How to use this internet radio:

The radio has 5 keys + M/A button (ch+, ch-, vol+, vol-, mute).

The keys are individually debounced to account for their usage.

There is also a debounce procedure for the remote control keys.

Great care has been taken so that whether using the manual keys or the remote control, switching is smooth without any clicks or noise.

For the compilation I use the latest version Arduino IDE (1.8.19).

You must choose the **ESP32 Dev Module module** with the Partition scheme: **default 4MB with spiffs**.

Of course, do not forget to load the Web server by ESP32 Sketch Data Upload from the IDE tool menu.

It is also necessary to modify the command codes of the IR remote control in the source before compilation (lines 117 to 122). On my remote control I programmed the ON/OFF key (CMD_RESTART) so that the system restarts.

To find the codes for your remote control, simply use the **MinimalReceiver** modified to fit this hardware.

There is an **OTA update procedure**, very easy to use and which avoids having to open the box to connect a USB cable.

Just type http://1912.168.xxx.yyy/update to use it.

I strongly advise to consult the site: https://randomnerdtutorials.com/esp32-ota-over-the-air-vs-code/ to understand the use.

I have attached the software and webserver (spiff) bin files in the program folder.

If you have any problems with the compilation, do not hesitate to contact me via the Labs!

First use

For the first use, it is essential to initialize the radio, by holding down the **'Mute' button while powering up.**

The display indicates when the button can be released.

The default values are as follows and can be set in the source before compilation**: SSID and Password** (lines 76 and 77 of firmware 1.50), as well as the string **String stations[ ]** (line 88) containing a series of values: url (without http:// or https://) and displayed name, url, name, …

The **maximum length of name is 12 characters** and **70 characters for url** .

At **default init**, these values are saved in the Flash (NVS) of the ESP32 after being **alphabetically sorted by name**.

The default volume (10) and default channel (1) values as well as the maximum number of stations that can be stored in the NVS Flash memory (MAX_STATION = 50) are also saved in the NVS.

Note: I haven't tested more than 50 stations, but I think you can go up to 70 stations without worry.

The software calculates the total number of stations.

Embedded Web Server

The web server is accessible at the IP address indicated on the display.

This web server allows you to:

- View date and time updated by NTP.

- To display the current values such as: volume, channel/total number, name of the station and url, the RSSI of the WiFi signal. These values are refreshed in real time (Websockets).

- You can also select the channel of your choice and the sound volume after confirming with ok.

- You can also configure SSID and PASSWORD of WiFi. The system reboots automatically in case of change.

- You can also enter a url, and name and number of the channel in the Flash memory.

The next available channel is automatically set after each input of new data or if the page is reloaded.

The maximum number of stations is adjusted accordingly (+1 if a station is added, no modification if it is just a replacement).

The list of channels is automatically sorted by alphabetic order if a new name and url is entered.

- There is a default button which is equivalent to the procedure (Mute + Power ON) described above and a boot button which allows you to restart the system.

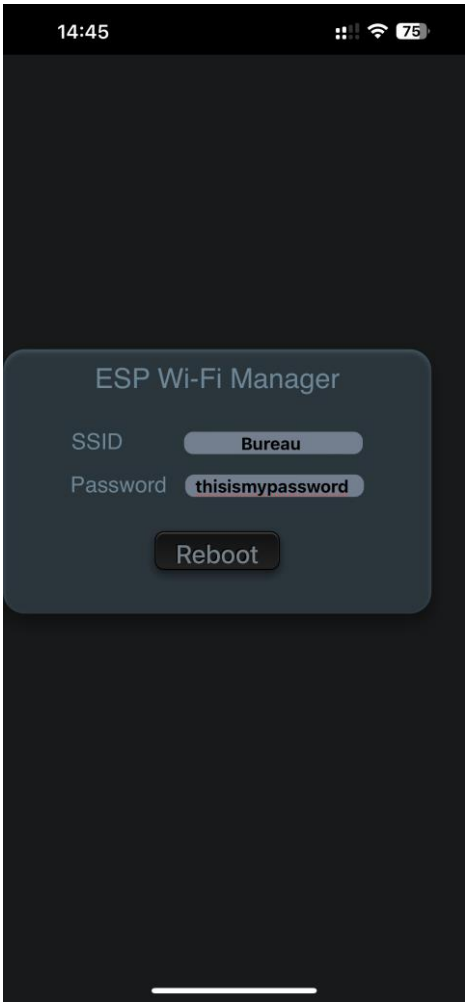
Debug

The serial port (115200 bauds) allows viewing a large amount of information and debugging.

New feature

At first startup or if the WiFi is no more available, connect to ESP32-RADIO on your Phone and launch http://192.168.4.1 .

A small embedded server allows you to choose the Wifi credentials. After Reboot and as long as the WiFi is ok, this page will never be displayed again.



My iPhone after connecting to 192.168.4.1 on ESP-RADIO WiFi