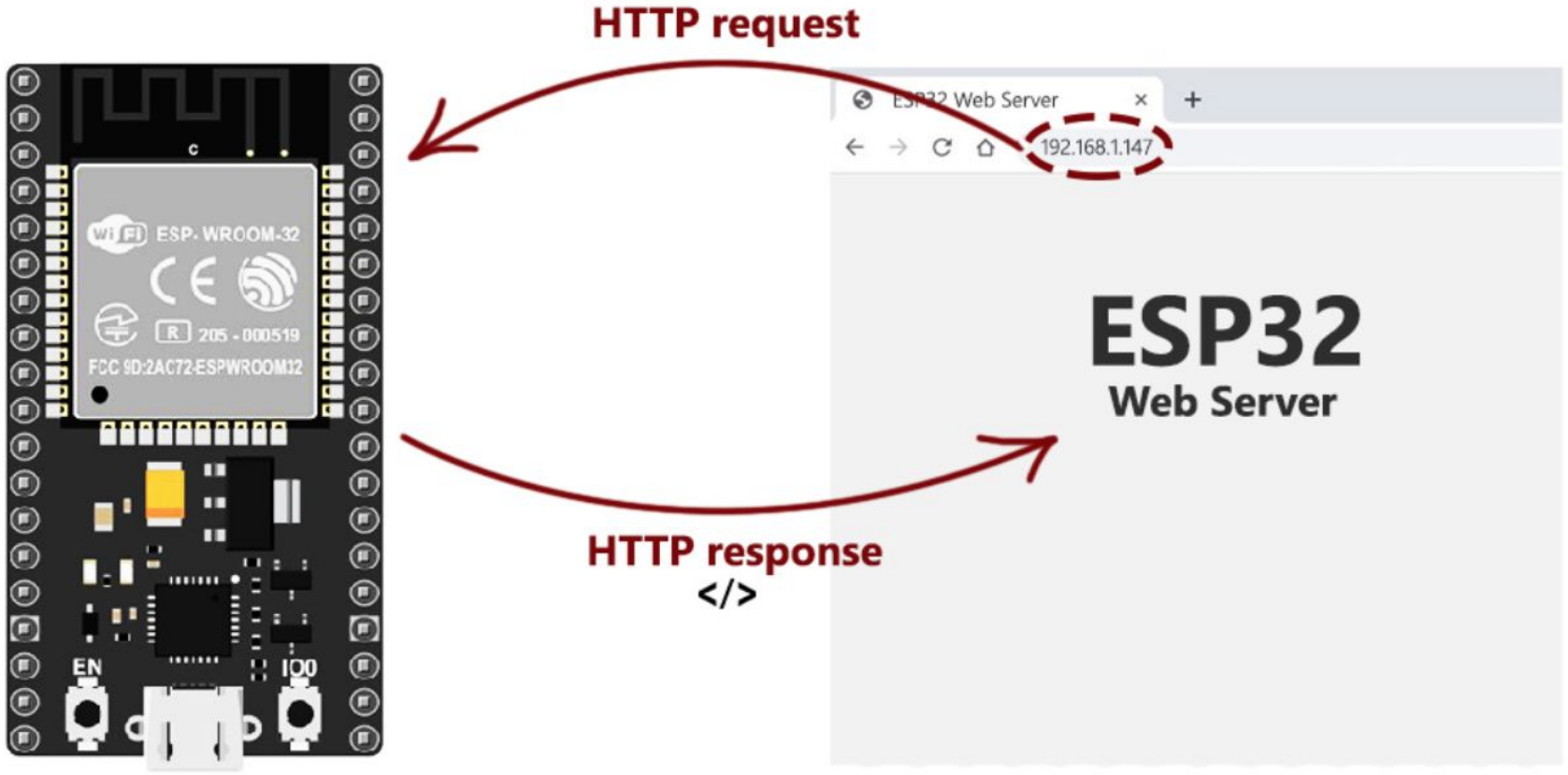
Microcontroladores II

Juan Esteban Giraldo Hoyos
Ingeniero Electrónico
Magíster en Gestión de Ciencia, Tecnología e Innovación

Escuela de Ingenierías
IUSH Institución Universitaria SALAZAR Y HERRERA
SER PERSONA DESDE EL SABER Y EL HACER

Microcontroladores II - Hoy trabajaremos



Escuela de Ingenierías
IUSH Institución Universitaria SALAZAR Y HERRERA
SER PERSONA DESDE EL SABER Y EL HACER

Microcontroladores II - Servidor web

Un servidor web es un software que proporciona servicios de acceso y transferencia de datos a través de la World Wide Web.

