

# Calliope-Kurs Kinder

06\_Tag4

---

Jogi Künstner, Turbine Brunnen

Herbst 2020



# **Kurs Programmieren Lernen mit Calliope Mini**

---

Heute wiederholen wir als erstes nochmal kurz das Gelernte vom letzten Nachmittag,  
Wir schauen uns nochmal an, was das mit Logik, Wahrheit, Wahr oder Falsch auf sich  
hat. Und damit natürlich auch, wie das bei Entscheidungen genutzt werden kann.

Also : **Wenn** xxx, **Dann** yyy



Anschliessend schauen wir uns auch nochmal ein wichtiges Kapitel für die Software-Entwicklung, das Programmieren an:

Die **Schleife** , neben der **Wenn-Dann** - Konstruktion sicherlich das zweitwichtigste Element beim Programmieren!

Dazu schauen wir uns an, wo uns die Schleifen-Programmierung das Programmierer-Leben erleichtert

Daraufhin schauen wir uns kurz die allgemeine Funktion eines Motors an, sehen was es für verschiedene Arten von Elektro-Motoren gibt.

Und dann fangen wir mit der Programmierung eines DC-Motors an, sehen uns die Möglichkeiten des Calliope ein oder zwei Motoren zu steuern und versuchen dann auf Tastendrücke zu reagieren.



- 01 Auffrischen: Logik und Wenn-Dann
- 02 Schleifen-Programmierung
- 03 Motoren Übersicht
- 04 Ansteuerung von DC-Motoren
- 05 Lagesensor



# 06\_01\_Auffrischen

Calliope-Kurs Kinder

---

Jogi Künstner, Turbine Brunnen

Herbst 2020



# Auffrischen

---

- Variablen genauer angeschaut,
- vor allem den Unterschied zwischen Belegung einer Variablen und Benutzung einer Variablen,
- sowohl vom Verständnis als auch in unserer Programmier-Oberfläche
- für Details bitte hier schauen





- Wir haben uns auch nochmals das Vorgehen zur LED-Ampel genauer angeschaut,
- vor allem die Umwandlung von einer Ampel-Schaltung mit drei Schaltern
- in eine Ampel-Schaltung mit Calliope und Software
- für Details bitte hier schauen



- Das alles sind sogenannte **logische** Entscheidungen.
- Dazu gibt es den Begriff des Wahrheitswertes :
  - Wahr
  - Falsch
- Weiss/Nicht oder Vielleicht, gibt es da nicht, denn dann weiss der Computer/Calliope nicht, was er tun soll
- **Logik** : **Wenn** etwas eintreffen wird, **Dann** wird etwas bestimmtes gemacht
- **Ansonsten** kann auch etwas anderes gemacht werden



- **Wenn-Dann** kann erweitert werden:
  - **wenn dann**
  - **ansonsten wenn**
  - **ansonsten wenn**
  - **ansonsten wenn**
  - **ansonsten**
- für Details bitte hier schauen



- Auslesen des Temperatur-Sensors, Speichern des aktuellen Temperatur-Wertes in einer Variablen
- Entscheidung (siehe oben) etwas zu tun, wenn eine bestimmte Temperatur überschritten wird.
- für Details bitte hier schauen



- Ansteuern der RGB-LED mit einfachen Farben
- Ansteuern geht auch die einzelnen Farbanteile selbst mischen
- Dazu kann man auch die Schleifen verwenden, die lernen wir heute kennen.
- für Details bitte hier schauen, erst in der zweiten Hälfte



- Umwandeln der Temperatur-Mess-Anlage mit LED
- in eine Licht-Mess-Anlage mit Ton-Ausgabe als Schubladen-Alarm-Anlage
- für Details bitte hier schauen



# 06\_02\_Schleifen

Calliope-Kurs Kinder

---

Jogi Künstner, Turbine Brunnen

Herbst 2020



# Schleifen-Programmierung

---



# Was sind Schleifen

- **Frage** : Wofür braucht man Schleifen?
- **Antwort** : Immer dann, wenn man etwas gleiches wiederholen will!

Beispiel : Man möchte beim Einschalten 5 mal ein Gesicht blinken lassen

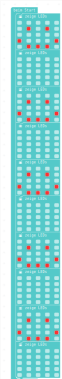
Mit den uns bekannten Möglichkeiten:

- **beim Start**-Block holen
- Gesicht malen
- LED-Bildschirm löschen
- Gesicht malen
- LED-Bildschirm löschen
- usw usw ... (alles in **Grundlagen**)



# Beispiel 1 : Ohne Schleife

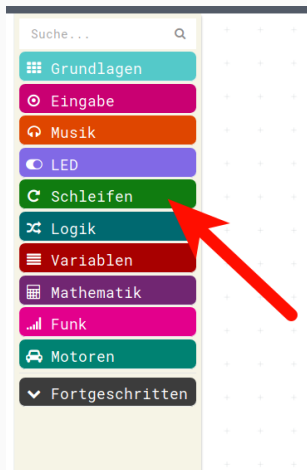
Beim Starten 5 mal ein Gesicht blinken lassen



# Beispiel 1 : Mit Schleife (1)

Wo finden wir Schleifen ?

Hauptmenu:



# Beispiel 1 : Mit Schleife (2)

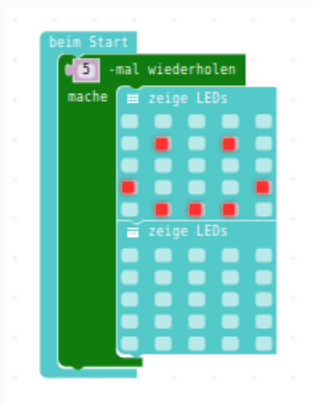
Hier finden wir verschiedene Schleifen.

Wir interessieren uns zuerstmal für die erste Variante

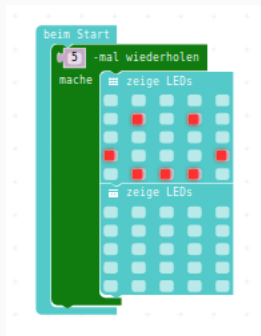
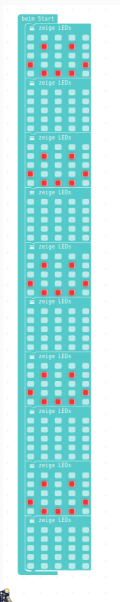


## Beispiel 1 : Mit Schleife (3)

- Wir ziehen diese **4 mal wiederholen** - Schleife in den **beim Start**-Block
- dann überschreiben wir die 4 mit einer 5
- und ziehen ein Gesicht und ein Löschen in den **mach**-Teil der Schleife
- Fertig...



# Beispiel 1 : Vergleich



- Welches Programm sieht kompakter aus?
- Welches Programm ist einfacher zu verstehen?
- Welches Programm ist einfacher zu warten, zu ändern?

Damit kommen wir zu Beispiel 2:



## Beispiel 2 : Beim Starten 5 mal ein Herz anzeigen

Nun kommt unser **Auftrag-Geber** von **Programm 1** und sagt:

- Ach, ich wollte doch lieber ein Herz blinken haben,
- und das wenn möglich 6 mal.

Nun wollen wir die beiden Varianten aus Beispiel 1 nehmen und entsprechend verändern.





## Beispiel 2 : Ohne Schleife

Wieviele Änderungen müssen wir machen, wie oft mit der Maus klicken, um aus Programm 1 das Programm 2 zu machen.

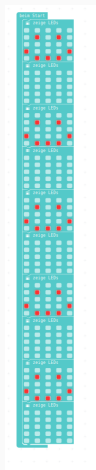


Figure 1: Von hier

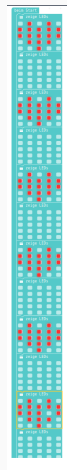


Figure 2: nach hier



## Beispiel 2 : Mit Schleife

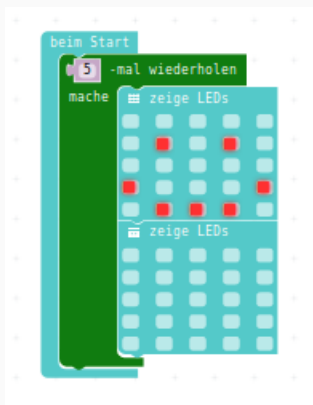


Figure 3: von hier

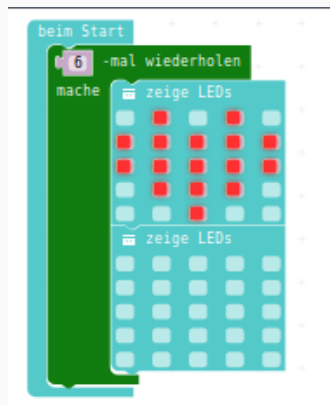


Figure 4: nach hier



## Beispiel 2 : Auswertung

- Welches Programm sieht kompakter aus?
- Welches Programm ist einfacher zu verstehen?
- Welches Programm ist einfacher zu warten, zu ändern?
- Welches Programm ist fehler-anfälliger?

**MERKE** : Sobald man anfängt, beim Software-Programmieren etwas zu kopieren, muss man darüber nachdenken, ob man das mit einer Schleife den Computer erledigen lassen könnte.



- **Frage** : Wofür braucht man Schleifen?
- **Antwort 2** : Immer dann, wenn man etwas sehr ähnliches wiederholen will, wobei sich dabei bestimmte Dinge ändern können, die vom Schleifendurchlauf abhängen.
  - Also beim **ersten** Schleifendurchlauf wird etwas mit einer **1** gemacht
  - Beim **zweiten** Durchlauf wird etwas mit einer **2** gemacht
  - usw. usw.
- Wir wollen nun einen Zähler bauen.



## Beispiel 3 : Schleife mit Zähler

Nun wollen wir innerhalb des sogenannten “Schleifenkörpers” die Anzahl der Schleifen-Durchgänge anzeigen.

- Dazu benutzen wir die gerade vorhandene Schleife,
- legen **VOR** der Schleife eine Variable namens **SchleifenZaehler** an,
- diese belegen wir mit 0.



## Beispiel 3 : Schleife mit Zähler

Im Schleifenkörper lassen wir uns den Wert dieser Variable anzeigen (mit “Zeige Nummer”) und erhöhen anschliessend die Variable/den Zähler.

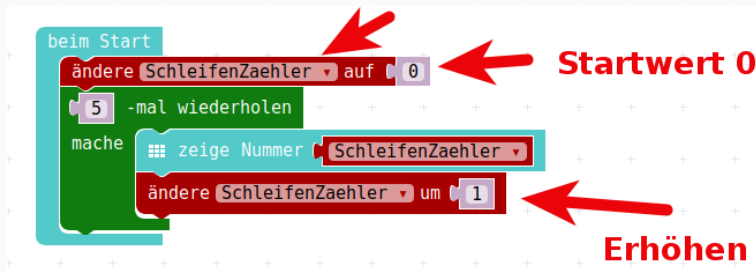


Figure 5: Schleife mit manuellem Zähler

Da wir den Zähler mit 0 vorbelegen und die Schleife 5 mal läuft, bekommen wir durch dieses Programm die Zahlen 0 bis 4 angezeigt.



## Beispiel 4 : Schleife mit eingebautem Zähler

Diese Art der Schleife wird sehr oft gebraucht:

eine Schleife, die eine bestimmte Anzahl von Durchläufen erlaubt und bei der man die Schleifendurchläufe mitzählt.

Darum gibt es dafür ein extra Programmier-Konstrukt.

Das ist die sogenannte **Index-For-Schleife** .



