Smart Windshield Monitoring and Control System Using ESP32 with Automated Wiper and Heater Functions

A PROJECT REPORT

Submitted by

Abishek C- RA2211004010556
Vishwa K -RA2211004010554
Chidambaresh N - RA2211004010555

Semester – V Academic Year: 2024–25

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

College of Engineering and Technology ,
SRM Institute of Science and Technology
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu.

ABSTRACT

In modern automotive systems, driver safety and visibility are crucial factors for ensuring a secure driving experience. This project presents the development of a Smart Windshield Monitoring and Control System using an ESP32 camera module. The system is designed to enhance visibility and automate vehicle responses to adverse weather conditions. The ESP32 camera module continuously monitors the windshield for visibility impairments such as rain, fog, or other obstructions. When the camera detects any vision obstacle, it collects data from the external environment and projects the information onto a dashboard display, providing real-time alerts to the driver. Additionally, the system incorporates a rain sensor to detect rainfall and automatically activate the windshield wipers, ensuring clear visibility without manual intervention.To further improve safety during foggy conditions, the system is equipped with a heater automation feature. When fog is detected, the heater is triggered to prevent condensation on the windshield, thus maintaining clear visibility. The integration of these features creates an intelligent and responsive windshield management system that optimizes driving safety in various environmental conditions. The implementation of this project highlights the efficiency of using low-cost IoT components like the ESP32, rain sensor, and temperature sensors to automate critical vehicular functions, ultimately contributing to safer and smarter vehicle operations.

OBJECTIVE

- Enhance driver safety by monitoring windshield visibility and providing real-time alerts.
- Automate windshield wiper activation using a rain sensor to ensure continuous visibility during rainfall.
- Project environmental data to a dashboard display when visibility is impaired or obstacles are detected on the windshield.

CHAPTER 1

INTRODUCTION

Visibility through the windshield is critical for safe driving, particularly in adverse weather conditions like rain, fog, or when obstacles obstruct the driver’s view. Traditional manual systems for controlling windshield wipers and defogging mechanisms can be slow and inefficient, increasing the risk of accidents. To address these challenges, there is a growing interest in smart automotive technologies that can autonomously detect and respond to environmental changes, ensuring safer driving experiences.

This project introduces a Smart Windshield Monitoring and Control System, designed to enhance driver visibility through real-time monitoring and automation. The system utilizes an ESP32 camera module to continuously assess the visibility conditions of the windshield. When vision is impaired by rain, fog, or other obstacles, the camera gathers data from the external environment and sends it to a dashboard display, alerting the driver with necessary information. By automatically identifying visibility issues, the system reduces the driver's reliance on manual intervention, allowing them to focus on navigating the road more safely.

The core of the system involves two key automation features: a rain sensor that controls the windshield wipers and a heater that prevents fogging. When rain is detected, the rain sensor triggers the wipers to automatically clear the windshield, removing the need for manual activation. Similarly, in foggy conditions, the system activates the heater to prevent condensation buildup on the glass, maintaining clear visibility. These automated functions not only enhance safety but also improve driver comfort by reducing the need for manual controls.

By leveraging IoT components like the ESP32, this project provides a low-cost and energyefficient solution for modern vehicles. The smart windshield system not only responds to environmental changes in real time but also offers scalability for further development, such as integrating additional sensors or expanding its applications to other automotive features. This project demonstrates the potential for smart, automated systems to play a crucial role in advancing vehicle safety and driver assistance technologies.

CHAPTER 2 LITERATURE SURVEY

1. **Intelligent Windshield for Automotive Vehicles**

   Published in: 2011 World Congress on Information and Communication Technologies
   Authors: Utpal V. Solanki and Nilesh H. Desai G.H. Patel College of Engineering and Technology, Gujarat Technological University Gujarat, India.

   This research paper discusses the development of an intelligent windshield control system for automotive vehicles. It integrates water, dust, and light sensors to control wiper speed based on rain intensity, clean dust from the windshield, and adjust the sun visor according to light levels.

2. **Smart Car Using IoT and Image Processing**

   Published in: 2023 7th International Conference On Computing, Communication, Control And Automation
   Authors: Vijay Gaikwad and and Neha Kamtalwar, Department of Computer Engineering, Vishwakarma Institute of Technology, Pune. India

   This research paper explains the development of a Smart Car System using IoT and image processing to automate critical functions in vehicles. This covers the integration of various functions such as automatic wipers triggered by a rain sensor, a parking alarm using an ultrasonic sensor, weather sensing to adjust vehicle conditions, and lane detection

3. **IoT-Based Smart Parking Management System Using ESP32 Microcontroller**

   Published in: 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI2022) - 6-7 October 2022
   Authors: Joni Welman Simatupang , Vincent, School of Electrical Engineering and Informatics Institut Teknologi Bandung Bandung 40132, Indonesia
   This research paper explains the design and implementation of an IoT-based smart parking management system using the ESP32 microcontroller. The system allows drivers to check for vacant parking spots in real-time and reserve a slot before arriving, reducing time spent searching for parking.

4. **An effective control of auto defog system to keep automobile windshield glass clear,**
   Published in: International Conference on Control, Automation and Systems 2010
   Oct. 27-30, 2010 in KINTEX
   Authors: Jeong-Hoon Lee , Sang-Ho Oh and Kyung-Soo Kim
   This paper, *An Effective Control of Auto Defog System to Keep Automobile Windshield Glass Clear*, focuses on an automatic defogging system designed to improve driver visibility by preventing fog buildup on windshields. It uses a humidity sensor to detect fog probability and control an HVAC system that adjusts air conditioning, fresh air intake, and fan speed to maintain clear visibility.

5. **Automatic rain sensing car wiper**
   Published in: International Journal of Advance Research, Ideas and Innovations in Technology,
   Authors: P. Abhilash Reddy, G. Sai Prudhvi, P J Surya Sankar Reddy, Dr. S. S. Subashka Ramesh

   This research paper, titled *Automatic Rain Sensing Car Wiper*, discusses a system that automates windshield wiper activation based on rain intensity. Using a rain sensor connected to an Arduino microcontroller, the system detects rainfall and adjusts the wiper speed accordingly. This setup includes a servo motor for wiper movement and an LCD display to show rainfall intensity levels. The goal is to improve driver convenience and safety by reducing the need for manual wiper control.

6. **ESP32 Based Smart Surveillance System**
   Published in: 2019 International Conference on Computing, Mathematics and Engineering Technologies – iCoMET 2019
   Authors: Pertab Rai and Murk Rehman

   This research paper, *ESP32 Based Smart Surveillance System*, details the development of a surveillance system using the ESP32 microcontroller. The system captures video via an Arducam Mini camera module and streams it wirelessly to a TFT display using Wi-Fi, creating a low-cost, IoT-enabled surveillance solution. It discusses both hardware and software configurations, including data packet transmission and decoding for real-time video display.

7. **Smart Helmet Wiper**

   Published in: 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science

   Authors: Sarika Kanetkar , Ankit Rathore , Krati Maheshwari, Prasanna Dubey and Ankit Saxena

   This research paper, titled *Smart Helmet Wiper*, presents a design for a helmet-mounted automatic rain-sensing mini wiper system. The system uses an Arduino Nano microcontroller and a rain sensor to detect water droplets on the helmet's eye shield and activate the wiper automatically. The wiper speed adjusts to the rainfall intensity, with a manual switch for optional control. This system aims to enhance safety and comfort for motorcyclists by improving visibility during rainy conditions.

   This project aligns with our work in automating visibility enhancements, as both use sensors and microcontrollers to address environmental challenges that impact clear vision for users.

8. **The design and implementation of rain sensitive triggering system for windshield wiper motor**

   Published in: *IEMDC 2001. IEEE International Electric Machines and Drives Conference* Authors: M. Ucar, H. M. Ertunc and O. Turkoglu

   This research paper discusses the design and implementation of a rain-sensitive triggering system for windshield wipers. It explains two existing wiper control methods—time-based and infrared reflection—and introduces a new method using rain sensors that detect the conductivity of raindrops on the windshield. The system operates based on the amount of rain detected, which controls the wiper motor's activity. Experimental results with a single sensor demonstrate its reliability, with the potential for increased accuracy by incorporating multiple sensors to better assess rain coverage.

9. **Effective Weather Monitoring System using MQTT Protocol**

   Published in: ICCES 2023,Proceedings of the 8th International Conference on Communication and Electronics Systems,

   Authors: R. L. Paulraj, M. Tejasree, S. M. D and S. Ragasahithi,

   This research paper presents an effective weather monitoring system using the MQTT protocol and an ESP32 microcontroller with built-in Wi-Fi capabilities. The system monitors environmental parameters like temperature, humidity, and rainfall using sensors and sends this data to the cloud. The setup displays real-time weather data on an LCD screen, with potential alerts sent via SMS, email, or social media when thresholds are reached. This low-cost, low-energy design is adaptable for applications in agriculture, household monitoring, and broader weather prediction efforts using IoT technology.

10. **Enhancing Weather Monitoring: A Comprehensive Study Utilizing IoT, ESP32, Sensor Integration, and Blynk Platform**

Published in: 2024 IEEE 10th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)

Authors: J. W. Simatupang, A. M. Lubis and Vincent

This research paper explores the integration of IoT technology using the ESP32 microcontroller to enhance weather monitoring systems. It focuses on using various sensors to collect real-time meteorological data such as temperature, humidity, rainfall, and wind speed. The study utilizes the Blynk platform for visualizing and remotely accessing weather data through smartphones. The proposed system aims to provide accurate and timely updates, improving weather forecasting and decision-making across various industries like agriculture and disaster management. The results demonstrate high accuracy in data measurements and effective use of IoT for real-time weather monitoring.

CHAPTER 3

SYSTEM DESCRIPTION

Block diagram of Environment data to Dashboard Display.

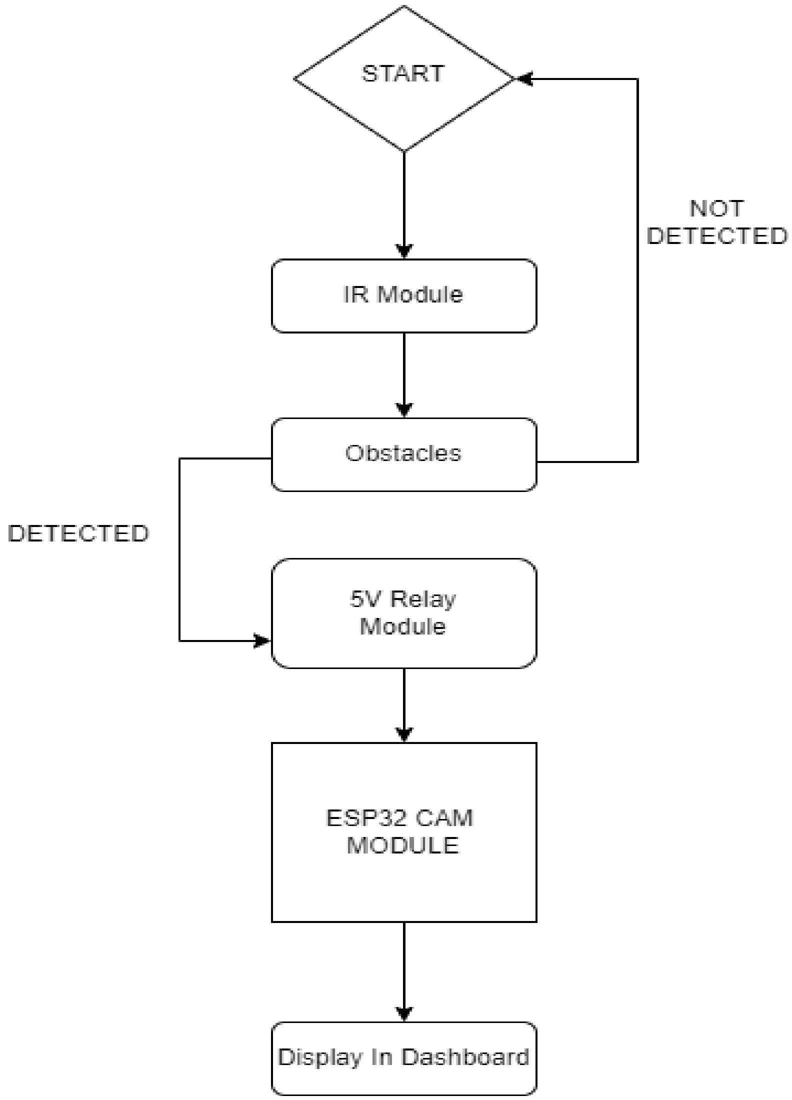

Fig 3.1

The IR module which is placed behind the windshield within the car detects some foreign particles which obstructs the visibility of the driver. That signal is then given to the 5v Relay which triggers/ switch on the ESP32 Cam Module. Then the ESP32 Cam Module continuously transfers data to Dashboard display, till the obstacle is removed.

Block diagram of sensing rain and triggering wiper.

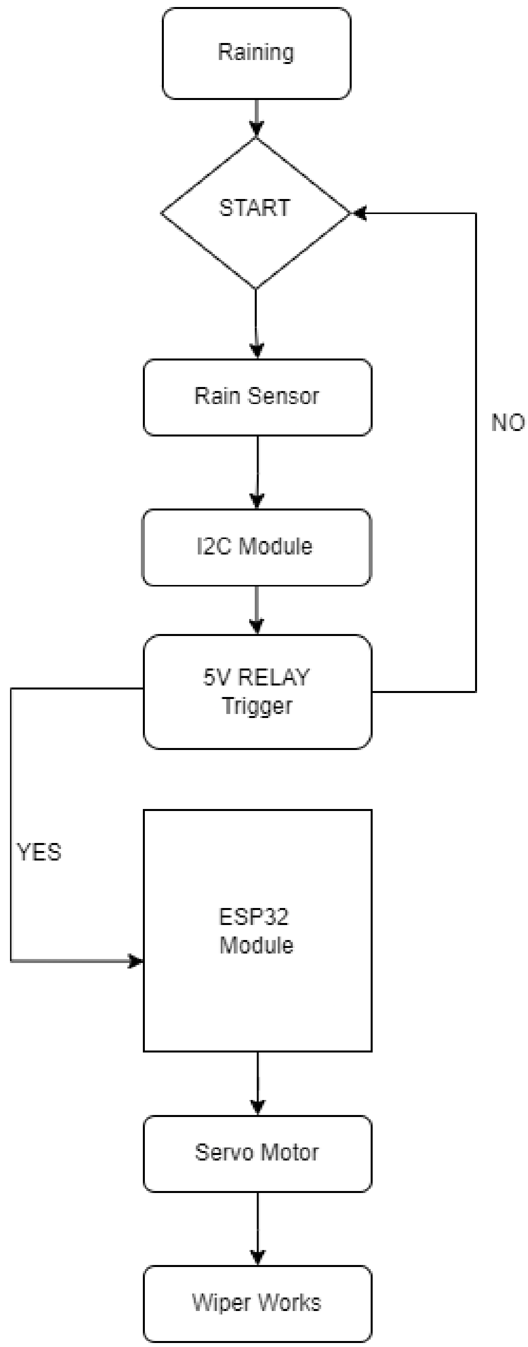

Fig 3.2

The rain sensor senses the presence of rain droplet in it by connecting the circuit with the Rain sensor. That signal is then transferred into 5v Relay, which then triggers the ESP32 module , which controls the servo motor which is then attached to the wiper.

Block diagram of sensing fog and automatic heater

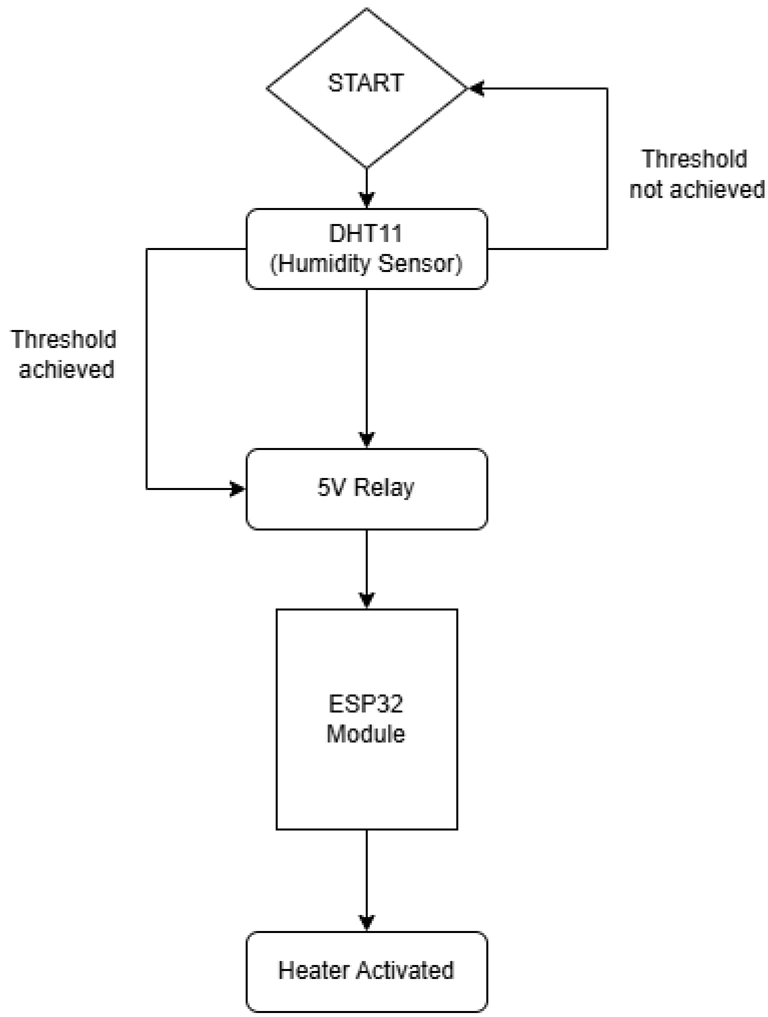

Fig 3.3

The DHT-11 sensor recogonizes the presence of fog in environment . The signal from DHT-11 is fed into the 5v relay, which then gives the trigger signal to activate the heater mechanism.

CHAPTER 4

HARDWARE SPECIFICATION

Rain Sensor

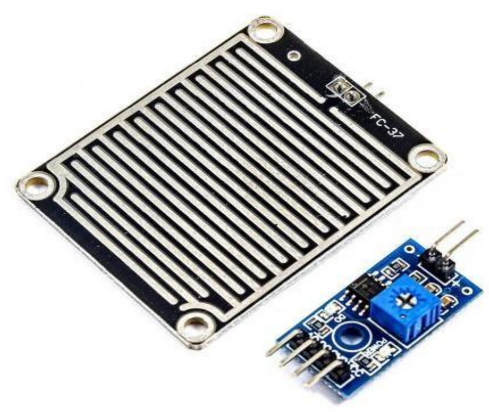

Fig 4.1 Rain Sensor

A rain sensor is a device used to detect the presence and intensity of rain by sensing moisture. It typically consists of a rain-sensing pad with conductive tracks and a controller. When raindrops hit the pad, the circuit between the conductive tracks closes, signalling the presence of rain. The strength of this signal can be used to determine rain intensity.

