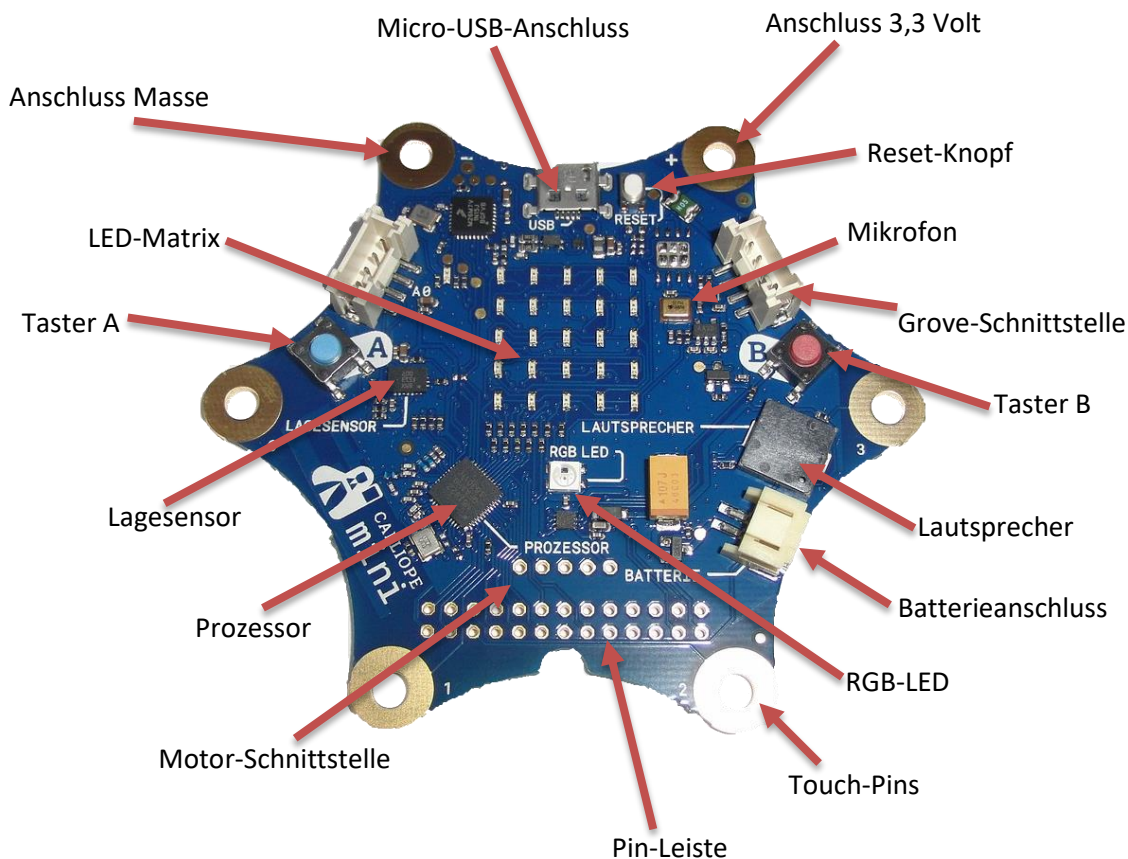


Programmieren mit dem Calliope-mini

Der Calliope-mini ist ein Einplatinencomputer mit „onboard-Elementen“, der ursprünglich für den Einsatz an Grundschulen entwickelt wurde. Neben ei

Hardware:



Sensoren und Aktoren

Die onboard-Elemente können dem EVA-Prinzip zugeordnet werden. Sensoren sind Eingabe-Elemente, Aktoren für die Ausgabe zuständig. Die Verarbeitung erfolgt mit Hilfe des Prozessors.

Sensoren: Taster, Reset-Knopf, Touch-Pins, Lagesensor (dieser wird auch als Bewegungssensor bzw. Kompass verwendet), Mikrofon, Temperatursensor (steckt im SoC – System-on-a-Chip, also dem Chip, den sich Controller, Speicher, Interfaces usw. teilen. Daher ist die Temperaturangabe oft zu hoch), Lichtsensor (der Lichtsensor ist kein eigenständiger Sensor, vielmehr werden die LEDs kurzerhand umfunktioniert. Jede LED arbeitet prinzipiell auch als Fotodiode; wenn Licht auf sie fällt, befreien die Photonen Elektronen in der Sperrschicht. Der Sperrstrom erhöht sich, was man wiederum mit dem Analog-Digital-Konverter messen und digitalisieren kann), weitere zusätzlich evtl. über die Grove- bzw. Pin-Schnittstelle.

Aktoren: LED-Display, RGB-LED, Lautsprecher, Motor-Schnittstelle, weitere zusätzlich evtl. über die Grove- bzw. Pin-Schnittstelle.

Software:

Der Calliope kann mit Hilfe einer blockbasierten Programmiersprache angesteuert werden. Aktuell werden hauptsächlich drei unterschiedliche Varianten verwendet:

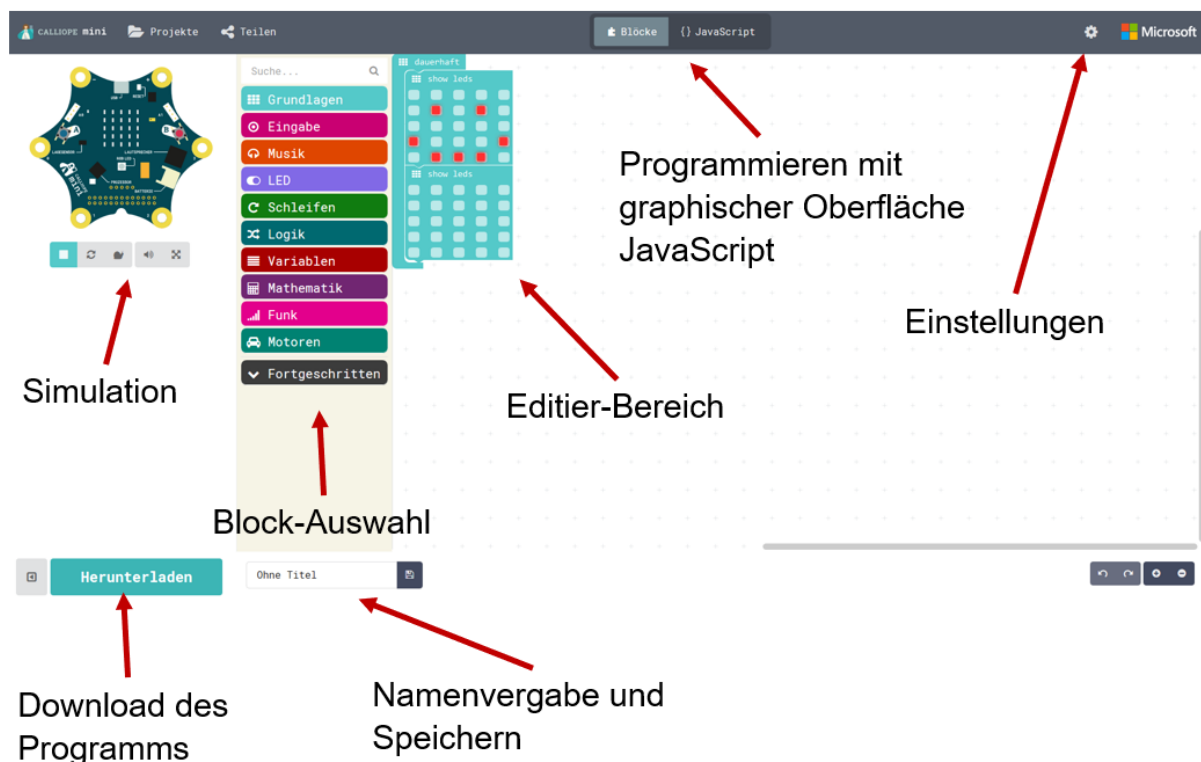
Calliope-mini Editor <http://miniedit.calliope.cc/>

