

# EASYPIC\_PRO PIC DEVELOPMENT BOARD

## Chương 1: Giới thiệu EASYPIC\_PRO Development Board System

1.1 Tổng quan sản phẩm .....	2
1.2 Giới thiệu tài nguyên board .....	3
1.3 Sơ đồ nguyên lý EasyPIC_PRO .....	4

## Chương 2: MPLAB IDE integrated environment

2.1 Cài đặt MPLAB.....	4
2.2 Ứng dụng đơn giản trong MPLAB .....	4

## Chương 3: Sử dụng On-line Programmer/Debugger

3.1 Cách sử dụng PICKit2 on-board programmer/ Debugger .....	9
3.2 Các công cụ Programmer/debugger tương thích với EasyPIC_PRO .....	10

## Chương 4: EASYPIC\_PRO Chức năng chi tiết các Module

4.1 MCU Socket and System Clock selection .....	12
4.2 POWER SUPPLY MODULE.....	15
4.3 128x64 LCD and 16x02 LCD Modules .....	15
4.4 SD/MMC Card Read/Write Module .....	16
4.5 LED đơn PORTB và PORTC .....	17
4.6 4x4 Matrix Keyboard .....	18
4.7 Push button switch module .....	20
4.8 DS1302 Real time clock module .....	21
4.9 A/D Converter Module.....	22
4.10 Giao Thức I2C và EEPROM 24CXX Module .....	23
4.11 Giao Thức SPI và 93LCXXX EEPROM Module .....	24
4.12 7- Segment display Module .....	26
4.13 IR Remote Control Receiver Module .....	27
4.14 DS18B20 Thermometer Module.....	28
4.15 BUZZER Module .....	29
4.16 Stepper Motor Module .....	30
4.17 RS232 Communication Module .....	32
4.18 USB Communication Module .....	33
4.19 PS/2 Communication Module .....	34
4.20 Direct PORT Access .....	34

## Chương 5: Thực hành thực tế board phát triển EASYPIC\_PRO .....

## Phụ lục 1: Thông tin liên hệ và chính sách bảo hành .....

## Chương 1 EASYPIC\_PRO Development Board System Introduction

### 1.1 Tổng quan sản phẩm

**EASYPIC\_PRO PIC DEVELOPMENT BOARD** (sau đây sẽ gọi là EasyPIC\_PRO) là board mạch phát triển PIC microcontroller đa chức năng. Thiết kế bởi [Thiên Minh Electronic Solutions Co., Ltd.](#) Dựa trên kinh nghiệm nhiều năm với các board mạch phát triển và công cụ lập trình đã được người dùng đánh giá cao.

**EASYPIC\_PRO** tích hợp nhiều thành phần ngoại vi. Có một số lượng lớn các ví dụ về các chương trình mà công ty chúng tôi cung cấp bao gồm ngôn ngữ lập trình ASM, C (HTPIC và CCS). Bảo đảm tương thích cho nhiều đối tượng, giúp bạn làm chủ công nghệ lập trình vi điều khiển trong thời gian ngắn nhất. EasyPIC\_PRO là công cụ hỗ trợ đắc lực cho các Kỹ sư lập trình PIC MCU dùng để chạy thử các ứng dụng của mình trước khi sản xuất, Đặc biệt EasyPIC\_PRO là một công cụ rất thích hợp cho việc giảng dạy tại các trường đại học, cao đẳng và THCN, cũng như những người tự học lập trình với PIC MCU.

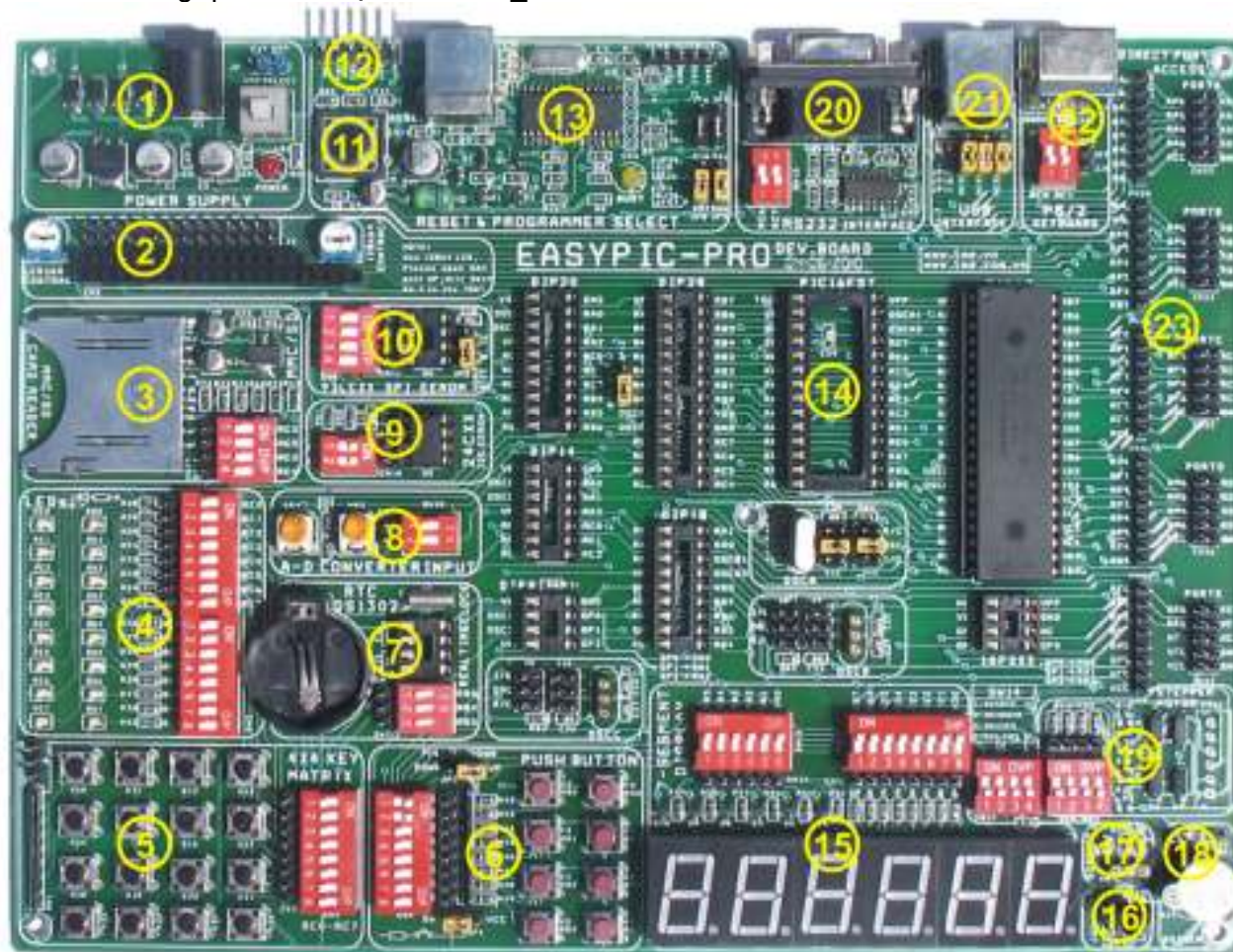
**EASYPIC\_PRO** Development Board có tính tùy biến cao, tích hợp sẵn PICKit 2 on-board programmer/debugger cho các MCU trên board và có thể dùng như một mạch nạp độc lập cho MCU bên ngoài, ngoài ra bạn có thể sử dụng các công cụ lập trình và gỡ lỗi bên ngoài khác như PICKit 3, ICD2 ...v.v

**EASYPIC\_PRO** là sự lựa chọn khôn ngoan cho người dùng PIC MCU, Sự lựa chọn khôn ngoan của bạn có thể được thể hiện ở các điểm sau.

- Board giàu tài nguyên, khả năng mở rộng kết nối vô hạn, tùy biến cao.
- Công nghệ sản xuất tiên tiến
- Hỗ trợ nhiều ví dụ mẫu phù hợp với nhiều đối tượng sử dụng trình biên dịch khác nhau như ASM, CSS, HI-TECH...
- Tối ưu hóa thiết kế kiểu modules, dễ dàng kết nối hoặc ngắt khỏi MCU bằng Coding Switch hoặc jumper, ngoài ra còn có thêm các Header giao tiếp giúp người dùng có thể kết nối đến Port khác hoặc MCU khác bằng [dupont wire](#) mà không ảnh hưởng đến hoạt động bình thường của toàn hệ thống
- Giá bán hợp lý
- Hướng dẫn Kỹ thuật toàn diện
- Luôn cập nhật các chương trình mẫu mới.
- Dịch vụ sau bán hàng hoàn hảo, bảo hành chu đáo, bảo trì suốt thời gian sử dụng

## 1.2 Giới thiệu tài nguyên board

Hình 1-1 là tổng quan của một EASYPIC\_PRO



**Hình 1-1** Tổng quan các thành phần trên EASYPIC\_PRO

- |                                  |                                          |
|----------------------------------|------------------------------------------|
| 1) Power supply                  | 13) PICKit2 on-board Programmer          |
| 2) LCD1602 and GLCD128x64 module | 14) MCU Sockets DIP8/14/18/28/40 Pin     |
| 3) MMC/SD read/write module      | 15) Six-digital 7-segment display module |
| 4) Single LED PORTC and PORTD    | 16) Remote control receiver & decoder    |
| 5) 4*4 matrix keyboard module    | 17) DS18B20 thermometer module           |
| 6) Push button switch module     | 18) Buzzer module                        |
| 7) DS1302 Real time clock module | 19) Stepper motor control module         |
| 8) A/D converter module          | 20) RS232 communication module           |
| 9) I2C 24XXX module              | 21) USB Communication module             |
| 10) SPI eeprom 93xxx module      | 22) PS/2 keyboard module                 |
| 11) MCU RESET Button             | 23) Direct Port Access                   |
| 12) External ICSP Header         |                                          |

### 1.3 Sơ đồ nguyên lý EASYPIC\_PRO

Vui lòng xem sơ đồ kèm theo trong đĩa CD

## Chương 2 MPLAB IDE integrated environment

MPLAB IDE (sau đây gọi là MPLAB) là phần mềm mạnh mẽ tích hợp môi trường phát triển cung cấp bởi Microchip cho các PIC microcontroller của họ. Nó cho phép người dùng tạo mới, lưu trữ, soạn thảo hoặc chỉnh sửa và biên dịch chương trình cho PIC Microcontrollers trên hệ thống máy tính của mình, và thậm chí nó còn có chức năng mô phỏng năng động, gỡ lỗi và chạy như các bài tập ảo.

### 2.1 Cài đặt MPLAB

MPLAB là phần mềm hoàn toàn miễn phí cung cấp bởi Microchip. Bạn có thể có được file cài đặt kèm theo trong đĩa CD. Phiên bản mới nhất có sẵn trên trang web của Microchip's: [www.microchip.com](http://www.microchip.com)

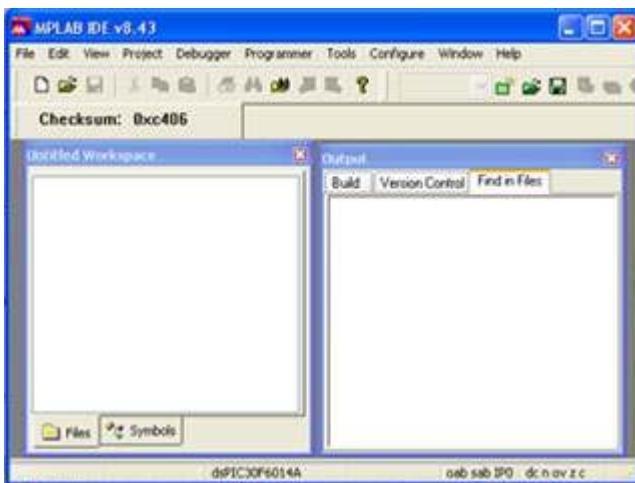
Sau khi download files, bạn chỉ cần sử dụng các phần mềm nén/giải nén ví dụ như WINZIP để giải nén trên máy tính của bạn, và chạy SETUP.EXE (hoặc Install.exe), và thực hiện từng bước gợi ý trong quá trình cài đặt (Bạn cũng có thể không cần thay đổi bất kỳ cài đặt, chỉ cần nhấp "Next") cho đến khi hoàn tất quá trình.

### 2.2 Ứng dụng đơn giản trong MPLAB

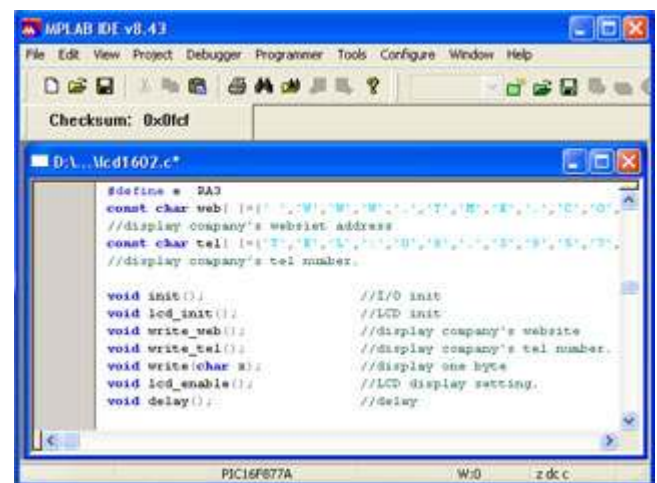
#### 2.2.1 Tạo một Project đơn giản

##### Soạn thảo mã nguồn

Click vào icon MPLAB trên Windows Desktop của bạn, hoặc chọn Start → All application → Microchip → MPLAB IDE V.xx → MPLAB (Vx.x là phiên bản của MPLAB) để bắt đầu chạy MPLAB integrated environment. Trình bày như hình 2 .1.



Hình 2-1 cửa sổ chính MPLAB



Hình 2-2 cửa sổ soạn thảo text

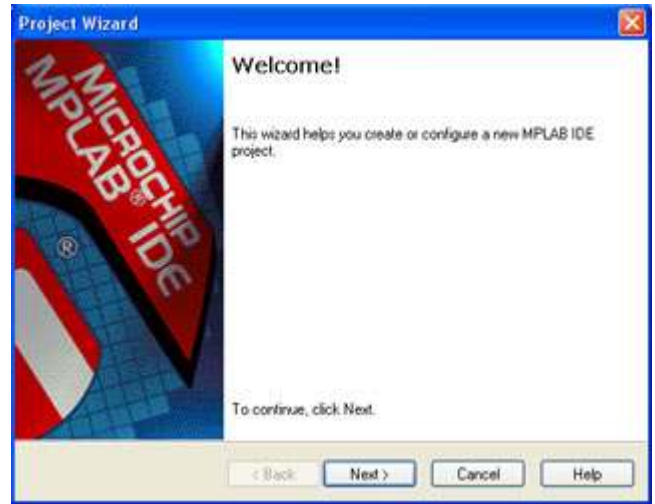
Chọn menu lệnh File → New, MPLAB sẽ tự động gọi MPLAB Editor (source editor), và khu vực làm việc sẽ có một cửa sổ soạn thảo, và bạn có thể hoàn thành mã nguồn của mình như hình 2-2.



