

Best practice – Calliope: Schere-Stein-Papier



(Calliope-) Vorbereitungen

- Calliope im Kästchen mit angeschlossenem Kabel austeilen
- Evt. USB-Verlängerung ausgeben, da Originalkabel nur 20 cm
(SuS drehen und schütteln Calliope)
- Ggf. Batteriefach verwenden
- Hinweise für SuS:
 - SANFT einstecken am PC, Stecker gegebenenfalls um 180° drehen
 - Beim Herausziehen am Stecker anfassen, NICHT am Kabel ziehen
 - Kabel am Calliope selbst NICHT abziehen
 - Reset-Knopf am Calliope zum Neustart des Programms
- Ordnerstruktur am PC bekannt? (Downloadordner, drag-n-drop, ...)
- Anleitung geben: Wie kommt mein Programm auf meinen Calliope?

Das erste Programm

Mein erstes Programm: verwende inf-schule.de

Für Makecode:

www.inf-schule.de/kids/calliope/ein-erstes-programm-makecode

Für NEPO (Open Roberta):

www.inf-schule.de/kids/calliope/ein-erstes-programm-openroberta

- (1) Editor (Programmierumgebung) aufrufen
- (2) programmieren
- (3) Programm übertragen (startet dann von selbst)

Dateien immer ins Laufwerk „MINI“ übertragen (Laufwerk „FLASH“ dient als dauerhafte Ablage für „vorinstallierte“ Programme)

Programmierumgebungen

Makecode

(Microsoft)

(makecode.calliope.cc)

NEPO (Open Roberta)

(Roberta-Initiative des Fraunhofer IAIS)

(lab.open-roberta.org)

Beide: Anleitungen, Beispiele; Export für Calliope im .hex-Format; Calliope-Simulation (mit Debug-Modus); Umschalten zum Quellcode

- Export/Import: .hex-Format, (.mkcd-Format?)
- Quellcode in **JavaScript**
- Erweiterungen einfach „installierbar“ (z.B. thingspeak, neopixel, uvm.)
- Alle Eingaben mit eigener „Schleife“, daher einfacher zu programmieren für SuS (wie in Scratch)

- Export/Import: .xml-Format
- Quellcode in **Python**

Teilweise Unterschiede in den Befehlen (z.B. beim Funk)

→ immer ausprobieren, welche Umgebung für welches Projekt besser geeignet

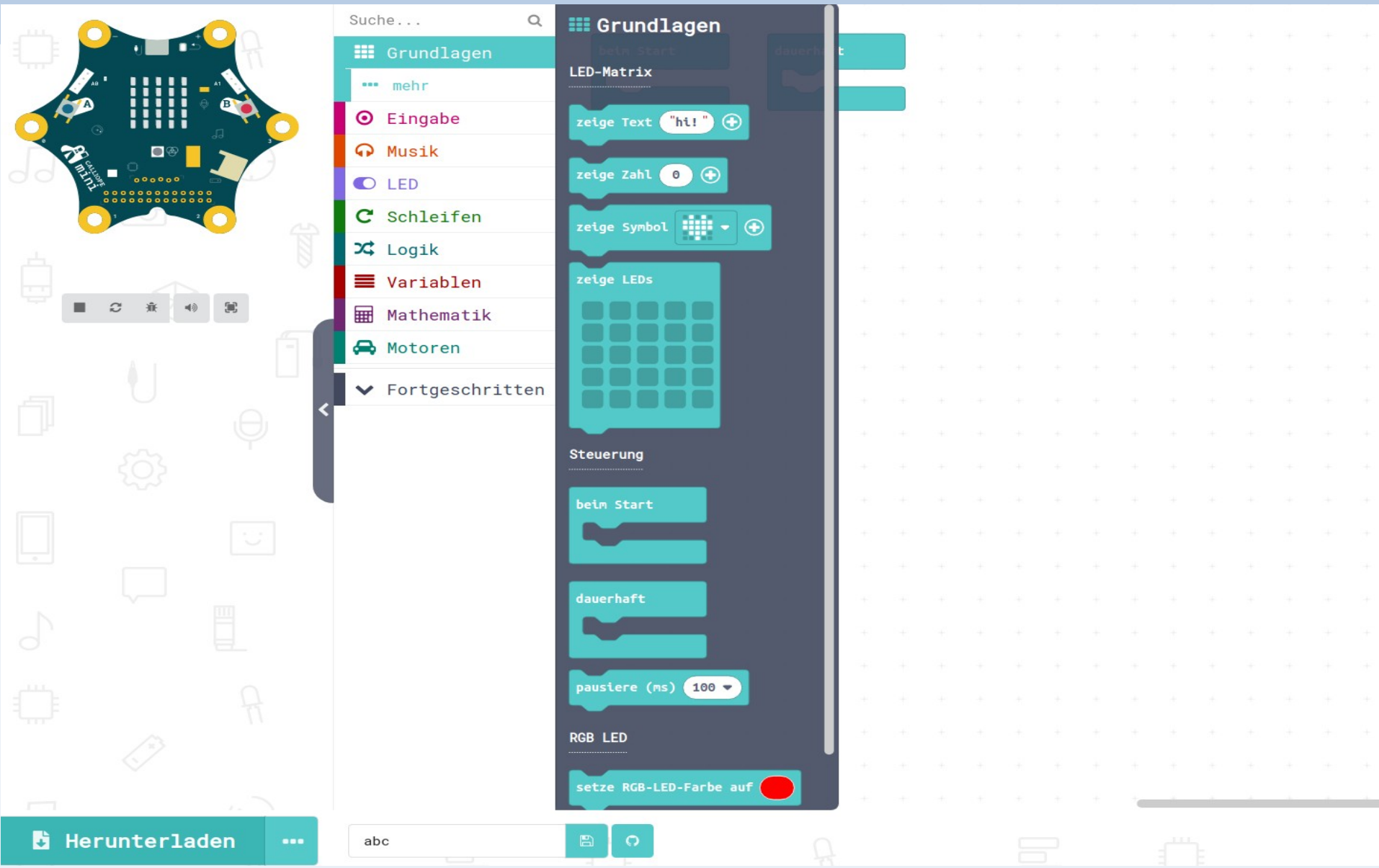
Unterrichtl. Voraussetzungen

- Algorithmische Grundstrukturen:
 - Anweisung
 - (Einfache) Fallunterscheidung
 - (Wiederholungsschleife)
 - Variablen (kann auch in diesem Projekt erlernt werden)
- Falls länger nicht programmiert, zunächst anhand einfacher Aufgaben wiederholen

Schere-Stein-Papier

- Funkkanal wählen
- Variablen festlegen
- Zufallszahl beim Schütteln erzeugen, in Variable speichern
 - 1: Schere
 - 2: Stein
 - 3: Papier
- LED an – warten – LED aus
- Zufallszahl zum Partner schicken
- Partner soll Zufallszahl auf Display anzeigen
- Partner entscheidet, wer gewonnen hat, und zeigt an

