



REPORTE DE PRÁCTICA

No. De la práctica:	3
Fecha:	12/12/2023
Lugar:	LABORATORIO
Páginas:	Página 1 de 14

DATOS GENERALES

NOMBRE DE LA PRÁCTICA:	COMUNICACIÓN WIFI ESP32 - MICROCONTRADOR
ASIGNATURA:	SISTEMAS PROGRAMABLES
UNIDAD TEMÁTICA:	U3 MICROCONTRALADORES
NOMBRE DEL DOCENTE:	ISC. SILVERIO RAMIREZ
INTEGRANTES DEL EQUIPO:	1- GONZALO MARTINEZ SILVERIO 2- MARIA CRUZ GOMEZ 3- SANDRA LIZBETH MARTINEZ MTZ. 4- RUTH YZELDA YAMILETH VAZQUEZ RAMOS
GRUPO:	7S1A
CARRERA:	INGENIERIA EN SISTEMAS COMPUTACIONALES

INTRODUCCIÓN:

Desde la invención del circuito integrado, el desarrollo constante de la electrónica digital ha dado lugar a dispositivos cada vez más complejos. Entre ellos los microprocesadores y los microcontroladores.

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de nuestro hogar.

En las aplicaciones sencillas resultan preferibles las soluciones no programables que no requieren desarrollo de software. Escribir software consume mucho tiempo



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

por lo que resulta más costoso y en aplicaciones sencillas y/o de poca tirada a menudo es más razonable efectuar tareas en hardware. Sin embargo, conforme aumenta la complejidad del sistema, aumentan las ventajas del uso de sistemas programables.

Una de las principales ventajas de los sistemas programables es su flexibilidad, lo que permite actualizar el funcionamiento de un sistema tan sólo mediante el cambio del programa sin tener que volver a diseñar el hardware. Esta flexibilidad es muy importante, al permitir que los productos se actualicen con facilidad y economía.

OBJETIVO:

Lograr la automatización y gestión de la iluminación mediante un sistema electrónico. Este proceso implica utilizar un microcontrolador para enviar señales de control al LED, permitiendo así que el LED se encienda o apague según las necesidades del usuario o del sistema en el que se integra.

COMPETENCIA A DESARROLLAR:

Específica(s):

- ✚ Identifica las características eléctricas de un microcontrolador.
- ✚ Conoce la arquitectura interna del microcontrolador.
- ✚ Comprende la estructura de registros del microcontrolador.
- ✚ Analiza dispositivos de entrada/salida y puertos del microcontrolador.

Genéricas:

- ✚ Capacidad de aplicar los conocimientos en la práctica
- ✚ Capacidad de abstracción, análisis y síntesis
- ✚ Capacidad para tomar decisiones

CRITERIO DE DESEMPEÑO:

Habilidad en el uso de las tecnologías, capacidad de abstracción, análisis, crítica. Capacidad de trabajar en equipo Usa competencias de asignaturas previas para presentar una práctica para programar al microcontrolador con sus dispositivos de entrada/salida que sean adaptables.

ATRIBUTO DE EGRESO:

Si estás buscando un atributo de ingreso para controlar un microcontrolador y, por ende, encender o apagar un LED, se puede considerar utilizar un interruptor (switch)



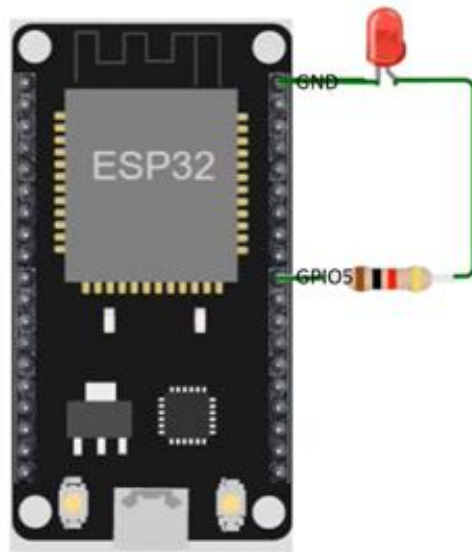
INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

como entrada. El interruptor puede ser físico (como un botón) o virtual (a través de una página web, con la programación en HTML).

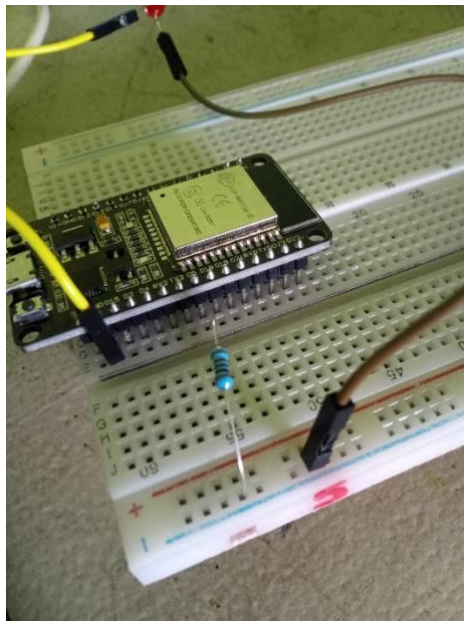
- ✚ Comunicación inalámbrica: esta se realizó mediante Wi-Fi, para enviar comandos al microcontrolador desde un dispositivo remoto. Esto nos permite tener un control más flexible y remoto del LED.
- ✚ Utilizamos la programación HTML para poder hacer la pagina web, donde indicaremos la opción de apagar o encender la led.
- ✚ De igual manera colocamos el código para el Arduino, ya que esta se enviarán los datos al microcontrolador.

