

# micro:bit Challenge-Cards

Physical Computing –  
Meistere die Challenges und erlebe, wie man die physische  
und die virtuelle Welt verbindet.

Version 2. Zyklus

PH <sup>SG</sup>



# Inhalt

## Grundlagen

- Der micro:bit (Ausstattung)
  - Zubehör
  - Ein Programm auf den micro:bit hochladen
  - Analoger Input und Output
  - Digitaler Input und Output
- 

## Challenges

- |  |  |   |
|--|--|---|
| 1. Hello World!                            | 8. Musik komponieren und abspielen       | 14. Die Farben des Regenbogens              |
| 2. Die Tasten A und B benutzen             | 9. Emojis mit der Fingerspitze verändern | 15. Eine Lichterkette erleuchten            |
| 3. Die Tasten A und B steuern das Licht    | 10. Den Kompass benutzen                 | 16. Einen Servo-Motor steuern               |
| 4. Eine Taste steuert das Licht            | 11. Die Helligkeit messen                | 17. Einen DC-Motor steuern                  |
| 5. Einen verstellbaren Widerstand benutzen | 12. Den Lagesensor benutzen              | 18. Einen linearen Motor steuern (Solenoid) |
| 6. Ein Licht dimmen                        | 13. Die Temperatur messen                |   |
| 7. Einen Vibrationsmotor steuern           |  |   |
- 

## Impressum

Version 2.6 (April 2019)

Dr. Dorit Assaf | Anpassung für 2. Zyklus M. Garzi

Pädagogische Hochschule St. Gallen

[dorit.assaf@phsg.ch](mailto:dorit.assaf@phsg.ch), [www.phsg.ch](http://www.phsg.ch)

Dieses Dokument basiert auf Version 2.4 (November 2018) von Dorit Assaf, PHSZ.

Bilder, Grafiken, Screenshots: Dorit Assaf

Icons: [thenounproject.com](http://thenounproject.com), Compass by FakehArtwork, Button Click by andriwidodo, LED by Arthur Shlain, Arcade Button by emma mitchell, Potentiometer by Hans, vibration motor by Hans, Alarm by Sergey Demushkin, LED by Victor Bolivar, brightness by Hermine Blanquart, Thermometer by Hopkins, Airplane by icon 54, Servo motor by Branis Panos, Electric motor by Arthur Shlain, Battery by Sergey Demushkin, Led Strip by adls.



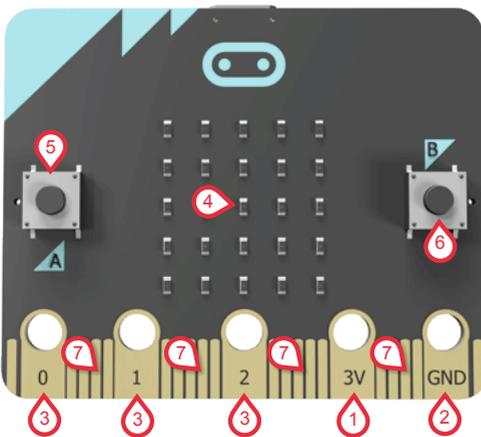
Namensnennung

Weitergabe unter gleichen Bedingungen

## Grundlagen

# Der micro:bit

### Ausstattung Teil 1



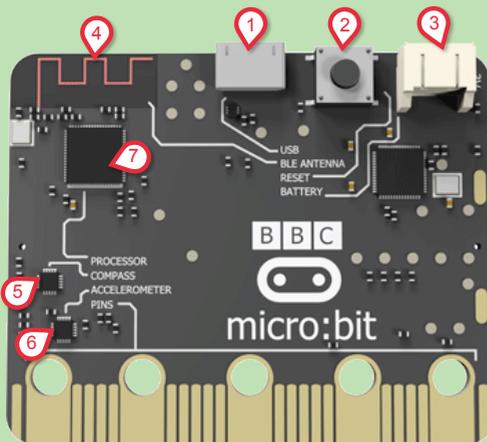
Achtung:  
VCC (+) und GND (-) nie direkt  
verbinden (Kurzschluss!)

- ① VCC (3.3V) (+)
- ② GND (Masse) (-)
- ③ Analoge oder digitale Input- und Output-Pins
- ④ 5x5 LED-Display, Helligkeitssensor
- ⑤ Taste A (digitaler Input)
- ⑥ Taste B (digitaler Input)
- ⑦ Zusätzliche analoge oder digitale Input- und Output-Pins (nur über Adapter oder Motor Board zugänglich)

## Grundlagen

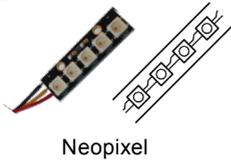
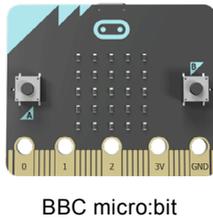
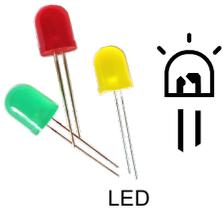
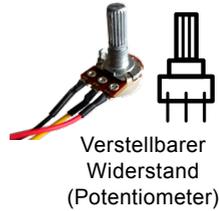
# Der micro:bit

### Ausstattung Teil 2



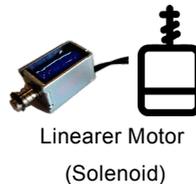
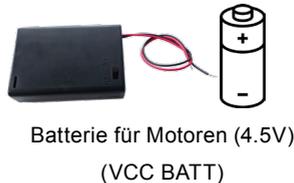
- ① Micro-USB-Anschluss  
(Programme übertragen,  
Stromversorgung)
- ② Reset-Taste  
(startet das Programm neu)
- ③ Steckplatz für externe 3V-Batterie-Packs (Stromversorgung ohne USB-Kabel)
- ④ Bluetooth-Antenne für kabellose Verbindung mit der micro:bit-App oder zwischen mehreren micro:bits
- ⑤ Kompass
- ⑥ Beschleunigungssensor
- ⑦ Prozessor (16 MHz 32-bit ARM Cortex-M0, 256 KB Flash-Speicher, 16 KB RAM) mit Temperatursensor

# Grundlagen Zubehör



---

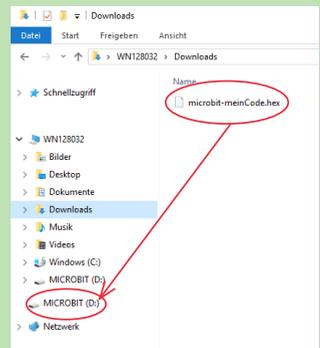
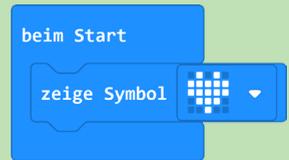
## Motoren



## Grundlagen

# Ein Programm auf den micro:bit hochladen

- 1 Öffne [makecode.microbit.org](https://makecode.microbit.org). Klicke auf «neues Projekt».
- 2 Der «beim Start»-Block ist bereits vorhanden. Wähle aus der Befehlsgruppe «Grundlagen» den Block «zeige Symbol» und füge ihn ein.
- 3 Wähle einen Namen für das Programm, z.B. «meinCode».
- 4 Klicke auf «Herunterladen» und speichere die Datei «microbit-meinCode.hex».
- 5 Schliesse den micro:bit über das USB-Kabel an.
- 6 Öffne den Datei-Explorer (Win) oder Finder (Mac) und ziehe die gespeicherte microbit-meinCode.hex Datei auf das Laufwerk «MICROBIT».
- 7 Solange das Programm auf den micro:bit hochgeladen wird, blinkt ein gelbes Licht auf der Rückseite. Das Programm startet anschliessend von selbst.
- 8 Bei jeder Änderung des Programms muss es neu auf den micro:bit hochgeladen werden (Schritt 4 - 7 wiederholen). Das alte Programm wird dabei überschrieben.



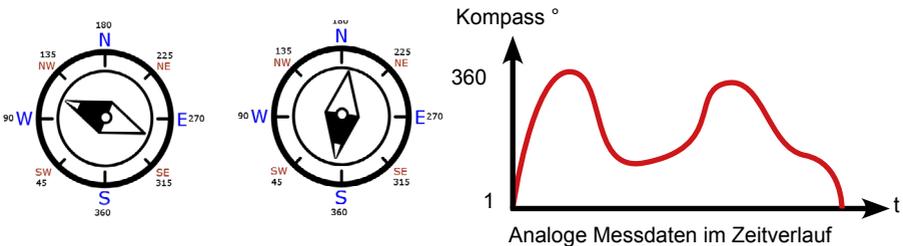
## Grundlagen

# Analoger Input und Output

---

### Analoges Signal

Bei einem analogen Sensor erhalten wir Messwerte, welche in einem bestimmten Zahlenbereich (Wertebereich) liegen. Bei einem Kompass ist dies beispielsweise ein Wertebereich von  $1^\circ$  bis  $360^\circ$ . Ein **analoger Input**, wie der des Kompasses, kann also 360 verschiedene Werte messen. Ein **analoger Output** liefert ebenfalls Werte innerhalb eines vorgegebenen Bereichs.



---

### Sensoren und Aktoren

Sensoren sind die «Fühler» der Aussenwelt: Sie wandeln physikalische Grössen und Gegebenheiten der Umwelt in elektrische Signale um. So kann der micro:bit mithilfe der Sensoren Informationen aus seiner Umwelt wahrnehmen. Sensoren liefern also immer Inputs.

Aktoren bewirken etwas in der Aussenwelt: Sie wandeln elektrische Signale in physikalische Aktionen um. Der micro:bit steuert Aktoren, also sind Aktoren immer Outputs.

## Grundlagen

# Digitaler Input und Output

---

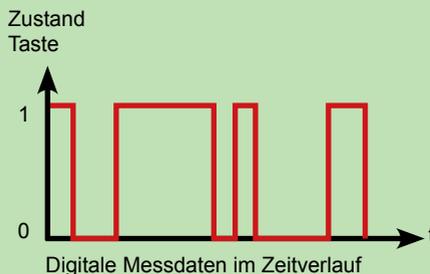
### Digitales Signal

Der Wertebereich eines digitalen Inputs begrenzt sich auf die Zahlen 0 und 1, die zwei Zustände darstellen. Eine Taste ist ein gutes Beispiel für einen digitalen Input: Sie kann entweder im Zustand gedrückt oder nicht gedrückt sein. Einen Zustand dazwischen (halbgedrückt) gibt es nicht. Ob der gedrückte Zustand dem Wert «1» oder dem Wert «0» entspricht, hängt vom elektrischen Schaltkreis ab. Digitale Outputs haben ebenfalls nur zwei Zustände.

Taste gedrückt  
(z.B. Zustand  
«1»)



Taste nicht  
gedrückt (z.B.  
Zustand «0»)



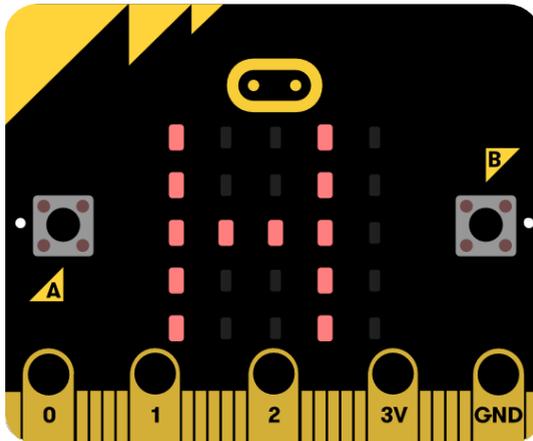
Schwierigkeitslevel «EINFACH»

1

# Hello World!



5 MINUTEN



## Challenge

Schreibe einen Lauftext deiner Wahl und lass ihn unendlich oft laufen.

# Lösung

Hello World!

---

VERWENDETE BEFEHLSGRUPPEN

 Grundlagen

---

Code

dauerhaft

zeige Zeichenfolge " Hallo! "

Hinweis

- Um ein neues oder verändertes Programm auf dem micro:bit zu testen, muss es jedesmal von Neuem hochgeladen werden (gemäss Grundlagenkarte). Dabei wird das alte Programm auf dem micro:bit überschrieben.
- Die hex-Dateien des Programms werden im Download-Ordner des Browsers bei jedem Herunterladen mit einer fortlaufenden Zahl versehen (z.B. «microbit-meinCode (9).hex»). Die hex-Datei ist nach dem Hochladen nicht auf dem «MICROBIT»-Laufwerk sichtbar und kann auch nicht mehr vom micro:bit zurückkopiert werden. Es lohnt sich, die hex-Dateien sinnvoll beschriftet auf dem Computer zu speichern.
- Nach dem Hochladen wird die USB-Verbindung kurz getrennt. Dabei kann eine Meldung erscheinen, dass ein USB-Speicher nicht ordentlich getrennt wurde. Das ist kein Problem und kann ignoriert werden.

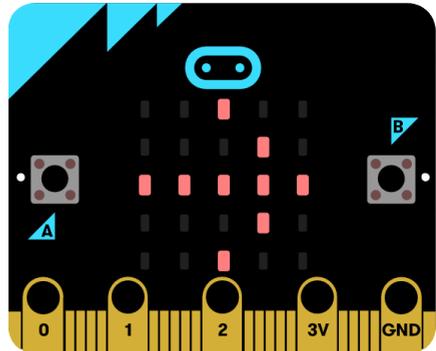
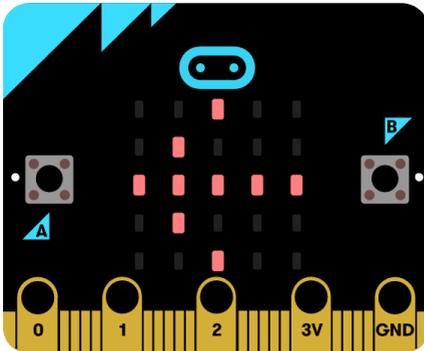
Schwierigkeitslevel «EINFACH»

2

# Die Tasten A und B benutzen



5 MINUTEN



## Challenge

Wenn du die Taste A drückst, erscheint ein Pfeil auf dem LED-Display, der nach links zeigt. Wenn du die Taste B drückst, zeigt der Pfeil nach rechts.

# Lösung

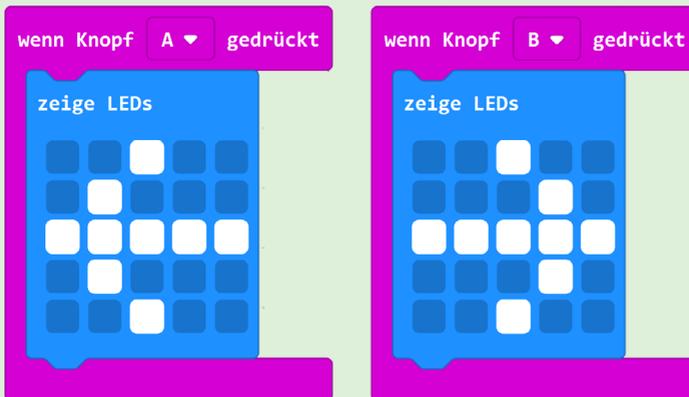
## Die Tasten A und B benutzen

VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Eingabe

## Code



## Hinweis

Hier wurden Ereignisblöcke gewählt. Es ist auch korrekt, das Verhalten über einen Bedingungsblock «wenn/dann» und den Parameterblock «Button A/B ist gedrückt» zu steuern.

Schwierigkeitslevel «EINFACH»

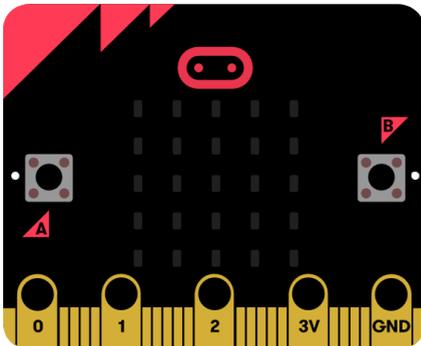
3

# Die Tasten A und B steuern das Licht



10 MINUTEN

ZUBEHÖR



## Challenge

Klemme eine LED an den micro:bit. Wenn du die Taste A drückst, wird die LED eingeschaltet. Wenn du die Taste B drückst, wird die LED wieder ausgeschaltet.

# Lösung

## Die Tasten A und B steuern das Licht

### VERWENDETE BEFEHLSGRUPPEN

Eingabe

Fortgeschritten

Pins

### Code

wenn Knopf A gedrückt

schreibe digitalen Wert von Pin P2 auf 1

wenn Knopf B gedrückt

schreibe digitalen Wert von Pin P2 auf 0

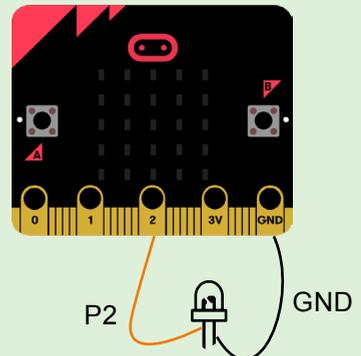
### Hinweis

Ein digitaler Wert von «1» bedeutet, dass der digitale Output am Pin «eingeschaltet» wird, d.h. der entsprechende Pin erhält eine elektrische Spannung von 3V. Der Wert «0» hingegen bedeutet «keine Spannung am Pin». Diese aktivierte Spannung kann z.B. dazu verwendet werden, um eine LED zum Leuchten zu bringen.

### Elektronik

LED (+/-)!

- Langes Bein → digitaler Output (P2)
- Kurzes Bein → GND (-)



Schwierigkeitslevel «MITTEL»

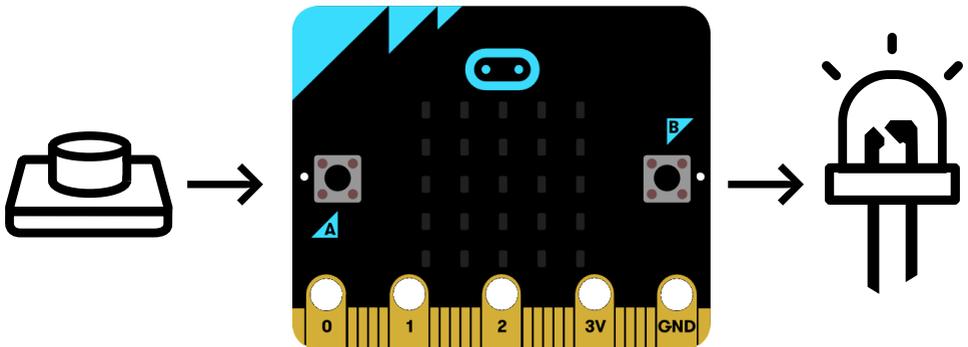
4

# Eine Taste steuert das Licht



10 MINUTEN

ZUBEHÖR



## Challenge

Klemme eine LED und eine Taste an den micro:bit.

Wenn du die Taste drückst, leuchtet die LED. Wenn du die Taste loslässt, schaltet die LED wieder aus.

# Lösung

## Eine Taste steuert das Licht

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Fortgeschritten

Pins

### Code

dauerhaft

```
schreibe digitalen Wert von Pin P2 auf digitale Werte von Pin P1
```

### Hinweis

Diese kleine Programm ist verschachtelt.

Der Parameterblock `digitale Werte von Pin P1` wird zuerst ausgeführt und das Resultat (der erkannte Zustand der Taste) dem Block «schreibe digitalen Wert von Pin P2 auf» übergeben, welcher die LED an- oder ausschaltet.

