DokumentationPong

Modul 242 Mikroprozessoranwendung realisieren

21.04.2023

Alexander Siegenthaler

Inhaltsverzeichnis

[1. Vorwort 3](#_Toc132984300)

[2. Informieren 3](#_Toc132984301)

[2.1. Ausgangslage 3](#_Toc132984302)

[2.2. Vorgaben 4](#_Toc132984303)

[3. Planen 6](#_Toc132984304)

[3.1. Pong 6](#_Toc132984305)

[3.2. Roomba 7](#_Toc132984306)

[4. Entscheiden 9](#_Toc132984307)

[5. Flussdiagramm 10](#_Toc132984308)

[5.1. Verwendete Sensoren/Aktoren 11](#_Toc132984309)

[6. Realisieren 11](#_Toc132984310)

[6.1. Vorarbeit 11](#_Toc132984311)

[6.2. Umsetzung 11](#_Toc132984312)

[7. Kontrollieren 13](#_Toc132984313)

[7.1. Testprotokoll 13](#_Toc132984314)

[8. Auswerten 13](#_Toc132984315)

[8.1. Reflexion 13](#_Toc132984316)

[9. Abbildungsverzeichnis 14](#_Toc132984317)

[10. Tabellenverzeichnis 14](#_Toc132984318)

[11. Quellenverzeichnis 14](#_Toc132984319)

[12. Anhangsverzeichnis 14](#_Toc132984320)

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.

1. Informieren

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

Vorgaben

Für die LB sind folgende Kriterien zu beachten:

Table

Description automatically generated

Abbildung : Bewertungskriterien a

Table

Description automatically generated

Abbildung : Bewertungskriterien b

1. Planen

Innerhalb der Planung der LB haben sich folgende Projektideen entwickelt:

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 : 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 : 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 : 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 : 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 : Testfall Nr. 5

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 : 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 : 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 : 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 : 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 : Testfall Nr. 10

1. 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.

1. Flussdiagramm

Diagram

Description automatically generated

Abbildung : Flussdiagramm Pong

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 Matrize | Die 8x16 blue LED Matrize wird für die Darstellung des Spielfeldes verwendet (Spieler/Ball). | 2 |
| Speaker | Der Speaker wird verwendet, um ein Punkt zu signalisieren. | 1 |

1. Realisieren

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.

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.

1. Kontrollieren

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 vorbeischiesst 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 : Testprotokoll

1. Auswerten

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

1. Abbildungsverzeichnis

[Abbildung 1: Bewertungskriterien a 4](#_Toc132984356)

[Abbildung 2: Bewertungskriterien b 5](#_Toc132984357)

[Abbildung 3: Flussdiagramm Pong 10](#_Toc132984358)

1. Tabellenverzeichnis

[Tabelle 1: Testfall Nr. 1 6](#_Toc132984345)

[Tabelle 2: Testfall-Nr. 2 6](#_Toc132984346)

[Tabelle 3: Testfall Nr. 3 7](#_Toc132984347)

[Tabelle 4: Testfall Nr. 4 7](#_Toc132984348)

[Tabelle 5: Testfall Nr. 5 7](#_Toc132984349)

[Tabelle 6: Testfall Nr. 6 8](#_Toc132984350)

[Tabelle 7: Testfall-Nr. 7 8](#_Toc132984351)

[Tabelle 8: Testfall Nr. 8 8](#_Toc132984352)

[Tabelle 9: Testfall Nr. 9 8](#_Toc132984353)

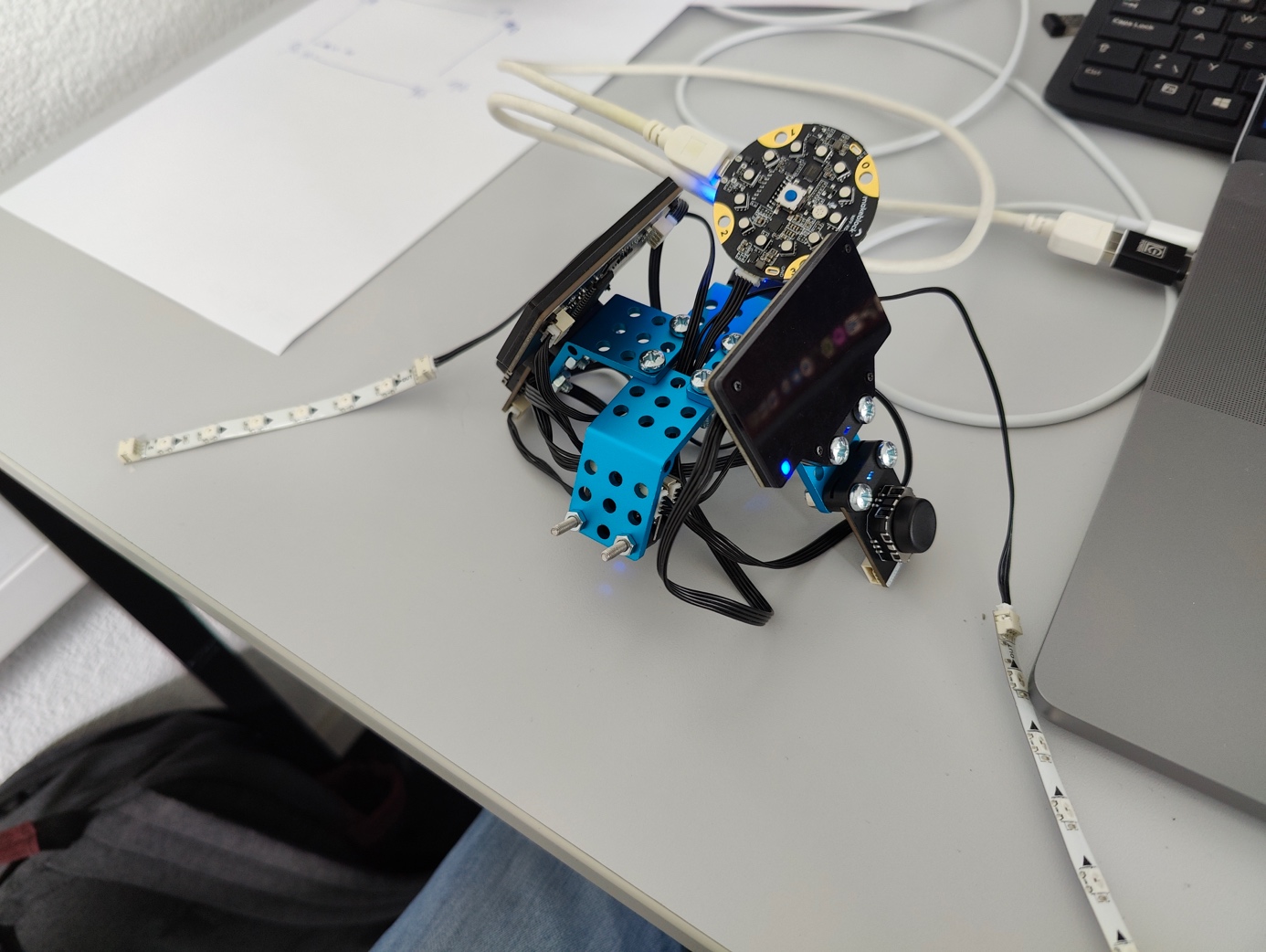
[Tabelle 10: Testfall Nr. 10 9](#_Toc132984354)

[Tabelle 11: Testprotokoll 13](#_Toc132984355)

1. Quellenverzeichnis

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

1. Anhangsverzeichnis

* Foto
* Source Code: https://github.com/alex92ch/242