

Sesión de Laboratorio 2: Temporizadores

Tiempo estimado: 1.5h (1 sesión)

Descripción

El objetivo de esta sesión es aprender las diferentes formas en que un microcontrolador puede realizar operaciones de temporización. Como se vio anteriormente, la función `delay()` se puede utilizar para crear una espera durante el tiempo requerido. Sin embargo, para resolver problemas ligeramente más complejos, se vuelve necesario utilizar *temporizadores*, que nos permiten interrumpir la ejecución normal del programa para manejar un evento periódico.

1. Trabajando con hardware real

1.1. Usando una biblioteca externa - *Ticker*

La [Biblioteca Ticker de Arduino](#) permite crear fácilmente callbacks de Ticker, que pueden llamar a una función en un intervalo predeterminado. El siguiente ejemplo muestra cómo realizar operaciones de temporización utilizando la biblioteca *Ticker* y las funciones `attach()` o `attach_ms()`.

```
#include <M5Core2.h>    // M5Stack Core2 library
#include <Ticker.h>      // Ticker library

#define LED_PIN 14      // GPIO pin where the LED is connected on the
                        // M5Core2

Ticker blinker;         // Create a Ticker object. It will call a function
                        // at regular intervals.

/*
   This function (callback) will be called by the Ticker at the configured
   time interval.
   It simply toggles the LED state:
   - digitalWrite(LED_PIN) reads the current state (HIGH/LOW)
   - ! inverts that state
   - digitalWrite writes the inverted state back to the pin
*/
void changeState()
{
    digitalWrite(LED_PIN, !(digitalRead(LED_PIN)));
}

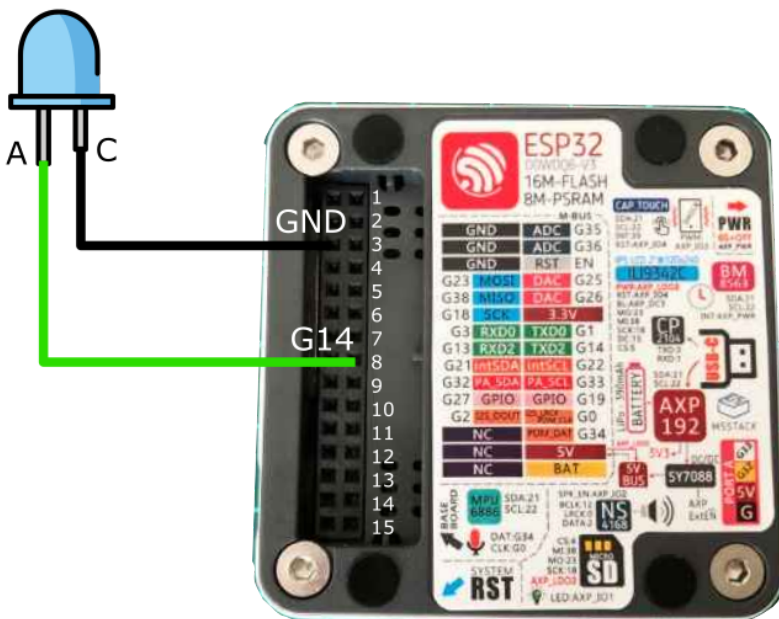
void setup()
{
    M5.begin();          // Initialize the M5Core2 hardware (display, I2C,
                        // etc.)
    pinMode(LED_PIN, OUTPUT); // Configure the LED pin as an output

    // Initialize the Ticker to call changeState() periodically.
```

```
// attach() accepts time in seconds (here 0.5 s = 500 ms).
// If you prefer milliseconds, use attach_ms(milliseconds, callback).
blinker.attach(0.5, changeState);
}

void loop()
{
    // Empty: the Ticker handles the periodic LED toggling asynchronously.
}
```

Conecta una LED al *pin 14* del *M5Core2* y modifica este ejemplo para que la LED parpadee cada segundo en lugar de cada 500 ms.



1.2. Usar temporizadores de hardware

Sin embargo, como se indica en la documentación de la biblioteca *Ticker*, "La biblioteca no utiliza interrupciones de los temporizadores de hardware y funciona con las funciones `micros()` / `millis()`". Esto significa que nuestra capacidad para desarrollar una aplicación utilizando diferentes configuraciones de temporizador o para implementar aplicaciones multitarea estaría muy limitada si nos restringimos a usar esta biblioteca.

Realiza el ejercicio anterior nuevamente, pero en este caso utiliza una interrupción generada directamente por el temporizador y configúralo para que la LED se encienda y apague cada segundo. Verifica con un reloj que funcione correctamente.

2. Trabajando en simulación

Puedes realizar exactamente los mismos ejercicios en simulación. Simplemente utiliza el mismo circuito con la LED que usamos en las sesiones de laboratorio anteriores.V