En términos simples, es un programa que atiende solicitudes de clientes (navegadores web u otras aplicaciones) y les proporciona recursos web, como páginas HTML, imágenes, archivos, etc, a través del protocolo HTTP.

Un ESP32 es un microcontrolador de bajo costo y bajo consumo de energía que es ampliamente utilizado en proyectos de IoT y sistemas embebidos. Puede ser programado para actuar como un servidor web, permitiendo a otros dispositivos acceder a sus recursos.

Microcontroladores II - Servidor web

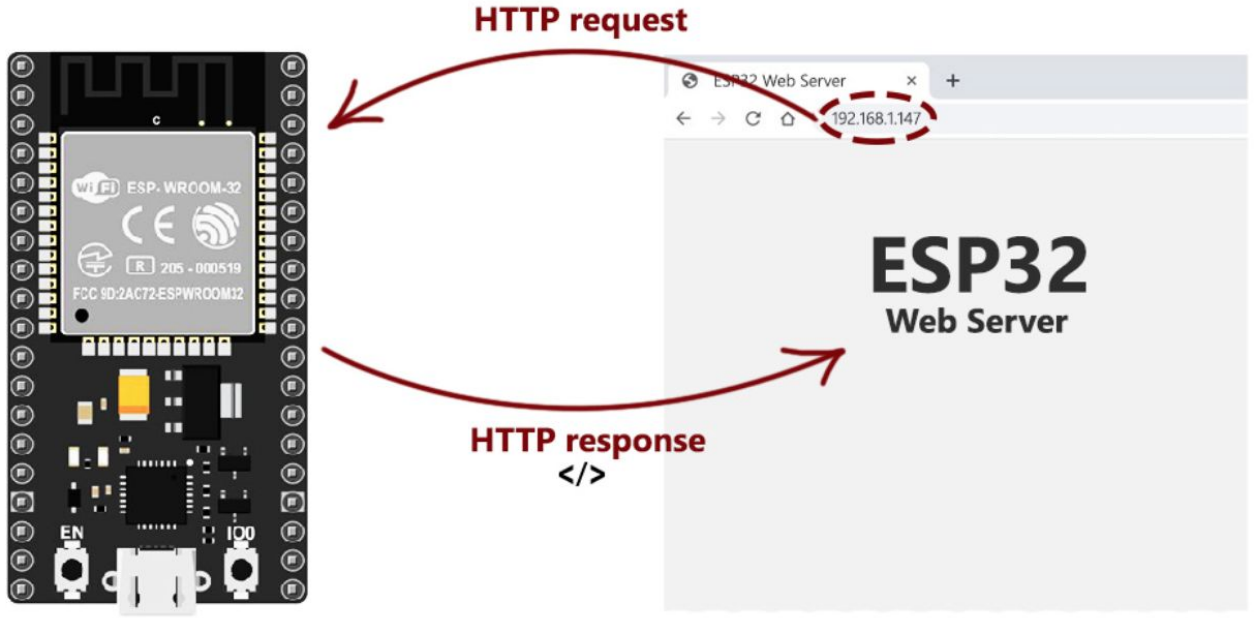
Algunos ejemplos:

1. Monitoreo de sensores: Recopilar datos de sensores ambientales, como temperatura, humedad, y luz, y luego exponer estos datos a través de una interfaz web para que los usuarios puedan monitorearlos desde cualquier navegador.
2. Control remoto de dispositivos: Podrías controlar dispositivos electrónicos, como luces, ventiladores, o cerraduras, desde una interfaz web en tu dispositivo móvil o computadora.
3. Sistema de alarma doméstica: El ESP32 podría actuar como el núcleo de un sistema de alarma doméstica, con sensores de movimiento y puertas que detecten intrusiones, y luego notificar a los propietarios a través de una página web o una aplicación móvil.

Microcontroladores II - Servidor web

Manos a la obra:

Vamos a implementar un firmware que permite conectarnos a una red wifi y crear un servidor web que nos muestre un mensaje de bienvenida al acceder a él



Escuela de Ingenierías
IUSH Institución Universitaria SALAZAR Y HERRERA
ER PERSONA DESDE EL SABER Y EL HACER

Microcontroladores II - Servidor web

FREERTOS_WEB_SERVER §

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

#define ssid  "NOMBRE DE RED WIFI"      // Cambia por tu SSID
#define password  "PASSWORD DE RED WIFI" // Cambia por tu contraseña

WebServer server(80); // Inicializa el servidor en el puerto 80

// Tarea para manejar el servidor web
void handleServer(void *param) {
  for (;;) {
    server.handleClient(); // Maneja las solicitudes del cliente
    vTaskDelay(10 / portTICK_PERIOD_MS); // Pequeña espera para reducir el uso de CPU
  }
}

// Página web para servir al cliente
String getHTML() {
  String html = "<!DOCTYPE html><html>";
  html += "<head><title>BIENVENIDO</title></head>";
  html += "<body>";
  html += "<h1>BIENVENID@ INGENIER@</h1>";
  html += "</body>";
  html += "</html>";
  return html;
}
```

Escuela de Ingenierías
IUSH Institución Universitaria SALAZAR Y HERRERA
SER PERSONA DESDE EL SABER Y EL HACER

Microcontroladores II - Servidor web

FREERTOS_WEB_SERVER §

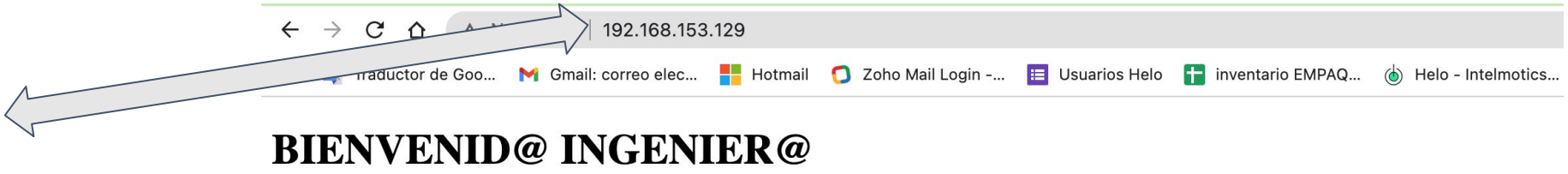
```
// Configura la conexión WiFi y el servidor web
void setup() {
  Serial.begin(115200);

  // Conectar al WiFi
  Serial.println("Conectando al WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConectado al WiFi");
  Serial.print("IP: ");
  Serial.println(WiFi.localIP());

  // Configurar el servidor
  server.on("/", HTTP_GET, []() {
    server.send(200, "text/html", getHTML()); // Enviar la página web con el texto correcto
  });

  server.begin(); // Iniciar el servidor

  // Crear la tarea para manejar el servidor web
  xTaskCreatePinnedToCore(handleServer, "ServerTask", 10000, NULL, 1, NULL, 1);
}

// Bucle vacío, FreeRTOS se encarga de las tareas
void loop() {
  // Nada que hacer aquí
}
```

Escuela de Ingenierías
IUSH Institución Universitaria SALAZAR Y HERRERA
SER PERSONA DESDE EL SABER Y EL HACER

Microcontroladores II - Servidor web

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
ets Jun  8 2016 00:22:57

rst:0x10 (RTCWDT_RTC_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 al WiFi...
..
Conectado al WiFi
IP: 192.168.153.129
```

En el monitor serial debes ver algo como esto



Conectado a la misma red del ESP32 (Servidor), usamos un navegador web y accedemos a la IP del servidor

Microcontroladores II - Servidor web

Ahora vamos a agregar un botón y nuevas funcionalidades para controlar el encendido de un led desde nuestro servidor web

FREERTOS_WEB_SERVER §

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

#define LED_PIN 23 // Puedes cambiar este pin según el ESP32 que uses

#define ssid "NOMBRE DE RED WIFI"      // Cambia por tu SSID
#define password "PASSWORD DE RED WIFI" // Cambia por tu contraseña

WebServer server(80); // Inicializa el servidor en el puerto 80
bool ledState = false; // Estado inicial del LED (apagado)

// Tarea para manejar el servidor web
void handleServer(void *param) {
  for (;;) {
    server.handleClient(); // Maneja las solicitudes del cliente
    vTaskDelay(10 / portTICK_PERIOD_MS); // Pequeña espera para reducir el uso de CPU
  }
}
```

Escuela de Ingenierías
IUSH Institución Universitaria SALAZAR Y HERRERA
SER PERSONA DESDE EL SABER Y EL HACER

Microcontroladores II - Servidor web

```
// Página web para servir al cliente, con texto del botón basado en el estado del LED
String getHTML(bool ledState) {
  String html = "<!DOCTYPE html><html>";
  html += "<head><title>Control de LED</title></head>";
  html += "<body>";
  html += "<h1>Control de LED</h1>";
  html += "<form action=\"/toggle\" method=\"POST\">";
  if (ledState) {
    html += "<button type=\"submit\">Apagar LED</button>";
  } else {
    html += "<button type=\"submit\">Encender LED</button>";
  }
  html += "</form>";
  html += "</body>";
  html += "</html>";
  return html;
}

// Controlador para el botón de encendido/apagado
void handleToggle() {
  ledState = !ledState; // Cambia el estado del LED
  digitalWrite(LED_PIN, ledState); // Enciende o apaga el LED
  server.send(200, "text/html", getHTML(ledState)); // Devuelve la página web con el texto adecuado
}
```

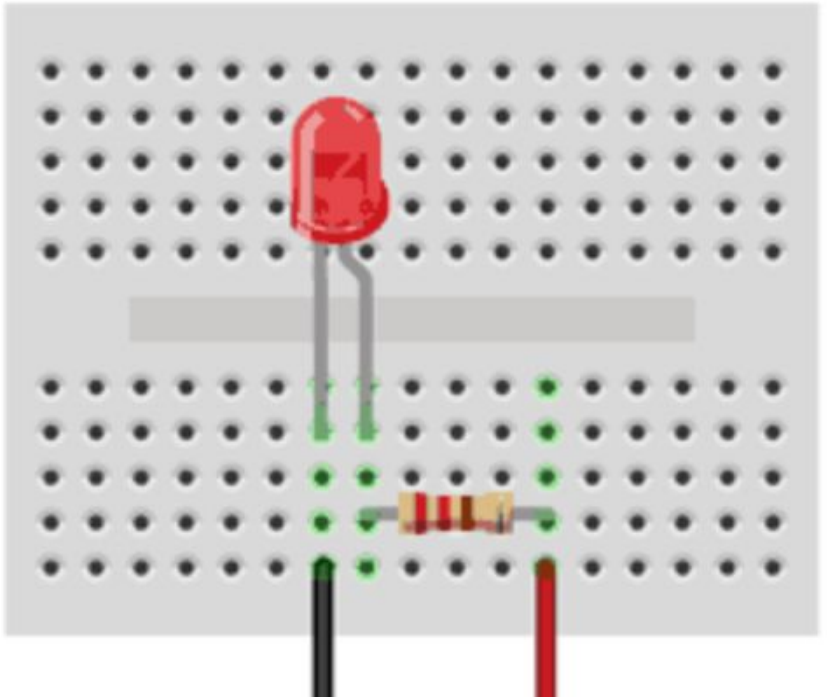
Escuela de Ingenierías
IUSH Institución Universitaria SALAZAR Y HERRERA
SER PERSONA DESDE EL SABER Y EL HACER

Microcontroladores II - Servidor web

FREERTOS_WEB_SERVER §

```
void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
  // Conectar al WiFi
  Serial.println("Conectando al WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConectado al WiFi");
  Serial.print("IP: ");
  Serial.println(WiFi.localIP());

  // Configurar el servidor
  server.on("/", HTTP_GET, []() {
    server.send(200, "text/html", getHTML(ledState)); // Enviar la página web con el texto correcto
  });

  server.on("/toggle", HTTP_POST, handleToggle); // Enlace para el botón
  server.begin(); // Iniciar el servidor

  // Crear la tarea para manejar el servidor web
  xTaskCreatePinnedToCore(handleServer, "ServerTask", 10000, NULL, 1, NULL, 1);
}

// Bucle vacío, FreeRTOS se encarga de las tareas
void loop() {
  // Nada que hacer aquí
}
```

Microcontroladores II - Servidor web



Escuela de Ingenierías
IUSH Institución Universitaria SALAZAR Y HERRERA
SER PERSONA DESDE EL SABER Y EL HACER

FREERTOS Y ESP32 WEB SERVER PRACTICA 15%

Implementar un programa que tenga:
La implementación de un servidor Web usando el ESP32 y FreeRTOS que contenga lo siguiente:

- 1 tarea para el manejo de las peticiones que recibe el servidor (handleServer)
- Una página web que tenga: i) Un botón de encendido y apago para el control del encendido y apagado de un led. ii) La visualización del valor de temperatura más reciente reportado.
- 1 Interrupción de software de Timer que cada 5 seg mande al servidor web el valor de temperatura sensado