


```

#include <math.h>//log für dB-Berechnung

int b,balt,bw,u,up,ualt, p10, x, y,on,mod10,mod30; //u = Spannung an A1
und b = Bereich b ...

int messen = 1;
int aktion = 0;
unsigned long t, startzeit,toverflow,tmessung;//für millis
float vfloat = 1.0;
float v10 = 0.975;//1.95;//Verhältnis Skalenendwert / Auflösung
float v3 = 3.58;//2.93; //Verhältnis Skalenendwert / Auflösung

String dim = ""; //für die Anzeige der Bereichsdimension
String banz = "";

void setup() {
//*****SETUP*****

  pinMode(A0, INPUT); //Schalterposition = Bereich = b

  pinMode(A3, INPUT); //Mess-Spannung u

  pinMode(A1, INPUT); //Mess-Spannung für Akku

  //Serial.begin(9600);

  tft.begin();
  tft.setRotation(1);
  tft.fillScreen(ILI9341_BLACK);

  //Akku abfragen

  akku();

bw = analogRead(A0);

  if (bw > 5) {

    bemitteln();
  }

  // Bildschirm löschen
  tft.fillScreen(ILI9341_BLACK);

  banzeigen();
  einheit();

//Hinweis für dB printen
if( b == 13){
  tft.setCursor(100, 207);
  tft.setTextColor(ILI9341_BLUE);
  tft.setTextSize(2);
  tft.print(" 0dB = 775 mV");
}
}

```

```

}

uakt = 0; //damit die Darstellung mit NULL beginnt und startet

//b = 8;////////////////////////////////////// TEST

skalen();

u = analogRead(A3); //Einlesen für den ersten Bildschirm

tmessung = millis(); //erste Darstellung wurde gestartet

if (mod10 == 1 ) {balken10(); anzeige10();}
if (mod30 == 1 ) {balken30(); anzeige30();}

if ( b == 13);{skalen();balken10();anzeige10();} //Testbetrieb anzeigen

}

void bermitteln(){

  if (bw > 5) {
    //auch hier muss der Positionswert > als das Rauschen sein, aber
    kleiner als die Stufe 13

    b = map(bw, 0, 1022, 1, 14); // mittlerer Wert 1020; anpassen

  }
}

void akku(){

  //Akku abfragen
  float afloat = 0.0;
  int uakku;

  uakku = analogRead(A1); //Akkuspannung

  //uakku = 646; //TEST 12,6v TEST ; bei 10,98 kommt die Warnung in Rot

  afloat = uakku * 0.017; //Spannungsteiler 4k7 / 2k2; bei max. 12,6V =
  4,02V an A1

  tft.setCursor(155, 225);
  tft.fillRect(185, 220, 150, 50, ILI9341_BLACK);

  if (afloat > 11.1)
  tft.setTextColor(ILI9341_GREEN);

  else tft.setTextColor(ILI9341_RED);
}

```

```

tft.setTextSize(2);
tft.print("Akku ");

tft.print(afloat);
tft.print (" V");

}

void ziffern10() {
  //Beschriftung der Balkenskalen
  x = 4;
  y = 60;

  tft.setCursor(x, y);
  tft.setTextColor(ILI9341_WHITE);
  tft.setTextSize(1);
  tft.println("0");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("1");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("2");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("3");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("4");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("5");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("6");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("7");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("8");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("9");
  x = x + 30;
  tft.setCursor(x, y);
  tft.print("10");

}

void ziffern30() {
  //Beschriftung der Balkenskalen
  x = 4;
  y = 60;

  tft.setCursor(x, y);

```

```

tft.setTextColor(ILI9341_WHITE);
tft.setTextSize(1);
tft.println("0");
x = x + 100;
tft.setCursor(x, y);
tft.print("1");
x = x + 100;

tft.setCursor(x, y);
tft.print("2");
x = x + 100;
tft.setCursor(x, y);
tft.print("3");
}

