PICkit 2 Debug Express

GIỚI THIỆU

Ngoài ứng dụng Nạp chương trình (Programmer), PICkit 2 Development có thể được sử dụng để Nạp/ Gỡ rối với MPLAB IDE , MPLAB IDE cho phép PICkit 2 được sử dụng như một in-circuit debugger as well as a programmer(chỉ cho những thiết bị được lựa chọn).

In-circuit debugger cho phép bạn chạy, khảo sát và sửa đổi chương trình của mình trong khi chip nằm trong phần cứng đích. Điều này rất giúp bạn rất nhiều trong việc gỡ lỗi vi chương trình (Firmware) và phần cứng (Hardware) cùng nhau.

Đặc biệt phần mềm PICkit 2 Debug Express tương tác với MPLAB IDE application để chạy (run), dừng (stop) và chạy từng bước (single-step). Một hoặc nhiều điểm dừng có thể được sử dụng và bộ xử lý có thể Reset lại. Một lần bộ xử lý dừng lại, nội dung của thanh ghi có thể được khảo sát và thay đổi.

Để biết chi tiết hơn về cách sử dụng MPLAB IDE, tham khảo các tài liệu sau:

- MPLAB® IDE User.s Guide (DS51519)
- MPLAB® IDE Quick Start Guide (DS51281)
- MPLAB® IDE On-line Help

CÀI ĐẶT PHẦN CỨNG VÀ PHẦN MỀM

Cài đặt Phần cứng PICkit 2 Như được chỉ rõ trong hướng dẫn sử dụng PICkit 2

Lưu ý: *Trong một số tài liệu và phần cứng cũ của Microchip có đề nghị sử dụng điện trở pull-down 4,7KΩ trên các chân ICSPCLK và ICSPDAT cho debugger. PICKit 2 mới của Microchip và của TMe đã có sẵn các điện trở này bên trong vì vậy không cần sử dụng thêm các điện trở này trên board mạch đích.*

Cài đặt phần mềm ứng dụng MPLAB IDE lấy từ website của microchip hoặc trong đĩa CD_ROM kèm theo, và cài đặt nó theo định hướng của phần mềm.

Lưu ý: *Debug Express yêu cầu phiên bản MPLAB IDE 7.50 hoặc mới hơn*

SỬ DỤNG PICKIT 2 DEBUG EXPRESS

1 Xác định Device Support

Một danh sách cho các chip hiện thời được hỗ trợ bởi PICkit 2 Debug Express, xem file "*Readme for PICkit 2.htm*" nằm trong thư mục con "Readmes" của thư mục cài đặt MPLAB IDE

Khi Lựa chọn một thiết bị như được nói đến trong mục **4.5 "Debug Express Tutorial"** ,Hộp thoại "Select Device" cho thấy trong Hình 4-11 hiện ra mức độ hỗ trợ cho thiết bị được chọn bởi Debugger Express. Trong mục của hộp thoại "Debugger", màu của hình tròn tại vị trí "PICkit 2" cho thấy mức độ hỗ trợ :

- Màu Đỏ - Thiết bị hiện chưa được hỗ trợ bởi PICkit 2 Debug Express
- Màu vàng - Thiết bị có sự hỗ trợ beta bởi PICkit 2 Debug Express
- Màu Xanh lục - Thiết bị có đầy đủ hỗ trợ bởi PICkit 2 Debug Express

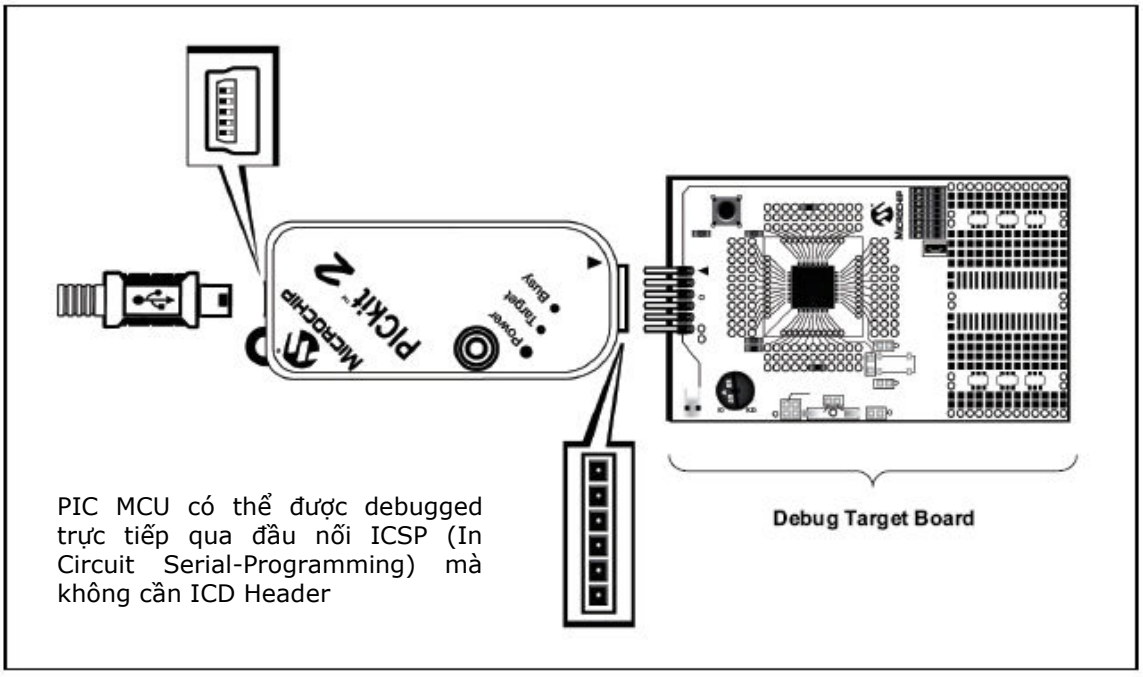
Chỉ báo sự hỗ trợ Beta là chỉ báo thiết bị được hỗ trợ, nhưng chưa được kiểm chứng với sự thử của Microchip.

2. Tài nguyên dự trữ

Vì các thiết bị có khả năng ICD xen kẽ chức năng in-circuit debugging và chức năng ICSP trong quá trình debugging, nên PICkit 2 Debug Express sử dụng một số tài nguyên on-chip trong khi debug, những vùng (vị trí) này không sẵn sàng cho sự sử dụng cho mã người dùng. Trong MPLAB IDE những thanh ghi được gắn mác "R" trong cửa sổ Register là đại diện cho những thanh ghi dự trữ

Để biết thông tin về tài nguyên thiết bị cần cho in-circuit debugging vui lòng xem trong MPLAB ICD2 Help, tìm trong *help→ topic*. Tài nguyên sử dụng trong MPLAB ICD2 thì cũng giống như trong PICkit 2 Debug Express.

3. Sử dụng ICSP Header



Hình 4.2 Nối demo board đến PICkit 2

4. Configuration Bits and Debug Express

PIC microcontroller devices không yêu cầu một ICD Header và có thể được gỡ lỗi trực tiếp bằng #DEBUG bit trong Configuration Word(s). Điều đó cho phép hoặc loại bỏ chế debugger của PIC Microcontroller.

Bit này được tự động thiết lập cho phù hợp bởi MPLAB IDE khi sử dụng PICkit 2 Debug Express và không nên được chỉ rõ trong source code configuration settings.

NHẮC NHỞ:

Ở những điều kiện thiết đặt bình thường giá trị DEBUG configuration bit Không nên được chỉ rõ trong Cấu hình mã nguồn. Vì vậy có thể gây ra bit sẽ được khẳng định Khi việc lập trình một thiết bị bên ngoài trình gỡ rối. Cái này sẽ gây ra thiết bị vận hành không thích hợp hay không đúng với mong muốn trong mạch ứng dụng.

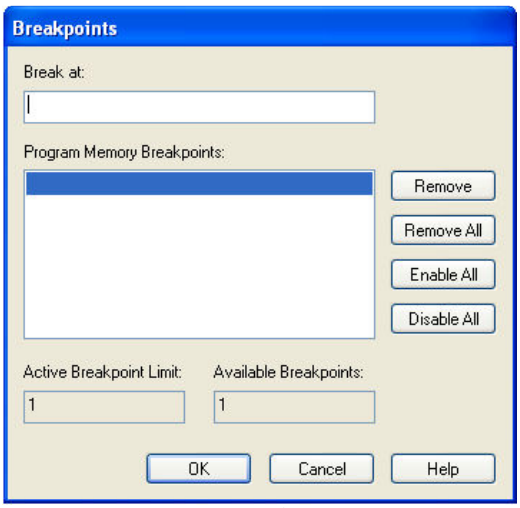
Nhiều chip 16-bit PIC microcontroller chẳng hạn như họ PIC24 và dsPIC33 có nhiều chân ICSP Programming và debugging port pins nhãn PGC1/EMUC1 và PGD1/EMUD1, PGC2/EMUC2 và PGD2/EMUD2, vv. Trong khi bất kỳ cổng ICSP nào

cũng có thể sử dụng để lập trình (nạp), chỉ có một cổng tích cực trong thời điểm debugging

Để kích hoạt EMU port thiết lập trong device Configuration bits. Nếu kích hoạt thiết lập không phù hợp Emu port nối tới PICkit 2, thì thiết bị không có khả năng vào chế độ debug. Trong hộp thoại Configuration Bits của MPLAB IDE, bit này thường được tham khảo tới như "Comm Channel Select" bits.

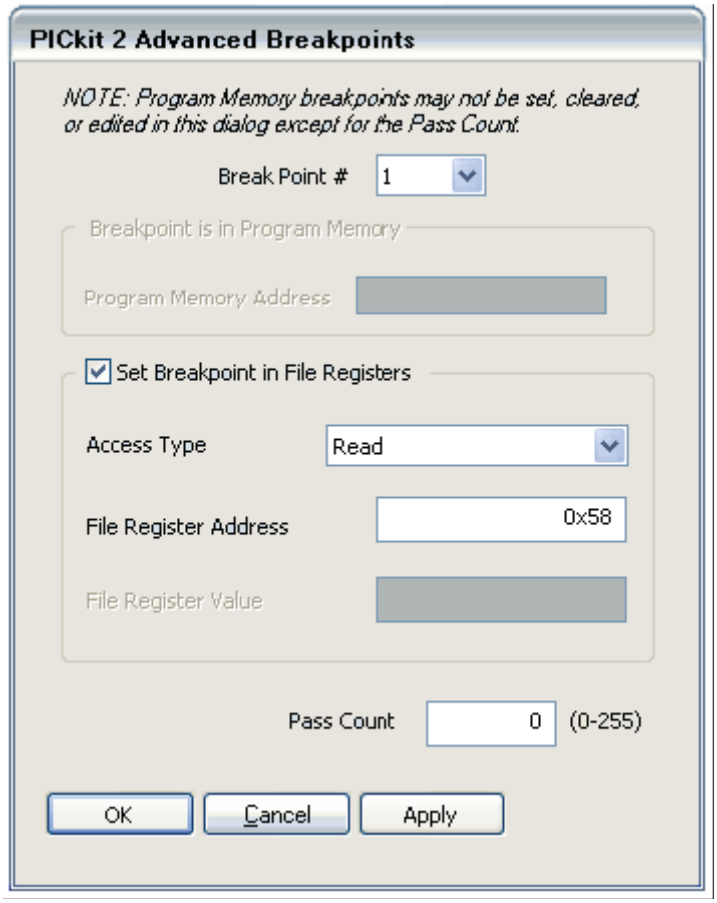
5. Debug Express Breakpoints (điểm dừng)

Số lượng breakpoint kích hoạt được PICkit 2 Debug Express hỗ trợ phụ thuộc vào thiết bị đích, đa số Baseline và Mid-Range devices hỗ trợ 1 breakpoint, một số PIC18 và 16-bit được hỗ trợ nhiều breakpoints hơn
Số điểm dừng tích cực sẵn có cho thiết bị hiện thời trong MPLAB IDE có thể được nhìn thấy bởi lựa chọn "*Debugger>Breakpoints"*. Cái này sẽ mở một hội thoại (Hình 4-3) cho thấy bất kỳ điểm dừng hiện thời được đặt nào trong Kí ức Chương trình (Program Memory). "Active Breakpoint Limit:" hộp văn bản cho thấy tổng bao nhiêu điểm dừng sẵn sàng cho thiết bị hiện thời. "Available Breakpoints:" hộp văn bản cho thấy tổng bao nhiêu điểm dừng cho thiết bị hiện thời không sử dụng



Hình 4-3 Hội thoại những điểm dừng (cho) PIC16F887

Một số PIC18 Và 16- bits cũng hỗ trợ những advanced breakpoints (advanced breakpoints). Những advanced breakpoints cho phép những điểm dừng được thiết lập trong File Register memory, và sẽ dừng sự thực hiện khi một File Register đặc biệt được đọc hay ghi tới. Điểm dừng này có thể cũng được thiết lập sao cho nó sẽ chỉ dừng khi một giá trị đặc biệt được đọc từ hay ghi tới một thanh ghi. Đồng thời, một "Pass Count" có thể thiết lập bất kỳ kiểu điểm dừng nào. Pass count này số của những lần điểm dừng quy định mà nó gặp trước đây ngưng thực hiện (halt execution). Chẳng hạn, thiết lập pass count là "2" điểm dừng trên một Program Memory có nghĩa rằng chỉ dẫn sẽ thực hiện hai lần không có việc dừng sự thực hiện, và thời gian một phần ba chỉ dẫn được thực hiện điểm dừng sẽ dừng sự thực hiện. Mặc định pass count cho tất cả các điểm dừng là "0", có nghĩa rằng sự thực hiện sẽ dừng lần đầu điểm dừng được gặp. Nếu những advanced breakpoints được hỗ trợ bởi thiết bị hiện thời, thì MPLAB IDE menu option *Debugger>Advanced Breakpoints* sẽ sẵn sàng mở một hộp thoại advanced breakpoint, nếu thiết bị không hỗ trợ advanced breakpoints thì menu option sẽ mờ đi xem như không có. Lựa chọn điểm dừng có thể soạn thảo trong hộp "Break Point #"



HÌNH 4-4 ADVANCED BREAKPOINTS DIALOG

Ghi chú:

Hộp thoại Advanced breakpoints sẽ trình bày bất kỳ Điểm dừng nào thiết lập trong Program Memory. Tuy nhiên, hộp thoại có thể không sử dụng để đặt hay xóa những điểm dừng trong Program Memory hay để soạn thảo địa chỉ (của) một Điểm dừng Program Memory hiện hữu. Chỉ giá trị Pass Count cho điểm dừng Program Memory được có thể soạn thảo trong hội thoại Advanced breakpoints. Để soạn thảo, thiết đặt hoặc xóa những điểm dừng Program Memory, sử dụng MPLAB IDE Editor hay menu Debugger>Breakpoints…

6. Breakpoint Skidding (Điểm dừng trượt)

The in-circuit debug implementation on PIC microcontrollers will halt execution on the instruction after the breakpoint instruction. This means the breakpoint instruction will have executed when the debugger halts. This is referred to as "breakpoint skidding".
As a result, there are some breakpoint behaviors to be aware of. When a breakpoint is set on a GOTO, CALL, or RETURN instruction, the debugger will halt at the destination instruction, as the program branch instruction with the breakpoint will have executed. Also, when using the debugger Step Over function, a breakpoint will be set on the instruction after the CALL instruction that the debugger is "stepping over" if there is an available breakpoint. If the CALL instruction is followed immediately by another CALL instruction, this will result in the debugger halting at the destination of the second CALL instruction. To prevent this, a NOP may be placed between the CALL instructions.
Note that 16-bit devices will halt two instructions after the breakpoint instruction.

7. Linker Scripts

Nếu project của bạn sử dụng một linker script, files ICD linker script đặc biệt phải được sử dụng khi debugging mà những tài nguyên dự trữ được sử dụng bởi PICkit 2 Debug Express. Mỗi chip có chứa một separate debug linker file, có kí tự "i" ở cuối tên chip đó
Chẳng hạn:

16F877i.lkr . In-Circuit Debug linker file for the PIC16F877 device
18F4520i.lkr . In-Circuit Debug linker file for the PIC18F4520 device

Khi việc gỡ lỗi với PICkit 2 Debug Express , ICD linker file cần phải được sử dụng thay vào linker file chuẩn.

4.5 HƯỚNG DẪN DEBUG EXPRESS

Hướng dẫn này sử dụng với một board Demo chứa PIC16F887 microcontroller. Board demo này được nối với PICkit 2 debug express KIT qua đường ICSP.

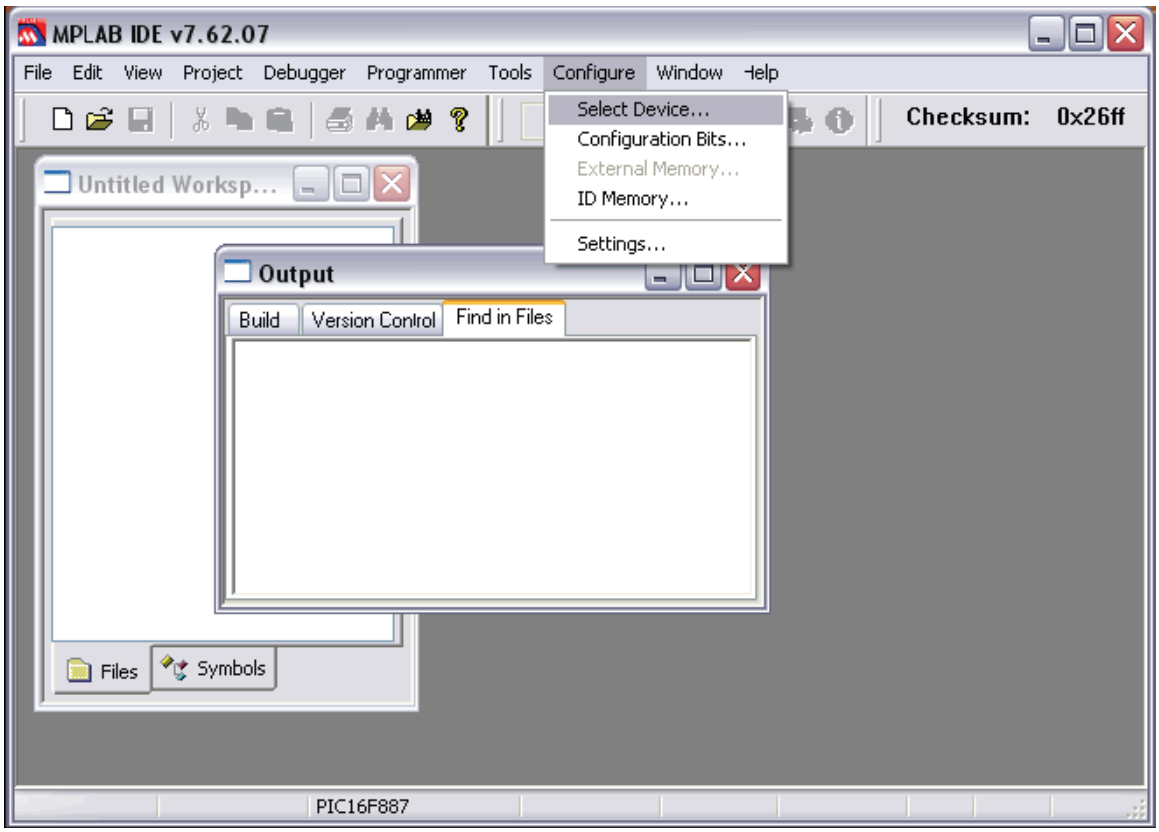
Nếu bạn không có board demo bạn cũng nên đọc qua phần hướng dẫn này để có cái nhìn tổng quan về việc sử dụng PICkit 2 như một trình gỡ rối trong MPLAB IDE.

File nguồn được sử dụng cho hướng dẫn này được cài đặt chung với PICkit 2 Programmer software.

Selecting the Device (Chọn chip)

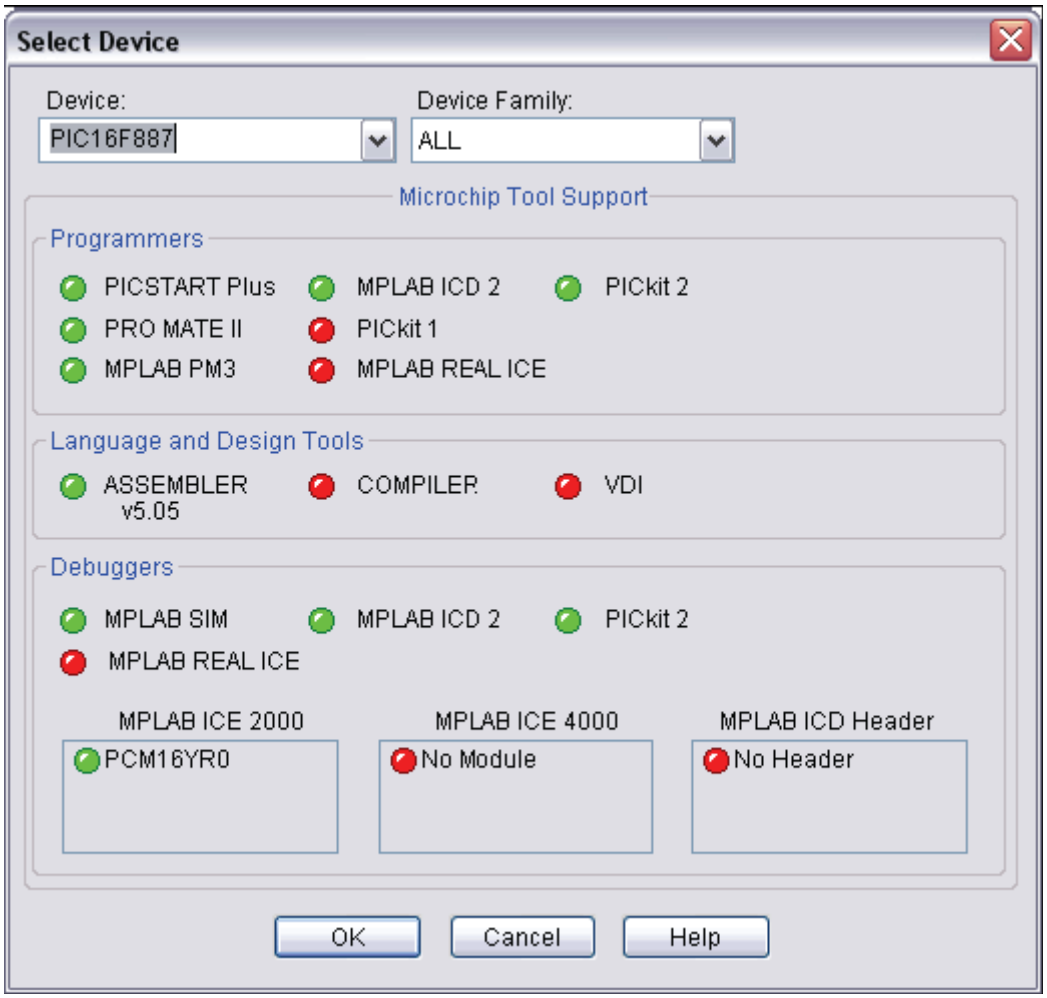
Để chọn chip trong MPLAB IDE:

1. Chạy chương tình ứng dụng MPLAB IDE
2. từ MPLAB IDE menu bar chọn *Configure>Select Device* (Hình 4-5).



Hình 4-5: MPLAB IDE Menu Bar

3. Trong hộp thoại Select Device (Hình 4-6) nhấn vào nút cuộn xuống trong hộp Device và chọn chip bạn cần trong list này. Trong hướng dẫn này, chọn PIC16F887

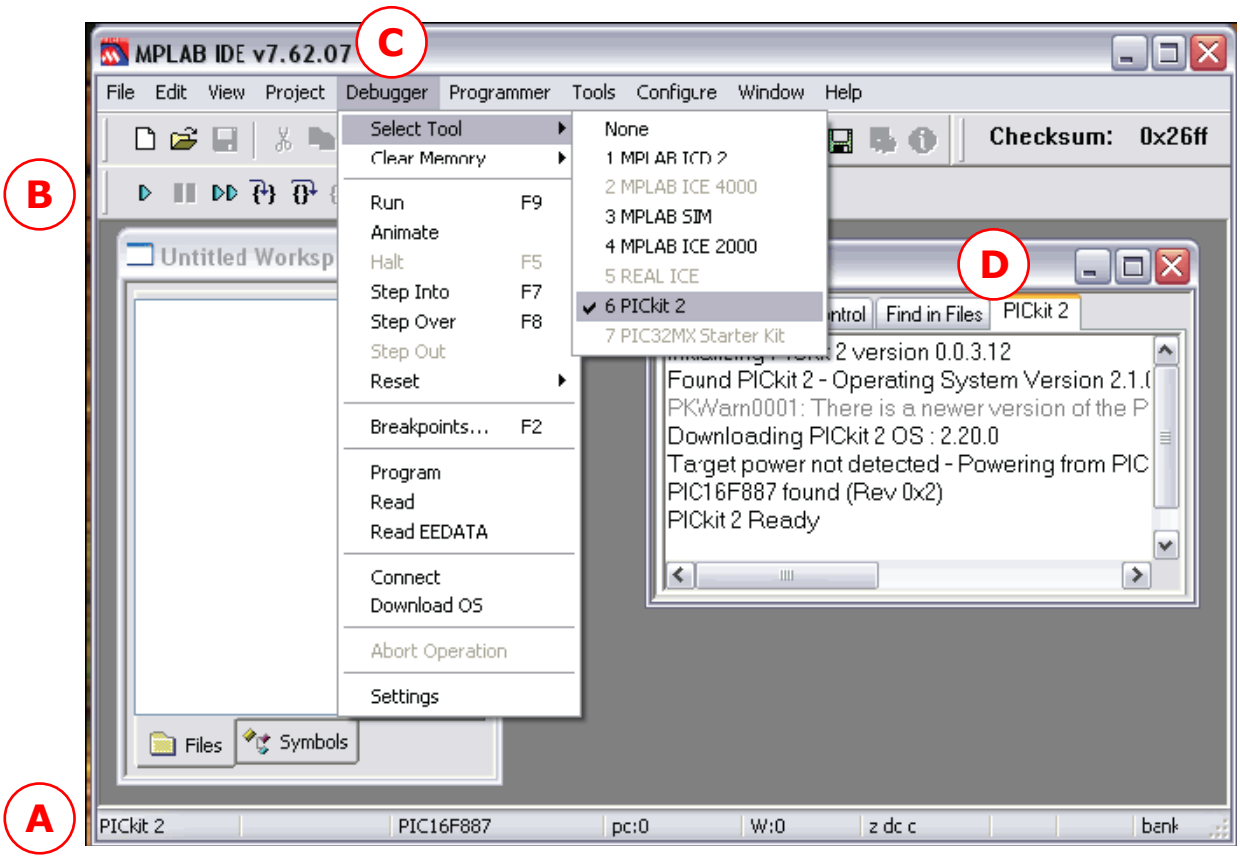


Hình 4-6 SELECT DEVICE DIALOG

4. Không có thay đổi gì khác trong hộp thoại này, Nhấn OK

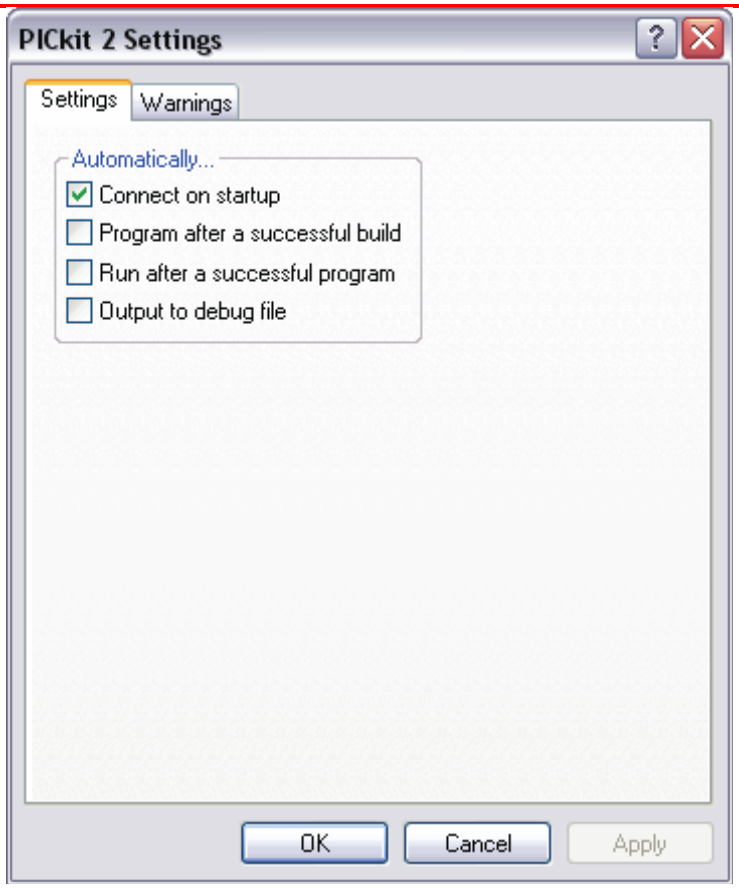
Chọn PICkit 2 như một Công Cụ Gỡ Lỗi

1. Chọn *Debugger>Select Tool>PICkit 2*. MPLAB IDE sẽ thêm PICkit 2 debug features (hình 4-7): (A) thanh trạng thái sẽ hiện ra PICkit 2 như công cụ gỡ lỗi , (B) Một PICkit 2 debug toolbar sẽ được thêm vào, (C) Debugger menu sẽ thay đổi để thêm những chức năng gỡ lỗi cho PICkit 2 và (D) cửa sổ Output sẽ trình bày tình trạng giao tiếp giữa PICkit 2 và board mạch đích trên PICkit 2 tab.



Hình 4-7 PICkit 2 Debug Tool

2. Chọn *Debugger>Settings* để thiết lập cho PICkit 2. Chắc chắn "Connect on Startup" checkbox phải được đánh dấu để cho phép đặc tính tự động kết nối feature (Hình 4-8). Rồi nhấn **OK**.



Hình 4-8 SETTINGS DIALOG

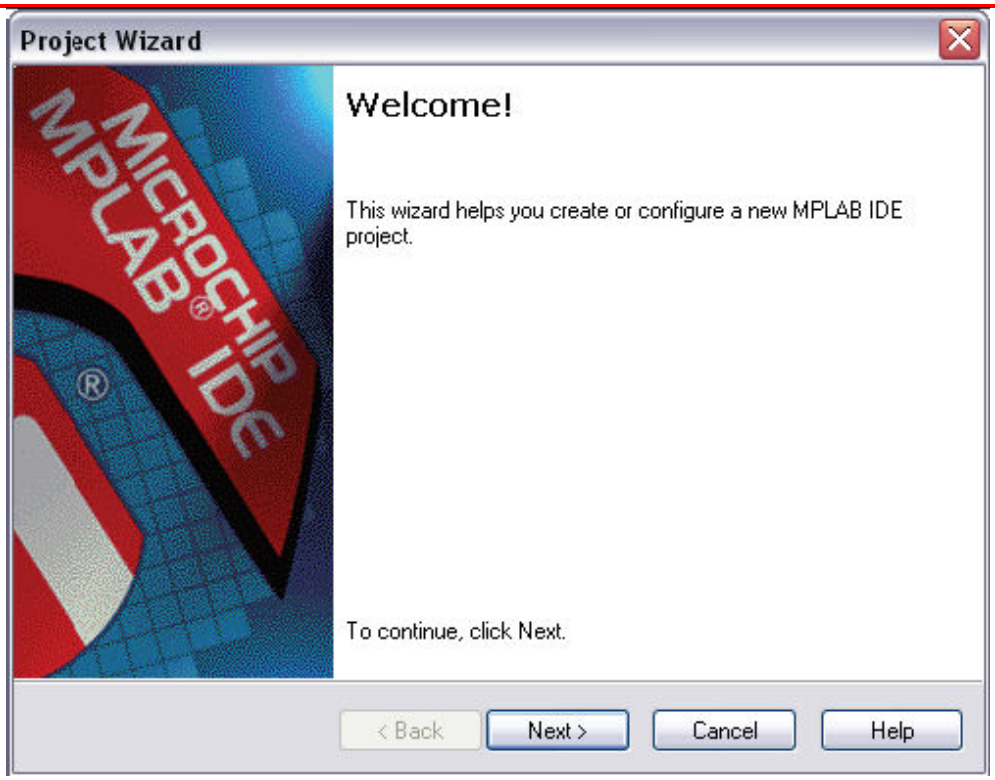
3. Nếu PICkit 2 không tự động kết nối khi nó được lựa chọn như một công cụ gỡ lỗi, bây giờ Chọn *Debugger>Connect* để kết nối. Tình trạng kết nối sẽ hiện ra trong cửa sổ Output. Tùy thuộc vào phiên bản của phần mềm MPLAB IDE hoặc Chip được lựa chọn, một thông báo có thể xuất hiện chỉ báo Firmware (PICkit 2 operating system) Cần cập nhật. MPLAB IDE sẽ tự động cập nhật firmware mới.

Tạo một MPLAB IDE Project

Một MPLAB IDE Project và không gian làm việc sẽ lưu giữ tất cả file và thiết đặt cho cùng một dự án phát triển. Project Wizard sẽ giúp bạn thiết lập một project mới.

Select *Project*> to set up the project. The screen will display (Figure 4-9). Click **Next** to continue to Step One.

1. Chọn *Project > Project Wizard* để thiết lập project. Sẽ hiện ra màn hình Project Wizard Welcome (Hình 4-9). Kích **Next** để tiếp tục bước 1.



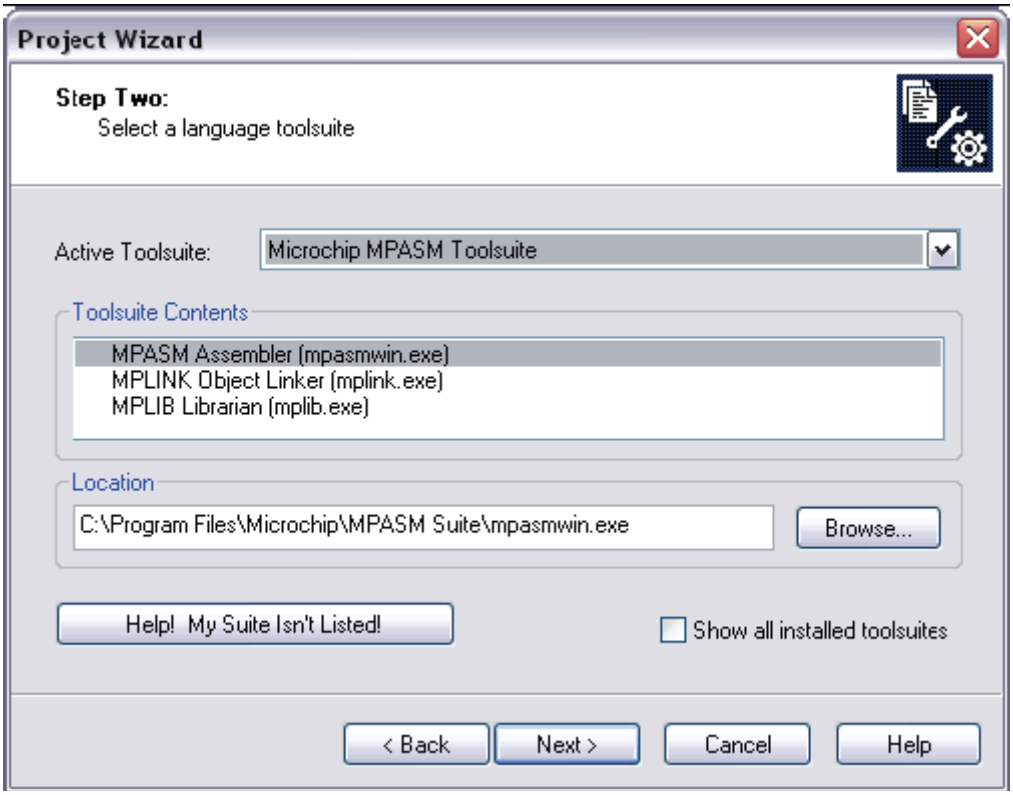
Hình 4-9 PROJECT WIZARD WELCOME

1. Chọn Chip **PIC16F887 tử** "Device" drop-down box, nếu nó chưa được chọn (Hình 4-10). Click **Next** để tiếp tục bước 2.

Hình 4-10 BƯỚC 1 – CHỌN CHIP

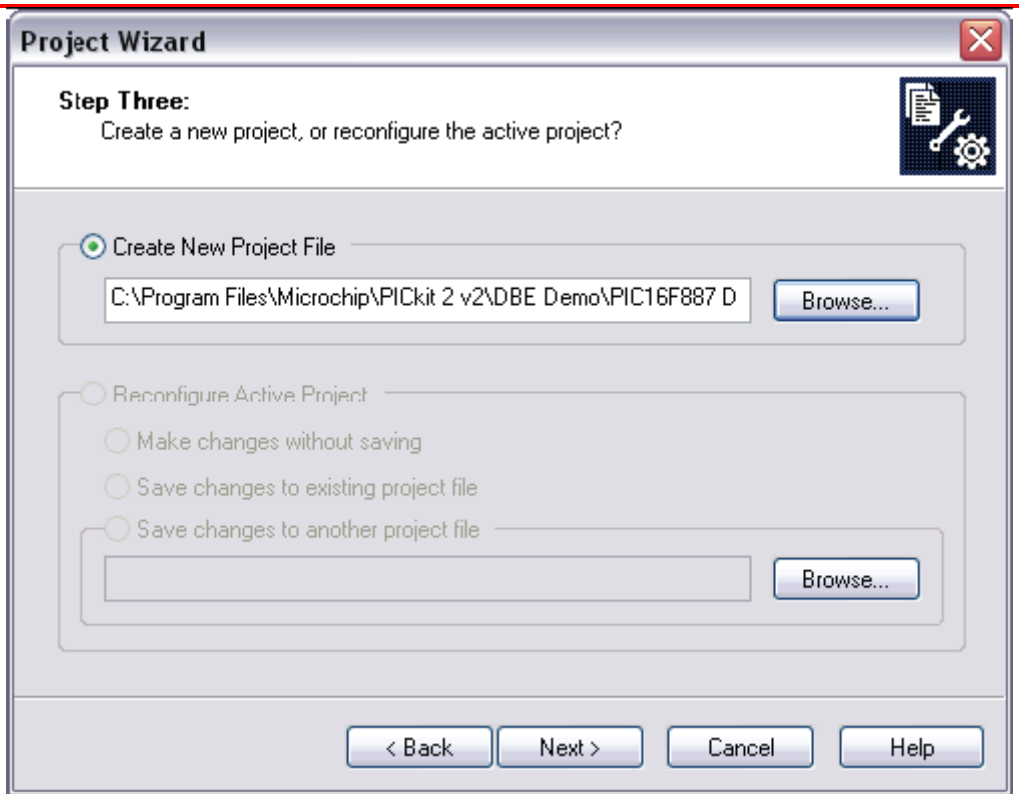
2. Cho project này, MPASM assembler tool sẽ được sử dụng. chọn "Microchip MPASM Toolsuite" từ Active Toolsuite drop-down menu (hình 4-11). Chắc chắn rằng các tool được set theo mặc định trong folder sau: C:\Program Files\Microchip\MPASM Suite:
   - MPASM assembler cần phải đang trỏ vào mpasmwin.exe.
   - MPLINK. object linker cần phải đang trỏ vào mplink.exe.
   - MPLIB. object librarian cần phải đang trỏ vào mplib.exe.

   Click **Next** để tiếp tục sang bước 3



HÌNH 4-11: BƯỚC 2 - CHỌN BỘ NGÔN NGỮ LẬP TRÌNH

3. Click **Browse** (Hình 4-12) để định vị hay tạo ra một thư mục project dự án mới và tên project. trong hướng dẫn này, chọn vị trí **C:\Program files\Microchip\PICkit 2 v2\DBE Demo** và đặt tên cho project, ví dụ như "PIC16F887 Debug Demo"



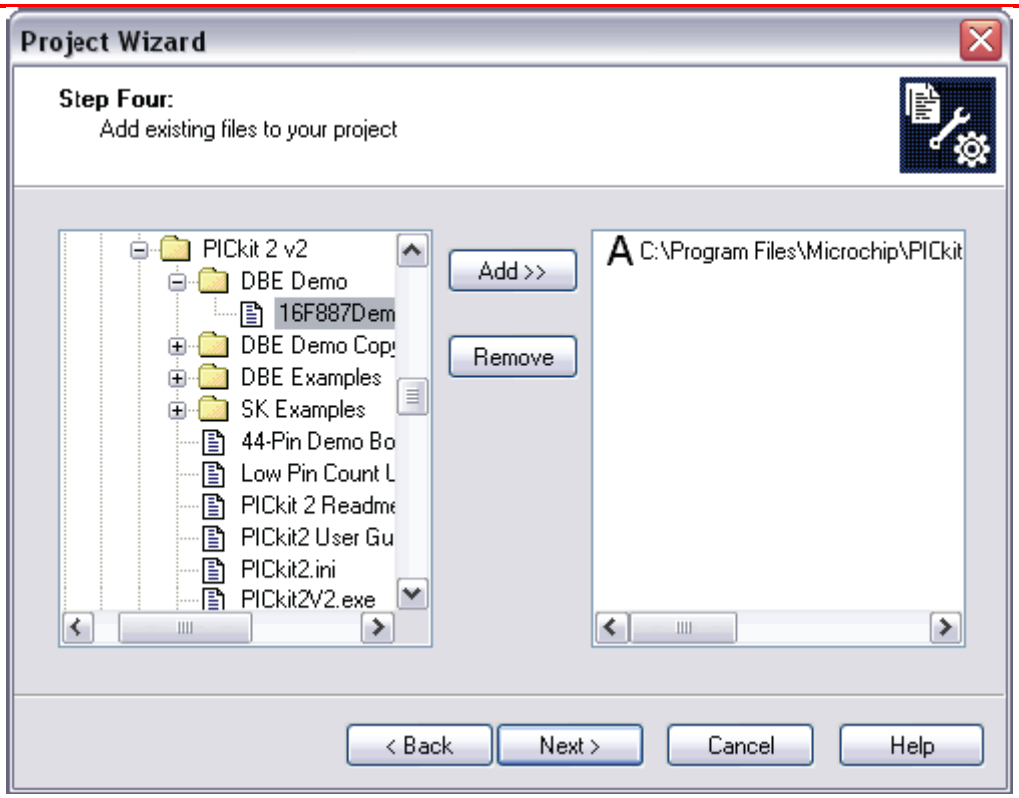
HÌNH 4-11: TẠO MỘT PROJECT MỚI

4. Thêm một file nguồn cho project (Hình 4-13) từ cửa sổ bên trái đi đến `C:\Program Files\Microchip\PICkit 2 v2\DBE Demo`. Chọn và bật sáng file 16F887demo.asm và nhấn **Add**. File này sẽ được đặt vào trong cửa sổ bên phải

Ghi chú: những file khác có thể bổ sung sau đó.

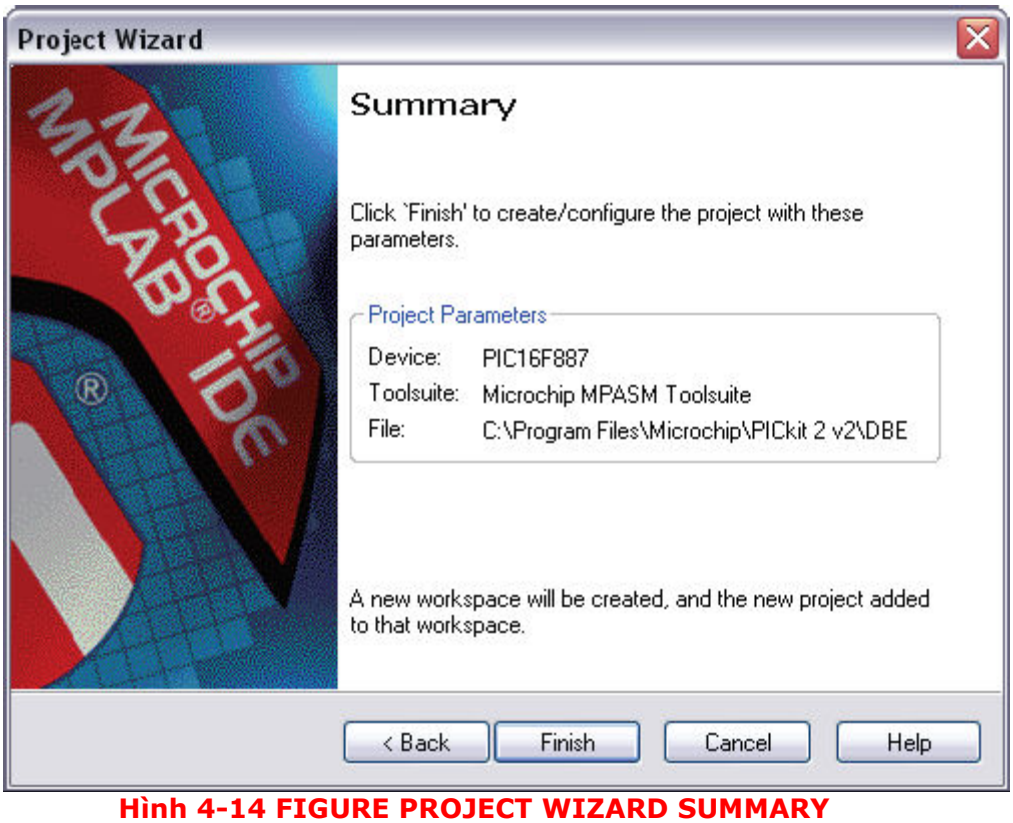
Dấu "A" trong hình 4-13 cho phép MPLAB IDE quyết định đường dẫn tới file là tương đối hay tuyệt đối cho project. Để biết mô tả và cách thay đổi những kiểu thêm file có thể xảy ra trong tài liệu của MPLAB IDE. Không thay đổi thiết đặt cho Project này. Click **Next** để tiếp tục sang cửa sổ tóm lược.

Ghi chú: với những project chứa nhiều file assembly (ví dụ Multifile project, C code project) bạn sẽ cần thêm một linker script file. Xem tài liệu công cụ ngôn ngữ cho thấy nhiều chi tiết hơn.



HÌNH 4-13 BƯỚC 4 – THÊM FILE

5. Nếu có lỗi trong khi làm, nhấn **Back** để trở lại bất kỳ thao tác nào trong Project Wizard (Hình 4-14) nếu không nhấn **Finish**

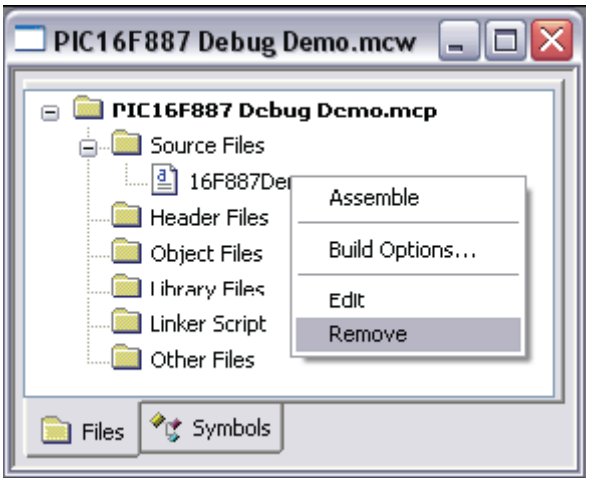


Hình 4-14 FIGURE PROJECT WIZARD SUMMARY

Xem Demo Project

Sau khi hoàn thành một project và thoát ra khỏi Project Wizard, cửa sổ Project sẽ trình bày trong MPLAB IDE desktop window (Hình 4-15). Nếu nó không mở ra thì chọn *View>Project* để mở nó lên.

