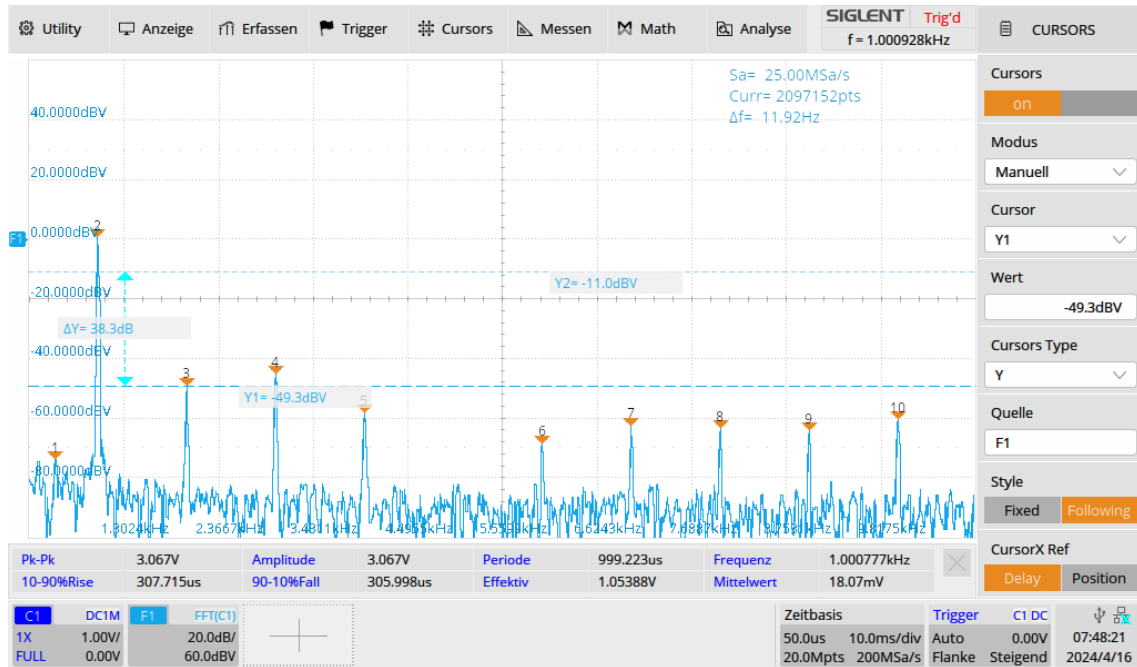


HiFi-VV

Messungen des THD-Pegels

Hier zunächst meine Signalquelle: HP 3310A



Die erste Oberwelle bei 2 kHz hat einen Pegel von fast -50dB. Das ist ein guter Wert für einen RC-Generator.

Und nun der Pegel am Eingang des Verstärkers.



Die Stellung des „Schleifers“ hatte keinen Einfluss auf das Messergebnis. Wir erkennen eine Zunahme des Pegels von +10dB bei 2 kHz. Man kann also vermuten, dass das elektr. Poti einen Einfluss hat. Dort wird im Datenblatt von -100dB für THD angegeben. Es könnte also auch der Eingang des Verstärkers die Ursache sein. Deshalb wurde der Eingang abgeklemmt. Es zeigte sich keine Verbesserung!

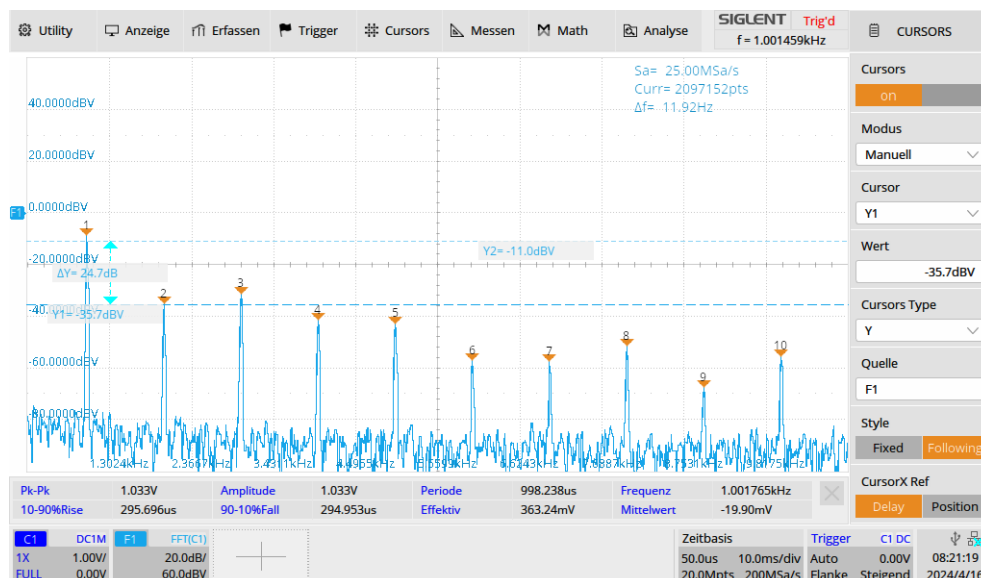
Erkenntnis

Der X90C103P verursacht eine Verschlechterung des THD um 10dB. Das könnte bei einem Poti von Alp mit Motorantrieb besser sein. Oder anders gesagt: „Der X9C103P verursacht zusätzliche Verzerrungen und ist für den HiFi-Betrieb NICHT geeignet!“

Ich werde über den Einsatz eines Alp-Potis nachdenken.

