# WEATHER MONITORING SYSTEM

## MICROCONTROLLER BASED SYSTEM DESIGN - 24ERT2115

**Submitted by:**
Nithya Zachariah:44

# INTRODUCTION

This project aims to develop a real-time weather monitoring system using the ESP8266 NodeMCU microcontroller interfaced with multiple sensors — the DHT11 for temperature and humidity, the BMP180 for atmospheric pressure, and an LDR sensor for measuring ambient light intensity. The system acquires environmental data from these sensors and displays the processed output on an LCD screen, providing a user-friendly local interface for instant weather information. This low-cost, compact embedded setup offers an accessible solution for monitoring crucial weather parameters useful in agriculture, environmental observation, and daily weather awareness without requiring complex or expensive instruments.

# OBJECTIVES

- To design and implement a cost-effective, embedded weather station using ESP8266 NodeMCU and sensors DHT11, BMP180, and LDR.
- To accurately measure real-time temperature, humidity, atmospheric pressure, and light intensity from the environment.
- To interface the sensors with the NodeMCU microcontroller and program it to process and display weather data on an LCD screen.
- To develop a standalone system that provides instant visual weather feedback suitable for local monitoring purposes.

# BLOCK DIAGRAM & EXPLANATION

The high-level functional blocks and explanation are as follows:

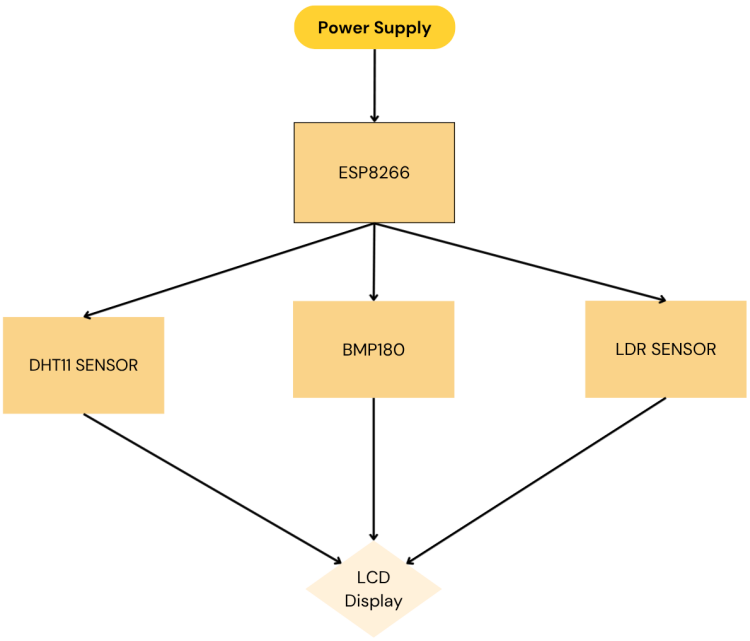
**Power Supply**
Provides stable 5V DC (via USB or adapter) to all components. The ESP8266 internally uses a 3.3V regulator for sensors and logic levels

**ESP8266 (NodeMCU)**
Acts as the main controller and Wi-Fi module. It reads data from sensors (BMP180 and LDR), displays values on the LCD, and sends sensor data to a cloud platform (like ThingSpeak or Blynk) for remote monitoring.

