



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE TIJUANA

**SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN**

SEMESTRE:

Enero - Junio 2024

CARRERA:

Ingeniería en Sistemas Computacionales

MATERIA:

Lenguajes de interfaz

TÍTULO ACTIVIDAD:

Prácticas

UNIDAD A EVALUAR:

Unidad 3

NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:

Muñoz Medina Ramiro Guadalupe - 21212004

Campos Ibarra Sebastian - 21211924

Práctica 0

Descripción General

Este código Arduino está diseñado para la placa Raspberry Pi Pico W. El programa controla un LED conectado a un pin GPIO específico, alternando entre encendido y apagado con un intervalo de 1 segundo. Además, se envían mensajes a través del puerto serial para indicar el estado del LED.

Dependencias

- **Arduino.h:** Biblioteca principal de Arduino que incluye funciones básicas y definiciones de pines.

Variables Globales

- `const int ledPin = 25;` Define el pin GPIO (pin 25) al cual está conectado el LED. Se utiliza `const` para indicar que esta variable no cambiará durante la ejecución del programa.

Funciones del Programa

setup()

Esta función se ejecuta una vez al inicio del programa y se utiliza para configurar el estado inicial de los pines y la comunicación serial.

```
void setup() {  
    pinMode(ledPin, OUTPUT); // Configurar el pin del LED como salida  
    Serial.begin(115200);  
    while (!Serial);  
    Serial.println("Hola, Raspberry Pi Pico W!");  
}
```

- `pinMode(ledPin, OUTPUT);`: Configura el pin definido en `ledPin` como salida.
- `Serial.begin(115200);`: Inicializa la comunicación serial a una velocidad de 115200 baudios.
- `while (!Serial);`: Espera hasta que el puerto serial esté disponible.
- `Serial.println("Hola, Raspberry Pi Pico W!");`: Envía un mensaje inicial por el puerto serial indicando que el setup ha sido completado.

loop()

Esta función se ejecuta repetidamente en un bucle infinito, alternando el estado del LED y enviando mensajes por el puerto serial.

```
void loop() {  
    digitalWrite(ledPin, HIGH); // Encender el LED  
    Serial.println("LED encendido");  
    delay(1000); // Esperar 1 segundo  
    digitalWrite(ledPin, LOW); // Apagar el LED  
    Serial.println("LED apagado");  
    delay(1000); // Esperar 1 segundo  
}
```

- digitalWrite(ledPin, HIGH);: Establece el pin del LED en alto (encendido).
- Serial.println("LED encendido");: Envía un mensaje por el puerto serial indicando que el LED está encendido.
- delay(1000);: Pausa la ejecución del programa durante 1000 milisegundos (1 segundo).
- digitalWrite(ledPin, LOW);: Establece el pin del LED en bajo (apagado).
- Serial.println("LED apagado");: Envía un mensaje por el puerto serial indicando que el LED está apagado.
- delay(1000);: Pausa la ejecución del programa durante 1000 milisegundos (1 segundo).