A simple Internet Radio introduction **Ron & Phil**

First – prepare the Arduino IDE

Start with a really simple, fixed-station sketch. This needs only the ESP32 Devkit board, a MAX98357 amplifier, and a small speaker:

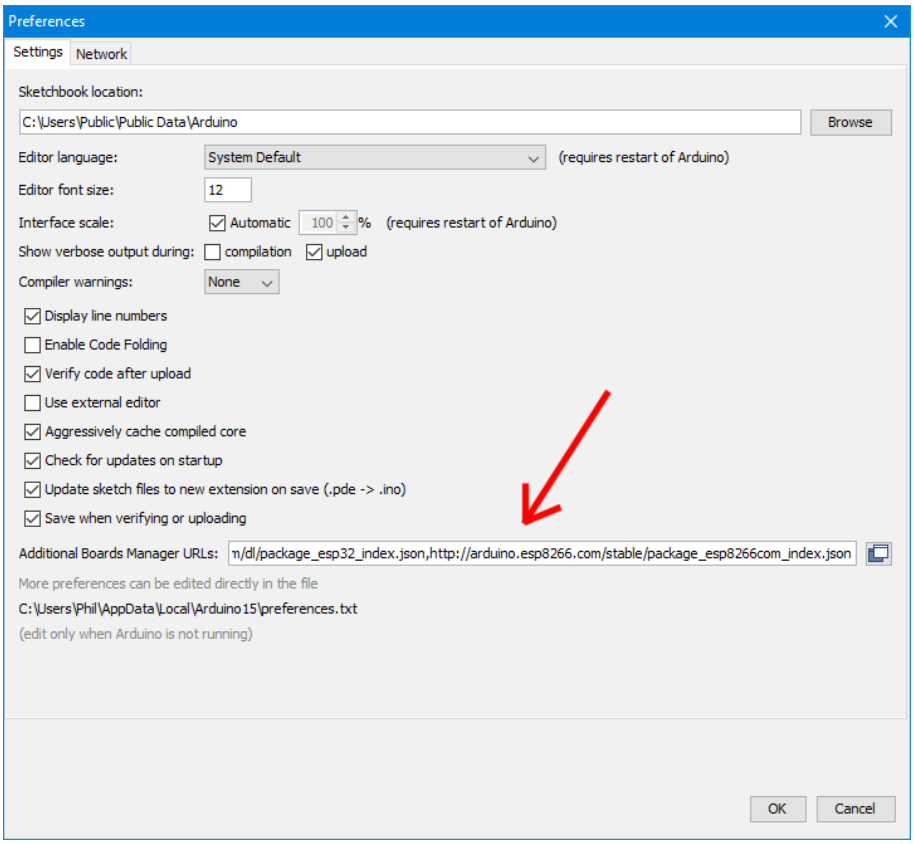
JZK ESP-32S ESP32 Development Board 2.4GHz WiFi + Bluetooth Dual Mode Antenna Module with Ultra Low Power:
https://www.amazon.co.uk/dp/B071JR9WS9/ref=cm_sw_r_wa_api_glt_fabc_GVJDD8CFA2SAV9EXWQK4?_encoding=UTF8&psc=1

...and the MAX98357 amplifier:
https://www.ebay.co.uk/itm/I2S-3W-Class-D-Amplifier-Breakout-MAX98357A-/142502938641?mkcid=16&mkevt=1&_trksid=p2349624.m2548.l6249&mkrid=710-127635-2958-0

