

Apéndice B

Programas y Diagrama de Bloques de las Pruebas con Microcontroladores

Prueba 1

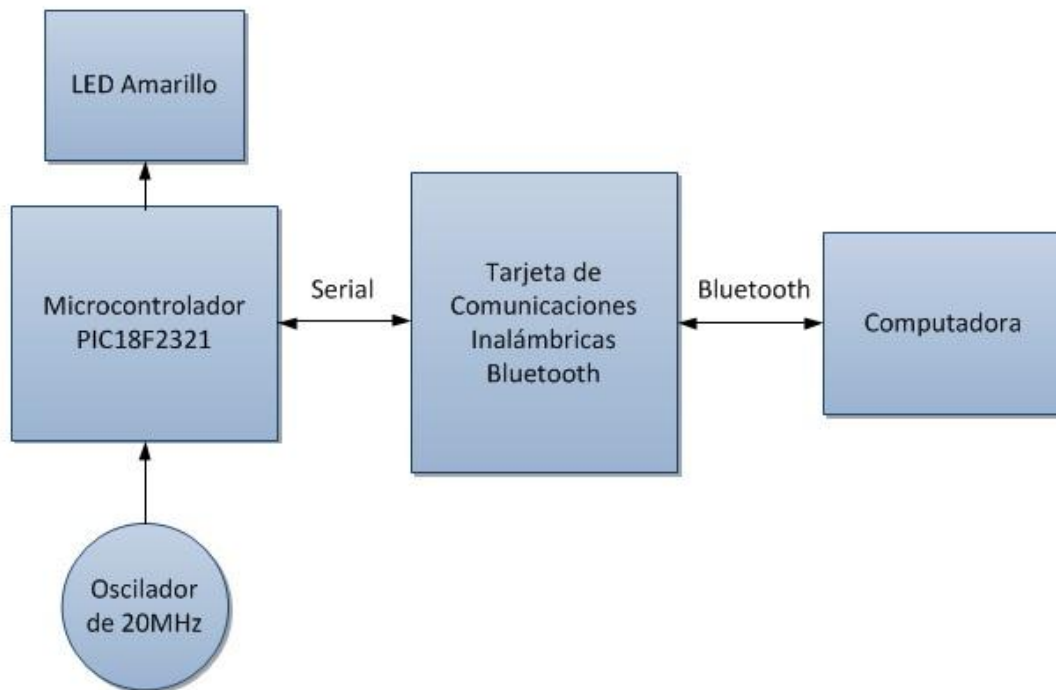


Figura B.1 Diagrama de Bloques de la Conexión Para la Prueba 1.

```
#include <p18f2321.h>
#include <USART.h>
#include <delays.h>
```

```
#pragma config OSC = INTIO1, WDT = OFF, LVP = OFF
```

```

char eco;

void envio (void);
void led (void);

#pragma code envio = 0x08//indica la direccion de memoria donde esta la interrupcion de alta prioridad
void la (void)
{
    _asm GOTO envio _endasm
}
#pragma code
#pragma interrupt envio
void envio (void)//interrupt
{
    PIE1bits.TXIE = 1;//habilita la interrupcion por transmision
    eco = ReadUSART();//lee el dato
    Delay10KTCYx(10);//retardo
    WriteUSART(eco);// pone el eco en el bufer de salida
}

#pragma code led = 0x18//indica la direccion de memoria del registro de baja prioridad
void lb (void)
{
    _asm GOTO led _endasm
}
#pragma code
#pragma interruptlow led
void led (void)//interruptlow
{
    PIE1bits.TXIE = 0;//deshabilita la interrupcion por transmision
    PORTCbits.RC4 ^= 1;//XOR para apagar el LED
}

void main (void)// programa principal
{
    OSCCON |= 0x70;//configura el oscilador a 8 [MHz]
    TRISCbits.TRISC4 = 0;//pin del led
    TRISCbits.TRISC6 = 1;//recepcion
    TRISCbits.TRISC7 = 1;//transmision

    OpenUSART(USART_TX_INT_OFF & //configuracion del puerto serie
              USART_RX_INT_OFF &
              USART_ASYNC_MODE &
              USART_EIGHT_BIT &
              USART_CONT_RX &
              USART_BRGH_HIGH, 51);

    RCONbits.IPEN = 1; //habilita la prioridad de las interrupciones
    INTCONbits.GIEH = 1;//habilita todas las interrupciones altas
    INTCONbits.GIEL = 1;//habilita todas las interrupciones bajas
    PIE1bits.RCIE = 1;//habilita interrupcion por recepcion
    IPR1bits.TXIP = 0;//interrupcion por transmision como baja
    while (1);
}

```

Prueba 2

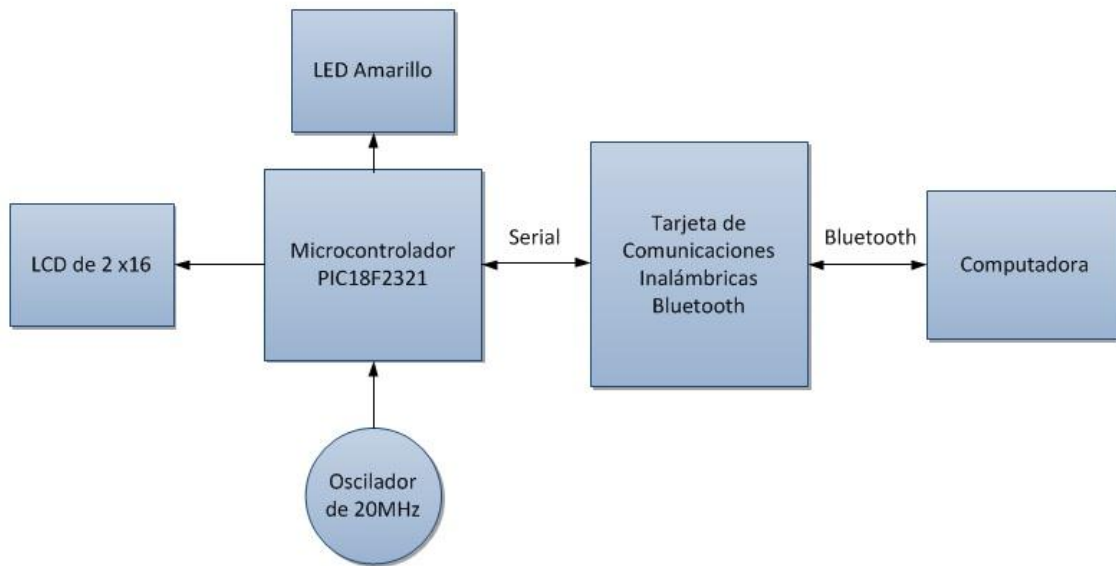


Figura B.2 Diagrama de Bloques de la Conexión Para la Prueba 2.

```

#include<p18f2321.h>
#include<usart.h>
#include<delays.h>
#include<stdlib.h>

#pragma config WDT = OFF, OSC = HS, LVP = OFF

#define RS PORTAbits.RA0
#define RW PORTAbits.RA1
#define DB4 PORTBbits.RB0
#define DB5 PORTBbits.RB1
#define DB6 PORTBbits.RB2
#define DB7 PORTBbits.RB3
#define I PORTCbits.RC3

void led(void);

#pragma code led = 0x08//indica la direccion de la interrupcion de alta prioridad
void la (void)
{
    _asm GOTO led _endasm
}
#pragma code
#pragma interrupt led
void led (void)//interrupcion para prender y pagar el led
{
    I ^= 1;
    INTCONbits.TMR0IF = 0;
}
  
```

```
#pragma code

void inicia_LCD (void);
void letra (char dato);
void E (void);

int tr = 6;
char eco;//variable que regresa lo escrito

void main (void){

    INTCONbits.PEIE = 1;//Habilita las interrupciones de los perifericos
    INTCONbits.GIE = 1;//Habilita las interrupciones
    INTCONbits.TMR0IE = 1;//Habilita la interrupcion por overflow

    TRISAbits.TRISA0 = 0;
    TRISAbits.TRISA1 = 0;
    TRISAbits.TRISA2 = 0;
    TRISBbits.TRISB0 = 0;//lcd 4 bits
    TRISBbits.TRISB1 = 0;
    TRISBbits.TRISB2 = 0;
    TRISBbits.TRISB3 = 0;
    TRISCbits.TRISC3 = 0;//led
    TRISCbits.TRISC6 = 1;//habilita puerto para Transmision
    TRISCbits.TRISC7 = 1;//habilita puerto para Recepcion

    Delay1KTCYx(80);//retardo adicional de mas de 15[ms]

    OpenUSART(USART_TX_INT_OFF &
              USART_RX_INT_OFF &
              USART_ASYNCH_MODE &
              USART_EIGHT_BIT &
              USART_CONT_RX &
              USART_BRGH_HIGH, 129);//configuracion de USART

    ADCON1 |= 0x0F;//salidas digitales en PORTA
    PORTA = 0;
    PORTB = 0;
    inicia_LCD();
    TOCON = 0x94;
    I = 1;//prende el led
    while (1)
    {
        while(!DataRdyUSART());//ve si hay dato en RCREG
        eco=ReadUSART();//lee el dato
        letra(eco);//pone la letra en el lcd
        WriteUSART(eco);//lo pone en el registro de salida
        while(BusyUSART());//espera a que se desocupe el registro de salida
    }
    CloseUSART();
}

void inicia_LCD (void){
    RS = 0;
    RW = 0;
```

```
Delay1KTCYx(80); // retardo de mas de 15ms
```

```
PORTB = 0x03;//1  
E();  
Delay100TCYx(tr);
```

```
Delay10TCYx(3);//retardo de mas de 4.7 us
```

```
PORTB = 0x03;//2  
E();  
Delay100TCYx(tr);
```

```
Delay100TCYx(6);//retardo de mas de 100 us
```

```
PORTB = 0x02;//3  
E();  
Delay100TCYx(tr);
```

```
PORTB = 0x02;//4  
E();  
Delay100TCYx(tr);  
PORTB = 0x08;  
E();  
Delay100TCYx(tr);  
PORTB = 0x00;  
E();  
Delay100TCYx(tr);  
PORTB = 0x0C;  
E();  
Delay100TCYx(tr);  
PORTB = 0x00;  
E();  
Delay100TCYx(tr);  
PORTB = 0x01;  
E();  
Delay1KTCYx(80);
```

```
PORTB = 0x00;  
E();  
Delay100TCYx(tr);  
PORTB = 0x06;  
E();  
Delay100TCYx(tr);
```

```
//initialization end
```

```
PORTB = 0x00;//display on/off control  
E();  
Delay100TCYx(tr);  
PORTB = 0x0C;  
E();  
Delay100TCYx(tr);
```

```
PORTB = 0x08;//posicion cursor  
E();
```

```
    Delay100TCYx(tr);
    PORTB = 0x05;
    E();
    Delay100TCYx(tr);
}

void letra (char dato){
    RS = 1;
    PORTB = dato >> 4;
    E();
    Delay100TCYx(tr);
    PORTB = dato;
    E();
    Delay100TCYx(tr);
    RS = 0;
}

void E(void)
{
    PORTAbits.RA2 = 1;
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    PORTAbits.RA2 = 0;
}
```