

ÁREA TECNOLOGÍA ELECTRÓNICA  
DPTO. INGENIERÍA ELECTROMECÁNICA  
UNIVERSIDAD DE BURGOS

MIGUEL ÁNGEL LOZANO PÉREZ

ASIGNATURA:

**SISTEMAS BASADOS EN  
MICROPROCESADOR**

# **[GUÍA RÁPIDA DE MPLAB X]**

El contenido de esta guía introduce al alumno en la creación de un proyecto con un microcontroladores PIC, para la realización de programas en ensamblador, simulación, depuración y programación del uC.

<b>GUÍA RÁPIDA PARA CREAR UN PROYECTO EN MPLAB X.....</b>	<b>3</b>
1 PROYECTOS CON MPLAB® X. FLUJO DE TRABAJO .....	3
2 ABRIR EL PROGRAMA MPLAB X.....	3
3 CREAR UN PROYECTO NUEVO.....	4
3.1 Paso 1 .....	4
3.2 Paso 2 .....	4
3.3 Paso 3 .....	5
3.4 Paso 4 .....	5
3.5 Paso 5 .....	6
3.6 Paso 6 .....	6
3.7 Paso 7 .....	7
4 ESTRUCTURA DE DIRECTORIOS AUTOMÁTICAMENTE GENERADA .....	8
5 ESTRUCTURA DE PROYECTO MPLABX .....	9
6 TRABAJANDO CON EL EDITOR EN ENSAMBLADOR.....	9
6.1 Añadir ficheros fuente en ensamblador. ....	9
6.2 Programación en ensamblador.....	11
6.3 Estadísticas de compilación.....	13
7 DEBUG DEL PROGRAMA .....	14
7.1 Elementos para la simulación: .....	14
7.2 Pasos para la simulación.....	14
7.3 Ventanas de simulación. ....	15
7.4 Botones para la depuración y simulación.....	18
7.5 Depuración-Simulación. ....	18
7.6 Ejecución del programa.....	19
7.7 Puntos de ruptura. Breakpoint.....	20
7.8 Realización de estímulos a las entradas. ....	22
7.9 Analizador Lógico.....	23
7.10 Ventana del simulador de entradas y salidas: I/O Pin. ....	24
8 TRABAJANDO CON EL EDITOR EN LENGUAJE C. ....	25
8.1 Editar el programa. ....	25
8.2 Simulación del programa.....	26
9 OTRAS OPCIONES DEL MPLAB X .....	27



### 3 Crear un proyecto nuevo.

MPLAB X IDE se basa en proyectos, por lo que debe poner en marcha un proyecto para trabajar en su aplicación.

Los nuevos proyectos se pueden crear mediante la selección de:

- *Start* page, “*Learn & Discover*” tab, “Dive In” section, “Create New Project” link
- *File>New Project*.(or Ctrl+Shift+N)

El Asistente para nuevo proyecto se ejecuta para guiarle a través de un nuevo proyecto.

#### 3.1 Paso 1

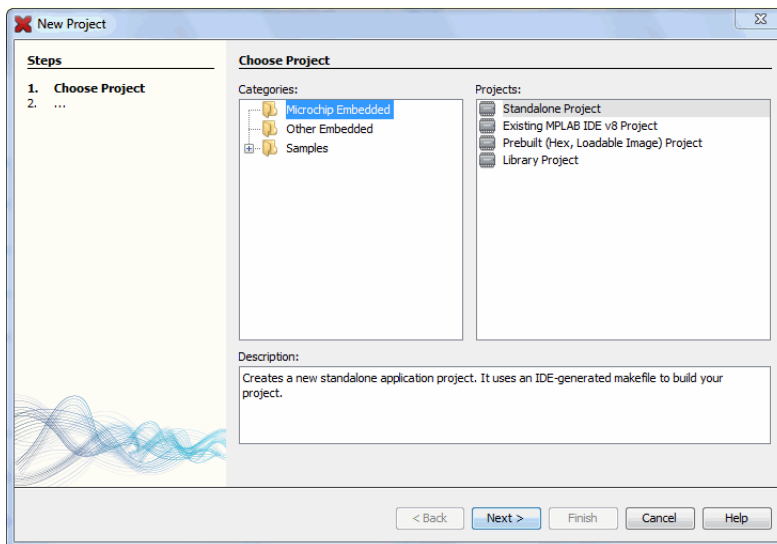
Paso 1 le pide que elija un tipo de categoría de proyecto.

Se trata de un diálogo de NetBeans. Para trabajar con productos Microchip, elija "microchip Embedded".

En segundo lugar, elija un tipo de proyecto. Para este tutorial, seleccione "proyecto independiente".

Clic **Next**> para pasar al siguiente diálogo.

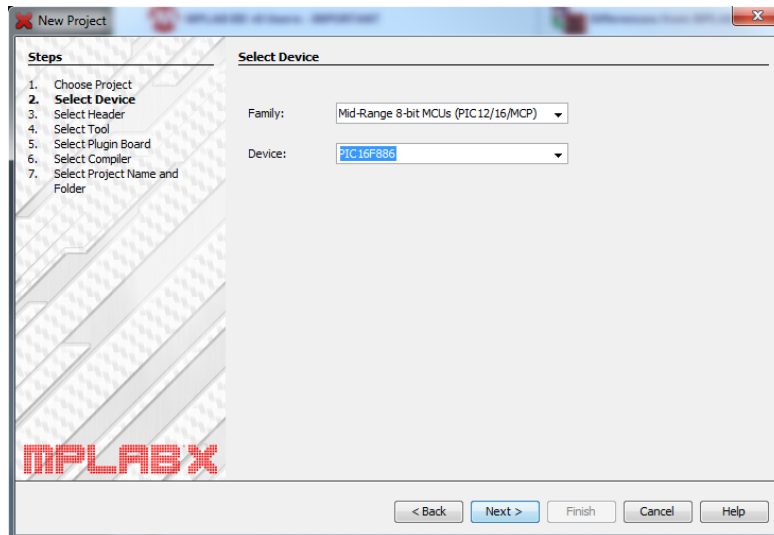
Figura: Asistente de Proyectos - Seleccione Proyecto



#### 3.2 Paso 2

El paso 2 es para la selección del dispositivo, en este caso PIC16F886. Cuando haya terminado, haga clic en **Next** >.

Figura: Asistente de proyecto - Seleccionar dispositivo



### 3.3 Paso 3

Paso 3 sólo aparece si está disponible una cabecera para el dispositivo seleccionado. Dado que no hay ningún encabezado para el dispositivo PIC16F886, MPLAB X IDE se salta este paso.

### 3.4 Paso 4

Paso 4 selecciona la herramienta.

Herramienta de apoyo para el dispositivo seleccionado está representada por los círculos de colores delante del nombre de la herramienta. El puntero del ratón sobre los círculos hace aparecer el texto acerca de la compatibilidad.

Light	Color	Support
●	Green	Full (Implemented and fully tested)
●	Yellow	Beta (Implemented but not fully tested)
●	Red	None

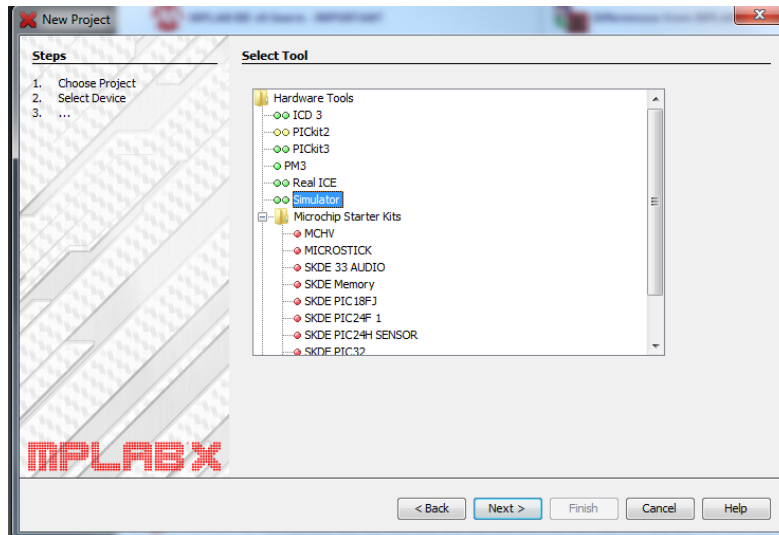
Para algunas herramientas, hay dos luces junto al nombre de la herramienta, en la primera luz es la luz más a la izquierda la luz, y el segundo está a la derecha de la primera.

