

## LAB 1: BLINKING LEDs

### Introducción

El objetivo de esta práctica es familiarizarse con el software de STM32CubeMX y del IDE de desarrollo (VS Code + PlatformIO) así como con el microcontrolador STM32F-F411RE.

El circuito debe constar de un pulsador y dos LEDs. El funcionamiento debe ser el siguiente:

1. LEDs encendidos por defecto.
2. Cuando se pulsa el botón:
  - a. Se enciende uno de los LEDs durante 1000ms.
  - b. Se apaga dicho LED y se enciende el otro durante 1000ms.
  - c. Se repiten a) y b) en 5 ocasiones.

Todo esto debe ser implementado mediante manipulación directa de registros. Es decir, sin recurrir a las funciones derivadas.

### Componentes

- Microcontrolador STM32F-F411RE
- Placa de expansión Base Shield Seeed Groove V2
- LED amarillo
- LED rojo
- Pulsador con conector de 4 pines
- Distintos cables macho-macho
- Conector con 4 pines (para botón)
- Resistencias

Para la elección de las resistencias adecuadas hemos realizado previamente la Ley de Ohm. Partimos de un voltaje de la placa de 3'3V. Dado que el voltaje de los LEDs es un intervalo, hemos elegido el mínimo V<sub>dd</sub> de estos ya que supone la mayor diferencia respecto del voltaje fuente. Los cálculos han sido los siguientes:

#### Led rojo:

$$R = \frac{V_f - V_L}{I} = \frac{1.5}{0.02} = 75 \, \Omega$$

Dónde V<sub>f</sub> es el voltaje de la fuente (placa) y V<sub>L</sub> el voltaje del LED. La intensidad es de 20 mA (0'02 A).

Por tanto, la resistencia elegida es la de 75  $\Omega$ .

#### Led amarillo:

$$R = \frac{V_f - V_L}{I} = \frac{1.9}{0.005} = 280 \, \Omega$$

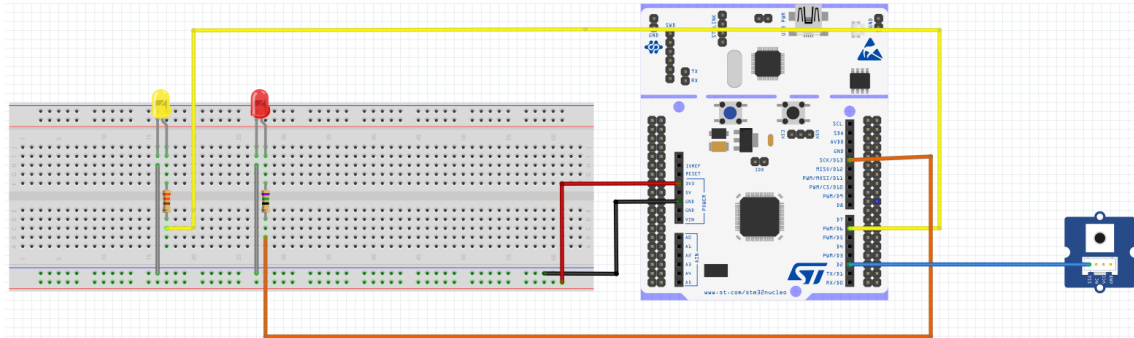
Dónde V<sub>f</sub> es el voltaje de la fuente (placa) y V<sub>L</sub> el voltaje del LED. La intensidad es de 5 mA (0'005 A).

Por tanto, la resistencia elegida es la de 330  $\Omega$ .

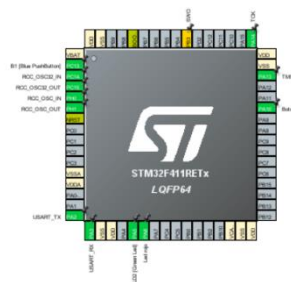
## Esquema del circuito (en aplicación Fritzing)

En primer lugar, debemos mencionar los pines que vamos a utilizar como entradas y salidas.

- Entradas:
  - o Pulsador
  - o Pin D2 (PA\_10)
- Salidas:
  - o LED rojo y LED amarillo
  - o Pin D13 (PA\_5) y Pin D6 (PB\_10) respectivamente.



## Esquema STM32 Cube MX



Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull-up...	Maximum ou...	User Label	Modified
PA5	n/a	Low	Output Push ...	No pull-up an...	Low	LD2 [Red Led]	<input checked="" type="checkbox"/>
PA10	n/a	n/a	Input mode	No pull-up an...	n/a	boton	<input checked="" type="checkbox"/>
PB10	n/a	Low	Output Push ...	No pull-up an...	Low	LD1[Yellow L...	<input checked="" type="checkbox"/>
PC13-ANTI_...	n/a	n/a	External Inter...	No pull-up an...	n/a	B1 [Blue Pus...	<input checked="" type="checkbox"/>

## Solución propuesta en PlatformIO

```

if ((GPIOA -> IDR & GPIO_PIN_10) != (uint32_t)GPIO_PIN_RESET) // Se comprueba si el pin A10 es 1
{
    for (size_t i = 0; i < 5; i++)
    {
        GPIOB -> ODR &= ~GPIO_ODR_OD10_Msk; //pin B10 se apaga
        GPIOA -> ODR |= GPIO_ODR_OD5_Msk; //pin A5 se enciende

        HAL_Delay(1000);
        GPIOA -> ODR &= ~GPIO_ODR_OD5_Msk; //pin A5 se apaga
        GPIOB -> ODR |= GPIO_ODR_OD10_Msk; //pin B10 se enciende
        HAL_Delay(1000);
        GPIOB -> ODR &= ~GPIO_ODR_OD10_Msk; //pin B10 se apaga
    }
}

//Los dos LEDs encendidos
GPIOA -> ODR |= GPIO_ODR_OD5_Msk; //pin A5 se enciende
GPIOB -> ODR |= GPIO_ODR_OD10_Msk; //pin B10 se enciende

```

Como podemos observar, la lógica del programa es bastante sencilla. Únicamente, se debe comprobar si el botón (INPUT) está activado y en caso afirmativo, mediante un bucle for, repetir 5 veces la intermitencia de los LEDs.

Para la elaboración del código hemos seguido las diapositivas del módulo 6.

El condicional del if nos sirve para saber el valor del registro del botón.

Para limpiar el valor del registro (apagar el LED) lo que hacemos aplicar una máscara negada ( $\sim$ ) y una operación AND, con lo que eliminamos el valor del registro sin alterar el resto.

De forma análoga, para establecer un valor en el registro (encender) se aplica una máscara y una operación OR.

### Secuencia del programa (Fotos)