MATERIAL Y EQUIPO (REQUERIMIENTOS):

- ESP32
- Libreta de apuntes
- Computadora
- IDE Arduino
- Led
- Jumpers
- Protoboard
- Cable USB
- Resistencia de 1KΩ

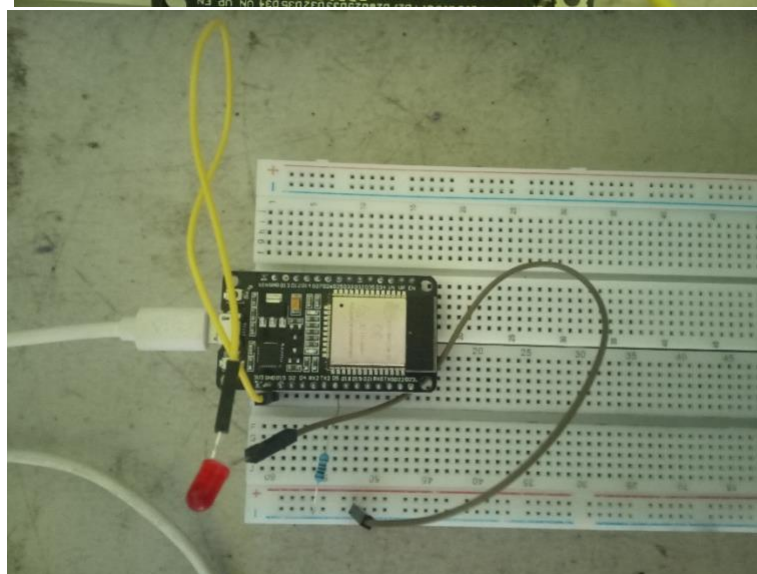
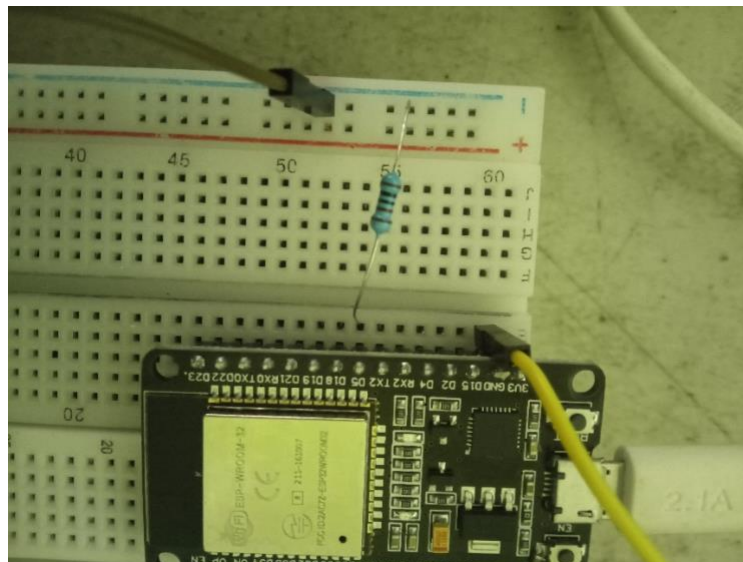
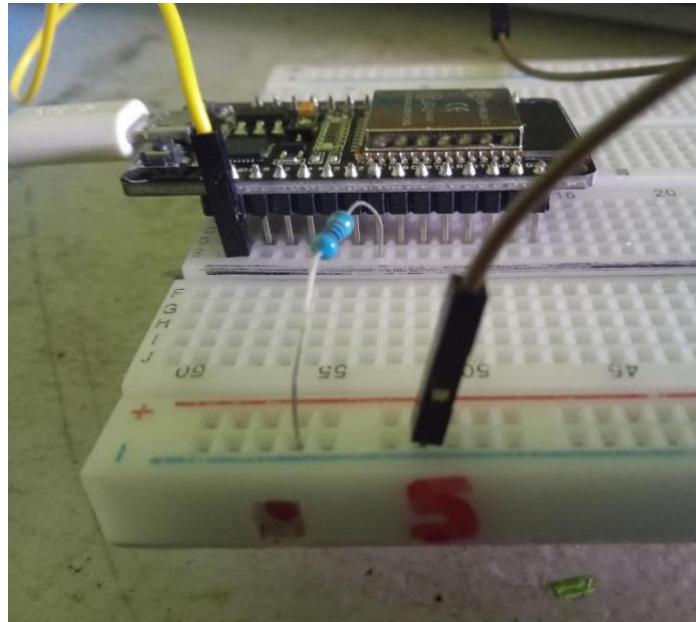


ESP32 Dev Module





INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE



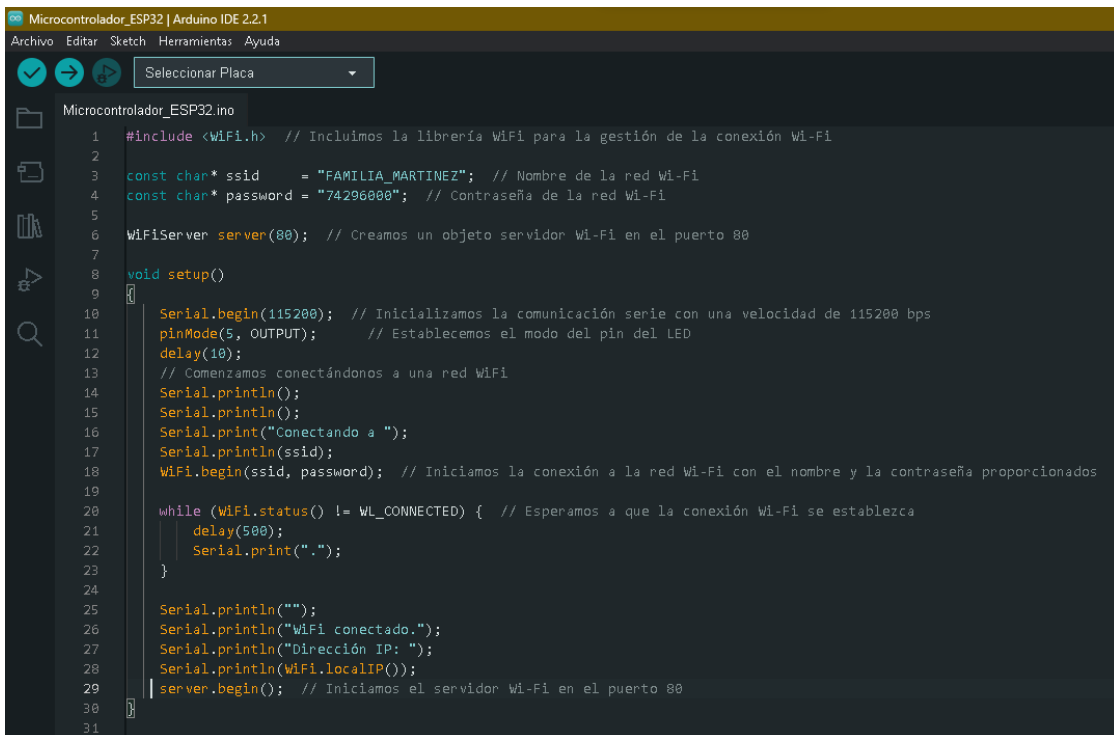
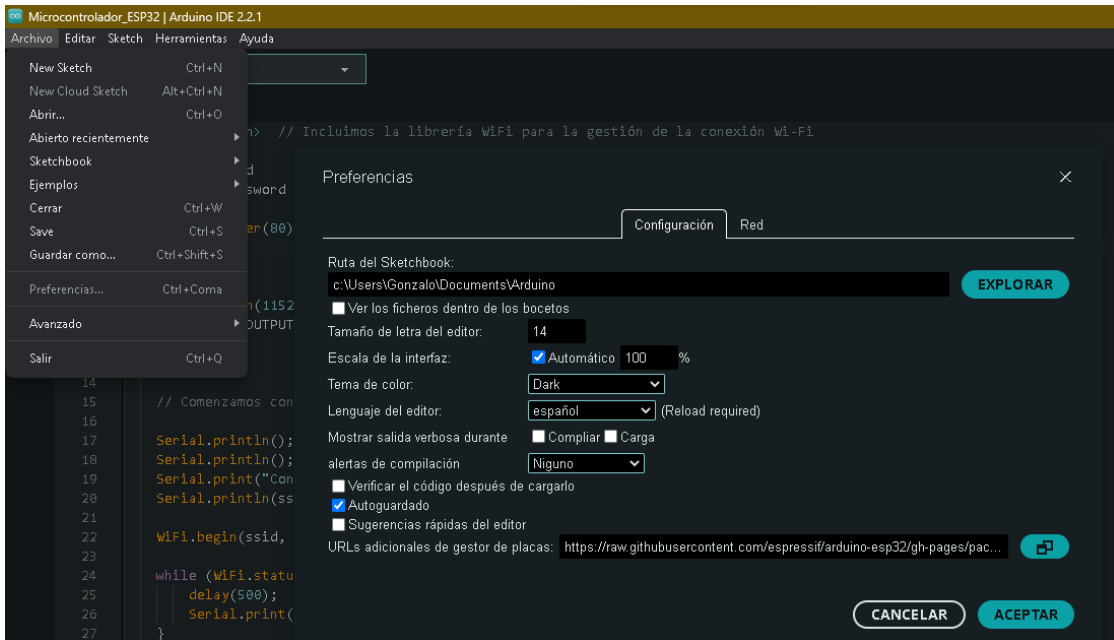


INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Para instalar la placa ESP32 en su IDE de Arduino, siga estas siguientes instrucciones:

En tu IDE de Arduino, ve a Archivo> Preferencias

En el campo URL adicionales de gestor de placas Ingrese: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json y hacemos clic en el botón "Aceptar":





INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

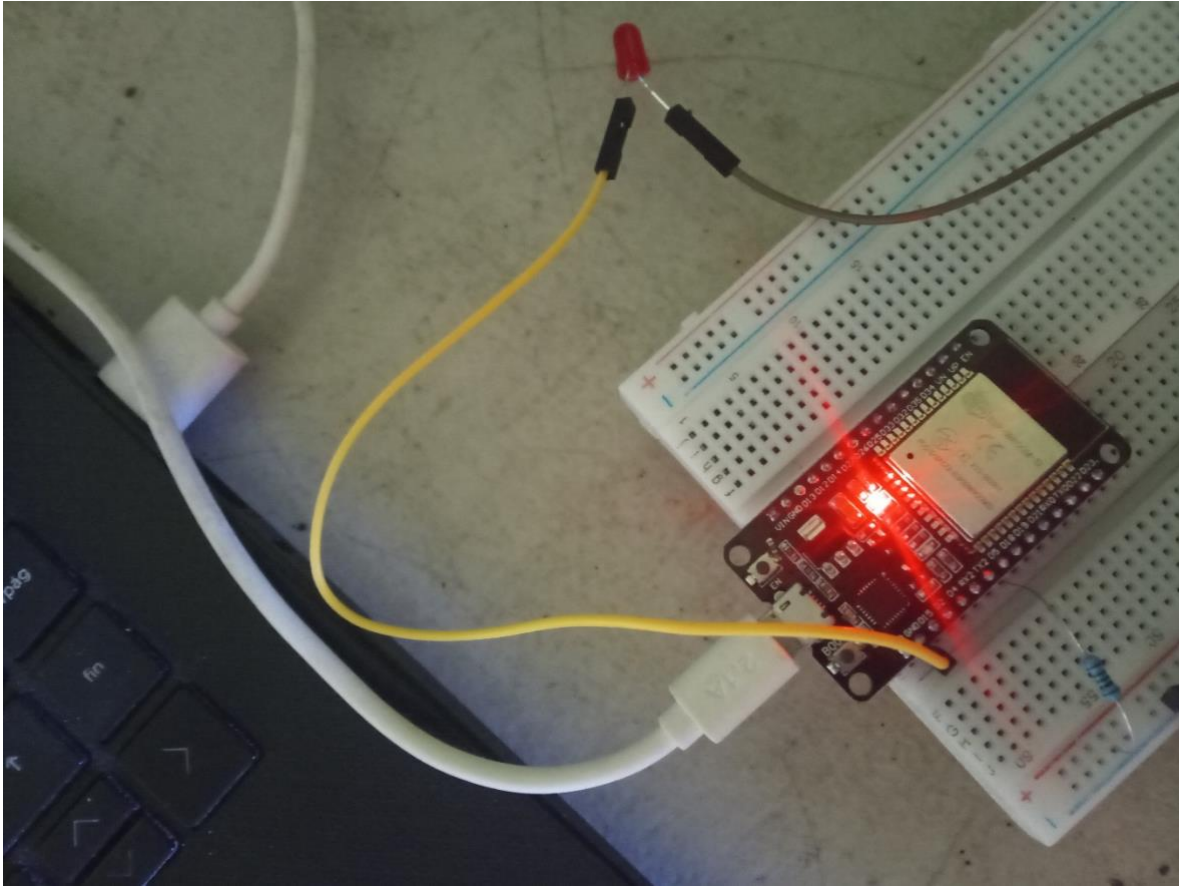
```
32 void loop() {
33   WiFiClient client = server.available(); // esperar clientes entrantes
34
35   if (client) { // si se recibe un cliente,
36     Serial.println("Nuevo Cliente."); // imprimir un mensaje por el puerto serie
37     String currentLine = ""; // hacer una cadena para almacenar los datos entrantes del cliente
38     while (client.connected()) { // bucle mientras el cliente esté conectado
39       if (client.available()) { // si hay bytes para leer del cliente,
40         char c = client.read(); // leer un byte, luego
41         Serial.write(c); // imprimirlo por el monitor serie
42         if (c == '\n') { // si el byte es un carácter de nueva línea
43           // si la línea actual está en blanco, se tienen dos caracteres de nueva línea seguidos.
44           // ese es el final de la solicitud HTTP del cliente, así que enviar una respuesta:
45           if (currentLine.length() == 0) {
46             // Las cabeceras HTTP siempre comienzan con un código de respuesta (por ejemplo, HTTP/1.1 200 OK)
47             // y un tipo de contenido para que el cliente sepa lo que viene, luego una línea en blanco:
48             client.println("HTTP/1.1 200 OK");
49             client.println("Content-type:text/html");
50             client.println();
51
52             // el contenido de la respuesta HTTP sigue a la cabecera:
53             client.print("Haz clic <a href=\"/H\">aquí</a> para encender el LED en el pin 5.<br>");
54             client.print("Haz clic <a href=\"/L\">aquí</a> para apagar el LED en el pin 5.<br>");
55
56             // La respuesta HTTP termina con otra línea en blanco:
57             client.println();
58             // salir del bucle while:
59             break;
60           } else { // si se recibe una nueva línea, entonces borrar currentLine:
61             currentLine = "";
62           }
63         } else if (c != '\r') { // si se recibe cualquier cosa que no sea un carácter de retorno de carro,
64           currentLine += c; // agregarlo al final de currentLine
65         }
66
67         // el contenido de la respuesta HTTP sigue a la cabecera:
68         client.print("Haz clic <a href=\"/H\">aquí</a> para encender el LED en el pin 5.<br>");
69         client.print("Haz clic <a href=\"/L\">aquí</a> para apagar el LED en el pin 5.<br>");
70
71         // La respuesta HTTP termina con otra línea en blanco:
72         client.println();
73         // salir del bucle while:
74         break;
75       } else { // si se recibe una nueva línea, entonces borrar currentLine:
76         currentLine = "";
77       }
78     } else if (c != '\r') { // si se recibe cualquier cosa que no sea un carácter de retorno de carro,
79       currentLine += c; // agregarlo al final de currentLine
80     }
81
82     // Comprobar si la solicitud del cliente fue "GET /H" o "GET /L":
83     if (currentLine.endsWith("GET /H")) {
84       digitalWrite(5, HIGH); // GET /H enciende el LED
85     }
86     if (currentLine.endsWith("GET /L")) {
87       digitalWrite(5, LOW); // GET /L apaga el LED
88     }
89   }
90
91   // cerrar la conexión:
92   client.stop();
93   Serial.println("Cliente Desconectado.");
94 }
```



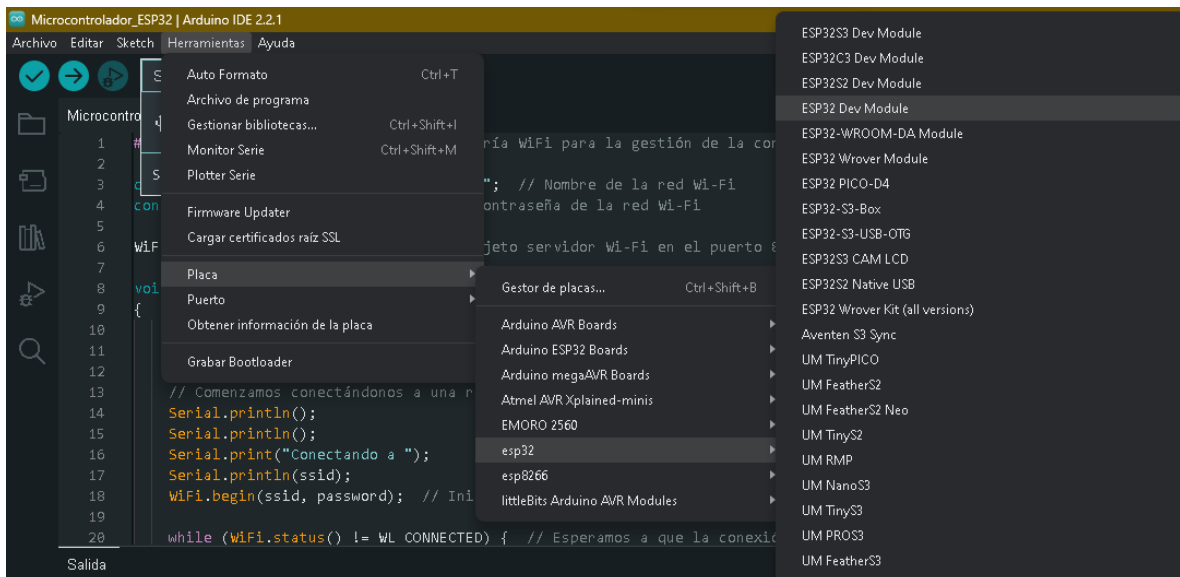

INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Ahora subiremos el código al ESP32 usando Arduino IDE siguiendo los pasos a continuación:

Conectamos el ESP32 a la PC mediante un cable USB.



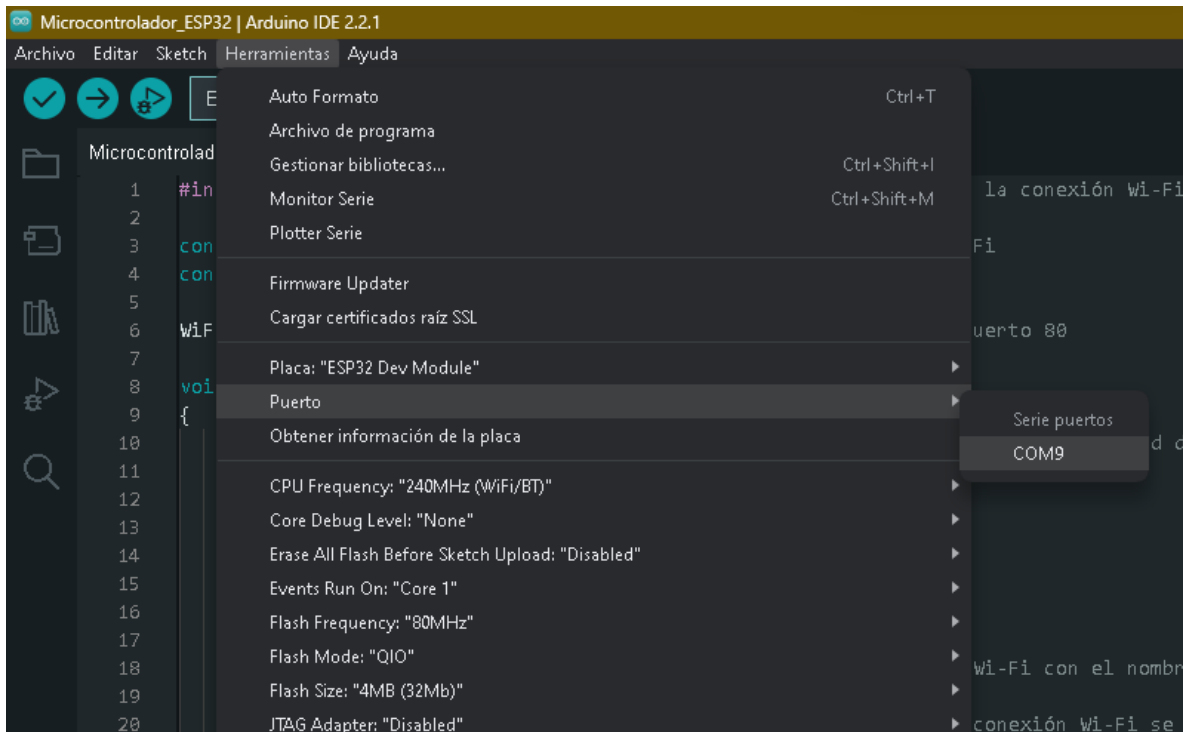
Nos dirigimos a Herramientas--> Placa--> módulo de desarrollo ESP32.



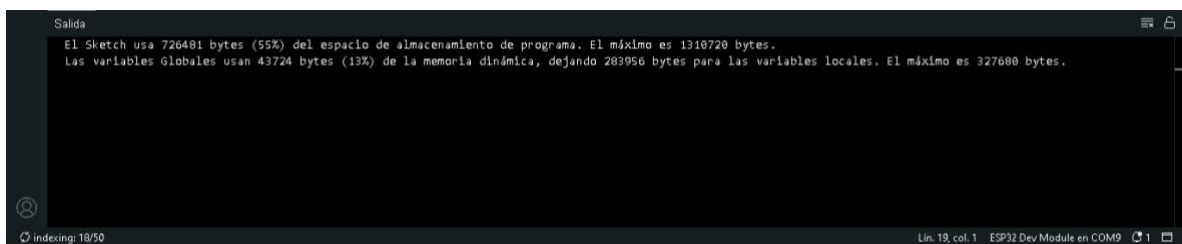
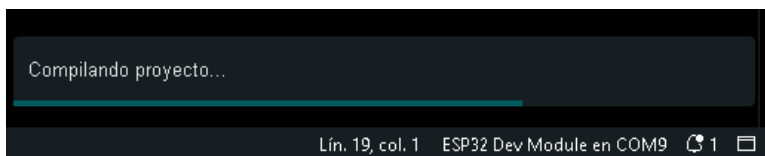


INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Ahora vamos a Herramientas-->Puerto y seleccionamos el puerto al que está conectado el ESP32.



Ahora hacemos clic en compilar y cargar el código.





INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

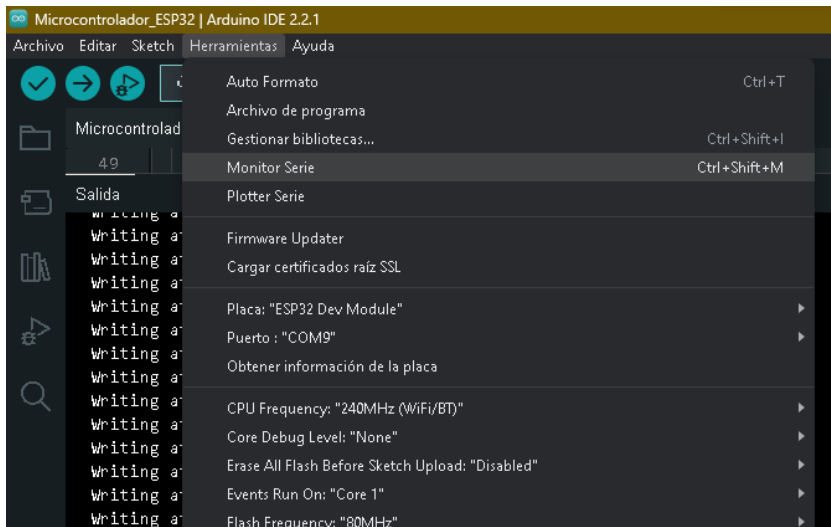
Después de completar la carga, encontrará un mensaje como este en su consola de salida.

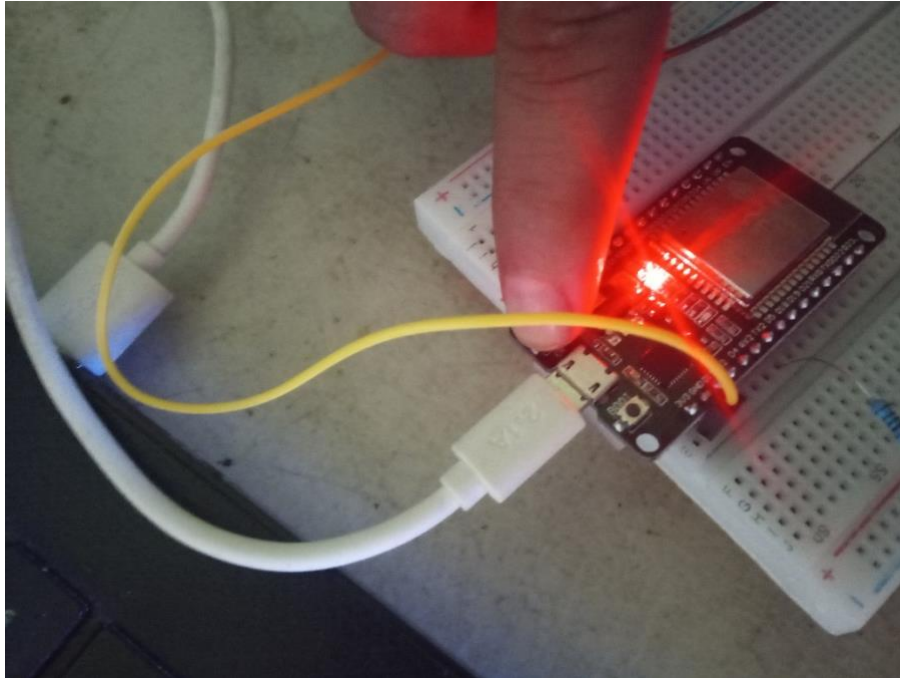
```
Salida
Writing at 0x00000000... (20 %)
Writing at 0x00000001... (23 %)
Writing at 0x00000002... (26 %)
Writing at 0x00000003... (30 %)
Writing at 0x00000004... (33 %)
Writing at 0x00000005... (36 %)
Writing at 0x00000006... (40 %)
Writing at 0x00000007... (43 %)
Writing at 0x00000008... (46 %)
Writing at 0x00000009... (50 %)
Writing at 0x0000000a... (53 %)
Writing at 0x0000000b... (56 %)
Writing at 0x0000000c... (60 %)
Writing at 0x0000000d... (63 %)
Writing at 0x0000000e... (66 %)
Writing at 0x0000000f... (70 %)
Writing at 0x00000010... (73 %)
Writing at 0x00000011... (76 %)
Writing at 0x00000012... (80 %)
Writing at 0x00000013... (83 %)
Writing at 0x00000014... (86 %)
Writing at 0x00000015... (90 %)
Writing at 0x00000016... (93 %)
Writing at 0x00000017... (96 %)
Writing at 0x00000018... (100 %)
Wrote 732784 bytes (477977 compressed) at 0x00010000 in 6.4 seconds (effective 912.5 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

indexing: 0/1
```

Ahora abrimos el monitor serie y presionamos el botón de reinicio del ESP32 para que comience a conectarse a la red.





Una vez conectado mostrara la IP del ESP.

```
Microcontrolador_ESP32 | Arduino IDE 2.2.1
Archivo  Editar  Sketch  Herramientas  Ayuda
[Checkmark] [Next] [Upload] [ESP32 Dev Module]
Microcontrolador_ESP32.ino
49      client.println("Content-type:text/html");
Salida  Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'ESP32 Dev Module' a 'COM9')
ets Jul 29 2019 12:21:46

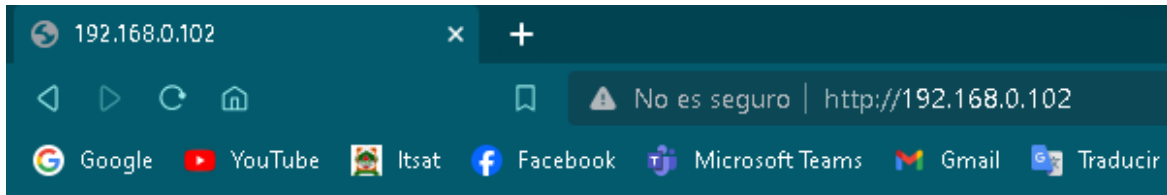
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0

Conectando a FAMILIA_MARTINEZ
..
WiFi conectado.
Dirección IP:
192.168.0.102
```



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Copiamos y pegamos la IP en nuestro navegador, nos dirigirá a la página web donde controlaremos el LED.



Haz clic [aquí](#) para encender el LED en el pin 5.

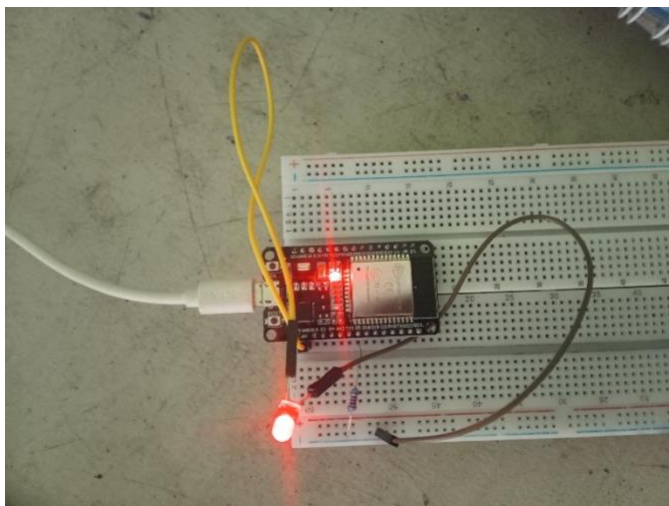
Haz clic [aquí](#) para apagar el LED en el pin 5.

Encendemos el LED.



Haz clic [aquí](#) para encender el LED en el pin 5.

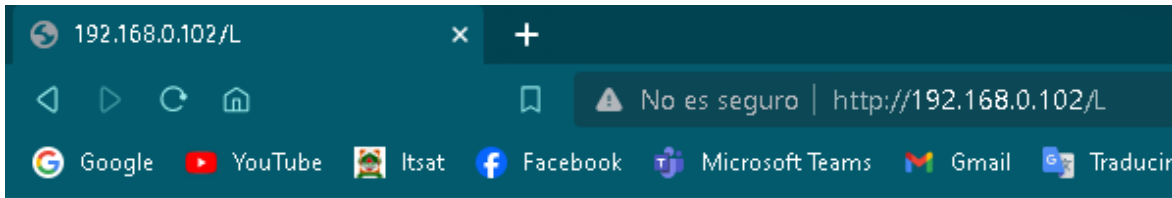
Haz clic [aquí](#) para apagar el LED en el pin 5.





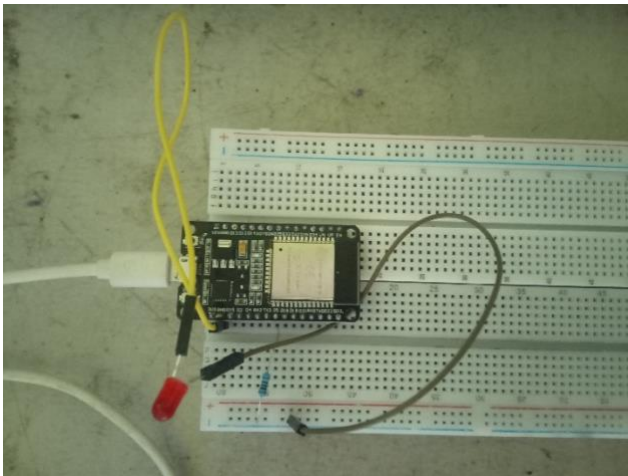
INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Apagamos el LED



Haz clic [aquí](#) para encender el LED en el pin 5.

Haz clic [aquí](#) para apagar el LED en el pin 5.








CONCLUSIONES:

El objetivo principal de los microcontroladores es proporcionar un centro de procesamiento integrado que pueda realizar tareas específicas de control, monitoreo o interacción en sistemas electrónicos.

La implementación y comprensión de la comunicación Wi-Fi utilizando un microcontrolador como el ESP8266 ha revelado un vasto potencial en la interconexión de dispositivos en entornos IoT. Esta práctica ha permitido entender la configuración y uso de un módulo Wi-Fi para establecer conexiones, transmitir datos y controlar dispositivos de forma remota. La versatilidad y la accesibilidad del ESP8266 en términos de conectividad Wi-Fi ofrecen oportunidades infinitas para el desarrollo de aplicaciones inteligentes y sistemas integrados, abriendo las puertas a innovaciones en automatización, monitoreo remoto y soluciones de conectividad avanzadas en diversos campos tecnológicos.

En resumen, los microcontroladores son el núcleo de la mayoría de los sistemas electrónicos, proporcionando control, capacidad de procesamiento y capacidad de comunicación en una amplia gama de aplicaciones, desde las más simples hasta las más complejas.

REFERENCIAS:

-  Brey, Barry B. (2006). Microprocesadores intel : arquitectura, programacion e interfaz : 8086 (7ª edición). : pearson educación isbn: 9789702608042.
-  Palacios, E., Remiro, F., Lopez, L. (2009). Microcontrolador PIC16f84 desarrollo de proyectos. Mexico, D. F.: alfa omega, ra-ma.
-  Brey, Barry B. (Ed.). (2008). Applying pic18 microcontrollers: Architecture, programming, and interfacing using c and assembly. : pearson/prentice hall isbn: 9780130885463.
-  Álvarez Antón, Juan C., Campo Rodríguez, Juan C., (2007). Instrumentación Electrónica.: Paraninfo
-  Pallas/Casas/Bragós. (2008) Sensores y Acondicionadores de Señal. Problemas Resueltos: Marcombo