

## Mission Objective:

- **Configure SysTick** on the LPC1769 to handle multiple channels and capture events ( cambio→**EINT**) to simulate a **security system**.
- **Use capture events** to simulate sensors (e.g., motion detectors) detecting movement and triggering actuators like **doors** or **alarms**.
- Set the **prescaler** correctly to generate a time base that allows for precise time measurements.
- When the **first sensor** detects motion, simulate a **door closing** (which takes 10 seconds to close over a distance of 10 meters).
- If the **second sensor** (near the door) detects a person during the closing sequence, trigger the **alarm**.
- The system will consider everything safe if no alarm is triggered and no motion is detected within 30 seconds, leaving the door open after that.
- Beware of **false positives** from rats! They might trigger the sensors but don't pose any real danger

**2 INPUT // 2 OUTPUT**

**Sensor 1:** detecta movimiento y dispara el cierre de una puerta

**Puerta:** simular que pasan 10s

**Sensor 2:** si se detecta una **persona** DURANTE EL CIERRE de la puerta, se dispara una alarma

**Alarma:** se dispara si detecta a una persona durante el cierre de la puerta que activa el sensor 1

**!** Si no se detecta movimiento (sensor 1) ni se dispara la alarma (sensor 2), dejar la puerta abierta (simular apertura de 10s si estaba cerrada?)

**!** Cuidado con falsos positivos de ratas: sería como el rebote de botón?

**!** EINT >> SysTick

**DIAGRAMAS:**

[https://lucid.app/lucidchart/6852a824-e6a5-442b-b502-7f997b129d71/edit?viewport\\_loc=298%2C-72%2C1845%2C1104%2C0\\_0&invitationId=inv\\_6c2460ee-2da3-4a25-9100-e1198b13c9cf](https://lucid.app/lucidchart/6852a824-e6a5-442b-b502-7f997b129d71/edit?viewport_loc=298%2C-72%2C1845%2C1104%2C0_0&invitationId=inv_6c2460ee-2da3-4a25-9100-e1198b13c9cf)

## CÓDIGO:

```
#include "lpc17xx.h" //Contiene las definiciones para registros específicos del
microcontrolador

#include "lpc17xx_gpio.h" //Proporciona funciones para manejar los pines Input/Output

#include "lpc17xx_pinsel.h" // Proporciona funciones para seleccionar las funciones de los pines

#include "lpc17xx_systick.h" //Contiene Funciones para el manejo del temporizador Systick

/ mi mama me dio la vida, el equipo azul las ganas de vivirla

/* Boolean Variables */

#define TRUE 1

#define FALSE 0

/* Time Variables */

#define SYSTICK_TIME_MS 100 /* Como máximo el Systick puede contar hasta 167 ms p/100 MHz */

#define NO_HARM_TIME 300

/* GPIO Direction Definitions */

#define INPUT 0

#define OUTPUT 1

/* Interrupt Variables */

#define ENABLE 1

#define DISABLE 0

/* Edge Variables */

#define RISING_EDGE 0

#define FALLING_EDGE 1

/* Input Pins Definitions */

#define SENSOR1_PIN((uint32_t)(1<<5)) //Se define el Pin del sensor 1

#define SENSOR2_PIN((uint32_t)(1<<6)) //Se define el Pin del sensor 2

/* Output Pins Definitions */
```

```
#define ALARM_PIN((uint32_t)(1<<7)) //Se define el Pin de la alarma
```

```
#define DOOR_PIN((uint32_t)(1<<8)) //Se define el Pin de la puerta
```

```
uint8_t sensor1Flag = 0; // Flag de disparo sensor 1
```

```
uint8_t sensor2Flag = 0; // Flag de disparo sensor 2
```

```
uint8_t flagPuerta = 0;
```

```
uint16_t count = 0; // TODO
```

```
int main(){
```

```
    SystemInit(); // Inicializa el LPC1769 con configuraciones básicas con el clock por defecto a 100MHz
```

```
    configure_systick();
```

```
    configure_pins();    //Configura los pines del LPC
```

```
    while(TRUE)
```

```
    {
```

```
    }
```

```
    return 0;
```

```
}
```

```
// @brief Configura el Temporizador Systick
```

```
void configure_systick()
```

```
{
```

```
    SYSTICK_IntCmd(ENABLE); //Habilita la interrupciones por Systick
```

```
    return;
```

```
}
```

