



C4.2 Programación Microcontrolador NodeMCU ESP32

Comunicación por medio de la conexión Wi-Fi



Instrucciones

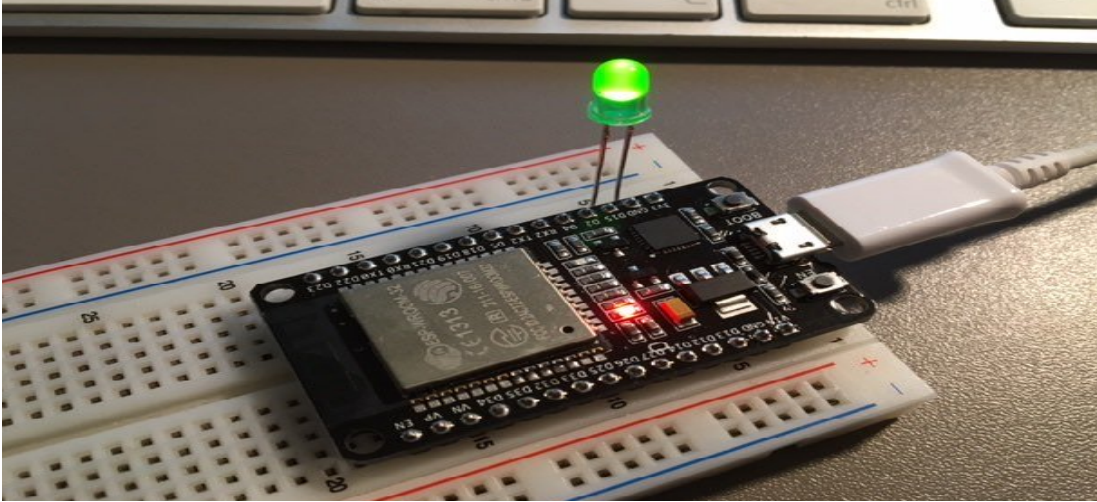
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces.
- Es requisito que el archivo .md contenga una etiqueta del enlace al repositorio de su documento en Github, por ejemplo **Enlace a mi GitHub**
- Al concluir el reto el reto se deberá subir a github el archivo .md creado.
- Desde el archivo .md se debe exportar un archivo .pdf con la nomenclatura **C4.2_NombreAlumno_Equipo.pdf**, el cual deberá subirse a classroom dentro de su apartado correspondiente, para que sirva como evidencia de su entrega; siendo esta plataforma **oficial** aquí se recibirá la calificación de su actividad por individual.
- Considerando que el archivo .pdf, fue obtenido desde archivo .md, ambos deben ser idénticos y mostrar el mismo contenido.
- Su repositorio además de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o índice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
| readme.md
| | blog
| | | C4.1_TituloActividad.md
| | | C4.2_TituloActividad.md
| | | C4.3_TituloActividad.md
| | | C4.4_TituloActividad.md
| | | C4.5_TituloActividad.md
| | img
| | docs
| | | A4.1_TituloActividad.md
| | | A4.2_TituloActividad.md
```



Desarrollo

1. Basado en el siguiente circuito, ensamblarlo, utilizando los elementos electrónicos observados.



Fuente de consulta: [Random Nerd Tutorials](#)

2. Analice y apóyese del programa que se muestra a continuación para elaborar el reto.

```
/*
  WiFi Web Server Simple
  */

#include <WiFi.h>
#include <WebServer.h>

const char* ssid = "<identificador>";
const char* password = "<password>";

WebServer server(80); // Object of WebServer(HTTP port, 80 is default)

void setup() {
  Serial.begin(115200);
  Serial.println("Try Connecting to ");
  Serial.println(ssid);

  // Connect to your wi-fi modem
  WiFi.begin(ssid, password);

  // Check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
}
```

```
Serial.println("WiFi connected successfully");
Serial.print("Got IP: ");
Serial.println(WiFi.localIP()); //Show ESP32 IP on serial

server.on("/", handle_root);

server.begin();
Serial.println("HTTP server started");
delay(100);
}

void loop() {
  server.handleClient();
}

// HTML & CSS contents which display on web server
String HTML = "<!DOCTYPE html>\n
<html>\n
<body>\n
<h1>Mi Primer Servidor Web with ESP32 - Station Mode &#128522;</h1>\n
</body>\n
</html>";

// Handle root url (/)
void handle_root() {
  server.send(200, "text/html", HTML);
}
```

3. Pruebe y observe los resultados obtenidos explicándolos en esta sección.
4. Al programa anterior agregue las instrucciones necesarias para que se despliegue en la interface un botón que permita encender y apagar un Led tal como se muestra en la figura 1.
5. Inserte aquí las imágenes que considere como evidencias para demostrar el resultado obtenido.

electro_moderno ☆

Meet para reunion: meet.google.com/kcj-byuo-mcn



Luis Alejandro Sanchez Gallegos 12:29

Hoy ▾

===== C4.2: =====

Vídeo:

https://mega.nz/file/BvJHIYSA#ydlldgKQ4E8bTx-Punja0Rvr3BiIR0gTO3_kKqvNkaRI

Código:

```
#include <WiFi.h>
#include <WebServer.h>

const char* ssid = "*****";
const char* password = "*****";

WebServer server(80); // Object of WebServer(HTTP port, 80 is default)

const int LED_PIN = 2;
bool led_on = false;

void setup() {
  pinMode(LED_PIN, OUTPUT);

  Serial.begin(115200);
  Serial.println("Try Connecting to ");
  Serial.println(ssid);
```

Enviar mensaje a **electro_moderno**

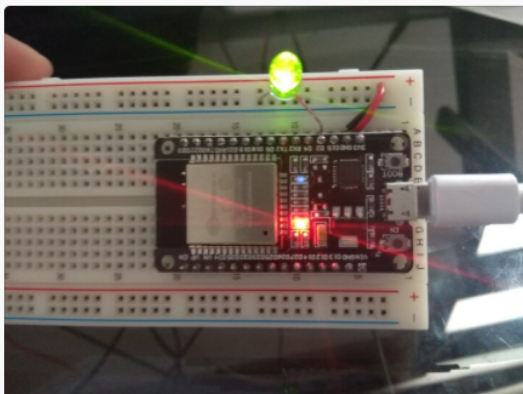
electro_moderno ☆

Meet para reunion: meet.google.com/kcj-byuo-mcn

Fotografía del circuito:

C4.2_foto.jpg ▾

Hoy ▾



tengo entendido que esta actividad se va a hacer en equipo, como si tuviera la letra "A" en lugar de "C"

(voy a tener que alejarme de mi computadora un rato y no tengo el Slack ahí, si tienen dudas por favor hablenme por messenger)



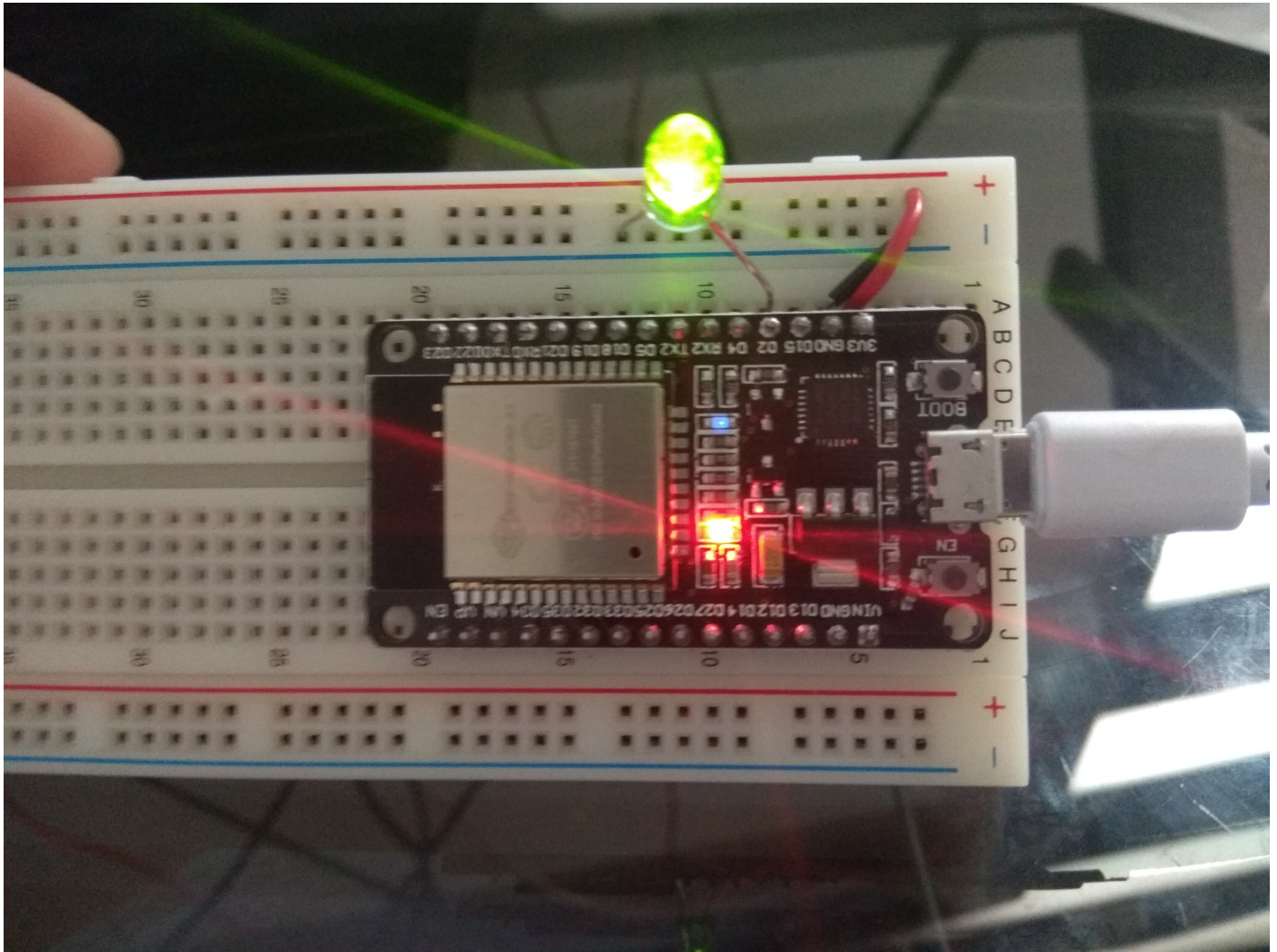
CAROLINA DOMINGUEZ CRUZ 12:34

Okis

Enviar mensaje a **electro_moderno**

6. Evidencias de ALEJANDRO en ESP32

Foto de mi compañero Alejandro



Video:

Explicacion realizada por mi compañero Alejandro. [Click](#)

Codigo

```
#include <WiFi.h>
#include <WebServer.h>
const char* ssid = "*****";
const char* password = "*****";
WebServer server(80); // Object of WebServer(HTTP port, 80 is default)
const int LED_PIN = 2;
bool led_on = false;
void setup() {
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(115200);
  Serial.println("Try Connecting to ");
```



```

Serial.println(ssid);
// Connect to your wi-fi modem
WiFi.begin(ssid, password);
// Check wi-fi is connected to wi-fi network
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected successfully");
Serial.print("Got IP: ");
Serial.println(WiFi.localIP()); //Show ESP32 IP on serial
server.on("/", handle_root);
server.on("/toggle", handle_toggle_led);
server.begin();
Serial.println("HTTP server started");
delay(100);
}
void loop() {
    server.handleClient();
}
// HTML & CSS contents which display on web server
String get_dom_string() {
    String action_name;
    if (led_on) action_name = "APAGAR";
    else action_name = "ENCENDER";
    String dom_string = "<!DOCTYPE html>";
    dom_string += "<html>";
    dom_string += "<body>";
    dom_string += "<h1>Mi Primer Servidor Web with ESP32 - Station Mode &#128522;";
    dom_string += "</h1>";
    dom_string += "<form action=\\\"/toggle\\\" target=\\\"_self\\\" method=\\\"post\\\">";
    dom_string += "    <input type=\\\"submit\\\" value=\\\"";
    dom_string += action_name;
    dom_string += " LED\\\">";
    dom_string += "</form>";
    dom_string += "</body>";
    dom_string += "</html>";
    return dom_string;
}
// Handle root url (/)
void handle_root() {
    server.send(200, "text/html", get_dom_string());
}
// Manejo de control para encender y apagar el LED
void handle_toggle_led() {
    led_on = !led_on;
    Serial.print("LED ");
    if (led_on) {
        Serial.println("ON");
    }
}

```

```
    digitalWrite(LED_PIN, HIGH);  
  }  
  else {  
    Serial.println("OFF");  
    digitalWrite(LED_PIN, LOW);  
  }  
  server.send(200, "text/html", get_dom_string());  
}
```

Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	20
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	80

 [Ir a comunicación](#)