

FACULTAD DE INGENIERÍA, UNAM

LABORATORIO DE MICROCOMPUTADORASS

SEMESTRE 2023-2

GRUPO 11

PREVIO PRÁCTICA 4

PUERTOS PARALELOS E/S

NOMBRE DEL ALUMNO:

ARRIAGA MEJÍA JOSÉ CARLOS

PROFESOR

ING. ROMAN V. OSORIO COMPARAN

FECHA DE ENTREGA: 24 DE MARZO DE 2023

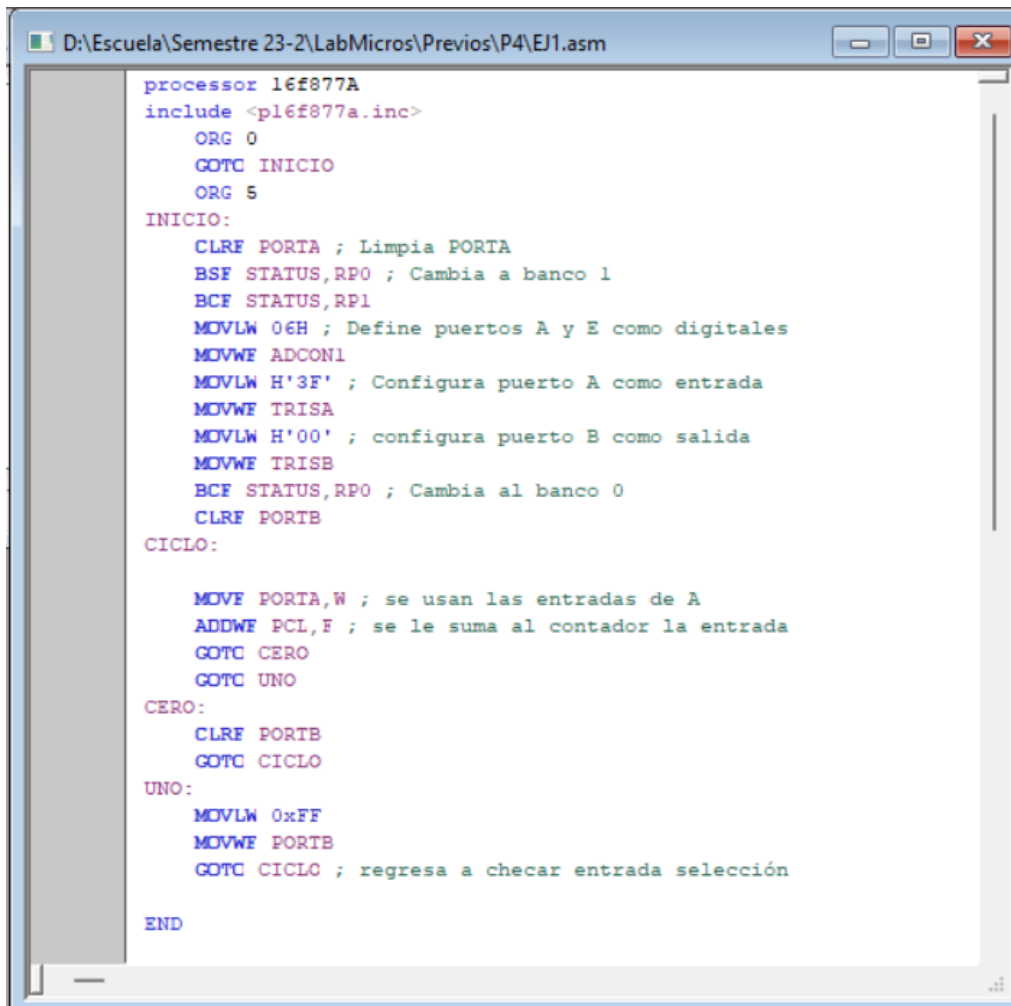
CALIFICACION

Objetivo

Emplear los puertos paralelos que contiene un microcontrolador para realizar funciones de control, configurando estos como entrada y salida.

1.- Empleando dos puertos paralelos del microcontrolador PIC, uno de ellos configurado como entrada y el otro como salida; realizar un programa que de acuerdo al valor del bit menos significativo del puerto A, se genere la acción indicada en el puerto B.

Valor PA0	Acción puerto B
0	00000000
1	11111111



```
processor 16f877A
include <pl6f877a.inc>
ORG 0
GOTO INICIO
ORG 5
INICIO:
    CLRF PORTA ; Limpia PORTA
    BSF STATUS,RP0 ; Cambia a banco 1
    BCF STATUS,RP1
    MOVLW 06H ; Define puertos A y E como digitales
    MOVWF ADCON1
    MOVLW H'3F' ; Configura puerto A como entrada
    MOVWF TRISA
    MOVLW H'00' ; configura puerto B como salida
    MOVWF TRISB
    BCF STATUS,RP0 ; Cambia al banco 0
    CLRF PORTB
CICLO:
    MOVE PORTA,W ; se usan las entradas de A
    ADDWF PCL,F ; se le suma al contador la entrada
    GOTO CERO
    GOTO UNO
CERO:
    CLRF PORTB
    GOTO CICLO
UNO:
    MOVLW 0xFF
    MOVWF PORTB
    GOTO CICLO ; regresa a checar entrada selección
END
```

Lo que hace nuestro código es configurar el puerto A como entrada y el puerto B como salida, después lo que se hace es tomar el valor que tiene nuestra entrada y colocarlo en W, luego el valor de W se suma a PCL que es el apuntador de la instrucción que se está realizando. En este caso, suma 0 o 1, si es 0 realiza la instrucción siguiente, si es uno se salta una línea y realiza la otra instrucción. En la etiqueta CERO se mantiene los leds apagados y en la etiqueta UNO se prenden los leds.

[illegible]

2.- Escribir un programa, el cuál realice las siguientes acciones de control indicadas, para lo cual requiere trabajar un puerto de entrada y otro puerto de salida, usar los sugeridos en el ejercicio anterior; generar retardos de ½ seg., en las secuencias que lo requieran.

DATO PUERTO A	ACCION PUERTO B	Ejecución
0x00	Todos los leds apagados	00000000
0x01	Todos los leds encendidos	11111111
0x02	Corrimiento del bit más significativo hacia la derecha	10000000 01000000 00100000 00000001
0x03	Corrimiento del bit menos significativo hacia la izquierda	00000001 00000010 00000100 10000000
0x04	Corrimiento del bit más significativo hacia la derecha y a la izquierda	10000000 01000000 00000001 00000010 10000000
0x05	Apagar y encender todos los bits.	00000000 11111111

```

processor 16f877a
include <pl6f877a.inc>
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
cte1 equ 0x80
cte2 equ 50h
cte3 equ 60h

ORG 0
GOTO INICIO
ORG 5
INICIO:
    CLRF PORTA ; Limpia PORTA
    BSF STATUS,RP0 ; Cambia a banco 1
    BCF STATUS,RP1
    MOVLW 06H ; Define puertos A y E como digitales
    MOVWF ADCON1
    MOVLW H'3F' ; Configura puerto A como entrada
    MOVWF TRISA
    MOVLW H'00' ; configura puerto B como salida
    MOVWF TRISB
    BCF STATUS,RP0 ; Cambia al banco 0
    CLRF PORTB
CICLO:
    MOVF PORTA,W ; se usan las entradas de A
    ADDWF PCL,F ; se le suma al contador la entrada
    goto CERO ; lo que nos lleva a una función.
    goto UNO
    goto DOS
    goto TRES
    goto CUATRO
    goto CINCO
    goto CERO ;Se repite para evitar que se salte a otra instrucción
    goto CERO ; y realice otro cosa que no queremos
CERO: ; todos apagados
    CLRF PORTB
    goto CICLO ; regresa a checar entrada selección
UNO: ; todos encendidos
    MOVLW 0xFF
    MOVWF PORTB
    goto CICLO ; regresa a checar entrada selección

DOS: ; corrimiento derecha
    BCF STATUS,C ; Limpia carry
    MOVLW 0x80 ; W <- 0x80
    MOVWF PORTB
    CALL retardo
DOS_CERO:
    BTFSC PORTB,0 ; checa bit 0
    GOTO CICLO ; regresa a checar entrada selección
    RRF PORTB,1
    CALL retardo
    goto DOS_CERO
TRES: ; corrimiento izquierda
    BCF STATUS,C ; Limpia carry
    MOVLW 0x01 ; W <- 0x01
    MOVWF PORTB
    CALL retardo
TRES_CERO:
    BTFSC PORTB,7 ; checa bit 7
    GOTO CICLO ; regresa a checar entrada selección
    RLF PORTB,1
    CALL retardo
    goto TRES_CERO

```

```

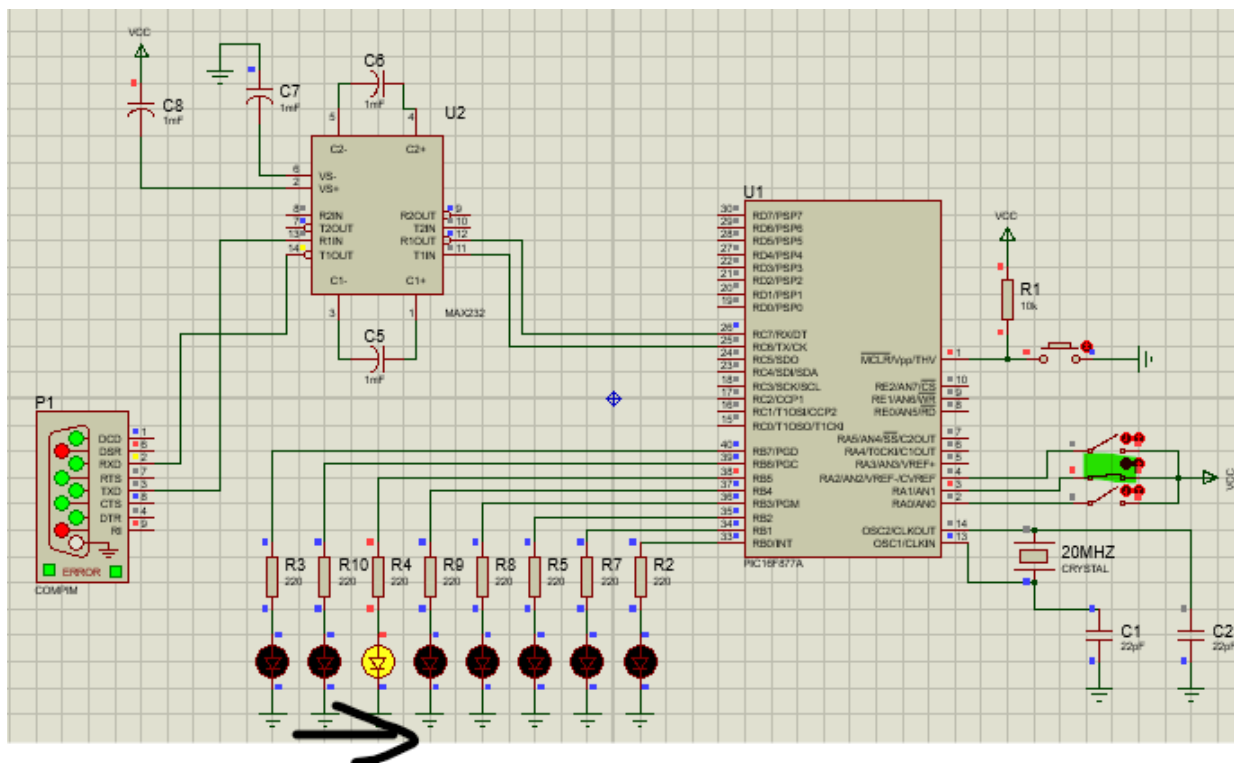
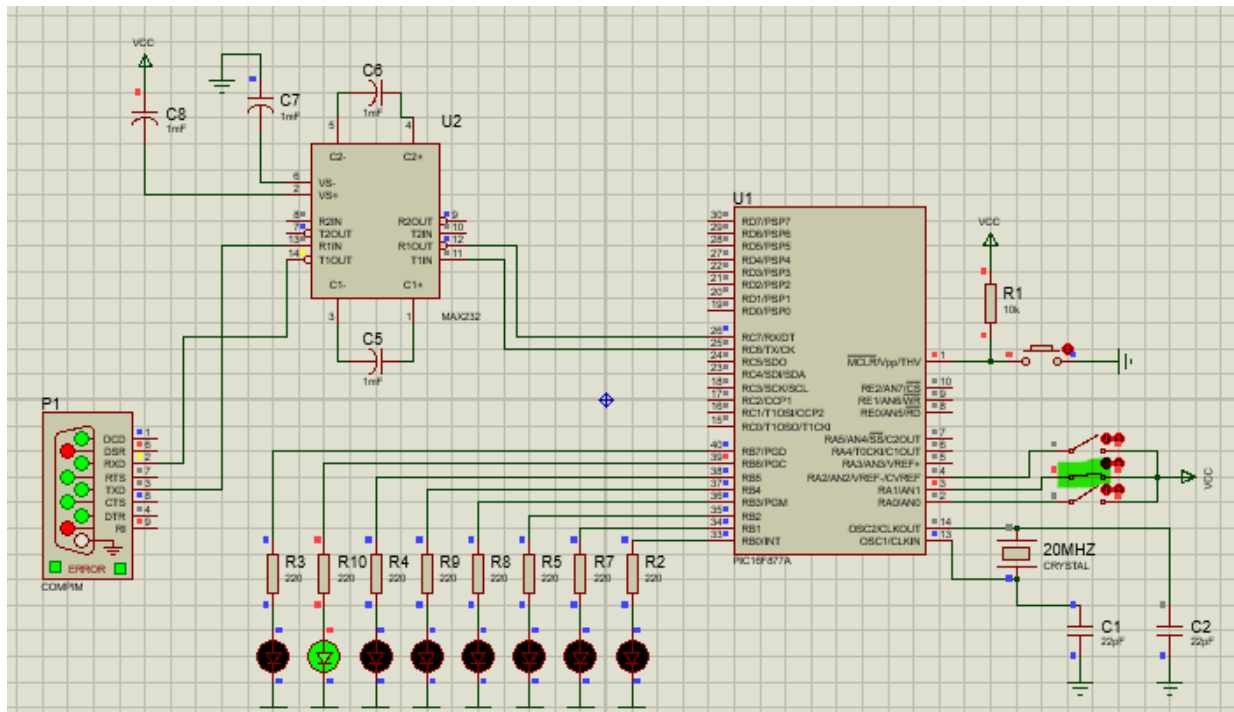
CUATRO: ; corrimiento derecha-izquierda
    BCF STATUS,C ; Limpia carry
    MOVLW 0x80 ; W <- 0x01
    MOVWF PORTB
    CALL retardo
CUATRO_CERO: ; derecha
    BTFSC PORTB,0 ; checa bit 0
    GOTO CUATRO_REG ; cambia corrimiento
    RRF PORTB,1
    CALL retardo
    goto CUATRO_CERO
CUATRO_REG: ; izquierda
    BTFSC PORTB,7 ; checa bit 7
    GOTO CICLO ; regresa a checar entrada selección
    RLF PORTB,1
    CALL retardo
    goto CUATRO_REG
CINCO: ; apagado-encendido
    CLRF PORTB ; apagado
    call retardo
    MOVLW 0xFF
    MOVWF PORTB ; encendido
    call retardo
    goto CICLO ; regresa a checar entrada selección
retardo ; subrutina de retardo
    MOVLW ctel
    MOVWF valor1
tres
    MOVLW cte2
    MOVWF valor2
dos
    MOVLW cte3
    MOVWF valor3
uno
    DECFSZ valor3
    GOTO uno
    DECFSZ valor2
    GOTO dos
    DECFSZ valor1
    GOTO tres
    RETURN
    END

```

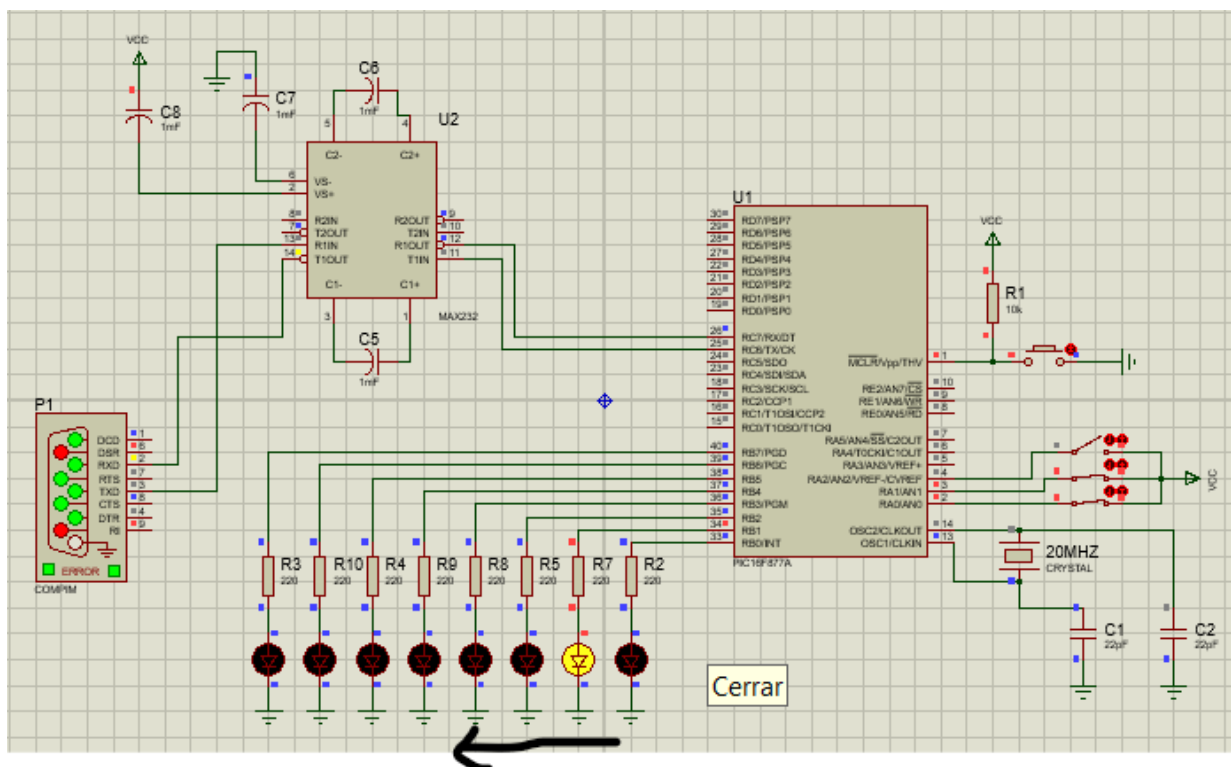
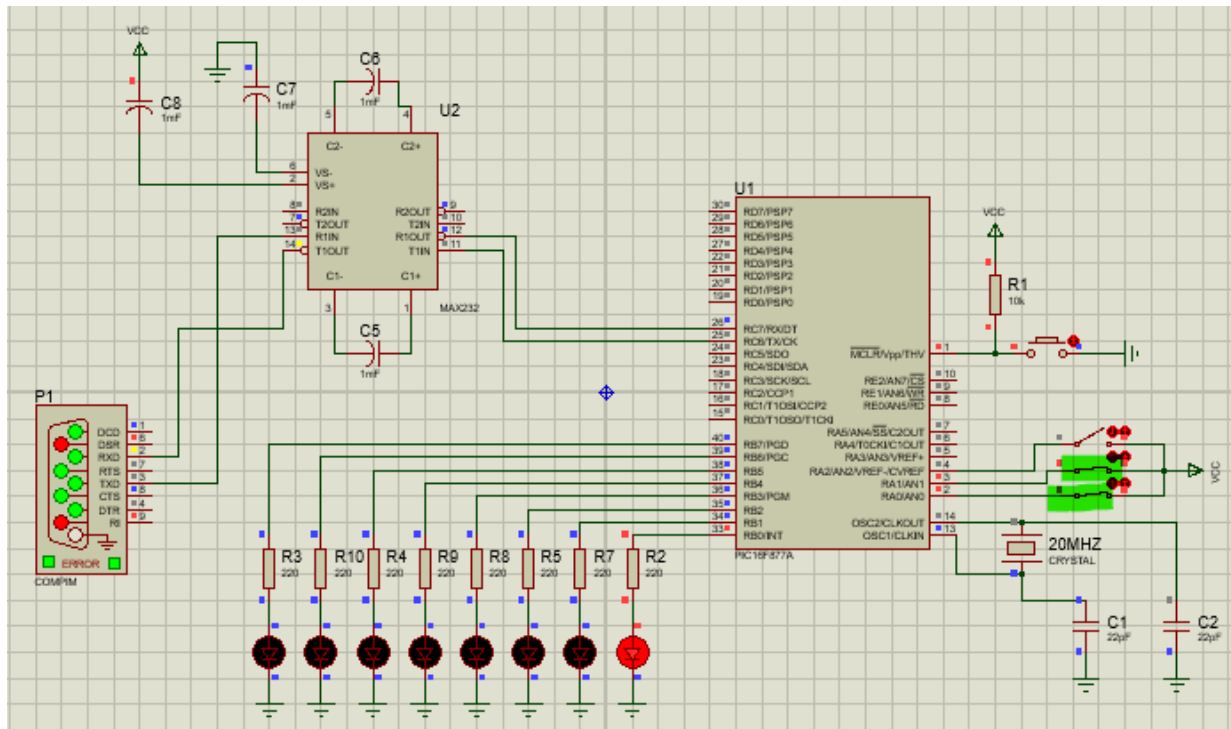
Para este ejercicio reusamos código de la practica anterior ya que los realizamos por separado, este ejercicio nos pide que hagamos un tipo de flujo de datos (CASE o SWITCH conocidos así en otros lenguajes), usamos el mismo concepto que el ejercicio, solamente que ahora usamos 3 bits del puerto A que nos ayudara a realizar las combinaciones de los valores del 0 al 5, pero tenemos que agregar 2 instrucciones que manda al 0x00 para evitar que se salte instrucciones de más y realice algo que no se solicitó.

The circuit diagram illustrates the connection between a PIC16F877A microcontroller (U1) and a MAX232 RS-485 transceiver (U2). The PIC16F877A is configured with various pins connected to power supply rails (VCC and GND), a crystal oscillator (20MHz), and a reset button (R1). The MAX232 is powered by VCC and GND, with its input/output pins connected to the PIC's serial communication pins (RD7/PSP7, RD6/PSPI6, etc.). The PIC's I/O pins are also connected to a series of LEDs (R2-R10) through current-limiting resistors (R3-R10).

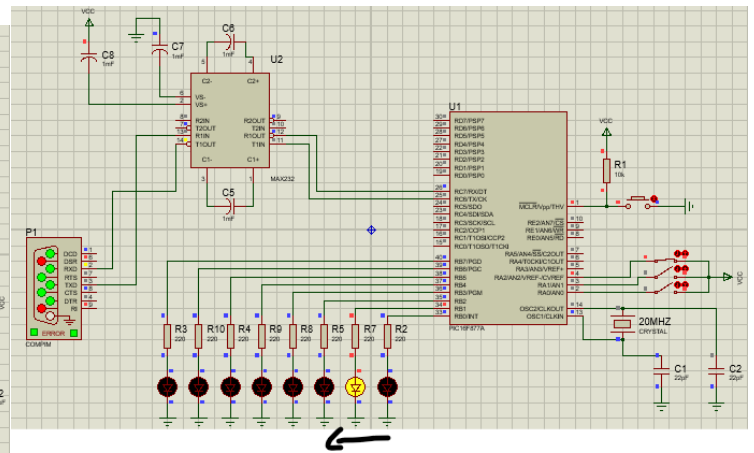
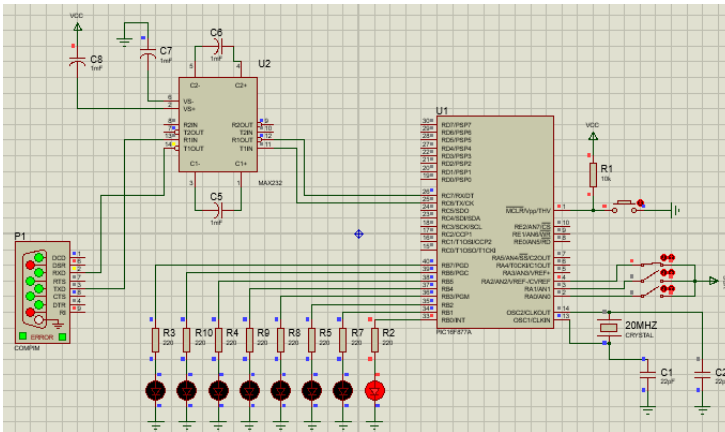
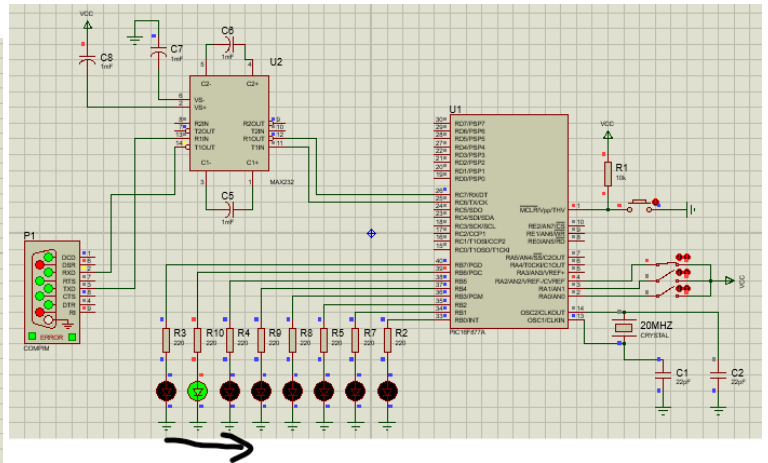
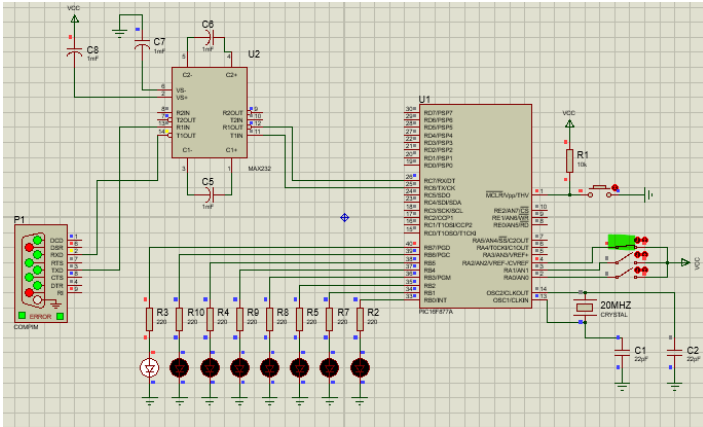
Para 0x02



Para 0x03



Para 0x04



Para 0x5