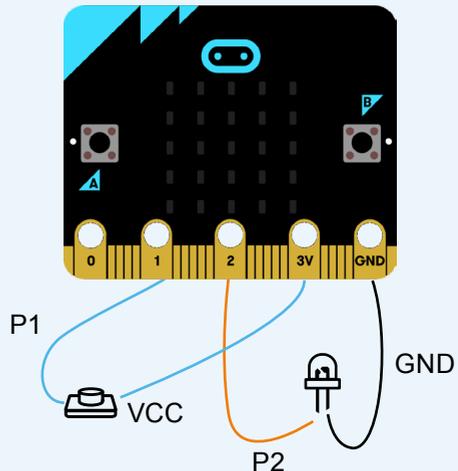
### Elektronik

LED (+/-)!

- Langes Bein → digitaler Output (P2)
- Kurzes Bein → GND

Taste

- Äusseres Bein → digitaler Input (P1)
- Äusseres Bein → VCC (+)



Schwierigkeitslevel «MITTEL»

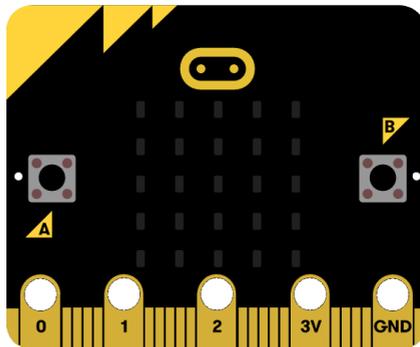
5

# Einen verstellbaren Widerstand benutzen



10 MINUTEN

ZUBEHÖR



## Challenge

Klemme einen verstellbaren Widerstand (Potentiometer) an den micro:bit. Drehe den Regler in verschiedene Positionen und zeige den jeweils aktuell ausgelesenen Sensorwert als Zahlenwert auf dem LED-Display an.

# Lösung

## Einen verstellbaren Widerstand nutzen

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Fortgeschritten

Pins

### Code

```
dauerhaft
  zeige Zeichenfolge "*"
  zeige Nummer analoge Werte von Pin P1
```

### Hinweis

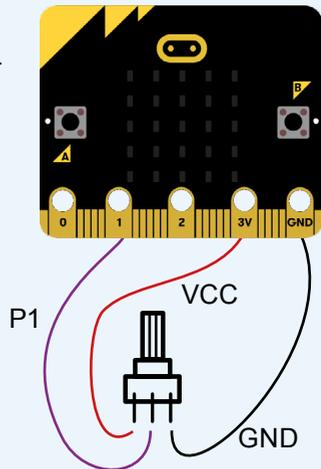
Die Zeichenfolge «\*» vor «zeige Nummer» hilft, auf dem LED-Display die Zahl besser zu erkennen (Beginn des Lauftextes).

Um Sensorwerte dauerhaft und nicht nur einmalig auszulesen, wird eine «dauerhaft-Schleife» verwendet.

### Elektronik

#### Potentiometer

- Mittleres Bein → analoger Input (P1)
- Äusseres Bein → GND (-)
- Äusseres Bein → VCC (+)



Schwierigkeitslevel «MITTEL»

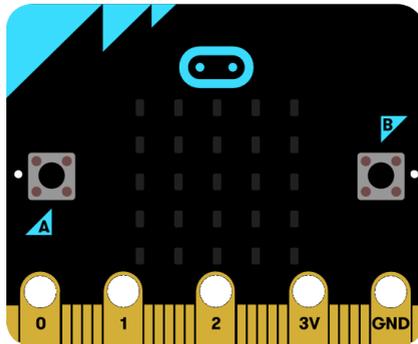
6

# Ein Licht dimmen



10 MINUTEN

ZUBEHÖR



## Challenge

Klemme einen verstellbaren Widerstand (Potentiometer) und eine LED an den micro:bit. Durch das Drehen des Reglers am Potentiometer wird die LED heller oder dunkler gedimmt.

# Lösung

## Ein Licht dimmen

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Fortgeschritten

Pins

### Code

dauerhaft

schreibe analogen Pin P2 auf analoge Werte von Pin P1

### Verwendete Codeblocks

schreibe analogen Pin P2 auf

analoge Werte von Pin P1

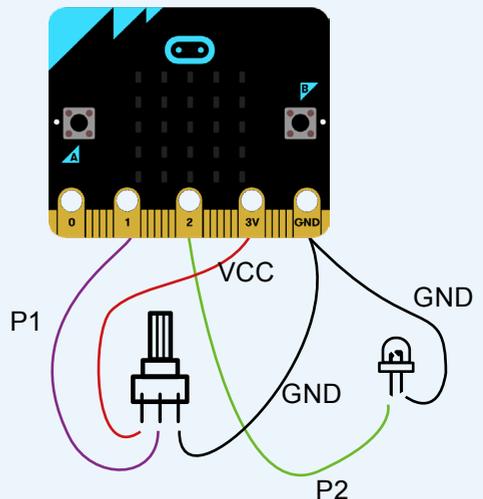
### Elektronik

#### LED (+/-)!

- Kurzes Bein → GND (-)
- Langes Bein → Analoger Output (P2)

#### Potentiometer

- Mittleres Bein → analoger Input (P1)
- Äusseres Bein → GND (-)
- Äusseres Bein → VCC (+)



Schwierigkeitslevel «MITTEL»

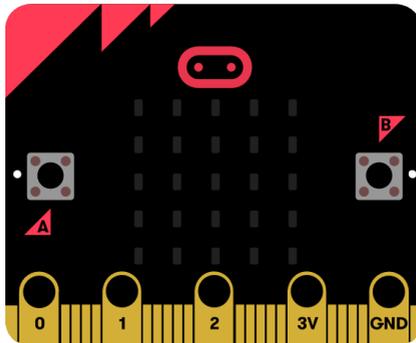
7

# Einen Vibrationsmotor steuern



10 MINUTEN

ZUBEHÖR



## Challenge

Klemme einen verstellbaren Widerstand (Potentiometer) und einen Vibrationsmotor an den micro:bit. Durch das Drehen des Reglers am Potentiometer wird der Motor gesteuert.

# Lösung

## Einen Motor steuern

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Fortgeschritten

Pins

### Code

dauerhaft

schreibe analogen Pin P2 auf analoge Werte von Pin P1

### Hinweis

Eine gedimmte LED und ein Vibrationsmotor sind beides analoge Outputs. Deshalb ist der Code genau gleich wie in Challenge 6 «Ein Licht dimmen». Ein Vibrationsmotor benötigt nicht viel Strom, darum kann er ohne externe Batterie an den micro:bit angehängt werden (im Gegensatz zur Challenge 16-18).

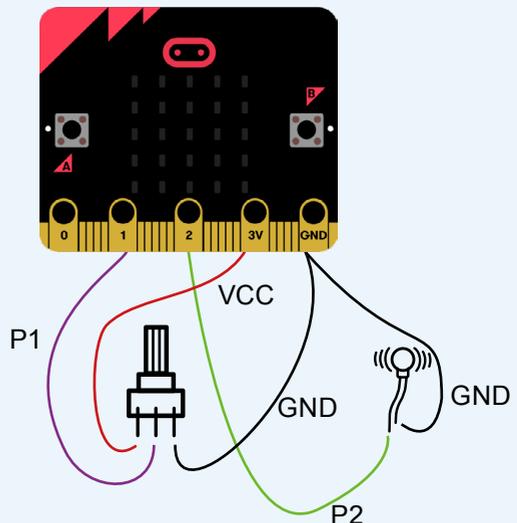
### Elektronik

#### Vibrationsmotor (+/-)!

- Schwarzes Kabel → GND (-)
- Rotes Kabel → analoger Output (P2)

#### Potentiometer

- Äusseres Bein → GND (-)
- Mittleres Bein → analoger Input (P1)
- Äusseres Bein → VCC (+)



Schwierigkeitslevel «EINFACH»

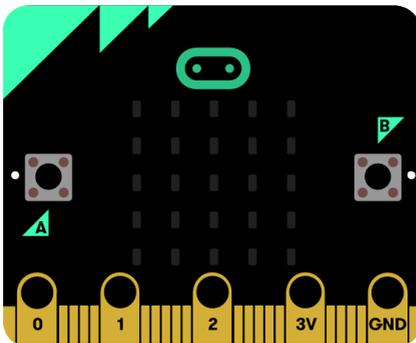
8

# Musik komponieren und abspielen



10 MINUTEN

ZUBEHÖR



## Challenge

Klemme einen Buzzer an den micro:bit. Komponiere deine eigene Musik und spiele sie ab.

# Lösung

## Musik komponieren und abspielen

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Musik

Schleifen

### Code

```
beim Start
  ändere die Geschwindigkeit (bpm) 40
  2 -mal wiederholen
  mache
    spiele Note Mittleres C für 1/8 Takt
    spiele Note Mittleres D für 1/8 Takt
    spiele Note Mittleres D für 1/4 Takt
  spiele Note Mittleres G für 1/8 Takt
  spiele Note Mittleres F für 1/8 Takt
  spiele Note Mittleres E für 1/4 Takt
```

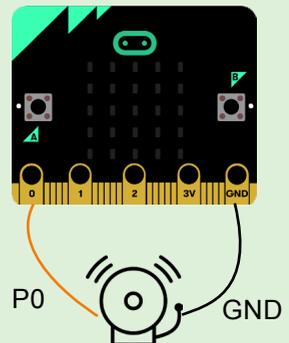
### Hinweis

Mit dem «beim Start»-Block wird die Musik einmal abgepielt. Mit der Reset-Taste auf dem micro:bit kann sie nochmals abgepielt werden. Um die Musik unendlich oft abzuspielen, kann der «dauerhaft»-Block verwendet werden.

### Elektronik

Buzzer (+/-)!

- Schwarzes Kabel → GND (-)
- Rotes Kabel → digitaler Output (P0)
- Beim micro:bit kann nur über P0 Musik gespielt werden!



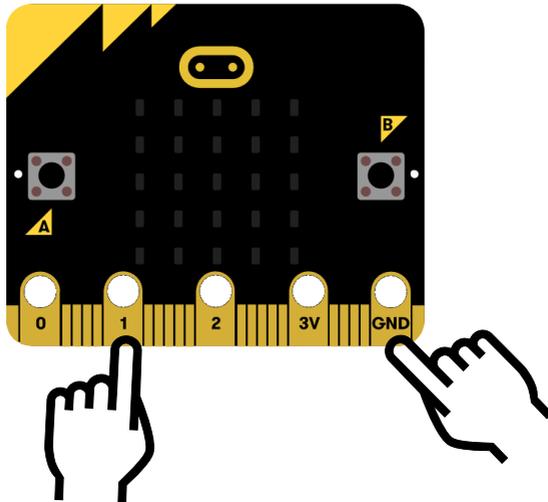
Schwierigkeitslevel «EINFACH»

9

# Emojis mit der Fingerspitze verändern



5 MINUTEN



## Challenge

Das LED-Display ändert das Emoji-Symbol, wenn du mit der Fingerspitze die Pins 0 bis 2 berührst.

# Lösung

## Emojis mit der Fingerspitze verändern

---

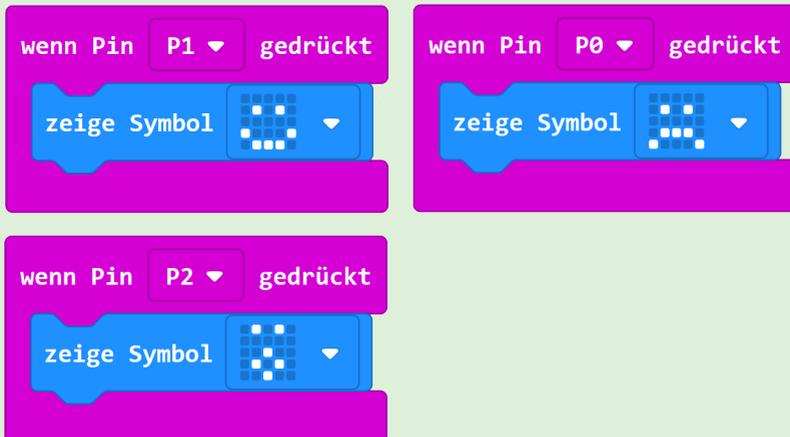
### VERWENDETE BEFEHLSGRUPPEN

 Grundlagen

 Eingabe

---

### Code



### Hinweis

Um mit der Fingerspitze die Pins zu «drücken», muss gleichzeitig mit einem Finger der **GND-Pin** und mit dem anderen Finger einer der **Pins 0 bis 2** berührt werden. Das funktioniert auch mit zwei Händen. Dabei dient der eigene Körper als elektrischer Leiter und der Stromkreis schliesst sich über unseren Körper.

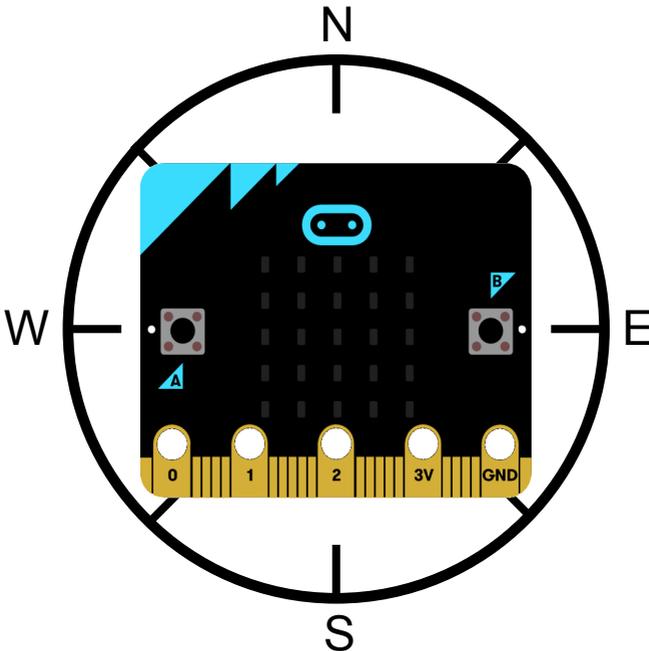
Schwierigkeitslevel «EINFACH»

10

# Den Kompass benutzen



5 MINUTEN



## Challenge

Zeige die Werte des Kompasses auf dem LED-Display an. Drehe den micro:bit in jede Richtung und zeichne die Werte auf einem Blatt Papier auf.