Open Roberta Lab <https://lab.open-roberta.org/>

MakeCode <https://makecode.calliope.cc/> (ehemals pxt)

In diesem Handout wird mit dem MakeCode-Editor gearbeitet.

Der MakeCode-Editor

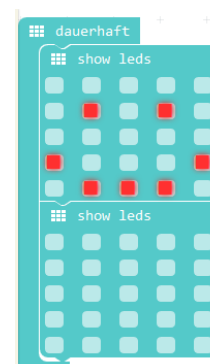


Kennenlernen des Calliope

Nach erstmaligem Start des Editors erscheinen folgende Blöcke:

Diese bewirken, dass der Smiley blinkt.

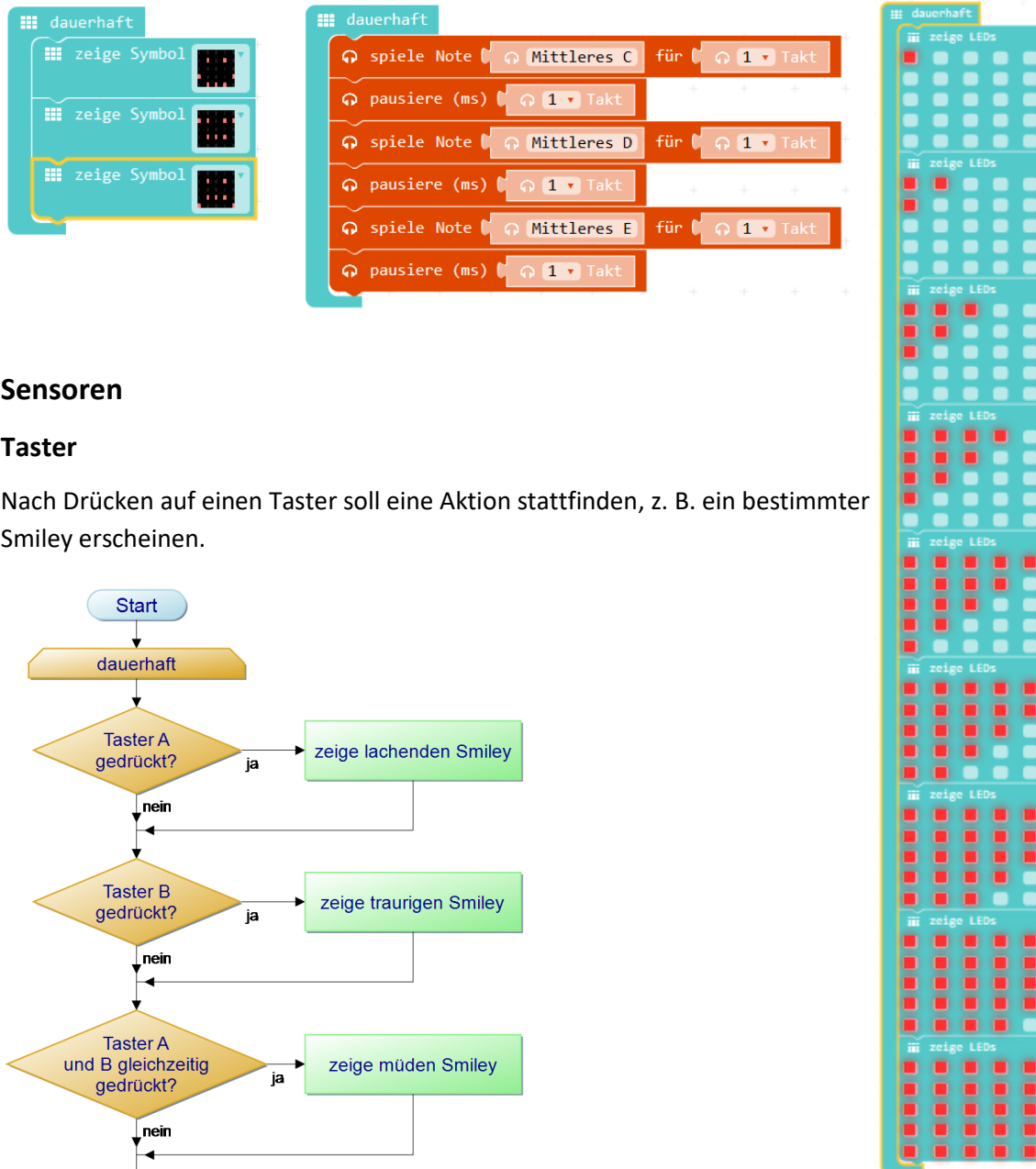
Ähnlich wie bei der Arduino IDE gibt es einen Block für Programmteile, die zu Beginn ausgeführt werden sollen und einen, der für ständige Wiederholung der Programmteile sorgt.



