# Raspberry PI Internet Radio

## Vintage Radio Supplement

Bob Rathbone Computer Consultancy

www.bobrathbone.com

23rd of June 2021

# Contents

# Figures

## Tables

# Introduction

This manual is a supplement to the Raspberry Pi Internet Radio Constructors Radio. It describes how to convert a Vintage Radio to an Internet Radio using the Raspberry PI educational computer. The source and basic construction details are available from the following web site:
http://www.bobrathbone.com/raspberrypi/pi_internet_radio.html

This manual is not standalone and must be used in conjunction with the main Raspberry Pi Internet Radio Constructors Guide. This can be downloaded from:
http://www.bobrathbone.com/raspberrypi/documents/Raspberry%20PI%20Radio.pdf

This document is not a step-by-step instruction set but is a collection of ideas that were used in this design and may be useful for similar projects. The radio used for this project was a Philips BX490A manufactured in the Netherlands in 1949. Alternative solutions for connecting a vintage radio to the Internet are also shown in this manual (See the section called *Alternative solutions* on page 25).

**Figure 1 Philips BX490A Bakelite Radio 1949**

The Philips BX490A is a six-valve radio with 6 bands namely Long wave, Medium wave and 4 Short wave bands which were selected by the right-hand switch. It also had a so-called Magic Eye tuning indicator.

## Repair or convert?

The decision to convert the Philips BX490A vintage radio instead of trying to repair it was not an easy one. In this particular case the potentiometers were either seized up or falling apart. Wires were snapped and only the Medium Wave band was working. The sound was crackling and unreliable. Also, a hole had been drilled in the side to mount an ugly on-off switch because the one on the front of the radio was broken. Also, the so-called magic eye for tuning indication was no longer working. Clearly a lot of effort was necessary to restore it. It was decided to convert it to an Internet radio whilst maintaining the original look and feel of the radio.

# Construction

Please note that most vintage valve radios use high voltages up to 240 Volt.
Before starting on your project please refer to the section called Safety on page 33.

## Stripping out the old internals

The first step taken was to strip out the old workings of the radio but to preserve the tuning mechanism for use in the Internet radio.

Figure 2 Stripping out old the workings of the radio

All of the control knobs were removed from the front of the radio. In this case the front plate which contains the speaker and tuning scale separated easily from the main radio chassis. The tuning fly-wheel and tuning mechanism is still attached to the right-hand side of the radio chassis. The tuning scale wire which attaches to the tuning scale on the front of the radio must be carefully disconnected. Make careful note of how the wire is threaded before disconnecting it.

## Using the original speaker(s)

In this case there was only a single loudspeaker so it was decided to replace the original speaker with two smaller speakers. Some radios have two loudspeakers, for example radios with FM stereo.  If the audio stage of the radio is still working then it is possible to feed the output of the Raspberry Pi to the auxiliary or gramophone input of the old radio and thus preserving the original sound of the radio. In such a case the original volume and tone controls of the radio can be used and the volume rotary encoder can be omitted and the volume out of the Raspberry Pi set to a pre-set level.

If only one speaker is available then the stereo output of the Raspberry Pi needs to be converted to Mono and a single Mono amplifier used. See the section Converting stereo output to mono on page 21.

## The basic radio case ready for construction

The following illustration shows the radio case with the original tuning mechanism stripped out. It was only the fly-wheel (left) and the wheel that drives the scale cable which was of interest. The tuning capacitor (right) was disconnected from the flywheel mechanism.

Figure 3 The tuning mechanism

## Fitting a new base

A new wooden base was mounted on the front panel using two stout metal brackets fastened with nuts and bolts.

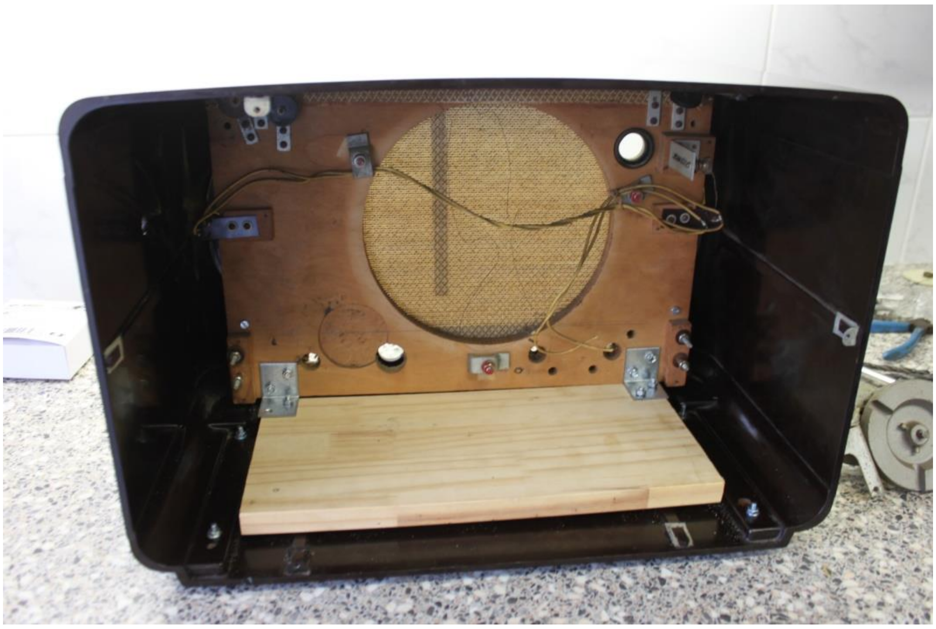


Figure 4 Fitting a new wooden base

The above picture shows the yellow wiring for the front panel lamps (6 volt) and the pulleys for the cable that drives the front scale.

## Mounting the tuning mechanism

Figure 5 below shows the tuning mechanism mounted on the base board, part of which had to be cut away to accommodate the fly-wheel. The tuning mechanism was mounted on two aluminium brackets created from a piece of right-angled aluminium profile 2 mm thick. A third bracket was used to mount the rotary encoder for channel selection.

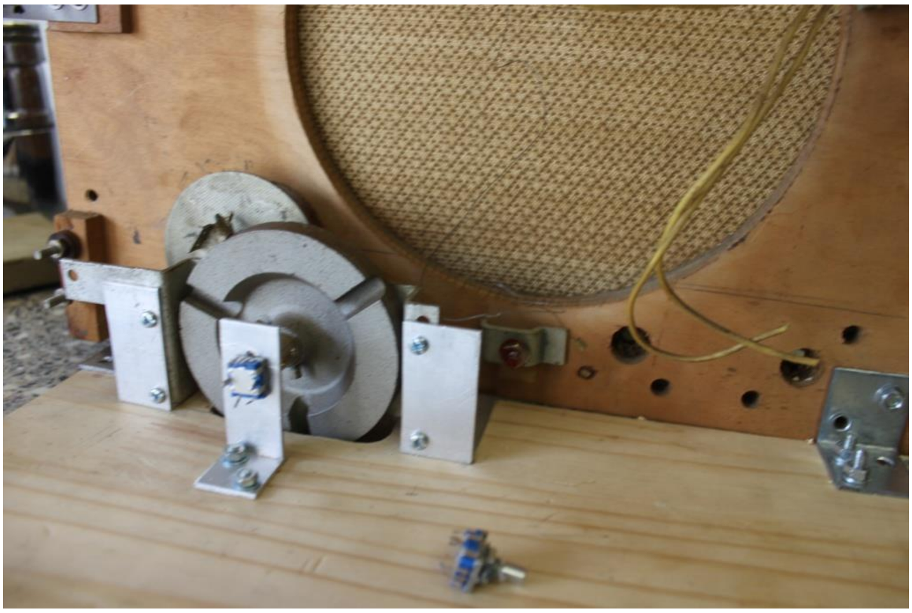


Figure 5 The radio tuning mechanism

Figure 6 Potentiometer shaft extender coupling

The shaft of the rotary encoder is connected using a potentiometer shaft extender coupling. This connects to the shaft of the fly-wheel in this case. Pot extender shafts can also be used with these.

Due to the mass of the fly-wheel it was found that the push switch of the rotary encoder would not operate correctly. The menu switch needed a separate push button which was mounted on the side of the radio case (There was already a hole drilled in this case).

## Fitting the new speakers

Two four-inch speakers were fitted to the original front panel of the radio on a piece of 5 mm ply-wood which was attached using the three old loud-speaker clamps.  Unfortunately, larger loud-speakers could not be used due to space restrictions and it was not possibly to create new speaker apertures without destroying the original front of the radio.

The tuning mechanism cable was re-attached to the tuning mechanism (Not an easy job) allowing the rotary encoder to be driven using the original tuning knob and front scale. This was one of the main objectives of this design.

On the right side a second aluminium was fitted bracket to hold the volume control rotary encoder. However, this was later moved to the next position along to keep the encoder wiring as short as possible.