Makecode - Grundlagen

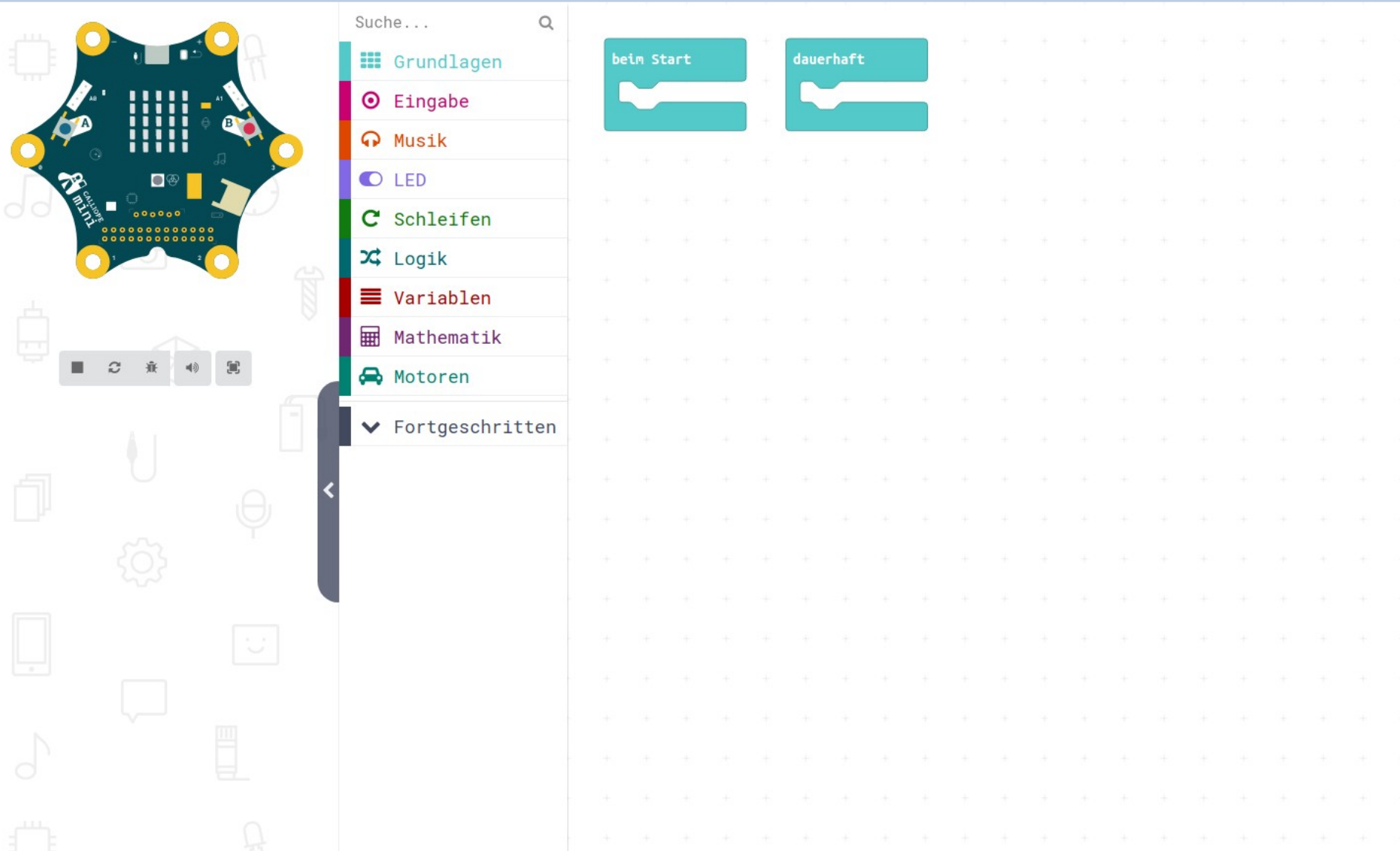


The screenshot displays the Makecode editor interface. On the left, there is a visual representation of a Raspberry Pi board with an LED matrix attached. The main workspace is a dark grey grid where a script is being built. A sidebar on the left lists various categories of blocks: Grundlagen, Eingabe, Musik, LED, Schleifen, Logik, Variablen, Mathematik, Motoren, and Fortgeschritten. The script in the workspace is titled 'Grundlagen' and includes the following blocks:

- LED-Matrix
- zeige Text "ht!"
- zeige Zahl 0
- zeige Symbol (LED matrix icon)
- zeige LEDs (a 4x4 grid of LEDs)
- Steuerung
- beim Start
- dauerhaft
- pausiere (ms) 100
- RGB LED
- setze RGB-LED-Farbe auf (red)

At the bottom left, there is a 'Herunterladen' button. At the bottom center, there is a search bar containing 'abc' and a refresh icon.

Makecode - Funk (I)



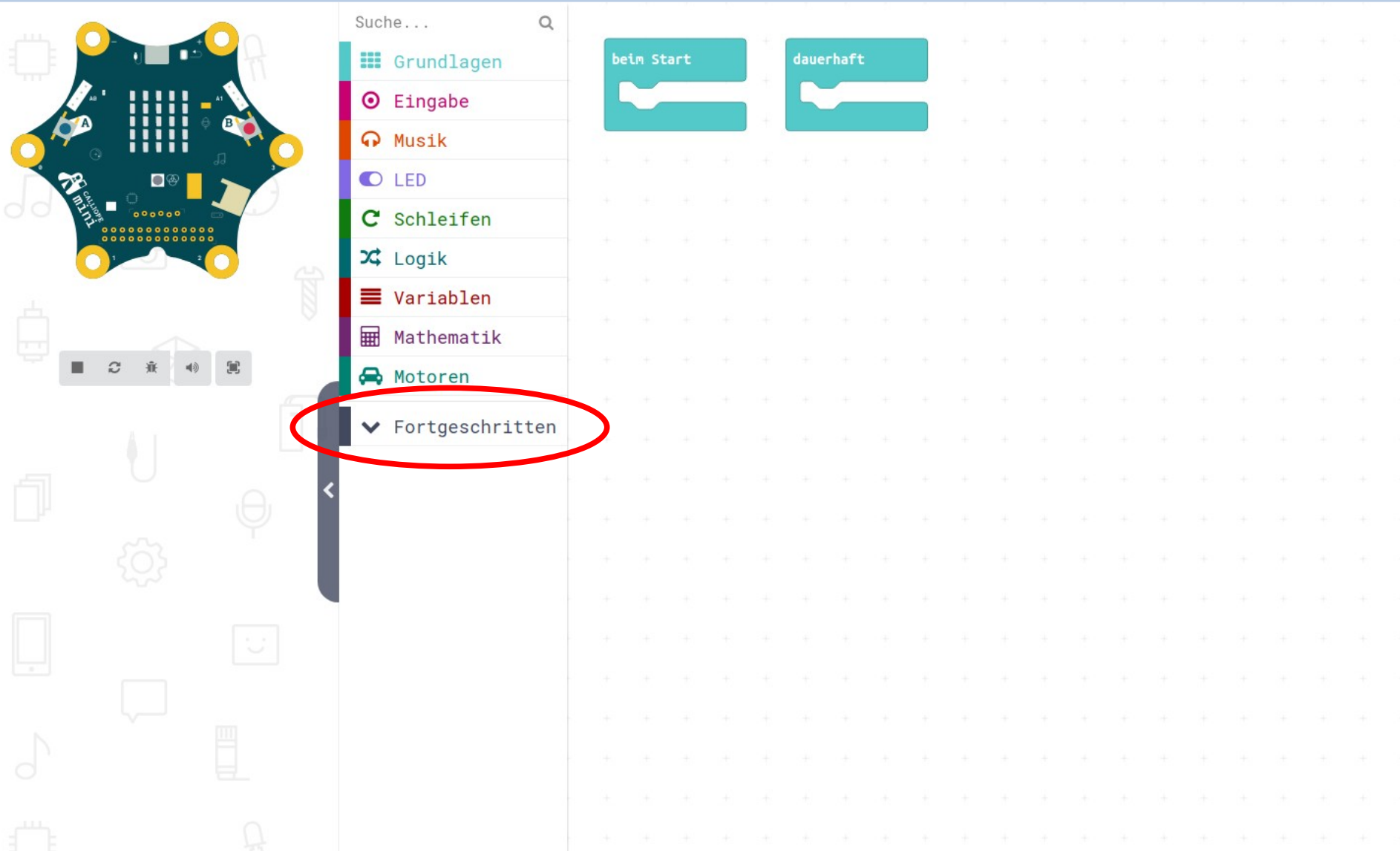
Suche...

