



Co-funded by the
Creative Europe Programme
of the European Union

Project 2020-1-TR01- KA201-094533



Der Schlüssel zum globalen Leben,
Digitaler Wandel der Natur



Gesamtdauer: 2 oder 3 Stunden



Alter des Schülers: 14–18 Jahre



- Anwendungsbereich:
- Klimawandel
- Physik
- Elektronik



Schlüsselwörter: Temperatur,
Datenanalyse, Klima, Arduino,
Technologie



G1 – Temperaturüberwachungssystem mit Arduino UNO



- Modul
- Globale Erwärmung

G1 – Deutsche Version

Materialien:

Papier (A3)

(Buntstifte)

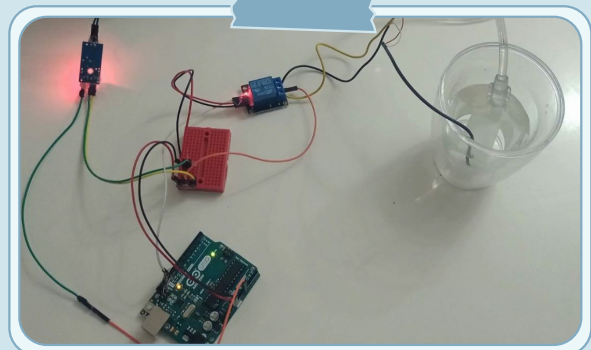
Laptop

Arduino (www.arduino.cc)

Steckbrett

Temperatursensor (LM35 oder alternativ)

Überbrückungsdrähte



- Anmerkungen:
- Größe jeder Gruppe: 3-4 Schüler.
- Phase 1: Machen Sie es zunächst in einer virtuellen Umgebung.
 - o Verwenden Sie Tinkercad (Schaltkreise):
 - o Skizzieren Sie eine übersichtliche, detaillierte Skizze des Aufbaus: inklusive Temperatursensor, Arduino und Steckbrett.
 - o Verbinden Sie alles miteinander, verwenden Sie dabei unterschiedliche Kabelfarben und verwenden Sie die richtige Kabelfarbe für Plus- und Minuspol.
 - o Erstellen Sie das Programm auf der Rennstrecke in Tinkercad und testen Sie es zunächst online.



@digitalchangeon

Einführung

Wir bauen eine erschwingliche, tragbare DIY-Installation zur Überwachung von Temperatur und Lichtintensität. Wir verwenden ein Arduino UNO zum Sammeln von Daten und verbinden es mit einem Computer, um die Daten in Excel anzuzeigen. Um diese Aktivität durchführen zu können, muss der Lehrer über Grundkenntnisse im Bereich Arduino, Arduino IDE und die grundlegende Verwendung von Excel verfügen.

Wetter und Klima sind in der heutigen Gesellschaft von großer Bedeutung. Die zunehmende Computerleistung ermöglicht es uns, Wettermodelle mit immer höheren Auflösungen auszuführen, aber dies führt zu einem Bedarf an mehr lokalen Wetterdaten. Stellen Sie sich ein Land vor, in dem jede Schule das Wetter überwacht. Dann können wir sehr genaue Bodendaten erhalten (Bild 1).

Messungen sind nicht nur wichtig, Sie müssen auch visualisieren, was Sie messen.

Wir bauen eine erschwingliche, tragbare DIY-Installation zur Überwachung von Temperatur und Lichtintensität. Wir verwenden einen Arduino zum Sammeln von Daten und verbinden ihn mit einem Computer, um die Daten in Excel anzuzeigen.

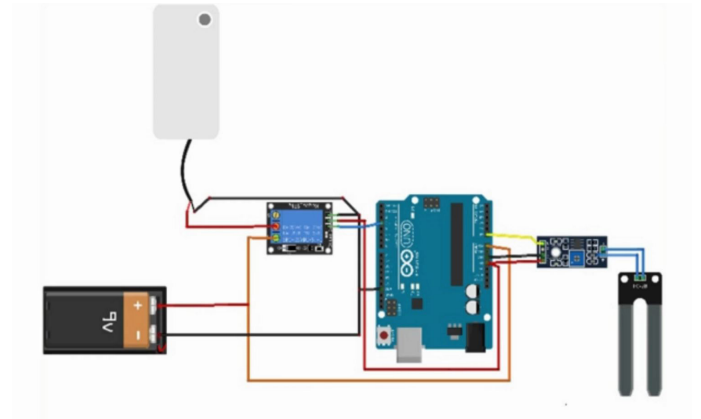


Bild 1. Tempurtare-Überwachungssystem

Überlegungen

Größe jeder Gruppe: 3-4 Schüler.

Phase 1: Machen Sie es zunächst in einer virtuellen Umgebung

Verwenden Sie Tinkercad (Schaltkreise):

Skizzieren Sie eine übersichtliche, detaillierte Skizze des Aufbaus: inklusive Temperatursensor, Arduino und Steckbrett.

Verbinden Sie alles miteinander, verwenden Sie dabei unterschiedliche Kabelfarben und verwenden Sie die richtige Kabelfarbe für Plus- und Minuspol.

Erstellen Sie das Programm auf der Rennstrecke in Tinkercad und testen Sie es zunächst online.

Phase 2: Machen Sie es wahr:

Finden Sie alle Komponenten und verbinden Sie alles.

Verbinden Sie den Arduino noch nicht mit dem PC! Der Lehrer prüft es zuerst.

Ziel der Aktivität

- Erleichtern Sie die Messung der Temperatur in regelmäßigen Abständen
- Verwenden Sie ein weithin bekanntes Analysetool wie Excel zum Sammeln und Visualisieren von Daten
- Lernen Sie, allgemein mit Sensoren zu arbeiten, um Daten zu sammeln:
- Verstehen Sie den Zusammenhang zwischen der gemessenen Spannung und der zu messenden physikalischen Größe
- Lernen Sie das Konzept der Sensorkalibrierung kennen und lernen Sie, mit einem technischen Datenblatt zu arbeiten
- Lernen Sie, in C++ zu programmieren
- Machen Sie sich mit linearen Funktionen vertraut



Bild 2. Ziel der Aktivität

Vor der Aktivität

- Erklären Sie die Aufgabe: Hintergrund, Ziel, Zeitrahmen für jeden Teil
- Teilen Sie die Klasse in Gruppen von 2 bis 4 Schülern auf, wobei jede Gruppe an ihrem eigenen Tisch sitzt. Jede Gruppe hat einen Laptop, Papier und Stifte.

ÜŞÜK MALİYETLİ SULAMA SİSTEMİ

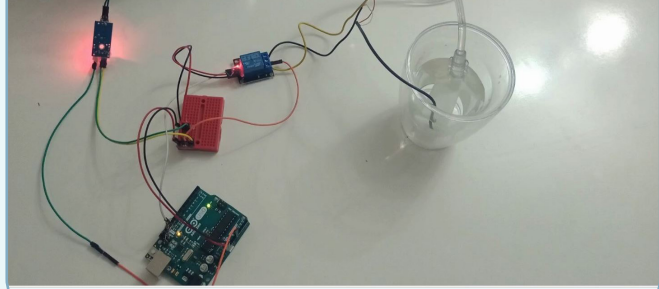


Bild 3. Mit einem Löffel vermischen

Lasst uns beginnen

1 Schaltungsentwurf (30 Minuten)

- Zeichnen Sie die Schaltung auf eine Seite.
- Sie können stattdessen Tinkercad Circuits verwenden, um es online zu zeichnen. Was Sie brauchen ist (Bild 3 und 4).
- ein Arduino
- ein Steckbrett
- der Temperatursensor (LM35 oder TMP36)



Bild 4. Materialien

2 Verbinde es!

- Positionieren Sie den Temperatursensor auf dem Steckbrett und achten Sie darauf, dass er richtig herum angebracht ist, damit die Stifte nicht kurzgeschlossen werden.
- Der Sensor benötigt zum Betrieb Spannung. Es kann der 5-V-Pin des Arduino verwendet werden. Überprüfen Sie die Abbildung auf die korrekte Pin-Anordnung. Verbinden Sie Pin 1 mit dem 5V-Pin des Arduino. Verbinden Sie den GND mit dem GND des Arduino.
- Der Arduino muss auf die Messwerte des Sensors „lauschen“. Dies ist ein analoger Eingang. Sie müssen einen der analogen Eingangspins des Arduino verwenden. Verbinden Sie den mittleren Pin des Sensors mit A0.

3 Programmierung: Anzeige der Ausgabe des Sensors

Wir beginnen zu lesen, was Ihr Arduino vom Sensor empfängt (den INPUT) und zeigen es im seriellen Monitor an. Dies erreichen Sie durch die Programmierung in C++. Wir beginnen mit einem einfachen Programm, das jede Sekunde den Sensorwert liest und ihn auf der Computerkonsole ausgibt (Bild 5).

```

1 int value;
2
3 void setup() {
4     Serial.begin(9600);
5 }
6
7 void loop() {
8     value = analogRead(A0);
9     Serial.println(value);
10    delay(1000);
11 }

```

Picture 5. C++ codes

- Um die Spannung an Pin A0 zu lesen, verwenden Sie `analogRead(A0)`.
- Sie speichern die Spannung in einer Variablen namens „Wert“. In dieser Programmiersprache (C++) müssen Sie beim Erstellen der Variablen immer angeben, welche Art von Informationen Sie in einer Variablen speichern möchten. Das Erstellen der Variablen wird als „Deklarieren“ bezeichnet und geschieht in Zeile 1 des Programms. In diesem Fall enthält die Variable eine ganze Zahl, die als „Ganzzahl“ bezeichnet wird. Daher der `int`-Wert ganz oben im Code.

Tabelle 1. Andere Arten von Informationen in C++

| | |
|---------------------------|--|
| <code>int</code> | Ganze Zahl (0, 1, 2, -3, 19839, -78, ...) |
| <code>schweben</code> | Dezimalzahl (0,13, 713,24, -3,0,) |
| <code>verkohlen</code> | Ein einzelnes Zeichen („a“, „B“, „c“, „D“, „0“, „*“, „µ“, ...) |
| <code>Zeichenfolge</code> | Text („Hallo Welt!“, „Das ist ein Test“, „123“, ...) |

4 Kommunikation zwischen Arduino und PC

- Wie Sie sich vielleicht erinnern, „laden“ Sie ein Programm auf den Arduino hoch, und das Programm läuft auf dem Arduino, nicht auf dem Computer. Es ist der Arduino, der den Sensorwert liest, aber der Computer muss diesen Wert auf dem Bildschirm anzeigen. Der Arduino muss diese Informationen zunächst an den Computer senden.
- Um eine Verbindung zwischen dem Arduino und dem Computer einzurichten, verwenden Sie `Serial.begin(9600)` im Setup-Block (da Sie die Verbindung nur einmal, ganz am Anfang, einrichten müssen). Die Zahl in Klammern () sagt dem Computer, wie viele Bits er pro Sekunde senden wird. Dies wird als „Baudrate“ bezeichnet. Je höher diese Zahl, desto mehr Daten werden pro Sekunde gesendet, desto höher ist aber auch die Wahrscheinlichkeit von Fehlern bei der Kommunikation (dass aus einer 0 versehentlich eine 1 wird und umgekehrt). Für manche Anwendungen ist es wichtig, eine hohe Geschwindigkeit zu erreichen, beispielsweise beim 3D-Druck. 9600 ist für die von uns genutzten Anwendungen mehr als ausreichend.
- Nachdem der Arduino die Kommunikation mit dem Computer aufgebaut und den Wert des mittleren Pins gelesen und in einer Variablen gespeichert hat, ist es an der Zeit, diesen Wert an den Computer zu senden.
- `Serial.println(value)` bewirkt, dass eine Zeile auf dem Computer gedruckt wird (daher `println`, übersetzt als „Zeile drucken“). Setzen Sie in Klammern, was gedruckt werden soll.



- Die Ausgabe finden Sie im „Serial Monitor“. In Tinkercad finden Sie dies, indem Sie unten im Code-Bildschirm auf „Serieller Monitor“ klicken (Bild 6).
- In der Arduino IDE können Sie es aufrufen, indem Sie im Menü „Extras“ auf „Serieller Monitor“ klicken. Oder klicken Sie auf das Symbol (Bild 7).



Bild 6. Serieller Monitor in Tinkercad

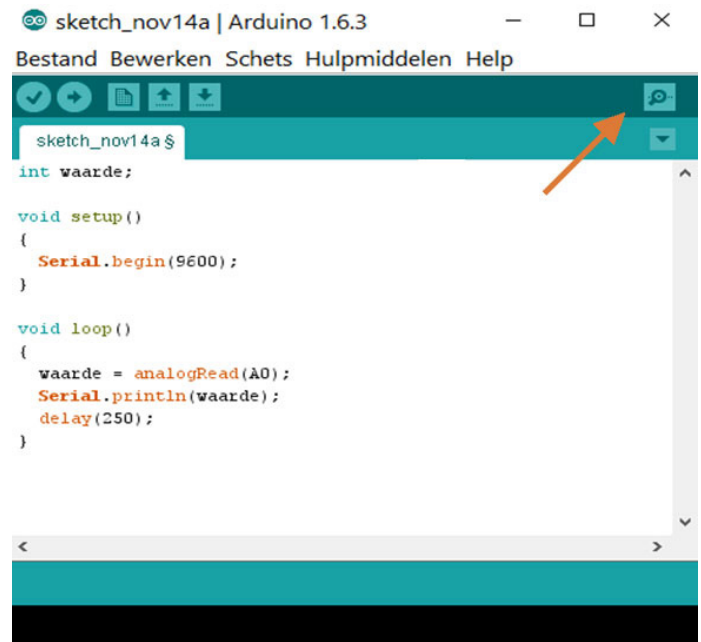


Bild 7. Serieller Monitor in der Arduino IDE

5 Programmierung: Kalibrierung des

Sie erhalten von Ihrem Sensor einen bestimmten Wert, aber was bedeutet dieser Wert genau? Deshalb ist es wichtig, dass Sie Ihren Sensor kalibrieren, damit Sie einen bestimmten Wert richtig interpretieren können. In diesem Fall: Was bedeutet der Wert 523? Ist es „warm“ oder „kalt“? Dieses Mal müssen wir im Datenblatt des LM35/TMP36-Sensors nachsehen, um die Spezifikationen zu finden. Wenn wir sie finden, wissen wir, wie wir den Messwert in die richtige Temperatur umrechnen können.



- Fügen Sie das



- Suchen Sie im Internet nach den Spezifikationen für den Temperatursensor. Notieren Sie in der Tabelle unten, was Sie gefunden haben.

| Zustand | Theoretischer Messwert |
|----------------------------|------------------------|
| Mindesttemperatur | |
| Maximale Temperatur | |
| Steigung (mV/Grad Celsius) | |
| Genauigkeit | |
| Vout bei 150°C | |
| Vout bei 0°C | |



Jetzt müssen wir eine Formel finden, um unsere Temperatur zu berechnen. Aber wie funktioniert `analogRead()`?

Arduino übersetzt die gemessene Spannung am analogen Pin (0-5V) immer in eine Zahl (0-1023).

0V entspricht 0

5V entspricht 1023

Sie müssen also durch 1023 dividieren und erneut mit 5 multiplizieren, um vom Wert zur Spannung zu gelangen.

