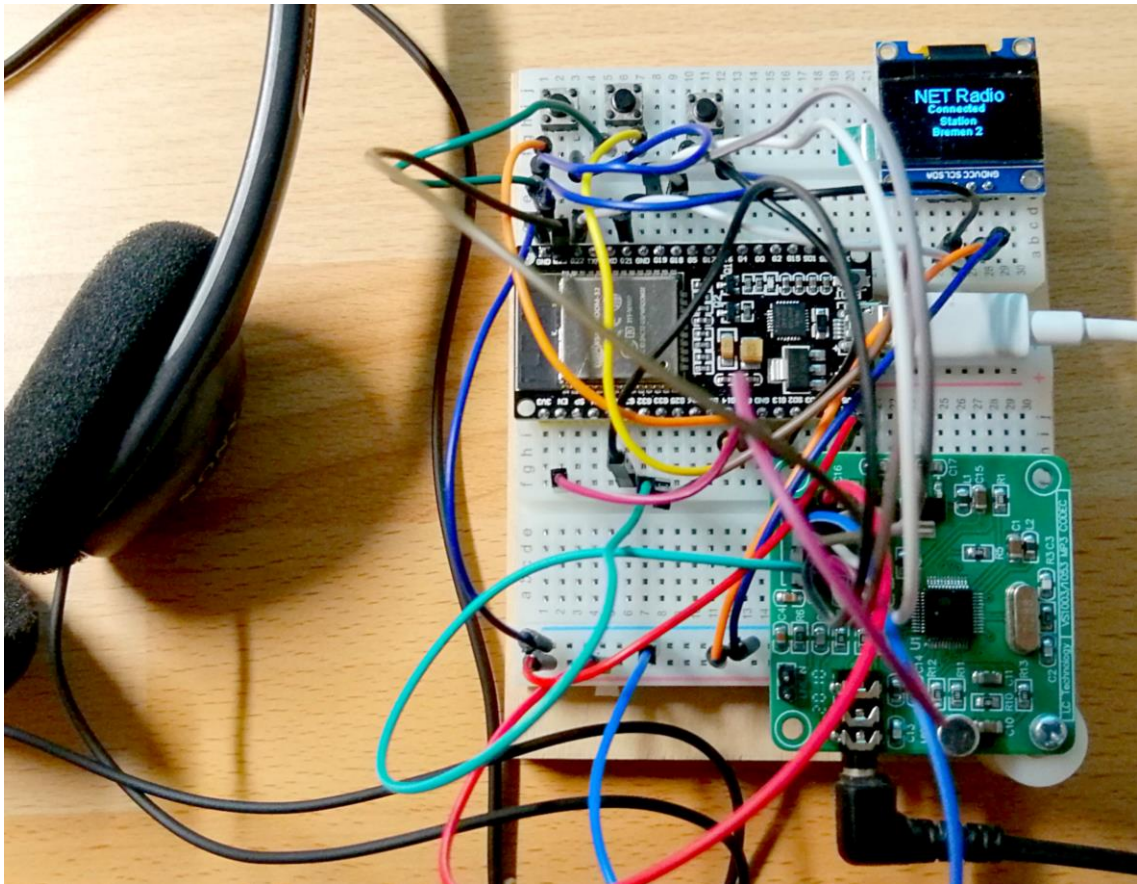


ARDUINO ESP32 Internet Radio V2



Um es vorweg zu nehmen: das Projekt ist eher Puristisch, nur mit dem nötigsten ausgestattet.

Zum Betrieb braucht das Radio keinen Computer. Er wird nur zur erstmaligen Versorgung mit Internetadressen der Radiostationen benötigt.

Ich beschreibe hier die Version 2 vom Net-Radio. Eine Neuprogrammierung wurde notwendig, da einige Streaming Anbieter ihren Audio Stream auf https umgestellt haben und zudem noch temporäre Umleitungen ihrer Internetadressen vornehmen.

