

Microcontroladores.

Microcontroladores.



Ambrosio España Iván
Carrisoza González Brandon Omar
Herrera Reyes Jovanni David
Saules Jhonatan

Instituto Politécnico Nacional – Escuela Superior de Cómputo.

Profesor: Pérez Pérez José Juan.

Palabra con interrupciones.

El display de 7 segmentos es un componente que se utiliza para la representación de números en muchos dispositivos electrónicos. Cada vez es más frecuente encontrar LCD's en estos equipos (debido a su bajísima demanda de energía), todavía hay muchos que utilizan el display de 7 segmentos por su simplicidad. Este elemento se ensambla o arma de manera que se pueda activar cada segmento (diodo LED) por separado logrando de esta manera combinar los elementos y representar todos los números en el display (del 0 al 9). El display de 7 segmentos más común es el de color, por su facilidad de visualización.

Cada elemento del display tiene asignado una letra que identifica su posición en el arreglo del display. Ver el gráfico arriba

Si se activan todos los segmentos: "a,b,c,d,f,g" se forma el número "8"

Si se activan sólo los segmentos: "a,b,c,d,f," se forma el número "0"

Si se activan sólo los segmentos: "a,b,g,e,d," se forma el número "2"

Si se activan sólo los segmentos: "b,c,f,g," se forma el número "4"

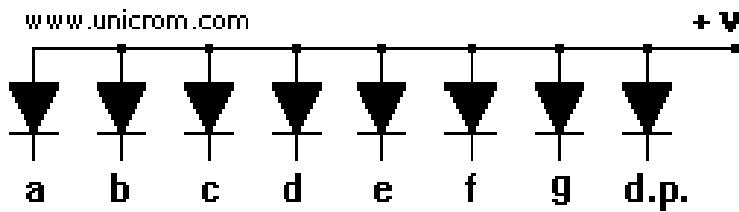
Si se activan sólo los segmentos: "a,b,g,c,d," se forma el número "3"

Si se activan sólo los segmentos: "a,f,g,c,d," se forma el número "5"

p.d. representa el punto decimal

El display ánodo común.

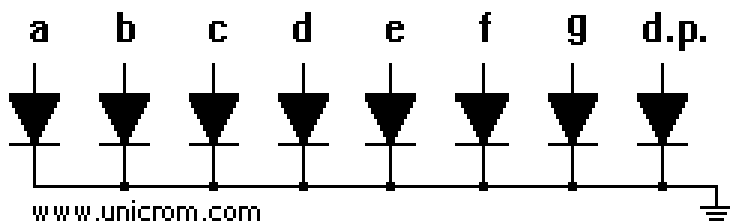
En el display ánodo común, todos los ánodos de los diodos LED unidos y conectados a la fuente de alimentación. En este caso para activar cualquier elemento hay que poner el cátodo del elemento a tierra a través de una resistencia para limitar la corriente que pasa por el elemento



Arreglo de LED's en un display de 7 segmentos ánodo común.

El display cátodo común.

El display cátodo común tiene todos los ánodos de los diodos LED unidos y conectados a tierra. Para activar un segmento de estos hay que poner el ánodo del segmento a encender a V_{cc} (tensión de la fuente) a través de una resistencia para limitar el paso de la corriente



Arreglo de LEDs en un display de 7 segmentos cátodo común

También hay display alfanuméricos que permiten representar tanto letras como números.

Es una de las características de los microcontroladores, de las más importantes que constituye la capacidad de sincronizar la ejecución de programas con acontecimientos externos; es decir, cuando se produce una interrupción, el micro automáticamente deja lo que está haciendo, va a la dirección 04h de programa y ejecuta lo que encuentre a partir de allí hasta encontrarse con la instrucción RETFIE que le hará abandonar la interrupción y volver al lugar donde se encontraba antes de producirse dicha interrupción. Hemos de diferenciar entre dos tipos de interrupciones posibles en un PIC:



1. - Mediante una acción interna. El desbordamiento de la Pila (Stack) por una operación indebida, por ejemplo:

- Al completarse la escritura de datos en una EEPROM.
- Por desbordamiento del registro TMR0 al rebasar el valor 255 (FFh) a 0.

2. - Mediante una acción externa, la más útil. Al producirse un cambio del nivel en uno de sus pines por una acción externa.

- Estando en el modo de reposo (SLEEP), un cambio de nivel en el pin RB0/INT .
- Un cambio de nivel en uno de los pines RB4 a RB7 estando configurados como entrada.

Una interrupción es un evento que hace que el microcontrolador deje de ejecutar la tarea que está realizando para atender dicho acontecimiento y luego regrese y continúe la tarea que estaba realizando antes de que se presentara la interrupción. El PIC 16F628 (y el 16F628A) tiene 10 fuentes de interrupción, si las interrupciones están habilitadas cada vez que una de estos acontecimientos se presente el PIC dejará de ejecutar el programa para ir a atender la interrupción y al término de la misma continuará ejecutando el programa donde lo había dejado. Las fuentes de interrupción son:

- o Interrupción externa RB0/INT
- o Interrupción por cambio lógico en el puerto B (pines RB7 a RB4)
- o Interrupción por desborde del timer 0 (TMR0)
- o Interrupción por desborde del timer 1 (TMR1)
- o Interrupción por comparación exitosa en TMR2
- o Interrupción del comparador
- o Interrupción del transmisor del USART
- o Interrupción del receptor del USART
- o Interrupción del módulo CCP
- o Interrupción del EEPROM

Cuando ocurre un evento de los descritos anteriormente, se produce una petición de interrupción, guardando el valor actual del PC (contador de programa) en la Pila, sea cual sea la fuente de la interrupción, se pone a cero el bit7 GIE (Global Interrupt Enable), con lo cual inhibe cualquier otra petición de interrupción, el registro PC se carga con el valor 0004h que, es la posición del vector de interrupción. Aquí, empieza la ejecución del programa de atención a la interrupción ISR (Rutina de Servicio de Interrupción). El tiempo de procesamiento de la ISR debe ser lo más breve posible, para que se ejecuten las otras interrupciones ya que, pueden habilitarse más de una de ellas. Además, cualquier tipo de interrupción también puede sacar al micro del modo de reposo (SLEEP).

Una interrupción puede ser inhibida solo si existe otra interrupción en curso. Esto se debe a que, una interrupción está controlada por dos bits que indican la fuente de la interrupción, un bit actúa como bandera (flag) indicando si se ha producido una interrupción y el otro bit, actúa como bit de inhibición o prohibición de la interrupción en su, debido a que existe otra interrupción en ejecución y todo esto se realiza de forma automática por parte del micro.

Es decir, el bit GIE es el responsable del permiso de interrupción que se borra automáticamente cuando se acepta una interrupción evitando así que se produzca ninguna otra interrupción mientras se atiende a la primera. Estos bits de control se encuentran en el registro INTCON (0Bh y 8Bh). Estos bits corresponden al registro INTCON que cambia de nivel 0 a 1 cuando se produce la interrupción, excepto el último bit (bandera) que se encuentra en el registro EECON1.

Los registros asociados con las interrupciones son el registro de control de interrupción INTCON, el registro habilitación de interrupciones de periféricos PIE1 y el registro de interrupciones de periféricos PIR1. En el registro INTCON se encuentra el bit de habilitación global de interrupciones GIE, el bit de habilitación de interrupción por periféricos PEIE y los bits de habilitación de algunas interrupciones como la interrupción externa del pin RB0 (INTE), la interrupción por cambio de estado en los pines RB4 a RB7 (RBIE) y la interrupción por desborde del timer 0 (TOIE), así como las banderas correspondientes a cada interrupción (INTF, RBIF y TOIF). En el registro PIE1 se encuentran los bits de habilitación de las demás interrupciones y en el registro PIR1 se encuentran las banderas asociadas con cada interrupción.

Para habilitar las interrupciones se deben seguir los siguientes pasos:

- Habilitar el bit correspondiente a cada interrupción.
- Limpiar la bandera correspondiente a la interrupción habilitada para evitar falsas interrupciones.
- En caso de ser necesario habilitar el bit PEIE del registro INTCON (necesario para todas las interrupciones con excepción de INTE y RBIE).
- Habilitar el bit de habilitación global de interrupciones GIE del registro INTCON.

Aunque el pic cuenta con 10 fuentes distintas de interrupción solamente tiene un vector de interrupción por lo que si se habilitan varias interrupciones al momento de presentarse cualquiera de ellas el programa saltara a la misma rutina de interrupcion y es responsabilidad del programador crear una rutina que identifique la fuente de la interrupcion.

Código.

```
.include "m8535def.inc"
```

```
.def d= r16
```

```
.org $000
```

```
rjmp main
```

```
.org INT0addr
```

```
rjmp cgrg2
```

```
.org INT1addr
```

```
rjmp cgrg3
```

```
main:
```

```
ldi d, $40
```

```
mov r0, d
```

```
mov r5, d
```

```
ldi d, $76
```

```
mov r4, d
```

```
ldi d, $5c
```

```
mov r3, d
```

```
ldi d, $30
```

```
mov r2, d
```

```
ldi d, $5f
```

```
mov r1, d
```

```
I: ldi d, low (RAMEND)
```

```
out spl, d
```

```
ldi d, high(RAMEND)
```

```
out sph, d
```

```
ser d
```

```
out ddrc, d
```

```
out ddra, d
```

```
out portd,d
```

```
ldi d, $0F
```

```
out MCUCR, d
```

```
ldi d, $C0
```

```
out GICR, d
```

```
sei
```

```
clr zh
```

```
barrer: ldi zl, 5
```

```
ldi d, $20
```

```
sigue:
```

```
out portc, d
```

```
ld r17,z
```

```
out porta,r17
```

```
rcall delay
```

```
out porta, zh
```

```
in d, pinc  
lsr d  
dec zl  
brpl sigue  
rjmp barrer
```

cgrg2:

```
ldi d, $40  
mov r0, d  
mov r5, d  
ldi d, $76  
mov r1, d  
ldi d, $5c  
mov r2, d  
ldi d, $30  
mov r3, d  
ldi d, $5f  
mov r4, d  
reti
```

cgrg3:

```
ldi d, $3f  
mov r5, d  
ldi d, $38  
mov r4, d  
ldi d, $38  
mov r3, d
```



```
ldi d, $79
mov r2, d
ldi d, $76
mov r1, d
ldi d, $40
reti
```

delay:

```
ldi r17, $0B
```

WGLOOP0:

```
ldi r18, $D3
```

WGLOOP1:

```
dec r18
```

```
brne WGLOOP1
```

```
dec r17
```

```
brne WGLOOP0
```

```
ldi r17, $01
```

WGLOOP2:

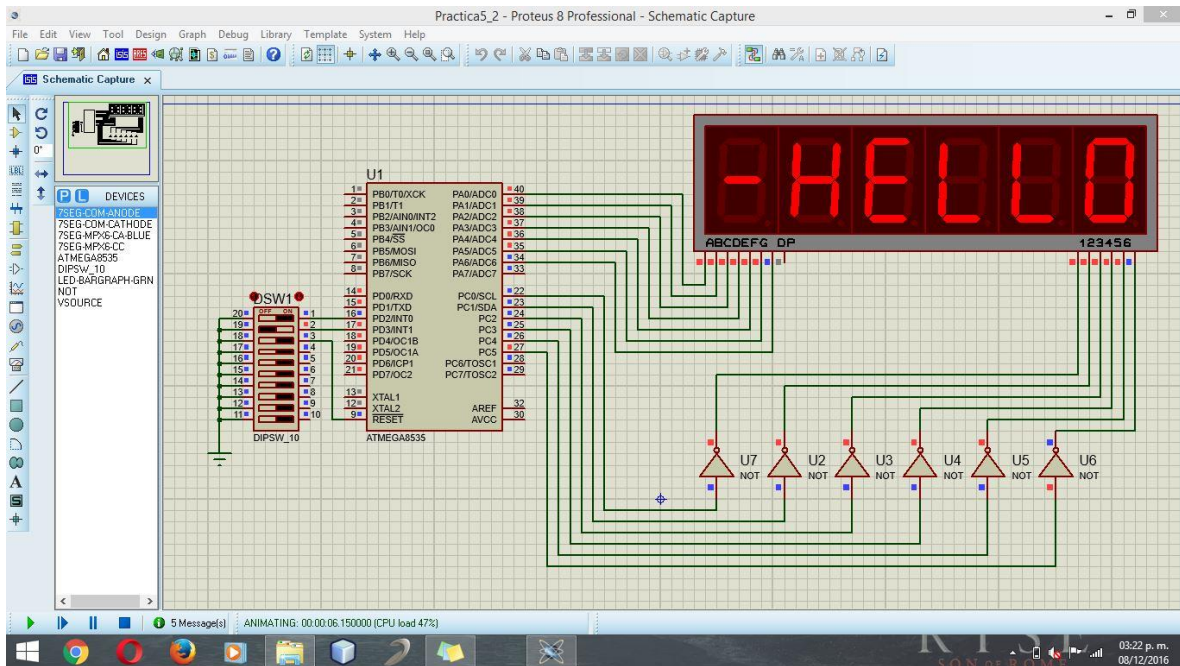
```
dec r17
```

```
brne WGLOOP2
```

```
nop
```

```
ret
```

Simulación.



Conclusión.

Interrupción es una señal recibida por el procesador de un ordenador, indicando que debe "interrumpir" el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación

El mecanismo de *interrupciones* fue la solución que permitió al procesador desentenderse de esta problemática, y delegar en el dispositivo periférico la subrutina (rutina de tratamiento). Este inconveniente se introdujo el sistema de interrupciones en. Podemos, pues, ver una interrupción como un salto.

Referencias.

https://es.wikipedia.org/wiki/Controlador_programable_de_interrupciones

www.sites.upiicsa.ipn.mx/polilibros/portal/polilibros/p...V/Unidad%20V_2.htm

www.fdi.ucm.es/profesor/jjruiz/web2/temas/Curso05_06/EC9.pdf

<http://unicrom.com/display-de-7-segmentos/>

https://es.wikipedia.org/wiki/Visualizador_de_siete_segmentos

http://www.sc.ehu.es/sbweb/webcentro/automatica/web_avr/archivos/Otros%20AVRs/ATmega/ATmega8535.htm