



**UNIVERSIDAD DE GUADALAJARA**

**CENTRO UNIVERSITARIO DE CIENCIA EXACTAS**

**SEMINARIO DE PROBLEMAS DE PROGRAMACIÓN DE SISTEMAS EMBEBIDOS.**

**Laboratorio #8 – Interrupciones**

**ALUMNO: MENESES LÓPEZ ARISAI RICARDO**

**DOCENTE: ALVARADO RODRIGUEZ FRANCISCO JAVIER**

**08 NOVIEMBRE DEL 2019**

## MATERIALES

### COMPONENTES.

- PROTOBOARD
- DIODO LED
- CABLE PARA PROTO
- PINZAS DE CORTE / AGARRE
- FUENTE DE VOLTAJE 5V
- RESISTENCIA 220Ω
- PROGRAMADOR PARA PIC – MASTER PROG.

### MICROCONTROLADOR

- PIC16F887

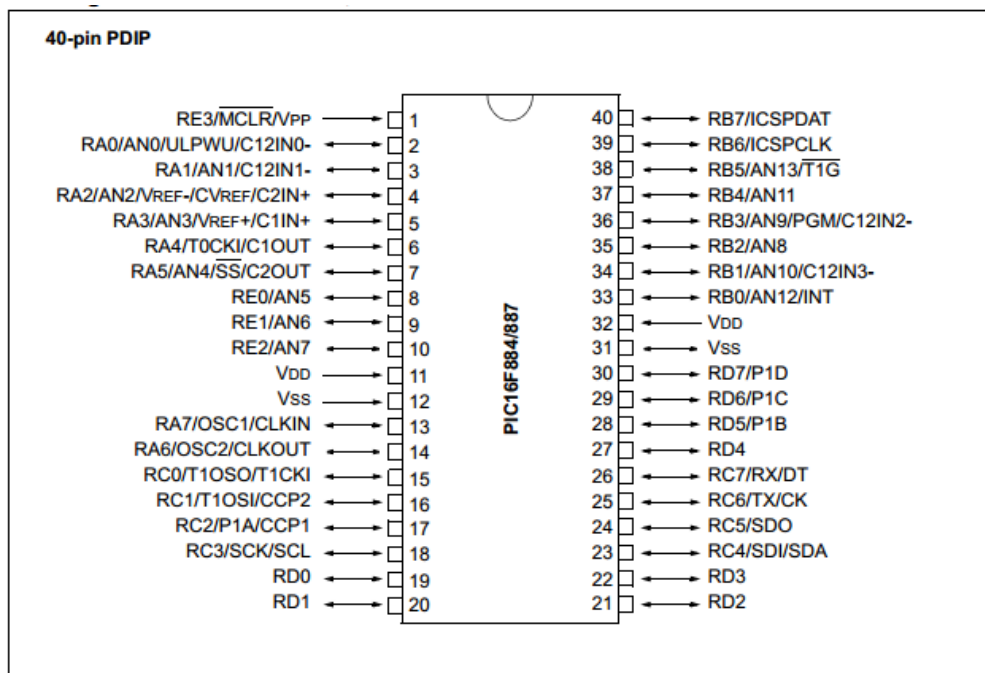
### SOFTWARE

- MIKROC PRO FOR PIC
- PROTEUS DESIGN SUITE
- MASTER – PROG+

## 1. OBJETIVO DEL PROYECTO

- El propósito de esta práctica es configurar una interrupción utilizando la interrupción local del TMR0IF del registro INTCON.