Light No.	Debug Tools	Simulator
1	Debugger Support	Core (Instruction Set) Support
2	Programmer Support	Peripheral Support

Para las herramientas de hardware, te darás cuenta de que no se especifica un número de serie (SN) por debajo de cualquier herramienta que está conectado a su ordenador. Esto le permite seleccionar entre varias herramientas de hardware conectados.

Seleccione la herramienta y haga clic en **Next >**.

Figura: Asistente de Proyectos - Herramienta de selección



### 3.5 Paso 5

El paso 5 sólo aparece si se ha seleccionado MPLAB REAL ICE emulador en circuito como la herramienta.

Para el MPLAB REAL ICE en circuito emulador, se puede especificar una tabla de plug-in para su uso. Un tablero de plug-in es la placa de circuito que se inserta en la ranura de tarjeta de control del emulador.

Seleccione la herramienta y haga clic en **Next>**.

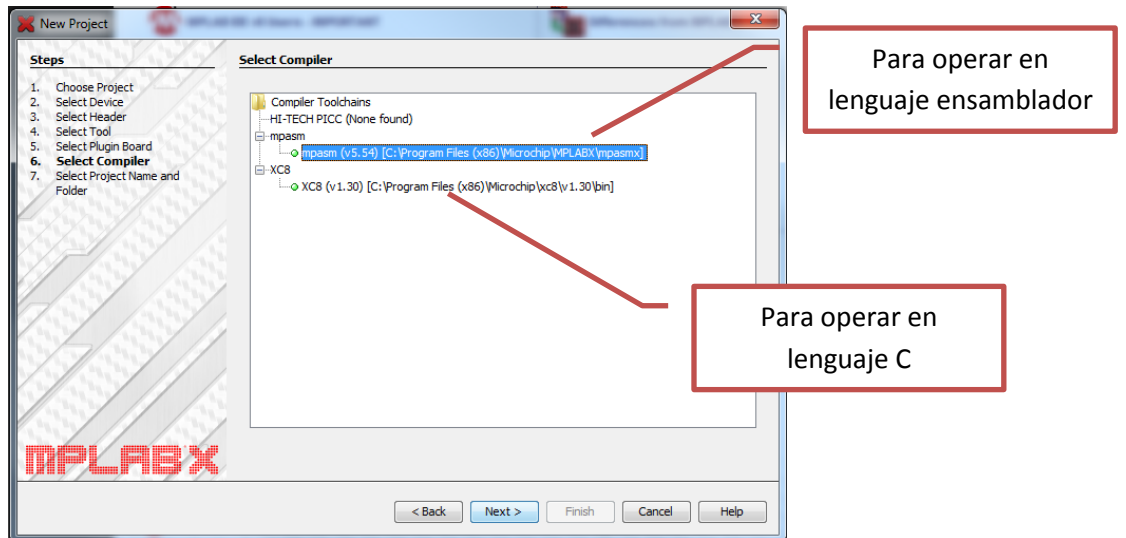
### 3.6 Paso 6

Paso 6 selecciona la herramienta del lenguaje de programación, ya sea un compilador C o ensamblador. Una vez más, el círculo de color delante del nombre del compilador significa el nivel de soporte de dispositivos. Pase el ratón sobre para el texto.

La versión y la ubicación de la instalación de una herramienta de idioma se muestran debajo de esa herramienta. Esto le permite seleccionar entre varias herramientas de idioma instalados.

Seleccione la herramienta y haga clic en **Next >**.

Figura: Asistente de Proyectos - Herramienta de selección del lenguaje de programación



### 3.7 Paso 7

El paso 7 selecciona el nombre del proyecto, ubicación y otras opciones de proyecto.

Introduzca el nombre del proyecto PRUEBA.

De forma predeterminada, los proyectos serán ubicados en:

- Windows XP -

C: \ Documents and Settings \ Usuario \ MPLABXProject

- Windows 7/8 -

C: \ Users \ Usuario \ MPLABXProjects

- Linux -

/ home / username / MPLABXProjects

- Mac -

/ Usuarios / nombre de usuario / MPLABXProjects

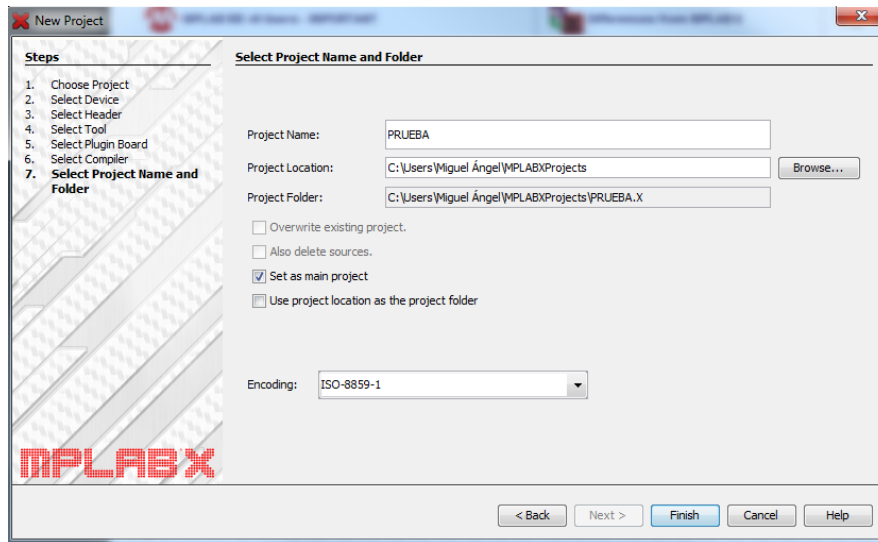
Si la ubicación del proyecto no apunta aquí, vaya a la ubicación adecuada.

**Marque "Establecer como proyecto principal" para hacer de este su proyecto principal.**

Este tutorial se realiza con la codificación establecida en la norma ISO-8859-1 (Latin 1), de modo que no es necesario cambiar esta configuración.

Cuando haya terminado, seleccione **Finish** para completar la creación de nuevos proyectos.

Figura: Asistente de Proyectos - Seleccione Nombre y Carpeta de proyecto



#### 4 Estructura de directorios automáticamente generada

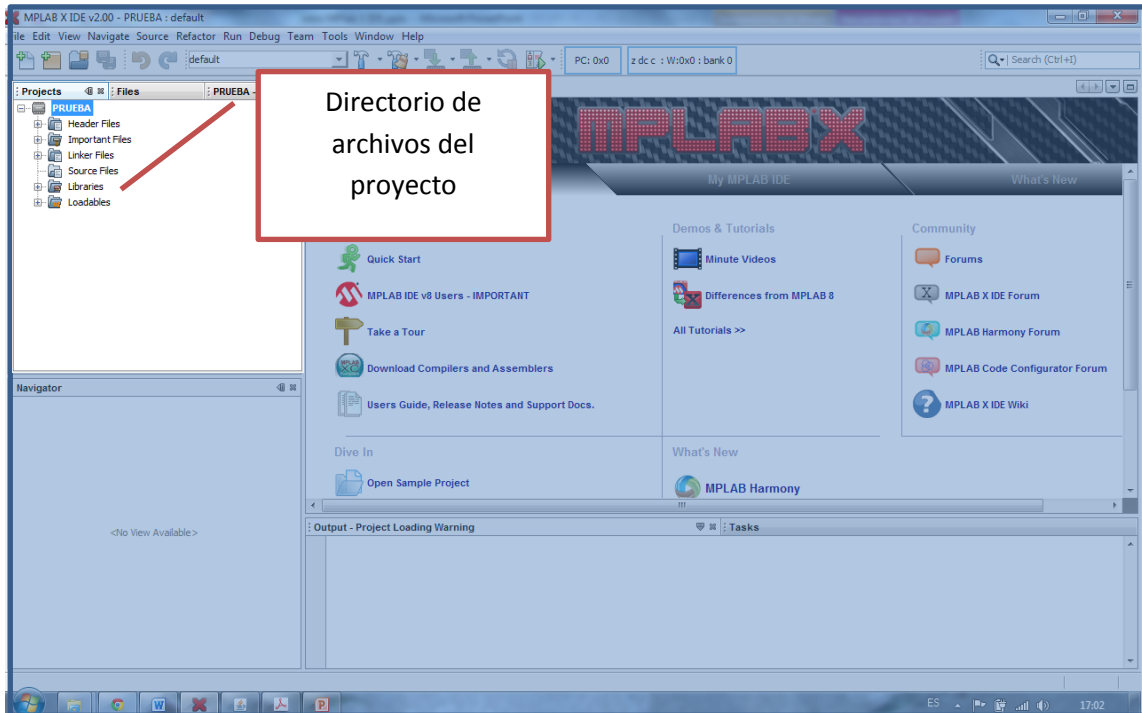
Un Proyecto está definido como un grupo de archivos dentro de una estructura específica de directorios y usada por el MPLAB X para almacenar y ordenar los archivos y configurar las aplicaciones.

Figura: Estructura de Proyectos - Nombres y ubicación de las carpetas de proyecto





## 5 Estructura de proyecto MPLABX



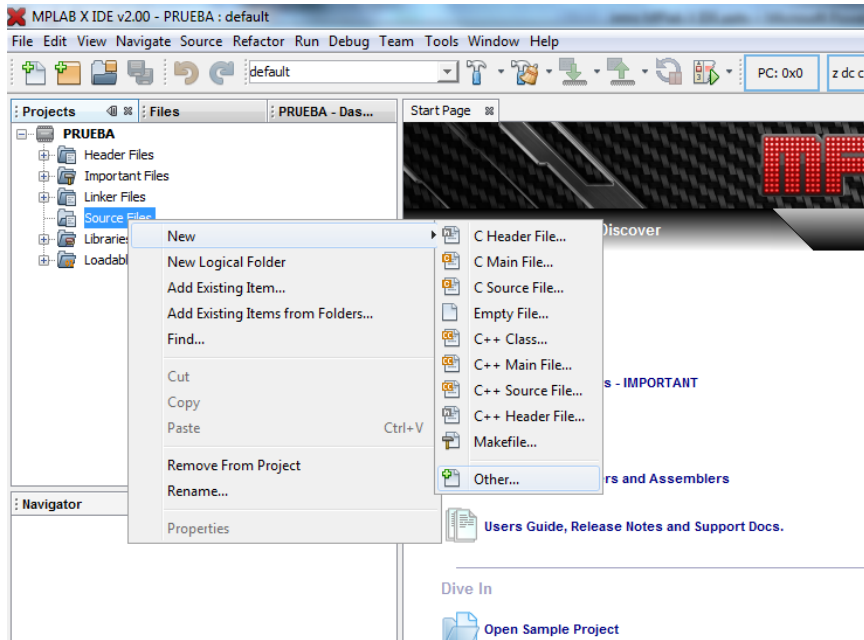
## 6 Trabajando con el Editor en ensamblador

Se puede elegir los ficheros que se desean programar en un lenguaje de programación en ensamblador, o bien programar en lenguaje C. La programación se realiza en los ficheros fuente (source) que se añaden sin más que hacer clic dentro de la ventana de proyecto en el directorio "source".

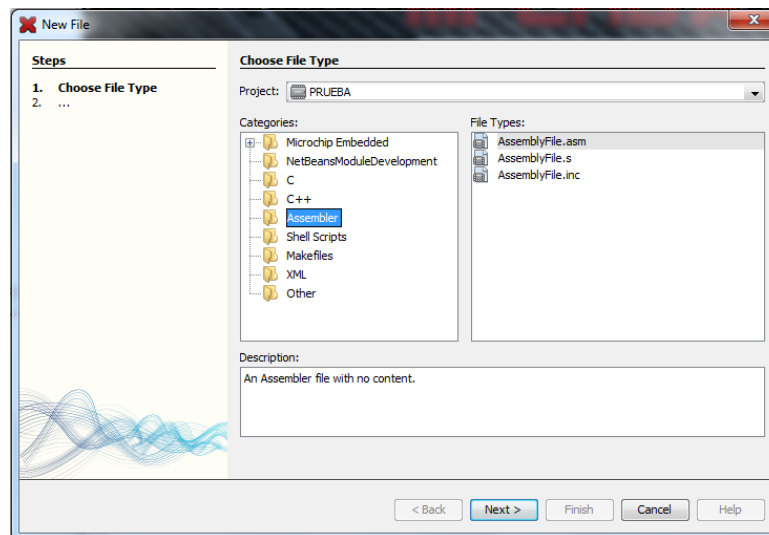
### 6.1 Añadir ficheros fuente en ensamblador.

Para insertar un fichero fuente en ensamblador, se realizan los siguientes pasos.

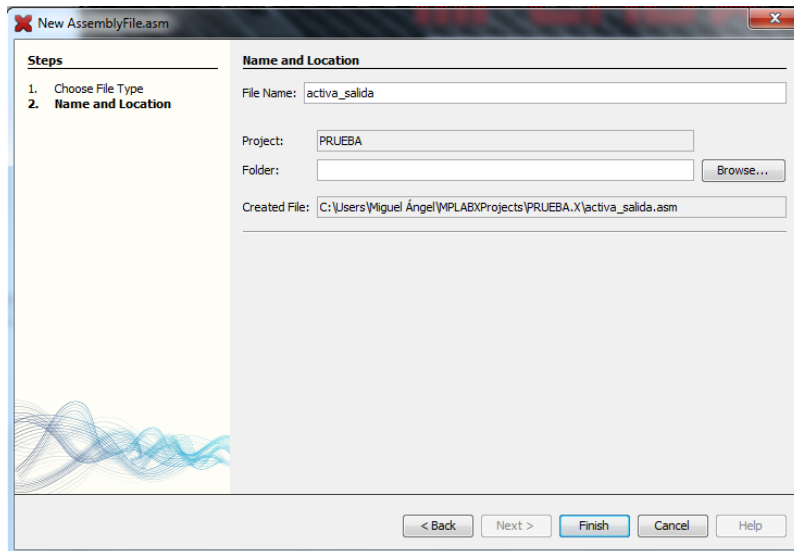
- Hacer clic con el botón derecho del ratón y elegir New> Other..



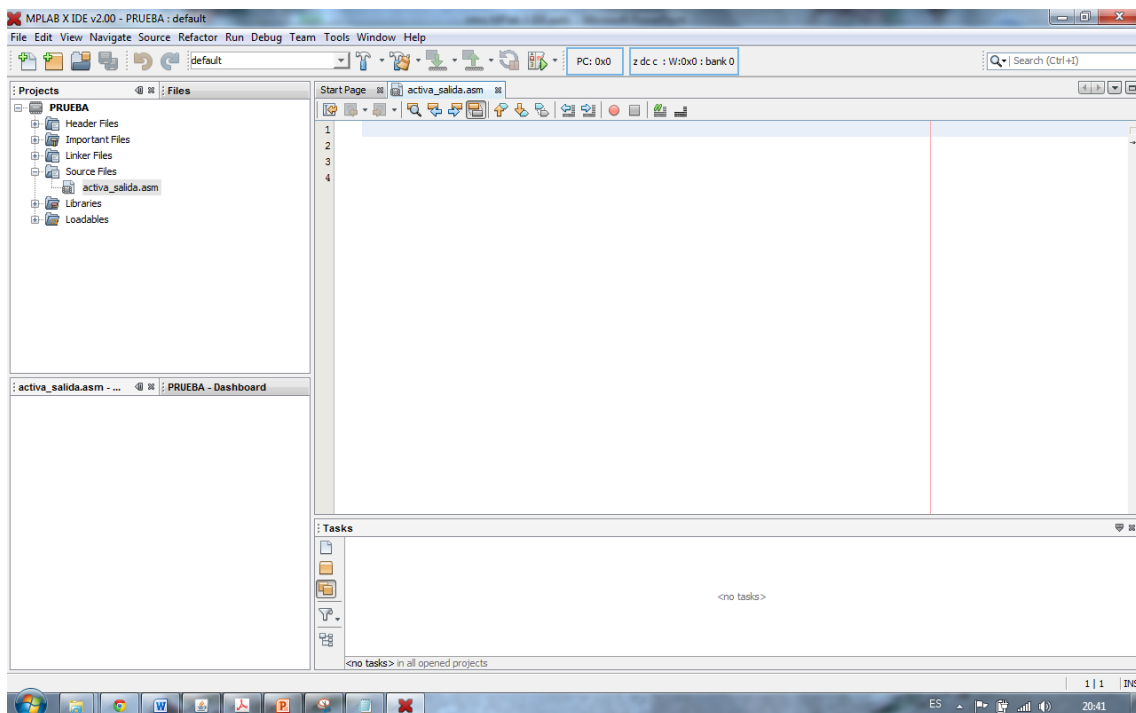
- Aparece la siguiente ventana.



