



**Universidad Autónoma de Baja California
FAC. DE CS. QUIM. E INGENIERIA
INGENIERIA EN COMPUTACION**

PRACTICA 10

Laboratorio de: Microprocesadores y microcontradores

Equipo:

López Madrigal Leonardo

Maestro:

García López Jesús Adán

Tijuana, B. C.

18 Mayo, 2017

Generador de Frecuencia mediante los Temporizadores del uC ATmega1280

Objetivo: Mediante esta práctica el alumno aprenderá la programación y uso avanzado del Temporizador 0 y 2 del microcontrolador ATmega1280.

Material:

- Computadora Personal (con AVR Studio)
- Tarjeta T-Juino.
- Componentes Electrónicos.

Equipo:

- Computadora Personal con USB, AVRStudio y WinAVR

Teoría:

- Teoría Básica de Música.
- Programación del Timer2 del microcontrolador como generador de frecuencia (Diagrama, Funcionamiento, Registros de configuración y operación)

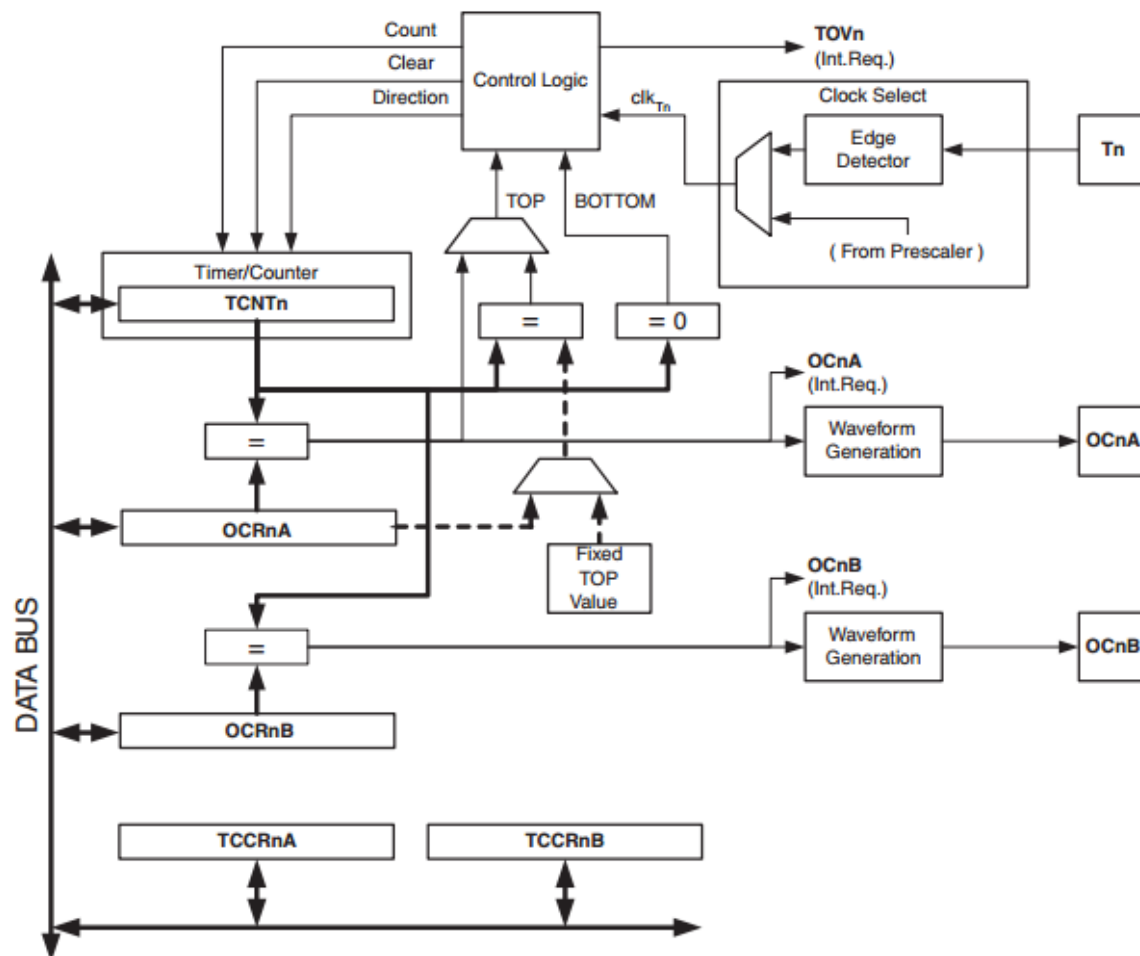
Teoría

Programación del Timer 0 del microcontrolador

El timer0 AVR es un temporizador contador de 8 bits, el registro donde se guardan los valores del timer0 AVR es el registro temporizador contador representado por TCNT0, cuando es utilizado como temporizador, sus valores aumentaran de uno en uno entre 0 y 255 con cada ciclo de reloj, por ejemplo si el oscilador con el que está funcionando el microcontrolador AVR es de 1MHz, entonces el registro TCNT0 aumentará una unidad en 1 us, si el registro TCNT0 se incrementa en 100 unidades habrán transcurrido 100us.

El registro TCNT0 del timer0 AVR puede ser leído y escrito, puede ser prescalado para que el tiempo en su incremento de valor en una unidad sea mayor, el timer0 AVR cuando alcanza su valor máximo de 255 se reinicia, volviendo a incrementar sus valores de 0 a 255, además cuando llega a su valor máximo y se reinicia puede producir una interrupción, lo cual se decide por programa.

Diagrama.



Registros

TCCR0A

TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Este registro será útil cuando se utilice el timer0 AVR en modo comparación y para la modulación de ancho de pulso PWM.

Bits para el modo de generación de la señal

Table 16-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	TOP	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Note: 1. MAX = 0xFF
2. BOTTOM = 0x00

TCCR0B

TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

El registro TCCR0B es el que permitirá utilizar el timer0 avr como contador.

Para utilizar el timer0 AVR como contador, del registro TCCR0B hay que manipular sus bits 0, 1 y 2 (CS00, CS01 y CS02) los demás bits se pondrán a 0, se tienen dos posibilidades, para que el conteo se realice cuando la señal que llega al pin T0 pase de 0 a 1 o por flanco de subida estos 3 bits se pondrán a 1 esto es CS00=1, CS01=1 y CS02=1; y si el conteo será cuando la señal que llega al pin T0 pase de 1 a 0 o por flanco de bajada solo se pondrán a 1 los bits 1 y 2, mientras el bit0 se deja a 0 esto es CS00=0, CS01=1 y CS02=1.

Table 16-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{IO}/(No \text{ prescaling})$
0	1	0	$clk_{IO}/8$ (From prescaler)
0	1	1	$clk_{IO}/64$ (From prescaler)
1	0	0	$clk_{IO}/256$ (From prescaler)
1	0	1	$clk_{IO}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge
1	1	1	External clock source on T0 pin. Clock on rising edge

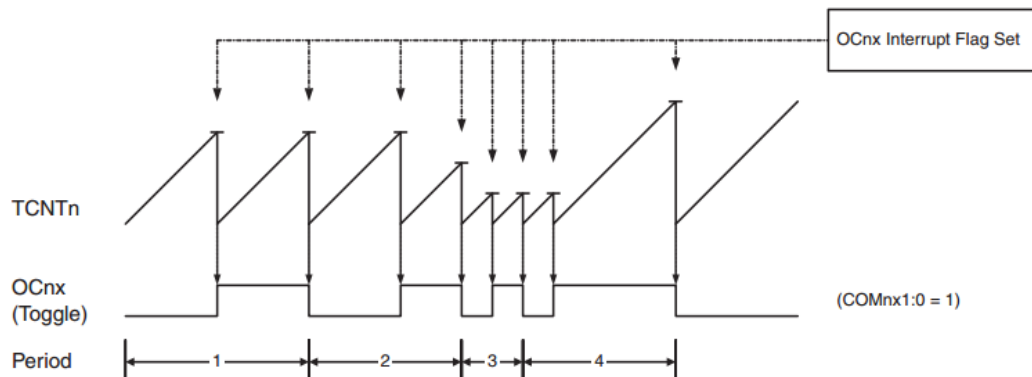
If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

Programación del Timer 2 como generador de frecuencia.

Para generar una forma de onda en modo CTC, se debe de hacer SET/Clear a la salida OC2 en cada compare match, para esto se debe poner COM2A1:0 en 1. El valor en OC2A no será visible si el pin en el que se encuentra no está de salida. La señal generada podrá tener una frecuencia máxima de $f_{OC2A} = f_{clk}/2$ cuando OCR2A está en cero. La frecuencia de la forma de onda está definida por la siguiente ecuación:

$$f_{OCnx} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

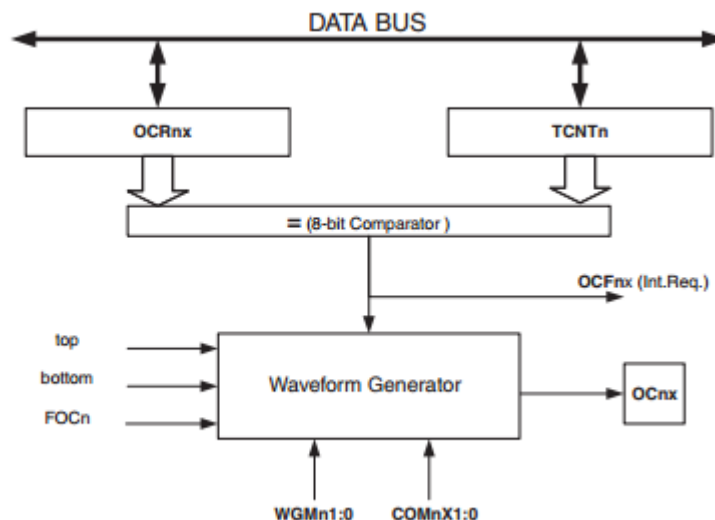
N representa el prescalador



Output compare unit

El comparador de 8-bit compara constantemente a TCNT2 con un registro de comparación de salida (OCR2A y OCR2B). Cuando TCNT2 es igual a OCR2A u OCR2B, el comparador genera una señal de coincidencia. El generador de frecuencias usa una señal de coincidencia para generar una salida de acuerdo al modo de operación puesto en WGM22:0 y COM2x1:0. Las señales de bottom y max son usadas por el generador de frecuencias para atender casos especiales donde hay valores extremos en algunos modos de operación.

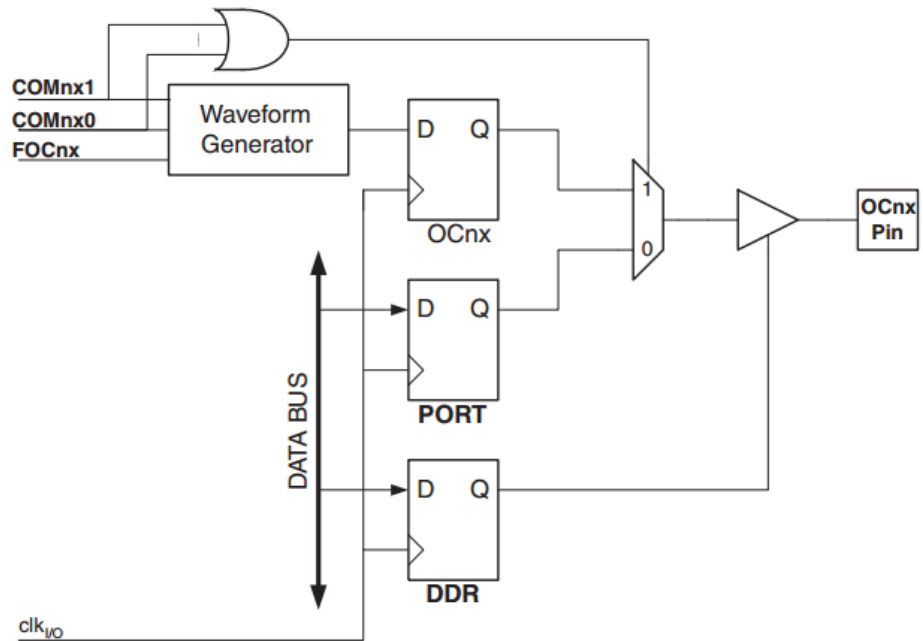
Figure 20-6. Output Compare Unit, Block Diagram



En modos de generación de frecuencia (no PWM), la salida de comparación puede ser forzada escribiéndole un 1 a FOC2x. Forzando un compare match no activará la bandera OCF2x ni limpiará ni recargará el Timer, pero el pin OC2x podrá ser actualizado si un compare match ha ocurrido realmente. (Los bits COM2x1:0 deben de ser configurados para definir el pin OC2x). Cambiar los bits

COM2x1:0 causaran cambios en el pin OC2x durante la generación de frecuencias, por lo tanto el estado de OC2x cambiara en el siguiente compare match.

Figure 20-7. Compare Match Output Unit, Schematic



Conclusiones

Esta práctica esta complicada debido a que tuve que realizar la combinación de 2 timers y utilizando PWM la cual generábamos una señal correspondiente a la nota de la marcha imperial, Primero en lo que concluyo es que la señal se puede generar de diferentes maneras, usando o no el modo FastPWM si no al modo en que mejor nos convenga e inclusive con cualquier timer, la clave para esta práctica fue **COM2B1 y COM2B0** porque al generar una señal era indicarle si era invertida o no. La otra funcionalidad solo fue el volumen el cual cambiaba con una pequeña ecuación dentro del generador de frecuencia donde al calcular los “ticks” el cual era el tiempo de una nota eso se podía reducir incluyéndolo en la formula, eso se hace con una tecla para bajar el volumen, la cual al bajarla tenia un rango y eso en la ecuación alteraba al OCR2A y se mostraba por el pin del PWM el volumen bajando o subiendo.

Bibliografía

Atmega1280 Datasheet

http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf