WEATHER REPORTING SYSTEM FOR VEHICLES

ICT 305 2.0

Embedded System

AS2020965

D.P.S Dahanayaka

CONTENT

PURPOSE

The purpose of this project can be discussed under the several main points.

Enhancing Vehicle Safety:

- Monitor environmental conditions to provide real-time information to drivers.
- Enable early detection of adverse weather conditions, enhancing overall road safety.
- Facilitate informed decision-making for drivers by providing relevant weather data.

Optimizing Vehicle Performance:

- Collect and analyze weather-related data to optimize vehicle performance under different conditions.
- Provide insights into how weather factors such as temperature, humidity, and pressure might affect the vehicle's operation.

Research and Development:

- Contribute valuable data to research in the field of transportation and weather-related impacts.
- Collaborate with researchers and institutions to improve our understanding of how weather affects vehicles.

INTRODUCTION

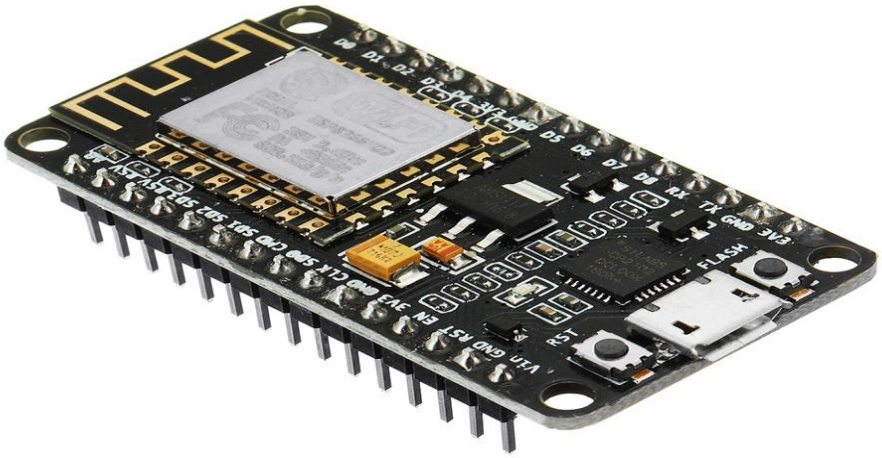
In today's dynamic and fast-paced world, advancements in technology continue to reshape various aspects of our daily lives. One such area that has witnessed significant transformation is the automotive industry. As vehicles become more connected and smarter, the integration of real-time environmental data becomes increasingly crucial. In response to this need, our project introduces a comprehensive Weather Reporting System for Vehicles, aimed at enhancing safety, optimizing performance, and providing intelligent insights for drivers.

Adverse weather conditions pose a considerable challenge to road safety and vehicle performance. Unforeseen weather events, such as sudden rain showers, fluctuations in temperature, and varying light conditions, can significantly impact driving conditions. Recognizing the importance of weather-aware driving, our project seeks to address these challenges by creating a sophisticated system that collects, processes, and disseminates real-time weather data directly to vehicles.

this project we are using various types of sensors, and we collect temperature, humidity value, rainfall, and pressure such as valuable data. We collect data output from display and clearly show data from display and send to the data from web server. After analyzing the data from using web server. By sending data to a mobile app through the web server, the user can easily analyze the data and come to conclusions.

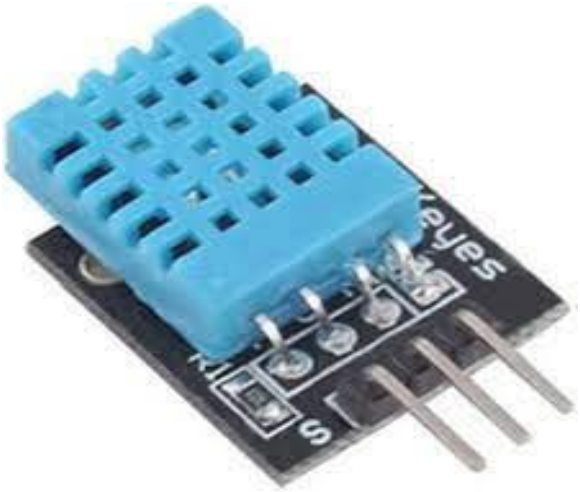
COMPONENTS

- **Nodemcu board**



NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added.

- **DHT11 Sensor**



The DHT11 sensor is a widely used and cost-effective device designed for measuring temperature and humidity levels in the surrounding environment. It is a digital sensor known for its simplicity, reliability, and ease of integration into various electronic projects.

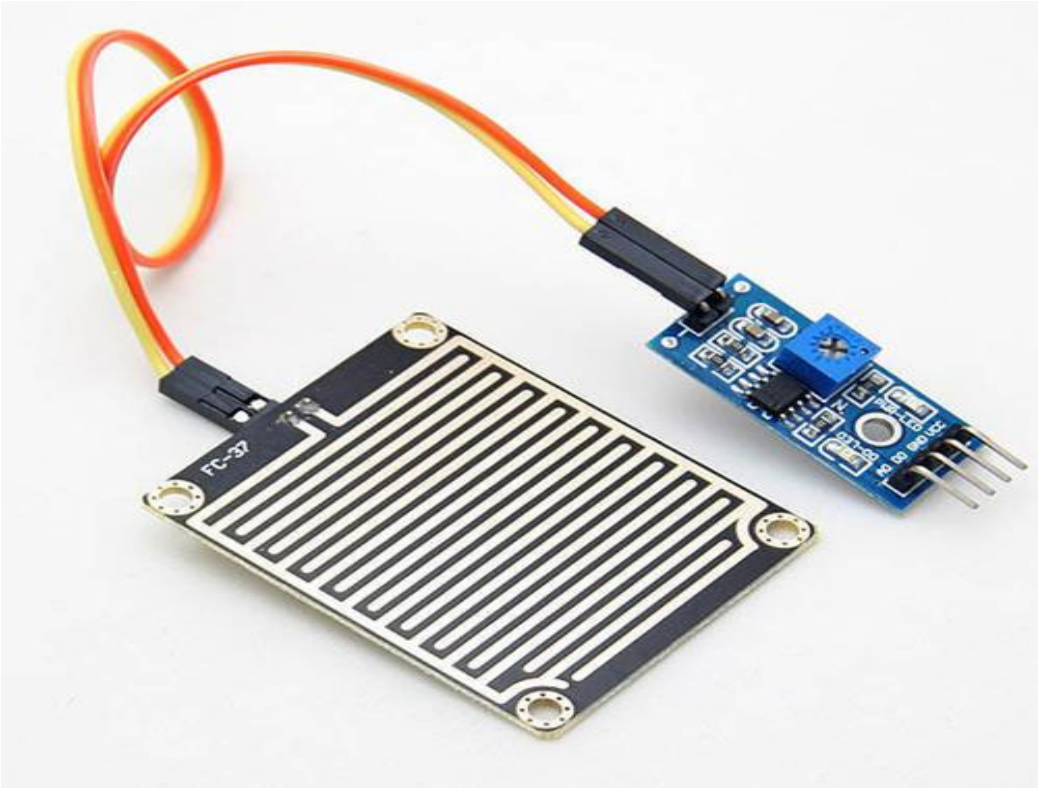
Developed by Aosong (Guangzhou) Electronics Co., Ltd., the DHT11 sensor provides a convenient solution for monitoring ambient conditions, making it popular among hobbyists, students, and professionals in the field of electronics and IoT (Internet of Things).

- **LDR Sensor**

The Light Dependent Resistor, commonly known as LDR or photoresistor, is a passive electronic component that exhibits a change in resistance based on the intensity of light it is exposed to. LDRs are widely used in various applications where detection or measurement of light levels is crucial. The fundamental principle behind LDR operation is the variation of its resistance with the incident light, making it a valuable component in electronic circuits and sensor systems.

LDRs are specifically designed to respond to changes in light levels. The resistance of the LDR decreases as the intensity of light increases and vice versa. This property makes it suitable for detecting ambient light conditions.

- **Rain Sensor**



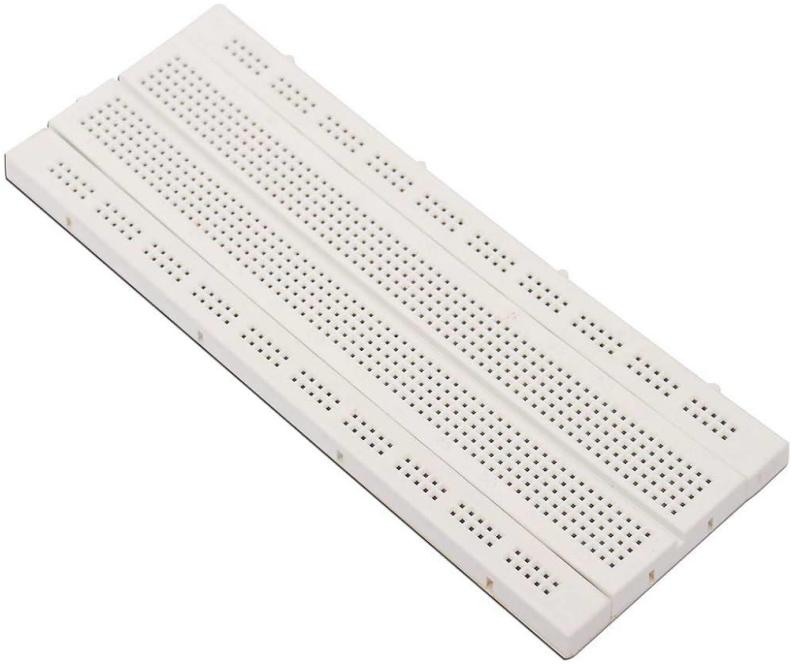
Rain sensors are devices designed to detect the presence of rainfall or water. These sensors play a crucial role in various applications, including weather monitoring systems, smart irrigation systems, and automotive technologies. The primary function of a rain sensor is to provide an electronic signal in response to the detection of rain, enabling automated systems to respond accordingly.

- **LCD display with I2c module.**

The LCD (Liquid Crystal Display) with an I2C module is a common and convenient way to display information in Arduino and other microcontroller projects. The I2C (Inter-Integrated Circuit) module simplifies the wiring and allows you to control the display using only two wires (SDA and SCL) instead of several individual wires.

To use an LCD with an I2C module, you'll typically need a library to simplify the code. For Arduino, the "LiquidCrystal_I2C" library is commonly used. You can install it via the Arduino Library Manager.

- **Breadboard**



A breadboard is a crucial tool in electronics prototyping and experimentation. It provides a platform for engineers, hobbyists, and students to build and test electronic circuits without the need for soldering.

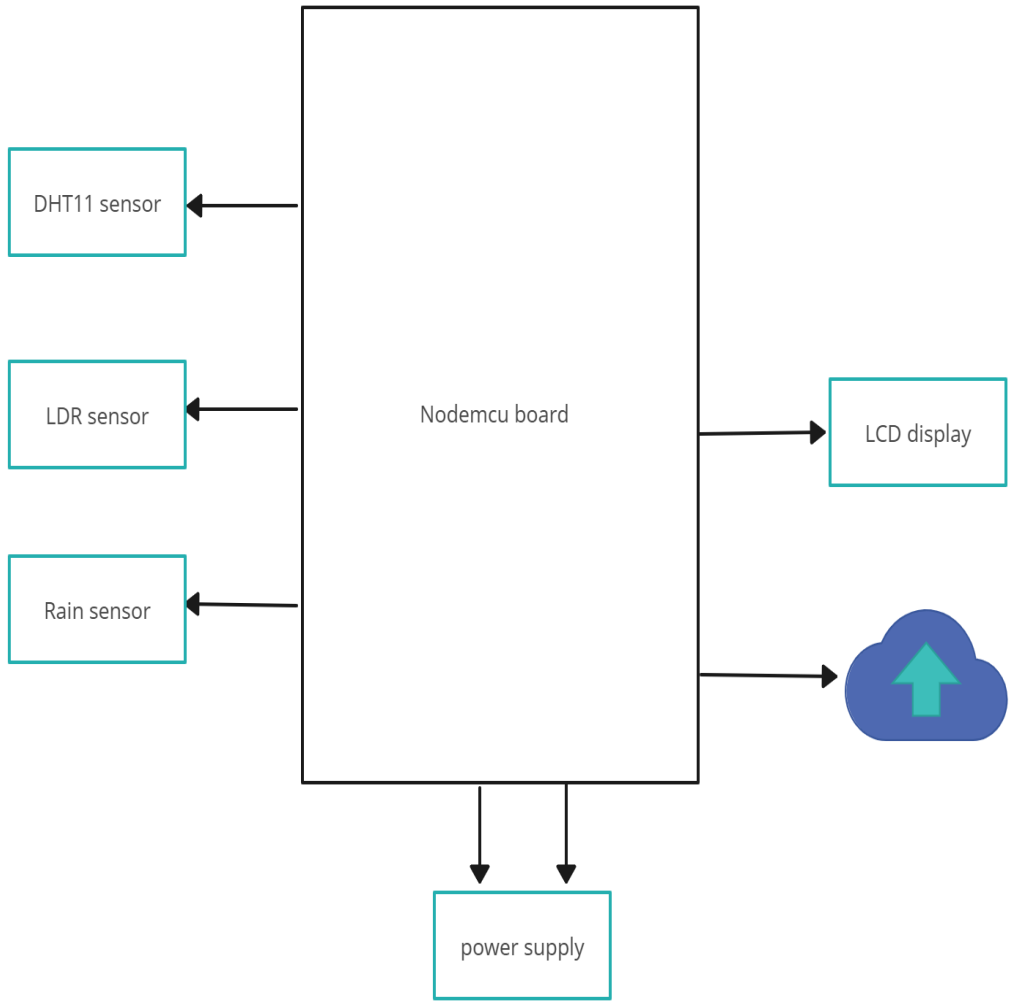
- **jumper wires**

Jumper wires are essential components in electronics and prototyping. They are short, flexible wires with connectors at both ends, typically used to make temporary connections on a breadboard or between components on a circuit.

