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 this device as well as 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.
- Can connect to a standalone mp3-file on a server.
- mp3 playlists supported.
- Uses a minimal number of components.
- Has a preset list of a maximum of 100 favorite radio stations in a configuration file.
- Can be controlled by a tablet or other device through the built-in webserver.
- Can be controlled by MQTT commands.
- Can be controlled by commands on the serial input.
- Optional one or three button control to skip to the next/previous/first preset station.
- The strongest available WiFi network is automatically selected. Passwords are kept in a configuration file on 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).
- Parameters like WiFi SSID and password can be configured in a .ini file. The .ini file can be edited in the web interface.
- Plays iHeartRadio streams.

Software:

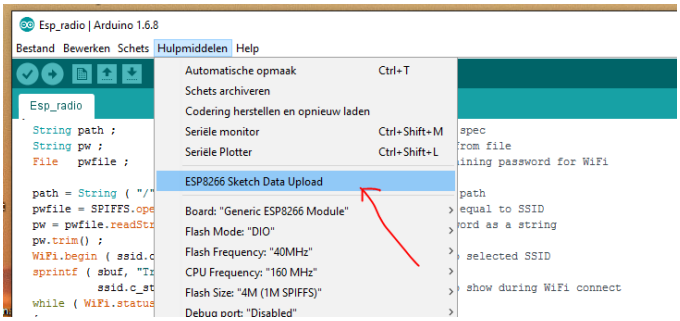
The software for the radio is supplied as an Arduino sketch that can be compiled for the ESP8266 using the Arduino IDE version 1.8, esp8266 software 2.2.0. No Arduino is required in this project.
The following libraries are used:

- 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)
- ESPAsyncWebServer – for remote controlling the radio via http.
- ESPAsyncTCP – Needed for webserver.
- ArduinoOTA for software update over WiFi.
- AsyncMqttClient to handle incoming MQTT messages.
- TinyXML for decoding the iHeartRadio xml.

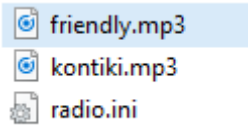
The map with the Esp-radio sketch must also contain the supplied headerfiles "index_html.h", "favicon_ico.h", "radio_css.h", "config_html.h" and "about_html.h". These files are included for the webinterface in PROGMEM.

Configuration:

In order to work properly, the software needs some configuration.
The filesystem (SPIFFS, set to 3 MB) of the ESP8266 must contain the files necessary for the webserver and a configuration file. 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.
At least the file "radio.ini" must be present in a map "data" in the Arduino project map, For Example:



The 2 extra files in the above figure can be played directly from SPIFFS. The webinterface allows upload of additional files or updates to existing files (although it seems pretty unreliable).

Files for the web interface are placed in PROGMEM, but you may include some extra webpages in the data directory.

radio.ini file

This initialization file contains lines that are used for initializations.
You may change the contents of the file with the “Config”-button in the web interface. An example of the ini-file is:

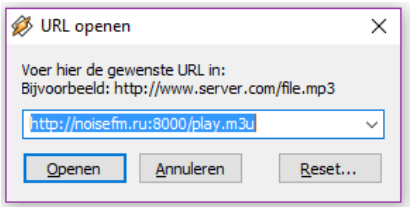
```
# radio.ini
# Initialization file for Esp-radio
#
# MQTT broker, credentials and topic  to subscribe
mqttbroker = broker.hivemq.com              # Broker to connect with
mqttport = 1883                             # Portnumber (1883 is default)
mqttuser = none                             # (No) username for broker
mqttpasswd = none                           # (No) password for broker
mqtttopic = espradio                        # Topic for receiving commands
mqttpubtopic = espradioIP                   # IP will be pubished here
#
# WiFi network credentials, may be more than on entry.  Strongest signal will be selected.
wifi_00 = NETGEAR-11/xxxxxxx
wifi_01 = ADSL-11/yyyyyyy
#
# VS1053 settings
volume = 72
toneha = 0
tonehf = 0
tonela = 0
tonelf = 0
#
# Presets
preset = 6                                                  # Start with preset 6
preset_00 = 109.206.96.34:8100                              #  0 - NAXI LOVE RADIO, Belgrade, Serbia
preset_01 = airspectrum.cdnstream1.com:8114/1648_128        #  1 - Easy Hits Florida 128k
preset_02 = us2.internet-radio.com:8050                     #  2 - CLASSIC ROCK MIA WWW.SHERADIO.COM
preset_03 = airspectrum.cdnstream1.com:8000/1261_192        #  3 - Magic Oldies Florida
preset_04 = airspectrum.cdnstream1.com:8008/1604_128        #  4 - Magic 60s Florida 60s Classic Rock
preset_05 = us1.internet-radio.com:8105                     #  5 - Classic Rock Florida - SHE Radio
preset_06 = icecast.omroep.nl:80/radio1-bb-mp3              #  6 - Radio 1, NL
preset_07 = 205.164.62.15:10032                             #  7 - 1.FM - GAIA, 64k
preset_08 = skonto.ls.lv:8002/mp3                           #  8 - Skonto 128k
preset_09 = 94.23.66.155:8106                               #  9 - *ILR CHILL and GROOVE
preset_10 = ihr/IHR_IEDM                                    # 10 - iHeartRadio IHR_IEDM
preset_11 = ihr/IHR_TRAN                                    # 11 - iHeartRadio IHR_TRAN
# End of file
```

