

FANPAGE: THT UTC

NGUỒN: DUY VŨ\_ THT UTC

GROUP: UTC ĐẠI CƯƠNG MÔN PHẢI (THT UTC)

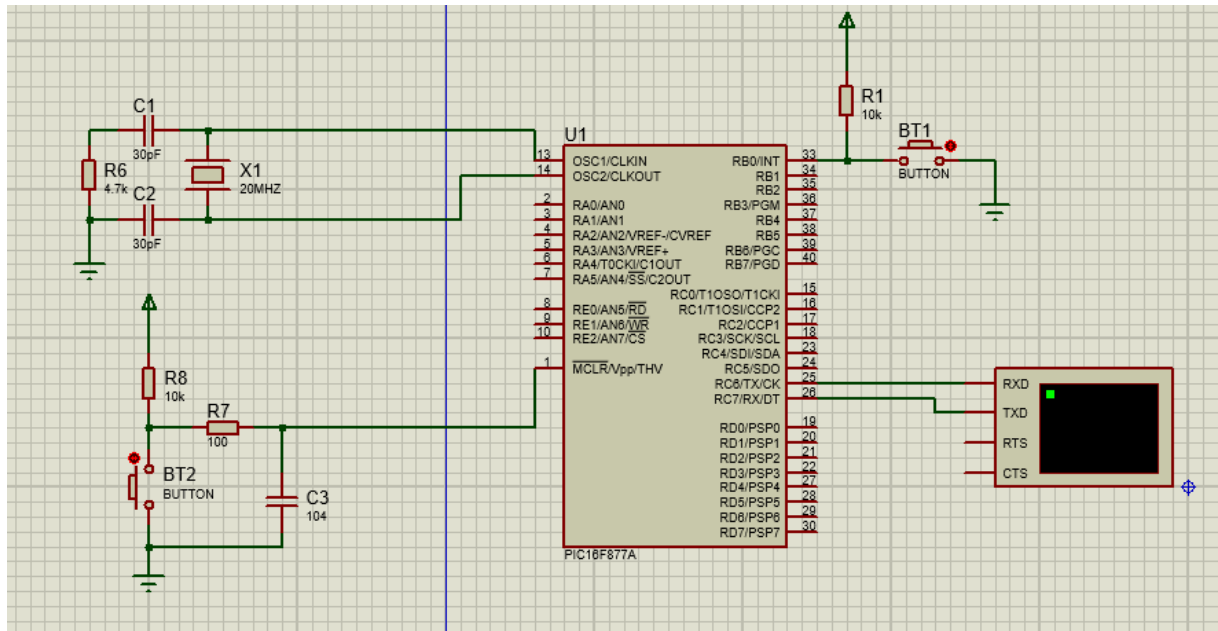
Câu 3 \*đề 6 : Xây dựng sơ đồ và lập trình sử dụng vi xử lý thực hiện đồng thời các nhiệm vụ sau:

-Khi 01 ấn phím đầu vào thì thực hiện truyền UART dữ liệu sau :

0x66	0x01	0x03	0x11	0x67
------	------	------	------	------

-Nhận và lưu vào mảng 5 byte dữ liệu từ vi xử lý khác qua cổng UART.

Tốc độ truyền thông truyền và nhận là 19200b/s.



/\*

\* File: mainde7.c

\* Author: VU VAN DUY

\*SO 6

\* Created on December 22, 2020, 11:08 PM

\*/

// CONFIG

#pragma config FOSC = HS

#pragma config WDTE = OFF

#pragma config PWRTE = OFF

#pragma config BOREN = ON

```
#pragma config LVP = OFF
#pragma config CPD = OFF
#pragma config WRT = OFF
#pragma config CP = OFF
```

```
#include <xc.h>
#define sw RB0
#define _XTAL_FREQ 20000000
```

```
unsigned char b[] = {0x66,0x01,0x03,0x11,0x67};
unsigned char b1[];
char UART_Init(const long int baudrate)
{
    unsigned int x;
    x = (_XTAL_FREQ - baudrate*64)/(baudrate*64);
    if(x>255)
    {
        x = (_XTAL_FREQ - baudrate*16)/(baudrate*16);
        BRGH = 1;
    }
    if(x<256)
    {
        SPBRG = x;
        SYNC = 0; //chon che do khong dong bo
        SPEN = 1; // cho phep truyen thong
        TRISC7 = 1;
        TRISC6 = 1;
        CREN = 1; //cho phep nhan bit lien tuc
        TXEN = 1; //cho phep truyen
        return 1;
    }
}
```

```

        return 0;
    }

// kiem tra du lieu truyen(tra ve 1 neu du lieu da co con tra ve 0 neu rong)
char UART_TX_Empty()
{
    return TRMT;
}

//Kiem tra du lieu nhan duoc hay khogn
char UART_Data_Ready()
{
    return RCIF;
}

//Doc mot ki tu,doi den khi nhan du 8 bit tu ben gui
char UART_Read()
{
    while(!RCIF);
    return RCREG;
}

//Doc mot chuoi
void UART_Read_Text(char *Output, unsigned int length)
{
    int i;
    for(int i=0;i<length;i++)
        Output[i] = UART_Read();
}

//viet mot ki tu
void UART_Write(char data)

```

```

{
    while(!TRMT);
    TXREG = data;
}

//viet mot chuoi
void UART_Write_Text(char *text)
{
    int i;
    for(i=0;text[i]!='\0';i++)
        UART_Write(text[i]);
}

void truyen_mang(){
    for(int i = 0;i<5;i++){
        UART_Write(b[i]);
        __delay_ms(100);
    }
}

void nhan_mang(){
    for(int i = 0;i<5;i++){
        if(UART_Data_Ready()){
            b1[i] = UART_Read();
            __delay_ms(100);
        }
    }
}

void main(void) {
    TRISB = 0xff;
    TRISCbits.TRISC6 = 0;//output(TX)

```

```
TRISCbits.TRISC7 = 1;//input(RX)
UART_Init(19200);
while(1){
    if(sw == 0){
        __delay_ms(15);
        if(sw == 1){
            truyen_mang();
        }
    }
}
```

Câu 3:\* đề 03

-Hãy thiết kế sơ đồ phần cứng ghép nối 4 LED 7 thanh và 1 phím bấm với vi điều khiển. Viết chương trình thực hiện đồng thời các công việc sau:

-Khi phím được bấm thì hiển thị một số đếm tăng dần từ 0 đến 9999, mỗi giây số đếm tăng 1 đơn vị. Khi phím không được bấm thì số hiển thị không tăng.

-Tạo một xung vuông tần số 20KHz tại chân P3.7 sử dụng ngắt Timer.(thời gian xung tồn tại xung là 50%)

/\*

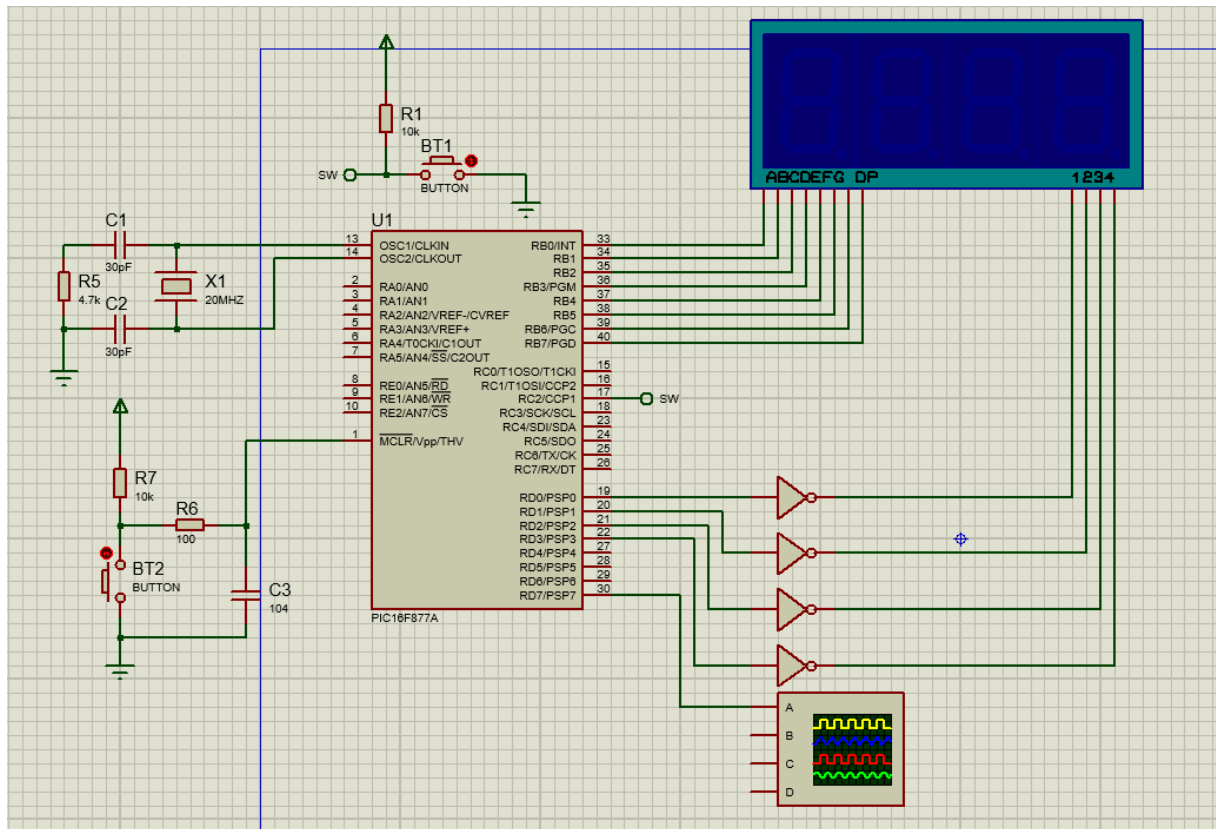
\* File: newmain.c

\* Author: VU VAN DUY

\* SO 3

\* Created on December 2, 2020, 4:58 PM

\*/



// PIC16F877A Configuration Bit Settings

// 'C' source line config statements

// CONFIG

```
#pragma config FOSC = HS      // Oscillator Selection bits (HS oscillator)
#pragma config WDTE = OFF     // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF    // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = OFF    // Brown-out Reset Enable bit (BOR disabled)
#pragma config LVP = OFF      // Low-Voltage (Single-Supply) In-Circuit Serial
                             // Programming Enable bit (RB3 is digital I/O, HV on MCLR must be used for
                             // programming)
#pragma config CPD = OFF      // Data EEPROM Memory Code Protection bit
                             // (Data EEPROM code protection off)
#pragma config WRT = OFF      // Flash Program Memory Write Enable bits
                             // (Write protection off; all program memory may be written to by EECON control)
```

```
#pragma config CP = OFF          // Flash Program Memory Code Protection bit
(Code protection off)
```

```
// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.
```

```
#include <xc.h>
#define _XTAL_FREQ 20000000
#define SW RC2
unsigned int count = 0;
#define Xung RD7
```

```
void timer2_init(){
    T2CONbits.T2CKPS0 = 0;
    T2CONbits.T2CKPS1 = 0;
    T2CONbits.TMR2ON = 1;
    T2CONbits.TOUTPS0 = 0;
    T2CONbits.TOUTPS1 = 0;
    T2CONbits.TOUTPS2 = 0;
    T2CONbits.TOUTPS3 = 0;
    TMR2=231;    // Load the timer value for 5us delay
    TMR2IE=1;    //Enable timer interrupt bit in PIE1 register
    GIE=1;       //Enable Global Interrupt
    PEIE=1;      //Enable the Peripheral Interrupt
}
```

```
void delay_timer0_5ms(unsigned int time)
{
    while (time-- > 0) //1? p 1? i time 1? n 5ms.
    {
        OPTION_REGbits.T0CS = 0; //Dùng xung n?i
        OPTION_REGbits.PSA = 0; //Dùng b? chia timer0
    }
}
```



```

    OPTION_REGbits.PS2 = 1;
    OPTION_REGbits.PS1 = 1;
    OPTION_REGbits.PS0 = 0; //Ch?n b? chia 1:128
    TMR0 = 60; //??t giá tr? cho TMR0
    while(!INTCONbits.TMR0IF); //??i c? tràn
    INTCONbits.TMR0IF = 0; //Xóa c? tràn
}
}
void hienthi(unsigned int s)
{
    unsigned int s1,s2,a,b,c,d;
    const unsigned char MALED[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99,0x92,
0x82,0xF8, 0x80, 0x90};
    a=(s/1000);
    b=(s% 1000)/100;
    c=(s% 100)/10;
    d=s% 10;

    for(int i=0;i<20;i++) //Tao tre 1s
    {
        PORTDbits.RD0=0; PORTB = MALED[a];
        delay_timer0_5ms(2); PORTDbits.RD0=1;
        PORTDbits.RD1=0; PORTB = MALED[b];
        delay_timer0_5ms(1); PORTDbits.RD1=1;
        PORTDbits.RD2=0; PORTB = MALED[c];
        delay_timer0_5ms(2); PORTDbits.RD2=1;
        PORTDbits.RD3=0; PORTB = MALED[d];
        delay_timer0_5ms(1); PORTDbits.RD3=1;
    }
}

//Khoi tao ham ngat timer2 voi do tre 5us

```

```

void __interrupt() timer2(){
    if(TMR2IF){
        count++;
        if(count <= 5)
        {
            Xung =1;
        }
        else if(count > 5 && count < 10)
        {
            Xung = 0;
        }
        if(count == 10){
            count = 0;
        }
        TMR2IF = 0;
        TMR2 = 231;
    }
}

```

```

void main(void) {
    timer2_init();
    TRISB = 0x00;
    TRISD = 0x00;
    TRISCbits.TRISC2 = 1;
    PORTB = 0x00;
    PORTD = 0x00;
    unsigned int d = 0;
    while(1)
    {
        if(SW == 0){
            d++;
        }
    }
}

```

```
    if(d>9999)
    {
        d=0;
    }
    hienthi(d);

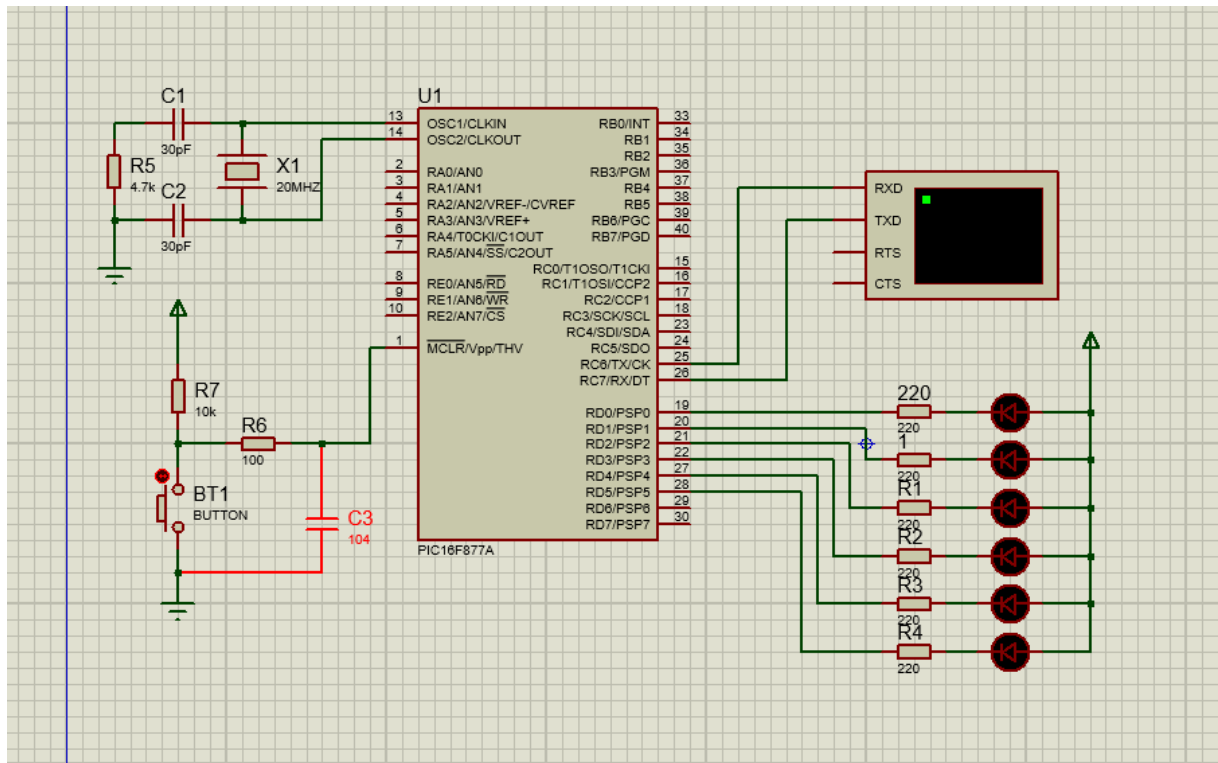
}

}
```

### Câu 3 \*Đề 07

Xây dựng sơ đồ và sử dụng vi xử lí thực hiện đồng thời thực hiện các nhiệm vụ sau:

- Sáng thực hiện các LED từ LED 1 đến LED 6 với chu kì 0,1s,sau đó nhấp nháy 3 lần cả 6 LED với chu kì 0,5s.Lặp lại quá trình trên.
- Cứ 0,6 giây truyền chuỗi kí tự “Blink LED”qua cổng UART với tốc độ 4800b/s.



```
#pragma config FOSC = HS
#pragma config WDTE = OFF
#pragma config PWRTE = OFF
#pragma config BOREN = ON
#pragma config LVP = OFF
#pragma config CPD = OFF
#pragma config WRT = OFF
#pragma config CP = OFF
```

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

```
#define _XTAL_FREQ 20000000
```

```

unsigned char dem = 0;
unsigned int count = 0, count1 = 0;
unsigned char LED[6]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf};
char UART_Init(const long int tocd0)
{
    unsigned int x;
    x = (_XTAL_FREQ - tocd0*64)/(tocdo*64);
    if(x>255)
    {
        x = (_XTAL_FREQ - tocd0*16)/(tocdo*16);
        BRGH = 1;
    }
    if(x<256)
    {
        SPBRG = x;
        SYNC = 0; //chon che do khong dong bo
        SPEN = 1; // cho phep truyen thong
        TRISC7 = 1;
        TRISC6 = 1;
        CREN = 1; //cho phep nhan bit lien tuc
        TXEN = 1; //cho phep truyen
        PIR1bits.RCIF = 1; //Ngat tran nhan du lieu
        PIR1bits.TXIF = 1; //Ngat tran truyen du lieuu
        INTCONbits.PEIE = 1; //cho phep ngat ngoai vi
        INTCONbits.GIE = 1; //cho phep ngat toan cuc.
        return 1;
    }
    return 0;
}

void truyen_kt(char kt)
{

```

```

while(!TRMT);
TXREG = kt;
}

//viet mot chuoi
void truyen_chuoi(char *txt)
{
    int i;
    for(i=0;txt[i]!='\0';i++)
        truyen_kt(txt[i]);
}

void timer0_init(){
    INTCONbits.GIE = 1;//kich hoat ngat toan cuc
    INTCONbits.PEIE = 1;//Kich hoat ngat ngoai vi
    INTCONbits.TMR0IE = 1;//kich hoat ngat dong tran TMR0
    INTCONbits.TMR0IF = 0;//co ngat tran TMR0
    OPTION_REGbits.T0CS = 0;//Chon chu ki lenh noi bo(CLK0)
    OPTION_REGbits.T0SE = 0;//
    OPTION_REGbits.PSA = 0;//Presale duoc gan cho modul timer0
    //Chon bo chia 1:256
    OPTION_REGbits.PS0 = 1;
    OPTION_REGbits.PS1 = 1;
    OPTION_REGbits.PS2 = 1;

}

void __interrupt() timer0(){
    if(TMR0IF){
        TMR0 = 101;
        TMR0IF = 0;
        count1++;
        count++;
    }
}

```

```

    if(count == 50 && dem != 6){
        PORTD = LED[dem];
        dem++;
        count = 0;
    }
    if(dem == 6){
        if(count <= 250 ){
            PORTD = 0x00;
        }
        else if(count >= 500 && count <= 750){
            PORTD = 0x00;
        }
        else if(count >= 1000 && count <= 1250){
            PORTD = 0x00;
        }
        else if(count == 1500){
            count = 0;
            dem = 0;
        }
        else{
            PORTD = 0xff;
        }
    }
    if(count1 == 600){
        truyen_chuoi("Blink LED");
        count1 = 0;
    }
}

void main(void) {
    TRISD = 0x00;
    TRISCbits.TRISC6 = 0;//output(TX)

```

```
TRISCbits.TRISC7 = 1;//input(RX)
PORTD = 0xff;
timer0_init();
UART_Init(4800);
while(1){

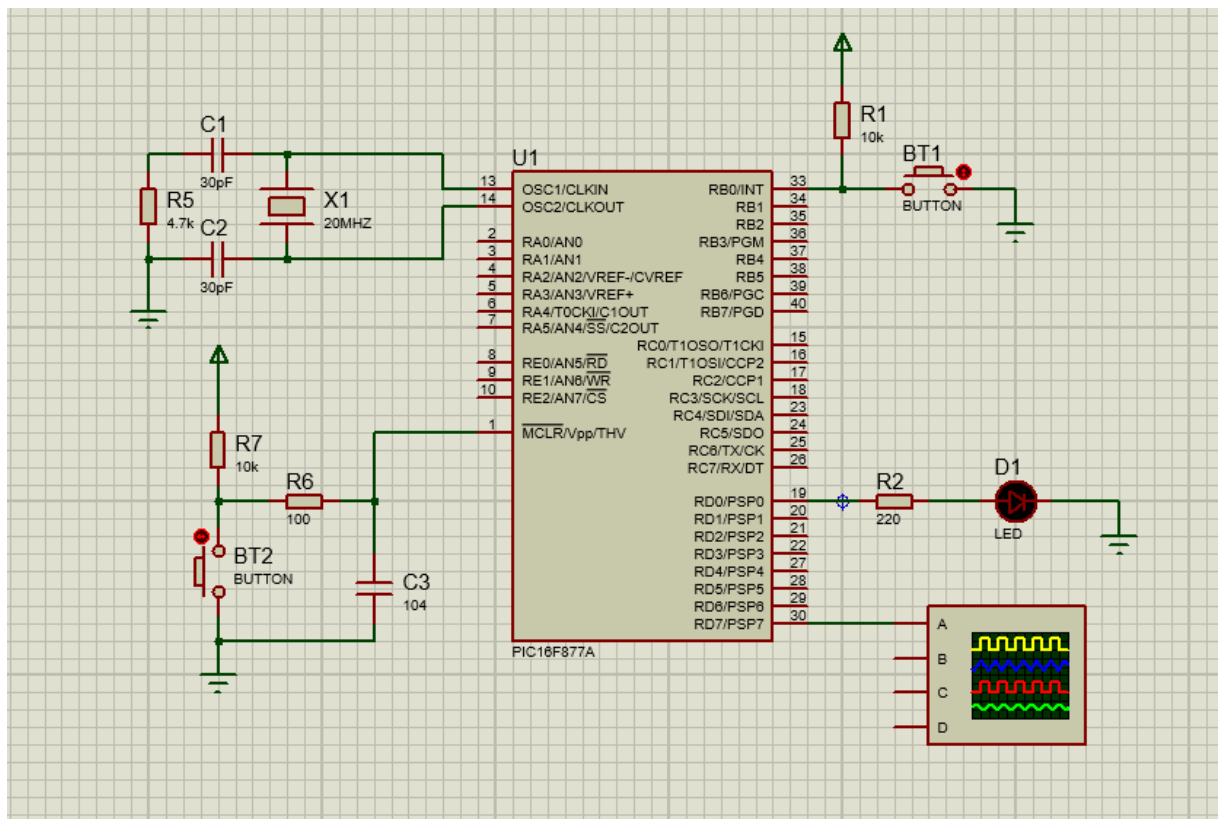
}
}
```



### Câu 3 \*Đề 12

Xây dựng sơ đồ và lập trình sử dụng vi điều khiển thực hiện đồng thời các nhiệm vụ sau:

- Điều khiển 1 LED đơn sang theo phím theo thuật toán: khi ấn phím Led sáng ngay, khi nhả phím sau 2s Led tắt.
- Tạo xung vuông tại cổng đầu ra 1 có chu kỳ 10ms và thời gian tồn tại xung 3ms sử dụng ngắt Timer.



/\*

\* File: mainde11.c

\* Author: VU VAN DUY

\*

\* Created on December 23, 2020, 8:40 PM

\*/

// CONFIG

#pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)

```
#pragma config PWRT = OFF    // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = ON     // Brown-out Reset Enable bit (BOR enabled)
#pragma config LVP = OFF      // Low-Voltage (Single-Supply) In-Circuit Serial
                             // Programming Enable bit (RB3 is digital I/O, HV on MCLR must be used for
                             // programming)
#pragma config CPD = OFF      // Data EEPROM Memory Code Protection bit
                             // (Data EEPROM code protection off)
#pragma config WRT = OFF      // Flash Program Memory Write Enable bits
                             // (Write protection off; all program memory may be written to by EECON control)
#pragma config CP = OFF       // Flash Program Memory Code Protection bit
                             // (Code protection off)
```

```
// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.
```

```
#include <xc.h>
#define _XTAL_FREQ 20000000
#define SBIT_PS2 2
#define Xung RD7
#define LED RD0
#define sw RB0
unsigned int count = 0, count1 = 0, check = 0;
```

```
void timer0_init(){
    OPTION_REG = (1<<SBIT_PS2);
    TMR0=100;    // Gia tri bat dau tao tre 1ms
    TMR0IE=1;
    GIE=1;
    PEIE=1;
}
```

```
void __interrupt() timer0(){
```

```

if(TMR0IF == 1){
    TMR0 = 101;
    TMR0IF = 0;
    count1++;
    if(count1 <= 3){
        Xung = 1;
    }
    else if(count1 > 3 && count1 < 10){
        Xung = 0;
    }
    if(count1 == 10){
        count1 = 0;
    }
}

if(sw == 0){
    LED = 1;
    check = 1;
}
if(sw == 1 && check == 1){
    count++;
    if(count == 2000){ //2s = 1ms * 2000
        LED = 0;
        count = 0;
        check = 0;
    }
}
}
}

```

```

void main(void) {
    timer0_init();
    TRISB = 0xff;
}

```

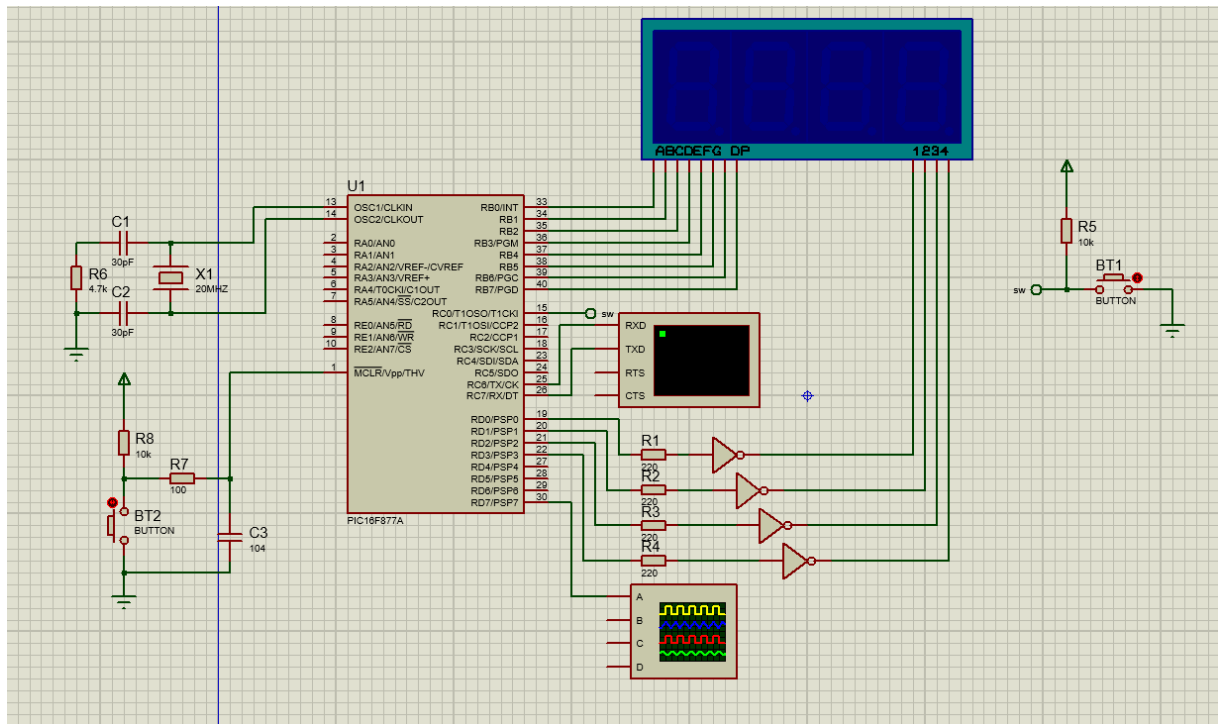
```
TRISD = 0x00;  
PORTD = 0x00;  
while(1){  
  
    }  
}
```

Câu 3:\*Đề 04:

Thiết kế sơ đồ phần cứng ghép nối 4LED 7 thanh và 1 phím bấm với vi điều khiển. Viết chương trình thực hiện đồng thời các công việc sau:

-Tạo xung vuông có chu kỳ 3ms và thời gian tồn tại xung 1ms tại P1.0 sử dụng ngắt Timer

Gửi chuỗi kí tự “Reading” qua cổng UART với chu kỳ 2s, tốc độ baudrate là 19200 b/s.



/\*

\* File: maind4.c

\* Author: VU VAN DUY

\* SO 04

\* Created on December 22, 2020, 1:28 AM

\*/

// CONFIG

#pragma config FOSC = HS

#pragma config WDTE = OFF

#pragma config PWRTE = OFF

#pragma config BOREN = ON

#pragma config LVP = OFF

#pragma config CPD = OFF

#pragma config WRT = OFF

```
#pragma config CP = OFF
```

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

```
#define _XTAL_FREQ 20000000
```

```
#define sw RC0
```

```
#define SBIT_PS2 2
```

```
#define Xung RD7
```

```
unsigned int count = 0, count1 = 0;
```

```
const unsigned char MALED[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92,  
0x82, 0xF8, 0x80, 0x90};
```

```
void timer0_init(){
```

```
    OPTION_REG = (1<<SBIT_PS2);
```

```
    TMR0=100;    //Khoi tao gia tri bat dau cho do tre 1ms
```

```
    TMR0IE=1;
```

```
    GIE=1;
```

```
    PEIE=1;
```

```
}
```

```
char UART_Init(const long int tocd0)
```

```
{
```

```
    unsigned int x;
```

```
    x = (_XTAL_FREQ - tocd0*64)/(tcd0*64);
```

```
    if(x>255)
```

```
    {
```

```
        x = (_XTAL_FREQ - tocd0*16)/(tcd0*16);
```

```
        BRGH = 1;
```

```
    }
```

```
    if(x<256)
```

```
    {
```

```
        SPBRG = x;
```

```

    SYNC = 0; //chon che do khong dong bo
    SPEN = 1; // cho phep truyen thong
    TRISC7 = 1;
    TRISC6 = 1;
    CREN = 1; //cho phep nhan bit lien tuc
    TXEN = 1; //cho phep truyen
    PIR1bits.RCIF = 1; //Ngat tran nhan du lieu
    PIR1bits.TXIF = 1; //Ngat tran truyen du lieuu
    INTCONbits.PEIE = 1; //cho phep ngat ngoai vi
    INTCONbits.GIE = 1; //cho phep ngat toan cuc.
    return 1;
}
return 0;
}

```

```

void truyen_kt(char kt)
{
    while(!TRMT);
    TXREG = kt;
}

```

```

//viet mot chuoi
void truyen_chuoi(char *txt)
{
    int i;
    for(i=0;txt[i]!='\0';i++)
        truyen_kt(txt[i]);
}

```

```

//Tao ngat voi do tre la 1ms
void __interrupt() timer0(){
    if(TMR0IF == 1)

```

```

{
    TMR0 = 101;
    TMR0IF = 0;
    count++;
    count1++;
    if(count <= 1)
    {
        Xung = 0;
    }
    else if(count >1 && count < 4)
    {
        Xung = 1;
    }
    if(count == 4)
    {
        count = 0;
    }
    if(count1 == 2000){
        //Gui chuoi voi chu ki 2s = 1ms * 2000 = 2000ms
        truyen_chuoi("Reading");
        count1 = 0;
    }
}
}

```

```

void main(void) {
    TRISD = 0x00;
    TRISB = 0x00;
    TRISCbits.TRISC6 = 0;//output(TX)
    TRISCbits.TRISC7 = 1;//input(RX)
}

```



```
TRISCbits.TRISC0 = 1;//Nut bam
PORTD = 0x00;
PORTB = 0x00;
timer0_init();
UART_Init(19200);
while(1){

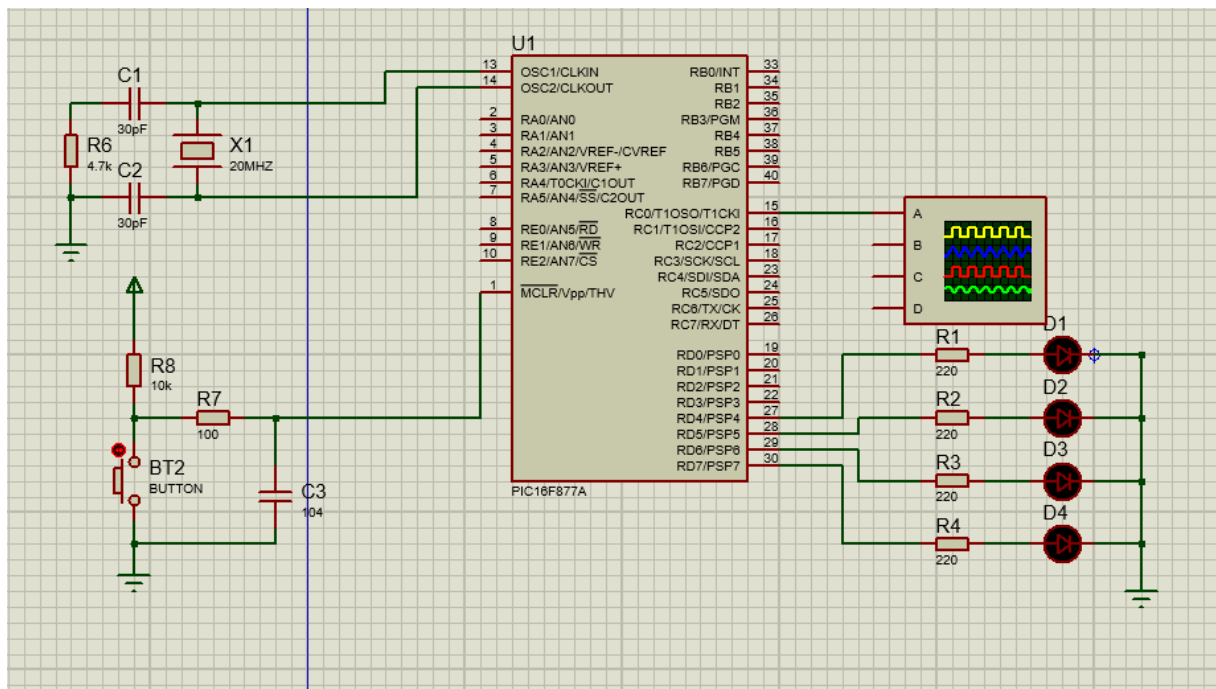
}
}
```

Câu 3: \*Đề 01:

Hãy thiết kế sơ đồ phần cứng ghép nối 4 LED đơn với vi điều khiển. Viết chương trình thực hiện đồng thời các công việc sau:

-Nháy sáng lần lượt các LED với tần số 0.5Hz, hết một vòng 4 Led thì nháy Led theo chiều ngược ..... lặp lại quá trên vô tận

-Tạo một xung vuông tần số 10KHz tại chân P3.0 sử dụng ngắt Timer.(thời gian tồn tại xung là 50%...



/\*

\* File: mainde5.c

\* Author: VU VAN DUY

\* SO 01

\* Created on December 22, 2020, 3:04 AM

\*/

// CONFIG

#pragma config FOSC = HS

#pragma config WDTE = OFF

#pragma config PWRTE = OFF

```

#pragma config BOREN = ON
#pragma config LVP = OFF
#pragma config CPD = OFF
#pragma config WRT = OFF
#pragma config CP = OFF
#include <xc.h>
#define _XTAL_FREQ 20000000
#define Xung RC0
#define SBIT_T2CKPS1 1
unsigned int count = 0;

```

```

unsigned char LED[10] =
{0x80,0xC0,0xE0,0xF0,0x00,0x10,0x30,0x70,0xF0,0x00};
void timer0_init(){
    OPTION_REG = 0x07;//Prescaler 256
    INTCONbits.GIE = 1;//kich hoat ngat toan cuc
    INTCONbits.PEIE = 1;//Kich hoat ngat ngoai vi
}
void timer2_init(){
    T2CON = (1<<SBIT_T2CKPS1);
    TMR2=240;
    TMR2IE=1;
    GIE=1;
    PEIE=1;
    TMR2ON = 1;
}
//Tao ngat voi tre 50us
void __interrupt() timer2(){
    if(TMR2IF==1){
        Xung = ~Xung;
        TMR2IF = 0;
        TMR2 = 241;
    }
}

```

```
    }  
}
```

```
void delay1s_timer0(){  
    for(int i=0;i<76;i++){  
        while(!T0IF);  
        T0IF = 0;  
    }  
}
```

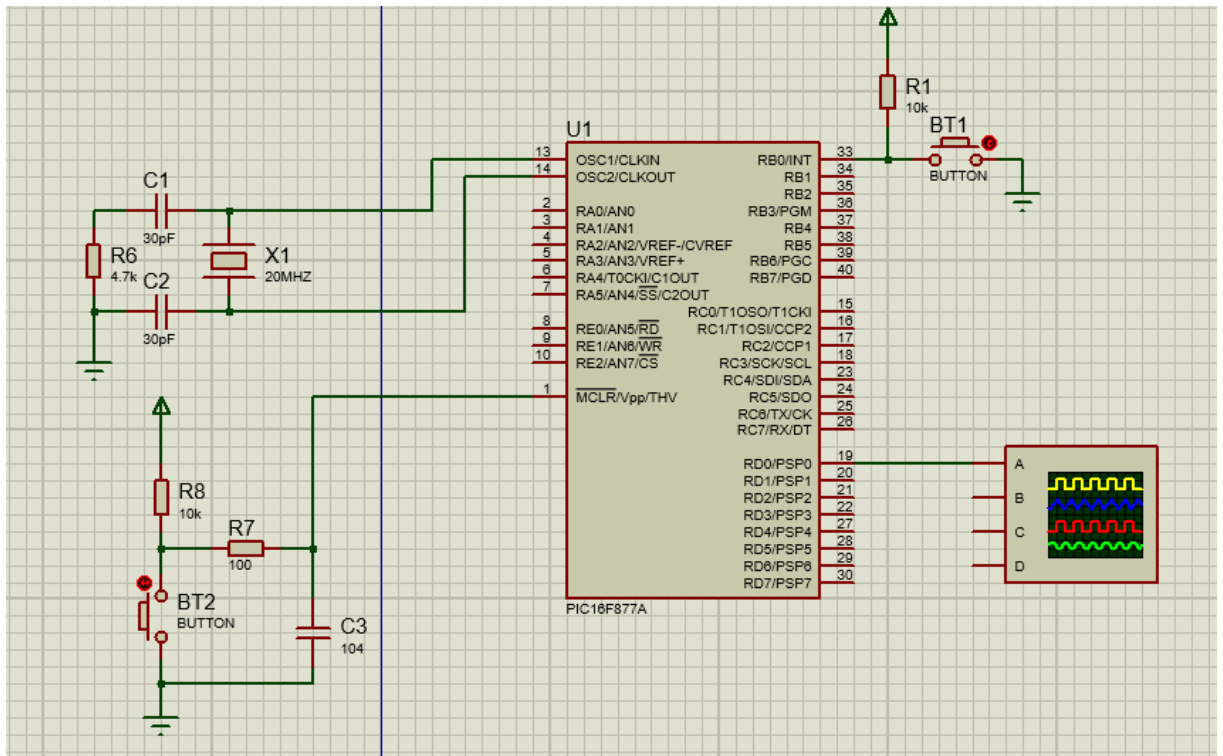
```
void nhayled(){  
    for(int i = 0;i < 10;i++){  
        PORTD = LED[i];  
        delay1s_timer0();  
    }  
}
```

```
void main(void) {  
    timer0_init();  
    timer2_init();  
    TRISD = 0x00;  
    TRISC = 0x00;  
    PORTC = 0x00;  
    PORTD = 0x00;  
    while(1){  
        nhayled();  
    }  
}
```

Câu 3 : \*Đề 08

Xây dựng sơ đồ và lập trình sử dụng vi xử lý thực hiện đồng thời các nhiệm vụ sau:

- Nhận và lưu 1 byte dữ liệu từ 1 vi xử lý khác ,khi ấn 1 phím sẽ truyền byte dữ liệu đã lưu.Quá trình nhận và truyền sử dụng cổng UART với tốc độ 2400 b/s.
- Liên tục tạo xung vuông tần số 1KHz,thời gian tồn tại xung là 0.2ms trên 1 cổng ra của vi xử lý.
-



/\*

\* File: mainde6.c

\* Author: VU VAN DUY

\*SO 8

\* Created on December 22, 2020, 3:05 PM

\*/

// CONFIG

#pragma config FOSC = HS

#pragma config WDTE = OFF

#pragma config PWRTE = OFF

#pragma config BOREN = ON

#pragma config LVP = OFF

#pragma config CPD = OFF

#pragma config WRT = OFF

#pragma config CP = OFF

#include <xc.h>

#define \_XTAL\_FREQ 20000000

#define SBIT\_T2CKPS1 1

```

#define Xung RD0
#define sw RB0
unsigned int count = 0, check = 0;
unsigned char byte = 0;

char UART_Init(const long int baudrate)
{
    unsigned int x;
    x = (_XTAL_FREQ - baudrate*64)/(baudrate*64);
    if(x>255)
    {
        x = (_XTAL_FREQ - baudrate*16)/(baudrate*16);
        BRGH = 1;
    }
    if(x<256)
    {
        SPBRG = x;
        SYNC = 0; //chon che do khong dong bo
        SPEN = 1; // cho phep truyen thong
        TRISC7 = 1;
        TRISC6 = 1;
        CREN = 1; //cho phep nhan bit lien tuc
        TXEN = 1; //cho phep truyen
        return 1;
    }
    return 0;
}

// kiem tra du lieu truyen(tra ve 1 neu du lieu da co con tra ve 0 neu rong)
char UART_TX_Empty()
{
    return TRMT;
}

```

```

}
//Kiem tra du lieu nhan duoc hay khogn
char UART_Data_Ready()
{
    return RCIF;
}

//Doc mot ki tu,doi den khi nhan du 8 bit tu ben gui
char UART_Read()
{
    while(!RCIF);
    return RCREG;
}

//Doc mot chuoi
void UART_Read_Text(char *Output, unsigned int length)
{
    int i;
    for(int i=0;i<length;i++)
        Output[i] = UART_Read();
}

//viet mot ki tu
void UART_Write(char data)
{
    while(!TRMT);
    TXREG = data;
}

//viet mot chuoi
void UART_Write_Text(char *text)

```



```

{
    int i;
    for(i=0;text[i]!='\0';i++)
        UART_Write(text[i]);
}

```

```

void timer2_init(){
    T2CON = (1<<SBIT_T2CKPS1); // prescaler 1:16
    TMR2=225;    // gia tri khoi tao ban dau tao tre 100us
    TMR2IE=1;
    GIE=1;
    PEIE=1;
    TMR2ON = 1;
}

```

//TAo ham ngat voi do tre 100us

```

void __interrupt(high_priority) ngat_timer2(void){
    if(TMR2IF == 1){
        TMR2 = 226;
        TMR2IF = 0;
        count++;
        if(count <= 2){
            Xung = 1;
        }
        else if(count > 2 && count <10){
            Xung = 0;
        }
        if(count == 10){
            count = 0;
        }
        if(sw == 0){
            check = 1;
        }
    }
}

```

```

        if(sw == 1 && check == 1){
            UART_Write(byte);
            check = 0;
        }
    }
}

//code nhan tin hieu cua vxl thu 2
void Nhan(void){
    if(UART_Data_Ready()){
        PORTB = UART_Read();
        __delay_ms(100);
    }
}

void main(void) {
    TRISD = 0x00;
    TRISB = 0xff;
    TRISCbits.TRISC6 = 0;//output(TX)
    TRISCbits.TRISC7 = 1;//input(RX)
    PORTB = 0x00;
    PORTD = 0x00;
    timer2_init();
    UART_Init(2400);
    while(1){

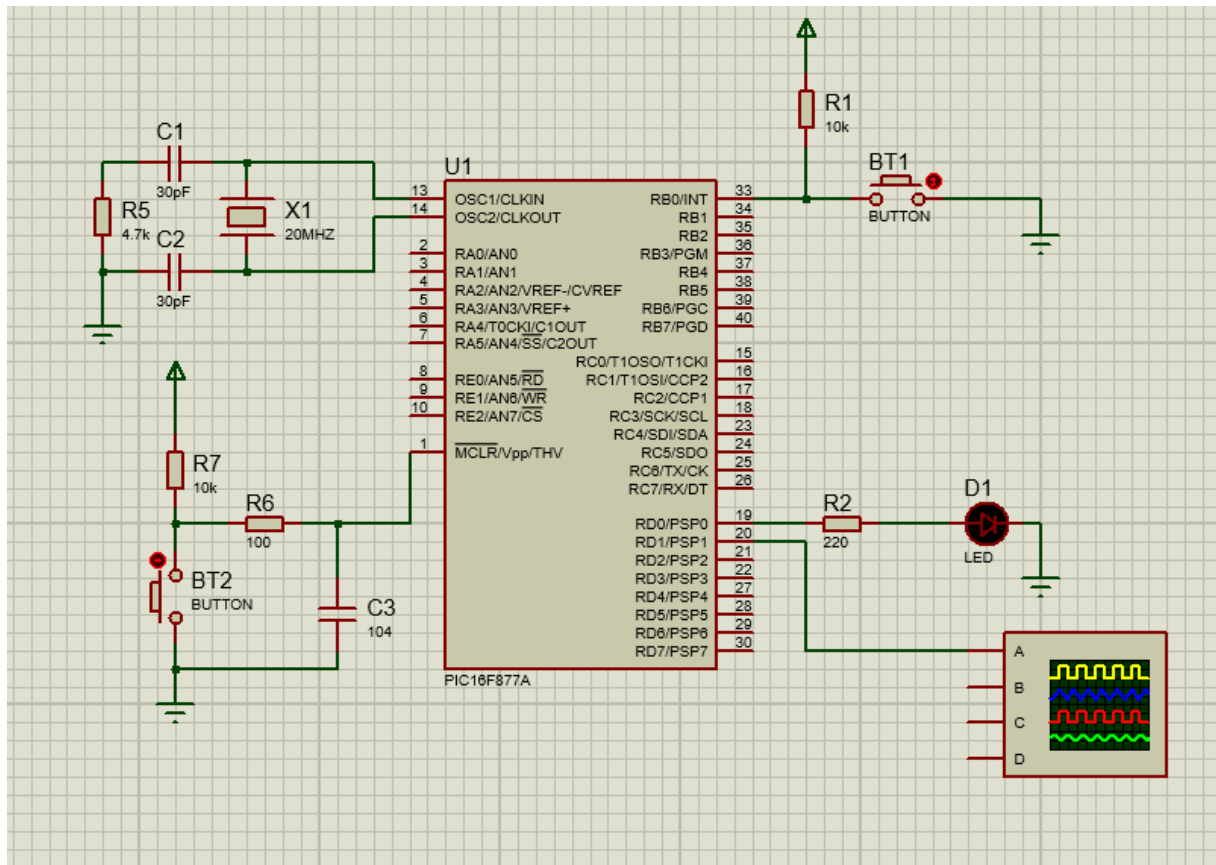
    }
}

```

Câu 3:\*Đề 10:

Xây dựng sơ đồ và lập trình sử dụng vi xử lí thực hiện đồng thời các nhiệm vụ sau:

- Điều khiển 1 Led đơn sang theo phím theo thuật toán: Khi ấn phím Led sang ngay,khi nhả phím sau 2s.... Led tắt.
- Liên tục tạo xung vuông tần số 2 KHz,thời gian tồn tại xung là 0,1ms trên 1 cổng ra của vi xử lí.



/\*

\* File: mainde3.c

\* Author: VU VAN DUY

\* SO 10

\* Created on December 19, 2020, 2:55 AM

\*/

// CONFIG

#pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)

#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)

#pragma config BOREN = ON

#pragma config LVP = OFF

#pragma config CPD = OFF

#pragma config WRT = OFF

#pragma config CP = OFF

```

#include <xc.h>
#define _XTAL_FREQ 20000000
#define sw RB0
#define LED RD0
#define Xung RD1
#define SBIT_T2CKPS1 1
unsigned int check = 0;
unsigned int count = 0, count1 = 0;
//Tao ham ngat timer2 voi do tre 100us = 0.1ms
void __interrupt() nutbam()
{
    if(TMR2IF==1)
    {
        TMR2 = 226;
        TMR2IF=0;
        count1++;
        if(count1 <= 1){
            Xung = 1;
        }
        else if(count1 > 1 && count1 <5){
            Xung = 0;
        }
        if(count1 == 5){
            count1 = 0;
        }
        if(sw == 0)
        {
            LED = 1;
            check = 1;
        }
        if(sw == 1 && check == 1)
        {

```

```

        count++;
        if(count == 20000) // 100us*20000 = 2s
        {
            LED = 0;
            count = 0;
            check = 0;
        }
    }
}

```

```

void timer2_init()
{
    T2CON = (1<<SBIT_T2CKPS1);
    TMR2=225;
    TMR2IE=1;
    GIE=1;
    PEIE=1;
    TMR2ON = 1;
}

```

```

void main(void) {
    timer2_init();
    TRISB = 0xff;
    TRISD = 0x00;
    LED = 0;

    while(1){

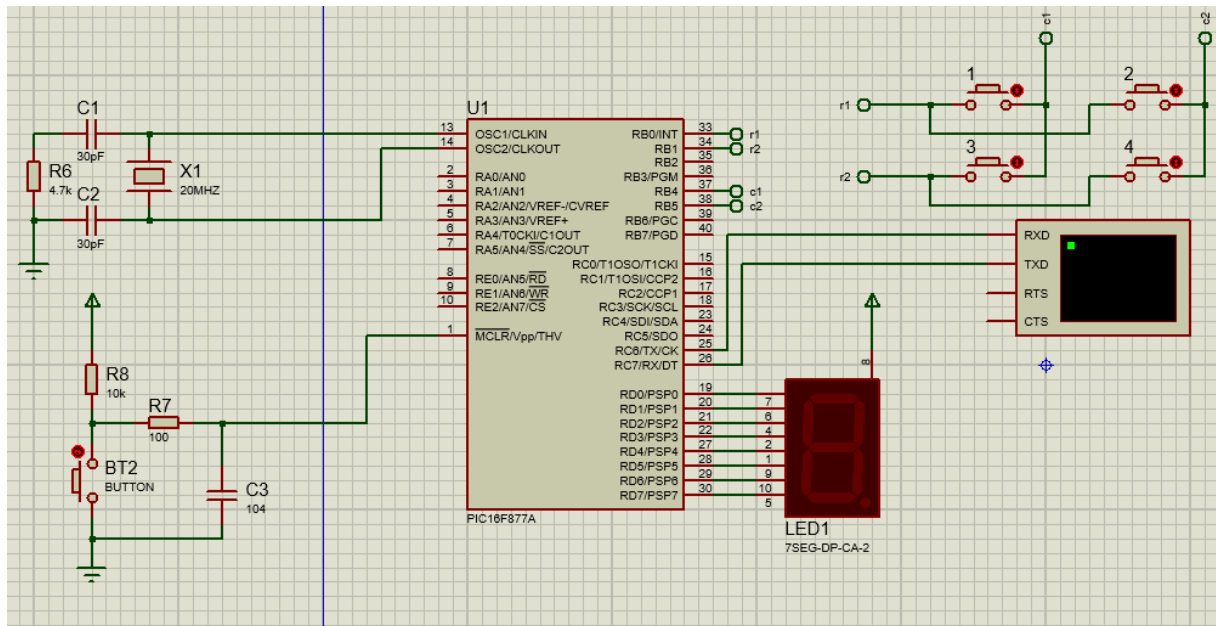
    }
}

```

Câu 3: \*Đề 9

Xây dựng sơ đồ ghép nối 1 LED bảy thanh và 1 ma trận phím 2x2(4 phím tương ứng với các .....sử dụng vi xử lí thực hiện đồng thời các nhiệm vụ sau:

- Hiện thị số thứ tự các phím trên LED 7 thanh đó khi bấm phím tương ứng.
- Truyền thông qua cổng UART tốc độ truyền 4800 b/s kỹ tự có mã ASCII tương ứng với số .....



/\*

\* File: mainde8.c

\* Author: VU VAN DUY

\*SO 9

\* Created on December 23, 2020, 12:50 AM

\*/

// CONFIG

```
#pragma config FOSC = HS
#pragma config WDTE = OFF
#pragma config PWRTE = OFF
#pragma config BOREN = ON
#pragma config LVP = OFF
#pragma config CPD = OFF
#pragma config WRT = OFF
#pragma config CP = OFF
```

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

```
#define _XTAL_FREQ 20000000
```

```
#define r1 RB0
```

```
#define r2 RB1
```

```
#define c1 RB4
```



```

#define c2 RB5
unsigned char LED7[] =
{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
char quet(void);
char UART_Init(const long int baudrate)
{
    unsigned int x;
    x = (_XTAL_FREQ - baudrate*64)/(baudrate*64);
    if(x>255)
    {
        x = (_XTAL_FREQ - baudrate*16)/(baudrate*16);
        BRGH = 1;
    }
    if(x<256)
    {
        SPBRG = x;
        SYNC = 0; //chon che do khong dong bo
        SPEN = 1; // cho phep truyen thong
        TRISC7 = 1;
        TRISC6 = 1;
        CREN = 1; //cho phep nhan bit lien tuc
        TXEN = 1; //cho phep truyen
        return 1;
    }
    return 0;
}

// kiem tra du lieu truyen(tra ve 1 neu du lieu da co con tra ve 0 neu rong)
char UART_TX_Empty()
{
    return TRMT;
}

```

//Kiem tra du lieu nhan duoc hay khogn

char UART\_Data\_Ready()

```
{  
    return RCIF;  
}
```

//Doc mot ki tu,doi den khi nhan du 8 bit tu ben gui

char UART\_Read()

```
{  
  
    while(!RCIF);  
    return RCREG;  
}
```

//Doc mot chuoi

void UART\_Read\_Text(char \*Output, unsigned int length)

```
{  
    int i;  
    for(int i=0;i<length;i++)  
        Output[i] = UART_Read();  
}
```

//viet mot ki tu

void UART\_Write(char data)

```
{  
    while(!TRMT);  
    TXREG = data;  
}
```

//viet mot chuoi

void UART\_Write\_Text(char \*text)

```
{
```

```

int i;
for(i=0;text[i]!='\0';i++)
    UART_Write(text[i]);
}

void keypad_init(){
    PORTB = 0x00;
    TRISB = 0xf0;
    OPTION_REG &= 0x7F; //Bat che do keo len tren portb
}

char key(){
    char k = 'n';
    while(k == 'n')
        k = quet();

    return k;
}

char quet(){
    r1 = 0; r2 = 1;
    if(c1 == 0){
        __delay_ms(100);
        while(c1 == 0);
        return '1';
    }
    if(c2 == 0){
        __delay_ms(100);
        while(c2 == 0);
        return '2';
    }

    r1 = 1; r2 = 0;
    if(c1 == 0){

```

```

        __delay_ms(100);
        while(c1 == 0);
        return '3';
    }
    if(c2 == 0){
        __delay_ms(100);
        while(c2 == 0);
        return '4';
    }
    return 'n';
}

void main(void) {
    TRISD = 0x00;
    PORTD = 0x00;
    PORTD = LED7[0];
    UART_Init(4800);
    keypad_init();
    char k = 'n';
    while(1){
        k = key();
        switch(k){
            case '1':
                PORTD = LED7[1];
                UART_Write(1+0x30);
                break;
            case '2':
                PORTD = LED7[2];
                UART_Write(2+0x30);
                break;
            case '3':
                PORTD = LED7[3];
                UART_Write(3+0x30);

```

```
        break;
    case '4':
        PORTD = LED7[4];
        UART_Write(4+0x30);
        break;
    }

}

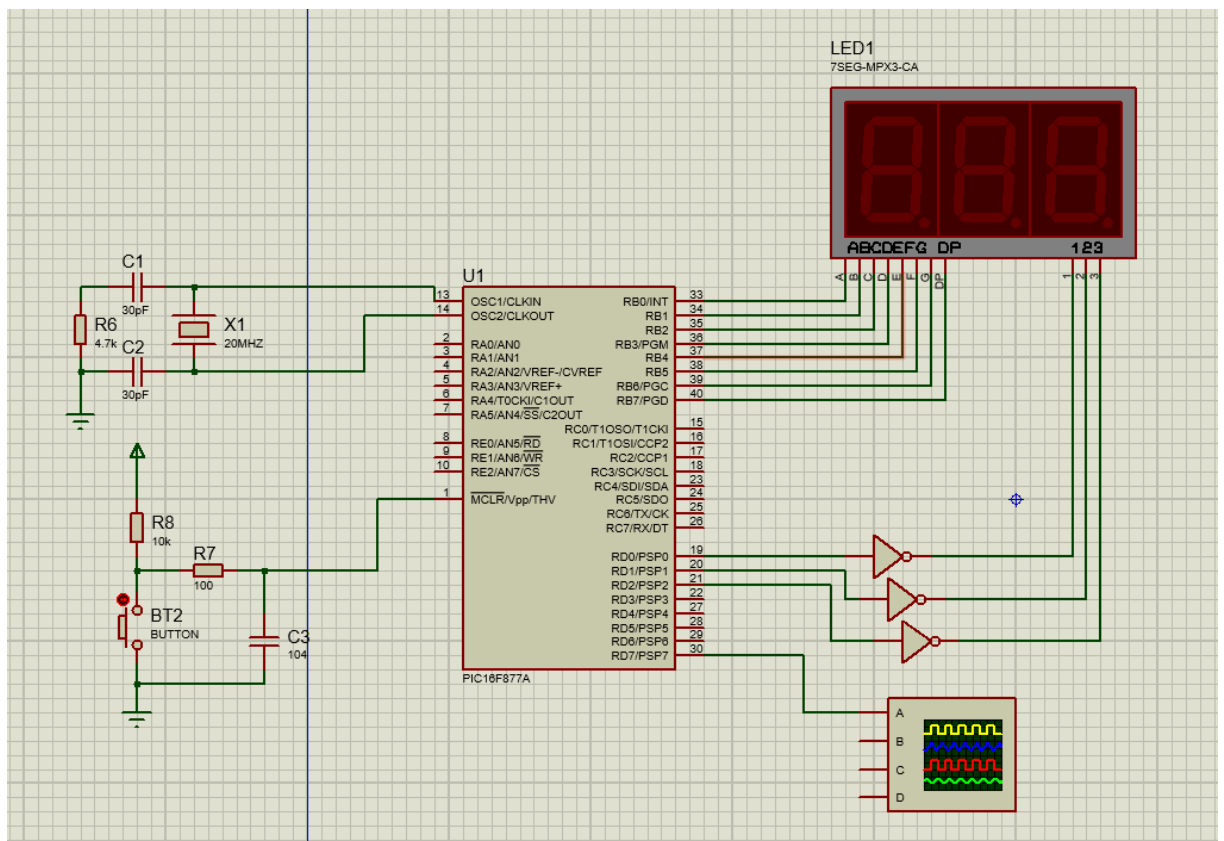
}
```

Câu 3:\*Đề 11:

Xây dựng sơ đồ và lập trình sử dụng vi điều khiển thực hiện đồng thời các nhiệm vụ sau:

-Hiện thị số đếm trên 3 Led 7 thanh,số đếm tăng giá trị từ 0 đến 999 với nhịp tăng 0,1s,khi đếm đến 999 thì giá trị đếm giảm dần về 0 với nhịp 2s.Lặp lại vô tận quá trình trên.

-Liên tục tạo xung vuông tần số 2KHz,thời gian tồn tại xung là 0,1ms trên 1 cổng ra của vi điều khiển.



/\*

\* File: mainde9.c

\* Author: VU VAN DUY

\*SO 11

\* Created on December 23, 2020, 2:43 AM

\*/

```

// CONFIG
#pragma config FOSC = HS
#pragma config WDTE = OFF
#pragma config PWRTE = OFF
#pragma config BOREN = ON
#pragma config LVP = OFF
#pragma config CPD = OFF
#pragma config WRT = OFF
#pragma config CP = OFF

#include <xc.h>
#define _XTAL_FREQ 20000000
#define SBIT_T2CKPS1 1
#define Xung RD7

unsigned int count = 0, count1 = 0, dem = 0;
unsigned int a, b, c;
unsigned int checktang = 1, checkgiam = 0;
const unsigned char MALED[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92,
0x82, 0xF8, 0x80, 0x90};
void timer2_init(){
    T2CON = (1<<SBIT_T2CKPS1);
    TMR2=225;
    TMR2IE=1;
    GIE=1;
    PEIE=1;
    TMR2ON = 1;
}
//tao ham ngat timer2 100us
void __interrupt() timer2(){
    if(TMR2IF == 1){
        TMR2 = 226;
        TMR2IF = 0;
        count++;
        count1++;
    }
}

```

```

        if(count1 <= 1){
            Xung = 1;
        }
        else if(count1 > 1 && count1 <= 4){
            Xung = 0;
        }
        if(count1 == 5){
            count1 = 0;
        }
    }
}

void demtang(unsigned int dem){
    unsigned int a,b,c;
    a = (dem/100);
    b = (dem%100)/10;
    c = dem % 10;

}

void main(void) {

    TRISB = 0x00;
    TRISD = 0x00;
    PORTD = 0x00;
    PORTB = 0x00;
    timer2_init();
    while(1)
    {
        a=(dem/100);
        b=(dem%100)/10;
        c=dem%10;
        PORTDbits.RD0=0; PORTB = MALED[a];
        __delay_ms(5); PORTDbits.RD0=1;
        PORTDbits.RD1=0; PORTB = MALED[b];
        __delay_ms(5); PORTDbits.RD1=1;
    }
}

```



```

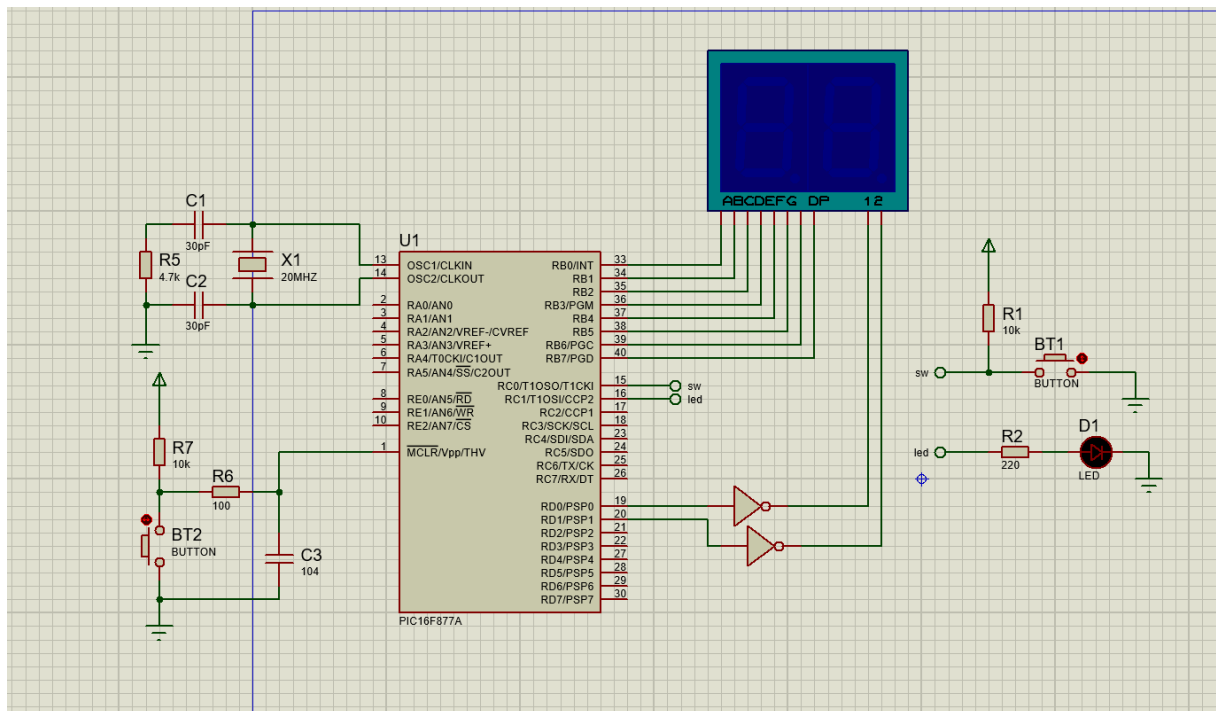
PORTDbits.RD2=0; PORTB = MALED[c];
__delay_ms(5); PORTDbits.RD2=1;
if(count >= 1000&&checktang == 1){ //200*500us = 100ms = 0.1s
    dem++;
    if(dem == 999){
        checktang = 0;
        checkgiam = 1;
    }
    count = 0;
}
if(count >= 20000 && checkgiam == 1){
    dem--;
    if(dem == 0){
        checktang = 1;
        checkgiam = 0;
    }
    count = 0;
}
}
}

```

### Câu 3 \* đề 05

Xây dựng sơ đồ và lập trình sử dụng vi điều khiển đồng thời thực hiện nhiệm vụ sau:

- Hiển thị số đếm trên 2 LED 7 thanh, số đếm tăng giá trị từ 0 đến 99 với nhịp tăng 0,3s, khi đếm đến 99 thì giá trị giảm dần về 0 với nhịp 1s. Lặp lại vô tận quá trình trên.
- Khi ấn 1 phím đầu vào thì sau 0,5s 1 Led đơn sang, khi nhả phím Led tắt.



\* File: maindel2.c

\* Author: VU VAN DUY

\*SO 5

\* Created on December 23, 2020, 2:43 AM

\*/

// CONFIG

#pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)

#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)

#pragma config BOREN = ON // Brown-out Reset Enable bit (BOR enabled)

#pragma config LVP = OFF // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit (RB3 is digital I/O, HV on MCLR must be used for programming)

#pragma config CPD = OFF // Data EEPROM Memory Code Protection bit (Data EEPROM code protection off)

#pragma config WRT = OFF // Flash Program Memory Write Enable bits (Write protection off; all program memory may be written to by EECON control)

#pragma config CP = OFF // Flash Program Memory Code Protection bit (Code protection off)

// #pragma config statements should precede project file includes.

// Use project enums instead of #define for ON and OFF.

#include <xc.h>

#define \_XTAL\_FREQ 20000000

#define SBIT\_T2CKPS1 1

#define sw RC0

#define LED RC1

unsigned int count = 0, count1 = 0, dem = 0;

unsigned int a, b;

unsigned int checktang = 1, checkgiam = 0, check = 0;

```
const unsigned char MALED[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99,0x92,  
0x82,0xF8, 0x80, 0x90};
```

```
void timer2_init(){
```

```
    T2CON = (1<<SBIT_T2CKPS1);
```

```
    TMR2=225;    //Khoi tao gia tri bat dau de tao tre 100us
```

```
    TMR2IE=1;
```

```
    GIE=1;
```

```
    PEIE=1;
```

```
    TMR2ON = 1;
```

```
}
```

```
//tao ham ngat timer2 100us = 0.1ms
```

```
void __interrupt() timer2(){
```

```
    if(TMR2IF == 1){
```

```
        TMR2 = 226;
```

```
        TMR2IF = 0;
```

```
        count++;
```

```
        count1++;
```

```
        if(sw == 0){
```

```
            if(count1 >= 5000){
```

```
                LED = 1;
```

```
                count1 = 0;
```

```
                check = 1;
```

```
            }
```

```
        }
```

```
        if(sw == 1 && check == 1){
```

```
            LED = 0;
```

```
            check = 0;
```

```
        }
```

```
    }
```

```
}
```

```
void demtang(unsigned int dem){
```

```
    unsigned int a,b,c;
```

```

    a = (dem/100);
    b = (dem%100)/10;

}

void main(void) {

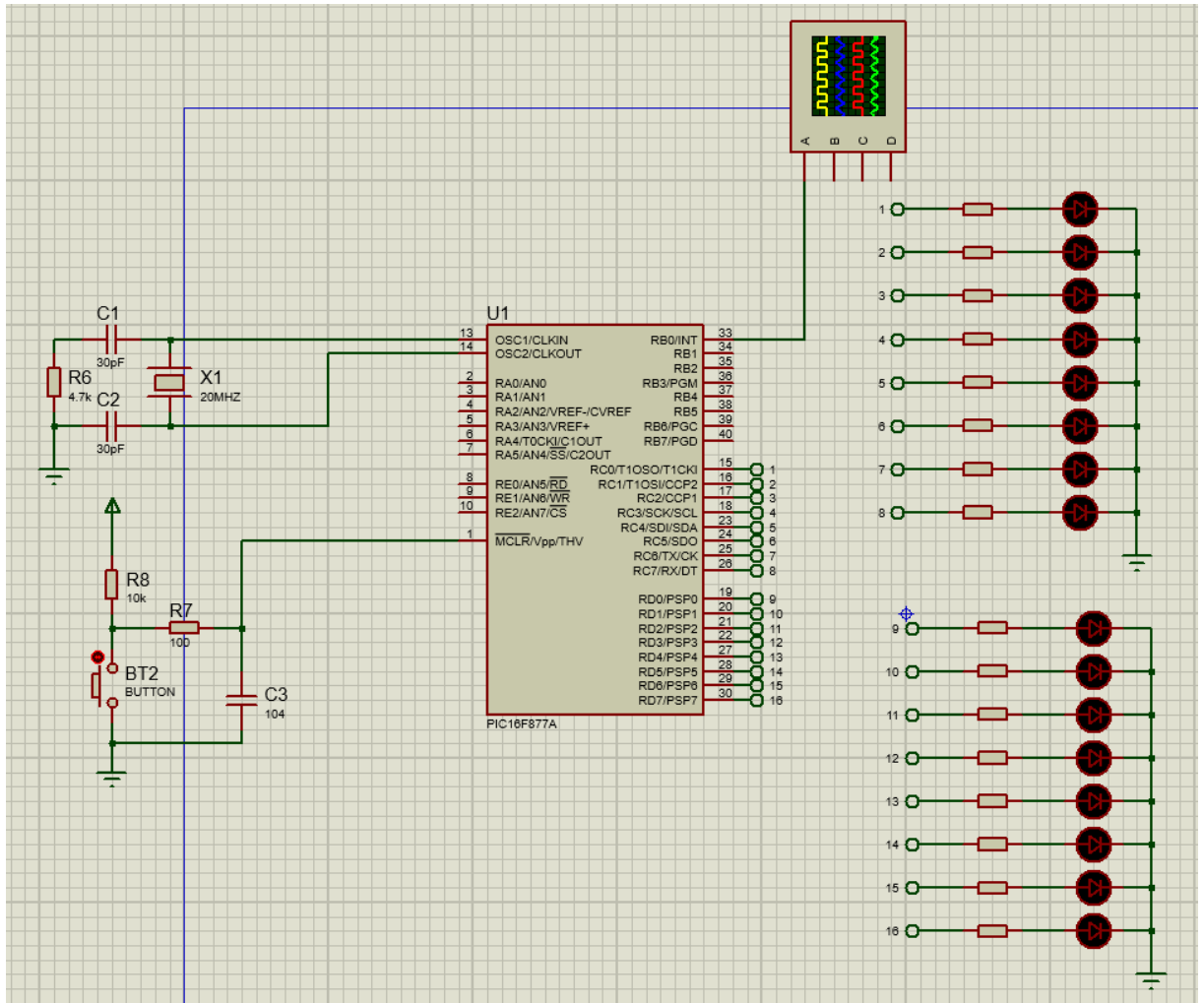
    TRISB = 0x00;
    TRISD = 0x00;
    TRISCbits.TRISC0 = 1;
    TRISCbits.TRISC1 = 0;
    LED = 0;
    PORTD = 0x00;
    PORTB = 0x00;
    timer2_init();
    while(1)
    {
        a=dem/10;
        b=dem%10;
        PORTDbits.RD0=0; PORTB = MALED[a];
        __delay_ms(5); PORTDbits.RD0=1;
        PORTDbits.RD1=0; PORTB = MALED[b];
        __delay_ms(5); PORTDbits.RD1=1;
        if(count >= 3000&&checktang == 1){ //3000*100us = 300ms = 0.3s
            dem++;
            if(dem == 99){
                checktang = 0;
                checkgiam = 1;
            }
            count = 0;
        }
        if(count >= 10000 && checkgiam == 1){ //10000 * 100us = 1s

```

```
    dem--;  
    if(dem == 0){  
        checktang = 1;  
        checkgiam = 0;  
    }  
    count = 0;  
}  
}
```

Câu 3 \* đề 02

Nháy lần lượt 16 LED theo chu kì và tạo xung



/\*

\* File: mainde10.c

\* Author: VU VAN DUY

\*SO 02

\* Created on December 23, 2020, 12:33 PM

\*/

// CONFIG

#pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)

#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)

```

#pragma config BOREN = ON      // Brown-out Reset Enable bit (BOR
enabled)
#pragma config LVP = OFF      // Low-Voltage (Single-Supply) In-Circuit
Serial Programming Enable bit (RB3 is digital I/O, HV on MCLR must be
used for programming)
#pragma config CPD = OFF      // Data EEPROM Memory Code Protection
bit (Data EEPROM code protection off)
#pragma config WRT = OFF      // Flash Program Memory Write Enable bits
(Write protection off; all program memory may be written to by EECON
control)
#pragma config CP = OFF      // Flash Program Memory Code Protection
bit (Code protection off)

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

```

```

#include <xc.h>
#define _XTAL_FREQ 20000000
#define LED1 PORTC
#define LED2 PORTD
#define Xung RB0
unsigned int count = 0, count1 = 0, count2 = 0, i = 0, check = 0;
unsigned char LED[6] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf};

```

```

void delay_timer0_5ms(unsigned int time)
{
    while (time-- > 0) //1? p 1? i time 1? n 5ms.
    {
        OPTION_REGbits.T0CS = 0; //Dùng xung n? i
        OPTION_REGbits.PSA = 0; //Dùng b? chia timer0
        OPTION_REGbits.PS2 = 1;
        OPTION_REGbits.PS1 = 1;
    }
}

```



```

OPTION_REGbits.PS0 = 0; //Chọn b? chia 1:128
TMR0 = 60; //?t giá tr? cho TMR0
while(!INTCONbits.TMR0IF); //?i c? tràn
INTCONbits.TMR0IF = 0; //Xóa c? tràn
}
}

```

```

void timer2_init(){
    T2CON = 0b00000100;
    TMR2=231;
    TMR2IE=1;
    GIE=1;
    PEIE=1;
}

//Tạo hàm ngắt timer2 5us
void __interrupt() timer2(){
    if(TMR2IF == 1){
        TMR2 = 232;
        TMR2IF = 0;
        count++;
        count1++;
        if(count <= 5){
            Xung = 1;
        }
        else if(count > 5 && count < 10){
            Xung = 0;
        }
        if(count == 10){
            count = 0;
        }
        if(count1 == 50000&&check == 0){
            i++;

```

```

        PORTC =0X01<<i;
        PORTD =0X80>>i;
        count1 = 0;
        if(i == 8){
            i = 0;
            check = 1;
        }
    }
    if(count2 == 100000&&check == 1){
        i++;
        PORTC =0X01<<i;
        PORTD =0X01<<i;
        count2 = 0;
        if(i == 8){
            i = 0;
            check = 0;
        }
    }
}

void main(void) {
    TRISB = 0x00;
    TRISC = 0x00;
    TRISD = 0x00;
    PORTB = 0x00;
    PORTD = 0x00;
    PORTC = 0x00;
    timer2_init();
    while(1){
    }
}

```