- Elegir un fichero tipo “assembler”, con la extensión .asm y clic **Next>**
- Aparece la ventana con el nombre y la dirección del fichero fuente. Se elige un nombre que refleje lo que se pretende hacer en el programa para poder identificarlo claramente si se usara en otros proyectos.



- Elegir un nombre y hacer clic en **Finish** se abre la ventana del editor en ensamblador.



## 6.2 Programación en ensamblador.

Se escribe el programa en ensamblador, por ejemplo se incrementa el registro de trabajo W, sumando uno cada vez que se ejecuta la instrucción de suma por cada ejecución del bucle.

```

;*****
;Programa para realizar una prueba simple con el ensamblador
;MPLABX.
;Funcionamiento: Se trata de incrementar el registro de trabajo W
;
;Autor M.A. Lozano
;@ubu.es
;Fecha: 23/01/2014
;*****



; PIC16F84A Configuration Bit Settings
#include "p16F84A.inc"

; CONFIG
; __config 0xFFFB
__CONFIG _FOSC_EXTRC & _WDTE_OFF & _PWRTE_OFF & _CP_OFF
;Directiva de primera dirección del programa
org 0;
;programa
; <editor-fold defaultstate="collapsed" desc="bloque suma desde alt+intro">
    clrw                ; se borra el registro W
bucle    addlw 1        ; suma 1 al valor anterior que tenía el registro W->W+1
        goto  bucle
        end            ; indicación del fin del programa;</editor-fold>

```

### 6.2.1 Compilación del programa en ensamblador.

Compilar el proyecto significa que se asegure que toda la sintaxis está correcta.

	<p><b>Build Project</b> Compila sólo los archivos que cambiaron desde la última vez</p>
	<p><b>Clean and Build Project</b> Compila todos los archivos sin importar si han cambiado o no.</p>

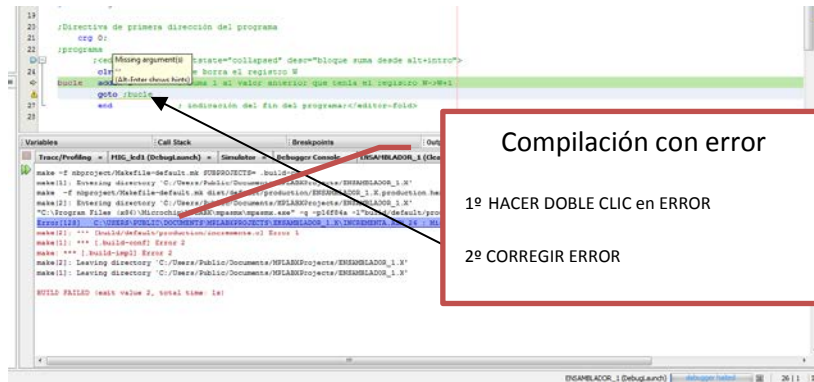
Haga clic en el ícono de **Clean and Build Project**. Este botón va a:

1. Borrar archivos temporales
2. Compilar el proyecto

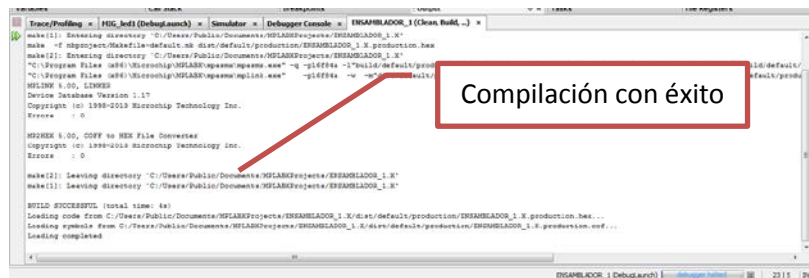
Al compilar se realiza la síntesis cuyo objetivo es generar un fichero objeto que suele estar formado por un código intermedio o código objeto y de Optimización de Código que busca obtener el código más eficiente.

Al compilar si aparece algún error, queda indicado y hay que realizar la interpretación del error y corregir el error. Y el resultado es una ventana en la que aparecen mensajes, errores y warnings.

Es útil comprobar el color que se utiliza para distinguir el código. En el caso de la figura se ha eliminado el salto al bucle, se ha convertido en comentario.



Si se cambia el punto y coma después de la etiqueta bucle, queda corregido y se compila con éxito.



### 6.3 Estadísticas de compilación.

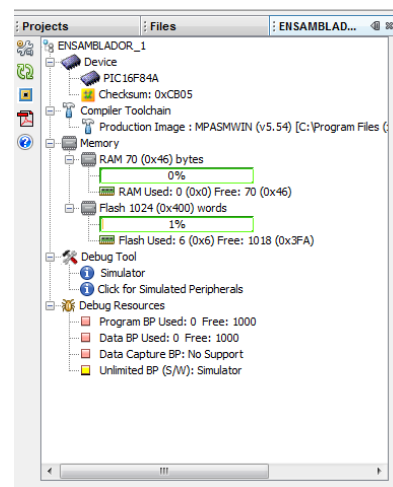
Seleccione su proyecto en el árbol de proyectos. En el menú seleccione **Window > Dashboard**

Vea la ventana dashboard, encontrará

- Dispositivo
- Checksum
- Uso de memoria RAM/FLASH
- Breakpoints disponibles
- Herramienta de debug

Además de una serie de botones a la izquierda con:

- Propiedades del proyecto
- Refrescar
- Toggle SW BP Simulator
- Datasheet del dispositivo
- Ayuda del compilador



## 7 *Debug del programa*

Para estudiar y controlar la ejecución de un programa se utiliza un simulador o debugger con ello, se puede detener y continuar la ejecución del programa y poder ver y modificar cualquier registro SFR o posición de memoria.

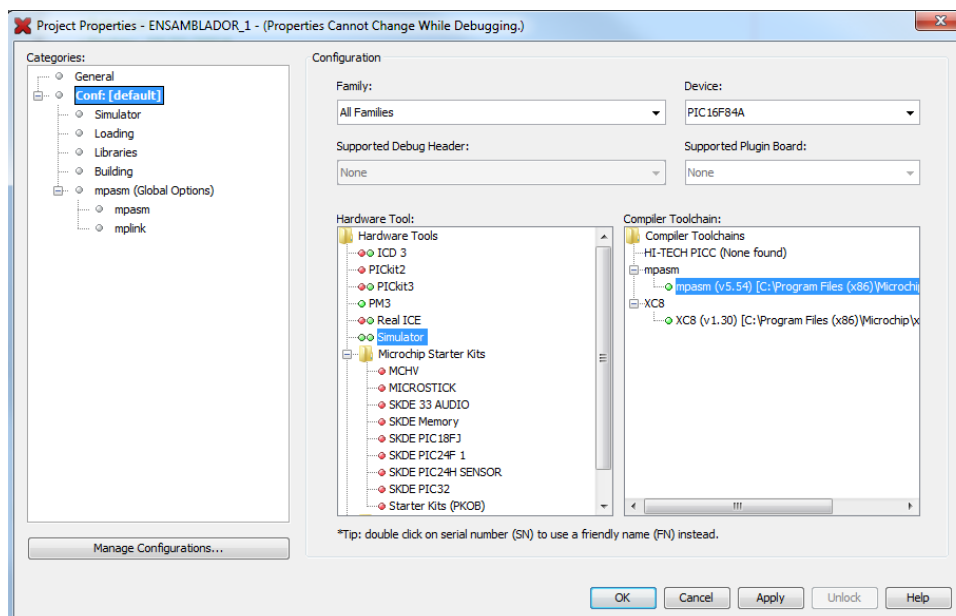
### 7.1 Elementos para la simulación:

- Comenzar una sesión de debug
- Controlar la ejecución del programa
- Ver registros especiales (SFRs)
- Observar o modificar posiciones de memoria
- Modificar memoria o SFRs
- Terminar la sesión de debug

### 7.2 Pasos para la simulación.

- 1 Seleccione una herramienta de Hardware o el simulador de software.

Desde la ventana Dashboard, o desde **File > Project Properties (nombre del proyecto)**, o bien desde el campo donde aparece default, elegir customize. Nos aparece la siguiente ventana.



Si no aparece en color desbloquear con el botón **unlock**.

