

Temperatursensor mit Display

Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- I2C Display
- 1 Wire Temperatursensor (DS18B20)
- 1x 4.7K pullup resistor
- 7x MM-Kabel
- Kleines Breadboard

Benötigte Software

- Bibliothek LiquidCrystal I2C von Marco Schwartz oder Frank de Brabander
- Bibliothek Arduino-Temperature-Control-Library von Miles Burton
- Bibliothek OneWire von Jim Studt

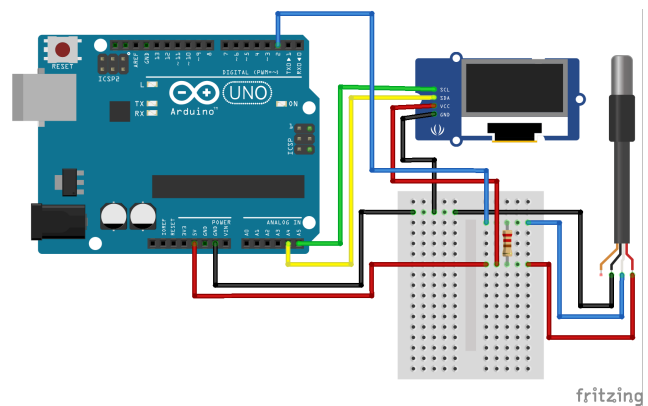
Beschreibung

Mit dem wasserdichten DS18B20 Sensor lässt sich die Temperatur eines Raumes oder von Flüssigkeiten bestimmen. Hier bauen wir ein einfaches Thermometer mit einer digitalen Anzeige in Grad Celsius. Der Sensor hat eine Messgenauigkeit von $\pm 0,5^{\circ}\text{C}$ von -10°C bis $+85^{\circ}\text{C}$ und einen Messbereich von -55°C bis $+125^{\circ}\text{C}$.

Verkabelung

| DS18B20 | Arduino | Anmerkungen |
|---------|---------|-----------------|
| GND | GND | Über Breadboard |
| VDD | 5V | Über Breadboard |
| DQ | 2 | |

| I ² C Display | Arduino | Anmerkungen |
|--------------------------|---------|-----------------|
| GND | GND | Über Breadboard |
| VCC | 5V | Über Breadboard |
| SDA | A4 | |
| SCL | A5 | |



Schaltplan

Code

```
#include <DallasTemperature.h>
#include <LiquidCrystal_I2C.h> // LiquidCrystal_I2C Bibliothek einbinden.
#include <OneWire.h>
#include <Wire.h> // Wire Bibliothek einbinden.

// Objekt für das LCD Display
LiquidCrystal_I2C lcd(0x27, 16, 2);

int ONE_WIRE_BUS = 2; // Verbindung vom OneWire auf Pin 2 des Arduinos
OneWire oneWire(ONE_WIRE_BUS); // Instanz zur Kommunikation mit einem OneWire
// Sensor
DallasTemperature sensors(&oneWire); // Nutze diese Instanz für die
// DallasTemperature Bibliothek

void setup() {
  // Starte serielle Schnittstelle zur Kommunikation mit dem Arduino
  Serial.begin(9600);
  // Starte Kommunikationsschnittstelle für den Temperatursensor
```

```

sensors.begin();
// Starte das Display
lcd.init();
// Hintergrundbeleuchtung fürs Display einschalten
lcd.backlight();
}

void loop() {
// Frage beim Sensor die Temperatur an, Start der Messung
sensors.requestTemperatures();

// Gebe die Temperatur in Grad Celsius auf der seriellen Schnittstelle aus
Serial.println("Temperatur: " + String(sensors.getTempCByIndex(0)) + "°C");

// Gebe die Temperatur auf dem Display aus
lcd.setCursor(0, 0);
lcd.print("Grad Celsius:");
lcd.setCursor(0, 1);
lcd.print(String(sensors.getTempCByIndex(0)) + "");

// Warte eine Sekunde
delay(1000);
}

```

Aufgaben

① Beschreibe, was der Code tut!

Der Pin für den Temperatursensor wird gesetzt und ein Objekt für das Display erzeugt. Zudem erzeugen wir ein Objekt zur Kommunikation mit dem Sensor und eines für die Nutzung der Bibliothek. Im Setup wird das Display gestartet und eingeschaltet. Den Messwert erhalten wir in der Loop vom Arduino und rufen ihn mit 'sensors' auf. Anschließend wird er auf der seriellen Schnittstelle und dem Display ausgegeben.

② Messt eure Körpertemperatur.

Es kann etwas dauern bis der richtige Wert ausgegeben wird, danach sollte er jedoch zwischen 36,5 und 37,4 °C liegen.

Häufige Fragen und Probleme

Meine Bildschirmausgabe zeigt -127 Grad an, was kann ich tun?

Steckt der Widerstand richtig drin? Nochmal nachschauen und zurechtrücken.

Wann weiß ich, welche Temperatur die richtige ist?

Der Sensor misst eigentlich sofort, jedoch kann er für das Messen stark unterschiedlicher Temperaturen ca. 30sec benötigen, um auf die Nachkommastellen genau zu messen.

Ich habe einen Sensor mit nur 3 Kabeln, worin unterscheiden sie sich?

Es ist relativ egal ob 3 oder 4 Kabel, wichtig zu wissen ist folgendes:

- 3 Kabel -> schwarz=GND, rot=3-5V, weiß/gelb=Data
- 4 Kabel -> schwarz=GND, rot=3-5V, weiß=Data, restliches Kabel=Abschirmung

Das Material und dessen Inhalte sind - sofern nicht anders angegeben - lizenziert unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (für den vollständigen Lizenztext siehe <https://creativecommons.org/licenses/by-sa/4.0/legalcode>)