# WEBRADIO MIT ESP32, OLED-DISPLAY UND WIRELESS UPDATE (OTA)

geschrieben von Peter S. | August 11, 2023

**Zusammenfassung**

Radioempfang über Frequenzmodulation hat seine Tücken. Nicht immer lassen sich alle Sender der Wahl empfangen, die zudem auch lokal begrent sind. Rauschfreiheit ist im Analogempfang ebenso nicht immer gewährleistet und schränkt die Wiedergabequalität teils deutlich ein. Mittles eines ESP32 Mikrokontrollers,eines 128×64 OLED-displays und einem Encoder wurde ein simples Webradio gebaut, das kabellos programmiert werden kann und zum Anschluss an einen Audioverstärker geeignet ist.

**Einleitung**

Zum linearen Radioempfang gibt es heutzutage – ergänzend zur klassischen Frequenzmodulation – einige Möglichkeiten. Neben der aktuellsten Erfindung der Digitalradios, die eine perfekte Audiowiedergabe ohne störendes Rauschen und lokale Senderbegrenzungen versprechen, existieren von den meisten Radiostationen Webstreams, die bei vorhandener Internetverbindung auf einer Vielzahl von Endgeräten abgespielt werden können. Denn auch bei Digitalradios kann es, ähnlich wie beim analogen Audioempfang, zum auftreten von Funklöchern kommen, sodass auch mit modernster Technik kein Hörgenuss entstehen kann.

Das Webradio eigenet sich in diesem Fall als verlässliche Rückfallebene. Der Stream wird aufgrund Zwischenspeicherung (Buffering) i.d.R. nicht in Echtzeit wiedergegeben, sondern kann einige Sekunden bis Minuten hinterherhängen. Für wen dies kein Problem darstellt eigenet sich die Wiedergabe von Audio-Streams als ideale Alternative, falls sowohl Analog-, als auch Digitalradio versagen.

In diesem Beitrag wurde auf Basis eines ESP32-Mikrocontrollers ein Internetradio entworfen und mittels 3D gedrucktem Gehäuse gebaut. Es lassen sich beliebige Online-Streams von Radiosendern einbinden. Die Programmierung erfolgt unkompliziert über ein Webinterface und W-LAN-Verbindung zum Modul.

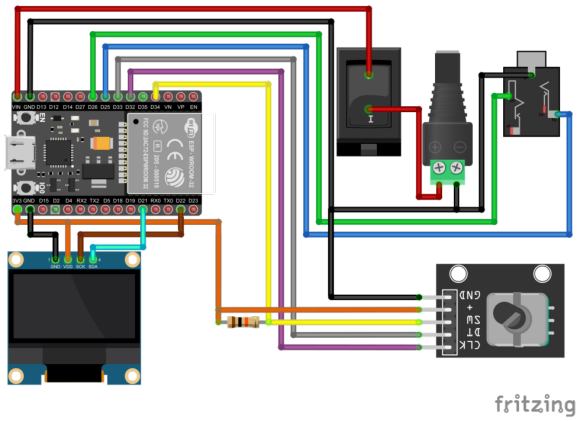
**Materialien & Methoden**

Alle Gehäuseteile wurden mit PLA-Filament auf einem 3D-Drucker gedruckt, andere Materialien sind auch möglich. Als Steckverbindungen wurden gecrimpte Dupont-Stecker mit 2,64 mm Durchmesser (auch als “Arduino-Kabel” bekannt) verwendet. Der Quellcode wurde initial über die Micro-USB-Schnittstelle auf den Mikroprozessor geflasht. Inspiriert wurde das Projekt vom Blogeintrag auf AZ-Delivery.

**Komponenten**

- ESP32 Entwicklerboard (nicht gelötet!)
- Encoder KY-040
- Hohlstecker-Buchse 5.5×2.1
- OLED-Display 128×64 (SSD_1306)
- Klinkenbuchsen PJ320A
- Schalter KCD11
- Netzteil 5 VDC 2000 mAh
- M3 Muttern (3 Stück)
- M3x8 Senkkopfschrauben (3 Stück)
- Widerstand 10 kOhm (1 Stück)
- Kondensator

**Schaltplan**




Schaltpläne des Webradios mit 5-12 VDC Input. Rot: 5 V, Orange: 3,3 V, Schwarz: Masse.

**Software**

- Autodesk Inventor Professional 2023 (Build: 158, Release: 2023 (03/07/2022))
- Fritzing beta (Ver. 0.9.4 )
- Arduino IDE 1.8.13 (Board: ESP32 Dev Module, Upload Speed: 115200, CPU Frequency: 240 Mhz, Flash Frequency: 80 Mhz, Flash Mode: QIO, Flash Size: 4 MB (32 Mb), Partition Scheme: Default 4 MB with spiffs, Core Debug Level: None, PSRAM: Disabled)

**Boardverwaltung**

Esp32 von Espressif Systems – **Version 2.0.9**
https://github.com/iotechbugs/esp32-arduino/blob/master/docs/arduino-ide/boards_manager.md

**Bibliothekenverwaltung**

Es ist absolut notwendig, **die hier angegeben Versionen** der Bibliotheken zu verwenden.

- ESP8266audio – **Version 1.9.7**
- LiquidCrystal_I2C.h – **Version 1.1.2**
- AiEsp32RotaryEncoder.h – **Version 1.4.0**
- AsyncTCP – **Version 1.1.1**
- Adafruit SSD1306 – **Version 2.5.7**
- Adafruit GFX Library – **Version 1.11.5**
- Adafruit BusIo – **Version 1.14.1**

**Sketche**

- audio.ino – streaming and decoding audio data
- lcdisplay.ino – show station data an LCD
- rotary.ino – initialize and handle choosing senders
- senderlist.ino – senderlist by flash-values – or, if not present, by defaults
- senderconfig.ino – asynchronus webinterface to store sendernames und -urls
- senderconf.h – html-template for above
- wlanconfig.ino – seperate program to get WLAN-config per webinterface
- wlanconf.h – html-template for above

Download STL

Download Sketches

**Code**

```
#include <Preferences.h>
```

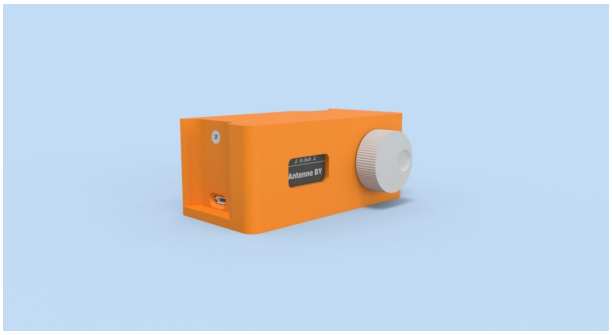
```
//structure for stationlist
typedef struct {
  char url[100];  //stream url
  char name[100]; //stations name
} Station;

#define STATIONS 13 //number of available stations

//station list (stations can now be modified by webinterface)
Station stationlist[STATIONS];

//instances of prefernces
Preferences pref;
Preferences sender;

//global variables
uint8_t curStation = 0;   //index for current selected station in stationlist
uint8_t actStation = 0;   //index for current station in station list used for streaming
uint32_t lastchange = 0;  //time of last selection change
char title[64];
bool newTitle = false;

//setup
void setup()
{
  Serial.begin(115200);
  delay(1000);
  // --------------------
  setup_senderList();
  setup_audio();
  setup_rotary();
  setup_lcdisplay();
  //init WiFi
  Serial.println("Connecting to WiFi");
  while (!makeWLAN())
  {
    Serial.println("Cannot connect :(");
    delay(1000);
  }
  Serial.println("Connected");
  setup_senderConfig();
  // --------------------

  //set current station to 0
  curStation = 0;
  //start preferences instance
   pref.begin("radio", false);
  //set current station to saved value if available
  if (pref.isKey("station")) curStation = pref.getUShort("station");
  //set active station to current station
  actStation = curStation;
  //show on display and start streaming
  showStation();
  startUrl();

}

// main loop
void loop() {
  if (newTitle==true)
  {
    stopscrolling();
```

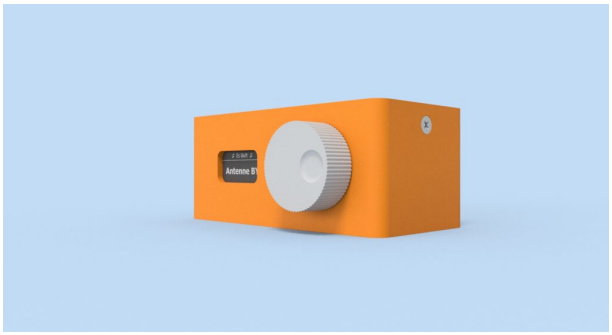
```
    displayTitle();
    newTitle = false;
  }
  //check if stream has ended normally not on ICY streams
  if (!loop_audio())
  {
    Serial.printf("MP3 done\n");

    // Restart ESP when streaming is done or errored
    delay(10000);

    ESP.restart();
  }
  //read events from rotary encoder
  rotary_loop();

}
```

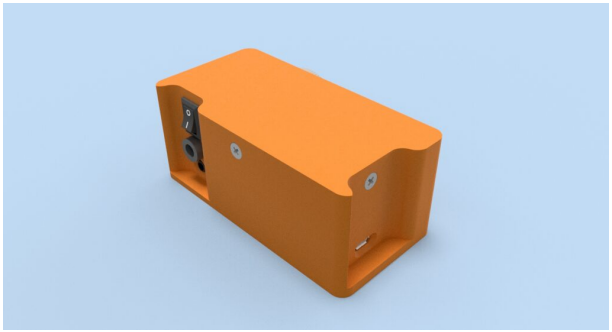
**Ergebnisse**

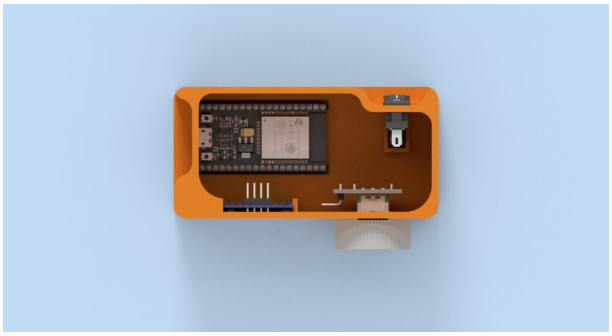
**Hardware**

Das mittels CAD entworfene Konzept wurde 3d-gedruckt und entsprechend des Schaltplans verkabelt. Zur Befestigung des flachen Dreh-Drückrades musste der Schaft des Encoders gekürzt werden. Mit der Verwendung eines Standardknopfs oder eines tieferen Bedienelements entfällt dieser Schritt. Das Display und die 3,5 mm Klinkebuchse wurden mit Heißkleber befestigt. Der Deckel wurde mit M3 Senkkopf-Schrauben befestigt.

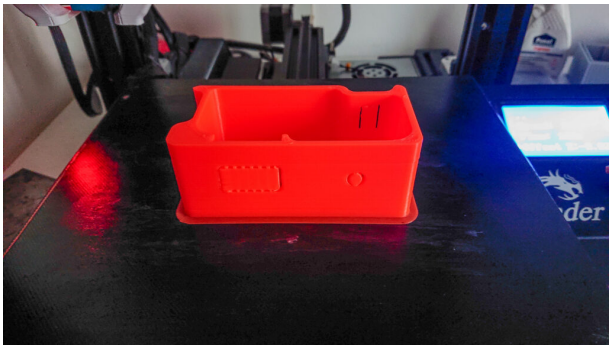


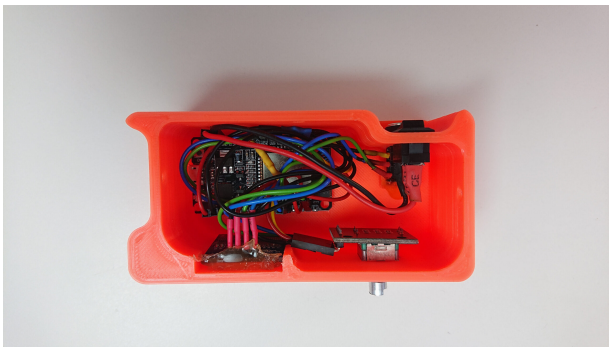


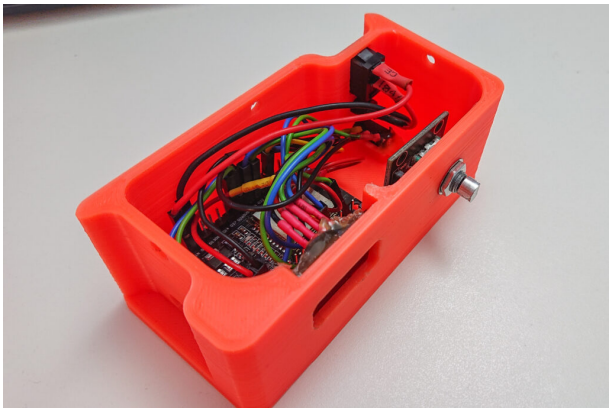




CAD Rendering des Konzepts aus 3D gedrucktem Gehäuse mit Deckel und Wahlrad für den Encoder.








Ergebnis des Konzepts aus 3D gedrucktem Gehäuse mit Deckel und Wahlrad für den Encoder.

Die Zusammenlegung von Klinkenbuchse und Hohlstecker für die Stromversorgung auf der Geräterückseite zeigt sich als optimale Lösung. Der zugängliche USB-Port zum ESP32-Modul erlaubt Codeanpassungen ohne das Modul öffnen zu müssen. Das Dreh-Drückrad des Encoders erlaubt eine benutzerfreundliche Bediunung des Moduls. Das 128×64 OLED-Display integriert sich passend ins Konzept und funktioniert ohne Probleme. Das Webradio eignet sich optimal, um es direkt an einen Verstärker anzuschließen. **Es ist kein Audio-Verstärker verbaut.**

**Software**

Bei erstmaligem Anschalten gibt das Display die Anweisung, sich per Wifi mit dem Radio zu verbinden und die WLAN-Einstellungen einzugeben. Danach ist das Radio mit dem Netzwerk verbunden.

Wird der Encoder vom Benutzer gedreht, wechselt die Anzeige der Kopfzeile im OLED-Display von “Es läuft:” zu “Senderwechsel”. In der Displaymitte wird die Bezeichnung des nächsten Senders eingeblendet und kann mit einem Druck auf den Encoder übernommen werden. Erfolgt im Modus des Senderwechsels 5 Sekunden keine Benutzereingabe wird dieser abgebrochen und der Inhalt der Kopfzeile wechselt zurück zu “Es läuft:”.

Aktuell wird nicht der Titel des aktuellen Songs im Display angezeigt. Das Problem ist hierbei, dass es zu einem hörbaren Knacken kommt, sobald sich der Text aktualisiert. Eine Deaktivierung der Funktion des Textwechsels empfiehlt sich. Stattdessen wird die aktuelle IP-Adresse des Radios im Netzwerk angezeigt. Über die Eingabe dieser Adresse im Browser lassen sich Sender hinzufügen und entfernen.

**Ausblick**

Die verzögerte Einschaltzeit wird sich aufgrund technischer Gegebenheiten nicht signifikant reduzieren lassen. Eine Lösung zur Unterdrückung des als störend empfundenen Knackens bei Textwechsel ließ sich nur durch die Deaktivierung der Funktion erreichen. Es ließen sich alle gewünschten Streams rauschfrei wiedergeben, das Ändern des Radiospeichers über das Webinterface funktioniert reibungslos. Die Wifi-Verbindung ist konstant.

Die Stromversorgung läuft über den USB-Anschluss deutlich zuverlässiger, als über das Netzteil. Daher wird empfohlen noch einen Kondensator vor den ESP zu bauen, falls der Betrieb mit Schalter und Hohlstecker erfolgen soll.