



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIA EXACTAS

SEMINARIO DE PROBLEMAS DE PROGRAMACIÓN DE SISTEMAS EMBEBIDOS.

Proyecto #3 – Temporizador con Display

ALUMNO: MENESES LÓPEZ ARISAI RICARDO

DOCENTE: ALVARADO RODRIGUEZ FRANCISCO JAVIER

19 NOVIEMBRE DEL 2019

MATERIALES

COMPONENTES.

- PROTOBOARD
- BUZZER DE 5 V
- DIP-SWITCH 4 BUSES
- TRANSISTORES BC337
- POTENCIOMETRO DE 10 K Ω
- CABLE PARA PROTO
- DISPLAYS DE 4 DIGITOS CATODOS
- PINZAS DE CORTE / AGARRE
- FUENTE DE VOLTAJE 5V
- RESISTENCIAS 2K Y 220 Ω
- PROGRAMADOR PARA PIC – MASTER PROG.

MICROCONTROLADOR

- PIC16F887

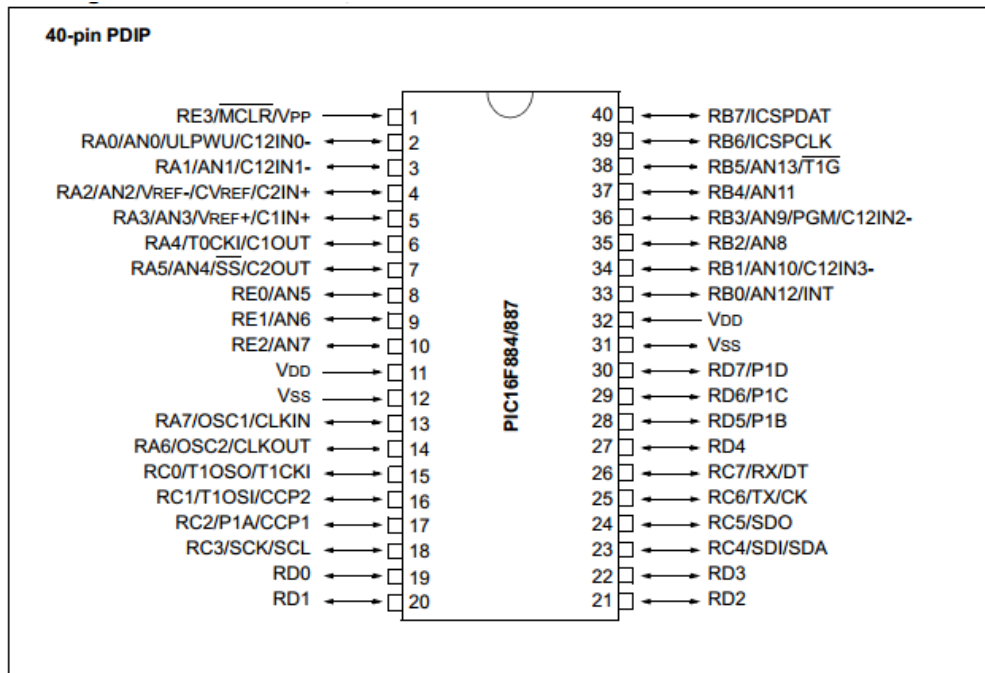
SOFTWARE

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

1. OBJETIVO DEL PROYECTO

- Realizar un Temporizador con Display de 4 dígitos que tenga control de segundos, minutos, botón de Stop y botón para activar/desactivar alarma.

2. MARCO TEÓRICO



3. DESARROLLO

Cuando hablamos de activar/desactivar alarma quiere decir que, si el conteo del tiempo finaliza y la alarma está desactivada, el buzzer no se activará sino hasta que dicha alarma esté activada y el conteo haya concluido.

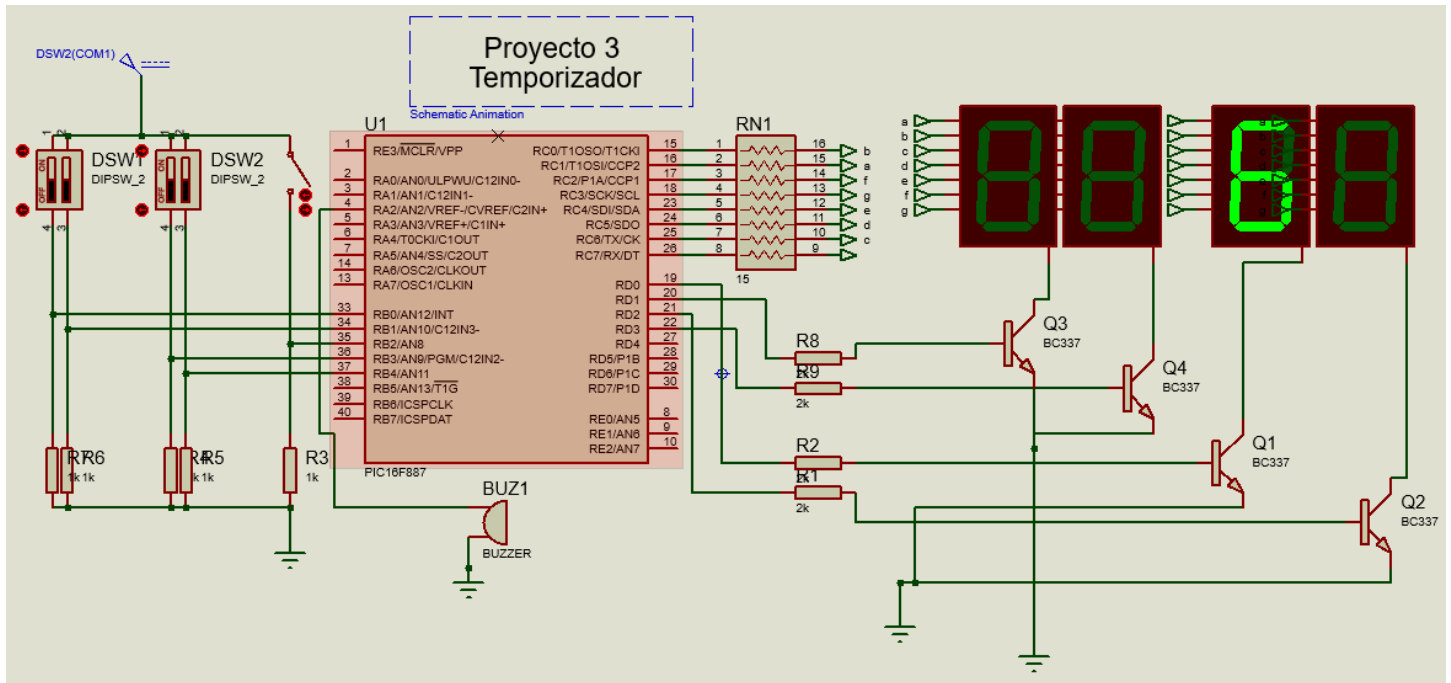
Se contemplarán dos Switches para ajustar los minutos y segundos.

Cuando la alarma halla terminado el proceso de multiplexado se desactivará para ahorrar energía y tareas al PIC.

