11:33:54 2021 from 192.168.1.200
pi@raspberrypi:~$
```

The prompt line is displayed ending with a **$** sign. The **pi@raspberrypi:~** string means user 'pi' on host machine called 'raspberrypi'. The ~ character means the user 'pi' home directory **/home/pi**.

In this tutorial if you are required to do something as user **pi** then only the **$** sign will be shown followed by the command as shown in the example below:

```
$ mpc status
```

Copy and paste commands from this guide without the $ sign.

Some commands produce output which does not need to be shown. In such a case a ':' is used to indicate that some output has been omitted.

```
$ aplay -l
**** List of PLAYBACK Hardware Devices ****
: {Omitted output}
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
```

END OF EXAMPLE COMMANDS.

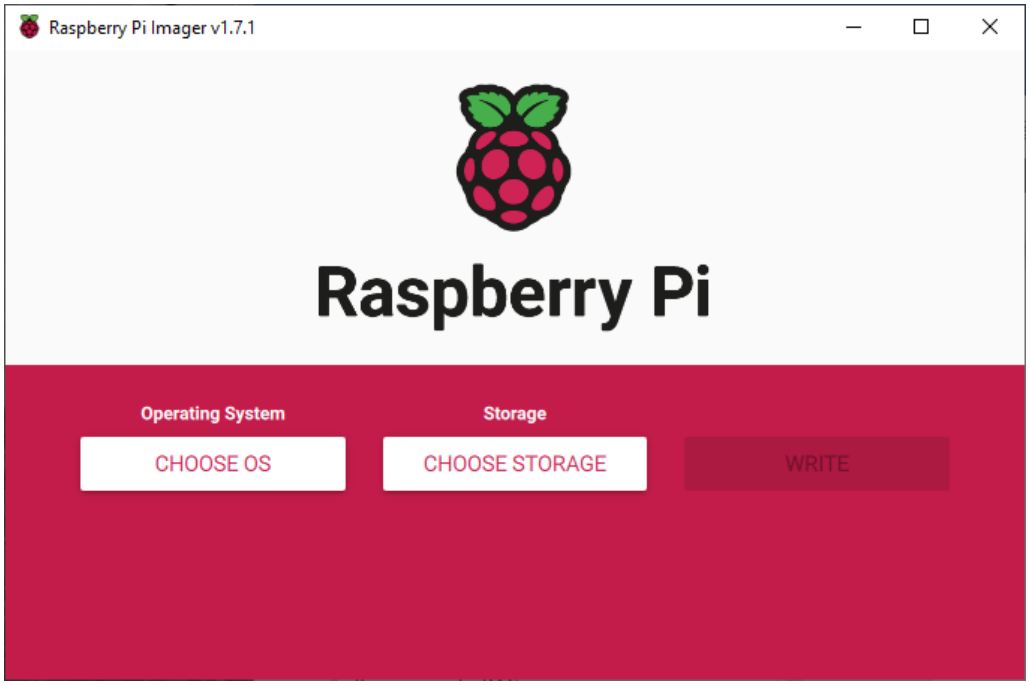
SD card creation using Raspberry Pi Imager

Use at least a 16 Gigabyte Card for **Bullseye Lite** or 32 Gigabyte for **Bullseye Desktop/Full**. Create an SD card running the latest **32-bit** version of **Raspberry Pi Bullseye** or **Bullseye Lite**.

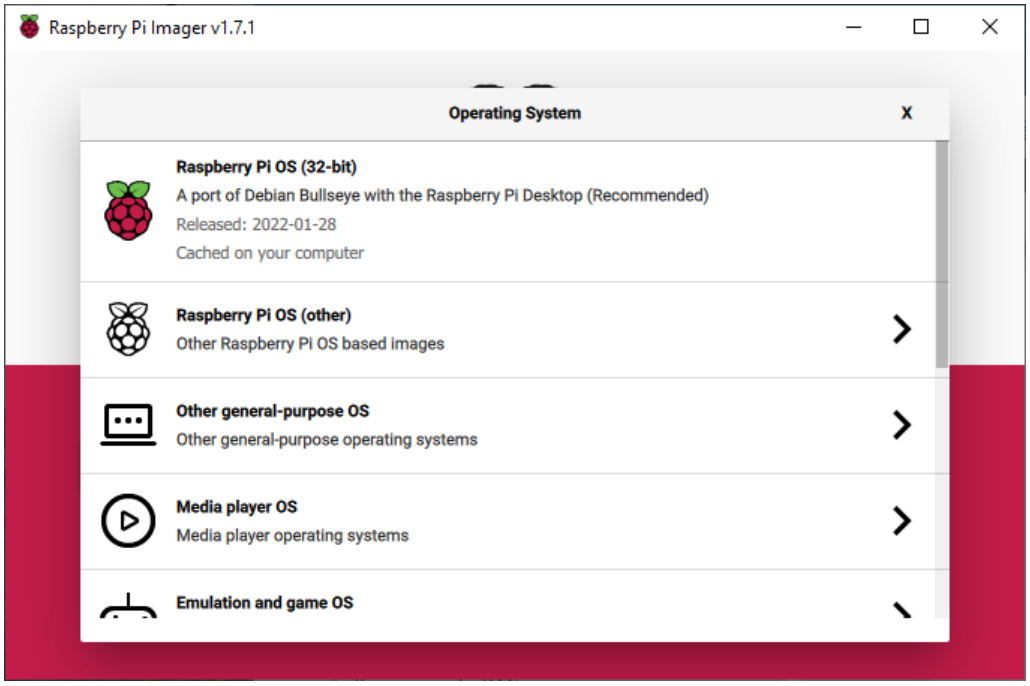
There are a couple of ways of doing this but in this tutorial, we are using the **Raspberry Pi Imager** software to create the SD card.

Using a Web browser, go to https://www.raspberrypi.org/software/
Download and install the **Raspberry Pi Imager** software for your PC Operating System (Normally Windows or Mac OS). Once installed you will see the Raspberry Pi Imager Icon on your desk-top. Click on the desktop Imager icon.

The following screen will be displayed:



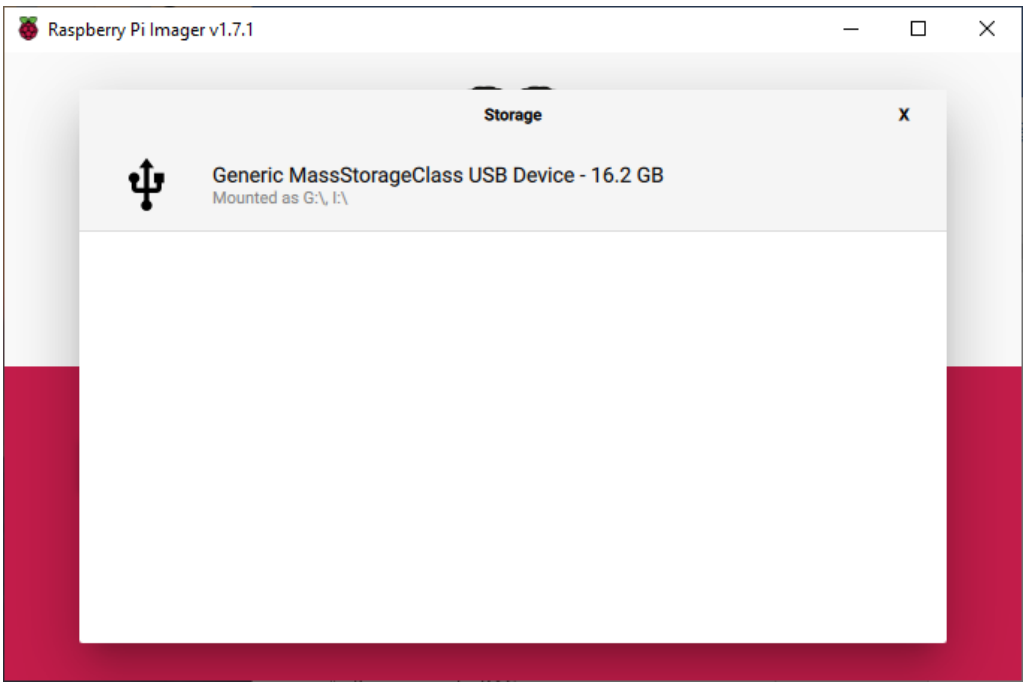
Insert the SD card into your card reader. Ignore any Windows messages to format the card as it isn't necessary. Click on "Choose OS"



The download options will be displayed.

The actual drive letter will have been assigned by your PC. You should see something like the following. In this example it is drive **G:**

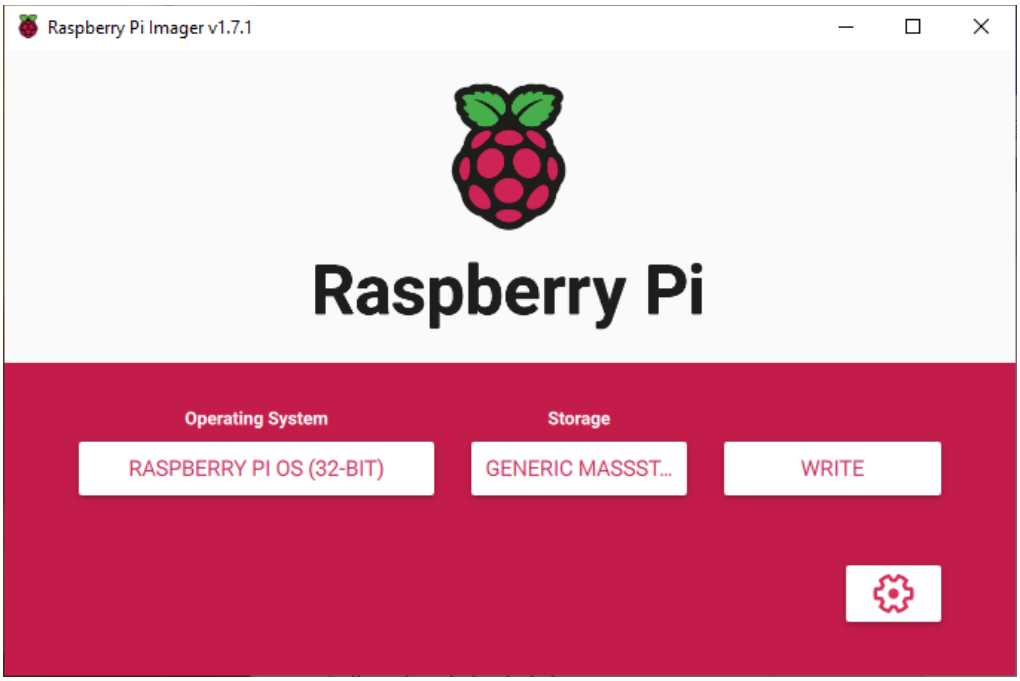
Select your SD card. The imager software should only display USB devices.



