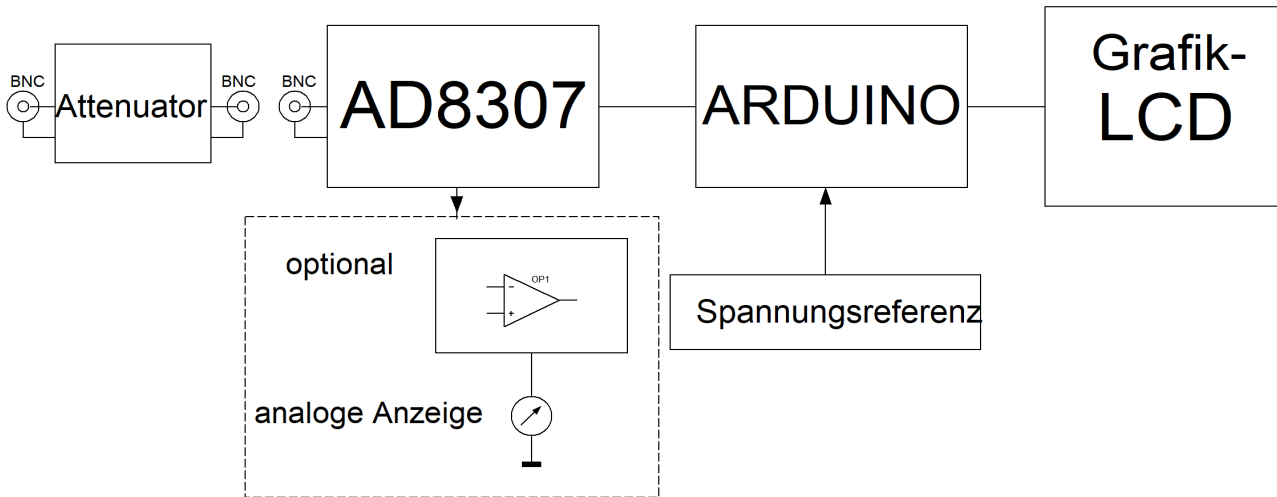


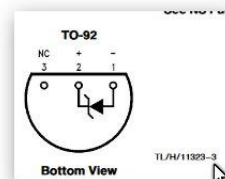
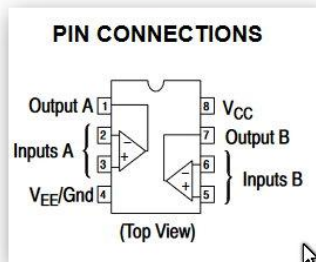
Projekt-Zusammenfassung: Aufbau Millivoltmeter mit dem AD8307 und Arduino im H39 Workshop (DL6OAA)

Prinzip: HF-Millivolt-Meter mit dem AD8307



Connections & Bauteile:

PIN	symbol
1	GND
2	VCC
3	SCK
4	SDA
5	RES
6	RS
7	CS
8	LEDA
9	LEDB
10	GT_CS
11	GT_SI
12	GT_SO
13	GT_SCL
14	GND



Die Schaltung:

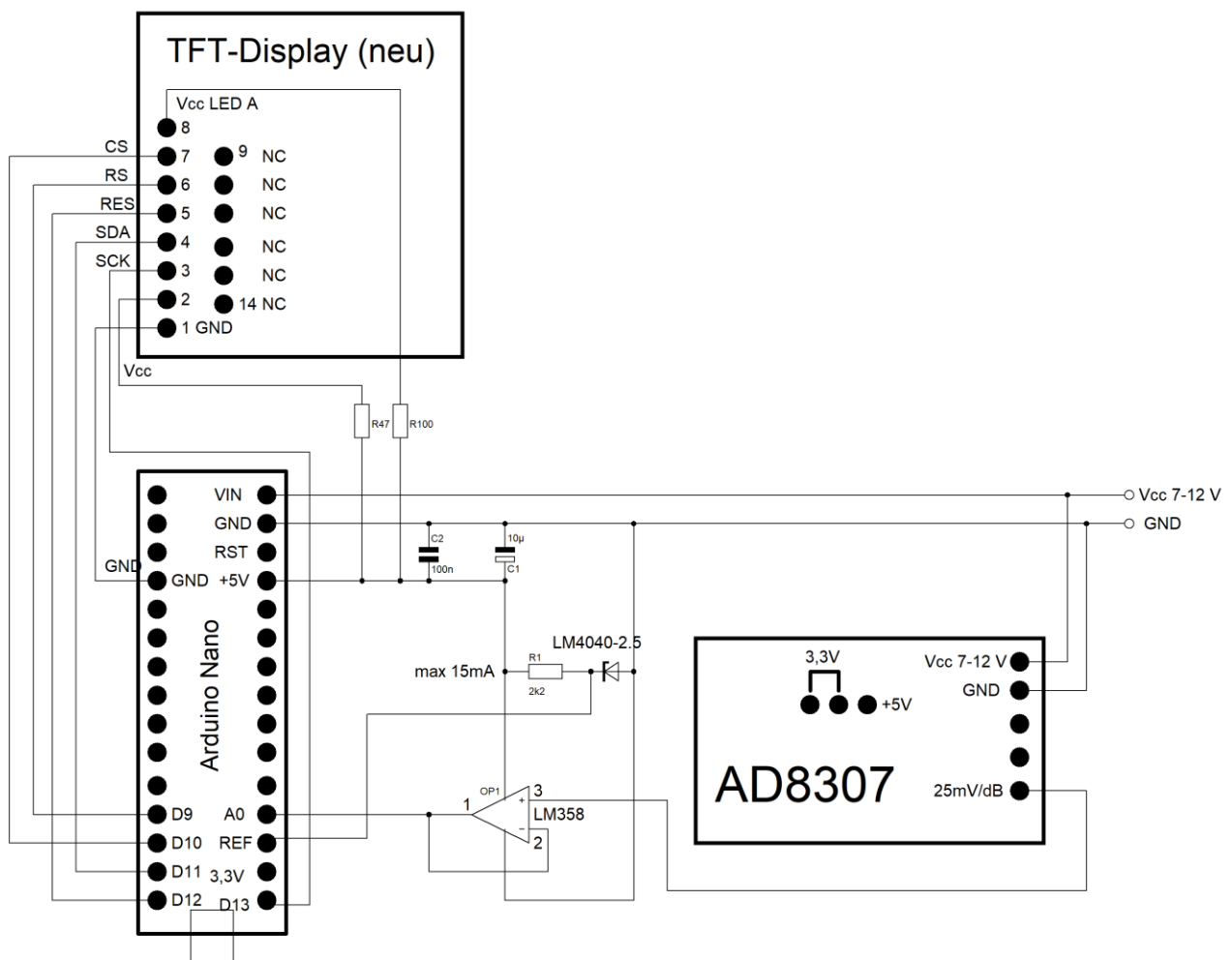
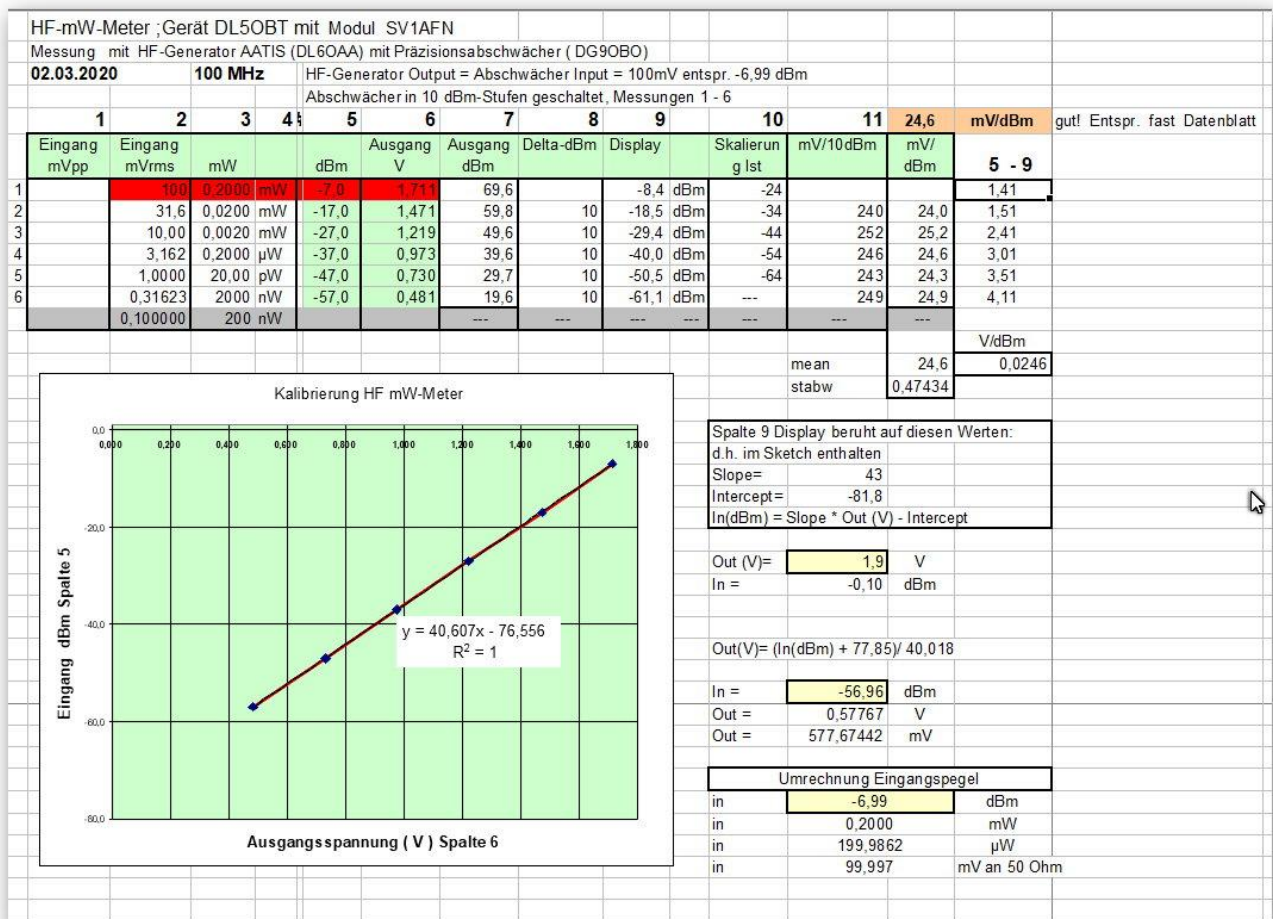
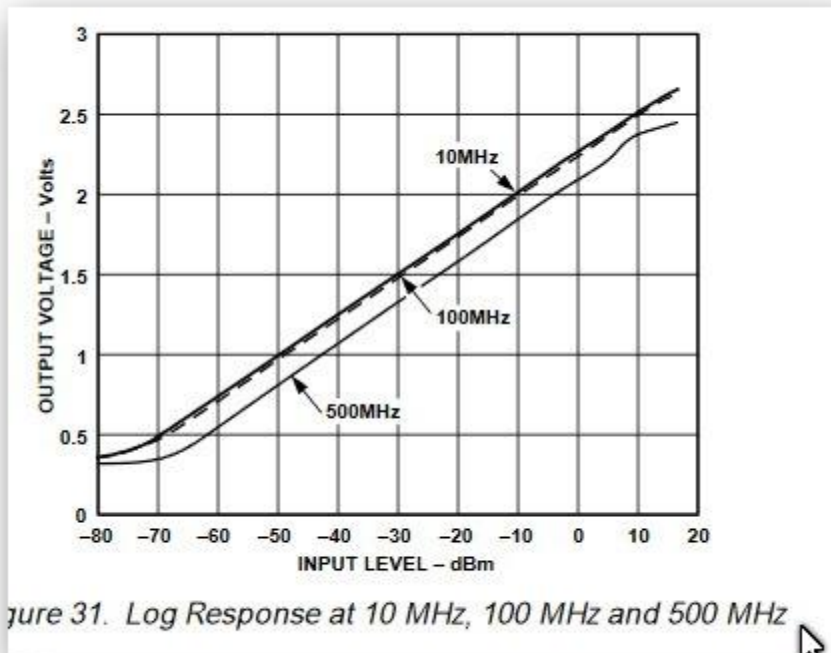


Tabelle: Zuordnung der Leitungsverbindungen:

Zuordnungen				
	Display neu			
Arduino	TFT (Pin)	LM358	AD8307	LM4040
GND	GND - 1	Pin4	GND	
5V	Vcc via R47 - 2	Pin8	5V	Kathode
A0		Pin1+Pin2		
Ref				Anode
D13	SLK - 3			
D12	RES - 5			
D11	SDA - 4			
D10	CS - 7			
D9	RS - 6			
5V	Vcc via R100 - 8			
	LED A			
		Pin3	25mV/dB	
			Jumper 3,3V	





Funktion Ausgangsspannung AD8307 (25mV/dB) und Eingangs-Level (dBm) lt. Datenblatt:

Weitere Messungen (DL6OAA):

Messung der Ausgangsspannungen am AD8307 mit dem Fluke-Digitalmultimeter (Genauigkeit 0,025%) im Vergleich mit den Rohwerten (Arduino).

Vergleich Ausgangsspannung am AD8307 ermittelt mit Arduino
vs Fluke 8600A

Signalgenerator AATiS Signalgenerator 10 MHz, **digitaler**
Abschwächer

Fluke 8600A Digitalmultimeter, Genauigkeit 0,025%

AD8303 out				
Arduino (mV)	Fluke (mV)	Dämpfung (dB)	Diff A:F (mV)	Diff (%)
1746	1752	0	6	0,34
1624	1629	-5	5	0,31
1503	1507	-10	4	0,27
1384	1388	-15	4	0,29
1255	1259	-20	4	0,32
1135	1140	-25	5	0,44
1013	1017	-30	4	0,39
888	891	-35	3	0,34
834	837	-40	3	0,36
		AVG	4,22	0,34

Vergleich Ausgangsspannung am AD8307 ermittelt mit Arduino
vs Fluke 8600A

Signalgenerator AATiS Signalgenerator 10 MHz, **analoger**
Abschwächer

Fluke 8600A Digitalmultimeter, Genauigkeit 0,025%

AD8303 out				
Aduino (mV)	Fluke (mV)	Dämpfung (dB)	Diff A:F (mV)	Diff (%)
1747	1753	0	6	0,34
1624	1629	-5	5	0,31
1498	1503	-10	5	0,33
1379	1383	-15	4	0,29
1294	1298	-20	4	0,31
1170	1174	-25	4	0,34
1048	1050	-30	2	0,19
920	922	-35	2	0,22
839	842	-40	3	0,36
		AVG	3,33	0,26

Kalibriermessungen am 9.März 2020 (im H39 Workshop):

Bei der letzten Kalibrierung gab es ja "Ausreißer" schon bei -10dBm, d.h. die Ausgangsspannung reichte nicht aus, diesen dB-Wert korrekt darzustellen. Peters Lösung liegt nun darin, die Steigung der Geradengleichung (Slope) von 25mV/dBm auf 16mV/dBm zu senken, indem er dem internen 12,5k-Widerstand einen 22k-Shunt parallel schaltet (ergibt rund 8k). Mit einem Slope von 16mV/dBm ergibt das sehr brauchbare Werte .

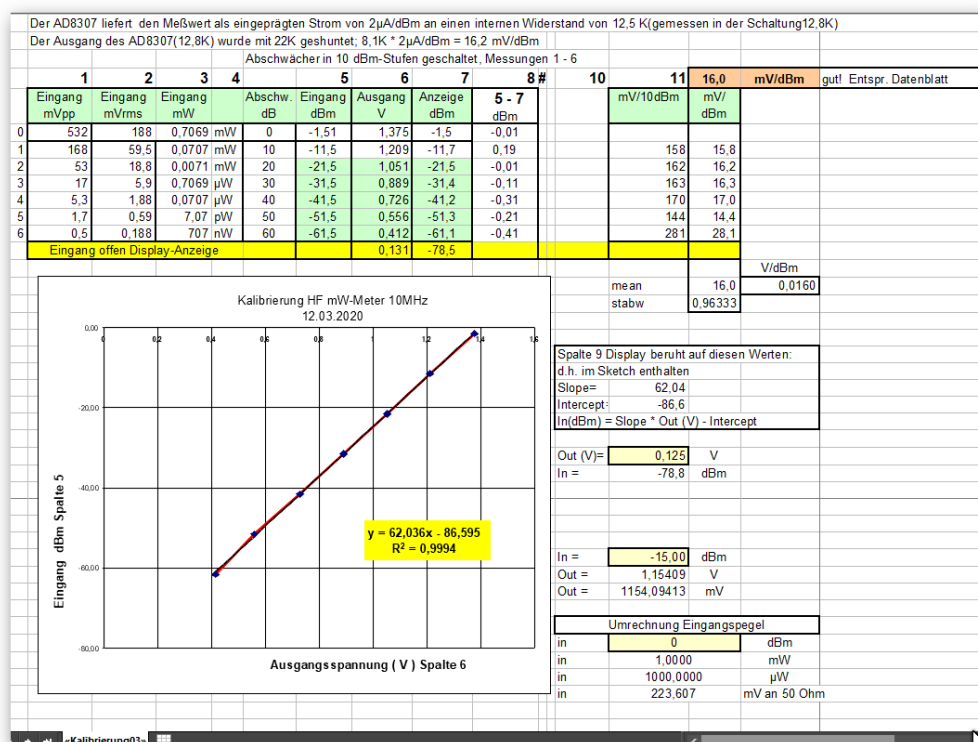
Der AD8307 liefert den Meßwert als eingepprägten Strom von 2µA/dBm an einen internen Widerstand von 12,5 K(gemessen in der Schaltung 12,8K)

*Der Ausgang des AD8307(12,8K) wurde mit 22K geshuntet; 8,1K
 * 2µA/dBm = 16,2 mV/dBm*

Es lohnt sich sicherlich, diese neue Einstellung nochmals mit einem präzise einstellbaren HF-Generator zu überprüfen.

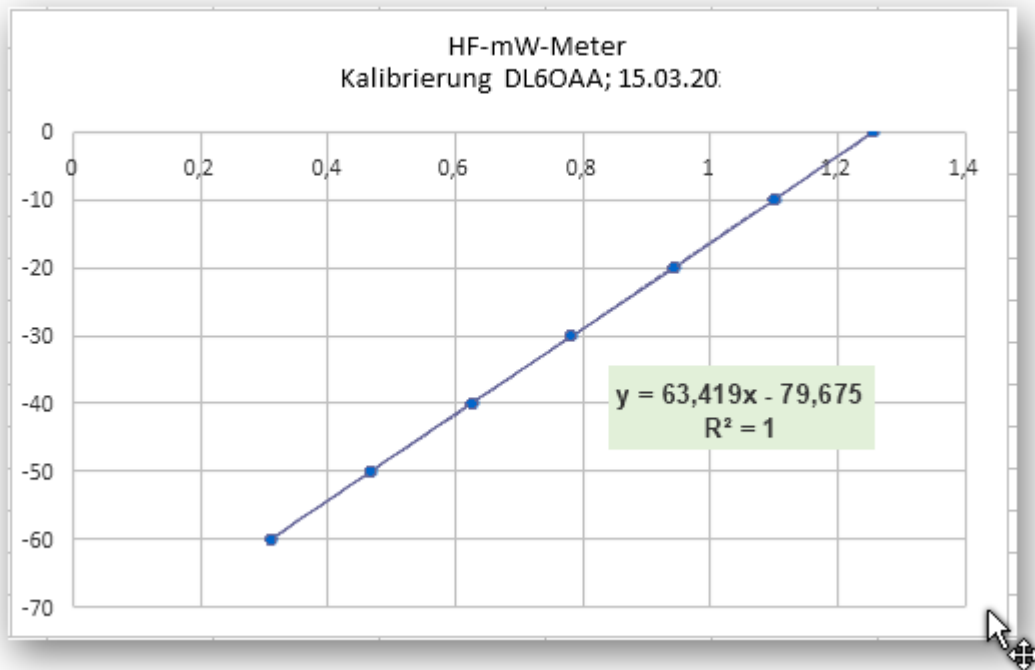
Kalibriermessung (DL6OAA) am 15.03.2020:

Kalibrierung 15.03.2020	DL6OAA						
Slope aus Diagramm entnommen: 63,96							
Intercept: 80,3							
Signalquelle: 0dB (630mV) Tektronix Sine Wave Generator / 10MHz							
	1	2	3	4	5	5 / 1	4 / 2
in (mV)pp	Ueff (mV)	ATT (dB)	ADC out (mV)	Anzeige (dBm)	Anzeige (mV)		
630	224	0	1,255	0	222,8	0,995	1
	70,7	-10	1,1	-9,9	71,168	1,007	0,990
	22,4	-20	0,943	-20	22,397	1,000	1,000
	7,07	-30	0,781	-30,3	6,794	0,961	1,010
	2,24	-40	0,626	-40,2	2,186	0,976	1,005
	0,707	-50	0,467	-50,4	0,673	0,952	1,008
	0,224	-60	0,311	-60,3	0,215	0,960	1,005



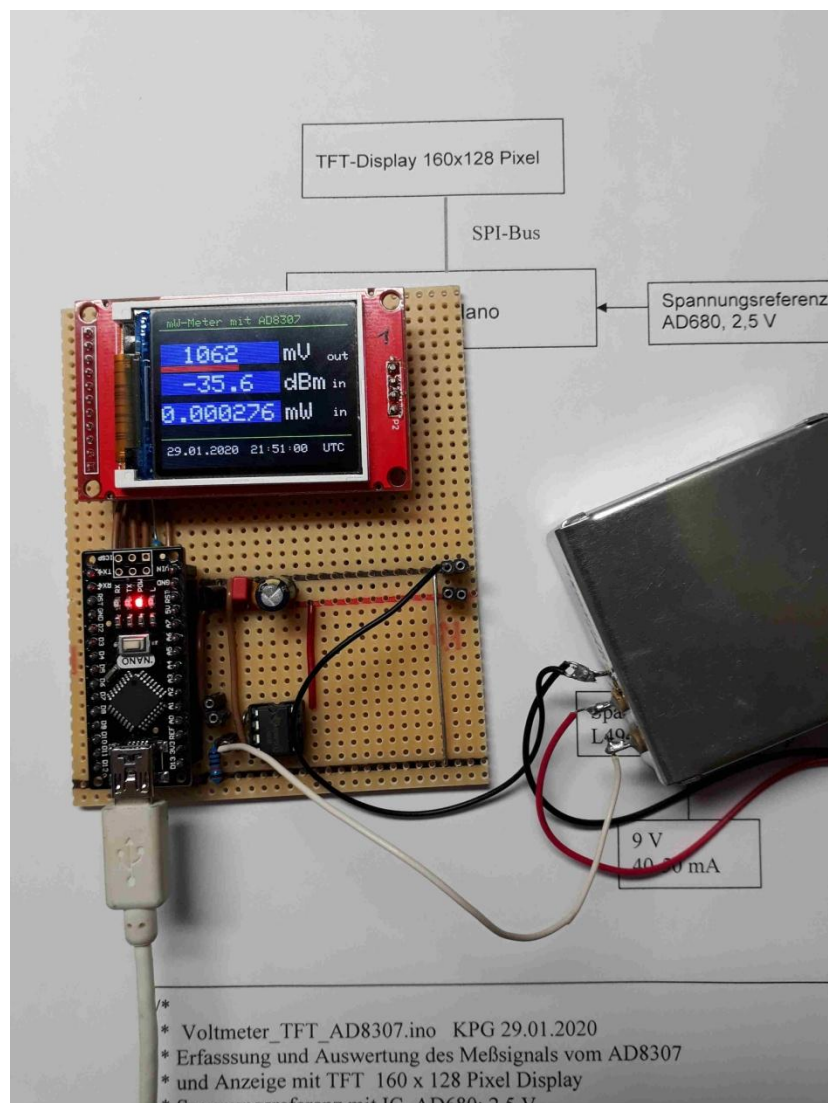
Slope= 63,42
Intercept= -79,68

Mittel:	0,979	1,003
Stabw:	0,022	0,007

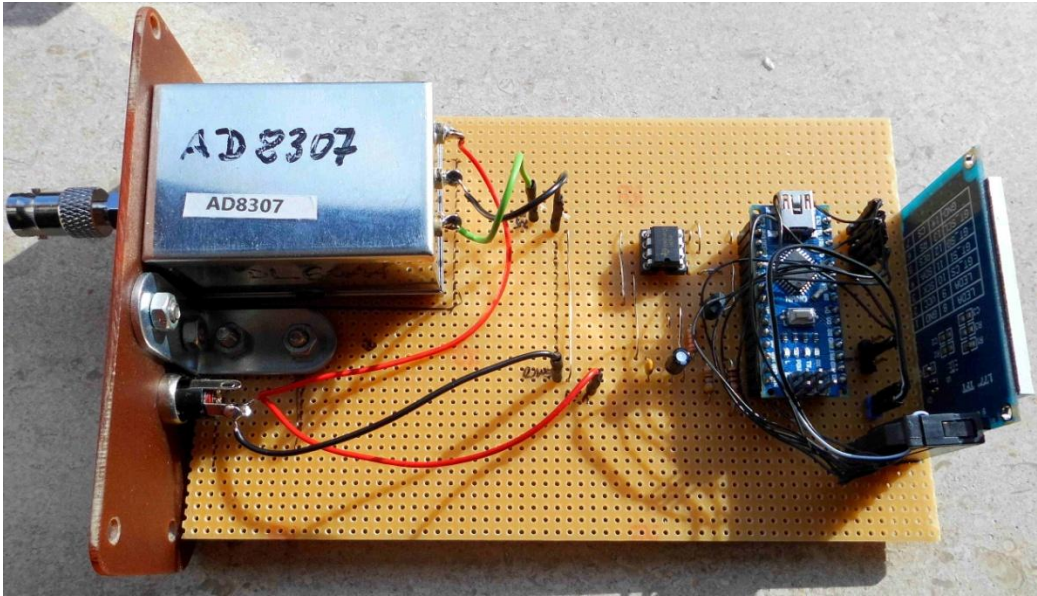


Es gilt (DL5OBT): Die einfache Übertragbarkeit auf ein anderes Gerät ist prinzipiell nicht zulässig, sie ist je nach Genauigkeitsansprüchen zu prüfen.

