Note : This material is to understand the concepts. Advised to correct syntax/logical error, if any.

1. **Program for external interrupt. ON and OFF LED connected to P1.23 whenever negative edge is applied at P2.12 (EINT2)**

```
#include<LPC17xx.h>

 void EINT2_IRQHandler(void);

 int main(void)

 {

        SystemInit();

        SystemCoreClockUpdate();

        LPC_PINCON->PINSEL4 |= (1<<24);       //P2.12 as EINT2 i.e FUNCTION-01

        LPC_PINCON->PINSEL3 &= ~(3<<14);      //P1.23 GPIO for LED ie Function-00)

        LPC_GPIO1->FIODIR = 0x00800000;                //P1.23 is assigned output

        LPC_GPIO1->FIOSET = 0x00800000;       //Initiall LED is kept on

        LPC_GPIO1->FIOMASK= ~(1<<23);

        LPC_SC->EXTINT = 0x00000004;          //writing 1 cleares the interrupt, get set if there is
interrupt

        LPC_SC->EXTMODE = 0x00000004;                  //EINT2 is initiated as edge senitive, 0 for level
sensitive

        LPC_SC->EXTPOLAR = 0x00000000;                 //EINT2 is falling edge sensitive, 1 for rising
edge

        NVIC_EnableIRQ(EINT2_IRQn);

                while(1) ;

         }


 void EINT2_IRQHandler(void)

 {

        LPC_GPIO1->FIOPIN = ~ LPC_GPIO1->FIOPIN ;

        LPC_SC->EXTINT = 0x00000004; //clears the interrupt

 }
```

2. **Toggle P0.2 every second using Timer interrupt.**

```
include<stdio.h>
#include<LPC17xx.h>
unsigned int ticks=0;
void TIMER0_IRQHandler(void)
{
    ticks++;
    if(ticks==1000)
    {
        ticks=0;
        LPC_GPIO0->FIOPIN=~(LPC_GPIO0->FIOPIN & 0x00000004);
    }
    LPC_TIM0->IR = 1;
    }
void init_timer0(void)
{
    //LPC_SC->PCONP |= (1<<1);    //powers the T0
    LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
    LPC_TIM0->CTCR =0x00;
    LPC_TIM0->MR0 = 3000;
    LPC_TIM0->EMR = 0X00;
    LPC_TIM0->PR = 0;
    LPC_TIM0->MCR = 0x00000003;
    LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
    return;
    }
int main(void)
{
    LPC_GPIO0->FIODIR=0x00000004;

    init_timer0();

    NVIC_EnableIRQ(TIMER0_IRQn);//timer 0 intr enabled in NVIC

        while(1);

}
```

3. **Toggle P0.2 whenever counter value reaches 3.   I. e for every 4 edges using counter interrupt.**

```
#include<stdio.h>

#include<LPC17xx.h>

void TIMER0_IRQHandler(void)

{

        LPC_GPIO0->FIOPIN=~(LPC_GPIO0->FIOPIN & 0x00000004);

        LPC_TIM0->IR = 1;

}

void init_timer0(void)

{

    //LPC_SC->PCONP |= (1<<1);    //powers the T0

    LPC_TIM0->TCR = 0x00000002;    // Timer0 Reset

    LPC_TIM0->CTCR =0x05;

    LPC_TIM0->MR0 = 3;

    LPC_TIM0->EMR = 0X30;

    LPC_TIM0->PR = 0;

    LPC_TIM0->MCR = 0x00000003;

    LPC_TIM0->TCR = 0x00000001;    // Timer0 Enable

    return;

    }

int main(void)

{

    LPC_GPIO0->FIODIR=0x00000004;
    LPC_PINCON->PINSEL3 |=((3<<22)|(3<<24));
    init_timer0();
    NVIC_EnableIRQ(TIMER0_IRQn);
    while(1);
}
```

4. **Timer interrupt for rectangular waveform generation**

```
include<stdio.h>
#include<LPC17xx.h>
unsigned char flag=1;
void TIMER0_IRQHandler(void)
{
    if(flag)
    {
        flag=0;
        LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
        LPC_GPIO0->FIOCLR=0x00000004;
        LPC_TIM0->MR0 = 500;
        LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
    }
    else
        {
        flag=1;
        LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
        LPC_GPIO0->FIOSET=0x00000004;
        LPC_TIM0->MR0 = 1500;
        LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
    }
        LPC_TIM0->IR = 1;
    }
```

```
void init_timer0(void)
{
        //LPC_SC->PCONP |= (1<<1);    //powers the T0
        LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
        LPC_TIM0->CTCR =0x00;
        LPC_TIM0->MR0 = 1500;
        LPC_TIM0->EMR = 0X30;
        LPC_TIM0->PR = 3000;
        LPC_TIM0->MCR = 0x00000005;
        LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
        LPC_GPIO0->FIOSET=0x00000004;
        return;
        }
int main(void)
{
        LPC_GPIO0->FIODIR=0x00000004;
        //LPC_PINCON->PINSEL3 |=((3<<22)|(3<<24));
        init_timer0();
        NVIC_EnableIRQ(TIMER0_IRQn);
                while(1);
}
```

5. **Timer polling for waveform generation**

```
#include<stdio.h>
#include<LPC17xx.h>
void delay(void)
{
        //LPC_SC->PCONP |= (1<<1);    //powers the T0
        LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
        LPC_TIM0->EMR = 0X20;
        LPC_TIM0->PR = 2999;
        LPC_TIM0->MCR = 0x00000004;     // stop PC and TC on MR0
        LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
        while ( !(LPC_TIM0->EMR & 0x01));
        return;
        }
int main(void)
{
        LPC_GPIO0->FIODIR=0x00000004;
                while(1)
        {
                LPC_GPIO0->FIOPIN=0x00000004;
                LPC_TIM0->MR0 = 1500;
                delay();
                LPC_GPIO0->FIOPIN=0x00000000;
                LPC_TIM0->MR0 = 500;
                delay();
}
}
```

6. **Waveform generation program with capture input demo. Whenever signal is input on CAP 0 and CAP1 inputs TC is captured into CR0 and CR1**

```
#include<stdio.h>
#include<LPC17xx.h>
void delay(void)
{
        //LPC_SC->PCONP |= (1<<1);    //powers the T0
        LPC_TIM0->CCR=9;//capture on positive edge
        LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
        LPC_TIM0->EMR = 0X20;//Set match bit upon match
        LPC_TIM0->PR = 3000; //for 1 ms
        LPC_TIM0->MR0 = 1000;           //for 1 second
        LPC_TIM0->MCR = 0x00000004;    // stop PC and TC on MR0
        LPC_TIM0->TCR = 0x00000001;    // Timer0 Enable
        while ( !(LPC_TIM0->EMR & 0x01)); // wait until match
        return;
        }

int main(void)
{
        LPC_GPIO0->FIODIR=0x00000004;
        LPC_PINCON->PINSEL3 |=(3<<20) | (3<<22);//select cap 0.0 and cap 0.1

                while(1)
        {
                LPC_GPIO0->FIOPIN=~(LPC_GPIO0->FIOPIN & 0x00000004);//toggle p0.2
                delay();



        }
}
```

7. Square waveform on MAT 0.0 output line by taking EM0 on the output pin.

```
#include<stdio.h>
#include<LPC17xx.h>
void delay(void)
{
        //LPC_SC->PCONP |= (1<<1);   //powers the T0
        LPC_TIM0->TCR = 0x00000002;    // Timer0 Reset
        LPC_TIM0->CTCR = 0x00000000;
        LPC_TIM0->CCR = 0x00000000;
        LPC_TIM0->EMR = 0X30;//Toggle bit upon match
        LPC_TIM0->PR = 0; //
        LPC_TIM0->MR0 = 3000000;       //
        LPC_TIM0->MCR = 0x00000002;     // Reset TC
        LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
        return;
        }
int main(void)
{
                LPC_PINCON->PINSEL3 |= (3<<24);//MAT pin activation
                delay();
                while(1);
}
```

8. MAT 1.1 toggles whenever count reaches 3. CAP 1.0 is counter clock

```
#include<stdio.h>
#include<LPC17xx.h>
void init_timer0(void)
{
        //LPC_SC->PCONP |= (1<<1);    //powers the T0
        LPC_PINCON->PINSEL3 |=(3<<18 | 3<<4);//  mat 1.1 and cap 1.0
        LPC_TIM1->TCR=2;
        LPC_TIM1->CTCR |=0x2;
        LPC_TIM1->MR1=0x03;
        LPC_TIM1->MCR=0x10;
        LPC_TIM1->EMR=0xC0;
        LPC_TIM1->TCR=1;
}
```

```
int  main(void)

{

        init_timer0();

        while(1);

}
```