```

```

void printskale10() {
  x = 4;
  y = 72;
  int k = 2;
  int l = 8;

  tft.setCursor(x, y);
  tft.drawLine(x, y, 310, y, ILI9341_WHITE);

  for (x = 4; x <= 310; x = x + 6) {
    //alle 5 Pixel ein kurzer Teilstrich; alle 30 ein langer Strich
    tft.fillRect(x, y, k, 4, ILI9341_WHITE);
  }
  for (x = 4; x <= 310; x = x + 30) {
    tft.fillRect(x, y, k, l, ILI9341_WHITE);
  }

  tft.fillRect(233, 100, k, 10, ILI9341_BLUE);

  tft.setCursor(236, 110);
  tft.setTextSize(1);
  tft.setTextColor(ILI9341_WHITE);
  tft.print("0 dB");
}

```

```

void printskale30() {
  x = 4;
  y = 72;
  int k = 2;
  int l = 8;

  tft.fillRect(0, 0, 310, 100, ILI9341_BLACK); //alte Beschriftung löschen
  tft.setCursor(x, y);
  tft.drawLine(x, y, 310, y, ILI9341_WHITE);

  for (x = 4; x <= 310; x = x + 10) {
    //alle 5 Pixel ein kurzer Teilstrich; alle 30 ein langer Strich
    tft.fillRect(x, y, k, 4, ILI9341_WHITE);
  }
  for (x = 4; x <= 310; x = x + 100) { //lange Striche
    tft.fillRect(x, y, k, l, ILI9341_WHITE);
  }
}

```

```

tft.fillRect(233, 100, k, 10, ILI9341_BLUE);

tft.setCursor(236, 110);

tft.setTextSize(1);
tft.setTextColor(ILI9341_WHITE);
tft.print("0 dB");

}

void banzeigen() { //Bereichsanzeige

//Feld für banz löschen

if (b == 13) banz = "";
if (b == 12) banz = "300V";
if (b == 11) banz = "100V";
if (b == 10) banz = "30V";
if (b == 9) banz = "10V";
if (b == 8) banz = "3V";
if (b == 7) banz = "1V";
if (b == 6) banz = "300mV";
if (b == 5) banz = "100mV";
if (b == 4) banz = "30mV";
if (b == 3) banz = "10mV";
if (b == 2) banz = "3mV";
if (b == 1) banz = "1mV";

tft.setCursor(160, 22);
//alte Bereichsanzeige löschen
tft.fillRect(118, 15, 122, 40, ILI9341_BLACK);

tft.setCursor(4, 30);
tft.setTextColor(ILI9341_WHITE);
tft.setTextSize(2);
tft.print("Bereich:");

tft.setCursor(120, 22);
tft.setTextSize(3); //Größe der Dimension im Feld "Bereich:"
tft.setTextColor(ILI9341_WHITE);
tft.print(banz);

}

void einheit() {

if (b > 0 & b < 7) dim = " mV";
else dim = " V";

}

////////////////////////////////////
////////////////////////////////////

```

```

void loop() {

mod10 = 0;
mod30 = 0;

//Akku-Spannung anzeigen bzw. warnen
if ( (startzeit + 10000) > millis() ) akku(); //Akkuspannung für 10 s
anzeigen
else {
  tft.setCursor(155, 225);
  tft.fillRect(150, 220, 150, 30, ILI9341_BLACK);
}

if ( millis() > tmessung + 2000) {
  u = analogRead(A3);

  display();

}
//Spannung messen
u = analogRead(A3); // read the input pin für u

uالت = u;

//auf Overflow überwachen und
if (u > 1022) toverflow = millis();
if (toverflow + 2000 > millis()) overflow(); //anpassen! TEST

//Abfrage für den Bereich
bw = analogRead(A0);

bermitteln();

delay(500); //Einpendeln der Mess-Spannung abwarten

if ( b != balt) skalen(); //dbchange();

balt = b;

//veränderte sich der vorangegangene Wert?
//Schwellwerte +/-5 wegen Rauschen

u = analogRead(A3); // read the input pin für u

if(b < 1 || b > 13){b = 13;skalen();balken10(); anzeige10();}

differenz();