Note: If there are multiple choices of USB drives make sure that you select the correct one. Do not proceed until you are absolutely sure you are selecting the correct drive and not for example a USB backup drive.

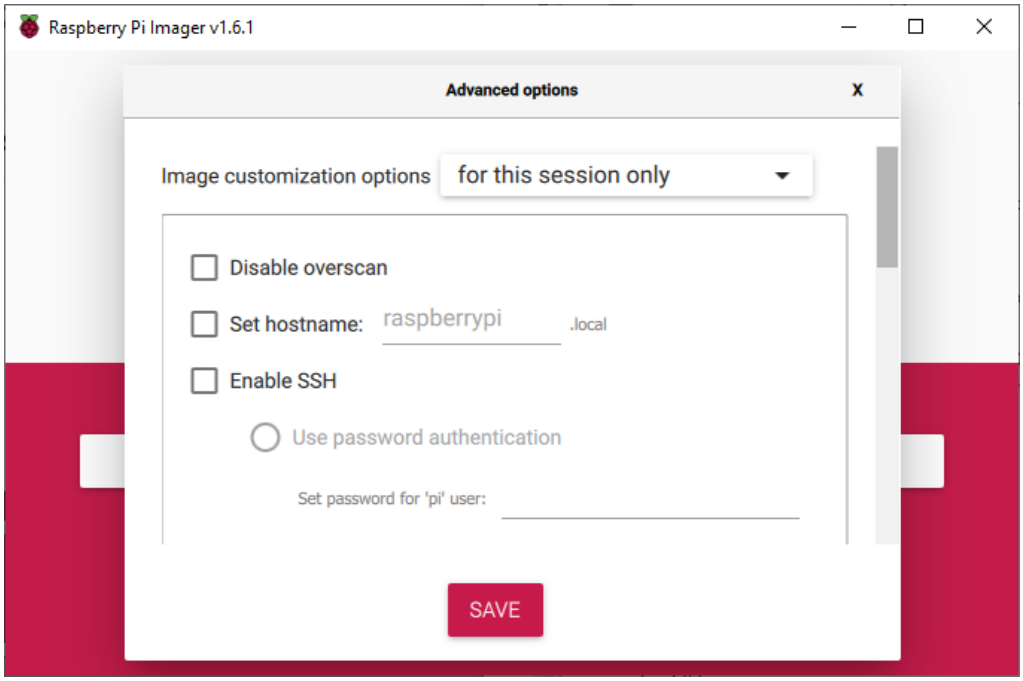
Once both the OS and USB drive have been selected the following screen is displayed.

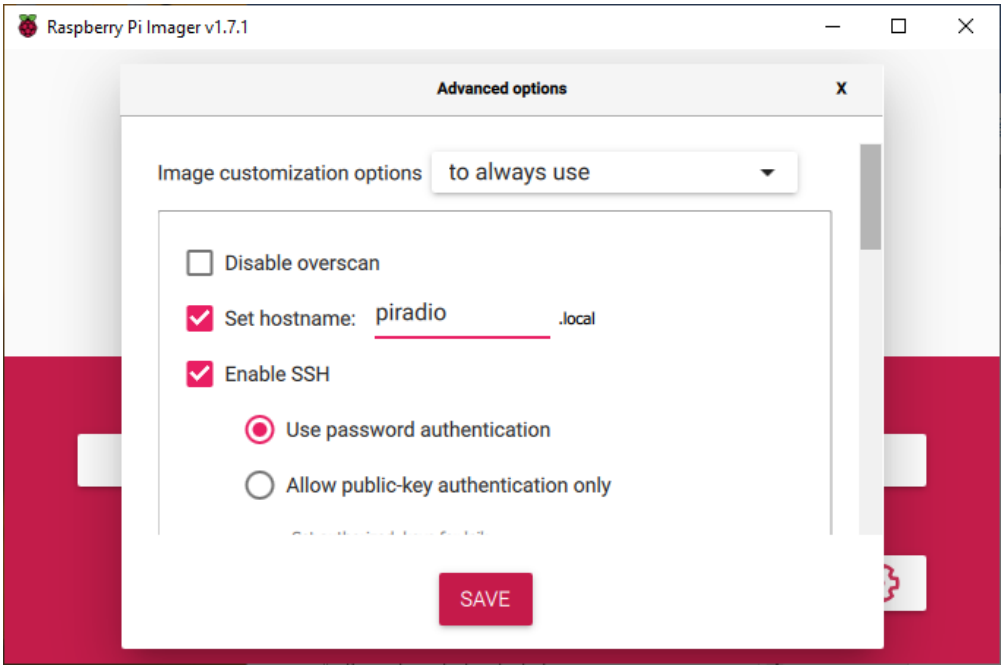
Note: Do **not** press write at this stage as you need to set up SSH (Secure Shell login) and your Wi-Fi identifier (SSID) and password and other optional changes.



To select the customisation menu press either the configuration icon (Gear wheel) in the bottom right or **CTRL**, **Shift** and **X** keys together. Clicking on WRITE will ask if you want to edit the parameters.

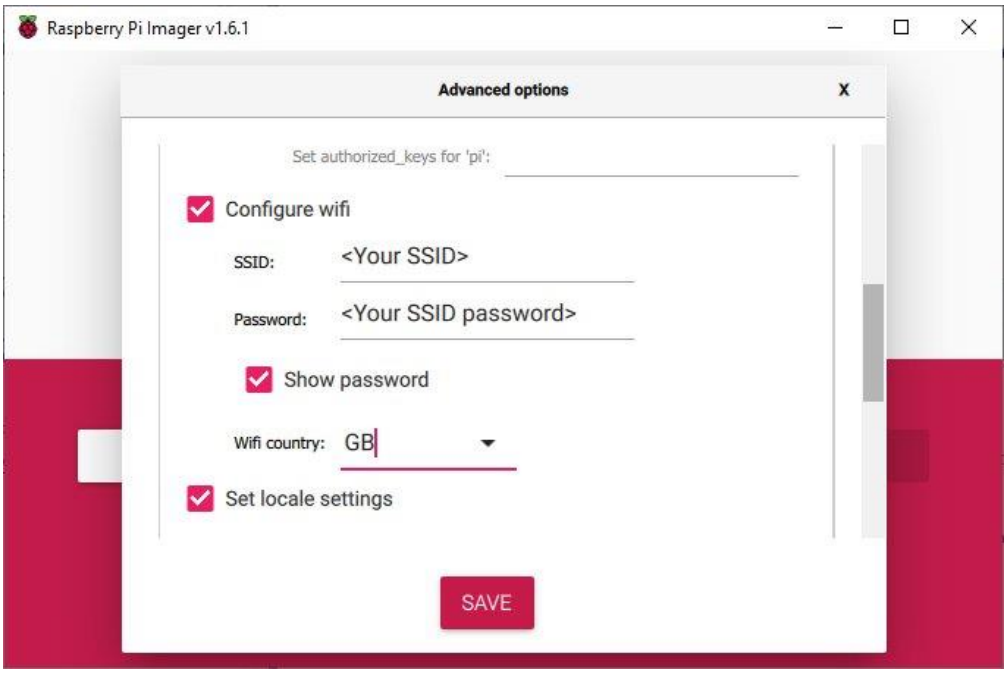
The following Advanced Options screen will be displayed





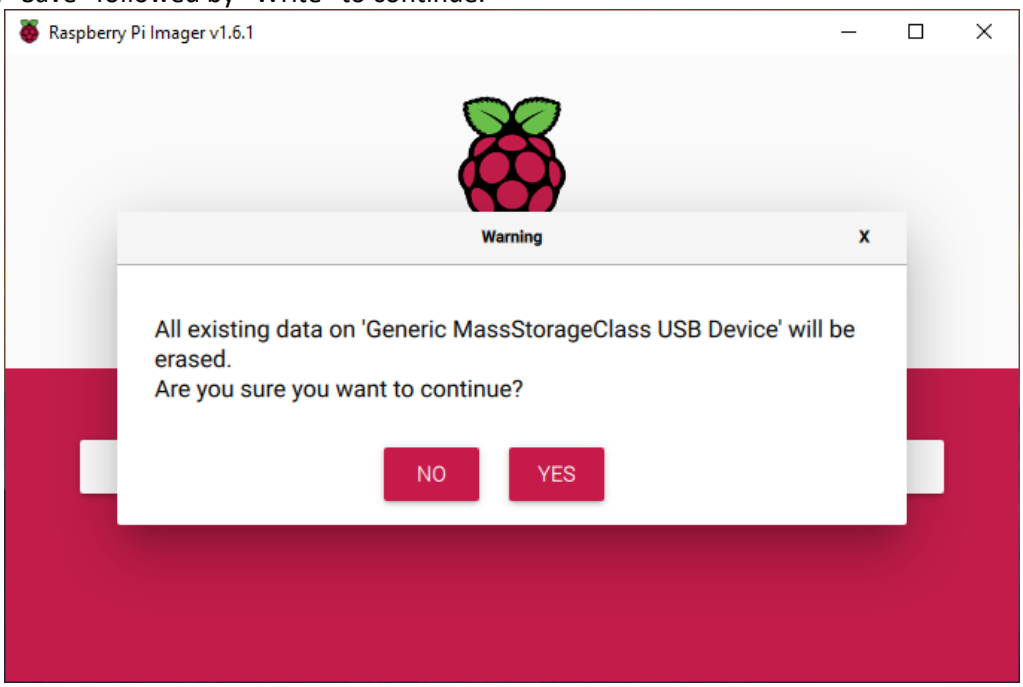
Click the “Set hostname” and “Enable SSH” tick boxes. Enter a hostname (such as **piradio**) and a password for user **pi**. Make a note of the user pi password. Do not press SAVE just yet. Ignore “Disable overscan” for now, as this is used to prevent a black border around a HDMI screen.

Now scroll down to the “Configure wifi” option using the right-hand scroll bar. Enter the SSID and password for your router. This information is usually given on the router itself or with router documentation.



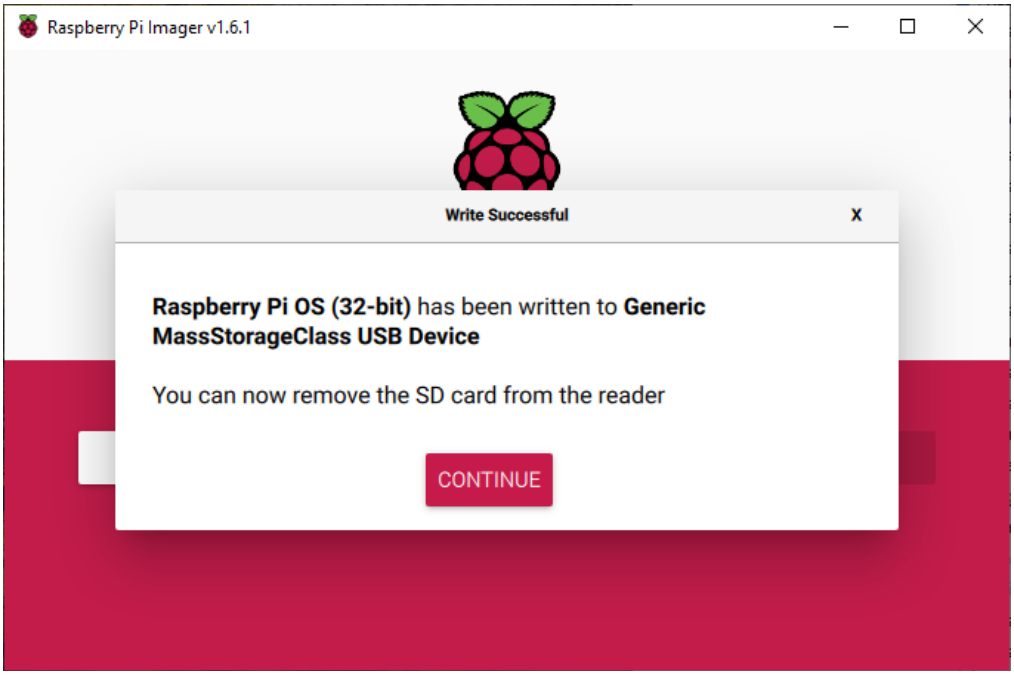
You must also set the locale using the two-letter country code for your location, for example GB (Great Britain), DE (Germany) or NL(Netherlands). It is required for correct set-up of Wi-Fi. There are other options in this menu but these are not required for this project.

Click on “Save” followed by “Write” to continue.



Click “YES” to

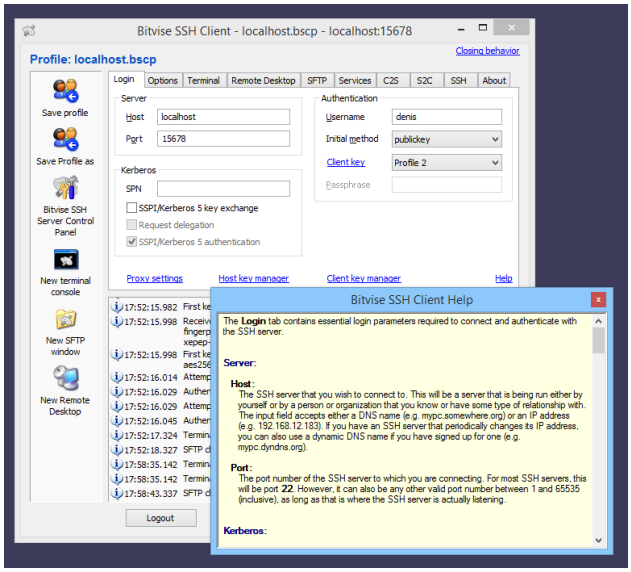
This will take about 10 minutes to complete.



Remove the SD card from your computer and insert it into the Raspberry Pi (Remove power from the Raspberry Pi first).

Power up the Raspberry Pi. The first boot will take a little longer as there are a few jobs that the OS must carry out, such as re-sizing the file system on the SD card.

Log into the RPi using Bitvise or Putty



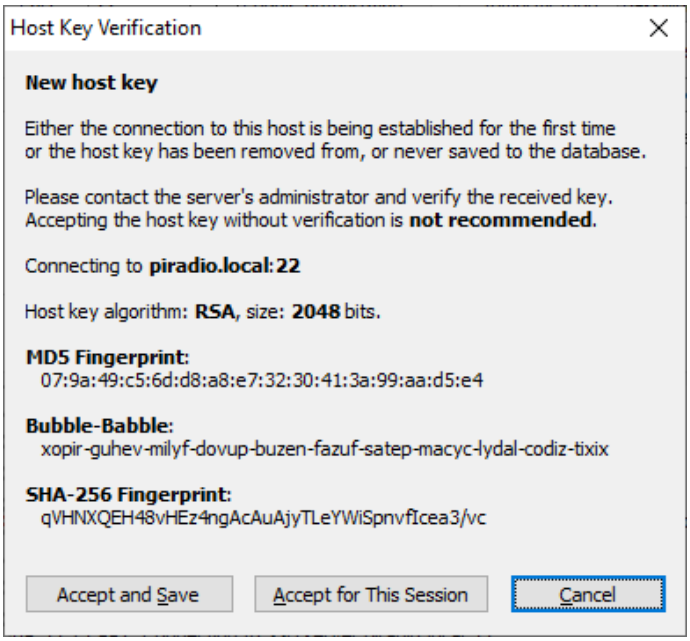
Bitvise is a free SSH client available for Windows or Mac to connect SSH server enabled Unix or Linux operating systems. It is a graphical based SSH client just like **Putty** with more features. It supports the File Transfer using SFTP Secure File Transfer Protocol). Once installed you can easily make a terminal connection to the Raspberry Pi. To find out the IP address of the Raspberry Pi using **Fing** which is described in the next section.

See https://www.bitvise.com/ssh-client for more information.

Alternatively, if you have installed the Desktop version of the OS then you can use the terminal application to log into the RPi.

Log into the Raspberry Pi using **Bitvise** or **Putty** or if booting up using a keyboard mouse and HDMI monitor or TV run the terminal program (top left black icon).
If logging in using **Bitvise** you will need to know the IP address of the Raspberry Pi which can be found using **Fing**. See *Finding the Raspberry Pi on a network using Fing* on page 19. You can also try

using piradio, piradio.local or piradio.home or piradio.lan as the hostname. The correct extension (local,home,lan etc.) will depend upon your router.



Once connected in click on “Accept and Save” to save the new host key (Used by SSH for encryption)

The user’s name is **pi** and the default password is **raspberry** (Unless you changed it). Once logged in the following screen will be displayed:

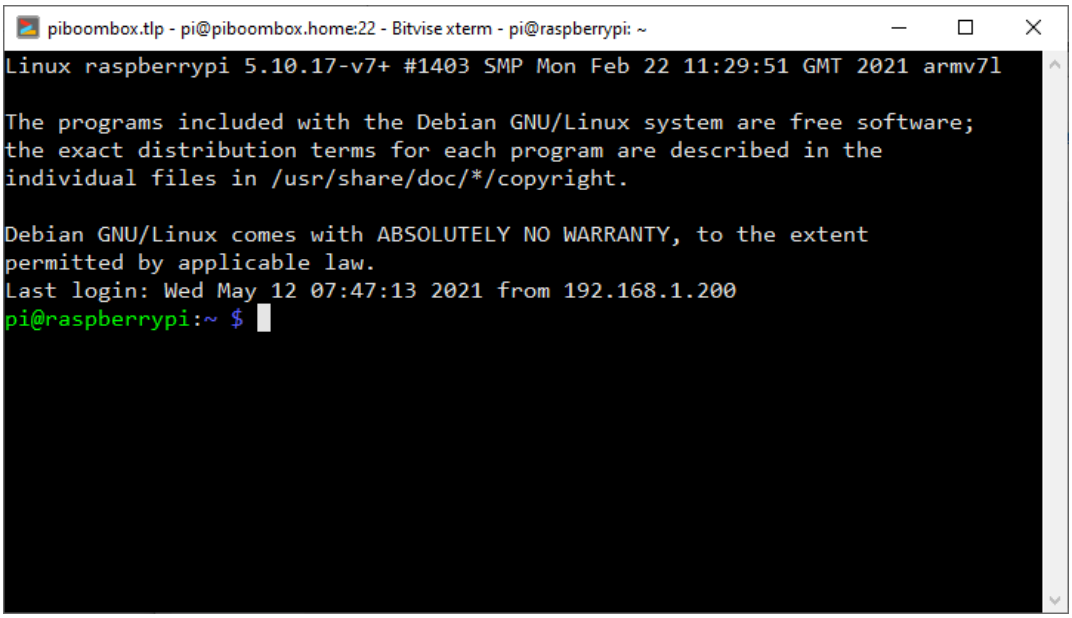


Figure 20 Raspberry Pi Command Line

You are now in the Raspberry Pi OS (Linux) command line. Linux is the Operating System that the Raspberry Pi OS is based upon. A guide to basic Linux commands can be found at: https://www.raspberrypi.org/documentation/linux/usage/commands.md

Update to the latest the packages

Unless you did the OS installation via the Linux desktop and updated the packages already, run the following command to update the packages list. If in doubt run the following command anyway.

