

Using the PROTON+ Compiler with MPLAB™

As of version 3.3.4.0 of the Proton+ compiler it can be used within the 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 or a Microchip™ Programmer.

I'll walk you through the method of operation step by step.

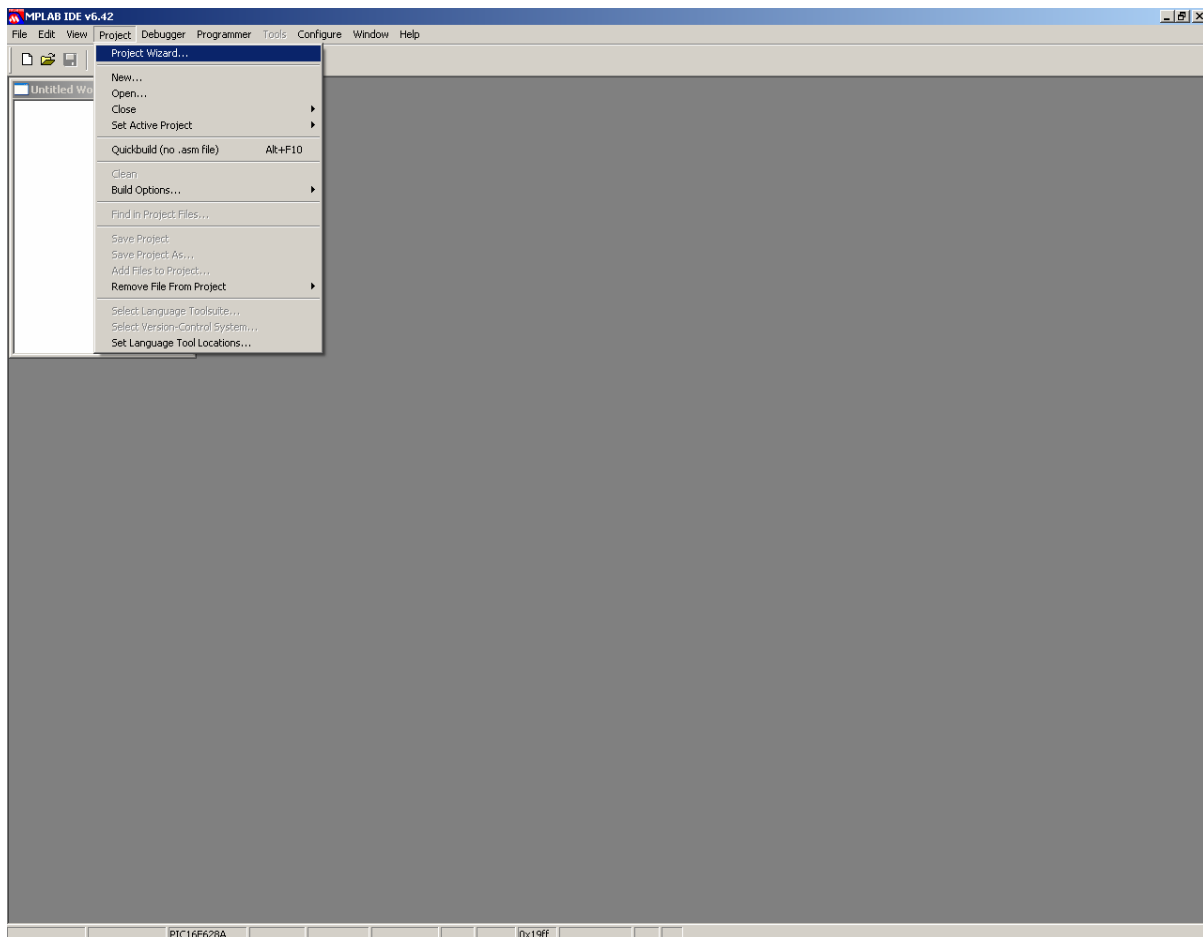
First, download a copy of the latest MPLAB™ IDE because this method will only work on versions 8.0 onwards. The release at the time of writing is 8.20, and I recommend using this version.

Locate the files **TLCHILL.INI** and **PROTON.MTC** within the compiler's folder (default location **C:\Program Files\Crownhill\PDS**) 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:

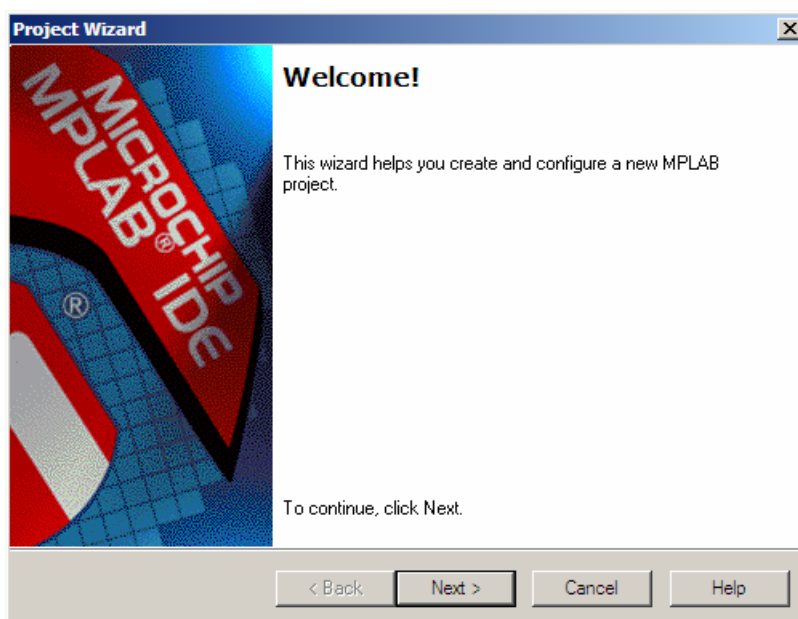
C:\Program Files\Microchip\MPLAB IDE\Core\MTC Suites.

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

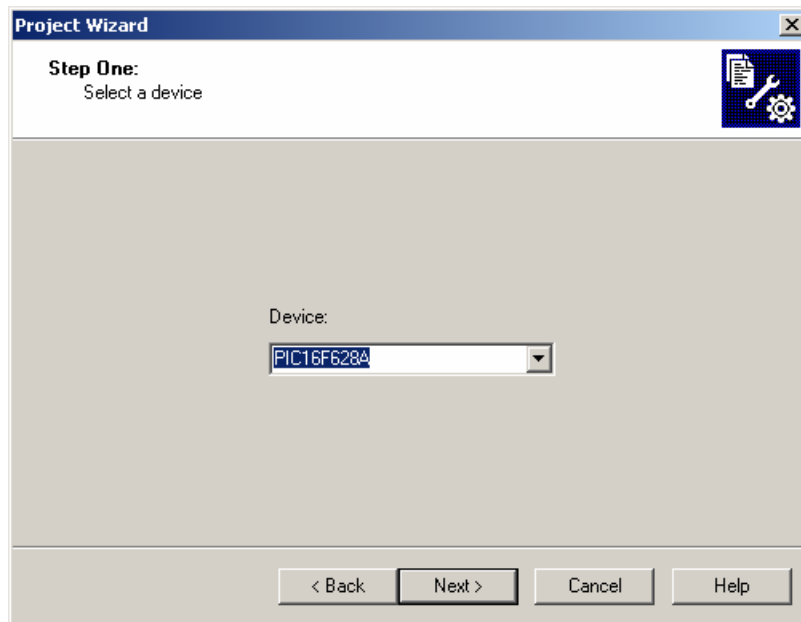
Open MPLAB™, then click on the *Project Wizard* menu option.



And you will be presented by the intro window as shown below.



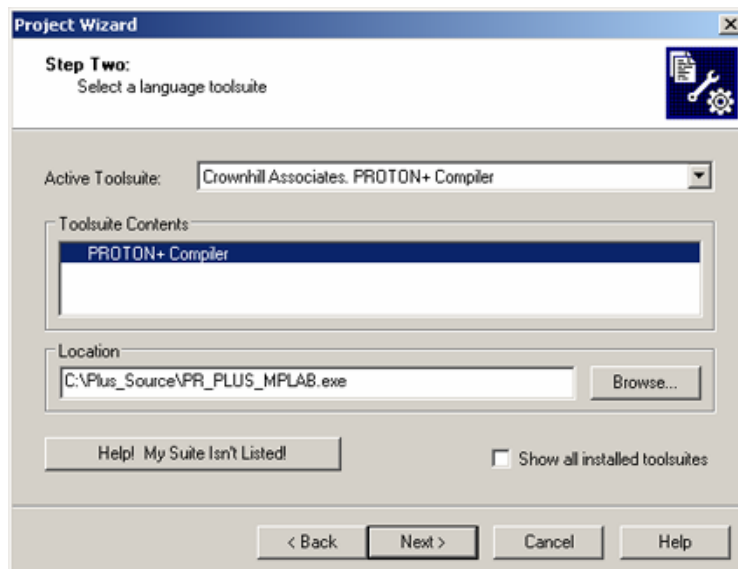
For this demonstration program, the PICmicro™ of choice is the 16F628A, so in the step 1 window, choose the 16F628A device.



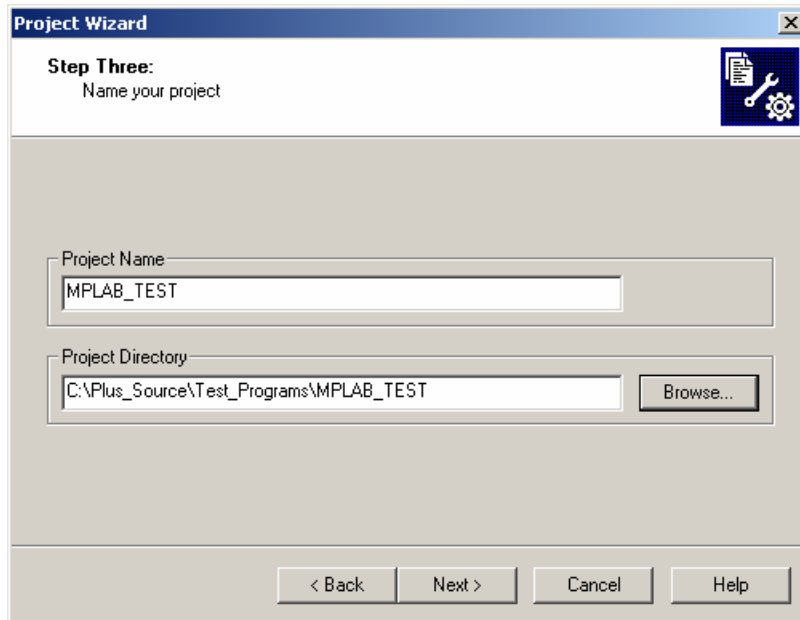
By default the device chosen in this window will be the device that the compiler uses, regardless of a `DEVICE` command within the BASIC listing. The `DEVICE` command will be ignored (see end of document to disable this).

Click *NEXT*, then choose the **Crownhill Associates PROTON+ Compiler** toolsuite, and *browse* to where the PROTON+ compiler's executable is stored. The default location for this is **C:\Program Files\Crownhill\PDS**.

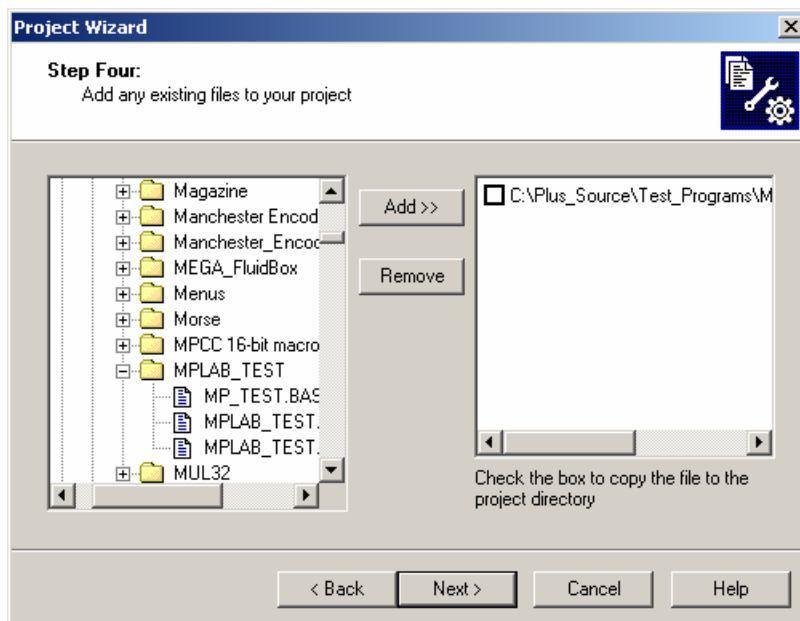
Browse to the file named **PrPlus_Mplab.exe** and enter this in the Location window. It should be within the compiler's folder.



After clicking *NEXT*, a project name and location needs to be chosen in the step 3 window. The name given to the demonstration project is **MPLAB_TEST**, and it's located, in this case, in the compiler's source code folder. But it can be placed virtually anywhere on the hard drive as long as it is not nested too deeply.

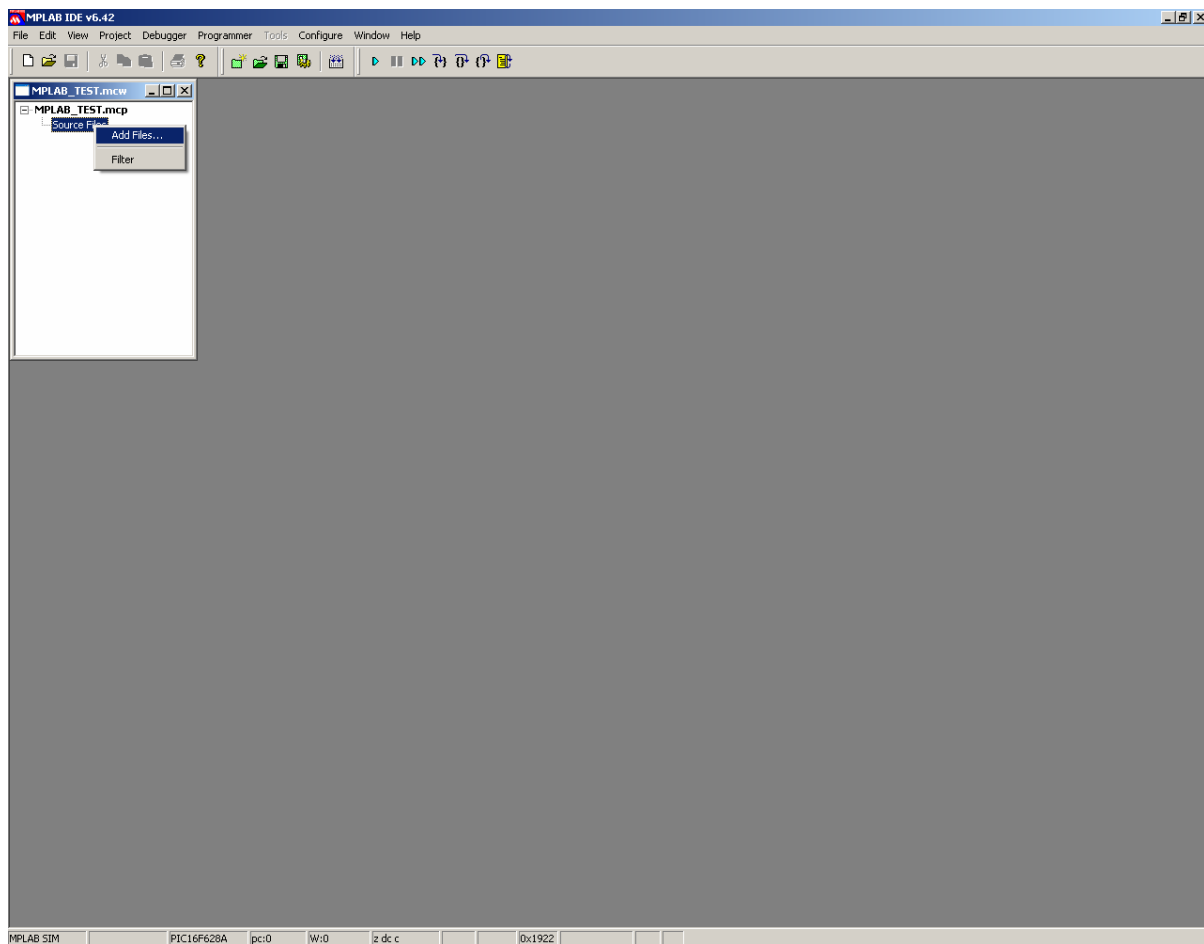


Now we need to add the BASIC file to the project. The BASIC file for the demonstration is named **MP_TEST.BAS**.

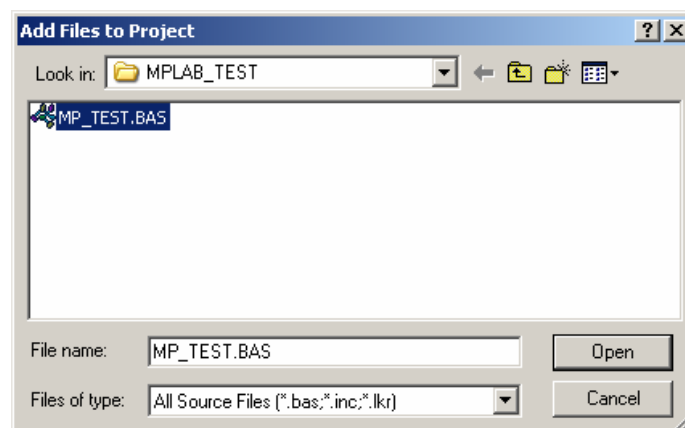


Tick the box in the right hand window only if the chosen **.BAS** file is not located in the projects folder. In this case it is, so leave it unticked.

Clicking *NEXT* a few times after step 4 will create the project. But no BASIC filename has been loaded into the IDE, so right click on *Source File* option located in the *MPLAB_TEST.MCW* window, and choose the appropriate BASIC file.



For the demonstration, the program file name is **MP_TEST.BAS**.



That's it!

Double click on the **MP_TEST.BAS** text in the **MPLAB_TEST.MCW** window, and the BASIC file will be opened ready to compile. Choose *Project* then *Build* or (Ctrl F10) to compile the program.

Open whatever windows as you require and single step or animate the code at BASIC level.

The screenshot displays the MPLAB IDE v6.42 interface with several windows open:

- Project Explorer (MPLAB_TEST.mcw):** Shows the project structure with 'Source Files' and 'MP_TEST.BAS'.
- Source Editor (C:\Plus_Source\Test_Programs\MPLAB_TEST\MP_TEST.BAS):** Contains the BASIC source code:


```

1 '
2 ' Program MP_TEST.BAS
3 '
4 ' Demonstration program testing the PROTON+ compiler
5 ' used in MPLAB 6.40 onwards
6 '
7
8 Device = 16F628A           ' Choose a 16F628A device
9 XTAL = 4                   ' Use a 4MHz crystal
10
11 Dim DWD as Dword          ' Declare a Dword
12
13 DWD = 0                    ' Reset the counting variable
14 While 1 = 1                ' Create an infinite loop
15   Inc DWD                  ' Increment the counting variable
16 Wend                       ' Do it forever
      
```
- Assembly Editor (C:\Plus_Source\Test_Programs\MPLAB_TEST\MP_TEST.ASM):** Shows the assembly code generated from the BASIC program:


```

14 DWDHH Equ 34
15 DWDHHH Equ 35
16 ; F1_000013 in [MP_TEST.BAS] DWD = 0      ' Reset the counting variable
17 Clrf DWDHHH
18 Clrf DWDHH
19 Clrf DWDH
20 Clrf DWD
21 ; F1_000014 in [MP_TEST.BAS] While 1 = 1    ' Create an infinite loop
22 bc@LL1
23 ; F1_000015 in [MP_TEST.BAS] Inc DWD        ' Increment the counting variable
24 Movlw 1
25 Addwff DWD,F
26 Clrw
27 Btfsc STATUS,0
28 Movlw 1
29 Addwff DWDH,F
30 Clrw
31 Btfsc STATUS,0
32 Movlw 1
33 Addwff DWDHH,F
34 Clrw
35 Btfsc STATUS,0
36 Movlw 1
37 Addwff DWDHHH,F
38 ; F1_000016 in [MP_TEST.BAS] Wend          ' Do it forever
39 Goto bc@LL1
40 bc@LL2
41 END
      
```
- Special Function Registers:** A table showing the status of various registers:

SFR Name	Hex	Dec
WREG	00	
INDF	--	
TMRO	00	
PCL	00	
STATUS	1C	
FSR	00	
PORTA	20	
PORTB	00	
PCLATH	00	
INTCON	00	
PIR1	00	
TMR1	0000	
TMRL	00	
TMRH	00	
T1CON	00	
TMR2	00	
T2CON	00	
CCPR1	0000	
CCPR1L	00	
- Watch Window:** Shows the current value of the DWD variable:

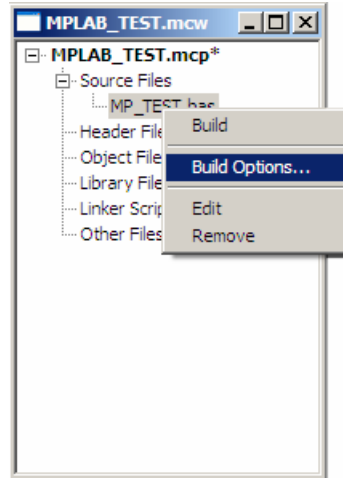
Address	Symbol Name	Value
0020	DWD	658
- Output Window:** Displays the execution log:


```

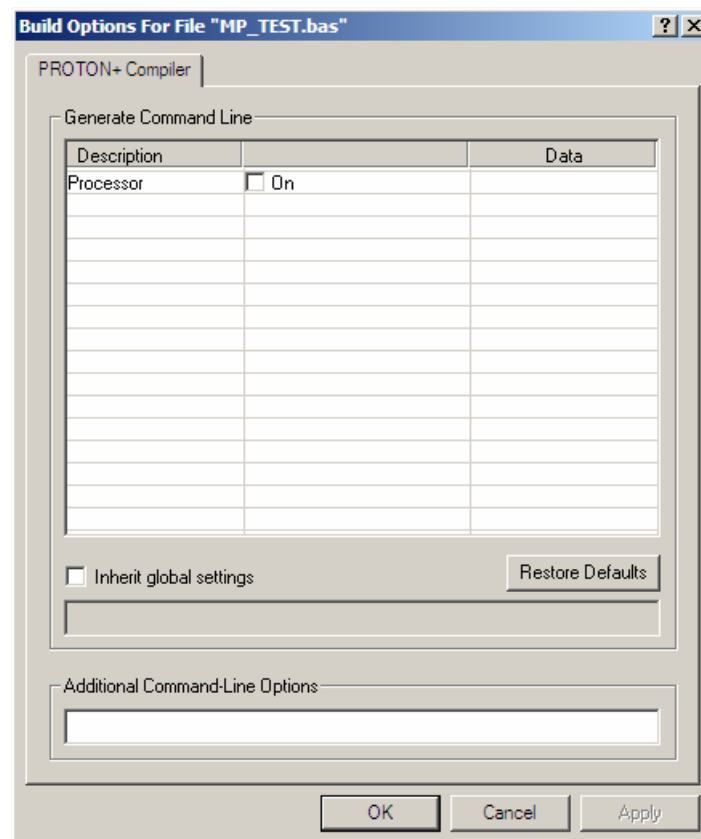
Executing: "C:\Plus_Source\PROTON_PL.exe" -p16F628A -q -m "MP_TEST.BAS"
Loaded C:\Plus_Source\Test_Programs\MPLAB_TEST\MP_TEST.COD
BUILD SUCCEEDED: Tue Feb 10 22:44:01 2004
      
```
- Bottom Status Bar:** Shows the target device (PIC16F628A), processor (pc:0), and other simulation parameters.

Disabling the Automatic DEVICE Selection.

By default, MPLAB™ forces the compiler to ignore any `DEVICE` directives within the BASIC program in favour of whatever device is chosen in the *Configure->Select Device* Options menu. This can be disabled by right clicking on the filename within the *MCW* window and choosing *Build Options*.



You will be presented with a configuration window containing a single switch.



Untick the On switch and click Apply.

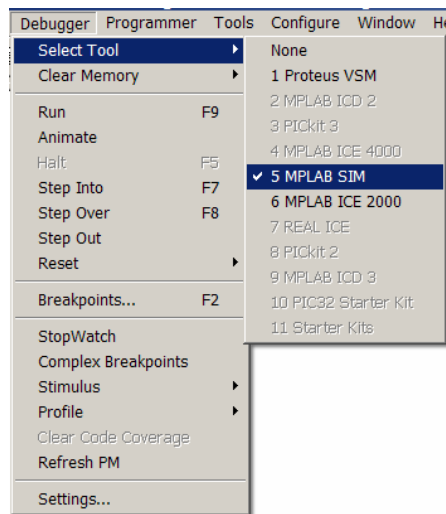
The device that the compiler recognises is now issued by the `DEVICE` directive within the BASIC program, therefore ensure that MPLAB[™] is configured for the correct PICmicro[™] device for any simulations of programming.

An Easier Method

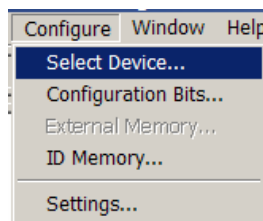
There is another way to simulate within MPLAB™ without going through the tedious process of creating a project.

When the directive `CREATE_COFF = ON` is placed within the BASIC program, a `cof` 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.

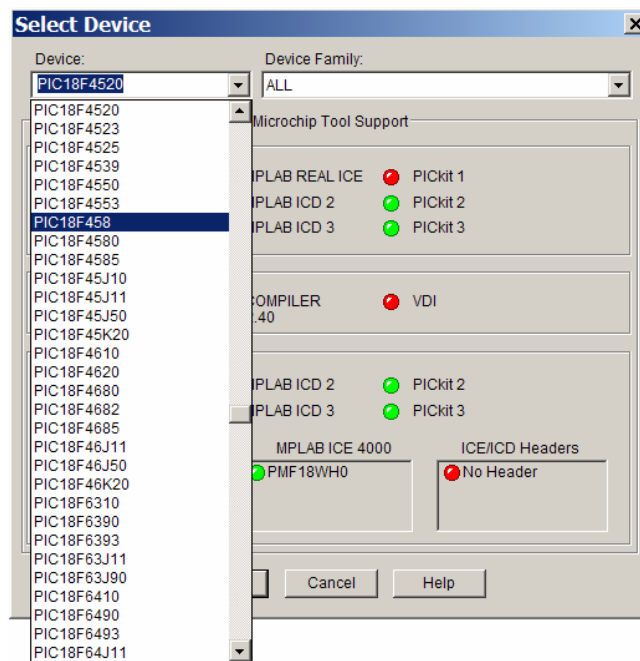
Open MPLAB™, and close any open projects, this is an important procedure. You should now be presented with an empty workspace. Choose the debugger of choice from the *debugger* toolbar menu.



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



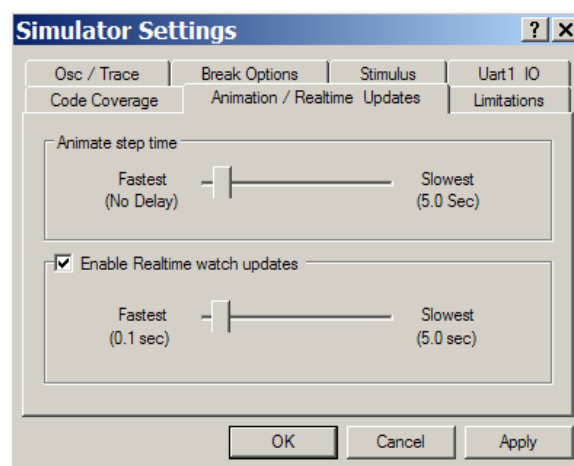
You can now choose the device from a list.



Open the folder where the BASIC file was situated, and drag the file with the extension '.cof' on to the the MPLAB™ workspace. It will be automatically opened to show the BASIC file.

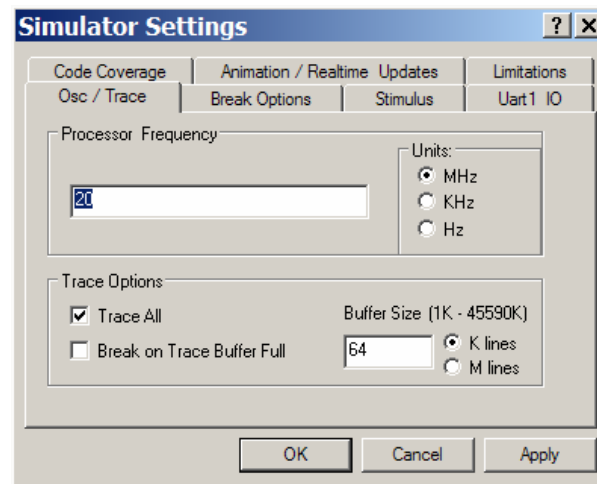
The program can now be simulated, either by animation or single stepping. However, there are still 2 steps to carry out that will improve the simulation. Click on the *debugger* toolbar menu, and choose the bottom option. i.e. *settings*.

When the window shows, click the *Animation/Realtime Updates* tab.



Tick *Enable Realtime watch updates*, and move the step time closer to the fastest side. i.e. far left.

Now click on the *Osc/Trace* tab, and choose the oscillator frequency used in the BASIC program.



Click Apply for the settings to take effect, then OK.

You can now open watch windows, dissassembly listings etc, and watch the variables update as the simulation is in progress.

This has been a very brief explanation on how to incorporate the Proton+ compiler into MPLABtm, but it gives you the basics (Pun intended). The rest is up to you. Have fun.