

# FM Broadcast Receiver

Basically a bit of a play around with AliExpress FM Modules while also some ESP32 coding for a linear frequency scale on the display and use of the rotary encoder. Three different FM Modules were tested – a TEA5767, SI4703 and a RDA5807. The lack of signal strength here impacted on the results somewhat but the RDA5807 was acceptable. I could not get the station and program information decoded reliably, again I think, from a lack of signal strength.

Another issue was that a number of the units received either didn't work at all or performed poorly. The packing wasn't that flash and while I couldn't see any physical damage there may have been some. All five of the RDA5807 received worked perfectly and these were my preference of the three types tested.

## TEA5767

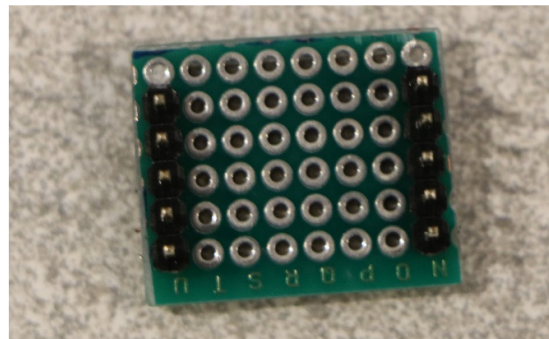
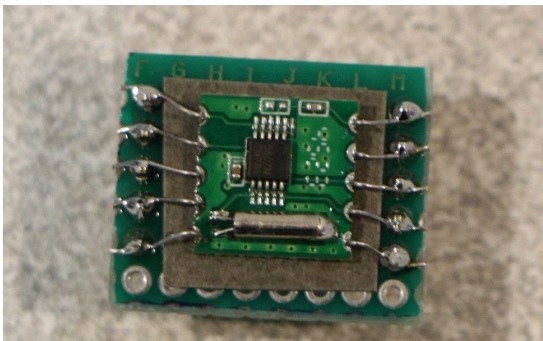
A fairly basic library was used with this. Only one of the three units purchased worked properly and even then there was a lot of background noise. Decided not to use these.

## SI4703

These worked fairly well, at least those that did work. The failure rate was fairly high and that was also the case for a second lot of three from a different manufacturer. The modules seem to be for testing with the earphone leads serving as an aerial. To use an amplifier for speakers and an external aerial a bit of board modification is needed but not too difficult to do. If the track connecting the aerial to the earphone connector is cut for an external aerial connection the common connection on the earphone socket needs to be grounded.

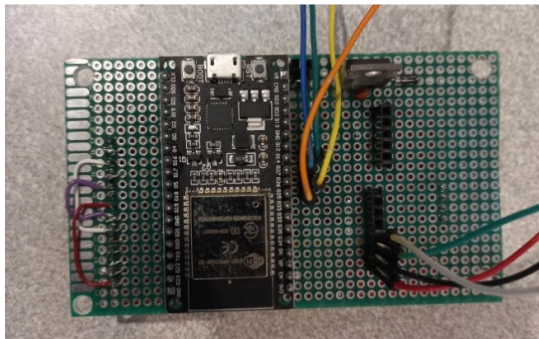
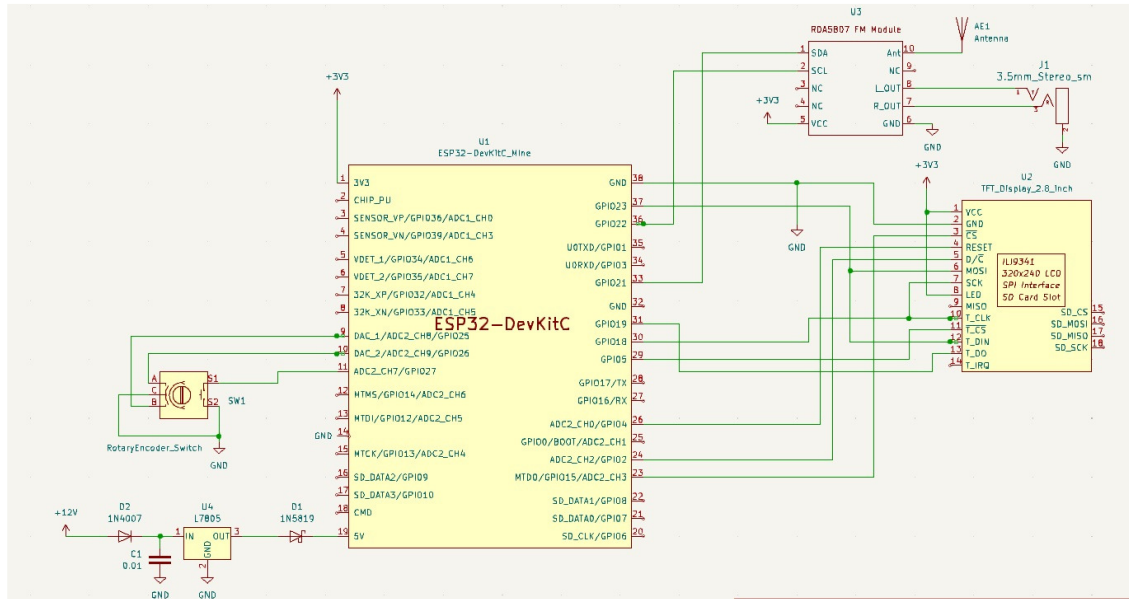
## RDA5807

These performed really well, are easily set up and have enough output to drive earphones directly. The only downside for old fellows like me is the small size – about 10mm square – however it is not too hard to mount them on a board as shown below. All five purchased worked properly.

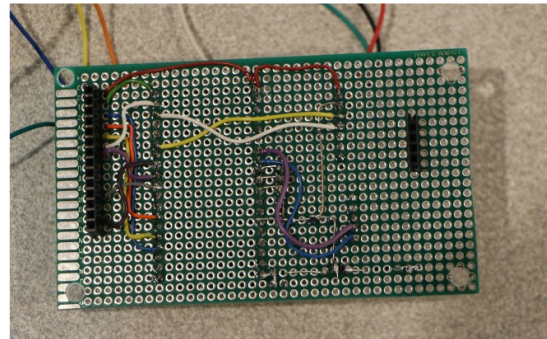


## Hardware

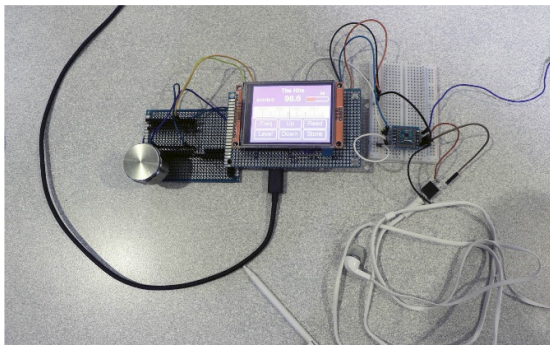
The unit uses an ESP32 Development board with a 2.8 inch TFT touch screen. A rotary encoder is also used. Have just used a test setup at present – the next step is to mount it all in an enclosure with a small audio amplifier to drive some speakers. A 12v supply may be used as shown in the circuit diagram.



ESP32 side of board.



Board top with display removed.



Test circuit layout.



Display Screen.

## Software

A modified copy of the library for the TEA5767 is included here. The other two libraries are:

<https://github.com/pu2clr/SI470X>

<https://github.com/pu2clr/RDA5807>

Both can be installed from github.com or from the Arduino IDE.

TFT\_eSPI needs to be installed and set up for the ESP32 with the display being used. The setup files for devices I used are included.

User\_Setup.h and User\_Setup\_Select.h are in the directory:

C:\Users\yourusername\Documents\Arduino\libraries\TFT\_eSPI

Setup42\_ILI9341\_ESP32.h is in the directory:

C:\Users\yourusername\Documents\Arduino\libraries\TFT\_eSPI\User\_Setups

The FMRx file for the RDA5807 is the most complete because that is the unit I decided to use. There is plenty of scope for changes to be made and the libraries have example programs to get things working.

DON'T GET CAUGHT OUT – If it is not working it might be because the volume is not set.

Press the rotary switch – the rotary will now set volume press it again and it is back to tuning. Find a good station, set the volume and press STORE. That will become the default when the radio is switched on or when READ is pressed.

UP and DOWN move up and down the stations that have been programmed into the code. Press FREQ to have the RDA5807 search up and down the FM Band when UP and DOWN are pressed. LEVEL is used to change/set the minimum signal level for the RDA5807 to stop searching at. When STORE is pressed that value gets stored as well.