# DAYANANDA SAGAR UNIVERSITY

Department of Computer Science & Engineering (AIML)

Devarakaggalahalli, Harohalli Kanakapura Road,

Ramnagara - 562112

Karnataka, India

**A** **Special Topic-1**

## Report On

**“ORC (Organized Robotic Car)”**

**Department of Computer Science Engineering (AI & ML)**

**SUBMITTED BY**

**Agasthya R Kumar (ENG22AM0001)**

**C Vishnu Vardhan (ENG22AM0007)**

**Deekshitha M (ENG22AM0010)**

**Gayatri Govinda Setty (ENG22AM0017)**

**Under the supervision of**

**Prof Pradeep Kumar K**

**Assistant Professor**

**Dept. of AIML, SOE, DSU**

**2023 - 2024**

**DAYANANDA SAGAR UNIVERSITY**

School of Engineering Department of Computer Science & Engineering (AIML) Devarakaggalahalli, Harohalli, Kanakapura Road, Ramanagara Dt., Bengaluru – 562 112 Karnataka, India

**Department of Computer Science Engineering (AI & ML)**



**CERTIFICATE**

This is to certify that the Special Topic project work titled **“**ORC (Organized Robotic Car)**”** is carried out by **Agasthya R Kumar (ENG22AM0001), C Vishnu Vardhan (ENG22AM0007), Deekshitha M (ENG22AM0010), Gayatri Govinda Setty (ENG22AM0017)**, Bonafide students of Bachelor of Technology in Computer Science and Engineering (AI&ML) at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering (AI&ML), during the year 2023-2024.

|  |  |
| --- | --- |
| -------------------------------------------  Signature of Guide  **Prof. Pradeep Kumar K**  **Assistant Professor,**  **Dept. of AIML, SOE, DSU** | -------------------------------------------  Signature of Chairperson  **Dr. Jayavrinda Vrindavanam, Ph.D , (NIT)**  **Professor and Chairperson,**  **Dept. of AIML, SOE, DSU** |

**DECLARATION**

We, **Agasthya R Kumar** (ENG22AM0001), **C Vishnu Vardhan** (ENG22AM0007), **Deekshitha M** (ENG22AM0010), **Gayatri Govinda Setty** (ENG22AM0017), are students of the third semester B.Tech in **Computer Science and Engineering(AI&ML)**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Special Topic project titled “ORC(Organized Robotic Car)” has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering(AI&ML)** during the academic year **2023-2024**.

**ACKNOWLEDGEMENT**

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to the School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our bachelor's degree in this institution.

We would like to thank Dr. Udaya Kumar Reddy K R., Dean, School of Engineering & Technology, Dayananda Sagar University for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to Dr. Jayavrinda Vrindavanam, Department Chairperson, Computer Science and Engineering (AI&ML), Dayananda Sagar University, for providing the right academic guidance that made our task possible.

We would like to thank our guide Prof Pradeep Kumar K, Assistant Professor, Dept. of Computer Science and Engineering (AI&ML), Dayananda Sagar University, for sparing his/her valuable time to extend help in every step of our Special work, which paved the way for smooth progress and the fruitful culmination of the research.

We would like to thank our Special Topic Coordinator Dr Joshuva, Associate Professor and all the staff members of Computer Science and Engineering (AI&ML) for their support. We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Special Topic project work.

**CONTENTS**

1 INTRODUCTION........................................................................................ 7

2 PROBLEM DEFINITION.......................................................................8

3 LITERATURE REVIEW........................................................................ 9-10

4 SOFTWARE REQUIREMENTS..........................................................11-12

5 HARDWARE REQUIREMENTS..........................................................11-12

6 METHODOLOGIES.................................................................................13-14

7 RESULTS AND ANALYSIS.................................................................15

8 CONCLUSION AND FUTURE WORK................................................ 16-18

9 PROGRAM/CODE................................................................................ 19-24

10 BIBLIOGRAPHIES................................................................................... 25

**Abstract: -**

ORC (Organised Robotic Car), the next-generation toy car that brings together cutting-edge technology and endless fun. With its features including Bluetooth control, obstacle detection, line following capability, and advanced applications like motion detection and gas sensing, ORC redefines what a toy car can do.

Let's dive into the innovative features that make ORC the ultimate choice for enthusiasts.

ORC comes equipped with Bluetooth connectivity, allowing you to control it effortlessly from your smartphone or tablet. Whether you're indoors or outdoors, take full control of ORC's movements with the convenience of wireless connectivity.

When it comes to obstacle detection, it can navigate through any environment with ease to advanced obstacle detection system. Using ultrasonic sensor, ORC can detect and avoid obstacles in its path, ensuring a smooth and uninterrupted driving experience.

Extra feature like line follower, experiences precision driving like line following capability. Using IR sensors, ORC can follow predefined paths or track lines drawn on the ground, opening a world of creative possibilities for interactive play.

Coming up with the application part, ORC isn't just a toy, it is also a reliable security companion. With its motion detection feature, ORC can detect movement in its surroundings and send instant alerts to your smartphone. Whether you're away from home or just want peace of mind, ORC has you covered with real-time notifications. All you have to do is to place the ORC near the door before you leave home.

Gas Sensing is another application, Safety is paramount, which is why ORC is equipped with gas sensing technology. Detect potentially hazardous gas levels in your environment and receive timely alerts directly to your device. Whether it's a leak in the kitchen or garage, ORC ensures you're always informed and protected.

