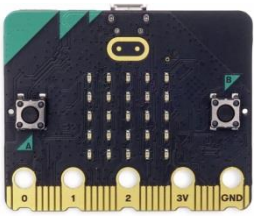
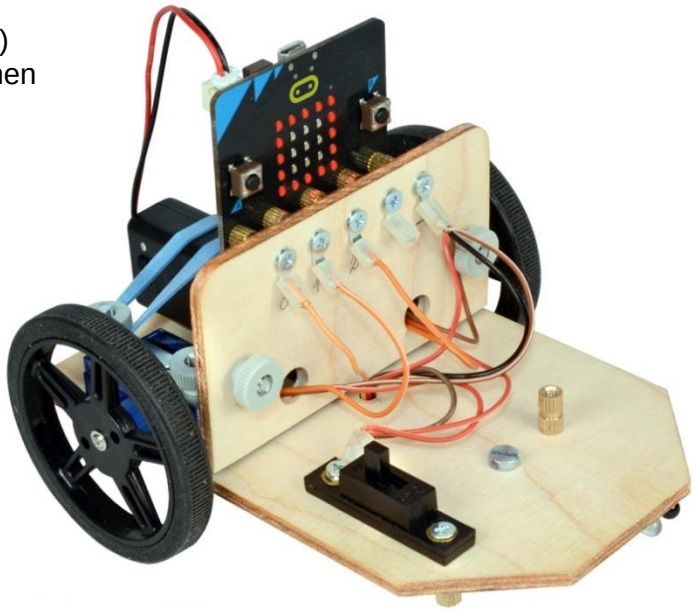
PROGRAMMIERANLEITUNG
Micro:bit SERVOBOT mit IR-Sensor



Dieser ServoBot wird von zwei Micro-Rotationsservos (360°) mit Rädern (Ø 60 mm) angetrieben. Weiters enthält der ServoBot einen Infrarot-Abstandssensor, um bei Bedarf Hindernisse erkennen zu können.
Die Programmieranleitung enthält zahlreiche grafische Programmierbeispiele (MakeCode).
Wir empfehlen den ServoBot für leicht fortgeschrittene Micro:bit-Programmierer.
Mit einem zweiten Micro:bit und dem Activity-Board (Art. Nr. 102525) kann der ServoBot sogar ferngesteuert werden.
Micro:bit, Batteriebox (2 x AAA), USB-Kabel und Batterien (2 x AAA und 4 x Mignon) sind nicht in der Werkpackung enthalten!



Grundlagen

1. Allgemeine Hinweise:

- Baue die Werkpackung **ServoBot + IR-Sensor** laut beiliegender Aufbauanleitung zusammen. Befestige einen **Micro:bit** mit fünf Senkkopfschrauben M3 x 8 mm an den Messinghülsen, setze zwei neue **AAA-Batterien** (je 1,5 V) in die **Micro:bit-Batteriebox** (3 V) und vier **Mignon-Batterien** für die Servos in den Batteriehalter auf der **Unterseite** des ServoBot.
  Achtung: Schwache Batterien können zu Fehlfunktionen der Servos und des IR-Sensors führen!
- Aufladbare Batterien (zB. NiMH, NiCD) haben eine Spannung von 1,2 Volt und sind daher nur bedingt einsetzbar. Ideal und nachhaltig wäre aber die Verwendung von **USB-Powerbanks**.
- Halte den Micro:bit von Feuchtigkeit fern und berühre ihn möglichst nicht an den Kontakten. Achte darauf, dass die Kontakte nicht kurzgeschlossen werden!

2. Anforderungen:

Um den Micro:bit in Betrieb zu nehmen, braucht man:

- ein Laptop oder einen PC mit Windows 10 (8, 7) oder Mac (OSX oder Linux)
- ein Micro-USB-Kabel zum Anschluss des Micro:bit an den Computer
- einen Internet-Zugang (Chrome, Edge, Firefox …) - **Aber**: Für den Betrieb ohne Internet gibt es eine **App**: https://makecode.microbit.org/offline-app

Der Micro:bit kann auch über eine App mit einem **Tablet / iPad oder Smartphone** via Bluetooth programmiert werden. Dazu muss aber der Micro:bit mit diesen Geräten **gekoppelt** werden.
Eine **Video-Anleitung** dafür findet man auf der **Homepage** von **microbit.org** unter:
https://microbit.org/get-started/user-guide/mobile/#pair-your-micro:bit-with-the-app

3. Den Micro:bit vorbereiten:

Schließe den Micro:bit über ein Micro-USB-Kabel an einen freien USB-Anschluss des PCs. Das Kabel dient sowohl zur Stromversorgung des Micro:bit als auch zur Datenübertragung. Der Micro:bit erscheint im Windows Explorer (PC) oder Filemanager (Mac) als Laufwerk mit dem Namen [MICROBIT] und einem Laufwerksbuchstaben, zB. [E:]. Der Micro:bit kann dann über dieses Laufwerk mit einer Programmdatei (*.hex) versorgt werden.

Bei neuen Micro:bits ist ein **Demo-Programm** vorinstalliert, das die Funktionen des Micro:bit zeigt und zu verschiedenen Aktivitäten aufruft, zB. Schütteln, Neigen, Taste drücken, usw. - Es wird später einfach durch eigene Programme überschrieben!

Bei Anschluss des Micro:bit an einen Computer muss die Batterieversorgung nicht getrennt werden, denn der Micro:bit schaltet automatisch auf USB-Versorgung um.

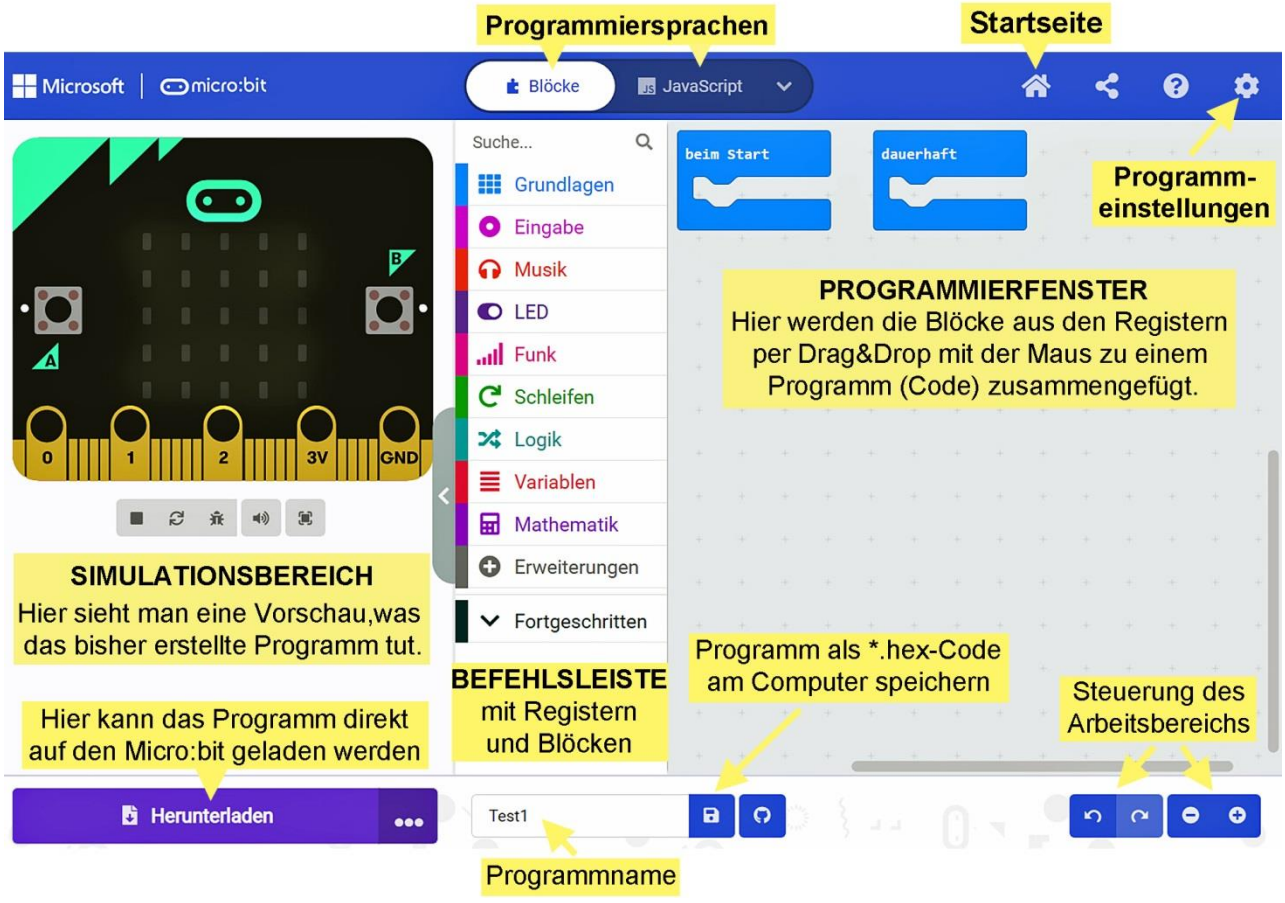
4. Der Makecode-Editor:

Zum Programmieren verwenden wir die grafische Programmierplattform **Makecode©** von Microsoft: https://makecode.microbit.org/. Eine grafische Programmierung ist ideal für Anfänger, die noch keine Programmiersprache kennen, denn sie ist intuitiv und leicht zu erlernen. **Makecode** läuft im Browser, daher braucht kein eigenes Programm installiert zu werden.

Programmierumgebung

1. Programmstart:

- Schließe den Micro:bit über ein **Micro-USB-Kabel** am Computer an.
- Der Micro:bit wird im Explorer als **Laufwerk** (zB. MICROBIT [E:]) angezeigt.
- Öffne den Browser (Chrome, Edge, Firefox …) und gib folgende **Programm-URL** ein: https://makecode.microbit.org/
- Wähle die Schaltfläche [Neues Projekt] aus und gib dem Projekt einen Namen (zB. Test1). Nun erscheint die **Programmieroberfläche**:



2. Programmbeschreibung:

Die Programmieroberfläche von **Makecode©** besteht aus drei Bereichen:
SIMULATIONSBEREICH, BEFEHLSLEISTE, PROGRAMMIERFENSTER

Im Simulationsbereich ist ein Micro:bit abgebildet, der das laufende Programm abspielt.

In der Befehlsleiste befinden sich verschiedenfarbige Register mit Blöcken zum Programmieren. Nach Anklicken der Register erscheinen diverse Blöcke, die man mit der Maus per Drag&Drop ins Programmierfenster ziehen kann. Die Blöcke erscheinen im Programmierfenster zuerst grau und nehmen ihre Originalfarbe erst wieder an, wenn sie richtig im Programm verankert sind. Die Blöcke lassen sich mit einem Klick der rechten Maustaste duplizieren und löschen oder man schiebt sie wieder zurück in die Befehlsleiste. Die Blöcke sind so geformt, dass sie nur

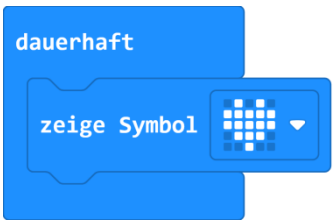
dann ineinander passen, wenn sie logisch zu den Programmbefehlen passen. Dadurch werden Programmierfehler stark reduziert. **Fortgeschrittene** können aber statt der grafischen **Block-Programmierung** auch **JavaScript** oder **Python** verwenden.

Nach einem Klick auf das Zahnradsymbol (rechts oben) können Einstellungen durchgeführt werden: zB. Sprache, Programme löschen, zusätzliche Blockregister einfügen usw.

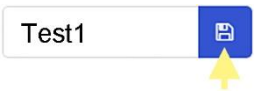
Ein Klick auf das Haus-Symbol (oben) öffnet die Startseite.

3. Ein erstes Testprogramm speichern:

Lösche den **Start-Block** [beim Start] durch Verschieben in den Registerbereich. Ziehe vom Register [Grundlagen] den **Symbol-Block »Herz«** in die **Block-Klammer »dauerhaft«**.



Klicke unten auf das **Diskettensymbol** neben dem **Programm-namen** (Test1). Das Programm wird nun lokal auf dem Computer als microbit-Test1.hex gespeichert.



4. Das Testprogramm auf den Micro:bit übertragen:

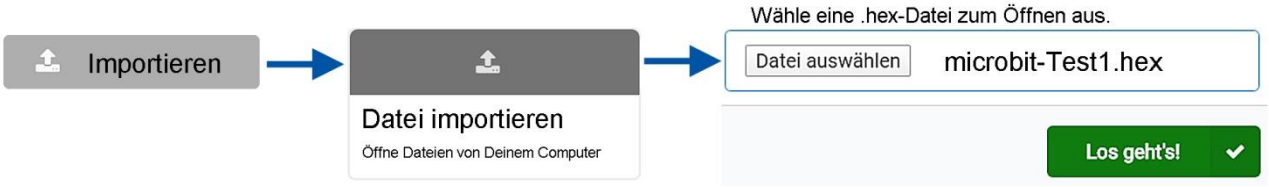
Die **Übertragung des Programms auf den Micro:bit** kann auf **zwei Arten** erfolgen:

- Öffne den Datei-Explorer und ziehe die Datei microbit-Test1.hex mit der Maus auf das Laufwerk [MICROBIT]. Bei der Übertragung blinkt zuerst ein gelbes Licht (Rückseite) und dann startet das Programm.
- Klicke erstmals im **Makecode-Editor** auf die Schaltfläche [Herunterladen], wähle das Laufwerk [MICROBIT] und klicke auf [Speichern]. Ab dem zweiten Mal kann jedes Programm durch einen einfachen Klick auf [Herunterladen] auf den Micro:bit übertragen werden.
- Mit der **Reset-Taste** auf der Rückseite des Micro:bit kann ein Programm neu gestartet werden.

5. Ein Programm (hex-Datei) importieren:

Um den **Programmcode einer hex-Datei** lesen und bearbeiten zu können, muss sie im Programmeditor **Makecode** geöffnet werden. Das kann auf **zwei Arten** erfolgen:

- Ziehe die entsprechende **hex-Datei** direkt **per Drag&Drop** vom Datei-Explorer auf das Programmierfenster von Makecode. Dort kann das Programm dann bearbeitet werden.
- Eine **hex-Datei** kann aber auch von der **Startseite von Makecode** importiert werden:
  Klicke auf die graue Schaltfläche [Importieren], dann auf [Datei importieren].
  Über [Datei auswählen] kann im Datei-Explorer die gewünschte hex-Datei gewählt werden.
  Nach einem Klick auf [Los geht's] wird das Programm auf dem Makecode-Editor geöffnet.

