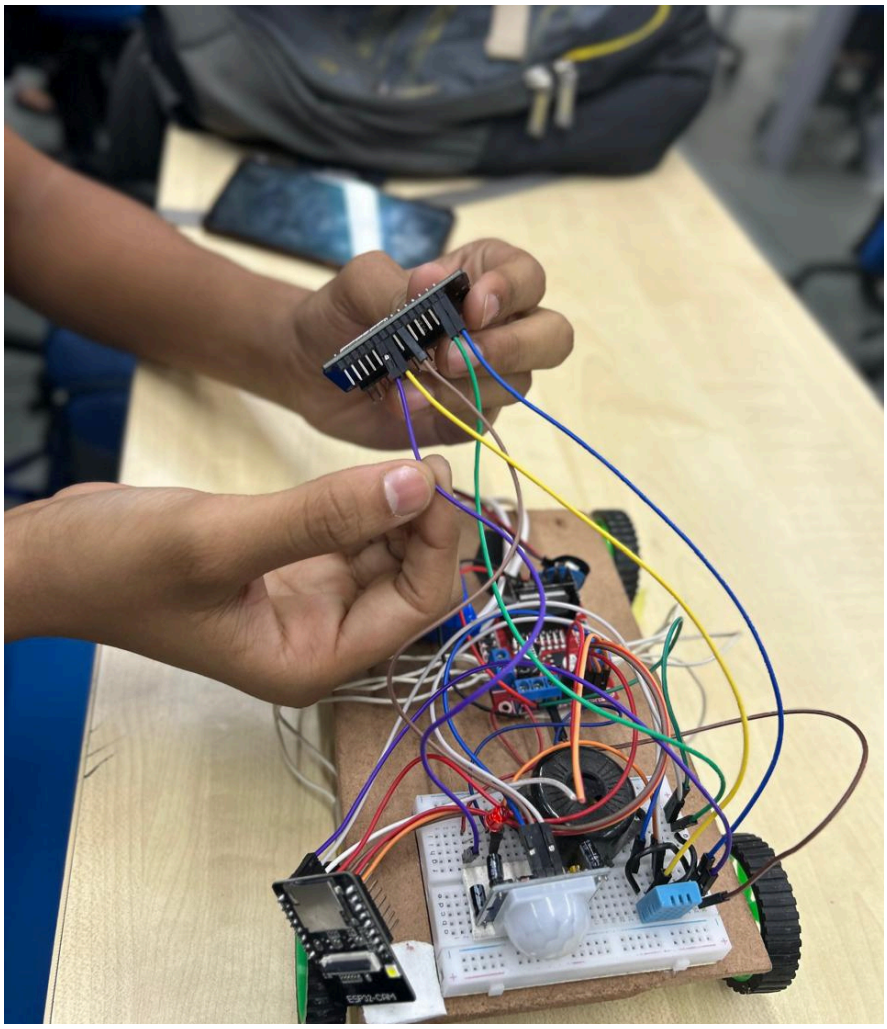
Project Documentation

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 ruppees
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

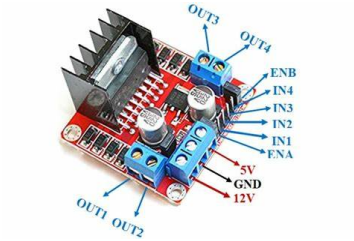
1. **Selection of Platform**: The project begins with the selection of an appropriate microcontroller platform to serve as the brain of the robot. The ESP microcontroller, renowned for its versatility and connectivity capabilities, is chosen as the foundation for the SARR's control system . It has inbuilt 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

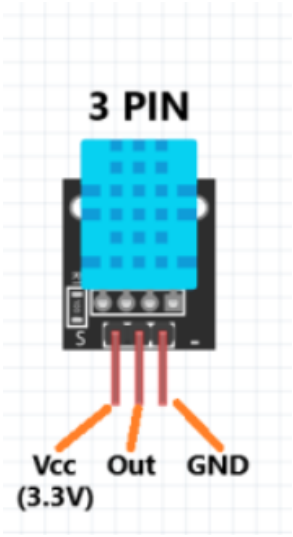
detection, and communication range assessment. Feedback from testing informs iterative improvements and refinements to the robot's design and functionality.

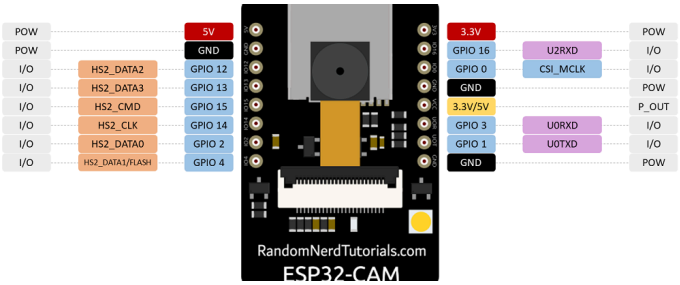
-

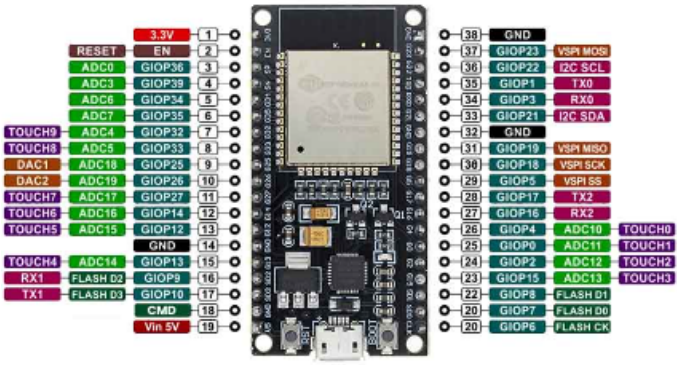
System Architecture

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

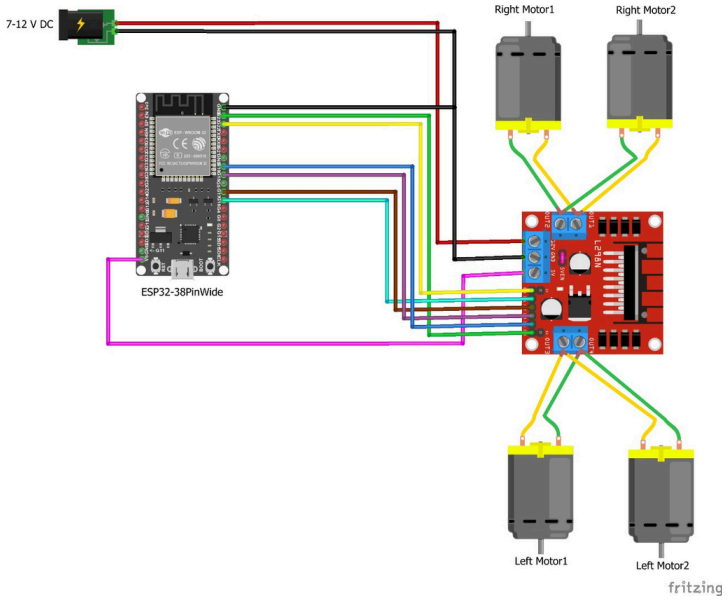


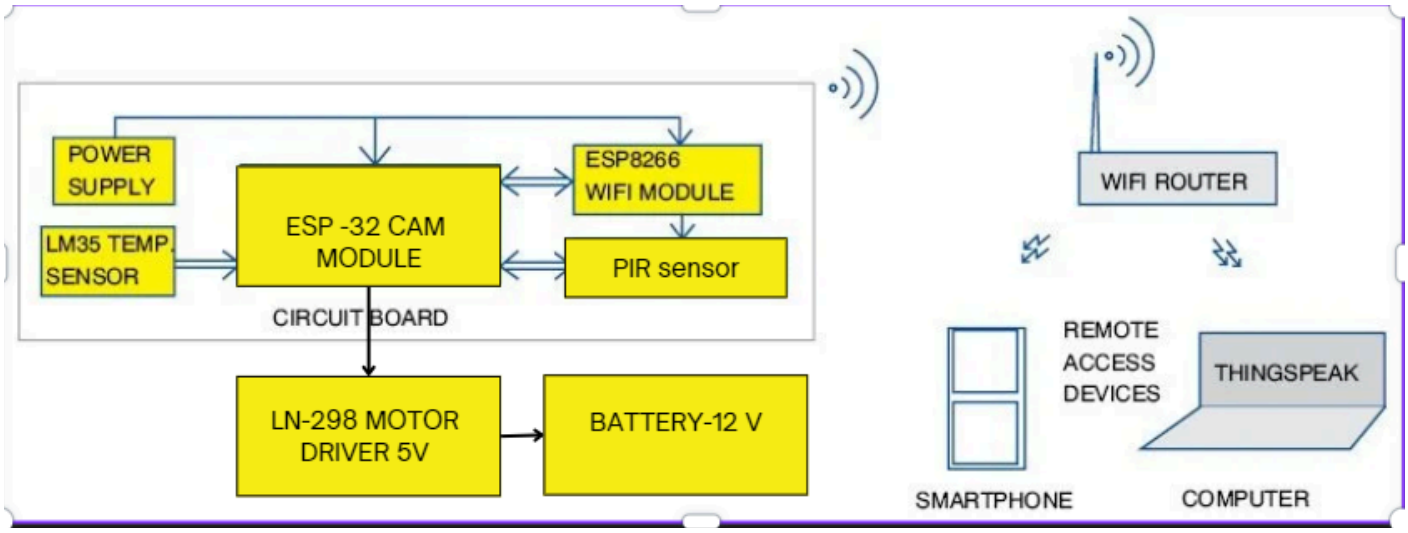






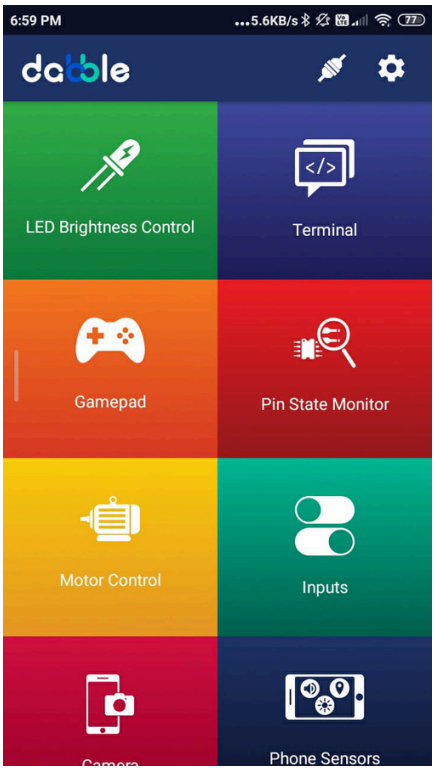
1. **Interaction:** motors are driving through the battery which is connected through motor driver . Our main components ESP-32 and ESP cam module are also integrated with it our peripherals like senors are also connected to the esp module.
2. **BLOCK DIAGRAM:**





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.

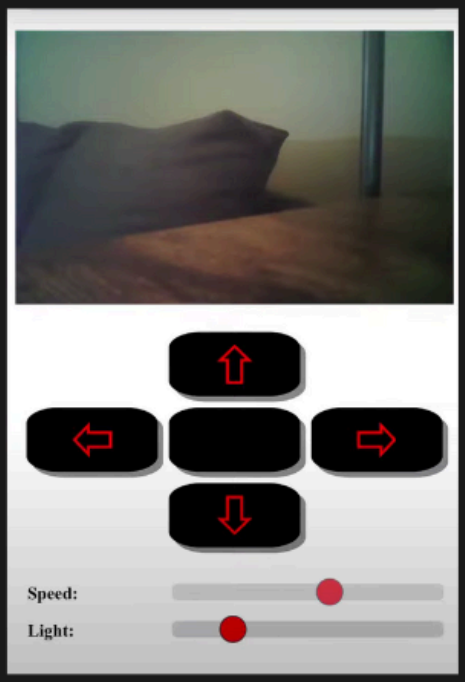


IMPLEMENTATION DETAILS

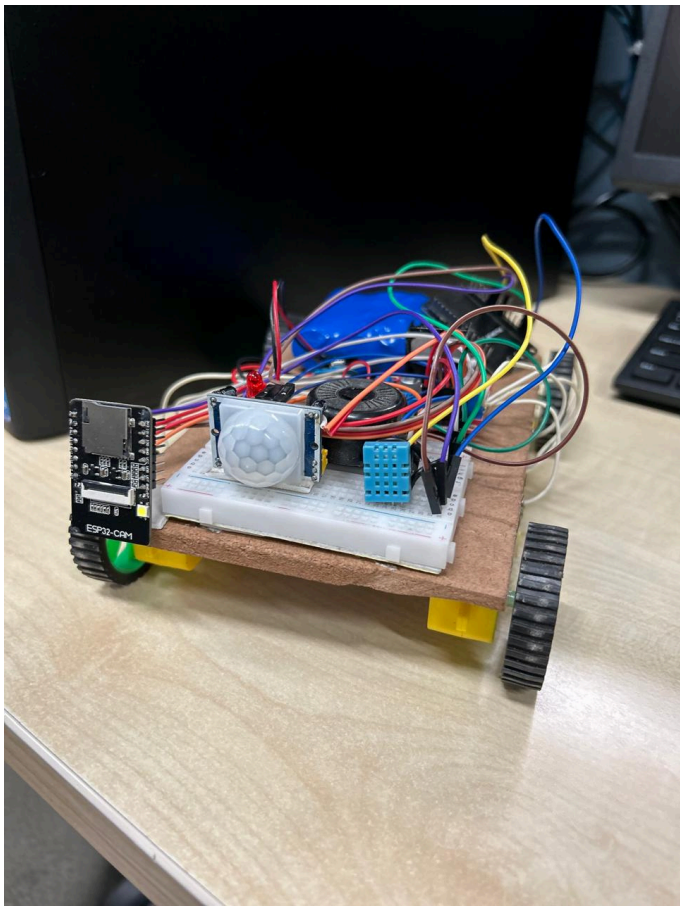
1. **Rover Assembly and Motor Control**:.
   - Dual-channel motor driver modules are utilized to control the movement of DC motors attached to the wheels.
   - The microcontroller (ESP32) sends motor control commands to the motor driver, enabling precise control over the robot's movement speed and direction.

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.

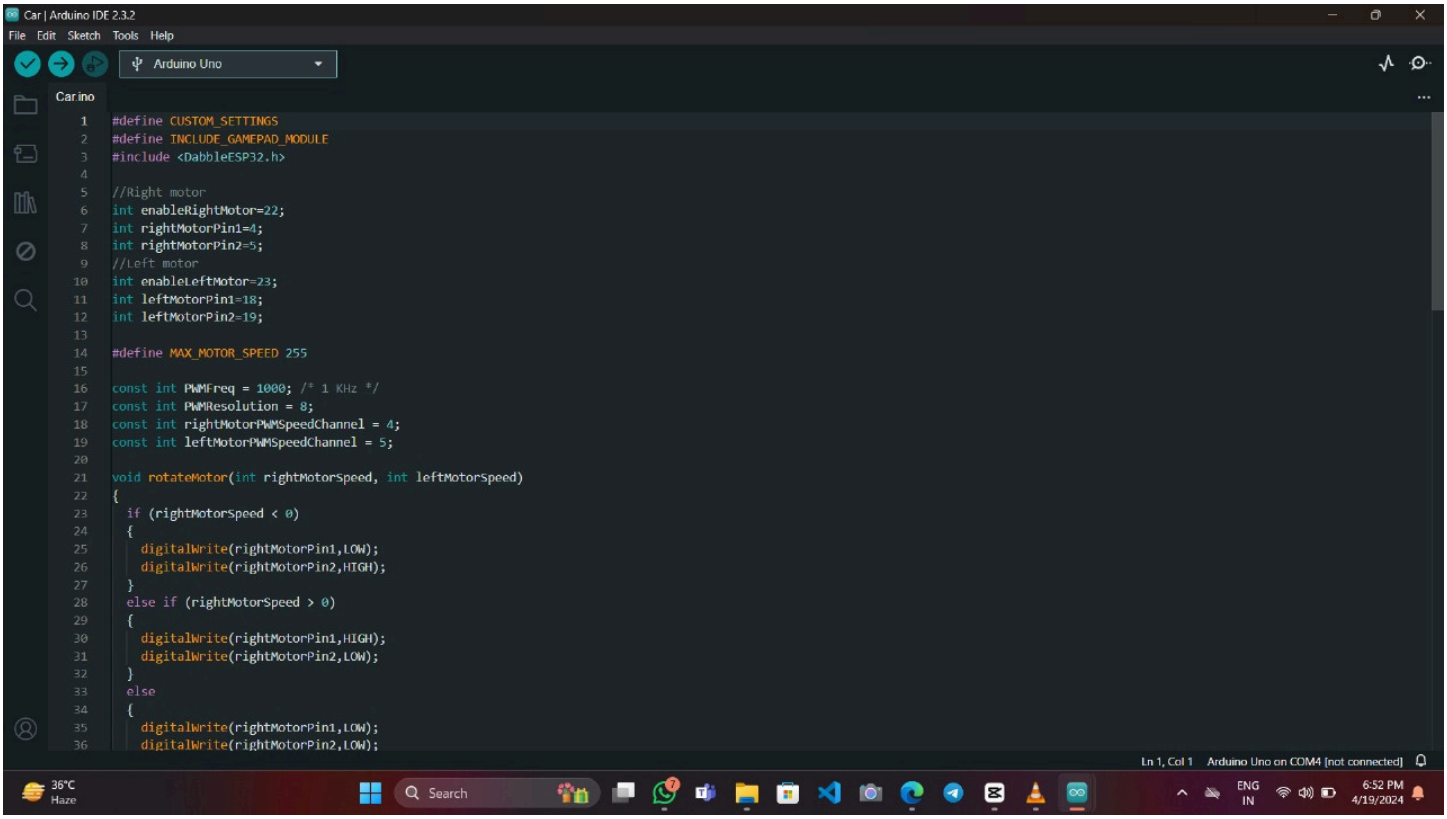


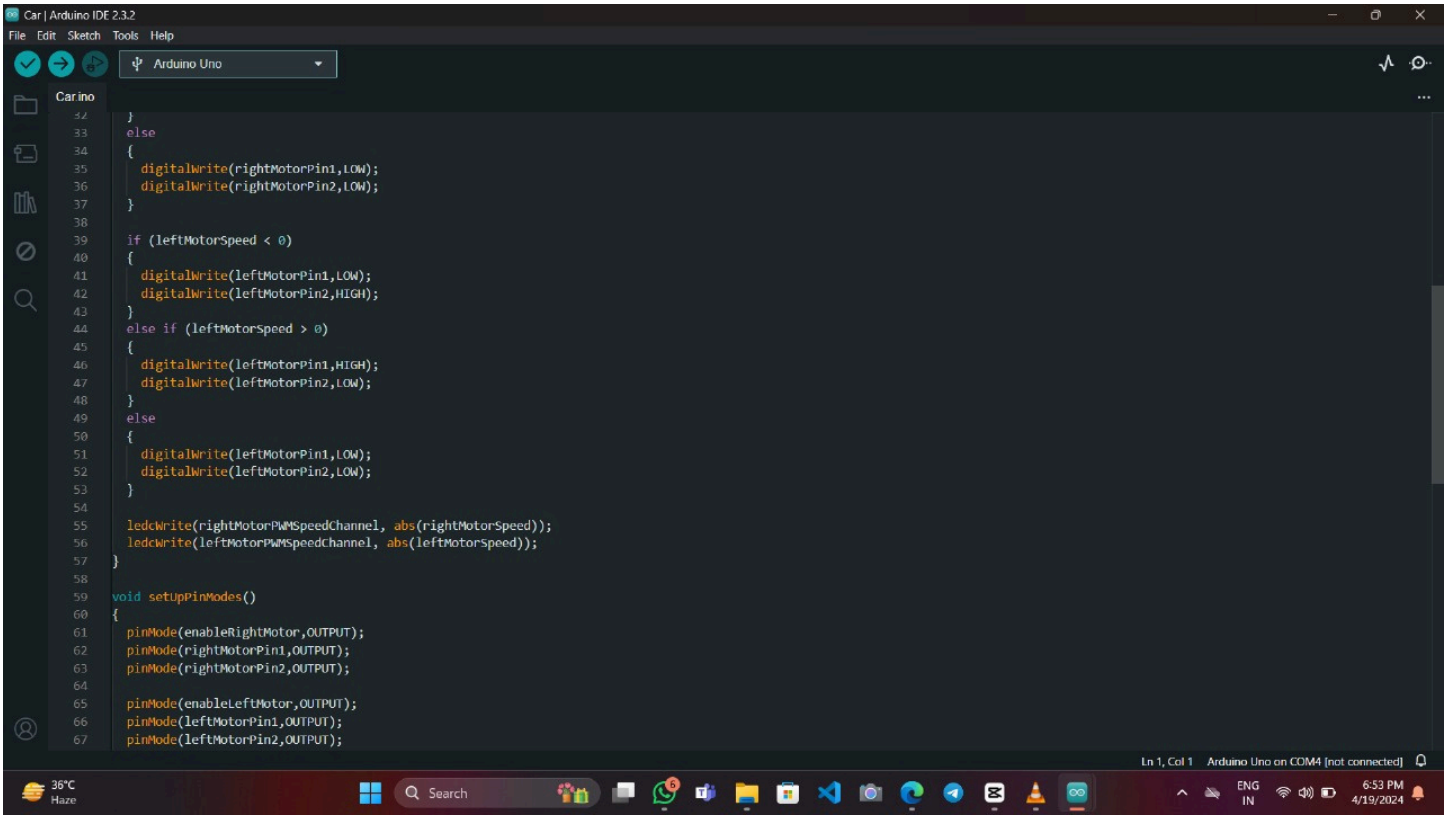
1. **Sensor Integration**:
   - Passive Infrared (PIR) sensors are strategically placed to detect human presence within the robot's vicinity.
   - Additional sensors, such as humidity and pressure sensors, are integrated to monitor environmental conditions.
   - Sensor data is collected and processed 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.



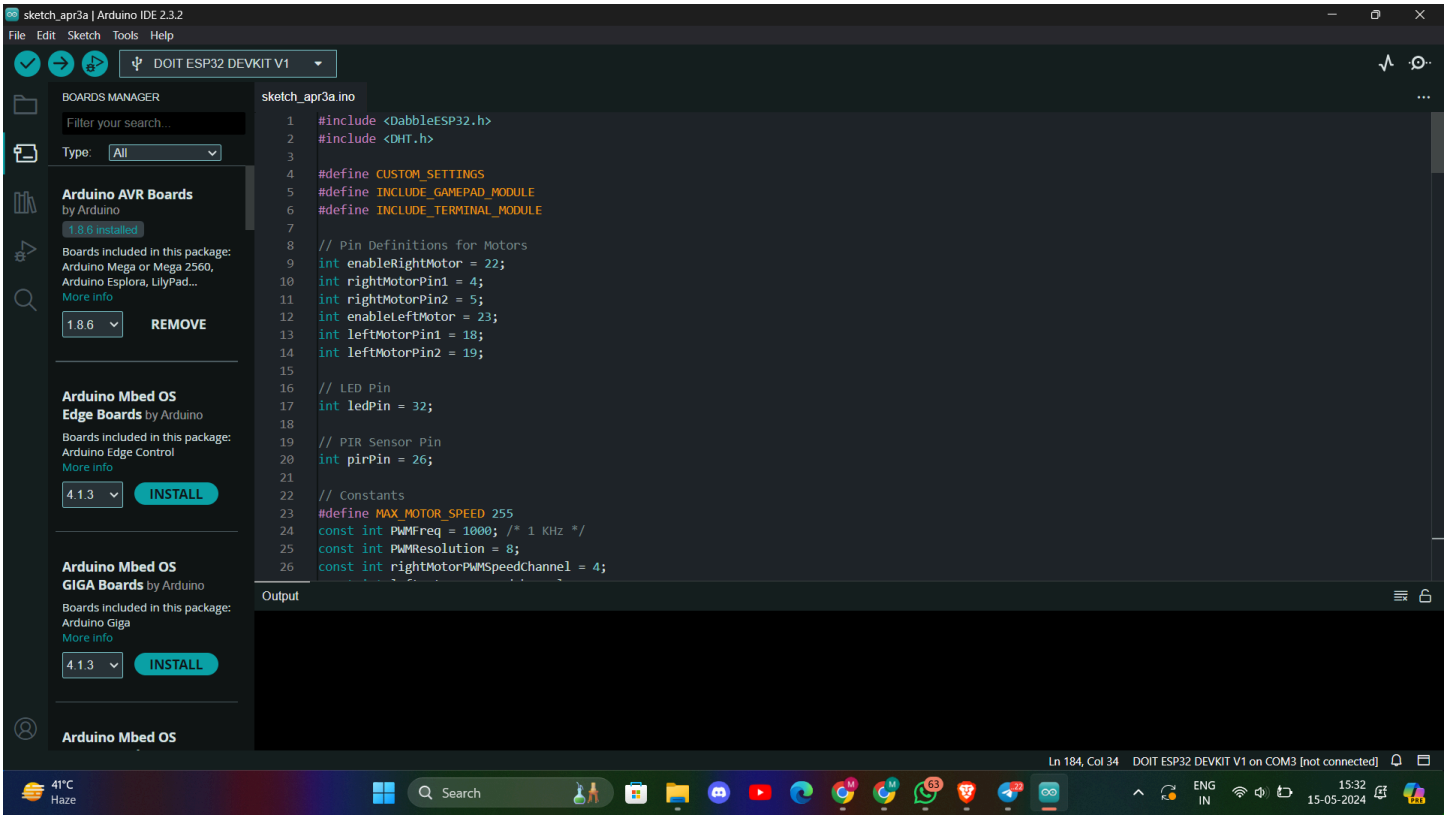
-

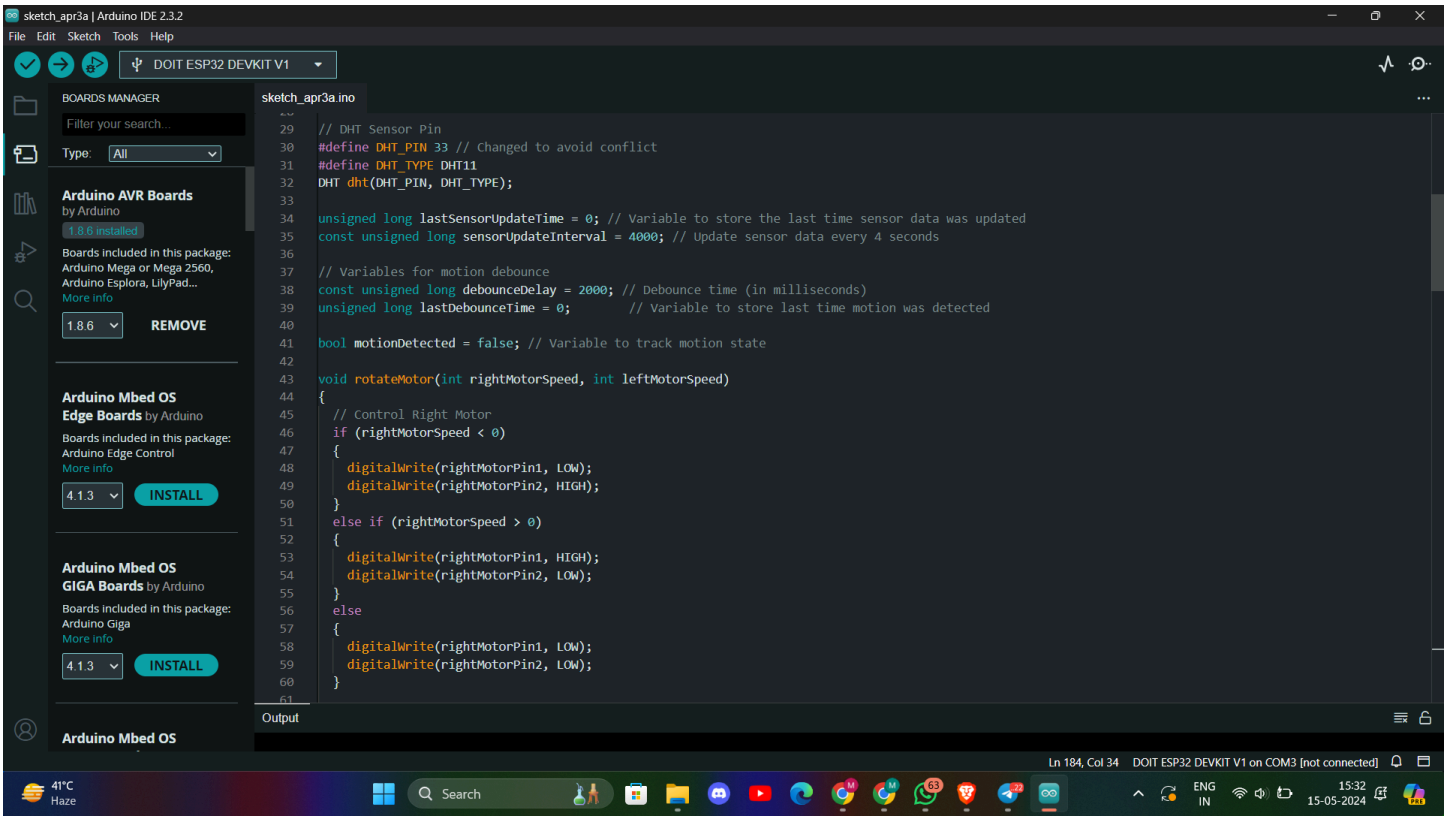
1. **Wireless Communication**:
   - The ESP32 microcontroller is equipped with built-in Wi-Fi and Bluetooth capabilities for wireless communication.
   - Wi-Fi communication enables the robot to transmit sensor data and video feeds to external devices, such as smartphones or control stations.
   - Bluetooth 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 implementated a sound system which can tell the detection
3. **Coding**
   - we have used audrino IDE for programming
   - All the useful librarier 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 .--