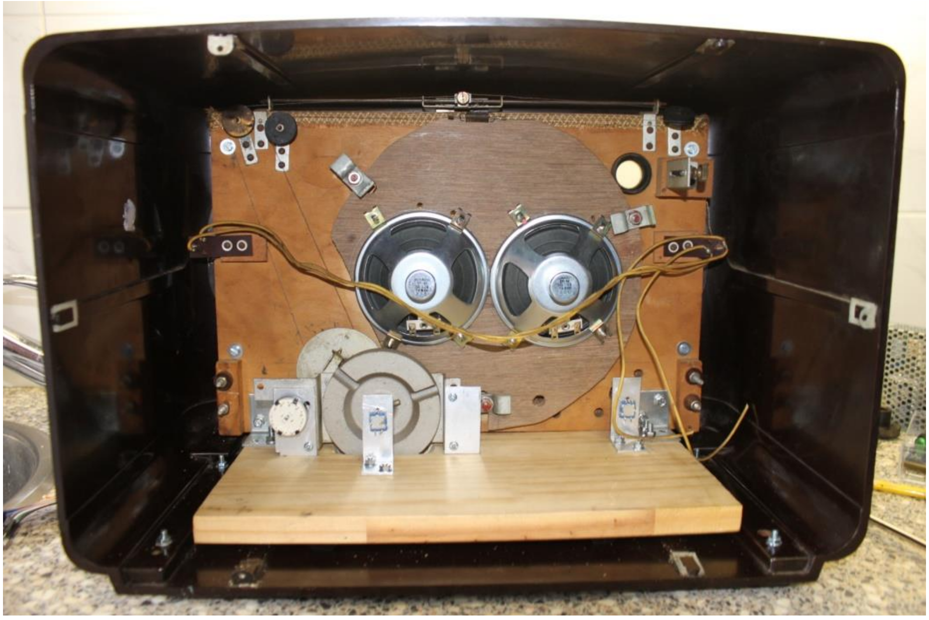


Figure 7 Fitting the loud-speakers

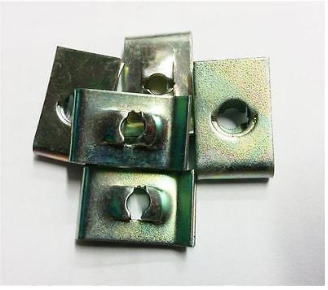


Figure 8 Speed clamps

The speakers were attached using speed-clamps which are fastened with self-tapping screws. Purpose made speaker clamps are a better choice if you can get them.

## Fitting the main components

The following figure shows the main components fitted to the base board. From left to right:

1. Raspberry Pi with a HifBerry DAC (Later replaced with a DAC plus)
2. A 26-way ribbon cable connecting to the switch interface board
3. A Velleman 35-watt stereo amplifier with a large black heatsink
4. A 12-volt transformer for the amplifier
5. The tone control interface board (in front of the 12-volt transformer)
6. A Velleman 5 volt switched power supply.
7. Top right the LED status and IR detector interface board

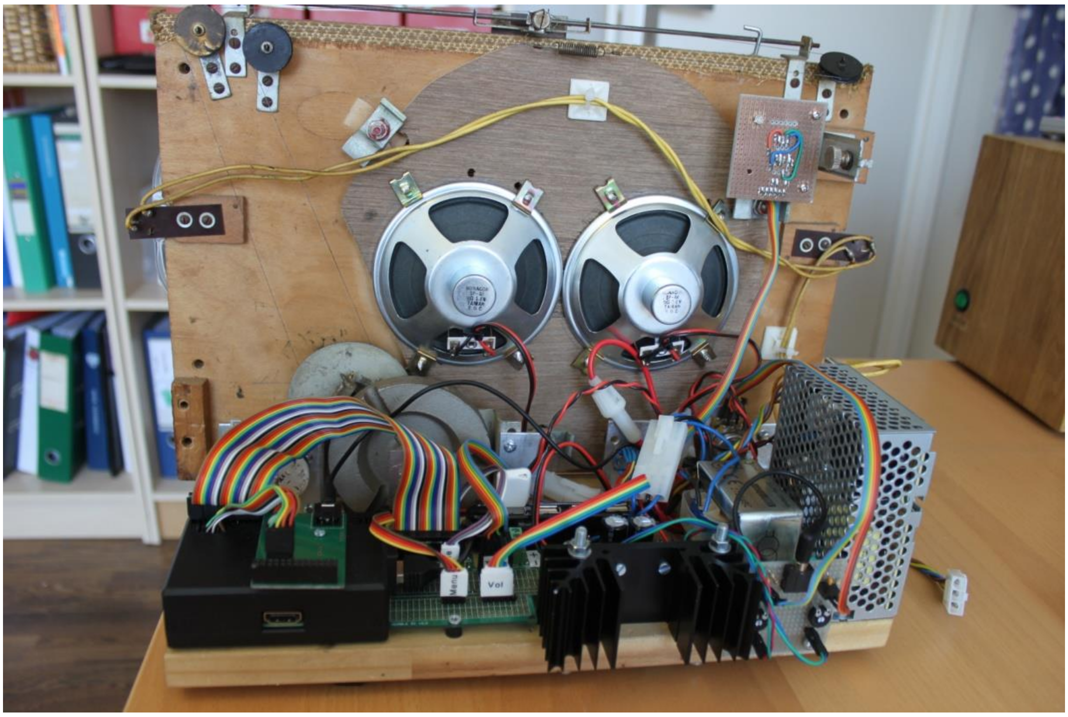


Figure 9 Fitting the main components

The individual components are described in the following pages. The components such as the amplifier, transformer and mains switch are connected using either two or three pin in-line connecters allowing individual components to be easily removed for maintenance. The connector sticking out of the right of the picture connects to the mains switch which is mounted in the radio case.

The HiFiBerry DAC output connects to the tone control board. The two ribbon connectors coming out of the tone control board connect to a dual 10K linear potentiometer mounted in the front panel.

This picture also clearly shows the tuning mechanism. The yellow wires connect the 6-volt panel lamps to the Velleman 5-volt power supply. See http://ww1.vellman.com/ (Yes the domain name is spelt differently from the company name)

## The front panel

The original front panel is completely preserved. The controls from left to right are: Tone control, Volume control (connected with a pot extender), the tuning mechanism and rotary menu switch.

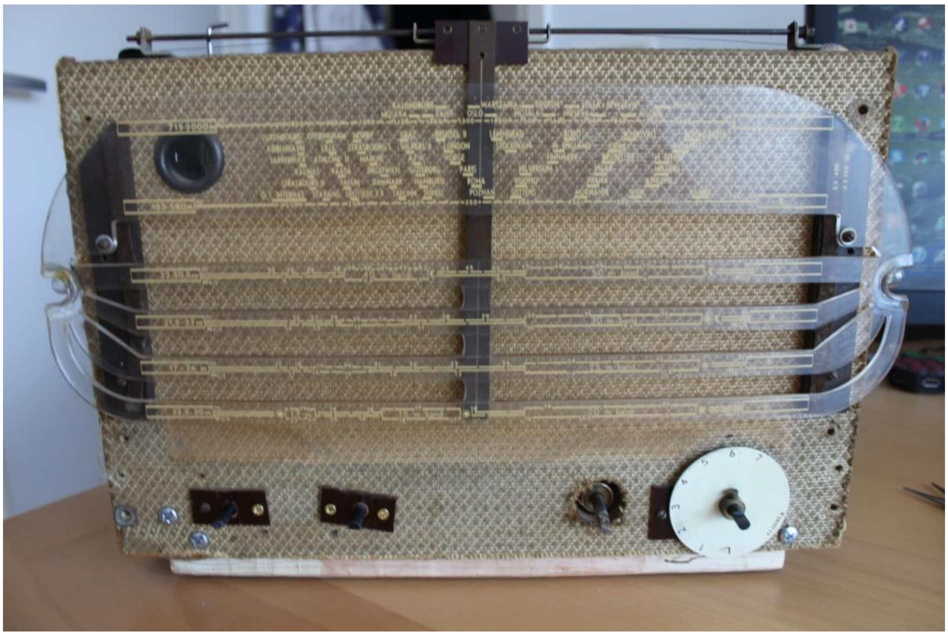


Figure 10 The radio front panel

The round aperture on the top left originally was for the so-called Magic Eye tuning indicator (See https://en.wikipedia.org/wiki/Magic_eye_tube). This window is now used for the RGB status LED and IR remote control detector. The tuner slide and pulley mechanism can be easily seen at the top of this illustration. In the final radio the inside of the aperture was fitted with a light diffuser to soften the light from the status RGB LED.

## The rotary menu switch

The rotary menu switch is totally optional. The radio will work fine without it using the normal Menu button.

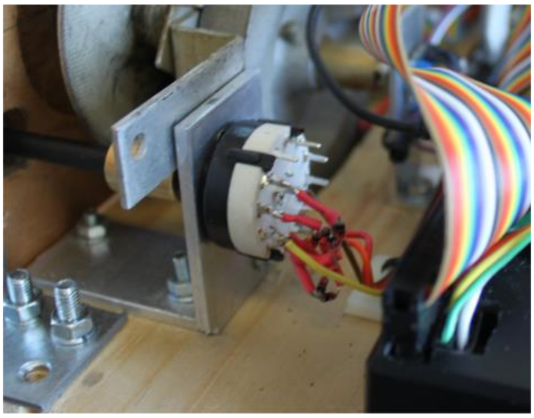


Figure 11 Rotary menu switch

The right-hand knob was originally used to select the short, medium and long wave bands. In this conversion a new eight rotary switch (not to be confused with a rotary encoder) is fitted to act as a menu switch. Using a series of diodes and three GPIO inputs and a ground wire the eight positions are used to generate a binary value of 0 to 7.
See the section called *Wiring the rotary menu switch* on page 18 for the actual switch wiring.

In the above diagram the rotary switch shaft is extended using an extender shaft (black) and collar (brass) to extend the switch out through the front panel.

## The status LED and IR detector board

The status LED and IR sensor are both optional. The radio will work fine without them.

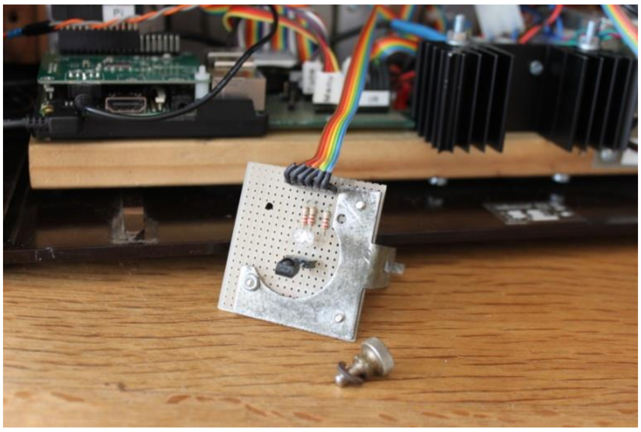


Figure 12 Status LED and IR sensor board

The status LED and IR detector board was constructed on a small piece of prototyping board and attached using the original bracket which held the Magic Eye tuning indicator.

The magic eye aperture is fitted with a diffuser made from some clear plastic covered with some semi opaque plastic sheet (Often used on windows to give privacy). This softens the light coming from the status LED.

If you wish to use an IR remote control with the radio then purchase an IR sensor TSOP38238 or similar. The output in this case will be connected to GPIO 11 (Pin 23).

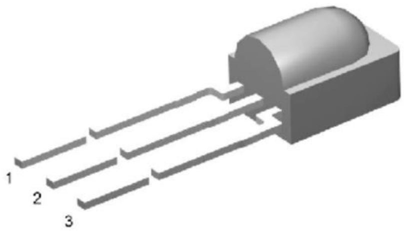



Figure 13 TSOP38238 IR sensor

The TSOP38xxx series works from 2.5 to 5.5 volts and is ideal for use the Raspberry PI.

| IR sensor | Description | RPi | Pin |
|---|---|---|---|
| Pin 1 | Signal Out | GPIO 9 | 23 |
| Pin 2 | Ground | GND | 6 |
| Pin 3 * | Vs 3.3 Volts | 3.3V | 1 |

***Caution:** Do <u>not</u> accidently connect to 5 volts

There are equivalent devices on the market such as the TSOP4838 which operate on 3.3 volts only.


Figure 14 RGB LED wiring

The RGB LED inputs are connected to the GPIO outputs via three 220 Ohm resistors. The common wire of the LED is connected to 0 volts (GND). 220 Ohms was chosen so that the status LED did not glow too brightly. A 1K preset potentiometer could be used in the ground wire to adjust brightness if so desired.

The GPIO outputs used for the status LED are:

Table 1 Status LED indications

| GPIO | Pin | LED | Function |
|---|---|---|---|
| **23** | 16 | Red | Error condition, shutdown in progress, IR activity (If configured) |
| **22** | 15 | Blue | Busy condition such as start-up, loading or changing radio stations or tracks. |
| **27** | 13 | Green | Normal operation such as playing stations or tracks. |

## The switch interface board

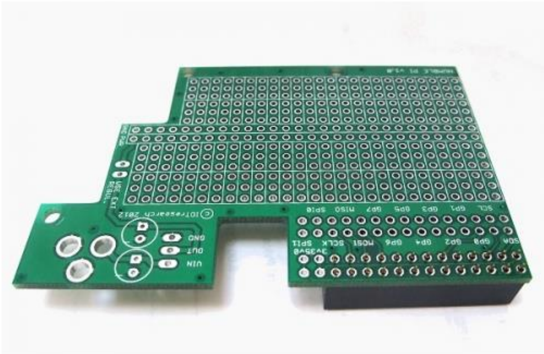
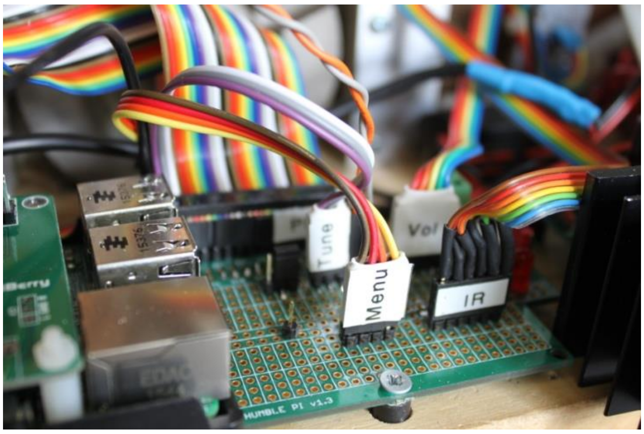


Figure 15 Ciseco Humble Pi

The switch interface board was built using a Ciseco Humble Pi extender board. This can be ordered with or without a five or three volt regulator. The board can be ordered from: http://www.hobbytronics.co.uk amongst others.

Figure 16 or 3 volt regulator




The connections are:

1. Tune - Channel selector (tuner) rotary encoder
2. Vol - Volume rotary encoder and mute switch
3. Menu rotary switch (not encoder)
4. LED status and IR detector board

The jumper to the left of the Tune connector allows the down switch to be selected between GPIO 18 or GPIO 10 to support the HiFiBerry DAC+.

The interface board connects to the 26 pin extender on top of the HiFiberry 40 pin male connector.

## The tone regulator and audio interface board

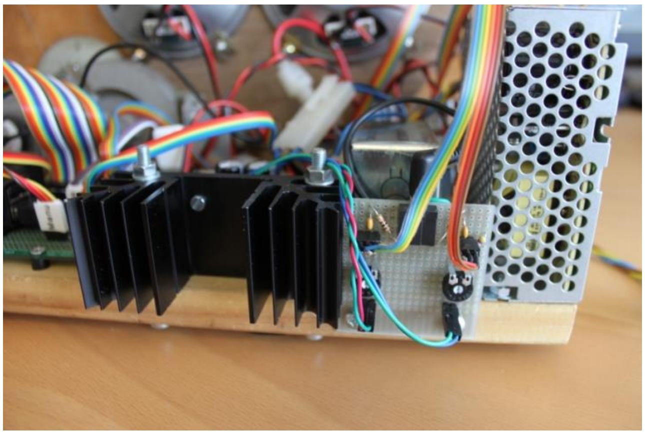


Figure 17 Tone regulator board

The tone regulator and audio interface board sits between the DAC+ output and the Velleman amplifier. The stereo input comes in via a miniature audio stereo jack plug and socket. The ribbon cable connects to 100K dual linear potentiometer to control the tone. The two 10K preset resistors set the final output into the amplifier (The red/green and blue/green wires.

See *Simple tone regulator* on page 20 for the circuit.

## The completed radio (Rear view)

Below is the completed radio viewed from the rear.

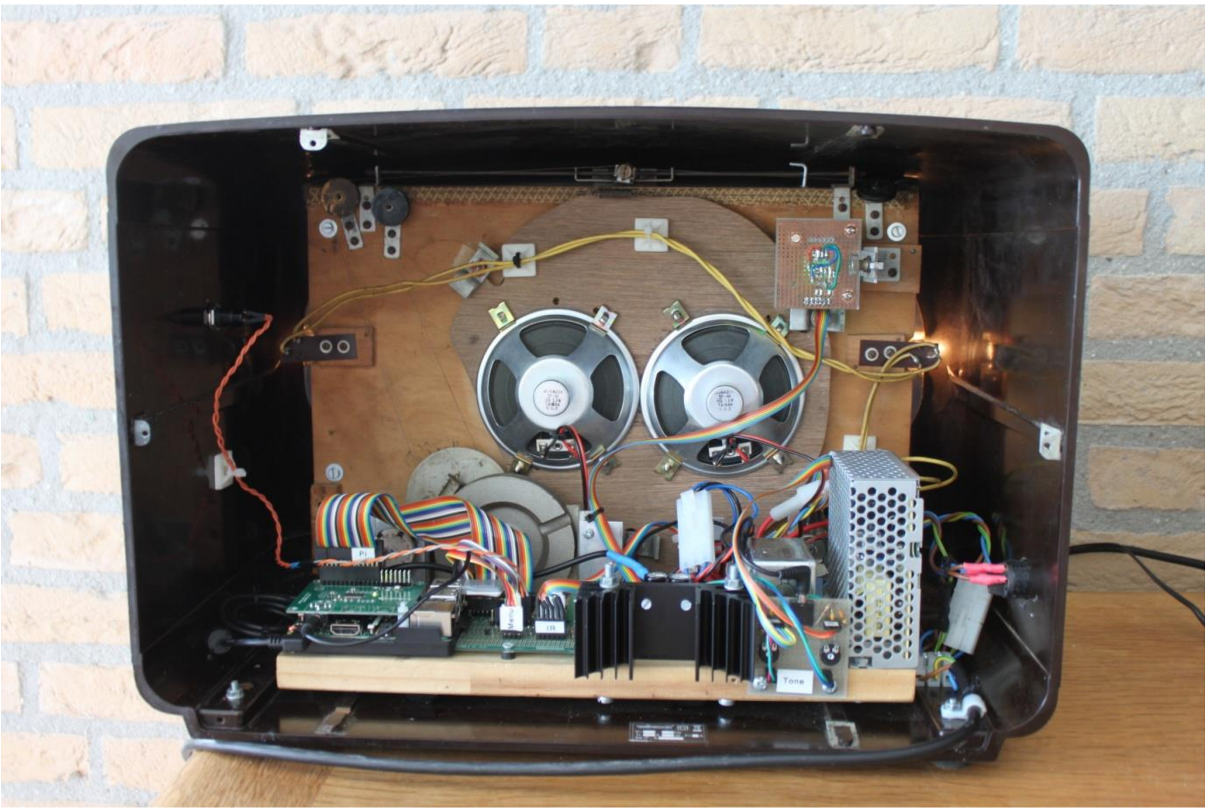


Figure 18 Completed radio (rear view)

From top left to right:

- Menu push button (orange twisted wire)
- Tuning scale slider pulley mechanism
- The status LED and IR sensor board (mounted in the original magic eye aperture)
- Tuning scale illumination lamps (connected by yellow wires to the +5 volt supply)
- Two four-inch 8 Ohm loud-speakers

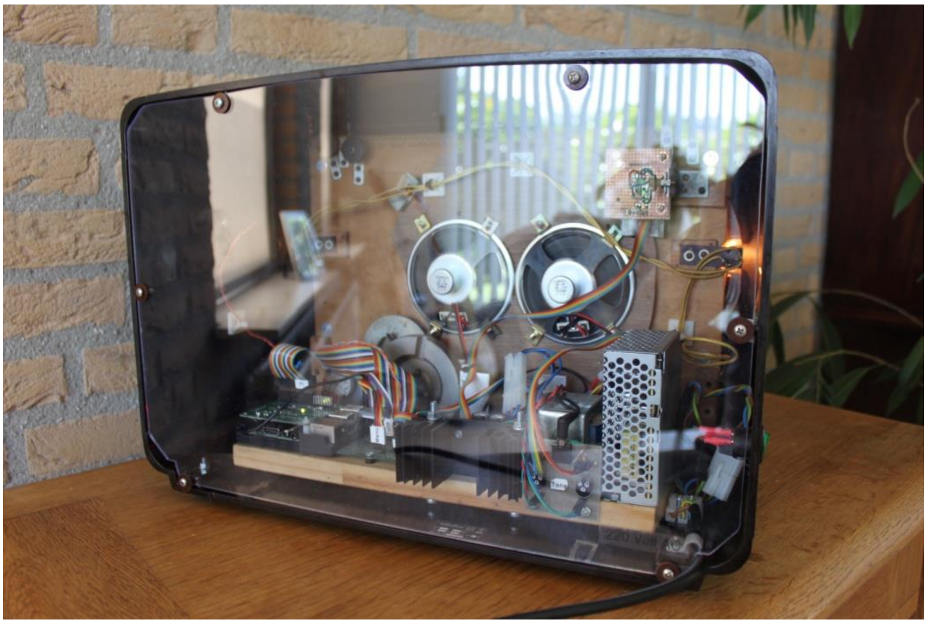
From bottom left to right:

- Raspberry Pi model 3 with HiFiBerry DAC+ mounted on top of the Pi
- 26-way rainbow coloured ribbon cable connecting to the switch interface board
- The switch interface board constructed from a Ciseco Raspberry PI breakout board
- A Velleman 35-watt stereo amplifier
- Audio interface board and tone regulator
- 12-volt AC transformer (12v-0v-12v) for the Velleman amplifier (Behind the tone regulator)
- Velleman 5-volt DC switched power supply
- Mains input cable with connector and clamp
- Double pole mains switch mounted in the radio case

Also, a 32G USB stick (grey/white) is connected to the Raspberry Pi via one of the USB ports. The Raspberry Pi 3 has inbuilt Wi-Fi so no separate Wi-Fi device is needed. The project originally started out using a Model 2B Pi plus DAC but ended up with a Model 3 Pi and DAC+.

## Creating a Perspex back cover

For some reason the backs of most vintage radios are manufactured from an unattractive pressed cardboard sheet. It is a nice idea to be able to see the inner workings of the radio so a 3 mm Perspex cover was created.  For tips on cutting and drilling Perspex search the internet (Strongly recommended).



## The completed radio front view

This final figure shows the final setting for the completed radio. The retro looking lamp and an old Bell telephone from Belgium complete a nostalgic setting for the radio.

Figure 19 The final radio

The status LED is glowing green in the round window at the top left indicating normal operation.

## Wiring

The following table shows the interface wiring for the Retro Radio using two rotary encoders and a rotary switch (Not encoder) for the menu.

**Table 2 Interface board wiring**

| Pin | Description | Function | Push Buttons | Encoder (Tuner) | Encoder (Volume) |
|---|---|---|---|---|---|
| **1** | 3V3 | +3V supply | COMMON | | |
| **2** | 5V | 5V for LCD | | | |
| **3** | GPIO2 | I2C Data*** | | | |
| **4** | Reserved | | | | |
| **5** | GPIO3 | I2C Clock*** | | | |
| **6** | GND | Zero volts | | Common | Common |
| **7** | GPIO 4 | Mute volume | | | Knob Switch |
| **8** | GPIO 14 | Volume down | LEFT | | Output A |
| **9** | Reserved** | GND | | | |
| **10** | GPIO 15 | Volume up | RIGHT | | Output B |
| **11** | GPIO 17 | Channel Up | UP | Output A | |
| **12** | GPIO 18**** | Channel Down/DAC+ | DOWN | Output B | |
| **13** | GPIO 27(21)* | Red status LED | | | |
| **14** | Reserved** | GND | | | |
| **15** | GPIO 22 | Blue status LED | | | |
| **16** | GPIO 23 | Red Status LED | | | |
| **17** | 3V3 | +3V supply | | | |
| **18** | GPIO 24 | | | | |
| **19** | GPIO 10**** | Channel Down | DOWN | Output A | |
| **20** | Reserved** | GND | | | |
| **21** | GPIO 9 | IR Sensor (1) | | | |
| **22** | GPIO 25 | Menu switch value 1 | MENU | | |
| **23** | GPIO 11 | IR LED out (1) | | | |
| **24** | GPIO 8 | Menu switch value 2 | | | |
| **25** | Reserved** | GND | | | |
| **26** | GPIO 7 | Menu Switch value 4 | | | |
| **33** | GPIO 13 | IR LED out (2) | | | |
| **35** | GPIO 19 | HiFiBerry DAC+ | | | |
| **36** | GPIO 16 | | | | |
| **37** | GPIO 26 | IR Sensor (2) | | | |
| **38** | GPIO 20 | HiFiBerry DAC+ | | | |
| **40** | GPIO 21 | HiFiBerry DAC+ | | | |

LCD Pin 3 (Contrast) may be connected to the centre tap of a 10K preset potentiometer (Not used in this project).
Pins 33 to 40 are for 40 pin versions only of the Raspberry PI.
* Pin 13 is GPIO27 on Rev 2 boards and GPIO21 on Rev 1 boards
** These pins were originally reserved and are connected to ground (GND 0V). These may now be used as extra GND pins.
*** These pins are used for the I2C LCD backpack if used instead of the directly wired LCD to GPIO pins.
**** Pin 12 is used by the HiFiberry DAC+, Use GPIO10 (Pin 19) if using the DAC plus.