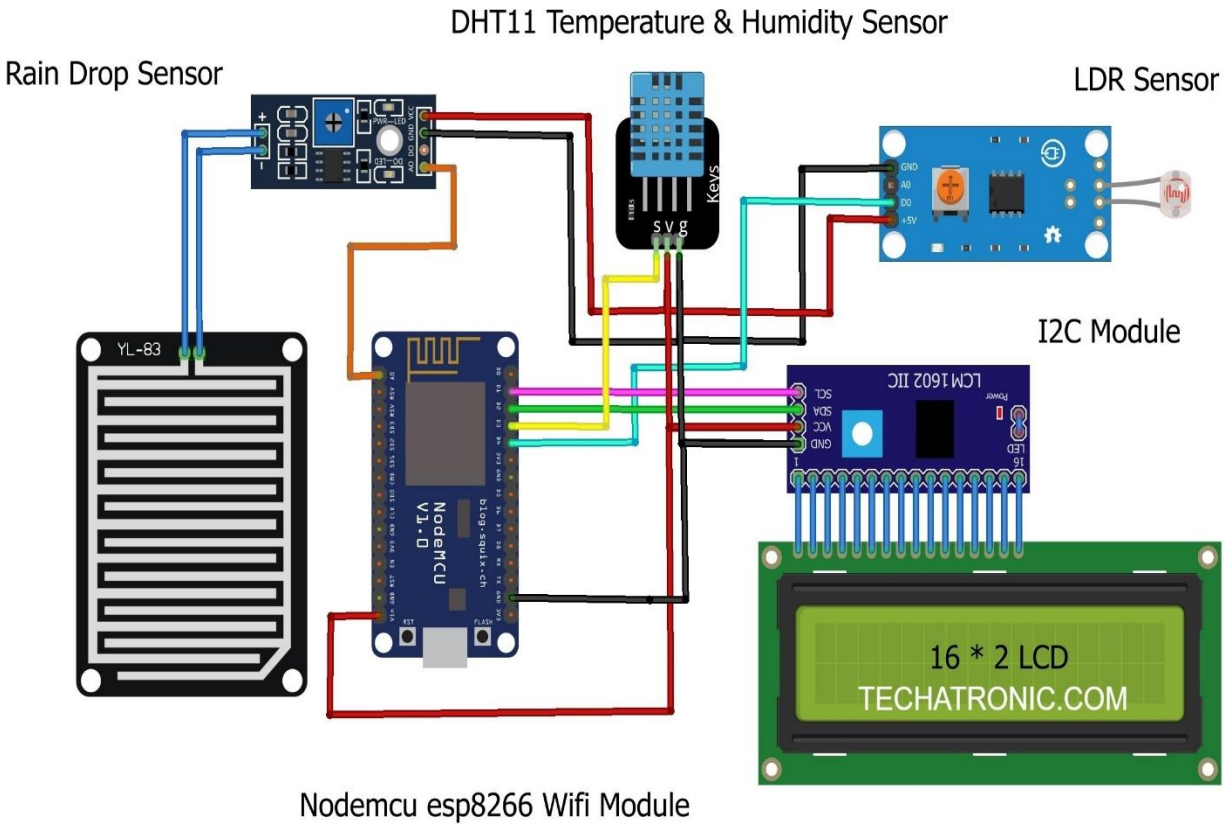
We are using different types of jumper wire male to male and male to female both we get the connect component. Jumper wires come in various lengths, from very short (a few centimeters) to longer ones suitable for reaching across a breadboard. They are often color-coded to help identify and organize connections. Common colors include red, black, yellow, green, and blue.

Design Overview

- **component diagram**



- **circuit diagram**



CODE

```
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <ThingSpeak.h>
#include <Wire.h>       // Include the Wire library for I2C communication
#include <LiquidCrystal_I2C.h>  // Include the LiquidCrystal I2C library

// Replace with your network credentials
const char* ssid = "Redmi12";
const char* password = "12345678";

// DHT sensor settings
#define DHTPIN D4       // DHT data pin (e.g., D2)
#define DHTTYPE DHT11  // DHT sensor type
DHT dht(DHTPIN, DHTTYPE);

// LDR sensor settings
const int ldrPin = A0;  // LDR sensor connected to analog pin A0

// Rain sensor settings
const int rainSensorPin = D3;  // Rain sensor connected to digital pin D1

// ThingSpeak settings
const char* server = "api.thingspeak.com";
const char* writeAPIKey = "WGFI8LV88T39KIWY";

WiFiClient client;

// LCD display settings
LiquidCrystal_I2C lcd(0x27, 20, 4);  // Change the I2C address and display size if needed

void setup() {
  Serial.begin(9600);
  delay(10);
```

```
  // Initialize the LCD display
  lcd.init();
  lcd.backlight();

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  dht.begin();
  ThingSpeak.begin(client);
}

void loop() {
  // Read data from the DHT sensor
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Read data from the LDR sensor
  int ldrValue = analogRead(ldrPin);

  // Read data from the rain sensor
  int rainValue = digitalRead(rainSensorPin);

  // Print data to the serial monitor
  Serial.print("Temperature: ");
  Serial.println(temperature);
  Serial.print("Humidity: ");
  Serial.println(humidity);
```

```
  Serial.print("LDR Value: ");
  Serial.println(ldrValue);
  Serial.print("Rain State: ");
  Serial.println(rainValue == HIGH ? "Rain Detected" : "No Rain");

  // Display data on the LCD
  lcd.clear();
  lcd.setCursor(3, 0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print("C");
  lcd.setCursor(0, 1);
  lcd.print("LDR:");
  lcd.print(ldrValue);
  lcd.setCursor(8, 1);
  lcd.print("Rain:");
  lcd.print(rainValue == HIGH ? "Yes" : "No");

  // Send data to ThingSpeak
  ThingSpeak.setField(1, temperature);
  ThingSpeak.setField(2, humidity);
  ThingSpeak.setField(3, ldrValue);
  ThingSpeak.setField(4, rainValue);
```

```

  int status = ThingSpeak.writeFields(2336604, writeAPIKey);

  if (status == 200) {
    Serial.println("Data sent to ThingSpeak successfully.");
  } else {
    Serial.println("Failed to send data to ThingSpeak.");
  }

  delay(5000); // Delay for 5 seconds before reading the sensors again
}


```

Budget

Component name	price
nodemcu esp8266	1100
DHT11 sensor	250
Rain sensor	150
LDR sensor	150
LCD display	300
I2c module	250
Jumper wires	200
breadboard	250
Micro usb cable	450
Total	**3100**

FUTURE IMPLEMENTATION

- Connecting some sensors that can pick up valuable weather information like wind speed and pressure(wind sensor and bmp180 sensor)
- Connecting the gsm module so that the mobile phone receives a message in case of sudden weather change.
- Providing a better experience to users by improving the mobile app and web server
- Being able to view information more clearly and accurately by connecting a large display.
- Connecting transistors, diodes and resistors to the circuit to get more accurate information.
- We can add the alarm sound and some voice alarms with the detection. It will be helpful too in case of emergency weather.

ISSUES

- It takes more time to transfer data when connected to the internet through the nodemcu board.
- Sometimes the sensor is not working properly, and the data is delayed.
- Unable to get more accurate data due to not connecting transistors, diode and resistor.
- Difficulty in purchasing some sensors and necessary accessories.
- Due to the use of a small display, it is difficult to clearly represent the data obtained.

DISCUSSION AND CONCLUSION

- **Discussion**

  The main target of this project is to report and analyze weather information and this is more suitable for weather and storm chasing vehicles. Throughout the project, variations in sensor readings were observed, prompting a closer examination of calibration methods. I try to minimize mistakes, occasional discrepancies in data collection underscored the importance of ongoing calibration and maintenance procedures.

  Sometimes the data obtained by this sensor is very unexpected. I often got these unexpected results when taking temperature and precipitation values. In those cases, he tried to get the data vary carefully.

- **Conclusion**

  The Weather Reporting System for Vehicles has successfully addressed its primary objectives of providing real-time, reliable weather information to enhance the driving experience. Consider advancements in sensor technology, communication protocols, or additional functionalities.

  So, the system does not work for the most accurate values. From future implementation we can make our system better.

REFERENCE

- **Espressif Systems. (2023). NodeMCU Documentation:** https://www.espressif.com/en/support/documents/technical-documents

- **ThingSpeak. (2023). ThingSpeak API Documentation:** https://www.mathworks.com/help/thingspeak/

- **datasheets and product manuals sensors** **https://www.sensorelectronics.com/data_sheets.html**

- **LCD display module datasheet** **https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet**