

MICROCONTROLADORES

Práctica No. 12. Interrupciones.

1. Objetivo

- Display LCD alfanumérico 2x16.
- Modulo PCF8574.
- PUSHBUTTON.

2. Material y Equipo.

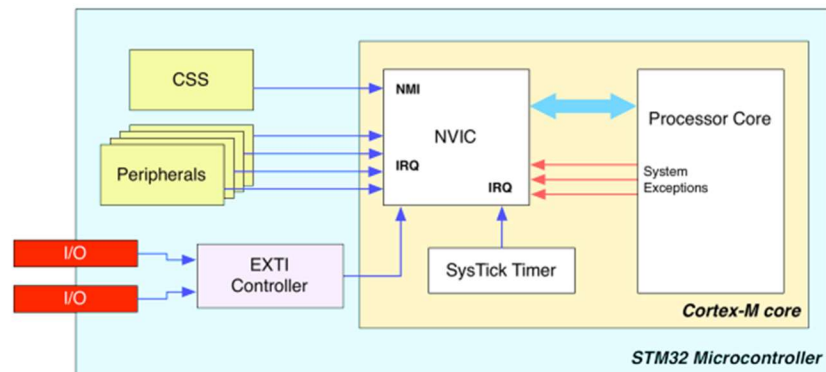
- Computador o laptop con el STM32CubeIDE.
- Un Display LCD alfanumérico.
- Un módulo PC8574.
- Un PUSHBUTTON.

3. Marco de Referencia.

Las interrupciones son una herramienta indispensable en cualquier microcontrolador pues estas nos permiten liberar al CPU de estar verificando constantemente el estado de una bandera de algún periférico. Las interrupciones son asíncronas y pueden ocurrir en cualquier momento y cuando ocurren el CPU termina de ejecutar la última instrucción que estuviera ejecutando y guarda en el Stack la dirección de la siguiente instrucción que va a ejecutar después de retornar de la interrupción una vez guardada esta dirección se guarda también el contexto actual de ejecución y brinca al vector de la interrupción detectada y dentro de ese vector hay un salto a una rutina especial llamada Rutina de Servicio a la Interrupción (ISR) donde se debe ejecuta un código muy corto para reducir la latencia de la interrupción y al finalizar se limpia la bandera que provoco la interrupción y el microcontrolador regresa al programa principal en la última instrucción que se iba a ejecutar y continua el flujo normal del programa.

Los STM32 manejan algo llamado excepción e interrupciones. Las excepciones son interrupciones especiales y tienen la mayor prioridad dentro de la tabla de interrupciones y dicha prioridad no se puede cambiar, para poder servir tanto a las interrupciones como a las excepciones los procesadores ARM tienen un controlador de excepciones especial llamado NVIC (Nested Vectored Interrupt Controller) este controlador está solo disponible en los microcontroladores ARM Cortex-M para los procesadores ARM Cortex-A se deja al fabricante del procesador agregar su propio controlador de interrupciones. El NVIC se encarga de dar servicio a las interrupciones que ocurren dentro y fuera del microcontrolador, tales como las interrupciones externas EXTI, USART, SPI, Timers, etc.

La siguiente imagen muestra la estructura simplificada del NVIC.



La tabla de interrupciones según la hoja de datos para el STM32F103C8T6 se muestra a continuación.



Position	Priority	Type of priority	Acronym	Description	Address
-	-	-	-	Reserved	0x0000_0000
-	-3	fixed	Reset	Reset	0x0000_0004
-	-2	fixed	NMI	Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000_0008
-	-1	fixed	HardFault	All class of fault	0x0000_000C
-	0	settable	MemManage	Memory management	0x0000_0010
-	1	settable	BusFault	Prefetch fault, memory access fault	0x0000_0014
-	2	settable	UsageFault	Undefined instruction or illegal state	0x0000_0018
-	-	-	-	Reserved	0x0000_001C - 0x0000_002B
-	3	settable	SVCALL	System service call via SWI instruction	0x0000_002C
-	4	settable	Debug Monitor	Debug Monitor	0x0000_0030
-	-	-	-	Reserved	0x0000_0034
-	5	settable	PendSV	Pendable request for system service	0x0000_0038
-	6	settable	SysTick	System tick timer	0x0000_003C
0	7	settable	WWDG	Window watchdog interrupt	0x0000_0040
1	8	settable	PVD	PVD through EXTI Line detection interrupt	0x0000_0044
2	9	settable	TAMPER	Tamper interrupt	0x0000_0048
3	10	settable	RTC	RTC global interrupt	0x0000_004C
4	11	settable	FLASH	Flash global interrupt	0x0000_0050
5	12	settable	RCC	RCC global interrupt	0x0000_0054
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000_0058
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000_005C
8	15	settable	EXTI2	EXTI Line2 interrupt	0x0000_0060
9	16	settable	EXTI3	EXTI Line3 interrupt	0x0000_0064
10	17	settable	EXTI4	EXTI Line4 interrupt	0x0000_0068
11	18	settable	DMA1_Channel1	DMA1 Channel1 global interrupt	0x0000_006C
12	19	settable	DMA1_Channel2	DMA1 Channel2 global interrupt	0x0000_0070
13	20	settable	DMA1_Channel3	DMA1 Channel3 global interrupt	0x0000_0074