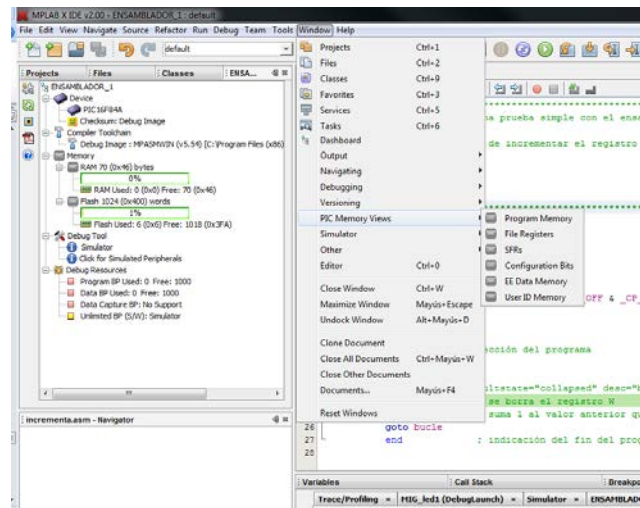
- 2 Elija un debugger de la columna del centro llamada Hardware Tools, en este caso se elige el simulador. Pulsar el botón **Apply** y luego **Ok**.

3 Haga click en el ícono **Debug Project**. Este botón va a:

- Compilar (**make**) su proyecto en modo **debug**.
- Programar el PIC en la placa destino, si ha sido seleccionada.
- Ejecutar el código

### 7.3 Ventanas de simulación.

Para la simulación del programa será necesario conocer la arquitectura interna del microcontrolador, esa información se dispone dentro del menú Windows, desde donde se abrirán las ventanas de memoria de programa y la de los registros de funciones especiales del microcontrolador (SFR's). Clic **Windows> PIC Memory Views>** Existen diversa opciones:

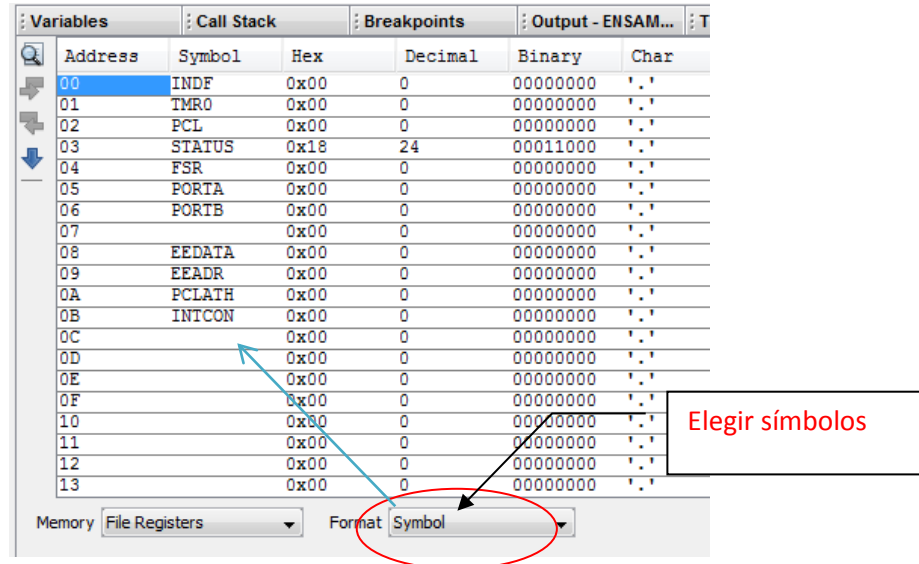


- **Program Memory:** Se visualiza la memoria de programa, desde aquí también se puede comprobar la ejecución del programa, aparte de ver el código generado que es el que posteriormente se envía al microcontrolador.

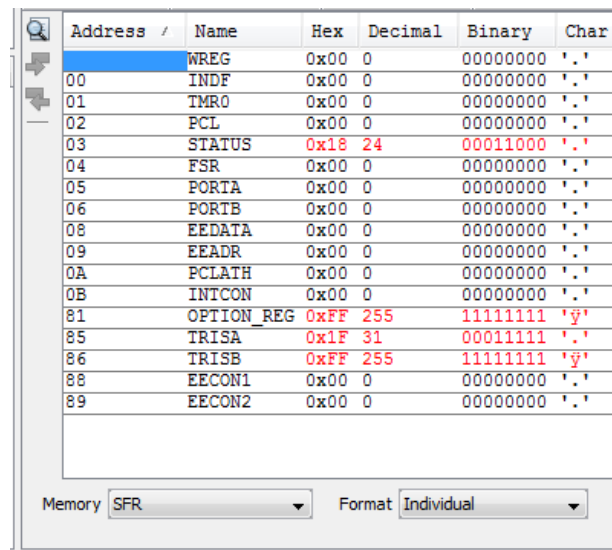
Line	Address	Opcode	Label	DisAssy
1	000	0103		CLRW
2	001	3E01	bucle	ADDLW 0x1
3	002	2801		GOTO 0x1
4	003	3400		RETLW 0x0
5	004	3400		RETLW 0x0
6	005	3FFF		ADDLW 0xFF
7	006	3FFF		ADDLW 0xFF
8	007	3FFF		ADDLW 0xFF
9	008	3FFF		ADDLW 0xFF
10	009	3FFF		ADDLW 0xFF
11	00A	3FFF		ADDLW 0xFF
12	00B	3FFF		ADDLW 0xFF
13	00C	3FFF		ADDLW 0xFF
14	00D	3FFF		ADDLW 0xFF
15	00E	3FFF		ADDLW 0xFF
16	00F	3FFF		ADDLW 0xFF
17	010	3FFF		ADDLW 0xFF
18	011	3FFF		ADDLW 0xFF
19	012	3FFF		ADDLW 0xFF
20	013	3FFF		ADDLW 0xFF

Elegir código

- **File registers:** Se visualizan los registros del programa, desde aquí también se puede comprobar la ejecución del programa, viendo los cambios producidos en los diferentes registros del microcontrolador y del usuario.



- **SFR's:** Se visualizan los registros especiales, de esta forma se pueden determinar las configuraciones necesarias del programa a realizar.



- **Bits de configuración (Configuration bits).** Se visualizan los parámetros de configuración mediante bits para elegir por ejemplo el tipo de oscilador, watch dog, protección del código, etc. Éstos depender del procesador utilizado.

Para obtener las palabras de configuración Windows> PIC Memory Views> Configuration bits.



- Se configuran los bits.
- Pulsar para generar los bits.
- Insertar los bits configurados después de la generación

3º Insertar los bits configurados después de la generación

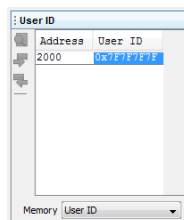
en la memoria tipo

1º Configurar los bits.










2º Generar los bits.

Ejemplo de configuración si se utiliza el PIC 16f886.

- **User ID Memory.** Se dispone de una serie de Byte para que el usuario deje su firma o identificación del proyecto que se está realizando.

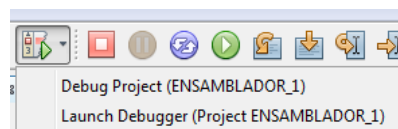


## 7.4 Botones para la depuración y simulación.

Función	MPLAB® X
Terminar sesión de debug	
Pause	
Ejecutar / Continuar	
Reset	
Step Over	
Step, paso a paso.	
Run desde el cursor	
Llevar el PC hasta al cursor	
Llevar al cursor hasta el PC	

## 7.5 Depuración-Simulación.

Desde la barra de herramientas se ha de seleccionar el botón de depuración, del cual se pueden obtener dos opciones. Al arrancar cualquiera de ellas aparecen los botones de simulación y depuración que ya se han indicado. De esta forma se pueden depurar los programas.



<b>Debug Project (nombre del proyecto)</b>	Compila, arranca el debugger (depurador) y se conecta al hardware o simulador. Si hay herramientas de hardware, programa al micro y ejecuta el código.
<b>Launch Debugger Project (nombre del proyecto)</b>	No compila, arranca el debugger (depurador) y se conecta al hardware o simulador

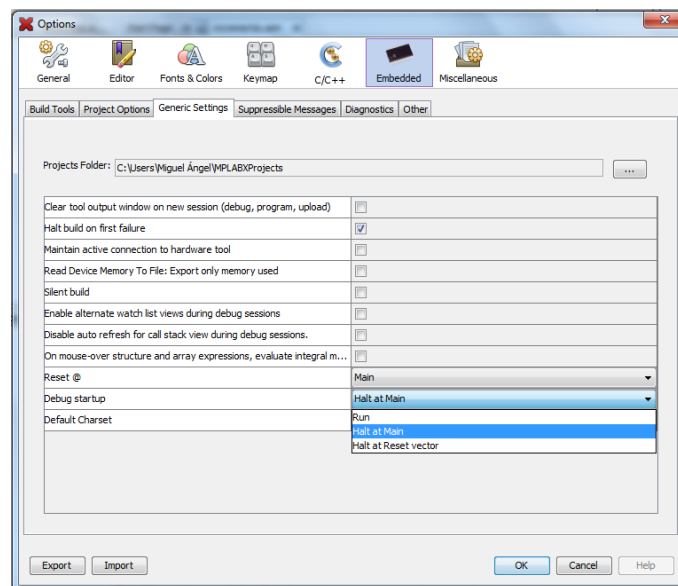


Al pulsar Debug Project se inicia la sesión de depuración. Si hay herramientas de hardware, programa al micro y ejecuta el código.



