Jens-Uwe Grabowski

Belegarbeit Physical Computing

Technische Universität Dresden

SS 2019

Inhaltsverzeichnis

Einleitung und zentrale Begriffe

Physical Computing bezeichnet programmierbare Geräte, die über Sensoren physikalische Werte aus der Umgebung aufnehmen und über aktive Komponenten Ausgaben tätigen.

Sensoren können z.B. Spannung, Licht u.a. aufnehmen. Über Eingabegeräte – z.B. Schalter, die Spannungen liefern – umfasst das auch Nutzereingaben. Ausgaben können angeschlossene Komponenten steuern oder Informationen an Nutzer ausgeben.

Die Programmierbarkeit basiert meist auf eingebauten Mikroprozessoren. Die Systeme arbeiten ohne persistente Massenspeicher. Um die Programme persistent im Gerät zu speichern, wird ein vom Datenspeicher getrennter Programmspeicher (Flash) benutzt.

Die Interaktion der Mikroprozessoren mit der Umgebung basiert auf analogen oder digitalen Pins, an die Sensoren und Ausgabekomponenten angeschlossen sind oder die dem Benutzer direkt zum Anschluss eigener Bauteile zur Verfügung stehen.

Die Programmierung erfolgt online (z.B. Webeditoren) oder mittels auf einem Computer lokal installierter Software. Die kompilierten Programme werden dann auf das Gerät übertragen. Eine Programmierung direkt auf dem Gerät selbst ist wegen der zu geringen Resourcen kaum möglich.

Die Systeme umfassen Hardware (Prozessor, Speicher, Sensoren, Ausgabe) und Software (Programmierumgebung, Compiler, Bibliotheken, Software zum Übertragen des Programms auf das Gerät). Die verschiedenen Systeme unterscheiden sich hier erheblich. Bezüglich der Hardware liegt ein wesentlicher Unterschied im Integrationsgrad der I/O-Komponenten: während der Arduino im wesentlichen rohe analoge oder digitale Pins zur bereitstellt, ist der Calliope mini durch integrierte Komponenten auch ohne eigene Beschaltung nutzbar. Bezüglich der Software werden grafische oder textorientierte Umgebungen basierend auf verschiedenen Programmiersprachen angeboten. Oft gibt es da zu einer Physical-Computing-Plattform verschiedene Möglichkeiten und Anbieter.

Die Trennung zwischen Physical Computing und Computern ist nicht scharf. So ermöglicht z.B. der Raspberry Pi die Umsetzung klassischer Steuerungs und Regelaufgaben, kann aber auch ein Betriebssystem wie Raspbian betreiben.

Für viele Geräte ist die Software Open-Source, für manche – wie Arduiono – auch die Hardware. Das muss jedoch nicht für alle verwendeten Bauteile gelten.

Kleines Glossar

Mikroprozessor. Integrierter Schaltkreis, der eine Ausführungseinheit (vor allem ALU und Arbeitsspeicher) enthält.

Speicher. Einheit zur Ablage des Programms und der im Programm verwendeten Daten. Bei Mikroprozessoren oft getrennte Speicher.

Compiler. Software zur Übersetzung von Programmen in einer menschenlesbaren Programmiersprache (z.B. JavaScript) in maschinenlesbare Form.

Hex-Datei. Häufiges Format für die maschinenlesbare Form eines Programms. Die wird über spezielle Hardware und Software in einen Mikroprozessor übertragen und kann dann von diesem Ausgeführt werden.

Sensor. Elektronisches Bauteil, dass pysikalische Größen in elektrische Spannungen umsetzt und so einer elektronischen Auswertung zugänglich macht.

LED. Elektronisches Bauteil, das leuchtet wenn es von elektrischem Strom durchflossen wird. Eine

der einfachsten Ausgabe- oder Anzeigemöglichkeiten.

Analog. Spannung, die beliebige Werte in einem gewissen Bereich annehmen kann.

Digital. Spannung, die nur 2 Werte annehmen kann.

Pin. Elektrischer Anschluss eines Geräts. Pins können vom Gerät mit einer Spannung versorgt werden, die von außen benutzt wird (Ausgabe) oder umgekehrt von außen mit einer Spannung versorgt werden, die im Gerät ausgewertet wird (Eingabe). Die Spannungen können jeweils analog oder digital sein.

Open Source. Software oder Hardware, bei der der Nutzer das Recht und die Möglichkeit der Nutzung, Kopie, Veränderung und Weiterverbreitung hat.

Einblick in 3 Physical-Computing-Plattformen

Arduino Uno

Der Arduino Uno besteht aus einem Board mit dem Mikroprozessor ATmega328P und einer Software zur Programmierung (beides Open Source). Das Board selbst enthält keine Sensoren oder Ausgabekomponenten. Deshalb wird der Arduino Uno meist mit einer Experimentierplatine benutzt, auf die diese Bauteile aufgesteckt und mit dem Board verbunden werden.

Im Original wird der Arduino Uno textuell mit einer angepassten C++ Sprache programmiert. Auf dieser Programmierumgebung basieren auch die meisten Dokumentationen, Beispiele und Tutorials. Es existieren aber auch grafische Umgebungen, z.B. Visuino (nicht Open Source).

Der Mikroprozessor

Der Mikroprozessor ATmega328P (Hersteller Microchip Technology Inc.) besteht hauptsächlich aus folgenden Funktionseinheiten ([Atmel01]):

- 8-bit CPU mit 32 Universalregistern,
- 2 KiB RAM Datenspeicher,
- 32 KiB Flash Programmspeicher,
- mehrere digitale Ein- und Ausgänge,
- mehrere A/D Wandler zur Eingabe analoger Pegel.

