**Experiment No: 4**

**Node MCU /Arduino / Raspberry Pi wireless communication Raspberry Pi as a web server for Traffic Signal Control.**

Name of the Student: Mardav Dorle

Div. & Roll No. :A-54

**Aim**: Node MCU /Arduino / Raspberry Pi wireless communication Raspberry Pi as a web server for Traffic Signal Control.

**Components Required:**

1. Node MCU – 1
2. Micro USB Cable – 1
3. PC/Laptop – 1
4. Connecting Wires
5. Bread Board – 1
6. Red Yellow Green LED – (each 1)
7. Resistor 200 Ohm – 3

**Software Required:**

Arduino IDE

**Procedure:**

Step 1: Connect Node MCU to PC / Laptop with the help of micro USB cable.

**HOW TO CONNECT LED :**

Make the circuit diagram on the bread board according to the connection diagram shown below. Positive terminal of LED (RED) long leg of theLED is connected to the one point of the Resistor 200 Ohm and another point is connected to the D5 pin of the Node MCU, negative terminal of LED (RED) [short leg of the LED] is connected to the ground pin.

Positive terminal of LED (YELLOW) long leg of the LED is connected to the one point of the Resistor 200 Ohm and another point is connected to the D6 pin of the Node MCU, negative terminal of LED (YELLOW) [short leg of the LED] is connected to the ground pin.

Positive terminal of LED (GREEN) long leg of the LED is connected to the one point of the Resistor 200 Ohm and another point is connected to the D7 pin of the Node MCU, negative terminal of LED (GREEN) [short leg of the LED]is connected to the ground pin.

Step 2: Open new Sketch, Go to file ----> New

Step 3: Write following code in new sketch

#include <ESP8266WiFi.h>

WiFiClient client;

WiFiServer server(80);

#define LED\_Red D5

#define LED\_Yellow D6

#define LED\_Green D7

const char\* ssid = "Anjali";

const char\* password = "Anjali@12";

void setup() {

  Serial.begin(9600);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println();

  Serial.println("NodeMCU is Connected");

  Serial.println(WiFi.localIP());

  server.begin();

  pinMode(LED\_Red, OUTPUT);

  pinMode(LED\_Yellow, OUTPUT);

  pinMode(LED\_Green, OUTPUT);

}

void loop() {

  client = server.available();

  if (client) {

    String request = client.readStringUntil('\n');

    Serial.println(request);

    request.trim();

    if (request == "GET /LED\_Red HTTP/1.1") {

      digitalWrite(LED\_Red, HIGH);

      digitalWrite(LED\_Yellow, LOW);

      digitalWrite(LED\_Green, LOW);

    }

    if (request == "GET /LED\_Yellow HTTP/1.1") {

      digitalWrite(LED\_Red, LOW);

      digitalWrite(LED\_Yellow, HIGH);

      digitalWrite(LED\_Green, LOW);

    }

    if (request == "GET /LED\_Green HTTP/1.1") {

      digitalWrite(LED\_Red, LOW);

      digitalWrite(LED\_Yellow, LOW);

      digitalWrite(LED\_Green, HIGH);

    }

    if (request == "GET /LED\_Blink HTTP/1.1") {

      blinkLEDs();

    }

  }

}

void blinkLEDs() {

  int blinkCount = 5; // Number of times to blink the LEDs

  int blinkInterval = 500; // Blink interval in milliseconds

  for (int i = 0; i < blinkCount; i++) {

    digitalWrite(LED\_Red, HIGH);

    digitalWrite(LED\_Yellow, HIGH);

    digitalWrite(LED\_Green, HIGH);

    delay(blinkInterval);

    digitalWrite(LED\_Red, LOW);

    digitalWrite(LED\_Yellow, LOW);

    digitalWrite(LED\_Green, LOW);

    delay(blinkInterval);

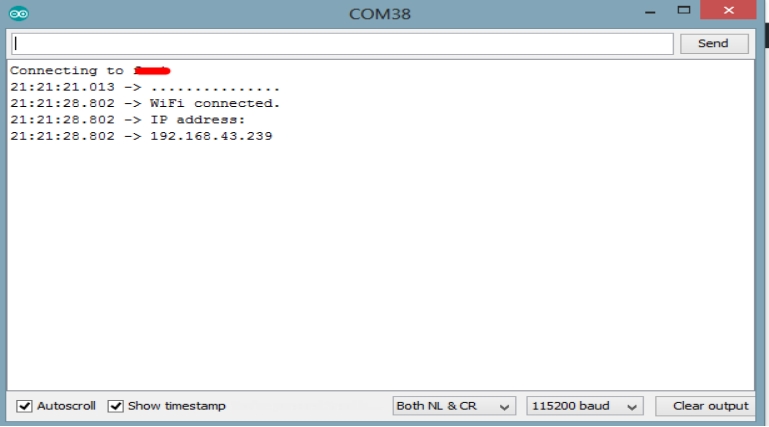
  }

}

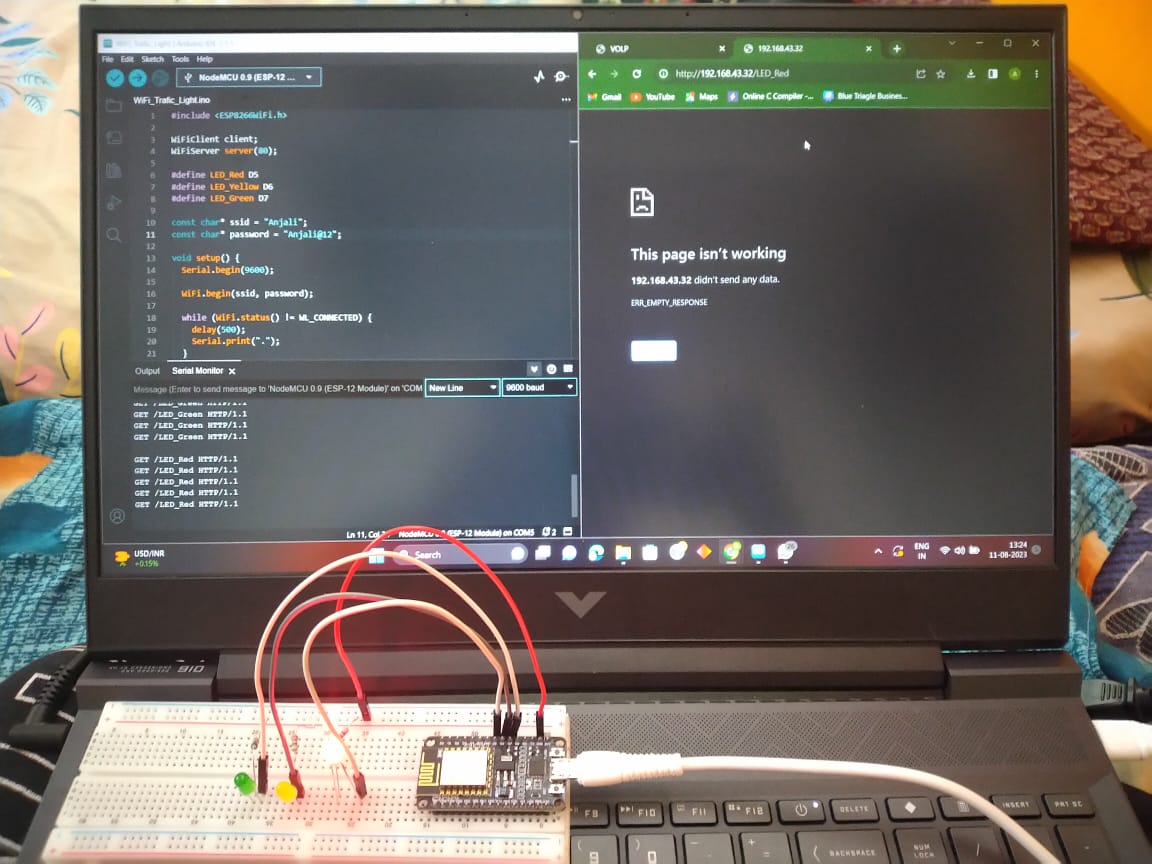
Step 5: Save the new sketch by appropriate name in a folder on your PC / Laptop

Step 6: Upload the sketch on Node MCU. Go to Sketch ----> Upload

Ensure everything is connected as described under the schematics section. After uploading the code, you should see the IP address of your web server displayed in the serial monitor as shown below.



Copy the IP address and paste it in a web browser on any device (Mobile or PC) connected to the same network as the Node MCU. You should see the web page and be able to toggle the connected appliances by clicking the buttons. Copy the IP which occur on serial monitor copy that and next to that write down \_ Red then RED led becomes glow. If you are writing Yellow than YELLOW LED becomes glow similarly for Green.



Step 7: Observe the outputs.

**Conclusion:**

The 3 LED’S of traffic signal are controlled by the web page (which has IP address displayed on serial monitor). The 3 resistors are connected serially with the respective LED.

The whole framework attached to breadboard is connected to Node MCU which is further connected to Laptop/PC through USB port.

**Extra Work:**

#include <ESP8266WiFi.h>

WiFiServer server(80); // Define port 80 as Wi-Fi server port

// Connect LEDs to ports D5, D6, and D7.

#define LED\_Red D5

#define LED\_Yellow D6

#define LED\_Green D7

void setup() {

Serial.begin(9600); // Set up baud rate.

WiFi.begin("Your\_SSID", "Your\_Password"); // Replace with your Wi-Fi credentials

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println();

Serial.println("NodeMCU is Connected");

Serial.println(WiFi.localIP());

server.begin();

pinMode(LED\_Red, OUTPUT);

pinMode(LED\_Yellow, OUTPUT);

pinMode(LED\_Green, OUTPUT);

}

void loop() {

dancingLEDs();

}

void dancingLEDs() {

// Red LED on, others off

digitalWrite(LED\_Red, HIGH);

digitalWrite(LED\_Yellow, LOW);

digitalWrite(LED\_Green, LOW);

delay(500);

// Yellow LED on, others off

digitalWrite(LED\_Red, LOW);

digitalWrite(LED\_Yellow, HIGH);

digitalWrite(LED\_Green, LOW);

delay(500);

// Green LED on, others off

digitalWrite(LED\_Red, LOW);

digitalWrite(LED\_Yellow, LOW);

digitalWrite(LED\_Green, HIGH);

delay(500);

// Turn off all LEDs

digitalWrite(LED\_Red, LOW);

digitalWrite(LED\_Yellow, LOW);

digitalWrite(LED\_Green, LOW);

delay(500);

}

**Output:**