## Beispiel 4 : Schleife mit eingebautem Zähler

Die Index-For-Schleife finden wir ebenso im Menu Schleifen:

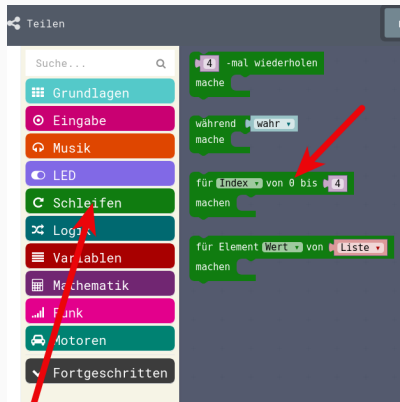


Figure 6: Schleifen-Menu





## Beispiel 4 : Schleife mit eingebautem Zähler

Wenn wir diese Schleife benutzen und unser Programm entsprechend umgestalten, sieht es nochmal um einiges einfacher aus:

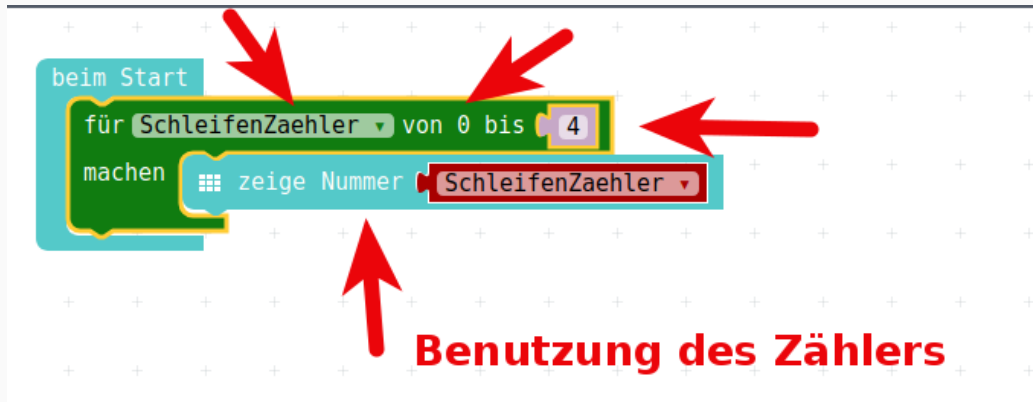
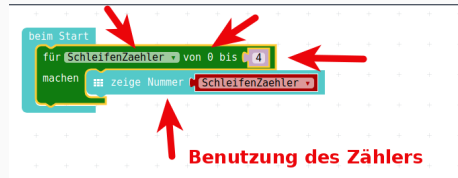
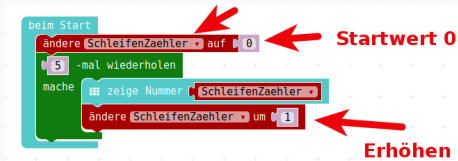


Figure 7: Schleife mit IndexZähler



# Vergleich der beiden Schleifen



# 06\_03\_Motoren

Calliope-Kurs Kinder

---

Jogi Künstner, Turbine Brunnen

Herbst 2020



# Elektro-Motoren

---

Es gibt sehr viele verschiedene Motoren:

- Dieselmotoren
- Benzin-Motoren
- Düsen-Antriebs-Motoren
- ...

Um die alle wollen wir uns heute sicher **NICHT** kümmern...



Sondern heute wollen wir - wie oben geschrieben - Elektro-Motoren anschauen. Aber auch bei den Elektro-Motoren gibt es sehr grosse Unterschiede, sowohl was deren elektrische Spannung als auch was deren Funktionsweise anbelangt.



Folgende Arten von Elektro-Motoren fallen mir auf Anhieb ein:

- DC-Motoren
- Schritt-Motoren
- Servo-Motoren
- DC-Motoren mit Getriebe
- und noch mehr
- ...



Grundprinzip ist im Allgemeinen immer ein Elektro-Magnet, der einen Permanent-Magneten anzieht und in dem Moment, wenn der Permanent-Magnet “angekommen” ist, wird der Elektro-Magnet ausgeschaltet und dann der nächste Elektro-Magnet ein Stück weiter in der Drehung eingeschaltet.

(Achtung: Das ist nur das Prinzip. Im Detail sieht das meist etwas anders aus, aber so kann man es sich am ehesten bildlich vortellen)





So kann man sich das vorstellen:

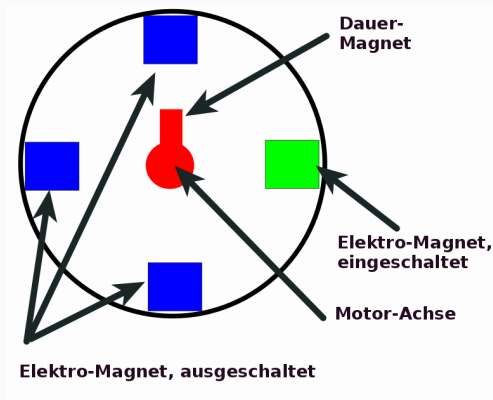


Figure 1: Prinzip-Bild Motor

- Der eingeschaltete Elektro-Magnet zieht den Dauer-Magneten an
- Der Dauer-Magnet bewegt sich in Richtung Elektro-Magnet
- Die Achse dreht sich mit
- Wenn der Dauer-Magnet den Elektro-Magnet erreicht hat, schaltet sich der Elektro-Magnet ab
- Der Elektro-Magnet eine viertels Umdrehung ( $90^\circ$ ) weiter schaltet sich ein
- Er zieht den Dauer-Magneten an
- usw. . . . usw.