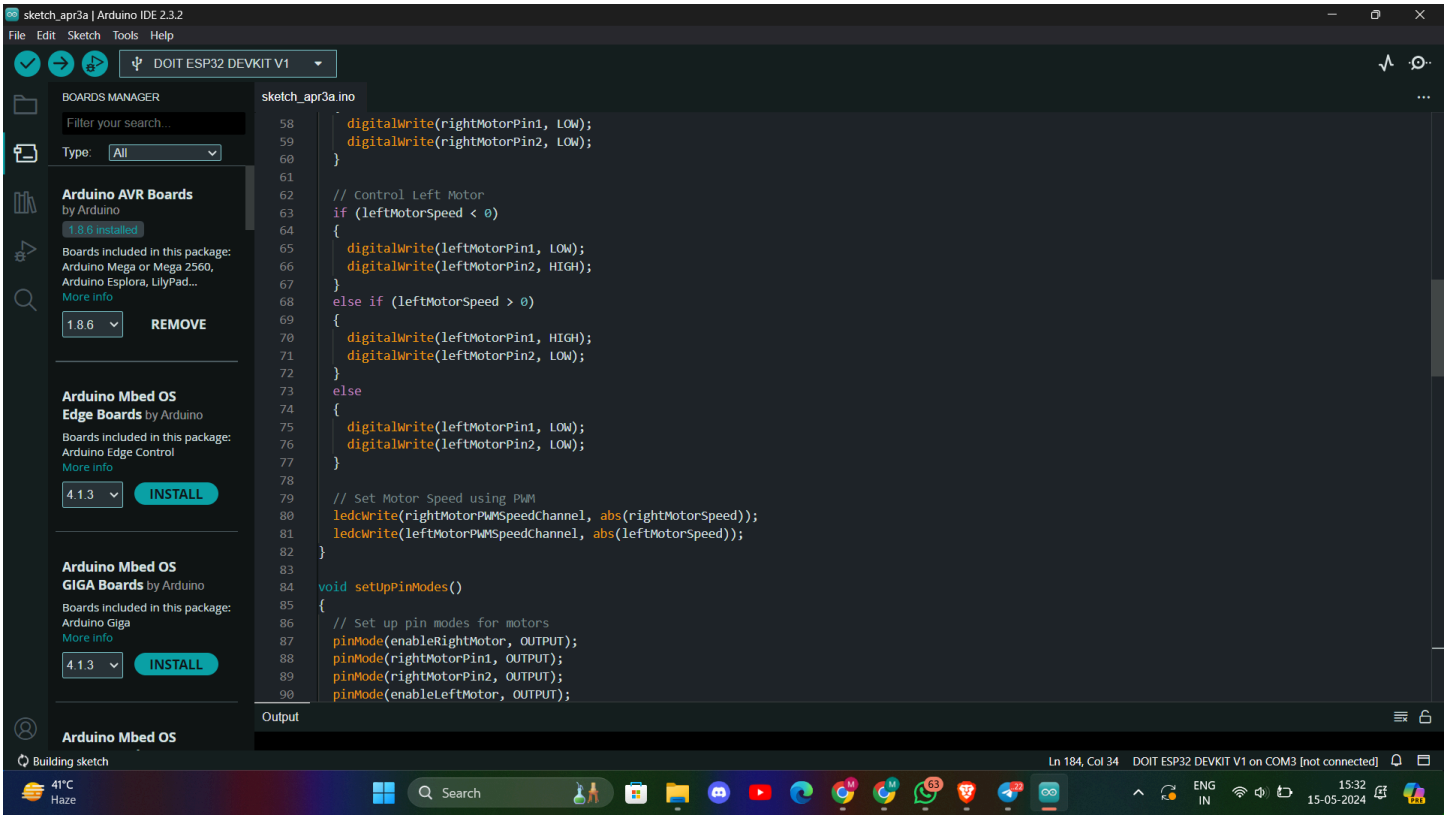




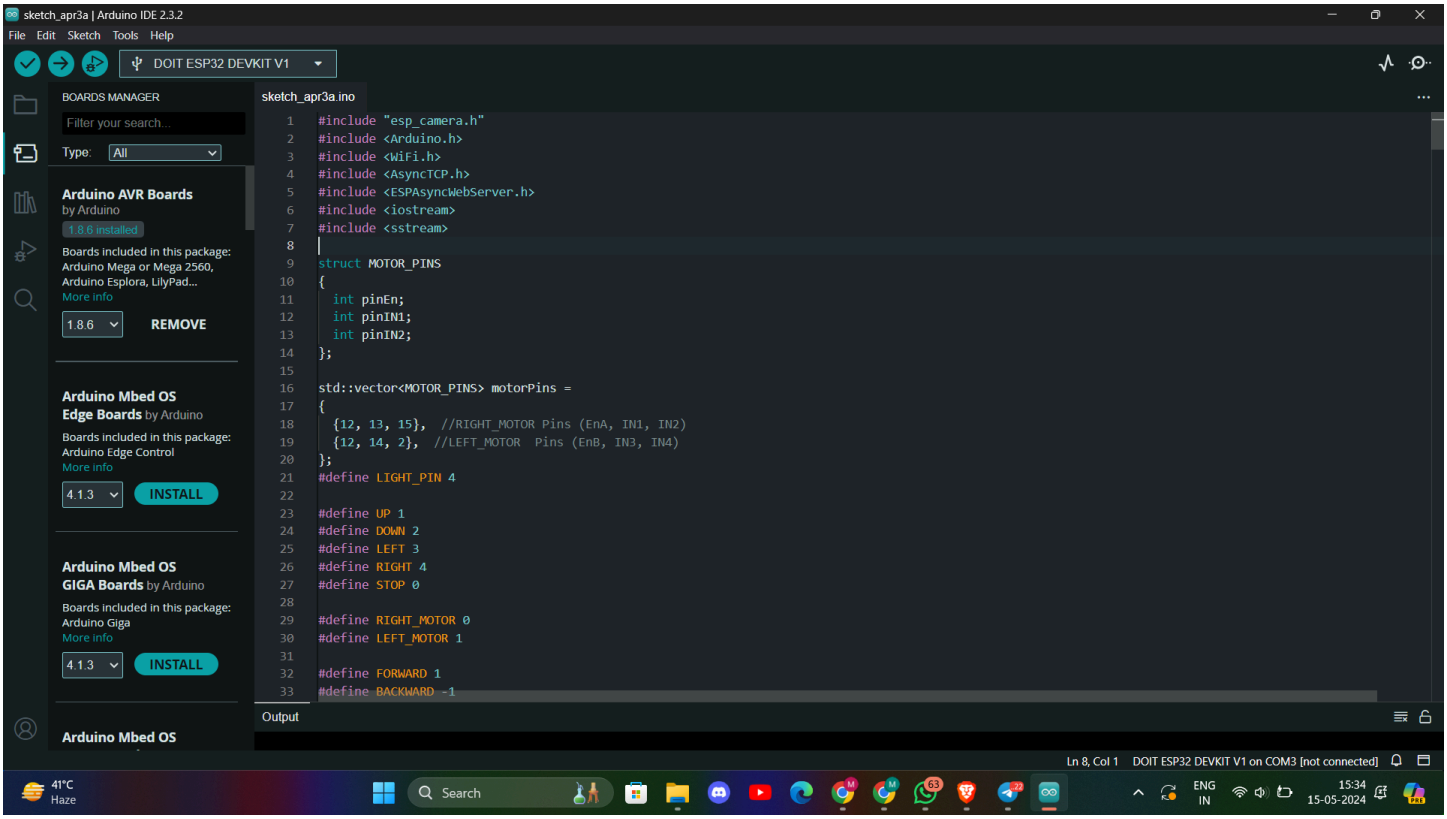
Code for the pin connection--

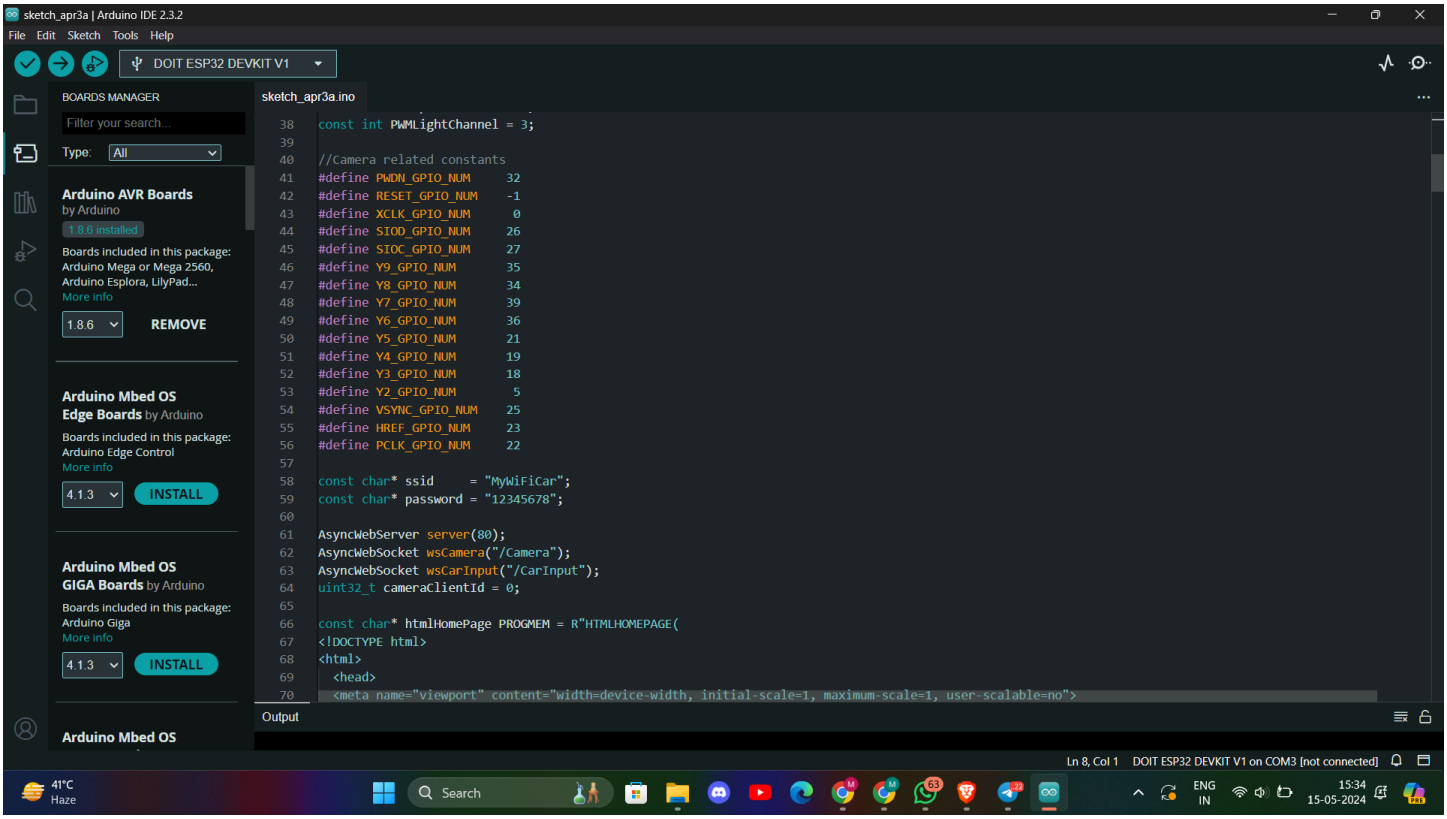

sketch_apr3a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
BOARDS MANAGER
Filter your search...
Type: All
Arduino AVR Boards
by Arduino
1.8.6 installed
Boards included in this package: Arduino Mega or Mega 2560, Arduino Esplora, LilyPad...
More info
1.8.6
REMOVE
Arduino Mbed OS 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
sketch_apr3a.ino
#include <DabbleESP32.h>
#include <DHT.h>
#define CUSTOM_SETTINGS
#define INCLUDE_GAMEPAD_MODULE
#define INCLUDE_TERMINAL_MODULE
// Pin Definitions for Motors
int enableRightMotor = 22;
int rightMotorPin1 = 4;
int rightMotorPin2 = 5;
int enableLeftMotor = 23;
int leftMotorPin1 = 18;
int leftMotorPin2 = 19;
// LED Pin
int ledPin = 32;
// PIR Sensor Pin
int pirPin = 26;
// Constants
#define MAX_MOTOR_SPEED 255
const int PWMFreq = 1000; /* 1 KHz */
const int PWMResolution = 8;
const int rightMotorPWMSpeedChannel = 4;
Output
Ln 184, Col 34 DOIT ESP32 DEVKIT V1 on COM3 [not connected]
41°C Haze
Search
ENG IN
15:32 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
sketch_apr3a.ino
// DHT Sensor Pin
#define DHT_PIN 33 // Changed to avoid conflict
#define DHT_TYPE DHT11
DHT dht(DHT_PIN, DHT_TYPE);
unsigned long lastSensorUpdateTime = 0; // Variable to store the last time sensor data was updated
const unsigned long sensorUpdateInterval = 4000; // Update sensor data every 4 seconds
// Variables for motion debounce
const unsigned long debounceDelay = 2000; // Debounce time (in milliseconds)
unsigned long lastDebounceTime = 0; // Variable to store last time motion was detected
bool motionDetected = false; // Variable to track motion state
void rotateMotor(int rightMotorSpeed, int leftMotorSpeed)
{
// Control Right Motor
if (rightMotorSpeed < 0)
{
digitalWrite(rightMotorPin1, LOW);
digitalWrite(rightMotorPin2, HIGH);
}
else if (rightMotorSpeed > 0)
{
digitalWrite(rightMotorPin1, HIGH);
digitalWrite(rightMotorPin2, LOW);
}
else
{
digitalWrite(rightMotorPin1, LOW);
digitalWrite(rightMotorPin2, LOW);
}
Output
Ln 184, Col 34 DOIT ESP32 DEVKIT V1 on COM3 [not connected]
41°C Haze
Search
ENG IN
15:32 15-05-2024

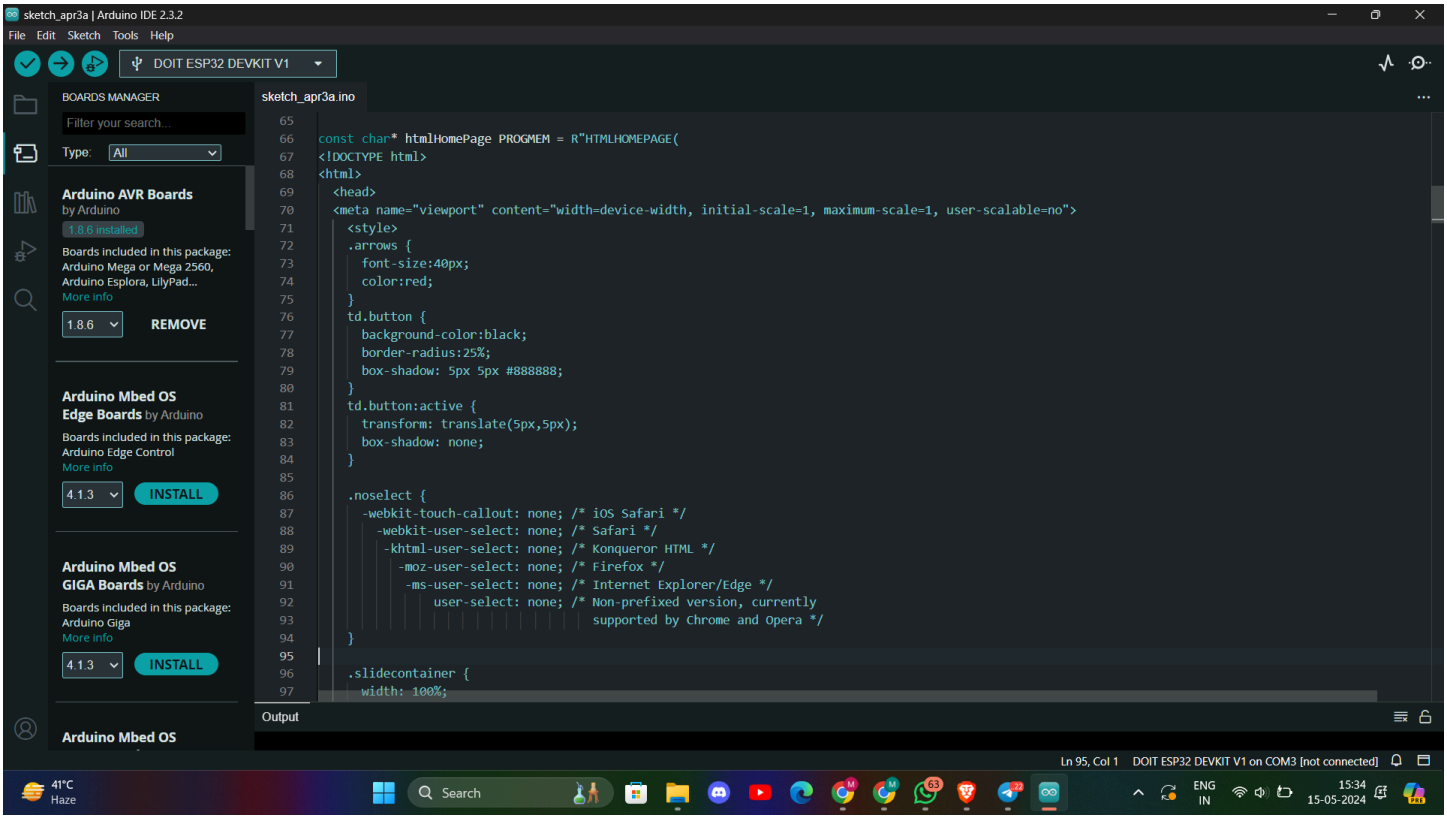


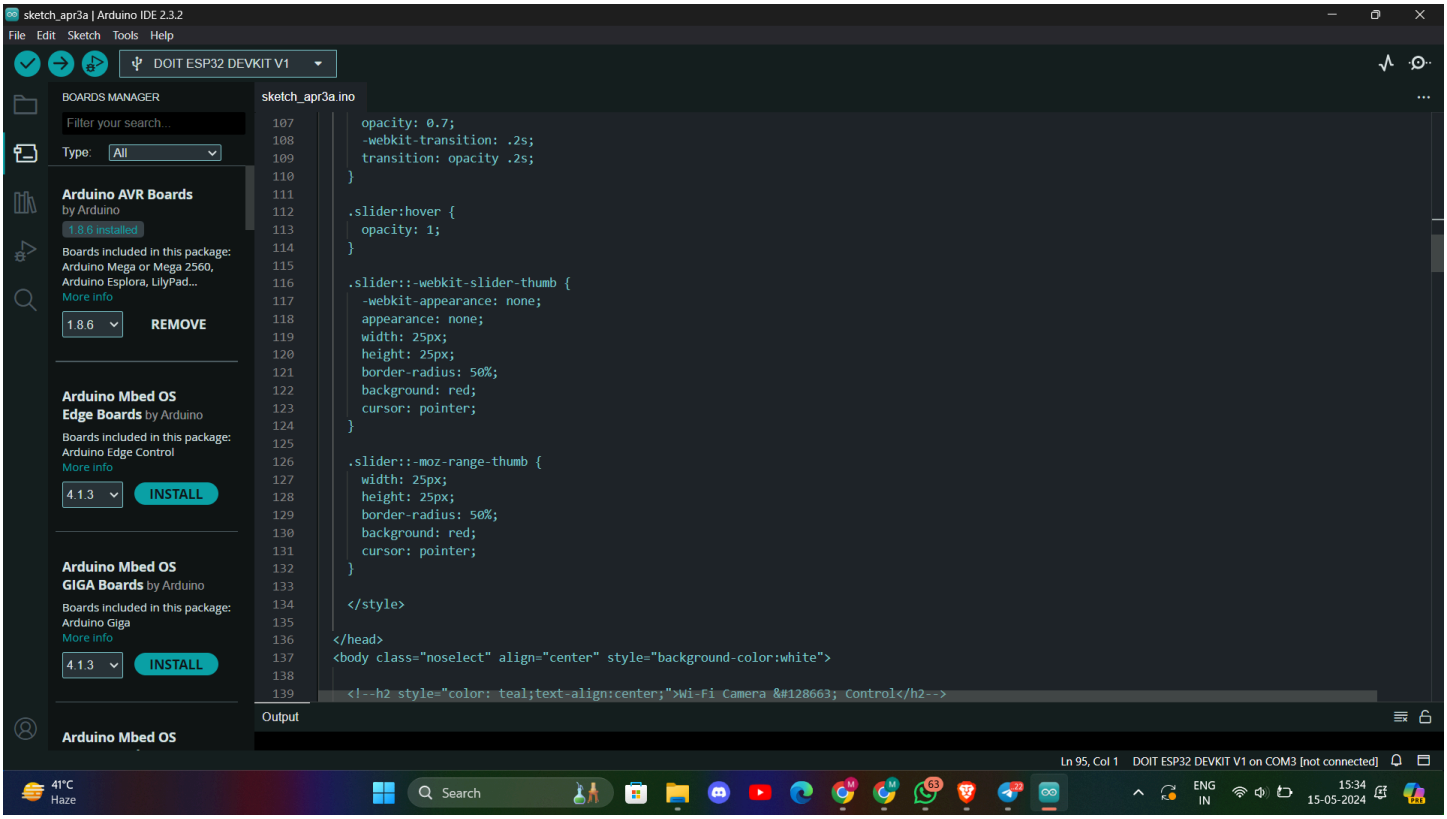
code for the cam module integration

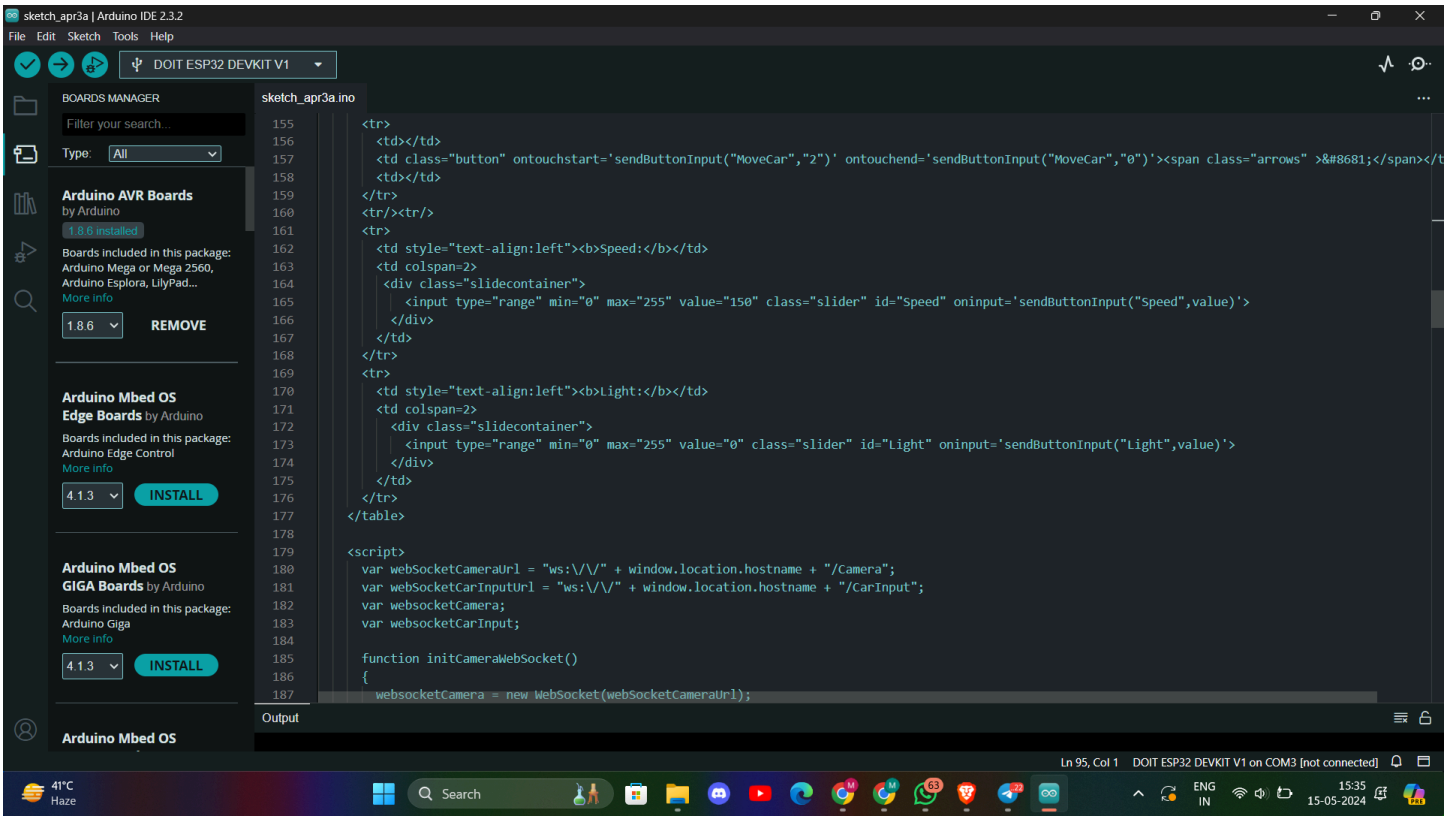




Interface Creation Code-




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
sketch_apr3a.ino
opacity: 0.7;
-webkit-transition: .2s;
transition: opacity .2s;
}
.slider:hover {
opacity: 1;
}
.slider::-webkit-slider-thumb {
-webkit-appearance: none;
appearance: none;
width: 25px;
height: 25px;
border-radius: 50%;
background: red;
cursor: pointer;
}
.slider::-moz-range-thumb {
width: 25px;
height: 25px;
border-radius: 50%;
background: red;
cursor: pointer;
}
</style>
</head>
<body class="noselect" align="center" style="background-color:white">
<!--h2 style="color: teal;text-align:center;">Wi-Fi Camera 🚗 Control</h2-->
Output
Ln 95, Col 1 DOIT ESP32 DEVKIT V1 on COM3 [not connected]
41°C Haze
Search
ENG IN
15:34 15-05-2024