Die Spezifik der CPU ist wegen des Compilers für den Nutzer (zunächst) weniger relevant. Die I/O-Komponenten spiegeln sich beim Programmieren direkt wieder. Programme bleiben auch beim Trennen der Stromversorgung erhalten (Flash). Auch mit den Speichergrößen macht der Nutzer schnell Bekanntschaft.

Das Board

In der Abbildung 1 sind die wesentlichen Bereiche des Boards gekennzeichnet:

- Der Mikroprozessor (1)
- Stromversorgung u.a. (2)
- USB-Anschluss. Die Umwandlung des USB-Signals erfolgt in einem weiteren Mikroprozessor, der jedoch beim Programmieren durch den Nutzer nicht in Erscheinung tritt (3)
- Digitale Ein- und Ausgänge (4)
- Analoge Eingänge (5)

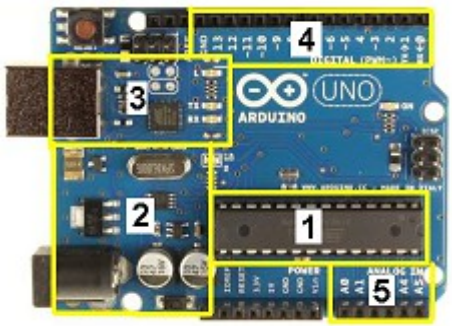

Abbildung 1: Arduino-Board

Die Ein- und Ausgänge (4) und (5) sind mit nur geringer Zusatzbeschaltung mit den entsprechenden Eingängen des Mikroprozessors verbunden.

Experimente

Der Schaltungsaufbau auf einer Experimentierplatine setzt ein Grundverständnis für elektrische Schaltungen voraus und ist auch anfällig für Wackelkontakte u.ä. Ich beschreibe daher zwei einfache Experimente – eins mit Sensor und eins mit LED-Ausgabe – mit ausführlicher Darstellung des Stromkreises.

LED-Ausgabe

In diesem Experiment wird eine LED an einen digitalen PIN angeschlossen und blinkt. Es muss auf die Polung der LED und den Wert des Widerstandes geachtet werden

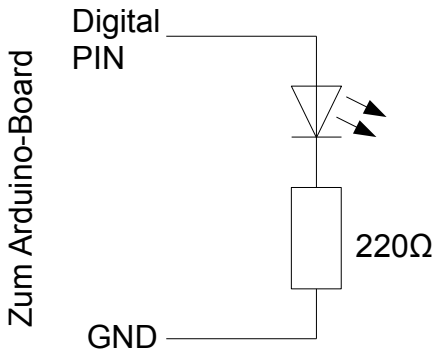

Abbildung 2: LED-Ausgabe Schaltbild

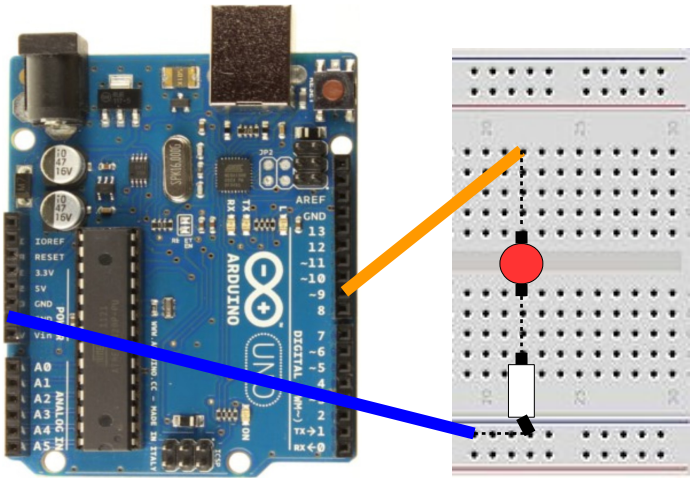

Abbildung 3: LED-Ausgabe Aufbau

Folgender Code lässt die LED blinken.

```
void setup() {                  // wird am Anfang ausgefuehrt
  pinMode(9, OUTPUT);           // der digitale Pin 9 soll Ausgabe sein
                                //   (LED ist mit Pin 9 verbunden)
}

void loop() {                   // wird immer wiederholt
  delay(500);                   // 0,5s warten, damit die LED nicht
                                //   zu schnell binkt
  digitalWrite(9, HIGH);        // LED anschalten - Spannung an Pin 9
  delay(500);                   // 0,5s warten
  digitalWrite(9, LOW);         // LED ausschalten
}
```

Im Unterricht könnte dieser Aufbau und Code zunächst nachvollzogen und dann variiert werden. Variationsmöglichkeiten:

- Blinkgeschwindigkeit und An-Aus-verhältnis ändern,
- SOS "Funken",
- mehrere LED betreiben.

Helligkeitssensor-Eingabe

In diesem Experiment wird ein Fotowiderstand an einen analogen PIN angeschlossen und die Spannung am Spannungsteiler ausgewertet.

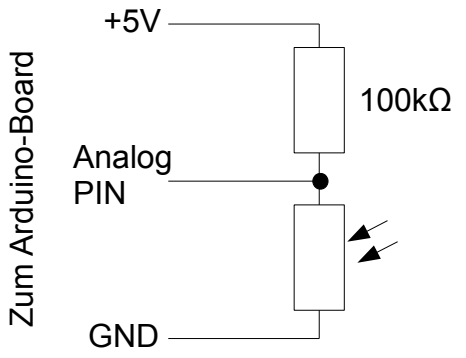

Abbildung 4: Helligkeitssensor Schaltbild

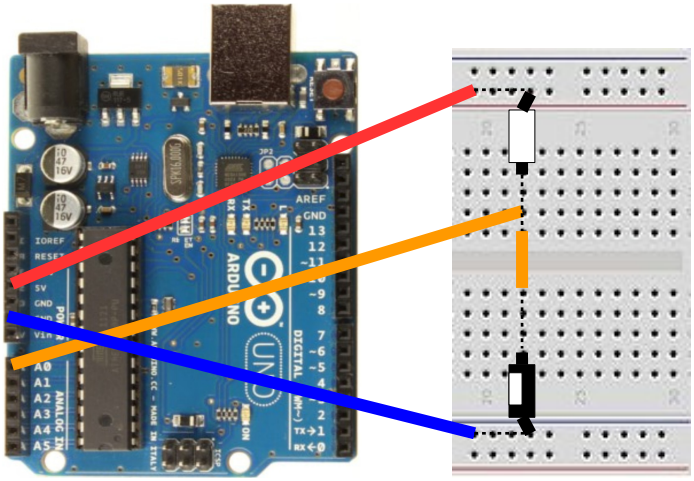

Abbildung 5: Helligkeitssensor Aufbau

Folgender Code gibt die Spannung am analogen PIN über die serielle Konsole aus.

```
int x;                  // Variable um Wert zu speichern

void setup() {
  Serial.begin(9600); // serielle Konsole initialisieren
}

void loop() {
  delay(500);           // 0,5s zwischen 2 Messungen warten
  x = analogRead(0);    // Spannung am Pin A0 messen, in x speichern
  Serial.println(x);    // .. und über die serielle Konsole anzeigen
}
```

Variationsmöglichkeiten:

- Fotowiderstand gegen +5V betreiben,
- maximale und minimale Rückgabe von `analogRead(0)` finden,
- Ausgabe von "hell" oder "dunkel" über die serielle Konsole,
- mit Ausgabe über LED kombinieren.

Calliope mini

Der Calliope mini ist ein Board mit folgenden Hauptkomponenten ([Cal01]):

- 32-bit Mikroprozessor ARM Cortex M0,
- 16 KiB RAM Datenspeicher,
- 256 KiB Flash Programmspeicher,
- USB-Anschluss zur Programmierung
- diversen Sensoren und Ausgabekomponenten.

Die Programmierung kann in einem Webbrowser erfolgen. Auf http://calliope.cc stehen verschiedene Editoren zur Verfügung. Der Editor Makecode erlaubt wahlweise grafische und textuelle Programmierung. Die kompilierten Programmdateien können heruntergeladen und in das USB-Laufwerk, als das der Calliope mini erkannt wird, kopiert werden.

Experimente

Helligkeit

In diesem Experiment wird beim Drücken von Knopf A der Helligkeitsensor ausgewertet und danach über die LED-matrix eine Sonne bzw. ein Mond angezeigt. Damit werden in einem einfachen Beispiel verschiedene Konzepte der Programmierung angesprochen:

- Ereignisbehandlung (Knopf gedrückt),
- Kontrollstrukturen (IF-THEN-ELSE),
- Eingabe (Sensor auswerten),
- Ausgabe.

Das Programm mit dem Makecode sieht so aus:

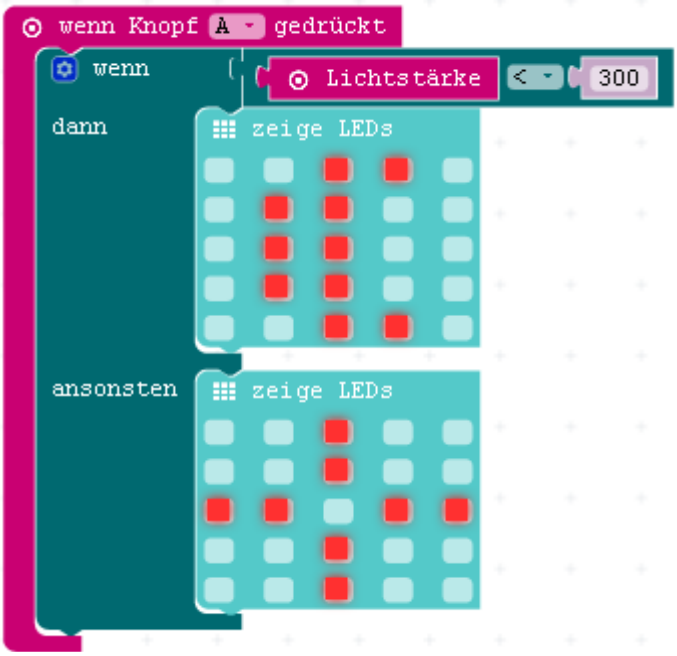

Abbildung 7: Makecode Blockdarstellung

```
1  input.onButtonPressed(Button.A, () => {
2      if (input.lightLevel() < 300) {
3          basic.showLeds(`
4              . . # # .
5              . # # . .
6              . # # . .
7              . # # . .
8              . . # # .
9              `)
10     } else {
11         basic.showLeds(`
12             . . # . .
13             . . # . .
14             # # . # #
15             . . # . .
16             . . # . .
17             `)
18     }
19 })
20
```

Abbildung 6: Makecode Textdarstellung

Größter Primfaktor einer Zahl

In diesem Experiment wird Problem 3 (Largest Prime Factor) von Project Euler ([PEul03]) gelöst. Da die Zahl aus dem gestellten Problem (600851475143) vom Makecode nicht angenommen wurde, wurde nur das angegebene Testbeispiel (13195) getestet.

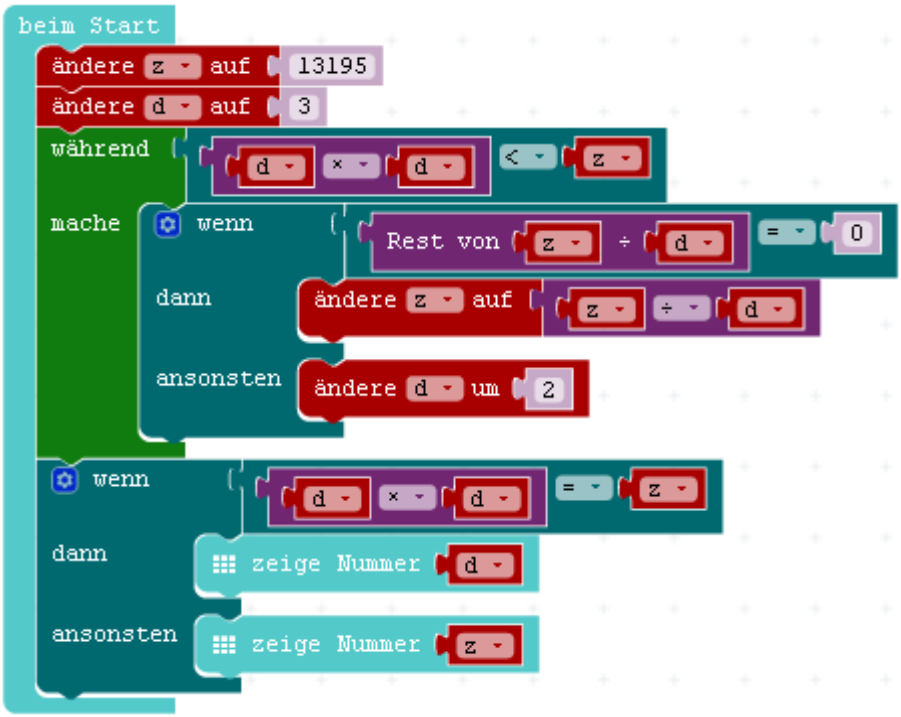

Abbildung 8: Largest Prime Factor

BBC micro:bit

Der BBC micro:bit kann als kleiner Bruder (bezüglich Preis und Ausstattung) des Calliope mini angesehen werden. Prozessor, Programm- und Datenspeicher entsprechen denen des Calliope mini, unterschiedlich sind die Sonsoren und Ausgabekomponenten ([BBC01]):

- analoge und digitale PINs (jeweils Ein- und Ausgabe möglich),
- Sensoren für Licht, Beschleunigung, Magnetfeld, Temperatur,
- Tasten,
- LED-Matrix,
- Funkmodul (Bluetooth und Low-Level).

Die Programmierung kann in einem Webbrowser erfolgen. Auf https://www.microbit.org/de/code/ stehen der Editor Makecode (grafische und textuelle Programmierung in JavaScript) und ein Editor für textuelle Programmierung in Python zur Verfügung. Die kompilierten Programmdateien können heruntergeladen und in das USB-Laufwerk, als das der micro:bit erkannt wird, kopiert werden.

Experimente

Dimmer

In diesem Experiment wird die Helligkeit einer LED beim Drücken von Knopf A verringert und beim Drücken von Knopf B erhöht (in etwa 10s wird der gesamte Helligkeitsbereich durchlaufen). Dabei wird ein Pin als analoge Ausgebe benutzt. Die Beschaltung kann genau wie im Abschnitt LED-Ausgabe im Kapitel Arduino Uno erfolgen. Den Anschluss GND gibt es auch hier und der Pluspol der LED wird hier mit Pin0 des micro:bit verbunden.

Das Programm mit dem Python-Editor sieht so aus:

```
1  from microbit import *
2
3  aOut = 500
4
5  while True:
6      if button_a.isPressed()==1 and aOut>0:
7          aOut -= 1
8          pin0.write_analog(aOut)
9      if button_b.isPressed()==1 and aOut<1023:
10         aOut += 1
11         pin0.write_analog(aOut)
12     sleep(10)
```

Abbildung 9: Dimmer

Pager

In diesem Experiment sendet ein Gerät auf Knopfdruck eine Nachricht an ein zweites, das die Nachricht über eine Ausgabe auf der LED-Matrix signalisiert.

Das Programme von Sender und Empfänger mit dem Python-Editor sehen so aus:

```
1 import radio
2 from microbit import *
3
4 radio.on()
5
6 while True:
7     if button_a.was_pressed():
8         radio.send('ping')
9     sleep(500)
```

Abbildung 10: Pager - Sender

```
1 import radio
2 from microbit import *
3
4 radio.on()
5
6 while True:
7     nachricht = radio.receive()
8     if nachricht == 'ping':
9         display.show("Klingelingeling")
10    sleep(500)
```

Abbildung 11: Pager - Empfänger

Komplexer Versuch

Mit 2 Calliope mini soll über Funk "Schere, Stein, Papier" gespielt werden. In 3 Vorversuchen werden benötigte Teile einzeln ausprobiert.

