

Interrupciones de GPIO

Interupciones de GPIO

- Características
- Registros asociados
- Handler
- Banderas

- Características

El microcontrolador LPC1769 permite configurar los pines del puerto cero (0) y puerto dos (2) de modo de generar interrupciones por GPIO, estas no son interrupciones externas, estas son interrupciones de GPIO, las cuales tienen pequeñas diferencias en su configuración.

Por lo tanto, podemos configurar los pines de estos puertos para generar una interrupción en un flanco de bajada, o un flanco de subida de al menos 1 pin.

En lo próximo solo mencionaremos los registros asociados a las interrupciones de GPIO del puerto 0, siendo que las interrupciones de GPIO del puerto 2 se tratan exactamente igual.

- Registros asociados

GPIO Interrupt Enable for port 0 Rising Edge (IO0IntEnR - 0x4002 8090).

Cada uno de estos bits son de lectura y escritura y permiten habilitar las correspondientes interrupciones de GPIO del puerto cero por flanco de subida.

9.5.6.2 GPIO Interrupt Enable for port 0 Rising Edge (IO0IntEnR - 0x4002 8090)

Each bit in these read-write registers enables the rising edge interrupt for the corresponding port 0 pin.

Table 115. GPIO Interrupt Enable for port 0 Rising Edge (IO0IntEnR - 0x4002 8090) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0ER		Enable rising edge interrupt for P0.0.	0
		0	Rising edge interrupt is disabled on P0.0.	
		1	Rising edge interrupt is enabled on P0.0.	
1	P0.1ER		Enable rising edge interrupt for P0.1.	0
2	P0.2ER		Enable rising edge interrupt for P0.2.	0
3	P0.3ER		Enable rising edge interrupt for P0.3.	0
4	P0.4ER ^[1]		Enable rising edge interrupt for P0.4.	0
5	P0.5ER ^[1]		Enable rising edge interrupt for P0.5.	0
6	P0.6ER		Enable rising edge interrupt for P0.6.	0
7	P0.7ER		Enable rising edge interrupt for P0.7.	0
8	P0.8ER		Enable rising edge interrupt for P0.8.	0
9	P0.9ER		Enable rising edge interrupt for P0.9.	0

- Registros asociados

GPIO Interrupt Enable for port 0 Falling Edge (IO0IntEnR - 0x4002 8090).

Cada uno de estos bits son de lectura y escritura y permiten habilitar las correspondientes interrupciones de GPIO del puerto cero por flanco de bajada.

[1] Not available on 80-pin package.

9.5.6.4 GPIO Interrupt Enable for port 0 Falling Edge (IO0IntEnF - 0x4002 8094)

Each bit in these read-write registers enables the falling edge interrupt for the corresponding GPIO port 0 pin.

Table 117. GPIO Interrupt Enable for port 0 Falling Edge (IO0IntEnF - address 0x4002 8094)
bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0EF		Enable falling edge interrupt for P0.0	0
		0	Falling edge interrupt is disabled on P0.0.	
		1	Falling edge interrupt is enabled on P0.0.	
1	P0.1EF		Enable falling edge interrupt for P0.1.	0
2	P0.2EF		Enable falling edge interrupt for P0.2.	0
3	P0.3EF		Enable falling edge interrupt for P0.3.	0
4	P0.4EF ^[1]		Enable falling edge interrupt for P0.4.	0
5	P0.5EF ^[1]		Enable falling edge interrupt for P0.5.	0
6	P0.6EF		Enable falling edge interrupt for P0.6.	0
7	P0.7EF		Enable falling edge interrupt for P0.7.	0
8	P0.8EF		Enable falling edge interrupt for P0.8.	0

- Registros asociados

GPIO overall Interrupt Status register (IOIntStatus - address 0x4002 8080) Este registro cuenta con 2 bits útiles, el bit 0 y el bit 2, los cuales me permiten determinar si existe la presencia de una interrupción pendiente en alguno de los pines del GPIO0 o GPIO2, respectivamente.

9.5.6.1 GPIO overall Interrupt Status register (IOIntStatus - 0x4002 8080)

This read-only register indicates the presence of interrupt pending on all of the GPIO ports that support GPIO interrupts. Only status one bit per port is required.

Table 114. GPIO overall Interrupt Status register (IOIntStatus - address 0x4002 8080) bit description

Bit	Symbol	Value	Description	Reset value
0	P0Int		Port 0 GPIO interrupt pending.	0
		0	There are no pending interrupts on Port 0.	
		1	There is at least one pending interrupt on Port 0.	
1	-	-	Reserved. The value read from a reserved bit is not defined.	NA
2	P2Int		Port 2 GPIO interrupt pending.	0
		0	There are no pending interrupts on Port 2.	
		1	There is at least one pending interrupt on Port 2.	
31:2	-	-	Reserved. The value read from a reserved bit is not defined.	NA

- Registros asociados

GPIO Interrupt Status for port 0 Rising Edge Interrupt (IO0IntStatR - 0x4002 8084) Además de saber qué puerto generó la interrupción, necesitamos saber cuál fue el pin del respectivo puerto que genera la interrupción. El presente registro de solo lectura permite determinar este aspecto para flanco de subida.

9.5.6.6 GPIO Interrupt Status for port 0 Rising Edge Interrupt (IO0IntStatR - 0x4002 8084)

Each bit in these read-only registers indicates the rising edge interrupt status for port 0.

Table 119. GPIO Interrupt Status for port 0 Rising Edge Interrupt (IO0IntStatR - 0x4002 8084) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0REI		Status of Rising Edge Interrupt for P0.0	0
		0	A rising edge has not been detected on P0.0.	
		1	Interrupt has been generated due to a rising edge on P0.0.	
1	P0.1REI		Status of Rising Edge Interrupt for P0.1.	0
2	P0.2REI		Status of Rising Edge Interrupt for P0.2.	0
3	P0.3REI		Status of Rising Edge Interrupt for P0.3.	0
4	P0.4REI ^[1]		Status of Rising Edge Interrupt for P0.4.	0
5	P0.5REI ^[1]		Status of Rising Edge Interrupt for P0.5.	0
6	P0.6REI		Status of Rising Edge Interrupt for P0.6.	0
7	P0.7REI		Status of Rising Edge Interrupt for P0.7.	0
8	P0.8REI		Status of Rising Edge Interrupt for P0.8.	0
9	P0.9REI		Status of Rising Edge Interrupt for P0.9.	0

- Registros asociados

GPIO Interrupt Status for port 0 Falling Edge Interrupt (IO0IntStatF - 0x4002 8088) Además de saber qué puerto generó la interrupción, necesitamos saber cuál fue el pin del respectivo puerto que genera la interrupción. El presente registro de solo lectura permite determinar este aspecto para flanco de bajada.

9.5.6.8 GPIO Interrupt Status for port 0 Falling Edge Interrupt (IO0IntStatF - 0x4002 8088)

Each bit in these read-only registers indicates the falling edge interrupt status for port 0.

Table 121. GPIO Interrupt Status for port 0 Falling Edge Interrupt (IO0IntStatF - 0x4002 8088) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0FEI		Status of Falling Edge Interrupt for P0.0	0
		0	A falling edge has not been detected on P0.0.	
		1	Interrupt has been generated due to a falling edge on P0.0.	
1	P0.1FEI		Status of Falling Edge Interrupt for P0.1.	0
2	P0.2FEI		Status of Falling Edge Interrupt for P0.2.	0
3	P0.3FEI		Status of Falling Edge Interrupt for P0.3.	0
4	P0.4FEI ^[1]		Status of Falling Edge Interrupt for P0.4.	0
5	P0.5FEI ^[1]		Status of Falling Edge Interrupt for P0.5.	0
6	P0.6FEI		Status of Falling Edge Interrupt for P0.6.	0
7	P0.7FEI		Status of Falling Edge Interrupt for P0.7.	0

- Registros asociados

GPIO Interrupt Clear register for port 0 (IO0IntClr - 0x4002 808C) Por último necesitamos poder bajar las banderas de las correspondientes interrupciones de GPIO0, las cuales las bajamos con el siguiente registro, escribiendo un ‘1’ en el bit del pin correspondiente. Para interrupciones de GPIO2 el tratamiento es idéntico, con sus correspondientes registros.

9.5.6.10 GPIO Interrupt Clear register for port 0 (IO0IntClr - 0x4002 808C)

Writing a 1 into a bit in this write-only register clears any interrupts for the corresponding port 0 pin.

Table 123. GPIO Interrupt Clear register for port 0 (IO0IntClr - 0x4002 808C) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0CI		Clear GPIO port Interrupts for P0.0	0
		0	Corresponding bits in IOxIntStatR and IOxIntStatF are unchanged.	
		1	Corresponding bits in IOxIntStatR and IOxStatF are cleared.	
1	P0.1CI		Clear GPIO port Interrupts for P0.1.	0
2	P0.2CI		Clear GPIO port Interrupts for P0.2.	0
3	P0.3CI		Clear GPIO port Interrupts for P0.3.	0
4	P0.4CI ^[1]		Clear GPIO port Interrupts for P0.4.	0
5	P0.5CI ^[1]		Clear GPIO port Interrupts for P0.5.	0
6	P0.6CI		Clear GPIO port Interrupts for P0.6.	0
7	P0.7CI		Clear GPIO port Interrupts for P0.7.	0

- Handler

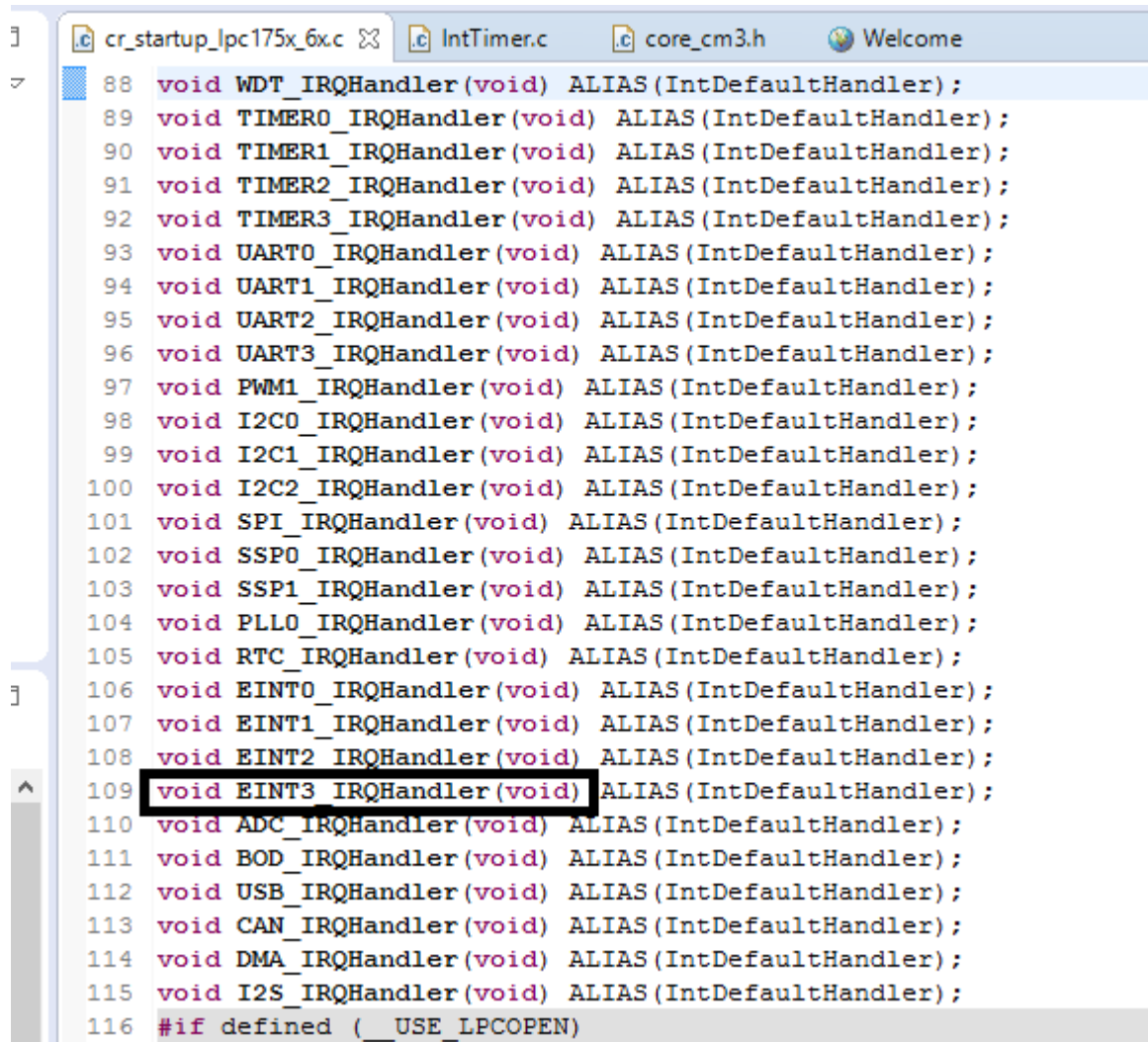
Un punto más que importante de las interrupciones de GPIO es escribir el Handler que se ejecutará cuando sobrevenga una interrupción de GPIO.

Si observamos en los Handler que están definidos en el archivo `cr_startup_lpc175x_6x.c` encontramos que no existe ningún handler asociado a las interrupciones de GPIO. El user manual aclara que las interrupciones de GPIO0 y GPIO2 comparten la misma posición en el NVIC que la interrupción externa 3 por lo tanto, tendremos que usar el handler de la interrupción externa 3

- GPIO0 and GPIO2 interrupts share the same position in the NVIC with External Interrupt 3.

- Handler

void EINT3_IRQHandler(void)



The screenshot shows an IDE with a file named `IntTimer.c` open. The code contains a series of function declarations, each representing an interrupt handler. These handlers are all defined as `void` functions that take no arguments and are aliased to `IntDefaultHandler`. The handlers are listed in descending order of priority, starting with `WDT_IRQHandler` at the top and ending with `EINT3_IRQHandler`. The `EINT3_IRQHandler` declaration on line 109 is highlighted with a black rectangular box. Below it, there are handlers for `ADC_IRQHandler`, `BOD_IRQHandler`, `USB_IRQHandler`, `CAN_IRQHandler`, `DMA_IRQHandler`, and `I2S_IRQHandler`. The file ends with a preprocessor directive `#if defined (__USE_LPCOPEN)` on line 116. The IDE's tab bar at the top shows the current file `IntTimer.c` and other open files: `cr_startup_lpc175x_6x.c`, `core_cm3.h`, and a window titled `Welcome`.

```
88 void WDT_IRQHandler(void) ALIAS(IntDefaultHandler);
89 void TIMER0_IRQHandler(void) ALIAS(IntDefaultHandler);
90 void TIMER1_IRQHandler(void) ALIAS(IntDefaultHandler);
91 void TIMER2_IRQHandler(void) ALIAS(IntDefaultHandler);
92 void TIMER3_IRQHandler(void) ALIAS(IntDefaultHandler);
93 void UART0_IRQHandler(void) ALIAS(IntDefaultHandler);
94 void UART1_IRQHandler(void) ALIAS(IntDefaultHandler);
95 void UART2_IRQHandler(void) ALIAS(IntDefaultHandler);
96 void UART3_IRQHandler(void) ALIAS(IntDefaultHandler);
97 void PWM1_IRQHandler(void) ALIAS(IntDefaultHandler);
98 void I2C0_IRQHandler(void) ALIAS(IntDefaultHandler);
99 void I2C1_IRQHandler(void) ALIAS(IntDefaultHandler);
100 void I2C2_IRQHandler(void) ALIAS(IntDefaultHandler);
101 void SPI_IRQHandler(void) ALIAS(IntDefaultHandler);
102 void SSP0_IRQHandler(void) ALIAS(IntDefaultHandler);
103 void SSP1_IRQHandler(void) ALIAS(IntDefaultHandler);
104 void PLL0_IRQHandler(void) ALIAS(IntDefaultHandler);
105 void RTC_IRQHandler(void) ALIAS(IntDefaultHandler);
106 void EINT0_IRQHandler(void) ALIAS(IntDefaultHandler);
107 void EINT1_IRQHandler(void) ALIAS(IntDefaultHandler);
108 void EINT2_IRQHandler(void) ALIAS(IntDefaultHandler);
109 void EINT3_IRQHandler(void) ALIAS(IntDefaultHandler);
110 void ADC_IRQHandler(void) ALIAS(IntDefaultHandler);
111 void BOD_IRQHandler(void) ALIAS(IntDefaultHandler);
112 void USB_IRQHandler(void) ALIAS(IntDefaultHandler);
113 void CAN_IRQHandler(void) ALIAS(IntDefaultHandler);
114 void DMA_IRQHandler(void) ALIAS(IntDefaultHandler);
115 void I2S_IRQHandler(void) ALIAS(IntDefaultHandler);
116 #if defined (__USE_LPCOPEN)
```

- Uso

Luego,

Habilitamos la interrupción externa 3 (ISERO)

Habilitamos la interrupción de GPIO0, por subida o por bajada

Escribimos el Handler (**void EINT3_IRQHandler(void)**) que se ejecutará cuando se dispare la interrupción.

Dentro del handler debemos determinar quien interrumpió (puerto y pin)

Por último antes de salir de la interrupción bajamos las banderas de las interrupciones correspondientes a los pines que dispararon interrupciones.

De más está decir que si utilizamos un único pin para generar interrupciones de GPIO no es necesario preguntar puerto y pin, sí es necesario bajar la bandera correspondiente.