|  |
| --- |
| **EMBEDDED SYSTEM LABORATORY** |
| **LAB 2** |

**USING TIMERS FOR MULTI-TASK PROGRAMMING FOR ARM MICROCONTROLLER**

### I. LAB OBJECTIVES

### - In this Lab students will learn about ARM-CORTEX M3 (LPC1768) Microcontroller.

### - This Lab experiments are intended to implement basic Timer of ARM-CORTEX M3 Microcotroller to pheriperal devices in MB1700 Kit and write C code programming to control these devices.

### II. PRE-LAB : Timer Register Review

### 

### A screenshot of a match Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

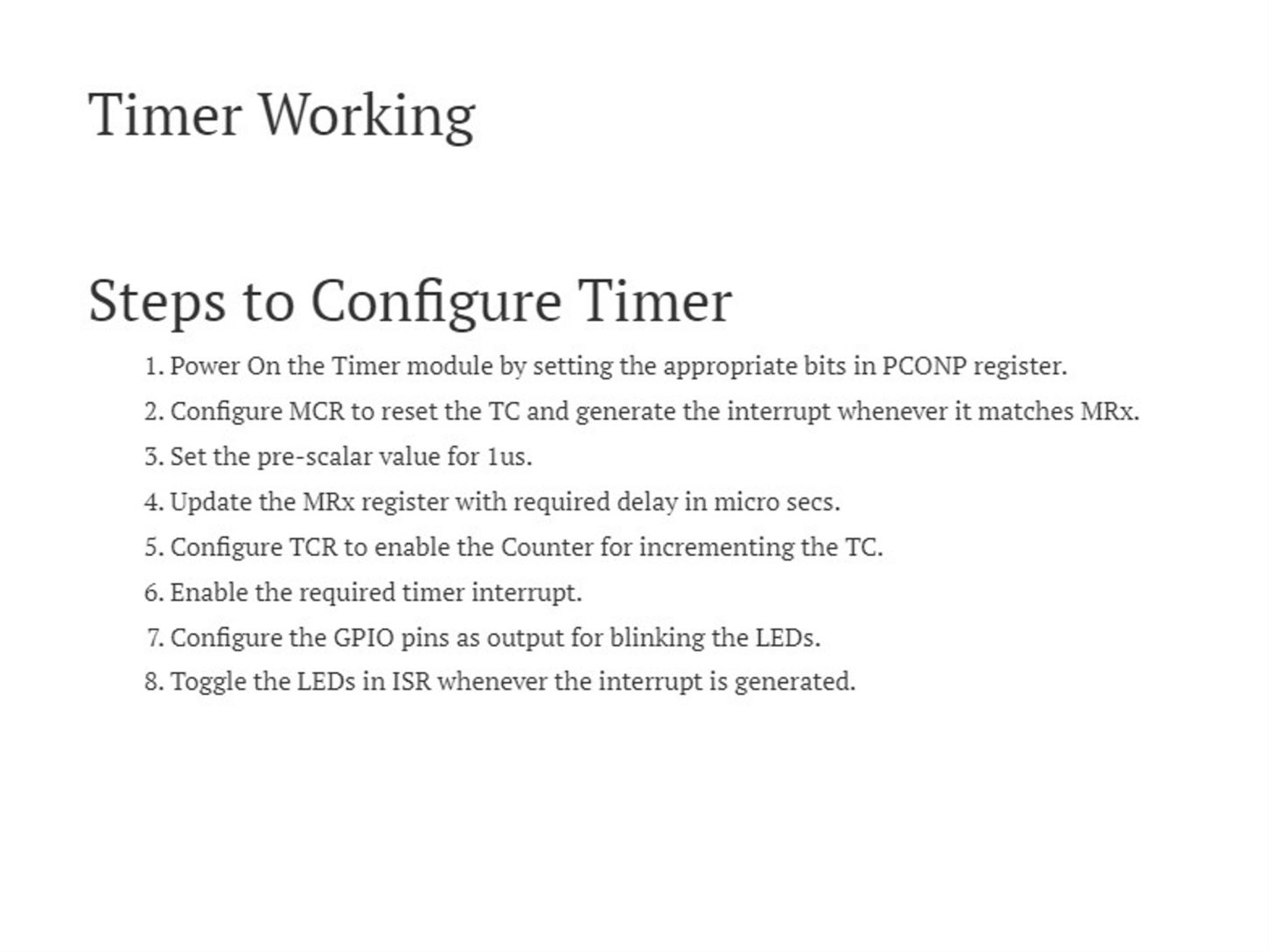
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



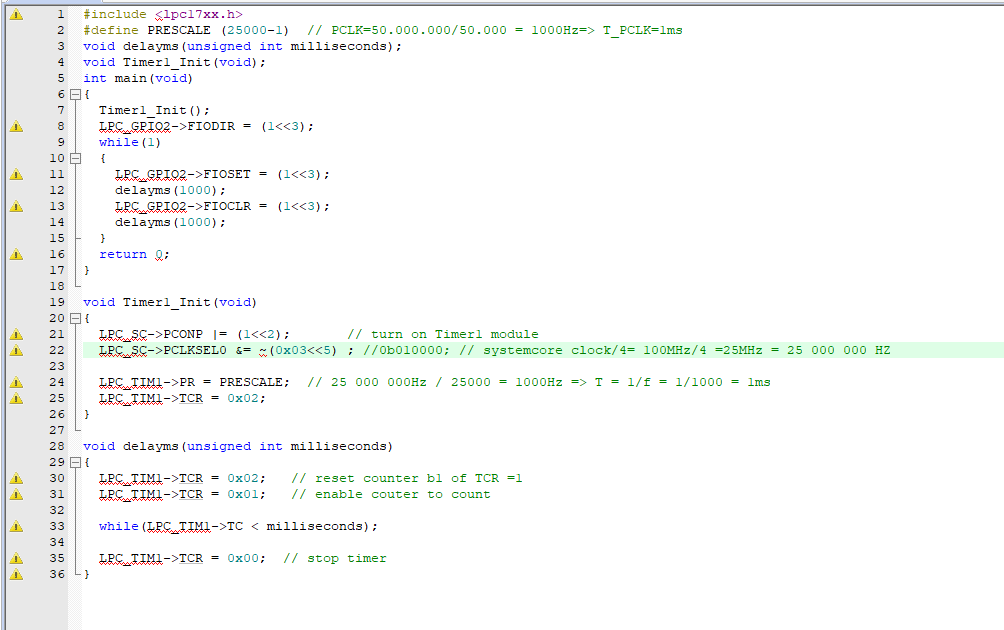
### III. LAB PROCERUCE

### The LPC1768 Microconttroler KIT using 100MHz system clock.

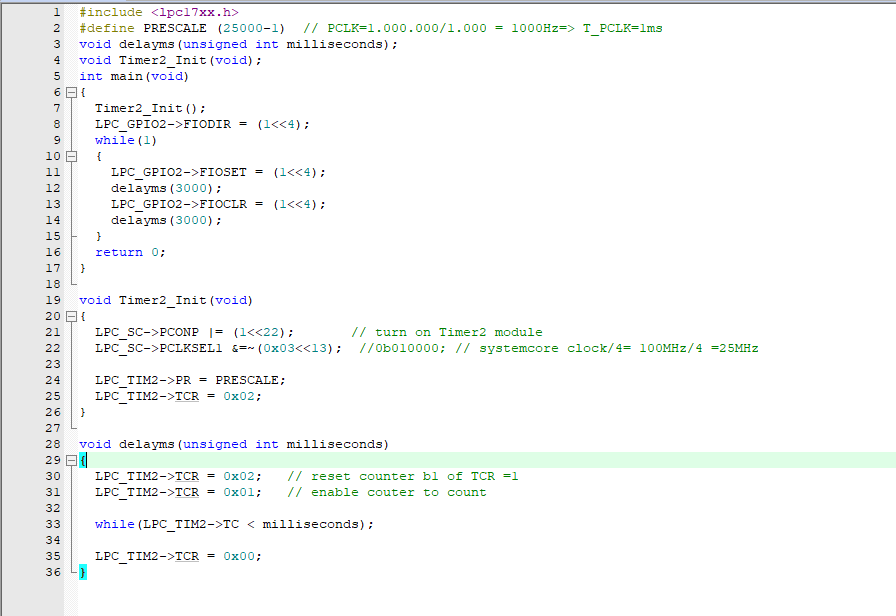
### III.1 Lab Experiment 1 : Write the code to turn on and turn off a led which is connected to P2.2 GPIO port pin with the time delay 1 second.using Timer0 polling method. Using PCLK=System Clock/2 mode, with period 1ms.

### 

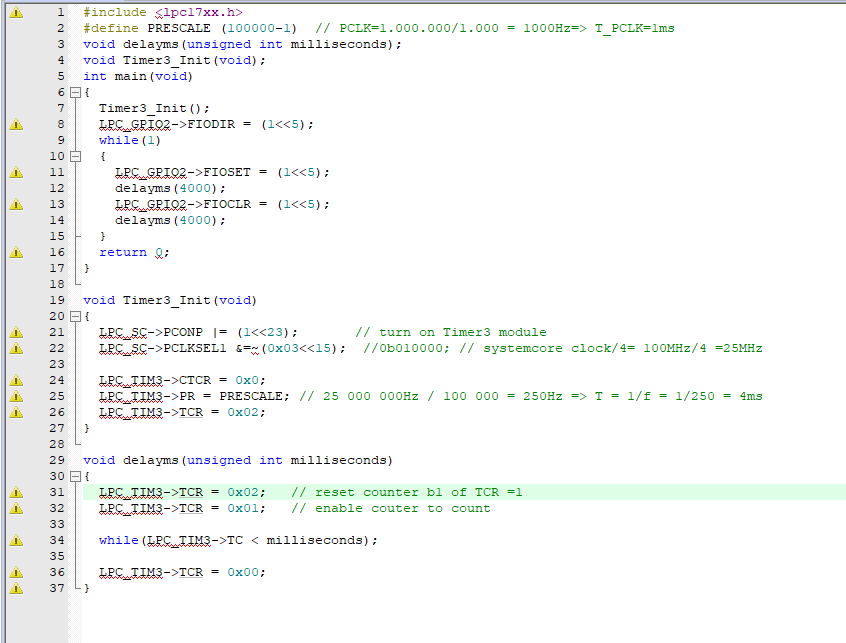
### III.2 Lab Experiment 2 : Write the code to turn on and turn off a led which is connected to P2.3 GPIO port pin with the time delay 2 second.using Timer1 polling method. Using PCLK=System Clock/4 mode with period 1ms.



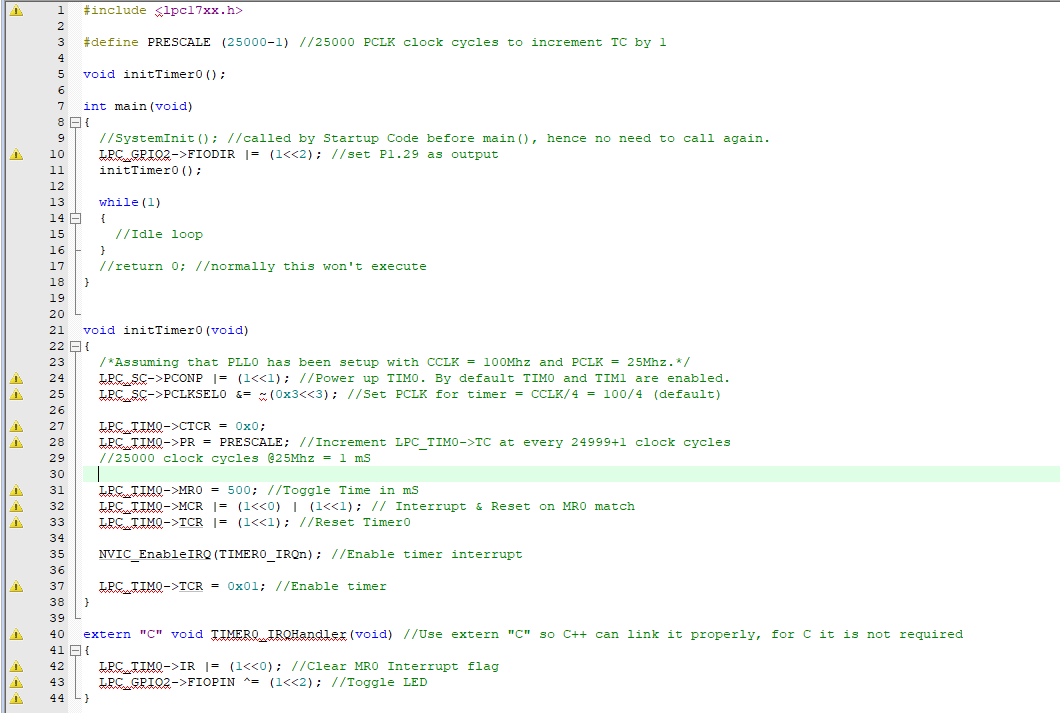
### III.3 Lab Experiment 3: Write the code to turn on and turn off a led which is connected to P2.4 GPIO port pin with the time delay 3 second.using Timer2 polling method. . Using PCLK=System Clock/8 mode with period 1ms.



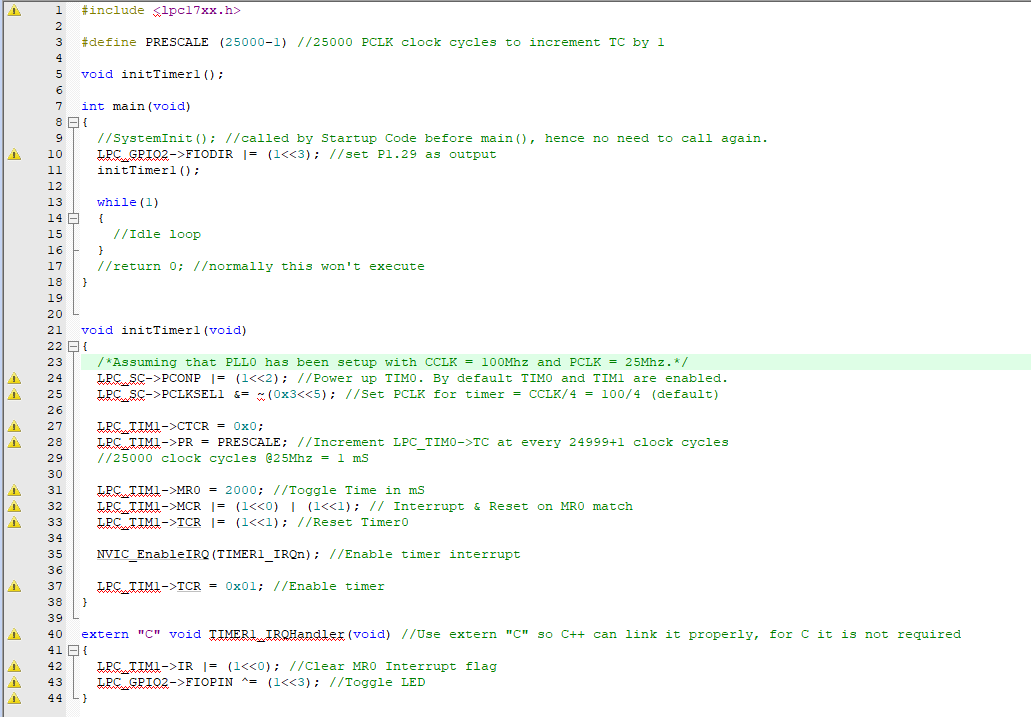
### III.4 Lab Experiment 4: Write the code to turn on and turn off a led which is connected to P2.5 GPIO port pin with the time delay 4 second.using Timer3 polling method. . Using PCLK=System Clock/4 mode with period 4ms.



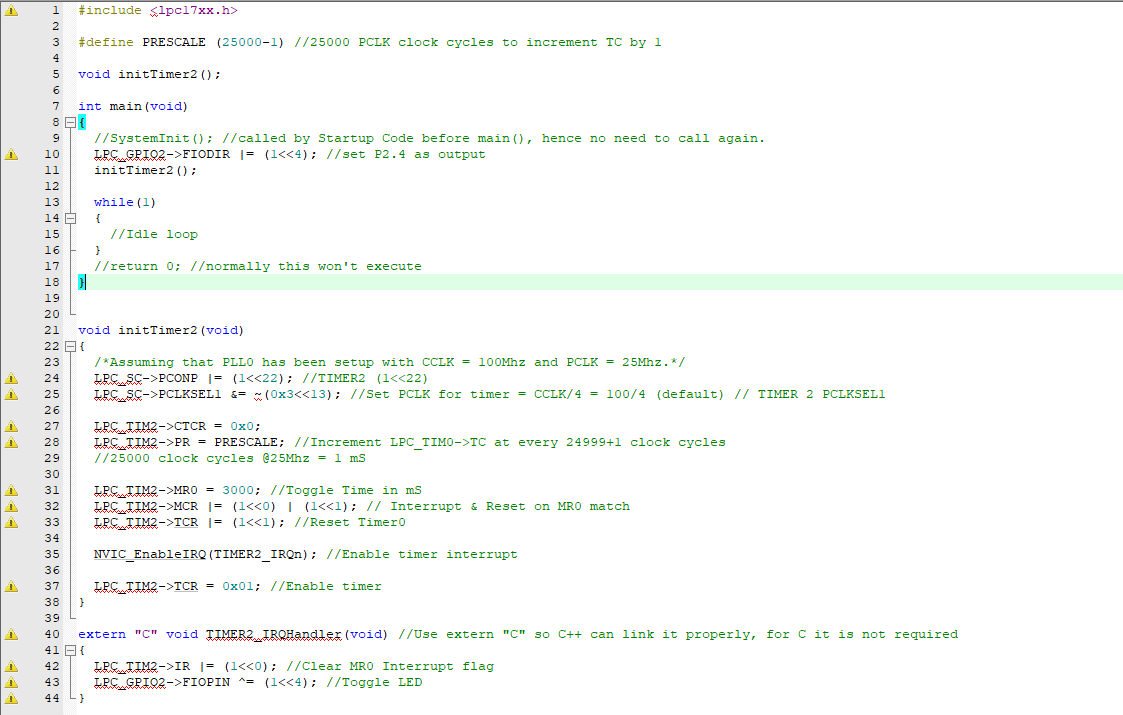
### III.5 Lab Experiment 5 : Write the code to turn on and turn off a led which is connected to P2.2 GPIO port pin with the time delay 1 second.using Timer0 Interupt method. Using PCLK=System Clock/2 mode, with period 1ms.



### III.6 Lab Experiment 6 : Write the code to turn on and turn off a led which is connected to P2.3 GPIO port pin with the time delay 2 second.using Timer1 Interupt method. Using PCLK=System Clock/4 mode with period 1ms.



### III.7 Lab Experiment 7: Write the code to turn on and turn off a led which is connected to P2.4 GPIO port pin with the time delay 3 second.using Timer2 Interupt method. . Using PCLK=System Clock/8 mode with period 1ms.

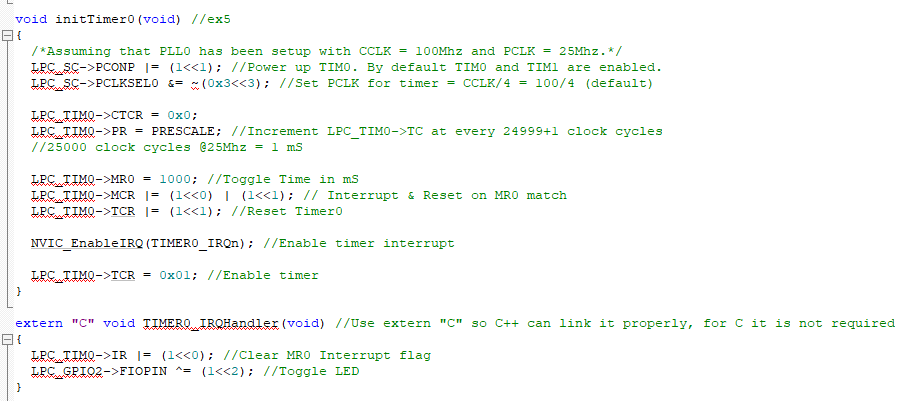


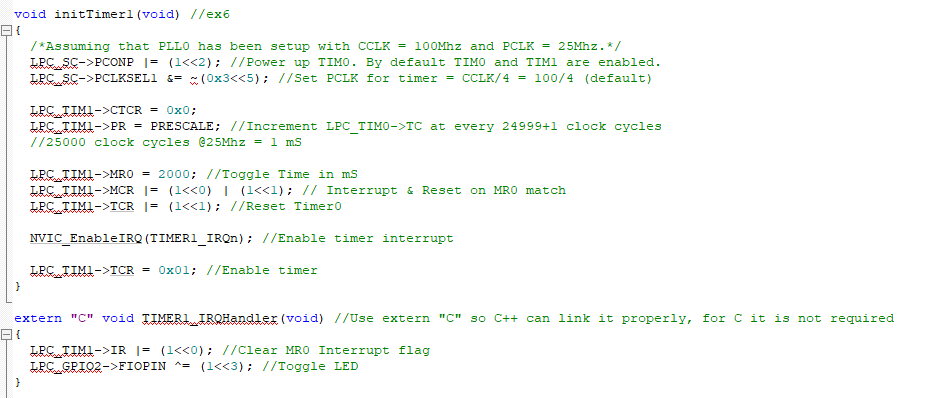
### III.8 Lab Experiment 8: Write the code to turn on and turn off a led which is connected to P2.5 GPIO port pin with the time delay 4 second.using Timer3 Interupt method. . Using PCLK=System Clock/4 mode with period 4ms.

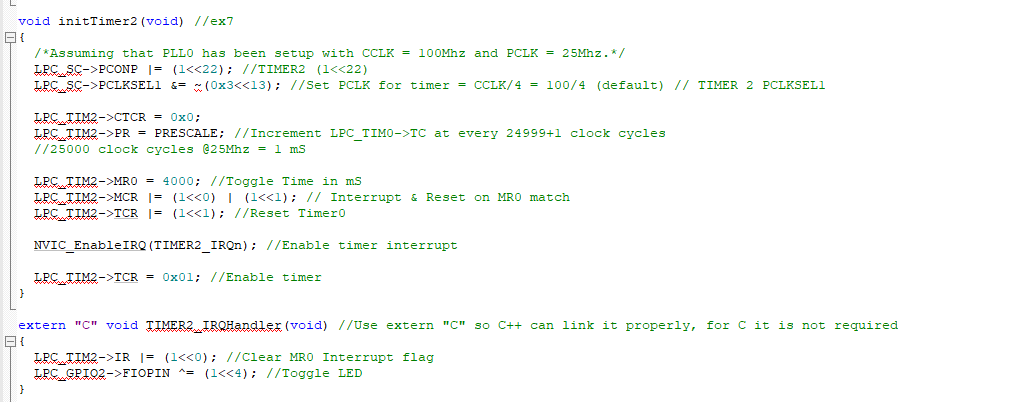
### 

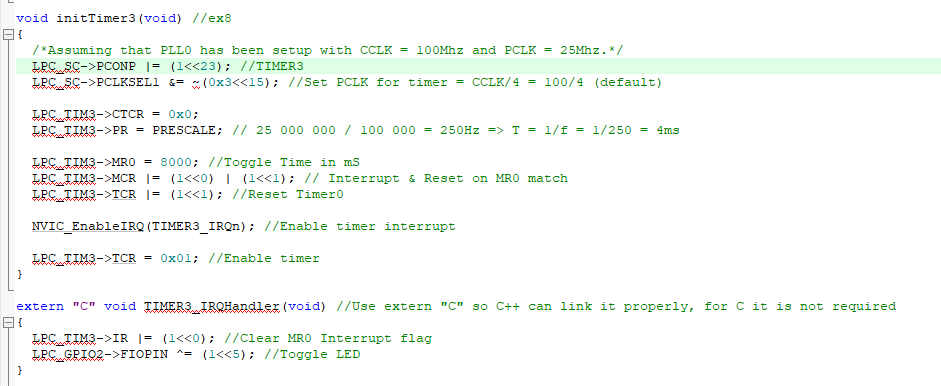
### III.9 Lab Experiment 9: Write the code to control 4 leds which is connected to P2.2, P2.3, P2.4 P2.5 GPIO port pins using Timer Interrupt methods. Led P2.2 will turn on-off with 1 second interval using Timer0, Led P2.3 will turn on-off with 2 second interval using Timer0, Led P2.3 will turn on-off with 4 second interval using Timer0, Led P2.4 will turn on-off with 8 second interval using Timer0.

### 









### III.10 Lab Experiment 10: Write a C program to control 8 leds in the KIT with 4 Led lighting modes (Mode 1: 8 lights gradually turn on, Mode 2: 8 lights gradually turn on, Mode 3: 8 chasing lights from left to right, Mode 4: 8 chasing lights from right to left) using Timer0 Interrupt.

#include <lpc17xx.h>

#define PRESCALE (25000-1) //25000 PCLK clock cycles to increment TC by 1

void initTimer0();

unsigned int led\_index=0,mode=0;

int main(void)

{

//SystemInit(); //called by Startup Code before main(), hence no need to call again.

LPC\_GPIO1->FIODIR |= (1<<28)| (1<<29)|(1<<31);

LPC\_GPIO2->FIODIR |= (1<<2)|(1<<3)|(1<<4) |(1<<5)|(1<<6);

initTimer0();

while(1)

{

//Idle loop

}

//return 0; //normally this won't execute

}

void initTimer0(void)

{

/\*Assuming that PLL0 has been setup with CCLK = 100Mhz and PCLK = 25Mhz.\*/

LPC\_SC->PCONP |= (1<<1); //Power up TIM0. By default TIM0 and TIM1 are enabled.

LPC\_SC->PCLKSEL0 &= ~(0x3<<3); //Set PCLK for timer = CCLK/4 = 100/4 (default)

LPC\_TIM0->CTCR = 0x0;

LPC\_TIM0->PR = PRESCALE; //Increment LPC\_TIM0->TC at every 24999+1 clock cycles

//25000 clock cycles @25Mhz = 1 mS

LPC\_TIM0->MR0 = 1000; //Toggle Time in mS

LPC\_TIM0->MCR |= (1<<0) | (1<<1); // Interrupt & Reset on MR0 match

LPC\_TIM0->TCR |= (1<<1); //Reset Timer0

NVIC\_EnableIRQ(TIMER0\_IRQn); //Enable timer interrupt

LPC\_TIM0->TCR = 0x01; //Enable timer

}

extern "C" void TIMER0\_IRQHandler(void) //Use extern "C" so C++ can link it properly, for C it is not required

{

LPC\_TIM0->IR |= (1<<0); //Clear MR0 Interrupt flag

if(mode==0)

{

switch(led\_index)

{

case 0: LPC\_GPIO1->FIOPIN |= (1<<28); //TURN ON LED0

break;

case 1: LPC\_GPIO1->FIOPIN |= (1<<29); //TURN ON LED0

break;

case 2: LPC\_GPIO1->FIOPIN |= (1<<31); //TURN ON LED0

break;

case 3: LPC\_GPIO2->FIOPIN |= (1<<2); //TURN ON LED0

break;

case 4: LPC\_GPIO2->FIOPIN |= (1<<3); //TURN ON LED0

break;

case 5: LPC\_GPIO2->FIOPIN |= (1<<4); //TURN ON LED0

break;

case 6: LPC\_GPIO2->FIOPIN |= (1<<5); //TURN ON LED0

break;

case 7: LPC\_GPIO2->FIOPIN |= (1<<6); //TURN ON LED0

break;

}

}

if(mode==1)

{

switch(led\_index)

{

case 0: LPC\_GPIO1->FIOPIN &=~ (1<<28); //TURN OFF LED0

break;

case 1: LPC\_GPIO1->FIOPIN &=~ (1<<29); //TURN OFF LED0

break;

case 2: LPC\_GPIO1->FIOPIN &=~ (1<<31); //TURN OFF LED0

break;

case 3: LPC\_GPIO2->FIOPIN &=~ (1<<2); //TURN OFF LED0

break;

case 4: LPC\_GPIO2->FIOPIN &=~ (1<<3); //TURN OFF LED0

break;

case 5: LPC\_GPIO2->FIOPIN &=~ (1<<4); //TURN OFF LED0

break;

case 6: LPC\_GPIO2->FIOPIN &=~ (1<<5); //TURN OFF LED0

break;

case 7: LPC\_GPIO2->FIOPIN &=~ (1<<6); //TURN OFF LED0

break;

}

}

led\_index++;

if(led\_index>8)

{

led\_index=0;

mode++;

if(mode>3)

mode=0;

}

}

### III.11 Lab Experiment 11: Write a program to interface LPC1768 GPIO port pin using Keil C to scan P1.23 pin, P2.4, P25 and P1.26 joystick buttons and change the Led controlling mode in Lab experiment 7 with an appropritate mode. The initial default LED controlling mode is Mode 1.

**IV. LAB PERFORMANCE GRADING AND LAB REPORT GUIDELINES**

For each Lab experiment Students show the successful running results to Lab Instructor for Lab Performance grading.

Students write a report which includes : Algorithm flowchart and C++ Code for each experiment. In each block of the code or line of code, give the comments for the meaning of this block of code.