## Table 3 Radio and IQAudio sound devices 40 pin wiring

| Pin | Description | Radio Function | Name | IQAudio Function | Push Button | Encoder (Tuner) | Encoder (Volume) |
|---|---|---|---|---|---|---|---|
| **1** | 3V3 | +3V supply | +3V | +3V | +3V | | |
| **2** | 5V | 5V for LCD | +5V | +5V | | | |
| **3** | GPIO2 | I2C Data | I2C Data | I2C Data | | | |
| **4** | 5V | | | +5V | | | |
| **5** | GPIO3 | I2C Clock | I2C Clock | I2C Clock | | | |
| **6** | GND | Zero volts | 0V | 0V | | Common | Common |
| **7** | GPIO 4 | Mute volume | | | | | Knob Switch |
| **8** | GPIO 14 | Channel up | UART TX | | UP | | Output A |
| **9** | GND | Zero Volts | | 0V | | | |
| **10** | GPIO 15 | Channel down | UART RX | | DOWN | | Output B |
| **11** | GPIO 17 | Menu switch | | | | Knob Switch | |
| **12** | GPIO 18 | | | I2S CLK | | | |
| **13** | GPIO 27 | | | | | | |
| **14** | GND | Zero Volts | | 0V | | | |
| **15** | GPIO 22 | | | Mute | | | |
| **16** | GPIO 23 | Volume down | | Rotary enc A | LEFT | Output A | |
| **17** | 3V3 | +3V supply | | 0V | | | |
| **18** | GPIO 24 | Volume up | | Rotary Enc B | RIGHT | Output B | |
| **19** | GPIO 10 | | SPI-MOSI | | | | |
| **20** | GND | Zero Volts | | | | | |
| **21** | GPIO9 | | SPI-MISO | | | | |
| **22** | GPIO 25 | IR Sensor | | IR sensor | | | |
| **23** | GPIO 11 | | SPI-SCLK | | | | |
| **24** | GPIO 8 | Menu Switch 1 | SPI-CE0 | | | | |
| **25** | GND | Zero Volts | | 0V | | | |
| **26** | GPIO 7 | Menu Switch 2 | SPI-CE1 | | | | |
| **27** | DNC | | | PiDac+ Eprom | | | |
| **28** | DNC | | | PiDac+ Eprom | | | |
| **29** | GPIO5 | Menu Switch 4 | | | | | |
| **30** | GND | Zero Volts | | | | | |
| **31** | GPIO6 | Red status LED | | | | | |
| **32** | GPIO12 | Green status LED | | | | | |
| **33** | GPIO 13 | Blue Status LED | | | | | |
| **34** | GND | Zero Volts | | | | | |
| **35** | GPIO 19 | IQAudio DAC+ | I2S | I2S | | | |
| **36** | GPIO 16 | IR LED out | | | | | |
| **37** | GPIO 26 | | | | | | |
| **38** | GPIO 20 | IQAudio DAC+ | I2S DIN | I2S DIN | | | |
| **39** | GND | Zero Volts | | | | | |
| **40** | GPIO 21 | IQAudio DAC+ | I2S DOUT | I2S DOUT | | | |

## Wiring the rotary menu switch

By wiring three GPIOs as inputs which are normally held high and by switching these to ground via a series of diodes a value of 0 through 7 can be generated (8 menu positions). See *Figure 11 Rotary menu switch* on page 11.

**Table 4 Rotary menu switch wiring**

| Switch pin | GPIO7 (Pin 26) | GPIO8 (Pin 24) | GPIO25 (Pin 22) | Menu Value | Diodes required | Function |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | Unused |
| 2 | 0 | 0 | 1 | 1 | 0 | Load Radio stations |
| 3 | 0 | 1 | 0 | 2 | 0 | Information (espeak required) |
| 4 | 0 | 1 | 1 | 3 | 2 | Search menu |
| 5 | 1 | 0 | 0 | 4 | 0 | Load media library |
| 6 | 1 | 0 | 1 | 5 | 2 | Recreate music library index |
| 7 | 1 | 1 | 0 | 6 | 2 | Unused |
| 8 | 1 | 1 | 1 | 7 | 3 | Unused |
| Common | GND | - | - | - | - | Common 0 volts (GND) |

**Figure 20 IN4148 diode**

The diodes used are any low voltage low current diodes such as the IN4148. So, to use all of the above combinations would require a single pole 8 way rotary switch and logic. The first switch position is off. Wire the centre pin of the switch to 0v (GND).

Wire as follows:

- Leave switch position 1 with nothing connected to it, this has a binary value 0.
- Wire GPIO25 (pin 22) to switch position 2, this has a binary value 1.
- Wire GPIO8 (pin 24) to switch position 3, this has a binary value 2.
- Wire GPIO7 (pin 26) to switch position 5, this has a binary value 4.
- Wire two diodes from switch pin 4 to switch pins 2 and 3 (value 3)
- Wire two diodes from switch pin 6 to switch pins 2 and 5 (value 5)
- Wire two diodes from switch pin 7 to switch pins 3 and 5 (Value 6 - Optional - not used)
- Wire three diodes from switch pin 8 to switch pins 1,3 and 5 (Value 7 – Optional - not used)
- Wire the centre tap of the switch to 0 volts (GND)

## Parts List

The following table shows the parts list for the Raspberry PI Internet Radio. This list is for the version using the HD44780 LCD directly connected to the GPIO pins.

**Table 5 Parts list**

| Qty | Part | Supplier |
|---|---|---|
| 1 | Raspberry Pi Computer | Farnell Element 14 |
| 1 | Clear Raspberry Case | RS Components |
| 1 | 8 GByte SD Card | Any PC or Photographic supplier |
| 1 | An old radio case (Bakelite or Wooden) | Look around |
| 1 | Raspbian Jessie OS | Raspberry Pi foundation downloads |
| 2 | Four-inch loudspeakers | From set of old PC speakers |
| 1 | Stereo Amplifier (10 to 35 watt) | Any electronics shop |
| 1 | 12-volt transformer for amplifier | Any electronics shop |
| 1 | LCD HD44780 2 x 16 Display * | Farnell Element 14 |
| 1 | Any prototyping board | Pi Hut and others |
| 1 | Eight way rotary switch (not an encoder!) | Any electronics shop |
| 1 | Square push button * | Any electronics shop |
| 2 | Rotary encoders | Sparkfun.com |
| 1 | 26-way ribbon cable | Tandy or Farnell Element 14 |
| 1 | Four port USB hub (Revision 1 & 2 boards only) | Any PC supplier |
| 1 | External power supply for USB hub (1200 mA) | Any PC supplier |
| 1 | 26-way PCB mount male connector | Any electronics shop |
| 1 | 26-way GPIO extender (model B+ boards only) | ModMyPi and others |
| 1 | Mains cable (black) and plug (brown or black) | Hardware shop |
| 1 | Double pole mains switch with neon | Farnell Element 14 |
| 5 | Male 2 pin PCB mount connectors | Any electronics shop |
| 2 | Female 4 pin PCB connectors | Any electronics shop |
| 1 | Female 2 pin PCB connectors | Any electronics shop |
| 1 | 16 pin male in-line PCB mount connector | Any electronics shop |
| 1 | Stereo jack plug socket | Any electronics shop |
| 1 | TSOP38238 IR Sensor | Adafruit industries and others |
| 1 | Red or Green LED and 220 Ohm resistor | Any electronics shop |
| 8 | IN4148 diodes (If using the rotary menu switch) | Any electronics shop |
|  | Shrink wrap and thin wire for PCB wiring | Any electronics shop |

# Miscellaneous

## Simple tone regulator

The following diagram and modified text came from Jack Orman at:
http://www.muzique.com/lab/swtc.htm

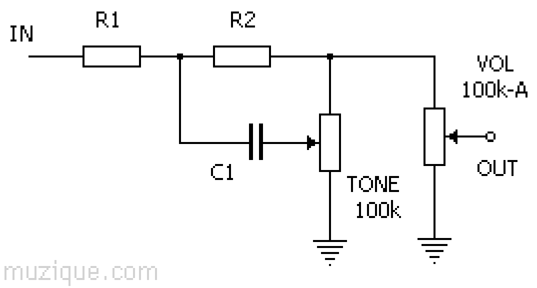



Figure 21 Simple tone control circuit

This tone control circuit that has a response that can be altered from high cut to high boost as the knob is turned. The output resistance is constant so the volume does not vary as the tone control is adjusted.

Suggested values for beginning experimentation with are R1=10k, R2=47k, C1=0.022uF and 100k for the tone and volume pots.

Figure 22 Dual 100K Linear potentiometer

Remember that the above circuit needs to be duplicated for right and left audio channels. This also means purchasing a dual linear 100K potentiometer for the tone control.

Note that the above circuit has a lot of attenuation of the audio output so using the onboard audio output of the Raspberry Pi might result in a disappointing level of volume. It is recommended to use a sound output DAC or USB sound dongle.

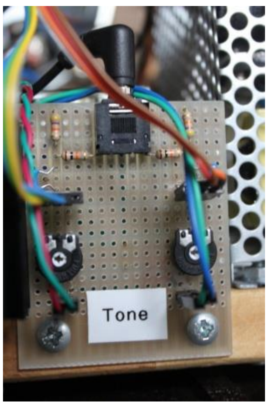



Figure 23 Tone control board

The illustration on the left shows a simple passive tone regulator board using the above circuit.

The audio output from the Raspberry Pi or DAC is fed into the board via a standard Audio socket.

Below the input are the connections to the tone regulator potentiometer mounted on the front panel of the radio.

Below the potentiometer connections are the two 100K pre-sets for adjusting the output level to the Audio Amplifier. Below these the Left and Right audio outputs connect to the Audio Amplifier.

## Converting stereo output to mono

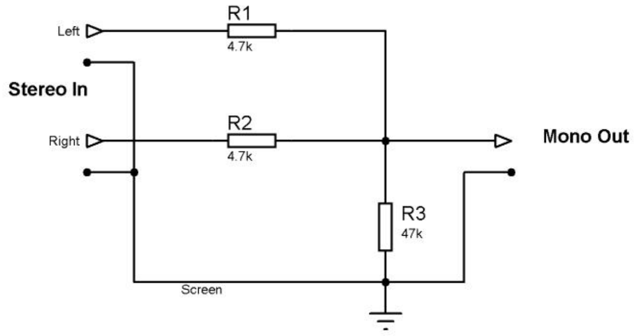



Figure 24 Stereo to mono conversion circuit

The circuit on the left is a simple passive stereo to mono circuit which can then be fed into a mono amplifier to drive the original speaker fitted in the radio. This solution has been tested by the author and seems to work fine. However other alternative circuits may be better. Search the internet with the title of this section for alternative circuits.

# Installing the radio Software

## Selecting the radio daemon

Follow instructions in the main Raspberry Pi Radio Constructors Guide. However, when asked to choose the user interface, select option 2 Radio with Rotary Encoders:

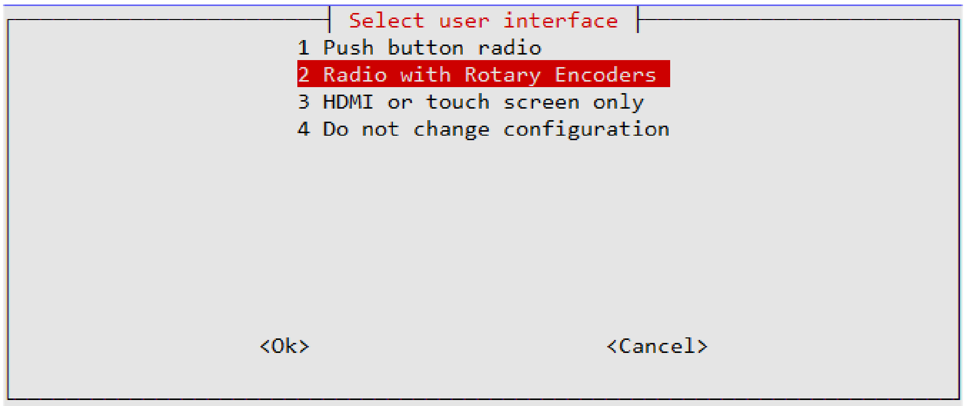



Figure 25 Selecting interface type

Next select option 1 for the 40-pin wiring scheme:

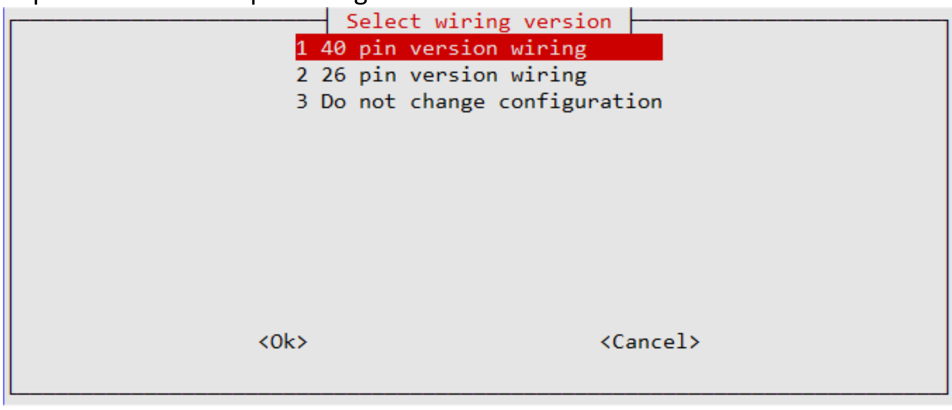
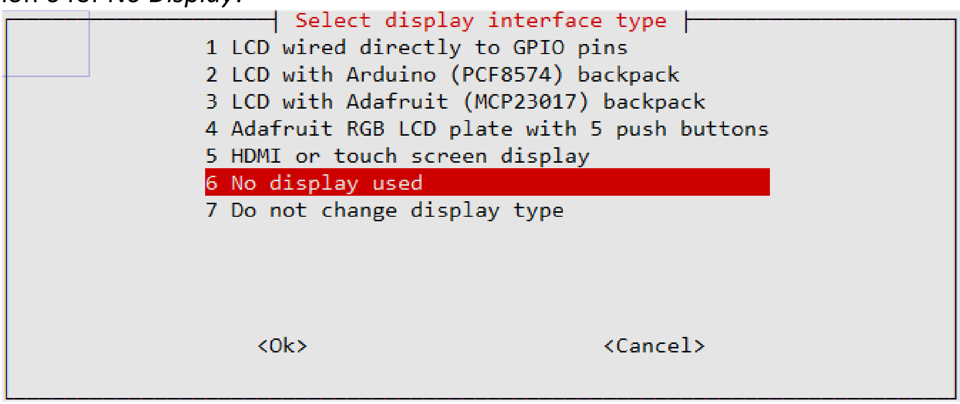



Figure 26 Wiring scheme selection

Select option 6 for *No Display*:




Follow the rest of the instructions for the installing the rest of the software.

## Configuring the IR sensor

The installation procedure for the IR remote is fully described in the constructor's guide. In this project the IR sensor output is connected to GPIO9 (physical pin 21). Edit the /boot/config.txt file and add the following line to the end of the file.

```
dtoverlay=lirc-rpi,gpio_in_pin=9,gpio_in_pull=high
```

## Configuring the remote-control activity LED

If using a remote control with an activity LED and using an RGB status LED then configure the activity LED to use the Red LED of the RGB LEDs.

Edit the **remote_led** parameter in **/etc/radiod.conf** to use GPIO 23 (Pin 16). This means that the RED LED of the RGB LED will flash when the IR receiver is active.

```
remote_led=23
```

The above is optional and if using a separate IR activity LED then configure this as shown in the main constructor's guide.

## Configuring the status LEDs

The following is optionally. Set to 0 if you do not want to use them

```
rgb_green=27
rgb_blue=22
rgb_red=23
```

## Installing espeak

Since this radio has no display it is not possible to use the search and information functions without using **espeak**. See http://espeak.sourceforge.net/. See the constructor's manual for instructions on how to install **espeak**.

## Configuring the rotary menu switch

The menu switch is entirely optional. Zero values disable usage. The following values were used in this project.

```
menu_switch_value_1=24
menu_switch_value_2=8
menu_switch_value_4=7
```

# Operation

## Rotary encoder operation

The volume knob when pushed in is the **Mute** sound function. Likewise the tuner knob when pushed in is the **Menu** switch. If **espeak** is enable then the Mute switch speaks the information about the radio channel or media track. In this case the radio is muted by pressing the mute switch in for two seconds.

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 6 Rotary Encoder Knob Operation**

| | Volume knob | | Tuner knob | |
|---|---|---|---|---|
| **Mode** | **Clockwise** | **Anti-clockwise** | **Clockwise** | **Anti-clockwise** |
| **Mode = MAIN(TIME)**<br>Line 1: Time<br>Line 2: Station or Track | Volume Up | Volume Down | Station/Track up | Station/Track down |
| **Mode = SEARCH**<br>If source = RADIO<br>Line 1: Search:<br>Line2: Radio Station | Volume Up | Volume Down | Scroll up radio station | Scroll down radio station |
| **Mode = SEARCH**<br>If source = MUSIC LIBRARY<br>Line 1: Search<br>Line2: MusicTrack/Artist | Scroll up through artists | Scroll down through artists | Scroll up through track | Scroll down through track |
| **Mode = SOURCE**<br>Line 1: Input Source:<br>Line2: Internet Radio or Music Library | Volume Up<br>Mute | Volume Down<br>Mute | Toggle mode between Radio and Music Library | Toggle mode between Radio and Music Library |
| **Mode = OPTIONS**<br>Line 1: Menu Selection<br>Line 2: <option><br><br>Options are Random, Consume, Repeat, Reload Music, Timer, | 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) |

## Mute function

Pressing both volume buttons together or in the case of a rotary encoder with a push button (Volume) will mute the radio. If voice is enabled then then operation is slightly different (See section on espeak). Press either the volume up or down switch to un-mute the radio. If you change channel or use the menu switch the radio will also be un-muted. If the alarm is set then the radio will go into sleep mode.

# Alternative solutions

## Using the PA input

As previously mentioned, it is possible to utilise the audio stage of the radio if it is still working. Many 1950's and 60's radios have an input directly to the audio stage of the radio. This is usually on the back of the radio and will be either a PA input (Personal Amplifier) or a gramophone input. Either can be used.

The output of the Raspberry Pi Internet Radio can be directly fed into the PA (Aux) or Gramophone input on the back of the radio. Since these are usually mono inputs it will be necessary to convert the stereo output of the Raspberry Pi to mono using the circuit as shown in *Converting stereo output to mono* on page 21.

