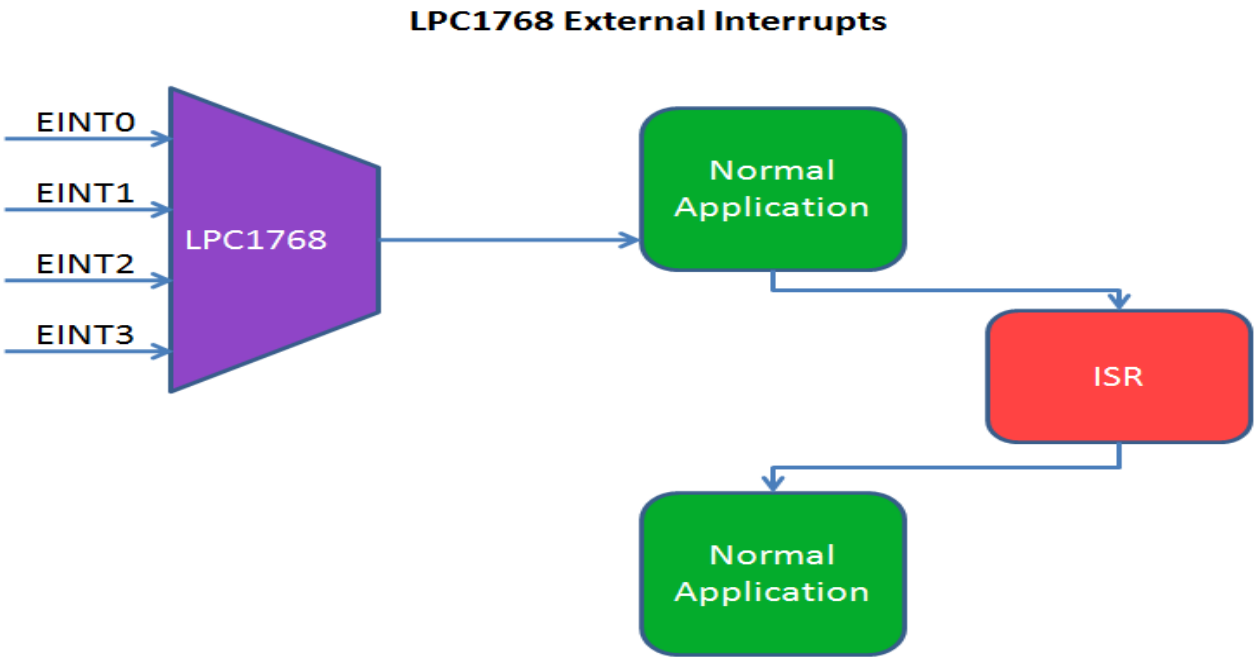
LPC1768: External Interrupts

In this tutorial we will discuss how to configure and use the LPC1768 external interrupts (EINT0-EINT3).

EINTx Pins

LPC1768 has four external interrupts EINT0-EINT3.
As LPC1768 pins are multi-functional, these four interrupts are available on multiple pins.
Below table shows mapping of EINTx pins.

Port Pin	**PINSEL_FUNC_0**	**PINSEL_FUNC_1**	**PINSEL_FUNC_2**
P2.10	GPIO	**EINT0**	NMI
P2.11	GPIO	**EINT1**	I2STX_CLK
P2_12	GPIO	**EINT2**	I2STX_WS
P2.13	GPIO	**EINT3**	I2STX_SDA



EINT Registers

Below table shows the registers associated with LPC1768 external interrupts.

Register	Description
PINSELx	To configure the pins as External Interrupts
EXTINT	External Interrupt Flag Register contains interrupt flags for EINT0,EINT1, EINT2 & EINT3.
EXTMODE	External Interrupt Mode register(Level/Edge Triggered)
EXTPOLAR	External Interrupt Polarity(Falling/Rising Edge, Active Low/High)

EXTINT				
31:4	3	2	1	0
RESERVED	EINT3	EINT2	EINT1	EINT0

EINTx: Bits will be set whenever the interrupt is detected on the particular interrupt pin. If the interrupts are enabled then the control goes to ISR. Writing one to specific bit will clear the corresponding interrupt.

EXTMODE				
31:4	3	2	1	0
RESERVED	EXTMODE3	EXTMODE2	EXTMODE1	EXTMODE0

EXTMODEx: This bits is used to select whether the EINTx pin is level or edge Triggered
0: EINTx is Level Triggered.
1: EINTx is Edge Triggered.

EXTPOLAR				
31:4	3	2	1	0
RESERVED	EXTPOLAR3	EXTPOLAR2	EXTPOLAR1	EXTPOLAR0

EXTPOLARx: This bits is used to select polarity (LOW/HIGH, FALLING/RISING) of the EINTx interrupt depending on the EXTMODE register.
0: EINTx is Active Low or Falling Edge (depending on EXTMODEx).
1: EINTx is Active High or Rising Edge (depending on EXTMODEx).

Steps to Configure Interrupts

1. Configure the pins as external interrupts in PINSELx register.
2. Clear any pending interrupts in EXTINT.
3. Configure the EINTx as Edge/Level triggered in EXTMODE register.
4. Select the polarity (Falling/Rising Edge, Active Low/High) of the interrupt in EXTPOLAR register.
5. Finally enable the interrupts by calling NVIC_EnableIRQ() with IRQ number.

Code

Below example shows the toggling of Leds depending on the external interrupts.

```
#include <lpc17xx.h>



#define PINSEL_EINT0    20
#define PINSEL_EINT1    22


#define LED1            0
#define LED2            1


#define SBIT_EINT0      0
#define SBIT_EINT1      1
```

```
#define SBIT_EXTMODE0   0
#define SBIT_EXTMODE1   1

#define SBIT_EXTPOLAR0  0
#define SBIT_EXTPOLAR1  1


void EINT0_IRQHandler(void)
{
    LPC_SC->EXTINT = (1<<SBIT_EINT0);  /* Clear Interrupt Flag */
    LPC_GPIO2->FIOPIN ^= (1<< LED1);   /* Toggle the LED1 every time INTR0 is generated */
}


void EINT1_IRQHandler(void)
{
    LPC_SC->EXTINT = (1<<SBIT_EINT1);  /* Clear Interrupt Flag */
    LPC_GPIO2->FIOPIN ^= (1<< LED2);   /* Toggle the LED2 everytime INTR1 is generated */
}


int main()
{
    SystemInit();

    LPC_SC->EXTINT      = (1<<SBIT_EINT0)    | (1<<SBIT_EINT1);     /* Clear Pending interrupts */
    LPC_PINCON->PINSEL4 = (1<<PINSEL_EINT0)  | (1<<PINSEL_EINT1);  /* Configure P2_10,P2_11 as
EINT0/1 */
    LPC_SC->EXTMODE     = (1<<SBIT_EXTMODE0) | (1<<SBIT_EXTMODE1);  /* Configure EINTx as Edge
Triggered*/
    LPC_SC->EXTPOLAR    = (1<<SBIT_EXTPOLAR0)| (1<<SBIT_EXTPOLAR0); /* Configure EINTx as Falling
Edge */
```

```
    LPC_GPIO2->FIODIR    = (1<<LED1)  | (1<<LED2);            /* Configure LED pins as OUTPUT */
    LPC_GPIO2->FIOPIN    =  0x00;



    NVIC_EnableIRQ(EINT0_IRQn);     /* Enable the EINT0,EINT1 interrupts */
    NVIC_EnableIRQ(EINT1_IRQn);



    while(1)
    {
      // Do nothing
    }
}
```

Reference:

https://www.exploreembedded.com/wiki/LPC1768:_External_Interrupts