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 information 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. Passwords are kept in the SPIFFS filesystem.
- Heavily commented source code, easy to add extra functionality.
- Debug information through serial output.
- Big ring buffer for smooth playback.

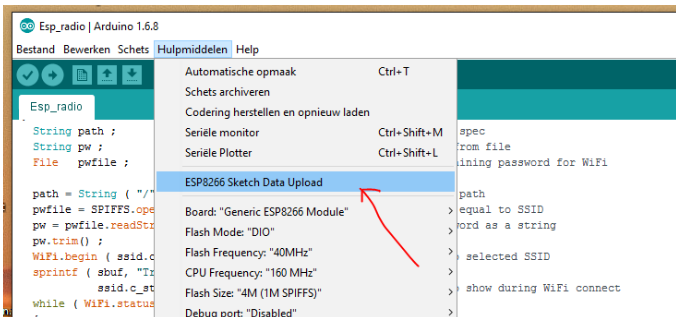
Software:

The software for the radio is supplied as an Arduino sketch that can be compiled for the ESP8266 in de Arduino IDE version 1.6.8, esp8266 software 2.2.0. 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
- ESPAsyncWebServer – for remote controlling the radio via http.
- ESPAsyncTCP – Needed for webserver.
- ESP8266mDNS (optional) for getting a local hostname for the device.

Configuration:

In order to work properly, the software needs some configuration.
The filesystem (SPIFFS, set to 1 MB) of the ESP8266 must contain the files necessary for the webserver and a password file for every acceptable network SSID. The plug-in for Data Upload must be present in your Arduino IDE:

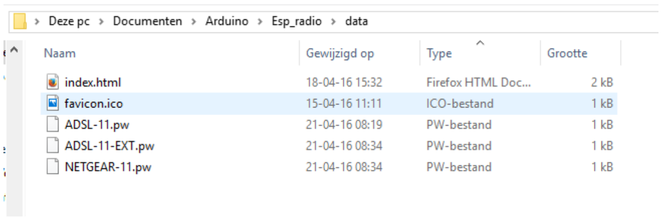

Esp_radio | Arduino 1.6.8
Bestand Bewerken Schets Hulpmiddelen Help
Esp_radio
Automatische opmaak Ctrl+T
Schets archiveren
Codering herstellen en opnieuw laden
Seriële monitor Ctrl+Shift+M
Seriële Plotter Ctrl+Shift+L
ESP8266 Sketch Data Upload

Board: "Generic ESP8266 Module"
Flash Mode: "DIO"
Flash Frequency: "40MHz"
CPU Frequency: "160 MHz"
Flash Size: "4M (1M SPIFFS)"
Debug port: "Disabled"

```
String path ;                                    spec
String pw ;                                     from file
File   pwfile ;                                 ining password for WiFi

path = String ( "/"                              path
pwfile = SPIFFS.ope                              equal to SSID
pw = pwfile.readStr                             ord as a string
pw.trim() ;
WiFi.begin ( ssid.c                              selected SSID
sprintf ( sbuf, "Tr
          ssid.c_st                              show during WiFi connect
while ( WiFi.status
```

The files must be present in a map “data” in the Arduino project map, For Example:



Naam	Gewijzigd op	Type	Grootte
index.html	18-04-16 15:32	Firefox HTML Doc...	2 kB
favicon.ico	15-04-16 11:11	ICO-bestand	1 kB
ADSL-11.pw	21-04-16 08:19	PW-bestand	1 kB
ADSL-11-EXT.pw	21-04-16 08:34	PW-bestand	1 kB
NETGEAR-11.pw	21-04-16 08:34	PW-bestand	1 kB

Optional:
The table “hostlist” must exist and 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:

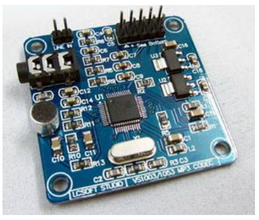
```
const char* hostlist[] = { "Stations from remote control",  // Reserved entry
                           "109.206.96.34:8100",
                           "us1.internet-radio.com:8180",
                           "us2.internet-radio.com:8050",
                           "50.7.173.162:8097",
     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

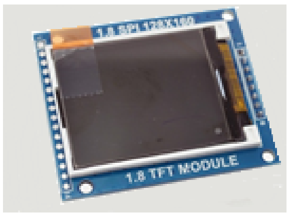
The first entry of this list is reserved. It is used at startup to determine if the EEPROM has been initialized or not. When the first entry (index 0) in EEPROM does not contain the string in hostlist[0], the stations in “hostlist” will be copied to the EEPROM. Additional stations can be stored in the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

Hardware:

The radio is built with the following hardware:

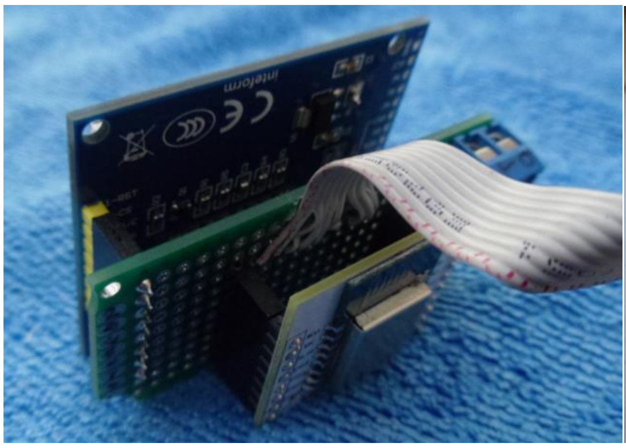
- An ESP-12 module. This is basically an ESP8266 on a small print. There is pull-up on CH_PD and a pull-down on GPI015. If the ESP8266 is used without this module you have to provide at least the pull-up to set the ESP8266 to work. See figure 1. The ESP8266 is running on 160 MHz.
- A VS1063 module. This can be ordered at several Chinese web shops. See figure 2.
- A 1.8 inch color TFT display. This can be ordered at several Chinese web shops. See figure 3.
- Two small speakers.