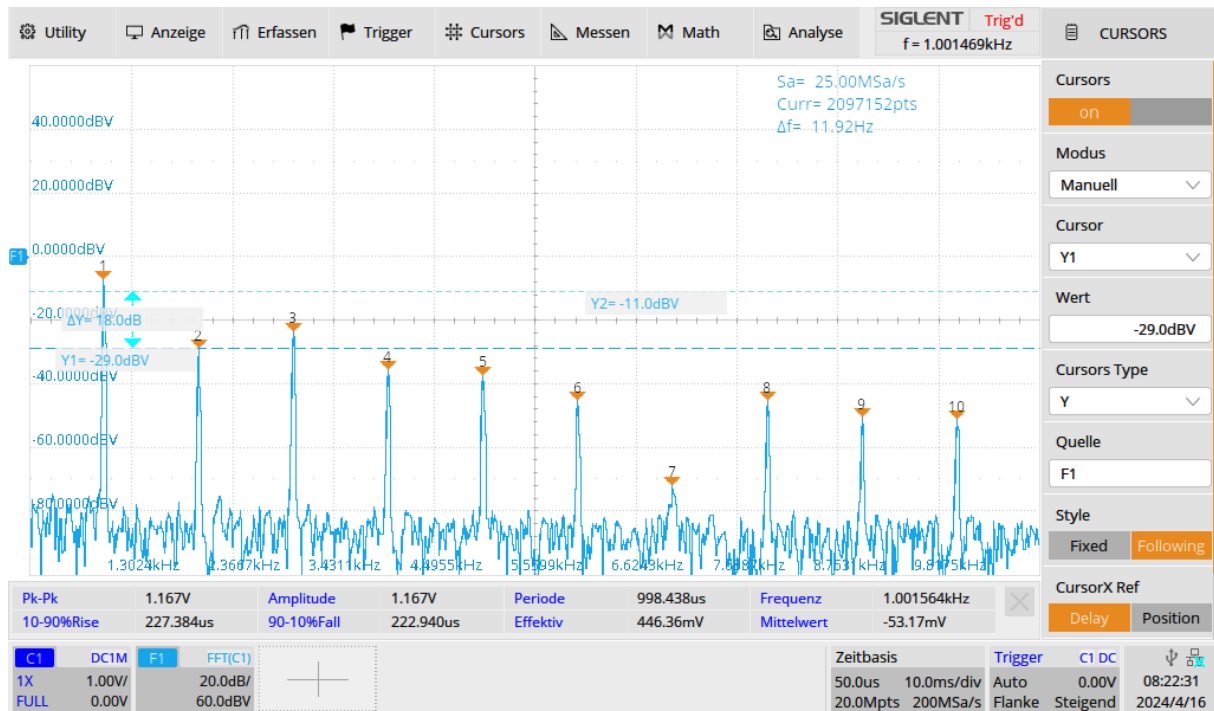
Was macht der Verstärker?

Dazu wurde am Eingang und am Ausgang gemessen.



Das Signal hatte -9dBV und bei 2kHz waren es dann -36 dBV.
Differenz: 27 dB.

Und am Ausgang des Verstärkers:



Signal -7 dB und bei 2kHz -29 dB. Differenz also 22dB.

Erkenntnis:

Der Verstärker macht beim Betrieb mit einem Signalpegel von 200mVeff eine Verzerrung von ca. 5dB. Bei größeren Eingangspegeln ist eine Übersteuerung am Oszillografen zu sehen. Line-Pegel sind mit 500mVeff spezifiziert. Da ich stets kurze Kabel zwischen Vorverstärker und Endverstärker habe, können die „Ausgangsimpedanzen“ eines sog. Vorverstärkers auch höher als 1k sein. Die üblichen 600 Ohm kommen vermutlich aus der Fernmeldetechnik. Hier wurden die Impedanzen denen der Freileitung angepasst. Und da im allg. die Pegel in Line-Verbindungen hoch sind, kann der Verlust

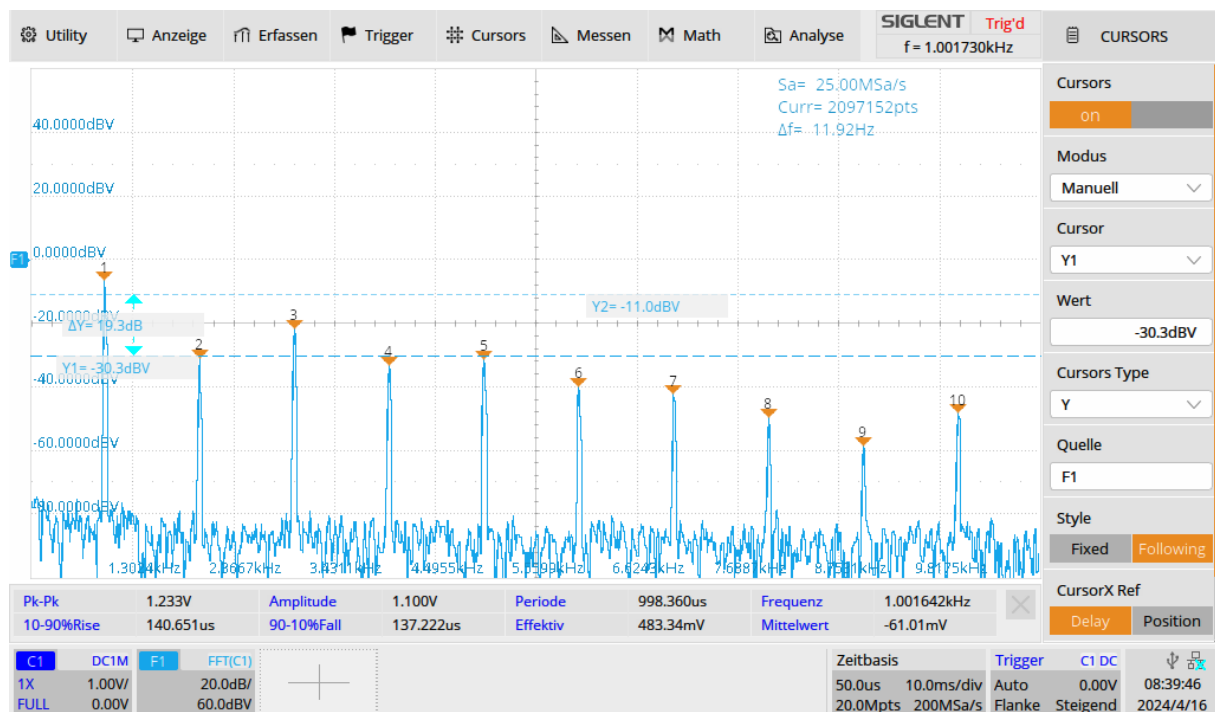
durch Spannungsteilung durch die Eingangsempfindlichkeit des Endverstärkers ausgeglichen werden. Der übliche Aussteuerungspegel von -10dB sind 316 mV. Ich könnte also durch die Potis(Spannungsteiler) für die Balance noch gut 120mV verlieren, ohne dass die Lautstärke zu sehr begrenzt wird. Die Vollverstärker(Sony) schalten ja auch das Signal vom Eingang direkt auf den Record-Ausgang.

Was macht der Verstärker ohne vorgeschalteten X9C103P?

Dazu wurde das Poti abgeklemmt. Man sieht im nächsten Bild.

Eingang: 7,7dB / 35,3 dB; D = 27,6 dB

Ausgang: 4,3 / 30,3 dB; D = 26 dB.



Erkenntnis

Eingeschlossen Ablesefehler erzeugt der Verstärker wohl kaum eine Verzerrung. Er ist für HiFi brauchbar!

Fazit

Der Verstärker erfüllt die Erwartungen. Das elektronische Poti(X9C103P) ist nur mit Bedenken anwendbar. Ich werde es durch ein Alps-Poti ersetzen. Inwieweit nun +10dB größere Verzerrungen hörbar sind, muss jeder für sich entscheiden. Der X9C103P ist im Vergleich natürlich verschleißfrei.

Frage

Warum brauche ich einen Vorverstärker? Ich betreibe keine Signalquellen, die einen zu geringen Pegel für den Endverstärker abgeben. Ich könnte ja auch die Signale direkt zum Ausgang schalten. Mein CD-Player gibt genug Pegel ab. Und so wird es vermutlich auch mit anderen Quellen sein. Für einen Plattenspieler kann man ja einen eigenen Phonoverstärker vorsehen. Auch der gibt vermutlich bis zu 1Veff heraus. Also wäre jeder zusätzliche Schaltungsaufwand eigentlich schädlich für die Signalqualität.

Die ferngesteuerte Lautstärke ist bequem und ein Feature, auf das ich nicht verzichten möchte. Aber mit einem Poti von Alps könnte man das machen. Das erzeugt keine zusätzlichen Verzerrungen. Also wäre mein Vorverstärker nur eine ferngesteuerte Einrichtung zur Lautstärkeregelung und Quellenumschaltung. OK, das leistet die Software und ich

muss sie nicht besonders anpassen. Den Teil für das elektronische Poti werde ich umschreiben für die Ansteuerung einer H-Brücke. Die beiden rückwärtigen Potis bleiben. Damit kann man eine einmalige Einstellung der Balance machen. Das Alps-Poti lässt sich auch von Hand bedienen. Also ...

Ein neues Konzept

Bisher hatte ich ja den Vollverstärker von Sony auch wegen der Quellenumschaltung betrieben. Den Signalweg über die ferngesteuerte Lautstärkereglung hatte ich nicht genutzt. Keine Verfälschungen der Tonqualität! Das Signal am Endverstärker war glasklar und der Pegel von den Recordausgängen quasi eine direkte Durchschaltung. Was mich störte, war die Lautstärkeregelung von Hand und das große Gehäuse des Vollverstärkers. Das veranlasste mich ursprünglich, über den Selbstbau nachzudenken.

ALPS-Poti

Nach einem Test war der Klang ohne den Verstärker ebenso gut wie mit ihm. Also blieb der Verstärker abgeklemmt. Ich baute ein ALPS-Poti ein. Die Steuerung erfolgte über den Baustein:

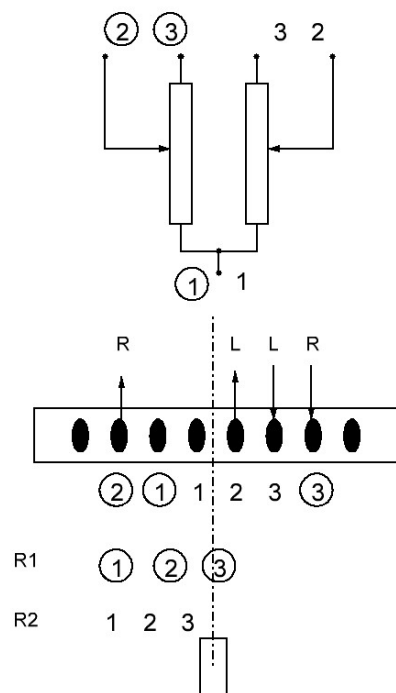
[MX1508 DC Motor Treiber Modul LN298N](#)



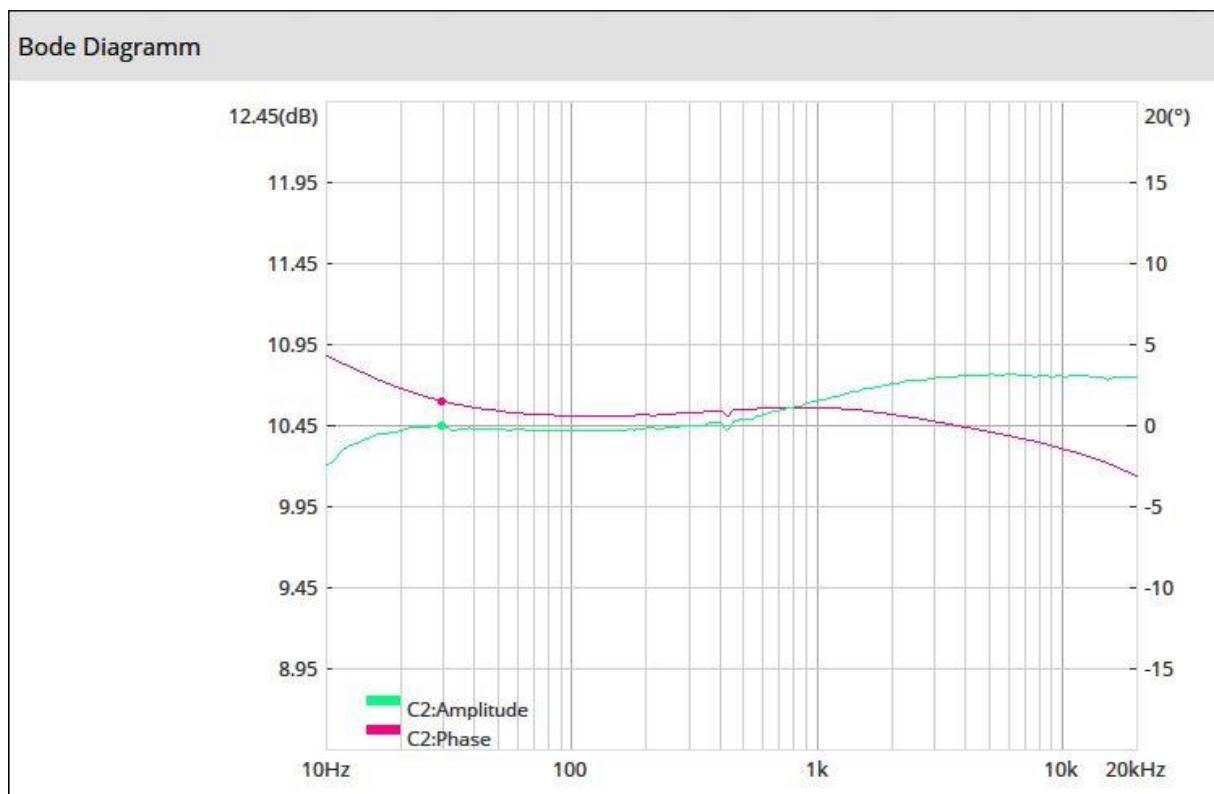
Durch Setzen auf LOW (beide IN1 und IN2) kommt der Motor zum Stillstand. Wenn man anhaltend die Taste drückt, wird über ein Delay von 200ms eine schrittweise Änderung der Lautstärke erreicht. Ein drittes Signal wird nicht benötigt. Die Software wurde angepasst. Man kann jetzt bequem und im „richtigen“ Tempo die Einstellung ändern. Nach dem Abschalten behält ja das Poti mechanisch die Stellung.

Hier noch die Belegung der Anschlusspins, die schwer im Netz zu finden ist:

ALPS 402075 RK16812MG 10KDX2 Motor-Potentiometer



Das Signal geht jetzt vom Eingang auf das Poti des Kanals und anschließend auf das Alps, an dem die Lautstärke ferngesteuert einstellbar ist. Von dort direkt zum Output. Dazwischen ist kein Verstärker. Jegliche unerwünschte Veränderungen sind also ausgeschlossen. Die Bandbreite wird durch die Koaxkabel(RG174) kaum beeinflusst. Das zeigen die Bode-Plots.

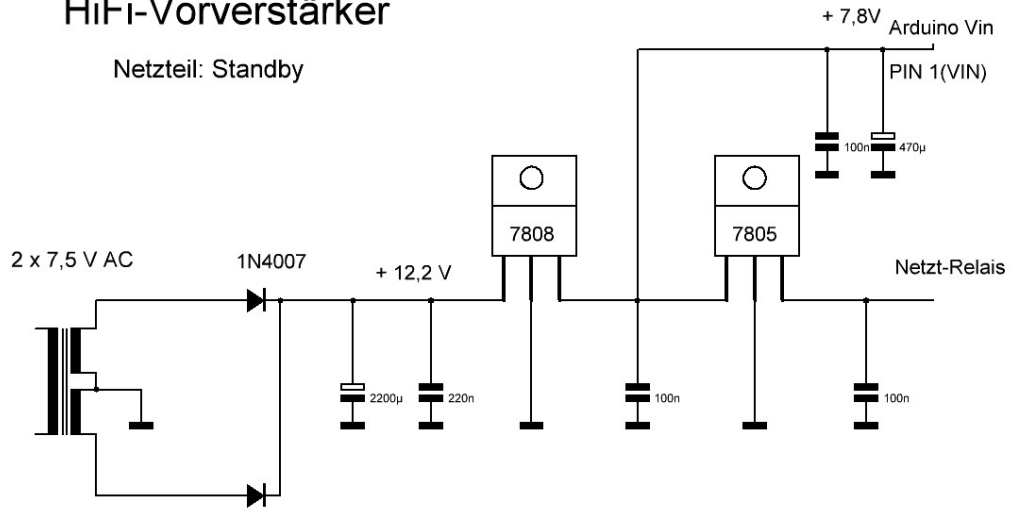


Die geringe Überhöhung nach 1kHz kommt von den drei Tastköpfen, die ich im 1:1/1M verwendete. Eigentlich ist ja nur ein Stück Draht und ein Spannungsteiler zwischen Eingang und Ausgang.

Mit Koaxkabeln(50 Ohm) zeigte sich der Bode-Plot so, wie im nächsten Bild. Das Oszilloscope war also kalibriert.

