**AUTOMATIC PRECIPITATION DETECTION FOR CAR**

***Submitted by-***

*Ahona Saha(19BLC1179)*

*Aman Sahu (19BEC1180)*

*P. Uday Bhaskar (19BEC1398)*

***A project report submitted to***

*Dr.Guga Priya G*

*in partial fulfilment of the requirements for the course of*

**ECE4003: Embedded System Design**

****

**SCHOOL OF ELECTRONICS ENGINEERING(SENSE)**

**VELLORE INSTITUTE OF TECHNOLOGY CHENNAI - 600127**

***June 2021***

**BONAFIDE CERTIFICATE**

This is to certify that this project report entitled “**AUTOMATIC****PRECIPITATION DETECTION FOR CAR**” is a bonafide work of AHONA SAHA 19BLC1179, P UDAY BHASKAR19BEC1398, AMAN SAHU19BEC1180 who carried out the Project work under my supervision and guidance for ECE4003-Embedded System Design.

Dr.GugaPriya G

Assistant Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SERIAL NUMBER** | **TITLE** | **PAGE NUMBER** |
| 1 | ABSTRACT | 3 |
| 2 | ACKNOWLEDGMENT | 4 |
| 3 | INTRODUCTION | 5 |
| 4 | SYSTEM IMPLEMENTATION AND ANALYSIS  5.1 SYSTEM IMPLEMENTATION  5.2 WORKING PRINCIPLE | 6-7  6  7 |
| 5 | CODE ALGORTITHM AND ANALYSIS  6.1 ALGORITHM  6.2 CODE  6.3 CODE ANALYSIS | 8-12  8  9-11  12 |
| 6 | FUTURE WORK | 15121 |
| 7 | REFERENCES | 16 |

**ABSTRACT**

This project is about an **automatic precipitation detection for car** made using an Arduino Uno, Rain Sensor, LDR and LED.

This project is designed to build a car wiper that automatically detects the rainfall intensity and regulates the frequency of wiper operation. It is built, using Arduino UNO board. A rain sensing module is used for measuring the intensity of rainfall. And a servo motor is used for controlling the wiper movements. An LCD module is also attached to the controller for displaying the rainfall intensity. By measuring the amount of rainfall, controller will adjust the speed of servo motor. Servo is controlled by generating PWM signal at its signal line.

**ACKNOWLEDGEMENT**

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr GugaPriyaG, Assistant Professor (Sr.), School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to Dr Sivasubramanian A, Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

