

PRACTICA N° 9

MODULADOR DE ANCHO DE PULSO RAPIDO DE UN MICROCONTROLADOR ATMEGA328P

OBJETIVOS.

HABILITAR Y USAR LOS TEMPORIZADORES MEDIANTE MODULADOR DE ANCHO DE PULSO RAPIDO

INTRODUCCION

En la practica 8 se vio el uso de los temporizadores para PWM en fase correcta, en esta práctica se empleara el modo de PWM rápido. Este modo es un PWM de una sola pendiente como se muestra en la figura 9.1

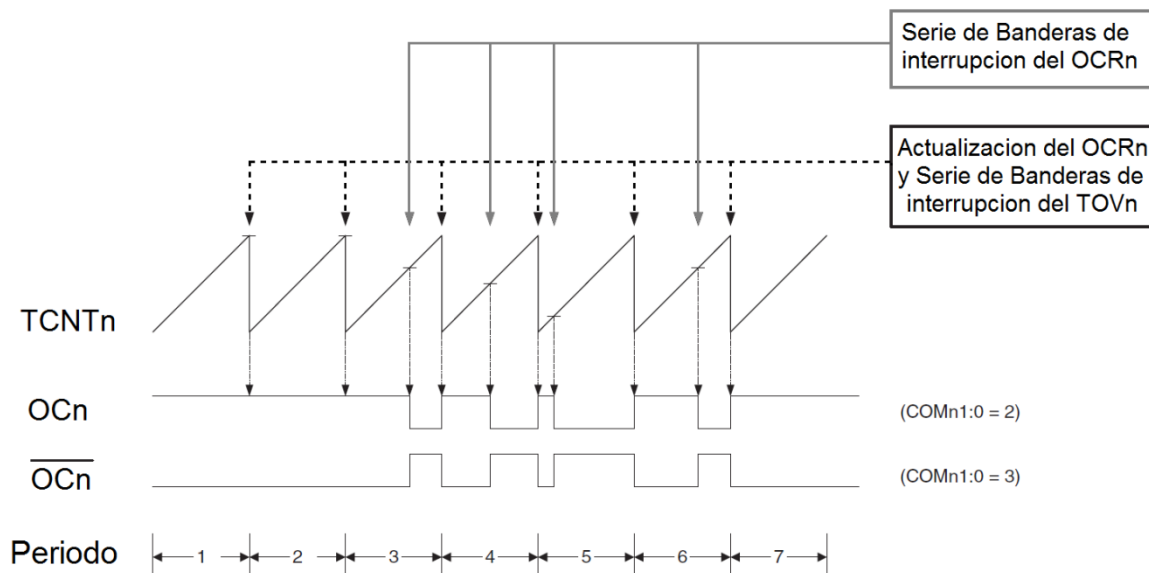


Figura 9.1 Grafica de PWM rapido

El PWM rápido maneja una frecuencia mayor comparado con el de doble pendiente, lo cual lo convierte en una buena opción para regulación de potencia, rectificación y aplicaciones con DAC, ya que su alta frecuencia permite realizar estas aplicaciones con pocos componentes externos (bobinas, capacitores, etc.) y con esto reducir el costo total de los componentes. La frecuencia de salida del PWM rápido puede ser calculada por la siguiente ecuación:

$$f_{OCnPWM} = \frac{f_{clk I/O}}{N * 256}$$

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

TABLA 9.1 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 |

TABLA 9.2 Modos de Comparación de la Salida, Modo PWM Rápido.

| COM0A1 | COM0A0 | Descripción |
|--------|--------|---|
| 0 | 0 | Operación Normal del Puerto (OC0 desconectado) |
| 0 | 1 | WGM02=0; Puerto Normal OC0A desconectado. WGM02=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) |

Los registros involucrados para el PWM en modo de fase correcta se muestran a continuación, cabe aclarar que la mayoría de los bits que intervienen se han descrito en las practicas 5, 6 y 7.

Registro e Control del Temporizador/Contador (TCCR0A)

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

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 (TIMSK0)

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

Registro Temporizador/Contador (TCNT0)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| | | | | | | | | |

Registro de Salida de Comparación (OCR0A)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| | | | | | | | | |

Registro de Salida de Comparación (OCR0B)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| | | | | | | | | |

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

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

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 se aumente o disminuir el ciclo de trabajo del PWM en modo de fase correcta mediante dos botones, el botón 1 se empleara para aumentar mediante la interrupción externa 0 y el botón 2 se empleara para disminuir mediante la interrupción externa 1.

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 desbordamiento 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

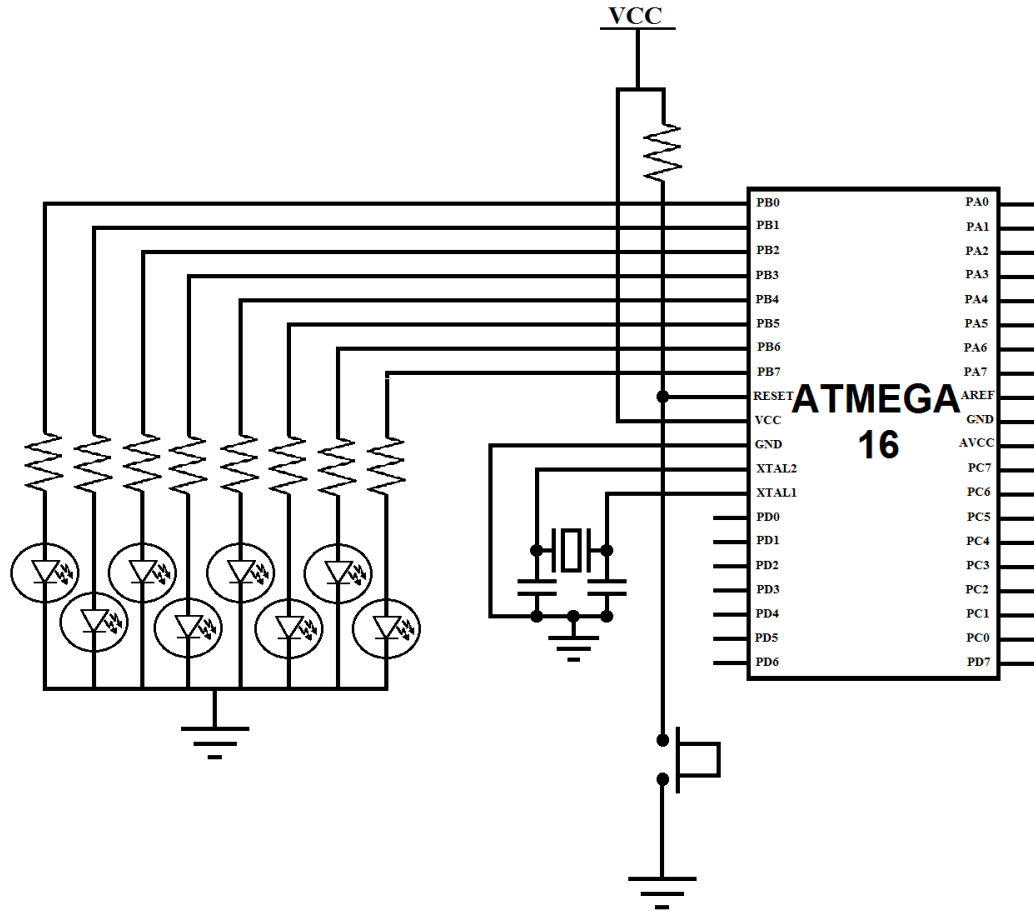


Figura 8.1 Diagrama del circuito para la practica 8.

CODIGO EN LENGUAJE C.

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <avr/cpufunc.h>

int i=250;

ISR(INT0_vect, ISR_NAKED){
    i=i+10;
    OCR0=i;
    reti();
}
ISR(INT1_vect, ISR_NAKED){
    i=i-10;
    OCR0=i;
    reti();
}
```

```
}
ISR(TIMERO_COMP_vect, ISR_NAKED){
    PORTB=PINB;
    reti();
}

int main(void)
{
    cli();
    DDRB = 0xff;
    DDRD = 0x00;
    PORTD=255;
    TCCR0 |= (1<<WGM01)|(1<<COM01)|(0<<CS02)|(1<<CS00); // CTC
    OCR0 = i;
    TIMSK |= (1<<OCIE0);

    GICR |= (1<<INT1)|(1<<INT0);
    MCUCR |= (1<<ISC11)|(1<<ISC10)|(1<<ISC01)|(1<<ISC00);

    sei();

    while(1)
    {
        _NOP();
        _NOP();
    }
}
```

ACTIVIDADES DE TRABAJO AUTÓNOMO.

- 1) Realice el Algoritmo, diagrama de flujo, dibuje el esquemático y desarrolle el código en C para usar el PWM modo fase correcta con el temporizador 2.
- 2) Realice el Algoritmo, diagrama de flujo, dibuje el esquemático y desarrolle el código en C para usar el PWM modo fase correcta con el temporizador 1.
- 3) Realice el Algoritmo, diagrama de flujo, dibuje el esquemático y desarrolle el código en C usar el PWM modo fase correcta con el temporizador 0 y comandar un servomotor de aeromodelismo.
- 4) Realice el Algoritmo, diagrama de flujo, dibuje el esquemático y desarrolle el código en C usar el PWM modo fase correcta con el temporizador 0 y comandar un motor de CD.
- 5) Practica propuesta por el profesor en clase.