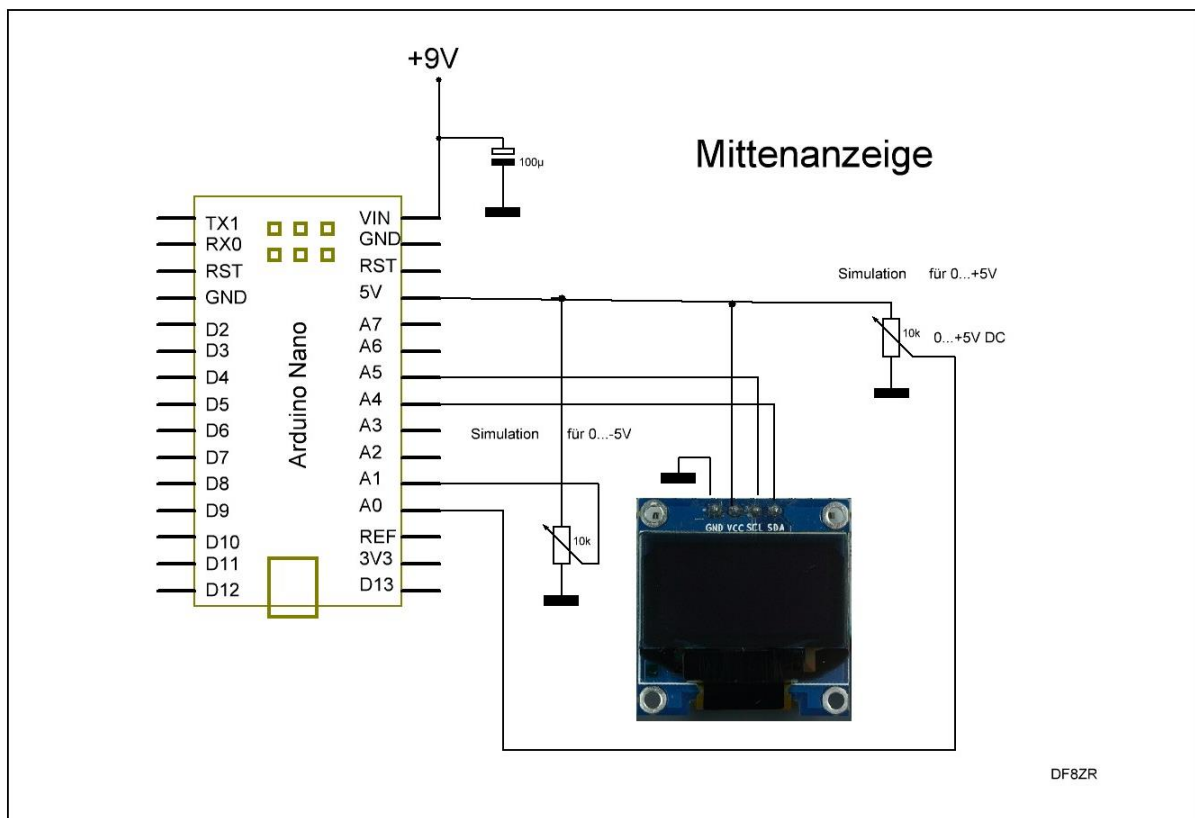


REVOX A76: Ersatz für Abstimmmanzeige

Mittenanzeige

Im Originalgerät sind Drehspulinstrumente. Leider gehen die eines Tages kaputt. Der Magnet klebt nicht mehr in der korrekten Position und verklemmt den Zeiger. Eine Reparatur ist fast aussichtslos. Hier wird nun ein erster Entwurf eines Ersatzes vorgestellt. Ein längliches OLED-Display wird von einem Arduino angesteuert.



Die Spannung am Demodulator wechselt von -2,5V bis + 2,5V. Im Modell wird sie mit zwei Potis simuliert. Event. muss man die Spannungen vom Demodulator durch einen Verstärker anpassen(OP). Hier ist ein 128 x 64 OLED eingezeichnet. Später wird das durch ein OLED 128 x 32 ausgetauscht. Die Software ist dann angepasst. Software

```
//die Mess-Spannung um den Mittelpunkt schwankt zwischen -2V nach +2V
//die negative Spannung wird mit einem OP ins Positive gedreht
//für beide Polaritäten wird je ein Messeingang bereitgestellt
//außerhalb dieser Messspannungen ist ein negativer Pegel um -30mV, also < 0V
```

```
#include <Arduino.h>
#include "U8g2lib.h"
#include <SPI.h>
#include <Wire.h>
//U8G2_SSD1312_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ SCL, /*
data=*/ SDA, /* reset=*/ 8);
//U8G2_SSD1306_128X64_NONAME_1_HW_I2C oled(U8G2_R0);
U8G2_SH1106_128X64_NONAME_1_HW_I2C oled(U8G2_R0); // es entsteht rechts ein
Randeffekt, der stört,
//wenn der Konstruktor SSD1306 verwendet wird
```

```
int position = 0;
int positionalt = 0;
int mo;
int mu;
```

```
void setup(void) {
  //Serial.begin(9600);

  oled.begin();

  mo = 0;
  mu = 0;
  balken();//Start ohne Bedingung
}
```

```
void loop(void) {

  mo = analogRead(A0);//für die positiven Pegel
  mu = analogRead(A1);// für die negativen Pegel

  if (mo >=500){
    mo = 500;}

  if (mu >=500){
    mu = 500;
  }

  if (mo >= 0 & mo <= 500) {
```

```

    position = mo ;// / 12;
    position = position + 62;//Zeiger bewegt sich nach rechts in den positiven
Bereich, Mittenlage + Pegel//62
}
if (position >= 120){
    position = 120;
}
if (mu >= 0 & mo <= 5) { // negativer Pegel am mu und Pegel an mo fast 0V,
Zeiger geht nach links
    position = mu ; // / 12;//8

    position = 62 - position;//118
}

```

```

if (positionalt <= 0)
    positionalt = 8;
if (positionalt >=120)
    positionalt = 120;

```

```

if (positionalt != position) { // nur bei Änderung des Pegels den neuen
Balken anzeigen
    balken();
}

```

```

//delay(500);
}

```

```

void balken() {

```

```

    /*Serial.print("position =");
    Serial.println(position);
    Serial.println("positionalt: ");
    Serial.println(positionalt);

    Serial.println("mo = ");
    Serial.println(mo);
    Serial.println("mu = ");
    Serial.println(mu);
    //Serial.print("\t");
    Serial.println("-----");
    */

```

```

if (position > 120) // rechtes Ende

```

```

    position = 120;
    if (position <= 4) // linkes Ende
        position = 8;

    oled.setDrawColor(1);
    oled.firstPage();
    do {

        oled.drawFrame(2, 2, 126, 60); //Rahmen zeichnen
        //-----

        oled.drawTriangle(72, 8, 72, 28, 120, 20); //rechtes Dreieck
        //oled.drawTriangle(int16_t x0, int16_t y0, int16_t x1, int16_t y1,
        int16_t x2, int16_t y2)

        oled.drawTriangle(8, 20, 56, 8, 56, 28); //linkes Dreieck

        oled.drawBox(position, 36, 5, 12); //Zeiger
        //oled.drawLine(u8g2_uint_t x1,
        u8g2_uint_t y1, u8g2_uint_t x2, u8g2_uint_t y2)

        oled.drawLine(8, 41, 120, 41); //Strich

    } while (oled.nextPage());

    positionalt = position;
}

