



Configuramos dos interrupciones externas: PB0 y PB3

Conectaremos un pulsador para provocar una interrupción

# Interrupciones externas en STM32F407

---

```
static void MX_GPIO_Init(void)
{
    ...

    /*Configure GPIO pins : PB0 PB3: Interrupcion botones externos */
    GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_3;
    GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    ...

    /* EXTI interrupt init*/
    HAL_NVIC_SetPriority(EXTI0_IRQn, configMAX_SYSCALL_INTERRUPT_PRIORITY -2,0);
    HAL_NVIC_EnableIRQ(EXTI0_IRQn);

    HAL_NVIC_SetPriority(EXTI3_IRQn, configMAX_SYSCALL_INTERRUPT_PRIORITY -1,0);
    HAL_NVIC_EnableIRQ(EXTI3_IRQn);

}
```

```
/* Funcion para el tratamiento de interrupciones - RTI */
```

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    long yield = pdFALSE;
    // Prevent unused argument(s) compilation warning
    UNUSED(GPIO_Pin);

    portYIELD_FROM_ISR(yield);
}
```

# Interrupciones externas en STM32F407

---

```
int main(void)
{

...

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_ADC1_Init();
MX_SPI1_Init();
MX_CAN1_Init();

/* Crear semáforo binario para sincronizar t-esporádica e interrupción */

/* Crear la tarea esporádica que se desbloqueará con la RTI */

/* Start scheduler */
osKernelStart();
```