



Universidad Tecnológica del Perú

Facultad de Ingeniería Electrónica y Mecatrónica

Curso : Microcontroladores

Profesor : Domínguez Jesús

Práctica Dirigida Nro : 1

Tema : Práctica Dirigida 2

Integrantes : Espinoza Valera, Jesús Alberto
Francisco
Alberto Francisco, Jaime Félix
Guerrero Isuiza, Mateo

Fecha del Experimento : Lunes, 7 y 10 de Septiembre del 2018

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

Fecha de entrega del informe : Martes, 11 de Septiembre 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.....	5
1.5. Microcontrolador.....	5
1.6. Display 7 segmentos.....	6
Capítulo 2: Resultados Obtenidos	7
2.1. Contador Módulo 10.....	7
2.1.1. Diagrama de flujo.....	7
2.1.2. Código.....	7
2.1.3. Lenguaje y Simulación.....	8
2.1.4. Circuito Real.....	9
2.2. Contador Definido por Switchs.....	9
2.2.1. Diagrama de flujo.....	9
2.2.2. Código.....	10
2.2.3. Lenguaje y simulación.....	10
2.2.4. Circuito Real.....	11
2.3. Contador Especial controlado por Switchs.....	12
2.3.1. Diagrama de flujo.....	12
2.3.2. Código.....	12
2.3.3. Lenguaje y simulación.....	14
2.3.4. Circuito Real.....	15
Link del video	16
Observaciones	17
Conclusiones	18
Bibliografía	19

Introducción

El interés del grupo de investigación se basó a partir de la observación de distintos programas que se realizaron en las clases de Microcontroladores. En consecuencia, nos planteamos la interrogante: ¿De qué manera se puede usar el PIC18F54550 como un circuito contador que siga determinadas instrucciones a partir de un switch conectado al mismo? Consideramos que es necesario hacer uso de vectores, punteros y funciones para crear un contador con instrucciones en el PIC 18F4550.

En primer lugar, se presenta el marco teórico donde se define el tipo de lenguaje a usar, una breve descripción del circuito usado en los PIC y un concepto básico de los programas usados para lograr este informe. En segundo lugar, se desarrolla la experimentación con distintos programas en el PIC18F4550, con la finalidad de obtener los contadores deseados. En tercer lugar, se mencionan las recomendaciones ofrecidas por el grupo que desarrolló la investigación, para mejorar el proceso del informe. Finalmente, en la conclusión se afirma que fue de gran importancia el uso de vectores, punteros y funciones en la creación del lenguaje de programación que se escribió en el PIC para cumplir con los objetivos de cada actividad del informe.

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.

Ahora, una vez definido cada instrumento usado en este proyecto, nosotros hicimos parte de armar todo lo mencionado. Aquí veremos cómo es que mediante el uso del lenguaje C se hizo la programación, y la compilación para generar el código, seria parte del primer proceso para implementar a lo que es

físico, tomando en cuenta que mientras se hacía el primer procedimiento, se tenía que estar implementando el circuito en el protoboard, de manera tal que cumpla con lo que se requiere como salida. Entonces en el proceso de uso de ciertas herramientas usamos el PicKit como un mediador ante lo que ya era simulado y lo que era real para plasmar todo lo que se requería

A continuación se verá cada una de las partes que se ha dado a expresar a manera de introducción, explicando paso a paso según a de desarrollarse.

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.

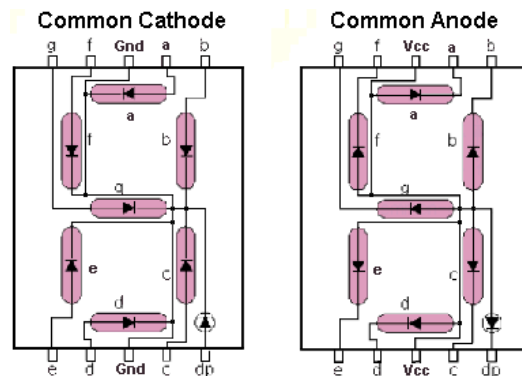
Inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, la cual se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de manera indistinta.

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

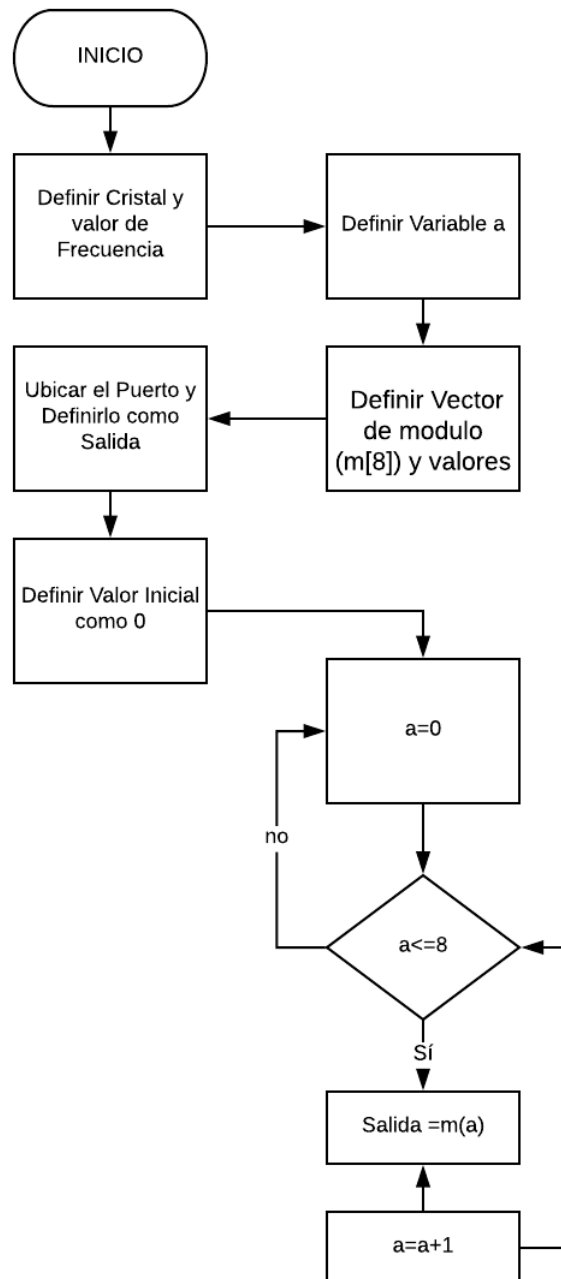


partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.

Capítulo 2: Resultados Obtenidos

2.1. Contador Módulo 10

2.1.1. Diagrama de flujo



2.1.2. Código

```
#define __XTAL_FREQ 4000000 //Definir el cristal y su frecuencia
```

```
int main (int argc, char** argv)
```

```
{
```

```

int a; // Definir a como entero

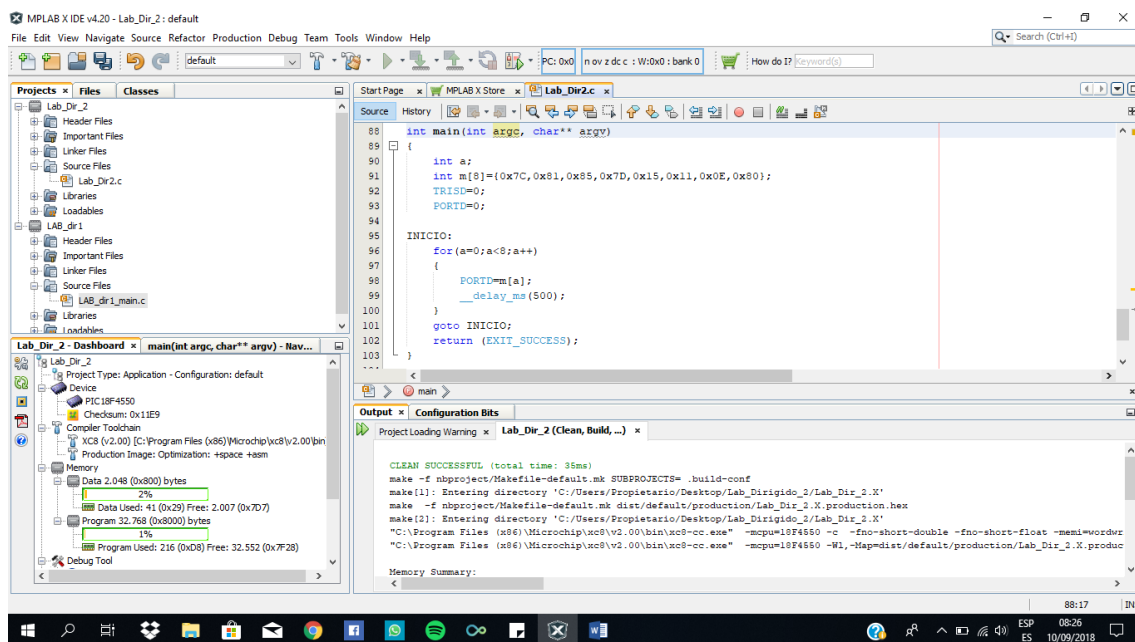
int m[8]={0x7C,0x81,0x85,0x7D,0x15,0x11,0x0E,0x80}; // Definir función
TRISD=0; // Definir puerto D como salida
PORTD=0; // Definir la salida como 0
INICIO: // Definir etiqueta INICIO
for (a=0;a<8;a++) // Establecer for como contador
{
    PORTD= m[a]; // Establecer el valor de salida como a
    __delay_ms(500); // Definir retraso 500 microsegundos
}

goto INICIO;
return (EXIT_SUCESS);
}

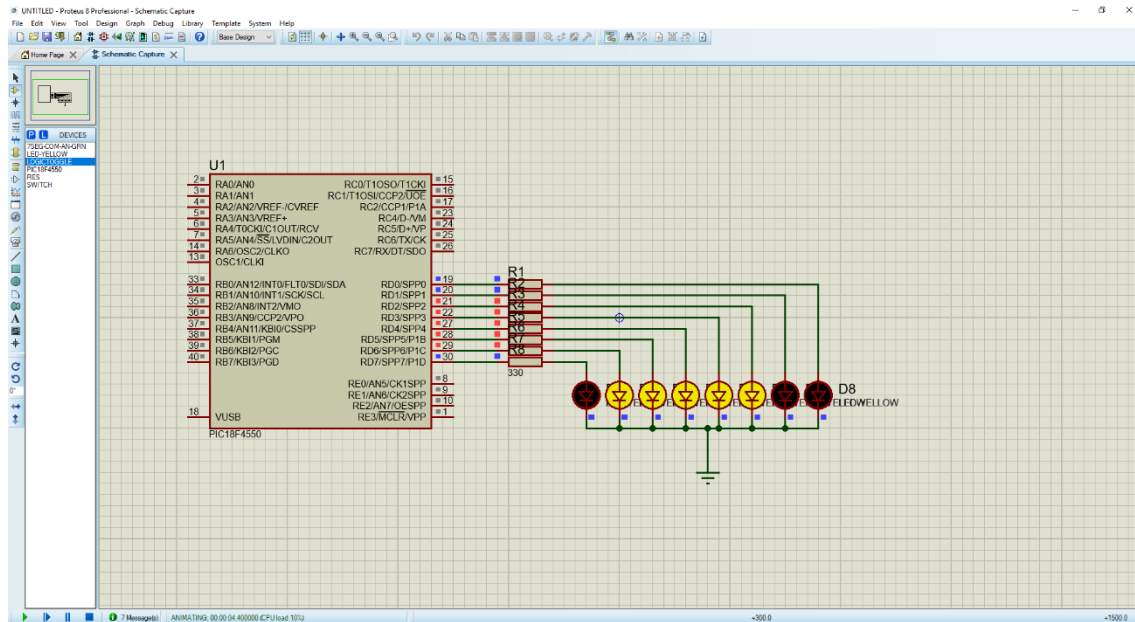
```

