```
/*
 * File:   Lab03.c
 * Author: Kenneth Aldana
 * Carnet: 18435
 * Pseudocódigo
 *
 * Created on January 31, 2021, 9:56 PM
 */



#include <xc.h>
#include <stdint.h>

//Aquí también incluyo mis librearías que utilizaré en el código

// PIC16F887 Configuration Bit Settings

// 'C' source line config statements

// CONFIG1
#pragma config FOSC = XT        // Oscillator Selection bits (XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN)
#pragma config WDTE = OFF       // Watchdog Timer Enable bit (WDT disabled and can be enabled by SWDTEN bit of the WDTCON register)
#pragma config PWRTE = OFF      // Power-up Timer Enable bit (PWRT disabled)
#pragma config MCLRE = OFF      // RE3/MCLR pin function select bit (RE3/MCLR pin function is MCLR)
#pragma config CP = OFF         // Code Protection bit (Program memory code protection is disabled)
#pragma config CPD = OFF        // Data Code Protection bit (Data memory code protection is disabled)
#pragma config BOREN = OFF      // Brown Out Reset Selection bits (BOR disabled)
```

```
#pragma config IESO = OFF       // Internal External Switchover bit (Internal/External Switchover mode is disabled)

#pragma config FCMEN = OFF      // Fail-Safe Clock Monitor Enabled bit (Fail-Safe Clock Monitor is disabled)

#pragma config LVP = OFF        // Low Voltage Programming Enable bit (RB3 pin has digital I/O, HV on MCLR must be used for programming)


// CONFIG2

#pragma config BOR4V = BOR40V   // Brown-out Reset Selection bit (Brown-out Reset set to 4.0V)

#pragma config WRT = OFF        // Flash Program Memory Self Write Enable bits (Write protection off)


// #pragma config statements should precede project file includes.

// Use project enums instead of #define for ON and OFF.


#include <xc.h>

// incluir la librerias necesarias, en este caso seran las del ADC


//************************************************************************************

// Variables

//
************************************************************************************


#define _XTAL_FREQ 8000000


//************************************************************************************

//Definir funciones

//************************************************************************************
```

```
//Aqui voy a declarer las variables que vaya a utilizar

//*****************************************************************************
***********
//Configuracion de puertos
//*****************************************************************************
***********
void Setup(void){

   ANSEL = 0b00000011;
   ANSELH = 0;

   TRISA = 0b00000011;
   PORTA = 0;

   //Declaro como entrada el RX
   TRISC = 0b0100000;
   PORTC = 0;

   TRISD = 0;
   PORTD = 0;



   INTCON = 0b11101000;
   IOCB = 0b00000011;
   //Aqui debo apagar el bit 4 que es la bandera de la comunicacion serial
   PIR1 = 0b00000000;
   //Aqui debo encender el bit 4 para encender el bit enable de la comunicacion serial
   PIE1 = 0b01000000;
```

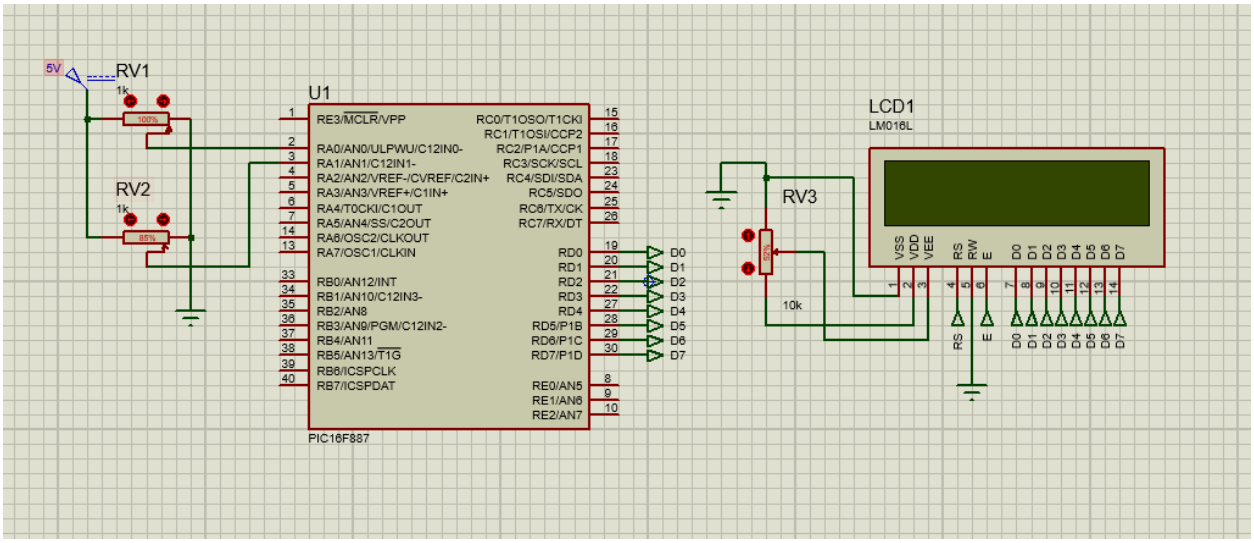
```
   //Debo trabajar con el registro TXSTA para ver los bits que debo encender
}

//*****************************************************************************
//Interrupciones
//*****************************************************************************

//Aqui debo trabajar la interrupcion de la comunicacion serial, tomando en cuenta
//si la hare sincrona o asincrona
//Configurar interrupcion de transmision
//Configurar interrupcion de recepcion
//Configurar USART mode
//Configurar Transmission width
//Configurar Maestro esclavo
//Configurar la recepcion
//Configurar el baud rate
//Configurar el address detect

//*****************************************************************************
//Principal
//*****************************************************************************
void main(void) {
   //En este bloque debo trabajar con el tema de la comunicacion serial
}
//*****************************************************************************
//                      FIN DEL PROGRAMA
//*****************************************************************************
```

PROTEUS


5V
RV1
1k
100%
RV2
1k
85%
U1
RE3/MCLR/VPP
RA0/AN0/ULPWU/C12IN0-
RA1/AN1/C12IN1-
RA2/AN2/VREF-/CVREF/C2IN+
RA3/AN3/VREF+/C1IN+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
RA6/OSC2/CLKOUT
RA7/OSC1/CLKIN
RB0/AN12/INT
RB1/AN10/C12IN3-
RB2/AN8
RB3/AN9/PGM/C12IN2-
RB4/AN11
RB5/AN13/T1G
RB6/ICSPCLK
RB7/ICSPDAT
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/P1A/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0
RD1
RD2
RD3
RD4
RD5/P1B
RD6/P1C
RD7/P1D
RE0/AN5
RE1/AN6
RE2/AN7
PIC16F887
D0
D1
D2
D3
D4
D5
D6
D7
RV3
10k
LCD1
LM016L
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7