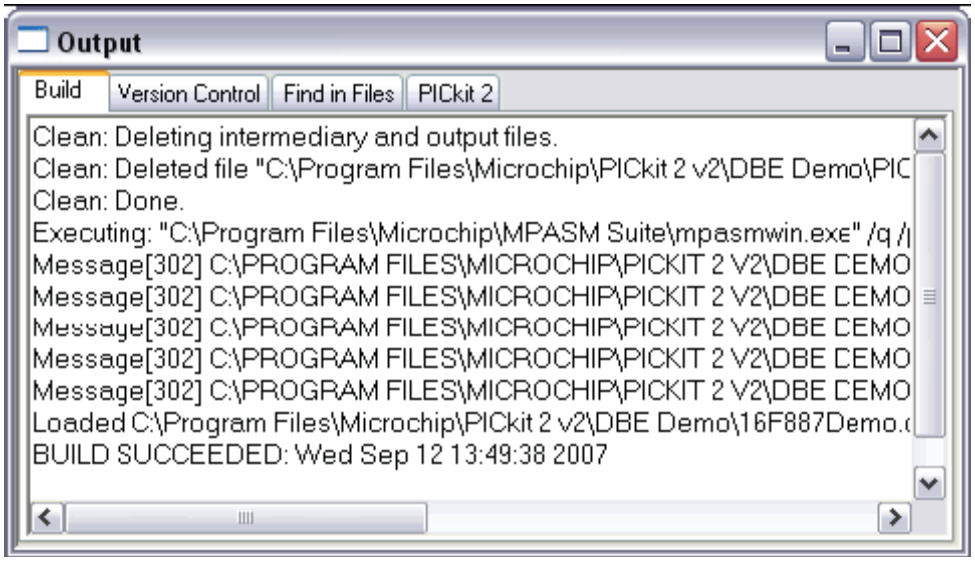
Khi cần, những file project có thể thêm vào hoặc loại bỏ bớt từ Project Window. Nhấn phím phải của chuột vào file trong Project Window để hiện ra pop-up menu với những tùy chọn, ở đó có thể thêm hoặc bỏ bớt file.



HÌNH 4-15 PROJECT WINDOW FILE MENU

Tạo ra một File HEX

Để tạo ra một file HEX để nạp vào Chip, bạn cần phải Build project này. Chọn *Project>Build All* hoặc right click vào tên project trong Proect Window và "Build All" từ pop-up menu MPASM assembler sẽ tạo ra một file HEX có tên trùng với tên của file .asm. Diễn tiến của quá trình sẽ rõ ràng trong thẻ **Build** của output window (Hình 4-16)



HÌNH 4-16: OUTPUT WINDOW . BUILD THE PROJECT

Kiểm tra giá trị Configuration Bit

Những Configuration bit sẽ được nạp và bên trong Chip là được set từ chương trình sử dụng chỉ thị _CONGIF. Một Project được xây dựng, những giá trị của những bits này có thể được xác minh sử dụng Config Bit Window chọn *Configure>Configuration Bits* (Hình 4-17).

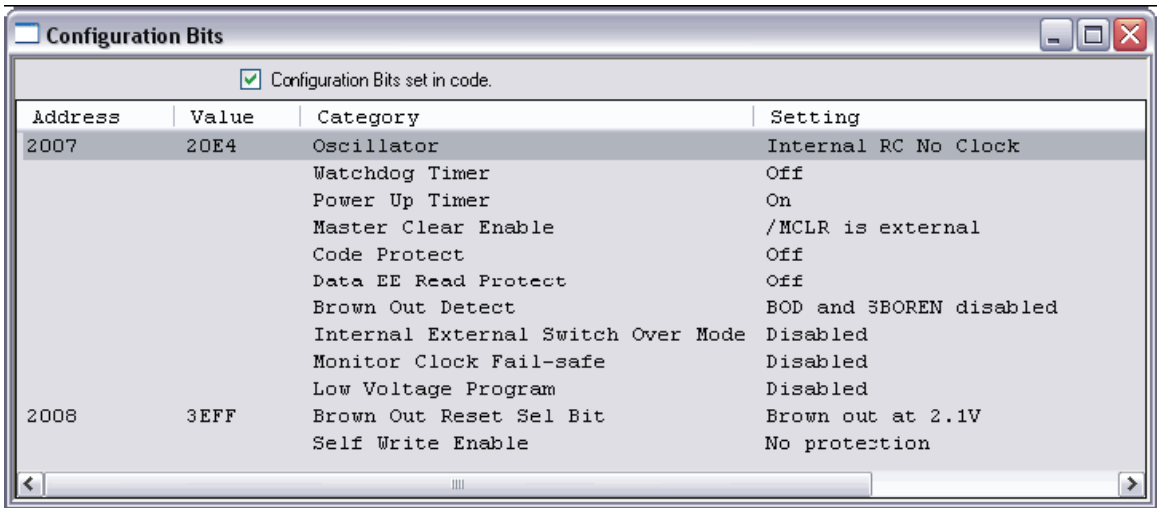
Những Congif bits sau đây được sử dụng trong hướng dẫn này:

Config1:

- Oscillator - Internal RC No Clock
- Watchdog Timer - Off
- Power-Up Timer - On
- Master Clear Enable - MCLR is external
- Code-Protect - Off
- Data EE Protect - Off
- Brown-Out Detect - BOD and SBOREN Disabled
- Internal-External Switch Over Mode - Disabled
- Monitor Clock Fail-safe - Disabled
- Low-Voltage Program – Disabled

Config 2:

- Self Write Enable . No Protection
- Master Brown-out Reset Sel Bit . Brown-out at 2.1V



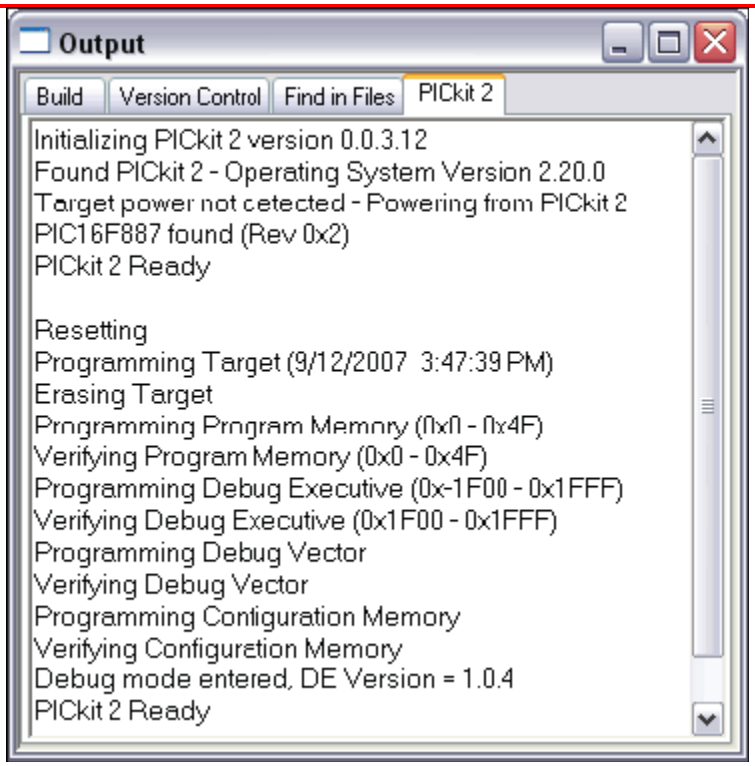
HÌNH 4-17 CONFIGURATION BIT SETTINGS

Tải mã chương trình để Debugging

Để nạp cho Chip, chọn *Debugger>Program* để nạp file *16F887demo.hex* vào Chip 16F887 trên board demo.

Việc nạp Chip chỉ mất vài giây, trong quá trình nạp thẻ **PICkit 2** của cửa sổ Output sẽ hiển thị những dòng các thao tác đã thực hiện. Đến khi nạp đầy đủ thì hộp thoại cần phải nhìn tương tự như hình 4-18

Ghi chú: Khi chạy debug thì code sẽ tự động nạp vào trong bộ nhớ chương trình (program memory) của PIC16F887 (Chip đích) khi PICkit 2 được chọn như một debugger. Debug code thì được nạp vào trong Chip đích để sử dụng những khả năng in-circuit debugging của PICkit 2



HÌNH 4-18 OUTPUT WINDOW – NẠP CHIP CHO DEBUG

Chạy PIC16F887 debug demo

The PICkit 2 Debug Express thực hiện chương trình trong thời gian thực (real-time Run) hặc chạy từng bước (Step Into, Step Over, Step Out, Animate.) chạy thời gian thực xảy ra khi bạn chọn **Run** trong MPLAB IDE. Khi chương trình được dừng bởi cả 2 thao tác **Halt** hoặc breakpoint (Điểm dừng), bạn có thể đi từng bước thông qua mã.