- A Class D stereo amplifier to drive the speakers. Best quality if powered by a separate power source.
- A 3.3 volt LDO to provide power for the ESP8266.

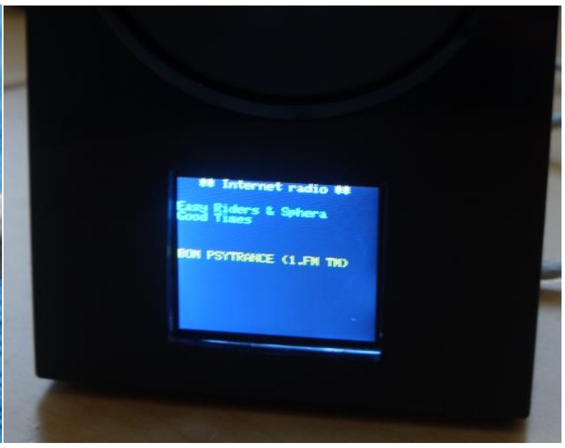




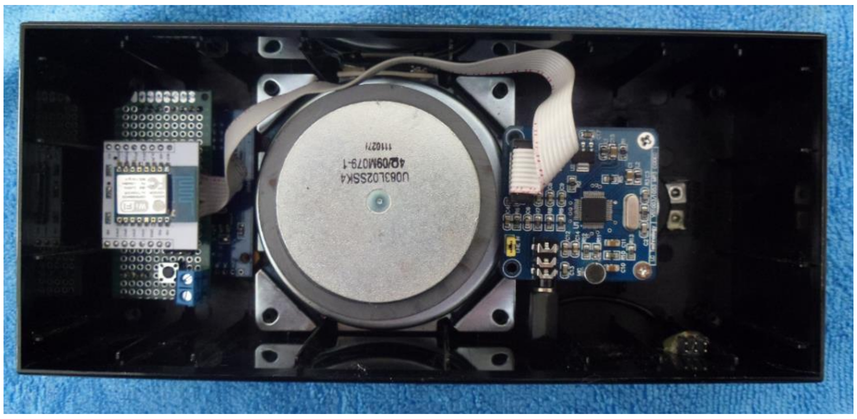
The radio is powered by a 5 V adapter. The radio will function on single LiPo cell as well, so I used a small charge circuit powered by the 5 V input. The TFT and VS1053 work on 3.8 to 5 Volt. For the ESP8266 a small regulator (LD1117S33TR), 3.3 Volt 800 mA is used.

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:



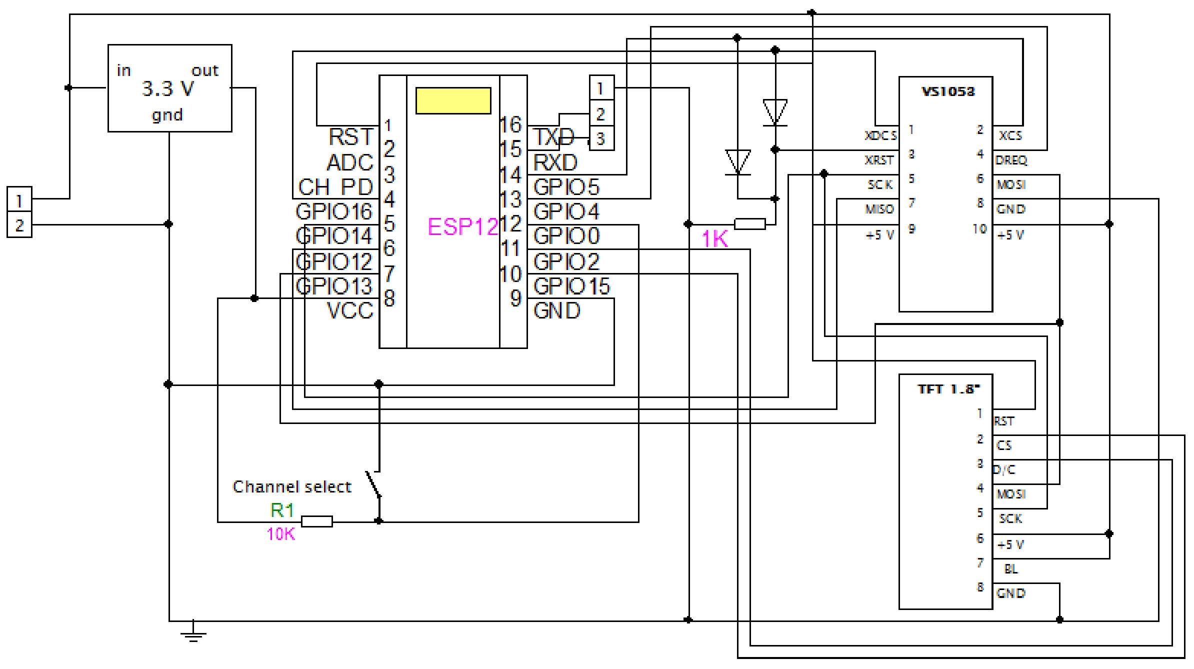


Most of the wiring is done on the green perfboard. The TFT is visible The VS1053 is connected by the grey cable:



The Class D amplifier is not shown in this picture. My version uses 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	-
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	GPI03	3 RXD0	-	-	Reserved for serial input
D10	GPIO1	1 TXD0	-	-	Reserved for serial output
GND	-	-	pin 8 (GND)	pin 8 GND	Power supply
VCC 3.3	-	-	pin 6 (VCC)	-	LDO 3.3 Volt
VCC 5 V	-	-	-	pin 9 5V	Power supply
RST	-	-	pin 1 (RST)	pin 3 RESET	Reset circuit

The reset circuit is a circuit with 2 diodes to GPIO5 and GPIO16 and a resistor to ground (wired OR gate) because there was not a free GPIO output available for this function.

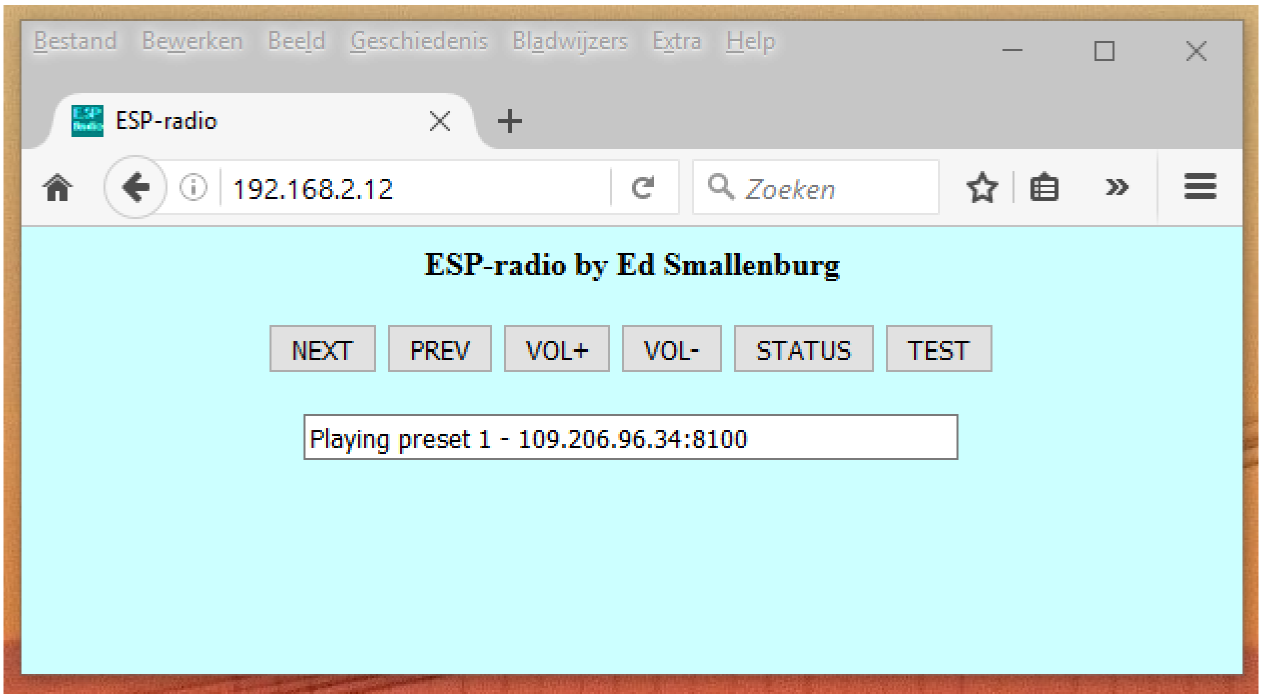
Web interface:

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 om 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 192.168.2.12. From your browser you can show a simple root page by entering the following URL: http://192.168.2.12. This will display the /index.html file from the SPIFFS as well as /favicon.ico.

The following simple web interface will be displayed:



ESP-radio by Ed Smallenburg

Playing preset 1 - 109.206.96.34:8100

Clicking on one of the available buttons will control the Esp-radio. The reply of the webserver will be visible in the status box below the buttons. A click will be translated into a command to the Esp-radio in the form:

http://192.168.2.13/?<parameter>=<value>

For example: http://192.168.2.13/?upvolume=2

Not all functions are available as buttons in the web interface shown above, but working commands are:

• volume	= 95	- Percentage between 0 and 100
• upvolume	= 2	- Add percentage to current volume
• downvolume	= 2	- Subtract percentage from current volume
• preset	= 5	- Select preset 5 station for listening
• uppreset	= 1	- Select next preset station for listening
• downpreset	= 1	- Select previous preset station
• station	= address:port	- Store new preset station and select it
• delete	= 0	- Delete current playing station
• delete	= 5	- Delete preset station number 5
• status	= 0	- Show current station:port
• test	= 0	- For test purposes
• debug	= 1 or 0	- Switch debugging on or off