

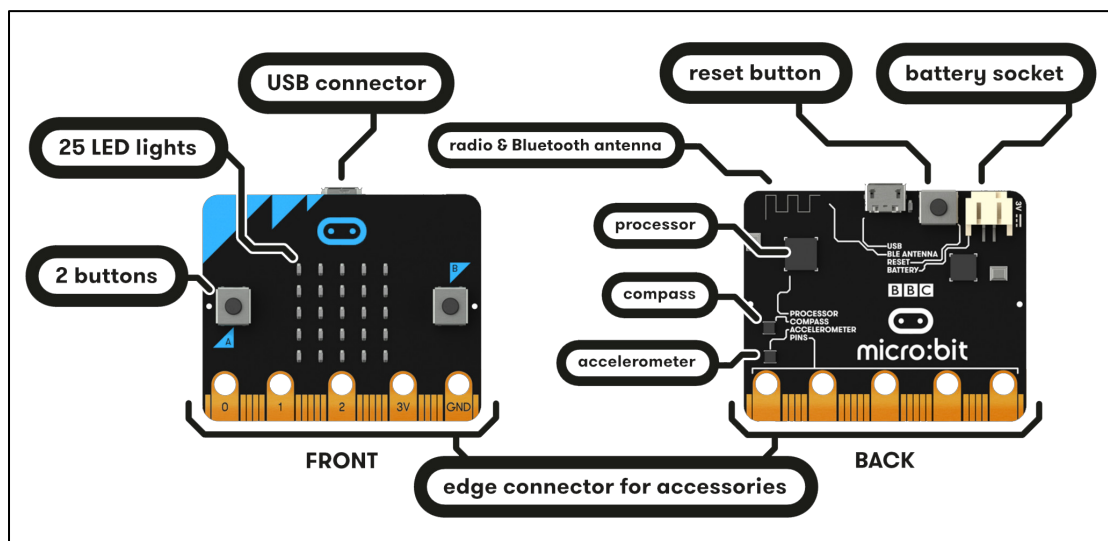
Den micro:bit im Unterricht einsetzen Einführung und Grundlagen

1. Allgemein

Der *micro:bit* ist ein von BBC entwickelter und programmierbarer Mikrocomputer. 2015 wurde er von BBC vorgestellt, seit 2016 ist er offiziell erhältlich. Der *micro:bit* wurde speziell für den Informatikunterricht entwickelt, um die Schulbildung in diesem Bereich zu verbessern. Verschiedene webbasierte Entwicklungsumgebungen ermöglichen es, diesen Mikrocomputer zu programmieren.

1.1. Hardware

Hardwaretechnisch bietet der *micro:bit* einen 16 MHz Mikrocontroller vom Typ ARM Cortex-M0 als Prozessor sowie 256 KB Flash-Speicher und 16 KB statischen RAM. Auf der Rückseite des *micro:bits* findet man einen Micro-USB-Anschluss, einen Anschluss für eine externe Stromquelle, einen Reset-Button, einen Beschleunigungssensor und ein Magnetometer sowie Funk- beziehungsweise Bluetoothempfang. Auf der Vorderseite befinden sich zwei Knöpfe A und B, die gleichzeitig gedrückt werden können und somit den Knopf AB bilden sowie 25 LEDs. Auf der Unterseite des *micro:bits* befindet sich ein Platinenstecker, durch den weitere Sensoren verbunden werden können.



Die einzelnen Komponenten des micro:bit.

