University of Hertfordshire

MSc Project Report

Development of an Autonomous Robot for Hazardous Environment Inspection

Name: Mohitkumar Kori

Supervisor: Mr. Emilio Mistretta

August 2023

SCHOOL OF ENGINEERING AND COMPUTER SCIENCE
UNIVERSITY OF HERTFORDSHIRE

www.herts.ac.uk

DECLARATION STATEMENT

I have read the detailed guidelines to Students on academic offenses and misconduct information

I have read the information about the Academic integrity, misconduct and plagiarism posted on the Canvas Project module page, attended the seminar and/or watched the recording delivered by the School Academic Integrity Officer (SAIO).

I understand the University process of dealing with suspected cases of academic misconduct and the possible penalties.

Therefore, I certify that the work submitted is my own and that any material derived or quoted from the published or unpublished work of other persons has been duly acknowledged.
(Ref. University Policy Regulation AS14, Appendix III – version 16.0)

Signature: Mohitkumar Kori

Student name: Mohitkumar Kori
SRN: 21070997

UNIVERSITY OF HERTFORDSHIRE
SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

This project designs an autonomous inspection robot tailored for hazardous environments, such as nuclear facilities and disaster zones. The robot is integrated with advanced sensors and cameras, ensuring safe and efficient navigation during inspections. Autonomous algorithms drive the robot's independent operation, eliminating the need for constant human intervention. Evaluations of the robot's efficacy are conducted in both simulated settings and real-world conditions. The primary deliverable is a prototype robot that can autonomously detect and relay information on hazardous objects and environmental anomalies within its sensor's reach. Additionally, an accompanying mobile application provides an option for remote control, offering flexibility in operation. This endeavour not only pioneer's advancements in hazardous inspection technologies but also offers valuable insights for future innovations in robotic design and functionality.

ACKNOWLEDGEMENTS

I would want to use this chance to offer my sincere gratitude to the people and businesses that have helped and contributed to the successful completion of this one-on-one initiative.

I would first like to express my profound gratitude to my project supervisor Emilio Mistretta for their excellent advice, knowledge, and ongoing assistance throughout the project. Its quality and direction have both been greatly influenced by their mentoring.

I also want to express my gratitude to **University of Hertfordshire** for supplying the tools that were required, such as access to the Arduino IDE software and the Arduino Uno R3 microcontroller, which made a big contribution to the creation of the autonomous robot.

I also want to express my gratitude for the help and encouragement I received from Haythem Elnashar, Martin Thomas, friends, and family members who helped me with my effort by offering encouragement, compassion, and insightful criticism.

I am appreciative of their contributions, which have been important in completing this project successfully. Their support as a group has been invaluable.

List of Figure:

List of Table:

TABLE OF CONTENTS

Introduction:

Autonomous robots have made a significant contribution to the quickly developing field of technology and have captivated the interest of both scientists and engineers as well as the general people. These advanced devices are an amazing example of how artificial intelligence, robotics, and cutting-edge sensor systems can all work together to enable autonomous operation and decision-making. Although the idea of autonomous robots has its origins in science fiction, recent developments in computing power, machine learning, and robotics have made these far-fetched ideas a reality.

A robot that is autonomous can be described as a self-governing organism with the ability to navigate its environment, sense its surroundings, and make defensible judgements based on its internal algorithms and data collected. These robots demonstrate a novel degree of adaptability, efficiency, and precision as a result of not requiring constant human management, allowing them to handle tough tasks in a broad range of industries.

Autonomous robot development has the potential to revolutionise several industries, including manufacturing, healthcare, transportation, agriculture, and exploration. By carrying out repetitive activities with unparalleled consistency and accuracy, they can improve efficiency in the manufacturing industry and streamline production processes. Autonomous robots in healthcare can support medical staff during operations, patient care, and drug distribution, reducing risks and improving patient outcomes. In terms of transportation, they hold out the prospect of safer and more effective travel on land and in the air, lowering accidents and improving traffic flow.

The aim of this research is to create an autonomous robot that can detect hazardous environments. Autonomous robots can effectively complete tasks that would otherwise be risky or difficult for people to complete in harmful circumstances, protecting humans in the process. These settings might contain regions with high pressure, radiation, poisonous substances, or extreme temperatures that pose a threat to human health.

Autonomous robots can move around and investigate dangerous places without the need for direct human intervention by using advanced sensor systems and transmit and receiving communication system. These robots can continuously monitor the environment, find leaks, evaluate the structural integrity of buildings, and gather useful data. They can react swiftly to changing conditions by sensing their environment and making educated decisions based on internal algorithms. This helps them avoid potential threats and reduce hazards.

Autonomous robots can be used in numerous industries while developing in hazardous environments, such as:

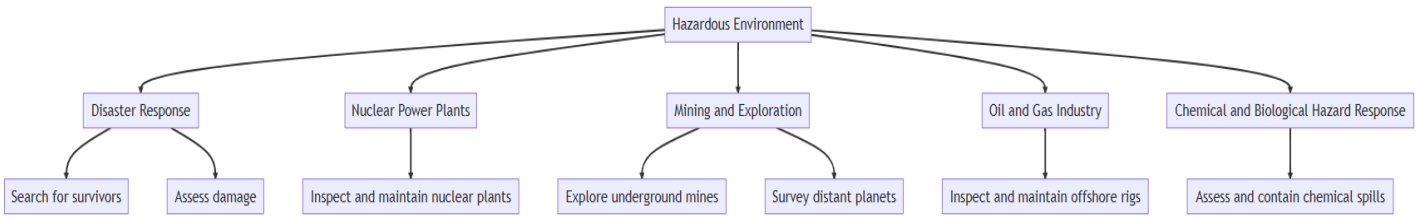

Figure 1.1

- Disaster Response: Autonomous robots can be used to search for survivors, assess the damage to infrastructure, and offer vital information to emergency personnel during natural catastrophes like earthquakes, hurricanes, or wildfires. They can explore locations that might be inaccessible to people by navigating through shaky constructions and other debris.

- Nuclear Power Plants: Nuclear power plants may be inspected and maintained using autonomous robots. Without endangering human operators, they can enter radioactive regions to look for leaks, examine reactor parts, and spot potential safety hazards.

- Mining and Exploration: In mining operations, autonomous robots can scout out underground mines and locate mineral reserves, exposing miners to less dangerous situations. They can be employed in space exploration to survey distant planets and moons and gather important scientific data in hostile conditions to people.

- Oil and Gas Industry: Autonomous robots can be used to perform inspections and maintenance duties in hazardous conditions, such as high-pressure situations and corrosive compounds, in offshore oil rigs and refineries.[1]

- Chemical and biological hazard response: Autonomous robots can be used to analyse the situation, control the spread, and help with cleanup efforts without putting human responders in danger when dealing with dangerous chemical spills or potential biological threats.

Figure 1.2

These robots' fundamental autonomy comes from their capacity to detect their environment using a variety of sensors, such as cameras, lidars, radars, and microphones, which continuously gather massive amounts of data. After that, sophisticated algorithms that read the surroundings, spot trends, and create the right responses to accomplish objectives digest this data.

Aims and Objectives

The aim of this project is to design and develop an autonomous robot that can be used for hazardous environment inspection, such as in nuclear power plants or disaster zones. The robot will be equipped with various sensors and cameras that can capture data on the environment, and use this data to navigate and perform inspections in hazardous areas. The robot will be designed to be resilient to environmental hazards, and be capable of carrying out inspections in a safe and efficient manner. The project will involve designing and building a prototype robot, developing algorithms for autonomous navigation and inspection, and testing the robot's performance in simulated and real-world environments.

- Research Objectives:

1. Design and Development:
   a. Design and build a prototype autonomous robot capable of safely operating in hazardous environments.
   b. Integrate a range of sensors and cameras on the robot to capture environmental data for inspection purposes.

2. Autonomous Navigation and Inspection:
   a. Develop algorithms for autonomous navigation, enabling the robot to navigate through hazardous areas and avoid obstacles.
   b. Implement algorithms for inspection tasks, allowing the robot to identify and assess potential hazards or damage in the environment.

3. Performance Testing:
   a. Conduct testing and evaluation of the robot's performance in simulated hazardous environments to assess its efficiency, accuracy, and safety.
   b. Validate the robot's capabilities by conducting real-world tests in representative hazardous environments, ensuring it meets the required standards for inspection tasks.

4. Practical Application:
   a. Assess the feasibility of deploying the autonomous robot in nuclear power plants and disaster zones for inspection purposes.
   b. Evaluate the robot's potential impact on improving inspection efficiency, reducing human exposure to hazards, and increasing overall safety in hazardous environments.
   c. Provide recommendations for further enhancements and adaptations of the robot's design and functionality to address specific challenges in different hazardous environments.

By accomplishing these research objectives, the project aims to contribute to the development of an autonomous robot that can effectively perform inspections in hazardous environments, improving safety and efficiency in critical industries and disaster response scenarios.

Literature Review:

In recent years, the push towards autonomous inspection in hazardous environments has gained significant traction. Such environments often pose serious threats to human inspectors due to their inherent risks, making the development of autonomous solutions a priority. The research focuses on collecting visual data in areas that are either dangerous or difficult for humans to visit. The advantages of UAVs include significant reductions in human danger as well as time and financial savings. Consider the substantial risk to anyone involved in assessing infrastructure such as electrical lines, boilers, bridges, and tunnels. UAV use can successfully address these issues, improving inspection efficiency and safety. Dynamic UAV flights in GPS-denied conditions are possible by combining inertial measurements and visual data from small onboard cameras.[1]

Hyperspectral imaging, which captures information across hundreds of narrow spectral bands, offers a unique diagnostic ability to identify materials and objects based on their spectral signatures. Such rich spectral information can be invaluable in hazardous environments, where traditional imaging might miss potential hazards. Anomaly detection, a branch of hyperspectral target location, focuses on identifying targets or materials without prior knowledge. This is particularly significant for autonomous robots in hazardous environments, where unforeseen hazards might exist, and prior knowledge of every potential danger is not feasible. Various methods have been proposed for hyperspectral anomaly detection, including statistic-based, distance-based, reconstruction-based, subspace-based, deep learning-based, among others. Each method offers different advantages, and their application can depend on the specific requirements of the inspection task. Of relevance to autonomous robots is real-time anomaly detection. Given that these robots operate autonomously, the ability to detect and react to anomalies in real-time is crucial for safety and effective inspection. The research in hyperspectral real-time anomaly detection provides insights that can be adapted or integrated into robotic systems designed for hazardous environment inspection [2].

Robotics and autonomous systems (RAS) have been quickly integrating and influencing a variety of fields and businesses. Robots, formerly restricted to production environments, are now found everywhere, including homes, restaurants, military facilities, therapeutic settings, and transportation hubs (Wong et al., n.d.). RAS have several advantages, which are highlighted by this change and their broad acceptance. They are excellent at doing complicated, repetitive tasks ceaselessly and not only have capacities beyond those of humans. Their use in dangerous occupations demonstrates their capacity to guarantee human safety. Their efficacy, high-speed functioning, and cost-effectiveness in particular circumstances further highlight their significance. However, the use of RAS in difficult or severe situations brings up several complications. Traditional RAS approaches, while they may be successful in controlled or moderate environments, frequently fail in harsh ones. The trajectory of RAS technologies points to a future in which they will be increasingly more thoroughly integrated in difficult environments, providing heightened advantages across a range of businesses and sectors. The research emphasises the significance of ongoing innovation and adaptation as this integration progresses in order to maximise RAS's potential in a variety of situations.[3]

In the pursuit of developing autonomous robots for hazardous environment inspections, real-time Simultaneous Localization and Mapping (SLAM) systems play a pivotal role. These systems provide robots with the capability to understand and navigate unfamiliar terrains, even when they are dynamic or cluttered. One such cutting-edge SLAM system is the ORB-SLAM, introduced by Raúl Mur-Artal, J.M.M. Montiel, and Juan D. Tardós.

ORB-SLAM: This is a feature-based monocular SLAM system designed to function in real-time across various environments, both indoor and expansive outdoor settings. Key characteristics of ORB-SLAM include Robustness, Versatility, Consistent Features. Systems like ORB-SLAM are a crucial resource for autonomous robots navigating hazardous areas. Because of its real-time mapping and localization capabilities, the robot may instantly modify its course or its course of action in response to the most recent environmental data. Furthermore, the system's resilience guarantees that mistakes are kept to a minimum even in chaotic or dynamic terrains, which are typical of dangerous conditions [4].

The sensing capabilities of autonomous robots play a crucial role in hazard detection and environment inspection. These robots are typically equipped with a range of sensors, such as LiDAR, cameras, gas detectors, and radiation sensors. LiDAR sensors enable the robot to generate 3D maps of its surroundings, allowing for obstacle detection and avoidance [5]. Integration of vision systems provides the ability to detect and recognize hazardous objects or leaks in pipes [6]. Advanced sensing techniques, such as hyperspectral imaging, have also been explored to enhance the robot's ability to identify hazardous materials [7].