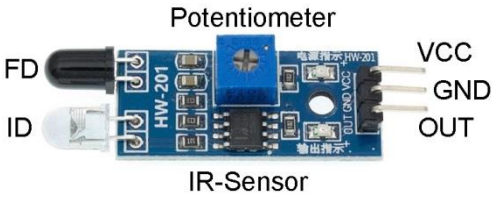


6. Rotations-Servo (360°):

Ein Servo ist ein elektronisch gesteuerter Getriebemotor. Im Gegensatz zu Winkel-Servos (180°) drehen sich **Rotations-Servos (360°)** kontinuierlich vor oder zurück. Ausgestattet mit Rädern eignen sich solche Servos sehr gut als Antrieb für kleine Roboterfahrzeuge.
Das Anschlusskabel besteht aus drei verschiedenfarbigen Drähten: **BRAUN** kommt an GND (-), **ROT** an 4,8 - 6 V (+) und **ORANGE** an **Pin 1** oder **2** des Micro:bit. Über die orange **Signalleitung** kann der Micro:bit die Geschwindigkeit und Drehrichtung der Servos steuern.
Da Servos in der Regel mit 4,8 V bis 6 V betrieben werden, benötigen sie eine separate Stromversorgung (zB. 4 x Mignon-Batterien).

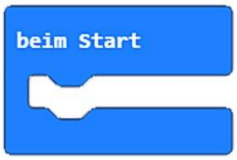
7. Infrarot-Sensor (IR):

Am **IR-Sensor** befindet sich eine **Infrarotdiode (ID)**, die unsichtbares Infrarotlicht abstrahlt. Nähert sie sich einem Gegenstand, dann wird das IR-Licht daran zurückreflektiert.
Eine dunkle **Fotodiode (FD)** reagiert auf dieses Licht und veranlasst, dass der Sensor ein Spannungssignal (0 - 3 V) an den Micro:bit sendet, der es in einen **Wert von 0 - 1023** umwandelt. Die Stärke des Signals ist abhängig von der Entfernung, Form und Oberfläche des Gegenstandes. Die **Lichtempfindlichkeit** des IR-Sensors kann am **Potentiometer** mit einem kleinen Schraubendreher eingestellt werden.
Beachte den richtigen **Anschluss** des Sensors:
VCC (+), **GND (-)** und **OUT** (Signal an **PIN 0**)
Leider reagiert die schwarze **Fotodiode (FD)** durch seitlichen Lichteinfall teilweise auch auf helles Tageslicht. Abhilfe schafft das Aufschieben eines dunklen Schlauchs (zB. Schrumpfschlauch) oder ein Isolierband.



8. Die wichtigsten Blöcke für den Anfang:

- Aus dem **Register [Grundlagen]**:



Alle Blöcke (= Programme) innerhalb der **Start-Klammer** werden nur einmal beim **Start** ausgeführt.

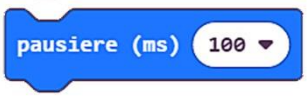
Blöcke innerhalb dieser Klammer werden vom Micro:bit so lange als **Endlosschleife** ausgeführt, bis man den Strom abschaltet.

Dieser Block zeigt die eingefügte **Zahl** (hier „3“) auf der **LED-Matrix** an.

Dieser Block zeigt den eingefügten **Text** (hier „Hallo!“) am Micro:bit als **LED-Laufschrift** an.

Dieser Block zeigt das gewählte **Symbol** (hier „Herz“) als **LED-Symbol**. Mit der **Pfeilauswahl** erscheint eine Auswahl von 40 Symbolen.

Diese Funktion stellt die **Micro:bit LED-Anzeige** mit 25 LEDs dar. Durch Anklicken der dunkelblauen Felder kann man die einzelnen LEDs an- und ausschalten und damit eigene Symbole erstellen.



Durch das Einfügen eines **Pause-Blocks** wird der Programmablauf für eine bestimmte Zeit (hier 100 ms) verzögert.
Die Angabe ist in **Millisekunden** (ms) → **1 Sekunde = 1000 ms**

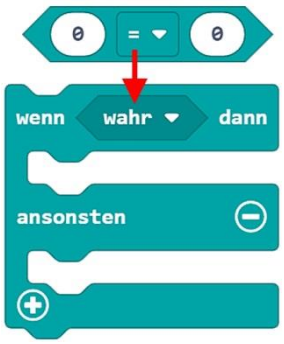
- Aus dem **Register [Eingabe]**:

Bei einem Klick auf **Knopf A** des Micro:bit wird der Programmblock innerhalb der Klammer ausgeführt.
Mit der Pfeilauswahl kann man weitere Knöpfe aktivieren: **B** und **A+B**

- Aus dem **Register [Schleifen]**:

Alle Befehle (Blöcke) innerhalb des **Wiederhol-Blocks** werden in der gewählten Anzahl (hier 4-mal) wiederholt.

- Aus dem **Register [Logik]**:



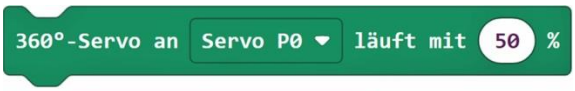
Dieser sechseckige **»Vergleichs-Block« (0 = 0)** vergleicht zwei Werte, zB. ob etwas kleiner, größer, gleich groß usw. ist und gibt das Ergebnis als **„wahr“** oder **„falsch“** zB. weiter an einen »**Wenn-Block«.**

Ein **»Wenn-Block« (wenn/dann)** mit Verzweigung prüft, ob eine eingegebene **Bedingung wahr** ist (z.B. 0 = 0). Ist sie **„wahr“**, wird der obere Programmteil ausgeführt, **ansonsten** der untere.
Am ⊕ kann der **»Wenn-Block«** erweitert, am ⊖ reduziert werden.

- Aus dem **Register [Fortgeschritten] + [Pins]**:

Dieser Block prüft ein **analoges Eingangssignal an P0** und gibt es als **Wert von „0 - 1023“** weiter an das Programm.

- **Register [⊕ Erweiterungen] + Bild Servo anklicken! → Neues Register [Servos]**:



Ein **Rotations-Servo** (360°) an **„P0“** soll sich mit **halber Geschwindigkeit** (50 %) vorwärts drehen.

Dieser Block **stoppt Servos**.

Micro:bit-Programme für den ServoBot + IR-Sensor

- Programme mit dem Zusatz **„nur V2“** laufen nur an einem **Micro:bit V2**.
- Die vorgeschlagenen **Namen** der **„hex-Dateien“** können natürlich geändert werden.

Programm 1: Begrüßung

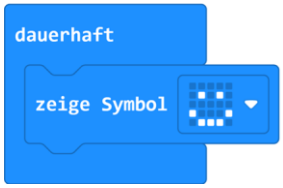
Öffne den **Makecode-Editor** (https://makecode.microbit.org/), klicke auf [Neues Projekt] und gib ihm den Namen „Hallo1“.

Vorgabe: Nach dem Einschalten („beim Start“) soll der Micro:bit einmal den Lauftext **„Hallo!“** anzeigen und dann **dauerhaft** einen **»freundlichen Smiley«**.

Programm-Code: (microbit-Hallo1.hex)

Im linken **Simulationsbereich** des **Makecode-Editors** sieht man bereits eine Vorschau, was das Programm tut.
Speichere, wie auf Seite 3 beschrieben, das fertige Programm auf dem Computer.
Verbinde den Micro:bit über ein Micro-USB-Kabel mit dem Computer und übertrage das Programm (microbit-Hallo1.hex) auf den Micro:bit.

Der Lauftext **„Hallo“** wird nur einmal angezeigt.



Solange der Micro:bit Strom hat, leuchtet der Smiley.

Weitere Aufgabe: Ändere den Text auf: **„Ich bin ein Microbit“** und das Symbol auf ein **„Herz“.**

Programm 2: Herzklopfen

Vorgabe: Ein großes und ein kleines **»Herz-Symbol«** sollen jeweils **200 ms** lang abwechselnd aufleuchten.

Programm-Code: (microbit-Herzklopfen1.hex)

Weitere Aufgabe: Ändere die **Herzfrequenz** durch längere Pause-Zeiten (zB. 500 ms).

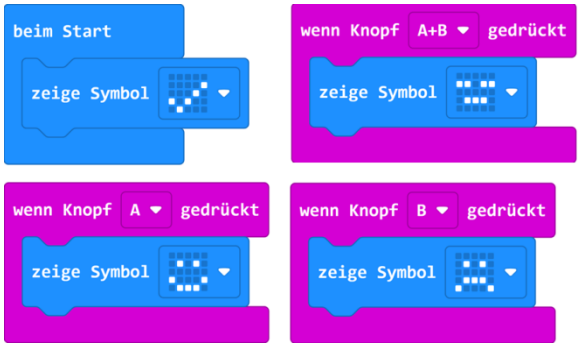
Der **Pause-Block** lässt das „Herz“ 200 ms lang aufleuchten.

Programm 3: Knopf A und B

Vorgabe: Nach Drücken der **Knöpfe A, B und A+B** sollen verschiedene **»Smileys«** aufleuchten.

Programm-Code: (microbit-KnopfAB-1.hex)

Weitere Aufgabe: Nach Drücken der **Knöpfe A, B** und **A+B** sollen die **Buchstaben „A“**, **„B“** und **„C“** angezeigt werden.

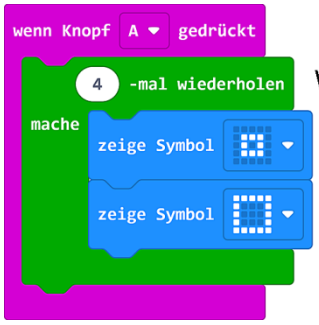


Programm 4: Wiederholung

Durch den Einsatz einer **»Wiederhol-Schleife«** aus dem **Register [Schleifen]** kann man die Anzahl der Wiederholungen von eingefügten Programmteilen genau festlegen.

Vorgabe: Ein kleines und ein großes **»Quadrat-Symbol«** sollen nach Betätigung von **Knopf A** **4-mal blinken**.

Programm-Code: (microbit-Wiederholung1.hex)



Wiederhol-Schleife

Weitere Aufgaben:

- Ändere die **Anzahl der Wiederholungen**.
- Setze an Stelle der Symbole **2 Zahlen** (zB. 0 / 1).
- Ändere die Blinkfrequenz durch **»Pause-Blöcke«**.

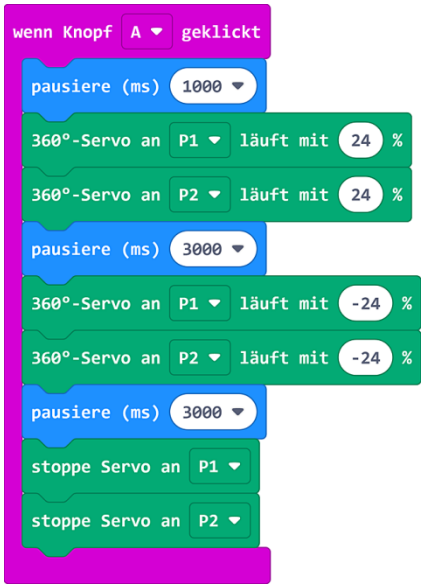
Programm 5: Der ServoBot dreht sich am Stand

Erstelle ein neues **Register [Servos]** durch Klicken auf das **Register [⊕ Erweiterungen] + Bild Servo**.
Durch **negative Prozentzahlen** (zB. -24) kann die **Drehrichtung** der Servos geändert werden.
Da Rotations-Servos relativ viel Strom brauchen, sollten sie nicht mit Geschwindigkeiten über 60% betrieben werden!

Vorgabe: Setze bei Betätigung von **Knopf A** die Geschwindigkeit der **Servos an „P1“** und **„P2“** für **3 Sekunden** auf **„24%“**, dann für **3 Sekunden** auf **„-24%“** und stoppe sie schließlich.

Programm-Code: (microbit-SB-Drehen1.hex)

Bei gleichen Werten an beiden Servos fährt der ServoBot nicht geradeaus, sondern dreht sich um die eigene Achse. Die Ursache liegt in der um 180° versetzten Befestigung der Servos am ServoBot.



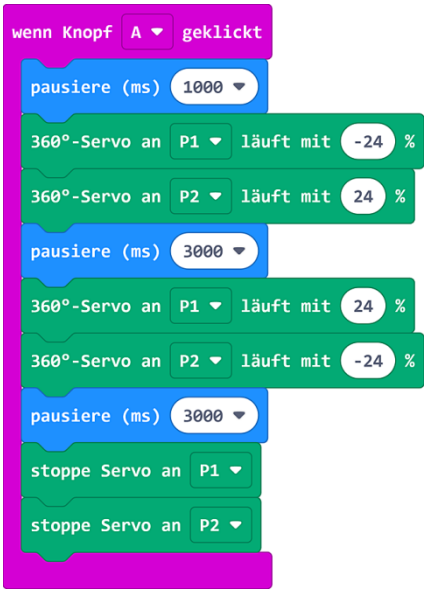
Programm 6: Der ServoBot fährt vor und zurück

Wie im Programm 5 gesehen, muss also **einer der beiden Servos** für Fahrten **geradeaus** auf einen **negativen Wert** gesetzt werden.
Tipp: Fährt der ServoBot trotz gleicher Prozentwerte nicht geradeaus, kann zum Ausgleich der Prozentwert eines Servos leicht erhöht bzw. erniedrigt werden.

Vorgabe: Setze bei Betätigung von **Knopf A** die Geschwindigkeit der **Servos** für **3 Sekunden** an **„P1“** auf **„-24%“** und **„P2“** auf **„24%“,** danach für **3 Sekunden P1“** auf **„24%“** und **„P2“** auf **„-24%“** und stoppe sie schließlich.

Programm-Code: (microbit-SB-gerade1.hex)

Weitere Aufgaben: Ändere die **Pause-Zeiten** und setze die Geschwindigkeit auf **„60%“** bzw. **„- 60%“**.

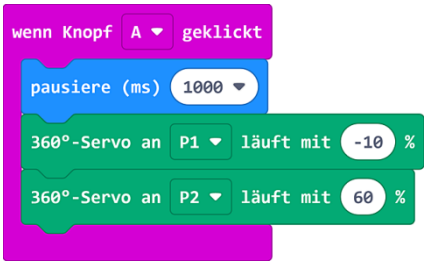


Programm 7: Der ServoBot fährt im Kreis

Für **Kreisfahrten** müssen an **„P1“** und **„P2“** verschiedene Prozentwerte für die **Geschwindigkeit** eingegeben werden.

Vorgabe: Setze bei Betätigung von **Knopf A** den Servo an **„P1“** auf **„-10%“** und an **„P2“** auf **„60%“.**

Programm-Code: (microbit-SB-Kreis1.hex)



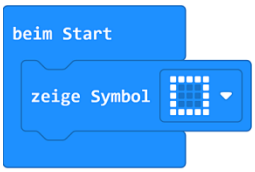
Weitere Aufgabe: Teste **verschiedene Prozentwerte** und beobachte dabei die verschiedenen Kurvenradien.

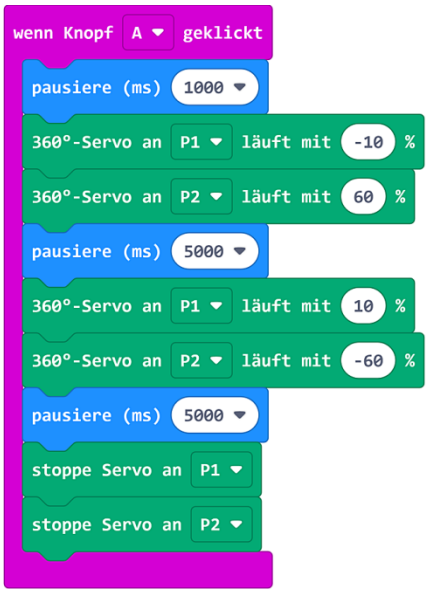
Programm 8: Der ServoBot fährt Kreise und Kurven

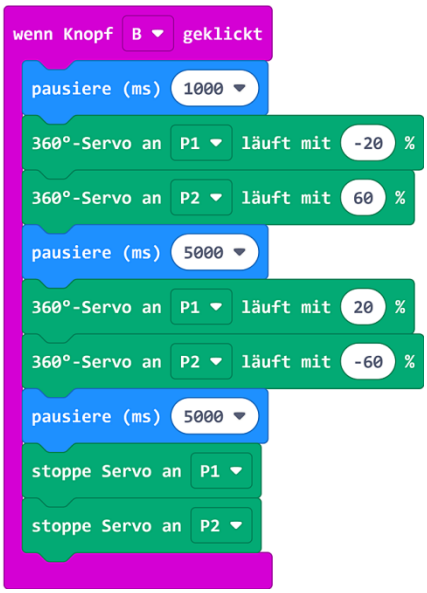
Je weiter die **Prozentwerte** an **„P1“ und „P2“** auseinander liegen, desto kleiner sind die Kreise, bzw. Kurvenradien.

Vorgabe: Setze beim **Start** ein **»Quadrat-Symbol«**. Bei Betätigung von **Knopf A** soll der ServoBot mit einem **kleinen Kreisradius** vor und zurück fahren. Bei einem Klick auf **Knopf B** soll sich der ServoBot mit einem **größeren Kurvenradius** vor und zurück bewegen.

Programm-Code: (microbit-Kreise1.hex)







Weitere Aufgabe:

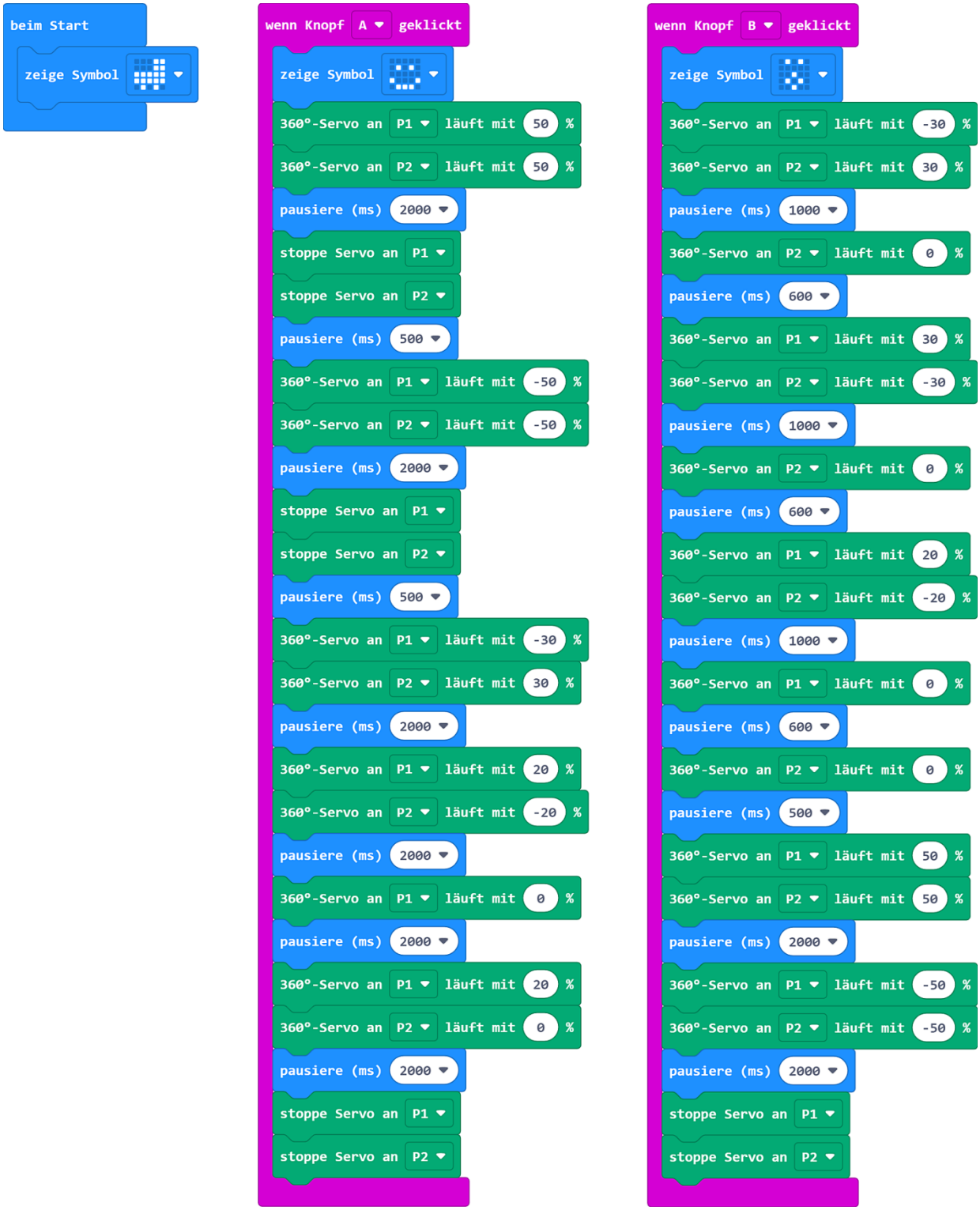
Erweitere das Programm so, dass der ServoBot bei gleichzeitigem Drücken der **Knöpfe A + B** mit einem **Kreisradius** von ca. **0,5 m** vorwärts fährt.

Programm 9: Der ServoBot tanzt

Bei diesem Programm soll der ServoBot verschiedene kurze Bewegungen ausführen.
Achtung: Um Motor und Getriebe zu schonen, sollten die Servos vor größeren Fahrtrichtungsänderungen kurz gestoppt werden!

Vorgabe: Setze beim Start ein **»Fahrzeug-Symbol«**. Nach Drücken von **Knopf A oder B** soll der ServoBot verschiedene **Dreh-, Vorwärts- und Rückwärtsbewegungen** durchführen.

Programm-Code: (microbit-SB-Tanz1.hex)



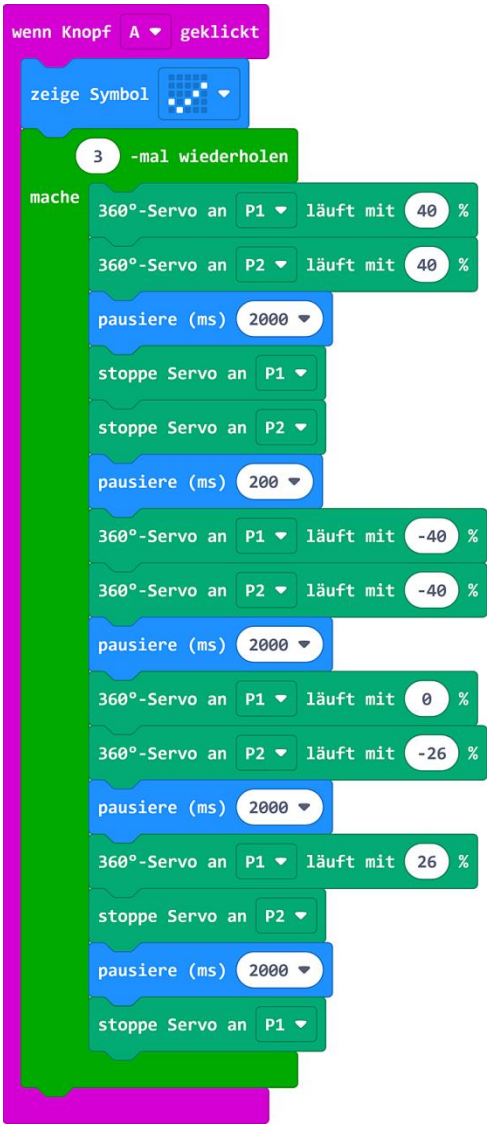
Weitere Aufgabe: Verfasse selbst ein „**Tanzprogramm**“ mit kurzen Vorwärts-, Rückwärts- und Drehbewegungen des ServoBot.

Programm 10: Wiederholen von Bewegungen

Durch den Einsatz einer **»Wiederhol-Schleife«** aus dem **Register [Schleifen]** kann man die Anzahl der Wiederholungen von eingefügten Programmteilen genau festlegen.

Vorgabe: Bei Betätigung von **Knopf A** soll zuerst ein **»Häkchen-Symbol«** erscheinen.
In einer **»Wiederhol-Schleife«** sollen verschiedene Bewegungen des ServoBot **3-mal** wiederholt werden.

Programm-Code: (microbit-SB-Tanz-WH1.hex)



Weitere Aufgaben:

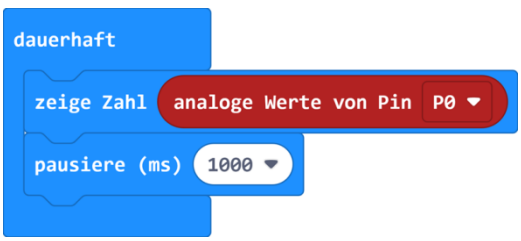
- Ändere die **Anzahl** der **Wiederholungen**.
- Ändere die Bewegungsabläufe durch eigene Ideen.

Programm 11: Analoge Werte des IR-Sensors

Infrarot-Sensoren erkennen durch Reflexion von unsichtbarem, infraroten Licht, ob sich ein Gegenstand in der Nähe befindet und geben entsprechende elektrische Signale an den Micro:bit weiter. Die Stärke der Signale ist abhängig von der Entfernung, Form und Oberfläche des Gegenstandes. Die Signale werden an den **Micro:bit-Eingängen** in **analoge Werte** von **0** - **1023** umgewandelt.
Die **Empfindlichkeit** des IR-Sensors kann am **Potentiometer** des Sensors mit einem kleinen Schraubendreher eingestellt werden.
Für das Programm benötigt man den **Block »analoge Werte von Pin …«** aus den **Registern [Fortgeschritten] + [Pins]**.

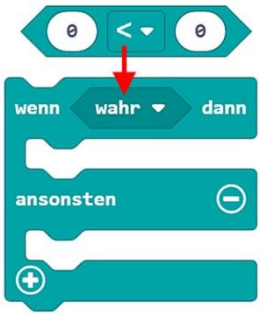
Vorgabe: Der Micro:bit soll den **analogen Wert** des **IR-Sensors** an **„P0“** dauerhaft anzeigen.

Programm-Code: (microbit-SB-IR-Wert1.hex)



Ein Tipp: Leider reagiert die **dunkle Fotodiode** des IR-Sensors durch seitlichen Lichteinfall teilweise auch auf helles Tageslicht. Abhilfe schafft das Aufschieben eines dunklen Schlauchs (zB. Schrumpfschlauch) oder ein Isolierband.

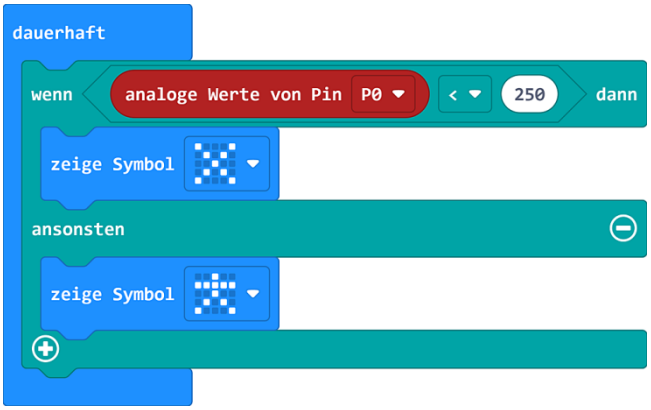
Programm 12: Annäherung



Für das Programm benötigen wir einen **»Wenn-Block« (wenn/dann)** mit Verzweigung und einen sechseckigen **»Vergleichs-Block« (0 < 0)** aus dem **Register [Logik]**.
Wenn die **Bedingung wahr** ist (**analoger Wert von P1 < 250**), dann wird der obere Programmteil ausgeführt, **ansonsten** der untere.
Am ⊕ kann der **»Wenn-Block«** erweitert, am ⊖ reduziert werden.

Der **Block »analoge Werte von P1«** befindet sich in den **Registern [Fortgeschritten] + [Pins]**.

Vorgabe: Wenn sich jemand dem **IR-Sensor** an **„P0“** nähert, soll ein **»X-Symbol«** aufleuchten - ansonsten ein **»Figur-Symbol«**.