In the illustration below is a 1950's Telefunken Operette 7 valve radio manufactured in West Germany. On top is a four-line LCD version of the Pi Radio with push buttons (In an old cream-cracker box). The output of the Internet Radio is fed into the gramophone input of the radio. The gramaphone input is selected by pressing in the second button from the left marked PLATTE (Record Player)


Figure 27 Vintage radio using the gramophone input

There are several advantages to this approach:

- There is no need to modify the original radio, just build a small Pi internet radio.
- The original band selection and tuning (if still working) can still be used.
- The original often warm sound of the radio can still be experienced.
- The original volume and tone controls can still be used.

The input into the audio section will normally require banana plugs. However if the radio is somewhat newer other types of connector may be used such as DIN or Phono plugs. This will depend upon the radio being used.

Figure 28 Banana plugs

The following illustration shows the output of the Raspberry Pi Internet Radio connected via a Stereo Jack cable to the Mono to stereo conversion unit.

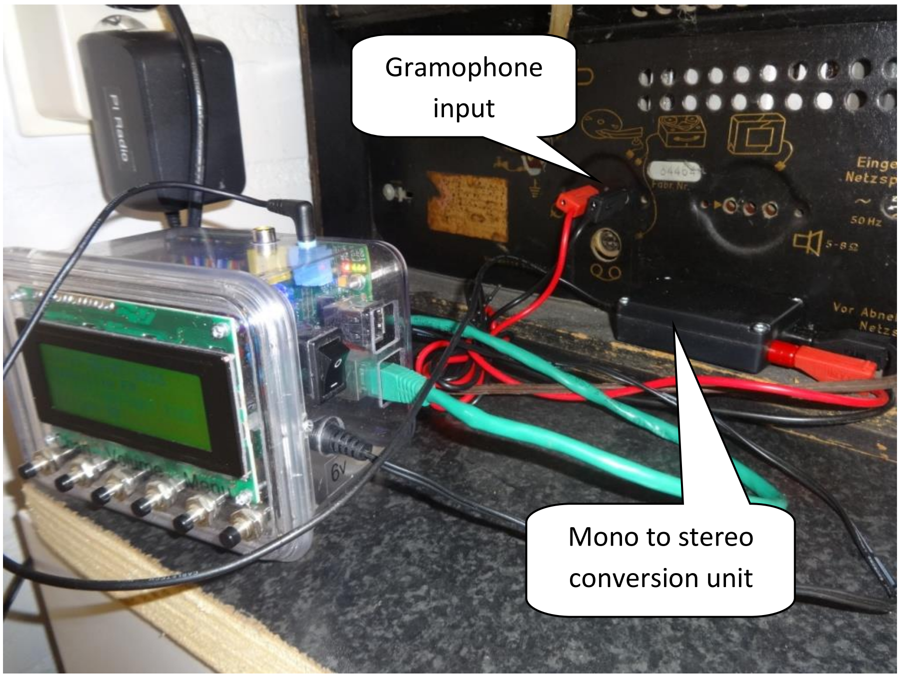



Figure 29 Stereo to mono converter

The output of the converter is then fed via banana plug leads to the Gramophone input of the radio.

The above is just another possible solution that can be used to utilise a vintage radio which has at least the audio section still working. If the radio which you are trying to use does not have a gramophone or auxiliary input the it may be possible to connect to the audio stage but this will require some electronics knowledge. These days it is usually possible to find the circuit diagram for most vintage radios on the Internet.

## Using an LCD screen

It is of course possible to convert a vintage radio and fit it with an LCD screen however this means cutting a hole in the casing to allow the LCD screen to be mounted which does mean altering the original look of the radio. In the solution shown below the LCD is mounted in the least conspicuous place possible. Use the LCD version from the main Raspberry Pi Internet Radio project.

See the main Raspberry Pi PDF documentation at:
http://www.bobrathbone.com/raspberrypi/Raspberry%20PI%20Radio.pdf

Figure 30 Old Zenith radio using rotary encoders

On the left is an example of the PI radio from James Rydell built into an old Zenith valve radio case. The pictures below show the inside and top view respectively. The two original controls have been replaced by two rotary encoders. The old valve radio inside has been completely removed and replaced with the Raspberry PI and radio components. The LCD display has been built into the top so as not to spoil the original face of the radio.

Figure 31 Zenith radio rear view

Figure 32 Zenith radio top view

Figure 33 Vintage radio with LCD and push buttons

Yet another design from Aubrey Kloppers using an old valve radio but this time using push buttons. Type of original radio unknown.

See his blog at:
http://aubreykloppers.wordpress.com/2013/09/13/insane-internet-radio-project/

## Using a touch screen

Since version 6.2 of the radio software, it is possible to use a touch screen with the radio. There are two touch screen versions of the radio namely **gradio.py** and **vgradio.py**. The first one is the full feature radio, the second one (**vgradio.py**) is designed to look like a vintage radio tuning scale. The **vgradio.py** program only fits on a Raspberry Pi 7-inch touch-screen screen.

The illustration below shows a French Radio Schneider Frères Rondo from the 1950's which has been converted to an internet radio by Franz-Josef Haffner, from Germany.

Figure 34 Vintage radio using a touch screen

What makes this project also very interesting is that he has removed all of the RF section of the valve radio leaving only the audio amplifier and power supply.

This is an excellent example of combining old and new technology to extend the life of these increasingly rare radios.

Franz-Josef's blog for this project will be found at:
https://radiobasteleien.blogspot.com/2019/01/schneider-freres-rondo-internetradio.html

Although the blog is in the German language it is well illustrated and how the radio was converted is very clear from these.

Figure 35 Valve chassis showing audio amplifier

The illustration on the left shows the original radio chassis. With the tuning mechanism and RF and IF valve stages completely removed.

This was done to make way for the Raspberry Pi components.

Figure 36 Touch-screen fitted to front panel

Here is the touch-screen fitted to the chassis. In front of the of the touch-screen is the original glass plate which originally had a tuning scale printed on it.

Franz-Josef carefully scratched this tuning scale off with his finger nail.

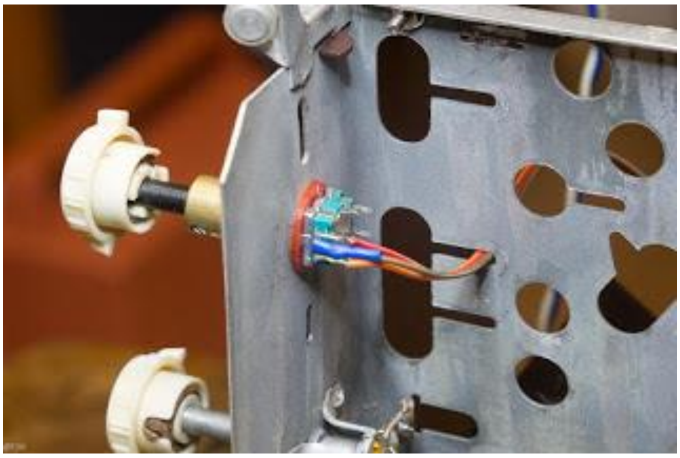


Figure 37 Fitting rotary encoders to the chassis

Because Franz-Josef used the original front glass meant that the touch-screen cannot be used, so rotary encoders are used instead to operate the radio.

The original knobs are connected to the rotary encoder using standard shaft extenders,

This illustration shows the rear of the completed radio.

The Raspberry Pi is shown on the left in a black plastic case.

The valve audio amplifier and power supply are on the right of chassis.

Figure 38 Completed radio - rear view

## Using a low power AM transmitter

Some vintage radios don't easily present the possibility of building an internet radio inside the case, nor do they have a PA input as shown in Figure 27 on page 25. An example of such a radio is the British made **Bush DAC 90A** medium/longwave receiver. A possible solution is to use a very low power AM band transmitter.

Such a low power transmitter is the Lusya DIY 3-channel AM transmitter is shown below. It isn't expensive and only has a transmission range of a few feet.


Figure 39: Lusya DIY 3-channel AM transmitter

The above illustration shows a 3.6 volt 2.4Ah Lithium battery. These are special batteries approximately 6.5 cm in length (Not AA size batteries). A rechargeable battery can be also be used but these are expensive and also require a charger. The device is switched off by pulling the jack plug out of the audio socket.

Since the crystals operate between 3.3V and 5V the 5-Volt supply on the Raspberry Pi can be used to power the device. Connect a female jumper wire from the RPi +5 Volt supply and solder the other end to the positive battery terminal on the underside of the board. The earth return is provided via the 3.5 mm jack lead.

Below are the details for frequency and wavelengths for the AM transmitter.

**Figure 40: Three Channel AM transmitter frequencies/wavelengths**

| Jumpers | Frequency | Wavelength | Radio dial |
|---|---|---|---|
| **1** | 1.000 MHz | 299.79246 Metres | 300 Meters |
| **2** | 3.000 MHz | 99.93082 Metres | 100 Metres |
| **3** | 5.000 MHz | 59.95849 Metres | 60 Metres |

Use jumper position 1 (300 Metres on the tuning dial) as 2 and 3 may be outside the capability of the radio, which was the case with the radio shown in Figure 41 below.

