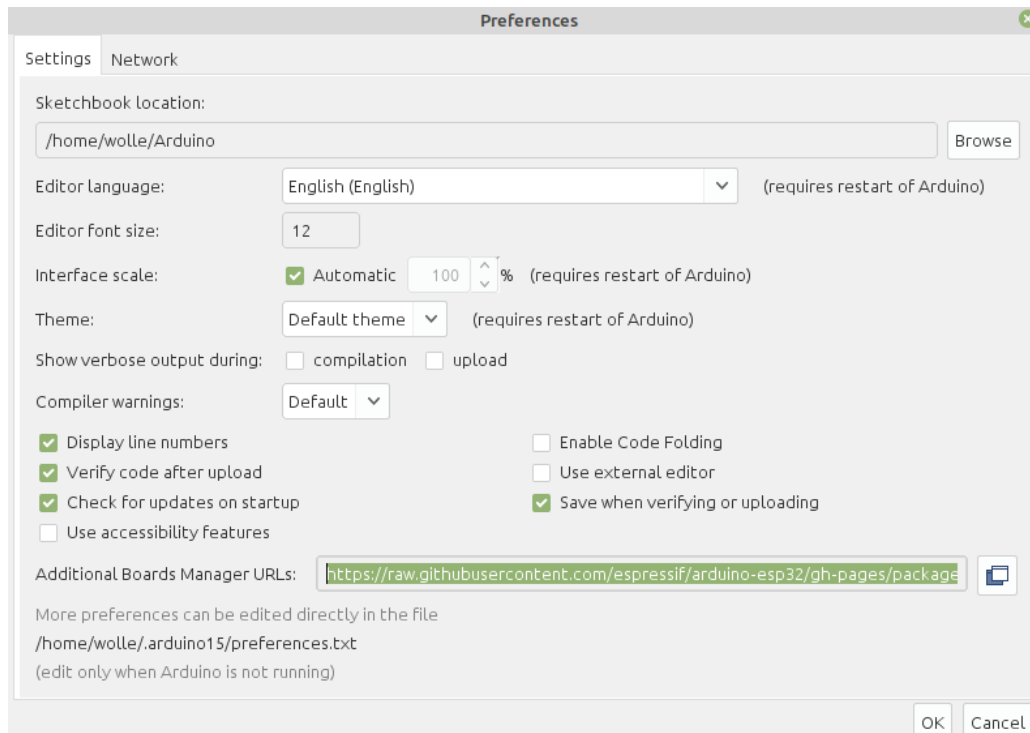


# Notes on programming with the Arduino IDE

In Preferences:

Additional Boards Manager URLs:

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json)



Install V1.0.6 via Boards Manager



In (user) AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.6\tools\partitions  
copy the partition table miniwebradio.csv

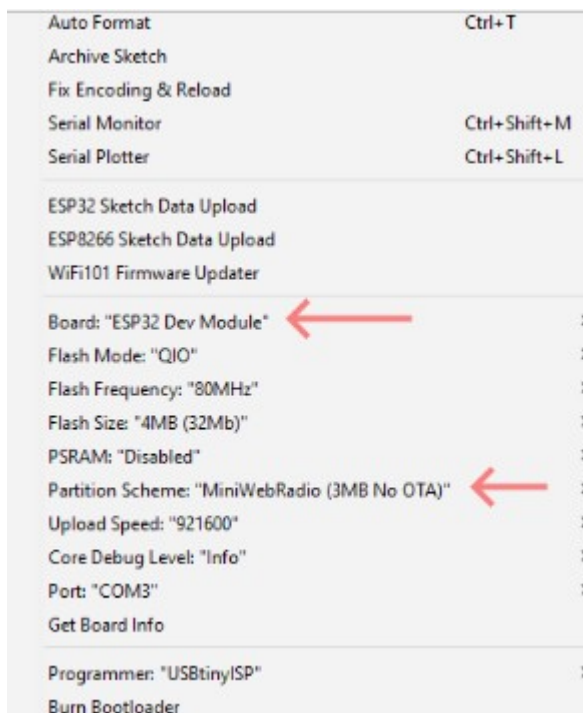
app3M_fat9M_16MB.csv	335 Bytes	Text
boot_app0.bin	8,2 kB	Binär
default.bin	3,1 kB	Binär
default.csv	262 Bytes	Text
default_8MB.csv	262 Bytes	Text
default_16MB.csv	262 Bytes	Text
default_ffat.csv	262 Bytes	Text
ffat.csv	335 Bytes	Text
huge_app.csv	217 Bytes	Text
large_spiffs_16MB.csv	262 Bytes	Text
minimal.csv	222 Bytes	Text
miniwebradio.csv	305 Bytes	Text
min_spiffs.csv	261 Bytes	Text
no_ota.csv	218 Bytes	Text
noota_3g.csv	218 Bytes	Text
noota_3gffat.csv	291 Bytes	Text
noota_ffat.csv	291 Bytes	Text

