#include <Arduino.h>

Esta línea incluye la librería principal de Arduino que permite usar todas las funciones y definiciones necesarias para programar el ESP32.

const int ledPin = 2; // Usualmente el pin 2 en los ESP32 está conectado a un LED integrado.

Aquí se define una constante ledPin que se usa para especificar el pin donde está conectado el LED. En muchos módulos ESP32, el pin 2 está conectado a un LED integrado.

void setup() {

// Inicializa la comunicación serial

Serial.begin(115200);

delay(1000);

// Configura el pin del LED como salida

pinMode(ledPin, OUTPUT);

// Crea las tareas

xTaskCreate(Tarea1, "Tarea1", 10000, NULL, 1, NULL);

xTaskCreate(Tarea2, "Tarea2", 10000, NULL, 1, NULL);

}

Inicia la comunicación serial a una velocidad de 115200 baudios y espera 1 segundo para asegurarse de que la inicialización esté completa.

Configura el pin ledPin (pin 2) como una salida digital. Esto es necesario para poder encender y apagar el LED.

Tarea1: Una tarea llamada "Tarea1" con un stack de 10000 bytes, prioridad 1, y sin parámetros adicionales ni manejador de tarea.

Tarea2: Una tarea llamada "Tarea2" con las mismas características que la anterior.

void loop() {

// El loop principal no hace nada, las tareas se ejecutan independientemente

delay(1000);

}

El loop principal no hace nada aparte de una espera de 1 segundo. En este programa, todas las acciones importantes son manejadas por las tareas creadas.

void Tarea1(void \* parameter) {

// Tarea 1: Imprimir "Hola Mundo" cada segundo

while (true) {

Serial.println("Hola Mundo");

vTaskDelay(1000 / portTICK\_PERIOD\_MS); // Espera 1 segundo

}

vTaskDelete(NULL); // Esto nunca se alcanzará, pero es buena práctica incluirlo

}

Bucle Infinito de Tarea1

Esta tarea imprime "Hola Mundo" en el puerto serial cada segundo. La función vTaskDelay(1000 / portTICK\_PERIOD\_MS) hace que la tarea espere 1 segundo (1000 milisegundos).

Esta línea elimina la tarea actual. Aunque no se alcanzará porque el bucle es infinito, es una buena práctica incluirla.

void Tarea2(void \* parameter) {

// Tarea 2: Parpadear el LED cada 200 ms

while (true) {

digitalWrite(ledPin, HIGH); // Enciende el LED

vTaskDelay(200 / portTICK\_PERIOD\_MS); // Espera 200 ms

digitalWrite(ledPin, LOW); // Apaga el LED

vTaskDelay(200 / portTICK\_PERIOD\_MS); // Espera 200 ms

}

vTaskDelete(NULL); // Esto nunca se alcanzará, pero es buena práctica incluirlo

}

Esta tarea hace que el LED parpadee encendiéndolo y apagándolo cada 200 ms. La función vTaskDelay(200 / portTICK\_PERIOD\_MS) hace que la tarea espere 200 milisegundos entre cada cambio de estado del LED.

Al igual que en Tarea1, esta línea elimina la tarea actual. No se alcanzará debido al bucle infinito, pero se incluye por buenas prácticas.