

Handreichung PhC

1. Kurzvorstellung

Programmieren eines Zahlenspiels mit MicroBit

2. Einordnung in die Lehrpläne

Klasse 8 an Oberschulen im Lernbereich 2 als Projektarbeit, in Zusammenarbeit mit dem Fach Physik, als Vorbereitung auf die 9. Klasse LB1(Halbleiter).

Lernbereich 2: Informationen verarbeiten: Modell – Algorithmus – Lösung 17 Ustd.

Kennen grundlegender Programmstrukturen <ul style="list-style-type: none"> - Begriff: Algorithmus (Endlichkeit, Eindeutigkeit, Ausführbarkeit, Allgemeingültigkeit) 	Auswahl schülergerechter Software Endlichkeit der Beschreibung und der Ausführung beachten Zusammenarbeit mit MA
--	--

OS – INF

2019

9

Oberschule

<ul style="list-style-type: none"> - Programmstrukturen: Folge, Wiederholung, Verzweigung - eine Darstellungsform Beherrschen der Umsetzung des Modells an einfachen Beispielen	Ausgangspunkt: umgangssprachliche Handlungsbeschreibung Struktogramm, Ablaufplan, Zustandsdiagramm
Kennen des Problemlöseprozesses <ul style="list-style-type: none"> - Problemanalyse, Lösungsentwurf, Umsetzung, Test, Dokumentation 	Teilschritte aus Problemen herauslösen: Zeichnen einer Figur, Vertauschen von zwei Elementen, Ansteuern eines Kanals ⇒ Methodenkompetenz
<ul style="list-style-type: none"> - Lösen eines einfachen Problems unter Nutzung der Programmstrukturen 	Entwurf: auch Modell/Algorithmus Umsetzung: auch Implementierung Test: auch Kritik und Korrektur Sortieren, einfache Steuerung, einfache Makros, Erzeugen von Grafiken
Übertragen der Kenntnisse zum Problemlöseprozess auf <ul style="list-style-type: none"> - selbstständiges Lösen einfacher Probleme 	Partnerarbeit Ampelsteuerung, einfache Automaten, Aufgaben in einfachen grafischen Programmierumgebungen
<ul style="list-style-type: none"> - kritische Bewertung der Resultate 	⇒ Werteorientierung: Handlungs- und Urteilsfähigkeit

3. Lernziele

Kognitive:

- Die SuS lernen den Umgang mit dem Micro:bit kennen.
- Die SuS gewinnen einen ersten Einblick in die Thematik Halbleiter.

Affektive:

- Die SuS sind in der Lage anderen SuS ihre Schaltung und ihr Programm vorzustellen

Psychomotorische:

- Die SuS können eine einfache Schaltung, mit dem Micro:bit als Zentrales Element, zusammenbauen

4. Voraussetzungen

Materiell:

Material pro Schüler oder pro Gruppe

- 1 Micro:bit + Micro:bit-Steckplatz
- 1 Micro USB-Kabel
- 1 Batterie + Kabel
- Min 11 Steckkabel
- 3 Rote LEDs
- 1 Grüne LED
- 1 Gelbe LED
- 1 Steckbrett

Fachlich:

