



Universidad Tecnológica del Perú

Facultad de Ingeniería Electrónica y Mecatrónica

Curso : Microcontroladores

Profesor : Domínguez Jesús

Laboratorio Nro : 4

Tema : Laboratorio Dirigido 4

Integrantes : Guerrero Isuiza, Mateo

Fecha del Experimento : Martes, 23 de Octubre del 2018

Hora : De 9:30 am a 11:00 am

Fecha de entrega del informe : Martes, 24 de Octubre del 2018

Hora : De 8:00 am a 6:00 pm

2018 - III



Contenido

Introducción	3
Capítulo 1: Compendio Teórico	4
1.1. Lenguaje C	4
1.2. MPLAB	4
1.3. PicKit	4
1.4. Copilador XC8	4
1.5. Microcontrolador	4
1.7. LCD	5
Capítulo 2: Resultados Obtenidos	6
2.1. Indicador de frases por combinación	6
2.1.1. Diagrama de flujo	6
2.1.2. Código	6
2.1.3. Lenguaje y Simulación.....	10
2.1.4. Circuito Real	14
2.2. Indicador de frases por combinación	16
2.2.1. Diagrama de flujo	16
2.2.2. Código	16
2.2.3. Lenguaje y Simulación.....	17
2.2.4. Circuito Real	18
Link del Proyecto	19
Observaciones	20
Conclusiones	21
Bibliografía	22

Introducción

El interés del grupo de investigación se basó a partir de la observación de los distintos programas que se realizaron en las clases de Microcontroladores, donde se puede generar mensajes en un LCD a partir del PIC18F4550. En consecuencia, nos planteamos la interrogante: ¿De qué manera es más eficaz el uso del PIC18F4550 para escribir mensajes en una pantalla LCD? Consideramos que la manera más eficaz y sencilla de hacer esto es usando librerías pre-diseñadas del LCD.

De esta manera, el informe se dividió en cinco partes. En primer lugar se detallan los conceptos de los sistemas o circuitos usados en el proyecto. En segundo lugar se presenta la elaboración del proyecto: El diagrama de flujo, el código de programación en Lenguaje C, las fotos de las simulaciones en Mplab y Proteus, y finalmente las fotos del circuito real programado con el PicKIT 3. En tercer lugar, se muestra el link del video en Youtube. En cuarto lugar, se muestra las observaciones obtenidas en el proyecto. Finalmente, se muestran las conclusiones.

La importancia del proyecto radica en la relevancia de las aplicaciones del PIC para generar saltos en rutinas, es decir, será posible generar programas capaces de reconocer u evitar posibles fallos y situaciones especiales en la programación. Siendo esta proyecto un medio eficaz para evidenciar que el PIC18F4550 puede ser un medio alterno de circuitos de división de frecuencia. Del mismo modo, el proyecto me permite desarrollar habilidades para la programación y construcción de sistemas eléctricos que serán de gran utilidad en mi carrera profesional como Ingeniero egresado de UTP.

Capítulo 1: Compendio Teórico

1.1. Lenguaje C

El lenguaje C, es un lenguaje que revoluciono ya que debido a su sencillez, y a su tamaño, pero que sobre todo no está dentro de una aplicación en específica, lo hace más potente. C trabaja con tipos de datos que son directamente tratables por el hardware de la mayoría de computadoras actuales, como son los caracteres, números y direcciones. Estos tipos de datos pueden ser manipulados por las operaciones aritméticas que proporcionan las computadoras.

1.2. MPLAB

Este es un programa que sirve para editar distintos micro controladores que este mismo programa puede soportar, sin embargo una de las cosas importantes es que se puede grabar circuitos integrados. Asimismo cabe recalcar que se debe a empezar a escribir el programa, respetando la directiva y/o parámetros necesarios para compilarlo y grabarlo al chip. Definimos directivas como palabras específicas o palabras que han sido separadas para ordenarle al Mplab que funciones debe configurar a la hora de compilar nuestro programa.

1.3. PicKit

El pickit hace posible la programación de micro controlador utilizando todo lo que incluye en el armado del código en el Mplab, desarrollado por la empresa Microchip. Este pickit es muy fácil de conectar ya que solo se conecta mediante una interfaz USB.

1.4. Copilador XC8

Un compilador es un programa informático que permite que el lenguaje avanzado usado por el programador (Lenguaje C, BASIC, etc) sea traducido al lenguaje de máquina, el único que puede leer el microcontrolador. El compilador XC8 es uno de los compiladores de la línea de compiladores MPLAB XC de Microchip, el cual es compatible con los microcontroladores PIC de 8 bits, utilizado en este proyecto.

1.5. Microcontrolador

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y

ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.

Inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, la cual se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de manera indistinta.

Después, se empezó a realizar microcontroladores con la nueva arquitectura Harvard, la cual se caracteriza por disponer de dos memorias independientes para datos y para instrucciones, permitiendo accesos simultáneos.

Estos microcontroladores necesitan ser programados para poder realizar las funciones. Debido a la gran dificultad de programar los microcontroladores en el código binario, la programación comúnmente se lleva a cabo a través de un lenguaje de alto nivel, es decir, un lenguaje que utilice frases o palabras semejantes o propias del lenguaje humano. Lenguajes como el C, utilizado en nuestro caso, o BASIC son comúnmente utilizados en la programación de los microcontroladores.

