


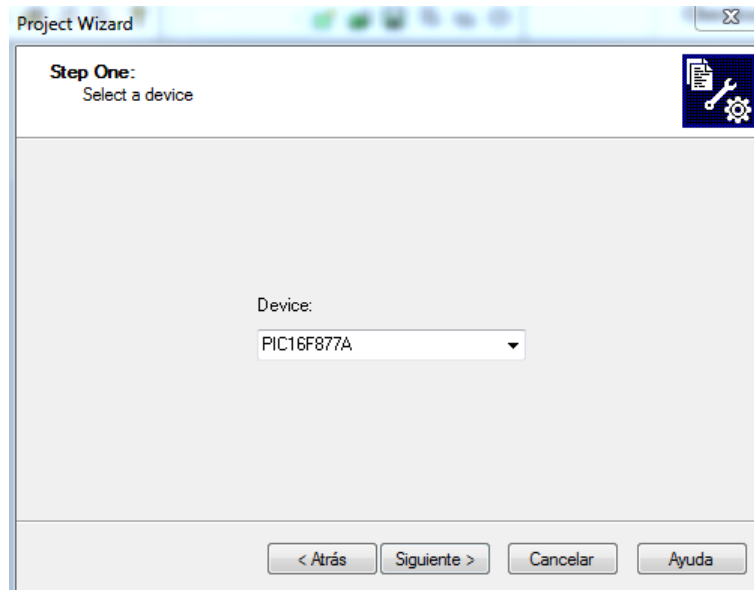
Entorno de trabajo MPLAB (v8.33)



- El MPLAB IDE es un **entorno de desarrollo integrado** distribuido gratuitamente por Microchip (fabricante de los microcontroladores PIC) en su página web: <http://www.microchip.com>
- Incluye un **editor de texto**, varias herramientas de programación como: un ensamblador (**MPASM**), compilador de C (**CCS C Compiler**), un simulador (**MPLAB SIM**), desarrolladores, debugger, programadores y un **gestor de proyectos** (entre otras cosas).
- A continuación se indican los **pasos a dar** para trabajar con este entorno (versión 8.33) hasta conseguir simular el funcionamiento de un programa.
- El fichero con el que se va a trabajar es **cuentaRA4.c**, primer ejemplo básico. Se realizará su **edición**, se definirá un **Proyecto** que incluirá como código fuente el programa editado y se realizará la compilación del mismo.
- Tras la simulación se procederá a comprobar el correcto funcionamiento del mismo utilizando el **depurador en circuito MPLAB ICD2** junto con la **tarjeta de entrenamiento PICDEM 2** (o **PICDEM 2 plus** según el caso).
- Por último, se procederá a **grabar este programa** en un microcontrolador PIC usando también el **MPLAB ICD2**.

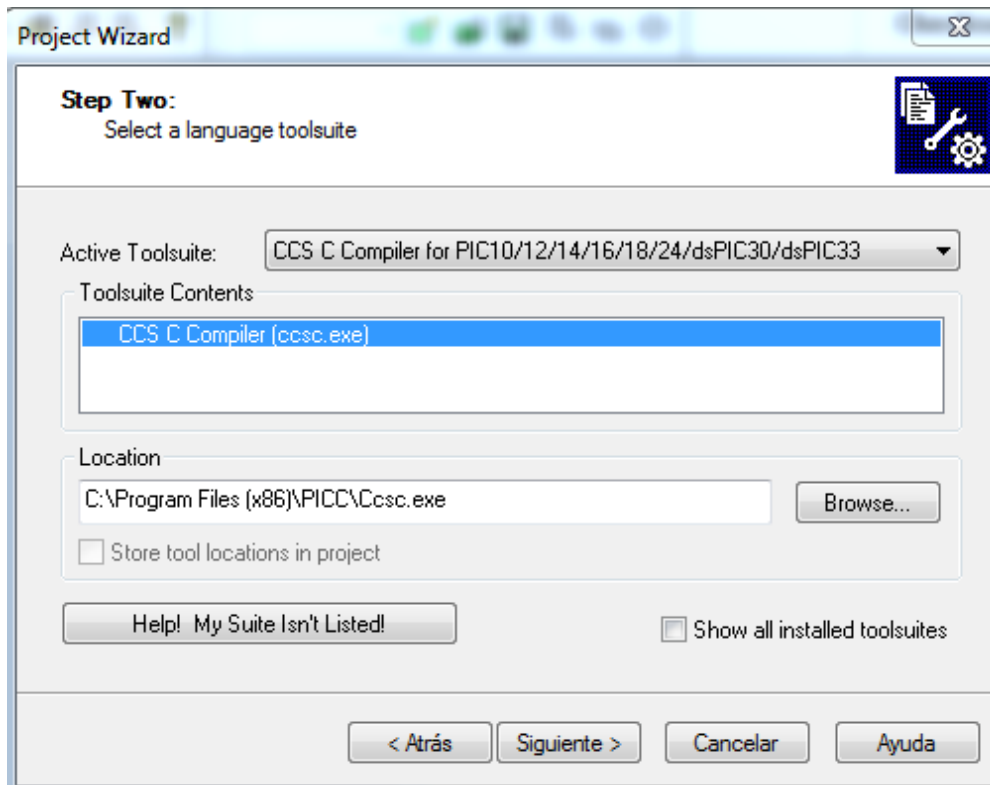
Primeros pasos

- Iniciar el programa haciendo click sobre el **icono** correspondiente. 
- Como método para crear proyectos se sugiere utilizar el asistente, para lo cual se seleccionará **Project >>Project Wizard...**
- Tras un mensaje de bienvenida, se nos solicita que indiquemos el **dispositivo** que pensamos utilizar. En este caso seleccionaremos el **PIC16F877**.

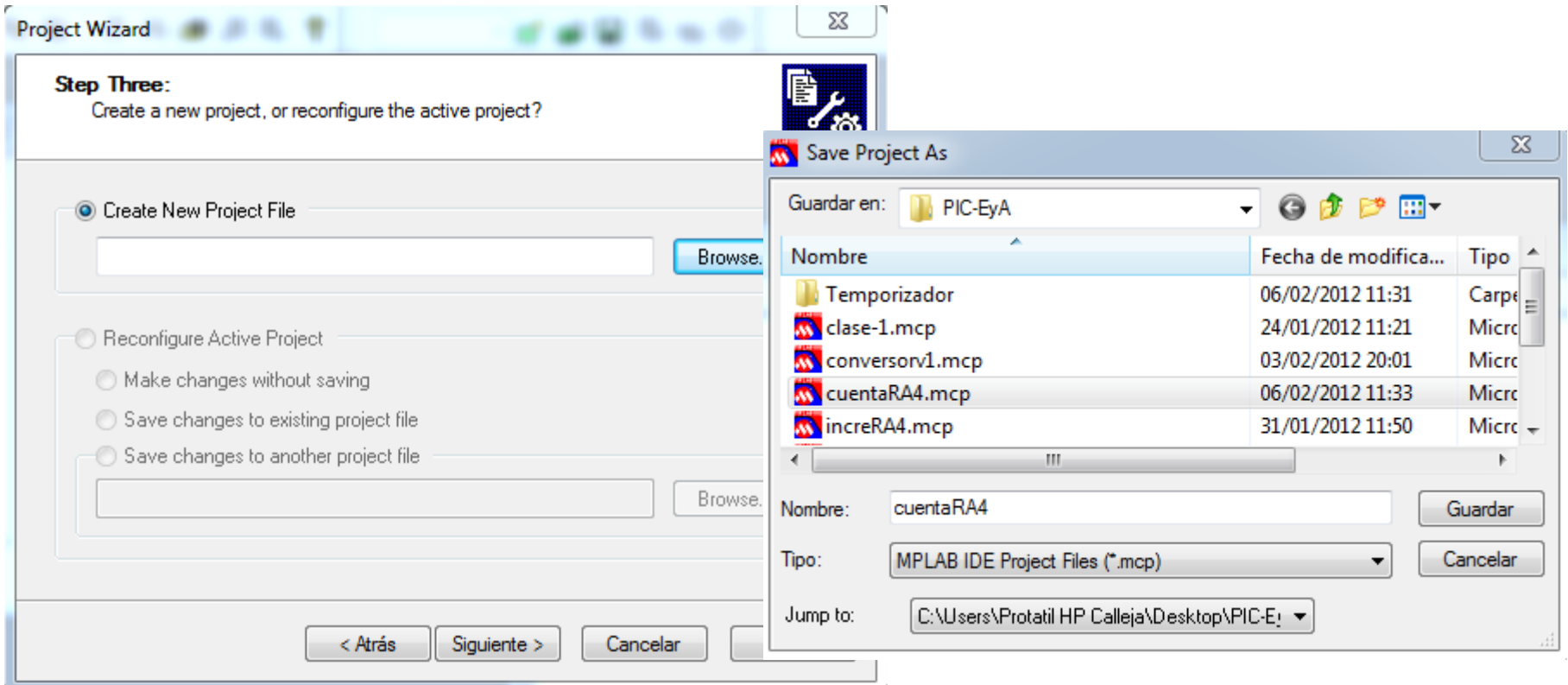


• La siguiente ventana pregunta por el **ensamblador/compilador que se va a usar**.

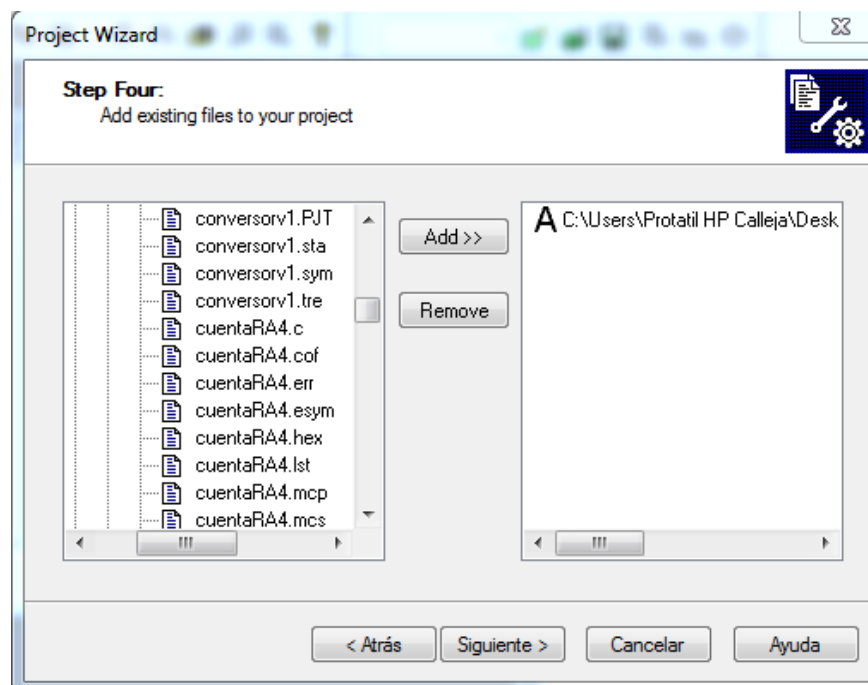
Seleccionaremos Microchip **CCS C Compiler (ccsc.exe)** asegurándonos de que el entorno MPLAB conoce la ubicación del ejecutable que Necesita (ccsc.exe)



- A continuación se le asignará un **nombre de proyecto** (por ejemplo **cuentaRA4**, sin extensión) y se indicará en qué **directorio** se va a ubicar. Se recomienda que el proyecto y el fichero fuente del programa en C se encuentren en el mismo directorio.

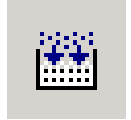


•En el cuarto paso se nos pide indicar los **ficheros que se van a incluir en el proyecto** que estamos creando. Debemos buscar el fichero **cuentaRA4.c**, seleccionarlo y hacer click sobre el botón **Add >>**. Si no hemos creado aún el fichero *.asm, simplemente haremos click sobre **Siguiente >**. Con esto ya queda creado el proyecto, y en el entorno MPLAB aparece una ventana (**cuenta.mcp**) en la que se muestra la información del proyecto que se acaba de definir.



