Software requirement specification (SRS) document template



Review history



Approval history

Project name: Date: Version:

By: Kevin Tran



Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Version description | Date completed |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Approving party | Version approved | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Reviewer | Version reviewed | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of contents



1. Introduction
   1. Product scope
   2. Product value
   3. Intended audience
   4. Intended use
   5. General description

2 Functional requirements

1. External interface requirements
   1. User interface requirements
   2. Hardware interface requirements
   3. Software interface requirements
   4. Communication interface requirements
2. Non-functional requirements
   1. Security
   2. Capacity
   3. Compatibility
   4. Reliability
   5. Scalability
   6. Maintainability
   7. Usability
   8. Other non-functional requirements

5 Definitions and acronyms



1

Introduction

Describe the purpose of the document.

* 1. Product scope

List the benefits, objectives, and goals of the product.

The main objective of my project is to deliver a prototype of a maze solving robot, which will follow walls, avoid collisions and detect ‘victims’ with a colour sensor beneath it. It will use the components of a Pico Pi with OOP code, servos, ultrasonic sensors, a line following sensor, and a colour sensor. However, if this project is realized as a real product, it will have to have the ability to navigate within a warehouse setting and stock shelves.

* 1. Product value

Describe how the audience will find value in the product.

The value of the finalized robot will come in the automation of repetitive tasks such as stocking a warehouse and ease of programming and use, which will be accounted for when programming the robot.

* 1. Intended audience

Write who the product is intended to serve.

My robot, if it is fully developed, is intended for owners of industrial companies like Amazon, or other businesses that utilize warehouses often. This is because the robot will assist in moving and stocking products without human intervention, potentially speeding up the process and reducing costs.

* 1. Intended use

Describe how will the intended audience use this product.

The prototype is intended to navigate a maze without colliding into walls, but the developed robot is intended for use in an industrial warehouse environment, navigating in the isles to the specified location and delivering the stock it was loaded with.

* 1. General description

Give a summary of the functions the software would perform and the features to be included.

The functions that the prototype would perform would include moving along a wall, avoiding obstacles and potentially detecting colours beneath itself to perform certain actions such as reporting a location.

# Design diagrams

**UML Class Diagram**

A diagram of a company

AI-generated content may be incorrect.

**Data Flow Diagrams**

DFD 0:

A diagram of a system

AI-generated content may be incorrect.

DFD 1:

A diagram of a system

AI-generated content may be incorrect.

**Flowchart**

A diagram of a flowchart

AI-generated content may be incorrect.

**Wiring diagram**

A diagram of a circuit

AI-generated content may be incorrect.

# Functional requirements



2

List the design requirements, graphics requirements, operating system requirements, and constraints of the product.

The functional requirements of my prototype are as follows:

Design requirements:

* Solve a maze within a reasonable amount of time
  + Follow a wall
  + Avoid contact with obstacles or walls
* Detect ‘victims’ with a colour sensor
* Display its current state on a screen for the user

Graphics requirements:

* LCD screen that displays the robot’s current state
* LEDs

Operating system requirements

* Pico Pi
* OOP code in Micropython

Constraints

* Components (using what I was given)

**Material components list**

The materials and components that I used in my prototype were:

Materials:

* Wooden chassis
* 2x wheels
* 1x omnidirectional wheel
* 1x battery pack
* 4x male to male wires
* 4x male to female wires

Components:

* 1x Raspberry Pi Pico 2
* 2x 3.7V batteries
* 2x DFrobot DF15RSMG servos
* 2x ultrasonic sensors
* 1x colour sensor
* 1x OLED screen
* 3x diodes
* 1x fuse
* 1x de amplifier
* 2x capacitors
* 2x polarized capacitors

**Power supply calculations**

|  |  |
| --- | --- |
| **Input/output** | **Volts** |
| Battery pack (2 3.7V batteries) | 7.4V |
| 2 capacitors (~0.5V down each) | 7.4V -> 6.4V |
| De amplifier (regulated to 5V) | 6.4V -> 5V |
| Servos | 5V received |

7.4V from batteries -> 5V into servo

External interface requirements



3

* 1. User interface requirements

Describe the logic behind the interactions between

the users and the software (screen layouts, style guides, etc).

The UI elements of the LCD screen and LEDs are there to show the user what state the robot is currently in, for example if it was currently moving, turning or stopping. These are there to add further clarity when making the robot fit the user’s specific needs.

* 1. Hardware interface requirements

List the supported devices the software is intended

to run on, the network requirements, and the communication protocols to be used.

The robot is designed to run on a Pico Pi, with an LCD screen and LEDs as the user interface. The LCD will be using the I2C communication protocol.

* 1. Software interface requirements

Include the connections between your product and other software components, including frontend/backend framework, libraries, etc.

The overall programming of the prototype will include connections between different libraries for each specific component, which will be the sensors,

* 1. Communication interface requirements

List any requirements for the communication programs your product will use, like emails or embedded forms.

The only communication program that the robot prototype will use is I2C, which will serve as the communication link between the Pico Pi and the LCD screen, and it will transfer the state the robot is in currently and display that.

# Non-functional requirements



4

* 1. Security

Include any privacy and data protection regulations that should be adhered to.

* 1. Capacity

Describe the current and future storage needs of your software.

The amount of storage that the prototype will use is most likely under 1MB for the program, which will fit on the 2MB of storage that the Pico Pi has. In the future, the robot may need more storage for advanced navigation or other functions.

* 1. Compatibility

List the minimum hardware requirements for your software.

The minimum hardware requirements for the robot would be:

* Pico Pi
* 2 servos
* 1 colour sensor
* 2 ultrasonic sensors
  1. Reliability

Calculate what the critical failure time of your product would be under normal usage.

* 1. Scalability

Calculate the highest workloads under which your software will still perform as expected.

* 1. Maintainability

Describe how continuous integration should be used to deploy features and bug fixes quickly.

Continuous integration will be used to deploy features and fix bugs quickly in the prototype by using Github to commit changes to the cloud, which allows the code to be accessed and edited from anywhere.

* 1. Usability

Describe how easy it should be for end-users to use your software.

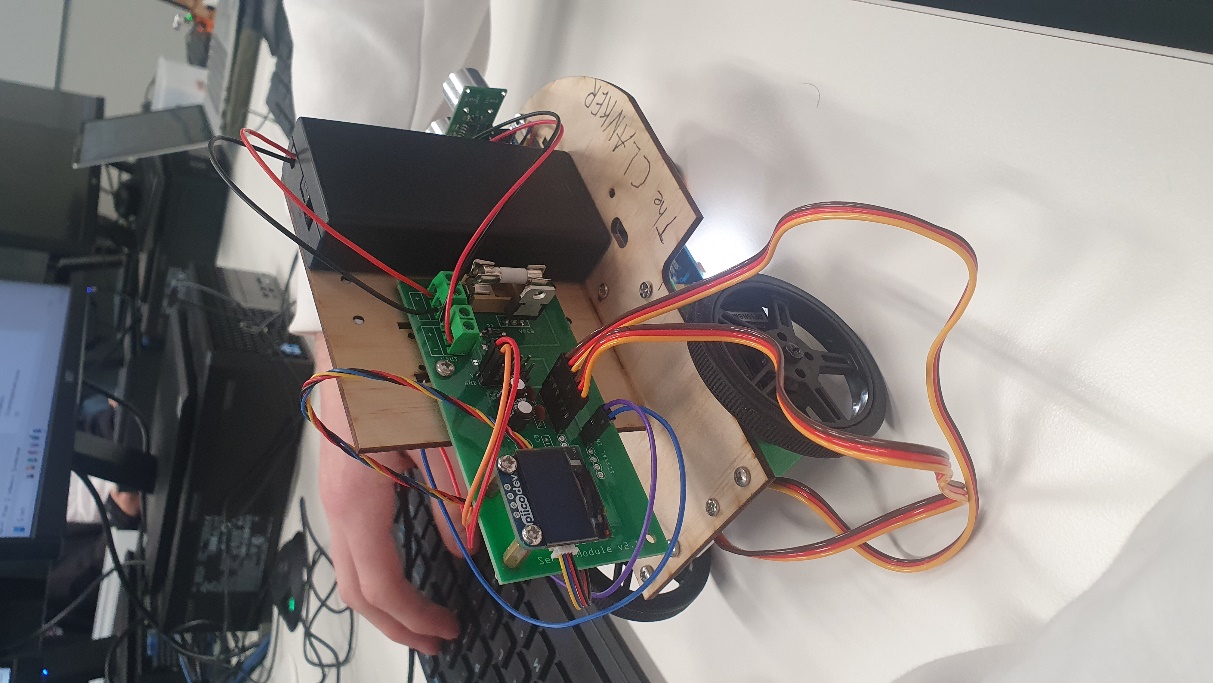
End users should be able to configure the prototype to their needs quite easily, as the code will be organized with understandable variable names and documentation for each library, and they will not need to change any of the wiring so long as they built it the same.

