

# Microcontroladores

## Semana 13

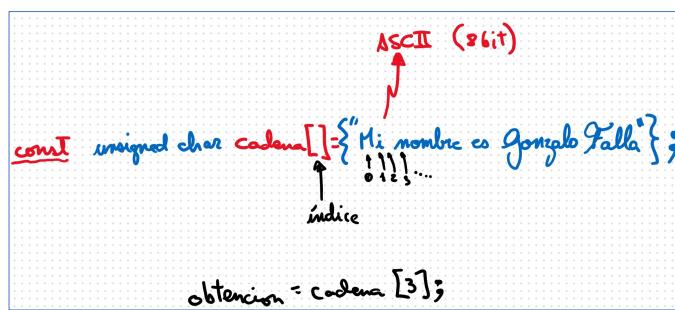
Sesión Teoría / Laboratorio

Profesor Kalun José Lau Gan

1

### Preguntas previas

- ¿Profesor veremos comunicación inalámbrica?
  - Puedo dar información y referencias acerca de ello ya que es un tema mucho mas extenso que lo verán en el curso de sistemas embebidos.
- ¿Hay alguna función en el xc8 q permita convertir un int en un string que podamos iterar y recorrer?
  - No es necesario ya que el compilador se encarga de las conversiones de tipo de variable pero hay que tener en cuenta que si hacer una igualdad dentro un tipo grande hacia un tipo chico se considerará los bits menos significativos.



2

## Preguntas previas:

- ¿Hay regulación en el Perú para el uso de vehículos aéreos no tripulados (drones)?
  - Si: <https://www.gob.pe/institucion/mtc/noticias/50511-mtc-conoce-los-requisitos-para-operar-un-drone>
- Respecto a las clases presenciales sabe si serán el próximo año?
  - Eso dependerá del levante del estado de emergencia, de que se tenga un alto porcentaje de ciudadanos vacunados y de la decisión que se tome a nivel de universidad.
- Veremos puerto serie en modo asíncrono?
  - En esta sesión se atenderá dicho tema
- ¿Cómo se inicializa un proceso de lectura para el PCF8574?
  - Cond(START), Slaveaddr+read, ACK, Lectura de dato 8bit, noACK, Cond(STOP)
- Encuestas finales

3

## Preguntas Previas

- ¿Nuestro informe del proyecto final tiene que estar en formato IEEE?
  - Formato a simple columna con todos los puntos mencionados en la guía de TF
- Si encuentro en internet alguna librería que me puede resultar útil en mi TF la puedo usar?
  - La puedes usar pero debes de explicar cómo la empleas en tu proyecto y especificar la fuente.
- Las carpetas compartidas cuando estarán disponibles?
  - En esta semana se estarán habilitando
- Tengo solo un puerto/cable USB desde la PC. ¿Cómo puedo hacer para los ejercicios?
  - Se usa el mismo cable para el PICKit3 y el USB-TTL, en el momento que se requiera grabar el microcontrolador conectas el PICKit3 y luego de programado intercambias el cable hacia el USB-TTL. Tener en cuenta que en este caso en particular el USB-TTL dará alimentación al protoboard (No hay conflicto de fuentes puesto a que el PICKit3 y el USB-TTL no están conectados al mismo tiempo).

4

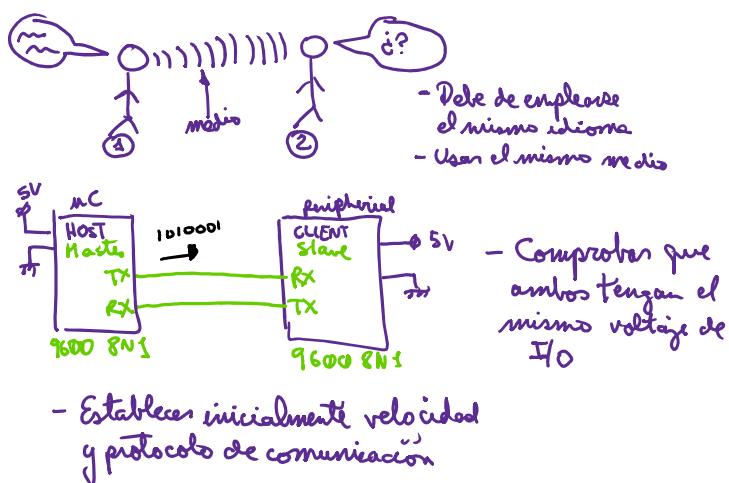
## Agenda

- Comunicación UART
  - Niveles lógicos RS232, TTL, RS485
  - Protocolo de comunicación
- El EUSART del PIC18F4550
  - Generador de baudios
  - Modo transmisión
  - Modo recepción

5

## Comunicación asíncrona UART

- Asíncrona: No hay una señal dedicada de reloj

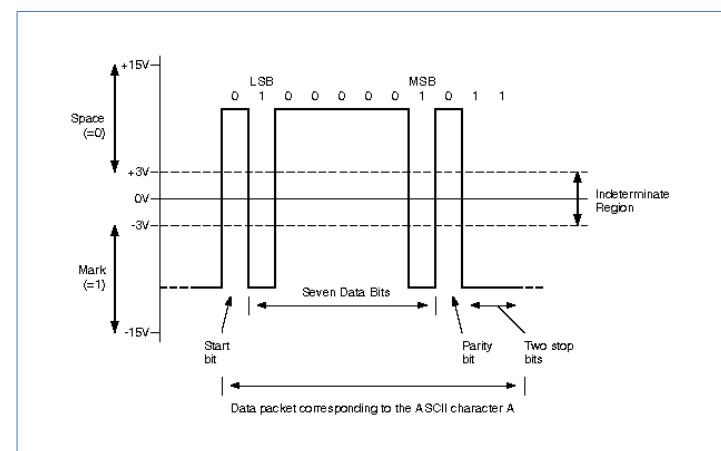
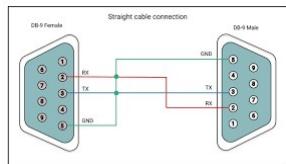
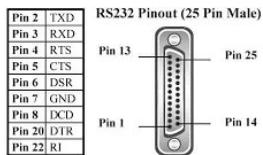


6

## Sobre niveles de voltaje en comunicación UART

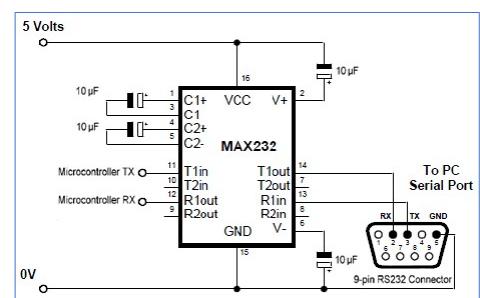
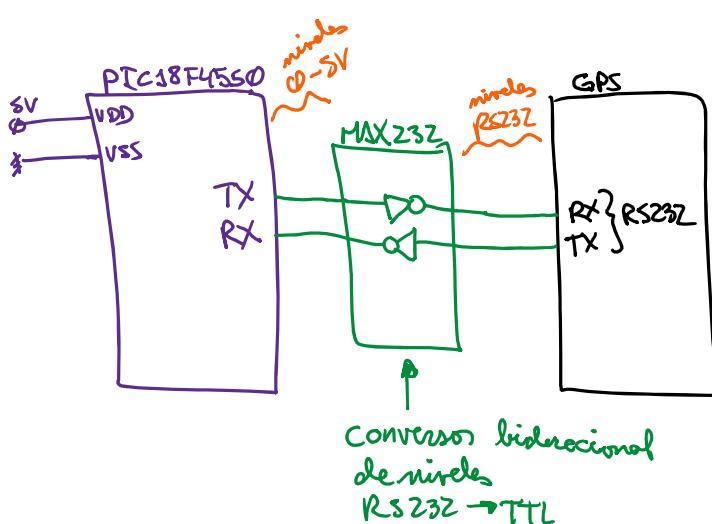
- RS232 (EIA/TIA 232) – Comunicación a distancias medianas (hasta 15 metros)

RS232 25 Pin



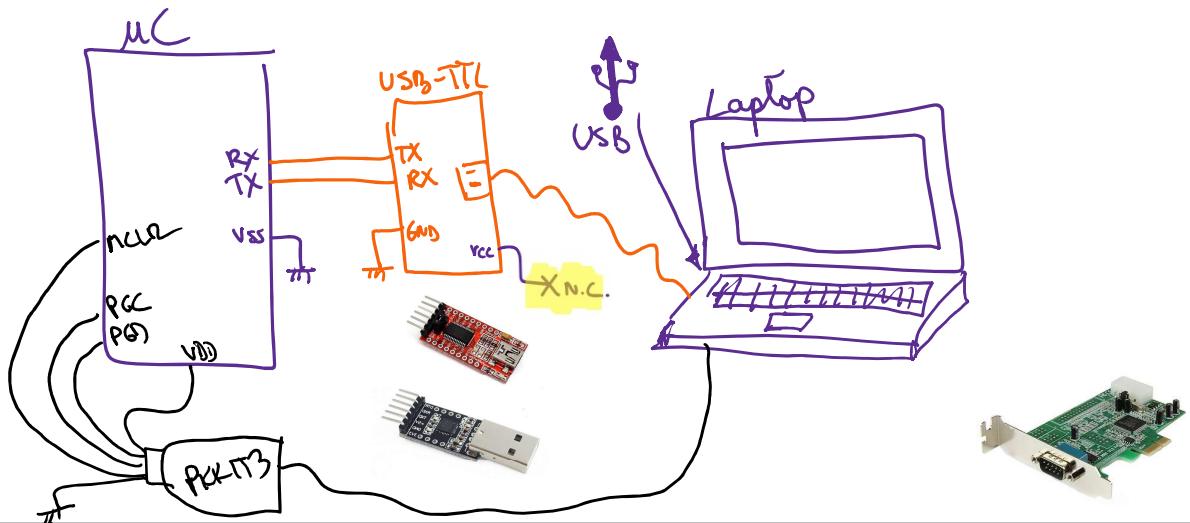
7

## ¿Cómo hago para conectar un dispositivo RS232 al PIC18F4550?



8

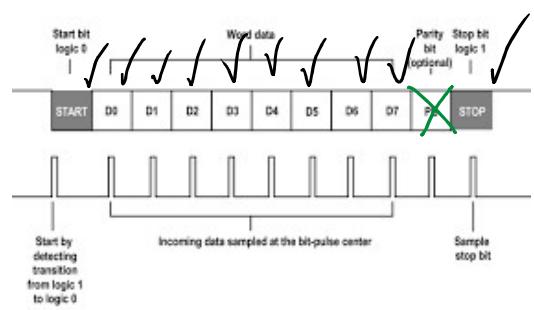
## Cómo conectar un microcontrolador PIC hacia un computador mediante USB



9

## Cálculo de la velocidad en comunicación UART

- Formato completo es
  - Velocidad – Protocolo
  - Ej. 9600 - 8N1
- Protocolo común: 8N1
  - 8: El dato es de 8 bits
  - N: no paridad (O:paridad impar, :paridad par)
  - 1: un bit de stop



- En total se están enviando 10 bits por cada dato de 8 bits

10

## Cálculo de la velocidad en comunicación UART

- Si tengo  $\overset{V_{TX}}{9600}$  8N1:

$$\text{Tiempo de bit: } T_{\text{bit}} = \frac{1}{V_{TX}} = \frac{1}{9600} = 1.04 \times 10^{-4} \text{ s} \\ = 0.1 \text{ ms}$$

- Si para enviar un dato de 8 bits usamos 10 bits:

$$\text{Tiempo para enviar un dato de 8bit} = 1.04 \times 10^{-4} \text{ s} = 1.04 \text{ ms.}$$

11

## Ejemplo de cálculo de cuánto tiempo demora en enviar datos vía comm. serial:

- ¿Cuánto demoro en enviar un archivo de 100KByte por una canal de comm. serial a 9600 8N1?

Recordar que 1 byte = 8 bit, 1Kbyte = 1024 bytes

1º Hallar cuántos bits se va a enviar:

$$100 \times 1024 = 102400 \text{ bytes}$$

Cada byte representará 80 bits enviados  $\Rightarrow 102400 \times 80$  bits enviados

$$\text{Tiempobit}_{9600} = 1.04 \times 10^{-4}$$

$$\text{Tiempo requerido} = 102400 \times 80 \times 1.04 \times 10^{-4} \text{ segundos}$$

12

Ejemplo de cálculo de cuánto tiempo demora en enviar datos vía comm. serial:

- ¿Cuánto demoro en enviar un archivo de música de 4MByte por una canal de comm. serial a 57600 8N1?

5 minutos

13

## El modulo EUSART

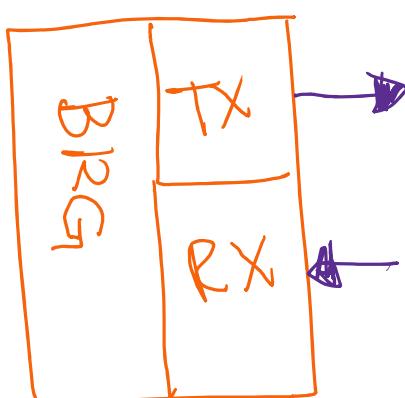


FIGURE 20-3: EUSART TRANSMIT BLOCK DIAGRAM

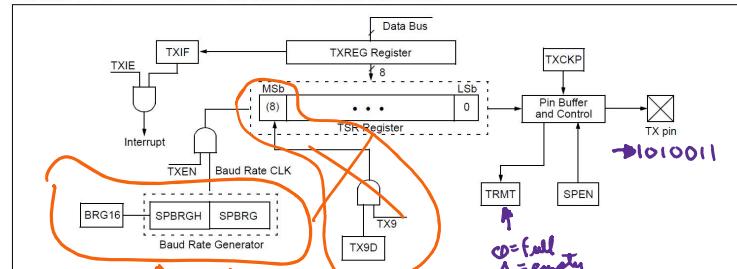
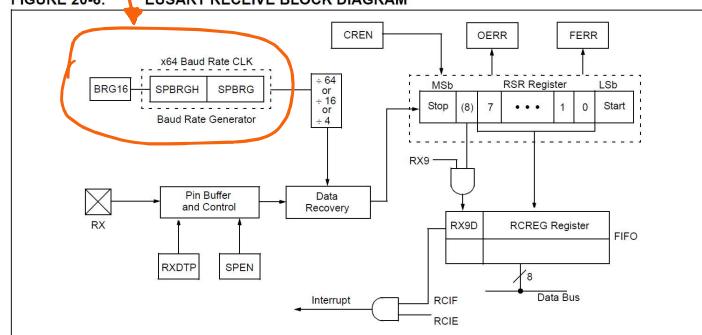


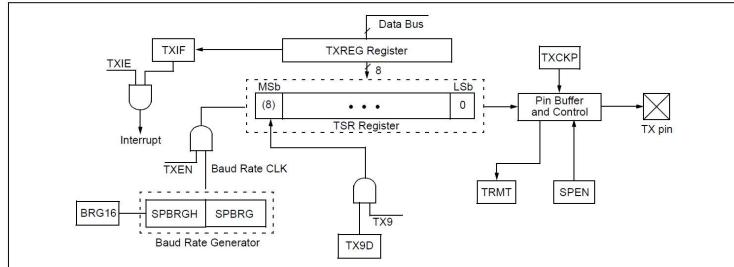
FIGURE 20-6: EUSART RECEIVE BLOCK DIAGRAM



14

## Módulo transmisor del EUSART

FIGURE 20-3: EUSART TRANSMIT BLOCK DIAGRAM



15

### Para transmitir un dato:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If the signal from the TX pin is to be inverted, set the TXCKP bit.
4. If interrupts are desired, set enable bit, TXIE.
5. If 9-bit transmission is desired, set transmit bit, TX9. Can be used as address/data bit.
6. Enable the transmission by setting bit, TXEN, which will also set bit, TXIF.
7. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
8. Load data to the TXREG register (starts transmission).
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.
10. Esperar a que TRMT cambie de estado

Para el cálculo de BRG :

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH	8-bit/Asynchronous	Fosc/[64 (n + 1)]
0	0	0		

Si  $V_{TX} = 9600 \Rightarrow BRG = ?$  despejan 'n'

$$m = \frac{(F_{OSC})}{(Bitrate)} - 1$$

$$m = \frac{48000000}{9600} - 1$$

$$m = 77.125 \rightarrow SPBRG = 77$$

$n = SPBRGH:SPBRG$   
registro 16 bit

pero como bit BRG16 = 0

$SPBRGH = 0$

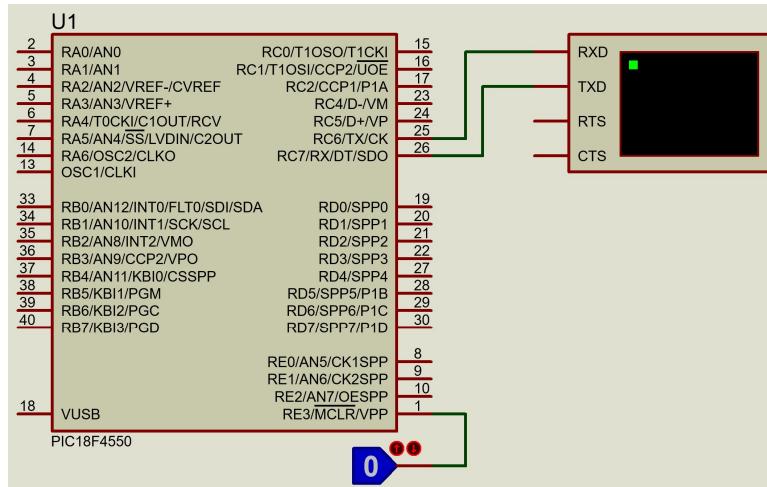
$$\text{Baudrate actual} = \frac{F_{OSC}}{(SPBRG + 1)} = \frac{48000000}{[64(77+1)]} = 9615$$

$$\text{Bitrate error: } \frac{9615 - 9600}{9600} \times 100 = 0.16\%$$

16

## Ejemplo de transmisión de datos vía EUSART

- Enviar "Hola mundo!" a través del módulo EUSART empleando 9600 8N1



17

(cont...)

- Configurar el EUSART para que trabaje a 9600 8N1

Aplicando la primera fórmula:

$$m = \frac{(\text{Fosc})}{\text{Bitrate}} - 1 \quad \text{conf. 8 bits}$$

$$m = \frac{\frac{48\text{E}6}{9600}}{64} - 1$$

$$m = 77.125$$

BRG: ~~SPBRGH~~ SPBRG

18

(cont...)

```

1 #include <xc.h>
2 #include "cabecera.h"
3 #define _XTAL_FREQ 48000000UL
4
5 void EUSART_conf(void){
6     SPBRG = 77;           //Vtx 9600
7     RCSTAbits.SPEN = 1;   //Encendemos el EUSART
8     TXSTAbits.TXEN = 1;   //Encendemos el transmisor del EUSART
9 }

```

```

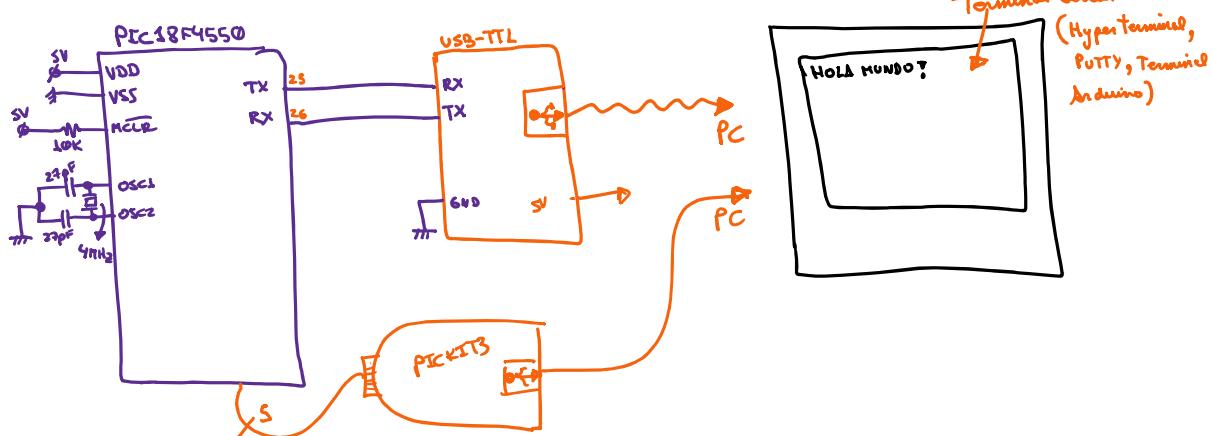
11 void main(void) {
12     EUSART_conf();
13     TXREG = 'H';
14     while (TXSTAbits.TRMT == 0);
15     __delay_ms(100);
16     TXREG = 'o';
17     while (TXSTAbits.TRMT == 0);
18     __delay_ms(100);
19     TXREG = 'l';
20     while (TXSTAbits.TRMT == 0);
21     __delay_ms(100);
22     TXREG = 'a';
23     while (TXSTAbits.TRMT == 0);
24     __delay_ms(100);
25     TXREG = ' ';
26     while (TXSTAbits.TRMT == 0);
27     __delay_ms(100);
28     TXREG = 'm';
29     while (TXSTAbits.TRMT == 0);
30     __delay_ms(100);
31     TXREG = 'n';
32     while (TXSTAbits.TRMT == 0);
33     __delay_ms(100);
34     TXREG = 'd';
35     while (TXSTAbits.TRMT == 0);
36     __delay_ms(100);
37     TXREG = 'o';
38     while (TXSTAbits.TRMT == 0);
39     __delay_ms(100);
40     TXREG = '!';
41     while (TXSTAbits.TRMT == 0);
42     __delay_ms(100);
43     TXREG = '!';
44     while (TXSTAbits.TRMT == 0);
45     __delay_ms(100);
46     while(1);
}

```

19

(cont...)

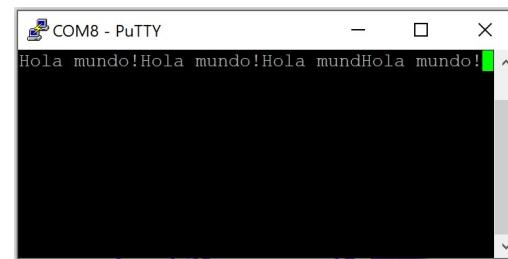
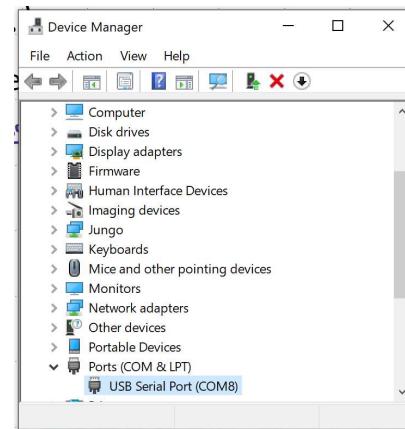
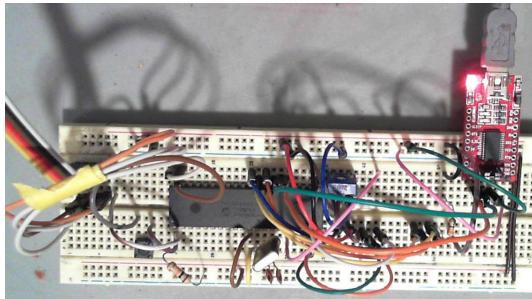
- Implementación física del ejemplo



Link de descarga de drivers FTDI: <http://www.ftdichip.com/FTDrivers.htm>

20

(cont...)



- Al conectar el USB-TTL a la PC el Windows debe de detectar correctamente el módulo.
- Ingresar al Administrador de Dispositivos (Device Manager) para verificar el puerto serial COMx creado
- Utilizar un terminal serial (i.e. PuTTY), colocar el COMx y la velocidad configurada en el EUSART

21

(cont...)

- Optimizando el código:

```

1  #include <xc.h>
2  #include "cabecera.h"
3  #define _XTAL_FREQ 48000000UL
4
5  unsigned char mensaje[]={"Hola mundo!"};
6
7  void EUSART_conf(void){
8      SPBRG = 77;           //Vtx 9600
9      RCSTAbits.SPEN = 1;   //Encendemos el EUSART
10     TXSTAbits.TXEN = 1;  //Encendemos el transmisor del EUSART
11 }
12
13 void main(void) {
14     EUSART_conf();
15     unsigned char x=0;
16     for(x=0;x<11;x++){
17         TXREG = mensaje[x];
18         while (TXSTAbits.TRMT == 0);
19         __delay_ms(100);
20     }
21     while(1);
22 }
```

22

(cont...)

- Parametrizando y creando funciones específicas para la comunicación serial:

```

1  #include <xc.h>
2  #include "cabecera.h"
3  #define _XTAL_FREQ 48000000UL
4
5  const unsigned char mensaje1[]{"Mi nombre es Kalun Lau"};
6  const unsigned char mensaje2[]{"UPC Electronica Mecatronica"};
7
8  void EUSART_conf(void){
9      SPBRG = 77;           //Vtx 9600
10     RCSTAbits.SPEN = 1;   //Encendemos el EUSART
11     TXSTAbits.TXEN = 1;   //Encendemos el transmisor del EUSART
12 }
13
14 void SERIAL_ESCRIBE_MENSAJE(const unsigned char *cadena, unsigned char tam){
15     unsigned char x=0;
16     for(x=0;x<tam;x++){
17         TXREG = cadena[x];
18         while (TXSTAbits.TRMT == 0);
19     }
20 }
21
22 void SERIAL_NEXTLINE(void){
23     TXREG = 0x0A;
24     while (TXSTAbits.TRMT == 0);
25     TXREG = 0x0D;
26     while (TXSTAbits.TRMT == 0);
27 }
28
29 void main(void) {
30     EUSART_conf();
31     SERIAL_ESCRIBE_MENSAJE(mensaje1,22);
32     SERIAL_NEXTLINE();
33     SERIAL_ESCRIBE_MENSAJE(mensaje2,27);
34     while(1);
35 }
```

23

(cont...)

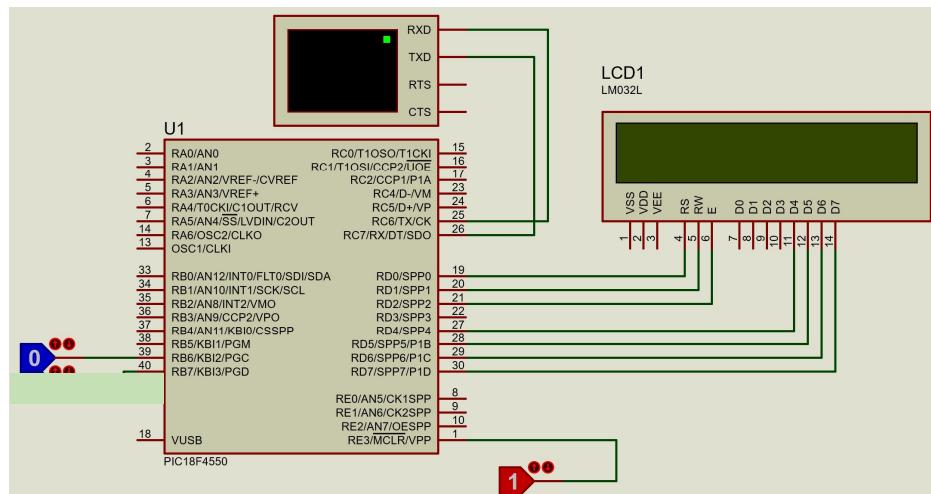
- Empleando la función “strlen” para obtener el numero de caracteres de la cadena y mejorar la función de escritura

```

1  #include <xc.h>
2  #include <string.h>
3  #include "cabecera.h"
4  #define _XTAL_FREQ 48000000UL
5
6  const unsigned char mensaje1[]{"Mi nombre es Kalun Lau"};
7  const unsigned char mensaje2[]{"UPC Electronica Mecatronica"};
8
9  void EUSART_conf(void){
10     SPBRG = 77;           //Vtx 9600
11     RCSTAbits.SPEN = 1;   //Encendemos el EUSART
12     TXSTAbits.TXEN = 1;   //Encendemos el transmisor del EUSART
13 }
14
15 void SERIAL_ESCRIBE_MENSAJE(const unsigned char *cadena){
16     unsigned char tam=0;
17     tam = strlen(cadena);
18     unsigned char x=0;
19     for(x=0;x<tam;x++){
20         TXREG = cadena[x];
21         while (TXSTAbits.TRMT == 0);
22     }
23 }
24
25 void SERIAL_NEXTLINE(void){
26     TXREG = 0x0A;
27     while (TXSTAbits.TRMT == 0);
28     TXREG = 0x0D;
29     while (TXSTAbits.TRMT == 0);
30 }
31
32 void main(void) {
33     EUSART_conf();
34     SERIAL_ESCRIBE_MENSAJE(mensaje1);
35     SERIAL_NEXTLINE();
36     SERIAL_ESCRIBE_MENSAJE(mensaje2);
37     while(1);
38 }
```

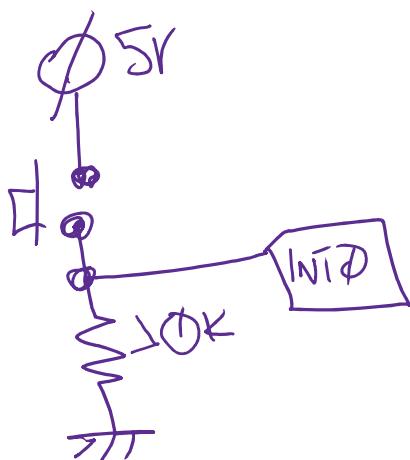
24

## Ejemplo: Detectar el estado de RB6 y visualizarlo a través del terminal serial



25

## Circuito de pulsador activo en bajo para INT0



26

## Código ejemplo:

```

12 #include <xc.h>
13 #define _XTAL_FREQ 48000000UL //frecuencia
14
15 unsigned char mensaje1[] = {"Boton presionado"};
16 unsigned char mensaje2[] = {"Boton soltado "};
17 unsigned char indicador = 0;
18
19 void init_conf(void){
20     TRISChbits.RC6 = 0; //Salida para
21 }
22
23 void EUSART_conf(){
24     SPBRG = 77; //Vtx = 9600
25     RCSTAbits.SPEN = 1; //Habilitamos el
26     TXSTAbits.TXEN = 1; //Habilitamos la
27 }

```

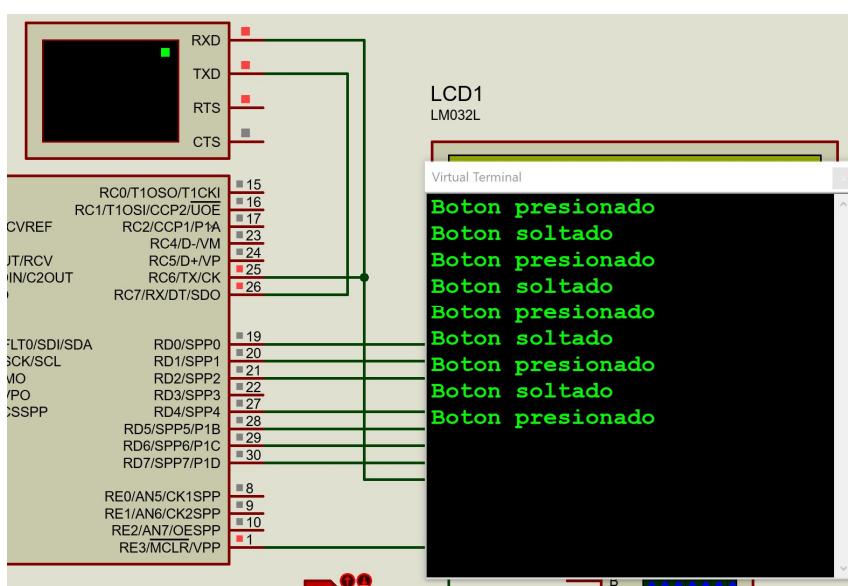
```

29 void main(void){
30     init_conf();
31     EUSART_conf();
32     while(1){
33         if(PORTBbits.RB6 == 1 && indicador == 0){
34             for(unsigned char x=0;x<16;x++){
35                 TXREG = mensaje1[x];
36                 while(TXSTAbits.TRMT == 0); //Esperar a
37             }
38             TXREG = 0x0A; //Comando para nueva linea
39             while(TXSTAbits.TRMT == 0); //Esperar a que
40             TXREG = 0x0D; //Comando para retorno de
41             while(TXSTAbits.TRMT == 0); //Esperar a que
42             indicador = 1;
43         }
44         else if(PORTBbits.RB6 == 0 && indicador == 1){
45             for(unsigned char x=0;x<16;x++){
46                 TXREG = mensaje2[x];
47                 while(TXSTAbits.TRMT == 0); //Esperar a
48             }
49             TXREG = 0x0A; //Comando para nueva linea
50             while(TXSTAbits.TRMT == 0); //Esperar a que
51             TXREG = 0x0D; //Comando para retorno de
52             while(TXSTAbits.TRMT == 0); //Esperar a que
53             indicador = 0;
54         }
55     }
56 }

```

27

## Simulación:



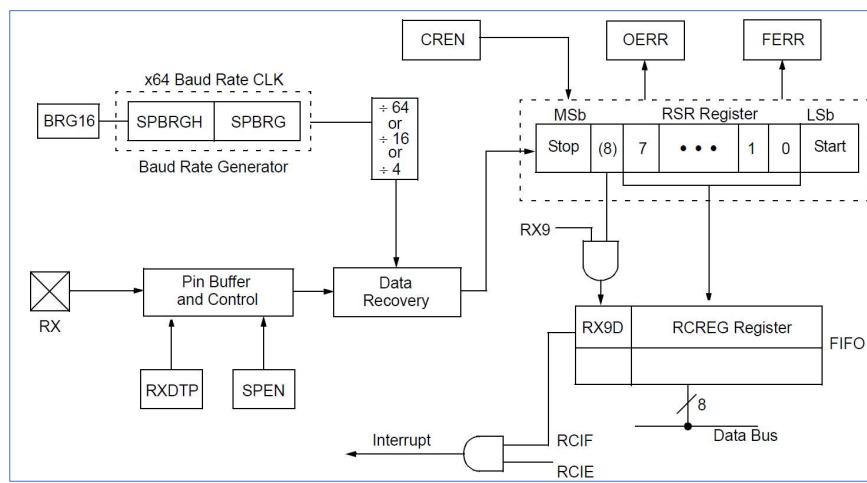
28

## Módulo Receptor en el EUSART

29

## Receptor del EUSART

- Diagrama de bloques



30

## Receptor del EUSART

- Emplea el mismo SPBRG que el transmisor.
- Seguir el procedimiento de configuración de la hoja técnica.
- Se recomienda el uso de interrupciones en esta etapa de recepción ya que el receptor carece de FIFO (memoria temporal).

31

## Receptor del EUSART

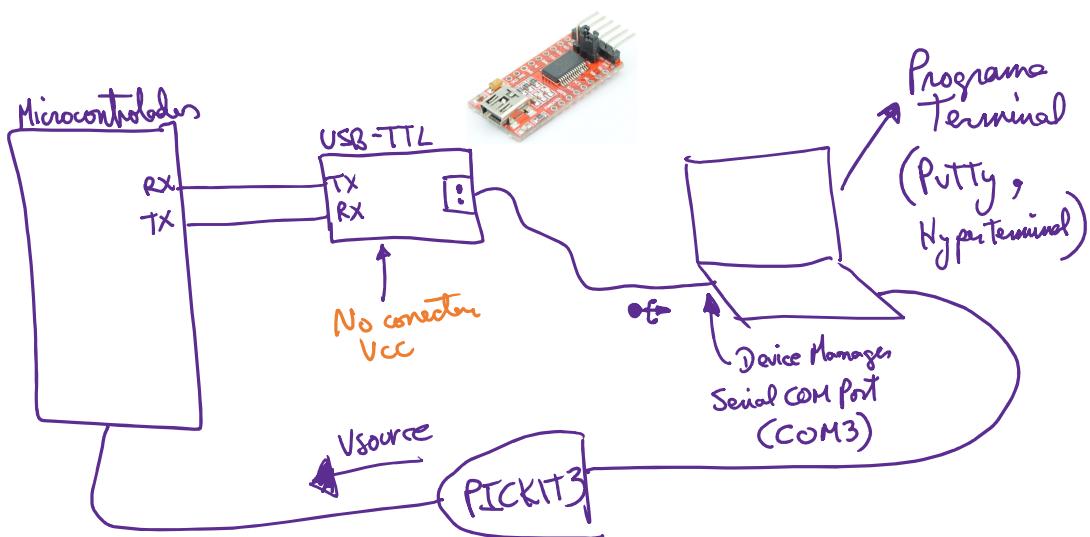
- Procedimiento para recibir un dato (según hoja técnica)

To set up an Asynchronous Reception:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If the signal at the RX pin is to be inverted, set the RXDTP bit.
4. If interrupts are desired, set enable bit, RCIE.
5. If 9-bit reception is desired, set bit, RX9.
6. Enable the reception by setting bit, CREN.
7. Flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCIE, was set.
8. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing enable bit, CREN.
11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

32

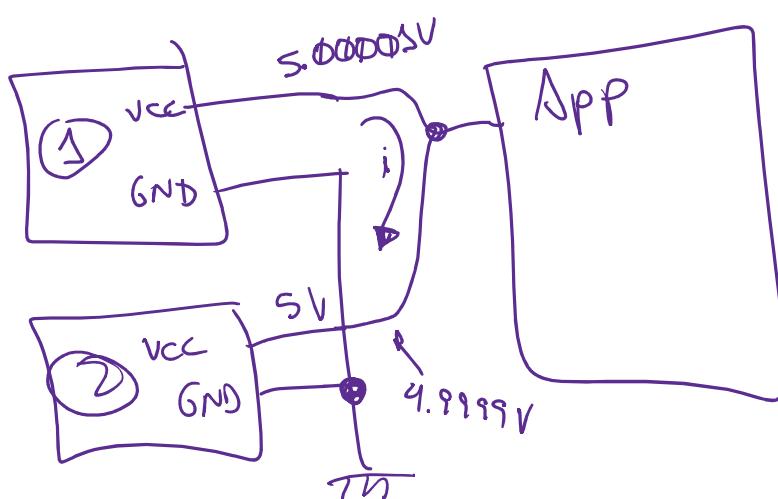
## Ejemplo de comunicación serial UART entre el microcontrolador y un terminal serial en la PC



33

Nota: Tener cuidado en no utilizar fuente de alimentación en paralelo.

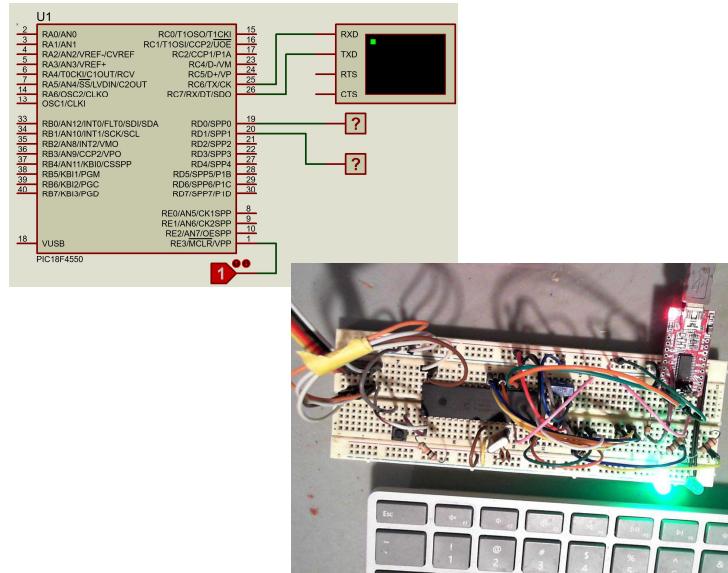
- El pickit3 suministra la alimentación al circuito de prueba, el conversor USB-TTL también tiene una línea de alimentación que viene directo del puerto USB por lo que solo deben de usar una fuente, o la del pickit3 o la del USB-TTL



34

## Ejemplo de transmisión y recepción en EUSART

- Se han conectado dos LEDs, uno en RD0 y otro en RD1.
- Los LEDs serán controlados a través de la comunicación EUSART con el módulo USB-TTL y conectado a la PC, en la PC corriendo un software de terminal serial (PuTTY) y enviando comandos desde el teclado



35

## Ejemplo de transmisión y recepción en EUSART (cont...)

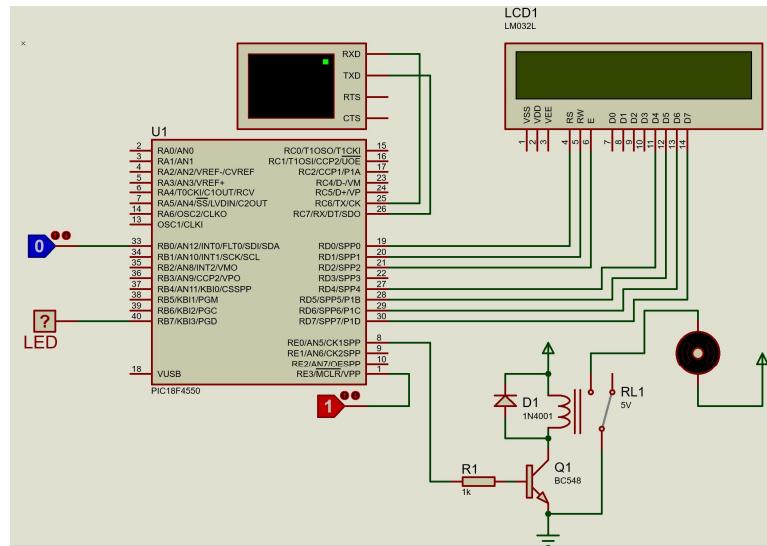
```

1  #include <xc.h>
2  #include <string.h>
3  #include "cabecera.h"
4  #define _XTAL_FREQ 48000000UL
5
6  const unsigned char mensaje1[]="Mi nombre es Kalun Lau";
7  const unsigned char mensaje2[]="UPC Electronica Mecatronica";
8
9  void init_conf(void){
10     INTCONbits.GIE = 1; //interruptor global on
11     INTCONbits.PEIE = 1; //interruptor de perifericos on
12     PIEbits.RCIE = 1; //habilitador de int del receptor del EUSART
13     TRISDbits.RD0 = 0;
14     TRISDbits.RD1 = 0;
15 }
16
17 void EUSART_conf(void){
18     SPBRG = 77; //Vtx 9600
19     RCSTAbits.SPEN = 1; //Encendemos el EUSART
20     TXSTAbits.TXEN = 1; //Encendemos el transmisor del EUSART
21     RCSTAbits.CREN = 1; //Encendemos el receptor del EUSART
22 }
23
24 void SERIAL_ESCRIBE_MENSAJE(const unsigned char *cadena){
25     unsigned char tam=0;
26     tam = strlen(cadena);
27     unsigned char x=0;
28     for(x=0;x<tam;x++){
29         TXREG = cadena[x];
30         while (TXSTAbits.TRMT == 0);
31     }
32 }
33
34 void SERIAL_NEXTLINE(void){
35     TXREG = 0xA;
36     while (TXSTAbits.TRMT == 0);
37     TXREG = 0xD;
38     while (TXSTAbits.TRMT == 0);
39 }
40
41 void main(void) {
42     init_conf();
43     EUSART_conf();
44     SERIAL_ESCRIBE_MENSAJE(mensaje1);
45     SERIAL_NEXTLINE();
46     SERIAL_ESCRIBE_MENSAJE(mensaje2);
47     SERIAL_NEXTLINE();
48     while(1);
49 }
50
51 void __interrupt() EUSART_RX_ISR(void){
52     PIR1bits.RCIF = 0;
53     switch(RCREG){
54     case '1':
55         LATDbits.LDO = 1;
56         SERIAL_ESCRIBE_MENSAJE("Tecla 1 presionada");
57         SERIAL_NEXTLINE();
58         break;
59     case '2':
60         LATDbits.LDO = 0;
61         SERIAL_ESCRIBE_MENSAJE("Tecla 2 presionada");
62         SERIAL_NEXTLINE();
63         break;
64     case '3':
65         LATDbits.LD1 = 1;
66         SERIAL_ESCRIBE_MENSAJE("Tecla 3 presionada");
67         SERIAL_NEXTLINE();
68         break;
69     case '4':
70         LATDbits.LD1 = 0;
71         SERIAL_ESCRIBE_MENSAJE("Tecla 4 presionada");
72         SERIAL_NEXTLINE();
73         break;
74     default:
75         SERIAL_ESCRIBE_MENSAJE("Tecla erronea");
76         SERIAL_NEXTLINE();
77     }
78 }
```

36

## Ejemplo de comunicación UART entre el microcontrolador y un terminal serial

- El PIC18F4550 enviará el menú de opciones vía EUSART hacia el terminal virtual a una tasa de 9600 8N1, se tendrá las opciones de encender el LED conectado en RB7



37

## Ejemplo de comunicación UART entre el microcontrolador y un terminal serial

```

1 #pragma config PLLDIV = 1      // PLL Prescaler Selection
2 #pragma config CPUDIV = OSC1_PLL2 // System Clock Frequency Division
3 #pragma config FOSC = XTPLL_XT // Oscillator Selection
4 #pragma config PWRT = ON       // Power-up Timer Enable
5 #pragma config BOR = OFF        // Brown-out Reset
6 #pragma config WDT = OFF        // Watchdog Timer Enable
7 #pragma config CCP2MX = ON      // CCP2 MUX bit (CCP2Mux)
8 #pragma config PBADER = OFF     // PORTB A/D Enable
9 #pragma config MCIRE = ON       // MCLR Pin Enable
10 #pragma config LVP = OFF       // Single-Supply ICSP
11
12 #include <xc.h>
13 #define _XTAL_FREQ 48000000UL    //frecuencia de oscilación
14
15 unsigned char menu1[] = {"Bienvenidos al ejemplo"};
16 unsigned char menu2[] = {"*Sección opción *"};
17 unsigned char menu3[] = {"*(1) Enciende LED *"};
18 unsigned char menu4[] = {"*(2) Apaga LED *"};
19 unsigned char menu5[] = {"*(m) Muestra el menu *"};
20 unsigned char ledon[] = {"*LED encendido *"};
21 unsigned char ledoff[] = {"*LED apagado *"};
22
23 unsigned char indicador = 0;
24
25 void PORT_config(void) {
26     TRISBbits.RB7 = 0;           //Salida en RB7
27 }
28
29 void EUSART_config(void) {
30     SPBRGH = 0;                //Ignorado debido a la velocidad
31     SPBRG = 77;                //Vtx es 9600 baudios
32     TRISEbits.RE6 = 0;          //Puerto RC6 como salida
33     RCSTAbits.SREN = 1;         //Encendemos el puerto
34     TXSTAbits.TXEN = 1;         //Encendemos el transmisor
35     RCSTAbits.CREN = 1;         //Encendemos el receptor
36 }
37
38 void INT_config(void){
39     INTCONbits.GIE = 1;          //Interruptor global habilitado
40     INTCONbits.PEIE = 1;          //Interruptor de perifericos habilitado
41     PIEbits.RCIE = 1;             //Habilitado interrupcion por recepcion
42 }
43
44 void EUSART_siguientelinea(void){
45     TXREG = 0xA7;               //Enviamos el caracter A
46     while(TXSTAbits.TRMT == 0);   //Esperamos que termine de transmitir
47     TXREG = 0x0D;               //Enviamos el caracter de salto de linea
48     while(TXSTAbits.TRMT == 0);   //Esperamos que termine de transmitir
49 }
50
51 void EUSART_enviacadena(const unsigned char *vector,unsigned char pos){
52     for (unsigned char x=0;x<pos;x++){
53         TXREG = vector[x];           //Enviamos el caracter
54         while(TXSTAbits.TRMT == 0);   //Esperamos que termine de transmitir
55     }
56 }
57
58 void show_menu(void){
59     EUSART_enviacadena(menu1,22);
60     EUSART_siguientelinea();
61     EUSART_enviacadena(menu2,22);
62     EUSART_siguientelinea();
63     EUSART_enviacadena(menu3,22);
64     EUSART_siguientelinea();
65     EUSART_enviacadena(menu4,22);
66     EUSART_siguientelinea();
67     EUSART_enviacadena(menu5,22);
68     EUSART_siguientelinea();
69 }
70
71 void main(void) {
72     PORT_config();
73     EUSART_config();
74     INT_config();
75     show_menu();
76     while(1);
77 }
78
79 void __interrupt(high_priority) RC_Isr(void){
80     if(RCREG == '1'){
81         LATBbits.LB7 = 1;
82         EUSART_enviacadena(ledon,22);
83         EUSART_siguientelinea();
84     }
85     else if(RCREG == '2'){
86         LATBbits.LB7 = 0;
87         EUSART_enviacadena(ledoff,22);
88         EUSART_siguientelinea();
89     }
90     else if(RCREG == 0x6D){
91         show_menu();
92     }
93 }
94
95 }

```

38

## Ejemplo de comunicación UART entre el microcontrolador y un terminal serial

Ejercicios adicionales:

- Agregando una opción para que se ingrese una cadena de caracteres desde el teclado y devuelva por el terminal virtual dicha cadena.
- Cambiando la opción de RB7 por la de RE0 donde esta conectado un relay y un motor DC.