1.6. Display 7 segmentos

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.

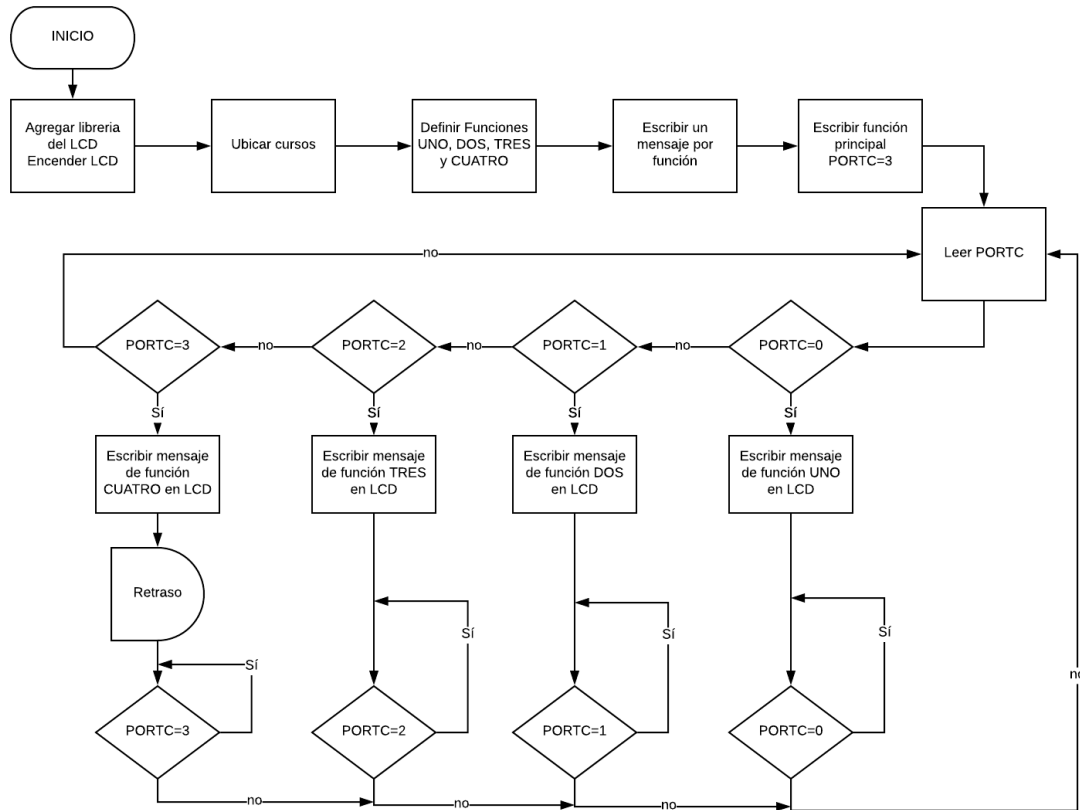
1.7. LCD

Una pantalla de cristal líquido o LCD (sigla del inglés Liquid Crystal Display) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. A menudo se utiliza en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.

Capítulo 2: Resultados Obtenidos

2.1. Indicador de frases por combinación

2.1.1. Diagrama de flujo



2.1.2. Código

```

#include <xc.h>
#include "Configuracion.h"
#include "LCD.h"
#include <stdio.h>
#include <stdint.h>
#define _XTAL_FREQ 4000000
  
```

```

void uno(void)
{
    OpenXLCD(FOUR_BIT & LINES_5X8);
    while(BusyXLCD());
    WriteCmdXLCD(0x06);
    WriteCmdXLCD(0x0C);
  
```



```
LCD_gotoXY(0,0);
putsXLCD("0");
return;
}
void dos(void)
{
    OpenXLCD(FOUR_BIT & LINES_5X8);
    while(BusyXLCD());
    WriteCmdXLCD(0x06);
    WriteCmdXLCD(0x0C);
    LCD_gotoXY(0,0);
    putsXLCD("1");
    LCD_gotoXY(0,2);
    putsXLCD("2");
    LCD_gotoXY(0,4);
    putsXLCD("3");
    LCD_gotoXY(0,6);
    putsXLCD("a");
    LCD_gotoXY(0,8);
    putsXLCD("b");
    LCD_gotoXY(0,10);
    putsXLCD("c");
    LCD_gotoXY(0,12);
    putsXLCD("A");
    LCD_gotoXY(0,14);
    putsXLCD("B");

    LCD_gotoXY(1,0);
    putsXLCD("@");
    LCD_gotoXY(1,2);
    putsXLCD("#");
    LCD_gotoXY(1,4);
    putsXLCD("!");
    LCD_gotoXY(1,6);
```



```
    putsXLCD("%");
    LCD_gotoXY(1,8);
    putsXLCD(";");
    LCD_gotoXY(1,10);
    putsXLCD("=");
    LCD_gotoXY(1,11);
    putsXLCD("(");
    LCD_gotoXY(1,14);
    putsXLCD("=");
    LCD_gotoXY(1,15);
    putsXLCD(")");
    return;
}

void tres(void)
{
    OpenXLCD(FOUR_BIT & LINES_5X8);
    while(BusyXLCD());
    WriteCmdXLCD(0x06);
    WriteCmdXLCD(0x0C);
IN: LCD_gotoXY(0,2);
    putsXLCD("Universidad");
    LCD_gotoXY(1,1);
    putsXLCD("Tecn. del Peru ");
    __delay_ms(1000);

    LCD_gotoXY(0,4);
    putsXLCD("Curso de");
    LCD_gotoXY(1,1);
    putsXLCD("uControladores");
    __delay_ms(1000);
    goto IN;
    return;
}

void cuatro(void)
```

