IoT Based Mine Security and Surveillance Robot

A Project Report

Submitted in Partial Fulfillment of the Requirements for ETE 3200 of

Bachelor of Science (B.Sc.)

in

Electronics & Telecommunication Engineering (ETE)

Submitted by:

Name: **Minhazul Islam**

Roll: **1804038**

Supervisor:

Md. Rakib Hossain

Assistant Professor,

Dept. of ETE, RUET

Dept. oF Electronics & Telecommunication Engineering (ETE)

Rajshahi University of Engineering & Technology (RUET)

Kazla, Rajshahi-6204

March 2023

Dept. of Electronics & Telecommunication Engineering (ETE)

Rajshahi University of Engineering & Technology (RUET)

Kazla, Rajshahi-6204

CERTIFICATE

I hereby certify that the Project titled ”IoT Based Mine Security and Surveillance Robot” which is submitted by Minhazul Islam, **Roll: 1804038** for fulfillment of the requirements for awarding of the degree of Bachelor of Science (B.Sc.) is a record of the project work carried out by the students under my guidance & supervision.

Place : Rajshahi,

Bangladesh

Date : March 2023

Md. Rakib Hossain

(Supervisor)

Assistant Professor

Department of ETE

Rajshahi University of Engineering & Technology (RUET)

ABSTRACT

The rapid growth of mining activities worldwide has increased the need for innovative security and surveillance solutions to ensure worker safety and protect valuable assets. The use of Internet of Things (IoT) technology in mining operations is gaining momentum, as it offers a reliable and cost-effective means of monitoring and controlling mine sites. This manuscript focuses on an IoT-based mine security and surveillance robot that uses a combination of sensors, cameras, and algorithms to detect and respond to potential security threats so that risks can be reduced with constant realtime monitoring and instant steps can be taken.

The robot is equipped with a range of sensors, including gas sensors, fire sensors, temperature sensors, and humidity sensors, which are used to monitor environmental conditions in the mine. It also has a camera system that captures live video footage and sends it to a particular device for analysis. The Brain of the Iot operation is NodeMCU. Through the use of NodeMcu, real time monitoring data can be observed in the Blynk Platform. If a certain level of range is crossed then there will be alert so that necessary steps can be taken for safety purposes. ESP32 CAM is going to help to surveillance the whole area and will send the data for real time monitoring. The gas sensor will tell if there is any leakage or life threatning gas is present in the mine or not. It'll also monitor the level of smoke along with gas. Temperature sensor will make sure that the mine is safe for the workers or not. The fire sensor will detect if a fire breaks out. Thus the security and surveillance purpose can be served in order to ensure the safety and protect valuable assests.

Keywords - IoT, Robot, ESP32 CAM, Blynk, NodeMCU.

ACKNOWLEDGEMENT

The successful completion of any task is incomplete and meaningless without giving any due credit to the people who made it possible without which the project would not have been successful and would have existed in theory.

First and foremost, we are grateful to **Dr. Md. Kamal Hosain**, HOD, Department of Electronics and Telecommunication Engineering, RUET, and all other faculty members of our department for their constant guidance and support, constant motivation and sincere support and gratitude for this project work. We owe a lot of thanks to our supervisor, **Md. Rakib Hossain**, Assistant Professor, Department of Electronics and Telecommunication Engineering, RUET for igniting and constantly motivating us and guiding us in the idea of a creatively and amazingly performed Major Project in undertaking this endeavor and challenge and also for being there whenever we needed his guidance or assistance.

We would also like to take this moment to show our thanks and gratitude to one and all, who indirectly or directly have given us their hand in this challenging task. We feel happy and joyful and content in expressing our vote of thanks to all those who have helped us and guided us in presenting this project work for our Major project. Last, but never least, we thank our well-wishers and parents for always being with us, in every sense and constantly supporting us in every possible sense whenever possible.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

The mining industry plays a vital role in the global economy, providing essential resources for various industries, from manufacturing to energy production. However, mining operations are fraught with various safety and security risks that pose significant challenges to miners and operators. The inherent dangers of mining, such as unstable terrain, gas leaks, and equipment failure, pose significant threats to worker safety and can result in substantial financial losses for mining companies. In recent years, the rapid development of Internet of Things (IoT) [1] technology has revolutionized the way mining operations are conducted. IoT-based solutions are increasingly being deployed in mining operations to provide real-time monitoring and control of critical mining infrastructure. One of the most promising applications of IoT in mining can be the use of IoT based mine security and surveillance robot.

The IoT based mine security surveillance robot is the combination of robotics and the collective network of connected devices and the technology that facilitates communication between devices and the cloud, as well as between the devices themselves. The combination of robotics and iot in this project can be said as Internet of Robotic Things which is probably upcoming next big thing. In this project smoke, gas, fire, temperature is going to be detected and measured using DHT11 sensor [2], MQ-2 sensor [3], flame sensor [4]. The surveillance will be done using ESP32 CAM module [5]. This whole operation is going to be monitored using NodeMCU [6]. For the build of robot and make it work Arduino Uno is going to be used along with L293d motor driver shield [7]. The DHT sensor is a type of

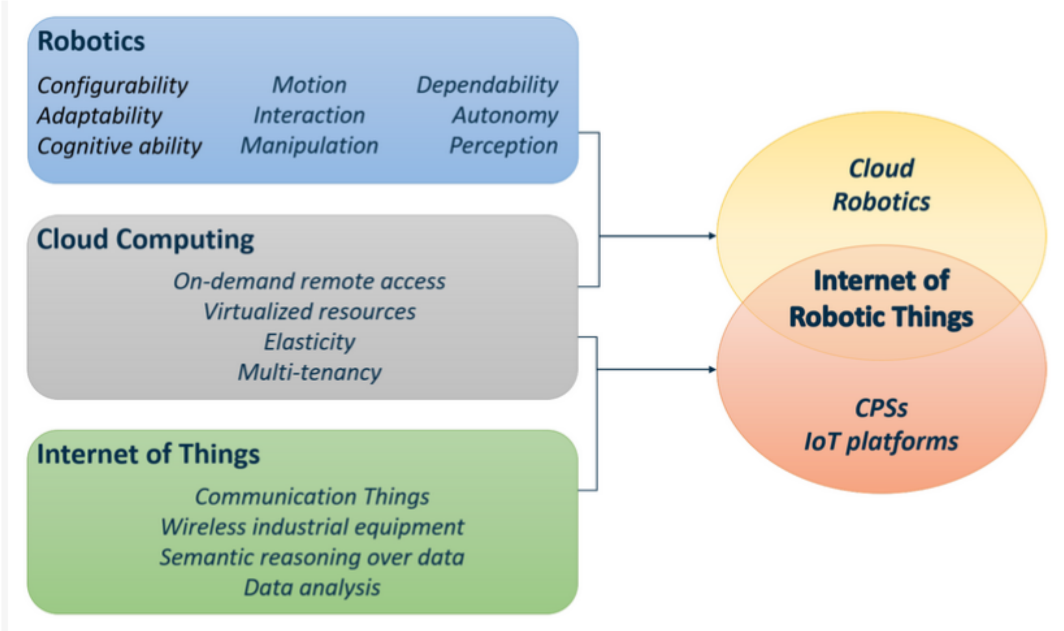

Figure 1.1: Internet of Robotic Things.

digital temperature and humidity sensor that can be used in various IoT applications. It is commonly used to measure the ambient temperature and humidity levels in a given environment. A thermistor is used to measure temperature. The microprocessor reads the thermistor's resistance and then translates it to a temperature value using a lookup table or an equation from the datasheet. Either Celsius or Fahrenheit can be used to read the temperature. A thermistor, a type of resistor that changes resistance in response to temperature, is used by the DHT11. The resistance of the thermistor fluctuates along with variations in temperature. The thermistor resistance is measured by the DHT11, which then translates the result to a temperature reading. DHT11, however, makes use of a capacitive humidity sensor. This kind of sensor consists of two electrodes separated by a layer that absorbs moisture. The capacity of the sensor to hold electrical charge changes as a result of the moisture-absorbing substance collecting more moisture as the air's humidity rises. The DHT11 calculates the humidity level from the change in capacitance. The MQ-2 sensor is a gas sensor that can detect various gases, including smoke, propane, methane, and carbon monoxide. It is commonly used in various IoT applications, such as fire detection systems and air quality monitoring systems. The MQ-2 sensor can provide analog output, which can be converted to a digital signal by an analog-to-digital converter (ADC) and can be interfaced with microcontrollers. The fire sensor is a type of sensor that can detect the presence of fire or heat. It is commonly used in various IoT applications, such as fire detection systems and smart home automation systems. The fire sensor can provide an analog output, which can be converted to a digital signal by an ADC and can be interfaced with microcontrollers. This Sensors are connected with NODEMCU which is the brain of IoT based operation. NodeMCU is an open-source firmware and development board based on the ESP8266 Wi-Fi chip. It is a popular platform for IoT projects due to its low cost, ease of use, and built-in Wi-Fi connectivity. NodeMCU can be programmed using the Lua scripting language or Arduino IDE, making it a versatile platform for developing IoT applications. NodeMCU and IoT are related in the sense that NodeMCU can be used as a development platform for building IoT applications. NodeMCU's built-in Wi-Fi connectivity makes it easy to connect to the internet and send and receive data to other devices or cloud platforms. By leveraging NodeMCU's processing power and connectivity features, developers can build IoT applications that collect data from various sensors, process the data, and send it to cloud platforms for analysis. The ESP32 Cam is a low-cost, low-power consumption Wi-Fi and Bluetooth enabled camera module. It is commonly used in various IoT applications, such as security camera systems and smart home automation systems. The ESP32 Cam can capture images and video and transmit them wirelessly to other devices, such as smartphones or computers. In the context of making a robot, an Arduino board and a driver shield can be used together to control the movement and actions of the robot. The Arduino board can be programmed to read inputs from various sensors, process the data, and send signals to the driver shield to control the motors and other components of the robot. In the near future, IoT will become broader and more complex in terms of scope. It will change the world by combining with robotics, AI and what not. This project is just a small leap towards that step.

1.2 Applications

IoT-based mine security and surveillance robots can have several applications in the mining industry, including:

<u>Oil and gas drilling:</u> IoT-based robots can be used in oil and gas drilling operations to monitor conditions, detect safety hazards, and optimize production.

<u>Mineral exploration:</u> IoT-based robots can be used in mineral exploration to collect

data on the composition and characteristics of rocks and minerals.

Coal mining: IoT-based robots can be used in coal mining operations to improve safety, monitor conditions, and increase efficiency.

Metal mining: IoT-based robots can be used in metal mining operations, such as copper or gold mines, to provide surveillance and monitoring of the mine environment.

Quarrying: IoT-based robots can be used in quarrying operations to monitor conditions, detect safety hazards, and improve efficiency.

Video Surveillance: The robots equipped with cameras can provide live video feeds of the mining site, allowing security personnel to monitor the site from a remote location. This can help to reduce the risk of accidents and theft. Most importantly if a catastrophe happens then video footage will help to rescue the people as soon as possible.

Underground mining and Open-pit mining: IoT-based robots can be particularly useful in underground mining operations where access is limited and hazardous conditions are common. IoT-based robots can be used in open-pit mining operations to monitor conditions, provide surveillance, and transport equipment and supplies.

Environmental Monitoring: IoT sensors can also be used to monitor environmental factors such as temperature, humidity, and air quality in the mine. This information can help to ensure that the working conditions are safe for the miners.

Emergency Response: In case of an emergency such as a fire or an accident, IoT-based mine security robots can be used to quickly identify the location of the incident and send alerts to emergency response teams.

Autonomous Operation: IoT-based mine security and surveillance robots can operate autonomously, without the need for human intervention. This can help to reduce labor costs, improve safety, and increase efficiency.

Ease of use and Efficiency: Depending on the situation, it may be easier to use voice commands to control the robot instead of manually manipulating a remote control. Having the option to use either method provides flexibility and convenience. And with the combination of IoT based system it'll be much more efficient.

1.3 Objectives

- To Monitor hazardous areas: The robot can monitor hazardous areas in mines where human access is difficult or dangerous.
- To Detect safety hazards: The robot can detect safety hazards, such as gas leaks or structural weaknesses, in the mine environment and alert workers.
- To Conduct surveillance: The robot can conduct surveillance of the mine, detecting and recording any suspicious activity or behavior.
- To Conduct safety checks: The robot can conduct safety checks in the mine, ensuring that all safety procedures and protocols are being followed.
- To Provide real-time data: The robot can collect and transmit real-time data on the mine environment, including temperature, air quality, and humidity.
- To Reduce labor costs: The robot can reduce labor costs associated with mine security and surveillance by providing continuous monitoring and reducing the need for human patrols.
- To Enhance safety: The robot can enhance safety in the mine by providing continuous monitoring and detection of safety hazards, reducing the risk of accidents and injuries.
- To Improve situational awareness: The robot can improve situational awareness in the mine by providing a real-time view of the mine environment to workers and managers.
- To Provide remote access: The robot can be remotely operated and controlled, allowing workers and managers to access the mine environment from a safe distance.
- To Increase efficiency: The robot can increase the efficiency of mine security and surveillance operations by automating repetitive tasks, such as patrol routes and safety checks.
- To Ensure compliance: The robot can ensure compliance with safety regulations and standards by monitoring and reporting on mine operations and conditions.

1.4 Motivation

The motivation behind the development of an IoT-based mine security and surveillance robot project is to improve safety and efficiency in mining operations. Mining is a hazardous and complex industry, with many potential safety risks, including gas explosions, cave-ins, and equipment malfunctions. Additionally, mining operations can be difficult to monitor and manage, with large distances, varying environmental conditions, and limited access to remote locations.

By developing an IoT-based mine security and surveillance robot, mining companies can overcome many of these challenges. The robot can provide real-time data on conditions within the mine environment, allowing workers and managers to identify safety hazards and potential risks before they become critical. The robot can also automate repetitive tasks, such as safety checks and patrol routes, reducing the need for manual labor and increasing efficiency. In a summary, the motivation behind the development of an IoT-based mine security and surveillance robot project is to create a safer and more efficient mining industry, by leveraging the latest in IoT technology to monitor and manage mining operations in real-time.

1.5 Challenges:

There are several challenges associated with the development and deployment of IoT-based mine security and surveillance robots. Here are a few:

1) To cope up with environmental condition as mining environments can be harsh, with extreme temperatures, high humidity, and exposure to dust and other particulate matter.

2) To maintain a reliable and stable communication link between the robot and the central command center.

3) IoT-based mine security and surveillance robots require a reliable and stable source of power, which can be difficult to provide in mining operations. Batteries may need to be replaced or recharged frequently, and power generators may be required in remote locations.

4) Mining operations often involve rough terrain, steep inclines, and uneven surfaces. IoT-based mine security and surveillance robots must be designed to operate in these

conditions, which can be challenging from a mobility perspective.

5) Many mining operations already have existing systems and equipment in place, such as conveyor belts, drilling rigs, and monitoring systems. IoT-based mine security and surveillance robots must be designed to integrate with these systems, which can add complexity to the project.

6) IoT-based mine security and surveillance robots are complex machines with many moving parts. Maintenance and repair of these robots can be challenging, particularly in remote locations where access to trained technicians and spare parts may be limited.

7) IoT-based mine security and surveillance robots can be expensive to design, develop, and deploy. Mining companies must carefully consider the cost-benefit ratio of investing in these robots, and whether the benefits of improved safety and efficiency outweigh the costs of implementation.

Chapter 2

Background Technologies and Circuit/Block Operation

2.1 Block Diagram

The block diagram representation of this project named ”IoT based Mine Security and Surveillance Robot” project is depicted below.

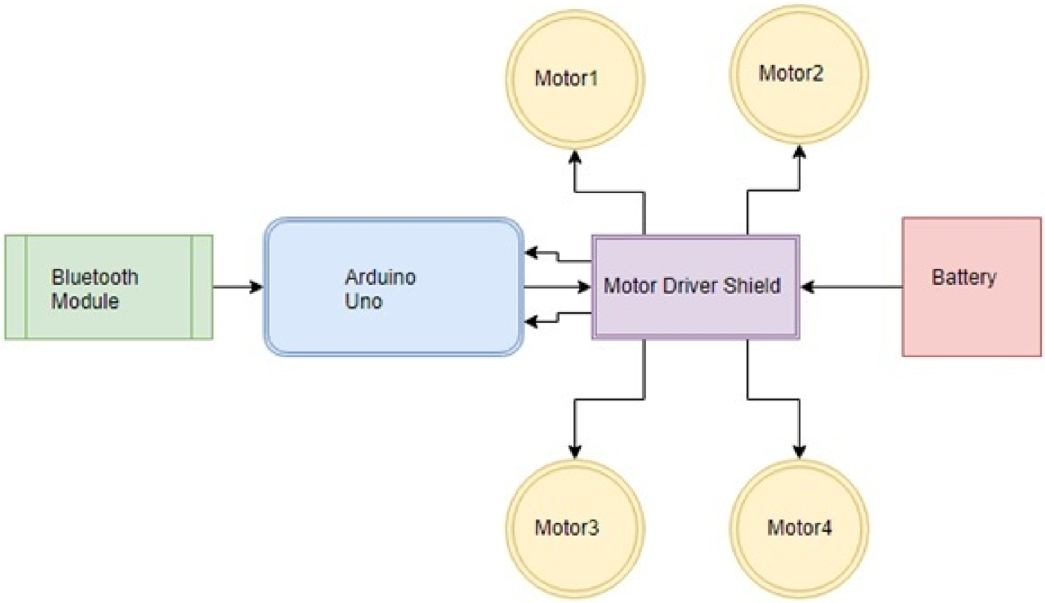

Figure 2.1: Block diagram representation of infrastructure of Robot

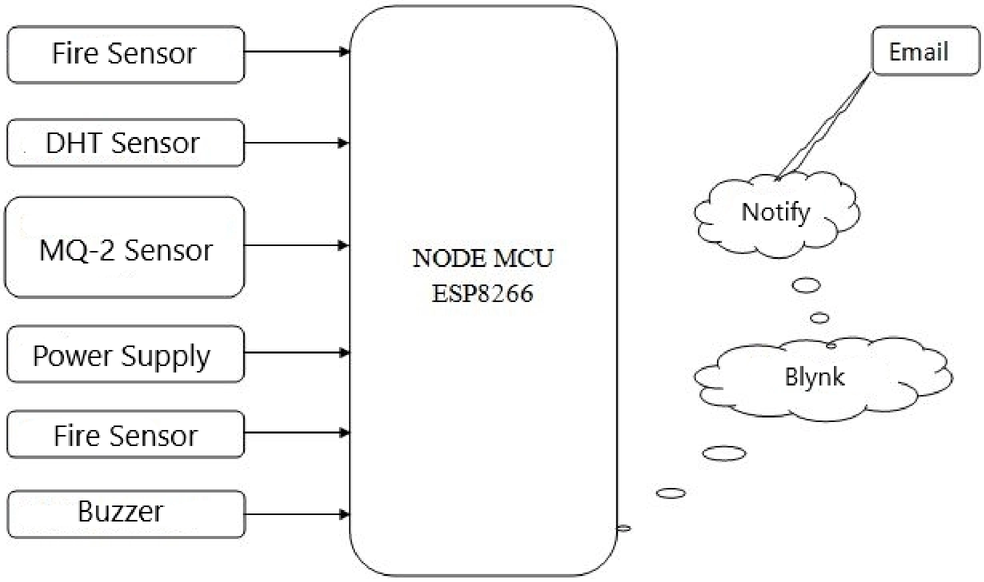

Figure 2.2: Block diagram representation of Interconnection between sensors and nodeMCU

2.2 Circuit Diagram

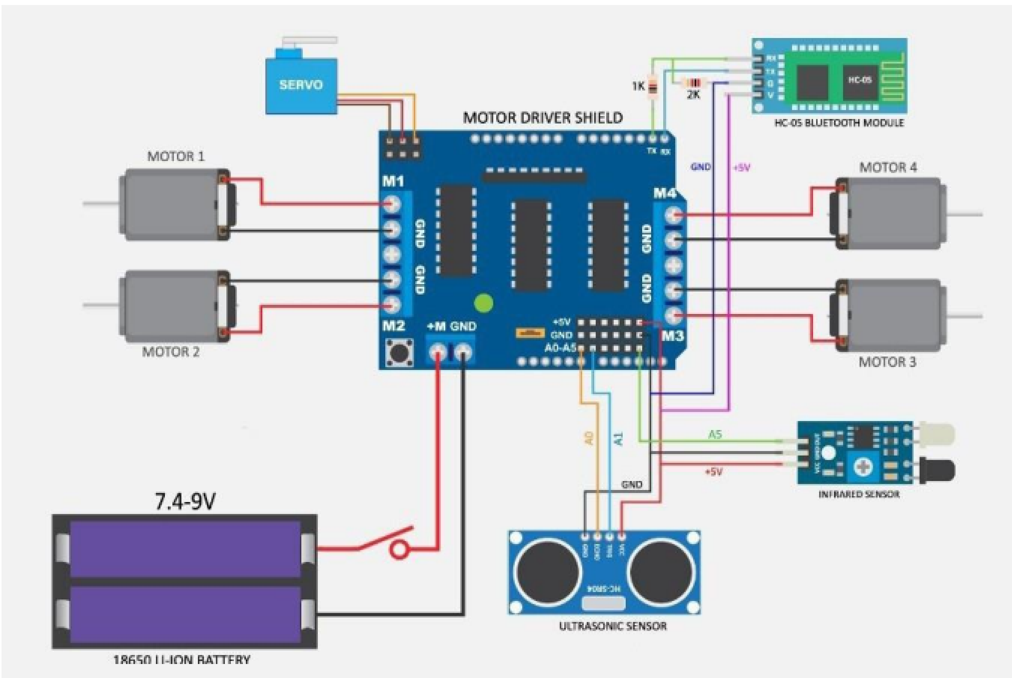

Figure 2.3: Circuit diagram of the infrastructure of the robot.

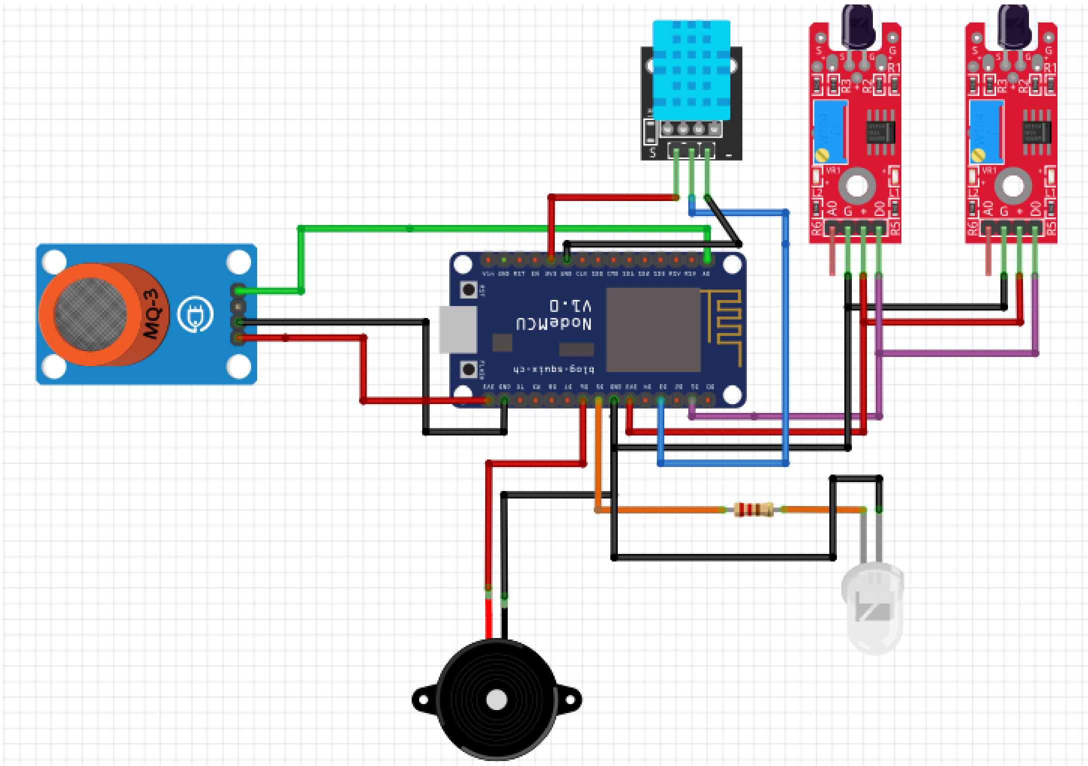

Figure 2.4: Circuit diagram of the interconnections between sensor and nodeMCU.

2.3 Required Sofware

1. Operating System: Windows 7/10/11
2. Express PCB
3. EasyEDA
4. Arudino IDE
5. Blynk
6. Fritzing

2.4 Required Equipment's

1. Arduino Uno
2. L293D Motor Driver
3. Gear Motors and Wheels
4. HC05 Bluetooth Module
5. Infrared Proximity Sensor
6. Ultrasonic Sensor

7. Servo Motor
8. Connecting Wires
9. 18650 lithium-ion battery
10. DC switch
11. Sheets
12. NodeMCU
13. DHT11 Sensor
14. Flame Sensor
15. MQ-2 sensor
16. Power Supply Module
17. ESP32 Cam
18. Mini Breadboard
19. Buzzer
20. LED
21. TTL module

The details descriptions of these components are described below:

1. Arduino Uno Technical Specifications [8]:

Microcontroller: ATmega-328P – 8 bit AVR family microcontroller;

Operating Voltage: 5V;

Recommended Input Voltage: 7-12V;

Input Voltage Limits: 6-20V;

Analog Input Pins: 6 (A0 – A5);

Digital I/O Pins: 14 (Out of which 6 provide PWM output);

DC Current on I/O Pins: 40 mA;

DC Current on 3.3V Pin: 50 mA;

Flash Memory: 32 KB (0.5 KB is used for Bootloader);

SRAM: 2 KB;

EEPROM: 1 KB.

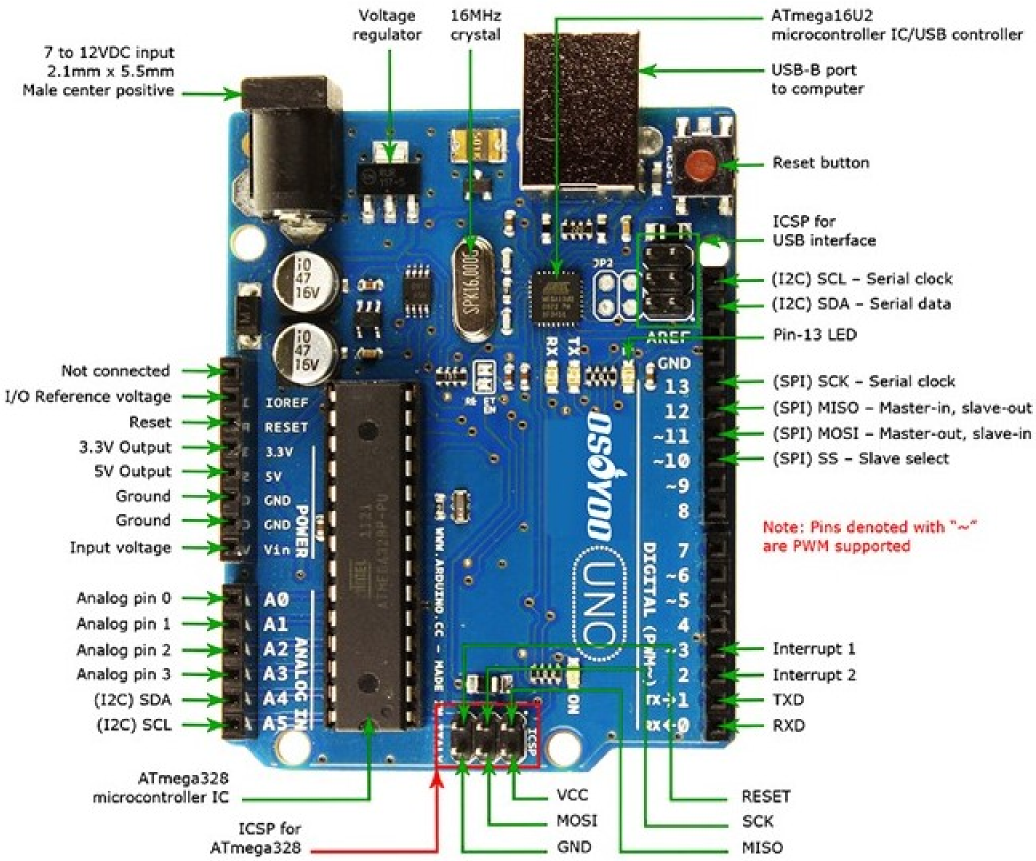

Figure 2.5: Arduino uno board description.

2. <u>NodeMCU 8266:</u>

Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106;

Operating Voltage: 3.3V;

Input Voltage: 7-12V;

Digital I/O Pins (DIO): 16;

Analog Input Pins (ADC): 1;

UARTs: 1;

SPIs: 1;

I2Cs: 1;

Flash Memory: 4 MB;

SRAM: 64 KB;

Clock Speed: 80 MHz.

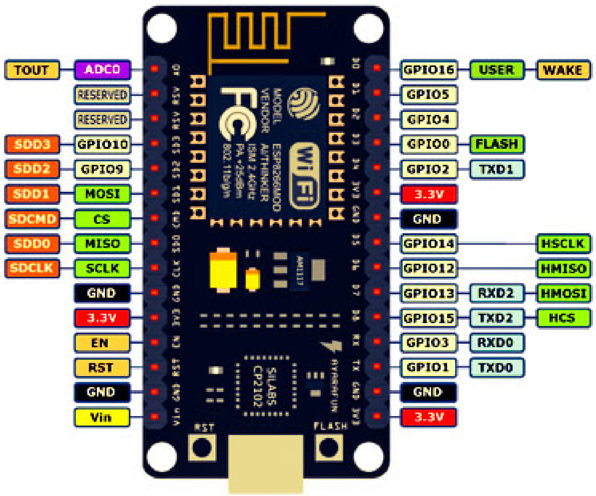

Figure 2.6: NodeMCU 8266 pin diagram

3. <u>ESP32 CAM:</u>

WIFI module: ESP-32S

Processor: ESP32-D0WD

Built-in Flash: 32Mbit

RAM: Internal 512KB + External 4M PSRAM

Antenna: Onboard PCB antenna

WiFi protocol: IEEE 802.11 b/g/n/e/i

Bluetooth: Bluetooth 4.2 BR/EDR and BLE

WIFI mode: Station / SoftAP / SoftAP+Station

Security: WPA/WPA2/WPA2-Enterprise/WPS

Peripheral interface: UART/SPI/I2C/PWM

IO port: 9

UART baudrate rate: default 115200bps

Power supply: 5V

Operating temperature: -20 °C   85 °C

Storage environment: -40 °C   90 °C, ¡90

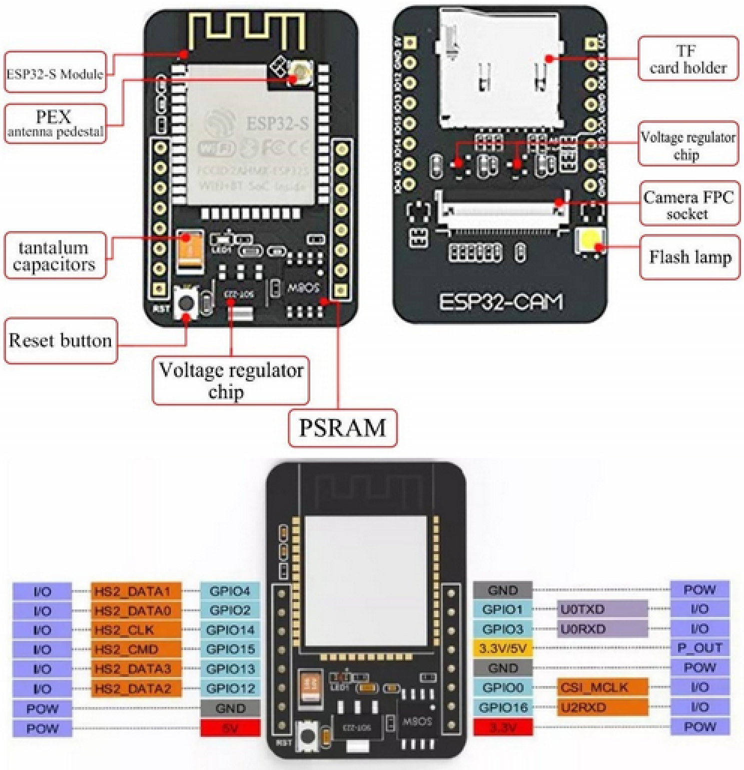

Figure 2.7: ESP32 CAM pin diagram

4. <u>TTL Module:</u> This section mentions some of the features and specifications of the FT232RL USB to TTL Converter:

Operating Voltage: 5V/3.3V DC

Max Current Draw: 5V - 500mA; 3.3V - 50mA

Connector: Mini USB

Fully integrated 1024 bit EEPROM storing device and CBUS I/O configuration

Data transfer rates from 300 baud to 3 Mbaud (RS422, RS485, RS232) at TTL levels

128 byte receive buffer and 256 bytes transmit buffer

Transmit and receive LED drive signals

Fully integrated clock generation with no external crystal required

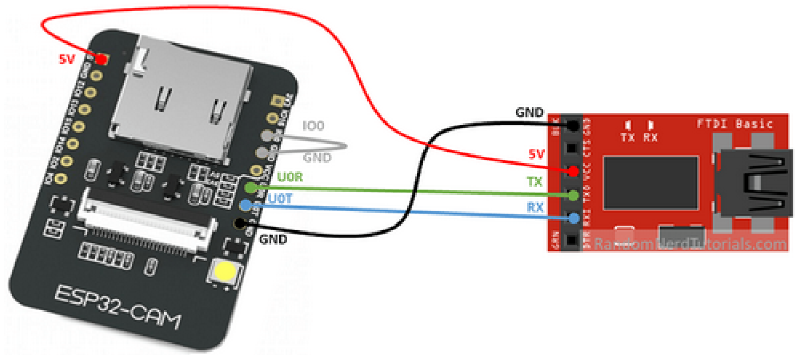

Figure 2.8: TTL Module connection with ESP32 CAM

5. DHT11 Humidity and Temperature Sensor: :

3 to 5V power and I/O

2.5mA max current use during conversion (while requesting data)

Good for 20-80percent humidity readings with 5percent accuracy

Good for 0-50°C temperature readings ±2°C accuracy

No more than 1 Hz sampling rate (once every second)

Body size 15.5mm x 12mm x 5.5mm

4 pins with 0.1” spacing

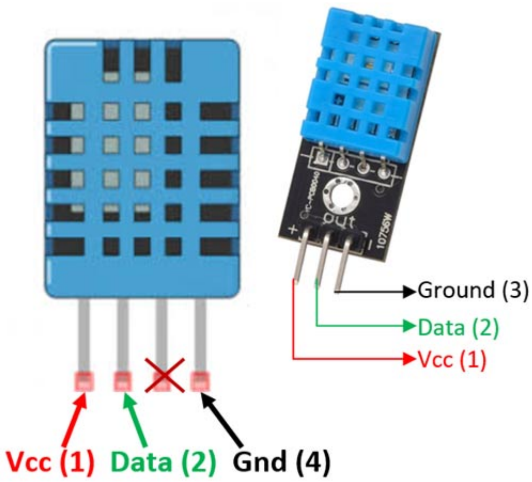

Figure 2.9: DHT11 pin Diagram

5. MQ-2 gas Sensor :

Sensitive for LPG, propane, hydrogen

Features boost circuit

Adjustable sensitivity

Signal output indicator

Power: 2.5V   5.0V

Dimension: 40.0mm * 21.0mm

Mounting holes size: 2.0mm

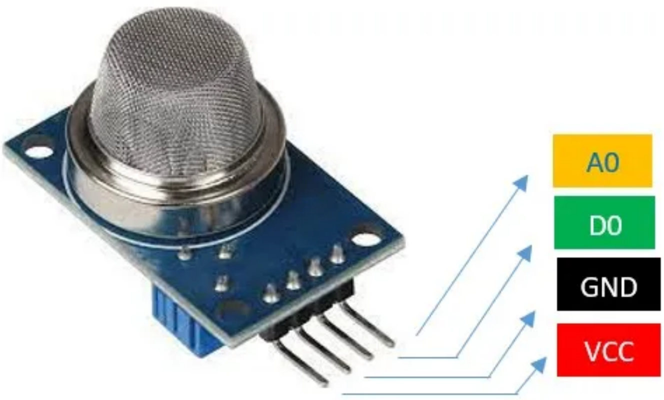

Figure 2.10: MQ-2 Gas and Smoke sensor

6. <u>Flame Sensor Sensor</u> :

Operating voltage: 4.75 – 5V max

Working current: 20 mA

Spectral bandwidth range: 760 – 1100 nm

Detection range: 0-1 m

-25 to 85°C

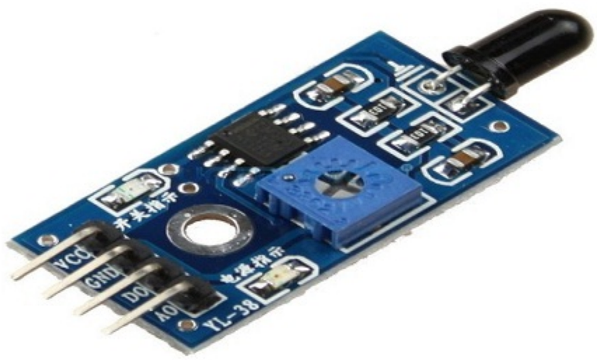

Figure 2.11: Flame sensor

7. <u>L293D Motor Driver Shield</u> :

Wide Supply-Voltage Range: 4.5 V to 36 V

Separate Input-Logic Supply

Internal ESD Protection

Thermal Shutdown

High-Noise-Immunity Inputs

Output Current 600 mA Per Channel

Peak Output Current 1.2 A Per Channel

Low Power Consumption, 80-µA Max ICC

Typical tpd = 13 ns

Low Input Current of 1

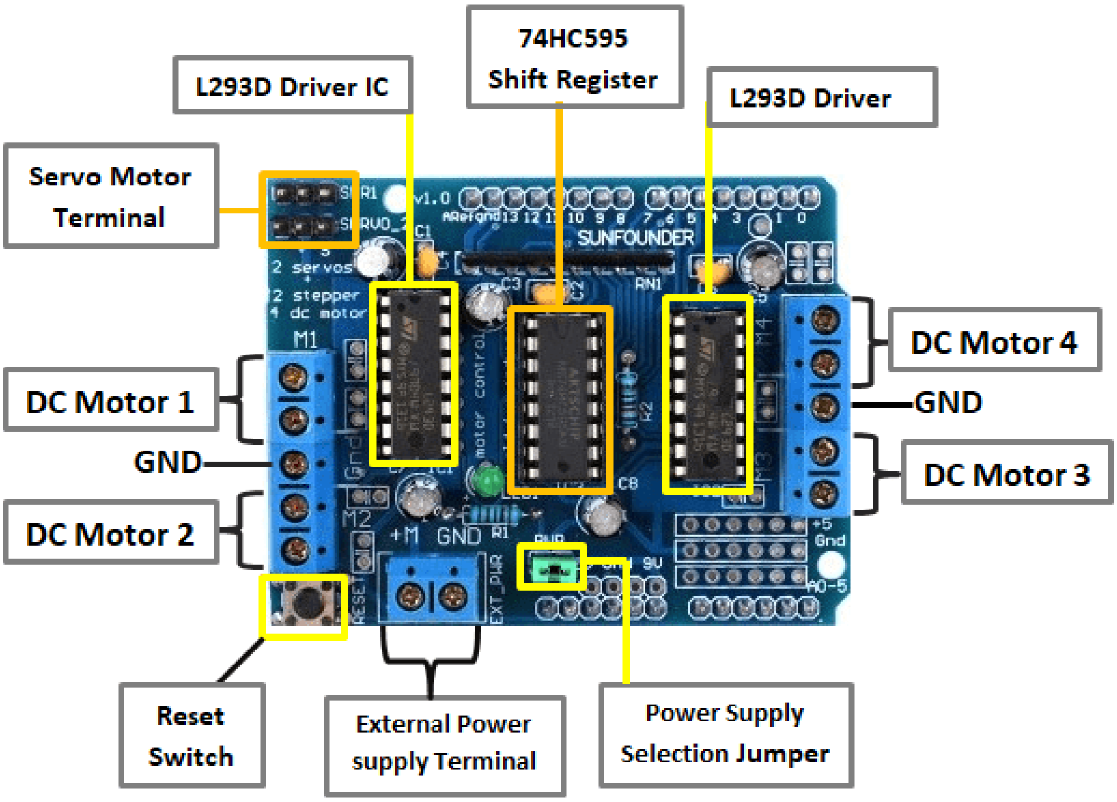

Figure 2.12: L293D motor driver shield pinout

8. <u>HC-06 Bluetooth Module</u> :

Bluetooth protocol: Bluetooth V2.0 protocol standard

Power Level: Class2 (+6dBm)

Band: 2.40GHz—2.48GHz, ISM Band

Receiver sensitivity: -85dBm

USB protocol: USB v1.1/2.0

Modulation mode: Gauss Frequency Shift Keying

Safety feature: Authentication and encryption

Operating voltage range: +3.3V to +6V

Operating temperature range: -20ºC to +55ºC

Operating Current: 40mA

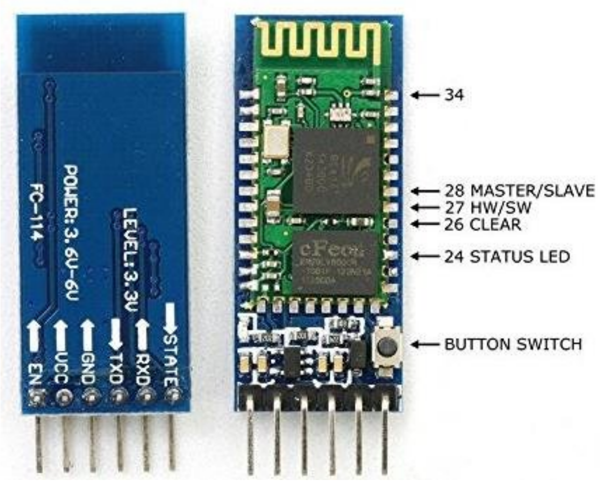

Figure 2.13: HC-06 Bluetooth Module

9. <u>Power Supply Module Module</u> :

Input Voltage (V): 6.5 to 12

Output Voltage(V):3.3/5

Maximum Output Current(mA): ¡700

Length (mm): 53

Width (mm): 32

Height (mm): 23

Weight (gm):8

Shipment Weight: 0.012 kg

Shipment Dimensions: 7 × 6 × 4 cm

Figure 2.14: Power Supply Module

10. <u>18650 Li-Ion Battery</u> :

Capacity: 1100-1450mAh

Actual Capacity:900mAh

Nominal Voltage: 3.7V

Charge Cycles: 1000

Fully Charge Current:4.2 V

Discharging Current: 3.0

Diameter: 50mm * 14mm

Weight: 18g

Figure 2.15: 18650 Li-ion battery

Chapter 3

Design & Implementation of the Project

3.1 PCB Layout Design

PCB is designed in ExpressPCB software. Which depicts the interconnection of the sensors and nodeMCU.

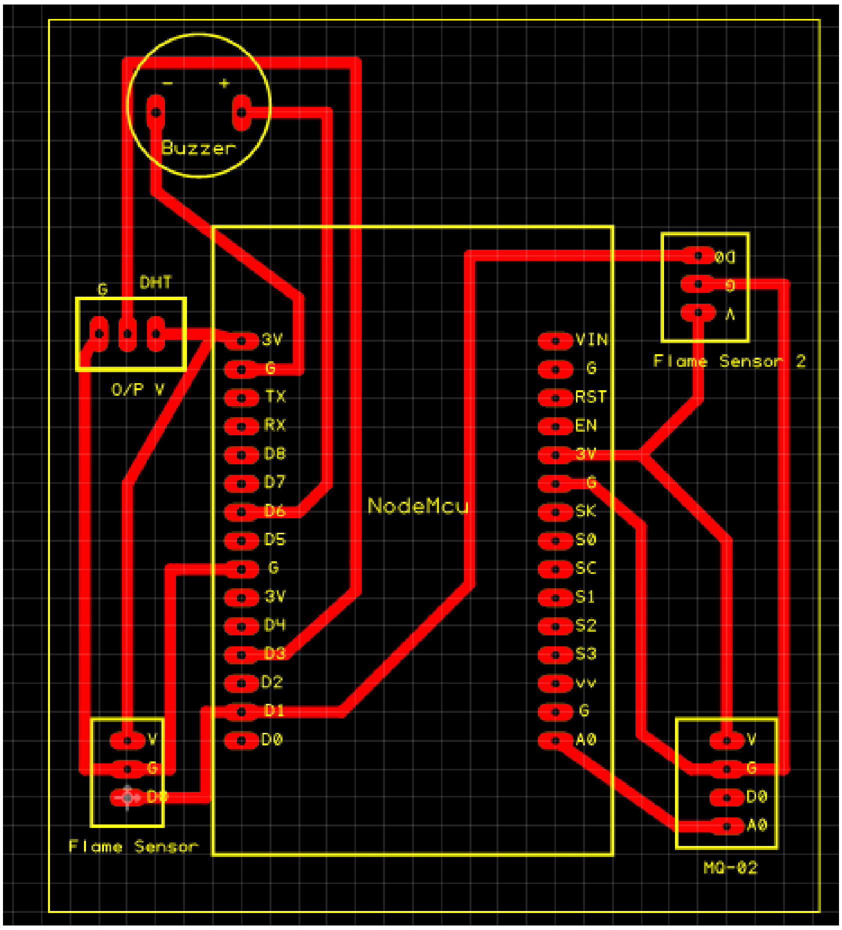

Figure 3.1: PCB layout Using ExpressPCB

3.1.1 PCB fabrication

The PCB layout fabrication is reresented below:

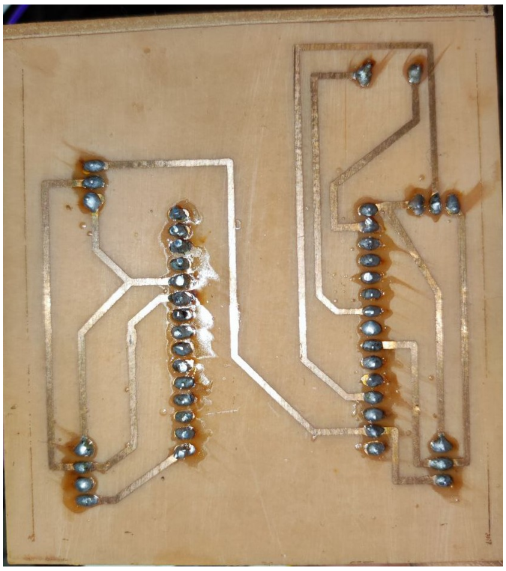

Figure 3.2: PCB layout after fabrication.

3.1.2 After Soldering

The fabricated pcb is sholdered. The final outcome after soldering is depicted here:

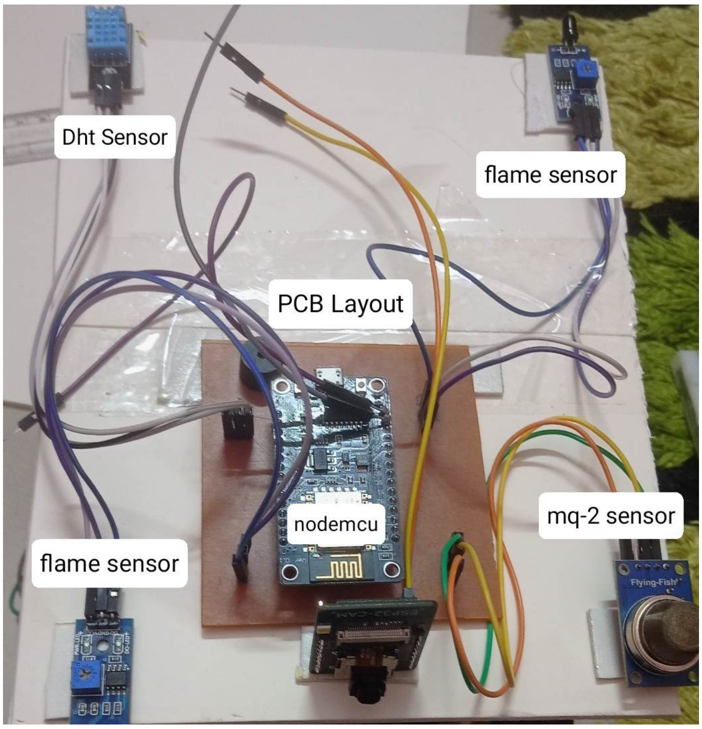

Figure 3.3: PCB layout after soldering.

3.2 Structure of the Robot

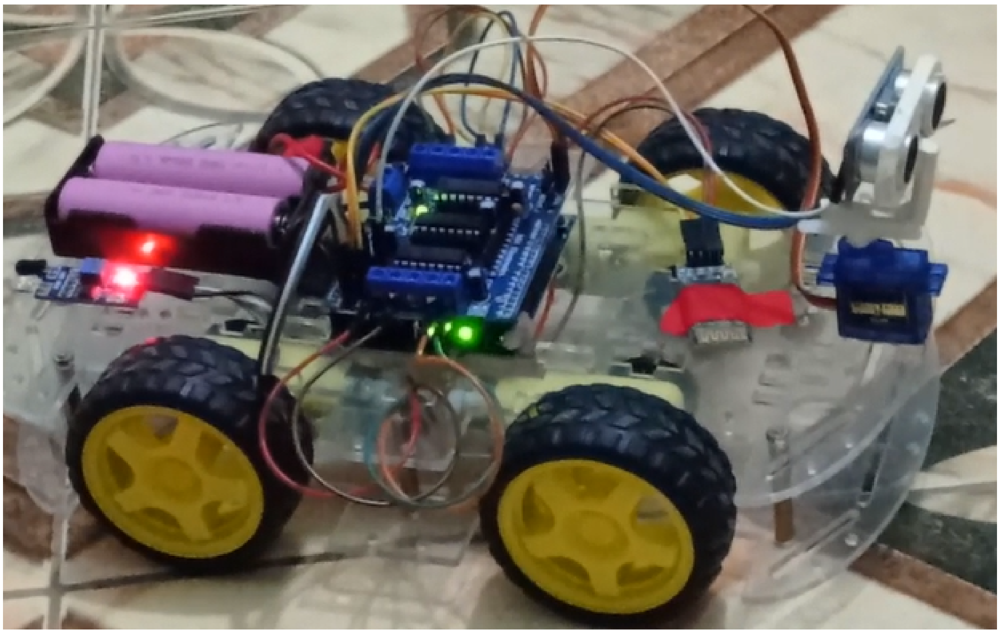

Figure 3.4: Structure of the robot

Chapter 4

Results Analysis & Discussion

4.1 Hardware Part

The set up is represented is depicted below which is the overall structure of the robot.

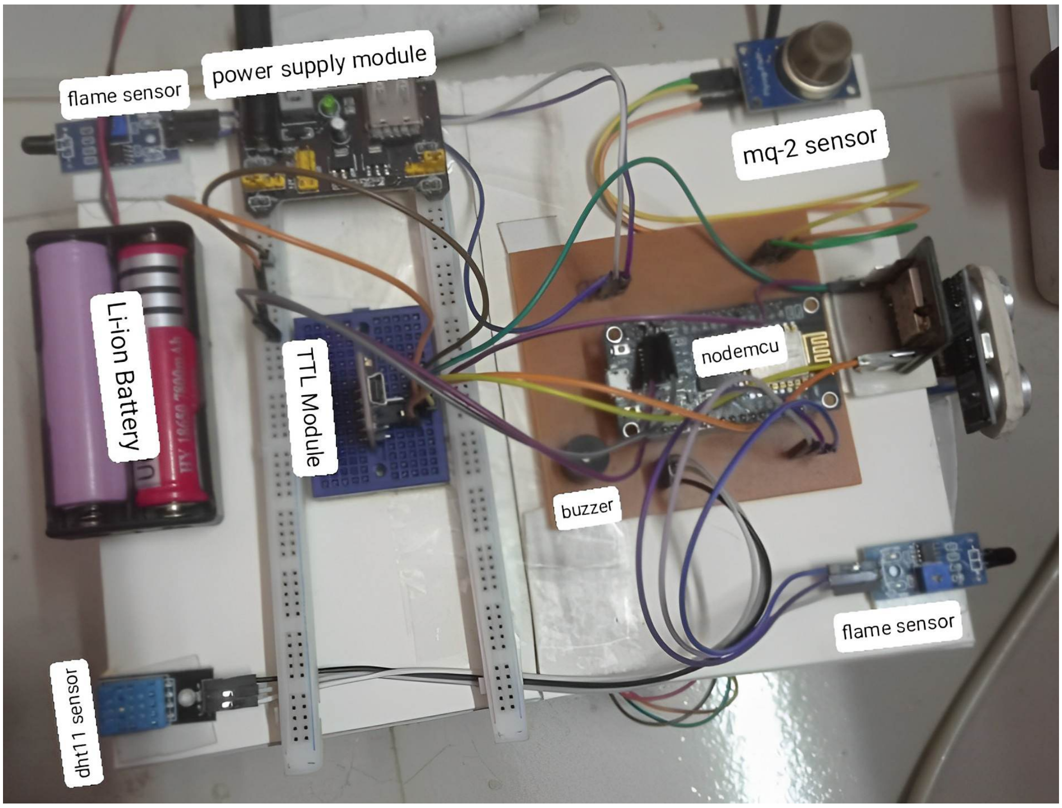

Figure 4.1: Upper Portion of The Robot after assembly

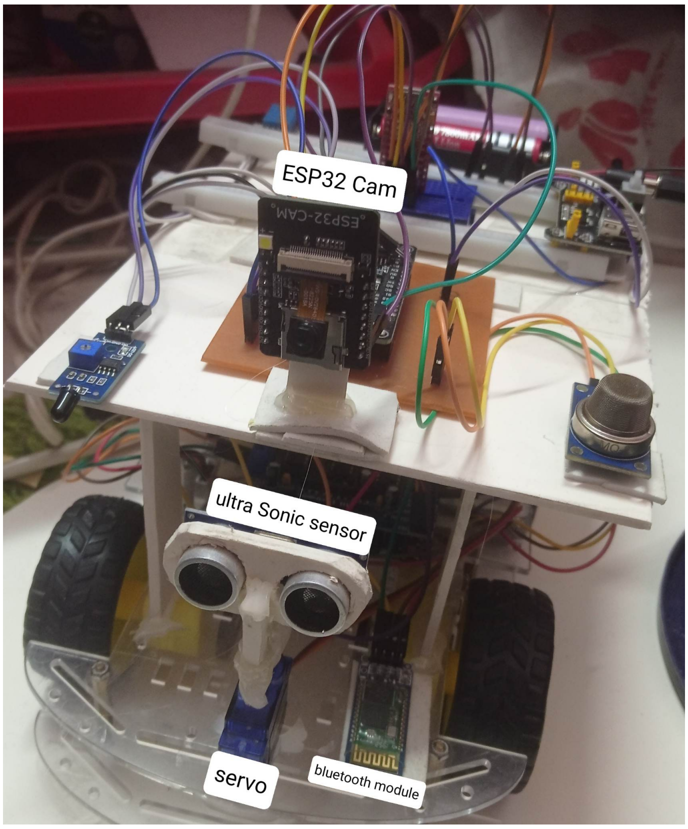

Figure 4.2: front side of the robot

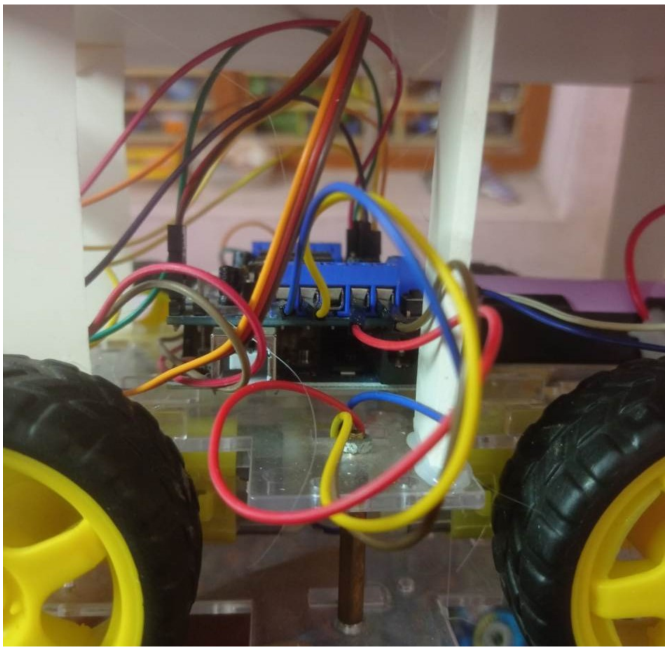

Figure 4.3: Connection of motor driver and arduino along with motors

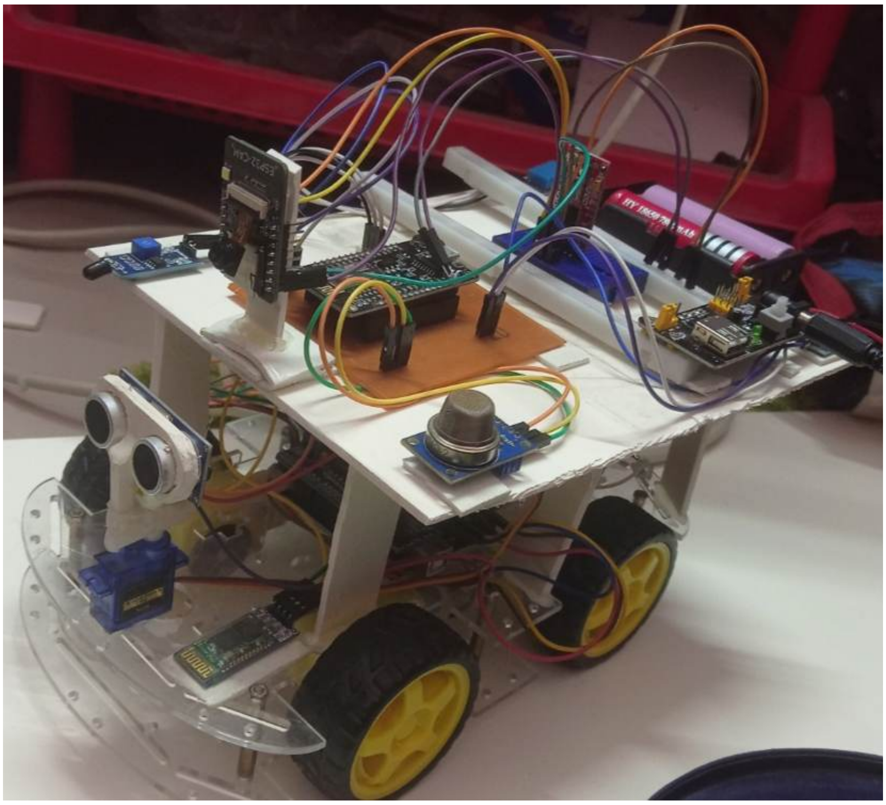

Figure 4.4: Front View of The Robot

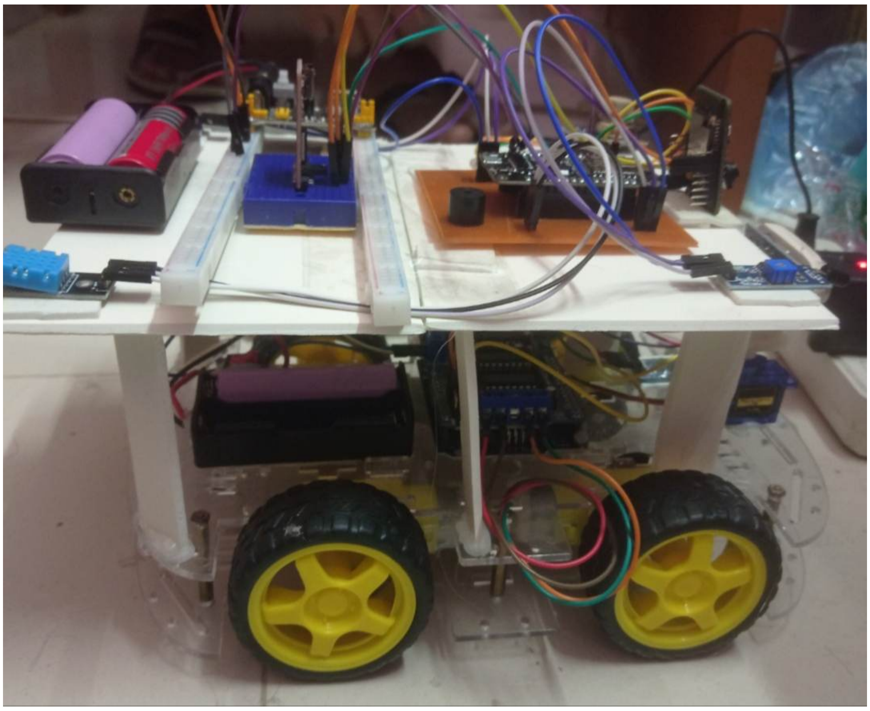

Figure 4.5: Side View of The Robot

4.2 Software Part

The environmental situation will be monitored through Blynk app and web dashboard. In the blynk dashboard [9] a template is created. three gauges are taken and datastream is set up. These datastreams are used in the ESP8266 code to send the sensor values to Blynk. The Blynk view is represented here:

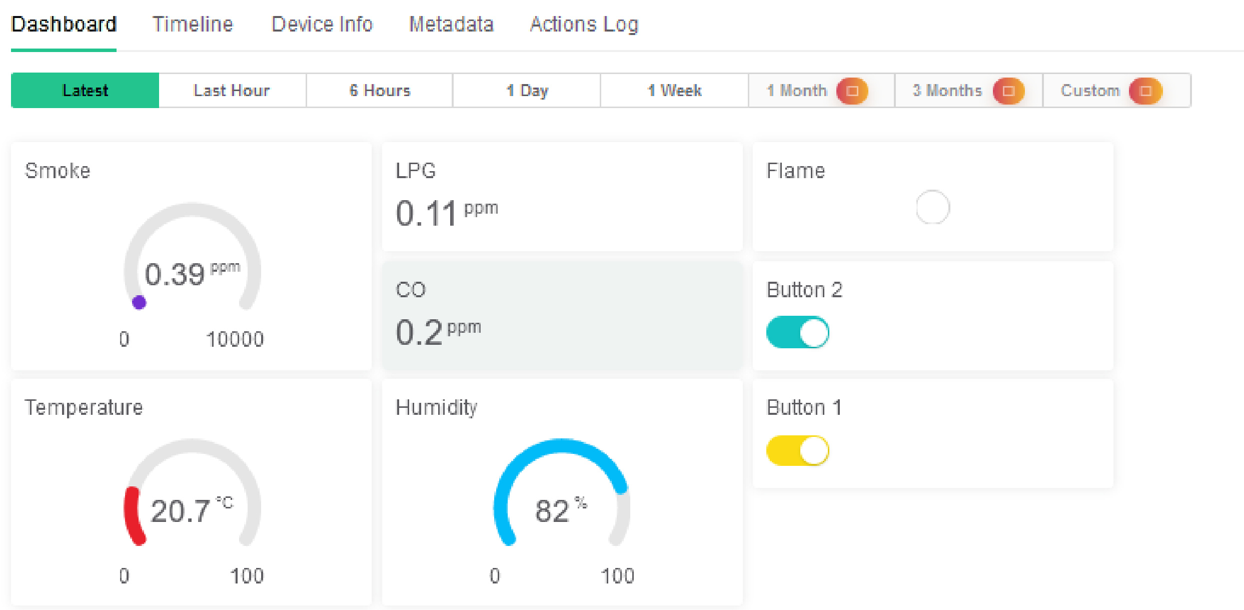

Figure 4.6: Environment monitoring in web dashboard of Blynk.

This figure represents a web dasdboard which is going to help monitoring the overall situation of the particular environment. Real time data will be presented through it.

There is also blynk app on mobile which is going to help in monitoring the situation in android and IOS devices. The representation given below:

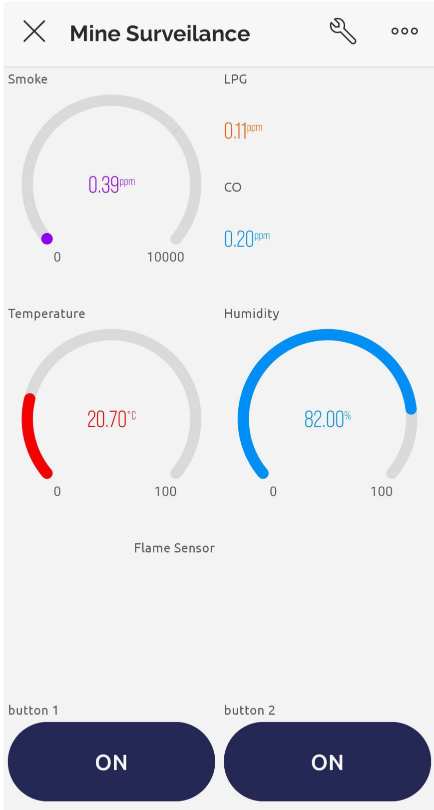

Figure 4.7: Environmental monitoring in Blynk app.

As soon as there will be any occurance which is not common and can be life threatening the blynk will notify through Email and notifications on devices. The Representation is shown below:

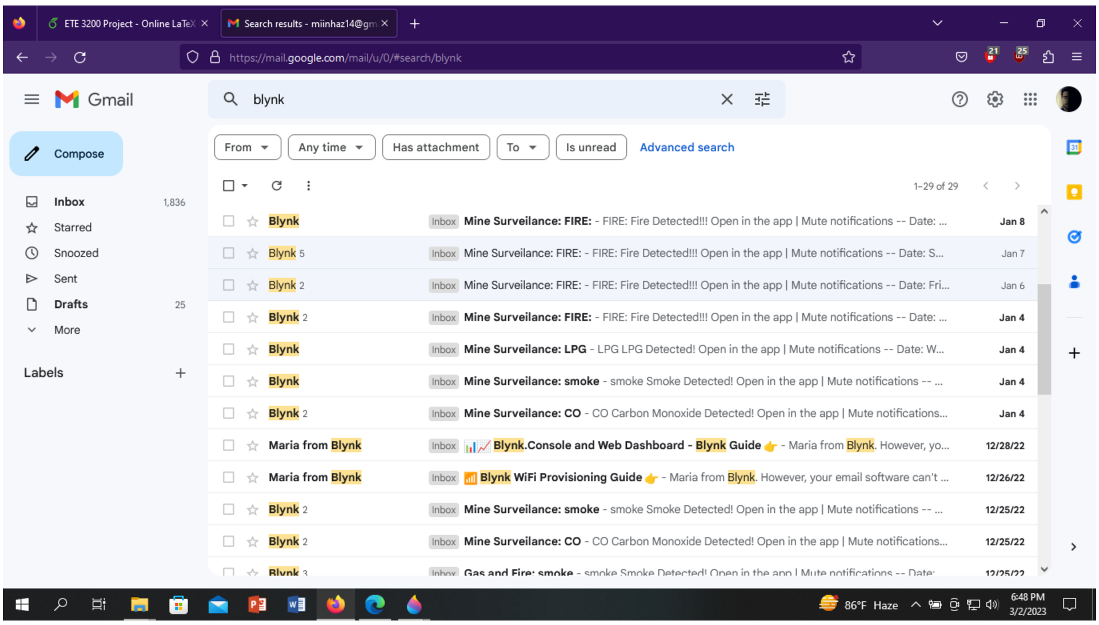

Figure 4.8: Notification through Gmail

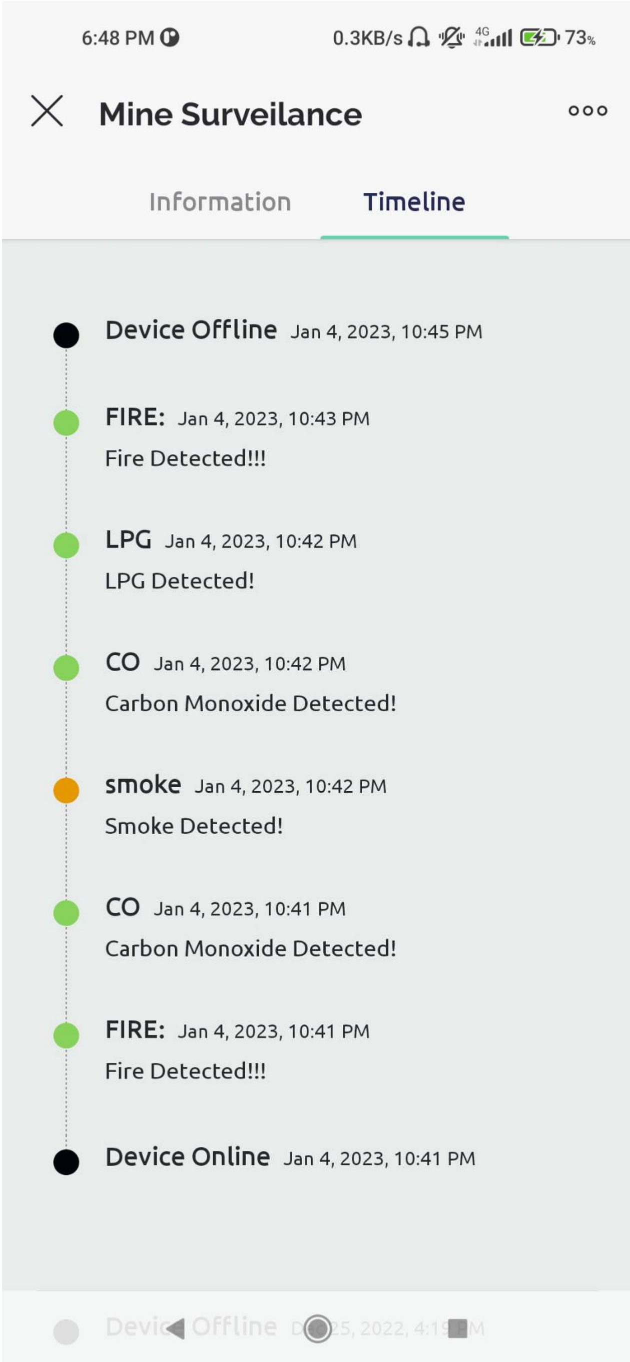

Figure 4.9: Notification through Blynk App

For the surveillance part a particular id is going to be generated. By using that one can see whats going on at that place in real time. Means the Monitoring is possible. speed and light adjustment is possible which is shown below:

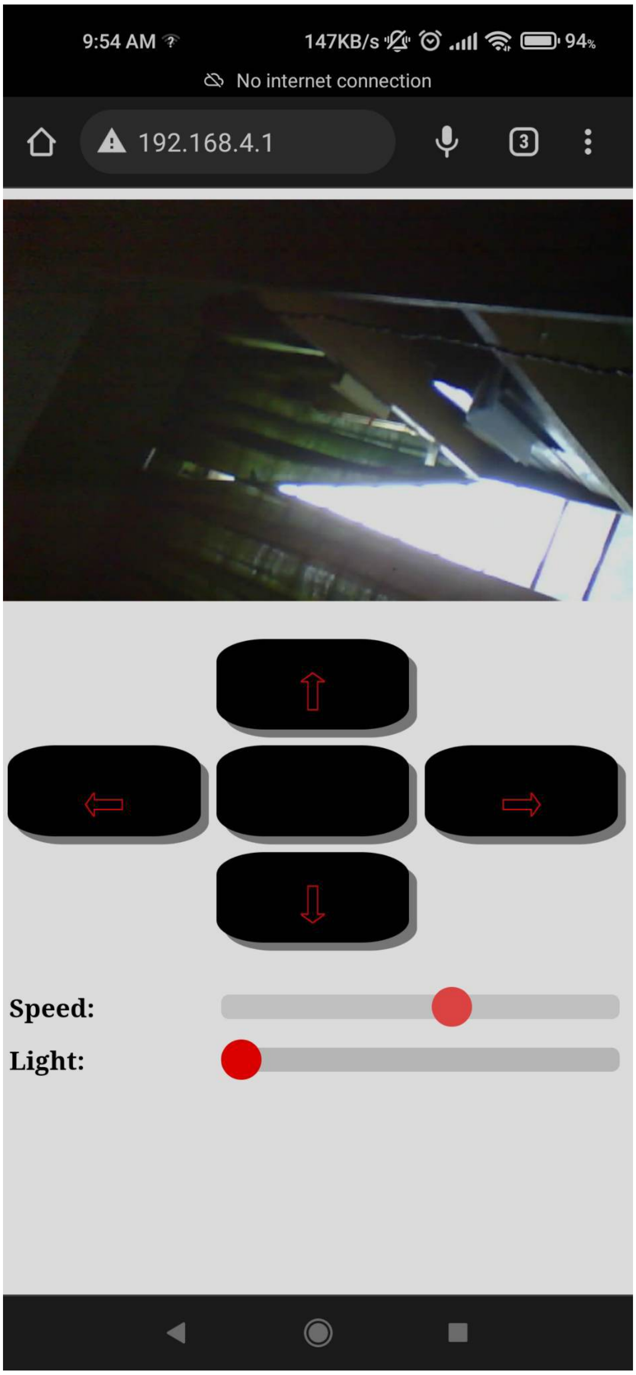

Figure 4.10: Surveillance through ESP CAM

The full project is linked in GITHUB [10].

Chapter 5

Future Works and Development

There is always room for improvement, and the robot can be modified and developed to be more advanced in the future. Here are some potential ways the robot can be upgraded:
1) The integration of AI can make the robot more intelligent and capable of making autonomous decisions. This would improve the robot's ability to detect potential hazards and security threats, respond to emergencies and carry out tasks more efficiently.
2) The robot's mobility can be improved by equipping it with more advanced sensors and actuators, enabling it to navigate more challenging terrains and work in more extreme conditions.
3) The robot's ability to collect and analyze data in real-time can be improved by incorporating more advanced algorithms and data processing techniques. This would enable the robot to provide more detailed and accurate information to operators, leading to improved decision-making and increased safety.
4) The addition of remote sensing technology such as drones or satellites could enhance the robot's capabilities by providing a wider scope of data collection and analysis. This would enable the robot to monitor and detect hazards and security threats over a larger area, improving the overall safety of mining operations.
5) Developing more efficient power sources, such as solar or hydrogen fuel cells, could increase the robot's operating time and reduce its environmental impact.
6) The robot's communication capabilities can be improved to allow for faster and more efficient communication with operators and other robots. This would enable faster response times in emergencies and improve overall efficiency.
7) The robot can be made more durable and rugged to withstand harsh environments and

conditions. This would reduce maintenance costs and increase the robot's lifespan.
8) The use of predictive maintenance algorithms can improve the robot's reliability by detecting potential problems before they occur. This would reduce downtime and maintenance costs.
9) Swarm technology involves the coordination of multiple robots to work together towards a common goal. Implementing swarm technology in the mining industry could lead to more efficient and effective mining operations.

Such modifications can be done in future to make the robot much more and efficient for future. Even though these further development may take a long time to be fruitful but by taking one in particular and then following one by one everything can be achievable.

Chapter 6

CONCLUSION

In conclusion, the IoT-based mine security and surveillance robot represents a significant technological advancement in the mining industry. Its ability to integrate real-time data collection and analysis through IoT technology enables the robot to detect potential hazards and respond to security threats more efficiently, ultimately enhancing safety and security in mining operations. The robot's capabilities to patrol and monitor remote areas and reduce the risk to human personnel are critical features that make it a reliable tool in the industry. The implementation of the IoT-based mine security and surveillance robot provides an opportunity to improve the safety of mining operations significantly. By reducing the risk to human personnel, it can help companies avoid costly accidents and prevent environmental hazards. Additionally, the remote control feature of the robot provides ease of operation and flexibility, enabling operators to oversee and monitor operations from a safe distance. This innovation could potentially revolutionize the mining industry's approach to safety and security, leading to more efficient and effective processes. As technology continues to advance, we can expect more innovative solutions like the IoT-based mine security and surveillance robot to be developed, implemented, and improved upon. The invention stands as an example of how technology can positively impact safety and efficiency in various industries.

So, the IoT-based mine security and surveillance robot has the potential to transform the mining industry's approach to safety and security. Its implementation can lead to more efficient and effective mining processes, improved worker safety, and reduced environmental hazards. As such, the mining industry can take advantage of this innovation and work towards a safer and more sustainable future.

Bibliography

[1] “Internet of things,” https://www.oracle.com/internet-of-things/what-is-iot/, 2023, Accessed on February 2023.

[2] “Dht11 sensor,” https://components101.com/sensors/dht11-temperature-sensor, 2023, Accessed on February 2023.

[3] “Mq-2 sensor,” https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/, 2023, Accessed on February 2023.

[4] “Flame sensor,” https://www.elprocus.com/flame-sensor-working-and-its-applications/, 2023, Accessed on February 2023.

[5] “Esp32 cam,” https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/, 2023, Accessed on February 2023.

[6] “Nodemcu,” https://www.make-it.ca/nodemcu-details-specifications/, 2023, Accessed on February 2023.

[7] “L293d,” https://www.etechnophiles.com/l293d-motor-driver-module-ic-datasheet-pinouts/, 2023, Accessed on February 2023.

[8] “Arduino uno,” https://store.arduino.cc/products/arduino-uno-rev3, note = Accessed on February 2023, 2023.

[9] “Blynk platform,” https://blynk.cloud/dashboard/225014/global/filter/1830591, 2023, Accessed on February 2023.

[10] “Github project link,” https://github.com/minhaz/iot-based-mine-security-surveillance-robot, 2023, Accessed on March 2023.

Appendix

Code for voice and remote controlled robot:

```
/*Bluetooth control, voice control robot car.*/
include < Servo.h >
include < AFMotor.h >
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()
Serial.begin(9600);
pinMode(Trig, OUTPUT);
pinMode(Echo, INPUT);
```

```
servo.attach(motor);
M1.setSpeed(Speed);
M2.setSpeed(Speed);
M3.setSpeed(Speed);
M4.setSpeed(Speed);
void loop()
Bluetoothcontrol();
//voicecontrol();
void Bluetoothcontrol()
if (Serial.available() ¿ 0)
value = Serial.read();
Serial.println(value);
if (value == ’F’)
forward();
else if (value == ’B’)
backward();
else if (value == ’R’)
left();
else if (value == ’L’)
right();
else if (value == ’S’)
Stop();
void voicecontrol()
if (Serial.available() ¿ 0)
value = Serial.read();
Serial.println(value);
if(value ==′)
forward();
else if (value == ’-’)
backward();
else if (value == ’¡’)
L = leftsee();
```

```
servo.write(spoint);
if (L ¿= 10 )
left();
delay(500);
Stop();
else if (L ¡ 10)
Stop();
else if (value == '¿')
R = rightsee();
servo.write(spoint);
if (R ¿= 10 )
right();
delay(500);
Stop();
else if (R ¡ 10)
Stop();
else if (value == '*')
Stop();
// Ultrasonic sensor distance reading function
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 forward()
M1.run(FORWARD);
M2.run(FORWARD);
M3.run(FORWARD);
```

```
M4.run(FORWARD);
void backward()
M1.run(BACKWARD);
M2.run(BACKWARD);
M3.run(BACKWARD);
M4.run(BACKWARD);
void right()
M1.run(BACKWARD);
M2.run(BACKWARD);
M3.run(FORWARD);
M4.run(FORWARD);
void left()
M1.run(FORWARD);
M2.run(FORWARD);
M3.run(BACKWARD);
M4.run(BACKWARD);
void Stop()
M1.run(RELEASE);
M2.run(RELEASE);
M3.run(RELEASE);
M4.run(RELEASE);
int rightsee()
servo.write(20);
delay(500);
Left = ultrasonic();
return Left;
int leftsee()
servo.write(180);
delay(500);
Right = ultrasonic();
return Right;
```

Code for Environment monitoring using nodeMCU:

```
define BLYNK TEMPLATE ID ”TMPLeTRBaqfd”
define BLYNK DEVICE NAME ”Gas and Fire”
define BLYNK AUTH TOKEN ”MoraUODkWK2IkOGMHj8Snj − J3pHAcPSR”
include ”DHT.h”
define BLYNK PRINT Serial
define APP DEBUG
define USE NODE MCU BOARD
define DHTTYPE DHT11 // DHT11 sensor
define DHTPIN 0 // Digital input pin connected to the DHT11 sensor
DHT dht(DHTPIN, DHTTYPE);
include < ESP8266WiFi.h >
include < BlynkSimpleEsp8266.h >
include ”Blynk.h”
include < MQ2.h >
include < SPI.h >
include < Wire.h >
char auth[] = ”MoraUODkWK2IkOGMHj8Snj − J3pHAcPSR”;
char ssid[] = ”Alpha”;
char pass[] = ”180403838”;
define BUZZ 12 //D6
define LED 14 //D5
int flag=0;
int pin = A0;
float lpg, co, smoke;
MQ2 mq2(pin);
int button1 = 0;
int button2 = 0;
SimpleTimer timer;
void setup()
Serial.begin(115200);
delay(100);
```

```
Blynk.begin(auth, ssid, pass);
pinMode(BUZZ, OUTPUT); pinMode(LED, OUTPUT);
digitalWrite(BUZZ, LOW);
digitalWrite(LED, LOW);
dht.begin();
mq2.begin();
timer.setInterval(1000L, sendSensorData);
void loop()
timer.run(); // Initiates SimpleTimer
Blynk.run();
void sendSensorData()
float* values = mq2.read(true);
co = mq2.readCO();
smoke = mq2.readSmoke();
lpg = mq2.readLPG();
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
Blynk.virtualWrite(V6, temperature);
Blynk.virtualWrite(V7, humidity);
if (temperature ¿ 50 )
Blynk.logEvent(”t”, ”High Temperature Alert!!!!”);
Serial.print(”Temperature: ”);
Serial.print(temperature);
Serial.println(”°C”);
Serial.print(”Humidity: ”);
Serial.print(humidity);
Serial.println(”
int isButtonPressed = digitalRead(D1); // falme sensor connected D1 pin
if (isButtonPressed == 1andflag == 0)
flag=1;
Blynk.virtualWrite(V0,0);
digitalWrite(BUZZ, LOW);
```

```
else if (isButtonPressed==0)
flag=0;
Serial.println(”Fire DETECTED”);
digitalWrite(BUZZ, HIGH);
Blynk.virtualWrite(V0,255);
Blynk.logEvent(”fire”, ”Fire Detected!!!”);
Blynk.virtualWrite(V1, smoke);
Blynk.virtualWrite(V2, lpg);
Blynk.virtualWrite(V3, co);
if (smoke ¿ 50 )
Blynk.logEvent(”smoke”, ”Smoke Detected!”);
digitalWrite(BUZZ, HIGH);
digitalWrite(LED, HIGH);
else
digitalWrite(BUZZ, LOW);
if (lpg ¿ 50 )
Blynk.logEvent(”lpg”, ”LPG Detected!”);
if (co ¿ 50 )
Blynk.logEvent(”co”, ”Carbon Monoxide Detected!”);
// in Blynk app writes values to the Virtual Pin 4
BLYNK WRITE(V4)
button1 = param.asInt(); // assigning incoming value from pin V4 to a variable
// in Blynk app writes values to the Virtual Pin 5
BLYNK WRITE(V5)
button2 = param.asInt(); // assigning incoming value from pin V5 to a variable
```

Code for ESP32 CAM for surveillance purpose:

include ”$esp-camera.h$”

include $< WiFi.h >$

// http://192.168.251.22/

define CAMERA MODEL AI THINKER // Has PSRAM

include ”camera pins.h”

// ===========================

// Enter your WiFi credentials

// ===========================

const char* ssid = ”$Alpha$”;

const char* password = ”180403838”;

void startCameraServer();

void setupLedFlash(int pin);

void setup()

Serial.begin(115200);

Serial.setDebugOutput(true);

Serial.println();

camera config t config;

config.ledc channel = LEDC CHANNEL 0;

config.ledc timer = LEDC TIMER 0;

config.pin d0 = Y2 GPIO NUM;

config.pin d1 = Y3 GPIO NUM;

config.pin d2 = Y4 GPIO NUM;

config.pin d3 = Y5 GPIO NUM;

config.pin d4 = Y6 GPIO NUM;

config.pin d5 = Y7 GPIO NUM;

config.pin d6 = Y8 GPIO NUM;

config.pin d7 = Y9 GPIO NUM;

config.pin xclk = XCLK GPIO NUM;

config.pin pclk = PCLK GPIO NUM;

config.pin vsync = VSYNC GPIO NUM;

config.pin href = HREF GPIO NUM;

```
config.pin sscb sda = SIOD GPIO NUM;
config.pin sscb scl = SIOC GPIO NUM;
config.pin pwdn = PWDN GPIO NUM;
config.pin reset = RESET GPIO NUM;
config.xclk freq hz = 20000000;
config.frame size = FRAMESIZE UXGA;
config.pixel format = PIXFORMAT JPEG; // for streaming
//config.pixel format = PIXFORMAT RGB565; // for face detection/recognition
config.grab mode = CAMERA GRAB WHEN EMPTY;
config.fb location = CAMERA FB IN PSRAM;
config.jpeg quality = 12;
config.fb count = 1;
// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
// for larger pre-allocated frame buffer.
if(config.pixelformat == PIXFORMATJPEG)
if(psramFound())
config.jpeg quality = 10;
config.fb count = 2;
config.grab mode = CAMERA GRAB LATEST;
else
// Limit the frame size when PSRAM is not available
config.frame size = FRAMESIZE SVGA;
config.fb location = CAMERA FB IN DRAM;
else
// Best option for face detection/recognition
config.frame size = FRAMESIZE 240X240;
if CONFIG IDF TARGET ESP32S3
config.fb count = 2;
endif
if defined(CAMERA MODEL ESP EYE)
pinMode(13, INPUT PULLUP);
pinMode(14, INPUT PULLUP);
```

endif

// camera init

esp err t err = esp camera init($config$);

if $(err! = ESP.OK)$

Serial.printf("Camera init failed with error 0x

return;

sensor t * s = esp camera sensor get();

// initial sensors are flipped vertically and colors are a bit saturated

if $(s-> id.PID == OV3660_PID)$

$s-> set_v flip(s,1)$;// flip it back

$s-> set_b rightness(s,1)$;// up the brightness just a bit

$s-> set_s aturation(s,-2)$; // lower the saturation

// drop down frame size for higher initial frame rate

if$(config.pixelformat == PIXFORMATJPEG)$

$s-> set_f ramesize(s,FRAMESIZEQVGA)$;

if $defined(CAMERAMODELM5STACKWIDE)$

$||defined(CAMERAMODELM5STACKESP32CAM)$

$s-> setvflip(s,1)$;

$s-> set_h mirror(s,1)$;

endif

if defined(CAMERA MODEL ESP32S3 EYE)

$s-> set_v flip(s,1)$;

endif

// Setup LED FLash if LED pin is defined in camera pins.h

if defined(LED GPIO NUM)

setupLedFlash(LED GPIO NUM);

endif

WiFi.begin(ssid, password);

WiFi.setSleep(false);

while (WiFi.status() != WL CONNECTED)

delay(500);

Serial.print(".");

```
Serial.println("");
Serial.println("WiFi connected");
startCameraServer();
Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
void loop()
// Do nothing. Everything is done in another task by the web server
delay(10000);
```