```

```

if(u < 0 || u > 1023) {u = 0;display();}

}

////////////////////////////////////
////////////////////////////////////

void differenz(){
  if (u > (ualt + 4)) display(); //JA
  else {
    if(mod10 == 1) {einheit(); anzeige10();mod10 = 0;tmessung = millis();}
  }
  if (u < (ualt - 4))display();//JA
  else {
    if(mod10 == 1) {einheit(); anzeige10();mod10 = 0;tmessung = millis();}
  }
  //-----

  if (u > (ualt + 4))display(); //JA
  else {
    if(mod30 == 1) {einheit(); anzeige30();mod30=0;tmessung = millis();}
  }
  if (u < (ualt - 4)) display();//JA
  else {
    if(mod30 == 1) {einheit(); anzeige30();mod30=0;tmessung = millis();}
  }
}

void display() {

if (b == 2 || b == 4 || b == 6 || b == 8 || b == 10 || b == 12){
balken30(); mod30 = 1;}
if (b == 1 || b == 3 || b == 5 || b == 7 || b == 9 || b == 11 || b ==
13){ balken10(); mod10 =1;}

}

void skalen(){

  if (b == 2 || b == 4 || b == 6 || b == 8 || b == 10 || b == 12) {
    mod30 = 1;

    tft.fillRect(4, 0, 310, 120, ILI9341_BLACK);
    printskale30();
    ziffern30();

    tft.setCursor(20, 80);
    tft.drawRect(4, 80, 310, 22, ILI9341_WHITE);

  }

  if (b == 1 || b == 3 || b == 5 || b == 7 || b == 9 || b == 11 || b ==
13) {
    mod10 = 1;

```

```

    tft.setCursor(4, 110);
    tft.fillRect(4, 0, 310, 120, ILI9341_BLACK);
    printskale10();
    ziffern10();

//Rechteck für die Skale zeichnen
    tft.drawRect(4, 80, 310, 22, ILI9341_WHITE);

    //den blauen Strich für 0 dB setzen
    tft.fillRect(233, 100, 2, 10, ILI9341_BLUE);
    tft.setCursor(236, 110);
    tft.setTextSize(1);
    tft.setTextColor(ILI9341_WHITE);

}

// dB-Add anzeigen
dbchange();//
//Bereich anzeigen
banzeigen();//TEST

if (b == 13){

    tft.setCursor(160, 22);
    tft.fillRect(118, 15, 122, 40, ILI9341_BLACK);
    tft.setCursor(120, 22);
    tft.setTextSize(3);
    tft.setTextColor(ILI9341_BLUE);
    tft.print("Test");

    tft.setCursor(100, 207);
    tft.setTextColor(ILI9341_BLUE);
    tft.setTextSize(2);
    tft.print(" 0dB = 775mV");

}

}

void balken10() {

    p10 = map(u, 0, 1024, 0, 300);//vorher 512

    tft.fillRect(4, 82, 308, 17, ILI9341_BLACK);
    p10 = p10 - 1;

    tft.setCursor(4, 80);
    // tft.fillRect(int16_t x, int16_t y, int16_t w, int16_t h, uint16_t
color)
    tft.fillRect(5, 82, p10, 17, ILI9341_RED);

    p10 = p10 + 1;