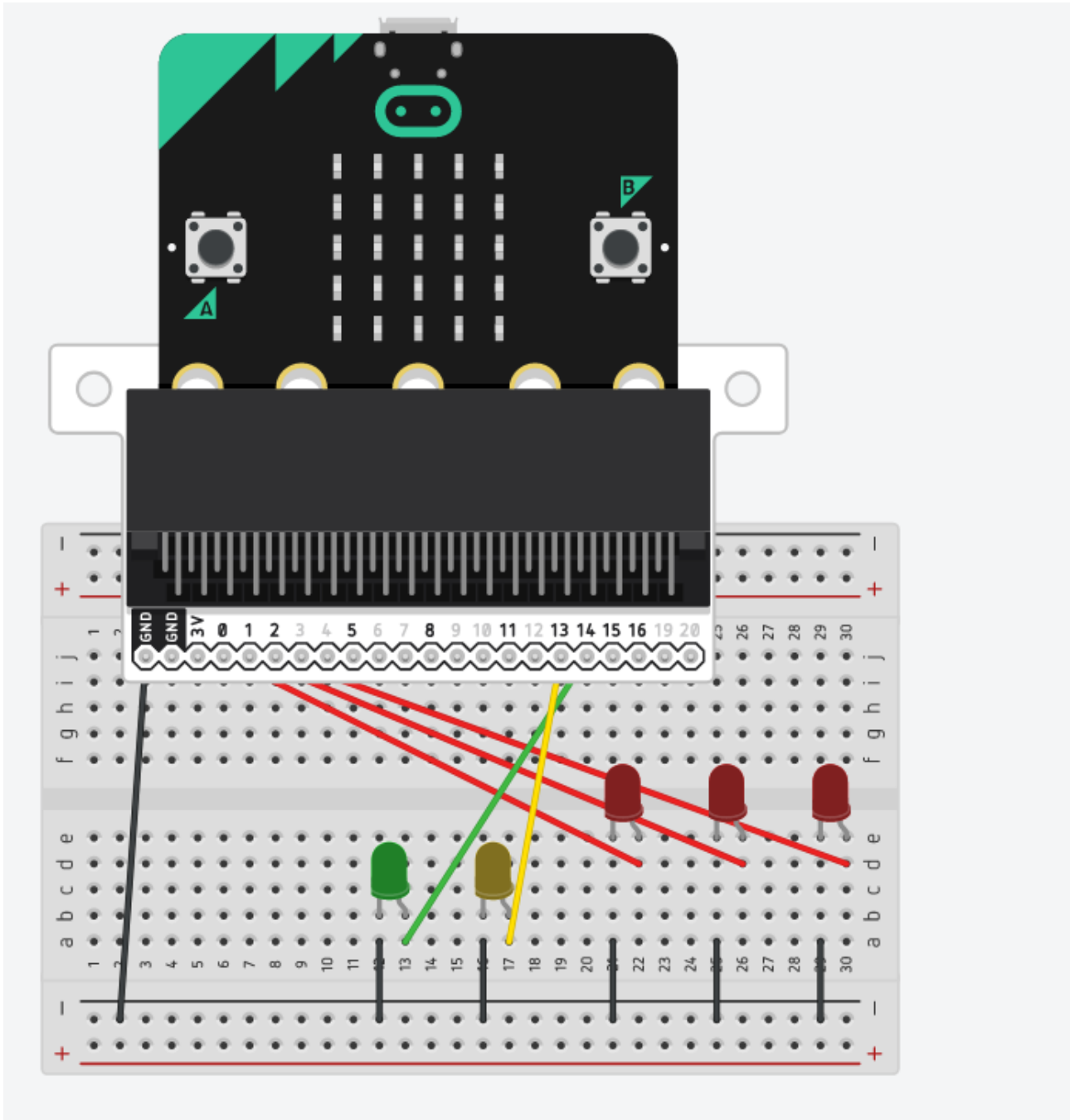
SuS sollten Vorwissen über den Micro:bit besitzen sowie dessen Programmierungsmöglichkeiten besitzen.

Technisch:

Alle SuS oder jedes Team sollte einen Rechner mit einem bekannten Betriebssystem und Internetanbindung zur Verfügung haben.

5. Kurzdarstellung

Schaltung

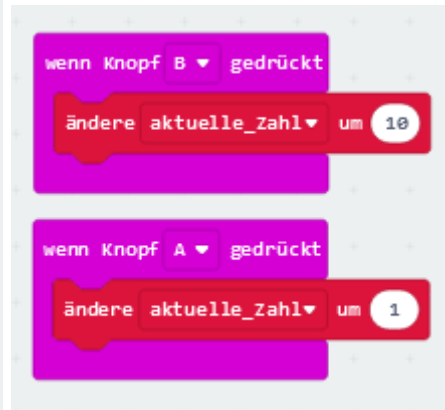


Programmierung

Blöcke

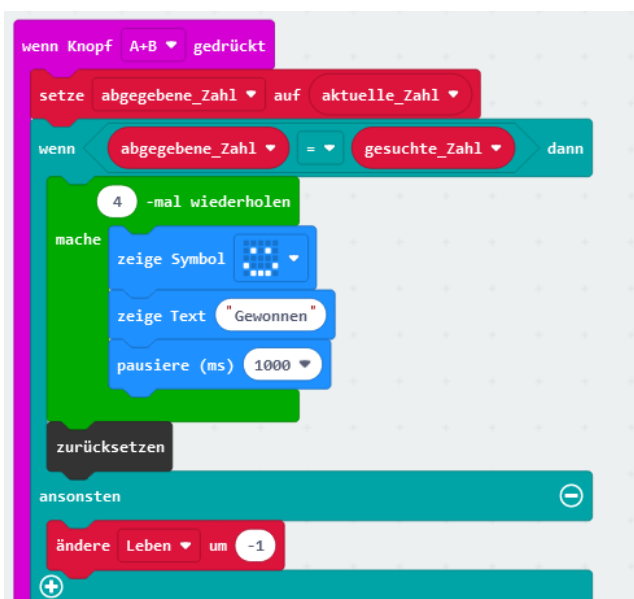


```
beim Start
setze gesuchte_Zahl auf wähle eine zufällige Zahl von 0 bis 100
setze aktuelle_Zahl auf 0
setze Leben auf 6
setze abgegebene_Zahl auf 1000
schreibe analogen Wert vorpin P0 auf 1023
schreibe analogen Wert vorpin P1 auf 1023
schreibe analogen Wert vorpin P2 auf 1023
```



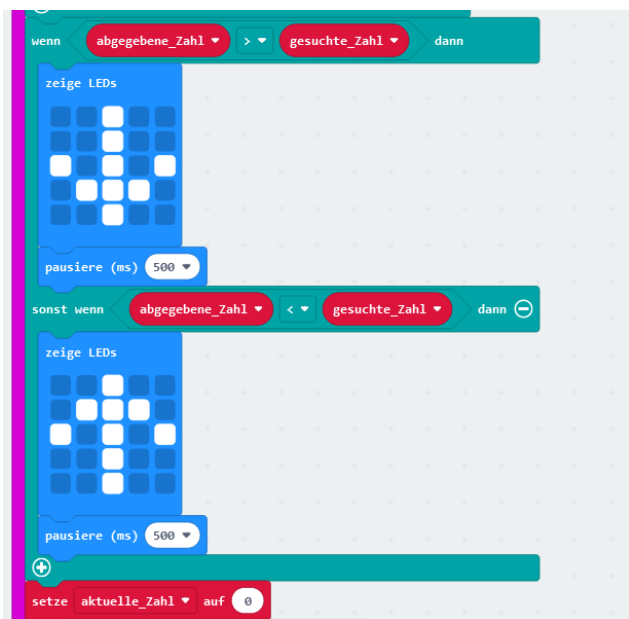
```
wenn Knopf B gedrückt
  ändere aktuelle_Zahl um 10

wenn Knopf A gedrückt
  ändere aktuelle_Zahl um 1
```



```
wenn Knopf A+B gedrückt
  setze abgegebene_Zahl auf aktuelle_Zahl
  wenn abgegebene_Zahl = gesuchte_Zahl dann
    mache 4 -mal wiederholen
      zeige Symbol
      zeige Text "Gewonnen"
      pausiere (ms) 1000
    zurücksetzen
  ansonsten
    ändere Leben um -1
```

1



```
wenn abgegebene_Zahl > gesuchte_Zahl dann
  zeige LEDs
  pausiere (ms) 500
sonst wenn abgegebene_Zahl < gesuchte_Zahl dann
  zeige LEDs
  pausiere (ms) 500
setze aktuelle_Zahl auf 0
```

2

```

wenn <Leben > = <5> dann
  schreibe analogen Wert von Pin P2 auf 50
sonst wenn <Leben > = <4> dann
  schreibe analogen Wert von Pin P2 auf 0
sonst wenn <Leben > = <3> dann
  schreibe analogen Wert von Pin P1 auf 50
sonst wenn <Leben > = <2> dann
  schreibe analogen Wert von Pin P1 auf 0
sonst wenn <Leben > = <1> dann
  schreibe analogen Wert von Pin P0 auf 50

```

3

```

sonst wenn <Leben > = <0> dann
  schreibe analogen Wert von Pin P0 auf 0
  4 -mal wiederholen
  mache
    zeige Text "Game Over"
    zeige Symbol
    pausiere (ms) 500
  zurücksetzen