- Grundlagen
- Eingabe
- Musik
- LED
- Schleifen
- Logik
- Variablen
- Mathematik
- Motoren
- Fortgeschritten

beim Start

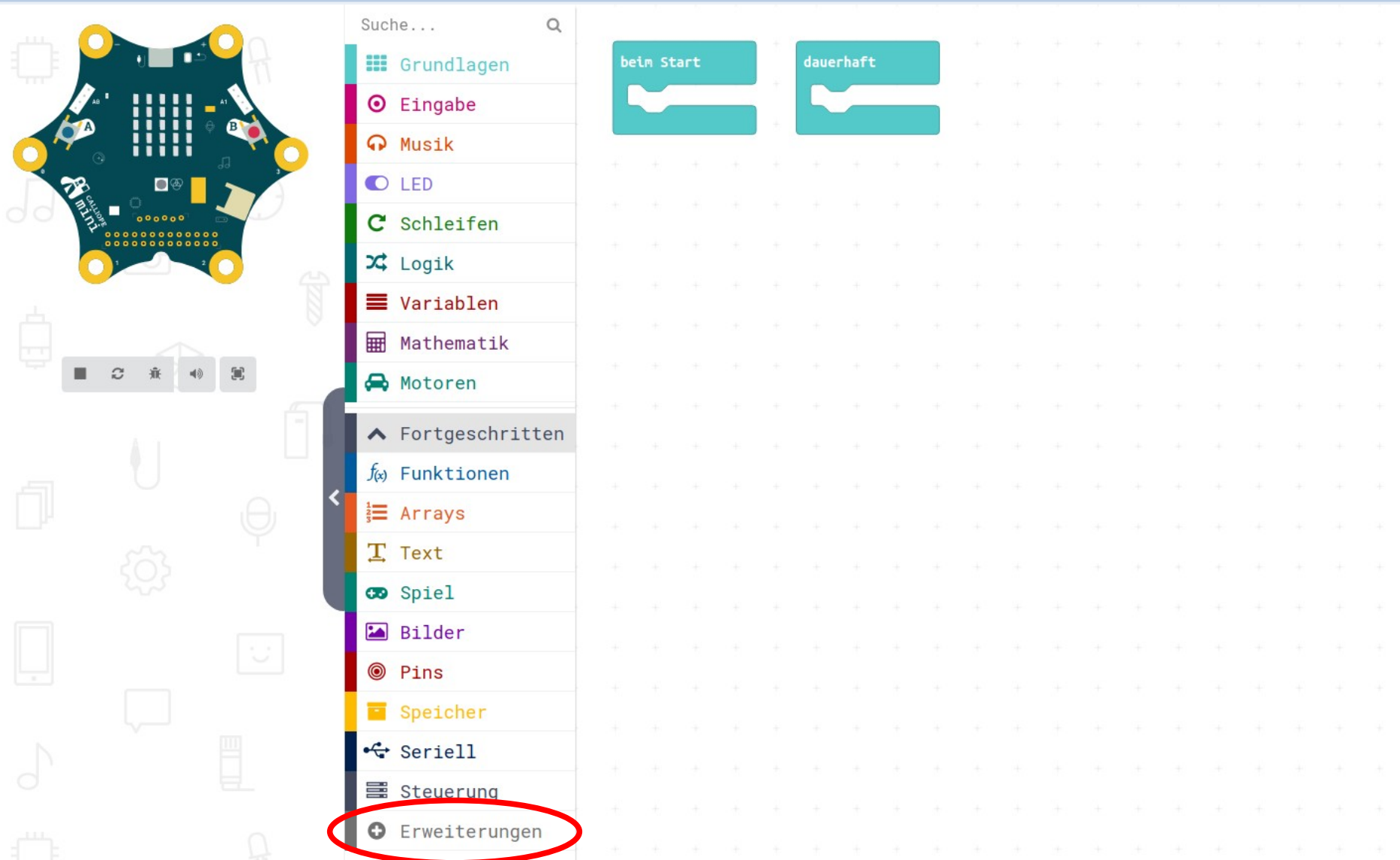
dauerhaft

Makecode - Funk (II)



The image shows the Makecode IDE interface. On the left, there is a visual representation of a microcontroller board (likely a Raspberry Pi Pico) with various components and pins. Below it are several icons representing different hardware components like a USB drive, a mouse, a microphone, a smartphone, a gear, a speech bubble, a music note, and a robot. In the center, a vertical sidebar contains a search bar labeled 'Suche...' and a list of categories: Grundlagen, Eingabe, Musik, LED, Schleifen, Logik, Variablen, Mathematik, Motoren, and Fortgeschritten. The 'Fortgeschritten' category is highlighted with a red oval. To the right of the sidebar, there are two teal buttons labeled 'beim Start' and 'dauerhaft'. The main workspace on the right is a large grid of small plus signs, indicating a blank canvas for code blocks.

Makecode - Funk (III)



The screenshot displays the Makecode IDE interface. On the left, there is a visual representation of a microcontroller board (Lilypad) with various components like an LED, a speaker, and a microphone. Below it are control buttons for play, stop, and volume. The central block palette is open, showing a search bar at the top and a list of categories. The 'Erweiterungen' (Extensions) category at the bottom is circled in red. The workspace on the right contains two teal blocks: 'beim Start' (when started) and 'dauerhaft' (forever), both containing a white key icon. The workspace background is a grid of small plus signs.

Suche...

- Grundlagen
- Eingabe
- Musik
- LED
- Schleifen
- Logik
- Variablen
- Mathematik
- Motoren
- Fortgeschritten
- Funktionen
- Arrays
- Text
- Spiel
- Bilder
- Pins
- Speicher
- Seriell
- Steuerung
- Erweiterungen

beim Start

dauerhaft

Makecode - Funk (IV)

← Zurück

Erweiterungen

Projekt-URL suchen oder eingeben...



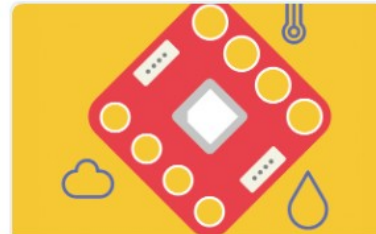
radio-broadcast

Adds new blocks for message communication in



servo

A micro-servo library



CO2-Sensor-SCD40

Calliope mini CO2-Sensor

[Weitere Informationen](#)



Dust-Sensor

HM3301 I2C dust detection sensor

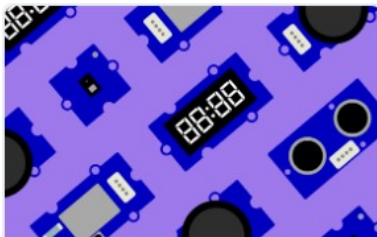
[Weitere Informationen](#)



RGB-Co

RGB-Co for Make

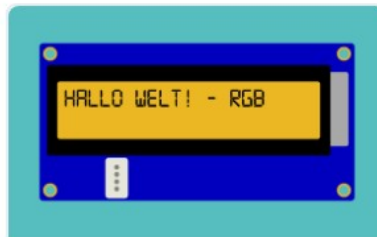
[We](#)



Grove

A Microsoft MakeCode package for Grove Seed

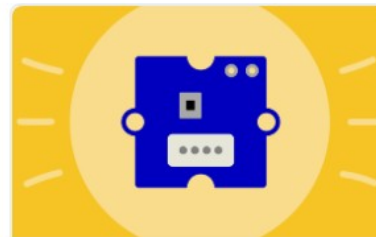
[Weitere Informationen](#)



rgblcd

RGB-LCD package for the Seed Studio Grove - LCD

[Weitere Informationen](#)



Sunlightsensor-SI11...

SI1145 for MakeCode

[Weitere Informationen](#)



SCD30

SCD30 for MakeCode

[Weitere Informationen](#)

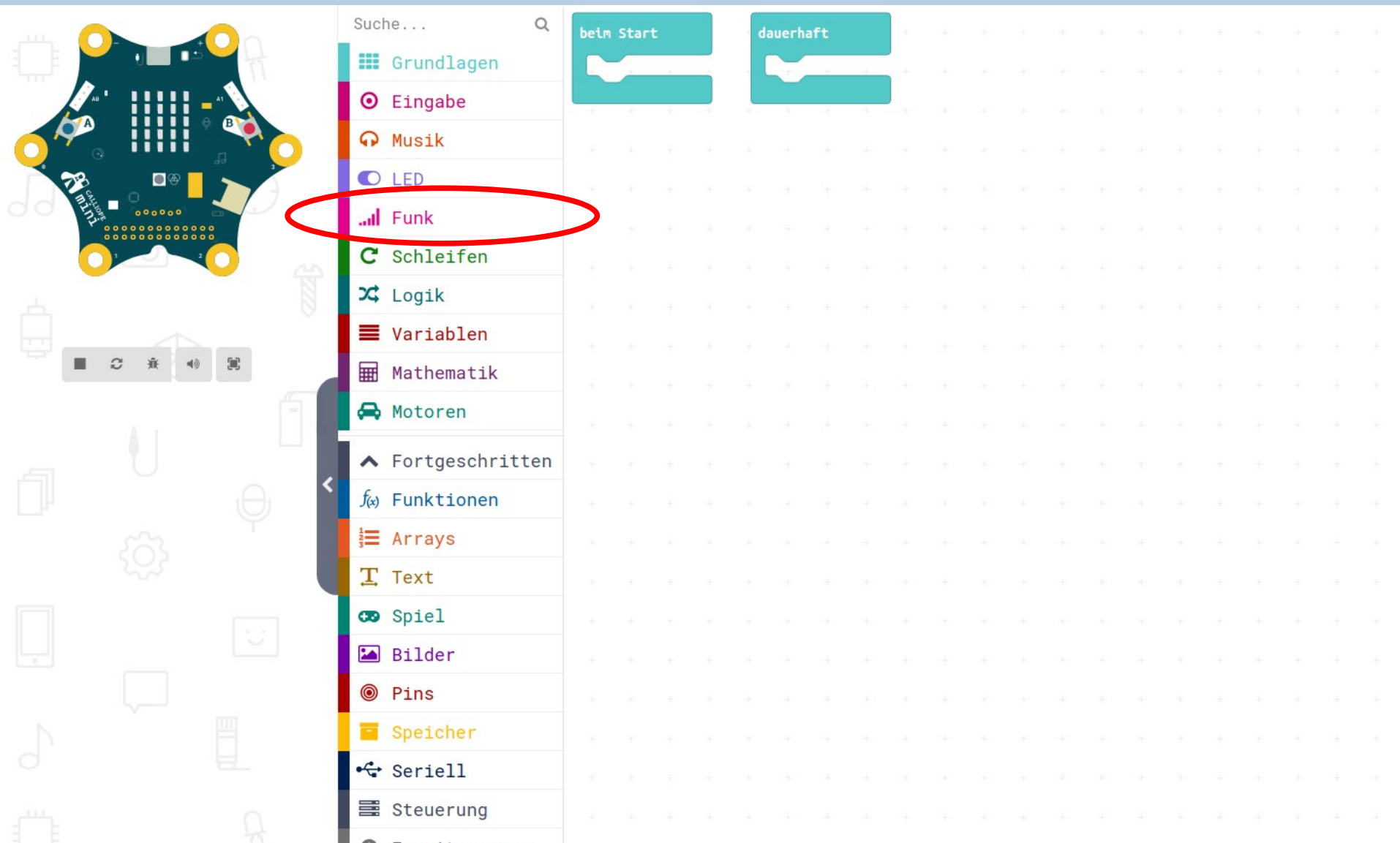


Motion

MakeCode MotionKi

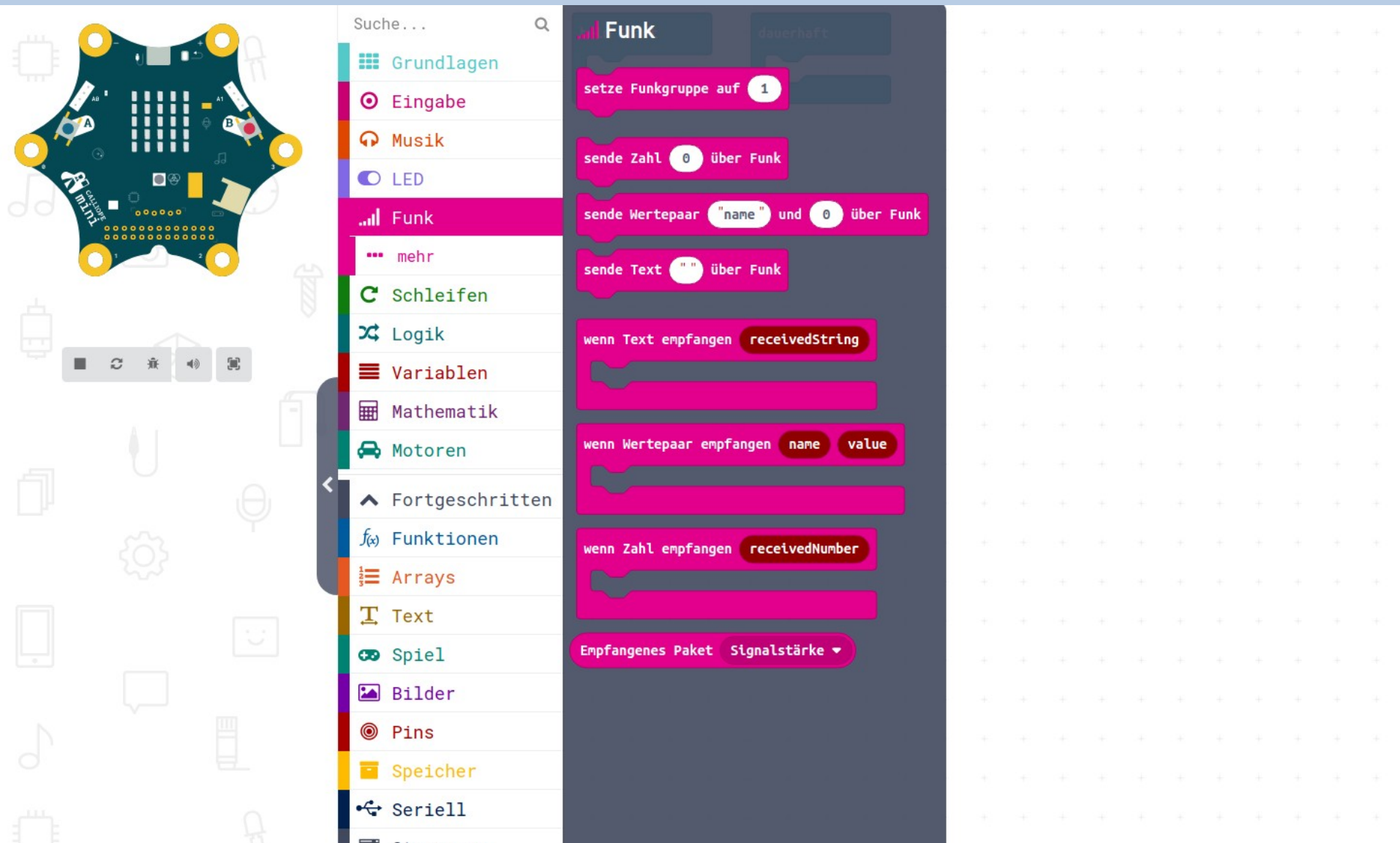
[We](#)

Makecode - Funk (V)



The screenshot displays the Makecode editor interface. On the left, there is a visual representation of a microcontroller board (likely an Arduino Uno) with various components and pins. Below it are several control icons: a square, a refresh symbol, a speaker with a slash, a volume icon, and a square with a plus sign. The central sidebar contains a search bar labeled 'Suche...' and a list of categories. The 'Funk' category, represented by a signal strength icon, is highlighted with a red oval. Other categories include Grundlagen, Eingabe, Musik, LED, Schleifen, Logik, Variablen, Mathematik, Motoren, Fortgeschritten, Funktionen, Arrays, Text, Spiel, Bilder, Pins, Speicher, Seriell, and Steuerung. On the right side of the editor, there are two teal-colored blocks: 'beim Start' and 'dauerhaft', each with a white keyhole icon. The background of the workspace is a light gray grid.

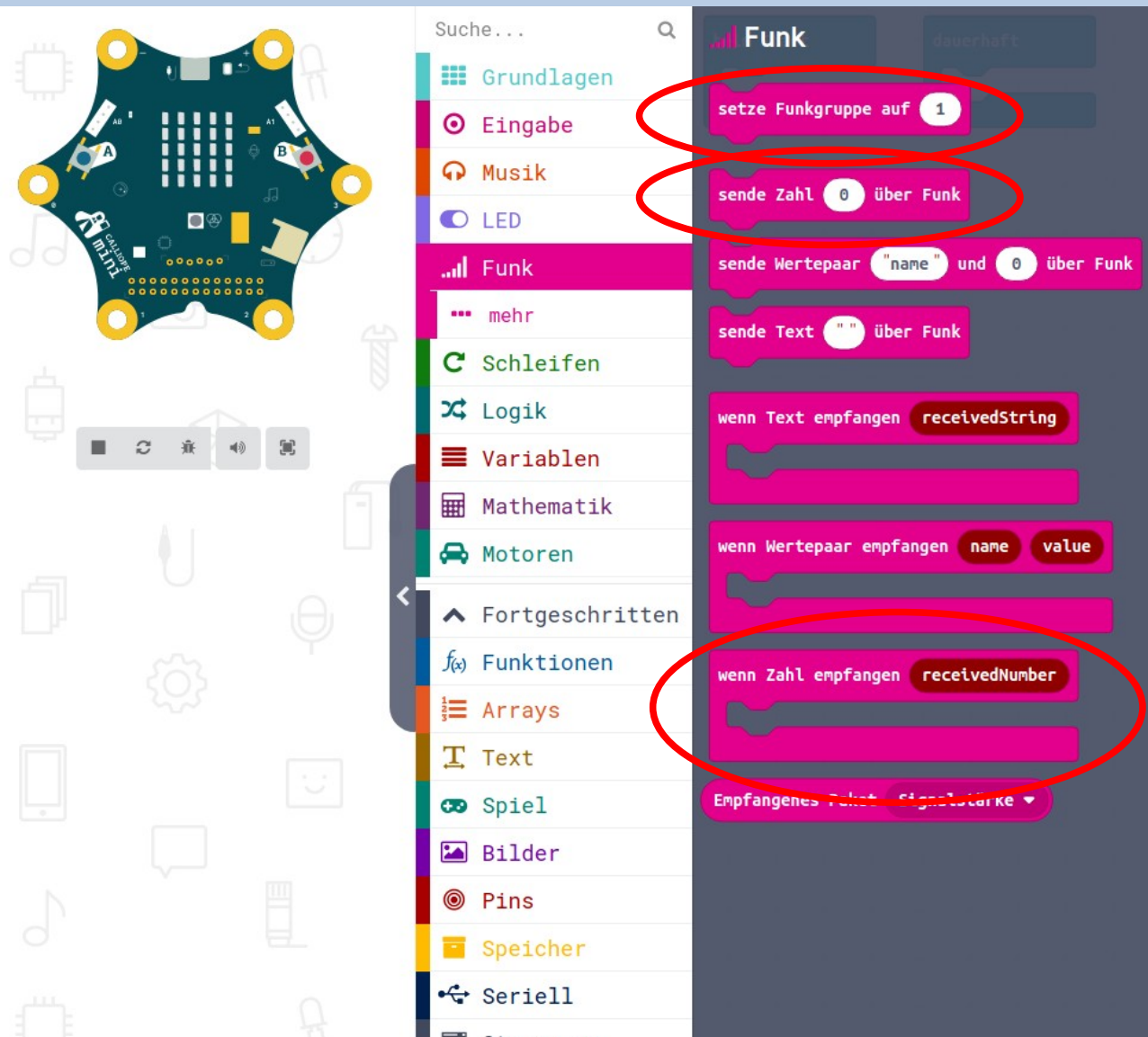
Makecode - Funk (VI)



The screenshot displays the Makecode IDE interface. On the left, a sidebar lists various categories: Grundlagen, Eingabe, Musik, LED, Funk (selected), mehr, Schleifen, Logik, Variablen, Mathematik, Motoren, Fortgeschritten, Funktionen, Arrays, Text, Spiel, Bilder, Pins, Speicher, and Seriell. The main workspace shows a sequence of code blocks under the 'Funk' category:

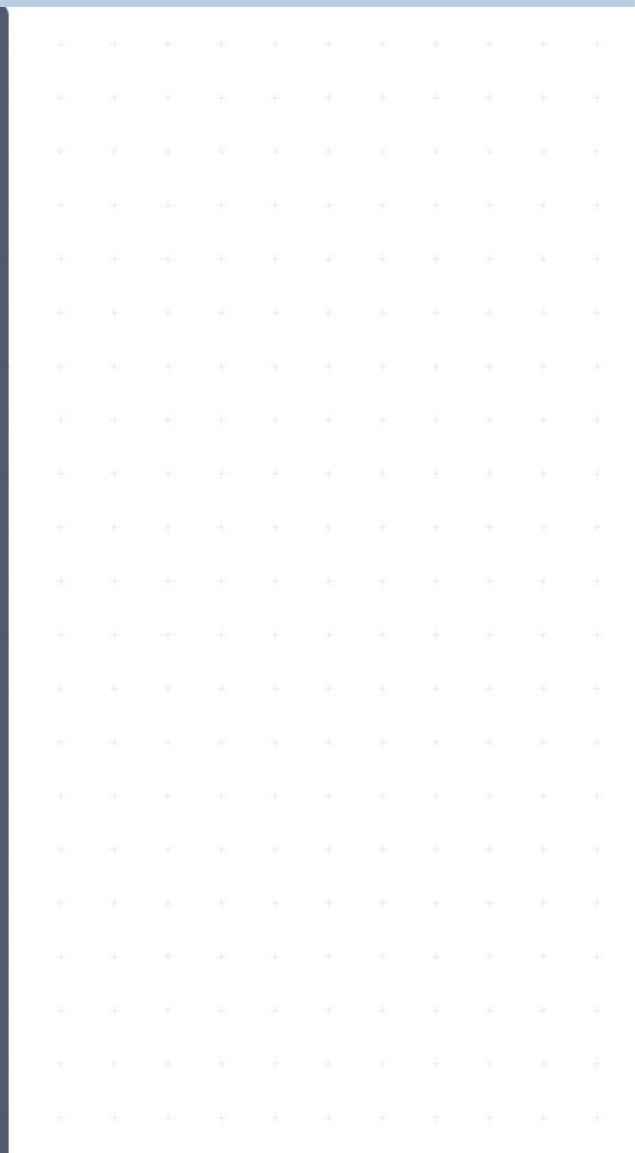
- setze Funkgruppe auf 1
- sende Zahl 0 über Funk
- sende Wertepaar "name" und 0 über Funk
- sende Text "" über Funk
- wenn Text empfangen receivedString
- wenn Wertepaar empfangen name value
- wenn Zahl empfangen receivedNumber
- Empfangenes Paket Signalstärke

Makecode - Funk (VII)

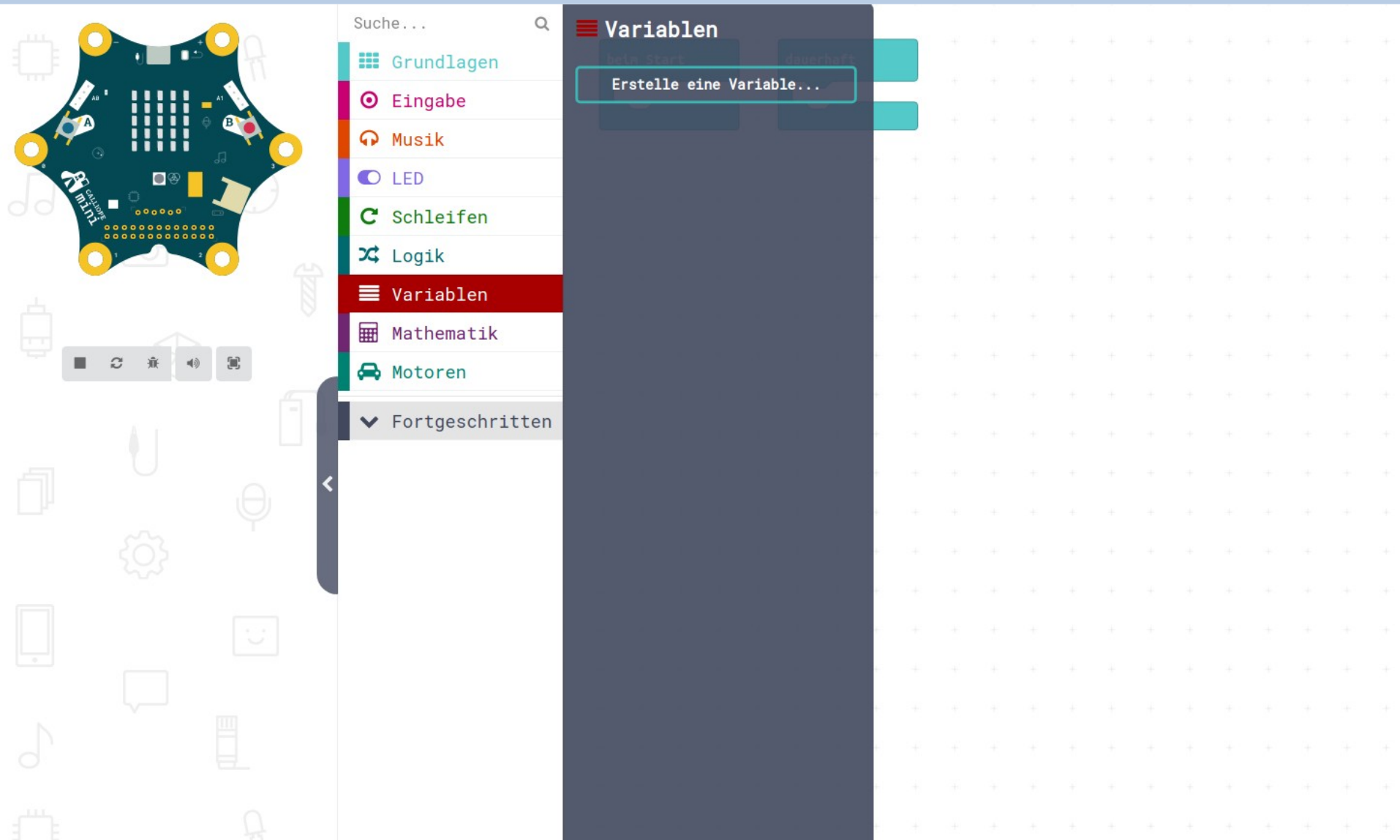


The image shows the Scratch Makecode interface. On the left is a small image of a Raspberry Pi Pico board. The main area is divided into a block palette on the left and a script area on the right. The 'Funk' (Radio) block category is selected in the palette. The script area contains the following blocks:

- setze Funkgruppe auf 1 (circled in red)
- sende Zahl 0 über Funk (circled in red)
- sende Wertepaar "name" und 0 über Funk
- sende Text "" über Funk
- wenn Text empfangen receivedString
- wenn Wertepaar empfangen name value
- wenn Zahl empfangen receivedNumber (circled in red)
- Empfangenes Text: Signalarke



Makecode - Variable



Suche...

- Grundlagen
- Eingabe
- Musik
- LED
- Schleifen
- Logik
- Variablen**
- Mathematik
- Motoren
- Fortgeschritten

Variablen

Erstelle eine Variable...

Makecode - Logik



The screenshot shows the Makecode editor interface. On the left is a visual representation of a microcontroller board. The central panel displays a search bar and a list of block categories: Grundlagen, Eingabe, Musik, LED, Schleifen, Logik (highlighted), Variablen, Mathematik, Motoren, and Fortgeschritten. The right panel shows the 'Logik' block palette with sections for 'Bedingungen' (Conditions), 'Vergleich' (Comparison), and 'Boolescher Wert' (Boolean Value). The 'Bedingungen' section includes 'wenn wahr dann' blocks with plus and minus icons. The 'Vergleich' section includes comparison operators like '0 = 0', '0 < 0', and '"" = ""'. The 'Boolescher Wert' section includes logical operators like 'und', 'oder', 'nicht', 'wahr', and 'falsch'.



Achtung:

- unter „Bedingungen“ sind Fallunterscheidungen gelistet
- unter „Vergleich“ stehen Bedingungen mit Vergleich
- unter „boolescher Wert“ stehen logische Verknüpfungen von Wahrheitswerten

Schere-Stein-Papier

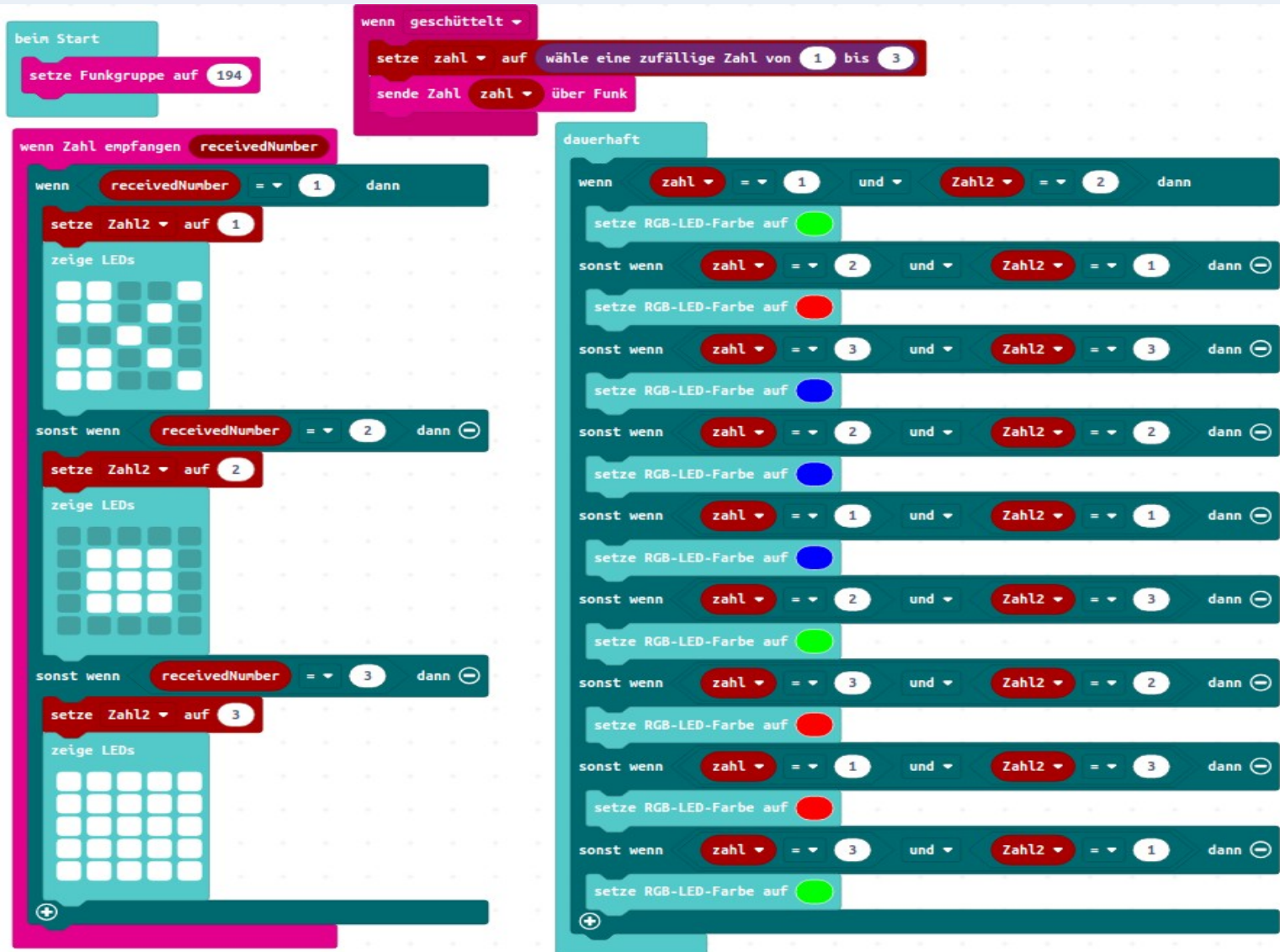
- Funkkanal wählen
- Variablen festlegen
- Zufallszahl beim Schütteln erzeugen, in Variable speichern
 - 1: Schere
 - 2: Stein
 - 3: Papier
- LED an – warten – LED aus
- Zufallszahl zum Partner schicken
- Partner soll Zufallszahl auf Display anzeigen
- Partner entscheidet, wer gewonnen hat, und zeigt an

Musterlösung

```
wenn geschüttelt  
  setze Funkgruppe auf 1  
  setze zufallszahl auf wähle eine zufällige Zahl von 1 bis 3  
  zeige Zahl zufallszahl  
  setze RGB-LED-Farbe auf  
  pausiere (ms) 1000  
  setze RGB-LED-Farbe auf  
  sende Zahl zufallszahl über Funk
```

```
wenn Zahl empfangen receivedNumber  
  zeige Zahl receivedNumber  
  wenn receivedNumber = 1 und zufallszahl = 2 dann  
    zeige Text "Gewonnen"  
  sonst wenn receivedNumber = 2 und zufallszahl = 3 dann  
    zeige Text "Gewonnen"  
  sonst wenn receivedNumber = 3 und zufallszahl = 1 dann  
    zeige Text "Gewonnen"  
  ansonsten  
    zeige Text "verloren"
```

Schülerlösung



The image displays a Scratch script for a number matching game. The script is organized into several sections:

- beim Start** (When green flag clicked):
 - setze Funkgruppe auf 194
- wenn geschüttelt** (When shaken):
 - setze zahl auf wähle eine zufällige Zahl von 1 bis 3
 - sende Zahl zahl über Funk
- wenn Zahl empfangen receivedNumber** (When received number):
 - wenn receivedNumber = 1 dann:
 - setze Zahl2 auf 1
 - zeige LEDs
 - sonst wenn receivedNumber = 2 dann:
 - setze Zahl2 auf 2
 - zeige LEDs
 - sonst wenn receivedNumber = 3 dann:
 - setze Zahl2 auf 3
 - zeige LEDs
- dauerhaft** (Forever loop):
 - wenn zahl = 1 und Zahl2 = 2 dann:
 - setze RGB-LED-Farbe auf green
 - sonst wenn zahl = 2 und Zahl2 = 1 dann:
 - setze RGB-LED-Farbe auf red
 - sonst wenn zahl = 3 und Zahl2 = 3 dann:
 - setze RGB-LED-Farbe auf blue
 - sonst wenn zahl = 2 und Zahl2 = 2 dann:
 - setze RGB-LED-Farbe auf blue
 - sonst wenn zahl = 1 und Zahl2 = 1 dann:
 - setze RGB-LED-Farbe auf blue
 - sonst wenn zahl = 2 und Zahl2 = 3 dann:
 - setze RGB-LED-Farbe auf green
 - sonst wenn zahl = 3 und Zahl2 = 2 dann:
 - setze RGB-LED-Farbe auf red
 - sonst wenn zahl = 1 und Zahl2 = 3 dann:
 - setze RGB-LED-Farbe auf red
 - sonst wenn zahl = 3 und Zahl2 = 1 dann:
 - setze RGB-LED-Farbe auf green

Schülerlösung (Zoom)

```
beim Start
  setze Funkgruppe auf 194

wenn geschüttelt
  setze zahl auf wähle eine zufällige Zahl von 1 bis 3
  sende Zahl zahl über Funk

wenn Zahl empfangen receivedNumber
  wenn receivedNumber = 1 dann
    setze Zahl2 auf 1
    zeige LEDs
  sonst wenn receivedNumber = 2 dann
    setze Zahl2 auf 2
    zeige LEDs

dauerhaft
  wenn zahl = 1 und Zahl2 = 2 dann
    setze RGB-LED-Farbe auf Grün
  sonst wenn zahl = 2 und Zahl2 = 1 dann
    setze RGB-LED-Farbe auf Rot
  sonst wenn zahl = 3 und Zahl2 = 3 dann
    setze RGB-LED-Farbe auf Blau
  sonst wenn zahl = 2 und Zahl2 = 2 dann
    setze RGB-LED-Farbe auf Blau
  sonst wenn zahl = 1 und Zahl2 = 1 dann
    setze RGB-LED-Farbe auf Blau
  sonst wenn zahl = 2 und Zahl2 = 3 dann
    setze RGB-LED-Farbe auf Grün
```

Worauf es ankommt / potentielle Schwierigkeiten

- Funkkanal mit Partner absprechen (identisch) und mit Gruppe (keiner sonst)
- Variable benutzen, um sich eigene Zahl zu merken
- Fallunterscheidung ist komplex!
- Gleichzeitigkeit: ein Programm für Senden und Empfangen

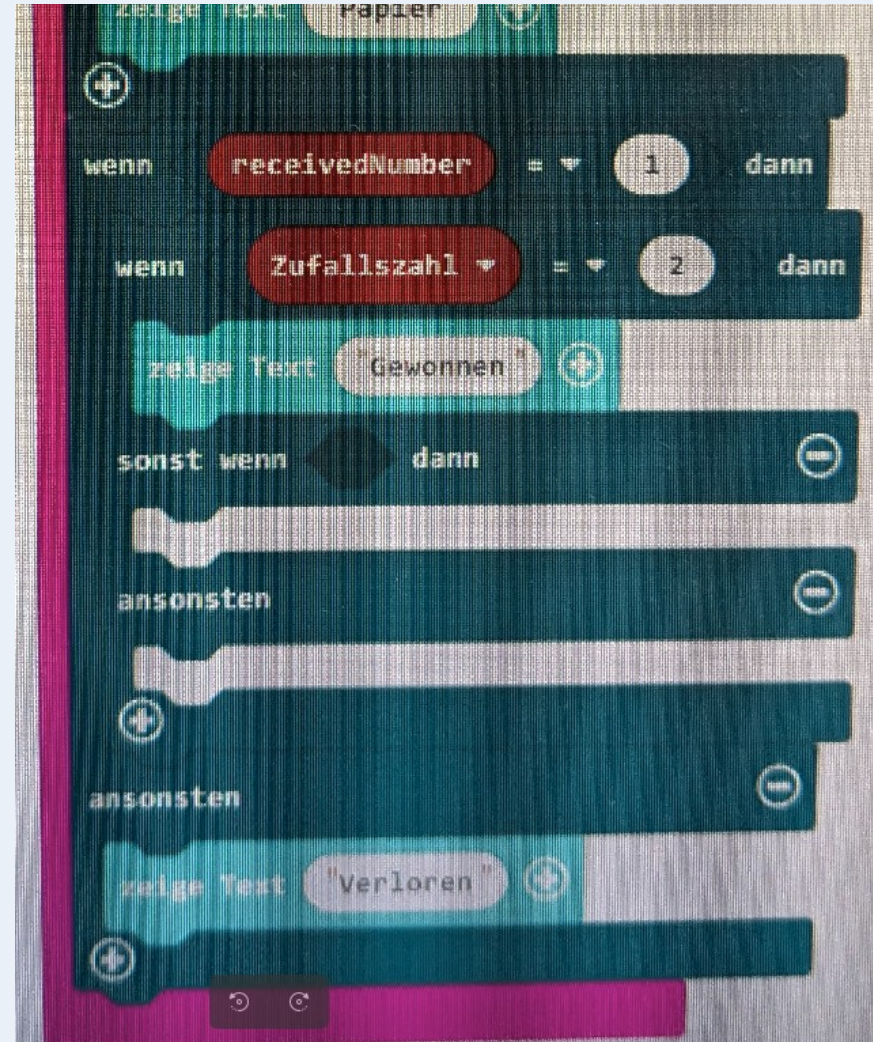
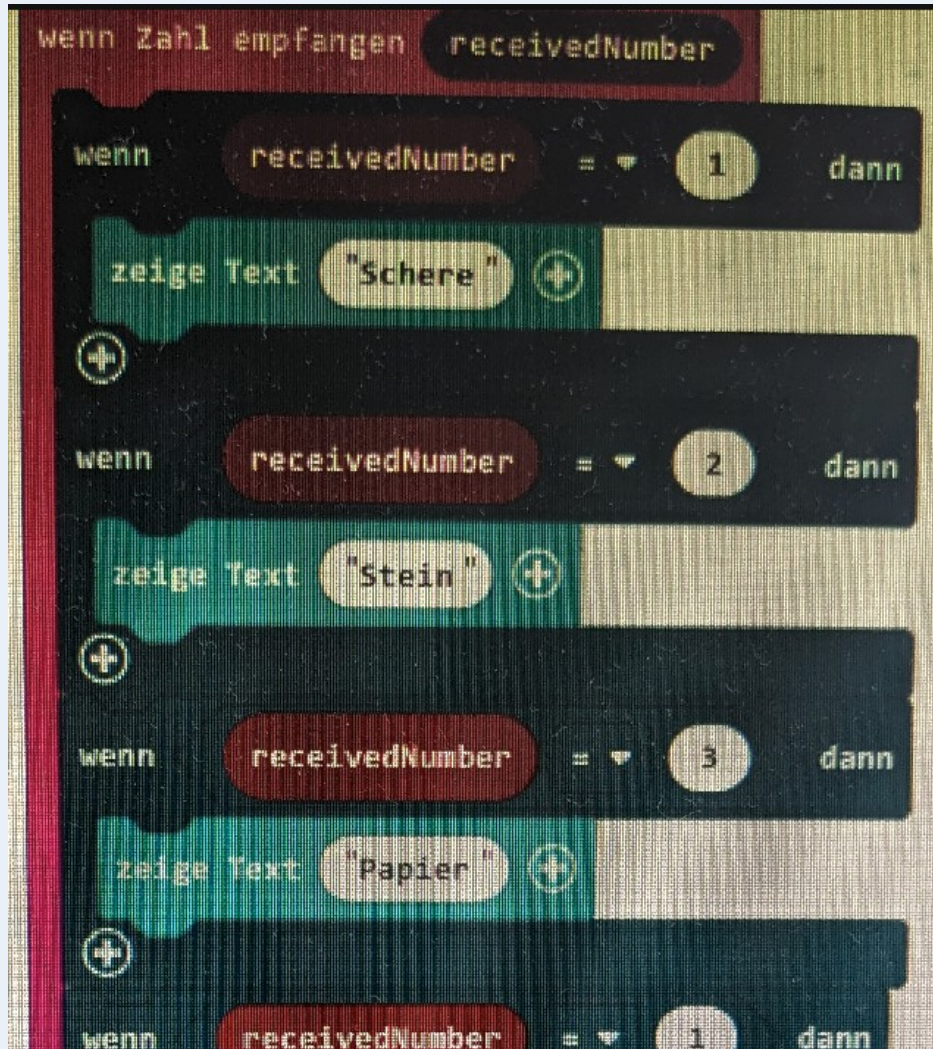
Schere-Stein-Papier – Variante

- Funkkanal wählen
- Variablen festlegen
- Zufallszahl beim Schütteln erzeugen, in Variable speichern
 - 1: Schere
 - 2: Stein
 - 3: Papier
- LED an – warten – LED aus
- Zufallszahl zum Partner schicken
- Partner soll Zufallszahl auf Display anzeigen
- Partner entscheidet, wer gewonnen hat, und zeigt an
- **Variante:** Partner entscheidet, wer gewonnen hat und sendet dies als Text zurück → Empfänger zeigt empfangenen Text an

Weitere Hinweise

- Merke: Trotz guter Anleitung auf inf-schule viel Hilfe durch Lehrkraft nötig
- Zusatzaufgaben für die Schnellen „erfinden“, z.B.:
 - eigene Ausgabezeichen erfinden
 - mit Farben arbeiten (z.B. Farbcode für „gewinnen“ und „verlieren“)
 - Gewinnmelodie ausdenken und einprogrammieren
 - „Geschwindigkeit“ der Anzeige ändern (→ mit Pausen arbeiten...)
- Zusatzaufgaben am besten, wenn Phantasie und eigenes Entdecken gefordert und gefördert
- Immer wieder klar machen: Beim Programmieren passieren Fehler → der erste Lösungsversuch funktioniert oft nicht so wie gewünscht...
- Was tun: Noch einmal die Anleitung lesen; nach dem Fehler suchen; den Nachbarn fragen; die Lehrkraft fragen (in dieser Reihenfolge)

Sammlung: SuS-Ideen



Fragen



Gibt es Fragen ?