Mögliche erste Aufgabenstellungen

- Zeichne ein Bild
- Spiele mehrere Bilder nacheinander ab, so dass eine Animation entsteht
- Stelle die Animation nach, die in einem gegebenen Video gezeigt wird
- Erstelle ein Programm gemäß des gegebenen Pseudocodes

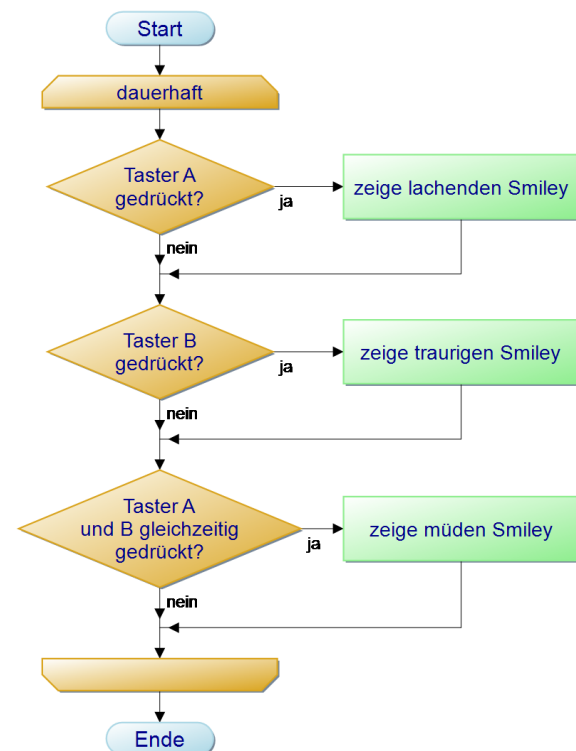
Programmierbeispiele:



Sensoren

Taster

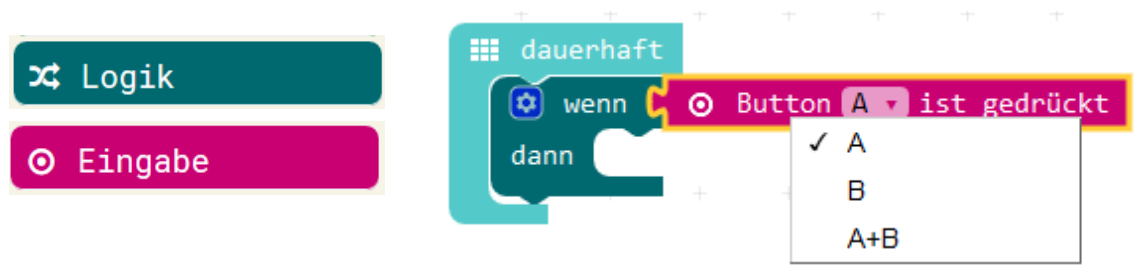
Nach Drücken auf einen Taster soll eine Aktion stattfinden, z. B. ein bestimmter Smiley erscheinen.



Taster findet man im Blockbereich „Eingabe“, man kann dabei auswählen, ob Taster A, Taster B oder Taster A und B gleichzeitig gemeint sind.



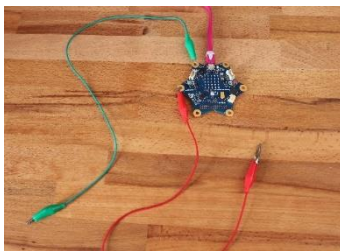
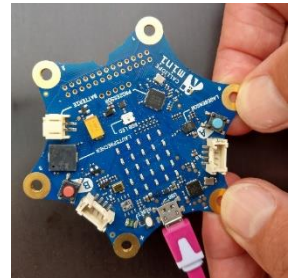
Desweiteren können Taster auch mithilfe einer bedingten Auswahl abgefragt werden. Eine solche findet man im Blockbereich „Logik“.



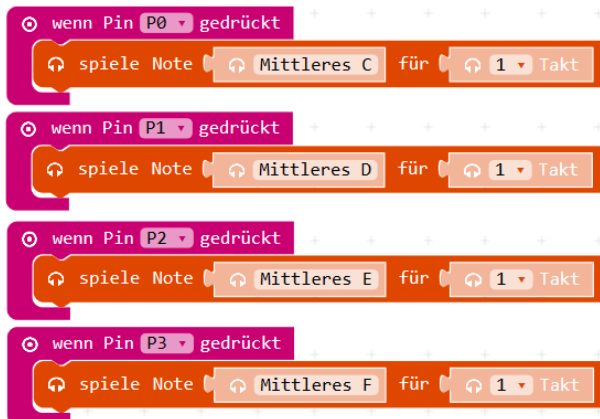
Touch-Pins

Touch-Pins werden ähnlich wie Taster programmiert. Wichtig ist dabei nur, dass nicht nur der Pin, sondern immer gleichzeitig auch der Groundpin (Masse) berührt wird, um einen Stromkreislauf gewährleisten zu können.

Anstatt Berührung mit den Händen können die Pins auch mit leitfähigem Material, wie z. B. Krokodilklemmen bzw. leitfähiger Knete angesprochen werden.



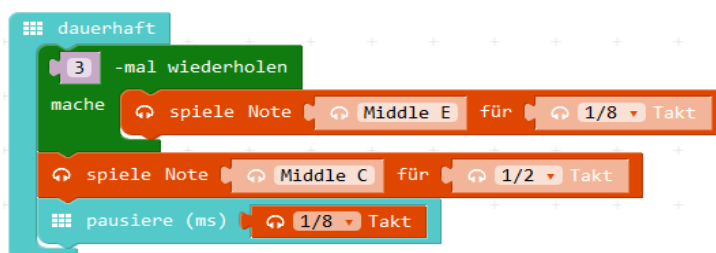
Programmierbeispiel „Klavier“:



Algorithmik - Wiederholungen

Diese sind im Blockbereich „Schleifen“ zu finden.

Programmierbeispiel:



Algorithmik - Variablen

Variablen können im Blockbereich „Variablen“ angelegt werden. Nach Deklaration einer Variablen, erscheint diese in diesem Bereich und kann mit Hilfe der rechten Maustaste auch umbenannt werden.

Variablen

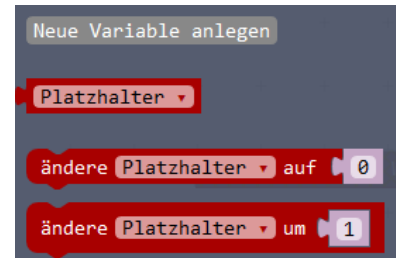
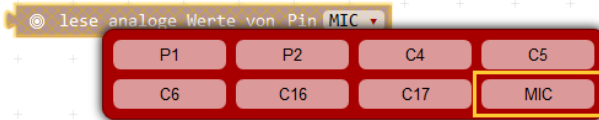
Weitere Sensoren

Programmblöcke für die übrigen Sensoren sind ebenfalls im Bereich „Eingabe“ zu finden.

