

MAESTRO

```
/*
 * File: Master_main.c
 * MINI PROYECTO
 */
Incluyo las librerias
necesarias para controlar
programar este PIC. Siendo
las del SPI, LCD y USART.

Configuro los bits para el
funcionamiento del PIC

//*****

// Variables

//
*****

//*****

//Definir funciones

//*****

//*****

//Definir variables
```

```
//*****

//*****

//Configuracion de puertos

//*****

//Llamo a la funcion del SPI
master para configurar el SPI
y el puerto C

"FUNCION SPI();"
}

//*****

//Principal

//*****

Void principal :

Llamo a las configuraciones
de los puertos, LCD y USART

Setup();

LCD_Init();

LCD_Clear();

Set_BaudRate();

Init_Trans();
```

```
Init_Receive();

while (1) {

    Aquí empiezo a tomar
    los datos de la comunicación
    SPI, para ello debo tomar
    datos de un bufer y luego
    seguir al otro.

    //meto a una variable
    tipo float el valor del
    potenciómetro y mapeo

    //meto los valores de los
    esclavos a floats y lo envío a
    la LCD y USART estos datos.

}

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

//                               FIN DEL
PROGRAMA

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

ESCLAVO 1

```

/*
 * File: Slave0_main.c
 * Author: Kenneth Aldana
 * Carnet: 18435
 Incluyo todas las librerías
 necesarias para poder
 manejar la operación de este
 PIC, para ello llamo las
 librerías del ADC, SPI
 Configuro el PIC para que
 funcione correctamente.
//*****

// Variables

//
*****

//*****

//Definir funciones

//*****

//*****

//Definir variables

//*****

//variables para obtener el
dato del ANSELH en la
conversion AC

//*****

//Configuracion de puertos

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

void Setup(void){

//Configuracion de puertos

//Llamo a la libreria de la
interrupcion del ADC

ADC_CONFIG();

//Llamo a la liberaria del
SPI

CONFIG_SPI();

}

//*****

//Interrupciones

//*****

//Interrupcion del ADC, en
donde obtengo el dato del
ANSELH

//Interrupcion para el
envio de datos del SPI

//*****

//Principal

//*****

void main(void) {

// tiempo de espera y
encendido del bit GO para la
conversion ADC

__delay_us(25);
ADCON0bits.GO_DONE
= 1;

while (1) {

}

}

//*****

//
FIN DEL
PROGRAMA

//*****

```

ESCLAVO 2

```
/*
* File: Slave1_main.c
* Author: Kenneth Aldana
* Carnet: 18435

Incluyo todas las librerias
necesarias para el
funcionamiento de este
esclavo, estas librerías son el
ADC, SPI y la interrupción del
puertoB

Configuro el PIC para que
funcione correctamente

//*****

// Variables

//
*****

//*****

//Definir funciones

//*****

//*****
```

```
//Definir variables

//*****

//variables para manejar el
SPI y el contador

//*****

//Configuracion de puertos

void Setup(void){

// Configuracion de los
puertos

//Llamo a la libreria de la
interrupcion del puerto B

INT_PORTB_CONFIG ();

//Llamo a la libreria del SPI

CONFIG_SPI();

}

//*****

//Interrupciones

//*****
```

```
//interrupcion para
manejar el contador
ascendente y descendente

//interrupcion para el envio
de datos del SPI

//*****

//Principal

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

void main(void) {

//llamar a la funcion de
configuracion de pines

//tiempo de espera y
encendido del bit GO para la
conversion AC

//*****

// FIN DEL
PROGRAMA

//*****
```

ESCLAVO 3

```
/*
* File: Slave2_main.c
* Author: Kenneth Aldana
* Carnet: 18435

Llamo todas las librerias para
que pueda realizar las
operaciones de este esclavo,
siento estas las del SPI y ADC.

Configuracion del PIC16F887
//*****

// Variables
//
*****

//*****

//Definir funciones
//*****

//*****

//Definir variables
//*****

//variables para la optencion
del adc del sensor LM35

//*****

//Configuracion de puertos
//*****

void Setup(void){

//configuracion de puertos

//llamo a la funcion de la
libreria del adc para
configurarlo

//llamo a la libreria del SPI

}

//*****

//Interrupciones
//*****

//Interrupcion del ADC,
para obtener el dato del
ANSELH

//Interrupcion del SPI para
el envio del dato por el SPI

//*****

//Principal
//*****

//Envio el dato al puerto
B para comprobar que si este
trabajando bien

//Condicionales para
que funcione correctamente
el semaforo, en este
apartado

//no hice ningun mapeo,
solo compruebo que bits se
encendian de forma binaria

//en el puerto B y asi
coloque las condiciones.
```