```

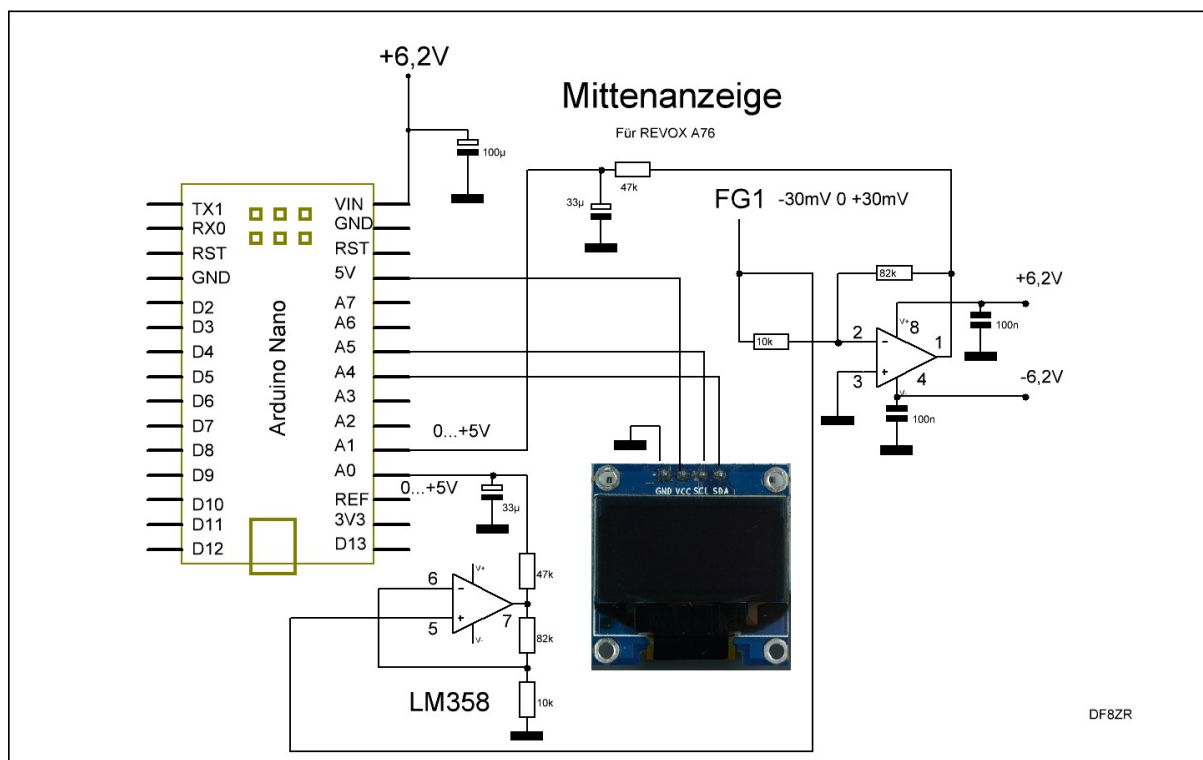
Spannungen

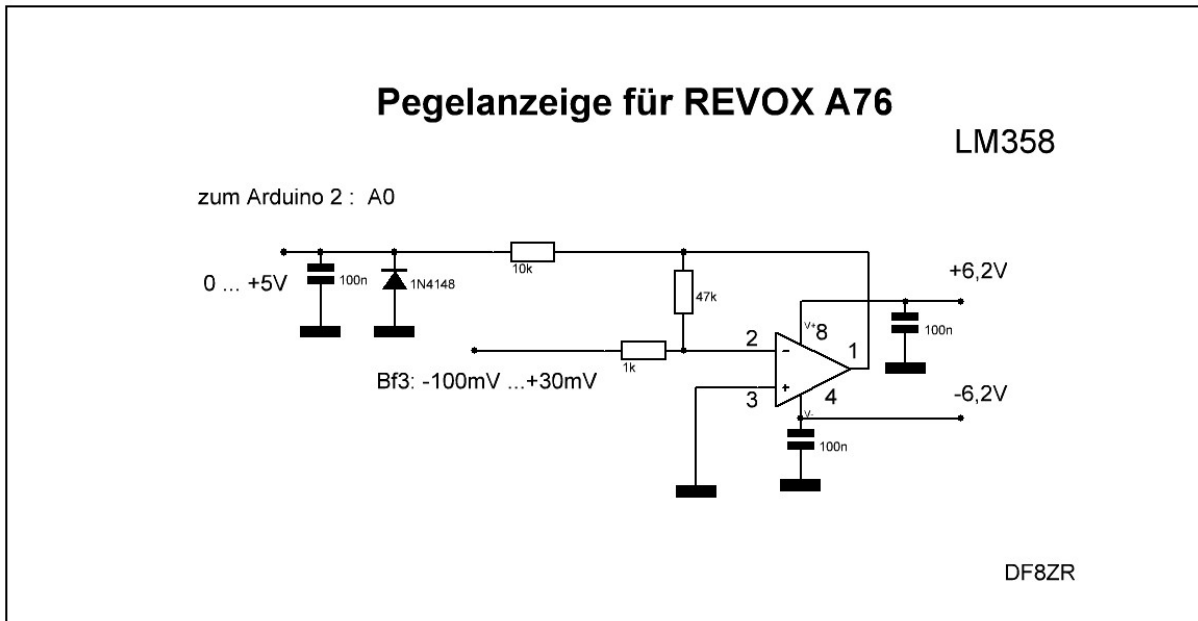
In der Originalschaltung fließt ein Strom über das Drehspulinstrument. Der Innenwiderstand ist ca. 1k. Ich habe im Abstimmbereich -30mV ...+30mV gemessen. Der Arduino soll aber +5V max. erhalten. Also muss man die Spannungen verstärken. Der OP macht das für beide Spannungsrichtungen. Wobei die Minusspannung um 180 Grad gedreht wird und auf den zweiten Input(A1) geht. Ist die

Abstimmung korrekt, wird 0V erzeugt. Der Anzeigebalken ist dann genau in der Mitte.

