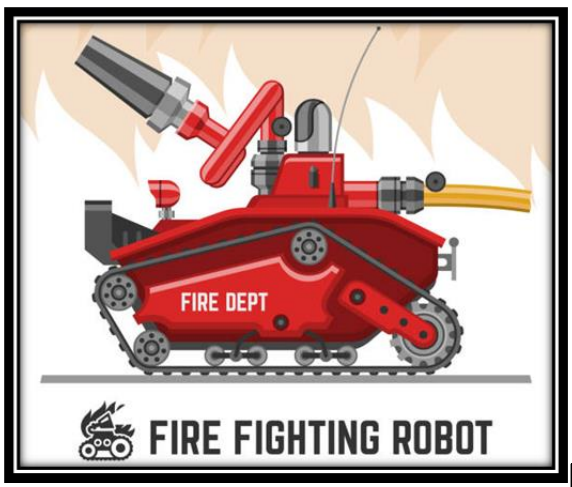
SYSC3010
Computer Systems Development Project

Household Fire Warden and Extinguisher Robot

Detailed Design



[1]

Group L3-G3

Bren-Gelyn Padlan, 101148482
Eline Elorm Nuviadenu, 101162869
Hiu Sum Jaime Yue, 101162929

TA: Roger Selzler

March 08, 2022

Table of Contents

1 Problem Statement

Oftentimes, firefighters are called on-site only to realize that there was little or no serious fire. This project aims to reduce the workload of firefighters by creating a reliable firefighting robot to handle small household fires called Household Fire Warden and Extinguisher Robot, or FiWER. This robot would be remotely controlled using a mobile android application. It would also be able to detect the presence of fire using temperature and infrared flame sensors, and to extinguish the fire by spraying water using water pumps.

1.1 Functional Requirements

Table 1 below describes the functional requirements of the FiWER and their corresponding test scenarios that would be used for testing later in the project.

Table 1. Functional requirements of the FiWER and their test scenario

Functional Requirement	Test Scenario
Can be controlled remotely	User attempts to extinguish fire while in another room
Able to spray water	User simply presses the spray water button on the mobile application
Able to detect temperature and fire	A lighter is brought close to the robot. The database is monitored for change in temperature and the mobile application is monitored for a push notification.
Provides live streaming for robot’s vision	A user who is not in the same room as the robot, uses the VNC (Virtual Network Computer) to see what is happening around the robot.

2 Design Overview

2.1 System Overview Diagram

The UML deployment diagram is shown in Figure 1 below. It shows the basic system overview of the FiWER where the three main nodes are the Robot, Firebase and Phone. The project was approached using this design since this design of Robot, Firebase and Phone communicating with each other through the cloud meets the project requirement of controlling the robot remotely. Additionally, having a Raspberry Pi and Arduino sub-nodes inside the Robot node allows for the fire detection and extinguisher feature of the robot.

The Robot consists of three sub-nodes namely Raspberry Pi (RPi), Arduino, and SenseHat. The RPi communicates to the Arduino via Serial communication. The Arduino operates the DC motors using a motor driver, servo motors, and water pumps of the robot and sends the status of these components to the RPi when requested. In the diagram below, the GPIO pins for the motors and pumps are shown. The RPi, on the other hand, is responsible for reading the data from the sensors (IR flame sensor and Pi Camera) and communicating with the Arduino to send commands to control the motors and pumps. The RPi also sends the sensor data to Firebase through the cloud via HTTP communication. The third sub-node is the SenseHat which contains the temperature sensor that is also used for fire detection. The SenseHat communicates with the RPi via I2C.

The Firebase contains the database for the robot which consists of the sensor and motor (DC and servo) data. It is connected to the cloud via HTTP. A single database table would be used to hold all the information gathered from Robot. The sample database table is presented and explained in Section 2.4.

The Phone contains the mobile application that the user interacts with to control the robot. This mobile application reads the sensor and motor data stored in the Firebase database through the cloud via HTTP communication. Inside this application, the user is able to control the movement – direction and speed – of the robot, the pan-tilt of the Pi Camera, and the extinguisher feature (water pump) to put out the fire.

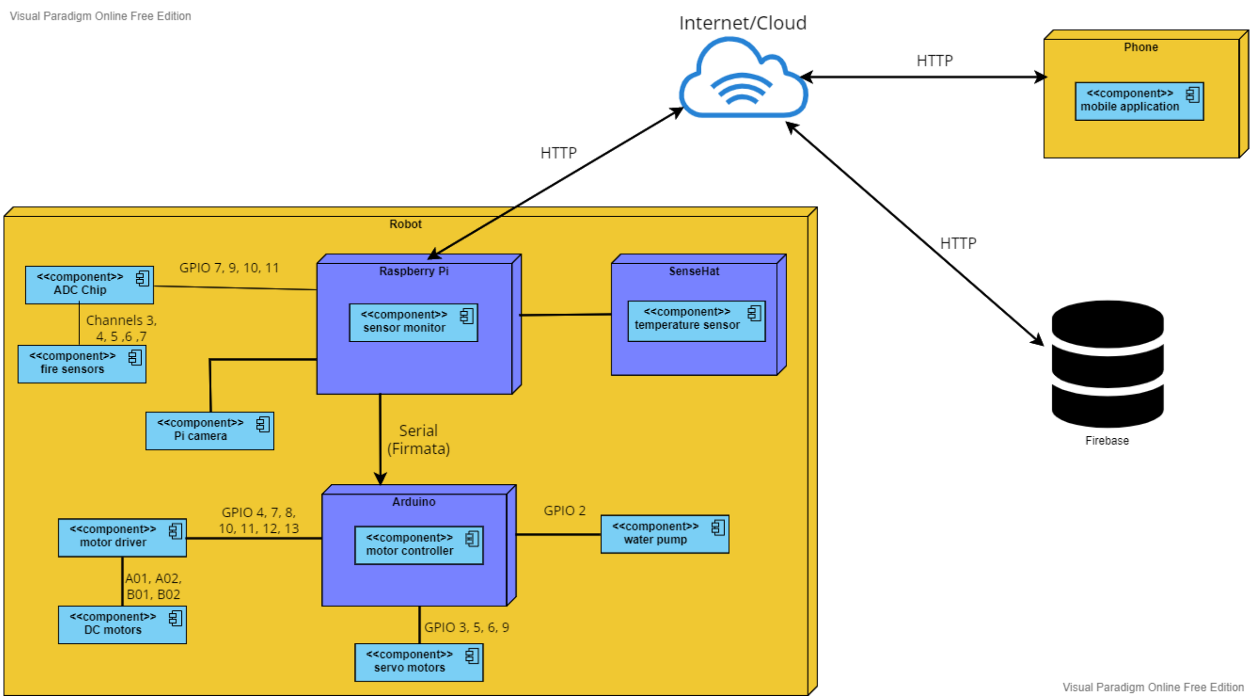


Figure 1: Deployment diagram of the FiWER project [Bren-Gelyn Padlan]

2.2 Communication Protocols

For the three main nodes (Robot, Firebase and Phone) shown in the deployment diagram in Figure 1, Hypertext Transfer Protocol (HTTP) would be used as the communication protocol between these nodes.

Inside the Robot, the Raspberry Pi would use a serial communication protocol called Firmata to communicate to the Arduino. The motors and pumps would communicate to the Arduino via GPIO pins. Some GPIO pins with Pulse Width Modulation (PWM) would be used for the previously mentioned components which would need to be programmed. In terms of the sensors, the SenseHat communicates with the Raspberry Pi via Inter-Integrated Circuit (I2C) connection while the infrared flame sensors are connected to the RPi via Serial Peripheral Interface (SPI) communication.

2.2.1 Communication Protocol Table

Figure 2 below shows the messages that can be sent and received between each node-node interface. There are four node-node interfaces, which are Mobile and Database, Robot and Database, SenseHat and Raspberry Pi, Raspberry Pi, and Arduino.

Mobile and Database			
Sender	Receiver	Message	Data Format
Mobile	Database	tempSet	36
Database	Mobile	showTemp	35
Database	Mobile	getFireStatus	1505
Mobile	Database	robotIdle	"Idle"
Mobile	Database	robotMoveForward	"MoveForward"
Mobile	Database	robotMoveBackward	"MoveBackward"
Mobile	Database	cameraMoveLeft	"MoveLeft"
Mobile	Database	cameraMoveRight	"MoveRight"
Mobile	Database	waterPumpOn	"WaterPumpOn"
Mobile	Database	cameraPanLeft	"PanLeft"
Mobile	Database	cameraPanRight	"PanRight"
Mobile	Database	cameraTiltUp	"TiltUp"
Mobile	Database	cameraTiltDown	"TiltDown"

Robot and Datebase			
Sender	Receiver	Message	Data Format
Robot	Database	sendFireSensorData	1006
Robot	Database	sendTemp	55
Robot	Database	determineFireStatus	"Serious"
Database	Robot	robotIdle	"Idle"
Database	Robot	robotMoveForward	"MoveForward"
Database	Robot	robotMoveBackward	"MoveBackward"
Database	Robot	RobotMoveLeft	"MoveLeft"
Database	Robot	RobotMoveRight	"MoveRight"
Database	Robot	waterPumpOn	"WaterPumpOn"
Database	Robot	cameraPanLeft	"PanLeft"
Database	Robot	cameraPanRight	"PanRight"
Database	Robot	cameraTiltUp	"TiltUp"
DataBase	Robot	cameraTiltDown	"TiltDown"

SenseHat and Rpi			
Sender	Receiver	Message	Data Format
SenseHat	Rpi	tempChange	36

Inside the robot

Rpi and Arduino			
Sender	Receiver	Message	Data Format
Rpi	Arduino	robotIdle	"Idle"
Rpi	Arduino	robotMoveForward	"MoveForward"
Rpi	Arduino	robotMoveBackward	"MoveBackward"
Rpi	Arduino	cameraMoveLeft	"MoveLeft"
Rpi	Arduino	cameraMoveRight	"MoveRight"
Rpi	Arduino	waterPumpOn	"WaterPumpOn"
Rpi	Arduino	cameraPanLeft	"PanLeft"
Rpi	Arduino	cameraPanRight	"PanRight"
Rpi	Arduino	cameraTiltUp	"TiltUp"
Rpi	Arduino	cameraTiltDown	"TiltDown"

Figure 2: Communication Protocol Table between nodes [Hiu Sum Jaime Yue]

2.3 Message Sequence Diagram(s)

2.3.1 Message Sequence Diagram 1: Detect Fire

The detect fire use case refers to the basic work of the robot. In this use case, the robot detects the temperature every few minutes and sends the data to the Firebase. The phone, through the mobile application, will get the temperature and the fire status from the Firebase. When the fire status says "Warning" or "Serious", it will alert the user about the fire status. Figure 3 illustrates the sequence diagram for this use case.

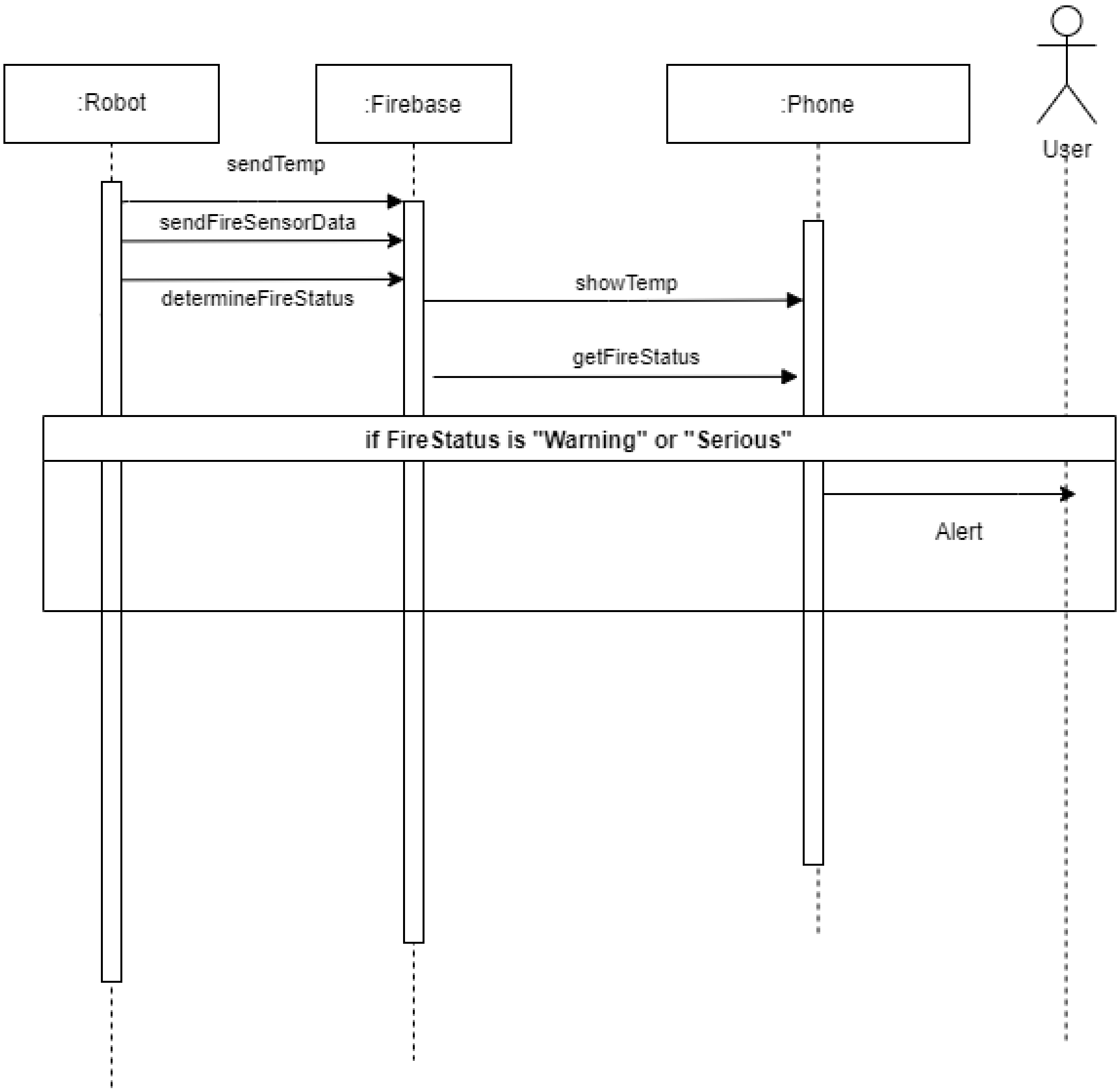


Figure 3: Message sequence diagram for use case 1: Detect Fire [Hiu Sum Jaime Yue]

2.3.2 Message Sequence Diagram 2: Extinguish Fire

The extinguish fire use case refers to the scenario where the robot has detected a fire and needs to perform actions to extinguish it before it becomes uncontrollable. This use case describes the steps that the robot would follow in order perform this task. To perform this task, the user will be controlling the robot. There is no order of how the user wants the robot to perform. The user can choose from moving the robot, adjusting the camera and using the water pump to extinguish the fire. More details about this use case can be seen in Figure 4 below as it shows a sequence diagram of the extinguish fire use case.