La sesión finaliza al pulsar el botón de stop. Si existe un programador hardware finaliza la comunicación USB. Debe ejecutarse este paso antes de modificar el código.

\* Para que el programa no se ejecute nada más arrancar el depurador, se puede cambiar en **Tools > Options**, Seleccione el ícono **Embedded**, luego seleccione la ficha **Generic Settings**. Para la configuración **Debug startup** seleccione Main o Reset vector en lugar de Run

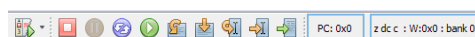


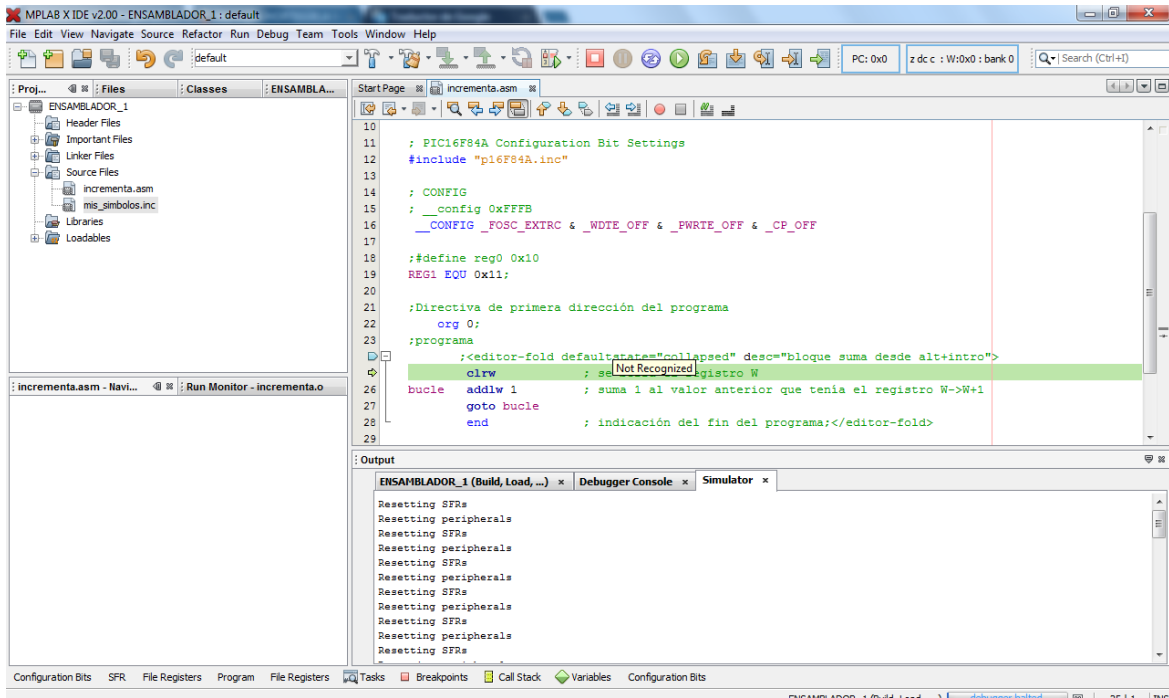
## 7.6 Ejecución del programa

Se pulsa Debug Project. Aparece la siguiente ventana y aparece el depurador con la herramienta de simulación.


El cursor, marcado en verde, apunta a la primera dirección de programa. Los pasos iniciales a seguir serían los siguientes:

1. Comprobar que el contador de programa está en 0x00.





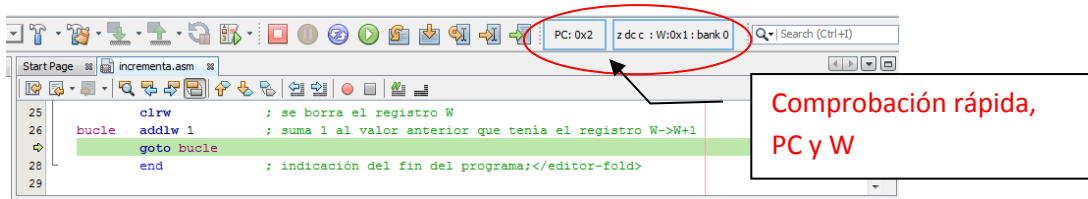
2. Se pueden observar otras condiciones de inicio como ciertos bits, el registro W, el banco actual de registros.

3. Para la ejecución paso a paso se debe pulsar 

```

25      ;<editor-fold defaultstate="collapsed" desc="bloque suma desde alt+intro">
26      clrw      ; se borra el registro W
27      bucle  addlw 1 ; suma 1 al valor anterior que tenía el registro W->W+1
28      goto bucle
29      end      ; indicación del fin del programa;</editor-fold>
    
```

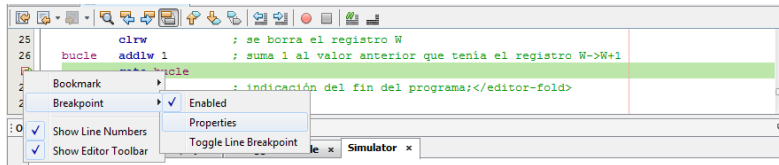
4. La instrucción `clrw` se ha ejecutado. Si se vuelve a pulsar se ejecuta la siguiente instrucción, y se pueden comprobar las variaciones en la memoria y en los registros, tal y como se ha indicado en el punto Ventanas de simulación.



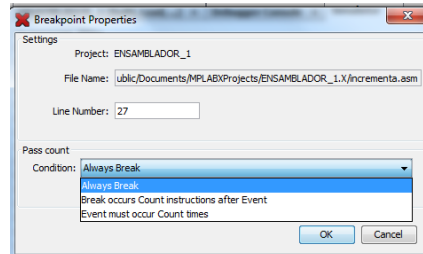
### 7.7 Puntos de ruptura. Breakpoint.

Cuando se ejecuta en modo Run se pueden disponer puntos de ruptura para parar la ejecución del programa en esa línea de programa (breakpoints).

- Creación de breakpoints. Sitúese sobre la línea de programa sobre la cual desea realizar la interrupción de la ejecución del programa. Haga clic sobre el número, aparecerá un cuadrado en rojo.



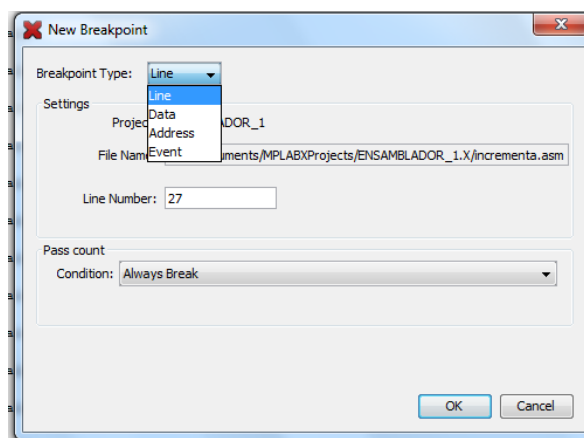
- Se puede elegir las siguientes acciones. Sitúese sobre el punto de ruptura y elija con el botón derecho del ratón las **propiedades**, son tres.



- Always break: Siempre se para la ejecución del programa.
- Breakpoint condicional de línea. Especifique el File y el Line Number (Se autocompleta en función de la posición del cursor) Especifique el Pass Count Condition (cantidad de ciclos).
- Seleccione Data como el tipo de breakpoint Especifique la dirección (Address) y la condicione Breaks on Especifique el Pass count Condition

Seleccione Debug >New Breakpoint. Doble click para ir a su fuente

Click derecho para: Crear un nuevo BP, habilitar todos los BP, deshabilitar todos los BP, borrar todos los BP, agrupar BP (Permite habilitar y deshabilitar por grupos), cambiar la configuración de un BP.



