
DOKUMENTATIONPONG

Modul 242 Mikroprozessoranwendung realisieren

21.04.2023

Alexander Siegenthaler

Inhaltsverzeichnis

1. Vorwort.....	3
2. Informieren.....	3
2.1. Ausgangslage.....	3
2.2. Vorgaben.....	4
3. Planen.....	6
3.1. Pong.....	6
3.2. Roomba.....	7
4. Entscheiden.....	9
5. Flussdiagramm.....	10
5.1. Verwendete Sensoren/Aktoren.....	11
6. Realisieren.....	11
6.1. Vorarbeit.....	11
6.2. Umsetzung.....	11
7. Kontrollieren.....	13
7.1. Testprotokoll.....	13
8. Auswerten.....	13
8.1. Reflexion.....	13
9. Abbildungsverzeichnis.....	14
10. Tabellenverzeichnis.....	14
11. Quellenverzeichnis.....	14
12. Anhangsverzeichnis.....	15

1. Vorwort

Die Dokumentation gehört zur LB des Moduls 242 von Alexander Siegenthaler und dokumentiert die Umsetzung der LB. Durch die ganze Arbeit hinweg wurde mit der Projektplanungsmethode IPERKA gearbeitet.

2. Informieren

2.1. Ausgangslage

Ziel der LB ist es im Rahmen von 2 Tagen eine Mikroprozessoranwendung mithilfe des Halocodes von Mblock zu realisieren und dokumentieren. Es besteht die Möglichkeit diverse Sensoren innerhalb dieses Projektes zu verwenden

2.2. Vorgaben

Für die LB sind folgende Kriterien zu beachten:

Projekttitel	Titel fehlt 0 Punkte	Falscher Titel 1 Punkte	Minimaler, fehlerhafter Titel 2 Punkte	Titel vorhanden 3 Punkte	Angepasster Titel 4 Punkte	Aussagekräftiger Titel 5 Punkte
Autor, Datum, Projektparameter	keine Beschreibung 0 Punkte	Minimale, fehlerhafte Angaben 1 Punkte	minimale, unvollständige Angaben 2 Punkte	Minimale Angaben 3 Punkte	Angaben korrekt 4 Punkte	Korrekte, vollständige Angaben 5 Punkte
Kurzbeschreibung	Kurzbeschreibung fehlt 0 Punkte	Minimale, fehlerhafte Kurzbeschreibung 1 Punkte	Minimale unvollständige Kurzbeschreibung 2 Punkte	Kurzbeschreibung vorhanden 3 Punkte	Korrekte Kurzbeschreibung 4 Punkte	Korrekte, vollständige Beschreibung 5 Punkte
Video/Fotodokumentation	Foto/Videodokumentation fehlt 0 Punkte	Minimalste Foto/Videodokumentation 1 Punkte	Minimale, Foto/Videodokumentation 2 Punkte	Foto/Videodokumentation vorhanden 3 Punkte	Foto/Videodokumentation vorhanden 4 Punkte	Korrekte, vollständige Videodokumentation 5 Punkte
Inline-Kommentare Code	Keine Inline Kommentare 0 Punkte	Minimale, fehlerhafte Kommentare 1 Punkte	Minimale, unvollständige Kommentare 2 Punkte	Kommentare vorhanden 3 Punkte	Wesentliche Kommentare sind korrekt und vollständig 4 Punkte	Korrekte, vollständige Kommentare 5 Punkte
Programmlogik (Flusslaufdiagramm)	Beschreibung der Programmlogik fehlt 0 Punkte	Minimale, fehlerhafte Beschreibung der Programmlogik 1 Punkte	Minimale, unvollständige Beschreibung der Programmlogik 2 Punkte	Minimale Beschreibung der Programmlogik 3 Punkte	Programmlogik der Kernfunktionen vorhanden 4 Punkte	Programmlogik komplett vorhanden 5 Punkte

Abbildung 1: Bewertungskriterien a

Projektidee	Kein Projekt (-idee) 0 Punkte	Projektidee kopiert 1 Punkte	Minimale Anpassung einer Projektidee 2 Punkte	Angepasste Projektidee 3 Punkte	Angepasste Projektidee mit eigenen Ergänzungen 4 Punkte	Eigene Projektidee 5 Punkte
Projektumsetzung	Kein Projekt 0 Punkte	Projekt ansatzweise umgesetzt 1 Punkte	Einige Punkte des Projektes sind umgesetzt 2 Punkte	Wichtigste Punkte des Projektes sind umgesetzt 3 Punkte	Fast alle Punkte des Projektes sind umgesetzt 4 Punkte	Alle Punkte des Projektes sind umgesetzt. 5 Punkte
Programm-Code	Kein Programmcode 0 Punkte	Minimaler, fehlerhafter Code 1 Punkte	Minimaler, unvollständiger Code 2 Punkte	Wichtigster Programmcode ist vorhanden und funktioniert 3 Punkte	Programmcode ist vollständig 4 Punkte	Programm-Code ist vollständig und optimiert 5 Punkte
Programm-Komplexität 1 Variablen/Listen	Kein Einsatz von Variablen und Listen 0 Punkte	Minimaler, fehlerhafter Einsatz von Variablen und Listen 1 Punkte	Minimaler, unvollständiger Einsatz von Variablen und Listen 2 Punkte	Variablen und Listen werden punktuell und korrekt eingesetzt 3 Punkte	Variablen und Listen werden korrekt eingesetzt 4 Punkte	Variablen und Listen werden optimal eingesetzt 5 Punkte
Programm-Komplexität 2 Funktionen	Kein Einsatz von Funktionen 0 Punkte	Minimaler, fehlerhafter Einsatz von Funktionen 1 Punkte	Minimaler, unvollständiger Einsatz von Funktionen 2 Punkte	Funktionen werden punktuell und korrekt eingesetzt 3 Punkte	Funktionen werden korrekt eingesetzt 4 Punkte	Funktionen werden optimal eingesetzt 5 Punkte
Programm-Komplexität 3 mbuild/Extension	Keine Einsatz von mbuild Elementen oder Extension 0 Punkte	Minimaler Einsatz von mbuid Elementen oder Extension 1 Punkte	Minimaler, unvollständiger Einsatz von mbuid Elementen oder Extension 2 Punkte	mbuid oder Extension werden punktuell und korrekt eingesetzt 3 Punkte	mbuid oder Extension werden korrekt eingesetzt 4 Punkte	mbuid oder Extension werden optimal eingesetzt 5 Punkte
Debugging/Test	Keine Angaben zum Debugging/Testfälle 0 Punkte	Minimale, fehlerhafte Angaben zum Debugging/Testfälle 1 Punkte	Minimale, unvollständige Angaben zum Debugging/Testfälle 2 Punkte	Angaben zum Debugging/Testfälle sind vorhanden 3 Punkte	Debugging/Testfälle sind vollständig vorhanden 4 Punkte	Debugging/Test ist vollständig und automatisiert 5 Punkte

Abbildung 2: Bewertungskriterien b

3. Planen

Innerhalb der Planung der LB haben sich folgende Projektideen entwickelt:

3.1. Pong

Beschreibung

Die Hardware besteht aus dem Halocode, Joysticks, LED-Driver, 8x16 blue LED Matrizen und einem Speaker. Die Idee ist es ein Spiel zu entwickeln welches sich wie «Pong» verhält. Das Spielfeld verläuft über die beiden LED-Matrizen und beide Mitspieler steuern mit einem eigenen Joystick einen Spieler welcher sich am unteren Ende des jeweiligen Screens befindet. Das Ziel ist es einen Ball am gegnerischen Spieler vorbeizuschlagen. Mithilfe zweier LED-Driver wird der aktuelle Punktestand des jeweiligen Spielers dargestellt und bei einem Punkt wird ein Sound abgespielt.

Testkonzept

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
1	Testen der Joysticks: Das Ziel dieses Tests ist es, zu überprüfen, ob die Bewegungen der Spieler auf dem Bildschirm korrekt auf die Joysticks reagieren und dass beide Joysticks ordnungsgemäß funktionieren.	Blackbox / Positivtest	Unit Test / optisch	Wenn der Joystick nach rechts gedrückt wird, bewegt sich der Spieler bis an den rechten Rand des Spielfeldes.

Tabelle 1: Testfall Nr. 1

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
2	Testen des LED-Displays: Das Ziel dieses Tests ist es, zu überprüfen, ob das LED-Display korrekt funktioniert und die Spielfläche korrekt dargestellt wird.	Blackbox / Positivtest	Unit Test / optisch	Wenn das Spiel gestartet wird, erscheinen auf beiden Displays jeweils ein 1x5 Pixel grosser Spieler.