```
$ sudo apt-get update
```

Run the following command to upgrade to the latest packages for this release.

```
$ sudo apt-get upgrade
```

This will take some time so be patient.

Disable PiWiz screen reader

Since December 2020 release, the **Raspberry OS** developers have installed the **PiWiz** screen reader. As a result, if using the desktop on an HDMI monitor, every 15 seconds a computerized voice says "To install the screen reader press control alt space".

The PiWiz package is not compatible with the radio software. It affects operation particularly on start-up and must be removed or otherwise disable. People with impaired vision can make use of the e-Speak facility provided with this package.

This feature can be disabled during installation using the desktop interface. Press CTL, Alt and Space bar together.

However, if you are using a headless installation (No HDMI screen and desktop) it may not be disabled. The message is not only very annoying, but hogs the sound system meaning that the radio (MPD) cannot play audio content. To disable PiWiz run the following.

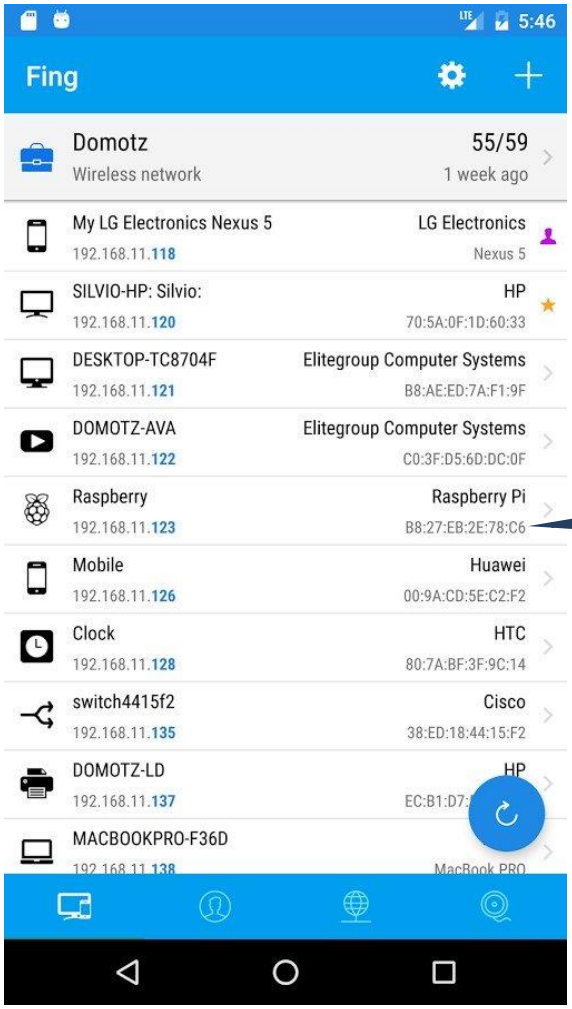
```
$ sudo rm /etc/xdg/autostart/piwiz.desktop
```

Reboot

```
$ sudo reboot
```

Useful tools

Finding the Raspberry Pi on a network using Fing



The **Fing** App is a free network toolkit and scanner for iOS (Apple) and Android. It will discover all the devices on your network, identify intruders and can also run Internet speed tests.

In this context it is very useful for finding out the IP address of any Raspberry Pi connected to the network. It will only see it of course once the Raspberry Pi has been set up first on the Wireless Network or hard-wired in. It scans both Wireless and hard-wired Ethernet devices.

Raspberry Pi detected

Once detected it is the possible to the SSH to the Raspberry Pi using **Bitvise** or **Putty**.

See https://www.fing.com/products/fing-app for further information.

There are also a number of Wi-Fi scanner Apps that also display the devices connected to the local LAN which can be used instead of **Fing**.

Radio software installation

Conventions used in this tutorial

Installation of the radio program requires you to enter lines at the command line prompt. This requires you to log into the Raspberry PI as user '**pi**'. The default password is **raspberry**.

Note: Don't carry out any of the following commands just yet. They are just examples.

```
Raspberrypi login: pi
Password: raspberry
pi@raspberrypi:~$ Last login: Wed Sep  9 12:17:35 2020 from 192.168.1.200
pi@raspberrypi:~$
```

The prompt line is displayed ending with a **$** sign. The **pi@raspberrypi:~** string means user 'pi' on host machine called 'raspberrypi'. The ~ character means the user 'pi' home directory **/home/pi**. In this tutorial if you are required to do something as user **pi** then only the $ sign will be shown followed by the command as shown in the example below:

```
$ mpc status
```

Copy and paste the highlighted text only to the command line. Omit the $ sign.

Some commands produce a lot output which does not need to be shown. In such a case a colon (':') is used to indicate that some output has been omitted.

```
$ mpd -V
Music Player Daemon 0.22.6 (0.22.6)
Copyright 2003-2007 Warren Dukes <warren.dukes@gmail.com>
Copyright 2008-2018 Max Kellermann <max.kellermann@gmail.com>
:
Other features:
 avahi dbus udisks epoll icu inotify ipv6 systemd tcp un
```

Note that when a command is shown with the resulting output the command which was entered is shown in **bold**.

You will see a lot of commands with the word sudo, for example:

```
$ sudo systemctl start radiod
```

This gives the user pi permissions to carry out instructions normally only allowed by the root user.

END OF EXAMPLE COMMANDS.

Installing the Music player daemon

The radio software uses the Music Player Daemon (MPD). More information on MPD can be found at https://www.musicpd.org/
After booting up the RPi, install the Music Player Daemon (mpd) and its client (mpc) along with the Python3 and **python3-rpi.gpio** library.

```
$ sudo apt-get install mpd mpc python3-mpd python3-rpi.gpio
```

Answer yes 'y' when asked to continue.

Installing the Radio Daemon

Before starting uninstall **pulseaudio** as it interferes with the operation of the radio:

```
$ sudo apt-get remove pulseaudio
```

The Raspberry PI Internet Radio software is distributed as a Debian package. This can be downloaded from http://www.bobrathbone.com/raspberrypi/pi_internet_radio.html
Either download it to your PC or Macintosh and copy it to the **/home/pi** directory or get it directly using the **wget** facility.

