

---

# Specifications

Line-Follower software for Pololu Zumo32U4

---

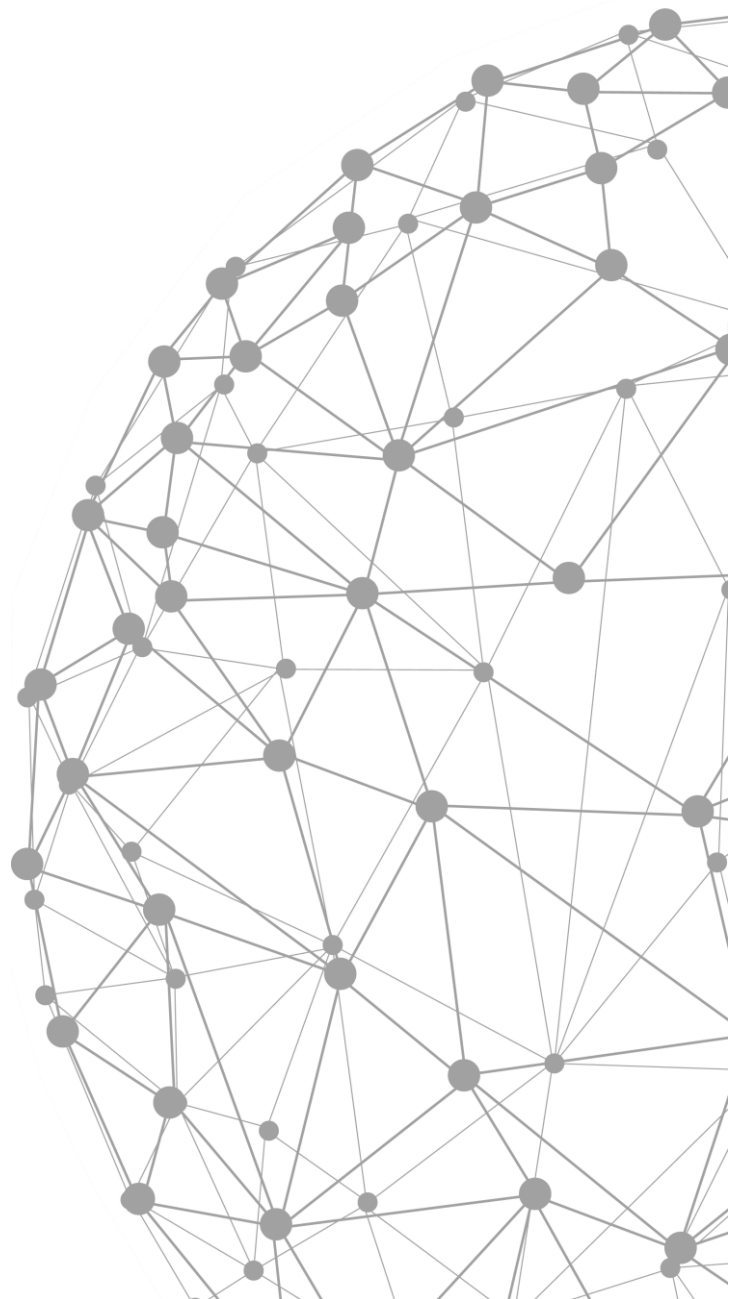
Doc.-Number: Specifications\_V3

Doc.-Version: V3

Customer:  
Martin Dembinsky, MSc  
Niclas Heitz, MSc  
Labor Software Engineering  
Hochschule Offenburg  
Germany

Author:  
Die Fantastischen Vier

Date of delivery: 28.06.2024



Change History

Doc.-Version	Description of Modification	Date
V1	Initial revision	02.04.2024
V2	Updated version	25.06.2024
V3	Final version	27.06.2024

Release

	Name	Responsibility	Date	Signature
Creation	Die Fantastischen Vier		28.03.2024	Fanta4Group
Verification				
Approval				
Release				