# Animierter Motor Langsam

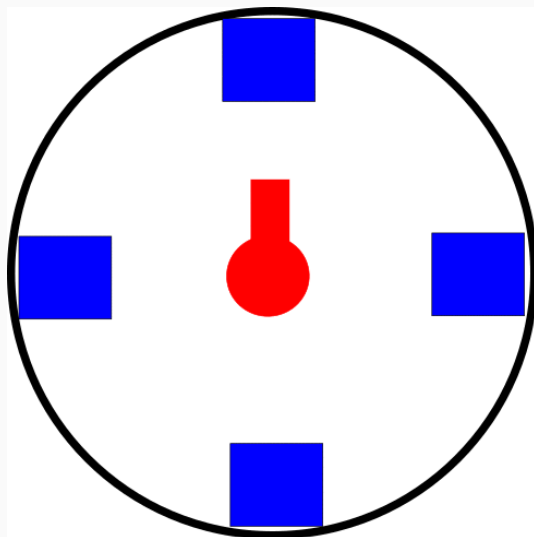
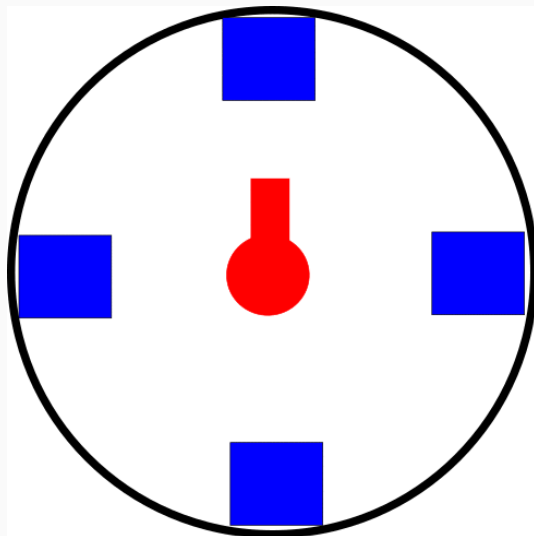


Figure 2: Animierter Motor langsam

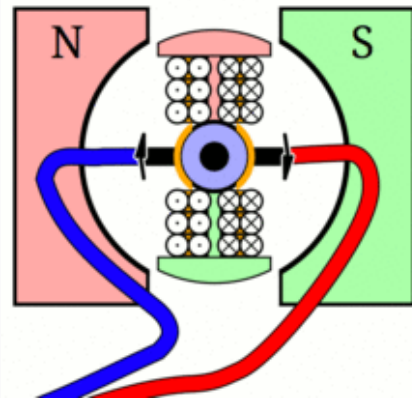
# Animierter Motor Schnell

Und wenn das ganze etwas schneller abläuft, dann sieht das so aus:



# Realistischere Animation

Wie geschrieben, das zeigt nur das Prinzip.



Dieses Model kommt der Wirklichkeit näher:

( <https://de.wikipedia.org/wiki/Elektromotor> , MichaelFrey, CC BY-SA 3.0 )

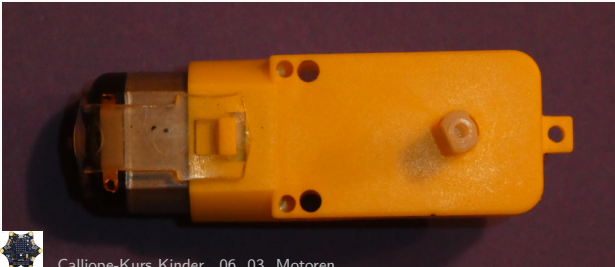
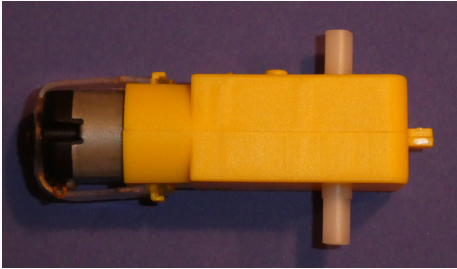


**DC-Motoren** benutzen dieses Prinzip um einen Motor einfach ständig komplett um seine Achse drehen zu lassen. Dabei wird durch einen sehr einfachen Mechanismus dafür gesorgt, dass die jeweiligen Magneten ständig weiter-geschaltet werden. Im Allgemeinen kann man (im Bereich wie es für diesen Motor festgelegt ist) sagen: Höhere Spannung  $\Rightarrow$  Schnellere Drehung. Wie weit sich der Motor dreht, kann normalerweise nicht haargenau vorherbestimmt werden.



# Robotik-Motor

Das hier ist ein "Standard"-Robotik-Motor mit eingebautem Getriebe, der Motor selbst ist ein DC-Motor.



# Lüfter-Motor

Das hier ist ein Lüfter-Motor, wie er im Computerbau verwendet wird, auch das ist ein normaler DC-Motor.



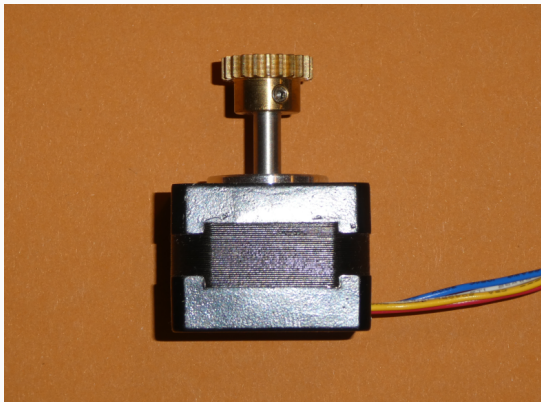


Bei **Schritt-Motoren** hat man das normalerweise selbst haargenau im Griff, man kann entweder durch Elektronik oder durch genaue Programmierung genau einzelne Schritte bestimmen. Da wird zum Beispiel ein Schritt-Motor mit 36 Schritten pro Umdrehung hergestellt, d.h. jeder Schritt dreht den Motor um  $10^\circ$  weiter und nach 36 Schritten hat er genau eine komplette Umdrehung gemacht. Die einzelnen Schritte müssen gesteuert werden.



# Schritt-Motoren

Das hier ist ein Schritt-Motor, er hat vier Anschlüsse:



Wenn man den Schritt-Motor von Hand dreht, kann man die einzelnen Schritte auch spüren.