Quelle: <https://microbit.org/guide/features>

1.2. Preis

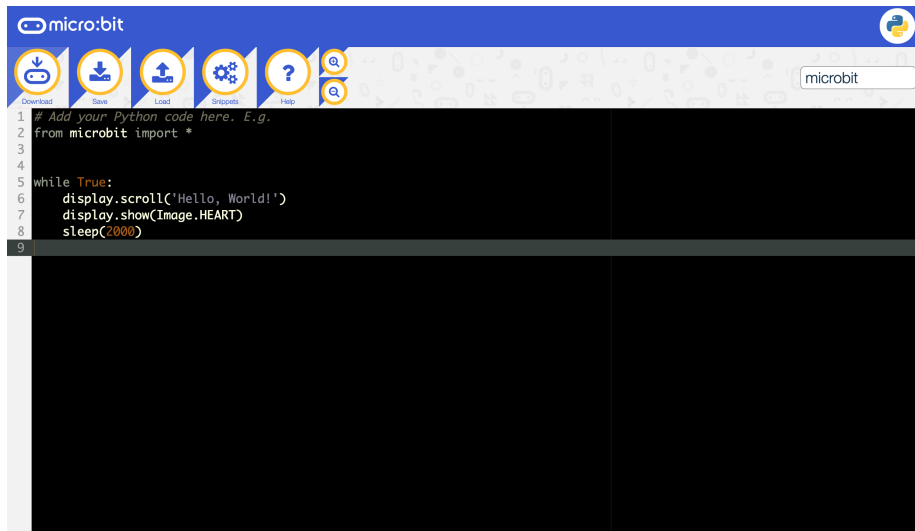
Der *micro:bit* alleine ist für ca. 20€, im Set mit einer Akkuhalterung bereits ab ca. 23€ erhältlich (Preise von Amazon: Stand Februar 2019).

1.3. Software

Der Code für den *micro:bit* kann direkt in verschiedenen Webeditoren (siehe Punkt 2. Programmierungsumgebungen) erstellt werden, wo der fertige Code auch kompiliert und heruntergeladen werden kann. Anschließend muss die kompilierte Datei (.hex) auf den *micro:bit* via USB-Kabel oder Bluetooth geladen werden.

2. Programmierumgebungen

2.1. Python-Editor



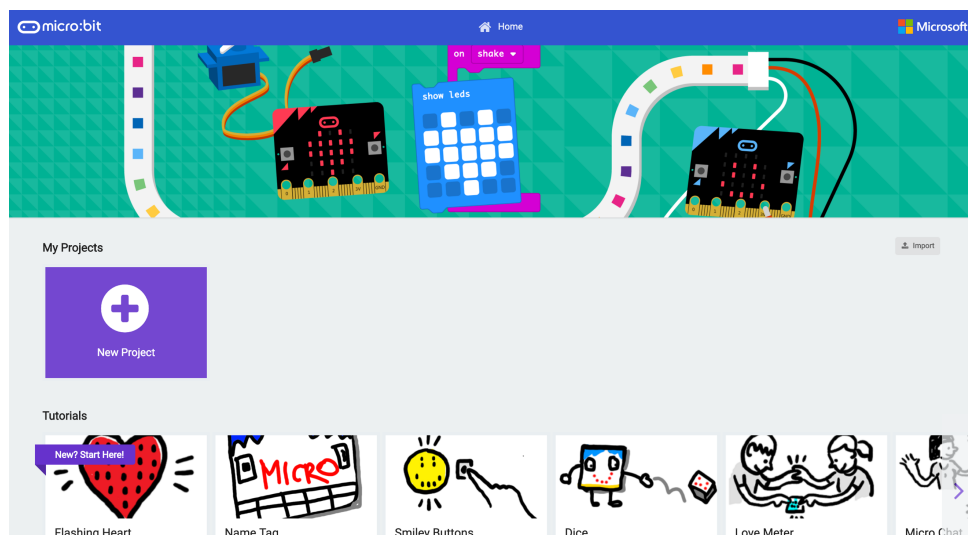
Python-Editor

Editor: <https://python.microbit.org/v/1.1>

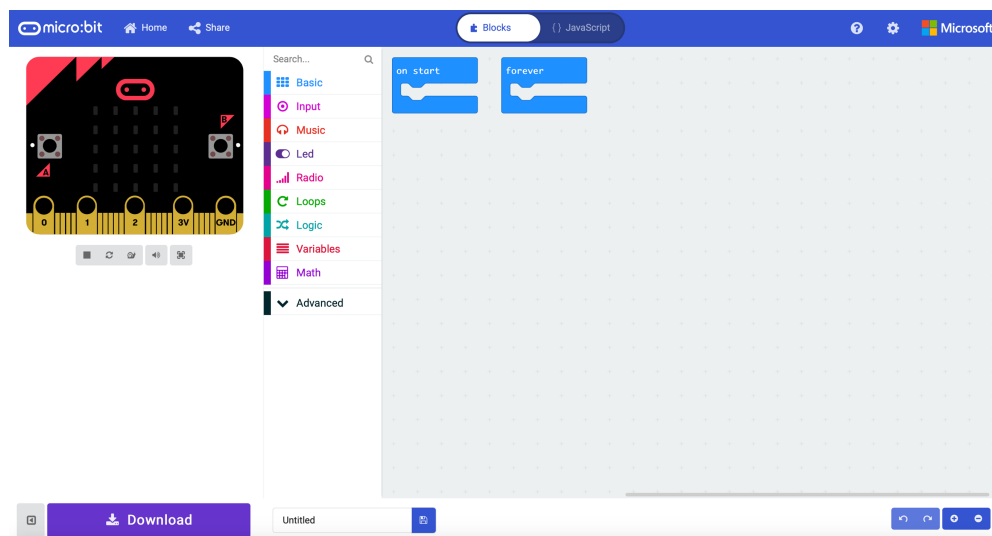
Tutorials und Hilfestellung:

<https://microbit-micropython.readthedocs.io/en/latest/index.html>

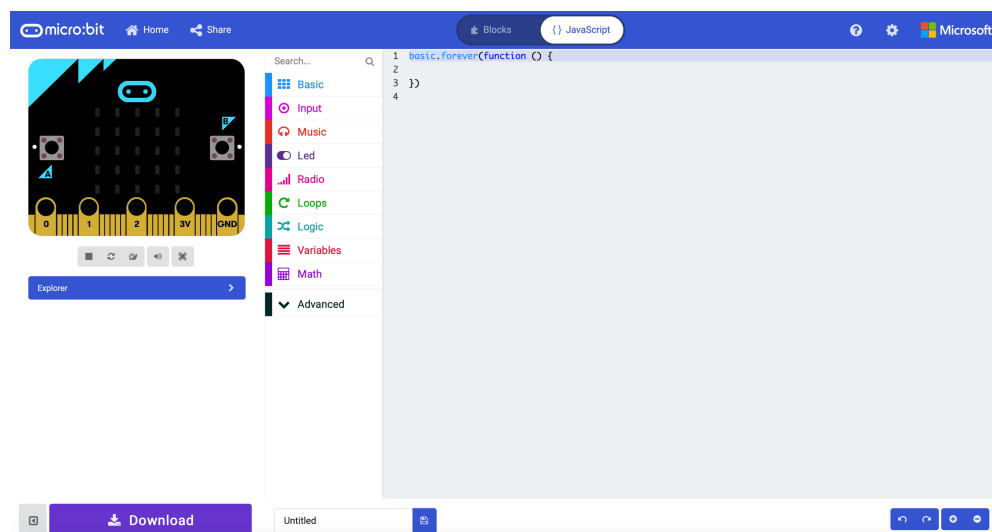
2.2. Microsoft MakeCode-Editor



Startseite Microsoft MakeCode



Block-Editor



JavaScript-Editor

Editor: <https://makecode.microbit.org>

Tutorials und Hilfestellung: <https://makecode.microbit.org/docs>

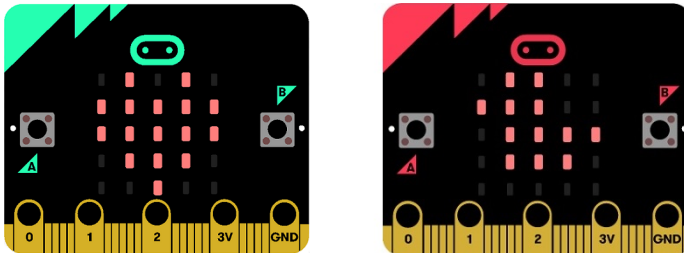
3. Übungen

3.1. Grundlagen und Ein-/Ausgabe

1. Mach dich mit dem Editor von *micro:bit MakeCode* vertraut:
<https://makecode.microbit.org/#editor>
2. Schreibe einen Code für den *micro:bit*, der die Ausgabe eines beliebigen Symbols implementiert:
 - a. Wenn Knopf „A“ gedrückt wird, sollen die LEDs ein Symbol darstellen.

- b. Wenn Knopf „B“ gedrückt wird, sollen die LEDs ein anderes Symbol darstellen.
3. Lade den fertigen Code auf den *micro:bit*.