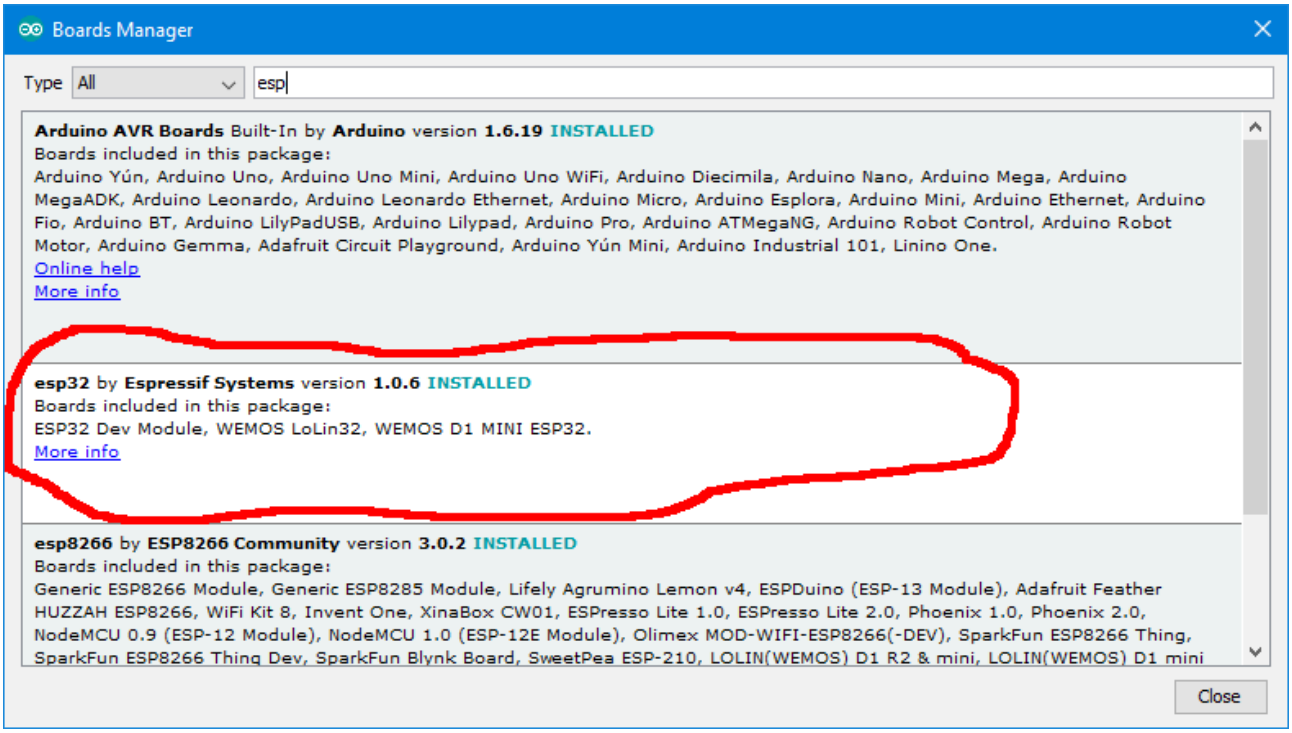
In the Arduino IDE, open 'Preferences' and add the following to Boards Manager, by pasting this line (note that it is TWO urls separated by a comma)

https://dl.espressif.com/dl/package_esp32_index.json,http://arduino.esp8266.com/stable/package_esp8266com_index.json

