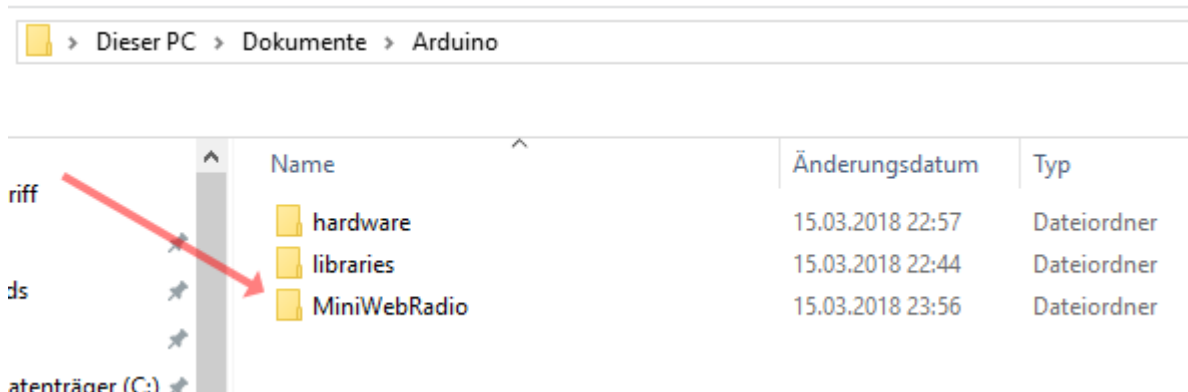


# Hinweise zur Programmierung mit der Arduino IDE

---

**Die Arduino IDE muss installiert und die Bibliotheken für den ESP32 eingebunden sein.**

Legen Sie einen neuen Sketch an und speichern Sie diesen als MiniWebRadio. Die IDE legt einen neuen Ordner mit dem Namen MiniWebRadio an.



Am einfachsten ist es, in diesem Ordner alle benötigten Bibliotheken hinzuzufügen. Die erforderlichen Dateien finden Sie in meinen Repositories.

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














<https://github.com/schreibfaul1/ESP32-IR-Remote-Control> (optional, für eine IR Fernbedienung)

Zusätzlich wird der Treiber für ein SPI-Display mit Touchpad benötigt. Für das Waveshare 2.8inch Display ist das

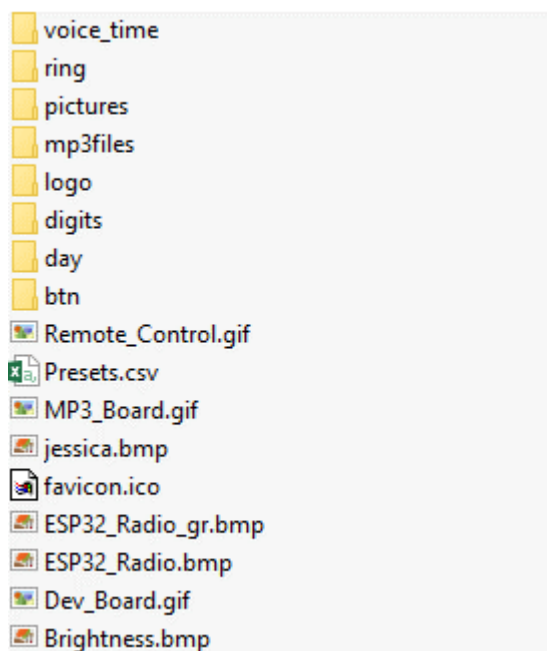
<https://github.com/schreibfaul1/ESP32-TFT-Library-ILI9431-HX8347D>

Für andere Displays ist eine Anpassung notwendig. Gut geeignet sind die TFT Libraries von Adafruit.

Ist alles eingebunden, sieht der Inhalt des Ordners so aus:

Name	Änderungsdatum	Typ	Größe
 IR.cpp	31.10.2017 07:39	CPP-Datei	6 KB
 IR.h	31.10.2017 07:39	H-Datei	1 KB
 fonts.h	14.03.2018 09:30	H-Datei	1.424 KB
 tft.cpp	14.03.2018 09:30	CPP-Datei	38 KB
 tft.h	14.03.2018 09:30	H-Datei	10 KB
 vs1053_ext.cpp	15.03.2018 10:50	CPP-Datei	44 KB
 vs1053_ext.h	15.03.2018 10:50	H-Datei	9 KB
 html.cpp	15.03.2018 13:11	CPP-Datei	10 KB
 html.h	15.03.2018 13:11	H-Datei	2 KB
 rtime.cpp	15.03.2018 13:11	CPP-Datei	3 KB
 rtime.h	15.03.2018 13:11	H-Datei	1 KB
 web.h	15.03.2018 13:11	H-Datei	25 KB
 MiniWebRadio.ino	15.03.2018 23:17	INO-Datei	46 KB

