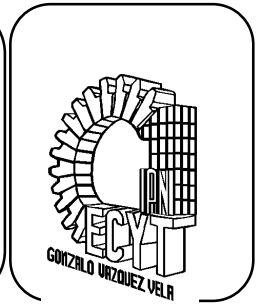


INSTITUTO POLITÉCNICO NACIONAL
Centro de Estudios Científicos y Tecnológicos N° 1
“Gonzalo Vázquez Vela”
Academia de Sistemas Digitales
Prácticas de Micro Electrónica Programable



NOMBRE DEL ALUMNO: _____
Apellido Paterno

_____ Apellido Paterno Nombre

N° DE BOLETA: _____ GRUPO: _____

ASIGNATURA: **Micro Electrónica Programable**

HOJA	DE	FECHA			EVALUACION
		DIA	MES	AÑO	

PROFESOR: _____

Práctica 4

Multiplexado de datos

Competencias de La Unidad:

- Emplea el microcontrolador en la comunicación de datos y control de periféricos de forma multiplexada.

Resultado de Aprendizaje Propuesto (RAP):

- Usa diferentes elementos periféricos utilizando los puertos del microcontrolador de forma multiplexada.
- Utiliza los puertos del microcontrolador de forma bidireccional en la solución de un problema.

Objetivos de la Práctica:

1. Realizar la simulación de un programa para comprobar su funcionamiento utilizando herramientas computacionales
2. Implementar técnicas de multiplexación de datos con el fin de tener un manejo de información más optima.
3. Implementar un programa en un circuito basado en microcontrolador, para comprobar su funcionamiento

Equipo Necesario	Material Necesario
<p>Computadora (con el Software MPLAB IDE, IC-PROG o similar, compilador C, Simulador de circuitos electrónicos "Proteus")</p> <p>Programador tipo JDM o similar.</p>	<p>Instrucciones del PIC 16F887 u otro de gama media o alta</p> <p>Microcontrolador PIC16F887 u otro de gama media o alta</p> <p>Capacitores</p> <p>Display de cátodo común</p> <p>Resistencias</p> <p>pushbutton</p> <p>Cristal de cuarzo de 4MZ</p> <p>Transistores BC548</p> <p>(Para los valores de estos elementos ver figura 4.1)</p>

Introducción Teórica

Multiplexado de dato

Multiplexar en computación tiene el significado compartir un bus de datos entre varios dispositivos.

En el caso de microprocesadores y microcontroladores esta idea es utilizada para visualizar información en varios display de 7 segmentos, en los cuales el bus de datos está conectado a todos ellos y a partir de ir habilitando cada display por un breve tiempo y pasar al siguiente dígito en un tiempo que no sea perceptible para el ojo humano se logra una visualización de información, esta misma idea es utilizada en las matrices de led.

ACTIVIDADES TEÓRICAS PREVIAS

Investigar los siguientes conceptos:

- Distintas formas para efectuar Multiplexado en display's de 7 segmentos.
- Menciona las características de una matriz de led's

ACTIVIDADES PREVIAS

- **Crear un proyecto de nombre pra4 en la carpeta c:\MEPIC\ practica4 en MPLAB o PIC C Compiler. Los programas de cada ejercicio deben ser guardados con el nombre practica4X.c con X= 1, 2, 3...,A.**
- **En el caso de utilizar MPLAB, realizar los siguientes pasos:**
 - Utilizar Project wizard y seleccionar el compilador de c**
 - Agregar al proyecto los archivos adecuados con extensión c y h.**

- c. **Habilitar Simulador MPLAB SIM y modificar la frecuencia del simulador a 4 Mhz.**
- d. **Utilizaremos la herramienta de stopwatch, para obtener la elija Debugger >> Stopwatch.**
- e. **Obtener la herramienta de watch, de la siguiente manera View>> watch.**
- f. **Y seleccione los registros PORTA, PORTB, PORTC, PORTD, PORTE, TRISA, TRISB, TRISC, TRISD, TRISE y W**

- **Si usa PIC C compiler crear el proyecto únicamente e incluir un archivo nuevo.**

