

Project Documentation

ELECTRONIC SYSTEM DESIGN

-PROF(ALOK MITTAL)

PROJECT-SEARCH ASSESS AND RESCUE ROBOT(group-12)

SUPERVISOR:

ALOK MITTAL SIR

Members:

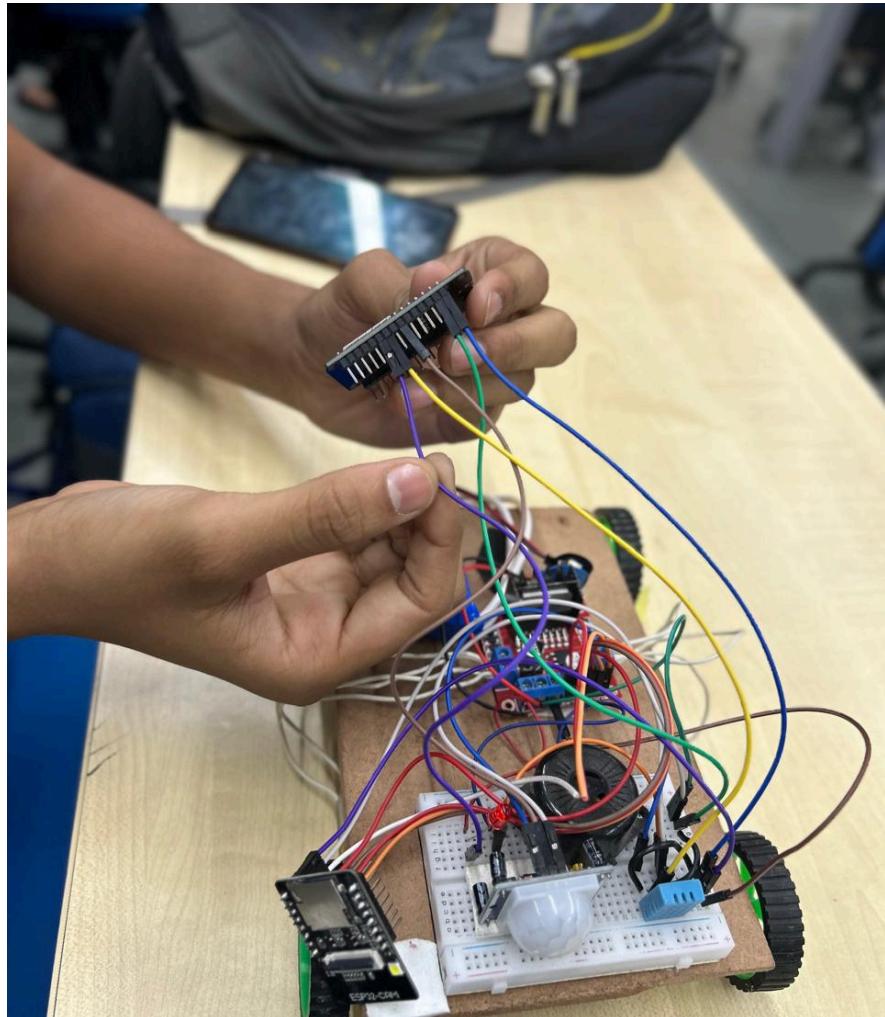
- MAYANK SINGH(2022287)
- HARIOM TIWARI(2022194)
- HARSH(2022195)
- HARSHIT SAGAR(2022210)
- HIMKESH(2022218)

Budget:

Around 1500 rupees

Timeline:

Mar 10, 2024 to May 15, 2024



Project Details

PROJECT OVERVIEW

- **INTRO-** The Search, Assess, and Rescue Robot (SARR) .The SARR project endeavors to address this need by conceptualizing and realizing a robotic system proficient In the face of escalating natural disasters, industrial accidents, and humanitarian crises, the development of a versatile and efficient robotic solution becomes imperative /
 - **OBJECTIVE-**The SARR project endeavors to address this need by conceptualizing and realizing a robotic system proficient in navigating hazardous environments, detecting human presence, and executing swift and effective rescue operations.
 - **SCOPE- 1) Sensors-**Passive Infrared (PIR) sensors for detecting human presence and Infrared (IR) sensors for environmental assessment.
- 2) **Navigation** - through the use of camera we can assess them in confined spaces

3)Communication-through wireless communication protocols, the robot can transmit vital data, including sensor readings and live video feeds, to aid in coordination and decision-making during rescue operations.

4) RESCUE OPERATIONS-The primary focus of the SARR project is to enable the robot to perform essential rescue tasks in hazardous environments. This includes assessing the condition of trapped individuals, delivering medical supplies, providing communication channels for survivors, and assisting in evacuation procedures

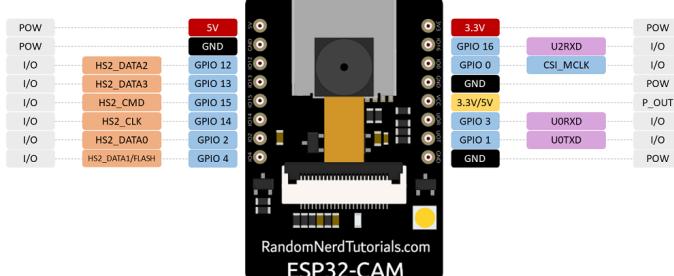
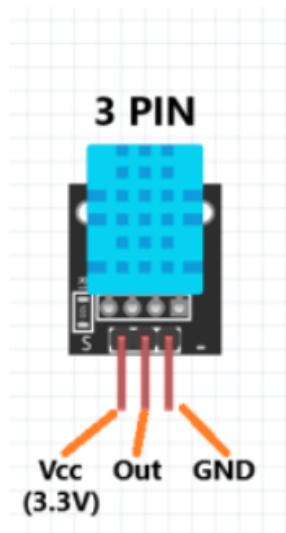
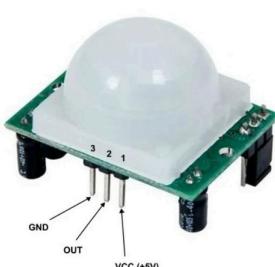
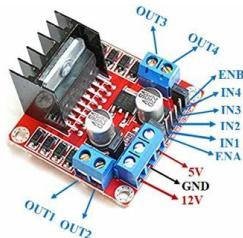
METHODOLOGY

