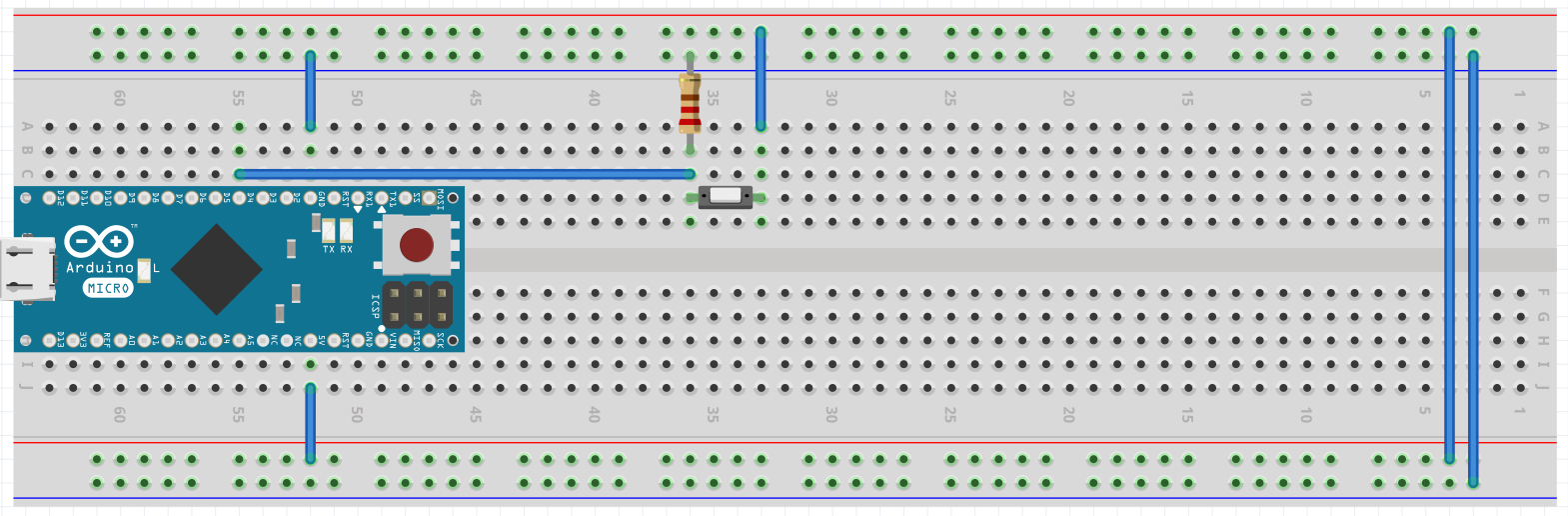
Übungsblatt 4 – Arduino

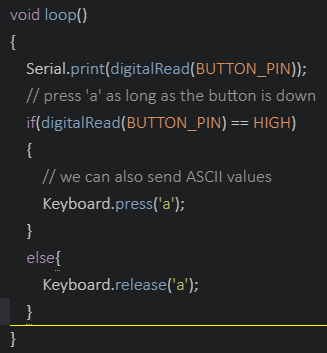
Für Videos/ Code siehe Ordner „Anhang“ und „Code“

# 4.1

Button wurde an den im Code vorgegebenen Pin (Pin 4) als Input angeschlossen (Wichtig: Pull-down Widerstand nicht vergessen!). Wenn Button gedrückt wird, wird quasi die Taste „a“ der Tastatur gedrückt.



### Anmerkung:

Die Eingabe durch den vorgegebenen Code fühlt sich „unnatürlich“ durch den 1000ms Delay an. Durch folgende Änderung ist die Eingabe „wie gewohnt“.

# 4.2

### Idee

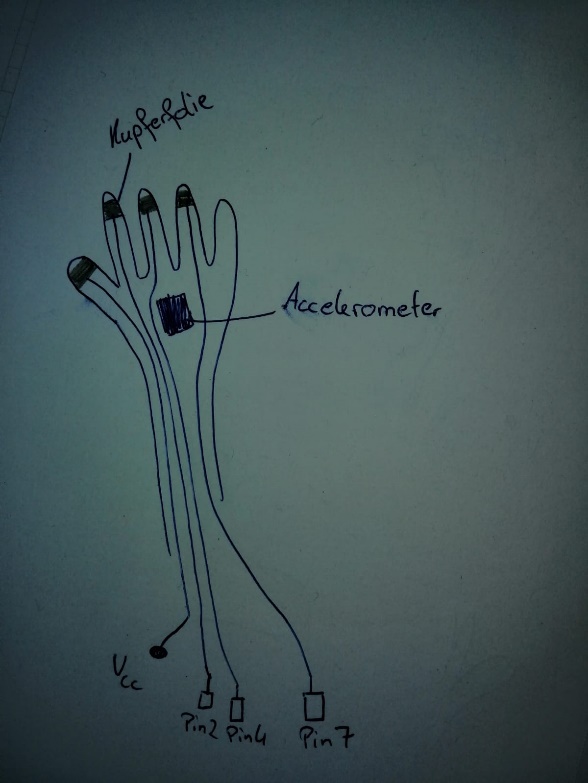
Für das Spiel Minecraft soll ein eigenes Eingabegerät entwickelt werden, welches die wichtigsten Ingame-Funktionen umfasst. Dazu zählen:

* Laufen (Taste ‚w‘)
* Umsehen (Maus)
* Abbauen/ Angreifen (linke Maustaste)
* Block setzen (Rechte Maustaste)

Ich habe mich für eine Art „Gaming-Handschuh“ entschieden, der die genannten Funktionen abdeckt.

|  |  |  |  |
| --- | --- | --- | --- |
| **Funktion** | **Eingabe normal** | **Neue Eingabe per** | **Am Handschuh** |
| Laufen | w-Taste | „Button“ | Berührung von Zeigefinger und Daumen |
| Umsehen | Maus | 3-Axis Accelerometer | Handschuhh bewegen |
| Abbauen/ Angreifen | Linke Maustaste | „Button“ | Berührung von Mittelfinger und Daumen |
| Block setzen | Rechte Maustaste | „Button“ | Berührung von Ringfinger und Daumen |

### Umsetzung

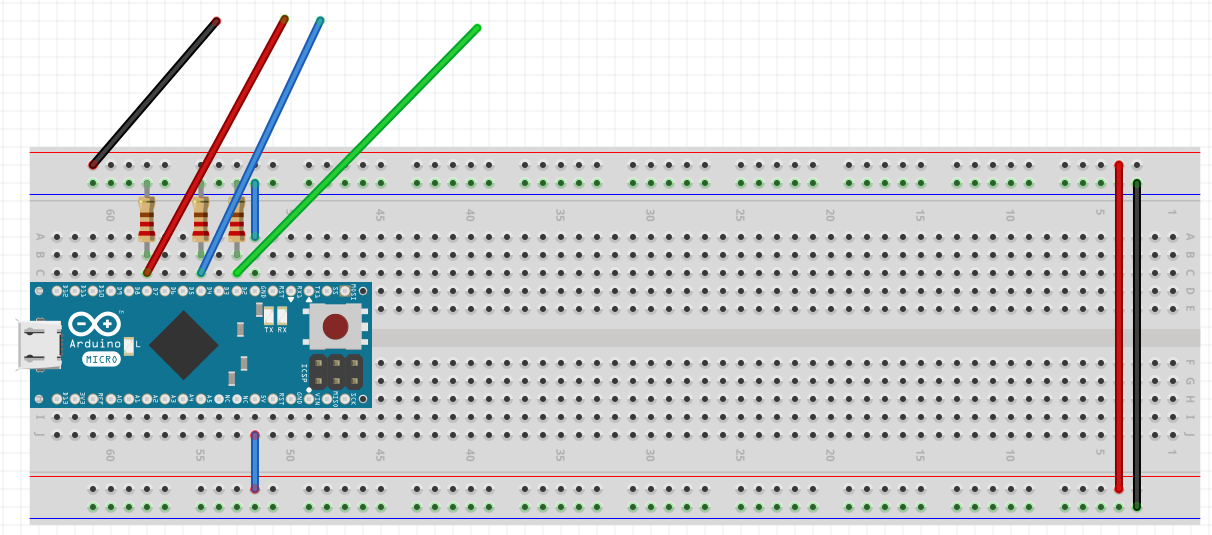
1. **Skizzieren/ Visualisieren meines Plans**

