Name: _____ | Class: D16 | Roll No: __

EXPERIMENT NO: 01

Aim: To interface temperature and humidity sensor with Node MCU (ESP8266) and obtain output on serial monitor with timestamp.

Apparatus: NODE MCU, Temperature and Humidity Sensor (DHT11), Micro USB Cable, Breadboard, Female to Female connectors.

Theory:

(A) HARDWARE REQUIRED

(1) NODE MCU

NodeMCU is an open source Lua based firmware for the ESP8266 WiFi SOC from Espressif and uses an on-module flash-based SPIFFS file system. NodeMCU is implemented in C and is layered on the Espressif NON-OS SDK. The prototyping hardware is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.

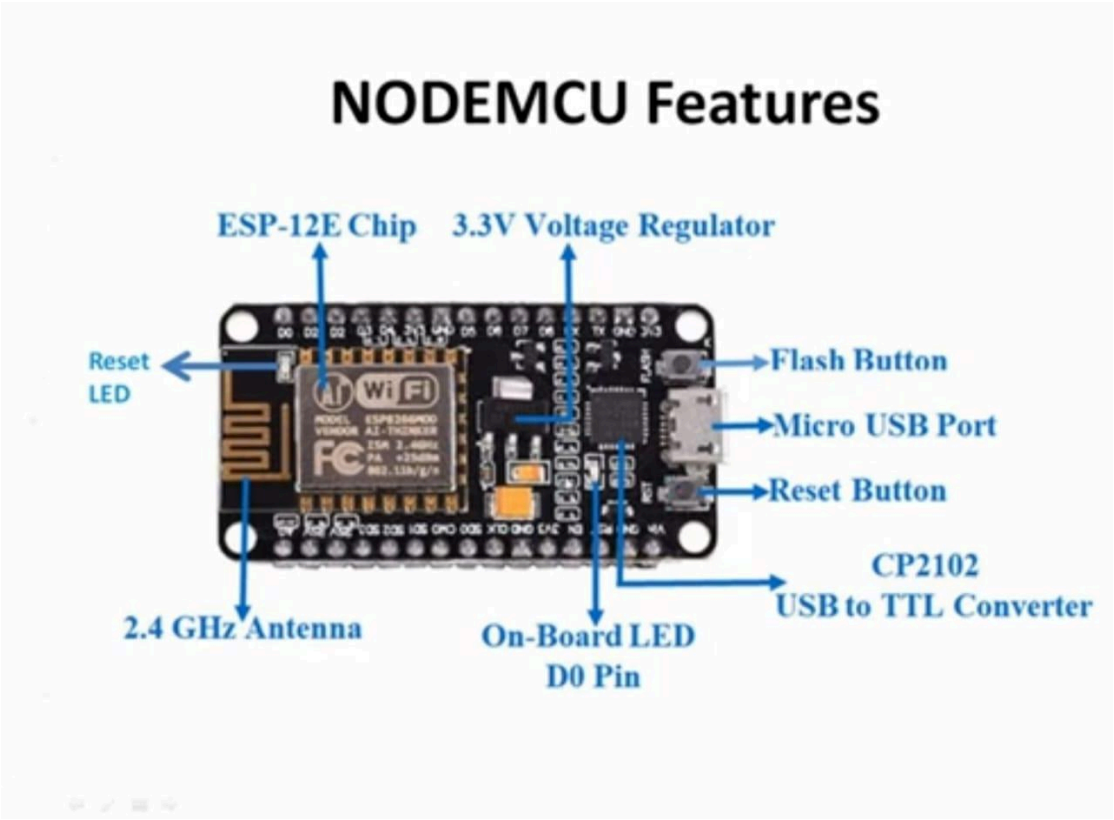

Fig. (1) NODE MCU GPIO pin Connection

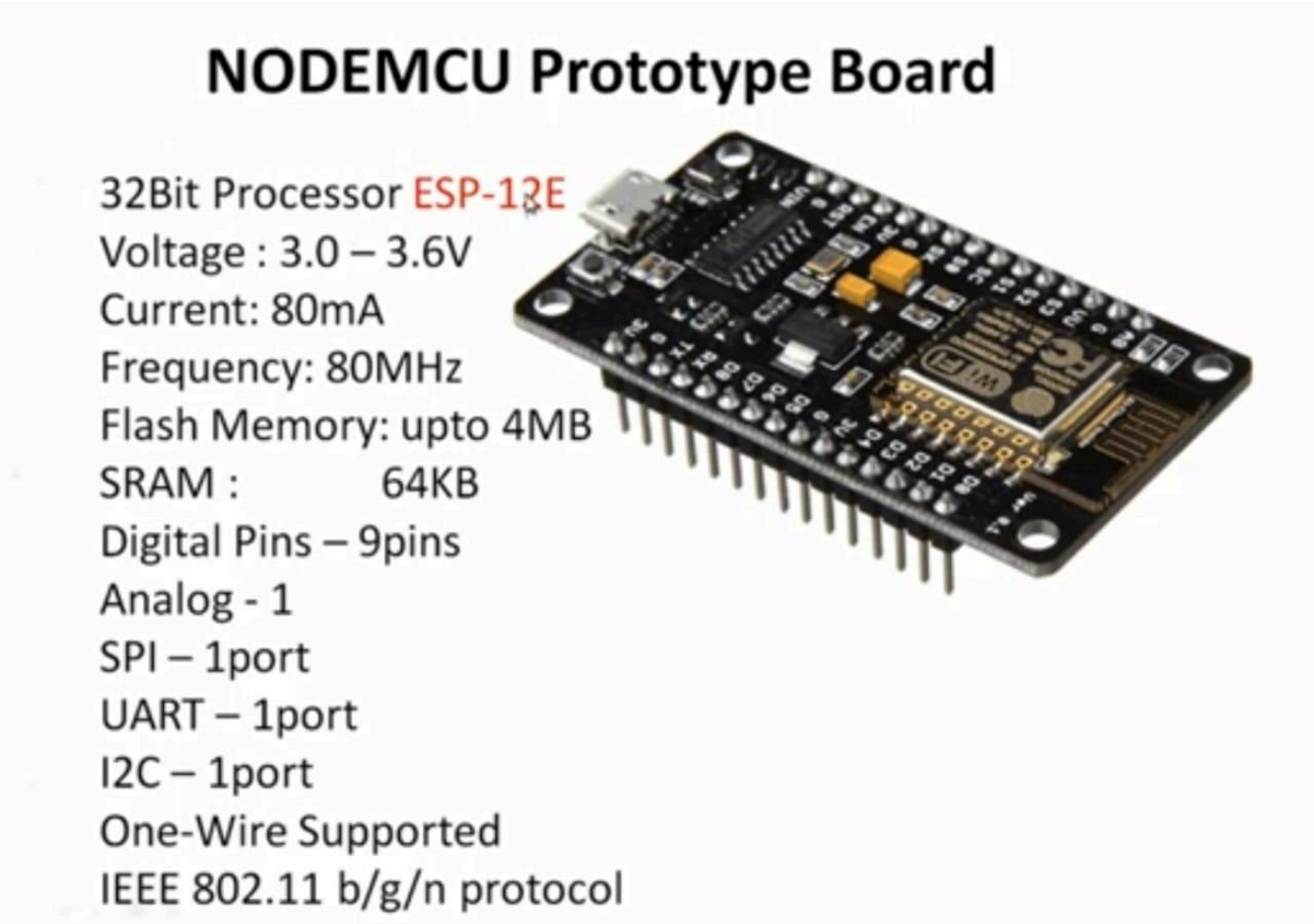

Fig. (2) NODE MCU GPIO Features

(2) DHT11 SENSOR

The DHT11 detects water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface. When water vapor is absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes

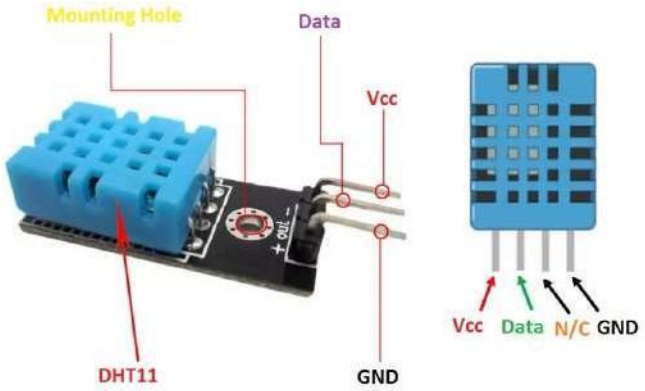

Fig. (3) DHT11 Sensor

For DHT11 Sensor module		
1	Vcc	Power Supply 3.5V to 5.5V
2	Data	Outputs both Temperature & Humidity through serial data
3	Ground	Connected to the ground of the circuit

DHT11 Specification:

- **Operating Voltage:** 3.5V to 5.5V
- **Operating current:** 0.3mA (measuring) 60uA (standby)
- **Output:** Serial Data
- **Temperature Range:** 0℃ to 50℃
- **Humidity Range:** 20% to 90%
- **Resolution:** Temperature and Humidity both are 16-bit
- **Accuracy:** ±1℃ and ±1%

(B) SOFTWARE

(1) ARDUINO IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++, It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

(2) LIBRARY FILE

Following two libraries will be required to run this code. This library file should be placed at the install folder of Arduino

- Adafruit Sensor Library
- DHT Sensor Library

(C) CONNECTIONS AND CIRCUIT DIAGRAM

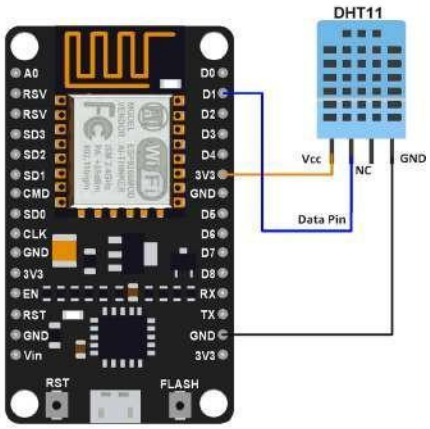

Fig. (4) General Circuit Diagram of Node MCU interface with DHT11 Sensor

Wiring the **DHT11** to the Node MCU is really easy, but the connections are different depending on which type you have either 3-pins or 4-pins

The **wiring connections** are made as follows:

- **Pin 1** of the DHT11 goes into **+3v** of the Node MCU
- **Pin 2** of the DHT11 goes into Digital Pin **D3** of the Node MCU.
- **Pin 3** of the DHT11 goes into Ground Pin (**GND**) of the Node MCU

PROCEDURE:

1. Make the connections as specified in the circuit diagram
2. Provide the power supply to NODE MCU by connecting it with the help of micro-USB cable to the CPU of the computer.
3. Open the **Arduino IDE** and then choose **New** where a new sketch opens up.
4. Write the code on to a new sketch and then click on save.
5. Before using the DHT11 with Node MCU, install the **DHTLib** library. It has all the functions needed to get the humidity and temperature readings from the sensor.
6. Then go to **Sketch > Include Library > Manage Libraries > Search DHTLib**
7. Upload the code to the NODE MCU by selecting the right board and port.
8. If there are no errors, the code will be compiled and uploaded to the Node MCU successfully else check for the errors mentioned.

CODE:

```
#include <ESP8266WiFi.h>
#include "DHT.h"

#define DHTPIN D7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  dht.begin();
  Serial.println("DHT11 Sensor Reading Started...");
}

void loop() {
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
```

```
    delay(2000);
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C  |  Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  delay(2000);
}
```

OUTPUT:

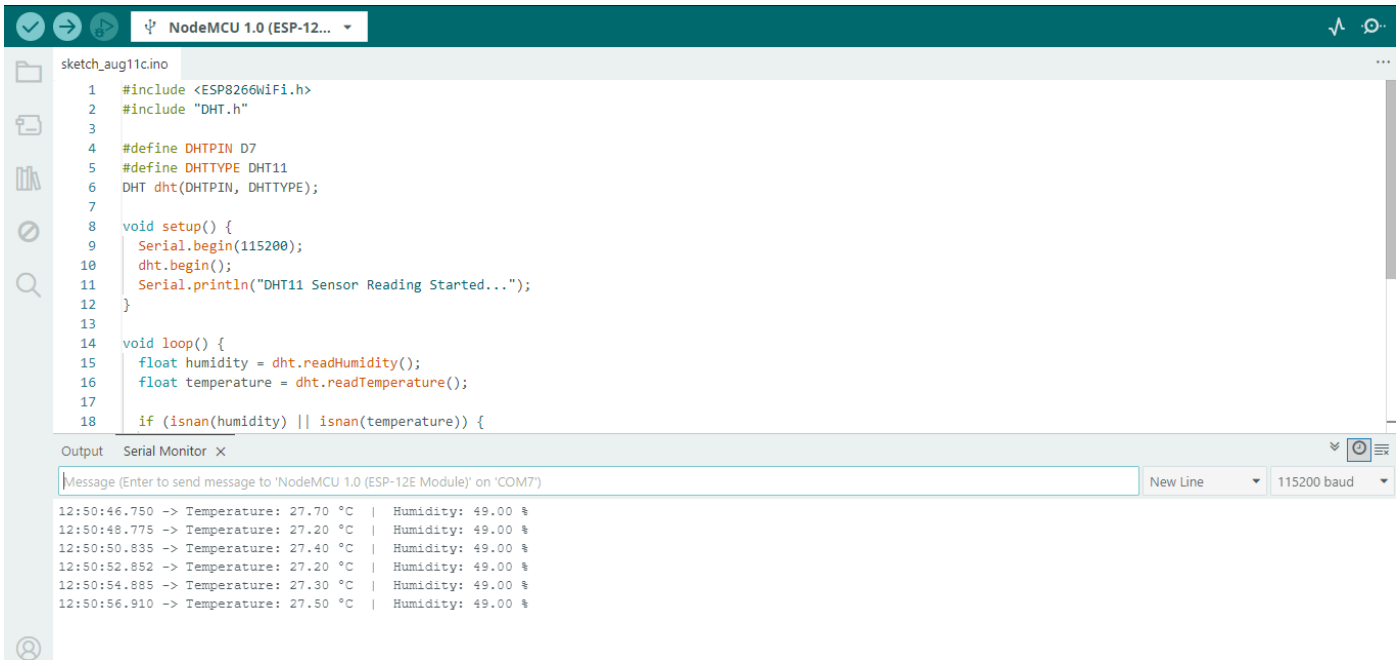

Fig. (5) Serial monitor showing Temperature and humidity after 2 Seconds of interval

CONCLUSION:

We designed and implemented a hardware circuit to monitor temperature and humidity using a DHT11 sensor and a NodeMCU (ESP8266). The DHT11 was connected to the NodeMCU through its General Purpose I/O pins. The code was developed using the Arduino IDE, resulting in the expected outputs. This project demonstrated that the DHT11 provides accurate measurements of temperature and humidity. To use the DHT11, it is essential to install the DHT library in the Arduino IDE. One practical application of the DHT11 is in weather monitoring systems.