Application in this Project:

In our Smart Windshield Monitoring System**,** the rain sensor plays a critical role in automating the windshield wiper function. When rain is detected on the windshield, the rain sensor sends a signal to the ESP32 microcontroller, which then activates the wipers at an appropriate speed based on the rain intensity. This automation reduces the need for manual wiper control, ensuring the windshield stays clear and the driver remains focused on the road, enhancing safety and convenience during wet conditions.

DHT 11 Sensor

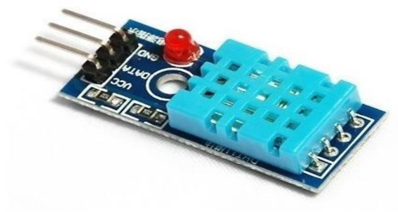

Fig 4.2 DHT 11 Sensor

The DHT11 Temperature and Humidity Sensor is a digital sensor used to measure both temperature and humidity levels in the environment. It consists of a thermistor for temperature sensing and a capacitive humidity sensor, producing accurate readings within a range of 20-90% humidity and 0-50°C temperature. The DHT11 outputs data as a digital signal, making it easy to interface with microcontrollers like the ESP32.

Application in this Project:

In our Smart Windshield Monitoring System, the DHT11 sensor can be used to detect high humidity levels on the windshield, which can indicate fog formation. When the DHT11 detects a high humidity level, it sends a signal to the ESP32 microcontroller to activate the heater. This prevents fog from accumulating on the windshield, helping to maintain clear visibility for the driver, especially in damp or foggy weather conditions.

IR Sensor Module

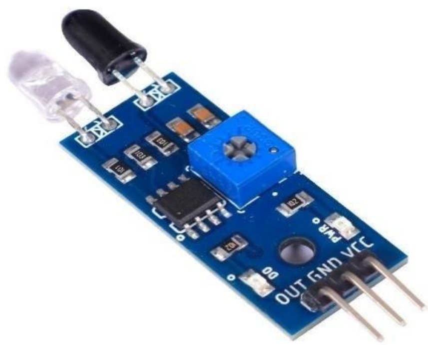

Fig 4.4 IR Sensor

An Infrared (IR) Module is commonly used to detect the presence of objects or changes in surface properties, including moisture. In the context of rain detection, an IR module can be configured with an IR transmitter and receiver positioned at an angle on a surface like a windshield. When rain droplets land on the windshield, they alter the refraction of the IR light, changing the reflection pattern received by the sensor. This variation signals the presence of rain.

Application in this Project:

In our Smart Windshield Monitoring System, the IR module can detect rain droplets on the windshield by sensing changes in the IR reflection pattern. When droplets are detected, the IR module sends a signal to the ESP32 microcontroller, which then activates the windshield wipers. This application offers a fast and non-intrusive way to detect rain, enhancing the automatic response of the wiper system and ensuring continuous visibility for the driver during wet weather.

ESP32 Cam Module

Fig 4.5 ESP32 Cam Modul

Microcontroller	ESP32 CAM
I/O Voltage	3.3V
Input voltage (nominal)	5V
DC Current per I/O Pin	40 mA
Power Supply Connector	USB port
Clock speed (Main Processor)	240 MHz
Memory (ESP 32)	520 KB SRAM, 4 MB Flash, 8 MB PSRAM

The ESP32-CAM Module **is** a low-cost camera module that integrates an ESP32 microcontroller with a built-in OV2640 camera sensor. It features Wi-Fi and Bluetooth connectivity, making it ideal for IoT applications where video capture, image processing, or wireless data transmission is required. The ESP32-CAM can capture images, stream video, and perform basic image processing tasks, with options for external storage via a microSD slot.

Application in this Project:

In your Smart Windshield Monitoring System, the ESP32-CAM module continuously monitors the windshield's visibility. By capturing real-time images or video, it can detect obstacles like dirt, rain, or fog that might impair the driver's view. The ESP32-CAM sends this data to the ESP32 microcontroller, which then triggers actions, such as activating the wipers or heater. This enables the system to provide real-time alerts and automated responses, enhancing visibility and safety for the driver in changing weather conditions.

Arduino UNO

Fig 4.6 Ardunio UNO

The Arduino UNO is a popular microcontroller board based on the ATmega328P microcontroller. It features 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, and a reset button. The board operates at 5V and is widely used for prototyping due to its simplicity, ease of use, and compatibility with various sensors and modules. It is an ideal choice for projects requiring straightforward digital and analog I/O operations.

Microcontroller	Arduino UNO
I/O Voltage	5V
Input voltage (nominal)	7-12V
DC Current per I/O Pin	20 mA
Power Supply Connector	Barrel Plug

Clock speed (Main Processor)	16 MHz
Clock speed (USB-Serial Processor)	16 MHz
Memory (Arduino UNO)	2KB SRAM, 32KB Flash, 1KB EEPROM

Application in this Project:

In this project, the Arduino UNO reads the signal from an IR sensor module to detect the presence of an object. When the IR sensor detects an object, the Arduino UNO activates a 5V relay, which then supplies power to the ESP32-CAM module. This setup allows the ESP32-CAM to be triggered only when an object is detected, enabling it to capture images or video as needed. By using the Arduino UNO to control the relay, the system conserves power, activating the camera only during specific events, making it ideal for applications like motion-activated cameras or automated detection systems.

CHAPTER 5

METHODOLOGY

Step1: an ESP32-CAM module for visibility monitoring, a rain sensor (or IR sensor) for rain detection, a DHT11 humidity sensor for fog detection, and possibly a BMP180 pressure sensor for atmospheric monitoring

Step2: Configure a display module (LCD or TFT) to project information and alerts to the driver.

Step3: Program the ESP32 to analyze the images to detect obstructions such as raindrops, dirt, or fog

Step4: Attach a rain sensor or IR module to detect moisture or rain droplets on the windshield. Configure the sensor to trigger the wiper control system automatically when rain is detected.

Step 5: Installed the DHT11 to monitor the humidity level, which is a key indicator for fog formation. If the DHT11 detects high humidity levels, the ESP32 triggers the heater system to clear condensation on the windshield, helping to maintain driver visibility in foggy conditions.

Step 6: The BMP180 sensor to detect drops in atmospheric pressure, which can indicate an increased chance of rain or fog. The ESP32 interprets data from the BMP180, allowing the system to prepare for potential rain by activating standby alerts or pre-activating the windshield functions.

Step 7: Develop control logic to integrate sensor data and automatically activate the wipers, heater, and display alerts:

- If rain is detected, activate the wipers.
- If fog is detected (high humidity), turn on the heater.
- If an obstruction is detected in the camera feed, alert the driver on the display

Hardware model development

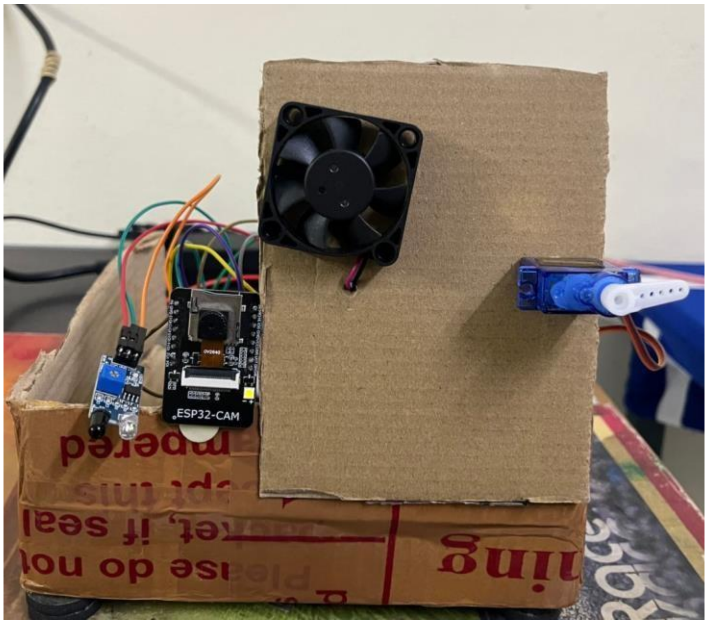

Fig 5.1

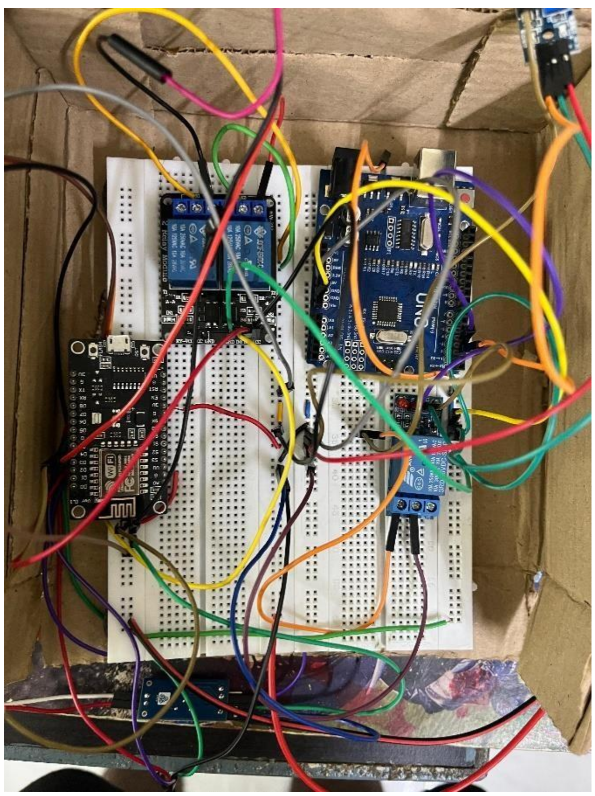

Fig 5.2

CHAPTER 6 RESULTS System Initialization

Purpose: Prepares the whole system for operation

Process: When powered on, all components (sensors, microcontrollers, displays, etc.) are set to a default "ready" state. Connections are verified, ensuring each part is responsive and primed for input.

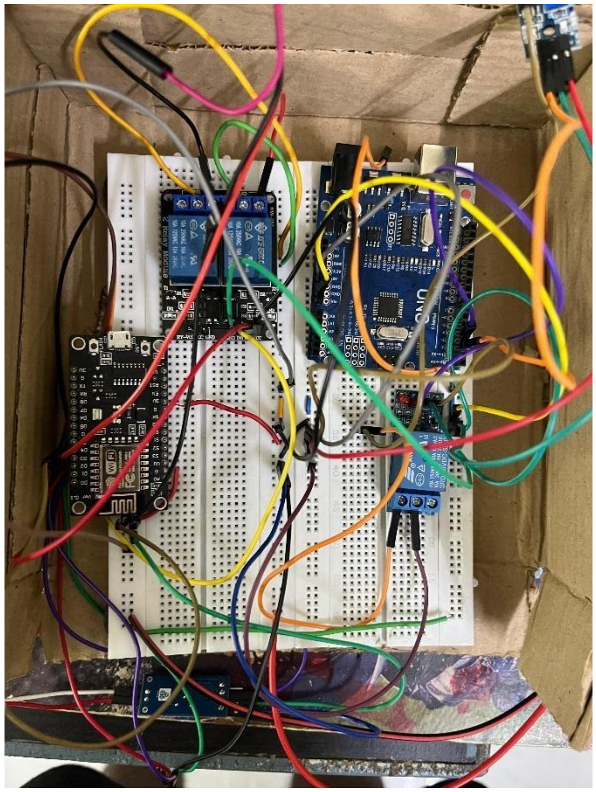

Fig 6.1 System initialization

Fig 6.2 OLED Initialization

Triggering IR module with Obstacle

Purpose: Activates IR module which triggers ESP CAM module.

Process: Photo diode in IR module Changes its resistance value proportional to the received IR light. Which triggers the ESP CAM module, then streams the environment activity to the car's dashboard.

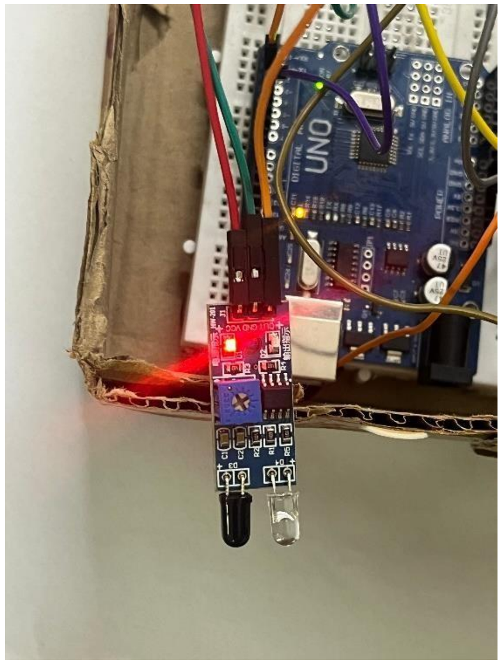

Fig 6.3 Output1 -Obstacles not detected

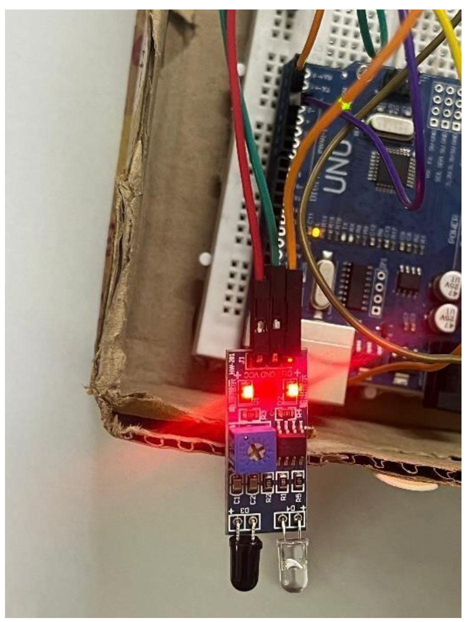

Fig 6.4 Output2- Obstacles detected

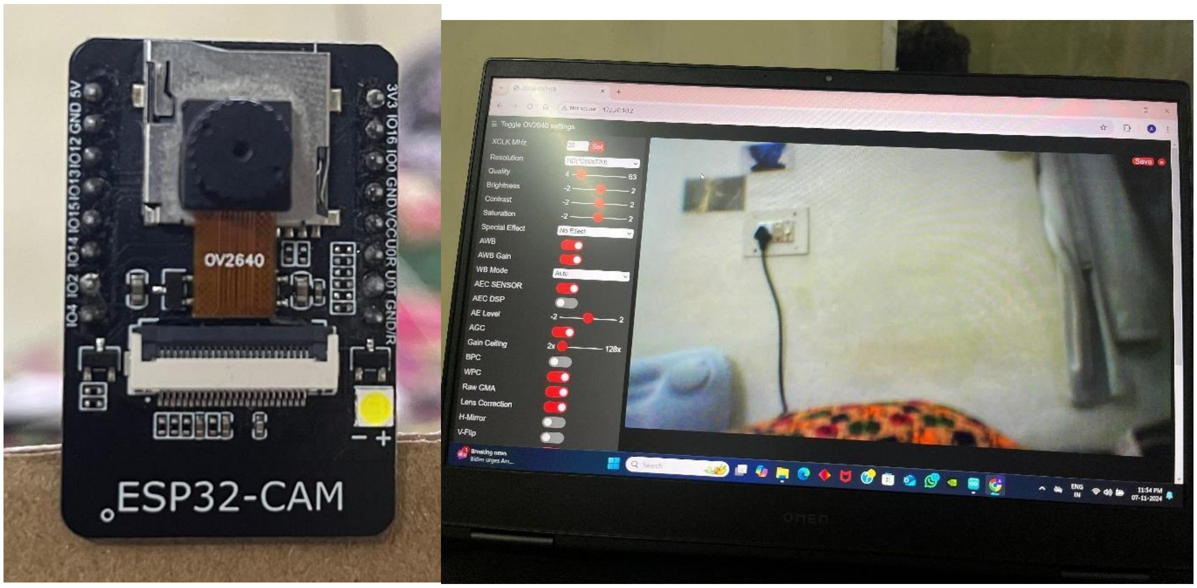

Fig 6.5 Dashboard Display ON.

Triggering DHT11 with Fog

Purpose: Senses the presence of fog in the environment and rectifies it.

Process: DHT11 uses a capacitive humidity sensor that measures the electrical resistance between two electrodes on a moisture-holding substrate. When water vapour is absorbed by the substrate. The DHT11 uses a thermistor, which is a variable resistor that changes its resistance with temperature.

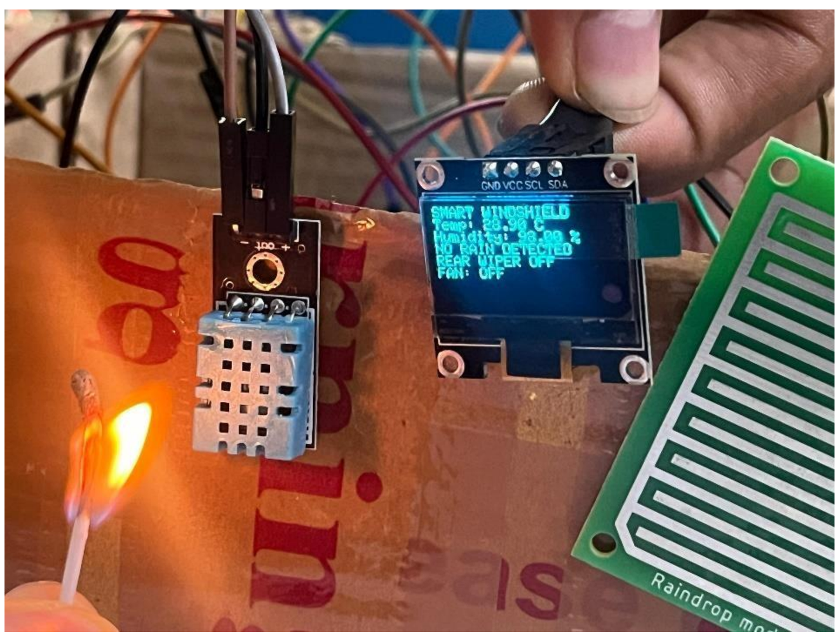

Fig 6.6 Activating DHT11 using varied Temperature

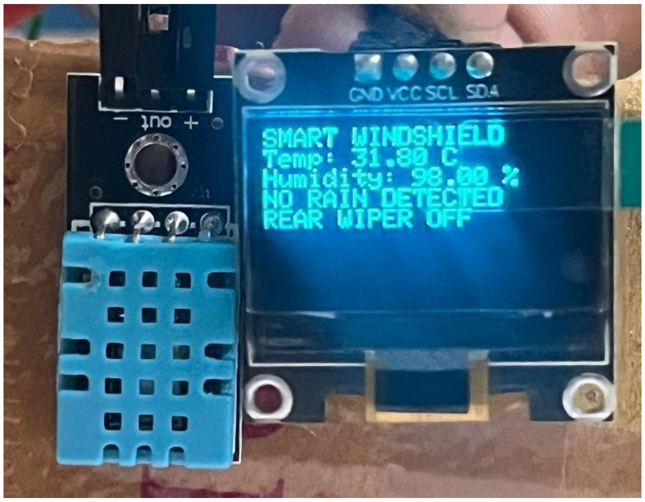

Fig 6.7 Output after temperature difference Triggering Rain sensor with Rain droplets

Purpose: Senses the presence of rain droplets in the environment and triggers the wiper mechanism to remove the rain droplets.

Process: The circuit within the rain sensor gets completed when rain droplets(Conductive in nature) land on the rain sensor .Which triggers the servo motor.

Fig6. 8 Presence of rain droplets

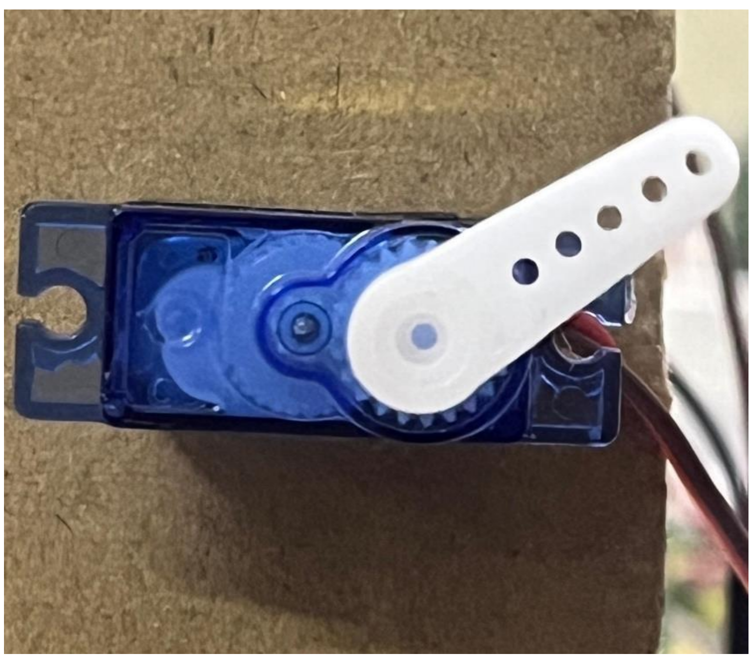

Fig 6.9 Actuating after presence of rain droplets

Software model development

The software model development for the Smart Windshield Monitoring and Control System, implemented in the Arduino IDE, begins by configuring the development environment. This includes installing board definitions for the ESP32 and ESP32CAM modules and integrating essential libraries such as WiFi.h for network connectivity, DHT.h for humidity sensing, and specific libraries for image capture and processing with the ESP32-CAM. These steps ensure compatibility and smooth operation across all system components.

The code is structured into several key modules. The initialization module sets up GPIO pins for sensors and actuators, including the camera, rain sensor, and DHT11 humidity sensor, enabling continuous data monitoring. In the sensor data acquisition module, data from the rain sensor, DHT11, and ESP32-CAM are processed to detect environmental changes, such as rain and fog, and potential obstructions. The decision-making module then applies algorithms to this data, automatically activating the wipers and heater based on weather conditions, while the dashboard update module provides real-time alerts to the driver, like “Wipers Activated” or “Heater On - Fog Detected.”

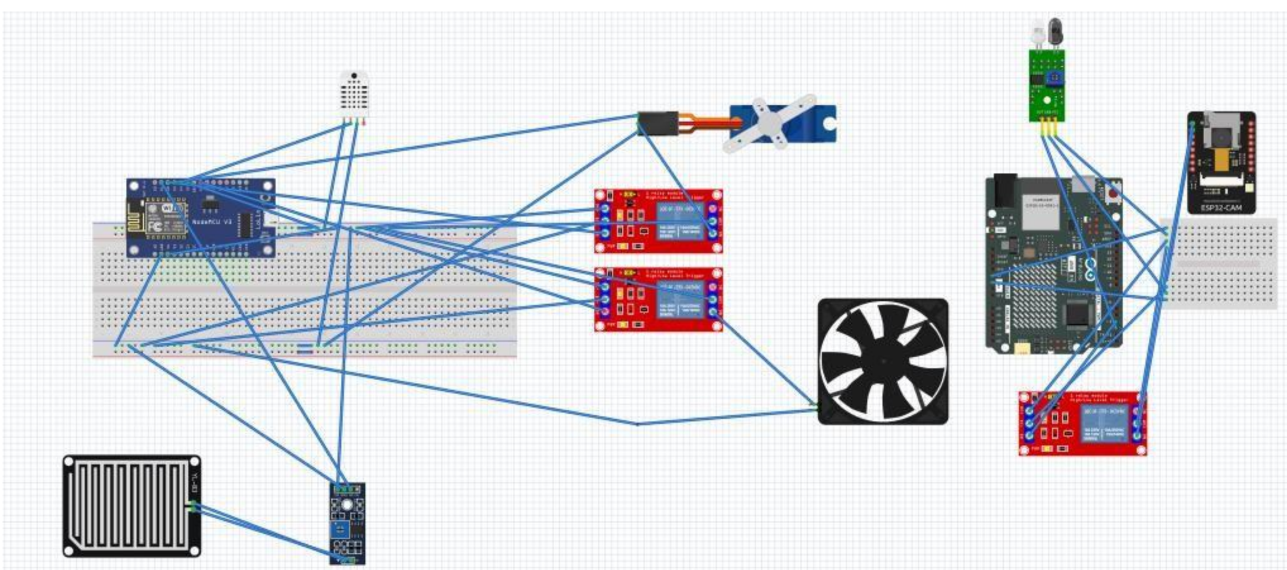

Fig 6.10 Simulation Circuit in Fritzing

Code Implementation

Language: C

Software: Aurdino IDE

Source Code

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <Servo.h>

// Pin Definitions for DHT Sensor and Fan Relay
#define DHTPIN D4       // DHT11 connected to D4 on ESP8266
#define FAN_RELAY_PIN D5 // Fan relay connected to D5 on ESP8266
#define DHTTYPE DHT11   // DHT 11 sensor type

// Pin Definitions for Rain Sensor, Wiper Relay, and Servo
#define RAIN_SENSOR_PIN D1    // Rain sensor connected to GPIO D1
#define WIPER_RELAY_PIN D2     // Wiper relay control pin connected to GPIO
D2
#define SERVO_PIN D3          // Servo signal pin connected to GPIO D3
DHT dht(DHTPIN, DHTTYPE);
Servo wiperServo; void setup()
{
  Serial.begin(115200);
// Initialize DHT sensor
dht.begin();
  // Initialize Fan Relay Pin
  pinMode(FAN_RELAY_PIN, OUTPUT);
  digitalWrite(FAN_RELAY_PIN, HIGH); // Ensure fan is off at the start
```

// Initialize Rain Sensor and Wiper Relay Pin

pinMode(RAIN_SENSOR_PIN,INPUT);

pinMode(WIPER_RELAY_PIN, OUTPUT);

digitalWrite(WIPER_RELAY_PIN, HIGH); // Wiper off (assumes active LOW relay)

// Attach Servo to its pin and set initial position

wiperServo.attach(SERVO_PIN); wiperServo.write(0);

// Initial position (no movement)

Serial.println("System initialized: Temperature control with fan and rain sensor with wiper.");

}

Temperature Control Section

if (isnan(temp)) {

Serial.println("Failed to read from DHT sensor!");

} else {

Serial.print("Temperature: ");

Serial.print(temp); Serial.println(" °C"); if (temp > 31) {

digitalWrite(FAN_RELAY_PIN, LOW); // Turns on the fan

Serial.println("Fan ON");

} else {

digitalWrite(FAN_RELAY_PIN, HIGH); // Turns off the fan

Serial.println("Fan OFF"); } }

Rain Sensor and Wiper Control Section

int rainDetected = digitalRead(RAIN_SENSOR_PIN);

if (rainDetected == LOW) { // Rain detected

Serial.println("Rain detected, activating wiper.");

digitalWrite(WIPER_RELAY_PIN, LOW); // Turns on wiper relay (active LOW)

// Servo wiper movement (sweeping back and forth)

for (int pos = 0; pos <= 90; pos += 10) {

```
wiperServo.write(pos);  // Moves from 0 to 90 degrees
delay(100);
    }
    for (int pos = 90; pos >= 0; pos -= 10) {
wiperServo.write(pos);  // Moves back from 90 to 0 degrees
delay(100);
    }
  } else { // No rain detected
    Serial.println("No rain detected, turning off wiper.");
digitalWrite(WIPER_RELAY_PIN, HIGH); // Turns off wiper relay
  }
delay(2000); // Delay before the next cycle
}
```

ARDUINO UNO Code

```
const int irSensorPin = 2;   // IR sensor output connected to pin 2
const int relayPin = 3;      // Relay input pin connected to pin 3
void setup() {
  Serial.begin(9600);          // Begin serial communication for debugging
pinMode(irSensorPin, INPUT);  // IR sensor as input pinMode(relayPin,
OUTPUT);    // Relay as output   digitalWrite(relayPin, LOW);  // Start with relay
off
} void loop() {   int irState =
digitalRead(irSensorPin);
  Serial.println(irState);      // Print IR sensor state for debugging
```

ESP32 CAM Control Section

```
  if (irState == HIGH) {       // Object detected    digitalWrite(relayPin,
HIGH);  // Turn on relay (and ESP32 CAM)
    Serial.println("Motor ON");
```

```
  } else {                       // No object detected     digitalWrite(relayPin,
LOW);   // Turn off relay (and ESP32 CAM)
    Serial.println("Motor OFF");
  }     delay(100);   //  Small  delay  for  stable
reading
}
```

CHAPTER 7

CONCLUSION

The Smart Windshield Monitoring and Control System offers a practical and effective solution for enhancing driver safety and visibility in adverse weather conditions. By integrating the ESP32-CAM with rain, humidity, and additional environmental sensors, the system automates the activation of wipers and the heater based on real-time conditions like rain, fog, and windshield obstructions. This automation minimizes the need for manual adjustments, allowing drivers to focus on the road while the system ensures a consistently clear windshield.

Testing demonstrated the system's responsiveness and reliability, with accurate detection and timely responses to changing environmental conditions. Overall, this IoT-driven approach reduces driver distractions, increases convenience, and promotes safer driving in varied weather, providing a valuable addition to modern vehicle technology.

REFERENCES

[1] J. -H. Lee, S. -H. Oh, S. -W. Oh, K. -S. Kim and S. -H. Kim, "An effective control of auto defog system to keep automobile windshield glass clear," ICCAS 2010, Gyeonggi-do, Korea (South), 2010, pp. 922-923, doi: 10.1109/ICCAS.2010.5669797.

[2] Gaikwad, Vijay & Kamtalwar, Neha & Karadbhajne, Harsh & Karmarkar, Mandar & Kendre, Harshvardhan & Ketkar, Ojas. (2023). Smart Car Using IoT and Image Processing. 1-7. 10.1109/ICCUBEA58933.2023.10392216.

[3] H. H. Qasim et al., "Enhancing Weather Monitoring: A Comprehensive Study Utilizing IoT, ESP32, Sensor Integration, and Blynk Platform," 2024 IEEE 10th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Bandung, Indonesia, 2024, pp. 156-161, doi: 10.1109/ICSIMA62563.2024.10675553.

[4] A. H. M. F. Elahi and M. S. Rahman, "Intelligent windshield for automotive vehicles," 2014 17th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2014, pp. 392-396, doi: 10.1109/ICCITechn.2014.7073112.

[5] J. W. Simatupang, A. M. Lubis and Vincent, "IoT-Based Smart Parking Management System Using ESP32 Microcontroller," 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Jakarta, Indonesia, 2022, pp. 305-310, doi: 10.23919/EECSI56542.2022.9946608.

[6] M. Ucar, H. M. Ertunc and O. Turkoglu, "The design and implementation of rain sensitive triggering system for windshield wiper motor," IEMDC 2001. IEEE International Electric Machines and Drives Conference (Cat. No.01EX485), Cambridge, MA, USA, 2001, pp. 329-336, doi: 10.1109/IEMDC.2001.939321

[7] S. Kanetkar, A. Rathore, K. Maheshwari, P. Dubey and A. Saxena, "Smart Helmet Wiper," 2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS), Bhopal, India, 2020, pp. 1-4, doi: 10.1109/SCEECS48394.2020.40

[8] P. Rai and M. Rehman, "ESP32 Based Smart Surveillance System," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-3, doi: 10.1109/ICOMET.2019.8673463.

[9] R. L. Paulraj, M. Tejasree, S. M. D and S. Ragasahithi, "Effective Weather Monitoring System using MQTT Protocol," 2023 8th International Conference on Communication and Electronics Systems(ICCES),Coimbatore,India,2023,pp.299-303,doi: 10.1109/ICCES57224.2023.10192800

[10] P. Abhilash Reddy, G. Sai Prudhvi, P J Surya Sankar Reddy, Dr. S. S. Subashka Ramesh. Automatic rain sensing car wiper, International Journal of Advance Research, Ideas and Innovations in Technology, ISSN: (Volume 4, Issue 5)