14	21	settable	DMA1_Channel4	DMA1 Channel4 global interrupt	0x0000_0078
15	22	settable	DMA1_Channel5	DMA1 Channel5 global interrupt	0x0000_007C
16	23	settable	DMA1_Channel6	DMA1 Channel6 global interrupt	0x0000_0080
17	24	settable	DMA1_Channel7	DMA1 Channel7 global interrupt	0x0000_0084
18	25	settable	ADC1_2	ADC1 and ADC2 global interrupt	0x0000_0088
19	26	settable	USB_HP_CAN_TX	USB High Priority or CAN TX interrupts	0x0000_008C
20	27	settable	USB_LP_CAN_RX0	USB Low Priority or CAN RX0 interrupts	0x0000_0090
21	28	settable	CAN_RX1	CAN RX1 interrupt	0x0000_0094
22	29	settable	CAN_SCE	CAN SCE interrupt	0x0000_0098
23	30	settable	EXTI9_5	EXTI Line[9:5] interrupts	0x0000_009C
24	31	settable	TIM1_BRK	TIM1 Break interrupt	0x0000_00A0
25	32	settable	TIM1_UP	TIM1 Update interrupt	0x0000_00A4
26	33	settable	TIM1_TRG_COM	TIM1 Trigger and Commutation interrupts	0x0000_00A8
27	34	settable	TIM1_CC	TIM1 Capture Compare interrupt	0x0000_00AC
28	35	settable	TIM2	TIM2 global interrupt	0x0000_00B0
29	36	settable	TIM3	TIM3 global interrupt	0x0000_00B4
30	37	settable	TIM4	TIM4 global interrupt	0x0000_00B8
31	38	settable	I2C1_EV	I ² C1 event interrupt	0x0000_00BC
32	39	settable	I2C1_ER	I ² C1 error interrupt	0x0000_00C0
33	40	settable	I2C2_EV	I ² C2 event interrupt	0x0000_00C4
34	41	settable	I2C2_ER	I ² C2 error interrupt	0x0000_00C8
35	42	settable	SPI1	SPI1 global interrupt	0x0000_00CC
36	43	settable	SPI2	SPI2 global interrupt	0x0000_00D0
37	44	settable	USART1	USART1 global interrupt	0x0000_00D4
38	45	settable	USART2	USART2 global interrupt	0x0000_00D8
39	46	settable	USART3	USART3 global interrupt	0x0000_00DC
40	47	settable	EXTI15_10	EXTI Line[15:10] interrupts	0x0000_00E0
41	48	settable	RTCAlarm	RTC alarm through EXTI line interrupt	0x0000_00E4
42	49	settable	USBWakeup	USB wakeup from suspend through EXTI line interrupt	0x0000_00E8
43	50	settable	TIM8_BRK	TIM8 Break interrupt	0x0000_00EC
44	51	settable	TIM8_UP	TIM8 Update interrupt	0x0000_00F0
45	52	settable	TIM8_TRG_COM	TIM8 Trigger and Commutation interrupts	0x0000_00F4
46	53	settable	TIM8_CC	TIM8 Capture Compare interrupt	0x0000_00F8
47	54	settable	ADC3	ADC3 global interrupt	0x0000_00FC
48	55	settable	FSMC	FSMC global interrupt	0x0000_0100
49	56	settable	SDIO	SDIO global interrupt	0x0000_0104
50	57	settable	TIM5	TIM5 global interrupt	0x0000_0108
51	58	settable	SPI3	SPI3 global interrupt	0x0000_010C
52	59	settable	UART4	UART4 global interrupt	0x0000_0110
53	60	settable	UART5	UART5 global interrupt	0x0000_0114
54	61	settable	TIM6	TIM6 global interrupt	0x0000_0118
55	62	settable	TIM7	TIM7 global interrupt	0x0000_011C
56	63	settable	DMA2_Channel1	DMA2 Channel1 global interrupt	0x0000_0120
57	64	settable	DMA2_Channel2	DMA2 Channel2 global interrupt	0x0000_0124
58	65	settable	DMA2_Channel3	DMA2 Channel3 global interrupt	0x0000_0128
59	66	settable	DMA2_Channel4_5	DMA2 Channel4 and DMA2 Channel5 global interrupts	0x0000_012C

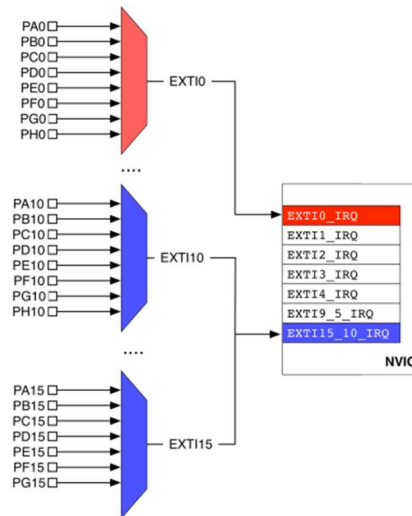
Para saber mejor cuales interrupciones son las que soporta el STM32F103C8T6 podemos consultar el archivo

“startup_stm32f103c8tx.s” en la carpeta “Core/Startup” dentro del proyecto.

Como se puede ver en la tabla anterior cada excepción (interrupción) tiene una prioridad la excepción con el número de prioridad más baja es la excepción con mayor prioridad sobre otras con número de prioridad más alto.

Los procesadores Cortex-M3 y Cortex-M4 tienen un registro de prioridades donde nosotros podemos cambiar la prioridad de alguna excepción para que sea atendida con mayor prioridad sobre otras este registro es el IPR y es de ocho bits de los cuales solo se utilizan los 4 bits más significativos de este registro para la prioridad teniendo las prioridades 0x10, 0x20, 0x30, 0x40, 0x50, 0x60, 0x70, 0x80, 0x90, 0xA0, 0xB0, 0xC0, 0xD0, 0xE0 y 0xF0. De este mismo registro nosotros podemos partir estos bits y asignar 1, 2 o 3 de los 4 bits para sub-prioridades. Las sub-prioridades solo dicen si dos interrupciones que están en espera cuál de esas dos tiene la sub-prioridad de mayor prioridad es reiniciada primero.

Cada pin del STM32 tiene la capacidad de emitir una interrupción por cambio de estado de alguno de sus pines, a estas interrupciones se les conoce como interrupciones externas (EXTI). Estas interrupciones se agrupan por ejemplo los pines PA0, PB0 y PC0 se agrupan al EXTI0 por lo cual no se puede usar el PA0 y PB0 con interrupción externa al mismo tiempo como se muestra en la siguiente figura.



Lo primero que tenemos que hacer es habilitar la interrupción externa para habilitar cualquier interrupción las librerías HAL nos proporcionan la siguiente función.

```
void HAL_NVIC_EnableIRQ(IRQn_Type IRQn);
```

Esta función recibe como parámetro la interrupción que se desea habilitar en nuestro caso será “EXTIO_IRQn”, para deshabilitar cualquier interrupción tenemos la siguiente función.

```
void HAL_NVIC_DisableIRQ(IRQn_Type IRQn);
```

Por último, tenemos otra función para poder establecer la prioridad de las interrupciones.

```
void HAL_NVIC_SetPriority(IRQn_Type IRQn, uint32_t PreemptPriority, uint32_t SubPriority);
```

Esta función recibe como parámetros la interrupción (en nuestro caso EXTI0_IRQn), el segundo parámetro es la prioridad que puede ser de 0 a 15 y por último la sub-prioridad que por el momento la vamos a manejar en 0.

La configuración de la interrupción externa sería de la siguiente forma.

```
void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin : PA0 */
    GPIO_InitStruct.Pin = GPIO_PIN_0;
    GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
    GPIO_InitStruct.Pull = GPIO_PULLDOWN;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

    /* EXTI interrupt init*/
    HAL_NVIC_SetPriority(EXTI0_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(EXTI0_IRQn);
}
```

Podemos ver en el código anterior que en la opción “GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING” con esta opción el STM32 está esperando un cambio de Bajo a Alto para provocar una interrupción.

Las opciones que tenemos para configurar las interrupciones externas son las siguientes.

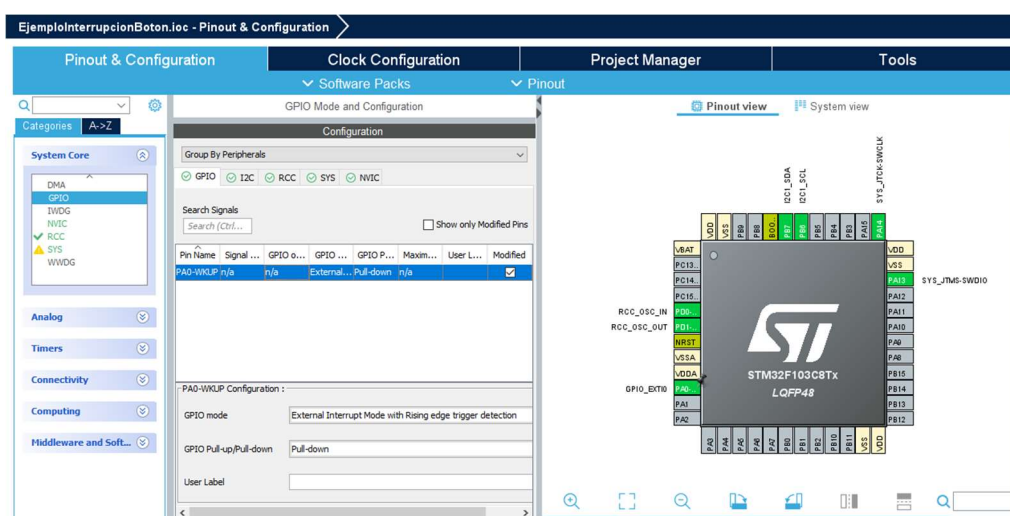
```
#define GPIO_MODE_IT_RISING
#define GPIO_MODE_IT_FALLING
#define GPIO_MODE_IT_RISING_FALLING
```

GPIO_MODE_IT_RISING: Detecta el cambio de flanco de bajo a alto.
GPIO_MODE_IT_FALLING: Detecta el cambio de flanco de alto a bajo.
GPIO_MODE_IT_RISING_FALLING: Detecta el cambio en ambos flancos.

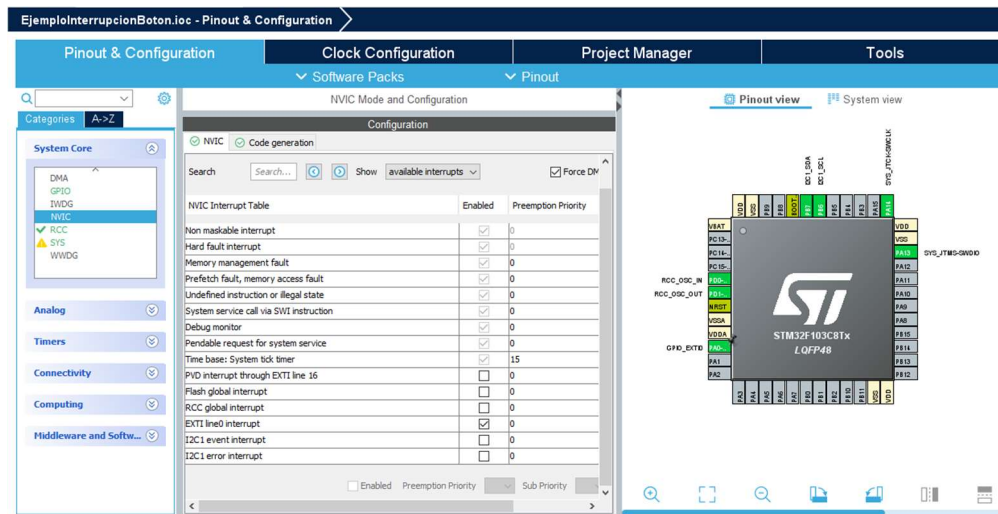
4. Desarrollo y Procedimiento.

Se creará un proyecto en el STM32CubeIDE como se indicó anteriormente.

Damos click sobre el pin PA0 y seleccionamos la opción “GPIO_EXTI0” después en la columna izquierda vamos a “System Core/GPIO” donde haremos la siguiente configuración.



Después vamos a “System Core/NVIC” se nos muestra una tabla en la columna central como se muestra a continuación y lo único que hacemos en la columna “enable” de la tabla palomeamos la opción “EXTI line0 interrupt”. La configuración deberá queda como se muestra en la siguiente imagen.



El código de la práctica es el siguiente.

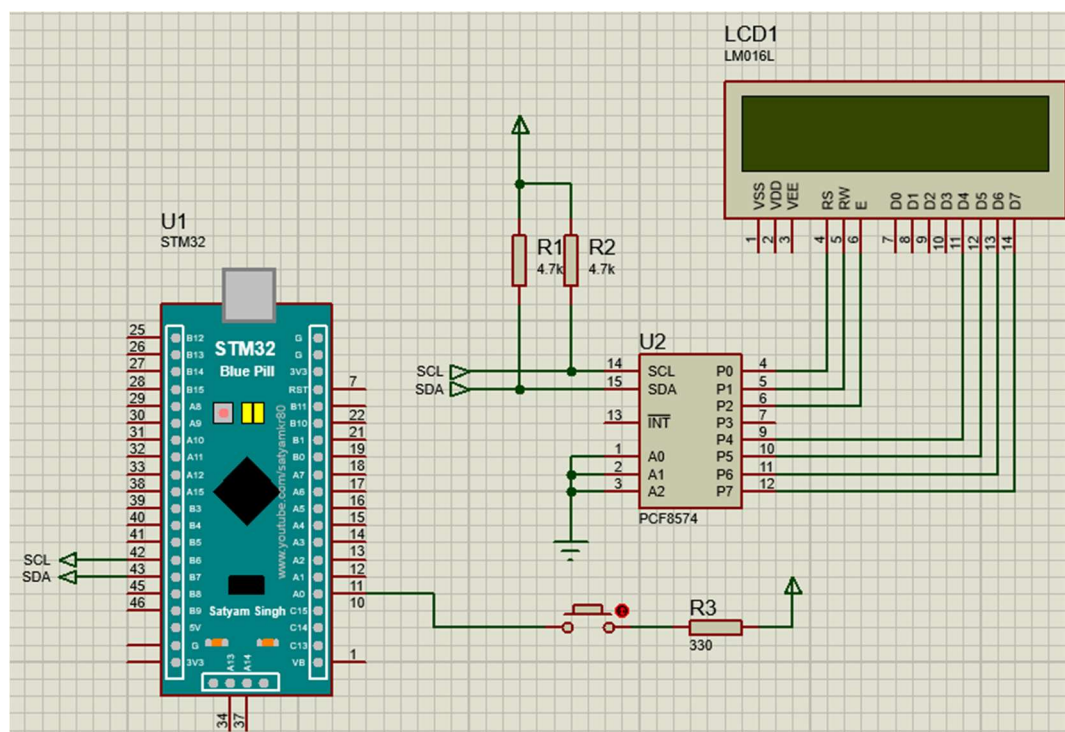
```
1 #include "main.h"
2 #include "i2c.h"
3 #include "gpio.h"
4 #include "lcd.h"
5 #include <stdio.h>
6
7 char msg1[] = "Contador";
8 char outStr[5];
9 volatile uint8_t conta;
10 volatile uint8_t isrFlag;
11
12 void SystemClock_Config(void);
13
14 int main(void)
15 {
16     HAL_Init();
17     SystemClock_Config();
18     MX_GPIO_Init();
19     MX_I2C1_Init();
20
21     Lcd_Init();
22     Lcd_Gotoxy(1, 1);
23     Lcd_Print(msg1);
24     Lcd_Gotoxy(1, 2);
25     sprintf(outStr, "%3d", conta);
26     Lcd_Print(outStr);
27
28     while (1)
29     {
30         if(isrFlag){
31             isrFlag = 0;
32             if(conta > 100)
33                 conta = 0;
34             sprintf(outStr, "%3d", conta);
35             Lcd_Gotoxy(1, 2);
36             Lcd_Print(outStr);
37         }
38     }
39 }
```

Función callback.

```
80 /* USER CODE BEGIN 4 */
81 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
82     isrFlag = 1;
83     conta++;
84 }
85 /* USER CODE END 4 */
```

5. Esquemático del circuito.

El esquemático de la práctica se muestra a continuación.



6. Mejora

Modifique el programa para habilitar la interrupción externa (EXTI1) y usarla para hacer un contador ascendente y descendente.



7. Observaciones.

Esta sección es para que el alumno anote sus observaciones.

8. Conclusiones.

Esta sección es para que el alumno anote sus conclusiones.

9. Importante.

La práctica deberá ser validada en el salón de clases antes de anexar el reporte al manual de prácticas. Una vez validada realizar el reporte de práctica como se anteriormente y anexar al manual de prácticas que se entregará a final del curso.