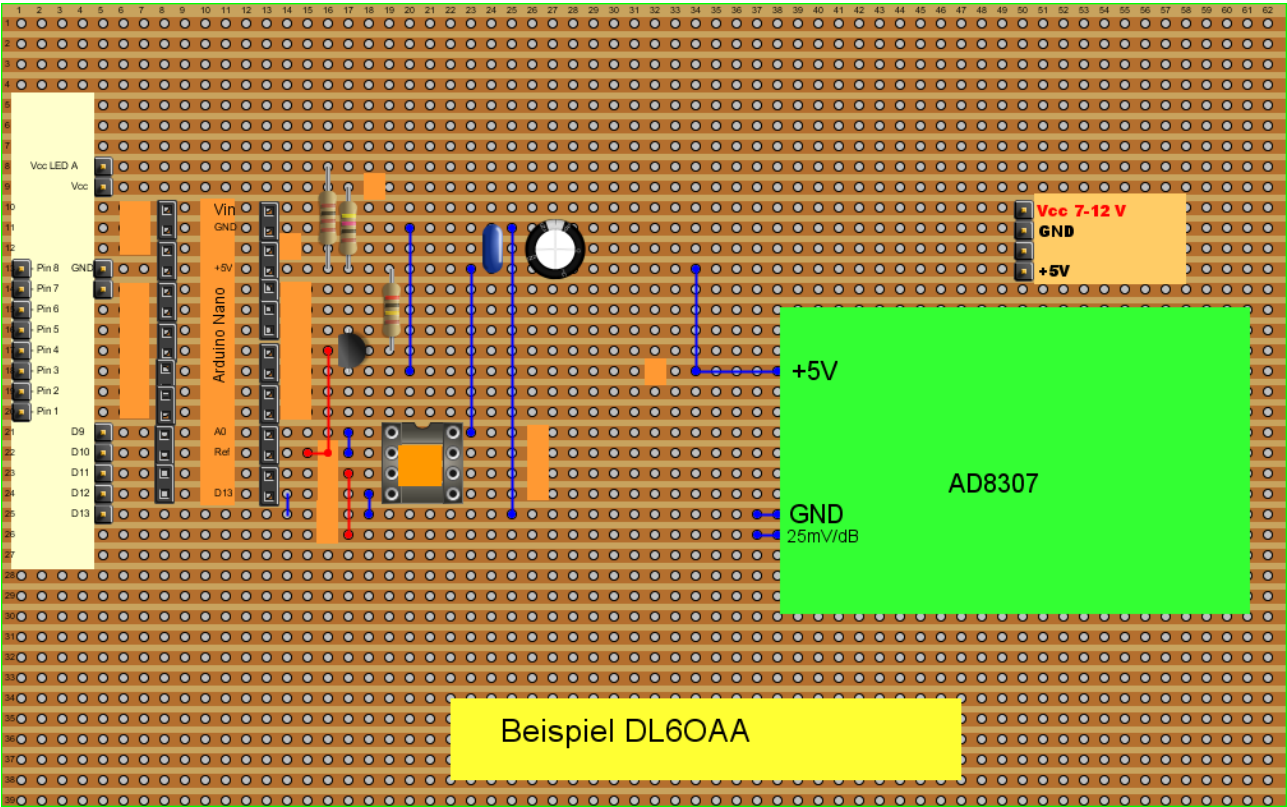
Aufbauvarianten:



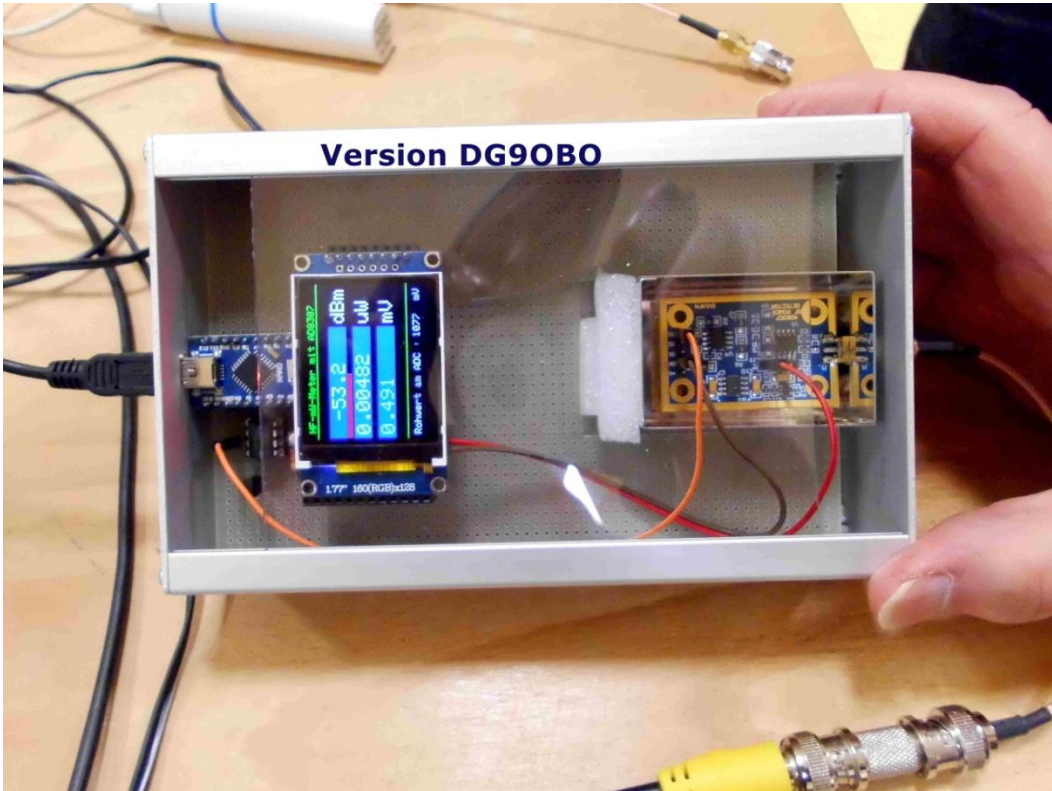
DL60AA



(Beispiel) Layout DL60AA



Version DG9OBO:



Version DL4ZIP



Display mit der Anzeige der verwendeten Kurvensteigung (Slope) und dem Intercept (Y-Achsenabschnitt); (Kalibrierung am 15.3.20; DL6OAA)



Version DB5GSS:



Aktueller Arduino-Sketch (DL5OBT):

```
/* DL5OBT 12.03.2020; 24.03.20; 06.04.2020/19.08.2020
 * mWmeter_AD8307_TFT.ino Erfassung und Auswertung des Meßsignals mit Log-
Verstärker AD8307 und Arduino-Nano
 * Anzeige mit TFT 1,77" 160 x 128 Pixel Display
 * Spannungsreferenz z.B. mit IC AD680; 2,5 V o. LM4040-2.5
 *
AZ Delivery TFT ---> Arduino
VCC ---> 470Ohm ----> 5V
SCK ---> 13
SDA ---> 11
RES ---> 12
RS ---> 9
CS ---> 10
LEDA --> 100 Ohm --> +VCC 5V

*/

#include <SPI.h>
// #include <Adafruit_GFX.h> // Adafruit Grafik-Bibliothek
#include <Adafruit_ST7735.h> // Adafruit ST7735-Bibliothek

#define TFT_PIN_CS 10
#define TFT_PIN_RS 9
#define TFT_PIN_RST 12
#define Sensor1 A0

// ST7735-Bibliothek Setup
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_PIN_CS, TFT_PIN_RS, TFT_PIN_RST);

/* *****K A L I B R I E R D A T E N *****

Zwischen der Meßspannung am AD8307 und dem Leistungsverhältnis in dBm am 50
Ohm Eingang
besteht ein linearer Zusammenhang: Leistungsverhältnis = f(Umess) ;
Die Geradengleichung lautet: dBm = slope * Umess + intercept
Kalibriermessungen mittels HF-Quelle und Abschwächer z.B an drei Meßpunkten
bei 40mV rms = -15 dBm dBm, 10 mVrms = -27 dBm und 1mVrms = -47dBm
ergeben nach Ausgleichsrechnung z.B. folgende Geradengleichung : Leistungsverh. in
dBm = 0,04 * Umess -88.2
in Zeilen 43 u. 44 können Daten einer individuellen aktuellen Kalibrierung eingetragen
werden
*/

// aus Kalibrierung am 12.03.20 Gerät von DL5OBT; shunt 22K am Ausgang AD8307
// float Slope =0.06204;
// float Intercept= -86.6;

//Kalibrierung am 17.08.20 mit HP-HF-Generator von DD4OI bei 10 MHz an 7 Messpunk-
ten 0 bis -60 dBm
float Slope =0.06251;
```

```

float Intercept= -88.07;

//aus Diagramm Abb. 31 im Datenblatt abgegriffene Werte ergeben folgende
Kalibrierdaten:
//float Slope = 0.04;
//float Intercept = -90;

//*****

const float ADC_Steps = 1023.0;
const float ADC_Referenz = 2490;
// max. Messwert entspr. -11dBm dBm = 0,07 mW ab hier bedingt durch 3,3V Betriebs-
spannung am AD8307 nicht mehr linear
float Max_mess = 1750;

float Pixel_mV =160/Max_mess;
float Leistung_dBm_alt=0;
int Me1_alt = 0;
int Me1 = 0;

void setup()
{
  Serial.begin(9600);
  analogReference(EXTERNAL);

  Serial.print(" Slope : ");Serial.println(Slope,5);
  Serial.print(" Intercept : ");Serial.println(Intercept,2);

  tft.initR(INITR_BLACKTAB);    // Initialisierung der Bibliothek
  //tft.initR(INITR_GREENTAB); // Initialisierung der Bibliothek
  tft.fillScreen(ST7735_BLACK); // Färbt Hintergrund Schwarz}
  tft.setRotation(3);          // Querformat oder (1) 180 Grad gedreht

  //delay(1000);

  tft.drawLine(0, 10, 160, 10, ST7735_GREEN);
  tft.drawLine(0, 103, 160, 103, ST7735_GREEN);

  tft.setCursor(7,1);
  tft.setTextSize(1);
  tft.setTextColor(ST7735_GREEN);
  tft.print("HF-mW-Meter mit AD8307");
  tft.fillRect(1, 105, 159, 20, ST7735_BLACK);
  tft.setCursor(130,110);
  tft.setTextColor(ST7735_WHITE);

  tft.fillRect(0, 107, 159, 21, ST7735_BLACK);
  tft.setCursor(5,110);
  tft.setTextSize(1); tft.setTextColor(ST7735_WHITE);
  tft.print("Rohwert am ADC : "); tft.print(0);tft.print("    mV");

```

```

tft.setCursor(5,119);
tft.print("Sl="); tft.print( Slope,4);tft.print(" Interc= ");tft.print(Intercept,2);


printEinheiten(1);
printEinheiten (4);


//Skala für den Balkenzeiger

tft.fillRect(1,46,160,3,ST7735_YELLOW);
for(int i=0; i <160;i= i+16) {
  tft.fillRect(i,46,3,3,ST7735_BLACK);

}

// -----Kopf der Datentabelle -----
Serial.println("Rohwert");
Serial.println("mV; dBm; uW; mV ");

}

void loop()
{
  Me1 = ADC_lesen () ; // Messwert vom AD3807 auslesen
  //if (Me1 >1800) { Me1=1800;}

  // -----Berechnung und Anzeige am Disp wenn sich etwas geändert hat-----
  if ( Me1 != Me1_alt) {

//----- Leistungsverh.[dBm] und Pegel an 50 Ohm [mW] u. [mV] berechnen ----

float Leistung_dBm = Me1*Slope+Intercept;
float Pegel_mW = pow(10,(Leistung_dBm/10)); // [mW]
float Pegel_mV = 1000*sqrt (Pegel_mW*50/1000); //[mV]

Serial.print(Me1);Serial.print(" ; ");
Serial.print(Leistung_dBm,1);Serial.print("; ");
Serial.print(Pegel_mW*1000,5);Serial.print("; ");
Serial.print(Pegel_mV,3);Serial.print("; "); Serial.println( " " );

Rohwert_disp(Me1);
Balken_disp (Me1);


//-----Leistungsverh._dBm ausgeben -----

tft.fillRect(1, 23, 115, 18, ST7735_BLUE);
tft.setTextSize(2);
tft.setTextColor(ST7735_WHITE);
tft.setCursor(20,25);

```

```
tft.print(Leistung_dBm,1);  
// printEinheiten(1);
```

```
//-----Pegel uW oder mW ausgeben -----
```

```
tft.setTextSize(2);  
tft.fillRect(1, 48, 115, 18, ST7735_BLUE);  
tft.setCursor(5,50);  
if (Pegel_mW <0.01) {  
tft.print(Pegel_mW*1000,3);  
printEinheiten(3);  
tft.setTextSize(2);  
}  
else {  
tft.print(Pegel_mW,3);  
printEinheiten(2);  
tft.setTextSize(2);  
}
```

```
//----- Pegel mV ausgeben-----
```

```
tft.fillRect(1, 73, 115, 18, ST7735_BLUE);  
// printEinheiten (4);  
tft.setCursor(5,75);  
  
if (Pegel_mV <= 1) { tft.print(Pegel_mV,4);}  
else if (Pegel_mV <= 10) { tft.print(Pegel_mV,3);}  
else if (Pegel_mV < 100) { tft.print(Pegel_mV,2);}  
else {  
tft.print(Pegel_mV,1);  
}
```

```
//-----Messwert auerhalb des gueltigen Kalibrierbereichs markieren -----
```

```
if( Me1 > Max_mess) {  
tft.setTextColor(ST7735_RED);  
tft.setCursor(100,23);  
tft.print("!");  
}  
}
```

```
Me1_alt = Me1;  
delay(100);
```

```
}
```

```
/******L O O P E N D *****
```

```
//-----ADC auslesen-----
```

```
float ADC_lesen (){  
int Abfragen=1000;
```



```

float Sum=0;
float Me1=0;
for(int i=0;i<Abfragen;i++){
    Me1 = map(analogRead(Sensor1), 0,ADC_Steps,0,ADC_Referenz) ;
    Sum = Sum+Me1;
}
int Me0 =Sum/Abfragen;
Me1=Me0;
Sum =0;
return Me1;
}

//-----Rohwert ausgeben -----
void Rohwert_disp(float Me1){
tft.fillRect(105, 107, 30, 10, ST7735_BLACK);
tft.setCursor(105,110);
tft.setTextSize(1); tft.setTextColor(ST7735_WHITE);
tft.print(Me1,0);
}

// -----Rohmeßwert als Balken -----
void Balken_disp (float Me1){
tft.fillRect(1,41,160,5,ST7735_BLACK); // Balkenbereich löschen
tft.fillRect(1,41,Me1*Pixel_mV,5,ST7735_RED); // neuen Balken zeichnen
}

void printEinheiten (int Einheit)
{
    tft.setTextSize(2);

    if (Einheit == 1){
        tft.fillRect(120, 23, 40, 18, ST7735_BLACK);
        tft.setCursor(120,25); tft.print("dBm");
    }

    if (Einheit == 2){
        tft.fillRect(120,48, 40, 18, ST7735_BLACK);
        tft.setCursor(120,50); tft.print("mW");
    }
    else if (Einheit ==3) {
        tft.fillRect(120, 48, 40, 18, ST7735_BLACK);
        tft.setCursor(120,50); tft.print("uW");
    }
    else if (Einheit == 4) {
        tft.fillRect(120, 73, 40, 18, ST7735_BLACK);
        tft.setCursor(120,75); tft.print( "mV");
    }

    else if (Einheit == 5) {
        tft.fillRect(120, 73, 40, 18, ST7735_BLACK);
        tft.setCursor(120,75); tft.print( "uV");
    }
}

```

```
else if (Einheit ==6) {  
    tft.fillRect(120, 48, 40, 18, ST7735_BLACK);  
    tft.setCursor(120,50); tft.print("pW");  
}  
  
}
```