

Instituto Tecnológico de Zitácuaro

INGENIERÍA EN SISTEMAS COMPUTACIONALES

“PRACTICA 1”

ESTUDIANTE:

ESPINOSA ROY MARIA DANIELA

Nº. DE CONTROL:

17650021

MAESTRO:

EDGAR JIMÉNEZ PINEDA

FECHA:

06/10/2020

H.ZITACUARO MICH., A 06 DE OCTUBRE DEL 2020.

Practica 1: Introducción al MPLAB y uso de puertos I/O en un microcontrolador PIC.

Objetivo general

- Utilizar el conjunto de programas MPLAB IDE v8.88 y MiniPROG+ para el desarrollo de programas en microcontroladores PIC.

Objetivos específicos

- Crear y compilar programas en lenguaje ensamblador por medio del MPLAB IDE.
- Desarrollar una aplicación basada en un sistema mínimo, inicializando el conjunto básico de periféricos necesarios para ello.
- Utilizar el programador MiniPROG+ para la descarga de los programas en el dispositivo físico.
- Comprender la utilización de los puertos de entradas y salidas digitales.

Introducción

Se aprenderá como utilizar el entorno de desarrollo del programa MPLAB IDE v8.88, su simulación, así como la descarga física del programa en lenguaje ensamblador al microcontrolador.

Se abordara los siguientes temas:

- Programación en lenguaje ensamblador
- Modos de direccionamiento
- Conjunto de instrucciones
- Instrucciones aritméticas
- Instrucciones lógicas
- Instrucciones de control de programa
- Estructura del programa
- Programación de puertos digitales

Materiales y equipo

- Computadora personal, sistema operativo Windows 7 o superior, software MPLAB IDE v8.88 y PICKit 2.
- Tarjeta entrenadora TEC-PIC877 V1.0 R1
- Fuente de poder de 12Vcd a 1A o un cable de USB A macho / B macho.
- Programador MiniPROG+.
- Microcontrolador pic16f877, oscilador de cuarzo 4 MHz, 2 capacitores de 15pF, 8 diodos LED con 8 resistencias de 330 Ω , 1 resistencias de 10K Ω .



Procedimiento

1. Construya el circuito presentado en la figura 7 en el simulador de PROTEUS. Dicho circuito se probará en la tarjeta entrenadora TEC-PIC877 V1.0 R1. Conecte 8 jumpers en el header P7-PORTD los cuales conectan al puerto D del microcontrolador a los 8 LED's de dicha tarjeta (LED1-LED8). Conectará la tarjeta a la fuente de voltaje de 12 Vcd. El circuito se encenderá más adelante.

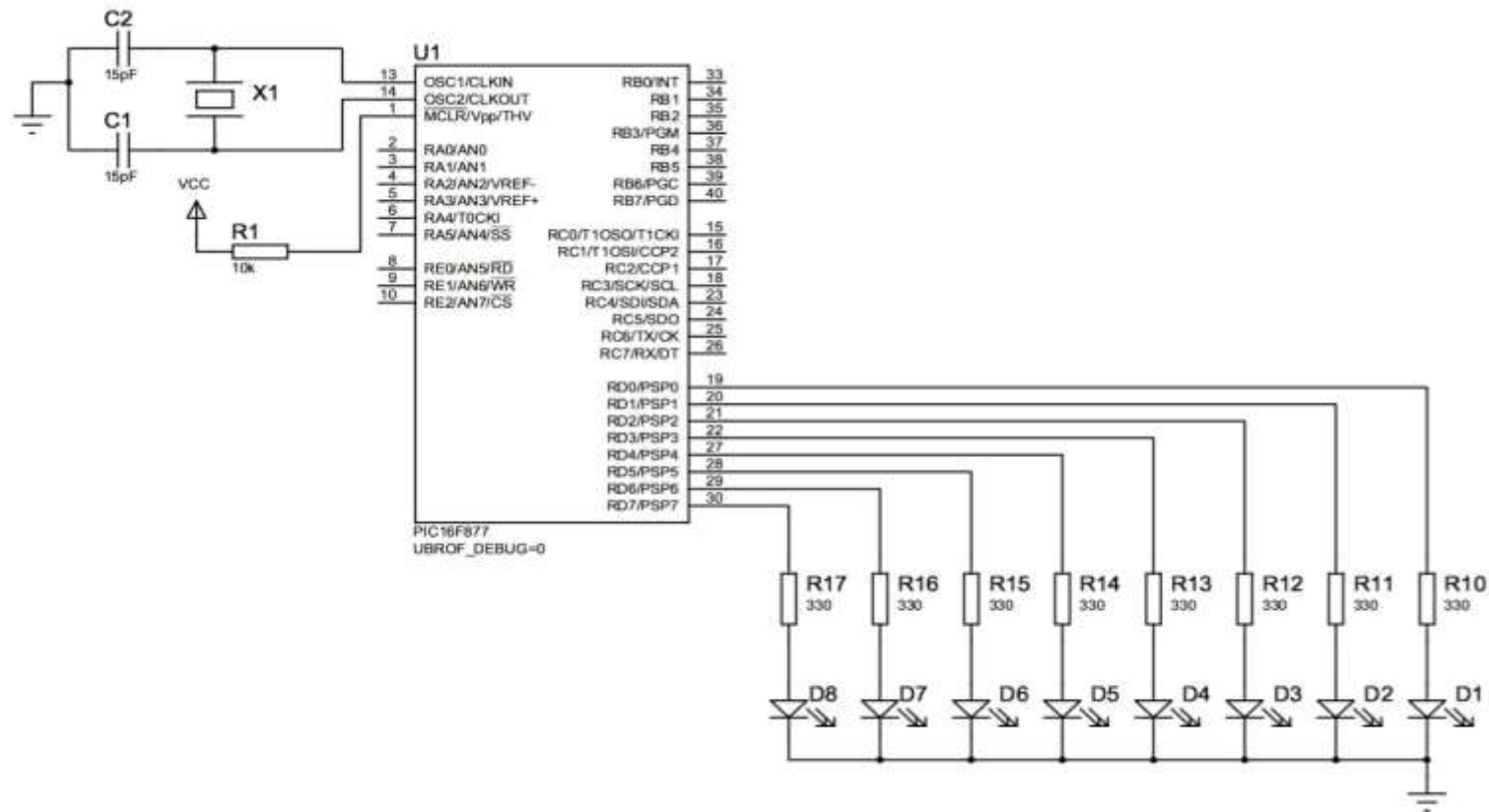


FIGURA 7 ESQUEMA PRACTICA 1

2. A continuación ejecute el programa MPLAB. Diríjase al menú “Project -> Project Wizard...” y aparecerá el cuadro de bienvenida al asistente de proyectos. Pulse el botón “Siguiente”.
3. En el cuadro de selección de dispositivo, seleccione el microcontrolador PIC16F877.
4. En el cuadro de dialogo de selección de herramientas de desarrollo (figura 1), asegúrese de usar el conjunto de programas del MPASM (Microchip MPASM toolsuite). Verifique también que las herramientas tengan sus rutas configuradas en cada una de las 3 categorías que aparecen (MPASM, MPLINK y MPLIB). Si los programas no se encuentran o las rutas están en blanco, consulte a su instructor para que le ayude a reconfigurarlas. Deje la casilla “Store tool locations in Project” en blanco.

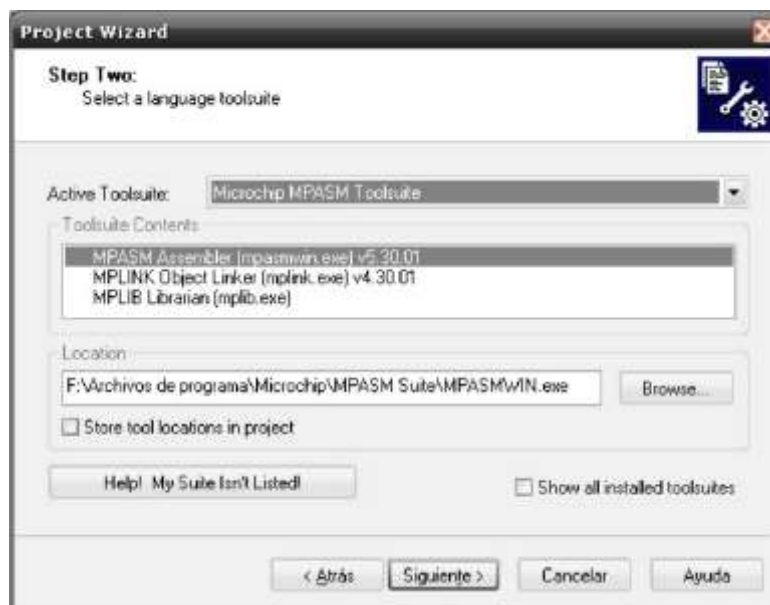


Figura 1: Cuadro de selección de herramientas de desarrollo.

5. A continuación el asistente preguntará dónde desea guardar su proyecto. Pulse el botón “Browse” y aparecerá una ventana de navegación de archivos. Diríjase al escritorio y cree una nueva carpeta ahí, luego cámbiese a esa carpeta y escriba un nombre para su proyecto. Pulse el botón “Guardar” y luego haga clic en “Siguiente”.

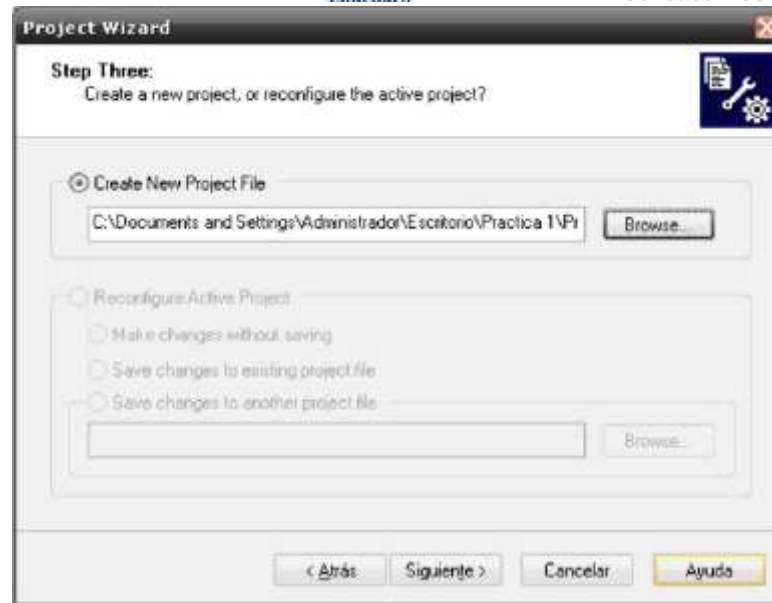


Figura 2: Cuadro de creación de proyectos.

6. A continuación aparecerá una ventana en la que se le solicita agregar archivos existentes a su proyecto. Como este es un proyecto nuevo, no existen archivos, así que déjela vacía y haga clic en "Siguiete".

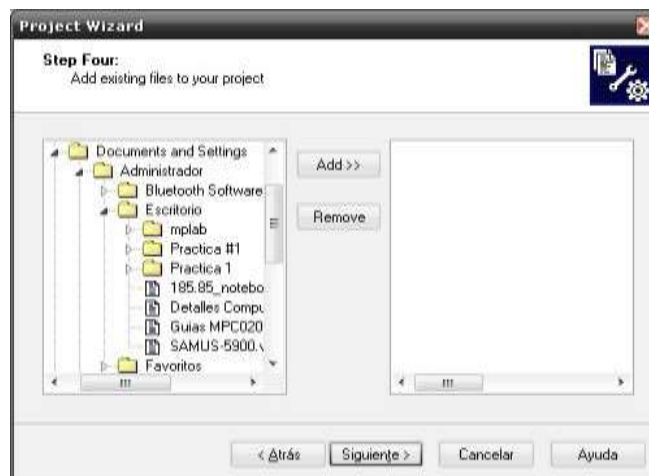


Figura 3: Ventana para agregar archivos existentes al nuevo proyecto.

7. Haga clic en "Finalizar" para cerrar el asistente y cargar el nuevo proyecto.
8. El siguiente paso consiste en agregar un archivo de código fuente. Para ello diríjase al menú "Project -> Add New File to Project". Asegúrese que se encuentra dentro de la carpeta que se creó en el paso 6 y escriba el nombre del archivo de código fuente. Nota importante: Asegúrese de agregar la extensión ".asm" al final del nombre del archivo, de lo contrario, no será reconocido como un archivo de código fuente.
9. Escriba el programa que se muestra en la Listado 1 (puede omitir los comentarios, los cuales inician con un punto y coma), si cuenta con una copia digital de este documento, puede copiar el texto directamente hacia el MPLAB.

```
List p=16f877A                ;list directive to define processor
#include <p16f877.inc>          ;processor specific variable definitions
;*****
ORG 0x000                     ; processor reset vector nop
                                ; nop required for icd goto main
                                ; go to beginning of program
ORG 0x004                     ; interrupt vector location

main                           ; remaining code goes here call
    INIT_PUERTOS
Rutina movlw 0xFF               ;Cargamos un 0xFF al registro W xorlw
    0x55                       ;Invierte los bits, las posiciones pares movwf
    PORTD                      ;Guarda el nuevo dato en el puerto C goto
Rutina                         ;Repite el proceso

INIT_PUERTOS
    Bsf STATUS, RP0 ;Selecciona el banco 1(inicialmente es 0) movlw
    0x00                  ;Establece todo puerto B como entradas movwf TRISD
    bcf STATUS, RP0      ;Selecciona el banco 0 return

END                            ; directive 'end of program'
```

Listado 1: Código fuente

10. El proyecto ahora está listo para ser ensamblado, presione la teclas ctrl + F10 o bien diríjase a "Project -> Build All". Aparecerá la ventana de resultados, indicando si el proceso se realizó correctamente. Si ocurren errores, verifique en dicha ventana la lista generada. Puede hacer doble clic sobre una línea de error para saltar directamente a la línea de código que lo genera.
11. Prepare el programador conectando primero el cable USB al programador y el otro extremo a la PC.
12. Diríjase al menú "Programmer -> Select Programmer -> 9 PICkit 2"

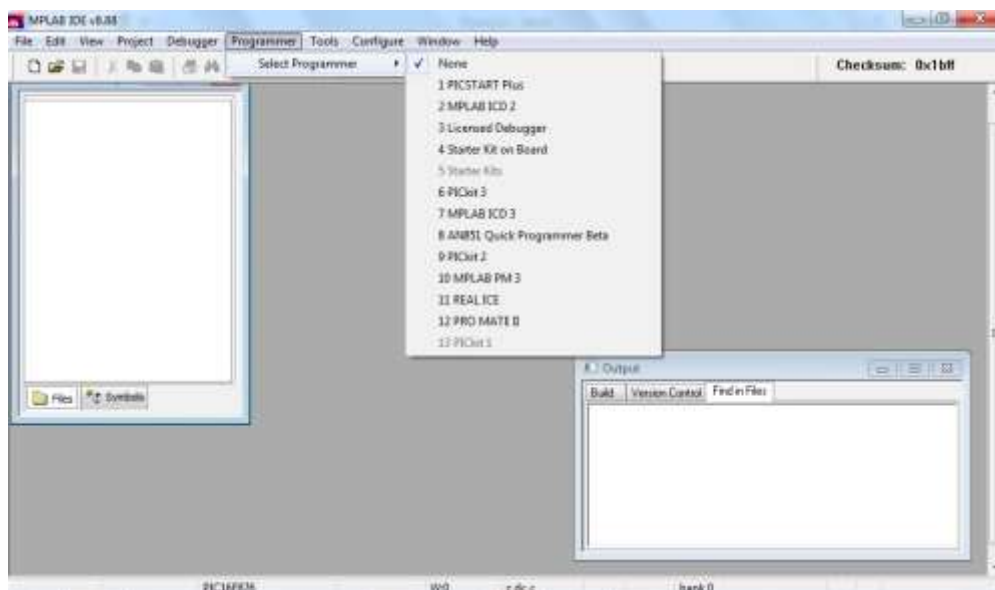


Figura 4: Ventana de configuración de hardware de MiniProg +.

13. Si ejecuta la detección automática sabrá si el hardware está operando correctamente al ser detectado con éxito el cual lo veremos en la ventana de Output

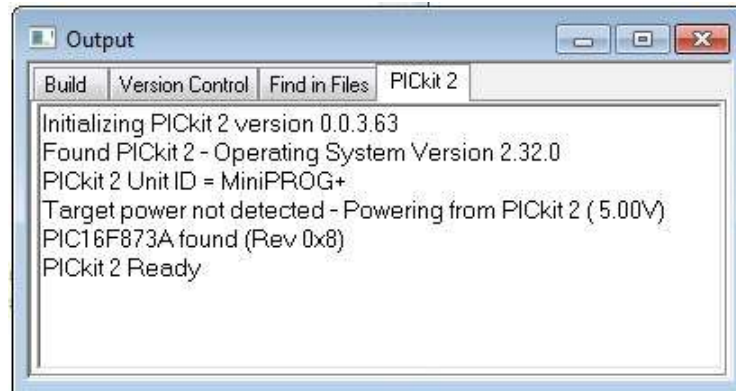


Figura 5: Ventana de programador con éxito.

14. Seleccione el menú “Configure -> Configuration Bits”, en este vamos a configurar el MPU, para ser grabado de forma adecuada, la configuración adecuada para nuestro sistema es como se muestra en la figura 8.

Configuration Bits set in code.				
Address	Value	Field	Category	Setting
2007	1879	FOSC	Oscillator Selection bits	XT oscillator
		WDTEN	Watchdog Timer Enable bit	WDT disabled
		PNRTE	Power-up Timer Enable bit	PNRT disabled
		BOREN	Brown-out Reset Enable bit	BOR enabled
		LVP	Low-Voltage (Single-Supply) RBS is digital I/O, HV on MCLR must be used for programming	
		CPD	Data EEPROM Memory Code Prot Data EEPROM code-protected	
		WRT	Flash Program Memory Write 0000h to 0FFFh write-protected; 0800h to 0FFFh may be written to	
		CP	Flash Program Memory Code Prot All program memory code-protected	

Figura 6: Configuración del MPU.

15. Seleccione el menú “Programmer -> program”, esto iniciará el proceso de grabación del programa en el microcontrolador. Una vez grabado, puede seleccionar el menú “Programmer > Verify” para corroborar que el programa fue grabado apropiadamente.
16. En el instante en que el programa es grabado en el microcontrolador, coloque este en la tarjeta entrenadora. Haga algunas modificaciones en el código fuente con MPLAB (como por ejemplo, cambiar la instrucción `xorlw` por `andlw` en la rutina principal) y ensamble nuevamente el programa (presionando `ctrl. + F10`). Corrobore nuevamente el funcionamiento del circuito.
17. Ahora se procederá a borrar el microcontrolador, para ello simplemente va al menú “Programmer -> Erase”.

Análisis de resultados

- ¿Qué efecto se tendría si dentro del código fuente se cambiara la instrucción “`xorlw 0x55`” por “`xorlw 0xF0`”? Explique.

Ninguno puesto que en las dos instrucciones; ya que en ambos casos sirve para lo mismo que en este caso es colocar unos en el registro w.