2.1.3. Lenguaje y Simulación

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

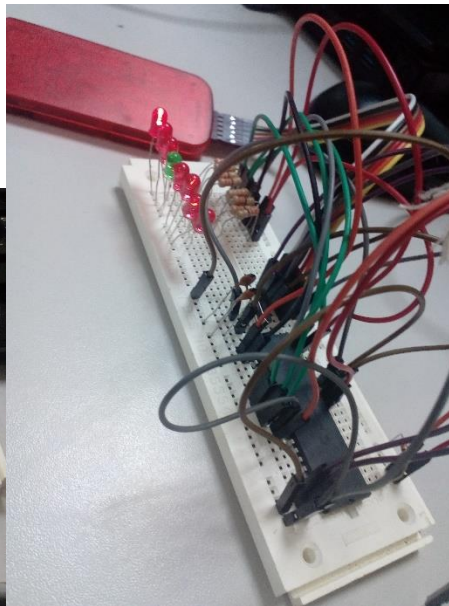
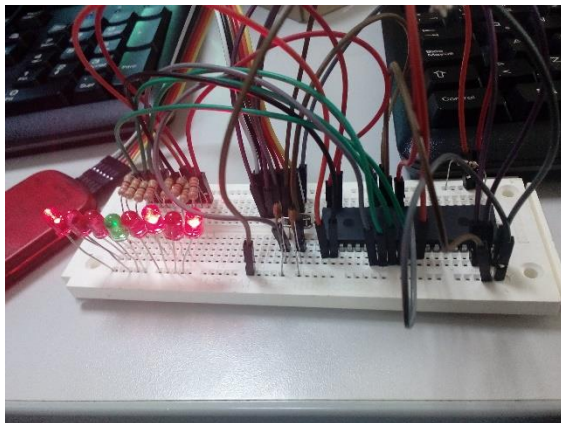


Del mismo modo, a continuación se muestra la simulación del circuito construido en Proteus 8:



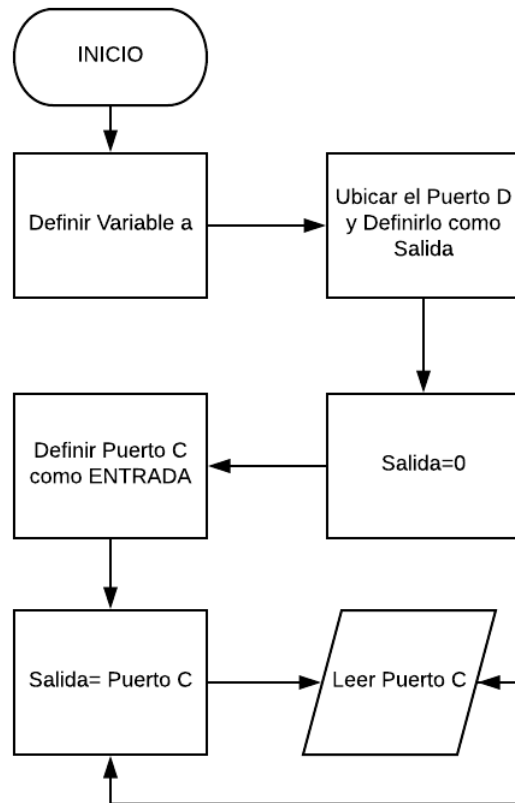
2.1.4. Circuito Real

Para la primera simulación se realizó un circuito en protoboard conectado a base de la simulación del Proteus y el manual del PIC18F4550:



2.2. Contador Definido por Switchs

2.2.1. Diagrama de flujo



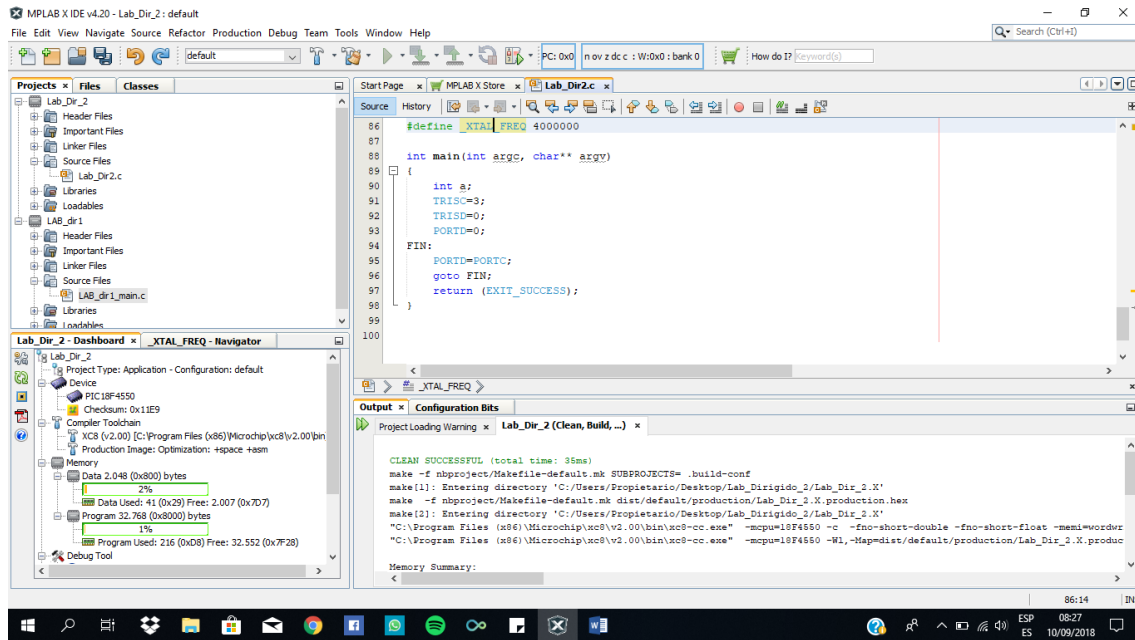
2.2.2. Código

```

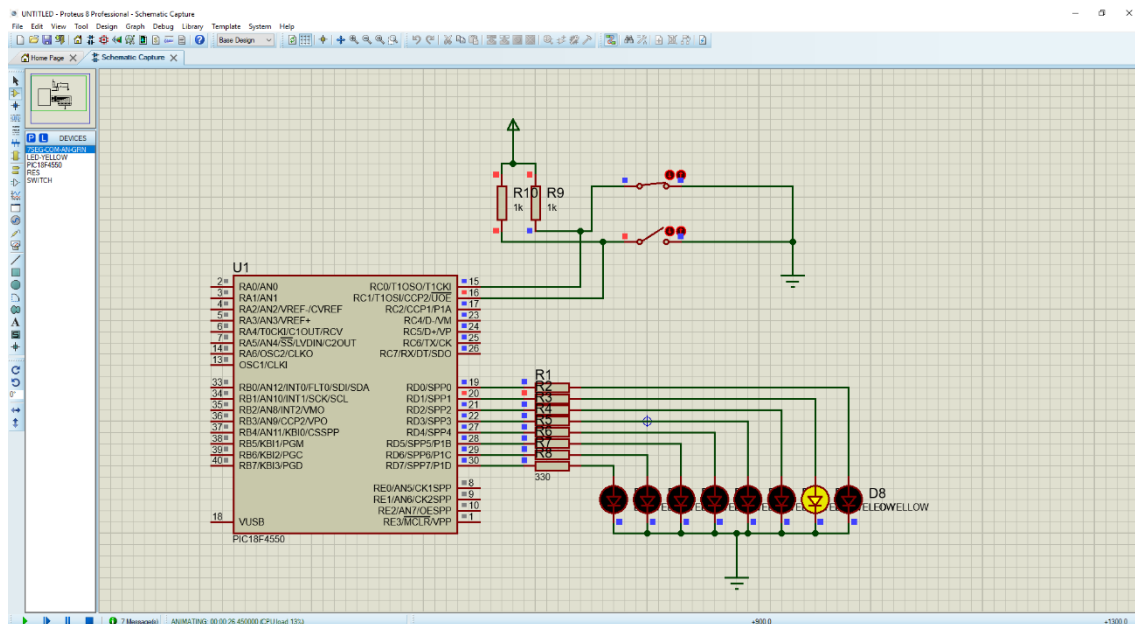
int main (int argc, char** argv)
{
    int a;                                // Definir a como entero
    TRISD=0;                              // Definir puerto D como salida
    TRISC=3;                              // Definir Puerto C salida y rango 3
    PORTD=0;                              // Definir la salida como 0
    FIN:                                  // Definir etiqueta INICIO
    PORTD=PORTC                            //Salida D binario es entrada C decimal
    goto FIN;                              // Volver a la etiqueta FIN
    return (EXIT_SUCESS);
}
  