Tabelle 2: Testfall-Nr. 2

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
3	Testen des Sound-Effekts: Das Ziel dieses Tests ist es, zu überprüfen, ob der Sound-Effekt korrekt abgespielt wird.	Blackbox / Positivtest	Unit Test / akustisch	Wenn der Spieler einen Punkt erzielt, wird ein Sound-Effekt abgespielt.

Tabelle 3: Testfall Nr. 3

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
4	Testen des Punktestands: Das Ziel dieses Tests ist es, zu überprüfen, ob der Punktestand korrekt auf beiden LED-Stripes dargestellt wird.	Blackbox / Positivtest	Unit Test / optisch	Wenn Spieler 1 den Ball an Spieler 2 vorbeischießt und er einen Punkt erzielt, leuchtet bei Spieler 1 exakt 1 LED.

Tabelle 4: Testfall Nr. 4

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
5	Testen der Kollisionserkennung: Das Ziel dieses Tests ist es, zu überprüfen, ob das Spiel korrekt auf Kollisionen zwischen dem Ball und den Spielern reagiert.	Blackbox / Positivtest	Unit Test / optisch	Wenn ein Spieler den Ball zentral berührt, bewegt sich der Ball vertikal in die Richtung des gegenüberliegenden Spielers.

Tabelle 5: Testfall Nr. 5

3.2. Roomba

Beschreibung

Die Hardware besteht aus dem Halocode, Akkus, DC Motor Driver, Ultrasonic Sensor und Servo-Driver. Die Idee ist es mithilfe des Motor Driver und des Servo-Driver ein Gefährt zu bauen welches durch den Ultrasonic Sensors Hindernisse erkennt, diese durch das WLAN-Modul vom Halocode als CloudVariablen an ein Device sendet und diese auf dem Device innerhalb des Mblock Editors als Sprites dargestellt werden.

Testkonzept

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
6	Testen der Hinderniserkennung: Das Ziel dieses Tests ist es, zu überprüfen, ob das Gerät Hindernisse korrekt erkennt und vermeidet.	Blackbox / Positivtest	Unit Test / optisch	Wenn das Gerät gestartet wird und es 5cm vor einem Hindernis ist stoppt es.

Tabelle 6: Testfall Nr. 6

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
7	Testen der Cloud-Kommunikation: Das Ziel dieses Tests ist es, zu überprüfen, ob die Daten korrekt an das Device gesendet und korrekt verarbeitet werden.	Blackbox / Positivtest	Unit Test / optisch	Wenn in der mblock IDE die Space Taste gedrückt wird, startet das Gerät mit der Hinderniserkennung.

Tabelle 7: Testfall-Nr. 7

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
8	Testen der WLAN-Verbindung: Das Ziel dieses Tests ist es, zu überprüfen, ob die WLAN-Verbindung ordnungsgemäß funktioniert und die Daten erfolgreich an das Device gesendet werden.	Blackbox / Positivtest	Unit Test / optisch	Wenn in der mblock IDE die Space Taste gedrückt wird, blinkt der Halocode 3 mal grün.

Tabelle 8: Testfall Nr. 8

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
9	Testen der Bewegung: Ziel dieses Tests ist es, zu überprüfen, ob sich das Gerät korrekt bewegt.	Blackbox / Positivtest	Unit Test / optisch	Wenn das Gerät auf ein Hindernis stösst, hält es an und bewegt sich nach links.

Tabelle 9: Testfall Nr. 9

Nr.	Zieldefinition	Testmethode	Testumgebung	Testfall
10	Testen der Geschwindigkeit: Das Ziel dieses Tests ist es, zu überprüfen, ob das Gerät mit der richtigen Geschwindigkeit fährt, um Hindernisse effektiv zu vermeiden.	Blackbox / Positivtest	Unit Test / optisch	Wenn das Gerät auf ein Hindernis stösst, dass sich auf der linken Seite befindet, hält es an und bewegt sich nach links. Anschliessend fährt es zurück und bewegt sich nach rechts.

Tabelle 10: Testfall Nr. 10

4. Entscheiden

Im Verlauf der Analyse, sind mir Probleme mit der Cloud-Verbindung aufgefallen. Da ich bei der Projektidee Roomba auf eine stabile Datenübertragung zwischen Roomba und Steuerungs-Device angewiesen bin, habe ich mich für die Projektidee Pong entschieden.

5. Flussdiagramm

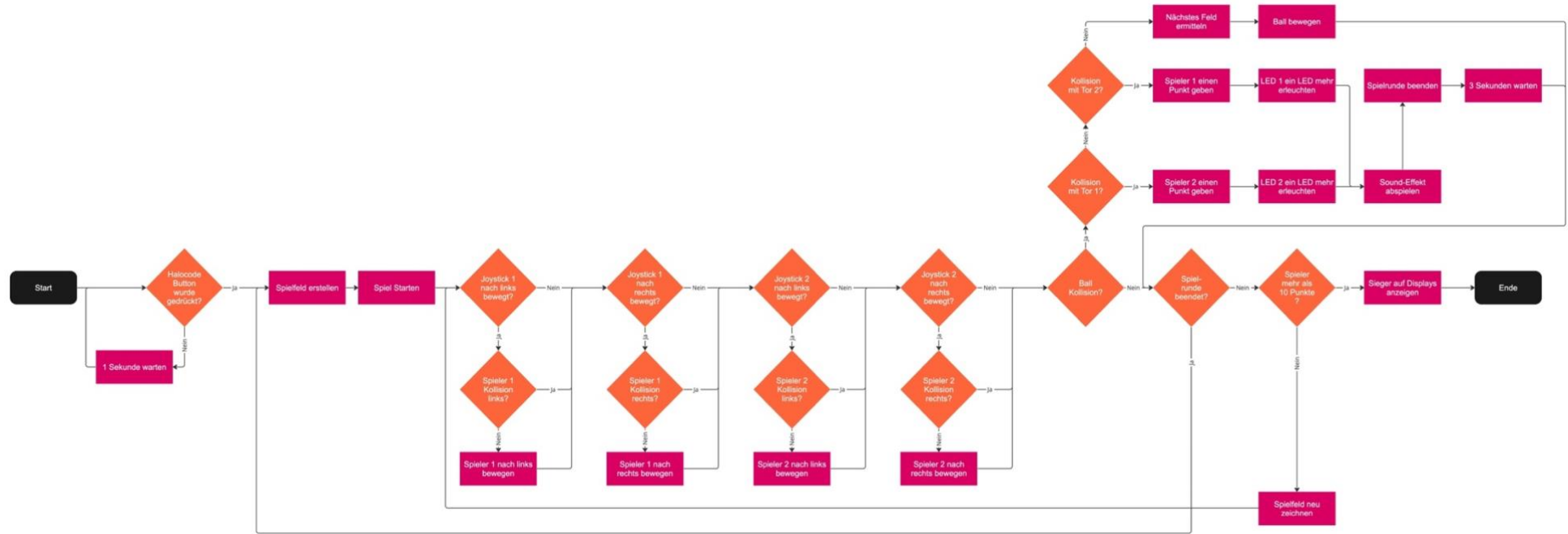


Abbildung 3: Flussdiagramm Pong

5.1. Verwendete Sensoren/Aktoren

Folgende Sensoren/Aktoren werden im Verlauf der LB verwendet:

Namen	Beschreibung	Stück
Halocode	Der Halocode enthält diverse Sensoren/Aktoren. Innerhalb von diesem Projekt wird nur der Zentrale Button verwendet.	1
Joystick	Der Joystick wird für die Steuerung des jeweiligen Spielers verwendet.	2
LED-Driver	Der LED-Driver wird für die Darstellung der Punkte des jeweiligen Spielers verwendet.	2
LED-Strip	Auf dem LED-Strip wird die effektive Punktzahl des jeweiligen Spielers dargestellt.	2
8x16 blue LED Matrice	Die 8x16 blue LED Matrice wird für die Darstellung des Spielfeldes verwendet (Spieler/Ball).	2
Speaker	Der Speaker wird verwendet, um ein Punkt zu signalisieren.	1

6. Realisieren

6.1. Vorarbeit

Für dieses Projekt wurden keine nennenswerten Vorarbeiten geleistet.

Es wurden einzig zu den beiden Projektideen, Roomba und Pong, stark vereinfachte POC's erstellt, welche hauptsächlich die Hardware und die Kommunikation mit der Cloud von Mblock betreffen.

Ausserdem habe ich das Uploaden von Python Scripts mit «Halocode-Tools» von Bailuk getestet.

6.2. Umsetzung

Hardware

Es wurde eine zweckdienliche Basis gebaut, auf welcher man die einzelnen Komponenten befestigen kann. Die einzelnen Komponenten wurden alle mit Grove-Kabel verbunden. Hierbei wurde darauf geachtet, Komponenten mit einem höheren Energieverbrauch näher am Halocode zu montieren (Halocode -> LED-Matrix ->LED-Driver->Joystick->Speaker).

Software

Da Mblock Halocode eine Python API liefert, habe ich mich für die Entwicklung in Python entschieden. Leider ist es nur möglich ein einzelnes Python Script in den Halocode zu laden, daher sind sämtliche Klassen und Funktionen in einem einzelnen Script definiert.

Klassen

Field: Definiert die Grösse des Spielfeldes und instanziert die einzelnen Komponenten (Player, Ball...). Ausserdem enthält es die Funktionen, um den Spieler und den Ball zu bewegen, er kontrolliert die gesamte Kollisionslogik ausserdem ist es für die Darstellung zuständig

Player: Der Spieler definiert die Spielergrösse sowie die Position auf dem Spielfeld als auch die Funktion sich selbst zu bewegen.

Ball: Der Ball definiert die Position von sich selbst auf dem Spielfeld und ist für die Kontrolle der Kollision mit einem Spieler, des Borders und der Wände zuständig.

Wall: Die Wand definiert die Position von sich selbst auf dem Spielfeld. Es stellt ein Hindernis auf dem Spielfeld dar.

Border: Der Border definiert die Position von sich selbst auf dem Spielfeld. Er stellt die Grenze zwischen den beiden Spielfeldern dar.

Goal: Das Goal definiert die Position von sich selbst auf dem Spielfeld. Wenn der Ball mit dem Goal kollidiert, erhält der gegenüberliegende Spieler einen Punkt.

Funktionen

startup(): Ein Endlos-Loop der auf einen Tastendruck wartet um das Spiel zu starten.

game_running(): Ein Endlos-Loop der ein gesamtes Spiel darstellen soll.

round_running(): Ein Endlos-Loop der eine einzelne Runde darstellen soll.

7. Kontrollieren

7.1. Testprotokoll

Nr.	Testfall	Durchführungs-Datum	Verantwortliche Person	Ergebnis
1	Wenn der Joystick nach rechts gedrückt wird, bewegt sich der Spieler bis an den rechten Rand des Spielfeldes.	21.04.23	Alexander Siegenthaler	Negativ
2	Wenn das Spiel gestartet wird, erscheinen auf beiden Displays jeweils ein 1x5 Pixel grosser Spieler.	21.04.23	Alexander Siegenthaler	Negativ
3	Wenn der Spieler einen Punkt erzielt, wird ein Sound-Effekt abgespielt.	21.04.23	Alexander Siegenthaler	Negativ
4	Wenn Spieler 1 den Ball an Spieler 2 vorbeischiess und er einen Punkt erzielt, leuchtet bei Spieler 1 exakt 1 LED.	21.04.23	Alexander Siegenthaler	Negativ
5	Wenn ein Spieler den Ball zentral berührt, bewegt sich der Ball vertikal in die Richtung des gegenüberliegenden Spielers.	21.04.23	Alexander Siegenthaler	Negativ

Tabelle 11: Testprotokoll

8. Auswerten

8.1. Reflexion

Leider habe ich mich innerhalb dieses Projektes etwas überschätzt. Aber da ich noch keinerlei Kontakt mit Python hatte und das Projekt aufwändig war, ist dies Gerechtfertigt. Im aktuellen Stand fehlen folgende Funktionen:

- LED-Stripes
- Speaker
- Die korrekte Darstellung des Spielers
- Der Wechsel des Balls von einem Screen zum andern
- Die Unit-Tests der einzelnen Tests

9. Abbildungsverzeichnis

Abbildung 1: Bewertungskriterien a	4
Abbildung 2: Bewertungskriterien b	5
Abbildung 3: Flussdiagramm Pong	10

