



Universidad Tecnológica del Perú

Facultad de Ingeniería Electrónica y Mecatrónica

Curso : Microcontroladores

Profesor : Domínguez Jesús

Laboratorio Nro : 3

Tema : Laboratorio Dirigido 3

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

Fecha del Experimento : Lunes, 2 de Octubre del 2018

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

Fecha de entrega del informe : Lunes, 15 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. Oscilador	5
1.8. Generador de Señales	6
Capítulo 2: Resultados Obtenidos	7
2.1. Indicador de frases por combinación	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. Indicador de frases por combinación	11
2.2.1. Diagrama de flujo	11
2.2.2. Código	11
2.2.3. Lenguaje y Simulación	12
2.2.4. Circuito Real	13
Link del Proyecto	15
Observaciones	16
Conclusiones	17
Bibliografía	18

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 procesos paralelos que sirvan de interrupción en procesos infinitos del PIC18F4550. En consecuencia, nos planteamos la interrogante: ¿En qué medida es posible obtener un divisor de frecuencia a partir de unas señales de entrada mediante el PIC18F4550? Consideramos que es posible construir un mecanismo que permita al PIC18F4550 ser un circuito divisor de frecuencia.

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 alternativo 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. Oscilador

Un osciloscopio es un instrumento de visualización electrónico para la representación gráfica de señales eléctricas que pueden variar en el tiempo. Es muy usado en electrónica de señal, frecuentemente junto a un analizador de espectro.

Presenta los valores de las señales eléctricas en forma de coordenadas en una pantalla, en la que normalmente el eje X (horizontal) representa tiempos y el eje Y (vertical) representa tensiones. La imagen así obtenida se denomina oscilograma. Suelen incluir otra entrada, llamada "eje THRASHER" o "Cilindro de Wehnelt" que controla la luminosidad del haz, permitiendo resaltar o apagar algunos segmentos de la traza.

Los osciloscopios, clasificados según su funcionamiento interno, pueden ser tanto analógicos como digitales, siendo el resultado mostrado idéntico en cualquiera de los dos casos, en teoría.

1.8. Generador de Señales

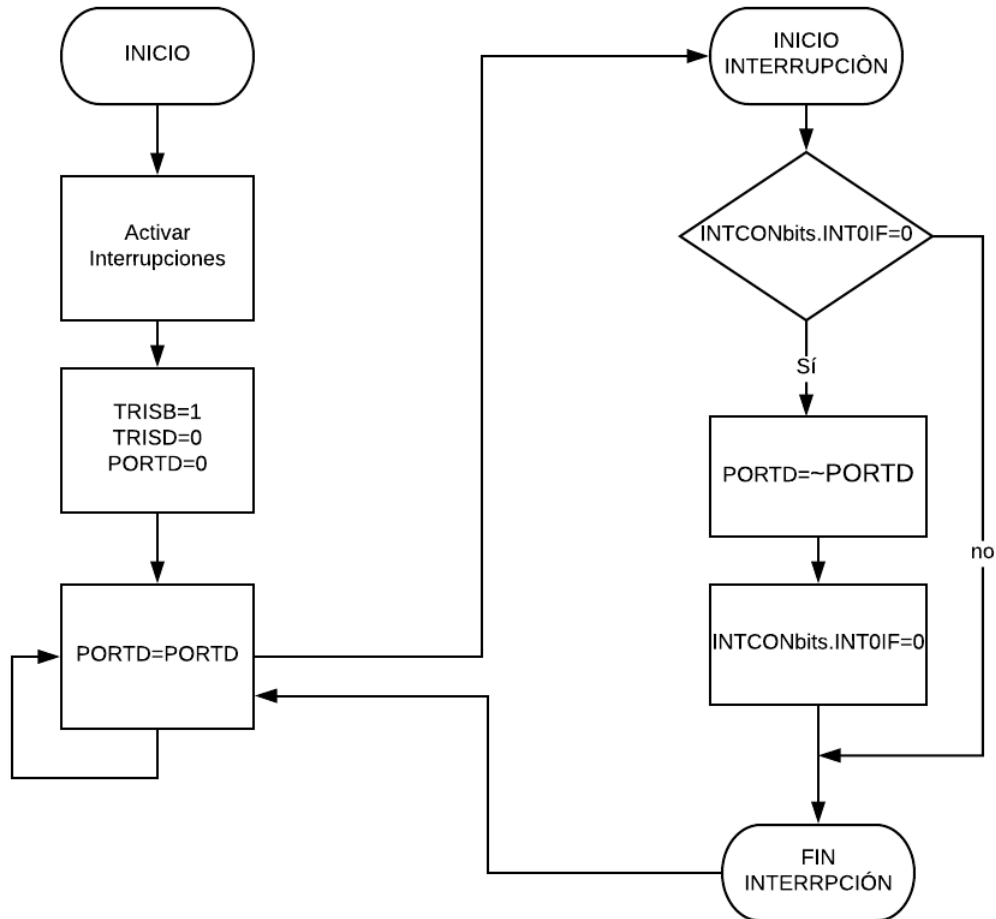
Un generador de señales, de funciones o de formas de onda es un dispositivo electrónico de laboratorio que genera patrones de señales periódicas o no periódicas tanto analógicas como digitales. Se emplea normalmente en el diseño, prueba y reparación de dispositivos electrónicos; aunque también puede tener usos artísticos.

Hay diferentes tipos de generadores de señales según el propósito y aplicación que corresponderá con el precio. Tradicionalmente los generadores de señales eran dispositivos estáticos apenas configurables, pero actualmente permiten la conexión y control desde un PC. Con lo que pueden ser controlados mediante software hecho a medida según la aplicación, aumentando la flexibilidad.

Capítulo 2: Resultados Obtenidos

2.1. Indicador de frases por combinación

2.1.1. Diagrama de flujo



2.1.2. Código

```
void __interrupt() tc_int (void);
```

```
void main (void)
```

```
{
```

```
    TRISD=0;
```

```
    TRISB=1;
```

```
    PORTD=0;
```

```
    INTCONbits.GIE=1;
```

```
    INTCONbits.PEIE=1;
```

```
    INTCONbits.INT0IE=1;
```

```

INTCONbits.INT0IF=0;
INTCON2bits.INTEDG0=1;
RCONbits.IPEN=0;
while(1)
{
    PORTD=PORTD;
}
return;
}

```

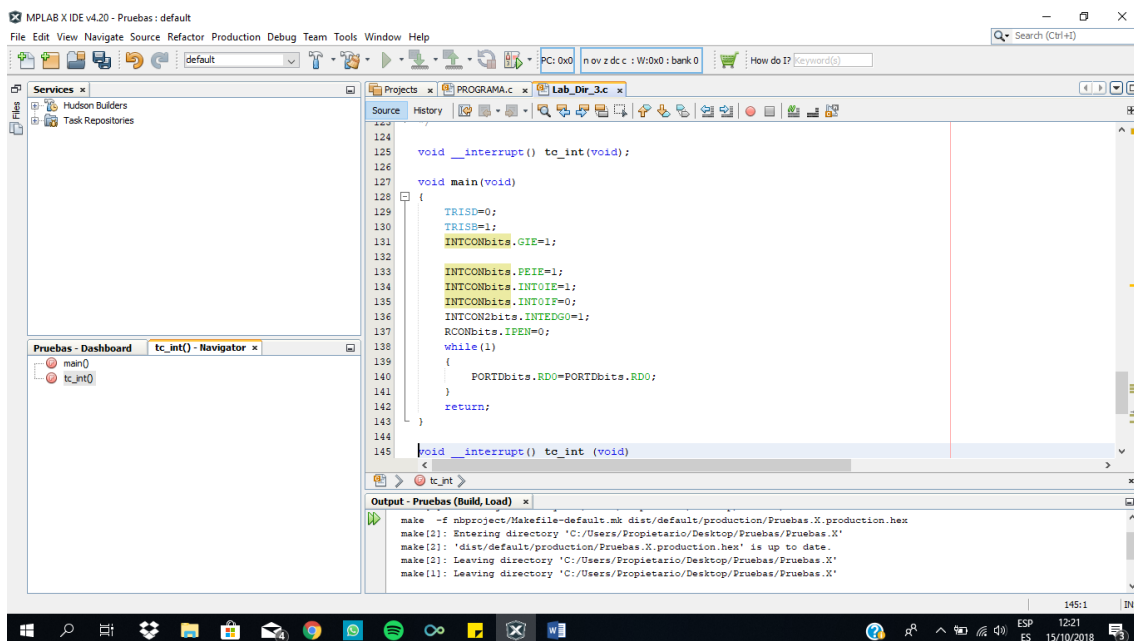
```

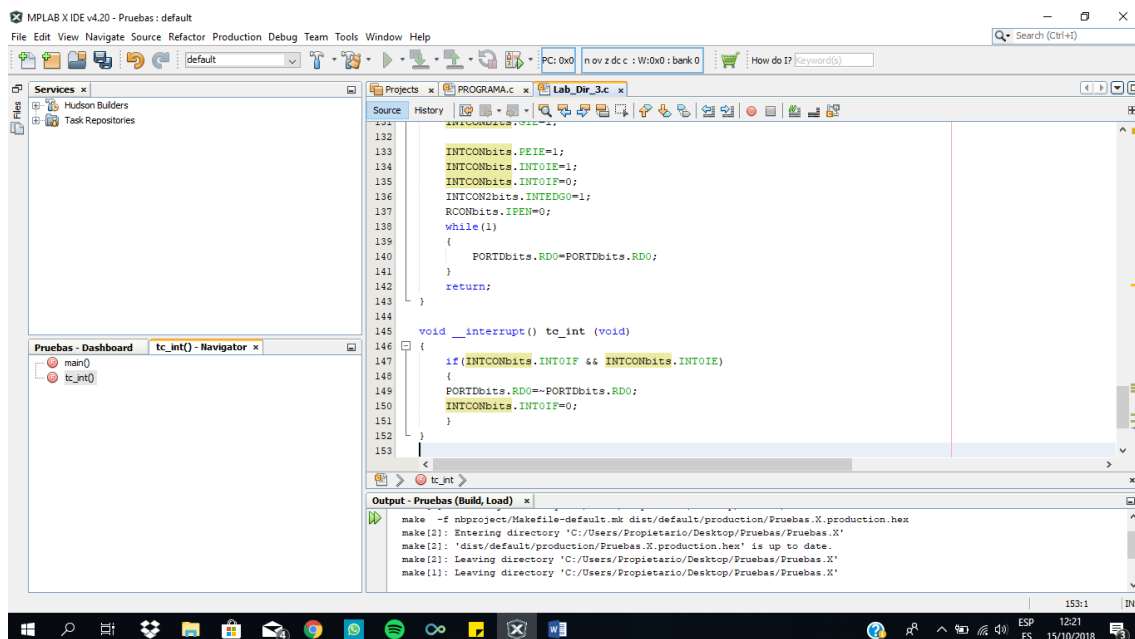
void __interrupt() tc_int (void)
{
    if (INTCONbits.INT0IF)
    {
        PORTD=~PORTD;
        INTCNbits.INT0IF=0;
    }
}

```

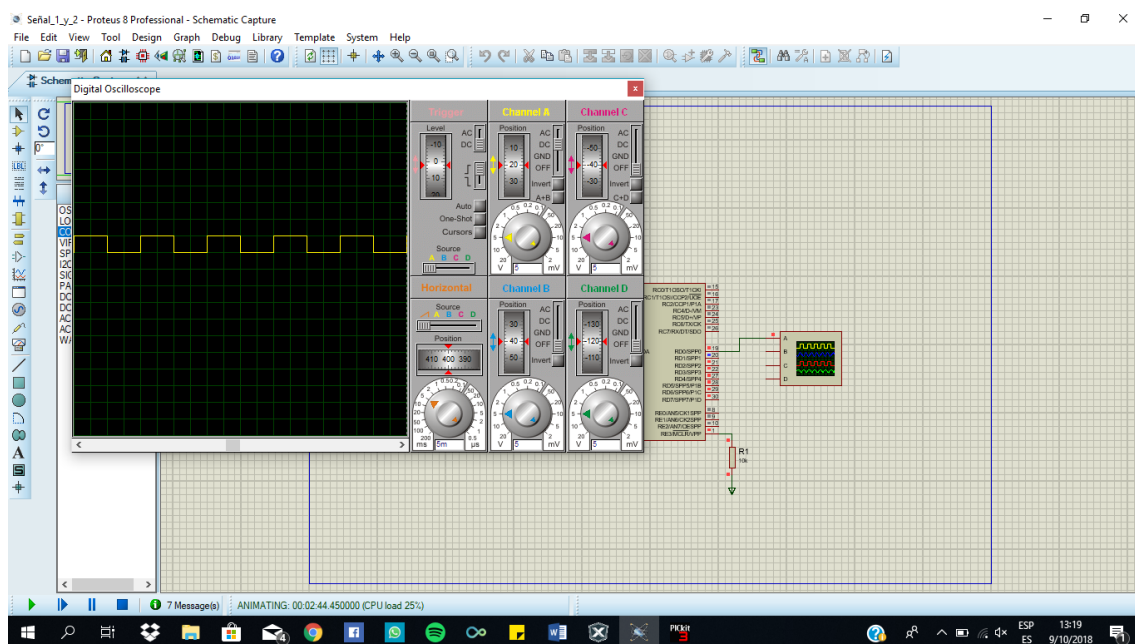
2.1.3. Lenguaje y Simulación

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





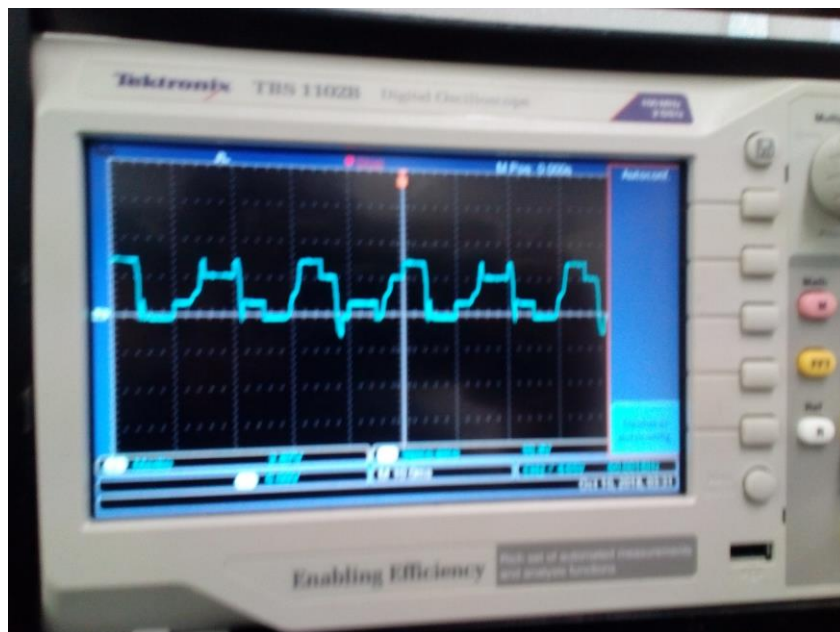
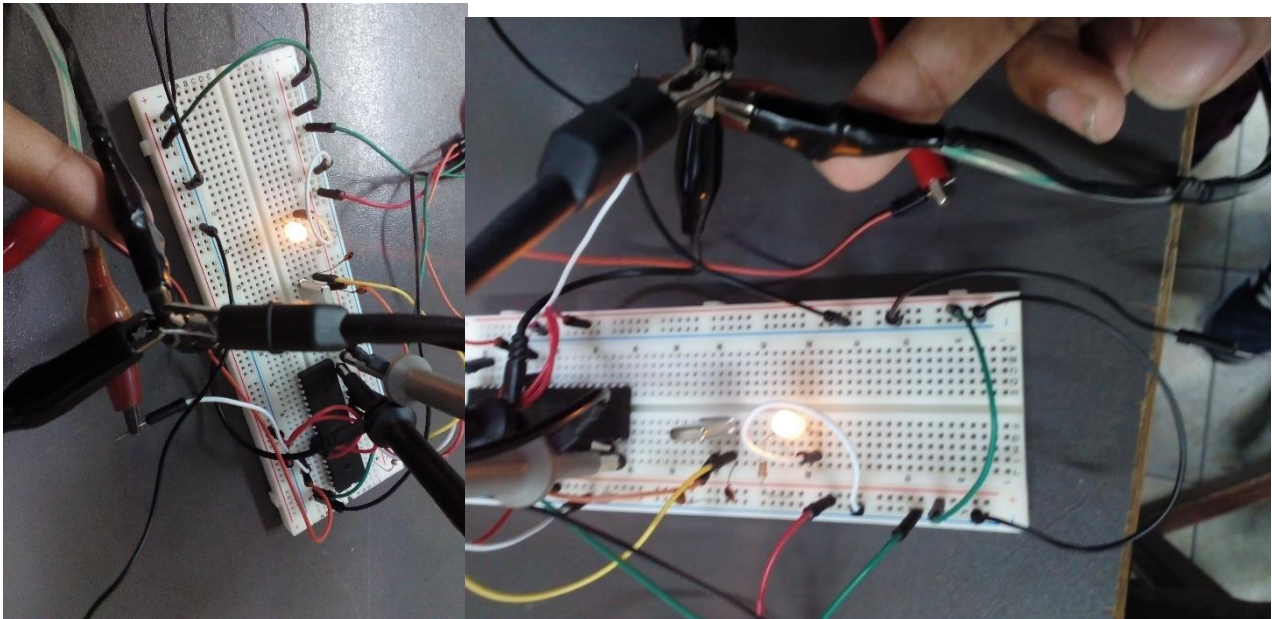
Del mismo modo, a continuación se muestra la simulación del circuito construido en Proteus 8:



2.1.4. Circuito Real

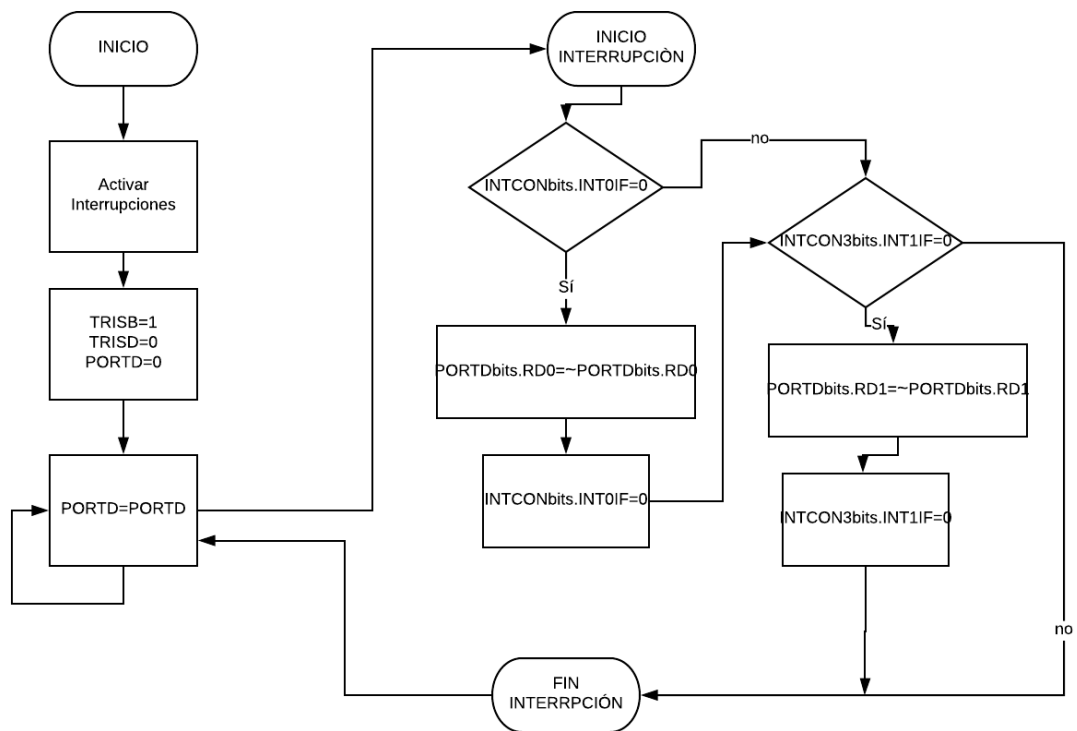
Para la primera simulación se realizó un circuito en Protoboard conectado a base de la simulación del Proteus y el manual del PIC18F4550. Asimismo, se

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

```
void INTO_init (void);
```

```
void INT1_init (void);
```

```
void main(void)
```

```
{
```

```
    TRISD=0;
```

```
    TRISB=1;
```

```
    INTCONbits.GIE=1;
```

```
    INTCONbits.PEIE=1;
```

```
    INTCONbits.INT0IE=1;
```

```
    INTCONbits.INT0IF=0;
```

```
    INTCON3bits.INT1IE=1;
```

```
    INTCON3bits.INT1F=0;
```

```
    INTCON2bits.INTEDG0=1;
```

```
    RCONbits.IPEN=0;
```

```
    while(1)
```

```

{
    PORTDbits.RD0=PORTDbits.RD0;
    PORTDbits.RD1=PORTDbits.RD1;
}

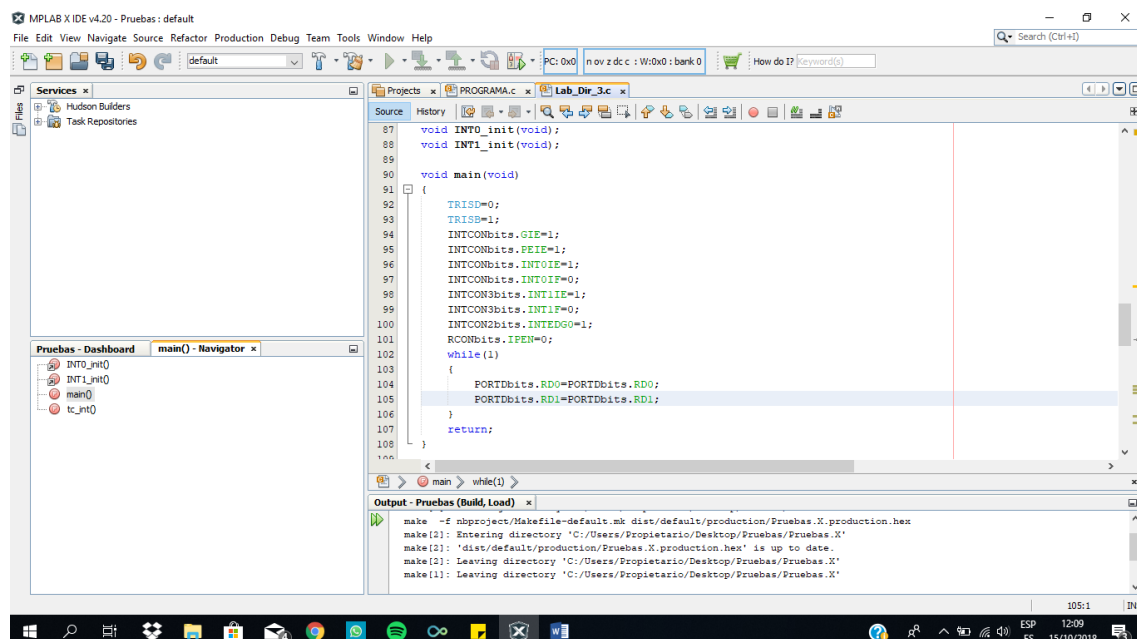
return;
}

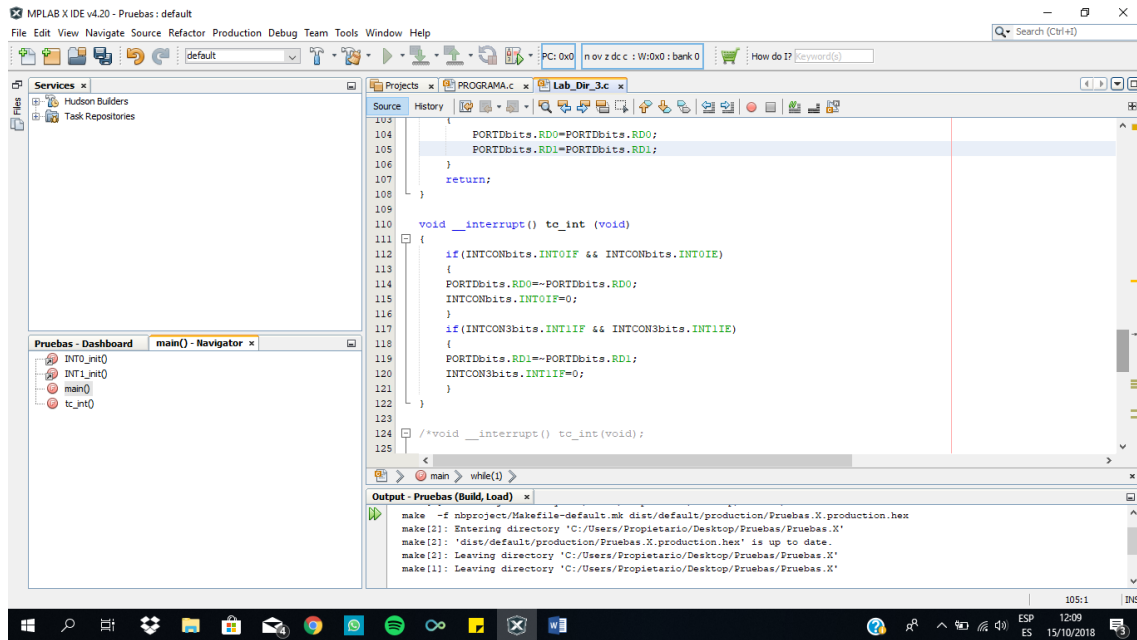
void __interrupt() tc_int (void)
{
    if (INTCONbits.INT0IF && INTCONbits.INT0IE)
    {
        PORTDbits.RD0=~PORTDbits.RD0;
        INTCONbits.INT0IF=0;
    }
    if (INTCON3bits.INT1IF && INTCON3bits.INT1IE)
    {
        PORTDbits.RD1=~PORTDbits.RD1;
        INTCON3bits.INT1IF=0;
    }
}

```

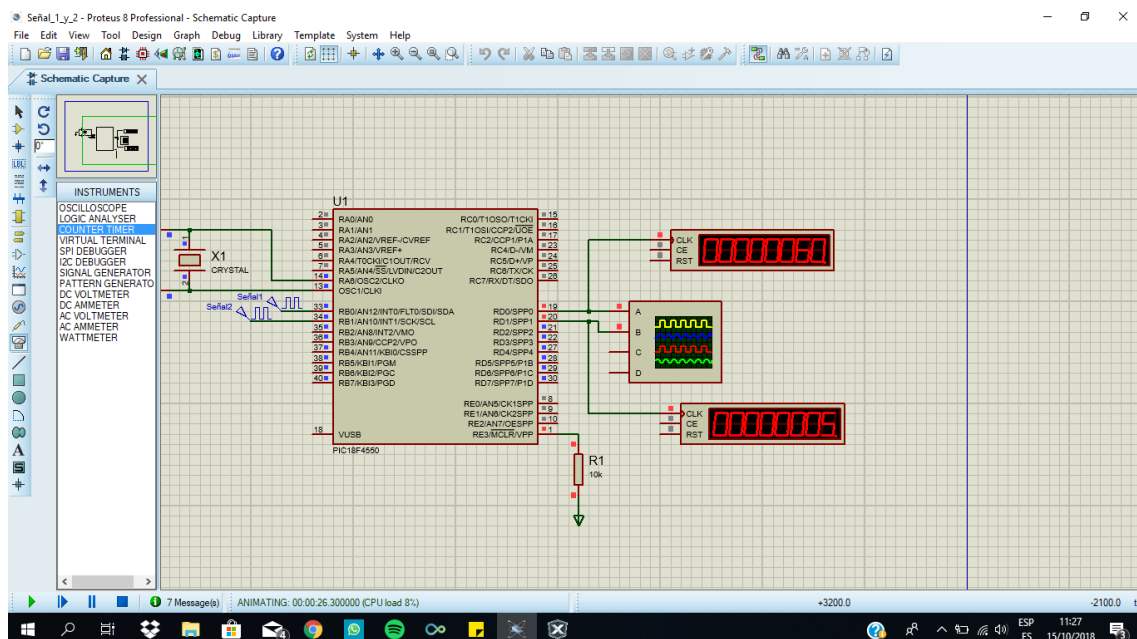
2.2.3. Lenguaje y Simulación

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





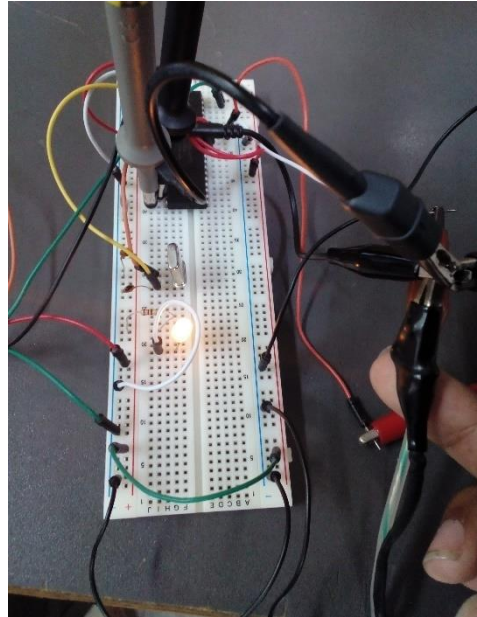
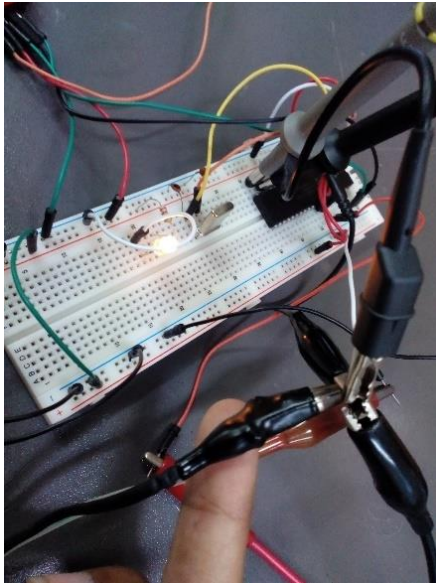
Del mismo modo, a continuación se muestra la simulación del circuito construido en Proteus 8:



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

[https://www](https://www.youtube.com/watch?v=Om-8DryS2g)
[.youtube.co](https://www.youtube.com/watch?v=Om-8DryS2g)
[m/watch?v=](https://www.youtube.com/watch?v=Om-8DryS2g)
[Om-](https://www.youtube.com/watch?v=Om-8DryS2g)
[8DryS2g](https://www.youtube.com/watch?v=Om-8DryS2g)



Observaciones

- Durante la realización de la verificación de las señales de salida en el osciloscopio se pudo observar que en ella se encontraba ruido, lo que genera que no se vea perfectamente la señal cuadrada. Sin embargo, la señal cuadrada no está del todo ausente en el osciloscopio.
- El ruido, según el encargado del laboratorio, se debía a los cables cocodrilos del generador de señal y el osciloscopio. Además, los cables usados en el circuito eran de baja calidad, por lo cual también fue un generador de ruido.
- Durante la realización del ejercicio 2, se verificó que la señal de salida RD0 había aumentado su frecuencia en 10HZ, siendo su valor de 60HZ de 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.
- Es necesario establecer los pines del puerto B como entradas digitales, ya que de esta manera las señales podrán ser reconocidas.
- Para el uso de las dos señales es necesario tener que activar, además de la INT0, la INT1 y realizar las respectivas configuraciones. Además, se debe tener en cuenta que en la función interrupción se pondrá tanto la INT0 como la INT1.
- Es fundamental escribir cada detalle del lenguaje C del programa de manera precisa para evitar errores.
- El regreso del bit INT0IF e INT1F a cero, durante la interrupción es muy esencial ya que de no hacerlo se reconocerá como que la interrupción ya se hubiera realizado y no se podrá generar un ciclo infinito.
- Las señales de salida tienen una frecuencia cuyo valor es la mitad de su respectiva entrada, 100Hz y 10Hz, cuando se trabajan independientemente, esto se debe a que durante la función de interrupción se genera una señal cuyo periodo es el doble de la inicial.



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.