

Ein Ampel-Thermometer

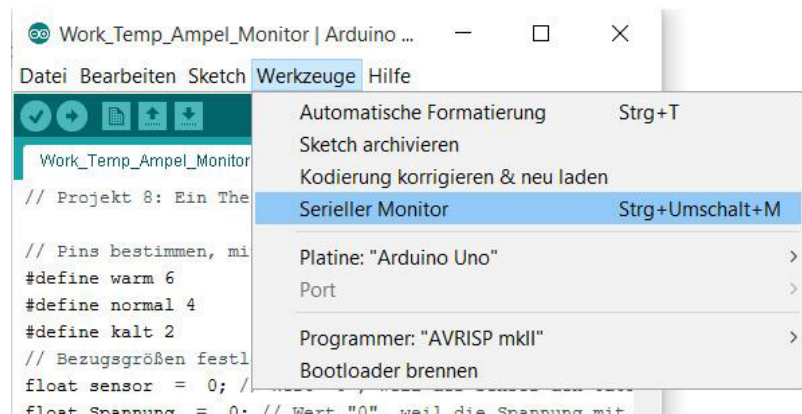
Stellt euch vor, in eurem Klassenraum funktionierte die Heizung nicht ordnungsgemäß: Mal ist es zu heiß, mal zu kalt. Beschwerden beim Hausmeister haben zu keinem Ergebnis geführt. Er behauptet, dass die Heizungsanlage ordnungsgemäß funktioniere. Sie würde elektronisch überwacht und er könne auf seinen Anzeigen keine Fehlfunktion feststellen. Schulleitung und Lehrkräfte teilen diese Ansicht. Da ihr euch trotzdem nicht so einfach abspesen lassen wollt, baut ihr mithilfe eines Temperatur-Sensors und drei LEDs ein Ampel-Thermometer, um die jeweils herrschenden Temperaturen objektiv anzeigen zu können.

Aufgabe

Das Ampel-Thermometer soll mithilfe eines Temperatur-Sensors und drei LEDs in den Farben Rot, Grün und Blau die Wohlfühltemperatur und davon abweichende Werte anzeigen können. Wenn also die Temperatur im angenehmen Bereich liegt, soll die grüne LED leuchten, wenn es dagegen zu warm ist, die rote LED und wenn es zu kalt ist, die blaue LED. Da vor der Messung Einigkeit über die einzustellenden Werte, also welche Temperatur als zu warm, als normal oder als zu kalt anzusehen ist, muss das Thermometer auf diese Werte einstellbar sein. Die vom Sensor übermittelten Werte sollen einsehbar sein, um die errechneten Temperaturwerte überprüfen zu können.

Hinweis

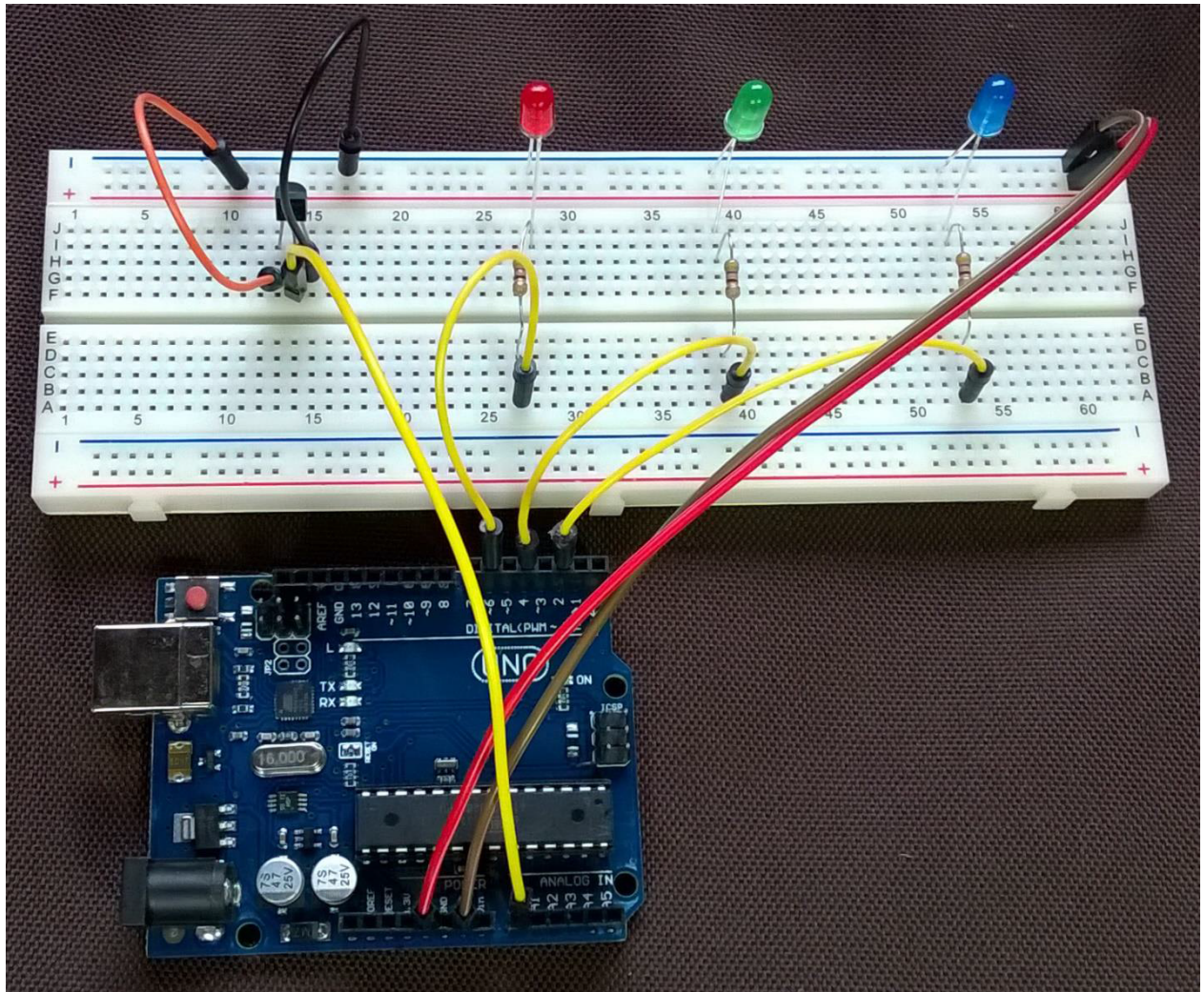
Der Temperatur-Sensor übermittelt laufend temperaturabhängige Rohwerte, die in Spannung und Celsius-Grade umgerechnet werden, ehe sie zur Anzeige gebracht werden. Die Rohwerte können mit dem seriellen Monitor der Arduino-Software zur Anzeige gebracht werden. Die nebenstehende Abbildung zeigt, wie man ihn startet.