## Table of Contents

Change History .....	2
Release.....	2
Table of Contents.....	3
List of Tables .....	4
List of Figures .....	4
<b>1 General.....</b>	<b>5</b>
1.1 Scope of Document.....	5
1.2 Abbreviations .....	5
1.3 Terminology .....	5
1.4 Referenced Documents.....	6
1.5 Applicable Standards .....	6
<b>2 Introduction .....</b>	<b>7</b>
2.1 System Overview .....	7
2.2 Interface Overview .....	7
2.3 Scenarios.....	7
<b>3 Requirements .....</b>	<b>7</b>
3.1 Functional Requirements.....	7
3.2 Non-Functional Requirements .....	8
3.2.1 Safety & Security.....	8
3.2.2 Data.....	8
3.2.3 Environmental Conditions.....	9
3.2.4 Quality .....	9
3.2.5 Computer Resources .....	9
3.2.6 Design Constraints.....	9
3.2.7 Product Documentation.....	9
3.2.8 Production.....	9
3.2.9 Logistic .....	9
3.2.10 Commercial Requirements .....	10
3.2.11 Further Requirements .....	10
<b>4 Interface .....</b>	<b>10</b>
4.1 External Interfaces .....	10
<b>5 Document Management .....</b>	<b>10</b>
5.1 Document Creation .....	10
<b>6 Appendix.....</b>	<b>11</b>
6.1 Use Case Diagrams .....	11
6.1.1 System Robot .....	11
6.1.2 System Software.....	16

List of Tables

Table 1: Abbreviations.....5

Table 2: Terminology.....6

Table 3: Referenced Documents.....6

Table 4: Applicable Standards.....6

List of Figures

Figure 1: Use case diagram System Robot .....11

Figure 2: Use case diagram System Software .....16

## 1 General

### 1.1 Scope of Document

This document serves as a specification sheet for the customer order. It contains functional and non-functional requirements for the system. It also contains two use case diagrams of the robot and software system with detailed descriptions. This document comprises a total of 25 pages.

### 1.2 Abbreviations

Abbreviation	Description
N/A	N/A

Table 1: Abbreviations

### 1.3 Terminology

Term	Description
<i>Start line</i>	The start of the <i>track</i> is crossed with a 90° crossbar.
<i>Stop line</i>	The end of the <i>track</i> is crossed with a 90° crossbar. The system distinguish between Start- and Stop lines depending on the program state. If the race has started it shall take a Start- line as an Stop-line.
<i>Track</i>	The <i>track</i> is a black line which the robot is going to follow.
<i>Standard track</i>	Is a pre-defined <i>track</i> , given by the customer.
<i>Alarm tone</i>	A short sound to notify in an <i>error state</i> . The sound shall follow the following schematic: 1/3 s sound 1/3 s silence 1/3 s sound at a frequency of 440 Hz. The sound shall be played with the maximum amplitude.
<i>Notification sound</i>	A short sound with a duration of 1 s at a frequency of 440 Hz. The sound shall be played with the maximum amplitude.
<i>Booting cycle</i>	After the robot is turned on or the reset button is pressed, the system starts the software.
<i>Error state</i>	The status the system jumps, after an error occurs. The system follows a strict procedure which is explained in the requirements.
<i>Race modes</i>	A selectable status of the system to change the behavior of the robot on the <i>track</i> .
<i>OLED</i>	Organic light-emitting diode. In this document OLED refers as a short form for the OLED-display.
<i>Pushbutton A</i>	Button on the robot labelt with A.
<i>Pushbutton B</i>	Button on the robot labelt with B.

Term	Description
<i>Pushbutton C</i>	Button on the robot labelt with C.
<i>System</i>	Full robot unit with program on it.
<i>Redection Pattern</i>	A combination of different actions the system execute after the track is lost.
<i>Safety mode</i>	The System drives slower than in the other modes to ensure the system loses the <i>track</i> less likely.
<i>Highspeed mode</i>	The system drives faster than in the other modes.
<i>Balanced mode</i>	This mode is compromise between the safety and <i>high-speed mode</i> .

Table 2: Terminology

## 1.4 Referenced Documents

Reference	Document-Identification	Description
[1]	11001_0099_0088_RD	Product specification for the Line-Follower Software for Pololu Zumo32U4
	N/A	N/A

Table 3: Referenced Documents

## 1.5 Applicable Standards

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

Table 4: Applicable Standards

## 2 Introduction

### 2.1 System Overview

The overall system of the Pololu Zumo32u4 has an IR proximity sensor system, a line sensor array, a dual motor drive system with encoder as well as an OLED display and buzzer also onboard a three-axis accelerometer, compass, and gyro.

### 2.2 Interface Overview

The board of the Arduino compatible ATmega32u4 microcontroller has a USB programming interface.

### 2.3 Scenarios

The Line-Follower software runs on the Zumo32U4 robot platform from Pololu. The robot drives a defined *track* as fast as possible from the start area to the finish area. The *track* is marked with a black line on white background. The team with the robot that drove the fastest complete round wins.

## 3 Requirements

### 3.1 Functional Requirements

1. If the *system* is turned on, the system shall execute all of the following actions in the given order
  - display the teams name for 2 s and
  - start the calibration process after the 2 s
  - select the *balanced mode* as initial *race mode*
2. The *system* shall have different sets of pre-defined parameters to change between the following *race modes*:
  - safety mode
  - highspeed mode
  - balanced mode
3. If the *pushbutton B* on the robot is pressed, the *system* shall change between the toggle through the *race modes* with each push.
4. If the user hasn't selected any *race mode*, the system shall use the *balanced mode*
5. If the *pushbutton C* on the robot is pressed, the *system* shall start the calibration process of the line sensor.
6. If the calibration process is running, the *system* shall disable all user inputs.
7. If the *pushbutton A* on the system is pressed, the *system* shall start driving after 3 s.

8. The robot shall start driving if all of the following instances apply
  - The sensors are calibrated
  - The user presses *pushbutton A*
9. If the *system* is driving it shall follow the detected *track*.
10. If the *track* in front of the robot ends, the *system* shall execute an *RedectionPattern*.
11. If the *start line* is detected, the *system* shall execute all of the following actions in the given order:
  - start the time measurement for the lap.
  - notify the user with an *notification sound*.
18. If the system detects an error, the *system* shall jump in an *error state*.
19. If the system enters the *error state*, it shall execute all of the following actions in the given order:
  - Stop all movements,
  - Notify with an *alarm tone*.
  - Display the error reason on the OLED.
20. The *system* shall jump in an *error state*, if either one of the following instances applies:
  - The *system* can't re-detect the *track* within 5 s.
  - The *system* can't complete one round on the standard *track* in maximum 20 s.
  - The *system* couldn't load the selected parameters correctly.
  - The *system* couldn't finish the calibration process correctly.
21. If the *system* detects the *stop line*, the system shall execute all of the following actions in the given order:
  - Stop all movements
  - Stop time measurement
  - Notify with a *notification sound*.
  - Display the measured time on the OLED .

## 3.2 Non-Functional Requirements

### 3.2.1 Safety & Security

none

### 3.2.2 Data

none



### 3.2.3 Environmental Conditions

1. The software shall work with normal daylight & workplace light conditions.
2. The *system* shall work on a playfield with a white background.
3. The *track* shall not cross itself
4. The *track* shall not contain any gaps larger than 10 cm.
5. Every gap shall continue with an opening angle of 30°.
6. The start and *stop line* are marked with 90° crossbeam.
7. The crossbeam are in a distance of 3.5 cm to the center line.
8. The crossbeam shall have a length of 5 cm.
9. The start and the stop are minimum 30 cm away from any gap in the *track*.
10. The minimum distance between playfield border and the *track* is 15 cm.
11. The minimum curve radius on a *track* is 10 cm.

### 3.2.4 Quality

1. The software shall work independently of the charge status of the batteries, unless the charge status is below 50%.
2. The flash memory usage shall never exceed equal to 80%.

### 3.2.5 Computer Resources

1. The software shall run on the Zumo32U4 robot platform from Pololu.
2. The software shall work in a way that allows to replace the hardware by e.g. a different robot or a hardware simulation. Unavailability of hardware features does not need to be considered.
3. The teams shall use the programming language C.

### 3.2.6 Design Constraints

none

### 3.2.7 Product Documentation

1. The teams shall write their documentations in English.
2. The teams shall use English for commit messages of version control system interactions.
3. The teams shall write the commit messages of version control system interactions descriptive.

### 3.2.8 Production

none

### 3.2.9 Logistic

none

### 3.2.10 Commercial Requirements

none

### 3.2.11 Further Requirements

1. The team shall not make any modifications on the hardware during executing the competition, except changing the batteries or small repairs.
2. Each team is able to test the robot on the *track* before the competition starts. During this phase the teams are allowed to change the software as well.
3. Each team shall have 3 attempts on the *track*. It counts the fastest complete round.
4. The system shall finish the race as fast as possible.

## 4 *Interface*

### 4.1 *External Interfaces*

not further specified

## 5 *Document Management*

### 5.1 *Document Creation*

All documents must be written in English.

## 6 Appendix

### 6.1 Use Case Diagrams

#### 6.1.1 System Robot

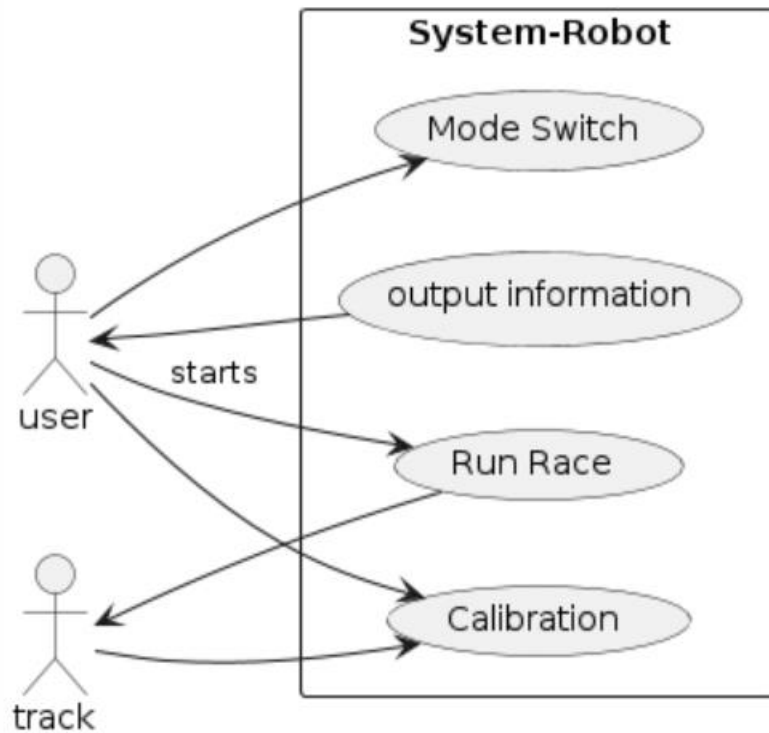


Figure 1: Use case diagram System Robot

Identifier	R01
Name	Mode Switch
Brief description	The users selects pre-defined parameters sets to change the behavior of the robot on the <i>track</i> .
Preconditions	The <i>system</i> is not in the use case "Run Race".
Postconditions	The desired <i>driving mode</i> is set.
Failure scenarios	The parameters couldn't be loaded correctly.
Actors	User
Trigger	User pushes <i>button B</i> .
Standard workflow	1. User pushes the <i>button B</i> . 2. The <i>driving mode</i> is switched between <i>highspeed mode</i> , <i>safety mode</i> or <i>balanced mode</i> .
Alternative workflow	

Identifier	R02
Name	<b>output information</b>
Brief description	The robot supplies the user with information through the <i>OLED</i> and the buzzer.
Preconditions	The <i>system</i> is not in the use case "Run Race" R03.
Postconditions	The <i>OLED</i> refreshed parameters and shows the team name.
Failure scenarios	The user couldn't be provided with all information.
Actors	User
Trigger	The robot was powered on./ The robot has passed the <i>stop line</i> .
Standard workflow	<ol style="list-style-type: none"> <li>1. The robot is powered on.</li> <li>2. The robot is initialized and calibrates.</li> <li>3. The <i>OLED</i> shows the team name.</li> <li>4. The robot finishes the race.</li> <li>5. The display shows the race time.</li> </ol>
Alternative workflow	<ol style="list-style-type: none"> <li>3a. The robot gets into an error case.</li> <li>4a. The error case is displayed</li> <li>5a. The buzzer beep</li> </ol>

Identifier	R03
Name	Run Race
Brief description	The user gives the command to start the race. The robot starts driving and follows the line.
Preconditions	Line sensor is calibrated, parameters are selected.
Postconditions	Robot has passed the <i>stop line</i> .
Failure scenarios	The robot loses the <i>track</i> .
Actors	User, <i>track</i>
Trigger	User
Standard workflow	<ol style="list-style-type: none"><li>1. User push the button A.</li><li>2. Robot checks for <i>start line</i>.</li><li>3. Robot detects the <i>start line</i>.</li><li>4. Robot detects the <i>track</i>.</li><li>5. Robot follows the <i>track</i>.</li><li>6. Robot checks for end line.</li><li>7. Robot stops driving.</li></ol>
Alternative workflow	<ol style="list-style-type: none"><li>5a. The robot loses the <i>track</i>.</li><li>6a. The robot redetects the <i>track</i>.</li><li>7a. Back to 6.</li><li>6b. The robot can't find the <i>track</i> within 5s.</li><li>7b. The robot stops driving.</li><li>8b. The robot notifies the user.</li></ol>

Identifier	R04
Name	Calibration
Brief description	Calibration starts on Init, or the users pushes the <i>button C</i> to start the calibration process.
Preconditions	The <i>system</i> is not in the use case "Run Race".
Postconditions	The line sensor is calibrated.
Failure scenarios	The calibration wasn't successful.
Actors	User, <i>track</i>
Trigger	User
Standard workflow	<ol style="list-style-type: none"><li>1. User lays the robot on the <i>track</i>.</li><li>2. User pushes the button C.</li><li>3. The robot starts the calibration process.</li><li>4. Robot notifies after the process is done.</li></ol>
Alternative workflow	<ol style="list-style-type: none"><li>4a. The robot notifies that the calibration wasn't successful.</li><li>5a. Back to 2.</li></ol>





Identifier	S01
Name	Init
Brief description	Initialization of the project parameters and <i>OLED</i> output of the team name.
Preconditions	The <i>system</i> is powered on.
Postconditions	Calibrate Sensors
Failure scenarios	Initialization failed and team name is not displayed.
Actors	OLED
Trigger	Power switch
Standard workflow	1. Power switch is on. 2. The team name is shown on the <i>OLED</i> .
Alternative workflow	

Identifier	S02
Name	<b>Calibrate Sensors</b>
Brief description	By pushing the <i>button C</i> the calibration process is started.
Preconditions	The <i>System</i> is initialised.
Postconditions	The line sensor is calibrated to be able to detect the <i>track</i> .
Failure scenarios	The sensor cannot be calibrated.
Actors	<i>push button C</i> , line sensor
Trigger	The <i>push button C</i> is pressed.
Standard workflow	1. The push <i>button C</i> is pressed. 2. The line sensor is calibrated.
Alternative workflow	

Identifier	S03
Name	Mode Switch
Brief description	With push button B the software parameters can be switched between the parameter sets <i>highspeed mode</i> , <i>safety mode</i> or <i>balanced mode</i> .
Preconditions	The Software isn't in the use case "Run Race".
Postconditions	The desired parameter set is selected.
Failure scenarios	
Actors	<i>Push button B</i> , <i>OLED</i>
Trigger	<i>Push button B</i> is pressed.
Standard workflow	1. <i>Push button B</i> is pressed. 2. The <i>system</i> loads parameters. 3. The <i>race mode</i> is shown on the <i>OLED</i> .
Alternative workflow	2a. The <i>system</i> fails loading parameters.

Identifier	S04
Name	output information
Brief description	The software outputs information through the display and the buzzer.
Preconditions	The software isn't in the use case "Run Race".
Postconditions	The software outputs information through the display or the buzzer.
Failure scenarios	The display doesn't show information or the buzzer doesn't output a signal.
Actors	display, buzzer
Trigger	<ol style="list-style-type: none"> <li>1. The system is powered on.</li> <li>2. The <i>system</i> is in an error case.</li> <li>3. The <i>system</i> finished use case "Run Race" S05.</li> </ol>
Standard workflow	<ol style="list-style-type: none"> <li>1. The software is powered on.</li> <li>2. The display shows the team name and the selected <i>driving mode</i>.</li> <li>3. The software is in use case "Run Race" S05.</li> <li>4. The software is in use case "Race Done" S09.</li> <li>5. The display shows the race time.</li> </ol>
Alternative workflow	5a. The display fails showing the race time.

Identifier	S05
Name	Run Race
Brief description	The <i>system</i> controls the robot to follow the line in the selected <i>driving mode</i> .
Preconditions	Startline was detected.
Postconditions	The <i>system</i> executes use case “Race Done” S09.
Failure scenarios	The software fails controlling the robot following the <i>track</i> , so the robot leaves the <i>track</i> .
Actors	line sensor, motor
Trigger	“Start Race” S08 is finished
Standard workflow	<ol style="list-style-type: none"><li>1. “Start Race” S08 is finished</li><li>2. The line sensor detects the <i>track</i>.</li><li>3. The software controls the motor.</li><li>4. The software is in use case “Race Done” S09.</li></ol>
Alternative workflow	<ol style="list-style-type: none"><li>2a. The line sensor doesn't detect the <i>track</i>.</li><li>3a. The software is in use case “Handle Error” S06.</li></ol>