 We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

**INTRODUCTION**

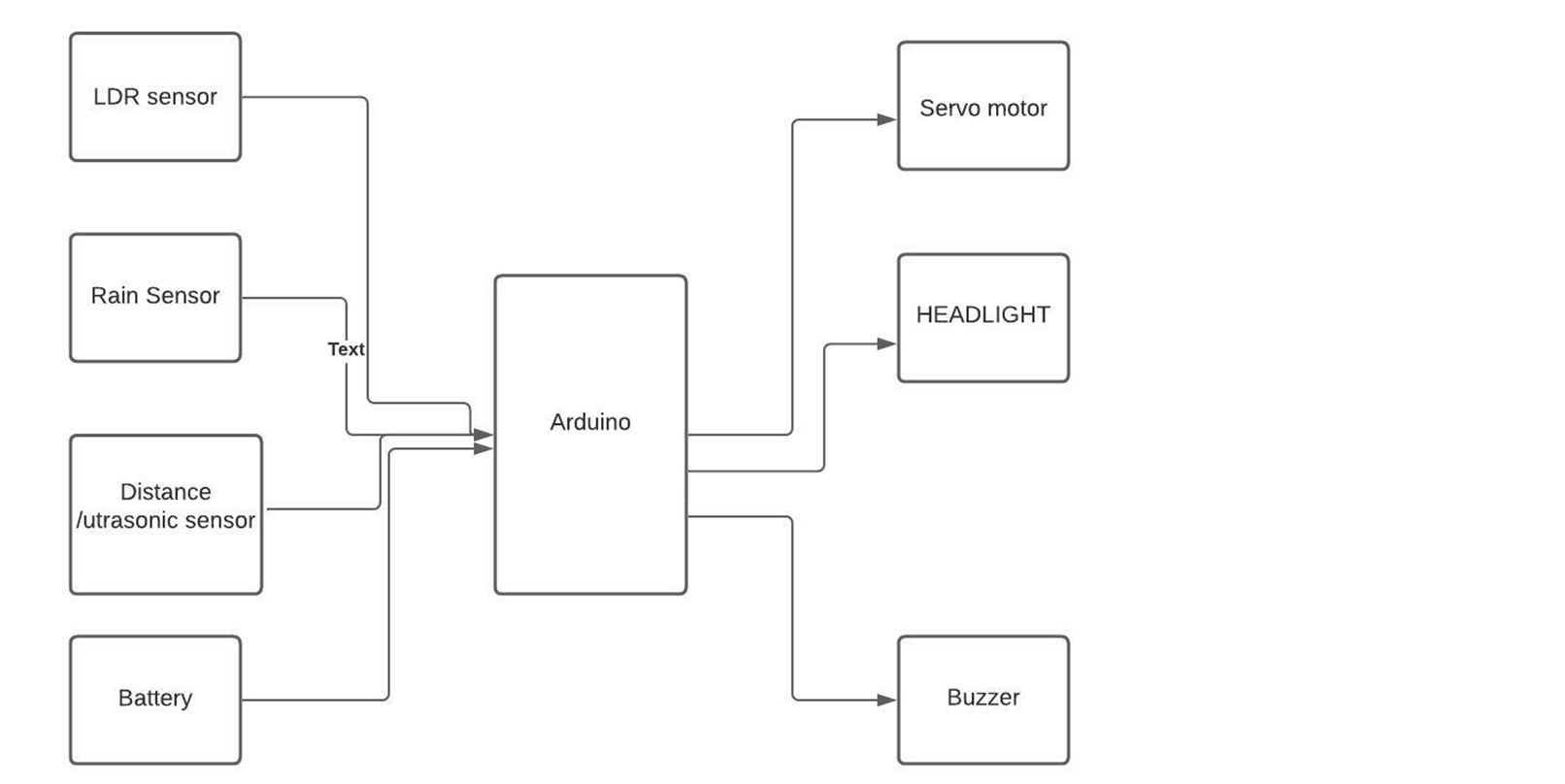
With advancement of technology things are becoming simpler and easier for us. Automation is the use of control systems and information technologies to reduce the need for human work in the production of goods and services.

The issue of driver's safety is of great importance in today's automotive industry. In many cases, a lack of proper vision is responsible for accidents during heavy rainfall. In many cases, manual errors like not increasing the speed of the wiper by the driver lead to accidents. Today's car wipers work on the principle of manual switching. In this paper, we proposed an automatic rain sensing wiper system that detects rain and starts automatically and stops when the rain stops. The automatic rain sensing car wiper system is not only automatic but also intelligent. The wiper system detects the rainfall automatically and starts itself. The Wiper system is also intelligent. When the droplets of rainfall on the sensor, the sensor detects the intensity and the speed of the wiper are automated accordingly. The higher speed of rotation indicates the higher rainfall. There will be no need for manual intervention for controlling wiper. In this project, we use Arduino along with a rain sensor, an LCD 16x2 module, and a servo motor. The moisture is measured via analog output pins which are present in the rain sensor, the wiper starts rotating when a threshold of moisture is exceeded. The module used here is completely based on LM393 op-amp. The information sensed by the rain sensor is sent to Arduino. The Arduino is an Atemga8 based microcontroller board. Interactive electronic devices can be designed and created by using Arduino, which is a platform to develop the working of electronic devices. It consists of an on-board power supply and a USB port to communicate with the pc. The collected information from the rain sensor is processed and analysed by Arduino and it further controls the servo motor based on the processed information. The information about the intensity of the rainfall and speed of the wiper is informed to the driver by means of a 4-bit LCD module which is kept near the driver's seat. The rain sensor is kept at the side of the windshield, outside the car. The rain sensor is connected to the servo motor. The blades of the wiper are connected to the servo motor. All the devices are connected to Arduino which is connected to the power source inside the car.

