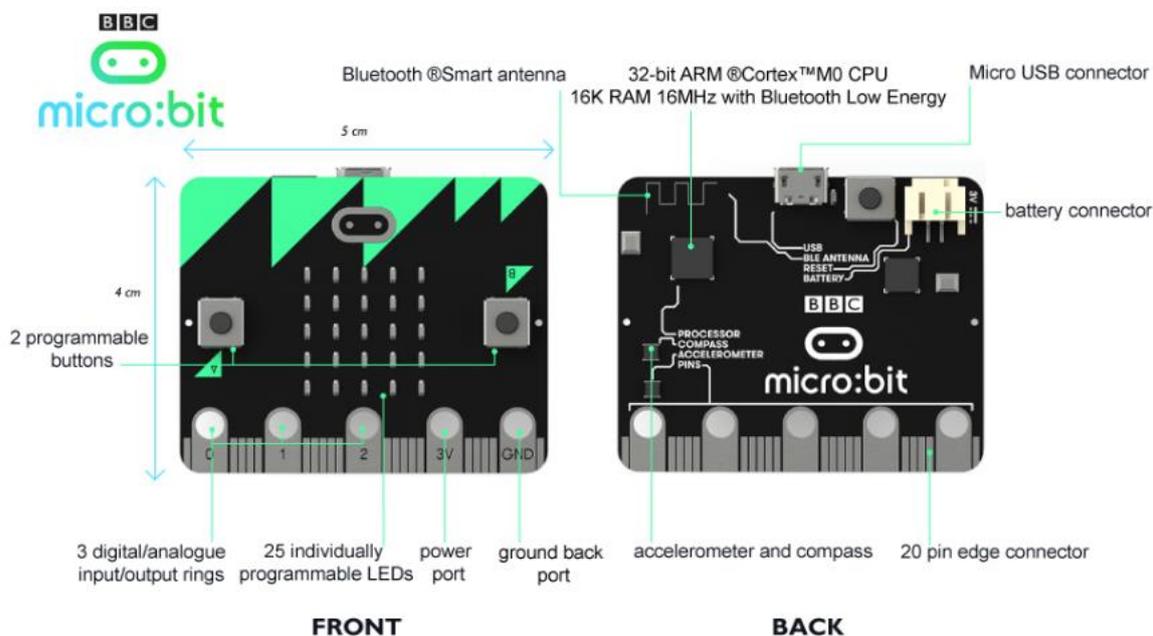


## micro:bit

Du kannst deinen BBC micro:bit für alle möglichen Arten cooler Kreationen benutzen, von Robotern zu Musikinstrumenten – die Möglichkeiten sind endlos. Dieses kleine Gerät hat viele Besonderheiten, wie zum Beispiel 25 rote LED-Lichter, die Nachrichten aufblitzen lassen können. Es gibt zwei programmierbare Tasten zur Kontrolle von Spielen oder um Lieder einer Playlist zu stoppen und überspringen. Dein BBC micro:bit kann Bewegung wahrnehmen und dir sagen in welche Richtung du gehst, und er kann eine Niedrigenergie-Bluetooth-Verbindung nutzen, um mit anderen Geräten oder dem Internet zusammenzuwirken – clever!

### Die Platine:



Erklärungen zur Hardware: <http://microbit.org/de/hardware/>

### Kodierung & Programmierung:

Hier: <http://microbit.org/de/code/>

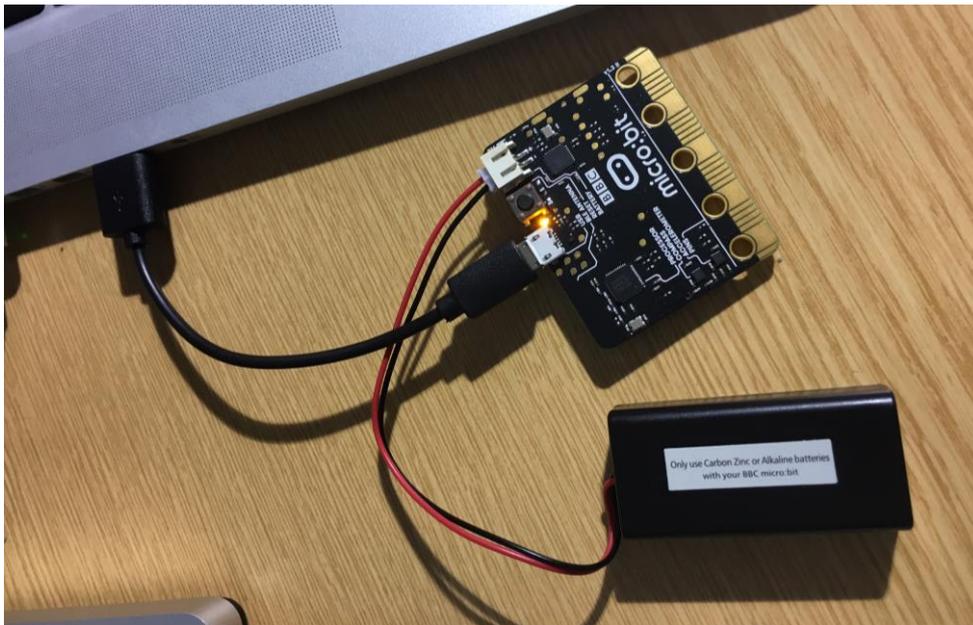
Du kannst online Editor für die Blockprogrammierung wählen oder für die Python-Programmierung. Ausserdem gibt es auch eine App dazu. Du findest alle entsprechenden Links auf der Seite, die oben angegeben ist.

Mit dem MU-Editor (separater Download: <https://codewith.mu/> ) kannst du Python-Programme schreiben und auf die Platine laden. Oder noch besser, du verwendest folgenden Editor für Python: <https://python.microbit.org/v/1.1>

### Zusammenbau:

Die Platine wird während dem Betrieb am Computer durch das USB-Kabel mit Strom versorgt. Sobald du den micro:bit vom Kabel trennst, braucht es eine Batterie (2x1.5V), welche die Platine mit Strom versorgt (siehe Bild).





Gute Einstiegs-Tutorials findest du auf YouTube, wie z.B. das folgende:

[part 1](#) – [part 2](#) – [part 3](#) – [part 4](#)

### Aufgaben mit dem online Block-Editor:

Sprach-Tipp: auf „English“ und nicht auf „Deutsch“ einstellen, da das spätere codierte Programmieren auch auf Englisch sein wird). Hier Sprache einstellen:



### Auftrag 1: Zufallszahl

Der micro:bit soll auf den LEDs eine zufällige Zahl zwischen 0 – 9 ausgeben, wenn man die Platine schüttelt.

Schüttele 3x und zähle zusammen. Wer macht die höchste Zahl in der Klasse?

Zahl 1: ..... Zahl 2: ..... Zahl 3: ..... Total: \_\_\_\_\_ (max. 27)

Speichern: *zufallszahl.hex*

### Auftrag 2: Würfel-Zufall

Der micro:bit soll auf den LEDs eine zufällige Augenzahl (wie beim Würfel) zwischen 1 und 6 ausgeben, wenn man den Knopf A auf der Platine drückt.

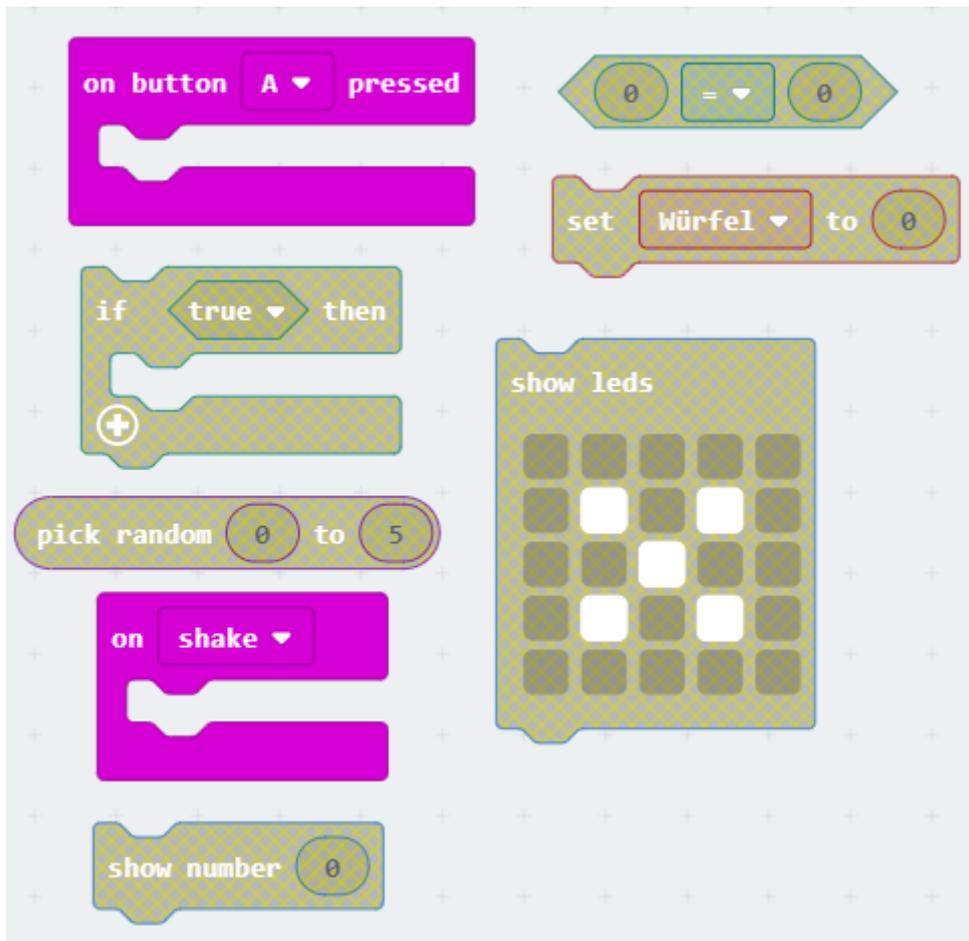
Betätige 3x den A-Knopf und zähle zusammen. Wer macht die höchste Zahl in der Klasse?

Zahl 1: ..... Zahl 2: ..... Zahl 3: ..... Total: \_\_\_\_\_ (max. 18)

Tipp: Du brauchst hier eine Variable «Würfel» (Name frei wählbar).

Hilfe: siehe Blöcke auf nächster Seite:



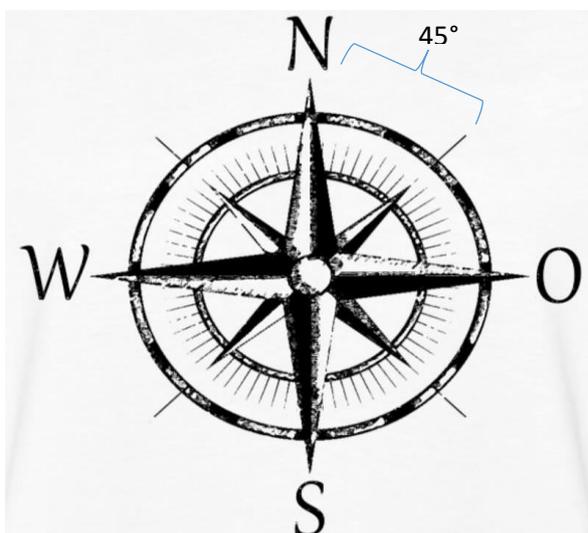


Speichern: wuerfelaugen.hex

### Auftrag 3: Kompass

Der micro:bit soll als Kompass dienen und auf den LEDs Richtungsangaben für N, O, S, W, ausgeben, wenn man die Platine in die entsprechende Himmelsrichtung dreht.

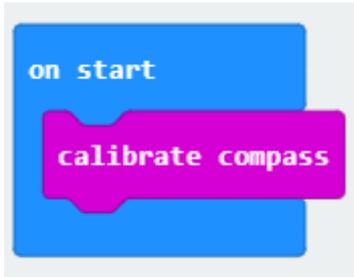
Hilfe: überlege, zwischen welchen Gradzahlen du welchen Pfeil zeigen willst.



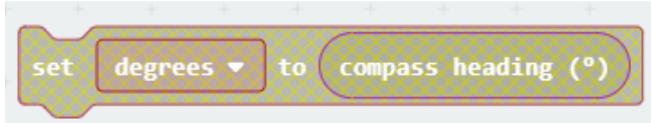
N:  $< 45^\circ$  oder  $> 315^\circ$   
 O:  $< 135$   
 usw.



Programmiere zuerst folgenden Block. Somit muss man vor Beginn immer zuerst kalibrieren.



Weiterer Tipp:



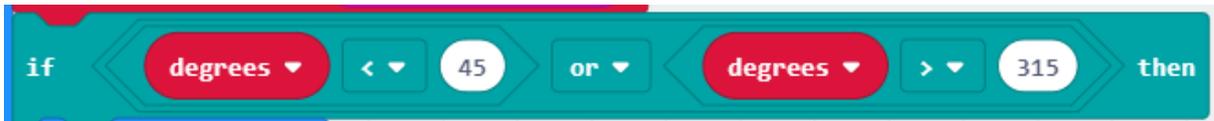
**Kalibrierung:** <https://support.microbit.org/support/solutions/articles/19000008874-calibrating-the-micro-bit-compass-what-does-it-mean-when-the-micro-bit-says-tilt-to-fill-screen>

**Erst danach läuft das Programm!**

Tipp: Du brauchst eine Variable für dieses Programm. Wir nennen sie «degrees».

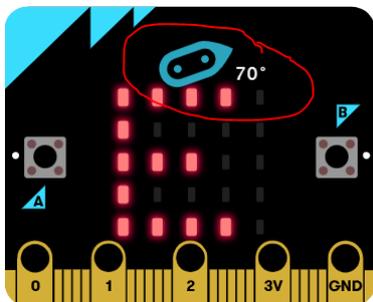
Wenn du bei Bedingung auf das PLUS klickst, kannst du weitere Bedingungen anfügen (elif).

Hilfe für Nord-Pfeil (hier muss auch noch ein «or» rein):



Die „or“ Bedingung braucht es bei den anderen Himmelsangaben nicht.

Teste dein Programm mit dem Simulator:



Ausbauvariante:  
Verfeinerungen mit Anzeigen NW, SW, SE, NE

Speichern: *kompass.hex*

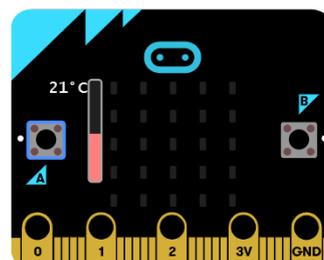
### Auftrag 4: Thermometer

Der micro:bit soll als Thermometer dienen und auf den LEDs die aktuelle Temperatur anzeigen, wenn du die A-Taste betätigst. Die Anzeige soll hinten noch «GRAD C» zeigen und 3x wiederholt werden.

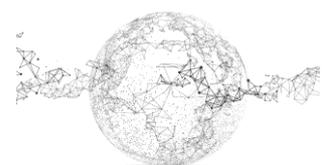
Tipp: Du brauchst eine Variable für dieses Programm.

Wir nennen sie «degrees».

Tipp:



Speichern: *thermometer.hex*

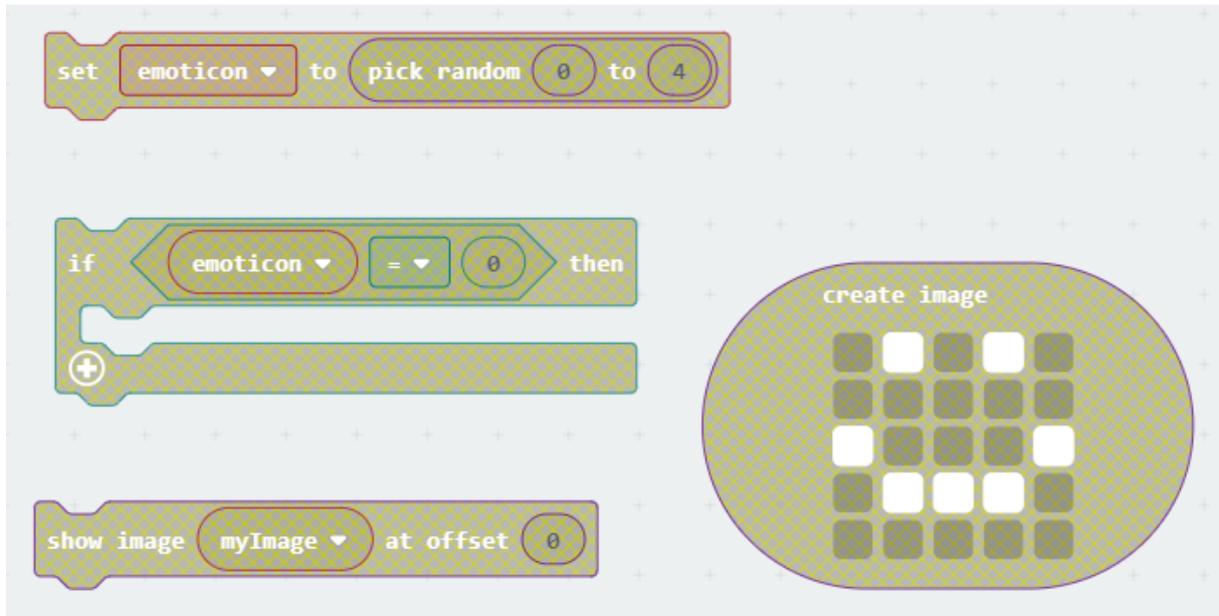


### Auftrag 5: Zufalls-Icons

Der micro:bit stellt dir die Frage: HOW ARE YOU? Wenn du Die Taste A drückst kommt ein zufälliges Icon. Die Icons (deren 5) sollst du selber erstellen. Bei B-Taste wird der Bildschirm gelöscht und Frage kommt erneut.

Tipp: Du brauchst eine Variable namens «image». Du brauchst «pick random» für zufällige Auswahl (siehe *Math*).

#### Einige Hilfen:



Speichern: *emoti-meter.hex*

### Auftrag 6: Stepcounter

Der micro:bit soll deine Schritte zählen und auf dem Display anzeigen. Die A-Taste soll den Zähler auf 0 setzen, die B-Taste, das Resultat der Schritte anzeigen. Bei jedem Schritt wird nach oben gezählt.

Falls du den Zähler (Variable: «counter») am Fuss testen willst, Sorge für eine sichere Befestigung (auch der Batterie). Die Platine darf keinen Schaden nehmen!

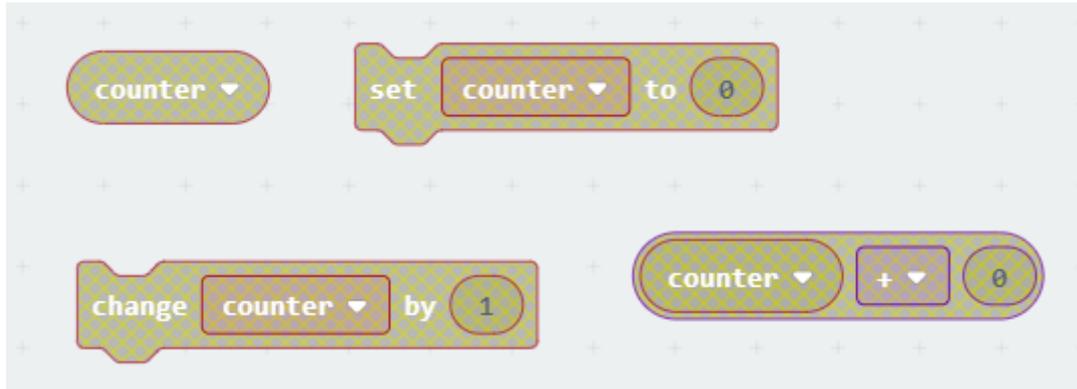


Tipp: grosse Schritte machen und hoch / Achtung: Du machst mit beiden Füßen Schritte. Was heisst das für den Zähler?

Probier aus, wie und wie schnell du gehen musst, damit es richtig zählt.



Hilfe:



Speichern: *stepcounter.hex*

### Auftrag 7: eigenes Projekt

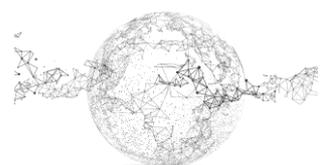
Erstelle dein eigenes «Ding». 😊 Ideen: <https://microbit.org/de/ideas/>

### Werde micro:bit Member!

Erstelle dir ein gratis Login!

Hol dir coole Ideen, speichere deine Projekte online.

<https://make.techwillsaveus.com/bbc-microbit>



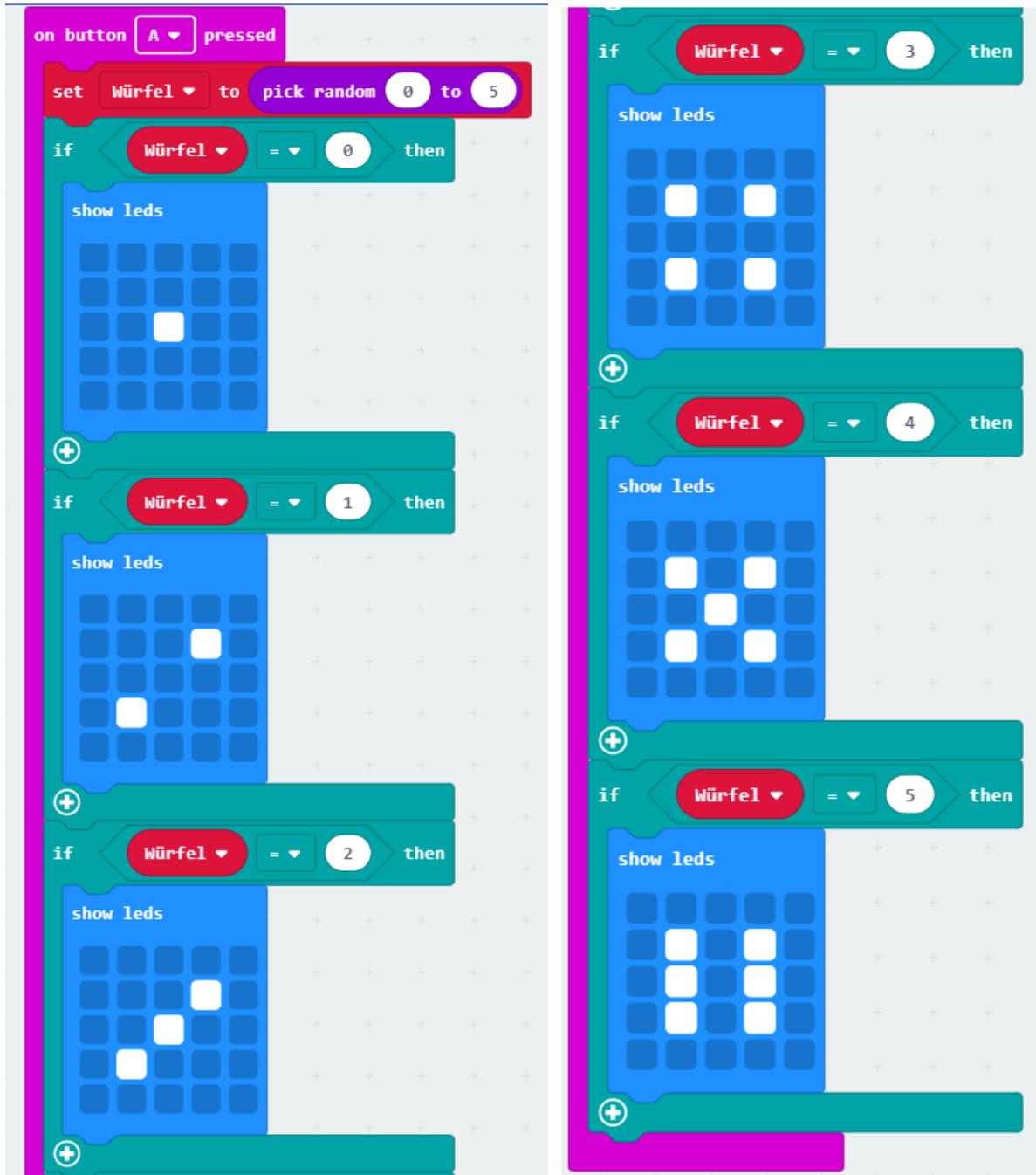
## Lösungen:

### Auftrag 1: Zufallszahl



```
on shake
  show number (pick random 0 to 9)
```

### Auftrag 2: Würfel



```
on button A pressed
  set Würfel to (pick random 0 to 5)
  if (Würfel = 0) then
    show leds
  if (Würfel = 1) then
    show leds
  if (Würfel = 2) then
    show leds
  if (Würfel = 3) then
    show leds
  if (Würfel = 4) then
    show leds
  if (Würfel = 5) then
    show leds
```



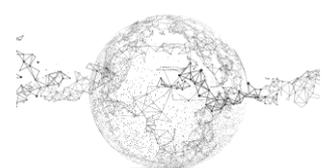
### Auftrag 3: Kompass

```
on start
  calibrate compass

forever
  set degrees to compass heading (°)
  if degrees > 315 or degrees < 45 then
    show string "N"
  else if degrees < 135 then
    show string "O"
  else if degrees < 225 then
    show string "S"
  else
    show string "W"
```

### Auftrag 4: Thermometer

```
on button A pressed
  set degrees to temperature (°C)
  repeat 3 times
    do
      show string " Grad C"
      show number degrees
```



### Auftrag 5: Emoti-Meter

```

on button A pressed
  set emoticon to pick random 0 to 4
  if emoticon = 0 then
    create image
    show image at offset 0
  else if emoticon = 1 then
    create image
    show image at offset 0
  else if emoticon = 2 then
    create image
    show image at offset 0
  else if emoticon = 3 then
    create image
    show image at offset 0
  else if emoticon = 4 then
    create image
    show image at offset 0

on button B pressed
  clear screen
  show string "HOW ARE YOU?"

on start
  show string "HOW ARE YOU?"
  
```



## Auftrag 6: Schrittzähler

```
on shake
  change counter by 2
  show number counter

on button B pressed
  show number counter

on button A pressed
  set counter to 0
  show number counter
```

The image shows three event-driven code blocks for a micro:bit. The first block, triggered by a 'shake' event, increments a 'counter' variable by 2 and then displays its value. The second block, triggered by button 'B' being pressed, displays the current value of the 'counter'. The third block, triggered by button 'A' being pressed, resets the 'counter' to 0 and then displays its value.

