|  |
| --- |
| \_\_\_\_  ***Project task 1***  *Pololu Zumo 32U4*  Doc.-Number: Dokument2  Doc.-Version: A3  299792458 |
| Customer:  NewTec  Development  Buchenweg 3  89284 Pfaffenhofen a. d. Roth  Germany |
|  |
| Author:  Team: ~~ o=o\  Hs Offenburg |

Change History

|  |  |  |
| --- | --- | --- |
| Doc.-Version | Description of Modification | Date |
| A1 | Initial revision |  |
| A2 | First version of use case diagrams, use case descriptions and requirements written | 20.03.24 |
| A3 | Use case diagrams, use case descriptions and requirements modified | 27.03.24 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Release

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Name | Responsibility | Date | Signature |
| Creation |  |  |  | i.A. |
| Verification |  |  |  | i.A. |
| Approval |  |  |  | i.A. |
| Release |  |  |  | i.A. |

Table of Contents

Change History 2

Release 2

Table of Contents 3

List of Tables 4

List of Figures 4

1 General 5

1.1 Scope of Document 5

1.2 Abbreviations 5

1.3 Terminology 5

1.4 Referenced Documents 5

1.5 Applicable Standards 6

2 Introduction 6

2.1 System Overview 6

2.2 Interface Overview 7

2.3 Scenarios 8

2.3.1 System 8

2.3.2 Interface 11

3 Requirements 15

3.1 Functional Requirements 15

3.2 Non-Functional Requirements 15

4 Interface 15

4.1 External Interfaces 15

5 Document Management 16

5.1 Document Creation 16

6 Appendix 16

List of Tables

[Table 1: Abbreviations 5](#_Toc162469300)

[Table 2: Terminology 5](#_Toc162469301)

[Table 3: Referenced Documents 5](#_Toc162469302)

[Table 4: Applicable Standards 6](#_Toc162469303)

List of Figures

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# General

## Scope of Document

## Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Description |
| N/A | Not applicable |

Table 1: Abbreviations

## Terminology

|  |  |
| --- | --- |
| Term | Description |
| Robot | Zumo32U4 |
| OLED | organic light emitting diode |
| OLED-display | display on the robot |
| StartButton | button on the robot, accessible for input |
| Speaker | buzzer on the robot |
| Redetect |  |
| Power-On |  |
| Short Beep | signal of frequency 440Hz, duration 100ms |
| Alarm Signal | signal of frequency 440Hz, duration 200ms, repeated thrice |

Table 2: Terminology

## Referenced Documents

|  |  |  |
| --- | --- | --- |
| Reference | Document-Identification | Description |
| [1] | N/A | N/A |
|  | N/A | N/A |

Table 3: Referenced Documents

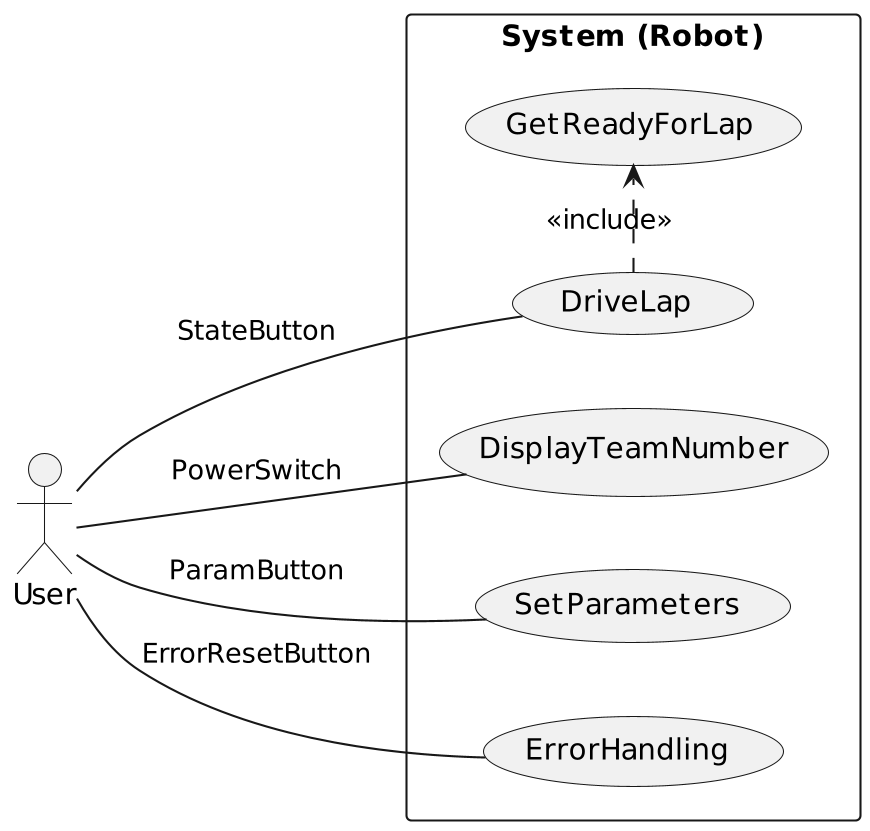
## Applicable Standards

|  |  |  |
| --- | --- | --- |
| Reference | Document-Identification | Description |
| [1] | N/A | N/A |
|  | N/A | N/A |