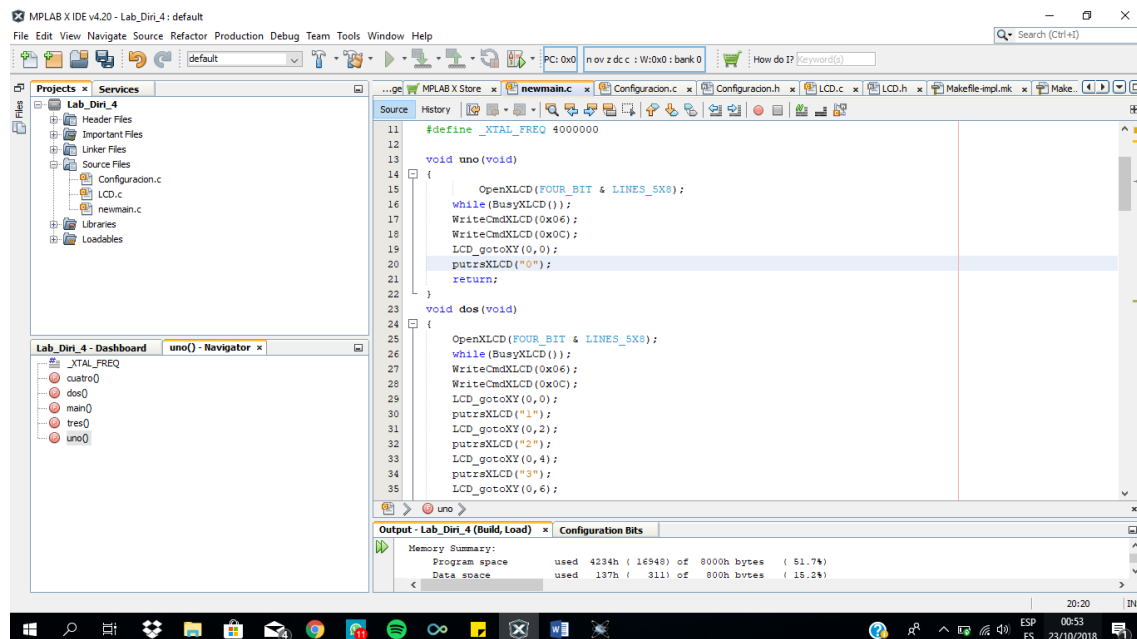



```
{  
    int dato[17];  
    int dato1=246;  
    float dato2=35.42;  
  
    OpenXLCD(FOUR_BIT & LINES_5X8);  
    while(BusyXLCD());  
    WriteCmdXLCD(0x06);  
    WriteCmdXLCD(0x0C);  
  
    sprintf(dato,"Valor 1 = %d",dato1);  
    LCD_gotoXY(0,0);  
    putsXLCD(dato);  
    sprintf(dato,"Valor 2 = %.2f",dato2);  
    LCD_gotoXY(1,0);  
    putsXLCD(dato);  
    return;  
}  
  
void main(void)  
{  
    PORTC=3;  
    LEE:  if(PORTC==0) goto UNO;  
         if(PORTC==1) goto DOS;  
         if(PORTC==2) goto TRES;  
         if(PORTC==3) goto CUATRO;  
    goto LEE;  
    UNO:  uno();  
    LEE2: if(PORTC==0) goto LEE2;  
         goto LEE;  
    DOS:  dos();  
    LEE3: if(PORTC==1) goto LEE3;  
         goto LEE;  
    TRES: tres();
```

```
LEE4:if(PORTC==2) goto LEE4;
    goto LEE;
CUATRO: cuatro();
LEE5:if(PORTC==3) goto LEE5;
    goto LEE;
}
```

2.1.3. Lenguaje y Simulación

A continuación se muestra el código escrito en MPLAB V4.20:





MPLAB X IDE v4.20 - Lab_Diri_4: default

File Edit View Navigate Source Refactor Production Debug Team Tools Window Help

Search (Ctrl+F)

default

PC: 0x0 nov z dc c : W:0x0 : bank 0 How do I? Keyword(s)

Projects Services

Lab_Diri_4

- Header Files
- Important Files
- Linker Files
- Source Files
 - Configuration.c
 - LCD.c
 - newmain.c
- Libraries
- Loadables

Lab_Diri_4 - Dashboard uno() - Navigator

- _XTAL_FREQ
- cuatro()
- dos()
- main()
- tres()
- uno()

Source History

```
35 LCD_gotoXY(0,6);
36 putraXLCD("a");
37 LCD_gotoXY(0,8);
38 putraXLCD("b");
39 LCD_gotoXY(0,10);
40 putraXLCD("c");
41 LCD_gotoXY(0,12);
42 putraXLCD("d");
43 LCD_gotoXY(0,14);
44 putraXLCD("e");
45
46 LCD_gotoXY(1,0);
47 putraXLCD("f");
48 LCD_gotoXY(1,2);
49 putraXLCD("g");
50 LCD_gotoXY(1,4);
51 putraXLCD("h");
52 LCD_gotoXY(1,6);
53 putraXLCD("i");
54 LCD_gotoXY(1,8);
55 putraXLCD("j");
56 LCD_gotoXY(1,10);
57 putraXLCD("k");
58 LCD_gotoXY(1,11);
59 putraXLCD("l");
```

Output - Lab_Diri_4 (Build, Load) Configuration Bits

Memory Summary:

Program space	used	4234h (16948) of 8000h bytes (51.7%)
Data space	used	137h (311) of 800h bytes (16.2%)

15:40 ESP 00:53 22/10/2018

MPLAB X IDE v4.20 - Lab_Diri_4: default

File Edit View Navigate Source Refactor Production Debug Team Tools Window Help

Search (Ctrl+F)

default

PC: 0x0 nov z dc c : W:0x0 : bank 0 How do I? Keyword(s)

Projects Services

Lab_Diri_4

- Header Files
- Important Files
- Linker Files
- Source Files
 - Configuration.c
 - LCD.c
 - newmain.c
- Libraries
- Loadables

Lab_Diri_4 - Dashboard uno() - Navigator

- _XTAL_FREQ
- cuatro()
- dos()
- main()
- tres()
- uno()

Source History

```
58 LCD_gotoXY(1,11);
59 putraXLCD("m");
60 LCD_gotoXY(1,14);
61 putraXLCD("n");
62 LCD_gotoXY(1,15);
63 putraXLCD("o");
64 return;
65
66 void tres(void)
67 {
68     OpenXLCD(FOUR_BIT & LINES_5X6);
69     while(BusyXLCD());
70     WriteCmdXLCD(0x06);
71     WriteCmdXLCD(0x0C);
72     IN: LCD_gotoXY(0,2);
73     putraXLCD("Universidad");
74     LCD_gotoXY(1,1);
75     putraXLCD("Tecn. del Peru ");
76     delay_ms(1000);
77
78     LCD_gotoXY(0,4);
79     putraXLCD("Curso de");
80     LCD_gotoXY(1,1);
81     putraXLCD("uControladores");
82     delay_ms(1000);
83 }
```

Output - Lab_Diri_4 (Build, Load) Configuration Bits

Memory Summary:

Program space	used	4234h (16948) of 8000h bytes (51.7%)
Data space	used	137h (311) of 800h bytes (16.2%)

15:40 ESP 00:53 23/10/2018



```
83 goto IN;
84 return;
85
86 void cuatro(void)
87 {
88     int dato[17];
89     int dato1=246;
90     float dato2=35.42;
91
92     OpenXLCD(FOUR_BIT & LINES_5X8);
93     while(BusyXLCD());
94     WriteCmdXLCD(0x06);
95     WriteCmdXLCD(0x0C);
96
97     sprintf(dato,"Valor 1 = %d",dato1);
98     LCD_gotoXY(0,0);
99     putraXLCD(dato);
100     sprintf(dato,"Valor 2 = %.2f",dato2);
101     LCD_gotoXY(1,0);
102     putraXLCD(dato);
103     return;
104 }
105
106 void main(void)
107 {
108     uno();
109 }
```

Output - Lab_Diri_4 (Build, Load) x Configuration Bits

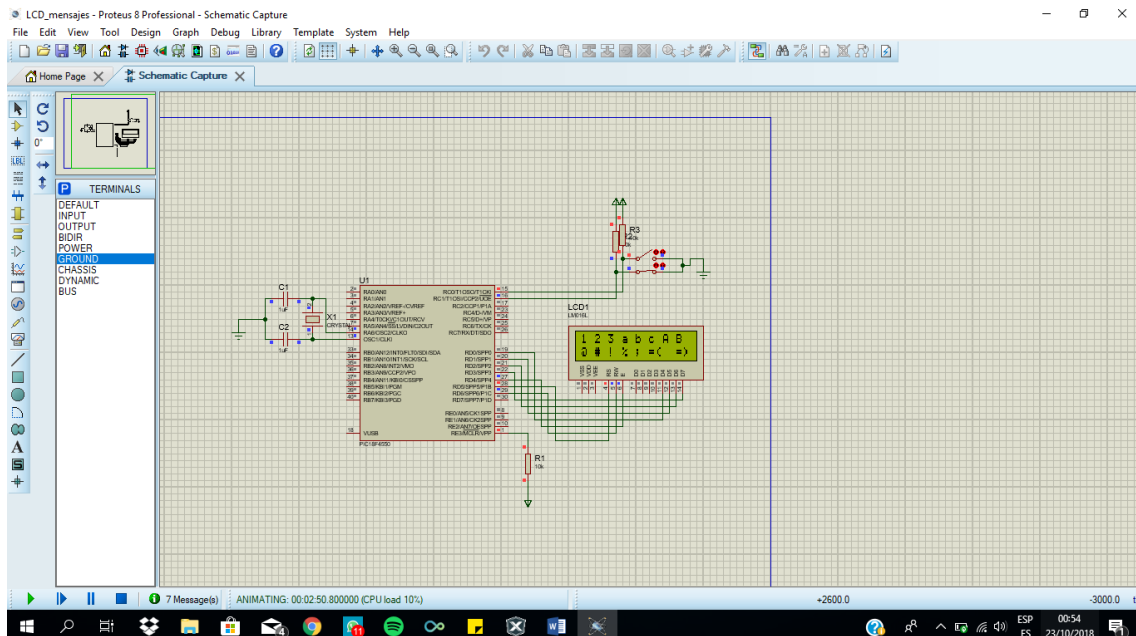
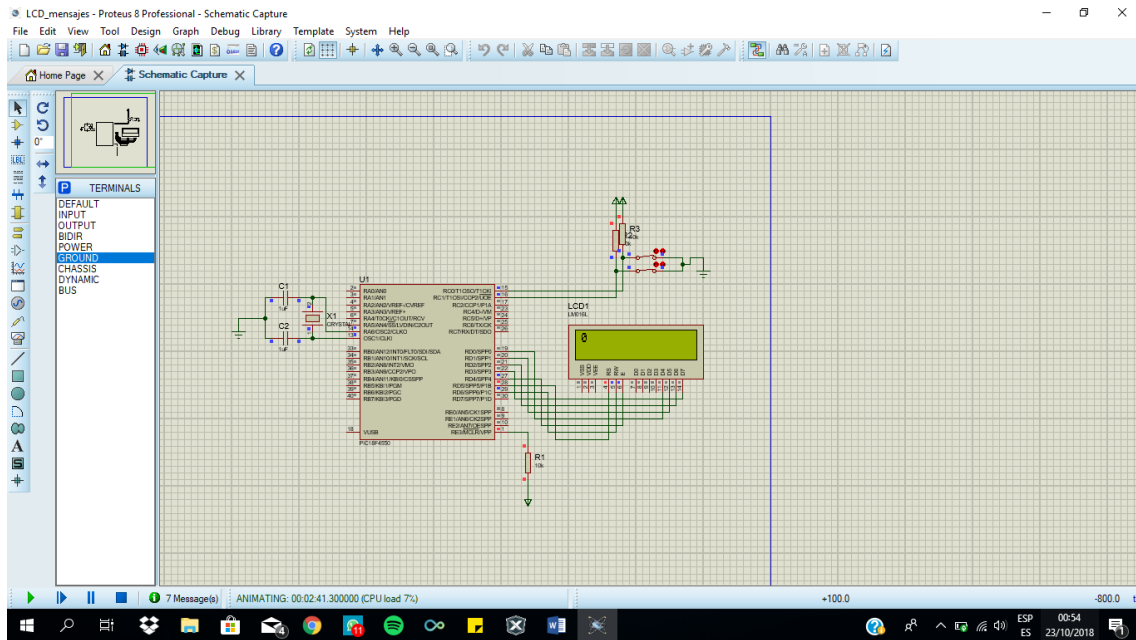
Memory Summary:		
Program space	used 4234h (16948) of 8000h bytes	(51.7%)
Data space	used 137h (311) of 800h bytes	(16.2%)

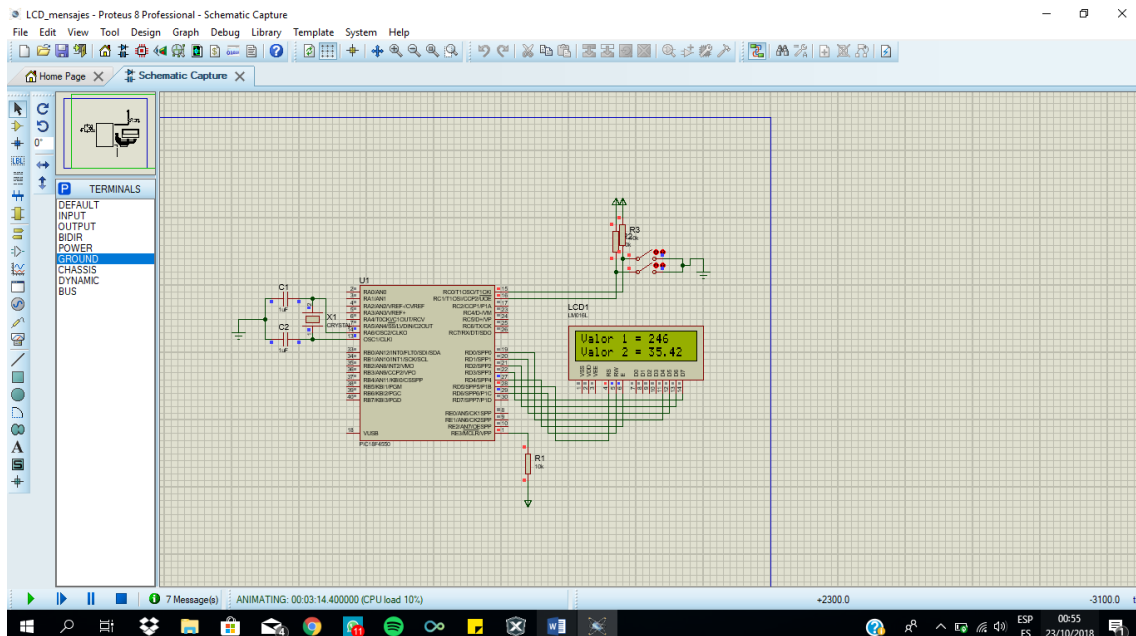
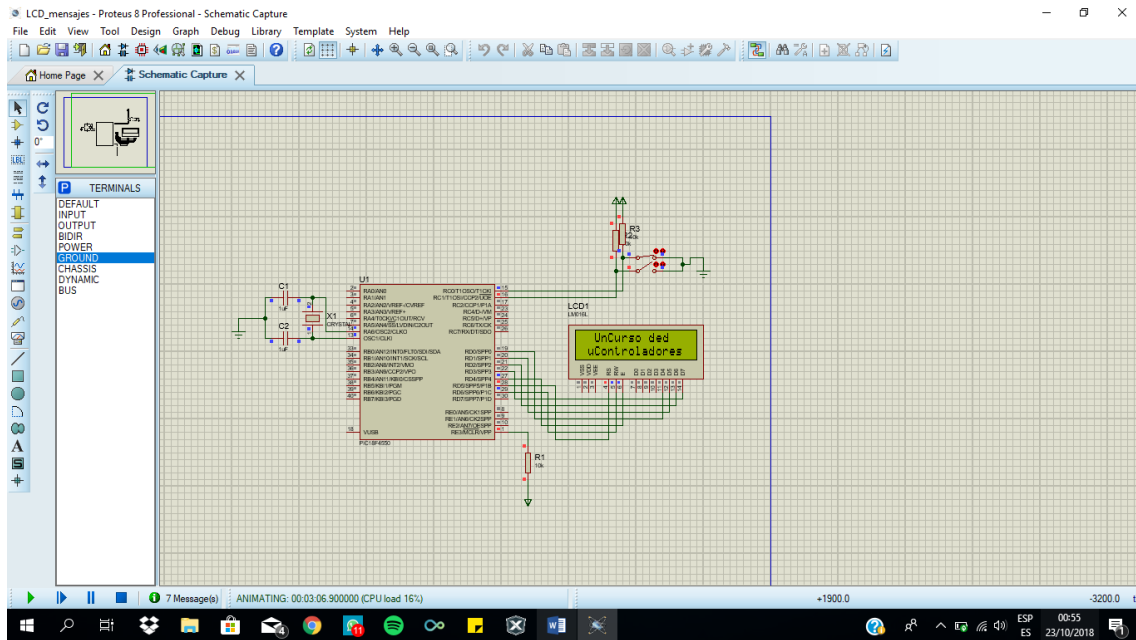
```
104 }
105
106 void main(void)
107 {
108     PORTC=3;
109     LEE: if(PORTC==0) goto UNO;
110         if(PORTC==1) goto DOS;
111         if(PORTC==2) goto TRES;
112         if(PORTC==3) goto CUATRO;
113     goto LEE;
114 UNO: uno();
115 LEE2:if(PORTC==0) goto LEE2;
116     goto LEE;
117 DOS: dos();
118 LEE3:if(PORTC==1) goto LEE3;
119     goto LEE;
120 TRES: tres();
121 LEE4:if(PORTC==2) goto LEE4;
122     goto LEE;
123 CUATRO: cuatro();
124 LEE5:if(PORTC==3) goto LEE5;
125     goto LEE;
126 }
127
128
```

Output - Lab_Diri_4 (Build, Load) x Configuration Bits

Memory Summary:		
Program space	used 4234h (16948) of 8000h bytes	(51.7%)
Data space	used 137h (311) of 800h bytes	(16.2%)

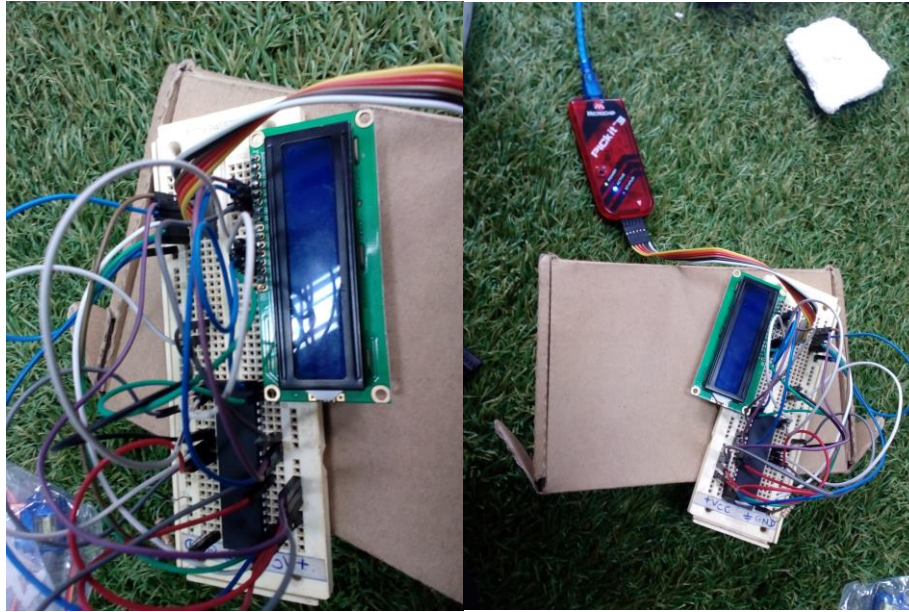
Del mismo modo, a continuación se muestra la simulación del circuito construido en Proteus 8:





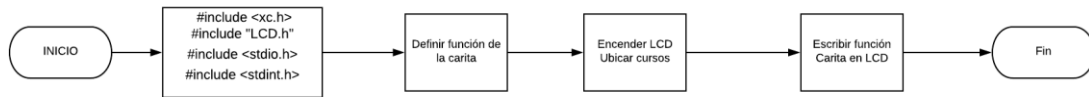
2.1.4. Circuito Real

Para la primera simulación se realizó un circuito en Protoboard conectado a base de la simulación del Proteus y el manual del PIC18F4550. Asimismo, se le colocó un Led, para ver que el circuito funcione correctamente, como se muestra a continuación:



2.2. Indicador de frases por combinación

2.2.1. Diagrama de flujo



2.2.2. Código

```
#include <xc.h>
```

```
#include "Configuracion.h"
```

```
#include "LCD.h"
```

```
#include <stdio.h>
```

```
#include <stdint.h>
```

```
#define _XTAL_FREQ 4000000
```

```
const char carita[]={0b00000000,  
                     0b00000000,  
                     0b00001010,  
                     0b00000000,  
                     0b00010001,  
                     0b00001110,  
                     0b00000000,  
                     0b00000000,0  
};
```

```
void main(void)
```

```
{
```

```
    OpenXLCD(FOUR_BIT & LINES_5X8);
```

```
    while(BusyXLCD());
```

```
    WriteCmdXLCD(0x06);
```

```
    WriteCmdXLCD(0x0C);
```

```
    LCD_WriteChr_CGRAM(carita,0);
```



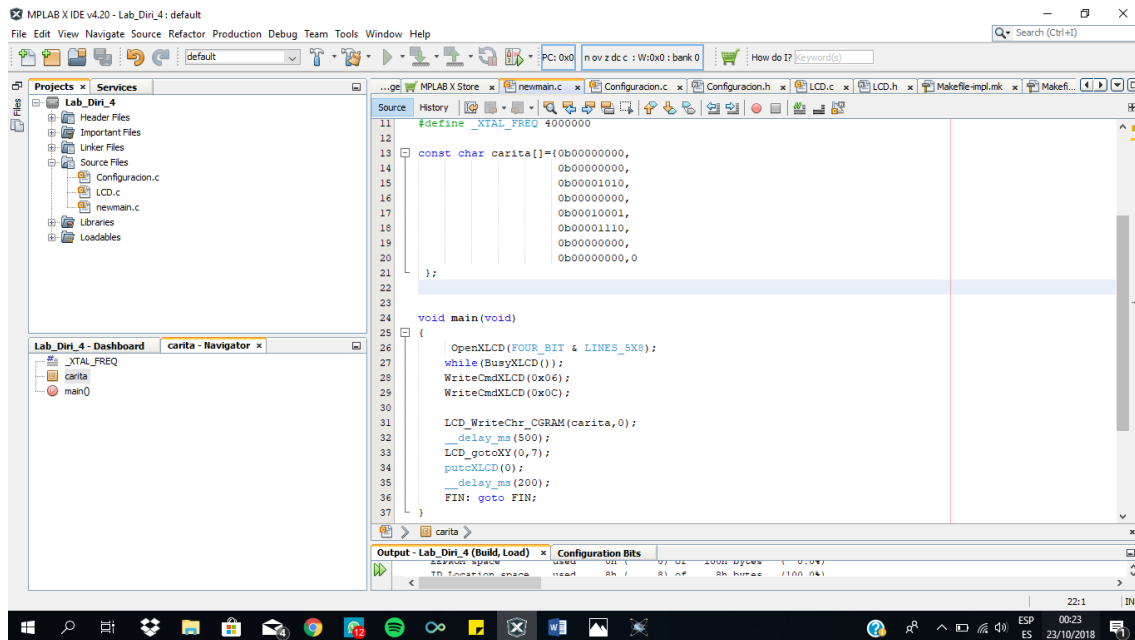
```

__delay_ms(500);
LCD_gotoXY(0,7);
putcXLCD(0);
__delay_ms(200);
FIN: goto FIN;
}

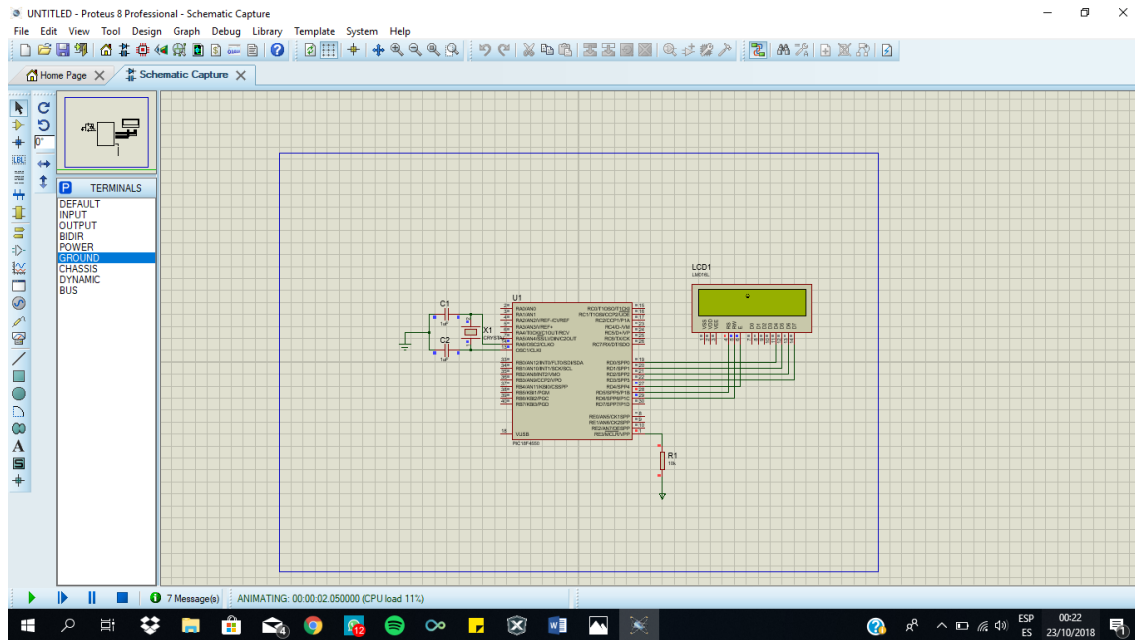
```

2.2.3. Lenguaje y Simulación

A continuación se muestra el código escrito en MPLAB V4.20:

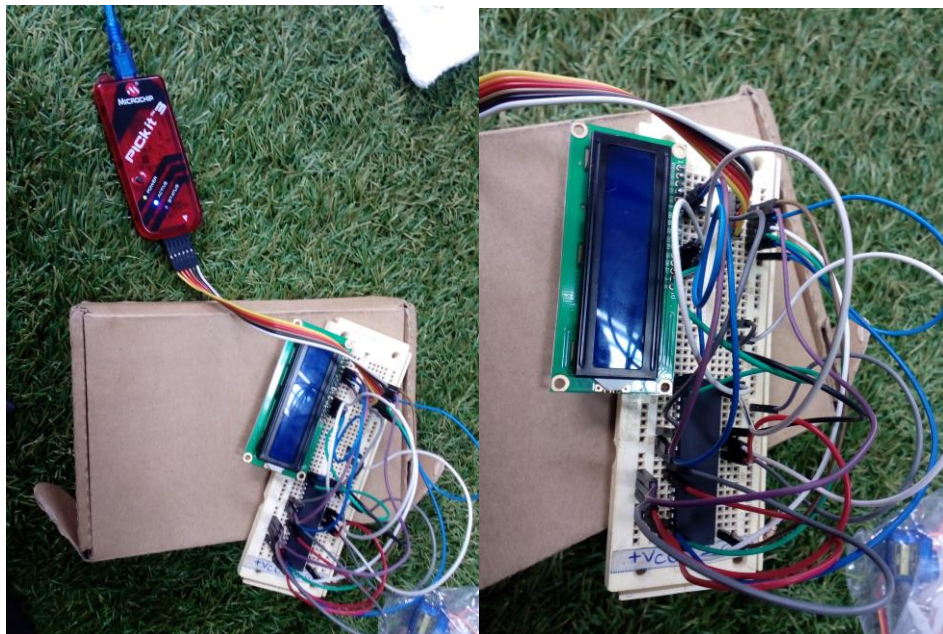


Del mismo modo, a continuación se muestra la simulación del circuito construido en Proteus 8:



2.2.4. Circuito Real

Para la segunda simulación se realizó un circuito en Protoboard conectado a base de la simulación del Proteus y el manual del PIC18F4550. Asimismo, se le colocó un Led, para ver que el circuito funcione correctamente, como se muestra a continuación:





Link del Proyecto

Leer observaciones



Observaciones

- Se tuvo que crear una función principal que llamará a cada mensaje en el LCD y se mantuviera hasta que lea otro valor del switch, porque si se ponía un retraso, el mensaje se escribiría nuevamente en un bucle y esto no es estéticamente favorable.
- Con el uso de la librería LCD fue más corto y sencillo escribir el código para escribir mensajes en el LCD con el PIC18F4550.
- El Pic Kit 3 del laboratorio estaba defectuoso y no se pudo realizar el video del proyecto, pues sin importar con cuantos pic y armados de circuitos en protoboards probará, este en ningún momento reconoció el PIC18F4550.

Conclusiones

- La configuración de los bits del PIC18F4550, en MPLABX, es esencial en la simulación ya que de hacerlo de manera correcta nos la facilitara, pero haciéndolo mal generará un error.
- Es fundamental la inclusión de la librería del LCD.
- Se puede simplificar la función principal al utilizar funciones y códigos establecidos por la librería
- Es fundamental escribir cada detalle del lenguaje C del programa de manera precisa para evitar errores.
- La librería hizo a la función de escritura más corta y sencilla.
- Existen dos maneras de escribir caracteres en el LCD, una para valores escritos en la función principal y la segunda para funciones pre definidas.
- El código Sprint permite escribir valores en el LCD de números como carácter.



Bibliografía

- MICROCHIP (2018) (<https://www.microchip.com/>) Sitio web oficial de Microchip; contiene información acerca del compilador XC8 y de MPLABX.
- ELECTRONICA ESTUDIO (2018)
(<http://www.electronicaestudio.com/index.htm>) Sitio web oficial de Electrónica estudio; contiene información acerca de los microcontroladores.
- WIKIPEDIA (2018)(<https://es.wikipedia.org/wiki/Wikipedia:Portada>) Sitio web oficial de Wikipedia; contiene información de interés.