La configuración de un BP se establece: por la línea de programa, un dato, la dirección o un evento (como sleep).

## 7.8 Realización de estímulos a las entradas.

Para forzar valores de entrada se abre desde el menú **Windows -> Simulator -> Stimulus**, con ello se abrirá una ventana en la que se debe añadir los pines de entrada que se quieran estimular.

### Pestaña de estímulos asíncronos.

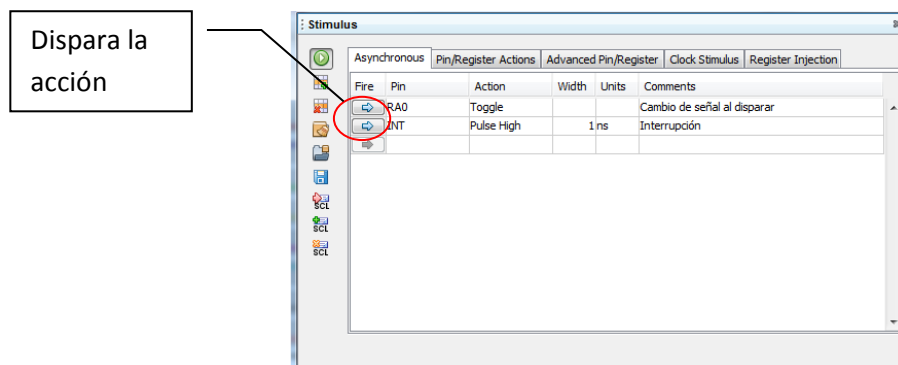
**Importante:** El diálogo Stimulus debe estar abierto para el estímulo que se activa durante la simulación.

Utilice la ficha *Asynch* para controlar eventos asíncronos generados por el usuario.

Introduzca cualquier señal asíncrona, fila por fila. Para eliminar una fila, seleccione la fila y haga clic en eliminar fila.

Hay dos tipos de estímulos: asíncrono de estímulo regular, para la mayoría de los pines / SFR y estímulo basado en mensajes, por SFR USART / UART.

En el momento que se desee activar una señal, se pulsa la flecha que realiza el disparo asíncrono de la señal existente en la fila, por ejemplo toggle que hace cambiar la señal de 0 a 1 o de 1 a 0, pulse genera un pulso durante un tiempo, o el forzado a nivel alto o bajo.



### Pestaña de Acciones Pin / Registro (estímulos síncronos).

Aquí se pueden introducir acciones síncronas básicas de pin y / o registros. Para acciones más complejas se utiliza la pestaña de Advanced Pin / Register.

Este es el estímulo más simple de tiempo. Algunos usos posibles para esta pestaña pueden ser:

- Inicialice estados para los pines en el instante de tiempo 0, para que , cuando cada vez que se vuelva a empezar una simulación se ejecuten los pins desde ese estado predeterminado, por ejemplo después de cada POR. Pines del puerto IO no cambian el

estado en el reinicio y el simulador comienza con el tratamiento de todos los pines IO como entradas de un estado cero.

- Establecer un registro con un valor en un momento específico.
- Establezca múltiples señales de interrupción al mismo tiempo para ver el efecto en el controlador de las interrupciones con prioridad.
- Crear un tren de pulsos con diferentes períodos y ancho de pulso en el tiempo, o una forma de onda irregular basado en tiempo de ejecución.
- Repetir una secuencia de eventos para la prueba de resistencia

Para introducir datos en esta ficha:

- Seleccione la unidad de tiempo en la " casilla de unidades de tiempo " lista que va a utilizar para desencadenar todos los estímulos.
- Haga clic en el texto que dice "Haga clic aquí para añadir señales " para abrir este cuadro de diálogo **Add/Remove Pin/Registers Dialog**). En ese cuadro de diálogo, selecciona los pines, registros u otras señales a las que se aplicará estímulo. Estas selecciones se convertirán en los títulos de las columnas.
- Rellene cada fila, entrando el tiempo de activación ( "Time" ) y el valor de cada pin / Registro columna. El tiempo de activación para cada fila es tiempo acumulativo ( desde el inicio de la simulación ), no es el tiempo de intervalo entre filas adyacentes .
- Para eliminar una fila, selecciónela y haga clic en Eliminar fila .
- Marque la casilla de verificación " Repite después de X (decimal) " para repetir el estímulo de la ficha después del último estímulo. Especifique un intervalo de retardo para cuando repetir el estímulo.
- Para reiniciar en un momento determinado, puede hacer una selección de la "Reiniciar en: (decimal) " cuadro de lista. Las selecciones del cuadro de lista se determinan los tiempos de activación (en la columna Time ) para cada fila.





Una vez que la ficha se rellena , se puede proceder a otra ficha o haga clic en

Aplicar para utilizar el estímulo. Para eliminar un estímulo aplicado previamente , haga clic en Eliminar .

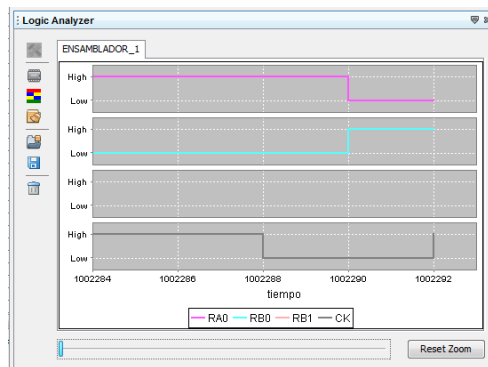
## **7.9 Analizador Lógico.**

Para utilizar el analizador lógico se abre desde el menú **Windows ->Simulator -> Analyzer**, se abrirá una ventana.

Se pueden incluir señales y buses.

	Ajustes
	Añade señales relativas a un pin.
	Añadir grupos
	Elimina las señales añadidas

Una vez añadidas las señales aparece una ventana como la que sigue.



## 7.10 Ventana del simulador de entradas y salidas: I/O Pin.

Para utilizar el analizador lógico se abre desde el menú **Windows -> Simulator -> IOPin**, se abrirá la ventana de pines de E / S es útil durante la depuración de un proyecto con el simulador. La pantalla muestra el estado actual de un pin y permite elegir el pin que se desea visualizar.

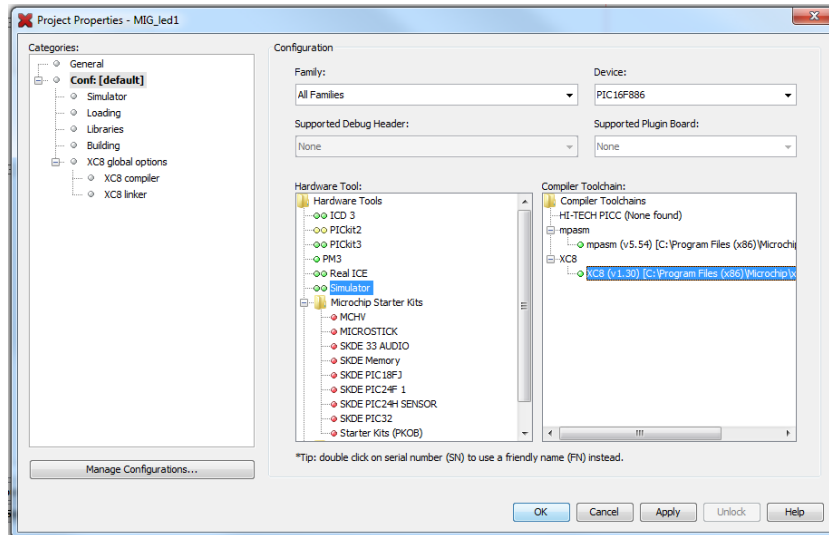
Durante una sesión de depuración, los valores se visualizan. Al ejecutar el programa, fundamentalmente en la inicialización va apareciendo la función que se ha asignado a ese pin, por ejemplo, si es función digital de entrada pone Din, si es analógica pone Ain los valores se muestran en la ventana I / O Pin, cambiando en consecuencia.