Then add in boards.txt (section ESP32 Dev Module):

```
esp32.menu.PartitionScheme.miniwebradio=MiniWebRadio (3MB No OTA)
esp32.menu.PartitionScheme.miniwebradio.build.partitions=miniwebradio
esp32.menu.PartitionScheme.miniwebradio.upload.maximum_size=3145728
```

```
esp32.menu.PartitionScheme.default=Default
esp32.menu.PartitionScheme.default.build.partitions=default
esp32.menu.PartitionScheme.minimal=Minimal (2MB FLASH)
esp32.menu.PartitionScheme.minimal.build.partitions=minimal
esp32.menu.PartitionScheme.no_ota=No OTA (Large APP)
esp32.menu.PartitionScheme.no_ota.build.partitions=no_ota
esp32.menu.PartitionScheme.no_ota.upload.maximum_size=2097152
esp32.menu.PartitionScheme.min_spiffs=Minimal SPIFFS (Large APPS with OTA)
esp32.menu.PartitionScheme.min_spiffs.build.partitions=min_spiffs
esp32.menu.PartitionScheme.min_spiffs.upload.maximum_size=1966080
esp32.menu.PartitionScheme.miniwebradio=MiniWebRadio (3MB No OTA)
esp32.menu.PartitionScheme.miniwebradio.build.partitions=miniwebradio
esp32.menu.PartitionScheme.miniwebradio.upload.maximum_size=3145728
```

## Select Board and Partition Scheme



Install libraries

Sketch/Include Library/Add .Zip Library

[https://github.com/schreibfaul1/ESP32-vs1053\\_ext](https://github.com/schreibfaul1/ESP32-vs1053_ext)

<https://github.com/schreibfaul1/ESP32-IR-Remote-Control> Optional, for a IR Remote Control)

<https://github.com/schreibfaul1/ESP32-TFT-Library-ILI9431-HX8347D>

Create the new project "MiniWebRadio"  
and copy the files into the project folder, rename main.cpp to MiniWebRadio.ino

If everything is included, the contents of the folder will look like this:

index.h	56,2 kB
miniwebradio.csv	305 Bytes
MiniWebRadio.ino	63,2 kB
rtime.cpp	13,4 kB
rtime.h	1,3 kB
websrv.cpp	28,5 kB
websrv.h	3,6 kB

The contents of the archive „Content\_on\_SD\_Card. zip "  
[https://github.com/schreibfaul1/ESP32-MiniWebRadio/blob/master/Content\\_on\\_SD\\_Card.zip](https://github.com/schreibfaul1/ESP32-MiniWebRadio/blob/master/Content_on_SD_Card.zip)  
will be unzipped to the SD card.

Name	Size
..	
voice_time	
ring	
png	
mp3files	
logo	
digits	
day	
btn	
unknown.jpg	3.187
stations.txt	1.635
Night_Gown.bmp	43.062
networks.txt	1.135
MiniWebRadio_gr.jpg	33.793
MiniWebRadio.jpg	22.706
favicon.ico	1.536
Brightness.bmp	112.374

voice_time	Language files for the time (can be played at any hour)
ring	MP3 file for the alarm tone
pictures	Bitmaps to test the display (not strictly required)
mp3files	Music files etc. for the MP3 player
logo	Sender logos as bitmap (96x96 pixels in size)
digits	Alarm clock and time bitmaps
day	Bitmaps for the day (alarm on/off)
btn	Bitmaps for the buttons
stations.txt	The channel list
networks.txt	If more than one WiFi network exists, the access data can be entered here
favicon.ico	is displayed by the browser on the Web portal.
miniwebradio.jpg	The Home screen
Brightness.bmp	Display Brightness menu graphic

Choose the tft controller

```
//objects
TFT tft(1); // parameter: (0)ILI9341, (1)HX8347D
VS1053 mp3(VS1053_CS, VS1053_DCS, VS1053_DREQ);
hw_timer_t* timer=NULL; // instance of the timer
```

And the timezone

```
// Timezone
#define TZName "CET-1CEST,M3.5.0,M10.5.0/3"
```

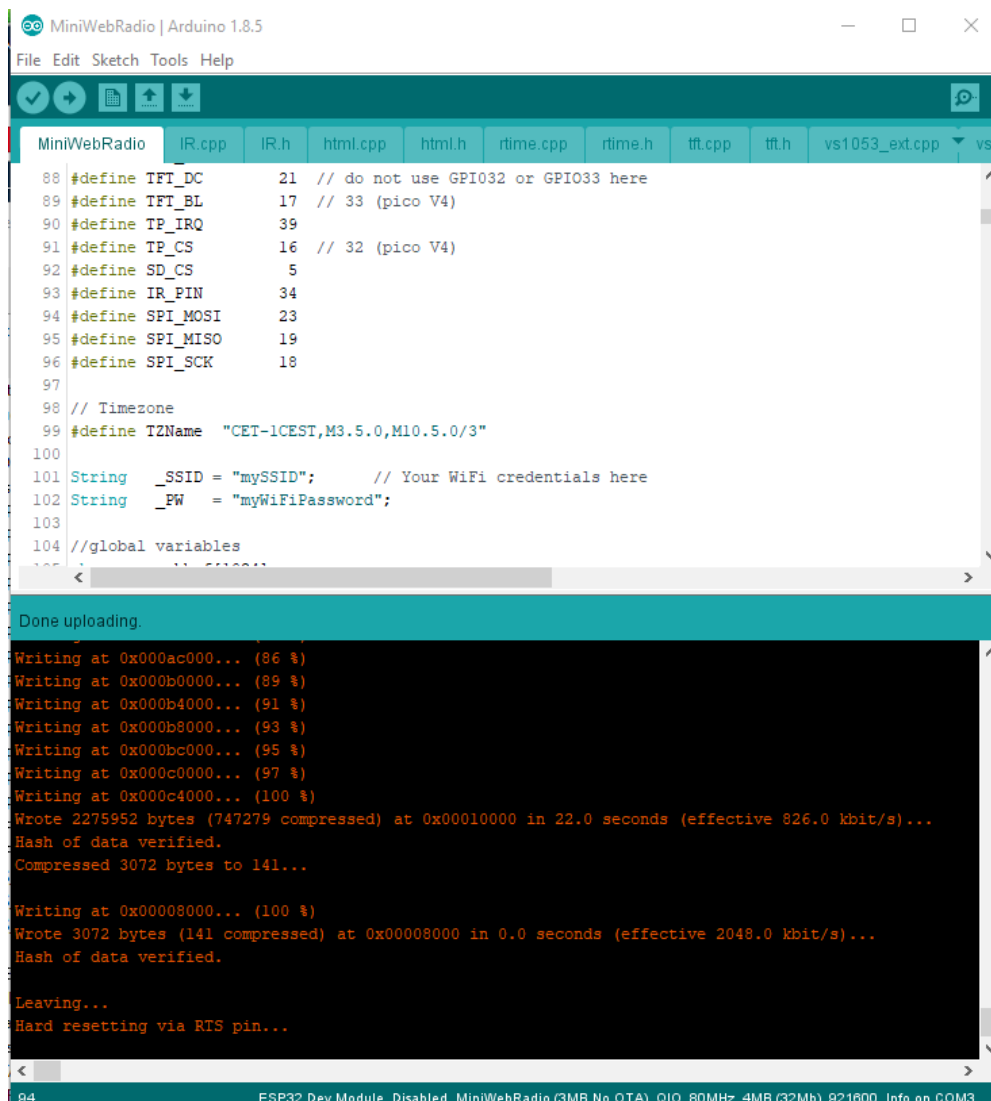
Goto tft.h and include (uncomment) the font Times\_New\_Roman

```
// this font needs a bigger partition change from "default" to "NO OTA (large app)"
#include "fonts/Times_New_Roman.h" // latin, greek, cyrillic with all extensions
```

Set Your credentials in the code or in networks.txt

```
String _SSID = "mySSID"; // Your WiFi credentials here
String _PW = "myWiFiPassword";
```

After that, the sketch can be compiled and uploaded.



The screenshot shows the Arduino IDE interface with the 'MiniWebRadio' sketch open. The code editor displays various preprocessor definitions for pins and hardware, followed by the timezone definition and WiFi credentials. The 'Tools' menu is open, showing the 'Upload' option. Below the code editor, a status bar indicates the upload progress and completion. The console window shows the upload process, including writing to memory, verifying the hash, and compressing the data.

```
MiniWebRadio | Arduino 1.8.5
File Edit Sketch Tools Help
MiniWebRadio IR.cpp IR.h html.cpp html.h rtime.cpp rtime.h tft.cpp tft.h vs1053_ext.cpp vs1
88 #define TFT_DC 21 // do not use GPIO32 or GPIO33 here
89 #define TFT_BL 17 // 33 (pico V4)
90 #define TP_IRQ 39
91 #define TP_CS 16 // 32 (pico V4)
92 #define SD_CS 5
93 #define IR_PIN 34
94 #define SPI_MOSI 23
95 #define SPI_MISO 19
96 #define SPI_SCK 18
97
98 // Timezone
99 #define TZName "CET-1CEST,M3.5.0,M10.5.0/3"
100
101 String _SSID = "mySSID"; // Your WiFi credentials here
102 String _PW = "myWiFiPassword";
103
104 //global variables
105
Done uploading.
Writing at 0x000ac000... (86 %)
Writing at 0x000b0000... (89 %)
Writing at 0x000b4000... (91 %)
Writing at 0x000b8000... (93 %)
Writing at 0x000bc000... (95 %)
Writing at 0x000c0000... (97 %)
Writing at 0x000c4000... (100 %)
Wrote 2275952 bytes (747279 compressed) at 0x00010000 in 22.0 seconds (effective 826.0 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 141...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (141 compressed) at 0x00008000 in 0.0 seconds (effective 2048.0 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
94 ESP32 Dev Module, Disabled, MiniWebRadio (3MB No OTA), QIO, 80MHz, 4MB (32Mb), 921600, Info on COM3
```

Sincerely,

**Wolle**