ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

Departamento de Eléctrica y Electrónica

Carrera de Electrónica y Automatización

SISTEMAS BASADOS EN MCU

Práctica 3.6

MANEJO DEL TEMPORIZADOR EN LA FUNCIÓN CONTADOR A TRAVÉS DE INTERRUPCION PARA EL PIC16F877

Autor:

Iza Tipanluisa Alex Paul

Docente:

Ing. Amparo Meythaler

NRC: 4891

1) OBJETIVOS

- Consolidar el manejo de las interrupciones en lenguaje C.
- Realizar una aplicación con el temporizador 1, en la función contador, en lenguaje C para el PIC16F877.

2) MARCO TEORICO

INTERRUPCIONES EN LENGUAJE C

La rutina de interrupción va antes de la función main y consta de la directiva del preprocesador que corresponde Ejemplo:

#int_TIMER0, con esta directiva el programa sabe a qué dirección de programa debe de ir cuando se dispare esta interrupción (podemos tener otras, si el microcontrolador tiene una tabla de vectorización), y de la función de interrupción propiamente dicha, donde tenemos que incluir el código, por ejemplo:

```
#int_TIMER0

Void TIMER0_isr(void){

//El código de la interrupción.}
```

Dentro de la función main se incluiría lo siguiente:

```
enable_interrupts(INT_TIMER0);

enable_interrupts(GLOBAL);
```

La primera instrucción habilita la interrupción particular del Timer 0 por desbordamiento y la otra habilita las interrupciones globalmente, NO hay que preocuparse por resetear el flag de interrupción TOIF, ni de desactivar las interrupciones globales mientras se esté dentro del código de interrupción por si se produce otra simultáneamente, como hay que hacer cuando se trabaja en ensamblador, estas funciones lo hacen también pero de una manera transparente al programador, por eso es un lenguaje de alto nivel.

3) EQUIPOS Y MATERIALES

- PC con el paquete PIC C
- PC con el paquete PROTEUS.

4) ACTIVIDADES

1) Trabajo Preparatorio:

a) Realizar el diagrama de flujo, la codificación e implementación correspondiente, de un programa que cuente el número de personas que ingresan a un bus, la visualización será mediante un LCD. Cuando ingresa la persona 20 (conteo decimal) se cierra automáticamente la puerta (led), la misma que se abrirá luego de un tiempo. El contaje lo realiza el TIMER 1 a través de interrupción

b) Verifique el funcionamiento del MPLAB IDE y del PROTEUS.

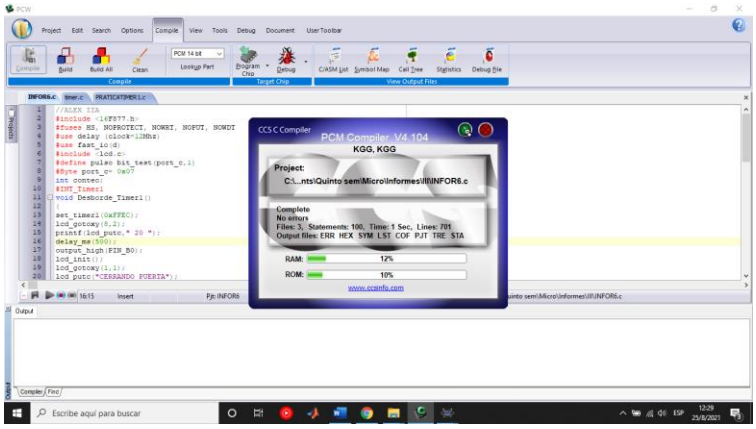

Ilustración 1, Compilación del programa

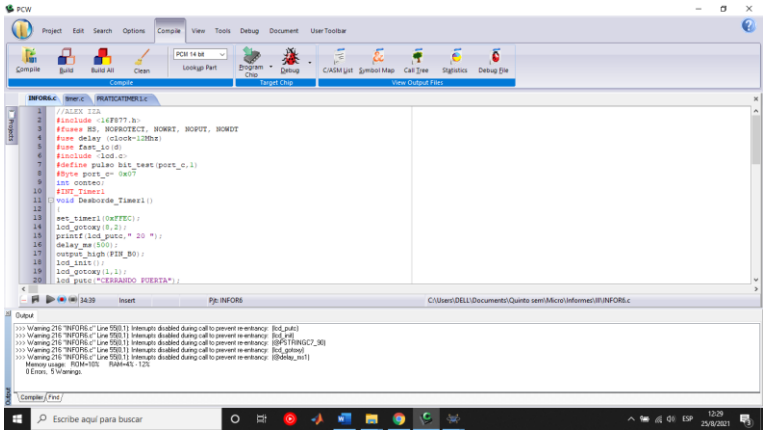

Ilustración 2, Compilación exitosa, 0 errores

5) RESULTADOS

- **Explique los errores cometidos en el ejercicio realizado (si los tuvo) y la forma de corregirlos.**

  Tuve un error al preguntar si el switch fue o no fue presionado; estaba preguntando por eso; lo solucione guardando el valor del timer1 en una variable y como esta variable cuenta de desde el dato inicial guardado le sume 20 ya que de lo contrario el valor era -20.

