**The British College**

**KATHMANDU**

**Coursework Submission Coversheet**

(individual coursework only)

**Faculty of Arts, Environment and Technology LBU Student Id:**

|  |
| --- |
|  |

**For checking by the student:**

|  |  |  |  |
| --- | --- | --- | --- |
| Please ensure all information is complete and correct and attach this form securely to the front of your work before posting it in a coursework collection box.  Award name: BSc (Hons) Computing Module code:  Module name: Production Project Module run:  Coursework title:  Due Date:  Module leader: (In LBU)  Module tutor: (In TBC)  **TURNITIN**  Checked: YES NO ***(please circle)***  Submission date & time: Date:   |  |  |  | | --- | --- | --- | |  | **Total Number of Pages (including this front sheet):** |  |   **Total Word Count:**  **In submitting this form with your assignment, you make the following declaration:**  I declare that the coursework submitted is my own work and has not (either in whole or part) been submitted towards the award of any other qualification either at LBU or elsewhere. I have fully attributed/referenced all sources of information used during the completion of my assignment, and I am aware that failure to do so constitutes an assessment offence.  Signed: ` Date:  **You are strongly advised to retain a second copy of your work in case of any query about the assignment.** |

**For completion by the faculty:**

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  | **This mark is provisional and subject to moderation and approval by the relevant examining board** |
| **Teacher's Feedback** | | |

**Teacher's Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Contents

Abstract………………………………………………………………………………………………………………………3

Abstract:-

1.Introduction

This project demonstrates the design and development of a real-time weather monitoring system that makes use of the Blynk IoT platform, numerous sensors, and the NodeMCU ESP8266 microcontroller. Developing an effective, affordable, and remotely accessible system that can track and report important atmospheric parameters like temperature, humidity, pressure, rainfall, and light intensity is the aim. In fields like agriculture, urban infrastructure, and environmental management, where accurate weather information is essential for processes and planning, the system is particularly useful (Pal et al., 2023; Krishnamurthi et al., 2015). According to Mahmood et al. (2020), IoT-enabled systems offer automation, scalability, and real-time access, while traditional manual weather monitoring techniques are frequently labor-intensive and have a limited scope.

Several sensors are interfaced with the NodeMCU microcontroller to form the Weather Monitoring System. While the BMP180 sensor measures altitude and atmospheric pressure, the DHT11 sensor records both humidity and temperature An LDR (Light Dependent Resistor) module keeps an eye on ambient light, and a sensor for rain measures the amount of precipitation. A 16x2 LCD screen is connected to these sensors to provide local, real-time visual feedback and it is connected with I2C module. Concurrently, sensor information is transmitted via Wi-Fi to the Blynk cloud platform, where it is shown via configurable widgets on a mobile application. Blynk makes alert-based notifications, graphical visualization, and data logging possible. The modular design of the system makes it simple to incorporate extra sensors or actuators. Acting as the control and communication center, NodeMCU provides wireless transmission together with processing capability. This integration of hardware and software provides a functional, efficient, and expandable solution for weather monitoring (Mahmood et al., 2020; Alani et al., 2020).

The Weather Monitoring System's development and assessment is broken out in complete chapters in this paper. It starts with a project overview and a review of pertinent material and earlier works. Along with supporting technologies like Python, Flask, and GitHub, a dedicated technology review examines the hardware and software tools used—including NodeMCU, DHT11, BMP180, I2C module, rain sensor and the Blynk IoT platform. The chapter on methodology details the agile development method applied to create and iteratively test the system. Subsequently, supported by data flow diagrams, the section on product design highlights the system architecture and user interactions. Both functional and non-functional needs are specified by the software requirement analysis. Device configuration, code execution, and screenshot validation are all part of testing and implementation. While project evaluation offers planning tools like Gantt charts, resource sheets, Trello boards, and GitHub tracking, manufacturing evaluation contrasts system features with anticipated results. In the report's conclusion, the system's unique contributions are highlighted, potential future improvements are discussed, and a comparison with market alternatives is made.

