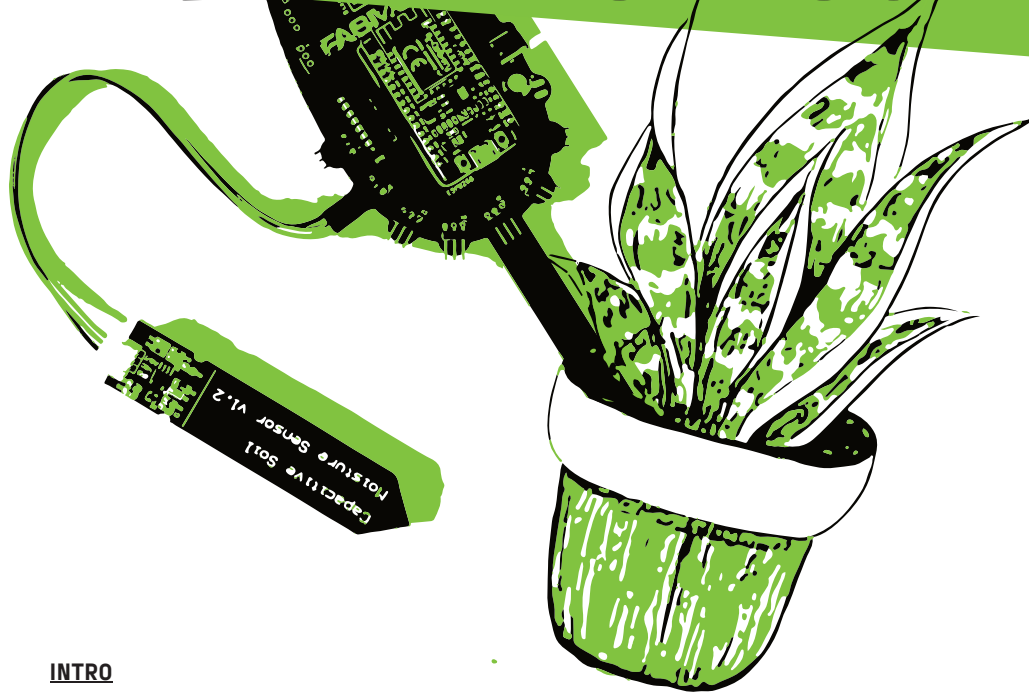


PFLANZENSENSOR



INTRO

Da unsere Zimmerpflanzen nicht so kommunikativ sind wie wir, geht es ihnen manchmal nicht so gut, wie sie es verdient hätten. Dieses Projekt soll helfen, Pflanzen eine Stimme zu geben! Mittels Sensoren sollen lebensnotwendige Daten erhoben, verarbeitet und ausgewertet werden, um die richtigen Schlüsse daraus zu ziehen. Gießen oder eben gerade nicht, mehr Sonne, mal wieder lüften oder einfach nur streicheln.

LEVEL

Fortgeschrittene –
Experten

ALTERSGRUPPE

ab Klassenstufe 9

DAUER

2-3 Tage á 5 bis 6 Stunden

ANZAHL PERSONEN

bis zu 12 Teilnehmende,
Kleingruppen, Single



TECHNOLOGIE

Mikrocontroller / 3D-Druck

SPECIAL SOFTWARE

MakeCode / Arduino IDE / Tinkercad /
Slicing-Software

NOTWENDIGE VORKENNTNISSE WORKSHOPLEITENDE

- Programmieren (für Calliope, Arduino, ESP8266)
- 3D-Druck + Software
- Elektrotechnik

MATERIAL UND MASCHINEN

- Laptops mit Maus und stabiler Internetverbindung
- 3D-Drucker
- Calliope Gruppensatz + USB-Kabel
- Bodenfeuchtesensor + Kabel
- Pflanzensensor-Platine
- 2 Gefäße mit (trockener/nasser) Erde
- Multimeter

MATERIAL ZUR VORBEREITUNG

- Calliope Lernkarten
- [▷1. Einführung C-Programmierung](#)
- [▷2. Benötigte Daten](#)



Fabmobil: Pflanzensensor

NOTWENDIGE VORKENNTNISSE TEILNEHMENDE:

Der Workshop soll Jugendlichen einen ersten Einstieg in die Sensorik vermitteln. Mithilfe verschiedener "Übersetzer" soll exemplarisch mit einer Pflanze kommuniziert werden, um sich dem Thema zu nähern. Dabei werden bestimmte, für die Pflanze essenzielle, Parameter gemessen und mithilfe eines Mikrochips in eine für Menschen verständliche Sprache übersetzt. Zum Beispiel; "Gieß mich", wenn der Feuchtigkeitsgehalt der Erde einen bestimmten Wert unterschreitet.

Einen guten Start in dieses Kapitel ermöglicht ein Calliope. Dieser Mikrocontroller wurde speziell für junge Menschen bzw. Programmier-Neulinge entwickelt. Auf einer Leiterplatte sitzt dort nämlich nicht nur der Chip, sondern auch verschiedenste und fest verbaute Aktoren, Sensoren und Anschlussmöglichkeiten für weitere elektronische Komponenten. Quasi ready to use. Per USB-Kabel kann der Chip vom Rechner aus programmiert werden und bietet gleichzeitig eine Möglichkeit der Energieversorgung.

Ein ESP8266 ist im Grunde genommen dem Calliope recht ähnlich, nur weicht die Art und Weise wie er programmiert wird, die Onboard Steuerungselektronik, sowie dessen Kosten und das Handling etwas vom Calliope ab. Der Workshop soll diese Varianzen aufzeigen, einen ersten Eindruck und verschiedene Einstiegsmöglichkeiten anbieten, damit die Neugierde der Jugendlichen für die Welt der Mikroelektronik geweckt wird.

TECHNISCHE AUSSTATTUNG AM WORKSHOP-ORT

Für den Workshop werden die Teilnehmenden in Kleingruppen von jeweils zwei Personen eingeteilt. Für jede dieser Gruppen sollte ein eigener Arbeitsplatz bereitgestellt werden, der mit einem Computer oder Tablet ausgestattet ist. Die Arbeitsplätze sollten mit Strom versorgt sein und eine stabile Internetverbindung aufweisen. Ein 3D-Drucker ist außerdem empfehlenswert, da Schutzhüllen für den Sensor damit gut herstellbar sind. Bausätze könnten ggf. aber auch mitgebracht werden.

KENNELERNRUNDE

Zu Beginn des Workshops erfolgt ein gegenseitiges Kennenlernen der Beteiligten. Gleichzeitig besteht die Möglichkeit, sich mit dem Raum und der technischen Ausstattung vertraut zu machen. Gemeinsam werden Werte und Regeln für die bevorstehende Workshopzeit festgelegt. Anschließend wird das Ziel des Workshops vorgestellt.

- Der Fabmobil e.V. hat in seiner langjährigen Workshop erfahrung best-practice Anwendungen aufgearbeitet und für Beginner in sogenannten Lernkarten dokumentiert. Auch zum Calliope gibt es sie auf der Fabmobil-Seite zum [>3. Download](#).

AUFGABE 1 // CALLIOPE KENNENLERNEN

Die erste Aufgabe besteht darin, sich das Calliope mit Hilfe dieser Lernkarte selbst zu erschließen. Eine grobe Orientierungshilfe und erste Schritte können auch in erklärender Weise von den Betreuenden übernommen werden.

AUFGABE 2 // PFLANZENSSENSOR BAUEN

Die nächste Aufgabe baut auf die vorhergehende auf und fordert die Teilnehmenden auf, mithilfe des Calliopes einen Pflanzensensor zu bauen. Die dazu benötigten Feuchtesensoren und Kabel werden vom betreuenden Personal ausgegeben.

Außerdem wird beispielhaft gezeigt, wie ein funktionierendes Pflanzensensorprogramm auf dem Calliope aussehen kann. Tipps und Hilfestellungen werden den Teilnehmenden in dieser Zeit simultan gegeben. Wenn der Calliope Pflanzensensor funktioniert, könnt ihr ihn mit feuchter und trockener Blumenerde kalibrieren und testen.

BONUS

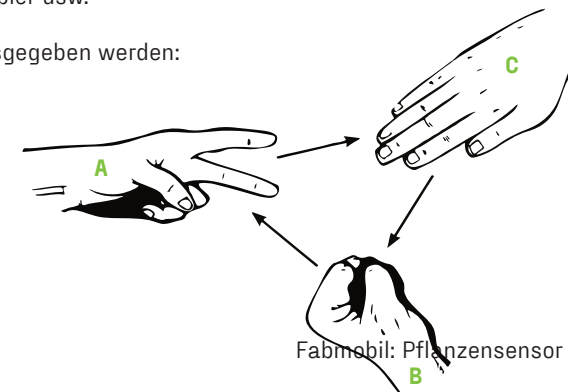
Falls einige früher als andere fertig sind, gibt es folgende Bonusaufgabe:
Sucht eine andere Gruppe, die auch schon fertig ist. Ziel soll es sein, ein Schnick-Schnack-Schnuck Spiel zu programmieren, das mit zwei Calliopes gegeneinander gespielt werden kann.

TIPPS

Zum Senden und Empfangen der Daten zwischen den Calliopes wird das Funkmodul benötigt. Es ist hinter dem Button im Online-Editor zu finden. Zu Beginn ist auf dem Calliope eine Variable festzulegen (z.B.: A = Schere, B = Stein, A + B = Papier).

Dann soll diese Wahl an den anderen Calliope gesendet werden. Parallel empfängt das eigene Calliope die Wahl des anderen Calliope und speichert sie in einer anderen Variable. Beide Variablen müssen dann miteinander verglichen werden:
Stein schlägt Schere, Schere schlägt Papier usw.

Danach muss nur noch das Ergebnis ausgegeben werden:
Wer hat gewonnen, wer verloren?



EINFÜHRUNG

Zu Beginn des zweiten Workshoptags ist es hilfreich, den Tagesplan gemeinsam zu besprechen. *Wie ist der Tag organisiert? Welche offenen Fragen müssen noch geklärt werden? Sind alle Teilnehmenden auf einem ähnlichen Stand? Wer benötigt Unterstützung und bei welchen Themen? Und wie kann diese Unterstützung am besten bereitgestellt werden?*

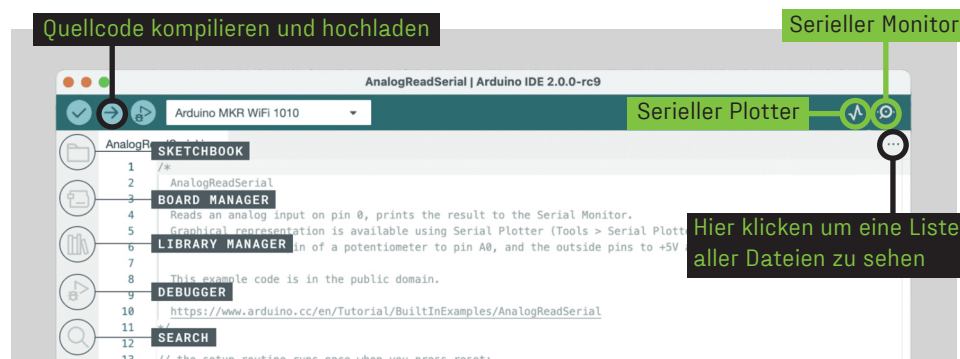
AUFGABE 1 // KENNENLERNEN DER ARDUINO IDE

Nachdem mit dem Calliope der Einstieg ins Programmieren gefunden wurde, geht es mit der Arduino-IDE und der Programmiersprache C++ weiter. Die Einteilung der Gruppen ist dabei weiterhin aktuell. Jede Gruppe benötigt folgende Teile:

- Laptop
- ESP8266-Microcontroller
- MicroUSB-Kabel um den Microcontroller mit dem PC zu verbinden
- die Software „Arduino IDE“ (im besten Fall schon auf dem Laptop vorinstalliert)
- den Bodenfeuchtesensor, der schon verwendet worden ist

Zu Beginn muss der Arduino-Ordner [▷4. hier](#) heruntergeladen und lokal in den Dokumente-Ordner auf dem Rechner gespeichert werden. Ist die Software schon auf dem Rechner installiert, ist dieser Ordner schon angelegt. Wenn das der Fall ist, muss dieser durch den heruntergeladenen ersetzt werden, um die aktuelle Version des Quellcodes zu garantieren.

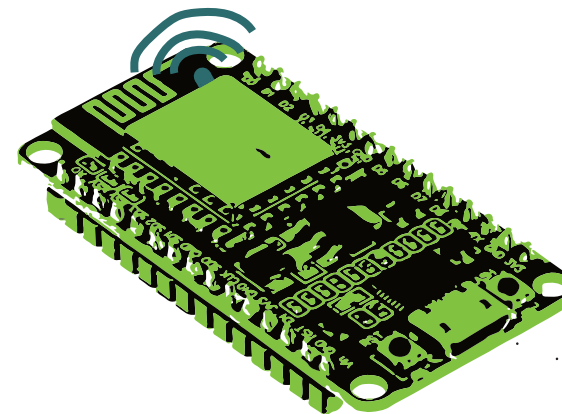
Im Anschluss kann das Programm Arduino IDE ausgeführt werden. Im folgendem Bild sind die wichtigsten Bereiche beschrieben:



FOLGENDE AUFGABEN DIENEN ALS ÜBUNG ZUM PROGRAMMIEREN DES ESP8266

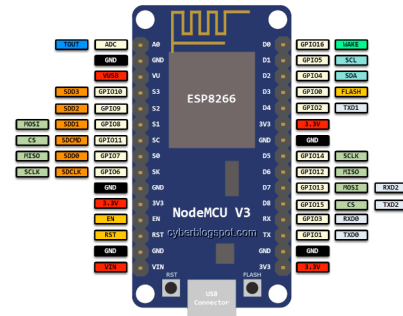
AUFGABE 2 // LED BLINKEN LASSEN

- öffne LED.ino in der Arduino IDE
- verbinde den ESP8266 per USB Kabel mit dem Laptop
- überspiele das Programm auf den ESP8266:
 - klicke unter Werkzeuge auf Board, wähle dann ESP8266 Boards und danach **LOLIN(WEMOS) D1 R2 & mini**, klicke anschließend unter Werkzeuge auf Port und wähle den COM-Port mit der höheren Nummer aus (Falls mit diesem das Überspielen nicht funktioniert, kann es auch sein, dass der andere Port der richtige ist).
- überprüfe, ob die interne LED an ist
- erweitere das Programm so, dass die LED immer abwechselnd 1 Sekunde an und 2 Sekunden aus ist. Dazu werden zusätzliche folgende zwei Befehle benötigt: **digitalWrite(LED, HIGH)** schaltet die LED aus und **delay(3000)** pausiert das Programm für 3000ms, also 3 Sekunden



AUFGABE 3 // FEUCHTESENSOR AUSLESEN

- Schließ den Bodenfeuchtesensor mit Hilfe von Jumperkabeln an den ESP8266 an: rot an 3V3 (Versorgungsspannung), schwarz an GND (Masse), Gelb an AO (Sensorwert): die Pins sind beschriftet.
- öffne [feuchtesensor.ino](#) in der Arduino IDE
- überspiele das Programm auf den ESP8266
- überprüfe die Messwerte im seriellen Monitor und im seriellen Plotter
- Wie verändern sich die Werte, wenn man den Sensor in die Hand nimmt?
- Der Wert liegt immer zwischen 0 und 1024.
- Die Aufgabe besteht nun darin, die interne LED entsprechend dem Wert des Analogsensors blinken zu lassen: Bei einem Sensorwert von 100 soll die LED für 100ms an und dann für 100ms aus sein. Bei einem Sensorwert von 1000 soll die LED für 1000ms an und dann für 1000ms aus sein.



AUFGABE 4 // SCHWELLVERTVERGLEICH

- öffne [schwellwert.ino](#) in der Arduino IDE
- vergleiche die Variable `schwellwert` mit der Variable `sensorwert`
- Wenn der Sensorwert größer als der Schwellwert ist, soll die LED an sein.
- Wenn der Sensorwert kleiner oder gleich dem Schwellwert ist, soll die LED aus sein.
- Wie du einen solchen Vergleich programmierst kannst du [>5. hier](#) nachlesen.

BONUS

Sollten Teilnehmende früher fertig sein als andere, können sie sich kleinere Arduino-projekte aus dem Netz (z.B. auf der [>6. Arduino-Website](#)) raussuchen und diese umsetzen. Sollten andere Sensoren benötigt werden als bisher genutzt, sollte das mit den Betreuenden rückgesprochen werden.



TAG 3

Für einen guten Start in den Tag empfiehlt sich wieder ein Zusammenkommen und Besprechen der Tagesziele. Wer braucht wobei noch welche Hilfestellungen, wer kann das übernehmen, etc.

PFLANZENSSENSOR

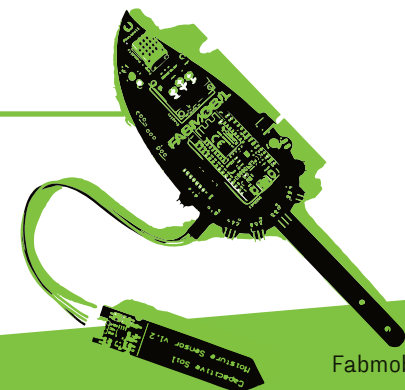
Jetzt ist es soweit! Der Pflanzensensor kann montiert werden. Dazu wird folgendes benötigt:

- ESP8266 Mikrocontroller
- Bodenfeuchtesensor
- Display
- Pflanzensensor PCB
- Laptop
- Micro USB-Kabel
- Abstandshalter und Schrauben aus Kunststoff
- 3D-gedruckte Hülle für den Bodenfeuchtesensor

AUFGABE 1 // MONTAGE DES PFLANZENSSENSORS

- Zu Beginn wird der ESP8266 auf den PCB montiert (**ACHTUNG: der USB-Port muss in Richtung des "Blattstiels" zeigen**).
- Das Display wird in die Fassung über dem ESP8266 gesteckt und zusätzlich mit Kunststoffabstandshaltern und Schrauben in den entsprechenden Löchern im PCB sowie im Display fixiert.
- Der Bodenfeuchtesensor wird an den Port X6 angeschlossen (**Hinweis: "-" = "GND"**)
- Die 3D-gedruckte Hülle schützt die empfindliche Elektronik des Sensors und sollte deshalb am oberen Teil desselben angebracht werden.
- Sollte die Hülle nicht von selbst halten, kann mit Klebeband oder Sekundenkleber nachgeholfen werden.

Pflanzensensor PCB



Fabmobil: Pflanzensensor

AUFGABE 2 // BESPIELEN DES ESP8266 MIT DEM PFLANZENSSENSORPROGRAMM

öffne die [Pflanzensensor.ino](#)-Datei in der Arduino IDE. Diese kann [▷7. hier](#) heruntergeladen werden. Der Quellcode des Pflanzensensorprogramms besteht aus vielen Dateien. Sie werden alle automatisch geöffnet und in der Leiste über dem Editor in der Arduino IDE angezeigt. Achtung: Ungezielte Änderungen im Programm können es unbrauchbar machen! Folgendes muss im Code angepasst werden:

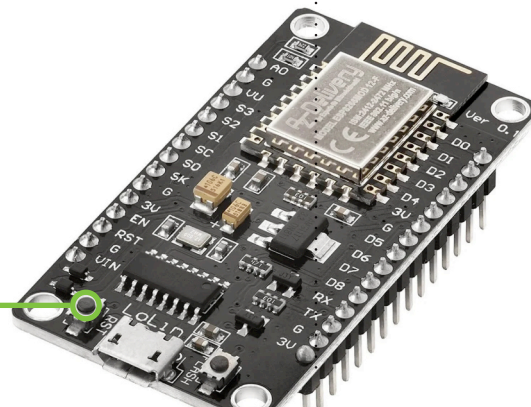
- Falls die [passwoerter.h](#)-Datei in der Leiste mit den Dateinamen nicht aufzufinden ist, dann rechts auf “...” klicken und dort aus der Liste auswählen.
- Die [passwoerter.h](#)-Datei öffnen und die Daten eintragen, die zu dem WLAN gehören, in dem der Pflanzensensor betrieben werden soll.
- Es können bis zu drei verschiedene WLANs konfiguriert werden. Die Variablen wifiSsid1, wifiSsid2 und wifiSsid3 speichern den Namen des WLANs und die Variablen wifiPassword1, wifiPassword2 und wifiPassword3 das Passwort. Der eigentliche Name des WLANs bzw. das Passwort muss zwischen den Anführungszeichen eingetragen werden.
- HINWEIS (Fabmobil): Die ausgefüllten Fabmobil WLAN Daten nicht löschen oder verändert da sonst keine Verbindung mit dem businternen WLAN mehr möglich ist und somit keine Änderung über das Webinterface mehr vorgenommen werden können
- Sollte es geplant sein, den Sensor mobil einzusetzen, ist es sinnvoll, die Hotspot-Daten des Handys einzutragen.
- Anschließend kann der Quellcode des Programms, wie schon in den vorangegangenen Übungen, kompiliert und auf den ESP übertragen werden.

Abschließend müssen noch ein paar Extradateien überspielt werden.

- dafür [Strg] + [Shift] + [P] gleichzeitig drücken
- “upload” eingeben
- dann “Upload LittleFS to Pico/ESP8266/ESP32” auswählen
- den seriellen Monitor öffnen und den ESP neu starten.
- HINWEIS: Auf die Zeile Start von Wifi-Modul... achten: Jetzt versucht sich der Pflanzensensor mit dem WLAN zu verbinden. Wenn es funktioniert hat, wird auf dem Display und auch im seriellen Monitor die IP Adresse angezeigt, die der Pflanzensensor bekommen hat. IP Adressen bestehen aus vier Zahlen getrennt durch einen Punkt und sehen z.B. so aus: 192.168.9.199. Diese Adresse bitte notieren.
- Zum Webinterface des Sensors gelangt man nun, indem man die notierte IP-Adresse in die Adresszeile im Browser des Laptops eingibt.



Resetknopf



AUFGABE 3 // KENNENLERNEN DES WEBINTERFACES

Nun kann das Webinterface des Pflanzensensors unter die Lupe genommen werden. Was stellt es dar? Wo ist die Adminseite zu finden und welche Einstellungen können dort getätigt werden.

TIPP: Im [▷8. Github](#) des Fabmobil sind weitere Erklärungen zu finden.

AUFGABE 4 // KALIBRIEREN DER SENSOREN

Damit der Bodenfeuchtesensor richtig funktioniert, muss er noch richtig kalibriert werden:

- Dafür muss der Bodenfeuchtesensor zunächst in sehr feuchte Erde gesteckt werden. ACHTUNG: Um ein genaues Ergebnis zu erzielen, sollte der Sensor immer einigermaßen gleich tief in der Erde stecken. Ein Sensor, der nur halb in der Erde steckt, wird andere Messwerte anzeigen als einer, der ganz in der Erde steckt. Ein weißer Strich auf dem Sensor kurz unterhalb des Gehäuses markiert die Stelle, bis zu der der Sensor in die Erde gesteckt werden sollte.
- Auf der Adminseite des Pflanzensensors, kann man nun unter dem “Bodenfeuchte”-Bereich den „aktuellen absoluten Messwert“ ablesen. Dieser muss nun darunter bei Maximalwert eingetragen werden.
- Anschließend den Bodenfeuchtesensor in eine sehr trockene Erde stecken.
- Wiederholt den aktuellen absoluten Messwert kopieren und diesmal unter “Minimalwert” eintragen.
- Anschließend sind die geänderten Werte zu speichern

Jetzt ist der Bodenfeuchtesensor vorläufig kalibriert. Andere Erde wird jedoch andere Werte erzeugen, sodass eine Kalibrierung am Einsatzort sehr zu empfehlen ist. Der Helligkeitssensor kann auf Wunsch nachkalibriert werden, indem man ihn für Wert eins abdeckt und für Wert zwei in die Mittagssonne hält. Dieser Schritt ist nicht zwangsläufig notwendig, da die voreingestellten Werte ein gutes Mittel darstellen.