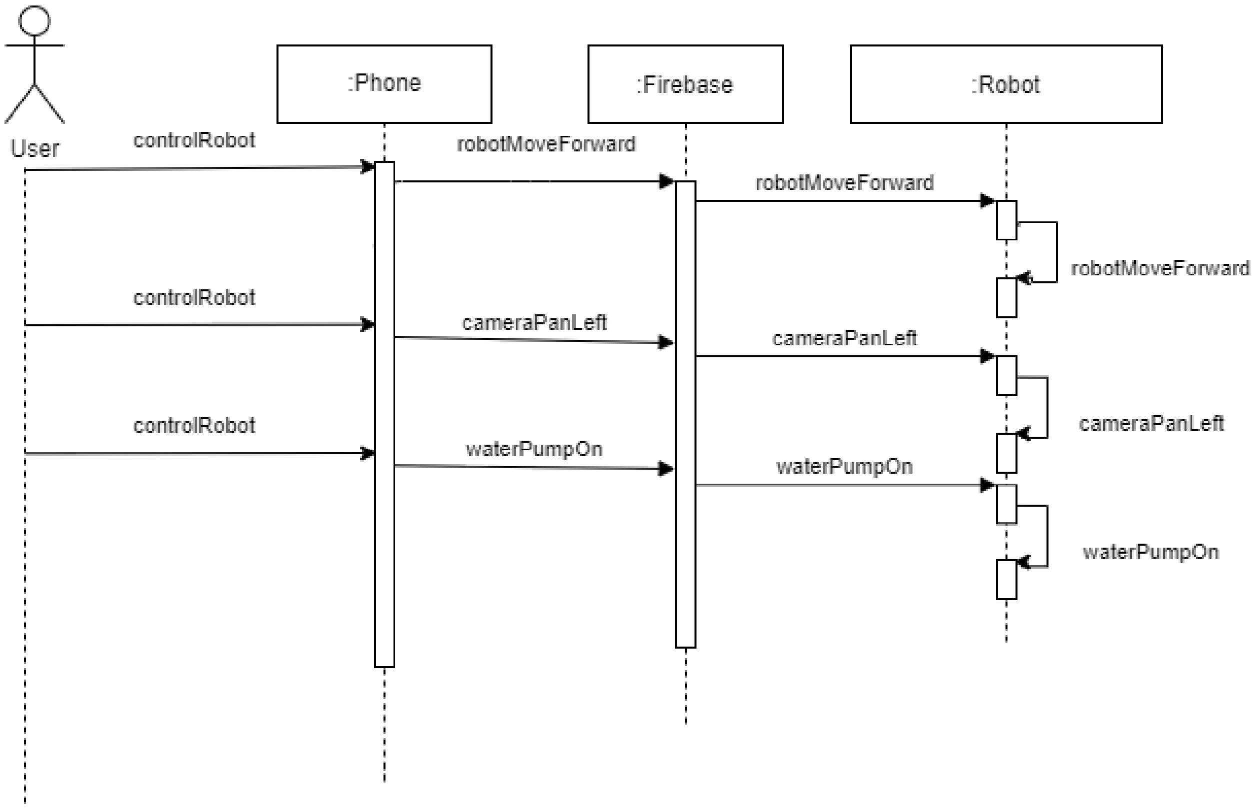


Figure 4: Message sequence diagram for use case 2: Extinguish Fire [Hiu Sum Jaime Yue]

2.4 Database Table Design/Schema

The structure of the database is shown in Figure 5 below. The database uses one index, which is the log. There are 6 different pieces of information inside the log. It includes Timestamp, Temperature, FireSensor, SetRoomTemp, Fire Status and Motor Status. In this index, only 1 user is interacting with the robot.

Log					
Timestamp	Temperature	FireSensor	SetRoomTemp	Fire Status	Motor Status
14:30:00	36	1200	25	"No"	"Idle"
14:32:00	52	1050	25	"No"	"Idle"
14:34:00	70	850	25	"Warning"	"MoveForward"
14:34:01	71	802	25	"Warning"	"TiltDown"

Fire Status List	Motor Status List
"No"	"Idle"
"Warning"	"MoveForward"
"Serious"	"MoveBackward"
	"MoveLeft"
	"MoveRight"
	"WaterPumpOn"
	"PanLeft"
	"PanRight"
	"TiltUp"
	"TiltDown"

Figure 5: Sample SQL table [Hiu Sum Jaime Yue]

2.5 Software Design for Raspberry Pi

The Raspberry Pi is programmed using the Python programming language. Code would be written to control the SenseHat temperature sensor, the flame sensors, and the Pi camera. Figure 6 below is the UML Diagram of the layout of the Raspberry Pi software design.

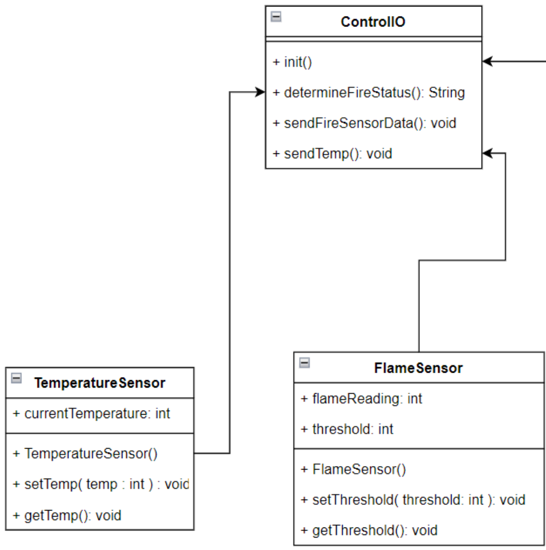


Figure 6: UML Class Diagram of Raspberry Pi [Eline Elorm Nuviadenu]

2.6 Software Design for Arduino

The Arduino is the device used to control the DC motors, servo motors, and the water pumps. The code used to control these hardware components is in C++. A UML diagram outlining the structure of the Arduino software design can be found in the diagram below.

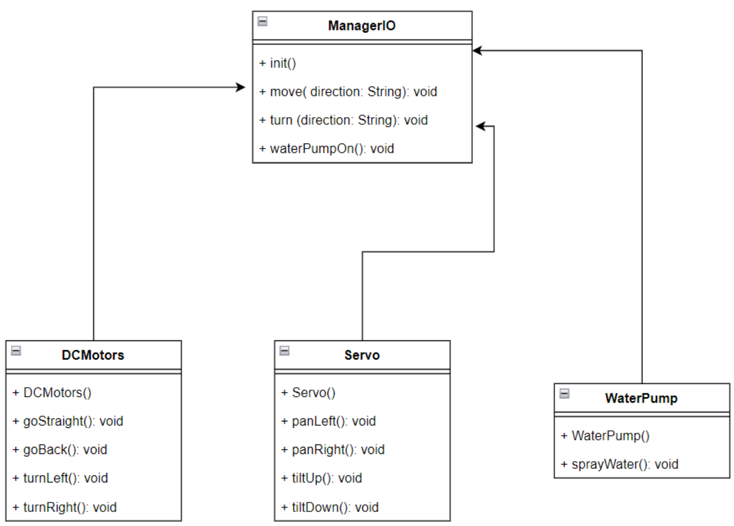


Figure 7: UML Class Diagram of Arduino [Eline Elorm Nuviadenu]

3 Hardware Design

Multiple hardware components would be put together to form a complete and functioning robot. These components include the Raspberry pi with mounted SenseHat, an Arduino uno, a robot chassis with two DC motors, a motor driver, two pan-tilt servo motors, and some water pumps. More about these components can be found in the sections below.

3.1 Hardware Design for Raspberry Pi with mounted SenseHat

The Raspberry Pi serves as the brain of our robot. It is the control for all the other components of the robot. The SenseHat mounted on the Raspberry Pi has a temperature sensor which will be used to determine whether there is a fire when the sensor reading reaches a specified maximum value. The Pi Camera and fire sensor are also connected to the Pi and would help in identifying any indicators of a fire. The Raspberry Pi has both 5V and 3.3V power sources as well as various USB (Universal Serial Bus) ports that allow easy connections to various devices.

As seen in the Figure below, the SenseHat uses six RPi GPIO pins 2, 3, 8, 23, 24 and 25, where pins 2 and 3 enable an I2C connection. These wire connections would not be made in the actual assembly of the robot for the SenseHat would be mounted directly on the Raspberry Pi. On the other hand, the infrared flame sensors, connected in series with a 330kΩ resistor, would be connected to the RPi using an Analog-to-Digital Converter (ADC) chip. This ADC chip would have an SPI (Serial Peripheral Interface) communication with the RPi as it uses the SPI GPIO pins 7, 9, 10, and 11 of the RPi. Furthermore, the Pi Camera is connected using the slot in the RPi board specifically designed for the camera.

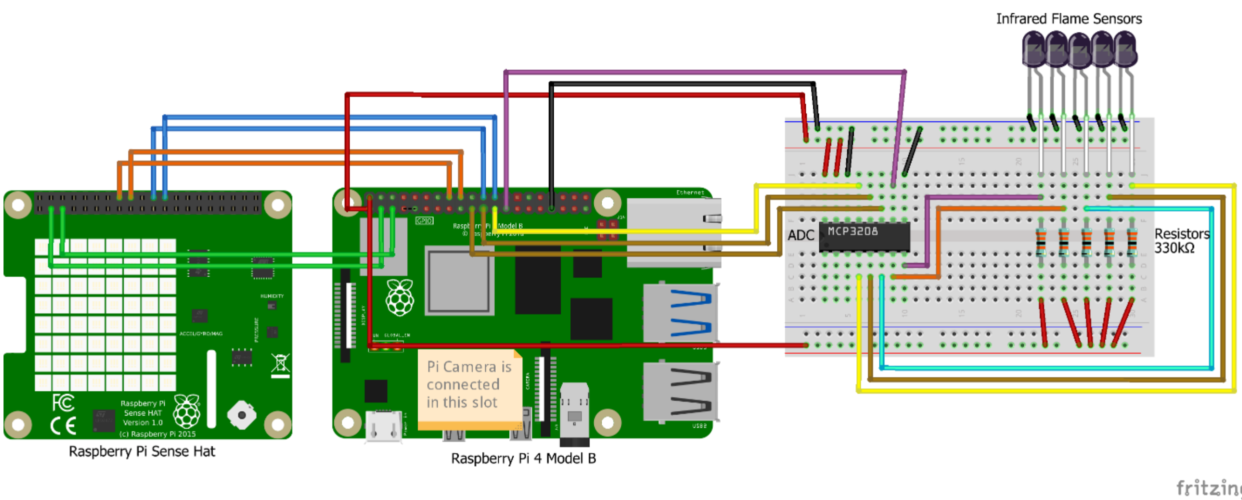


Figure 8: Breadboard view for the RPi and the sensors [Bren-Gelyn Padlan]

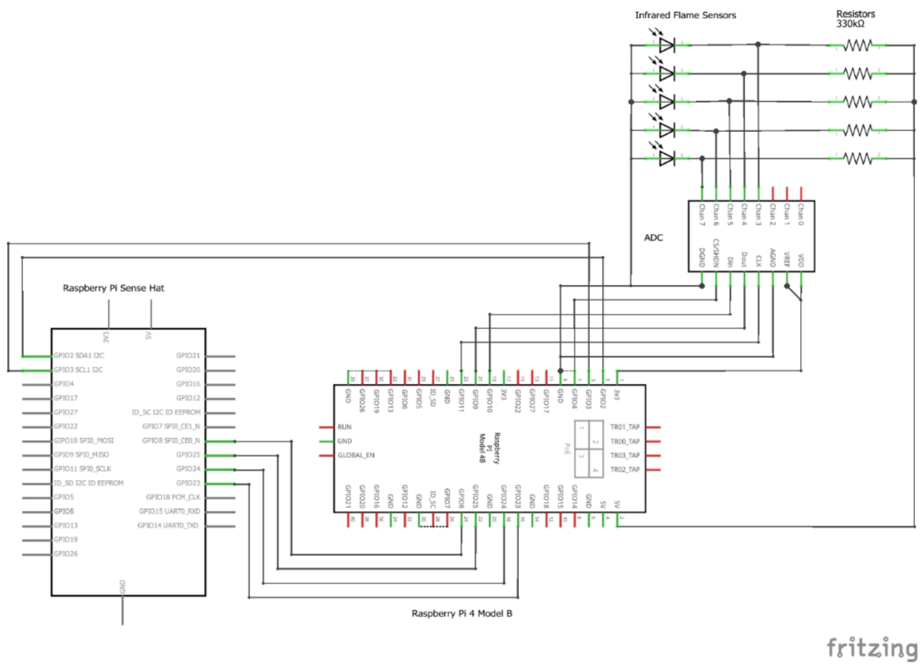


Figure 9: Schematic view of the RPi and its sensors

3.2 Hardware Design for Arduino

The Arduino serves as the muscle of the robot. It controls all the motors and other major components of the robot. The Arduino, like the Raspberry Pi, has both 5V and 3.3V power sources making it very versatile in terms of what can be connected to it. It also has various GPIO pins that support both digital and PWM (Pulse Width Modulation) signals. These pins are used to interface the hardware components with a user-defined program.

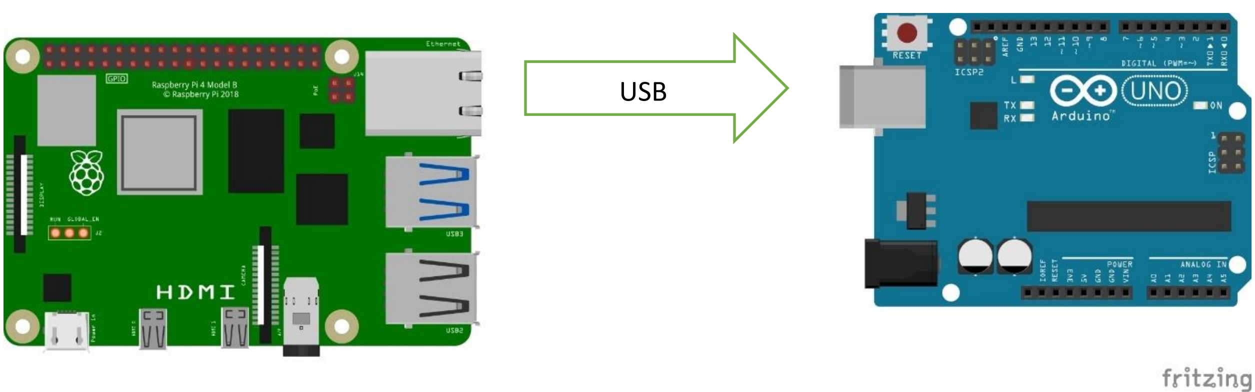


Figure 10: Connection between RPi and Arduino

3.3 Hardware Design for Robot Chassis

The Robotistan Multi-Purpose Mobile Robot Platform chassis was chosen because of its robust nature. The chassis has multiple mounting holes which would allow us to mount all the other components seamlessly and successfully on it. It was also chosen because it contains two DC motors and a castor wheel which allows the robot to be multidirectional.

Figure 11: The Robot chassis

3.4 Hardware Design for Motor Driver

The SparkFun Electronics TB6612FNG Motor Driver Board was chosen mainly to control the direction and speed (using PWM) of the DC motors, allowing the robot to move about as expected. Without this, the DC motors, which controls the wheels, could only run in one direction. Additionally, since a high current signal was needed to make the motors run, the motor driver can turn the low current (5V) it receives from the Arduino, into a higher current to power the motors using an external 6V battery.

Figure 12 below shows the breadboard and schematic view of how the motor driver connects to the system. The motor driver has two PWM pins, eight input and output (I/O) pins (four I/O pins for each motor), a voltage VCC pin for the motor board itself, a voltage Vm for the external battery input and a ground. As seen in the schematic diagram below, the motor driver uses the Arduino GPIO pins 4, 7, 8, 12, 13 for its input pins, and GPIO pins 10 and 11 for its PWM signal pins. The outputs of the motor board are connected to the two DC motors of the robot. Input pin A01 and A02 controls the left DC motor (left wheel), while input pin B01 and B02 controls the right DC motor (right wheel).

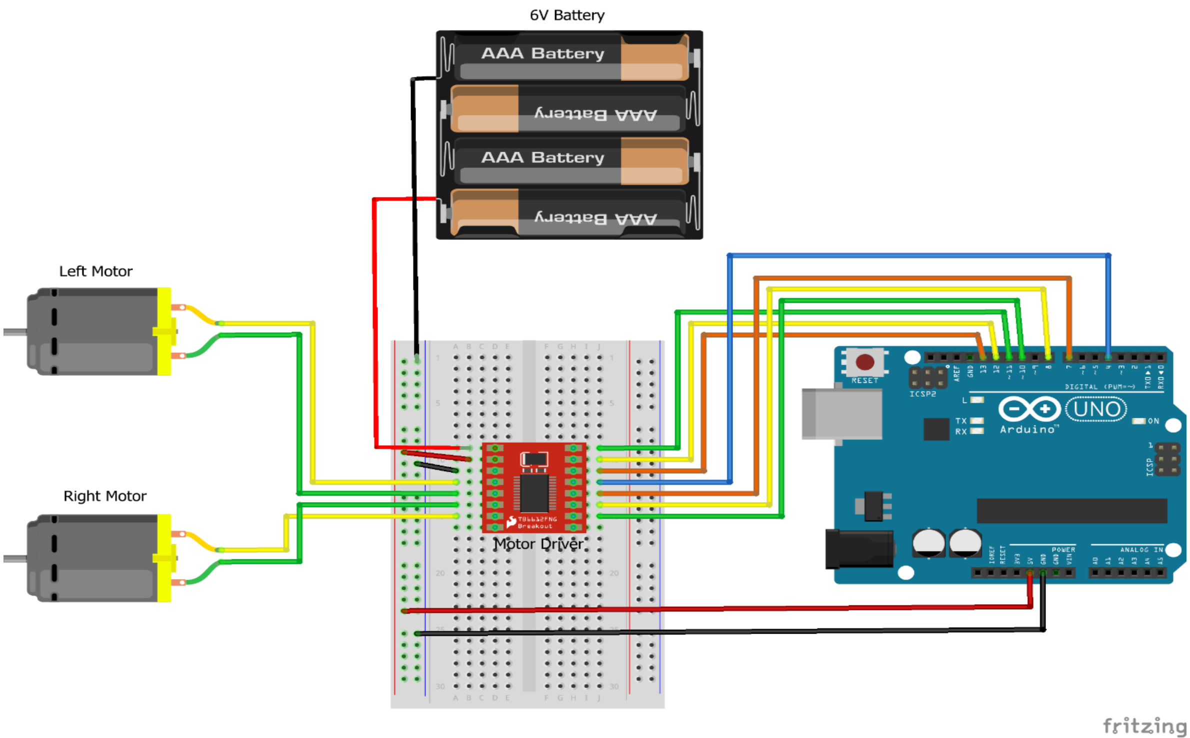


Figure 12: Breadboard Connection of a Motor Driver [Bren-Gelyn Padlan]

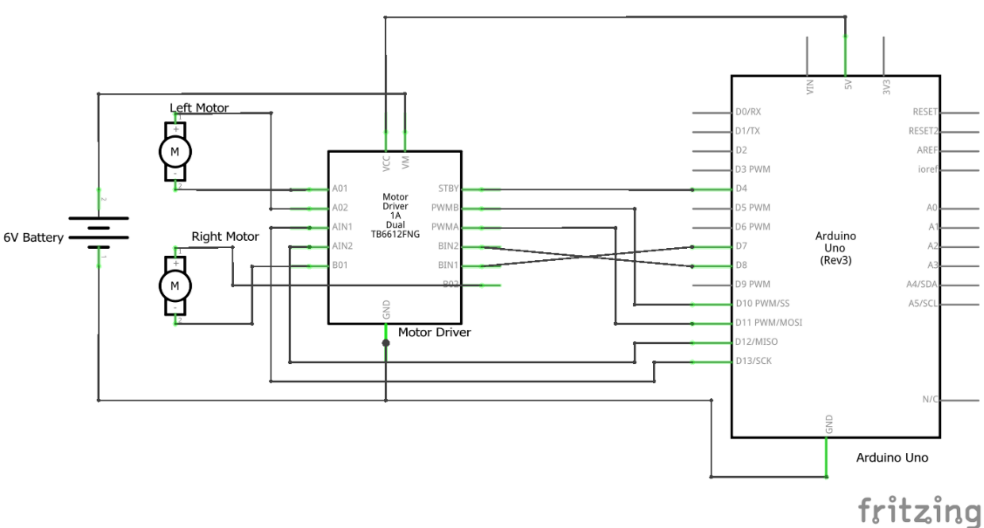


Figure 13: Schematic Diagram of Motor Driver Connection [Bren-Gelyn Padlan]

3.5 Hardware Design for Pan-Tilt Servo Motors

The SparkFun Electronics Pan/Tilt Bracket Kit was chosen to control the direction of both the pi camera, and the tube of the water pump. Due to the positioning of the bracket, the camera as well as the arm controlling the tube, can be turned 180 degrees horizontally and vertically. This increases the field of vision of the robot and allows for easier extinguishing of fires.

To control the Servo Motors, a pin with Pulse Width Modulation (PWM) is needed for each one of them. The fire fighting robot needs four servos to function (two controlling the camera and the other two controlling the spraying arm), and these servos will be connected to the 3, 5, 6, and 9 GPIO pins as seen in the figures that follow. The other two pins of the servo motor are connected to 5V and ground.

Figure 14: The Pan-Tilt Motor

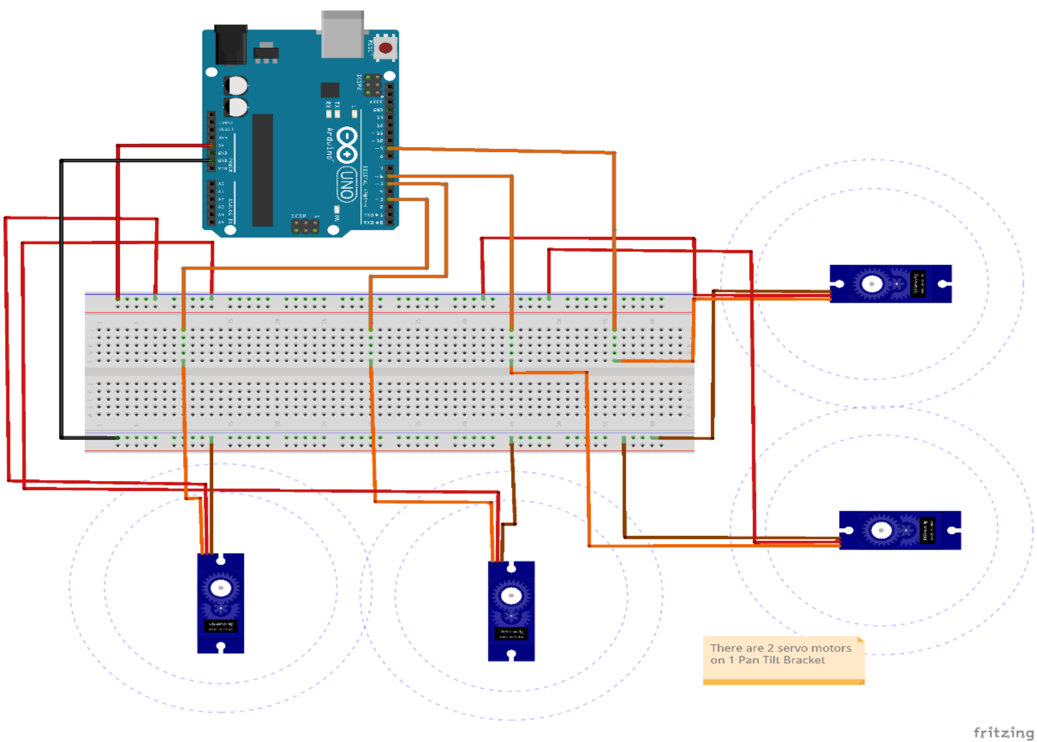


Figure 15: Breadboard Connection of Servo Motors [Eline Elorm Nuviadenu]

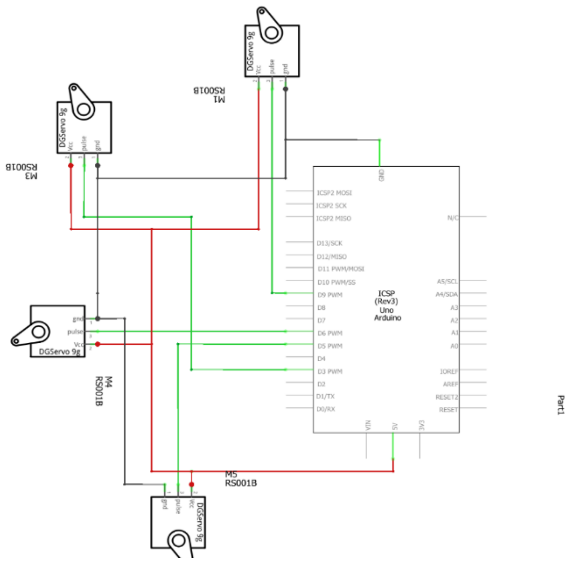


Figure 16: Schematic diagram of Servo Motor connections
[Eline Elorm Nuviadenu]

3.6 Hardware Design for Water Pumps

The Gikfun Micro Submersible Mini Water Pump was chosen to operate the fire extinguishing characteristic of the robot. The utilization of the water pump reduces the complexity of creating an object to press down on a regular fire extinguisher. The water pump is connected to the Arduino and controlled by the user through the mobile application. The water pump requires 3V to 4.5V and a current of 0.18A to work, therefore, when connecting to the Arduino, a 3.3V is used.

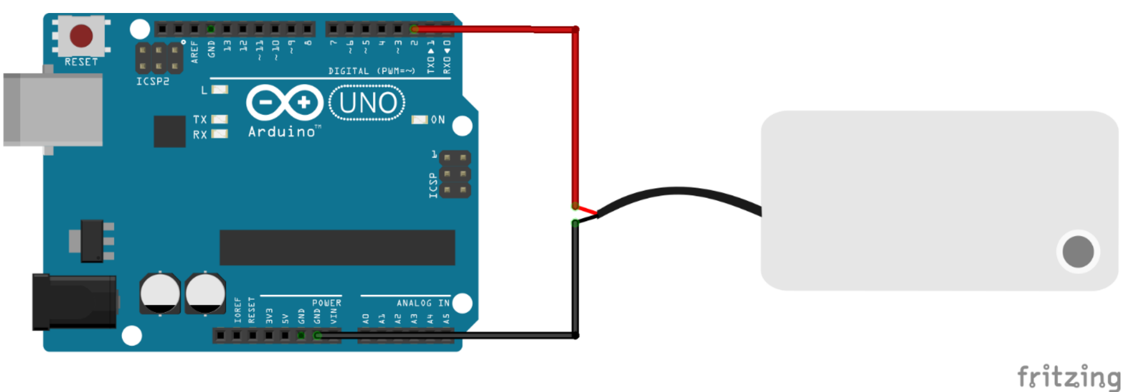


Figure 17: Breadboard Connection of Water Pump

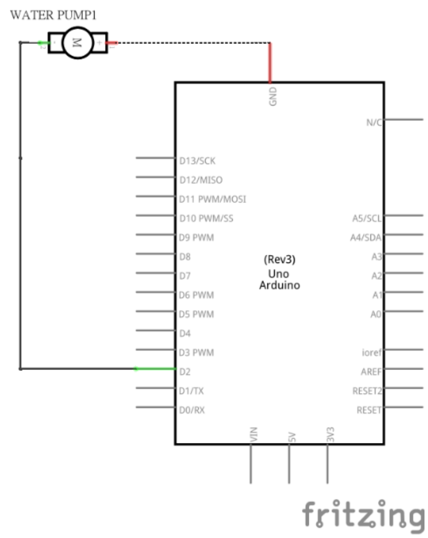


Figure 18: Schematic diagram of Water Pump connection

4 GUI Designf

The figures that follow depict the proposed graphical user interface for the Mobile Application for the robot (FiWER). It provides the user with the ability to control the robot, and to notify the robot of the average room temperature, allowing the robot to estimate a range of temperatures deemed as normal in the user's home. The mobile application also allows the user to keep track of the temperature in their home, as well as send a notice if there is a possible fire.

Figure 19: Front Page of Prototype of FiWER GUI [Eline Elorm Nuviadenu]

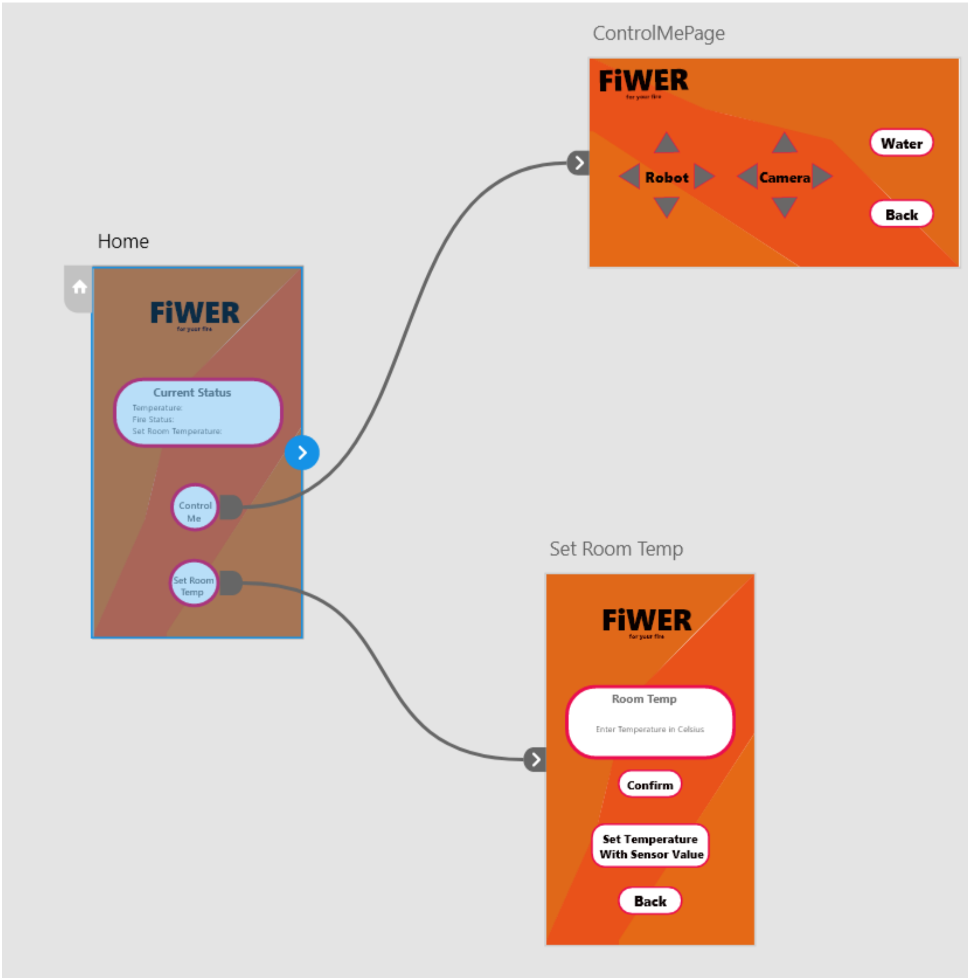


Figure 10: Prototype of FiWER GUI [Eline Elorm Nuviadenu]

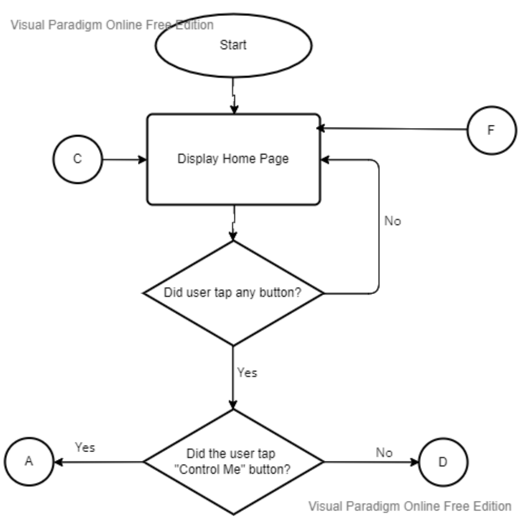


Figure 10: Main Flowchart of FiWER GUI [Bren-Gelyn Padlan]

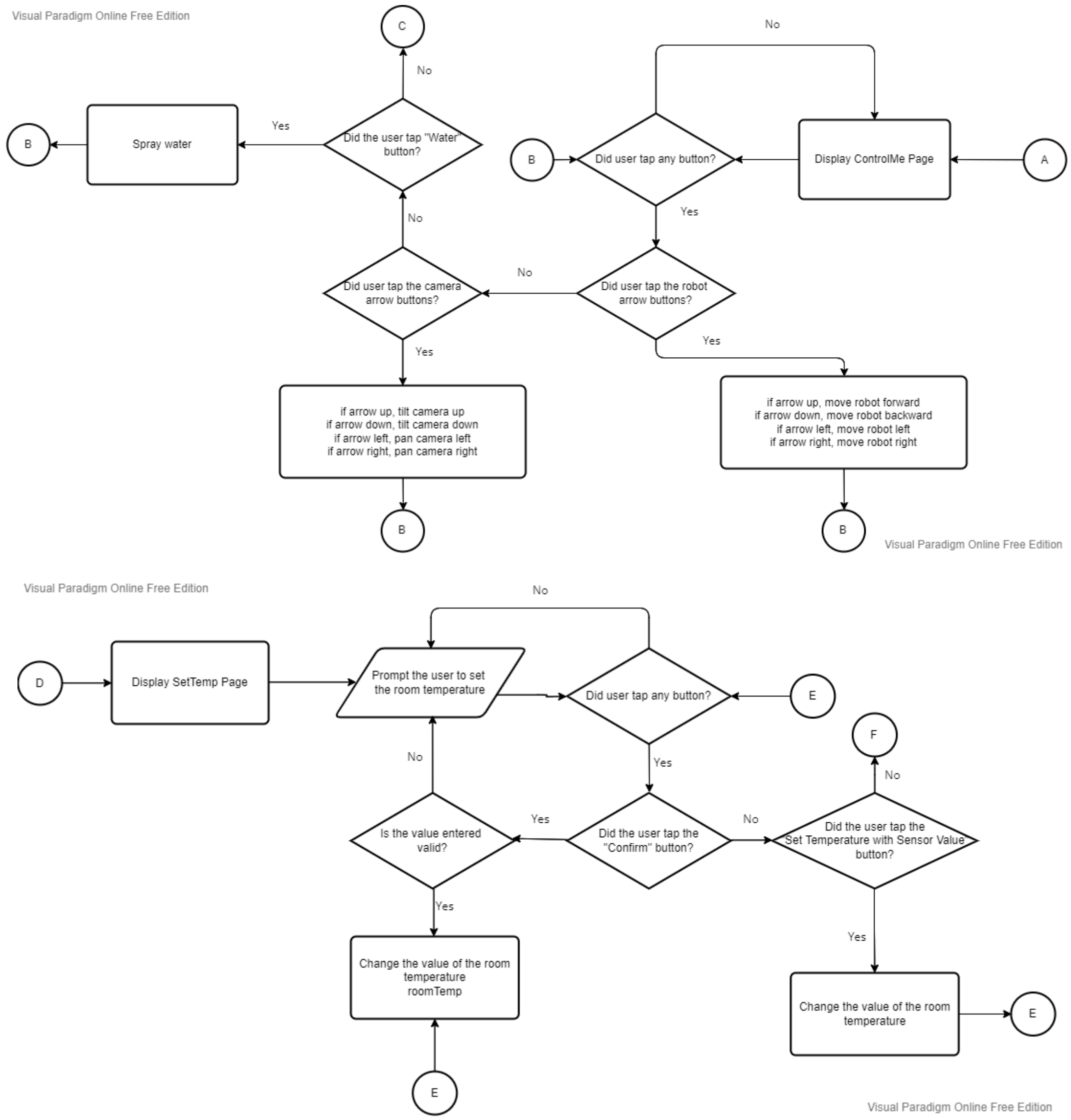


Figure 20. Flowchart for the GUI

4.1 Table of Users/Roles

Users	Interaction
House owner	Robot will notify owner when a fire is detected, owner then uses GUI to control the robot to extinguish the fire.

5 Test Plans

5.1 End-to-end Communication Demo Test Plan

The components and nodes of the project which involve end-to-end communication are the Raspberry Pi and the Arduino, the Android mobile app and the database, and the Raspberry Pi and the database. These are the components which we would write tests for. The main functionalities we would test would perform the following:

- We would confirm that the Arduino can successfully receive messages from the Raspberry Pi. This would prove that we would be able to send instructions to the Arduino and utilize these instructions to perform specified tasks.
- The Raspberry Pi's communication with the database would also be tested to confirm that we would be able to successfully write and read data to and from the database. Confirming this would allow us to write functions to work on the data that is read from the database and send results back to the database to be stored and used.
- The communication between the mobile app and the database would also be tested as it is a main part of our project. This test would be done by using the mobile application to send information to the database and by displaying information from the database on the mobile application. This would validate that sending instructions using the mobile application is feasible.

5.2 Unit Test Demo Test Plan

For the Unit Test Demo, our main aim would be to test all the parts of our system to make sure that they work as expected.

- Pytests and AUnit tests would be written to test each of the methods used to control all hardware devices. This would be set to confirm that all hardware components are functional.
- In order to test data integrity, known values would be sent into the database and would be retrieved. This would confirm that the right values are being successfully saved. Tests would also be written to determine if data can be deleted and modified successfully.
- The basic functionality of the mobile application would be tested (controlling the camera, moving the robot, utilizing the water pump, and setting a threshold temperature).
- Code reviews would also be done by team members who worked on other code to improve the chances of uncovering any problems that may arrive.
- A lighter would be put in front of the robot and the values in the database would be monitored to confirm that the data is being logged correctly.

5.3 Final Demo Test Plan

For the final demonstration, the following scenarios would be taken into consideration in proving that our project works successfully as expected:

- A lighter would be placed in front of the robot. The user, who is close by, simply activates the water pump through the application to quench the fire.
- A small piece of paper would be set on fire in a room with reduced airflow (e.g., a chemistry lab). The user who has the mobile application stays outside, waits for the push notification, connects to VNC (Virtual Network Computing) with a computer, and directs the robot to move towards the fire and extinguish it.
- A lighter would be put in front of the robot which is connected to a network. The user would make sure that the phone with the mobile application is connected to another network. The user would then use the mobile application to extinguish the fire. This test proves that the robot can be controlled from anywhere.

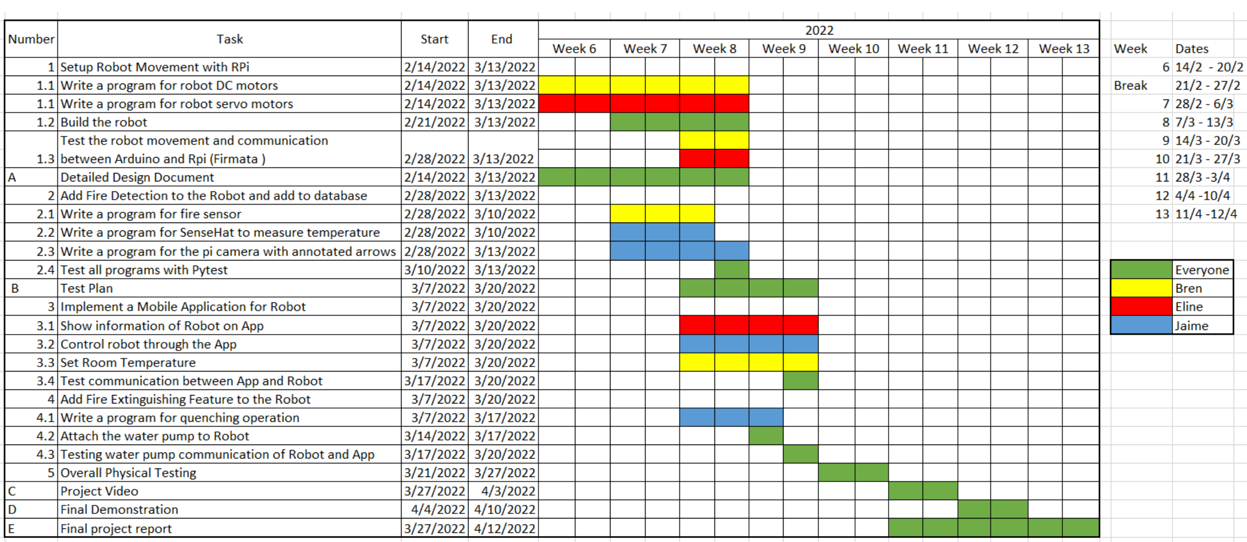
6 Project Update

Our team has been making good progress on the project. As the project kept progressing, we made multiple omissions and additions so as to minimize cost and increase efficiency while keeping time constraints in mind. We decided to replace the fire extinguisher with a water pump, as it would be more convenient for a user to replace and reuse. We also decided to leave out a smoke detector, as it was agreed that the combination of the PiCamera, fire sensors, and SenseHat would be just enough to get the work done. Due to the late arrival of some components, we were falling a little behind schedule, but we reassessed our plans and timelines and are due to be back on track by the end of the week.

6.1 Project Milestones

Milestone 1: Setup Robot Movement with RPi (February 14 – March 13)
The team must write a program to control and move the robot in real-time.
Milestone 2: Add Fire Detection to the Robot and add to database (February 28 – March 13)
The different sensors (fire sensor, SenseHat, and pi camera) must be integrated into the robot. Sensors must take data and send it to the database.
Milestone 3: Implement a Mobile Application for Robot (March 7 – March 20)
The mobile application to communicate with the robot must be done using Android Studio.
Milestone 4: Add Fire Extinguishing Feature to the Robot (March 7 – March 20)
The water pump must be attached to the robot. Once the fire is detected, a user must be able to quench it.
Milestone 5: Physical testing of the Robot (March 21 - 27)
Physical testing will be performed. A controlled fire (e.g., using a lighter) will be set up to test the ability of the Robot.

6.2 Schedule of Activities



Number	Task	Start	End	2022 Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
1	Setup Robot Movement with RPi	2/14/2022	3/13/2022								
1.1	Write a program for robot DC motors	2/14/2022	3/13/2022								
1.1	Write a program for robot servo motors	2/14/2022	3/13/2022								
1.2	Build the robot	2/21/2022	3/13/2022								
1.3	Test the robot movement and communication between Arduino and Rpi (Firmata)	2/28/2022	3/13/2022								
A	Detailed Design Document	2/14/2022	3/13/2022								
2	Add Fire Detection to the Robot and add to database	2/28/2022	3/13/2022								
2.1	Write a program for fire sensor	2/28/2022	3/10/2022								
2.2	Write a program for SenseHat to measure temperature	2/28/2022	3/10/2022								
2.3	Write a program for the pi camera with annotated arrows	2/28/2022	3/13/2022								
2.4	Test all programs with Pytest	3/10/2022	3/13/2022								
B	Test Plan	3/7/2022	3/20/2022								
3	Implement a Mobile Application for Robot	3/7/2022	3/20/2022								
3.1	Show information of Robot on App	3/7/2022	3/20/2022								
3.2	Control robot through the App	3/7/2022	3/20/2022								
3.3	Set Room Temperature	3/7/2022	3/20/2022								
3.4	Test communication between App and Robot	3/17/2022	3/20/2022								
4	Add Fire Extinguishing Feature to the Robot	3/7/2022	3/20/2022								
4.1	Write a program for quenching operation	3/7/2022	3/17/2022								
4.2	Attach the water pump to Robot	3/14/2022	3/17/2022								
4.3	Testing water pump communication of Robot and App	3/17/2022	3/20/2022								
5	Overall Physical Testing	3/21/2022	3/27/2022								
C	Project Video	3/27/2022	4/3/2022								
D	Final Demonstration	4/4/2022	4/10/2022								
E	Final project report	3/27/2022	4/12/2022								

Week	Dates
6	14/2 - 20/2
Break	21/2 - 27/2
7	28/2 - 6/3
8	7/3 - 13/3
9	14/3 - 20/3
10	21/3 - 27/3
11	28/3 -3/4
12	4/4 -10/4
13	11/4 -12/4

	Everyone
	Bren
	Eline
	Jaime

Figure 21. Project activities and deliverables timeline [Hiu Sum Jaime Yue]

References

[1] Bucharskiy, Y., 2022. Fire fighting robot flat illustration with mobile autonomous tracked.... [online] iStock. Available at: <https://www.istockphoto.com/vector/fire-fighting-robot-flat-illustration-with-mobile-autonomous-tracked-vehicle-gm1173310487-325834986> [Accessed 8 February 2022].

[2] "Robotistan - Multi-Purpose Mobile Robot Platform - R.E.X chassis series - red color - suitable for STEM," *Amazon.ca: Home*. [Online]. Available: https://www.amazon.ca/gp/product/B08P5WH611/ref=ppx_yo_dt_b_asin_image_o00_s00?ie=UTF8&psc=1. [Accessed: 08-Mar-2022].

[3] "Rob-14391," *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/sparkfun-electronics/ROB-14391/7675361. [Accessed: 08-Mar-2022].