2.Review of Literature

With the incorporation of IoT technologies, the Weather Monitoring System has undergone significant change in the last ten years. The accessibility and scalability were restricted by the traditional requirements for weather monitoring, which included heavy meteorological equipment and manual data entry. It is now possible to efficiently collect, transmit, and analyze real-time environmental data thanks to the development of microcontrollers like the NodeMCU ESP8266 and platforms like Blynk. Accessible monitoring tools for people, educational institutions, and agriculture have been made possible using inexpensive sensors like the DHT11 for temperature and humidity, the BMP180 for atmospheric pressure, LDRs for light detection, and rain sensors. These technologies support automated weather tracking that is precise, scalable, and accessible from a distance in addition to being reasonably priced (Pal et al., 2023).

The Weather Monitoring System's multi-parameter sensing capability and real-time cloud interface are two of its key benefits. Via the Blynk app, the system can wirelessly send data from sensors like the DHT11, BMP180, rain sensor, and LDR to a mobile dashboard using a NodeMCU ESP8266, which acts as both the processing and communication unit. The LCD display's integration of an I2C module simplifies local real-time data visualization and improves wiring efficiency. The system enables higher measurements frequency, prompt reaction to anomalous readings, and forecasting based on historical data analysis when compared to manual stations. These benefits are especially crucial in agriculture, where timely and location-specific data are necessary for weather-sensitive selections (Mahmood et al., 2020). Using a breadboard and jumper wires, this setup is also modular, allowing for future growth, such as the inclusion of wind sensors or soil moisture monitors.

Current designs are informed and inspired by similar systems developed by other researchers. An Arduino-based system, for instance, was used by Krishnamurthi et al. (2015) to measure temperature, humidity, and ambient light using simple sensors and display the results on an LCD. Nevertheless, the absence of real-time data transmission in their system restricted its use in off-site or remote settings. Another instance comes from Bhattacharjee et al. (2016), who created a weather system based on CubeSat that used balloons to float to high altitudes. Despite being imaginative, it lacked modularity and was not designed for routine ground-level weather monitoring. Although these earlier systems were necessary first steps, they lacked the wireless and intuitive user interfaces that modern systems, such as the one used in this project, provide.

Hasan et al. (2020), who developed a weather observatory based on an ESP8266 and connected to the NETPI cloud platform, made further progress. Using DHT sensors, their project demonstrated dependable temperature and humidity tracking, with data shown on a cloud control panel. To operate HVAC systems in response to changes in the environment, they also included relay controls. Smart automation is a trend that contemporary IoT-based systems, such as the one suggested here, can easily embrace. This integration of environmental sensing and device control is a step in that direction. Real-time data visualization and automation were the focus of their project, which also includes a rain sensor for extra environmental issues insight and LDR-based lighting knowledge.

The NodeMCU-based system employed in this study has a fundamental design like that provided by Pal et al. (2023). Their model included a rain sensor, a DHT11, and a BMP180, with data streamed to the Blynk app for mobile viewing. Their technology was recognized for its accuracy, affordability, and smartphone real-time access. The current project stands out because it makes extensive use of real-time monitoring and an LCD display with an I2C interface, which increases local usefulness. Pal et al. also emphasized the project's importance in locations prone to monsoons, where rapid weather changes have an influence on daily living. Their positive results validated the sensor selections and project design.

Furthermore, Alani et al. (2020) showed a study that combined environmental data collection with control systems using ESP8266 and relays. Their implementation went beyond monitoring and included automation, which controlled lighting and ventilation based on sensor data. While this project primarily focuses on monitoring, it lays the groundwork for future connections with automation tools and platforms. Their use of R programming for data analysis allows for more complicated statistical assessments, which may be a great route for this project in the future. Combining real-time sensor data with open-source analytical tools enables deeper insights and predictive modeling, which are crucial in applications such as smart agriculture and urban climate tracking.