10. Tabellenverzeichnis

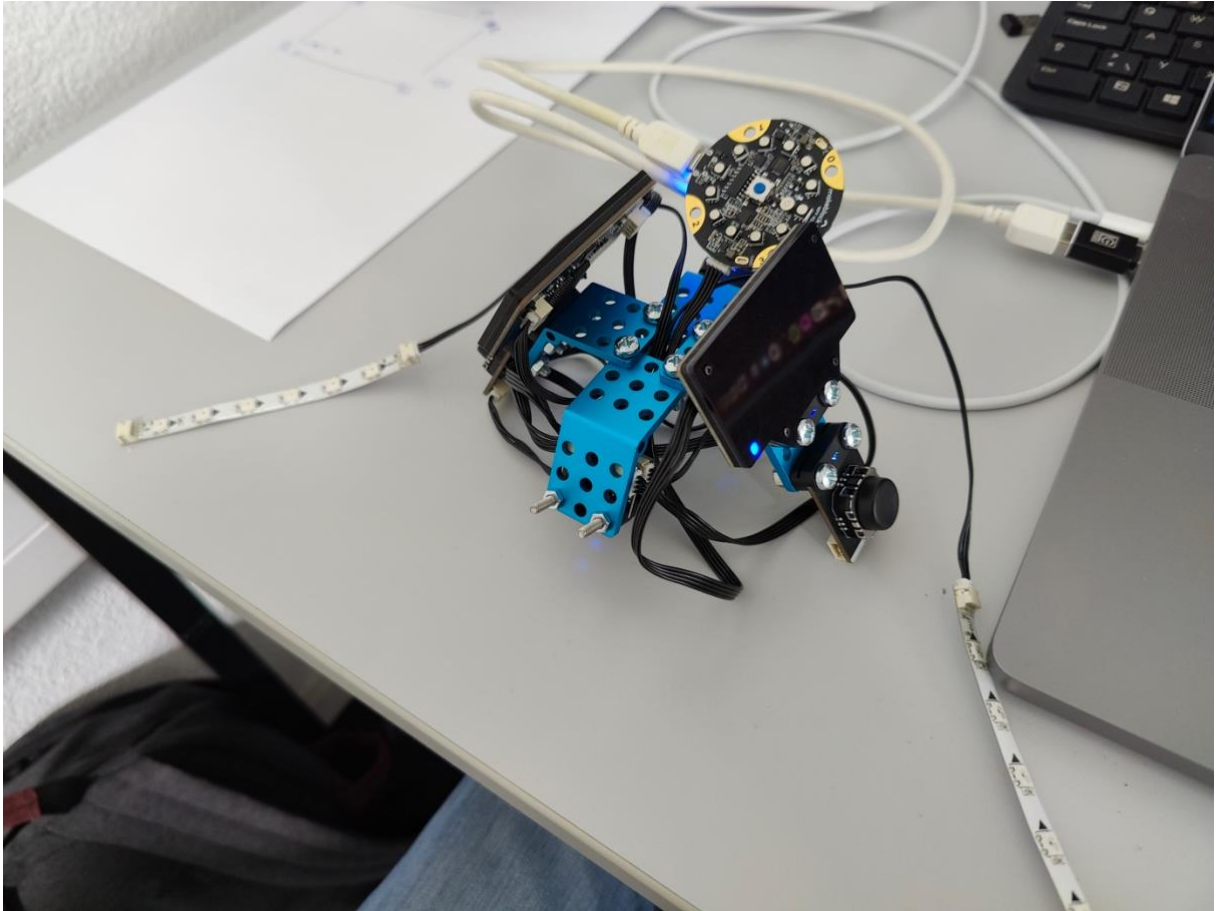
Tabelle 1: Testfall Nr. 1	6
Tabelle 2: Testfall-Nr. 2	6
Tabelle 3: Testfall Nr. 3	7
Tabelle 4: Testfall Nr. 4	7
Tabelle 5: Testfall Nr. 5	7
Tabelle 6: Testfall Nr. 6	8
Tabelle 7: Testfall-Nr. 7	8
Tabelle 8: Testfall Nr. 8	8
Tabelle 9: Testfall Nr. 9	8
Tabelle 10: Testfall Nr. 10	9
Tabelle 11: Testprotokoll	13

11. Quellenverzeichnis

- <https://chat.openai.com/>
- <https://stackoverflow.com>
- <https://www.mblock.cc>
- <https://www.w3schools.com>
- <https://github.com/bailuk/halocode-tools>

12. Anhangsverzeichnis

- Foto



- Source Code: <https://github.com/alex92ch/242>