Arduino-Workshop

am Robert-Havemann-Gymnasium, 19.11.2019

von Franz Boczianowski, Didaktik der Physik, Humboldt-Universität zu Berlin

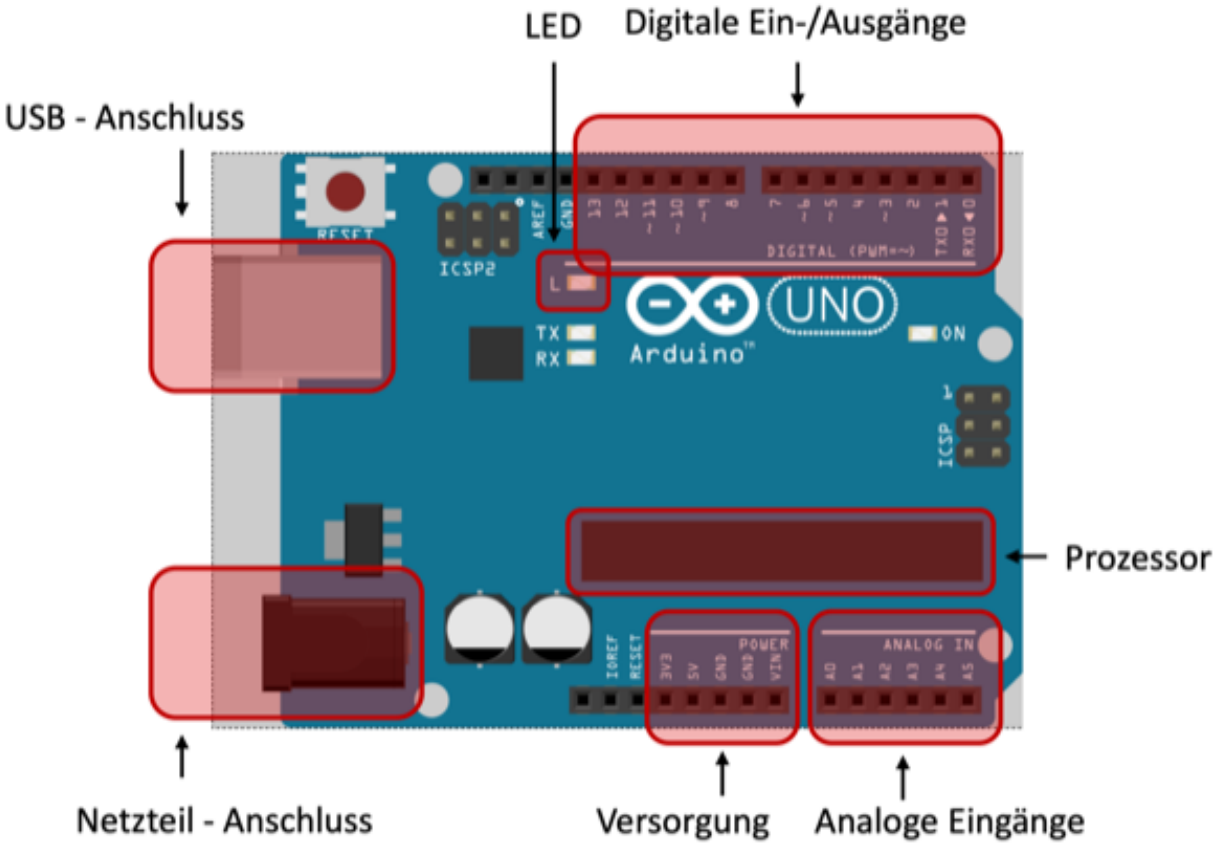


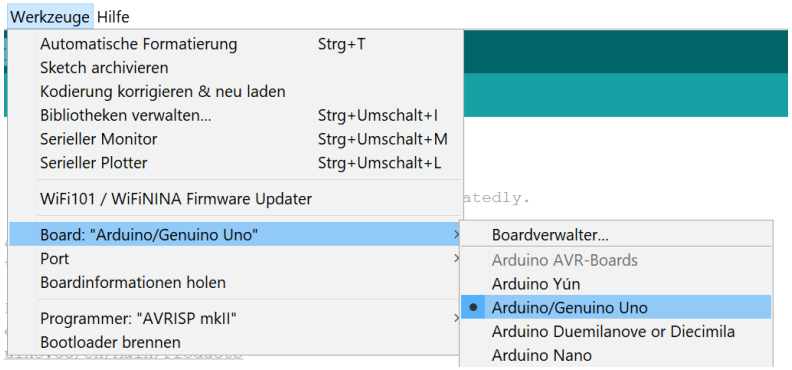
Abbildung wurde erstellt mit Fritzing.

A. Erste Schritte

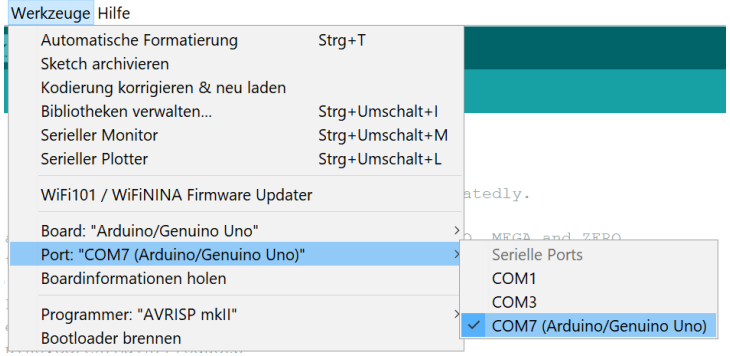
- Starte die Arduino Programmierumgebung.

- Schließe den Arduino mit dem USB-Kabel am PC an.

- Prüfe die Einstellungen

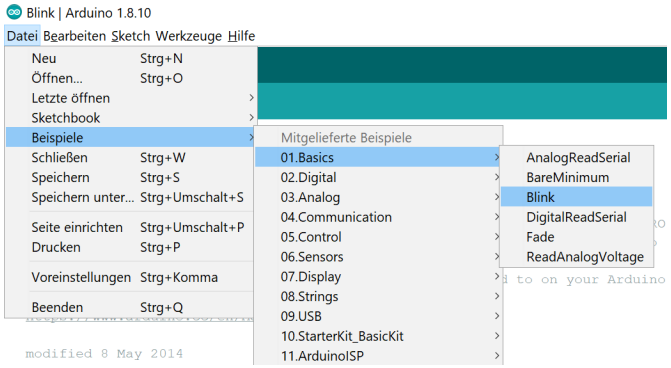


- Wähle den USB-Port, an dem der Arduino angeschlossen ist. Achte auf *Arduino/Genuino Uno*, siehe Abb. unten.



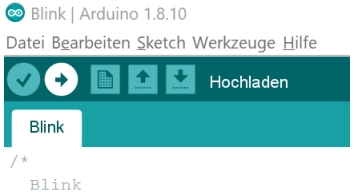
B. Beispielprogramm *Blink*

- Öffne das Beispiel-Programm Blink:



- Ein neues Fenster öffnet sich.

- Übertrage das Programm auf den Arduino (Klick auf Pfeil-Taste):



- Das Programm *Blink* lässt eine orange LED auf dem Board blinken.

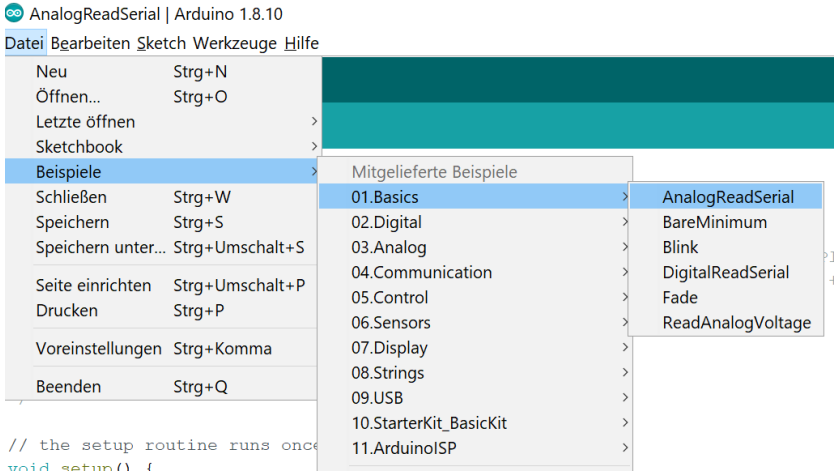
Tipp: Eine Dokumentation dieses und der folgenden Beispiel-Programme findest Du unter https://www.arduino.cc/en/Tutorial/BuiltInExamples oder auch hier https://funduino.de/nr-1-blinkende-led.

Aufgaben

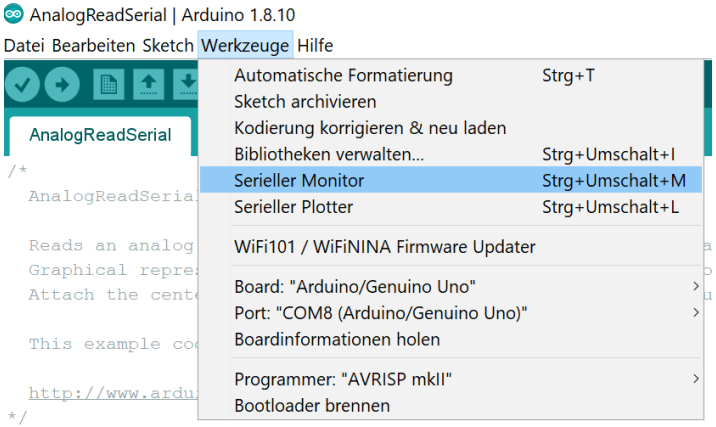
1. Verändere das Programm *Blink* so, dass die LED in einem anderen Takt blinkt.
   Suche dazu die Zeilen, in der die LED angeschaltet wird und die Zeile in der das Programm anschließend wartet.
   Übertrage das Programm erneut auf den Arduino.
   (Musterlösungen und Anmerkungen zum Programm findest Du im Anhang.)
2. Verändere den Takt so, dass man das Blinken mit bloßem Auge nicht mehr erkennen kann.
3. Verändere die Dunkelzeit so, dass die LED nur noch schwach zu leuchten scheint.

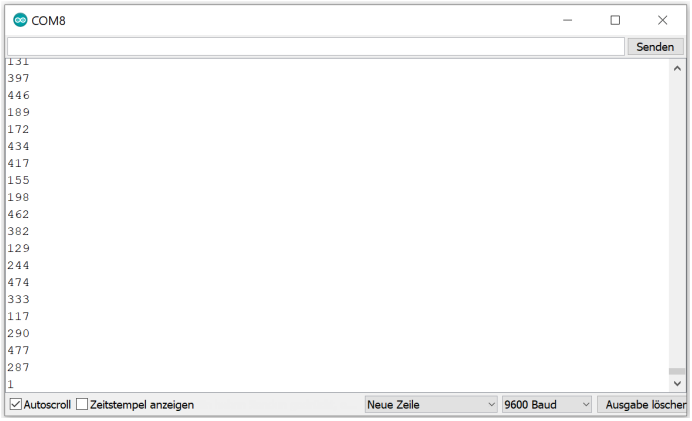
C. Beispielprogramm AnalogReadSerial

- Öffne das Beispielprogramm AnalogReadSerial.
  In diesem wird mit dem Arduino eine Spannung gemessen und an den PC gesendet.



- Öffne den *Seriellen Monitor*, siehe Abb. unten links.
  Dieser zeigt die Nachrichten an, die der Arduino an den PC sendet. Z.B. eine Vielzahl von Messwerten, siehe unten rechts.





Mit dem Arduino lassen sich Spannungen zwischen 0 und 5 V messen.
Im Programm AnalogReadSerial wird dazu der Anschluss A0 abgefragt.

Aufgaben

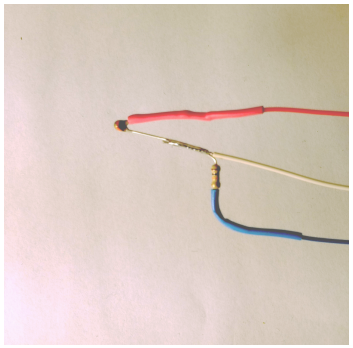
1. Suche am Arduino den Anschluss 5 V („5-Volt-Pluspol“),
   den Anschluss GND (für ground, d.h. Masse, 0 V oder „Minuspol“) und den analogen Eingang A0.
   Verbinde erst GND mit dem Eingang A0 und betrachte die im *Seriellen Monitor* ausgegeben Werte.
   Verbinde anschließend den 5 V Anschluss mit A0 und beobachte.
2. Bonusaufgabe: Wie fein kann der Arduino Spannungen messen? Berechne den kleinsten möglichen Spannungsunterschied.

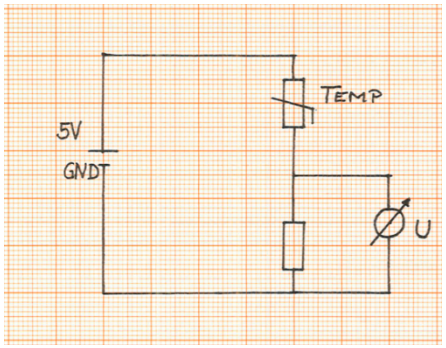
Musterlösungen und das Programm mit Anmerkungen findest Du wieder im Anhang.

D. Temperaturmessung

Ein Temperatursensor verändert seinen elektrischen Widerstand, wenn die Temperatur variiert.
Mit einem solchen Sensor und einem Arduino lässt sich ein Thermometer bauen.
(Im Detail: Der Temperatursensor verändert den elektrischen Strom, der im Stromkreis fließt.
Der Arduino miss an seinem analogen Eingang die Spannung über einen unveränderlichen Widerstand.
Diese Spannung verhält sich proportional zum elektrischen Strom.)

Die linke Abbildung zeigt den Temperatursensor oben und den unveränderlichen Widerstand unten.
In der rechten Abbildung ist der Schaltplan zu sehen.





Weiterführendes ist z. B. zu finden unter https://bildungsserver.berlin-brandenburg.de/unterricht/faecher/mathematik-naturwissenschaften/mint/i-mint-akademie/weiterfuehrende-schulen/fachset-informatik/, https://www.mymakerstuff.de/2018/05/18/arduino-tutorial-der-temperatursensor/, https://www.youtube.com/watch?v=2GkJwD-lRAA.)

Aufgaben

1. Verbinde den vorgefertigten Temperatursensor (siehe Abb.) mit dem Arduino:
   - rotes (bzw. oranges) Kabel an den 5 V Anschluss
   - blaues (bzw. grünes) an GND
   - weiß an A0

   Starte den *Seriellen Monitor*, wie vorher. (Auf dem Arduino soll nach wie vor das Programm *AnalogReadSerial* laufen.)
   Welche Werte werden vom Arduino zurückgegeben? Nimm den Temperatursensor zwischen zwei Finger. Wie ändern sich die Werte?

2. Bonusaufgabe: Schließe das Fenster des *Seriellen Monitors* und starte den *Seriellen Plotter* (auch Menüpunkt Werkzeug).
   Nimm erneut den Sensor zwischen die Finger und beobachte.

E. Kalibration

Die analogen Eingänge des Arduino liefern Werte zwischen 0 und 1023 zurück.
Diese Werte sollen im Folgenden Temperaturwerten in der physikalischen Einheit °C zugeordnet werden.
Abschließend soll das Programm so verändert werden, dass der Arduino die Umrechnung übernimmt und direkt Temperaturen in °C liefert.

Aufgaben

1. Bestimme die Temperatur im Raum mit einem herkömmlichen Thermometer. Welchen Wert gibt er Arduino zurück?
Skizziere die Werte (grob) in einem Diagramm (horizontale Achse: Arduino, senkrecht: Temperatur in °C).

2. Zeichnet eine Ausgleichsgrade. Welche mathematische Funktion f(x) = a x + b beschreibt die Ausgleichsgrade?

3. Verändere den Programmcode so, dass der Arduino direkt die Temperaturwerte in °C zurückliefert.
Beachte: Für einen Malpunkt ist ein * zu verwenden. Statt des Kommas einer Dezimalzahl ist im Englischen ein Punkt zu verwenden.

F. Bonusaufgabe

In dem Programm *Fade* (zu finden unter Datei/Beispiele/01.Basics) wird eine große externe LED langsam gedimmt.
Dazu wird im Programm eine Schleife verwendet: https://www.arduino.cc/en/Tutorial/Fading

Verändere das Programm so, dass sich die Helligkeit der Lampe mit der gemessenen Temperatur verändert.

Literaturempfehlung

- Erik Bartmann *Die elektronische Welt mit Arduino entdecken*

Anhang: Musterlösungen und Hilfen

zum B. Beispielprogramm *Blink*

Lösung Aufgabe 1

In einem Programm können Kommentare eingefügt werde, die beim Übertragen des Programms gelöscht werden.
Kommentare beginnen immer mit // und sind im Folgenden farblich hervorgehoben.

Programm.

```
void setup() {                  // Im Abschnitt Setup werden Konstanten und Variablen einmalig für das gesamte Programm festgelegt.
  pinMode(LED_BUILTIN, OUTPUT); // Die eingebaute LED wird zur Ausgabe festgelegt.
}                               // Die Klammer beendet das Setup.

void loop() {                   // Hier startet die „Loop“: Der Arduino wiederholt alle folgenden Schritte immer und immer wieder.
  digitalWrite(LED_BUILTIN, HIGH); // Die LED wird eingeschaltet (und bleibt bis auf Weiteres eingeschaltet.)
  delay(2000);                  // Das Programm wartet für 2000 ms (Millisekunden), also 2 Sekunden.
  digitalWrite(LED_BUILTIN, LOW); // Die LED wird ausgeschaltet.
  delay(500);                   // Das Programm wartet für eine halbe Sekunde.
}                               // Die Loop ist am Ende und wird von vorn abgearbeitet.
```

Lösung Aufgabe 2:

Programmausschnitt**.**

```
digitalWrite(LED_BUILTIN, HIGH); // Die LED wird eingeschaltet (und bleibt bis auf weiteres eingeschaltet.)
  delay(10);                    // Das Programm wartet für 10 ms (Millisekunden).
  digitalWrite(LED_BUILTIN, LOW); // Die LED wird ausgeschaltet.
  delay(10);                    // Das Programm wartet für 10 ms.
                                //Das Auge ist zu träge, um die 10 ms Dunkelheit wahrzunehmen. Die LED erscheint uns weniger hell.
                                // Auf diese Weise lässt sich eine LED dimmen.
```

Lösung zu Aufgabe 3

Programmausschnitt:

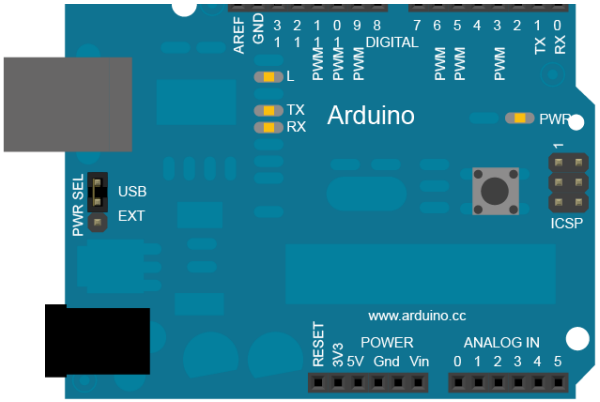
```
digitalWrite(LED_BUILTIN, HIGH);
delay(5);                         // Die LED leuchtet nur 5 ms.

digitalWrite(LED_BUILTIN, LOW);
delay(10);                        // Die LED bleibt für 15 ms ausgeschaltet. Also dreimal so lang, wie oben.
```

Zu C. Beispielprogramm *AnalogReadSerial*

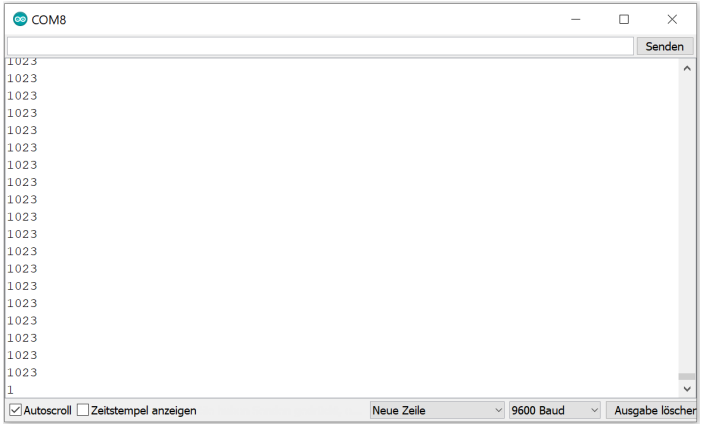
Lösung zu Aufgabe 1

Die Anschlüsse 5V, GND und A0 befinden sich alle an der Unterseite des Arduinos, siehe Abb. unten.



(Bildquelle: https://www.arduino.cc/en/Tutorial/AnalogReadSerial)

Lösung: Verbindet man den Analogeingang A0 mit Ground, zeigt der *Serielle Monitor* eine 0.
Verbindet man A0 mit 5 V wird die Zahl 1023 angezeigt.



Programmcode mit Kommentaren

```
void setup() {
  Serial.begin(9600);                 // Die Datenübertagung über das USB-Kabel wird eingestellt. (9600 bits per second)
}

void loop() {
  int sensorValue = analogRead(A0);   // Am Anschluss A0 wird ein Wert eingelesen. Dieser wird in der Variable sensorValue gespeichert.
                                      // Die Messwerte werden allerdings nicht in Volt gemessen, sondern in 1024 Abstufungen von fünf Volt.
                                      // Dabei bedeutet eine 0 auch 0 Volt, 1023 entsprechend 5 Volt.
                                      //Das ergibt eine Abstufung von 4,89 mV.

  Serial.println(sensorValue);        // Der Wert der Variable SensorValue wird an den PC gesendet. (Println = print line, Serial = serielle Datenübertragung)
  delay(1);                           // Das Programm wartet für eine Millisekunde, weil die Übertragung etwas Zeit braucht.
}
```

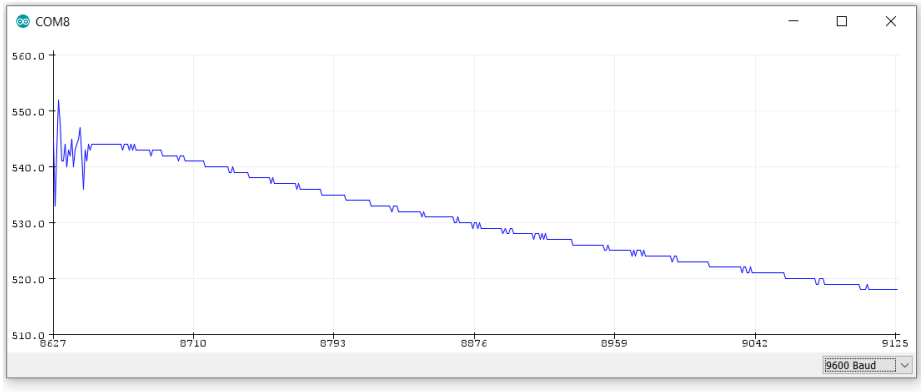
Lösung Aufgabe 2

Siehe Kommentare im Programm oben.

Zu D. Temperaturmessung

1. Anschlüsse wie in der Aufgabe zuvor.
   Der Arduino liefert bei angeschlossenem Temperatursensor Werte im Bereich von 400 bis 600.
   Je nach Temperatursensor steigt oder fällt dieser Wert, wenn der Sensor erwärmt wird.
   (Details siehe z. B. https://de.wikipedia.org/wiki/Kaltleiter, https://de.wikipedia.org/wiki/Heißleiter)

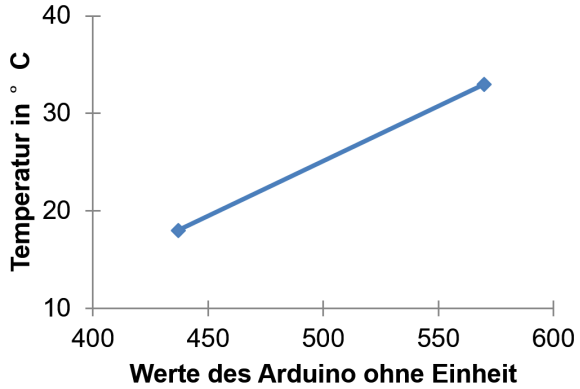
2. Temperaturverlauf für einen abkühlenden Sensor dargestellt mit dem *Seriellen Plotter*.



Zu E. Kalibration

Lösung zu Aufgabe 1 und 2

Beispiel für eine Temperaturmessung. (Je nach verwendetem Sensor kann die Ausgleichsgerade ansteigen oder auch fallen.)



Im Diagramm oben ist der Wert 437 einer Temperatur von 18°C zugeordnet und analog der Wert 570 zu 33°C.
Die Steigung der Ausgleichsgerade berechnet sich (Steigungsdreieck) mit (33°C - 18°C)/(570-437) = 15°C / 133 ≈ 0,113°C
In der Funktion f(x) = a x + b ist damit a = 0,113°C festgelegt.

Setzt man in die Funktion eines Wertepaares ein, z. B. x=437 und f(x)=18°C ergibt sich b wie folgt:
18°C = 0,113°C · 437 + b. Umstellen nach b ergibt b = 18°C - (0,113°C · 437) ≈ -31,4°C

Zusammengefasst: f(x) = 0,113°C · x - 31,4°C liefert für die Messwerte des Arduinos die Temperatur in °C zurück.

Lösung zu Aufgabe 3

Die Funktion ist (ohne physikalische Einheit) im Programm wie folgt einzufügen:

```
…
    void loop() {
      int sensorValue = 0.113 * analogRead(A0) - 31.4;        // analogRead(A0) entspricht dem x-Wert der Funktion f(x)= 0,113 x - 31,4
      Serial.print("Temperatur in °C: ");                     // Mit dieser zusätzlichen Zeile, wird die Ausgabe im Seriellen Monitor übersichtlicher.
      Serial.println(sensorValue);
      delay(1);
    }
```