**HTL Saalfelden**

**Systemplanung und Projektentwicklung**

****

**Projektdokumentation**

**2023 / 2024**

|  |  |
| --- | --- |
| **Projektbezeichnung** | MBot2 |
| **Projektteam** | Philipp Kirchtag, Sebastian Krallinger, Fabian Scharfetter, Eduard Voicescu |
| **Erstellt am** | 10.01.2024 |
| **Letzte Änderung am** | 21.02.2024 |
| **Status** | [in Bearbeitung] |
| **Aktuelle Version** | 1.0 |

**Änderungsverlauf**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr.** | **Datum** | **Version** | **Geänderte Kapitel** | **Art der Änderung** | **Autor** |
| 1 | 10.01.2024 | 1.0 | Alle | Erstellung | Sebastian Krallinger |
| 2 | 24.01.2024 | 1.0 | 2 | Anforderungen | Sebastian Krallinger |
| 3 | 21.02.2024 | 1.0 | 6.1 | Sprint 1 Doku, | Philipp Kirchtag |
| 4 | 22.02.2024 | 1.0 | 3,4.2 | Nichtfunktionale Anforderungen, Machbarkeitsstudie | Philipp Kirchtag |
| 5 | 07.03.2024 | 1.0 |  |  | Fabian Scharfetter |
| 6 | 11.4.2024 | 1.0 | 5.3 | Endpoint-Doku | Fabian Scharfetter |

**Inhalt**

[1. Allgemeines / Projektübersicht 4](#_Toc163737326)

[1.1 Projektbeschreibung 4](#_Toc163737327)

[1.2 Projektteam und Schnittstellen 4](#_Toc163737328)

[2. Funktionale Anforderungen 4](#_Toc163737329)

[2.1 Use Cases 4](#_Toc163737330)

[2.1.1 Netzwerkanbindung 4](#_Toc163737331)

[2.1.2 Web App 5](#_Toc163737332)

[2.1.3 suicide-Prevention 5](#_Toc163737333)

[2.1.4 Linien Folgen 5](#_Toc163737334)

[3. Nichtfunktionale Anforderungen 5](#_Toc163737335)

[3.1.1 Website 5](#_Toc163737336)

[3.1.2 LEDs 5](#_Toc163737337)

[4. Projektplanung 5](#_Toc163737338)

[4.1 Variantenbildung 6](#_Toc163737339)

[4.2 Machbarkeitsstudie 6](#_Toc163737340)

[4.3 Allgemeine Planungsinformationen 6](#_Toc163737341)

[4.4 Projektumfeldanalyse 6](#_Toc163737342)

[5. Softwarearchitektur 6](#_Toc163737343)

[5.1 Aktivitätsdiagramme 7](#_Toc163737344)

[5.1.1 Aktivitätsdiagramm 1 Name 7](#_Toc163737345)

[5.1.2 Aktivitätsdiagramm n Name 7](#_Toc163737346)

[5.2 Sequenzdiagramme 7](#_Toc163737347)

[5.2.1 Sequenzdiagramm 1 Name 7](#_Toc163737348)

[5.2.2 Sequenzdiagramm n Name 7](#_Toc163737349)

[5.3 Endpoint Beschreibung 7](#_Toc163737350)

[5.4 Verteilungsdiagramme 7](#_Toc163737351)

[5.5 Softwarekomponenten / Programme 7](#_Toc163737352)

[5.5.1 SW Programme 7](#_Toc163737353)

[5.5.2 SW Komponenten 8](#_Toc163737354)

[6. Projektdurchführung 9](#_Toc163737355)

[6.1 Sprint 1 9](#_Toc163737356)

[6.1.1 Sprintplanung 9](#_Toc163737357)

[6.1.2 Sprint Demo 9](#_Toc163737358)

[6.1.3 Sprint Retrospektive 9](#_Toc163737359)

[6.1.4 Sprint Zusammenfassung 10](#_Toc163737360)

[6.2 Sprint 2 11](#_Toc163737361)

[6.2.1 Sprintplanung 11](#_Toc163737362)