```
$ wget http://www.bobrathbone.com/raspberrypi/packages/radiod_7.4_armhf.deb
```

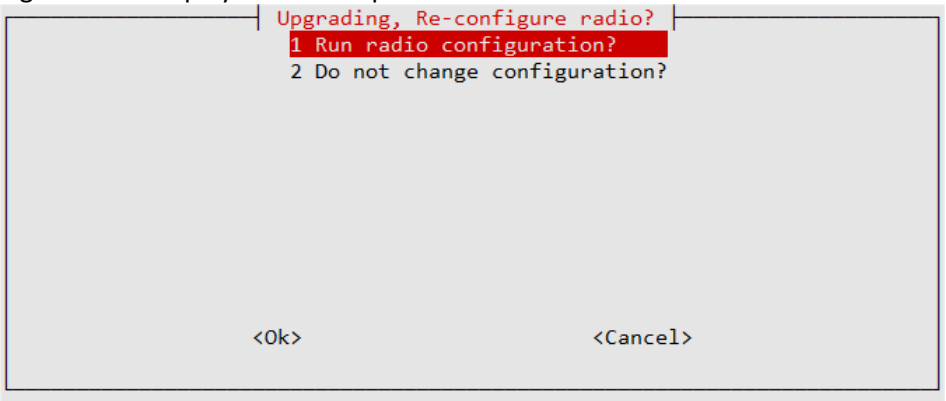
Run **dpkg** to install the **radiod** package.

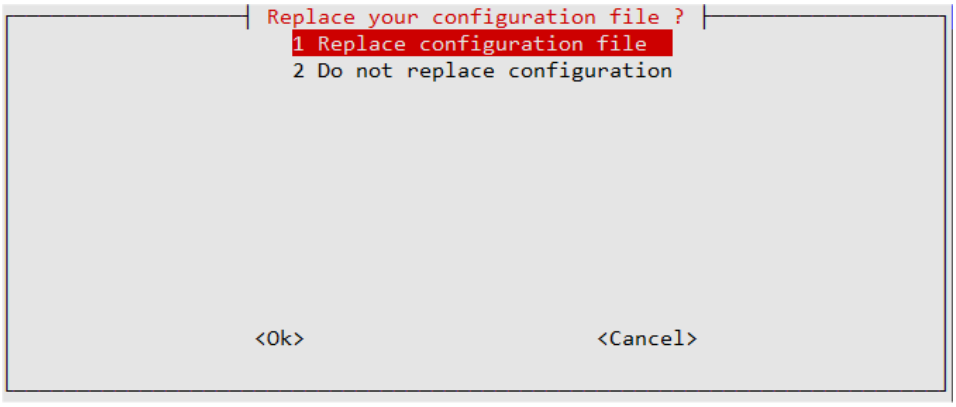
```
$ sudo dpkg -i radiod_7.4_armhf.deb
```

The **dpkg** program will install the files.

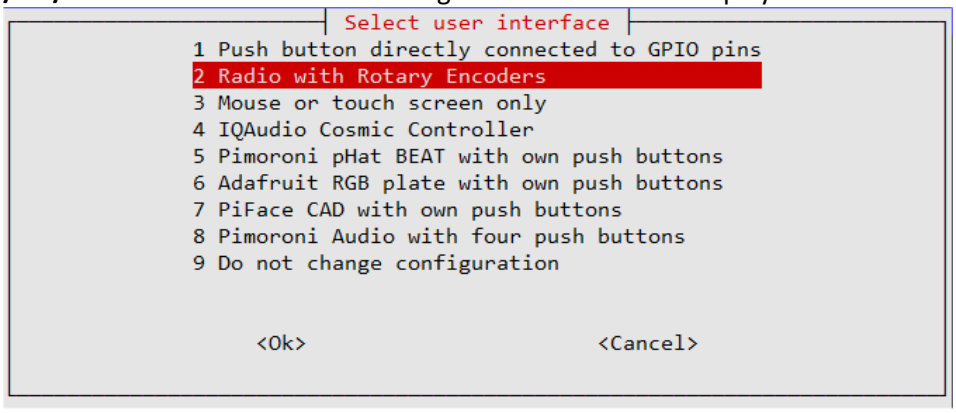
```
(Reading database ... 131542 files and directories currently
installed.Preparing to unpack radiod_7.4_armhf.deb ...
Raspberry PI internet radio installation
Stopping radiod service
Unpacking radiod (7.4)
```

The following screen is displayed. Select option 1 to continue.

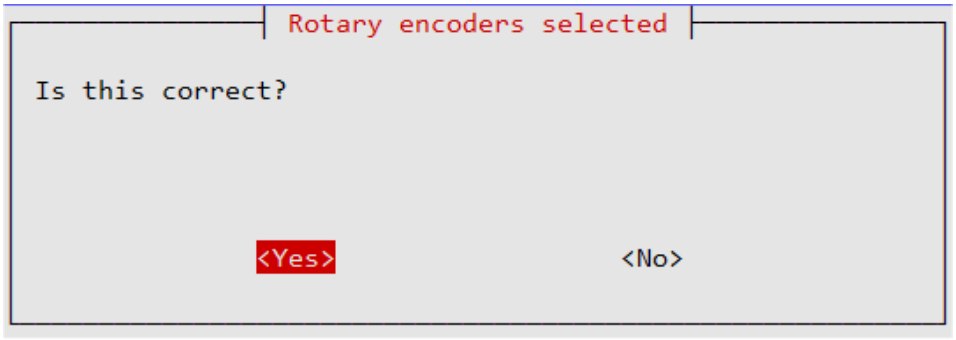




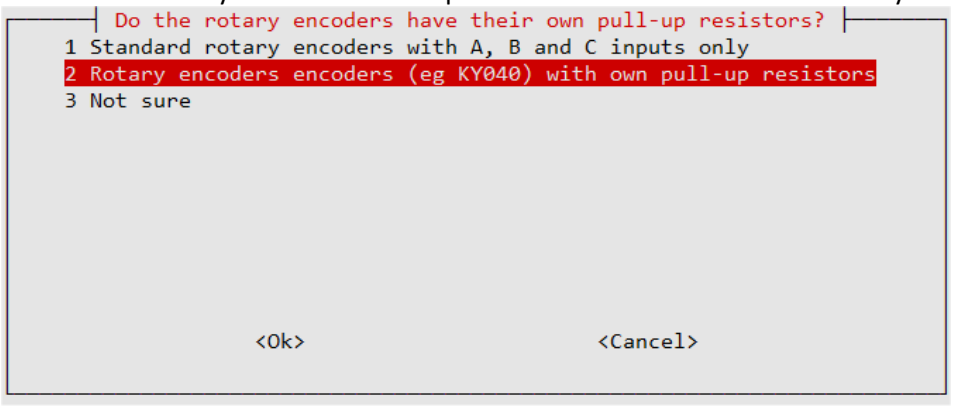
Select option 1 to replace the configuration file. A backup copy of the original configuration is written to **/etc/radiod.conf.save**. The following screen will then be displayed:

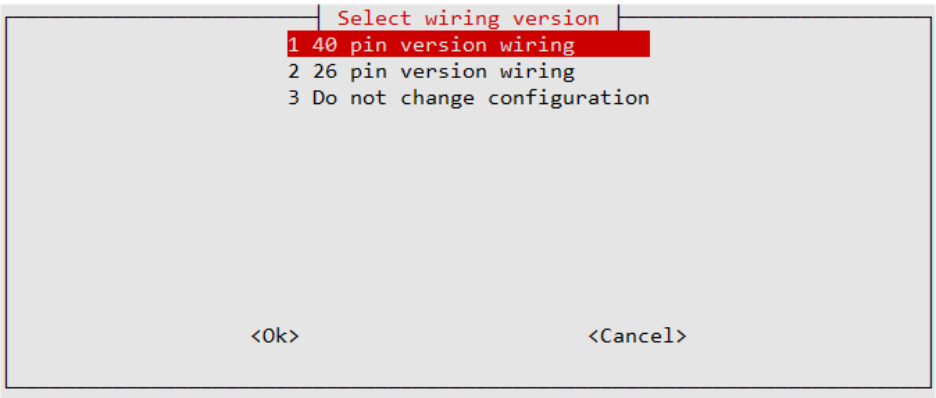


Select option 2 “Radio with rotary encoders” followed by “OK”. Confirm your choice on the next screen:

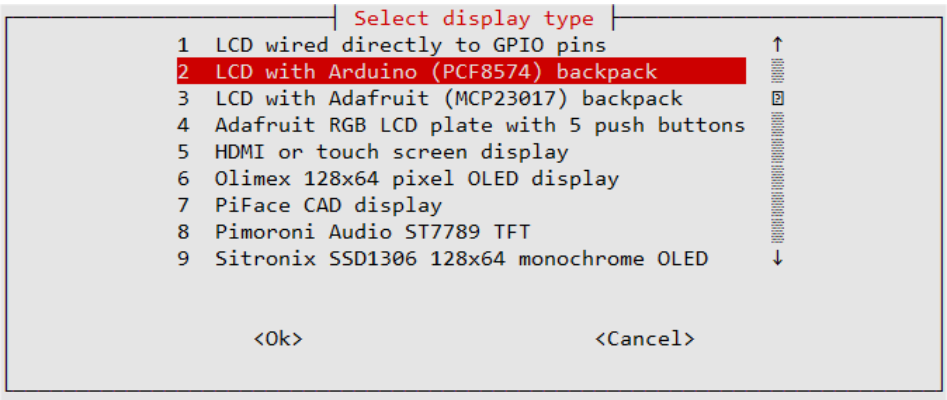


If you have used KY040 rotary encoders select option 2. You will be asked to confirm your choice.





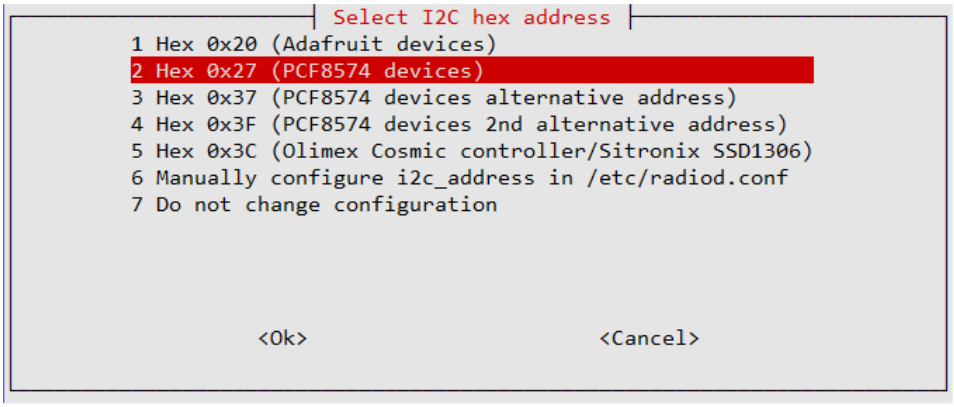
Select option 1 “40 pin wiring” and confirm your choice. Option 2 is for very old 26-pin Raspberry Pi’s



Select option 2 “LCD with Arduino (PCF8547) backpack.

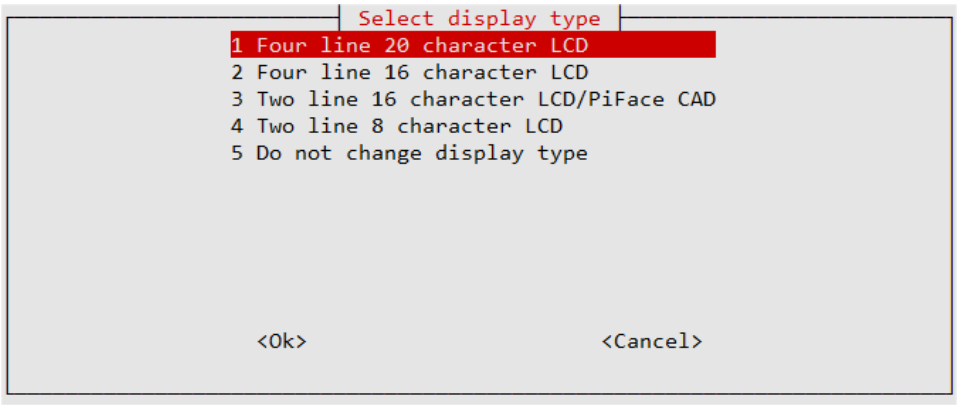
Notice that there is a scroll bar to the right of the options. Use the Up/Down keys to scroll down to the remaining options.

Now select the hex address for the I2C backpack. This is option 2 0x27 (PCF8574 devices).



The installation program now configures the **I2C Device Overlay** (Driver software) to be loaded at boot time.

The program now asks you to select the display type.

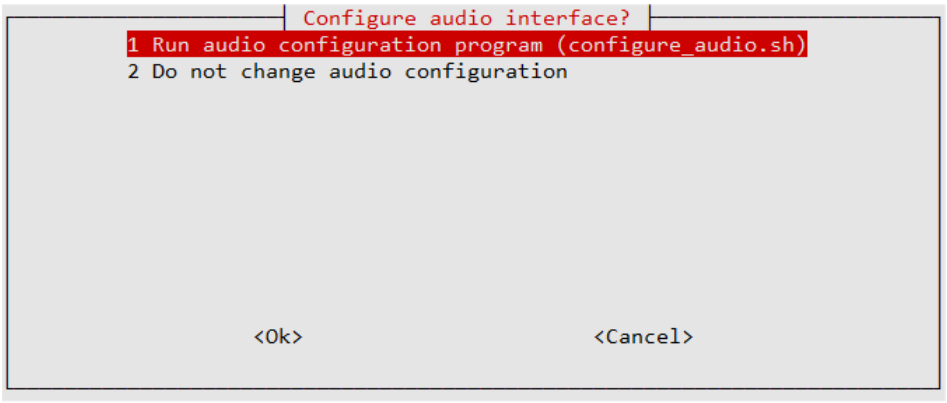


Select either option 1 or 2 for the type of display you are using.

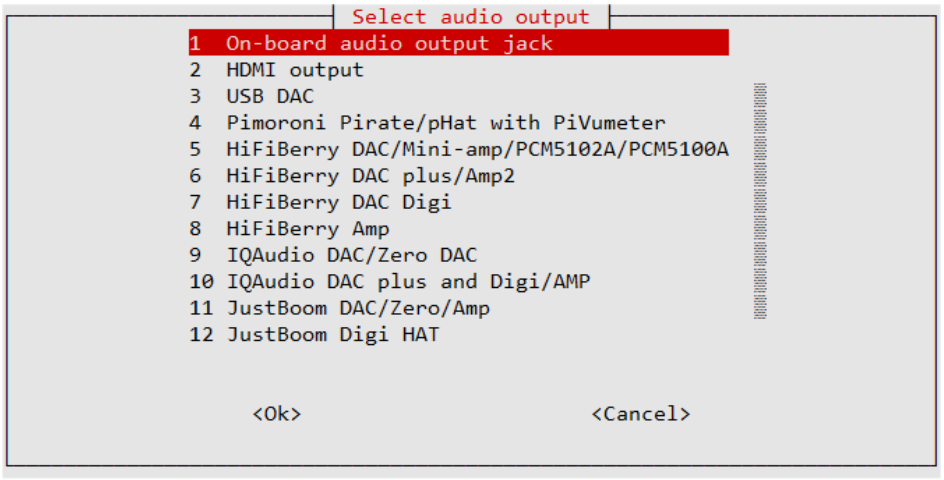
The installation program will display a summary oy the actions it is carrying out.