Những nút sau đây có thể thường sử dụng cho việc truy nhập nhanh khi thao tác debug:

Debugger Menu	Toolbar Buttons
Run	
Halt	
Animate	
Step Into	
Step Over	
Step Out	
Reset	

Để chạy Demo code:

1. Double click lên file 16F887Demo.asm từ Project window hoặc chọn *File>Open* từ toolbar menu. Code sẽ hiện ra một cửa sổ file này.
2. chọn *Debugger>Run* Hoặc click **Run**.
3. xoay biến trở (RP1), được gắn trên demo board, và quan sát LEDs.
   Nếu chương trình hoạt động đúng thì các led sẽ chạy nhanh hay chậm hơn tùy thuộc vào chiều và vị trí của RP1, tuy nhiên có một "bẫy" cố ý đặt trong mã với mục đích trình diễn cho debug. Xem mục **4.5.9 "Debugging the PIC16F887 Debug Demo Code"** cho chỉ dẫn cách gỡ lỗi.
4. chọn *Debugger>Halt* hoặc click **Halt** để ngưng chạy chương trình. Một mũi tên màu xanh sẽ đánh dấu vị trí dòng code trong của sổ file khi chương trình dừng
5. chọn *Debugger>Reset>Processor Reset* để khởi động chương trình. Mũi tên biến mất, có nghĩa thiết bị đã được Reset lại.

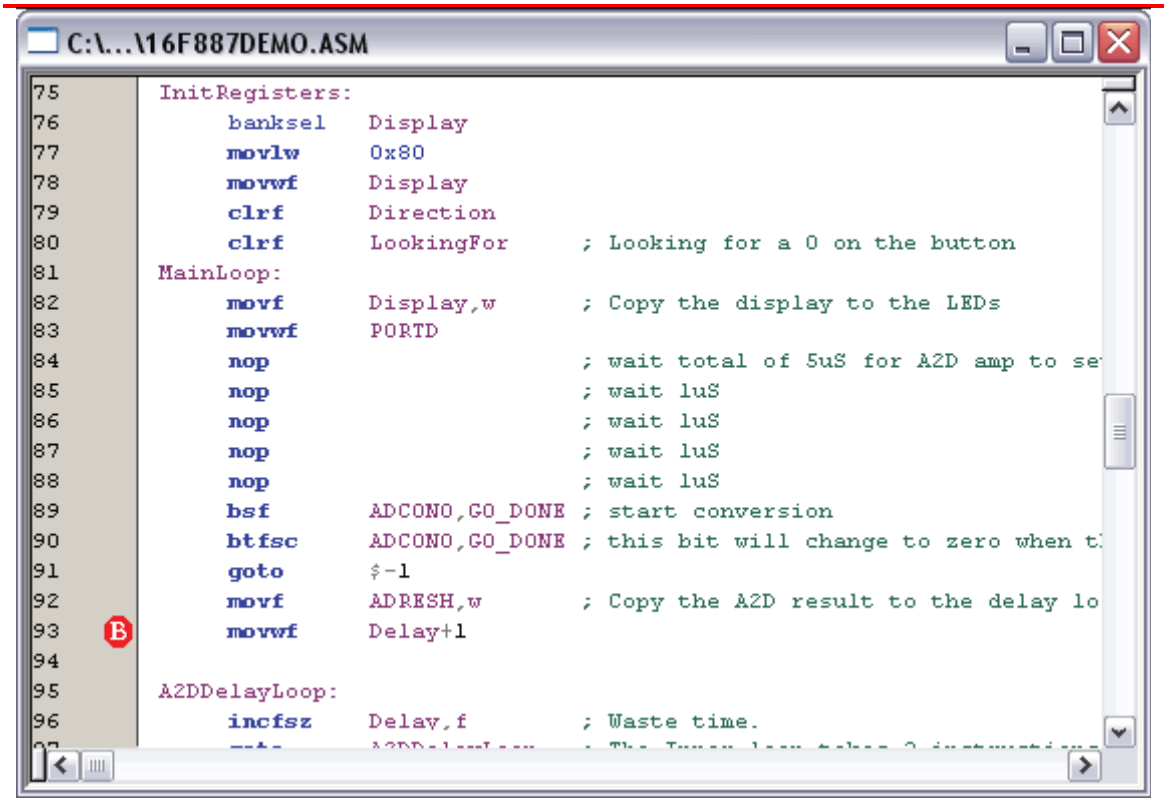
9 Debugging the PIC16F887 Debug Demo Code

Bất kỳ những vấn đề sau đây co thể làm PIC16F887 làm việc không đúng

1. Giá trị A/D converter không được viết đúng mức đến delay routine.
2. A/D converter không được chọn (enable) hay không thiết lập chuyển đổi.
3. Một lỗi đánh máy trong chương trình có thể gây ra chương trình vận hành không thích hợp.

Để khám phá vấn đề được liệt kê đầu tiên, đặt một điểm dừng tại dòng của code mà nó ghi tới giá trị A/D kết quả high-oder delay byte:

1. Đặt một con trỏ trên hàng sau đây trên dòng của code trong file 16F887demo.asm: **movwf delay+1** nhìn thấy như hình 4-19. Tại điểm dừng này, chương trình sẽ dừng lại một lần khi A/D biến đổi hoàn thành.
2. Right click 2 lần lên dòng đó để hiện ra drop-down menu và chọn "Breakpoint" hoặc double click lên dòng một kí tự breakpoint sẽ xuất hiện tiếp đến dòng như chữ B trong hình tám cạnh màu đỏ như nhìn thấy trong hình 4-19



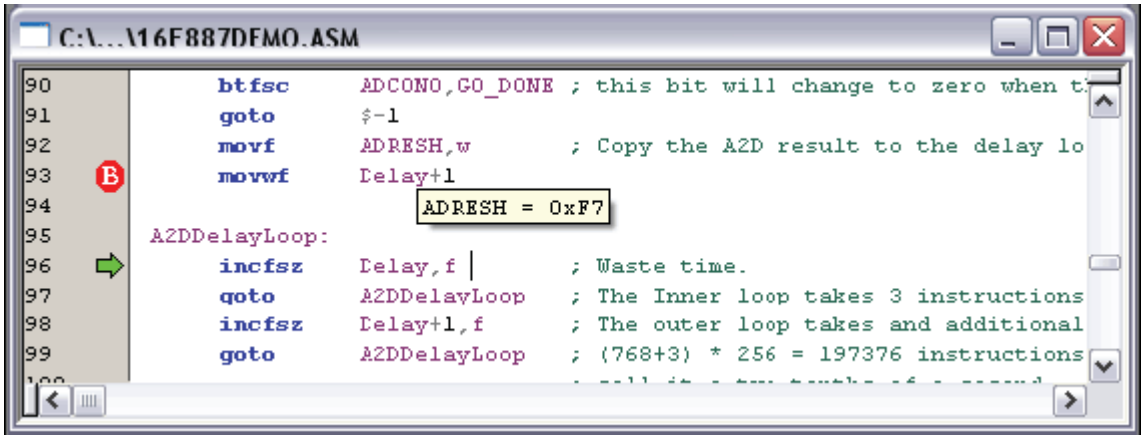
HÌNH 4-19: SET BREAKPOINT

3. Chọn *Debugger>Run*, hoặc click **Run** để chạy chương trình. Điểm dừng sẽ dừng chương trình lại khi chương trình chạy đến dòng có đánh dấu như một điểm dừng

   Ghi chú: tùy thuộc vào Chip , điểm dừng có thể "trượt" qua khỏi vị trí định vị, trong trường hợp này nó sẽ dừng tại dòng :

   ```
   incfsz Delay,f
   ```

4. đưa muose qua ADRESH trong dòng trên điểm dừng và nó sẽ hiển thị giá trị của file register (Hình 4-20).

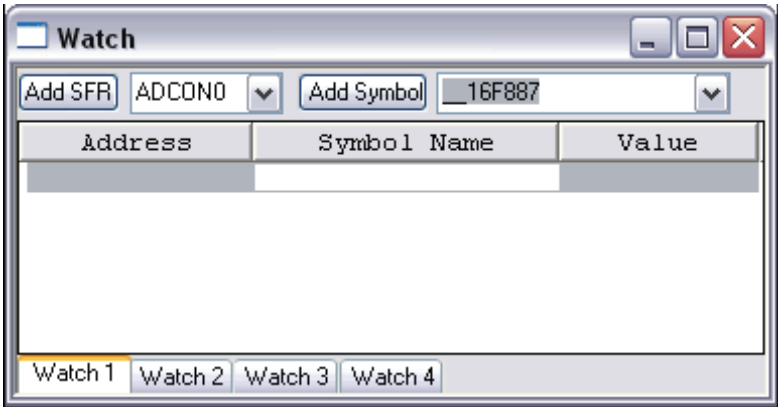


HÌNH 4-20 GIÁ TRỊ THANH GHI ADRESH

5. Điều chỉnh biến trở RP1 và tiếp tục chạy chương trình bằng cách chọn *Debugger>Run* chương trình sẽ chạy vòng qua và dừng lại lần nữa.
6. Đưa mouse qua ADRESH lần nữa và xem kết quả A/D không hề thay đổi. Như vậy là hình như biến đổi A/D không làm việc. bộ biến đổi A/D được khởi tạo và cài đặt xảy ra ở đầu chương trình.
7. chọn *Debugger>Reset* để reset lại chương trình

Select *View>Watch* to open a new Watch window. This window will allow you to watch the A/D register value change as the program executes. The Watch dialog opens with the **Watch 1** tab selected, as shown in Figure 4-21.