Lines starting with “#” are ignored. Comments per line (after “#”) will also be ignored, except for the “preset_” lines. The comments on the “preset_” lines are used to identify the presets in the web interface. Note that the numbers has to be consecutive ranging from 00 to 99. If the highest numbered station is reached, the next station will be 00 again. URLs with mp3 files or mp3 playlists (.m3u) are allowed.
Presets starting with "ihr/" are iHeartRadio stations.

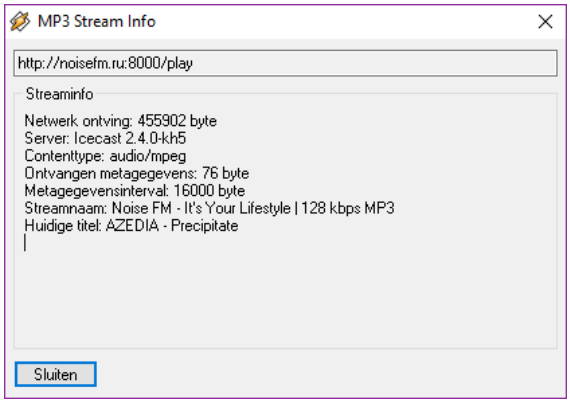
The .ini-file itself can be edited in the webinterface. Changes will be effective after restart of the Esp-radio. Note that there might be problems with long ini-files due to write problems in SPIFFS.

Using Winamp to find out the correct preset line for a station.

Start Winamp, press Ctrl-L in the main window and enter the URL in the edit box:



Open the URL. Then press Alt-3 in the main window (left picture). You will see info for the playing station (right picture).



The top line (with “http”) will contain the information for the preset, in this example:

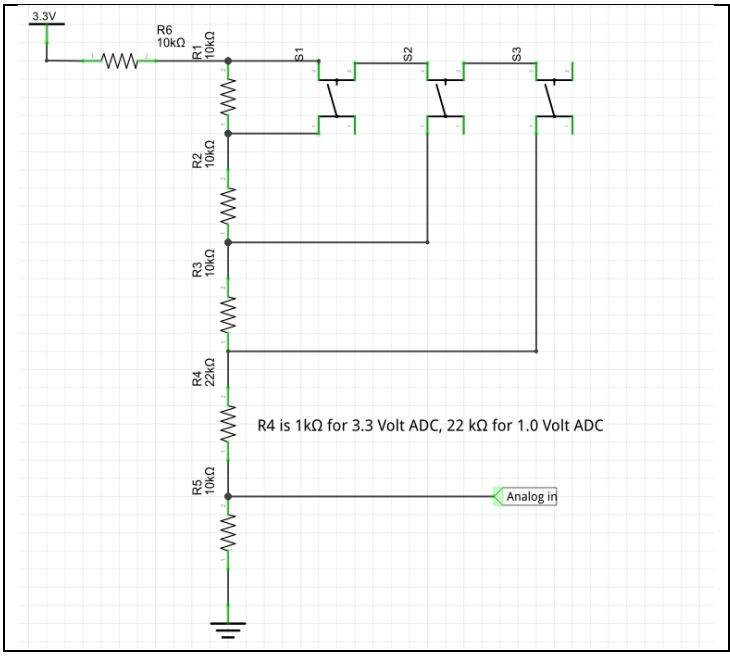
“noisefm.ru:8000/play”. The complete line for this station in the .ini-file would be:

```
preset_05 = noisefm.ru:8000/play            #  5 - Noise FM
```

Optional:

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). Some use 3.3 Volt as maximum input, other use 1.0 Volt. Therefore the value of R4 may be 1kΩ (for 3.3 V ADC) or 22 kΩ (for 1 V ADC). The definitions of asw1, asw2 and asw3 have to be corrected in the source. In the web interface you may use ”http:// <IP>/?analog” to read the actual voltage (ADC units) while pressing one of the 3 buttons.
You may also use the command "analog" in the serial interface.

