



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Eléctrica y Electrónica

Carrera de Electrónica y Automatización

SISTEMAS BASADOS EN MCU

Práctica 1.1

**MANEJO DEL SIMULADOR MPLAB IDE DE LA
MICROCHIP**

Autor:

Iza Tipanluisa Alex Paul

Docente:

Ing. Amparo Meythaler

NRC: 4891

1) OBJETIVOS

- Identificar la interface y opciones básicas del MPLAB IDE de la MICROCHIP.
- Realizar una aplicación que permita realizar un corrido en pasos de las instrucciones de la fábrica MICROCHIP en el MPLAB IDE.

2) MARCO TEORICO

MPLAB IDE DE LA MICROCHIP

El MPLAB es un entorno de desarrollo es decir es un recipiente que incluye varias herramientas.

Contiene un editor de textos que no permite ingresar el programa expresado en códigos nemónico (o simplemente llamado ensamblador), normalmente este se guarda en un archivo con extensión ASM. Una vez que hemos ingresado el programa dentro de un archivo creamos un proyecto dentro del MPLAB el proyecto puede contener a su vez varios archivos ASM que se relacionen a través de llamadas a rutinas o compartan y/o variables, adicionalmente el proyecto tiene un grupo de variables que debemos configurar como es el tipo de microcontrolador que vamos a usar ya que el MPLAB soporta todas las familias de microcontroladores Microchip (MPLAB es producido por Microchip). A continuación, procedemos a llamar al programa ensamblador que lleva el MPLAB capaz de transformar los códigos nemónicos (instrucciones) a los correspondientes valores binarios que a su vez grabaremos en el microcontrolador.

El ensamblador (MPASMWIN) también genera otros archivos de salida que ayudan en el diseño de aplicaciones. Dentro del MPLAB encontramos también al MPSIM que es un potente simulador que nos permitirá observar el comportamiento del programa antes de proceder grabar el programa. El archivo. HEX es el que contiene los códigos binarios a grabar en el

Microcontrolador.

Existen varias versiones, e inclusive una versión tipo X. Pero todas presentan opciones para editar, compilar y simular ejercicios realizados en assembler.

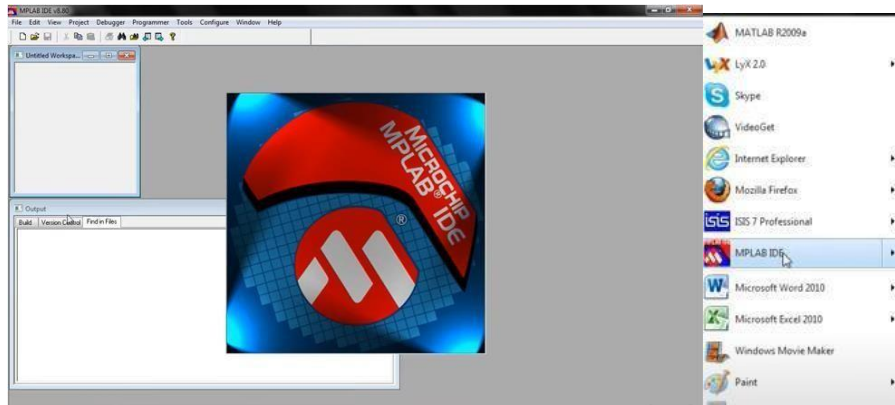
3) EQUIPOS Y MATERIALES

- PC con el paquete MPLAB IDE de la MICROCHIP.

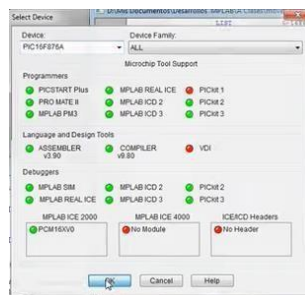
4) ACTIVIDADES

a) Interface inicial.

Ingresa la aplicación.

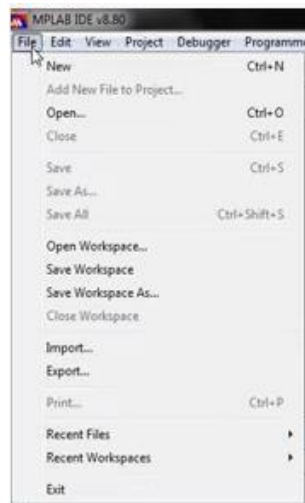


Cierre la ventana de Output y minimice la ventana Workspace. Escoja el microcontrolador PIC16F877 en Configure, Select Device



b) Digitar el programa.