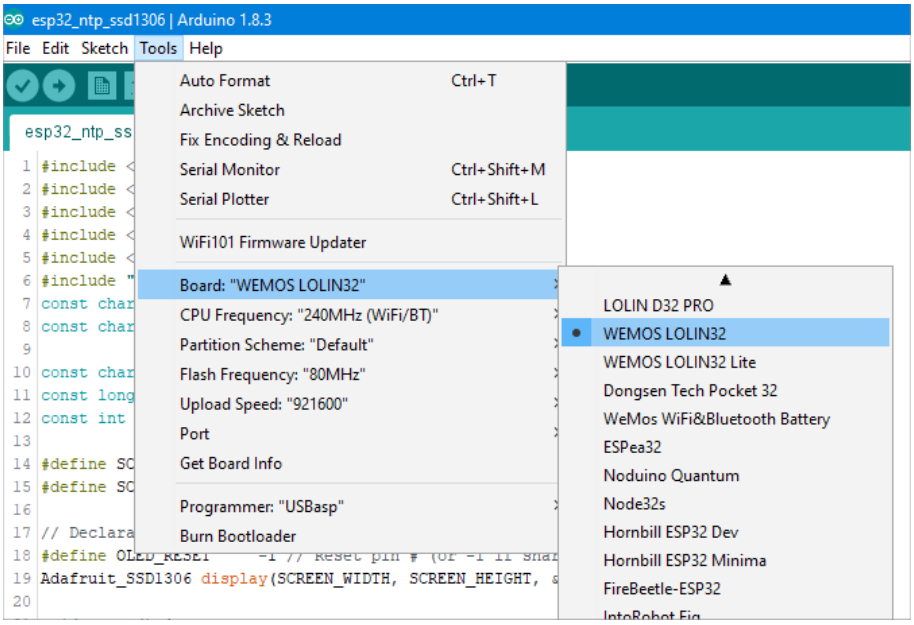
Like this:



Next go to Boards Manager, and search for 'ESP', highlight ESP32.
Click 'More info' then click 'Install'



In the Arduino IDE, select the Board as “WEMOS LOLIN32” like this:



Accept all the defaults for clock speeds, baid rates and memory.

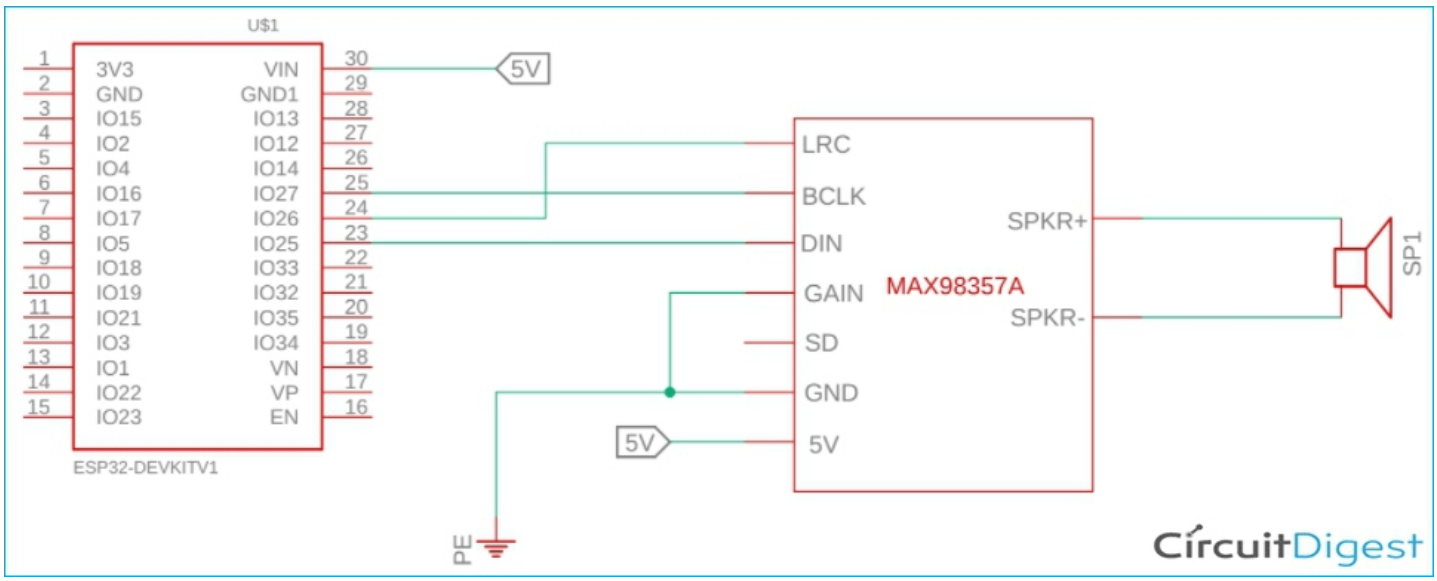
Now, if you connect the ESP32 board just as you would a Nano, via the USB, it should appear as a com port exactly as you'd expect in Arduino.

Try the standard 'Blink sketch – the LED is on D2, this will confirm everything is working ok between the ESP chip and the IDE.

The internet Radio:

“Circuit Digest” have a very simple diagram and sketch:
https://circuitdigest.com/microcontroller-projects/esp32-based-internet-radio-using-max98357a-i2s-amplifier-board

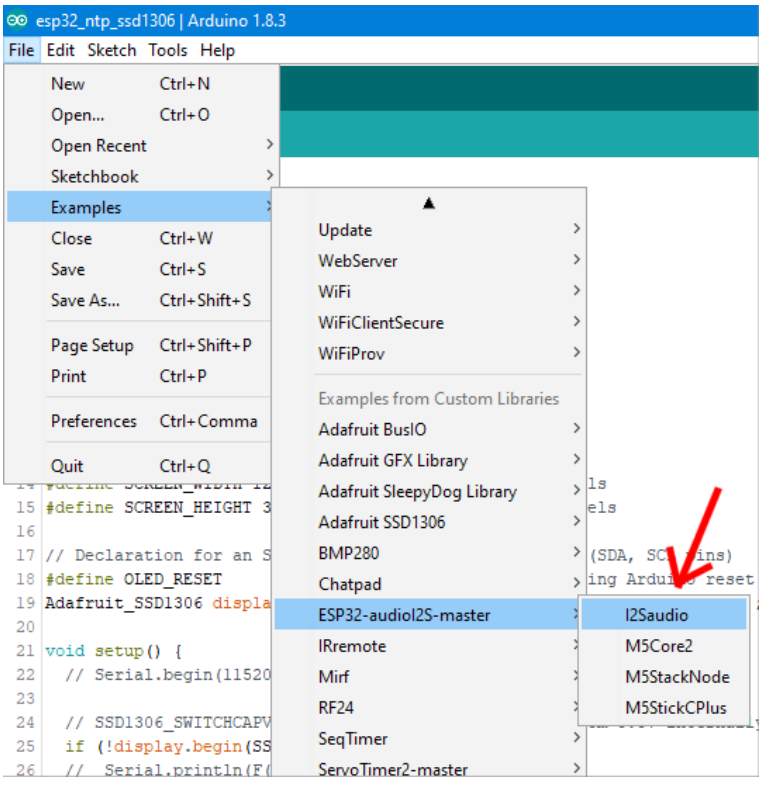
The basic introductory circuit is simplicity itself. All the ESP connections are down one side, so a single header-socket row will suffice. Note that this is a corrected version of their diagram (pin25):



The speaker can be 4 or 8 ohms. The Circuit Digest article suggests the following sketch:

https://circuitdigest.com/microcontroller-projects/esp32-based-internet-radio-using-max98357a-i2s-amplifier-board

However we've found the supplied (built-in to Arduino) example is much better on the same hardware:



This example gives loads of interesting but optional information via the serial Monitor.
In an advanced project you can pick out elements of this information for display.

Either way you will need to add the I2S library (this is how the ESP talks to the amplifier):
https://github.com/schreibfaul1/ESP32-audioI2S

You also need to edit in your own SSID and Wifi password, and your 'station' link of choice.

Some station examples can be found here:
https://groups.google.com/g/uk.d-i-y/c/1Y1KcQ22RJw?pli=1

and here, on the 'Countries' tab:
http://fmstream.org/index.php

Power:
We find a single-cell 18650 Powerbank is ideal. These give a nice stable 5v, automatically switch on & off, and last many hours on a charge.

Next steps:
You'll notice on the max chip the gain pin is grounded. Remove that connection and put a 100k fixed resistor between gain and ground. This will give 15db of gain. Run the max chip from 5 volts.