3 button digital control:

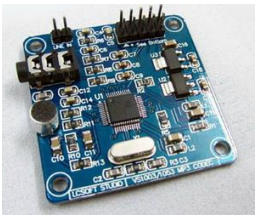
Normally the radio is controlled by the web interface. However, one digital input (GPIO0) may be connected to a button to skip to the next preset station.
If TFT output is not required, 2 extra digital inputs are available. You may use the 2 extra inputs for 2 extra control buttons (GPIO2 and GPIO15): “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:

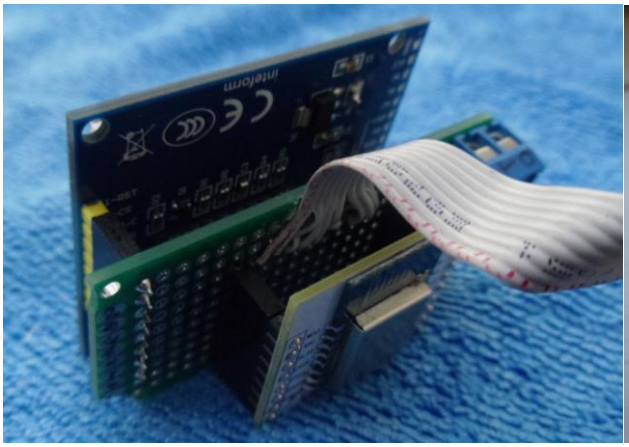
The radio is built with the following hardware:

- 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 VS1053 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.

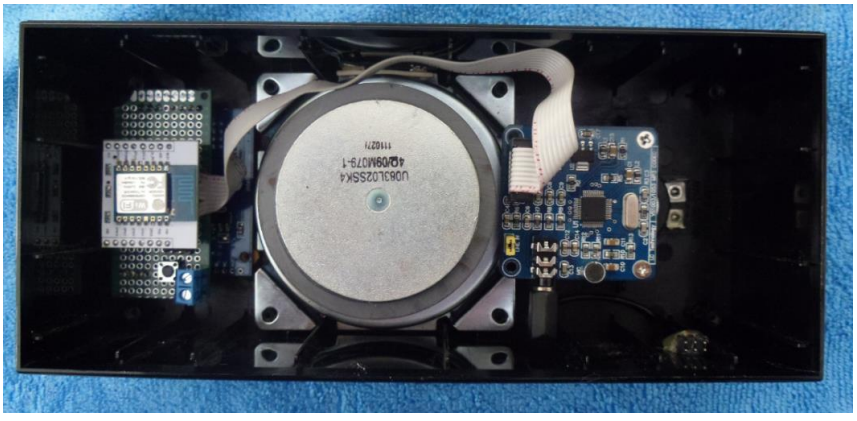


The radio is powered by a 5 V adapter. The radio will function on single LiPo cell as well, so I used a small charge circuit powered by the 5 V input. The amplifier uses a separate LiPo cell to minimize noise caused by the ESP8266. The TFT and VS1053 work on 3.8 to 5 Volt. For the ESP8266 a small regulator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

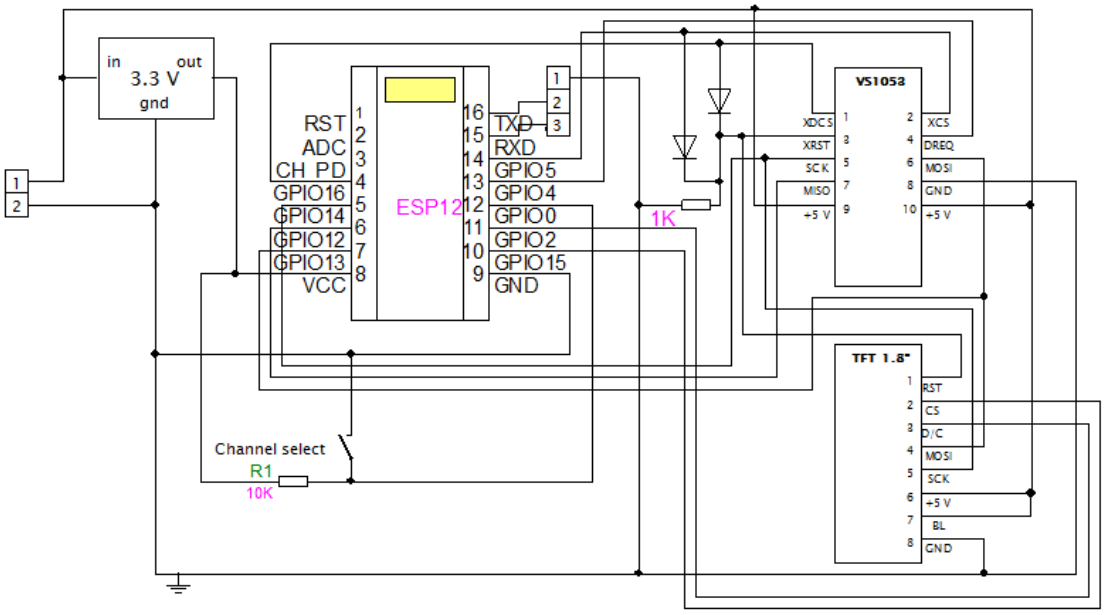


Most of the wiring is done on the green perfboard. The TFT is visible The VS1053 is connected by the grey cable:



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 7 (BL)	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.

