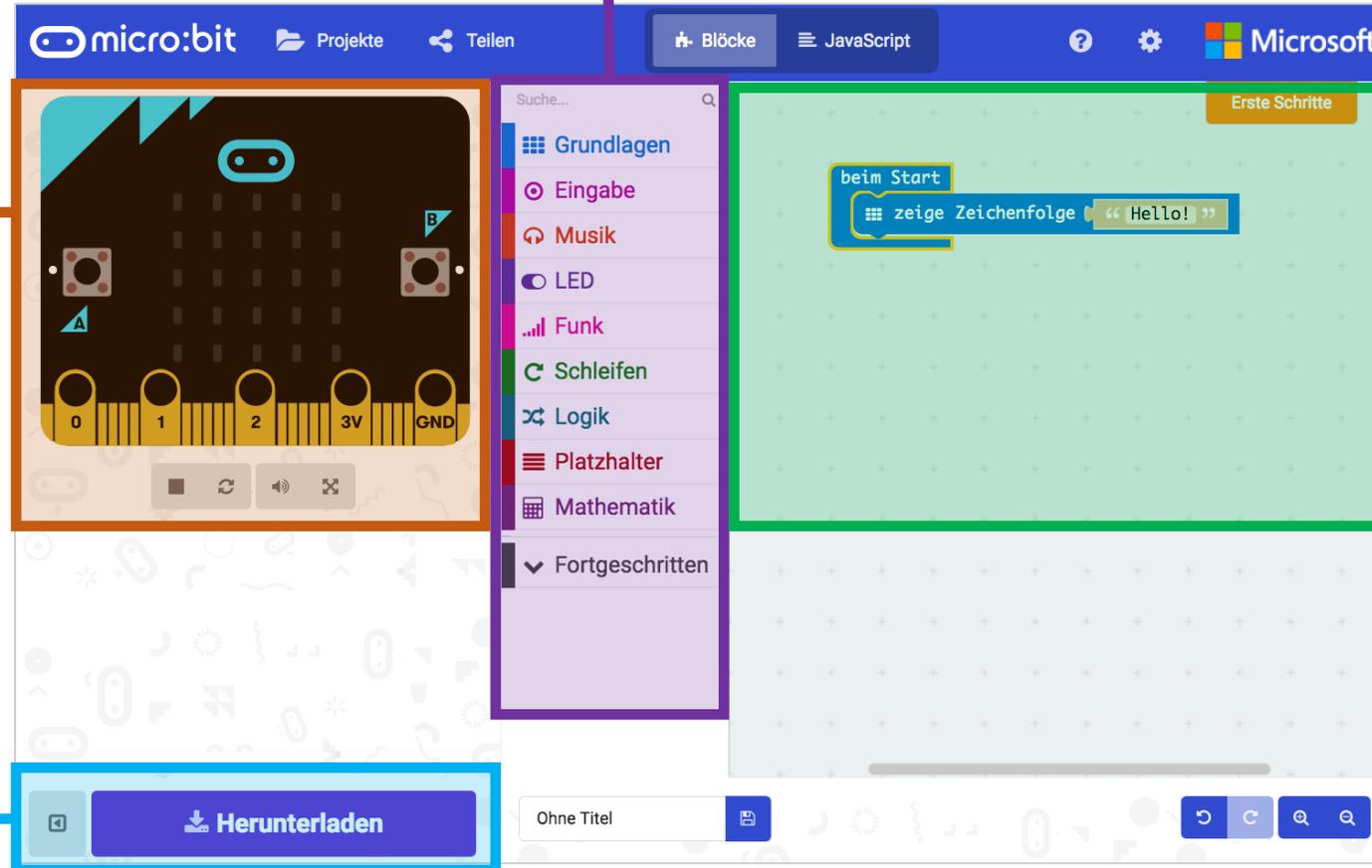


EINSTIEG: Die Oberfläche von pxt.microbit.org

micro:bit Simulation

Programmausgabe wird simuliert. Der Simulator reagiert auf Eingaben (z.B. Knöpfe)



Code-Repertoire

Kategorien mit den zur Verfügung stehenden Programmblöcke

Programmcode

Effektiver Programmcode. Die Codeblöcke werden aus dem Code-Repertoire hierhin gezogen

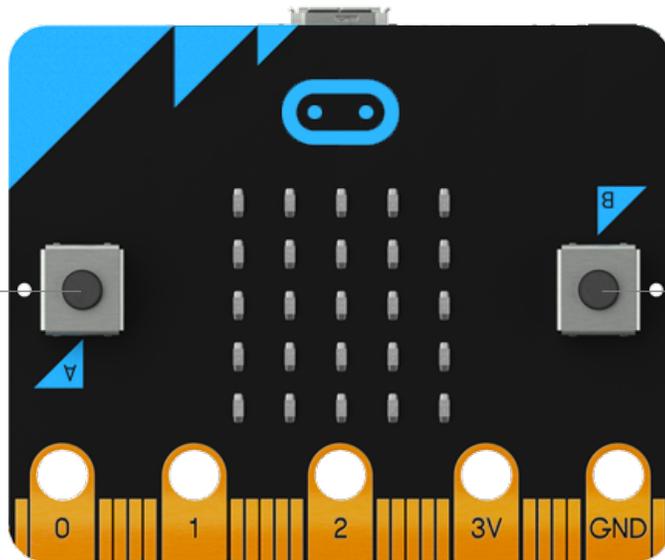
Download Maschinencode

Heruntergeladene Datei wird im Explorer/Finder via USB auf die Platine kopiert

TASK I: Der Welt Hallo Sagen

Knopf A:

Textanzeige LED:
Hallo Welt



Knopf B:

Smiley anzeigen

TASK

- › Nach Knopfdruck Displayanzeige aktivieren / wechseln

INHALTE

- › Einstieg in die PXT Programmumgebung
- › Auf Ereignisse reagieren

Benötigte Blöcke

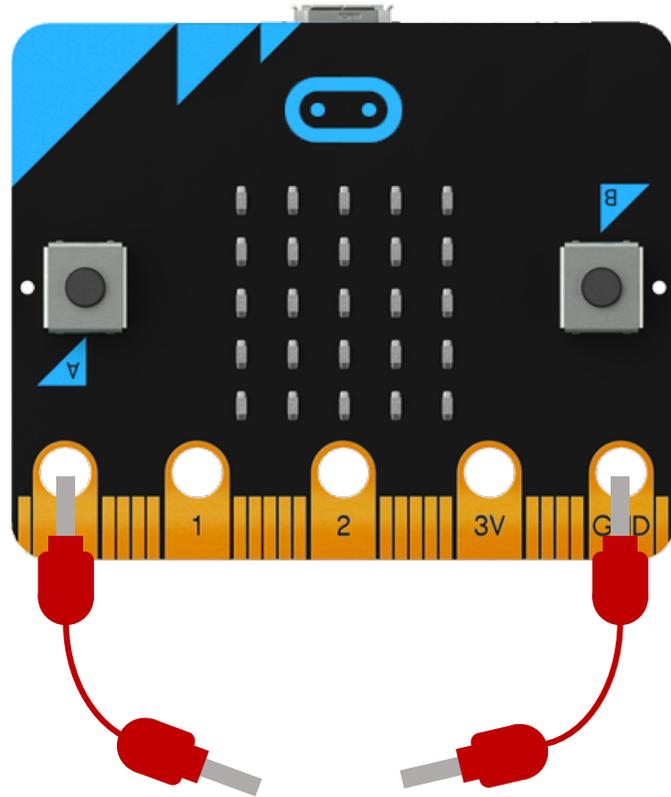
zeige Zeichenfolge []

zeige LEDs

wenn Knopf [] gedrückt

halte Animation an

TASK II: Pin-Kontakt (Bewässerungsmelder)



Kein Kontakt
Zeige Anti-Smiley

Kontakt
Zeige Smiley

TASK

- › Durch den Pinkontakt ändert sich die Displayanzeige von einem Anti-Smiley zu einem Smiley

INHALTE

- › Nutzung der analogen Pins
- › Einsatz einer Schleife

Benötigte Blöcke

dauerhaft []

zeige LEDs

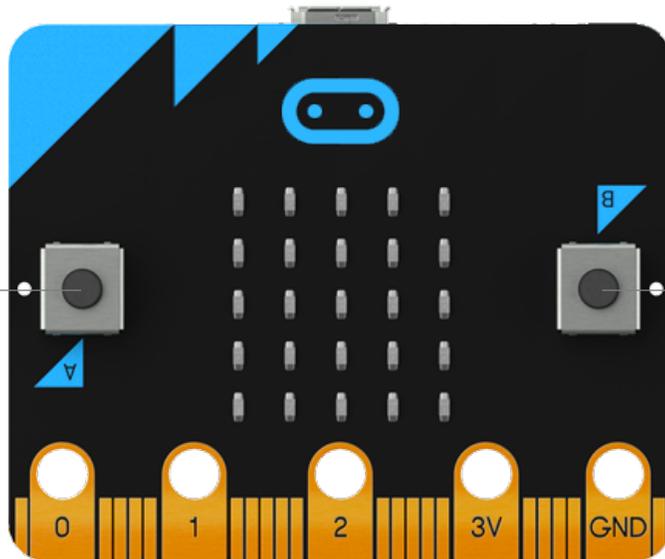
pin [] ist gedrückt

während [] mache []

TASK III: Zähler

START:

Zahl in Display **auf 0**



Knopf A:

Zahl in Display **-1**

Zusatz:

Minimalwert = **0**

Knopf B:

Zahl in Display **+1**

Zusatz:

Minimalwert = **9**

SHAKE:

Zahl in Display = **Zufallszahl**

TASK

- › Eine Variable mit dem Namen «Nummer» wird generiert (Platzhalter → neuen Platzhalter)
- › Je nach Knopfdruck wird der Wert der Variabel erhöht oder reduziert
- › Ein «Shake» definiert die Variable zufällig neu (1–9)

INHALTE

- › Variablen einsetzen und deren Wert verändern
- › Einsatz von «Wenn – Dann» Bedingungen
- › Fehlerkorrektur: Eingaben vor der Verarbeitung validieren

Benötigte Blöcke

beim Start

dauerhaft

wenn Knopf [] gedrückt

wenn [geschüttelt]

wenn [] dann

ändere [Platzhalter] auf []

[] + []

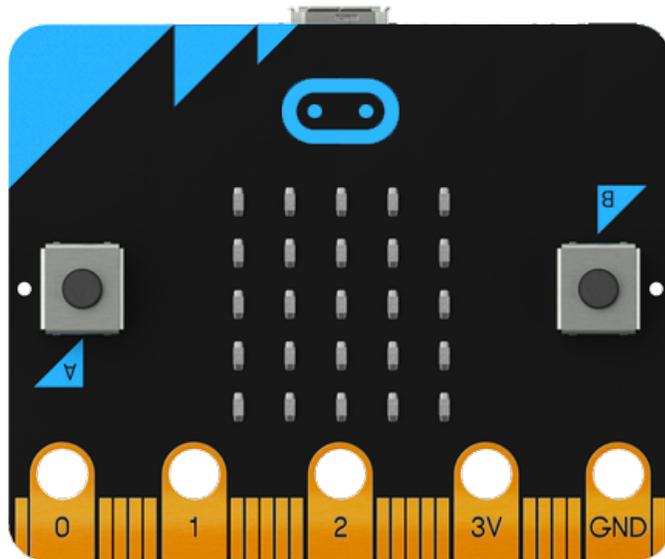
[] - []

wähle eine zufällige Zahl...

TASK IV: Kompass

Drehen der Platine:

Je nach Richtung **N**, **O**,
S oder **W** anzeigen



ACHTUNG:

Zu Beginn wird der Kompass des micro:bit kalibriert. Dazu drehen sie die Platine bis ein Kreis auf dem Display entsteht

TASK

- › Je nach Ausrichtung der Platine wird die Himmelsrichtung angezeigt
- › Die Ausrichtung wird in Grad gemessen und wird in einer Variable gespeichert
- › 0 Grad entspricht Norden. «N» soll entsprechend zwischen 315° und 360° sowie zwischen 0° und 45° angezeigt werden

INHALTE

- › Variablen einsetzen und deren Wert durch einen Sensorwert bestimmen
- › Einsatz von «Wenn – Dann – Ansonsten» Bedingungen
- › Einsatz von logischen Operatoren

Benötigte Blöcke

dauerhaft

zeige Zeichenfolge []

Kompassausrichtung

wenn [] dann [] ansonsten

[] und []

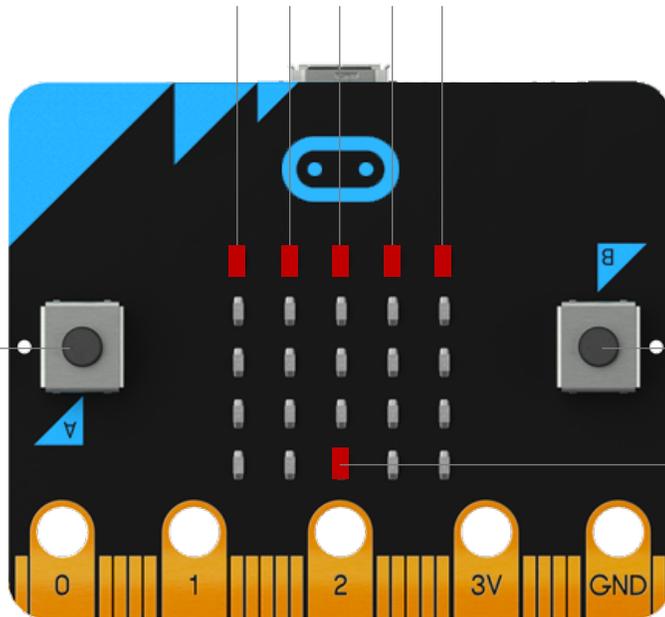
[] oder []

ändere [Platzhalter] auf []

TASK V: Game Astroids

In jeder Spalte fallen nach einer Zufallszeit (0-5sec) Astroiden in der Y-Achse herunter

Knopf A:
Spieler nach **Links**



Knopf B:
Spieler nach **Rechts**

Spieler, Start
Sprite bei [2, 4]

**Erfolgreiches
Ausweichmanöver:**
Punkte **+1**

**Kollision mit
Astroiden:**
Game Over

TASK

- > Variablenwerte zu Begin initiieren (z.B. Punkte = 0, spiel_an = wahr)
- > Routine für Spielerbewegung entwickeln
- > Routine für ersten (Y=0) fallenden Astroiden entwickeln
- > Kollisionsereignis entwickeln
- > Routine für restliche Astroiden (Y=1 bis Y=4) übertragen

INHALTE

- > Einsatz von Game-Blöcken (z.B. Sprite)
- > Einsatz von «Wenn – Dann» Bedingungen
- > Einsatz von Schleifen

Benötigte Blöcke (nur Game-Block)

erzeuge Sprite an Position x: [] y: []

[Platzhalter] y

[Platzhalter] set y to []

[Platzhalter] change y by 1

set score []

spiel beendet