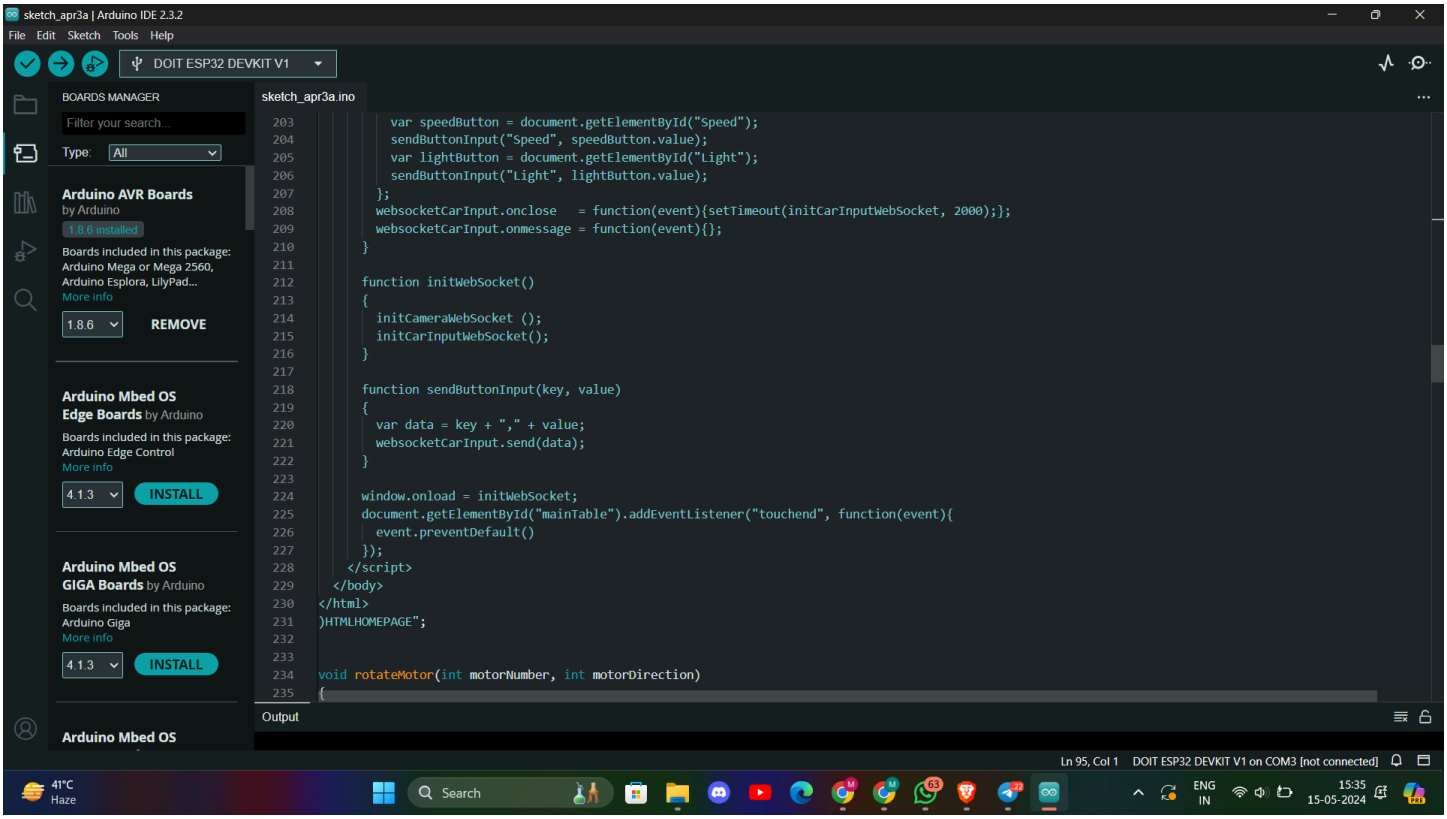

sketch_apr3a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
BOARDS MANAGER
Filter your search...
Type: All
Arduino AVR Boards
by Arduino
1.8.6 installed
Boards included in this package: Arduino Mega or Mega 2560, Arduino Esplora, LilyPad...
More info
1.8.6
REMOVE
Arduino Mbed OS 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
sketch_apr3a.ino
<tr>
<td></td>
<td class="button" ontouchstart='sendButtonInput("MoveCar","2")' ontouchend='sendButtonInput("MoveCar","0")'>⇩</t
<td></td>
</tr>
<tr/><tr/>
<tr>
<td style="text-align:left">Speed:</td>
<td colspan=2>

<input type="range" min="0" max="255" value="150" class="slider" id="Speed" oninput='sendButtonInput("Speed",value)'>

</td>
</tr>
<tr>
<td style="text-align:left">Light:</td>
<td colspan=2>

<input type="range" min="0" max="255" value="0" class="slider" id="Light" oninput='sendButtonInput("Light",value)'>

</td>
</tr>
</table>
<script>
var webSocketCameraUrl = "ws:\/\/" + window.location.hostname + "/Camera";
var webSocketCarInputUrl = "ws:\/\/" + window.location.hostname + "/CarInput";
var websocketCamera;
var websocketCarInput;
function initCameraWebSocket()
{
websocketCamera = new WebSocket(webSocketCameraUrl);
Output
Ln 95, Col 1 DOIT ESP32 DEVKIT V1 on COM3 [not connected]
41°C Haze
Search
ENG IN
15:35 15-05-2024

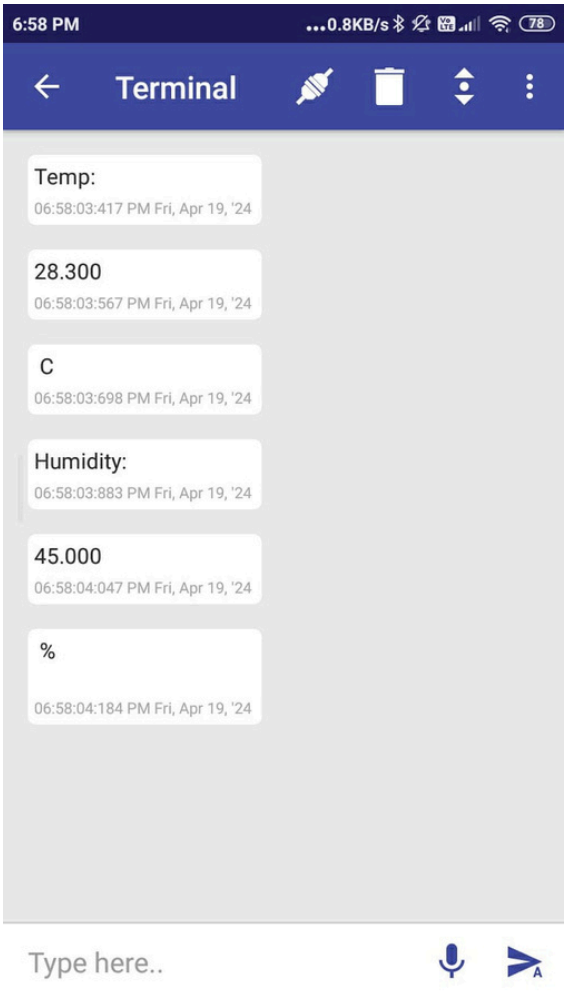


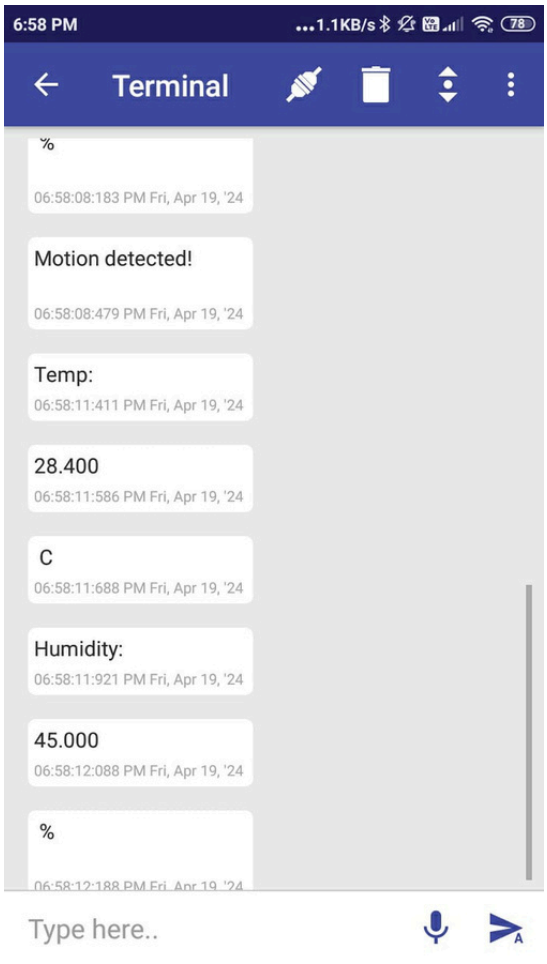
1. **Integration and Deployment**:
   - Once testing is completed successfully, the SARR is integrated into the search and rescue ecosystem.

Testing and Evaluation

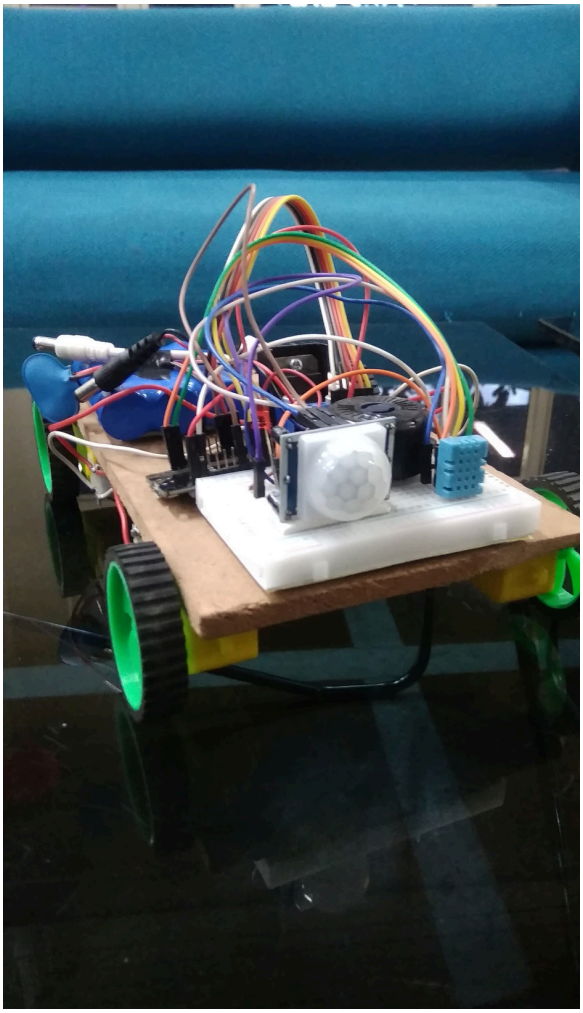
we have tested on various conditions and collected the data like the range of the data transmission (10m) for the wifi availability (depends on environment to env.) and the sensors data and prediction

we are attaching the sensed data -





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:	Approved by:
@all team members	@Alok Mittal Sir