

Raspberry PI Internet Radio

Vintage Radio Supplement



Bob Rathbone Computer Consultancy

www.bobrathbone.com

14th of August 2016

Contents

Introduction	5
Repair or convert?	5
Construction.....	6
Stripping out the old internals	6
Using the original speaker(s)	6
The basic radio case ready for construction	7
Fitting a new base	7
Mounting the tuning mechanism	8
Fitting the new speakers.....	9
Fitting the main components.....	10
The front panel	11
The rotary menu switch	11
The status LED and IR detector board	12
The switch interface board	13
The tone regulator and audio interface board	13
The completed radio (Rear view).....	14
Creating a Perspex back cover	15
The completed radio front view	15
Wiring.....	16
Wiring the rotary menu switch	17
Parts List.....	18
Miscellaneous	19
Simple tone regulator	19
Converting stereo output to mono.....	20
Installing the radio Software.....	21
Selecting the radio daemon.....	21
Configuring the IR sensor.....	21
Configuring the remote control activity LED	21
Installing espeak.....	21
Operation	22
Rotary encoder operation.....	22
Mute function	22
Alternative solutions	23

Using the PA input	23
Using an LCD screen.....	25
Source files	26
The Radio Daemon.....	26
The Status LED class	26
The Menu Switch class.....	26
Licences	27
Disclaimer.....	27
Technical support.....	28
Acknowledgements.....	29
Glossary.....	29
Index.....	30

Figures

Figure 1 Philips BX490A Bakelite Radio 1949	5
Figure 2 Stripping out old the workings of the radio.....	6
Figure 3 The tuning mechanism.....	7
Figure 4 Fitting a new wooden base	7
Figure 5 The radio tuning mechanism	8
Figure 6 Potentiometer shaft extender coupling.....	8
Figure 7 Fitting the loud-speakers	9
Figure 8 Speed clamps	9
Figure 9 Fitting the main components.....	10
Figure 10 The radio front panel	11
Figure 11 Rotary menu switch	11
Figure 12 Status LED and IR sensor board	12
Figure 13 TSOP38238 IR sensor	12
Figure 14 RGB LED wiring.....	12
Figure 15 Ciseco Humble Pi.....	13
Figure 16 or 3 volt regulator	13
Figure 17 Tone regulator board	13
Figure 18 Completed radio (rear view)	14
Figure 19 The final radio	15
Figure 20 IN4148 diode.....	17
Figure 22 Simple tone control circuit.....	19
Figure 23 Dual 100K Linear potentiometer.....	19
Figure 84 Tone control board	19
Figure 21 Stereo to mono conversion circuit.....	20
Figure 24 Selecting the radio daemon	21

Figure 25 Vintage radio using the gramophone input	23
Figure 26 Banana plugs	24
Figure 27 Stereo to mono converter	24
Figure 28 Old Zenith radio using rotary encoders	25
Figure 29 Zenith radio rear view	25
Figure 30 Zenith radio top view	25
Figure 31 Vintage radio with LCD and push buttons	25

Tables

Table 1 Status LED indications	12
Table 2 Interface board wiring	16
Table 3 Rotary menu switch wiring	17
Table 4 Parts list	18
Table 5 Rotary Encoder Knob Operation	22

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_radio.htm

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/Raspberry%20PI%20Radio.pdf>

This document is not a step by step instruction but is a collection of ideas that were used in this project and may be useful for similar projects. The radio used for this project was a Philips BX490A manufactured in the Netherlands in 1949. An alternative (simpler) approach is shown at the end of this manual.



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

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 preset 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 20.

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.

