

# Pflanzensensor Workshop

Willkommen beim Fabmobil Pflanzensensorworkshop! Dieses Dokument fasst die Inhalte und Aufgaben zusammen, die wir in den kommenden 3 Tagen gemeinsam bearbeiten werden und die du benötigst, damit du am Ende einen funktionierenden Pflanzensensor mit nach Hause nehmen kannst.

**Achtung:** Damit der Pflanzensensor am Ende bei dir zu Hause funktioniert musst du die WLAN Logindaten des WLANs eurer Wohnung eingeben. Du solltest also spätestens an Tag 3 den Namen und das Passwort des WLANs dabei haben, in dem du den Pflanzensensor betreiben willst bzw. vorher sicherstellen, dass du sie aus deinem Smartphone auslesen kannst (Family Link verhindert das z.B. manchmal). Sei achtsam beim Abschreiben: falls nur ein Zeichen falsch ist (Groß- und Kleinschreibung ist relevant!), wird es nicht funktionieren! Der eingesetzte Mikrocontroller kann nur 2,4GHz WLAN. Falls ihr zu Hause also 5 Ghz WLAN und 2,4 GHz WLAN habt benötigst du die Logindaten für letzteres.

## Inhaltsverzeichnis

Ablauf.....	2
Tag 1.....	2
Tag 2.....	2
Tag 3.....	2
Aufgaben.....	3
Kennenlernen.....	3
Calliope-Programmierung.....	3
Bonus.....	3
Kennenlernen der Arduino-IDE.....	4
Übungen zum Programmieren des ESP8266.....	6
Aufgabe 1: LED blinken lassen.....	6
Aufgabe 2: Feuchtesensor auslesen.....	6
Aufgabe 3: Schwellwertvergleich.....	7
Bonus.....	7
Pflanzensensor.....	7
Zusammenbau.....	8
Bespielen des ESP8266 mit dem Pflanzensensorprogramm.....	8
Kennenlernen des Webinterfaces.....	10
Kalibrieren der Sensoren.....	10
Einstellen der Grenzen für die LED Ampel.....	10
Bonus.....	11
Verwendete Programme und Links.....	12
Calliope.....	12
Pflanzensensor.....	12
Fabmobil und Silicon Saxony.....	12

# Ablauf

## Tag 1

- Anmeldung, Kennenlernen, Bustour
- Calliope Programmierung

## Tag 2

- Pflanzensensor mit Calliope bauen, kalibrieren, testen
- Arduino IDE kennenlernen

## Tag 3

- Einführung C-Programmierung in Arduino-IDE
- Pflanzensensor zusammenbauen
- Pflanzensensor konfigurieren und flashen
- Sensoren über Webinterface kalibrieren
- Testen
- Abschluss

# Aufgaben

## Kennenlernen

- macht euch ein Namensschild mit Kreppband: wenn ihr nicht fotografiert werden dürft nehmt ihr einen roten Stift, ansonsten nehmt ihr einen schwarzen Stift
- lasst euch den Bus zeigen und erklären
- meldet euch über das [Anmeldeformular](#) an

## Calliope-Programmierung

Eure WorkshopleiterInnen werden euch eine Einführung ins Thema geben. Wenn ihr die bekommen habt findet euch in Zweiergruppen zusammen. Pro Gruppe benötigt ihr:

- Einen Laptop inkl. Maus und Netzteil
- Einen Calliope inklusive MicroUSB-Kabel
- Ein Calliope Lernkartenbuch

Eure Aufgabe ist es, dass Calliope Lernkartenbuch durchzuarbeiten. Es erklärt euch alles, was ihr über den Calliope und dessen Programmierung wissen müsst. Am Ende gibt es ein paar kleine Programmierherausforderungen. Wenn ihr die geschafft habt, seid ihr gut vorbereitet für die Programmierung des Pflanzensensors.


Meldet euch dann bei euren WorkshopleiterInnen. Die geben euch einen Bodenfeuchtesensor und die notwendigen Kabel, um sie mit dem Calliope zu verbinden. Außerdem wird euch gezeigt, wie ein funktionierendes Pflanzensensorprogramm auf dem Calliope aussehen kann. Ihr sollt jetzt auch ein solches Programm entwerfen. Nutzt dafür das Wissen, was ihr durch die Lernkarten erworben habt. Wenn ihr nicht weiter kommt, fragt die WorkshopleiterInnen um Hilfe.

Wenn euer Calliope Pflanzensensor funktioniert, könnt ihr ihn mit feuchter und trockener Blumenerde kalibrieren und testen.

## Bonus

Falls ihr früher als die anderen fertig seid, gibt es folgende Bonusaufgabe:

Sucht euch eine andere Gruppe die auch schon fertig ist. Ihr sollt ihr ein Schnick-Schnack-Schnuck Spiel programmieren, was mit zwei Calliopes gegeneinander gespielt werden kann. Hier ein paar Tipps:

- Zum Senden und Empfangen der Daten zwischen den Calliopes benötigt ihr das Funk-Modul. Ihr findet es hinter dem  Erweiterungen Button im Online-Editor
- als erstes wählt ihr auf eurem Calliope (z.B.: A = Schere, B = Stein, A + B = Papier) und speichert diese Wahl in einer Variable
- dann schickt diese Wahl an den anderen Calliope
- parallel empfängt ihr die Wahl des anderen Calliope und speichert sie in einer anderen Variable
- diese zwei Variablen müssen dann miteinander verglichen werden: Stein schlägt Schere, Schere schlägt Papier usw.
- danach müsst ihr nur noch das Ergebnis ausgeben: habt ihr gewonnen oder verloren?

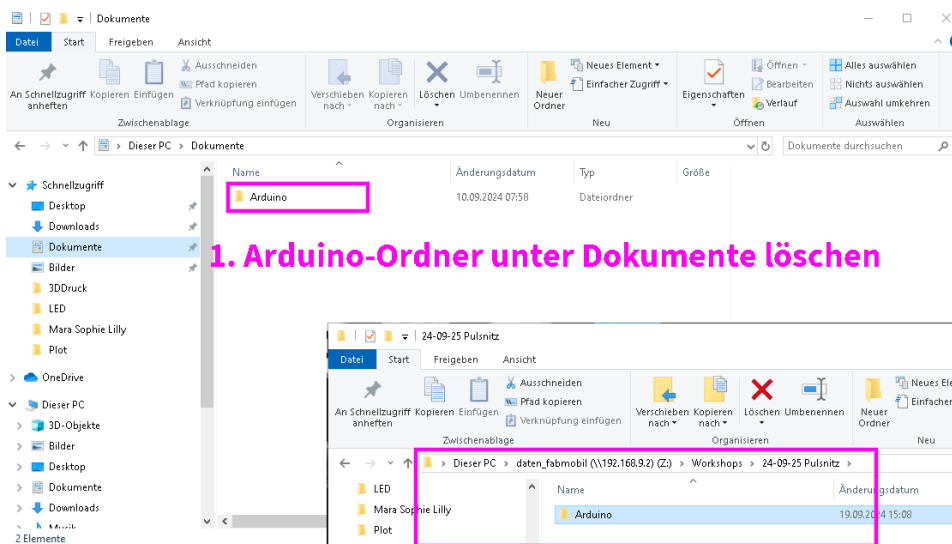
## Kennenlernen der Arduino-IDE

Nachdem ihr mit dem Calliope den Einstieg ins Programmieren gefunden habt, geht es mit der Arduino-IDE und der Programmiersprache C++ weiter.

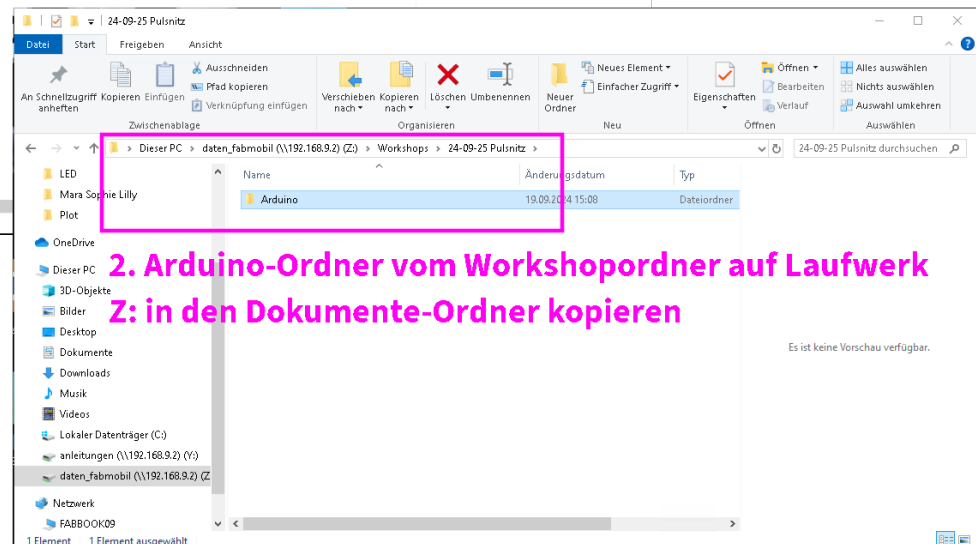
Ihr arbeitet weiterhin in Zweiergruppen und benötigt für diesen Teil:

- euren Laptop
- einen ESP8266-Microcontroller
- ein MicroUSB-Kabel um den Microcontroller mit eurem PC zu verbinden
- die Software „Arduino IDE“ (sie ist schon auf eurem Laptop installiert)
- den Bodenfeuchtesensor, den ihr schon mit eurem Calliope verwendet habt
- Den Spickzettel zur C-Programmierung (ein PDF im gleichen Verzeichnis wie diese Datei)

Als erstes kopiert ihr euch den Arduino-Ordner aus dem Workshopordner auf dem Netzlaufwerk in den Dokumente Ordner auf eurem Laptop. Dort sollte es schon einen Arduino Ordner geben, löscht ihn vorher damit ihr die aktuellen Versionen des Quellcodes habt:



1. Arduino-Ordner unter Dokumente löschen



2. Arduino-Ordner vom Workshopordner auf Laufwerk Z: in den Dokumente-Ordner kopieren

Dann öffnet ihr die Arduino IDE und schaut euch um. Hier ist ein Bild in dem die wichtigsten Bereiche markiert sind:



# Übungen zum Programmieren des ESP8266

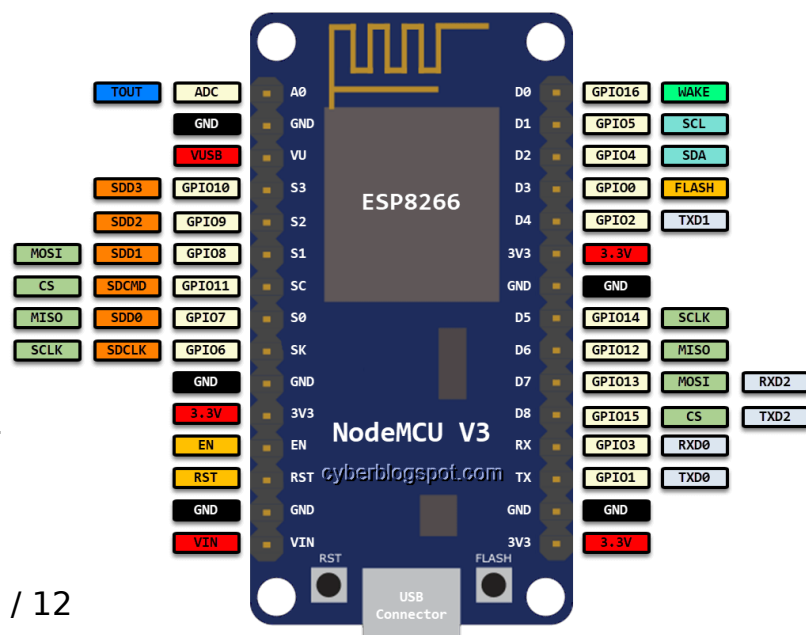
Lest euch den Spickzettel durch. Auch wenn die dahinter liegenden Konzepte die Gleichen sind, ist das Programmieren in C ganz anders als die grafische Programmierung des Calliopes. Lasst euch nicht abschrecken: es ist noch kein\*e Programmierer\*In vom Himmel gefallen. Versucht euch durchzuhangeln und helft euch gegenseitig bzw. fragt die Workshopleiter\*Innen wenn ihr nicht weiter kommt!

## Aufgabe 1: LED blinken lassen

- öffne `LED.ino` im LED Verzeichnis in der Arduino IDE
- verbinde den ESP8266 per USB Kabel mit deinem Laptop
- überspiele das Programm auf deinen ESP8266:
  - klicke unter Werkzeuge auf Board, wähle dann ESP8266 Boards und danach `LOLIN(WEMOS) D1 R2 & mini`
  - klicke 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 es 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 benötigst du die folgenden zwei zusätzlichen Befehle:
  - `digitalWrite(LED, HIGH);` schaltet die LED aus
  - `delay(3000);` pausiert das Programm für 3000ms, also 3 Sekunden

## Aufgabe 2: Feuchtesensor auslesen

- Schließe den Bodenfeuchtesensor mit Hilfe von Jumperkabeln an deinen ESP8266 an: rot kommt an 3V3 (Versorgungsspannung), schwarz an GND (Masse) und Gelb an A0 (Sensorwert). Die Pins sind beschriftet bzw. kannst du sie rechts ablesen.
- öffne `feuchtesensor.ino` in der Arduino IDE



- überspiele das Programm auf deinen ESP8266
- schau dir die Messwerte im seriellen Monitor und im seriellen Plotter an.
- Nimm den Sensor in die Hand und beobachte, wie sich der Wert verändert
- Der Wert liegt immer zwischen 0 und 1024
- deine Aufgabe ist es nun, die interne LED entsprechend dem Wert des Analogensors 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 3: Schwellwertvergleich

- 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 hier nachlesen: <https://docs.arduino.cc/language-reference/de/struktur/control-structure/else/>

### Bonus

Wenn du früher als alle anderen fertig bist, kannst du dir ein Arduinobuch aus der Fabmobil Bibliothek nehmen und dir dort ein kleines Arduino-Programmierprojekt herausuchen. Sprich mit deinen WorkshopleiterInnen: fast alle der in den Büchern verwendeten Sensoren haben wir auch im Fabmobil und du kannst sie gern ausprobieren.

Die Arduino IDE hat auch viele Beispielprogramme die ausprobiert und verstanden werden wollen. Klick doch mal auf `Datei` und dann auf `Beispiele` und schau, was du da so findest!

### Pflanzensensor

Endlich ist es so weit und wir kümmern uns um euren ganz persönlichen Pflanzensensor! Du benötigst:

- einen ESP8266 Mikrocontroller
- einen Bodenfeuchtesensor
- ein Display
- ein Pflanzensensor PCB

- einen Laptop
- ein MicroUSB-Kabel
- Abstandshalter und Schrauben aus Plastik
- 3D-gedruckte Hülle für den Bodenfeuchtsensor

## Zusammenbau

Baue als erstes deinen Pflanzensensor zusammen:

- der ESP8266 kommt in die Fassung auf dem PCB. **Achtung:** der USB-Port muss nach unten zeigen!
- Das Display kommt in die Fassung über dem ESP8266. Fixiert es mit Plastikabstandshaltern und Schrauben: es gibt entsprechende Löcher auf dem PCB und im Display
- steckt den Bodenfeuchtesensor an: er kommt an den Port der mit X6 gelabelt ist. Passt auf die Orientierung des Kabels auf: – am PCB muss an GND am Sensor
- Setzt das Gehäuse um den oberen Teil des Bodenfeuchtesensors. Es schützt die elektronischen Bauteile davor, dass sie beim Blumengießen Spritzwasser abbekommen und dadurch eventuell kaputt gehen.
  - Falls das Gehäuse nicht hält könnt ihr es mit Klebeband oder Sekundenkleber zusammenkleben.

