

PIC MICRO ESTUDIO

Temporizador con el 16F872

www.electronicaestudio.com

Guía de Uso

Temporizador con el PIC16F872

Los “temporizadores” los encontramos en muchos lugares de nuestra vida diaria, generalmente integrados en aparatos domésticos. Por ejemplo en los hornos de microondas para controlar el tiempo de calentamiento de nuestros alimentos, o en las lavadoras de ropa para seleccionar el tiempo lavado.

También a escala industrial las aplicaciones de los temporizadores son múltiples. Los hay para controlar el tiempo de arranque de algún proceso, se usan en máquinas herramientas, en dosificadores, fotografía, etc. La lista es interminable.

Muchos de estos temporizadores pueden resultar muy caros o difíciles de conseguir. En este artículo el lector aprenderá a diseñar sus propios temporizadores mediante el uso del **PIC16F872**. Es por esta razón que [Picmicro Estudio](#) ofrece este módulo con el nombre de **Tempoind (Clave 714)**. El cual consta de las siguientes características:



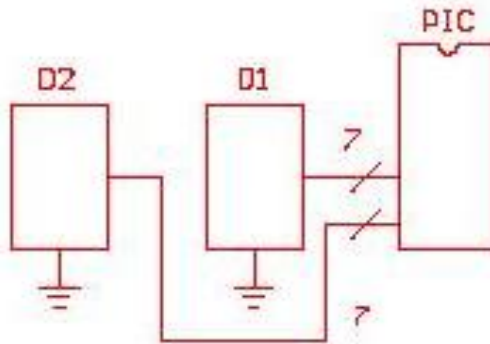
- **2 Displays de 7 segmentos**
- **1 Microcontrolador PIC16F872**
- **1 cristal de 4 Mhz**
- **1 Relevador de +12V**
- **Modo Descendente**
- **Valor a temporizar por el usuario, grabado en la EEPROM interna del microcontrolador.**

Además este modulo puede funcionar para medir:

- **Segundos 0 a 99**
- **Minutos 0 a 9.9**
- **Minutos 0 a 99**
- **Horas 0 a 99**

Descripción del Circuito

Los 7 segmentos de cada uno de los 2 Displays están conectados directamente al **PIC16F872**. Tal y como se muestra a continuación.



En la Fig. 1 se muestra el diagrama esquemático del circuito.

Modo de Utilización:

Escoja el Rango de operación.

Mediante los puentes que se encuentran a un lado de los botones se puede seleccionar el rango de operación. Vea la siguiente tabla

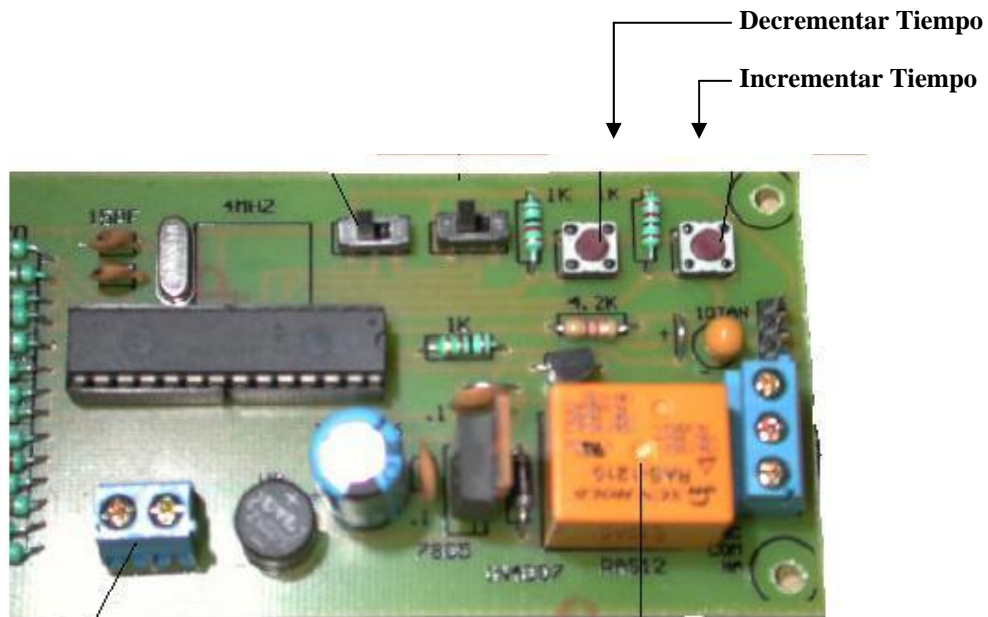
0 a 99 segundos	izquierda	izquierda
0 a 9.9 minutos	derecha	izquierda
0 a 99 minutos	izquierda	derecha
0 a 99 horas	derecha	derecha