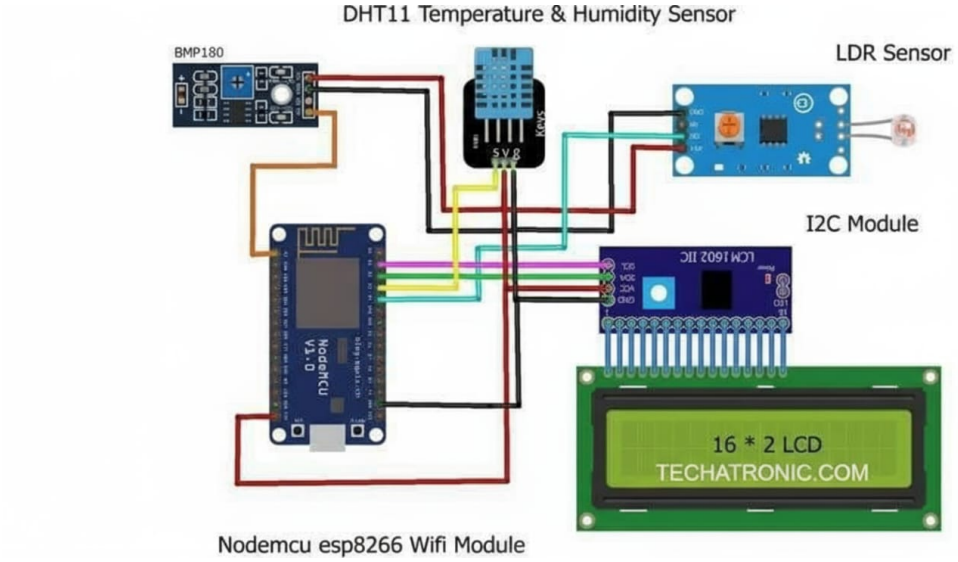
**BMP180 Sensor**
A digital barometric pressure and temperature sensor. Provides atmospheric pressure and ambient temperature readings. Uses I2C communication with the ESP8266.

**LDR (Light Dependent Resistor)**
Detects ambient light intensity. Resistance decreases as light intensity increases. Connected to the analog input pin (A0) of the ESP8266. Helps measure daylight level or brightness.

**LCD Display (16x2 with I2C module)**
Displays real-time values of temperature, pressure, and light intensity. The I2C interface reduces wiring, using only SDA and SCL pins.




# CIRCUIT DIAGRAM & EXPLANATION




## Power

- Laptop USB → NodeMCU USB port (5V). NodeMCU onboard regulator provides 3.3V for sensors and LCD.

## Common Rails

NodeMCU GND → all module GNDs (BMP180, LCD, DHT11, LDR).

## I²C Connections

- NodeMCU D2 (GPIO4) → SDA → BMP180 SDA, I2C LCD SDA

- NodeMCU D1 (GPIO5) → SCL → BMP180 SCL, I2C LCD SCL

### DHT11 Sensor

- VCC → 3.3V
- GND → GND
- DATA → NodeMCU D3 (GPIO0)

### LDR Sensor

- VCC → 3.3V
- GND → GND
- Output → NodeMCU A0

## CODE DESCRIPTION & INTERFACING LOGIC

The program integrates the DHT11, BMP180, and LDR sensors with an ESP8266 NodeMCU microcontroller to monitor weather parameters and send data to the Blynk IoT platform. The required libraries for Wi-Fi, Blynk, I2C communication, LCD, DHT11, and BMP180 are included at the beginning. Pin connections are defined where the DHT11 sensor's data pin is connected to digital pin D3 (with a pull-up resistor between the 3V pin and data line), the LDR sensor is connected to A0, and the I2C pins (SDA and SCL) for the LCD and BMP180 are connected to D2 and D1 respectively. The NodeMCU is powered through the laptop via USB, providing 3.3V to the sensors. The code initializes the Wi-Fi connection using the given SSID and password and authenticates the Blynk connection through the defined token. In the `setup()` function, the serial monitor, sensors, LCD display, and I2C communication are initialized. A timer function triggers `sendSensorData()` every 2 seconds to continuously read temperature and humidity from the DHT11, atmospheric pressure from the BMP180 (converted to hPa), and light intensity from the LDR sensor. The obtained values are sent to the Blynk app using virtual pins, printed on the serial monitor, and displayed on the LCD. The interfacing logic ensures proper communication between all components while the ESP8266 handles Wi-Fi and IoT connectivity.


Figure 1: LCD Display

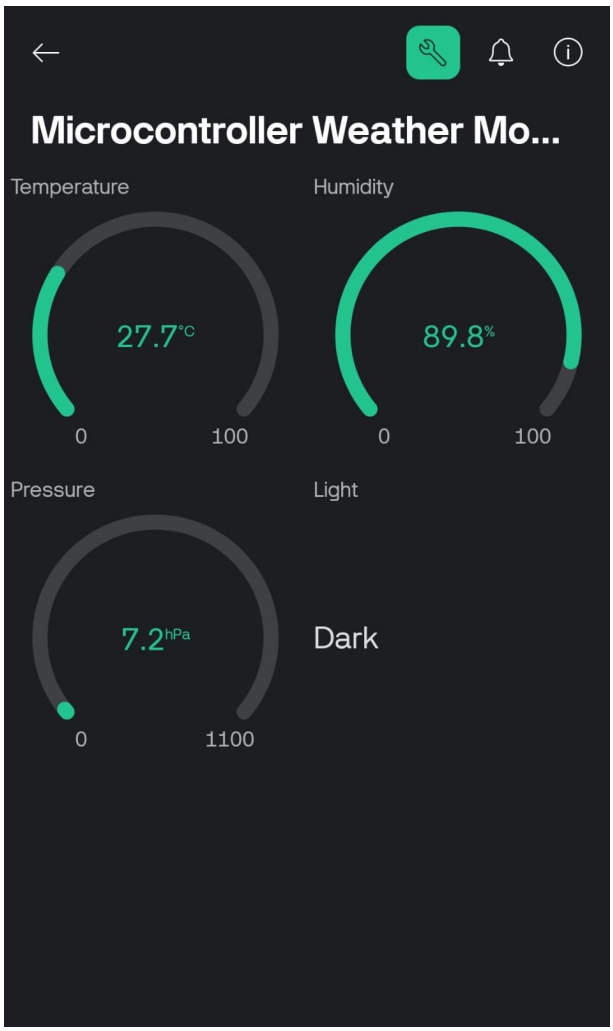



Figure 2: Blynk App

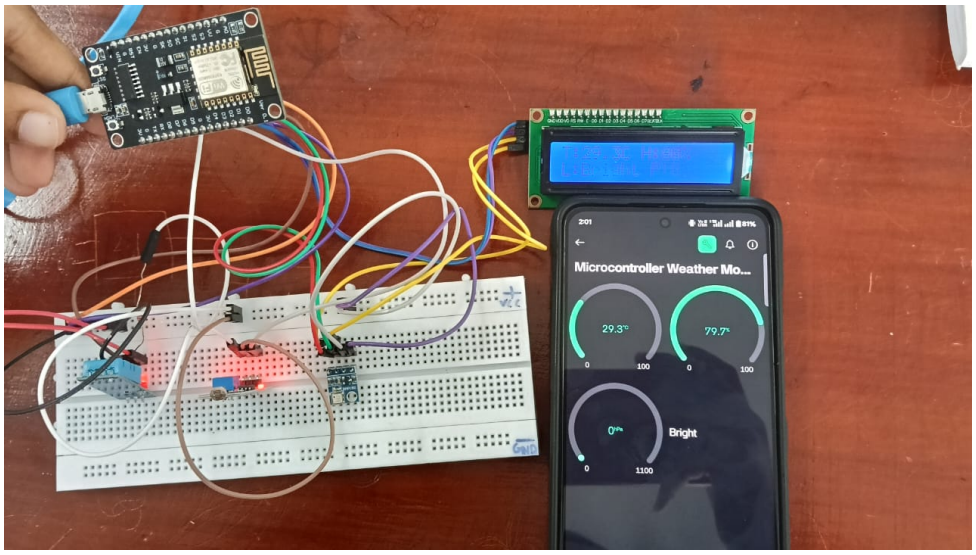



Figure 3: Working Model