**Servo-Motoren** hingegen haben eine ganz andere Funktion:

Sie können sich normalerweise nur ca eine halbe Umdrehung umdrehen. Sie werden üblicherweise mit Gradzahlen angesteuert.

So kann man einen Servomotor von 0 - 180 Grad ( Eine volle Umdreheung sind immer 360 Grad) ansteuern, normalerweise ist die Null-Stellung bei 90 Grad, man kann also einen Servomotor um eine viertels Drehung nach links auf 0 und um eine viertels Drehung nach rechts auf 180 Grad bewegen.

Im Modellbau werden Servos sehr oft benutzt, z.B. um zu lenken, um ein Höhenruder beim Flugzeug zu verstellen.

Unser Calliope-Männchen hier in der Turbine nutzt einen Servo-Motor um zu winken.



Das hier ist ein Servo-Motor, er hat drei Anschlüsse,



Figure 4: 05\_ServoMotor

- einmal Plus,
- einmal Minus
- und einmal den Ansteuer Anschluss

Wir schauen uns heute die Ansteuerung von normalen DC-Motoren (mit Getriebe) an.



# 06\_04\_DC\_Motoren

Calliope-Kurs Kinder

---

Jogi Künstner, Turbine Brunnen

Herbst 2020



# DC-Motoren

---

DC steht für **D**irect **C**urrent und heisst Gleichstrom. ( AC steht für **A**lternate **C**urrent und heisst Wechselstrom. Die australische Musikgruppe AC/DC heisst also eigentlich Wechselstrom/Gleichstrom)

Ein **DC**-Motor funktioniert prinzipiell wie gerade beschrieben :

- Ein elektrischer Magnet zieht einen Permanent-Magnet an.
- Sobald der Permanent-Magnet in die Nähe des elektrischen Magneten kommt, schaltet dieser ab und der etwas weiter entfernte Elektro-Magnet wird eingeschaltet.
- So geht es in einem fort, immer im Kreis herum.

Der DC-Motor läuft einfach mit einer Batterie, je grösser die Spannung um so schneller dreht sich der Motor.

Er darf aber nicht mehr Spannung bekommen, als er verträgt!





# Motor ausprobieren

Nun probieren wir einfach mal direkt den Motor an eine Batterie anzuschliessen.

So sieht das dann als Schaltbild aus:

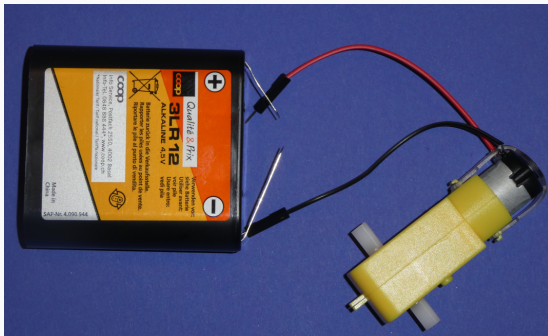


**Figure 1:** Motor-Schaltbild



# Batterie ranhalten

Wir halten hier die Kabel nur an die Batterie ran, wir wollen nur wissen ob das so funktioniert. **Achtung** : Wie wir am zweiten Nachmittag gelernt haben, ist elektrischer Strom gefährlich. Wir haben aber auch gelernt, dass normalerweise eine Batterie mit 4.5 V uns nicht gefährlich werden kann. Darum können wir das einfach zusammenhalten!



# Motor-Anschluss an den Calliope

Der Calliope hat eine zusätzliche elektrische Schaltung mit auf der Platine, die den Anschluss eines Motors überhaupt erlaubt.

Ein Motor braucht so viel Leistung, dass man den normalerweise nicht direkt an den Mikro-Prozessoren (das sind die “Gehirne” bei unseren Calliopes und anderen Bastel-Platinen) anschliessen darf.

Der Calliope hat aber die notwendigen “Motor-Treiber” gleich mit eingebaut. Die Schaltung erlaubt sogar den zusätzlichen Anschluss einer stärkeren Batterie (Achtung: bis zu **9V**) so dass man mit dem Calliope sogar Motoren mit bis zu 9V anschliessen und ansteuern kann.



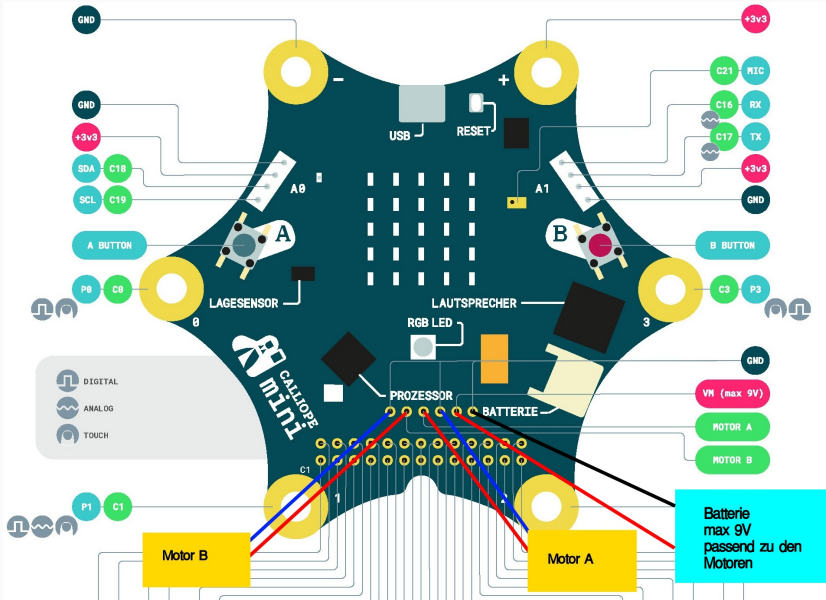
Je nach Verwendungszweck kann man an den Calliope entweder

- 1 Motor anschliessen, der kann dann vorwärts und rückwärts drehen
- 2 Motoren anschliessen, die können dann nur einzeln vorwärts drehen

