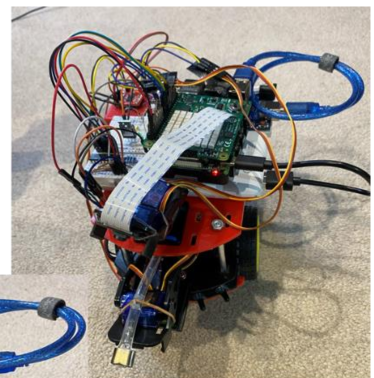
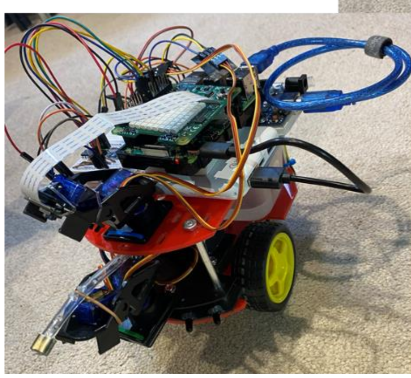
SYSC3010

Computer Systems Development Project

Household Fire Warden and Extinguisher Robot (FiWER)





Group L3-G3

Bren-Gelyn Padlan, 101148482

Eline Elorm Nuviadenu, 101162869

Hiu Sum Jaime Yue, 101162929

TA: Roger Selzler

April 12, 2022

Table of Contents

1 Project Description

1.1 Motivation

House fires are one of the common emergencies that are usually caused by preventable events [1]. A kitchen is the most common area where household fires occur as unattended cooking and unsupervised open flames are the leading cause of house fires [2]. This cause of fires can be easily prevented; however, most often than not, the fire gets huge when it does not need to. Thus, more calls are being made to fire stations for fires that could have been initially prevented. Firefighters are called on-site on an average of 20 times a day, only to realize that there was little or no serious fire [3]. The project vision was to develop a safer and more comfortable home environment.

1.2 Problem Statement

The objective of this project was to reduce the workload of firefighters by developing a reliable firefighting robot that could detect and extinguish small household fires. This robot was called the Household Fire Warden and Extinguisher Robot, or FiWER.

1.3 Overview of Design Solution

This project developed a robot that can detect and extinguish fires. Flame and temperature sensors were integrated into the system to detect the presence of fire. When a fire is detected, the mobile application created for this system should notify the user to control the robot and extinguish the fire through push notifications. For the user to control the robot, a module that controls the robot's hardware components – DC motors, servo motors, and pumps – was implemented. Furthermore, a Firebase database was created and configured to store all the data that the mobile application and the robot needed to communicate with each other.

2 Final Design Solution

2.1 Deployment 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 - the Robot, Firebase and Phone communicating with each other through the cloud - to meet the project requirement of controlling the robot remotely. Additionally, having a Raspberry Pi and Arduino sub-nodes inside the Robot node allowed for the fire detection and extinguisher feature of the robot.

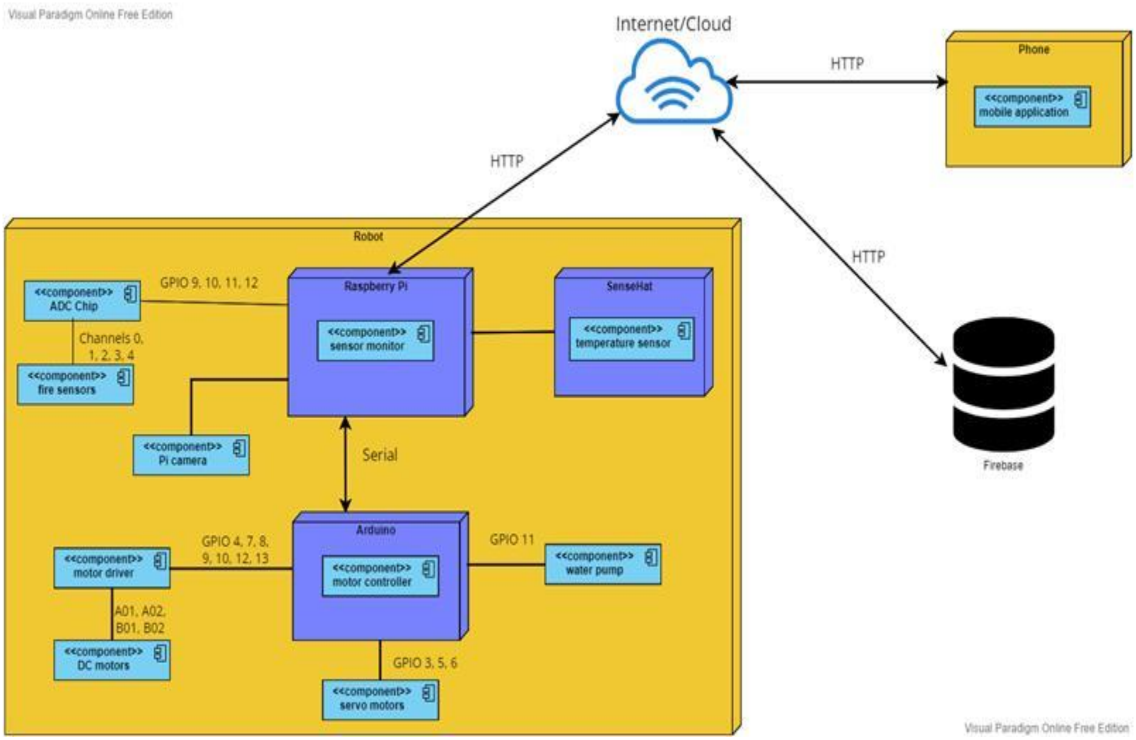


Figure 1: Deployment diagram of the FiWER project [Bren-Gelyn Padlan]

The Robot node had three sub-nodes namely Raspberry Pi (RPi), Arduino, and SenseHat. The Arduino was used to operate the DC motors, servo motors, and water pump of the robot and to send 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, was responsible for getting the camera view on Pi Camera, reading the data from the IR flame sensors (connected via SPI), and communicating with the Arduino to send commands to control the motors and pumps through Serial communication. The RPi was also responsible for sending the sensor data to Firebase through the cloud via HTTP communication. The third sub-node was the SenseHat which contained the temperature sensor that was also used for fire detection. The SenseHat was also made to communicate with the RPi via I2C.

Firebase was used to host the database for the robot logs, containing the sensor and motor (DC and servo) data. It was connected to the cloud via HTTP. A single database table was used to hold all the information gathered from the Robot.

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

2.2 Message 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	"No"
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"

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

Robot and Datebase			
Sender	Receiver	Message	Data Format
Robot	Database	sendFireSensorData	4906
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"

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"
Arduino	Rpi	robotIdle	"Idle"
Arduino	Rpi	robotMoveForward	"Moving Forward"
Arduino	Rpi	robotMoveBackward	"Moving Backward"
Arduino	Rpi	cameraMoveLeft	"Moving Left"
Arduino	Rpi	cameraMoveRight	"Moving Right"
Arduino	Rpi	waterPumpOn	"WaterPumpOn"
Arduino	Rpi	cameraPanLeft	"Paning Left"
Arduino	Rpi	cameraPanRight	"Paning Right"
Arduino	Rpi	cameraTiltUp	"Tilting Up"
Arduino	Rpi	cameraTiltDown	"Tilting Down"
Arduino	Rpi	WaterPumpOff	"Water Pump Off"

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

Figure 3 below shows the summarized version of the communication protocol table using JSON format.

Sender	Receiver	MessageContent
Rpi	Database	{"sendDataFromRpi":{"EntryNumber": INT, "Temperature" :INT,"FireSensor":INT,"FireStatus": TEXT}}
Database	Rpi	{"receiveDataFromDataRpi":{ "Timestamp":TIME, "MotorStatus": TEXT}}
Database	mobile	{"receiveDataFromDataMobile":{"EntryNumber": INT, "Temperature" : INT,"FireSensor":INT,"FireStatus": TEXT}}
mobile	Database	{"sendSetTempFromMobile":{"SetRoomTemp": INT}}
mobile	Database	{"sendMotorStatusFromMobile":{ "Timestamp":TIME, "MotorStatus": TEXT}}
SenseHat	Rpi	{"sendTempFromSenseHat":{ "Temperature": INT}}
Rpi	Arduino	{"sendMotorStatusToArduino":{"MotorStatus": TEXT}}

Figure 3: Summarized Communication Protocol Table using JSON format [Hiu Sum Jaime Yue]

The structure of the database is shown in Figure 4 below. The database uses three indices, namely RobotDevice, Daily Log, and UserInfo. Under RobotDevice, the robotDevice Id is used as the index. Under the robotDevice Id, there are three different pieces of information, including Log, MobileDevice, and user.

Under Log, there are five different pieces of information, including FireSensor, Fire Status, Motor Status, SetRoomTemp and Temperature. Each of this information would have a timestamp or an entry number and their corresponding values. Under Daily Log, there are four information shown, which are Activities, AvgFireSensor, AvgTemp and LatestSetRoomTemp. Under Activities, the number of Warning and Serious shown during that day will be shown respectively. Lastly, under UserInfo, the username is used as the index. Under the username, there are three information, namely, Email, MobileDevice, and RobotDevice.