Der Code für die Audio Wiedergabe und der Netzwerksteuerung stammt von:
© GPL3+ Lizenz https://github.com/schreibfaul1/ESP32-vs1053_ext Oct 31 2021, Author: Wolle.

Die Kosten für die Bauteile des Net-Radio beliefen sich Ende 2020 auf ca. 19 Euro.
Hinweis: wer bereits die Version 1 nachgebaut hat, muss an der Hardware nichts ändern.

Und wer keine Anforderungen an das Aussehen stellt, belässt die Hardware auf dem Breadboard (Foto). Ich habe das Net-Radio in ein kleines Gehäuse gepackt. Das Audiosignal besitzt Hi-Fi Qualität.

Am Audioausgang kann auch ein Kopfhörer oder der Line- In Eingang eines End-Verstärkers angeschlossen werden.

Der verbaute Decoder Chip VS1053B verarbeitet eine Vielzahl von Audio Formaten. Dieses NET-Radio benutzt das MP3 Streaming Format (MPEG Layer 1+2) mit bis zu 256 Kbit/s bei einer Sampling Rate von 44,1 KHz.

Zur Betriebsweise:

Das ESP32 Internet Radio wird via http, https mit der Radio Station verbunden (Protokoll http 1.1). Die Streaming-Daten werden mit einer Blocklänge von 32 Byte übertragen.

Über den Micro-USB-Anschluss des ESP32 wird das Radio mit Strom versorgt (5V).

Um die Verbindung zu Ihrem WiFi-Netz herzustellen, ersetzen Sie die SSID und das Passwort am Anfang des Codes:

```
String ssid = "????????"; // your network SSID (name)
String password = "????????"; // your network password
```

Nach dem Start (Reset) sagt das Radio zuerst "Hallo", um anzuzeigen, dass es betriebsbereit ist.

Nach kurzer Wartezeit, wird die Verbindung zu Ihrem WLAN Netzwerk hergestellt. In der Zeit wird „Warten auf Verbindung“ am Display angezeigt.

Dann wird die eingestellte Radiostation mit dem ESP32-Radio verbunden.

Ich habe 8 Internet Stationen meiner Wahl am Ende des Programm Codes voreingestellt.

Die Liste kann beliebig verändert werden:

```
void station_connect (int station_no ) // Senderliste
{
    delay(250);
    Serial.println("Connected now");

    if (station_no == 0) x4=" Jazz Radion Berlin "; // Display station names
    if (station_no == 1) x4="          DLF          ";
    if (station_no == 2) x4="   Radio Bremen 2   ";
    if (station_no == 3) x4="          DLF Kultur   ";
    if (station_no == 4) x4="          NDR Blue    ";
    if (station_no == 5) x4="          NDR 1      ";
    if (station_no == 6) x4=" Swiss Jazz Radio  ";
    if (station_no == 7) x4="          NPO Jazz     ";

    x=true;
    preferences.putUInt("counter",new_counter);

    display.clear();
    display.setFont(ArialMT_Plain_16);
    display.drawString(0,0,"          NET Radio");
    display.setFont(ArialMT_Plain_10);

    String x1="Connected";
    display.drawString(35,15,x1);
    x1="Station";
    display.drawString(45,27,x1);
    display.drawString(20,37,x4);
    display.display();
```

```

switch (station_no) {    // play station index-number

    case 0:
        mp3.connecttohost("http://streaming.radio.co/s774887f7b/listen");
        break;

    case 1:
        mp3.connecttohost("https://st01.sslstream.dlf.de/dlf/01/128/mp3/stream.mp3");
        break;

    case 2:
        mp3.connecttohost("https://icecast.radiobremen.de/rb/bremenzwei/live/mp3/128/stream.mp3");
        break;

    case 3:
        mp3.connecttohost("https://st02.sslstream.dlf.de/dlf/02/128/mp3/stream.mp3");
        break;

    case 4:
        mp3.connecttohost("http://icecast.ndr.de/ndr/ndrblue/live/mp3/128/stream.mp3");
        break;

    case 5:
        mp3.connecttohost("http://icecast.ndr.de/ndr/ndr1wellenord/kiel/mp3/128/stream.mp3");
        break;

    case 6:
        mp3.connecttohost("stream.srg-ssr.ch/m/rsj/mp3_128");
        break;

    case 7:
        mp3.connecttohost("http://icecast.omroep.nl/radio6-bb-mp3");
        break;
}

```

Mit der Taste am Digitaleingang 13, wird der Sender gewählt. Nach dem letzten Sender wird zur ersten Station zurück gewechselt. Der eingestellte Sender wird am Display angezeigt.

