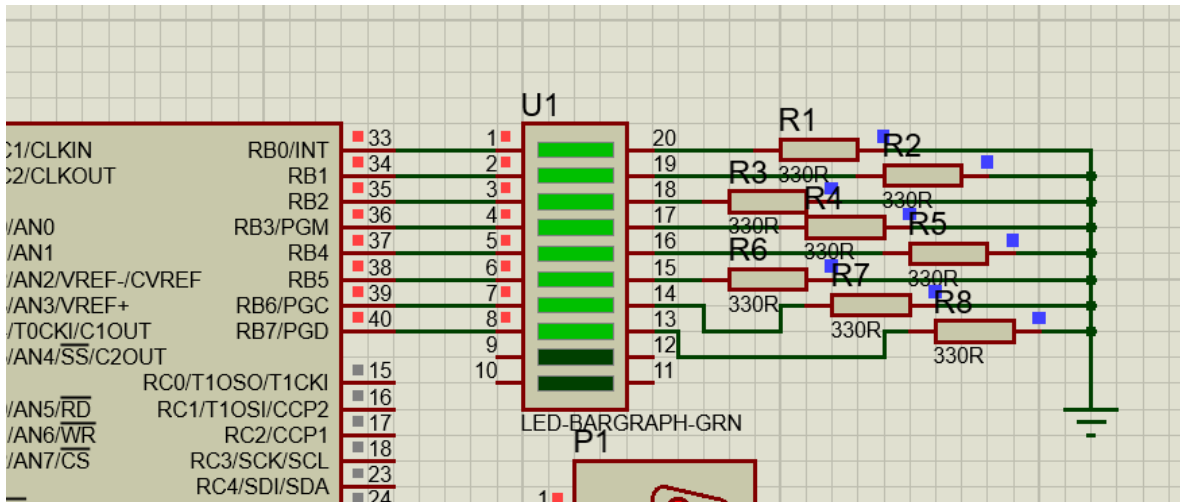


Tarea1



FUNCIONAMIENTO DEL PROGRA

El programa hace el complemento A1 del puerto B para que se prendan o apaguen las luces cada que se desborda el temporizador que corre a una división de 128 del reloj del sistema.

Codigo asm comentado

```

processor 16f877
include<p16f877.inc>

Contador equ 0x20
ORG 0
GOTO Inicio ;Vector de Reset
ORG 4
GOTO Interrupciones ;vector de interrupcion

ORG 5 ;inicio del programa
Inicio: BSF STATUS,5 ;ponemos 1 en el bit 5 del reg status
BCF STATUS,6 ;ponemos 0 en bit 6 del reg status estamos en el banco 01 de memoria
CLRF TRISB ;colocamos el reg b como salida
MOVLW B'00000111' ;cargamos el dato a w
MOVWF OPTION_REG ;colocamos w en el reg Option_Reg, poeniendo ps2:0 en alto
;haciendo la division de frecuencia a 256

BCF STATUS,5 ;regresamos al banco 00
BCF INTCON,TOIF ;ponemos la bandera del timer0 en 0
BSF INTCON,TOIE ;habilitamos la interrupcion por desbordamiento de timer 0
BSF INTCON,GIE ;Habilitamos la interrupciones generales
CLRF PORTB ;limpiamos el puerto b
CLRF Contador ;limpiamos el reg 0x20
GOTO $

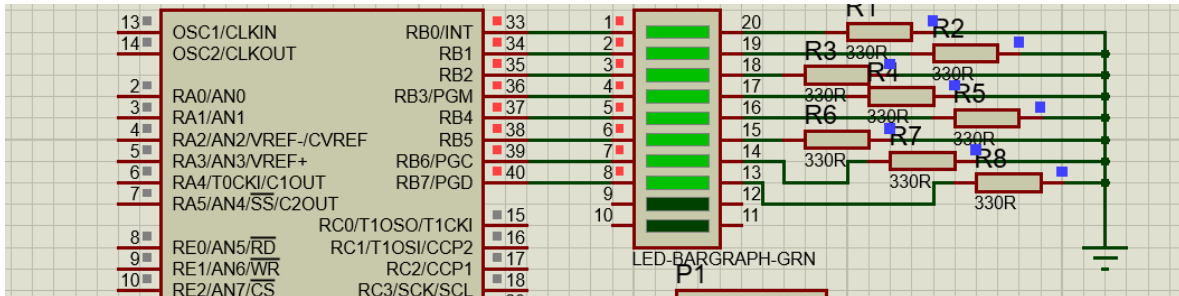
Interrupciones: BTFSS INTCON,TOIF;cheamos la interrupcion si fue por desbordamiento del timer0
GOTO SAL_INT ;en caso de que no sea nos salimos de la interrupcion de forma directa
COMF PORTB ;se hace el complemento A1 del reg (prende y apaga)
BCF INTCON,TOIF ;limpia la bandera para que pueda volver a ocurrir el desbordamiento
SAL_INT: RETFIE ;nos regresamos de la interrupcion
END

```

El puerto B cambia su estado conforme se mueve el tiempo sin necesidad de colocar este procedimiento dentro del cuerpo principal del programa, incluso el programa llega a su fin, pero la interrupción sigue ocurriendo mientras que el micro siga vivo.

Cambia sus valores cada 0.02 segundos aproximadamente, según la fórmula matemática ocurre cada 0.013 seg

Tarea2



Se prende y apagan los led que están puestos con el puerto B a un paso diferente que con el ejercicio anterior dado que primero checamos si se desbordo el timer0, cada que se desborda se incrementa en 1 el contador, si el contador es igual a 150 (las veces que se ha desbordado el timer0) hacemos la operación de complemento al puerto B y reiniciamos el contador, sino solo apagamos la bandera del desborde (para que pueda volver a ocurrir) y regresamos.

Funcionalmente hace lo mismo que el programa anterior, pero, con la adición del contador nos permite incrementar aun más la cantidad del tiempo que pasa entre cada que hacemos nuestro procedimiento.

El programa prende y apaga los leds cada 2seg, según la formula matemática tarda 1.966 seg en ocurrir el cambio

Codigo asm comentado

```
processor 16f877
include<p16f877.inc>
Contador equ 0x20
ORG 0
GOTO Inicio ;Vector de Reset
ORG 4
GOTO Interrupciones ;vector de interrupcion

ORG 5 ;inicio del programa
Inicio: BSF STATUS,5 ;ponemos 1 en el bit 5 del reg status
BCF STATUS,6 ;ponemos 0 en bit 6 del reg status estamos en el banco 01 de memoria
CLRF TRISB ;colocamos el reg b como salida
MOVLW B'00000111' ;cargamos el dato a w
MOVWF OPTION_REG ;colocamos w en el reg Option_Reg, poeniendo ps2:0 en alto
;haciendo la division de frecuencia a 256
BCF STATUS,5 ;regresamos al banco 00
BCF INTCON,TOIF ;ponemos la bandera del timer0 en 0
BSF INTCON,TOIE ;habilitamos la interrupcion por desbordamiento de timer 0
```

```

BSF INTCON,GIE          ;Habilitamos la interrupciones generales
CLRF PORTB              ;limpiamos el puerto b
CLRF Contador           ;limpiamos el reg 0x20
GOTO $

Interrupciones:  BTFSS INTCON,TOIF;chechamos la interrupcion si fue por desbordamiento del timer0
GOTO SAL_NO_FUE_TMR0    ;en caso de que no sea NOS VAMOS A OTRA ETIQUETA
INCF Contador           ;incrementamos el registro contador
MOVLW D'150'            ;PASAMOS EL DATO A W
SUBWF Contador,W         ;Hacemos la resta Contador - w y la guardamos en w
BTFSS STATUS,Z          ;CHECAMOS LA BANDERA Z ES DECIR SI EL RESULTADO FUE 0
GOTO SAL_INT            ;NOS SALTAMOS A SAL_INT
COMF PORTB              ;aqui saltamos cuando es 0 la resta se hace el complemento A1 del reg
(prinde y apaga)
CLRF Contador           ;limpiamos el registro contador
SAL_INT: BCF INTCON,TOIF ;limpia la bandera para que pueda volver a ocurrir el desbordamiento
SAL_NO_FUE_TMR0: RETFIE  ;nos regresamos de la interrupcion
END

```

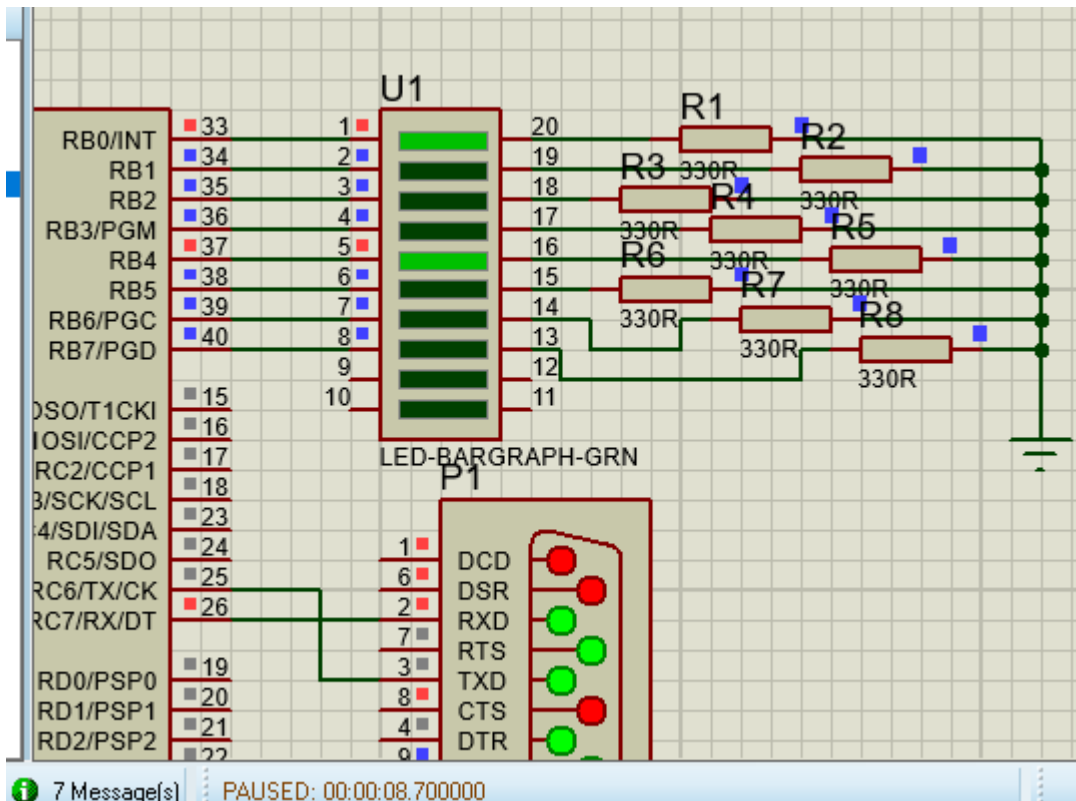
Tarea3

Realizar un contador binario (puertoB) usando el timer0 con cambios de estado cada .5seg

Para poner el cambio cada 0.5 seg es necesario encontrar el valor que se le tiene que poner al registro TMR0

Como sabemos que tiempo es 0.5 es lo que queremos obtener y que este valor se obtiene de $T(PRE)(256-VALOR_INICIAL)$ despejando para obtener valor_inicial nos da un resultado de 720 el cual es un valor mayor de 8 bits, por lo que nos indica que debemos usar un contador auxiliar (como en el ejercicio 2) que nos ayude a ampliar el tiempo de ejecución

Si tenemos que el máximo de tiempo que tarda en desbordarse TMR0 es de .013 seg podemos hacer una división del tiempo buscado/tiempo de desborde lo que nos da el valor de 38.14 ahora como no podemos utilizar decimales se debe de utilizar un valor mayor en este caso 39, lo que nos da que ocurre el cambio cada 0.5111808 segundos lo más cercano posible



Contador binario en 17 a los 8.7 segundos (17 pasos de .5 seg)

Codigo Asm comentado

```

processor 16f877
include<p16f877.inc>

Contador equ 0x20
Binario equ 0x21

ORG 0
GOTO Inicio                ;Vector de Reset
ORG 4
GOTO Interrupciones        ;vector de interrupcion

ORG 5                      ;inicio del programa
Inicio: BSF STATUS,5        ;ponemos 1 en el bit 5 del reg status
        BCF STATUS,6        ;ponemos 0 en bit 6 del reg status estamos en el banco 01 de memoria
        CLRF TRISB          ;colocamos el reg b como salida
        MOVLW B'00000111'   ;cargamos el dato a w
        MOVWF OPTION_REG    ;colocamos w en el reg Option_Reg, poeniendo ps2:0 en alto
                                ;haciendo la division de frecuencia a 256

        BCF STATUS,5        ;regresamos al banco 00
        BCF INTCON,TOIF     ;ponemos la bandera del timer0 en 0
        BSF INTCON,TOIE     ;habilitamos la interrupcion por desbordamiento de timer 0
        BSF INTCON,GIE      ;Habilitamos la interrupciones generales
        CLRF PORTB          ;limpiamos el puerto b
        CLRF Contador       ;limpiamos el reg 0x20
        CLRF Binario        ;limpiamos el registro que funge como nuestro contador binario
        GOTO $

```

```

Interrupciones:   BTFSS INTCON,T0IF;cheamos la interrupcion si fue por desbordamiento del timer0
                  GOTO SAL_NO_FUE_TMRO           ;en caso de que no sea NOS VAMOS A OTRA ETIQUETA
                  INCF Contador                   ;incrementamos el registro contador
                  MOVLW D'39'                     ;PASAMOS EL DATO A W
                  SUBWF Contador,W                 ;Hacemos la resta Contador - w y la guardamos en w
                  BTFSS STATUS,Z                 ;CHECAMOS LA BANDERA Z ES DECIR SI EL RESULTADO FUE 0
                  GOTO SAL_INT                     ;NOS SALTAMOS A SAL_INT
                  INCF Binario                     ;aqui saltamos cuando es 0 se incrementa el contador binario
                  MOVF Binario,W                   ;w tiene el valor del contador binario
                  MOVWF PORTB                     ;portb tiene el valor del contador binario
                  CLRF Contador                   ;limpiamos el registro contador
SAL_INT: BCF INTCON,T0IF                         ;limpia la bandera para que pueda volver a ocurrir el desbordamiento
SAL_NO_FUE_TMRO: RETFIE                          ;nos regresamos de la interrupcion
                  END

```