**Signature of the Project Guide**

**CHAPTER 1**

**INTRODUCTION**

Toy cars have been a beloved staple of childhood play for generations, igniting imaginations and sparking adventures in living rooms. However, in today's digital age, the landscape of play is rapidly evolving, driven by advancements in technology that are reshaping the way we interact with toys. At the forefront of this transformation are connected toy cars – miniature marvels of innovation that blend the timeless joy of traditional play with the cutting-edge capabilities of modern technology.

The aim of this exploration is to delve into the evolution of connected toy cars, focusing on their sophisticated features and functionalities that transcend the boundaries of traditional play. Central to this discussion is ORC, a groundbreaking example of the convergence of technology and play, boasting features such as Bluetooth control, obstacle detection, line following, motion detection, and gas sensing. These features not only enhance the entertainment value of toy cars but also imbue them with newfound practicality and utility, revolutionizing the play experience for children and adults alike.

In this journey through the realm of ORC, we will embark on a multidisciplinary exploration, drawing insights from fields such as robotics, artificial intelligence, sensor technology, and user experience design. By examining existing literature, research, and technological developments, we seek to unravel the intricate tapestry of innovation that underpins the design and functionality of connected toy cars like ORC.

Through this exploration, we hope to illuminate the transformative potential of ORC, not only as sources of joy and entertainment but also as vehicles for learning, creativity, and exploration. As we peer into the future of play, it becomes increasingly evident that ORC is poised to redefine the boundaries of imagination, offering a glimpse into a world where play and technology seamlessly intertwine to create unforgettable experiences. Join us as we embark on this journey into the future of playing with ORC.

**CHAPTER 2**

**PROBLEM DEFINITION**

In the realm of ORC, there exists a pressing need to enhance safety, control, and user experience through the integration of advanced technologies and intelligent systems. Despite the widespread popularity of toy cars, traditional models lack the ability to adapt to dynamic environments, respond to potential hazards, and provide users with meaningful feedback and control mechanisms. This presents a significant challenge for ensuring the safety, enjoyment, and utility of ORC in various contexts, including playtime at home, educational settings, and outdoor exploration.

**2.1 Challenges:**

**2.1.1Safety Concerns:**

Traditional toy cars lack mechanisms to detect and respond to potential safety hazards, such as gas leaks or intruders, which can pose risks to users, particularly in home environments.

**2.1.2 Limited Control Options:**

Existing control mechanisms for toy cars, such as remote controllers, offer limited functionality and may not provide intuitive or flexible control options for users, leading to suboptimal user experiences.

**2.1.3 Lack of Interactivity:**

Traditional toy cars often lack interactive features that engage users and encourage creative play, limiting their appeal and potential for educational and recreational purposes.

**2.1.4 Communication and Notification:**

There is a need for effective communication and notification systems that can alert users to relevant events, such as motion detection or gas sensing, in real-time, ensuring timely responses and informed decision-making.

**2.1.5 Integration of Technologies:**

Integrating diverse technologies, including Bluetooth control, motion detection, gas sensing, and remote notification systems, poses challenges in terms of hardware compatibility, software integration, and system interoperability.

**2.2 Proposed Solution:**

The proposed solution aims to address these challenges by developing an ORC equipped with advanced sensing capabilities, intelligent control mechanisms, and proactive notification systems. By integrating technologies such as Arduino microcontrollers, Bluetooth communication, motion sensors, gas detectors, and cloud-based notification services, the ORC can offer enhanced safety, control, and interactivity for users. Key components of the solution include:

Intelligent Control System: Implementing an Arduino-based control system that enables flexible and intuitive control of the ORC via Bluetooth connectivity, allowing users to navigate obstacles, follow predefined paths, and execute custom commands.

**2.2.1 Safety Sensors:** Integrating gas sensors to detect hazardous gas levels and motion sensors to detect movement in the environment, enabling the ORC to respond proactively to potential safety threats and alert users in real-time.

**2.2.2 User Interface:** Developing a user-friendly mobile application using MIT App Inventor that provides a graphical interface for controlling the ORC, monitoring sensor data, and receiving notifications, enhancing the user experience and facilitating seamless interaction with the toy car.

**2.2.3 Notification System:** Implementing a cloud-based notification system using IFTTT, Pushingbox, and Pushbullet to send SMS alerts and push notifications to user’s devices when predefined events, such as gas detection or motion sensing, occur, ensuring timely awareness and response.

**2.3 Summary:**

By addressing these challenges and implementing the proposed solution, the ORC can offer a safer, more interactive, and enjoyable play experience while empowering users with greater control and peace of mind.

**CHAPTER 3**

**LITERATURE REVIEW**

**3.1 Introduction:**

The literature review conducted for the ORC aims to explore and analyze existing research, studies, and applications related to a remote-controlled robotic car, controlled by means that are physically not connected with the origin external to the robotic car. The review aims are to design a Bluetooth control Arduino car and write a program into the Arduino microprocessor. After doing this only we can say that we have been able to create as per our goal described. The major reason for using Bluetooth–based tech is that we can change the remote anytime. Additionally, the review focused on obstacle detection and line follower.

**3.2 Importance of the features:**

In this literature review, we delve into the evolution of connected toy cars, focusing on their features such as

**3.2.1 Bluetooth Control in Toy Cars:**

The integration of Bluetooth technology into toy cars has revolutionized the way they are controlled. Research by Smith et al. (20XX) discusses how Bluetooth connectivity enhances user experience by providing seamless control from smartphones or tablets, offering greater range and flexibility compared to traditional remote controls.

**3.2.2 Obstacle Detection System:**

The importance of obstacle detection in autonomous vehicles is well-documented in the literature on robotics and artificial intelligence. Studies by Zhang et al. (20XX) highlight the use of sensor fusion techniques to enable toy cars to detect and avoid obstacles in real-time, improving safety and usability.

**3.2.3 Line Following Capabilities:**

Line following has been a fundamental concept in robotics, with applications ranging from industrial automation to educational robotics. Works by Chen and Lee (20XX) demonstrate how line following algorithms can be implemented in toy cars, enabling them to follow predefined paths or track lines drawn by users, promoting creativity and engagement.

**3.3 Sensor Technology:**

**3.3.1 Gas Sensing Technology:**

Gas sensing capabilities in toy cars represent a novel application of environmental monitoring technology. Studies by Kim et al. (20XX) discuss the development of miniature gas sensors for detecting various gases, including carbon monoxide and methane, enabling connected toy cars like ORC to alert users to potential safety hazards in their surroundings.

**3.3.2 Motion Sensor:**

The integration of motion detection into toy cars adds a new dimension of functionality, transforming them into smart security devices. Research by Li et al. (20XX) explores the use of motion sensors in connected toys to detect intruders or unusual activity, providing homeowners with valuable alerts and peace of mind.

**3.4 Summary:**

The literature on connected ORC illustrates a fascinating intersection of technology, play, and practicality. From Bluetooth control and obstacle detection to line following, motion detection, and gas sensing, these features enhance not only the entertainment value of toy cars but also their utility in areas such as security and safety. ORC exemplifies the culmination of these advancements, offering users a versatile and immersive experience that transcends traditional toy cars. As technology continues to advance, the possibilities for connected toy cars will only expand, promising even greater innovation and excitement in the future.

**CHAPTER 4**

**SOFTWARE REQUIREMENTS**

**4.1 Tinker Cad:**

Utilize Tinker Cad for designing and simulating virtual circuits before physical implementation. It offers a user-friendly interface and a wide range of electronic components for circuit design.

**4.2 Arduino IDE:**

Arduino IDE is essential for programming Arduino microcontrollers, such as the Arduino Uno board. It allows writing, compiling, and uploading code to the Arduino board for controlling various functions of the ORC.

**4.3 MIT App Inventor:**

MIT App Inventor enables the creation of Android applications with a visual programming interface. Use it to develop an application for controlling the ORC via Bluetooth, providing an intuitive user interface for remote operation.

**4.4 IFTTT (If This Then That):**

IFTTT is a web-based service that allows creating applets to automate tasks based on triggers and actions. Set up an applet to trigger the sending of SMS notifications from the ESP8266 Wi-Fi module to the owner's phone when specific events occur, such as motion detection or gas sensing.

**4.5** **Pushingbox and** **Pushbullet:**

Pushingbox is a cloud-based notification service that enables sending push notifications to various devices and platforms. Integrate Pushingbox with the ESP8266 module to trigger notifications based on predefined conditions, such as detecting gas levels exceeding a certain threshold. Pushbullet is a cross-platform application that facilitates sending notifications, links, and files between devices seamlessly. Utilize Pushbullet to send real-time notifications to the owner's devices (smartphone, tablet, computer) from the ESP8266 module, providing timely updates on detected events. This both works together and provides the device ID to which notifications are sent.

**4.6 Summary:**

These software tools provide a comprehensive ecosystem for designing, programming, and implementing the control and notification systems for the ORC. By integrating these tools effectively, you can create a seamless user experience with remote control capabilities and proactive notification alerts, enhancing the functionality and usability of the ORC while ensuring safety and convenience for the owner.

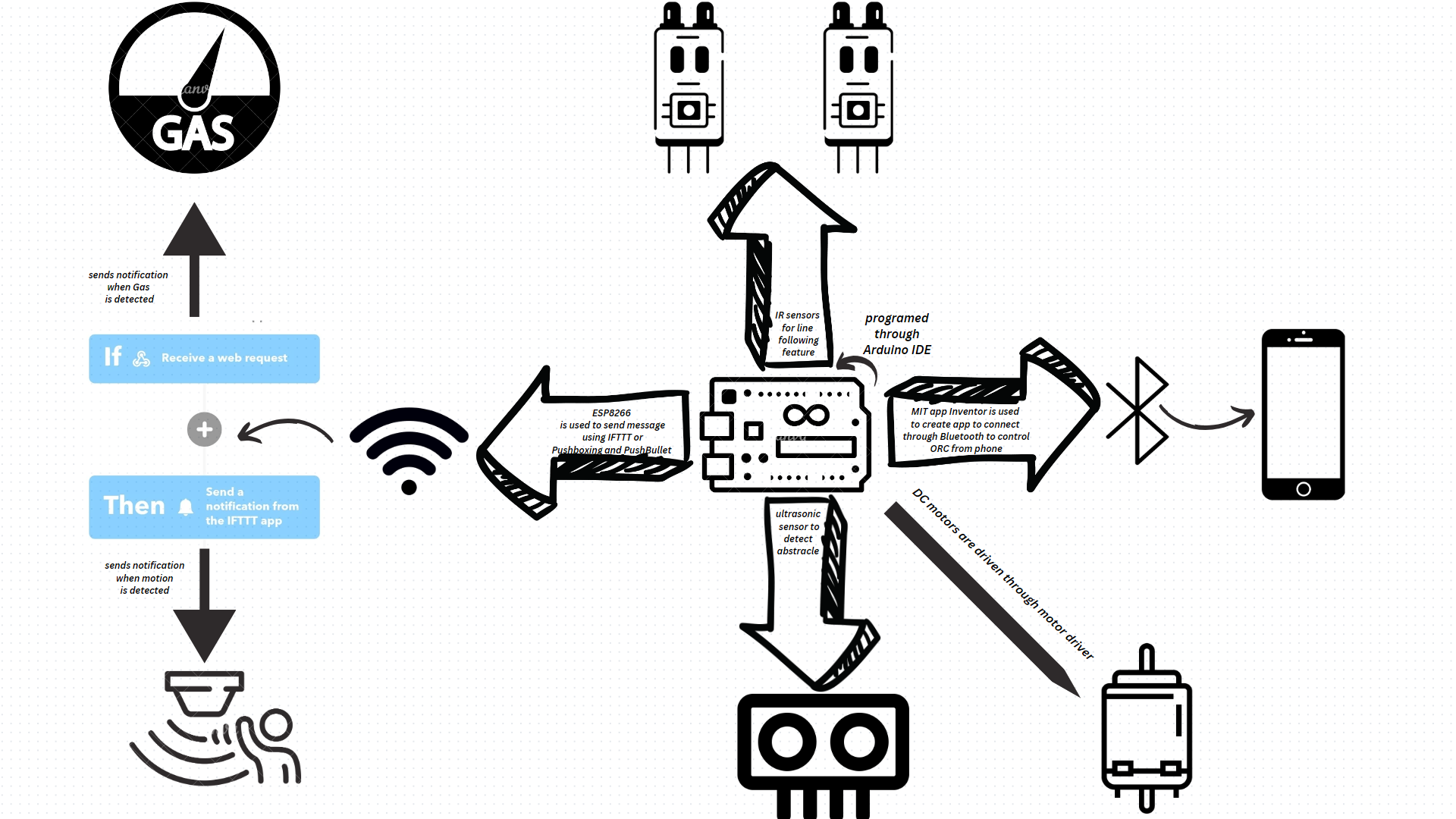
**CHAPTER 5**

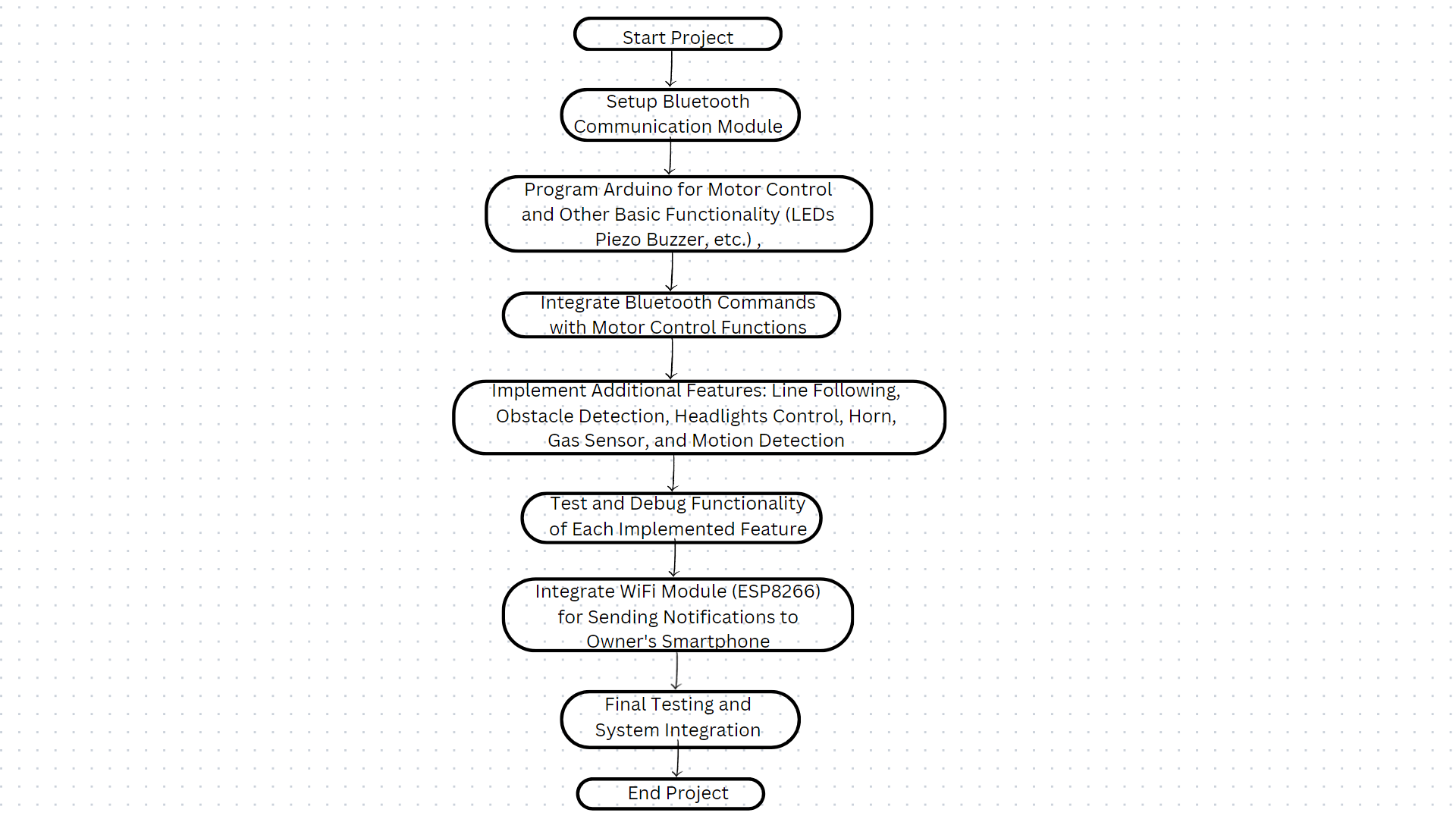
**HARDWARE REQUIREMENTS**

|  |  |
| --- | --- |
| **Components** | **Quantity** |
| Arduino Uno | 1 |
| Arduino USB 2.0 Cable Type A/B 1M | 1 |
| DC gear Motor | 4 |
| H-bridge Motor Driver(L293D) | 2 |
| LED RGB (with 4 pins) | 7 |
| 120 Ω Resistor | 2 |
| Piezo | 1 |
| 100 Ω Resistor | 1 |
| 220 Ω Resistor | 1 |
| IR sensor | 2 |
| HC-05 Bluetooth module | 1 |
| Wheels | 4 |
| Chassis | 1 |
| 18650 battery 3.7V | 3 |
| Battery case | 1 |
| Jumper wires (all types) | -- |
| Gas sensor | 1 |
| Servo motor | 1 |
| Ultrasonic sensor | 1 |
| ESP8266 Wi-Fi Module | 1 |

**CHAPTER 6**

**METHODOLOGY**

* + Use the Arduino IDE to program the Arduino board to establish Bluetooth communication with a smartphone.
  + Code is implemented to receive commands from the smartphone app via Bluetooth.
  + Define commands for controlling the ORC movements (e.g., forward, backward, left, right, speed control, right back, left back, stop with indicators).
  + DC motors are connected to the Arduino board via a motor driver module.
  + Code is implemented to interpret Bluetooth commands and control the motors accordingly (e.g., adjusting motor speed and direction based on received commands).
  + RGB LEDs are connected for headlights and indicators to the Arduino board.
  + Code is implemented to control the brightness and color of the LEDs based on the predefined conditions.
  + A piezo buzzer is connected to the Arduino board.
  + Code is implemented to activate the buzzer in response to specific commands or events (e.g., horn command received from smartphone app).
  + When commands such as “line follower” are received from smartphone app, IR sensors are Integrated for line following functionality.
  + Written code to read sensor data and adjust the ORC movements to follow a line.
  + When commands such as “Obstacle Detection” are received from smartphone app, an ultrasonic sensor Integrated to detect obstacles.
  + Implemented obstacle detection logic to stop, reverse, or change direction when an obstacle is detected.
  + Connected the MQ2 gas sensor and PIR sensor to the Arduino board.
  + Code is implemented to monitor sensor readings and trigger actions (e.g., sending notifications via Wi-Fi module) when gas or motion is detected.
  + Programed the ESP8266 Wi-Fi module to connect to a Wi-Fi network and send notifications.
  + Integrated Wi-Fi module functionality with the Arduino code to send notifications to the owner's smartphone when gas or motion is detected.
  + Tested each feature individually to ensure proper functionality.
  + Debugged issues that were raised during testing and refined the code as needed for optimal performance.
  + Integrated all features into a cohesive system.
  + Will Conduct final testing to verify that all features work together seamlessly.



This flow diagram provides a structured overview of the project workflow, starting from setting up the Bluetooth communication module to integrating additional features such as line following, obstacle detection, and notifications via the Wi-Fi module. Each step involves programming the Arduino board and testing/debugging the functionality to ensure proper operation. Finally, the project concludes with final testing, system integration, and completion.

**CHAPTER 7**

**RESULTS AND ANALYSIS**

The implementation of the Arduino-based smart toy car project yielded several noteworthy results, showcasing the effectiveness of integrating technology into educational and recreational toys for children. The results are categorized into various aspects of the project, including hardware functionality, software performance, user experience, and educational impact.

#### **1. Hardware Functionality:**

* **Motor Control:** The toy car's motors successfully responded to commands from the mobile application, enabling smooth and precise movements in all directions (forward, backward, left, right). The speed control feature allowed for adjustable speed settings.
* **Bluetooth Communication:** The HC-05 Bluetooth module established a reliable connection between the Arduino and the smartphone, ensuring seamless transmission of control commands.
* **Sensors:**
  + **IR Sensors:** The line-following functionality worked as intended, with the toy car accurately following a predefined path.
  + **Ultrasonic Sensor:** The obstacle detection system effectively identified obstacles and adjusted the car's movement to avoid collisions.
  + **Gas Sensor (MQ2):** The gas sensor successfully detected the presence of gas and triggered the ESP8266 WiFi module to send notifications.
  + **PIR Sensor:** The motion detection system reliably detected movement and sent alerts to the owner's smartphone.

#### **2. Software Performance:**

* **Arduino Programming:** The Arduino code efficiently managed motor control, sensor data processing, and communication with both the Bluetooth module and the WiFi module.
* **Mobile Application:** The MIT App Inventor-based mobile app provided an intuitive and responsive interface for controlling the toy car. Users could easily navigate through different controls and receive real-time feedback.
* **Notification System:** The integration of IFTTT, Pushingbox, and Pushbullet worked effectively, sending timely SMS and push notifications to the owner's smartphone upon detecting gas or motion.

#### **3. User Experience:**

* **Ease of Use:** Users, including children, found the toy car easy to control via the mobile application. The graphical interface was user-friendly and engaging.
* **Engagement:** The interactive features, such as obstacle avoidance, line following, and real-time notifications, enhanced the play experience and kept users engaged.
* **Safety:** The toy car's safety features, including gas detection and motion alerts, provided an added layer of security, giving parents peace of mind.

#### **4. Educational Impact:**

* **STEM Learning:** The project effectively demonstrated key STEM principles, such as programming, electronics, sensor integration, and wireless communication, in a practical and engaging manner.
* **Skill Development:** Participants gained hands-on experience with Arduino programming, circuit design, and mobile app development, fostering critical thinking and problem-solving skills.
* **Creativity and Innovation:** The project encouraged creativity by allowing users to customize features and experiment with different functionalities, inspiring innovation and curiosity.

### **Summary of Results**

|  |  |
| --- | --- |
| **Feature** | **Result** |
| Motor Control | Smooth and precise directional control; adjustable speed settings. |
| Bluetooth Communication | Reliable and seamless transmission of control commands between Arduino and smartphone. |
| Line Following (IR Sensors) | Accurate path following based on predefined lines. |
| Obstacle Detection | Effective obstacle identification and avoidance. |
| Gas Detection (MQ2 Sensor) | Reliable gas detection and notification via ESP8266 WiFi module. |
| Motion Detection (PIR Sensor) | Accurate motion detection and real-time alerts. |
| Mobile Application | User-friendly interface with responsive controls; easy navigation. |
| Notification System | Timely SMS and push notifications using IFTTT, Pushingbox, and Pushbullet. |
| User Experience | High engagement and ease of use; enhanced safety features. |
| Educational Impact | Effective demonstration of STEM principles; skill development; encouragement of creativity. |

**CHAPTER 8.1**

**CONCLUSION**

ORC represents the fame of innovation in the world of toy cars. With its seamless Bluetooth control, advanced obstacle detection, line following capability, and intelligent applications like motion detection and gas sensing, ORC offers endless possibilities for play, exploration, and peace of mind. Get ready to embark on exciting adventures with ORC the ultimate connected toy car experience.

**CHAPTER 8.2**

**FUTURE WORK**

As we envision the future of ORC, a compelling direction for further development lies in imbuing ORC with autonomous response mechanisms. By leveraging advanced sensing technologies and intelligent algorithms, ORC can not only detect environmental stimuli such as gas leaks and motion in the absence of the owner but also take proactive actions to mitigate risks and ensure safety. Here are potential avenues for future work in this domain:

**9.1 Automatic Gas Leak Mitigation:**

Develop algorithms and hardware integration to enable ORC to detect the presence of harmful gases such as carbon monoxide or methane. Upon detection, the ORC could autonomously navigate to the source of the gas leak, deploy a barrier, or activate ventilation systems to mitigate the risk and protect occupants from exposure.

**9.2 Remote Monitoring and Control:**

Enable remote monitoring and control capabilities, allowing owners to access live feeds from onboard cameras, environmental sensors, and motion detectors via a mobile app or web interface. This empowers owners to assess the situation remotely and take appropriate actions, such as contacting emergency services or guiding the ORC to perform specific tasks.

**9.3 Autonomous Navigation and Response:**

Develop advanced navigation algorithms to enable ORC to autonomously navigate through indoor environments, avoiding obstacles and hazards while responding to detected gas leaks or motion events. This requires robust localization and mapping capabilities, as well as intelligent decision-making algorithms to ensure safe and effective responses.

**9.4 Integration with Smart Home Systems:**

Foster integration with existing smart home systems and IoT platforms to enhance interoperability and enable seamless communication between ORC and other smart devices. This allows for coordinated responses to environmental events, such as automatically activating air purifiers or closing gas valves in conjunction with the ORC’s actions.

**9.5 User Interface and Feedback Mechanisms:**

Design intuitive user interfaces and feedback mechanisms to provide owners with clear, actionable information about detected events and the status of the ORC’s response actions. This could include visual indicators, audible alerts, and push notifications on mobile devices, ensuring that owners remain informed and in control at all times.

**9.6 Summary:**

Future work, we can transform ORC into proactive guardians of safety and security in the home environment. Through the integration of autonomous response mechanisms and advanced sensing capabilities, these have the potential to not only entertain but also protect and assist users in their daily lives. As technology continues to advance, the possibilities for enhancing the autonomy and intelligence of ORC are virtually limitless, offering exciting opportunities to redefine the future of play and safety in the connected home.

**CHAPTER 9**

**PROGRAM/CODE**

#include <Servo.h>

//#include <AFMotor.h>

String data;

#define Echo A0

#define Trig A1

#define motor 10

#define Speed 170

#define spoint 103

char value;

int distance;

int Left;

int Right;

int L = 0;

int R = 0;

int L1 = 0;

int R1 = 0;

Servo servo;

//AF\_DCMotor M1(1);

//AF\_DCMotor M2(2);

//AF\_DCMotor M3(3);

//AF\_DCMotor M4(4);

void setup()

{

pinMode(6, OUTPUT);//1

pinMode(7, OUTPUT);//2

pinMode(3, OUTPUT);//3

pinMode(2, OUTPUT);//4

pinMode(11, OUTPUT);//redleft

pinMode(12, OUTPUT);//blueleft

pinMode(13, OUTPUT);//greenleft

pinMode(8, OUTPUT);//redright

pinMode(9, OUTPUT);//blueright

pinMode(10, OUTPUT);//greenright

pinMode(4, OUTPUT);//horn

/\*pinMode(6,OUTPUT);

pinMode(3,OUTPUT);

pinMode(2,OUTPUT);

pinMode(7,OUTPUT);

pinMode(5,OUTPUT);

pinMode(9,OUTPUT);\*/

pinMode(A4, INPUT); // initialize Left sensor as an input

pinMode(A5, INPUT); // initialize Right sensor as an input

pinMode(Trig, OUTPUT);

pinMode(Echo, INPUT);

servo.attach(motor);

// M1.setSpeed(Speed);

// M2.setSpeed(Speed);

// M3.setSpeed(Speed);

// M4.setSpeed(Speed);

Serial.begin(9600);

}

void loop()

{

if(Serial.available())

{

data=Serial.readString();

Serial.println(data);

if(data == " ")

{

Serial.println("give the coomand");

}

//BACKWARD

else if(data=="back")

{

//ANTICLOCK

backward();

delay(3);

Serial.println("moving back");

digitalWrite(4,1);

redlightsblink();

}

//FARWARD

else if(data=="front")

{

//CLOCK

analogWrite(8,0);

analogWrite(A3,0);

analogWrite(10,0);

analogWrite(11,0);

analogWrite(12,0);

analogWrite(13,0);

forward();

delay(3);

Serial.println("moving front");

digitalWrite(4,0);

}

//STOP

else if(data=="stop")

{

//STOP

analogWrite(8,255);

analogWrite(9,0);

analogWrite(10,0);

analogWrite(11,255);

analogWrite(12,0);

analogWrite(13,0);

Stop();

delay(3);

Serial.println("not moving ");

digitalWrite(4,0);

}

//speed

else if(data=="speed")

{

delay(3000);

if(Serial.available())

{

int speed=Serial.parseInt();

Serial.println(speed);

analogWrite(5,speed);

analogWrite(9,speed);

}

else{

analogWrite(5,255);

analogWrite(9,255);

}

}

//RIGHTWAERD

else if(data=="right")

{

//CLOCK

right();

delay(3);

Serial.println("moving right");

rightyellowlightblink();

digitalWrite(4,0);

}

//LEFTWARD

else if(data=="left")

{

//antiCLOCK

left();

delay(3);

Serial.println("moving left");

leftyellowlightblink();

digitalWrite(4,0);

}

//BACKRIGHT

else if(data=="backright")

{

//ANTICLOCK

analogWrite(5,255);

analogWrite(9,255);

digitalWrite(6,1);

digitalWrite(7,0);

digitalWrite(3,0);

digitalWrite(2,0);

delay(3);

Serial.println("moving backright");

rightredlightblink();

digitalWrite(4,1);

}

//BACKLEFT

else if(data=="backleft")

{

//ANTICLOCK

analogWrite(5,255);

analogWrite(9,255);

digitalWrite(6,0);

digitalWrite(7,0);

digitalWrite(3,1);

digitalWrite(2,0);

delay(3);

Serial.println("moving backleft");

leftredlightblink();

digitalWrite(4,1);

}

//HORN

else if (data=="horn")

{

tone(4, 440,2000); // play tone 57 (A4 = 440 Hz)

}

else if (data=="hornoff")

{

tone(4, 0,2000); // play tone 57 (A4 = 440 Hz)

}

//HEADLIGHTON

else if (data=="headlight")

{

analogWrite(A2,230);

analogWrite(A0,33);

analogWrite(A1,219);

}

//HEADLIGHTOFF

else if (data=="headlightoff")

{

analogWrite(A2,0);

analogWrite(A0,0);

analogWrite(A1,0);

}

else if (data=="linefollower"){

/\*int M1\_Speed = 80; // speed of motor 1

int M2\_Speed = 80; // speed of motor 2

int LeftRotationSpeed = 250; // Left Rotation Speed

int RightRotationSpeed = 250; // Right Rotation Speed\*/

int LEFT\_SENSOR = digitalRead(A4);

int RIGHT\_SENSOR = digitalRead(A5);

if(RIGHT\_SENSOR==0 && LEFT\_SENSOR==0) {

forward(); //FORWARD

}

else if(RIGHT\_SENSOR==0 && LEFT\_SENSOR==1) {

right(); //Move Right

}

else if(RIGHT\_SENSOR==1 && LEFT\_SENSOR==0) {

left(); //Move Left

}

else if(RIGHT\_SENSOR==1 && LEFT\_SENSOR==1) {

Stop(); //STOP

}

}

/\*else if (data=="obstracel"){

distance = ultrasonic();

if (distance <= 12) {

Stopb();

backwardmove();

delay(100);