Ajuste del tiempo

Observe los dos botones Pulsadores, los cuales tienen como función incrementar o decrementar, el tiempo que se desea temporizar. Este número queda memorizado en la memoria EEPROM del microcontrolador

Operación

Una vez que se selecciona el tiempo, se procede a presionar **el botón Iniciar**, (externo) el cual activará inmediatamente el relevador e iniciará la cuenta regresiva. Cuando en los dos Displays se visualice el número 00, este indicará el final de la temporización desactivando el relevador. Después de unos instantes el display digital muestra nuevamente el tiempo seleccionado y se puede iniciar otro ciclo de operación. Se hace notar que el programa regresa al inicio, en donde el usuario podrá nuevamente seleccionar el tiempo que desee.



Cabe mencionar que el microcontrolador debe estar programado con el programa **TW5.bas**, el cual puede encontrar en el disco adjunto a este producto y que a continuación se procederá a explicar.

Código Fuente "TW2.bas"

Primeramente se procede a declarar la variables a utilizar en este caso son:

```

tiempo      var      byte
segundos   var      byte
minutos    var      byte
Selector   var      byte
I          var      byte
Y         var      byte
dig_1     var      byte
dig_2     var      byte
incrementa VAR      PORTA.1
decrementa VAR      PORTA.2

SWITCH_SALIDA VAR PORTA.5
PONER_tiempo VAR PORTA.4
boton        var PORTA.0
JumperIz     VAR PORTA.4
JumperDer    VAR PORTA.3

```

El siguiente paso es configurar los puertos como salidas, esto se hace con las siguientes instrucciones.

```

ADCON1 = 7
      TRISA = %11111111
      TRISB = 0
      TRISC = 0

@ device      pic16F872, xt_osc, wdt_off, pwrt_on, bod_on,
lvp_off,CPD_OFF,WRT_OFF,DEBUG_OFF, protect_on

EEPROM 0,[55] 'Tiempo inicial al momento de grabar el PIC

```

Como se observa, existe una configuración especial para el Puerto A, ya que este puede actuar como entradas analógicas y lo que se quiere en este caso, es que el Puerto A actúe como entradas de señales digitales, por esta razón se utiliza la instrucción **ADCON1=7**.

Una vez configurados los puertos, se procede a insertar una línea en lenguaje ensamblador, esto se hace con el símbolo **@** la cual establecerá al momento de efectuar la programación, los parámetros de configuración de este microcontrolador. Acto seguido se procede a grabar en la dirección 0 de la memoria EEPROM del microcontrolador, el dato 55.

La siguiente instrucción genera un salto incondicional a la etiqueta inicio

Goto Inicio

Inicio:

```
Selector = 0
PORTB = 0
PORTC = 0
```

Inicia_Timer:

```
read 0,tiempo
```

Checa_Botones:

```
dig_1 = tiempo dig 0
dig_2 = tiempo dig 1
Gosub Visuliza_display
```

```
IF boton=1 THEN Ejecutar
```

```
if incrementa=1 then           'si boton incre apretado
  gosub INC_TIEMPO
  Goto Checa_Botones
ENDIF
```

```
if decrementa=1 then         'si boton decr apretado
  gosub DEC_TIEMPO
```

```

Goto Checa_Botones
ENDIF
GOTO Checa_Botones

```

La variable **tiempo** será la encargada de guardar el numero que el usuario desea temporizar.

