Báo cáo thí nghiệm số 4

GVHD :Thầy Nguyễn Xuân Minh

Sinh viên :Cổ Chí Hào 1510888

Nguyễn Đình Hoàng Qúy

1.Mục đích bài thí nghiệm :

- Mục đích của bài thí nghiệm là nhằm khảo sát các chế độ hoạt động của bộ khảo sát định thời timer0 và timer1.

- Tạo module Timer.c để khởi động và sử dụng bộ định thời.

- Khai thác module Lcd.c để hiển thị ra màn hình Lcd.

2.Thực hiện các yêu cầu :

Nhóm em đã thực hiện các yêu cầu chính của buổi thí nghiệm gồm có 4 yêu cầu như sau :

2.1. Yêu cầu 1 : Sử dụng bộ Timer0 tạo ngắt quãng thời gian thực 10ms theo công thức tính t.

-Mức độ hoàn tất : Đã hoàn thành.

Các bước thực hiện yêu cầu đã hoàn thành :

-Bước 1: Tạo dự án mới Tn04_Timer, tập tin Tn04_Timer.c

-Bước 2: Khởi động port LED.

-Bước 3: tạo mới module định thì có tên Timer.c

-Bước 4: Tham khảo thanh ghi điều khiển bộ định thì Timer0 T0CON

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Trong module Timer.c viết hàm khởi động timer0_init() :

```
#include<xc.h>

#include "tn04.h"

#define SODEM10MS -12500

void timer0_reset();

void timer0_init()

{

   RCONbits.IPEN =1;          //cho phep dung uu tien

   INTCON2bits.TMR0IP = 0;     // timer0 uu tien thap

   INTCONbits.TMR0IE = 1;      // cho phep ngat timer0

   INTCONbits.GIEH = 1;        //cho phep ngat uu tien cao
```

```
   INTCONbits.GIEL = 1;        // cho phep ngat uu tien thap
   T0CON =0;                   // dung Fosc/4,prescaler =2
   timer0_reset();
}
```

-Bước 5: Viết hàm timer0_reset() thực hiện : xóa cờ ngắt, ngưng đếm, nạp số đếm vào bộ đếm, bắt đầu đếm lại.

```
void timer0_reset()
{
   TMR0IF = 0;     //xoa co ngat
   TMR0ON = 0;     // ngung dem
   TMR0 = SODEM10MS; //Nap so dem
   TMR0ON = 1;      //bat dau dem
}
```

-Bước 6: Tính số đếm sử dụng công thức

$$t_{timer} = \frac{4}{F_{OSC}} * prescaler * SODEM10MS$$

Trong đó :

-t_{timer} là đơn vị thời gian cần tạo ra, ở đây là 10 ms.

-F_{OSC} là tần số xung clock cấp cho CPU (10MHz).

-prescaler =2 do T0CON qui định.

Sau khi tính ra giá trị số đếm là : $SODEM10MS = -12500$, dấu trừ để đếm lên 0.

-Bước 7: Viết chương trình phục vụ ngắt quãng cho Timer0 (timer0_isr())

```
void __interrupt(low_priority) timer0_isr()
{
   if(TMR0IE & TMR0IF)
   {
      timer0_reset();     // nap lai so dem
      timer_process();
```

```
    }
}
```

-**Bước 8:** Trong cây dự án, vào mục Header files chọn new→C Header File thêm vào file Timer.h.

```
#ifndef TIMER_H
#defineTIMER_H
extern void timer0_init();
#endif  /* TIMER_H */
```

-Bước 9: Thêm header file Tn04.h với nội dung như sau:

```
#ifndef TN04_H
#defineTN04_H
extern void timer_process();
#endif  /* TN04_H */
```

-Bước 10: thêm vào đầu module Timer.c dòng #include "Tn04.h"

-Bước 11,12,13,14: Viết module chính Tn04_Timer.c chứa các hàm init(), timer_process() và main(), sau đó dịch và chạy

```
#include<xc.h>
#pragma config OSC=HS, WDT = OFF, LVP = OFF
#include "Timer.h"
volatile unsigned char led __at(0xF8C); //LATD
volatile unsigned char led_io __at(0xF95); //TRISD
#define XUAT 0
#define NHAP 1
#define SODEM500MS 50
unsigned char dem500ms;

void timer_process()
{
```

```
    if((--dem500ms)==0)
    {
        led++;
        dem500ms = SODEM500MS;
    }
}
void init()
{
    ADCON1 = 0x0F;
    led_io = XUAT;
    led = 0x00;
    dem500ms = SODEM500MS;
}
void main()
{
    init();
    timer0_init();
    while(1);
}
```

Yêu cầu 2 : Trên cơ sở ngắt thời gian 10 ms, kết hợp thêm biến đếm phụ dem1s (với số lần đếm SODEM500MS*2), viết hàm tre1s() để có thời gian trễ 1 s dùng cho việc điều khiển hiển thị led.

-Mức độ hoàn tất : Đã hoàn thành

Các bước thực hiện yêu cầu :

-**Bước 15:** Trong module Tn04B.c, định nghĩa thêm biến dem1s, thêm vào hàm tre1s()và sửa lại hàm timer_process() :

```
#include<xc.h>
```

```
#pragma config OSC=HS, WDT = OFF, LVP = OFF
#include "Timer.h"
volatile unsigned char led __at(0xF8C); //LATD
volatile unsigned char led_io __at(0xF95); //TRISD
#define XUAT 0
#define NHAP 1
#define SODEM500MS 50
unsigned char dem1s;

void timer_process()
{
   if(dem1s>0) dem1s--;
}
Void tre1s()
{
 Dem1s=SODEM500MS*2;
While(dem1s>0);
}

void init()
{
   ADCON1 = 0x0F;
   led_io = XUAT;
   led = 0x00;
   dem500ms = SODEM500MS;
}
void main()
```

```
{
   init();
   timer0_init();
   while(1)
{ tre1s() ; led++ ;}
}
```

Yêu cầu 3 :

-Bước 17 : Thay đổi giá trị prescaler thành 8 , tính lại số đếm để thời gian xảy ra ngắt không đổi.

-Mức độ hoàn tất : Đã hoàn thành

Trong hàm timer0_init() ta thay đổi giá trị của T0CON để prescaler là 8

```
void timer0_init()
{
   RCONbits.IPEN =1;          //cho phep dung uu tien
   INTCON2bits.TMR0IP = 0;     // timer0 uu tien thap
   INTCONbits.TMR0IE = 1;      // cho phep ngat timer0
   INTCONbits.GIEH = 1;        //cho phep ngat uu tien cao
   INTCONbits.GIEL = 1;        // cho phep ngat uu tien thap
   T0CON =0x02;                // dung Fosc/4,prescaler =8
   timer0_reset();
}
```

-Bước 18 : Tính lại hằng số SODEM10MS để giữ nguyên thời gian ttimer = 10ms.

Với precscaler = 8 ta có :

$$SODEM10MS = \frac{t_{timer} * F_{OSC}}{4 * prescaler} = \frac{10 * 10^{-3} * 10 * 10^{6}}{4 * 8} = 3125$$

-Bước 19 : chạy lại chương trình như đã thực hiện ở trên

-Bước 20 : Kết hợp tra bảng dữ liệu như trong bài Tn03 để xuất ra các quy luật chạy led.

```
#include <xc.h>
#pragma config OSC=HS, WDT=OFF, LVP=OFF
// High speed, tan so = 10MHz
#include "Timer.h"
volatile unsigned char led @ 0xF8C; //LATD
volatile unsigned char led_io @ 0xF95; //TRISD

#define XUAT        0
#define NHAP        1
#define MAXIDX  8
#define MAXROM  4
#define SODEM500MS  10

const unsigned char dl_rom[MAXROM][MAXIDX] = {
    {0xFF,0x7E,0x3C,0x18,0x00,0x18,0x3C,0x7E},
    {0x01,0x03,0x07,0x0F,0x1F,0x3F,0x7F,0xFF},
    {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80},
    {0x81,0xC3,0xE7,0xFF,0xFF,0xE7,0xC3,0x81}
};
unsigned char dl_ram[MAXIDX],ramidx,romidx;
unsigned char dem500ms;
unsigned char dem1s;

/*void timer_process()
{
    if ((--dem500ms) == 0) //kiem tra du 500ms moi lam
    {
```

```
        led++;
        dem500ms = SODEM500MS;
    }
}

void xuatled()
{
    led = dl_ram[ramidx++];
    ramidx %= MAXIDX;
}
void timer_process()
{
    if ((--dem500ms) == 0)
    {
        xuatled();
        dem500ms = SODEM500MS;
    }
}

void doi_ql()
{
    unsigned char i;
    for (i = 0; i < MAXIDX; i++)
        dl_ram[i] = dl_rom[romidx][i];
    romidx++;
    romidx %= MAXROM;
    ramidx = 0;
```

```
}
void tre1s()
{
   dem1s = SODEM500MS*2;
   while(dem1s > 0);
}

void init()
{
   ADCON1 = 0x0F;
   led_io = XUAT;
   led = 0x00;
   dem500ms = SODEM500MS;
}

void main()
{
   init();
   timer0_init();
   while(1)
   {
      tre1s();
      led++;
   }
}
```

Yêu cầu 4 : sử dụng thêm module Lcd.c có sẵn để hiển thị ra màn hình LCD các ký tự sau mỗi giây.

-Mức độ hoàn tất : Đã hiển thị được 1 kí tự số lên màn hình LCD.

-Bước 21 : code hiển thị kí tự số lên màn hình LCD

```
#include <xc.h>
#include <stdio.h>
#include "Lcd.h"
#pragma config OSC=HS, WDT=OFF, LVP=OFF
#define N


void init()
{
   ADCON1 = 0x0F; //nhap digital
}


void main()
{
   float x;
   unsigned char i;
   init();
   lcd_init();
   for(i = 1,x = 1; i <= N; i++)
      x *= i;
   printf("\f%d!=\n%e",N,x);
   while(1);
}
```

3.Bài tập

a) Làm lại bài thí nghiệm nhưng sử dụng Timer1 thay vì Timer0 (thêm vào module Timer.c các hàm timer1_init(), timer1_reset(), timer1_isr() và tạo thêm file Timer1.h.

Hàm timer1_init()

```
void timer1_init()
{
   RCONbits.IPEN =1;          //cho phep dung uu tien
   IPR1bits.TMR1IP = 0;     // timer1 uu tien thap
   PIE1bits.TMR1IE = 1;      // cho phep ngat timer1
   INTCONbits.GIEH = 1;        //cho phep ngat uu tien cao
   INTCONbits.GIEL = 1;        // cho phep ngat uu tien thap
   T1CON =0x10;                 // dung Fosc/4,prescaler =2
   timer1_reset();
}
```

Hàm timer1_reset()

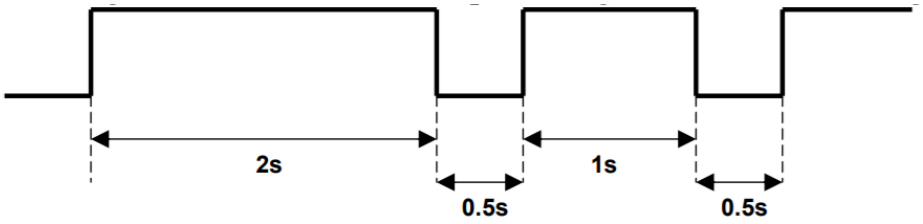
```
void timer1_reset()
{
   TMR1IF = 0;     //xoa co ngat
   TMR1ON = 0;     // ngung dem
   TMR1 = SODEM10MS; //Nap so dem
   TMR1ON = 1;      //bat dau dem
}
```

Hàm timer1_isr()

```
void __interrupt(low_priority) timer0_isr()
{
   if(TMR0IE & TMR0IF)
   {
```

```
        timer1_reset();     // nap lai so dem
        timer_process();
    }
}
```

b) Viết và sử dụng hàm tre500ms() để phát xung ra chân RD7 có dạng như sau




```
#include<xc.h>
#pragma config OSC=HS, WDT = OFF, LVP = OFF
#include "Timer1.h"
volatile unsigned char led __at(0xF8C); //LATD
volatile unsigned char led_io __at(0xF95); //TRISD
#define XUAT 0
#define NHAP 1
#define SODEM500MS 50
unsigned char dem1s;
unsigned char dem500ms ;

void timer_process()
{
    if(dem500ms>0) dem500ms--;
}
void tre500ms()
{
```

```
dem500ms=SODEM500MS;
while(dem500ms>0);
}

void init()
{
   ADCON1 = 0x0F;
   led_io = XUAT;
   led = 0x00;
   dem500ms = SODEM500MS;
}
void main()
{
   init();
   timer1_init();
   while(1)
{
       led = 0x80 ;
       tre500ms() ;
       tre500ms() ;
       tre500ms() ;
       tre500ms() ;
       led = 0x00 ;
       tre500ms() ;
       led = 0x80 ;
       tre500ms() ;
       tre500ms() ;
```

```
        led = 0x00 ;
        tre500ms() ;
    }
}
```

c) Dùng bộ định thời tạo xung vuông chu kì 10ms, duty cycle 30%.

Ta tính lại SODEM1MS như sau :

$$SODEM1MS = \frac{t_{timer} * F_{OSC}}{4 * prescaler} = \frac{10^{-3} * 10 * 10^{6}}{4 * 2} = 1250$$

Khai báo lại trong module Timer.c

```
#define SODEM1MS -1250
```

Trong module Timer.c ta sửa lại hàm timer0_reset với số đếm là 1ms

```
void timer0_reset()
{
    TMR0IF = 0;     //xoa co ngat
    TMR0ON = 0;     // ngung dem
    TMR0 = SODEM1MS; //Nap so dem
    TMR0ON = 1;      //bat dau dem
}
```

Tạo tập tin C source file có tên BAITAPC.c, để dùng bộ định thời tạo xung vuông chu kì 10ms, duty cycle 30% ta sẽ viết hàm trễ 1ms để xuất ra port led sau mỗi khoảng thời gian tương ứng.

```
#include<xc.h>
#pragma config OSC=HS, WDT = OFF, LVP = OFF
#include "Timer1.h"
volatile unsigned char led __at(0xF8C); //LATD
volatile unsigned char led_io __at(0xF95); //TRISD
```

```
#define XUAT 0
#define NHAP 1
#define DEM1MS 1
unsigned char dem1ms;


void timer_process()
{
   if(dem1ms>0) dem1ms--;
}
void tre1ms()
{
dem1ms=DEM1MS;
while(dem1ms>0);
}

void init()
{
   ADCON1 = 0x0F;
   led_io = XUAT;
   led = 0x00;
   dem1s = DEM1MS;
}
void main()
{
   init();
   timer1_init();
```

```
   while(1)
{

      led = 0x80 ;
      tre1ms() ;
      tre1ms() ;
      tre1ms() ;
      led = 0x00 ;
      tre1ms() ;
      tre1ms() ;
      tre1ms() ;
      tre1ms() ;
      tre1ms() ;
      tre1ms() ;
      tre1ms() ;
   }
}
```