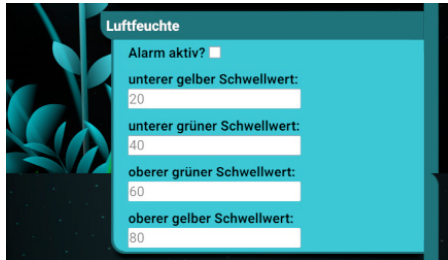


Fabmobil: Pflanzensensor

ABSCHLUSS

AUFGABE 5 // EINSTELLEN DER GRENZEN FÜR DIE LED AMPEL

Die Grenzwerte der LED Ampel für die unterschiedlichen Sensoren können auf der Admin-seite des Pflanzensensors eingestellt werden:



Diese Schwellwerte sind folgendermaßen definiert:



Abhängig davon, welche Pflanze überwacht werden soll, können ihre Bedürfnisse über die Einstellungen der Skale visualisiert werden.

BONUS // BENACHRICHTIGUNG PER MAIL ODER TELEGRAM

Sollte sich Leerlauf für besonders flinke Teilnehmende ergeben, können sie sich mit einer Erweiterungsfunktion beschäftigen. Der Benachrichtigung per Email oder Telegram, sollten bestimmte voreingestellte Werte unter- oder überschritten werden. Dafür muss folgendes getan werden:

- In der Arduino IDE die Datei `einstellungen.h` öffnen und in der Zeile `#define MODUL_WEBHOOK false` das "false" zu einem "true" ändern. Dies aktiviert das Webhook-Modul, welches für die Benachrichtigung notwendig ist. Eine detaillierte Anleitung ist unter diesem [Link](#) zu finden.

PRÄSENTATION

Zu guter Letzt können sich die Teilnehmende gegenseitig ihre Arbeitsergebnisse vorstellen und ein kurzes Feedback darüber abgeben, was sie besonders interessant fanden. Wo gab es Schwierigkeiten, war eine Aufgabe unlösbar und wie könnte das beim nächsten Mal besser laufen? Welches Schlüsselerlebnis würden die Teilnehmenden gern mit Außenstehenden teilen?

ABSCHLUSS

Zum Abschluss des Workshops wird zunächst gemeinsam aufgeräumt. Anschließend können Halbleiter-Fachkräfte eine spannende Vorstellung zu Berufsaussichten in der Branche geben.



WEITERLERNEN

ARDUINO

- Das Internet ist voll mit Tutorials über Arduinos!
- Alle verbauten Teile können für neue Projekte wiederverwendet werden!
- Es muss nicht immer "C" sein, wie wäre es mit [>10. MicroPython](#) oder einer [>11. grafischen Programmieroberfläche?](#)

LÖTEN

- Die [>12. Ersa Lötspitze](#) geht ins Detail.

PROGRAMMIEREN

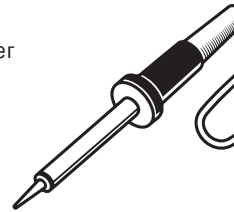
- [>13. Hour of Code](#) bietet eine Vielzahl gut vorbereiteter 45-minütiger Programmieraufgaben für Kinder.
- [>14. Scratch](#) ist eine frei verfügbare grafische Programmiersprache, welche vom MIT explizit für junge Programmieranfänger entworfen wurde.

3D DRUCK

- [>15. Grundlagen des 3D Drucks](#) von Josef Prusa gibt einen sehr guten Überblick.

CAD

- [>16. Blender](#) ist der Star am CAD-Himmel: vom kompletten Kinofilm über das Vorbereiten schlechter 3D-Scans für den Druck bis hin zum Sculpting oder dem 2D-Maskendesign ist mit dieser Open-Source-Software alles möglich, und die riesige Community bietet Unmengen an Tutorials für alle Bereiche in vielen Sprachen.
- [>17. TinkerCAD](#) ist die niedrigschwellige Variante: sehr viel weniger komplexes Interface und es läuft komplett im Browser. Für Einsteiger ideal!



FABMOBI

VEREIN ZUR FÖRDERUNG KREATIV-TECHNOLOGISCHER PRAXIS

GEFÖRDERT VON:



Diese Maßnahme wird mitfinanziert durch
Steuermittel auf der Grundlage des vom
Sächsischen Landtag beschlossenen Haushaltes.

drosos (...)

UNTERSTÜTZT VON:



BOSCH



GlobalFoundries®

xfab

infineon