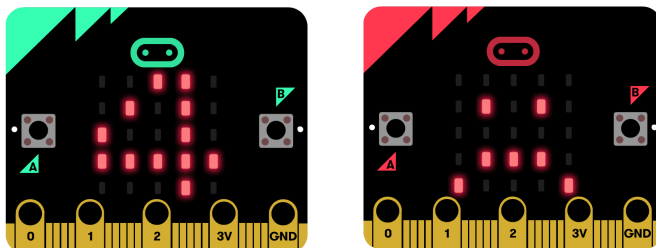
Beispiele:



3.2. Variablen, Verzweigungen und Schleifen

1. Eine Variable soll in einer Schleife von 0 bis 20 erhöht werden.
2. Schreibe einen Code für den *micro:bit*, der folgendes Verhalten implementiert:
 - a. Eine Variable wird von 0 bis 20 erhöht.
 - b. Wenn eine gerade Zahl an der Reihe ist, soll die Zahl selbst ausgegeben werden.
 - c. Wenn eine ungerade Zahl an der Reihe ist, soll ein trauriger Smiley ausgegeben werden.
3. Teste dein Programm im Simulator.
4. Lade den fertigen Code auf den *micro:bit*.

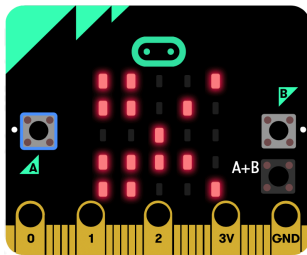
Beispiele:



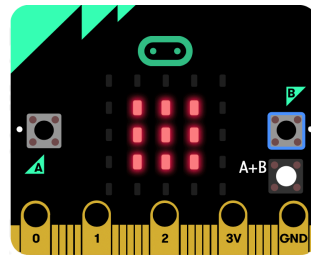
3.3. Schere, Stein, Papier – Anfänger

1. Schreibe einen Code für den *micro:bit*, der das Spiel *Schere, Stein, Papier* implementiert.
 - a. Wenn Knopf „A“ gedrückt wird, sollen die LEDs eine Schere darstellen.
 - b. Wenn Knopf „B“ gedrückt wird, sollen die LEDs einen Stein darstellen.
 - c. Wenn beide Knöpfe gedrückt werden, sollen die LEDs Papier darstellen.
2. Lade den fertigen Code auf den *micro:bit*.
3. Teste dein Programm ausführlich.

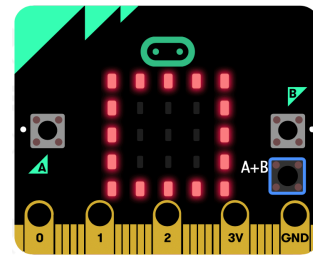
Die Symbole:



Schere



Stein

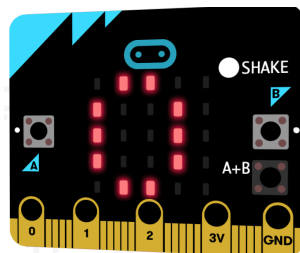


Papier

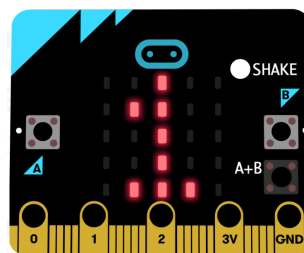
3.4. Schere, Stein, Papier – Fortgeschritten

1. Schreibe einen Code für den *micro:bit*, der das Spiel *Schere, Stein, Papier* implementiert.
 1. Wenn Knopf „A“ gedrückt wird, sollen die LEDs eine Schere darstellen.
 2. Wenn Knopf „B“ gedrückt wird, sollen die LEDs einen Stein darstellen.
 3. Wenn beide Knöpfe gedrückt werden, sollen die LEDs Papier darstellen.
2. Die Anzahl der Siege soll gespeichert und angezeigt werden.
 1. Durch Schütteln wird die Anzahl der Siege angezeigt.
 2. Durch Neigen nach rechts wird diese Zahl um 1 erhöht.
 3. Durch Neigen nach links wird diese Zahl um 1 verkleinert.
3. Beachte auch, ...
 1. ...dass die Zahl der Siege nicht kleiner als 0 sein kann ;-)
 2. ...dass die Zahl nicht nur erhöht/ verkleinert, sondern auch angezeigt wird.

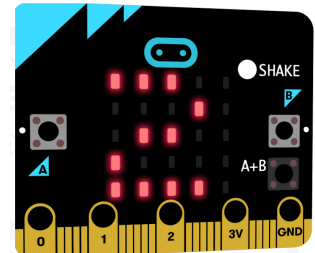
Beispiele:



Neigen nach links



Schütteln



Neigen nach rechts

3.5. Senden und Empfangen – Anfänger

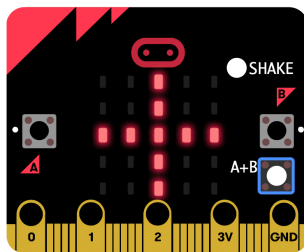
1. Suche dir einen Teampartner.
2. Stellt sicher, dass im Editor Funk (bzw. Radio) aktiviert ist.
3. Wählt einen Kanal aus (Zahl zwischen 0 und 255).
4. Macht untereinander aus, wer Sender und wer Empfänger ist.
 - a. Der Sender implementiert einen Code, der beim Drücken von Knopf A, B oder AB Daten überträgt (z.B.: eine Zahl oder einen Text).

- b. Der Empfänger implementiert einen Code, der wenn er die gesendeten Daten empfängt, diese auch ausgibt.

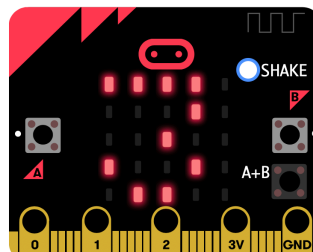
3.6. Senden und Empfangen – Fortgeschritten

1. Erstelle ein Programm, sodass sich dein *micro:bit* mit einem anderen *micro:bit* über Funk verbinden kann.
2. Implementiere die Funktionen eines Rechners (+/-)
 - a. Knopf A = 1. Operand um 1 erhöhen.
 - b. Knopf B = 2. Operand um 2 erhöhen.
 - c. Knopf A + B = Operation wählen (+ oder -).
3. Beachte auch, dass die entsprechende Auswahl auch immer angezeigt wird.
4. Der *micro:bit* soll anschließend das Ergebnis per Funk an einen anderen *micro:bit* schicken, der das Ergebnis ausgibt.

Beispiele:



Operand wählen



Ergebnisübertragung

3.7. *micro:bit* App und Bluetooth

1. Ladet dir die App für den *micro:bit* auf dein Smartphone.
2. Verbinde den *micro:bit* über Bluetooth mit deinem Smartphone. Folge dazu den Anweisungen in der App.
3. Schreibe in der App einen Code für den *micro:bit*, der die Ausgabe eines beliebigen Symbols implementiert:
 - a. Wenn Knopf „A“ gedrückt wird, sollen die LEDs ein Symbol darstellen.
 - b. Wenn Knopf „B“ gedrückt wird, sollen die LEDs ein anderes Symbol darstellen.
4. Lade den fertigen Code anschließend direkt von deinem Smartphone auf den *micro:bit*.

3.8. Handysteuerung via Bluetooth

1. Schreibe einen Code für den *micro:bit*, der dir ermöglicht, die Kamera deines Smartphones mit dem *micro:bit* zu steuern.
2. Beim Start soll...
 - a. ...ein Symbol angezeigt werden.
 - b. ...für 1 Sekunde pausiert werden.
 - c. ...der Bildschirminhalt gelöscht werden.
3. Wenn der Knopf A gedrückt wird soll...
 - a. ...ein Symbol angezeigt werden.
 - b. ...für 1 Sekunde pausiert werden.
 - c. ...der Fotomodus gestartet werden.
 - d. ...für 1 Sekunde pausiert werden.
 - e. ...ein Foto aufgenommen werden.
 - f. ...der Bildschirminhalt gelöscht werden.
4. Beachte auch, dass ausgegeben wird, ob die Verbindung mit deinem Smartphone erfolgreich war.
 - a. Wenn eine Verbindung hergestellt werden konnte, soll der *micro:bit* „C“ (für *connected*) ausgeben.
 - b. Wenn die Verbindung unterbrochen wurde, soll der *micro:bit* „D“ (für *disconnected*) ausgeben.
5. Lade den fertigen Code auf den *micro:bit*.
6. Teste dein Programm ausführlich.

4. Anhang

