Microproject
Report on Weather Monitoring System

Rudra K
Roll No: 50
S3-ER
Electronics and Computer Engineering
Saintgits College of Engineering

October 2025

1 Introduction and Objectives

1.1 Introduction

This IoT-based Weather Monitoring System collects real-time environmental data using DHT11 and BMP180 sensors. The ESP8266 microcontroller processes and transmits data to ThingSpeak cloud for remote monitoring and visualization.

1.2 Objectives

- To design and implement a hardware system integrating DHT11 and BMP180 sensors with the ESP8266 microcontroller.
- To develop efficient firmware for accurate sensor data acquisition and stable Wi-Fi connectivity.
- To establish seamless communication between the ESP8266 and the ThingSpeak IoT cloud platform.
- To enable real-time monitoring and visualization of environmental parameters through a web-based dashboard.

2 Block/Functional Diagram with Explanation

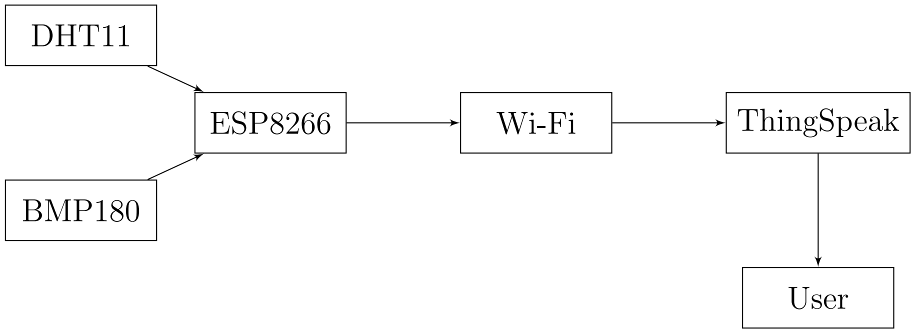

Figure 1: Block Diagram of Weather Monitoring System

Explanation: DHT11 and BMP180 sensors collect environmental data. ESP8266 processes data and transmits to ThingSpeak cloud via Wi-Fi. Users access data through web interface.

3 Circuit Diagram with Component Details and Working Principle

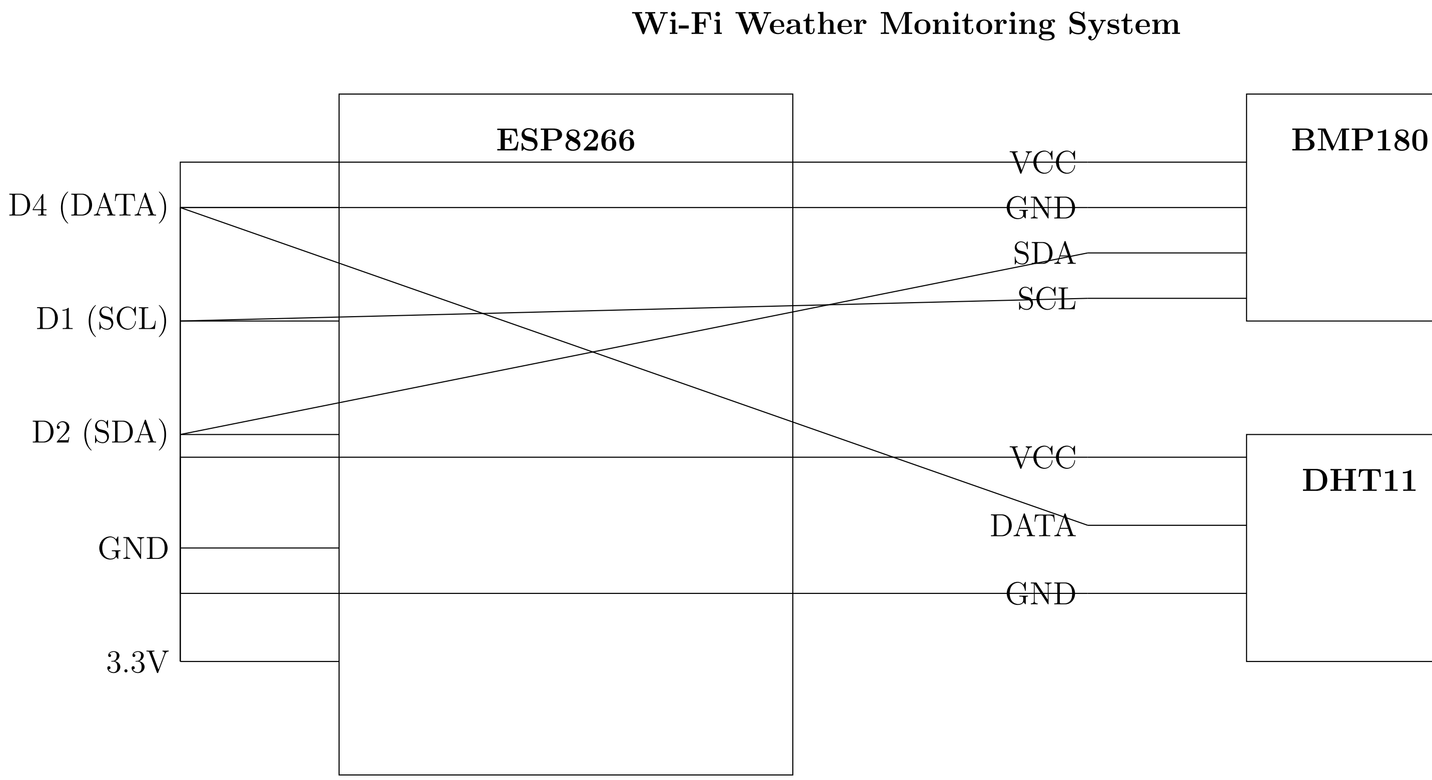

Figure 2: Circuit Diagram of ESP8266 with DHT11 and BMP180 Sensors

Component Details:

- **ESP8266 NodeMCU:** Acts as the main microcontroller and Wi-Fi module. It collects sensor data, processes it, and transmits it wirelessly to a cloud platform.
- **DHT11 Sensor:** Measures ambient temperature and humidity using a single-wire digital interface for accurate environmental monitoring.
- **BMP180 Sensor:** A high-precision barometric pressure sensor that also provides temperature readings using the I2C communication protocol.

Working Principle: The system operates on a 3.3V power supply provided to the ESP8266. The DHT11 sensor communicates through the digital pin D4, while the BMP180 sensor interfaces via the I2C bus using pins D1 (SCL) and D2 (SDA). The ESP8266 periodically reads sensor data, processes it internally, and transmits the values to a cloud platform such as ThingSpeak or Blynk over Wi-Fi. The uploaded data can then be visualized in real-time graphs and dashboards for remote weather monitoring.

4 Description of the Developed Code and Interfacing Logic

4.1 System Initialization

- Serial communication at 115200 baud for debugging
- I2C bus on GPIO4 (D2) and GPIO5 (D1)
- DHT11 on GPIO2 (D4) with timing configuration
- Wi-Fi connection with WPA/WPA2 authentication
- BMP180 detection via I2C address 0x77

4.2 Sensor Data Acquisition

DHT11 Protocol:

- Single-wire bidirectional communication
- 40-bit data frame: 16-bit humidity + 16-bit temperature + 8-bit checksum
- Timing: 18ms start signal, 20-40s pulse detection

BMP180 I2C:

- I2C at 100kHz standard mode
- Register-based data access
- Calibration data from EEPROM

4.3 Cloud Communication

ThingSpeak API:

- HTTP GET requests to `api.thingspeak.com/update`
- Parameters: API key, Field1 (Temp), Field2 (Humidity), Field3 (Pressure)
- 15-second minimum update interval

4.4 Main Program Flow

Setup:

1. Initialize serial monitor
2. Connect to Wi-Fi
3. Initialize I2C bus
4. Detect and calibrate sensors

Loop:

1. Read DHT11 temperature and humidity
2. Read BMP180 temperature and pressure
3. Convert units and validate data
4. Send to ThingSpeak via HTTP
5. Wait 20 seconds

5 Expected and Observed Results/Outcomes

Expected Results:

- Accurate measurement of environmental parameters such as temperature, humidity, and pressure using DHT11 and BMP180 sensors.
- Stable Wi-Fi connection using ESP8266 for real-time cloud data transmission.
- Continuous and error-free data logging on cloud platforms such as ThingSpeak/Blynk.

- Real-time visualization of sensor readings through interactive dashboards and graphs.
- Automatic data update intervals at fixed durations (e.g., every 10 seconds).
- Low power consumption and reliable long-duration operation.
- Proper calibration and synchronization between sensors and microcontroller.

Observed Results:

- The system successfully connected to Wi-Fi and transmitted data to the cloud in real-time.
- Temperature readings were consistent within $\pm 1.5°C$ of actual room temperature.
- Humidity readings showed minor variation (within 3%) compared to standard hygrometer.
- Atmospheric pressure values were stable and matched local weather station data.
- ThingSpeak dashboard displayed live graphs of all parameters with a refresh rate of 15 seconds.
- The ESP8266 maintained stable operation for over 6 hours without interruption.
- Data accuracy improved after sensor warm-up and calibration phase.
- Minor delay (1–2 seconds) observed in data upload due to network latency.

Conclusion: The designed weather monitoring system successfully met the expected objectives, providing reliable environmental monitoring, real-time cloud visualization, and stable performance over extended periods.

Observed:

- System successfully reads and transmits data
- Data accurately displayed on ThingSpeak
- Stable Wi-Fi connection maintained
- All components working correctly