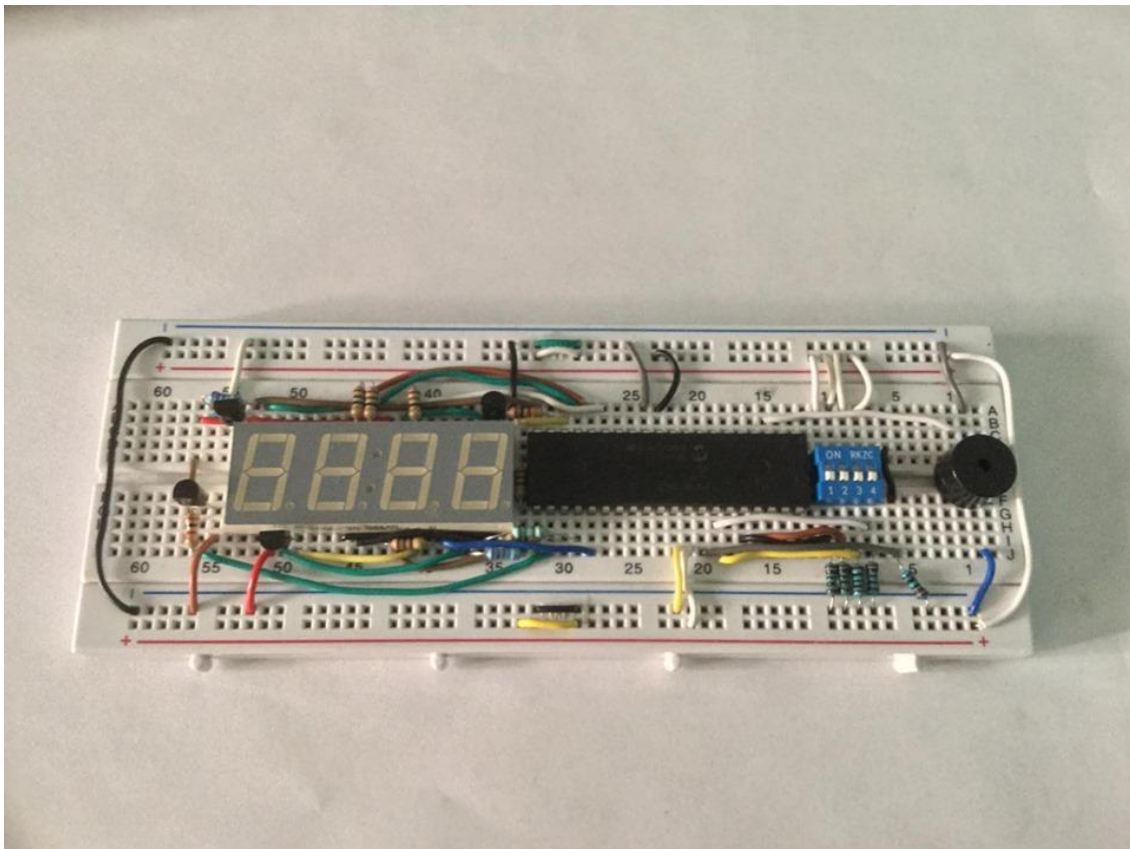
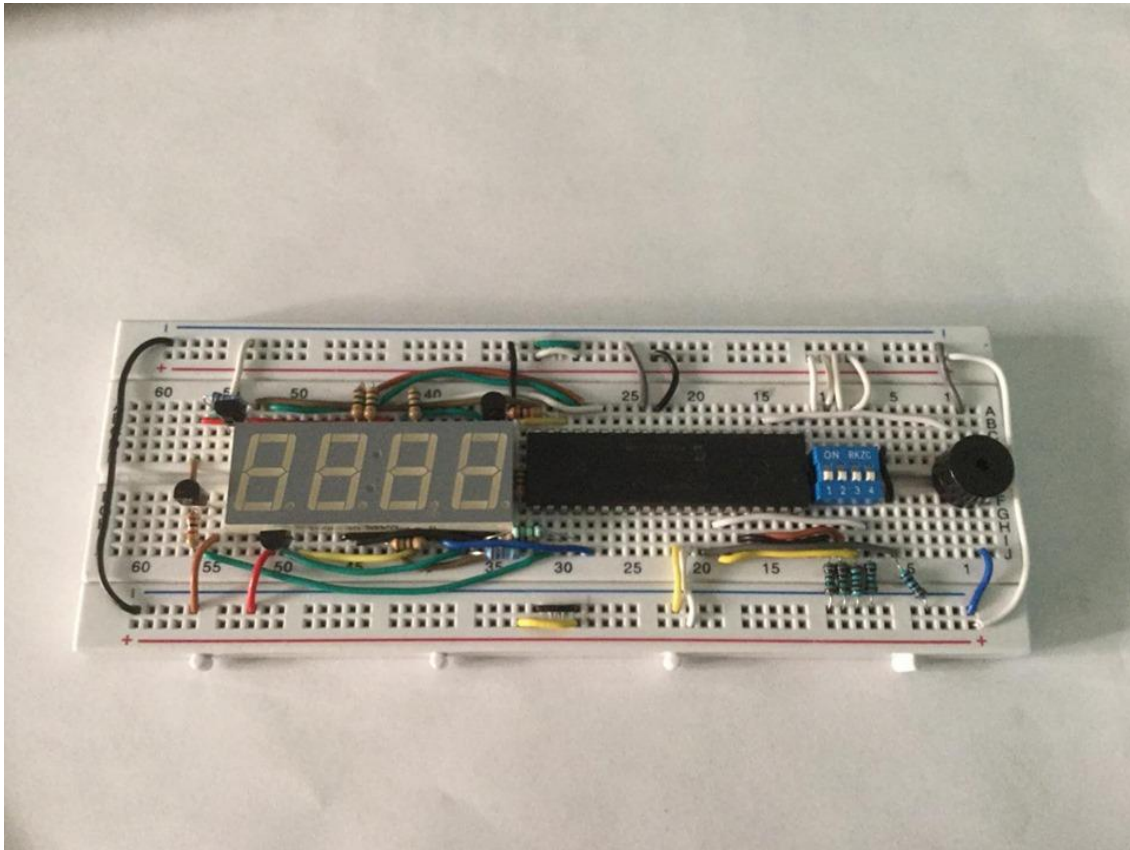
La lógica del código es similar al anterior proyecto 2 que se basaba en multiplexar una secuencia ascendente/descendente para un display de dos dígitos. Así que básicamente se reciclaron algunas funciones de dicho proyecto.