Pin	Mode	Value	Owner or Mapping
RA0	Din	0	RA0/AN0/ULPWJ/C12IN0-
RB0	Dout	0	RB0/AN12/INT
RB1	Dout	1	RB1/AN10/P1C/C12IN3-
RB2	Dout	0	RB2/AN8/P1B
RB2	Dout	0	RB2/AN8/P1B
RB3	Dout	0	RB3/AN9/PGM/C12IN2-
RB4	Dout	0	RB4/AN11/P1D
RB5	Dout	0	RB5/AN13/T1G
RB6	Dout	0	RB6/ICSPCLK
RB7	Dout	0	RB7/ICSPDAT
<New Pin>			



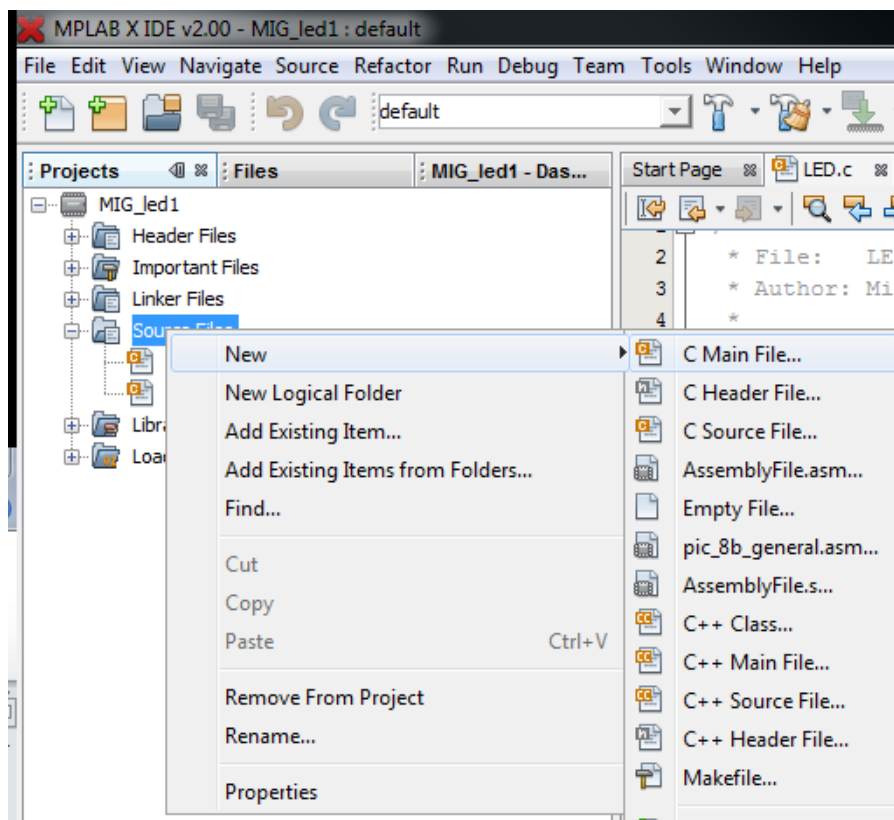
## 8 Trabajando con el Editor en lenguaje C.

Antes de operar en lenguaje C, a la hora de crear el proyecto se siguen los mismos pasos que en ensamblador, con la excepción de que se debe de elegir el compilador XC8 como se indica en la figura.

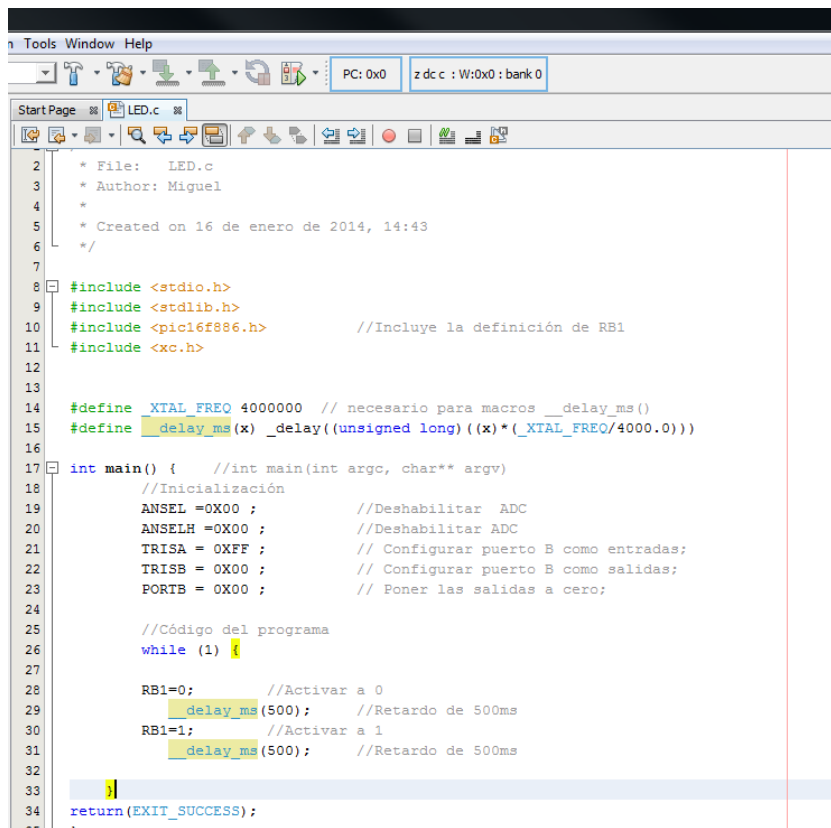


### 8.1 Editar el programa.

Desde el navegador de proyecto, se debe de elegir la carpeta de ficheros fuente, y pulsando el botón derecho del ratón se inserta el fichero principal en C, tal y como se indica en la figura.



Posteriormente se incluye el código del programa. En el ejemplo se ha creado un programa que realiza la intermitencia en un pin de salida del microcontrolador. Si se conectara un led se produciría una intermitencia a una frecuencia de 1Hz.



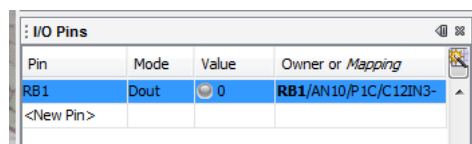
```
2  * File: LED.c
3  * Author: Miguel
4  *
5  * Created on 16 de enero de 2014, 14:43
6  */
7
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <pic16f886.h> //Incluye la definición de RB1
11 #include <xc.h>
12
13
14 #define XTAL_FREQ 4000000 // necesario para macros __delay_ms()
15 #define delay_ms(x) _delay((unsigned long)((x)*(_XTAL_FREQ/4000.0)))
16
17 int main() { //int main(int argc, char** argv)
18 //Inicialización
19 ANSEL =0X00 ; //Deshabilitar ADC
20 ANSELH =0X00 ; //Deshabilitar ADC
21 TRISA = 0XFF ; // Configurar puerto B como entradas;
22 TRISB = 0X00 ; // Configurar puerto B como salidas;
23 PORTB = 0X00 ; // Poner las salidas a cero;
24
25 //Código del programa
26 while (1) {
27
28 RB1=0; //Activar a 0
29 delay_ms(500); //Retardo de 500ms
30 RB1=1; //Activar a 1
31 delay_ms(500); //Retardo de 500ms
32
33 }
34 return(EXIT_SUCCESS);
35 }
```

## 8.2 Simulación del programa.

Las mismas técnicas de simulación que se han utilizado en ensamblador se pueden realizar aquí. Atendiendo al tipo de simulación se utilizarán unas u otras. Se activará

### 8.2.1 Simulación con la ventana I/O pin.

Para utilizar el analizador lógico se abre desde el menú **Windows ->Simulator ->IOPin**, se abrirá la ventana de pines de E / S es útil durante la depuración de un proyecto con el simulador. La pantalla muestra el estado actual de un pin y permite elegir el pin que se desea visualizar. Se comprueba como se produce la intermitencia.



## 9 Otras opciones del MPLAB X

Es interesante también comprobar qué opciones contiene el Project Properties. Se puede acceder desde **File-> Project Properties (nombre)**. Por ejemplo, se puede cambiar el dispositivo, el emulador, el simulador y los compiladores. Es posible elegir el tipo de microprocesador sobre el que simular y **activar el modo de simulación (MPLAB X-SIM simulator)**, no siempre activo por defecto.

Con **Project** y sus **Select languages tools** suites podrá cambiar la forma de programar con el editor (por ejemplo, cambiar de ensamblador a C, si tiene el C).

Desde **Windows->PIC Memory views-> Configuration bits** se podrán cambiar las opciones de reset, reloj, WatchDog y otras opciones que nos ofrece cada microcontrolador. Se pueden configurar y luego generar (dar el botón Generate Source Code to output) las palabras de configuración que se necesitan para el micro. En éste ejemplo se ha añadido en el fichero fuses.c, sólo se han de copiar y pegar en el fichero.