HiFi-Vorverstärker

Netzteil: Standby

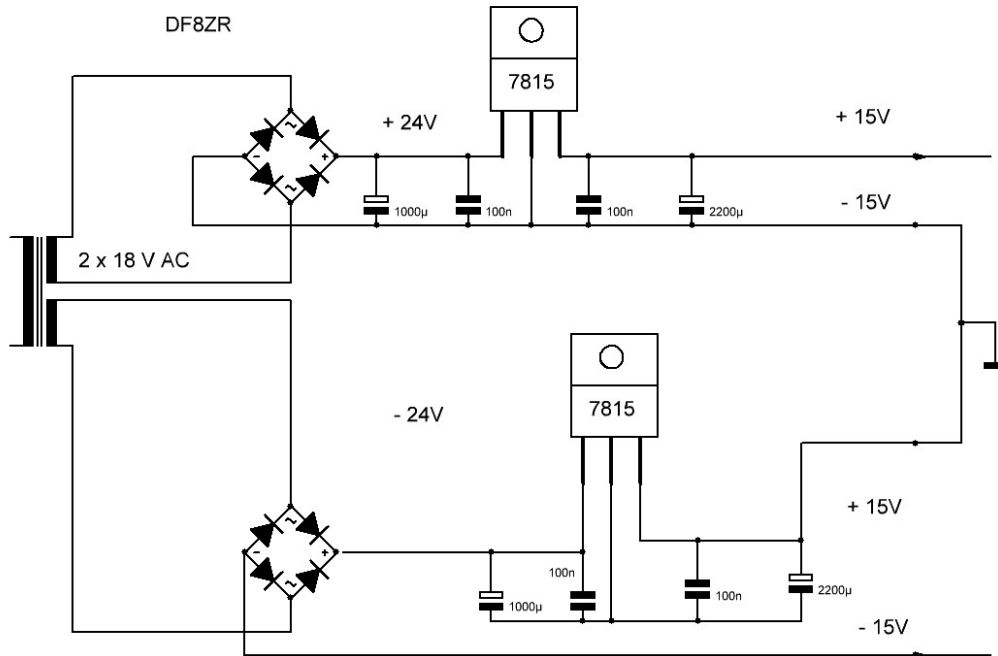


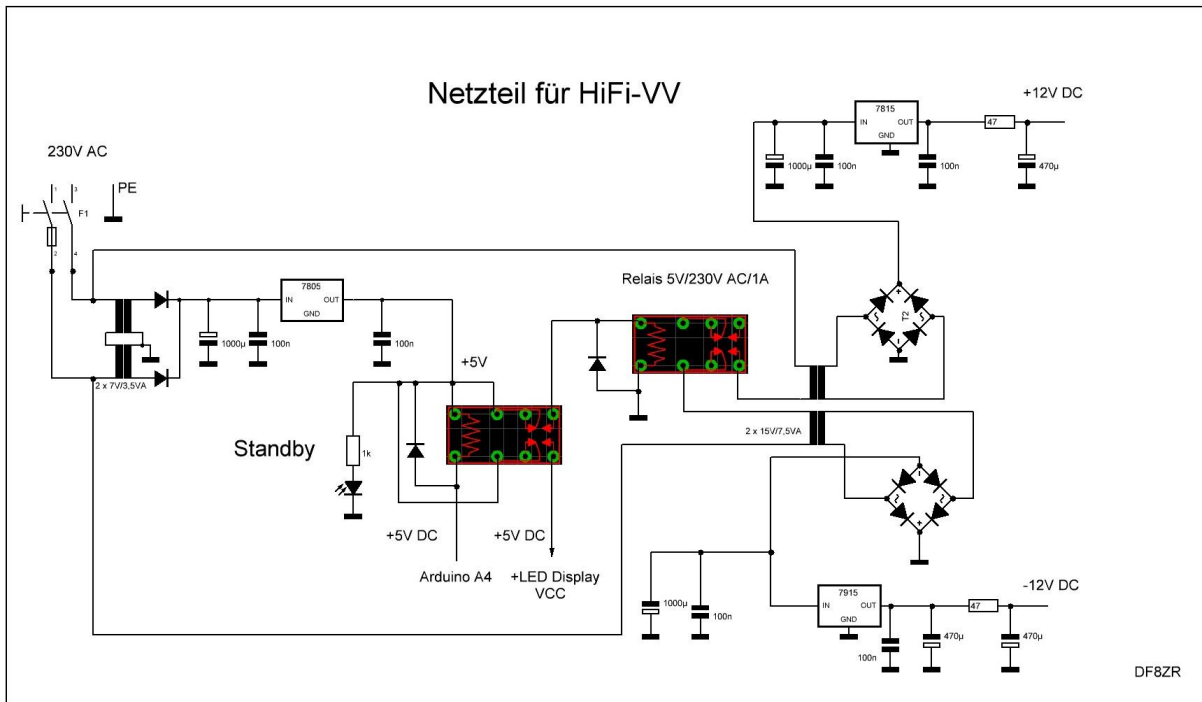
DF8ZR

HiFi-Vorverstärker

Netzteil: 2 x 15V DC

DF8ZR





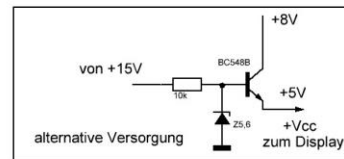
Steckerbelegung elektr. Potis:

1	2	3	4	5
sw	rt	ge		
br		or		

- 1 GND = sw GND am Arduino
- 2 U/D = br A1 am Arduino
- 3 INC = rt A2 am Arduino
- 4 CS = orange A3 am Arduino
- 5 + Vcc = ge +5V

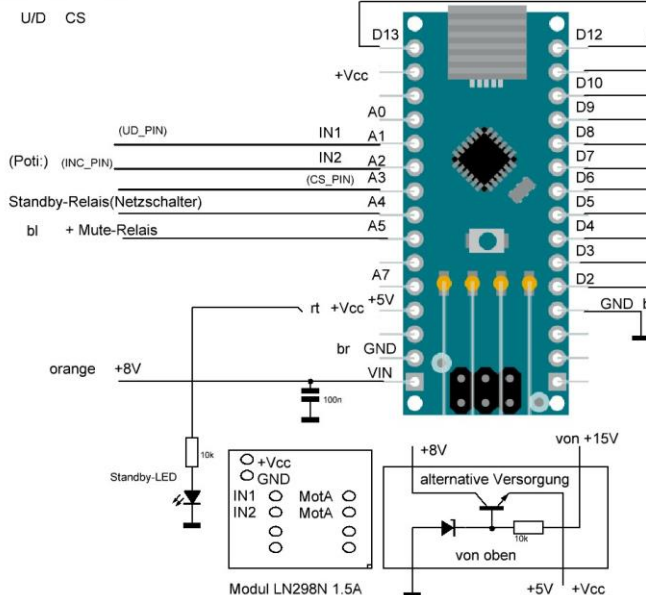
Steckerbelegung IR-Empfänger:

- 1 GND sw
- 2 +Vcc br
- 3 S rt an D12 Arduino



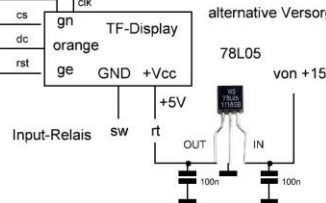
GND INC +Vcc

U/D CS



Steckerbelegung Rückseite:

siehe auch alternative Versorgung



Steckerbelegung Rückseite:

- 1 sw TV = D2 sw am Arduino
- 2 gn = DAB/WEB D3 br
- 3 rt = TUNER D4 rt
- 4 bl = CD D5 orange
- 5 w = AUX1 D6 ge
- 6 ge = AUX2 D7 gn
- 7 rt + = MUTE + A5 bl
- 8 br - = MUTE GND violett

Ich betreibe die alternative Versorgung des Nano mit einem geschalteten Transistor. Der Einsatz eines +5V-Stabis zog zu

viel Strom aus den +15V. Mit einem stärkeren Trafo ginge das aber elegant zu machen.

Software

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

// hier wird anstelle des X9C103 ein Poti von ALPS verwendet
//
//
// Old Bootloader aktivieren beim Arduino-Clone !
// benötigte Bibliotheken einbinden

#include "IRremote.h"
#include "EEPROM.h"
// Size of the EEPROM
#define EESIZE

// der Pin, an dem der Infrarot-Empfänger angeschlossen ist
int EmpfaengerPin = 12; // D12

#include <TFT.h>
#include <SPI.h>
// ST7735S
// #include <Adafruit_GFX.h> // Core graphics library
// #include <Adafruit_ST7735.h> // Hardware-specific library

// 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;
```

```

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                       ß    =pinDIL5

int GSM1 = A1;
int in1 = A2;
int in2 = A3;

float voltage = 0;
int volume;
int vadresse = 10;

//=====

//*****SETUP*****
*****
void setup()
{
  pinMode(12, INPUT);
  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(A5, OUTPUT); //für das Mute-

    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

    //Störungen durch andere Fernsteuerungen vermeiden:

    if (pos != 64 && pos != 67 && pos != 68 && pos != 70 && pos != 71 && pos
    != 8 && pos != 13 && pos != 24 && pos != 28 && pos != 82 && pos != 90 && pos
    != 22){
        pos = 68;
        posalt = pos;
    }

    switchON(); //das Relais

    pos = posalt; //nach dem Einschalten gültiger Wert
    anzeigen(); //gespeichertes pos = posalt
    //IR-Empfang starten
    //Serial.println(posalt);
    IrReceiver.begin(EmpfaengerPin); //PIN 12
    delay(100);

    //Motoransteuerung
    pinMode (GSM1, OUTPUT);
    pinMode (in1, OUTPUT);
    pinMode (in2, OUTPUT);

    //Serial.begin(9600);

    //digitalWrite(INC_PIN, HIGH);

```

```

//digitalWrite(CS_PIN, LOW); // Enable the X9C103P chip;
hier dauerhaft //könnte auch durch Verbindung
nach Masse hergestellt werden //Cs ist nicht dauerhaft LOW,
weil es zum Speichern einen positiven Impuls braucht
//Serial.print ("Initial Voltage Setting: ");
//PrintVoltage();

posalt = (int (EEPROM[IN]));

//jetzt den Poti-Chip aktivieren
/*digitalWrite(UD_PIN, LOW); delay(1);//delayMicroseconds(10);

digitalWrite(INC_PIN, HIGH);
digitalWrite(INC_PIN, LOW); delay(1);//delayMicroseconds(50); // Pulse
für den Schleifer DOWN

digitalWrite(INC_PIN, HIGH);
delay(1);
digitalWrite(UD_PIN, LOW);
digitalWrite(CS_PIN, LOW); delay(1);//delayMicroseconds(10);*/

}

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 switchOFF(int posalt){

if (posalt == 69)   gate = 2;//Taste 1; Tasten auf der Fernsteuerung
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 LautstUP()
{
digitalWrite(in2,HIGH);
digitalWrite(in1,LOW);
delay(200);
digitalWrite(in2,LOW);
analogWrite(GSM1,10);//Geschwindigkeit bis 255
  delay(10);
}

void LautstDOWN()
{
digitalWrite(in2,LOW);
digitalWrite(in1,HIGH);
delay(200);
digitalWrite(in1,LOW);
analogWrite(GSM1,10);//Geschwindigkeit bis 255
  delay(10);
}

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 && pos
!= 22 && pos !=13){
```

```
//wenn die Codes mit dem gespeicherte Code nicht übereinstimmen,
// 8 und 90 sind die waagerechten Pfeiltasten
// 24 und 82 sind die Mute-Tasten Pfeil oben und unten
// 13 und 22 sind die Netz-Ein und Netz-Aust
//muss ein anderes Relais angesteuert werden
//dazu wird zunächst das "alte" Relais abgeworfen:
```

```
switchOFF(posalt);//bisheriges Relais abschalten
```

```
switchON();//aktuelles Relais einschalten
```

```
//if (pos!=22)//verhindert, dass die *-Taste = Netzaus gespeichert ; es
gilt die pos der zuletzt gespeicherten Relais-Buchse
```

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

}

}

//Standby:

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

mute = 1;
Mute();
}

void Netzaus(){
    analogWrite(A4,0);
    mute = 0;
    Mute();

}

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
}

}
```

Hier noch eine Hilfe für die Anschaltung des Displays:

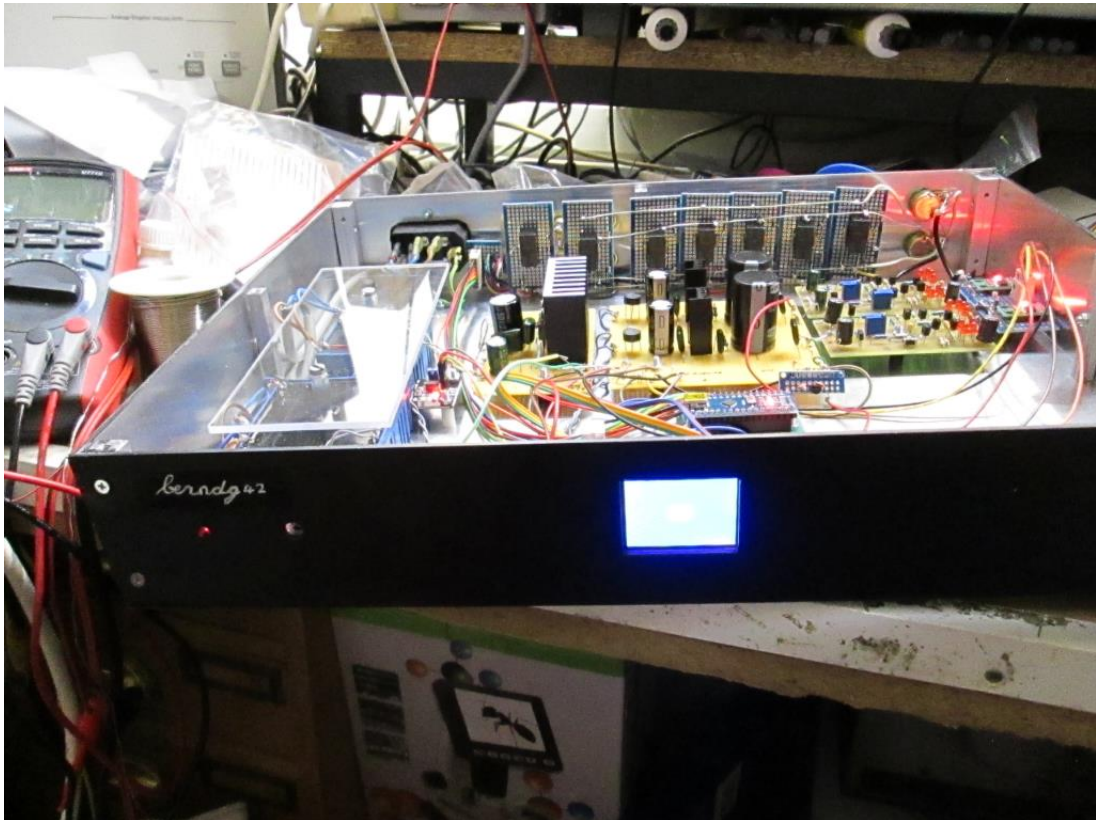
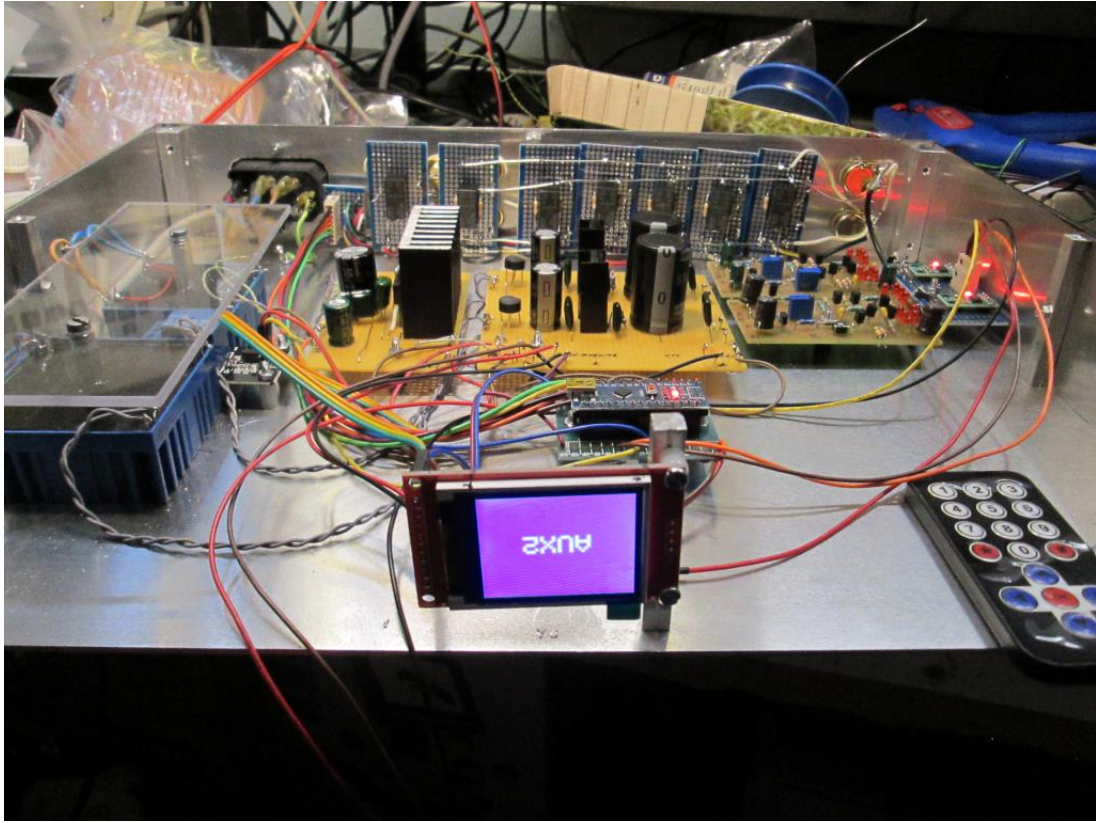
```
TFT side ----- microprocessor
- Vcc      -->   +3V3V(!!!!)
- Gnd      -->   Gnd
- CS       -->   CS pin (3v3 level!)
- RST RES  -->   connect to a MCU pin or tie to +3V3 or 10K to 3V3 (do NOT leave float!)
- A0  RS DC -->   DC or RS pin (3v3 level!)
- SDA      -->   Mosi (3v3 level!)
- SCK SCL  -->   Sclk (3v3 level!)
- LED      -->   Some display need a resistor (see note below)
```

Die Pinbezeichnungen für das St7735 sind manchmal unterschiedlich. Und auch die Displays mit 1.8“ haben 3,3V-Versorgung statt 5V. Man muss vor dem Einsatz ganz sicher sein, was vor einem liegt. Manche ST7735 müssen mit Konvertern von 5V auf 3,3V am Arduino betrieben werden!

Am beste versucht man die Inbetriebnahme mit einem Testprogramm, die es auf GitHub gibt.

Fotos

Es folgen nacheinander Bilder, die chronologisch gemacht wurden. Vielleicht werde ich einen Film bei YT einstellen.







Da ich das Material stets neu kaufte, kamen einige Euros zusammen. Aber ich schätze, dass man mit < 200 EUR

auskommen kann. Leider sind die Bauteile unglaublich teuer geworden. Und der Zuschnitt der Alu-Platten war nicht immer präzise, weshalb ich das Gehäuse nicht ganz maßgenau zusammenkriegte. Aber das fällt kaum auf. Mit Abziehbildern versuchte ich den Schriftzug links oben aufzubringen. Ging mehrfach schief. Letztlich nahm ich einen Filzstift mit weißer Farbe und schrieb den selbst. Und so ein Drehknopf für die Lautstärke kann eben mal um die 20 EUR kosten. Abzocke beim Kleinkram!

Dennoch sollte man sich nicht vom Selbstbau abschrecken lassen. Immer noch billiger als gekauft. Und überhaupt nicht schlechter in der Qualität.

Viel Spaß beim Nachbau!

DF8ZR; im April 2024

Film auf YouTube:

<https://www.youtube.com/watch?v=vOf0WSBbPUs>

Meine E-Mail: bernd.grupe@t-online.de