1. Teilversuch: Eingabe und Anzeige einer Wahl

Aufgabe: Schreibe ein Programm, bei dem der Nutzer durch betätigen der Knöpfe A, B oder A+B eine Auswahl aus Schere, Stein oder Papier trifft. Die Auswahl wird als Zeichenkette in einer Variablen `meine_wahl` gespeichert und dem Nutzer seine Auswahl auf der LED-Matrix als Symbol angezeigt.

Eine mögliche Lösung ist (hier ist nur die Aktion für Knopf A abgebildet):

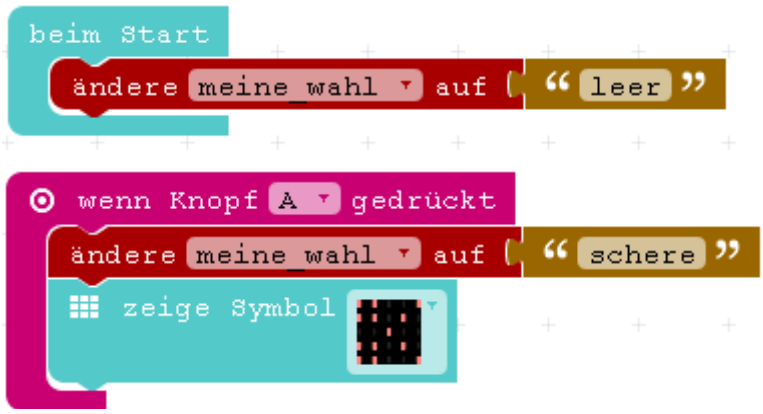

Abbildung 12: Teilversuch: Eingabe und Anzeige einer Wahl

2. Teilversuch: Eingabe von 2 Spielern

Aufgabe: In diesem Teilversuch wird das Spiel "Schere, Stein, Papier" von 2 Spielern auf einem Calliope mini gespielt. Dazu machen die Spieler nacheinander durch Knopfdruck ihre Auswahl.

Erweitere das Programm aus dem 1. Teilversuch. Wenn zum zweiten mal ein Knopf gedrückt wird (2. Spieler), wird die Auswahl in einer Variablen `gegner_wahl` gespeichert, auf der LED-Matrix als Symbol angezeigt und eine Funktion `auswertung` aufgerufen. Die Auswertung zeigt einen lachenden Smily, wenn der erste Spieler gewonnen hat, einen traurigen Smily, wenn der zweite Spieler gewonnen hat und bei einem Unentschieden einen neutralen Smiley.

Eine mögliche Lösung ist (hier ist nur die Aktion für Knopf A abgebildet):

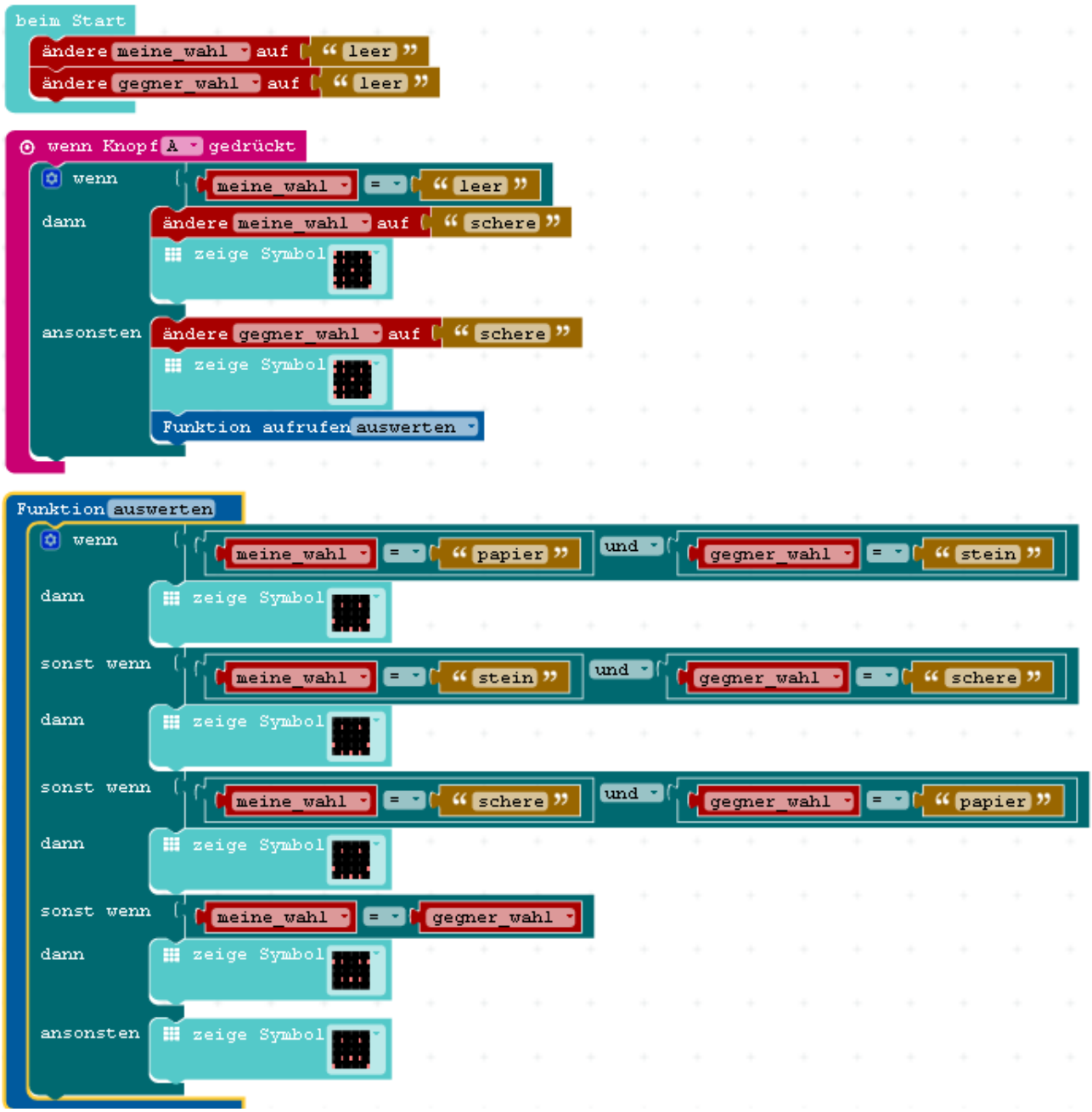

Abbildung 13: Teilversuch: Eingabe von 2 Spielern

3. Teilversuch: Funkübertragung

Aufgabe: Erweitere das Programm aus dem 1. Teilversuch, so dass die gemachte Auswahl per Funk auf den 2. Calliope mini übertragen, dort in einer Variablen `gegner_wahl` gespeichert und die Farb-LED ihre Farbe ändert (als Signal, dass etwas empfangen wurde).

Eine mögliche Lösung ist (hier ist nur die Aktion für Knopf A abgebildet):

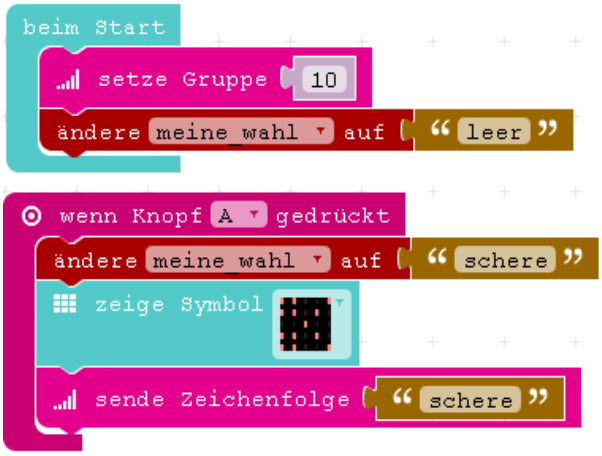

Abbildung 15: Teilversuch: Funkübertragung Sender

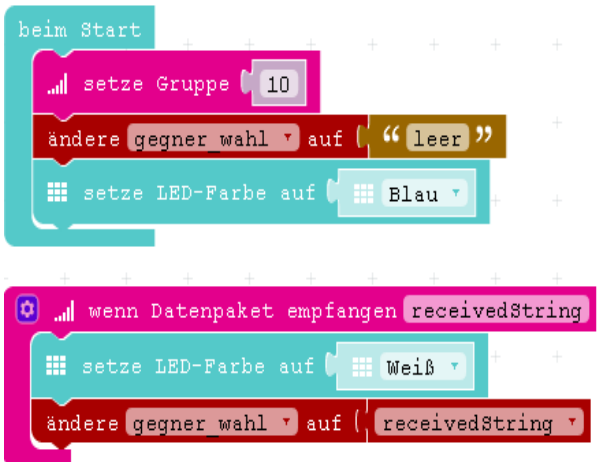

Abbildung 14: Teilversuch: Funkübertragung Empfänger

Gesamter Versuch

Aufgabe: Kombiniere die Teilversuche, so dass 2 Spieler über Funk "Schere, Stein, Papier" spielen können. Auf beide Calliope mini kommt das gleiche Programm. Es soll egal sein, wer zuerst drückt.

Erweitere dazu das Programm aus den 3. Teilversuch. Sowohl beim Drücken eines Knopfes als auch beim Empfang eines Datenpakets (der andere Spieler hat gedrückt), wird am Schluss geprüft, ob beide Variablen einen anderen Wert als "leer" haben. Wenn ja wird

- die Auswahl des anderen Spielers als Symbol angezeigt und
- die Funktion `auswertung` (2. Teilversuch) gestartet.

Je nachdem, ob auf dem eigenen oder dem anderen Calliope mini zuerst gedrückt wird, ist der Ablauf verschieden:

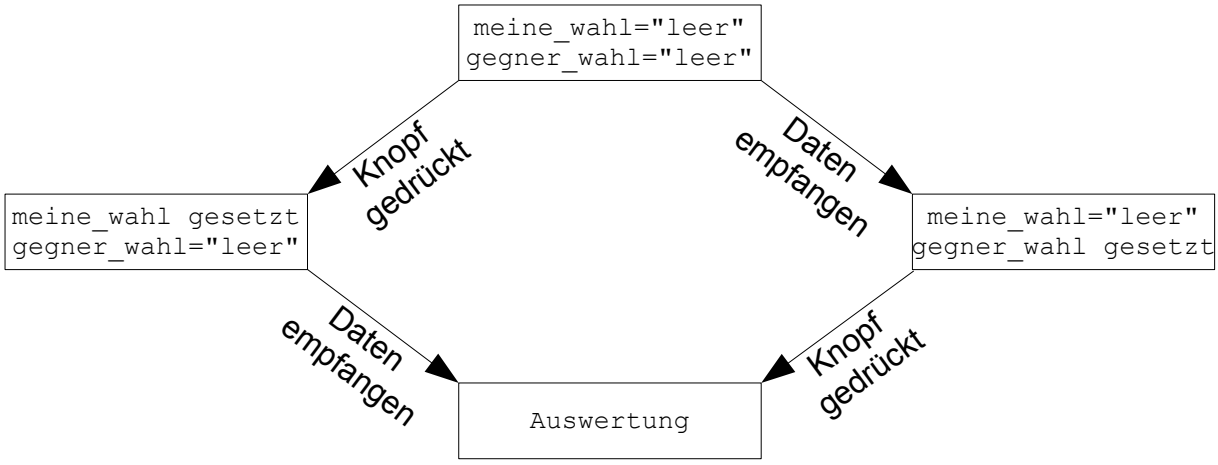

Abbildung 16: Ablauf: Schere, Stein, Papier

Eine mögliche Lösung ist (hier ist nur die Aktion für Knopf A abgebildet):

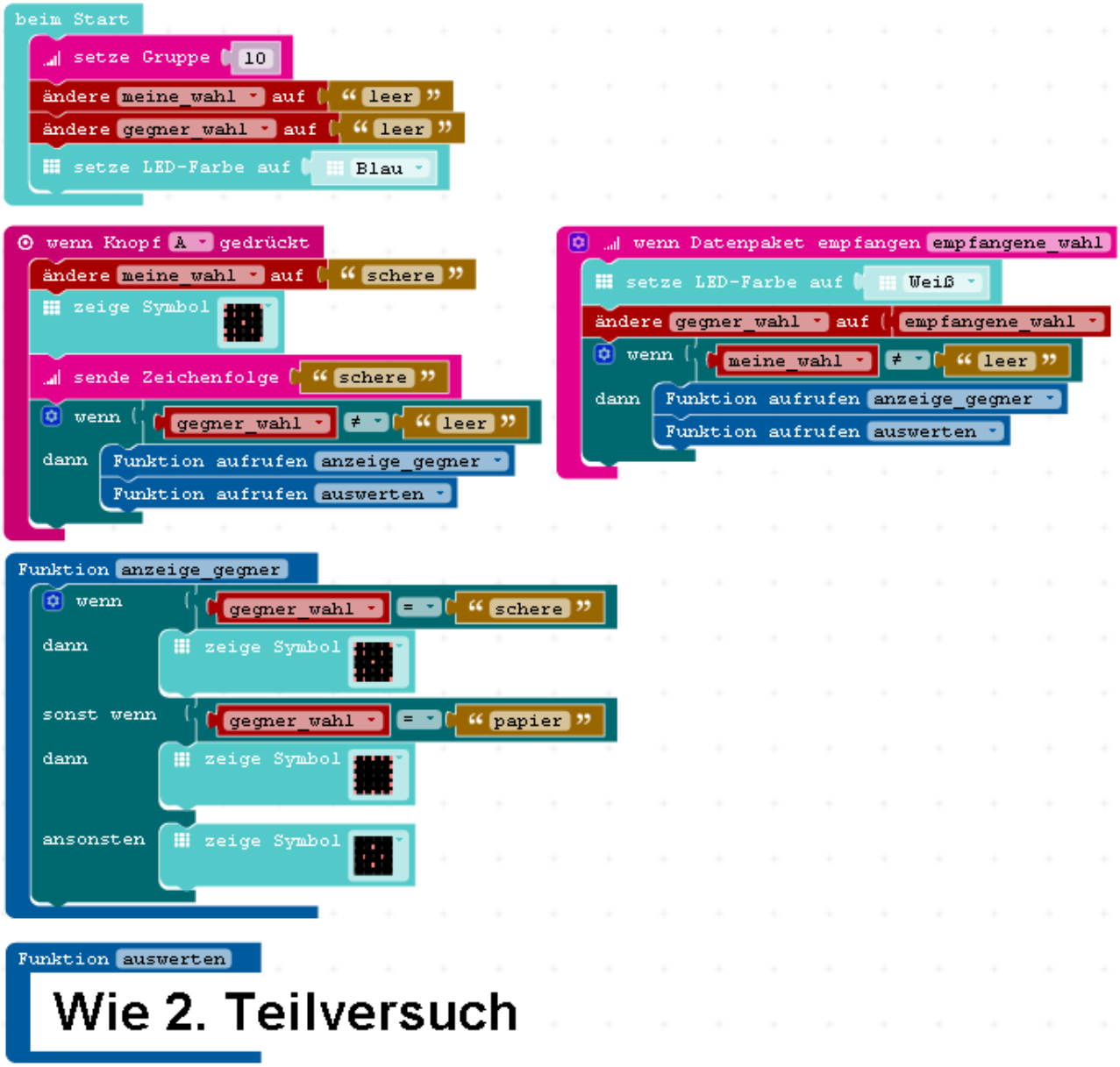

Abbildung 17: Schere, Stein, Papier

Einsatz im Unterricht

Der Lehrplan Informatik ([LPInf]) bietet mehrere Schnittstellen zur Integration der betrachteten Physical-Computing-Plattformen:

- Klasse 7, LB1 Computer verstehen
  - EVA-Prinzip
- Klasse 9/10, LB4 Algorithmen und Programme
  - Einfache Datentypen
  - Algorithmische Grundstrukturen
  - Modularisierung
- Klasse 9/10, WP1: Messen, Steuern, Regeln
  - Einfache Steuerungen, Sensoren und Aktoren
- Klasse 11/12, LB 8B: Technische Informatik
  - Zusätzlich: Regelkreise, Signalwandler, Schnittstellen

Besonders im Hinblick auf den Arduino Uno lohnt ein Blick zur Physik ([LPPhy]):

- Klasse 6, LB4: Elektrische Stromkreise
- Klasse 7, LB2: Stromstärke und Spannung, WP2: Elektrische Schaltungen
- Klasse 8, LB3: Elektrischer Widerstand
- Klasse 9, LB1: Grundlagen der Elektronik

Alle 3 betrachteten Plattformen ermöglichen nachhaltige – weil experimentelle – Erfahrungen mit allen oben genannten Prinzipien des Lehrplans Informatik. Die inhaltliche Einstiegshürde ist meist viel geringer als bei der Beschäftigung mit einer "ausgewachsenen" Programmiersprache. Der technische und organisatorische Aufwand für die Einbeziehung im Unterricht ist moderat, beim Arduino Uno etwas höher.

Beim Einsatz in Klasse 7 oder 8, dort z.B. lehrplangerechtfertigt über das EVA-Prinzip oder im Rahmen eines fächerübergreifenden Unterrichts Informatik-Physik, können die Konzepte der höheren Klassestufen propädeutisch und in vereinfachter Form integriert werden.

Bei der Erarbeitung algorithmischer Grundstukturen sehe ich die einfachen Ausgabemöglichkeiten als Hindernis an. Die serielle Konsole des Arduino Uno ist hier noch die leistungsfähigste Möglichkeit. Die Erarbeitung einer Schleife durch ein Beispiel wie

```
int i;
for (i=0; i<5; i++) {
     print(i);
}
```

ist zumindest umständlich.

Arduino Uno

Wegen der textuellen Programmierung ist der Arduino Uno eher für höhere Klassenstufen oder interessierte Schülergruppen (GTA o.ä.) geeignet, wenn ein Vorverständnis algorithmischer

Grundstukturen vorhanden ist. In jedem Fall benötigt man Beispiele und Lösungshilfen mit feinen Schwierigkeitsabstufungen.

Ein weiters besonderes Merkmal ist die große Rolle der Außenbeschaltung. Auch hier sind Vorkenntnisse und ein vergrößerter Betreuungsaufwand – besonders bei der Fehlersuche – erforderlich. Die Arbeit auf der Experimentierplatine nutzt die Flexibilität durch die einfache Integration zusätzlicher Bauteile und -gruppen aus. Ich habe den Umgang jedoch als fehleranfällig wahrgenommen. Sofern die Schüler hier keine Übung haben, wäre eine Außenbeschaltung mit robusteren Komponenten – wie im Physikunterricht bei einfachen Stromkreisen oft benutzt – geeignet.

BBC micro:bit und Calliope mini

Diese beiden Plattformen sind durch die übliche grafische Programmierung und den kompakten technischen Aufbau leichter und mit weniger Vorbedingungen einsetzbar. Durch die Umschaltung auf textuelle Programmierung (JavaScript) und beim BBC micro:bit die zusätzliche Alternative Python kann hier der Übergang zur fortgeschrittenen Form gemacht werden.

Versuch einer Bewertung

Zusammenfassend stelle ich die Eignung nach verschieden Lernvorrausetzungen und -zielen tabellarisch dar. Das ist natürlich subjektiv und kann durch entsprechende Unterrichtsgestaltung verbessert werden. Auch habe ich Erweiterungen (Hard- und Software) der Plattformen durch Dritte nicht mit einbezogen, da hier oft weniger Projekte und Materialien zur Nachnutzung zu finden sind oder die Erweiterungen zusätzliche Kosten verursachen.

	Arduino Uno	Calliope mini	BBC micro:bit
			
Kosten		✔	✔
Aufwand		✔	✔
Elektrische Schaltungen	✔		
Grafische Programmierung		✔	✔
Textuelle Programmierung	✔		✔
Technische Informatik	✔	✔	✔
Regulärer Unterricht		✔	✔
Klassenstufe 5-7		✔	✔
GTA	✔	✔	✔
Projekt oder Fächerverbindend	✔	✔	✔

Literaturverzeichnis

Atmel01: Atmel Corp., ATmega328P Datasheet, 2015,
http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

Cal01: Calliope gGmbH, Calliope mini: Produktdatenblatt Einzelset, 2019,
https://calliope.cc/content/2_idee/2_ueber-mini/produktblatt_einzelset_final.pdf

PEul03: Colin Hughes, Project Euler - Problem 3 - Largest Prime Factor, ,
https://projecteuler.net/problem=3

BBC01: Micro:Bit Educational Foundation, Micro:Bit Harware description, 2019,
https://tech.microbit.org/hardware/

LPInf: Sächsisches Staatsministerium für Kultus, Lehrplan Gymnasium Informatik, 2018

LPPhy: Sächsisches Staatsministerium für Kultus und Sport, Lehrplan Gymnasium Physik, 2011

Bildquellen

Abbildung 1, 3, 5, Tabelle: https://commons.wikimedia.org/wiki/File:Arduino-uno.jpg,
Imik tech [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], bearbeitet

Abbildung 3, 5: https://commons.wikimedia.org/wiki/File:Fritzing-steckplatine.jpg,
fritzing.org [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], bearbeitet

Tabelle: https://commons.wikimedia.org/wiki/File:BBC_micro_bit.jpg, Les Pounder from Blackpool, UK [CC BY-SA 2.0 (https://creativecommons.org/licenses/by-sa/2.0)]

https://commons.wikimedia.org/wiki/File:Calliope_mini_weiss_JoernAlraun.jpg, Jørn Alraun [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)]