



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

Microcontroladores

Materia

Práctica No. 1

**Miguel Ángel González Escalona, Roberto López Arce, Ramiro Lázaro
Bonilla, Alfredo Camarillo Bautista**

Integrantes del equipo

Ingeniería en mecatrónica

Carrera

Ricardo Álvarez González

Docente

08 de febrero de 2020

Fecha de entrega

Marco teórico

El pic 18f4550 es un microcontrolador de 8 bits de la empresa Microchip. Este microcontrolador cuenta con una gran cantidad de memoria RAM, diferentes módulos de comunicación, una gran cantidad de pines de entrada y salida y algunas otras grandes cualidades. Algunas de sus características son:

- 40 pines tipo DIP
- Interfaz USB 2.0 de alta velocidad, EEPROM 256 bytes
- Memoria RAM 2048 bytes, EEPROM 256 bytes
- Memoria de programa (memoria flash) 32 kb
- Voltaje de operación 2 a 5.5 V
- Frecuencia máxima 48 MHz
- 35 pines de entrada / salida

Las instrucciones utilizadas en la práctica se enlistarán y explicarán a continuación.

| Instrucción | Sintaxis | Operación | Palabras y Ciclos | Descripción |
|-------------|-----------------------|---|------------------------|--|
| Bcf | [label] BCF f,b[,a] | 0 -> f | 1 palabra 1 ciclo | El bit 'b' en el registro 'f' es limpiado. Si 'a' es 0, el Access bank será seleccionado, sobre escribiendo el valor de BSR. Si 'a' es 1, entonces el banco será seleccionado según el valor de BSR (default). |
| Bra | [label] BRA n | (PC) + 2 + 2n -> PC | 1 palabra 2 ciclos | Suma el complemento a 2 '2n' al pc |
| Bsf | [label] BSF f,b[a] | 1 -> f | 1 palabra 1 ciclo | El bit 'b' en el registro 'f' es colocado. Si 'a' es 0 el Access bank será seleccionado, sobre escribiendo el valor de BSR. Si 'a' es 1, entonces el banco será seleccionado según el valor de BSR (default). |
| Btfss | [label] BTFSS f,b[,a] | Skip if (f) = 1 | 0 palabras 0 ciclos | Si el bit 'b' en el registro 'f' es 1, entonces la siguiente instrucción es omitida. Si el bit 'b' es 1, entonces la siguiente instrucción traída durante la instrucción ejecutada actualmente. |
| Call | [label] CALL k[,s] | (PC) + 4 -> TOS, k -> PC<20:1>, if s = 1 (W) -> WS, (STATUS) -> STATUSS, (BSR) -> BSRS | 0 palabras 0 ciclos | Es un llamado de subrutina de un rango de memoria de 2 Mbytes. |
| Clrf | [label] CLRF f[,a] | 00h -> f 1 -> Z | 1 palabra 1 ciclo | Limpia el contenido de un registro especificado. |

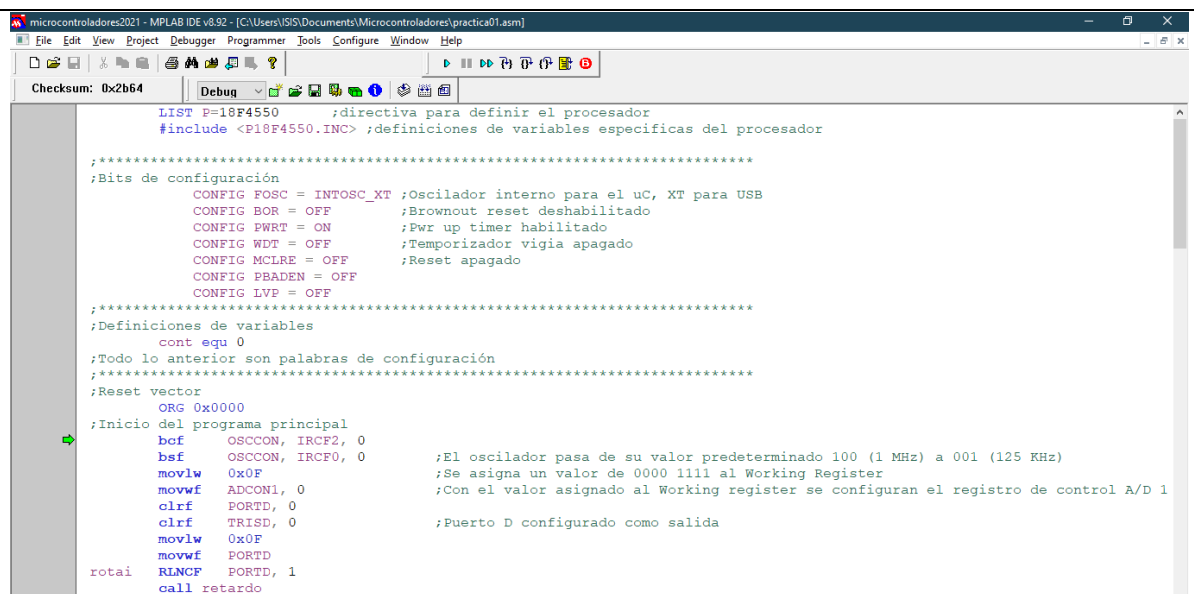
| | | | | |
|--------|----------------------------|--|------------------------|---|
| Decfsz | [label] DEFSZ f[,d[,a]] | (f) -- 1 -> dest, skip if result = 0 | 0 palabras 0 ciclos | Decrementa un valor al registro seleccionado, salta si es 0 |
| Movlw | [label] MOVLW k | K -> W | 1 palabra 1 ciclo | Mueve el valor literal de K al working register |
| Movwf | [label] MOVWF f[,a] | W -> f | 0 palabras 0 ciclos | Mueve el valor de W al registro seleccionado |
| Nop | [label] NOP | Ninguna operación | 1 palabra 1 ciclo | Ocupa simplemente el tiempo de 1 palabra y 1 ciclo |
| Return | [label] RETURN [s] | (TOS) -> PC; if s = 1 (WS) -> W; (STATUS) -> STATUS; (BSR) -> BSR; PCLATU, PCLATH are unchanged | 0 palabras 0 ciclos | Regresa de la subrutina |
| Rlncf | [label] RLNCF f[,d[,a]] | (f<n>) -> dest <n+1> (f<7>) -> dest<0> | N, Z | El contenido del registro f rota un bit hacia la izquierda |
| Rrncf | [label] RRNCF f[,d[,a]] | (f<n>) -> dest <n-1> (f<0>) -> dest<7> | N, Z | El contenido del registro f rota un bit hacia la derecha |

Desarrollo práctico

Para la práctica realizada se utilizaron los siguientes materiales a enlistar.

| Cantidad | Concepto |
|----------|-------------------------------------|
| 1 | Pic 18F4550 |
| 1 | Laptop |
| 8 | LEDS |
| 1 | Display de 7 segmentos cátodo común |
| 8 | Resistencias de 330 Ohms |
| 1 | Software MPLAB v8.92 |
| 1 | Datasheet del PIC 18F4550 |

Lo primero a realizar fue la decisión de que secuencia iban a realizar los LEDS en nuestra práctica para después poder escribir el código en lenguaje ensamblador.



```
microcontroladores2021 - MPLAB IDE v8.92 - [C:\Users\ISD\Documents\Microcontroladores\practica01.asm]
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x2b64 Debug
LIST P=18F4550 ;directiva para definir el procesador
#include <P18F4550.INC> ;definiciones de variables especificas del procesador

;*****
;Bits de configuración
CONFIG FOSC = INTOSC_XT ;Oscilador interno para el uC, XT para USB
CONFIG BOR = OFF ;Brownout reset deshabilitado
CONFIG PWRT = ON ;Pwr up timer habilitado
CONFIG WDT = OFF ;Temporizador vigia apagado
CONFIG MCLRE = OFF ;Reset apagado
CONFIG PBADEN = OFF
CONFIG LVP = OFF
;*****
;Definiciones de variables
cont equ 0
;Todo lo anterior son palabras de configuración
;*****
;Reset vector
ORG 0x0000
;Inicio del programa principal
bcf OSCCON, IRCF2, 0
bsf OSCCON, IRCF0, 0 ;El oscilador pasa de su valor predeterminado 100 (1 MHz) a 001 (125 KHz)
movlw 0x0F ;Se asigna un valor de 0000 1111 al Working Register
movwf ADCON1, 0 ;Con el valor asignado al Working register se configuran el registro de control A/D 1
clrf PORTD, 0
clrf TRISD, 0 ;Puerto D configurado como salida
movlw 0x0F
movwf PORTD
rotai RINCF PORTD, 1
call retardo
```

Imagen 1. Parte 1 del código en mplab

Imagen 2. Parte 2 del código en mplab

Imagen 3. Parte 3 del código en mplab

Cálculos

Los cálculos que se realizaron en la realización de esta práctica fueron utilizados para aproximar el tiempo que duraría la instrucción de retardo a 1 segundo. Sabemos que:

$$F_{oscilación} = 32 \text{ KHz}, T_{oscilación} = 125 \mu\text{segundos}$$

$$Tiempo = (Ciclos \text{ de máquina})(F_{oscilación}) \rightarrow 1 \text{ segundo} = Ciclos (125 \mu\text{segundos})$$

$$Ciclos = \frac{1 \text{ segundo}}{125 \times 10^{-6} \text{ segundos}} = 8000 \text{ Ciclos}$$

La duración de la rutina “retardo” y “nada” tiene 1029 ciclos de máquina, para aproximar los ciclos a 8000 y no alterar la lógica de la rutina simplemente se aumentaron más instrucciones “nop”.

| Rutina | | | Ciclos de máquina (Tcy) |
|---------|--------|------------|---|
| Retardo | movlw | 0xff | 1 |
| | Movwf | cont, 0 | 1 |
| Nada | nop | | 1, 1 |
| | Decfsz | cont, 1, 0 | 1 |
| | Bra | nada | 2 |
| | Return | | 2 |
| | | | 2+(1+1+2)256+3=5+1024=1029 ciclos de máquina. Si aumentamos la cantidad de instrucciones “nop” para lograr 8000 ciclos, tendríamos que hacer los siguientes cálculos. |

$$(x + 1 + 2)256 = 8000 \rightarrow (x + 3)256 = 8000 \rightarrow x + 3 = \frac{8000}{256} \rightarrow x + 3 = 31.25 \rightarrow x = 28.25$$

Por lo tanto, tendríamos que agregar 28 o 29 instrucciones “nop” aproximadamente para lograr que sea un retardo de 1 segundo.

Simulación

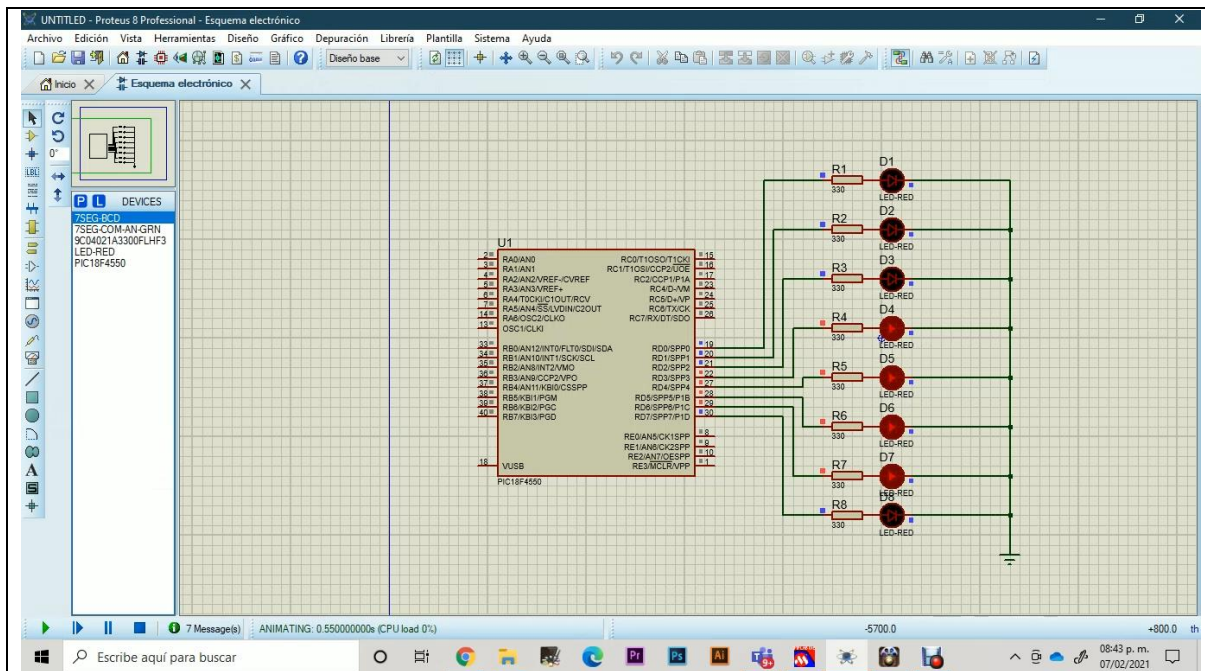


Imagen 4. Simulación en proteus

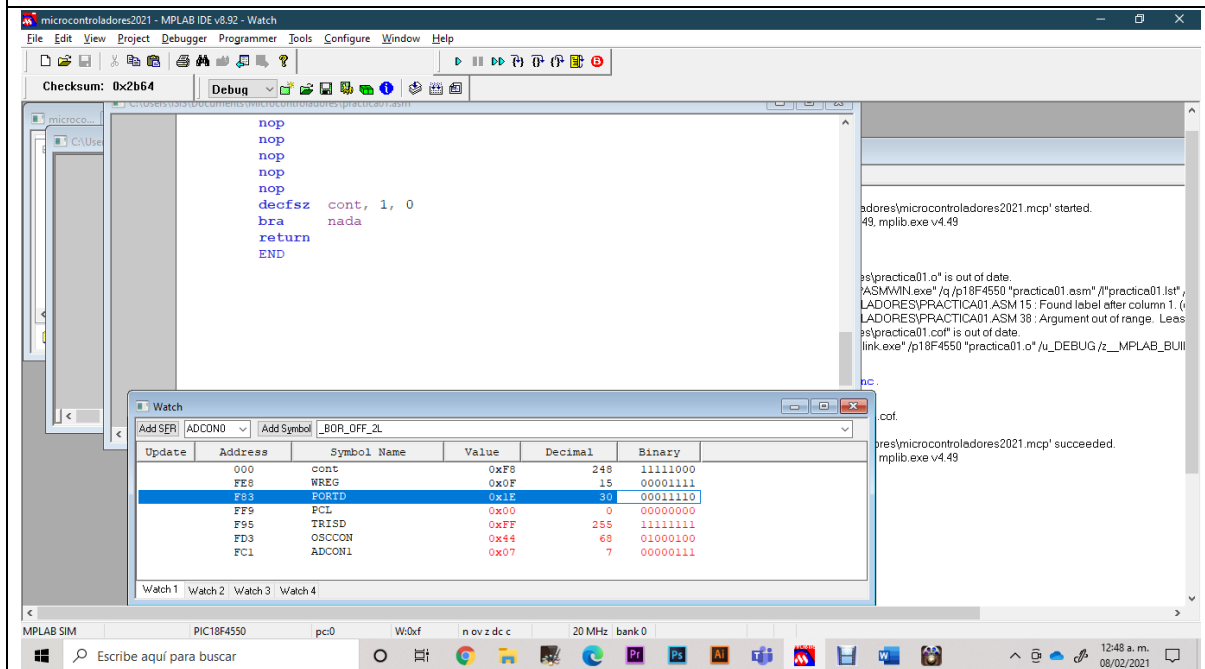


Imagen 5. Simulación en mplab

Resultados

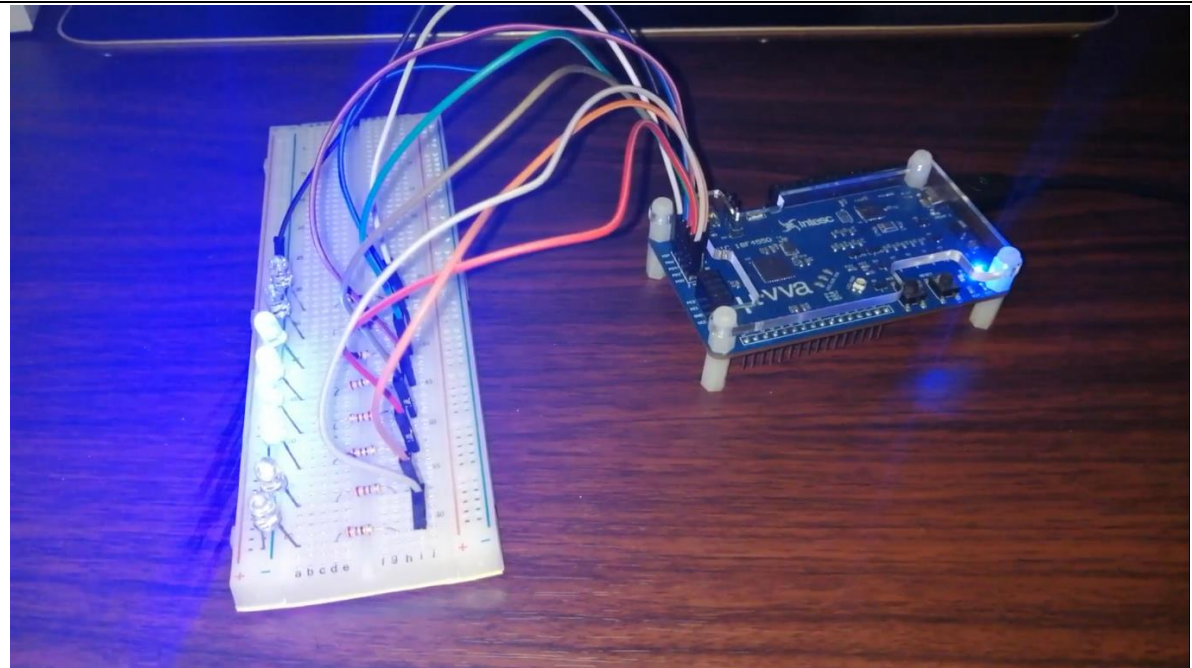


Imagen 6. Funcionamiento del código de la práctica implementado en la tarjeta de desarrollo miuva.

Conclusiones

De la práctica pudimos aprender la estructura básica para realizar un código en lenguaje ensamblador que nos permita poder establecer algunos pines como salida de datos, además de que se aprendieron las instrucciones básicas (o al menos unas de las más comunes) dentro del lenguaje ensamblador.

Bibliografía

Microchip PIC18F Instruction Set. (2021). Retrieved 9 February 2021, from http://technology.niagarac.on.ca/staff/mboldin/18F_Instruction_Set/