RobotDevice											
FIWER1											
Log									MobileDevice		user
FireSensor		FireStatus		MotorStatus		SetRoomTemp	Temperature		1	"SM-G9980"	"Jaime"
0	4094	0	"No"	4/11/2022 8:02:49	"Idle"	34	0	38			
1	4000	1	"Warning"	4/11/2022 8:03:00	"PanLeft"		1	42			

Daily Log							
Activities		AvgFireSensor		AvgTemp		LatestSetRoomTemp	
04--11--2022		04--11--2022	4904	04--12--2022	39	04--11--2022	45
Serious	Warning						
0	0						

UserInfo					
Jaime					
Email	MobileDevice		RobotDevice		
xxx@gmail.com	1	"SM-G9980"	1	"FIWER1"	

Figure 4: FIWER database table [Hiu Sum Jaime Yue]

2.3 Sequence Diagrams

2.3.1 Message Sequence Diagram 1: Detect Fire

The detect fire use case demonstrated the basic work of the robot. In this use case, the robot should detect temperature every few minutes and send the data to Firebase. The phone, through the mobile application, should get the temperature and the fire status from the Firebase. When the fire status says “Warning” or “Serious”, it must alert the user about the fire status. Figure 5 illustrates the sequence diagram for this use case.

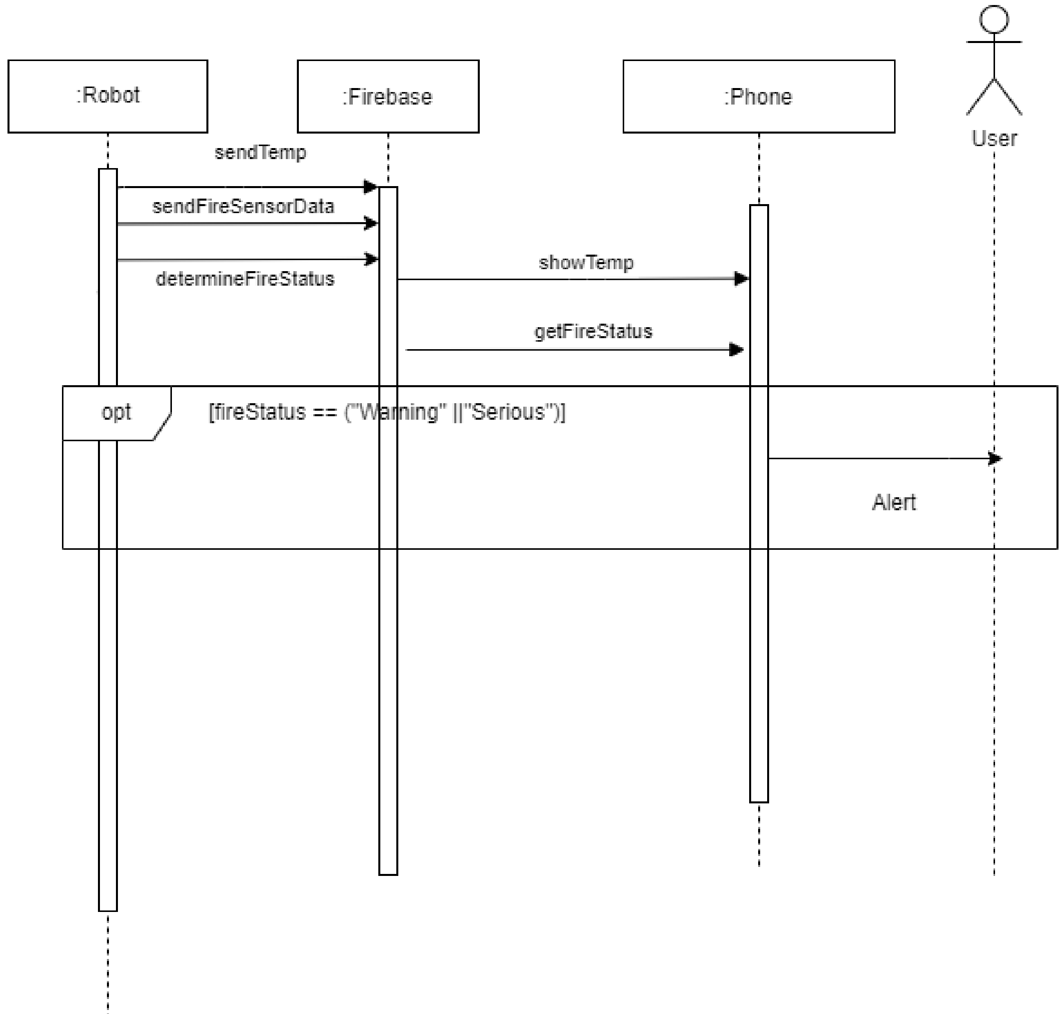


Figure 5: Message sequence diagram for use case 1: Detect Fire [Hiu Sum Jaime Yue]

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 described the steps that the robot would follow to perform this task. To perform this task, the user should control the robot. There was no order of how the user wanted 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 6 below, as it shows a sequence diagram of the extinguish fire use case.

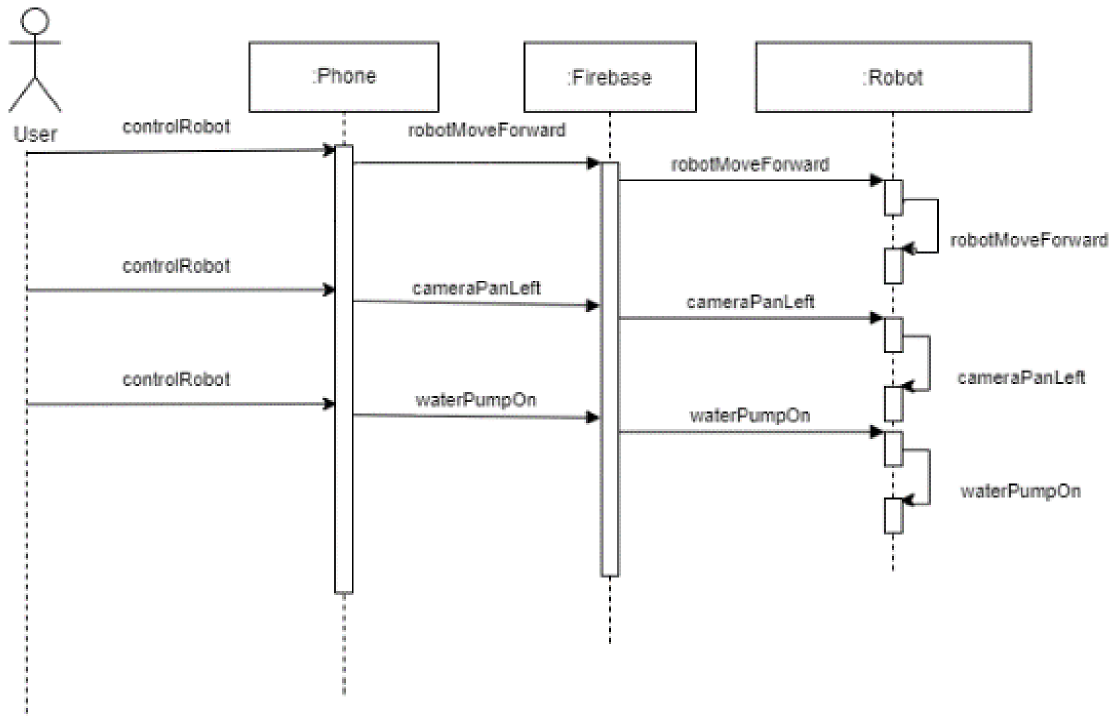


Figure 6: Message sequence diagram for use case 2: Extinguish Fire [Hiu Sum Jaime Yue]

3 Final Design Discussion

As the project progressed, we made several changes to our original plan which was stated in the proposal document. These changes and the reasons behind them will be outlined in the paragraphs that follow.

We dropped the fire extinguisher and opted for a water pump instead as the cost of obtaining a fire extinguisher was too high. We also found it difficult to obtain a fire extinguisher of the desired size in stock. The water pump was chosen as a substitute, as it is reusable, and cheap but effective.

We also intended to use two pan-tilts which would have ended up with a total of four servo motors – two controlling the camera motions, and the other two controlling the direction of the stream of water when extinguishing. While combining all components, we realized that the water pump would need a PWM pin to be controlled as desired, and it was unnecessary for the stream of water to be tilted, prompting us to replace the tilt servo with the water pump.

The hand to be used to control the spraying tube was initially going to be 3D printed, but due to the size of our chassis, we opted to leave it out as we could easily do without it.

The camera was initially supposed to come with annotations that would assist in leading a user toward the source of the fire, but as we started working on the project, we decided to prioritize fixing major components of the robot, which made this is a secondary task to be added in future versions of the project.

The smoke detector was also left out as smoke could be seen on the camera, and the five fire sensors present would record and notify the user of the presence of any fire.

The database structure was also changed to make any future additions of extra phones, or robots easy to implement.

4 Contributions

Table 1: Authors of the source codes used for FIWER system

Team Member	Code
Bren-Gelyn Padlan	FiWER_Arduino_Manager.ino main_fiwer.py fiwer.py flame_sensor.py database.py SetRoomTemp.java
Eline Nuviadenu	FiWER_Arduino_Manager.ino camera.py arduino.py database.py HomePage.java
Hiu Sum Jaime Yue	temp_sensor.py database.py ControlPanel.java HomePage.java MainActivity.java NotificationReceiver.java User.java SetRoomTemp.java

Table 2: Authors of the sections of Final Report

Team Member	Section
Bren-Gelyn Padlan	Overview of Design Solution Deployment Diagram (Update) Appendices A Appendices C, D (Update)
Eline Nuviadenu	Motivation (Update) Problem Statement (Update) Final Project Discussion Appendix B Appendix C (Update)
Hiu Sum Jaime Yue	Message Protocol Table (Update) Sequence Diagrams (Update) Final Project Discussion

5 Reflections

Bren-Gelyn Padlan: *At the beginning, working on the project was very challenging. I had little experience in building a robot, so finding the right hardware components and building the circuits required a lot of research and self-learning. I also feel that even if we were a group of three, the project was still a little tough since all of us have almost the same levels of knowledge and experience with programming and robotics, that we had to delay our original timeline to accommodate for the time we needed to learn the necessary skills for the project. However, in the end, with the help of our TA, we pushed through, and we ended up building a robot that moves, detects fire, and sprays water.*

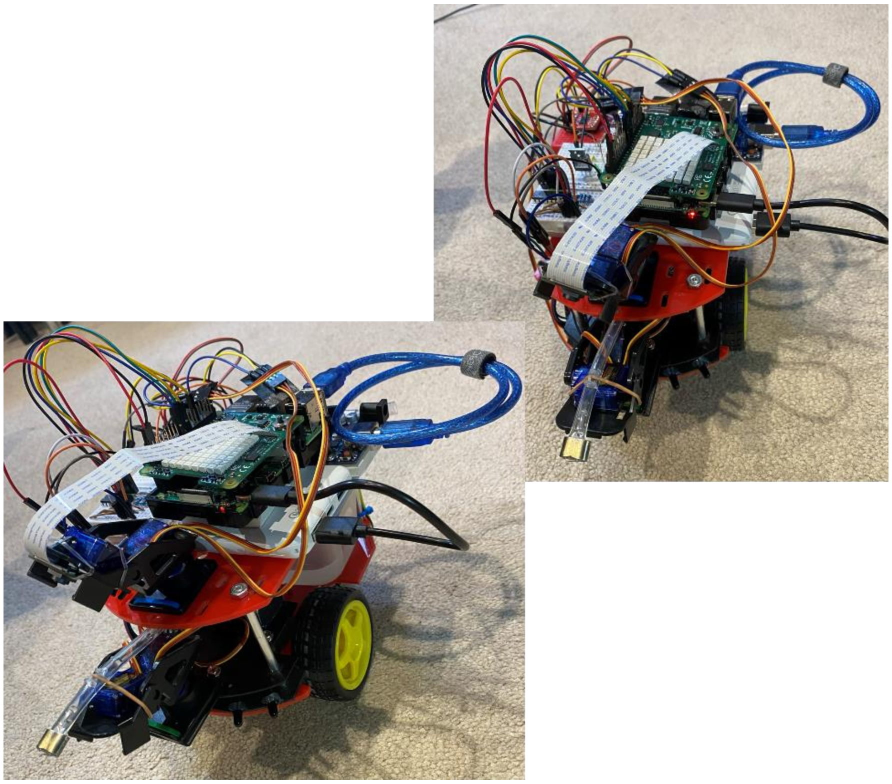
For the most part in this project, I learned a lot about building the robot - making all the wiring connections, understanding what each hardware component (motor board and ADC chip) does, and writing the code that makes the component work. I also learned about the importance of collaboration and distribution of work. If we were given more time, I would like to upgrade the chassis of our robot into a bigger one, change the DC motors of the wheels, and figure out how to live stream the robot's camera into the mobile application.

Eline Nuviadenu: *In seeing this project from start to completion, I have learnt the benefits of using online simulation platforms like Fritzing and Tinker CAD to simulate circuit connections to avoid blowing components. I have also learnt a lot about how the version control software allows for easier collaboration and tracking of changes. This project has also helped me realize the amount of work, effort and planning that goes into manufacturing the simplest of components we see around us. I would have liked to work on making the mobile application more aesthetically pleasing, but with the amount of research and self-learning this project exposed me to, I feel confident that I can tackle it outside of class and still end up with great results.*

Hiu Sum Jaime Yue: *This project was a whole new experience for me, from starting to think of a project topic, to making a robot. The process was quite challenging, but everything went well in the end, at which I am extremely thankful for my teammates. We were able to finish most of what we had planned on doing. I have learned the importance of setting milestones for projects, so I need to try to keep up with the pace of the project. I also learned how to create an app with Firebase. If we have more time for this project, I will put annotations on the camera view, so when the user is using the camera, arrows will be added toward the direction of the fire.*

6 Appendix A: README

Household Fire Warden and Extinguisher Robot (FiWER)



Group number: L3G3

Team members: Eline-Elorm Nuviadenu, Bren-Gelyn Padlan, Hiu Sum Jaime Yue

TA: Roger Selzler

Course Code: SYSC 3010

Project Description

The objective of this project is to create a fire fighting robot that can detect and extinguish fires. The robot is placed in a room or an enclosed area, and once the code is run, the robot's sensors are turned on and the robot starts logging its sensor values. The user can remotely control the robot through a mobile application. When the robot detects fire, the user is notified through push notifications from the app. The user is also able to livestream by connecting via VNC to the RPi and observe what the robot is currently seeing.

The robot is built with a two-wheeled chassis with a ball-bearing roller for support. The

Raspberry Pi is the brain of the robot as it monitors different sensors such as flame sensors, temperature sensor and the camera. It also communicates with the Arduino for controlling the movement of the robot and for extinguishing the fire. This project uses a Firebase database to store all the information collected from the user such as name and email, and the information used by the robot such as temperature, fire status and motor status.

GitHub Repository Description

The repo consists of three main directories. Since this repo was used by the team throughout the semester, directories `Lab4` and `WeeklyUpdates` were added here. The `Project` directory was structured such that the codes for each demo were contained in a separate directory.

Inside `Final`, `MobileApp` and `Robot` directories contains all the source codes for the app and robot, respectively. The breakdown of the GitHub repo is shown below:

```
├── Lab4
├── Project
│   ├── Demo
│   ├── Final
│   │   ├── Mobile App
│   │   │   ├── (all source files for the app)
│   │   ├── Robot
│   │   │   ├── (all source files for the robot)
│   ├── UnitTestDemo
├── WeeklyUpdates
└── README.md
```

Installation

Hardware Connections

The main hardware used in this project are the Raspberry Pi 4 Model B and Arduino Uno. The RPi and Arduino are connected to sub-components that the robot needs to function. The circuit schematics for these connections can be found here.

Packages

There are packages required for this project.

In your RPi, type the following commands:

```
sudo apt-get install sense-hat
```

```
python -m pip install pyrebase4
```

Also, make sure to enable Camera, SSH, VNC, SPI and I2C by running `sudo raspi-config` on the terminal and selecting `3 Interface Options`.
In your Arduino IDE, add these libraries by downloading the zip of the following GitHub repositories:

- SparkFun_TB6612FNG_Arduino_Library - library used for controlling the Motor Driver
- Timer2ServoPWM - library for configuring servo motors to use Timer2 of Arduino

Mobile App

To install the app to your Android phone, follow these steps:

1. Install Android Studio to your local computer by following the instructions on this link: https://developer.android.com/studio
2. Download the source codes from `MobileApp` in this repo, or you can clone this repo instead (*see **Running the system** section*)
3. Open Android Studio, and run MainActivity.java.
   To open the app through your Android phone:
   i. Go to your phone settings, turn on Developer Options and USB debugging
   ii. Connect your phone to your computer via a USB cord
   iii. Then you will see your phone as an option in Device Manager

Running the system

Before running the system, make sure your RPi is connected to your Wi-Fi and you have your RPi's IP address.
To run the system, follow these steps:

1. Turn on RPi, and open VNC in your local computer.
2. Connect to your RPi by entering its IP address in VNC.
3. Open terminal and clone this repo to your RPi.

```
git clone https://github.com/elineelorm/SYSC3010W22_L3_G3
```

4. Go to Project > Final > Robot.

```
cd SYSC3010W22_L3_G3/Project/Final/Robot
```

If you type `ls`, you should see:

```
arduino.py   database.py                 fiwer.py         main_fiwer.py
camera.py    FiWER_Arduino_Manager.ino   flame_sensor.py  README.txt
```

5. Open Arduino IDE, and open the [FiWER_Arduino_Manager.ino]() file. Click the upload button to upload code to Arduino.
6. Back to the terminal, run `main_fiwer.py`.

```
python3 main_fiwer.py
```

7. Open another terminal, run `camera.py`.

```
python3 camera.py
```

8. Open a third terminal, run vlc by typing `vlc`. VLC will open. Go to Media > Open Network Stream > Network. Type:

```
rtsp://<IP Address>:8554/
```

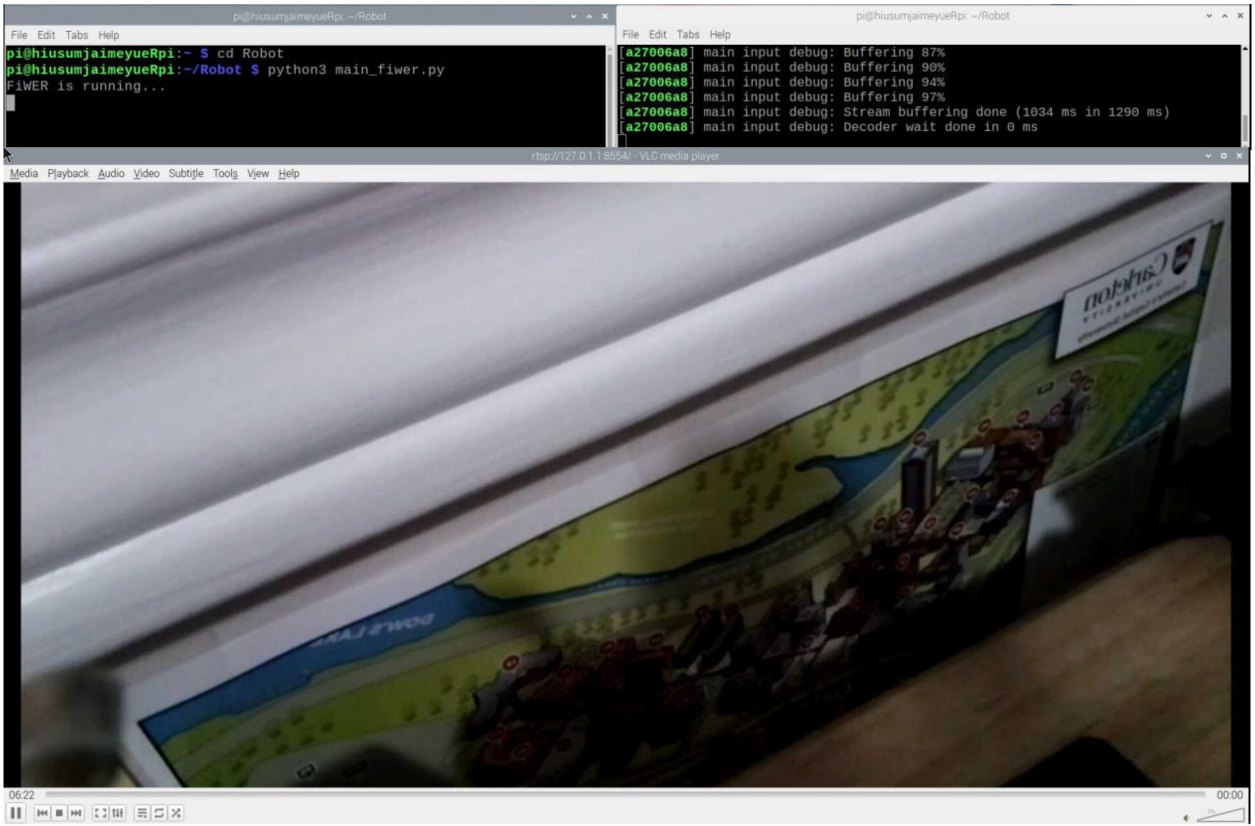
9. Open the FIWER app on your Android phone. Sign-in using your email and password, then tap the `ControlMe` button. Control the robot by tapping the appropriate buttons.

Output

Once the installation is complete, and the system is run, below shows the sample output.

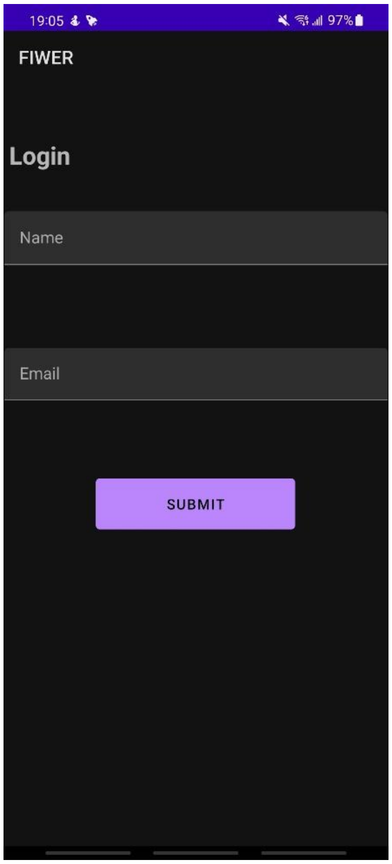
RPi

This is what you should see once you run `main_fiwer.py`, `camera.py`, and `vlc`.

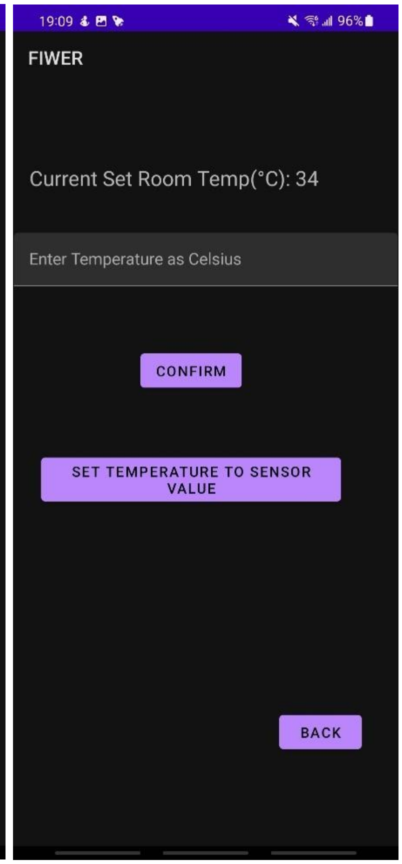


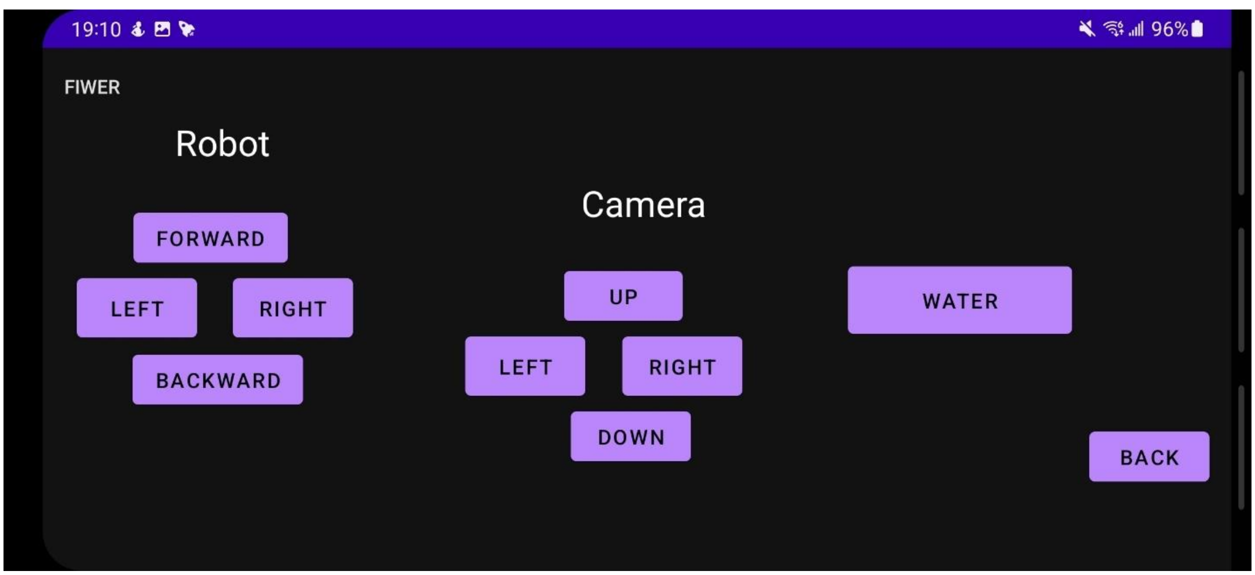
App

Once you open the app, you should see the `Login` page. Once you logged in, you will be directed to the `Home` page where the `Control Panel` page and `Set Room Temp Page` can be accessed.


19:05
97%
FIWER
Login
Name
Email
SUBMIT


19:09
96%
FIWER
Temperature(°C): 34
Fire Status: No
Set Room Temperature(°C): 34
CONTROL ME
SET ROOM TEMPERATURE
LOGOUT


19:09
96%
FIWER
Current Set Room Temp(°C): 34
Enter Temperature as Celsius
CONFIRM
SET TEMPERATURE TO SENSOR VALUE
BACK


19:10
96%
FIWER
Robot
FORWARD
LEFT
RIGHT
BACKWARD
Camera
UP
LEFT
RIGHT
DOWN
WATER
BACK

7 Appendix B: Class Diagrams

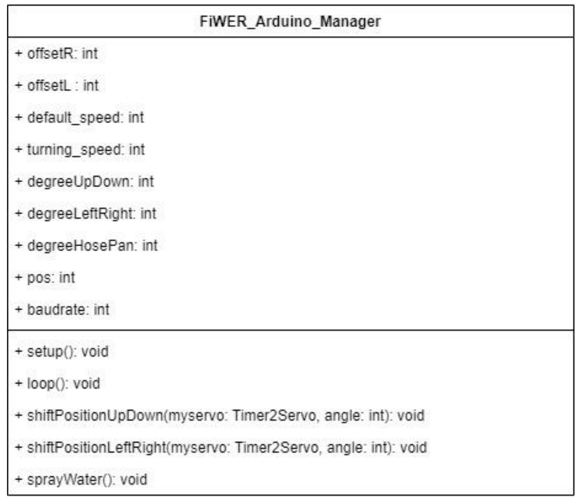


Figure 7: Class Diagram for Arduino Manager [Eline Elorm Nuviadenu]

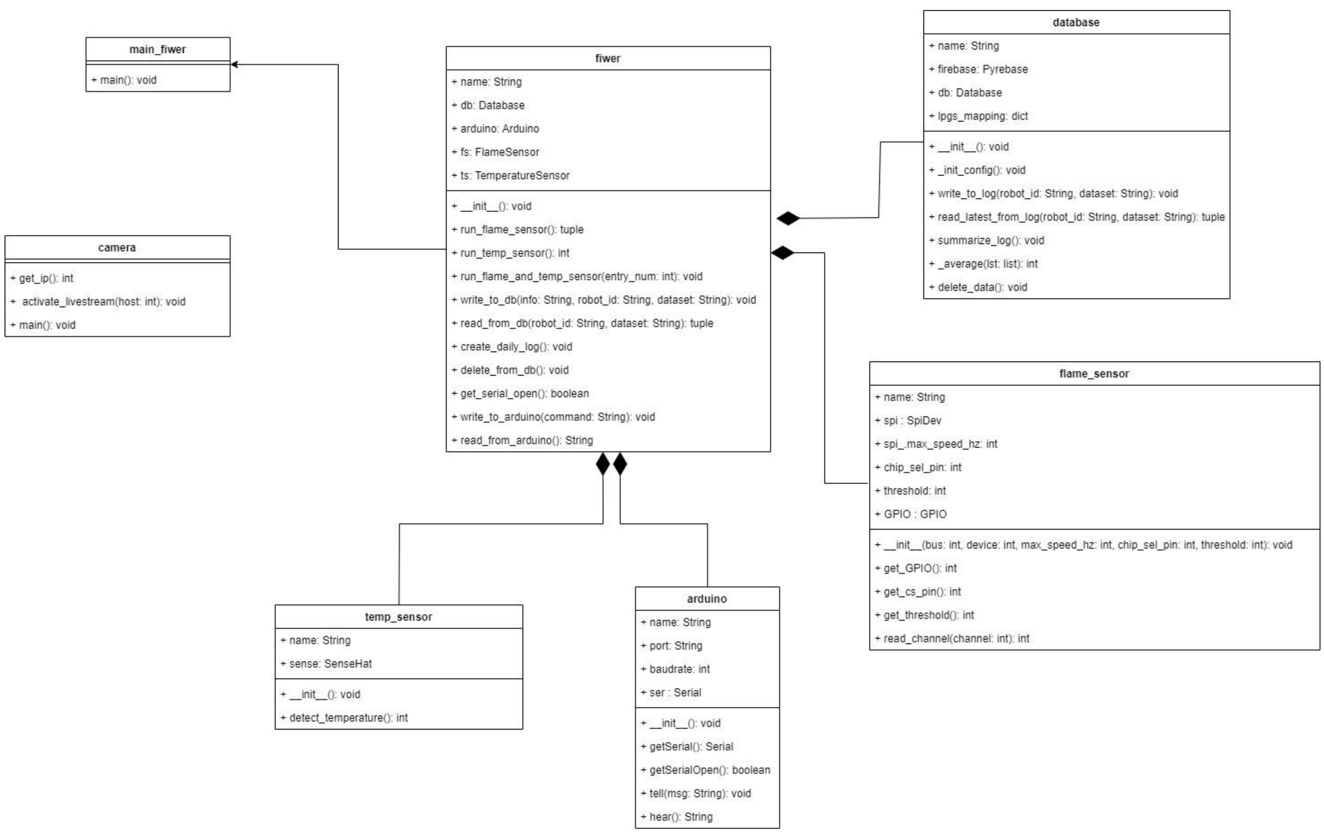


Figure 8: Class Diagram for the components of Raspberry Pi [Eline Elorm Nuviadenu]

8 Appendix C: Wiring Diagrams/Schematics

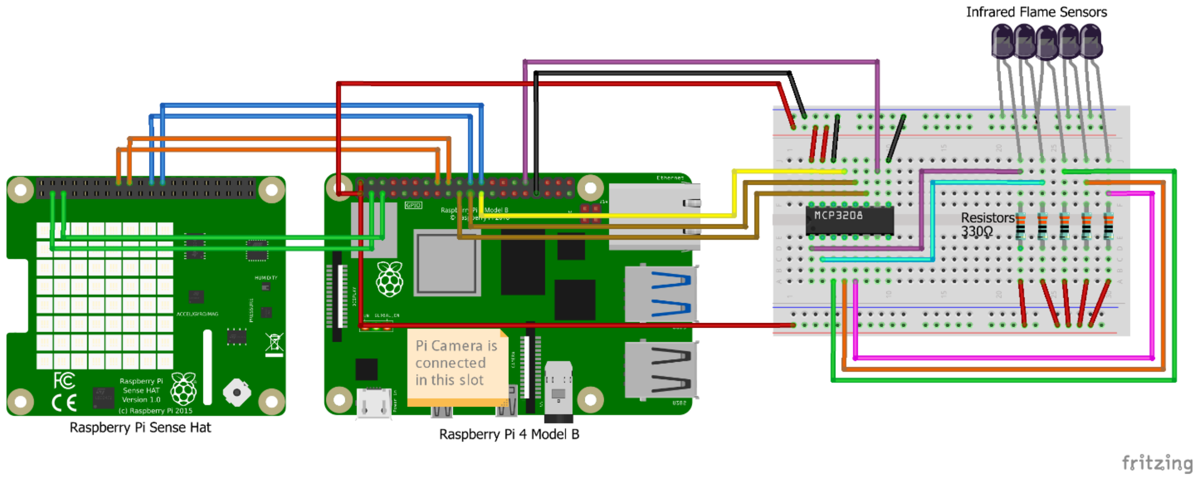


Figure 9: Breadboard view for the sensor connections to the RPi [Bren-Gelyn Padlan]

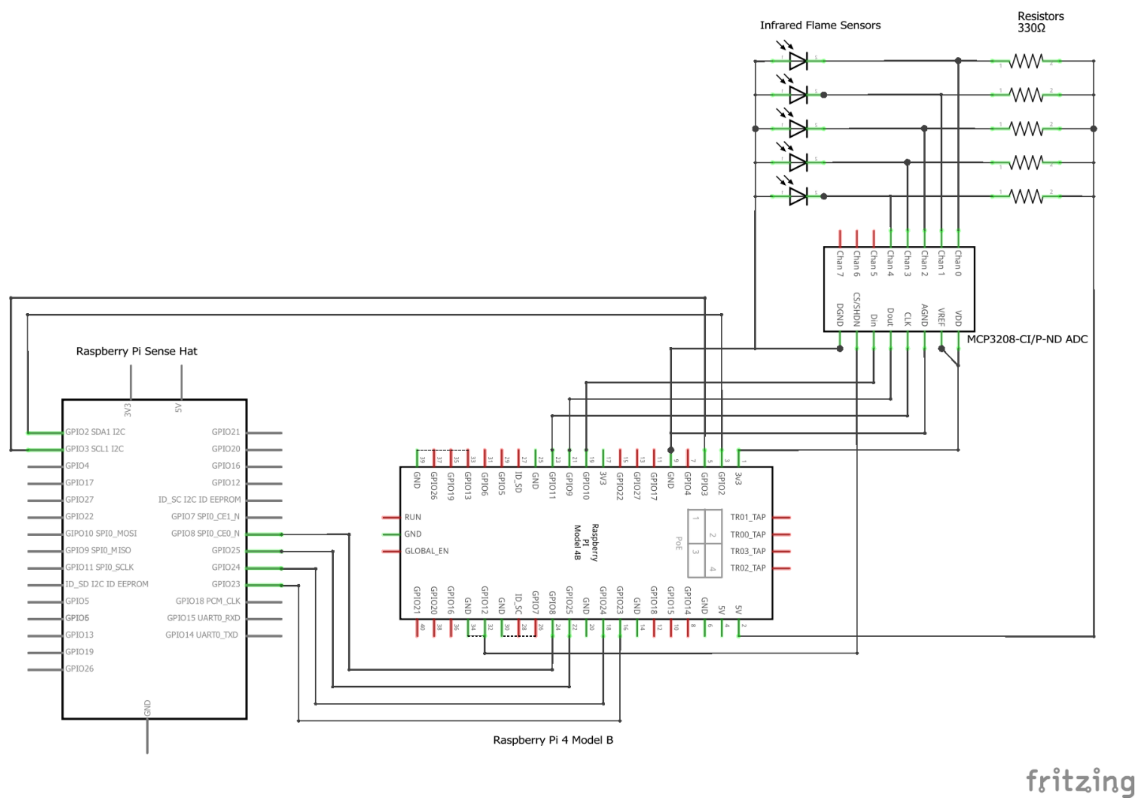


Figure 10: Schematic view of the sensor connections to the RPi [Bren-Gelyn Padlan]

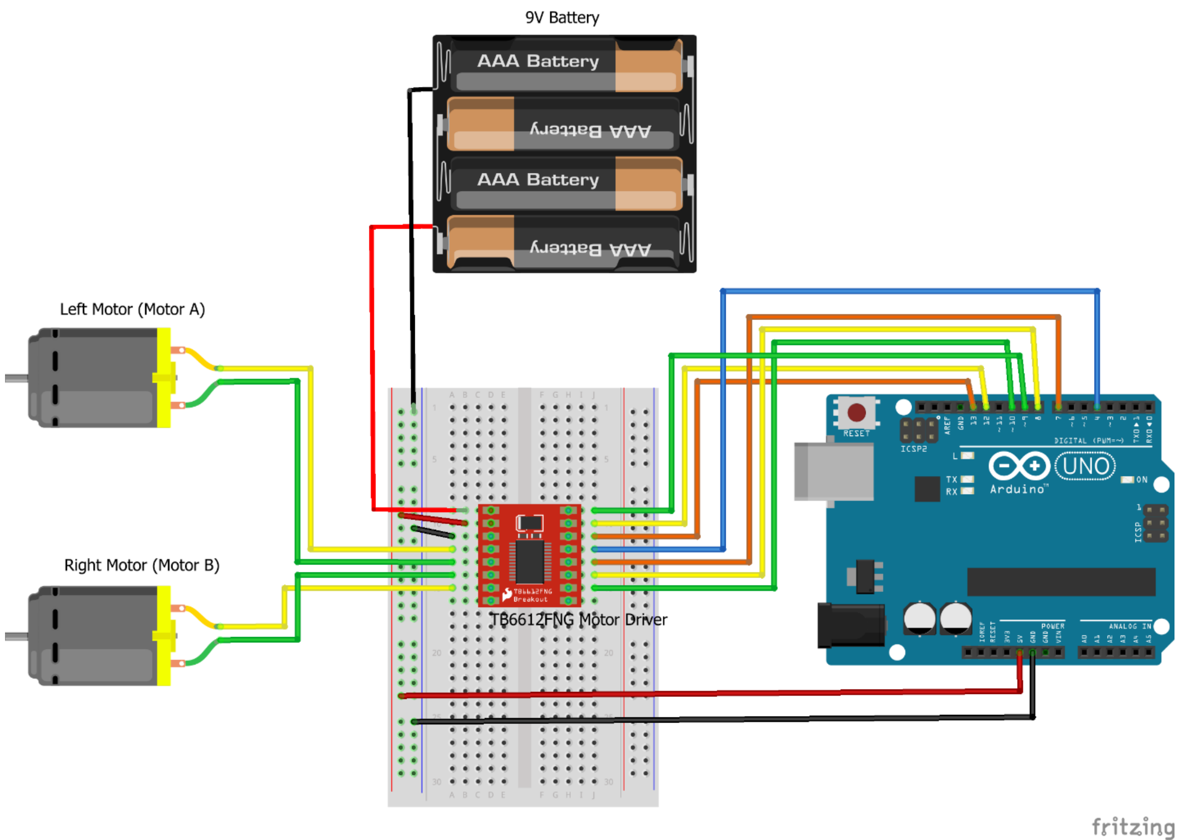


Figure 11: Breadboard view of the connections of the motor board and DC motors to the Arduino [Bren-Gelyn Padlan]

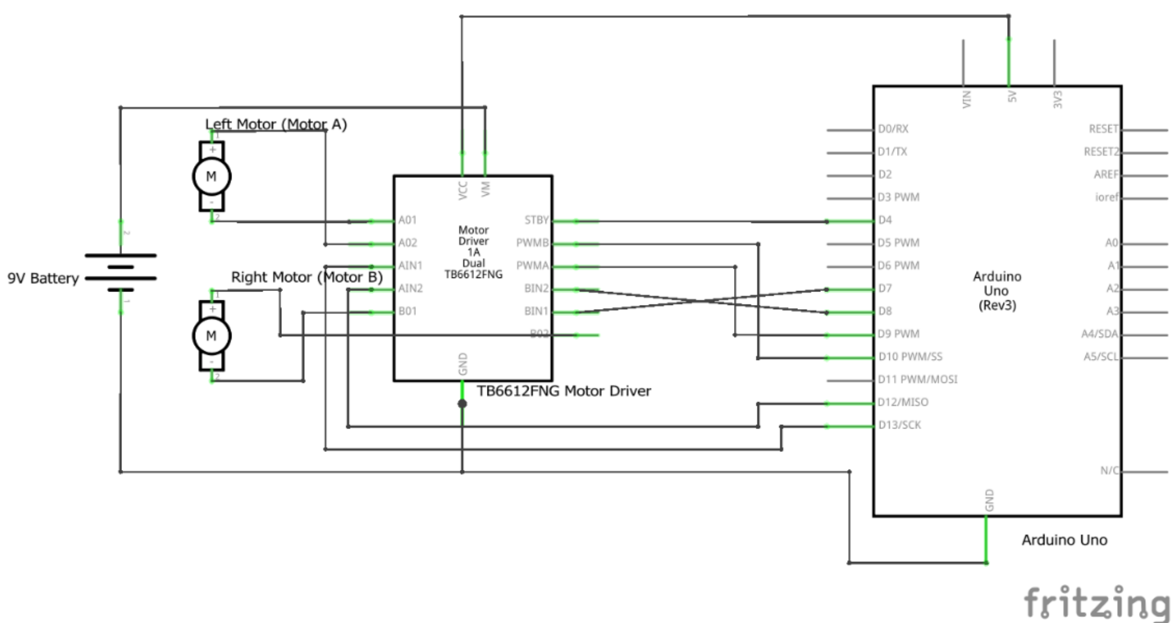


Figure 12: Schematic view of the motor board, DC motors and Arduino connections [Bren-Gelyn Padlan]

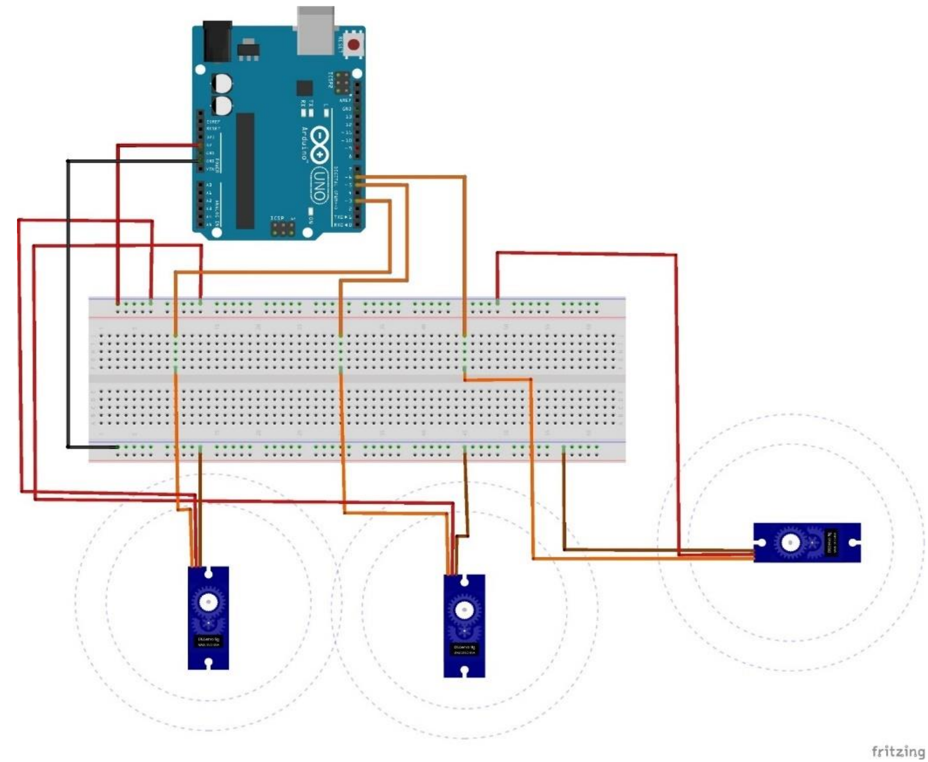


Figure 13: Breadboard view of the connections of servo motors to the Arduino [Eline-Elorm Nuviadenu]