## 2. MARCO TEÓRICO



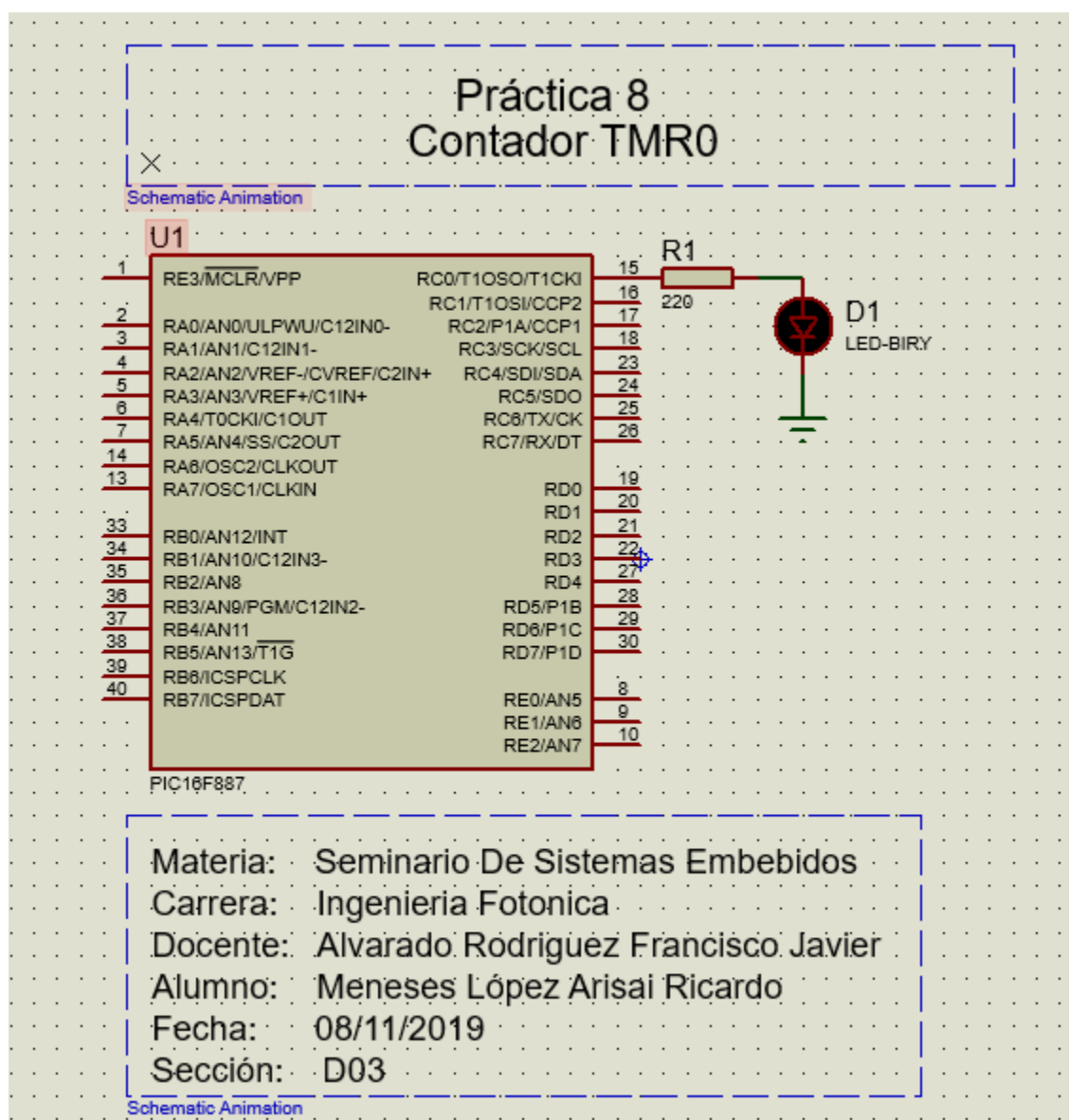


### 3. DESARROLLO

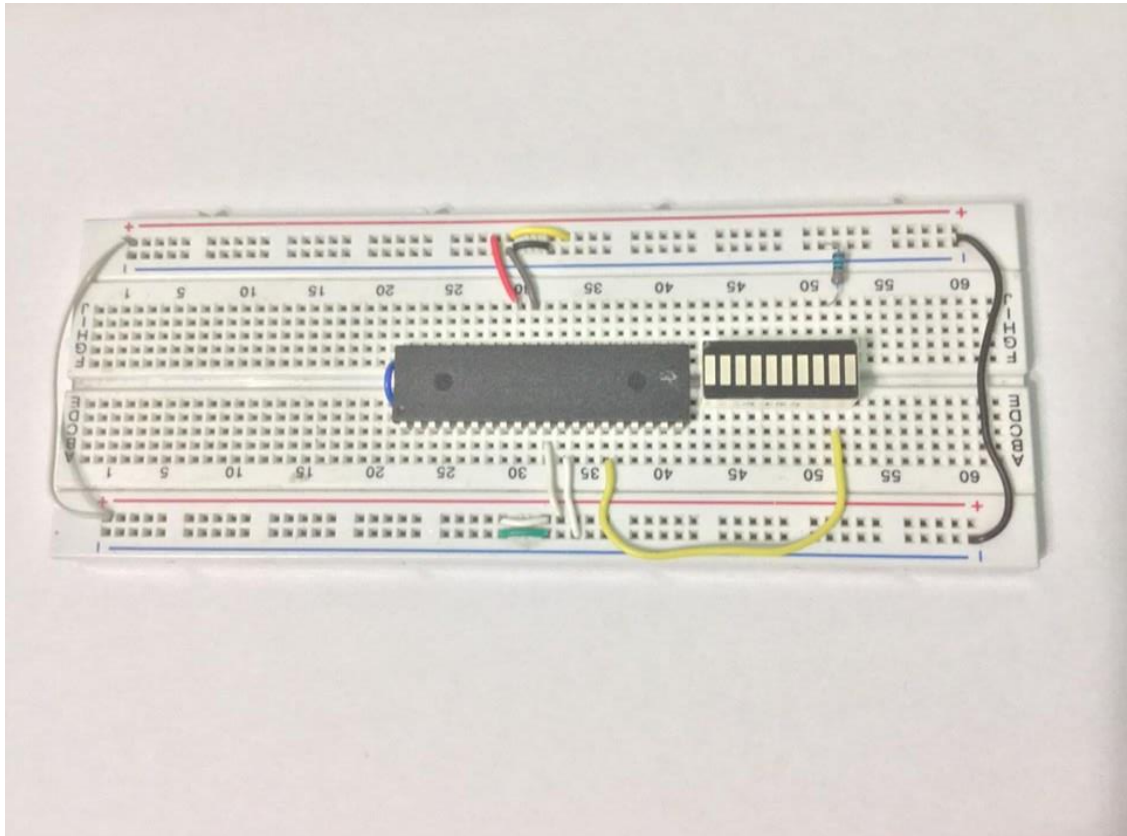
Una interrupción es un mecanismo que permite ejecutar un bloque de instrucciones interrumpiendo la ejecución de un programa, y luego restablecer la ejecución del mismo sin afectarlo directamente. De este modo un programa puede ser interrumpido temporalmente para atender alguna necesidad urgente del computador y luego continuar su ejecución como si nada hubiera pasado.

La intención de la práctica es extraer el valor del TMR0 hacia el PORTC y ver cómo se refleja la interrupción creada en una función interrup() que hará la función de mandar el TMR0IF a LOW y el PORTC.RC0 al negado del mismo cada vez que se cumpla un conteo por parte de la función interrupt().

### 4. SIMULACIÓN



## 5. PROTOBOARD



## 6. PROGRAMACIÓN

### CÓDIGO

```
unsigned char cuenta;

void main()
{
    OPTION_REG.T0CS = 0; //Contador interno (Modo Timer).
    OPTION_REG.T0SE = 1; //Flanco de bajada
    OPTION_REG.PSA = 0; //Asignación del Pre-escalador al TMR0 por lo tanto
                        // el pre-escalador no funciona para el WDT