Der Inhalt des Archivs „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) wird auf die SD Card entpackt.



voice_time	Sprachdateien für die Uhrzeit (kann zu jeder vollen Stunde gespielt werden)
ring	mp3 Datei für den Weckton
pictures	Bitmaps zum Testen des Displays (nicht unbedingt erforderlich)
mp3files	Musikdateien etc. für den mp3 Player
logo	Senderlogos als Bitmap (96x96 Pixel groß)
digits	Bitmaps für Wecker und Uhrzeit
day	Bitmaps für den Tag (Wecker ein/aus)
btn	Bitmaps für die Buttons
preset.csv	die Senderliste, kann editiert werden, die ersten 256 Einträge werden im






favicon.ico	internen nvs gespeichert wird vom Browser auf dem Webportal angezeigt. Die Standard URL ist: <a href="http://esp32radio/index.html">http://esp32radio/index.html</a>
ESP32_Radio.bmp	der Startbildschirm
Brightness.bmp	Grafik des Menüs Display Helligkeit

Da mehr nvs Speicher für die Senderliste erforderlich ist muss die Partitionstabelle geändert werden.

```
// if not enough space in nvs: change default.csv
//
// Name,      Type, SubType, Offset,  Size,      Flags
// otadata,   data, ota,      0xe000,   0x2000,
// app0,      app,  ota_0,    0x10000,  0x140000,
// app1,      app,  ota_1,    0x150000, 0x130000,
// nvs,       data, nvs,      0x280000, 0x10000,
// eeprom,    data, 0x99,    0x290000, 0x1000,
// spiffs,    data, spiffs,   0x291000, 0x169000
//
```

Das kann mit einem Texteditor gemacht werden.

> Dokumente > Arduino > hardware > espressif > esp32 > tools > partitions

Name	Änderungsdatum	Typ	Größe
 _default.csv	15.03.2018 23:29	Microsoft Excel-C...	1 KB
 boot_app0.bin	09.03.2018 02:16	BIN-Datei	8 KB
 default.bin	09.03.2018 02:16	BIN-Datei	3 KB
 default.csv	10.12.2017 19:53	Microsoft Excel-C...	1 KB
 minimal.csv	09.03.2018 02:16	Microsoft Excel-C...	1 KB

Oder alternativ wird die default.csv mit der Datei aus den additional\_info überschrieben.

Danach kann der Sketch kompiliert und hochgeladen werden.

The screenshot shows the Arduino IDE interface. The top menu bar includes 'Datei', 'Bearbeiten', 'Sketch', 'Werkzeuge', and 'Hilfe'. The toolbar contains icons for opening, saving, and uploading files. The file explorer shows the project 'MiniWebRadio' with files: IR.cpp, IR.h, fonts.h, html.cpp, html.h, rtime.cpp, rtime.h, tft.cpp, tft.h, vs1053\_ext.cpp, vs1053\_ext.h, and web.h. The main editor displays the code for 'web.h', which includes 'rtime.h' and 'web.h'. It defines various pins and global variables. The status bar at the bottom indicates 'ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 921600, None auf COM3'. The bottom panel shows the upload progress, indicating that the upload is complete and the device is resetting.

```
#include "rtime.h"
#include "web.h"

// Digital I/O used
#define VS1053_CS      2
#define VS1053_DCS     4
#define VS1053_DREQ    36
#define TFT_CS         22
#define TFT_DC         21
#define TFT_BL         17
#define TP_IRQ         39
#define TP_CS          16
#define SD_CS          5
#define IR_PIN         34

//global variables

char sbuf[256], myIP[100];
String _station="", _title="", _info="", _myIP="", _stationname="", _alarmtime="", _time_s="", _hour="", _bitrate="";
String _mp3Name[10], _pressBtn[5], _releaseBtn[5];
```

Hochladen abgeschlossen.

writing at 0x0005c000... (76 %)  
Writing at 0x00060000... (80 %)  
Writing at 0x00064000... (84 %)  
Writing at 0x00068000... (88 %)  
Writing at 0x0006c000... (92 %)  
Writing at 0x00070000... (96 %)  
Writing at 0x00074000... (100 %)  
Wrote 1003696 bytes (424212 compressed) at 0x00010000 in 10.4 seconds (effective 770.0 kbit/s)...  
Hash of data verified.  
Compressed 3072 bytes to 119...

Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (119 compressed) at 0x00008000 in 0.0 seconds (effective 4915.1 kbit/s)...  
Hash of data verified.

Leaving...  
Hard resetting...

ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 921600, None auf COM3

**Mit freundlichen Grüßen,**

**Wolle**