Table 4: Applicable Standards

# Introduction

## System Overview



## PlantUML diagramInterface Overview

## Scenarios

### System

|  |  |
| --- | --- |
| Reference number | 2.3.1.1 |
| Name | *GetReadyForLap* |
| Short description | The robot shows a count down from 3s to 0s and then starts driving. |
| Precondition | No other scenario is active |
| Postcondition | The robot recognizes the *StartFinishLine* |
| Error case | The *StartFinishLine* is not recognizes after 5s |
| Actors | *User* |
| Trigger | The *StateButton* has been pressed |
| Standard sequence | * The robot shows a count down from 3s to 0s on the *OledDisplay* * The robot begins driving * The robot recoginzes the *StartFinishLine*, immediately plays a *ShortBeep* on the *Buzzer* and switches to the state *DriveLap* |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.1.2 |
| Name | *DriveLap* |
| Short description | The robot drives one lap as fast as possible |
| Precondition | The state *GetReadyForLap* is finished |
| Postcondition | The *StartFinishLine* is recognized and the robot has stopped |
| Error case | The *StartFinishLine* is not recognized after 20s |
| Actors | *none* |
| Trigger | The state *GetReadyForLap* is finished |
| Standard sequence | * The state *GetReadyForLap* is finished * The robot continues driving for one lap * The robot recognizes the *StartFinishLine* and stops |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.1.3 |
| Name | *DisplayTeamNumber* |
| Short description | The *TeamNumber* is shown on the *OledDisplay* |
| Precondition | No other scenario is active |
| Postcondition | The *TeamNumber* is still being displayed |
| Error case | none |
| Actors | *User* |
| Trigger | *PowerSwitch* is turned on |
| Standard sequence | * The *PowerSwitch* is turned on * The *TeamNumber* is shown on the *OledDisplay* * (Only user input or an error ends this state) |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.1.4 |
| Name | *SetParameters* |
| Short description | Allows individual parameters to be adjusted |
| Precondition | Robot is in state *DisplayTeamNumber* |
| Postcondition | Robot goes to state *DisplayTeamNumber* |
| Error case | none |
| Actors | *User* |
| Trigger | The *ParamButton* is pressed |
| Standard sequence | * Robot is in state *DisplayTeamNumber and the ParamButton is pressed* * Shows the variables to be set on the display and allows them to be adjusted * The robot goes back to the state *DisplayTeamNumber* |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.1.5 |
| Name | *ErrorHandling* |
| Short description | Displays an error message until the user resets it |
| Precondition | An error occurred |
| Postcondition | Robot goes to state *DisplayTeamNumber* |
| Error case | none |
| Actors | *User* |
| Trigger | An error occurs |
| Standard sequence | * An error occurs * The robot stops immediately * The *AlarmSignal* is played on the *Buzzer* * An error message is displayed on the *OledDisplay* * The User resets the state to *DisplayTeamNumber* via the *ErrorResetButton* |
| Alternative sequences | none |

