

## **Mein HiFi-Vorverstärker**

Die professionellen Geräte sind sündhaft teuer. Also bleibt der Selbstbau. Er soll mit einer Fernsteuerung bedienbar sein. Über die Stromaufnahme der Steuerung mit einem Arduino denke ich noch nach. Eventuell kommt das Schalten der Netzversorgung hinzu. Dann wird der Arduino im Standby betrieben.

Zunächst mal die Planungen:

### **Gehäuse**

Ich habe die Alu-Platten im Format 43 x 30 cm bestellt. Vorn und hinten sind Aluplatten in der Höhe von 65mm. Die Seitenplatten aus Alu ebenfalls 6,5cm hoch. Damit ist die elektrische Abschirmung gegeben.

### **Anzeigen**

Die Frontplatte enthält ein Fenster. Darin wird ein Display mit 40 x 30 mm sein. Angezeigt werden die geschalteten Audioeingänge(Relais) und der MUTE-Zustand. Im Standby leuchtet eine LED. Es sind 6 Audio-Eingänge vorgesehen. Ein zusätzlicher Audioausgang ist für die Aufzeichnung parallel geschaltet.

### **Fernbedienung**

Ich nehme die Standardfernbedienung für Arduino, die man überall kaufen kann. Die Zifferntasten sind für die

Eingangswahl. Die Pfeiltasten oben/unten für die Stummschaltung(Mute). Die Pfeiltasten horizontal für Volume. Für das Schalten des Netzbetriebs werde ich die Tasten \* und # wählen.

## Software

Ich habe mich für den Einsatz eines Arduino NANO entschieden. Die Funktionalität kann ausreichend erfüllt werden. Die fertigen Clone sind preiswert(China) zu kaufen.

Nachfolgend der Entwurf:

```
//"selection-vorverstärker" oder "selection-amp"; DF8ZR
//mit dieser Software kann man die Audio-Eingänge
//des Verstärkers umschalten und auf den Ausgang
//für den Betrieb des Endverstärkers legen

//Vorsicht beim Einbau des X9C103(elektronisches Poti)!
//Wenn man versehentlich den INPUT(hier A0) am Arduino
//als OUTPUT programmiert hat und dieser
//Port am Schleifer (Wiper) liegt, geht das IC kaputt

//Old Bootloader aktivieren beim Arduino-Clone !
// benötigte Bibliotheken einbinden
#include "IRremote.h"
#include "EEPROM.h"

// der Pin, an dem der Infrarot-Empfänger angeschlossen ist
int EmpfaengerPin = 12;

#include <TFT.h>
#include <SPI.h>

//PIN-Einstellungen
#define cs 10
#define dc 9
#define rst 8
TFT TFTscreen = TFT(cs, dc, rst);

int IN = 0; //Speicheradresse für den Zustand

//Zustände sind die Werte() im Speicher

int pos = 0;
```

```

int posalt = 0;
byte gate = 0;
int mute = 0;

/*U = Increase wiper setting
D = Decrease wiper setting
S = Save current setting into memory for recall after power cycle
*/
const float V_REF = 5.0;          // Change if using different Vref
const int UP = 0;
const int DOWN = 1;
const int UD_PIN = A1;           // Goes to X9C103P U/D pin - Up = HIGH, Down
= LOW                             =pinDIL2
const int INC_PIN = A2;          // Goes to X9C103P INC pin - active falling
edge                               =pinDIL1
const int CS_PIN = A3;           // Goes to X9C103P CS pin - active
LOW                               =pinDIL7
const int WIPER_PIN = A0;        // Goes to X9C103P VW pin - Analog voltage
output of pot                       β      =pinDIL5
float voltage = 0;
//=====
//*****SETUP*****
*****
void setup()
{
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(2, OUTPUT);

    //Pins für X9C1003P
    pinMode(A0, INPUT);
    pinMode(A1, OUTPUT);
    pinMode(A2, OUTPUT);
    pinMode(A3, OUTPUT);
    pinMode(A4, OUTPUT); // für das Netzrelais
    pinMode(A7, OUTPUT); //für das Mute-Relais

    TFTscreen.begin();
    //Hintergrund: Schwarz
    //TFTscreen.background(0, 0, 0);