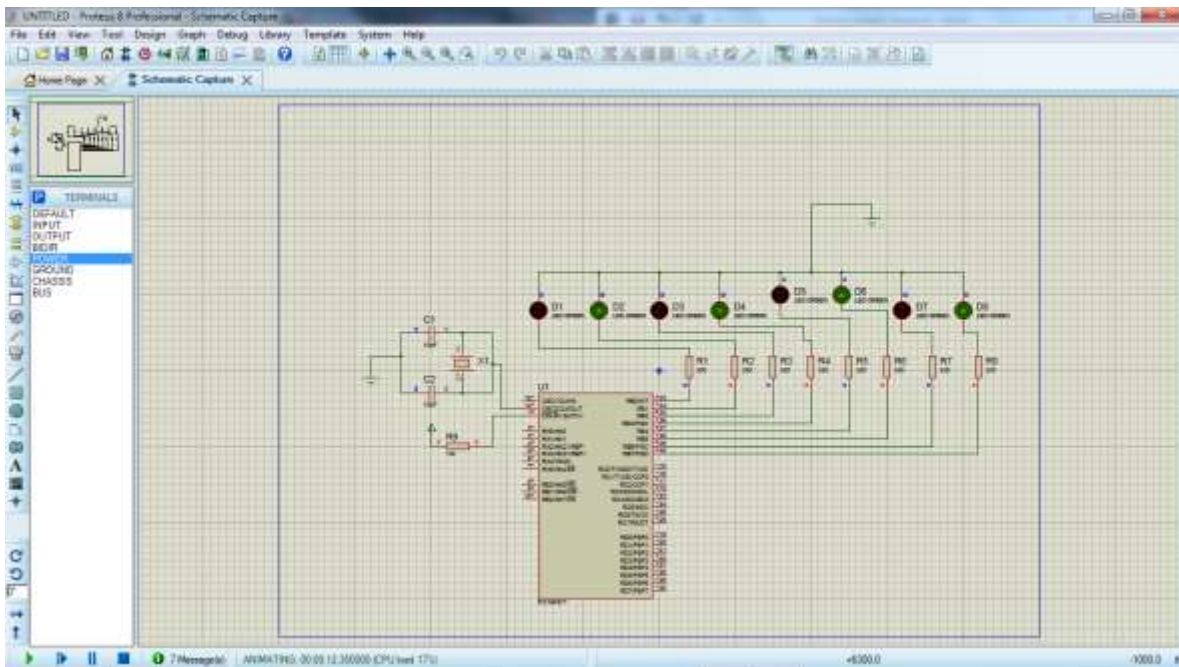
¿Es posible intercambiar la función de los puertos C y D del microcontrolador? **Si**

- De ser así, ¿Qué cambios habría que hacer tanto al hardware como al software para lograrlo?

```
main
call INIT_PUERTOS

Rutina
movlw 0x00 ;Cargamos un 0xFF al registro W
xorlw 0xAA ;Invierte los bits, las posiciones pares
movwf PORTB ;Guarda el nuevo dato en el puerto C
goto Rutina ;Repite el proceso

INIT_PUERTOS
Bsf STATUS, RP0 ;Selecciona el banco 1(inicialmente es 0)
movlw 0x00 ;Establece todo puerto B como entradas
movwf TRISB
bcf STATUS, RP0 ;Selecciona el banco 0
return
```



Investigación complementaria

- Investigue el uso y sintaxis de los archivos de guion de enlace (linker script). Estudie la línea de programa "PROG1 code" y explique qué correlación existe con el guión de enlace utilizado.



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Zitácuaro

Referencias bibliográficas

- Hoja técnica del microcontrolador PIC16F877:
<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>
- Manual de usuario del MPLAB:
http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB_User_Guide_51519c.pdf

Practica 1: Introducción al MPLAB y uso de puertos I/O en un microcontrolador PIC.

Alumno: Espinosa Roy Maria Daniela

Puesto No:

Docente: Edgar Jiménez Pineda

Fecha: 06 - OCT - 2020

EVALUACION					
	%	1-4	5-7	8-10	Nota
CONOCIMIENTO	25	Explica deficientemente la operación de los puertos de I/O.	Explica parcialmente la operación de los puertos.	Explica correctamente la operación de los puertos.	
APLICACIÓN DEL CONOCIMIENTO	25	No crea el proyecto ni compila el programa mediante MPLAB.	Crea el proyecto, pero tiene problemas compilando el programa.	Crea el proyecto y genera exitosamente el archivo de programación.	
	25	No ensambla correctamente el circuito de la práctica.	Ensambla el circuito, pero el mismo presenta problemas al operar.	Ensambla el circuito correctamente y funciona sin problemas.	
	20	No descarga el programa al hardware mediante MiniPROG+.	Descarga el programa, pero tiene mal configurado el software.	Descarga el programa y toda la operación ocurre sin advertencias.	
ACTITUD	2.5	Es un observador pasivo.	Participa ocasionalmente o lo hace constantemente pero sin coordinarse con su compañero.	Participa propositiva e integralmente en toda la práctica.	
	2.5	Es ordenado pero no hace un uso adecuado de los recursos.	Hace un uso adecuado de los recursos, respeta las pautas de seguridad, pero es desordenado.	Hace un manejo responsable y adecuado de los recursos conforme a pautas de seguridad e higiene.	
TOTAL	100				