Der nächste Teil hängt vom verwendeten Sensor ab. Achten Sie also darauf, die richtigen Spezifikationen zu verwenden! Unser Beispiel unten gilt nur für LM35:

Das wissen wir aus den Angaben im Datenblatt

0°C entspricht 0V

150°C entsprechen 150mV = 0,150V

5000 °C entsprechen 5000 mV = 5 V

Jede 10 mV (0,01 V) entspricht also 1 °C.

Sie müssen die gemessene Spannung mit 100 multiplizieren, um die °C zu erhalten

Die Formel zur Berechnung der Temperatur aus dem Wert als Ergebnis von `analogRead()`

Temperatur = $(5,0 * \text{Wert} * 100,0) / 1023,0$



Unser Programm zur Messung und Berechnung der Temperatur mit einem LM35-Sensor (Bild 8)



Seien Sie vorsichtig, Dezimalzahlen werden mit einem Punkt (.) und nicht

```

1 int value = 0;
2 float temperature = 0;
3
4 void setup() {
5   Serial.begin(9600);
6 }
7
8 void loop() {
9   value=analogRead(A0);
10  Serial.print("our sensor reads: ");
11  Serial.println(value);
12  temperature = 5.0 * value * 100.0 / 1023.0;
13  Serial.print("this means in degrees Celsius: ");
14  Serial.println(temperature);
15  delay(1000);
16 }
```

Bild 8. DBerechnen Sie die Temperatur



Für die klugen Schüler, die schneller fertig waren:
Berechnen Sie es für einen anderen Sensortyp wie TMP36, TMP37 oder den sehr häufig verwendeten DHT11 oder DHT22.

6

Erfassen Sie die Daten in Excel

Über den Serial Monitor können wir die Werte bereits ausdrucken. Jetzt sehen wir, wie wir diese Daten stattdessen nach Excel streamen können. Das könnte nützlich sein.

Es gibt keine Änderung am Eingang oder an Ihrer Schaltung. Wir müssen lediglich ein Add-In in Excel aktivieren und konfigurieren. Dadurch werden die Daten erfasst, die an den seriellen Monitor gesendet werden, und diese Werte werden in einer Reihe von für diesen Zweck konfigurierten Zellen angezeigt.

Das von uns benötigte Add-In heißt „Microsoft Data Streamer for Excel“ und ist standardmäßig in den neuesten Versionen von Microsoft Excel vorhanden. Sie müssen es noch aktivieren.

Klicke auf:

„Datei“ >> „Optionen“ >> „Add-Ins“

Klicken Sie auf den Pfeil neben „Verwalten: Excel-Add-Ins“ >> „COM-Add-Ins“ >> „Start“.

Aktivieren Sie das Kontrollkästchen neben „Microsoft Data Streamer für Excel“.

OK'

Nun erscheint ganz rechts im Menüband eine neue Registerkarte mit dem Namen „Data Streamer“. Öffnen Sie nun den Reiter „Data Streamer“. Klicken Sie auf „Gerät verbinden“ und wählen Sie „Arduino Uno (COMX)“.

Eine neue Arbeitsmappe wird erstellt. Um Daten zu streamen, klicken Sie auf „Daten starten“. Hier werden nun jede Sekunde Zeit und Temperatur ausgedruckt. Die Zeit wird von Excel automatisch hinzugefügt. Im Arbeitsblatt „Einstellungen“ können Sie unter anderem anpassen, wie viele Datenzeilen sichtbar sein sollen (max. 500). Stellen Sie nun die Anzahl der Datenzeilen auf 50 ein.

Ändern Sie das Programm so, dass nur die Werte gedruckt werden, nicht der Text. Geben Sie nun beide Werte durch ein Komma getrennt in derselben Zeile aus.

```
Serial.print(Wert);
```

```
Serial.print(",");
```

```
Serial.println(temperatur);
```

Dadurch wird eine Zeile gedruckt, die aus der Temperatur, einem Komma und der Luftfeuchtigkeit besteht. Wenn Sie das Komma dazwischen nicht verwenden, würde Excel dies als einen großen Wert und nicht als zwei separate Werte behandeln. Beachten Sie, dass die an den seriellen Monitor gesendeten Daten auf die gleiche Weise formatiert sind.

7

Grafiken

Erstellen Sie zwei Liniendiagramme, eines für den Messwert und eines für die Temperatur. Wenn alles gut gelaufen ist, sollten diese Grafiken live aktualisiert werden.

8

Speichern

Sie können Daten auch zur späteren Verwendung speichern. Klicken Sie dazu auf den Button „Daten erfassen“. Nachdem Sie die Aufnahme beendet haben, werden Sie gefragt, wo Sie die Datei speichern möchten. Das Format dieser Datei ist „.csv“, kurz für „Comma Separated Values“. Sie können dieses Format anschließend in vielen datenbezogenen Programmen lesen, einschließlich Excel.

9

Jetzt basteln, erweitern



- Verwenden Sie einen zweiten Sensor, z.B. ein LDR, Bodenfeuchtesensor, ... Fügen Sie die Sensorwerte, die Sie gelesen haben, in Excel hinzu. Berechnen Sie für den Bodenfeuchtesensor die relative Luftfeuchtigkeit.
- Fassen Sie alles in einer grafischen Visualisierung zusammen

Bewertung

Auswertung

Das Design der Schüler kann innerhalb der Schule ausgestellt werden. Durch die Diversifizierung der verwendeten Abfallmaterialien können unterschiedliche Produkte hergestellt werden.

| Ziele | Muss verbessert werden (1) | Mittel (2) | Gut (3) | Sehr gut (4) |
|--|----------------------------|------------|---------|--------------|
| Ein Problem so (neu) formulieren, dass es von einem Computer oder einem anderen Werkzeug gelöst werden kann. | (.....) | (.....) | (.....) | (.....) |
| Aktive Teilnahme | (.....) | (.....) | (.....) | (.....) |
| Die Erstellung einer Reihe von Anweisungen, um von einem Ausgangspunkt (= Algorithmus) aus ein Ziel zu erreichen, gelingt selbstständig. | (.....) | (.....) | (.....) | (.....) |
| Die Auswahl und Umsetzung nützlicher Informationen aus der angegebenen Quelle gelingt selbstständig. | (.....) | (.....) | (.....) | (.....) |
| Kritisches Denken | (.....) | (.....) | (.....) | (.....) |

Links

Farbenfrohe Welt. (2018). Wie man für wenig Geld ganz einfach einen Pinguin, Eis und ein Iglu bastelt / DIY. <https://www.youtube.com/watch?v=Es-rCelq6YU>

Cristofari, R., Liu, X., Bonadonna, F., Cherel, Y., Pistorius, P., Le Maho, Y., . . . Trucchi, E. (2018). Klimabedingte Verbreitungsverschiebungen des Königspinguis in einem fragmentierten Ökosystem. *Nature Climate Change*, 8(3), 245-251.

ScienceBuddies. (2020). Meeresströmungen: Modellierung des „globalen Förderbandes“ in Ihrer Küche. Abgerufen am 10.10.2022 von https://www.sciencebuddies.org/science-fair-projects/project-ideas/OceanSci_p012/ocean-sciences/ocean-currents-modeling-global-conveyor-belt

- <https://www.sciencebuddies.org/stem-activities?s=global%20warming>
- <https://www.sciencebuddies.org/stem-activities/polar-ice-caps-melting>
- https://www.sciencebuddies.org/science-fair-projects/project-ideas/OceanSci_p015/ocean-sciences/will-ice-melting-at-poles-cause-sea-levels-to-rise
- <https://www.tinkercad.com/things/c3BkCJdQxel>
- <https://www.tinkercad.com/things/9UeZJTri0zD>
- <https://www.youtube.com/watch?v=Gkw45JaEQio>
- <https://www.youtube.com/watch?v=ztQYbRwBboU>
- <https://science.howstuffworks.com/environmental/earth/oceanography/ocean-current.htm>
- <https://web.ics.purdue.edu/~braile/edumod/convect/convect.htm>