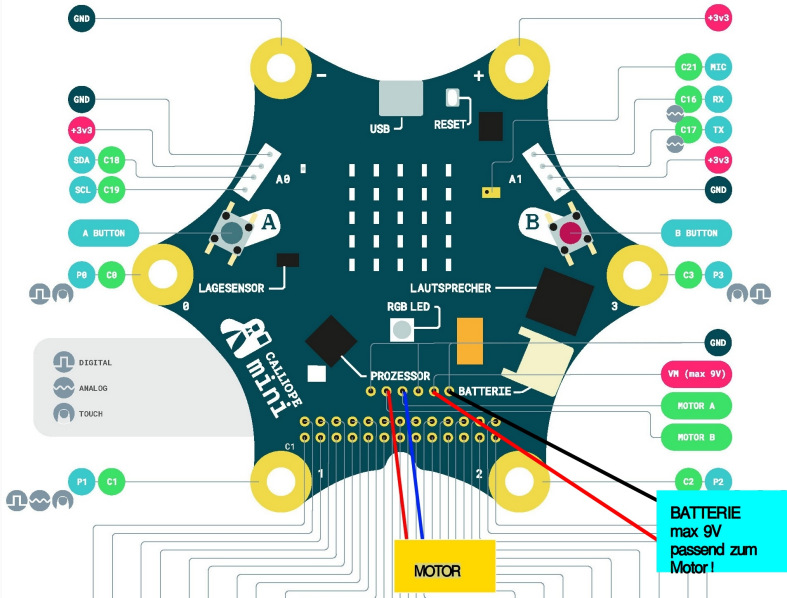
So sehen die beiden Möglichkeiten zum Anschluss von zwei oder einem Motor aus:



# Zwei Motoren



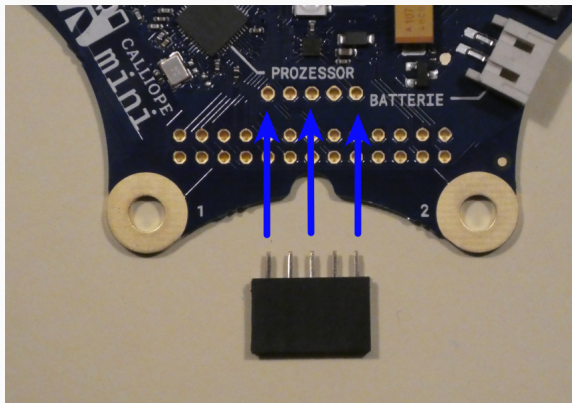
# Ein Motor



# Pfostenstecker

Um dies zu ermöglichen, müssen wir nun zuerst einmal Pfostenstecker oder Pfostenbuchsen an den Calliope löten.

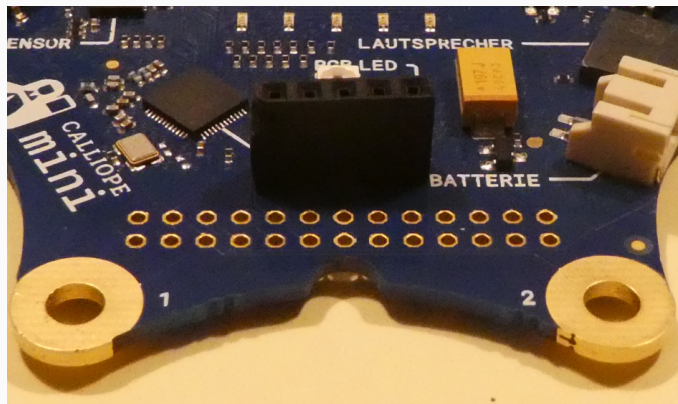
So sieht das aus:



**Figure 2:** Pfostenstecker



# Pfostenstecker aufgelötet



**Figure 3:** Pfostenstecker aufgelötet



Nachdem wir nun wissen, wie wir einen einzelnen Motor an den Calliope elektrisch anschliessen, wollen wir den Ausgang für den Motor auch mit Software programmieren. Im ersten Schritt wollen wir nur ganz einfach den Motor ein- und ausschalten können. Dazu wollen wir mit dem linken Knopf ein und mit dem rechten Knopf ausschalten.



Die Motor-Ansteuerung findet sich im Menu Motoren:

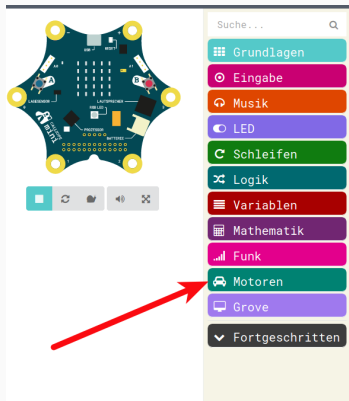


Figure 4: Menu Motor

Es gibt nicht viele Befehle zum Steuern von Motoren:

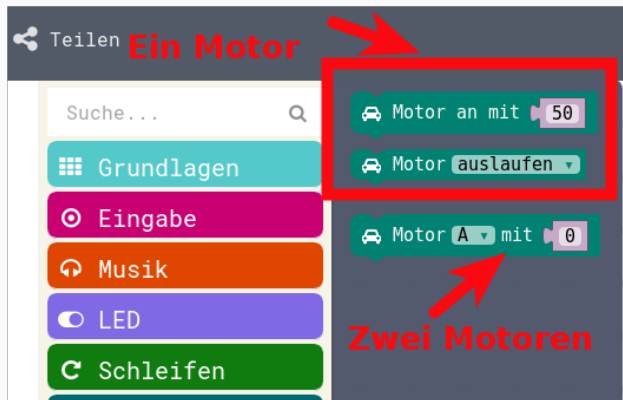


Figure 5: Motor Befehle

Wie oben beschrieben, gibt es die Möglichkeit entweder zwei Motoren anzuschliessen, bei zwei Motoren kann man dann

- Den Motor A oder B auswählen
- Die Geschwindigkeit von 0 - 100 angeben.