```

4

```

wenn <abgegebene_Zahl > - <gesuchte_Zahl > <≤ > 5 und <abgegebene_Zahl > - <gesuchte_Zahl > <≥ > -5 dann
  schreibe digitalen Wert von Pin P14 auf 1
  schreibe digitalen Wert von Pin P13 auf 1
sonst wenn <abgegebene_Zahl > - <gesuchte_Zahl > <≤ > 10 und <abgegebene_Zahl > - <gesuchte_Zahl > <≥ > -10 dann
  schreibe digitalen Wert von Pin P14 auf 0
  schreibe digitalen Wert von Pin P13 auf 1
ansonsten
  schreibe digitalen Wert von Pin P14 auf 0
  schreibe digitalen Wert von Pin P13 auf 0

```

5

```

dauerhaft
wenn <aktuelle_Zahl > <≥ > 101 dann
  ändere aktuelle_Zahl um -101
sonst wenn <aktuelle_Zahl > <≤ > -1 dann
  ändere aktuelle_Zahl um 101
zeige Text aktuelle_Zahl

```

Python

```
def on_button_pressed_a():  
    global aktuelle_Zahl  
    aktuelle_Zahl += 1  
input.on_button_pressed(Button.A, on_button_pressed_a)
```

```
def on_button_pressed_ab():  
    global abgegebene_Zahl, Leben, aktuelle_Zahl  
    abgegebene_Zahl = aktuelle_Zahl  
    if abgegebene_Zahl == gesuchte_Zahl:  
        for index in range(4):  
            basic.show_icon(IconNames.HAPPY)  
            basic.show_string("Gewonnen")  
            basic.pause(1000)  
        control.reset()  
    else:  
        Leben += -1  
    if abgegebene_Zahl > gesuchte_Zahl:  
        basic.show_leds("""  
            . . # . .  
                . . # . .  
                # . # . #  
                . # # # .  
                . . # . .  
            """)  
        basic.pause(500)  
    elif abgegebene_Zahl < gesuchte_Zahl:  
        basic.show_leds("""  
            . . # . .  
                . # # # .  
                # . # . #  
                . . # . .  
                . . # . .  
            """)
```

```

        """)
        basic.pause(500)
aktuelle_Zahl = 0
if Leben == 5:
    pins.analog_write_pin(AnalogPin.P2, 50)
elif Leben == 4:
    pins.analog_write_pin(AnalogPin.P2, 0)
elif Leben == 3:
    pins.analog_write_pin(AnalogPin.P1, 50)
elif Leben == 2:
    pins.analog_write_pin(AnalogPin.P1, 0)
elif Leben == 1:
    pins.analog_write_pin(AnalogPin.P0, 50)
elif Leben == 0:
    pins.analog_write_pin(AnalogPin.P0, 0)
    for index2 in range(4):
        basic.show_string("Game Over")
        basic.show_icon(IconNames.SAD)
        basic.pause(500)
    control.reset()
if abgegebene_Zahl - gesuchte_Zahl <= 5 and abgegebene_Zahl - gesuchte_Zahl >= -5:
    pins.digital_write_pin(DigitalPin.P14, 1)
    pins.digital_write_pin(DigitalPin.P13, 1)
elif abgegebene_Zahl - gesuchte_Zahl <= 10 and abgegebene_Zahl - gesuchte_Zahl >= -10:
    pins.digital_write_pin(DigitalPin.P14, 0)
    pins.digital_write_pin(DigitalPin.P13, 1)
else:
    pins.digital_write_pin(DigitalPin.P14, 0)
    pins.digital_write_pin(DigitalPin.P13, 0)
input.on_button_pressed(Button.AB, on_button_pressed_ab)

def on_button_pressed_b():
    global aktuelle_Zahl
    aktuelle_Zahl += 10

```

```
input.on_button_pressed(Button.B, on_button_pressed_b)
```

```
abgegebene_Zahl = 0
```

```
Leben = 0
```

```
aktuelle_Zahl = 0
```

```
gesuchte_Zahl = 0
```

```
gesuchte_Zahl = randint(0, 100)
```

```
aktuelle_Zahl = 0
```

```
Leben = 6
```

```
abgegebene_Zahl = 1000
```

```
pins.analog_write_pin(AnalogPin.P0, 1023)
```

```
pins.analog_write_pin(AnalogPin.P1, 1023)
```

```
pins.analog_write_pin(AnalogPin.P2, 1023)
```

```
def on_forever():
```

```
    global aktuelle_Zahl
```

```
    if aktuelle_Zahl >= 101:
```

```
        aktuelle_Zahl += -101
```

```
    elif aktuelle_Zahl <= -1:
```

```
        aktuelle_Zahl += 101
```

```
    basic.show_string("" + str((aktuelle_Zahl)))
```

```
basic.forever(on_forever)
```

6. Mögliche Aufgaben

Erstelle Mithilfe des Micro:bit ein Spiel mit folgenden Vorgaben. Hinweis: Erledige nacheinander alle Teilaufgaben um am Ende das fertige Spiel erstellt zu haben.

Erklärung für das Spiel:

Das Spiel soll ein Zahlenratespiel sein. Hierfür soll der Micro:bit eine zufällige Zahl zwischen 0 und 100 generieren und der Spieler soll diese dann erraten. Der Spieler hat 6 Leben und pro falsch abgegebener Zahl verliert er eines. Diese Leben sollen durch 3 rote LEDs angezeigt werden wobei eine LED zwei Leben anzeigt. Um eine Unterscheidung möglich zu machen nutze für 2 Leben den analogen Wert 1023, für ein Leben den analogen Wert 50 und für kein Leben den Wert 0. Bei 0 Leben ist der Spieler Game Over. Die aktuelle Zahl sollte immer auf dem LED Bildschirm angezeigt werden. Mit den Knöpfen A und B soll die Zahl entweder um 1 oder 10 erhöht werden. Der Knopf A+B soll die aktuelle Zahl vom Bildschirm dann abgeben. Nach jeder abgegebenen Zahl soll der Micro:bit anzeigen ob die gesuchte Zahl kleiner oder größer als die abgegebene Zahl ist. Wenn die gesuchte Zahl nur noch 10 von der abgegebenen Zahl abweicht soll eine gelbe LED leuchten, wenn die gesuchte Zahl sogar nur 5 von der abgegebenen abweicht so soll zusätzlich noch eine grüne LED leuchten. Wenn die abgegebene Zahl und die gesuchte Zahl übereinstimmen hat der Spieler gewonnen.

- a) Lies dir die Erklärung für das Spiel nochmals genau durch und arbeite auf einem Zettel heraus wie viele Variablen du benötigst. Schreibe dafür den Namen und ihren Zweck auf. (Am besten nicht auf das Aufgabenblatt sondern vorne an die Tafel Projizieren)
- b) Baue eine Schaltung auf die folgenden Anforderungen entspricht.
 - Micro:bit + Micro:bit-Steckplatz
 - 3 Rote LEDs verbunden mit GND und den Pins 0,1,2
 - Gelbe LED verbunden mit GND und dem Pin 13
 - Grüne LED verbunden mit GND und dem Pin 14
- c) Programmiere den Start
 - Initialisiere die gesuchte Zahl
 - Initialisiere die Anzahl der Leben
 - Initialisiere die aktuelle Zahl
 - Initialisiere die abgegebene Zahl
 - Initialisiere alle LED-Pins sinnvoll
- d) Programmiere Knopf A und Knopf B
- e) Programmiere Knopf A+B
 - Setze abgegebene Zahl auf aktuelle Zahl
 - Überprüfe ob abgegebene Zahl gleich gesuchte Zahl
 - Zeige gewonnen an falls abgegebene Zahl gleich gesuchte Zahl und setze das Spiel zurück
 - Falls nicht zeige an ob die gesuchte Zahl größer oder kleiner ist
 - Zieh ein Leben ab wenn abgegebene Zahl ungleich gesuchte Zahl
 - Überprüfe ob die Gelbe und Grüne LED leuchten sollten
 - Überprüfe wie viele Leben der Spieler noch hat und schalte entsprechend die roten LEDs
 - Falls der Spieler 0 Leben hat zeig ein Game Over und setze das Spiel zurück

f) Programmiere Dauerhaft

- Die aktuelle Zahl sollte sich nur zwischen 0 und 100 bewegen können
- Die aktuelle Zahl soll auf dem LED Display angezeigt werden