INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Ingeniería Mecánica y Eléctrica

"Unidad Culhuacan"

Academia de Comunicaciones y Electrónica

Prácticas de Teoría de Control Digital

ASIGNATURA: ***Teoría de Control Digital***

GRUPO	FECHA			FIRMA
	DIA	MES	AÑO	

Práctica 8
Comunicación Serie (USART)

Objetivos de la Práctica:

1. Realizar la simulación de un programa para comprobar la comunicación de tipo serie asíncrona, utilizando el módulo EUSART/SCI

2. Implementar programas en un circuito basado en microcontrolador, para comprobar su funcionamiento.

Introducción Teórica

Los microcontroladores PIC cuentan con módulo de comunicación serie denominado; módulo **Transmisor/Receptor Universal Síncrono/Asíncrono mejorado** (Enhanced Universal Synchronous Asynchronous Receiver Transmitter - EUSART) o también llamado Interfaz de comunicación serie (Serial Communications Interface - SCI), el cual tiene la característica de contar con todos los generadores de señales de reloj, registros de desplazamiento y búfers de datos para poder llevar acabo la transmisión de datos serie de entrada/salida, independientemente de la ejecución de programa del dispositivo. Como indica su nombre, además de utilizar el reloj para la sincronización, este módulo puede realizar una conexión asíncrona.

El módulo EUSART realiza la transmisión y recepción de datos de manera serie, transfiriendo tramas de 8 o 9 bits permitiendo la transmisión y detección de errores, además puede generar interrupciones al ocurrir una recepción de datos o cuando la transmisión ha sido concluida. Las características más destacadas del módulo son:

- Modo dúplex (Full Duplex) asíncrono.
- Semi-duplex (Half-Duplex) síncrono.
- Longitud del carácter programable a 8 o 9 bits.
- Detección de error en transmisión.

En este caso únicamente nos enfocaremos al modo asíncrono que emplea reloj tanto en el emisor como en el receptor, cuya frecuencia debe ser la misma, la cual se acuerda antes de la transmisión a partir de la configuración de la velocidad. Por otra parte, cada trama de información cuenta con un tamaño fijo y tiene un bit de arranque y un bit de parada que permite realizar la sincronización en la transmision
Adicionalmente este módulo permite la detección automática de baudios, entre otras características.

Las instrucciones o directivas que utiliza el compilador C para poder usar el módulo EUSART son:

#use RS232(opciones) : Esta directiva permite configurar algunos parámetros del módulo: como la velocidad de trasmisión, pines, entre otros.
Un ejemplo común del uso de esta directiva es:

#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8, parity=N)
En este caso los parámetros de las directivas significan: Una velocidad de 9600 baudios, el pin C6 es de transmisión, el pin C7 de recepción y sin paridad, para ver otras opciones ver el manual de referencia de CCS.

Las instrucciones para transmitir datos son:

- Envía un carácter de 8 bits
  putc(dato)
  putchar(dato)
- Envía una cadena de caracteres
  puts (cadena)
  printf(fname, cadena de a enviar, valor)

fname: Son las funciones utilizadas para escribir la cadena, se usa la función putc() por defecto.
valor: Son valores que se incluyen en la cadena , indicado con el formato %nt.

ACTIVIDADES TEÓRICAS PREVIAS

Investigar los siguientes conceptos:

- Menciona las características de protocolo RS232
- ¿Qué ventaja tiene usar comunicación serial?
- Menciona los registros que se afectan al realizar una comunicación serie.
- De cuantos bits está conformada la trama de transmisión de la comunicación serial.

ACTIVIDADES PRÁCTICAS

Parte 1

A. Armar el siguiente circuito,

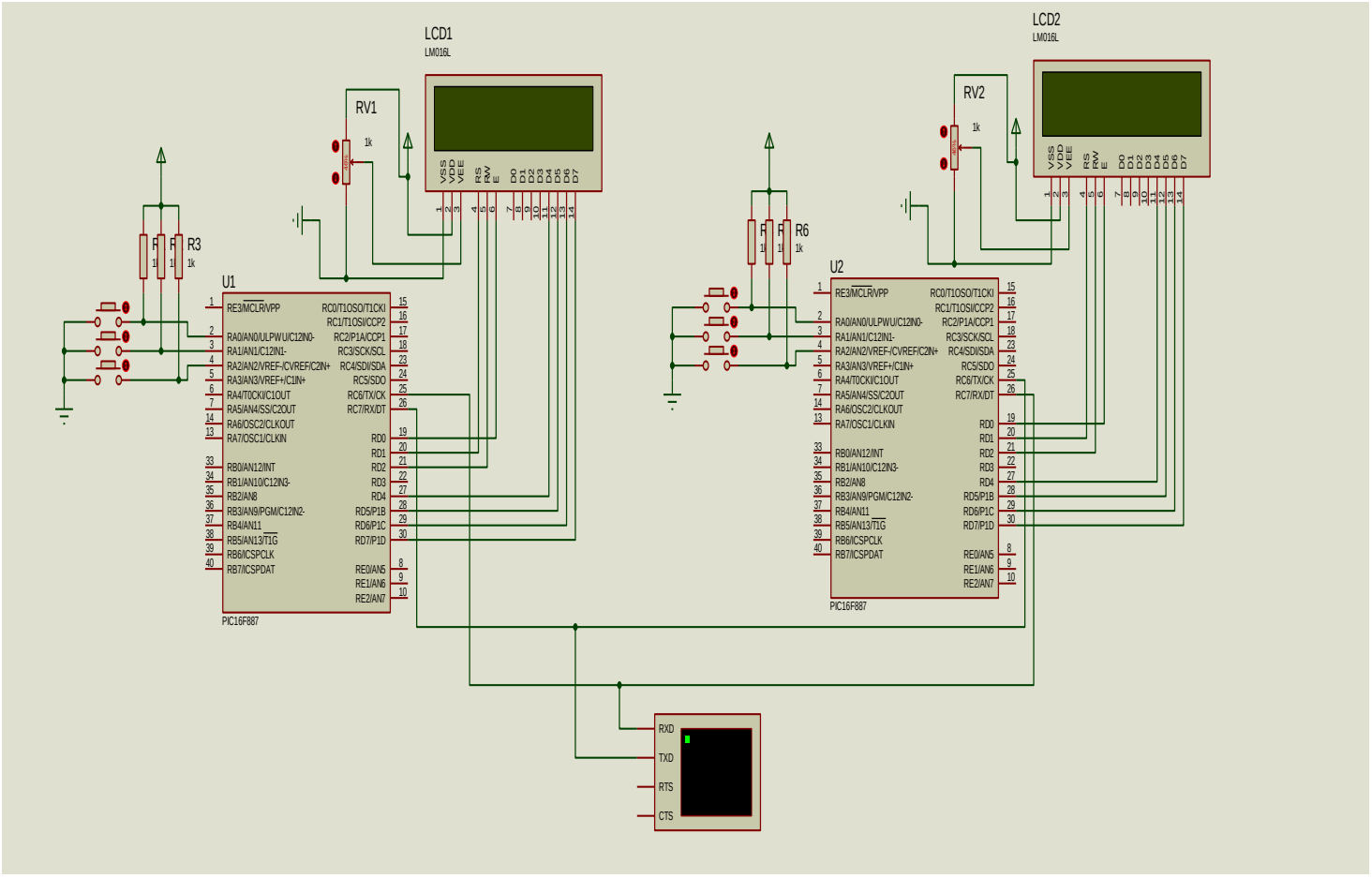

Figura 8.1

Ejemplo 1

Realizar el siguiente programa y observar su funcionamiento en el circuito de la figura 8.1, así como simularlo en PROTEUS, el cual muestra la comunicación entre dos microcontroladores mediante el módulo USART y se visualiza el valor en el LCD.

Código para el microcontrolador que transmite (PIC de la izquierda figura 8.1)

```
#include <16F887.h>
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use delay(xtal=20,000,000)
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7,bits=8,parity=N)
#include <LCD.C>

void main()
{
 int16 valor;

   lcd_init();

while(true){
   for (valor=0;valor<=255;valor++)
      {
    PUTC(valor);
     printf(lcd_putc,"\fenviando=%lD",valor);
     delay_ms(100);
       }
            }
}
```

Código para el microcontrolador que transmite (PIC de la derecha figura 8.1)

```
#include <16F887.h>
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use delay(xtal=20,000,000)
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8,parity=N)
#include <LCD.C>
#BYTE TRISA=0X85
#BYTE PORTA=0X05

  int16 valor;
#int_RDA
void RDA_isr(void)
{
 valor=GETC();
}

void main() {

  bit_clear(TRISA,0);
  lcd_init();
   enable_interrupts(INT_RDA);
   enable_interrupts(GLOBAL);

for (;;) {
       lcd_gotoxy(1,1);

       printf(lcd_putc,"recibiendo=%3ld",valor);
       output_b(valor);

   }
}
```

Ejemplo 2

Realiza la comunicación entre pic y un puerto serie de PC, para observar los datos que se envían se utiliza el elemento COMPIM de proteus, armar el circuito de la figura 8.2 y simularlo en PROTEUS

```
#include <16F887.h>
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#device adc=10
#use delay(xtal=20,000,000)
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8, parity=N)
#include <LCD.C>
void main() {
   int16 q;
   float p;
   setup_adc_ports(sAN0);
   setup_adc(ADC_CLOCK_INTERNAL);
   lcd_init();
   for (;;) {
      set_adc_channel(0);
      delay_us(10);
      q = read_adc();
      p = 5.0 * q / 1024.0;
      printf(lcd_putc, "\fADC = %4ld", q);
      printf(lcd_putc, "\nVoltage = %01.2fV", p);
      printf("ADC = %4ld ", q);
      printf("Voltage = %01.2fV\r", p);  // El \r permite cambiar de línea.
      delay_ms(100);
      }
}
```

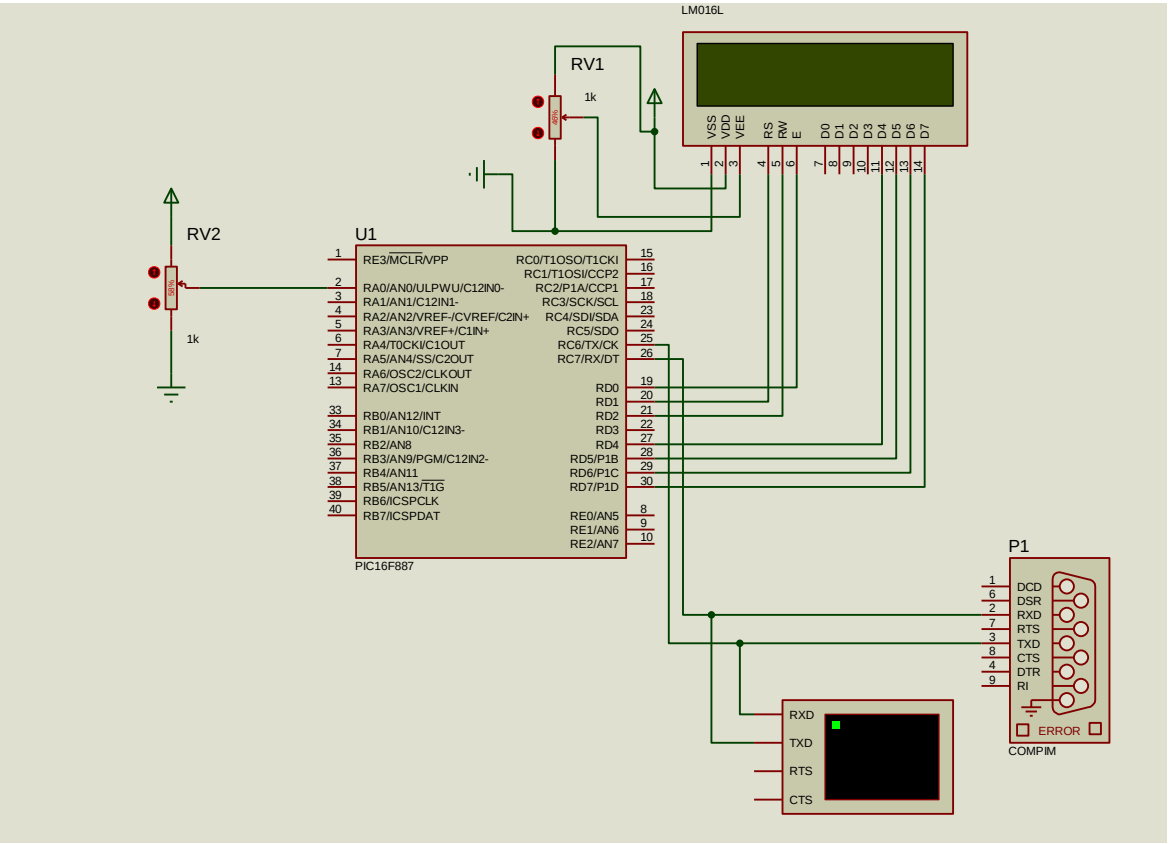

Figura 8.2

1. Conclusiones

A. Realizar conclusiones de manera individual.

Comentarios Finales

- **El alumno entrega un reporte de la práctica, como el profesor lo indique.**

				ASISTENCIA	OBSERVACIONES
INTEGRANTE 1:	Apellido Paterno	Apellido Materno	Nombre (S)		
INTEGRANTE 2:	Apellido Paterno	Apellido Materno	Nombre (S)		
INTEGRANTE 3:	Apellido Paterno	Apellido Materno	Nombre (S)		
INTEGRANTE 4:	Apellido Paterno	Apellido Materno	Nombre (S)		