    OPTION_REG.PS2 = 1; // Debido a lo anterior se toma la Pre-escala 1:256
    OPTION_REG.PS1 = 1; // Tiempo = (Fosc/4) x 256(pre-escala) x 256 (TMR0)
    OPTION_REG.PS0 = 1; // Tiempo = (Fosc/4) x 65536

    INTCON.GIE = 1; //Habilitacion global de las interrupciones
    INTCON.T0IE = 1; //Habilitacion de la Interrupción de la Interrupcion Local
    TMR0 = 0x00;    //Inicializacion del TMR0

    TRISC.RC0 = 0; // Bit 0 del puerto C como salida
    PORTC.RC0 = 0; // Limpieza del puerto

    while(1)
    {
        //PORTC = TMR0; //Valor del TMR0 en el PORTC
    }
}

void interrupt()
{
    if(INTCON.TMR0IF)
    {
        cuenta++;
        if(cuenta==7)
        {
            cuenta = 0;
            PORTC.RC0 =~ PORTC.RC0;
            INTCON.TMR0IF = 0;
        }
        INTCON.TMR0IF = 0;
    }
}
```

## 7. CONCLUSIONES

- Cada vez que la interrupción se realiza se refleja en cómo se apaga el Diodo Led conectado al C0 lo que hace que parezca una secuencia intermitente, pero en realidad es una interrupción repetida del sistema interno.

## 8. BIBLIOGRAFÍA

[ 1 ] - J. M. Morán Loza. Programación de Sistemas Embebidos Con Aplicaciones Para El PIC16F8XX. MEXICO: PEARSON, 2014.