- **Si en el lenguaje C podemos utilizar los retardos con la directiva delay, ¿Cuándo y por qué sería necesario utilizar los timers en la función temporizador?**

  Cuando se requiera activar una interrupción interna se puede usar el Timer.

6) DISEÑO

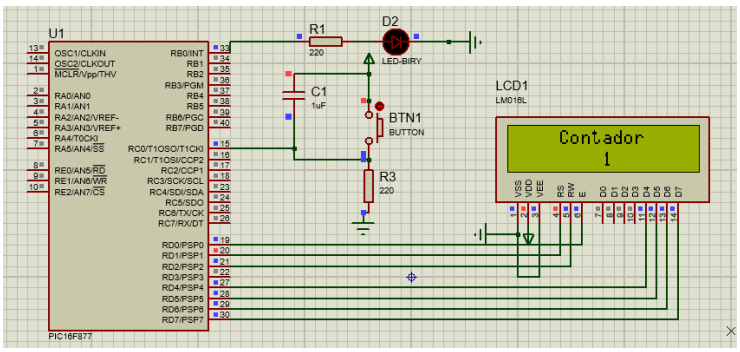

Ilustración 3, Visualización contador dato 1

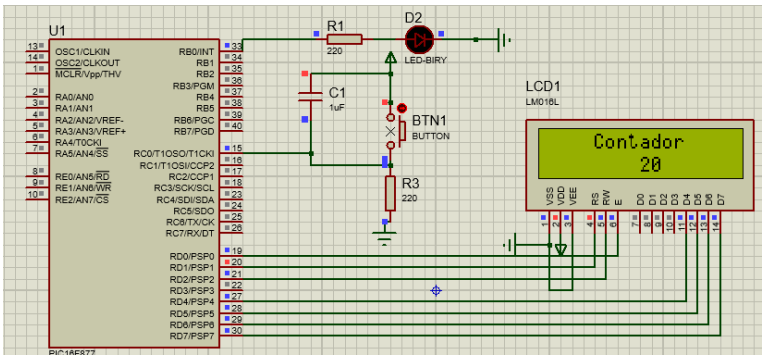

Ilustración 4, Visualización del contador dato 20

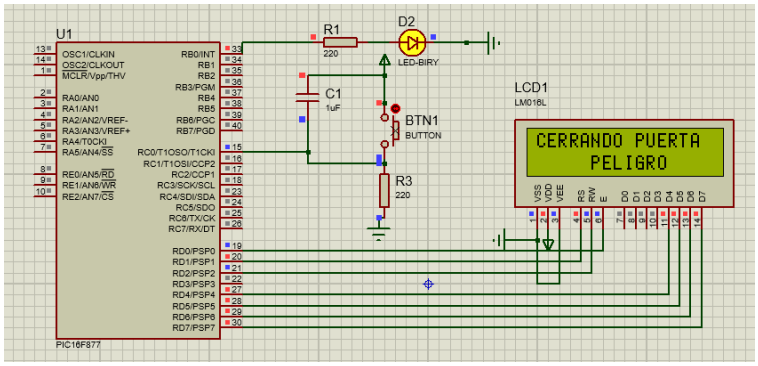

Ilustración 5, Visualización de la interrupción

7) DIAGRAMA DE FLUJO

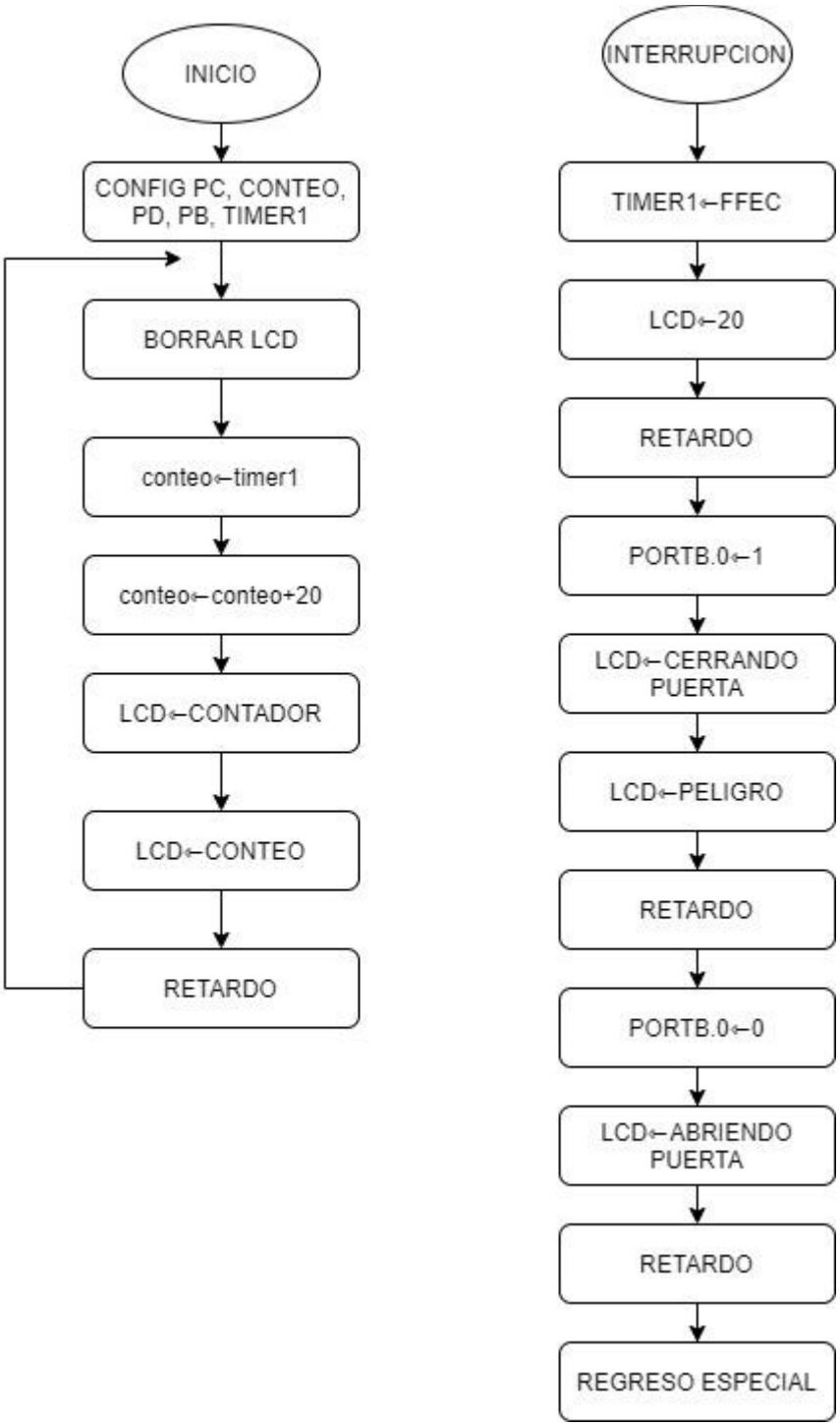

Diagrama 1

8) PROGRAMA

```
//ALEX IZA
#include <16F877.h>
#fuses HS, NOPROTECT, NOWRT, NOPUT, NOWDT
#use delay (clock=12Mhz)
#use fast_io(d)
#include <lcd.c>
#define pulso bit_test(port_c,1)
#Byte port_c= 0x07
int conteo;
#INT_Timer1
void Desborde_Timer1()
{
set_timer1(0xFFEC);
lcd_init();
lcd_gotoxy(8,2);
printf(lcd_putc," 20 ");
delay_ms(500);
output_high(PIN_B0);
lcd_gotoxy(1,1);
lcd_putc("CERRANDO PUERTA");
lcd_gotoxy(6,2);
lcd_putc("PELIGRO");
delay_ms(1000);
output_low(PIN_B0);
lcd_gotoxy(1,1);
lcd_putc("ABRIENDO PUERTA");
```

```
lcd_gotoxy(6,2);
lcd_putc("CUIDADO");
delay_ms(1000);
lcd_putc("\f");
}
void main()
{