Mit den Tasten am Digitaleingang 12 und 14 wird die Lautstärke in kleinen Schritten erhöht oder abgesenkt, mit 50% der möglichen Lautstärke beginnend.

Wenn der ESP32 mit der Arduino Entwicklungsumgebung verbunden ist, kann am Seriellen Monitor der Betriebszustand und die tatsächliche URL des eingestellten Senders abgelesen werden.

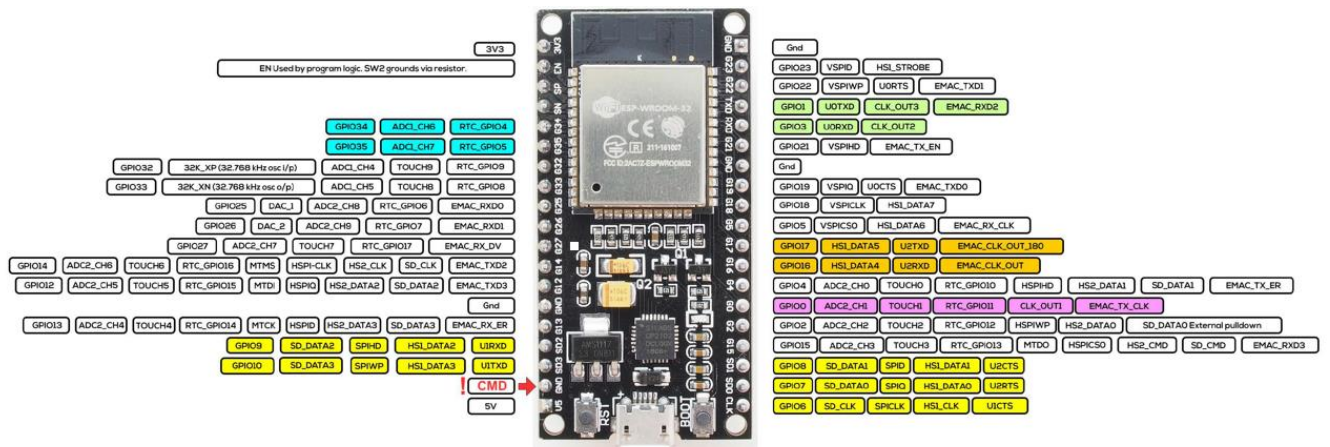
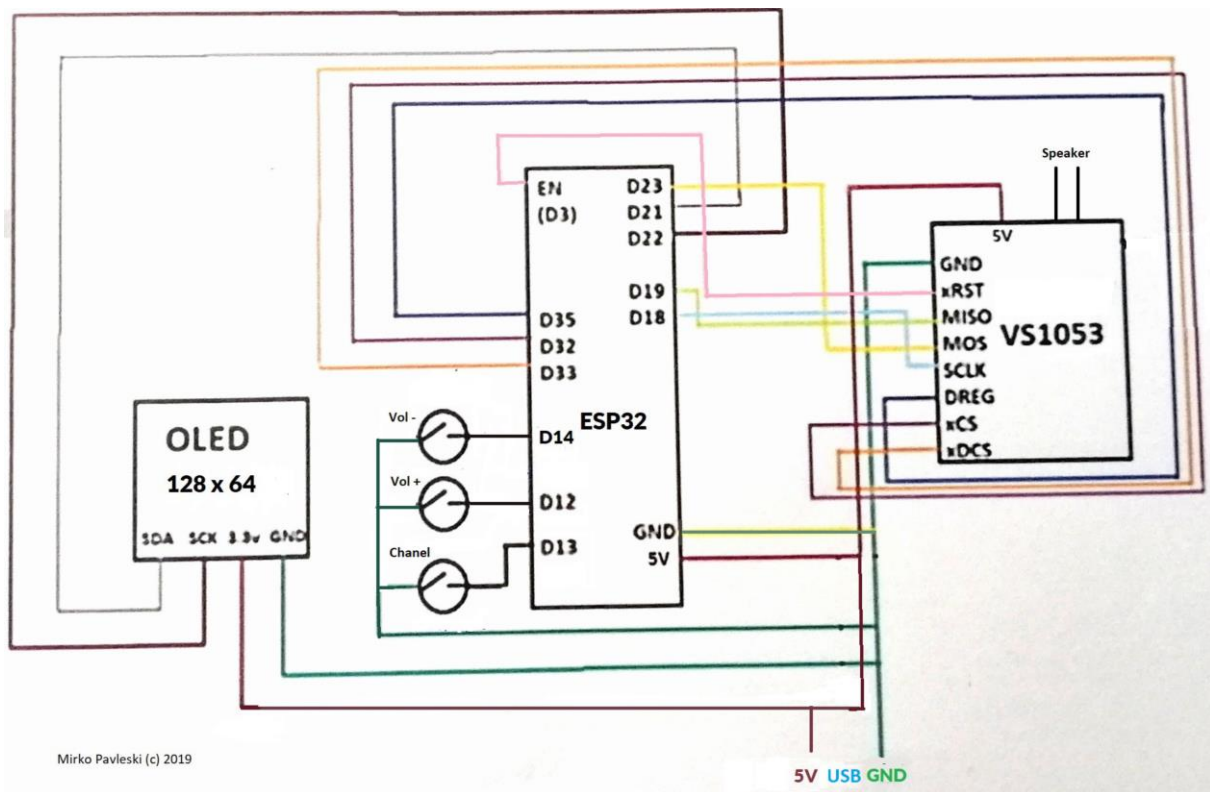
Der Decoder Chip VS1053B bietet viele Einstellungen. So können z.B. Bass und Höhen als 4 Bit Nibble, genauso wie die Lautstärke in % Schritten eingestellt werden.

