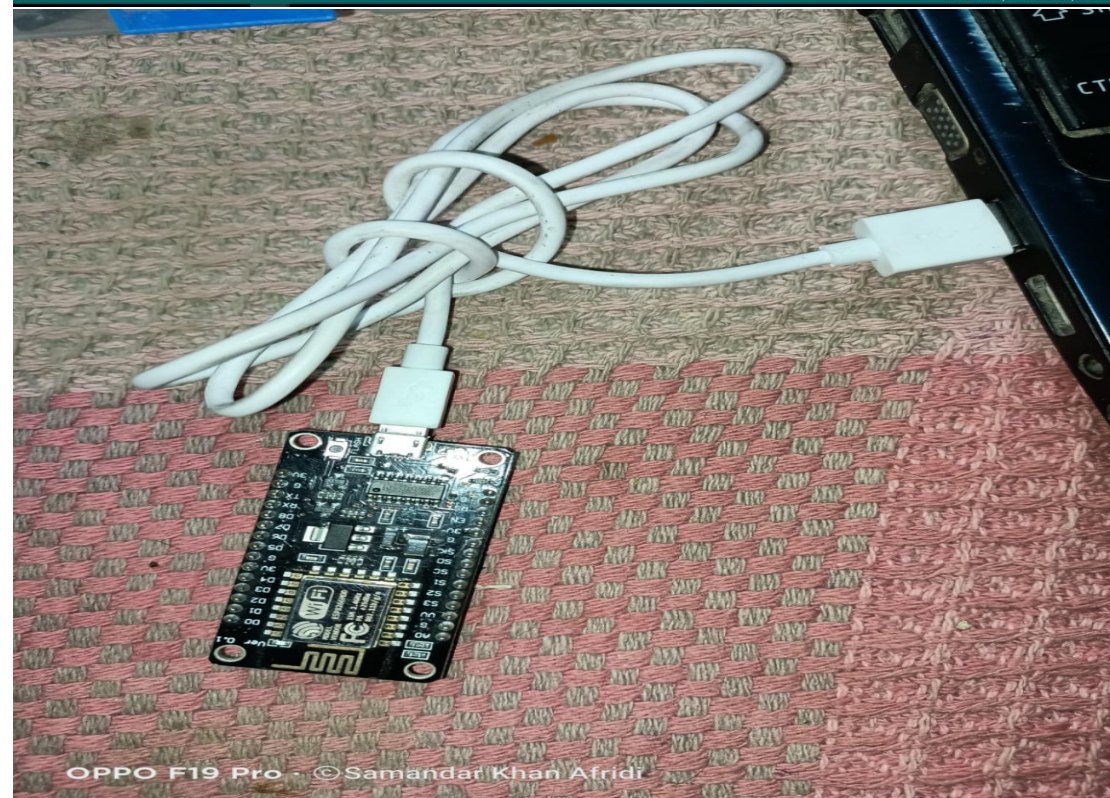
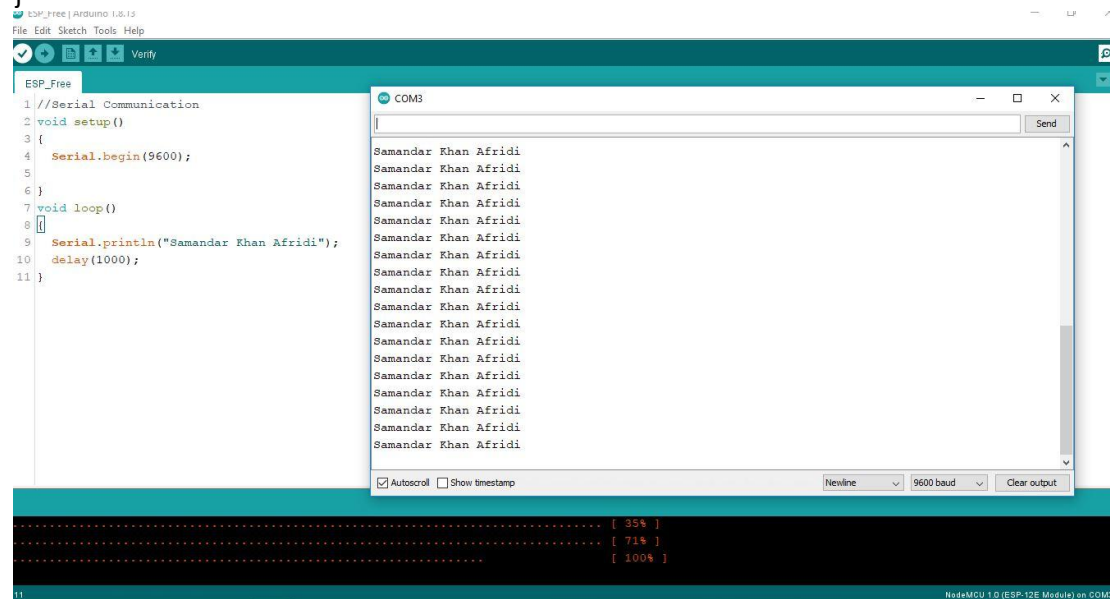


Name: Mr. Samandar Khan Afridi
Program: Internet Of Things (IoT)
Booklet: All NodeMCU Programs
Date: 15-Dec-2021

No#00//Serial Communication

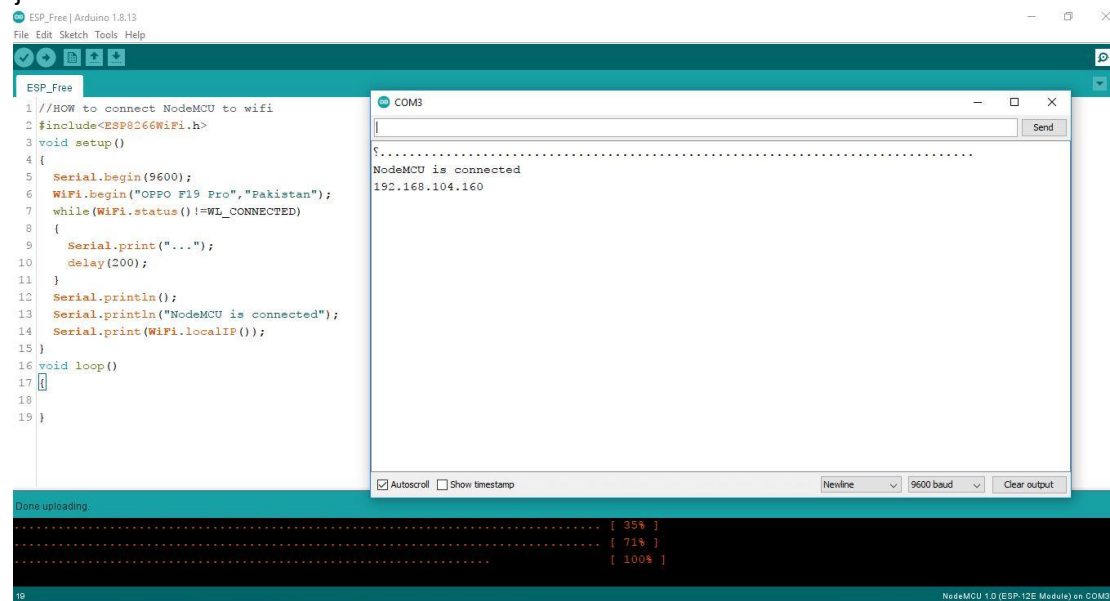
```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Samandar Khan Afridi");
  delay(1000);
}
```



No#01//HOW to connect NodeMCU to wifi

```
#include<ESP8266WiFi.h>
void setup()
{
  Serial.begin(9600);
  WiFi.begin("OPPO F19 Pro","Pakistan");
  while(WiFi.status()!=WL_CONNECTED)
  {
    Serial.print("...");
    delay(200);
  }
  Serial.println();
  Serial.println("NodeMCU is connected");
  Serial.print(WiFi.localIP());
}
void loop()
{
}
```



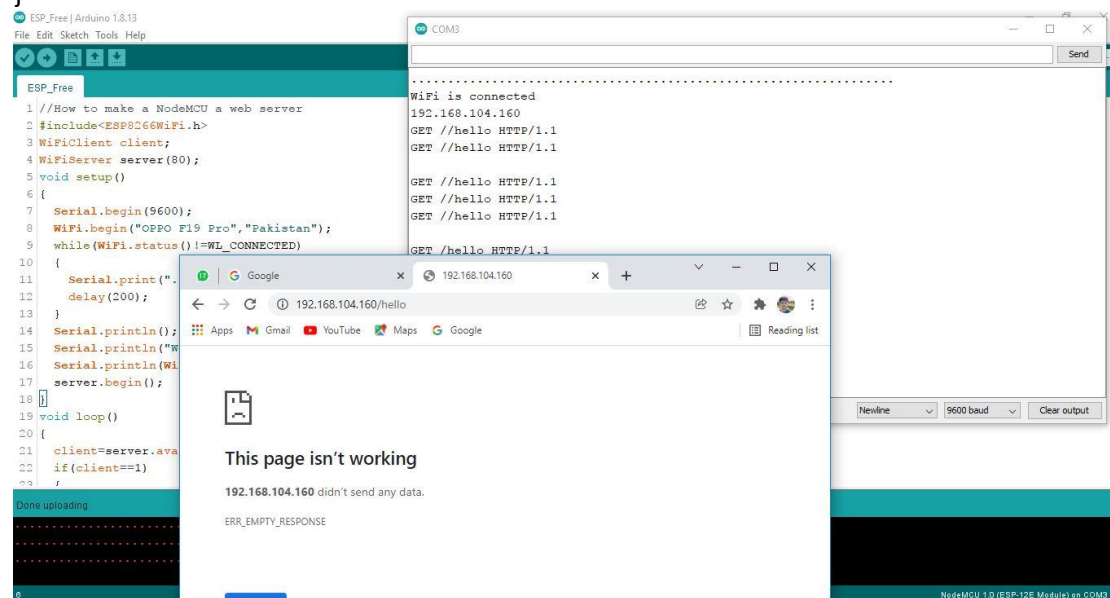
No#02//How to make a NodeMCU a web server

```
#include<ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
void setup()
{
  Serial.begin(9600);
  WiFi.begin("OPPO F19 Pro","Pakistan");
  while(WiFi.status()!=WL_CONNECTED)
  {
```

```

        Serial.print("...");
        delay(200);
    }
    Serial.println();
    Serial.println("WiFi is connected");
    Serial.println(WiFi.localIP());
    server.begin();
}
void loop()
{
    client=server.available();
    if(client==1)
    {
        String request=client.readStringUntil('\n');
        Serial.println(request);
    }
}
}

```



No#03//Led ON OFF NodeMCU

```

#include<ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
#define int D5
void setup()
{
    Serial.begin(9600);
    WiFi.begin("OPPO F19 Pro","Pakistan");
    while(WiFi.status()!=WL_CONNECTED)
    {
        Serial.print(".");
        delay(100);
    }
}

```



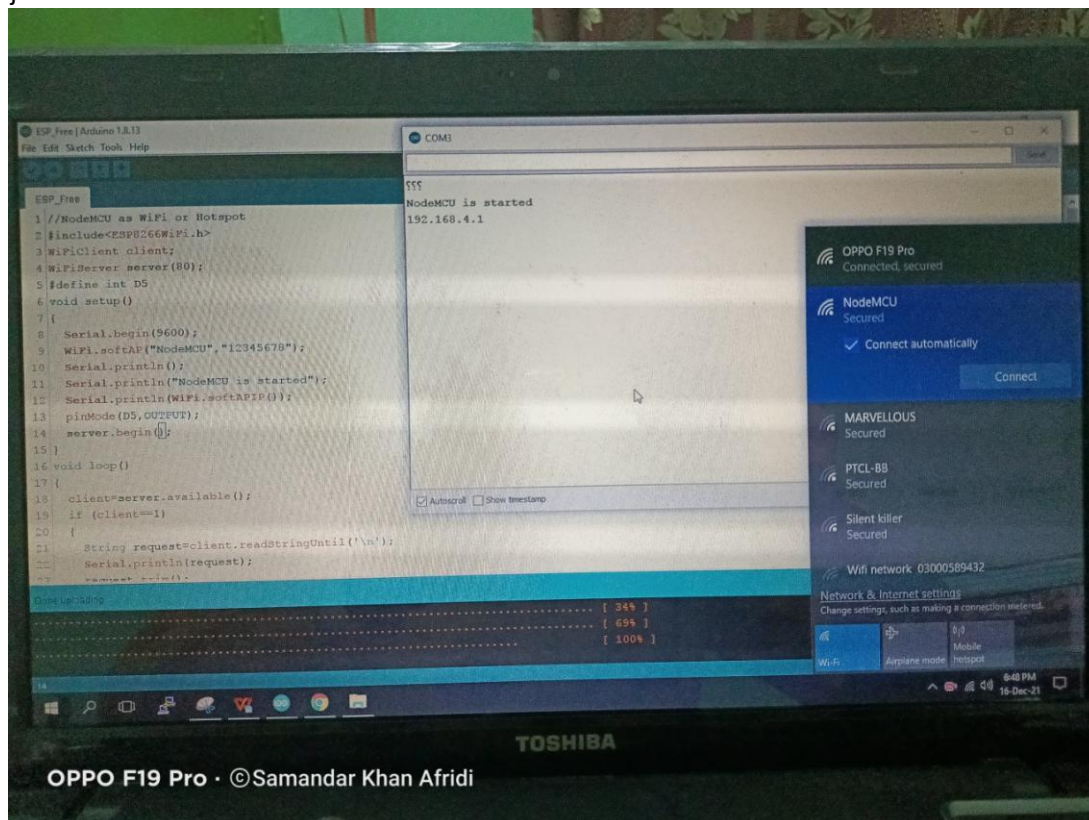

No#04//NodeMCU as WiFi or Hotspot

```
#include<ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
#define int D5
void setup()
{
  Serial.begin(9600);
  WiFi.softAP("NodeMCU","12345678");
  Serial.println();
  Serial.println("NodeMCU is started");
  Serial.println(WiFi.softAPIP());
  pinMode(D5,OUTPUT);
  server.begin(9600);
}
void loop()
{
  client=server.available();
```

```

if (client==1)
{
    String request=client.readStringUntil('\n');
    Serial.println(request);
    request.trim();
    if (request=="GET /ledon HTTP/1.1")
    {
        digitalWrite(D5,HIGH);
    }
    if (request=="GET /ledoff HTTP/1.1")
    {
        digitalWrite(D5,LOW);
    }
}
}
}

```



No#05//ESP8266 GET request ThingSpeak

```

#include<ESP8266WiFi.h>
#include<DHT.h>
#include<ESP8266HTTPClient.h>

```

```

DHT dht(D5,DHT11);

```

```

String host="api.thingspeak.com";
int httpPort=80;

```

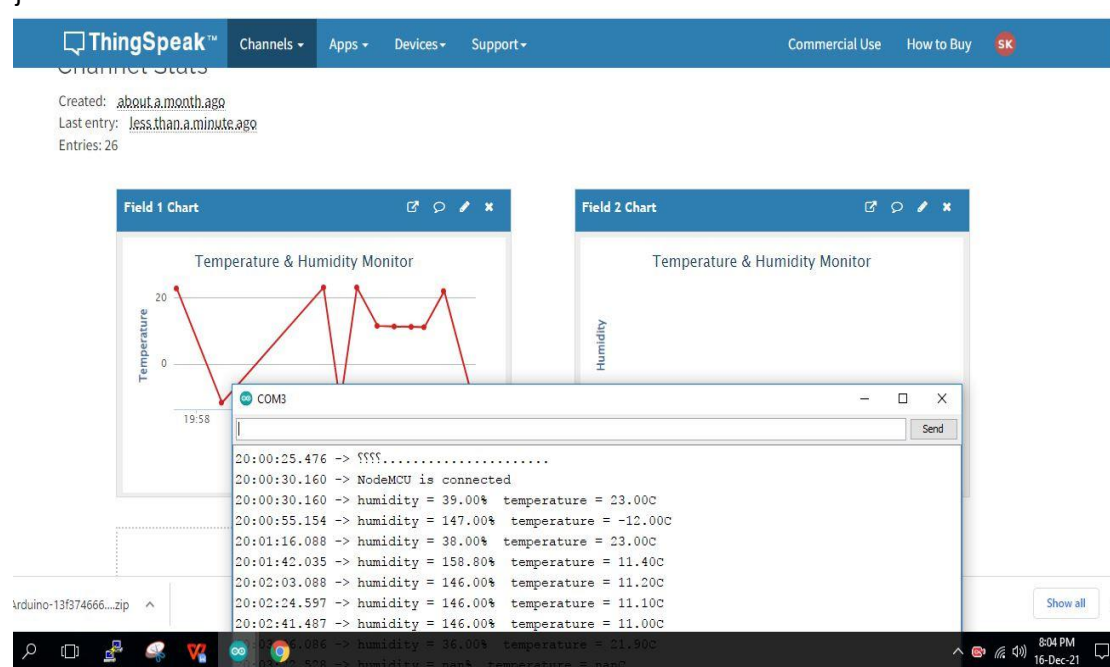
```
String url1="/update?api_key=SKZPFTC6IEWCGJAI&field1=";  
String url2="/update?api_key=SKZPFTC6IEWCGJAI&field2=";
```

```
HTTPClient HTTP;  
WiFiClient client;
```

```
void setup()  
{  
  Serial.begin(9600);  
  WiFi.begin("OPPO F19 Pro","Pakistan");  
  while(WiFi.status()!=WL_CONNECTED)  
  {  
    Serial.print(".");  
    delay(200);  
  }  
  Serial.println();  
  Serial.println("NodeMCU is connected");  
  Serial.println(WiFi.localIP());  
  dht.begin();  
}  
void loop()  
{  
  float h=dht.readHumidity();  
  float t=dht.readTemperature();  
  Serial.println("Temperature:"+(String) t);  
  Serial.println("Humidity:"+(String) h);  
  
  String link1 = url1 + (String) t;  
  HTTP.begin(client,host,httpPort,link1);  
  int httpCode=HTTP.GET();  
  Serial.println(httpCode);  
  delay(2000);  
  
  String link2 = url2 + (String) h;  
  HTTP.begin(client,host,httpPort,link2);  
  httpCode=HTTP.GET();  
  Serial.println(httpCode);  
  delay(2000);  
}
```

No#06//ESP ThingSpeak Temp:&Humi: Monitor

```
#include<ESP8266WiFi.h>  
#include<DHT.h>  
#include<ThingSpeak.h>  
WiFiClient client;  
#define dhtPin D4
```


[illegible]

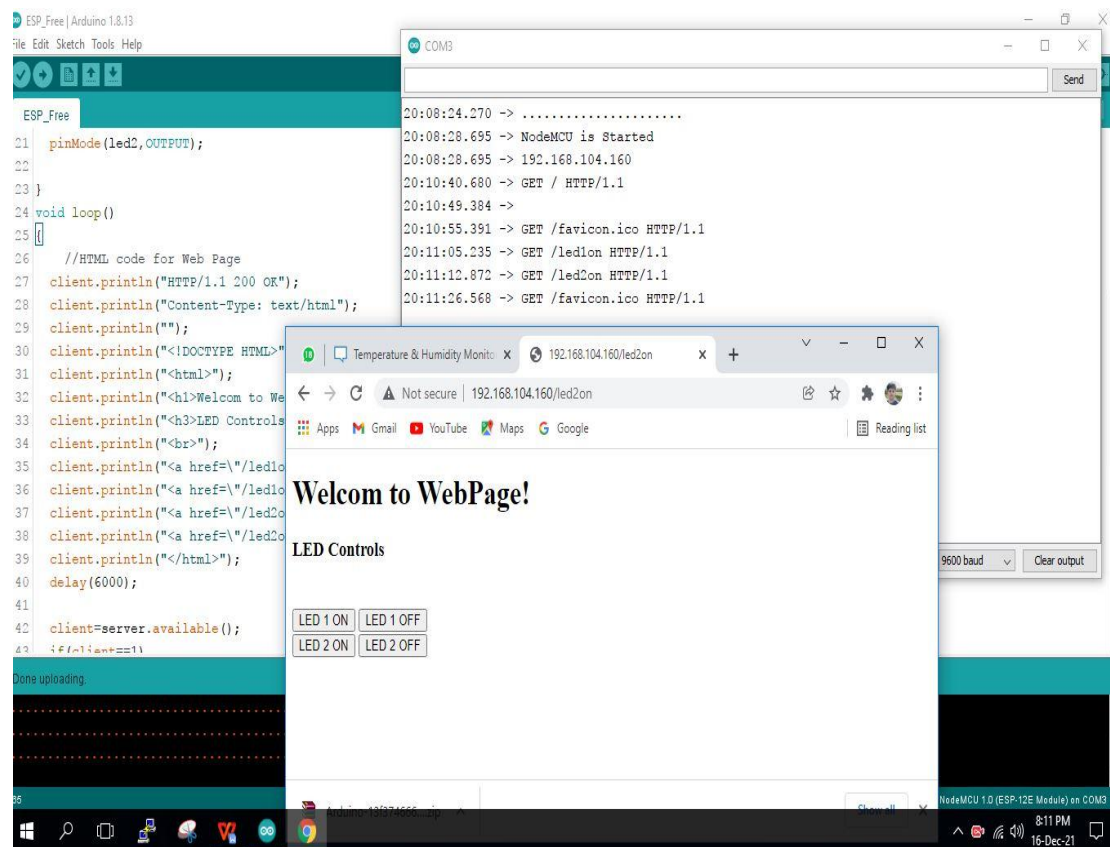
No#07//LED ON OFF HTML Webpage

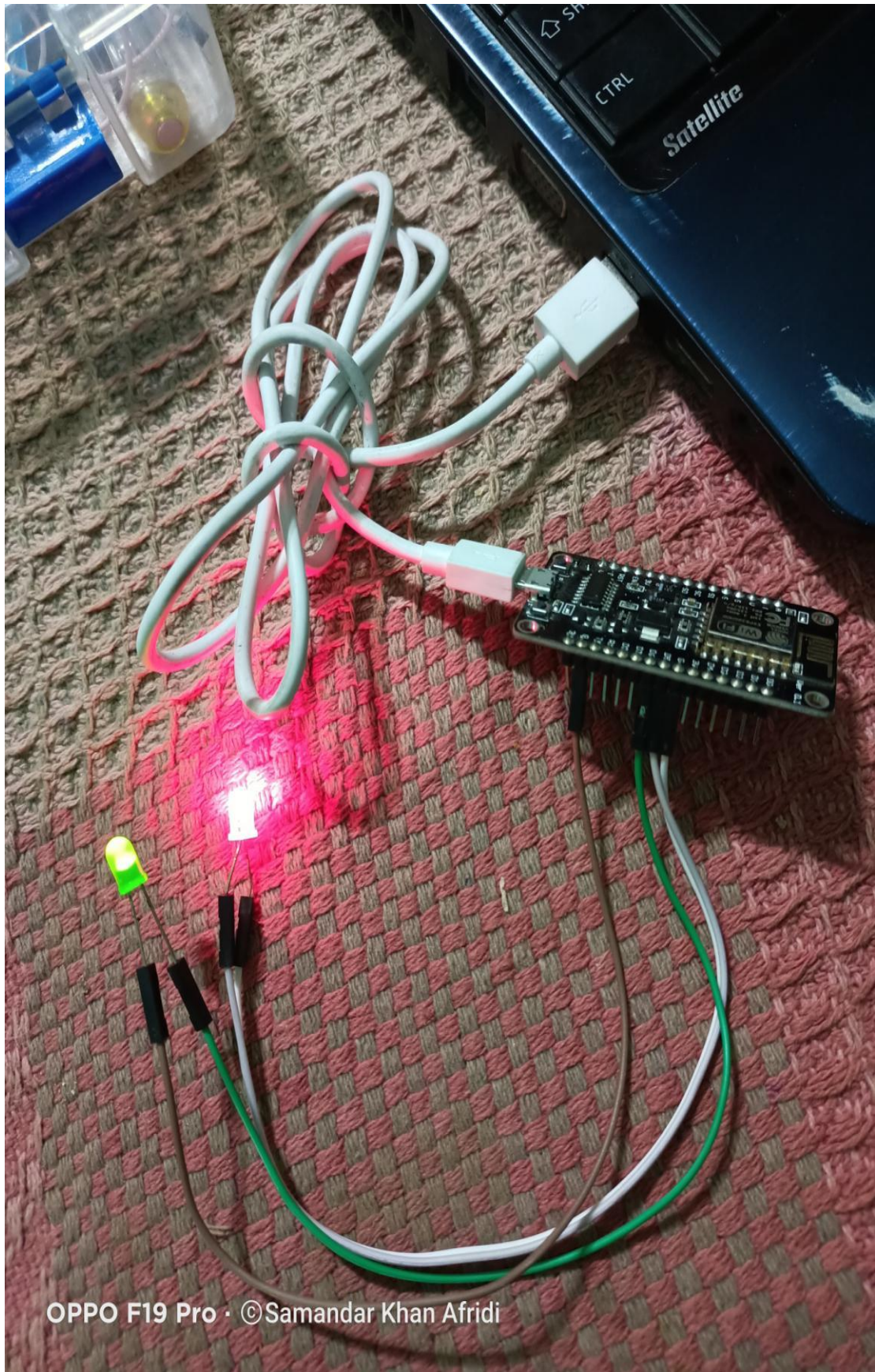
```
#include<ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
#define led1 D5
#define led2 D6
void setup()
{
    Serial.begin(9600);
    WiFi.begin("OPPO F19 Pro","Pakistan");
    while(WiFi.status()!=WL_CONNECTED)
    {
        Serial.print(".");
        delay(200);
    }
    Serial.println();
    Serial.println("NodeMCU is Started ");
    Serial.println(WiFi.localIP());
    server.begin();
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
}
void loop()
{
    //HTML code for Web Page
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("<h1>Welcom to WebPage!</h1>");
    client.println("<h3>LED Controls</h3>");
    client.println("<br>");
    client.println("<a href=\"/led1on\"><button>LED 1 ON</button></a>");
    client.println("<a href=\"/led1off\"><button>LED 1 OFF</button></a><br/>");
    client.println("<a href=\"/led2on\"><button>LED 2 ON</button></a>");
    client.println("<a href=\"/led2off\"><button>LED 2 OFF</button></a><br/>");
    client.println("</html>");
    delay(6000);
}
```

```

client=server.available();
if(client==1)
{
    String request=client.readStringUntil('\n');
    Serial.println(request);
    request.trim();
    if(request=="GET /led1on HTTP/1.1")
    {
        digitalWrite(led1,HIGH);
    }
    if(request=="GET /led1off HTTP/1.1")
    {
        digitalWrite(led1,LOW);
    }
    if(request=="GET /led2on HTTP/1.1")
    {
        digitalWrite(led2,HIGH);
    }
    if(request=="GET /led2off HTTP/1.1")
    {
        digitalWrite(led2,LOW);
    }
}
}

```





No#08//ESP ThingSpeak Potentiometer

```
#include "ThingSpeak.h"
//#include "secrets.h" // secrets.h code is already given above in the first example
#include <ESP8266WiFi.h>
#define SECRET_SSID "OPPO F19 Pro"           // replace MySSID with your WiFi
network name
#define SECRET_PASS "Pakistan"           // replace MyPassword with your WiFi password

#define SECRET_CH_ID 1512414               // replace 0000000 with your channel
number
#define SECRET_WRITE_APIKEY "AO5OHWAFWSR83VWZ" // replace XYZ with
your channel write API Key
char ssid[] = SECRET_SSID; // your network SSID (name)
char pass[] = SECRET_PASS; // your network password
int keyIndex = 0;          // your network key Index number (needed only for
WEP)
WiFiClient client;

unsigned long myChannelNumber = SECRET_CH_ID;
const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

// Initialize our values
int number1 = 0;
int number2 = random(0,100);
int number3 = random(0,100);
int number4 = random(0,100);
String myStatus = "";

// sensor
int Pot = A0;
void setup() {
  Serial.begin(115200); // Initialize serial
  pinMode(Pot,INPUT);
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client); // Initialize ThingSpeak
}

void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line
if using open or WEP network
      Serial.print(".");
```

```

        delay(5000);
    }
    Serial.println("\nConnected.");
}
int data = map(analogRead(Pot),0,1023,0,255);
    // set the fields with the values
    ThingSpeak.setField(1, data);
// ThingSpeak.setField(2, number2);
// ThingSpeak.setField(3, number3);
// ThingSpeak.setField(4, number4);

    // figure out the status message
    if(number1 > number2){
        myStatus = String("field1 is greater than field2");
    }
    else if(number1 < number2){
        myStatus = String("field1 is less than field2");
    }
    else{
        myStatus = String("field1 equals field2");
    }

    // set the status
    ThingSpeak.setStatus(myStatus);

    // write to the ThingSpeak channel
    int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    if(x == 200){
        Serial.println("Channel update successful.");
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " + String(x));
    }

    // change the values
    number1++;
    if(number1 > 99){
        number1 = 0;
    }
    number2 = random(0,100);
    number3 = random(0,100);
    number4 = random(0,100);

    delay(20000); // Wait 20 seconds to update the channel again
}

```

No#09//ESP_Cloud_Write_On_ThingSpeak

```
//ESP ThingSpeak Temp:&Humi: Monitor
#include<ESP8266WiFi.h>
#include<DHT.h>
#include<ThingSpeak.h>
WiFiClient client;
#define dhtPin D4
DHT dht(dhtPin,DHT11);
const int myChannelNumber=1546421;
char myWriteAPIKey[]="SKZPFTC6IEWCGJAI";
void setup()
{
  Serial.begin(9600);
  WiFi.begin("OPPO F19 Pro","Pakistan");
  while(WiFi.status()!=WL_CONNECTED)
  {
    Serial.print(".");
    delay(200);
  }
  Serial.println();
  Serial.println("NodeMCU is connected");
  ThingSpeak.begin(client);
}
void loop()
{
  int h =random(50,100);//dht.readHumidity();
  int t =random(50,100);//dht.readTemperature();
  Serial.print("humidity = ");
  Serial.print(h);
  Serial.print("%  ");
  Serial.print("temperature = ");
  Serial.print(t);
  Serial.println("C  ");
  ThingSpeak.setField(1,h);
  ThingSpeak.setField(2,t);
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }
  delay(5000);
}
```

No#10//ESP_Cloud_Read_On_ThingSpeak

```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
const char ssid[] = "OPPO F19 Pro"; // your network SSID (name)
const char pass[] = "Pakistan"; // your network password
int statusCode=0;
WiFiClient client;

//-----Channel Details-----//
unsigned long counterChannelNumber = 1546421; // Channel ID
const char * myCounterReadAPIKey = "L2ARX4VR5WCQYYY2"; // Read API Key
const int FieldNumber1 = 1; // The field you wish to read
const int FieldNumber2 = 2; // The field you wish to read
//-----//

void setup()
{
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
}

void loop()
{
  //----- Network -----//
  if (WiFi.status() != WL_CONNECTED)
  {
    Serial.print("Connecting to ");
    Serial.print(ssid);
    Serial.println(" ....");
    while (WiFi.status() != WL_CONNECTED)
    {
      WiFi.begin(ssid, pass);
      delay(5000);
    }
    Serial.println("Connected to Wi-Fi Succesfully.");
  }
  //----- End of Network connection-----//

  //----- Channel 1 -----//
  long temp = ThingSpeak.readLongField(counterChannelNumber, FieldNumber1,
myCounterReadAPIKey);
  //statusCode = ThingSpeak.getLastReadStatus();
  if (statusCode == 200)
  {
    Serial.print("Temperature: ");
    Serial.println(temp);
  }
}
```



```

    }
    else
    {
        Serial.println("Unable to read channel / No internet connection");
    }
    delay(100);
    //----- End of Channel 1 -----//

    //----- Channel 2 -----//
    long humidity = ThingSpeak.readLongField(counterChannelNumber, FieldNumber2,
myCounterReadAPIKey);
    statusCode = ThingSpeak.getLastReadStatus();
    if (statusCode == 200)
    {
        Serial.print("Humidity: ");
        Serial.println(humidity);
    }
    else
    {
        Serial.println("Unable to read channel / No internet connection");
    }
    delay(100);
    //----- End of Channel 2 -----//
}

```

No#11//connection signal Strength

```

#include<ESP8266WiFi.h>
const char* SSID="OPPO F19 Pro";
const char* password="Pakistan";
void setup()
{
    Serial.begin(115200);
    Serial.println();
    Serial.print("Setting WiFi Mode");
    WiFi.mode(WIFI_STA);
    WiFi.begin(SSID,password);
    Serial.print("Connecting to ...");
    Serial.print(SSID);
    while(WiFi.status()!=WL_CONNECTED)
    {
        delay(500);
        Serial.print("...");
    }
    Serial.print("connected IP address");
    Serial.println(WiFi.localIP());
}

```

```

void loop()
{
  Serial.printf("Single Strength in dB=%d \n",WiFi.RSSI());
  delay(3000);
}

```

No#12//Count 1 to 10

```

#include<ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
#define led D5
void setup()
{
  Serial.begin(9600);
  WiFi.begin("OPPO F19 Pro","Pakistan");
  while(WiFi.status()!=WL_CONNECTED)
  {
    delay(200);
    Serial.print("...");
  }
  Serial.println();
  Serial.println("NodeMCU is connected");
  Serial.println(WiFi.localIP());
  server.begin();
  pinMode(led,OUTPUT);
}
void loop()
{
  client=server.available();
  if(client==1)
  {
    String request=client.readStringUntil('\n');
    Serial.println(request);
    request.trim();
    if(request=="GET /Count1to10 HTTP/1.1")
    {
      for(int i=1;i<=10;i++)
      {
        Serial.println(i);
      }
    }
  }
}
}

```

No#13//Getting MAC address

//SamandarKhanAfridi

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

```
void setup()
```

```
{
```

```
    delay(10);
```

```
    Serial.begin(115200);
```

```
}
```

```
void loop()
```

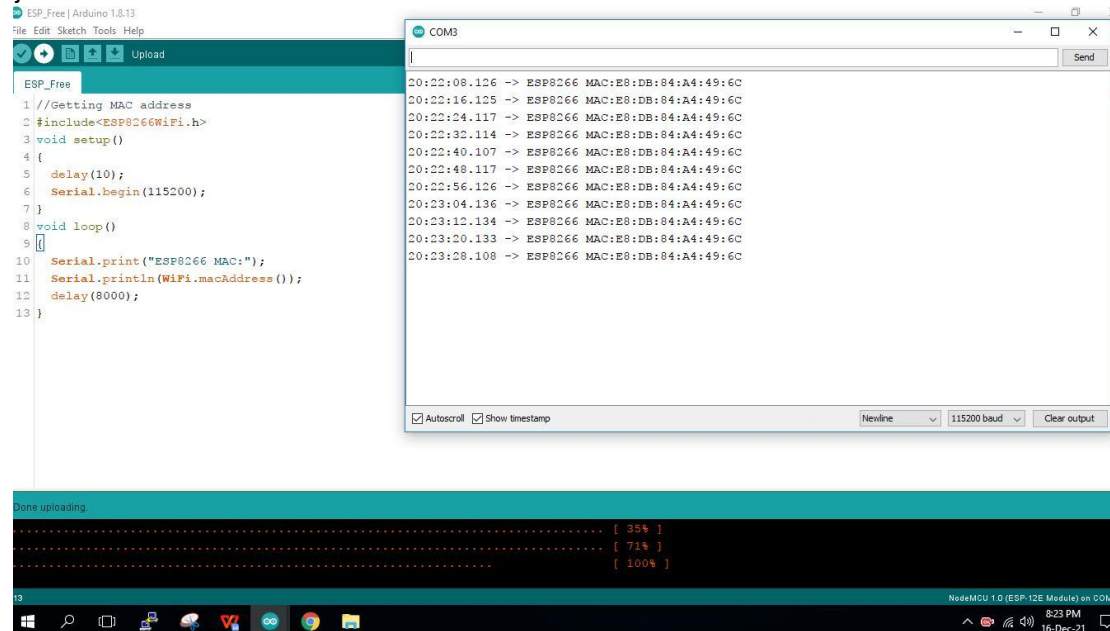
```
{
```

```
    Serial.print("ESP8266 MAC:");
```

```
    Serial.println(WiFi.macAddress());
```

```
    delay(8000);
```

```
}
```



No#14//HTTP LED ON OFF

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

```
#include<WiFiClient.h>
```

```
#include<ESP8266WebServer.h>
```

```
#define int D2
```

```
/*const char MAIN_page[] PROGMEM = R"=====(
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>Circuits4you</h2>
```

```
<h3> HTML Form ESP8266</h3>
```

```
<form action="/action_page">
```

```
    First name:<br>
```

```

    <input type="text" name="firstname" value="Mickey">
    <br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse">
    <br><br>
    <input type="submit" value="Submit">
</form>

```

```

</body>

```

```

</html>

```

```

)=====";*/

```

```

const char MAIN_page[] PROGMEM=R"=====

```

```

<HTML>

```

```

    <HEAD>

```

```

        <TITLE>POST request Demo</TITLE>

```

```

    </HEAD>

```

```

    <BODY>

```

```

        <CENTER>

```

```

            <B>LED Status=@@LEDState@@</B>

```

```

        </CENTER>

```

```

        <FORM method="POST" action="/">

```

```

            <BUTTON name="LED"

```

```

            value="1">ON</BUTTON>

```

```

            <BUTTON name="LED"

```

```

            value="0">OFF</BUTTON>

```

```

        </FORM>

```

```

        <marquee behavior="alternate">NodeMCU ESP8266 Communication Methods and
        protocols</marquee>

```

```

    </BODY>

```

```

</HTML>

```

```

)=====";

```

```

//SSID and Password of your WiFi router

```

```

const char* ssid = "OPPO F19 Pro";

```

```

const char* password = "Pakistan";

```

```

ESP8266WebServer server(80); //Server on port 80

```

```

//=====

```

```

// This routine is executed when you open its IP in browser

```

```

//=====

```

```

void handleRoot() {

```

```

    String s = MAIN_page; //Read HTML contents

```

```

    server.send(200, "text/html", s); //Send web page

```

```

    String LEDstate ;

```

```

        //webPage = htmlPage;

```

```

        LEDstate = server.arg("LED");

```



```

    Serial.print("Argument Received:");
    Serial.println(LEDstate);
    if(LEDstate=="1"){
        digitalWrite(D2,HIGH);
        s.replace("@@LEDstate@", "ON");
    }
    if(LEDstate=="0")
    {
        digitalWrite(D2,LOW);
        s.replace("@@LEDstate", "OFF");
    }
    //server.send(200,"text/html",webPage);

}
//=====
// This routine is executed when you press submit
//=====
void handleForm() {
    // String firstName = server.arg("firstname");
    // String lastName = server.arg("lastname");
    //
    // Serial.print("First Name:");
    // Serial.println(firstName);
    //
    // Serial.print("Last Name:");
    // Serial.println(lastName);

    String s = "<a href='/'> Go Back </a>";
    server.send(200, "text/html", s); //Send web page
}
//=====
//                      SETUP
//=====
void setup(){
    Serial.begin(9600);
    pinMode(D2,OUTPUT);
    WiFi.begin(ssid, password);    //Connect to your WiFi router
    Serial.println("");

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

    //If connection successful show IP address in serial monitor
    Serial.println("");

```

```

Serial.print("Connected to ");
Serial.println("WiFi");
Serial.print("IP address: ");
Serial.println(WiFi.localIP()); //IP address assigned to your ESP

server.on("/", handleRoot); //Which routine to handle at root location
server.on("/action_page", handleForm); //form action is handled here

server.begin(); //Start server
Serial.println("HTTP server started");
}
//=====
//                      LOOP
//=====
void loop(void){
    server.handleClient(); //Handle client requests
}

```

No#15//HTTP post

```

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
const char* wifiName = "OPPO F19 Pro";
const char* wifiPass = "Pakistan";
//Web Server address to read/write from
const char *host = "http://httpbin.org/post";
void setup() {
    Serial.begin(115200);
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(wifiName);
    WiFi.begin(wifiName, wifiPass);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("Wi-Fi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP()); //You can get IP address assigned to ESP
}
void loop() {
    HTTPClient http; //Declare object of class HTTPClient
    WiFiClient Client;
    String ADCData = String(analogRead(A0));
}

```

```

String postData;
//POST Data
postData = "data=" + ADCData + "&sensor=temperature";
Serial.print("Post Data:");
Serial.println(postData);
http.begin(Client,host);//Specify request destination
http.addHeader("Content-Type", "application/x-www-form-urlencoded");//Specify
content-type header
int httpCode = http.POST(postData);//Send the request
String payload = http.getString();//Get the response payload
Serial.print("Response Code:"); //200 is OK
Serial.println(httpCode);//Print HTTP return code
Serial.print("Returned data from Server:");
Serial.println(payload);//Print request response payload
http.end();//Close connection
delay(20000);//POST Data at every 5 seconds
}

```

No#16//LED Controlling by Sliding button Webpage

```

// Import required libraries
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>

// Replace with your network credentials
const char* ssid = "OPPO F19 Pro";
const char* password = "Pakistan";

#define output 2

String sliderValue = "0";

const char* PARAM_INPUT = "value";

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>ESP Web Server</title>
  <style>
    html {font-family: Arial; display: inline-block; text-align: center;}
    h2 {font-size: 2.3rem;}
    p {font-size: 1.9rem;}

```

```

    body {max-width: 400px; margin:0px auto; padding-bottom: 25px;}
    .slider { -webkit-appearance: none; margin: 14px; width: 360px; height: 25px;
background: #FFD65C;
    outline: none; -webkit-transition: .2s; transition: opacity .2s;}
    .slider::-webkit-slider-thumb {-webkit-appearance: none; appearance: none;
width: 35px; height: 35px; background: #003249; cursor: pointer;}
    .slider::-moz-range-thumb { width: 35px; height: 35px; background: #003249;
cursor: pointer; }
</style>
</head>
<body>
    <h2>ESP Web Server</h2>
    <p><span id="textSliderValue">%SLIDERVALUE%</span></p>
    <p><input type="range" onchange="updateSliderPWM(this)" id="pwmSlider"
min="0" max="1023" value="%SLIDERVALUE%" step="1" class="slider"></p>
<script>
function updateSliderPWM(element) {
    var sliderValue = document.getElementById("pwmSlider").value;
    document.getElementById("textSliderValue").innerHTML = sliderValue;
    console.log(sliderValue);
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "/slider?value="+sliderValue, true);
    xhr.send();
}
</script>
</body>
</html>
)rawliteral";

```

// Replaces placeholder with button section in your web page

```

String processor(const String& var){
    //Serial.println(var);
    if (var == "SLIDERVALUE"){
        return sliderValue;
    }
    return String();
}

```

```

void setup(){
    // Serial port for debugging purposes
    Serial.begin(115200);

    analogWrite(output, sliderValue.toInt());

    // Connect to Wi-Fi
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {

```



```

    delay(1000);
    Serial.println("Connecting to WiFi..");
}

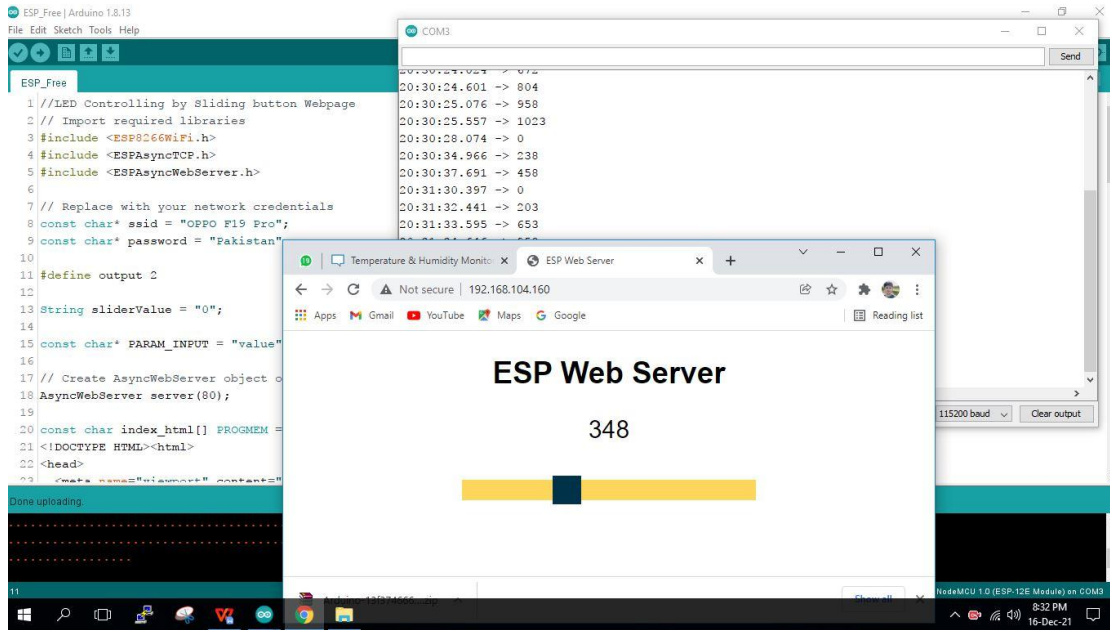
// Print ESP Local IP Address
Serial.println(WiFi.localIP());

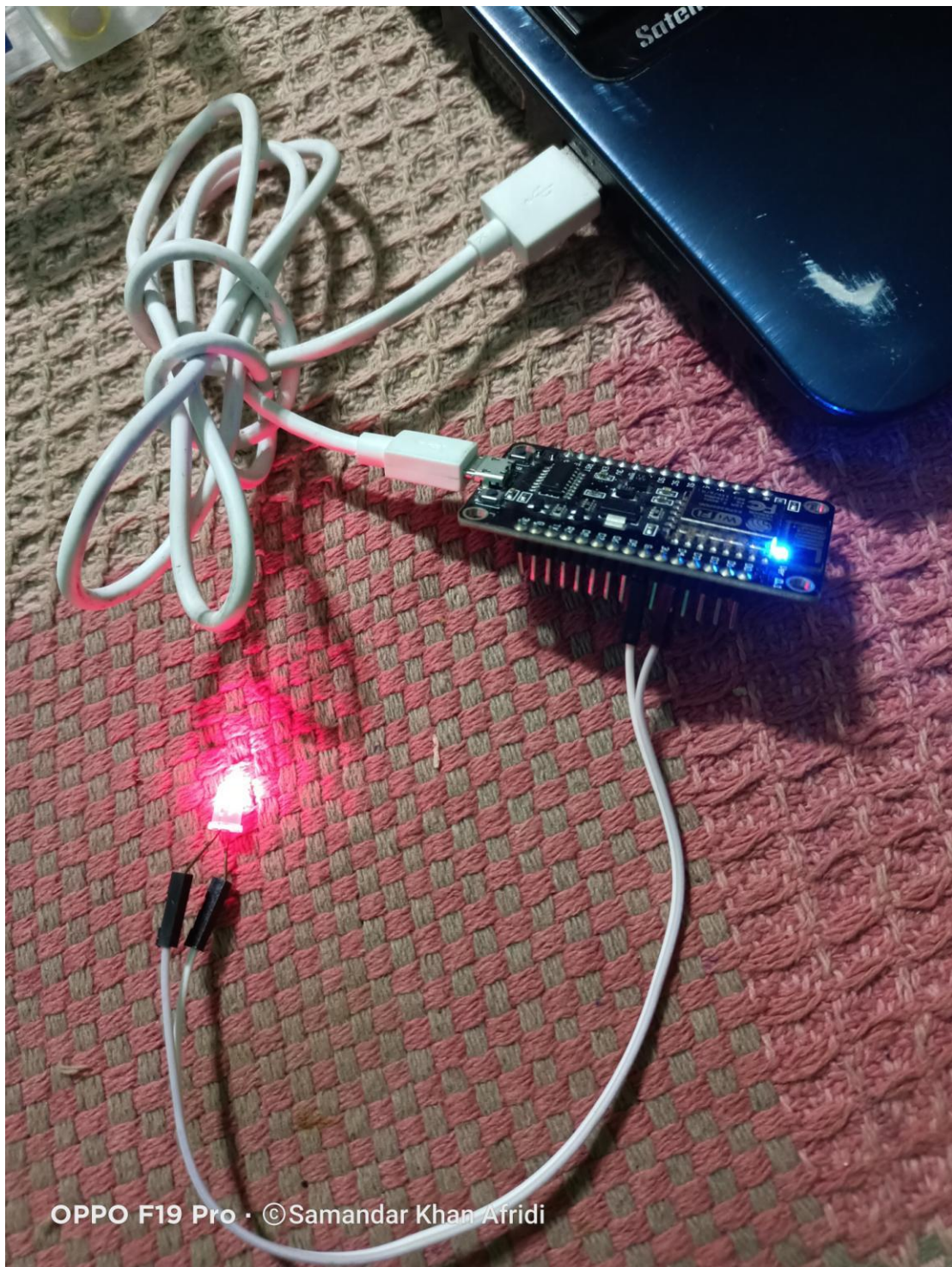
// Route for root / web page
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
});

// Send a GET request to <ESP_IP>/slider?value=<inputMessage>
server.on("/slider", HTTP_GET, [] (AsyncWebServerRequest *request) {
    String inputMessage;
    // GET input1 value on <ESP_IP>/slider?value=<inputMessage>
    if (request->hasParam(PARAM_INPUT)) {
        inputMessage = request->getParam(PARAM_INPUT)->value();
        sliderValue = inputMessage;
        analogWrite(output, sliderValue.toInt());
    }
    else {
        inputMessage = "No message sent";
    }
    Serial.println(inputMessage);
    request->send(200, "text/plain", "OK");
});
// Start server
server.begin();
}

void loop() {
}

```





No#17//LED controlling using Webserial

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <WebSerial.h>
```

```
#define LED 2
```

```

AsyncWebServer server(80);

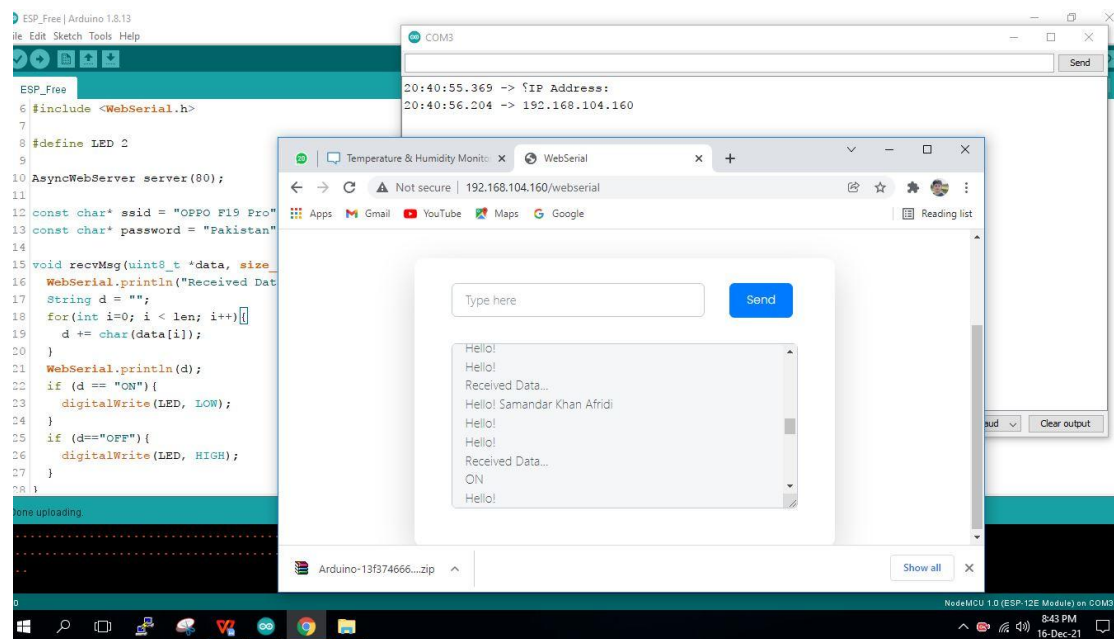
const char* ssid = "OPPO F19 Pro";           // Your WiFi SSID
const char* password = "Pakistan"; // Your WiFi Password

void recvMsg(uint8_t *data, size_t len){
    WebSerial.println("Received Data...");
    String d = "";
    for(int i=0; i < len; i++){
        d += char(data[i]);
    }
    WebSerial.println(d);
    if (d == "ON"){
        digitalWrite(LED, LOW);
    }
    if (d=="OFF"){
        digitalWrite(LED, HIGH);
    }
}

void setup() {
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    if (WiFi.waitForConnectResult() != WL_CONNECTED) {
        Serial.printf("WiFi Failed!\n");
        return;
    }
    Serial.println("IP Address: ");
    Serial.println(WiFi.localIP());
    // WebSerial is accessible at "<IP Address>/webserial" in browser
    WebSerial.begin(&server);
    WebSerial.msgCallback(recvMsg);
    server.begin();
}

void loop() {
    WebSerial.println("Hello!");
    delay(2000);
}

```

No#18//NodeMCU GET Request Example

```
#include<ESP8266WiFi.h>
#include<WiFiClient.h>
#include<ESP8266HTTPClient.h>
const char* wifiName="One";
const char* wifiPass="12345678";
const char* host="http://httpbin.org/get";
void setup()
{
    Serial.begin(115200);
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(wifiName);
    WiFi.begin(wifiName,wifiPass);
    while(WiFi.status()!=WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println("Wi-Fi Connected");
    Serial.println("IP Address");
    Serial.println(WiFi.localIP());
}
void loop()
{
    WiFiClient client;
    HTTPClient http;//declare object of class HTTPClient
    String ADCData=String(analogRead(A0));
    String getData,link;
    //Get Data
    //Note"? " added at front and "&" is used after each new parameter as per GET
    format
    getData="?data="+ADCData+"&sensor=temperature";
    String Link=host+getData;
    Serial.print("Request Link");
    Serial.println(Link);
    http.begin(client,Link);
    int httpCode=http.GET();
    String payload=http.getString();
    Serial.print("Response Code:");
    Serial.println(httpCode);//Print HTTP return code
    Serial.print("Returned data from Server:");
    Serial.println(payload);//Print request response payload
    http.end();//Close connection
    delay(5000);
}
```

```
}
```

No#19//ESP ThingSpeak Ultrasonic distance measuring in 'cm'

```
#include "ThingSpeak.h"
//#include "secrets.h" // secrets.h code is already given above in the first example

#include <ESP8266WiFi.h>
#define SECRET_SSID "OPPO F19 Pro"           // replace MySSID with your WiFi
network name
#define SECRET_PASS "Pakistan"             // replace MyPassword with your WiFi password

#define SECRET_CH_ID 1513885                // replace 0000000 with your channel
number
#define SECRET_WRITE_APIKEY "63STKM8QWFZ4N1GC" // replace XYZ with your
channel write API Key
#define trigPin D5
#define echoPin D4
long duration,cm;
char ssid[] = SECRET_SSID;  // your network SSID (name)
char pass[] = SECRET_PASS;  // your network password
int keyIndex = 0;           // your network key Index number (needed only for
WEP)
WiFiClient  client;
unsigned long myChannelNumber = SECRET_CH_ID;
const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

void setup() {
  Serial.begin(115200); // Initialize serial
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client); // Initialize ThingSpeak
}

void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
```

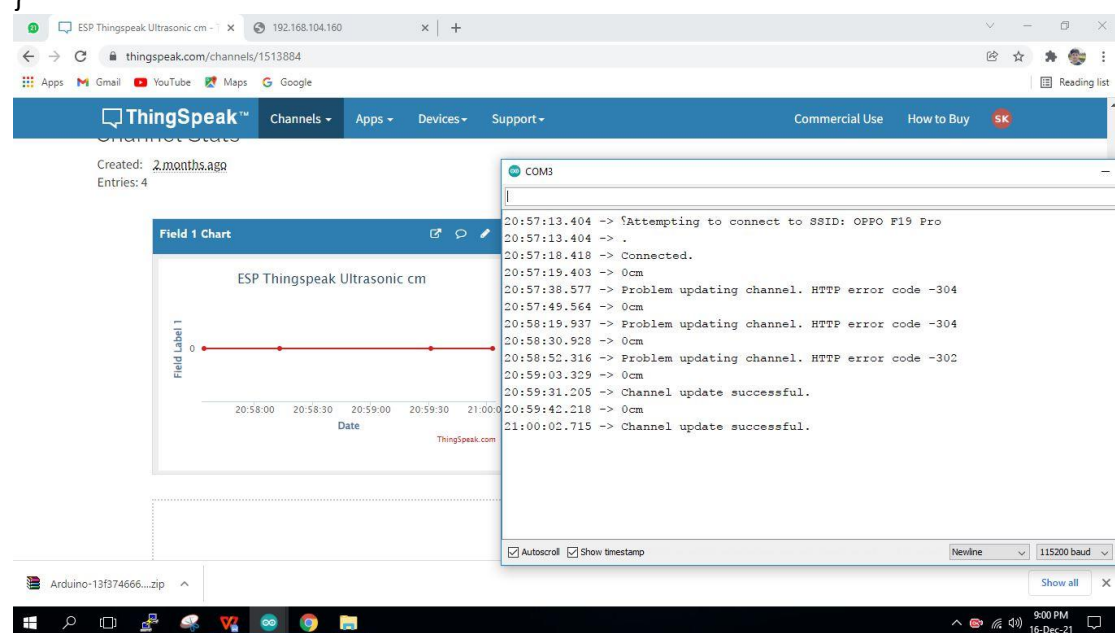


```

        WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line
if using open or WEP network
        Serial.print(".");
        delay(5000);
    }
    Serial.println("\nConnected.");
}
digitalWrite(trigPin,LOW);
delayMicroseconds(5);
digitalWrite(trigPin,HIGH);
delayMicroseconds(5);
digitalWrite(trigPin,LOW);
duration=pulseIn(echoPin,HIGH);
cm=(duration/2)*0.0343;
Serial.print(cm);
Serial.print("cm");
Serial.println(" ");
delay(5000);
    // set the fields with the values
    ThingSpeak.setField(1, cm);

// write to the ThingSpeak channel
int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
if(x == 200){
    Serial.println("Channel update successful.");
}
else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
}
delay(10000); // Wait 20 seconds to update the channel again
}

```





No#20//ESP Web_Based Serial Monitor

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <WebSerial.h>

#define LED 2

AsyncWebServer server(80);

const char* ssid = "OPPO F19 Pro";           // Your WiFi SSID
const char* password = "Pakistan";         // Your WiFi Password

void recvMsg(uint8_t *data, size_t len){
```

```

WebSerial.println("Received Data...");
String d = "";
for(int i=0; i < len; i++){
    d += char(data[i]);
}
WebSerial.println(d);
if (d == "ON"){
    digitalWrite(LED, LOW);
}
if (d=="OFF"){
    digitalWrite(LED, HIGH);
}
}

void setup() {
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    if (WiFi.waitForConnectResult() != WL_CONNECTED) {
        Serial.printf("WiFi Failed!\n");
        return;
    }
    Serial.println("IP Address: ");
    Serial.println(WiFi.localIP());
    // WebSerial is accessible at "<IP Address>/webserial" in browser
    WebSerial.begin(&server);
    WebSerial.msgCallback(recvMsg);
    server.begin();
}

void loop() {
    WebSerial.println("Hello!");
    delay(2000);
}

```