



## Microcontroladores PIC Generación de Proyecto

### Uso del MPLAB

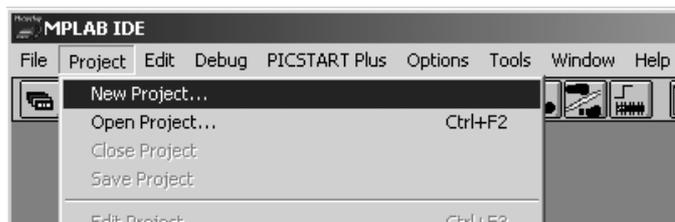
El MPLAB IDE, es un entorno de desarrollo integral, provisto por la misma firma Microchip, en el cual se puede desarrollar un proyecto en assembler y su simulación. Incorporando herramientas de software y/o hardware permite que en el mismo entorno se puedan emular un microcontrolador, programar con lenguaje C y/o grabarlo al PIC con el proyecto que se realizó.

### Generación de un Proyecto

Se describe a continuación la creación de un proyecto, hasta grabarlo al Microcontrolador mediante un grabador de tipo PICSTART Plus

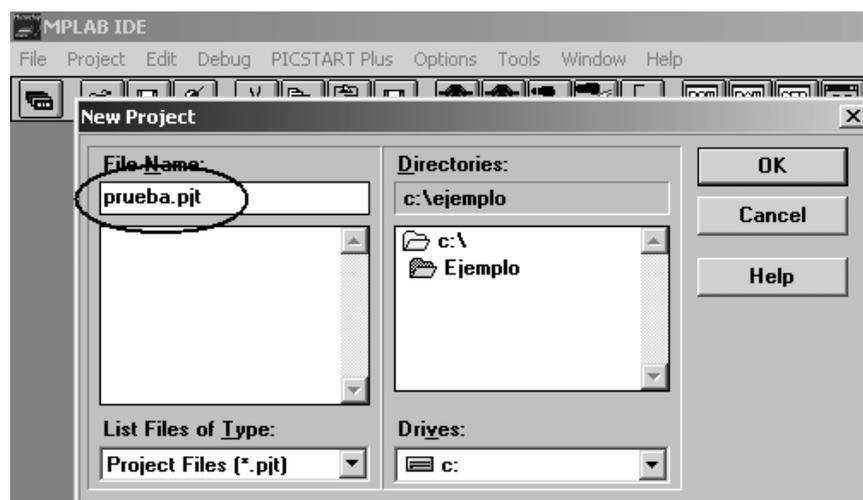
#### Paso 1

Una vez abierto el MPLAB, se crea un proyecto mediante Project → New Project...



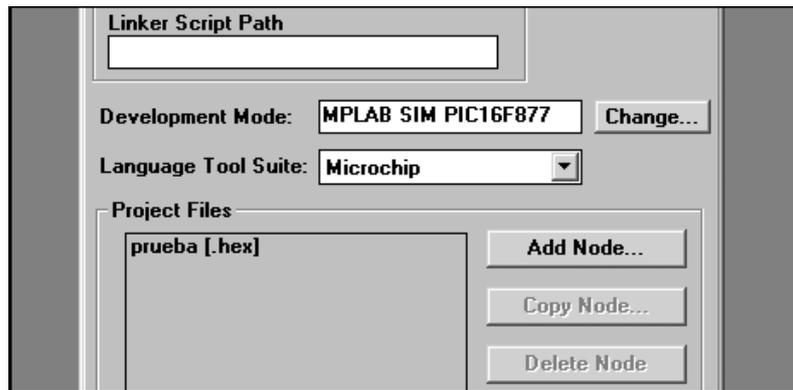
#### Paso 2

En la ventana “New Project” se ingresa el nombre del proyecto y la ubicación, para este caso será **prueba.pjt** el nombre del proyecto ubicado en **c:\ejemplo**



#### Paso 3

Al presionar el botón OK de la pantalla “New Project” se crea el proyecto y se accede a la pantalla de edición del proyecto “Edit Project”



en esta pantalla se debe colocar en el “Development Mode:” el microcontrolador a usar, esta indicación es importante para la simulación y el grabado.

#### Paso 4

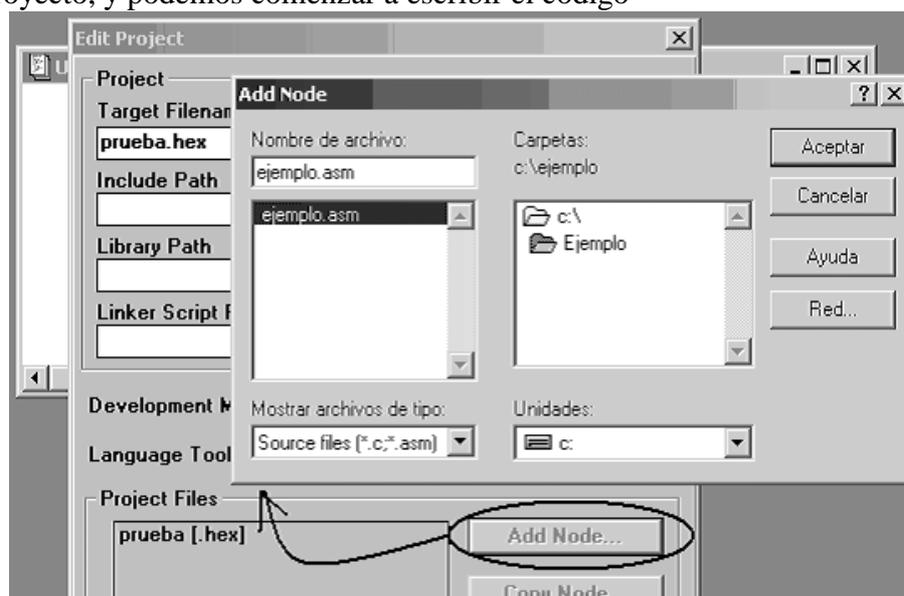
Una vez creado el proyecto se debe comenzar a escribir el código, para esto desde el menú principal se selecciona File → New, este archivo creado deberá ser incorporado a nuestro proyecto para permitir su compilación y simulación, es por esto que, antes de comenzar a escribir el código propiamente dicho, se deberá grabar el archivo recién creado preferentemente en la misma ubicación de nuestro proyecto.