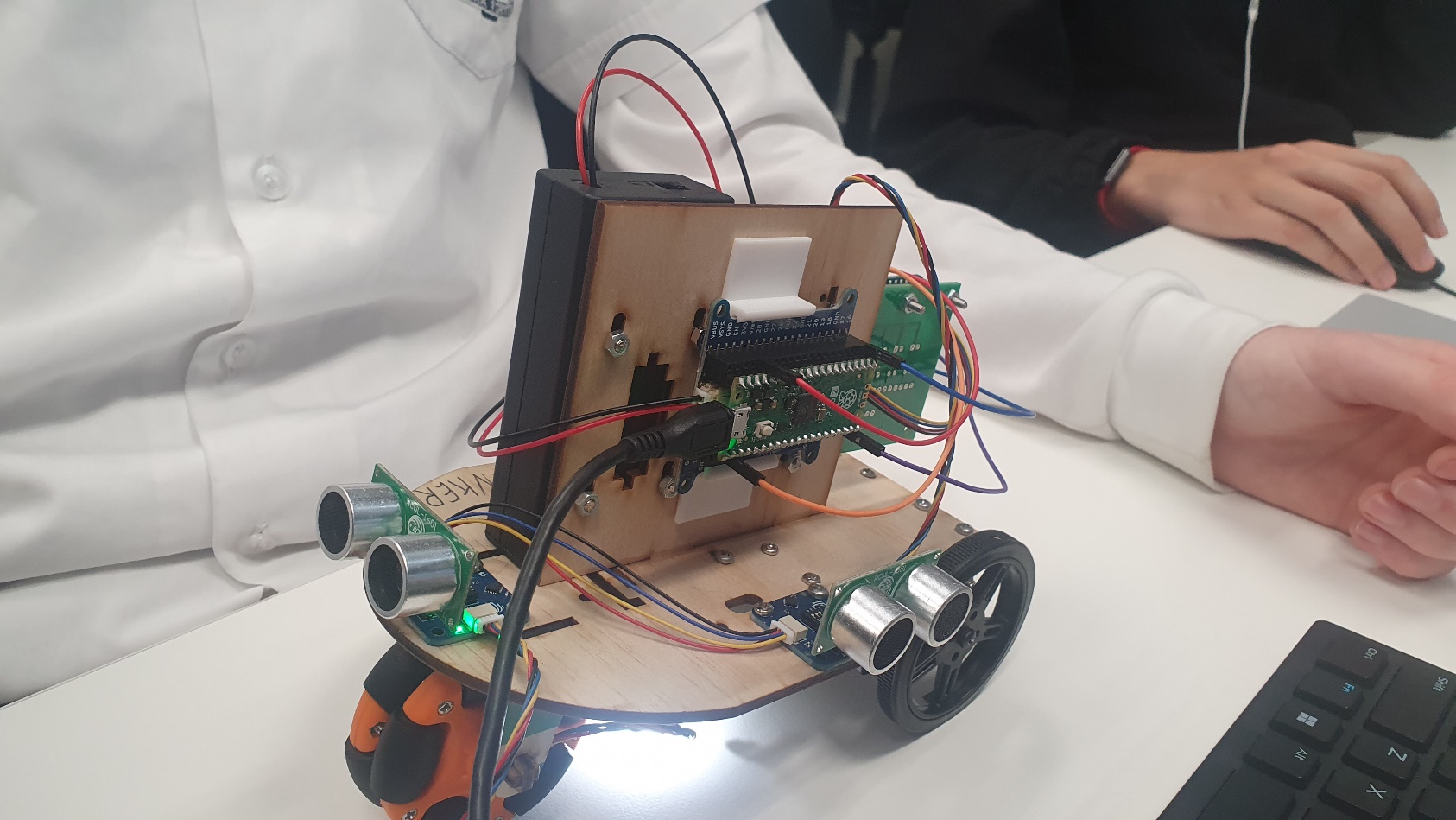
* 1. Other

List any additional non-functional requirements.

# Producing and implementing

Finished prototype:

A hand holding a device

AI-generated content may be incorrect.

# Testing and evaluating

**Unit testing:**

**Justification of techniques:**

Definitions and acronyms



5

|  |  |
| --- | --- |
| Continuous integration (CI) | A software development practice where developers frequently integrate their code changes into a shared repository. |
| UI | An abbreviation for user interface, which is the part of the program that the user interacts with |
| OOP | An abbreviation for object-oriented paradigm, which is a software design method that organizes the software around objects rather than functions |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |