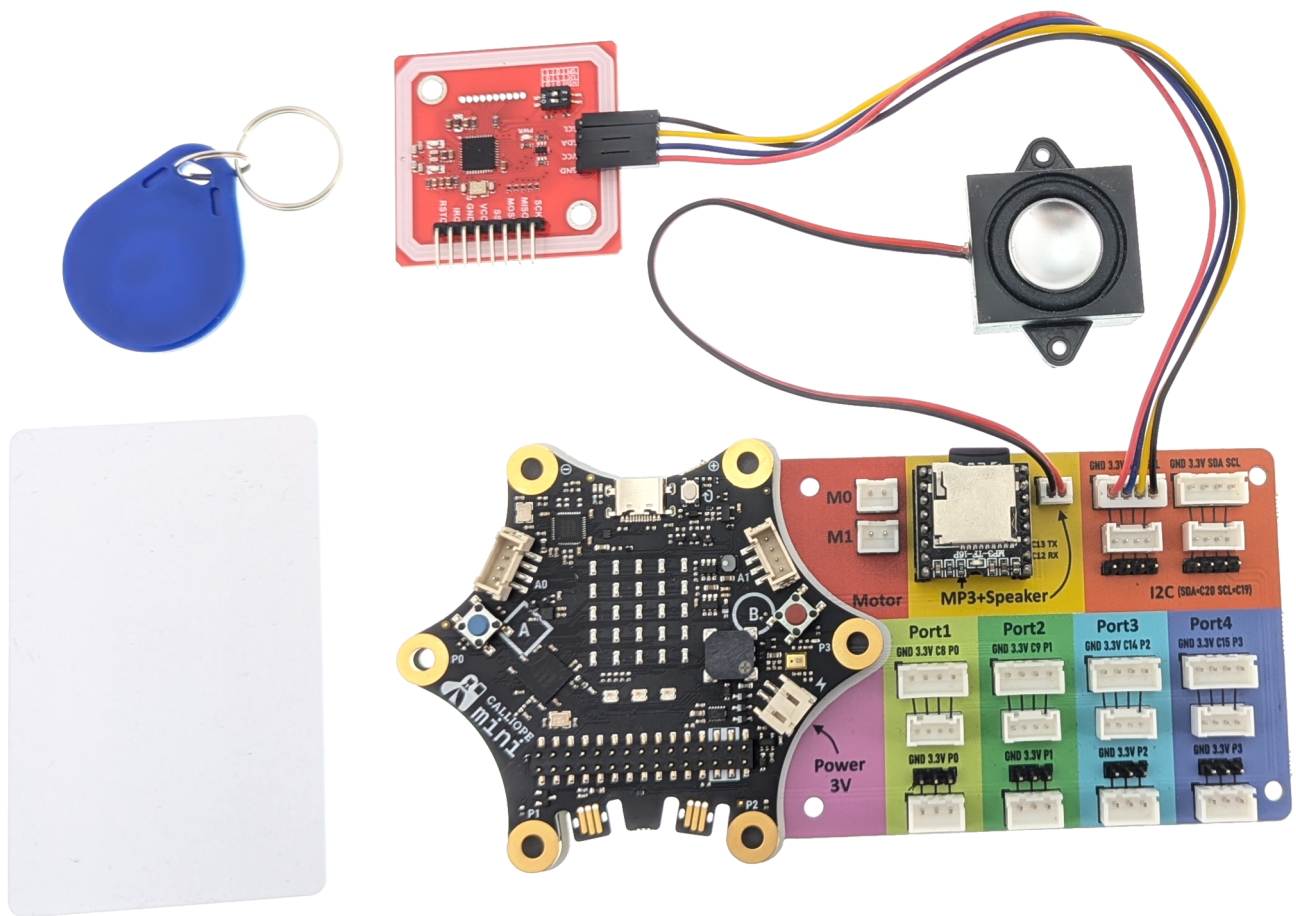


## Expander für Calliope Mini V3

Modul zur Erweiterung der Programmierfähigkeiten mit externen Komponenten

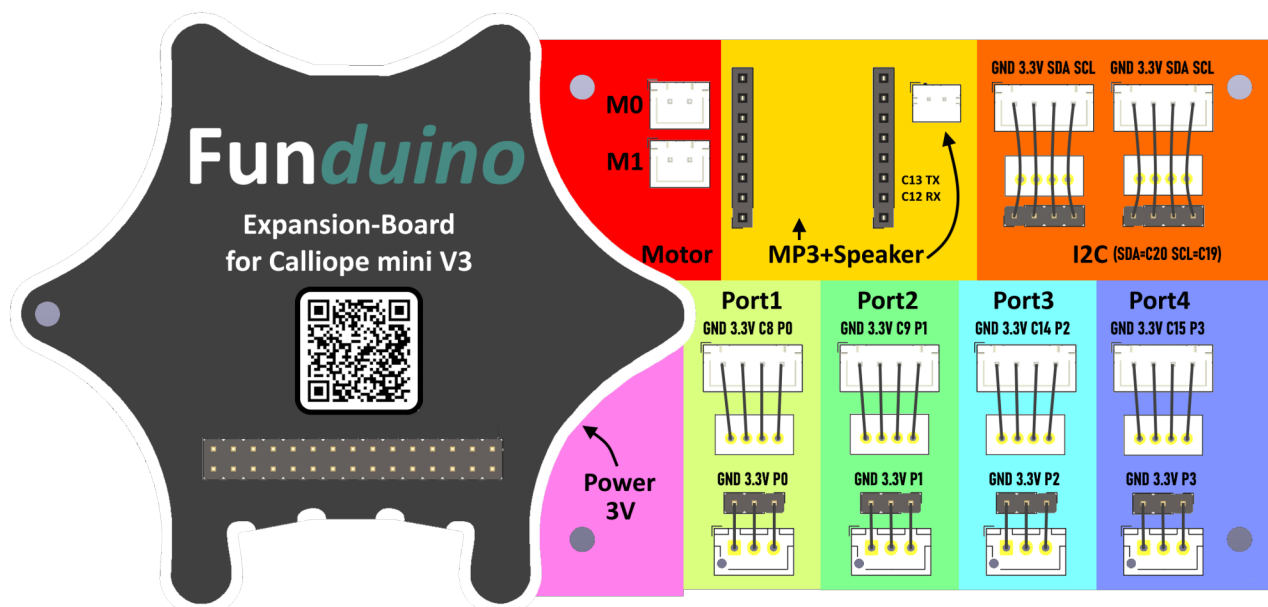
## Anleitung und Projektbuch



# Inhaltsverzeichnis

1. Grundlagen.....	5
1.1 Was ist der Calliope Mini V3?.....	5
1.2 Was ist ein „Expander für Calliope Mini V3“?.....	6
1.2.1 Didaktische Vorteile im Unterricht.....	6
1.2.2 Didaktischer Mehrwert des Expanders.....	7
1.2.3 Technische Details.....	8
1.3 Expander und seine Komponenten.....	8
1.3.1 Expander.....	9
1.3.2 LED-Modul.....	10
1.3.3 Taster.....	10
1.3.4 Drehregler.....	10
1.3.5 Feuchtigkeitssensor.....	11
1.3.6 Motor.....	11
1.3.7 Servomotor.....	11
1.3.8 MP3-Modul und Lautsprecher.....	12
1.3.9 RFID Sender und Empfänger.....	12
1.3.10 OLED Display.....	13
2. Programmieren.....	14
2.1 Programmieroberfläche.....	14
2.2 Makecode Projekt erstellen.....	14
3. Programmieren.....	19
3.1 Wie schreibe ich ein einfaches Programm?.....	19
3.2 Beispiel für Serielle Kommunikation.....	22
3.2.1 Text an den Computer senden.....	23
3.2.2 Gesendete Daten am Computer ansehen.....	26
4. Praxisaufgaben.....	29
4.1 Eine Leuchtdiode blinken lassen.....	29
4.2 Taster verwenden.....	32
4.3 Drehregler und Serielle Kommunikation.....	36
4.4 Drehregler verwenden.....	40
4.5 Feuchtigkeitssensor.....	45
Aufgabenteil 1 – Messwerte anzeigen.....	47
Aufgabenteil 2 – Messwerte verwenden.....	48
4.6 Elektromotor ansteuern.....	51
Aufgabenteil 1 – Wind erzeugen.....	52
Aufgabenteil 2 – Kühlung automatisieren.....	53

4.7 Servomotor.....	57
4.8 MP3 – Lass den Calliope sprechen!.....	61
Aufgabenteil 1 – SD-Karte vorbereiten.....	62
Aufgabenteil 2 – Eine MP3-Datei abspielen.....	62
Aufgabenteil 3 - Eine MP3-Datei abspielen, wenn es zu warm wird.....	65
4.9 RFID Karten lesen.....	68
Teilaufgabe 1 – RFID-TAG auslesen.....	69
Teilaufgabe 2 – Melodie in Abhängigkeit der Chipkarte abspielen.....	74
4.10 Tonibox nachbauen.....	79
4.11 Ein Display verwenden – LCD 1602 oder OLED.....	82
4.11.1 Ein Display verwenden – OLED.....	83
Teilaufgabe 1 - Ein OLED soll deinen Namen anzeigen.....	86
Teilaufgabe 2 - Ein OLED soll die Temperatur anzeigen.....	87
4.11.2 Ein Display verwenden – LCD.....	91
Teilaufgabe 1 - Ein LCD soll deinen Namen anzeigen.....	94
Teilaufgabe 2 - Ein LCD soll die Temperatur anzeigen.....	96
4.12 Ende oder Anfang?.....	99



# 1. Grundlagen

Um mit dem Calliope Mini V3 und dem Expanderboard arbeiten zu können, möchten wir zunächst die einzelnen Teile kennenlernen.

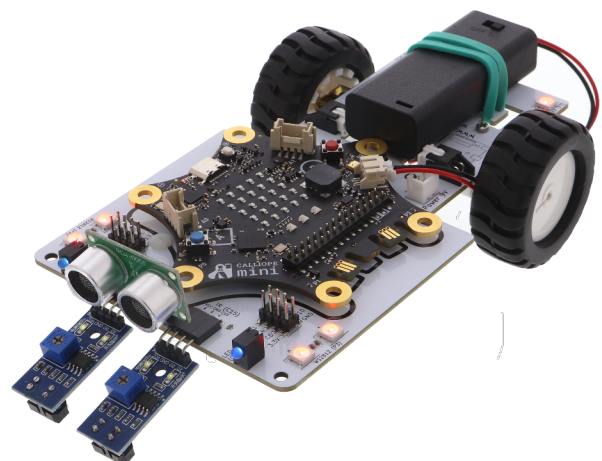
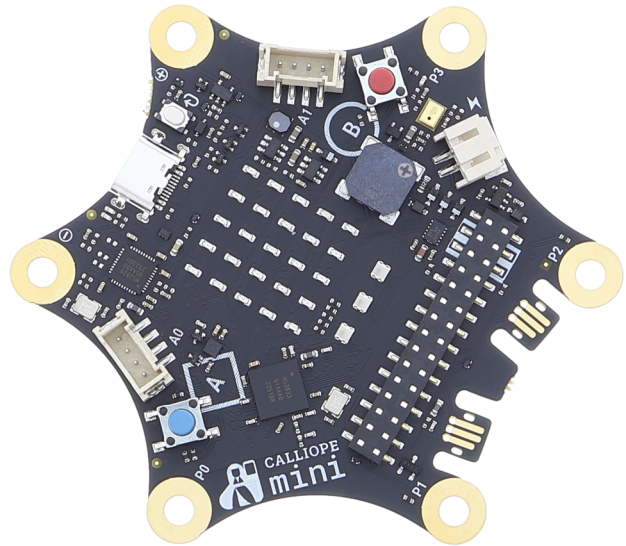
## 1.1 Was ist der Calliope Mini V3?

Der Calliope Mini V3 ist ein kleiner Computer, den man programmieren kann, um viele coole Dinge zu machen. Stell dir vor, es ist wie ein kleines, magisches Board, das du in deiner Hand halten kannst. Es hat bunte Lichter, Knöpfe und sogar Sensoren, die Dinge wie Bewegung oder Temperatur messen können. Mit dem Expanderboard von Funduino können diese Anwendungsmöglichkeiten sogar noch erweitert werden.

Mit dem Calliope Mini V3 kannst du lernen, wie man Programme schreibt, um ihn zu steuern. Zum Beispiel kannst du ihn so programmieren, dass er Musik spielt, Lichter leuchten lässt oder auf Bewegungen reagiert. Es ist ein bisschen wie ein Roboter, den du selbst zum Leben erwecken kannst – und du bestimmst, was er tut!

Der Calliope Mini V3 ist wurde für Kinder entwickelt, damit sie spielerisch lernen können, wie Computer funktionieren und wie man sie programmiert. Das macht nicht nur Spaß, sondern ist auch sehr nützlich für das spätere Leben.

Es gibt auch noch Interessantes Zubehör, wie den „Calle“. Calle ist ein Roboterfahrgestell, auf dem man den Calliope Mini V3 aufstecken kann. Zusammen kann man Calle so programmieren, das er einer Linie folgt, oder Objekten ausweicht.



## 1.2 Was ist ein „Expander für Calliope Mini V3“?

Expander für Calliope Mini V3 – Ideal für den MINT- und Informatikunterricht

Der Expander für Calliope Mini V3 ist das perfekte Zubehör für den Einsatz im MINT-Bereich. Insbesondere im Informatik- und Technikunterricht bietet er viele neue Möglichkeiten rund um den Einsatz des Calliope Mini V3. Mit ihm schaffen Sie spannende, praxisnahe Projekte für Schüler, die das Grundverständnis für Programmierung, Elektronik und Technik fördern. Speziell entwickelt, um Lehrkräften dabei zu helfen, Schülerinnen und Schülern ein tieferes Verständnis für Informatik und Mikroelektronik zu vermitteln. Alle beiliegenden Module sind gut dokumentiert und in einer umfassenden Anleitung beschrieben. Die Anleitungen sind schülergerecht und in leichter Sprache geschrieben. Dadurch können die Schüler selbstständig mit dem Material arbeiten.

### 1.2.1 Didaktische Vorteile im Unterricht

#### **Anleitungen**

In den zugehörigen Anleitungen sind der Expander und alle beiliegenden Module ausführlich und schülergerecht beschrieben.

#### **Förderung von Kreativität und Problemlösung**

Der Expander erlaubt es den Lernenden, eigene Projekte zu realisieren – von Sprachausgaben über den Bau eigener Tonboxen bis hin zu motorisierten Modellen. Dadurch entwickeln sie nicht nur technische Fähigkeiten, sondern auch Kreativität und Problemlösungskompetenzen.

Einfache Integration in den Unterricht: Die vorgefertigten Stecker und beschrifteten Kontakte machen es einfach, den Expander sicher und schnell zu verbinden – ideal für den Einsatz im Klassenraum ohne komplizierte Vorbereitung. So können sich Lehrkräfte ganz auf die didaktischen Inhalte konzentrieren.

#### **Vielseitigkeit und offene Erweiterungsmöglichkeiten**

Dank der universellen Anschlussmöglichkeiten lässt sich der Expander mit einer Vielzahl von Mikroelektronikkomponenten kombinieren. Das fördert den Forscherdrang der Schülerinnen und Schüler und bietet unzählige Möglichkeiten, technische Konzepte praxisnah zu erleben.

#### **Direkter Zugang zu allen wichtigen Kontakten**

Der Expander macht schwer zugängliche Calliope-Pins zugänglich und bietet auch speziell für Elektromotoren vorgesehene Ausgänge. Dies eröffnet vielfältige Einsatzmöglichkeiten im Bereich Robotik und Automatisierung – perfekt für die MINT-Fächer.

### **Praxisorientierte Technik- und Informatikprojekte**

Mit dem integrierten MP3-Player und dem leicht anschließbaren Lautsprecher können Lernende einfache Programmierprojekte realisieren, die sie in die Welt der Informatik einführen. Egal, ob im Rahmen von Projekttagen, im Informatikunterricht oder in technischen Arbeitsgemeinschaften – der Expander bietet praxisnahe Anwendungen, die technisches Verständnis und Programmierfähigkeiten fördern.

### **Keine zusätzliche Stromquelle erforderlich**

Der Expander benötigt keine eigene Stromversorgung, sondern kann problemlos über das Batteriefach des Calliope Mini betrieben werden – so bleibt der Aufbau im Unterricht einfach und effizient.

### **Perfekt für den Unterricht in Informatik und Technik (MINT)**

Mit dem Expander für Calliope Mini V3 können Sie Schülerinnen und Schülern die Welt der Mikrocontroller, Elektronik und Programmierung praxisnah und fächerübergreifend näherbringen. Ob als Einführung in die Informatik oder als Technikprojekt: Dieser Expander macht den Unterricht greifbar und spannend.

## **1.2.2 Didaktischer Mehrwert des Expanders**

Wer den Expander im Unterricht einsetzt, hat vermutlich schon vorher mit dem Calliope als einzelnes Modul gearbeitet. Welchen Mehrwert bietet an dieser Stelle der Expander?

Der Calliope verfügt zwar über eine Schnittstelle, an der alle Pins des Mikrocontrollers herausgeführt sind, jedoch ist die Verkabelung nur mit Breadboardkabeln häufig eine Hürde, die Lehrkräfte und Schüler vor Probleme stellt. An dieser Stelle bietet der Expander einen perfekten Übergang zwischen dem „All-in-One“ Board Calliope und der weiterführenden Nutzung externer elektrischer Module sowie deren Programmierung.

### **Zusammengefasst**

- Erlernen der Verwendung von komplexen Bauteilen mit dem Calliope mini V3.
- Nutzung von Sprachausgabe am Calliope mit Hilfe eines MP3-Moduls.
- Nutzung von RFID Funkkommunikation.
- Erlernen der Verkabelung komplexer Bauteile mit vorkonfektionierten Steckverbindern aber auch mit individueller Verkabelung zur Hinführung zu Individuellen Mikroelektronik-Projekten.
- Erlernen der Nutzung von Programmbibliotheken zur Erweiterung der blockbasierten Programmierung

- Erlernen der technischen Grundlagen digitaler Daten.
- Erlernen der technischen Grundlagen analoger Daten.
- Erlernen der Nutzung des I2C Datenbus Systems.
- Erlernen der Zusammenhänge von Programmbausteinen und externen Modulen und Funktionen.
- Erlernen der Kombination von erweiterten elektronischen Komponenten in Projekten.
- Hinführung zu erweiterten technischen Systemen im Bereich der Mikroelektronik wie bspw. Arduino.

### **1.2.3 Technische Details**

Es ist ein Lautsprecher enthalten, der ganz leicht per JST-Stecker am Expander befestigt werden kann.

Es kann nichts falsch verkabelt werden, da die Stecker vor-konfektioniert sind. Die Kontakte des Calliope werden herausgeführt und über herkömmliche Buchsen zugänglich gemacht. Alle herausgeführten Kontakte können per JST-Stecker, Grove Stecker oder mit herkömmlichen Breadboardkabeln verbunden werden.

Über die universellen Anschlussmöglichkeiten ist man an kein teures System gebunden. Das System lässt sich mit beliebigen Komponenten aus dem Mikroelektronikbereich erweitern. Egal ob die Artikel aus Fernost oder aus der EU stammen.

Der Calliope V3 mini verfügt über zwei Ausgänge, die speziell für Elektromotoren vorgesehen sind. Diese Ausgänge sind auf dem Expander über JST-Buchsen direkt zugänglich.

Für die Spannungsversorgung des Calliope und des Expanderboards wird keine zusätzliche Energiequelle benötigt. Das Batteriefach des Calliope reicht für die meisten Anwendungen vollkommen aus.

Alle verfügbaren Kontakte des Calliope wurden auf dem Expander beschriftet und können zu 100% genau zugeordnet werden.

Erst durch den Expander werden viele schwer zugängliche Kontakte des Calliope nutzbar gemacht. Die Kontakte P0, P1, P2, P3, C8, C9, C12, C13, C14, C15, C19, C20, M1 und M2 können nun direkt am Expander verwendet werden. Dabei werden C12 und C13 vom abnehmbaren MP3-Modul verwendet und mit den Kontakten C19 und C20 sind auch alle Möglichkeiten der I2C Konnektivität nutzbar.

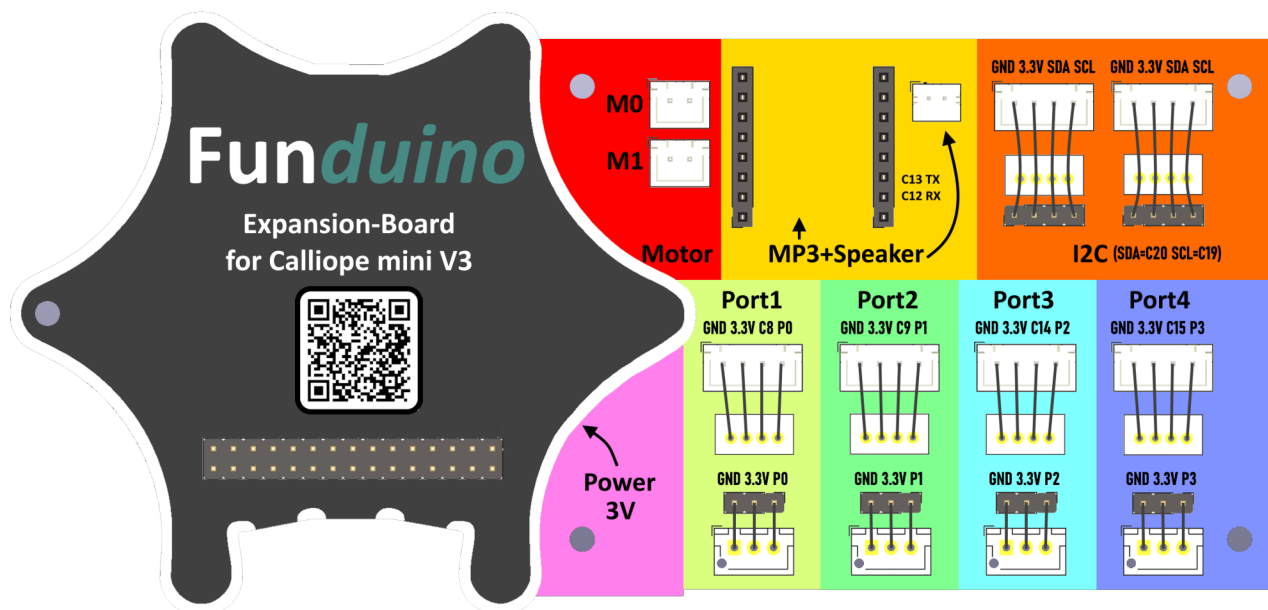
## **1.3 Expander und seine Komponenten**

Der Expander für den Calliope Mini V3 gibt es als Set mit den folgenden Bauteilen, in den Anleitungen aufeinander aufbauend beschrieben und programmiert werden.

## 1.3.1 Expander

Die Expander Grundplatine ist in farbige Sektionen unterteilt. Im linken schwarzen Feld wird der Calliope Mini V3 aufgesteckt. Die farbigen Bereiche auf der rechten Seite haben folgende Bedeutungen.

Rot	Anschlüsse für zwei Gleichstrom Elektromotoren an den Kontakten M0 und M1
Gelb	Anschluss für das MP3-Modul und rechts daneben der Anschluss für den Lautsprecher
Orange	Anschlüsse für Module die über den I2C Datenbus angesteuert werden. Die Linien zeigen die verbundenen Kontakte
Rosa	Markierung für die Spannungsversorgung per Batterie
Hellgrün	Port1 mit den Kontakten C8 und P0 – die Linien zeigen die verbundenen Kontakte
Grün	Port2 mit den Kontakten C9 und P1 – die Linien zeigen die verbundenen Kontakte
Türkis	Port3 mit den Kontakten C14 und P2 – die Linien zeigen die verbundenen Kontakte
Blau	Port4 mit den Kontakten C15 und P3 – die Linien zeigen die verbundenen Kontakte



### 1.3.2 LED-Modul

Einzelnes LED-Modul mit einer Leuchtdiode und JST-Buchse.



### 1.3.3 Taster

Modul mit einem Taster und JST-Buchse.



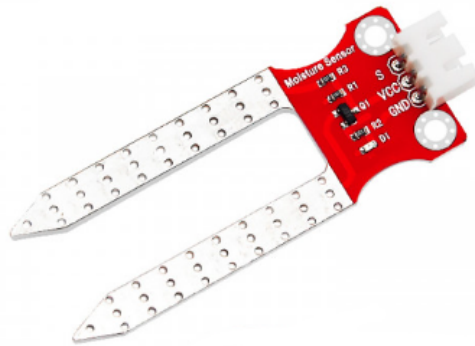
### 1.3.4 Drehregler

Einzelnes Modul mit einem Potenziometer und Drehreglerkappe. Auch hier sorgt eine JST-Buchse für eine unkomplizierte Verkabelung.



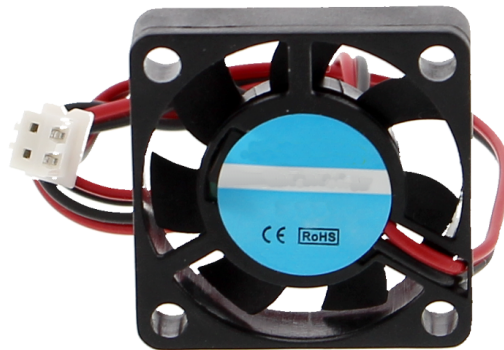
### 1.3.5 Feuchtigkeitssensor

Der Sensor kann direkt zur Kontaktmessung von Feuchten Böden, Schwämmen, Tüchern etc. verwendet werden. Er eignet sich nicht für die Messung von Luftfeuchtigkeit. Über die JST-Buchse kann bei der Verkabelung nichts schief gehen.



### 1.3.6 Motor

Als Elektromotor verwenden wir einen klassischen Lüfter, in dem ein Elektromotor integriert ist. Der Stecker des Lüfters passt in die Motorbuchse des Expanders und wird von der Leistungselektronik des Calliope mit Strom versorgt. Der Expander benötigt keine gesonderte Hardware zur Versorgung des Motors.



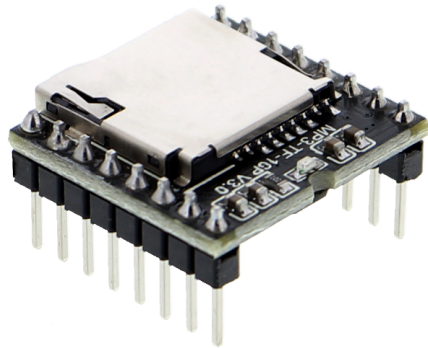
### 1.3.7 Servomotor

Bei dem Servo handelt es sich um einen Servo SG90. Ein typisches Bauteil aus dem Modellbaubereich. Wie die meistens Servos verfügt er über einen JR-Stecker. Dieser wird am Expander auf die vorhandenen Header-Pins aufgesteckt.



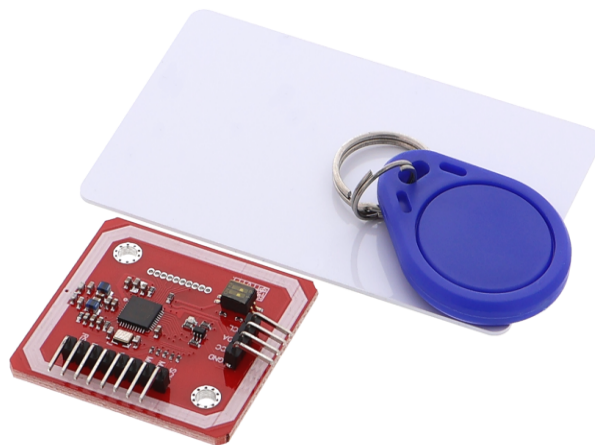
### 1.3.8 MP3-Modul und Lautsprecher

Für das MP3-Modul verfügt der Expander über einen dafür vorgesehenen Sockel. Das Modul wird dort so eingesteckt, dass die SD-Karte vom Rand des Expanders entnommen werden kann.



### 1.3.9 RFID Sender und Empfänger

Der RFID-Empfänger ist bereits mit einem Kabel versehen und kann in die vorhandenen JST-Buchsen im I2C Bereich des Expanders eingesteckt werden. Außerdem sind zwei RFID-Tags, also Sender, enthalten. Es handelt sich um eine Chipkarte und um einen Schlüsselanhänger.

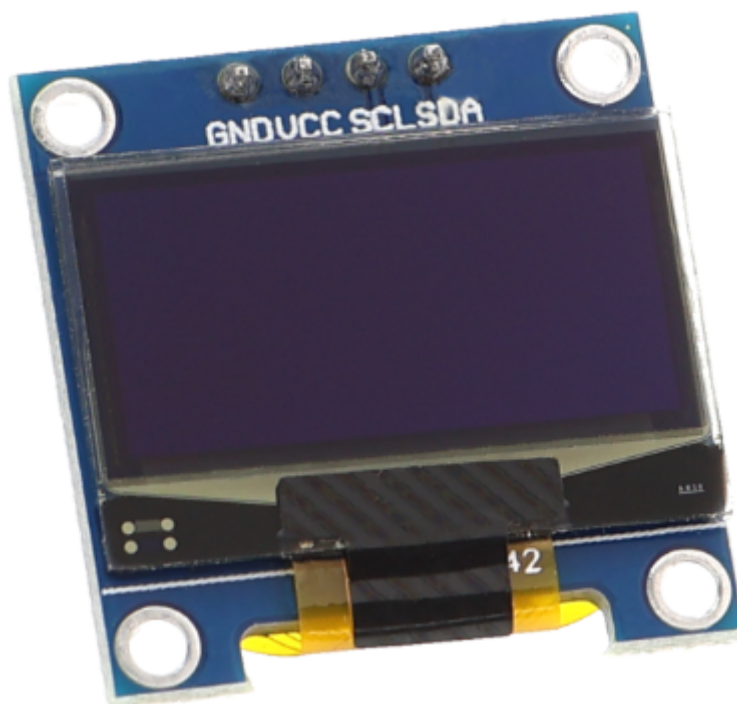


Ganz wichtig ist bei dem Modul, dass es über zwei Schiebeschalter in den I2C-Modus gebracht wird. Dazu wird die Schutzfolie über den Schaltern entfernt und der eine Schalter wird auf „ON“ gestellt, wie auf dem folgenden Bild zu sehen ist.



### 1.3.10 OLED Display

Das Display eignet sich besonders gut, um Informationen zu teilen. Zum Beispiel die gemessene Temperatur oder einen Timer. Das OLED hat auch der Rückseite vier Kontakte. Es handelt sich um GND, VCC, SCL und SDA. Auf der Vorderseite sind die Bezeichnungen an den Kontakten aufgedruckt. Mit Hilfe von Kabeln kann das OLED im I2C Bereich des Expanders eingesteckt werden.



## 2. Programmieren

Nun möchten wir mit der Programmierung beginnen. Bevor wir jedoch richtig loslegen, schauen wir uns an, wie man den Calliope programmieren kann. Wenn du schon mit dem Calliope gearbeitet und programmiert hast, kannst du diesen Teil überspringen.

### 2.1 Programmieroberfläche

Um einen Minicomputer wie den Calliope programmieren zu können, benötigt man einen Computer und eine Art Software, mit der man das Programm für den Calliope schreiben kann.