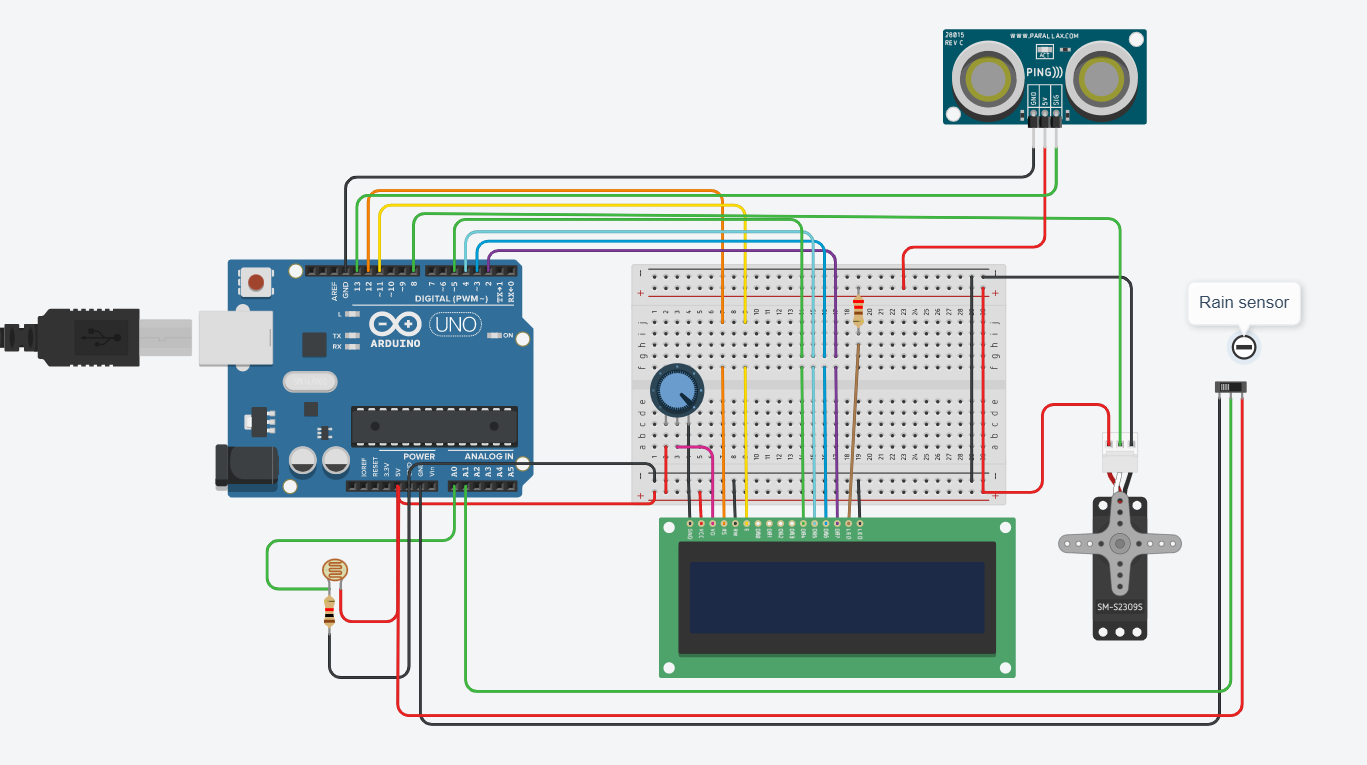
**AUTOMATIC PRECIPITATION DETECTION FOR CAR**

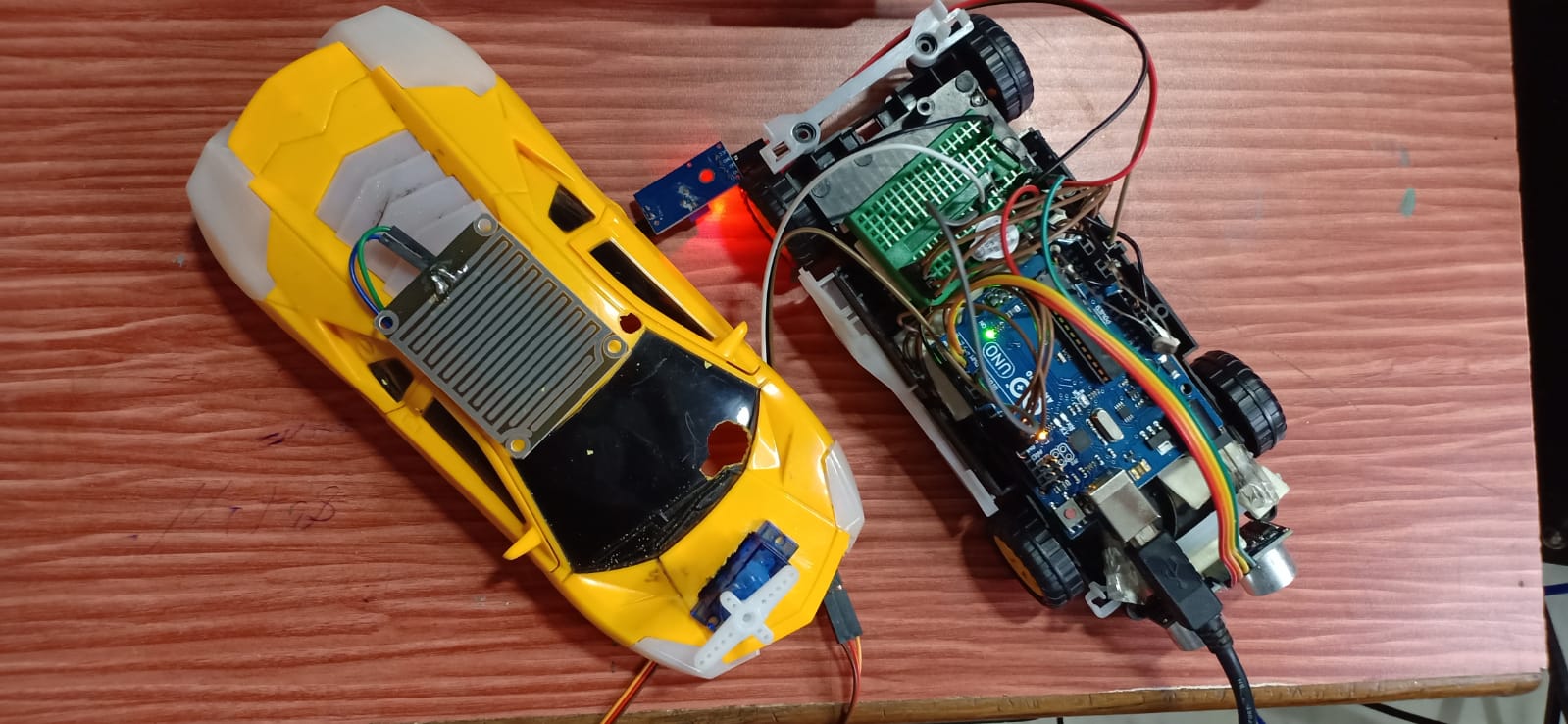
**BLOCK DIAGRAM AND ANALYSIS**

This section describes system implementation and results with inferences.

****

**5.1 SYSTEM IMPLEMENTATION:**



****

**5.2 WORKING PRINCIPLE:**

This working principle of this project is mainly based on the implementation of 3 main sensor modules namelr, Rain Sensor, Ultrasonic Sensor, LDR.The project consists of 3 main functions

1. AUTOMATIC WIPER SYSTEM: When the rain drops fall on the rain sensor module the resistance between the anode and cathode change accordingly which results in the variable sensor output value. Based on the value of the sensor we have defined 3 wiper modes, SLOW, MEDIUM and FAST. In the SLOW condition the delay between the servo motor working is long around 5 secs. In the MEDIUM setting the delay between the servo motor working is a bit less around 2-3 secs. In the FAST setting the delay between the servo motor working is minimal around 1/2- 1 sec.
2. OBSTACLE DETECTION: We use the ultra-sonic sensor to send and receive pulse through its trig and echo pins and based on it calculate approximate distance of the obstacle, once the obstacle is at or near 30 cm mark , the buzzer starts to sound indicating the obstacle.
3. AUTOMATIC HEADLIGHT: Using the light dependent resistor(LDR) we calculate the intensity of the ambient light and then according to it turn led. It also has 2 modes low and light. If the ambience is not very dark but dim, then only 2 headlights are switched on. But if the ambience is very dim, all 4 of the headlights will be switched on.

All of these functions when embedded into a vehicle can help in the time of rainfall.

**6.1 ALGORITHM**

1. Include the header file of servo and define its pin, echo and trig pins.
2. Now define the pins of for Ldr, Led and for buzzer.
3. In setup function, we have to define mode of each pin as input or output. And define servo pin also.
4. Begin the serial with baud rate 9600.
5. In loop function, define duration and distance as variable. Duration will be the input pulse width and distance will be the distance to the obstacle in centimeters.
6. Generate Output pulse with 1ms width on trigPin and Measure the pulse input in echo pin. Then distance is half the duration divided by 29.1.
7. if distance less than 0.5 meter and more than 0 (0 or less means over range) then buzz the buzzer else don’t buzz it.
8. if Ldr reading is less than 200 and more than 50. Then serial prints Its DARK, turn on the 2 LEDs and if Ldr reading is less than 50 then serial prints Its Very DARK, turn on the all the LEDs.
9. if rain sensor reading is more than 800 then serial prints “no rain”, if rain sensor reading is less than 800 and more than 600 then serial prints “low rain”, if rain sensor reading is less than 600 and more than 460 then serial prints “High rain” and if rain sensor reading is less than 460 then serial prints “Very High rain”.

**6.1.1APIs Used:**

**(APIs for working with the LCD)**

**LiquidCrystallcd(rs, enable, d4, d5, d6, d7) :**- To use the LiquidCrystal library, create a named LiquidCrystal object. In this case the object is lcd.

**·      lcd.begin(cols, rows) :-** Initializes the interface to LCD, and specifies dimensions.

**o   cols:** Initialize column position i.e., starting column to display the first character/value.

**o   rows:** Initialize row position i.e., starting row to display/print the first character.

**·      lcd.setCursor(col, row) :-** Position the LCD cursor to set row and column value across 16x2.

**-        cols:** Set column position i.e., column to display the character/value input.

**-        rows:** Set row position i.e., row to display/print the character input.

**·      lcd. clear() :-** Clears the LCD screen and positions the cursor on upper-left corner.

**·      lcd.print(data)/lcd.write(data) :-** The character to write to the display.

**-        data:** The value/character that needs to be displayed onto the LCD.

**(APIs for working with the Servo)**

**·    servo.attach(pin) :-** Check whether the Servo variable is attached to the Arduino pin. Return true if the servo is attached topin; false otherwise.

**-        pin:** the number of the Arduino pin that the servos attached to.

**·   servo.write(angle) :-** Writes a value to the servo, controlling the shaft accordingly.

-        **angle:** The angle value to write to the servo, from 0 to 180 degrees.

**6.2 CODE:**

#include <Servo.h>

Servo myservo;

//distance sensor....