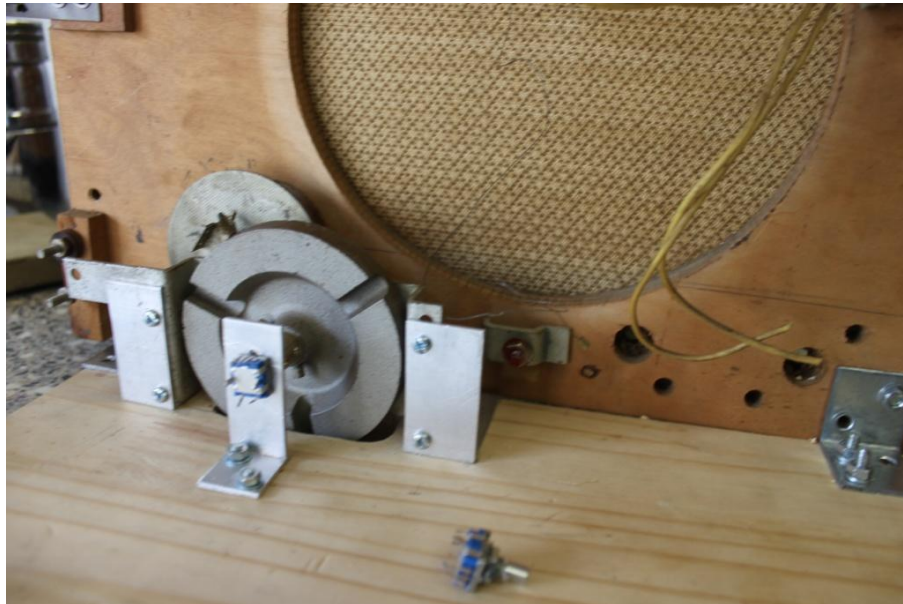


Figure 5 The radio tuning mechanism



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 possible 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

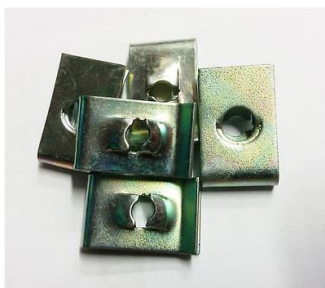


Figure 8 Speed clamps

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

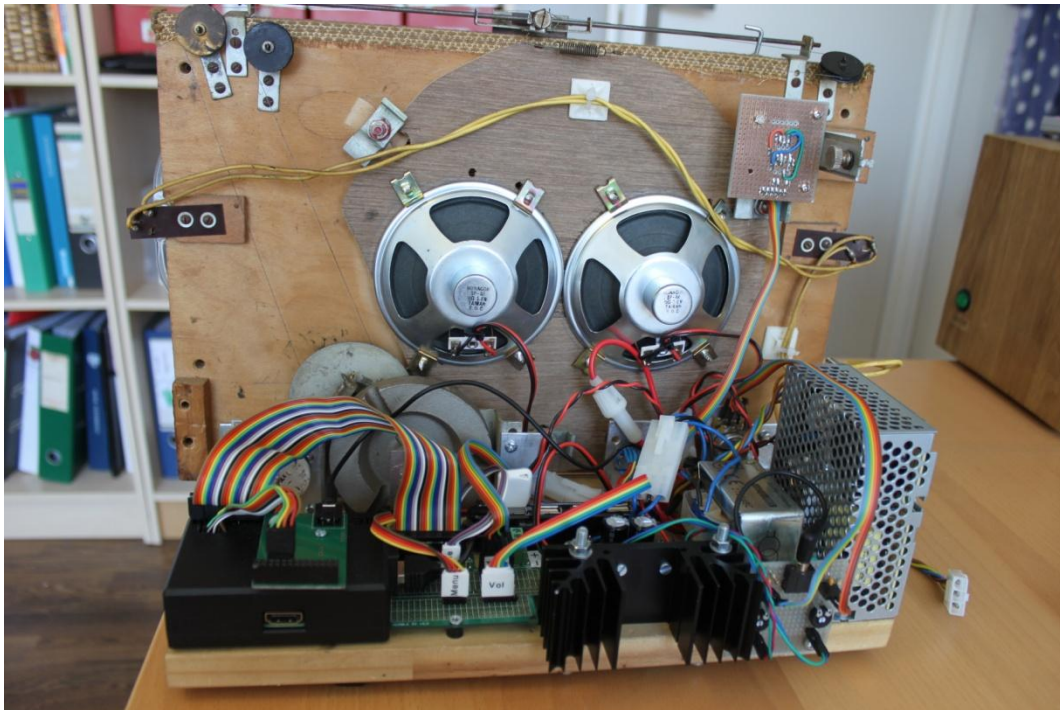


Figure 9 Fitting the main components

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 connectors 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.