A stereo version is almost as easy – all the MAX98357 connections except SD are commoned.
The SD pin can be 'programmed' as follows:
SD open circuit – Mono, mix of half-left and half-right.
SD to Vcc – Left channel only
SD via 587k to Vcc – right channel only

Sleep timers, alarms, weather, NTP time displays are all possible once you have his 'starter' working, as is multiple AP wifi access and signal-strength (RSST) indication.

The example plays one fixed station, which can be changed via the serial monitor, but station selection from a list can be added via pushbuttons, a rotary encoder, a hex rotary switch or a pot.
Similarly , the example has a fixed volume. This can also be controlled by adding a pot.
Note that the 'hot' end of any pot or button pull-ups must be to 3v3 NOT to 5v !!!

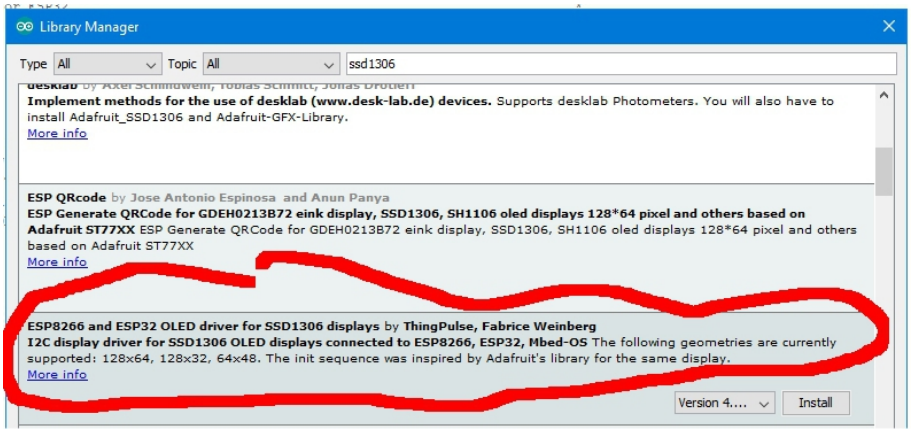
We have used a variety of displays – OLED and LCD, all work well with the appropriate libraries.

We have standardised on using certain ESP pins so that we can share sketches:

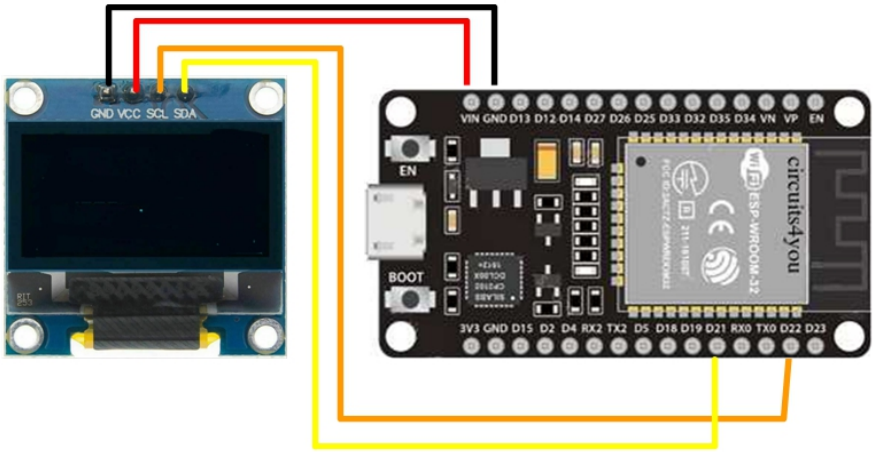
```
#define LED             2 // we use this to indicate wifi connect
#define hex_switch_8    4 // hex switch weight 8 or band switch
#define SD_CS           5
#define hex_switch_4   12 // hex switch weight 4 or band switch
#define hex_switch_1   14 // hex switch weight 1 or band switch
#define hex_switch_2   15 // hex switch weight 2 or band switch
#define SPI_SCK        18
#define SPI_MISO       19
#define display_SDA    21 // for OLED, eg 0.96" or 1.3"
#define display_SCL    22 // for OLED, eg 0.96" or 1.3"
#define SPI_MOSI       23
#define I2S_DOUT       25
#define I2S_LRC        26
#define I2S_BCLK       27
#define volumePin      33   // wiper of volume pot (ADC1_5)
#define tunerPin       39   // optional, wiper of tuning pot if used (ADC1_3)
```

