



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



INTRODUCCIÓN A LOS MICROCONTROLADORES

PRACTICA #6

Integrantes:

- ✓ *Lomelí García Martín*
- ✓ *Pacchiano Alemán Alain*
- ✓ *Trejo Martínez Francisco*

PROFESOR: Pérez Pérez José Juan

GRUPO: 3CM3

22/11/2016

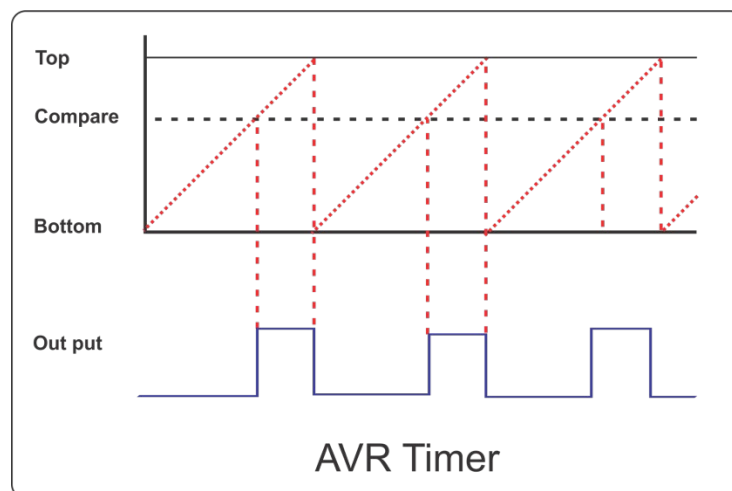
MARCO TEÓRICO

Timers

Los timers o temporizadores son características estándar de casi todos los microcontroladores, así que es muy importante aprender su uso. Los microcontrolador AVR tiene temporizadores muy poderosos y multifuncionales, pueden medir tiempo, generar frecuencias, contar eventos externos, hacer comparaciones, sincronizar tareas, generar una onda PWM, etc.

El timer0 del ATmega16 es un contador/temporizador de 8 bits (puede contar hasta 255) síncrono, es decir, que depende del reloj del CPU (sólo su reloj, no el CPU) para funcionar, es un módulo independiente del núcleo del AVR lo que reduce la carga del micro.

Cada timer tiene asociado un reloj que fija el paso que debe marcar y una o más unidades comparadoras. El reloj del timer0 es derivado del reloj del CPU y puede tener dos fuentes distintas: si se usa la fuente de reloj interna del CPU, el reloj del CPU pasa primero por un divisor de frecuencia (prescaler) y la salida de este divisor va al timer, si se usa una fuente de reloj externa por el pin T0, primero será sincronizada con el reloj interno antes de pasar al prescaler. Los valores de división (prescaler) ya están establecidos para cada timer, en el caso del timer0 la frecuencia del CPU puede ser dividida por un factor de 1, 8, 64, 256 o 1024, generando así la frecuencia de conteo. Por cada unidad comparadora existe un pin asociado a ella, normalmente marcado como OCx (Output Compare x) que es donde se puede generar una frecuencia de salida por la unidad comparadora del timer.



Código fuente del programa

Esta práctica consistió en la aplicación de timers para simular un tiempo específico de ejecución o de espera dentro del programa al igual que se hizo previamente con el uso de rutinas, el programa era un contador del 0-9.

A continuación se muestra el código implementado para este programa.

```
.include "m8535def.inc"
.DEF D=R16
.DEF aux=R17
.DEF outout=R18

RJMP main
.ORG $009
RJMP count

main:
    LDI D, LOW(RAMEND)
    OUT SPL, D
    LDI D, HIGH(RAMEND)
    OUT SPH, D
    SER D
    OUT DDRA, D
    LDI D, 5
    OUT TCCR0, D
    LDI D, 1
    OUT TIMSK, D
    SEI

    LDI R20, $C0
    LDI R21, $F9
    LDI R22, $A4
    LDI R23, $B0

init:
    LDI ZL, 20
    ADD ZL, aux
    LD outout, Z
    OUT PORTA, outout
    RJMP init

count:
    INC aux
    CPI aux, 10
    BREQ zero
    RETI

zero:
    CLR aux
    RETI
```

CONCLUSIONES

Lomelí García Martín: En la práctica 6 aprendimos a programar en ensamblador con el uso de timers y nos dimos cuenta que su uso era muy parecido al de las rutinas solo que a mi parecer es un poco más difícil de implementar y entender a diferencia de las rutinas.

Pacchiano Alemán Alain: La práctica solo consistió en la implementación de timers muy parecidos a la practica 3 de subrutinas.

Trejo Martínez Francisco: En esta ocasión aprendimos a implementar el uso de timers, recapitulando en la practica 3 usamos rutinas para mostrar en display una cuenta de 0-9 y en esta ocasión se hizo lo mismo pero con el uso de timers.