Diese Software für den Calliope heißt Makecode.

Makecode muss man nicht auf dem Computer installieren, man kann Sie einfach wie eine Internetseite im Internet aufrufen und loslegen. Diese Internetseite heißt:

**<https://Makecode.calliope.cc>**

Das fertige Programm wird im Anschluss direkt vom Browser – also von der Internetseite – auf den Calliope übertragen.

### 2.2 Makecode Projekt erstellen

In dieser Anleitung lernst du Schritt für Schritt, wie du dein erstes Projekt in Makecode startest.

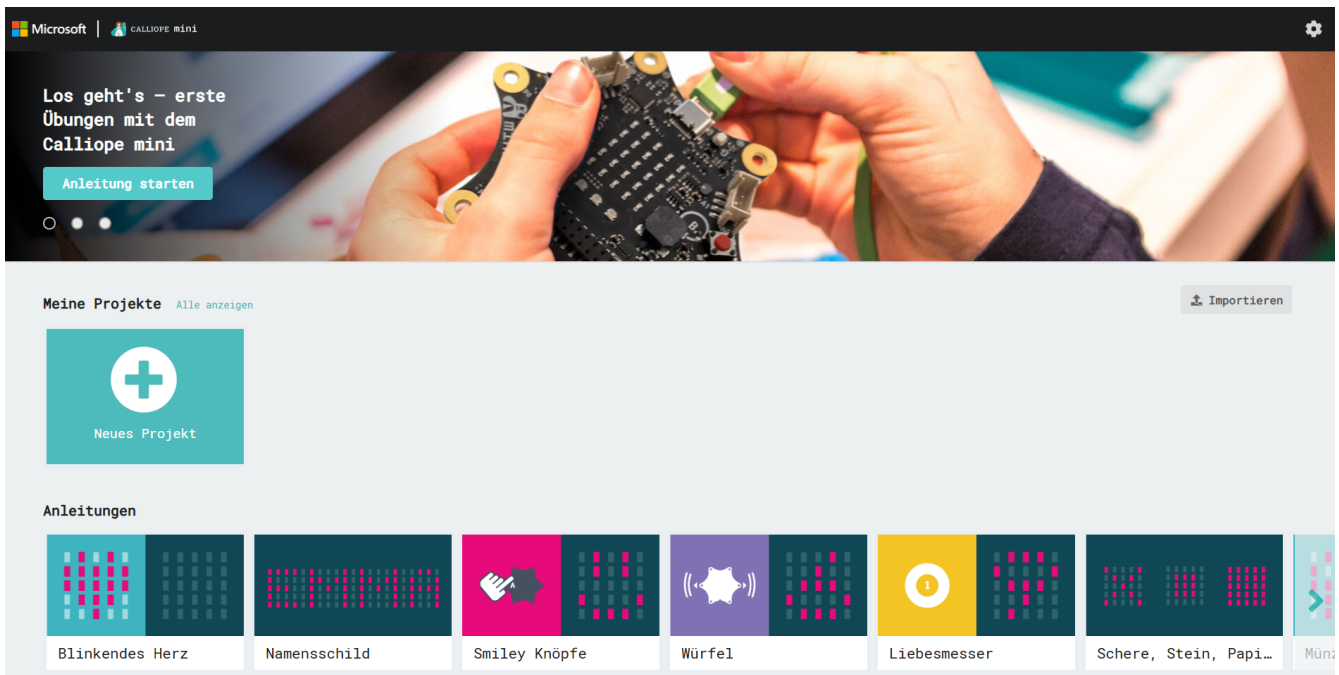
#### Schritt 1: Makecode öffnen

Starte deinen Computer oder dein Tablet und gehe ins Internet.

Gib in die Suchleiste „Makecode Calliope“ ein und klicke auf den ersten Link, der erscheint. Er lautet:

**<https://Makecode.calliope.cc>**

Jetzt bist du auf der Makecode-Seite. Hier kannst du Programme für den Calliope Mini erstellen.



## Schritt 2: Ein neues Projekt erstellen

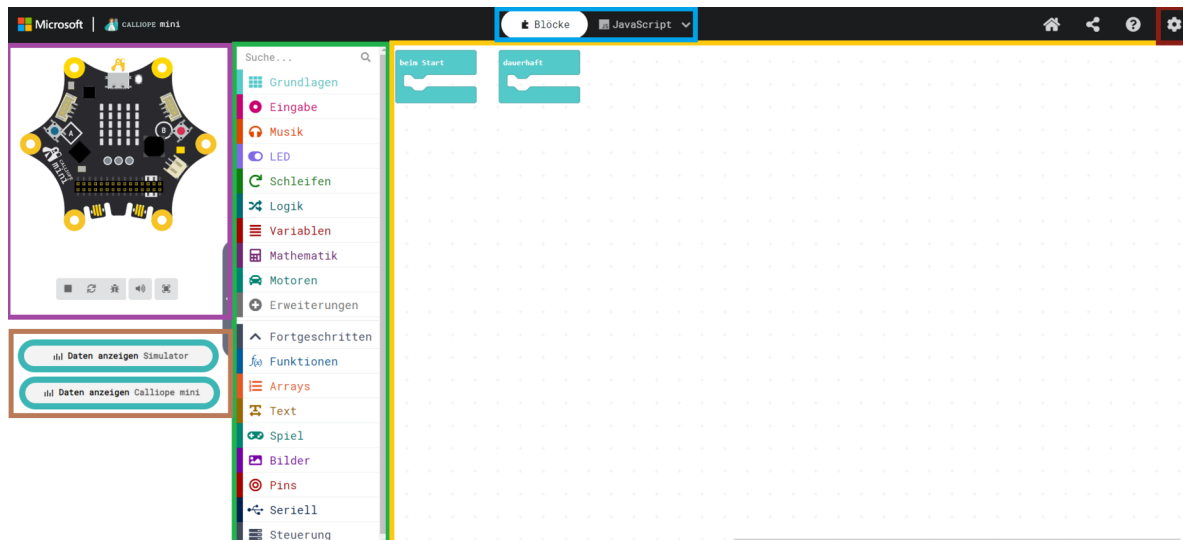
Auf der Startseite von Makecode siehst du eine große Schaltfläche mit der Aufschrift „Neues Projekt“. Klicke darauf.



Jetzt öffnet sich ein Fenster, in dem du deinem Projekt einen Namen geben kannst. Überlege dir einen coolen Namen, zum Beispiel „Mein erstes Projekt“ oder „Calleprogramm Nummer1“. Klicke dann auf „Erstellen“. Jetzt hast du ein leeres Projekt und kannst loslegen!

### Schritt 3: Die Programmieroberfläche kennenlernen

Damit es etwas übersichtlicher wird, haben wir die verschiedenen Bereiche der Benutzeroberfläche farbig markiert.



In der Mitte siehst du eine große weiße Fläche mit einem gelben Rahmen. Das ist dein Programmierbereich, wo du die Programmblöcke hinziehst. Wir starten mit dem Block „beim Start“ und dem Block „dauerhaft“. Für diejenigen, die mit der Arduino IDE gearbeitet haben, ist dies gleichzusetzen mit „setup“ und „loop“.

Links siehst du einen kleinen Calliope mit lila Rahmen. Das ist die Vorschau, in der du direkt sehen kannst, was dein Programm macht. Es ist also eine Simulation des echten Calliope.

Daneben sind viele bunte Blöcke mit einem grünen Rahmen. Das sind die Programmier-Bausteine oder Programmbausteine, mit denen du dein Projekt „baust“. Hier gibt es die Bereiche **Grundlagen** und **Fortgeschritten**. Um beispielsweise in die Kategorie „Pins“ zu gelangen, müssen wir die Hauptkategorie „Fortgeschritten“ aufklappen. Ein paar wichtige Kategorien sind hier die Kategorien „Pins“, in der wir Blöcke wie „Digital Lesen“, „Analog Lesen“, „Servo“ etc. finden. „Seriell“ ist auch wichtig, da wir darüber Daten an unseren Computer senden können, wie z. B. Lichtwert, Knopfstatus etc. Damit können wir die ganzen Daten analysieren und beispielsweise beschließen, dass wir eine LED einschalten wollen, wenn ein bestimmter Lichtwert erreicht wurde.

Oben am Bildrand haben wir in der Skizze einen blauen Kasten. Dort befinden sich die Optionen „Blöcke“ und „JavaScript“. Wenn man auf den Pfeil bei „JavaScript“ klickt, findet man eine weitere Option, nämlich „Python“. Für diejenigen, die sich später mit dem Calliope vertraut gemacht haben, ist das eine sehr interessante Idee, um zu lernen, wie dieser Code eigentlich funktioniert.

```
1 basic.showString("hi!")
2
3 basic.forever(function () {
4
5 })
6
```

Dann haben wir oben rechts einen ganz kleinen Kasten mit einem Zahnrad. Hier ist der Menüpunkt Einstellungen. Dort gibt es Optionen wie „Projekteinstellungen“, wo wir den Namen des Projektes ändern und andere kleine Einstellungen vornehmen können.



### 2.3 Serielle Kommunikation

Wenn du noch nie mit einem Calliope gearbeitet hast, ist der folgende Abschnitt schwer zu verstehen. Du kannst ihn erst mal weglassen, wenn du Anfänger bist. Wenn wir aber später Messwerte von Sensoren verwenden wollen, muss du diesen Abschnitt unbedingt lesen und verstehen.

Das Expanderboard ist dafür vorgesehen, viele weitere Module, wie Sensoren, mit dem Calliope zu verbinden. Sensoren geben uns die Möglichkeit, Eigenschaften unserer Umwelt digital zu erfassen. Ein einfaches Beispiel ist ein Temperatursensor.

Genau wie unsere menschliche Haut kann ein Temperatursensor Unterschiede der Temperatur feststellen. Unsere Haut trifft aber keine Entscheidung darüber, ob etwas zu heiß ist.

Unser Gehirn verarbeitet die „Daten“, die es von der Haut empfängt und entscheidet was passieren soll. Zum Beispiel meldet das Gehirn an die Muskeln der Hand „Nimm schnell die Finger von der heißen Herdplatte“.

Der Calliope und ein angeschlossener Temperatursensor funktionieren ähnlich. Der Temperatursensor sendet die Temperaturwerte in elektrischer Form an den Calliope. Hier wird dann entschieden, was passieren soll, wenn ein bestimmter Messwert erreicht wurde. **An dieser Stelle bist du gefragt.** Wenn du eine Temperaturwarnanlage für die Schule bauen sollst, die bei 30°C lauten Lärm macht, musst du im Programm festlegen, dass bei 30°C genau das passieren soll. Du musst es also selber in das Programm schreiben.

Aber welche Werte kommen genau am Calliope an und wie kannst du diese Werte sehen und verarbeiten?

**Dazu benötigst du die Serielle Kommunikation!**

Wie man die Serielle Kommunikation verwendet, lernst du später im Bereich **Programmieren.**

### 3. Programmieren

Bevor wir mit dem Expanderboard starten, schreiben wir zur Wiederholung erst mal ein kleines Programm.

#### 3.1 Wie schreibe ich ein einfaches Programm?

Aufgabe:

Unser Programm soll ein Linienmuster auf der LED-Matrix des Calliope anzeigen, wenn der Knopf A gedrückt wird.

Ziehe den Block „wenn wahr dann“ aus der linken Leiste in den Programmierbereich.



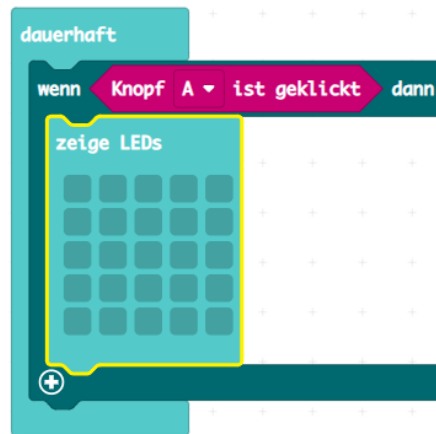
Diesen Block heften wir an den Block „dauerhaft“.



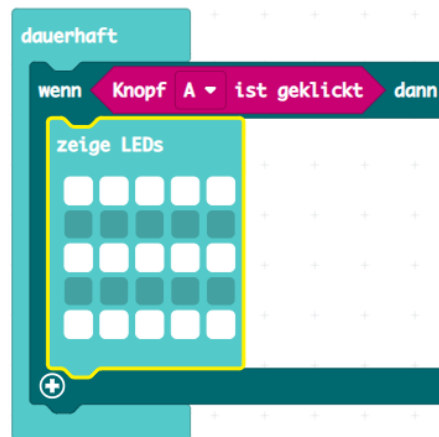
Jetzt ziehen wir uns den Block „Knopf A ist geklickt“ aus dem Bereich Eingabe in den dunkelgrünen Programmblock mit dem Namen „wenn wahr dann“.



Das ist nun unsere Bedingung. Wenn der Knopf A gedrückt wird, dann soll das passieren, was im inneren des dunkelgrünen Programmblocks steht. Bis jetzt ist da aber nichts. Also holen wir uns aus dem Bereich „Grundlagen“ den Block „zeige LEDs“ und setzen ihn in den dunkelgrünen Programmblock.

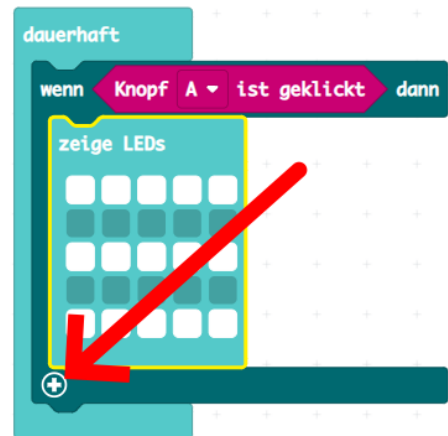


In diesem neuen Block mit dem Namen „Zeige LEDs“ markieren wir nun mit dem Mauszeiger ein Muster. Dieses Muster soll auf dem Calliope erscheinen, wenn der Knopf A gedrückt wird. Unser Muster sieht so aus:

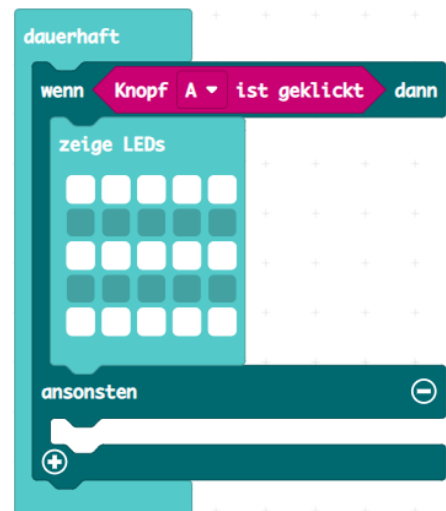


Danach klickst du auf „Herunterladen“ um das Programm auf dem Calliope zu speichern. Das erste Programm ist fertig! Du hast den Calliope so programmiert, dass er ein Muster zeigt, wenn du Knopf A drückst.

Oh aber was ist das? Wenn ich einmal den Knopf gedrückt habe, geht das Muster nicht wieder weg. Ok, wir müssen das Programm so erweitern, dass kein Muster angezeigt wird, wenn die Taste A nicht gedrückt wird. Um den dunkelgrünen „Wenn dann“ Programmblock zu erweitern, klickst du unten links auf das kleine + Zeichen.



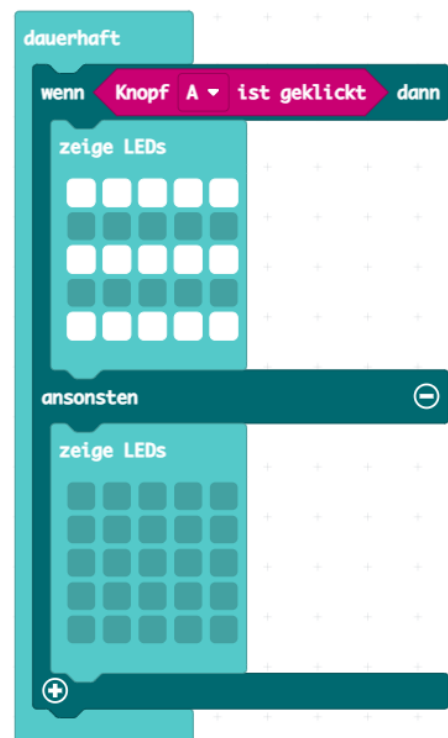
Der Programmblock wird dann um einen Bereich erweitert, der alternativ ausgeführt wird, wenn der Knopf A nicht gedrückt wird.



In diesen neuen Bereich fügen wir aus der Kategorie „Grundlagen“ erneut einen Block mit dem Namen „zeige LEDs“ ein. Nur lassen wir die Matrix nun leer. Erneut laden wir das Programm auf den Calliope, indem wir auf „Herunterladen“ klicken.

Funktioniert dein Programm jetzt so wie es sein soll?

Kannst du das Programm so verändern, dass andere Muster angezeigt werden?



## 3.2 Beispiel für Serielle Kommunikation

In der Einleitung hast du die Serielle Kommunikation kennengelernt. Jetzt möchten wir ein Beispielprogramm schreiben, in dem diese Kommunikation verwendet wird.

Um mit der Seriellen Kommunikation zu arbeiten, klickst du in den Programmblöcken erst auf **Fortgeschritten** und dann auf die Kategorie **Seriell**.



Hier finden wir die Programmblöcke „seriell Zeile ausgeben“ und „seriell umleiten auf USB“.

Den Block „seriell umleiten auf USB“ ziehen wir in den Arbeitsbereich und docken ihn an den grünen Block „beim Start“ an.

Den Block „seriell Zeile ausgeben“ ziehen wir in den Arbeitsbereich und docken ihn an den grünen Block „dauerhaft“ an.



### 3.2.1 Text an den Computer senden

Im ersten Schritt wollen wir vom Calliope das Wort „Hallo“ an den Computer senden.

Im „dauerhaft“-Block klicken wir in den den Programmblock „seriell Zeile ausgeben“ und schreiben „Hallo“ zwischen die Anführungszeichen.

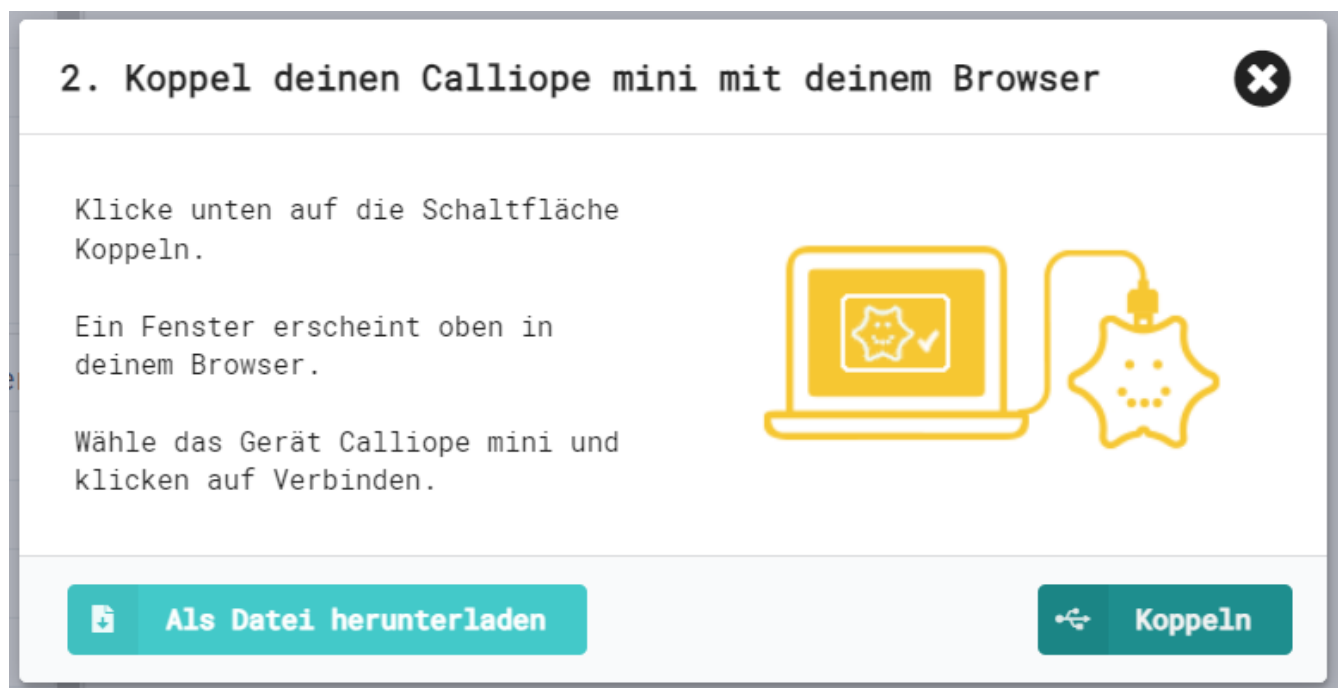


Dieses winzige Programm werden wir nun auf dem Calliope abspeichern. Dieses Abspeichern wird bei der Arbeit mit Mikrocontrollern und Kleincomputern auch „**Herunterladen**“ genannt.

Wenn wir nun unten rechts auf „Herunterladen“ klicken, wird der Code auf dem Calliope gespeichert. Es könnte sein, dass der PC den Calliope nicht automatisch erkennt. In diesem Fall fragt der Computer danach, ob der Calliope verbunden werden soll.



Wir klicken nun auf „Weiter“. Es kommt eine Nachricht, bei der wir auf „Koppeln“ klicken müssen, wenn der Calliope mit dem USB-Kabel am PC angeschlossen wurde.



Nun erscheint ein Browser-Popup, in dem wir den Calliope auswählen können. Wenn wir auf das Gerät klicken, um es auszuwählen, und auf „Verbinden“ drücken, wird er verbunden und wir erhalten eine Popup-Nachricht mit dem Inhalt „Verbunden mit Calliope mini“.



Jetzt musst du nur noch auf „Herunterladen“ klicken, und schon macht der Calliope, was wir ihm mit den Programmblöcken aufgetragen haben.

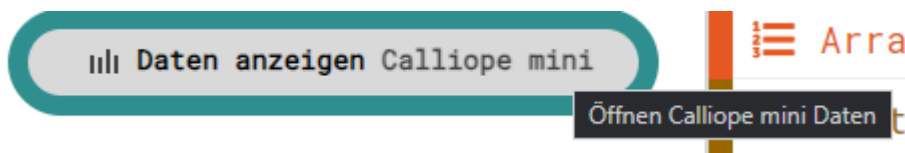
**Das erste Ziel ist erreicht. Das Programm wurde auf den Calliope geladen.**

## 3.2.2 Gesendete Daten am Computer ansehen

Das Programm auf dem Calliope sendet nun ununterbrochen das Wort „Hallo“ an den Computer. Aber wo ist das Wort? Ich kann es nirgendwo sehen!

Jetzt kommt der Baustein „Daten anzeigen“ ins Spiel. Wir erinnern uns, dass der Baustein am Computer unten links unter dem dargestellten Calliope angezeigt wird.

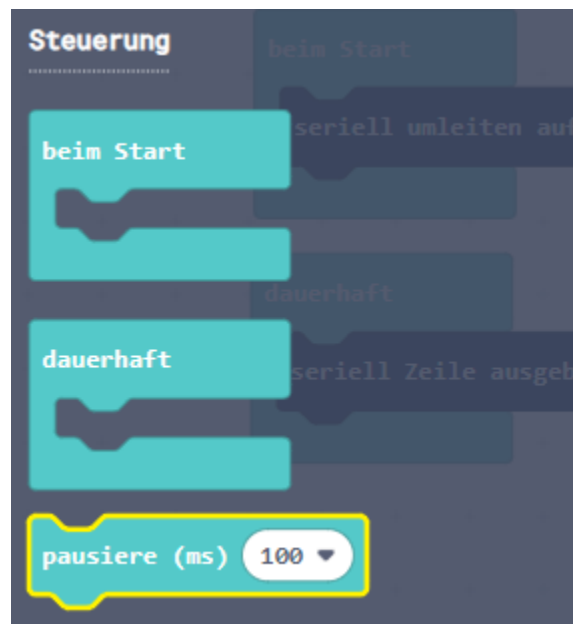
Wenn wir nun auf „**Daten anzeigen Calliope mini**“ drücken, gelangen wir zu den Daten, die vom Calliope mini V3 an den Computer gesendet werden.



Wenn wir „Hallo“ sehen, bedeutet das, dass es funktioniert.

Später wird es wichtig, dass wir das Versenden der Daten an den Computer verlangsamen. Der Calliope ist ein recht schneller Computer, der in einer Sekunde tausende Dinge durchführen kann. Er würde daher auch enorm viele Daten pro Sekunde versenden. Die würden dann so schnell auf dem Seriellen Monitor erscheinen, dass es für uns schwer wird, alles zu erkennen.

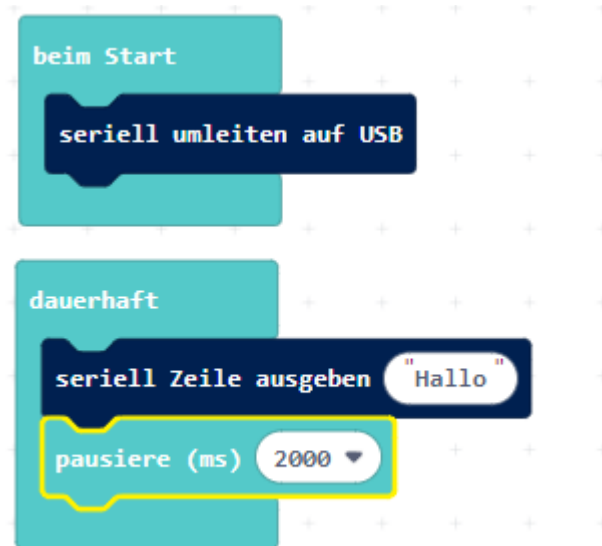
Mit einem kleinen Trick können wir die Menge der Daten die versendet wird reduzieren. Im Editor, also in der Programmieroberfläche, können wir unsere Verlangsamung einführen. Dazu gehen wir in die Kategorie „**Grundlagen**“. Dort finden wir die Option „**pausiere (ms) 100**“, wie im folgenden Bild dargestellt.



Diesen „**pausiere (ms) 100**“ – Block setzen wir nun in den „dauerhaft“ Block hinter den Programmblock „seriell Zeile ausgeben Hallo“.

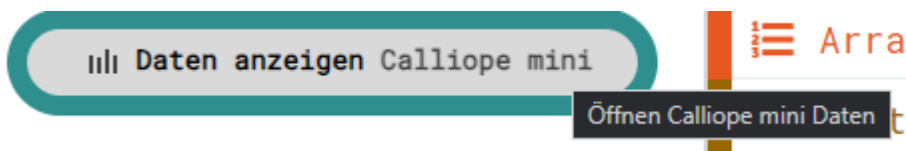


Wenn wir nun auf das Feld im „pausiere“-Block klicken, können wir eine andere Wartezeit einstellen. Wir nehmen in unserem Fall 2 Sekunden. Damit sollten wir genügend Zeit haben, die Ergebnisse anzusehen. Die zwei Sekunden werden jedoch in Millisekunden eingetragen. Daher tippen wir in das Feld 2000.



Das neue Programm möchten wir nun erneut auf den Calliope übertragen. Dazu klicken wir wieder auf „Herunterladen“. Wie das funktioniert, hast du schon kennengelernt.

Im Bereich „Daten anzeigen“ schauen wir uns die Daten nun noch einmal an. Wir erkennen, dass das Wort „Hallo“ in viel größeren Abständen am PC angezeigt wird.



**Geschafft! Du weißt jetzt, wie man Daten vom Calliope an den Computer sendet.** Diese Funktion werden wir später mit echten Messwerten nutzen.

## 4. Praxisaufgaben

Jetzt starten wir mit der Programmierung des Calliope zusammen mit dem Expanderboard und den dafür vorgesehenen Modulen.

Typisch für einen solchen Start ist das Programmieren einer Leuchtdiode. Los gehts!

### 4.1 Eine Leuchtdiode blinken lassen

Eine Leuchtdiode, oft einfach LED genannt, ist ein kleines Licht, das in vielen Geräten zu finden ist. Stell dir eine winzige Lampe vor, die sehr wenig Strom braucht und in vielen Farben leuchten kann.

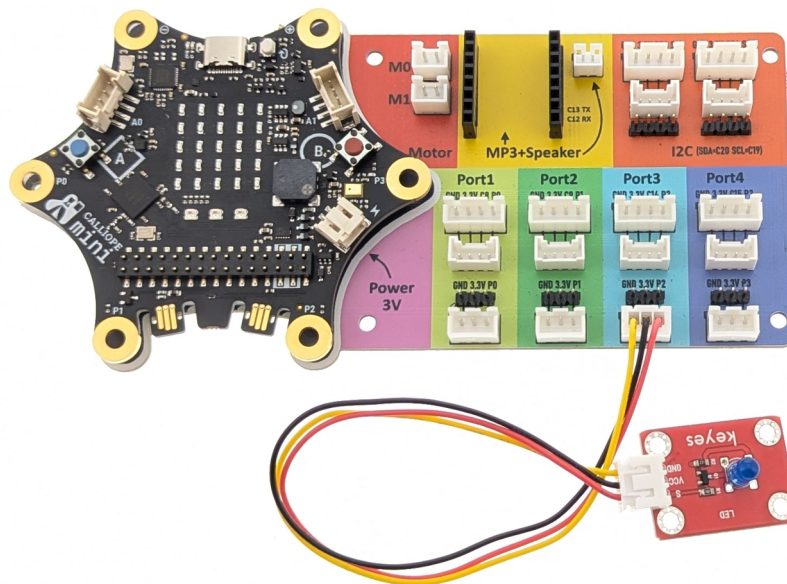
Eine LED funktioniert, indem Strom durch sie fließt, wodurch sie zu leuchten beginnt. Sie ist besonders, weil sie viel weniger Energie als normale Glühlampen oder Leuchtstoffröhren verbraucht und auch nicht heiß wird. Man sieht LEDs oft in Taschenlampen, Lichterketten, Ampeln und sogar in Spielzeugen.

#### Aufgabe

Eine LED soll im Abstand von zwei Sekunden blinken, also an und aus gehen.