```

```

TFTscreen.background(ST7735_RED);//wird aber blau angezeigt!?
//Textfarbe: Weiß
TFTscreen.stroke(255, 255, 255);

//Serial.begin(9600);

posalt = (int (EEPROM[IN]));//gespeicherten Zustand(alt) abrufen

pos = posalt;//nach dem Einschalten gültiger Wert

switchON();//das Relais

IrReceiver.begin(EmpfaengerPin); //PIN 12
delay(100);

anzeigen();//gespeichertes pos = posalt
//IR-Empfang starten

pinMode (CS_PIN, OUTPUT);
pinMode (UD_PIN, OUTPUT);
pinMode (INC_PIN, OUTPUT);
Serial.begin(9600);

digitalWrite(INC_PIN, HIGH);
digitalWrite(CS_PIN, LOW); // Enable the X9C103P chip
//Serial.print ("Initial Voltage Setting: ");
//PrintVoltage();
for (int i = 0; i < 100; i++)
{
LautstDOWN();
}

for (int i = 0; i < 30; i++)
{
LautstUP();
}

}

void anzeigen(){

if (mute == 0) {

if (gate == 2) {

```

```

TV();
}
if (gate ==3){
DAB();
}
if (gate == 4){
TUNER();
}
        if (gate == 5){
CD();
}
        if (gate == 6){
AUX1();
}
if (gate == 7){
AUX2();
}

}
}

```

```

void Auswerten(){

        if (pos == 69) {
TV();
}

        if (pos == 70){
DAB();
}

        if (pos == 71){
TUNER();
}

        if (pos == 68){
CD();
}

        if (pos == 64){
AUX1();
}

        if (pos == 67){
AUX2();
}
}

```

```
}
```

```
void LautstUP(){
```

```
digitalWrite(UD_PIN, HIGH); delayMicroseconds(5); // Set to increment  
digitalWrite(INC_PIN, LOW); delayMicroseconds(5); // Pulse INC pin low  
digitalWrite(INC_PIN, HIGH);
```

```
delay(10);
```

```
//PrintVoltage();
```

```
}
```

```
void LautstDOWN(){
```

```
digitalWrite(UD_PIN, LOW); delayMicroseconds(5); // Set to decrement  
digitalWrite(INC_PIN, LOW); delayMicroseconds(5); // Pulse INC pin low  
digitalWrite(INC_PIN, HIGH);
```

```
delay(10);
```

```
//PrintVoltage();
```

```
}
```

```
void switchOFF(int posalt){
```

```
if (posalt == 69) gate = 2;//Taste 1  
if (posalt == 70) gate = 3;//Taste 2  
if (posalt == 71) gate = 4;//Taste 3  
if (posalt == 68) gate = 5;//Taste 4  
if (posalt == 64) gate = 6;//Taste 5  
if (posalt == 67) gate = 7;//Taste 6  
//if (posalt == 7) gate = 8;//Taste 7  
//if (posalt == 21) gate = 9;//Taste 8  
//if (posalt == 9) gate = 10;//Taste 9  
//if (posalt == 25) gate = 12;//Taste 0  
//if ( pos == 13) gate = ;// #  
//if ( pos == 90) gate = 14 ;//<  
//if ( pos == 8) gate = 15 ;//>
```

```
//if ( pos == 82) gate = ;//Pfeil nach unten  
//if ( pos == 28) gate = ;//Taste OK  
//if (gate ==16) digitalWrite(A7, LOW);
```

```
digitalWrite(gate, LOW);//gate ist die PinNr der D-Ausgänge
```

```

}

void switchON(){

if ( pos == 69)   gate = 2;//Taste 1 //TV           D2
if ( pos == 70)   gate = 3;//Taste 2 //DAB/WEB      D3
if ( pos == 71)   gate = 4;//Taste 3 //TUNER        D4
if ( pos == 68)   gate = 5;//Taste 4 /CD           D5
if ( pos == 64)   gate = 6;//Taste 5 //AUX1         D6
if ( pos == 67)   gate = 7;//Taste 6 //AUX2         D7
//if ( pos == 7)   gate = 8;//Taste 7 //
//if ( pos == 21)  gate = 9;//Taste 8
//if ( pos == 9)   gate = 10;//Taste 9
//if ( pos == 25)  gate = 12;//Taste 0
//if ( pos == 13)  gate = ;// #

//if ( pos == 90)  gate = 14;//>
//if ( pos == 8)   gate = 15 ;//<

/*if ( pos == 82)  gate = ;//Pfeil nach unten
if ( pos == 28)   gate = ;//Taste OK
*/
/*if (IrReceiver.decodedIRData.command == 13) Serial.println("Taste #");
   if (IrReceiver.decodedIRData.command == 90) Serial.println("Pfeil
links");
   if (IrReceiver.decodedIRData.command == 8) Serial.println("Pfeil
rechts");
   if (IrReceiver.decodedIRData.command == 24) Serial.println("Pfeil
oben");
   if (IrReceiver.decodedIRData.command == 82) Serial.println("Pfeil
unten");
   if (IrReceiver.decodedIRData.command == 28) Serial.println("OK");*/
//if ( gate ==16) digitalWrite(A7, HIGH);

posalt=pos;
digitalWrite(gate, HIGH);//gate ist die PinNr der D-Ausgänge

}

/*void PrintVoltage()
{
int sampleADC = analogRead(WIPER_PIN);      // Take reading on wiper pin
float volts = (sampleADC * V_REF) / 1023.0; // Convert to voltage
Serial.print("  ADC = ");
Serial.print(sampleADC);
Serial.print("  Voltage = ");
Serial.println(volts, 3);
}*/

void Mute(){

```

```
    TFTscreen.setTextSize(3);
    TFTscreen.background(0,0,0);
    TFTscreen.background(ST7735_RED);//wird aber blau angezeigt!?
    TFTscreen.text("MUTE", 46, 58);
}
```

```
void TV(){
    TFTscreen.setTextSize(3);
    TFTscreen.background(0,0,0);
    TFTscreen.background(ST7735_RED);//wird aber blau angezeigt!?
    TFTscreen.text("TV", 58, 58);
}
```

```
void DAB(){

    TFTscreen.setTextSize(3);
    TFTscreen.background(0,0,0);
    TFTscreen.background(ST7735_RED);//wird aber blau angezeigt!?
    TFTscreen.text("DAB", 54, 28);
    TFTscreen.text("WEB", 54, 58);
    TFTscreen.text("USB", 54,88);
}
```

```
void TUNER(){
    TFTscreen.background(0,0,0);
    TFTscreen.background(ST7735_RED);//wird aber blau angezeigt!?
    TFTscreen.setTextSize(3);
    //Text ausgeben
    TFTscreen.text("FM", 59, 28);
    TFTscreen.text("TUNER", 33, 58);
}
```

```
void CD(){
    TFTscreen.setTextSize(3);
    TFTscreen.background(0,0,0);
    TFTscreen.background(ST7735_RED);//wird aber blau angezeigt!?
    TFTscreen.text("CD", 62, 58);
}
```

```
void AUX1(){

    TFTscreen.setTextSize(3);
    TFTscreen.background(0,0,0);
    TFTscreen.background(ST7735_RED);//wird aber blau angezeigt!?
    TFTscreen.text("AUX1", 47, 58);

}
```

```

void AUX2(){
TFTscreen.setTextSize(3);
TFTscreen.background(0,0,0);
TFTscreen.background(ST7735_RED);//wird aber blau angezeigt!?
TFTscreen.text("AUX2", 46, 58);
}

void Abfragen(){

    if ( pos > 0 && pos < 91){

//andere Codes von fremden Fernsteuerungen bleiben unbeachtet
//allerdings sind Störungen durch fremde Fernbedienungen möglich

    if ( pos != posalt && pos!= 8 && pos!= 90 && pos != 24 && pos != 82){

//wenn die Codes mit dem gespeicherte Code nicht übereinstimmen,
//(8 und 90 sind die waagerechten Pfeiltasten)
//(24 und 82 sind die Mute-Tasten Pfeiloben und unten)
//muss ein anderes Relais angesteuert werden
//dazu wird zunächst das "alte" Relais abgeworfen:

switchOFF(posalt);//bisheriges Relais abschalten

    switchON();//aktuelles Relais einschalten

    EEPROM.put(IN, pos );// wird als posalt gespeichert

        // dadurch wird beim Wiedereinschalten
        // der vorherige Zustand hergestellt

    delay(100);

}

//bis hier wurden nur die Relais-Tasten als posalt gespeichert

if (pos == 24) {
    digitalWrite(A5, HIGH); // Pfeil = MUTE ON

