Experiment 8

Aim: 1.To Control LED & Devices from anywhere using ThingSpeak & NodeMcu

2. Getting Started with ThingSpeak Cloud Code.

Objective: 1) To know about Programming the NODE MCU.

2) To Control a LED Light by creating a HTTP web server running on ECP8266.

Software/Hardware Requirements:

Sr.No	Components	specification	Quantity
1.	NodeMCU	ESP8266	1
2.	LED	10 mm	3
3.	Jumper Wires	-	As per the required
4.	Breadboard		1

Theory:

The ESP8266 NodeMCU is a widely used Wi-Fi module that is programmable for a variety of Internet of Things (IoT) applications. For your experiment, the goal is to control an LED using the NodeMCU by establishing a web server while in station mode. Here's a breakdown of the essential concepts:

1. ESP8266 NodeMCU: This is an affordable microcontroller equipped with built-in Wi-Fi capabilities. It can be programmed using the Arduino IDE or other platforms, making it suitable for IoT projects.

2. Station Mode: In station mode, the NodeMCU connects to an existing Wi-Fi network, enabling communication with other devices and the internet. This connection is crucial for creating a web server since it requires internet access to serve web pages.

3. HTTP Web Server: An HTTP (Hypertext Transfer Protocol) web server is a software application that awaits incoming HTTP requests from clients, typically web browsers. It responds by delivering HTML pages or other resources. In your experiment, you'll construct a basic HTTP server on the NodeMCU to manage the LED.

4. LED Control: To govern an LED, you'll need to link it to one of the GPIO (General Purpose Input/Output) pins on the NodeMCU. You can switch the LED on and off by altering the state of the GPIO pin.

Gather the Hardware and Software:

1) Connect the Hardware

- Connect the anode (longer lead) of the LED to one of the GPIO pins on the NodeMCU (e.g., D2).
- Connect the cathode (shorter lead) of the LED to the ground (GND) pin on the NodeMCU.
- Ensure the NodeMCU is powered using a USB cable or an external power source.

2) Write the Arduino Sketch

```
//Welcome to E for Engineer---SUBSCRIBE Now

#include "ThingSpeak.h"

#include <ESP8266WiFi.h>


//Replace your wifi credentials here

const char* ssid     = "Sakshi";//Replace with your Wifi Name

const char* password = "sakshi3040";// Replace with your wifi Password


//change your channel number here

unsigned long channel =2266186;//Replace with your own ThingSpeak Account Channle ID


//1,2 and 3 are channel fields. You don't need to change if you are following this tutorial. However, you can modify it according to your application

unsigned int led1 = 1;

unsigned int led2 = 2;

unsigned int led3 = 3;


WiFiClient  client;
```

```
void setup() {
  Serial.begin(9600);
  delay(100);

  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  digitalWrite(D1, 0);
  digitalWrite(D2, 0);
  digitalWrite(D3, 0);
  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
```

```
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.print("Netmask: ");
  Serial.println(WiFi.subnetMask());
  Serial.print("Gateway: ");
  Serial.println(WiFi.gatewayIP());
  ThingSpeak.begin(client);

}

void loop() {

  //get the last data of the fields
  int led_1 = ThingSpeak.readFloatField(channel, led1);
  int led_2 = ThingSpeak.readFloatField(channel, led2);
  int led_3 = ThingSpeak.readFloatField(channel, led3);

  if(led_1 == 1){
    digitalWrite(D1, 1);
    Serial.println("D1 is On..!");
  }
  else if(led_1 == 0){
    digitalWrite(D1, 0);
    Serial.println("D1 is Off..!");
  }
```

```
 if(led_2 == 1){
   digitalWrite(D2, 1);
   Serial.println("D2 is On..!");
 }
 else if(led_2 == 0){
   digitalWrite(D2, 0);
   Serial.println("D2 is Off..!");
 }

 if(led_3 == 1){
   digitalWrite(D3, 1);
   Serial.println("D3 is On..!");
 }
 else if(led_3 == 0){
   digitalWrite(D3, 0);
   Serial.println("D3 is Off..!");
 }

 Serial.println(led_1);
 Serial.println(led_2);
 Serial.println(led_3);
 delay(5000);

}
```

3) Upload the Sketch

- Connect your NodeMCU to your computer via USB, select the correct board in the Arduino IDE and upload the sketch to the NodeMCU.

4) Connect to the Web Server

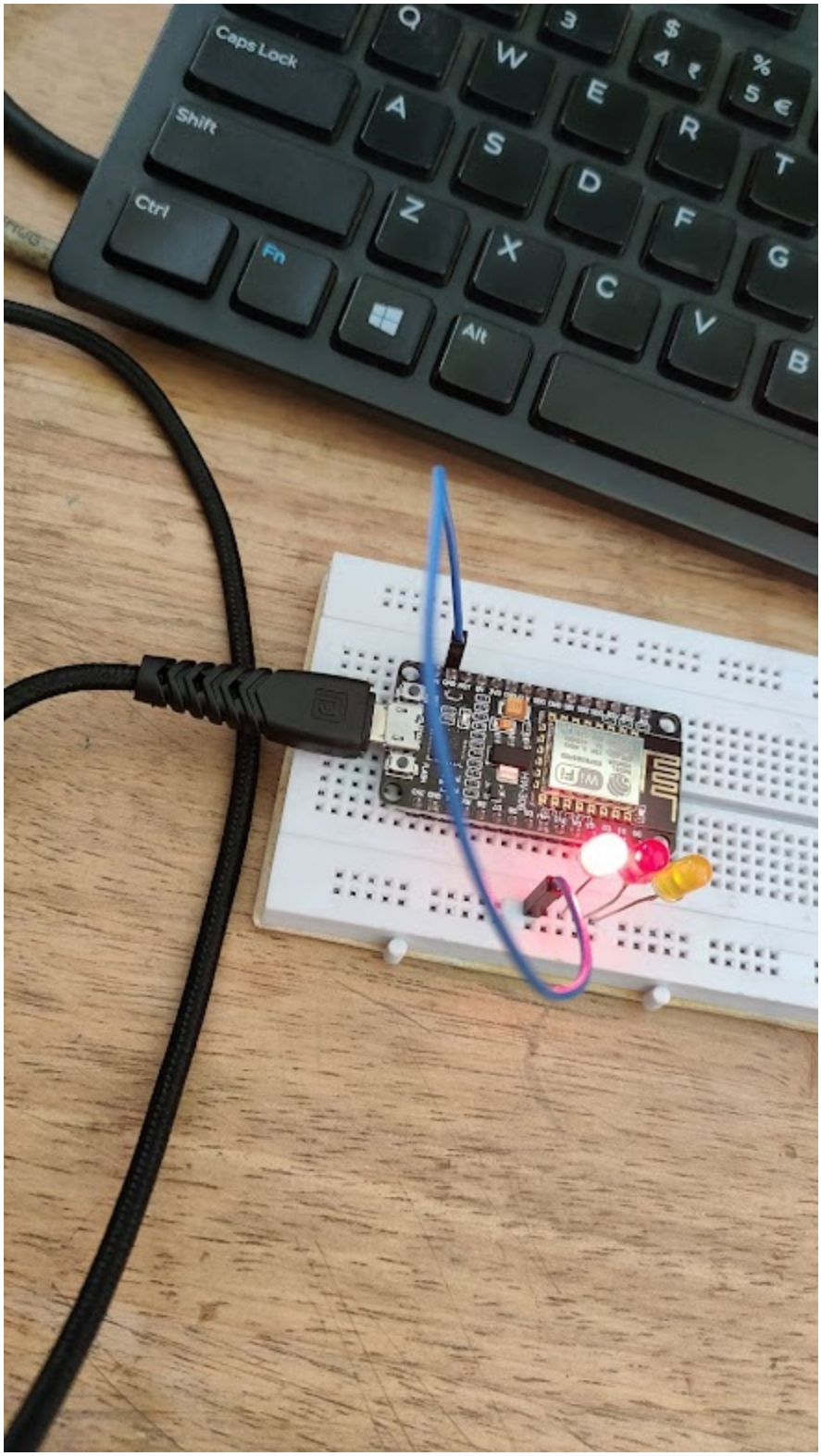
- After the upload is complete, open the Arduino IDE's Serial Monitor, and you should see the NodeMCU connecting to your Wi-Fi network.
- Once connected, it will display the NodeMCU's local IP address. Enter this IP address in a web browser on your computer or mobile device.

5) Control the LED

- You should now see a web page with a button to toggle the LED on and off. Click the button to control the LED remotely. The LED's state will change according to the button's status on the web page.

```
10:40:20.531 -> D2 is On..!
10:40:20.531 -> D3 is On..!
10:40:20.531 -> 1
10:40:20.531 -> 1
10:40:20.531 -> 1
10:40:29.271 -> D1 is On..!
10:40:29.337 -> D2 is On..!
10:40:29.337 -> D3 is On..!
10:40:29.337 -> 1
10:40:29.337 -> 1
10:40:29.337 -> 1
10:40:37.215 -> D1 is On..!
10:40:37.215 -> D2 is On..!
10:40:37.215 -> D3 is On..!
10:40:37.215 -> 1
10:40:37.215 -> 1
10:40:37.215 -> 1
10:40:45.275 -> D1 is On..!
10:40:45.321 -> D2 is On..!
10:40:45.321 -> D3 is On..!
10:40:45.321 -> 1
10:40:45.321 -> 1
10:40:45.321 -> 1
10:40:53.911 -> D1 is On..!
10:40:53.957 -> D2 is On..!
10:40:53.957 -> D3 is On..!
10:40:53.957 -> 1
10:40:53.957 -> 1
10:40:53.957 -> 1
```

Output


Caps Lock
Shift
Ctrl
Fn
Alt