Identifier	S06
Name	Handle Error
Brief description	If an error occurs the <i>system</i> must handle it.
Preconditions	The <i>system</i> must detect an error.
Postconditions	The <i>system</i> stops the motor. The <i>system</i> outputs that an error occurred through the display and the buzzer.
Failure scenarios	The <i>system</i> fails detecting the error.
Actors	motor, buzzer, display
Trigger	The <i>system</i> detects an error.
Standard workflow	1. The <i>system</i> detects an error. 2. The <i>system</i> stops the motor. 3. The <i>system</i> outputs that an error occurred on the OLED. 4. The <i>system</i> will beep for 0,33s, pause for 0,33s, then beep again for 0,33s. 5. The <i>system</i> will be restarted.
Alternative workflow	2a. The <i>system</i> fails stopping the motor. 3a. The <i>system</i> fails displaying that an error occurred. 4a. The <i>system</i> fails putting out the beep tone.

Identifier	S07
Name	<b>Redetect <i>Track</i></b>
Brief description	In case there is a gap in the line or the line isn't straight forward, the system must redetect the <i>track</i> .
Preconditions	The line sensor can't detect the <i>track</i> .
Postconditions	The line sensor redetects the <i>track</i> . The software is in the use case "Run Race" S05 again.
Failure scenarios	The line sensor cannot redetect the <i>track</i> .
Actors	line sensor, motor
Trigger	The line sensor can't detect the <i>track</i> .
Standard workflow	<ol style="list-style-type: none"> <li>1. The software is in the use case "Handle Error".</li> <li>2. The line sensor doesn't detect the track.</li> <li>3. The software controls the motor to turn the robot left and right to redetect the line.</li> <li>4. The software redetects the line.</li> <li>5. Back to "Run Race" S05.</li> </ol>
Alternative workflow	<ol style="list-style-type: none"> <li>4a. The software cannot redetect the line within 5s.</li> <li>5a. The software stops the motor.</li> <li>6a. The software is in use case "Handle Error" S05.</li> </ol>

Identifier	S08
Name	Start Race
Brief description	The <i>system</i> controls the robot to start the race.
Preconditions	The <i>system</i> is powered on. The line sensor is calibrated. The <i>driving mode</i> is selected.
Postconditions	The <i>system</i> executes use case "Run Race" S05.
Failure scenarios	The software fails detecting the <i>start line</i> .
Actors	<i>push button A</i> , line sensor, motor, buzzer
Trigger	<i>push button A</i>
Standard workflow	<ol style="list-style-type: none"> <li>1. The push button A is pressed.</li> <li>2. The <i>system</i> waits for 3s.</li> <li>3. The <i>system</i> controls the motor to drive forward.</li> <li>4. The <i>system</i> detects the <i>start line</i>.</li> <li>5. The <i>system</i> outputs a short beep through the buzzer for 1s.</li> <li>6. The <i>system</i> starts measuring the time.</li> <li>7. The <i>system</i> executes use case "Run Race" S05.</li> </ol>
Alternative workflow	<ol style="list-style-type: none"> <li>2a. The <i>system</i> fails the time measurement.</li> <li>3a. The <i>system</i> executes use case "Handle Error" S06.</li> <li>3b. The <i>system</i> fails detecting the <i>start line</i>.</li> <li>4b. The <i>system</i> executes use case "Handle Error" S06.</li> </ol>



Identifier	S09
Name	Race Done
Brief description	The <i>system</i> controls the robot to end the race.
Preconditions	The <i>system</i> is powered on. The <i>system</i> executes use case “Run Race” S05.
Postconditions	The <i>system</i> displays the measured time.
Failure scenarios	The software fails detecting the <i>stop line</i> .
Actors	line sensor, motor, display, buzzer
Trigger	The <i>stop line</i> is detected.
Standard workflow	<ol style="list-style-type: none"> <li>1. The <i>system</i> detects the <i>stop line</i>.</li> <li>2. The <i>system</i> controls the motor to stop.</li> <li>3. The <i>system</i> stops measuring the time.</li> <li>4. The <i>system</i> outputs the measured time through the display.</li> <li>5. The <i>system</i> outputs a short beep through the buzzer for 1s.</li> </ol>
Alternative workflow	<ol style="list-style-type: none"> <li>1a. The <i>system</i> fails detecting the <i>stop line</i>.</li> <li>2a. The <i>system</i> fails controlling the motor to stop.</li> <li>3a. The <i>system</i> fails stopping measuring the time.</li> <li>4a. The <i>system</i> fails putting out the measured time through the display.</li> <li>5a. The <i>system</i> fails putting out a short beep through the buzzer.</li> </ol>