Cuando la Alarma es activada, pero no hay ninguna alarma programada, el conteo pasa desde 60 minutos y 60 segundos. El cuál se podría modificar para que no comience ningún conteo sin alguna alarma detectada.

4. SIMULACIÓN



5. PROTOBOARD



6. PROGRAMACIÓN

CÓDIGO

```
unsigned char U2=0,U1=0,D1=0,D2=0,M1=0,M2=0,S1=0,S2=0;
unsigned char TABLA[]={0x77,0x41,0x3B,0x6B,0x4D,0x6E,0x7E,0x43,0x7F,0x6F};
unsigned int band=0,alarma=0,cont=0;

void TEMPORIZADOR();
void alarma();
void multiplexado(unsigned char veces);

void main()
{
    TRISA=0xFD;
    TRISC=0x00;
    TRISD=~0x0F;
    OPTION_REG = OPTION_REG & 0x7F,
    ANSELH=0x00;
    PORTB=0x3F;
    PORTA=0x00;

    while(1)
    {
        if(PORTB.RB2==1) // ACTIVAR/DESACTIVAR ALARMA
        { band=1; cont=1; } else { band=0; alarma=0; PORTA.RA1=0; }
        if(PORTB.RB3==1) // AJUSTE DE MINUTOS
        { if(M1==6){M1=0; } else{ M1++; } D1=M1; }
        else if(PORTB.RB4==1) // AJUSTE DE SEGUNDOS
        { if(S1==6){S1=0; } else{ S1++; } D2=S1; }

        else if(PORTB.RB1==1) // REINICIO DE ALARMA
        {
            D1=0; U1=0; D2=0; U2=0; band=0; alarma=0; PORTA.RA1=0;
        }
        else if(PORTB.RB0!=1) // STOP/CONTINUE
        {
            if(alarma==1) //ALARMA
            {
                if(band==0) { alarma=0; }
                else{ PORTA.RA1=1; cont=0; band=0; }
            }
            else if((alarma==0)&&(cont==1))
            { TEMPORIZADOR(); }
        }

        if(alarma!=1){ multiplexado(25); }
    }
}
```

```

void TEMPORIZADOR()
{
    if(U2==0)    //Unidades SEGUNDOS
    {
        U2=9;
        if(D2==0) //Decenas SEGUNDOS
        {
            D2=6; U2=0;

            if(U1==0) // Unidades Minutos
            {
                U1=9;
                if(D1==0) // Decenas Minutos
                {
                    U2=0; D2=6; U1=0; D1=6;
                }
                else{ D1--; }
            }
            else{ U1--; }
        }
        else{ D2--; }
    }
    else{ U2--; }

    if(U2==0&&U1==0&&D1==0&&D2==0)
    { alarma=1; cont=0; }
}

```

```

void multiplexado(unsigned char veces)
{
    PORTD.RD0=0; PORTD.RD1=0; PORTD.RD2=0; PORTD.RD3=0;

    while(veces)
    {
        PORTC=TABLA[U2]; //Unidades Segundos
        PORTD.RD2=1;
        delay_ms(3);
        PORTD.RD2=0;

        PORTC=TABLA[D2]; //Decenas Segundos
        PORTD.RD0=1;
        delay_ms(3);
        PORTD.RD0=0;

        PORTC=TABLA[U1]; //Unidades Minutos
        PORTD.RD3=1;
        delay_ms(3);
        PORTD.RD3=0;

        PORTC=TABLA[D1]; //Decenas Minutos
    }
}

```

```
PORTD.RD1=1;  
delay_ms(3);  
PORTD.RD1=0;
```

```
    veces--;  
    }  
}
```

7. CONCLUSIONES

- La lógica del conteo de minutos y segundos es muy eficiente.
- Sólo se puede programar la alarmar por cada decima (cada 10) y las unidades no se ajustan.

8. BIBLIOGRAFÍA

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