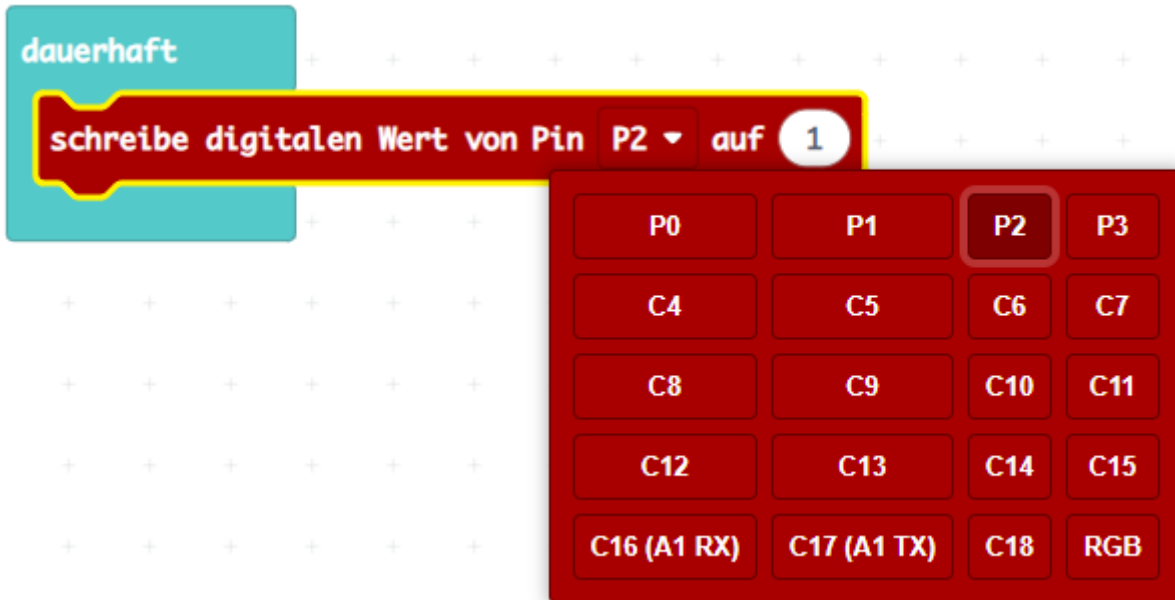
#### Verkabelung

Bei uns ist die LED zusammen mit einer JST-Buchse auf einer kleinen Platine fest gelötet. In diese Buchse stecken wir ein Expanderkabel. Mit der anderen Seite des Expanderkabels verbinden wir die LED-Platine mit dem hellblauen Port 3 am Expanderboard.



## Programm

In der Programmieroberfläche von Makecode suchen wir uns im Bereich **Fortgeschritten** → **Pins** den Programmblock „schreibe digitalen Wert von Pin auf 1“ und heften ihn an den vorhandenen Block „dauerhaft“.



Wie auf dem Bild zu sehen, müssen wir angeben, an welchen Pin die elektrische Spannung gesendet werden soll. In unserem Fall klicken wir auf P2, denn P2 ist der Kontakt des Calliope, der mit dem **Port3** am Expanderboard verbunden ist. Schau dir einmal ganz genau den Steckplatz am Expanderboard an. Du siehst, dass dort die Bezeichnung „P2“ aufgedruckt ist.

Von nun an wird der Calliope also eine elektrische Spannung an den Kontakt P2 senden. Die LED würde dauerhaft leuchten. Das soll sie aber nicht, denn Sie soll blinken.

Daher arbeiten wir nun mit dem Programmblock „**Pause**“ aus dem Bereich „**Grundlagen**“. Eine Pause stoppt das von uns geschriebene Programm um die angegebene Zeit.



Wir programmieren mit diesem Block den folgenden Ablauf:

**LED an – 2000 Millisekunden warten – LED aus – 2000 Millisekunden warten**

Falls du es noch nicht wusstest, 2000 Millisekunden sind das gleiche wie 2 Sekunden. Das fertige Programm sollte bei dir nun folgendermaßen aussehen.



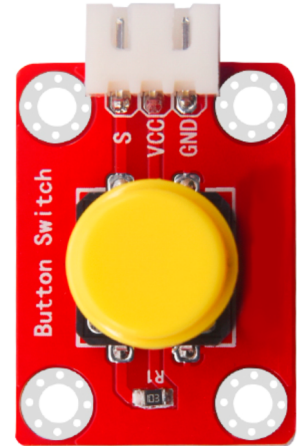
Hast du schon auf Herunterladen geklickt?

## 4.2 Taster verwenden

Ein weiteres typisches Bauteil im Programmierbereich ist der Taster. Ein Taster ist wie ein kleiner Knopf, den man drücken kann, um etwas zu starten oder zu stoppen, zum Beispiel eine Klingel oder ein Licht. Stell dir vor, ein Taster ist wie ein Schalter, aber er funktioniert nur, solange du ihn drückst.

Elektrisch funktioniert ein Taster so: Innen hat er zwei kleine Metallteile, die sich nicht berühren, wenn der Taster nicht gedrückt ist. Wenn du den Taster drückst, bringst du diese Metallteile zusammen, und das schließt den Stromkreis. Das bedeutet, dass der Strom fließen kann, und dadurch wird dann etwas eingeschaltet.

Sobald du den Taster loslässt, trennen sich die Metallteile wieder, der Stromkreis ist unterbrochen, und der Strom hört auf zu fließen.



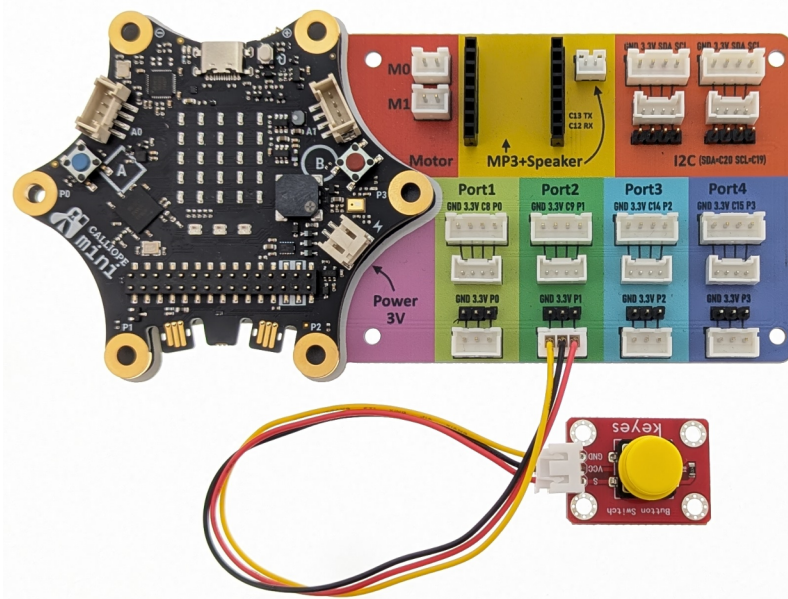
In unserem Fall ist der Taster auf einer kleinen Platine verbaut, die dauerhaft eine Spannung an den Calliope geleitet. Wenn der Taster gedrückt wird, dann wird der weitergeleitete Strom unterbrochen.

### Aufgabe

Wenn ein Taster gedrückt wird, soll auf der Matrix am Calliope ein X erscheinen.

### Verkabelung

Bei uns ist der Taster zusammen mit einer JST-Buchse auf einer kleinen Platine fest gelötet. Mit einem Expanderkabel verbinden wir die Taster-Platine mit dem grünen Port 2 am Expanderboard.



## Programm

Das folgende Programm ist sehr ähnlich wie das Programm aus der Einführung. Dabei wurde noch der Taster auf dem Calliope verwendet. Der Unterschied liegt darin, dass wir das Programm so abwandeln, dass die neue Tasterplatine am grünen Port2 abgefragt wird. Schau dir den Port2 genau an. Erkennst du, welcher Kontakt des Calliope dort verfügbar ist?

Richtig! Es ist der Kontakt P1!

Wir starten mit dem Grundgerüst. An den Block „dauerhaft“ klemmen wir den Block „wenn wahr dann“.

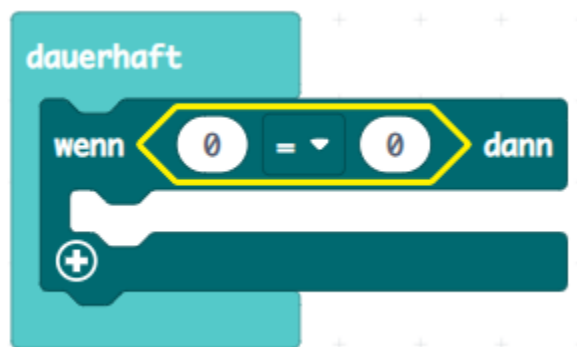


Und schon kommen wir zum ersten Unterschied. Im Vergleich zum Programm in der Einführung. Der Taster am Calliope ist als fester Programmblock hinterlegt. Der externe Taster auf der Platine ist für den Calliope zunächst unbekannt. Aber das ist nicht weiter schlimm, denn wir möchten ja nur prüfen, ob am Calliope eine Spannung ankommt oder nicht.

Daher nehmen wir uns aus dem Bereich Logik einen Vergleichsblock.



Diesen Block setzen wir in unseren „wenn wahr dann“ Block ein.



Diesen Vergleichsblock füllen wir nun mit einem echten Vergleich. Wir möchten wissen, ob an Pin P1 eine Spannung ankommt, oder nicht.

Eine Spannung **Ja** oder **Nein**?

Das ist beim Programmieren ein sogenannter **Boolescher Wert**, der auch als **Wahr (true)** oder **Unwahr (false)** bezeichnet wird.

Der Calliope macht es sich einfach und gibt diese Werte mit **1** und **0** an.

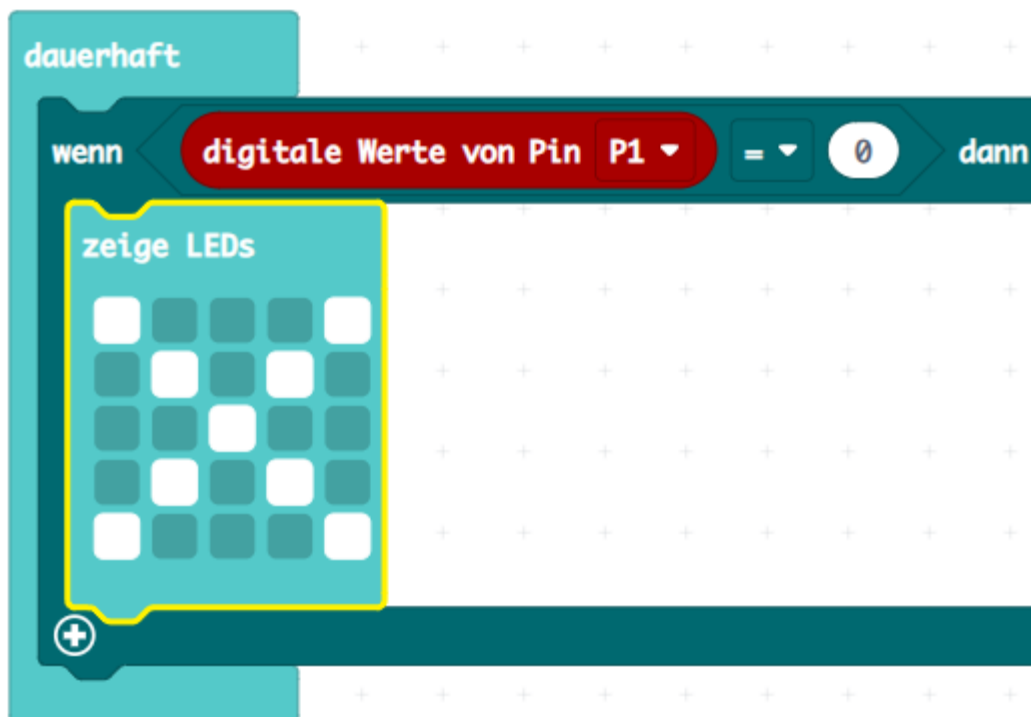
Wir müssen jetzt zuerst prüfen, was am Calliope ankommt, und ob der Wert bei 1 oder 0 liegt. Das machen wir, indem wir folgenden Programmblock einbauen.



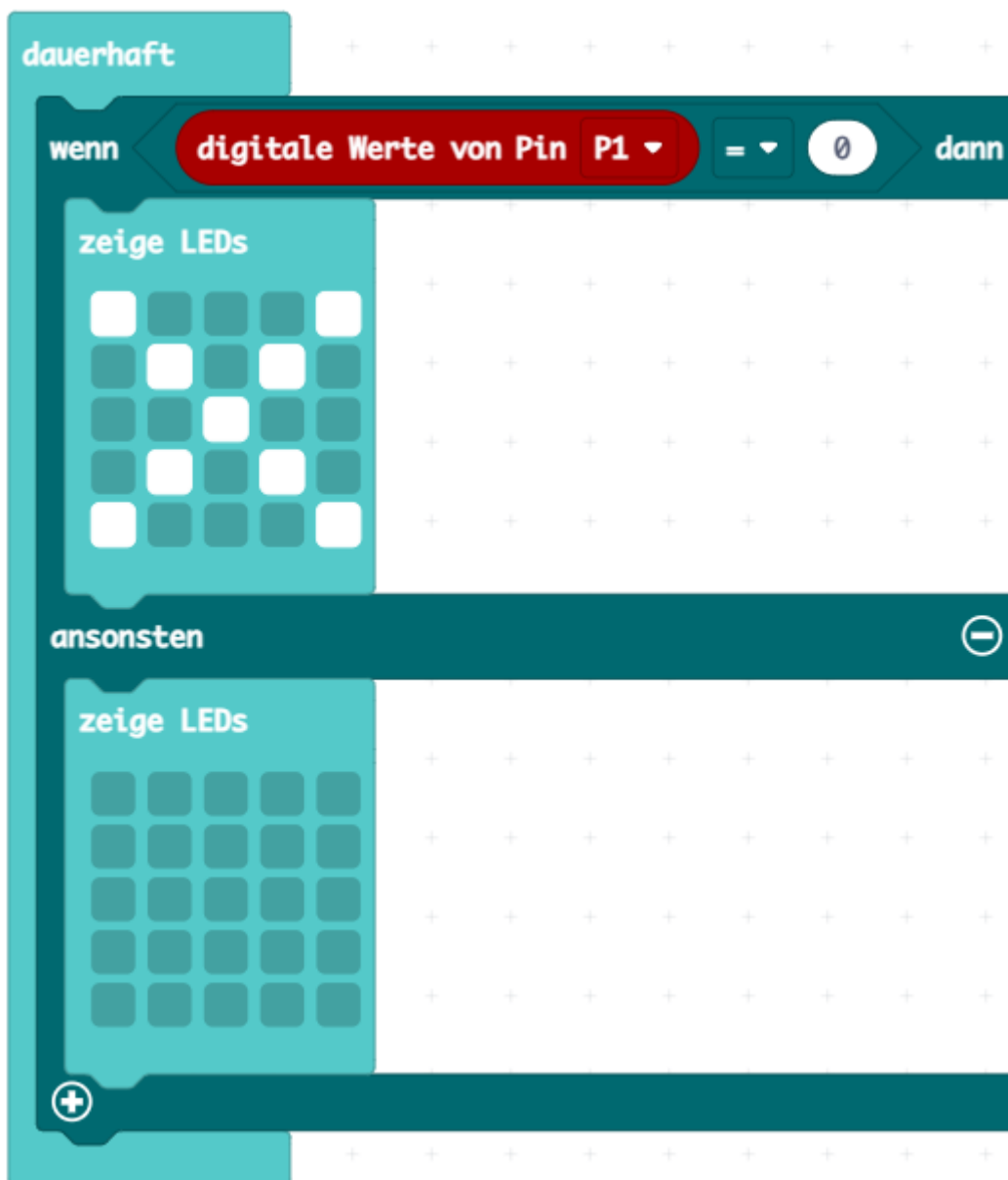
Jetzt wird geprüft, ob der digitale Wert, der an Pin P1 anliegt „0“ ist. Und wenn es so ist, dann soll der nachfolgende Programmblock ausgeführt werden. Aber da ist noch Keiner.

Also machen wir damit weiter, was passieren soll. Es soll laut der Aufgabenstellung ein X auf der LED Matrix erscheinen.

Also nehmen wir aus den Programmblöcken in „Grundlagen“ den Block „Zeige LEDs“ und markieren dort ein X.



Du weißt, was passieren wird! Da die Matrix bis jetzt an keiner Stelle ausgeschaltet wird, müssen wir den Programmblock über das kleine „+“ Zeichen unten am dunkelgrünen Programmblock erweitern und mit einer leeren Matrix füllen. Das fertige Programm sieht folgendermaßen aus.



## 4.3 Drehregler und Serielle Kommunikation

Ein Drehregler ist ein kleines Rad oder Knopf, das man drehen kann, um etwas zu verändern, wie zum Beispiel die Lautstärke bei einem Radio oder die Helligkeit bei einer Lampe. Ein Drehregler kann in vielen elektrischen Geräten vorkommen und wird oft benutzt, um Einstellungen vorzunehmen.

Das richtige Wort für einen elektrischen Drehregler ist **Potentiometer**.

Elektrisch gesehen verändert das Potentiometer den Widerstand im Stromkreis, je nachdem, wie weit der Drehregler bewegt wird. Der Widerstand bestimmt, wie viel Strom durch den Stromkreis fließt. Wenn der Widerstand hoch ist, fließt weniger Strom, und wenn der Widerstand niedrig ist, fließt mehr Strom. Ein Potentiometer funktioniert also ähnlich wie ein Wasserhahn: Wenn der Hahn nur ein bisschen offen ist, fließt wenig Wasser, und wenn er weit geöffnet ist, fließt viel Wasser. Genauso lässt der Widerstand beim Potentiometer mehr oder weniger Strom durch.

Wie verarbeitet der Calliope die Werte eines Potentiometers?

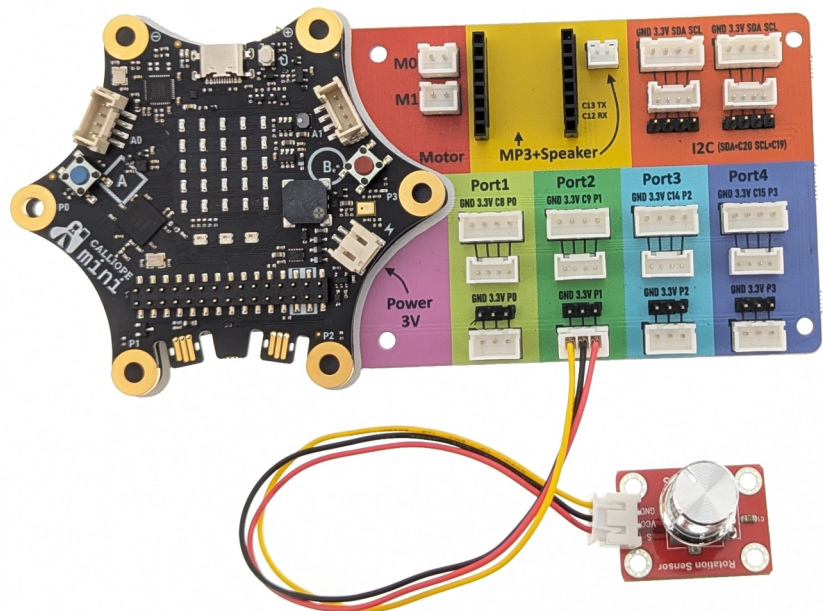
Ein Mikrocontroller, also auch unser Calliope, kann die Stellung des Potentiometers messen. Dafür verwendet der Calliope einen sogenannten Analog-Digital-Wandler (ADC). Der ADC liest die Spannung, die durch das Potentiometer erzeugt wird, und wandelt diese Spannung in eine Zahl um, die der Calliope verstehen kann. Zum Beispiel kann die Spannung zwischen 0 und 3,3V Volt liegen, und der Calliope könnte diese Spannung als Zahlenwert zwischen 0 (für 0 Volt) und 1023 (für 3,3 Volt) darstellen. So kann der Calliope genau erkennen, wie weit das Potentiometer gedreht ist und darauf entsprechend reagieren, z.B. die Lautstärke anpassen oder die Helligkeit ändern.

### Aufgabe

Wenn ein Drehregler gedreht wird, sollen die Messwerte am Seriellen Monitor erscheinen.

### Verkabelung

Bei uns ist der Drehregler zusammen mit einer JST-Buchse auf einer kleinen Platine fest gelötet. Mit einem Expanderkabel verbinden wir die Drehregler-Platine mit dem grünen Port2 am Expanderboard.



## Programm

Falls du dir in der Einführung noch nicht den Bereich mit der Seriellen Kommunikation angesehen haben, solltest du es nun nachholen. Denn wir werden nun die Sensorwerte des Drehreglers auswerten und die Messwerte in der Seriellen Kommunikation ansehen.

Warum ist es notwendig, die Messwerte zu sehen?

Nur wenn wir die Messwerte eines Sensors kennen, können wir später in Abhängigkeit der Werte eine neue Aktion auslösen.

Dazu ein konkretes Beispiel:

Sobald ein Drehregler um eine viertel Umdrehung gedreht wurde, soll auf der LED-Matrix am Calliope ein Herz zu sehen sein.

Aber welcher Sensorwert, also welchen Zahlenwert erkennt der Calliope bei einer Viertel Umdrehung? Das werden wir wissen, wenn wir die Messwerte auslesen und uns anzeigen lassen.

Los gehts!

Wir starten mit den ersten Blöcken und besorgen uns den Block „seriell umleiten auf USB“. Diesen klemmen wir an den Block „beim Start“.



Dann klemmen wir den Block für die Ausgabe von Daten an die Serielle Kommunikation an den „dauerhaft“ Block.

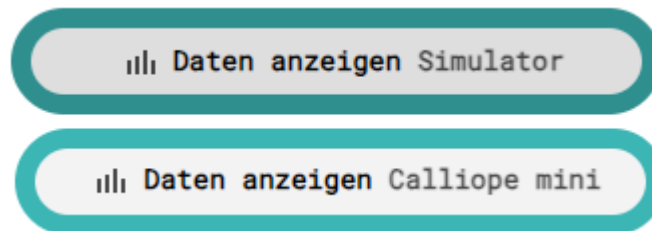


Innerhalb des neuen Block wird angegeben, welche Daten an die Serielle Kommunikation gesendet werden sollen. In unserem Fall sollen hier die analogen Messwerte des Calliope-Kontakts P1 aufgeführt werden. Also ergänzen wir den Block „seriell Zeile ausgeben“ um den Block „analoge Werte von Pin P1“.



Das war es schon! Der Code ist fertig und wir laden den Code mit dem „Herunterladen“ Button auf den Calliope.

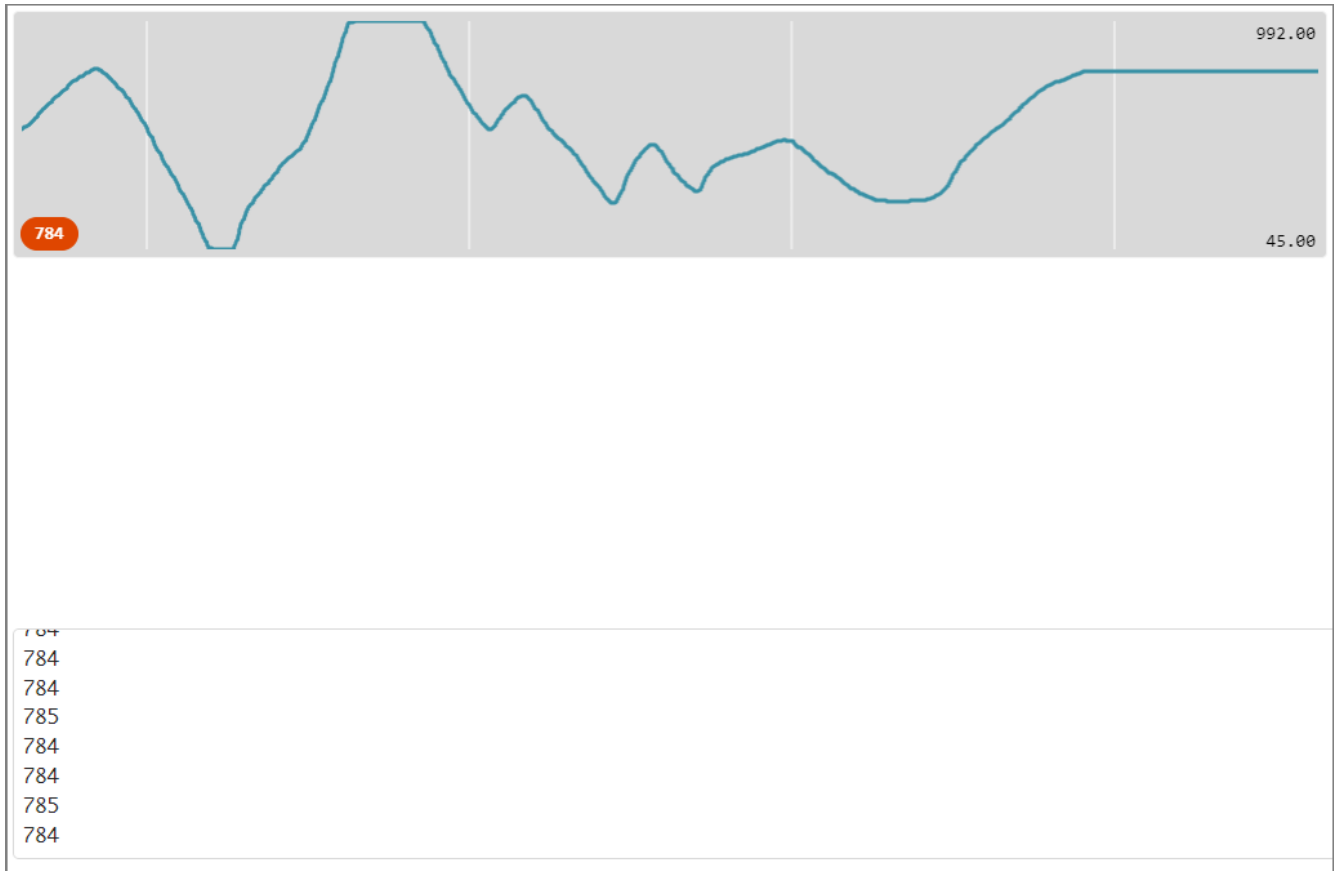
Jetzt können wir uns die Messwerte des Drehreglers ansehen, indem wir auf die Schaltfläche „Daten anzeigen Calliope Mini“ klicken.



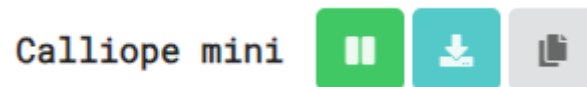
Es öffnet sich dann unmittelbar ein neues Fenster.

#### Hier können wir einiges erkennen:

1. Die orange dargestellte Zahl, im folgenden Bild die 784, zeigt den aktuellen Messwert.
2. Oben erscheint ein grauer Balken, indem ein Diagramm aufgezeichnet wird. Dieses Diagramm stellt die Messwerte grafisch dar.
3. Rechts im grauen Balken steht oben und unten jeweils eine Zahl. Diese beiden Zahlen zeigen den höchsten und niedrigsten Messwert der aktuellen Messperiode an. In dem folgenden Bild sind das die Zahlenwerte 992 und 45.
4. Ganz unten erscheint eine Tabelle mit allen Messwerten, die hier tabellarisch aufgelistet werden. Man könnte die Messwerte hier herauskopieren und beispielsweise in Microsoft Excel oder in LibreOffice Calc weiter bearbeiten, auswerten oder als Diagramm darstellen.



Auch das Abspeichern der Messwerte in Form einer CSV-Datei ist auf dieser Seite über die folgenden Schaltflächen möglich.



Nun drehen wir den Drehregler hin und her und betrachten die Messwerte. Wir stellen fest, dass der kleinste Messwert bei ca. 50 liegt und der größte Messwert bei ca. 990.

Kommen wir zurück zur Aufgabe! Wenn wir den Drehregler ganz nach links drehen, wird ein Wert von ca. 50 angezeigt. Nun drehen wir den Drehregler um eine Vierteldrehung nach Rechts. Es wird ein Messwert von ca. 260 angezeigt.

Das bedeutet für die Lösung unserer Aufgabe, dass wir eine Bedingung einbauen müssen. Diese Bedingung sagt:

„Wenn der Messwert über 260 ist, soll in der LED-Matrix ein Herz erscheinen.“

## 4.4 Drehregler verwenden

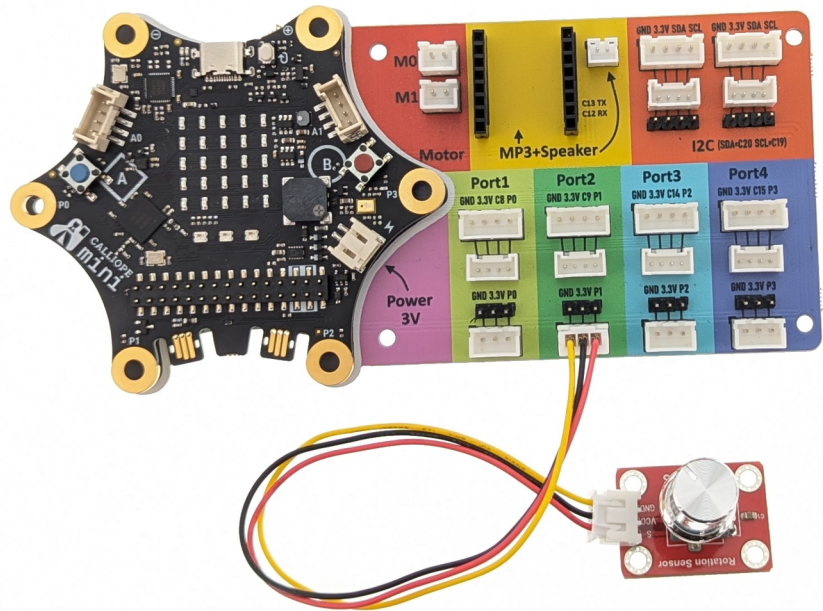
Jetzt möchten wir den Drehregler verwenden, um auf der LED-Matrix des Calliope verschiedene Symbole anzuzeigen. Wir wissen schon, dass die Messwerte des Drehreglers im Zahlenbereich zwischen ca. 50 und 990 liegen. Wie können wir nun in Abhängigkeit des Drehkopfes verschiedene Symbole anzeigen?

### Aufgabe

Wenn der Drehregler links ist, soll ein ein Pfeil nach Links auf der LED-Matrix erscheinen. Wenn der Drehregler rechts ist, soll ein Pfeil nach Rechts auf der LED-Matrix erscheinen.

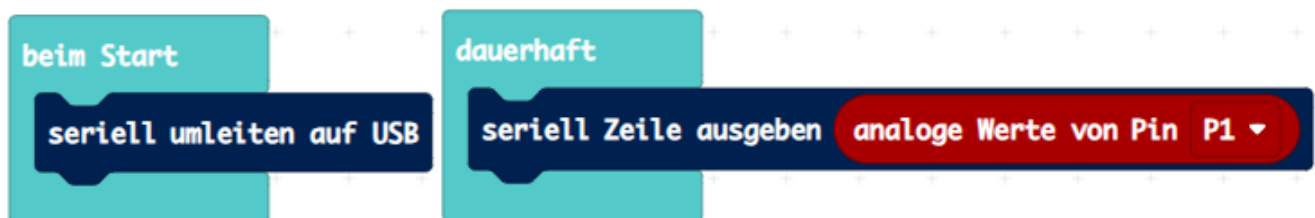
### Verkabelung

Bei uns ist der Drehregler zusammen mit einer JST-Buchse auf einer kleinen Platine fest gelötet. Mit einem Expanderkabel verbinden wir die Drehregler-Platine mit dem grünen Port2 am Expanderboard.



