



INSTITUTO POLITECNICO NACIONAL
SECRETARIA ACADEMICA
DIRECCION DE EDUCACION MEDIA SUPERIOR



CENTRO DE ESTUDIOS CIENTIFICOS Y TECNOLOGICOS num. 9
“JUAN DE DIOS BATIZ ”

Unidad de aprendizaje: MICROELECTRÓNICA PROGRAMABLE.	Turno: MATUTINO
Nivel: 6°	Periodo: Primer Cuestionario Departamental
Especialidad o área: SISTEMAS DIGITALES.	Ciclo escolar: 2020-2021 “B”.
Fecha del examen: 24 de Marzo de 2021.	Contenido a evaluar: Unidades I y II
Horario del examen: 11:00 a 12:50 hrs.	Duración del examen: Dos horas

Tipo de examen: **Único**

Calificación:

Alumno Morales Martínez José Antonio Boleta 2019090265 Firma _____
Profesor de esta asignatura en este grupo: Jesús Alberto Olivares Vargas Grupo 6IM2

I.- Sección Primera. **PARTE A.**

INSTRUCCIONES

CONTESTE EN EL PARENTESIS LA LETRA DE LA RESPUESTA CORRECTA.

- 1.- UN BIT DEL REGISTRO OPTION_REG ES:.....(**C**)
A) RCIE . B) SSPIE.
C) PSA. D) TMR2IE.
- 2.- EL BIT PARA HABILITAR LA INTERRUPCIÓN DEL TIMER 0 ES:.....(**D**)
A) PEIE. B) RBIE.
C) ADIE. D) T0IE.
- 3.- EL VALOR MAXIMO QUE PUEDE TOMAR EL PRESCALADOR PARA EL TIMER 0 ES:.....(**B**)
A) 2. B) 256.
C) 8. D) 512.
- 4.- EL VALOR MAXIMO DE TIEMPO DE INTERRUPCIÓN GENERADO POR EL TIMER 0 ES:.....(**B**)
A) 65.5 s. B) 65.5 ms.
C) 6.5 s. D) 6.5 ms.
- 5.- EL BIT T0IF ES UN BIT DE :.....(**B**)
A) ESTADO ALEATORIO. B) BANDERAS.
C) HABILITACION. D) PRUEBA DE UN PUERTO.
- 6.-LA SUBROUTINA DE ATENCIÓN A LA INTERRUPCIÓN SE LOCALIZA EN LA DIRECCION:.....(**C**)
A) 0X0002 B) 0X0008
C) 0X0004 D) 0X0016
- 7.- LA INSTRUCCIÓN PARA REGRESAR DE UNA SUBROUTINA DE INTERRUPCIÓN ES:.....(**A**)
A) RETFIE. B) RETURN.
C) MOVLW K. D) RETLW.

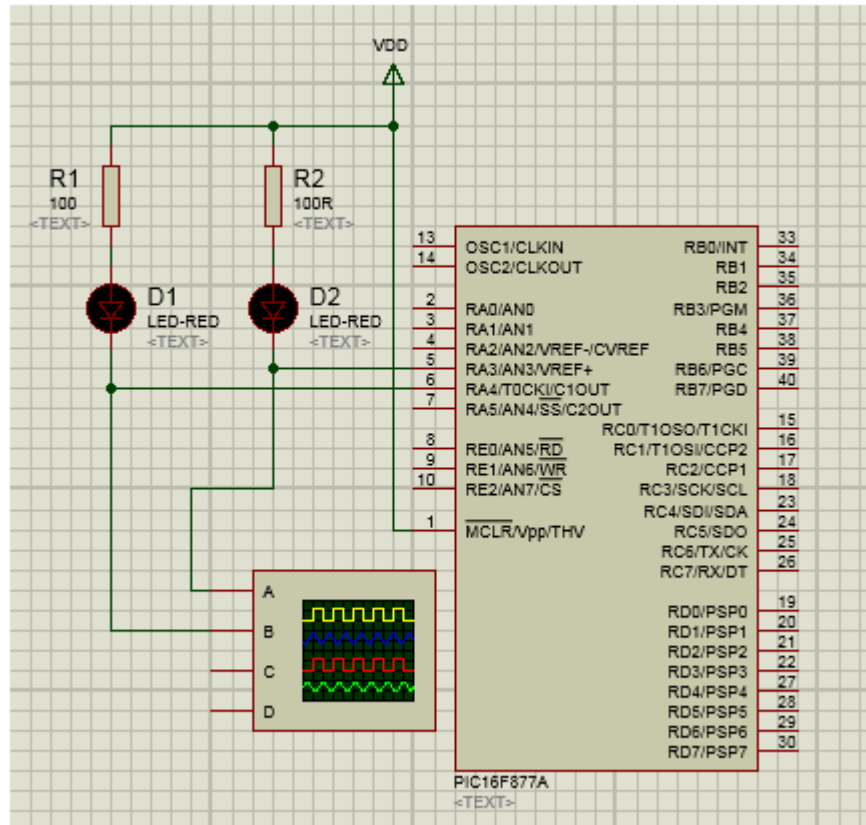
- 8.- LAS INTERRUPCIONES SON DEL TIPO:.....(**C**)
A) CONMUTATIVO
B) CONTINUÚO
C) ALEATORIO
D) SINCRONO
- 9.- PARA PRESERVAR EL ESTADO DE LA CPU SE UTILIZA UNA ESTRUCTURA DE SOFTWARE LLAMADA :.(**D**)
A) COLAS.
B) LISTAS.
C) PIPES.
D) PILA.
- 10.- EL BIT GIE DEL REGISTRO INTCON SIRVE PARA :.....(**B**)
A) DESHABILITAR TODAS LAS INTERRUCIONES.
B) HABILITAR LAS INTERRUPCIONES.
C) PROBAR EL DESBORDAMIENTO DEL TIMER 0.
D) PROBAR EL ESTADO DEL TIMER 0.

Valor por reactivo: **0.1 de punto.**
Valor de la sección **PARTE A** : **1 punto.**

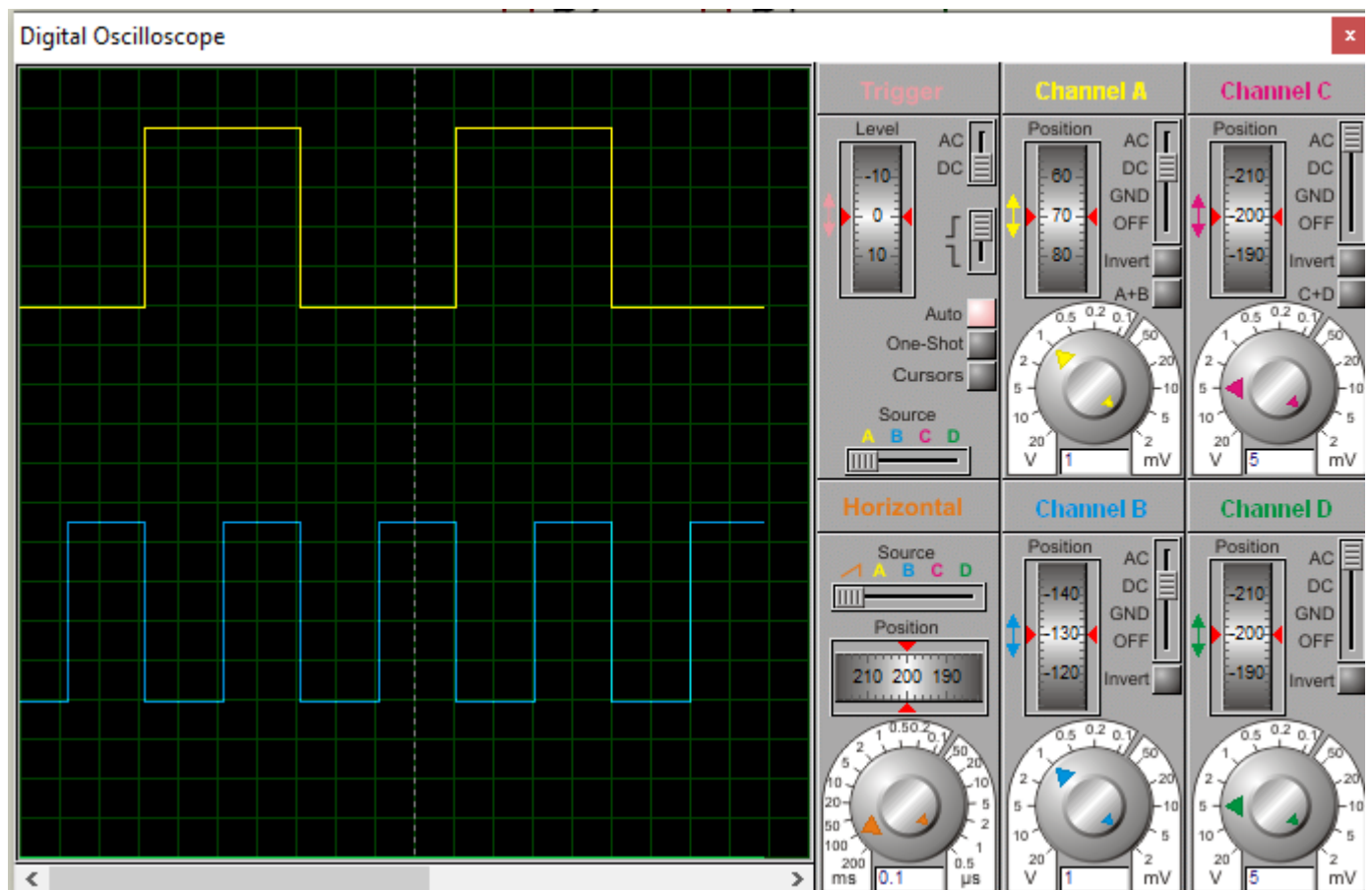
II.- Sección Segunda **PARTE B.**

INSTRUCCIONES
CONTESTE UNICAMENTE LO QUE SE PIDE.

PROBLEMA: DESARROLLE UN SISTEMA DIGITAL QUE CUENTE CON LAS SIGUIENTES CARACTERISTICAS.

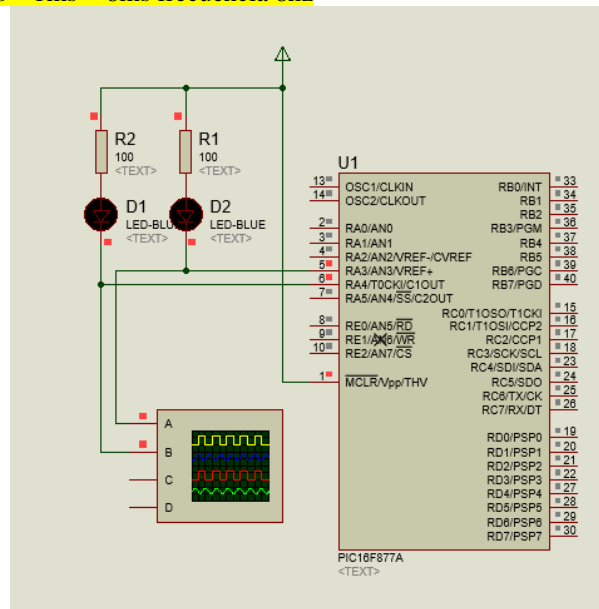


- A) EL SISTEMA CUENTA CON DOS LEDS ¿DIBUJE LOS DIAGRAMAS DE TIEMPO PARA HACER OSCILAR EL LED “D1” A 4 Hz Y EL LED “D2” A 8 Hz.?



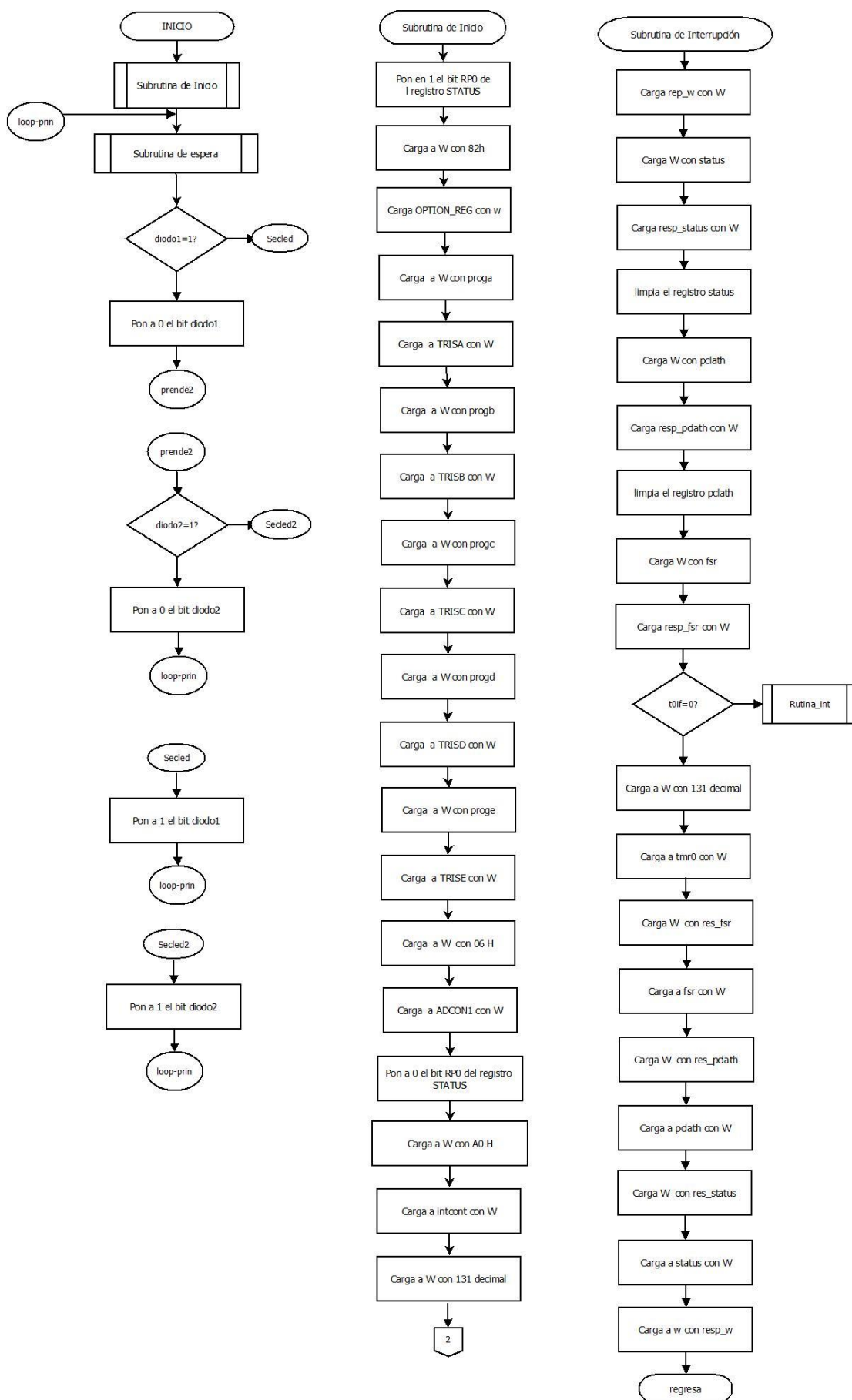
Periodo del diodo1= 4 cuadros * 1ms = 4ms frecuencia 4hz

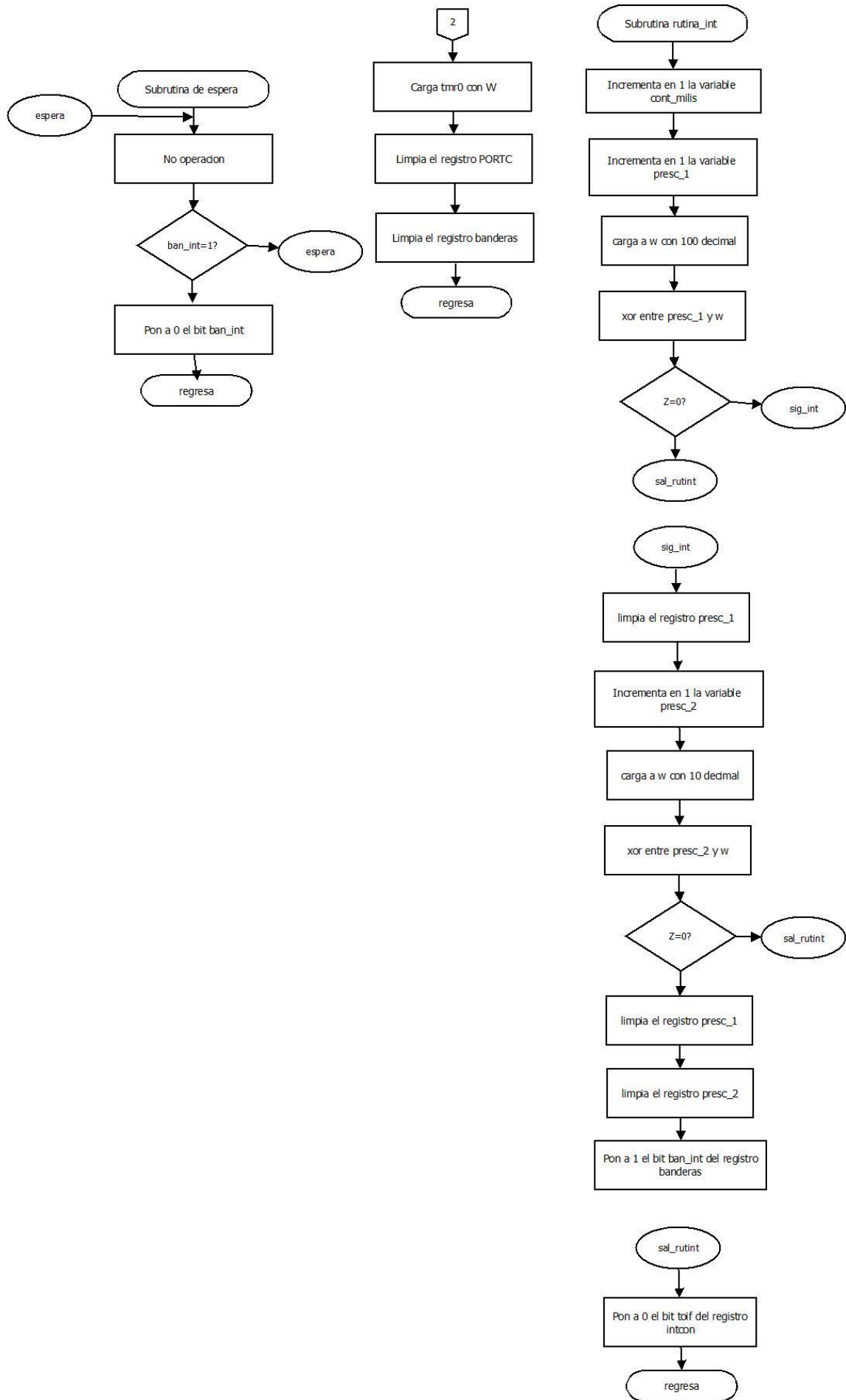
Periodo del diodo2= 8 cuadros * 1ms = 8ms frecuencia 8hz



NOTA: OBTENGA LOS PERIODOS DE LAS SEÑALES POR LAS FRECUENCIAS DADAS.

B) DIBUJE EL DIAGRAMA DE FLUJO PARA HACER OSCILAR A LOS LEDS A LAS FRECUENCIAS PEDIDAS EN EL INCISO ANTERIOR UTILIZANDO INTERRUPCIONES GENERADAS POR EL TIMER 0.





C) ESCRIBA EL PROGRAMA EN LENGUAJE ENSAMBLADOR EN MPLAB DEL INCISO ANTERIOR Y AGREGUELO AL EXAMEN.

```
;INSTITUTO POLITECNICO NACIONAL
;CECYT 9 JUAN DE DIOS BATIZ
```

```
;
;EXAMEN 1.
```

```
;
;GRUPO:6IM2
```

```
;
;INTEGRANTE
;Morales Martínez José Antonio
```

```
;
;El programa controlara 2 leds uno a 4hz y otro a 8hz
```

```
;
;-----
List    p=16f877A;
#include "c:\Program files (x86)\Microchip\Mpasm Suite\p16f877a.inc";
```

```
__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_OFF & _XT_OSC & _WRT_OFF & _LVP_OFF
& _CPD_OFF;
```

```
;-----
;
```

```
; Fosc = 4 MHz.
; Ciclo de trabajo del PIC = (1/fosc)*4 = 1 µs.
; T int =(256-tmr0)*(P)*((1/4000000)*4) = 1 ms.  // Tiempo de interrupción.
; tmr0=131, P=8.
; frec int = 1/ t int = 1 KHz.
```

```
;-----
;
```

```
;Def. de variables del programa en RAM.
```

```
res_w          equ          0x29;  //Dirección de la memoria RAM para el registro de respaldo de "w"
res_status     equ          0x30;  //Dirección de la memoria RAM para el registro de respaldo
de "status"
res_pclath     equ          0x31;  //Dirección de la memoria RAM para el registro de respaldo
de "pclath"
res_fsr        equ          0x32;  //Dirección de la memoria RAM para el registro de respaldo
de "fsr"
presc_1        equ          0x33;  //Dirección de la memoria RAM para el registro del
prescalador 1
presc_2        equ          0x34;  //Dirección de la memoria RAM para el registro del
prescalador 2
banderas       equ          0x35;  //Dirección de la memoria RAM para el registro "banderas"

cont_milis     equ          0x36;  //Dirección de la memoria RAM para el registro contador de
milisegundos
osc_diodo1     equ          0x37;
```

```
;-----
```

```
; Def. de constantes a utilizar.
; Cod. de caracteres alfanuméricos en 7 segmentos.
```

```
Car_A          equ    b'01110111';
Car_b          equ    0xc7;
Car_0          equ    0x3f;
Car_1          equ    0x06;
```

```
; banderas del registro banderas.
```

```

ban_int      equ    .0;
sin_bd1      equ    .1;
sin_bd2      equ    .2;
sin_bd3      equ    .3;
sin_bd4      equ    .4;
sin_bd5      equ    .5;
sin_bd6      equ    .6;
sin_bd7      equ    .7;
;-----
;
;Asignacion de los bits de los puertos de I/O.
;Puerto A.
Sin_UsoRA0    equ    .0; // Señal de control de Comando o dato en la LCD
Sin_UsoRA1    equ    .1; // Señal de ingreso de información a la LCD
Sin_UsoRA2    equ    .2; // Sin Uso RA2.
Diodo2        equ    .3; // Sin Uso RA3.
Diodo1        equ    .4; // Sin Uso RA4.
Sin_UsoRA5    equ    .5; // Sin Uso RA5.

proga         equ    b'100111'; // Programacion Inicial del Puerto A.

;Puerto B.
Sin_UsoRB0    equ    .0; // Sin Uso RB0.
Sin_UsoRB1    equ    .1; // Sin Uso RB1.
Sin_UsoRB2    equ    .2; // Sin Uso RB2.
Sin_UsoRB3    equ    .3; // Sin Uso RB3.
Sin_UsoRB4    equ    .4; // Sin Uso RB4.
Sin_UsoRB5    equ    .5; // Sin Uso RB5.
Sin_UsoRB6    equ    .6; // Sin Uso RB6.
Sin_UsoRB7    equ    .7; // Sin Uso RB7.

progb         equ    b'11111111'; // Programacion Inicial del Puerto B.

;Puerto C.
D0_LCD        equ    .0; // Sin Uso RC0.
D1_LCD        equ    .1; // Sin Uso RC1.
D2_LCD        equ    .2; // Sin Uso RC2.
D3_LCD        equ    .3; // Sin Uso RC3.
D4_LCD        equ    .4; // Sin Uso RC4.
D5_LCD        equ    .5; // Sin Uso RC5.
D6_LCD        equ    .6; // Sin Uso RC6.
D7_LCD        equ    .7; // Sin Uso RC7.

progC         equ    b'11111111'; // Programacion Inicial del Puerto C como Entrada.

;Puerto D.
Sin_UsoRD0    equ    .0; // Sin Uso RD0.
Sin_UsoRD1    equ    .1; // Sin Uso RD1.
Sin_UsoRD2    equ    .2; // Sin Uso RD2.
Sin_UsoRD3    equ    .3; // Sin Uso RD3.
Sin_UsoRD4    equ    .4; // Sin Uso RD4.
Sin_UsoRD5    equ    .5; // Sin Uso RD5.
Sin_UsoRD6    equ    .6; // Sin Uso RD6.
Sin_UsoRD7    equ    .7; // Sin Uso RD7.

progd         equ    b'11111111'; // Programacion Inicial del Puerto D como entradas.

;Puerto E.
Sin_UsoRE0    equ    .0; // Sin Uso RE0.

```



```

Sin_UsoRE1      equ          .1; // Sin Uso RE1.
Sin_UsoRE2      equ          .2; // Sin Uso RE2.

proge           equ          b'111'; // Programacion inicial del Puerto E.
;-----
;=====
;==Vector Reset==
;=====
org 0x0000; // dirección de inicio de la memoria donde
el IDE comenzara a ensamblar
vec_reset       clrf PCLATH; // Limpia el registro PCLATH
                goto prog_prin; // ve para la etiqueta prog_ini
;-----
;=====
;== Subrutina de Interrupciones ==
;=====
org 0004h;
vec_int         movwf res_w; //Respaldar el estado del registro w
                movf status,w; //Mover el contenido del registro status a el registro de trabajo
                movwf res_status; //Respaldar las banderas de la alu
                clrf status; //Limpia el registro STATUS
                movf pclath,w; //Mover el contenido del registro res_pclath a el registro de trabajo
                movwf res_pclath; //Respaldar el estado del registro pclath
                clrf pclath; //Limpia el registro pclath
                movf fsr,w; //Mover el contenido del registro res_fsr a el registro de trabajo
                movwf res_fsr; //Respaldar el estado del registro fsr

                btfsc intcon,t0if; //Si el bit t0if del registro intcon es igual a 0 salta
                call rutina_int; //LLamada a la subrutina de interrupciones

sal_int         movlw .62; //Mover la constante 131 al registro de trabajo
                movwf tmr0; //Mover el contenido del registro de trabajo al registro tmr0
                movf res_fsr,w; //Mover el contenido del respaldo res_fsr a el registro de trabajo
                movwf fsr; //Mover el contenido del registro de trabajo al registro far
                movf res_pclath,w; //Mover el contenido del respaldo res_pclath a el registro de trabajo
                movwf pclath; //Mover el contenido del registro de trabajo al registro pclath
                movf res_status,w; //Mover el contenido del respaldo res_status a el registro de trabajo
                movwf status; //Mover el contenido del registro de trabajo al registro status
                movf res_w,w; //Mover el contenido del respaldo res_w a el registro de trabajo

                retfie; //Regresar al programa principal
;-----
;=====
;== Subrutina de Interrupciones ==
;=====
rutina_int      incf cont_milis,f; //Incrementa la variable cont milis y guarda en el mismo registro
                incf presc_1,f; //Incrementa la variable presc 1 y guarda en el mismo registro

                movlw .125; //Mover la constante 100 al registro de trabajo
                xorwf presc_1,w; //XOR entre registro presc 1 y el registro de trabajo
                btfsc status,z; //Si el bit z del registro status es igual a 0 salta
                goto sal_rutint; //Ve para la etiqueta sal_rutint

                bsf banderas,ban_int; //Pon a 1 el bit ban int del registro banderas

sal_rutint      bcf intcon,t0if; //Pon a 0 el bit ban t0if del registro intcon
                return; //Regresar al programa principal

```

```

;-----

;=====
;==Subrutina de inicio==
;=====

prog_ini      bsf STATUS,RP0;          //colocate en el bco. 1 de ram
               movlw 0x82;              // Mueve la constante 0X81 al registro w
               movwf OPTION_REG ^0x80; // Configura el preescalador y descactiva los pull-up
               movlw prog_a;            // Mueve el contenido de w a el registro prog_a
               movwf TRISA ^0x80;       // Mueve la constante 0X80 al registro TRISA
               movlw prog_b;            // Mueve el contenido de w a el registro prog_b
               movwf TRISB ^0x80;       // Mueve la constante 0X80 al registro TRISB
               movlw prog_c;            // Mueve el contenido de w a el registro prog_c
               movwf TRISC ^0x80;       // Mueve la constante 0X80 al registro TRISC
               movlw prog_d;            // Mueve el contenido de w a el registro prog_d
               movwf TRISD ^0x80;       // Mueve la constante 0X80 al registro TRISD
               movlw prog_e;            // Mueve el contenido de w a el registro prog_e
               movwf TRISE ^0x80;       // Mueve la constante 0X80 al registro TRISE
               movlw 0x06;              // Mueve la constante 0X06 al registro w
               movwf ADCON1 ^0x80;      // Mueve la constante 0X80 al registro ADCON1
               bcf STATUS,RP0;          //regresa al bco. 0 de ram

               movlw 0xa0;

               movwf intcon;

               movlw .131;
               movwf tmr0;

               clrf banderas;

               clrf porta;

               return;

;-----

;=====
;== Programa principal ==
;=====

prog_prin      call prog_ini;
Loop_prin      call esp_int;

               btfss porta,diodo2;
               goto sec_led;
               bcf porta,diodo2; Prende el led.
                                   goto prende2;

               goto Loop_prin;
sec_led        bsf porta,diodo2; Apaga el led.
               goto Loop_prin;

prende2        btfss porta,diodo1;
               goto sec_led2;
               bcf porta,diodo1; Prende el led.
               goto Loop_prin;
sec_led2       bsf porta,diodo1; Apaga el led.
               goto Loop_prin;

;-----

```

```

;=====
;== Subrutina de espera de int. de 125 ms ==
;=====
esp_int      nop;
             btfss banderas,ban_int;
             goto esp_int;
             bcf banderas,ban_int;

             return;
;-----
end

```

Valor del inciso A y B: **1 punto.**
 Valor del inciso C: **2 puntos.**

Valor de la sección **PARTE B: 4 puntos.**

VALOR TOTAL DEL EXAMEN: 5 PUNTOS.