```
Enabling I2C interface in /boot/config.txt
Original /etc/radiod.conf copied to /etc/radiod.conf.org
LCD pinouts disabled
Configuring 40 Pin wiring
Configuring splash screen as bitmaps/raspberry-pi-logo.bmp

Changes written to /etc/radiod.conf
----------------------------------
user_interface=rotary_encoder
display_type=LCD_I2C_PCF8574
display_lines=4
display_width=20

lcd_select=0
lcd_enable=0
lcd_data4=0
lcd_data5=0
lcd_data6=0
lcd_data7=0

menu_switch=17
mute_switch=4
up_switch=24
down_switch=23
left_switch=14
right_switch=15

pull_up_down=up
flip_display_vertically=no
splash=bitmaps/raspberry-pi-logo.bmp
volume_range=20
----------------------------------

sudo systemctl enable radiod.service
sudo systemctl disable mpd.service
sudo systemctl enable mpd.socket
Daemon radiod configured
```

The installation program will now ask you to configure the audio output. In this project we will be using the on-board audio output jack.



Select option 1 to configure the audio output.



Select option 1: On-board audio output jack or option 3 for USB connected loud-speakers.

The installation program displays the following (Some output omitted).

```
/usr/share/radio/configure_audio.sh configuration log, Wed Jun  2 07:29:38
BST 2021
sudo systemctl stop radiod.service
sudo systemctl stop mpd.service
sudo systemctl stop mpd.socket
On-board audio output Jack selected
Card 0, Device hw:0,0, Mixer software
:
:
PI Radio software successfully installed
See /usr/share/doc/radiod/README for release information

Installation complete
It is necessary to reboot the system to start the radio
```

Reboot the Raspberry Pi.

```
$ sudo reboot
```

Operation

Starting and stopping the radio

The **radiod** program should start automatically after boot up.

To stop the radio from the command line (Log in first with SSH).

```
$ sudo systemctl stop radiod
```

To start the radio

```
$ sudo systemctl start radiod
```

To prevent the radio from starting up at boot time

```
$ sudo systemctl disable radiod
```

To re-enable it at boot time.

```
$ sudo systemctl enable radiod
```

Radio operation

The radio is operated using the two Rotary Encoder Knobs.

The left-hand knob controls volume. Pressing the knob inwards mutes and unmutes the radio. The right-hand knob changes stations/mp3 tracks. Pressing the knob inwards steps through the menu.

Menu operation

This option is for a radio with rotary encoders with push buttons. The volume knob when pushed in is the **Mute** sound function. Likewise, the tuner knob when pushed in is the **Menu** switch.
The Menu button (Tuner knob depressed) changes the display mode and the functions of the clockwise and anti-clockwise operation of the knobs as shown in the following table.

Table 4 Rotary Encoder Knob Operation

	Volume knob		Tuner knob	
LCD Display Mode	Clockwise	Anti-clockwise	Clockwise	Anti-clockwise
Mode = TIME **Line 1: Time** **Line 2: Station or Track**	Volume Up	Volume Down	Station/Track up	Station/Track down
Mode = SEARCH **If source = RADIO** **Line 1: Search:** **Line2: Radio Station**	Volume Up	Volume Down	Scroll up radio station	Scroll down radio station
Mode = SEARCH **If source = MUSIC LIBRARY** **Line 1: Search** **Line2: MusicTrack/Artist**	Scroll up through artists	Scroll down through artists	Scroll up through track	Scroll down through track
Mode = SOURCE **Line 1: Input Source:** **Line2: Radio or Media playlist or Airplay**	Volume Up Mute	Volume Down Mute	Cycle up through Airplay, Radio and Media playlists	Cycle down through Airplay, Radio and Media playlists
Mode = OPTIONS **Line 1: Menu Selection** **Line 2: <option>** **Options are Random, Consume, Repeat, Reload Music, Timer, Alarm and Alarm Time (Hours), Alarm Time (Minutes) set and Change colour (1), Streaming on/off.**	Toggle selected mode on or off. Set timer and Alarm	Toggle selected mode on or off. Set timer and Alarm	Cycle through Random, Consume, Repeat, Reload Music, Timer, Alarm Time Set, Streaming and Background colour(1)	Cycle through Random, Consume, Repeat, Reload Music, Timer, Alarm , Alarm Time Set, Streaming and Background colour(1)
Mode = RSS (2) **Line 1: Time** **Line 2: RSS feed**	Volume Up	Volume Down	Station/Track up	Station/Track down
MODE = IP address **Line 1: IP address** **Line 2: Station or Track**	Volume Up	Volume Down	Scroll up through track or radio station	Scroll down through track or radio station

The above table shows operation for a two-line display. If using a four-line LCD then then following lines are used:
Line 1: Time display/Menu selection
Line 2: Station or artist name/Menu options
Line 3: Station information (song playing etc)/Track
Line 4: Volume display

Playlist creation

There are two types of playlists namely *Radio Streams* or *Media* (MP3) files. The both types of MPD playlist are stored in the **/var/lib/mpd/playlists** directory.

The radio playlist

The **/var/lib/radiod/stationlist** file is the file that should be maintained by you to create Radio playlists. When this *create_stations.py* program is first run it copies the distribution file **station.urls** to the **/var/lib/radiod/stationlist** file. You may then modify the **/var/lib/radiod/stationlist** file.

The format is: **(<playlist name>)**
Example: **(Radio)**

The above will create a playlist called **_Radio.m3u** and will contain the title and URLs for each station. Now add or remove radio station definitions in the **stationlist** file. The first statement in the station definition is the name of the playlist in brackets:

The format is: **[<title>] http://<url>**
Example: **[BBC Radio 4 extra] http://www.bbc.co.uk/radio/listen/live/r4x.asx**

After modifying the **stationlist** file run the *create_stations.py* program to create the Music Player Daemon playlists.

Note: When installing the radio software for the first time a file called **station.urls** will be copied to the **stationlist** file. It will not be overwritten when upgrading or re-installing the software. The user is totally responsible for maintaining the **stationlist** file from then on.

Below is an example of part of a **stationlist** file stored in **/var/lib/radiod** directory. This file is the source of all radio playlists:

```
# Radio stations
(Radio)
# United Kingdom
# The following links are iPhone streams (m3u files)
[BBC Radio 1] http://www.radiofeeds.co.uk/bbcradio1.pls
[BBC Radio 2] http://www.radiofeeds.co.uk/bbcradio2.pls
[BBC Radio 3] http://www.radiofeeds.co.uk/bbcradio3.pls

# Dutch stations
[NPO Radio 1] http://icecast.omroep.nl/radio1-bb-mp3
[NPO Radio 2] http://icecast.omroep.nl/radio2-bb-mp3
[NPO Radio 3fm] http://icecast.omroep.nl/3fm-bb-mp3
```

In the above example the Radio playlist is defined by the name in round brackets namely; (Radio).

The *create_stations.py* program itself is very easy to use. Just run it with **sudo** in the **/usr/share/radio** directory:

```
$ cd /usr/share/radio
$ sudo ./create_stations.py
```

This will create the playlist files in the **/var/lib/mpd/playlists** directory. Using the example shown above this will produce a file called **_Radio.m3u** in the MPD playlists directory.

To edit the the **/var/lib/radiod/stationlist** file use the **nano** editor program. See *Appendix A – Editing files with nano* on page 38.

Creating media playlists

The radio software also allows you to play music (mp3) from a USB stick.

A playlist for the USB stick can be created using the **create_playlist.sh** program.