```

2.2.3. Lenguaje y simulación

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

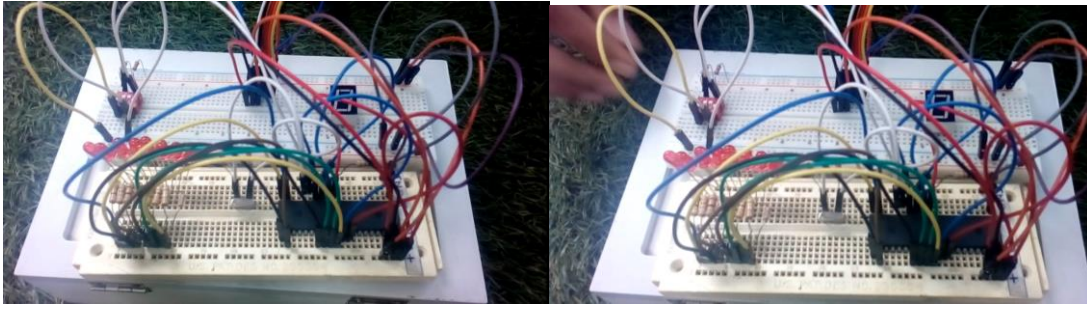


Del mismo modo, a continuación se muestra la simulación del circuito construido en Proteus 8:



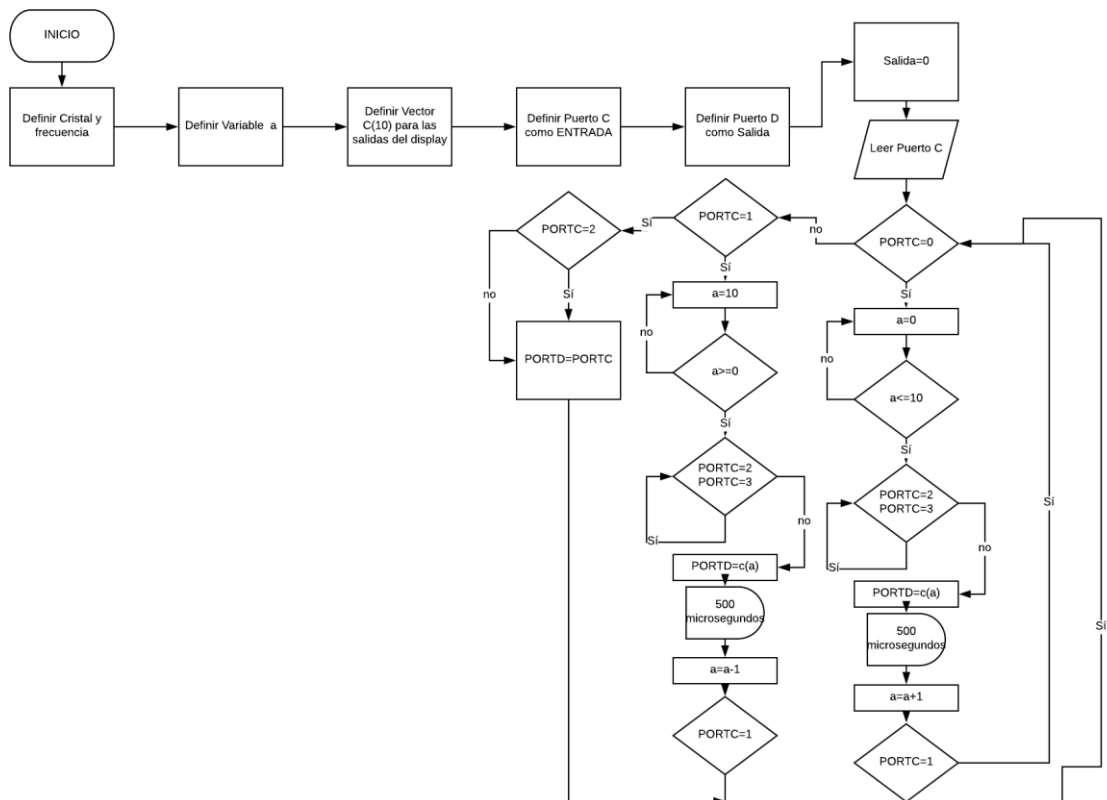
2.2.4. Circuito Real

A continuación se muestra el mismo circuito que el que se realizó en la simulación anterior, con la diferencia de que este ya se encontraba reparado y posee un cristal de 4Mhz en total funcionamiento:



2.3. Contador Especial controlado por Switchs

2.3.1. Diagrama de flujo



2.3.2. Código

```
#define __XTAL_FREQ 4000000 //Definir el cristal y su frecuencia
```

```
int main (int argc, char** argv)
```

```
{
```

```
    int a; // Definir a como entero
```

```
    int c[10]={0x40,0xF9,0x24,0x30,0x19,0x12,0x02,0xF8,0x00,0x10};
```



```
    TRISC=3;           // Definir puerto C como entrada de rango 3
    TRISD=0;           // Definir puerto D como entrada
    PORTD=0x40;        // Definir la salida como 40 decimal
INICIO:               // Definir etiqueta INICIO
    if (PORTC==0) goto UP; // Establecer primera condicional
    if (PORTC==1) goto DN; // Establecer segunda condicional

    UP:               // Definir etiqueta UP
    for (a=0;a<10;a++) // Establecer contador ascendente
    {
LEE1:               // Definir etiqueta LEE1
    if (PORTC==2 || PORTC==3) goto LEE1; // Establecer condicional
    PORTD= c[a];      // Establecer el valor de salida como función a
    __delay_ms(500);  // Definir retraso 500 microsegundos
    if (PORTC==1) goto DN2; // Establecer segunda condicional
UP2:;              // Definir etiqueta UP2
    }
    goto INICIO;      // Volver a la etiqueta INICIO

    DN:               // Definir etiqueta DN
    for (a=9;a>=9;a--) // Establecer contador descendente
    {
LEE2:               // Definir etiqueta LEE2
    if (PORTC==2 || PORTC==3) goto LEE2; // Establecer condicional
    PORTD= c[a];      // Establecer el valor de salida como función a
    __delay_ms(500);  // Definir retraso 500 microsegundos
    if (PORTC==0) goto UP2; // Establecer segunda condicional
DN2:;              // Definir etiqueta DN2
    }
    goto INICIO;      // Volver a la etiqueta INICIO

    return (EXIT_SUCESS);
}
```



2.3.3. Lenguaje y simulación

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

MPLAB X IDE v4.20 - Lab_Dir_2: default

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

Search (Ctrl+F)

nov zdc c : W:\00 : bank 0

How do I? (keyword)

Projects Files Classes

Lab_Dir_2

Header Files

Important Files

Linker Files

Source Files

Lab_Dir_2.c

Libraries

Loadables

LAB_dir1

Header Files

Important Files

Linker Files

Source Files

LAB_dir1_main.c

Libraries

Loadables

Lab_Dir_2: Dashboard

main(int argc, char** argv) - Nav...

Lab_Dir_2

Project Type: Application - Configuration: default

Device: PIC18F4550

Checksum: 0x11E9

Compiler Toolchain: XC8 (v2.00) [C:\Program Files (x86)\Microchip\xc8\v2.00\bin] Production Image: Optimization: -space -asm

Memory

Data 2,048 (0x800) bytes

2%

Data Used: 41 (0x29) Free: 2,007 (0x7D7)

Program 32,768 (0x8000) bytes

1%

Program Used: 216 (0x08) Free: 32,552 (0x7F28)

Debug Tool

Source

History

StartPage MPLAB X Store Lab_Dir_2.c

```
86 #define XTAL_FREQ 4000000
87
88 int main(int argc, char** argv)
89 {
90     int a;
91     int c[10]={0x40,0xF8,0x24,0x30,0x19,0x12,0x02,0xF8,0x00,0x10};
92     TRIS0=3;
93     TRIS0=0;
94     PORT0=0x40;
95
96     INICIO:
97     if(PORT0==0) goto UP;
98     if(PORT0==1) goto DN;
99
100     UP:
101     for(a=0;a<10;a++)
```

Output Configuration Bits

Project Loading Warning Lab_Dir_2 (Clean, Build, ...)

CLEAN SUCCESSFUL (total time: 35ms)

make -f nbproject\Makefile-default.mk SUBPROJECTS= build-conf

make[1]: Entering directory 'C:/Users/Propietario/Desktop/Lab_Dirigo_2/Lab_Dir_2.X'

make -f nbproject\Makefile-default.mk dist/default/production/Lab_Dir_2.X.production.hex

make[2]: Entering directory 'C:/Users/Propietario/Desktop/Lab_Dirigo_2/Lab_Dir_2.X'

"C:\Program Files (x86)\Microchip\xc8\v2.00\bin\cc8-cc.exe" -mcpu=18F4550 -c -fno-short-double -fno-short-float -mml=wordwr

"C:\Program Files (x86)\Microchip\xc8\v2.00\bin\cc8-cc.exe" -mcpu=18F4550 -Wl,-Map=dist/default/production/Lab_Dir_2.X.produc

Memory Summary:

MPLAB X IDE v4.20 - Lab_Dir_2: default

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

Search (Ctrl+F)

nov zdc c : W:\00 : bank 0

How do I? (keyword)

Projects Files Classes

Lab_Dir_2

Header Files

Important Files

Linker Files

Source Files

Lab_Dir_2.c

Libraries

Loadables

LAB_dir1

Header Files

Important Files

Linker Files

Source Files

LAB_dir1_main.c

Libraries

Loadables

Lab_Dir_2: Dashboard

main(int argc, char** argv) - Nav...

Lab_Dir_2

Project Type: Application - Configuration: default

Device: PIC18F4550

Checksum: 0x11E9

Compiler Toolchain: XC8 (v2.00) [C:\Program Files (x86)\Microchip\xc8\v2.00\bin] Production Image: Optimization: -space -asm

Memory

Data 2,048 (0x800) bytes

2%

Data Used: 41 (0x29) Free: 2,007 (0x7D7)

Program 32,768 (0x8000) bytes

1%

Program Used: 216 (0x08) Free: 32,552 (0x7F28)

Debug Tool

Source

History

StartPage MPLAB X Store Lab_Dir_2.c

```
100 UP:
101 for(a=0;a<10;a++)
102 {
103     if(PORT0==2 || PORT0==3) goto LEE1;
104     PORT0=c[a];
105     delay_ms(500);
106     if(PORT0==1) goto DN2;
107
108     UP2:
109     goto INICIO;
110
111     DN:
112     for(a=9;a>0;a--)
113     {
114         LEE2: if(PORT0==2 || PORT0==3) goto LEE2;
115         PORT0=c[a];
```

Output Configuration Bits

Project Loading Warning Lab_Dir_2 (Clean, Build, ...)

CLEAN SUCCESSFUL (total time: 35ms)

make -f nbproject\Makefile-default.mk SUBPROJECTS= build-conf

make[1]: Entering directory 'C:/Users/Propietario/Desktop/Lab_Dirigo_2/Lab_Dir_2.X'

make -f nbproject\Makefile-default.mk dist/default/production/Lab_Dir_2.X.production.hex

make[2]: Entering directory 'C:/Users/Propietario/Desktop/Lab_Dirigo_2/Lab_Dir_2.X'

"C:\Program Files (x86)\Microchip\xc8\v2.00\bin\cc8-cc.exe" -mcpu=18F4550 -c -fno-short-double -fno-short-float -mml=wordwr

"C:\Program Files (x86)\Microchip\xc8\v2.00\bin\cc8-cc.exe" -mcpu=18F4550 -Wl,-Map=dist/default/production/Lab_Dir_2.X.produc

Memory Summary:

MPLAB X IDE v4.20 - Lab_Dir_2: default

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

Search (Ctrl+F)

nov zdc c : W:\00 : bank 0

How do I? (keyword)

Projects Files Classes

Lab_Dir_2

Header Files

Important Files

Linker Files

Source Files

Lab_Dir_2.c

Libraries

Loadables

LAB_dir1

Header Files

Important Files

Linker Files

Source Files

LAB_dir1_main.c

Libraries

Loadables

Lab_Dir_2: Dashboard

main(int argc, char** argv) - Nav...

Lab_Dir_2

Project Type: Application - Configuration: default

Device: PIC18F4550

Checksum: 0x11E9

Compiler Toolchain: XC8 (v2.00) [C:\Program Files (x86)\Microchip\xc8\v2.00\bin] Production Image: Optimization: -space -asm

Memory

Data 2,048 (0x800) bytes

2%

Data Used: 41 (0x29) Free: 2,007 (0x7D7)

Program 32,768 (0x8000) bytes

1%

Program Used: 216 (0x08) Free: 32,552 (0x7F28)

Debug Tool

Source

History

StartPage MPLAB X Store Lab_Dir_2.c

```
111 DN:
112 for(a=9;a>0;a--)
113 {
114     LEE2: if(PORT0==2 || PORT0==3) goto LEE2;
115     PORT0=c[a];
116     delay_ms(500);
117     if(PORT0==0) goto UP2;
118
119     DN2:
120     goto INICIO;
121
122     return (EXIT_SUCCESS);
123 }
124
```

Output Configuration Bits

Project Loading Warning Lab_Dir_2 (Clean, Build, ...)

CLEAN SUCCESSFUL (total time: 35ms)

make -f nbproject\Makefile-default.mk SUBPROJECTS= build-conf

make[1]: Entering directory 'C:/Users/Propietario/Desktop/Lab_Dirigo_2/Lab_Dir_2.X'

make -f nbproject\Makefile-default.mk dist/default/production/Lab_Dir_2.X.production.hex

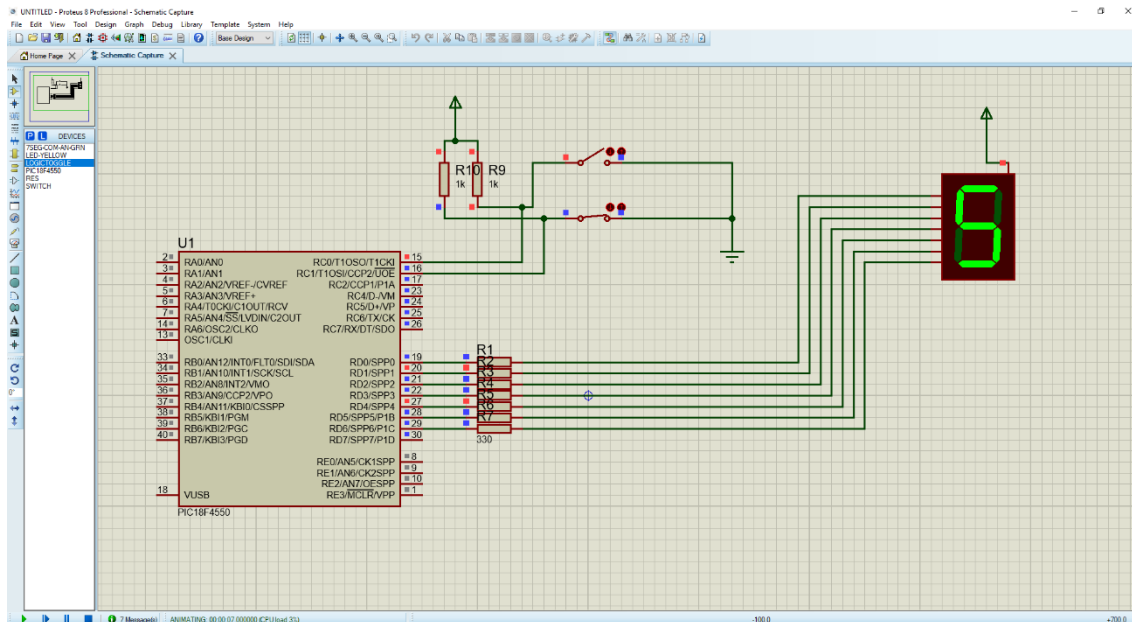
make[2]: Entering directory 'C:/Users/Propietario/Desktop/Lab_Dirigo_2/Lab_Dir_2.X'

"C:\Program Files (x86)\Microchip\xc8\v2.00\bin\cc8-cc.exe" -mcpu=18F4550 -c -fno-short-double -fno-short-float -mml=wordwr

"C:\Program Files (x86)\Microchip\xc8\v2.00\bin\cc8-cc.exe" -mcpu=18F4550 -Wl,-Map=dist/default/production/Lab_Dir_2.X.produc

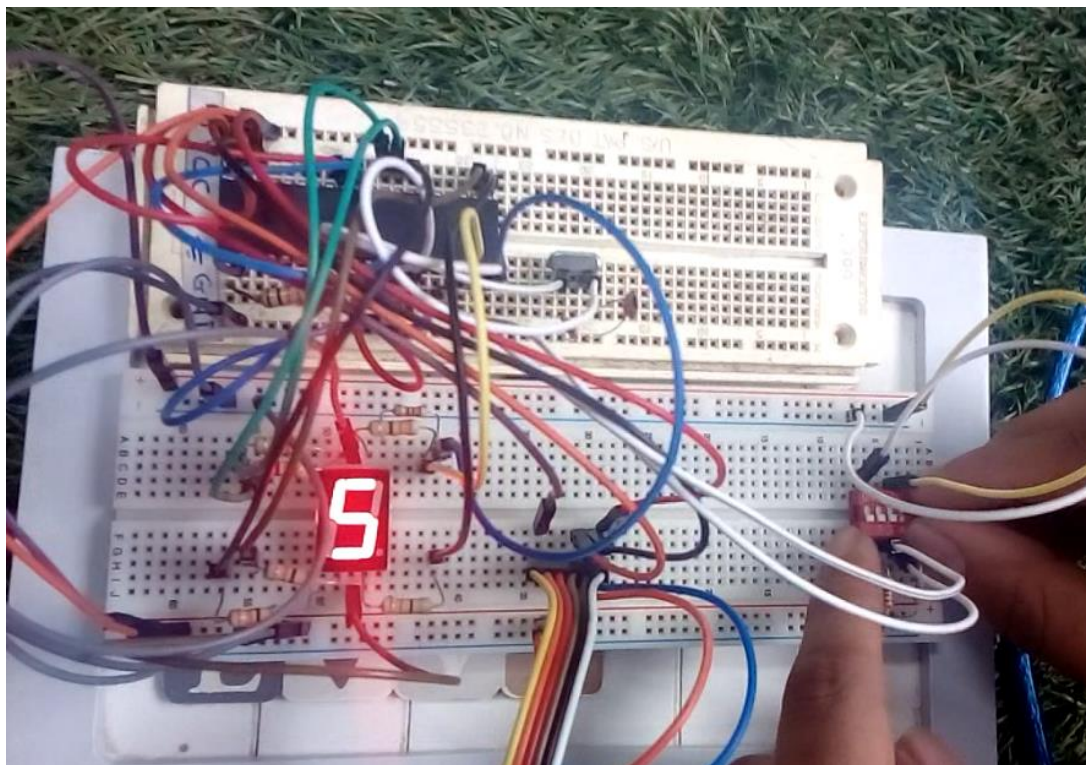
Memory Summary:

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



2.3.4. Circuito Real

A continuación se muestra el mismo circuito que el que se realizó en la simulación anterior:





Link del video

https://youtu.be/_mYbgGANH00



Observaciones

- Se configuró una función de salida especial para el display con la finalidad de que se mostrará el valor deseado.
- Para poder usar el retraso “delay” en microsegundos, fue necesario definir el cristal y su frecuencia

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.
- Los vectores, punteros y funciones fueron fundamental en la escritura del programa del PIC para lograr el funcionamiento deseado.
- Es necesario definir el cristal y su frecuencia para usar el retraso delay.
- Al usar un display, se debe crear una función de salida especial para que se muestren los valores deseados, ya que el display posee una combinación de bits que no necesariamente coincide con el valor que se desea mostrar.
- Es fundamental escribir cada detalle del lenguaje C del programa de manera precisa para evitar errores.



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.