

Universidad autónoma de baja California

Ingeniería en computación

Microcontroladores

Practica 3 delay por software

Erik garcia Chávez 01275973

Jesús Adán Garcia López

28 de febrero del 2025

Ecuaciones para los retardos de 103uS, 1mS y 1S:

103uS:

Ecuación:

14+ 4x + 3xyz + xy(2z-1) + 4xy + x(2y-1) + 2x + (2x-1) -> 7x + 5xy + 5xyz+ 13 = 1648

* x(7+ 5y + 5yz) + 13 = 1648, donde X = 5, Y = 8 y Z =7

7(5 ) + 5(5\*8) + 5 (5 \* 8 \* 7) = 1648//

Formulación de la ecuación:

retardo\_103uS:

;4 ciclos rcall

clr r20 ;1

clr r21 ; 1

clr r22 ;1

ldi r22,5 ;-> 1 -> x

ldi r21, 7

;--------

;5 ciclos total

nxt0:

nop ; ->1x

nop; 1x

nop ;1x

ldi r20, 8 ;1x

;-------

;4x

nxt1:

dec r21 ;1zyx

nop ;1zyx

nop ;1xyz

brne nxt1 ; xy(2z-1)

;----------

;3xyz + xy(2z-1)

nxt2:

ldi r21,7 ;-> 1xy

nop ; 1xy

nop ;1xy

dec r20 ;-> 1xy

brne nxt1 ; x(2y-1)

;---------------------

;4xy + x(2y-1)

dec r22 ;x -> 1x

nop ; 1x

brne nxt0 ;-> (2x-1)

ret ;5 ciclos

;-------------

;2x+(2x-1)+5

;--------------------------

;7x + 5xy + 5xyz + 13

Retardo 1mS:

4 + 5 + 2 + 2x+ 1xy+ 1xyz+ xy(2z-1) + 1xy + x(2y-1) + 1x+ (2x-1) ->

4x + 3xy + 3xyz + 10 = 16000 -> x(4+ 3y + 3yz)+ 10 = 16,000

x-> 15, y-> 59, z-> 5

4(15) + 3(15\*59) + 3(15\*59\*5) + 10 = 16,000

Formulación de la ecuación:

retardo\_1mS:

; rcall -> 4 ciclos

; para un retardo de 1mS es necesario 16,000 ticks

; esto porque (1x10^-3)\*(16x10^6) -> 16,000

ldi r20, 15 ; 1

nop ; 1

nxt\_mS:

; este es el ciclo superior a todos

ldi r21, 59 ; 1x

nop ;1x

nxt1\_mS

ldi r22, 5 ; 1x

nxt2\_mS:

; este en el ciclo mas pequenio. en el ciclo

dec r22 ; 1xyz

brne nxt2\_mS ; xy(2z-1

dec r21 ;1xy

brne nxt1\_mS ; x(2y-1)

dec r20 ; 1x

brne nxt\_mS ; 2x-1

ret ; final del delay

retardo de 1S:

1+ 4 + 5+ 4x+ 3xy + 1xyz + xy(2z-1) + xy + x(2y-1) + x + (2x-1)

6x + 5xy + 3xyz + 9 = 16,000, 000

X = 241, Y= 144, Z=15

6(241) = 5(241\* 144) + x(241\*144 \*152 ) + 9 = 16,000,000

Formulación de ecuación:

retardo\_1S:

ldi r20, 241; /1

nxt\_S: ldi r21, 144; /x

nop; /x

nop; /x

nop; /x

nxt2\_S:ldi r22, 152 ; /xy

nop; /xy

nop; /xy

nxt3\_S:dec r22; /xyz

brne nxt3\_S; /xy(2z-1)

dec r21; /xy

brne nxt2\_S; /x(2y-1)

dec r20; /x

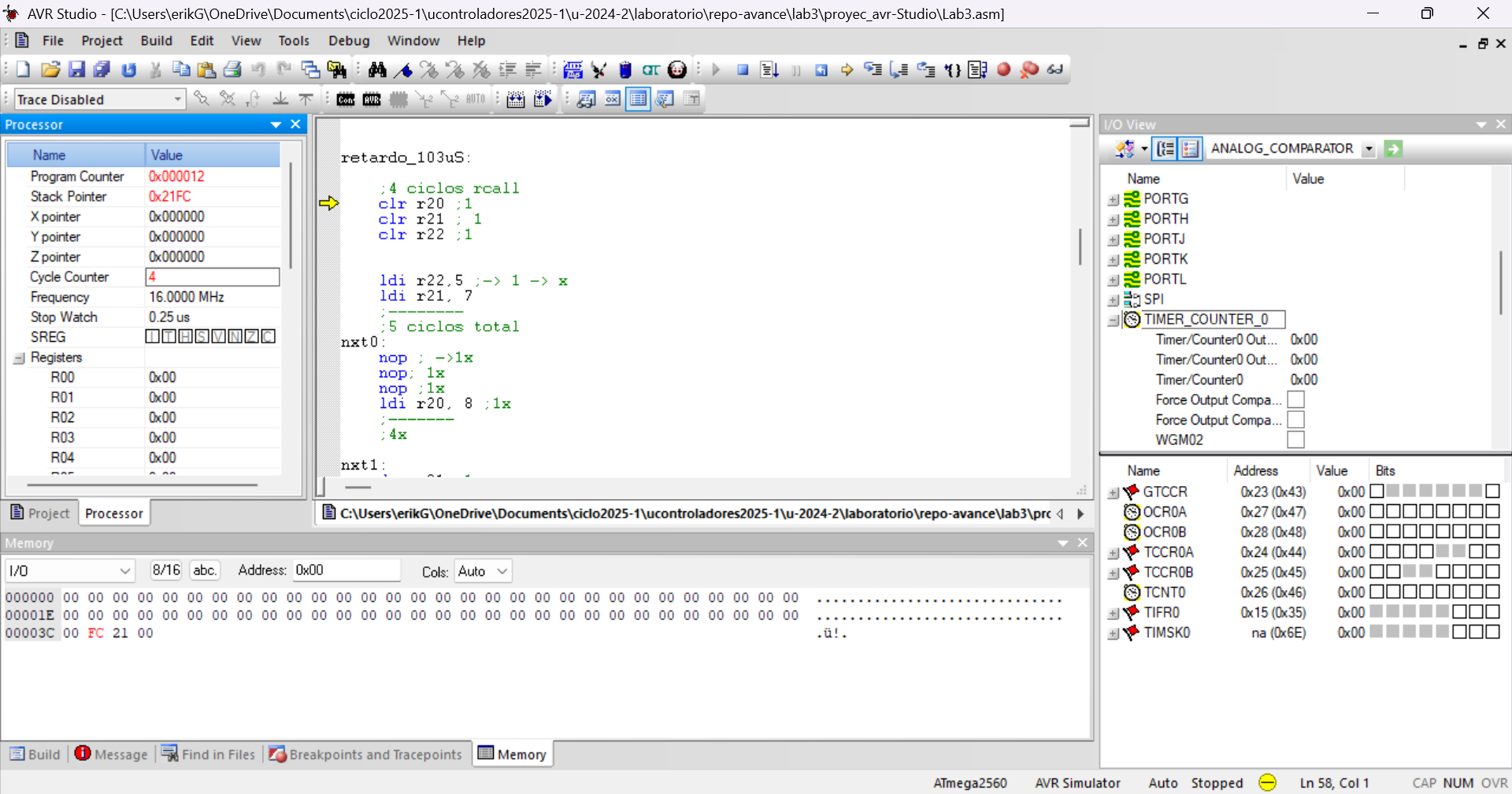
brne nxt\_S;   /2x-1

ret

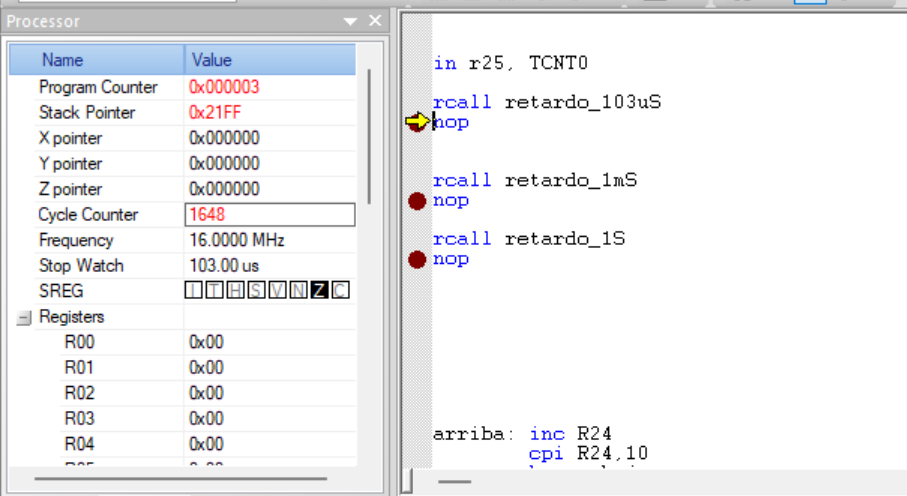
implementacion:

103uS:

Inicio, entrando a la subrutina:

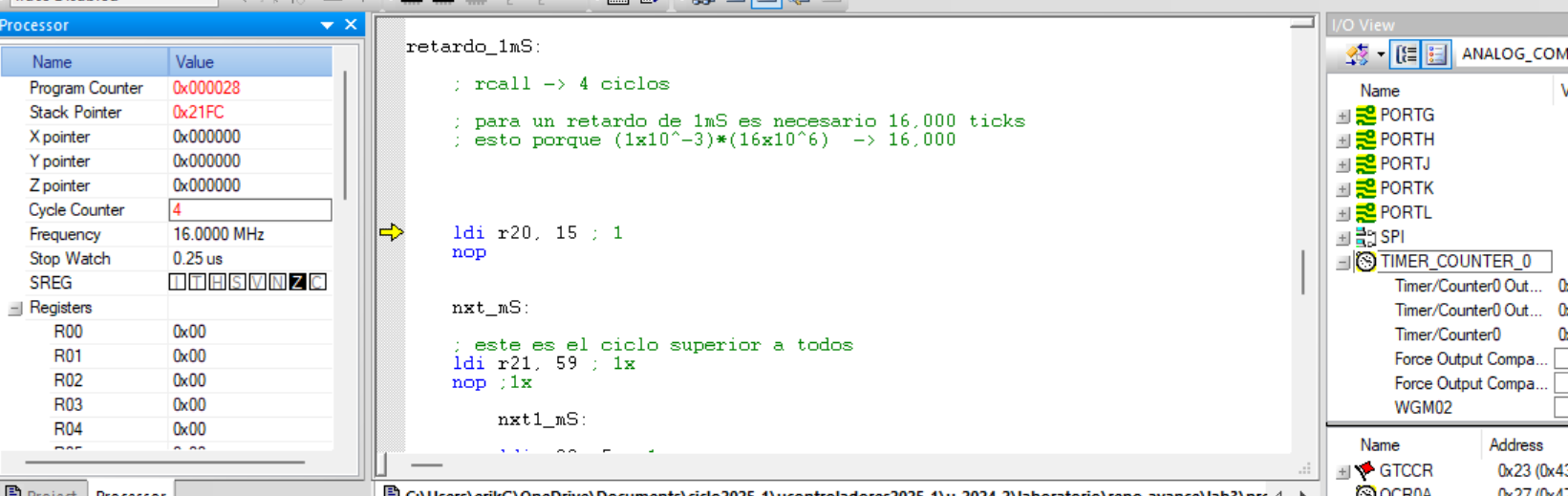


Finalización:

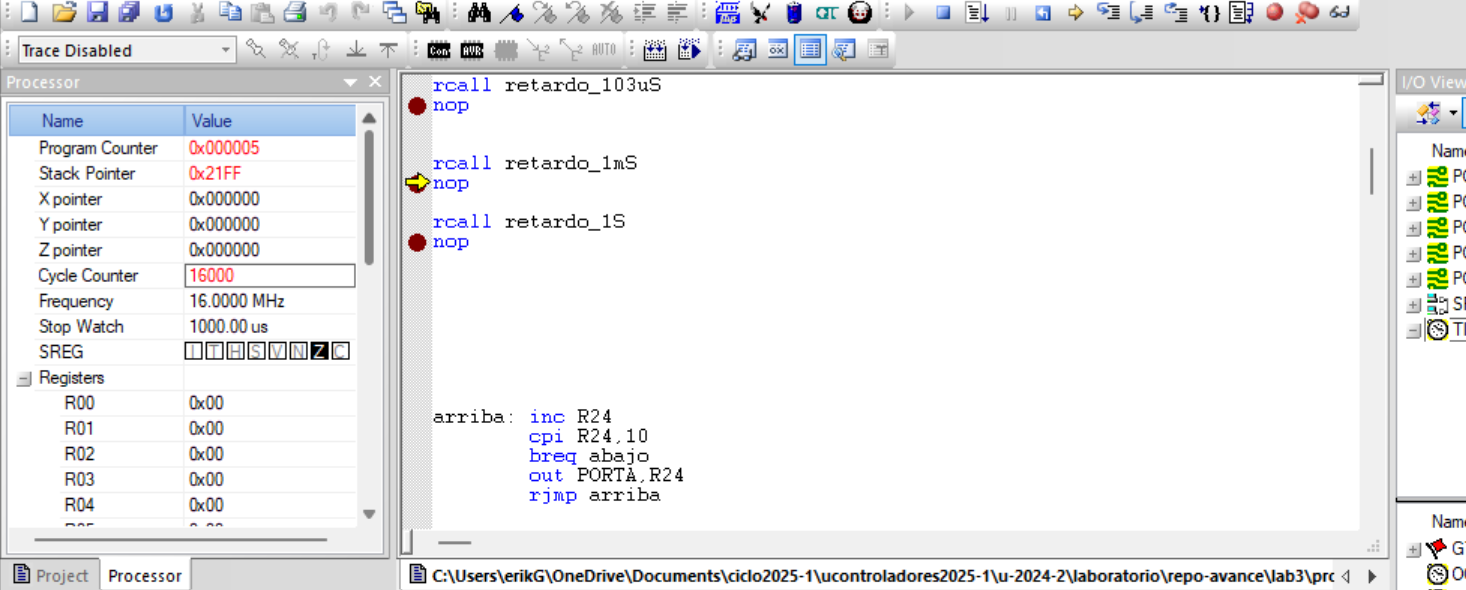


Implementación 1mS:

Inicio:

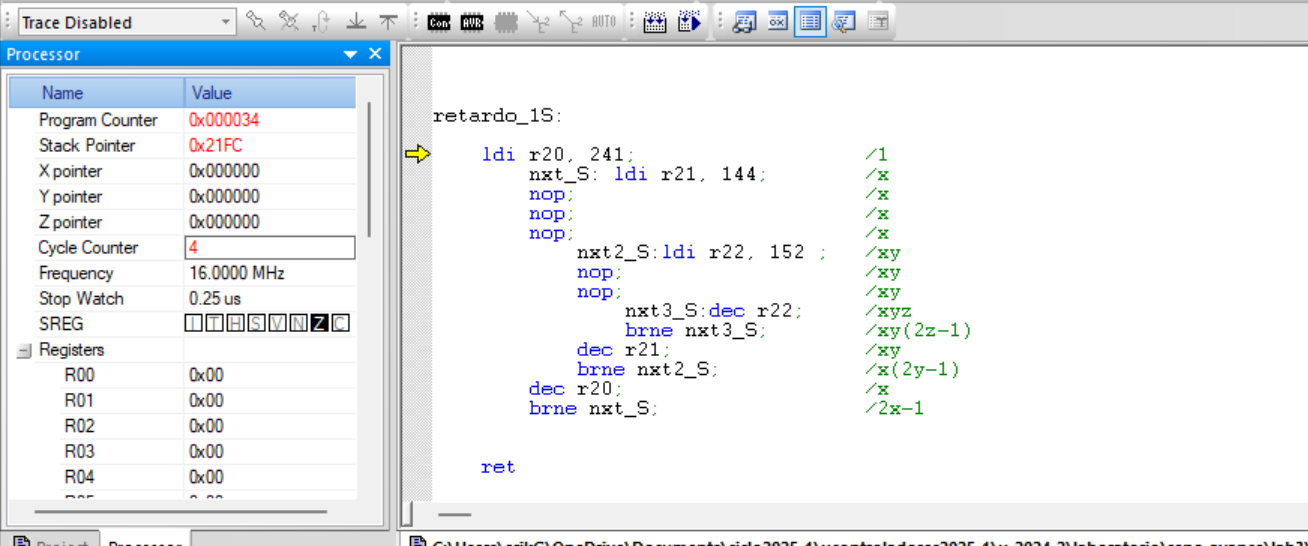


Finalizacion del delay:

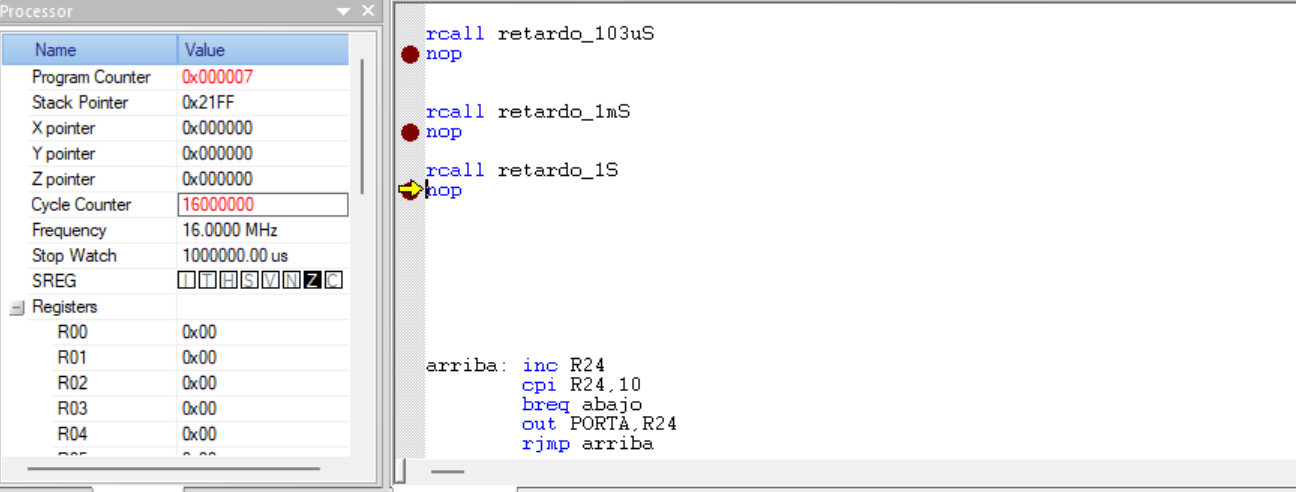


Delay de 1S:

inicializacion

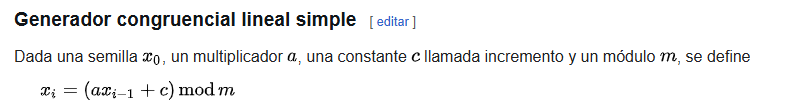


Finalización:



Generador del número aleatorio: generación congrencial lineal simple

Este genreador responde a la siguiente propiedad:



Donde:

Xn = el valor actual de la secuencia

a= el multiplicador

c= es el incremento

m = es el modulo

Xn+1= el siguiente valor de la secuencia:

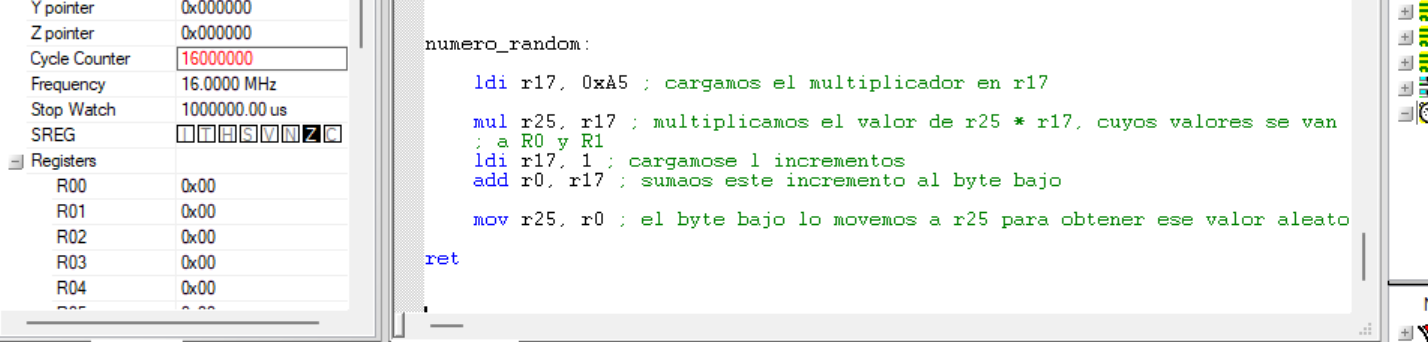
Para implementarlo en ASM para un numero de 8 bits debemos establecer los siguiente parámetros

M= 256 dado que se estará trabajando con números de 8 bits,

A y C, se establecen en 165 y 1 respectivamente dado que estos son valores que se dan para tener la mayor distribución de numero alearorios el siguiente que se puede usar es demasiado grande para ser alamcenado en un registro de 8 bits.

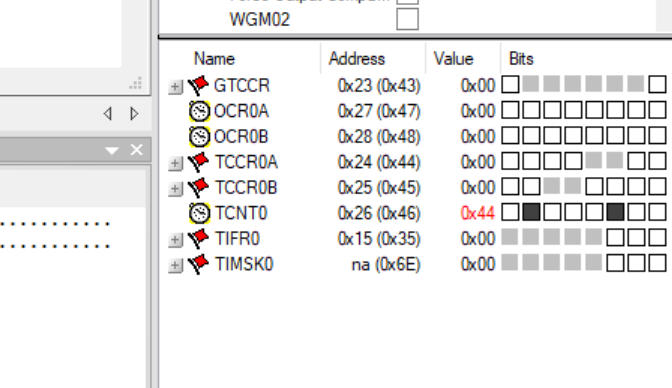
X0, este valor es la semilla, el cual puede ser obtenido de varias formar, en el caso que se implemento utilizamos el valor que hay en el timer0, cunado se simula este código es necesario introducir el valor manualmente al timer para que este pueda generar los valores aleatorios dado que en la simulación todos empiezan en 0, en físico esto no será asi porque este iniciara en un valor X.

Implementación:

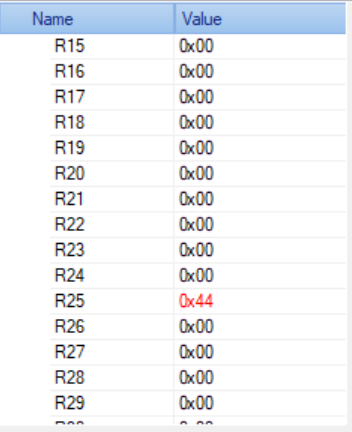


Ejemplo:

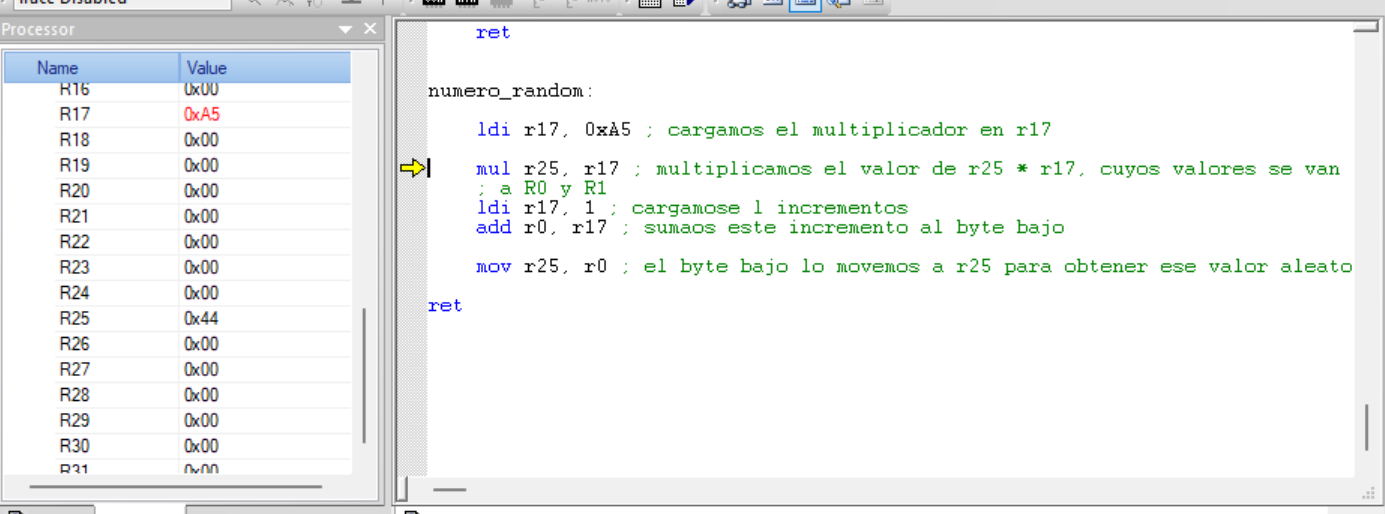
Ingresamos un valor X al timer0:

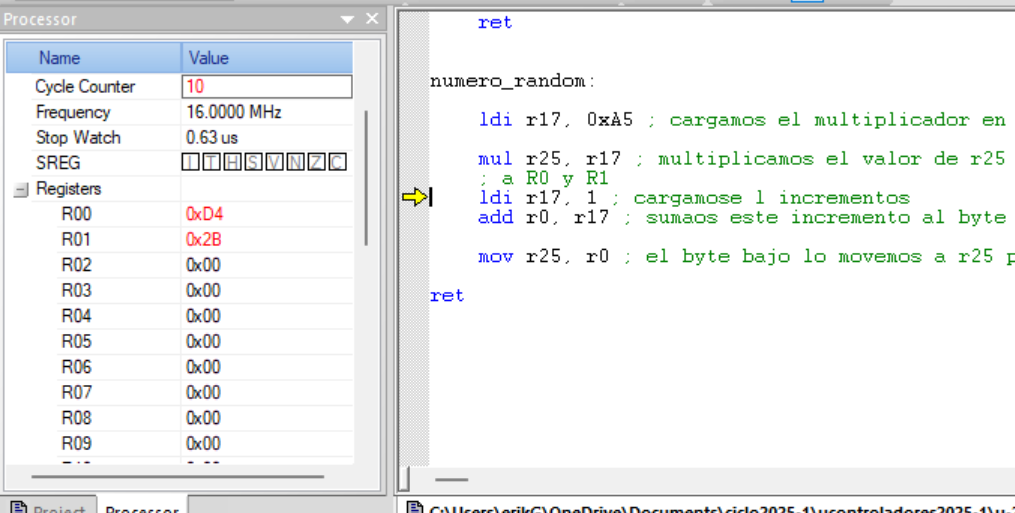


La semilla se cargo a R25:

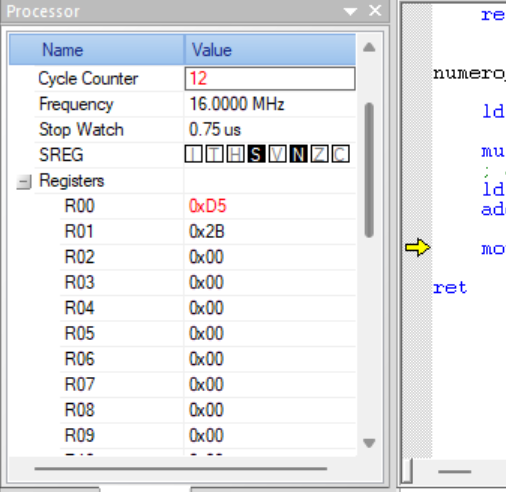


Se carga el 165(0XA5) a R17 que es nuestro multiplicador

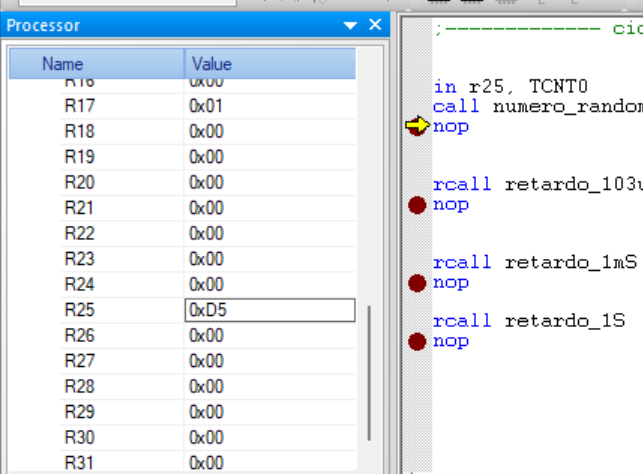


Realizamos la multiplicación: 

Le sumamos el incremento a R0 que seria la parte baja de la multiplicación:



R25 ahora tiene el numero aleatorio que se generó:



Conclusiones:

En la practica pudimos poner en practica los retardos por software, todo lo que conlleva el realizar uno de cierta cantidad de tiempo, lo cual es un ciclo que lo que hace es quemar tiempo. Ticks, para cumplir con su objetivo, asi como poder crear una subrutina que crea un pseudo numero aletatorio, hay muchas formas de hacerlo, en mi caso use la formula el cual lleva el nombre de generación congrencial lineal simple, usando como semilla el registro del timer0. Este fue el que mas se me complico dado que no sabia como representar la formula en ASM.