```
$ cd /usr/share/radio
$ sudo ./create_playlist.sh
```

Put your music tracks on a USB stick (MP3 and WMA files only) and insert it into the USB port of the Raspberry PI. Run the **create_playlist.sh** program as shown above.

Playing media tracks from the USB stick

Reboot the Raspberry PI. Once the Radio program is running again, push the Menu button until "Input source" is displayed. Press either the left or right button to change the source to "Music Library". Now press the Menu button again. The music on the USB stick will now be loaded and played.

Chapter 3 - Troubleshooting

There is an extensive troubleshooting guide in the Raspberry Pi Internet Radio Constructors Guide. See https://bobrathbone.com/raspberrypi/documents/Raspberry%20PI%20Radio.pdf

No network

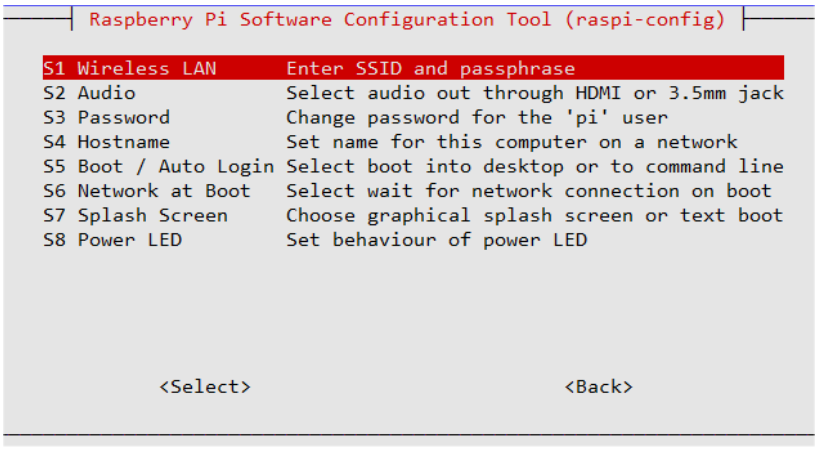
If the problem is with Wi-Fi check the **/etc/wpa_supplicant/wpa_supplicant.conf** configuration file.

```
$ cat /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB

network={
        ssid="<Your-SSID>"
        psk="<Your-Router-Password>"
        key_mgmt=WPA-PSK
}
```

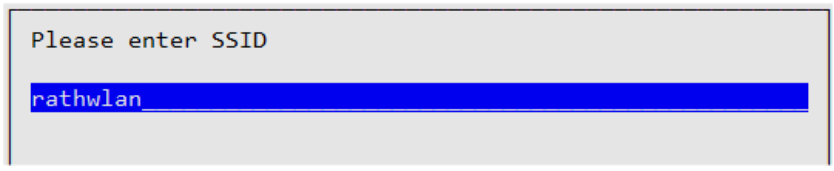
Either edit the above file directly or run **raspi-config** to configure the Wi-Fi interface.

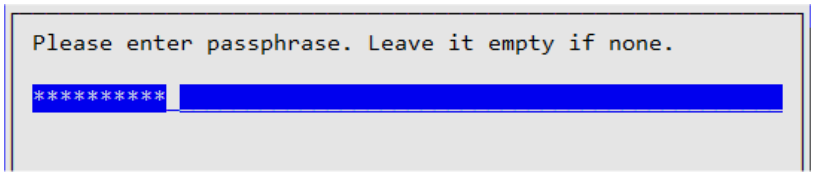
```
$ sudo raspi-config
```

Select option 1 System options followed by option S1 Wireless LAN:



Now enter the SSID and passphrase for your network. Save the settings and reboot the Raspberry PI. In this e





MPD fails to start

The MPD daemon logs to the **/var/log/mpd/mpd.log** file. Examine this file for errors. The MPD daemon is dependent on good M3U files so check that these are correct as described in the section called *Playlist creation* on page 28.

Run the following command to display the status of the MPD daemon.

```
$ systemctl status mpd
● mpd.service - Music Player Daemon
   Loaded: loaded (/usr/local/lib/systemd/system/mpd.service; disabled;
vendor preset: enabled)
   Active: active (running) since Sat 2021-07-03 12:06:44 BST; 2 days ago
     Docs: man:mpd(1)
           man:mpd.conf(5)
  Process: 874 ExecStartPre=/bin/chown mpd:audio /var/run/mpd (code=exited,
status=0/SUCCESS)
 Main PID: 875 (mpd)
    Tasks: 6 (limit: 1431)
   CGroup: /system.slice/mpd.service
           └─875 /usr/local/bin/mpd --no-daemon /etc/mpd.conf

Jul 03 12:06:43 raspberrypi systemd[1]: Starting Music Player Daemon...
Jul 03 12:06:44 raspberrypi mpd[875]: Jul 03 12:06 : exception: Decoder
plugin 'wildmidi' is unavailable
Jul 03 12:06:44 raspberrypi systemd[1]: Started Music Player Daemon.
```

If any errors occurred, they will be displayed by the status command.

No output on LCD screen

Check that the I2C driver sees the Adafruit I2C backpack. Run the **i2cdetect** program.

```
$ i2cdetect -y 1
```

The following output should be seen. In particular the device at address 0x27.

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- 27 -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- UU -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

If not check that the i2c device driver is loaded.

```
$ ls -la /dev/i2c-1
crw-rw---- 1 root i2c 89, 1 Jul  3 11:17 /dev/i2c-1
```

If not found first check the backpack wiring, if OK re-run the radio configuration program.

```
$ cd /usr/share/radio
$ sudo ./configure_radio.sh
```

Check that I2C is configured in **/boot/config.txt**. Edit it with nano.

```
$ sudo nano /boot/config.txt
```

It should contain the following line without a # at the beginning.

```
dtparam=i2c_arm=on
```

Check that the I2C driver module is present in /etc/modules

```
$ cat /etc/modules
:
i2c-dev
```

Now run the test program **lcd_2c_pcf8574.py**.

```
$ cd /usr/share/radio
$ ./lcd_i2c_pcf8574.py
```

Output should be seen on the LCD. If not then it may be that the blue contrast adjustment is turned down low.

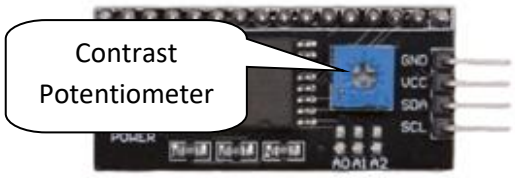


Figure 21 Adjusting the contrast

The first thing to do is to adjust the contrast potentiometer. Turn fully right and left to adjust the contrast.

Rotary encoders not working

Run the following:

```
$ cd /usr/share/radio
$ ./rotary_class.py
```

The following will be displayed:

```
Test rotary encoder Class
Left switch GPIO 14
Right switch GPIO 15
Up switch GPIO 24
Down switch GPIO 23
Mute switch GPIO 4
Menu switch GPIO 17
```

Now turn each of the rotary encoders and press the encoder knobs.

```
Tuner event  1 CLOCKWISE
Tuner event  1 CLOCKWISE
Tuner event  1 CLOCKWISE
Tuner event  1 CLOCKWISE
Tuner event  2 ANTICLOCKWISE
Tuner event  2 ANTICLOCKWISE
Tuner event  2 ANTICLOCKWISE
Tuner event  3 BUTTON DOWN
Volume event  1 CLOCKWISE
Volume event  1 CLOCKWISE
Volume event  1 CLOCKWISE
Volume event  2 ANTICLOCKWISE
Volume event  2 ANTICLOCKWISE
Volume event  2 ANTICLOCKWISE
Volume event  3 BUTTON DOWN
```

Press Ctrl and C to exit the program.
Correct any wiring faults, in particular jumper wires which have dropped off and re-test.

No sound coming out of the on-board audio socket

Have you got the Raspberry Pi plugged into the HDMI monitor? If so un-plug it and reboot.

Check that the on-board output jack is configured correctly.

```
$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: Headphones [bcm2835 Headphones], device 0: bcm2835 Headphones
[bcm2835 Headphones]
  Subdevices: 7/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
```

If the above isn't seen then run the audio configurator program.

```
$ cd /usr/share/radio
$ sudo ./configure_audio.sh
```

Reboot the radio:

```
$ sudo reboot
```

MPD Logging

All logging for the MPD daemon is to the **/var/log/mpd/mpd.log** file by default.

Radio program logging

The running Radio program logs to a file called **/var/log/radio.log.** You can observe it with the **tail** command. See example log below:

```
$ tail -f /var/log/radio.log
2020-04-22 15:11:40,906 INFO ===== Starting radio =====
2020-04-22 15:11:40,912 INFO Translation code page in radiod.conf = 0
2020-04-22 15:11:41,296 INFO Display code page 0x2
2020-04-22 15:11:41,297 INFO Loaded 'codes.Russian'
2020-04-22 15:11:41,297 INFO Loaded 'codes.European'
2020-04-22 15:11:41,297 INFO Loaded 'codes.English'
2020-04-22 15:11:41,298 INFO Screen LCD Lines=4 Width=20
2020-04-22 15:11:43,467 INFO Romanize True
2020-04-22 15:11:44,676 INFO Board revision 2
2020-04-22 15:11:44,695 INFO OS release: Raspbian GNU/Linux 10 (Bullseye)
2020-04-22 15:11:44,704 INFO Linux Bullseye01 4.19.97-v7l+ #1294 SMP Thu Jan
30 13:21:14 GMT 2020 armv7l GNU/Linux
2020-04-22 15:11:44,705 INFO radio.startMpdDaemon: Starting MPD
2020-04-22 15:11:46,345 INFO radio.startMpdDaemon: MPD started pid=1375
2020-04-22 15:11:46,377 INFO Connected to MPD port 6600
2020-04-22 15:11:46,564 INFO UDP Server listening on localhost port 5100
2020-04-22 15:11:46,569 INFO UDP listen:remote 0.0.0.0 port 5100
2020-04-22 15:11:46,569 INFO IP 192.168.1.152 192.168.1.153
2020-04-22 15:11:53,801 INFO Radio ['/usr/share/radio/radiod.py',
'nodaemon'] Version 7.4
2020-04-22 15:11:53,802 INFO Radio running pid 1275
```

There are six levels of logging namely CRITICAL, ERROR, WARNING, INFO, DEBUG or NONE. This is configured in the **/etc/radiod.conf** file. Use DEBUG for more information.

```
# loglevel is CRITICAL,ERROR,WARNING,INFO,DEBUG or NONE
loglevel=INFO
```

To switch on debugging change the **loglevel** statement in the **/etc/radiod.conf** file.

```
# loglevel is CRITICAL,ERROR,WARNING,INFO,DEBUG or NONE
loglevel=DEBUG
```

Technical support

Technical support is on a voluntary basis by e-mail only at bob@bobrathbone.com. If there are any problems with this email address then also CC r.h.rathbone@gmail.com. Before asking for support, please first consult the troubleshooting section on page 30. I will always respond to e-mails requesting help and will never ignore them. I only ask that you do the same (i.e. Did my suggestions help or not?). Be sure to provide the following information:

- A clear description of the fault.
- What you have already done to locate the problem?
- Is anything displayed on the LCD or Graphics screen?
- Did you run the test programs and what was the result?
- Switch on DEBUG logging as described on page 34, run the program and include the **/var/log/radio.log** file.
- Did you vary from the procedure in the manual or add any other software?
- Please supply any information requested.

Run the configuration display and send the **/usr/share/radio/config.log.tar.gz** that it produces to bob@bobrathbone.com. This will save a lot of questions about your configuration.

```
$ cd /usr/share/radio
$ ./display_config.sh
:
This configuration has been recorded in /usr/share/radio/config.log
A compressed tar file has been saved in /usr/share/radio/config.log.tar.gz
Send /usr/share/radio/config.log.tar.gz to bob@bobrathbone.com if required
```

Please note that support for general Raspberry PI problems is not provided. Only issues relating to the Radio software will be investigated.

For general Raspberry PI support see the following site:
http://www.raspberrypi.org/forums/

For support on Music Player Daemon issues see the help pages at the following link:
http://www.musicpd.org/

Chapter 4 - Licences, disclaimer and support

Licences

The software and documentation for this project is released under the GNU General Public Licence.

See http://www.gnu.org/licenses for up-to-date information on the GNU General Public License.

The licences for the source and documentation for this project are:
GNU General Public License. See http://www.gnu.org/licenses/gpl.html
GNU AFFERO General Public License. See http://www.gnu.org/licenses/agpl.html
GNU Free Documentation License. See http://www.gnu.org/licenses/fdl.html
MPD uses an older version of the license at https://www.gnu.org/licenses/gpl-2.0.txt

Disclaimer

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BELIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE OR DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Support

The author does not directly support the Music Player Daemon software. You are advised to contact the MPD forum at https://forum.musicpd.org/. However, if you have any comments or feedback about this procedure or document, please contact bob@bobrathbone.com

Acknowledgements and Copyrights

Thanks to the Music Player Daemon (MPD) foundation for their excellent audio software. See https://www.musicpd.org/

To Luboš Ruckl for his work on the Rotary encoder class (adapted from code by Ben Buxton) and the PCF8574 LCD class (adapted from code by an unknown author but believed to be from the Arduino community).

Thanks to Joaquin Perez for the photograph of his Raspberry Pi Internet radio on the front page. Ashley Yates for the photograph of his Raspberry Pi Internet radio in the instructions section.

To all constructors of this project who have sent in photos of their radio's and their ideas for improvement and the many appreciative e-mails that I have received from them.

References

Raspberry Pi Internet Radio Constructor's manual.
https://www.bobrathbone.com/raspberrypi/documents/Raspberry%20PI%20Radio.pdf

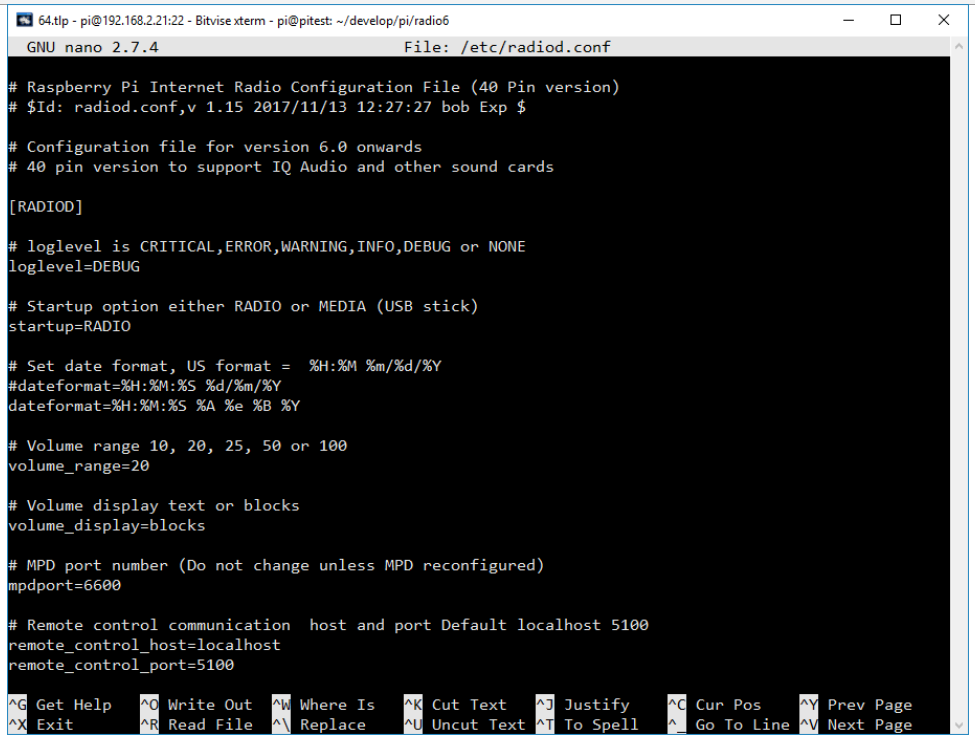
Glossary

HDMI	High-Definition Multimedia Interface – connection to HD monitors and TVs.
I2C	A two wire, multiple device communication protocol developed by Philips Industries
RPi	Raspberry Pi
SD	Secure Digital – Type of non-volatile memory card
SIL	Single in Line Header, either male or female.
SSH	Secure Shell – Encrypted terminal
SSID	Service Set Identifier. An SSID is the public name of a wireless network.

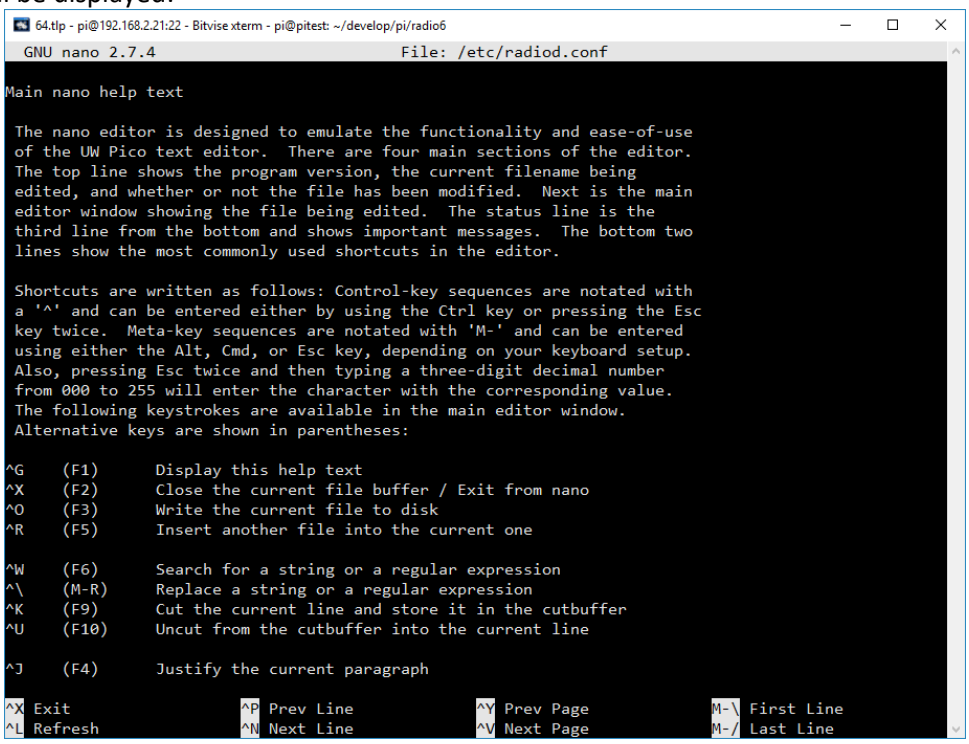
Appendix A – Editing files with nano

The program used in this tutorial is called **nano**. For example, editing **/etc/radiod.conf**.

```
$ sudo nano /etc/radiod.conf
```



Hold down the Ctrl key and press the letter G on the keyboard to display the help text. The following screen will be displayed:



The ^ character means the Control-key (Ctrl). So, for example ^O above is Ctrl + O. For more information on **nano** see https://www.nano-editor.org/dist/v2.0/nano.html

Index