INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo

Introducción a los microcontroladores

PRÁCTICA: “Hola timerscroll”

Integrantes del equipo:

Cebada Velázquez Luis

Galindo García José Jorge

Martínez Estrada Adriana Leticia

Martínez Guerrero Juan De Dios

Grupo: 3CM7

INTRODUCCION

Existen diferentes formas de realizar un retardo, lo importante es entender que un retardo se logra por medio de un lazo que se repite varias veces, dentro de otro lazo que se repite otro número de veces y así sucesivamente hasta alcanzar el tiempo que necesitamos en nuestra aplicación, las formas sencillas normalmente son inexactas y pueden ser usadas cuando las aplicaciones que estamos construyendo no requieren estrictamente el tiempo que nos piden.

Por ejemplo, en nuestra primera actividad tenemos una sirena de 1seg, realmente nadie notaría la diferencia si la sirena suena 0.8 segundos o 1segundo. Pero en otros casos la exactitud es crítica, como en nuestra tercera práctica donde se quiere mostrar valores en un display de 7 segmentos o en aplicaciones de comunicaciones por ejemplo los retardos deben tener exactitud intachable.

En éstos casos utilizamos algunas técnicas que nos toman más tiempo, pero garantizan mayor precisión.

Si utilizamos de 4MHz tenemos que todas las instrucciones se ejecutan en 1uS porque cada instrucción se toma 4 ciclos de reloj:

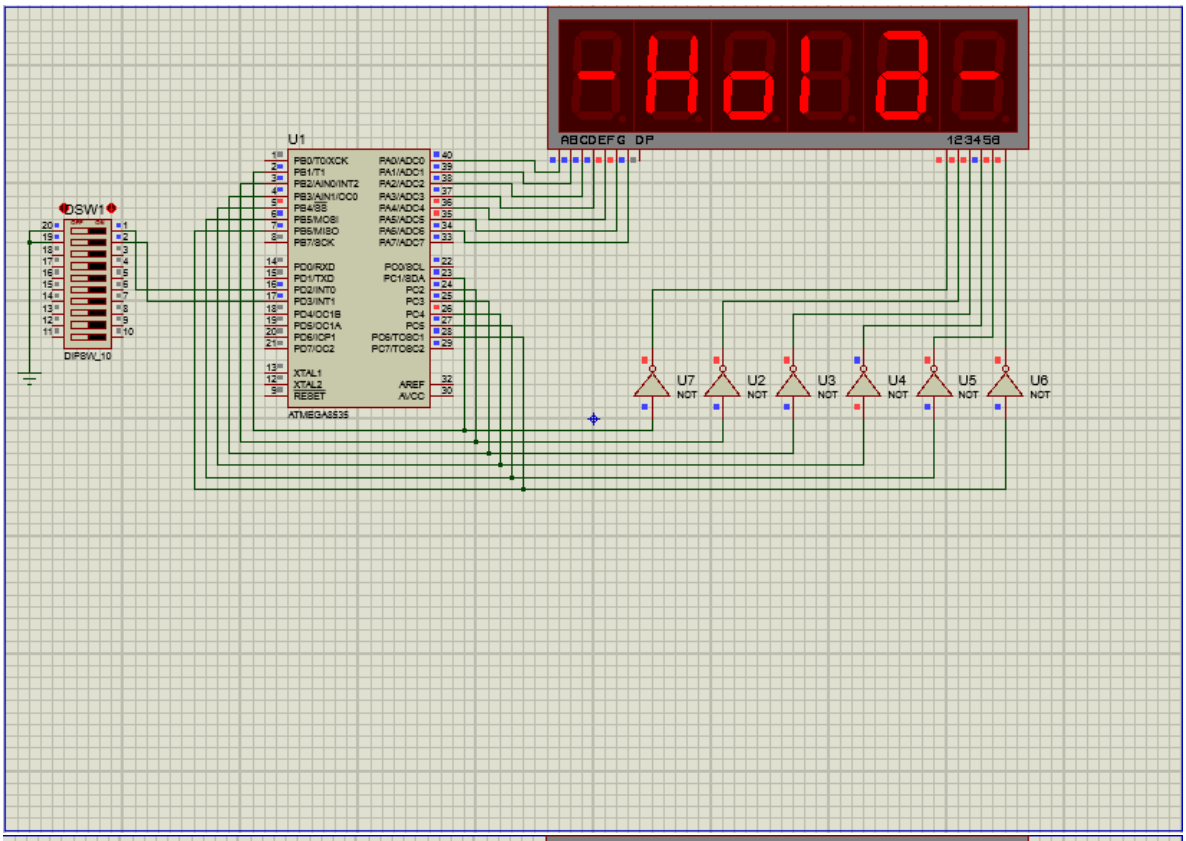
Ciclo de reloj: 1/4MHz: 0,00000025 segundos

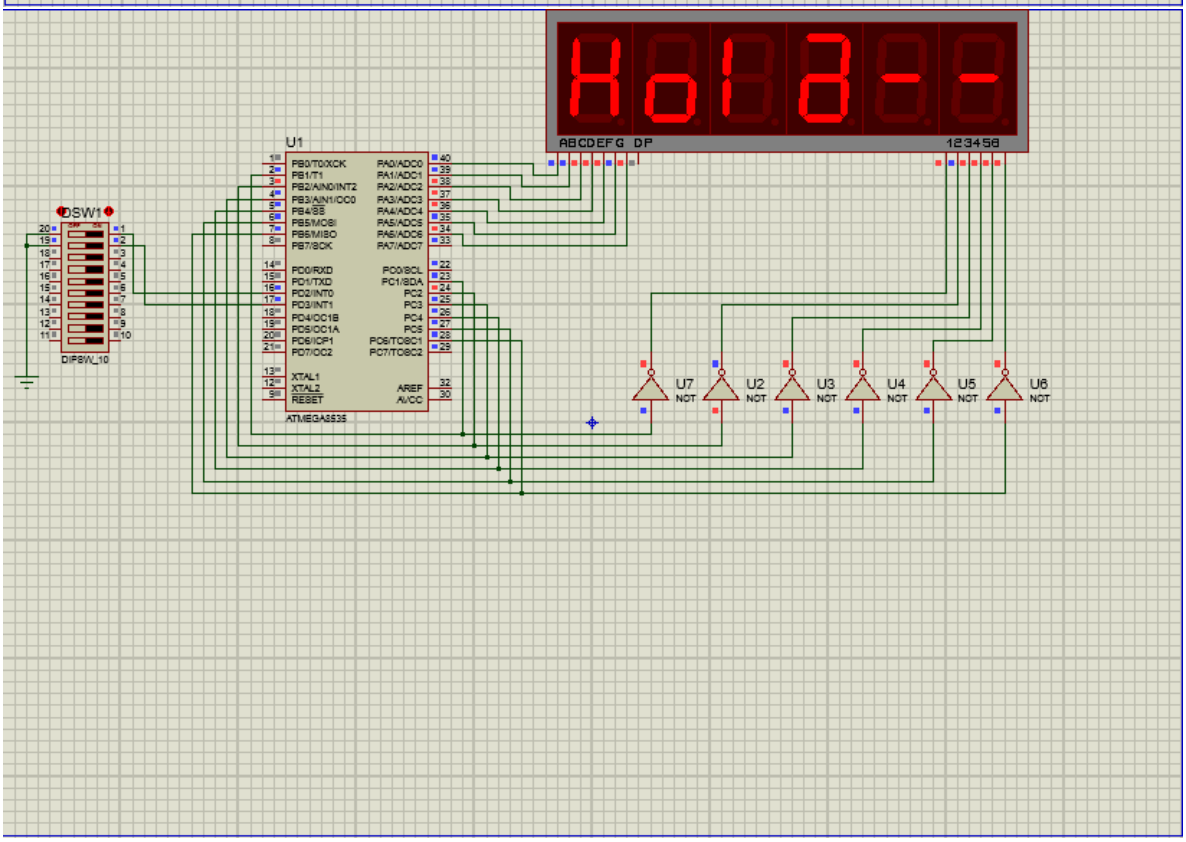
4 Ciclos de reloj: 4 x 0,00000025= 0,000001 = 1uS

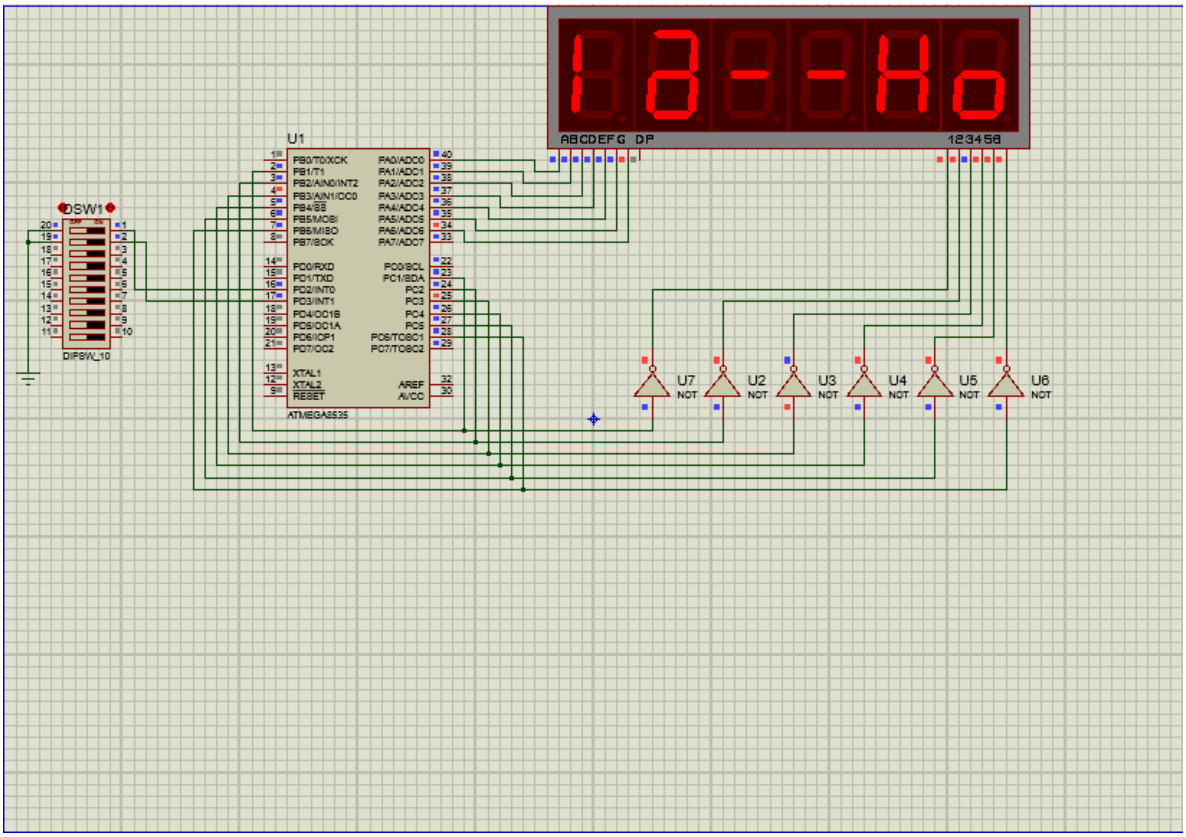
Cada línea que contenga instrucción tarda 1 micro segundo en ejecutarse.

Exceptuando las instrucciones que tiene saltos que tardan 2 ciclo, es decir 2uS. Tomando en cuenta eso vamos a generar nuestros retardos, pero siempre partiendo de la suposición de que se está trabajando a una frecuencia de 4MHz, si es así.

CÓDIGO DE LA PRÁCTICA








```
.include "m8535def.inc"
        .def aux = r16
        .def col = r17
        .def aux2 = r18
        .def aux3 = r19
        rjmp main
        .org $08
        rjmp scroll
        rjmp barre
main: ldi aux,low(RAMEND)
        out spl,aux
        ldi aux,high(RAMEND)
        out sph,aux
        ser aux
        out DDRA,aux
        out DDRC,aux
        ldi aux,$40
        mov r0,aux
        ldi aux,$76
        mov r1,aux
        ldi aux,$5c
        mov r2,aux
        ldi aux,$30
        mov r3,aux
        ldi aux,$5f
        mov r4,aux
```

```
        ldi aux,$40
        mov r5,aux
        ldi aux,1
        out TCCR0,aux
        ldi aux,2
        out TCCR1B,aux
        ldi aux,5
        out TIMSK,aux
        sei
        clr zh
        ldi zl,5
        ldi col,$40
uno: out PORTC,col
        ld aux,z
        out PORTA,aux
        rjmp uno
barre: ldi aux2,192
        out TCNT0,aux2
        out PORTA,zh
        lsr col
        brcs dos
        dec zl
        reti
dos: ldi col,$40
        ldi zl,5
        reti
scroll: mov aux3,r5
        mov r5,r4
        mov r4,r3
        mov r3,r2
        mov r2,r1
        mov r1,r0
        mov r0,aux3
        reti
```

CONCLUSIONES

Esta práctica a diferencia de la pasada es más compleja ya que no solo se tenía que buscar el mejor tiempo de los retardos si no que se tenía que recoger la palabra display a display, lo cual hizo más complejo esta práctica ya que se tuvo que buscar que es lo se podía hacer para realizar esta práctica y solucionar esta parte de la práctica, donde como primer paso se cargó los valores de la palabra que se quería mostrar posteriormente se realizaba un scroll que realizaba el intercambio de los valores de los display hacía la derecha con lo cual se da el efecto que la palabra se recorre de manera secuencial. Sin embargo, se tienen retardos para mostrar los valores en los display de manera adecuada.