EXPERIMENT NO 6

Title: Interfacing LED with NodeMCU

Components Required:

1. NodeMCU (ESP8266)
2. LED
3. Resistor (220 ohms)
4. Breadboard
5. Jumper wires
6. USB cable for NodeMCU
7. Computer with Arduino IDE installed
8. Access to a web browser

Theory:

NodeMCU (ESP8266):

NodeMCU is an open-source firmware and development board based on the ESP8266 WiFi module. The ESP8266 is a highly integrated chip designed for the needs of a new connected world. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

Specifications:

1. **Microcontroller:** Tensilica L106 32-bit microcontroller
2. **Clock Frequency:** Up to 80 MHz
3. **Wi-Fi Standards:** 802.11 b/g/n
4. **Operating Voltage:** 3.3V (Typical)
5. **Digital I/O Pins:** Typically 17 GPIO pins (varies based on module)
6. **Analog Input Pins:** Typically 1 analog input pin (varies based on module)
7. **Flash Memory:** Typically 512KB to 4MB of integrated flash memory
8. **Operating Temperature:** -40°C to +125°C

ESP8266 Wi-Fi Module: The ESP8266 Wi-Fi module is a highly integrated chip designed for wireless communication. It features a Tensilica L106 32-bit microcontroller, integrated Wi-Fi transceiver, and on-chip memory. The module supports 802.11 b/g/n Wi-Fi standards and can operate as a station, access point, or both simultaneously.

Features of NodeMCU:

Low Power Consumption: The ESP8266 is designed for low power consumption, making it suitable for battery-powered IoT applications.

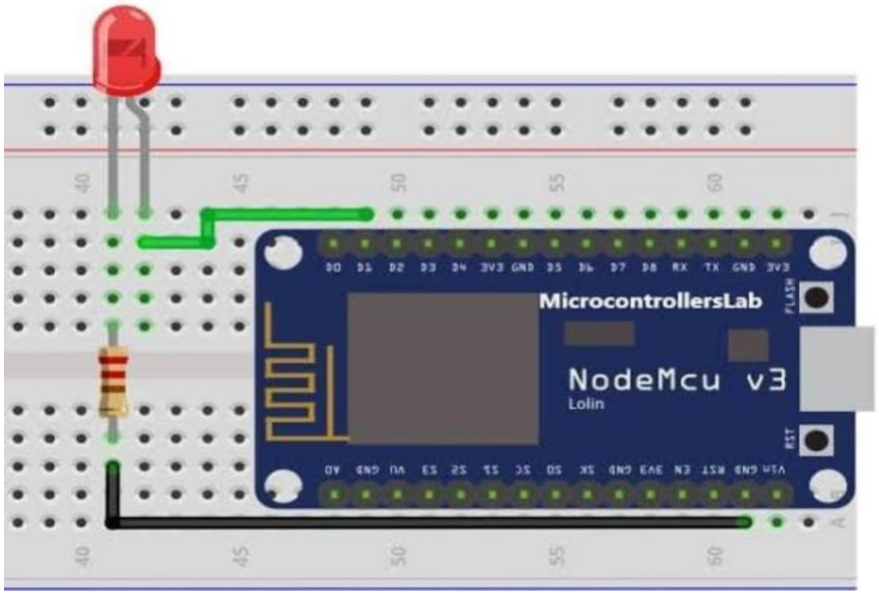
Highly Integrated: Despite its compact size, the ESP8266 integrates a microcontroller, Wi-Fi transceiver, and memory, reducing the need for external components.

OTA (Over-The-Air) Updates :OTA updates, allowing firmware to be updated wirelessly without the need for physical access to the device.

Advanced Networking Features: It supports features such as TCP/IP protocol stack, DNS resolution, DHCP, and UDP for robust network communication.

Wide Range of GPIO Pins: it has variety of GPIO pins, which can be used for digital input/output, analog input, PWM output, and other purposes.

Interfacing diagram:



Procedure:

Setup Arduino IDE:

Install the ESP8266 board in Arduino IDE by going to File -> Preferences

Go to Tools -> Board -> Boards Manager, search for “esp8266” and install the package.

Select the appropriate NodeMCU board from Tools -> Board menu (e.g., NodeMCU 1.0 (ESP-12E Module)).

Write the Code:

Open Arduino IDE and create a new sketch.

Write the code to configure the GPIO pin connected to the LED as an output and set up a web server to control the LED.

Code:

```
Const int ledPin = 2;

Void setup() {

 pinMode(ledPin, OUTPUT);

}

Void loop() {

  digitalWrite(ledPin, HIGH);

  Delay(1000);

  digitalWrite(ledPin, LOW);

  Delay(1000);

}
```

Upload the Code:

Connect the NodeMCU to your computer using a USB cable.
Select the appropriate port from **`Tools -> Port`** menu.
Upload the code to the NodeMCU by clicking on the upload button (right arrow).

Conclusion:

In this experiment, you successfully interfaced an LED with NodeMCU