Bei Anschluss von nur einem Motor kann man

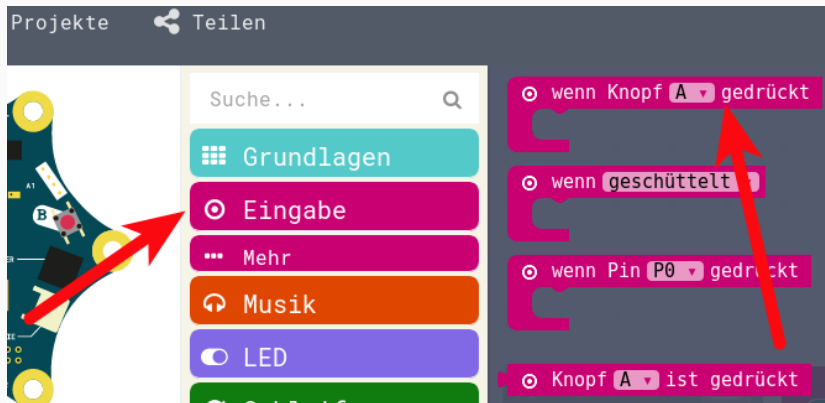
- Eine Geschwindigkeit von -100 bis +100 eingeben
- Bremsen oder auf Leerlauf schalten



# Knopfdruck einschalten

Wir wollen nun - wie beschrieben - nur den einen Motor entweder mit 100 einschalten oder beim Druck des rechten Knopfes mit 0 einschalten, was so viel wie Ausschalten heisst. . .

Mit den Befehlen zum Auswerten der Knöpfe :



Und den Befehlen zum Ansteuern des Motors :

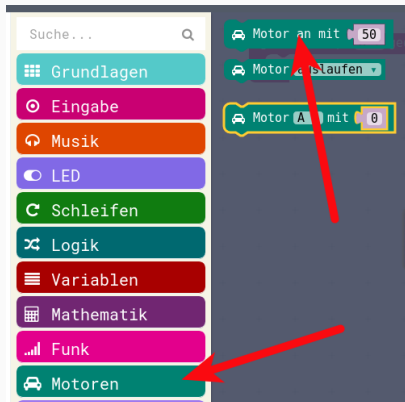


Figure 7: Motor Befehle

# Erstes Motor-Programm

Ergibt das unser erstes Motor-Programm

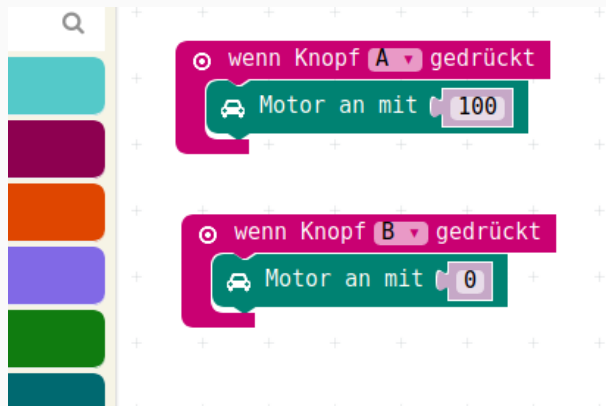


Figure 8: Motor Programm 1

Diese Programm können wir leider im Simulator gar nicht nutzen.

  müssen wir das Programm auf den Calliope runterladen und dort ausprobieren.

## Java-Script-Code

```
input.onButtonPressed(Button.A, () => {  
  motors.motorPower(100)  
})  
input.onButtonPressed(Button.B, () => {  
  motors.motorPower(0)  
})
```

## Download Hex-Code

Hex-code





Da wir nun einen Motor angeschlossen haben, können wir nun auch versuchen, den Motor in die andere Richtung laufen zu lassen.

Dazu brauchen wir einen dritten Informations-Eingang, neben Knopf A und Knopf B.

Wir können ja zum Beispiel das Ereignis Knopf A **UND** Knopf B gleichzeitig gedrückt auswerten.

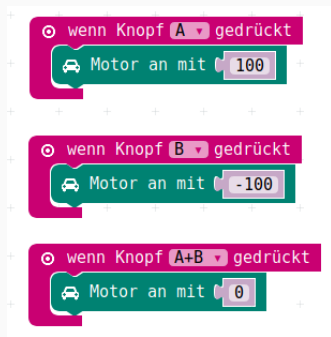
Dazu bauen wir das Programm um:

- Knopf **A** => vorwärts
- Knopf **B** => rückwärts
- Knopf **A UND B** => Stop



# Mit Motor rückwärts

Wenn wir nun unser Programm im Arbeits-Bereich entsprechend abändern, sieht das nun so aus:



**Figure 9:** Motor Programm Vor / Rück

Auch dieses Programm können wir leider im Simulator nicht nutzen und müssen wir das Programm auf den Calliope runterladen und dort ausprobieren.



## Java-Script-Code

```
input.onButtonPressed(Button.A, () => {  
    motors.motorPower(100)  
})  
input.onButtonPressed(Button.B, () => {  
    motors.motorPower(-100)  
})  
input.onButtonPressed(Button.AB, () => {  
    motors.motorPower(0)  
})
```

## Download Hex-Code

Hex-code



# 06\_05\_LageSensor

Calliope-Kurs Kinder

---

Jogi Künstner, Turbine Brunnen

Herbst 2020



# Motorsteuerung mit Lage-Sensor

---

Das funktioniert ja schon mal ganz gut.

Nun möchten wir mit diesem einfachen Motor-Steuerungs-Programm auch noch eine andere Eingangs-Möglichkeit ausprobieren:

## **Den Lage-Sensor!**

Der Calliope hat einen Lage-Sensor eingebaut, der in allen Raumrichtungen funktioniert.