* Der Daumen ist mit der Spannungsquelle verbunden. Sobald einer der Finger mit dem Daumen in Berührung kommt, ist fließt Strom an den jeweiligen Pin des Arduinos. Der Zeigefinger ist soll also an Pin2 anliegen und soll bei Kontakt mit dem Daumen das Event „Laufen“ bzw. die w-Taste simulieren (gleiches Prinzip für die anderen zwei Finger). Um den Anschluss/ die Funktion des Accelerometers kümmere ich mich später, ich will zunächst erstmals die Buttons zum laufen kriegen.

1. **Löten**

* Als nächstes habe ich angefangen, die benötigten Drähte/ Kabel/ Kupferfolien zusammenzulöten, und diese dann auch gleich am Handschuh angebracht (dabei habe ich auch gleich geschaut, dass die verwendeten Litzen lange genug sind, dass man später auch etwas Freiraum in der Armbewegung hat/ falls immernoch zu kurz, kann man einfach noch eine Verlängerung zusammenlöten). Für das Accelerometer habe ich auch gleich drei Stecker verlängert, die ich später für die Daten der x, y, und z-Achse benötige

1. **Schaltung für die Buttons planen und umsetzen**

* Jetzt habe ich versucht, die Schaltung/ Anschlüsse am Steckbrett umzusetzen. Das ging relativ schnell, da die Finger ja jetzt eigentlich einfach nur als Schalter angesehen werden können, die bei Berührung (mit der Spannungsquelle, also dem Daumen) zu einem gewissen Pin Strom leiten sollen, ansonsten nicht. Wichtig war hier aber wieder, die Pull-Down Widerstände nicht zu vergessen, dass kein „schwebender“ Zustand an den Pins vorliegt, wenn kein Strom anliegt, sondern diese klar definiert 0V empfangen.
* Zeigefinger (Grüner Draht) soll an Pin 2, Mittelfinger (Blauer Draht) an Pin 4, Ringfinger (Roter Draht) an Pin 7, Daumen (schwarzer Draht) an Vcc

1. **Code für die Fingerbuttons**

* Nach kurzem Rumprobieren habe ich mich für drei

Aufeinanderfolgende if-else-Abfragen (und nicht

ein zusammenhängendes if-else) entschieden, weil

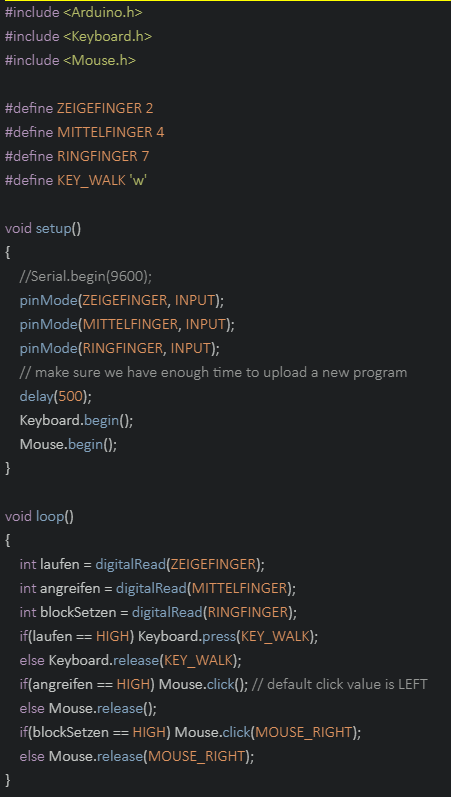
auf diese Weise mehrere Aktionen gleichzeitig

durchgeführt werden können (z.B. nach vorne

laufen und schlagen).

* Ergebnis sehe Video „Fingerbuttons –

Zwischenergebnis.mp4“

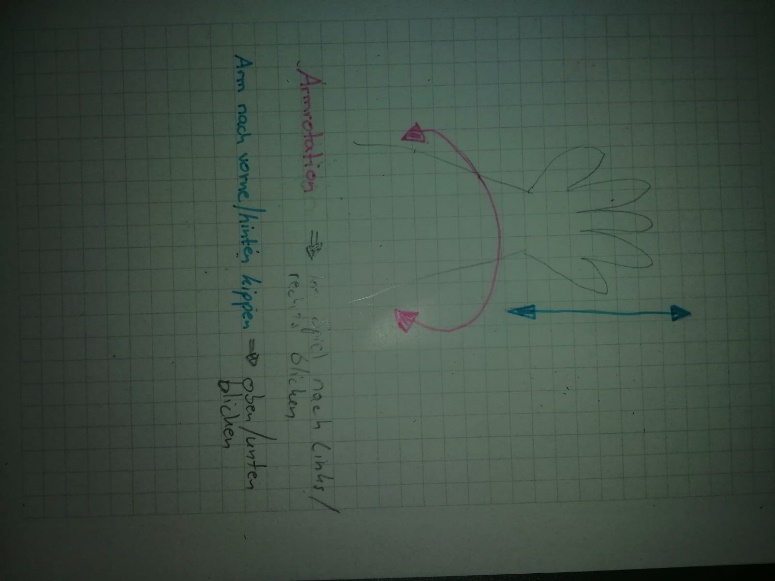
****

1. Accelerometer einbauen

* Für mein Eingabegerät will ich, dass…

… der Spieler nach links/ rechts blickt, wenn der Arm nach rechts/ links rotiert wird

… der Spieler nach oben/ unten blick, wenn der Arm nach vorne/ hinten gekippt wird

* 

5. Accelerometer Notizen

* damit kann man die Beschleunigung auf die 3 Raumachsen (x,y,z) messen 🡪 Ausrichtung steht auf dem Acc mit drauf.
* in unserem Fall 3-5V Inputspannung akzeptabel
* 3 Pins für jede Achse als Output
* Wenn z-Achse nach oben schaut, wirkt die Gravitation auch nur auf die Z-Achse
* Resultat der eingehenden Spannung:
* 416 entspricht ca. 2V
* 340 entspricht ca. 1.65V
* Wenn y-Achse nach oben schaut:
* Wenn x-Achse nach oben schaut:
* Wenn x, y- oder z-Achse nach unten schauen, sind Werte von ca. 265 (1.3V) zu erwarten
* Das sind allerdings nur die Werte, die wir bekommen, wenn sich der Sensor in Ruhe befindet. Bei Bewegung wirkt dementsprechend mehr oder weniger Kraft auf die Achsen
* Im freien Fall wird für einen kurzen Moment keine Beschleunigung an den 3 Achsen gemessen/ bzw. Werte von ca. 335
* Wenn ich Sensor entlang der x-Achse um 45 Grad kippe, erhalte ich für y- und z- nahezu identische Werte, da die Kraft auf beide Achsen gleich ist

Für mein Eingabegerät will ich, dass…

… der Spieler sich nach links/ rechts dreht, wenn der Arm nach rechts/ links rotiert wird

… der Spieler nach oben/ unten schaut, wenn der Arm nach vorne/ hinten kippt