



Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería

Microprocesadores y Microcontroladores

Práctica 10:
"Uso de Temporizadores/contadores del uC ATmega1280"

Chávez Padilla Ignacio
1246720

Grupo : 562

Jesús García

Viernes, 10 de Mayo de 2019

El timer0 AVR cuenta con lo que se conoce como el prescaler esto hace que la frecuencia de trabajo F_{CPU} se divida por este prescaler, con lo que se logra que el timer0 AVR tarde un poco mas en aumentar su valor en una unidad; el prescaler puede tomar el valor de 1, 8, 64, 256 o 1024; estos valores se eligen programando los bits 0, 1 y 2 del registro TCCR0B, los bits 7 a 3 en este caso se pondrán a 0.

Al utilizar los prescaler se tiene la ventaja de lograr tiempos mas largos para cada incremento en una unidad del registro TCNT0 del timer0 AVR, y por lo tanto realizar temporizaciones más largas mientras el timer0 AVR incrementa sus valores.

De acuerdo a los tiempos que se quieran obtener al utilizar el timer0 AVR, habrá que utilizar el prescaler adecuado así como inicializar el registro TCNT0 también con un valor adecuado, no olvidar que este registro es de 8 bits y por lo tanto solo puede contener valores enteros entre 0 y 255.

El timer0 tiene 4 modos de funcionamiento que se pueden configurar programando sus registros asociados:

1. Modo Normal: El timer cuenta desde 0 a 255 y se desborda reiniciando la cuenta. Puede generar interrupción al desbordarse o cuando la comparación del conteo concuerde con un valor determinado.
2. Modo CTC: En este modo el timer0 es reiniciado a 0 cuando una comparación entre el timer y un valor determinado coincide. Opcionalmente puede configurarse para que al haber una coincidencia, genera una interrupción o cambie el estado de un pin.
3. Modo Fast PWM: Este modo permite generar una onda PWM de alta frecuencia. El timer cuenta desde 0 a 255 y reinicia la cuenta. Con cada cuenta el valor del timer0 se compara con un valor determinado que cuando coinciden cambia el estado de uno de los pines de salida PWM, y cuando se reinicia el timer este pin vuelve a cambiar su estado.
4. Modo Phase Correct PWM: Este modo ofrece una onda PWM de alta resolución a diferencia del modo Fast PWM. El timer cuenta hacia adelante y hacia atrás antes de hacer el cambio de estado del pin PWM, es decir cuenta de 0 a 255 al llegar a 255 cuenta de 255 a 0, obteniendo una salida PWM más limpia pero de menor frecuencia.

Programación del Timer 2 del microcontrolador

El timer2 es muy similar a su hermano timer0 sin embargo a pesar de ser tanta similares en muchos sentidos, el timer2 presenta algunas diferencias.

este temporizador no funciona como contador, solo como temporizador, y puede ser usado en modo comparador, y también en modo PWM. El timer2 como temporizador tiene dos opciones, el timer2 AVR puede trabajar con el mismo cristal o reloj con el cual trabaja el microcontrolador, o puede trabajar con un cristal o reloj independiente conectado exteriormente a los pines TOSC1 y TOSC2, en este caso se dice que trabaja en forma asíncrona al sistema.

Conclusiones y Comentarios

Durante el desarrollo de esta practica se practicó lo visto en clase, haciendo uso del timer0 y timer2, este ultimo se configuro para que funcionara con un cristal externo, esto con el fin de poder tener un tiempo base de 1ms mucho más exacto.

Es importante recordar que ambos timer cuentan con registros de 8 bits, por lo cual en ocasiones no es posible generar el tiempo base deseado, sin embargo la mayoría de las ocasiones es posible tener una buena aproximación al tiempo deseado haciendo uso del preescalador.

Bibliografía

Timer2 AVR temporizador - MICROCONTROLADORES. (2019). Retrieved from <http://microcontroladores-mrelberni.com/timer2-avr-temporizador/>

alfreedom, V. (2019). AVR programación en C – 09 Timer/Counter0 del ATmega16/32 – Parte 1 (Modo Normal). Retrieved from <https://vidaembebida.wordpress.com/2014/08/17/avr-programacion-en-c-09-timercounter0-del-atmega16-parte1/>