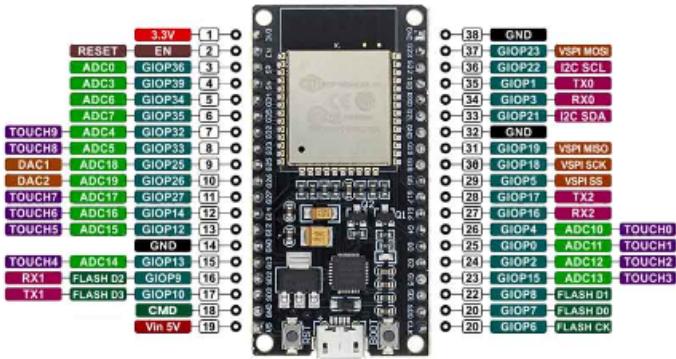
1. **Selection of Platform:** The project begins with the selection of an appropriate microcontroller platform to serve as the brain of the robot. The ESP microcontroller, renowned for its versatility and connectivity capabilities, is chosen as the foundation for the SARR's control system . It has inbuilt bluetooth(CAM-ESP) module for wireless communication .
2. **Rover Assembly:** The initial phase focuses on assembling the rover chassis and integrating motor control mechanisms. A dual-channel motor driver module is utilized to power the motors, enabling the robot to move in a controlled manner.
3. **Sensor Integration:** Sensor integration plays a pivotal role in enhancing the SARR's perception and environmental awareness capabilities. Various sensors, including Passive Infrared (PIR) sensors for human detection, humidity sensors, and pressure sensors, are strategically incorporated into the robot's architecture. These sensors provide real-time data on environmental conditions and potential hazards, empowering the robot to make informed decisions during operation.
4. **Camera Integration:** Visual perception is augmented through the integration of a camera module, enabling the SARR to capture live video feeds of its surroundings. The camera module is interfaced with the ESP microcontroller, facilitating real-time image processing and analysis.This visual feedback enhances situational awareness and aids in object detection and navigation.
5. **Wireless Communication:** The ESP microcontroller establishes wireless connections with external devices, such as smartphones or remote control stations, enabling remote monitoring and control of the robot. Data transmission protocols are optimized for reliability and low latency, ensuring timely delivery of critical information.
6. **Software Development:** The software layer of the SARR encompasses firmware development, sensor data processing algorithms, and user interface design in the audrino IDE. Custom software routines are implemented to handle sensor data acquisition, motor control, and communication protocols. Emphasis is placed on optimizing code efficiency and modularity to facilitate future enhancements and scalability.
7. **Testing and Iteration:** Rigorous testing procedures are employed to validate the functionality and performance of the SARR in simulated and real-world scenarios. Test cases encompass various operational scenarios, including obstacle navigation, human

detection, and communication range assessment. Feedback from testing informs iterative improvements and refinements to the robot's design and functionality.

System Architecture

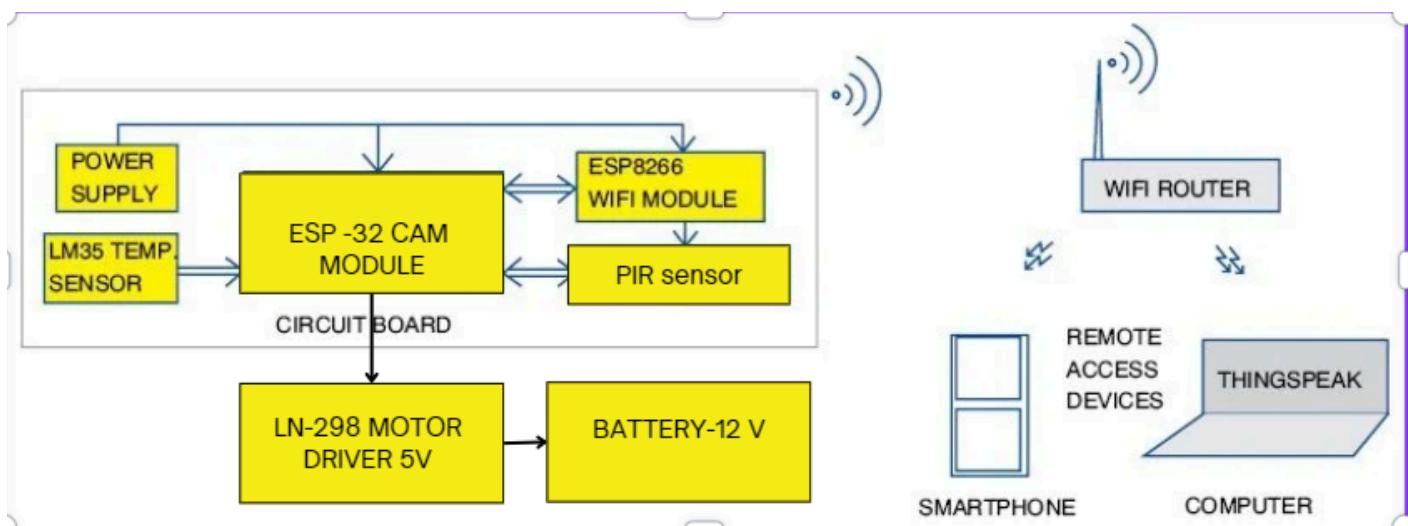
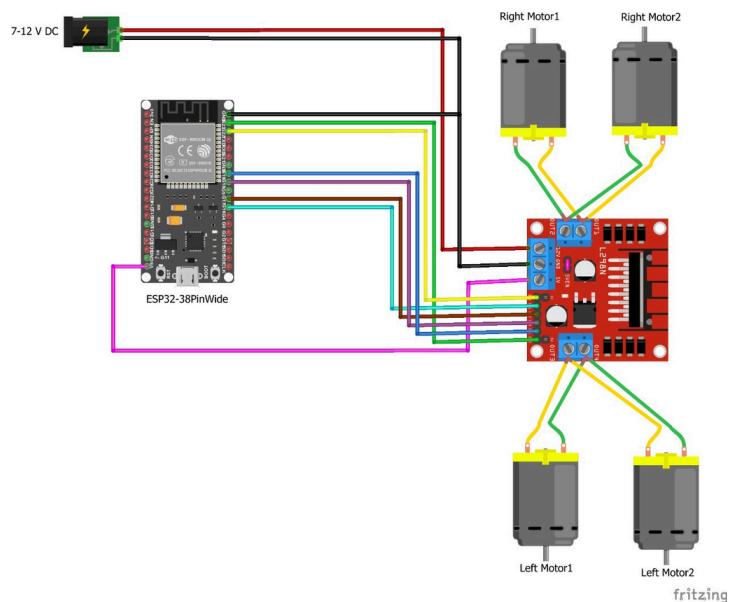
1. **Components:** All the major components of the SARR, including sensors(PIR Sensors,humidity and temperature senors(DHT-11), microcontroller(ESP- WROOM 32,ESP CAM MODULE), Wifi and bluetooth, LN 298 motor driver module, battery-12 V ,motors





1. **Interaction:** motors are driving through the battery which is connected through motor driver . Our main components ESP-32 and ESP cam module are also integrated with it our peripherals like senors are also connected to the esp module.

