Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

**ПРАКТИЧЕСКАЯ РАБОТА №9**

**«МДК 01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ»**

**Специальность: «09.02.07 Информационные системы и программирование*»***

Квалификация: программист

**Тема: Работа с WiFi-модулем WeMos D1 R2**

|  |  |  |
| --- | --- | --- |
| Проверил преподаватель: | Выполнил: | Выполнил: |
| Чернышев И.В. \_\_\_\_\_\_\_\_ | Студент группы П50-4-21 | Студент группы П50-1-18 |
| «\_\_\_» \_\_\_\_\_\_\_\_ 2024 год | \_\_\_\_\_\_\_\_\_\_\_\_ Игошев Р.В. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_И.И. Иванов |
| «\_\_\_» \_\_\_\_\_\_\_\_\_2024 год | | «\_\_\_» \_\_\_\_\_\_\_\_\_2020 год | |

Москва 2024

Цель работы: Собрать схему из трех светодиодов и резисторов, поработав с хостом сайта.

Начать следует с создания макета.

Выглядеть макет будет следующим образом:



Рисунок 1 – Фото макета

Снизу же видно код, в котором описана логика.

#include <ESP8266WebServer.h>

#include <ESP8266WiFi.h>

const char\* ssid = "deadpiggarduino"; // Название сети WiFi

const char\* password = "abcde1234"; // Пароль для подключения

const char\* ledOnAddress = "/led/on";

const char\* ledOffAddress = "/led/off";

ESP8266WebServer server(80); // Создаем веб сервер на 80 порту

void handleRoot() {

String s = "<h1>LED</h1>";

s += "<a href=";

s += ledOnAddress;

s += ">LED On</a>";

s += "<a href=";

s += ledOffAddress;

s += ">LED Off</a>";

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

}

void ledOn() {

digitalWrite(BUILTIN\_LED, HIGH);

server.sendHeader("Location", String("/"), true);

server.send(302, "text/plain", "");

}

void ledOff() {

digitalWrite(BUILTIN\_LED, LOW);

server.sendHeader("Location", String("/"), true);

server.send(302, "text/plain", "");

}

// Функция настройки. Выполняется один раз при запуске

void setup() {

delay(1000); // Ждём секунду

WiFi.softAP(ssid, password); // Создаём точку WiFi

// Указываем по каким роутам какие методы запускать

server.on("/", handleRoot);

server.on("/led/on", ledOff);

server.on("/led/off", ledOn);

server.begin();

// Диод по умолчанию выключен

pinMode(BUILTIN\_LED, OUTPUT);

digitalWrite(BUILTIN\_LED, LOW);

}

// Основной цикл программы

void loop() {

// Ждём подключения

server.handleClient();

}

Второе задание

Код второго задания:

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <ESPAsyncTCP.h>

#include <ESPAsyncWebServer.h>

#include "LittleFS.h"

#include <Arduino\_JSON.h>

// Replace with your network credentials

const char\* ssid = "Bistro vikluchi";

const char\* password = "22092022";

// Create AsyncWebServer object on port 80

AsyncWebServer server(80);

// Create a WebSocket object

AsyncWebSocket ws("/ws");

// Set LED GPIO

const int ledPin1 = 14;

const int ledPin2 = 12;

const int ledPin3 = 13;

String message = "";

String sliderValue1 = "0";

String sliderValue2 = "0";

String sliderValue3 = "0";

int dutyCycle1;

int dutyCycle2;

int dutyCycle3;

//Json Variable to Hold Slider Values

JSONVar sliderValues;

//Get Slider Values

String getSliderValues(){

sliderValues["sliderValue1"] = String(sliderValue1);

sliderValues["sliderValue2"] = String(sliderValue2);

sliderValues["sliderValue3"] = String(sliderValue3);

String jsonString = JSON.stringify(sliderValues);

return jsonString;

}

// Initialize LittleFS

void initFS() {

if (!LittleFS.begin()) {

Serial.println("An error has occurred while mounting LittleFS");

}

else{

Serial.println("LittleFS mounted successfully");

}

}

// Initialize WiFi

void initWiFi() {

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

Serial.print("Подключение к сети Wi-Fi");

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

Serial.print('.');

delay(1000);

}

Serial.println(WiFi.localIP());

}

void notifyClients(String sliderValues) {

ws.textAll(sliderValues);

}

void handleWebSocketMessage(void \*arg, uint8\_t \*data, size\_t len) {

AwsFrameInfo \*info = (AwsFrameInfo\*)arg;

if (info->final && info->index == 0 && info->len == len && info->opcode == WS\_TEXT) {

data[len] = 0;

message = (char\*)data;

if (message.indexOf("1s") >= 0) {

sliderValue1 = message.substring(2);

dutyCycle1 = map(sliderValue1.toInt(), 0, 100, 0, 1023);

Serial.println(dutyCycle1);

Serial.print(getSliderValues());

notifyClients(getSliderValues());

}

if (message.indexOf("2s") >= 0) {

sliderValue2 = message.substring(2);

dutyCycle2 = map(sliderValue2.toInt(), 0, 100, 0, 1023);

Serial.println(dutyCycle2);

Serial.print(getSliderValues());

notifyClients(getSliderValues());

}

if (message.indexOf("3s") >= 0) {

sliderValue3 = message.substring(2);

dutyCycle3 = map(sliderValue3.toInt(), 0, 100, 0, 1023);

Serial.println(dutyCycle3);

Serial.print(getSliderValues());

notifyClients(getSliderValues());

}

if (strcmp((char\*)data, "getValues") == 0) {

notifyClients(getSliderValues());

}

}

}

void onEvent(AsyncWebSocket \*server, AsyncWebSocketClient \*client, AwsEventType type, void \*arg, uint8\_t \*data, size\_t len) {

switch (type) {

case WS\_EVT\_CONNECT:

Serial.printf("WebSocket client #%u connected from %s\n", client->id(), client->remoteIP().toString().c\_str());

break;

case WS\_EVT\_DISCONNECT:

Serial.printf("WebSocket client #%u disconnected\n", client->id());

break;

case WS\_EVT\_DATA:

handleWebSocketMessage(arg, data, len);

break;

case WS\_EVT\_PONG:

case WS\_EVT\_ERROR:

break;

}

}

void initWebSocket() {

ws.onEvent(onEvent);

server.addHandler(&ws);

}

void setup() {

Serial.begin(115200);

pinMode(ledPin1, OUTPUT);

pinMode(ledPin2, OUTPUT);

pinMode(ledPin3, OUTPUT);

initFS();

initWiFi();

initWebSocket();

// Web Server Root URL

server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){

request->send(LittleFS, "/index.html", "text/html");

});

server.serveStatic("/", LittleFS, "/");

// Start server

server.begin();

}

void loop() {

analogWrite(ledPin1, dutyCycle1);

analogWrite(ledPin2, dutyCycle2);

analogWrite(ledPin3, dutyCycle3);

ws.cleanupClients();

}

Макет второго задания:

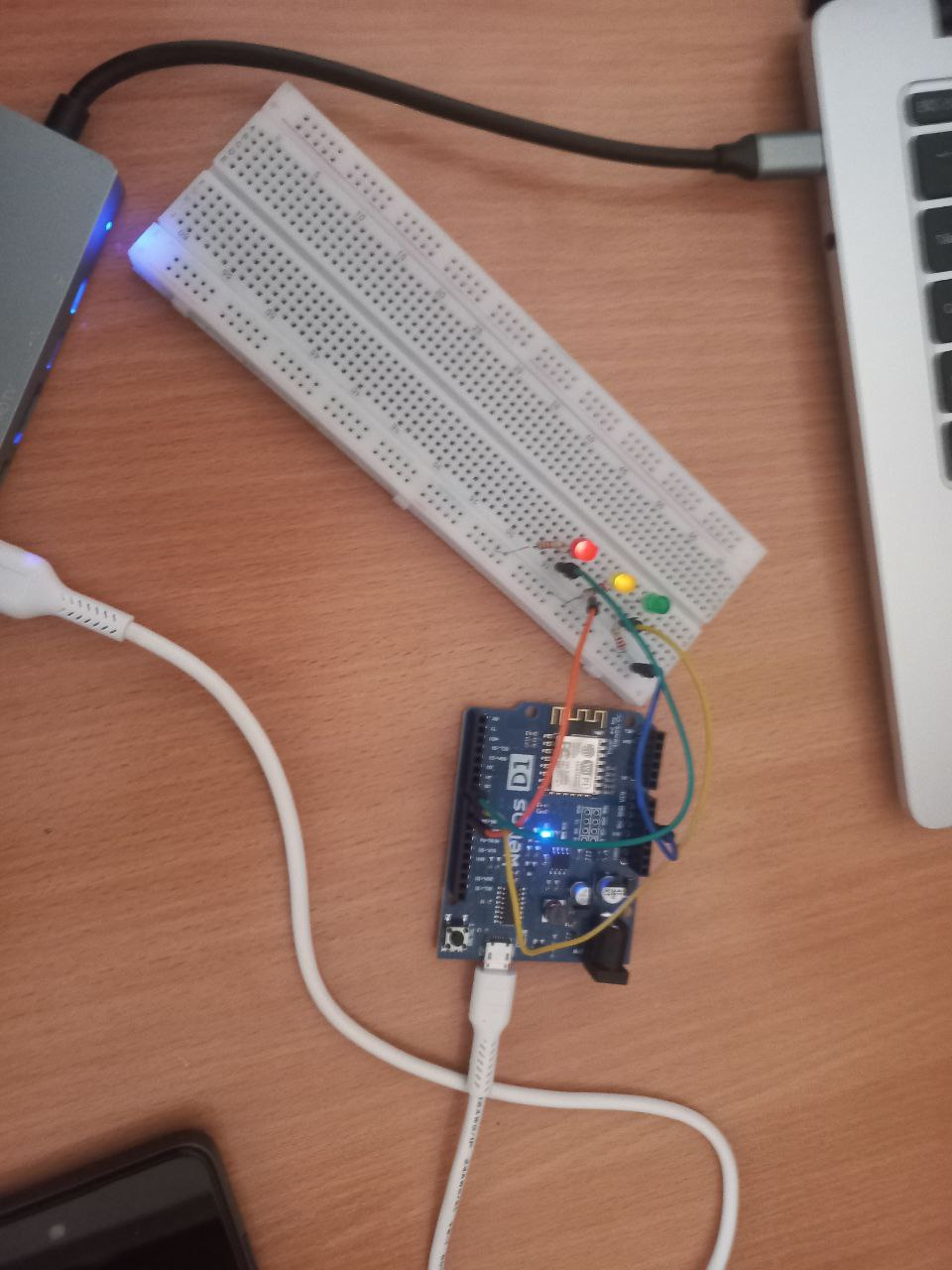


Рисунок 2 – Макет второго задания

Вывод: Собрали схему из трех светодиодов и резисторов, поработав с хостом сайта.