

## ESP-radio

This document describes the realization of an Internet radio based on an ESP8266 WiFi chip.

The ESP8266 is a remarkable thing. It has a Wi-Fi interface and a powerful processor with enough memory to store a complicate application. The Internet radio described here uses a VS1053 module to decrypt the MP3 stream and a 1.8 TFT color display to give some info about the radio station that's playing.

### Features:

- Can connect to thousands of Internet radio stations that broadcast MP3 audio streams.
- Uses a minimal number of components.
- Has a preset list of maximal 63 favorite radio stations in EEPROM.
- Can be controlled by a tablet or other device through a build-in webserver.
- Optional one button control to skip to the next preset station.
- The strongest available WiFi network is automatically selected.
- Heavily commented source code, easy to add extra functionality.
- Debug information through serial output.

### Software:

The software for the radio is supplied as an Arduino sketch that can be compiled for the ESP8266 in de Arduino IDE. No Arduino is required in this project.

The following libraries are used:

- ESP8266WiFi – for establishing the communication with WiFi
- SPI – for communication with VS1053 and TFT display
- Adafruit\_GFX – for writing info on the TFT screen
- TFT\_ILI9163C – driver for the TFT screen
- EEPROM – to store favorite stations
- ESP8266WebServer – for remote controlling the radio via http.

### Configuration:

In order to work properly, the software needs some configuration. In the global data section of the program the available networks (SSID) and their passwords must be specified in the following 2 lines:

```
const char*    networks[] = { "NETW-01",    "NETW-02",    "NETW-03" } ;
const char*    passwords[] = { "PASSWD-01", "PASSWD-02", "PASSWD-03" } ;
```

Optional:

The table "hostlist" can be edited to create an initial list of presets. You can search for suitable stations at <http://www.internet-radio.com>. The format of the list is like:

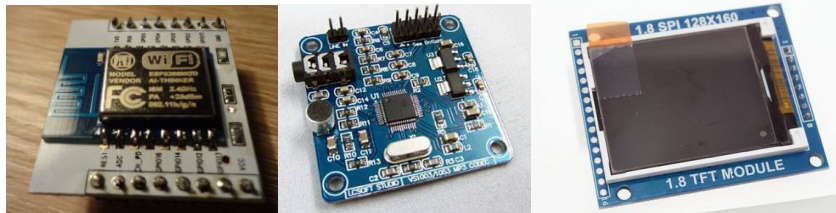
```
const char* hostlist[] = { "stations from remote control", // Reserved entry
                           "109.206.96.34:8100",
                           "us1.internet-radio.com:8180",
                           "us2.internet-radio.com:8050",
                           "50.7.173.162:8097",
                           "195.154.167.62:7264",
                           "198.154.106.104:8985",
                           NULL } ;
```

The first entry of this list is reserved. It is used at startup to determine if the EEPROM has been initialized or not. When the first entry (index 0) in EEPROM does not contain the string in hostlist[0], the stations in "hostlist" will be copied to the EEPROM. Additional stations can be stored in the EEPROM by the web interface.

## Hardware:

The radio is built with the following hardware:

- An ESP-12 module. This is basically an ESP8266 on a small print. There is pull-up on CH\_PD and a pull-down on GPIO15. If the ESP8266 is used without this module you have to provide at least the pull-up to set the ESP8266 to work. See figure 1.
- A VS1063 module. This can be ordered at several Chinese web shops. See figure 2.
- A 1.8 inch color TFT. This can be ordered at several Chinese web shops. See figure 3.
- Two small speakers.
- A Class D stereo amplifier to drive the speakers.
- A 3.3 volt LDO to provide power for the ESP8266.



The radio is powered by a 5 V adapter. The TFT and VS1053 work on 5 Volt. For the ESP8266 a small regulator (LD1117S33TR), 3.3 Volt 800 mA is used.

I used a small perfboard to connect the ESP8266 and the TFT and to mount it in a small speakerbox. The TFT is visible through a hole in the front of the box:

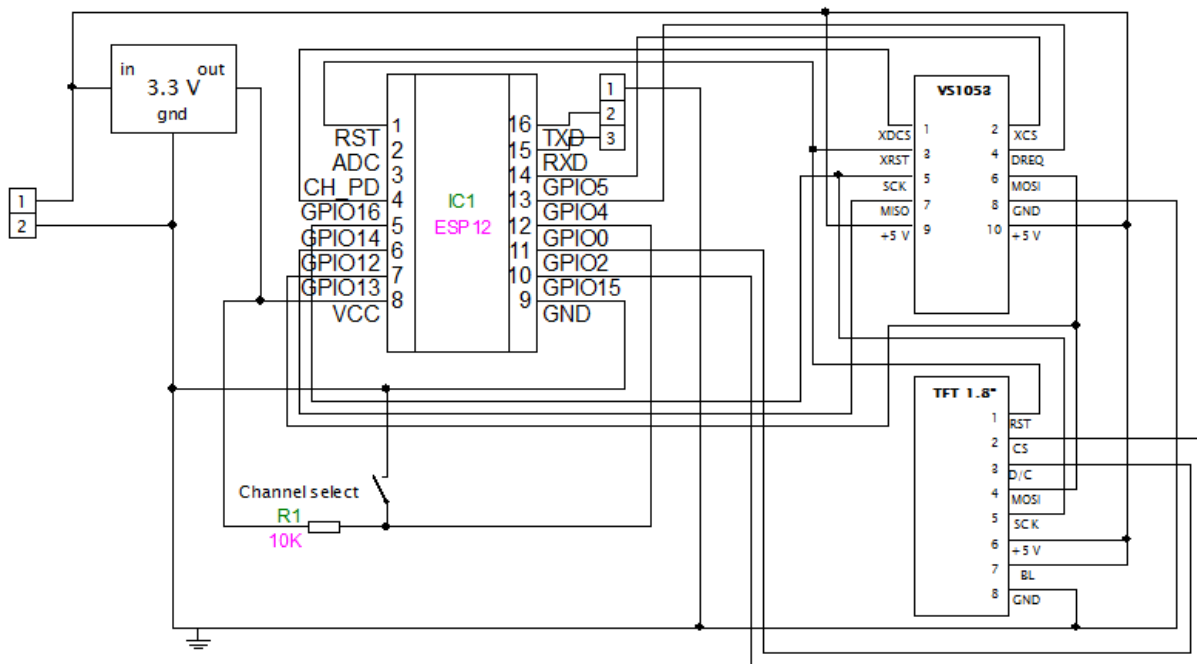


Most of the wiring is done on the green perfboard. The TFT is visible. The VS1053 is connected by the grey cable:



The Class D amplifier is not shown in this picture. In my version I use just mono playback.

## Schematic diagram:



## Wiring:

The logic wiring as shown in the diagram is also presented in the table below. The analog amplifier and the speakers are not included.

NodeMCU	GPIO	Pin to program	Wired to LCD	Wired to VS1053	Wired to rest
D0	GPIO16	16	-	pin 1 DCS	-
D1	GPIO5	5	-	pin 2 CS	LED on nodeMCU
D2	GPIO4	4	-	pin 4 DREQ	-
D3	GPIO0	0 FLASH	-	-	Control button
D4	GPIO2	2	pin 3 (D/C)	-	-
D5	GPIO14	14 SCLK	pin 5 (CLK)	pin 5 SCK	-
D6	GPIO12	12 MISO	-	pin 7 MISO	-
D7	GPIO13	13 MOSI	pin 4 (DIN)	pin 6 MOSI	-
D8	GPIO15	15	pin 2 (CS)	-	-
D9	GPIO3	3 RXD0	-	-	Reserved for serial input
D10	GPIO1	1 TXD0	-	-	Reserved for serial output
VCC	-	-	pin 6 (VCC)	-	-
RST	-	-	pin 1 (RST)	pin 3 RESET	-

## Web interface:

The web interface is not completed yet. The basic idea is to have a html page with embedded javascript that displays an interface to the radio. Command to the radio can be sent to the http server on the ESP8266. The IP address of the webserver will be displayed during startup.

## Capabilities of the webserver:

Let's assume that the IP of the Esp-radio is 192.168.2.13. From your browser you can show a simple root page by entering the following URL: <http://192.168.2.13>

The following will be displayed for example:

```
Esp-radio by Ed Smallenburg  
Listening to NAXI LOVE RADIO (NAXI,Belgrade,Serbia, NAXI,Beograd,Srbija) - 128k  
StreamTitle=' Bob Marley - One Love
```

To select the next station type: <http://192.168.2.13/command?next=1> and the next preset station will be selected. All commands start with `"/command?"`. This will be followed by a parameter, a `"="`-sign and a value. The next commands are available:

- `volume = [up] [down] [percentage]` - Percentage between 0 and 100
- `next = 1` - Select next channel for listening
- `station = address:port` - Store new station and select it
- `delete = 1` - Delete current playing station