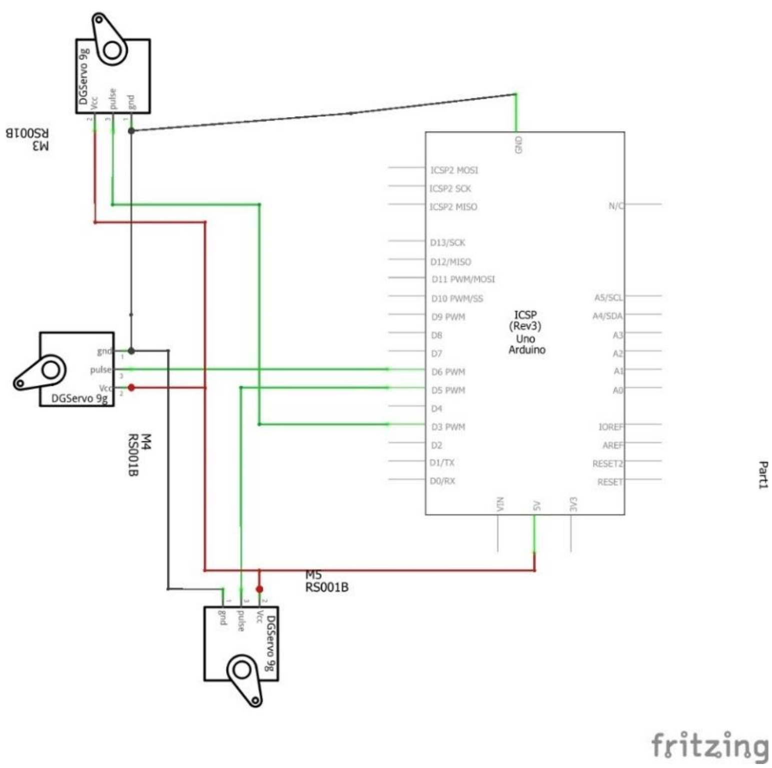


Figure 14: Schematic view of the servo motor and Arduino connections [Eline-Elorm Nuviadenu]

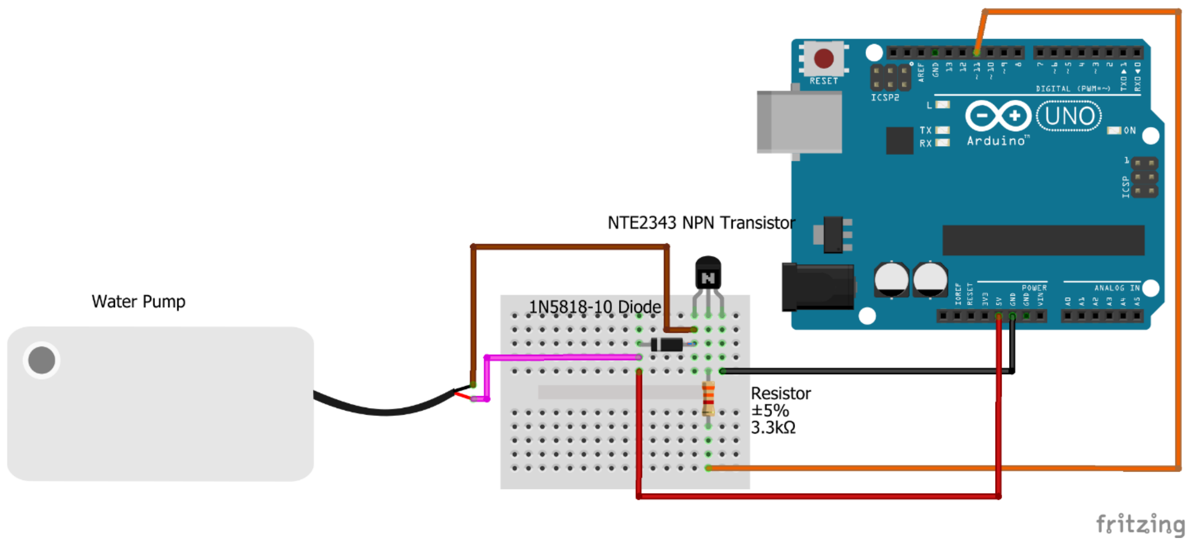


Figure 15: Breadboard view of the wiring connections for the water pump [Bren-Gelyn Padlan]

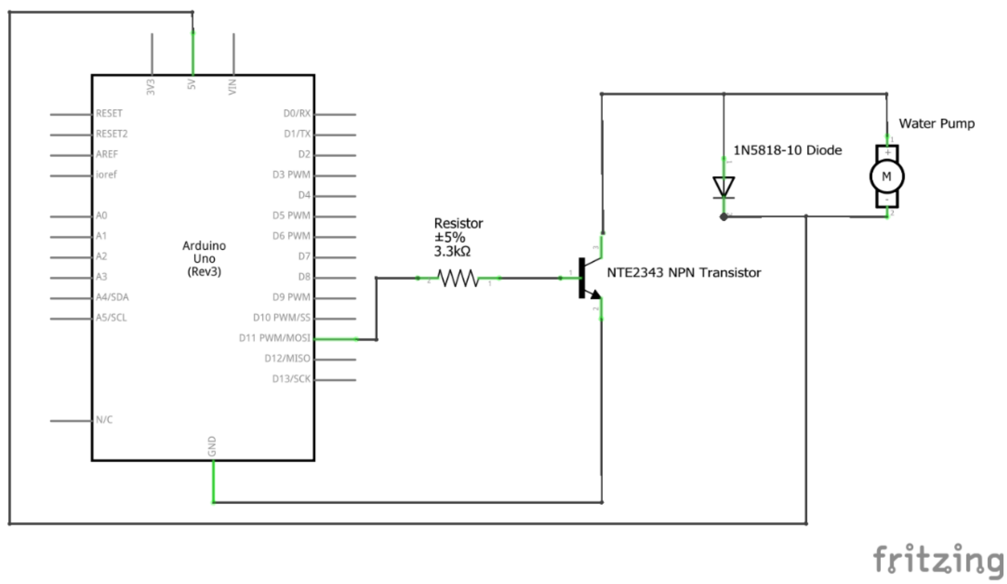


Figure 16: Schematic view of the wiring connections for the water pump [Bren-Gelyn Padlan]

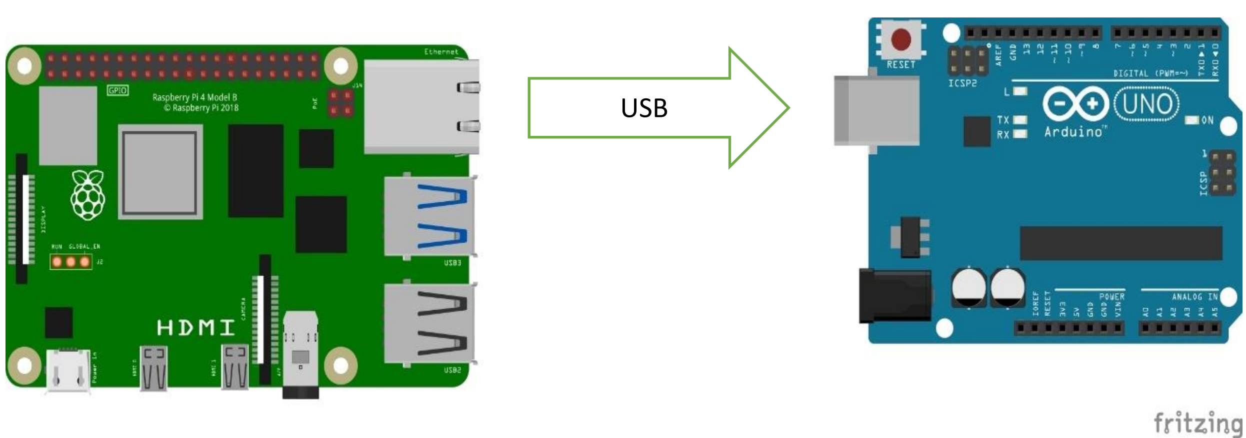


Figure 17: Serial connection between RPi and Arduino [Eline-Elorm Nuviadenu]