### Programm

Wir beginnen mit dem fertigen Programm aus der vorherigen Anleitung. Das sieht so aus:

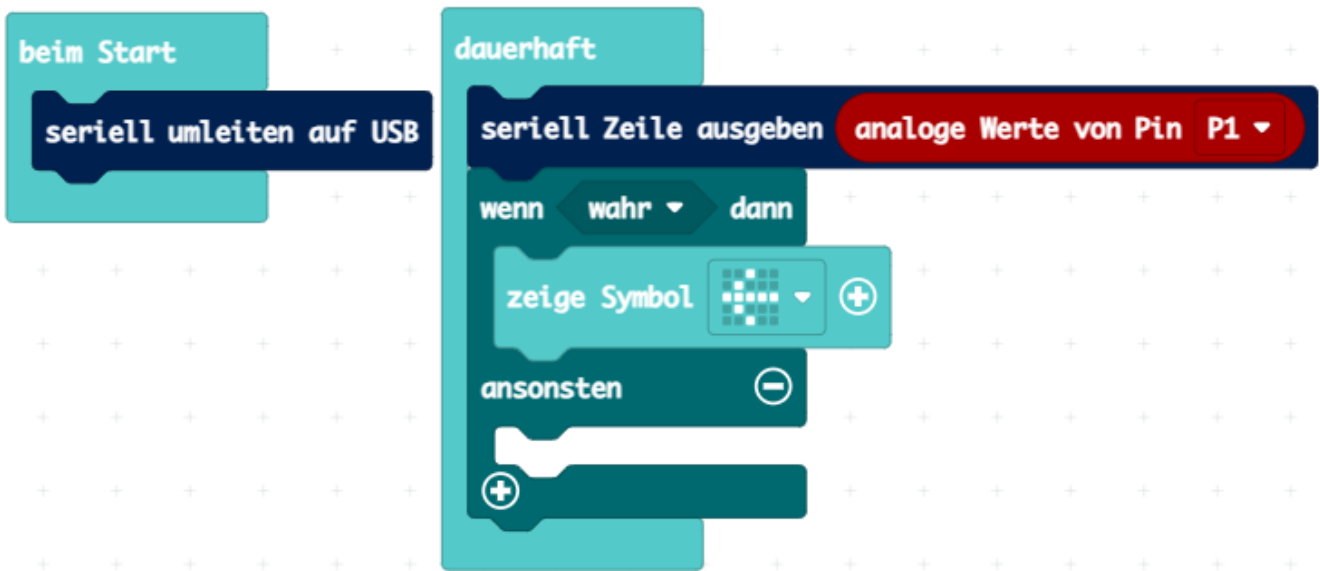


Jetzt ergänzen wir die Bedingung für die Symbole, die erscheinen sollen.

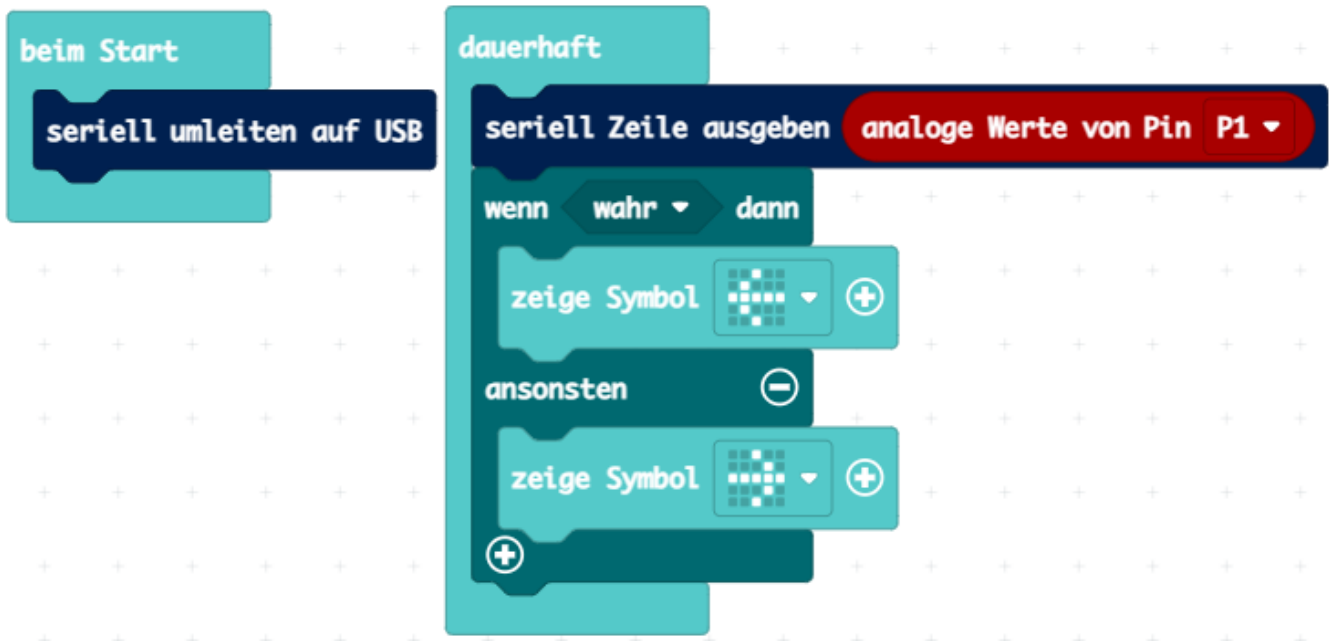
Dazu verwenden wir den dunkelgrünen Baustein „wenn wahr dann“.



Komm, wir tragen schon mal ein, was passieren soll! Wenn diese Sache mit dem Drehregler auf der linken Seite wahr ist, soll der Pfeil nach links auftauchen. Also nehmen wir aus den „Grundlagen“ den Programmblock „zeige Symbol“ und wählen dort den Pfeil nach links aus.



Außerdem weiß man schon, was angezeigt werden soll, wenn der Pfeil nichts links ist... nämlich rechts! Also ergänzen wir auch schon den Pfeil nach rechts, der „ansonsten“ angezeigt werden soll.

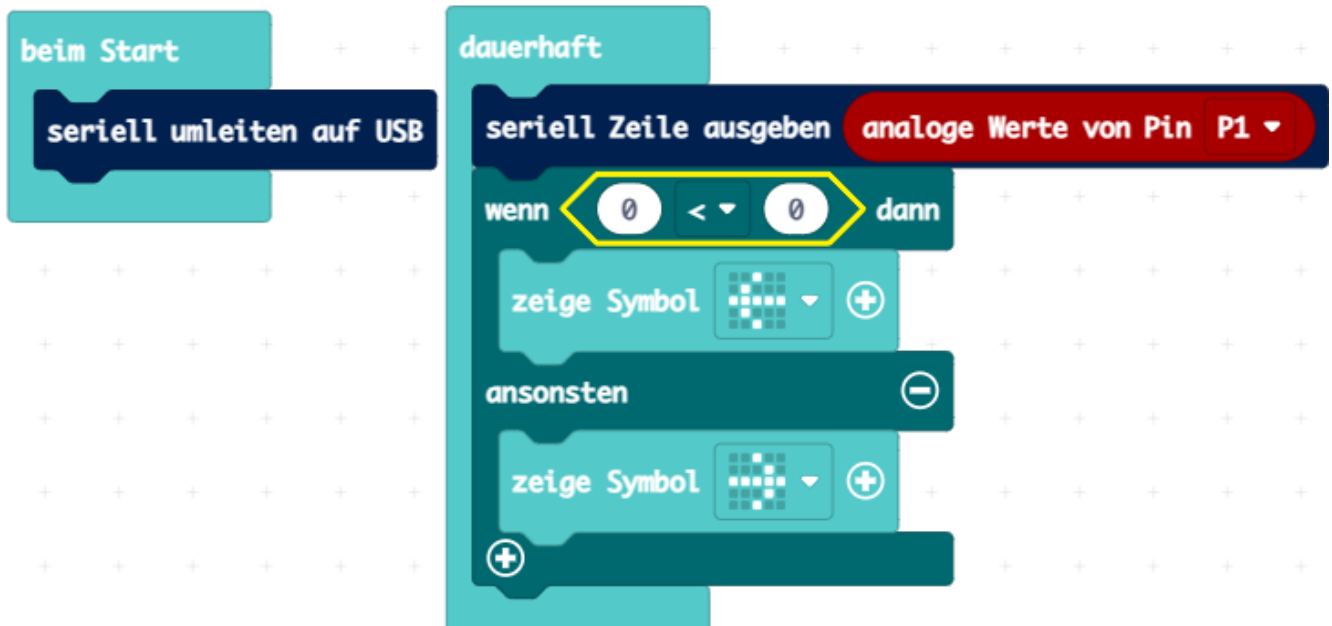


Ok, das war einfach. Aber wie sieht nun die Bedingung oben im dunkelgrünen Feld aus? Was soll wahr sein, damit der Pfeil nach links erscheint?

Denken wir noch einmal über die Messwerte des Drehreglers nach. Diese liegen zwischen 50 und 960. Welche Zahl ist denn genau in der Mitte? Das können wir ein bisschen runden. Eine Zahl zwischen 50 und 960 ist doch so ähnlich wie eine Zahl zwischen 0 und 1000. Das ist jetzt einfach... dazwischen liegt die Zahl 500.

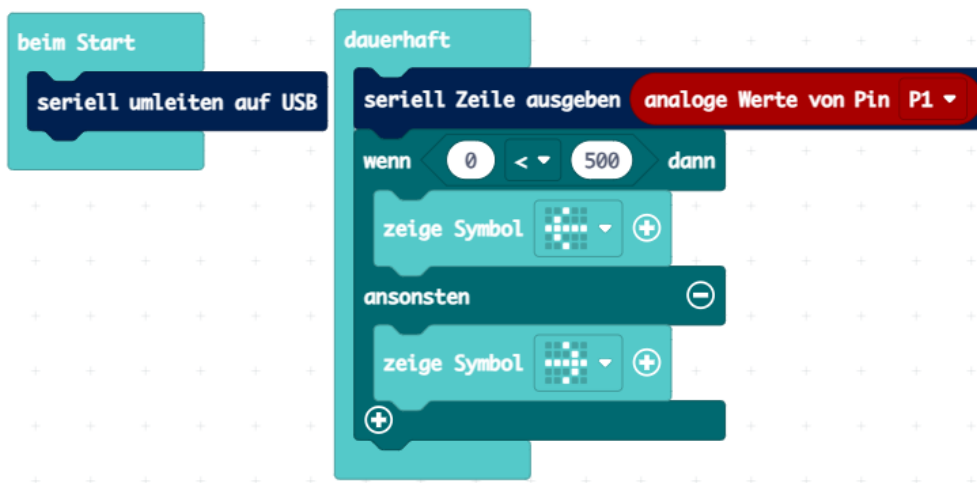
Also schreiben wir in die Bedingung: **„Wenn der Messwert unter 500 ist, soll der Pfeil nach Links angezeigt werden“**.

Wir ziehen uns aus dem Programmbereich „Logik“ den Vergleich **“0 < 0“**. Mit diesem Block kann man größer und kleiner vergleichen. Den Block setzen wir oben in den dunkelgrünen Block **„Wenn wahr dann“**.

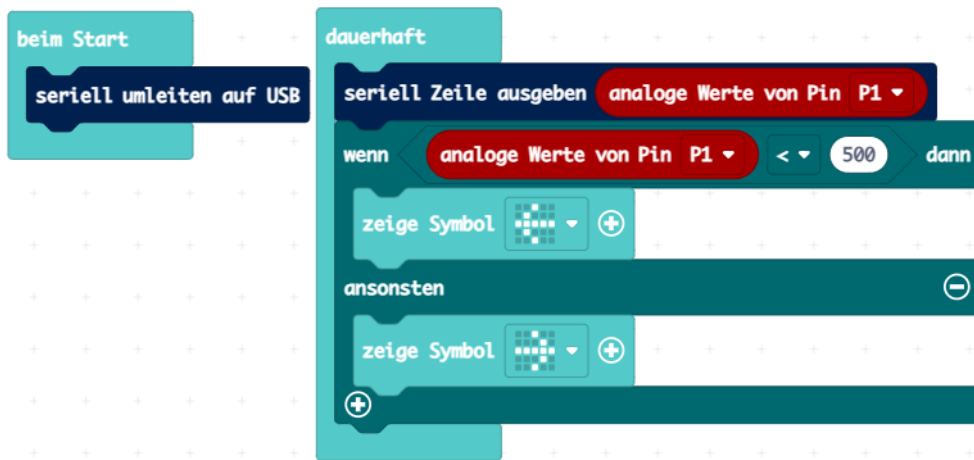


Jetzt müssen wir nur noch angeben, was kleiner als was sein soll. Weißt du es?

Richtig, der Messwerte soll kleiner als 500 sein, um den Pfeil nach links zu aktivieren. Also tippen wir in dem Vergleichsfeld rechts schon die 500 ein.



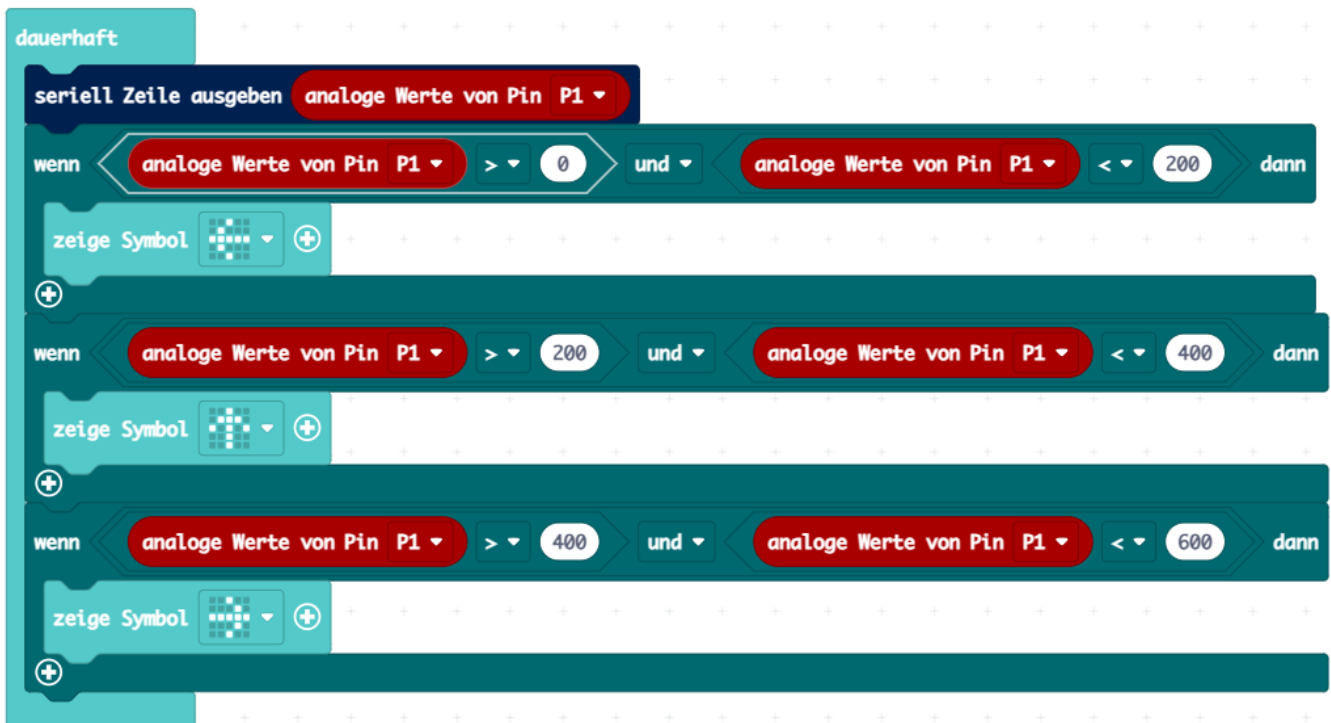
Im linken Bereich des Vergleichsfeldes muss der Messwert eingetragen werden. Dazu verwenden wir den gleichen Block, den wir auch schon oben bei der Ausgabe der Seriellen Kommunikation verwenden. Der Block heißt „**analoge Werte von Pin P1**“.



Herunterladen! Und? ... es funktioniert!

Schaffst du es, viele verschiedene Symbole beim Drehen am Drehrad erscheinen zu lassen? Dazu kann es sinnvoll sein, mit dem „und“ Programmblock aus dem Bereich „Logik“ zu arbeiten.

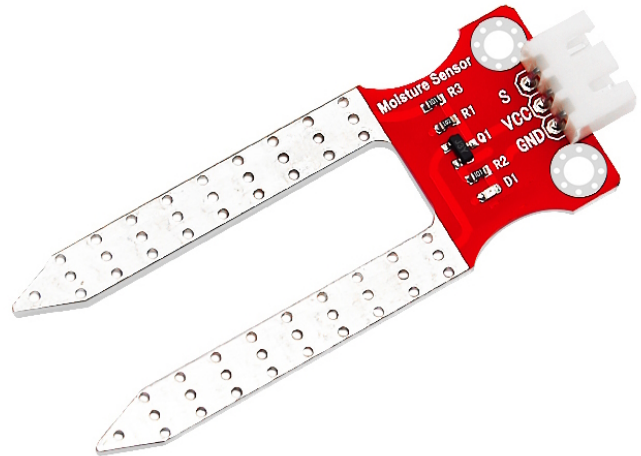
Hier ist ein kleiner Tipp:



## 4.5 Feuchtigkeitssensor

Der Feuchtigkeitssensor gehört wie der Drehregler zu den analogen Sensoren.

Ein Feuchtigkeitssensor ist ein kleines elektrisches Bauteil, das messen kann, wie hoch die Feuchtigkeit in der Umgebung ist. Dieser Sensor misst jedoch nicht die Luftfeuchtigkeit, sondern die Feuchtigkeit, die sich genau am Sensor befindet. Zum Beispiel die Feuchtigkeit der Erde in einem Blumentopf.



Damit eine Pflanze gut wachsen kann, muss die Erde feucht sein, aber nicht zu nass oder zu trocken. Ein Feuchtigkeitssensor hilft uns herauszufinden, ob die Erde genug Wasser hat.

### Wie funktioniert ein Feuchtigkeitssensor?

Der Sensor hat zwei kleine Stifte aus Metall, die du in die Erde stecken kannst. Diese Teile messen, wie gut Strom durch die Erde fließen kann.

### Wasser leitet Strom

Wenn die Erde feucht ist, also Wasser enthält, kann Strom besser fließen. Das liegt daran, dass Wasser Strom leiten kann. Je mehr Wasser in der Erde ist, desto besser fließt der Strom.

### Messung der Leitfähigkeit

Der Sensor misst, wie gut der Strom fließt, und gibt diese Information als Zahl an den Calliope Mikrocontroller weiter, genau so, wie es in der Anleitung zum Drehregler der Fall ist. Diese Zahl kann uns sagen, ob die Erde trocken, feucht oder sehr nass ist.

### Der Calliope zeigt das Ergebnis an

In der Seriellen Kommunikation kannst du dir die Messwerte anzeigen. Mit den Werten kannst du dann über passende Programmblöcke eine Aktion auslösen. Zum Beispiel eine LED einschalten oder eine Pumpe aktivieren, wenn die Erde zu trocken ist.

Mit einem Feuchtigkeitssensor kannst du also ganz einfach überprüfen, ob deine Pflanzen genug Wasser haben, ohne den Finger in die Erde zu stecken!

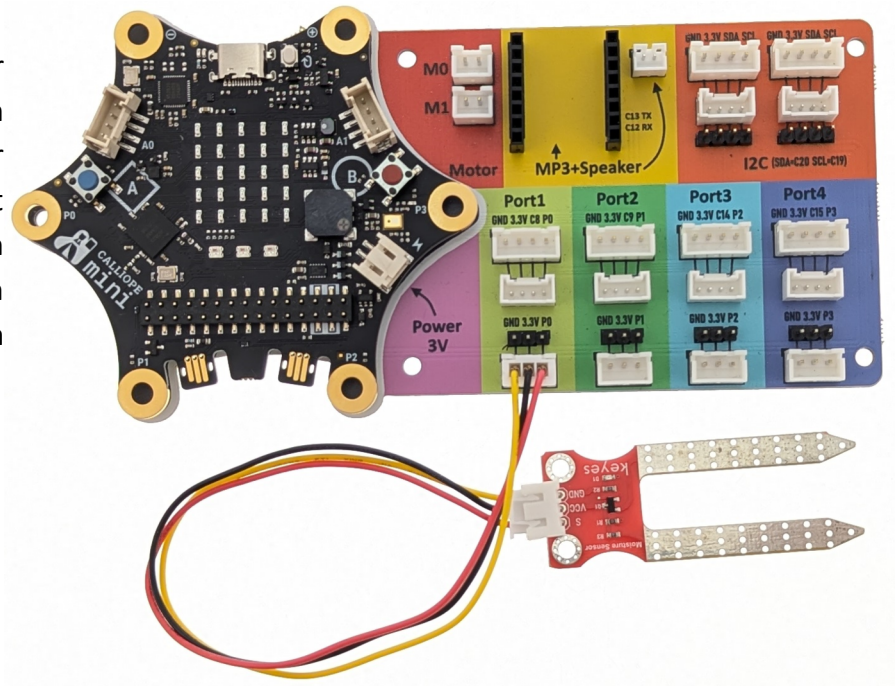
## Aufgabe

Teil1: Sieh dir die Messwerte des Feuchtigkeitssensor an bei Trockenheit und bei Feuchtigkeit. Um feuchten Boden zu simulieren, kannst du den Sensor mit den Kontaktstiften auf ein feuchtes Taschentuch drücken.

Teil2: ALARM! Wenn die Feuchtigkeit zu gering ist, soll ein Alarmton erklingen und auf der LED-Matrix des Calliope soll ein X erscheinen.

## Verkabelung

Bei uns ist der Feuchtigkeitssensor zusammen mit einer JST-Buchse auf einer kleinen Platine fest gelötet. Mit einem Expanderkabel verbinden wir die Sensorplatine mit dem **hellgrünen Port1** am Expanderboard.



## Aufgabenteil 1 – Messwerte anzeigen

Um den Feuchtigkeitssensor zu programmieren, verwenden wir als Grundlage das Programm aus der Anleitung mit dem Drehregler. Das sieht so aus:



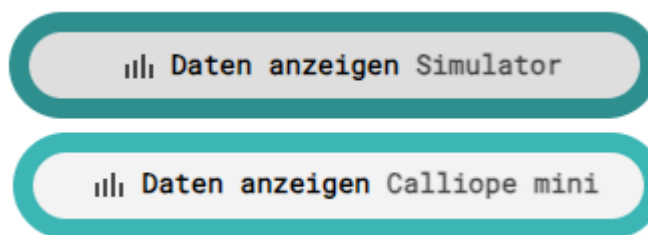
**Aber Vorsicht!** In der Anleitung haben wir noch einen anderen Steckplatz benutzt. Jetzt verwenden wir auf dem Expanderboard den **hellgrünen Steckplatz Port1**. Der dazugehörige Kontakt des Calliope hat den Namen „**P0**“.

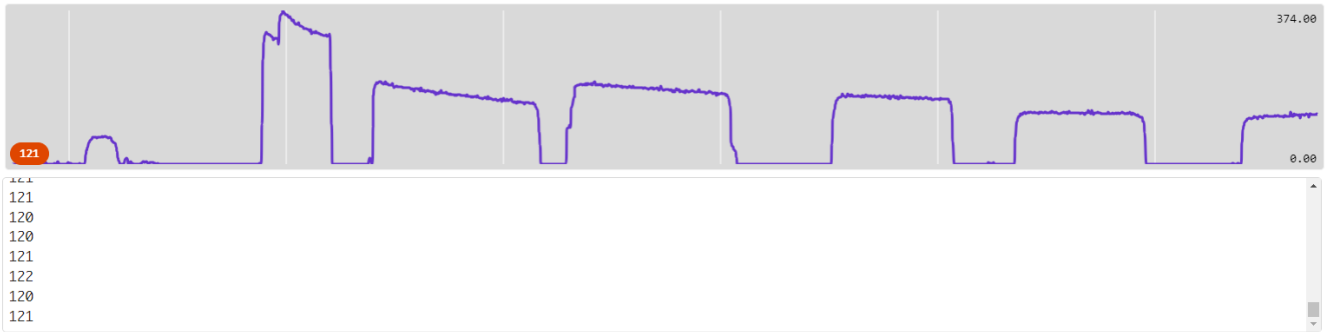


Nun klicken wir direkt wieder auch „**Herunterladen**“ und sehen uns danach die Messwerte in der Seriellen Kommunikation an. Dazu klicken wir auf „**Daten anzeigen Calliope mini**“.

### Ergebnis

Die Messwerte liegen in unseren Versuchen zwischen 0 und 374. Das kann man auf dem folgenden Bild gut erkennen.





Wenn der Sensor in der Luft ist, wird keine Feuchtigkeit erkannt und der Messwert liegt bei 0.  
 Wenn man den Sensor auf ein feuchtes Taschentuch drückt, dann liegt der Messwert in unserem Versuch bei ca. 120.

## Aufgabenteil 2 – Messwerte verwenden

Jetzt werden wir wieder eine Abhängigkeit programmieren. Wenn die Feuchtigkeit zu gering ist, soll ein Alarmton erklingen und auf der LED-Matrix des Calliope soll ein X erscheinen.  
 In diesem Beispiel wird die Bedingung also so formuliert:

**„Wenn der Messwert unter 50 ist, dann soll auf der Matrix ein X erscheinen und ein Alarmton erklingen“**

Dazu erweitern wir nun den vorhandenen Code mit dem bekannten Codeblock „wenn wahr dann ansonsten“ .



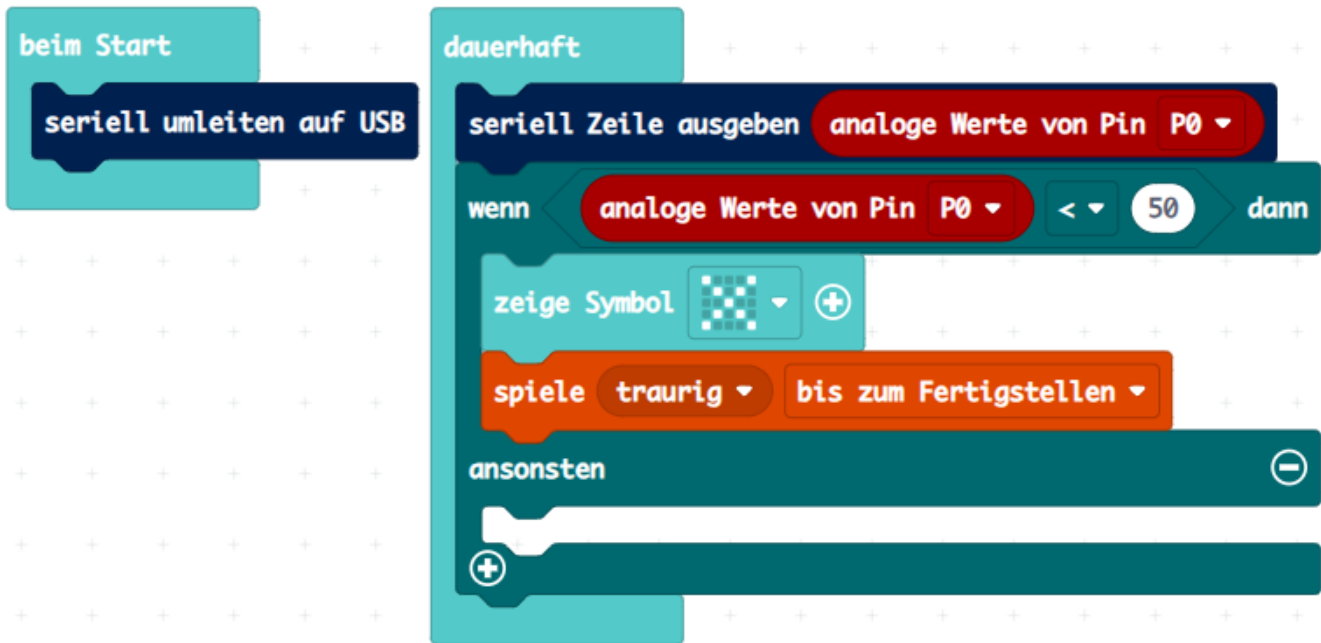
Was soll in diesem Fall „wahr“ sein? Hier müssen wir wieder einen Vergleich anstellen zwischen dem Messwert und dem Wert, der nicht unterschritten werden darf. Also lautet der erforderliche Programmblock „Kleiner als“ ( $0 < 0$ ).



Die beiden Werte, die hier verglichen werden sollen, sind der Messwert und die Zahl 50. Also ergänzen wir hier die beiden Elemente.



Fast geschafft. Wir müssen noch eintragen, was passieren soll, wenn der Messwert unter 50 ist. Dazu nehmen wir aus der Rubrik „Grundlagen“ den Programmblock „zeige Symbol“ und wählen hier das X aus. Zusätzlich klicken wir auf die Rubrik „Musik“ und anschließend auf die darunterliegenden drei Punkte mit dem Namen „mehr“. Hier wählen wir den Programmblock „spiele kichern bis zum Fertigstellen“. Wir klicken dann noch auf „kichern“ und wählen stattdessen eine traurige Melodie.



Geschafft. Alles funktioniert. Aber Moment, das X auf der LED-Matrix geht nicht weg. Weißt du, was noch fehlt?

Tipp: Die Lösung findest du in der Anleitung zum Taster.

## 4.6 Elektromotor ansteuern

Ein Motor ist ein elektrisches Bauteil, das Dinge in Bewegung bringt, zum Beispiel Räder an einem Auto oder den Propeller von einem Ventilator. Aber wie macht der Motor das? Dafür benutzt er die Kraft von Magneten!

### Magnete und Strom

Ein Motor benutzt Magnete und Strom, um sich zu drehen. Magnete haben zwei magnetische Pole, den Nord- und den Südpol. Gegensätzliche Seiten (Nord und Süd) ziehen sich an, und gleiche Seiten (Nord und Nord oder Süd und Süd) stoßen sich ab.

### Elektromagnet

Im Elektromotor gibt es einen speziellen Magneten, einen Elektromagneten. Das bedeutet, dass er nur dann zu einem Magneten wird, wenn Strom durch ihn fließt. Ein Elektromagnet kann sogar seine Nord- und Südseiten ändern, wenn der Strom in die andere Richtung fließt.

### Drehen durch Anziehen und Abstoßen

Der Elektromagnet im Motor wird immer wieder ein- und ausgeschaltet, und seine Nord- und Südseiten wechseln ständig. Dadurch wird er im richtigen Takt von festen Magneten im Motor angezogen und abgestoßen. Diese Anziehung und Abstoßung bringt den Elektromagneten dazu, sich zu drehen.

### Bewegung übertragen

Wenn der Elektromagnet sich dreht, dreht sich auch der Propeller, der an ihm befestigt ist. Diese drehende Bewegung wird verwendet um ein bisschen Wind zu erzeugen.

### Aufgabe

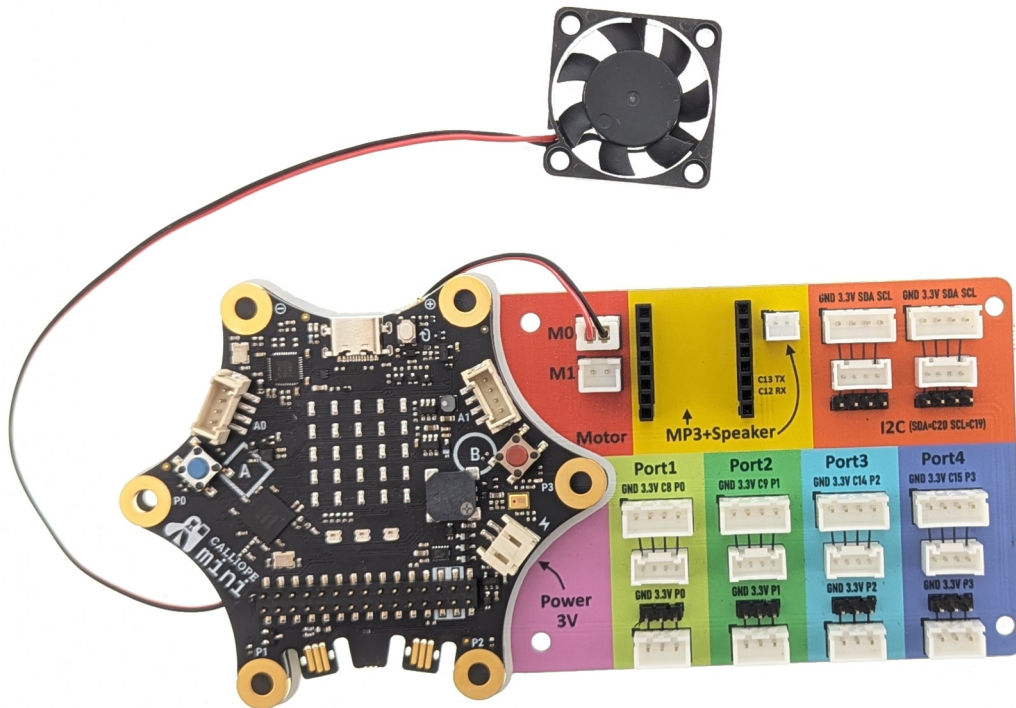
Teil1: Eine kleiner Ventilator soll Wind erzeugen

Teil2: Eine kleiner Ventilator soll nur dann Wind erzeugen, wenn der Temperatursensor am Calliope mit dem Daumen erwärmt wurde.

### Verkabelung

Der Elektromotor des Lüfters verfügt bereits über einen sogenannten JST XH Stecker. Auf dem Expanderboard findest du einen mit roter Farbe markierten Bereich. Dieser Motorbereich verfügt über zwei Steckplätze, **M0** und **M1**.

Wir verbinden den Motor mit dem **Steckplatz M0**.



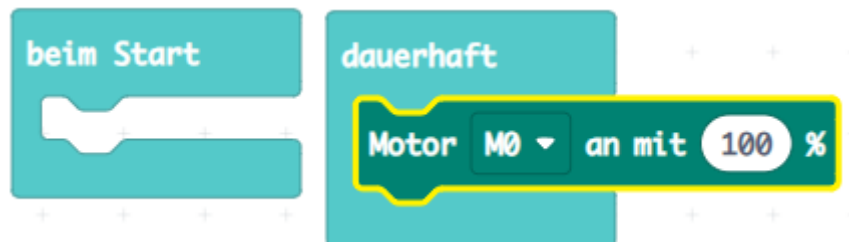
## Aufgabenteil 1 – Wind erzeugen

Erzeuge etwas Wind mit dem Ventilator

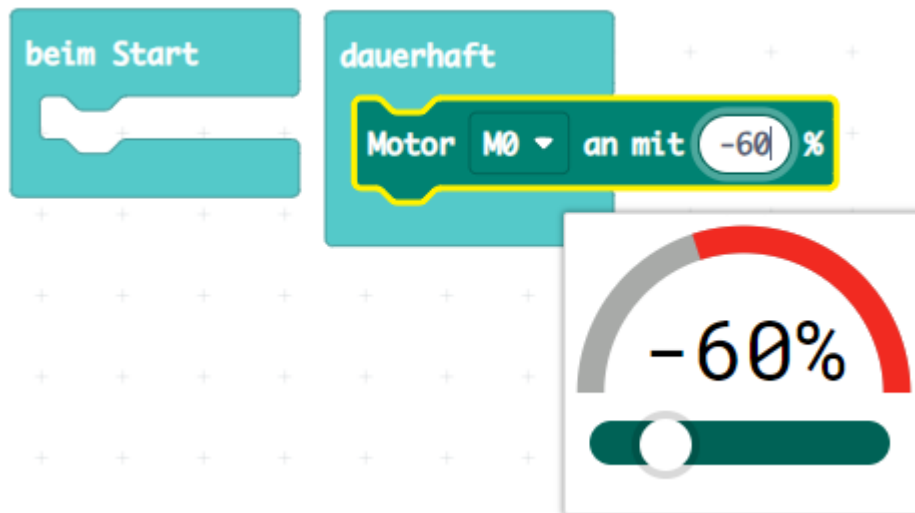
Wir starten mit der üblichen Programmoberfläche.



Aus den Programmblöcken holen wir uns aus der Rubrik „Motoren“ den Block „Motor M0 an mit 100%.“



Das war es schon fast. In dem Programmblock zum Motor steht nun „100%“. Der Motor würde also mit 100% Leistung angesteuert werden. Aber in welche Richtung? **Linksherum** oder **rechtsherum**? Dafür kann man in dem Block zur Motorsteuerung **-100%** und **+100%** auswählen. Klick dazu auf die Zahl und verwende dann den Schieberegler, so wie im folgenden Bild.



Einige Motoren, so wie unser Lüfter, können sich nur in eine Richtung drehen. Finde heraus, ob dein Motor mit **+100%** oder mit **-100%** bewegt werden kann.

Nach einem Klick auf „Herunterladen“ hast du wieder ein Ziel erreicht. Aufgabenteil 1 ist erledigt.

## Aufgabenteil 2 – Kühlung automatisieren

Nun soll der Lüfter nur dann drehen, wenn der Calliope wärmer wird. Das ist eine neue Herausforderung, aber das können wir schaffen!

Erneut starten wir als Grundlage mit dem Anzeigen der Sensorwerte in der Seriellen Kommunikation. Die Codeblöcke solltest du bereits auswendig finden können. Aus den Programmblöcken im Bereich „Eingabe“ setzen wir nur noch schnell die Temperaturwerte des Temperatursensors ein, der sich auf dem Calliope befindet. Der Temperatursensor befindet sich in dem schwarzen quadratischen Bauteil neben der Taste „A“.



Nach dem Herunterladen klicken wir auf „Daten anzeigen Calliope mini“ und sehen die Temperatur des Calliope. Manchmal ist der Calliope etwas wärmer als die Umgebungstemperatur, da zusammen mit dem elektrischen Strom auch immer ein bisschen Wärme entsteht.

In unserem Beispiel wird bei 22° Raumtemperatur eine Gerätetemperatur von 27°C angezeigt. Wenn wir nun auf den Temperatursensor unseren Daumen legen, steigt die Temperatur an. In unserem Fall auf 31°C. In der Seriellen Kommunikation sieht das so aus:



Welche Temperatur wird bei dir angezeigt? Wie ändert sich die Temperatur wenn du deinen Daumen auf den Sensor legst?

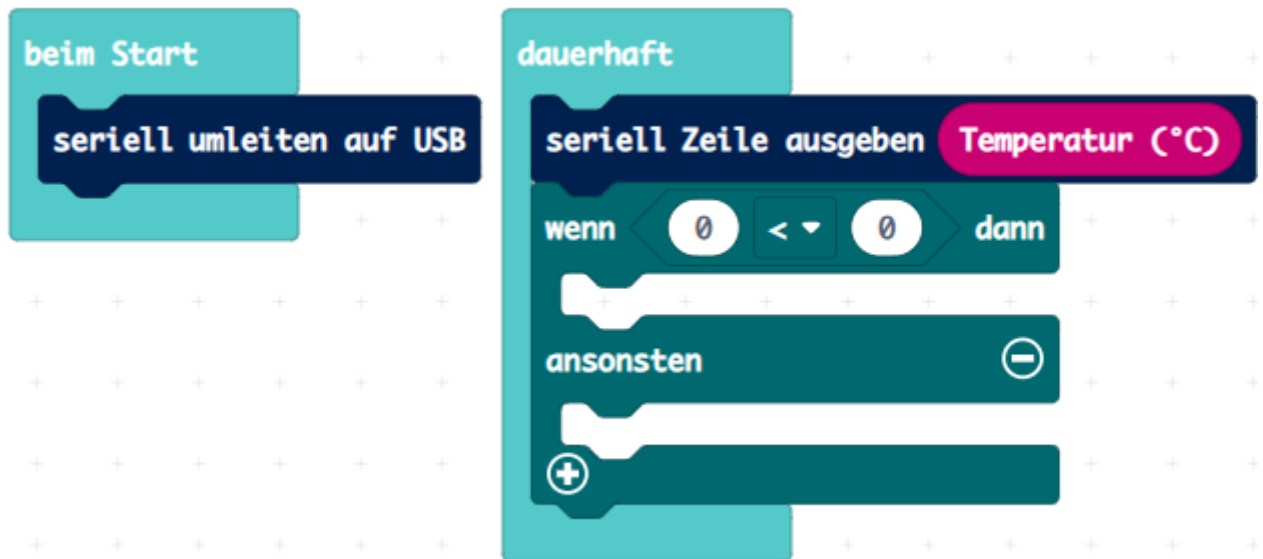
Wir möchten nun erreichen, dass der Lüfter an geht, sobald die Temperatur deutlich vom Normalwert abweicht. Unser Normalwert liegt gerade bei 27°C. Also lautet die Formulierung für unsere Programmbedingung:

**Wenn die Temperatur über 28°C liegt oder genau 28°C beträgt, soll der Lüfter an gehen.**

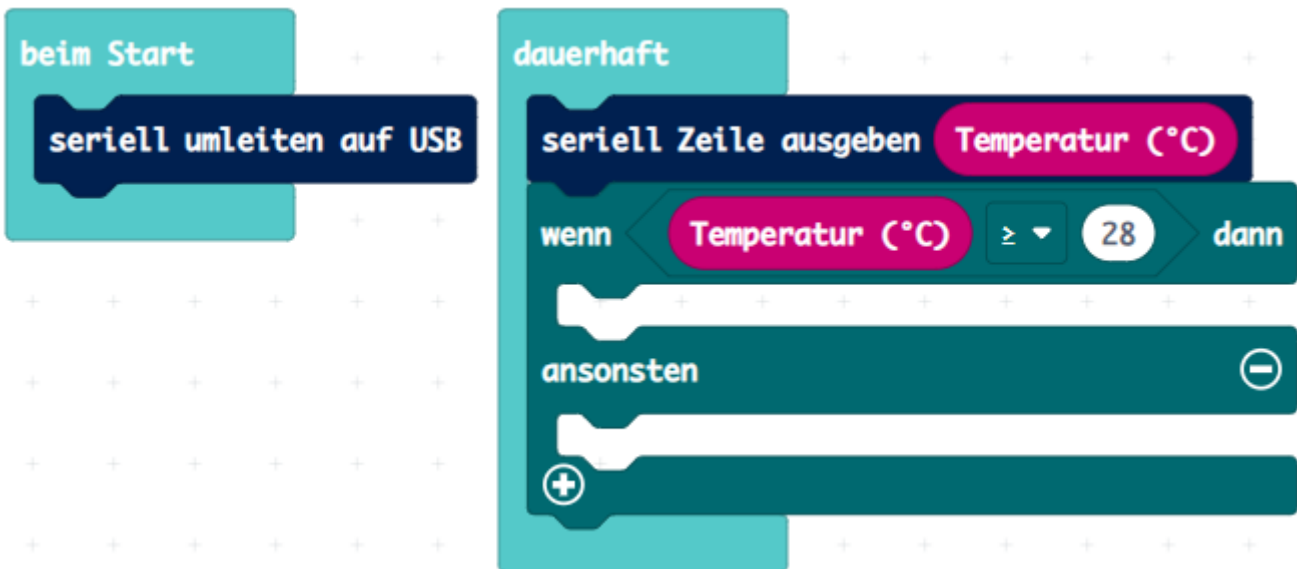
Um diese Bedingung in unserem Programm einzubauen, verwenden wir wieder den Programmblock „wenn wahr dann“.



Wie in der früheren Anleitung setzen wir nun einen Vergleich aus dem Programmbereich „Logik“ ein.



In dem Vergleich setzen wir die gemessene Temperatur und unser Temperaturlimit ein, bei dem der Lüfter starten soll.

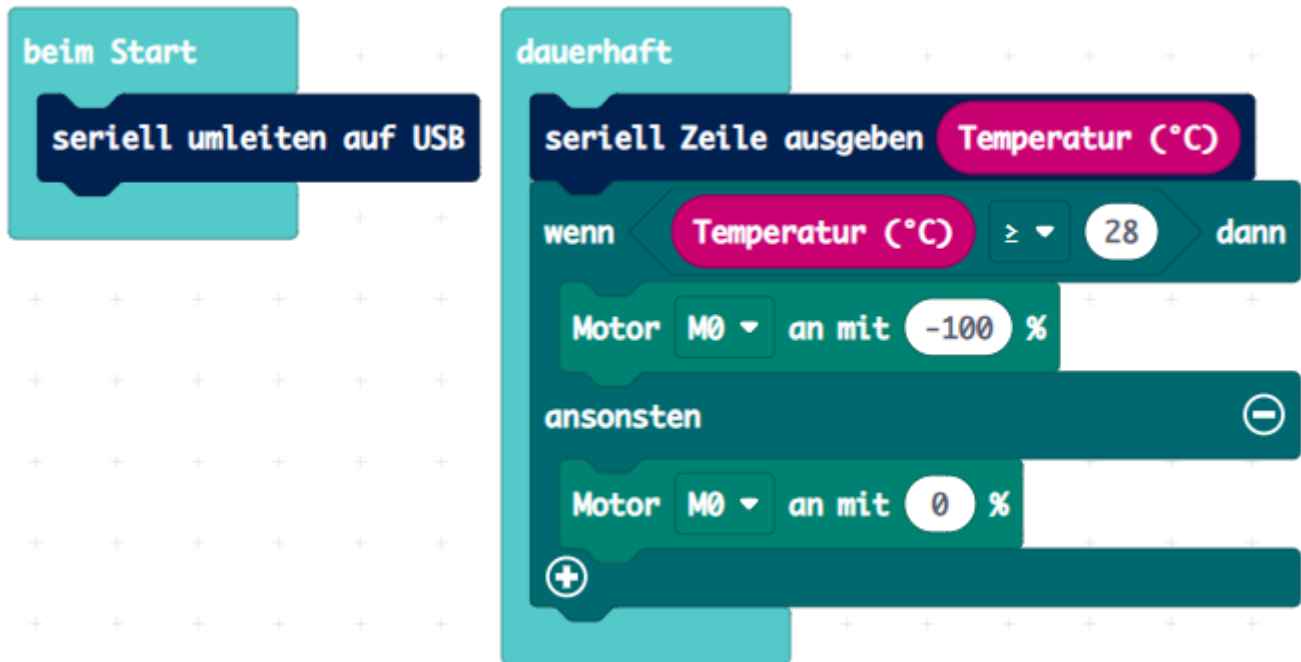


Schau genau hin! Als Vergleichszeichen wählen wir nun einen **Pfeil (>)** mit einem **Strich** darunter. In der Mathematik heißt das „**GrößerGleich**“. In dem Programm bedeutet das nun:

„Wenn die gemessene Temperatur **größer** oder **gleich** 28°C ist, dann mache folgendes...“

Nun folgt natürlich das was folgen muss – wenn es größer oder gleich 28°C ist, soll der Motor (der Ventilator) laufen. Also nehmen wir aus dem Programmbereich „**Motoren**“ erneut den Programmblock „**Motor M0 an mit -100%**“. In diesem Fall wählen wir **-100%**, da unser Lüfter sich nur in diese Richtung drehen kann.

Zusätzlich ergänzen wir den „**ansonsten**“ Bereich. Denn wenn die Bedingung **GrößerGleich 28°C** nicht erfüllt ist, soll der Motor sich nicht drehen. Die Leistung ist dann **0%**.



Lade das Programm auf den Calliope. Was passiert, wenn du den Daumen auf den Sensor legst? Wenn du alles richtig gemacht hast, fängt der Lüfter nach kurzer Zeit an, einen leichten Wind zu erzeugen. Stelle den Ventilator genau neben den Temperatursensor des Calliope. So wird der Calliope automatisch gekühlt, wenn er zu warm wird.

## 4.7 Servomotor

Ein Servo ist ein spezieller Motor, der sich nur in einem bestimmten Bereich drehen kann, nämlich in unserem Fall um maximal 180 Grad. Das ist eine halbe Umdrehung. Das bedeutet, dass er sich von einer festen Ausgangsposition aus entweder 90 Grad nach links oder 90 Grad nach rechts bewegen kann, aber nicht weiter. Servomotoren dieser Art werden oft in Modellbau, Robotik und anderen Anwendungen verwendet, bei denen eine präzise Steuerung der Bewegung erforderlich ist.

Ein Servo besteht im Wesentlichen aus drei Hauptteilen.

**Motor**- Der eigentliche Antrieb, der die Bewegung erzeugt.

**Getriebe** - Übersetzt die Drehbewegung des Motors, um sie stärker oder langsamer zu machen.

**Steuerungselektronik** - Sie sorgt dafür, dass der Servo in die gewünschte Position gebracht wird.

Der Servo wird vom Calliope über ein spezielles Signal, die sogenannte Pulsweitenmodulation (PWM), angesteuert. Das bedeutet, dass man dem Servo Signale schickt, um zu bestimmen, in welche Position er sich bewegen soll. Je nachdem, wie lange die Impulse (Pulse) andauern, bewegt sich der Servo in eine bestimmte Richtung.

Ein kurzer Impuls (z. B. 1 Millisekunde) könnte den Servo in die linke Endposition bewegen ( $0^\circ$ ).

Ein längerer Impuls (z. B. 2 Millisekunden) könnte den Servo in die rechte Endposition bewegen ( $180^\circ$ ).

Ein mittellanger Impuls (z. B. 1,5 Millisekunden) bringt den Servo in die Mittelposition ( $90^\circ$ ).

### Aufgabe

Ein Servo soll vom Calliope angesteuert werden. Ein am Servo befestigter Gegenstand soll sich hin und her bewegen.

### Verkabelung

Wir stecken den Stecker des Servos auf die drei herausragenden Kontaktstifte am hellgrünen Port 1.

Wir drehen den Stecker so, dass die braune Leitung links und die gelbe Leitung rechts ist.

Aber keine Angst. Selbst wenn du es beim ersten Mal falsch machst, kann der Calliope oder der Servo nicht kaputt gehen.

Was hat es mit den drei Leitungen im Stecker auf sich?

Um einen 180° Servo an einen Mikrocontroller oder Kleincomputer anzuschließen, benötigt man drei Leitungen. Diese drei Leitungen sind im Stecker des Servos bereits gemeinsam enthalten.

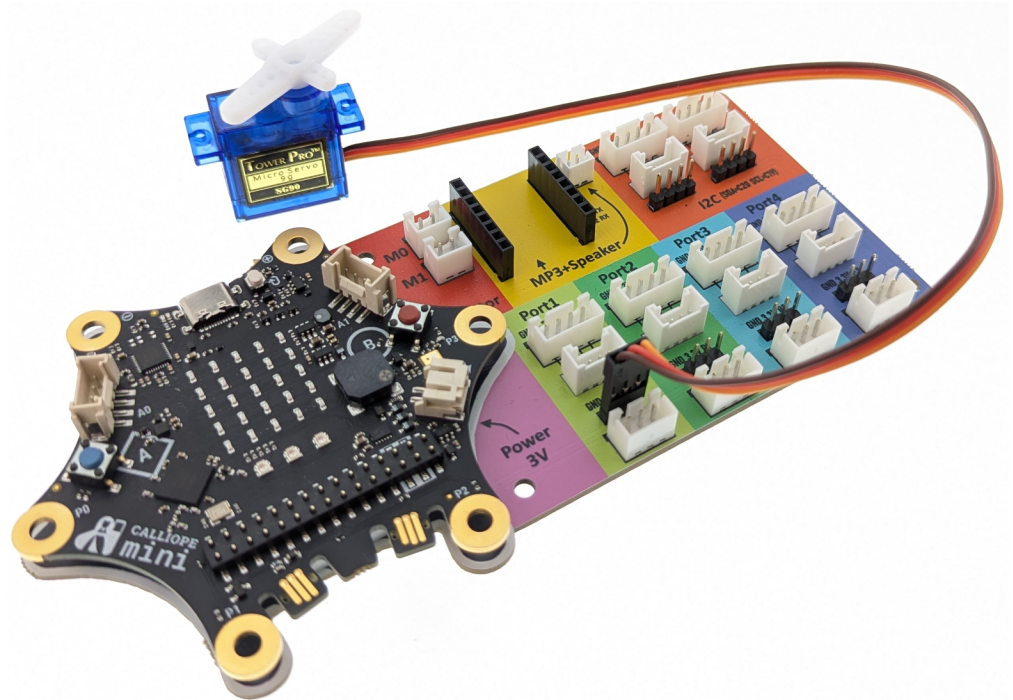
GND (Ground): Dies ist die Erdung oder der Minuspol, der mit dem GND-Pin des Calliope verbunden wird und hat die Leitungsfarbe braun.

VCC (Versorgungsspannung): Das ist die Stromversorgung des Servos. Dieser Pin wird mit dem 3,3 Volt Pin des Calliope verbunden und hat die Leitungsfarbe Rot.

Signal: Hierüber erhält der Servo die Steuerbefehle. Dieser Pin wird mit einem digitalen Pin des Calliope verbunden. Nicht jeder Mikrocontroller hat an jedem Kontakt auch die Möglichkeit, das Signal als PWM (Pulsweitenmodulation) auszugeben. Beim Calliope ist das kein Problem. Die möglichen Kontakte werden auch im Programmblock angezeigt.

## Programm

Wir starten mit den beiden Standardblöcken und nehmen uns zusätzlich den Block für die Servosteuerung. Diesen finden wir in der Programmrubrik **Fortgeschritten** → **Pins**. Er heißt „**setze Winkel von Servo an P0 auf 180°**“





Den Block setzen wir direkt in den „dauerhaft“ Block und ändern den Winkel auf 90°. Das entspricht beim Servo eine mittlere Drehposition.



Den Servo-Block kopieren wir, indem wir mit einem Rechtsklick darauf klicken und „Duplizieren“ auswählen. Wir kopieren den Block dreimal. So kommen wir auf den Bewegungsablauf

„Mitte-Links-Mitte-Rechts“

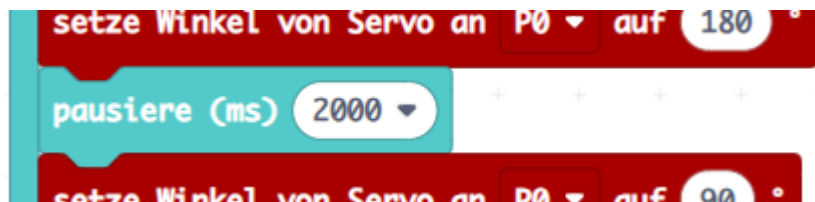
Die entsprechenden Winkel außen sind 0° und 180°. Diese tragen wir natürlich in die Blöcke ein.



Lade das Programm auf den Calliope! Na? Du merkst sicherlich, dass hier etwas faul ist oder? Die Kommandos sind für den Servo zu schnell nacheinander. Der Servo benötigt Zeit, die einzelnen Positionen zu erreichen.

**Wie kannst du das Problem lösen?**

Tipp: Versuche es mit kurzen Pausen zwischen den Positionen!



## 4.8 MP3 – Lass den Calliope sprechen!

Jetzt wirds richtig cool. Auf dem Expanderboard ist ein MP3-Modul vorhanden. Das Modul kann zum einen MP3-Dateien mit Hilfe einer Speicherkarte speichern, aber es kann diese auch über einen Lautsprecher abspielen!

So ergeben sich ganz neue Möglichkeiten, deinen Calliope zu verwenden. Hast du schon neue Projektideen?

Der Calliope kann jetzt **Musik abspielen**.

Der Calliope kann eine **Sprachnachricht** vorlesen, in Abhängigkeit der Temperatur. Dein Calliope kann dadurch „**HITZEFREEEEEEI!!!**“ rufen, wenn es wärmer ist als 28°C.

Eine Pflanze kann mit Hilfe des Calliope nun **nach Wasser rufen**, wenn die Blumenerde zu trocken ist.

Weißt du was eine **Tonibox** ist? Auch so etwas kannst du nun selber bauen.

Aber erst musst du dich durch diese Anleitung arbeiten – Kein Problem, du bist bis hierher gekommen, also schaffst du auch den Rest.

### Aufgabe

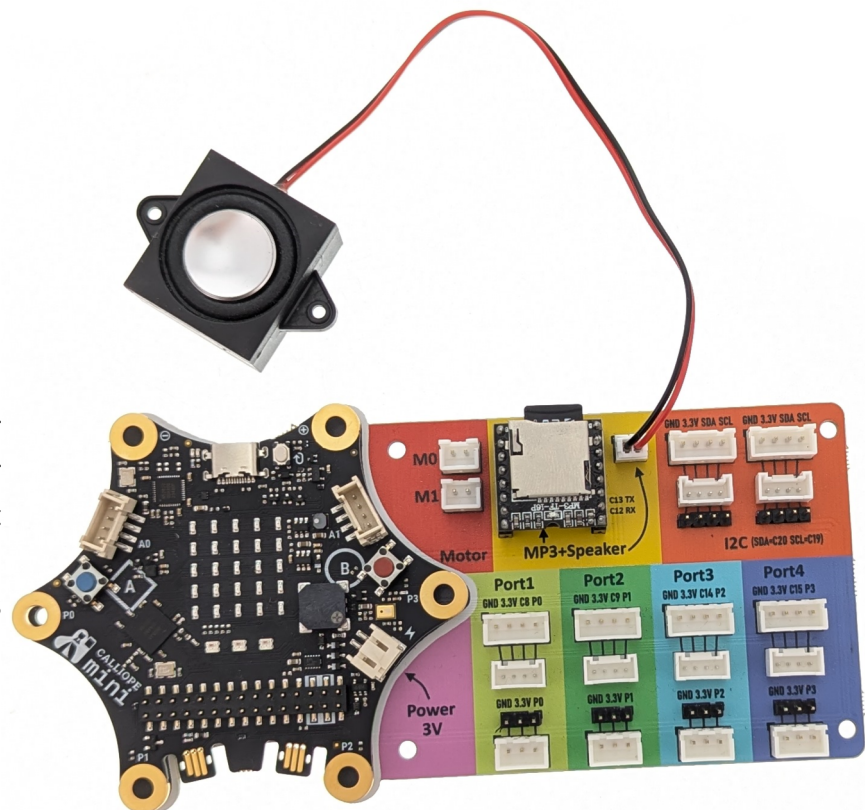
Teil1: Micro-SD-Karte vorbereiten

Teil2: Eine MP3-Datei abspielen

Teil3: Eine MP3-Datei abspielen, wenn es zu warm wird

### Verkabelung

Prüfe, ob das MP3-Modul mit der Micro-SD-Karte auf dem Expander aufgesteckt ist. Rechts daneben ist der Anschluss für den Lautsprecher. Auch dieser muss nun eingesteckt werden.



## **Aufgabenteil 1 – SD-Karte vorbereiten**

Deine SD-Karte muss im Format FAT32 formatiert sein und wir müssen ein paar MP3-Dateien aufspielen. Dazu steckst du die Speicherkarte in deinen PC oder Laptop. Wenn es dafür keinen Steckplatz gibt, kannst du einen Adapter verwenden.

Keine Sorge, das Formatieren der Speicherkarte ist ganz einfach. Du klickst einfach mit der rechten Maustaste auf das Symbol deiner Speicherkarte auf dem Computer. Dann wählst du „Formatieren...“ aus. In dem Menü suchst du nach „Dateisystem“ und wählst „FAT32“ aus. Aber Vorsicht: Alle Daten auf der Speicherkarte werden dabei gelöscht!

Damit du nach dem Aufbau direkt loslegen kannst, brauchst du ein paar Sounddateien im MP3-Format. Diese werden später vom Calliope mit Hilfe des MP3-Moduls abgespielt. Du kannst am Computer oder Laptop selber Sprachdateien aufnehmen. Das Ziel könnte zum Beispiel sein, ein Gerät zu bauen, das dir mit einer Sprachdatei sagt, was es braucht. Zum Beispiel könnte eine Pflanze „sprechen“ und sagen, dass sie Wasser braucht, wenn die Erde zu trocken ist.

Wenn du deine eigene Stimme nicht gerne hören magst, kannst du auch im Internet MP3-Dateien aus Texten erstellen. Es gibt viele Seiten, die das kostenlos anbieten. Dafür suchst du nach „Text zu MP3“ oder „Text to MP3“. Für diese Anleitung speichern wir drei MP3-Dateien auf der Micro-SD-Karte ab. Die Dateien nennen wir 0001.MP3, 0002.MP3 und 0003.MP3.

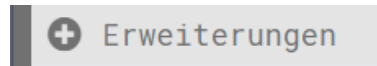
Es ist wichtig, alle Dateien in so zu benennen! Immer vier Stellen mit vorhergehenden Nullen.

## **Aufgabenteil 2 – Eine MP3-Datei abspielen**

Jetzt kommt etwas ganz neues! Der Calliope kann mit hunderten von elektrischen Bauteilen zusammenarbeiten. Aber manchmal sind diese Bauteile sehr komplex. So ist es auch mit dem MP3-Modul. Damit wir dennoch ganz entspannt und einfach damit arbeiten können, verwenden wir eine Programmbibliothek. Diese Bibliothek komprimiert komplexe Programmteile auf die notwendigsten Programmblöcke.

Für das MP3-Modul werden wir nun erst einmal eine Bibliothek hinterlegen und danach fangen wir erst mit der Programmierung an.

