Edison Radio

A tribute to a great inventor

Internet Radio
MP3 Player
Wifi
Spectrum Analyzer
Vu Meter
Elegant Design

Building Manual

THE ELECTRONIC ENGINEER

M.Donners 20 januari 2021

Table of Contents

M Donners Document version 1.0

1. Disclaimer and safety

I, Mark Donners, The Electronics Engineer, may or may not endorse various Do-It-Yourself (DIY) projects and all DIY projects are purely “at your own risk”. As with any DIY project, unfamiliarity with the tools and process can be dangerous. Posts should be construed as theoretical advice only.

If you are at all uncomfortable or inexperienced working on these projects (especially but not limited to electronics and mechanical), please reconsider doing the job yourself. It is very possible (but not likely) on any DIY project to damage belongings or void your property insurance, create a hazardous condition, or harm or even kill yourself or others.

I will not be held responsible for any injury due to the misuse or misunderstanding of any DIY project.

By using the information provided by me, (Website,Youtube, facebook and other social media), you agree to indemnify me, affiliates, subsidiaries and their related companies for any and all claims, damages, losses and causes of action arising out of your breach or alleged breach of this agreement(disclaimer).

The materials on this site are distributed “as is” and appear on the site without express or implied warranties of any kind, except those required by the relevant legislation. In particular, I make no warranty as to the accuracy, quality, completeness or applicability of the information provided.

The information provided is for entertainment and promotional purposes only. You may not rely on any information and opinions expressed in it for any other purpose.

Disclaimer short version:

This is a DIY project, use any provided information and/or materials at your own risk! I am not responsible for what you do with it!

2. About this project

This project is bases on the internet radio by Ed Smallenburg. He did an awesome job in creating a working internet-based radio. He shared his coding and hardware on the world wide web so that people like you and me can learn from it. He shared his build on GitHub using the copyleft license. (It basically comes down to: use it, modify it as needed, don't copyright it and share….)

You can find Ed's original build and data here:

Edzelf/ESP32-Radio: Internet radio based on ESP32, VS1053 and a TFT screen. (github.com)

So, I started with Ed's Arduino sketch and I added a few extra options:

- Search engine to find Internet Radio station by country.
- Find and implement the station's Logo and display it on the browser interface.
- Spectrum analyser on the TFT Screen
- LED VU meter with Pixel leds WS2812
- Sliders for changing Audio Settings.

The modified software and all project data will be available here:

https://github.com/donnersm/EdisonRadio

TIP: download all with one click by pressing the green button and choosing "download ZIP"

As you might have noticed, this project is named the Edison Radio. Thomas Edison was a great inventor who, amongst other things, improved the microphone for telephones by developing a carbon microphone. In ways, more than one, Edison contributed to worldwide communication. However, his greatest invention was not this carbon microphone or even his well-known phonograph but, it was the invention of the light bulb by far! Without Edison, we might still be using candles to bring light in darkness. That is why this radio has a light bulb as centrepiece. .. O yeah, this 100% optional attribute also looks kind of cool 😊

3. Tools needed.

You will be working with low voltage (5V max) but you should know your way around basic electronics. Using the wrong voltage or polarity might kill your project in an instance!

Being successful in building this device requires some adequate skills and tools. You should be able to solder small components on a PCB which will require a soldering station with a small tip as you will be handling components like 0805 sized and a LQFP-48 package. If this is beyond your possibilities, then you should ask someone to do it for you or order an already assembled PCB.

(Availability will depend on demand).

Furthermore, you be programming the microcontroller by using the Arduino IDE environment. Although the steps to do so will be described in this document, it is advisable for you to get to know this Arduino software.

If you are considering building the Edison housing, you will need to produce the panels for it. It can be done by hand, but it would be best to use a Laser cutter. Remember, there are some companies that offer a laser cutting service. Maybe there is a company near you who can do it for you?

And yes, you will need some basic tools like a screwdriver 😉

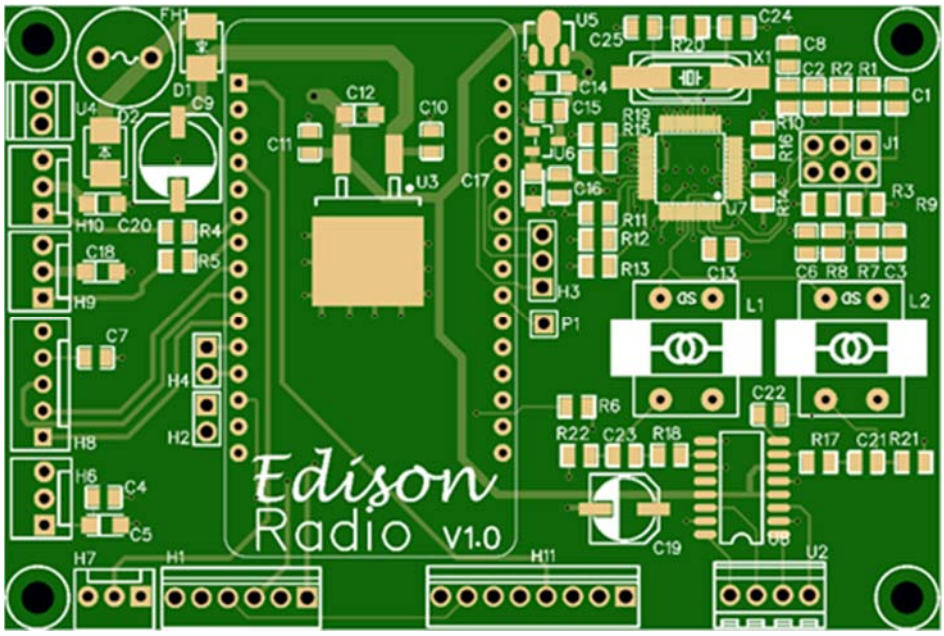
4. Hardware

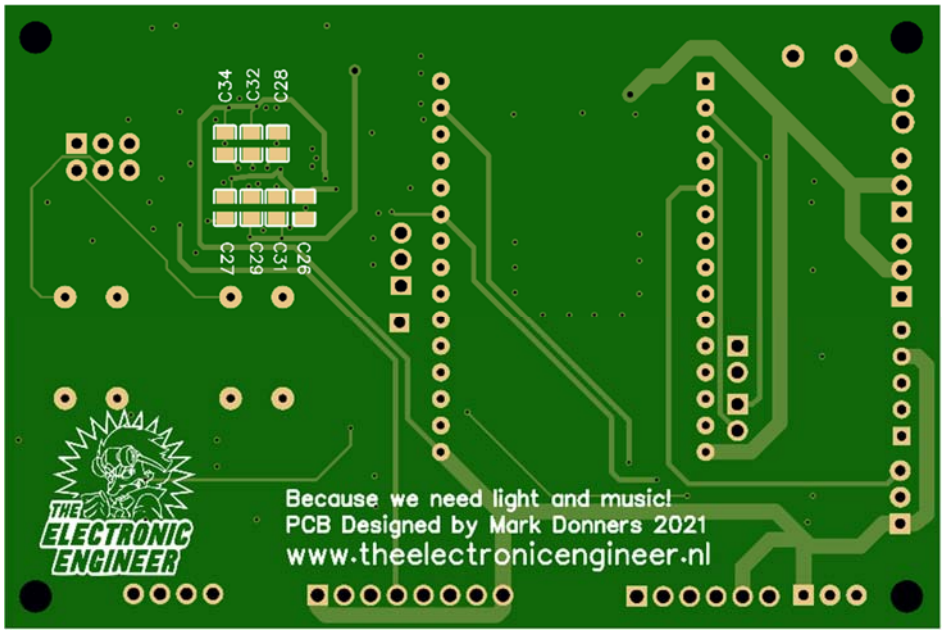
4.1. Main PCB

The pcb without components is available at my Tindie store here: Tindie.

Shipping is only possible within the EU because shipping to other countries will involve import Tax, declarations of conformity and other complicated stuff. Nobody is looking for that kind of complicated formalities or import tax upon receiving the goods. If you are a resident of a country that I don't ship to, you can always order a pcb directly from a PCB supplier near you. The Gerber productions files you need for that are available for download, although it might be more expensive than what you are hoping for. If you need 1 or 2 PCB's, a Tindie order is the cheapest way to go.

4.1.1. Assembly





Assembly notes:

- Assembly of the bottom side are not that difficult all the only components on the bottom are 100nf smd 0805 capacitors (C26,27,28,29,31,32,34)
- Make sure you have the orientation of specific components correct. Resistors and ceramic capacitors are not critical but Electrolytic capacitors and semiconductors have to go on the right way! Do not mix up Cathode and Anode.
- The audio transformers L! and L2 also have an orientation. Make sure to align the letters 'SD' or dot, with the silkscreen of the PCB. **WARNING, I encountered several versions that had this marking on the wrong side of the transformer. This makes it hard to tell which side is primary and which side is secondary. Best is to try it. If you solder everything else first, you can connect a speaker and play some music. You can then simply try what way to mount the transformers. Or use a socket so it is easy to reverse when necessary. When mounted correctly, a clear sound will be what you hear but if it is reversed, all you will hear is digital noise.**
- I Used MOLEX connectors and headers but you can use any type you like if it fits the PCB. The raster used is 2.54"
- Connectors H2, H4, H3 and P1 can be left out. Those are connected to the unused pins of the ESP32 board in case I need them later for experimenting and debugging.
- Use a socket for the fuse and for the ESP32. A socket makes swopping them a lot easier.
- In my experience, it works best to assemble U7 and U8 before the rest. U7 might be the most difficult of the whole PCB and you will need a steady hand as well as experience in soldering SMD components. Safe all the connectors for last.

Here are some links for some of the used components. A more detailed list of components is available on one of the following pages of this document.

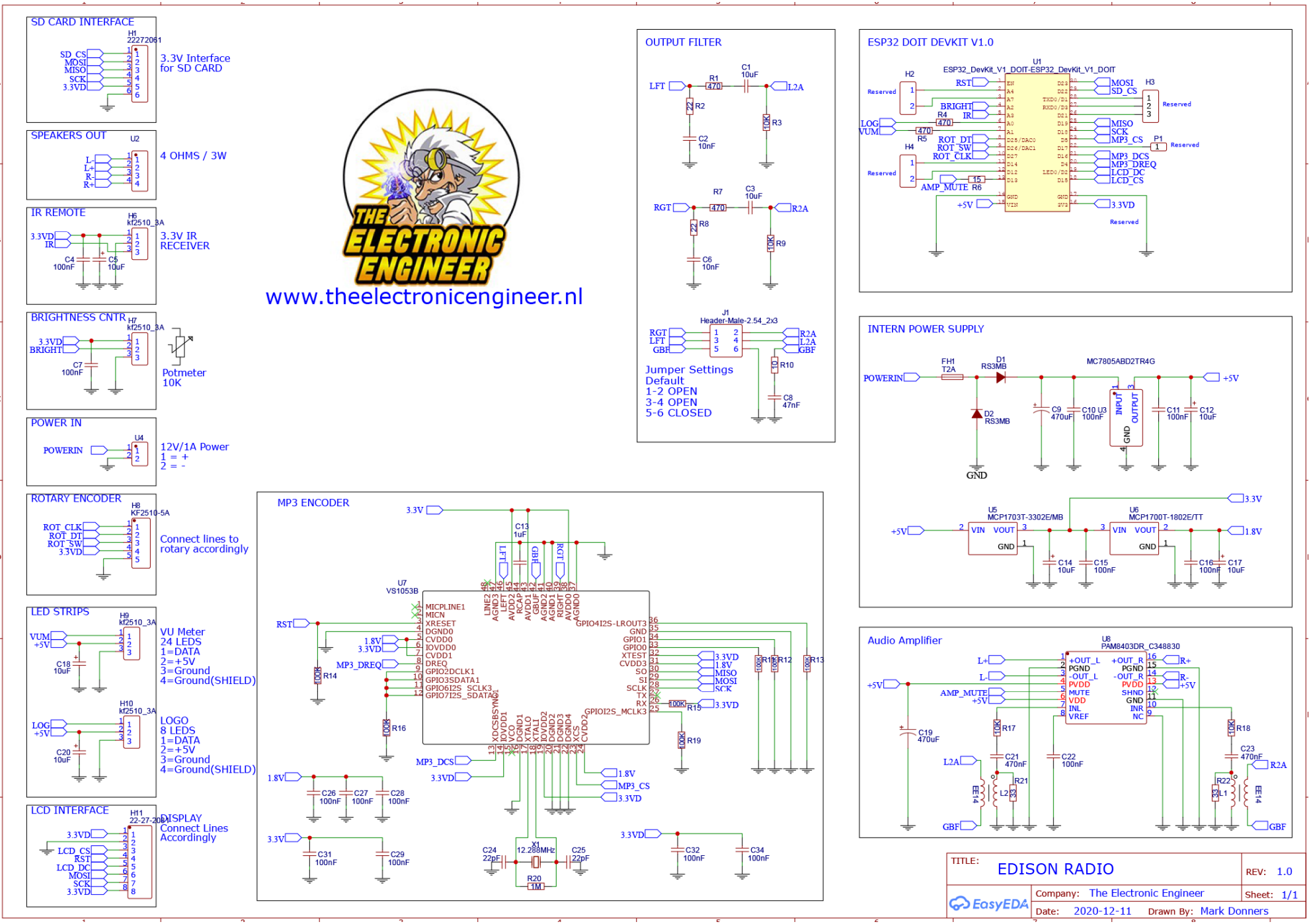
ESP32 controller

https://nl.aliexpress.com/item/4000152270368.html?spm=a2g0s.9042311.0.0.1ec64c4ddCHl0d

Audio transformer

https://nl.aliexpress.com/item/32913784557.html?spm=a2g0s.9042311.0.0.27424c4d6fwORq

4.1.2. Schematic



4.1.3. PCB Part list main PCB

The complete part list for the main component is here:

Designator	Name	Footprint	Quantity	Manufacturer Part	Manufacturer	Supplier	Supplier Part
C13	1uF	C0805	1	CL21B105KBFNNNE	SAMSUNG	LCSC	C28323
C14,C17,C18,C20,C5,C12	10uF	CAP-SMD_L3.2-W1.6-R-RD	6	TAJA106K016RNJ	AVX	LCSC	C7171
C19	470uF	CAP-SMD_BD6.3-L6.6-W6.6-FD	1	VZH471M0JTR-0607	LELON	LCSC	C134834
C22,C7,C11,C10,C4,C15,C16,C27, C28,C29,C31,C32,C34,C26	100nF	C0805	14	CC0805KRX7R9BB104	YAGEO	LCSC	C49678
C23,C21	470nF	C0805	2	CL21B474KBFNNNE	SAMSUNG	LCSC	C13967
C24,C25	22pF	C0805	2	CL21C220JBANNNC	SAMSUNG	LCSC	C1804
C3,C1	10uF	C0805	2	CL21A106KAYNNNE	SAMSUNG	LCSC	C15850
C6,C2	10nF	C0805	2	CL21B103KBANNNC	SAMSUNG	LCSC	C1710
C8	47nF	C0805	1	CL21B473KBCNNNC	SAMSUNG	LCSC	C53134
C9	470uF	CAP-SMD_BD8.0-L8.3-W8.3-RD	1	VZH471M1CTR-0810	LELON	LCSC	C164069
D1,D2	RS3MB	SMB_L4.6-W3.6-LS5.3-RD	2	RS3MB	Shandong Jingdao ME	LCSC	C123923
FH1	T2A	FUSE-TH_BD9.5-P5.08-D1.0	1	56000001009	Littelfuse	LCSC	C142909
H1 *	22272061	CONN-TH_22272061	1	22272061	MOLEX	LCSC	C504992
H11 *	22-27-2081	CONN-TH_22-27-2081	1	22-27-2081	MOLEX	LCSC	C240822
H2,H4 *	Header-Male-2.54_1x2	HDR-TH_2P-P2.54-V	2	826629-2	TE Connectivity	LCSC	C86471
H3 *	Header-Male-2.54_1x3	HDR-TH_3P-P2.54-V	1	Header2.54mm 1*3P	BOOMELE	LCSC	C49257
H7,H6,H9,H10 *	kf2510_3A	CONN-TH_3P-P2.00_KF2510_3A	4	kf2510_3A	BOOMELE	LCSC	C29275
H8	KF2510-5A	CONN-TH_5P-P2.54_KF2510-5A	1	KF2510-5A	BOOMELE	LCSC	C27544
J1	Header-Male-2.54_2x3	HDR-TH_6P-P2.54-V-R2-C3-S2.54	1	Header-Male-2.54_2x3	BOOMELE	LCSC	C65114
L2,L1	EE14	1300:8 Ohm audio transformer	2	1E0836	OOTDTY	Alieexpress	EE14
P1 *	Header-Male-2.54_1x1	HDR-TH_1P-P2.54-V-M	1	Header-Male-2.54_1x1	ReliaPro	LCSC	C81276
R10	10	R0805	1	0805W8F100JT5E	UniOhm	LCSC	C17415
R19,R16,R13,R11,R12,R14,R15	100K	R0805	7	0805W8F1003T5E	UniOhm	LCSC	C17407
R20	1M	R0805	1	0805W8F1004T5E	UniOhm	LCSC	C17514
R21,R22	33	R0805	2	0805W8F330JT5E	UniOhm	LCSC	C17634
R6	15	R0805	1	0805W8F150JT5E	UniOhm	LCSC	C17480
R7,R1,R4,R5	470	R0805	4	0805W8F4700T5E	UniOhm	LCSC	C17710
R8,R2	22	R0805	2	0805W8F220JT5E	UniOhm	LCSC	C17561
R9,R3,R18,R17	10K	R0805	4	0805W8F1002T5E	UniOhm	LCSC	C17414
U1	DOIT DEVKIT ESP32 V1	custom DOIT DEVKIT ESP32 V1	1			Alieexpress	
U2 *	DG308-2.54-04P-14-00A(H)	CONN-TH_DG308-2.54-04P-14-00A-H	1	DG308-2.54-04P-14-00A(H)	DEGSON	LCSC	C708735
U3	MC7805ABD2TR4G	TO-263-2_L10.0-W9.1-P5.08-LS15.2-TL	1	MC7805ABD2TR4G	ON	LCSC	C152462
U4 *	KF128-2.54-2P	CONN-TH_P2.54_KF128-2.54-2P	1	KF128-2.54-2P	Cixi Kefa Elec	LCSC	C474920
U5	MCP1703T-3302E/MB	SOT-89-3_L4.5-W2.5-P1.50-LS4.2-BR	1	MCP1703T-3302E/MB	MICROCHIP	LCSC	C27287
U6	MCP1700T-1802E/TT	SOT-23-3_L2.9-W1.3-P1.90-LS2.4-BR	1	MCP1700T-1802E/TT	MICROCHIP	LCSC	C150796
U7	VS1053B	LQFP-48_L7.0-W7.0-P0.50-LS9.0-BL	1	VS1053B	VLSI	LCSC	C9922
U8	PAM8403DR_C348830	SOP-16_L10.3-W5.4-P1.27-LS7.8-BL	1	PAM8403DR	Diodes Incorporated	LCSC	C348830
X1	12.288MHz	HC-49S_L11.4-W4.8	1	6CS12288F20UCG	SHENZHEN CRYSTAL TECH	LCSC	C252314

* = You can use any header that fits the PCB 2,54 raster. Header ,, ,, ,, are optional, you cn leave them out, the only function for that header is debugging or future options.
I would recommend that you solder the Fuse in a socket for easy exchange when things go south….the same goes for the ESP32 controller board.

4.2. Optional Light Bulb and PCB

To give this radio more of an “Edison” look I decided to implement the optical illusion of a Edison Light Bulb. It is optional and without it, the radio will function just fine. However, this ornament justifies the name of this radio. Doesn’t it? You can leave it out but, you will be missing out on this very cool looking tribute to the world's best inventor.

The light bulb is actually a modified decorative LED light. (Edison Light Bulb) The one I bought is from CALEX holland and is called the E27 Titanium Rustic Lamp art. 473894

https://www.partsnl.nl/led-glasfiber-titanium-st64-rustieklamp-35w-e27-dimbaar-473894

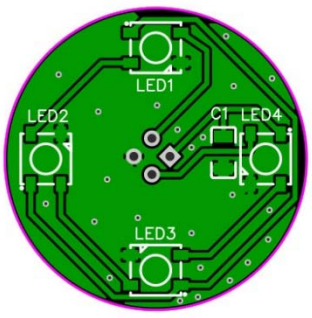
There are several models available and you might have to adjust your wiring and voltage supply if you decide to buy a different light bulb.

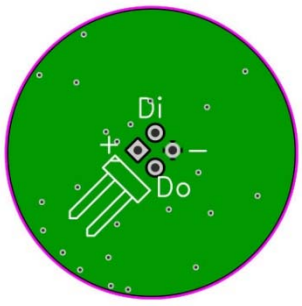
The one I bought was surprisingly easy to modify to my needs. Basically, you can unscrew the outer glass and take out the inner part and socket. I then pulled out the inner cylinder from it's socket by force and removed the LED PCB from the socket. I then saw off the lower part (lamp foot)and re-attached the cylinder to the socket so that it screws back into the outer glass bulb. So, this leaves me with the outer light bulb and the inner cylinder in place. The only difference is that now, the bottom of the cylinder is open and I can attach a small PCB with pixel LEDS to it.

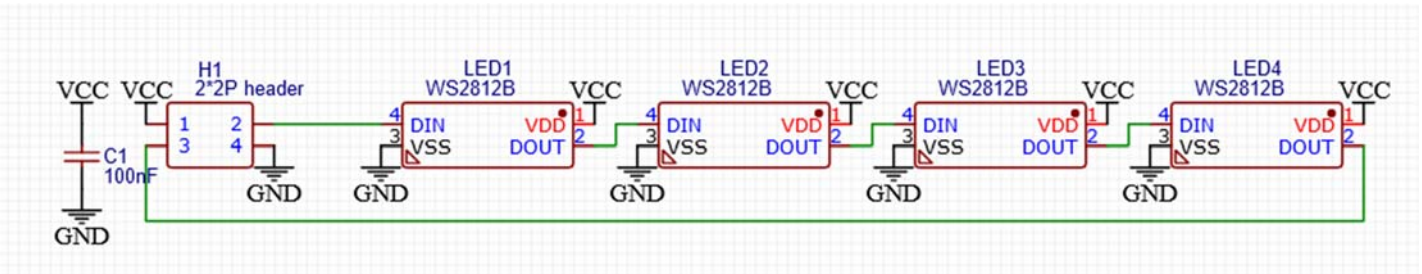
Basically, this PCB only contains 4 pixel LEDS WS2812B and an optional connector or wire input. You can look at it as a ledstrip of 4 LEDS that you can connect serially with the Logo ledstrip. The LEDS can be purchased here:

https://www.banggood.com/10pcs-RGB-WS2812B-4Pin-Full-Color-Drive-LED-Lights-CJMCU-for-Arduino-products-that-work-with-official-Arduino-boards-p-1103143.html

Make sure you place those LEDS in the correct orientation. There is a marked corner on the silkscreen for each LED that must match with the marking on the LED’s. The light bulb PCB will be connected to the WS2812B-LED Strip that is used with the Logo. If you don't use the logo, you can connect the light bulb PCB directly to the main PCB logo Ledstrip connector.







Designator	Name	Footprint	Quantity	Manufacturer Part	Manufacturer	Supplier	Supplier Part
LED1,LED2, LED3,LED4	WS2812B	LED-SMD_4P-L5.0-W5.0-TL	4	WS2812B	Worldsemi	LCSC	C114585

H1	2*2P header	HDR-TH_4P-P2.54-H-R2-C2-S2.54-W10.0	1	2*2P header	BOOMELE	LCSC	C60561
C1	100nF	C1206	1	CL31B104KBCNNNC	SAMSUNG	LCSC	C24497

4.3. Interface components

Ok, so you got a finished PCB? Good! Now it is time to add the surrounding components

You'll need:

Potentiometer
1 potentiometer. This can by any potentiometer with a value between 1K and 50K. A value below 1K is not recommended because of the current it will consume. A value beyond 50K will work but might cause too much noise. I prefer to use a potentiometer with a linear scale of 10K and I choose one that has a switch so that I can use this as a power switch to turn on/off the radio.

Rotary switch
I use a rotary switch for basic operation of the radio. This is optional but if you decide not to use it, you'll have to adjust the configuration file using the browser interface. (More of that later). The rotary switch I used is the one that comes with most Arduino sensor kids and can also be obtained here:
https://www.banggood.com/KY-040-Rotary-Decoder-Encoder-Module-AVR-PIC-p-914010.html?cur_warehouse=CN&rmmds=search

Speakers
The PCB has an onboard amplifier that can handle 2 speakers of 3W / 4Ω. I used these speakers:
Visaton FR7 2.5"art. 305266

https://www.conrad.nl/p/visaton-fr-7-25-inch-64-cm-breedband-luidsprekerchassis-5-w-4-305266

Display
You can choose from several different displays but you'll have to adjust the firmware accordingly. I used a ILI9341 240 x 320 LCD: (This is the default setting):

https://www.banggood.com/2_8-Inch-ILI9341-240x320-SPI-TFT-LCD-Display-Touch-Panel-SPI-Serial-Port-Module-p-1206782.html

VU-Meter LEDSTRIP
the ledstrip to be used for the Vu-Meter is a WS2812B based Ledstrip. You'll have to adjust the firmware to match the number of leds you'll use. Default it is set to 24. I used a ledstrip 96leds/meter:

https://nl.aliexpress.com/item/4001222557975.html?spm=a2g0s.9042311.0.0.27424c4dKZDVlW

Logo leds
I used some of the Vu- meter ledstrip to illuminate the logo. The Logo Ledstrip has to be connected serially with the Light Bulb PCB.

4.4. Acrylic Laser cut Casing.

Although I might not be completely neutral in this matter, I believe that this 'EDISON' Design is very cool and elegant looking…..Wouldn't you want one? If you do, feel free to download the AutoCAD files to 'laser cut' or mechanically engineer your own. It is fine if you prefer to modify that design or use your own creation instead but know that me design is freely available.

I used a laser cutter to cut all the acrylic parts (5mm thickness) for this casing. The space that is to be filled with the VU meter has a plastic cover made of 1mm transparent or 'milk' acrylic that is glued into place. Using a milky colour will defuse the light to look more elegant fluent. You can experiment with different colours of (semi) transparent cut out acrylic to find whatever gives you the best vibe.

All the parts of the casing are glued together using proper glue for the used materials.

Except for the bottom plate what is hold in place by two small screws.

5. Assembly

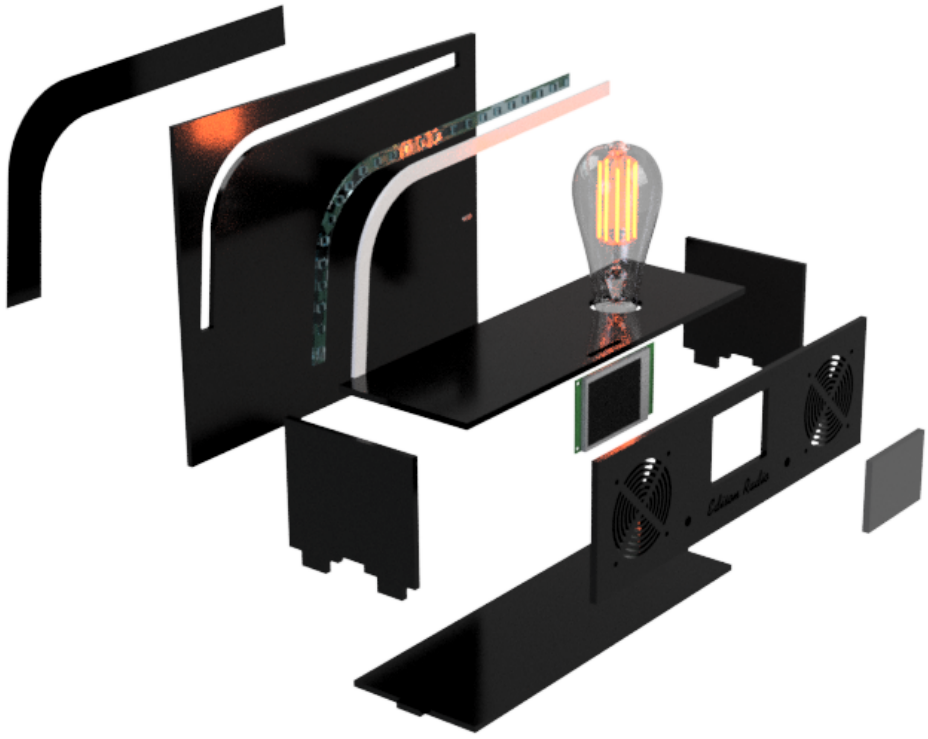
When you have all the components and parts, you can start the assembly.

5.1. Casing

The bottom panel is the only panel that is hold in place by two screws. All other panels are glued into place with acrylic glue. I used Acrifix(1R0192) but your local store might have something similar.

The first step is to glue the backplate of the ledstrip onto the back panel. You can also mount the LCD protection glass if you decide to use it. You will have to experiment with the size because it depends on the size of your cutters' laser beam. I designed mine to be 0,5mm larger in width and height to compensate for it. That way it will fit in place perfectly.

Tip: To assemble and glue everything together, a solid 90° angle will help. I use a metal profile and some clamps as support.



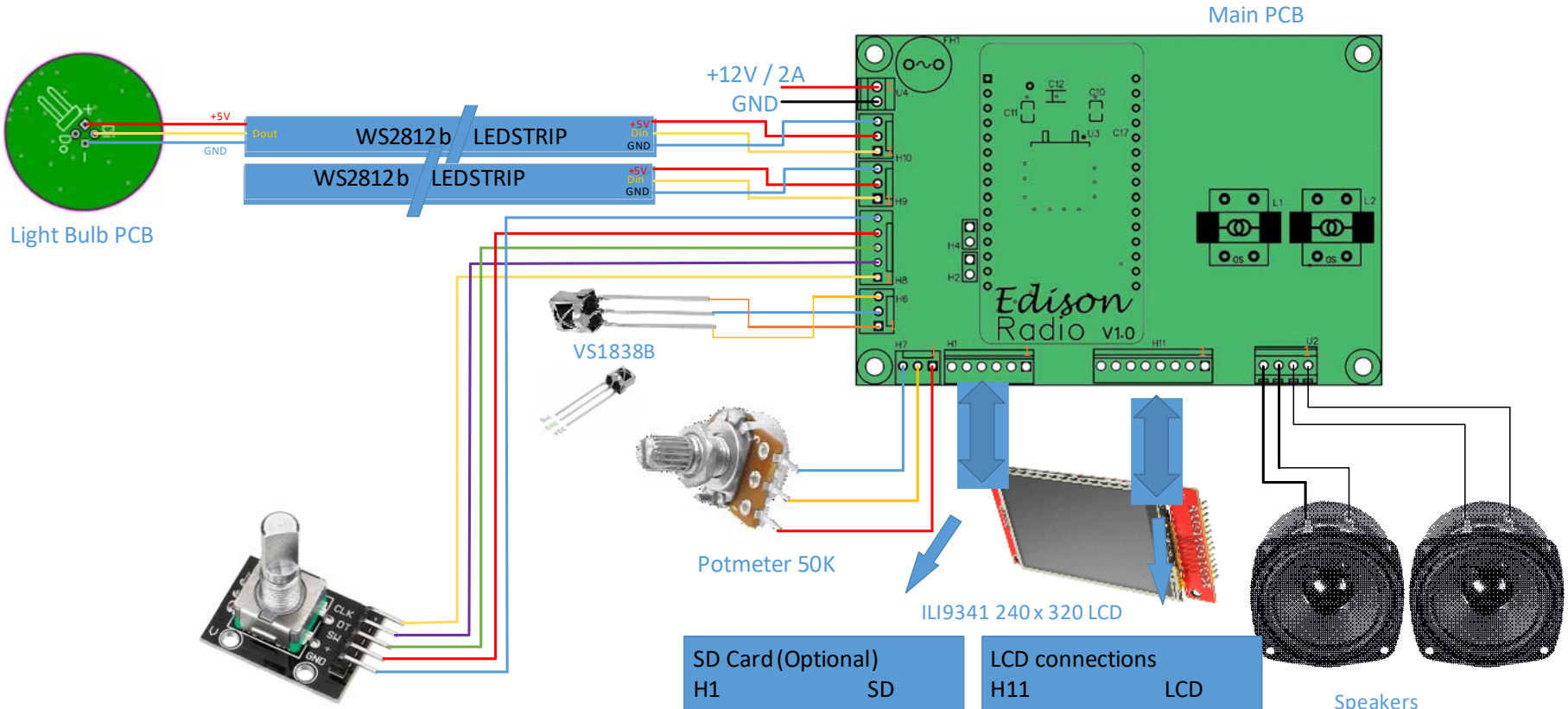
5.2. Mounting everything in its place

This is what I Did:

- First, I glued the front top and sides together.
- I mounted the display, speakers, potmeter, rotary switch, power switch and power entry.
- I started wiring as much as possible.
- I mounted the main PCB to the bottom plate.
- I glued the whole assembly to the backplate.
- I connected the wires to the PCB.
- This would be the time to program your ESP32.
- Test It!
- Close it up by attaching the bottom plate with two little screws.
- I had no use for operating switches or IR receiver. But, you can implement it if you like.

5.3. Wiring

The wiring is not that spectacular. The internal wire to connect the power input to the main PCb should be thick enough to handle about 3A. The Gnd and power wire to the LEDStrips should be able to handle 1A. The rest is not that big of a deal. However, I did twist the speaker wires by channel. It's an old habbit. The rest of the wires I bound together using some straps like T-Raps. Just follow the wiring diagram.



SD Card (Optional)	
H1	SD
1	CD
2	MOSI
3	MISO
4	CLK
5	3.3VDC
6	GND

LCD connections	
H11	LCD
1	3.3VDC
2	GND
3	CS
4	Reset
5	DC
6	MOSI
7	SCK
8	3.3VDC

Speakers
3W / 4 each.

I used:
Visaton FR7 2.5
art. 305266

6. Software

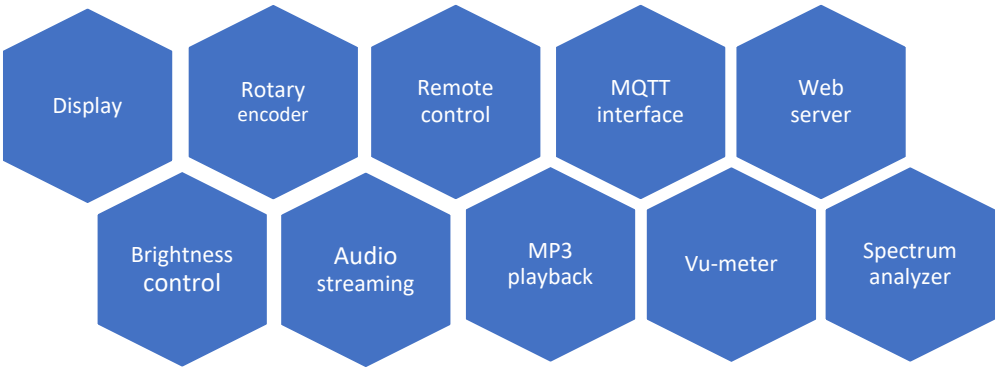
6.1. Firmware

6.1.1. Dual Core usage ESP32

The ESP32 DOIT bord is operating at 160Mhz and it has a dual core! One core is used for streaming the audio to the VS1053 Decoder while the other core is used for all other activities like user interface, Spectrum analyser and webserver.

6.1.2. Firmware sections

The firmware can be divided into several sections:



Display

Rotary Encoder

The rotary encoder switch can control some essential functions of the ESP32-radio.

The default function is volume control. Turning the knob will result in lower or higher volume. Pressing the knob will mute/unmute the signal. The text “Mute” or “Unmute” will be shown on the TFT during 4 seconds.

A double click selects the preset-mode. Rotation of the switch will select one of the preset stations. The preset will be shown on the TFT. Once a preset is selected, you can activate this preset by a single click.
Without rotation, the next preset is selected.

A triple click will select the mp3-player (SD card required). Rotation of the switch will select one of the tracks on the SD card. The track will be displayed on the TFT. Once a track is selected, you can activate it by a single click.
Without rotation, the next track is selected.
After the triple click the player stops, as reading filenames will overload the SD card I/O.

A long click (longer than 1 second) will start playing random tracks from SD card.

After an inactivity of 4 seconds the rotary encoder will return to its default function (VOLUME).

IR Remote Control

Several commands van be linked to any remotecode. While the program is running, a code received by the IR receiver is interpreted by the software as a command if that command is defined in the configuration file.

The radio can be controlled by an IR remote control like this:

To use this interface VS1838B receiver must be connected to Connector H6 of the Edison main PCB

Add the assigned GPIO pin to the preferences through the config page in the web interface

```
pin_ir = 35                    # GPIO Pin number for IR receiver VS1838B
```

The software will read the raw code of the IR transmitter, making it possible to use virtually any remote control to be used. I tested it with the 21 button remote as well as with an LG TV remote.

To assign functions to the buttons, watch the debug log output while pressing a button. For example, press the +volume button. You will see something like:

```
D: IR code 807F received, but not found in preferences!
```

Now add the command:

```
ir_807F = upvolume = 2
```

to the preferences in the config page of the web interface. Likewise you can assign functions to all buttons, for example:

```
ir_8A31 = uppreset = 1
ir_719A = station = us1.internet-radio.com:8105
ir_1F6B = mute
```

MQTT interface

Please take a look at the original coding documentation by ED:

https://github.com/Edzelf/ESP32-Radio/tree/master/doc

Web Server

Please take a look at the original coding documentation by ED:

https://github.com/Edzelf/ESP32-Radio/tree/master/doc

Brightness Control

A Potentiometer is used to controll the brightness of the logo and optional light Bulb. While running, the code will look at the voltage at the potmeter Pin D34 to convert it into a level of brightness.

Audio streaming

One core is dedicted to communicate with the VS1053 chip and ensures a smooth audio stream.

Mp3 playback

If connected, MP3 files that are stored on a SD card can be played.

Vu-Meter

The Vu-meter shows a live interpretation of the Spectrum analyzer data to display the sound level of the audio stream at any given time.

Spectrum Analyzer

The spectrum analysis is done by a plugin that is loaded onto the VS1053 chip. The code retrieves the live data from the VS1053 and update's the LCD screen accordingly.

6.1.3. Operating the radio without webinterface

Rotary encoder (Right knob)

The default function is volume control. Turning the knob will result in lower or higher volume. Pressing the knob will mute/unmute the signal. The text “Mute” or “Unmute” will be shown on the TFT during 4 seconds.

A double click selects the preset-mode. Rotation of the switch will select one of the preset stations. The preset will be shown on the TFT. Once a preset is selected, you can activate this preset by a single click.
Without rotation, the next preset is selected.

A triple click will select the mp3-player (SD card required). Rotation of the switch will select one of the tracks on the SD card. The track will be displayed on the TFT. Once a track is selected, you can activate it by a single click.
Without rotation, the next track is selected.
After the triple click the player stops, as reading filenames will overload the SD card I/O.

A long click (longer than 1 second) will start playing random tracks from SD card.

After an inactivity of 4 seconds the rotary encoder will return to its default function (VOLUME).

Note: I experienced some issue's with response time. I'm not yet sure if this is caused by the added features that take up processor time (like the spectrum analyzer and VU-meter) or if I selected a input pin that is not optimal…I will figure this out in a next update. Honestly, I don't use it that much, other than volume control..

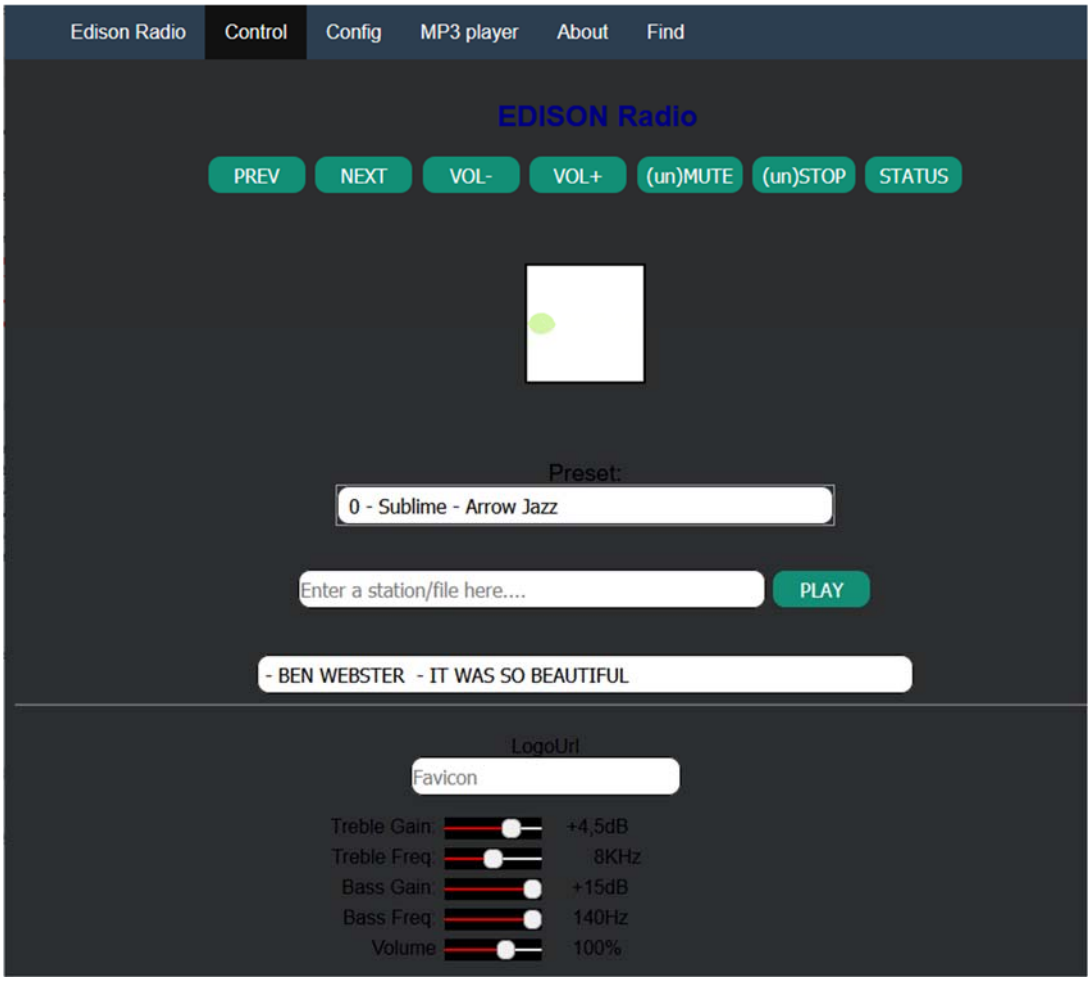
Brightness control (Left knob)

You can use this knob for changing the intensity of the VU meter.

6.2. Browser interface

The browser interface has several screens that can be shown:

Control Screen



This is the screen for main operation. It has the following features:

- Change station to one from the preselected list
- Enter a link to a station manually
- Read back the information that the stream gives you like song title or station name
- See the logo of the station if available in the preselected list
- Adjust the volume, treble and bass
- Mute / unmute

By clicking on the Preset pulldown list, you can change to a pre-selected station

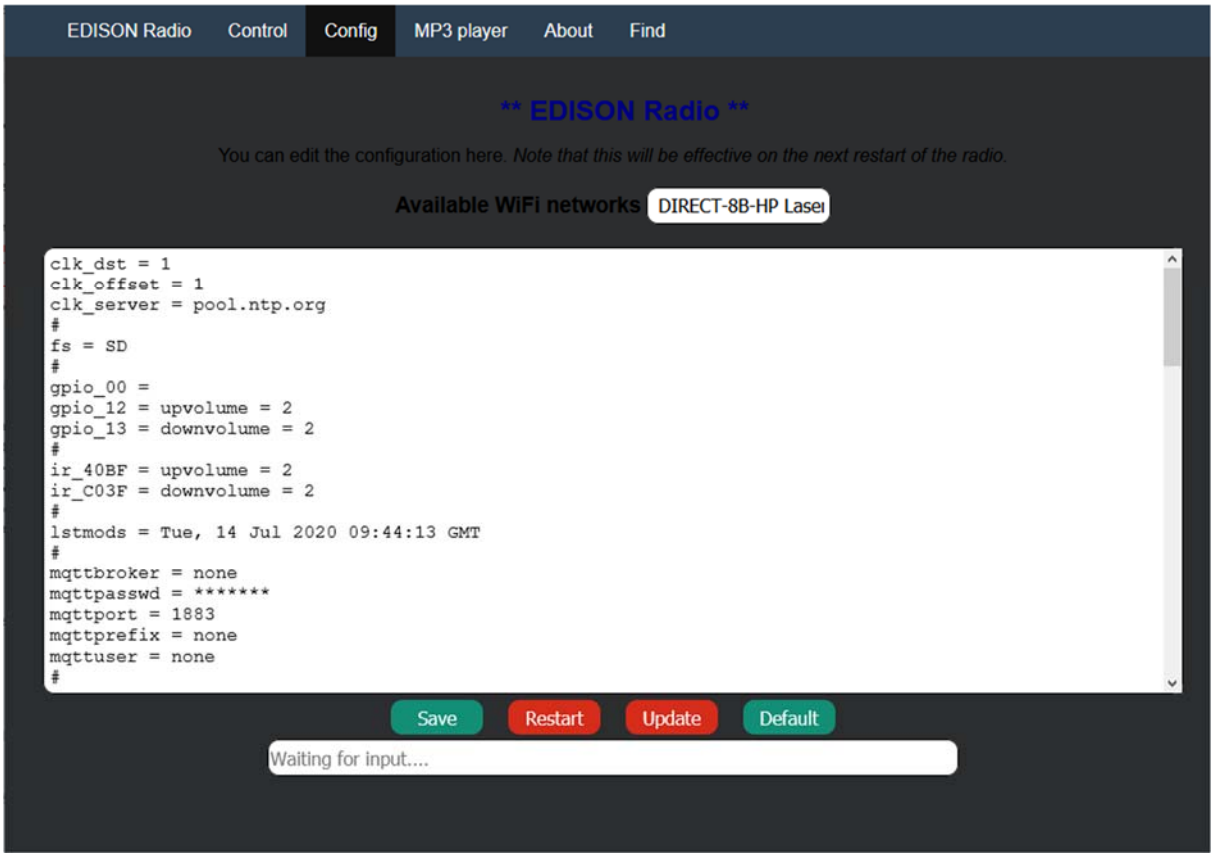
The second input bar can be used to listen to a station by entering a weblink manually and pressing the play button

The third text bar will give you information about the song that is playing or whatever the radio station wants you to read.

The Favicon/Logourl textbar will be removed in a future update. The picture below the buttons will show you the stations Favicon if available. (only for the first 8 presets)

This screen also lets you change volume, treble and gain settings by changing the sliders

Config Screen



You can use this screen to adjust some important setting. Some settings should not be changed if you use the EDISON PCB but, if you are using your own setup it might be necessary. Some settings or functionalities are not used in this edition because I don't use them. However, most of the functionality that has to do with coding is still available if you configure them. I Didn't strip them from the original code as I only implemented new functions. So if needed you can reintroduce those functions if you like. For all the available functions, you best look at the documentation of the original code. You can find it here:

https://github.com/Edzelf/ESP32-Radio/tree/master/doc

Some of the settings that are important for the EDISON edition of this radio are the following:

Settings for the IR receiver

You can link commands for the radio to any IR code by defining it in the configuration file:

Example:

```
ir_8A31 = uppreset = 1
ir_719A = station = us1.internet-radio.com:8105
ir_1F6B = mute
```

Settings for MQTT

Please take a look at the original coding documentation by ED:

https://github.com/Edzelf/ESP32-Radio/tree/master/doc

Settings for Rotary encoder

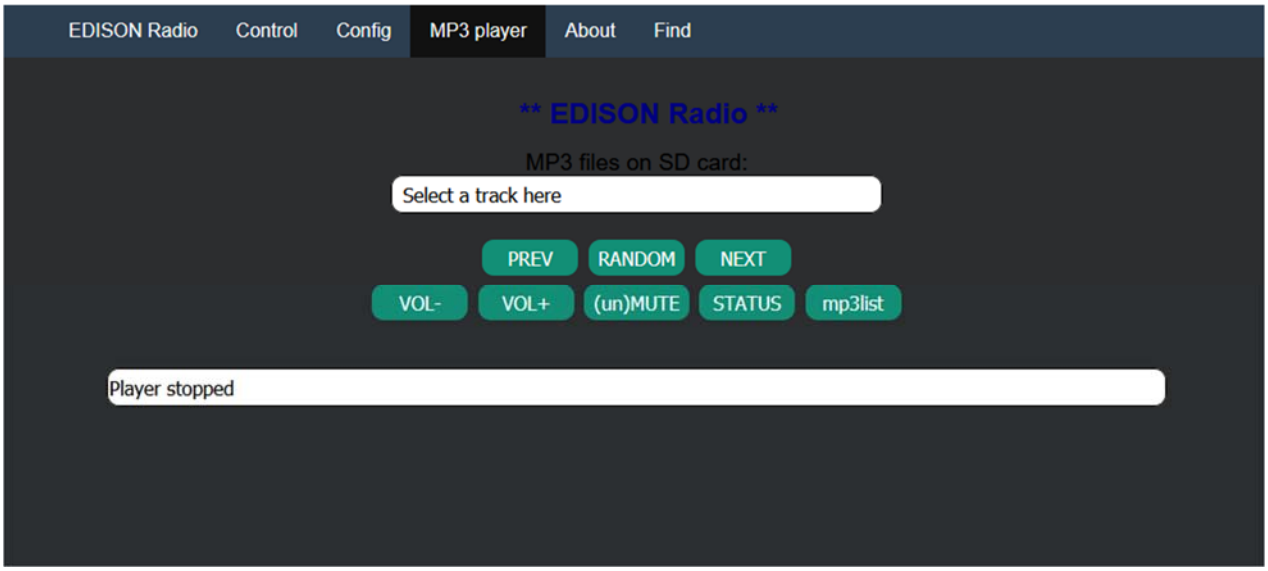
The rotary encoder has a few settings that are fixed if you are using the EDISON PCB. However, if you prefer to switch left and right rotation (clockwise vs counter clockwise) you can exchange the pins on: dt and clk

```
pin_enc_clk = 25          # GPIO Pin number for rotary encoder CLK
pin_enc_dt = 26           # GPIO Pin number for rotary encoder DT
pin_enc_sw = 27           # GPIO Pin number for rotary encoder SW
```

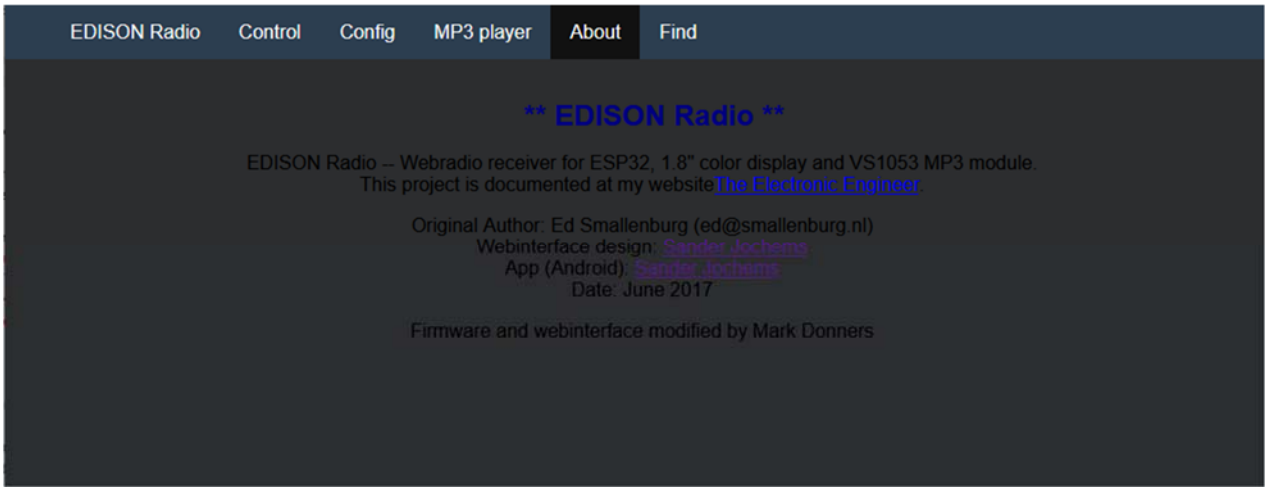
Settings for TFT
Pre-settings for stations and logo's
Preset for startup station
Settings for WIFI

MP3 Screen

If you connect and configure the SD card correctly and it has MP3 Files stored on it, the Radio will scan this card during the boot sequence. You can play back a mp3 file by selecting it from the pull-down menu. You can also play them using a random mode.

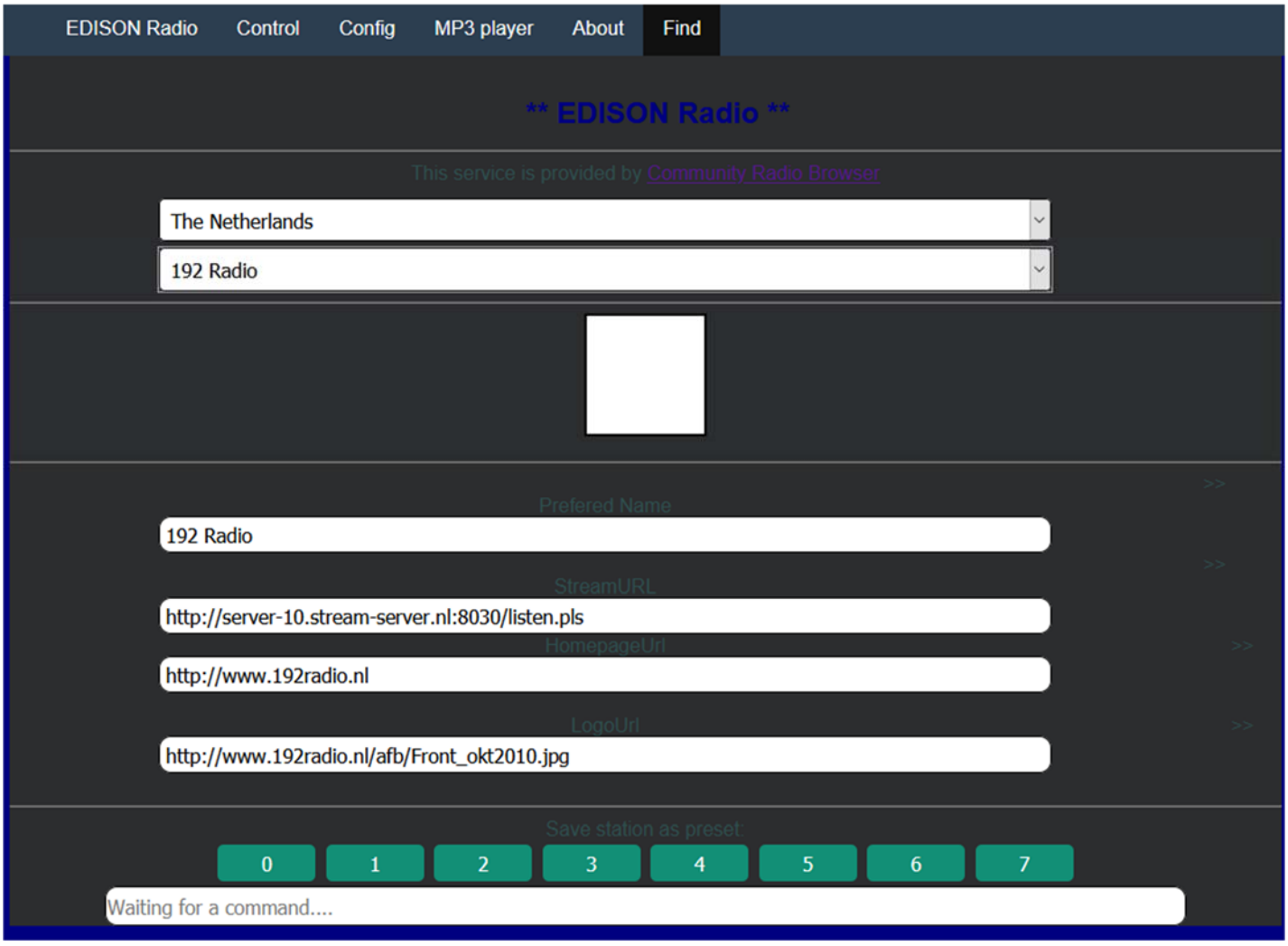


About Screen



The about screen will give you some basic information and show you some credits and stuff

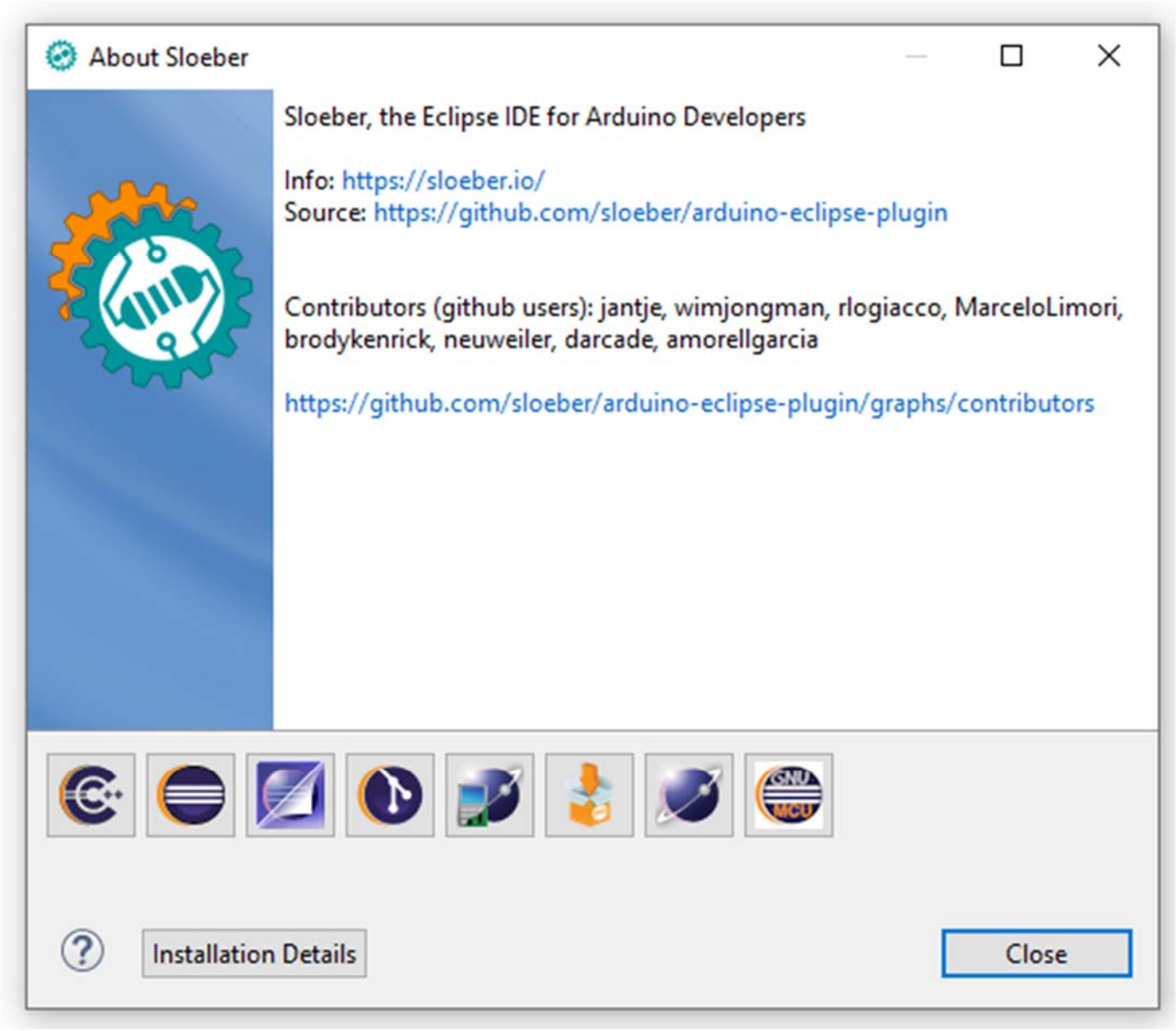
Find Screen



You can use this search engine to find online radio stations, filtered by country. If available, it will show you the streaming URL and the station logo. You can store any station to your pre-selection list by clicking on the buttons at the bottom of the screen. The logo will also be stored. If you want to store more than 8 stations, you can enter the address of the station manually in the configuration file but the logo will not be visible for the stations with number 9 and higher.

7. Programming your ESP32

I used the Arduino IDE. It is freely available online and it does the job. However, I recently stumbled on something called Sloeber Beryllium which is a great tool that offers a much better compiler interface. However, it has a bit of a learning curve but I promise, it's worth it! Why don't you check it out? You can also use Visual Studio or some other great IDE. However, it is important to install the right library and it is best not to install what you don't need as it might give you errors when compiling.



7.1. Here a few libraries that you'll need for sure:

C:\Users\chord\Documents\Arduino\libraries\PubSubClient

C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\SD

Using library Adafruit_NeoPixel at version 1.7.0 in folder:
C:\Users\chord\Documents\Arduino\libraries\Adafruit_NeoPixel

Using library PubSubClient at version 2.6 in folder: C:\Users\chord\Documents\Arduino\libraries\PubSubClient

Using library WiFi at version 1.0 in folder:
C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\WiFi

Using library ESPmDNS at version 1.0 in folder:
C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\ESPmDNS

Using library SPI at version 1.0 in folder:
C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\SPI

Using library ArduinoOTA at version 1.0 in folder:
C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\ArduinoOTA

Using library Update at version 1.0 in folder:
C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\Update

Using library Ethernet at version 2.0.0 in folder: C:\Program Files (x86)\Arduino\libraries\Ethernet

Using library Adafruit_ILI9341 at version 1.5.6 in folder:
C:\Users\chord\Documents\Arduino\libraries\Adafruit_ILI9341

Using library Adafruit_GFX_Library at version 1.10.4 in folder:
C:\Users\chord\Documents\Arduino\libraries\Adafruit_GFX_Library

Using library FS at version 1.0 in folder:
C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\FS

Using library SD at version 1.0.5 in folder:
C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\SD

Using library Adafruit_BusIO at version 1.7.1 in folder: C:\Users\chord\Documents\Arduino\libraries\Adafruit_BusIO

Using library Wire at version 1.0.1 in folder:
C:\Users\chord\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\libraries\Wire

Remark: I had some trouble compiling when I started. Turned out that Arduino IDE had many libraries activated and it decided to choose the wrong ones whenever it had to choose between libraries. I solved it by uninstalling the Arduino IDE and re-installing it from scratch.

7.2. Preparing the ESP32 to use the NVS

WARNING: DO NOT SKIP THIS STEP IF YOU WANT A FUNCTIONAL RADIO

If you start by running the main program without running this little tool first, you will find yourself a radio that doesn't work. It won't start because it cannot find the table with settings that is stored in NVS, causing it to reboot…and.. reboot…and reboot…

So your first step should be to run the initialization tool. It is called Esp32_radio_init and you can find it in the project folder on GITHUB

https://github.com/donnersm/EdisonRadio/tree/main/ArduinoSketch/InitializeTool

You need to run it at least once. Do not worry about the content it puts in the NVS, you can change that later in the main program.

To run it, open the "Esp32_radio_init.ino" with your Arduino IDE (of your own software if you prefer), upload it and start…You can monitor what it is doing by using your serial monitor.

7.3. Uploading the main program

The main program can be downloaded from GITHUB here:

https://github.com/donnersm/EdisonRadio/tree/main/ArduinoSketch/MainProgram

Open it with your Arduino IDE, compile and upload. If you used the same configuration as I did, then it should work without trouble. However, If you used a different display or you want to activate some of the disabled features, you'll need to adjust the code accordingly. If you can't get the information on how to do that from this document, I would suggest to check out the documentation of the original design that was done by Ed Smallenburg. You can find it here:

https://github.com/Edzelf/ESP32-Radio/tree/master/doc

After uploading, there is a big change you will start hearing one of my favourite Jazz stations. Feel free to change them to your likings.