VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
INSTITUTE OF COMPUTER SCIENCE
INFORMATION TECHNOLOGIES STUDY PROGRAM

Problem-based Project

Requirements Specification

Group name : **Rover**
Done by: : **Nedas Janušauskas,**
Adomas Jonavičius,
Aistė Grigaliūnaitė,
Gabrielius Drungilas
Supervisor: : **Linas Bukauskas**

Vilnius

2022

Contents

1 Introduction

1.1 Purpose of this document

This is a Requirements Specification document for an autonomous robot companion. The robot companion will move around a room, detect obstacles, avoid detected obstacles, map the room, and inform the user of errors. This document is intended to direct the design and implementation of the robot.

1.2 Scope of this document

The scope of this document is to:

- Provide a high-level overview of the project.
- Specify the functional requirements.
- Specify the non-functional requirements.
- Outline an implementation plan.

2 Use Case

The robot will be turned on with a push of a button located on the robot, then it will start to move around the room. It will scan the environment using two ultrasound distance sensors and two USB cameras. It will detect obstacles, avoid them, and will remember the layout of the room. If stuck, it will alert the user with a sound signal using a beeper. It will have the ability to be controlled and receive tasks using a remote controller.

3 Functional Requirements

3.1 High Priority

- The robot will have tracks and will be able to move around the room, which includes moving forwards, backwards, and turning left and right.
- The robot will decide how to move by itself, with a user-given task.
- The robot will be able to detect obstacles (objects in the robot's path – furniture, walls, etc.) that are in front of it and behind.
- The robot will make decisions on where to move when an obstacle is detected.
- The robot can be turned off with a power switch.

3.2 Medium Priority

- The robot will remember the layout of the room.
- The robot will be able to inform the user about navigation errors with sound signals.

3.3 Low Priority

- The robot will be able to recognise and follow the user defined objects.

4 Non-Functional requirements

4.1 Reliability

- The robot will be operational until turned off unless it runs out of power.
- The robot will have an expected battery life of 2 hours.

5 Implementation Plan

5.1 Version Plan

Version	Features
0.1	The robot can move forwards and backward in a pre-programmed path.
0.2	The robot can move around the room – forwards, backward, and can turn.
0.3	The robot can detect surroundings using cameras and sensors.
0.4	The robot can avoid obstacles it detects.
0.5	The robot can produce sound signals.
0.6	The robot can map and remember its surroundings.
0.7	The robot can follow objects and/or people.

5.2 Tools Used

Software used on the Raspberry Pi

- Linux OS - OS that will be run on the Raspberry Pi.
- Python - code for the robot's functionality.
- C++ - code for hardware tests.
- OpenCV - libraries used for computer vision.
- Bash - used to execute any scripts required on startup.

Software used for development

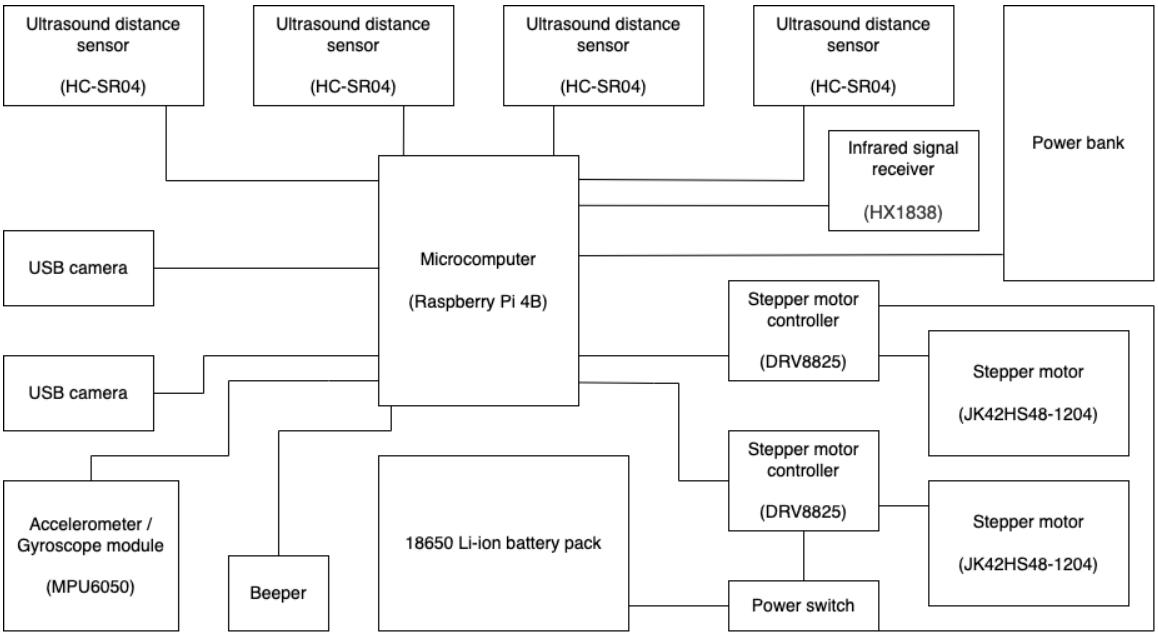
- Git and GitLab - version control.
- Webots - robot simulation environment.
- VSCode - development environment.

Miscellaneous software

- AutoCAD - 3D modeling.
- 3D printer - manufacturing of robot body parts.

5.3 Hardware Diagram

The robot's hardware is represented in the following diagram:



5.4 Hardware Components

1. Raspberry Pi 4 Model B (4GB RAM version)
2. (4) HC-SR04 ultrasound distance sensors
3. (2) USB cameras
4. (2) NEMA 17 stepper motors
5. (2) DRV8825 stepper motor controllers
6. Power switch
7. Power bank
8. (4) 18650 batteries
9. 18650 battery holder
10. (4) 1kΩ resistors
11. (4) 2kΩ resistors
12. (2) 100 μF capacitors
13. Beeper
14. 32GB MicroSD card
15. Remote controller
16. Accelerator/gyroscope