Using the Amicus18 Compiler with MPLAB™ IDE

The Amicus18 compiler can be used within the Microchip MPLAB™ IDE environment and allows single stepping of the code on a high-level basis. i.e. BASIC lines of code, or the use of the ICD2™, PICkit2™ or a Microchip™ Programmer.

We'll walk through the method of operation step by step.

First, download a copy of the latest MPLAB™ IDE because this method will only work on versions 8.30 onwards. The release at the time of writing is 8.40, and it is recommend to use this version. MPLAB™ can be downloaded from www.microchip.com

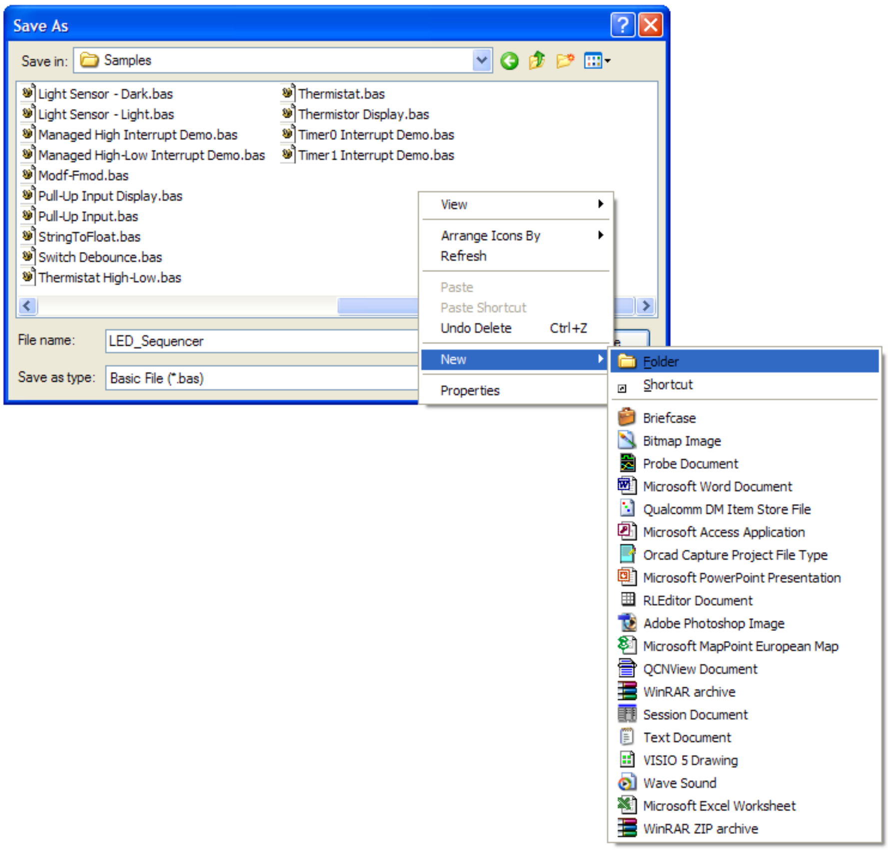
Locate the files ***tlAmicus18.ini*** and ***Amicus18.mtc*** within the compiler's folder (*default location C:/Program Files/AmicusIDE/Mplab*) and copy them into MPLAB's folder *Core/MTC Suites*, overwriting any previous files. MPLAB™ will default to location *C:/Program Files/Microchip/MPLAB IDE*, therefore, the legacy folder should be located at:

C:/Program Files/Microchip/MPLAB IDE/Core/MTC Suites.

Once these files have been copied, locate and run the file ***Mplab_Amicus18.reg***, which can also be found within the compiler's folder. This will add entries into the registry that will register the Proton Amicus18 Compiler as a toolsuite within the MPLAB™ IDE

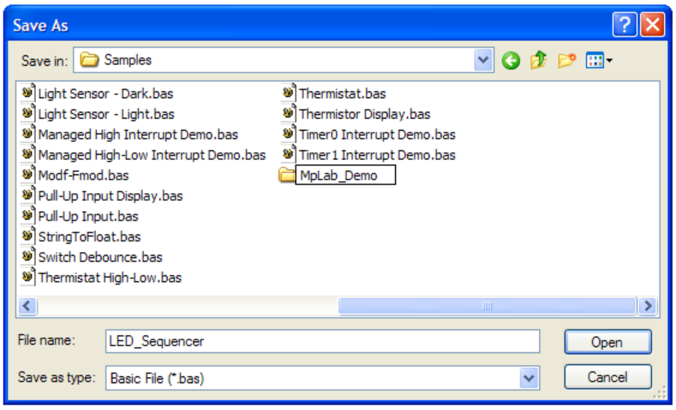
Create the Project files.

We require a program to demonstrate MPLAB™ integration, so from within the compiler's IDE; load the file *LED_Sequencer.bas* from the *Samples* folder, and save it into a newly created folder named *MpLab_Demo*. To do this choose ***Save As*** from the ***Files*** menu. Once the save as dialogue is open, right click the mouse and choose ***New Folder***.



 Version 1.0 06-10-2009

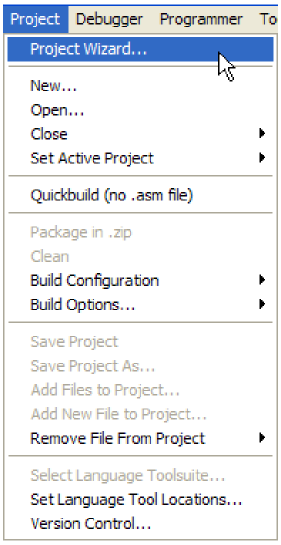
Once the new folder is created, rename it to *MpLab_Demo* then double click on it in order to open it.



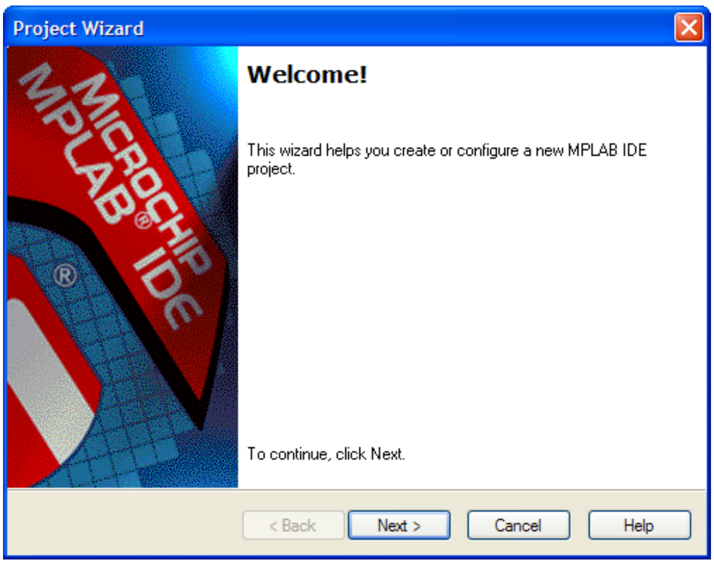
Once inside the *MPLab_Demo* folder, click the *Save* button. The *LED_Sequence.bas* file is now copied to the newly created folder.

Creating an MPLAB^{tm} Project

Open MPLAB^{tm}, then click on the ***Project Wizard*** menu option.

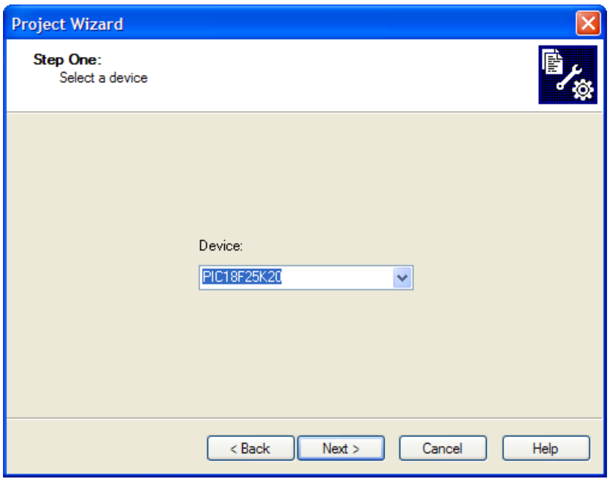


This will open the Wizard intro window as shown below. Click the ***Next*** button.



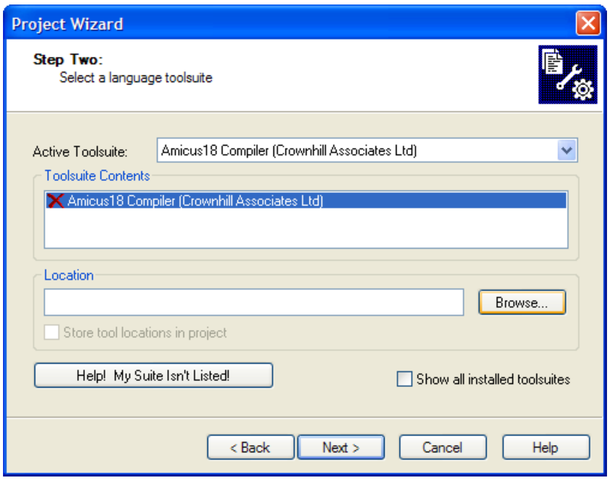
 Version 1.0 06-10-2009

In step 1 of the wizard, choose the 18F25K20 device from the scroll down box. This is the microcontroller device used by the Amicus18 board.

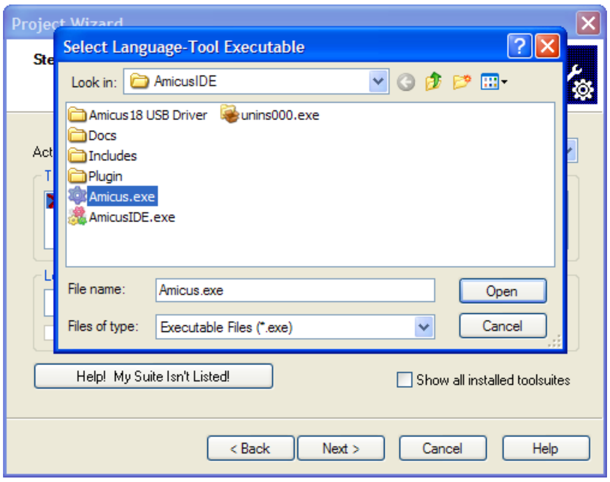


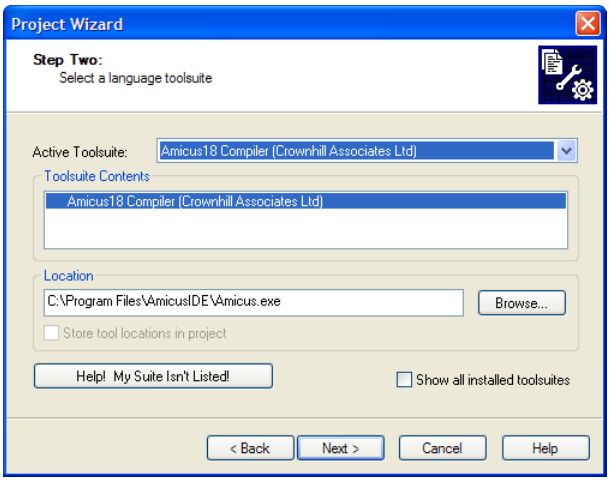
Click ***Next***, then choose the *Amicus18 Compiler* toolsuite, and browse to where the compiler's executable is stored.

The default location for this is *C:/Program Files/AmicusIDE*.

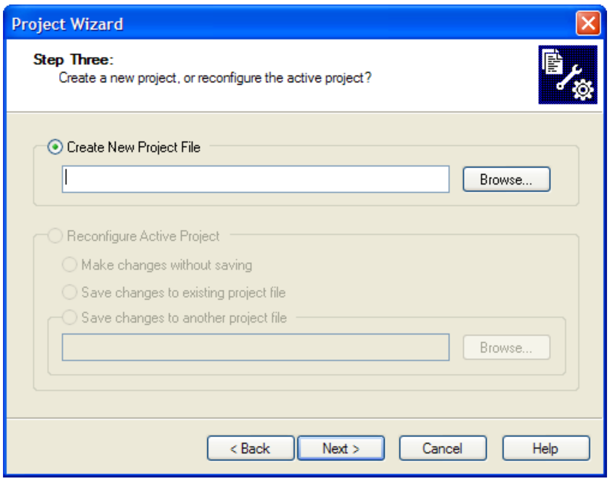


Choose the file named *Amicus.exe* and enter this in the *Location* window.



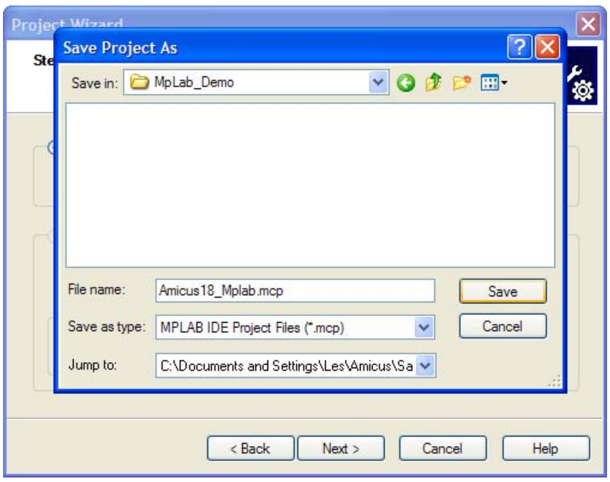


After clicking ***Next***, a project name and location needs to be chosen in the step 3 window. The location for this demonstration project is *MpLab_Demo*.



The project can be placed virtually anywhere on the hard drive as long as it is not nested too deeply, and has the correct permissions. However, it is important to remember that any BASIC file name used within the MPLAB^{tm} environment must contain no spaces. i.e. *LED_Sequencer* instead of *LED Sequencer*. It should be pointed out that this is a quirk of MPLAB^{tm} and not the compiler.

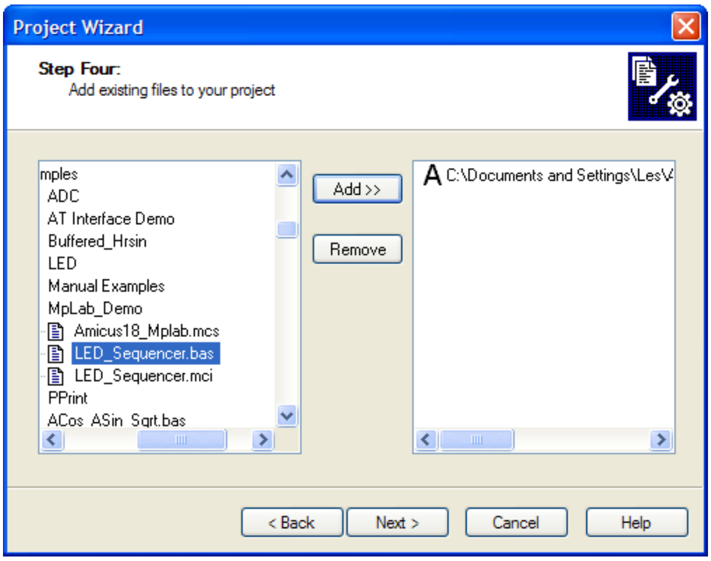
For this demonstration, we'll use our earlier prepared *MPLab_Demo* folder. Click on the *browse* button and navigate to the *Samples* folder. This is located in the *Documents and Settings/User Name/Amicus* directory, (*Users/User Name/Amicus*) if using Microsoft Vista.



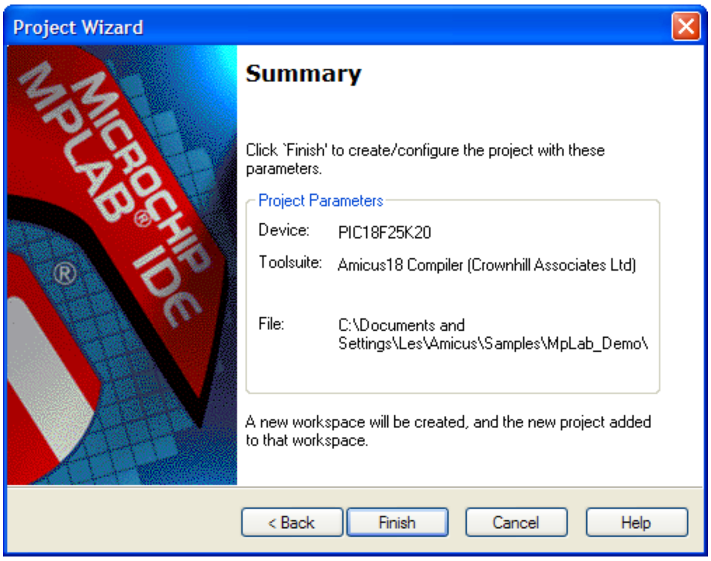
 Version 1.0 06-10-2009

Give the project the name of *Amicus18_Mplab*, and save it, then click ***Next***.

Now we need to add the BASIC file to the project. The BASIC file for the demonstration is the earlier created *LED_Sequencer.bas*. Choose the *LED_Sequencer.bas* file from the left window and click the ***Add*** button. The file name will then also appear within the right hand window. It's very important that only a file with the extension of *.BAS* is added, as any other file will not work.

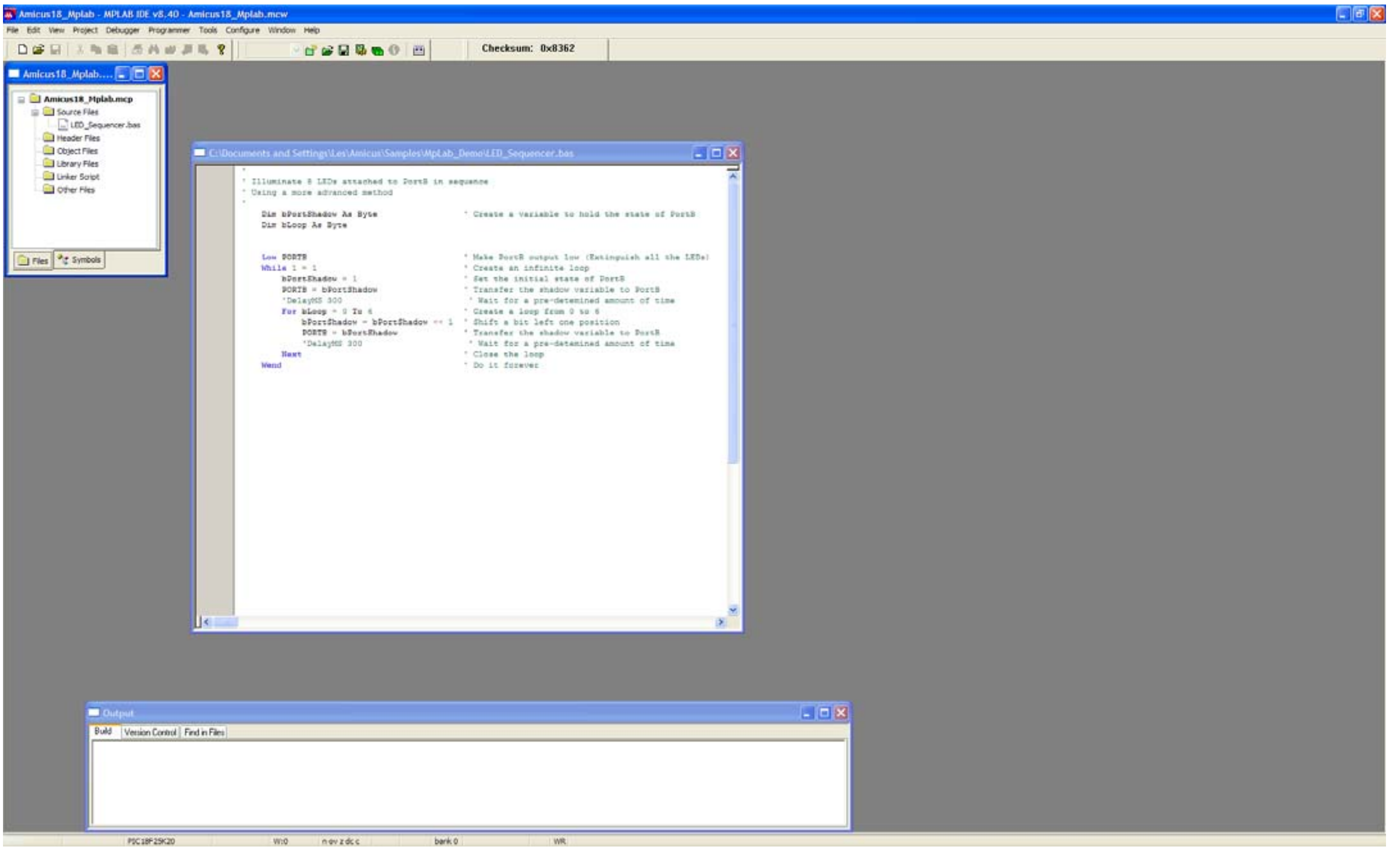


Click on the ***Next*** button and the final wizard window will appear.



Click on the ***Finish*** button.

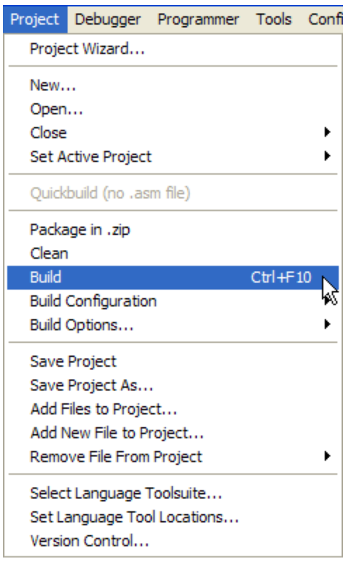
 Version 1.0 06-10-2009



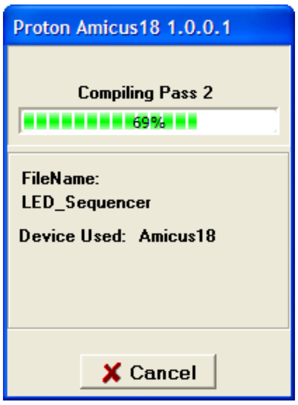
Double click on the *LED_Sequencer.bas* file that appears in the upper left window. This will open the BASIC program within the MPLAB[tm] workspace. At this point edit the BASIC code and comment out the lines containing any **`DelayMs`** commands, as this will slow down the simulation significantly, because MPLAB[tm] does not simulate in real-time.

Building the project

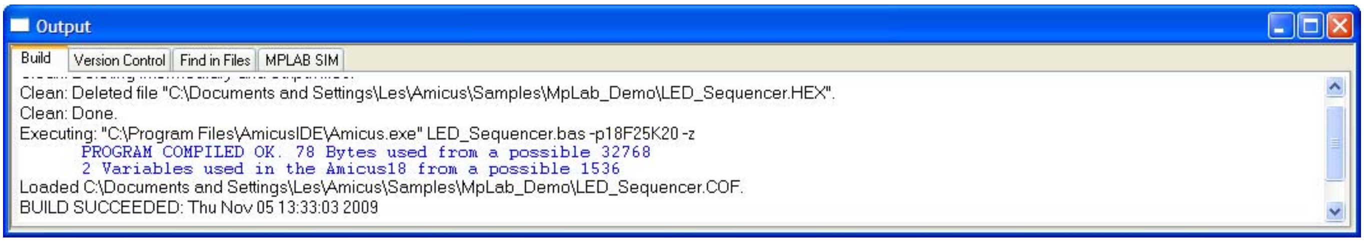
Choose ***Project*** from the top toolbar then the ***Build*** option or press ***Ctrl F10*** to compile the BASIC program.



The Amicus18 compiler will then be called and the code will be compiled as normal.

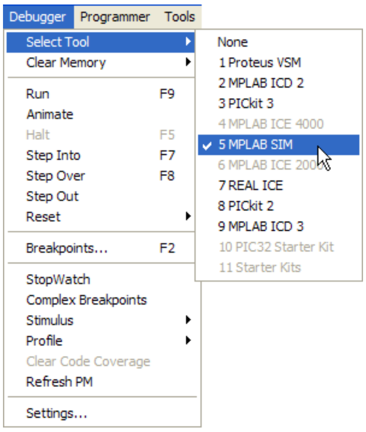


The bottom window of the MPLAB[tm] workspace will show the results of the compilation. Anything other than a *BUILD SUCCEEDED* message will mean that the program failed to compile, and any error messages will be displayed.

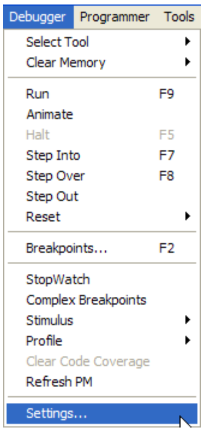


Simulating the program

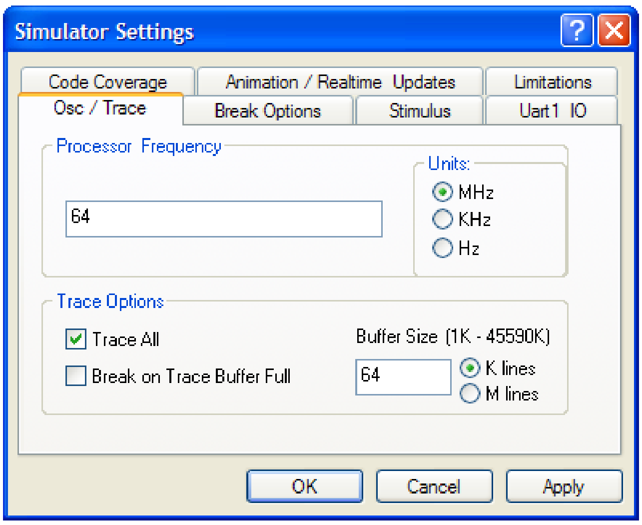
Click on the ***Debugger*** toolbar option, then click ***Select Tool***. Choose the ***MPLAB_SIM*** tool.



We now need to setup the speed of the simulation. Click on the ***Debugger*** toolbar option, then click ***Settings***.

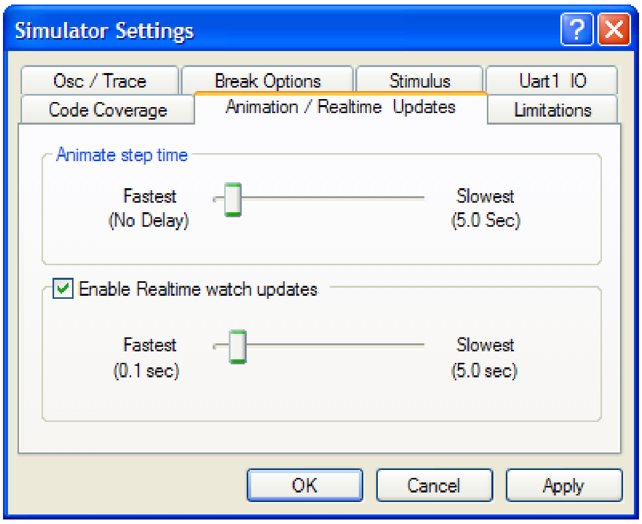


This will open the *Simulator Settings* window. Choose the ***Osc/Trace*** tab and change the *Processor Frequency* value to 64, making sure that the radio button is set for MHz.



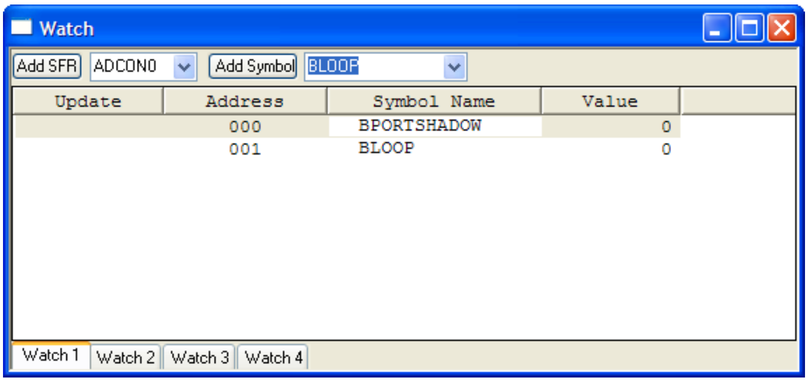
 Version 1.0 06-10-2009

Now choose the ***Animation/Realtime Updates*** tab, and check the ***Enable Realtime watch updates*** option. Also move both sliders to the left (near the Fastest text).

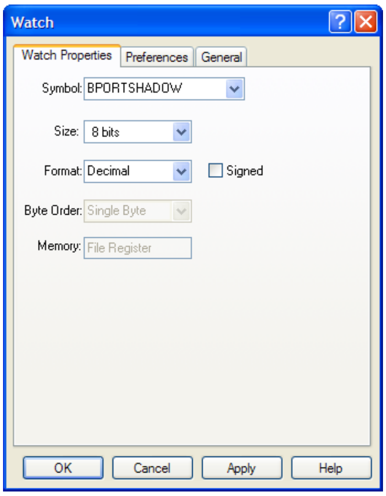


Click the ***Apply*** button, the ***OK*** button.

We want to see the variables within the program alter as it simulates so we need to setup a *Watch* window. Choose ***View*** on the toolbar, then the ***Watch*** option. Once the window opens, choose the variables to watch from the ***Add Symbol*** drop down menu.

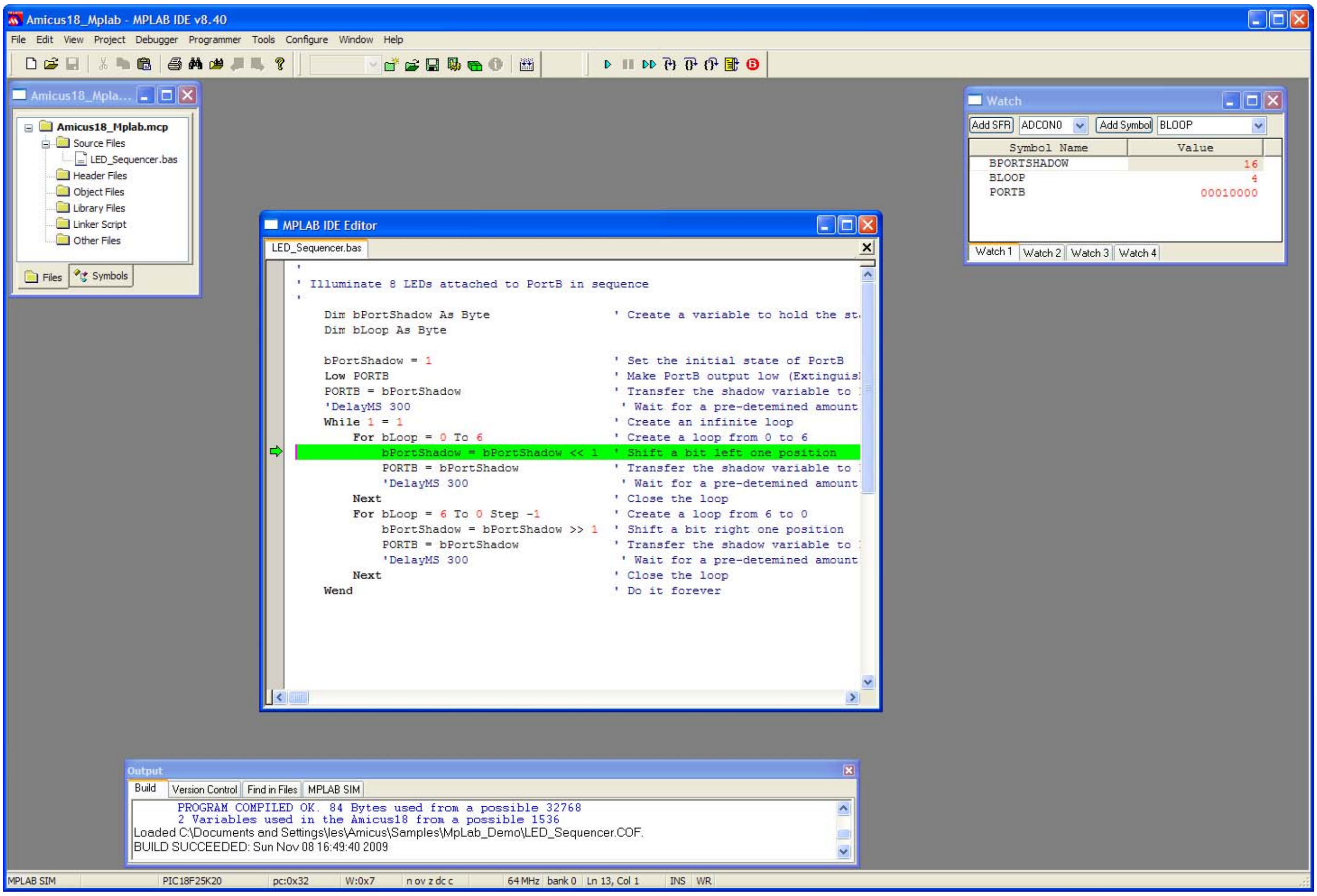


To configure how the variables are displayed, right click on one and a properties window will appear.



For this demonstration choose a format of *Decimal* and untick the *Signed* box. Then click ***Apply*** and ***OK***.

Were now ready to start the simulation by choosing the ***Animate*** button on the toolbar ▷▷. The code line that is currently being simulated will be highlighted green, while the Watch window will show the variables updating as the program simulates.



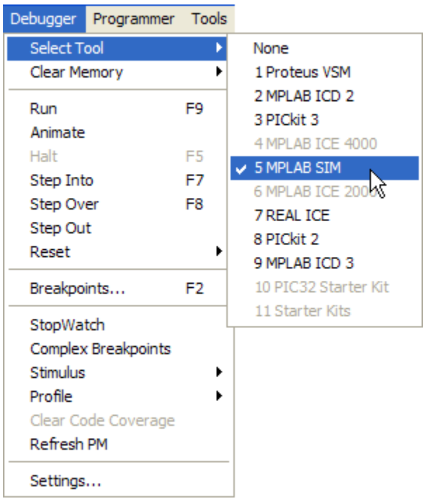
This has been very brief overview of how to simulate a compiled program, but hopefully, it has wetted your appetite. The MPLAB^{tm} manual is full of information, and I suggest you give it a good reading.

Simulating within MPLAB^tm without creating a Project

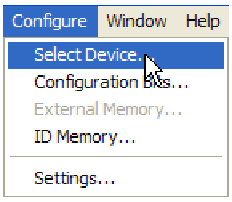
There is another way to simulate within MPLAB^tm without going through the tedious process of creating a project.

When the declare **Create_Coff** = On is placed within the BASIC program, a cof file (Common Object File) is produced during compilation. A cof file has all the information required for simulation, and is as close as it gets to a standard format.

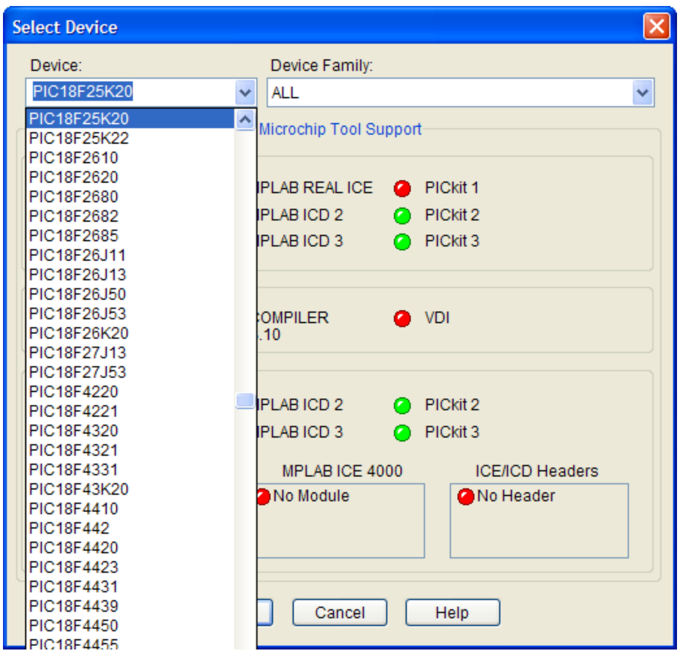
Open MPLAB^tm, and close any open projects, this is an important procedure. You should now be presented with an empty workspace. Choose the Debugger of choice from the ***Debugger*** toolbar menu.



Then choose the appropriate device that the BASIC program is compiled for, by clicking on the ***Configure***->***Select Device*** toolbar menu.



You must now choose the PIC18F25K20 device from the drop down menu, and click the ***OK*** button.



Open the folder where the BASIC file was situated, and drag the file with the extension '.cof' on to the MPLAB[tm] workspace. It will be automatically opened to show the BASIC file. If the BASIC listing doesn't show straight away, it will once the toolbar's ***Animate*** button is used, then the ***Pause*** button.

The program can now be simulated, either by animation or single stepping, as previously documented.