```

```
}
```

```
void anzeige10() {
```

```
    //gesamtes Werte-Feld löschen durch Überschreiben mit Rechteck
```

```
tft.fillRect(0, 130, 320, 50, ILI9341_BLACK);  
    //bei Überlauf rechts auch die nächste Zeile löschen  
tft.fillRect(0, 170, 32, 50, ILI9341_BLACK);
```

```
vfloat = u * v10; //  
if (b == 3 || b == 5 || b == 11 ) vfloat = vfloat / 10.0;  
if (b == 1 || b == 7 || b == 13) vfloat = vfloat / 1000.0;  
if ( b == 9 ) vfloat = vfloat / 100.0;
```

```
tft.setCursor(20, 130);  
tft.setTextColor(ILI9341_ORANGE);  
tft.setTextSize(6);  
int pegel;  
pegel = (int) vfloat;  
if ( b == 3 || b == 11){  
    tft.print(pegel); tft.print(dim);  
} else { tft.print(vfloat);tft.print(dim); }
```

```
    db();
```

```
}
```

```
void anzeige30() {
```

```
    //gesamtes Werte-Feld löschen durch Überschreiben mit Rechteck  
tft.fillRect(0, 130, 320, 50, ILI9341_BLACK);  
    //bei Überlauf rechts auch die nächste Zeile löschen  
tft.fillRect(0, 170, 32, 50, ILI9341_BLACK);
```

```
    //delay(100),  
    //u = analogRead(A3);
```

```
vfloat = u * v3 ;  
if(vfloat < 1000) vfloat = vfloat * 1.15;//Korrekturen  
if(vfloat > 1900) vfloat = vfloat * 0.96;
```

```
if ( b == 8 || b == 9 || b == 2) vfloat = vfloat / 10; //Bereich 300V
```

```
if (b == 10 || b == 6 || b == 4 || b==8 || b == 2) vfloat = vfloat /  
100; //Bereich 30mV
```

```
if ( b == 6 || b == 12) vfloat = vfloat / 10; //Bereich 3V
```

```
int pegel;  
tft.setCursor(20, 130);  
tft.setTextColor(ILI9341_ORANGE);  
tft.setTextSize(6);
```

```

    pegel = (int) vfloat;
    if ( b == 10 || b == 12 ){
        tft.print(pegel);
        tft.print(dim);
    }

    else {tft.print(vfloat);

    tft.print(dim);
    }

    db();
}

void balken30() {
    vfloat = u * 1.2 ;

    if (vfloat < 310.0)vfloat = vfloat * 1.15;//Korrekturen
    if(vfloat > 650.0) vfloat = vfloat * 0.98;

    int u30 = vfloat;

    p10 = map(u30, 0, 1024, 0, 300);//vorher 512

    //p10 = p10 - 1;
    //alten Balken löschen
    tft.fillRect(5, 82, 308, 17, ILI9341_BLACK);
    //dann neuen Balken anzeigen
    tft.fillRect(5, 82, p10, 17, ILI9341_RED);

    //p10 = p10 + 1;
}

void overflow() {

    //solange u > 1023 ist, blinkt das Wort "Overflow"
    tft.setCursor(220, 10);
    tft.setTextColor(ILI9341_RED);
    tft.setTextSize(2);
    tft.println("Overflow");

    delay(500);

    tft.setCursor(220, 10);
    tft.setTextColor(ILI9341_BLACK);
    tft.setTextSize(2);
    tft.println("Overflow");

    delay(500);

}

void db(){

```

```

//dLog = log(u2/u1); //Library ist in der IDE enthalten!

double wert;
int pdb;
float dbfloat = 1.0;

//u = analogRead(A3);

if (b == 1 || b == 3 || b == 5 || b == 7 || b == 9 || b == 11 || b == 13)
dbfloat = u * v10; // 1V-Skale!

if ( b == 2 || b== 4 || b == 6 || b == 8 || b == 10 || b == 12)dbfloat =
u * v3 * (3/10); //3er-Skale

//löschen des Textfeldes für dB-Wert
tft.fillRect(4, 200, 105, 30, ILI9341_BLACK);

wert = 10 * log(dbfloat / 775);

tft.setCursor(20, 200);
tft.setTextSize(3);
tft.setTextColor(ILI9341_CYAN);

pdb = (int) wert; //Integer ist genau genug

tft.print(pdb); tft.print("dB");

}

void dbchange(){

//rechtes Feld zunächst löschen
tft.fillRect(118, 195, 200, 40, ILI9341_BLACK);

tft.setCursor(120, 200);
tft.setTextColor(ILI9341_BLUE);
tft.setTextSize(3);

if (b == 8) tft.print("+10 dB"); if (b == 9) tft.print("+20 dB"); if (b ==
10) tft.print("+30 dB");

if (b == 11) tft.print("+40 dB"); if (b == 12) tft.print("+50 dB"); if (b
== 7) tft.print(" ");

if (b == 6) tft.print("-10 dB"); if (b == 5) tft.print("-20 dB"); if (b
== 4) tft.print("-30 dB");

if (b == 3) tft.print("-40 dB"); if (b == 2) tft.print("-50 dB"); if (b
== 1) tft.print("-60 dB");

}

```

