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

Problem-Based Project

Technical specification

Done by:

Gabrielius Drungilas

Aistė Grigaliūnaitė

Nedas Janušauskas

Adomas Jonavičius

Supervisor:
dr. Linas Bukauskas

Vilnius
2022

Contents

1 Overview

1.1 Project overview

The team has made a lot of progress on the project. The main components of the robot have been assembled and the team is now working on the code that will allow the robot to function as desired. The team is currently working on the code that will allow the robot to navigate its surroundings and avoid obstacles. The team is also working on the code that will allow the robot to communicate with the user.

1.2 Hardware diagram

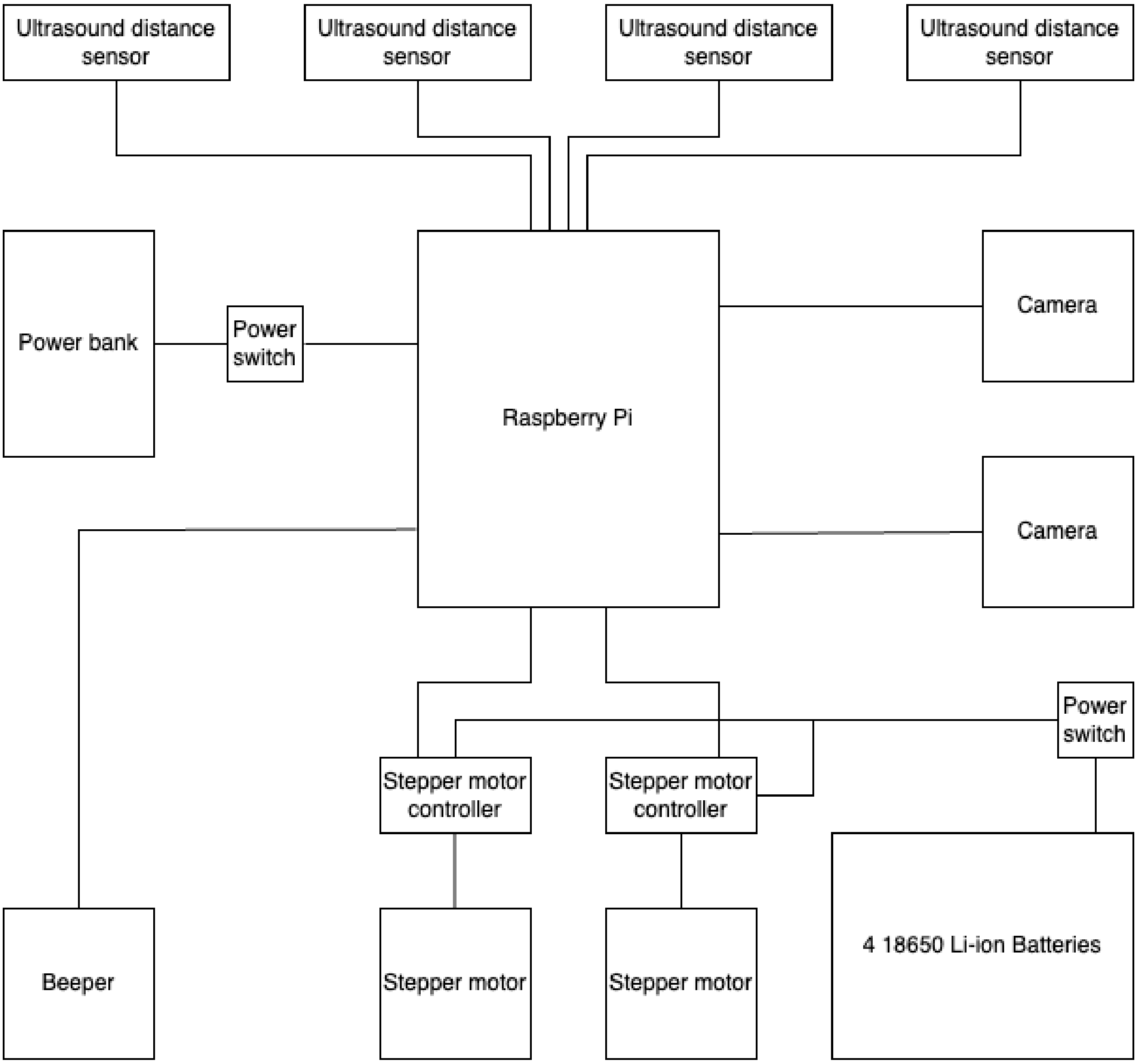

Figure 1. Hardware diagram

1.3 Context diagram

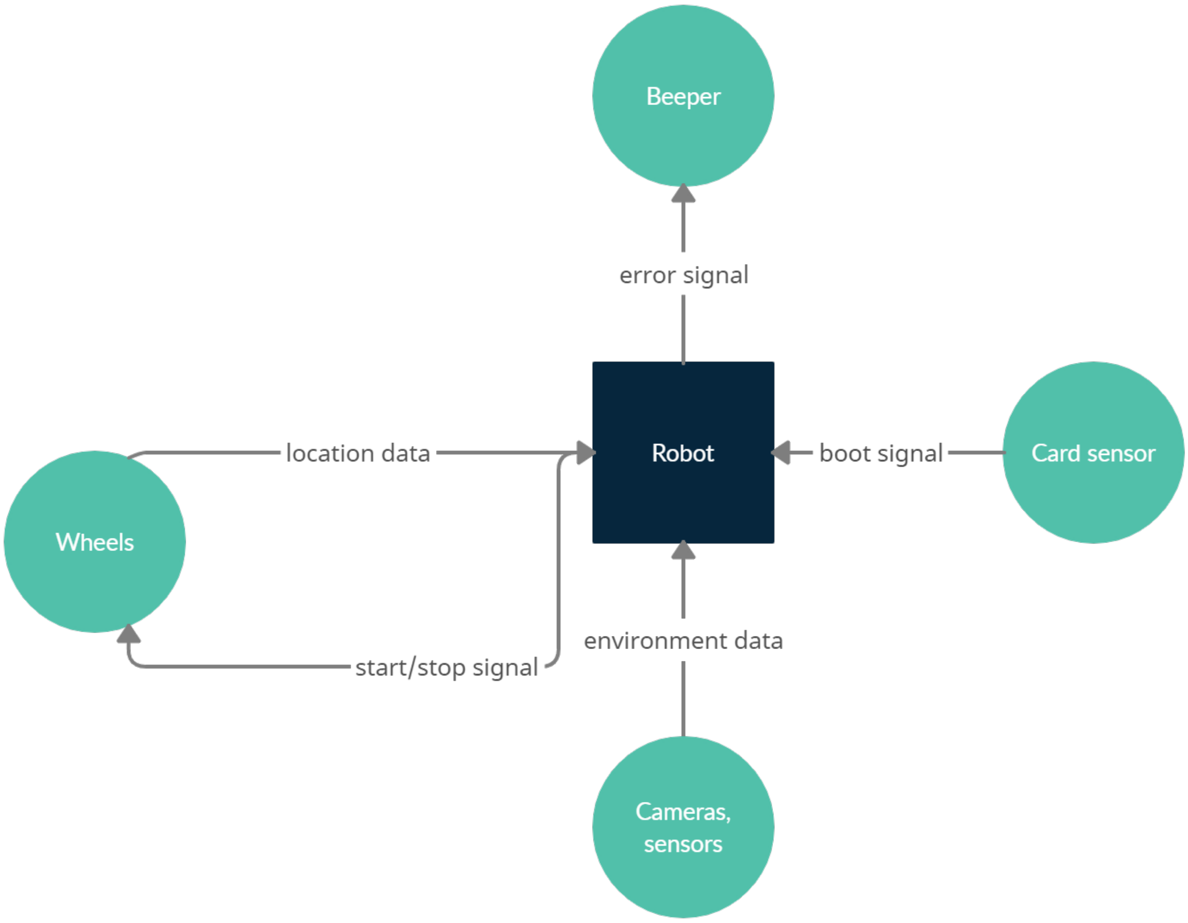

Figure 2. Context diagram

1.4 UML deployment diagram

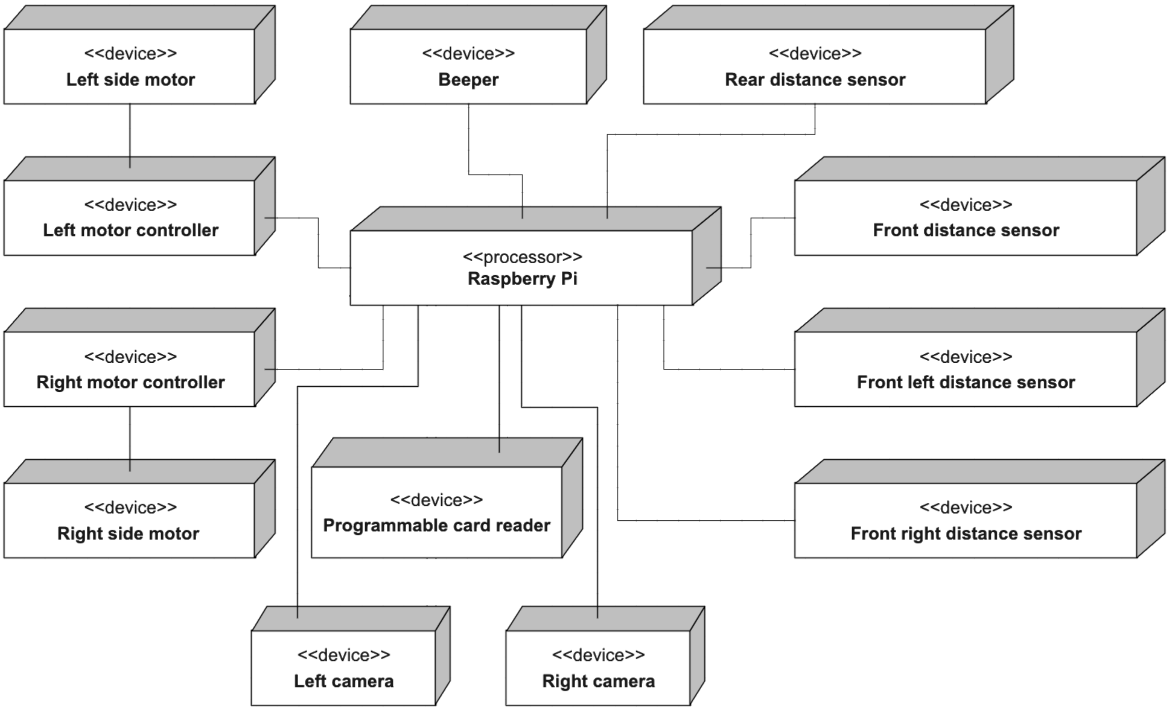

Figure 3. Deployment diagram

2 Deliverable internals

2.1 Structural aspects