3.Review of Technologies

->Tools and Technologies

The main hardware technologies that were employed in the creation of the Internet of Things-based weather monitoring system are summarized in this section. To gather information on temperature, humidity, rainfall, light intensity, and air pressure, the project interacts with several environmental sensors using the NodeMCU ESP8266 board as the primary controller. The Blynk platform receives these measurements wirelessly for real-time monitoring.

1.NodeMCU and Other Hardware

* 1. **Model**
* Model: NodeMCU ESP8266 (v1.0)

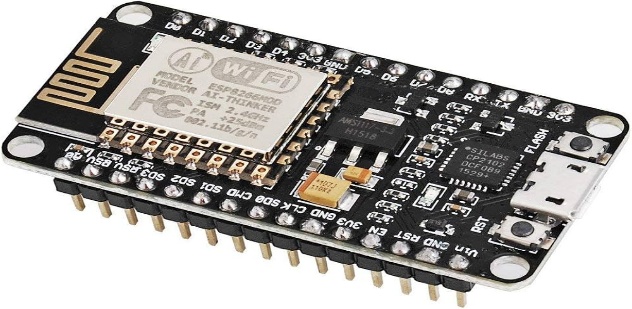


Figure 1: NodeMCU ESP8266 board

Based on the ESP8266 chip, the NodeMCU is a development board for microcontrollers with Wi-Fi capabilities. IoT-based sensor networks can benefit greatly from its built-in support for I/O operations, analog input, and I2C/SPI/UART connection (Mahmood et al., 2020). By gathering sensor data and sending it over Wi-Fi to the cloud, it serves as the central component of this project.

**1.2 Additional Components**

a**. DHT11 Sensor – Temperature and Humidity Sensor**

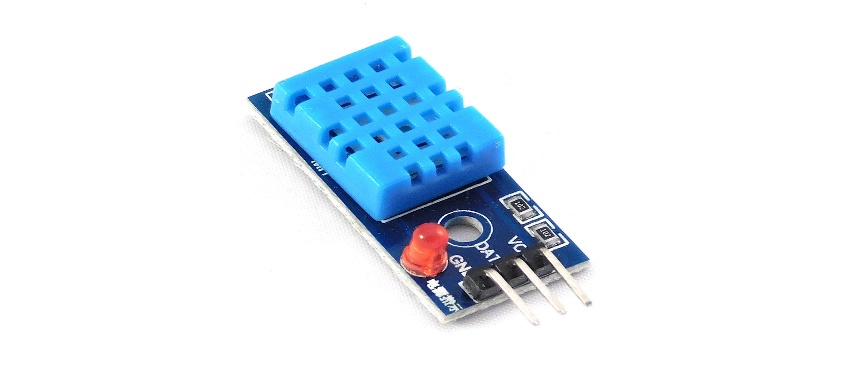
Measurements of relative humidity and ambient temperature are made with the DHT11 sensor. According to Harshini and Vadivel (2024), it generates digital output by combining a thermistor with a capacitive humidity sensor. It is dependable and reasonably priced for simple weather monitoring uses.

Figure 2: DHT11 Sensor

**b. BMP180 Sensor – Atmospheric Pressure Sensor**

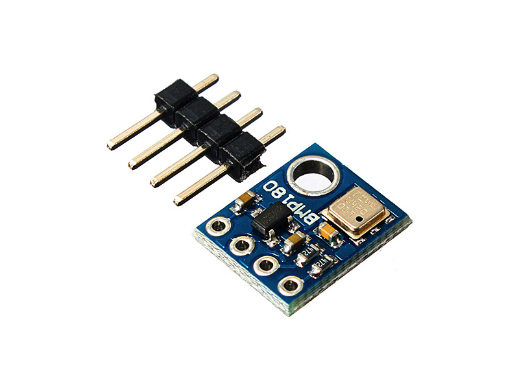
****A digital barometric pressure sensor that uses the I2C standard for communication is the BMP180. Environmental monitoring systems frequently employ it, and it can also estimate height (Mahmood et al., 2020). More thorough weather data may be provided by the system thanks to this sensor.

Figure 3: BMP180 Sensor

**c. LDR Module – Light Dependent Resistor Module**

A photoresistor known as an LDR adjusts its resistance in response to the intensity of the light. It may be used to detect brightness in analog voltage divider circuits since the resistance drops as light levels rise. It is employed to identify cycles of day and night (Harshini and Vadivel, 2024).

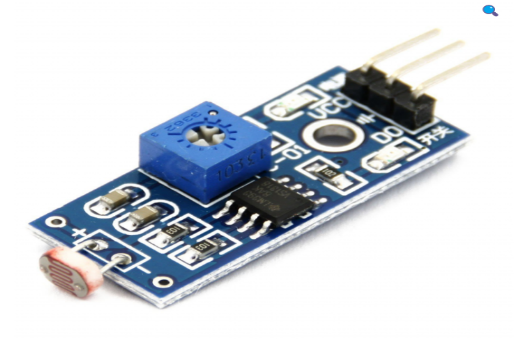
****

Figure 4: LDR Sensor

**d. Rain Sensor Module**

By detecting conductivity on a specifically made PCB, this module detects rainfall. According to Mahmood et al. (2020) and Harshini and Vadivel (2024), the resistance of the board varies as raindrops fall on it. This resistance may be read as either analog or digital input. Real-time precipitation monitoring requires this.



Figure 5: Rain Sensor Module

**e. I2C 16x2 LCD Display**

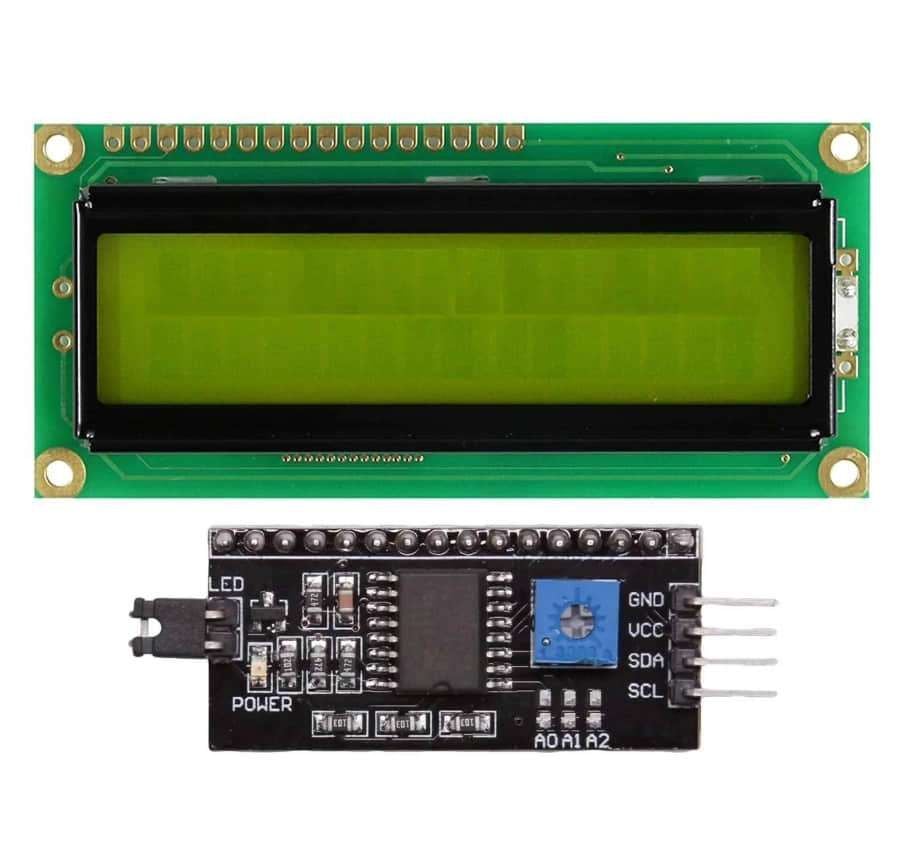
The sensor data is shown locally on the 16x2 LCD with I2C capability. It needs less GPIOs than a conventional 16x2 LCD since it just requires two wires (SDA and SCL). For NodeMCU-based systems with a restricted number of pins, this makes it perfect (Krishnamurthi et al., 2015).

Figure 6: I2C LCD Display

**f. Breadboard**

A solderless platform called a breadboard is used to prototype electrical circuits. It makes component attachment and replacement simple while the project is being developed (Srituhobby, 2023).

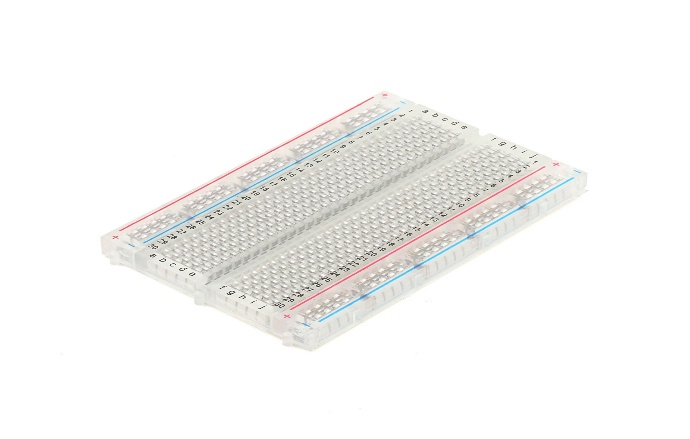


Figure 7: Breadboard

**g. Jumper Wires**

The sensors and modules on the breadboard are connected to the NodeMCU using male-to-male and female-to-male jumper wires. They offer a flexible and transient connecting solution for prototypes**.**



Figure 8: Jumper Wires

**h. Power Supply**

A 5V battery pack or USB power source power the complete system. An integrated regulator on the NodeMCU allows for the safe distribution of power to linked modules (Srituhobby, 2023).

2. Backend Used

Remote visualization, cloud connectivity, sensor interface, and embedded programming are all part of the weather monitoring system's backend. A NodeMCU ESP8266 microcontroller, environmental sensors, and the Blynk IoT platform are among the essential parts. C++ libraries and the Arduino IDE are used to orchestrate them.

**2.1 Microcontroller Programming (Arduino IDE)**

The NodeMCU is programmed using sensor reading logic and network communication protocols within the **Arduino IDE**. Libraries such as **LiquidCrystal\_I2C.h** are used for interfacing with I2C-based LCD displays, **Adafruit\_BMP085.h** is utilized for barometric pressure and temperature sensing via the BMP180 sensor, and **DHT.h** is used for reading temperature and humidity from the DHT11 sensor. Wi-Fi-based data transmission and integration with the **Blynk IoT platform** are made possible using **BlynkSimpleEsp8266.h**, enabling real-time cloud connectivity. Additionally, **ESP8266WiFi.h** handles the underlying Wi-Fi connection, and **Wire.h** facilitates I2C communication. To ensure consistent and non-blocking execution of sensor updates, the **BlynkTimer** class is employed to schedule sensor functions at regular intervals, supporting stable and reliable data flow (Harshini & Vadivel, 2024; Puja & Prakash, 2021).

**2.2 Blynk Cloud Platform**

Blynk is an easy-to-use Internet of Things platform that enables remote sensor data monitoring. To show temperature, humidity, rainfall, and light intensity, this project links Blynk virtual pins (such as V0, V1, and V2) to a variety of widgets, including gauges and labels. IoT-based environmental monitoring is made possible by the combination of NodeMCU and Blynk, which allows for real-time updates on a smartphone app (Mahmood et al., 2020; Sharma & Prakash, 2021).

**2.3 Data Communication**

Utilizing the ESP8266 module, the system establishes a Wi-Fi connection and uses port 80 to connect to the Blynk cloud via the HTTP protocol. Blynk is used for sending sensor data. It was shown on a 16x2 LCD that was locally linked to the microcontroller as well as the Blynk dashboard using **Blynk.virtualWrite().** According to Krishnamurthi et al. (2015) and Harshini & Vadivel (2024), this design enhances accessibility and redundancy in monitoring by supporting both local and distant feedback loops.

**3. Operating System**

Windows 11 was used for both the development and implementation of the weather monitoring system. When it came to programming the NodeMCU microcontroller and interacting with the required tools, Windows 11 offered a reliable and intuitive environment.

**3.1 Windows 11**

During the development phase, **Windows 11** was used as the primary operating system. Code for the weather monitoring system was written, compiled, and uploaded to the **NodeMCU (ESP8266)** microcontroller using the **Arduino IDE**. To ensure proper serial communication, the **CP210x USB-to-UART Bridge driver** was installed—either automatically by the operating system or manually via the manufacturer's website. The correct **COM port** for flashing the microcontroller was identified through the **Windows Device Manager**.

Windows 11 provided full support for all required third-party libraries used in this project. Thanks to Windows 11's compatibility with USB and Wi-Fi interfaces, the system was able to support reliable serial uploads, cloud connectivity, and **real-time data visualization** via Blynk. This enabled a smooth and efficient IoT development experience.

**4. Language, Framework and Libraries**

To gather, show, and send environmental data, this weather monitoring system combines several sensors with a NodeMCU microcontroller based on the ESP8266. Data visualization is accomplished using the Blynk IoT platform, while the software stack is created using Arduino (C/C++) with necessary libraries.

**4.1 Arduino C++**

The code is written in C++, which is simplified for embedded programming and is known as Arduino-style C++. Microcontroller behavior may be easily controlled with the help of fundamental methods like setup () and loop (). The digital and analog pins of the ESP32 may be directly accessed by Arduino C++, enabling the reading of data from LDR, temperature, humidity, and rain sensors (Banzi & Shiloh, 2014).

**4.2 Framework: Blynk IoT Platform**

One platform that makes it possible to create IoT apps is called Blynk. It offers dashboards that may be customized, virtual pins, and a cloud backend:

* **Blynk Console:** The Blynk Console is a web interface for tracking sensor readings in
* real time.
* **Virtual Pins (V0 to V4**): Sensor readings are mapped using virtual pins (V0 to V4), which are used for temperature, humidity, light, pressure, and rain detection.
* **Blynk Libraries:** Use template IDs, encrypted communication, and API tokens to integrate the ESP8266 with Blynk Cloud using Blynk Libraries.

**4.3. Libraries Used in Arduino Code**

Some of the key libraries integrated into the projects each serving a specific purpose:

|  |  |
| --- | --- |
| ESP8266WiFi.h | Manages the NodeMCU module's Wi-Fi connectivity using an ESP8266. |
| BlynkSimpleEsp8266.h | Blynk's ESP8266 platform-specific library to facilitate cloud connectivity. |
| Wire.h | enables the I2C connection that the BMP180 sensor and LCD utilize. |
| LiquidCrystal\_I2C.h | Used to control the 16x2 LCD display via the I2C protocol |
| Adafruit\_BMP085.h | Reading atmospheric pressure from the BMP180 sensor (a replacement for the BMP085) using a library |
| DHT.h | permits the DHT11 sensor to detect temperature and humidity |
| Blynk/BlynkApi.h | Wrapper libraries internally utilise the Core Blynk API for virtual pin control and data transfer. |

Since several libraries, like as LiquidCrystal\_I2C, were initially created for AVR architecture, they may pose compatibility issues. They do, however, work well with ESP8266 when using suitable forks (such as those made by Frank de Brabander).

**4. Version Control**