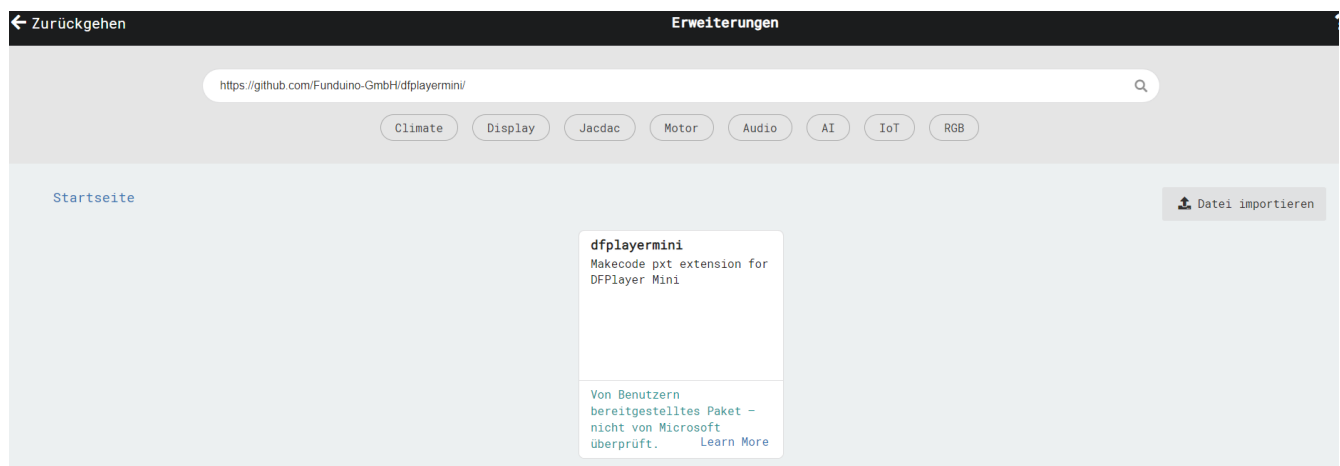
Wir öffnen ein neues Projekt in Makecode und klicken und bei den Programmblöcken auf Erweiterungen:



In der Suchleiste geben wir folgenden Link ein:

<https://github.com/Funduino-GmbH/dfplayermini/>

Das sieht dann so aus:

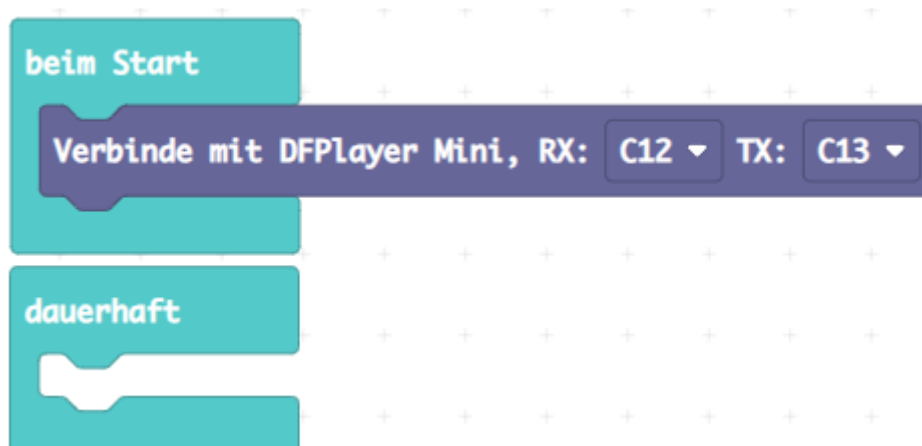


Hier klicken wir mitten in den kleinen Kasten mit dem Namen „dfplayermini“ und wenige Sekunden später erscheint in unseren Programmblöcken eine neue Rubrik mit dem Namen „**DFPlayer Mini**“.

### Jetzt kann das Programmieren losgehen!

Als erstes benötigen wir den Block „Verbinde mit DFPlayer Mini, RX: P0 TX: P1“. Diesen Block ziehen wir in den „**beim Start**“-Block. Dieser Block ist sehr wichtig, denn es wird angegeben, über welche Kontakte des Calliope das MP3-Modul angesteuert wird.

Wir verwenden zum Senden (TX) den Kontakt C13 und zum Empfangen (RX) den Kontakt C12. Die Werte setzen wir in den genannten Programmblock ein. Das sollte dann so aussehen:



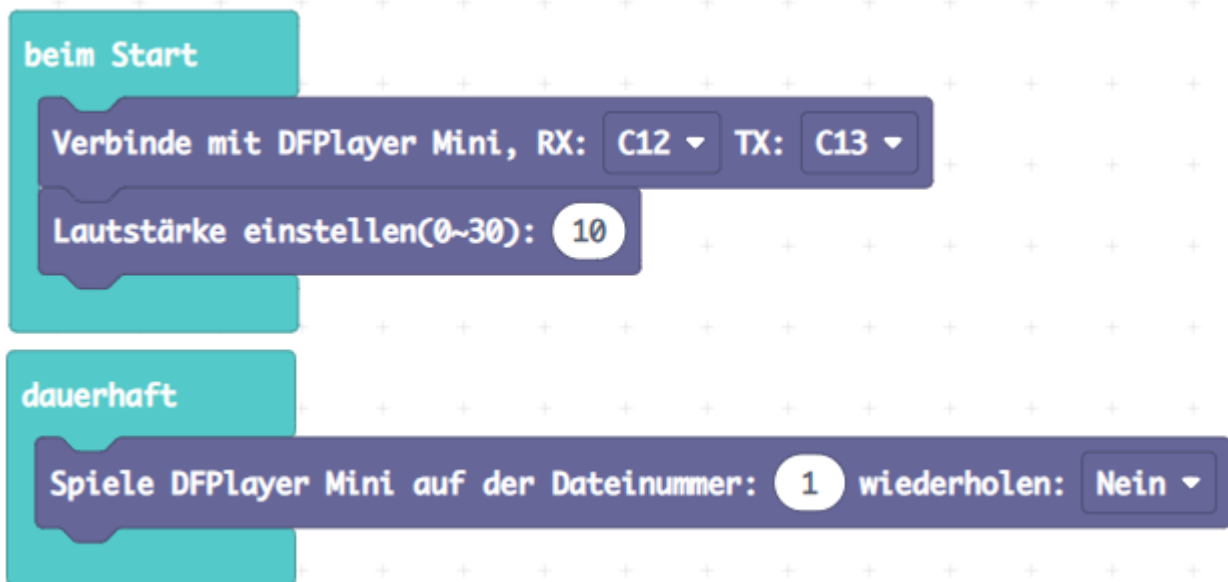
Jetzt nehmen wir aus der Rubrik „DFPlayer Mini“ den Block „Lautstärke einstellen (0~30): 0“ und ziehen ihn in den „beim Start“-Block. Wir haben die Lautstärke auf 10 gestellt, damit man etwas hört, jedoch keine Ohrenscherzen bekommt. Achtung, wenn die Lautstärke hoch ist, zum Beispiel durch eine hohe Lautstärke in der MP3-Datei, kann es passieren, dass sich das MP3-Modul abschaltet. Das liegt daran, dass der Calliope nur mit einer Spannung von 3,3Volt arbeitet und daher keine hohe elektrische Leistung abgeben kann.

Verwende daher zunächst immer nur eine leise Lautstärke, die du später nach und nach erhöhen kannst.

**Wichtig – Der Block für die Lautstärke muss nach dem „Verbinde mit DFPlayer Block“ kommen, da die Lautstärke ansonsten nicht übernommen wird!**



Als letztes ziehen wir den Block „Spiele DFPlayer Mini auf der Dateinummer: 0 wiederholen Nein“ in den „dauerhaft“-Block. Nun können wir das Programm herunterladen und die MP3-Datei hören, die wir vorher auf der Speicherkarte gespeichert haben.



Wenn du alles richtig gemacht hast, ertönt jetzt ein bisschen Krach. Denn der Calliope startet die Datei permanent von vorne.

Achtung: Wenn deine MP3-Datei am Anfang mit Stille (also ohne Geräusche) beginnt, wirst du nichts hören.

Wir müssen dem Programm nun also eine Zeit vorgeben, in der die Datei abgespielt wird. Dazu verwenden wir die Pause. In unserem Beispiel ist die MP3-Datei 5 Sekunden lang. Daher verwenden wir eine Pause von 5000ms.

Den Programmblock „**pausiere**“ findest du in der Rubrik „**Grundlagen**“.

### **Aufgabenteil 3 - Eine MP3-Datei abspielen, wenn es zu warm wird**

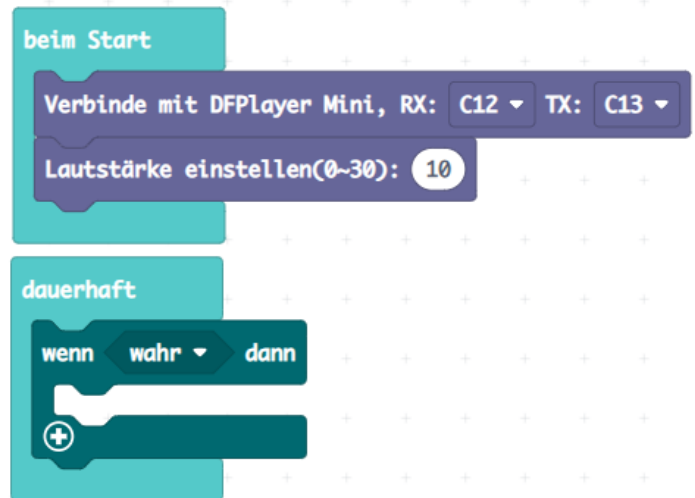
Als Vorbereitung erstellen wir eine MP3-Datei mit Hilfe einer Online Text-zu-MP3 Funktion. Wir speichern den folgenden Satz als MP3-Datei ab:

„Meine Güte ist das warm hier, kann mal jemand ein Eis bringen?“

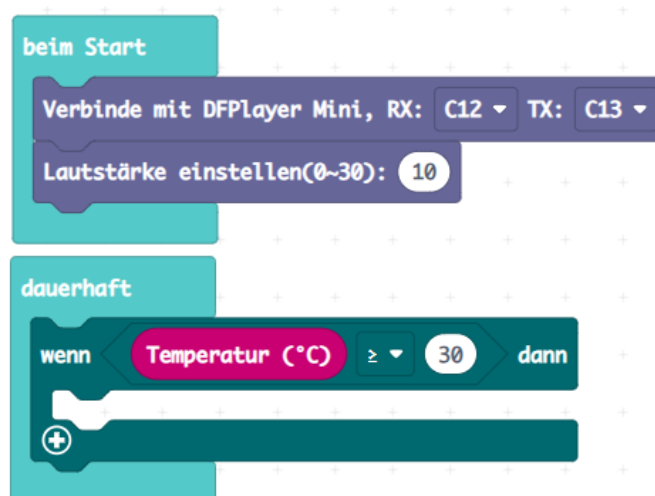
Die Datei speichern wir unter dem Namen 0001.MP3 auf der Speicherkarte ab.

## Programm

Wenn du die vorherige Aufgabe gelöst hast, ist es jetzt natürlich nicht mehr so schwer. Wir müssen nun mal wieder eine Bedingung einbauen. Du weißt wie das geht! Vorab bringen wir schon einmal die notwendigen Programmblöcke für das MP3-Modul in Position.



Die Bedingung ist, dass die Temperatur über oder gleich 30°C sein muss, damit die MP3-Datei abgespielt wird.



Als letzten Schritt setzen wir das Abspielen der MP3-Datei ein, für den Fall, dass die Bedingung (30°C) erfüllt ist.

```
beim Start
  Verbinde mit DFPlayer Mini, RX: C12 TX: C13
  Lautstärke einstellen(0~30): 10

dauerhaft
  wenn Temperatur (°C) ≥ 30 dann
    Spiele DFPlayer Mini auf der Dateinummer: 1 wiederholen: Nein
  pausiere (ms) 5000
```

Falls etwas nicht funktioniert kann es an dieser Stelle sinnvoll sein, sich den Messwert des Temperatursensors dauerhaft anzeigen zu lassen. Weißt du selber, wie man das machen könnte?

Hier ein Beispiel:

```
beim Start
  Verbinde mit DFPlayer Mini, RX: C12 TX: C13
  Lautstärke einstellen(0~30): 10

dauerhaft
  zeige Zahl Temperatur (°C)
  wenn Temperatur (°C) ≥ 30 dann
    Spiele DFPlayer Mini auf der Dateinummer: 1 wiederholen: Nein
  pausiere (ms) 5000
```

## 4.9 RFID Karten lesen

RFID (Radio Frequency Identification) ist eine Technik, mit der man Daten per Funk übertragen kann.

Das System wird häufig zur Identifizierung von Personen verwendet. Zum Beispiel wenn du oder deine Eltern kontaktlos mit der EC-Karte Geld bezahlen, werden die erforderlichen Bankdaten vom eigenen Bankkonto übertragen. Oder wenn sich Arbeiter bei der Arbeit einstempeln, halten Sie häufig eine Chipkarte oder einen Schlüsselanhänger vor ein Lesegerät um die Arbeitszeit zu messen. Der Empfänger erkennt dann ganz eindeutig, welcher der vielleicht Tausend Arbeiter sich ein gestempelt hat.

### Wie funktioniert das?

Es gibt zwei wichtige Teile bei einem RFID-System: den Sender (oft auch Tag genannt) und den Empfänger (oft Reader genannt).

**RFID-Sender (Tag):** Ein RFID-Sender ist ein kleiner Chip, der Daten enthält, wie zum Beispiel eine individuelle Nummer oder eine Information über Personen. Dieser Chip ist mit einer Antenne verbunden, damit er funken kann. Es gibt aktive RFID-Tags mit einer eigenen Spannungsversorgung aber auch winzige, oder sogar versteckte RFID-Tags, die keine eigene Batterie haben.

**RFID-Empfänger (Reader):** Der RFID-Empfänger ist ein Gerät, das ständig nach Sendern „sucht“. Wenn ein RFID-Tag in die Nähe des Empfängers kommt, sendet der Empfänger ein Energiefeld aus. Dieses Feld weckt den RFID-Tag auf und gibt ihm genug Energie, um seine Daten zurückzufunkeln.

### Projektbeispiel:

Stell dir vor, du baust mit dem Calliope ein kleines Sicherheitssystem für dein Zimmer. Du hängst den RFID-Empfänger an deine Tür. Wenn du deinen speziellen RFID-Tag (vielleicht versteckt in deinem Schlüsselanhänger) an die Tür hältst, erkennt der RFID-Empfänger den RFID-Tag und der Calliope gibt das Signal, die Tür zu entriegeln.

Das RFID-System funktioniert wie ein unsichtbarer Schlüssel: Nur wenn der richtige Sender in die Nähe kommt, weiß der Empfänger, dass er eine bestimmte Aktion auslösen soll.

### Aufgabe

Teil 1: Lese die UID (Unique Identifier) der RFID-TAGs aus und lasse Sie in der Seriellen Kommunikation anzeigen.

Teil 2: Wenn der richtige RFID-Tag an den Empfänger gehalten wird, soll eine Melodie ertönen.

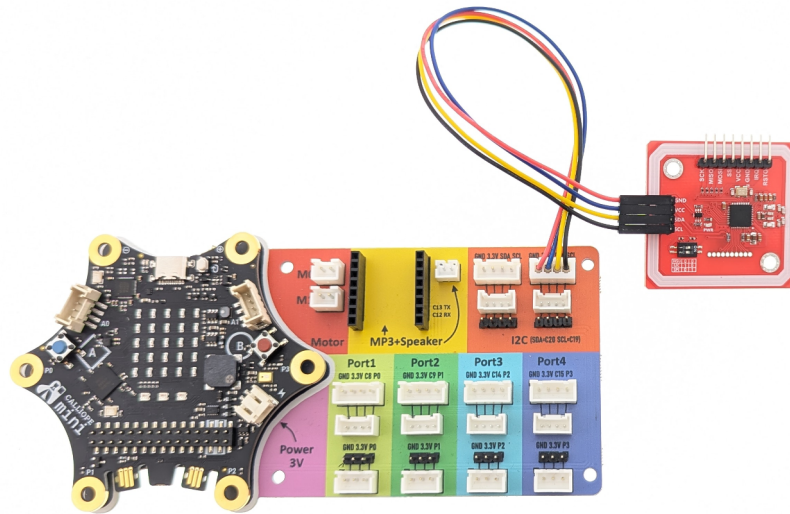
## Verkabelung

Auch mit dem Calliope, kannst du ein RFID-System aufbauen. Dazu schließen wir den RFID-Empfänger an eine der I2C Buchsen des Expanderboards an. **Aber Achtung!** Für den RFID-Empfänger gibt es kein konfektioniertes Kabel. Wir stecken zwar die eine Seite des Kabels mit dem Stecker in den Expander, aber die andere Seite muss am Empfänger Kabel für Kabel angesteckt werden.

Achte auf die Beschriftung der Kontakte am Expander und an dem Empfänger.

1. Kabel (links außen): Am Expanders steht GND. Das Kabel wird am Empfänger an GND angesteckt.
2. Kabel: Am Expanders steht 3.3V. Das Kabel wird am Empfänger an VCC angesteckt.
3. Kabel: Am Expanders steht SDA. Das Kabel wird am Empfänger an SDA angesteckt.
4. Kabel (rechts außen): Am Expanders steht SCL. Das Kabel wird am Empfänger an SCL angesteckt.

Das ganze muss nach der Verkabelung folgendermaßen aussehen.



Ganz wichtig ist bei dem RFID-Modul, dass es vor der Programmierung über zwei Schiebeschalter in den I2C-Modus gebracht wird. Dazu wird zunächst die Schutzfolie über den Schaltern entfernt und der eine Schalter wird auf „ON“ gestellt, wie auf dem folgenden Bild zu sehen ist.



## Teilaufgabe 1 – RFID-TAG auslesen

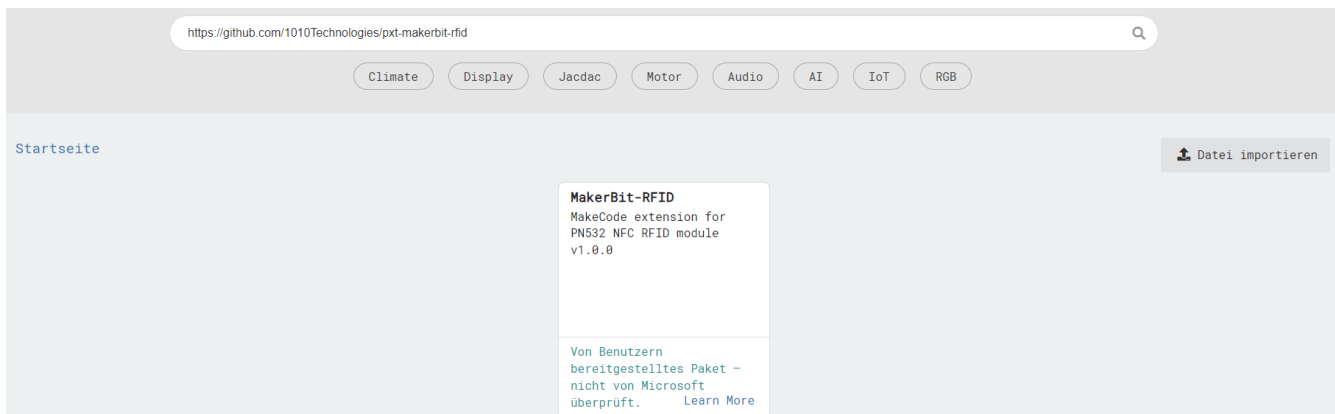
Als erstes möchten wir den RFID-Tag auslesen. Genauer gesagt möchten wir die UID, also die Individuelle Nummer des Tags auslesen. Wenn du einen RFID-Tag in die Nähe des Empfängers hältst, sendet der Tag seine UID (seine individuelle Nummer) an den Calliope. Der Calliope soll uns diese Nummer in der Seriellen Kommunikation anzeigen. Später möchten wir damit etwas Bestimmtes auslösen, z. B. Ein Symbol auf der LED-Matrix zeigen oder eine LED einschalten.

Lets go! Wir starten direkt mit etwas neuem! Das RFID-Modul ist in der Anwendung vergleichsweise komplex. Daher verwenden wir hier erneut, wie auch in der Anleitung zum MP3-Player, eine Produktbibliothek.

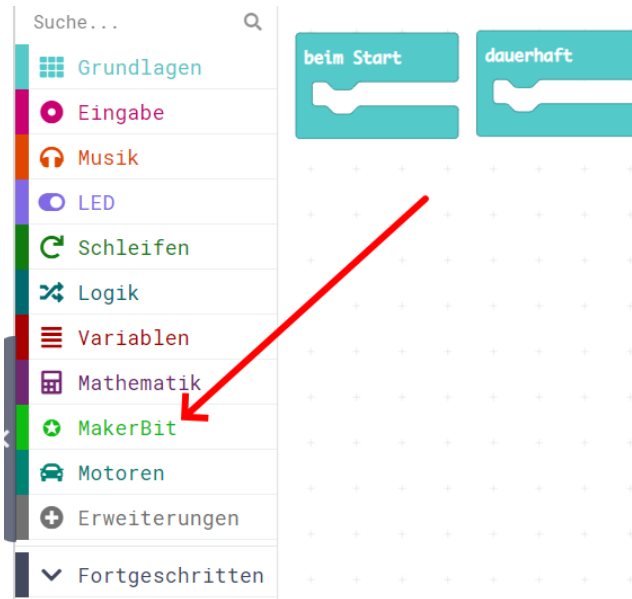
Die Produktbibliothek wird wieder im Bereich „Erweiterungen“ hinzugefügt. In diesem Fall können wir die Bibliothek jedoch nicht so einfach über die Suchfunktion finden. Dieses Mal müssen wir den Link zur Bibliothek selber in die Suchleiste eintippen! Sei konzentriert und vertippe dich nicht!

### Programmbibliothek

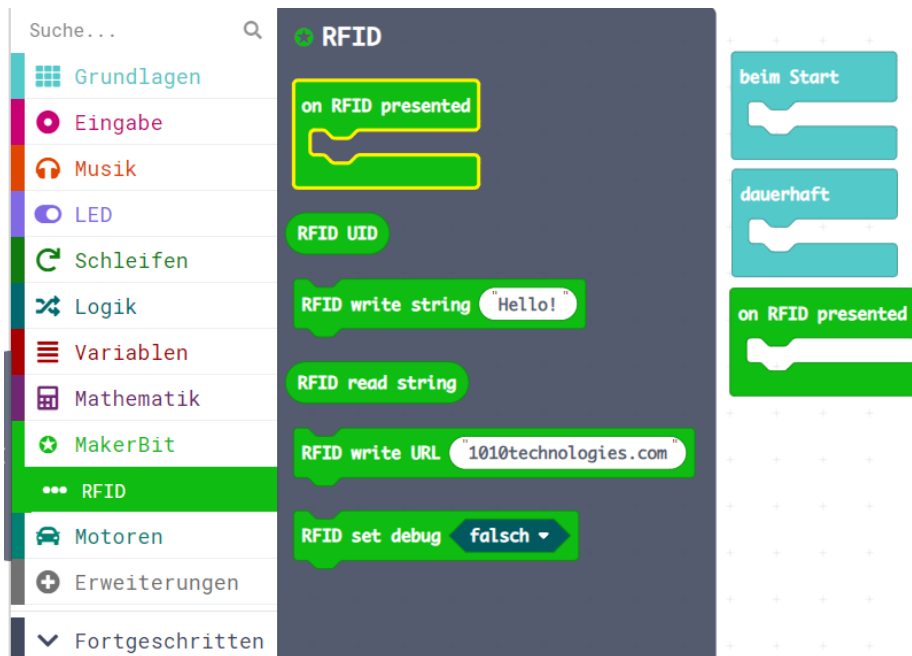
<https://github.com/1010Technologies/pxt-makerbit-rfid>



In der Mitte wird dann die passende Bibliothek angezeigt. Diese klicken wir an und kurz danach ist die Bibliothek nutzbar. Das erkennen wir daran, dass in den Programmblöcken eine neue Rubrik mit dem Namen „MakerBit“ erschienen ist.

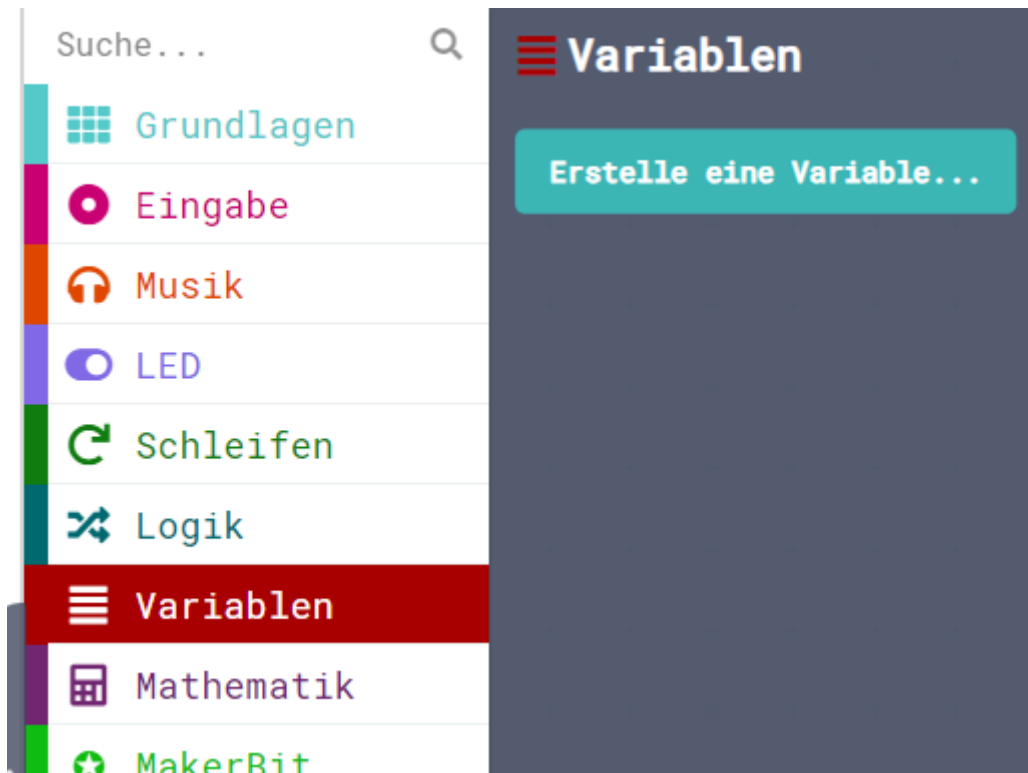


Die Programmibliothek kann nun verwendet werden. Wir nehmen zuerst den Programmblock „on RFID presented“ und legen ihn neben die anderen beiden Programmblöcke. Dass der Block daneben liegt hat einen guten Grund. Der Programmblock arbeitet nun parallel zum Hauptprogramm und wartet im Hintergrund darauf, dass ein RFID-Tag in die Nähe des Empfängers kommt.

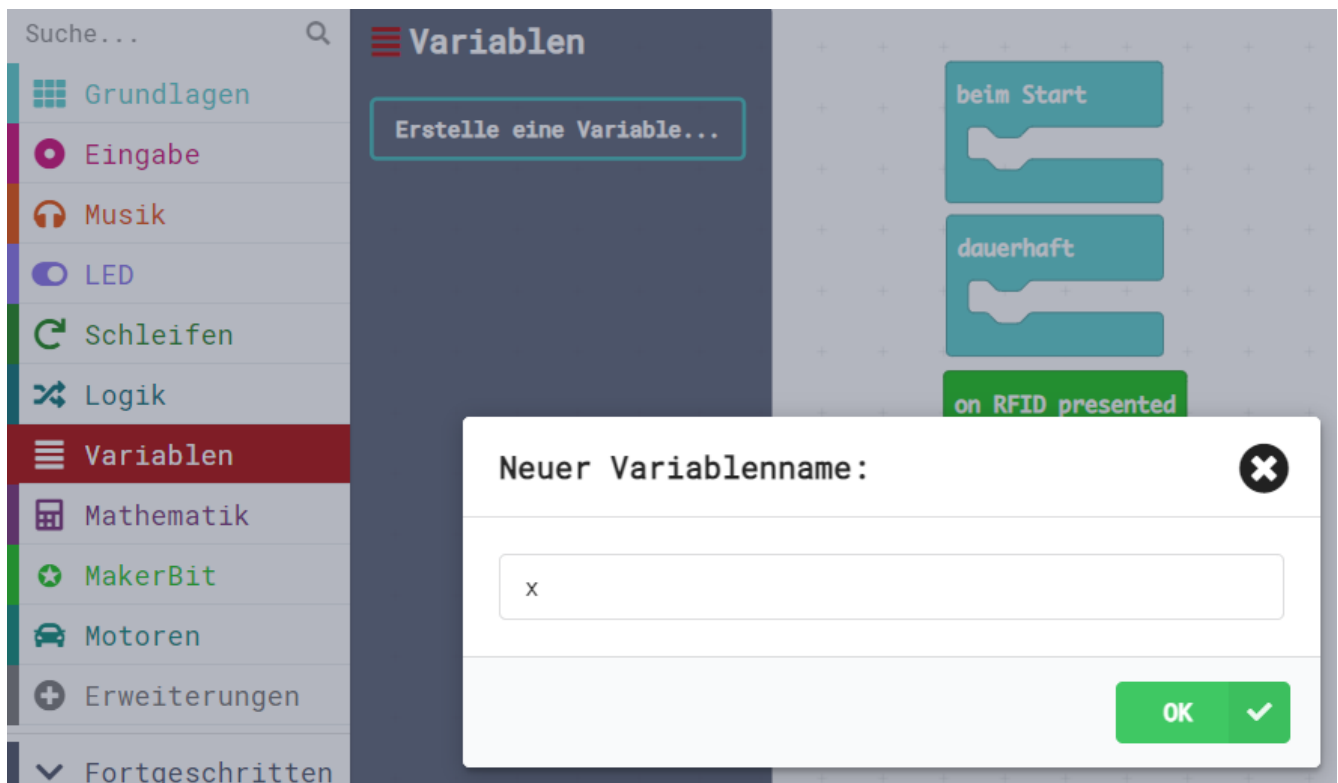


Wenn dieser Fall eintritt, soll der Empfänger die UID des RFID-Tags lesen und speichern. Aber wo soll das jetzt abgespeichert werden? Hat der Calliope eine Festplatte? Naja nicht so ganz, aber es gibt die Möglichkeit, einen kleinen Speicherplatz mit eigenen Daten, zum Beispiel Zahlen oder Text zu füllen. Ein solcher Speicherplatz wird auch „Variable“ genannt. Das kommt vielleicht daher, dass man verschiedene Sachen dort beliebig abspeichern kann.

Wir legen nun eine solche Variable an und geben der Variablen einen Namen. Zum Beispiel „Hund“, „Katze“, „Schneemann“ oder einfach nur „x“. Was nehmen wir? Wir nennen die Variable „x“. Dazu klicken wir auf „Variablen“ und dann auf „Erstelle eine Variable“



Dort tragen wir dann den Namen ein.



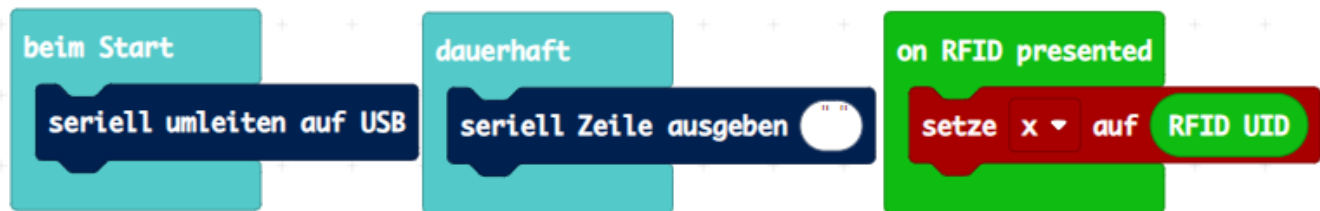
Jetzt haben wir neue Programmblöcke im Angebot. Wir verwenden „setze x auf 0“ und schieben ihn in den Programmblock „on RFID presented“.

Was bedeutet das? Der Block wartet darauf, dass ein RFID-Tag dem Empfänger präsentiert wird (daher der Begriff „presented“). Sobald das eintritt, wird im Speicherplatz der Variablen „x“ die Ziffer 0 abgespeichert. Aber das ist quatsch. Es soll ja nicht die „0“, sondern die UID der Karte unter dem Namen x gespeichert werden. Also tauschen wir die „0“ gegen den Programmblock „RFID UID“ aus. Diesen Block findest du in der Programmrubrik „MakerBit“.

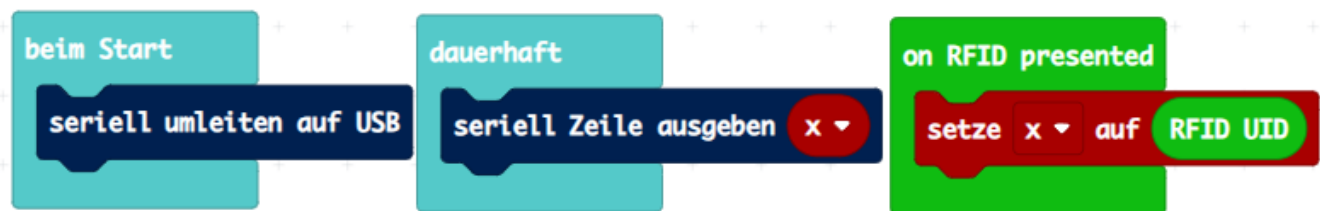


Was auch immer im Hauptprogramm (**dauerhaft**) abläuft; sobald ein RFID-Tag an den Empfänger gehalten wird, wird unter der Bezeichnung „x“ die UID des RFID-TAGs abgespeichert.

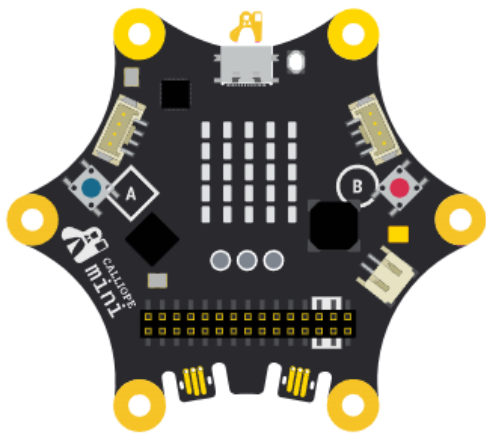
Das werden wir nun testen. Wir erinnern uns an das Programm zur Seriellen Kommunikation und senden den ausgelesenen Wert „x“ zunächst einmal an die Serielle Kommunikation.



In die Zeile, die ausgegeben werden soll, ergänzen wir nun unsere Variable „x“.



Nun klicken wir auf „Daten anzeigen Calliope mini“. Im unteren Bereich des Anzeigefensters wird nun die UID meines RFID-Tags angezeigt. In diesem Beispiel haben wir zwei verschiedene RFID-Tags. Der eine Tag gibt die UID **D3B06A92** zurück und der zweite RFID-TAG sendet die UID **87CCF467**  
ACHTUNG!!! Bei dir steht mit Sicherheit eine andere UID, da du nicht den gleichen TAG benutzt.



||| Daten anzeigen Simulator

||| Daten anzeigen Calliope mini

← Zurück

Werte werden protokolliert, wenn der Calliope mini Daten sendet

D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92  
D3B06A92



Geschafft, Teilaufgabe 1 haben wir erfolgreich gelöst!

## Teilaufgabe 2 – Melodie in Abhängigkeit der Chipkarte abspielen

Wenn der richtige RFID-Tag an den Empfänger gehalten wird, soll eine Melodie ertönen. In unserem Fall verwenden wir zwei RFID-TAGs. Eine RFID-Chipkarte und einen RFID-Schlüsselanhänger.

Jetzt kommt ein Thema, das nicht wirklich neu ist. Wir müssen nun wieder eine Bedingung programmieren. In diesem Konkreten Fall lautet die Bedingung so:

**„Wenn der RFID-TAG mit der UID D3B06A92 erkannt wird, soll eine Melodie ertönen.“**

Wir werden daher nun das vorhandene Programm aus Teilaufgabe 1 mit dem Programmblock **„wenn wahr dann“** um eine Bedingung erweitern.



Welche Bedingung tragen wir ein? Wir verwenden erst mal einen Vergleich und befüllen den Programmblock anschließend.



„Wenn der RFID-TAG mit der UID D3B06A92 erkannt wird, soll eine Melodie ertönen.“

Also tragen wir hier unsere Variable „x“ ein, die mit der vorher ausgelesenen UID verglichen wird. Der „wenn = dann“ Block prüft nun, ob die UID des RFID-TAGS S3B06A92 ist. Und nur wenn dieser Fall eintritt, wird der innere Programmblock (der jetzt noch nicht vorhanden ist) ausgeführt.



Wir ergänzen nun laut der Aufgabenstellung eine Melodie, die abgespielt wird, wenn der richtige RFID-Tag vor den Empfänger gehalten wird. Dazu nehmen wir aus der Unterkategorie „...mehr“ der

Programmrubrik „Musik“ den Programmblock „spiele kichern bis zum Fertigstellen“ und setzen ihn in den grünen Programmblock „wenn = dann“ ein.



Ok, das Programm funktioniert schon! Aber die Melodie hört nicht wieder auf, oh je was für ein Krach. Warum ist das so?

Die Variable x wurde einmal mit der UID „C33B8F14“ beschrieben und bleibt solange unverändert, bis erneut ein RFID-TAG vor den Empfänger gehalten wird. Nur für den Fall, dass ein **ANDERER** RFID-TAG, mit einer anderen UID erkannt wurde, wird unter der Variablen x auch ein anderer Wert gespeichert. Erst dann ist in dem Programm die Bedingung „wenn x=C33B8F14“ nicht mehr erfüllt und der Krach hört auf.

Es gibt aber einen klügeren Weg, das Abspielen der Melodie zu begrenzen. Und zwar ändern wir den unter der Variablen x gespeicherten Wert, nachdem die Melodie einmal abgespielt wurde.

Dazu verwenden wir erneut den Programmblock „setze x auf 0“ aus der Programmrubrik „Variablen“. Hier gibt es eine Sache besonders zu beachten!

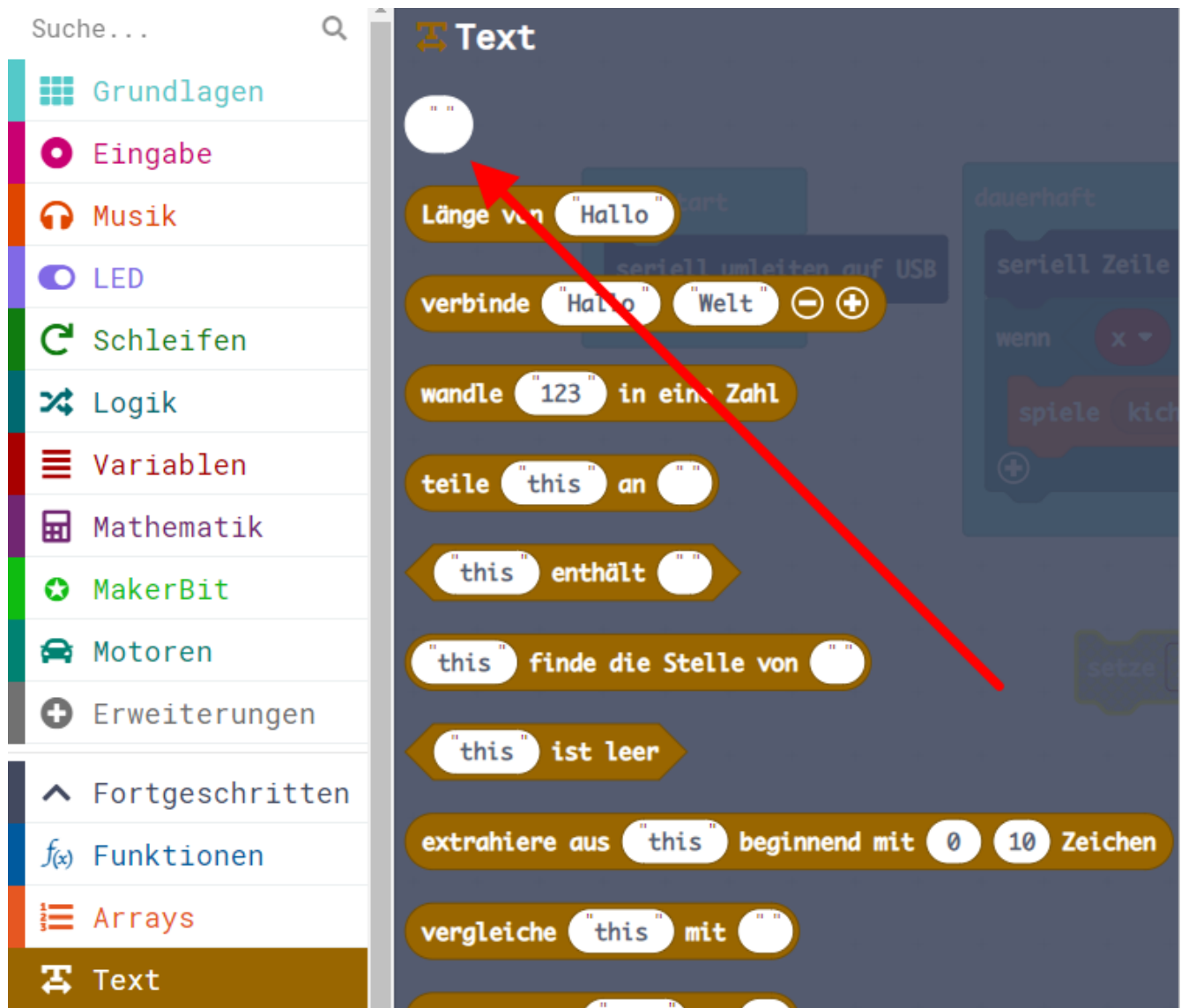
Der standardmäßige Programmblock hat am Ende eine Zahl, die Null:



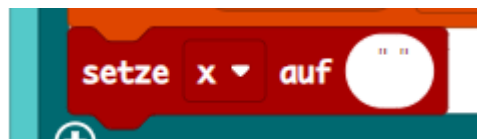
Die Variable x wurde jedoch als Text abgespeichert. Auch im Vergleich haben wir mit der folgenden Zeile die Variable mit einer Textzeile verglichen. Die Textzeile erkennt man daran, dass in der Zeichenfolge auch Buchstaben enthalten sind.



Daher muss man in dem Programmblock für die Variable manuell einen Textblock einfügen. Den Textblock finde ich in der Programmrubrik „Fortgeschritten“ → „Text“



Zusammen sieht der Block folgendermaßen aus.



In dem Textblock tragen wir vorher einen Beliebigen Wert ein. Wir haben das Wort „Leer“ gewählt. Das soll verdeutlichen, dass die Variable wieder „leer“ ist.



Nun ist es geschafft. Das Programm wird hochgeladen und nur wenn der erste RFID-TAG mit der UID „C33B8F14“ vor den Empfänger gehalten wird, wird ein einziges Mal die Melodie abgespielt. Wenn ein anderer RFID-TAG vor den Empfänger gehalten wird, dann ertönt auch keine Musik.

## 4.10 Tonibox nachbauen

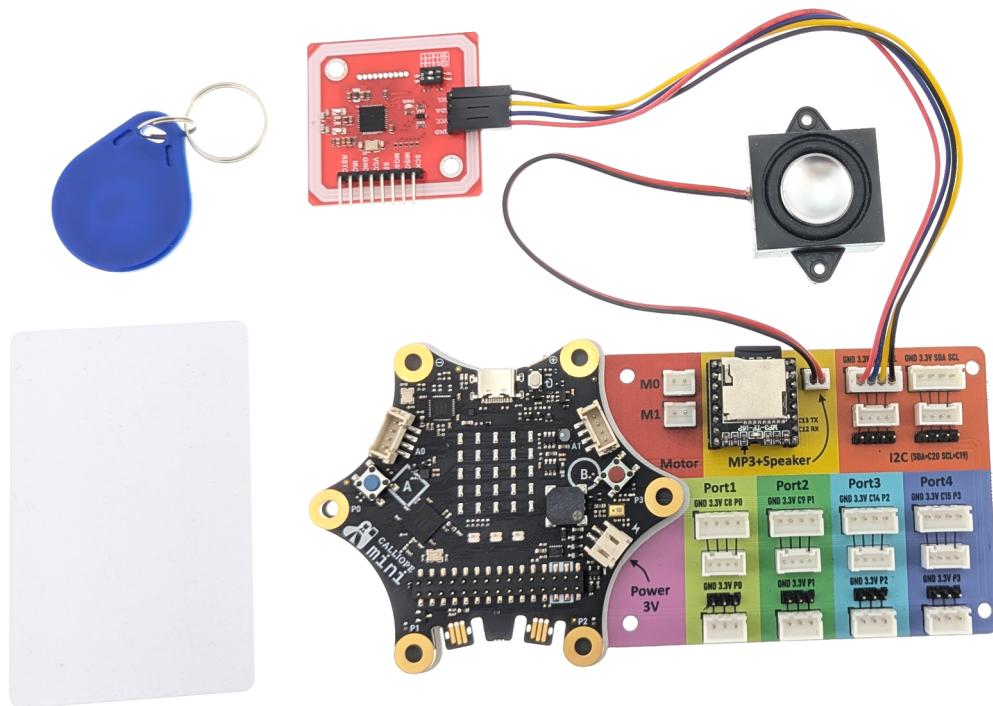
Eine Toniebox ist ein kindgerechtes, einfach zu bedienendes Audiosystem, das speziell für Kinder entwickelt wurde. Sie dient dazu, Hörspiele, Musik und andere Audiodateien abzuspielen. Das Besondere an der Toniebox ist, dass sie ohne klassische Tasten funktioniert und stattdessen durch sogenannte "Tonies" gesteuert wird – das sind kleine, bunt gestaltete Spielfiguren.

Wenn ein Kind eine Tonie-Figur auf die Toniebox stellt, wird die entsprechende Audiodatei automatisch abgespielt.

In unserem Fall soll eine MP3-Datei abgespielt werden, wenn der zur Datei passende RFID-Tag an den RFID-Empfänger gehalten wird. Wir verwenden den RFID-Schlüsselanhänger für die erste MP3-Datei und die RFID-Chipkarte für die zweite MP3-Datei.

### Verkabelung

Für diese Aufgabe muss der RFID-Empfänger, das MP3-Modul und der Lautsprecher angeschlossen werden. Die Verkabelung sieht dann aus wie auf dem folgenden Bild.



### Programm

In den vorherigen Anleitungen hast du schon die verwendeten Module kennengelernt. Daher sollst du nun selber aktiv werden und möglichst selbstständig arbeiten.

Orientiere dich an folgendem Ablauf.

1. RFID-Empfänger, MP3-Modul und Lautsprecher anschließen
2. Auf der SD-Karte zwei Dateien abspeichern mit den Bezeichnungen 0001.mp3 und 0002.mp3
3. Finde die beiden UID der RFID-Tags heraus. Du kannst folgenden Code verwenden.



Notiere dir die beiden Codes: 1. \_\_\_\_\_ 2. \_\_\_\_\_

4. Entferne die Blöcke für die serielle Kommunikation, da das MP3-Modul ansonsten nicht funktionieren kann.

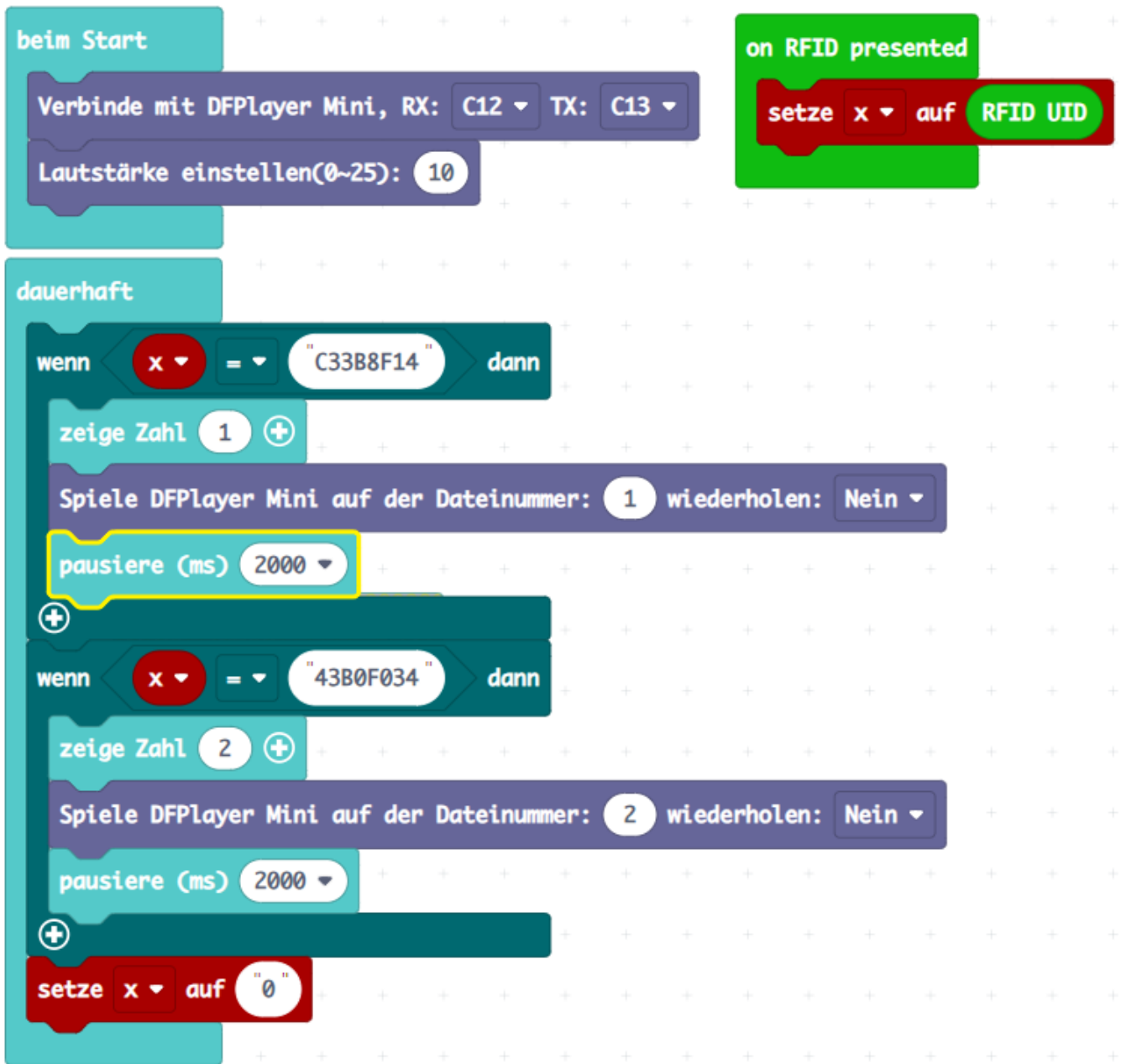
5. Überlege dir die passenden Bedingungen

- Wenn die UID xxxxxxx erkannt wird, dann spiele die erste Datei ab
- Wenn die UID xxxxxxx erkannt wird, dann spiele die zweite Datei ab

6. Denke auch daran, Pausen einzubauen, damit die Dateien vollständig abgespielt werden können

Mehr wird nicht verraten. Verwende das folgende Beispielprogramm nur dann, wenn du ansonsten wirklich nicht weiter kommst.

**Vorsicht!** In dem Beispielprogramm sind die UIDs aus unserem Beispielprogramm eingetragen. Deine UIDs werden andere sein!

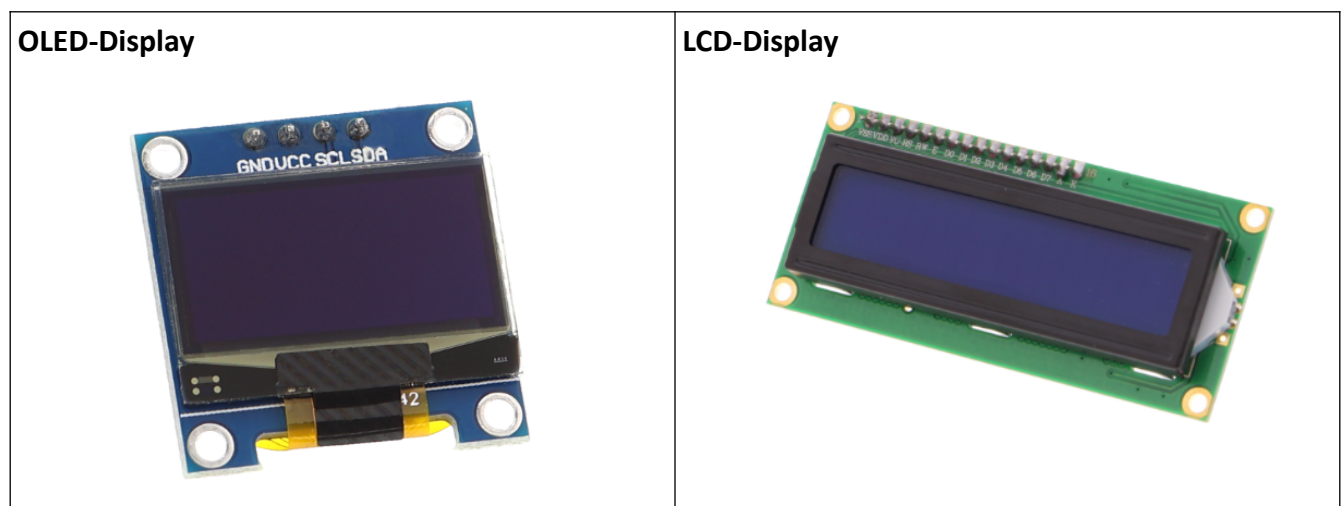


## 4.11 Ein Display verwenden – LCD 1602 oder OLED

Ein Display ist ein typisches Bauteil in elektronischen Geräten. Es kann Informationen anzeigen und über weitere Schritte informieren.

Es gibt sehr viele unterschiedliche Displays. Für den Calliope eignen sich **LCD-Displays** und **OLED-Displays** besonders gut.

**ACHTUNG** – Der Calliope-Expander wird je nach Verfügbarkeit entweder mit einem **OLED-Display** oder mit einem **LCD-Display** ausgeliefert.



Prüfe zunächst, ob du ein OLED-Display oder ein LCD-Display hast.

Wenn du ein OLED-Display hast, kannst du in diesem Kapitel mit der Anleitung fortfahren. Wenn du ein LCD-Display hast, dann findest du die Fortsetzung im nächsten Kapitel.

## 4.11.1 Ein Display verwenden – OLED

### Aufgabe

Teil 1: Ein OLED soll deinen Namen anzeigen.

Teil 2: Ein OLED soll die Temperatur anzeigen.

### Verkabelung

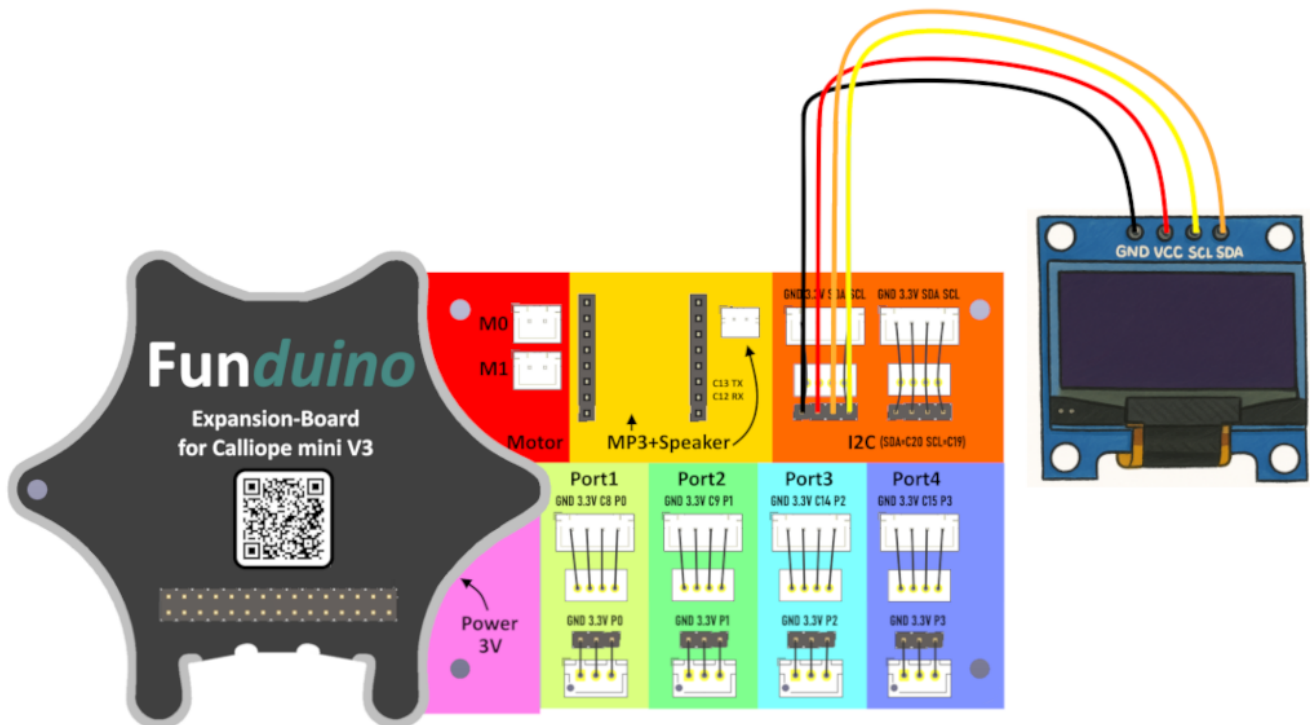
Die Verkabelung ist etwas schwieriger als sonst, da es kein vorkonfektioniertes Kabel gibt. Du musst das Display selbst verkabeln.

Das Display erhält seine Daten über die I2C Schnittstelle. Daher verbinden wir das Display mit den Kontakten im orangefarbenen I2C-Bereich des Expanders.

Welchen Steckplatz wir verwenden ist hier irrelevant. Das liegt daran, dass sehr viele I2C Module parallel zueinander, also gleichzeitig, mit einem Mikrocontroller verbunden werden können.

Wir entscheiden uns jetzt für die Kontaktstifte ganz rechts außen im orangefarbenen Bereich und verkabeln der Reihe nach:

1. Kabel: Am Display in den Kontakt GND und die andere Seite am Expander in den Kontakt GND.
2. Kabel: Am Display in den Kontakt VCC und die andere Seite am Expander in den Kontakt 3.3V.
3. Kabel: Am Display in den Kontakt SDA und die andere Seite am Expander in den Kontakt SDA.
4. Kabel: Am Display in den Kontakt SCL und die andere Seite am Expander in den Kontakt SCL.



## Programm

Um das Display ansteuern zu können, benötigen wir erneut eine Programmbibliothek. Dazu klicken wir auf die Rubrik „**Erweiterungen**“ und geben in das Suchfeld „**OLED**“ ein.

Die folgende Programmbibliothek erscheint.



In den Programmblöcken erscheint eine neue Rubrik mit dem Namen „**OLED12864\_I2C**“. Hier finden wir nun alle Funktionen rund um das Display.

- Grundlagen
- Eingabe**
- Musik
- LED
- Schleifen
- Logik
- Variablen
- Mathematik
- Motoren
- OLED12864\_I2C**
- Erweiterungen
- Fortgeschritten
- Funktionen
- Arrays
- Text
- Spiel
- Bilder
- Pins

```
OLED12864_I2C  
init OLED with addr 60  
show a Number at x 0  
y 0  
number 12  
color 1  
show string at x 0  
y 0  
text "Hello!"  
color 1  
draw a rectangle at x1 0  
y1 0  
x2 60  
y2 30  
color 1  
draw a vertical line at x 0  
y 0  
number 10
```

## Teilaufgabe 1 - Ein OLED soll deinen Namen anzeigen

Das Display wird mit der sogenannten I2C Datenleitung mit dem Calliope verbunden. Wir haben bereits erfahren, dass man bei dieser Art der Verkabelung ganz viele I2C-Fähige Module gleichzeitig anschließen kann. Aber woher weiß der Calliope, an welches der angeschlossenen Module die Daten gesendet werden sollen?

Dazu hat jedes I2C-Modul, also auch unser Display, eine eigene Adresse. Das ist genauso wie die Adresse von dem Haus in dem du lebst. Der Postbote weiß ganz genau, wo er hin muss. Denn die Adresse ist ganz klar definiert.

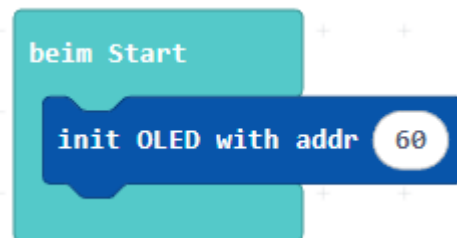
Die I2C-Adresse bei digitalen Bauteilen ist lediglich eine Zahl, die entweder als HEX-Zahl (mit Buchstaben und Ziffern) oder als reine Dezimalzahl angegeben wird.

Die Adresse bei den OLED-Displays in der HEX-Schreibweise ist üblicherweise 0x3C. Das entspricht in der Dezimalschreibweise dem Zahlenwert 60.

Diese Zahlen benötigen wir gleich beim Programmieren des OLED-Displays.

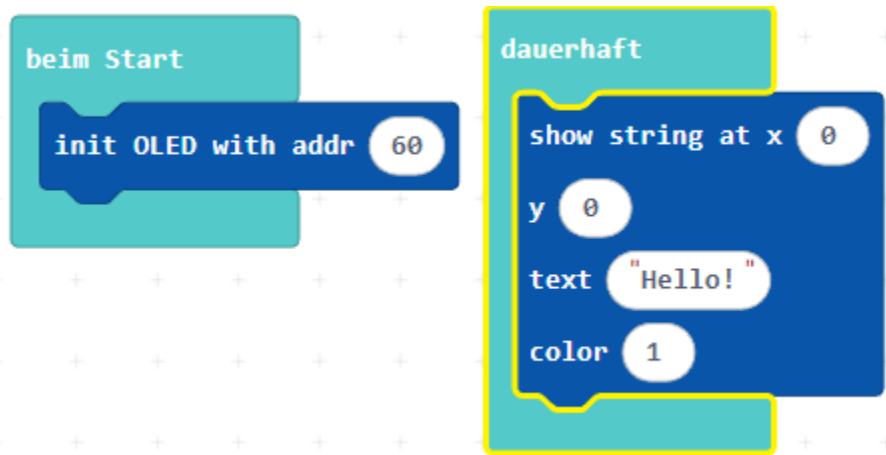
### Programm

In dem Programmblock „**beim Start**“ fügen wir aus der Rubrik „**OLED12864\_I2C**“ den Programmblock „**LCD initialize with Address 39**“ ein.



