INT Timer Function

(1) Power up timer > (2) Clear all interrupts

(3) Set prescaler > (4) set match register and match control register > (5) Disable timer as default > (6) Enable timer IRS

(7) Set GP2.10 as INT0 > (8)Enable falling edge INT > (9) Set edge trigger EXTMODE > (10) Enable IRS

```
int main(void)
{
/* Power up timer0 */
LPC_SC->PCONP |= BIT(1);
/* Clear all interrupts */
LPC_TIM0->IR = 0x3F;
/* Set prescaler based on sysclk to get 1000us minimum interval
*/
switch (PCLKSEL_TIMER0(LPC_SC->PCLKSEL0)) {
case 0x00:
LPC_TIM0->PR = SystemCoreClock / 4000;
break;
case 0x01:
LPC_TIM0->PR = SystemCoreClock / 1000;
break;
case 0x02:
LPC_TIM0->PR = SystemCoreClock / 2000;
break;
case 0x03:
default:
LPC_TIM0->PR = SystemCoreClock / 8000;
break;
}
/* Set the interval at Match Control 0 */
```

Fig. 1

```
/* Set the interval at Match Control 0 */
LPC_TIM0->MR0 = TIMER_INTERVAL;
/* Enable Match Control 0 */
LPC_TIM0->MCR = BIT(0) | BIT(1);
/* Disable Timer as default */
LPC_TIM0->TCR = 0;
/* Enable Timer IRQ */
NVIC_EnableIRQ(TIMER0_IRQn);

/* Set P2.10 as EINT0 function */
LPC_PINCON->PINSEL4 &= ~GEN_MASK(2, 20);
LPC_PINCON->PINSEL4 |= BIT(20);
/* Enable falling edge trigger for P2.10 */
LPC_GPIOINT->IO2IntEnF |= BIT(10);
/* Edge trigger for EINT0 */
LPC_SC->EXTMODE |= BIT(0);
/* Enable EINT IRQ */
NVIC_EnableIRQ(EINT0_IRQn);

printf("Main Function: Waiting for interrupt\n");

    while(1);

    return 0;
}
```

Fig. 2

Power Up Timer

(1) Special Purpose Registers: PCONP power up timer (LPC_SC -> PCONP), table 46 (datasheet), bit 1 for TIM0

table 46

Bit	Symbol	Description	Reset value
0	-	Reserved.	NA
1	PCTIM0	Timer/Counter 0 power/clock control b	1
2	PCTIM1	Timer/Counter 1 power/clock control b	1

```
int main(void)
{
/* Power up timer0 */
LPC_SC->PCONP |= BIT(1);
/* Clear all interrupts */
LPC_TIM0->IR = 0x3F;
/* Set prescaler sysclk, get 1000us interval */
switch (PCLKSEL_TIMER0(LPC_SC->PCLKSEL0)) {
case 0x00:
LPC_TIM0->PR = SystemCoreClock / 4000;
break;
}
/* Set the interval at Match Control 0 */
```

Fig. 2

Clear Timer INT

Timer hardware has match register MR (chapter 21), its values compared to the Timer Counter value. When they are equal, interrupt can be generated, then reset Timer Counter, or stop the timer, which are controlled by MCR register.

```
int main(void)
{
/* Power up timer0 */
LPC_SC->PCONP |= BIT(1);
/* Clear all interrupts */
LPC_TIM0->IR = 0x3F;
```

Table 426. TIMER/COUNTER0-3 register map

Generic Name	Description	Access
IR	Interrupt Register. The IR can be written to clear interrupts. The IR R/W can be read to identify which of eight possible interrupt sources are pending.	

Table 427

Table 427. Interrupt Register (T[0/1/2/3]IR - addresses description

Bit	Symbol	Description
0	MR0 Interrupt	Interrupt flag for match channel 0.
1	MR1 Interrupt	Interrupt flag for match channel 1.
2	MR2 Interrupt	Interrupt flag for match channel 2.
3	MR3 Interrupt	Interrupt flag for match channel 3.

If interrupt generated then the corresponding bit in IR will be high. Otherwise, the bit will be low. Writing a logic one to the corresponding IR bit will reset the interrupt.

Set Prescaler

Special purpose register PCLKSEL0 select Timer0, then set PR (prescale register).

Table 427

PR Prescale Register. When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.

Note: TC (timer counter register) see table 426, PC (prescale counter register)

```
int main(void)
{
/* Power up timer0 */
LPC_SC->PCONP |= BIT(1);
/* Clear all interrupts */
LPC_TIM0->IR = 0x3F;
/* Set prescaler sysclk, get 1000us interval */
switch (PCLKSEL_TIMER0(LPC_SC->PCLKSEL0)) {
case 0x00:
LPC_TIM0->PR = SystemCoreClock / 4000;
break;
}
/* Set the interval at Match Control 0 */
```

Set Match Register MR and MCR

Special purpose register MR: match register MR (chapter 21), its values compared to the Timer Counter value, when equal, interrupt can be generated, then reset Timer Counter, or stop the timer, which are controlled by MCR register . The Match Control Register MCR controls operations performed when one of the Match Registers matches the Timer Counter.

Table 430

Table 430. Match Control Register (T[0/1/2/3]MCR - addresses 0x4000 4014, 0x4000 8014, 0x4009 0014, 0x40 bit description

Bit	Symbol	Value	Description
0	MR0I	1	Interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC.
		0	This interrupt is disabled
1	MR0R	1	Reset on MR0: the TC will be reset if MR0 matches it.
		0	Feature disabled.
2	MR0S	1	Stop on MR0: the TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC.
		0	Feature disabled.

```
/* Set the interval at Match
Control 0 */
LPC_TIM0->MR0 = TIMER_INTERVAL;
/* Enable Match Control 0 */
LPC_TIM0->MCR = BIT(0) | BIT(1);
/* Disable Timer as default */
```

Timer Architecture

Special purpose registers for basic configuration, *datasheet Chapter 21,*

(1) Power: In the PCONP (table 46);

(2) Peripherial clock: PCLKSEL0 (table 40);

(3) Interrupts: T0/1/2/3MCR match control reg.;

table 40

Bit	Symbol	Description	Reset value
1:0	PCLK_WDT	Peripheral clock selection for WDT.	00
3:2	PCLK_TIMER0	Peripheral clock selection for TIMER0.	00
5:4	PCLK_TIMER1	Peripheral clock selection for TIMER1.	00

Fig. 2