[6.2.2 Sprint Demo 11](#_Toc163737363)

[6.2.3 Sprint Retrospektive 11](#_Toc163737364)

[6.2.4 Sprint Zusammenfassung 11](#_Toc163737365)

[6.3 Sprint 3 13](#_Toc163737366)

[6.3.1 Sprintplanung 13](#_Toc163737367)

[6.3.2 Sprint Demo 13](#_Toc163737368)

[6.3.3 Sprint Retrospektive 13](#_Toc163737369)

[6.3.4 Sprint Zusammenfassung 13](#_Toc163737370)

[6.4 Sprint n 14](#_Toc163737371)

[7. Installation / Software deployment 15](#_Toc163737372)

[8. Projektabschluß 15](#_Toc163737373)

[8.1 Projektzusammenfassung 15](#_Toc163737374)

[8.2 Attachments 15](#_Toc163737375)

# Allgemeines / Projektübersicht

## Projektbeschreibung

Der mBot2 von mblock ist ein Roboter, der viele verschiedene Sensoren besitzt, mit denen er mit der Außenwelt kommunizieren kann. Programmiert wird er über MicroPython, eine abgespeckte Variante von Python3. Über einen cyberpi, eine „Controllereinheit“, welche auf den mBot2 gesteckt wird, kommuniziert man mit dem Roboter. Durch die verschiedenen Sensoren sollen verschiedenen Anforderungen, wie zum Beispiel die Steuerung über ein Endgerät, Anzeige des Status mittels LEDs oder die Implementierung eines Sicherheitsmodus, realisiert werden.

Das Projekt wird mit SCRUM umgesetzt und der Quellcode sowie die Dokumentation in einem GITHUB Repository abgelegt.

## Projektteam und Schnittstellen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rolle(n)** | **Name** | **Telefon** | **E-Mail** | **Team** |
| Projektleiter | Fabian Scharfetter | 0664 1900355 | fabian.scharfetter@htl-saalfelden.at | 3 |
| Projektmitarbeiter | Philipp Kirchtag | 0664 75039947 | philipp.kirchtag@htl-saalfelden.at | 3 |
| Projektmitarbeiter | Eduard Voicescu | 0677 6200810504 | eduard.voicescu@htl-saalfelden.at | 3 |
| Projektmitarbeiter | Sebastian Krallinger | 0660 2384312 | sebastian.krallinger@htl-saalfelden.at | 3 |

# Funktionale Anforderungen

## Use Cases

### Netzwerkanbindung

Es ist davon auszugehen das der Roboter und das Gerät sich immer im gleichen Netzwerk befinden. Es soll möglich sein, dass sich der mBot2 nach dem Start mit dem WLAN verbindet und die Anwendung automatisch findet. Mit den LEDs der Controller Einheit soll der Verbindungsstatus angezeigt werden (Blau: Verbindung wird aufgebaut, Grün: WLAN-Verbindung erfolgreich, Rot: WLAN-Verbindung fehlgeschlagen). Bei erfolgreicher Verbindung wird für 5 Sekunden die IP-Adresse am Display angezeigt.

### Web App

Für die Steuerung des mBots2 soll eine Web-App erstellt werden und so eine Cross Plattform Anwendung entwickelt werden. Diese muss Steuerbefehle an den Roboter senden können und die Daten der Sensoren sowie die Geschwindigkeit empfangen und anzeigen. Es soll die Möglichkeit bestehen mehrere Roboter über eine Anwendung zu steuern und der gerade verbundenen soll mit einem eindeutigen Namen und seiner IP-Adresse angezeigt werden. Ebenso soll eine Einstellungsmöglichkeit vorhanden sein bei denen man die LEDs des mBots steuern kann.

### suicide-Prevention

Dieses Feature soll ein/ausgeschaltet werden können und sicherstellen das der mBot2 nicht mutwillig gegen eine Wand gefahren wird.

### Linien Folgen

Es soll einen Modus geben, bei dem der mBot2 von allein eine Linie erkennt und dieser folgt, damit er zum Beispiel eine „Rennstrecke“ abfahren kann.

# Nichtfunktionale Anforderungen

### Website

Um eine schöne Oberfläche für den Benutzer zu bieten, soll die Website die über html, css und js erstellt wird und zur Steuerung des mBots dient mit einem responsiven Design ausgestattet sein. Die Steuerelemente auf der Website sollen unkompliziert und schön dargestellt werden.

### LEDs

Um verschiedene Zustände des mBots anzeigen zu können sollen die 4 LEDs in verschiedenen Farben je nach Zustand leuchten.

Gibt es spezielle „Nichtfunktionale Anforderungen“ so sind diese in diesem Kapitel anzugeben. z.B.: Verwendetes System, zu erwartender Speicherbedarf, zu erwartende Anzahl an Transaktionen, ....

# Projektplanung

In diesem Kapitel sollen grundlegende Fragen geklärt werden bevor mit der Projektdurchführung begonnen wird. Obwohl es klar ist, das dieses Projekt mit Hilfe von agilen PM Methoden durchgeführt wird, sollen vorab grundsätzliche Entscheidungen geklärt werden wie z.B. verwendete Programmiersprache, Variantenbildung, Betriebssystem, ...

Manche dieser Punkte können auch auf Grund der gegebenen Anforderungen als „gesetzt“ hingenommen werden.

## Variantenbildung

Auf Basis der Projektanforderungen, welche Varianten für die Umsetzung wurden ausgearbeitet und welche wurde schließlich gewählt und warum. Welche Varianten wurden verworfen und aus welchem Grund.

## Machbarkeitsstudie

Um mit dem mBot vertraut zu werden, wurden vor dem Projektstart in einem demo Projekt verschiedene Funktionen getestet wie z. B. die Verbindung im Netzwerk zwischen PC, Smartphone und mBot sowie die Fortbewegung des mBots über Buttons auf einer Website welche von einem Endgerät gesteuert werden können.

Gab es einzelne Punkte, die vorher grob getestet wurden um zu Zeigen das die Umsetzung grundsätzlich möglich ist.

z.B. Kommunikation mit einem MQTT-Server mit Hilfe der Programmiersprache Python, ...

## Allgemeine Planungsinformationen

Andere Planungsinformationen welche nicht durch andere Kapitel abgedeckt sind werden hier eingetragen.

## Projektumfeldanalyse

Führen Sie eine Analyse des Projektumfeldes durch. Welche vergleichbaren Produkte gibt es bereits am Markt. Wie erfolgt die Abgrenzung zu diesen bereits bestehenden Produkten? Wer sind die relevanten Stakeholder des Projektes.

# Softwarearchitektur

In diesem Kapitel soll der Aufbau der Software/Hardware beschrieben werden.

aus welchen Komponenten besteht das SW-Produkt.

Wie interagieren die einzelnen Komponenten miteinander.

Auf welcher Hardware läuft das System bzw. handelt es sich vielleicht sogar um ein verteiltes System. Wie kommunizieren diese Komponenten miteinander.

## Aktivitätsdiagramme

### Aktivitätsdiagramm 1 Name

Beschreibung der entsprechenden Aktivität und einfügen des Aktivitätsdiagramm

### Aktivitätsdiagramm n Name

Beschreibung der entsprechenden Aktivität und einfügen des Aktivitätsdiagramm

## Sequenzdiagramme

### Sequenzdiagramm 1 Name

Beschreibung der entsprechenden Sequenz und Einfügen des Sequenzdiagramm

### Sequenzdiagramm n Name

Beschreibung der entsprechenden Sequenz und Einfügen des Sequenzdiagramm

## Endpoint Beschreibung

* /mBot

Html Model wird übergeben

Endpoint gibt Hauptseite (index.html) zurück

IP-Adresse des mBots wird an Hauptseite wird auf der Hauptseite angezeigt

* /mBotIP

Gibt mBotIP.html zurück

* /getDevice

Wird von der mBotIP Seite aufgerufen

Übergeben wird die eingegebene IP-Adresse als String

Danach wird man zu */mBot* redirected

* /suicidePrevention

Ein String wird von der vom Hauptseiten-Script *index.js* übergeben   
(Entweder ON oder OFF)

Dieser wird in ein Packet verpackt und an den mBot gesendet

* /buttonControl

Ein String wird von der vom Hauptseiten-Script *index.js* übergeben  
(UP, DOWN, LEFT, RIGHT oder STOP)

Dieser wird in ein Packet verpackt und an den mBot gesendet

* /arrowControl

Ein HttpServletRequest wird von der vom Hauptseiten-Script *eventlistener\_keys.js* übergeben (UP, DOWN, LEFT, RIGHT oder STOP)

Dieser wird in ein Packet verpackt und an den mBot gesendet

* /joystickControl

Ein HttpServletRequest wird von der vom Hauptseiten-Script *eventlistener\_keys.js* übergeben (UP, DOWN, LEFT, RIGHT, LEFT\_B, RIGHT\_B oder STOP)

Dieser wird in ein Packet verpackt und an den mBot gesendet

* /speedControl

Ein HttpServletRequest wird von der vom Hauptseiten-Script *index.js* übergeben   
Dieser beinhaltet den Wert des Geschwindigkeits-Sliders (1 bis 100)

Dieser wird in ein Packet verpackt und an den mBot gesendet

* /getSensordata

Ein String mit allen Sensordaten wird vom mBot übergeben

Lecker Bierchen

## Verteilungsdiagramme

Zeigt an wie die einzelnen Teile der Software auf die Hardwarekomponenten verteilt sind und wie die Hardwarekomponenten miteinander verbunden sind.

Auf welchem Rechner läuft welcher Software. Wie sind diese über ein Netzwerk miteinander verbunden.

## Softwarekomponenten / Programme

### SW Programme

Auflistung aller verwendeten SW Programme die bei der Umsetzung des Projektes verwendet worden sind. inkl. Angabe der Versionsnummer

z.B.: Visual Studio 2022,...

### 5.5.2 SW Komponenten

Auflistung aller verwendeten SW Komponenten welche für den Betrieb der SW benötigt werden. z.B.: Java Version, Apache Webserver, DotNet Framework, SW Library XY

inkl. Versiosnummer, Hersteller, Bezugsquelle (Downloadlink, ...) und SW-Lizenz (GPL, LGPL, Apache License, ...

# Projektdurchführung

## Sprint 1

### Sprintplanung

Dauer: 07.02.2024 – 27.02.2024

Ausgewählte User Stories:

* Steuerung PC, 5 SP, 11 VP
  + Der Benutzer soll über einen PC mit Buttons auf der Website sowie über Pfeil- und WASD – Tasten den mBot steuern können.
* Statusanzeige, 2 SP, 2 VP
  + Die LEDs des mBots sollen je nach Verbindungsstatus angezeigt werden
* Verbindung herstellen, 3 SP, 11 VP
  + Alle mBots im verbundenen Netzwerk werden angezeigt und können ausgewählt werden, um eine Verbindung herzustellen.

Anzahl Story points: 10

### Sprint Demo

Die User Stories Steuerung PC sowie Satusanzeige wurden erfolgreich umgesetzt. Der mBot kann über Pfeil- oder WASD-Tasten angesteuert werden und die LEDs leuchten je nach Verbindungsstatus in anderen Farben.

Die User Story Verbindung herstellen wurde aufgrund von … nicht abgeschlossen und wird in den nächsten Sprint übernommen.

### Sprint Retrospektive

Was gut lief:

* Sprint-Planung
* Dokumentation
* Schätzung des Aufwands (Value Points)
* Umgang mit GitHub

Verbesserungsbedarf:

* Arbeitsaufteilung
* Konzept bei Verbindung mit mBot

### Sprint Zusammenfassung

Während des Sprints wurden keine neuen User Stories in das Product Backlog eingefügt bzw. entfernt.

Burndownchart:

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung

Von 10 eingeplanten Story-Points wurden 7 komplett abgeschlossen.  
Die übrigen 3 Story -Points sind bereits in Bearbeitung und fast fertig – sie werden in den nächsten Sprint übernommen.

Durchschnittliche Sprint Velocity über alle bisherigen Sprints: 7

## Sprint 2

### Sprintplanung

Dauer: 29.02.2024 – 20.03.2024

Ausgewählte User Stories:

* Website: 20 SP, 10 VP
  + Als Benutzer möchte ich eine Website über die ich den mBot2 steuern kann und alle Werte angezeigt bekomme.
* Steuerung Handy: 8 SP, 11 VP
  + Als Benutzer möchte ich den mBot2 über mein Handy steuern können, um herumzufahren.
* Sicherheitsmodus: 8 SP, 8 VP
  + Als Benutzer möchte ich eine Sicherheitsmodus ein- und ausschalten können, um nicht gegen die Wand fahren zu können.

Anzahl Story points: 36

Ausgewählte Punkte aus der Impediment Liste:

* Verbindung herstellen: 3 SP, 11 VP

### Sprint Demo

Kurze Beschreibung welche User Stories umgesetzt worden sind und welche Funktionen erfolgreich präsentiert worden sind.

Auflistung welche Punkte nicht umgesetzt werden konnten und warum.

### Sprint Retrospektive

Was lief gut/schlecht während des Sprints.

Gibt es neue Impediment Punkte die während des Sprints identifiziert worden sind.

Auflistung der Impediment Taskliste.

### Sprint Zusammenfassung

Kurze Zusammenfassung der Sprintdurchführung.

Wurden in diesem Sprint neue User Stories in das Product Backlog eingefügt und wenn ja, welche.

Wurden in diesem Sprint User Stories aus dem Product Backlog entfernt und wenn ja, welche und warum.

Burndownchart

Sprint Velocity

Auf Basis der Sprint Velocity, wie hat sich der geplante Endtermin verändert.

Durchschnittliche Sprint Velocity über alle bisherigen Sprints.

## Sprint 3

### Sprintplanung

Dauer:

Ausgewählte User Stories:

<Auflistung der User Stories welche in diesem Sprint bearbeitet werden sollen. Pro User Story: Angabe des Namens, Zenhub ID, Kurzbeschreibung sowie Story Points>

Anzahl Story points: <Geplante Anzahl an Story Points welche in diesem Sprint umgesetzt werden sollen>

Ausgewählte Punkte aus der Impediment Liste: <Geplante Punkte der Impediment Liste welche in diesem Sprint umgesetzt werden sollen>

### Sprint Demo

Kurze Beschreibung welche User Stories umgesetzt worden sind und welche Funktionen erfolgreich präsentiert worden sind.

Auflistung welche Punkte nicht umgesetzt werden konnten und warum.

### Sprint Retrospektive

Was lief gut/schlecht während des Sprints.

Gibt es neue Impediment Punkte die während des Sprints identifiziert worden sind.

Auflistung der Impediment Taskliste.

### Sprint Zusammenfassung

Kurze Zusammenfassung der Sprintdurchführung.

Wurden in diesem Sprint neue User Stories in das Product Backlog eingefügt und wenn ja, welche.

Wurden in diesem Sprint User Stories aus dem Product Backlog entfernt und wenn ja, welche und warum.

Burndownchart

Sprint Velocity

Auf Basis der Sprint Velocity, wie hat sich der geplante Endtermin verändert.

Durchschnittliche Sprint Velocity über alle bisherigen Sprints.

## Sprint n

# Installation / Software deployment

Anleitung welche Schritte notwendig sind um das fertige SW Produkt zu installieren und in Betrieb zu nehmen.

# Projektabschluß

## Projektzusammenfassung

Zusammenfassung der Projektdurchführung. Was lief gut/schlecht. Welche Erkenntnisse wurden während der Durchführung des Projektes gewonnen. Was würde man, nun anders machen bzw. wieder gleich machen?

## Attachments

Tabellarische Auflistung der Projektdateien.

z.B.: ZIP-File mit dem Quellcode, Projektpräsentationen, ...