Die Zahl 60 steht hier für die I2C-Adresse. Das Display wird von nun an vom Calliope angesteuert. Jetzt müssen wir nur noch Daten über die Datenleitung an das Display senden.

Dazu verwenden wir einen weiteren Block aus der Programm rubrik „**OLED12864\_I2C**“. Der Block den wir nun verwenden heißt „**show string**“. Das vollständige Programm sieht jetzt folgendermaßen aus:



Was kann dieser Block? Im ersten und zweiten Feld geben wir die Position auf dem Display an, an dem der Text erscheinen soll und zwar in X-Koordinaten und Y-Koordinaten. Die X-Koordinate verschiebt die Startposition nach links und rechts. Die Y-Koordinate verschiebt die Startposition nach oben und unten. In beiden Feldern lassen wir die „0“ stehen.

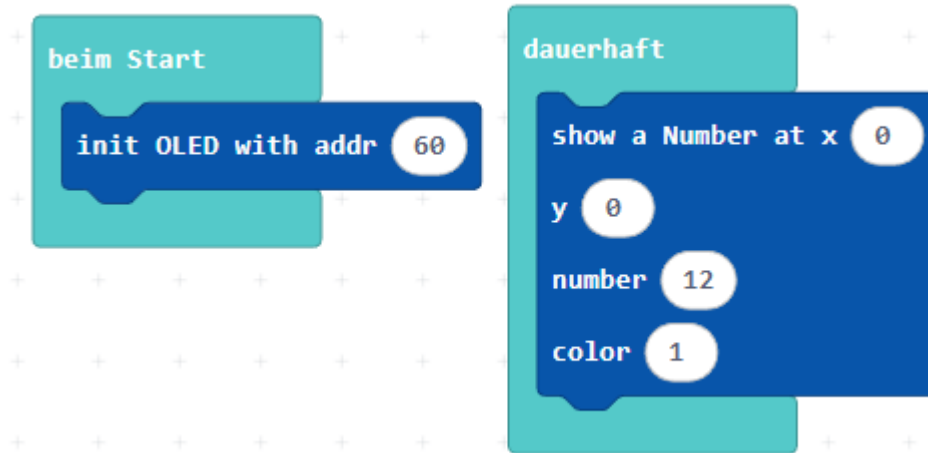
Im dritten Feld tragen wir den Text ein, der auf dem Display angezeigt werden soll. Jetzt steht dort „Hello“. Das Feld „Color“ ist zunächst irrelevant. Hier kann man später eine „0“ eintragen um das Textfeld zu invertieren.

Schreibe nun anstelle des Wortes „Hello“ deinen Namen in das Feld und lade das Programm hoch.

## Teilaufgabe 2 - Ein OLED soll die Temperatur anzeigen

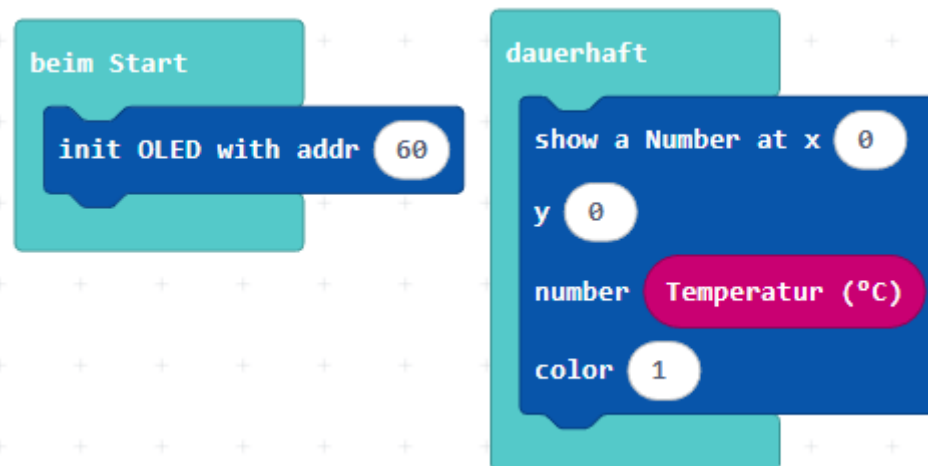
Nun verwenden wir den Temperatursensor des Calliope, und zeigen die Temperatur auf dem Display. Das ist viel einfacher als du vielleicht denkst. Es gibt in der Programmruhrubrik „**OLED12864\_I2C**“ einen Programmblock mit dem Namen „**show a number**“.

Dieser Block zeigt eine Zahl an der angegebenen Stelle. Den Block verwenden wir nun in unserem Programm.



Jetzt würde im OLED die Zahl 12 erscheinen. Daher tauschen wir noch schnell die Zahl zehn gegen den Temperaturmesswert aus. Der Programmblock für die Temperatur befindet sich in der Rubrik „Eingabe“ und heißt „Temperatur (°C)“.

Unser Programm sieht damit folgendermaßen aus.

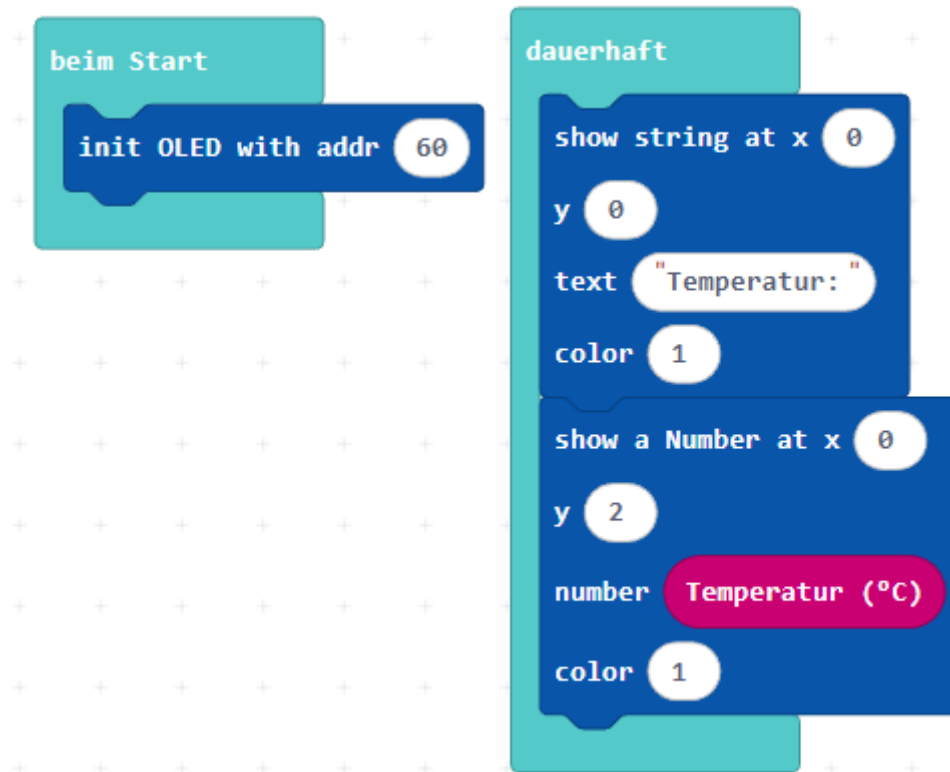


Das Programm laden wir hoch und schon sehen wir die aktuelle Temperatur auf dem OLED-Display.

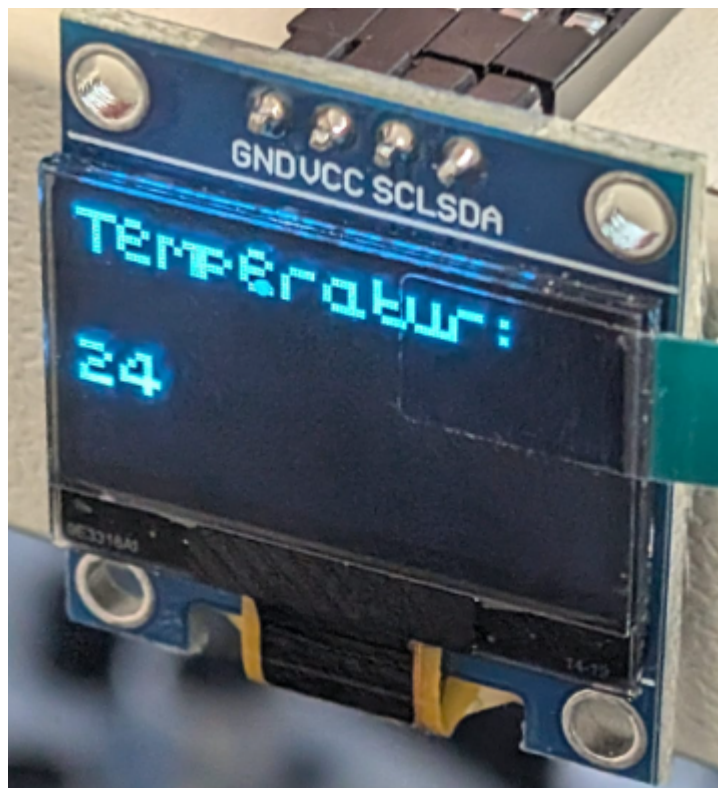
Das sieht ein bisschen langweilig aus. Es sollte schon deutlich zu erkennen sein, dass es sich um die Temperatur handelt. Dazu müssen wir zwei Dinge getrennt darstellen.

Einmal den Text „Temperatur: „ und einmal den gemessenen Zahlenwert.

Wie setzen nun also einen Textblock vor den Zahlenwert. Mit Hilfe der Koordinaten verschieben wir Text und Zahl so, dass sie hintereinander angezeigt werden.



So sieht es gut aus!

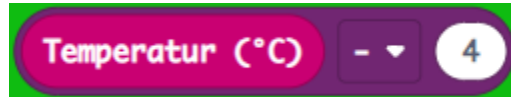


Übrigens – da sich die beiden Programmblöcke im „dauerhaft“ Block befinden, wird die Temperatur natürlich dauerhaft neu abgefragt. Halte deinen Daumen auf den Temperatursensor! Siehst du die Veränderung am Display?

### Ausblick

Uns ist aufgefallen, dass der Temperatursensor etwas zu hohe Werte anzeigt. Mit einem Programmblock aus der Rubrik „Mathematik“ können wir hier etwas verändern.

Was könntest du mit den folgenden Blöcken machen?



## 4.11.2 Ein Display verwenden – LCD

### Aufgabe

Teil 1: Ein LCD soll deinen Namen anzeigen.

Teil 2: Ein LCD soll die Temperatur anzeigen.

### Verkabelung

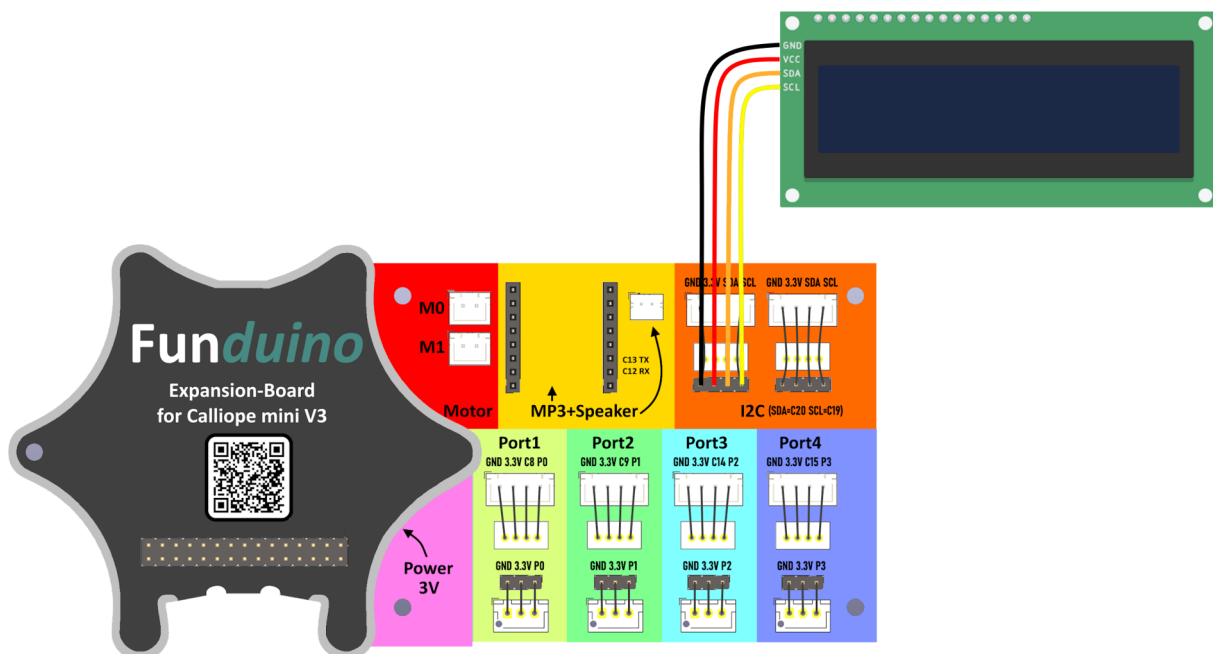
Die Verkabelung ist etwas schwieriger als sonst, da es kein vorkonfektioniertes Kabel gibt. Du musst das Display selber verkabeln.

Das Display erhält seine Daten über die I2C Schnittstelle. Daher verbinden wir das Display mit den Kontakten im orangefarbenen I2C-Bereich des Expanders.

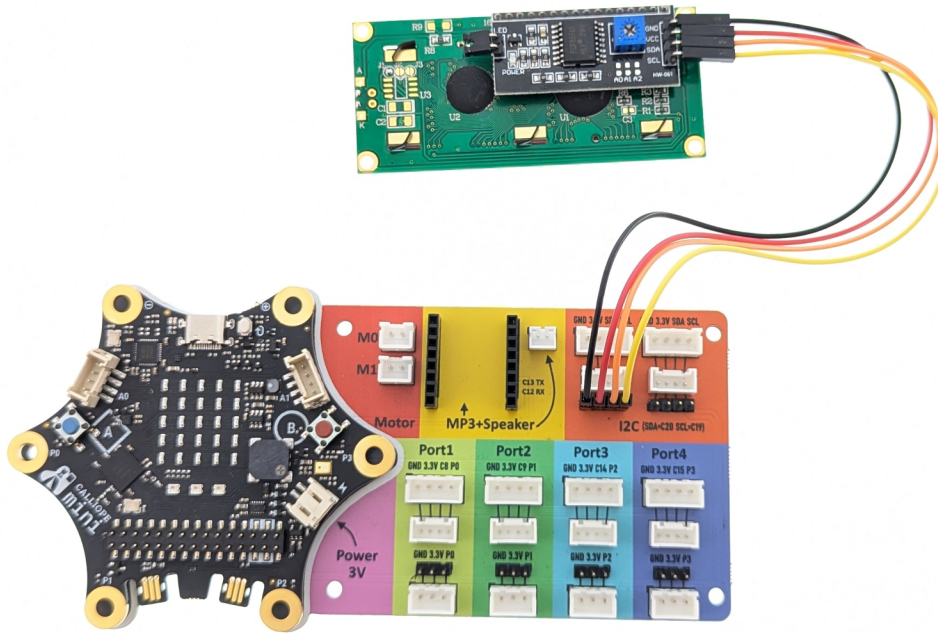
Welchen Steckplatz wir verwenden ist hier irrelevant. Das liegt daran, dass sehr viele I2C Module parallel zueinander, also gleichzeitig, mit einem Mikrocontroller verbunden werden können.

Wir entscheiden uns jetzt für die Kontaktstifte ganz rechts außen im orangefarbenen Bereich und verkabeln der Reihe nach:

1. Kabel: Am Display in den Kontakt GND und die andere Seite am Expander in den Kontakt GND.
2. Kabel: Am Display in den Kontakt VCC und die andere Seite am Expander in den Kontakt 3.3V.
3. Kabel: Am Display in den Kontakt SDA und die andere Seite am Expander in den Kontakt SDA.
4. Kabel: Am Display in den Kontakt SCL und die andere Seite am Expander in den Kontakt SCL.



Richtig verkabelt sieht das am LCD von der Rückseite folgendermaßen aus.



## Programm

Um das Display ansteuern zu können, benötigen wir erneut eine Programmbibliothek. Dazu klicken wir auf die Rubrik „**Erweiterungen**“ und geben in das Suchfeld „**LCD1602**“ ein.

Übrigens – die Bezeichnung 1602 steht für die Anzahl der Zeichen pro Reihe und die Anzahl der Reihen. Unser Display hat also zwei Reihen mit je 16 Zeichen.

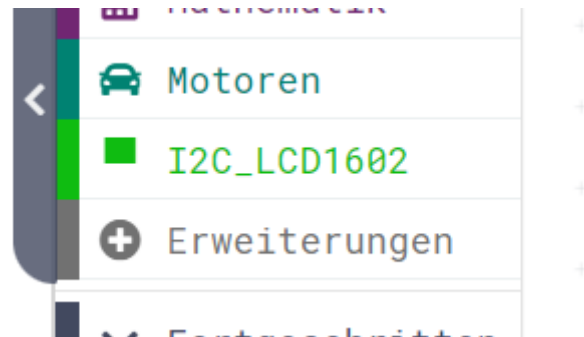
Die folgende Programmbibliothek erscheint.



**i2cLCD1602**  
makecode I2C LCD1602  
package for microbit

[Learn More](#)

In den Programmblöcken erscheint eine neue Rubrik mit dem Namen „I2C\_LCD1602“. Hier finden wir nun alle Funktionen rund um das Display.



## Teilaufgabe 1 - Ein LCD soll deinen Namen anzeigen

Das Display wird mit der sogenannten I2C Datenleitung mit dem Calliope verbunden. Wir haben bereits erfahren, dass man bei dieser Art der Verkabelung ganz viele I2C-Fähige Module gleichzeitig anschließen kann. Aber woher weiß der Calliope, an welches der angeschlossenen Module die Daten gesendet werden sollen?

Dazu hat jedes I2C-Modul, also auch unser Display, eine eigene Adresse. Das ist genauso wie die Adresse von dem Haus in dem du lebst. Der Postbote weiß ganz genau, wo er hin muss. Denn die Adresse ist ganz klar definiert.

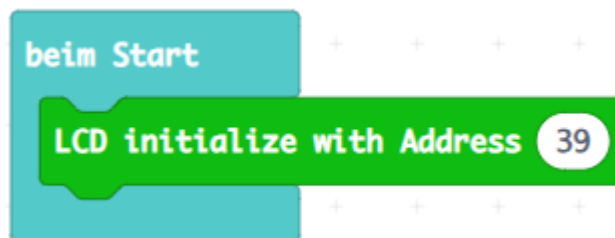
Die I2C-Adresse bei digitalen Bauteilen ist lediglich eine Zahl, die entweder als HEX-Zahl (mit Buchstaben und Ziffern) oder als reine Dezimalzahl angegeben wird.

Die Adresse bei den Displays in der HEX-Schreibweise ist üblicherweise 0x3F oder 0x27. Das entspricht in der Dezimalschreibweise den Zahlen 39 oder 63.

Diese Zahlen benötigen wir gleich beim Programmieren des Displays.

### Programm

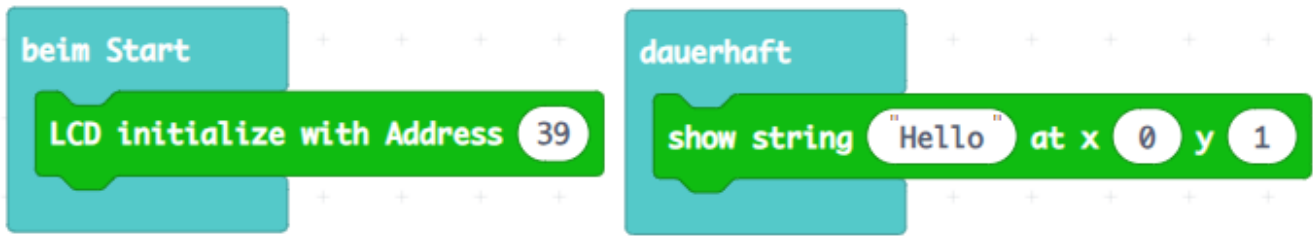
In dem Programmblock „**beim Start**“ fügen wir aus der Rubrik „**I2C\_LCD1602**“ den Programmblock „**LCD initialize with Address 39**“ ein.



Die Zahl 39 steht hier für die I2C-Adresse. **Die 39 ist der übliche Fall.** Nur wenn es später nicht funktioniert, kann man es hier alternativ mit der Zahl 63 versuchen.

Das Display wird von nun an vom Calliope angesteuert. Jetzt müssen wir nur noch Daten über die Datenleitung an das Display senden.

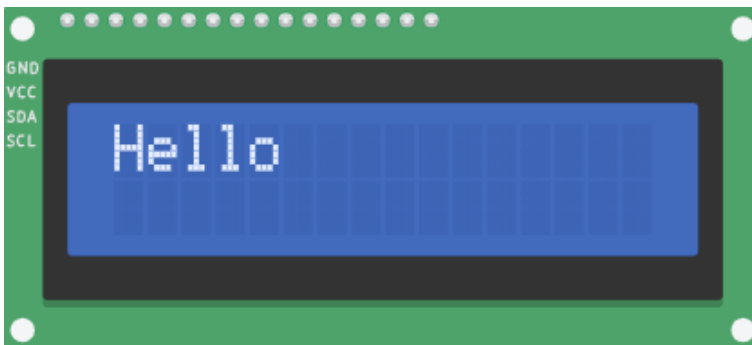
Dazu verwenden wir einen weiteren Block aus der Programmrubrik „**I2C\_LCD1602**“. Der Block den wir nun verwenden heißt „**show string „Hello“ at c0 y0**“. Das vollständige Programm sieht folgendermaßen aus:



Was kann dieser Block? Im ersten Feld tragen wir den Text ein, der auf dem Display angezeigt werden soll. Jetzt steht dort „Hello“. Im zweiten und dritten Feld geben wir die Position auf dem LCD an, an dem der Text erscheinen soll und zwar in X-Koordinaten und Y-Koordinaten. Die X-Koordinate verschiebt die Startposition nach links und rechts. Die Y-Koordinate entscheidet über obere und untere Zeile, wobei 0 die erste Zeile ist und 1 die zweite Zeile ist.

### Beispiele für Startpositionen

X:0 Y:0 → Die Startposition ist in der oberen Zeile, und beginnt an Position 1.



X:4 Y:0 → Die Startposition ist in der oberen Zeile, und beginnt an Position 5.



X:0 Y:1 → Die Startposition ist in der unteren Zeile, und beginnt an Position 1.



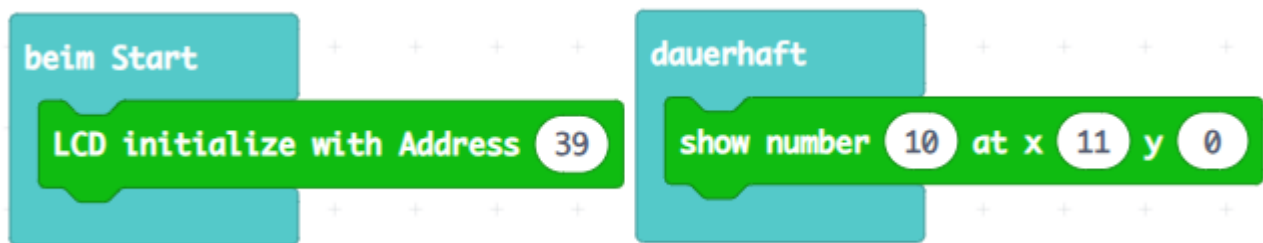
X:7 Y:1 → Die Startposition ist in der unteren Zeile, und beginnt an Position 8.



## Teilaufgabe 2 - Ein LCD soll die Temperatur anzeigen

Nun verwenden wir dem Temperatursensor des Calliope, und zeigen die Temperatur auf dem Display. Das ist viel einfacher als du vielleicht denkst. Es gibt in der Programmrubrik „I2C\_LCD1603“ einen Programmblock mit dem Namen „**show number 10 at x0 y0**“.

Dieser Block zeigt eine Zahl an der angegebenen Stelle. Den Block verwenden wir nun in unserem Programm.

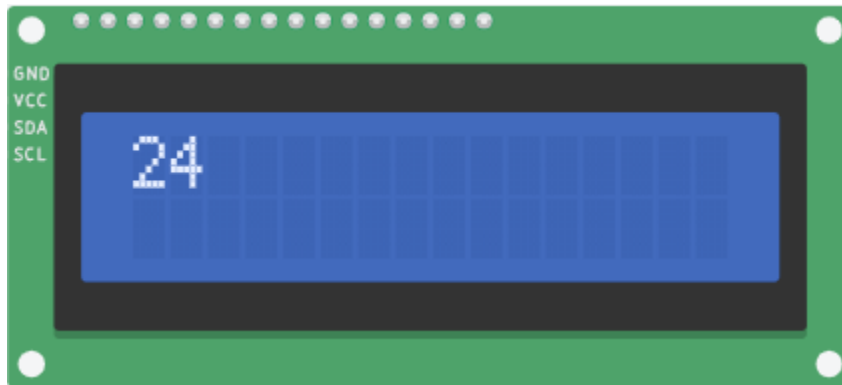


Jetzt würde im LCD die Zahl 10 erscheinen. Daher tauschen wir noch schnell die Zahl zehn gegen den Temperaturmesswert aus. Der Programmblock für die Temperatur befindet sich in der Rubrik „Eingabe“ und heißt „Temperatur (°C)“.

Unser Programm sieht damit folgendermaßen aus.

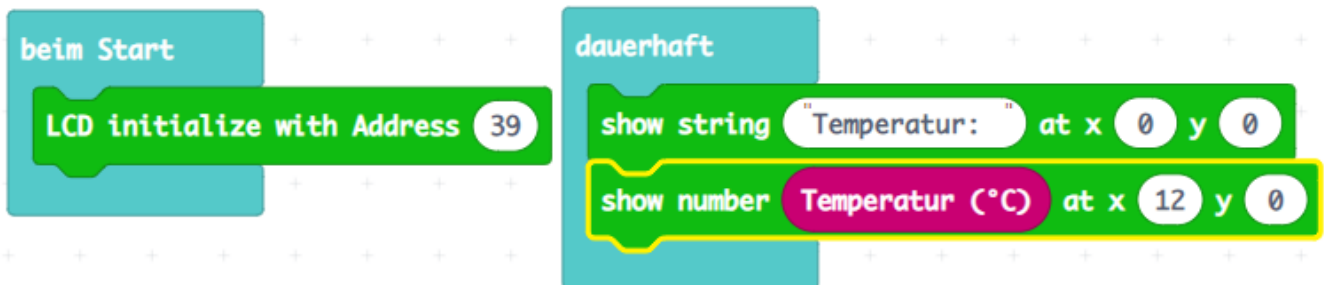


Das Programm laden wir hoch und sehen das Ergebnis.



Das sieht ein bisschen langweilig aus. Es sollte schon deutlich zu erkennen sein, dass es sich um die Temperatur handelt. Dazu müssen wir zwei Dinge getrennt darstellen. Einmal den Text „Temperatur:“, und einmal den gemessenen Zahlenwert.

Wie setzen nun also einen Textblock vor den Zahlenwert. Mit Hilfe der Koordinaten verschieben wir Text und Zahl so, dass sie hintereinander angezeigt werden.



So sieht es gut aus!

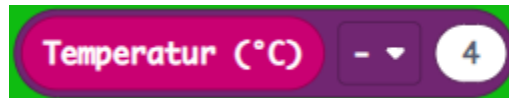


Übrigens – da sich die beiden Programmblöcke im „dauerhaft“ Block befinden, wird die Temperatur natürlich dauerhaft neu abgefragt. Halte deinen Daumen auf den Temperatursensor! Siehst du die Veränderung am Display?

### Ausblick

Uns ist aufgefallen, dass der Temperatursensor etwas zu hohe Werte anzeigt. Mit einem Programmblock aus der Rubrik „Mathematik“ können wir hier etwas verändern.

Was könntest du mit den folgenden Blöcken machen?



## 4.12 Ende oder Anfang?

Du hast eine Menge über Elektronik und dem Programmieren gelernt. War's das jetzt?

**Nein!**

Du befindest dich jetzt in einer sehr guten Ausgangsposition! Du bist an einem Punkt, an dem du grundlegende Programmierstrukturen des block-basierten Programmierens verstanden hast.

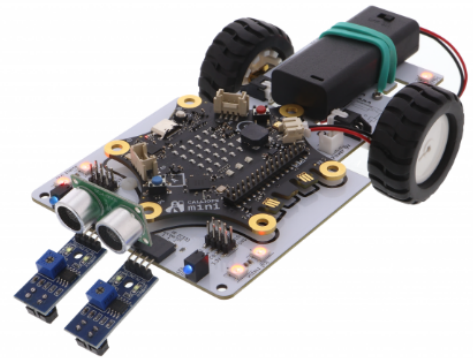
## Wie geht es weiter?

Weiter mit **Calliope**?

Es gibt viele Wege, die du nun gehen kannst. Zum einen steckt noch viel Potenzial in der Arbeit mit dem Calliope Mikrocontroller. Genau wie in den Beispielen aus dieser Anleitung gibt es hunderte weitere Module, die mit dem Calliope blockbasiert programmiert werden können.

Weiter mit dem **Bau eines Roboters**?

Roboterfahrzeuge wie der „Calle für Calliope Mini v3“ können programmiert werden, um Linien zu verfolgen, Gegenstände zu umfahren oder einfach nur zu leuchten und zu piepsen. Für dich kein Problem, Calliope programmieren kannst du ja schon.



Weiter mit **Arduino**?

Ein anderer Weg wäre die **Fortführung** und **Weiterentwicklung** des Lernprozesses mit Arduino. Neben der blockbasierten Programmierung gibt es hier einige Grundlegende Dinge, die dich im Bereich der Programmierung große Fortschritte machen lassen.

Dazu gehören:

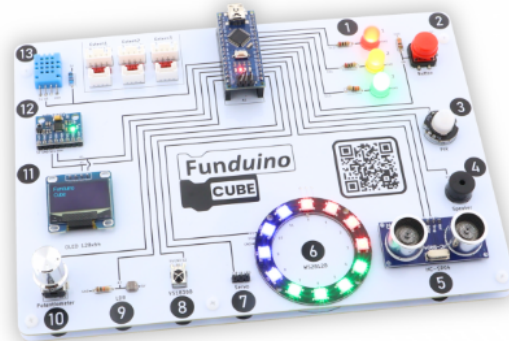
- Blockbasiertes programmieren mit erweiterten Parametern (ohne vorkonfektionierte Module)
- Programmieren in Hochsprache (Arduino IDE)
- Physical Computing und Prototyping mit erweitertem Verständnis für Technik und Elektronik.



- Aufbau von individuellen Kombinationen aus Mikroelektronik und Elektronik
- Programmieren im Team mit künstlicher Intelligenz (Arduino mit ChatGPT)

Ein leichter Einstieg in den Arduino-Bereich wäre die Programmierung des **Funduino-Cube**.

Weiter mit **ESP32**, **Raspberry Pi** oder Chips von **Infineon**?



Es gibt sehr viele Möglichkeiten. Bleibe interessiert und trau dich, neue Themen zu entdecken!