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 complex 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 or three 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.
- Update of software over WiFi (OTA).
- Volume and preset are saved over restart.

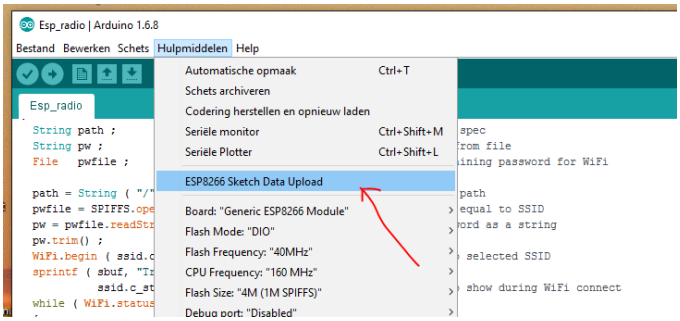
Software:

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:

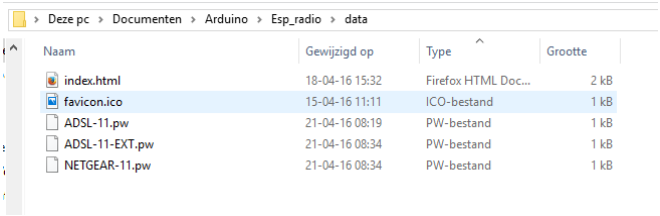
- ESP8266WiFi – for establishing the communication with WiFI
- SPI – for communication with VS1053 and TFT display
- Adafruit_GFX – for writing info on the TFT screen (if configurated)
- TFT_ILI9163C – driver for the TFT screen (if configurated)
- EEPROM – to store favorite stations
- ESPAsyncWebServer – for remote controlling the radio via http.
- ESPAsyncTCP – Needed for webserver.
- ArduinoOTA for software update over WiFi.

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:



If you don't see this feature, check the installation guide at the end of this document.
The files must be present in a map "data" in the Arduino project map, For Example:



Note: If only one password file (*.pw) is present in the filesystem, the radio will connect to that SSID. This is very useful when using hidden SSID.

.pw file(s)

A ".pw" file must be present in the filesystem for every WiFi network you intend to connect to. The content is in readable format, editable by any text editor like "Notepad".
Example: Assume your SSID is "PS4" and your password is "welcome123". In this case you must prepare a file with the name "PS4.pw" that contains the string "welcome123". The file SSID.pw in my Github repository is an example. Rename this file according to your SSID and edit the first line in it.

Optional:

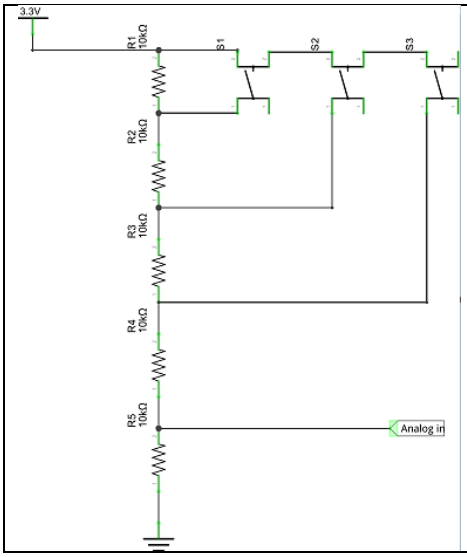
The table "hostlist" must exist and can be edited to create an initial list of presets. You can search for suitable stations at http://www.internet-radio.com. The format of the list is like:

```
const char* hostlist[] = { "Stations from remote control",  // Reserved entry
                           "109.206.96.34:8100",
                           "us1.internet-radio.com:8180",
                           "us2.internet-radio.com:8050",
                           "skonto.ls.lv:8002/mp3",
                           "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.

Analog input:

The analog input can be used to control the radio by connecting the analog input of the ESP8266 to the following circuit:



The voltage of the analog input may differ depending on the ESP8266 module used (nodeMCU or standalone). Therefore the definitions of asw1, asw2 and asw3 have to be corrected in the source. In the web interface you may use ”http:// <IP>/?analog=1” to read the actual voltage (ADC units) while pressing one of the 3 buttons.

3 button digital control:

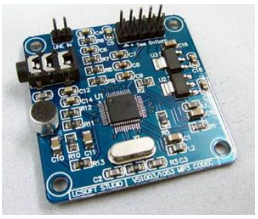
Normally the radio is controlled by the web interface. However, one digital input (GPIO0) may be connected to a button to skip to the next preset station.
If TFT output is not required, 2 extra digital inputs are available. You may use the 2 extra inputs for 2 extra control buttons (GPIO2 and GPIO15): “goto preset station 1” and “goto previous preset station”.
In this case you have to comment out the definition “USETFT” in the source code of the sketch.

Note that GPIO15 has to be LOW when starting the ESP8266. The button for GPIO15 must therefore be connected to VCC (3.3V) instead of GND. The software will invert this input.

Hardware:

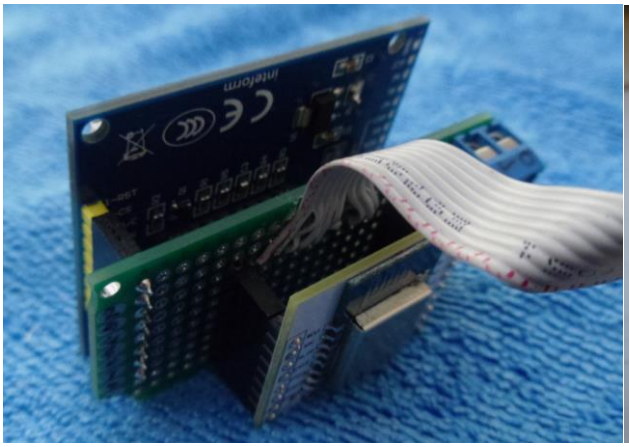
The radio is built with the following hardware:

- 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.

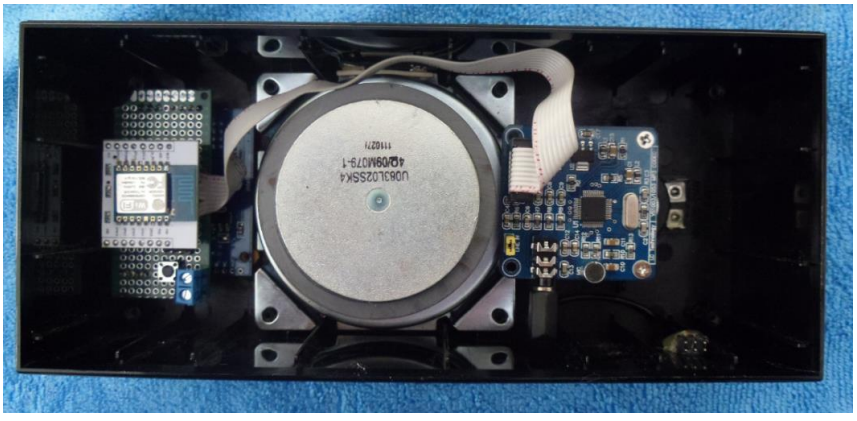


The radio is powered by a 5 V adapter. The radio will function on single LiPo cell as well, so I used a small charge circuit powered by the 5 V input. The amplifier uses a separate LiPo cell to minimize noise caused by the ESP8266. The TFT and VS1053 work on 3.8 to 5 Volt. For the ESP8266 a small regulator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

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

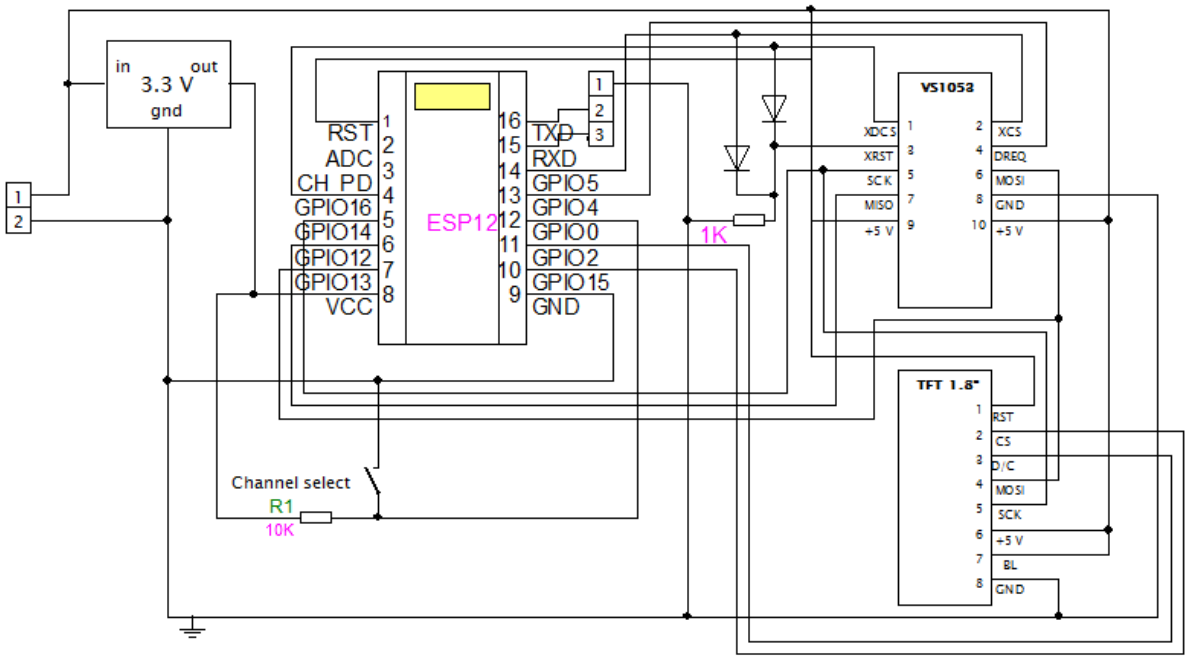


Most of the wiring is done on the green perfboard. The TFT is visible The VS1053 is connected by the grey cable:



The Class D stereo 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

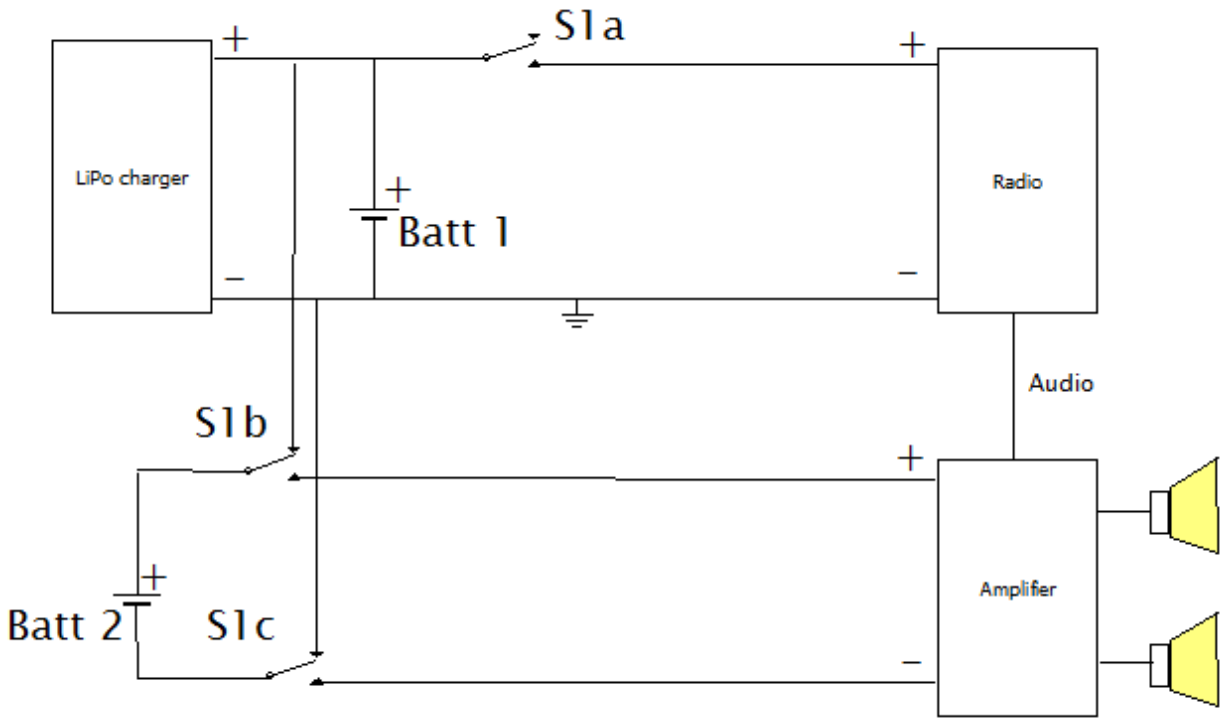
The reset circuit is a circuit with 2 diodes to GPIO5 and GPIO16 and a resistor to ground (wired OR gate) because there was not a free GPIO output available for this function.

*) GPIO2 and GPIO15 may be used for extra control buttons if no TFT is configured, see “3 button digital control”. The extra input buttons are not shown in the diagram.

Amplifier and power circuit.

The amplifier is a class D stereo amplifier. If the power is shared with the power supply of the radio, you will hear much noise. So I used a separate LiPo battery (Batt 2) for the amplifier.

During operation only Batt 1 will be charged. If the radio is switched off, both batteries will be recharged by the LiPo charger. S1a, S1b and S1c is a triple On-On switch.



Web interface:

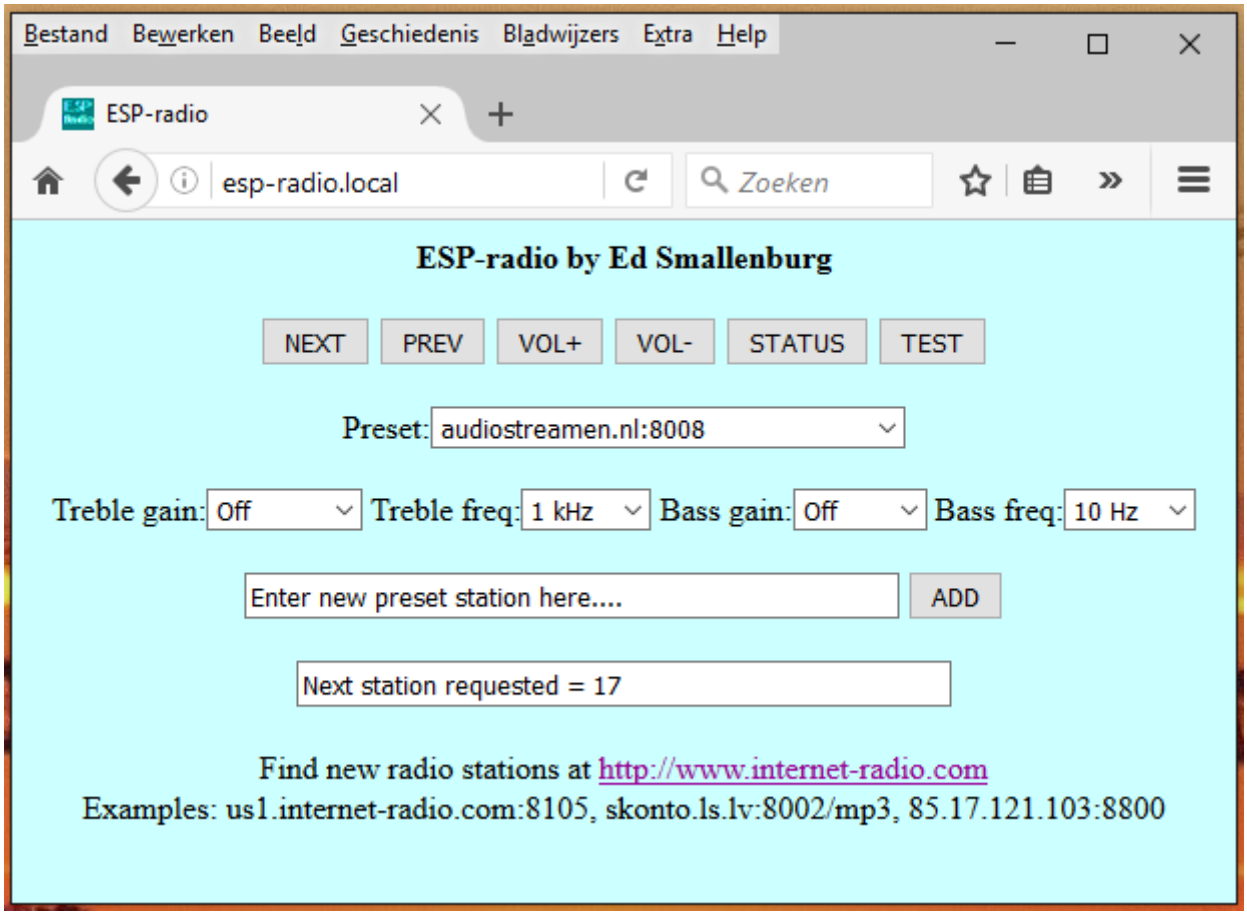
The web interface is not completed yet. The basic idea is to have a html page with embedded javascript that displays an interface to the radio. Command to the radio can be sent to the http server on the ESP8266. The IP address of the webserver will be displayed on the TFT during startup.

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.

If your computer is configurated for mDNS, you can also use http://esp-radio.local in your browser.

The following simple web interface will be displayed:



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`
- `toneHA     = 3              - Set treble gain`
- `toneHF     = 3              - Set treble limit frequency`
- `toneLA     = 3              - Set bass gain`
- `toneLF     = 3              - Set bass limit frequency`
- `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`
- `reset      = 0              - Will reset the ESP8266`
- `list       = 0              - Will list all presets in raw format`
- `analog     = 0              - Shows current analog input`

`Station may also be of the form` "skonto.ls.lv:8002/mp3".

Arduino IDE installation for Esp-radio.

- Download Windows installer for Arduino version 1.6.8 from https://www.arduino.cc/download_handler.php?f=/arduino-1.6.8-windows.exe and start the installation.
- Start Arduino and open Preferences window.
- Enter `http://arduino.esp8266.com/stable/package_esp8266com_index.json` into *Additional Board Manager URLs* field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install *esp8266* platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).
- Download "Async TCP Library for ESP8266 Arduino" from https://github.com/me-no-dev/ESPAsyncTCP and install in the IDE (add .zip library).
- Download "Async Web Server for ESP8266 Arduino" from https://github.com/me-no-dev/ESPAsyncWebServer and install in the IDE (add .zip library).
- Download "TFT_ILI9163C library" from https://github.com/sumotoy/TFT_ILI9163C and install in the IDE (add .zip library). There is a bug in this library: a part of the display is missing when the display is used in "landscape"-mode (mode "3"). See next paragraph for the patch.
- Install Adafruit GFX library in the IDE (library manager).
- Load the sketch. You should be able to compile it.
- Install Python 2.7 for Windows. Select option "Add python.exe to Path".
- Download the tool: https://github.com/esp8266/arduino-esp8266fs-plugin/releases/download/0.2.0/ESP8266FS-0.2.0.zip and unpack it in your Arduino sketchbook directory, create `tools` directory if it doesn't exist yet. The path will look like `<home_dir>/Arduino/tools/ESP8266FS/tool/esp8266fs.jar`.
- Restart Arduino IDE
- Update boards.txt according to https://github.com/esp8266/Arduino/blob/master/boards.txt

TFT library patch.

The display is used in mode "3". The TFT-library has a bug for this mode. Height and width are reversed here.

To correct it, find the source code TFT_ILI9163C.cpp on your computer (mine is at "documents/Arduino/libraries/TFT_ILI9163C-master") and change the 2 lines 1096 and 1097 to:

```
_width  = _TFTHEIGHT;//-__OFFSET;
_height = _TFTWIDTH;
```

Debug (serial 115600 Baud) output.

This is an example of the debug output.

```
D: FS Total 957314, used 10291
D: /ADSL-11-EXT.pw                  -     12
D: /ADSL-11.pw                      -     12
D: /favicon.ico                     -    766
D: /index.html                      -   6563
D: /NETGEAR-11.pw                   -     12
D: /SSID.pw                         -     73
D: Starting ESP Version 23-05-2016...  Free memory 13200
D: Sketch size 306872, free size 2838528
D: EEPROM is already filled. Available stations:
D: 00 - Stations from remote control
D: 01 - 109.206.96.34:8100
D: 02 - us1.internet-radio.com:8180
D: 03 - us2.internet-radio.com:8050
D: 04 - us1.internet-radio.com:15919
D: 05 - us2.internet-radio.com:8132
D: 06 - us1.internet-radio.com:8105
D: 07 - stream.radioreklama.bg:80/radio1.ogg
D: 08 - 205.164.62.15:10032
D: 09 - 109.206.96.11:80
D: 10 - 85.17.121.216:8468
D: 11 - 85.17.121.103:8800
D: 12 - 85.17.122.39:8530
D: 13 - 94.23.66.155:8106
D: 14 - 205.164.62.22:7012
D: 15 - skonto.ls.lv:8002/mp3
D: Restored settings: volume 76, preset 1
D: Reset VS1053...
D: End reset VS1053...
D: Slow SPI,Testing VS1053 read/write registers...
D: Fast SPI, Testing VS1053 read/write registers again...
D: endFillByte is 0
D: * Scan Networks *
D: Analog button 1 pushed, v = 12
D: Number of available networks: 8
D:  1 - Ziggo                    Signal: -86 dBm Encryption ????
D:  2 - Ziggo0A484               Signal: -85 dBm Encryption WPA2
D:  3 - NETGEAR-11               Signal: -64 dBm Encryption WPA2  Acceptable
D:  4 - Private_FON              Signal: -62 dBm Encryption WPA
D:  5 - FON_Roulet_11            Signal: -63 dBm Encryption None
D:  6 - ADSL-11                  Signal: -86 dBm Encryption Auto  Acceptable
D:  7 - ADSL-11-EXT              Signal: -93 dBm Encryption Auto  Acceptable
D:  8 - Wireless_MN2             Signal: -82 dBm Encryption Auto
D: ------------------------------------
D: Selected network: NETGEAR-11
D: Try WiFi NETGEAR-11
D: IP = 192.168.2.16
D: Start server for commands
D: Connect to new host
D: EEprom entry is 109.206.96.34:8100
D: Connect to preset 1, host 109.206.96.34 on port 8100, extension /
D: Connected to server
D: ICY 200 OK
D: icy-notice1:<BR>This stream requires <a href="http://www.winamp.com/">Winamp</a><BR>
D: icy-notice2:SHOUTcast Distributed Network Audio Server/Linux v1.9.8<BR>
D: icy-name:NAXI LOVE RADIO (NAXI,Belgrade,Serbia, NAXI,Beograd,Srbija) - 128k
D: icy-genre:Love and Romance
D: icy-url:http://www.naxi.rs
D: content-type:audio/mpeg
D: icy-pub:1
D: icy-metaint:8192
D: icy-br:128
D:
D: Switch to DATA
D: First chunk:
D: A1 A4 8E EA 68 48 F0 3E
D: 01 8F 14 C8 BE 8E 85 A8
D: 5A 38 54 50 33 A7 83 80
D: 28 92 52 78 1A DB 24 44
pm open,type:2 0
D: Metadata block 64 bytes
D: StreamTitle=' Lisa Stansfield - All Woman
```