ACTIVIDADES PRÁCTICAS

Parte 1

A. Armar el siguiente circuito

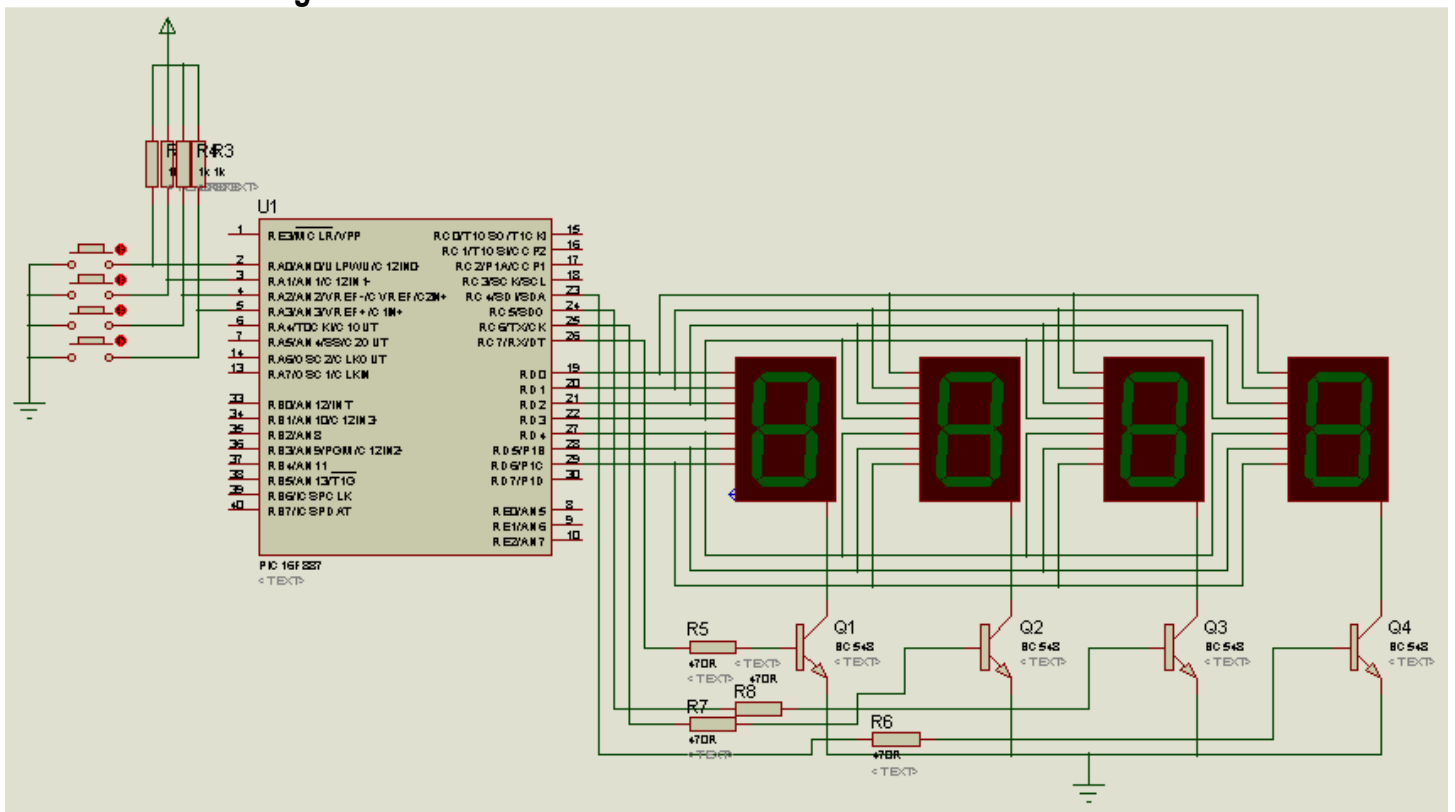


Figura 4.1

Nota: La terminal 12 o 31 del PIC16F887 se conectan a tierra.

Realizar el siguiente programa y observar su funcionamiento utilizando la opción watch y stopwatch del simulador MPLAB (en caso de usarlo) y posteriormente grabarlo en el circuito de la figura 4.1, así como simularlo en PROTEUS

- **Nota: A grabarlo deshabilitar WDT y LVP y habilitar PWRTE y BODEN en la palabra de configuración, además recuerde seleccionar el tipo de oscilador a XT**

Ejemplo 1

```
#include <16f887.h>
#use delay(clock=4000000)
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use standard_io(d)
#use standard_io(c)
void main()
{
while(true)
{
output_c(0x10);
output_d(0x7D);
delay_ms(1000);
output_c(0x20);
output_d(0x7D);
delay_ms(1000);
output_c(0x40);
output_d(0x7D);
delay_ms(1000);
output_c(0x80);
output_d(0x7D);
delay_ms(1000);
}
}
```

Ejemplo 2

```
#include <16f887.h>
#use delay(clock=4000000)
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use standard_io(d)
#use standard_io(c)
void main()
{
while(true)
{
output_c(0x10);
output_d(0x40);
delay_ms(50);
output_c(0x20);
output_d(0x54);
delay_ms(50);
output_c(0x40);
output_d(0x73);
delay_ms(50);
output_c(0x80);
output_d(0x06);
delay_ms(50);
}
}
```