In der dazugehörigen Bibliothek vs1053_ext.h, finden sich die nötigen Aufrufe um dieses Programm zu erweitern.

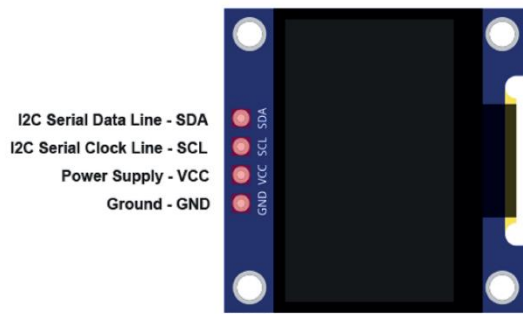
Die ARDUINO Entwicklungsumgebung muss vor der Kompilierung mit dem passenden **ESP32** CPU Stack versorgt werden. Fügen Sie im Boardverwalter der IDE den folgenden Link ein:

https://dl.espressif.com/dl/package_esp32_index.json

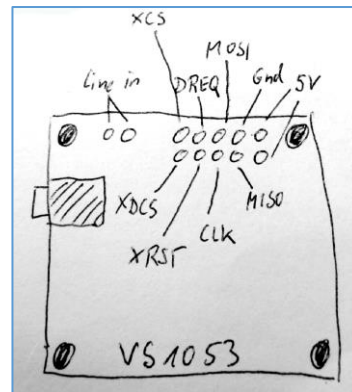
Beschaltung:



Die Beschaltung der ESP32 Node MCU Modul
kann bei einigen Herstellern abweichen!



Display 1306



VS1053 MP3-Modul

Folgende Bauteile werden benötigt:

- **ESP32 Node MCU Modul**
- **VS1053 MP3-Modul-Entwicklungsboard**
- **128X64 OLED Display Module Für Arduino 0.96" I2C**
- **3 Taster**
- **2 x Breadboard**

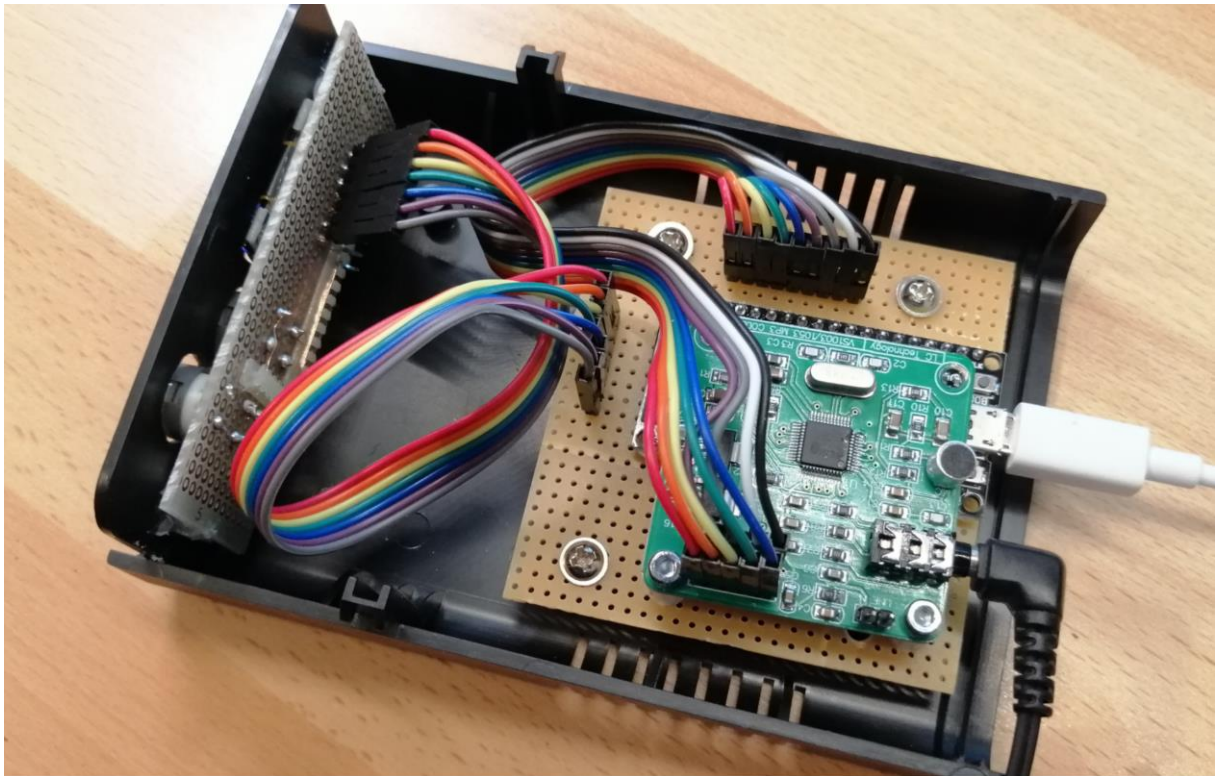
Den Programmcode „netradio.ino“ und die benötigten Bibliotheken finden sich im ZIP-Archiv auf meiner www.film-wwerk56.de Homepage unter LINUX. Das Video zum Projekt auf:

<https://youtu.be/9K4-kPh8ZXU>

<https://www.youtube.com/channel/UCsiwk9sO0zN03tXcwTzV5vQ>



Das NET-Radio im Gehäuse (Selbstbau Tipp).



Viel Spaß beim Nachbau!

Der Autor: Peter Klingbeil, Dezember 2021

Hinweis: für Links außerhalb meiner Domain, übernehme ich keine Verantwortung.