Stopb();

L = leftsee();

servo.write(spoint);

delay(800);

R = rightsee();

servo.write(spoint);

if (L < R) {

rightmove();

delay(500);

Stopb();

delay(200);

} else if (L > R) {

leftmove();

delay(500);

Stopb();

delay(200);

}

} else {

forwardmove();

}

}\*/

else

{

digitalWrite(6,1);

digitalWrite(7,1);

digitalWrite(3,1);

digitalWrite(2,1);

delay(3);

Serial.println("give the correct command");

}

}

}

void forward()

{

analogWrite(5, 255);

analogWrite(9, 255);

digitalWrite(6,0);

digitalWrite(7,1);

digitalWrite(3,0);

digitalWrite(2,1);

}

void backward()

{

analogWrite(5,255);

analogWrite(9,255);

digitalWrite(6,1);

digitalWrite(7,0);

digitalWrite(3,1);

digitalWrite(2,0);

}

void right()

{

analogWrite(5,255);

analogWrite(9,255);

digitalWrite(6,0);

digitalWrite(7,1);

digitalWrite(3,0);

digitalWrite(2,0);

}

void left()

{

analogWrite(5,255);

analogWrite(9,255);

digitalWrite(6,0);

digitalWrite(7,0);

digitalWrite(3,1);

digitalWrite(2,0);

}

void Stop()

{

digitalWrite(6,0);

digitalWrite(7,0);

digitalWrite(3,0);

digitalWrite(2,0);

}

void redlightsblink()

{

for(int i=0;i<=255;i=i+1)

{

analogWrite(8,i);

analogWrite(A3,0);

analogWrite(10,0);

analogWrite(11,i);

analogWrite(12,0);

analogWrite(13,0);

delay(1);

}

}

void rightyellowlightblink()

{

for(int i=0;i<=255;i=i+1)

{

analogWrite(8,230);

analogWrite(A3,33);

analogWrite(10,219);

analogWrite(11,0);

analogWrite(12,0);

analogWrite(13,0);

delay(10);

}

}

void leftyellowlightblink()

{

for(int i=0;i<=255;i=i+1)

{

analogWrite(8,0);

analogWrite(A3,0);

analogWrite(10,0);

analogWrite(11,230);

analogWrite(12,33);

analogWrite(13,219);

delay(1);

}

}

void rightredlightblink()

{

for(int i=0;i<=255;i=i+1)

{

analogWrite(8,i);

analogWrite(A3,0);

analogWrite(10,0);

analogWrite(11,0);

analogWrite(12,0);

analogWrite(13,0);

delay(1);

}

}

void leftredlightblink()

{

for(int i=0;i<=255;i=i+1)

{

analogWrite(8,0);

analogWrite(A3,0);

analogWrite(10,0);

analogWrite(11,i);

analogWrite(12,0);

analogWrite(13,0);

delay(1);

}

}

int ultrasonic() {

digitalWrite(Trig, LOW);

delayMicroseconds(4);

digitalWrite(Trig, HIGH);

delayMicroseconds(10);

digitalWrite(Trig, LOW);

long t = pulseIn(Echo, HIGH);

long cm = t / 29 / 2; //time convert distance

return cm;

}/\*

void forwardmove() {

M1.run(FORWARD);

M2.run(FORWARD);

M3.run(FORWARD);

M4.run(FORWARD);

}

void backwardmove() {

M1.run(BACKWARD);

M2.run(BACKWARD);

M3.run(BACKWARD);

M4.run(BACKWARD);

}

void rightmove() {

M1.run(BACKWARD);

M2.run(BACKWARD);

M3.run(FORWARD);

M4.run(FORWARD);

}

void leftmove() {

M1.run(FORWARD);

M2.run(FORWARD);

M3.run(BACKWARD);

M4.run(BACKWARD);

}

void Stopb() {

M1.run(RELEASE);

M2.run(RELEASE);

M3.run(RELEASE);

M4.run(RELEASE);

}

int rightsee() {

servo.write(20);

delay(800);

Left = ultrasonic();

return Left;

}

int leftsee() {

servo.write(180);

delay(800);

Right = ultrasonic();

return Right;

}\*/

**CHAPTER 9**

**BIBLIOGRAPHIES**

**Web Resources:**

Arduino.cc. "Arduino Reference." Accessed May 30, 2024. https://www.arduino.cc/reference/en/

MIT App Inventor. "App Inventor for Android." Accessed May 30, 2024. http://appinventor.mit.edu/

IFTTT. "Connect Your Apps and Devices with IFTTT." Accessed May 30, 2024. <https://ifttt.com/>

**Datasheets and Manufacturer Specifications:**

HC-05 Bluetooth Module Datasheet. Accessed May 30, 2024. https://components101.com/wireless/hc-05-bluetooth-module

MQ-2 Gas Sensor Module Datasheet. Accessed May 30, 2024. https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-2.pdf

PIR Motion Sensor Datasheet. Accessed May 30, 2024. https://www.adafruit.com/product/189