

National University of Sciences & Technology

School of Electrical Engineering and Computer Science Department of Computing

**IT-863: Internet of Things**

**Spring 2023**

**SEMESTER PROJECT**

IOT BASED WEATHER STATION

**Instructor: Dr. Rafia Mumtaz**

|  |  |
| --- | --- |
| Maasooma  429876  MS CS 2022 | Ayesha Sana  402433  MS CS 2022 |
|  |  |
| Tabinda Ghazal  402420  MS CS 2022 | Saad Mahmud Mirza  400432  MS CS 2022 |

**End Semester Project:**

**IoT-based Weather Station**

→ **Description** – Humidity, Temperature and Pressure are three basic parameters to build any Weather Station and to measure environmental conditions. The objective of this project is to implement an IoT based Weather station that monitors the mentioned environmental conditions and offer real-time notifications when there is considerable change in parameters. Refer to fig to grasp the basic idea.

→ **Hardware Options** – temperature & humidity sensors, pressure sensors, rain sensor etc.

→ **Online IoT servers for visualization** – Aurdino IOT,ThingSpeak, Ubidots, Google IoT, BLYNK, etc

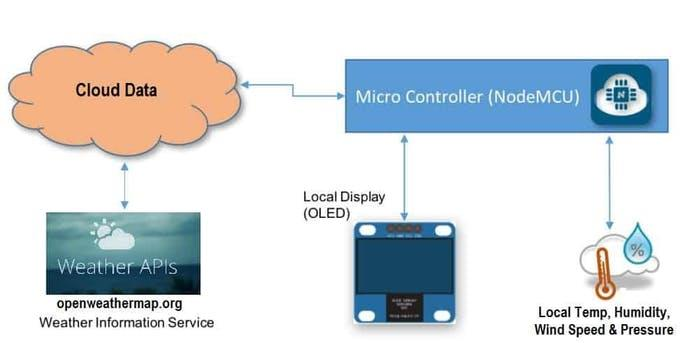


Figure : A basic representation of an IoT-based Weather Station

# **Details of Sensors:**

# NodeMCU ESP8266



NodeMCU is an open-source Lua based firmware and ****development board**** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

### ****NodeMCU Development Board Pinout Configuration****

|  |  |  |
| --- | --- | --- |
| ****Pin Category**** | ****Name**** | ****Description**** |
| Power | Micro-USB, 3.3V, GND, Vin | ****Micro-USB:**** NodeMCU can be powered through the USB port  ****3.3V:**** Regulated 3.3V can be supplied to this pin to power the board  ****GND:**** Ground pins  ****Vin:****External Power Supply |
| Control Pins | ****EN, RST**** | The pin and the button resets the microcontroller |
| GPIO Pins | GPIO1 to GPIO16 | NodeMCU has 16 general purpose input-output pins on its board |
| SPI Pins | SD1, CMD, SD0, CLK | NodeMCU has four pins available for SPI communication. |
| UART Pins | TXD0, RXD0, TXD2, RXD2 | NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program. |
| I2C Pins |  | NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C. |
| Analog Pin | A0 | Used to measure analog voltage in the range of 0-3.3V |

### ****NodeMCU ESP8266 Specifications & Features****

* Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
* Operating Voltage: 3.3V
* Input Voltage: 7-12V
* Digital I/O Pins (DIO): 16
* Analog Input Pins (ADC): 1
* UARTs: 1
* SPIs: 1
* I2Cs: 1
* Flash Memory: 4 MB
* SRAM: 64 KB
* Clock Speed: 80 MHz
* USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
* PCB Antenna
* Small Sized module to fit smartly inside your IoT projects

### ****Brief About NodeMCU ESP8266****

The ****NodeMCU ESP8266 development board**** comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.

NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.



### ****Programming NodeMCU ESP8266 with Arduino IDE****

The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use.

Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All we need is the Arduino IDE, a USB cable and the NodeMCU board itself.

### ****Uploading your first program****

Once Arduino IDE is installed on the computer, connect the board with the computer using the USB cable. Now open the Arduino IDE and choose the correct board by selecting ****Tools>Boards>NodeMCU1.0**** (ESP-12E Module), and choose the correct Port by selecting ****Tools>Port****. To get it started with the NodeMCU board and blink the built-in LED, load the code.Once the code is loaded into your IDE, click on the ‘upload’ button given on the top bar. Once the upload is finished, you should see the built-in LED of the board blinking.

### ****Applications****

* Prototyping of IoT devices
* Low power battery operated applications
* Network projects
* Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities

**3.Communication Technology: Provide a detailed justification of selecting a specific communication**

**technology(s) at physical, internet and application layers and discuss the conceptualization of**

**possible network topologies if the system is to be deployed in the real environment**

To deploy the 5.IoT-based Weather Station system in a real environment, we need to consider suitable communication technologies at the physical, internet, and application layers. Let's discuss each layer and provide a justification for selecting specific technologies.

**Physical Layer:**

At the physical layer, we need to choose a communication technology that can establish a reliable connection between the sensors and the central monitoring system. In this case, the NodeMCU ESP8266 board can be utilized, which combines the ESP8266 Wi-Fi module and a microcontroller. The ESP8266 module supports Wi-Fi connectivity and is widely used for IoT applications. It provides a low-cost and efficient solution for wireless communication between the weather station sensors and the central system.

**Justification**: The ESP8266 board offers Wi-Fi connectivity, enabling the weather station to connect to the existing internet infrastructure. It provides a reliable and widely supported wireless communication technology for IoT devices, making it suitable for transmitting sensor data to the central monitoring system.

**Internet Layer:** The internet layer facilitates communication between the weather station system and the central monitoring system over the internet. For this layer, we can utilize the TCP/IP protocol suite, which is the foundation of internet communication. It ensures reliable and secure transmission of data packets over the internet.

**Justification**: The TCP/IP protocol suite is widely adopted and supported, making it compatible with various network infrastructures. It provides robust data transmission, error detection, and correction mechanisms, ensuring the integrity of sensor data during transmission. Additionally, it supports IP addressing, allowing seamless integration with existing internet-based systems.

**Application Layer:** At the application layer, we need to select a communication protocol that allows efficient and standardized data exchange between the weather station system and the central monitoring system. In this case, we can choose the MQTT (Message Queuing Telemetry Transport) protocol. MQTT is a lightweight publish-subscribe messaging protocol designed for IoT applications. It provides efficient communication between resource-constrained devices and a central server.

**Justification**: MQTT is suitable for IoT-based weather station systems as it minimizes the bandwidth and power requirements. It follows a publish-subscribe model where sensors publish data to specific topics, and the central monitoring system subscribes to those topics to receive updates. This asynchronous communication model ensures efficient data transmission and minimizes network overhead.

**Network Topologies:** Based on the conceptualization of possible network topologies, the following options can be considered for deploying the .IoT-based Weather Station :

**Star Topology:** In a star topology, each weather station sensor, including the NodeMCU ESP8266 board, is connected directly to the central monitoring system. This can be achieved by configuring the sensors to connect to the central system's Wi-Fi network. The central monitoring system acts as a hub, receiving data from each sensor and processing it accordingly.

**Mesh Topology:** A mesh topology involves establishing direct communication links between the weather station sensors themselves. Each sensor can act as a relay point, forwarding data to other sensors until it reaches the central monitoring system. This topology can be useful in scenarios where direct Wi-Fi connectivity to the central system is challenging or when redundancy and fault tolerance are essential.

**Hybrid Topology:** A hybrid topology combines elements of star and mesh topologies. Some weather station sensors can directly connect to the central monitoring system in a star-like configuration, while others can form a mesh network for inter-sensor communication. This approach provides a balance between simplicity and flexibility.

The choice of the network topology depends on factors such as the physical environment, the number of sensors, the distance between sensors and the central system, and the desired fault tolerance. Each topology has its advantages and considerations, so a careful evaluation is necessary to determine the most suitable configuration for the specific deployment scenario.

**4. Data Visualization/Analysis and system interface:** Perform data analysis using descriptive, diagnostic,

and predictive analytics. To increase the volume of data for analytic purposes, you can download publicly available online datasets. You should make a suitable interface for your system. For data visualization, please create a dashboard for displaying infographics.

**MAX VALUES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SR.#** | **VALUE RECORDED DATES** | **TEMPERATURE**  **℃** | **HUMIDITY**  **%** | **RAIN**  **%** | **PRESSURE**  **mb** |
| 1 | 22-05-2023 | 41 | 46 | 28 | 1011 |
| 2 | 21-05-2023 | 40 | 35 | 12 | 1003 |
| 3 | 20-05-2023 | 39 | 23 | 13 | 1005 |
| 4 | 19-05-2023 | 41 | 40 | 6 | 1005 |
| 5 | 18-05-2023 | 36 | 25 | 18 | 1011 |
| 6 | 17-05-2023 | 36 | 40 | 38 | 1009 |

**MIN VALUES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SR.#** | **VALUE RECORDED DATES** | **TEMPERATURE**  **℃** | **HUMIDITY**  **%** | **RAIN**  **%** | **PRESSURE**  **mb** |
| 1 | 22-05-2023 | 26 | 40 | 9 | 1006 |
| 2 | 21-05-2023 | 25 | 30 | 6 | 1000 |
| 3 | 20-05-2023 | 23 | 18 | 17 | 1003 |
| 4 | 19-05-2023 | 22 | 24 | 2 | 1002 |
| 5 | 18-05-2023 | 23 | 17 | 10 | 1006 |
| 6 | 17-05-2023 | 23 | 20 | 23 | 1004 |

|  |  |
| --- | --- |
| **Hardware** | **Details** |
| NodeMCU ESP8266 Board |  |
| BMP180 Sensor |  |
| DHT11 Sensor |  |
| Rain Sensor FC-37 |  |
| Resister 4.7k |  |
| Connecting Wires |  |
| Breadboard |  |

Power Consumption

Communication Technology

Tools and Technologies