Programm-Code:

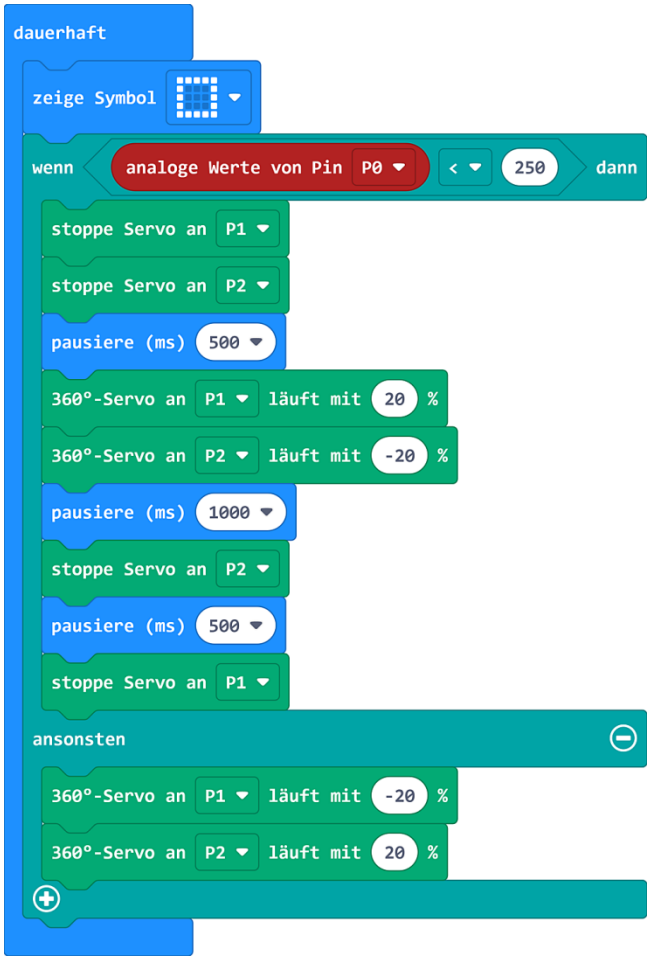
(microbit-SB-Näherung1.hex)

Weitere Aufgabe: Anstelle des **»X-Symbols«** soll der **Text „Stopp!“** angezeigt werden.

Programm 13: ServoBot mit IR-Abstandssensor

Wenn der IR-Sensor ein Hindernis erkennt, soll der ServoBot mit Ausweichbewegungen reagieren und dann seine Fahrt fortsetzen.

Vorgabe: In einer **»Dauerhaft-Schleife«** soll am Micro:bit ein **»Quadrat-Symbol«** aufleuchten.
Wenn der **analoge Wert** des **IR-Sensors unter „250“** liegt, soll der ServoBot kurz stoppen, retour fahren und die Richtung ändern.
Ansonsten soll der ServoBot mit **„20%“ geradeaus** fahren.



Programm-Code: (microbit-SB+IR1.hex)

Ein Tipp: Da der IR-Sensor zum Teil auch auf helles Sonnenlicht reagiert, sollte der ServoBot mit diesem Programm nur in Innenräumen betrieben werden!

Weitere Aufgabe:

Programmiere eigene Ausweichbewegungen, wenn der ServoBot auf ein Hindernis trifft.

Programm 14: Fernsteuerung des ServoBot (nur V2)

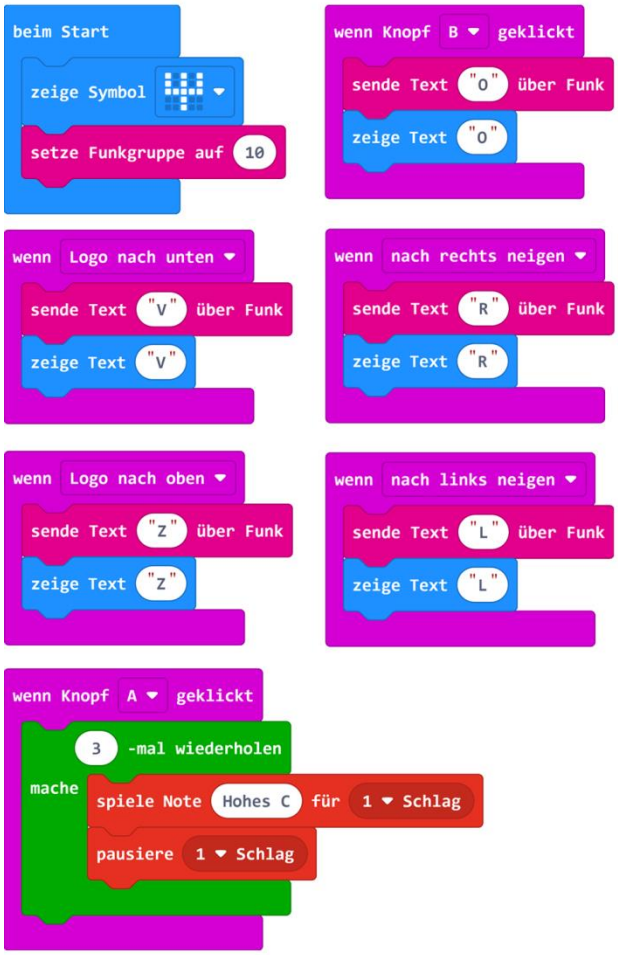
Wenn **zwei Micro:bits** auf einen gleichen **Funkkanal** (1 - 250) gesetzt werden, können sie miteinander in Verbindung treten. Senden kann man **Zahlen, Werte** und kurze **Texte**.
Die erforderlichen **Blöcke** befinden sich im **Register [Funk]**.

Vorgabe - Sender: Nach dem Einstellen eines **Funkkanals** (hier 10) soll durch **4 Neigungen** des **Sende-Micro:bit** jeweils ein **Buchstabe (V, Z, L, R)** gesendet werden. Bei Druck auf **Knopf A** soll ein **Ton** erklingen und bei Klicken von **Knopf B** der Buchstabe **„O“** gesendet werden.

Vorgabe - Empfänger: Nach Einstellung des **Funkkanals** (hier 10) sollen im **Empfangs-Block »wenn Text empfangen …«** in einer verzweigten **»Wenn-Schleife«** die empfangenen Buchstaben (= String) in verschiedene Bewegungen des ServoBot umgewandelt werden. **Knopf A** soll alle Servos stoppen.

Programm für Sende-Micro:bit:

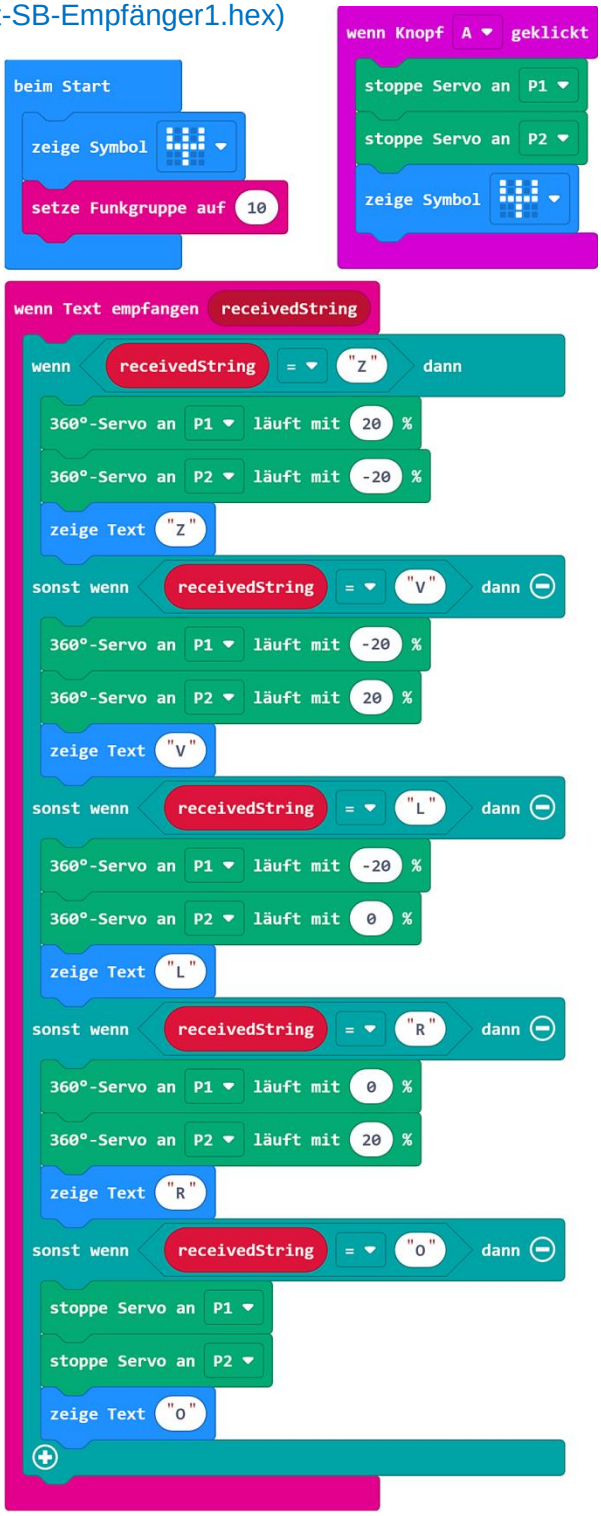
(microbit-SB-Sender1.hex)



Ein Tipp: Wenn man beim Sende-Micro:bit die Funktion von **Knopf A** (Ton) weglässt, dann funktioniert das Programm auch auf einem **Micro:bit V1**.

Programm für Empfänger-Micro:bit:

(microbit-SB-Empfänger1.hex)

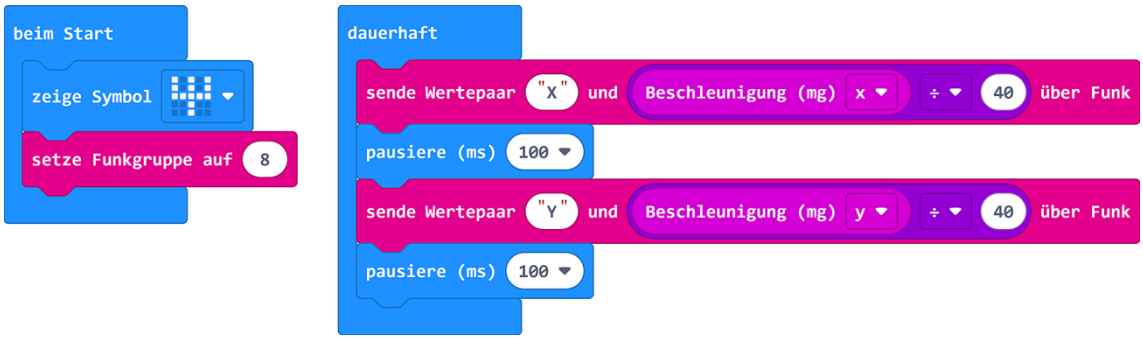


Programm 15: Stufenlose Fernsteuerung des ServoBot mittels Neigungssensor

Für etwas fortgeschrittene Programmierer nun eine stufenlose Fernsteuerung des ServoBot. Der **Block »Beschleunigung (mg) …«** aus dem **Register [Eingabe]** erzeugt beim **Neigen** des Micro:bit an der **x- und y-Achse** Werte zwischen **-1023 und +1023**. Um servogerechte Geschwindigkeitswerte zwischen -25 und +25 zu erhalten, müssen die **Neigungswerte** mit einem **Rechenblock »0:0«** aus dem **Register [Mathematik]** durch **40** dividiert werden.
Um **Wertepaare** senden zu können, benötigt man den **Block »Sende Wertepaar „Name“ und „0“ über Funk**« aus dem **Register [Funk]**.

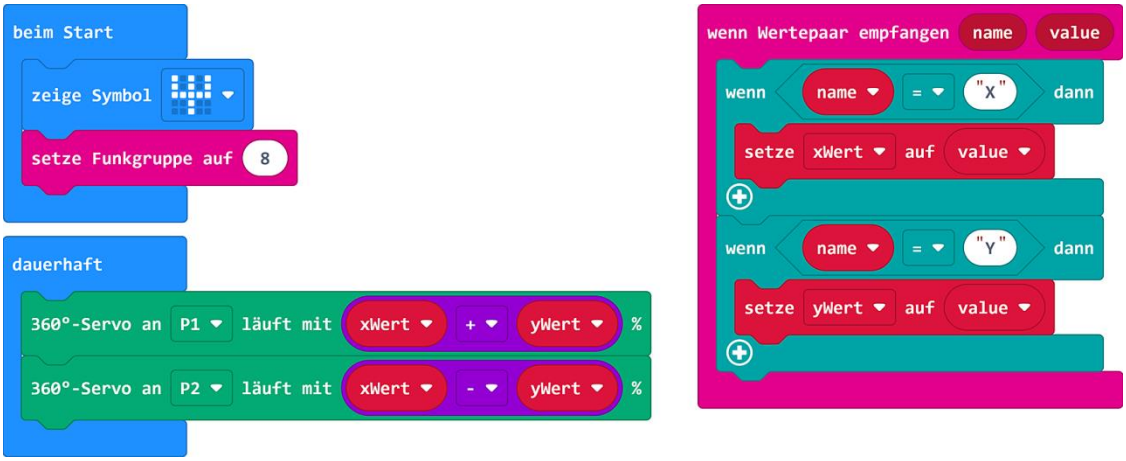
Vorgabe - Sender: Nach Setzen der **Funkgruppe** (hier 8) soll der Micro:bit je nach Neigung alle **100 ms** ein **Wertepaar** senden. Ein **„X“** für Neigen vor und zurück und ein **„Y“** für seitliches Neigen. Die **Neigungswerte** (= Beschleunigung) sollen **durch 40 dividiert** werden.

Programm für Sende-Micro:bit: (microbit-SB-Sender2.hex)



Vorgabe - Empfänger: Erstelle im **Register [Variablen]** zwei Variablen mit den Namen **„xWert“** und **„yWert“**. Setze beim **Start** ein **»Antennen-Symbol«** und die **Funkgruppe** (hier 8).
Die **Geschwindigkeit** der zwei Servos in der **»Dauerhaft-Schleife«** soll mit **»Rechenblöcken«** auf eine **Addition** und **Subtraktion** der Variablen **„xWert“** und **„yWert“** gesetzt werden.
Im **Empfangs-Block »wenn Wertepaar empfangen …«** aus dem **Register [Funk]** sollen in zwei **»Wenn-Schleifen«** den **Variablen** die empfangenen **Werte** (= value) zugeordnet werden.

Programm für Empfänger-Micro:bit: (microbit-SB-Empfänger2.hex)



Nachwort:

Mit dieser Programmieranleitung haben wir versucht, die wichtigsten Grundfunktionen des Micro:bit mit einfachen bis fortgeschrittenen Programmideen an einem selbst gebauten Roboter mit zwei 360°- Servos und Infrarot-Sensor zu veranschaulichen und zu festigen.
Die gezeigten Programme sollen als Basis für weiterführende eigene Programmideen dienen.