

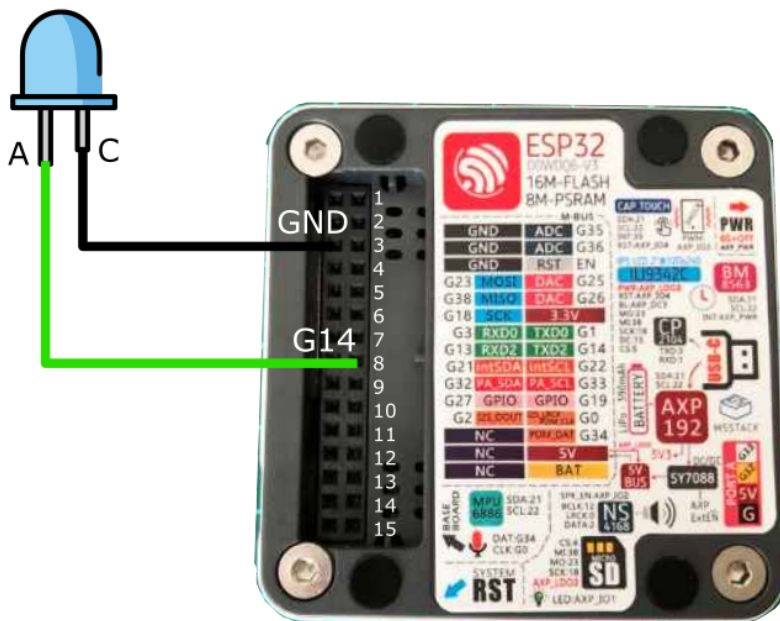
Sesión de Laboratorio 1: Introducción a la programación de microcontroladores

Tiempo estimado: 1.5h (1 sesión)

1. Trabajando con hardware real

1.1. Encender y apagar un LED

Conecta un LED como se muestra:



Ejecuta el siguiente programa:

```
#include <M5Core2.h>
#define LED_PIN 14

void setup() {
  M5.begin(); // Initialize M5Core2
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_PIN, HIGH);
  delay(500);
  digitalWrite(LED_PIN, LOW);
  delay(500);
}
```

PREGUNTA:

- ¿Qué hace `#define`?

- ¿Qué hace la función `pinMode()`?
- ¿Qué hace la función `digitalWrite()`?

PREGUNTA:

- Te habrás dado cuenta de que has tenido que incluir `#include <M5Core2.h>` y `M5.begin(); // Initialize M5Core` ¿Por qué tuviste que hacerlo? ¿Para qué sirven estas instrucciones?

1.2. Ejercicios adicionales

1.2.1. Parpadear a una cierta frecuencia

- Usando el mismo circuito del ejercicio anterior, escribe un script que haga parpadear el LED a una frecuencia de 10 Hz.

PREGUNTA:

- ¿Durante cuánto tiempo debe estar ENCENDIDO o APAGADO el LED en cada ciclo?

1.2.2. Parpadear y detener

- Usando el mismo circuito del ejercicio anterior, escribe un script que haga parpadear el LED a una frecuencia de 2 Hz y detenga el parpadeo después de 5 segundos.

PREGUNTA:

- ¿Cuántas veces parpadea el LED?
- ¿Cuál es el estado final del LED cuando se detiene el parpadeo, permanece ENCENDIDO o APAGADO?

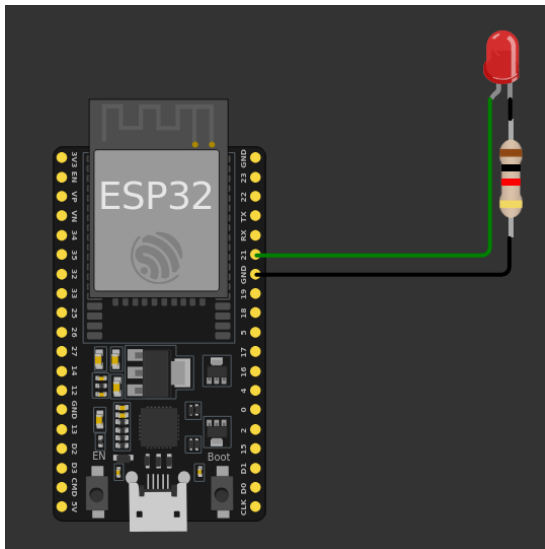
PISTA:

- Puede que necesites usar el bucle `for`, o la condición `if...else`.

2. Trabajando en simulación

2.1. Encender y apagar un LED

Conecta un LED como se muestra:



Ejecuta el siguiente programa:

```
#define LED_PIN 21
#define BUTTON_PIN 35

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

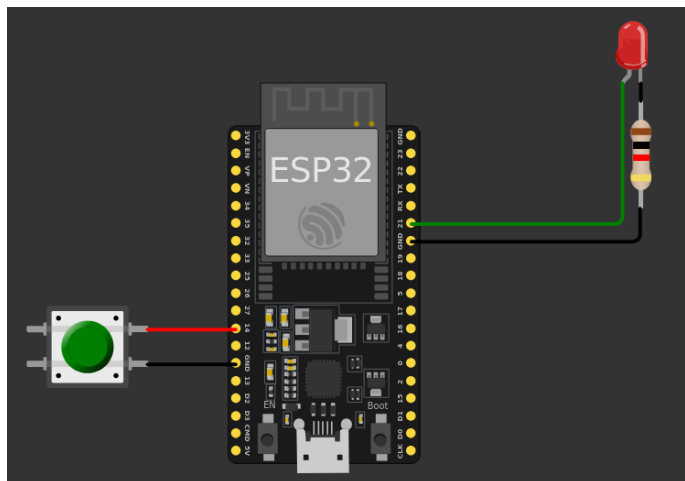
void loop() {
  digitalWrite(LED_PIN, HIGH);
  delay(500);
  digitalWrite(LED_PIN, LOW);
  delay(500);
}
```

2.2. Controlar el estado del LED con un pulsador

- Implementa el circuito mostrado en el siguiente diagrama y simula un programa que:
 - Encienda el LED cuando se pulse el botón.
 - Apague el LED cuando el botón no esté pulsado.

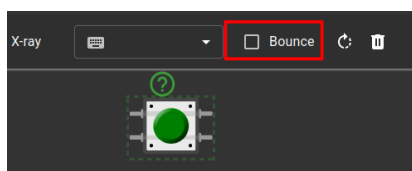
PISTA:

- Usa la función `digitalRead()` para obtener el estado del botón.
- Puede que necesites usar la condición `if...else`.



INFO:

- Al colocar el botón, no olvides deseleccionar la opción *bounce* para evitar problemas de rebotes.



PREGUNTA:

- ¿Qué información se incluye en el `diagram.json`? ¿Cambió con respecto al ejercicio anterior?
- ¿Se puede hacer el ejercicio sin usar la condición `if...else`? ¿Cómo?

2.3. Ejercicios adicionales

2.3.1. Parpadear

- Usando el mismo circuito del ejercicio anterior, escribe un script que haga parpadear el LED mientras se mantiene pulsado el botón.

PISTA:

- Puede que necesites usar la función `millis()`.

2.3.2. Pulsación corta vs pulsación larga

- Usando el mismo circuito del ejercicio anterior, escribe un script que haga lo siguiente:
 - Pulsación corta (< 500 ms): Alternar el LED (si el LED está ENCENDIDO, apagarlo, y viceversa).
 - Pulsación larga (≥ 500 ms): Hacer parpadear el LED.

PISTA:

- Puede que necesites usar la función `millis()` y la condición `if...else`.