Also :

- Oben / Unten
- Links / Rechts
- Vorne / Hinten



Die Abfragen, um den Lage-Sensor genau auszuwerten, sind recht kompliziert. Man muss Koordinaten-Systeme verstehen und man sollte Winkelrechnung verstehen. Beides ist in Euerem Alter wahrscheinlich noch nicht der Fall.

Zusätzlich zu den genauen Abfrage-Möglichkeiten, die schwierig zu verwenden sind, hat der Calliope aber auch die Möglichkeit, sehr einfach den Lage-Sensor abzufragen.

Das wollen wir nun tun:

- Beim Gerade halten des Calliope soll der Motor aus sein.
- Beim Kippen nach links soll er sich nach vorne drehen
- Beim Kippen nach rechts soll er sich nach hinten drehen.



Sowohl die genauen, schwierigeren Befehle als auch die Einfachen befinden sich im Menu Eingabe:

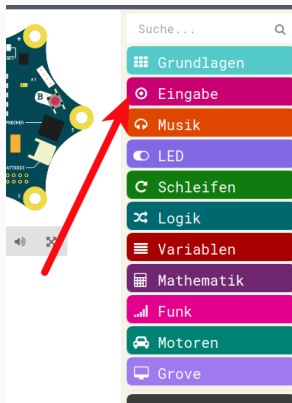


Figure 1: Menu Eingabe



# Inhalte Eingabe-Menü

Calliope mini Projekte Teilen

Suche...

- Grundlagen
- Eingabe**
- Mehr
- Musik
- LED
- Schleifen
- Logik
- Variablen
- Mathematik
- Funk
- Motoren
- Grove
- Fortgeschritten

**Einfach**

**Komplex**

- wenn Knopf A gedrückt
- wenn geschüttelt
- wenn Knopf B gedrückt
- wenn Pin P0 gedrückt
- wenn Knopf A+B gedrückt
- Knopf A ist gedrückt
- Pin P0 ist gedrückt
- Beschleunigung (mg)
- Lichtstärke
- Kompassausrichtung (°)
- Temperatur (°C)

Figure 2: Menu Eingabe Inhalt



# Wenn geschüttelt

Nun ziehen wir drei mal das **wenn geschüttelt** in unseren Arbeits-Bereich:

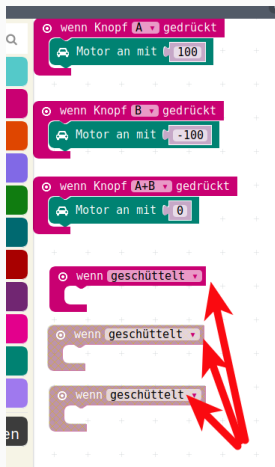


Figure 3: Dreimal Geschuettelt



Diese wandeln wir nun alle durch Druck auf das Dreieck:



Figure 4: Dreieck

um in drei verschiedene Reaktionen:

- “nach links neigen”
- “Display nach oben”
- “nach rechts neigen”



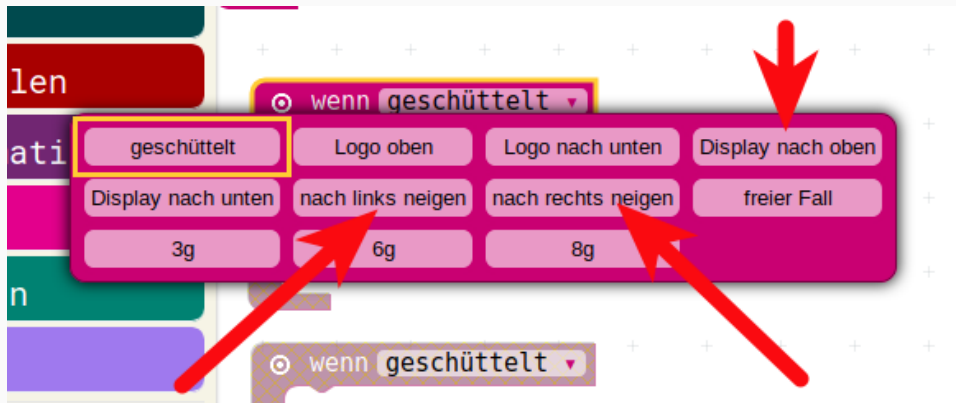


Figure 5: Lage-Sensor

Damit sieht unser Programm nun so aus:

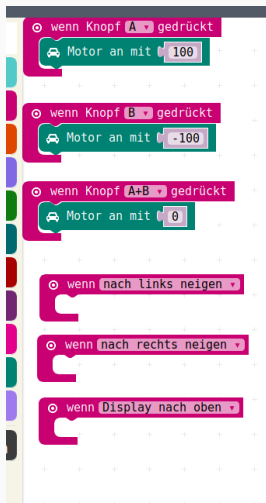


Figure 6: Lage-Sensor drin



und wenn wir nun die entsprechenden Befehle von oben nach unten schieben, dann können wir unseren Motor durch kippen steuern.



Dieses Programm können wir nun auch in den Calliope laden.

Achtung: Zumindest bei manchen Kombinationen von Calliope und Computer (und vermutlich angeschlossenen Computer-Ladegerät) hat der Lage-Sensor **NICHT** richtig funktioniert.

==>

Bitte steckt in diesem Fall das USB - Kabel aus und betreibt Euren Calliope nur über Batterie.



## Java-Script-Code

```
input.onGesture(Gesture.TiltLeft, () => {  
  motors.motorPower(100)  
})  
input.onGesture(Gesture.TiltRight, () => {  
  motors.motorPower(-100)  
})  
input.onGesture(Gesture.ScreenUp, () => {  
  motors.motorPower(0)  
})
```

## Download Hex-Code

Hex-code





- Zurück zu Tag 3 Hausaufgaben (SchubladenAlarm)
- Hoch zur Übersicht
- Weiter zu Tag 4 Hausaufgaben (Schleifen)



Für alle Bilder auf diesen Folien/Seiten (sofern nicht anders angegeben) gilt:

- Autor: Jörg Künstner
- Lizenz: CC BY-SA 4.0