**// @brief Maneja las interrupcion del Systick**

```
void systick_Handler(void){  
  
    SYSTICK_ClearCounterFlag(); // Limpia la bandera de la Interrupción  
  
    if( (count == 100) & flagPuerta ){           //Pasaron 10 segundos  
  
        GPIO_ClearValue(PINSEL_PORT0, DOOR_pin)  
  
        count = 0;  
  
    }  
  
    if(count == 300){           //Pasaron 30 segundos  
  
        if(sensor2Flag)  
  
        {  
  
            GPIO_SetValue(PINSEL_PORT_0, ALARM_PIN);  
  
        }  
        GPIO_ClearValue(PINSEL_PORT0, DOOR_pin);  
  
        count = 0;  
  
    }  
  
    count ++;  
  
    return;  
  
}
```

**// @brief Maneja la interrupcion externa 3(EINT3) generada por algún sensor**

```
void EINT3_IRQHandler(void){  
  
    if(GPIO_GetIntStatus(PINSEL_PORT_0, SENSOR1_PIN, EDGE_FALLING)) //Se activa el Sensor1  
  
    {  
  
        sensor1Flag = 1;  
  
        flagPuerta = 1;  
  
    }
```

```
SYSTICK_Internallnit(SYSTICK_TIME_MS);    //Inicializa el temporizador Systick con un  
intervalo en milisegundos
```

```
SYSTICK_Cmd(ENABLE);  //Habilita el Contador
```

```
}
```

```
if(GPIO_GetIntStatus(PINSEL_PORT_0,SENSOR2_PIN,EDGE_FALLING))
```

```
{
```

```
    sensor2Flag = 1;
```

```
}
```

```
}
```

**// @brief Configura los pines GPIO para los sensores**

**void configure\_pins(void)**

**{**

**PINSEL\_CFG\_Type PinCfg;**

**PinCfg.Portnum = PINSEL\_PORT\_0;**

**PinCfg.Pinnum = PINSEL\_PIN\_5;**

**PinCfg.Funcnum = PINSEL\_FUNC\_0;**

**PinCfg.Pinmode = PINSEL\_PINMODE\_PULLUP;**

**PINSEL\_ConfigPin(&PinCfg);** **//Configuracion del Sensor1**

**PinCfg.Pinnum = PINSEL\_PIN\_6;**

**PINSEL\_ConfigPin(&PinCfg);** **//Configuracion del Sensor2**

**PinCfg.Pinnum = PINSEL\_PIN\_7;**

**PINSEL\_ConfigPin(&PinCfg);** **//Configuracion de la alarma**

**PinCfg.Pinnum = PINSEL\_PIN\_8;**

**PINSEL\_ConfigPin(&PinCfg);** **//Configuracion de la puerta**

**GPIO\_SetDir(PINSEL\_PORT\_0, SENSOR1\_PIN|SENSOR2\_PIN, INPUT);**

**GPIO\_SetDir(PINSEL\_PORT\_0, ALARM\_PIN|DOOR\_PIN, OUTPUT);**

**GPIO\_ClearValue(PINSEL\_PORT\_0, ALARM\_pin);**

**GPIO\_SetValue(PINSEL\_PORT\_0, DOOR\_pin);**

**/\* Al ser los 2 el mismo tipo de interrupción por GPIO (por el mismo flanco), tienen que estar en la misma instrucción \*/**

```
GPIO_IntCmd(PINSEL_PORT_0, SENSOR1_PIN|SENSOR2_PIN, FALLING_EDGE );
```

```
NVIC_EnableIRQ(EINT3_IRQn);
```

```
return;
```

```
}
```