Be aware that there are a lot of libraries so compile time is very slow.

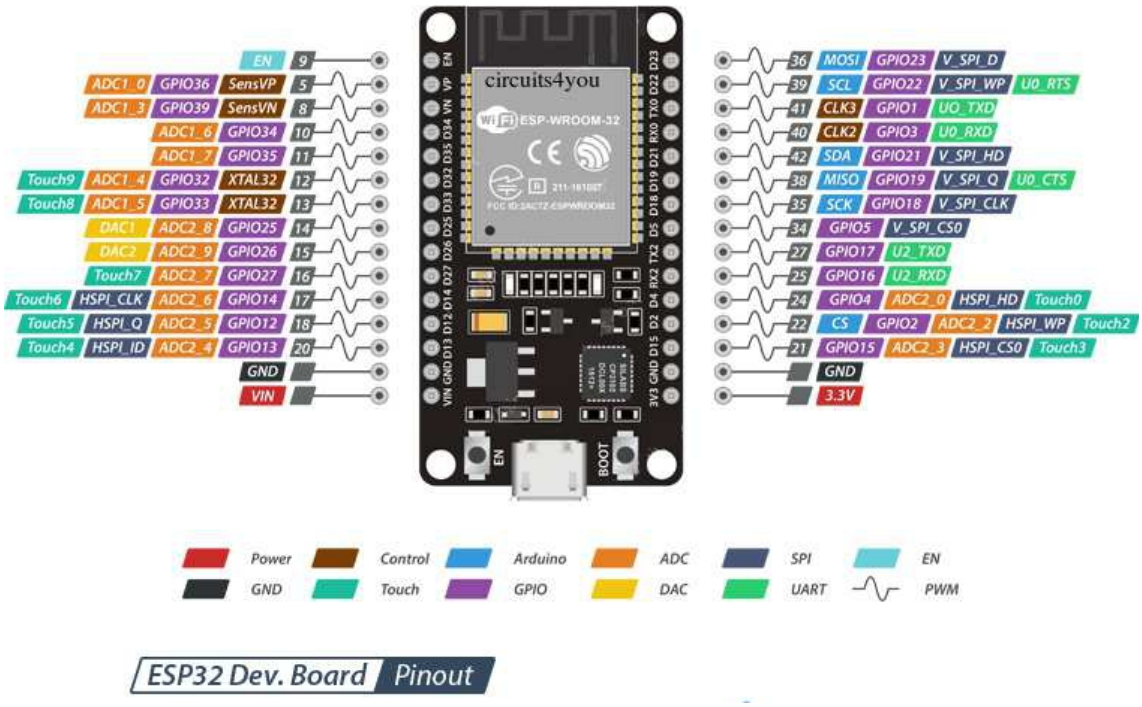
0.96” OLED library:



Example 0.96” OLED (SSD1306) connections:



Here's a good MAX98357A reference:
https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/pinouts



SD / MODE

This pin is used for shutdown mode but is *also* used for setting which channel is output. It's a little confusing but essentially:

- If **SD** is connected to ground directly (voltage is under 0.16V) then the amp is **shut down**
- If the voltage on **SD** is between 0.16V and 0.77V then the output is (Left + Right)/2, that is the stereo average.
- If the voltage on **SD** is between 0.77V and 1.4V then the output is just the Right channel
- If the voltage on **SD** is higher than 1.4V then the output is the Left channel.

This is compounded by an *internal* 100K pulldown resistor on **SD** so you need to use a pullup resistor on SD to balance out the 100K internal pulldown.

For the breakout board, there's a 1Mohm resistor from **SD** to **Vin** which, when powering from 5V will give you the 'stereo average' output. If you want left or right channel only, or if you are powering from non-5V power, you may need to experiment with different resistors to get the desired voltage on **SD**

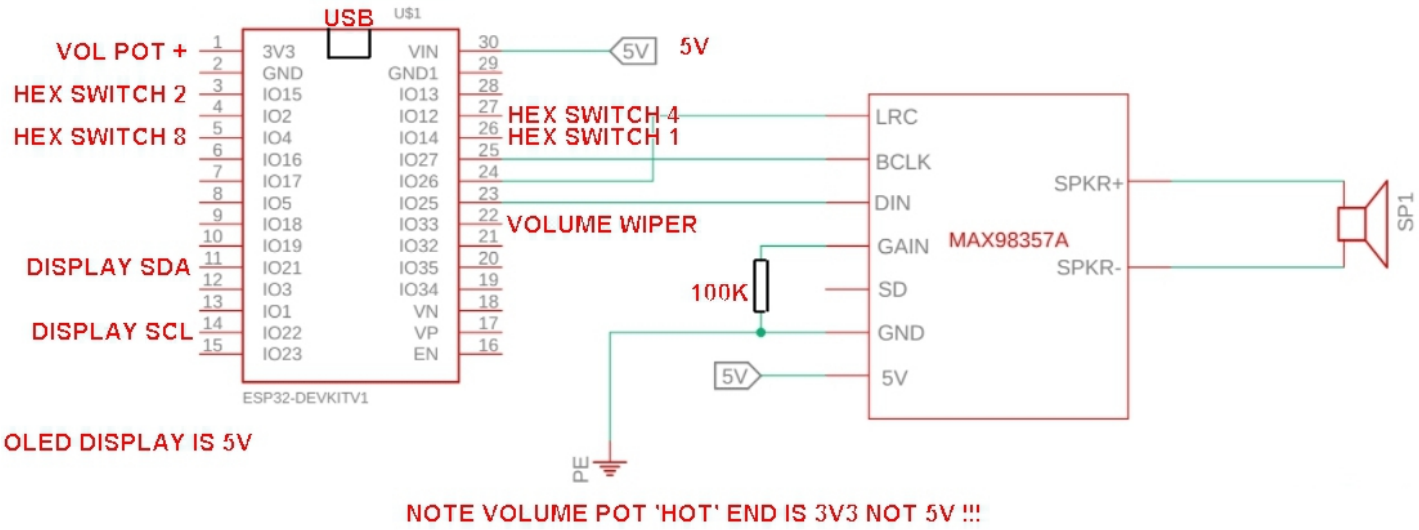
Gain

GAIN is, well, the gain setting. You can have a gain of **3dB, 6dB, 9dB, 12dB** or **15dB**.

- **15dB** if a 100K resistor is connected between **GAIN** and **GND**
- **12dB** if **GAIN** is connected directly to **GND**
- **9dB** if **GAIN** is not connected to anything (this is the default)
- **6dB** if **GAIN** is conneted directly to **Vin**
- **3dB** if a 100K resistor is connected between **GAIN** and **Vin**

This way, the default gain is 9dB but you can easily change it by tweaking the connection to the **GAIN** pin. Note you may need to perform a power reset to adjust the gain.

Phils 16-channel Internet Radio using a rotary hexadecimal switch:



This is the hexadecimal rotary switch I used, RS 197-915
https://uk.rs-online.com/web/p/rotary-switches/0197915/



Ron used this one: RS 655-3235
https://uk.rs-online.com/web/p/rotary-switches/6553235/



BE CAREFUL THAT THE HOT END OF THE VOLUME POT IS 3v3 NOT 5v !!!