The performance of autonomous robots for hazardous environment inspection needs to be rigorously evaluated to ensure their effectiveness, efficiency, and safety. Researchers have conducted tests in simulated environments that replicate hazardous conditions to assess the robot's capabilities [8]. Real-world experiments in nuclear power plants and disaster zones have also been conducted to evaluate the robot's performance in practical scenarios [9]. Metrics such as inspection time, detection accuracy, and operator satisfaction are often used to assess the robot's performance.

Methodology:

• Hardware Design

○ Overview

The inspection capabilities of the robot are facilitated by an array of sensors and cameras, carefully chosen for their ability to capture key environmental data. These sensors include IR sensors for distance estimation, an ultrasonic sensor for object detection and distance measurement, a thermal detector sensor for temperature monitoring, and a gas detector sensor for identifying harmful gases. These sensors allow the robot to perceive and understand the environment, enabling autonomous navigation and efficient inspection.

The robot is designed to operate without human intervention once deployed in the field. The autonomous nature of the robot is achieved through the development of sophisticated algorithms, which govern navigation and inspection tasks. The algorithms utilize the data captured by the onboard sensors to direct the robot's movements and actions, effectively creating a reactive system that can adapt to the dynamic conditions within hazardous environments.

The project also includes a rigorous testing phase, wherein the robot's performance is evaluated in both simulated and real-world environments. These tests aim to assess the efficiency, accuracy, and safety of the robot while operating in hazardous conditions. The outcomes of these tests will provide valuable insights into the robot's operational capabilities, identifying areas of strength and potential areas for improvement.

This project is expected to make significant contributions to the field of robotics, particularly in the development of autonomous systems for hazardous environment inspections. Through the design, development, and testing of this robot, we aim to advance our understanding of autonomous robot operation and provide a robust, practical solution for hazardous environment inspection tasks.

- **Design and Structure:**

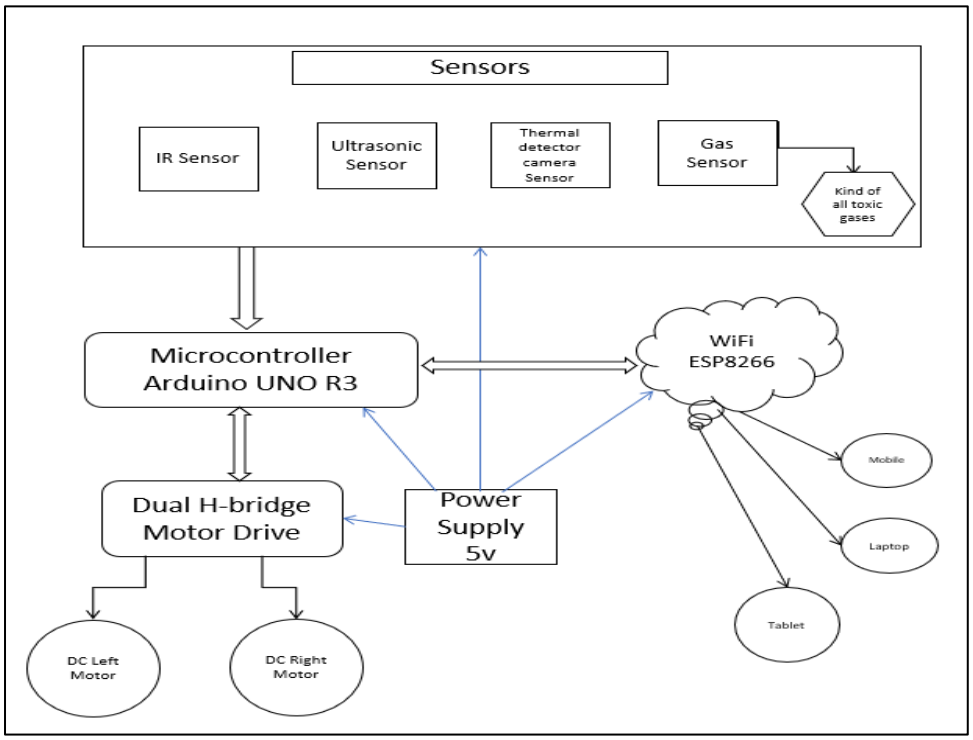

Figure 3.1 Architecture

a) Purpose and Functionality:

- The robot must be autonomous and capable of navigation and inspecting hazardous environments such as nuclear plants or disaster zones without human intervention.
- The robot should be able to capture data on the environment using its equipped sensor and use this data to navigation and inspection.

b) Hardware and components:

- Microcontroller: The robot should be built around the Arduino Uno R3 microcontroller, which will serve as the brain of the robot.
- Motor Drive and Motor: A dual H bridge motor drive is a fundamental component in controlling the direction and speed of DC motors. At its core, the H-bridge is a set of our four switches that can be opened or closed in various combination to control the flow of current through the motor, thus determined its direction and speed.
- Servo: A micro servo SG90 is used for precise angular movement for Ultrasonic sensor positioning.
- Sensor:
  - IR Sensors (MH-Sensor series): At least 2 IR sensors can be integrated for obstacle detection and avoidance.
  - Ultrasonic Sensor (HC-SR04): For distance measurement and to aid in navigation.
  - IR temp. radiation detector camera sensor (MLX90614-DCC): To detect temperature variation in the environment.
  - Gas Detector Sensor (MQ-135): To detect the harmful gases like NH3, NOx, CO2, benzene, smoke, and other dangerous gases from the atmosphere.

- Transceiver (ESP-12F (Fire Beetle ESP8266 board)): This WiFi module is using for data transmission.

c) Schematic circuit diagram:

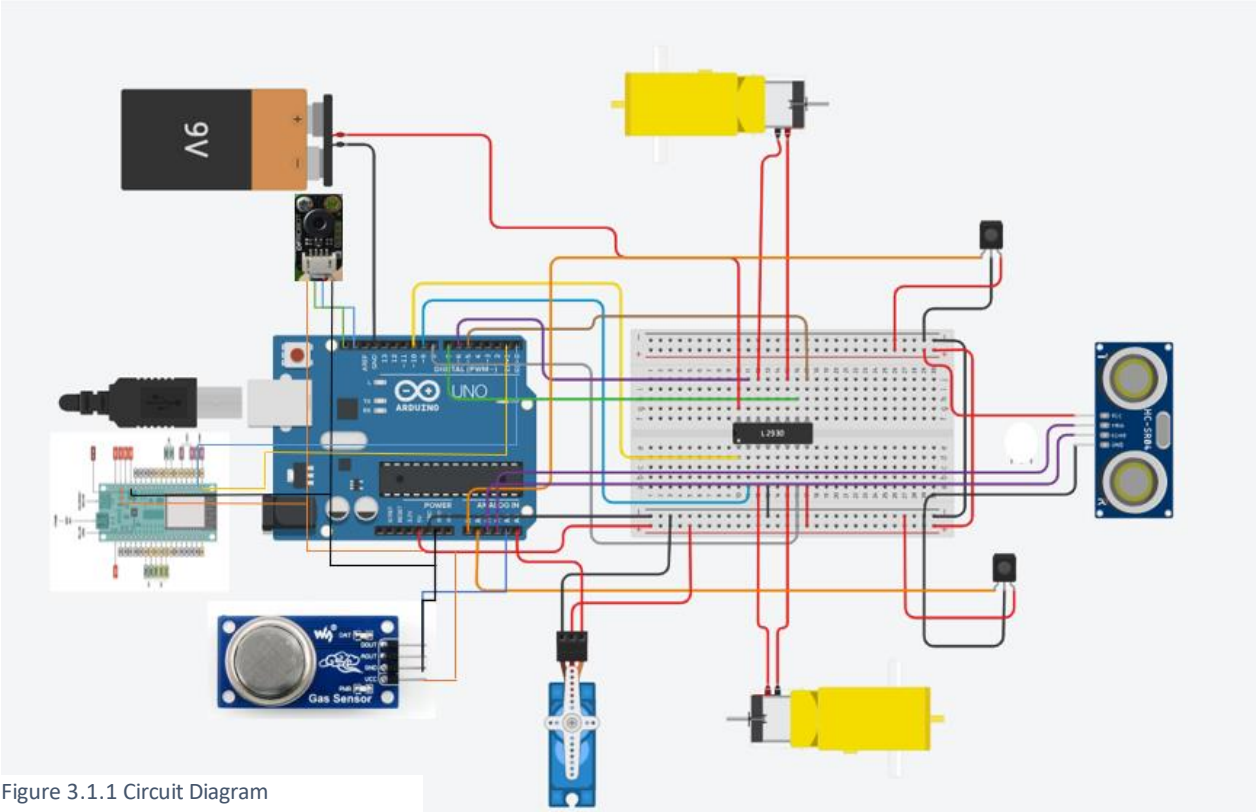

Figure 3.1.1 Circuit Diagram

d) Body Structure:

i) Top View:

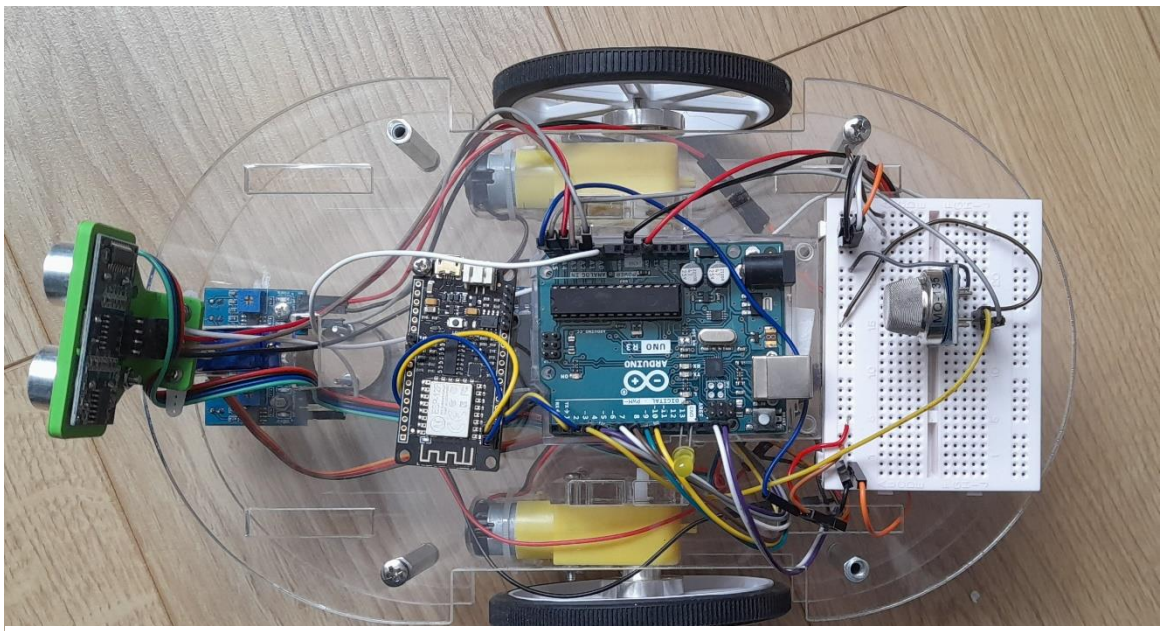

Figure 2.1.2 Top view

ii) Front View:

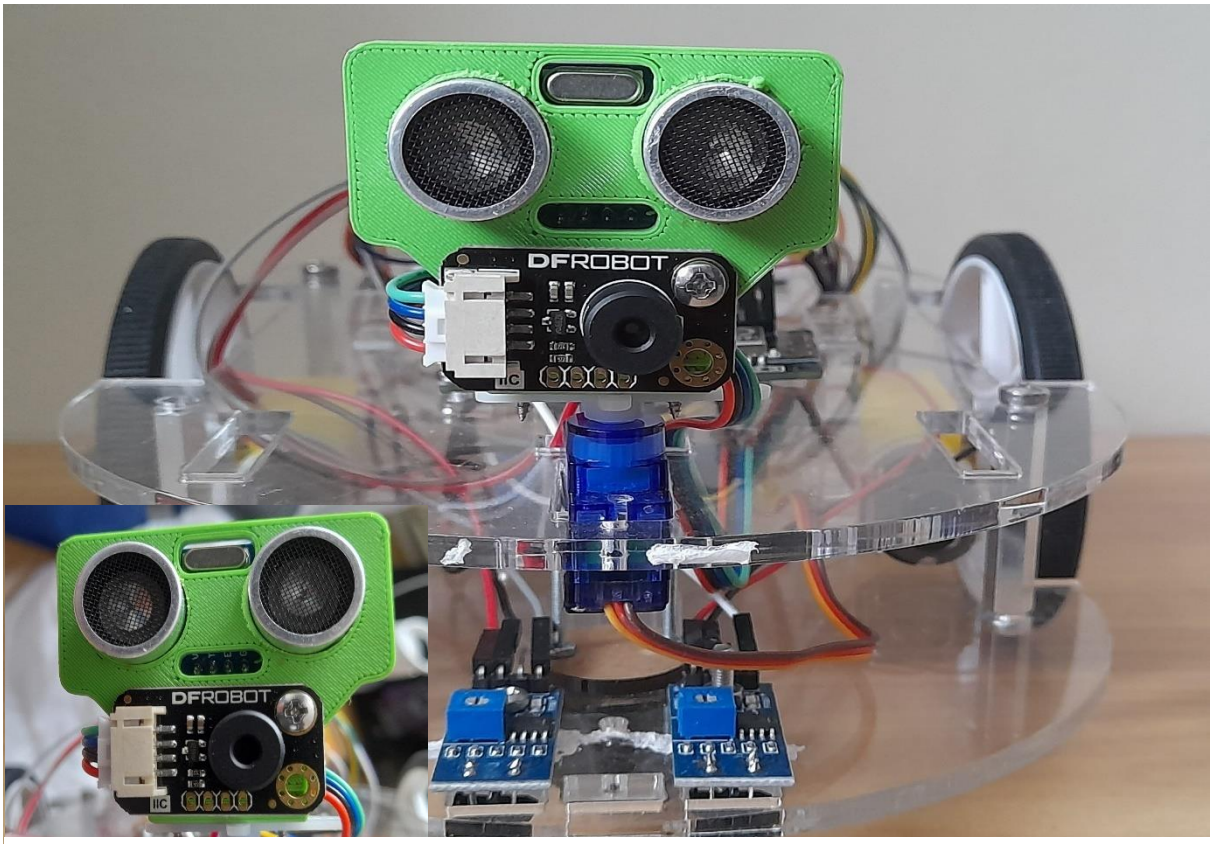

Figure 3.1.3

iii) Bottom view:

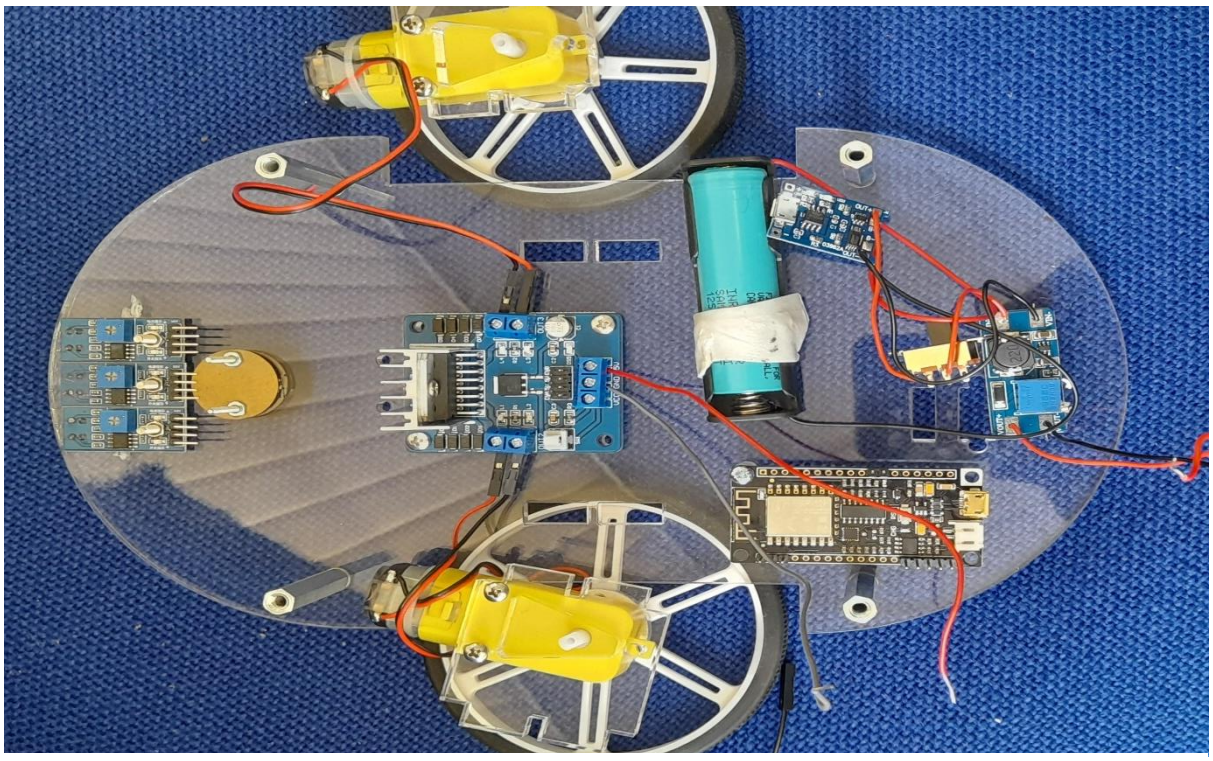

Figure 3.1.4

iv) Side view:

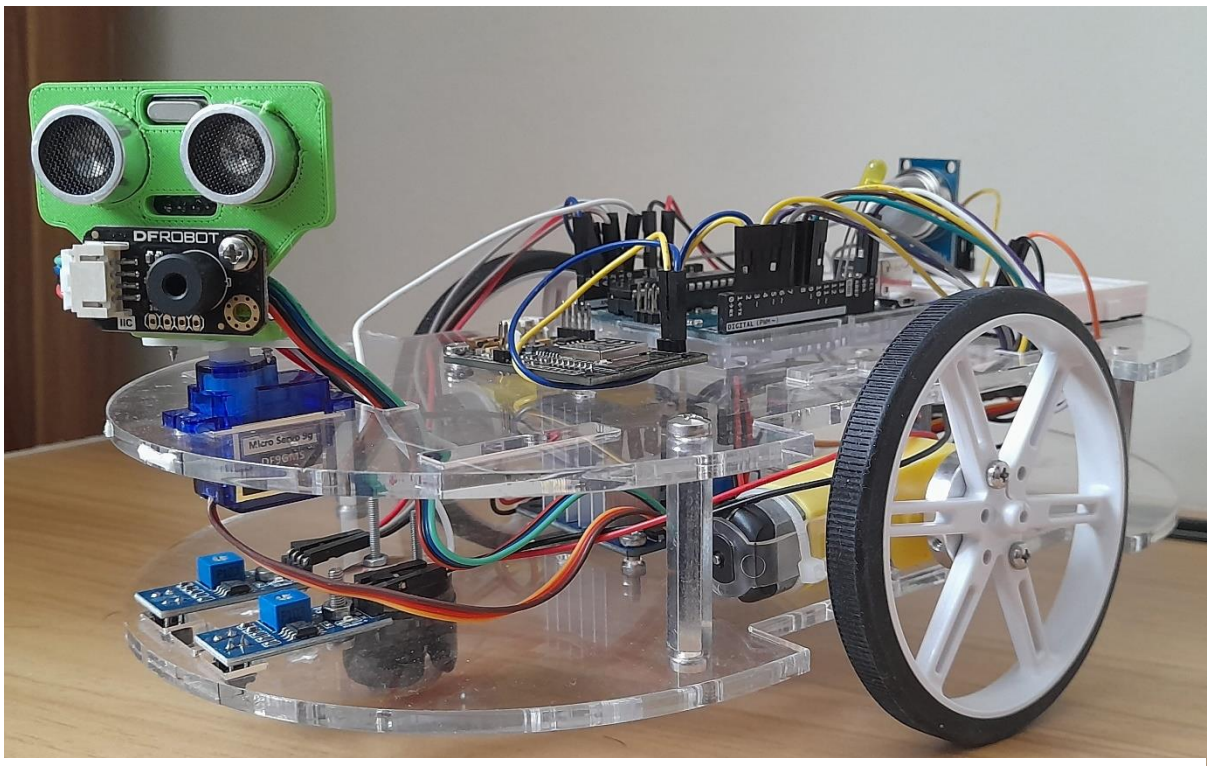

Figure 3.1.5

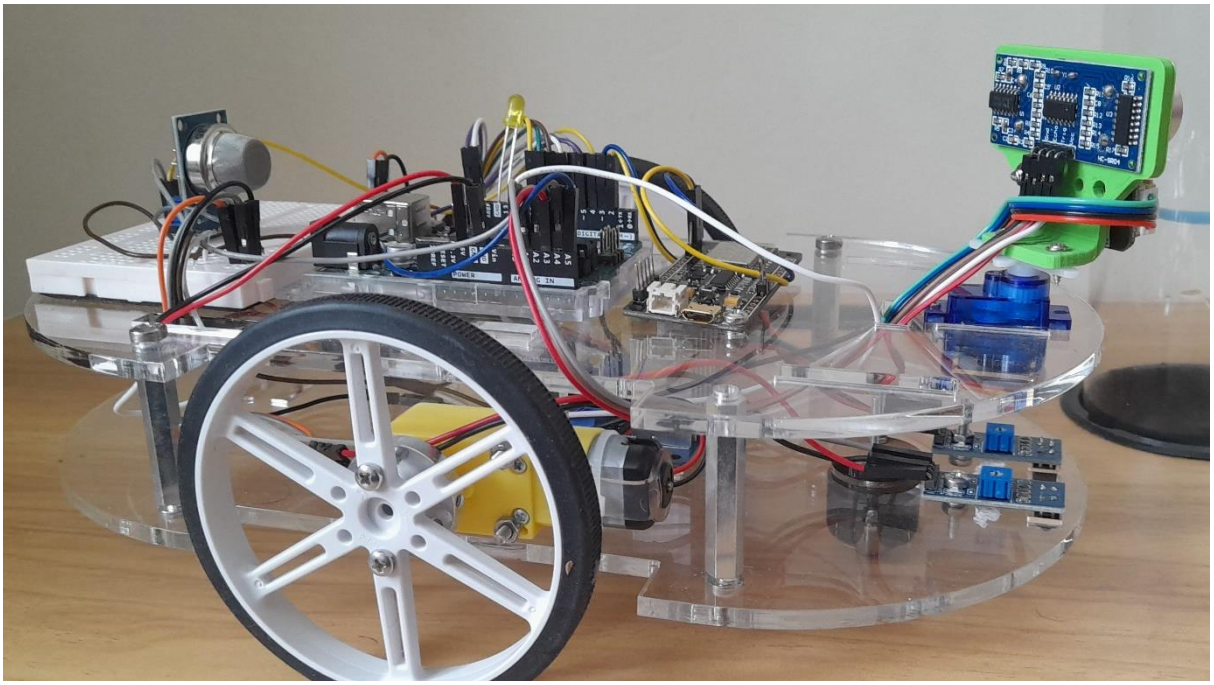

Figure 3.1.6

v) Power Supply: 5 vdc Battery

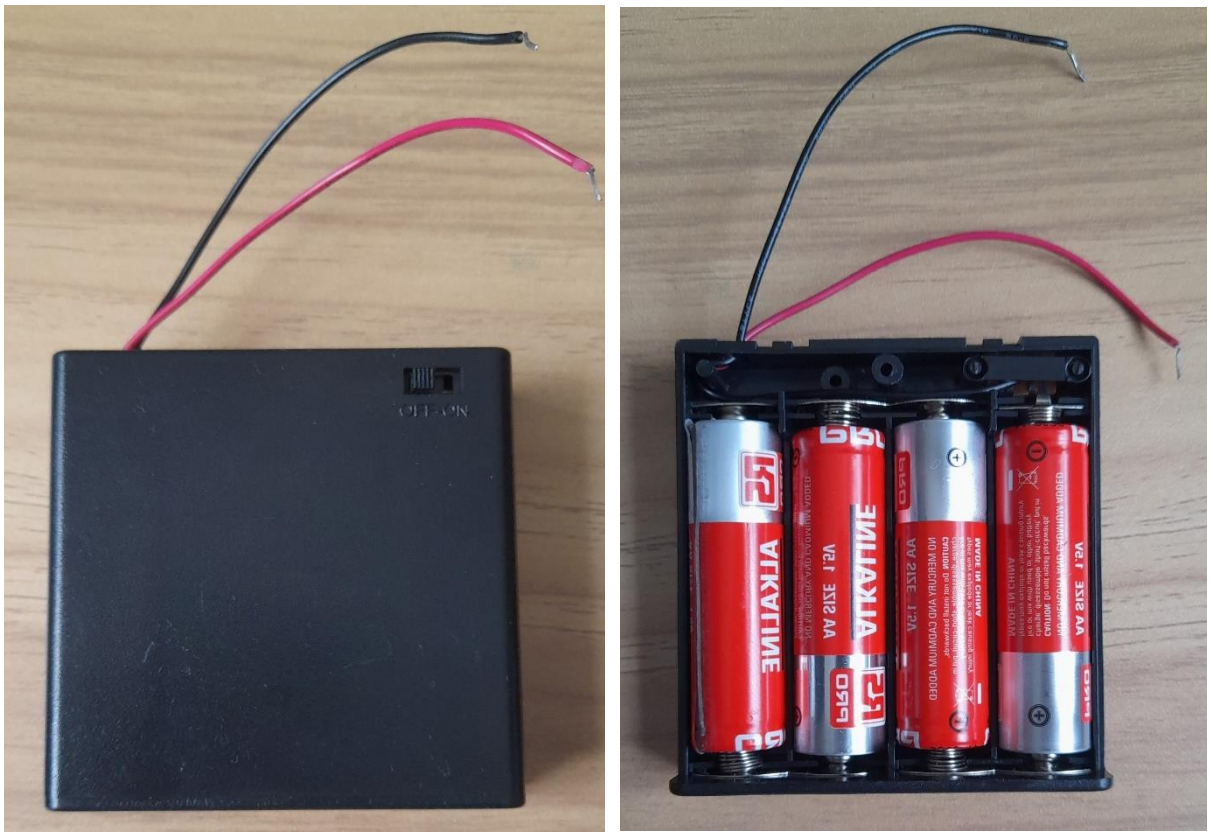

Figure 3.1.7

- **Sensors Integration:**

a) IR sensor: The infrared sensor, often known as an IR sensor, is a type of electronic part that uses IR radiation to emit or detect properties from its environment. Additionally, the heat and motion of a target may be detected or measured using these sensors. The IR sensor circuit is a critical component in many electrical devices.

Figure 3.2.1

As seen in figure1, A photodiode works as the detector in this sensor while an IR LED works as the emitter. When an IR light beam strikes the photodiode, the output voltage and resistance will vary according to the strength of the infrared light that was received. This uses a photodiode that is extremely sensitive to the infrared light produced by an infrared LED. According on the amount of acquired infrared light, the photodiode's resistance and output voltage may be adjusted. This is how an IR sensor works in general.
In this project, The IR sensor used as for line or edge detection. They help the robot stay on a path or within certain boundaries. They do not measure distance but detect specific conditions like the presence of a line or edge.

b) Ultrasonic sensor: An ultrasonic sensor is a component of electronics that uses ultrasonic waves to determine the distance to an object and then turns the sound that is reflected into an electrical signal. Compared to audible noises, ultrasound travels more quickly. A transmitter (which emits sound using a piezoelectric crystal) and a receiver make up an ultrasonic sensor.

Figure 3.2.2

An ultrasonic sound is transmitted via the module's transmitter. If there is an object in front of the ultrasonic sensor, this sound will be reflected. The receiver built into the same module hears the sound that is reflected. A wave propagates an ultrasonic signal at a 30-degree angle.

The distance is determined by measuring the travel time of ultrasonic sound its speed. Distance = Time × Speed of Sound / 2.
In this project, The Ultrasonic sensor is primarily used for object detection and avoidance. It measures the distance to objects and helps the robot navigate around obstacles.
Functionality:

- The Ultrasonic sensor measures the distance to objects in its path using the **'Ultrasonic_read()'** function.
- In the **'loop()'** function, the distance measured by the Ultrasonic sensor **'(distance_F)'** is used to determine if there's an obstacle in front of the robot. If the distance is less than a set threshold (Set), it indicates the presence of an obstacle.
- The robot then uses the '**Check_side()'** function to measure distances on its left and right sides using the Ultrasonic sensor. Based on these measurements, the robot decides which direction has more space and turns in that direction to avoid the obstacle.

c) MQ-135 Gas Sensor: The MQ-135 sensor is a part of the MQ series used to identify various atmospheric gases. Gases including NH3, NOx, alcohol, Benzene, smoke, CO2, etc. may all be detected with the MQ-135 sensor.

The MQ-135 gas sensor module has both analogue and digital outputs that are retrieved from its AO pin and DO pin, respectively. The analogue output voltage ranges from 0 to 5 volts, and when more gas vapours encountered the sensor, the output voltage rises correspondingly. Under normal circumstances, the sensor's output voltage is directly inversely proportional to the amount of CO2 gas present in PPM. Through Arduino's analogue to digital converter, this output voltage is transformed into a digital value (0–1023). This value is equal to the gas concentration in PPM.

Figure 3.2.3

However, the analogue output from the LM393 comparator, which is located on the rear of the sensor module, is acquired before the digital output voltage (0/1) is obtained. The built-in potentiometer is manually adjusted to adjust the digital output's sensitivity and establish a threshold value. The DOUT LED is used to assist with this. The digital output will be LOW if the gas vapor concentration exceeds the threshold value specified. The DOUT LED will light up, making this simple to monitor. Additionally, the potentiometer's sensitivity increases as it is turned clockwise.

In this project, its used for sensing the gas that which kind of gases floating in the air which respect to PPM for Hazardous environment.

- Reading Sensor Values:

'sensorValue = analogRead(GAS_SENSOR_PIN);' : This line reads the analog value from the MQ-135 gas sensor using the defined GAS_SENSOR_PIN (which is set to A4).

'digitalValue = digitalRead(GAS_SENSOR_DIGITAL_PIN);' : This line reads the digital value from the MQ-135 gas sensor using the defined GAS_SENSOR_DIGITAL_PIN (which is set to 2)

Below mention the table, its gases and PPM range that it is used in the project.

Table 1

Gases	PPM ranges
NH_3 (Ammonia)	10 ppm – 300 ppm
Benzene	300 ppm – 1000 ppm
CO_2 (Carbon Dioxide)	1000 ppm – 2000 ppm

d) IR temp. radiation detector camera sensor (MLX90614-DCC): This is non-contact sensor, which is uses infra-red radiation to measure the temperature and does not require a direct touch. Additionally, this type of measuring allows for rapid and precise readings. The optical system, photoelectric detector, amplifier, signal processing and output module, and the IR temperature probe are all separate components. The thermometer's position and the optical system's field of vision determine the size of the infrared radiation that is collected, and when that radiation converges on the photoelectric detector, it transforms its energy into matching electrical impulses. The signal is transformed into the target's temperature value after being processed by the amplifier and signal processing circuit and calibrated using the algorithm and emissivity of the target.

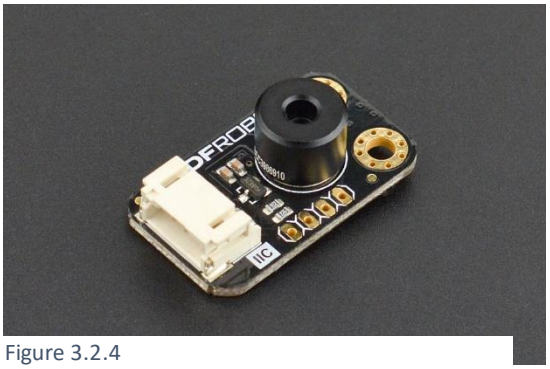

Figure 3.2.4

- **Motor control Algorithm:**

a) Component required:

- 2 DC motors
- Dual H-bridge motor drive

b) DC Motor Configuration:

- Connect the two terminals of the first DC motor to the output terminals (usually labeled 'OUT1' and 'OUT2') of the Dual H-bridge motor drive.
- Similarly, connect the two terminals of the second DC motor to the other output terminals (usually labeled 'OUT3' and 'OUT4') of the Dual H-bridge motor drive.

c) Dual H-bridge motor:

- Connect the input pins of the H-bridge (usually labeled 'IN1', 'IN2', 'IN3', and 'IN4') to the digital pins of the Arduino Uno R3. These pins will be used to control the direction and speed of the DC motors.
- Connect the ground (GND) of the H-bridge to the ground (GND) of the Arduino.
- Connect the VCC or power input of the H-bridge to a suitable power supply, ensuring it matches the voltage requirements of the DC motors.

d) Motor algorithm:

- In below table, it is mentioning control logic for DC motor:

Table 2

IN1	IN2	IN3	IN4	Direction
LOW	HIGH	HIGH	LOW	Forword
HIGH	LOW	LOW	HIGH	Backword
LOW	HIGH	LOW	HIGH	Right turn
HIGH	LOW	HIGH	LOW	Left turn
LOW	LOW	LOW	LOW	Stop

- Forward Movement ('forword' function): This function drives the robot forward. The left motor is set to rotate forward by setting in1 to LOW and in2 to HIGH. Similarly, the right motor is set to rotate forward by setting in3 to HIGH and in4 to LOW.
- Backward Movement ('backword' function): This function drives the robot backward. The left motor is set to rotate backward by setting in1 to HIGH and in2 to LOW. The right motor is also set to rotate backward by setting in3 to LOW and in4 to HIGH.
- Right Turn ('turnRight' function): This function makes the robot turn to the right. The left motor is set to rotate forward (to push the robot to the right) by setting in1 to LOW and in2 to HIGH. The right motor is set to rotate backward (to pull the robot to the right) by setting in3 to LOW and in4 to HIGH.
- Left Turn ('turnLeft' function): This function makes the robot turn to the left. The left motor is set to rotate backward (to pull the robot to the left) by setting in1 to HIGH and in2 to LOW. The right motor is set to rotate forward (to push the robot to the left) by setting in3 to HIGH and in4 to LOW.
- Stop Movement (Stop function): This function stops both motors, halting the robot. Both motors are turned off by setting all input pins (in1, in2, in3, and in4) to LOW.

- Microcontroller (Arduino Uno R3): Here, its used as main controller board just like brain control the body. One of the most well-liked microcontrollers for students and beginners is the Arduino Uno R3, and it provides several benefits that make it appropriate for a variety of applications. It has some reason that why choose this controller.

a) Ease of Use: The Uno R3, which is part of the Arduino platform, is made with novices in mind. It is a great option for individuals new to electronics and programming due to its simple design and abundance of information. Due to its low cost, the Arduino Uno R3 is a popular option for student projects and prototyping.

b) Community Support:

- Huge Global Community: There is a huge global community for Arduino. Accordingly, most of the difficulties or problems you could encounter have previously been resolved by someone in the community.

- Comprehensive Libraries: Since practically every sensor and module has a library, you may avoid writing new code from scratch. Sensor libraries for your project, such as those for the MLX90614 or MQ-135, are easily accessible.

c) Open-source Platform like both the Arduino hardware and software are open-source. This means we can understand its workings entirely, modify it, and even create custom versions if needed. The Arduino Uno R3 has a lot of community support and is simple to use, which makes it a valuable instructional tool. It is a great resource for learning both programming and electronics.

d) When we compare with other microcontroller likewise Raspberry pi are fully-fledged computers, they may be excessive for applications that merely call for basic sensor readings and actuator control. However, Raspberry Pi do have higher processing capability. A further difference between the Raspberry Pi and the Arduino is that the Raspberry Pi runs an operating system, which may cause delay.

- Another controller brand for example STM32, ESP32. These microcontrollers offer feature like built in WiFi (ESP32) or greater processing capabilities (STM32), they might have steeper learning curves, less community support or might be over-spec'd for basic projects.

- **Protocol:**

1) I2C (Inter-Integrated Circuit) Protocol: I2C, pronounced "I-squared-C," is a synchronous, multi-master, multi-slave, packet-switched, single-ended, serial communication bus. This protocol is used in between IR thermal detector to Arduino Uno R3 in this project.

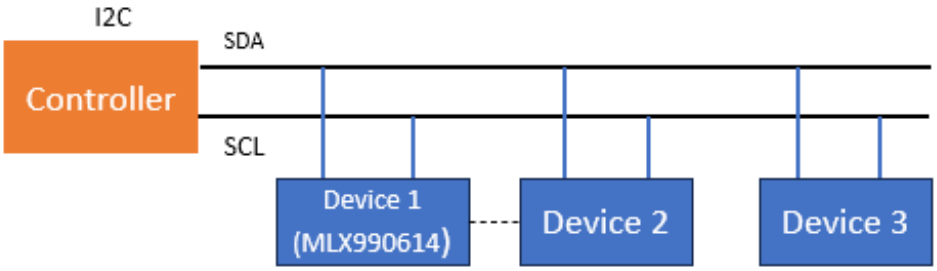

Figure 3.4.1

Serial Data Line (SDA) and Serial Clock Line (SCL), both bidirectional open-drain lines that are pushed up with resistors, are the sole lines used by I2C. Open-drain refers to a line that is passively pulled high (often by resistors) yet actively pushed low by devices. One or more slave devices reply to the master device via I2C communication, which is initiated by a master device. Data can be sent or received by both, but only the master can start a conversation. Every slave device has a unique address. The address of the slave with whom the master wants to contact is sent out whenever the master wishes to speak with that slave.

Data Transfer: Data is transferred in sequences of 8 bits. After each byte of data, an acknowledge bit is sent by the receiver. A START condition (S) signals the beginning of communication, while a STOP condition (P) signals its end. The SDA line is changed from high to low by the master while the SCL line remains high,

indicating a START state. The SDA line is switched from low to high while the SCL line is high to indicate the STOP state.

2) Serial (UART) Communication Protocol: Universal Asynchronous Receiver-Transmitter is referred to as UART. The data format and transmission rates are both programmable in this hardware communication protocol for asynchronous serial communication. This protocol we used in WiFi (ESP8266) communication with Arduino Uno R3.

Figure 3.4.2

Figure 3.4.3

UART typically consists of a pair of communication lines: Tx (Transmit) and Rx (Receive). Data sent from the Tx line of a transmitting UART is received on the Rx line of the receiving UART. Unlike synchronous communication, UART does not use a clock signal. Instead, both devices must agree on the data rate (baud rate) beforehand. They sample the data based on this agreed rate.

- Data Frame: Data is sent in packets or frames.
- A typical frame includes: Start bit (indicates the beginning of data frame), Data bits (usually 7 or 8, but can vary), Optional parity bit (for error checking), Stop bits (indicates the end of data frame)

Baud Rate: It is the speed at which data is transmitted, usually measured in bits per second (bps). Both devices must operate at the same baud rate to communicate correctly. UARTs often have FIFO (First In, First Out) buffers that allow them to store incoming data, which can be read out at a pace convenient to the receiving system.

3) HTTP (Hypertext Transfer Protocol): HTTP is an application-layer protocol designed for the transfer of hypertext documents, such as web pages. It is the foundation of data communication on the World Wide Web. Over the years, HTTP has evolved, with the most recent version being HTTP/2, but the basic principles remain the same.

- Request-Response Model: A request-response system is used by HTTP. In this case, the ESP8266 microcontroller or a web browser served as the client makes a request to the server. A suitable answer is then returned when the server has processed the request.
- HTTP Methods: HTTP methods, also known as "HTTP verbs," define the type of action to be performed on a resource. Here are the commonly used methods GET, POST, PUT and DELETE.
    - GET: Requests data from a specified resource.
    - POST: Submits data to be processed to a specified resource.
    - PUT: Updates a current resource with new data.
    - DELETE: Deletes a specified resource.

In IoT (Internet of Things) applications, such as the one in our code, data is frequently sent from devices to cloud platforms using HTTP. In your situation, the ESP8266 gathers sensor data and uses an HTTP GET request to communicate it to ThingSpeak.
For online data transport, HTTP is a flexible and extensively used protocol. It is a well-liked option for web and IoT applications because to its resilience and simplicity. It's vital to take employing HTTPS or other secure communication protocols into consideration when working with sensitive data or applications that demand increased security.

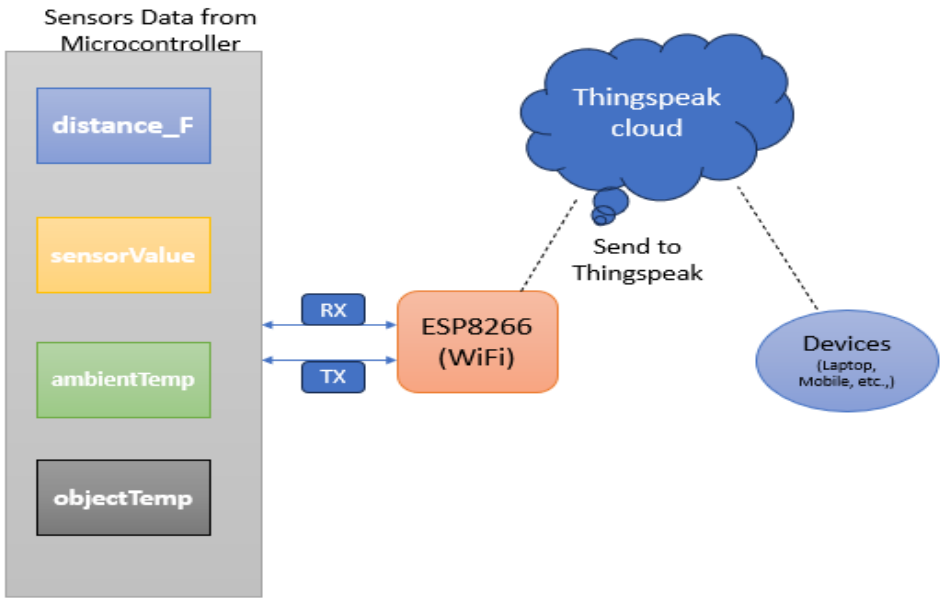

Figure 3.4.4

4) Communication between ESP8266 and ThingSpeak using HTTP: A cloud-based IoT platform called ThingSpeak enables users to gather, visualize, and analyze real-time data streams. A low-cost Wi-Fi microprocessor called the ESP8266 may be configured to broadcast and receive data over the internet. In the following section, we will go into depth on how to use the HTTP protocol to communicate data from the ESP8266 to ThingSpeak.

a) Setting Up the ESP8266:

- Initialization: Set up the Wi-Fi credentials (SSID and password) for the ESP8266 to connect to the internet and initialize the serial connection to monitor the process first.

```
const char* ssid = "VM9517632";
const char* password = "rv6RdxyfVfxk";
const char* host = "api.thingspeak.com";
const char* apiKey = "IBVKT4Q1M8P9W0UW";
```

Figure 3.4.5

- Wi-Fi Connection: The ESP8266 tries to join the designated Wi-Fi network. The gadget will keep attempting until a successful connection is made thanks to a loop.
- IP Address Retrieval: Once connected, the ESP8266 retrieves and prints its local IP address, which can be useful for debugging purposes.

b) Data Collection:

- Serial Input: The ESP8266 waits for data input via the serial port. This data can be from sensors or other devices connected to the ESP8266.
- Data Parsing: The received data string is parsed to extract individual data points. In the provided code, the data string is expected to contain values for ‘distance_F’, ‘sensorValue’, ‘ambientTemp’, and ‘objectTemp’.

c) HTTP Communication:

- Building the HTTP Request:

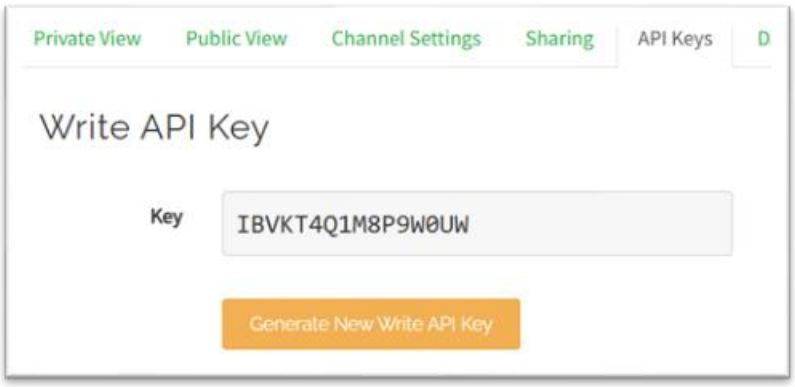

Figure 3.4.6

  - An HTTP GET request is constructed to send the parsed data to ThingSpeak. The request includes the API key (which identifies the ThingSpeak channel) and the data fields.
  - The request also contains essential HTTP headers like "Host" and "Connection".
- Sending the Request: The ESP8266 establishes a connection to the ThingSpeak server and sends the constructed HTTP GET request.
- Receiving the Response: After sending the request, the ESP8266 waits for a response from the ThingSpeak server. This response can provide information about the success or failure of the data upload.

d) Delay and Repeat: The ESP8266 waits for a set amount of time—in this case, five seconds—after delivering data and analyzing a response before looking for fresh data to send. Continuous data transfer to ThingSpeak is ensured via this loop.

- **Flowchart:**

1) Main function:

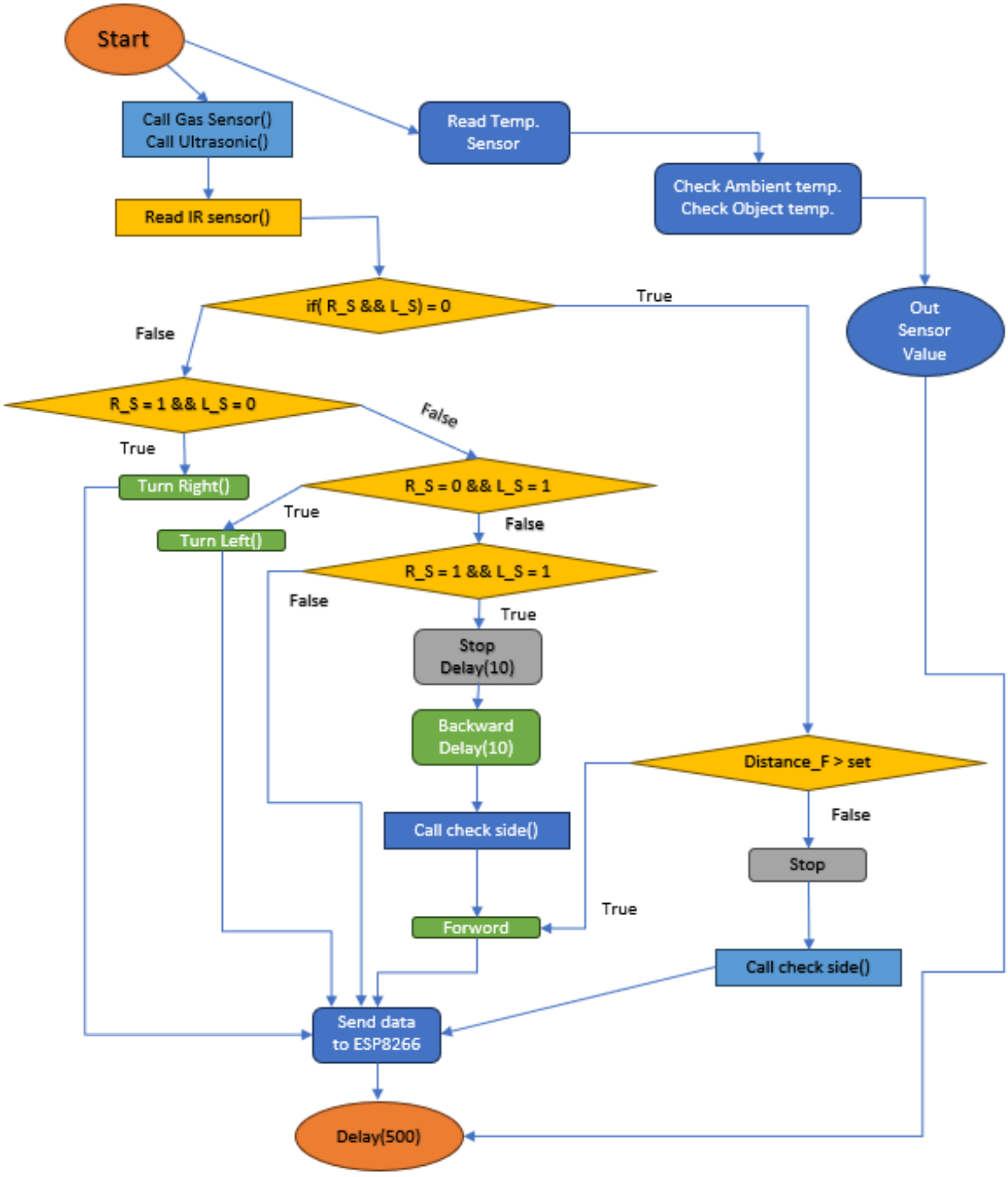

Figure 3.5.1

2) Gas Sensor;

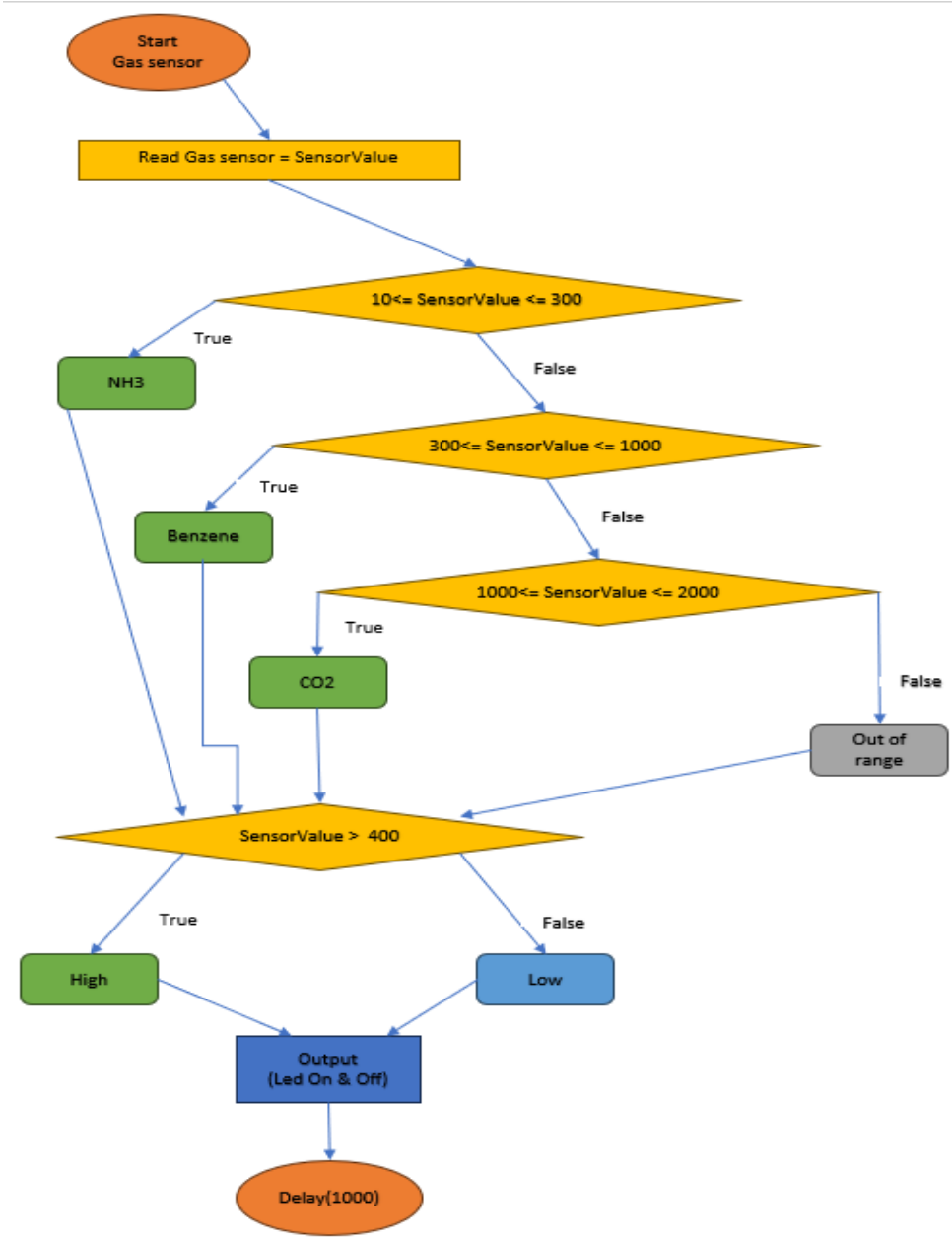

Figure 3.5.2

- Code:
  1) Code for Arduino:

```
#include <DFRobot_MLX90614.h>
#include <SoftwareSerial.h>

SoftwareSerial espSerial(2, 3); // Software serial for communication with ESP-12F

// Libraries and definitions from code_1

// Define pins for motor driver L298
#define enA 10
#define in1 9
#define in2 8
#define in3 7
#define in4 6
#define enB 5

// Define pins for infrared sensors
#define L_S A0
#define R_S A1

// Define pins for ultrasonic sensor
#define echo A2
#define trigger A3

#define servo A5   // Servo motor pin

// Define pins for gas sensor
#define GAS_SENSOR_PIN A4
#define GAS_SENSOR_DIGITAL_PIN 4

// Libraries and definitions from code_2
#include <Adafruit_MLX90614.h>
Adafruit_MLX90614 mlx = Adafruit_MLX90614();  // Initialize MLX90614 sensor

// Variables from code_1
int Set = 15;  // Threshold for distance
int distance_L, distance_F, distance_R; // Variables for storing distances
int sensorValue;  // Analog value from gas sensor
int digitalValue; // Digital value from gas sensor

void setup() {
  // Initialize ESP-12F communication
  espSerial.begin(9600);

  // Setup code from code_1
  Serial.begin(9600);

  // Setup pins for IR sensors, ultrasonic sensor, motor driver, and LEDs
  pinMode(R_S, INPUT);
  pinMode(L_S, INPUT);
  pinMode(echo, INPUT);
  pinMode(trigger, OUTPUT);
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
```

```
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(enB, OUTPUT);
    analogWrite(enA, 200);
    analogWrite(enB, 200);

    pinMode(13, OUTPUT);  // LED pin
    pinMode(GAS_SENSOR_DIGITAL_PIN, INPUT); // Digital pin for gas sensor
    pinMode(servo, OUTPUT); // Servo pin

    // Move servo to initial position
    for (int angle = 70; angle <= 140; angle += 5) {
      servoPulse(servo, angle);
    }
    for (int angle = 140; angle >= 0; angle -= 5) {
      servoPulse(servo, angle);
    }
    for (int angle = 0; angle <= 70; angle += 5) {
      servoPulse(servo, angle);
    }

    distance_F = Ultrasonic_read(); // Read distance from forward direction
    delay(500);

    // Setup code from code_2
    while (!Serial);  // Wait for Serial connection
    if (!mlx.begin()) {
      Serial.println("Error connecting to MLX sensor. Check wiring.");
      while (1); // Hang if sensor is not connected
    };
}

void loop() {
    // Main loop of the program

    // Read and process gas sensor data
    readAndProcessGasSensor();

    // Read distance from forward direction
    distance_F = Ultrasonic_read();
    Serial.print("D F="); Serial.println(distance_F);

    // IR based line follower logic
    if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){
      if(distance_F > Set){forword();}
      else{
        Stop();
        Check_side();
      }
    }
    else if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){ turnRight();}
    else if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();}
    else if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 1)){
      Stop();
      delay(10);
      backword();
```

```
        delay(10);
        Check_side();
        forword();
    }

    // Print temperatures from MLX sensor
    Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempC());
    Serial.print("*C\tObject = "); Serial.print(mlx.readObjectTempC()); Serial.println("*C");
    Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempF());
    Serial.print("*F\tObject = "); Serial.print(mlx.readObjectTempF()); Serial.println("*F");
    Serial.println();

    // Send data to ESP-12F
    String dataToSend = String(distance_F) + "," + String(sensorValue) + "," +
String(mlx.readAmbientTempC()) + "," + String(mlx.readObjectTempC());
    espSerial.println(dataToSend);
    Serial.println(dataToSend);
    delay(500);
}

// Functions from code_1

void servoPulse(int pin, int angle) {
    // Generate pulse for servo motor to move it to a specific angle
    int pwm = (angle * 11) + 500;
    digitalWrite(pin, HIGH);
    delayMicroseconds(pwm);
    digitalWrite(pin, LOW);
    delay(50);
}

long Ultrasonic_read() {
    // Read distance using ultrasonic sensor
    digitalWrite(trigger, LOW);
    delayMicroseconds(2);
    digitalWrite(trigger, HIGH);
    delayMicroseconds(10);
    long time = pulseIn(echo, HIGH);
    return time / 29 / 2;  // Convert time to distance
}

void readAndProcessGasSensor() {
    // Read and process data from gas sensor
    sensorValue = analogRead(GAS_SENSOR_PIN);
    digitalValue = digitalRead(GAS_SENSOR_DIGITAL_PIN);

    // Determine the type of gas based on the sensor reading
    if (sensorValue >= 10 && sensorValue <= 300) {
        Serial.println("NH3");
    } else if (sensorValue > 300 && sensorValue <= 1000) {
        Serial.println("Benzene");
    } else if (sensorValue > 1000 && sensorValue <= 2000) {
        Serial.println("CO2");
    } else {
        Serial.println("Unknown or out of range");
    }
```

```
    // Turn on LED if sensor value exceeds threshold
    if (sensorValue > 400) {
      digitalWrite(13, HIGH);
    } else {
      digitalWrite(13, LOW);
    }

    Serial.println(sensorValue, DEC);
    Serial.println(digitalValue, DEC);
    delay(1000);
}

// Additional movement functions for the robot

void compareDistance() {
    // Compare distances from left and right and decide the direction to move
    if(distance_L > distance_R){
      turnLeft();
      delay(500);
      forword();
      delay(600);
      turnRight();
      delay(500);
      forword();
      delay(600);
      turnRight();
      delay(400);
    }
    else {
      turnRight();
      delay(500);
      forword();
      delay(600);
      turnLeft();
      delay(500);
      forword();
      delay(600);
      turnLeft();
      delay(400);
    }
}

void Check_side() {
    // Check distances from left and right and decide the direction to move
    Stop();
    delay(100);

    // Move servo to right and read distance
    for (int angle = 70; angle <= 140; angle += 5)  {
      servoPulse(servo, angle);
    }
    delay(300);
    distance_R = Ultrasonic_read();
    Serial.print("D R=");Serial.println(distance_R);
    delay(100);
```

```
    // Move servo to left and read distance
    for (int angle = 140; angle >= 0; angle -= 5)  {
      servoPulse(servo, angle);
    }
    delay(500);
    distance_L = Ultrasonic_read();
    Serial.print("D L=");Serial.println(distance_L);
    delay(100);

    // Move servo to center
    for (int angle = 0; angle <= 70; angle += 5)  {
      servoPulse(servo, angle);
    }
    delay(300);
    compareDistance();
}

// Movement functions for the robot

void forword() {
    // Move forward
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}

void backword() {
    // Move backward
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
}

void turnRight() {
    // Turn right
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
}

void turnLeft() {
    // Turn left
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}

void Stop() {
    // Stop the robot
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
```

```
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
}
```

2) Code for WiFi ESP-12F(esp8266):

```
#include <ESP8266WiFi.h>

// Define constants for WiFi and ThingSpeak settings
const char* ssid = "VM9517632";                // WiFi SSID
const char* password = "rv6RdxyfVfxk";         // WiFi Password
const char* host = "api.thingspeak.com";       // ThingSpeak API host
const char* apiKey = "IBVKT4Q1M8P9W0UW";       // ThingSpeak API key

void setup() {
  Serial.begin(9600);                    // Initialize serial communication at 9600 baud
rate
  WiFi.begin(ssid, password);                // Start WiFi connection using SSID and
password

  // Wait until WiFi is connected
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);                          // Delay for half a second
    Serial.print(".");                   // Print a dot to indicate waiting
  }

  Serial.println("");                    // Move to a new line after the dots
  Serial.println("WiFi connected!");         // Print WiFi connection success message
  Serial.print("IP address: ");              // Print local IP address of the ESP8266
  Serial.println(WiFi.localIP());
}

void loop() {
  // Check if there's any data available in the serial buffer
  if (Serial.available()) {
    String dataReceived = Serial.readStringUntil('\n');  // Read the data until a
newline character
    Serial.println(dataReceived);              // Print the received data for debugging

    // Split the received data into its respective parts
    int commaIndex1 = dataReceived.indexOf(',');
    int distance_F = dataReceived.substring(0, commaIndex1).toInt();

    int commaIndex2 = dataReceived.indexOf(',', commaIndex1 + 1);
    int sensorValue = dataReceived.substring(commaIndex1 + 1,
commaIndex2).toInt();

    int commaIndex3 = dataReceived.indexOf(',', commaIndex2 + 1);
    float ambientTemp = dataReceived.substring(commaIndex2 + 1,
commaIndex3).toFloat();
```

```
    float objectTemp = dataReceived.substring(commaIndex3  + 1).toFloat();

    // Start the process to send data to ThingSpeak
    WiFiClient client;
    if (client.connect(host,  80)) {              // Check if client is connected to the
ThingSpeak API host
      String url = "/update?api_key=";
      url += apiKey;
      url += "&field1=" + String(distance_F);   // Add distance_F data to the URL
      url += "&field2=" + String(sensorValue);  // Add sensorValue data to the URL
      url += "&field3=" + String(ambientTemp);  // Add ambientTemp data to the URL
      url += "&field4=" + String(objectTemp);   // Add objectTemp data to the URL

      // Send the GET request to ThingSpeak API
      client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                   "Host: " + host + "\r\n" +
                   "Connection: close\r\n\r\n");
      delay(10);
      // Print the response from ThingSpeak API for debugging
      while (client.available()) {
        String line = client.readStringUntil('\r');
        Serial.print(line);
      }
    }
  }

  delay(5000);  // Delay for 5 seconds before checking for new data again
}
```

- **Application:**
  1) Nuclear Power Plants: Areas of nuclear power facilities are exposed to high radiation levels. Such radiation exposure can have negative health effects on people. Radiation-resistant materials and radiation-detection sensors can be used to build the autonomous robot. It may conduct regular inspections of the reactor containment structures, look for coolant leaks, and guarantee the machinery is operating correctly. Furthermore, the robot can be the first response in the case of an emergency or failure, delivering real-time information without endangering human life.[10]

  2) Chemical Factories: Numerous poisonous, flammable, and reactive substances are processed and stored in chemical industries. Gas sensors and thermal cameras may be added to the robot to detect chemical leaks and unexpected heat sources that can be caused by a fire or chemical reaction, respectively. The robot may warn production staff to problems by seeing them early, helping to avert greater events or tragedies.[11]

  3) Mining Industry: Mines are vulnerable to collapses, gas buildups, and other dangers, especially those that are underground. The robot can explore the dark, cramped areas of mines using infrared cameras and gas sensors to search for pockets of poisonous or explosive gas. Additionally, it has the ability to examine the stability of tunnel ceilings and walls, assuring miner safety and assisting in the early identification of probable cave-ins. [12]

  4) Oil and Gas Industry: Pipelines and offshore drilling platforms are only two examples of the enormous and weather-exposed oil and gas infrastructure. The robot may be used to check pipes that are underwater for corrosion, leaks, or obstructions. It may check the structural soundness of offshore platforms and guarantee that safety features like fire suppression systems are working.[13]

  5) Disaster Zones: When disasters like earthquakes or tsunamis, structures may collapse, highways may become impassable, and regions may flood. Using its cameras and sensors, the robot can go across these treacherous terrains in order to find survivors, evaluate damage, or distribute supplies. Because of its autonomy, operations may run continuously, maximizing rescue efforts.[14]

  6) Military and Defense: Soldiers run a danger when conducting reconnaissance in hazardous settings. The robot may be used to examine battlegrounds, find landmines, and spy out possible dangers. It can acquire intelligence covertly thanks to its stealth technology, safeguarding the safety of personnel.[15]

  7) Space Exploration: Outer celestial bodies or planets frequently have harsh and hostile environments. The robot can be built to endure the harsh environments, radiation, and low gravity that are typical of planets like Mars. Future human missions can be facilitated by its ability to collect soil samples, investigate geological formations, and transmit data.[16]

  8) Agriculture: Large farms cover a lot of ground, making hand inspection time-consuming. The machine can independently go around fields, checking crops for

pests, illnesses, or water stress. Early problem detection allows farmers to take corrective action, resulting in higher yields.[17]

9) Marine Exploration: One of the least studied areas of our world is the ocean floor. The robot has sonar technology and pressure-resistant shells that allow it to operate underwater. It can investigate submerged ecosystems, examine shipwrecks, and keep an eye on cables and pipelines.[18]

10) Civil Infrastructure: Regular inspections are necessary to maintain the security of infrastructure including bridges, tunnels, and skyscrapers. With cameras to spot cracks, corrosion, or structural problems, the robot can scale heights or navigate through confined locations. Accidents and infrastructure breakdowns can be avoided by these inspections.[19]

Results:

```
16:51:44.173 ->
16:51:44.173 -> 0,752,23.67,23.21
16:51:44.629 -> Benzene
16:51:44.629 -> 751
16:51:44.629 -> 0
16:51:46.308 -> D F=0
16:51:48.216 -> D R=0
16:51:50.951 -> D L=0
16:51:54.744 -> Ambient = nan*C Object = 23.33*C
16:51:54.777 -> Ambient = 74.61*F       Object = 73.99*F
16:51:54.811 ->
16:51:54.811 -> 0,751,23.67,23.33
16:51:55.256 -> Benzene
16:51:55.256 -> 752
16:51:55.256 -> 0
16:51:56.964 -> D F=0
16:51:58.860 -> D R=0
16:52:01.619 -> D L=0
16:52:05.369 -> Ambient = nan*C Object = 23.33*C
16:52:05.401 -> Ambient = 74.61*F       Object = 73.99*F
16:52:05.434 ->
16:52:05.434 -> 0,752,23.67,23.33
16:52:05.924 -> Benzene
16:52:05.924 -> 751
16:52:05.924 -> 0
16:52:07.621 -> D F=0
16:52:09.492 -> D R=0
16:52:12.264 -> D L=0
```

Figure 3.1

a) Data Acquisition and Processing:

- Gas Sensor Analysis: The device made use of a gas sensor to identify various gases in its surroundings. Using the recorded measurements and the chosen thresholds. The equipment consistently gave a value of "752," which indicated the presence of benzene in the surrounding environment. These measurements are essential because they may be used to monitor and guarantee safe gas levels, particularly in industrial settings or places that are prone to pollution.
- Ultrasonic Sensor Insights: In order to assess distances in the front (D F), right (D R), and left (D L), ultrasonic sensors were used. These sensors measure distance by sending out sound waves and timing how long it takes for the echo to return. A reading of 0 may suggest that there is no obstacle within the range of the sensor.
- MLX90614 Temperature Sensor: The ambient and object temperatures were recorded using the MLX90614 infrared temperature sensor.

Particularly in applications where contactless temperature measurements are necessary, its values are important.

- Ambient Temperature: A reading mistake is likely the cause of the reported 'NaN' (Not a Number). This could be attributed to a sensor malfunction because the provided temp. is in Fahrenheit “74.61 °F”.
- object Temperature: A constant value of about “23.33 °C” and “73.99 °F” was obtained, suggesting that the temperature of the intended item was stable.

b) Data Transmission and Cloud Integration:

```
16:51:35.088 -> 0,751,23.69,23.17
16:51:45.231 -> 0,752,23.67,23.21
16:51:55.376 -> 0,751,23.67,23.33
16:52:05.610 -> 0,752,23.67,23.33
16:52:20.740 -> 0,751,23.69,23.41
16:52:30.882 -> 0,751,23.69,23.15
16:52:41.025 -> 0,752,23.67,23.17
16:52:51.124 -> 0,751,23.69,23.21
16:53:01.284 -> 0,752,23.65,23.11
16:53:11.426 -> 0,751,23.67,23.27
16:53:21.569 -> 0,751,23.69,23.27
16:53:31.712 -> 0,752,23.69,23.29
16:53:41.825 -> 0,751,23.63,23.29
16:53:51.989 -> 0,751,23.67,23.35
16:54:07.201 -> 0,751,23.65,23.39
16:54:17.345 -> 0,751,23.63,23.27
16:54:27.461 -> 0,751,23.67,23.27
16:54:37.575 -> 0,751,23.65,23.33
16:54:47.716 -> 0,752,23.63,23.17
16:54:57.834 -> 0,751,23.69,23.23
16:55:08.072 -> 0,752,23.67,23.17
```

Figure 5.2

Data collection and initial processing were handled via the Arduino Uno R3. The ESP-12F was used for data transfer and cloud integration. The ESP-12F successfully received a concatenated string of the type 0,752,23.67,23.21, demonstrating the smooth interaction between the Arduino and ESP-12F.

c) ThingSpeak Cloud Platform:

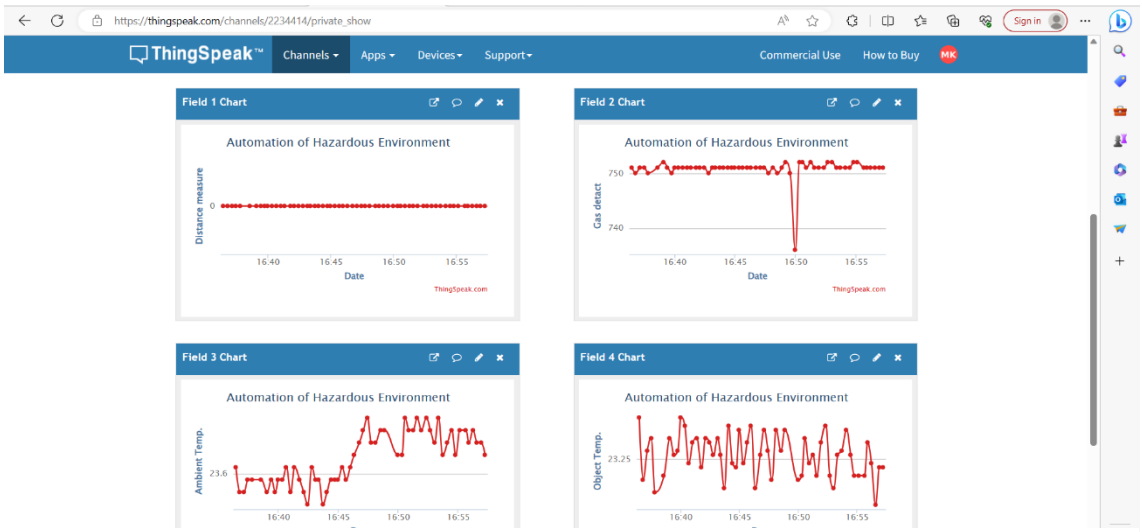

Figure 5.3

The transmitted data is represented visually by ThingSpeak, giving users an understanding of how well the system is functioning and how the environment is doing.

- Front Distance: Based on the results, it appears that the environment is still clear or that there may be sensor-related problems.
- Gas Sensor Reading: Benzene is consistently detected and shown.
- Temperature measurements: The ambient temperature appears to be having a problem, yet the temperature of the item shows a consistent curve.

The cloud visualization is essential, particularly for remote monitoring and possible IoT system or automation routine integration.

Conclusion:

The project achieved its goal of developing an autonomous robot for hazardous environment inspection. The robot's ability to operate in places like nuclear power plants and catastrophe areas is ensured using algorithms for autonomous navigation and inspection together with a durable architecture. Successful ThingSpeak integration demonstrates real-time data transfer to cloud systems, highlighting its capacity for remote monitoring. This initiative not only confirms robotics' promise for safety and surveillance applications, but it also prepares the way for future developments in the field of autonomous inspection robots.

Bibliography:

- WWW.Github.com
- WWW.Googlescholar.com
- WWW.Researchgate.net
- WWW.IEEEXplore.IEEE.org
- WWW.Thingspeak.com

References:

1. Omari S., Gohl P., Burri M., Achtelik M., Siegwart R., (2014). Visual Industrial Inspection Using Aerial Robots. In Proceedings of the IEEE International Conference on Applied Robotics and Power Industries (ICRA), 3212-3217.

2. Su H., Wu Z., Zhang H., Du Q., (2022)., Hyperspectral Anomaly Detection: A survey IEEE Geoscience and Remote Sensing Magazine (2022), 64-90, 10(1).

3. Cuebong Wong, Erfu Yang, Xiu-Tian Yan and Dongbing Gu (2017). An overview of robotics and autonomous systems for harsh environments. In Proceedings of the IEEE 23rd International Conference on Autonomous and Computing (pp. 1).

4. Mur-Artal R., Montiel J., Tardos J. (2015), ORB-SLAM: A Versatile and Accurate Monocular SLAM System., IEEE Transactions on Robotics, (2015), 1147-1163, 31(5)

5. Rusu, R. B., Blodow, N., & Beetz, M. (2011). Fast Point Feature Histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 3212-3217.

6. Razjigaev, A., Sievers, D., & Wahl, F. M. (2019). Visual recognition of hazardous materials for inspection robots in industrial environments. IEEE Transactions on Industrial Informatics, 15(3), 1792-1801.

7. Maione, G., Bicego, M., & Petrioli, C. (2020). Hyperspectral imaging for gas detection: a survey. IEEE Sensors Journal, 20(2), 493-509.

8. Bagnell, J., Choi, B., Hebert, M., & Hebert, M. (2019). Learning to explore using active SLAM. IEEE Transactions on Robotics, 35(6), 1505-1520.

9. Santos, T., Mendes, R., & Martins, R. (2020). Experimental validation of an autonomous robot for nuclear power plant inspection. IEEE Transactions on Automation Science and Engineering, 17(1), 290-300.

10. Tsai, C.-C., Liao, Y.-H., & Huang, C.-Y. (2017). Development of an inspection robot for nuclear power plants. Robotics and Computer-Integrated Manufacturing, 44, 218-228.

11. Guarnieri, M., Bove, M., & Lunghi, E. (2019). Drones in chemical plants for inspection services. Reliability Engineering & System Safety, 183, 173-181.

12. Ye, S., & Baiden, G. (2011). Integrating 3D modeling, photogrammetry and design for immersive asset information modelling. Automation in Construction, 20(4), 418-432.

13. Al-Kaabi, S., Ayer, S. K., & Fiaz, A. (2018). Drones in the oil and gas industry: applications and challenges. Energy Exploration & Exploitation, 36(1), 2-15.

14. Murphy, R. R., Steimle, E., Griffin, C., Cullins, C., Hall, M., & Pratt, K. (2008). Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. Journal of Field Robotics, 25(3), 164-180.

15. Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. Robotics and Autonomous Systems, 61(12), 1258-1276.

16. Pedersen, L., Bualat, M., Kunz, C., Lee, S. Y., Smith, T., Staritz, P., & Fong, T. (2007). Autonomous instrument placement for Mars exploration rovers. Autonomous Robots, 22(4), 351-371.

17. Slaughter, D. C., Giles, D. K., & Downey, D. (2008). Autonomous robotic weed control systems: A review. Computers and Electronics in Agriculture, 61(1), 63-78.

18. Alcocer, A., Oliveira, P., & Pascoal, A. (2006). Study and implementation of an EKF GIB-based underwater positioning system. Control Engineering Practice, 14(9), 1099-1110.

19. Dorafshan, S., & Thomas, R. J. (2018). Bridge inspection: Human performance, unmanned aerial vehicles, and machine learning. Journal of Structural Engineering, 144(4), 04018024.

APPENDIX A: PROJECT AGREEMENT FORM

UNIVERSITY OF HERTFORDSHIRE
SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

MSc Project Supervision Agreement

Your individual project is an important part of your MSc Course, a great deal of staff time is also contributed to assist and supervise MSc project students. You are therefore required to abide and acknowledge the following agreement.

1. I have agreed with the choice of project title and allocated supervisor, and I will not seek to make any change.
2. I will make every possible effort to attend regular meetings with my project supervisor on the agreed day and will arrive promptly at the agreed time. If I am unable to attend, I will inform mysupervisor prior to the scheduled meeting by phone or Email in all possible way.
3. I understand that my project supervisor may sometimes be unavoidably delayed due to other teaching commitments and responsibilities, I will wait at least ten minutes for him/her to return.
4. I also accept that my project supervisor may occasionally need to reschedule my appointment due to other commitments.
5. I will check my Email and/or MS Teams messages on daily basis and will act promptly on any request from my project supervisor.
6. If I cannot see my project supervisor in his/her office, I will leave a note on his/her desk stating the date and time when I tried to see him/her.
7. If I miss an appointment with my project supervisor for any reason, I will make contact with him/her as soon as possible to arrange another appointment.
8. If I am finding the project difficult, I will express my concern to my supervisor and seek for advice.
9. I will still attend regular meeting with my project supervisor even I feel as I am making good progress of the project work.
10. I understand that it is my responsibility to back up all project data using the University MS Office 365 Cloud service. Loss of data resulting from technical problems will not be accepted as a mitigating circumstance.
11. I will keep an electronic project logbook and have it checked and signed regularly by my supervisor.

APPENDIX B: PLAGIARISM STATEMENT

UNIVERSITY OF HERTFORDSHIRE
SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

I have read the detailed guidelines to Students on academic offenses and misconduct information (Click this link to read the Assessment Offences and Academic Misconduct web page)

I have read the information about the Academic integrity, misconduct and plagiarism posted on the Canvas Project module page, attended the seminar and/or watched the recording delivered by the School Academic Integrity Officer (SAIO).

I understand the University process of dealing with suspected cases of academic misconduct and the possible penalties.

Therefore, I certify that the work submitted is my own and that any material derived or quoted from the published or unpublished work of other persons has been duly acknowledged.
(Ref. University Policy Regulation AS14, Appendix III – version 16.0)

Signature: Mohitkumar Kori

Student name: Mohitkumar Kori
SRN: 21070997

YOU MUST SIGN THIS AGREEMENT AND ATTACHED THIS TOGETHER WITH THE INTERIM AND FINAL REPORT SUBMISSION

APPENDIX C: ASSESSMENT MARKING SCHEMES

Interim report (20%)	
Marking Scheme - Criterion	**Max mark**
General Presentation, Structure and Layout	10
Project management and Analysis of work stages (i.e. timeline, work plan, task specifications, H&S, ethical implication, project risk & capital resources)	20
Literature review (including correct & appropriate referencing)	25
Project Content (i.e. Introduction and background, clarity of aims, objectives & applied methods)	25
Evidence project Engagement and commitment (i.e. supervisory meeting log & project book)	20
TOTAL:	**100**

Project Report (60%)	
Marking Scheme - Criterion	**Max mark**
General Presentation, Structure and Layout	10
Literature review (including correct & appropriate referencing)	20
Originality (Professionalism & project ownership)	10
Rigour and technical depth	15
Quality of Analysis & Discussion of Outcomes (critical evaluation)	30
Objective achieved & Project management Review	15
TOTAL:	**100**

Project defence (20%)	
Marking Scheme - Criterion	**Max mark**
General Presentation, Structure and Layout (including appropriate referencing)	10
Pace and clarity	10
Selection of material presented, results and critical analysis, etc.	40
Level of technical competence demonstrated (including answering questions)	15
Project Engagement and commitment (i.e. supervisory meeting log and project book)	25
TOTAL:	**100**

APPENDIX D: LOGBOOK TEMPLATE

Name:	**Mohitkumar Kori**
Project title:	**Development of an Autonomous Robot for Hazardous Environment Inspection**
Supervisor:	**Mr. Emilio Mistretta**

Student notes & date:	**8 Jun, 2023**
Planned tasks and Work Undertaken:	**first encounter, It's a simple project discussion. Specifically, where it will be useful and what the project's major goals are. I was assigned the literature review duty during the first meeting. Every week, Discussion with professor, he cleared my doubt and assigned me the task.**
Problems encountered/ questions for supervisor:	**I asked my confusion about component list and Prototype. Due to some internal issue of Nucleuo-F767Zi microcontroller and Mbed cloud, professor change my microcontroller and gave me Arduino Uno R3 with Arduino IDE software at the end of July.**

Project Meeting date & staff(s):	**11 August, 2023**
Meeting notes:	**The robot successfully works but communication is not generated. Start the working of Dissertation report.**
Action point(s):	**As per the guidance of professor, the communication successfully generated and made transferring the data from microcontroller to thingspeak IOT cloud**
Checked by:	
Date:	

APPENDIX E: PROJECT RISK AND THREAT ASSESSMENT CHECKLIST

This appendix is to be discussed during the initial supervisory meetings and must be included with the Interim report.

Please **PRINT** all information **CLEARLY**

Student's Name:	**Mohitkumar Kori**
SRN:	**21070997**
Project Title:	**Development of an Autonomous Robot for Hazardous Environment Inspection**
Supervisor's Name:	**Mr. Emilio Mistretta**

Please enter below your BEST ESTIMATES for the information requested.

1	Is your project likely to involve either deliberate, or possible accidental use or contact with: Voltages above 30V High currents at low voltage (i.e. lead acid batteries) Rotating machines High temperatures Hazardous fluids or gases	Yes	No
2	Will your project involve attaching electrodes to yourself or someone else in such a way as to produce low impedance contact?	Yes	No
3	If the answer to 2 is Yes, has an assessment of the risk in the planned procedure been carried out?	Yes	No
4	If the answer to 2 or 3 is Yes, does the analysis show an acceptable level of risk?	Yes	No
5	Are any substances classified as being 'Hazardous to Health' likely to be used in the course of your work?	Yes	No
6	Are any flammable substances likely to be used in the course of your work? (e.g. volatile cleaning agents, paint, etc.)	Yes	No
7	Are there any other aspects of your projected work which might impose a danger to yourself or to others? (Please specify):	Yes	No

Notes: If the answer to any of these questions is YES, you have to complete the Risk and Hazard table on the next page and ask your Supervisor to sign and approve it prior to submission of your Project Outline.

APPENDIX F: PROJECT RESOURCES CHECKLIST

This appendix is to be discussed during the initial supervisory meetings and must be included with the Interim report.

Please **PRINT** all information **CLEARLY**

Student's Name:	**Mohitkumar Kori**
SRN:	**21070997**
Project Title:	**Development of an Autonomous Robot for Hazardous Environment Inspection**
Supervisor's Name:	**Mr. Emilio Mistretta**
Date:	**24 Jun, 2023**

Please enter below your BEST ESTIMATES for the information requested. You will not be penalised for small errors and later genuine changes - only for not bothering!

1	Will materials or components be ordered?	Yes	No
2	Will you be designing a PCB or drawing which will require manufacture during the course of your project? (Note that this should only be contemplated where absolutely necessary for the success of the project!)	Yes	No
3	Will PCB or drawing assemblies already designed and manufactured within the School be required? Please specify:	Yes	No
4	Will a PC be required for work other than for report writing?	Yes	No
5	If the answer to 4 is Yes, will you require any special hardware installed/attached (i.e.PROM emulation, USB interface). Please specify:	Yes	No
6	Will you require specific software, other than word processing? (Please specify):	Yes	No
7	Will your project require technical staff to make items requiring workshop facilities?	Yes	No
8	If the answer to 7 is Yes, which of the following workshop activities will be required (please circle): Drilling Sheet metal forming Lathe work Milling, Other (Please specify)		
9	Are there other resources required, not covered above? (Please specify):	Yes	No

APPENDIX G: INFORMATION OF ETHICS

Use of Human Volunteers in Project work - Guidance notes for students can be found at https://ask.herts.ac.uk/ethics-approval

The following section has been taken from a memorandum issued by the University Ethics Committee relating to the use of human volunteers in research and teaching. Bear in mind this code of practice when it is applicable to your project work.

Code of practice for the use of human subjects in research projects & teaching exercises in undergraduate & taught postgraduate courses:

- The remit of the University Ethics Committee is to consider and approve projects where there are ethical issues arising from research projects or teaching exercises involving human subjects. As part of its work, the Ethics Committee has produced the document "Regulations and Guidelines for the use of Human Volunteers in Research Projects and Practical Classes".
- These regulations are mandatory requirements of the University and all lecturers, researchers and students using human volunteers in their work must comply with these.
- This Code of Practice is a summary of these regulations and is intended to assist when setting up research projects in undergraduate and postgraduate courses where it is thought that ethical issues may be involved. For example, such issues may concern the administration of a substance, sampling of body fluids or assessing the fitness of a participant during physical exertion. Concerning issues that are mainly psychological, examples may include the amount of distress that might be caused by questions or procedures, and the measures to cope with that distress should it occur. Thus, the prime aim of this Code of Practice is to maintain ethical standards across the University and to involve Supervisors and students in the ethical assessment of their own projects.
- **The main points to consider are:**
  - Will the volunteers be recruited in a manner that allows them either to give consent or refuse to participate? Particular care should be exercised when the researcher has disciplinary or Supervisory control over the volunteer, or controls the academic progress of the volunteer. The right of a volunteer to withdraw from the project at any time must be respected.
  - Does the volunteer have an appropriate knowledge of his/her involvement in the nature of the study to the investigation (i.e. informed consent)? However, it should be noted that in some cases too much knowledge can interfere with the investigation.
  - Will the confidentiality of the volunteer be maintained at all times? If the viewing of case notes is a necessary part of the study, will the confidentiality of the subject (and the institution) be maintained?
  - Has the volunteer's written or oral consent been obtained? If children are involved, has the consent of a teacher or parent been obtained?
  - Is distress (physical and/or emotional) likely to be caused by participating in this study? Have adequate measures been taken to look after the volunteer should this occur?

If the answers to the above suggest that the proposed research is unethical, it will be necessary to amend the investigation. Where you are undecided, you may wish to talk this through with a member of the University Ethics Committee before making a formal submission for approval.

APPENDIX H: INDUSTRIAL PROJECTS

Industrially based project proposals may be submitted by **part-time students**. For these projects, an industrial guarantor is required.

Student and industry proposed projects are particularly important for part-time students who do not have official timetabled project sessions at the University and are expected to undertake their project at their place of work. Part-time students are encouraged to make early investigations into the possibility of undertaking a project at their place of employment.

Industrial guarantor's agreement:
This form is relevant to those projects that are being undertaken in industry. For an industrial project to run, this form must be completed and signed by the industrial guarantor and the student.

The role of the industrial guarantor:
All MSc students are required to undertake an engineering project as part of their degree. Some students, particularly those undertaking a part-time course while working in industry, choose to undertake projects in an industrial environment. For this group of students, each will have an academic supervisor at the University and an industrial guarantor at the workplace.

The role of the industrial guarantor is to provide the student with technical support in the workplace, and to guarantee that the student has undertaken the work as reported and without undue assistance.

Agreement:
Before completing this section, the prospective industrial guarantor should discuss the project with the student and broadly understand the terms of reference of the project and the student's role in the project.

The letter on company notepaper should be posted to MSc Project Tutor and should contain the wording given below before the project work is started.

I (industrial guarantor's name printed) _____ agree to

my role as industrial guarantor for the (student's name printed) _____

_____.

Signed (guarantor):		Date:	
Contact address & telephone number of guarantor:			
Signed (student):	Mohitkumar Kori	Date:	24 Jun, 2023
Contact address & telephone number of student:	27, Poets Green, LU4 0LQ, Luton 07436960671		