Digite un ejercicio de TRES INSTRUCCIONES en una hoja en blanco en File, New.



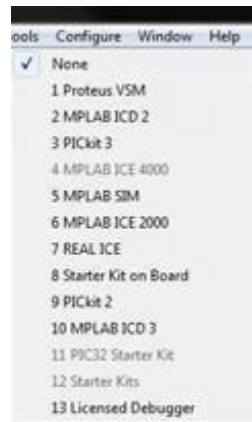
Siga las instrucciones del docente, desde que digita un programa hasta que almacena el programa realizado.

c) Compilar el programa.

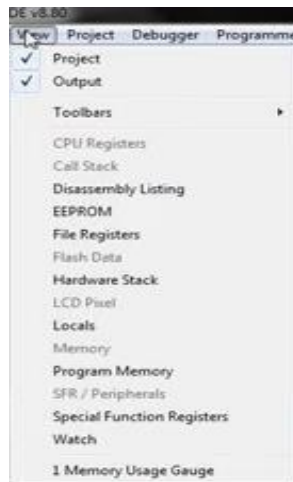
Digita ALT F10, para compilar el programa, o su equivalente, hasta que NO tenga errores de Sintaxis.

d) Correr el programa en pasos

Escoger que MPLAB sea la herramienta de depuración, en Debugger. Esto es en el ítem MPLAB SIM.



Siga las instrucciones del docente para correr el ejercicio en pasos. Revise los resultados en W, las banderas y la memoria de datos (File Registers).



e) Repita todos los pasos anteriores para comprobar los resultados del ejercicio siguiente:

LIST P = PIC16F877 ORG 0

MOVLW DF

MOVWF 36

XORLW B7

MOVWF 42

COMF 36,0

SUBWF 42,0

RRF 36,1

SWAPF 42,1

XORWF 36,0

ADDLW 5A

IORWF 42,1

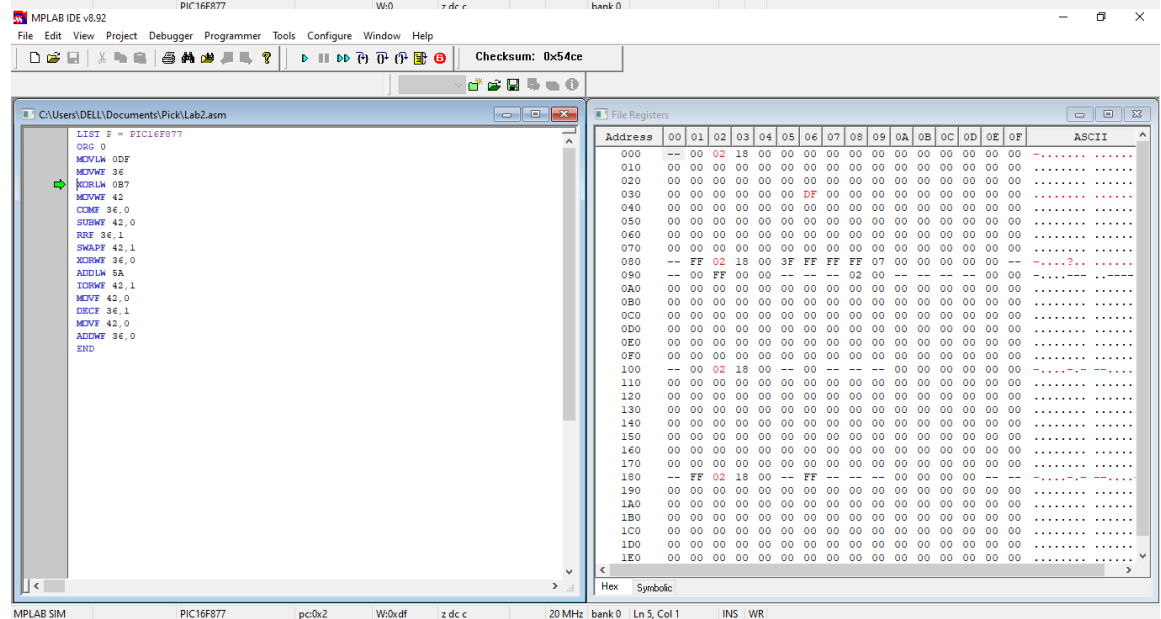
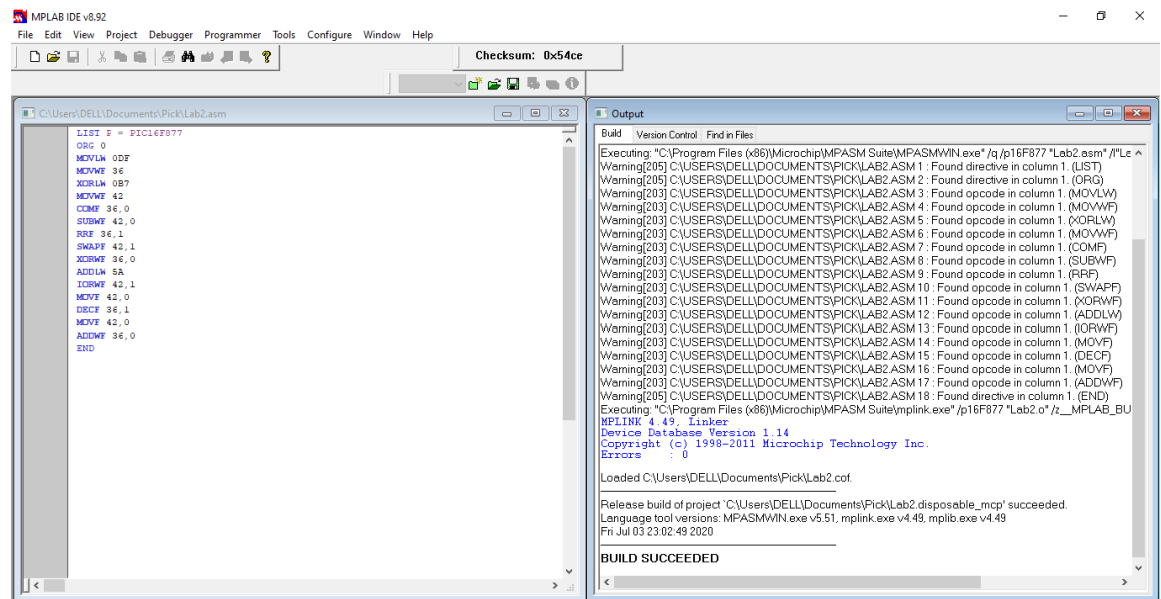
MOVF 42,0

