

PRACTICA N° 7

TEMPORIZADORES POR COMPARACION

DE UN MICROCONTROLADOR ATMEGA328P

OBJETIVOS.

HABILITAR Y USAR LOS TEMPORIZADORES 0 Y 2 MEDIANTE COMPARACION

INTRODUCCION

La terminal PD6 se encuentra en el puerto D esta terminal al igual que la mayoría de las terminales del microcontrolador tienen una función alterna que en este caso su función alterna para esta terminal es la OC0A (Output Compare o Comparación de salida), dicha función alterna sirve para generar señales digitales de salida en dicha terminal, en el caso donde los bits COM0A1 y COM0A0 del registro TCR0 estén en cero la función alternativa esta deshabilitada.

Cada vez que se haya dado una comparación y su resultado sea cero debido a que los registros TCNT0 y OCR0 se igualan da una comparación de salida, esto origina que se levante la bandera correspondiente en el registro TIFR0 y se baje cuando la interrupción es atendida, cada vez que haya una comparación salida y dependiendo de la configuración de los bits COM0A1 y COM0A0, la terminal OC0A tiene un cambio el estado lógico (toggle),

TABLA 6.1 Modos de Comparación de la Salida

COM0A1	COM0A0	Descripción
0	0	Operación Normal del Puerto (OC0 desconectado)
0	1	Cambio Lógico de OC0 (al igualar en la comparación)
1	0	Pone en cero lógico OC0 (al igualar en la comparación)
1	1	Pone en uno lógico OC0 (al igualar en la comparación)

TABLA 6.2 Modos de la generación de la forma de onda

WGM02	WGM01	WGM00	Descripción
0	0	0	Normal
0	0	1	PWM Fase Correcta
0	1	0	Limpia temporizador tras la comparación (CTC) OCRA
0	1	1	PWM Rápido
1	0	0	Reservado
1	0	1	PWM Fase Correcta OCRA
1	1	0	Reservado
1	1	1	PWM Rápido OCRA

En la figura 6.1 se ilustra la salida por comparación (OCn), la bandera de interrupción por salida por comparación, el registro TCTn empieza a contar desde 0 y se va comparando con el valor del registro OCRn, al igualarse estos dos registros, esto origina que el registro TCTn se limpie y a su vez origina una conmutación en OCn. En este modo se puede cambiar el periodo de la señal y también ayuda a la operación de conteo de eventos externos.

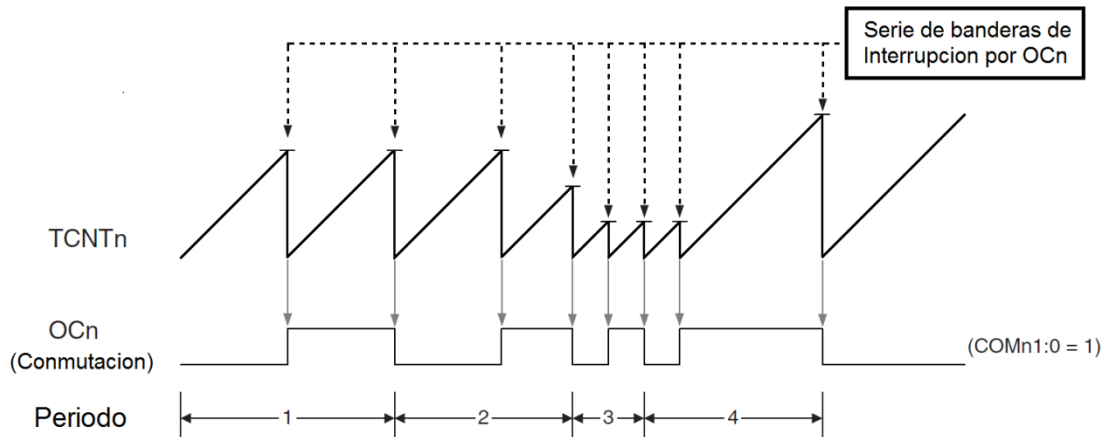


Figura 6.1 Modo de Limpia temporizador al igualar en la comparación (CTC)

Para generar una forma de onda en el modo CTC, el valor del pin del puerto OCO debe estar puesto como salida, y ajustar los bits COM00 y COM01 para forma de conmutación (Toggle) o cualquier forma que se quiera según la tabla 6.1. La frecuencia de la forma de onda se defina por la siguiente ecuación:

$$f_{ocn} = \frac{f_{clk\ i/o}}{2 * N * (1 + OCRn)}$$

La variable N representa el factor de pre-escala (1, 8, 64, 256 o 1024).

Registro e Control del Temporizador/Contador (TCCR0A)

BIT							
7	6	5	4	3	2	1	0
COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00

Los Bits WGM00, WGM01 y WGM002 (*Waveform Generation Mode*) se emplean para definir la generación de la forma de onda que será usada, esta dependerá de los modos a usarse, los cuales son CTC y dos tipos de PWM (ver tabla 6.2). El modo normal se indica en la tabla para definir el uso normal del puerto.

Los Bits COM0A0 y COM0A1 (*Compare Match Output Mode*) control el comportamiento del bit de comparación de salida OC0A si uno ambos están en uno la salida OC0A se invalida la función normal del puerto, sin embargo el bit correspondiente al DDRD debiera de estar habilitado como salida.

Registro e Control del Temporizador/Contador (TCCR0B)

BIT							
7	6	5	4	3	2	1	0
FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00

Registro de la Máscara de Interrupción del Temporizador/Contador (TIMSK)

BIT							
7	6	5	4	3	2	1	0
-	-	-	-	-	OCIE0B	OCIE0A	TOIE0

Los bits 2 y 1 **OCIE0B** y **OCIE0A**, es el bit de habilitación de la interrupción del temporizador 0 por comparación, cuando este bit esta en uno y junto con el bit de interrupción general en registro de estado también esta en uno la interrupción por comparación del Temporizador 0 es habilitada.

El bit 0 **TOIE0**, es el bit de habilitación de la interrupción del temporizador 0 por desbordamiento, cuando este bit esta en uno y junto con el bit de interrupción general en registro de estado también esta en uno la interrupción por desbordamiento del Temporizador 0 es habilitada. Es decir si ocurre un desbordamiento en el registró TCNT0 del temporizador 0 se ejecutara la interrupción.

Registro Temporizador/Contador (TCNT0)

BIT							
7	6	5	4	3	2	1	0

Este registro sirve para llevar la cuenta del temporizado, al ser un registro de 8 bits cuenta desde 0 hasta 255 antes de desbordarse y llevar el registro de nueva cuenta a ceros, el cambio de un numero a otro dentro del registro dependerá de el preescalador y el frecuencia del reloj.

Registro de Salida de Comparación (OCR0A)

BIT							
7	6	5	4	3	2	1	0

Este registro sirve para la comparación con el registro TCNT0, aquí se pone el numero con el cual se quiere comparar.

Registro de Banderas de Interrupción del Temporizador/Contador (TIFR0)

BIT							
7	6	5	4	3	2	1	0
-	-	-	-	-	OCF0B	OCF0A	TOV0

Los bits OCF0B, OCF0A y TOV0, se ponen a uno cuando ocurre el desbordamiento del temporizador correspondiente, estas bandera se limpian por hardware cuando se ejecuta la interrupción correspondiente.

MATERIAL Y EQUIPO

1 Protoboard
1 C.I. AVR ATMEGA 16
8 LEDs
8 resistencias de 330Ω
Fuente de Alimentación Regulada de 5 volts
Cables para conexión rápida
1 Multímetro

DESARROLLO

Planteamiento del problema

Se requiere escribir un programa en el cual emplee el modo CTC, al principio se introduzca en el registro OCR0 la cantidad de 125 y después de cada 10 interrupciones por comparación incremente la cantidad 5, hasta llegar a 255 para así poder reiniciar con 125 en OCR0.

Procedimiento

1. Realizar el diagrama de flujo.
2. Incluir las bibliotecas necesarias
3. Desactivar la interrupción global
4. Definir la pre escala
5. Habilitar la interrupción por comparación del temporizador 0
6. Habilitar la interrupcion global
7. En la función de interrupción sacar a puerto la variable que incrementa
8. Incrementar la variable

Diagrama del circuito

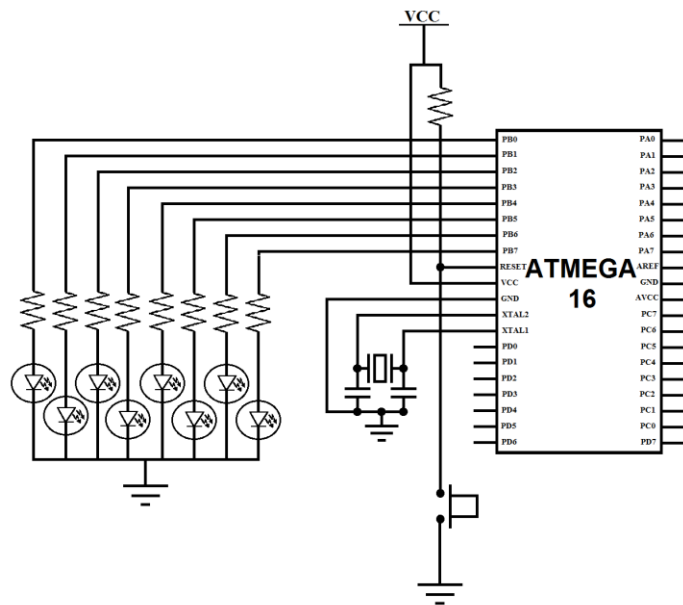


Figura 6.1 Diagrama del circuito para la practica 6.

CODIGO EN LENGUAJE C.

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/cpufunc.h>
int numero=128;

ISR(TIMER0_COMP_vect){
    PORTB = PINB;
```

```
}  
  
int main(void)  
{  
    cli();  
    DDRD |= (1<<DDD6);  
    TCCR0A = (1<<WGM01)|(0<<WGM00)|(1<<COM0A1)|(1<<COM0A0);  
    TCCR0B = (1<<CS02)|(0<<CS01)|(1<<CS00)|(0<<WGM02);  
    OCR0A = numero;  
    TIMSK0 = (1<<OCIE0A);  
    sei();  
  
    while(1)  
    {  
        // el teclado o leer ADC  
    }  
}
```

ACTIVIDADES DE TRABAJO AUTÓNOMO.

- 1) Realice el Algoritmo, diagrama de flujo, dibuje el esquemático y desarrolle el código en C para usar el temporizador 2 en comparación.
- 2) Realice el Algoritmo, diagrama de flujo, dibuje el esquemático y desarrolle el código en C para usar el temporizador 1 en comparación para 1A.
- 3) Realice el Algoritmo, diagrama de flujo, dibuje el esquemático y desarrolle el código en C para usar el temporizador 0 y 2 en comparación uno en cuenta ascendente y el otro en cuenta descendente y desplegarlo por los puertos A y B respectivamente.
- 4) Practica propuesta por el profesor en clase.