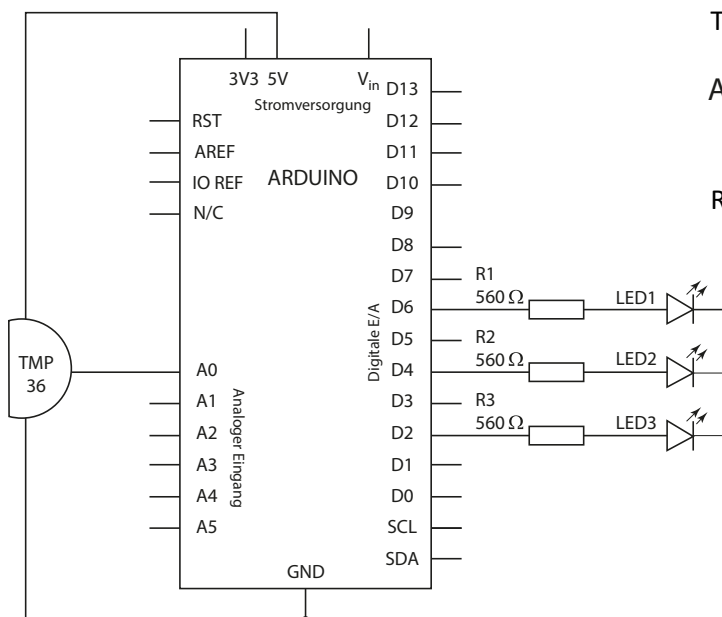
Benötigte Materialien

- Computer mit der Arduino-Software
- Arduino UNO
- USB-Kabel
- eine rote LED
- eine gelbe LED
- eine blaue LED
- drei 330-Ω-Widerstände
- eine Steckplatine (Breadboard)
- Steckbrücken

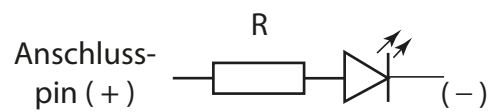
Lösung



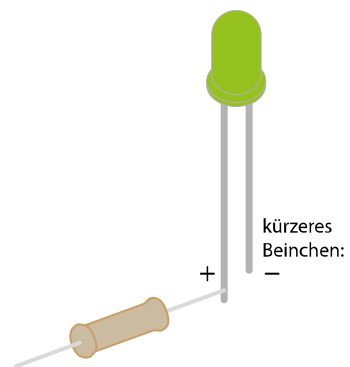
Schaltplan



Teilskizze



Realitätsnahe Darstellung



Programmcode

```
// Pins bestimmen, mit denen die LEDs zu verbinden sind:
#define warm 6
#define normal 4
#define kalt 2
// Bezugsgrößen festlegen
float sensor = 0; // "0", weil der Sensor den tatsächlichen Wert ermittelt
float Spannung = 0; // "0", weil die Spannung mit Hilfe der Sensorrohdaten errechnet wird
float normalTemp = 0; // "0", weil die Temperatur durch Umrechnung der Spannung ermittelt wird
float warmTemp = 26; // 26 Grad Celsius als "warm" definieren
float kaltTemp = 20; // 20 Grad Celsius als "kalt" definieren

// Sensor im System anmelden
int TMP36;

void setup()
{
  TMP36 = analogRead(A0); // Auslesen der Rohdaten
  Serial.begin(9600); // Aktivieren des Monitors

  pinMode(warm, OUTPUT);
  pinMode(normal, OUTPUT);
  pinMode(kalt, OUTPUT);
}

void loop()
{
  Serial.println(TMP36); // Gibt die Sensorrohwerte im Monitor aus
  delay(1);

  sensor = analogRead(0); // Übernimmt den Sensorrohwert
  Spannung = (sensor * 5000) / 1024; // Sensorrohwert in Spannung (mV) umrechnen
  Spannung = Spannung - 500; // Abweichung entfernen
  normalTemp = Spannung / 10; // Spannung in die jeweiligen Temperaturwerte umrechnen

  // Die Normaltemperatur liegt zwischen den als kalt (20 Grad und weniger)
  // und als warm (26 Grad und mehr) definierten Temperaturbereichen

  if ( normalTemp < kaltTemp )
  {
    digitalWrite(kalt, HIGH);
    delay(1000);
    digitalWrite(kalt, LOW);
  }
  else if ( normalTemp > kaltTemp && normalTemp <= warmTemp )
  {
    digitalWrite(normal, HIGH);
    delay(1000);
  }
}
```

```
    digitalWrite(normal, LOW);  
}  
else  
{  
    // Temperatur ist > hotTemp  
    digitalWrite(warm, HIGH);  
    delay(1000);  
    digitalWrite(warm, LOW);  
  
}  
}
```