setup_timer_1(T1_EXTERNAL|T1_clk_out);
set_timer1(0xffec);
enable_interrupts(INT_Timer1);
enable_interrupts(GLOBAL);
lcd_init();

while(true)
{
lcd_putc("\f");
conteo=get_timer1();
conteo=conteo+20;
lcd_gotoxy(5,1);
lcd_putc("\Contador");
lcd_gotoxy(8,2);
printf(lcd_putc," %d ",conteo);
delay_ms(200);
}
}
```

9) CONCLUSIONES

- Se ha concluido que la función Timer1 cuenta desde un dato inicial dado hasta FFFF
- Se ha concluido que la interrupción del Timer1 puede ser activada mediante una perturbación externa
- Se ha concluido que para esta practica no es recomendable preguntar si el switch fue presionado o no fue presionado ya que este pin esta usado para que se realice la interrupción del Timer1 por lo cual es mejor usar el desbordamiento del timer para que se genere la interrupción

10) RECOMENDACIONES

- Se recomienda revisar la aplicación y forma de trabaja de un Timer ya que l desconocimiento de este puede hacer que el programa sea extenso
- Se recomienda conocer correctamente el uso de los #fuses ya que tienen una función especifica cada uno de ellos en el PIC16F877
- Se recomienda que el pulsante vaya aterrizado a tierra

11) BIBLIOGRAFIA

Meythaler, A. (2021). Sistemas basados en MCU. Ecuador: UFA ESPE