#define echoPin 2

#define trigPin 3

long duration;

int distance;

//LDR...............

constintldrPin = A0;

//LED......

constint ledPin1=9;

constint ledPin2=10;

constint ledPin3=11;

constint ledPin4=12;

//buzzer..........

intbuzzPin=4;

intpos = 0;

intsensorValue = 0;

void setup() {

Serial.begin(9600);

pinMode(trigPin,OUTPUT);

pinMode(echoPin,INPUT);

pinMode(buzzPin, OUTPUT);

pinMode(ledPin1, OUTPUT);

pinMode(ledPin2, OUTPUT);

pinMode(ledPin3, OUTPUT);

pinMode(ledPin4, OUTPUT);

pinMode(ldrPin, INPUT);

myservo.attach(7);

Serial.println("Intelligent Rain");

Serial.println("Sensing CarViper");

delay(3000);

Serial.println("Rainfall");

Serial.println("Intensity-");

}

void loop()

{

//.............................................

// Duration will be the input pulse width and distance will be the distance to the obstacle in centimeters

int duration, distance;

// Output pulse with 1ms width on trigPin

digitalWrite(trigPin, HIGH);

delay(1);

digitalWrite(trigPin, LOW);

// Measure the pulse input in echo pin

duration = pulseIn(echoPin, HIGH);

// Distance is half the duration devided by 29.1 (from datasheet)

distance = (duration/2) / 29.1;

// if distance less than 0.5 meter and more than 0 (0 or less means over range)

if (distance <= 50 && distance >= 0) {

// Buzz

digitalWrite(buzzPin, HIGH);

} else {

// Don't buzz

digitalWrite(buzzPin, LOW);

}

//......................

intldrStatus = analogRead(ldrPin);

if (ldrStatus<= 200&&ldrStatus> 50) {

digitalWrite(ledPin1, HIGH);

digitalWrite(ledPin2, HIGH);

digitalWrite(ledPin3, LOW);

digitalWrite(ledPin4, LOW);

Serial.print("Its DARK, Turn on the LED : ");

Serial.println(ldrStatus);

} else if(ldrStatus< 50){

digitalWrite(ledPin1, HIGH);

digitalWrite(ledPin2, HIGH);

digitalWrite(ledPin3, HIGH);

digitalWrite(ledPin4, HIGH);

Serial.print("Its VERY DARK, Turn on the LED : ");

Serial.println(ldrStatus);

}

else{

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, LOW);

digitalWrite(ledPin3, LOW);

digitalWrite(ledPin4, LOW);

Serial.print("brightg: ");

Serial.println(ldrStatus);

}

//.rain...................................

sensorValue = analogRead(A3);

Serial.println(sensorValue);

if(sensorValue>800){

myservo.write(180);

Serial.print(" NO RAIN ");

delay(1000);

}

if(sensorValue<=800 &&sensorValue>600){

Serial.print("Low Rain");

for (pos = 180; pos>= 0; pos-=1) {

myservo.write(pos);

delay(3);

}

for (pos = 0; pos<= 180; pos+=1) {

myservo.write(pos);

delay(3);

}

delay(2000);

}

if(sensorValue<=600 &&sensorValue>460){

Serial.print("AMOUNT: MEDIUM ");

for (pos = 180; pos>= 0; pos-=1) {

myservo.write(pos);

delay(3);

}

for (pos = 0; pos<= 180; pos+=1) {

myservo.write(pos);

delay(3);

}

delay(1000);

}

if(sensorValue<460){

Serial.print("AMOUNT: HIGH ");

for (pos = 180; pos>= 0; pos-=1) {

myservo.write(pos);

delay(3);

}

for (pos = 0; pos<= 180; pos+=1) {

myservo.write(pos);

delay(3);

}

delay(100);

}

}

**6.3 CODEANALYSIS:**

Number of Lines: 110

**FUTURE WORK**

**• Design and integration of an mobile application to notify user**

**• Voice operated system can be implemented**

**REFERENCES:**

* <https://www.arduino.cc/en/Guide/ArduinoUno>
* <https://www.arduino.cc/reference/en/libraries/servo/>
* <https://www.arduino.cc/reference/en/language/functions/communication/serial/>