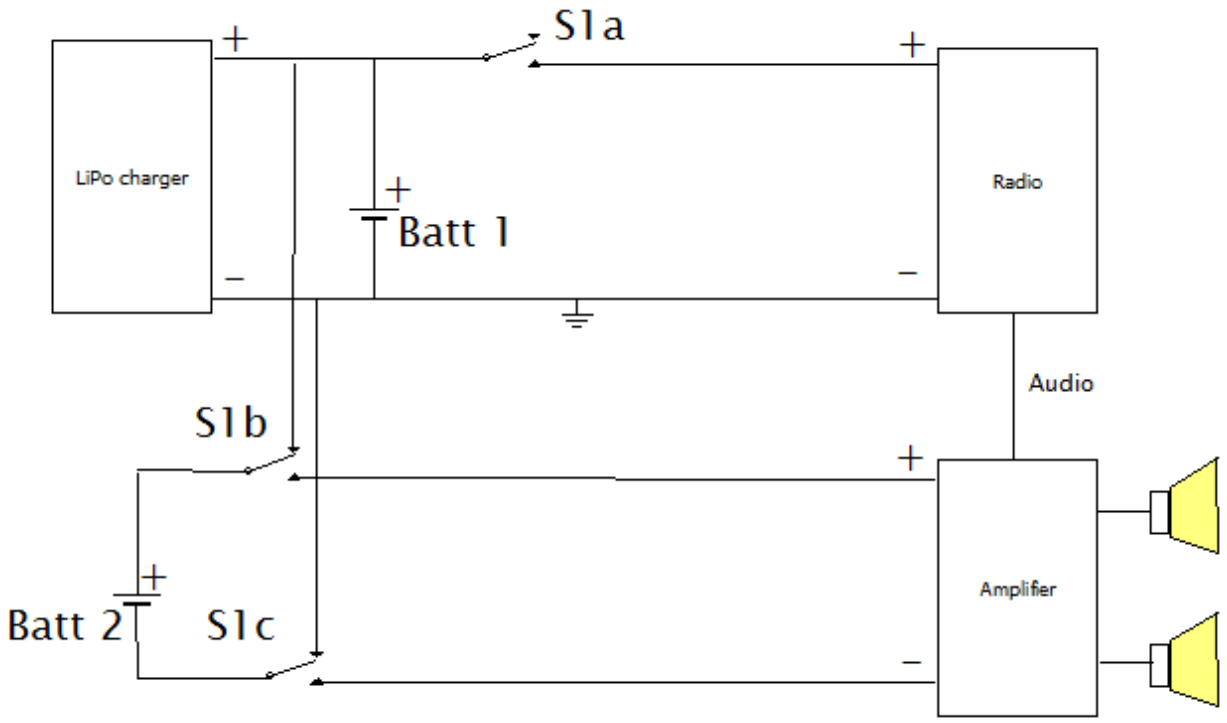
*) GPIO2 and GPIO15 may be used for extra control buttons if no TFT is configured, see “3 button digital control”. The extra input buttons are not shown in the diagram.

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.

Note that there may be high currents in the “off”-position if Batt 1 is fully discharged. Use protected batteries only!



Web interface:

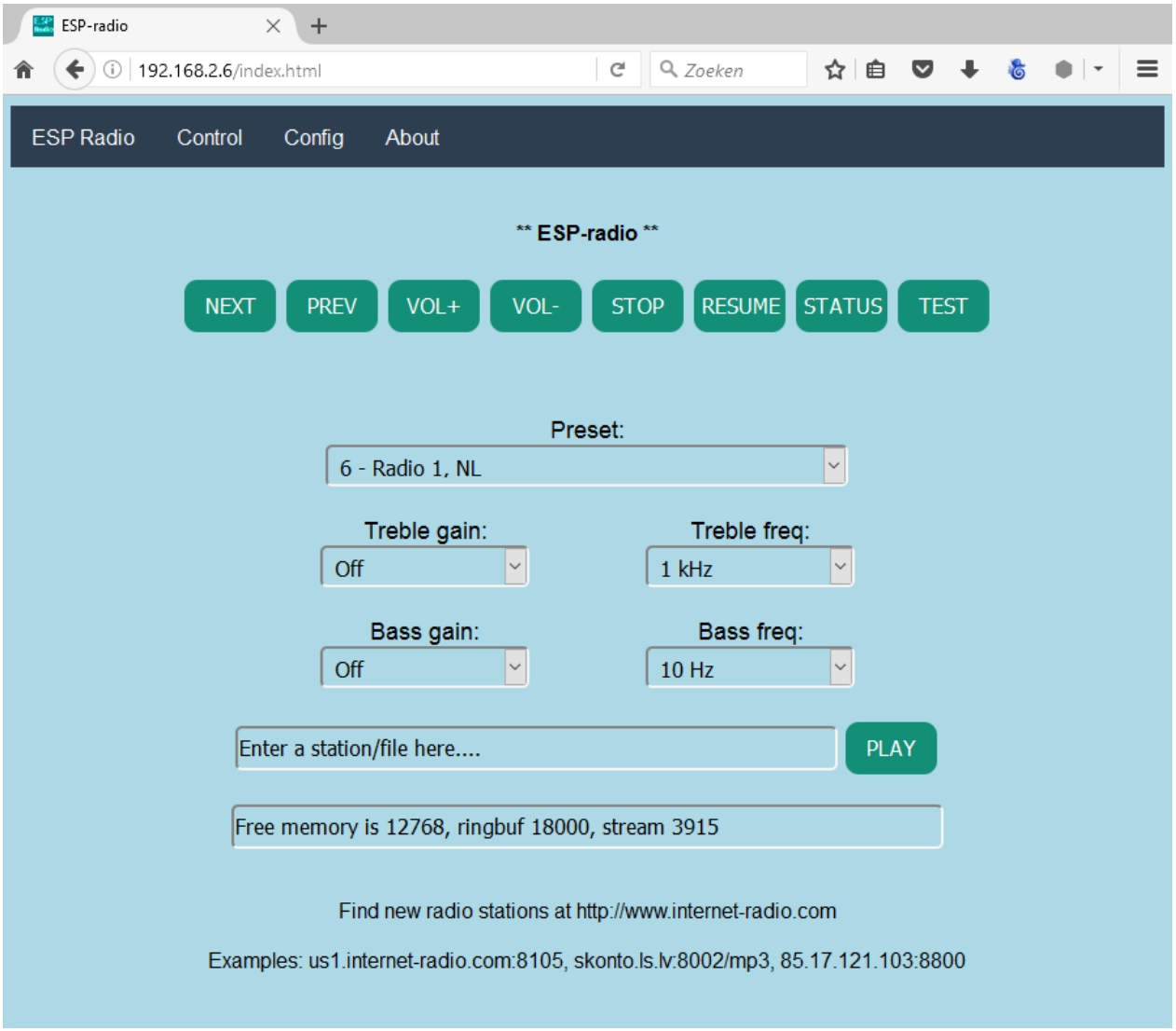
The web interface is simple and can be adapted to your needs. The basic idea is to have a html page with embedded javascript that displays an interface to the radio. Command to the radio can be sent to the http server on the ESP8266. The IP address of the webserver will be displayed on the TFT during startup. The webpages are defined in PROGMEM.

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 configured 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. Commands may also come from MQTT or serial input. Not all commands are meaningful on MQTT or serial input. Working commands are:

```
preset     = 12                        Select start preset to connect to
uppreset   = 1                         Select next preset or playlist entry
downpreset = 1                         Select previous preset or playlist entry
preset_00  = <mp3 stream>              Specify station for a preset 00-99 *)
volume     = 95                        Percentage between 0 and 100
upvolume   = 2                         Add percentage to current volume
downvolume = 2                         Subtract percentage from current volume
toneha     = <0..15>                   Setting treble gain
tonehf     = <0..15>                   Setting treble frequency
tonela     = <0..15>                   Setting bass gain
tonelf     = <0..15>                   Setting treble frequency
station    = <mp3 stream>              Select new station (will not be saved)
station    = <URL>.mp3                 Play standalone .mp3 file (not saved)
station    = <URL>.m3u                 Select playlist (will not be saved)
xml        = <Mountpoint>              Select iHeartRadio station (not saved)
mute                                   Mute the music
unmute                                 Unmute the music
stop                                   Stop player
resume                                 Resume player
wifi_00    = mySSID/mypassword         Set WiFi SSID and password *)
mqttbroker = mybroker.com              Set MQTT broker to use *)
mqttport   = 1883                      Set MQTT port (default 1883) to use *)
mqttuser   = myuser                    Set MQTT user for authentication *)
mqttpasswd = mypassword                Set MQTT password for authentication*)
mqtttopic  = mytopic                   Set MQTT topic to subscribe to *)
mqttpubtopic  = mypubtopic             Set MQTT topic to publish to *)
status                                 Show current URL to play
test                                   For test purposes
debug      = 0 or 1                    Switch debugging on or off
reset                                  Restart the ESP8266
analog                                 Show current analog input
```

Commands marked with "*)" are sensible in ini-file only.

Station may also be of the form “skonto.ls.lv:8002/mp3”. The default port is 80.
Station may also point to an mp3 playlist. Example: “www.rockantenne.de/webradio/rockantenne.m3u”.
Station may be an .mp3-file on a remote server. Example: "www.stephaniequinn.com/Music/Rondeau.mp3".
Station may also point to a local .mp3-file on SPIFFS. Example: “localhost/friendly.mp3”. Note that there is only limited space to store .mp3-files on SPIFFS.

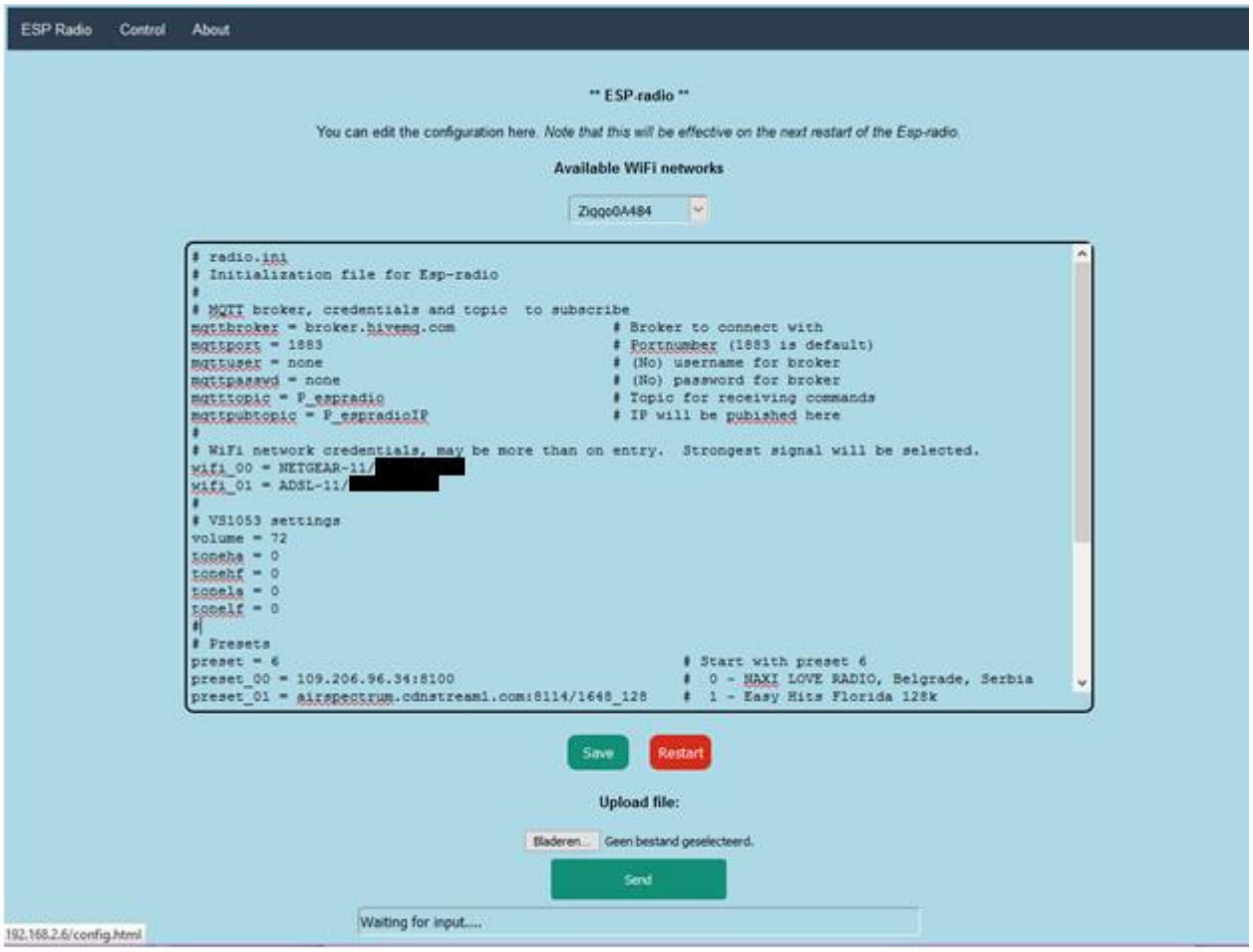
It is allowed to have multiple (max 100) “preset_” lines. The number after the “_” will be used as the preset number. The comment part (after the “#”) will be shown in the webinterface.

It is also allowed to have multiple “wifi_” lines. The strongest Wifi accesspoint will be used.

Configuration

The Config button will bring up a second screen . Here you can edit the ini-file or upload an arbitrary file to the SPIFFS. The available Wifi networks are listed as well. The config screen will be shown automatically if the ESP8266 cannot connect to one of the WiFi stations specified in the ini-file.
In that case the ESP8266 will act as an accesspoint with the name “Esp-radio”. You have to connect to this AP with password “Esp-radio”. Than the ESP-radio can be reached at http:// 192.168.4.1.

After changing the contents of the ini-file, it must be saved to the SPIFFS by clicking the “Save” button. Changes will have effect on the next restart of the ESP-radio, so click the “Restart” button.



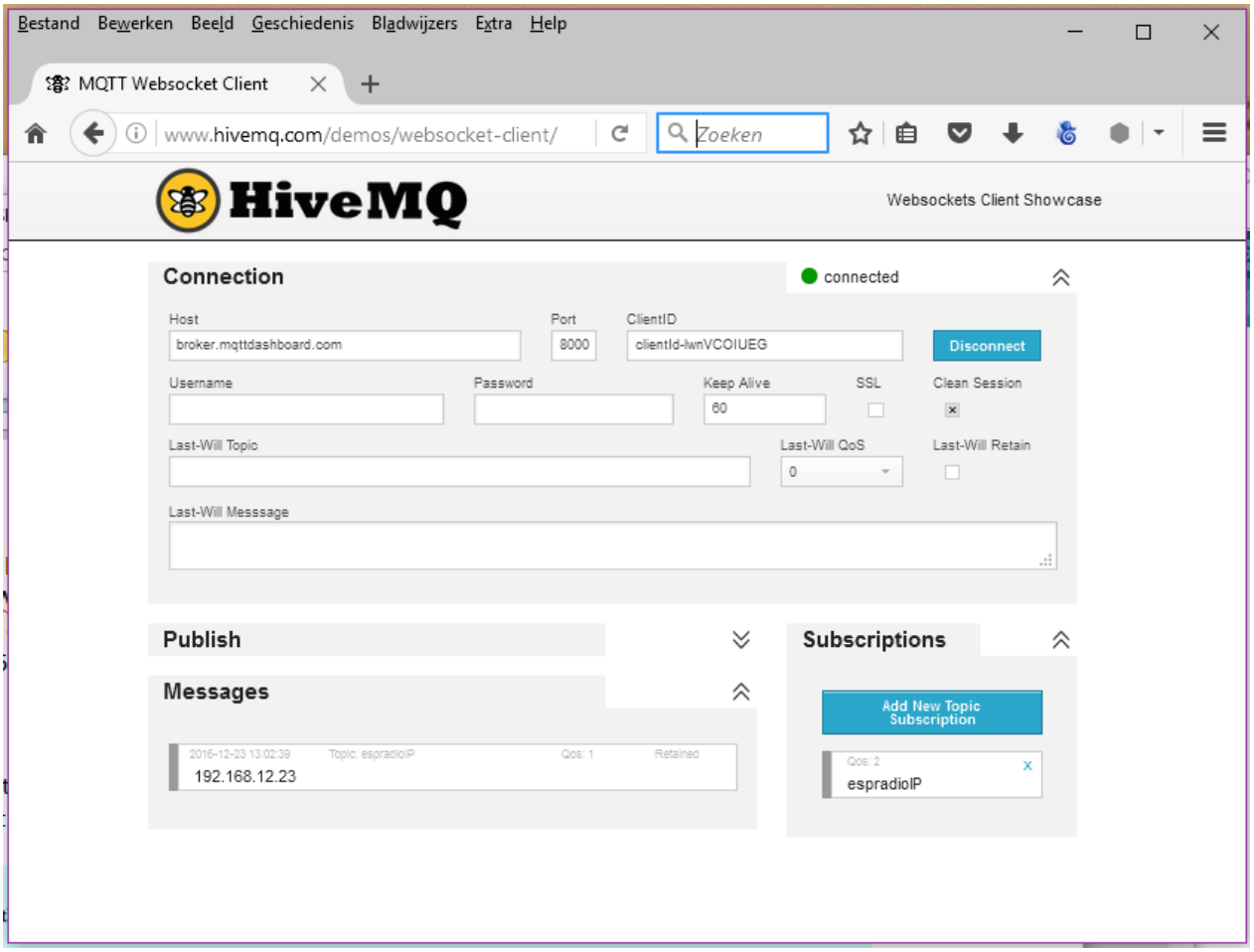
MQTT interface.

The MQTT interface can handle the same commands as the web interface.
As publish command on a Linux system may look like:

```
$ mosquitto_pub -h broker.hivemq.com -t espradio -m volume=80
```

Note that the broker in this example is heavily used, this may cause some delay. If you use your own broker the reaction on commends will be much better.
Remove the lines starting with “mqtt” if no MQTT is required.

You can use an MQTT online client like http://www.hivemq.com/demos/websocket-client/



Subscribe to “espradioIP” and you will see the IP-address of your radio. You will see the IP address of your radio or the IP address of a different user. So be sure to use unique names for your topics.
The parameters in radio.ini for this example are:

```
# radio.ini
# Initialization file for Esp-radio
#
# MQTT broker, credentials and topic  to subscribe
mqttbroker = broker.hivemq.com          # Broker to connect with
mqttport = 1883                         # Portnumber (1883 is default)
mqttuser = none                         # (No) username for broker
mqttpasswd = none                       # (No) password for broker
mqtttopic = espradio                    # Topic for receiving commands
mqttpubtopic = espradioIP               # IP will be published here
#
```

In this example I published the command “uppreset=1” to the radio. The radio published the IP-address 192.168.12.23 to the broker (once every 10 minutes).

Arduino IDE installation for Esp-radio on Windows.

- Download Windows installer for Arduino version 1.8 from http://www.arduino.org/downloads and start the installation.
- Start Arduino IDE 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.
- Download library for "AsyncMqttClient" version 0.5.0 from https://github.com/marvinroger/async-mqtt-client and install in the IDE.
- Install Adafruit GFX library in the IDE (library manager).
- Download TinyXML library from https://github.com/adafruit/TinyXML and install it in the IDE.
- 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 patches.

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;
```

Find TFT_ILI9163C_settings.h and edit it so that the right board will be selected. As an example you will find the configuration for the "blue 1.8 SPI 128x160 board" in the TFT_ILI9163C-master.zip file.

Debug (serial 115600 Baud) output.

This is an example of the debug output.

```
D: FS Total 2949250, used 2145297
D: /radio.ini                       -    1780
D: /friendly.mp3                    -  281941
D: /kontiki.mp3                     - 1843200
D: Added SSID 00 = NETGEAR-11 to acceptable networks
D: Added SSID 01 = ADSL-11 to acceptable networks
D: * Scan Networks *
D: Number of available networks: 8
D:  1 - HZN240825082              Signal: -73 dBm Encryption WPA2
D:  2 - Private_FON               Signal: -56 dBm Encryption WPA
D:  3 - FON_Roulet_11             Signal: -56 dBm Encryption None
D:  4 - ADSL-11                   Signal: -86 dBm Encryption Auto  Acceptable
D:  5 - Roulet 9 Gast             Signal: -88 dBm Encryption Auto
D:  6 - Roulet 9                  Signal: -87 dBm Encryption Auto
D:  7 - NETGEAR-11                Signal: -58 dBm Encryption WPA2  Acceptable
D:  8 - TWEED                     Signal: -86 dBm Encryption WPA2
D: ------------------------------------
D: Command: wifi_00 with parameter NETGEAR-11/XXXXXXXX
D: Command: wifi_01 with parameter ADSL-11/YYYYYYY
D: Command: volume with parameter 72
D: Command: toneha with parameter 0
D: Command: tonehf with parameter 0
D: Command: tonela with parameter 0
D: Command: tonelf with parameter 0
D: Command: preset with parameter 6
D: Command: preset_00 with parameter 109.206.96.34:8100
D: Command: preset_01 with parameter airspectrum.cdnstream1.com:8114/1648_128
D: Command: preset_02 with parameter us2.internet-radio.com:8050
D: Command: preset_03 with parameter airspectrum.cdnstream1.com:8000/1261_192
D: Command: preset_04 with parameter airspectrum.cdnstream1.com:8008/1604_128
D: Command: preset_05 with parameter us1.internet-radio.com:8105
D: Command: preset_06 with parameter icecast.omroep.nl:80/radio1-bb-mp3
D: Command: preset_07 with parameter 205.164.62.15:10032
D: Command: preset_08 with parameter skonto.ls.lv:8002/mp3
D: Command: preset_09 with parameter 94.23.66.155:8106
D: Command: preset_10 with parameter ihr/IHR_IEDM
D: Command: preset_11 with parameter ihr/IHR_TRAN
D: Command: preset_12 with parameter ihr/WQTR
D: Command: preset_13 with parameter ihr/CKMXAM
D: Starting ESP Version Wed, 04 May 2017 10:00:00 GMT...  Free memory 12448
D: Sketch size 349760, free size 696320
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: Selected network: NETGEAR-11
D: Try WiFi NETGEAR-11
D: IP = 192.168.2.9
D: Start server for commands
D: STOP requested
D: Song stopped correctly after 0 msec
D: New preset/file requested (-1/0) from icecast.omroep.nl:80/radio1-bb-mp3
D: Connect to new host icecast.omroep.nl:80/radio1-bb-mp3
D: Connect to icecast.omroep.nl on port 80, extension /radio1-bb-mp3
D: Connected to server
D: Server: Icecast 2.4.2
D: Content-Type: audio/mpeg
D: Cache-Control: no-cache
D: Expires: Mon, 26 Jul 1997 05:00:00 GMT
D: Pragma: no-cache
D: icy-br:192
D: ice-audio-info: bitrate=192
D: icy-br:192
D: icy-genre:Talk
D: icy-name:Radio 1
D: icy-pub:0
D: icy-url:http://www.radio1.nl
D: icy-metaint:16000
D: Switch to DATA, bitrate is 192
D: First chunk:
D: CB 72 91 06 24 03 FA EF
D: BB AF 1B 74 53 6F 86 9E
D: A6 95 48 DB 1B C1 0A 23
D: 88 08 93 30 C4 80 E4 2A
D: Metadata block 64 bytes
D: Streamtitle found, 50 bytes
D: StreamTitle='NPO Radio 1 - De Ochtend - KRO-NCRV';
```