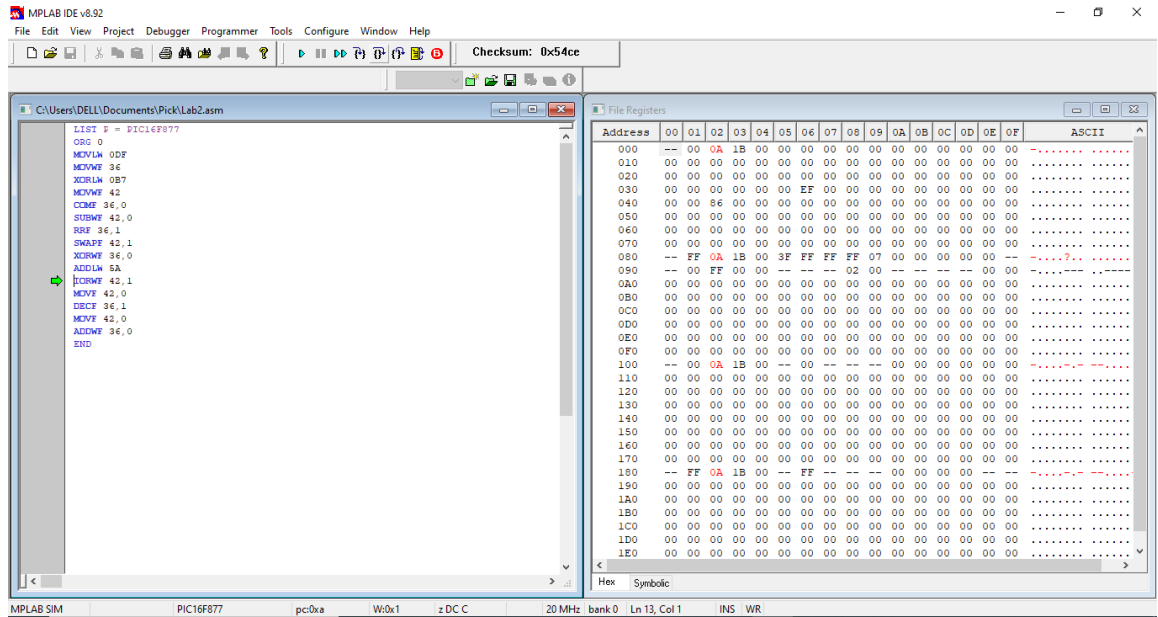
DECF 36,1

MOVF 42,0

ADDWF 36,0

END





Ponga los resultados en W, las localidades y las banderas junto con cada instrucción.

LIST P = PIC16F877		XORLW B7	
ORG 0	ORIGEN 0	DF → 11011111	Z=0
MOVLW DF	W = DF	B7 → 10110111	DC=0
MOVWF 36	[36] = DF	68 → 01101000	C=0
XORLW B7	W = 68 Z=0		
MOVWF 42	[42] = 68		
COMF 36,0	W = 20		
SUBWF 42,0	W = 48		
RRF 36,1	[36] = EF		
SWAPF 42,1	[42] = 86		
XORWF 36,0	W = A7		
ADDLW 5A	W = 01		
IORWF 42,1	[42] = 87		
MOVF 42,0	W = 87		
DECF 36,1	[36] = EE		
MOVF 42,0	W = 87		
ADDWF 36,0	W = 75 Pero no lo realiza		
END	FIN PROGRAMA		

ADDWF 36,0	Z=0	
EE	21-16=5	DC=1
+ 87	23-16=7	C=1
1 75		

DF → 11011111	Z=0
B7 → 10110111	DC=0
68 → 01101000	C=0
COMF 36,0	Z=0
DF → 11011111	DC=0
00100000	C=0
2 0	
SUBWF 42,0	Z=0
- 68	DC=1
20	C=1
48	
RRF 36,1	Z=0
DF → 11011111	DC=1
11101111	C=1
E F	
XOR 36,0	Z=0
EF → 11101111	DC=1
- 48	C=1
01001000	
10100111	
A 7	
ADDLW 5A	Z=0
+ A7	10+7=17-16=1
SA	16-16=0
101	C=1
1	
IORWF 42,1	Z=0
86 → 10000110	DC=1
01 → 00000001	C=1
10000111	
8 7	
DECF 36,1	Z=0
[36] - 1	DC=1
EF	C=1
- 1	
EE	

5) RESULTADOS

- **Explique la forma de correr en pasos un programa en el paquete utilizado, luego de que compiló satisfactoriamente**

El corrido en pasos permite compilar al programa línea por línea, permitiendo así verificar los datos que se encontraron calculándolos a mano.

- **¿En las instrucciones de direccionamiento directo, explique qué significa la d y qué valor puede tener?**

El valor de d puede ser 0 o 1; el dígito 1 se usa para que la respuesta se guarde en una localidad y el dígito 0 se usa para que la solución se guarde en el registro W

- **Explique, qué debe hacer para que en el simulador aparezcan las respuestas de la última instrucción (antes del END).**

Se debe colocar una instrucción llamada NOP la cual realiza ninguna función, pero permite que el programa permita presentar en pantalla la última instrucción.

6) CONCLUSIONES

- Se ha concluido que la plataforma MPLAB es una herramienta que ayuda a simular y verificar que los comandos que realizan operaciones estén correctamente escritos, así también nos permite visualizar los registros en los que se pueden guardar los datos y la modificación de sus banderas.
- Se ha concluido que en la plataforma MPLAB tiene un solo registro llamado W en el cual se pueden guardar o asignarse números para que posteriormente se use como dato para realizar las operaciones esto en ciertos comandos.
- Se ha concluido que el operando directo llamado registro es de gran utilidad ya que nos permite asignarle el valor de la respuesta ya sea en una localidad o en el registro W

7) RECOMENDACIONES

- Se recomienda revisar y realizar correctamente las operaciones ya que, si alguna de ellas se encuentra mal, las demás operaciones también tendrán errores, lo cual crea confusiones al momento de comprobar las soluciones.
- Se recomienda conocer los operandos ya que estos se pueden guardar la solución de las tareas en el registros o en una localidades según corresponda, generando así confusiones en el momento de buscar el dato de la solución.
- Se recomienda que, al inicio del programa, este inicie con LIST P=PIC16F877 (para este caso), así como que el programa debe iniciar en la localidad cero y todos los datos que inicien con una letra se debe ante poner un cero y que el nombre del programa se guarde con la extensión **.asm**

8) BIBLIOGRAFIA

Abel, P. (1996). Lenguaje ensamblador y programación para IBM PC y compatibles. México: Prentice Hall.