

USO DEL SIMULADOR EN MPLAB IDE

SIMULACIÓN DE UN PROGRAMA QUE REALIZA EL MANEJO DE LOS BITS EN EL PUERTO B Y DE UNA ENTRADA DIGITAL EN EL BIT RA1

Descripción: el software MPLAB IDE incluye un programa simulador, que es una herramienta de desarrollo sumamente útil, no solamente para localizar errores (bugs) en un programa, sino también como auxiliar didáctico. El usuario puede seguir paso por paso cada instrucción del programa, insertar breakpoints, generar entradas digitales en cualquier bit de los puertos y observar el contenido de cualquier registro interno del microcontrolador, así como de las localidades de la memoria de datos y de código, la EEPROM y el Stack.

El simulador cuenta también con la opción de una función “Stopwatch” para medir tiempos de ejecución.

Mediante este ejemplo, se explica la forma de simular un programa escrito en lenguaje ensamblador por medio del MPLAB IDE, observando el comportamiento de los bits del Puerto B, incluyendo la simulación de entradas digitales, concretamente el bit RA1.

The screenshot shows the MPLAB IDE v8.40 interface. The main window displays assembly code for a 200ms delay routine. A green arrow points to the 'goto jloop' instruction. Below the code, the 'Stimulus' window shows a digital input event for RA1 (Set High). The 'Special Function Registers' window shows PORTA at 0x02 and PORTB at 0x40. The status bar at the bottom indicates PIC16F628A, pc:0x21, W:0x2, Z dc c, 4 MHz, bank 0, and WR.

```
RETURN

;rutina de retraso de 200 milisegundos

retro: movlw  D'2'      ; w = 250 decimal
      movwf  J      ; J = w
jloop: movwf  K      ; K = w
kloop: decfsz K,f    ; K = K-1, omite si es cero
      goto  kloop
      decfsz J,f    ; J = J-1, omite si es cero
      goto  jloop
      return

END
```

Address	SFR Name	Hex
005	PORTA	0x02
006	PORTB	0x40

Fire	Pin / SFR	Action
>	RA1	Set High

1. Dentro del programa MPLAB v8.50, abra, seleccionando “File”, y “Open” el archivo “LEDSCORRIMIENTO2F88TINY.asm”. Este archivo lo encuentra en el siguiente link: www.puntoflotante.net/progensamblador.htm (nombre abreviado: ”LEDSCORR2TINY.asm”) . Este programa realiza el corrimiento hacia la derecha ó izquierda, de un bit encendido en el puerto B. La dirección del corrimiento es controlada por el estado del bit RA1, que en el sistema EVOLUPIC Bootloader 16F88 corresponde al switch SW1.
2. Modifique la constante en la subrutina de retraso “retro” y escriba D’2’ en lugar de D’250’ para escalar el tiempo y poder hacer la simulación, como se muestra en la ventana.
3. >Debugger> Select Tool> MPLAB SIM
4. >Debugger> Settings> Processor frequency= 4 Mhz
5. >Debugger> Stimulus > New Workbook
6. >Pin/SFR elija RA1
7. >Action elija “Set High”. Acomode la ventana para observar RA1, como se muestra.
8. >Project> Quickbuild> LEDSCORRIMIENTO2F88TINY.asm
9. >Debugger> Reset> Processor Reset (aparece flecha verde)
10. >View >Special Function Registers (acomode la ventana para observar PORTA Y PORTB)
11. Oprimiendo repetidamente F7 realice la simulación paso por paso y observe el corrimiento en los bits del puerto B con los valores: 01-02-04-08-10-20-40-80-01...etc. Recuerde que todos estos valores están en notación hexadecimal.
12. En cualquier instante, suspenda la opresión de F7 y seleccione con el mouse el botón de “FIRE” a la izquierda de RA1, para activar la entrada digital RA1=1
13. A la siguiente opresión de F7 aparecerá el valor PORTA=x02, es decir 00000010, (el bit RA1 se activa).
14. Continúe con la simulación oprimiendo F7 y ahora observará los valores del puerto B con el patrón de corrimiento invertido: ...80-40-20-10-08-02...etc.
15. Modifique en la ventana “Stimulus”, “Action”, “Set Low” y, junto a la ventanita de RA1, oprima “FIRE” y observe que al oprimir de nuevo F7, el valor PORTA=0x00, desactivando el bit RA1=0
16. Cuando termine de hacer la simulación, modifique de nuevo el valor en la subrutina “retro” a D’250’
17. >Project> Quickbuild> LEDSCORRIMIENTO2F88TINY.asm
18. Mediante el software TINY, cargue el programa ejecutable .hex en el sistema EVOLUPIC Bootloader 16F88 y observe el programa funcionando en tiempo real. Controle el corrimiento mediante el switch SW1.