ESP32-radio

This document describes the realization of an Internet radio based on an ESP32 WiFi module.
The ESP32 is the successor of the ESP8266. For the ESP8266 an Internet radio was build and now the software is ported to the ESP32. The Internet radio described here uses the ESP32 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. This display has also a SD card slot. Mp3 tracks on an SD card can be played by the ESP32-radio. Other types of display are also supported.

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.
- Can play MP3 files from SD card.
- Uses a minimal number of components.
- Has a preset list of a maximum of 100 favorite radio stations as preferences in flash.
- Can be controlled by a tablet or other device through the built-in webserver.
- Can be controlled by MQTT commands. Status info is published by using MQTT.
- Can be controlled by commands on the serial input.
- Can be controlled by IR.
- Can be controlled by rotary switch encoder.
- Up to 14 input ports can be configured as control buttons like volume, skip to the next/previous/first/favorite preset station.
- I/O pins for TFT, VS1053, IR and rotary encoder can be configured in preferences.
- The strongest available WiFi network is automatically selected. Passwords are kept as preferences in flash. Heavily commented source code, easy to add extra functionality.
- Debug information through serial output.
- 2nd processor is used for smooth playback.
- Update of software over WiFi (OTA).
- Parameters like WiFi information and presets can be edited in the web interface.
- Plays iHeartRadio streams.
- Preset, volume, bass and treble settings are saved in preferences.
- Displays time of day on TFT.
- Optional displays remaining battery charge.
- Bitrates up to 320 kbps.

Software:

The software for the radio is supplied as an Arduino sketch that can be compiled for the ESP32 using the Arduino IDE version 1.8.5. No Arduino is required in this project.
The following libraries are used:

- WiFi – for establishing the communication with WiFI, part of ESP32 core library
- SPI – for communication with VS1053 and TFT display, part of ESP32 core library
- WiFiMulti - to select the strongest WiFi network
- Adafruit_ST7735– driver for the TFT screen (if configurated).
  This driver works also for the red ST7735 display.
- ArduinoOTA for software update over WiFi.
- PubSubClient to handle MQTT messages, see:
  https://github.com/knolleary/pubsubclient
- SD and FS for reading from SD card.

The map with the ESP32-radio sketch must also contain the supplied headerfiles “index_html.h”, “favicon_ico.h”, “radio_css.h”, “config_html.h”, "defaultprefs.h", "mp3play_html.h" and “about_html.h”. These files are included for the webinterface in PROGMEM.
Furthermore the files “Dummytft.h”, “LCD1602”, “SSD1306.h”, “bluetft.h” must be included in this map for the different LCD/OLED displays.

Preferences

Preferences for ESP32-Radio are kept in Flash memory (NVS).
You may change the contents of NVS with the “Config”-button in the web interface. An example of the contents is:

```
bat0   = 2178   # ADC for 0% battery capacity left
bat100 = 3256   # ACD for 100% battery capacity
#
clk_dst = 1
clk_offset = -1
clk_server = pool.ntp.org
#
gpio_00 = uppreset = 1
gpio_12 = upvolume = 2
gpio_13 = downvolume = 2
#
ir_40BF = upvolume = 2
ir_C03F = downvolume = 2
#
mqttbroker = mqtt.smallenburg.nl
mqttpasswd = *******
mqttport = 1883
mqttuser = edzelf
#
pin_enc_clk = 25     # GPIO rotary encoder CLK
pin_enc_dt = 26      # GPIO rotary encoder DT
pin_enc_sw = 27      # GPIO rotary encoder SW
pin_ir = 35          # GPIO IR receiver VS1838B
pin_sd_cs = 21       # GPIO SD card CS
pin_tft_cs = 15      # GPIO TFT CS
pin_tft_dc = 2       # GPIO TFT DC
pin_vs_cs = 5        # GPIO VS1053 CS
pin_vs_dcs = 16      # GPIO VS1053 DCS
pin_vs_dreq = 4      # GPIO VS1053 DREQ
#
preset = 4
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 MIAMI 256k
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
preset_12 = www.classicfm.nl/player/classicfm.m3u     # 12 - Playlist Classic FM
#
toneha = 0
tonehf = 0
tonela = 0
tonelf = 0
#
volume = 72
#
wifi_00 = ADSL-11_plus/*******
wifi_01 = NETGEAR-11/*******
wifi_02 = ADSL-11/*******
```

Not that the size of the NVS is limited to 20 kB.
Lines starting with “#” are comment lines and 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.
The maximum line length is 150 characters.
Note that the preset numbers ranges 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.

"pin_ir", " pin_enc_clk", " pin_enc_dt", " pin_enc_sw", " pin_tft_cs", " pin_tft_dc", " pin_sd_cs", " pin_vs_cs", "pin_vs_dcs" en " pin_vs_dreq" specify the GPIO pins that are wired to the various devices. For a FEATERBOARD you have to change some settings, especially for TFT and SD card. The pins for SPI default to SCK=18, MISO=19 and MOSI=23, but they may be configured by “pin_spi_sck”, “pin_spi_miso” and “pin_spi_mosi” respectively.

For an I2C display (OLED) there must be definitions for " pin_tft_scl" and " pin_tft_sda".
Remove the lines if a device is not connected.

“pin_shutdown” is the pin that will be set if the radio is not playing. This output-pin can be used to shut down the amplifier.

The "gpio_" lines specify input pins and the command that is executed if the input pin goes from HIGH to LOW.

The "touch_" lines specify the GPIO input pins for this signal and the command that is executed if the input pin is activated.

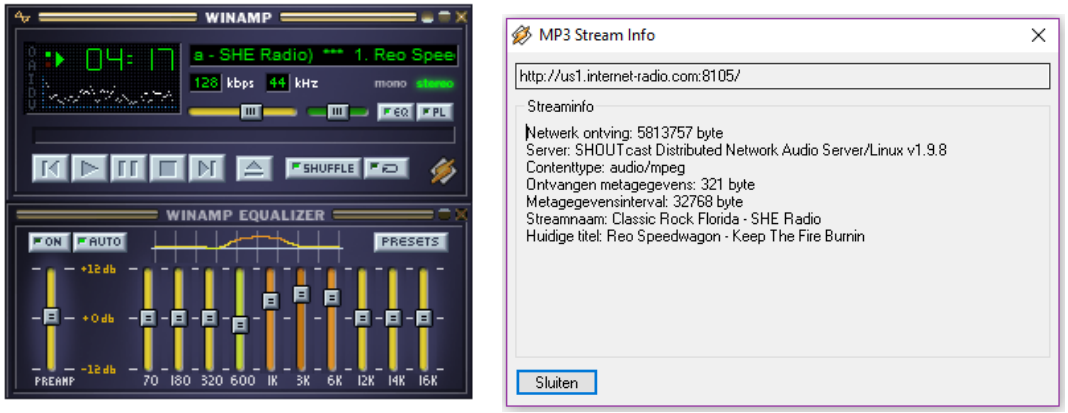
The "ir_XXXX" lines specify IR-codes and the command that is executed if the IR-code is received. The code part is a hexadecimal number in upper-case characters.

The "clk_" lines are used for the display of the current time on the TFT. Please change the values for your timezone. The values in the example are valid for the Netherlands.

The “bat0” and “bat100” lines are for displaying the remaining battery capacity on the display. See the details at the paragraph with optional features.

The preferences can be edited in the web interface. Changes will in some cases be effective after restart of the Esp-radio. If the list of preferences is empty (first start), you may use the default button. The list will show some default values which can be edited.

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



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:

"us1.internet-radio.com:8105". The complete line for this station in the preferences would be:

```
preset_05 = us1.internet-radio.com:8105            #  5 - Classic Rock Florida - SHE Radio
```

Optional features:

Digital control through input pins:

Normally the radio is controlled by the web interface. However, free digital inputs (GPIO) may be connected to buttons to control the radio. Their function can be programmed using the webinterface.

You can assign commands to the digital inputs by adding lines in the configuration (Webinterface, "Config" page). Examples:

```
gpio_00 = uppreset = 1
gpio_12 = upvolume = 2
gpio_13 = downvolume = 2
gpio_14 = stop
gpio_17 = resume
gpio_21 = station = icecast.omroep.nl:80/radio1-bb-mp3
```

In this example the ESP32-Radio will execute the command "uppreset=1" if GPIO0 will go from HIGH to LOW. The commands are equal to the commands that are handled by the serial input or by the MQTT interface.

The same format can be used for the touch-inputs:

```
touch_04 = uppreset = 1
touch_13 = upvolume = 2
```

In this example the ESP32-Radio will execute the command "uppreset=1" if TOUCH0 (GPIO04) will be activated. The commands are equal to the commands that are handled by the serial input or by the MQTT interface.

IR Interface.

The radio can be controlled by an IR remote control like this:

To use this interface, the "out" pin of a VS1838B receiver must be connected to a GPIO pin of the ESP32:

Add the assigned GPIO pin to the preferences through the config page in the web interface. Example:

```
pin_ir = 35                    # GPIO Pin number for IR receiver VS1838B
```

VCC is connected to 3.3 Volt. A 220 µF capacitor should be connected between VCC and GND.
The software will read the raw code of the IR transmitter, making it possible to use virtually any remote control to be used. I tested it with the 21 button remote as well as with an LG TV remote.
To assign functions to the buttons, watch the debug log output while pressing a button. For example, press the +volume button. You will see something like:

```
D: IR code 807F received, but not found in preferences!
```

Now add the command:

```
ir_807F = upvolume = 2
```

to the preferences in the config page of the web interface. Likewise you can assign functions to all buttons, for example:

```
ir_8A31 = uppreset = 1
ir_719A = station = us1.internet-radio.com:8105
ir_1F6B = mute
```

Display remaining battery capacity:

The remaining battery capacity is computed by measuring the battery voltage on ping GPIO36 (ADC0).
The battery voltage must be supplied to ADC0 through a resistor voltage divider. The maximum voltage fort his

input is 1.0 volt. I used 100 kΩ and 22 kΩ fort his purpose.
Furthermore you need to specify ADC value for both the full and the empty voltages. These values must be in the preferences like:

```
bat0   = 2178   # ADC for 0% battery capacity left
bat100 = 2690   # ACD for 100% battery capacity
```

To calibrate the settings you may use the “TEST” command in the serial monitor. This will display the current reading of the ADC0 pin. Do this for a fully charge battery and for an (almost) empty one.
Note that it will take about 6 seconds before the reading is stable. This is a result of a filter in the measurement of the ADC.

The remaining capacity is displayed as a green/red bar near the top of the display.

Hardware:

The radio is built with the following widely available hardware:

- An ESP-32 module. This is basically an ESP32 on a small print. I used a DOIT ESP32 Development Board. See figure 1 below. The ESP32 is running on 160 MHz. On Aliexpress: this.
- A VS1053 module. See figure 2.
- A 1.8 inch color TFT display. Optional, see figure 3 and 4. On Aliexpress: this or this.
- An OLED 128x64 display. Optional, see figure 7. On Aliexpress: this.
- An 1602 LCD display with I2C backpack. Optional, see figure 8.
- Two small speakers.
- A Class D stereo amplifier to drive the speakers. Best quality if powered by a separate power source.
- A rotary encoder switch. Optional, see figure 4.
- A IR receiver. Optional, see figure 6.

Fig. 1

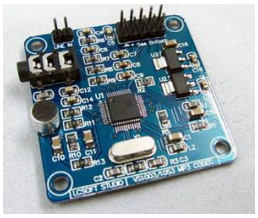


Fig. 2

Fig. 3

Fig. 4

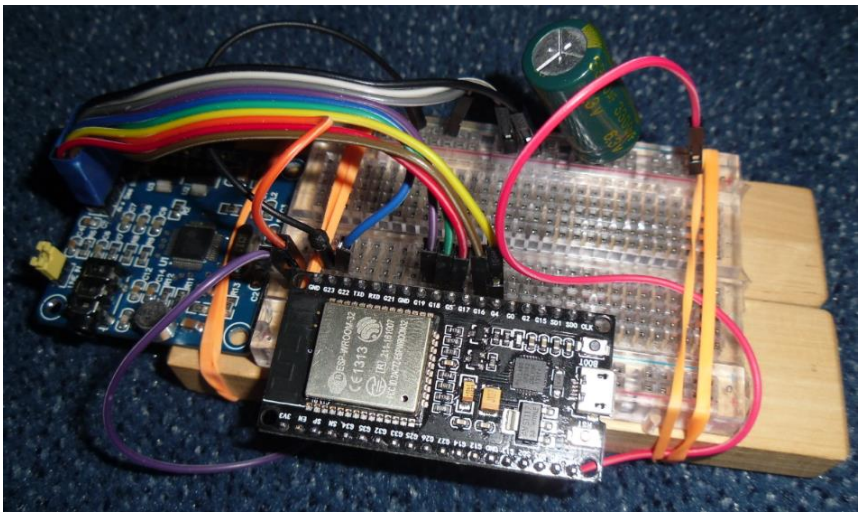
Fig. 5

Fig. 6

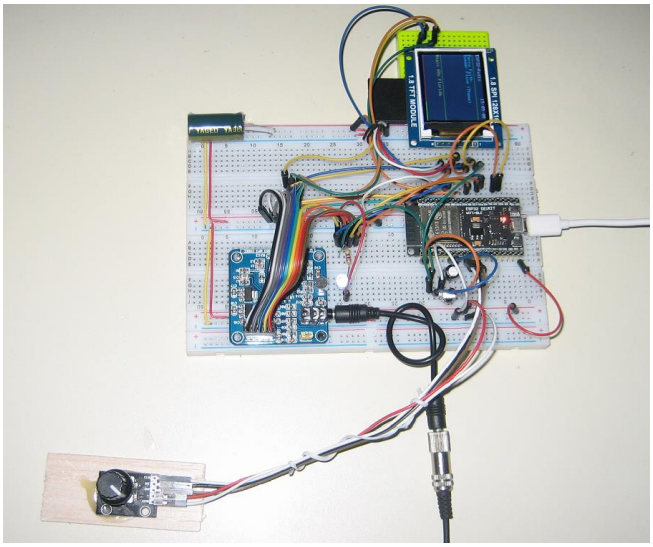
Fig. 7

Fig. 8

Here is a picture of the radio in a test configuration. Only the VS1053 is connected.



The module is not very breadboard friendly. The (hanging over) +5 Volt of the ESP32 is wired to the 5Volt rail of the breadboard. The big capacitor (3300 µF) is added to allow powering the module from USB. Without it, the USB has insufficient power to drive the ESP32.



Another prototype. This time with SD-card, TFT, IR and rotary switch.

The radio may be powered by a 5 V adapter. The radio will function on single LiPo cell as well, so I added a small charge circuit powered by the 5 V input. The amplifier uses a separate LiPo cell to minimize noise caused by the ESP32. The TFT and VS1053 work on 3.8 to 5 Volt.
I used a small perforated board to connect the ESP32 module 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:

Wiring:

The logic wiring in the table below. The analog amplifier and the speakers are not included.

Note that the GPIO pins are just an example. The assignment of GPIO pins is defined in the preferences that can be edited in the "config" page of the web interface.

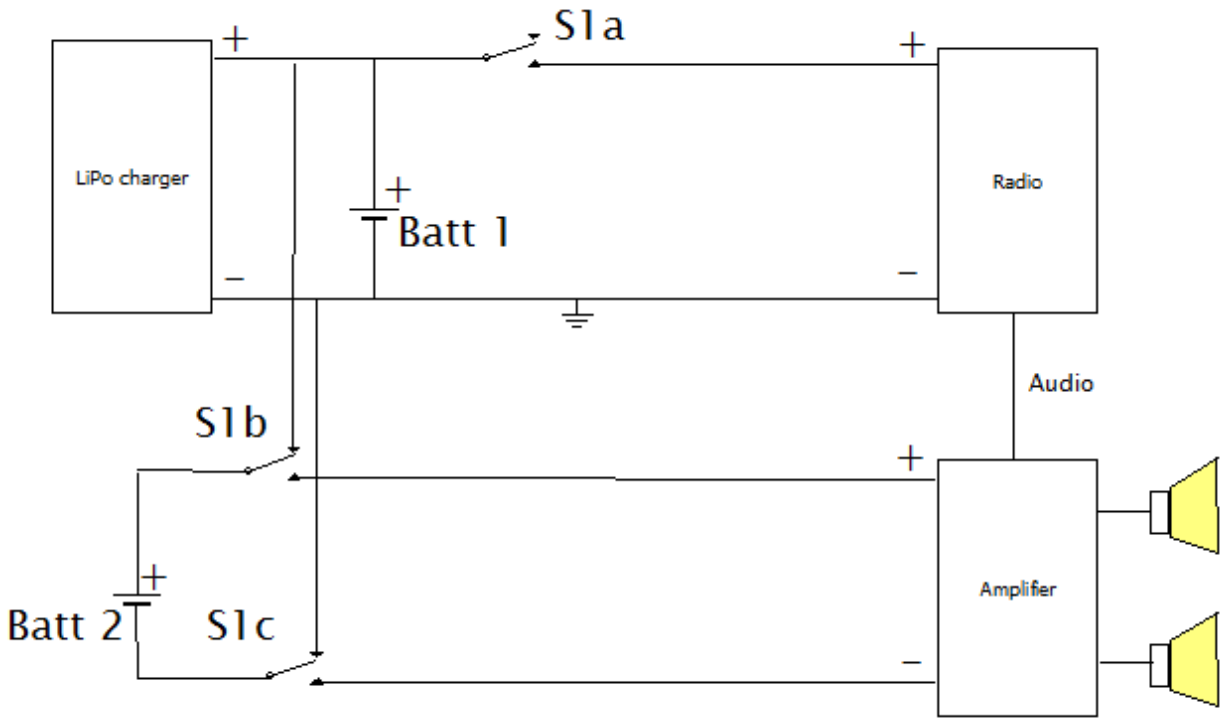
ESP32dev	Signal	Wired to LCD	Wired to VS1053	SDCARD	Wired to the rest
GPIO16	-	-	pin 1 XDCS	-	-
GPIO5	-	-	pin 2 XCS	-	-
GPIO4	-	-	pin 4 DREQ	-	-
GPIO2	-	pin 3 D/C or A0	-	-	-
GPIO18	SCLK	pin 5 CLK	pin 5 SCK	CLK	-
GPIO19	MISO	-	pin 7 MISO	MISO	-
GPIO23	MOSI	pin 4 DIN or SDA	pin 6 MOSI	MOSI	-
GPIO21	-	-	-	CS	-
GPIO15	-	pin 2 CS	-	-	-
GPI03	RXD0	-	-	-	Reserved serial input
GPIO1	TXD0	-	-	-	Reserved serial output
GPIO34	-	-	-	-	Optional pull-up resistor
GPIO35	-	-	-	-	Infrared receiver VS1838B
GPIO25	-	-	-	-	Rotary encoder CLK
GPIO26	-	-	-	-	Rotary encoder DT
GPIO27	-	-	-	-	Rotary encoder SW
GND	-	pin 8 GND	pin 8 GND		Power supply GND
VCC 5 V	-	pin 7 BL	-		Power supply
VCC 5 V	-	pin 6 VCC	pin 9 5V		Power supply
EN	-	pin 1 RST	pin 3 XRST		

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.

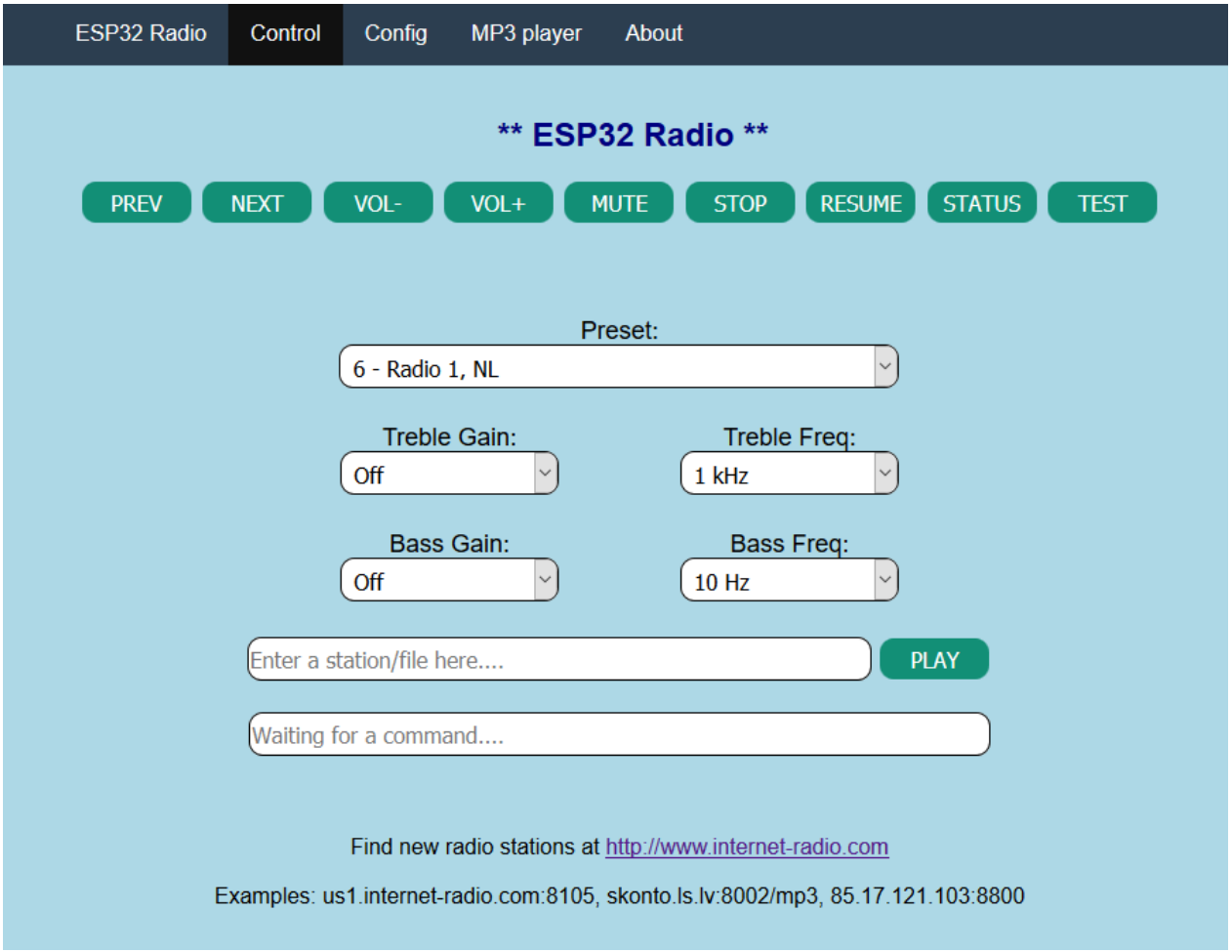
The web interface is simple and can be adapted to your needs. The basic idea is to have a html page with embedded javascript that displays an interface to the radio. Command to the radio can be sent to the http server on the ESP32. The IP address of the webserver will be displayed on the TFT during startup. The webpages are defined in PROGMEM.

Capabilities of the webserver:

Let's assume that the IP of the ESP32-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.h page from the PROGMEM as well as favicon_ico.h.

If your computer is configured for mDNS, you can also use http://ESP32Radio.local in your browser.

The following simple web interface will be displayed:



Clicking on one of the available buttons will control the ESP32-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 ESP32-radio in the form:

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

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

The "STATUS" and "TEST" buttons give some info about the playing stream/file.

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 (or 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*)
mqttprefix = none                     Set MQTT prefix for pub/sub.
clk_server = pool.ntp.org             Time server to be used *)
clk_offset = <-11..+14>               Offset with respect to UTC in hours *)
clk_dst    = <1..2>                   Offset during DST in hours *)
mp3track   = <nodeID> or 0            Play MP3 track from SD card, 0 = random
status                                Show current URL to play
test                                  For test purposes
debug      = 0 or 1                   Switch debugging on or off
reset                                 Restart the ESP32
```

Commands marked with "*)" are sensible during power-up 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 SD card. Example: “localhost/friendly.mp3”.
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 access point will be used.

Configuration

The "Config" button will bring up a second screen. Here you can edit the preferencese. The available Wifi networks are listed as well. The config screen will be shown automatically if the ESP32 cannot connect to one of the WiFi stations specified in the preferences. In that case the ESP32 will act as an accesspoint with the name “ESP32Radio”. You have to connect to this AP with password “ESP32Radio”. Than the ESP32-radio can be reached at http:// 192.168.4.1.
NOTE: It is strongly advised to change the default AP network name and password in the source code.
Change the line with:

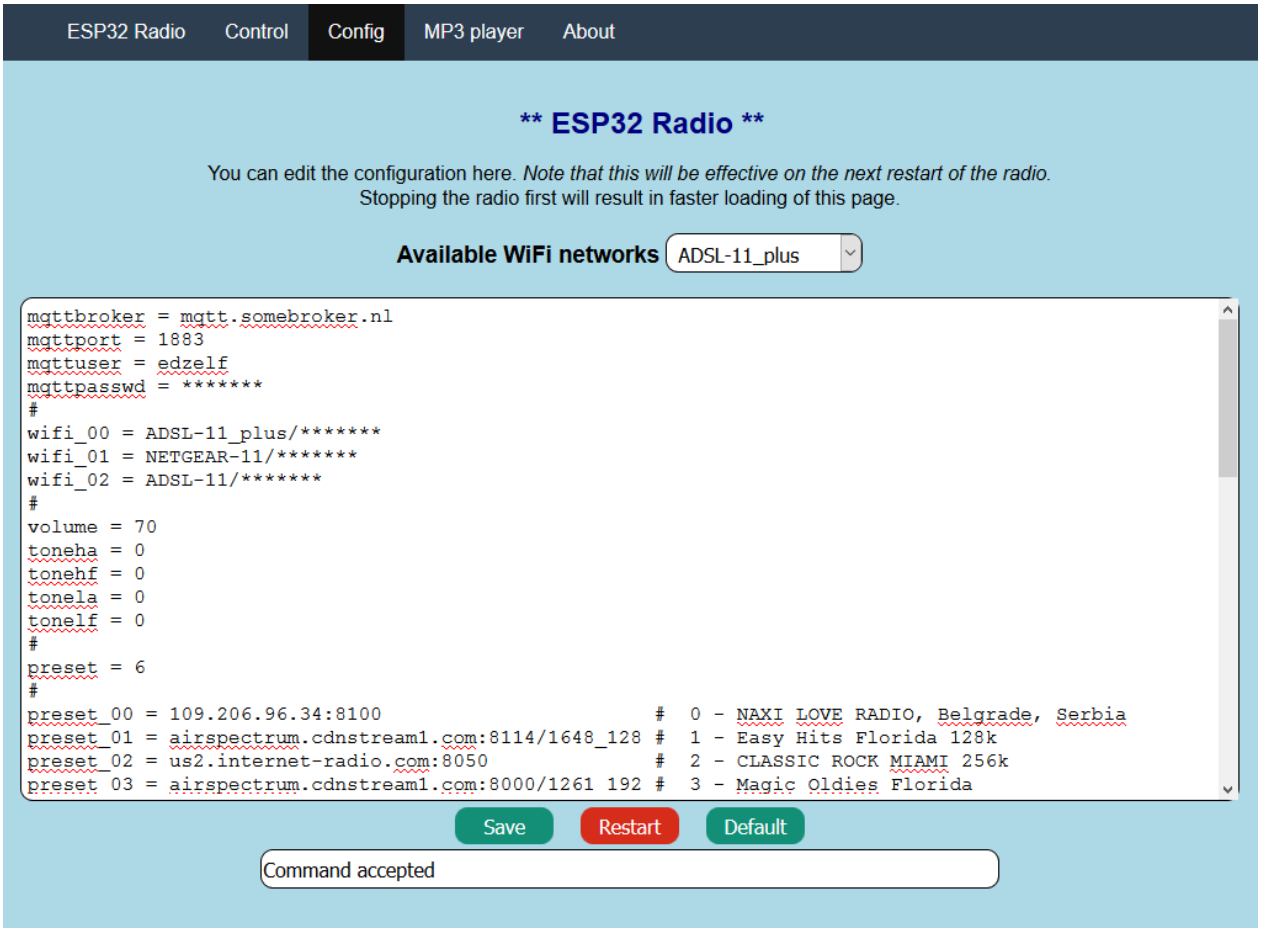
```
WiFi.softAP ( NAME, NAME ) ;
```

to (for example):

```
WiFi.softAP ( ”NW2356”, ”Q8QPREP” ) ;
```

Then you have the opportunity to edit the preferences. A "default" button is available to fill the editbox with some example data. To a quick start, just fill in you WiFi network name and password and click "Save". Restart the radio afterwards.
Note that passwords are hidden in this screen to give some protection against abuses. You may however change them here. If unchanged, the original values are preserved.

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



It may take some time to fill this screen.

MP3 player

ESP32-Radio can also be used as an mp3 player if an SD card interface is available. The screen will look like this:

When this page is called for the first time, a search for all tracks on the SD card will be executed. This may take some time. All found tracks will be available in the drop-down list on the screen. You may pick a track and it will play. When the track has finished, the next track in the drop-down list will be played automatically.

You may also click the "RANDOM" button and a random track will be played. If the track has finished a new random song will be selected.

It may take some time to fill this page because the directories on the SD card are searched for MP3 tracks.

Rotary encoder interface.

The rotary encoder switch can control some essential functions of the ESP32-radio. The "GND"- and "+"-pins must be connected to ground and 3.3 Volt pins of the ESP32 DEV module.

The default function is volume control. Turning the knob will result in lower or higher volume. Pressing the knob will mute/unmute the signal. The text “Mute” or “Unmute” will be shown on the TFT during 4 seconds.

A double click selects the preset-mode. Rotation of the switch will select one of the preset stations. The preset will be shown on the TFT. Once a preset is selected, you can activate this preset by a single click.
Without rotation, the next preset is selected.

A triple click will select the mp3-player (SD card required). Rotation of the switch will select one of the tracks on the SD card. The track will be displayed on the TFT. Once a track is selected, you can activate it by a single click.
Without rotation, the next track is selected.
After the triple click the player stops, as reading filenames will overload the SD card I/O.

A long click (longer than 1 second) will start playing random tracks from SD card.

After an inactivity of 4 seconds the rotary encoder will return to its default function (VOLUME).

Supported displays.

The preferred display is a 1.8 inch TFT with 160x128 pixels. Other displays may also be used:

1. No display. Edit the sketch (around line 123), so that “DUMMYTFT” is defined and all other display types are commented out.
2. The preferred display. Define “BLUETFT” and comment out the other types.
3. A 128x64 OLED. Define “”OLED” and comment out the other types.
4. AN LCD 1602 display with I2C backpack. Define “LCD1602I2C” and comment out the other types. Note that this display is limited as it has only 32 characters.

The OLED and 1602 displays use the serial I2C bus. There must be definitions for this in the preferences, for example:

```
pin_tft_scl = 13                        # GPIO Pin number for SCL
pin_tft_sda = 14                        # GPIO Pin number for SDA
```

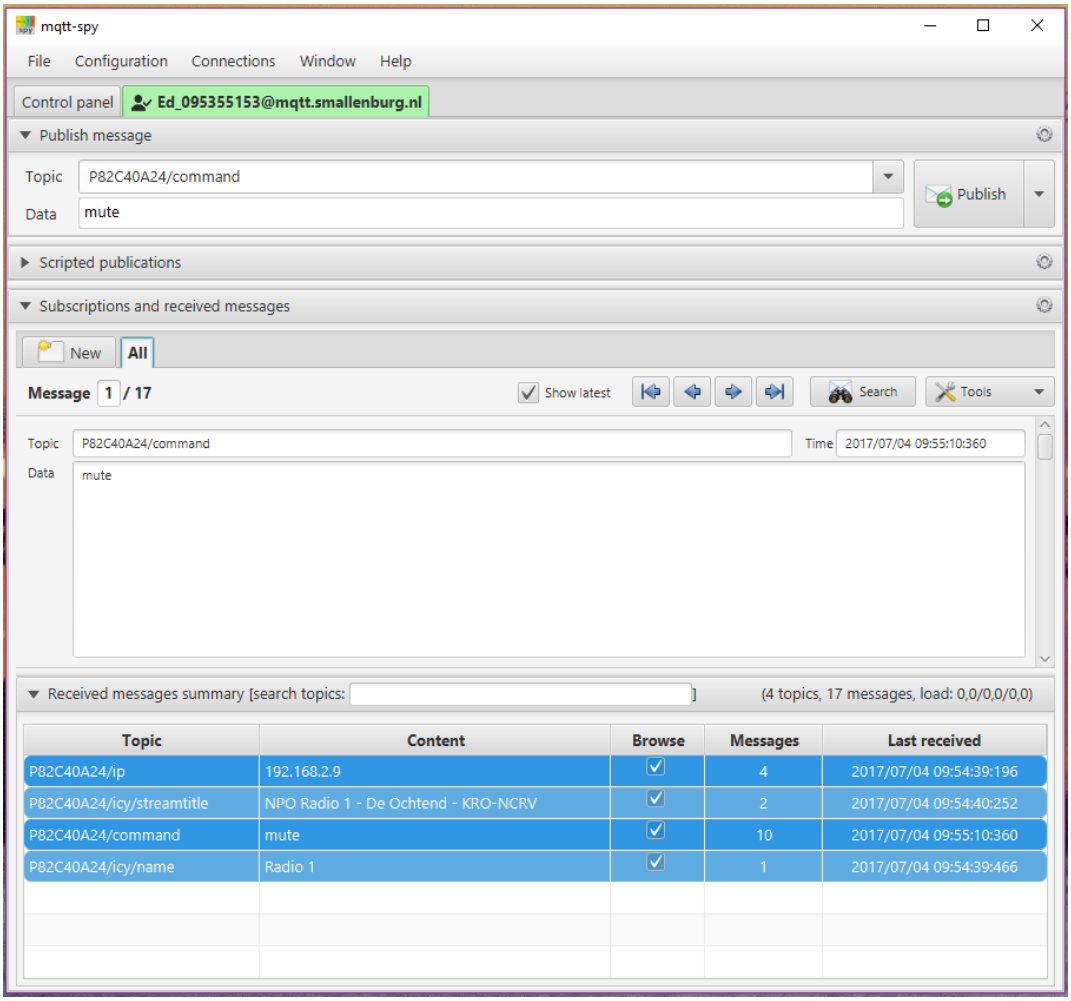
MQTT interface.

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 `broker.hivemq.com` is heavily used, this may cause some delay. If you use your own broker the reaction on commands will be much better.
Remove the lines starting with “mqtt” if no MQTT is required or set mqttbroker to "none".

You can use an MQTT client like https://kamilfb.github.io/mqtt-spy/ to view the MQTT interface.



Subscribe to “P82C40A24/ip” 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. This can be accomplished by specifying a unique prefix in the preferences, see below.
The parameters in preferences for this example are:

```
# MQTT broker, credentials and topic  to subscribe
mqttbroker = broker.hivemq.com          # Broker to connect with
mqttprefix = P82C40A24                  # Prefix for pub/sub.  Default is part of MAC-address.
mqttport = 1883                         # Portnumber (1883 is default)
mqttuser = none                         # (No) username for broker
mqttpasswd = none                       # (No) password for broker
#
```

In this example I published the command “mute” to the radio. The radio published the IP-address 192.168.2.8 to the broker (once every 10 minutes).

MQTT PUB topics.

In this version 7 topics will be published to MQTT and can be subscribed to by an MQTT client:

prefix/ip	The IP-address of the ESP32-Radio
prefix/icy/name	The name of the station
prefix/icy/streamtitle	The name of the stream
prefix/nowplaying	Track information
prefix/preset	Preset currently plaing
prefix/volume	Current volume setting
prefix/playing	1 if playing, 0 if stopped

Command to control the ESP32-Radio can be published to "prefix/command".

Alternative way of configuration.

An extra sketch "Esp32_radio_init" is supplied as an alternative to initialize the preferences (in Non-Voltile Storage of the ESP32). Just change lines 39 and 40 (the specs for WiFi networks) to match your network(s).

Upload and run the sketch once and then load the ESP32-Radio.

Debug (serial 115200 Baud) output.

This is an example of the debug output.

```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
ets Jun  8 2016 00:22:57

rst:0x10 (RTCWDT_RTC_RESET),boot:0x17 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:956
load:0x40078000,len:0
load:0x40078000,len:13076
entry 0x40078ad0

D: Starting ESP32-radio running on CPU 1 at 240 MHz.  Version Thu, 07 Dec 2017 09:30:00 GMT.  Free memory 171632
D: Partition nvs found, 20480 bytes
D: Read 44 keys from NVS
D: pin_ir pin set to 35
D: pin_enc_clk pin set to 25
D: pin_enc_dt pin set to 26
D: pin_enc_sw pin set to 27
D: pin_tft_cs pin set to 15
D: pin_tft_dc pin set to 2
D: pin_sd_cs pin set to 21
D: pin_vs_cs pin set to 5
D: pin_vs_dcs pin set to 16
D: pin_vs_dreq pin set to 4
D: pin_spi_sck pin set to -1
D: pin_spi_miso pin set to -1
D: pin_spi_mosi pin set to -1
D: GPIO0 is HIGH
D: GPIO2 is HIGH
D: GPIO4 is HIGH
D: GPIO5 is HIGH
D: GPIO12 is HIGH
D: GPIO13 is HIGH
D: GPIO14 is HIGH
D: GPIO15 is HIGH
D: GPIO16 is HIGH
D: GPIO17 is HIGH
D: GPIO18 is HIGH
D: GPIO19 is HIGH
D: GPIO21 is HIGH
D: GPIO22 is HIGH
D: GPIO23 is HIGH
D: GPIO25 is HIGH
D: GPIO26 is HIGH
D: GPIO27 is HIGH
D: GPIO32 is HIGH
D: GPIO33 is HIGH
D: GPIO34 is LOW, probably no PULL-UP
D: GPIO35 is HIGH
D: gpio_00 will execute uppreset = 1
D: gpio_12 will execute upvolume = 2
D: gpio_13 will execute downvolume = 2
D: Start TFT
D: Locate mp3 files on SD, may take a while...
D: 190 tracks on SD
D: Added ADSL-11_plus to list of networks
D: Added NETGEAR-11 to list of networks
D: Added ADSL-11 to list of networks
D: Scan Networks
D: Scan completed
D: Number of available networks: 6
D:  1 - ADSL-11_plus              Signal: -60 dBm, Encryption WPA_WPA2_PSK, Acceptable
D:  2 - NETGEAR-11                Signal: -61 dBm, Encryption WPA2_PSK, Acceptable
D:  3 - ADSL-11                   Signal: -68 dBm, Encryption WPA_WPA2_PSK, Acceptable
D:  4 - KPN Fon                   Signal: -69 dBm, Encryption OPEN,
D:  5 - Roulet 9 Gast             Signal: -85 dBm, Encryption WPA_WPA2_PSK,
D:  6 - Roulet 9                  Signal: -86 dBm, Encryption WPA_WPA2_PSK,
D: End of list
D: Command: clk_dst with parameter 1
D: Command: clk_offset with parameter -1
D: Command: clk_server with parameter pool.ntp.org
D: Command: gpio_00 with parameter uppreset = 1
D: Command: gpio_12 with parameter upvolume = 2
D: Command: gpio_13 with parameter downvolume = 2
D: Command: ir_40bf with parameter upvolume = 2
D: Command: ir_c03f with parameter downvolume = 2
D: Command: mqttbroker with parameter mqtt.smallenburg.nl
D: Command: mqttpasswd with parameter *******
D: Command: mqttport with parameter 1883
D: Command: mqttuser with parameter edzelf
D: Command: pin_enc_clk with parameter 25
D: Command: pin_enc_dt with parameter 26
D: Command: pin_enc_sw with parameter 27
D: Command: pin_ir with parameter 35
D: Command: pin_sd_cs with parameter 21
D: Command: pin_tft_cs with parameter 15
D: Command: pin_tft_dc with parameter 2
D: Command: pin_vs_cs with parameter 5
D: Command: pin_vs_dcs with parameter 16
D: Command: pin_vs_dreq with parameter 4
D: Command: preset with parameter 4
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 www.classicfm.nl/player/classicfm.m3u
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: volume with parameter 72
D: Command: wifi_00 with parameter ADSL-11_plus/*******
D: Command: wifi_01 with parameter NETGEAR-11/*******
D: Command: wifi_02 with parameter ADSL-11/*******
D: Slow SPI, Testing VS1053 read/write registers...
D: Fast SPI, Testing VS1053 read/write registers again...
D: endFillByte is 0
D: Connect to WiFi
D: Connected to ADSL-11_plus
D: IP = 192.168.2.18
D: Start server for commands
D: Network found. Starting mqtt and OTA
D: MQTT uses prefix P82C40A24
D: Init MQTT
D: MDNS responder started
D: Rotary encoder is enabled
D: Sync TOD
D: Sync TOD, new value is 13:51:51
D: STOP requested
D: (Re)connecting number 1 to MQTT mqtt.smallenburg.nl
D: New preset/file requested (4/0) from airspectrum.cdnstream1.com:8008/1604_128
D: Connect to new host airspectrum.cdnstream1.com:8008/1604_128
D: Connect to airspectrum.cdnstream1.com on port 8008, extension /1604_128
D: Connected to server
D: Switch to HEADER
D: Headerline: Content-Type: audio/mpeg
D: audio/mpeg seen.
D: Headerline: icy-br:128
D: Headerline: icy-description:Magic 60s Florida - The Best Hits of the 60s  Decade 24/7 Non Stop!
D: Headerline: icy-genre:60s Oldies Top 40 Light Rock Pop
D: Headerline: icy-name:Magic 60s Florida
D: Headerline: icy-pub:1
D: Headerline: icy-url:http://www.MagicOldiesFlorida.com
D: Headerline: Server: Icecast 2.4.0-kh4
D: Headerline: Cache-Control: no-cache, no-store
D: Headerline: Access-Control-Allow-Origin: *
D: Headerline: Access-Control-Allow-Headers: Origin, Accept, X-Requested-With, Content-Type
D: l-Allow-Headers: Origin, Accept, X-Requested-With, Content-Type seen.
D: Headerline: Access-Control-Allow-Methods: GET, OPTIONS, HEAD
D: Headerline: Connection: Close
D: Headerline: Expires: Mon, 26 Jul 1997 05:00:00 GMT
D: Headerline: icy-metaint:16000
D: Switch to DATA, bitrate is 128, metaint is 16000
D: Metadata block 80 bytes
D: Streamtitle found, 75 bytes
D: StreamTitle='Heard It Through the Grapevine - Gladys Knight & Pips (1967)';
D: Metadata block 64 bytes
D: Streamtitle found, 56 bytes
D: StreamTitle='The Beatles - You Really Got a Hold On Me';
```