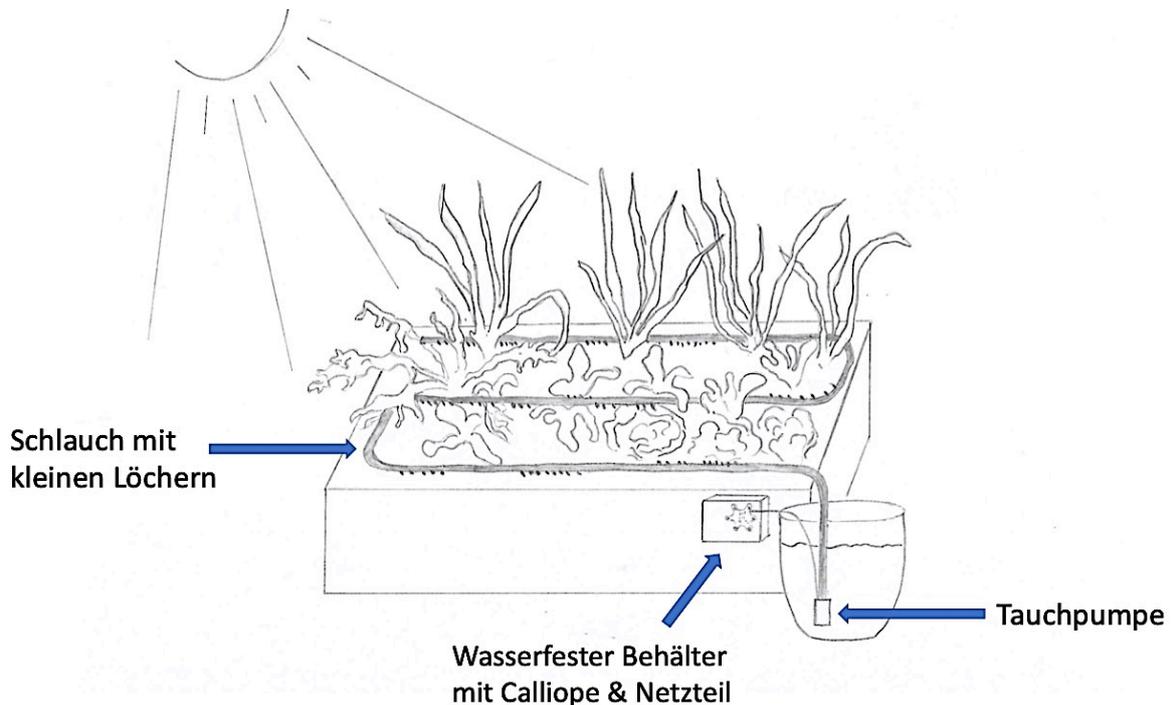


# Smartplants - Bewässerung mit Calliope



## Kurzbeschreibung

Bei dem Projekt soll eine Gruppe von zwei bis fünf Schüler\*innen (SuS) eine automatische Bewässerung für ein Hochbeet entwickeln. Mit einer kleinen Modifizierung könnte es auch für Topfpflanzen genutzt werden. Die Steuerung der Bewässerung soll über einen Calliope, welches ein günstiger Einplatinencomputer ist, erfolgen. Mit der kostenfreien Plattform „<https://makecode.calliope.cc/>“ lässt sich der Calliope auch für Einsteiger\*innen unkompliziert mit Hilfe von Textbausteinen programmieren. Für das Projekt sollten trotzdem schon Vorerfahrungen mit dem Calliope oder dem Programmieren vorhanden sein, da es etwas umfangreicher ist.

Ziel des Projektes ist es, dass die SuS mit hoher Eigenleistung eine automatische Bewässerungsanlage, welche täglich aber nur bei Trockenheit den Boden gießt, herstellen. Dafür wird eine Bodenfeuchtemessung entwickelt und eine Tauchpumpe mit Netzteil und einem Schalter, der sich über den Calliope schalten lässt, verkabelt und programmiert.

Als Unterrichtsmaterial gibt es mehrere Skizzen zur Unterstützung bei der Verkabelung sowie aufeinander aufbauende Programmier-Aufgaben, welche die Gruppen schrittweise zum fertigen Programm führen sollen. (siehe Anhang)

## Differenzierungsmöglichkeiten

Differenzierung ist bei diesem Projekt durch offene Lösungswege und individuelle Problemlösemethoden der SuS möglich. Es führen viele richtige Wege zum Ziel bzw. sind

viele Wege zur Entstehung des Projekts zulässig, indem bspw. aufgrund verschiedener Programmiermöglichkeiten vielfältige Gestaltungsspielräume gegeben werden.

Die Aufgabenblätter (siehe Anhang) enthalten für jede Programmier-Aufgabe gezielte Hinweise, welche das Programm verbal erklären und auch jeweils eine komplette Lösung der Aufgabe enthalten. Diese können gezielt für SuS mit weniger Erfahrung bzw. Fähigkeiten im Programmieren oder bei generell aufkommenden Schwierigkeiten ausgeteilt und verwendet werden.

SuS mit höherem Förderbedarf könnten bei dem Projekt die Aufgaben, welche weniger intellektuelle Fähigkeiten erfordern, bewältigen und damit trotzdem am Erfolg des Gesamtproduktes teilhaben. Sie könnten den für die Bewässerung benötigten Schlauch mit Löchern präparieren, diesen in das Hochbeet legen und z.B. mit Zeltheringen befestigen, das Gehäuse für den Calliope und das Netzteil vorbereiten, dieses Gehäuse am Hochbeet anbringen und die Regentonnen mit Regenwasser am Hochbeet aufstellen.

## Voraussetzung für die Unterrichtseinheit

Die Gruppe sollte vertraut mit dem Calliope und seiner Programmierung sein. Falls diese Vorkenntnisse nicht vorhanden sind, würden sich als Einführung die „Calliope Mini Lernkarten“ vom Digital Literacy Lab eignen. Die Lernkarten dienen als kleinschrittige Hinführung zur Programmierung des Calliope. Sie lassen sich über folgenden Link herunterladen: [http://tueftelakademie.de/wp-content/uploads/2020/02/DLL\\_Sense-Your-School\\_UE3\\_Lernkarten\\_calliope-mini.pdf](http://tueftelakademie.de/wp-content/uploads/2020/02/DLL_Sense-Your-School_UE3_Lernkarten_calliope-mini.pdf)

## Unterrichtsmaterialien

- Calliope Education Set (<https://www.conrad.de/2145193.html>)
- Tauchpumpe (<https://www.conrad.de/1435461.html>)
- Netzteil für Tauchpumpe (<https://www.conrad.de/510822.html>)
- Schlauch für Tauchpumpe (<https://www.conrad.de/538868.html>)
- Elektronischer Schalterplatine für Calliope (<https://www.amazon.de/dp/B07HBQ2SZR>)
- Beet mit Regenfass und Strom (damit Calliope und Tauchpumpe Strom und Wasser haben)
- Regenfestes Gehäuse für Calliope und Netzteil

# Ablauf der Unterrichtsstunden

---

## 1. Stunde: Einführung mit Brainstorming

An der Schule gibt es Hochbeete für die eine automatische Bewässerung konstruiert werden soll. Als Einführung sollen die SuS mit der DAB-Methode über folgende Fragen nachdenken:

- Wofür benötigt man eine automatische Bewässerung?
- Wann sollte die Bewässerung angeschaltet werden?
- Wie könnte es bei unserem Hochbeet umgesetzt werden?

Falls die SuS keine Ideen haben und um die Fragen gezielter beantworten zu können, könnte die Lehrkraft die Tauchpumpe zeigen und diese erklären.

Die SuS tauschen sich untereinander aus und schreiben ihre Ergebnisse auf Kärtchen, die an der Tafel gesammelt und besprochen werden.

### Lösungsvorschlag

Die SuS sollen zu dem Ergebnis kommen, dass das Beet einmal am Tag gegossen werden sollte und dieses nur erforderlich ist, wenn der Boden trocken ist.

Dafür wird eine Regentonne benötigt, aus der die Tauchpumpe über den Calliope gesteuert Wasser pumpen kann. Der Schlauch bekommt kleine Löcher, um eine Tröpfchenbewässerung möglich zu machen und wird schlangenförmig in das Hochbeet gelegt. Außerdem soll der Calliope die Bodenfeuchtigkeit messen können (Feuchtigkeitsmessung wird in Stunde 6 behandelt).

## 2. Stunde: Bewässerung mit Tauchpumpe testen

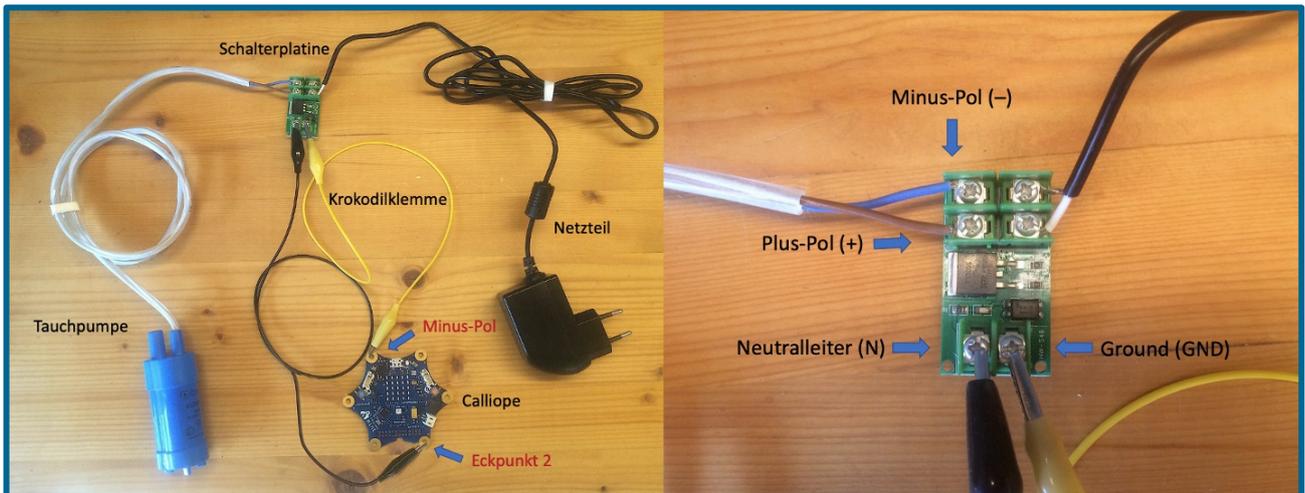
In dieser Stunde sollen die Gruppen den Schlauch der Tauchpumpe testweise in das Hochbeet legen und anschließend entsprechende Löcher an den richtigen Stellen markieren. Nun sollen viele sehr kleine Löcher (kleinster Bohrer, ca 2mm) in den Schlauch gebohrt werden. Danach kann getestet werden, wie lange die Pumpe angeschaltet werden muss, um das Hochbeet vollständig zu wässern. Diese Zeit wird später bei der Programmierung benötigt.



Quelle: [rekubik.de/magazin/troepfchenbewaesserung/](http://rekubik.de/magazin/troepfchenbewaesserung/)



### 3. Stunde: Tauchpumpe mit Calliope und Schalter verkabeln



In dieser Stunde wird die Verkabelung von der Tauchpumpe mit dem Netzteil und dem Calliope an der Schalterplatine durchgeführt. Die SuS lernen die jeweiligen Bauteile kennen und müssen eventuell Kabel von der Tauchpumpe und dem Netzteil entmanteln. Die Schalterplatine schließt den Stromkreis zwischen Netzteil und Tauchpumpe, wenn zwischen N und GND Strom fließt. Um dieses mit dem Calliope zu steuern, muss alles wie in dem Bild (s.o.) verkabelt werden (Im Anhang befindet sich ebenfalls eine Skizze als Kopiervorlage). Zum Ansteuern wird die Ecke 2 (P2) vom Calliope genutzt. Wenn nun der analoge Wert 1023 auf diesem Ausgang ausgegeben wird, ist die Pumpe an; bei dem Wert 0 ist die Pumpe aus.

1023 ist der höchste analoge Wert der vom Calliope ausgegeben werden kann. Bei kleineren Werten würde die Pumpe mit geringerer Leistung laufen, die Schalterplatine funktioniert als Dimmer.

**Hinweis:** Falls sich die Pumpe mit der Schalterplatine nicht schalten lässt, versucht die Anschlüsse des Netzteils und der Tauchpumpe zu tauschen, indem das Netzteil an der linken Seite der Schalterplatine angeschlossen wird und die Tauchpumpe an der rechten Seite.

### 4. Stunde: Tauchpumpe über den Calliope ein- und ausschalten

In dieser Stunde soll zunächst ein Programm geschrieben werden um die Tauchpumpe über den Calliope per Knopfdruck ein- und auszuschalten.

Im Anhang befindet sich dafür die Programmier-Aufgabe 1. Alle Programmier-Aufgaben enthalten einen ausführlichen Hinweis, welcher von den Schüler\*innen nur bei Bedarf genutzt werden sollte. Dafür könnten die Arbeitsblätter in der Mitte geknickt oder zerschnitten werden. Außerdem gibt es jeweils einen Lösungsvorschlag, welcher ebenfalls nur bei Bedarf oder zur Kontrolle verteilt werden sollte. Auf den Arbeitsblättern wird zum Programmieren die Plattform „<https://makecode.calliope.cc/>“ genutzt.

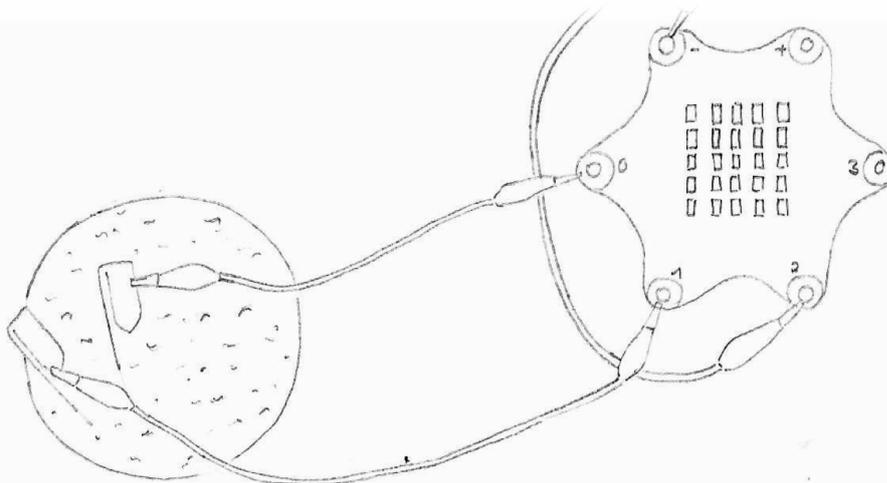
Wenn sich die Pumpe nun über den Calliope steuern lässt, soll anschließend in der zweiten Programmier-Aufgabe das Programm erweitert werden, sodass die Pumpe jedes Mal für die beim Testlauf gemessene Zeit angeschaltet wird.

## 5. Stunde: Tauchpumpe täglich mit Calliope einschalten

Bisher lässt sich die Tauchpumpe manuell über einen Knopfdruck am Calliope einschalten. Nun soll das Programm erweitert werden, sodass der Calliope die Pumpe automatisch täglich einschaltet. Da der Calliope keine Uhr eingebaut hat, muss dafür die Zeit im Programm gemessen werden.

Dieser Schritt ist etwas umfangreicher, deshalb ist die Aufgabe in zwei Teile aufgeteilt: Zunächst wird bei der Programmier-Aufgabe 3 (siehe Anhang) die Zeitmessung programmiert. Anschließend wird in der Programmier-Aufgabe 4 das Programm erweitert, sodass der Calliope die Pumpe alle 24h einschaltet.

## 6. Stunde: Bodenfeuchtemessung



In dieser Stunde wird das Thema „Messung der Bodenfeuchtigkeit“ behandelt. Mit der DAB-Methode sollen die SuS einleitend überlegen, wie der Calliope die Bodenfeuchtigkeit messen könnte. Als Hinweis könnte man ihnen sagen, dass der Calliope Strom oder die Leitfähigkeit an seinen Ausgängen messen kann.

Wie in der Skizze zu sehen ist, nutzen wir zwei Büroklammern, die nebeneinander in die Erde gesteckt und mit dem Calliope verbunden werden. Im Programm setzen wir eine der Büroklammern unter Strom und messen an der anderen wie viel Strom ankommt. Je feuchter der Boden ist, umso mehr Strom leitet er. Das Programm wird in der nächsten Stunde geschrieben.

## 7. Stunde: Testreihe zur Bodenfeuchtemessung

Diese Stunde dient als Test zur Messung der Bodenfeuchte und der Interpretation der vom Calliope gemessenen Werte. Dafür ist es sinnvoll ein eigenes kleines Programm zu

schreiben, das auf dem Display vom Calliope die gemessenen Werte anzeigt. Für diesen Schritt befindet sich im Anhang die Programmier-Aufgabe 5.

Nachdem das Programm geschrieben wurde, soll an einem Blumentopf oder auch im Hochbeet ermittelt werden, welche Werte der Calliope anzeigt, wenn die Erde trocken oder nass ist. Diese Werte müssen notiert werden.

## 8. Stunde: Pumpe alle 24h bei Trockenheit einschalten

In dieser Stunde soll das Programm erweitert werden, sodass der Calliope die Pumpe nur bei Trockenheit einschaltet. Für die Aufgabe werden die Werte aus der vorherigen Stunde zur Interpretation der Bodenfeuchte benötigt. Im Anhang befindet sich für diese Stunde die Programmier-Aufgabe 6. Dort wird die Pumpe nur bei Werten unter 900 eingeschaltet.

## 9. Stunde: Calliope und Tauchpumpe am Hochbeet befestigen

In dieser Stunde soll das Projekt fertiggestellt und am Hochbeet angebracht werden. Dafür wird ein regenfestes Gehäuse (z.B. eine Brotdose) benötigt, in welchem das Netzteil und der Calliope verstaut werden können. Neben das Gehäuse muss die Regentonne mit der Tauchpumpe platziert werden. Anschließend kann der Schlauch in das Hochbeet gelegt werden. Zum Befestigen des Schlauches könnten bspw. Zeltheringe genutzt werden. Wenn alles angeschlossen ist, kann der Calliope eingeschaltet und die Bewässerung in Betrieb genommen werden. Das erste Gießen kann bei unserem Programm über den Knopf A erfolgen. Es ist sinnvoll die Bewässerung die nächsten Tage zu überwachen und auf Fehler zu überprüfen.

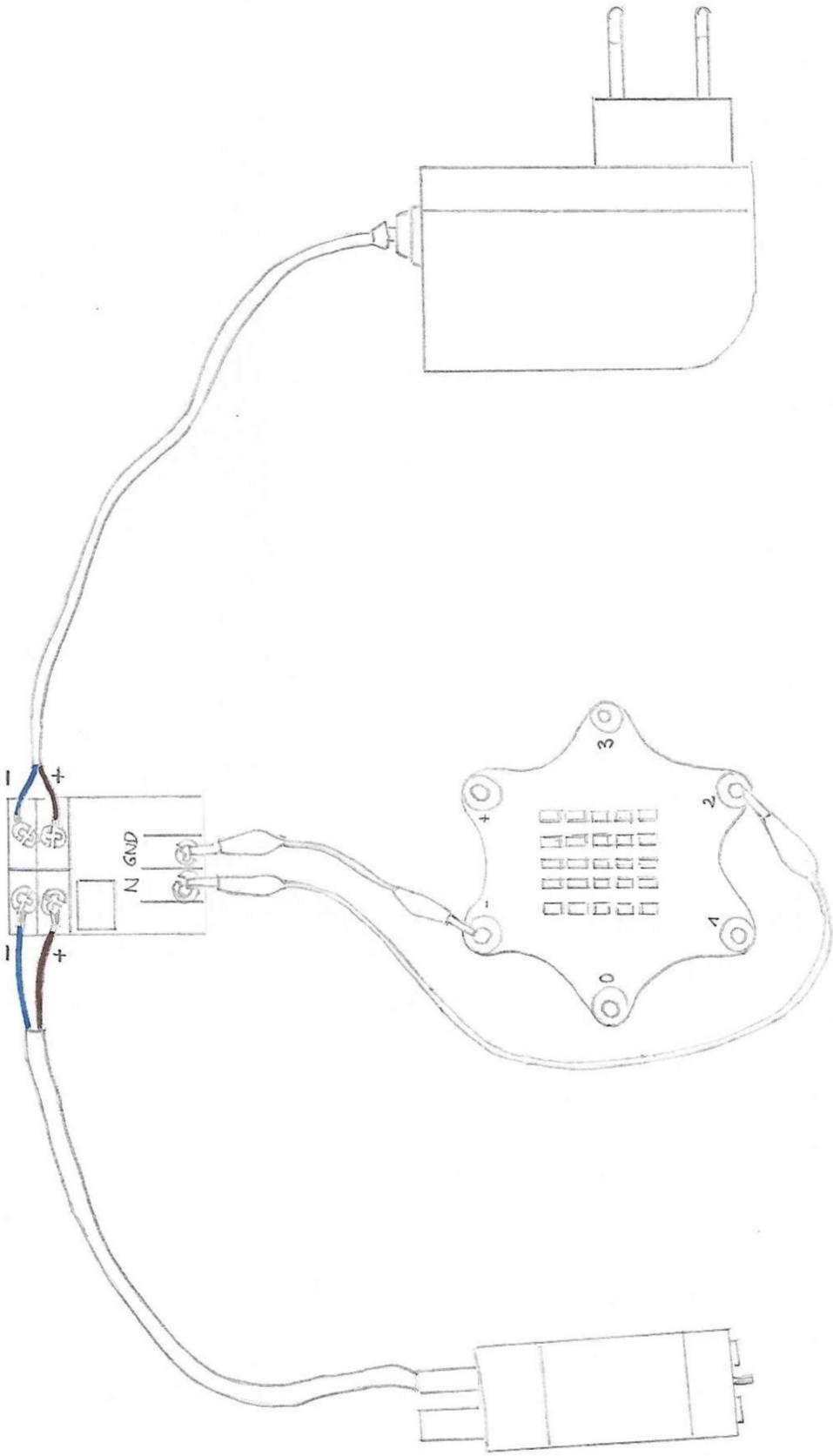
## 10. Stunde: Auswertung

In dieser Stunde liegt der Fokus hauptsächlich auf die Gesamtauswertung des Projektes. Hier werden Fragen dazu thematisiert, wie gut das Projekt funktioniert hat bzw. sich umsetzen ließ. Es werden Anregungen gesammelt, ob das Projekt aus Sicht der SuS praktikabel ist bzw. ob es Ideen zu Verbesserungen gibt. Dabei werden die SuS dazu animiert, ihr Projekt auch aus kritischem Blickwinkel zu betrachten und darüber in den Austausch zu kommen. Es wird folglich darüber nachdacht, welche Schwierigkeiten ggf. in Zukunft mit der in Betrieb genommenen Bewässerung entstehen und wie diese eventuell gelöst werden können. Wird die Bewässerung auf lange Sicht funktionieren?

Für die Programmier-Aufgabe 1

Verkabelung der Schalterplatine – Pumpe – Netzteil – Calliope

---



## Programmier-Aufgabe 1

### Pumpe mit Knopfdruck am Calliope An- und Ausschalten

---

Die Schalterplatine (siehe Skizze) schließt den Stromkreis zwischen Netzteil und Tauchpumpe, wenn zwischen N und GND Strom fließt. Um dieses mit dem Calliope zu steuern, muss alles wie in der Skizze verkabelt werden. Zum Steuern nutzen wird die Ecke 2 (P2). Wenn nun der Analoge Wert 1023 auf diesem Ausgang ausgegeben wird, ist die Pumpe an; bei dem Wert 0 ist die Pumpe aus.

1023 ist der höchste Analoge Wert der vom Calliope ausgegeben werden kann. Bei kleineren Werten würde die Pumpe mit geringerer Leistung laufen.

Schreibt nun zwei Funktionen die per Knopfdruck am Calliope die Pumpe anschaltet und die Pumpe ausschaltet.

---

## Programmier-Aufgabe 1

### Hinweis

---

Erstellt eine Variable für den *Pumpenstatus*, welcher 0 oder 1023 betragen soll.

Erstellt die Funktion „pumpeAn“, welche die Pumpe anschaltet, dafür den *Pumpenstatus* auf 1023 setzt und diesen als analogen Wert über P2 ausgibt. Dadurch wird die Schalterplatine über die Ecke 2 angesteuert. Erstellt auf gleiche Weise die Funktion „pumpeAus“ zum Ausschalten mit dem Wert 0.

Wenn ein Knopf am Calliope gedrückt wird, soll die Pumpe nun eine der beiden Funktion aufrufen. Um zu gucken, ob die Pumpe dabei an oder ausgeschaltet werden muss, könnt ihr den *Pumpenstatus* vergleichen. Dafür benötigt ihr die „wenn-dann-ansonsten“ Logik.

## Programmier-Aufgabe 1

### Lösungsvorschlag

---

```
whenKnopfA gedrueckt
  wenn (pumpenstatus == 0)
    dann (funktion aufrufen pumpeAn)
    ansonsten (funktion aufrufen pumpeAus)

funktion pumpeAn
  setze pumpenstatus auf 1023
  schreibe analogen Wert von Pin P2 auf pumpenstatus

funktion pumpeAus
  setze pumpenstatus auf 0
  schreibe analogen Wert von Pin P2 auf pumpenstatus
```

The image shows a Scratch-style code editor with a grid background. It contains three main code blocks:

- Event block:** "wenn Knopf A gedrückt" (when button A is pressed).
- Conditional block:** "wenn (pumpenstatus == 0)" (if pump status is 0).
- Then block:** "dann (funktion aufrufen pumpeAn)" (then call function pumpeAn).
- Else block:** "ansonsten (funktion aufrufen pumpeAus)" (otherwise call function pumpeAus).
- Function block "pumpeAn":** "setze pumpenstatus auf 1023" (set pump status to 1023) and "schreibe analogen Wert von Pin P2 auf pumpenstatus" (write analog value of pin P2 to pump status).
- Function block "pumpeAus":** "setze pumpenstatus auf 0" (set pump status to 0) and "schreibe analogen Wert von Pin P2 auf pumpenstatus" (write analog value of pin P2 to pump status).

## Programmier-Aufgabe 2

### Pumpe für die beim Testlauf gemessene Zeit einschalten

---

Verändert das Programm aus Aufgabe 1, sodass es die Pumpe per Knopfdruck für die beim Pumpentest gemessene Zeit (um das Hochbeet vollständig zu wässern) anschaltet.

Dafür könnt ihr die „dauerhaft“-Funktion, welche vom Calliope automatisch dauerhaft unendlich oft aufgerufen wird, abändern, sodass sie nur einmal jede Sekunde ausgeführt wird. In der Funktion wird dann die Pumpe an- oder ausgeschaltet.

Erstellt eine Variable für die Pumpzeit, welche per Knopfdruck auf die gemessene Zeit gesetzt wird.

## Programmier-Aufgabe 2

### Hinweis

---

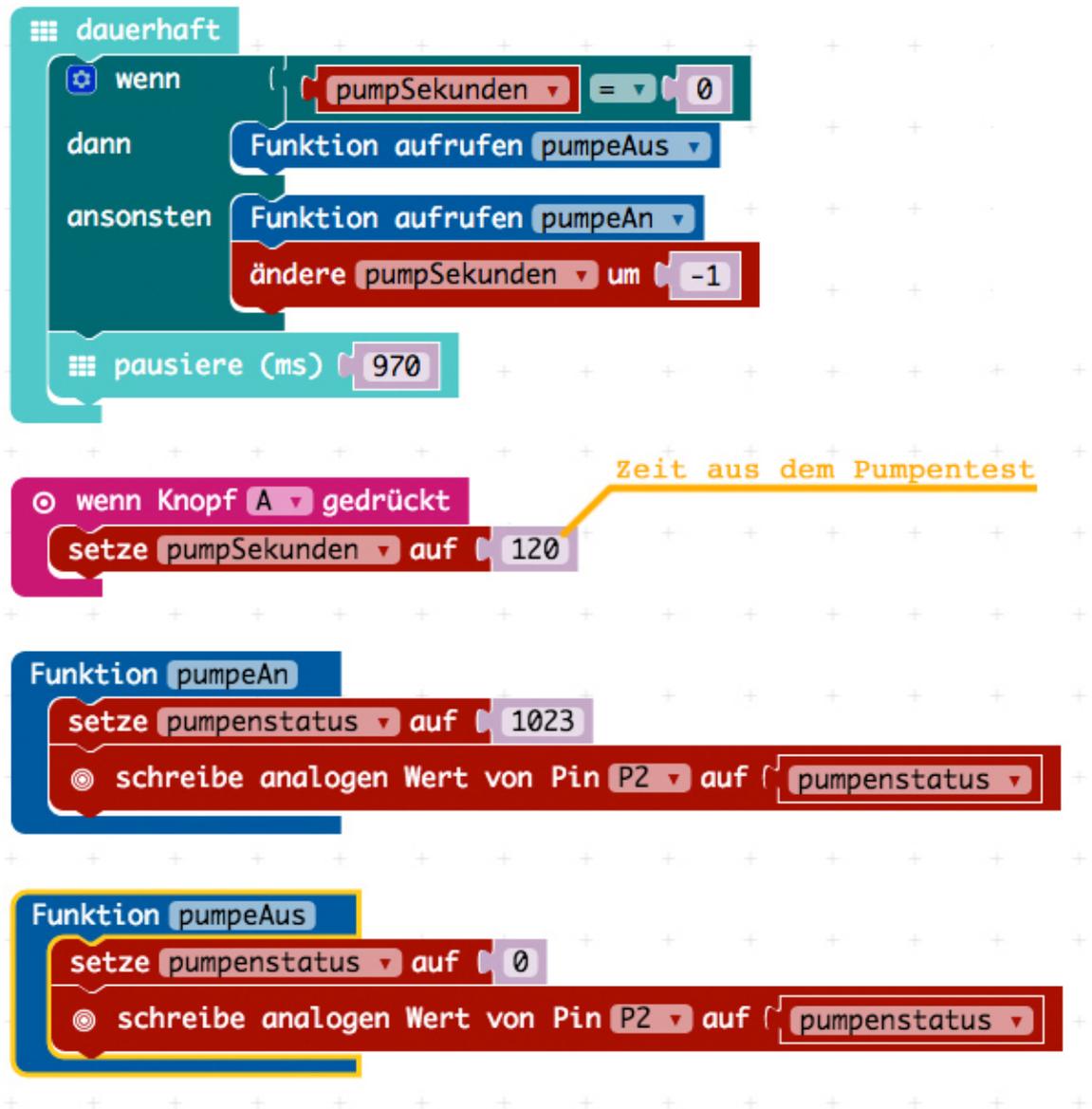
Fügt am Ende der „dauerhaft“-Funktion den Befehl „pausiere(970ms)“ ein, so wird die Funktion nur einmal pro Sekunde aufgerufen. (30ms benötigt der Calliope im  $\emptyset$  um die Funktion auszuführen, daher 970ms)

Erstellt die Variable „pumpSekunden“, welche angibt, wie viele Sekunden die Pumpe eingeschaltet sein soll. Wenn ein Knopf gedrückt wird, soll diese Variable nun auf die beim Testlauf gemessene Pumpzeit gesetzt werden.

In der „dauerhaft“-Funktion wird nun geregelt, ob die Pumpe an oder aus sein soll: Wenn „pumpSekunden = 0“ ist, ist die Pumpzeit vergangen und die Pumpe soll aus sein, ansonsten soll die Pumpe an sein und „pumpSekunden“ um -1 geändert werden (da beim Aufrufen der Funktion eine Sekunde vergangen ist).

## Programmier-Aufgabe 2 Lösungsvorschlag

---



### Programmier-Aufgabe 3

## Zeit messen seit die Pumpe das letzte Mal angeschaltet wurde

---

Der Calliope hat leider keine Funkuhr eingebaut, da wir das Beet nur alle 24h gießen lassen wollen, müssen wir die Zeit seit dem letzten Gießen selbst messen.

Erweitert dafür das Programm um Variablen für Sekunden, Minuten und Stunden seit dem letzten Gießen. Diese sollen nun mit einer Funktion wie auf einer digitalen Uhr hochgezählt werden. Die Funktion kann in der „dauerhaft“-Funktion jede Sekunde aufgerufen werden.

Wenn die Pumpe angeschaltet wird, müssen die Variablen wieder auf 0 gesetzt werden.

### Programmier-Aufgabe 3

## Hinweis

---

Erstellt Variablen mit den Namen „sekunden\_seit\_giessen“, „minuten\_seit\_giessen“, „stunden\_seit\_giessen“.

Erstellt eine Funktion „sekundeVergangen“, welche diese Variablen wie auf einer digitalen Uhr jede Sekunde verändert. Dafür kann diese Funktion auch in der „dauerhaft“-Funktion aufgerufen werden. Die Funktion „sekundeVergangen“ erhöht die Variable „sekunden\_seit\_giessen“ um +1. Wenn 60 Sekunden vergangen sind, wird „sekunden\_seit\_giessen“ auf 0 gesetzt und „minuten\_seit\_giessen“ um +1 erhöht. Wenn 60 Minuten vergangen sind, passiert das gleiche bei den Minuten und Stunden.

Erweitert die „pumpeAn“-Funktion, sodass sie die drei Zeit Variablen auf 0 stellt wenn die Pumpe angeschaltet wird, um so die Zeit seit dem Gießen erneut von 0 an zu zählen.

## Programmier-Aufgabe 3 Lösungsvorschlag

```

dauerhaft
  wenn (pumpSekunden = 0)
  dann
    Funktion aufrufen pumpeAus
  ansonsten
    Funktion aufrufen pumpeAn
    ändere pumpSekunden um -1
  pausiere (ms) 970
  Funktion aufrufen sekundeVergangen

wenn Knopf A gedrückt
  setze pumpSekunden auf 120

Funktion pumpeAn
  setze pumpestatus auf 1023
  schreibe analogen Wert von Pin P2 auf pumpestatus
  setze sekunden_seit_giessen auf 0
  setze minuten_seit_giessen auf 0
  setze stunden_seit_giessen auf 0

Funktion pumpeAus
  setze pumpestatus auf 0
  schreibe analogen Wert von Pin P2 auf pumpestatus

Funktion sekundeVergangen
  ändere sekunden_seit_giessen um 1
  wenn (sekunden_seit_giessen = 60)
  dann
    setze sekunden_seit_giessen auf 0
    ändere minuten_seit_giessen um 1
  wenn (minuten_seit_giessen = 60)
  dann
    setze minuten_seit_giessen auf 0
    ändere stunden_seit_giessen um 1

```

## Programmier-Aufgabe 4

### Pumpe alle 24h einschalten

---

Erweitert das Programm, sodass alle 24h die Pumpe für die benötigte Zeit zum Gießen eingeschaltet wird.

Lasst die Zeit seit dem letzten Gießen per Knopfdruck auf dem Display anzeigen, so kann man überprüfen wann gegossen wurde.

## Programmier-Aufgabe 4

### Hinweis

---

Überprüft in der „dauerhaft“-Funktion, ob die Variable „stunden\_seit\_gießen“ 24 beträgt. Wenn das der Fall ist, setzt die „pumpSekunden“ auf die zum Gießen benötigte Zeit. Dadurch wird alle 24h die Pumpe für die benötigte Zeit angeschaltet.

Schrift könnt ihr mit der Funktion „zeige Text“ anzeigen lassen. Um die Zeit wie auf einer Uhr darzustellen (HH:MM:SS), könnt ihr die Text-Funktion „verbinde“, welche zwei Textteile miteinander verbindet, nutzen. Wenn ihr die Funktion ineinander verschachtelt aufruft, könnt ihr die drei Zeitvariablen mit Doppelpunkten miteinander verbinden und anschließend anzeigen lassen.

## Programmier-Aufgabe 4 Lösungsvorschlag

```

dauerhaft
  wenn pumpSekunden = 0
  dann Funktion aufrufen pumpeAus
  ansonsten Funktion aufrufen pumpeAn
  ändere pumpSekunden um -1

  wenn stunden_seit_giessen = 24
  dann setze pumpSekunden auf 120

  pausiere (ms) 970
  Funktion aufrufen sekundeVergangen

wenn Knopf A gedrückt
  setze pumpSekunden auf 120

wenn Knopf B gedrückt
  zeige Text
  verbinde stunden_seit_giessen
  verbinde ":"
  verbinde minuten_seit_giessen
  verbinde ":"
  verbinde sekunden_seit_giessen

Funktion pumpeAn
  setze pumpestatus auf 1023
  schreibe analogen Wert von Pin P2 auf pumpestatus
  setze sekunden_seit_giessen auf 0
  setze minuten_seit_giessen auf 0
  setze stunden_seit_giessen auf 0

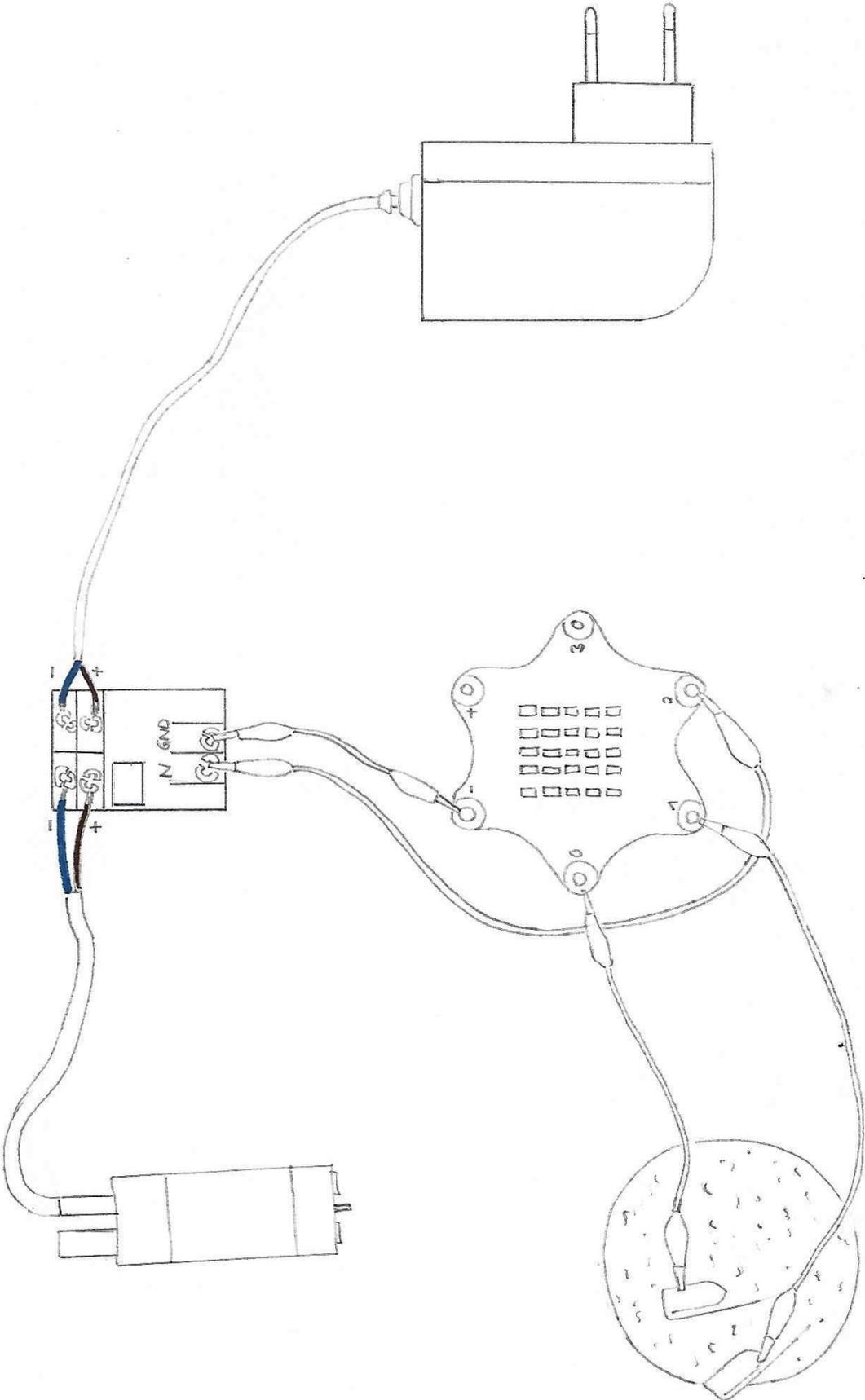
Funktion pumpeAus
  setze pumpestatus auf 0
  schreibe analogen Wert von Pin P2 auf pumpestatus

Funktion sekundeVergangen
  ändere sekunden_seit_giessen um 1
  wenn sekunden_seit_giessen = 60
  dann setze sekunden_seit_giessen auf 0
  ändere minuten_seit_giessen um 1
  wenn minuten_seit_giessen = 60
  dann setze minuten_seit_giessen auf 0
  ändere stunden_seit_giessen um 1

```

Für die Programmier-Aufgabe 5

Verkabelung der Büroklammern zur Bodenfeuchtemessung



## Programmier-Aufgabe 5

### Testreihe zur Bodenfeuchtemessung

---

Der Calliope soll nur gießen, wenn der Boden trocken ist. Wir nutzen zwei Büroklammern um mit dem Calliope zu messen, wie viel Strom die Erde leitet. Über diesen Wert können wir die Bodenfeuchte ableiten. Wir verkabeln den Calliope mit den Büroklammern wie in der Skizze.

Um eine Büroklammer unter Strom zu setzen, geben wir den digitalen Wert „1“ an P0 aus. Anschließend messen wir, welcher analoge Wert bei der anderen Büroklammer, bei P1, ankommt. Der Wert repräsentiert die Leitfähigkeit der Erde.

Schreibt ein neues Programm um eine Testreihe, zum Ermitteln der Bodenfeuchte, durchzuführen. Lasst dafür den gemessenen Wert auf dem Display anzeigen. Ermittelt anschließend, bei welchen Werten die Erde trocken bzw. nass ist.

## Programmier-Aufgabe 5

### Hinweis

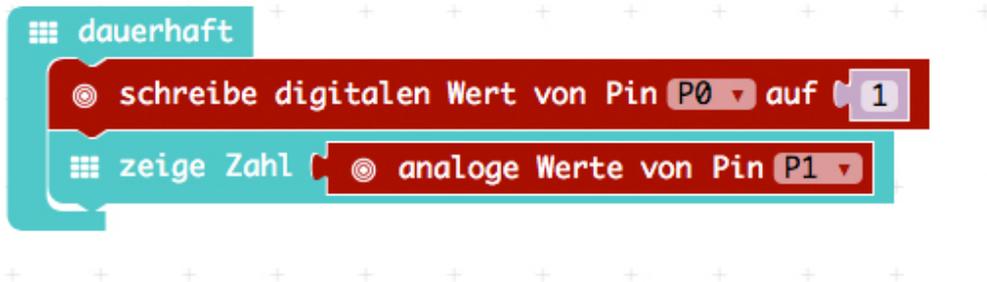
---

Nutzt in der „Dauerhaft Funktion“ die Funktion „schreibe digitalen Wert von Pin P0 auf 1“, um die eine Büroklammer dauerhaft unter Strom zu setzen. Lasst darauffolgend den gemessenen Wert mit der Funktion „zeige Zahl (analogen Wert von Pin P1)“ anzeigen. Das Display zeigt nun repräsentativ wie viel Strom die Erde leitet.

## Programmier-Aufgabe 5

### Lösungsvorschlag

---



## Test-Ergebnis

---

Bei Werten ab 900 ist die Erde feucht. Also wenn über die Büroklammer weniger als 900 gelesen wird, muss das Beet gegossen werden. Ansonsten kann das Wasser gespart werden.

## Programmier-Aufgabe 6

### Pumpe alle 24h bei Trockenheit einschalten

---

Erweitert das Programm, sodass alle 24h vor dem Einschalten der Pumpe die Bodenfeuchte gemessen wird und sie nur bei Trockenheit eingeschaltet wird.

## Programmier-Aufgabe 6

### Hinweis

---

Erweitert das Programm um eine Funktion zum Messen der Bodenfeuchte. Dafür könnt ihr Teile aus dem Programm zur Testreihe der Bodenfeuchtemessung nutzen. Erstellt außerdem die Variable „bodenfeuchte“, um den gemessenen Wert abzuspeichern.

Misst in der „dauerhaft“-Funktion nach 24h die Bodenfeuchte. Anschließend überprüft ihr vor dem Gießen mit einer Wenn-Dann-Logik ob der Boden trocken ist, um nur bei „bodenfeuchte < 900“ die Pumpe anzuschalten.

## Programmier-Aufgabe 6 Lösungsvorschlag

```

dauerhaft
  wenn ( pumpSekunden == 0 )
  dann Funktion aufrufen pumpeAus
  ansonsten Funktion aufrufen pumpeAn
  ändere pumpSekunden um -1

  wenn ( stunden_seit_giessen ≥ 24 )
  dann Funktion aufrufen messeBodenfeuchte
  wenn ( bodenfeuchte < 900 )
  dann setze pumpSekunden auf 120

  pausiere (ms) 970
  Funktion aufrufen sekundeVergangen

wenn Knopf A gedrückt
  setze pumpSekunden auf 120

wenn Knopf B gedrückt
  zeige Text
  verbinde stunden_seit_giessen
  verbinde ":"
  verbinde minuten_seit_giessen
  verbinde ":"
  verbinde sekunden_seit_giessen

Funktion pumpeAn
  setze pumpestatus auf 1023
  schreibe analogen Wert von Pin P2 auf pumpestatus
  setze sekunden_seit_giessen auf 0
  setze minuten_seit_giessen auf 0
  setze stunden_seit_giessen auf 0

Funktion pumpeAus
  setze pumpestatus auf 0
  schreibe analogen Wert von Pin P2 auf pumpestatus

Funktion sekundeVergangen
  ändere sekunden_seit_giessen um 1
  wenn ( sekunden_seit_giessen == 60 )
  dann setze sekunden_seit_giessen auf 0
  ändere minuten_seit_giessen um 1
  wenn ( minuten_seit_giessen == 60 )
  dann setze minuten_seit_giessen auf 0
  ändere stunden_seit_giessen um 1

Funktion messeBodenfeuchte
  schreibe digitalen Wert von Pin P0 auf 1
  setze bodenfeuchte auf ( analoge Werte von Pin P1

```