Compliado del programa

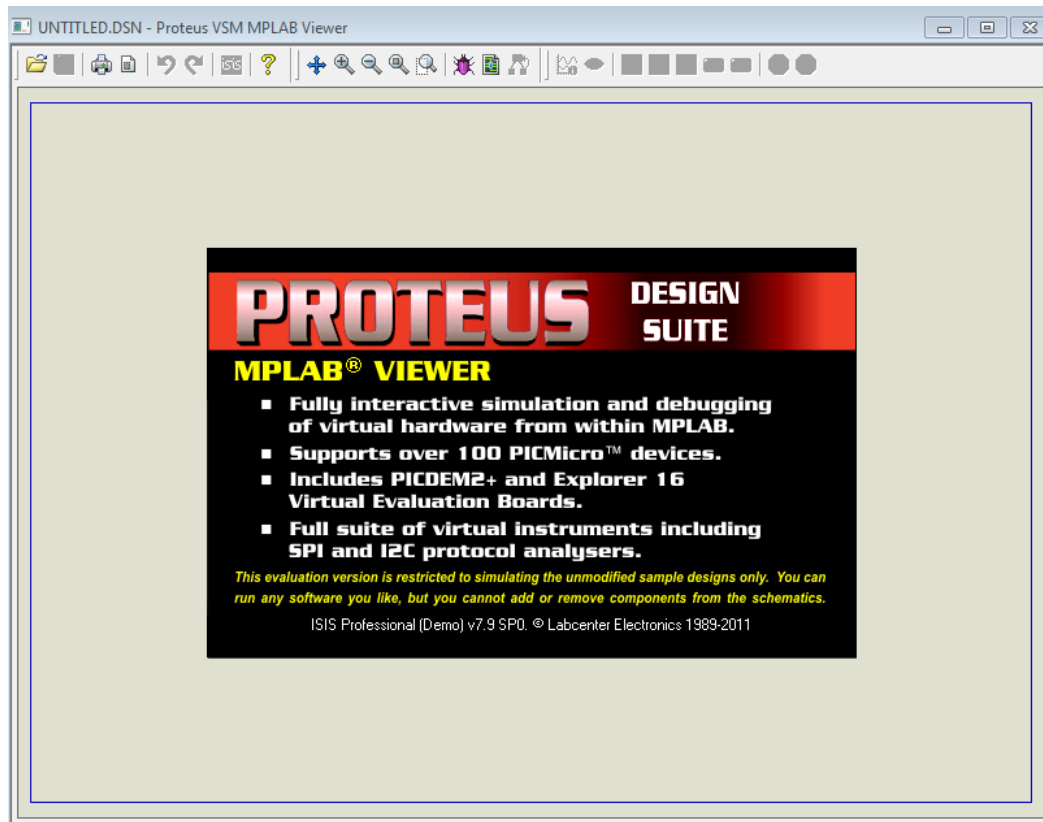
- Para complilar el programa basta con hacer doble click en el **icono** correspondiente, que equivale a seleccionar **Project > >Build All**.



- En esta etapa se realiza el **compliado** del fichero fuente y el traspaso de éste a la **memoria de simulación**.
- Aparecerá momentáneamente una ventana indicando el **proceso de compliado**
- Al final aparece la pantalla **Output** en la que se indica si el compliado se ha llevado a cabo con éxito (**BUILD SUCCEEDED**) o si, por el contrario, se han localizado fallos (**BUILD FAILED**).
- En caso de existir fallos, se indica de qué tipo son y en qué línea están. Haciendo doble click sobre la línea de la ventana Output en la que se muestra esta información, se accede directamente a la posición donde se ha detectado el fallo en el fichero fuente.

Simulación del programa

- Una vez ensamblado el programan **sin errores**, simularemos su comportamiento seleccionando la opción **Debugger > > Select Tool ► PROTEUS VSM**.
- Aparecerá una pantalla como la que se muestra a continuación.



- Para poder simular el programa debemos cargar el fichero de diseño correspondiente a la placa PICDEM 2 PLUS. Para ello tendremos que buscar el archivo: **PICDEM2+with 16f877.DNS**

- Para encontrar dicho archivo seguiremos el siguiente:

- C: Archivos de programa

- Labcenter Electronic

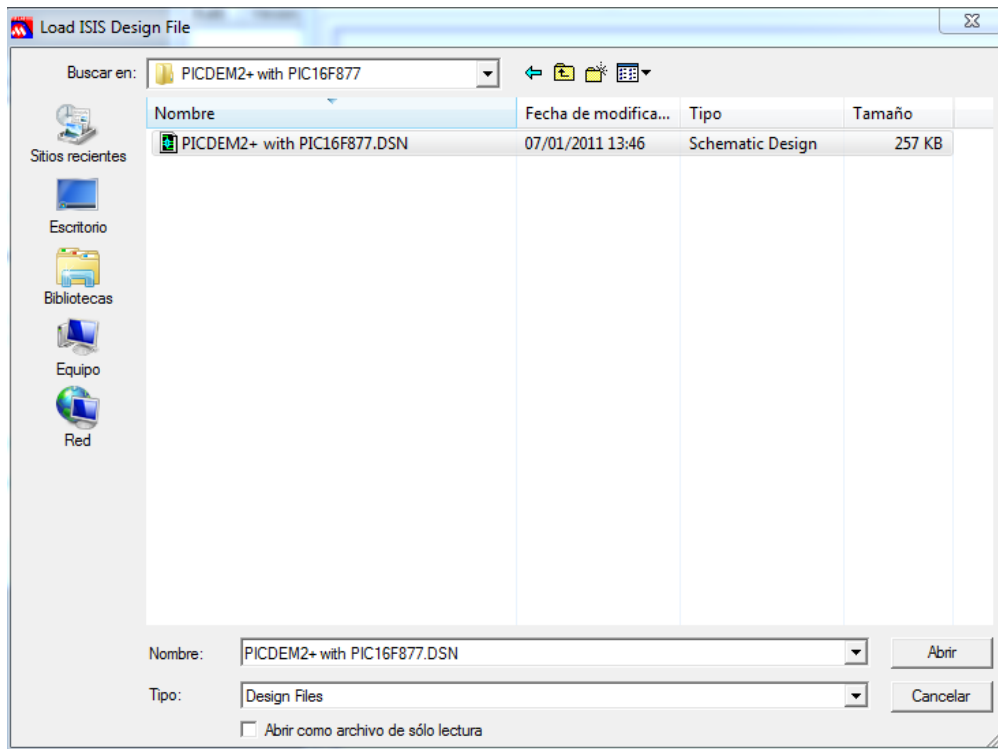
- Proteus Demostration

- Samples

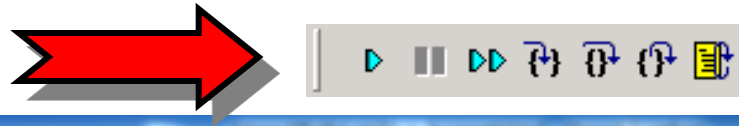
- VMS for PICMICROS

- Eval Borads

- PICDEM2.....



- Ejecutamos la simulación pulsando la opción paso a paso (animate) o bien RUN



The screenshot displays the MPLAB IDE v8.83 environment. On the left, the 'cuentaRA4.c' source file is open, showing C code for a PIC16F877A microcontroller. The code includes a delay function, fuse settings, and a main loop that increments a counter and toggles LEDs. The right pane shows the Proteus VSM simulation of the PICDEM2+ board with the PIC16F877A. The circuit includes an LCD1, five LEDs (D1-D5), a speaker (LS1), a frequency meter, and various passive components like resistors and capacitors. The Proteus logo and 'PICDEM2+ with PIC16F877' are visible at the bottom of the simulation window.

```
/* PRÁCTICA 1 */
#include "16F877A.h" // Incluye fichero de simbolo
#define delay (clock=4000000) // Define frecuencia del oscilador

#define XT, NOWDT, PUT, BROWNOUT, NOPROTECT, NOLVP, NOCPD, NOWRT

#define PORTB = 0x06 // Se definen los puertos
#define PORTA = 0x05 // y bits a utilizar.
#define RA4 = PORTA.4

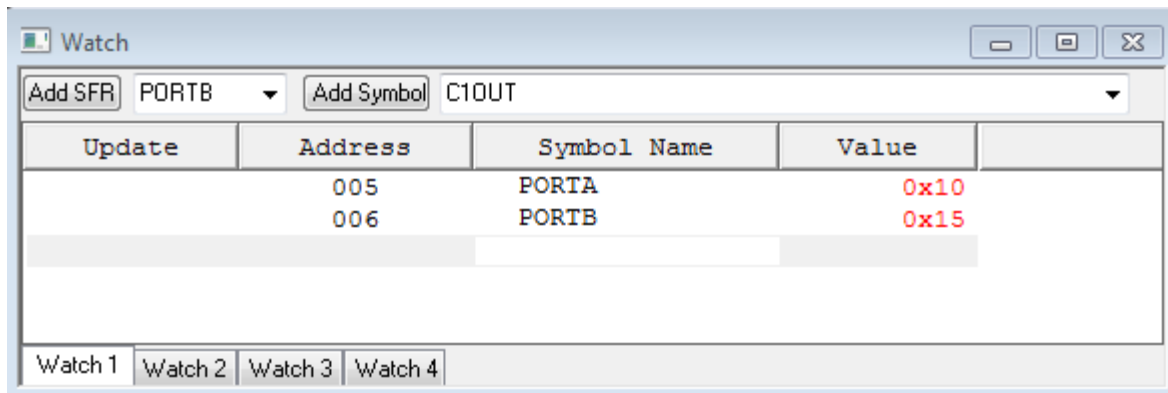
main() /* Comienza el programa principal */
{
    int contad; /* Contador del número de pulsaciones */

    PORTB=0; // Inicialmente los LEDs están apagados
}
```

• Se pueden visualizar resultados de simulación. Para ello, se activará una **ventana personalizada Watch** que permite supervisar el contenido de los registros de interés. Esta ventana se activa mediante **View > Watch**. Los registros se añaden haciendo click en el botón **Add SFR** o **Add Symbol**.


• En la barra de estado también se muestra información sobre el contenido del **registro W** y el valor de los **flags de STATUS** (minúscula = '0').

W:0 z dc c



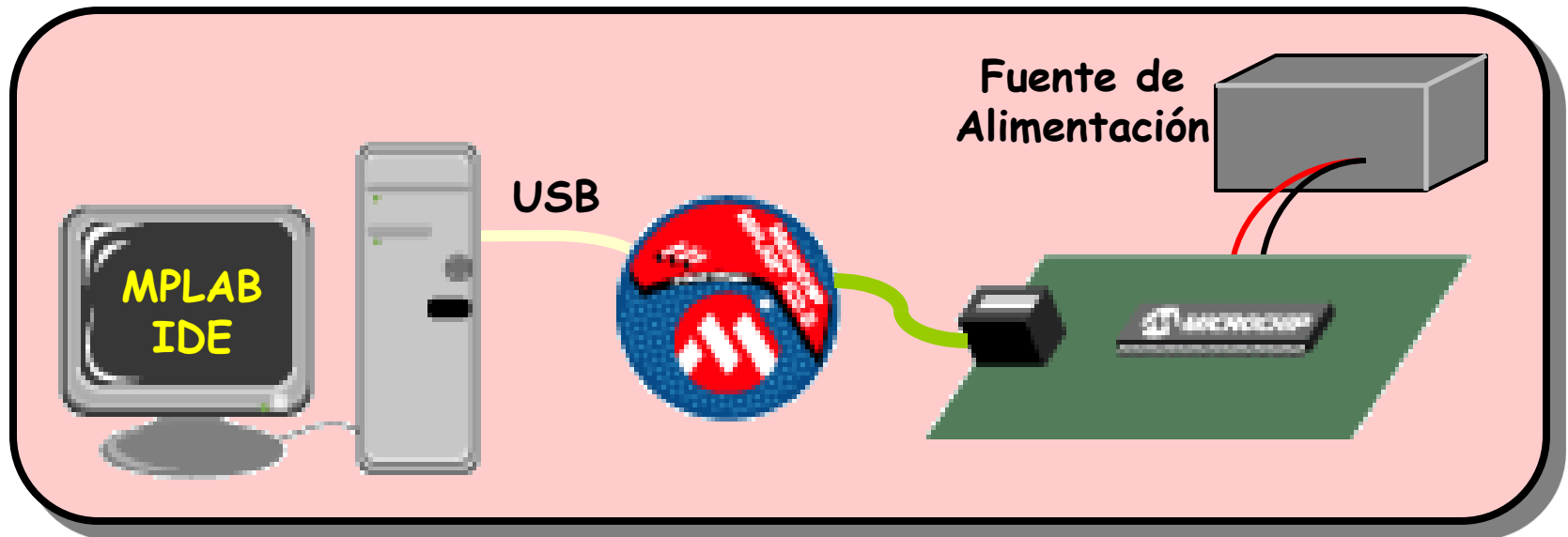
• Los cinco comandos más importantes para la simulación (localizados dentro del menú **Debugger**) son los que se indican a continuación.

- ▶ **Run** (**Ejecución continua**). Ejecuta el programa constantemente. La ventana Watch no se actualiza hasta que no se detiene la simulación.
- ▶▶ **Animate** (**Ejecución animada**). Ejecuta el programa de forma continua pero actualizando el contenido de la ventana Watch cada vez que ejecuta una instrucción.
- || **Halt** (**Paro**). Detiene la ejecución del programa y actualiza todos los valores de las ventanas de visualización.
- ↵ **Step Into** (**Ejecución paso a paso**). Ejecuta una sola instrucción del programa y actualiza la información de las ventanas de visualización.
- 📄 **Reset**. Equivale a un reset por activación del pin **/MCLR**.

- Otro elemento asociado a la simulación son los **puntos de ruptura o breakpoints**, que constituyen puntos o instrucciones donde el usuario decide que debe detenerse la ejecución del programa.
- Para **ubicar un breakpoint** sobre una línea señalada por el cursor, se seleccionará la opción **Set Breakpoint** del menú que aparece al pulsar el botón derecho del ratón. Otra posibilidad es hacer doble click sobre la línea donde se quiere colocar. En cualquier caso, aparecerá una "B" de color rojo en la posición donde se ha situado el punto de ruptura. 
- Al simular una ejecución continua, el programa se detendrá en la instrucción que se ha marcado con el punto de ruptura. Para continuar con la simulación desde ese punto hay que volver a lanzar la simulación.
- Para **eliminar puntos de ruptura**, basta con hacer doble click sobre la línea en que se encuentran. Si hay muchos, puede resultar más útil seleccionar **Breakpoints ► Remove All Breakpoints** en el menú que se activa con el botón derecho del ratón la opción. Este menú también ofrece la opción de activar/desactivar los puntos de ruptura colocados en el programa.

Depuración con MPLAB ICD 2


- Permite comprobar el funcionamiento del programa **sobre la tarjeta** en la que se va a colocar el microcontrolador.
- El control de la depuración se lleva a cabo con los **mismos controles que se usan durante la simulación** (Run, Animate, Halt, ...).
- Es preciso **conectar el dispositivo** entre el PC y la PCB que albergará el PIC.



- Para lanzar el depurador es preciso seleccionar la opción **Debugger >> Select Tool ▶ MPLAB ICD 2.**



- Antes de establecer la conexión, puede resultar adecuado configurar el equipo entrando en **Debugger >> Settings...** y seleccionando las siguientes pestañas.
 - **Communication.** Indicamos que vamos a conectar el ICD 2 al puerto USB. También podría utilizarse alguno de los puertos serie (COMn).
 - **Program.** Seleccionamos la opción *Allow ICD 2 to select memories and ranges* para hacer que sea el depurador el que decida qué zonas de memoria deben grabarse en el PIC.

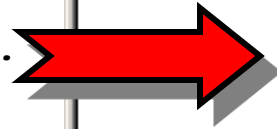
- A continuación es necesario establecer la **conexión con el depurador**, para lo cual se selecciona **Debugger > > Connect** o se hace click sobre el icono. 
- Si ha habido errores de conexión, deberá verificarse el cableado del equipo y la configuración indicada en **Debugger > > Settings...**, fijándose ahora también en las pestañas
 - **Status**. Nos indicará si la **conexión** se ha hecho de manera adecuada. Permite además configurar el sistema para que esta conexión se establezca de manera automática.
 - **Power**. Indica los valores de las **tensiones del sistema**. Para que todo funcione correctamente se necesita Target Vdd≈5V (si no es así, la placa está mal alimentada), Target Vpp≈13V y MPLAB ICD 2 Vpp≈13V.
- Una vez establecida la conexión, el MPLAB ICD 2 ya está listo para **transferir nuestro programa al microcontrolador que se encuentra en la placa de la aplicación**, lo que permitirá llevar a cabo la **depuración**.
- En modo depuración se puede ejecutar, parar, transferir contenidos, etc.

- La **depuración con MPLAB ICD2** se puede realizar porque el microcontrolador admite trabajar en modo depuración cuando el **bit DEBUG** de la palabra de configuración se graba como '0'

- Si el microcontrolador está trabajando en **modo depuración** hay ciertos **recursos** que necesita este modo de trabajo: 2 pines (RB6 y RB7), 1 posición de la pila hardware del PC, las últimas 256 posiciones de la memoria de programa y varias posiciones de RAM 0x070 (0x0F0, 0x170, 0x1F0), 0x1EB-0x1EF

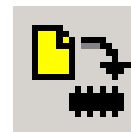
- Se deben cargar los bits de la palabra de configuración:

Configure >> Configuration bits...



Address	Value	Category	Setting
2007	3FFF	Oscillator	RC
		Watchdog Timer	On
		Power Up Timer	Off
		Brown Out Detect	On
		Low Voltage Program	Enabled
		Flash Program Write	Enabled
		Data EE Read Protect	Off
		Code Protect	Off

- Para grabar tanto la palabra de configuración como la memoria de programa (tras haber ensamblado) se seleccionará **Debugger >> Program**.



PROGRAMAR

• Una vez programado el dispositivo, se procede del mismo modo que se hizo con el simulador con la diferencia de que ahora, además de poder ver el valor de los registros en las ventanas de visualización, **comprobaremos el funcionamiento real en la placa.**



Run : para lanzar la ejecución continua del programa



Animate : ejecución con animación (actualización de posición y reg.)



Halt : parada de la ejecución



Reset : reinicio del microcontrolador



Step into : ejecución paso a paso



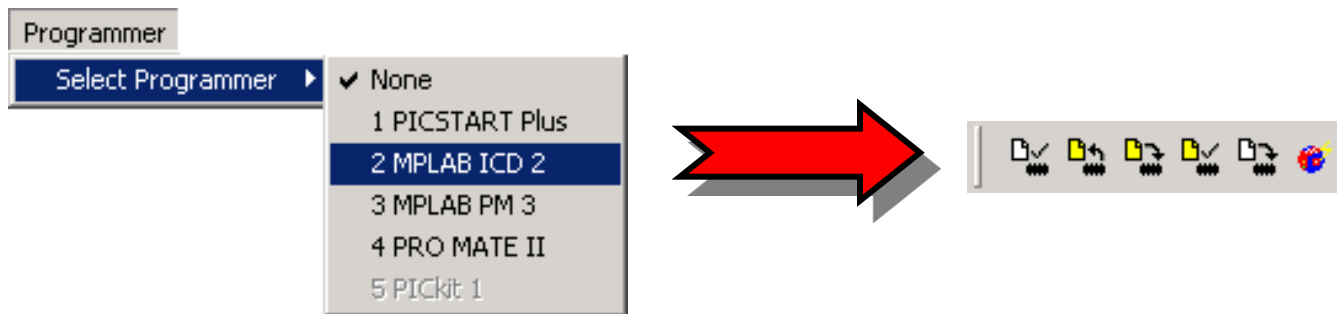
Step over : paso a paso ejecutando los subprogramas completos



Step out : ejecución hasta que se sale de subprograma

Grabado de PICs con MPLAB ICD 2

- Además de llevar a cabo procesos de depuración, el MPLAB ICD 2 permite grabar los microcontroladores que se incluirán en la placa final.
- Para ello, hay que seleccionar este dispositivo como elemento programador, lo cual se consigue con **Programmer >> Select Programmer ▶ MPLAB ICD 2**.



- El MPLAB ICD 2 no puede funcionar como depurador y programador al mismo tiempo. Por ello, **para seleccionar el MPLAB ICD 2 como programador, deberemos asegurarnos de que no está seleccionado como depurador (y viceversa)**.

- Las opciones de grabación que se ofrecen son las siguientes:
 - **Program**. La que se usa para **grabar** el PIC. Es la más habitual.
 - **Read**. **Lee** el programa que está grabado en el PIC y lo carga en la memoria de programa del entorno MPLAB.
 - **Verify**. **Comprueba** que la grabación se ha efectuado correctamente.
 - **Erase Part**. **Borra** completamente el PIC antes de programarlo.
 - **Blank Check**. **Comprueba que el PIC está borrado**.
- Después de programado, el PIC puede desmontarse del *header* (placa con zócalo) y conectarlo directamente en la placa.
- El programa comenzará a ejecutarse en cuanto el microcontrolador reciba la tensión de alimentación.