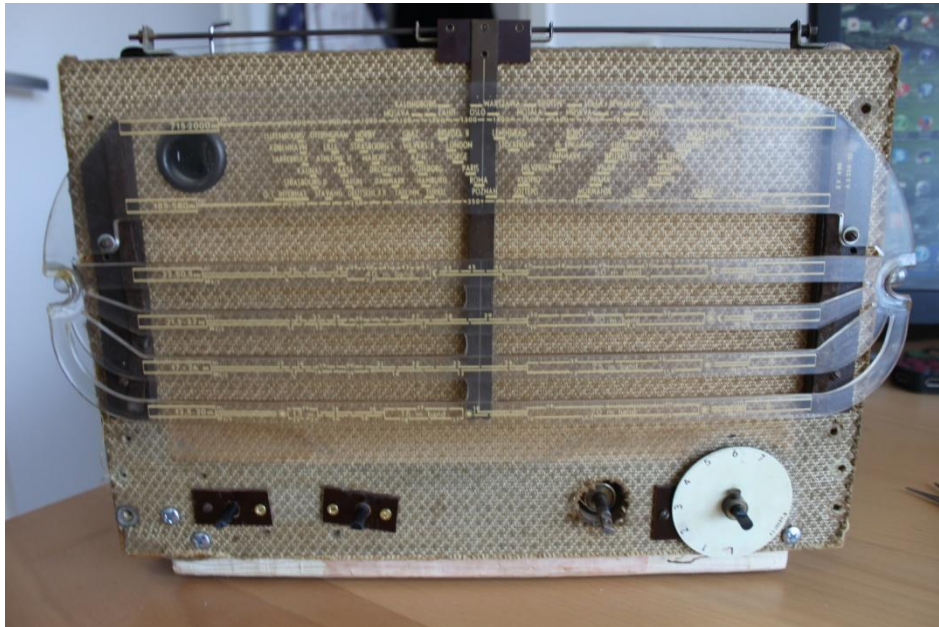


Figure 10 The radio front panel

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.

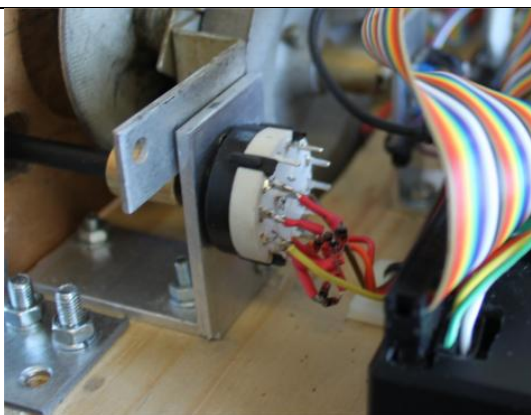


Figure 11 Rotary menu switch

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. New software has been written to handle this from version 5.3 of the radio onwards. See the section called *Wiring the rotary menu switch* on page 17 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.

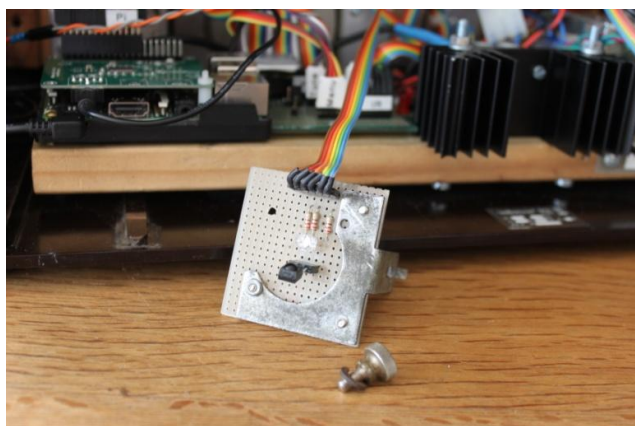


Figure 12 Status LED and IR sensor board

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).

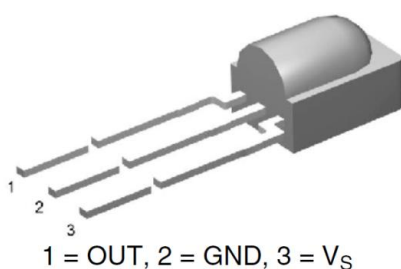


Figure 13 TSOP38238 IR sensor

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 not accidentally 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

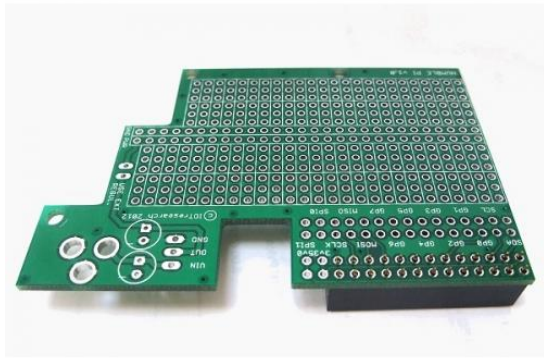


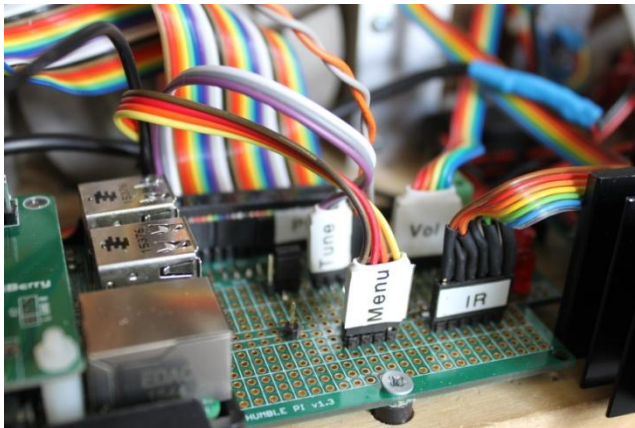
Figure 15 Ciseco Humble Pi

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

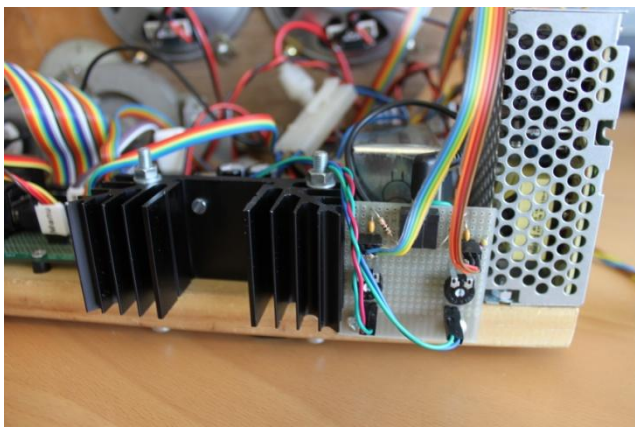


Figure 17 Tone regulator board

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 19 for the circuit.

The completed radio (Rear view)

Below is the completed radio viewed from the rear.

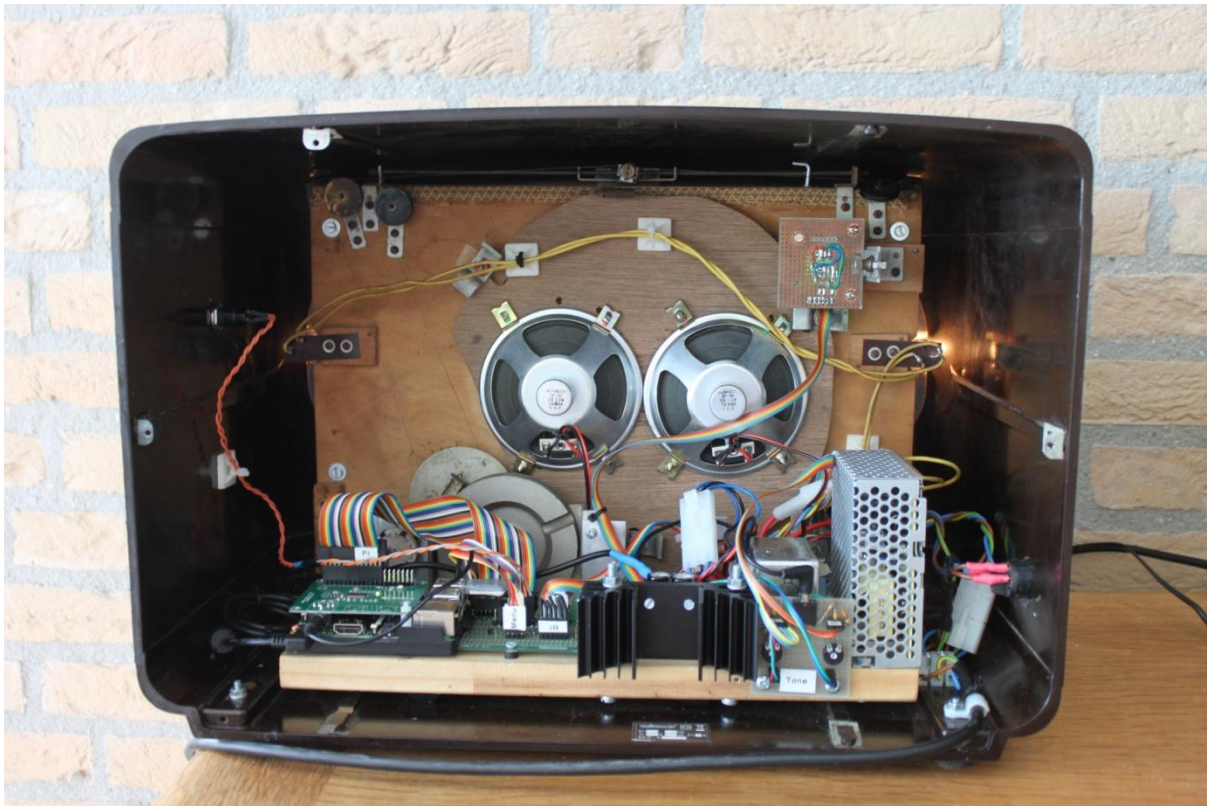


Figure 18 Completed radio (rear view)

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

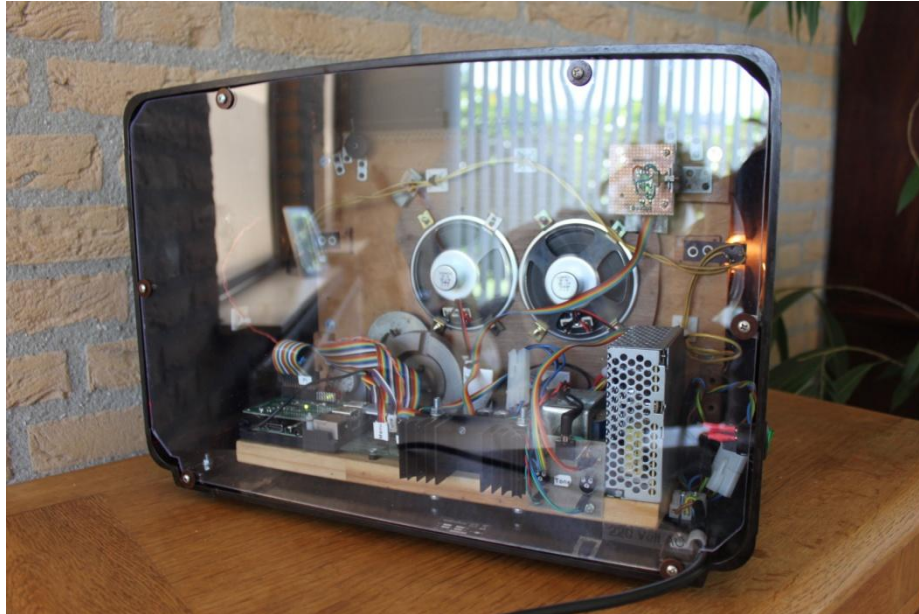
From bottom left to right:

- 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 WiFi so no separate WiFi 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	LCD Data 7				
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	Voice switch(future)				
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.

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 3 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 4 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	Ciseco Humble Pi extender board	Ciseco PLC
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>

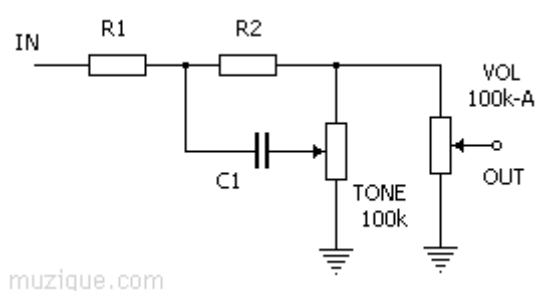


Figure 21 Simple tone control circuit

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 $R1=10k$, $R2=47k$, $C1=0.022\mu F$ 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.

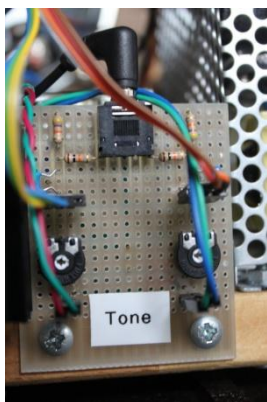


Figure 23 Tone control board

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 presets 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

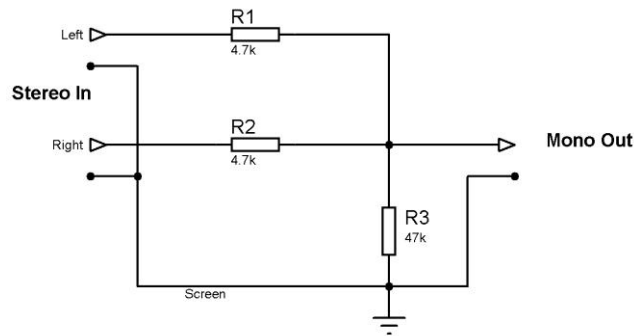


Figure 24 Stereo to mono conversion circuit

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 radio daemon select option 9 for the Retro radio software.

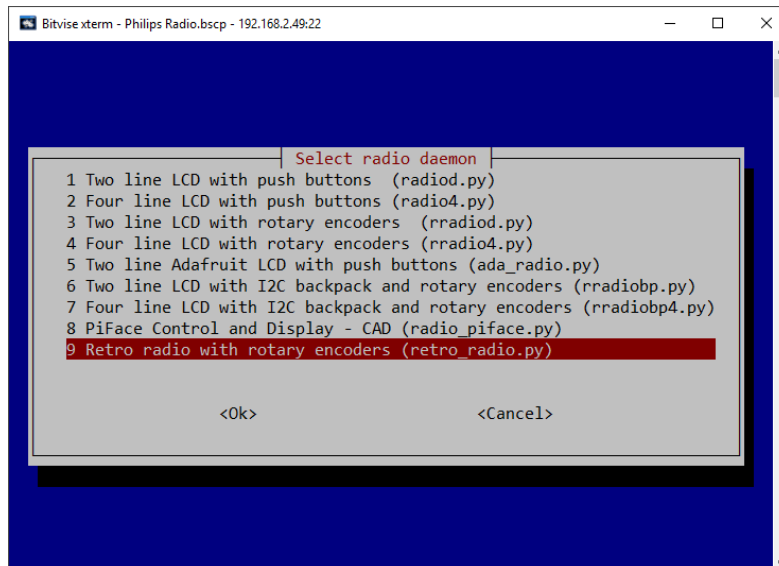


Figure 25 Selecting the radio daemon

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 constructors 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 constructors guide.

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 constructors manual for instructions on how to install **espeak**.

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 5 Rotary Encoder Knob Operation

Volume knob			Tuner knob	
Mode	Clockwise	Anti-clockwise	Clockwise	Anti-clockwise
Mode = MAIN(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: Internet Radio or Music Library	Volume Up Mute	Volume Down Mute	Toggle mode between Radio and Music Library	Toggle mode between Radio and Music Library
Mode = OPTIONS Line 1: Menu Selection Line 2: <option> 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 20.

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 gramophone input is selected by pressing in the second button from the left marked PLATTE (Record Player)



Figure 26 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 27 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.

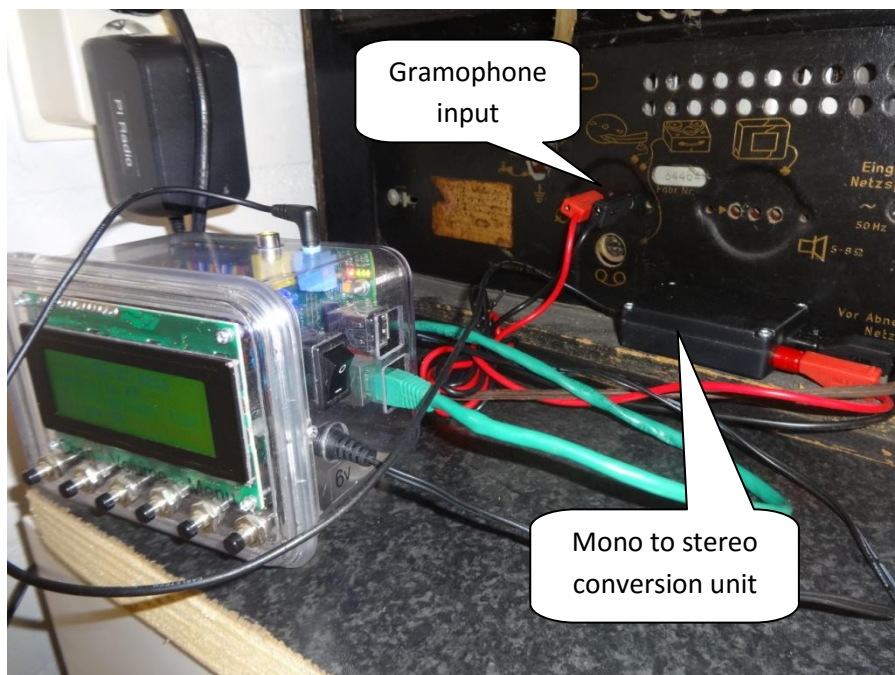


Figure 28 Stereo to mono converter

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 then 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. It uses the **rradio4.py** program 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 29 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 30 Zenith radio rear view



Figure 31 Zenith radio top view



Figure 32 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/>

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 Radio Daemon

The *retro_radio.py* source is the version used with rotary encoder switches. It has no support for any of the displays used in other versions of the radio. It also has a limited menu. Facilities such as RSS feeds, IP display and setting of alarm functions are not supported in this variant of the radio software as there is no display.

The Status LED class

The *status_led_class.py* is called by the *retro_radio.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 *retro_radio.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 BE LIABLE 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.

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. Please 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

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

Telefunken AG for the Operette 7 vintage radio.

Glossary

AC	Alternating Current
DAC	Digital to Analogue Converter
DC	Direct Current
HiFi	High Fidelity audio
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MPD	Music Player Daemon
PA	Personal Amplifier (or Auxillary input)
PDF	Portable Document Format - Adobe Systems Incorporated
RGB	Red, Blue, Green. In this case an RGB LED
WiFi	Wireless Internet synonymous with Wireless Local Area Network

Index

26 way ribbon cable, 18
DAC, 16
GPIO, 12, 16, 18
GPIO pins, 16, 18
HD44780, 18
HiFiBerry, 16
I2C, 16
IN4148, 17, 18
IR, 12, 16, 18
IR sensor, 12
IR Sensor, 16, 18
LCD, 16, 18, 25
MPD, 29

OS, 18
PA, 23, 29
PC, 18
potentiometer, 16
Raspberry PI, 1, 5, 12, 16, 18, 25, 28
remote control, 12
rotary encoder, 16, 22, 26
rotary encoders, 25
Rotary encoders, 18
TSOP38238, 12, 18
USB, 18
vintage radio, 26
wiring, 16, 17, 18