```

```

mute = 1;
Mute();

}

if ( pos == 82) {
    digitalWrite(A5, LOW); //Pfeil = MUTE OFF
    mute = 0;
    pos = posalt;

    Auswerten();
}

if ( pos == 90) LautstUP();

if ( pos == 8) LautstDOWN();
//Serial.println(pos);

if (pos == 13) Netzein(); // # = Netzein
if (pos == 22) Netzaus(); // * = Netzaus

}

}

void Netzein(){
    analogWrite(A4, 255);
}

void Netzaus(){
    analogWrite(A4,0);
}

void loop() {

//if (Serial.available()) DoSerial(); // Just loop looking for user input
//nur aktivieren, wenn Daten auf den Empfänger kommen

if (IrReceiver.decode()) {

    //Abfrage des IR-Wertes, der empfangen wird
    pos = (int (IrReceiver.decodedIRData.command));

delay(200);

```

```
Abfragen();  
//endet mit dem aktuellen pos-Wert  
  
//jetzt anzeigen des Kanals mit aktuellem pos  
anzeigen();  
  
delay(200);//warten bis zum nächsten Tastenklick  
  
  IrReceiver.resume(); // Enable receiving of the next value  
}  
  
}
```

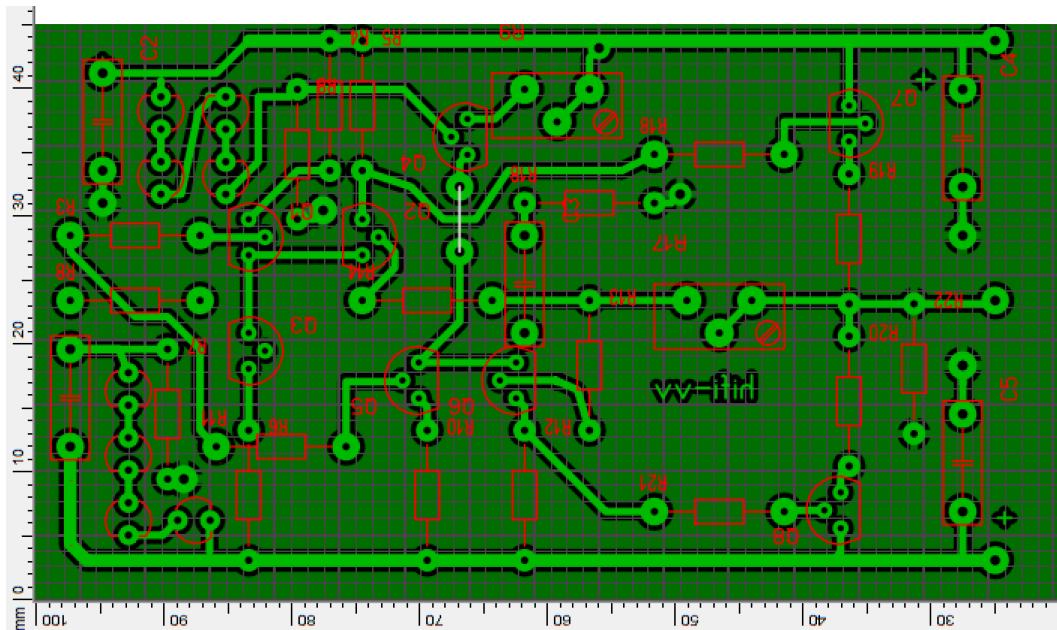
---

Wenn eine andere Fernbedienung verwendet wird, kann man die Codes anpassen.

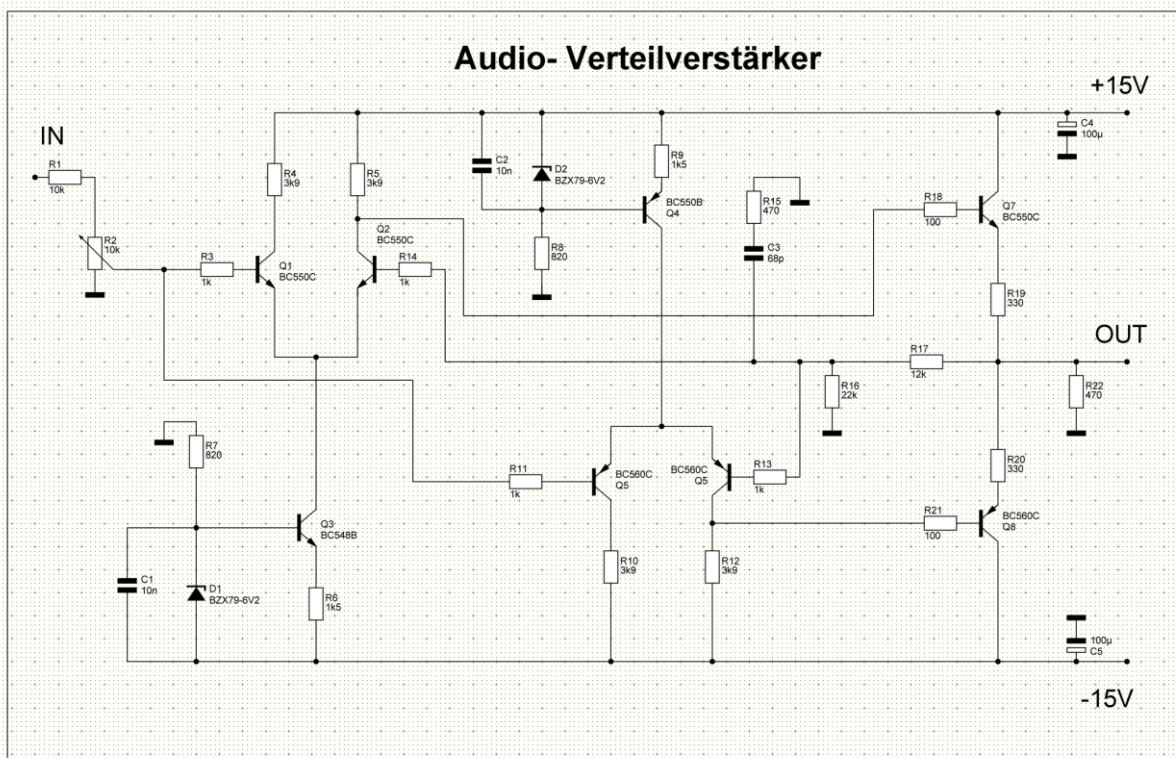
## Platinen

Ich habe mich an die Bauanleitung von [@frankswerkstatt](#)(YouTube)

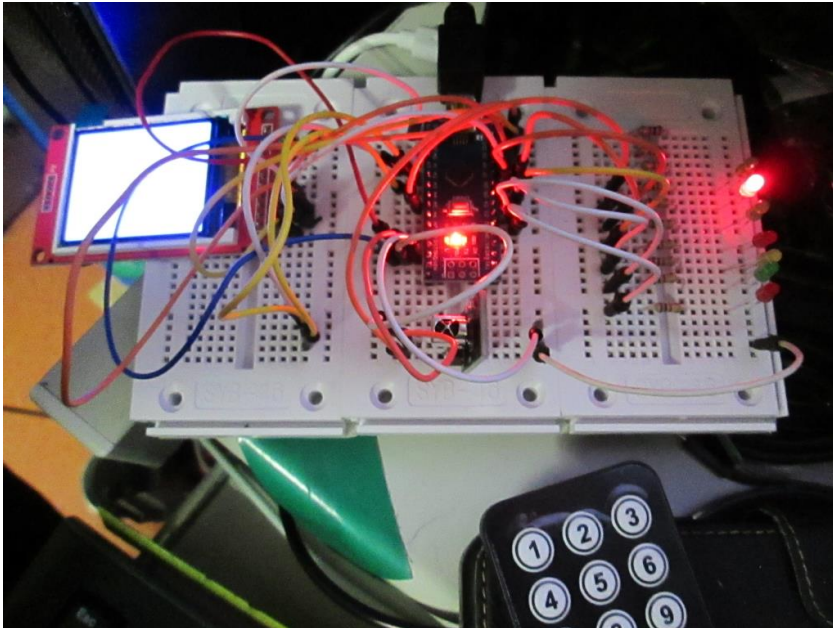
Gehalten. Und sogar die Gleichspannungsreferenzen mit LEDs ausgeführt. Es gibt allerdings moderne ICs, die ebenfalls gute Ergebnisse liefern können. Die zwei Platinen werde ich vom Platinen-Joker(Berlin) für 41,-- EUR Gesamtkosten fertigen lassen. Bei dem üblichen Lieferanten in Bulgarien sind sie inzwischen ebenso teuer. Man kann die Schaltungen aber auch problemlos auf Lochraster aufbauen. Die Gerberdatei werde ich verlinken. Hier das Layout, das ich mit Print-Layout (Abacom) entworfen habe:



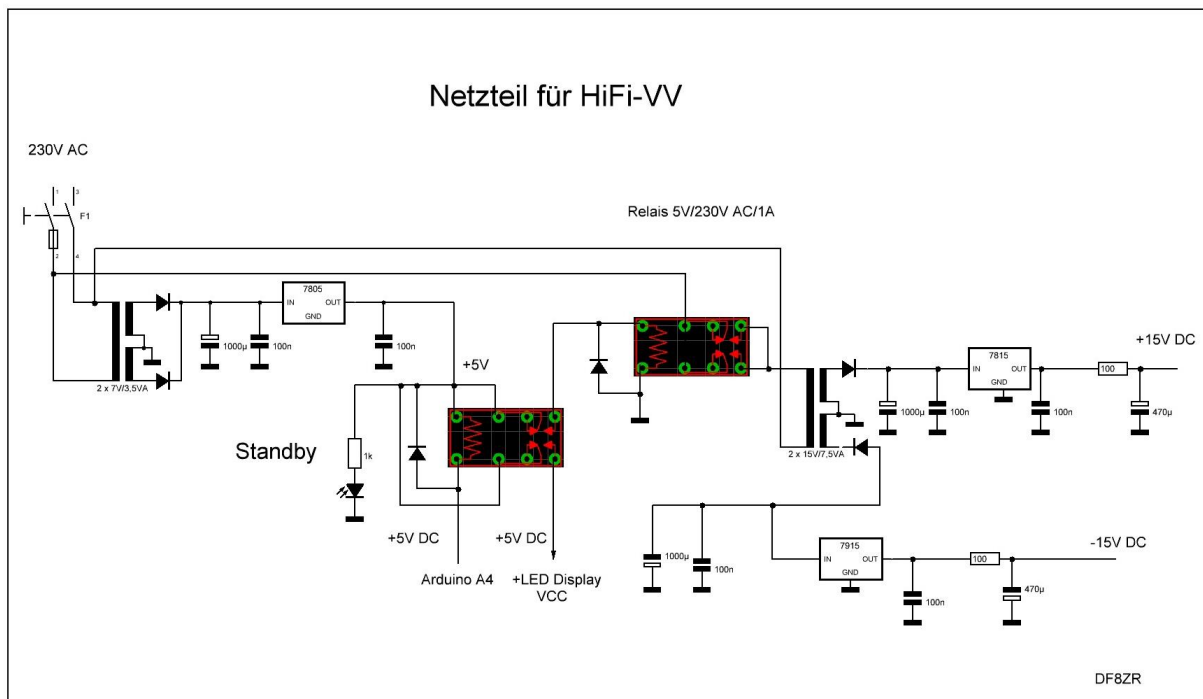
Nachfolgend die Schaltung, die aber die Spannungsreferenzen mit Zenerdioden macht. Frank hat in seinem Video darauf hingewiesen, dass die Reihenschaltung mit LEDs einen „schärferen Knick“ hat. Zum Zeitpunkt seiner Entwicklung traute man den Z-Dioden nicht zu, dass sie rauschfrei sind. Selbstverständlich sollten die LEDs kein Fremdlicht empfangen. Das könnte Brummen erzeugen.



## Der Arduino-Versuchsaufbau



## Netzteil



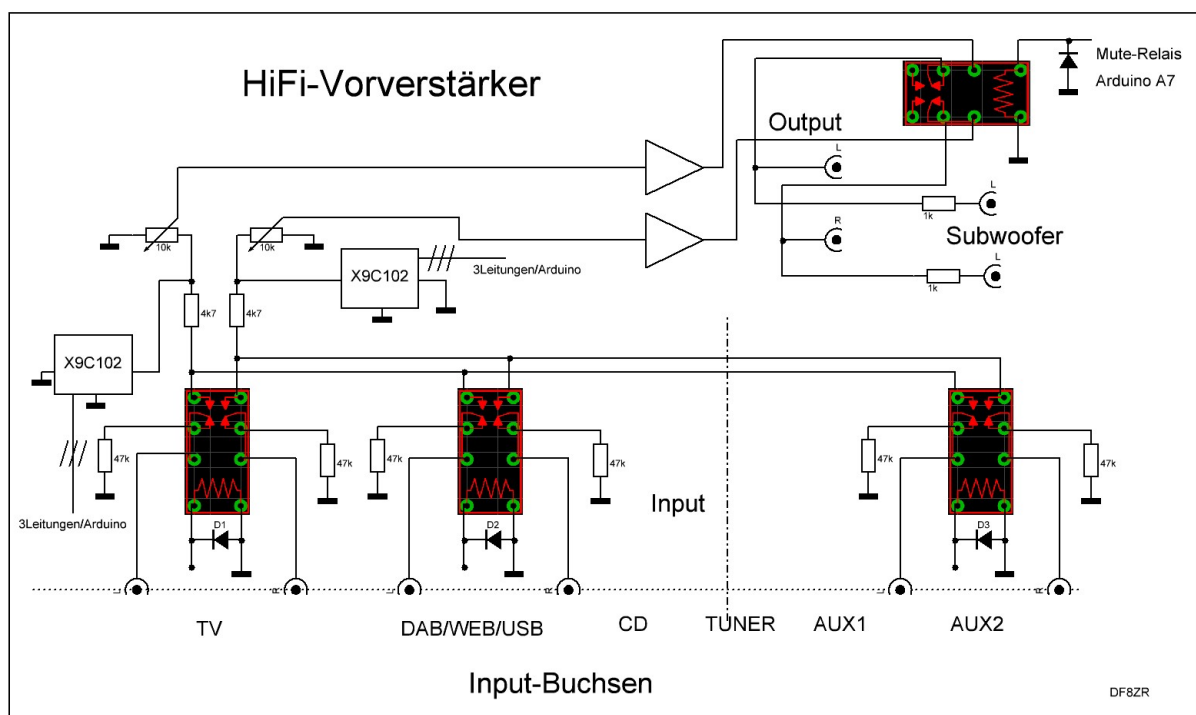
Das Gerät hat einen Netzschalter mit Sicherungshalter. Man kann es also ganz abschalten. Normal befindet es sich im

Standby-Betrieb. Ein 3,5VA-Trafo versorgt bei eingeschaltetem Netzschalter ständig den Arduino mit 5V DC. Erst wenn mit der Fernbedienung der Betrieb zugeschaltet wird, erhält das Display +5V für die Led, die die Hintergrundbeleuchtung macht. Der Betriebszustand ist dadurch signalisiert. Im Standby ist das Display abgeschaltet und dunkel. Allein eine kleine LED zeigt diesen Zustand an.

Falls die einfache Gleichrichtung mit der halbwelle zu einem zu großen Ripple führt, werde ich an beide Wicklungen einen Vollweggleichrichter vorsehen. Die Stromaufnahme der beiden Vorverstärker ist gering. Ich gehe davon aus, dass die Siebung ausreicht.

## Die Buchsen

Nachfolgend der Entwurf.

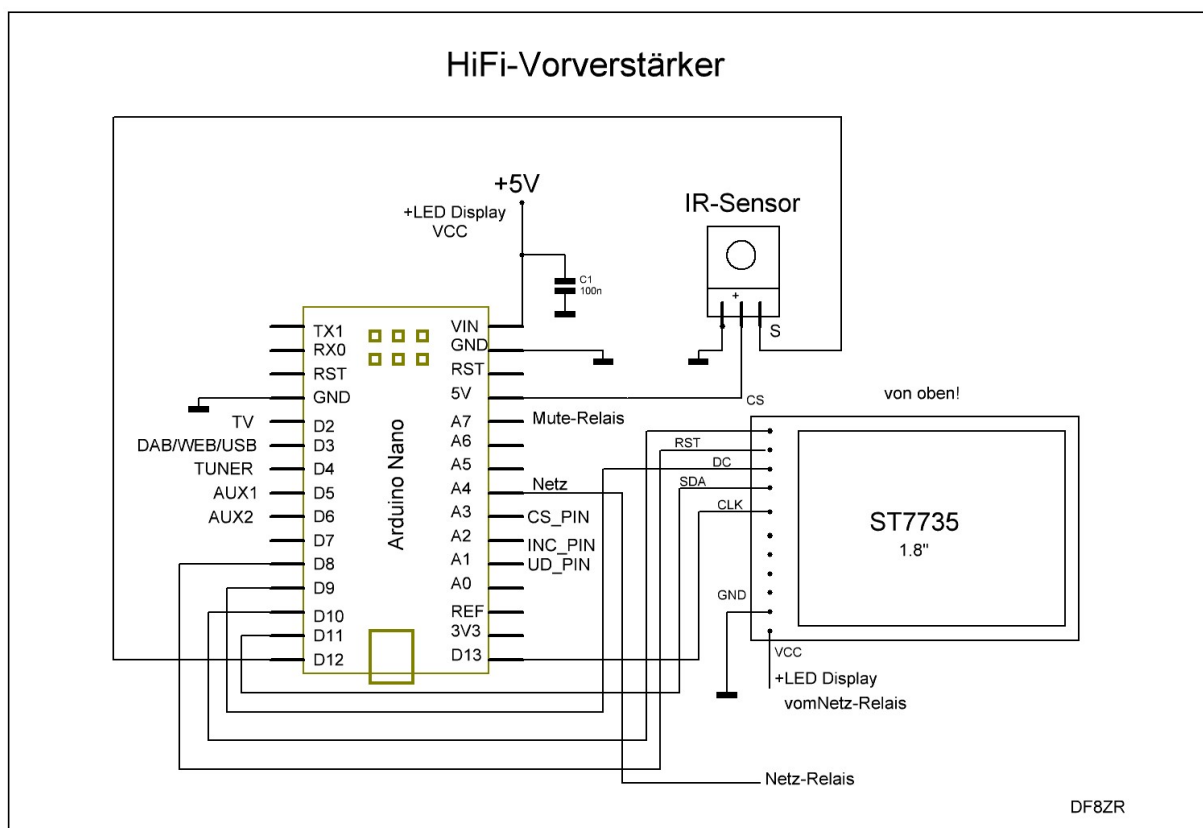


Die beiden X9C102 sind elektronisch steuerbare Potis. Sie werden vom Arduino eingestellt. Damit ist eine Lautstärkeregelung über die Fernsteuerung möglich. Beim Start wird eine untere Lautstärke vom Programm vorgegeben. Der zuletzt eingestellte Wert wird nicht gespeichert.

Die beiden Potis lassen eine Einstellung der Balance zu. Die meisten Vorverstärker haben diese Funktion nicht.

## Arduino

Die Schaltung wird auch im Standby betrieben. Verwendet man ein anderes Display, muss die Pinbelegung angepasst werden. Es sind leider viele unterschiedlich ST7735 unterwegs.



## **Zusammenfassung**

Es können 6 Kanäle ausgewählt werden. Diese Einstellung wird gespeichert, damit beim Wiedereinschalten der zuvor gewählte Kanal durchschaltet. Die Speicherung wird im Arduino gemacht. Diese Einstellung wird auch dauerhaft bei einem Netzausfall gesichert.

Die Balance stellt man gewöhnlich nur einmal fest ein. Die Lautstärke regelt man an der Fernsteuerung. Natürlich geht das auch mit den Potis, denn die Pegel sind meistens hoch genug, wodurch eine Reserve besteht.

Der Stromverbrauch des Arduinos ist gering. Damit ist eine Fernsteuerung des Betriebs möglich. Im Urlaub kann man das Gerät mit dem hinten angebrachten Netzschalter vom Netz trennen.

Mehr ist im Augenblick nicht vorgesehen.

DF8ZR, 25.02.2024; wird fortgesetzt....