Eine einfache Schaltung sollte genügen. Falls Probleme auftreten, kann man mit Trimpotis die Pegel anpassen. Und natürlich muss das Board so montiert werden, dass es von der Empfängerschaltung gut abgeschirmt wird. Denn sonst könnten Störimpulse lästig werden.

Die Pegelanzeige ist ähnlich aufgebaut. Hier kommt der Optokoppler als Fotoelement zum Einsatz. Er ist ein Strom-/Spannungswandler.





Und nachstehend die Software für die Pegelanzeige. Sie wird mit einem zweiten Arduino Nano gemacht.

```
//Pegelanzeige für 1,3 Zoll-Displays
//grafik-SH1106
//size = Größe des Speichers für die Mittelwertbildung
//Summe aller Pegelwert
//-Pakete so wählen, dass die Reaktionszeit nicht zu träge wird
// im Original A76 bildet der zeiger des Drehspulinstrumentes den Mittelwert
//ohne Mittelwerterrechnung zittert die Anzeige zu stark, was störend ist
//_____
```

```
#include <Arduino.h>
#include "U8g2lib.h"
#include <SPI.h>
#include <Wire.h>
//U8G2_SSD1312_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ SCL, /*
data=*/ SDA, /* reset=*/ 8);
//U8G2_SSD1306_128X64_NONAME_1_HW_I2C oled(U8G2_R0);
U8G2_SH1106_128X64_NONAME_1_HW_I2C oled(U8G2_R0); // es entsteht rechts ein
Randeffekt, der stört,
//wenn der Konstruktor SSD1306 verwendet wird

int x = 0;
int wert = 0;
int wertalt = 0;

unsigned long millisOld;
int analogValue;
long getAv;
```

```

int singleValues[500]; //Größe des Arrayes

void setup(void) {
  // Serial.begin(9600);

  pinMode(A0, INPUT);
  // pinMode(A1, INPUT);

  oled.begin();
}

int getAnalogValue() { //Pegel einlesen
  int value;
  value = analogRead(A0);
  return value;
}

long getAverage(long value, int size){ // Mittelwert bilden
long sum = 0;
long tmpSum = 0;
long average = 0;

singleValues[size - 1] = value;
for (int i=0; i < size; i++) {
  sum = sum + singleValues[i];
}

  average = sum / size;

  for (int i = 0; i < size; i++){
    singleValues[i - 1] = singleValues[i];
  }

  return average;
}

void loop(void) {

if (millis() > millisOld + 5){

  analogValue = getAnalogValue();
  getAv = getAverage(analogValue, 10); //aus 10 Pegelwerten den Mittelwert
  bilden
}
}

```

```

wert = getAv;
wert = wert / 6;

if (wert < 4)
    wert = 4;

if (wert >= 118)
    wert = 118;

if (wert != wertalt)
    balken();

    millisOld = millis();

}

}

void balken() {

    if (wert < 122) {

        // Serial.print("wert =");
        // Serial.print(wert);
        // Serial.print("\t");

        oled.setDrawColor(1);
        oled.firstPage();
        do {

            oled.drawFrame(2, 2, 126, 60);

            oled.drawBox(6, 12, 2, 8);
            oled.drawBox(26, 12, 2, 8);
            oled.drawBox(46, 12, 2, 8);
            oled.drawBox(66, 12, 2, 8);
            oled.drawBox(86, 12, 2, 8);
            oled.drawBox(106, 12, 2, 8);

            oled.drawBox(6, 26, wert, 22);

```



```
    } while (oled.nextPage());  
  
    wertalt = wert;  
  }  
}
```

Erzeugung der symmetrischen Betriebsspannungen

Aus dem Netzteil des A76 stehen +/- 23V zur Verfügung. Mit diesen hohen Spannungen wollte ich die OPs nicht betreiben. +/- 8V sollten genügen. Und die Arduinos sollen mit +8V versorgt werden.

Regler

Die positive Spannung erzeugt ein 7808. Die negative Spannung von -8V wollte ich ebenso mit einem 7908 machen. Dabei stellte sich heraus, dass diese bei Ebay erstandenen Regler bereits bei -15V statt -8V aber -13V herausgeben. Die Spezifikationen laut Datenblatt lassen max. -35V Input zu. Bei -23V geben aber meine Regler -14V ab. Bleibe ich mit einer Inputspannung unter -12V, dann zeigen sich stabile -8V am Ausgang. Ich vermute, dass es sich um gefakte Bausteine handelt. Ein Regler 7908 von TI zeigt diesen Mangel nicht. Also Vorsicht beim Einkauf!

DF8ZR; im Januar 2024