The main components of the robot are the raspberry pi, cameras and sensors, wheels, and the beeper. The cameras and sensors will be used to detect the robots surroundings, the wheels will move the robot, as well as transmit location data to the raspberry pi, the beeper will be used to warn the user of errors, and the raspberry pi will be used to execute the required code.

2.2 Dynamic aspects

In order to accomplish the desired functionality, the robot's components will send data to the raspberry pi. The card sensor, when activated, will add a task to the robot's task queue. The stepper motors will make it possible for the robot to keep track of its location in the room, as well as move a specified distance.

The internal logic of USB cameras: Creating the "camera" class as an extension of the thread class to be able to keep cameras working together at the same time. Create a separate thread that will compare the view of the different cameras and calculate the disparity between different areas of the picture. Then, take the output of the disparity calculation, analyze if the path ahead is clear, and use it to update the 2D array representing the known map of the surrounding area. Then, calculate the next move the robot should make to get to the goal.

The known map will hold all data about the surrounding area: Each element in the array will represent the state associated with a particular area. It can hold states like Unknown, current rover position, obstacle, human, destination, starting position, and any additional useful info that could impact the decision-making of the rover.

3 Testing

The robot is tested using two methods:

1. Manually in an indoor environment. This is a reliable way to test the functionality of the code.

2. Using simulation software "Webots". This makes testing more accessible by sharing the code in a simulation environment.

4 Technologies and Tools

The following technologies will be used to build and test the robot, as well as maintain the required code:

- Linux OS
- Python
- BASH
- Git
- OpenCV
- Webots
- GitLab
- VSCode
- AutoCAD
- 3D printer