Ejemplo 3

```
#include <16f887.h>
#use delay(clock=4000000)
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use standard_io(d)
#use standard_io(c)
int j;
void main()
{
while(true)
{
for(j=0;j<=20;j++)
{
output_c(0x10);
output_d(0x77);
delay_ms(50);
output_c(0x20);
output_d(0x78);
delay_ms(50);
output_c(0x40);
output_d(0x39);
delay_ms(50);
output_c(0x80);
output_d(0x3F);
delay_ms(50);
}
for(j=0;j<=20;j++)
{
output_c(0x10);
output_d(0x00);
delay_ms(50);
output_c(0x20);
output_d(0x77);
delay_ms(50);
output_c(0x40);
output_d(0x78);
delay_ms(50);
output_c(0x80);
output_d(0x39);
delay_ms(50);
}
for(j=0;j<=20;j++)
{
output_c(0x10);
output_d(0x3f);
delay_ms(50);
output_c(0x20);
output_d(0x00);
delay_ms(50);
output_c(0x40);
}
```

```

output_d(0x77);
delay_ms(50);
output_c(0x80);
output_d(0x78);
delay_ms(50);
}
for(j=0;j<=20;j++)
{
output_c(0x10);
output_d(0x39);
delay_ms(50);
output_c(0x20);
output_d(0x3f);
delay_ms(50);
output_c(0x40);
output_d(0x00);
delay_ms(50);
output_c(0x80);
output_d(0x77);
delay_ms(50);
}
for(j=0;j<=20;j++)
{
output_c(0x10);
output_d(0x78);
delay_ms(50);
output_c(0x20);
output_d(0x39);
delay_ms(50);
output_c(0x40);
output_d(0x3f);
delay_ms(50);
output_c(0x80);
output_d(0x00);
delay_ms(50);
}
}
}
}

```

Ejemplo 4

Los siguientes programas permiten manejar matriz de led's conectada como se muestra en la figura 4.2. Armar en proteus y verificarlo.

a)Ejemplo 4.1

```

#include <16f887.h>
#use delay(clock=4000000)
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use standard_io(d)
#use standard_io(c)

```

```

byte CONST unidad[7] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40};
int i,j;
void main()
{
i=0;
j=0;
while(true)
{
for(i=0;i<=5;i++)
{
output_d(0x80);
output_c(unidad[i]);
delay_ms(1000);
output_c(0x00);
delay_ms(100);
}
}
}

```

b) Ejemplo 4.2

```

#include <16f887.h>
#define delay(clock=4000000)
#define fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#include <standard_io(d)>
#include <standard_io(c)>
void main()
{
while(true)
{
output_c(0x01);
output_d(0x80);
delay_ms(50);
output_c(0x02);
output_d(0xFE);
delay_ms(50);
output_c(0x04);
output_d(0xFE);
delay_ms(50);
output_c(0x08);
output_d(0xFE);
delay_ms(50);
output_c(0x10);
output_d(0x80);
delay_ms(50);
output_c(0x00);
output_d(0xFF);
delay_ms(100);
}
}

```

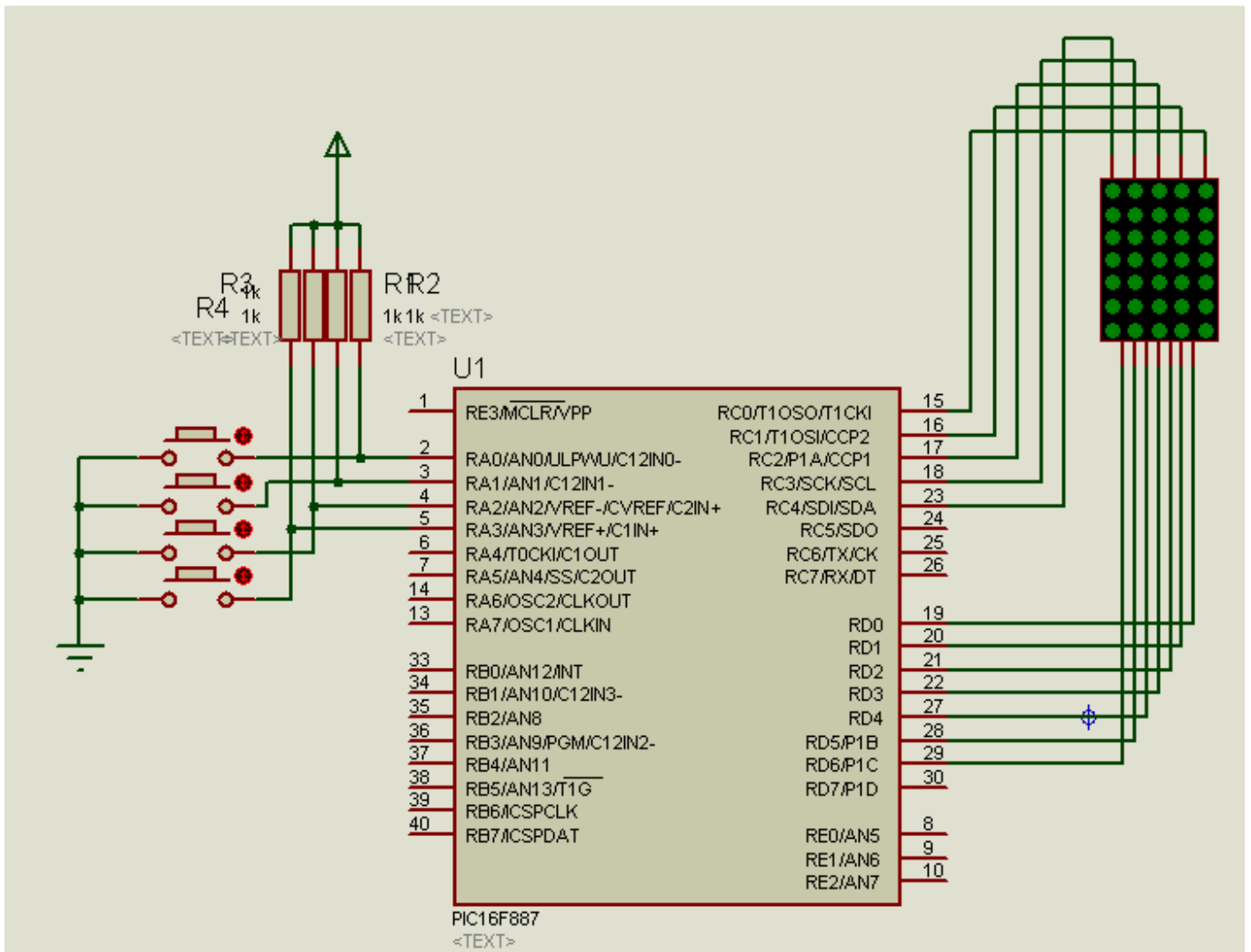


Figura 4.2

Nota: tome en cuenta que para poder implementar debe realizar las conexiones del oscilador y el reset, así como la alimentación correspondiente

Parte 2 Ejercicios

- Realizar un programa que recorra la letra B de izquierda a derecha y viceversa.
- Realizar un programa que recorra el número 1 de izquierda a derecha y viceversa en todos los display's y que efectúe el mismo procedimiento hasta el 9. (Sugerencia utilice arreglos y ciclos for).
- Proponga un programa utilizando dos dígitos para realizar un contador de 0 a99 de manera automática. (sugerencia utilizar a arreglos)

Para los siguientes ejercicios modifique el circuito de la figura 4.1 como considere pertinente.

D. Modifique el programa del inciso anterior que se incremente en uno al oprimir el botón RA0.

E. Realice un programa que efectúe las siguientes acciones

- **Al activar el botón de RA0 se visualicen en el display's "GVV1"**
- **Al activar el botón de RA1 se visualicen en el display's "247A"**
- **Al activar el botón de RA2 se visualicen en el display's "6IVX" o de acuerdo su grupo**
- **Al activar el botón de RA3 se visualicen en el display's "SD-1"**

Cada letra o símbolo debe visualizarse en un display.

F. Utilizando el circuito de la figura 4.2 implemente un programa que realice las siguientes acciones (Utilice sentencia switch, para la lectura de datos)

- **Al activar el botón de RA0 se visualicen en la matriz "CECyT1"**
- **Al activar el botón de RA1 se visualicen en la matriz "IPN"**
- **Al activar el botón de RA2 se visualicen en la matriz "SDTV"**

Cada letra debe visualizarse el tiempo adecuado para observarse claramente.

(Únicamente simularlo en proteus, la implementación física es opcional)

1. Conclusiones

A. Realizar conclusiones de manera individual.

2. Cuestionario

- ¿Qué significa multiplexar datos?**
- Explique las formas de visualizar datos en display utilizando técnicas de multiplexado**
- En el ejemplo 3, ¿Cuál es la función de las instrucciones for?.**
- ¿Cuál es la ventaja de utilizar arreglos en la visualización de datos?**
- ¿Menciona como simplificarías el número de líneas de datos, tanto en el caso de los display, como en el de la matriz de led's ?**

Comentarios Finales

- **El alumno entrega un reporte de la práctica, como el profesor lo indique.**
- **El reporte debe contener el diagrama de flujo o algoritmo (Seudo código) de cada uno de los programas.**
- **Además, en el reporte deben anexarse las conclusiones y cuestionario contestado.**