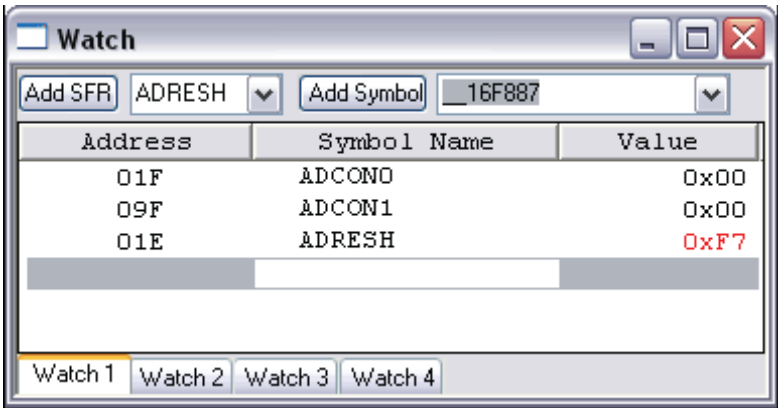
8. Chọn *View>Watch* để mở một cửa sổ Watch, cửa sổ này cho phép bạn đi tới thay đổi giá tri thanh ghi A/D như chương trình đang chạy. Hộp thoại Watch mở ra có một bảng chọn **Watch 1** như trình bày trên hình 4-21



HÌNH 4-21 NEW WATCH WINDOW

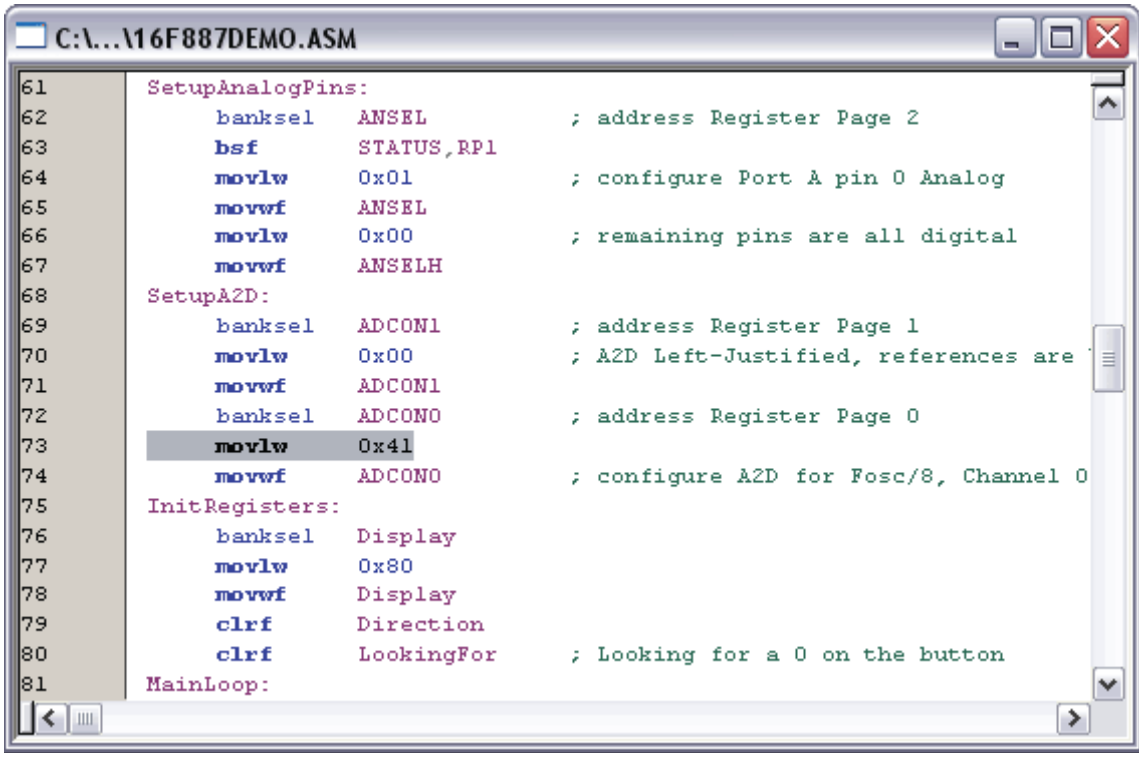
Ghi chú: Nó không đề nghị sử dụng cửa sổ *View>File Registers* và *View>Special Function Registers* khi gỡ lỗi với PICkit 2 Debug Express. Khi cửa sổ này mở, toàn bộ nội dung được ghi rồi đọc lại từ chip đích trong mọi thao tác, điều này làm chậm quá trình debug một cách đáng kể. Thay vào đó, chỉ thêm SFR và biến file Register mà ta quan tâm đến cửa sổ Watch như được mô tả trong phần hướng dẫn này.

9. chọn ADCON0 từ đầu danh sách sổ xuống và click **Add SFR** để thêm ADCON0 vào cửa sổ Watch. Lặp lại điều này để thêm ADCON1 và ADRESH vào cửa sổ Watch. FSRs được lựa chọn thì cần phải rõ ràng như trong hình 4-22



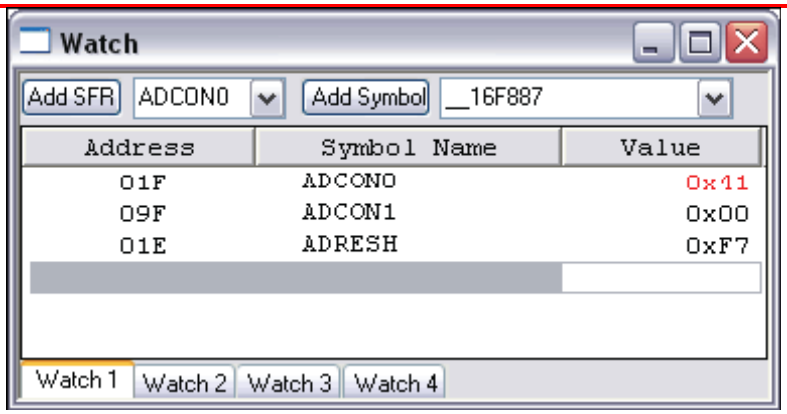
HÌNH 4-22 THÊM SFRs VÀO CỬA SỔ WATCH

10. Chọn *Debugger>Run* để chạy chương trình lần nữa, chương trình sẽ dừng lại sau khi nó đến vị trí điểm dừng.
11. Khảo sát những giá trị ADCON0 và ADCON1 trong cửa sổ Watch, giá trị ADCON0 là "0x40" (b'01000000'). Giá trị này tương ứng với giá trị hex được chỉ định trong chương trình, tuy nhiên đây thì không đúng. Xem lại *"PIC16F882/883/884/886/887 Data Sheet"* (DS41291), mục Analog – to – Digital (A/D) module, chỉ báo cuối cùng của bit này là "1" (b'01000001') để bật module A/D. để cố định cái nhiễu này, thay đổi:
movlw 0x40
Thành
movlw 0x41 trình bày như hình 4-23



HÌNH 4-23: AD MODULE CODE

12. Chọn *file>save* để lưu sự thay đổi của bạn
13. bạn phải build lại và nạp lại cho Chip để thay đổi hiệu ứng. Chọn *Project>Build All* để xây dựng lại project và *Debugger>Program* để nạp lại cho Chip sử dụng PICkit 2
14. Chọn *Debugger>Run* để chạy chương trình lần nữa, chương trình sẽ dừng lại sau khi nó đến vị trí điểm dừng.
15. Khảo sát những giá trị trong cửa sổ Watch, bây giờ, giá trị ADCON0 cần phải là "0x41" (Hình 4-24).



HÌNH 4-24 GIÁ TRỊ ĐÚNG ADCON

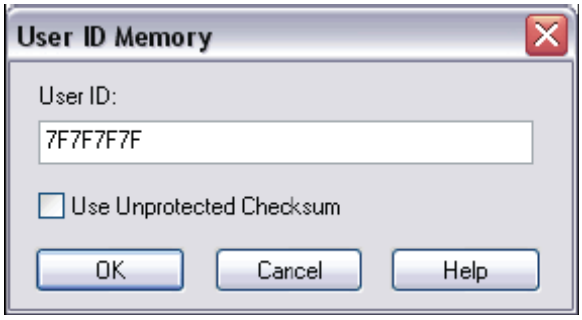
16. Right click trên dòng của code có đán dấu điểm dừng, và chọn *Remove>BreakPoint* hoặc double click trên dòng đó để gỡ bỏ điểm dừng.
17. Chọn *Debugger>Run* để chạy chương trình thời gian thực, xoay biến trở RP1 để thay đổi giá trị hiển thị trên LEDs, Bây giờ nó đã làm việc.

Mã nguồn trong hướng dẫn này chứa đựng chỉ một bẫy. Tuy nhiên, mã thực sự có thể có nữa. Sử dụng MPLAB IDE với chức năng debug của PICkit 2 một cách thành công, bạn có thể tìm thấy và fix những vấn đề trong code của các bạn.

Ứng dụng Lập trình và Gỡ lỗi

Khi chương trình thành công được gỡ lỗi và chạy, bước tiếp theo là nạp chương trình vào chip và chạy độc lập trong khi kết thúc thiết kế. Khi làm điều này, những tài nguyên dự trữ bởi ICD được thả tự do cho sử dụng bởi ứng dụng. để nạp chương trình ứng dụng, sử dụng những bước sau:

1. Bỏ chọn PICkit 2 như một công cụ debug bằng cách chọn *Debugger>Select Tool>None*
2. Chọn PICkit 2 như một công cụ nạp trong menu *Program>Select Programmer*
3. Lựa chọn: Thiết lập ID trong *Configure>ID Memory* (Hình 4-25)



HÌNH 4-25 SỬ DỤNG ID MEMORY

4. Thiết lập những thông số để lập trình trên *Programmer>Settings* , Bảng chương trình
5. Chọn *Programmer>Program*

Bây giờ Chip đích sẽ chạy độc lập