El programa queda en un bucle, esperando que usuario presione cualquier botón de los que están disponibles, como inicialmente el usuario debe seleccionar un tiempo, el programa entra en las subrutinas **INC_TIEMPO** o **DEC_TIEMPO**

```

'-----
INC_TIEMPO:
  IF tiempo < 99 THEN
    tiempo = tiempo + 1
    Pause 200
    goto Fin_INC_TIEMPO
  ENDIF
  IF tiempo >= 99 Then
    tiempo=0
    Pause 200
  ENDIF
Fin_INC_TIEMPO:
  RETURN
'-----
DEC_TIEMPO:
  IF tiempo > 0 Then
    tiempo = tiempo - 1
    Pause 200
    goto Fin_DEC_TIEMPO
  Endif
  IF tiempo <= 0 THEN
    tiempo= 99
    Pause 200
  Endif
Fin_DEC_TIEMPO:
  RETURN

```

Dichas subrutinas, lo único que hacen es incrementar o decrementar la variable **tiempo**. Como resultado de entrar en estas subrutinas, se obtiene en los Displays el número en el que el usuario desee, hasta un máximo de 99 y como mínimo de 00. Se observa que existe una pausa, dicha condición ejecuta un tiempo de espera para eliminar el efecto de los rebotes cada vez que se aprietan físicamente los botones incrementar o decrementar.

Cuando el usuario presione sobre el botón **Inicio** (ver diagrama esquemático) este tendrá **un valor de un “1” lógico**, por lo tanto el programa ejecutará un salto incondicional a la etiqueta **Ejecutar**.

Ejecutar:

```

write 0,tiempo
Gosub Calcula_Selector
    
```

Aquí se observa que: con la instrucción write se graba el “tiempo” en la memoria interna EEPROM del micro y que con la subrutina **Calcula_Selector** se determina el valor de la variable Selector en función de la posición de los puentes de ajuste del rango.

'-----

Calcula_Selector:

```

Selector = 0
Selector.0 = JumperIz
Selector.1 = JumperDer
    
```

Return

A continuación se realiza un bucle que representa el retardo de tiempo que se a escogido:

Otro_delay:

```

dig_1 = tiempo dig 0
dig_2 = tiempo dig 1
Gosub Visualiza_display

If tiempo=0 Then
low SWITCH_SALIDA
    
```



```
    pause 3000
    goto Inicia_Timer
Else
    high SWITCH_SALIDA
Endif

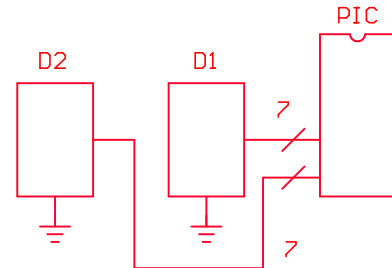
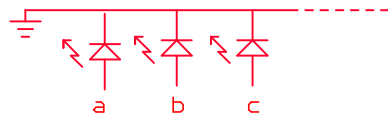
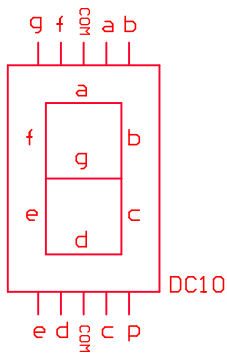
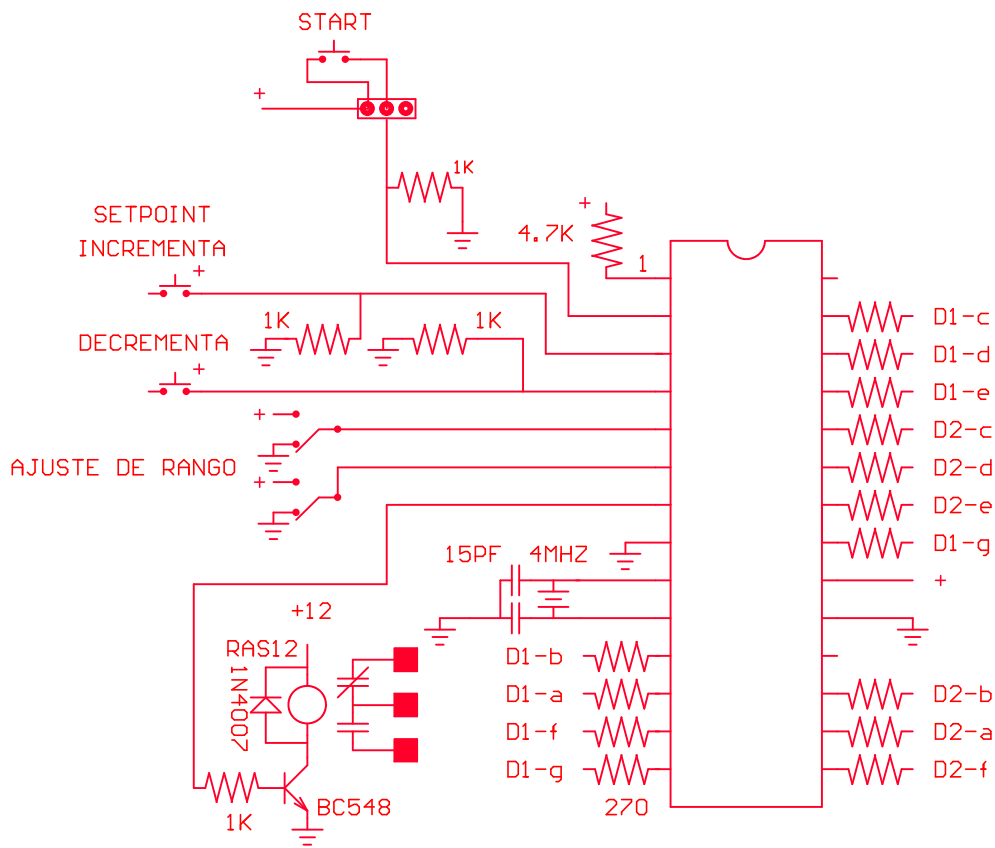
Gosub Ejecuta_Delay

tiempo= tiempo -1

goto Otro_delay
```

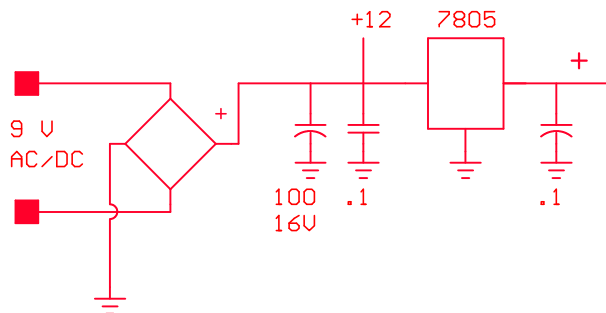
Si el tiempo es diferente de 00, se enciende el relevador y mediante la subrutina **Gosub Ejecuta_Delay** se da la pausa correspondiente al rango ajustado.

En este bucle después de cada pausa , la variable “**tiempo**” se decrementa en una unidad hasta que al llegar a 0, se desactiva el relevador y se inicia un nuevo ciclo.



AJUSTE DE RANGO

- 0 A 99 SEGUNDOS
- 0 A 9.9 MINUTOS
- 0 A 99 MINUTOS
- 0 A 99 HORAS



PIC micro Estudio

TEMPOIND

TER-1

Fig. 1