Den Soundsensor (Mikrofon) dagegen ist im Bereich „Fortgeschritten“-„Pins“ und dort unter „lese analoge Werte von Pin MIC“ enthalten.

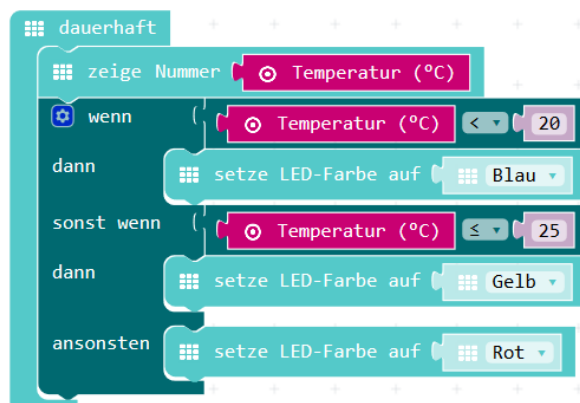
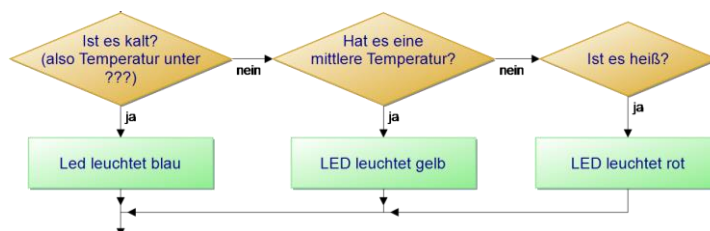
Fortgeschritten

Pins

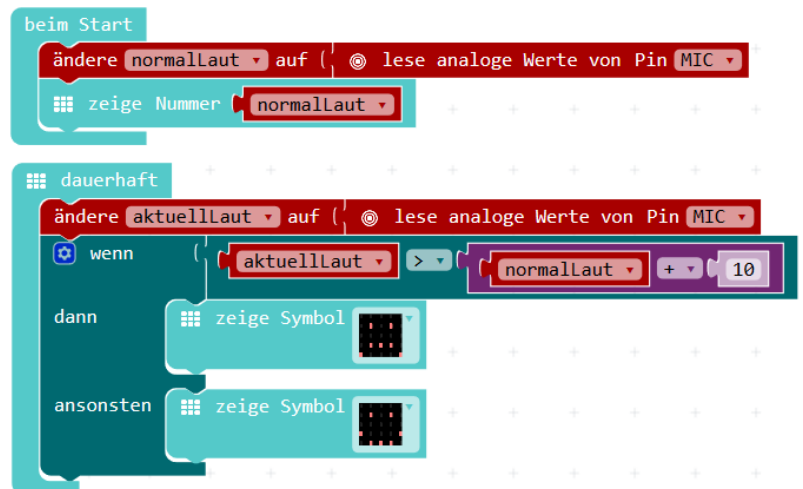
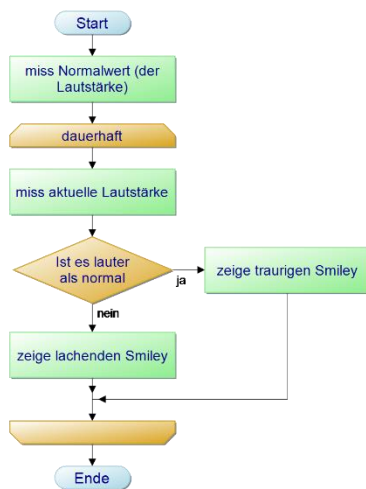


Mögliche Programmierbeispiele sind dazu z. B.

- die RGB-LED entsprechend der Temperatur blau, gelb oder rot (kalt, mittel, warm) leuchten zu lassen



- einen fröhlichen Smiley anzeigen, falls es im Raum nicht zu laut ist, ansonsten einen traurigen Smiley



Stein-Schere-Papier

Das Spiel Stein-Schere-Papier, auch also Schnick-Schnack-Schnuck bekannt, soll mit dem Calliope programmiert werden. In einem ersten Schritt wird dabei zunächst ein Symbol für Stein, Schere oder Papier nach Schütteln des Geräts zufällig angezeigt. In folgenden Schritten soll das Ergebnis über Funk an einen weiteren Calliope übertragen werden und der Calliope nach Knopfdruck auswerten, ob er gewonnen oder verloren hat.

Schritt 1:

Nach Schütteln des Calliope soll zufällig ein Symbol für Stein, Schere oder Papier erscheinen.

Schritt 2:

Durch Druck auf Taster B wird das „Geschüttelte“ an einen anderen Calliope übertragen.

Schritt 3:

Nach Druck auf die Taster A+B erfolgt eine Auswertung. Als Ergebnis wird entweder ein lachender, ein trauriger oder ein neutral schauender Smiley angezeigt. Nach Druck auf Taster A soll wieder das geschüttelte Symbol angezeigt werden.

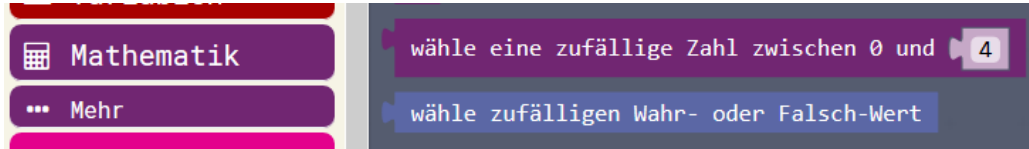
Für die Auswertung gilt:

	Stein	Schere	Papier
Stein	0	+	-
Schere	-	0	+
Papier	+	-	0

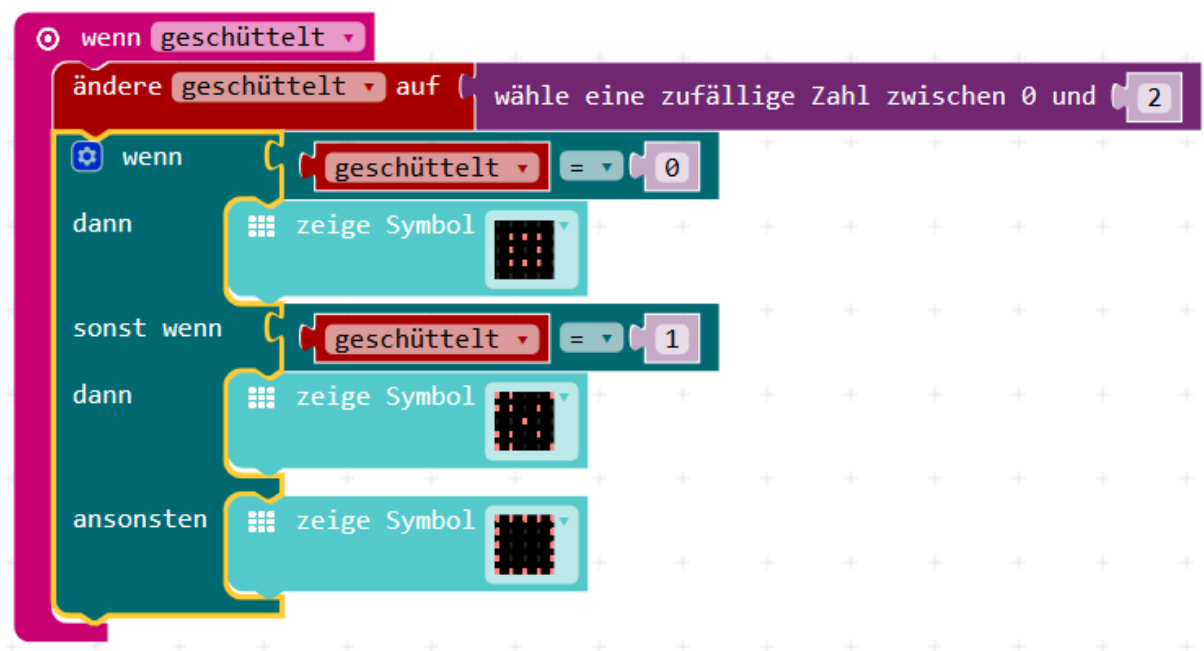
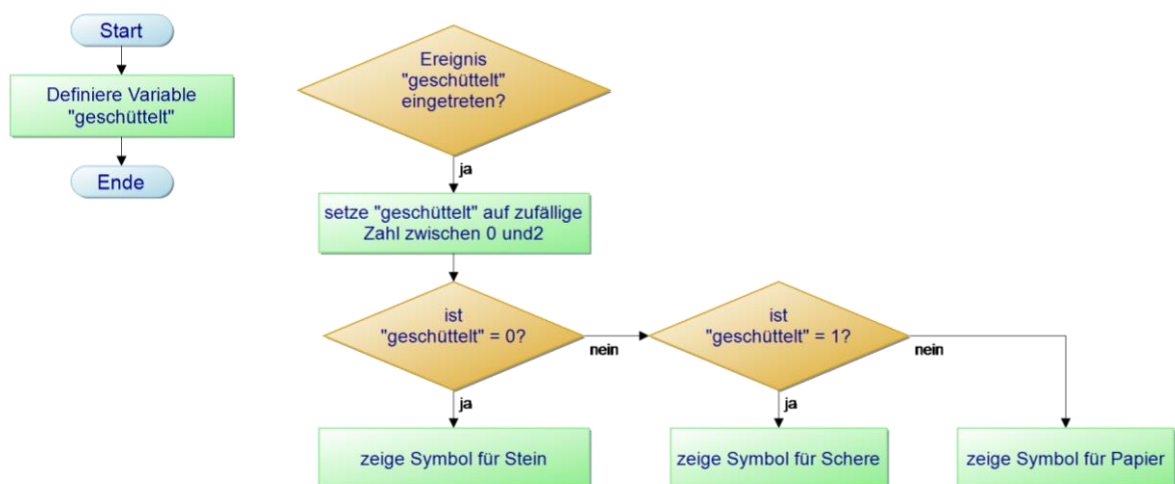
Schritt 1

Nach Schütteln des Calliope soll zufällig ein Symbol für Stein, Schere oder Papier erscheinen.

Um ein Symbol zufällig erscheinen zu lassen, findet man eine entsprechende Funktion im Bereich Mathematik. Hier kann eine Zufallszahl innerhalb eines vorgegebenen Bereichs bestimmt werden.



Der Zufallswert der Zahlen 0, 1 und 2 wird in eine neue erstellte Variable geschüttelt gespeichert. Je nach Zahl wird ein entsprechendes Symbol angezeigt.



Schritt 2

Durch Druck auf Taster B wird das „Geschüttelte“ an einen anderen Calliope übertragen.

Befehle, um Nachrichten übertragen zu können, findet man im Blockbereich „Funk“.

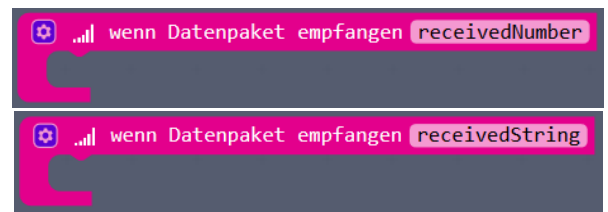
Unter „Mehr“ kann man eine Gruppe von Calliope festlegen, die miteinander kommunizieren können. Wichtig dabei, dass alle zueinander gehörenden Geräte in der gleichen Gruppe sind, andere Geräte jedoch eine andere Gruppennummer (oder keine Gruppe) haben.

Gesendet werden können Zahlen bzw. Zeichenfolgen.

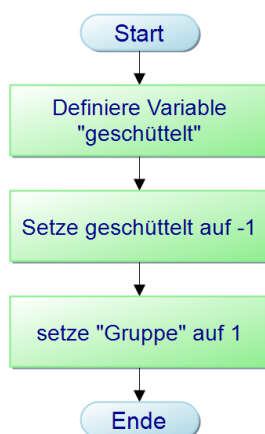


Um übertragene Daten empfangen zu können, werden entsprechende Befehle benötigt. Die übertragenen Daten befinden sich dann automatisch in „receivedNumber“ bzw. „receivedString“, je nach Übertragungsart. Sobald dieser Befehl ausgewertet wurde, ist zusätzlich eine Variable „receivedNumber“

bzw. „receivedString“ im Variablenblock zu finden. Man kann den Wert dann einer anderen Variablen zuweisen. Es handelt sich hierbei jedoch um lokale Variablen, andere Blöcke kennen den Wert der Variable nicht.

