2.7 INTERFACING KEY WITH LPC1768 USING INTERRUPTS

2.7.1 INTERFACING DIAGRAM

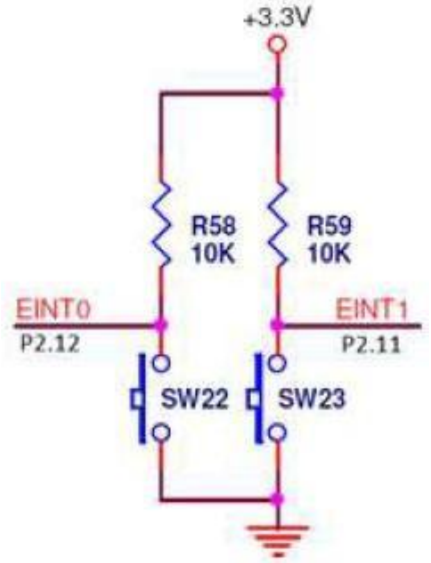


Fig.: Keys connected to P2.12($\overline{EINT0}$) **and P2.11** ($\overline{EINT1}$)

- **External Interrupt Configuration:**

  The microcontroller can configure P2.11 and P2.12 as external interrupt sources. The interrupt can be triggered on:

  - ○ Falling edge (HIGH to LOW transition when the button is pressed) ○ Rising edge (LOW to HIGH transition when the button is released).
  - ○ Level-sensitive (continuously detecting LOW level when pressed).

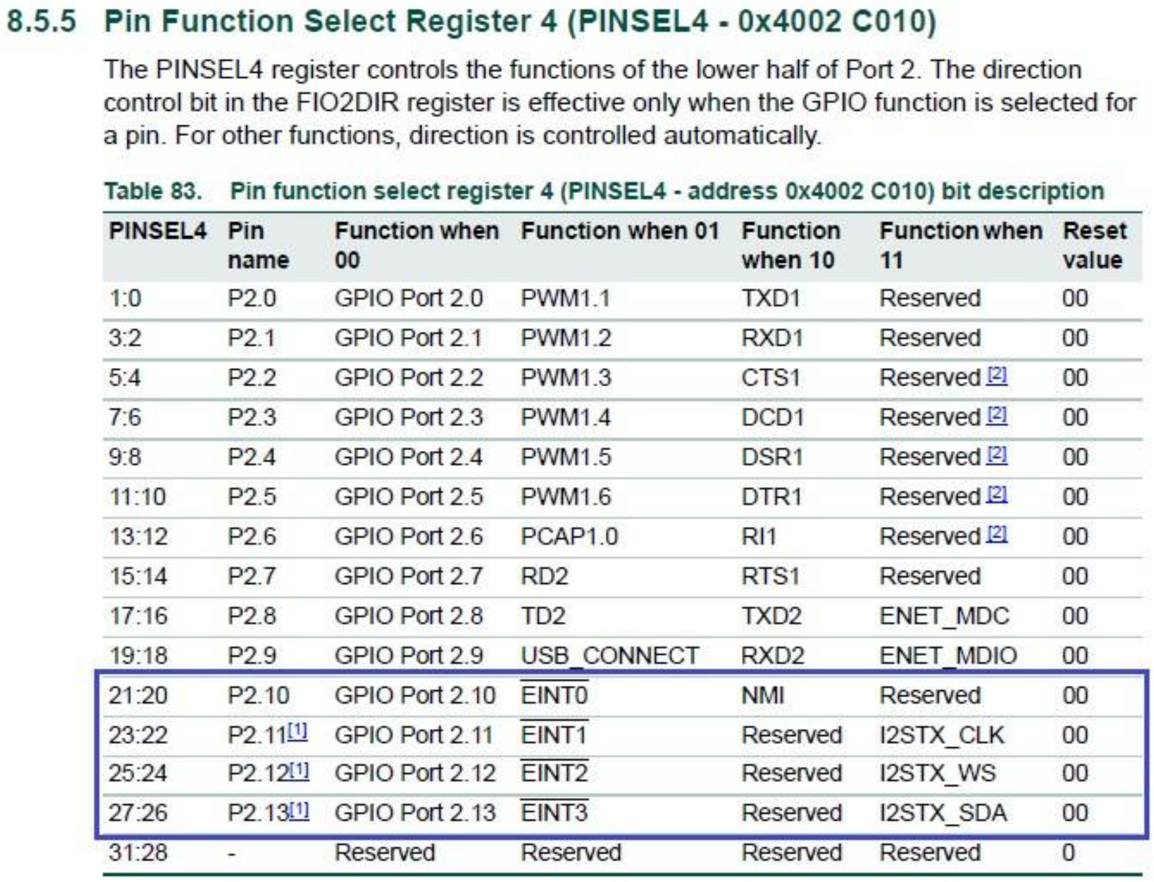
2.7.2 INTERRUPT REGISTERS CONFIGURATIONS

a) Selecting External Interrupt Function

There are four External Interrupts $\overline{\mathbf{EINT0}}$ to $\overline{\mathbf{EINT3}}$ and are available on pins P2.10 to P2.13. **The keys on board connected to EINT~~1 and EINT2~~**

$\overline{\mathbf{EINT0}}$	**P2.10**
$\overline{\mathbf{EINT1}}$	**P2.11**
$\overline{\mathbf{EINT2}}$	**P2.12**
$\overline{\mathbf{EINT3}}$	**P2.13**

The PINSEL4 register controls the functions of the lower half of Port 2. The external interrupts are alternate function 1 for the pins and can be enabled by writing 01 to corresponding bit positions in PINSEL4 register.



8.5.5 Pin Function Select Register 4 (PINSEL4 - 0x4002 C010)

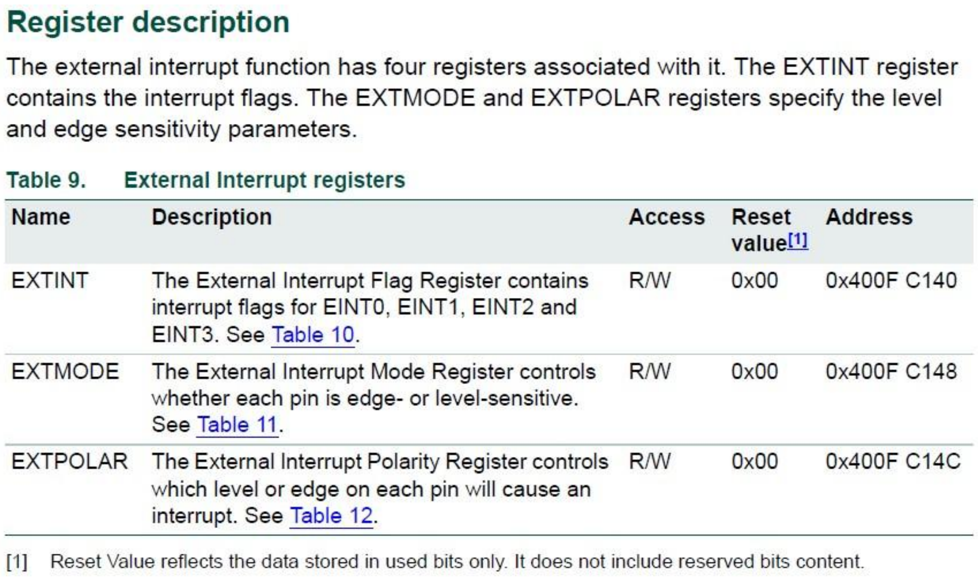
The PINSEL4 register controls the functions of the lower half of Port 2. The direction control bit in the FIO2DIR register is effective only when the GPIO function is selected for a pin. For other functions, direction is controlled automatically.

Table 83. Pin function select register 4 (PINSEL4 - address 0x4002 C010) bit description

PINSEL4	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P2.0	GPIO Port 2.0	PWM1.1	TXD1	Reserved	00
3:2	P2.1	GPIO Port 2.1	PWM1.2	RXD1	Reserved	00
5:4	P2.2	GPIO Port 2.2	PWM1.3	CTS1	Reserved [2]	00
7:6	P2.3	GPIO Port 2.3	PWM1.4	DCD1	Reserved [2]	00
9:8	P2.4	GPIO Port 2.4	PWM1.5	DSR1	Reserved [2]	00
11:10	P2.5	GPIO Port 2.5	PWM1.6	DTR1	Reserved [2]	00
13:12	P2.6	GPIO Port 2.6	PCAP1.0	RI1	Reserved [2]	00
15:14	P2.7	GPIO Port 2.7	RD2	RTS1	Reserved	00
17:16	P2.8	GPIO Port 2.8	TD2	TXD2	ENET_MDC	00
19:18	P2.9	GPIO Port 2.9	USB_CONNECT	RXD2	ENET_MDIO	00
21:20	P2.10	GPIO Port 2.10	$\overline{\text{EINT0}}$	NMI	Reserved	00
23:22	P2.11[1]	GPIO Port 2.11	$\overline{\text{EINT1}}$	Reserved	I2STX_CLK	00
25:24	P2.12[1]	GPIO Port 2.12	$\overline{\text{EINT2}}$	Reserved	I2STX_WS	00
27:26	P2.13[1]	GPIO Port 2.13	$\overline{\text{EINT3}}$	Reserved	I2STX_SDA	00
31:28	-	Reserved	Reserved	Reserved	Reserved	0

PINSEL4	Pin Name	Function when 00	Function when 01	Function when 10	Function when 11	Reset Value
23:22	P2.11	GPIO Port 2.11	$\overline{\text{EINT1}}$	Reserved	I2STX_CLK	00

```
LPC_PINCON->PINSEL4 |= (1 << 22);
LPC_PINCON->PINSEL4 &= ~(1 << 23);
```



Register description

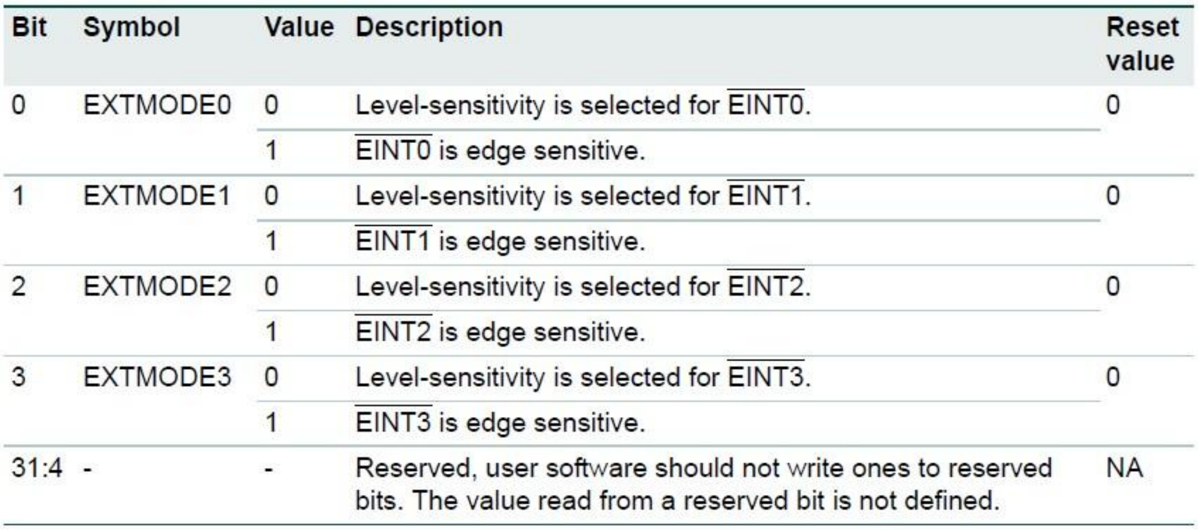
The external interrupt function has four registers associated with it. The EXTINT register contains the interrupt flags. The EXTMODE and EXTPOLAR registers specify the level and edge sensitivity parameters.

Table 9. External Interrupt registers

Name	Description	Access	Reset value[1]	Address
EXTINT	The External Interrupt Flag Register contains interrupt flags for EINT0, EINT1, EINT2 and EINT3. See Table 10.	R/W	0x00	0x400F C140
EXTMODE	The External Interrupt Mode Register controls whether each pin is edge- or level-sensitive. See Table 11.	R/W	0x00	0x400F C148
EXTPOLAR	The External Interrupt Polarity Register controls which level or edge on each pin will cause an interrupt. See Table 12.	R/W	0x00	0x400F C14C

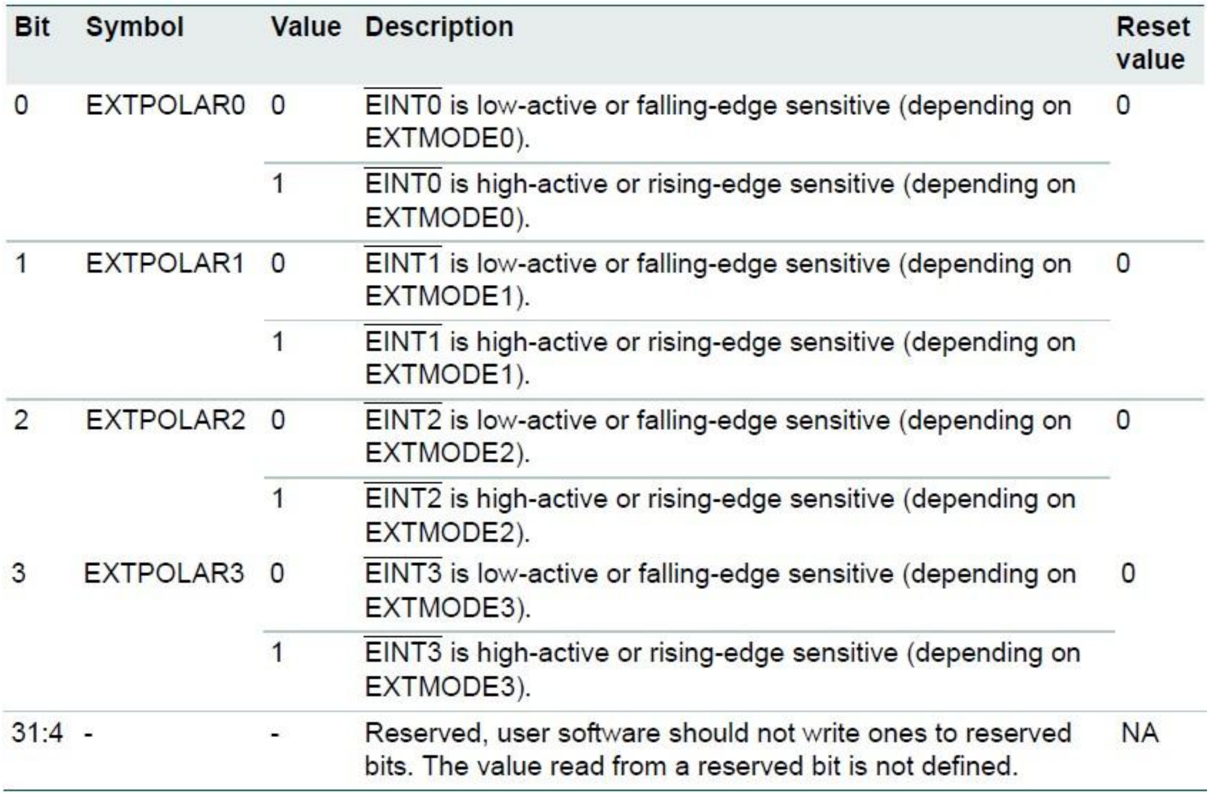
[1] Reset Value reflects the data stored in used bits only. It does not include reserved bits content.

External Interrupt Mode register (EXTMODE - 0x400F C148)



Bit	Symbol	Value	Description	Reset value
0	EXTMODE0	0	Level-sensitivity is selected for $\overline{\text{EINT0}}$.	0
		1	$\overline{\text{EINT0}}$ is edge sensitive.	
1	EXTMODE1	0	Level-sensitivity is selected for $\overline{\text{EINT1}}$.	0
		1	$\overline{\text{EINT1}}$ is edge sensitive.	
2	EXTMODE2	0	Level-sensitivity is selected for $\overline{\text{EINT2}}$.	0
		1	$\overline{\text{EINT2}}$ is edge sensitive.	
3	EXTMODE3	0	Level-sensitivity is selected for $\overline{\text{EINT3}}$.	0
		1	$\overline{\text{EINT3}}$ is edge sensitive.	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

External Interrupt Polarity register

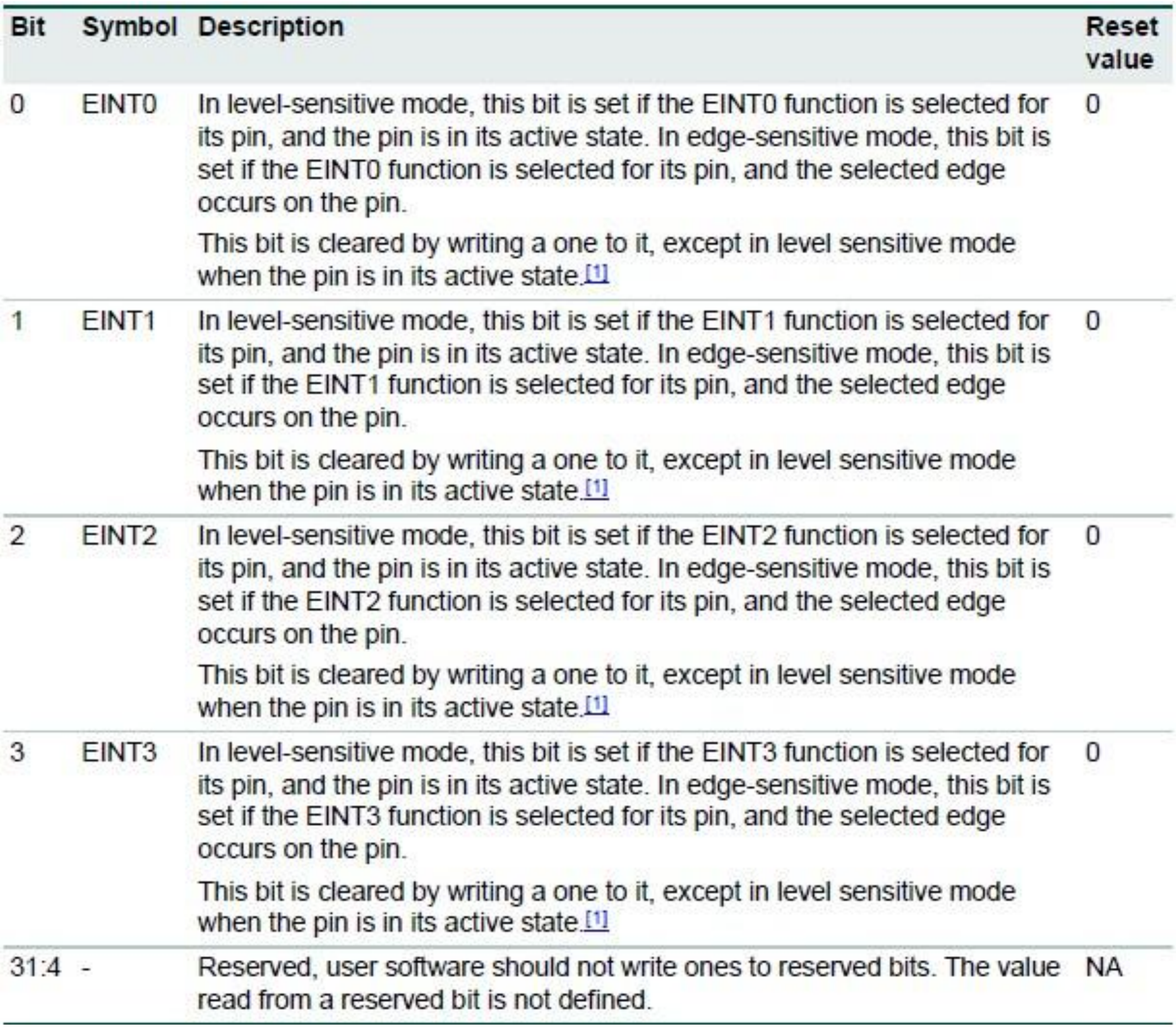


Bit	Symbol	Value	Description	Reset value
0	EXTPOLAR0	0	$\overline{\text{EINT0}}$ is low-active or falling-edge sensitive (depending on EXTMODE0).	0
		1	$\overline{\text{EINT0}}$ is high-active or rising-edge sensitive (depending on EXTMODE0).	
1	EXTPOLAR1	0	$\overline{\text{EINT1}}$ is low-active or falling-edge sensitive (depending on EXTMODE1).	0
		1	$\overline{\text{EINT1}}$ is high-active or rising-edge sensitive (depending on EXTMODE1).	
2	EXTPOLAR2	0	$\overline{\text{EINT2}}$ is low-active or falling-edge sensitive (depending on EXTMODE2).	0
		1	$\overline{\text{EINT2}}$ is high-active or rising-edge sensitive (depending on EXTMODE2).	
3	EXTPOLAR3	0	$\overline{\text{EINT3}}$ is low-active or falling-edge sensitive (depending on EXTMODE3).	0
		1	$\overline{\text{EINT3}}$ is high-active or rising-edge sensitive (depending on EXTMODE3).	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

External Interrupt flag register (EXTINT - 0x400F C140)

When a pin is selected for its external interrupt function, the level or edge on that pin (selected by its bits in the EXTPOLAR and EXTMODE registers) will set its interrupt flag in this register. This asserts the corresponding interrupt request to the NVIC, which will cause an interrupt if interrupts from the pin are enabled.

Writing ones to bits EINT0 through EINT3 in EXTINT register clears the corresponding bits. In level-sensitive mode the interrupt is cleared only when the pin is in its inactive state. Once a bit from EINT0 to EINT3 is set and an appropriate code starts to execute (handling wake-up and/or external interrupt), this bit in EXTINT register must be cleared. Otherwise event that was just triggered by activity on the EINT pin will not be recognized in future.



Bit	Symbol	Description	Reset value
0	EINT0	In level-sensitive mode, this bit is set if the EINT0 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT0 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0
1	EINT1	In level-sensitive mode, this bit is set if the EINT1 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT1 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0
2	EINT2	In level-sensitive mode, this bit is set if the EINT2 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT2 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0
3	EINT3	In level-sensitive mode, this bit is set if the EINT3 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT3 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0
31:4	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

2.7.3 EXAMPLE CODE

Write a C program for the LPC17xx microcontroller that reads the status of a switch connected to an interrupt. When the switch is pressed, turn on an LED connected to P1.19. Additionally, implement a blinking LED on P1.26 that toggles its state at regular intervals.

```
#include <lpc17xx.h>

#define LED          19        // LED controlled by key press
#define BLINKLED     26        // LED blinking continuously
#define EX_INT       1         // External Interrupt 1
#define KEY_PIN      11        // Switch connected to P2.11
 void led_init(void); void
interrupt_init(void); void
EINT1_IRQHandler(void);
```

```
int main()
{
     // Variables for non-blocking LED blink
volatile unsigned int blinkcounter = 0;

     led_init();
     interrupt_init();

     while (1)
     {
          //Turn OFF LED when key is released
     if (LPC_GPIO2->FIOPIN & (1 << KEY_PIN))
                 LPC_GPIO1->FIOCLR = (1 << LED);

          // Non-blocking LED blinking
     blinkcounter++;          if
(blinkcounter >= 1000000)
          {
               LPC_GPIO1->FIOPIN ^= (1 << BLINKLED);
     blinkcounter = 0;
          }
     }
}

void led_init(void)
{
     // Configure LED pins as output
     LPC_GPIO1->FIODIR |= (1 << LED) | (1 << BLINKLED);

     // Configure P2.11 as input for key detection
     LPC_GPIO2->FIODIR &= ~(1 << KEY_PIN);
}  void
interrupt_init(void)
{
     // Clear pending interrupts
     LPC_SC->EXTINT = (1 << EX_INT);

     // Configure P2.11 as EINT1
     LPC_PINCON->PINSEL4 |= (1 << 22);
     LPC_PINCON->PINSEL4 &= ~(1 << 23);
     // Configure EINT1 as Edge Triggered and Falling Edge
     LPC_SC->EXTMODE |= (1 << EX_INT);
     LPC_SC->EXTPOLAR &= ~(1 << EX_INT);

     // Enable the EINT1
     NVIC_EnableIRQ(EINT1_IRQn);
}

//Interrupt Handler void
EINT1_IRQHandler(void)
{
     // Clear interrupt flag
     LPC_SC->EXTINT = (1 << EX_INT);

     //Switch ON LED
     LPC_GPIO1->FIOSET = (1 << LED);
}
```

Further exploration:

Write an application to make use of both the interrupts?