# Lösung

## Den Kompass benutzen

---

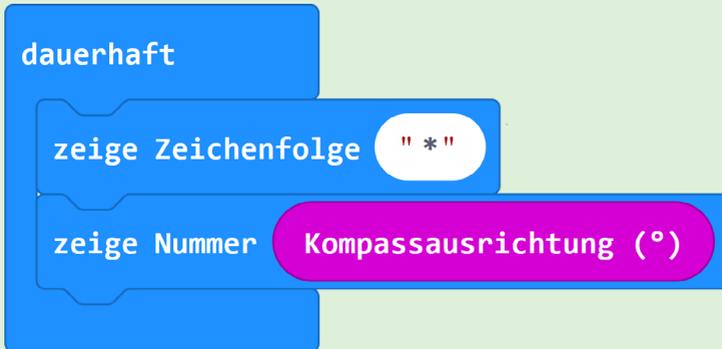
### VERWENDETE BEFEHLSGRUPPEN

 Grundlagen

 Eingabe

---

### Code



### Hinweis

- Nach jedem Hochladen eines Programms, bei dem der Kompass verwendet wird, muss dieser neu kalibriert werden. Der micro:bit verlangt dazu, einen Kreis zu zeichnen: «draw a circle». Kippe den micro:bit so, bis der Kreis komplett ist.
- Die Zeichenfolge «\*» vor «zeige Nummer» hilft, auf dem LED-Display die Zahl besser zu erkennen (Beginn des Lauftextes).
- Halte den micro:bit mit dem LED-Display nach oben zeigend parallel zum Boden und drehe ihn wie einen Kompass um 360°. Schwankungen in der Messung sind normal.

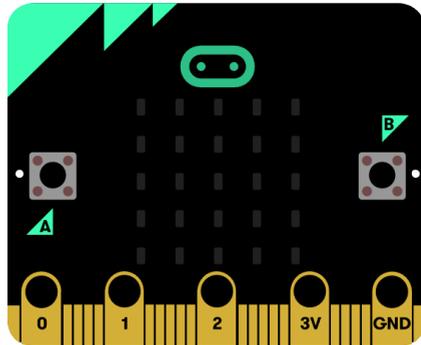
Schwierigkeitslevel «EINFACH»

11

# Die Helligkeit messen



5 MINUTEN



## Challenge

Zeige die Werte des Helligkeitssensors auf dem LED-Display an. Bringe den micro:bit in verschiedene Lichtverhältnisse.

# Lösung

## Die Helligkeit messen

---

### VERWENDETE BEFEHLSGRUPPEN

 Grundlagen

 Eingabe

---

### Code



### Hinweis

- Die Zeichenfolge «\*» vor «zeige Nummer» hilft, auf dem LED-Display die Zahl besser zu erkennen (Beginn des Lauftextes).
- Das LED-Display ist gleichzeitig auch der Helligkeitssensor.

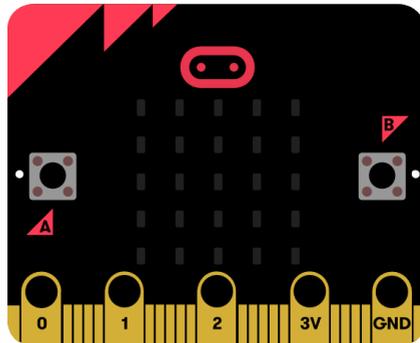
Schwierigkeitslevel «SCHWER»

12

# Den Lagesensor benutzen



10 MINUTEN



## Challenge

Zeige die Werte des Gyroskops als Rotationswinkel in Grad auf dem LED-Display an. Lass dein Programm erkennen (detektieren), wenn der micro:bit geschüttelt wird.

# Lösung

## Den Lagesensor benutzen

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Eingabe

### Code

```
dauerhaft
  zeige Zeichenfolge "N"
  zeige Nummer Rotation (°) Winkel
  zeige Zeichenfolge "R"
  zeige Nummer Rotation (°) rollen
  wenn geschüttelt
    zeige Symbol
```



### Hinweis

- Makecode enthält einen eigenen Ereignisblock für den Lagesensor zur Verfügung.
- Ein Lagesensor besteht aus einem Gyroskop, Beschleunigungssensor und Kompass. Diese Sensoren können auch einzeln ausgelesen werden.
- Ein Beschleunigungssensor zeigt immer auch die Erdbeschleunigung an.

```
zeige Nummer Rotation (°) Winkel
zeige Nummer Rotation (°) rollen
zeige Nummer Beschleunigung (mg) x
zeige Nummer Beschleunigung (mg) y
zeige Nummer Beschleunigung (mg) z
zeige Nummer Kompassausrichtung (°)
```

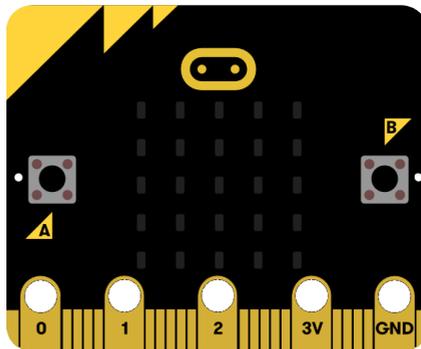
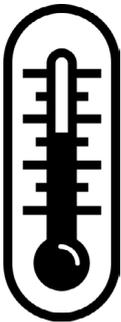
Schwierigkeitslevel «EINFACH»

13

# Die Temperatur messen



5 MINUTEN



## Challenge

Zeige die Werte des Temperatursensors auf dem LED-Display an.

# Lösung

## Die Temperatur messen

---

### VERWENDETE BEFEHLSGRUPPEN

 Grundlagen

 Eingabe

---

### Code



### Hinweis

- Die Zeichenfolge «\*» vor «zeige Nummer» hilft, auf dem LED-Display die Zahl besser zu erkennen (Beginn des Lauftextes).
- Der Temperatursensor benötigt einige Minuten, bis er sich eingependelt hat.

Schwierigkeitslevel «SCHWER»

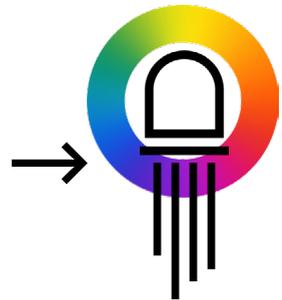
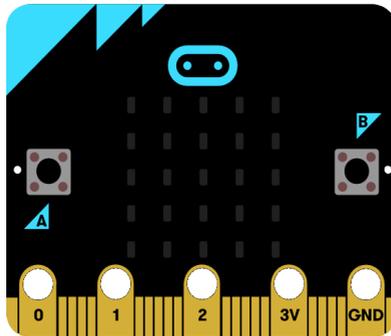
14

# Die Farben des Regenbogens



10 MINUTEN

ZUBEHÖR



## Challenge

Klemme eine RGB-LED an den micro:bit. Zeige die Farben des Regenbogens nacheinander an.

# Lösung

## Die Farben des Regenbogens

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Fortgeschritten

Pins

### Code

Hier ein Codebeispiel mit zwei Farben.  
Für weitere Farben müssen die Blöcke lediglich kopiert und die Werte eingefügt werden

### Hinweis

Dies sind die 10-Bit RGB-Werte des Regenbogens.

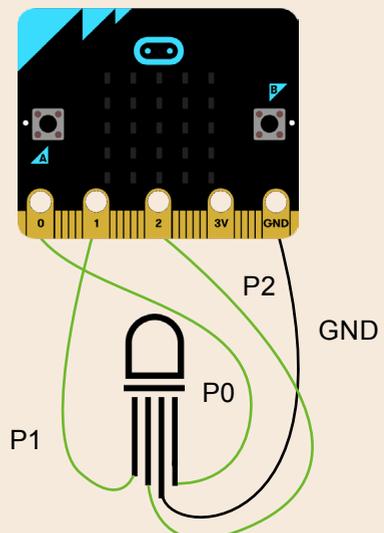
594, 0, 846
300, 0, 521
0, 0, 1023
0, 1023, 0
1023, 1023, 0
1023, 509, 0
1023, 0, 0

### Elektronik

#### RGB-LED (+/-)!

- Kürzestes Bein (Grün) → analoger Output (P1)
- Längstes Bein → GND (-)
- Bein neben Grün (Blau) → analoger Output (P2)
- Äusseres Bein (Rot) → analoger Output (P0)

```
dauerhaft
schreibe analogen Pin P0 auf 594
schreibe analogen Pin P1 auf 0
schreibe analogen Pin P2 auf 846
pausiere (ms) 300
schreibe analogen Pin P0 auf 300
schreibe analogen Pin P1 auf 0
schreibe analogen Pin P2 auf 521
pausiere (ms) 300
```



Schwierigkeitslevel «SCHWER»

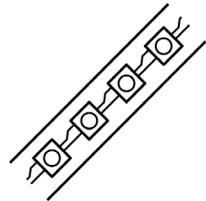
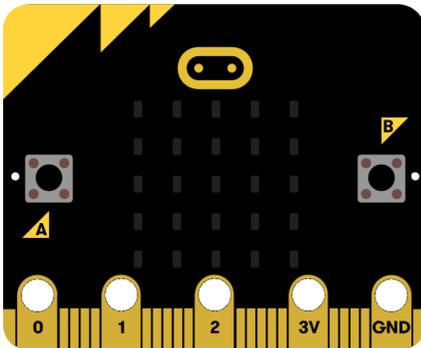
15

# Eine Lichterkette erleuchten



10 MINUTEN

ZUBEHÖR



## Challenge

Klemme einen Neopixel-Strip (LED Leuchtstreifen) an den Micro:bit. Lasse ihn in verschiedenen Farben leuchten.

# Lösung

## Eine Lichterkette erleuchten

### VERWENDETE BEFEHLSGRUPPEN



Grundlagen



Schleifen



NeoPixel

Fortgeschritten

Erweiterungen

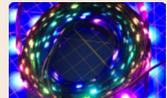
### Code

```
beim Start
  ändere strip auf NeoPixels an Pin P2 mit 5 Pixeln und Modus RGB (GRB Format)
  strip setze Helligkeit 190
  strip zeige Regenbogen von Farbton 1 bis 360
  pausiere (ms) 3000
  ändere indexVorher auf 0

dauerhaft
  für Index von 0 bis 4
  machen
    strip setze Farbe von NeoPixel Index auf indigo
    strip setze Farbe von NeoPixel indexVorher auf schwarz
    strip anzeige
    ändere indexVorher auf Index
  pausiere (ms) 500
```

### Hinweis

Für die Steuerung von Neopixel muss die Neopixel Erweiterung (Extension) importiert werden. Im Menu «Fortgeschritten» → «Erweiterungen» → Suche nach «Neopixel».



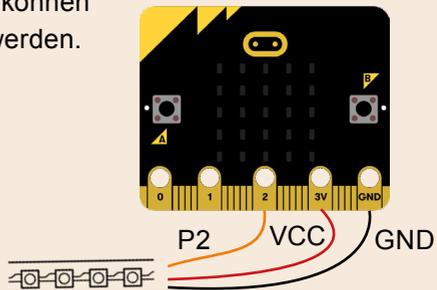
neopixel  
AdaFruit NeoPixel driver

Über eine Verbindung DOUT → DIN können mehrere Strips aneinandergehängt werden.

### Elektronik

#### Neopixel-Strip

- GND → GND (-)
- DIN → digitaler Output (P2)
- 5VDC → 3V



Schwierigkeitslevel «SCHWER»

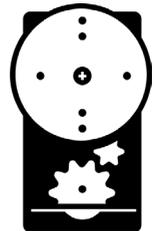
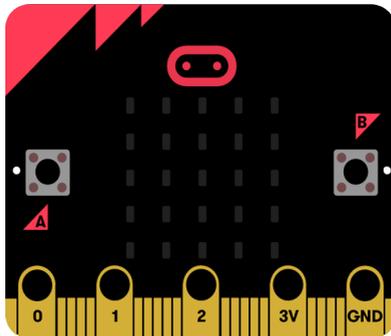
16

# Einen Servo-Motor steuern



15 MINUTEN

ZUBEHÖR



## Challenge

Klemme einen Potentiometer und einen Servo-Motor an den micro:bit. Durch das Drehen des Reglers wird der Motor gesteuert.

# Lösung

## Einen Servo-Motor steuern

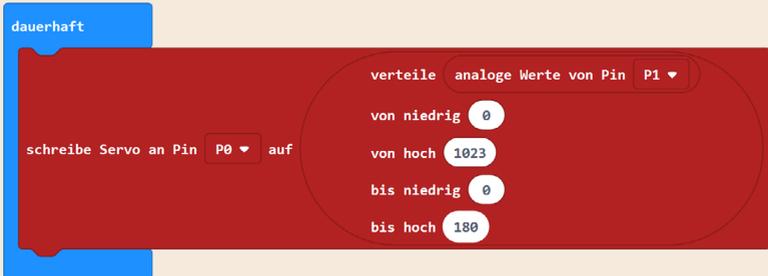
### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Fortgeschritten

Pins

### Code



### Hinweis

Für die Steuerung von Servo-Motoren benötigt man eine externe Stromversorgung (VCC BATT), da Motoren mehr Strom und/oder eine höhere Spannung benötigen, als der micro:bit liefern kann.

Es gibt auch Servo-Motoren, die um 360° drehen. Dort bedeutet die Position 0° max. Drehgeschwindigkeit in die eine Richtung, 180° max. Drehgeschwindigkeit in die andere Richtung und 90° Stillstand.

### Elektronik

Servo-Motor (+/-)!

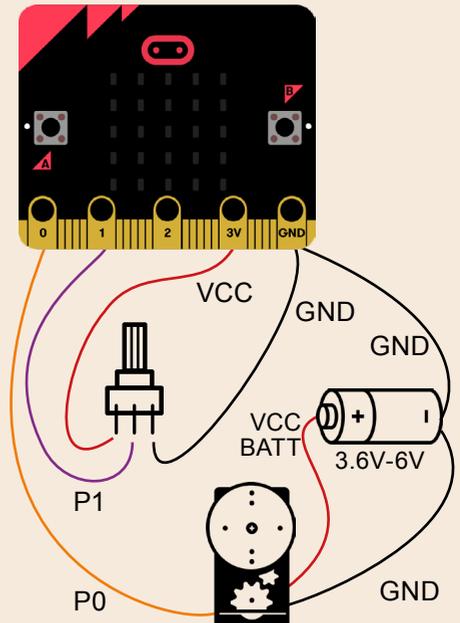
- Rotes Kabel → VCC BATT (+)
- Schwarzes Kabel → GND (-)
- Gelbes Kabel → digitaler Output (P0)

Potentiometer

- Mittleres Bein → analoger Input (P1)
- Äusseres Bein → GND (-)
- Äusseres Bein → 3V

Batterie

- Schwarzes Kabel → GND (-)



Schwierigkeitslevel «SCHWER»

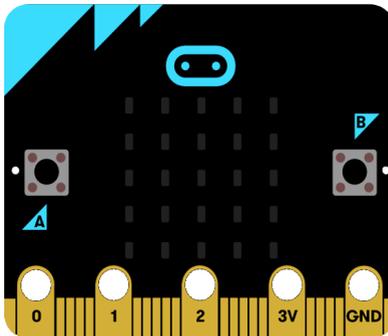
17

# Einen DC-Motor steuern



15 MINUTEN

ZUBEHÖR



## Challenge

Klemme einen DC-Motor an den micro:bit. Lasse ihn zunächst 1 Sekunde vorwärts und dann 1 Sekunde rückwärts drehen mit jeweils einer Sekunde Pause (Stillstand) dazwischen.

# Lösung

## Einen DC-Motor steuern

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Motor Driver

Fortgeschritten

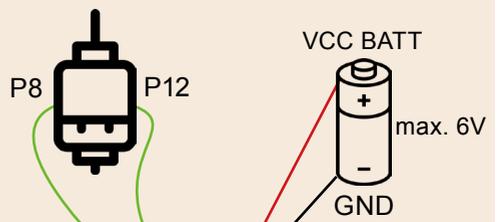
Erweiterungen

### Code

```
dauerhaft
motor 1 on direction forward speed 100
pausiere (ms) 1000
turn off motor 1
pausiere (ms) 1000
motor 1 on direction reverse speed 50
pausiere (ms) 1000
turn off motor 1
pausiere (ms) 1000
```

### Hinweis

Für die Steuerung von DC-Motoren benötigt man ein Motor-Board und eine externe Stromversorgung (Batteriepack). Der micro:bit muss im Motor-Board stecken. Für die Steuerung muss die Motor Extension importiert werden. Im Menu «Fortgeschritten» → «Erweiterungen» → Suche nach «kitronik-motor-driver».



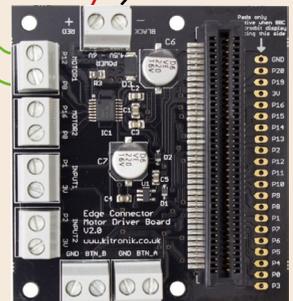
### Elektronik

#### DC-Motor

- Eine Seite → analoger Output (P8)
- Andere Seite → analoger Output (P12)

#### Batterie

- Rotes Kabel → VCC BATT (+)
- Schwarzes Kabel → GND (-)



Schwierigkeitslevel «SCHWER»

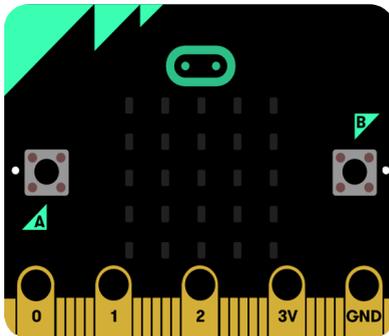
18

# Einen linearen Motor steuern (Solenoid)



10 MINUTEN

ZUBEHÖR



## Challenge

Steuere einen linearen Motor (Solenoid) über ein Relais. Beim Drücken der Taste A wird der Stift in eine Richtung geschaltet, sonst in die andere.

# Lösung

## Einen linearen Motor steuern (Solenoid)

### VERWENDETE BEFEHLSGRUPPEN



Grundlagen

Fortgeschritten

Pins



Logik



Eingabe

### Code

```
Code Editor Snippet:
dauerhaft
wenn Button A ist gedrückt dann
  schreibe digitalen Wert von Pin P1 auf 1
ansonsten
  schreibe digitalen Wert von Pin P1 auf 0
```

### Hinweis

Für die Steuerung von linearen Motoren benötigt man eine externe Stromversorgung, da sie mehr Strom und/oder eine höhere Spannung benötigen, als der micro:bit liefern kann. Ein Relais trennt die Stromkreise und funktioniert wie ein Schalter. Der micro:bit steuert das Relais, welches anschließend den Stromkreis des Motors öffnet oder schliesst.

### Elektronik

#### Linearer Motor (Solenoid)

- Eine Seite → Relay Board OUT
- Andere Seite → Batterie GND

#### Batterie

- Rotes Kabel → Relay Board OUT

#### Relay Board

- Relay Board GND → micro:bit GND (-)
- Relay Board IN → digitaler Output (P1)