Below is a proof-of-concept set-up of the AM transmitter. The transmitter power is fed from Raspberry Pi Internet radio which is controlled through the web interface. The arial ideally has to be within 20 centimetres of the radio.

**Figure 41 Bush DAC 90A with AM transmitter**

**Warning:** In general, a license is required by all world states to operate a radio transmitter and have restrictions on which frequencies and power output can be used. This is generally required to prevent interference with any adjacent operator frequencies or equipment. Many world states have exemptions for very low power equipment such as loop repeaters for the hard of hearing or RFID systems used by mobile telephones. CB bands are also usually exempt. The output from the AM transmitter shown in this project is in the order of one milli-Watt with a range of less than one meter. Although this virtually eliminates any possibility of any interference it may not stop you falling foul of local laws and restrictions. You should therefore check with the local body which regulate the use of radio frequencies in your country before using such devices.

Please also see disclaimer on page 35.

## Using a DAC sound card to drive the AM transmitter

Using a DAC with the AM transmitter is probably a complete waste of time and money. The DAC card outputs high-fidelity audio signal which is then degraded by the radio transmitter and the vintage AM radio. The audio output from the on-board 3.5 audio jack is more than sufficient. However, it isn't a problem to use one if you feel the need to do so.

## Using a battery supply for the transmitter

When operating the vintage radio with a normal Raspberry Pi power supply you may notice a slight mains hum when used with the AM transmitter. This is not normally a problem as most music will drown this out. If, however you are listening to some soft classical music this may be irritating. This can be eliminated by using a 5V battery pack.


Figure 42 Raspberry Pi 5 Volt battery pack

Various manufacturer's supply a range of battery 5 Volt packs specifically for the Raspberry Pi. These are charged from a normal 5V charger.

Try to use a battery pack that directly connects to the GPIO header rather than connecting via the micro-USB or USB-C port.

The reason is that if connected by the USB port the internal regulator it will drop the voltage to 4.8 Volts which may give problems with the Wi-Fi connection.

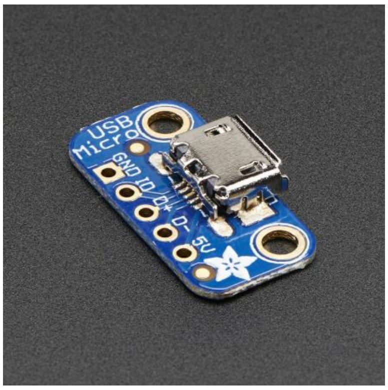



Figure 43 Micro-USB breakout board

If the battery pack chosen doesn't connect directly to the GPIO header then purchase a micro-USB breakout board. This allows the 5 volt and GND connections to be connected directly to pins 2 and 6 respectively on the GPIO header.

# Safety Precautions

## General

Working on vintage valve equipment is completely different from working on modern electronic equipment. The voltages required for valve operation are much higher, up to 240 volts AC and typically around the 200 volts DC. Also, many vintage radios have a so-called **Live Chassis** also known as **Hot Chassis** in the USA.

It is essential that you observe sensible precautions when working on such equipment and treat them with respect. Switch off before connecting such things as voltage meters or other test equipment.

**Never** hold or touch the radio with two hands when it is out of its case and is plugged in!!!

## Radios with a live chassis

From the mid 1930's to the early 1960's, there were millions of domestic radios sold throughout the world. A lot of these radios were produced incorporating a so-called voltage dropper which connected directly to mains live(hot) wire and the other side to the metal chassis of the radio. Mains droppers were also used, at the time, to allow the radio to work on DC mains. Later radios starting using a mains transformer which isolates the chassis and radio circuits from the AC mains.

Many countries such as Germany, Holland and others have un-polarized plugs which can be inserted into the mains socket either way around. Depending upon which way it is inserted the metal chassis can become live (hot) presenting a deadly hazard. See https://en.wikipedia.org/wiki/Schuko for more information on these types of plug.

Make sure that you either use a polarised plug that can only be inserted one way or mark the plug so that when inserted only the neutral wire connects to the chassis. This can be confirmed using a mains neon test screwdriver and then marking the plug so that it is plugged in the correct way around. The other way is to connect the radio through a one-to-one ratio isolating transformer.

The following articles are useful reading.
http://www.geojohn.org/Radios/MyRadios/Safety.html
https://www.bvws.org.uk/405alive/tech/safety.html

## Using an Oscilloscope on a Live Chassis

A modern Oscilloscope is connected to Live, Neutral and Earth. If the radio chassis has become Live (Hot) then this is a significant hazard as well as the possibility of damaging the Oscilloscope as it shorts the Live chassis to ground. For years TV and Radio engineers always disconnected the earth wire on the Oscilloscope temporarily whilst working on a live chassis. This was an accepted practice for the early decades of radio and TV repair. It would be frowned upon these days and professional repairers will most likely now use an isolating transformer. However, if you don't have an isolating transformer then disconnecting the earth on the Oscilloscope is less of a hazard than leaving it connected whilst working on a live chassis.

**Always consider safety first and make sure that no-one including yourself can receive an electric shock from your project including when the case is closed. Also see disclaimer on page 35. Take extra care when working on vintage radios.**

# Source files

The main source files are explained in the main constructors guide however the following files are special to the Vintage Radio project:

## The Status LED class

The *status_led_class.py* is called by the *radiod.py* software for use with a vintage radio. A Red Blue Green LED is driven to indicate status of the radio as there is no LCD screen. In this project it is placed behind a diffuser behind the magic eye tuning indicator window. It uses GPIOs originally used by the LCD driver so is not compatible with any directly wired LCD versions of the radio.

## The Menu Switch class

The *menu_switch_class.py* code supports an 8-position rotary switch (Not an encoder) as an alternative method of operating a simple menu system. It is used with the *radiod.py* software for use with a vintage radio. It also uses GPIOs originally used by the LCD driver so is not compatible with any directly wired LCD versions of the radio.

# Licences

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

The GNU General Public License (GNU GPL or GPL) is the most widely used free software license, which guarantees end users (individuals, organizations, companies) the freedoms to use, study, share (copy), and modify the software. Software that ensures that these rights are retained is called free software. The license was originally written by Richard Stallman of the Free Software Foundation (FSF) for the GNU project.

The GPL grants the recipients of a computer program the rights of the Free Software Definition and uses *copyleft* to ensure the freedoms are preserved whenever the work is distributed, even when the work is changed or added to. The GPL is a *copyleft* license, which means that derived works can only be distributed under the same license terms. This is in distinction to permissive free software licenses, of which the BSD licenses are the standard examples. GPL was the first *copyleft* license for general use. This means that you may modify and distribute the software and documentation subject to the conditions of the licences.

See http://www.gnu.org/licenses for further 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 |

## 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 OR INJURY (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.

PLEASE ALSO READ THE SECTION SAFETY PRECAUTIONS ON PAGE 33.

## Technical support

Technical support is on a voluntary basis by e-mail only at bob@bobrathbone.com. Before asking for support, please first consult the troubleshooting in the constructor's manual. 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 description of what you have you built.
- Which program and version are you running?
- A clear description of the fault.
- What you have already done to locate the problem?
- Did you run the test programs and what was the result?
- Switch on DEBUG logging as described in the constructors manual, run the program and include the log file.
- Did you vary from the procedure in the manual or add any other software?
- Include the **/var/log/radio.log** file (if relevant) with the email.
- Do not answer my questions with a question. Pleas supply the information requested.

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/

For issues relating to Icecast2 streaming see:
http://www.icecast.org

For those of you who want to amend the code to suit your own requirements please note: I am very happy to help people with their projects but my time is limited so I ask that you respect that. Please also appreciate that I cannot engage in long email conversations with every constructor to debug their code or teach them Python.

# Acknowledgements

Jack Orman at http://www.muzique.com/lab/swtc.htm for his beautifully simple tone control.

James Rydell for his ideas for the Internet radio built into an old Zenith valve radio

Aubry Kloppers for his push button version of a vintage radio.

Philips BV, the Netherlands, for the original BX490A radio

Telefunken AG for the Operette 7 vintage radio.

Franz-Josef Haffner, from Germany, for his conversion of a Schneider Frères Rondo vintage radio.

Other acknowledgements will be found in the main *Raspberry Pi Radio Constructors Manual*.

# Glossary

| | |
|---|---|
| AA | In this context a standard battery size |
| AC | Alternating Current |
| AM | Amplitude Modulated radio signal |
| DAC | Digital to Analogue Converter |
| DC | Direct Current |
| HiFi | High Fidelity audio |
| IF | Intermediate Frequency (Oscillator) used in Super-Heterodyne radio receivers |
| LCD | Liquid Crystal Display |
| LED | Light Emitting Diode |
| MPD | Music Player Daemon |
| PA | Personal Amplifier (or Auxiliary input) |
| PDF | Portable Document Format - Adobe Systems Incorporated |
| RF | Radio Frequency such as an RF amplifier stage in a radio |
| RGB | Red, Blue, Green. In this case an RGB LED |
| Wi-Fi | Wireless Internet synonymous with Wireless Local Area Network |

# Index