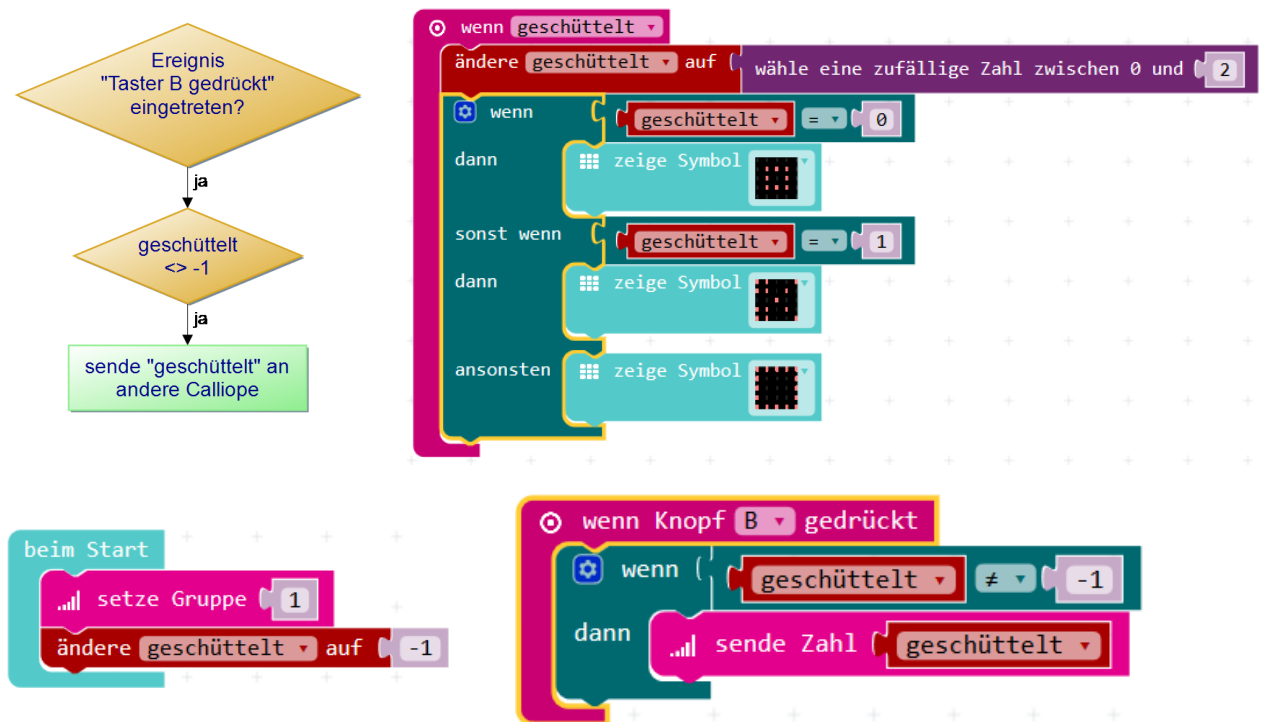


Zu Beginn des Programms sollte eine vorher definierte Variable „geschüttelt“ auf einen nicht im Zufallsbereich enthaltenen Wert, z. B. -1, gesetzt werden. Somit kann abgefragt werden, ob überhaupt geschüttelt wurde. Zusätzlich wird eine Gruppennummer, z. B. 1, bestimmt.



Wurde der Calliope geschüttelt, so erscheint das zugehörige Symbol auf der LED-Matrix. Die Programmierung dazu erfolgt wie unter Schritt 1 beschrieben.

Nach Druck auf Taster B wird zunächst überprüft, ob überhaupt geschüttelt wurde, also ob die zugehörige Variable einen Wert ungleich -1 hat. Falls dies der Fall ist, wird die geschüttelte Zahl auf den anderen Calliope übertragen.



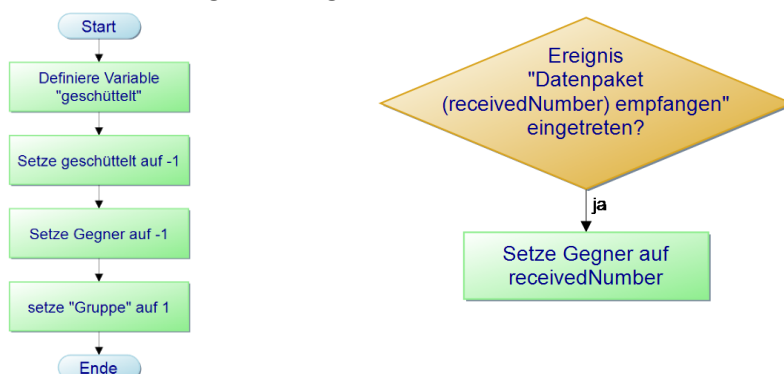
Schritt 3

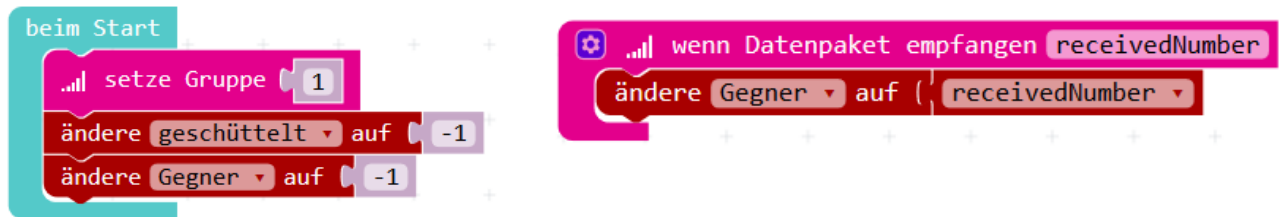
Nach Druck auf die Taster A+B erfolgt eine Auswertung. Als Ergebnis wird entweder ein lachender, ein trauriger oder ein neutral schauender Smiley angezeigt. Nach Druck auf Taster A soll wieder das geschüttelte Symbol angezeigt werden.

Für die Auswertung gilt:

	Stein	Schere	Papier
Stein	0	+	-
Schere	-	0	+
Papier	+	-	0

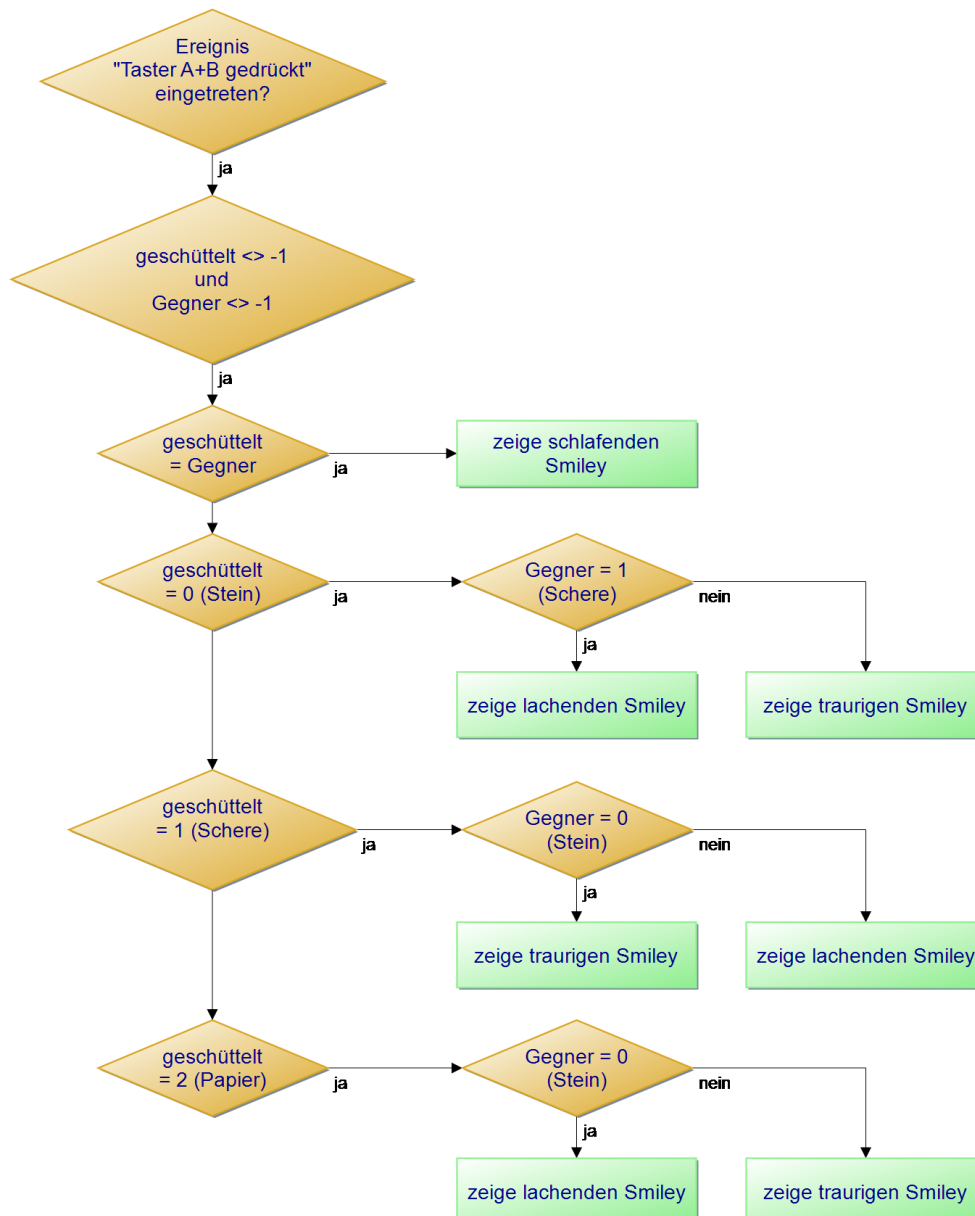
Um eine Auswertung durchführen zu können, wird eine neue Variable, z. B. „Gegner“ für die geschüttelte Zahl des Gegners benötigt. Um absichern zu können, dass diese von einem anderen Calliope übertragen wurde, wird sie zunächst mit -1 initialisiert. Nach der Datenübertragung kann sie dann auf den richtigen Wert gesetzt werden.