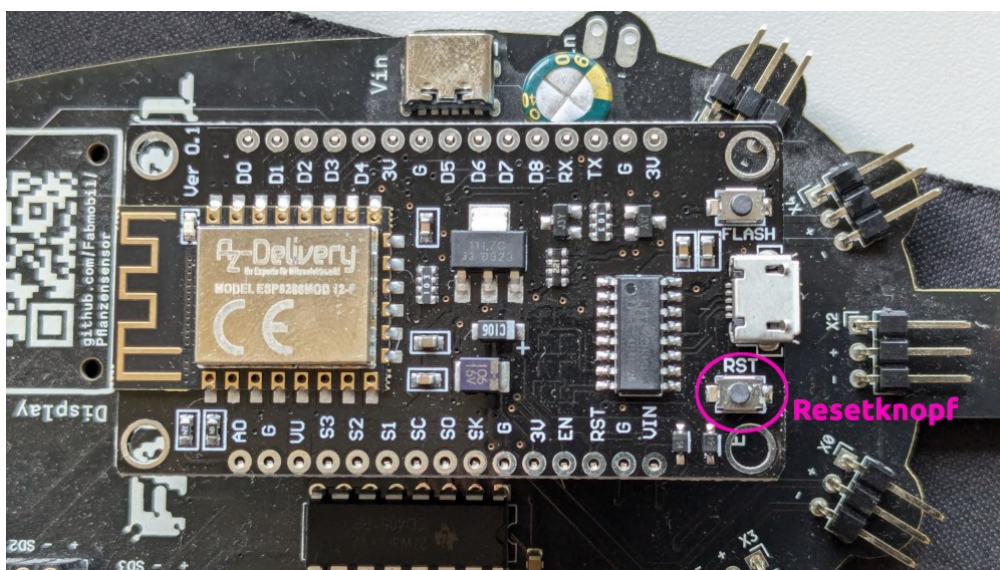
## Bespielen des ESP8266 mit dem Pflanzensensorprogramm

Folgende Schritte sind zu tun:

- öffnet die `Pflanzensensor.ino` – Datei in eurer Arduino IDE. Sie liegt im `Dokumente/Arduino/Pflanzensensor`-Ordner
- der Quellcode des Pflanzensensorprogramms besteht aus vielen Dateien. Sie werden alle automatisch geöffnet und ihr seht sie in der Leiste über dem Editor in der Arduino IDE. Schaut euch ruhig um und in die Dateien rein, falls es euch interessiert. Es ist allerdings wichtig, dass ihr nichts verändert, denn dann ist die Chance groß, dass ihr Fehler einbaut und das Programm sich später nicht mehr überspielen lässt
- Verändern müsst ihr nur die `passwoerter.h`- Datei. Falls ihr diese in der Leiste mit den Dateinamen nicht findet könnt ihr rechts auf die drei ... klicken, dann bekommt ihr eine Liste mit allen geöffneten Dateien angezeigt und könnt sie dort auswählen
- Öffnet die `passwoerter.h`-Datei und tragt dort die Daten des WLANs ein, in dem ihr den Pflanzensensor betreiben wollt:



- Ihr könnt bis zu drei verschiedene WLANs konfigurieren. Die Variablen `wifiSsid1`, `wifiSsid2` und `wifiSsid3` speichern den Namen des WLANs und die Variablen `wifiPassword1`, `wifiPassword2` und `wifiPassword3` speichern das Passwort der WLANs. Der eigentliche Name des WLANs bzw. das Passwort kommt zwischen die Anführungszeichen.
- Passt auf, dass ihr die schon ausgefüllten Fabmobil WLAN Daten nicht löscht oder verändert, sonst kann sich euer Sensor nicht mit dem WLAN im Fabmobil verbinden und ihr könnt euch nicht mit eurem Laptop zum Webinterface verbinden
- Falls ihr plant, den Sensor mobil zu benutzen, ist es eventuell sinnvoll, auch die Daten eures Handy WLAN Hotspots einzutragen
- Danach könnt ihr den Quellcode kompilieren und überspielen so wie ihr das bei den Programmierübungen auch schon gemacht habt.
- Abschließend müsst ihr noch ein paar Extradateien überspielen. Das funktioniert, in dem ihr `[Strg] + [Shift] + [P]` gleichzeitig drückt, `upload` eingibt und `Upload LittleFS to Pico/ESP8266/ESP32` auswählt.
- Öffnet jetzt den seriellen Monitor und startet den ESP neu. Achtet auf die Zeile `Start von Wifi-Modul...`: jetzt versucht sich euer Pflanzensensor mit dem WLAN zu verbinden. Wenn das geklappt hat wird euch 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`. Merkt oder kopiert sie euch.



- Gebt diese IP-Adresse jetzt im Browser eures Laptops als Adresse ein. Ihr solltet jetzt auf dem Webinterface eures Pflanzensensors landen.

## Kennenlernen des Webinterfaces

- Schaut euch das Webinterface eures Pflanzensensors im Browser an, versteht was es darstellt, findet die Adminseite und lest [hier](#) nach, was da alles eingestellt werden kann.

## Kalibrieren der Sensoren

Damit euer Bodenfeuchtesensor richtig funktioniert, müsst ihr ihn jetzt kalibrieren:

- Stecke den Bodenfeuchtesensor in sehr feuchte Erde.  
**Achtung:** Du solltest den Sensor immer gleich tief in die Erde stecken damit die Messung einigermaßen genau ist. Ein Sensor, der nur halb in der Erde steckt, wird dir 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 ihr den Sensor in die Erde stecken solltet.
- Geh auf die Adminseite deines Pflanzensensors, finde dort den Bodenfeuchte-Bereich und lese den aktueller absoluter Messwert ab. Dieser muss nun darunter bei Maximalwert eingetragen werden.
- Stecke den Bodenfeuchtesensor in sehr trockene Erde.
- Lies wieder den aktuellen absoluten Messwert ab und trage ihn diesmal bei Minimalwert ein.
- Speichere deine Änderungen.
- Jetzt ist dein Bodenfeuchtesensor kalibriert. Falls du allerdings später zu Hause andere Erde hast kann es sein, dass du die Kalibrierung dort noch einmal wiederholen musst.
- Wenn du willst, kannst du auch deinen Helligkeitssensor kalibrieren: bedecke ihn damit es ganz dunkel ist und halte ihn in die Mittagssonne. Die voreingestellten Werte passen aber ganz gut, so dass du das nicht zwingend machen musst.

## Einstellen der Grenzen für die LED Ampel

Die Grenzwerte der LED Ampel für die unterschiedlichen Sensoren können auf der Adminseite deines Pflanzensensors eingestellt werden:

**Luftfeuchte**

Alarm aktiv? ☐

unterer gelber Schwellwert:  
20

unterer grüner Schwellwert:  
40

oberer grüner Schwellwert:  
60

oberer gelber Schwellwert:  
80

Diese Schwellwerte sind folgendermaßen definiert:



Abhängig davon, was die Pflanze die du mit dem Sensor überwachen willst für Bedürfnisse hat, stellst du jetzt die Skale für die unterschiedlichen Sensoren ein.

## Bonus

Wenn ihr noch genügend Zeit und Motivation habt könnt ihr eurem Pflanzensensor noch beibringen, euch per Telegram oder Email zu benachrichtigen, falls eure Pflanze eure Aufmerksamkeit benötigt. Dazu müsst ihr:

- in der Arduino IDE die Datei `einstellungen.h` öffnen und in der Zeile `#define MODUL_WEBHOOK false` das `false` zu einem `true` ändern. Das aktiviert das Webhook-Modul, welches für die Benachrichtigung notwendig ist.
- Öffnet dann diese [Anleitungsseite](#) in eurem Browser und arbeitet sie durch.

# Verwendete Programme und Links







## Calliope

- Calliope Lernkarten aus dem Fabmobil
- [Calliope Online Editor](#)
- [Einführungs Tutorial auf Youtube](#)

## Pflanzensensor

- [Arduino IDE](#)
- [Pflanzensensor Quellcode und Erklärungen](#)
- Anleitungen im Pflanzensensor Wiki:
  - [Pflanzensensor konfigurieren und bespielen](#)
  - [Erklärungen zum Webinterface](#)
  - [Webhook für Telegram- oder Mailbenachrichtigungen einrichten](#)
  - [make.com Website für Mail- oder Telegrambenachrichtigungen](#)

## Fabmobil und Silicon Saxony

-  [Fabmobil Homepage](#) 
-  [Silicon Saxony](#) 
-  [Bosch](#) 
-  [Global Foundries](#) 
-  [Infineon](#) 
-  [Xfab](#) 