9 Appendix D: GUI Flowchart

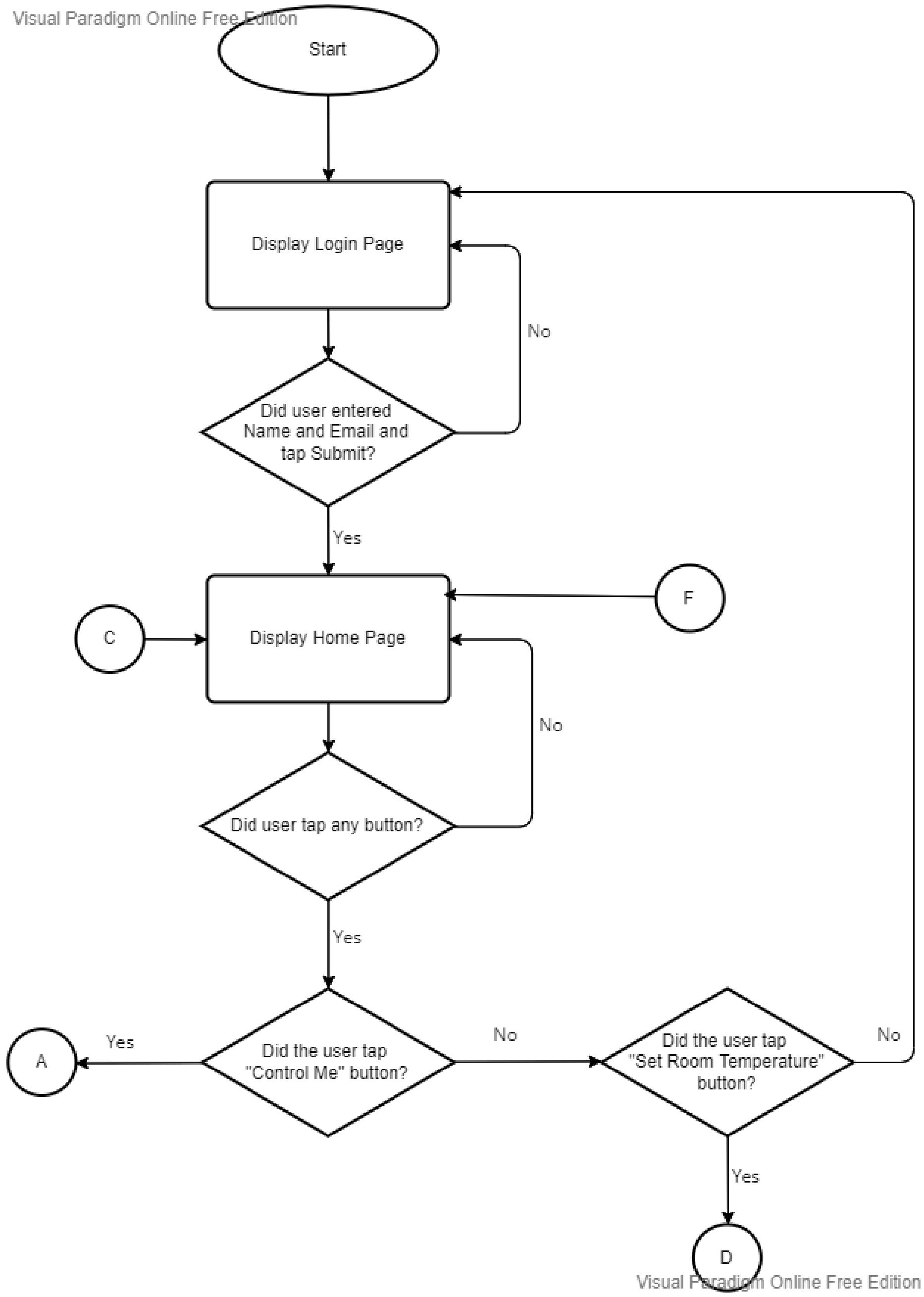


Figure 18: Main flowchart of FIWER GUI [Bren-Gelyn Padlan]

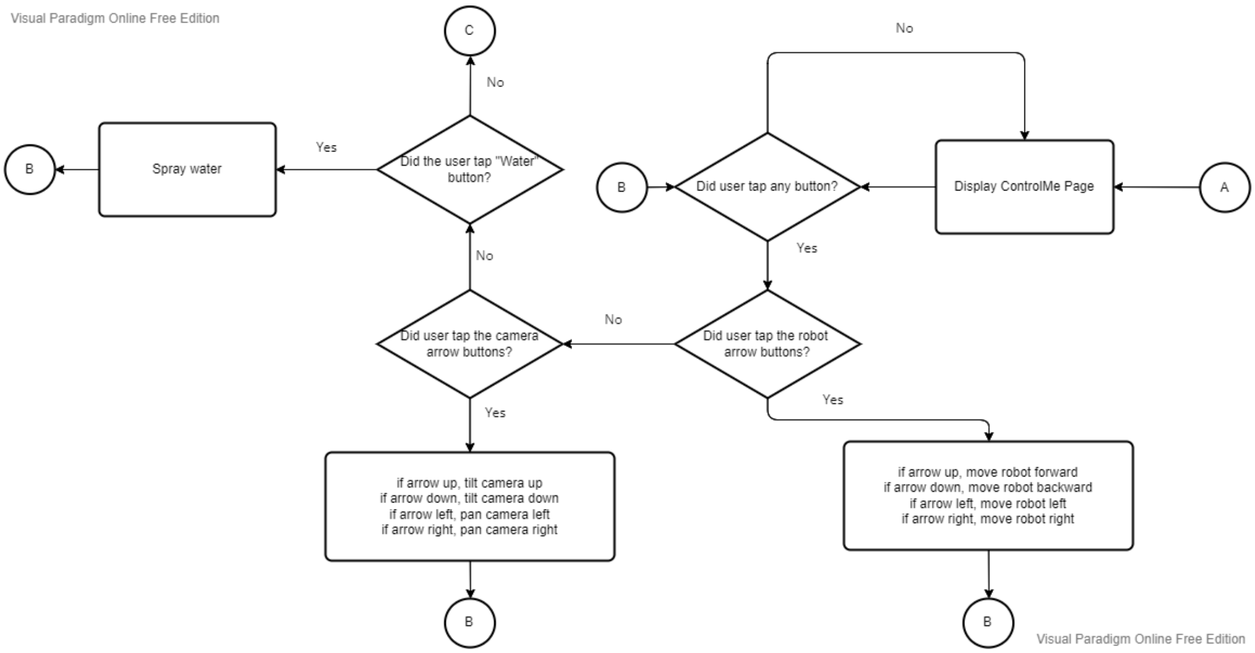


Figure 19: Sub-flowchart showing when ControlMe/ControlPanel Page is displayed [Bren-Gelyn Padlan]

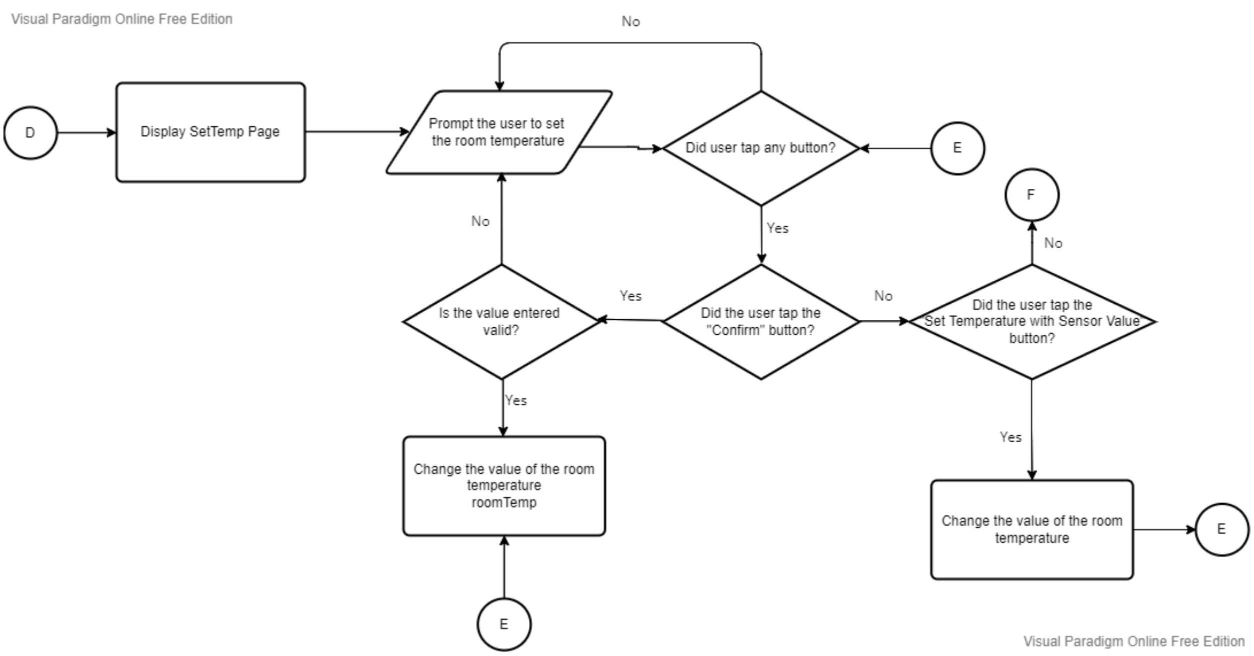


Figure 20: Sub-flowchart showing when SetRoomTemp Page is displayed [Bren-Gelyn Padlan]

10 References

[1] Help, H., Canada, E., Emergencies, T. and Fires, H., 2022. Home Fires: Information & Facts. [online] Red Cross Canada. Available: https://www.redcross.ca/how-we-help/emergencies-and-disasters-in-canada/types-of-emergencies/home-fires/home-fires-information-facts#:~:text=During%20winter%20months%2C%20heating%20equipment,to%20butt%20out%20a%20cigarette. [Accessed 8 February 2022].

[2] Canadian Firefighter Magazine. 2022. Study confirms cooking, smoking are top causes of house fires - Canadian Firefighter Magazine. [online] Available: https://www.cdnfirefighter.com/study-confirms-cooking-smoking-are-top-causes-of-house-fires-13275/. [Accessed 7 February 2022].

[3] "Ottawa firefighters sent to false alarms an average of 20 times a day,". [Online]. Available: https://www.cbc.ca/news/canada/ottawa/ottawa-firefighters-false-alarms-1.4615628. [Accessed: 13 March 2022].