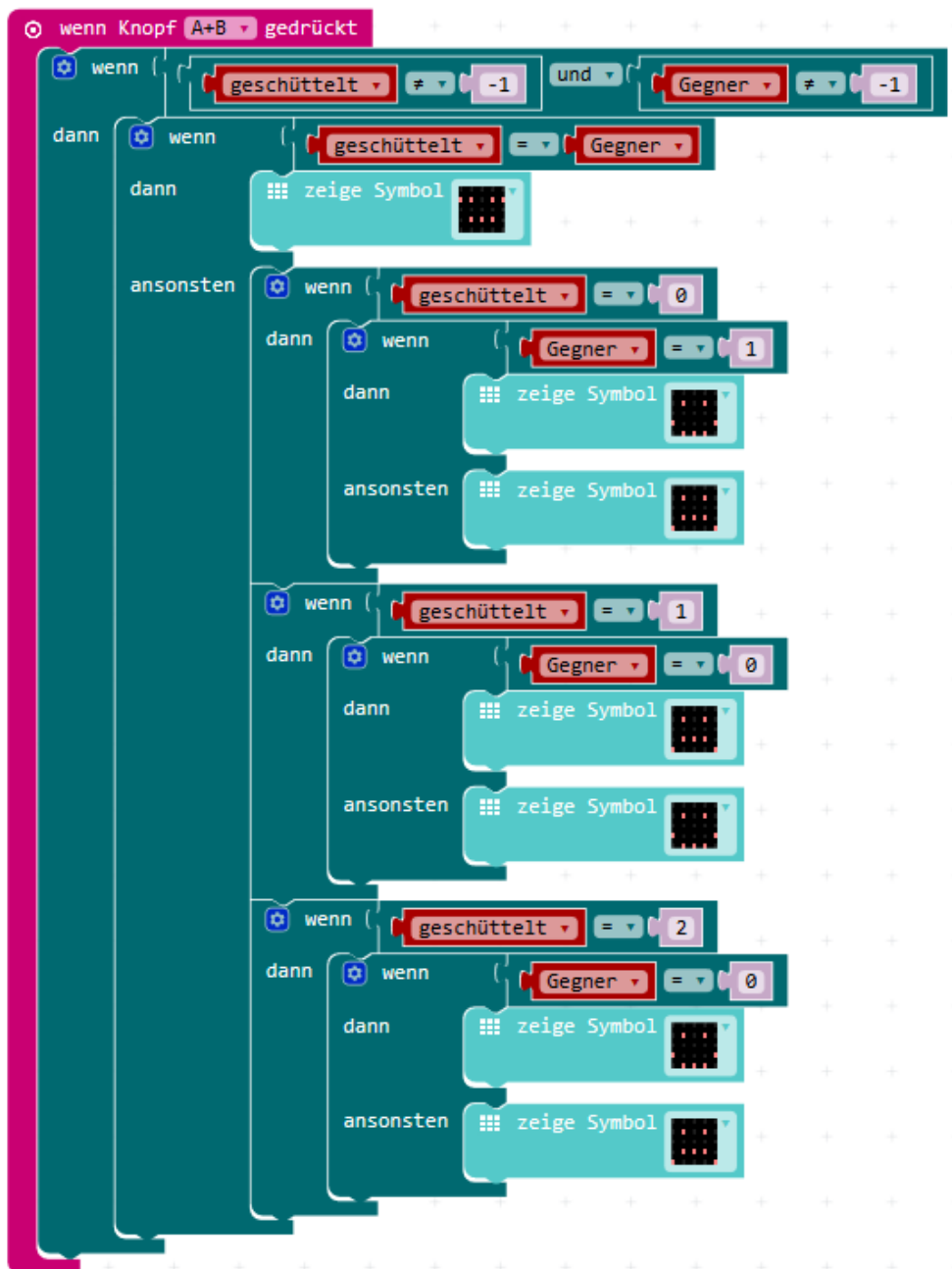




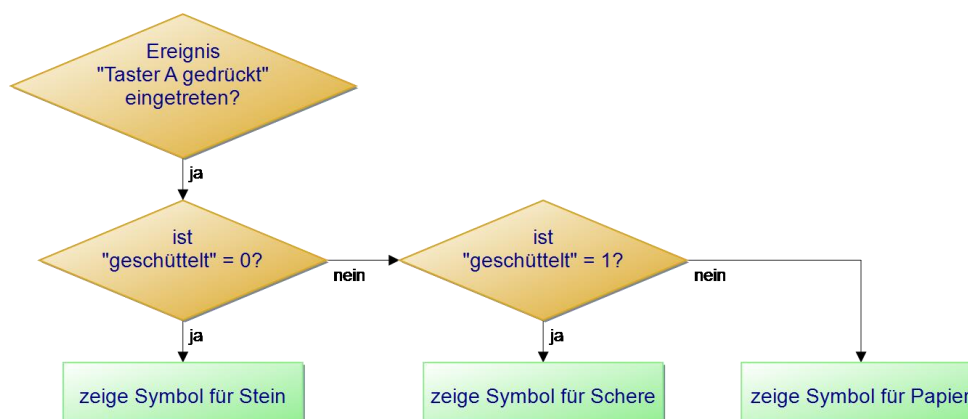
Nach gleichzeitigem Druck auf die Taster A und B soll die Auswertung in Form eines lachenden, traurigen oder neutral schauenden (schlafenden) Smileys erfolgen. Dies geschieht jedoch nur, wenn beide Calliope geschüttelt und der gegnerische Wert übertragen wurde.

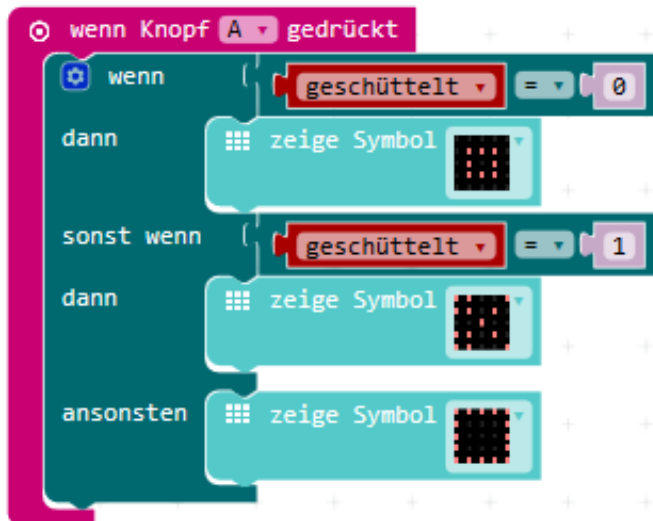
	Stein	Schere	Papier
Stein	0	+	-
Schere	-	0	+
Papier	+	-	0





Nach Druck auf Taster A soll wieder das geschüttelte Symbol angezeigt werden.





Die Anzeige des Symbols erfolgt auf gleiche Art und Weise sowohl nach dem Schütteln, als auch nach Druck auf Taste A. Es empfiehlt sich demnach, eine eigene Funktion „Anzeige“ zu erstellen.

Funktionen findet man im Bereich Fortgeschritten/Funktionen.

Fortgeschritten

Funktionen

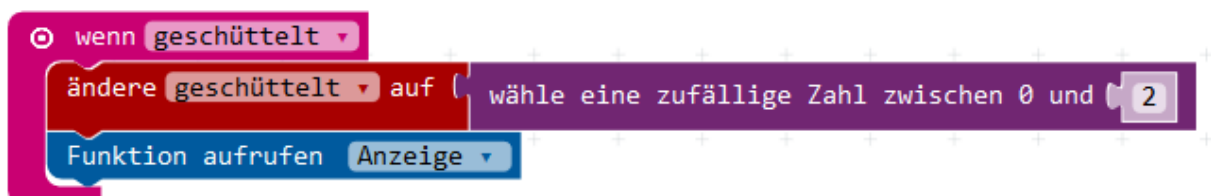
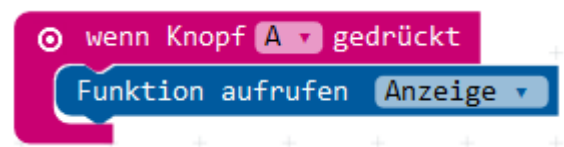
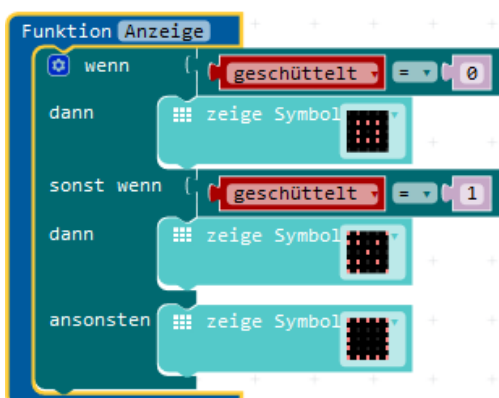
Eine neue Funktion kann erstellt werden.

Erstelle eine Funktion

Nach dem Erstellen erscheint der Funktionsname im Block „Funktion aufrufen“.

Funktion aufrufen Anzeige

Geänderte Blöcke:



Varianten

Neben den „Schüttelergebnissen“ Stein, Schere und Papier kann auch noch ein Brunnen bzw. ein Streichholz dazugenommen werden. Die Auswertung erfolgt dann gemäß der beiden Tabellen:

	Stein	Schere	Papier	Brunnen
Stein	0	+	-	-
Schere	-	0	+	-
Papier	+	-	0	+
Brunnen	+	+	-	0

	Stein	Schere	Papier	Brunnen	Streichholz
Stein	0	+	-	-	+
Schere	-	0	+	-	+
Papier	+	-	0	+	-
Brunnen	+	+	-	0	-
Streichholz	-	-	+	+	0

Weitere Erweiterungsmöglichkeiten:

Der Fantasie sind keine Grenzen gesetzt. Es kann z. B. mitgezählt werden, wie oft gewonnen bzw. verloren wurde und damit eine Gesamtauswertung gemacht werden

oder...