### Interface

|  |  |
| --- | --- |
| Reference number | 2.3.2.1 |
| Name | *Initialization* |
| Short description | During this step, all required features and variables are initialized |
| Precondition | Robot is switched off, hardware reset button or *ErrorReset* has been pressed |
| Postcondition | All variables must be initialized and the robot is ready for input |
| Error case | Jump to *ErrorHandling* and output the corresponding error code |
| Actors | *PowerSwitch* |
| Trigger | PowerSwitch, *ErrorReset* if the robot is in *ErrorHandling* and Hardware reset button |
| Standard sequence | All required system features such as memory, GPIO, timer, variables, etc. are initialized |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.2.2 |
| Name | *WaitingForInput* |
| Short description | The robot is initialized and waits for input from the user, while he is waiting the team name is displayed |
| Precondition | The initialization must be completed |
| Postcondition | User must have made an input and depending on this the robot behaves accordingly |
| Error case | Jump to *ErrorHandling* and output the corresponding error code |
| Actors | *OLED-Display* |
| Trigger | Initialization has been successfully completed |
| Standard sequence | The team name is shown on the OLED display until input is received from the user |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.2.3 |
| Name | *GetReadyForLap* |
| Short description | The robot starts with a count down from 3 to 0 and then starts moving. |
| Precondition | The robot must be in *WaitingForInput* |
| Postcondition | Robot must travel as fast as possible |
| Error case | Jump to *ErrorHandling* and output the corresponding error code |
| Actors | *OLED-Display, Motors* |
| Trigger | The StateButton has been pressed |
| Standard sequence | The robot starts a count down from 3 to 0 which is shown on the display. The robot then moves off until it just passes the start/finish line |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.2.4 |
| Name | *DriveLap* |
| Short description | The robot drives the fastest possible lap |
| Precondition | The robot must cross the start/finish line in *GetReadyForLap* |
| Postcondition | The robot must stop |
| Error case | Jump to *ErrorHandling* and output the corresponding error code |
| Actors | *OLED-Display, Motors* |
| Trigger | The robot has crossed the start/finish line in *GetReadyForLap* |
| Standard sequence | The robot drives the fastest possible lap around the given course |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.2.5 |
| Name | *LineRecognition* |
| Short description | It recognizes where the guideline is and whether the start/finish line has been crossed |
| Precondition | *Initialization* is finished |
| Postcondition | none |
| Error case | Jump to *ErrorHandling* and output the corresponding error code |
| Actors | *LineSensor, Buzzer* |
| Trigger | Robot is either in *GetReadyForLap* or in *DriveLap* |
| Standard sequence | It recognizes where the guideline is and every time the start/finish line is crossed, the *buzzer* is activated |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.2.6 |
| Name | *TimeMeasurement* |
| Short description | Measures the time of a lap |
| Precondition | Robot must cross the start/finish line while it is in *GetReadyForLap* |
| Postcondition | Timer must be stopped |
| Error case | Jump to *ErrorHandling* and output the corresponding error code |
| Actors | none |
| Trigger | The timer is triggered by crossing the start/finish line in GetReadyForLap |
| Standard sequence | The timer is started by crossing the start/finish line in GetReadyForLap and is stopped as soon as the start/finish line is crossed again |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.2.7 |
| Name | *DisplayLapTime* |
| Short description | Displays the completed lap time |
| Precondition | One round must have been successfully completed |
| Postcondition | Time is no longer displayed |
| Error case | Jump to *ErrorHandling* and output the corresponding error code |
| Actors | *OLED-Display, StateButton* |
| Trigger | Start/finish line was crossed in DriveLap |
| Standard sequence | The completed lap time is displayed |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.2.8 |
| Name | *ErrorHandling* |
| Short description | Displays an error message |
| Precondition | The program must have thrown an error during its runtime |
| Postcondition | New initialization of the program |
| Error case | none |
| Actors | *OLED-Display, Buzzer, Motors* |
| Trigger | An error has occurred in the program |
| Standard sequence | The robot must stop immediately, a warning tone is emitted via the buzzer and an error message is shown on the display |
| Alternative sequences | none |

|  |  |
| --- | --- |
| Reference number | 2.3.2.9 |
| Name | *SetParameters* |
| Short description | Allows individual parameters to be adjusted during runtime |
| Precondition | Robot must be in *WaitingForInputs* or *DisplayLapTime* |
| Postcondition | Robot is in *WaitingForInput* |
| Error case | Jump to *ErrorHandling* and output the corresponding error code |
| Actors | *OLED-Display, ParamButton* |
| Trigger | The ParamButton button has been pressed |
| Standard sequence | Shows the variables to be set on the display and allows them to be adjusted |
| Alternative sequences | none |

# Requirements

## Functional Requirements

The Robot shall start driving 3s after the operator presses the StartButton.

If the Robot detects the start line the Robot shall do ALL of the following steps

* emit a short beep via Buzzer
* start the time measurement for the lap

If the Robot detects the end line the Robot shall do ALL of the following steps

* stop any motion
* emit a short beep via Buzzer
* end the time measurement for the lap
* display the measured time for the lap on the OLED-Display

The Robot shall complete one lap in 20s or less otherwise the Robot shall detect an error

If the Robot leaves the track, the robot shall Redetect the track in 5s or less otherwise the Robot shall detect an error

The Robot shall run during daytime- or office light conditions

If the Robot detects an error the Robot shall do the following steps in order

* stop any motion
* emit an alarm signal via Buzzer
* display the error reason on the OLED

After Power-On the Robot shall display the team-number on the OLED-display for at least 2s

## Non-Functional Requirements

The software shall be hardware independent

The programmer shall write the software in the programming language C

During the competition the only changes to the software shall be the selection of a predefined list of parameters

The software shall at most use 80% of the available flash memory

# Interface

## External Interfaces

# Document Management

## Document Creation

# Appendix