2. BLOCK DIAGRAM:



3) Interfaces

1. Sensor Interfaces:

- **Sensor to Microcontroller:** Data interface between sensors (such as PIR sensors, humidity sensors, pressure sensors) and the microcontroller (e.g., ESP32). This interface transmits environmental data such as temperature, humidity, pressure, and human presence detection to the microcontroller for processing.
- **Camera to Microcontroller:** Interface for transmitting video data captured by the camera module to the microcontroller. This interface allows the microcontroller to process visual information for object detection, navigation, and situational awareness.

2. Actuator Interfaces:

- **Microcontroller to Motor Driver:** Command interface between the microcontroller and the motor driver module responsible for controlling the movement of the robot's motors (e.g., DC motors). This interface sends motor control commands (e.g., speed, direction) from the microcontroller to the motor driver for execution.

3. Communication Interfaces:

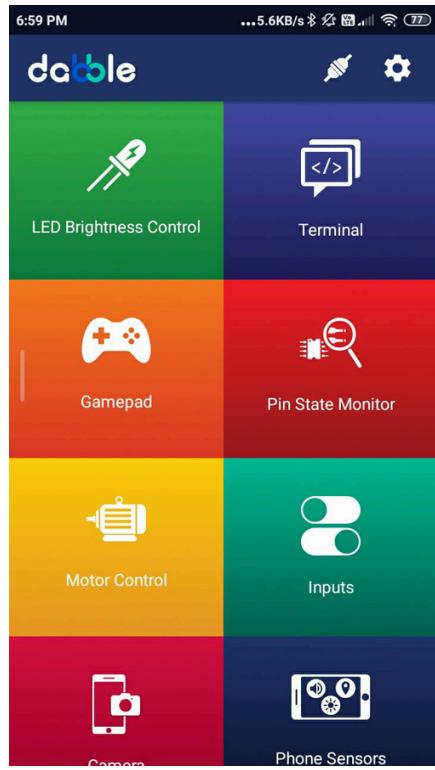
- **CAM Microcontroller to Wi-Fi Module:** Interface for establishing a wireless connection between the microcontroller and the Wi-Fi module (e.g., ESP8266 or ESP32). This interface enables the robot to communicate with external devices, such as smartphones or control stations, over a Wi-Fi network.
- **Microcontroller to Bluetooth Module:** Interface for enabling Bluetooth communication between the microcontroller and external devices (e.g., smartphones, Bluetooth-enabled controllers). This interface facilitates remote control and monitoring of the robot via Bluetooth connectivity.

4. Power Interfaces:

- **Power Distribution Interface:** Interface for distributing power from the main power source (e.g., battery pack or external power supply) to the various subsystems and components of the robot. This interface ensures adequate power supply to all components for uninterrupted operation.
- **Charging Interface:** Interface for charging the robot's battery pack from an external power source. This interface allows the battery pack to be recharged when depleted, ensuring prolonged operational capability of the robot.

5. Control Interfaces:

- **User Interface:** Interface for user interaction with the robot, such as a smartphone app or a control panel. This interface allows users to send commands, receive status updates, and monitor the robot's operation remotely.



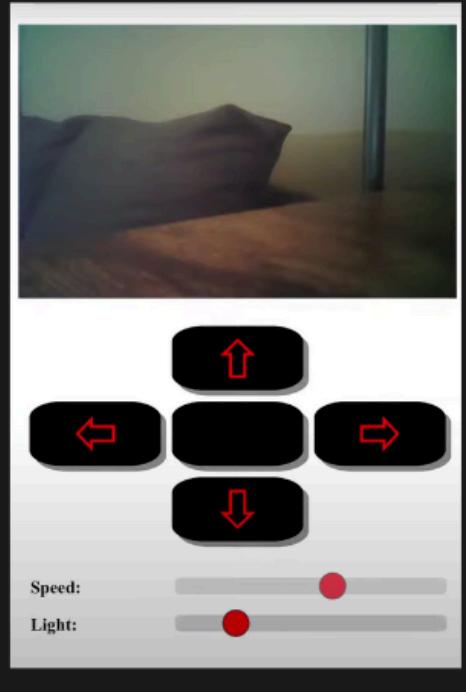
IMPLEMENTATION DETAILS

1. Rover Assembly and Motor Control:

- Dual-channel motor driver modules are utilized to control the movement of DC motors attached to the wheels.
- The microcontroller (ESP32) sends motor control commands to the motor driver, enabling precise control over the robot's movement speed and direction.

Rover Controller

You can control the rover's movement, speed, and flash intensity through the ESP cam module using the terminal. The terminal also provides live video feed.

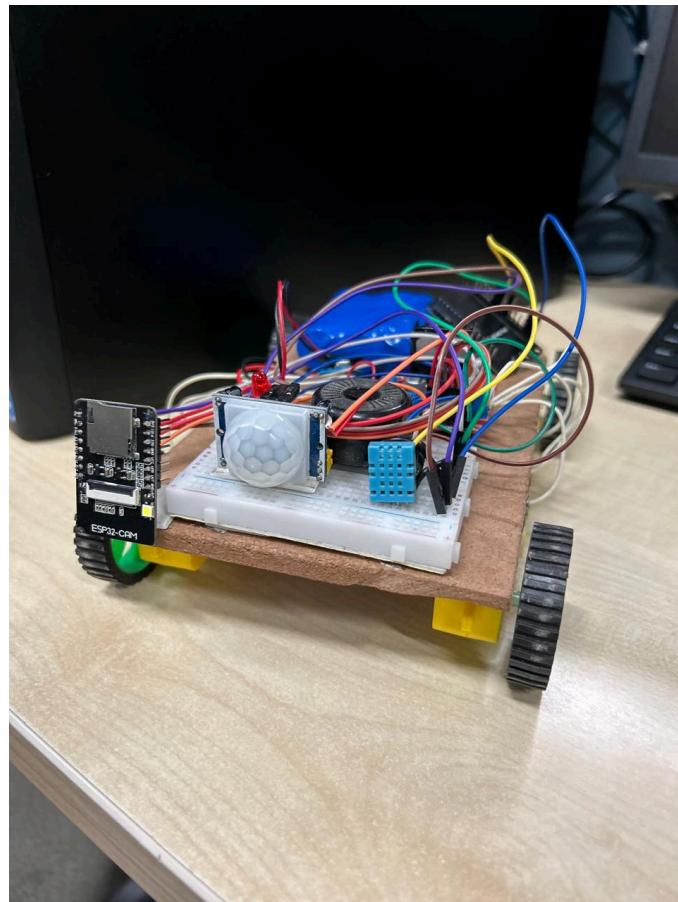


1. Sensor Integration:

- Passive Infrared (PIR) sensors are strategically placed to detect human presence within the robot's vicinity.
- Additional sensors, such as humidity and pressure sensors, are integrated to monitor environmental conditions.
- Sensor data is collected and processed by the microcontroller, providing real-time insights into the robot's surroundings.

2. Camera Integration:

- A camera module is mounted on the robot to capture live video feeds of its surroundings.
- The camera interface is established with the microcontroller, allowing it to capture and process visual information.



-

1. Wireless Communication:

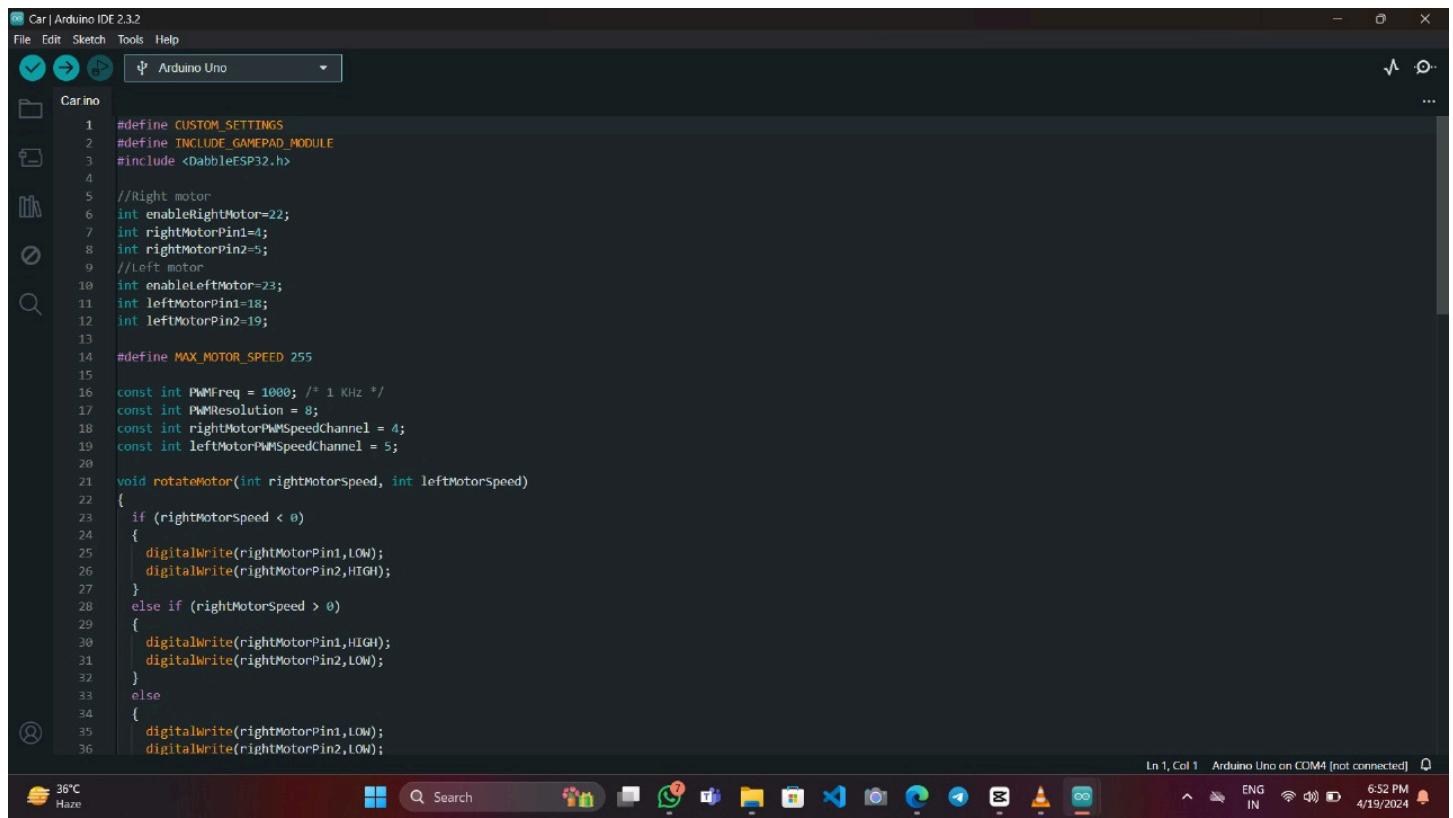
- The ESP32 microcontroller is equipped with built-in Wi-Fi and Bluetooth capabilities for wireless communication.
- Wi-Fi communication enables the robot to transmit sensor data and video feeds to external devices, such as smartphones or control stations.
- Bluetooth communication facilitates remote control and monitoring of the robot via a smartphone app or a Bluetooth-enabled controller.

2. Decision-Making Algorithms:

- Decision-making algorithms analyze environmental data to identify potential hazards, assess the situation, and determine appropriate actions.
- For this purpose we have implemented a sound system which can tell the detection

3. Coding

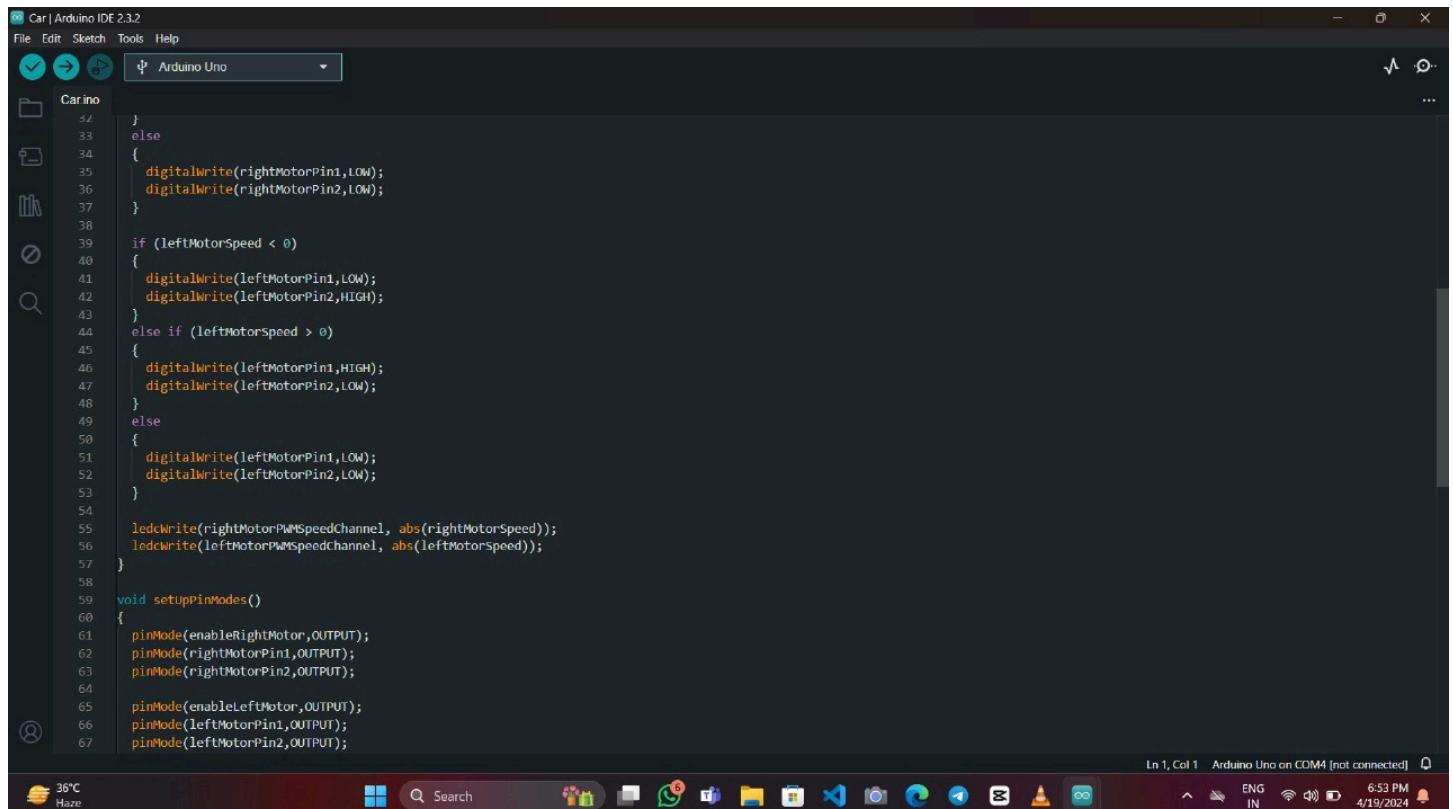
- we have used audrino IDE for programming
- All the useful librariers has been imported.
- cam module has been programmed externally as it has no ports so we have imported code into it for the use cases.
- code for the rover movement .--



```
Car | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
Carino
1 #define CUSTOM_SETTINGS
2 #define INCLUDE_GAMEPAD_MODULE
3 #include <DabbleESP32.h>
4
5 //Right motor
6 int enableRightMotor=22;
7 int rightMotorPin1=4;
8 int rightMotorPin2=5;
9 //Left motor
10 int enableLeftMotor=23;
11 int leftMotorPin1=18;
12 int leftMotorPin2=19;
13
14 #define MAX_MOTOR_SPEED 255
15
16 const int PWMFreq = 1000; /* 1 KHz */
17 const int PWMResolution = 8;
18 const int rightMotorPWMSpeedChannel = 4;
19 const int leftMotorPWMSpeedChannel = 5;
20
21 void rotateMotor(int rightMotorspeed, int leftMotorspeed)
22 {
23     if (rightMotorspeed < 0)
24     {
25         digitalWrite(rightMotorPin1,LOW);
26         digitalWrite(rightMotorPin2,HIGH);
27     }
28     else if (rightMotorspeed > 0)
29     {
30         digitalWrite(rightMotorPin1,HIGH);
31         digitalWrite(rightMotorPin2,LOW);
32     }
33     else
34     {
35         digitalWrite(rightMotorPin1,LOW);
36         digitalWrite(rightMotorPin2,LOW);
37     }
38
39     if (leftMotorspeed < 0)
40     {
41         digitalWrite(leftMotorPin1,LOW);
42         digitalWrite(leftMotorPin2,HIGH);
43     }
44     else if (leftMotorspeed > 0)
45     {
46         digitalWrite(leftMotorPin1,HIGH);
47         digitalWrite(leftMotorPin2,LOW);
48     }
49     else
50     {
51         digitalWrite(leftMotorPin1,LOW);
52         digitalWrite(leftMotorPin2,LOW);
53     }
54
55     ledcWrite(rightMotorPWMSpeedChannel, abs(rightMotorspeed));
56     ledcWrite(leftMotorPWMSpeedChannel, abs(leftMotorspeed));
57 }
58
59 void setUpPinModes()
60 {
61     pinMode(enableRightMotor,OUTPUT);
62     pinMode(rightMotorPin1,OUTPUT);
63     pinMode(rightMotorPin2,OUTPUT);
64
65     pinMode(enableLeftMotor,OUTPUT);
66     pinMode(leftMotorPin1,OUTPUT);
67     pinMode(leftMotorPin2,OUTPUT);
}

```

Ln 1, Col 1 Arduino Uno on COM4 [not connected] 6:52 PM 4/19/2024



```
Car | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
Carino
59 void setUpPinModes()
60 {
61     pinMode(enableRightMotor,OUTPUT);
62     pinMode(rightMotorPin1,OUTPUT);
63     pinMode(rightMotorPin2,OUTPUT);
64
65     pinMode(enableLeftMotor,OUTPUT);
66     pinMode(leftMotorPin1,OUTPUT);
67     pinMode(leftMotorPin2,OUTPUT);
}

```

Ln 1, Col 1 Arduino Uno on COM4 [not connected] 6:53 PM 4/19/2024

Code for the pin connection--

sketch_apr3a | Arduino IDE 2.3.2

File Edit Sketch Tools Help

DOIT ESP32 DEVKIT V1

BOARDS MANAGER

Filter your search... Type: All

Arduino AVR Boards by Arduino 1.8.6 installed

Boards included in this package: Arduino Mega or Mega 2560, Arduino Explora, LilyPad... More info

1.8.6 REMOVE

Arduino Mbed OS Edge Boards by Arduino 4.1.3 INSTALL

Boards included in this package: Arduino Edge Control More info

4.1.3 REMOVE

Arduino Mbed OS GIGA Boards by Arduino 4.1.3 INSTALL

Boards included in this package: Arduino Giga More info

4.1.3 REMOVE

Arduino Mbed OS

Output

```
1 #include <DabbleESP32.h>
2 #include <DHT.h>
3
4 #define CUSTOM_SETTINGS
5 #define INCLUDE_GAMEPAD_MODULE
6 #define INCLUDE_TERMINAL_MODULE
7
8 // Pin Definitions for Motors
9 int enableRightMotor = 22;
10 int rightMotorPin1 = 4;
11 int rightMotorPin2 = 5;
12 int enableLeftMotor = 23;
13 int leftMotorPin1 = 18;
14 int leftMotorPin2 = 19;
15
16 // LED Pin
17 int ledPin = 32;
18
19 // PIR Sensor Pin
20 int pirPin = 26;
21
22 // Constants
23 #define MAX_MOTOR_SPEED 255
24 const int PWMFreq = 1000; /* 1 KHz */
25 const int PWMResolution = 8;
26 const int rightMotorPMSpeedchannel = 4;
```

Ln 184, Col 34 DOIT ESP32 DEVKIT V1 on COM3 [not connected] 15:32 15-05-2024 ENG IN

sketch_apr3a | Arduino IDE 2.3.2

File Edit Sketch Tools Help

DOIT ESP32 DEVKIT V1

BOARDS MANAGER

Filter your search... Type: All

Arduino AVR Boards by Arduino 1.8.6 installed

Boards included in this package: Arduino Mega or Mega 2560, Arduino Explora, LilyPad... More info

1.8.6 REMOVE

Arduino Mbed OS Edge Boards by Arduino 4.1.3 INSTALL

Boards included in this package: Arduino Edge Control More info

4.1.3 REMOVE

Arduino Mbed OS GIGA Boards by Arduino 4.1.3 INSTALL

Boards included in this package: Arduino Giga More info

4.1.3 REMOVE

Arduino Mbed OS

Output

```
29 // DHT Sensor Pin
30 #define DHT_PIN 33 // changed to avoid conflict
31 #define DHT_TYPE DHT11
32 DHT dht(DHT_PIN, DHT_TYPE);
33
34 unsigned long lastSensorUpdateTime = 0; // Variable to store the last time sensor data was updated
35 const unsigned long sensorUpdateInterval = 4000; // Update sensor data every 4 seconds
36
37 // Variables for motion debounce
38 const unsigned long debounceDelay = 2000; // Debounce time (in milliseconds)
39 unsigned long lastDebounceTime = 0; // Variable to store last time motion was detected
40
41 bool motionDetected = false; // Variable to track motion state
42
43 void rotateMotor(int rightMotorSpeed, int leftMotorSpeed)
44 {
45     // Control Right Motor
46     if (rightMotorSpeed < 0)
47     {
48         digitalWrite(rightMotorPin1, LOW);
49         digitalWrite(rightMotorPin2, HIGH);
50     }
51     else if (rightMotorSpeed > 0)
52     {
53         digitalWrite(rightMotorPin1, HIGH);
54         digitalWrite(rightMotorPin2, LOW);
55     }
56     else
57     {
58         digitalWrite(rightMotorPin1, LOW);
59         digitalWrite(rightMotorPin2, LOW);
60     }
61 }
```

Ln 184, Col 34 DOIT ESP32 DEVKIT V1 on COM3 [not connected] 15:32 15-05-2024 ENG IN

The screenshot shows the Arduino IDE 2.3.2 interface. The top menu bar includes File, Edit, Sketch, Tools, Help, and a board selection dropdown for DOIT ESP32 DEVKIT V1. The left sidebar features the Boards Manager with sections for Arduino AVR Boards (1.8.6 installed) and Arduino Mbed OS Edge Boards (4.1.3). The main central area displays the sketch code for motor control:

```
58     digitalWrite(rightMotorPin1, LOW);
59     digitalWrite(rightMotorPin2, LOW);
60 }
61
62 // Control Left Motor
63 if (leftMotorSpeed < 0)
64 {
65     digitalWrite(leftMotorPin1, LOW);
66     digitalWrite(leftMotorPin2, HIGH);
67 }
68 else if (leftMotorSpeed > 0)
69 {
70     digitalWrite(leftMotorPin1, HIGH);
71     digitalWrite(leftMotorPin2, LOW);
72 }
73 else
74 {
75     digitalWrite(leftMotorPin1, LOW);
76     digitalWrite(leftMotorPin2, LOW);
77 }
78
79 // Set Motor Speed using PWM
80 ledcWrite(rightMotorPWMspeedChannel, abs(rightMotorSpeed));
81 ledcWrite(leftMotorPWMspeedChannel, abs(leftMotorSpeed));
82
83
84 void setUpPinModes()
85 {
86     // Set up pin modes for motors
87     pinMode(enableRightMotor, OUTPUT);
88     pinMode(rightMotorPin1, OUTPUT);
89     pinMode(rightMotorPin2, OUTPUT);
90     pinMode(enableLeftMotor, OUTPUT);
91 }
```

The status bar at the bottom indicates "Ln 184, Col 34 DOIT ESP32 DEVKIT V1 on COM3 [not connected]" and shows system icons for battery, signal, and date/time.

code for the cam module integration

The screenshot shows the Arduino IDE 2.3.2 interface. The top menu bar includes File, Edit, Sketch, Tools, Help, and a board selection dropdown for DOIT ESP32 DEVKIT V1. The left sidebar features the Boards Manager with sections for Arduino AVR Boards (1.8.6 installed) and Arduino Mbed OS Edge Boards (4.1.3). The main central area displays the sketch code for camera module integration:

```
1 #include "esp_camera.h"
2 #include <Arduino.h>
3 #include <WiFi.h>
4 #include <AsyncTCP.h>
5 #include <ESPAsyncWebServer.h>
6 #include <iostream>
7 #include <sstream>
8
9 struct MOTOR_PINS
10 {
11     int pinEn;
12     int pinIN1;
13     int pinIN2;
14 };
15
16 std::vector<MOTOR_PINS> motorPins =
17 {
18     {12, 13, 15}, //RIGHT_MOTOR Pins (EnA, IN1, IN2)
19     {12, 14, 2}, //LEFT_MOTOR Pins (EnB, IN3, IN4)
20 };
21 #define LIGHT_PIN 4
22
23 #define UP 1
24 #define DOWN 2
25 #define LEFT 3
26 #define RIGHT 4
27 #define STOP 0
28
29 #define RIGHT_MOTOR 0
30 #define LEFT_MOTOR 1
31
32 #define FORWARD 1
33 #define BACKWARD -1
```

The status bar at the bottom indicates "Ln 8, Col 1 DOIT ESP32 DEVKIT V1 on COM3 [not connected]" and shows system icons for battery, signal, and date/time.

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch_apr3a | Arduino IDE 2.3.2
- File Menu:** File Edit Sketch Tools Help
- Sketch Manager:** DOIT ESP32 DEVKIT V1
- Boards Manager:** Boards included in this package: Arduino Mega or Mega 2560, Arduino Esplora, LilyPad... More info
- Sketch:** sketch_apr3a.ino (selected)
- Code Preview:** The code defines GPIO pins for a camera and sets up an AsyncWebServer to handle camera feed requests.
- Output:** Shows the current line number (Ln 8, Col 1) and the connection status (DOIT ESP32 DEVKIT V1 on COM3 [not connected]).

Interface Creation Code-

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch_apr3a | Arduino IDE 2.3.2
- File Menu:** File Edit Sketch Tools Help
- Sketch Manager:** DOIT ESP32 DEVKIT V1
- Boards Manager:** Boards included in this package: Arduino Mega or Mega 2560, Arduino Esplora, LilyPad... More info
- Sketch:** sketch_apr3a.ino (selected)
- Code Preview:** The code contains a large block of CSS (PROGMEM) defining styles for various HTML elements like head, body, arrows, button, and select.
- Output:** Shows the current line number (Ln 95, Col 1) and the connection status (DOIT ESP32 DEVKIT V1 on COM3 [not connected]).

sketch_apr3a | Arduino IDE 2.3.2

File Edit Sketch Tools Help

DOIT ESP32 DEVKIT V1

BOARDS MANAGER

Filter your search... Type: All

Arduino AVR Boards by Arduino 1.8.6 installed

Boards included in this package: Arduino Mega or Mega 2560, Arduino Esplora, LilyPad... More info

1.8.6 REMOVE

Arduino Mbed OS Edge Boards by Arduino

Boards included in this package: Arduino Edge Control More info

4.1.3 INSTALL

Arduino Mbed OS GIGA Boards by Arduino

Boards included in this package: Arduino Giga More info

4.1.3 INSTALL

Arduino Mbed OS

Output

```
107 opacity: 0.7;
108 -webkit-transition: .2s;
109 transition: opacity .2s;
110
111 .slider:hover {
112   opacity: 1;
113 }
114
115 .slider::-webkit-slider-thumb {
116   -webkit-appearance: none;
117   appearance: none;
118   width: 25px;
119   height: 25px;
120   border-radius: 50%;
121   background: red;
122   cursor: pointer;
123 }
124
125 .slider::-moz-range-thumb {
126   width: 25px;
127   height: 25px;
128   border-radius: 50%;
129   background: red;
130   cursor: pointer;
131 }
132
133 </style>
134
135 </head>
136 <body class="noselect" align="center" style="background-color:white">
137
138 <!--h2 style="color: teal;text-align:center;">Wi-Fi Camera &#128663; Control</h2-->
139
```

Ln 95, Col 1 DOIT ESP32 DEVKIT V1 on COM3 [not connected] ENG IN 15:34 15-05-2024

sketch_apr3a | Arduino IDE 2.3.2

File Edit Sketch Tools Help

DOIT ESP32 DEVKIT V1

BOARDS MANAGER

Filter your search... Type: All

Arduino AVR Boards by Arduino 1.8.6 installed

Boards included in this package: Arduino Mega or Mega 2560, Arduino Esplora, LilyPad... More info

1.8.6 REMOVE

Arduino Mbed OS Edge Boards by Arduino

Boards included in this package: Arduino Edge Control More info

4.1.3 INSTALL

Arduino Mbed OS GIGA Boards by Arduino

Boards included in this package: Arduino Giga More info

4.1.3 INSTALL

Arduino Mbed OS

Output

```
155 <tr>
156   <td></td>
157   <td class="button" ontouchstart='sendButtonInput("MoveCar", "2")' ontouchend='sendButtonInput("MoveCar", "0")'><span class="arrows" >&#8681;</span></td>
158 </tr>
159 <tr><td>
160   <td style="text-align:left"><b>Speed:</b></td>
161   <td colspan=2>
162     <div class="slidecontainer">
163       <input type="range" min="0" max="255" value="150" class="slider" id="Speed" oninput='sendButtonInput("Speed",value)'>
164     </div>
165   </td>
166 </tr>
167 <tr>
168   <td style="text-align:left"><b>Light:</b></td>
169   <td colspan=2>
170     <div class="slidecontainer">
171       <input type="range" min="0" max="255" value="0" class="slider" id="Light" oninput='sendButtonInput("Light",value)'>
172     </div>
173   </td>
174 </tr>
175 </table>
176
177 <script>
178   var webSocketCameraUrl = "ws:///" + window.location.hostname + "/Camera";
179   var webSocketCarInputUrl = "ws:///" + window.location.hostname + "/CarInput";
180   var webSocketCamera;
181   var webSocketCarInput;
182
183   function initCameraWebSocket()
184   {
185     websocketCamera = new WebSocket(webSocketCameraUrl);
186   }
187
```

Ln 95, Col 1 DOIT ESP32 DEVKIT V1 on COM3 [not connected] ENG IN 15:35 15-05-2024

```
var speedButton = document.getElementById("Speed");
sendButtonInput("Speed", speedButton.value);
var lightButton = document.getElementById("Light");
sendButtonInput("Light", lightButton.value);
};

websocketCarInput.onclose = function(event)setTimeout(initCarInputWebSocket, 2000);
websocketCarInput.onmessage = function(event){};

function initWebSocket()
{
    initCameraWebSocket ();
    initCarInputWebSocket();
}

function sendButtonInput(key, value)
{
    var data = key + "," + value;
    websocketCarInput.send(data);
}

window.onload = initWebSocket;
document.getElementById("mainTable").addEventListener("touchend", function(event){
    event.preventDefault()
});

</script>
</body>
</html>
)HTMLHOMEPAGE';

void rotateMotor(int motorNumber, int motorDirection)
{
```

1. Integration and Deployment:

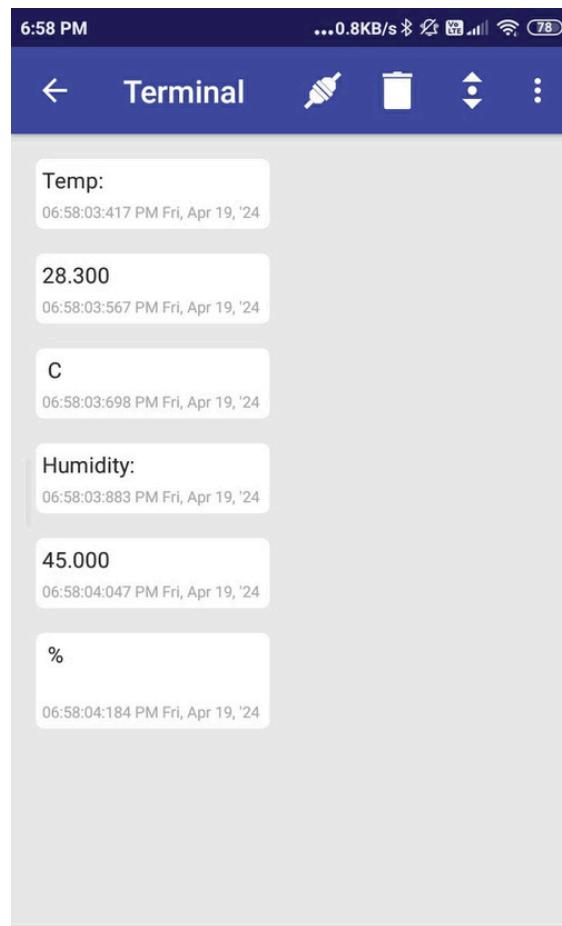
- Once testing is completed successfully, the SARR is integrated into the search and rescue ecosystem.



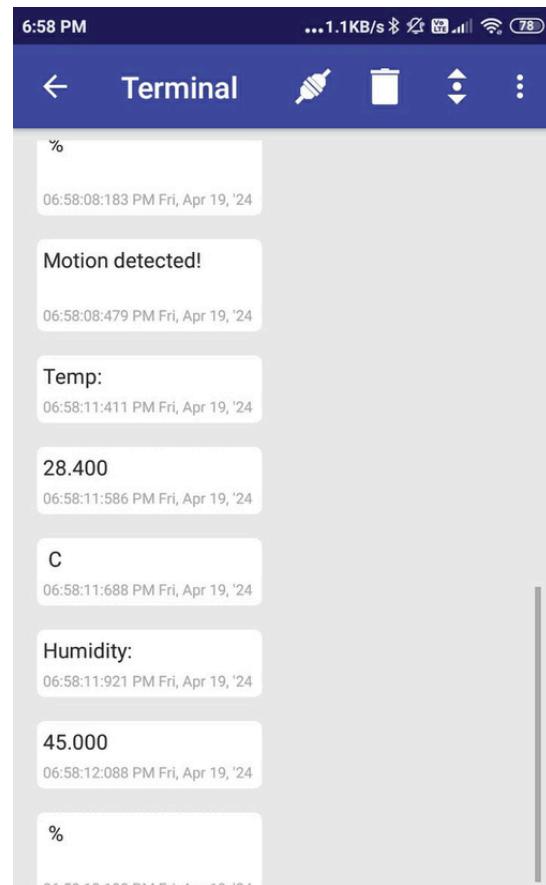
Testing and Evaluation

we have tested on various conditions and collected the data like the range of the data transmission (10m) for the wifi availability (depends on environment to env.) and the sensors data and prediction

we are attaching the sensed data -

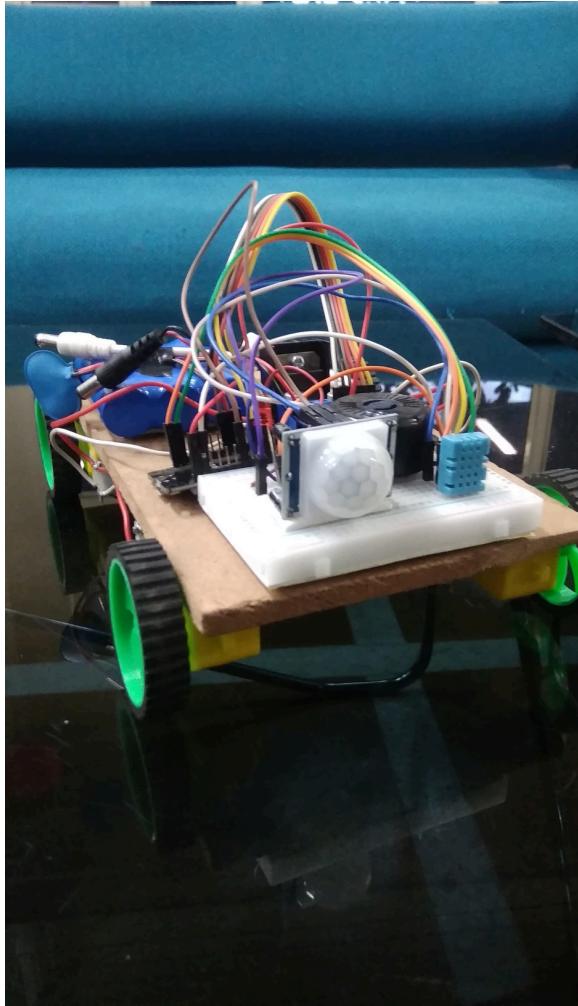


Type here..



Type here..

FINALLY!! THIS IS OUR PRODUCT



FUTURE ENHANCEMENTS

- WE ARE MAKING PROJECT TO BE ABLE TO ASSESSIBLE TO LONG RANGES
- WE ARE ENCHANCING THE ROBOT SO THAT IT CAN BE ABLE TO MOVE TERRAIN SURFACES
- VIDEO STREAMING WE ARE ENHANCING SO THAT IT STREAM CLEAR AND PROPER VIDEO
- BY USE OF MACHINE LEARNING ALGORITHM AND COMPUTER VISION SO THAT IT CAN IDENTIFY OBJECTS
- WE ARE MAKING EFFICIENT OBJECT DETECTION SO THAT IT CAN MOVE CURVED PATHS
- WE ARE MAKING IT TO MAKE USE FOR CLEANING PURPOSE TO CLEAN VENTS, DUCTS

Prepared by:

@all team members

Approved by:

@Alok Mittal Sir