#### Paso 5

Para agregar el archivo al proyecto se debe abrir de nuevo la pantalla de “Edit Project”, ubicada en Project → Edit Project...

#### Paso 6

En “Edit Project” se agrega el nodo, creado en el paso 4, para esto hacemos clic en el botón “Add Node” y seleccionamos el archivo, de esta forma ya tenemos incluido el archivo a trabajar en nuestro proyecto, y podemos comenzar a escribir el código





## Guía rápida del menú

- **File:** Posee todas las opciones para poder crear, grabar o abrir un archivo
- **Project :** Posee todas las opciones para poder crear, grabar o abrir un proyecto
  - Project → Make Project Compila el proyecto actualmente abierto
- **Edit:** Funciones de edición de texto
- **Debug:** Opciones para la simulación
  - **Run:** Opciones para correr el programa
    - **Run:** Corre el programa en memoria
    - **Reset:** Corta la ejecución del programa restableciendo el Program Counter al Vector de Reset
    - **Halt:** Detiene la ejecución del programa
    - **Animate:** realiza la ejecución del programa paso a paso, mostrando actualizando en cada ejecución de instrucción los valores de los registros
    - **Step:** realiza un paso de programa, a partir del lugar posicionado por el Program Counter
    - **Step Over:** realiza la ejecución de la instrucción, indicada por el Program Counter, si esta es una instrucción CALL, Step Over correra esa subrutina deteniéndose en la instrucción siguiente al CALL
- **PICSTART Plus:** en este lugar se habilita el grabador y graba el programa que se encuentra en la memoria al PIC
  - **Enable Programmer:** habilita el uso del programador.

## Programa de Ejemplo

Programa de ejemplo para juego de leds

```
INCLUDE "P16F84.INC" ; Librería de las variables y ctes del 16F84
; -----
;                               Definición de Variables
; -----
CBLOCK 0x0C ; comienzo de la zona de variable (0x0C)
  TEMPTAB ; variable temporal de las tablas
  TEMP1_1 ; variables temporales para funciones de Nivel 1
  TEMP1_2
  TEMP1_3
  CONTADOR ; contador de led
  VAR_WORD:2 ; Declaración de variable de 16 bits
ENDC
; -----
;                               Definiciones de constantes
; -----
; cte de 1 btye
#define CONSTANTE1 D'64'
#define CONSTANTE2 B'00000010'
; cte de bit
#define LED1 PORTB,0 ;
; -----
;                               Definiciones de macros
; -----
BANK0 MACRO ; Banco 0
    BCF STATUS,RP0
ENDM
BANK1 MACRO ; Banco 1
    BSF STATUS,RP0
ENDM
```



```
; =====  
; Vectores  
; =====  
; Vector de reset  
ORG 0x000  
goto START  
; Vector de Interrupción  
ORG 0x004  
; comienzo del programa  
ORG 0x007  
; =====  
; TABLAS  
; =====  
; tabla de LED  
TABLA_LED    movwf TEMPTAB  
              movlw HIGH TABLA_LED      ; cargo el PCLACH con el BMS  
              movwf PCLATH              ; de la dirección actual  
              movf  TEMPTAB,w           ; le sumo al PCL el puerto al  
              addwf PCL,f                ; que tengo que acceder  
              retlw B'00011000'  
              retlw B'00100100'  
              retlw B'01000010'  
              retlw B'10000001'  
              retlw B'01000010'  
              retlw B'00100100'  
              retlw 0  
; =====  
; COMIENZO DEL PROGRAMA PRINCIPAL  
; =====  
; configurar el puerto A como salida  
START        BANK1  
              movlw B'00000000'  
              movwf TRISB  
              BANK0  
  
; bucle principal  
OTRO         clrf CONTADOR              ; borra contador  
PRINCIPAL    movf  CONTADOR,w           ; carga el contador al registro w p/  
              call TABLA_LED           ; usarlo como puntero de la tabla  
              andlw 0xFF                ; verific.si no llego al final ( W = 0 )  
              btfsc STATUS,Z           ; si W = 0 comienzo de nuevo  
              goto  OTRO                ; el valor leído de tabla pasa a PORTB  
              movwf PORTB              ; retardo  
              call  RETARDO             ; incremento el contador  
              incf  CONTADOR,f          ; incremento el contador  
              goto  PRINCIPAL  
  
; -----  
; Retardo Nivel 1  
; utilizo TEMP1_2 para guardar el valor del reg. w para recuperarlo  
; después, las demás variables son para realizar 2 bucles anidados  
; retardo = ( 3 ciclos * 256 + 3 ) * 256  
; -----  
RETARDO      movwf TEMP1_2  
              clrf  TEMP1_3  
RETloop2     clrf  TEMP1_1  
RETloop1     decfsz TEMP1_1,f  
              goto RETloop1  
              decfsz TEMP1_3,f  
              goto RETloop2  
              movf  TEMP1_2,w  
              return  
              end
```