Soạn thảo hoặc chỉnh sửa mã nguồn trong " Text editor window ", và sau đó chọn menu lệnh File → Save để lưu source file vào thư mục chỉ định của mình, như trên hình 2-3.



**Hình 2-3** Save source codes



**Hình 2-4** Project wizard welcome interface

Sử dụng wizard để tạo một project

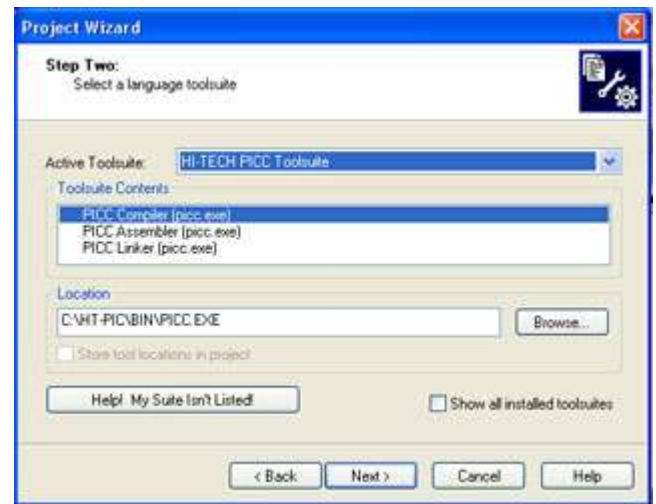
Bước 1, Chọn menu Project → Project Wizard để đến giao diện welcome! Như hình 2-4.

Bước 2, Click "Next", và chọn chip model, như trên hình 2-5.

Bước 3, click "Next", chọn công cụ biên dịch thích hợp cho ngôn ngữ mã nguồn và chip được sử dụng như trình bày trên hình 2-6.



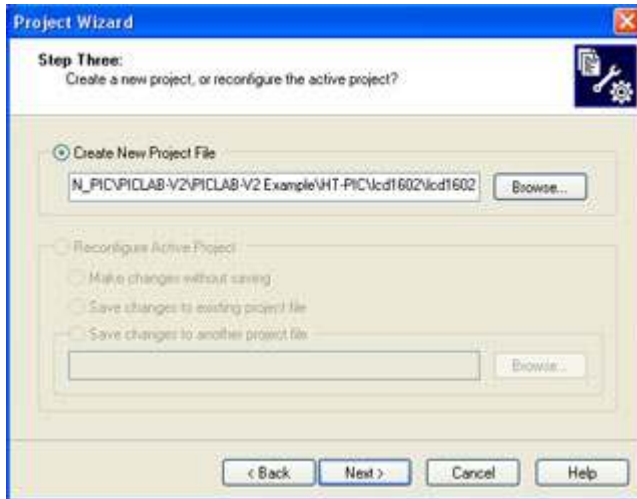
**Hình 2-5** Chọn chip model



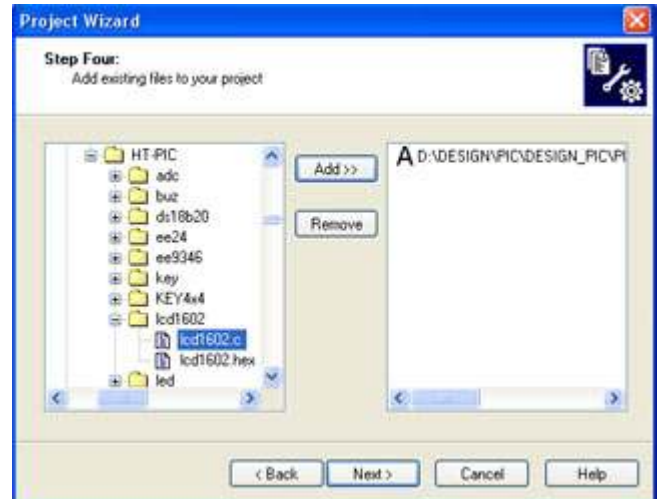
**Hình 2-6** Chọn trình biên dịch

Bước 4, click "Next", chọn thư mục nơi mà project được lưu lại và hoàn tất tên của project, như trình bày trên hình 2-7.

Bước 5, Click "Next", và thêm (add) source codes vào project, như trình bày trên hình 2-8.

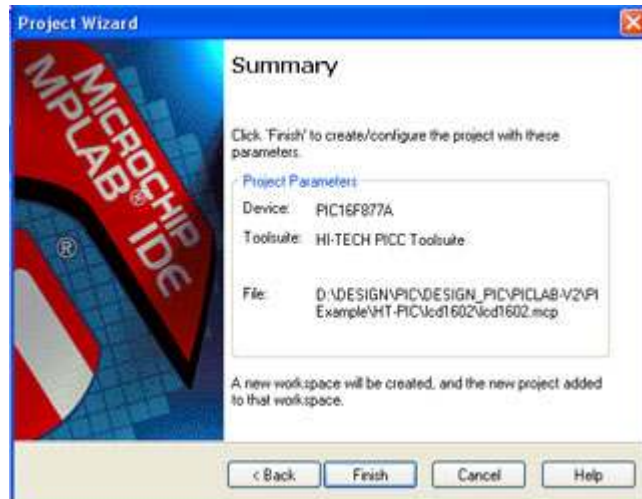


Hình 2-7 Chọn thư mục để lưu project



Hình 2-8 chọn source codes đưa vào project

Bước 6: Click "Next", như trình bày trên hình 2-9 để đến bảng tips interface.



Hình 2-9 Tip interface

Bước 7, Click "Finish", và sẽ thoát wizard.

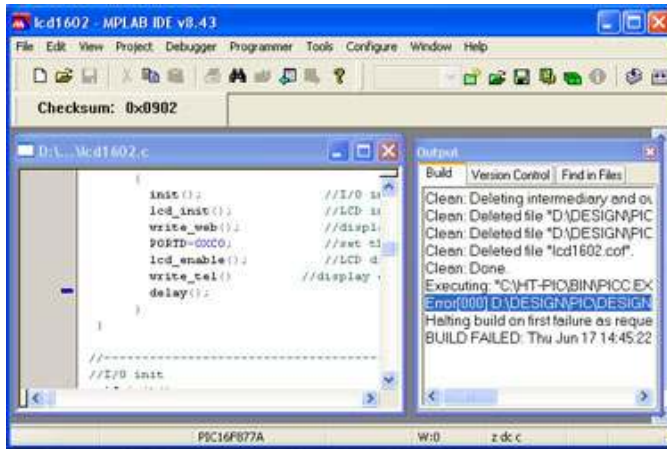
Đến đây, ta đã hoàn thành việc tạo một project mã nguồn. Để hiểu rõ hơn vui lòng đọc thêm MPLAB Operation Manual.

### 2.2.2 Gỡ lỗi chương trình (Debugger program)

Dựa trên mã nguồn đã soạn thảo trong Project đã tạo ra ở trên, phần này giới thiệu ngắn gọn về các bước biên dịch và gỡ lỗi chương trình.

#### Biên dịch (Compile)

Thực hiện menu lệnh Project → Build All và MPLAB sẽ tự động gọi các công cụ được đề cập ở bước 3 để bạn có thể biên dịch mã nguồn này. Khi biên dịch hoàn tất sẽ có giao diện hiển thị như Hình 2-10.



Hình 2-10 Kết quả biên dịch



Hình 2-11 Biên dịch thành công

Từ cửa sổ output của hình 2-10, chúng ta có thể thấy những kết quả mà chương trình biên dịch không thành công vì một lỗi trong mã nguồn, kích đúp vào tin nhắn lỗi này, con trỏ sẽ tự động nhảy đến dòng có lỗi, và tại lề bên trái của dòng này có một "mũi tên màu xanh". Rõ ràng, đằng sau hàm "write\_tel ()" là thiếu ";". Bây giờ sửa lỗi đó và biên dịch nó một lần nữa, các kết quả được thể hiện trong hình 2-11.

Nếu có một lỗi trong chương trình, nó sẽ không tạo ra các file HEX, để có được file HEX bạn cần sửa tất cả các sai sót trong các mã nguồn.

### Gỡ lỗi (Debugging)

Gỡ lỗi chương trình là để kiểm tra xem chương trình bạn thiết kế đang hoạt động, liệu nó tạo ra kết quả đúng như mong muốn, liệu có bất kỳ khiếm khuyết trong thiết kế của bạn, liệu thiết kế thuật toán là hợp lý, liệu nó chính xác có thể kiểm soát các nguồn tài nguyên phần cứng khác nhau, và liệu nó có thể đạt được kết quả mong muốn...

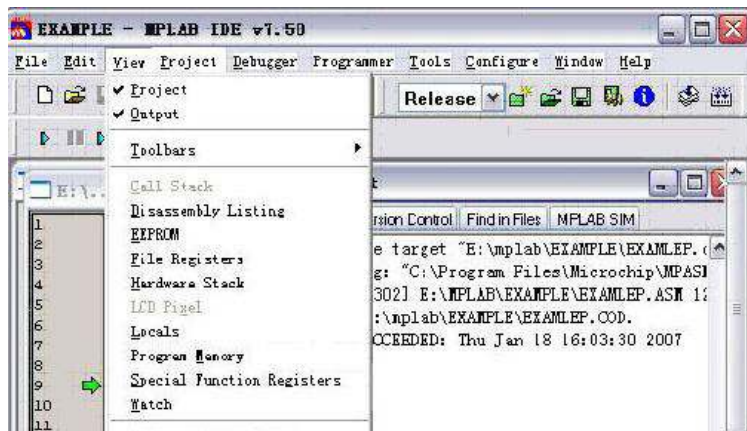
### Chọn công cụ gỡ lỗi (Choose debugging tools)

Chọn menu lệnh Debugger → Select Tool, để chọn simulator kết nối với PC như công cụ gỡ lỗi, hoặc bạn có thể chọn phần mềm sửa lỗi đi kèm với phần mềm MPLAB như dò lỗi cho chương trình đích. Sau khi lựa chọn, nó sẽ mở thanh công cụ tương ứng

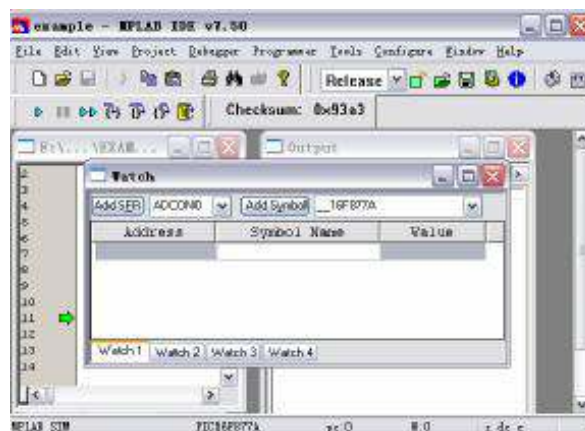
### Quan sát kết quả gỡ lỗi

Các khu vực lưu trữ bên trong vi điều khiển PIC có thể được chia thành nhiều đoạn: Program memory, Hardware stack, File registers, các thanh ghi đặc biệt và EEPROM data memory. Trong quá trình hoạt động của chương trình, nó sẽ liên tục đọc, ghi hoặc thay đổi các nội dung trong khu vực lưu trữ. Vì vậy, chúng ta có thể quan sát những thay đổi của nội dung trong khu vực lưu trữ tương ứng với các hoạt động của chương trình để hiểu được hoạt động của chương trình, và đạt được mục đích của gỡ lỗi. Để mở khu vực lưu trữ chúng ta có thể chọn các menu lệnh View, như trong hình 2-12.

Ngoài việc sử dụng các khu vực lưu trữ này để quan sát quá trình gỡ lỗi, chúng ta cũng có thể thêm các module cụ thể liên quan đến cửa sổ quan sát để theo dõi các kết quả. Thực hiện lệnh menu View → Watch và sẽ mở ra cửa sổ quan sát, như trong hình 2-13



Hình 2-12 Menu lệnh để mở vùng lưu trữ



Hình 2-13 Cửa sổ quan sát

Như vậy, chúng ta đã được giới thiệu một cách đơn giản sử dụng MPLAB, và thông qua việc học tập trong chương này, chúng ta sẽ có thể hoàn thành soạn thảo, chỉnh sửa mã nguồn, biên dịch và gỡ lỗi.

### **Lưu ý:**

**Thông tin chi tiết trên MPLAB xin vui lòng vào trang web của Microchip và tải "MPLAB User Manual". Hoặc xem file kèm theo trong đĩa CD**

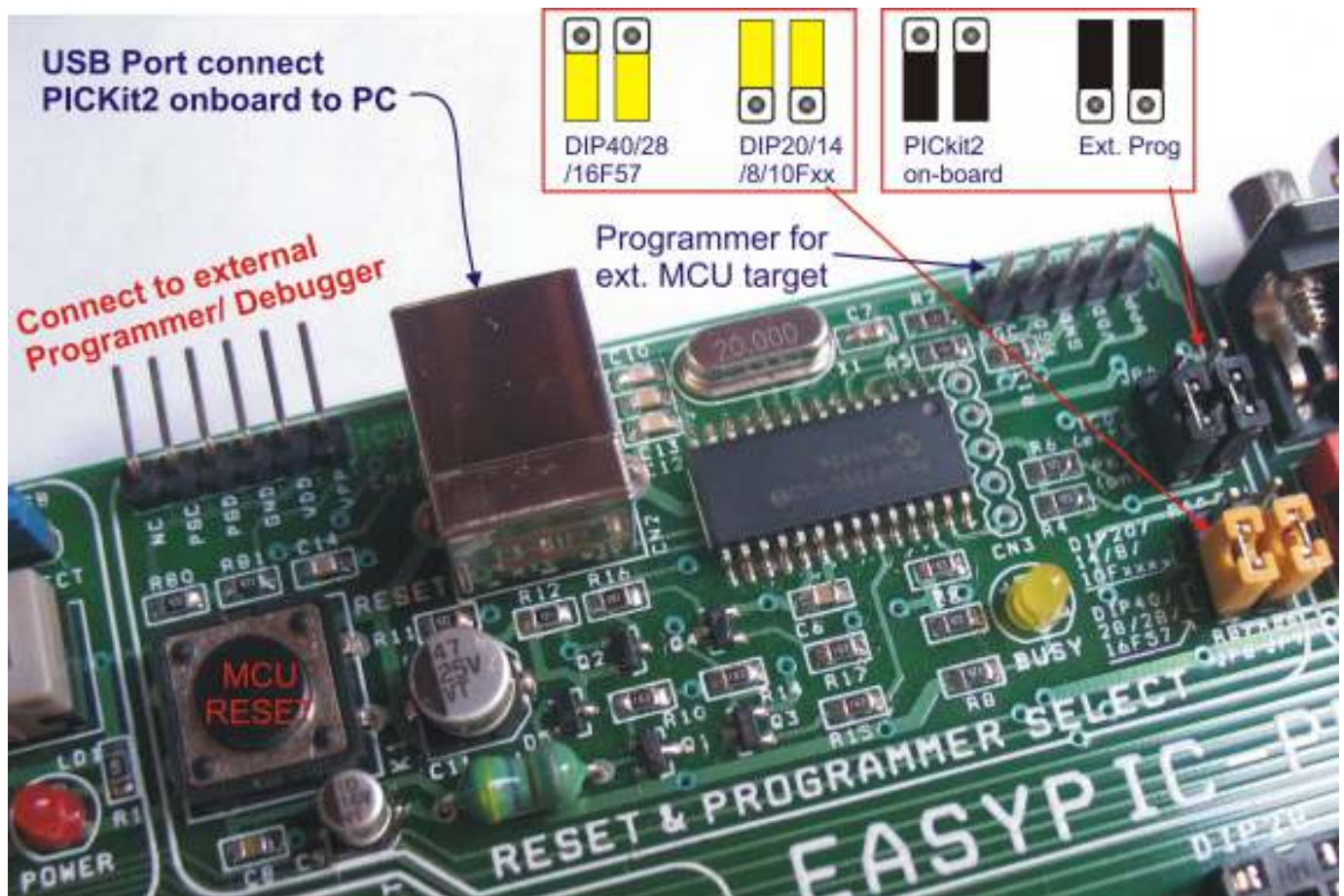


### Chương 3: Sử dụng On-line Programmer/Debugger

EASYPIC\_PRO đã tích hợp sẵn PICKit 2 on-board dùng để nạp và gỡ lỗi chương trình, tính năng và hiệu quả làm việc của nó bạn có thể tham khảo trên website của chúng tôi [www.tme.com.vn](http://www.tme.com.vn) hoặc của Microchip [www.microchip.com](http://www.microchip.com). Tuy nhiên EasyPIC\_PRO vẫn thiết kế thêm một header Ext.ICSP cho phép bạn kết nối với các Tools khác như PICKit 3 hoặc ICD2.

Chương này giới thiệu ngắn gọn các công cụ dùng để nạp và gỡ lỗi tương thích với EASYPIC\_PRO. Bạn có thể xem chi tiết các công cụ này trên website của chúng

#### 3.1 Cách sử dụng PICKit2 on-board programmer/ Debugger



Hình 3-1: Các chú thích PICKit 2 on-board Programmer trên EasyPIC\_PRO

##### 3.1.1 Sử dụng phần mềm PICKit 2 Programmer

1. Cài đặt phần mềm PICKit2 bằng cách chạy file setup.exe trong thư mục *PICKit 2 v2.61*.
2. Nối cáp USB với EasyPIC\_PRO và PC. PC sẽ tự nhận USB Driver
3. Set các Jumper JP6, JP7 về vị trí **PK2 on-board** như hình 3-1, Set các Jumper JP8, JP9 đến vị trí phù hợp với chip đích của bạn như hình 3-1

4. nhấn đúp vào biểu tượng **PICKit 2 v2.61** trên windows desktop, phần mềm sẽ chạy và tự nhận dạng phần cứng và chip đích, giao diện hiển thị như Hình 3-2
5. Vào menu File → Import HEX, để load file HEX cần nạp vào phần mềm
6. Nhấn Write để nạp chương trình vào chip, quá trình nạp diễn ra như mô tả trong hình 3-3. sau khi nạp xong chương trình sẽ tự động thực thi



Hình 3-2 Giao diện phần mềm PICKit 2



Hình 3.3: giao diện sau khi kết nối và quá trình nạp

**Lưu ý:** để nạp chương trình cho MCU bên ngoài, hãy tháo các Jumper J8,J9 để ngắt các tín hiệu ICSP đến PIC MCU trên EasyPIC\_PRO.

### 3.1.2 Sử dụng PICKit2 trong MPLAB IDE

Phần này sẽ được đề cập chi tiết trong chương 5 (**thực hành thực tế board phát triển EasyPIC\_PRO**)

## 3.2 Các Công cụ Programmer / debugger bên ngoài có thể sử dụng cho EASYPIC\_PRO

Đôi khi do nhu cầu cao hơn, hoặc do sở thích riêng, bạn có nhu cầu sử dụng các công cụ Programmer /Debugger khác? EasyPIC\_PRO có thiết kế một header ICSP, cho phép bạn thực hiện điều đó mà không ảnh hưởng gì đến các tính năng khác của EasyPIC\_PRO

Hãy tham khảo hình 3-1 để set cặp Jumper JP6 và JP7 sang vị trí **Ext.PROG** để sử dụng công cụ ngoài. Dưới đây là danh sách các công cụ có thể sử dụng như một Ext. Programmer / Debugger được đề nghị cho EasyPIC\_PRO. Bạn có thể sử dụng công cụ khác nếu chúng phù hợp với đường Ext. ICSP này. Thứ tự và ký hiệu các chân Ext. ICSP được chú thích chi tiết trên board. Hãy chú ý kết nối đúng với thiết bị ngoài.

Tên công cụ	Chức năng tóm tắt
<b>PICKit 3 debug express</b> 	<ul style="list-style-type: none"> <li>- Chức năng: Programmer, Debugger</li> <li>- USB (Full speed 12 Mbits/s interface to host PC)</li> <li>- Real-time execution</li> <li>- MPLAB IDE compatible (free copy included)</li> <li>- Built-in over-voltage/short circuit monitor</li> <li>- Firmware upgradeable from PC/web download</li> <li>- Totally enclosed</li> <li>- Supports low voltage to 2.0 volts (2.0v to 6.0v range)</li> <li>- Diagnostic LEDs (power, busy, error)</li> <li>- Read/write program and data memory of microcontroller</li> <li>- Erase of program memory space with verification</li> <li>- Freeze-peripherals at breakpoint</li> </ul>
<b>ICD2 debug express</b> 	<ul style="list-style-type: none"> <li>- Chức năng: Programmer/debugger</li> <li>- USB (Full Speed 2 M bits/s) interface to host PC</li> <li>- Real time background debugging</li> <li>- MPLAB IDE GUI (free copy included)</li> <li>- Built in over-voltage/short circuit monitor</li> <li>- Firmware upgradeable from PC</li> <li>- Totally enclosed</li> <li>- Supports low voltage to 2.0 volts. (2.0 to 6.0 range)</li> <li>- Diagnostic LEDs (Power, Busy, Error)</li> <li>- Reading/Writing memory space and EEDATA areas of target microcontroller</li> <li>- Programs configuration bits</li> <li>- Erase of program memory space with verification</li> <li>- Peripheral freeze-on-halt stops timers at breakpoints</li> </ul>

**PICKit 3 và ICD2 debug express** là các công cụ mạnh về tính năng debugger (Gỡ lỗi) cũng như tính năng Programmer (Nạp chương trình). Các công cụ này không kèm theo gói sản phẩm, bạn có thể mua thêm nếu có nhu cầu

Các dấu tam giác gần ngõ ICSP trên hộp kí hiệu vị trí chân số 1 VPP (MLCR) hoàn toàn trùng khớp với các chân trên Ext. ICSP của EasyPIC\_PRO. Thứ tự của chúng như sau:

**1-MCLR**

**2-VDD**

**3-GND**

**4-PGD**

**5-PGC**

**6-NC or LVP**

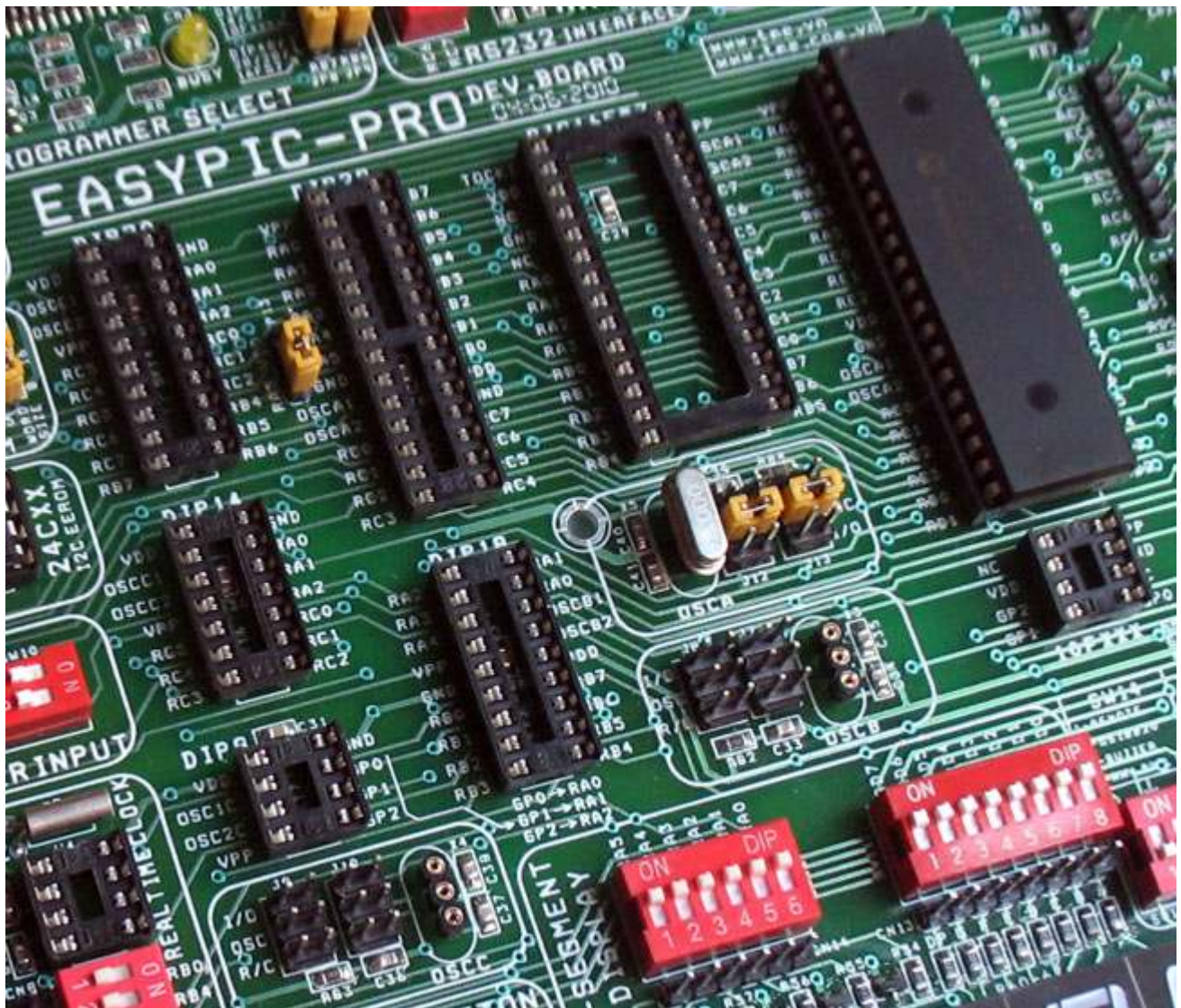


## Chương 4: Chức năng chi tiết các Module trên EASYPIC\_PRO

Chương này sẽ mô tả cụ thể hơn về chức năng các module trên Board phát triển EASYPIC\_PRO, bằng cách minh họa sơ đồ nguyên lý của các module và các điểm cần chú ý trong quá trình sử dụng. Trong đĩa CD-ROM được cung cấp cùng với sản phẩm, chúng tôi đã đưa vào các ví dụ cho từng chức năng của các module. Trong ví dụ cũng đã bao gồm các chú ý về cách bố trí phần cứng và phần mềm, trong đó hỗ trợ cho người dùng, đặc biệt là người mới bắt đầu. Theo board thử nghiệm là thiết kế kiểu module, cấu hình dự án của bạn có thể đa dạng với số lượng lớn và các Port đầu ra sẵn sàng cho kết nối với thiết bị bên ngoài. Các phần cứng và phần mềm sẽ thuộc về sở hữu người sử dụng. người dùng có thể thay đổi theo nhu cầu của mình và rút ra kết luận từ các trường hợp khác từ những ví dụ.

### 4.1 MCU Socket and System Clock selection

Module này là phần cốt lõi của EasyPIC\_PRO. Trình bày trên hình 4-1.



Hình 4-1 MCU socket and system clock selection



Module này bao gồm các thành phần sau:

1. 40-pin chip socket
2. 28-pin for PIC16F57 chip socket
3. 28-pin chip socket
4. 20-pin chip socket
5. 18-pin chip socket
6. 14-pin chip socket
7. 8-pin chip socket
8. 8-pin for PIC10FXXX chip socket
9. OSCA (lựa chọn clock sử dụng cho DIP40/28 and PIC16F57).
10. OSCB (lựa chọn clock sử dụng cho DIP18).
11. OSCC (lựa chọn clock sử dụng cho DIP20/14/8).

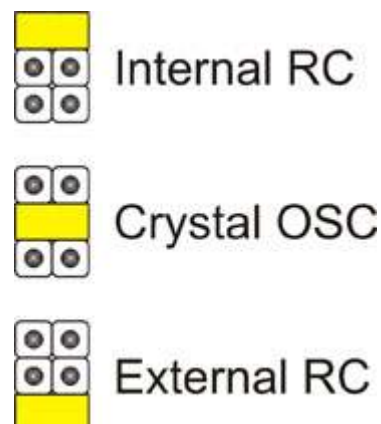
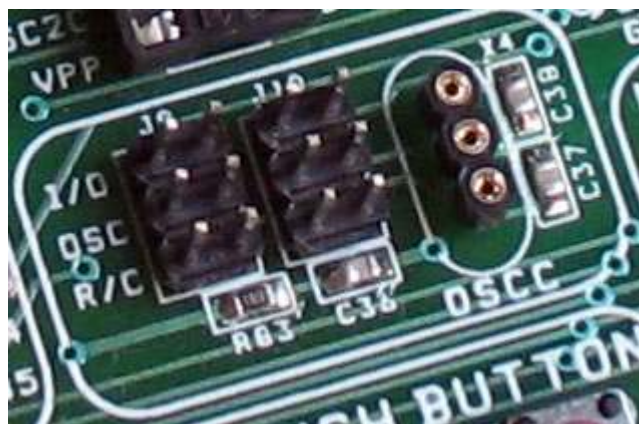
### Tài nguyên Chip-pin

Thông tin về chip-pin của mỗi chip, xin vui lòng tham khảo hướng dẫn tương ứng của nó. Ở đây chỉ liệt kê một số chân cụ thể được sử dụng trong board mạch này mà người sử dụng cần phải chú ý đến.

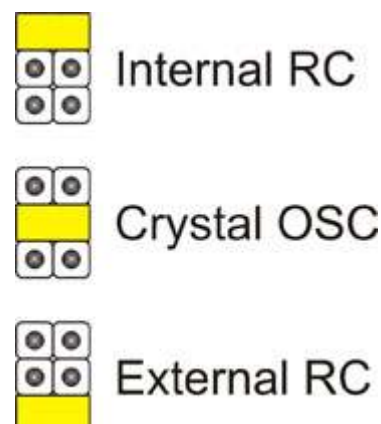
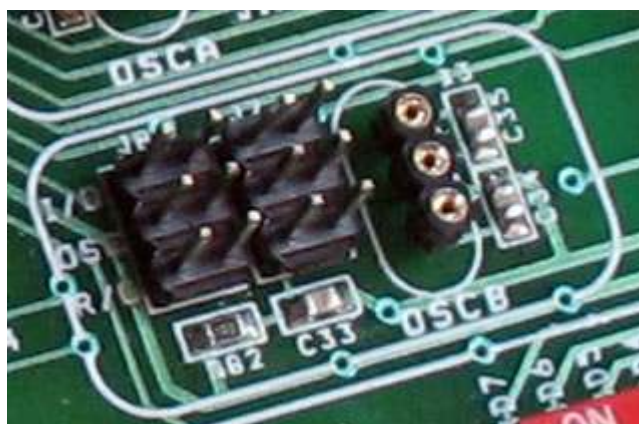
1. Chân số 8 của PIC10Fxx được sử dụng cho VPP và MCLR, không sử dụng cho I/O GP
2. Chân số 4 của DIP8 được sử dụng cho VPP và MCLR, không sử dụng cho I/O GP3
3. Chân số 4 của DIP14 được sử dụng cho VPP và MCLR, không sử dụng cho I/O RA3
4. Chân số 4 của DIP18 được sử dụng cho VPP và MCLR, không sử dụng cho I/O RA5
5. Chân số 4 của DIP20 được sử dụng cho VPP và MCLR, không sử dụng cho I/O RA3
6. TOCKI của PIC15F57 phải kết nối lên nguồn VDD
7. Chân số 7 của DIP28 có thể kết nối với RA5 hoặc VDD thông qua lựa chọn của jumper (Pin 7 của 18FXX31 là VDD, và Pin 7 của PIC 28-pin khác là RA5).
8. Các chân 2,3 của DIP8/14/20 không những sử dụng cho Clock input và output, mà còn được sử dụng làm I/O Port (Trong điều kiện chế độ dao động nội được chọn tại thời điểm lựa chọn clock)
9. GPIO của 8-pin Chip được kết nối đến các chip khác. Ví dụ GP0 được nối tới và dùng chung Port với RA0
10. Các chân 15, 16 của DIP18 không những sử dụng cho Clock input và output, mà còn được sử dụng làm I/O Port (Trong điều kiện chế độ dao động nội được chọn tại thời điểm lựa chọn clock)
11. Các chân 9, 10 của DIP28 và 13, 14 của DIP40 không những sử dụng cho Clock input và output, mà còn được sử dụng làm I/O Port (Trong điều kiện chế độ dao động nội được chọn tại thời điểm lựa chọn clock)
12. RA4 có một điện trở pull-up 10K (Có thể kiểm soát bằng S4 của SW14 – gần module LED 7-đoạn) dùng cho một số chip, điện trở kéo lên phải được kích hoạt để đưa RA4 lên mức cao.

**System Clock Selections:**

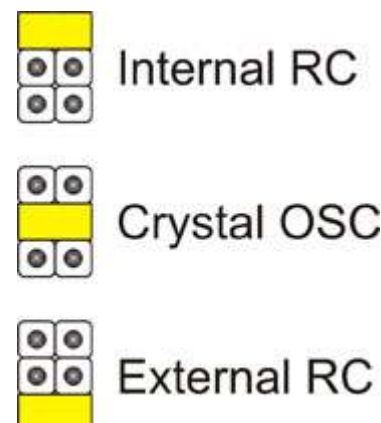
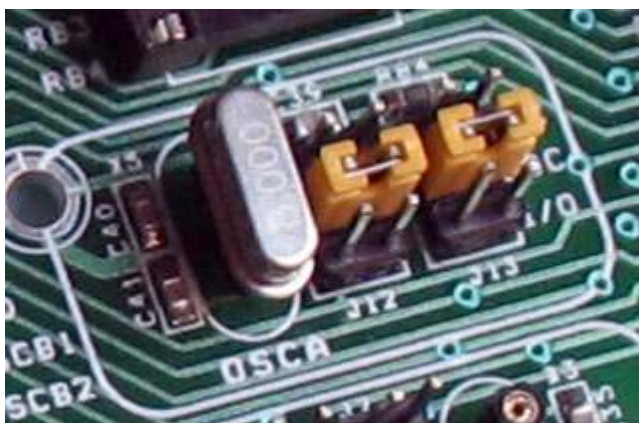
1. PIC10Fxx clock selection: PIC10Fxx chỉ sử dụng dao động nội (Internal RC oscillation)
2. DIP8/14/20 Clock selection: Trình bày như hình 4-2

**Hình 4-2** DIP8/14/20 Clock selection (OSC-C)

3. DIP18 Clock selection: Trình bày như hình 4-3

**Hình 4-3** DIP8/14/20 Clock selection (OSC-B)

4. DIP40/28/PIC16F57 clock selection Trình bày như hình 4-4

**Hình 4-4** DIP8/14/20 Clock selection (OSC-A)

## 4.2 POWER SUPPLY MODULE

Power module là nguồn cung cấp điện cho toàn bộ board phát triển EasyPIC\_PRO. Nó cung cấp đầu ra 5V DC cho tất cả các module trên board mạch. Module này thể hiện như hình 4-5



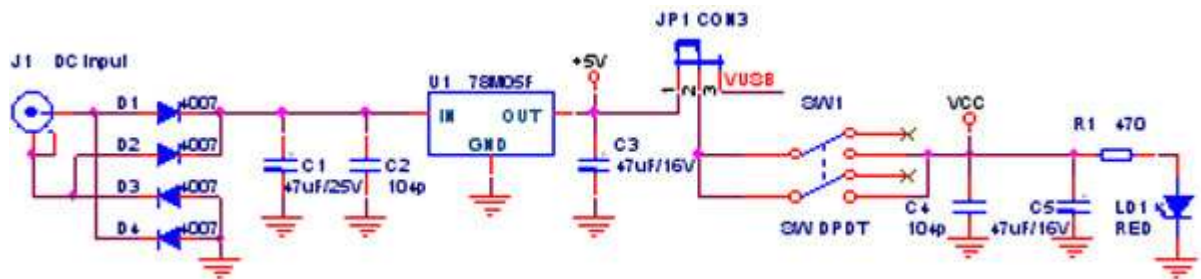
Module này bao gồm các thành phần sau:

- External power input: ngõ vào cấp nguồn cho Board
- Power indication: LED chỉ báo nguồn
- Power selection Jumper: cho phép lựa chọn nguồn ngoài hoặc nguồn 5V từ các cổng USB trên Board.
- Power Switch: cho phép ON/OFF để nối hoặc ngắt nguồn trên Board mạch

Nguồn cung cấp bên ngoài AC/DC 7~12V 200mA trở lên (Thích hợp nhất là 9V DC @500mA) không phân biệt cực tính.

**Hình 4-5:** Power Supply module

Sơ đồ nguyên lý của module này thể hiện như hình 4-6



**Hình 4-6:** Power Supply Schematic

## 4.3 128x64LCD and 16x02 LCD Modules

12864LCD và 1602LCD mô-đun được hiển thị trong hình 4-7. Các mô-đun bao gồm các thành phần sau:

1. Character LCD1602 và chiết áp điều chỉnh độ tương phản.
2. Graphic LCD12864 và chiết áp điều chỉnh độ tương phản



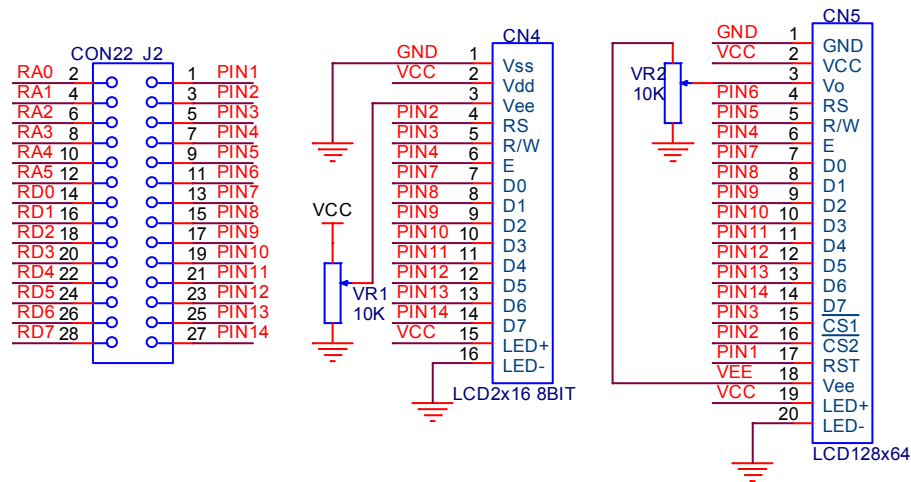
**Hình 4-7** 12864LCD and 1602LCD Modules



Mô tả các module như sau:

1. 12864 LCD and 1602 LCD cả 2 sử dụng PortA làm bit điều khiển và PortD là bit dữ liệu
2. Tất cả các chân của chúng nối tới các chân của MCU thông qua một loạt các Jumper mà không dùng đến các DIP Switch. Đề nghị không rút LCD ra khỏi socket khi đang có nguồn điện
3. Trong các thí nghiệm, điện trở kéo lên RA4 bắt buộc phải được kích hoạt bằng cách SW14\_4 ON
4. Datasheet của chúng kèm theo trong CD
5. Các ví dụ kèm theo trong đĩa CD khi mua sản phẩm

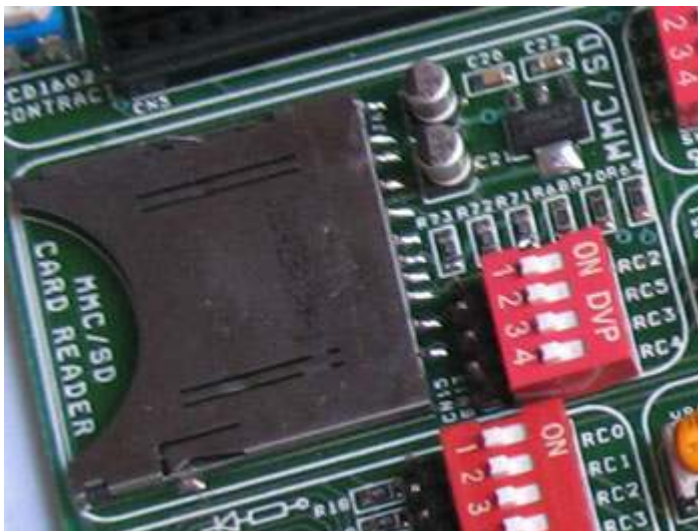
Sơ đồ nguyên lý của các module như hình 4-8



Hình 4-8 Schematics of 12864LCD and 1602LCD

#### 4.4 SD/MMC Card Read/Write Module

Module này chủ yếu cho thấy việc đọc, ghi MMC/SD. Thông qua việc nghiên cứu phần này chúng ta có thể hiểu thêm về các kiến thức liên quan đến việc đọc ghi SD/MMC card. trình bày như hình 4-9



Hình 4-9 MMC/SD card Module

Thẻ MMC/SD được sử dụng như một phương tiện lưu trữ cho các thiết bị cầm tay. Trong hình thức có thể dễ dàng gỡ ra để truy cập bằng máy tính Ví dụ như máy ảnh kỹ thuật số sẽ sử dụng một MMC card để lưu giữ các hình ảnh chụp được. nó có thể được chép sang máy vi tính bởi một đầu đọc nối qua cổng USB hoặc một số kết nối nối tiếp khác

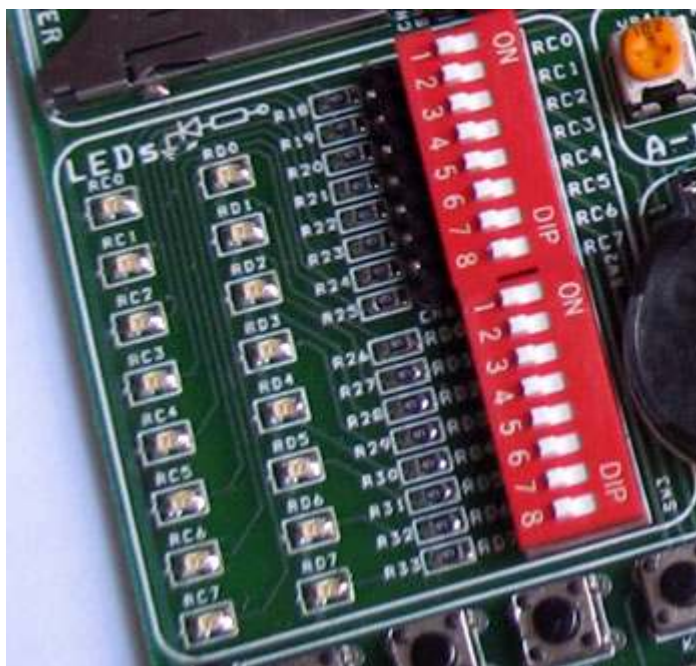
Giao tiếp với multi media card thông qua giao thức SPI



[illegible]

#### 4.5 LED đơn PORTB và PORTC

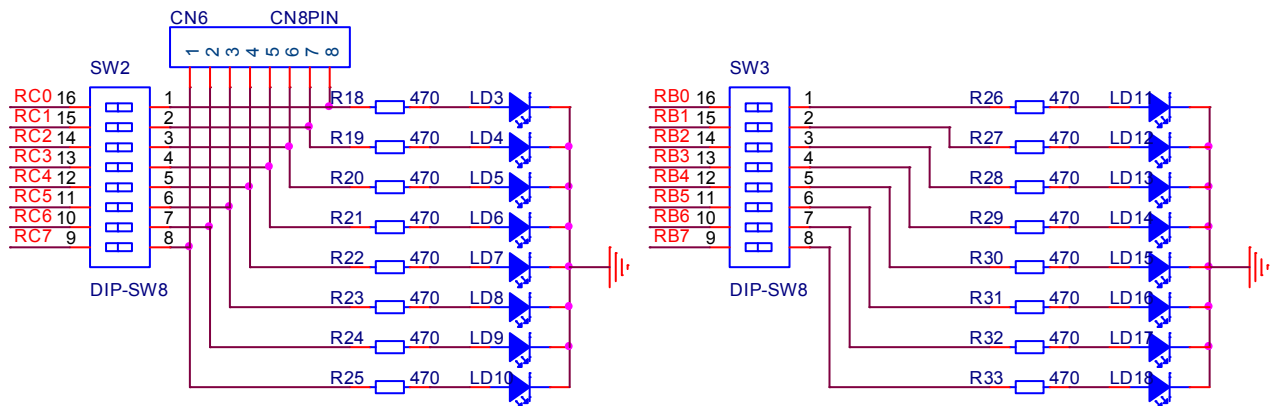
1. 8 LED đơn sử dụng cho PORT B
2. 8 LED đơn sử dụng cho PORT C



1. Tất cả 2 Port tương ứng với 8 bit ngõ ra
2. Tất cả 2 Port đều có Coding switch riêng biệt để kiểm soát chúng. Đề nghị hãy OFF tất cả các coding switch khi không sử dụng để tránh ảnh hưởng đến hoạt động bình thường của các module khác

- Thông qua Header giao tiếp ở PORTC, bạn có thể tiến hành thử nghiệm với các port khác (ví dụ PORTA, PORTD...). Tuy nhiên khi sử dụng điều này, hãy chắc chắn các coding switch đã được OFF
- Tất cả các LED có cực âm nối tới GND, nếu bạn muốn LED sáng, chỉ cần một điện áp mức cao tại đầu ra ở chân tương ứng của PORT.
- Các ví dụ trong CD bao gồm “Một LED sáng” và “LED sáng luân phiên” được kèm theo để tham khảo

Sơ đồ nguyên lý của module này trình bày như hình 4-12

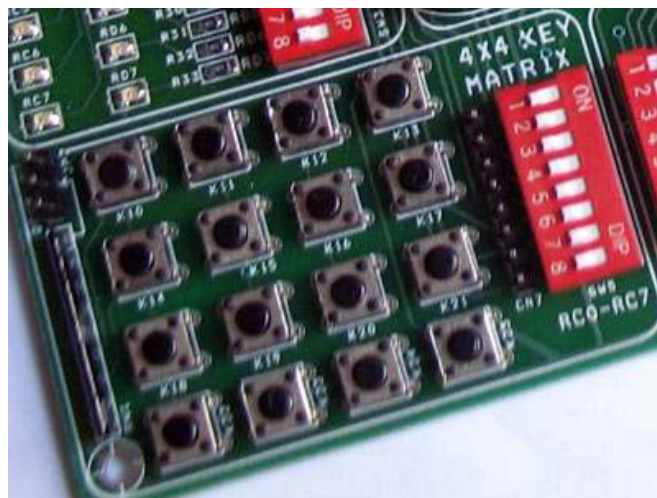


Hình 4-12 Sơ đồ nguyên lý modules PORTB và PORTC

#### 4.6 4x4 Matrix Keyboard

Module này trình bày như trên hình 4-13. bao gồm các thành phần sau:

- 16 phím nhấn
- Coding switch
- Header giao tiếp
- điện trở Pull-up / Pull-down

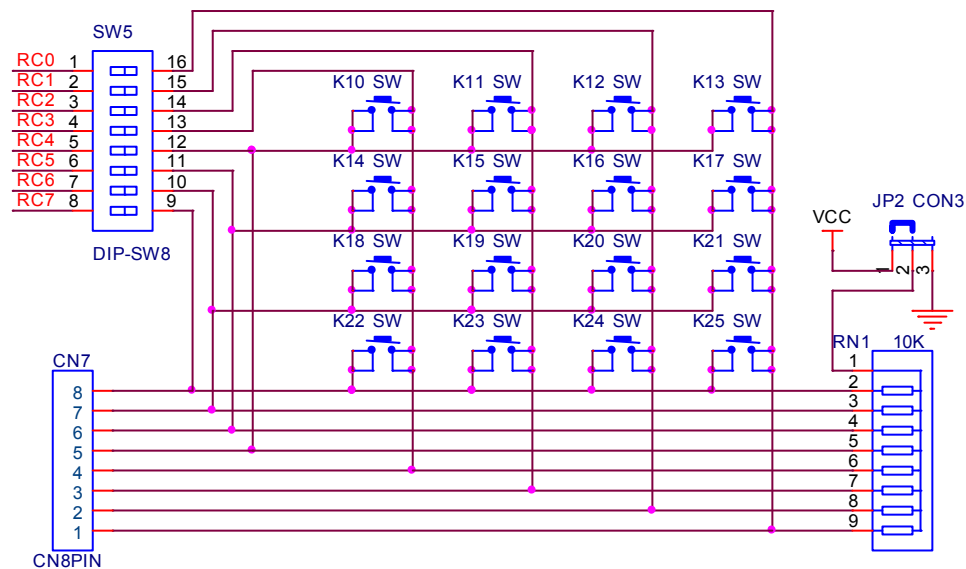


Hình 4-13: Key matrix 4x4 Module

Mô tả về module này như sau:

1. Các phím được nối tới 8 bit của PortC trong mảng 4x4
2. Module này sử dụng coding switch để kiểm soát. Khi không sử dụng, các bit tương ứng phải được OFF để không ảnh hưởng đến module khác
3. Thông qua Header giao tiếp bạn có thể thực hành với các Port khác (bạn hãy chắc chắn rằng coding switch đã được OFF)
4. Có một điện trở pull-up (kéo lên) hoặc pull-down (kéo xuống) lựa chọn bằng một jumper để đảm bảo sự ổn định của điện áp khi nhấn phím.
5. Các ví dụ về key matrix 4x4 kèm theo trong CD để tham khảo

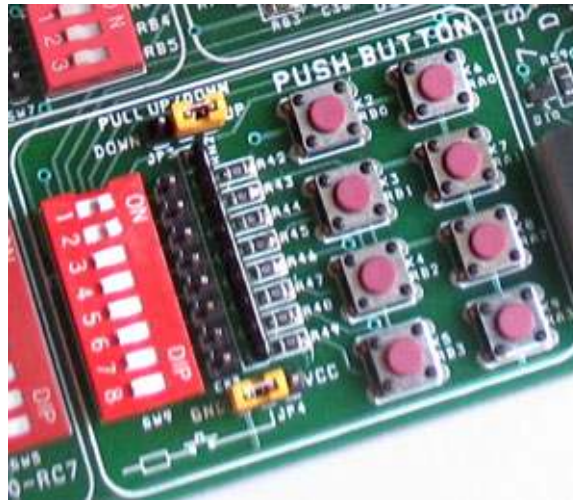
Sơ đồ nguyên lý trình bày như hình 4-14



Hình 4-14 sơ đồ nguyên lý module key matrix 4x4

#### 4.7 Push button switch module

Đây là module sử dụng các phím nhấn độc lập, nó được trình bày như hình 4-15



**Hình 4-15** Push button switch module

Module này bao gồm các thành phần sau:

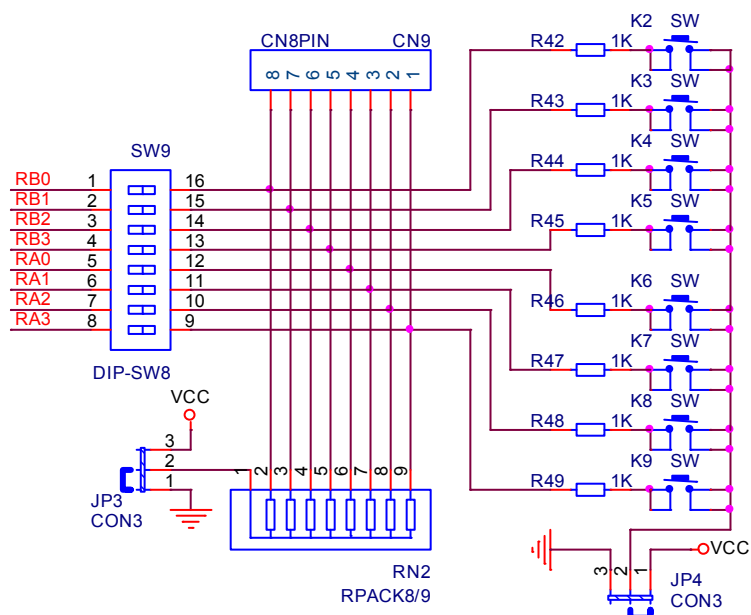
1. 4 phím trên PORTA A0~A3
2. 4 phím trên PORTB B0~B3
3. Điện trở 10Kx8 pull-up hoặc pull-down lựa chọn bằng Jumper
4. Jumper lựa chọn, cho phép phím nhấn đưa ra mức cao hoặc mức thấp
5. Coding switch: cho phép nối hoặc ngắt từng phím đến các PORT
6. Header giao tiếp cho phép giao tiếp bằng dây nối đến port khác

Mô tả của Module này như sau:

1. 8 phím được nối đến PORTA0~3, PORTB0~3
2. Khi phím được nhấn, mức đầu ra có thể chọn mức thấp hay mức cao (chọn jumper JP4)
3. 8 phím được điều khiển bởi coding switch để quyết định phím nào được nối đến chân của MCU. Khi không sử dụng, hãy chắc chắn coding switch được OFF để không ảnh hưởng đến sự làm việc bình thường của các module khác
4. Thông qua Hader giao tiếp, bạn có thể thử nghiệm với các Port khác của MCU (Khi đó coding switch phải ở trạng thái OFF)
5. Trong trường hợp các phím trên PortB0~3 được sử dụng, nếu điện trở Pull-up bên trong chip được kích hoạt, jumper pull-up có thể không cần dùng đến. Nhưng khi PORTA0~A3 được sử dụng, để ổn định mức điện áp, Bạn nhất thiết phải ON jumper này.
6. các ví dụ về sử dụng module này kèm theo trong đĩa CD



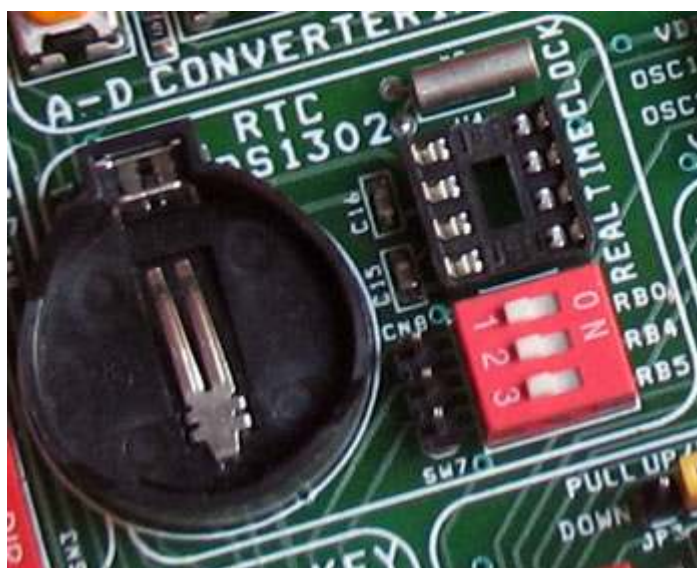
Sơ đồ nguyên lý của module này trình bày trên hình 4-16



Hình 4-16 Sơ đồ nguyên lý push button module

#### 4.8 DS1302 Real time clock module

Module này chủ yếu thử nghiệm giao tiếp với IC đồng hồ thời gian thực, module trình bày như hình 4-17



Hình 4-17 DS1302 RTC module

Module này bao gồm các thành phần chính như sau:

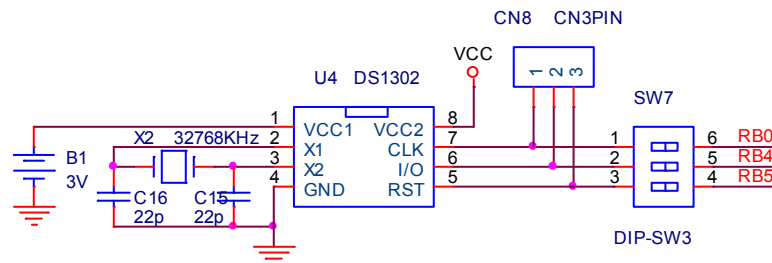
1. IC đồng hồ thời gian thực DS1302
2. PIN 3V
3. Thạch anh giao động 32.768 KHz
4. Coding switch cho phép nối hoặc ngắt module với MCU

- Header giao tiếp cho phép kết nối với các port khác của MCU

Mô tả module này như sau:

- Datasheet kèm theo trong đĩa CD
- Các Port giao tiếp CLK, I/O và Reset của DS1302 được nối tới RB0, RB4 và RB5 của MCU thông qua coding switch
- Khi module này được sử dụng, hãy chắc chắn coding switch ON, khi không sử dụng hãy OFF để khỏi ảnh hưởng đến hoạt động bình thường của các module khác
- Thông qua Hader giao tiếp, bạn có thể thử nghiệm với các Port khác của MCU (Khi đó coding switch phải ở trạng thái OFF)
- Ngay cả các tụ điện Star-up (C15, C16) của thạch anh 32768KHz cũng không phải hàn vào, và nó sẽ không ảnh hưởng đến star-up của thạch anh này
- các ví dụ về truy xuất DS1302 kèm theo trong đĩa CD-ROM để tham khảo

Sơ đồ nguyên lý của module này trình bày trên hình 4-18



Hình 4-18 Schematic of DS1302 RTC module

#### 4.9 A/D Converter Module

Module này mục đích chính là việc chuyển đổi tín hiệu tương tự (analog) sang tín hiệu số (Digital). Được trình bày như hình 4-19



Hình 4-19 D-D converter input

Module này bao gồm các thành phần chính sau:

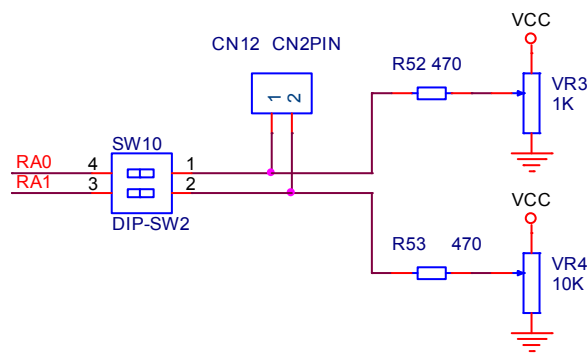
- 2 biến trở 1K

2. 1 coding switch 2 bit
3. header giao tiếp

Mô tả module này như sau:

1. 2 biến trở được nối tới các chân RA0 và RA1 tương ứng
2. khi chức năng A/D được sử dụng, hãy chắc chắn coding switch tương ứng được ON, khi không sử dụng hãy OFF coding switch để không ảnh hưởng đến module khác
3. Thông qua Header giao tiếp, bạn có thể thử nghiệm với các Port có chức năng AD khác của MCU (Khi đó coding switch phải ở trạng thái OFF)
4. Ví dụ mẫu về module này kèm theo trong đĩa CD

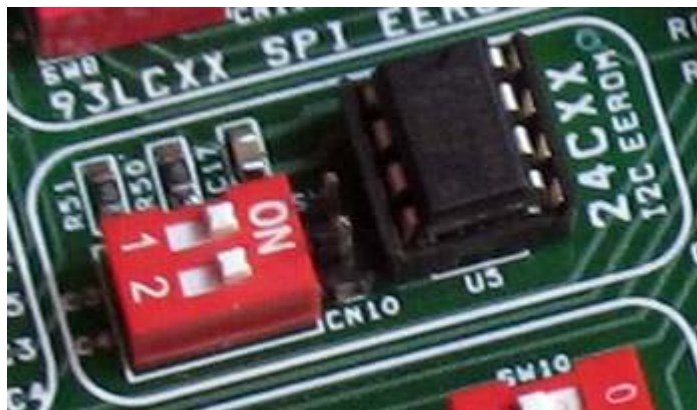
Sơ đồ nguyên lý của module này trình bày trên hình 4-20



**Hình 4-20** Schematic of A/D converter module

#### 4.10 Giao Thức I2C và EEPROM 24CXX Module

Module này chủ yếu là thí nghiệm tiếp cận thiết bị ngoại vi EEPROM 24Cxx và truy cập của MCU đến 24Cxx qua giao thức I2C. Xem hình 4-21



**Hình 4-21** 24Cxx EEPROM

Module này bao gồm các thành phần chính như sau:

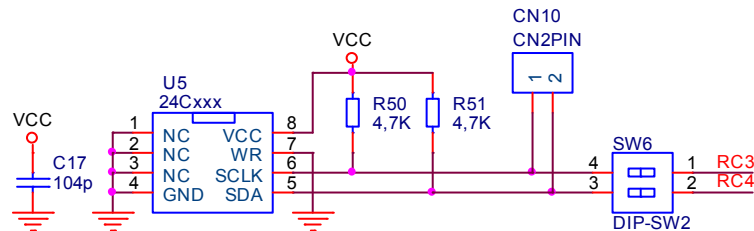
1. 24Cxx EEPROM
2. 2 bit Coding switch

### 3. 2 pin header giao tiếp

Mô tả module này như sau:

1. Datasheet của 24Cxx kèm theo trong đĩa CD-ROM
2. Port giao tiếp I2C SDA và SCK kết nối với Port giao tiếp I2C RC4 và RC3 của MCU thông qua coding switch
3. Khi Module này được sử dụng, hãy chắc chắn coding switch tương ứng được ON, khi không sử dụng hãy OFF coding switch để không ảnh hưởng đến module khác
4. Thông qua Header giao tiếp, bạn có thể thử nghiệm với các Port khác của MCU (Khi đó coding switch phải ở trạng thái OFF)
5. Ví dụ mẫu về EEPROM 24C01B kèm theo trong đĩa CD

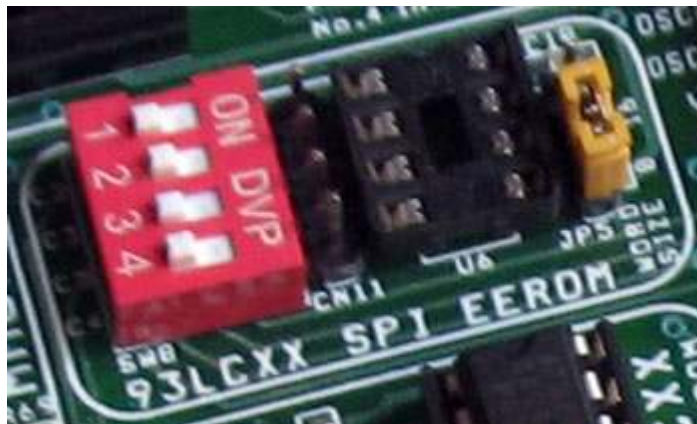
Sơ đồ nguyên lý của module này trình bày trên hình 4-22



**Figure 4-22** Schematic of access of MCU to 24CXX EEPROM

#### 4.11 Giao Thức SPI và 93LCXXX EEPROM Module

Module này chủ yếu là các thí nghiệm tiếp cận thiết bị ngoại vi EEPROM 93LCXXX, và truy cập của MCU đến EEPROM 93LCXXX qua giao thức SPI. Xem hình 4-23



**Hình 3-23** 93LCxx EEPROM

Module này bao gồm các thành phần chính như sau:

1. 93Cxx EEPROM
2. Coding switch
3. Jumper chọn ROM WORD SIZE

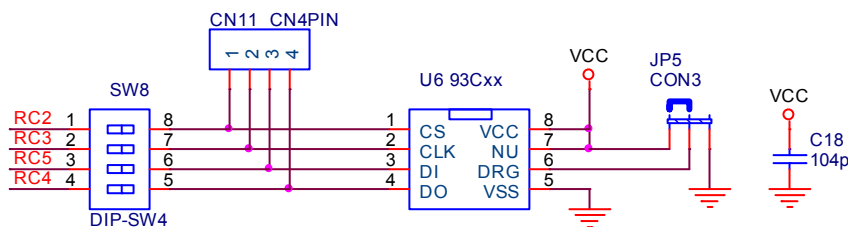


#### 4. Header giao tiếp

Mô tả module này như sau:

1. Datasheet của 93Cxx kèm theo trong đĩa CD-ROM
2. PORT giao tiếp SPI SDI, SDO và SCL được kết nối tới port giao tiếp SPI của MCU RC5,RC4 và RC3 thông qua coding switch, vì vậy bạn có thể sử dụng điều khiển bằng phần cứng
3. Khi Module này được sử dụng, hãy chắc chắn coding switch được ON, khi không sử dụng hãy OFF coding switch để không ảnh hưởng đến module khác
4. Thông qua Header giao tiếp, bạn có thể thử nghiệm với các Port khác của MCU (Khi đó coding switch phải ở trạng thái OFF)
5. Khi 93LCXXA (ROM SIZE là 8 bits) hoặc 93LCXXB (ROM SIZE là 16 bits) được sử dụng, Jumper WORD SIZE không có hiệu lực. khi 93CXX được sử dụng, jumper này sẽ quyết định việc chọn lựa WORD SIZE là 8 hoặc 16 bit
6. Ví dụ mẫu về EEPROM 93LC46 kèm theo trong đĩa CD

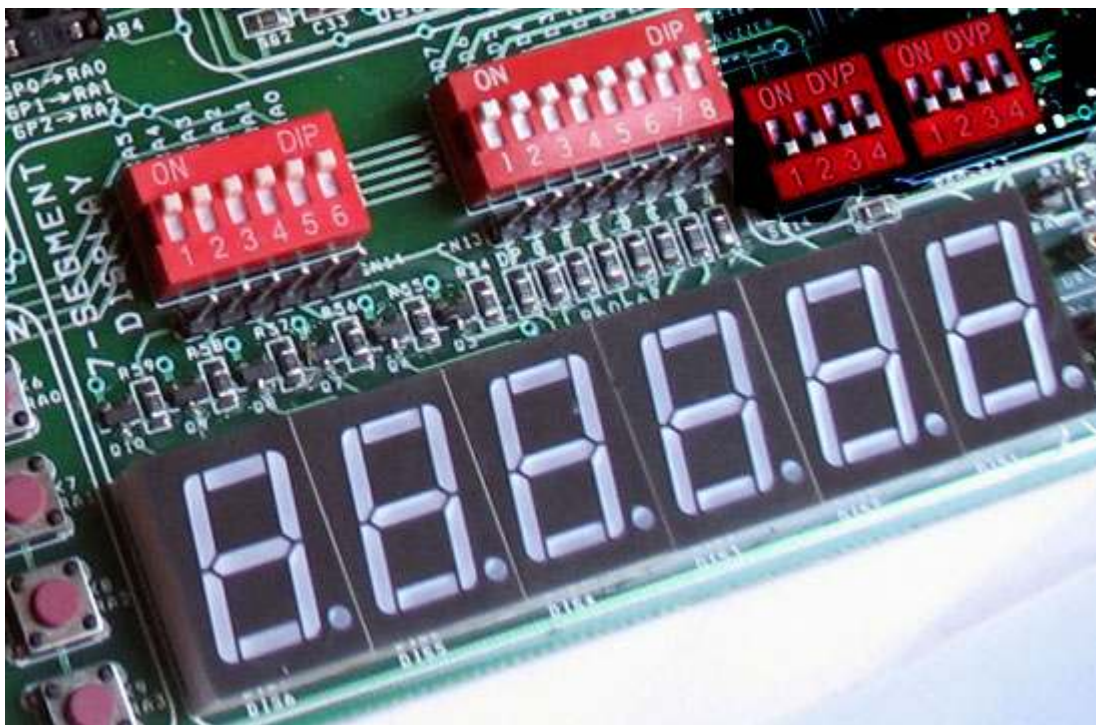
Sơ đồ nguyên lý của module này trình bày trên hình 4-24



**Hình 4-24** 93LCXXX EEPROM schematic

#### 4.12 7- Segment display Module

Module trình giới thiệu việc sử dụng một multi digital LED, như trong hình 4-25



**Hình 4-25** Six Digital LED Module

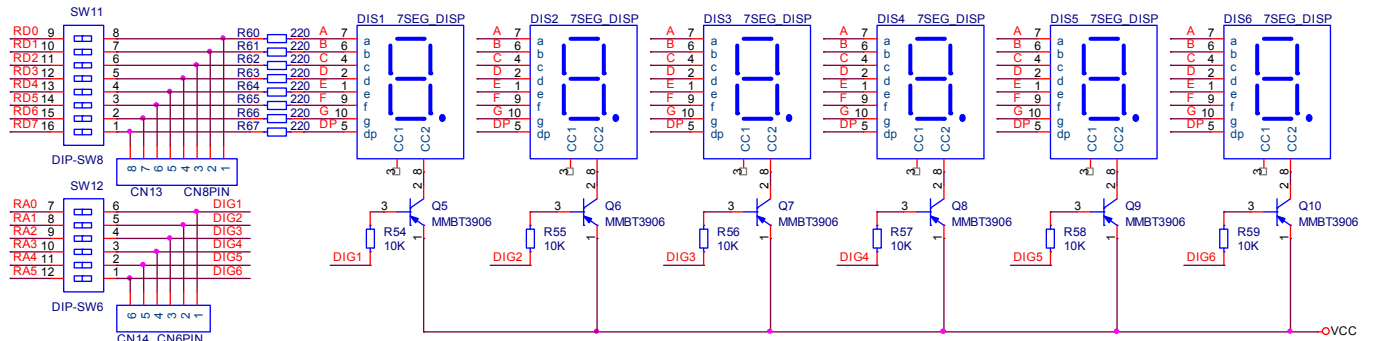
Module này bao gồm các thành phần chính như sau:

1. 6 LED số 7- đoạn anode chung
2. Coding Switch cho bit điều khiển (bit control) và phần kiểm soát (section control)
3. Mạch lái
4. Header giao tiếp

Mô tả module này như sau:

1. Section control được kết nối tới PORTD thông qua Coding Switch
2. Bit control của LED được nối tới PORTA thông qua Coding Switch, Khi module này không sử dụng, đề nghị coding switch phải ở vị trí OFF để khỏi ảnh hưởng đến hoạt động bình thường của các module khác
3. Thông qua header giao tiếp, bạn có thể thử nghiệm với các Port khác của MCU (Khi đó coding switch phải ở trạng thái OFF)
4. 6 led 7 đoạn thuộc loại Anode chung
5. Chương trình mẫu về module này có kèm theo trong đĩa CD

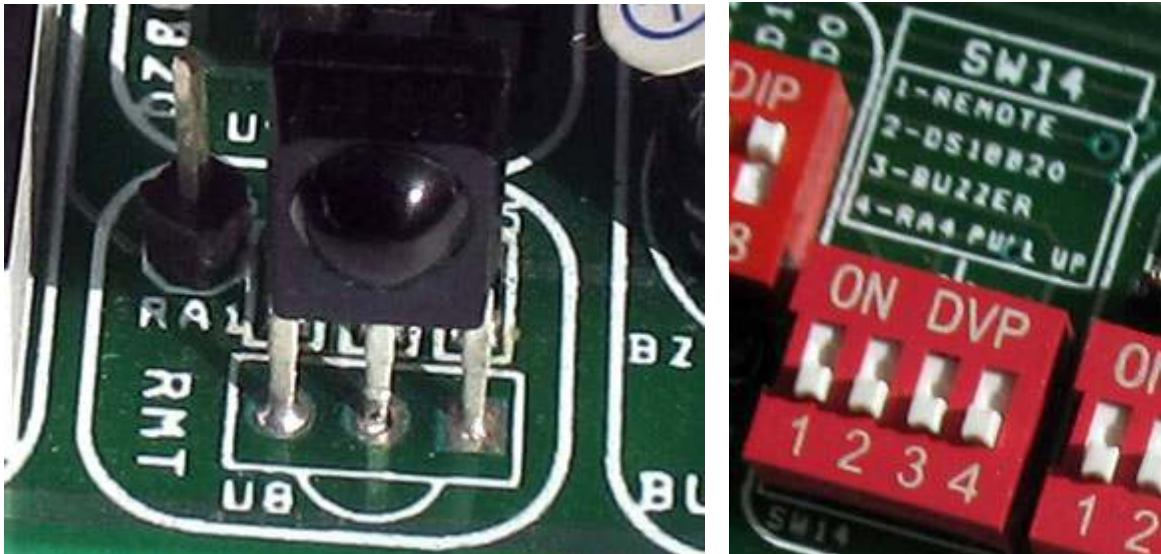
Sơ đồ nguyên lý trình bày trên Hình 4-26



Hình 4-26 Six digital LED schematic

#### 4.13 IR Remote Control Receiver Module

Module này chủ yếu là các thí nghiệm tiếp nhận và giải mã điều khiển từ xa hồng ngoại, như trong hình 4-27



Hình 4-27 Remote Control Receiver Module and coding switch

Module này bao gồm các thành phần chính như sau:

1. Bộ tiếp nhận điều khiển hồng ngoại
2. Coding switch
3. Header giao tiếp

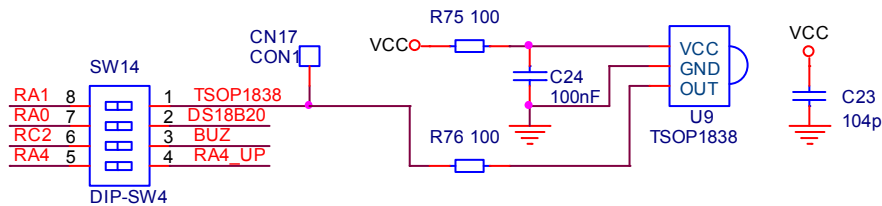
Mô tả module này như sau:

1. Datasheet của chip điều khiển từ xa bằng hồng ngoại 6121 kèm theo trong đĩa CD
2. Đầu ra của bộ thu hồng ngoại được nối tới RA1 thông qua coding switch SW14\_1
3. Module này sử dụng bit thứ 1 của coding switch SW14, các bit còn lại của coding switch chia sẻ với các module khác



4. Khi module này được sử dụng, hãy ON bit 1 của SW14, khi không sử dụng, để nghị OFF nó để khỏi ảnh hưởng đến các module khác
5. Bằng việc sử dụng Header giao tiếp, bạn có thể nối module này đến các port khác của MCU (Lúc đó bit 1 của SW14 phải là OFF)
6. Chương trình mẫu kèm theo trong đĩa CD

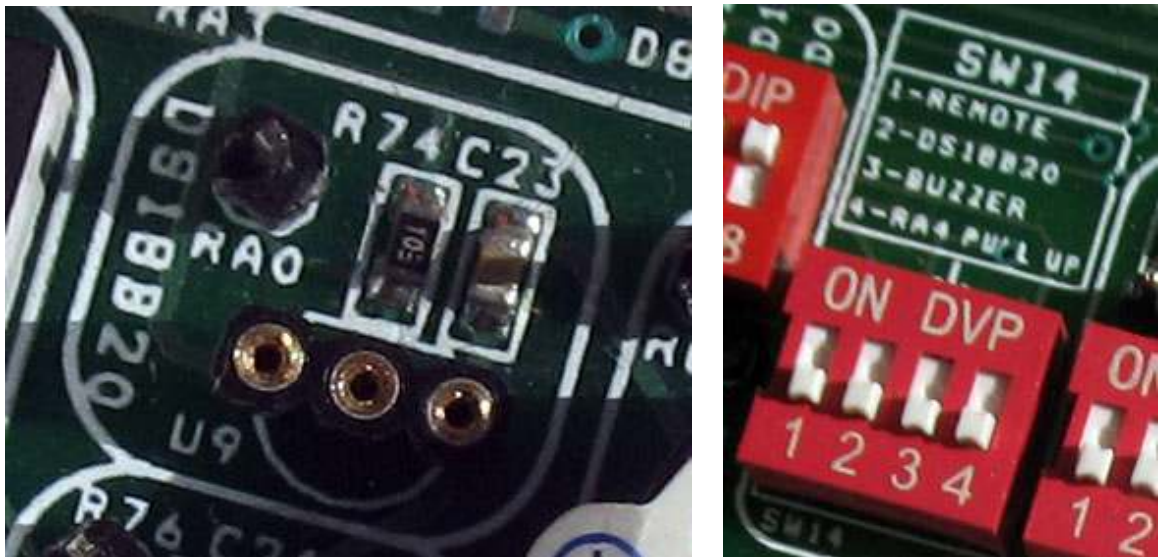
Sơ đồ nguyên lý trình bày trên Hình 4-28



**Hình 4-28** Remote Control Receiver and Decoder Module

#### 4.14 DS18B20 Thermometer Module

Module này chủ yếu sử dụng cảm biến nhiệt độ DS18B20, như hình 4-29



**Hình 4-29** DS18B20 module and coding switch

Module này bao gồm các thành phần chính như sau:

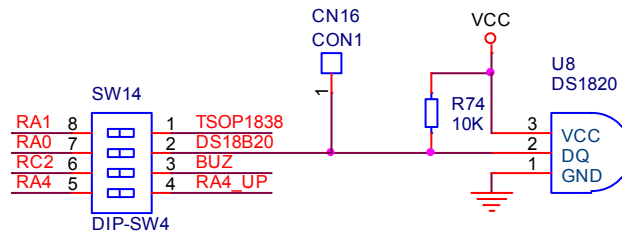
4. Cảm biến nhiệt độ DS18B20
5. Coding switch
6. Header giao tiếp

Mô tả module này như sau:

1. Datasheet của DS18B20 kèm theo trong đĩa CD
2. Ngõ ra của DS18B20 nối tới RA0 của MCU thông qua Coding Switch

- Module này sử dụng Bit thứ 2 của coding switch SW14, trong khi 3 bit còn lại của SW14 chia sẻ với các module khác.
- Khi module này được sử dụng, hãy ON bit 2 của SW14, khi không sử dụng, đề nghị OFF nó để khỏi ảnh hưởng đến các module khác
- Bằng việc sử dụng Header giao tiếp, bạn có thể nối module này đến các port khác của MCU (Lúc đó bit 2 của SW14 phải là OFF)
- Chương trình mẫu kèm theo trong đĩa CD

Sơ đồ nguyên lý trình bày trên Hình 4-30



Hình 4-30 Schematic of DS18B20 module

#### 4.15 BUZZER Module

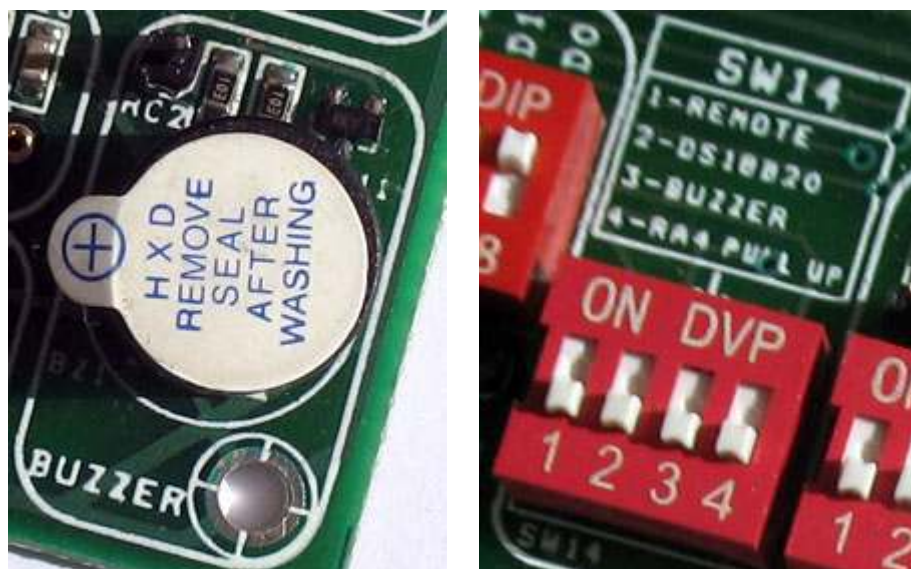
Module này chủ yếu cho thấy làm thế nào để điều khiển một Buzzer bằng MCU, như hình 4-31

Module này bao gồm các thành phần chính như sau:

- BUZZER
- Coding switch
- Header giao tiếp

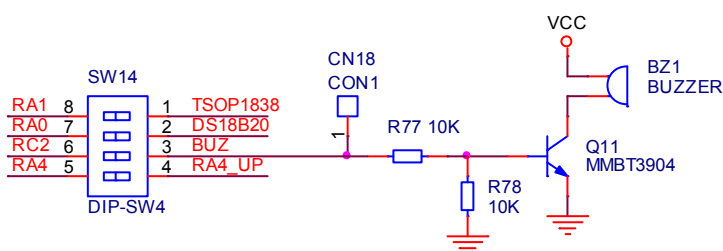
Mô tả module này như sau:

- Buzzer được nối tới RC2 của MCU thông qua Coding Switch
- Module này sử dụng Bit thứ 3 của coding switch SW14, trong khi 3 bit còn lại của SW14 chia sẻ với các module khác.
- Khi module này được sử dụng, hãy ON bit 3 của SW14, khi không sử dụng, đề nghị OFF nó để khỏi ảnh hưởng đến các module khác
- Bằng việc sử dụng Header giao tiếp, bạn có thể nối module này đến các port khác của MCU (Lúc đó bit 3 của SW14 phải là OFF)
- Chương trình mẫu kèm theo trong đĩa CD



Hình 4-31 Buzzer Module và coding switch

Sơ đồ nguyên lý trình bày trên Hình 4-32



Hình 4-32 Schematic of Buzzer Module

#### 4.16 Stepper Motor Module

Module này chủ yếu cho thấy hoạt động của động cơ bước (Step motor), thông qua phần này chúng ta nên tìm hiểu và bổ sung thêm các kiến thức có liên quan đến động cơ bước. Module này trình bày như hình 4-33

Module này bao gồm các thành phần chính như sau:

1. Header giao tiếp với step motor
2. Mạch lái
3. Coding switch



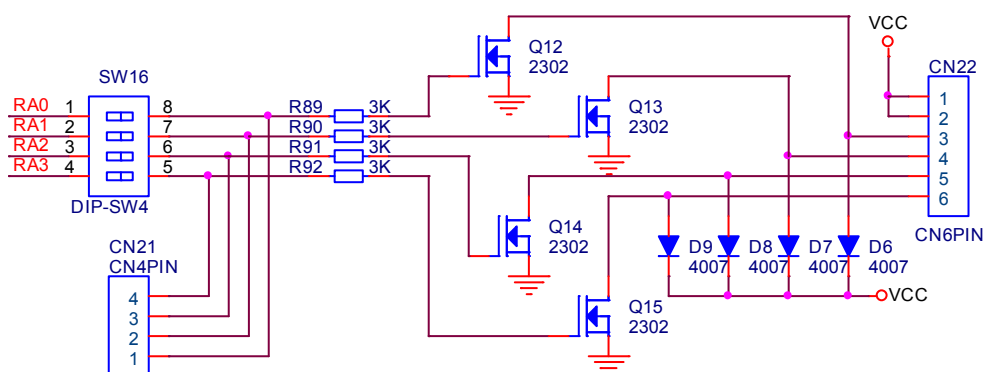


**Hình 4-33** Stepper motor module

Mô tả module này như sau:

1. Stepper Motor sử dụng trong board phát triển này là loại 35BY48S053 hoặc tương đương. Thông tin về loại động cơ bước này có kèm theo trong đĩa CD
2. Mạch lái đơn giản sử dụng FET để lái
3. dòng điện tiêu thụ cho động cơ bước là khá lớn, do vậy yêu cầu sử dụng nguồn ngoài và OFF coding switch khi không sử dụng để tránh tác động đến các module khác
4. Kèm theo trong đĩa CD là các ví dụ để tham khảo, nó cũng bao gồm điều khiển tốc độ, điều khiển chiều quay và điều khiển từ xa

Sơ đồ nguyên lý của module này trình bày trên Hình 4-32



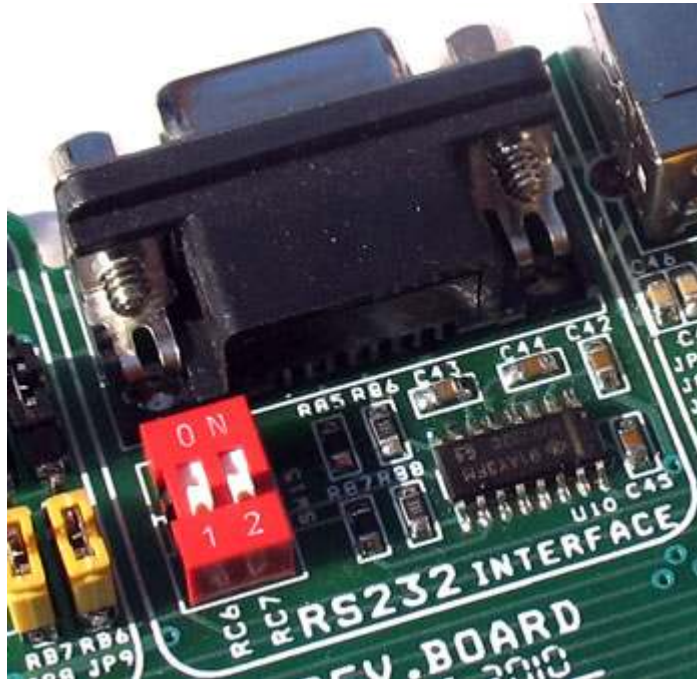
**Hình 4-32** Schematic of Stepper motor module

#### 4.17 RS232 Communication Module

Module này chủ yếu trình bày thí nghiệm cách sử dụng chức năng USART giao tiếp với các thiết bị ngoài (ví dụ như giao tiếp với PC). Module này như hình 4-33

Module này bao gồm các thành phần chính như sau:

1. IC chuyển mức RS232
2. Port giao tiếp RS232 (9-pin serial Port)
3. Coding switch
4. Header giao tiếp

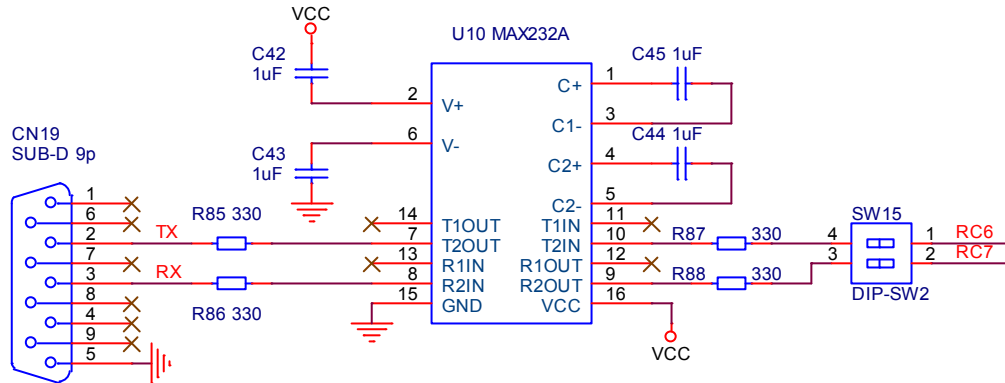


**Hình 4-33** RS232 Communication module

Mô tả module này như sau:

1. Serial module này kết nối với module giao tiếp USART của MCU RC6 và RC7 thông qua coding switch
2. Coding switch sẽ điều khiển việc kết nối, khi module này được sử dụng, hãy chắc chắn coding switch ở trạng thái ON, khi không sử dụng đề nghị coding switch phải là OFF để không ảnh hưởng đến sự làm việc bình thường của các module khác.
3. Bằng việc sử dụng Header giao tiếp, bạn có thể thử nghiệm việc giao tiếp của các chân khác của MCU với Serial Port (sử dụng phần mềm để mô phỏng)
4. Chương trình mẫu về module này có kèm theo trong đĩa CD

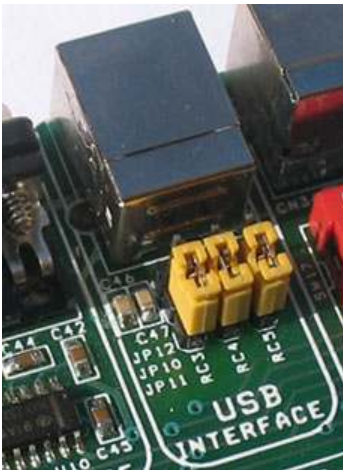
Sơ đồ nguyên lý của module này trình bày trên hình 4-34



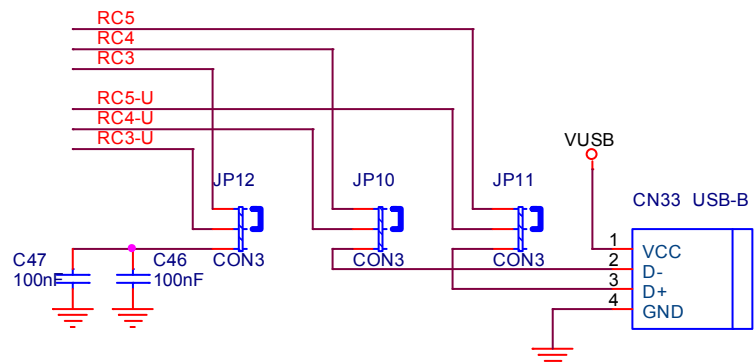
Hình 4-34 Schematic of RS232 Communication module

#### 4.18 USB Communication Module

EasyPIC\_PRO có một socket truyền thông USB. Nó được sử dụng cho những PIC microcontroller đặc biệt có hỗ trợ giao tiếp USB như 18F2450, 18F2550, 18F4550... Ngoài khả năng giao tiếp USB nó còn là bộ phận cấp nguồn 5V cho EasyPIC. Để cho phép kết nối giữa Microcontroller và socket USB communication, cần phải nối cả 3 jumpers JP10, JP11 và JP12 lên phía trên. Kết quả là Microcontroller có RC3, RC4 và RC5 được tách ra khỏi các phần tử khác trên board và nối với USB socket. Module này trình bày như hình 4-35, sơ đồ nguyên lý như hình 4-36



Hình 4-35 USB Module



Hình 4-36 Schematic of USB Module

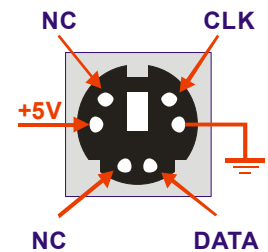
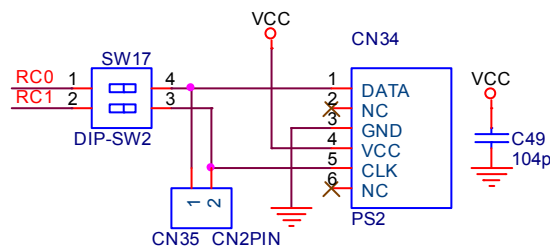
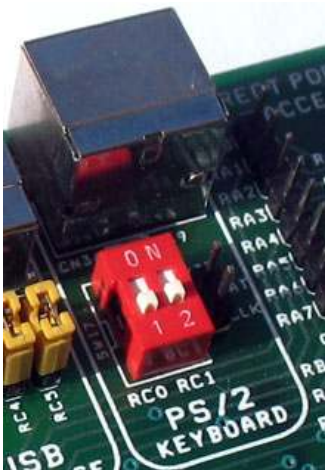
Mô tả module này như sau:

1. Khi không sử dụng các Jumper JP10, JP11 và JP13 sẽ nối các chân RC3, RC4 và RC5 của MCU đến các module khác (Như hình 4-35), Khi Module USB được sử dụng, các chân RC3, RC4 và RC5 của MCU sẽ ngắt khỏi các module khác và nối đến các port của USB
2. Nguồn cung cấp từ USB sẽ cung cấp nguồn cho board mạch khi jumper lựa chọn nguồn JP1 trong power module ở vị trí USB
3. Module này chỉ hỗ trợ các PIC18F có tích hợp USB như 18F2450, 18F2550, 18F4550...



#### 4.19 PS/2 Communication Module

Đầu nối PS/2 cho phép kết nối trực tiếp giữa EasyPIC với những thiết bị có sử dụng PS/2 communication, như PC, bàn phím (keyboard), hoặc con chuột (mouse). Ví dụ: Microcontroller được nối với bàn phím để đọc những phím được nhấn, hoặc nó có thể kết nối với PC để hoạt động như một bàn phím. Đường DATA và CLK được sử dụng cho việc truyền dữ liệu. Trong trường hợp này chúng được nối tới những pin tương ứng RC0 và RC1 của Microcontroller. Như hình 4-37



Hình 4-37 PS/2 module

Module này bao gồm các thành phần chính sau:

1. PS/2 socket
2. Coding switch
3. Header giao tiếp

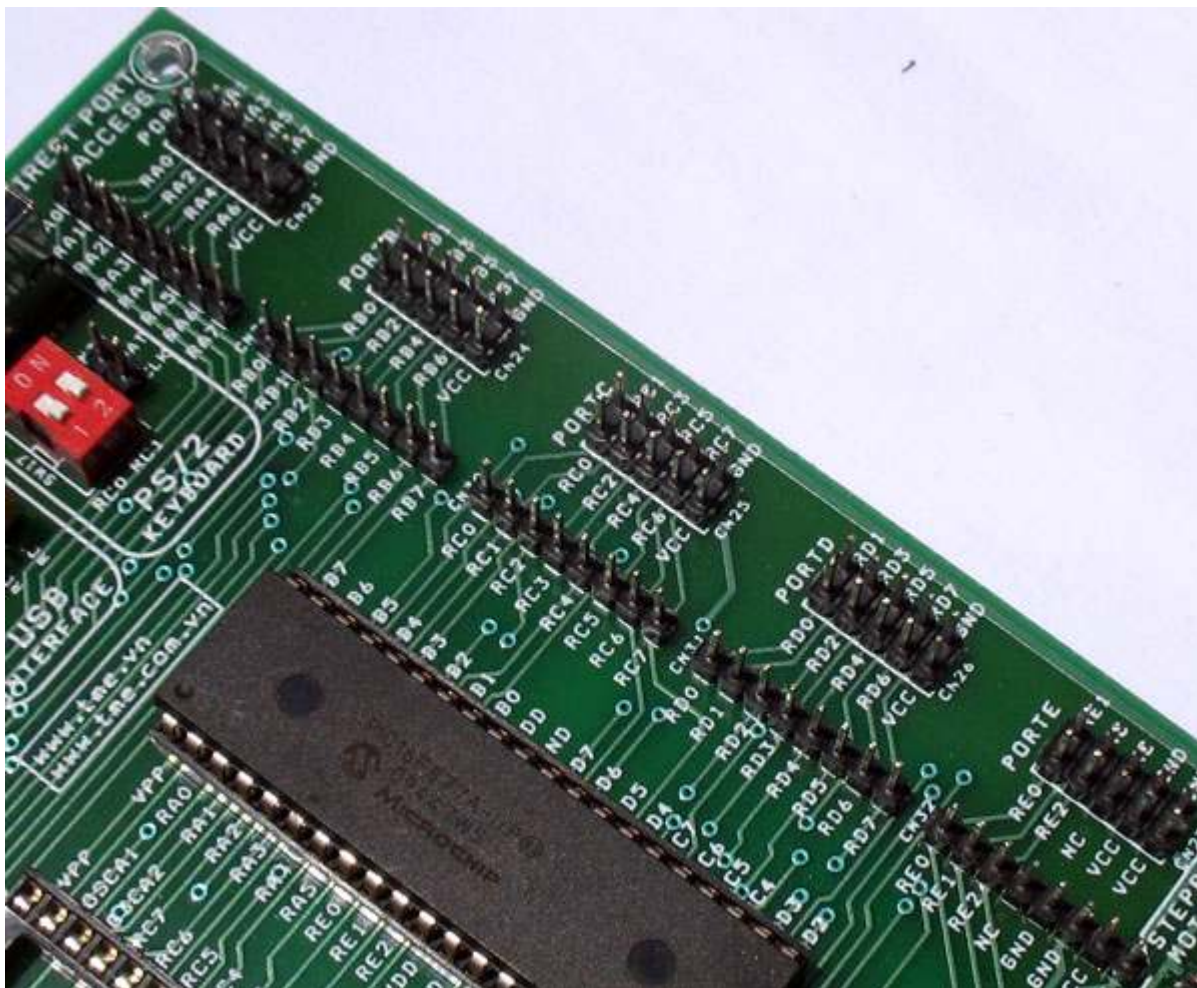
Mô tả module này như sau:

1. Các chân Data và CLK của PS/2 module nối đến các chân RC0 và RC1 của MCU thông qua coding switch
2. khi module này được sử dụng, hãy chắc chắn rằng coding switch ở vị trí ON, khi không sử dụng coding switch phải là OFF để không ảnh hưởng đến module khác
3. Bằng cách sử dụng Header giao tiếp, bạn có thể nối module này đến các port khác của MCU (Lúc đó coding switch phải là OFF)

#### 4.20 Direct PORT Access

Tất cả các pin I/O (input/output) của microcontroller có thể được truy nhập trực tiếp qua các đầu nối (Header) đặt bên mép phải của EasyPIC\_PRO cho mỗi PORTA, PORTB, PORTC, PORTD và PORTE. Để thuận lợi cho việc sử dụng, các PORT I/O trên EasyPIC\_PRO được thiết kế theo 2 kiểu Header, kiểu header 10-pin (5x2) cung cấp VDD, GND và 8 pin port thuận lợi cho việc kết nối với bên ngoài bằng flat

cable , kiểu 8x1 bao gồm 8 bit Port thích hợp cho việc tùy biến kết nối với các module trên board qua header giao tiếp của từng module bằng [Dupont wire](#). Trình bày như trên hình 4-38



Hình 4-38 Direct PORT Access

### Sử dụng Header giao tiếp theo nhu cầu riêng:

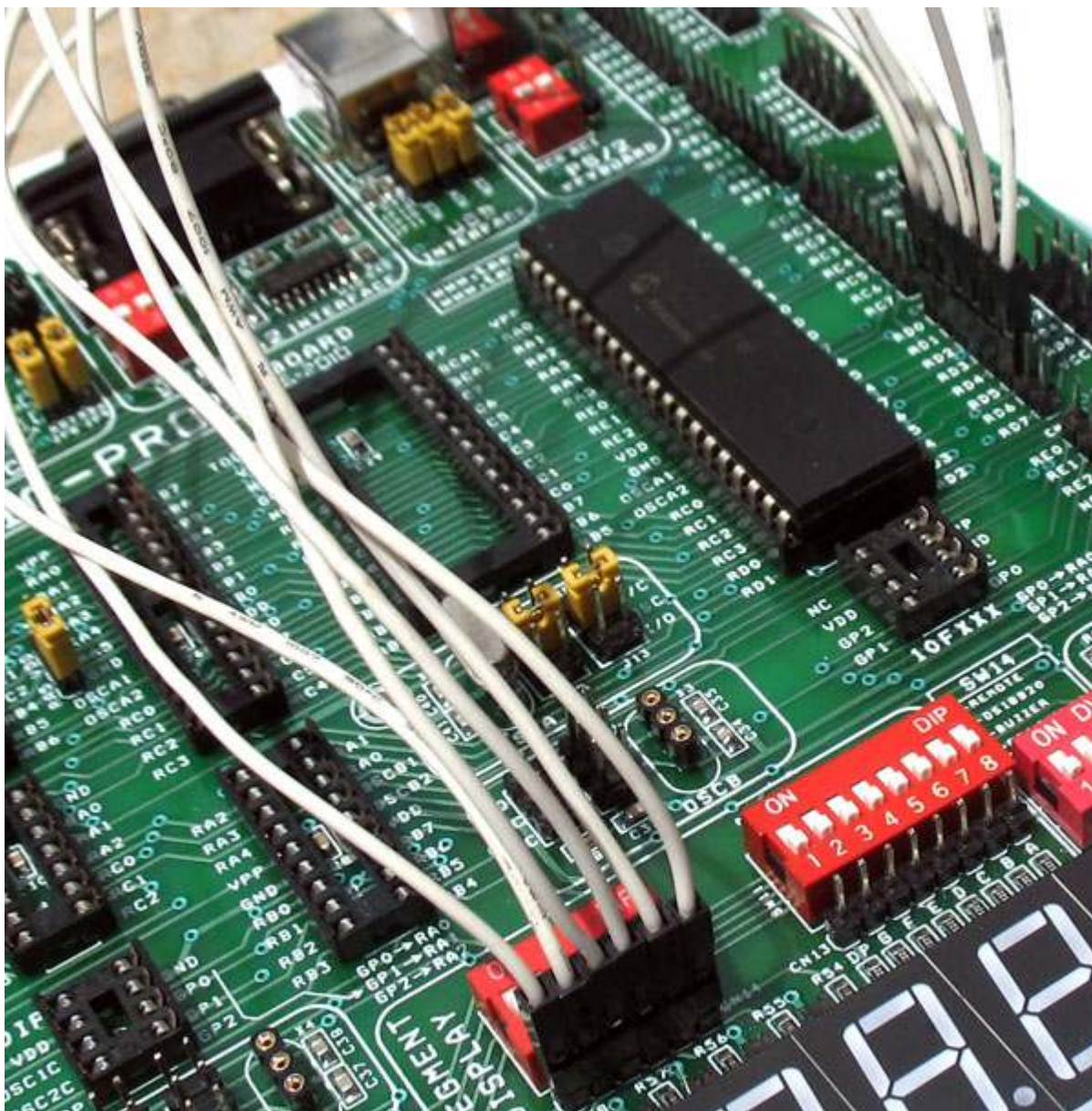
Hình 4-39 bên dưới trình bày một ví dụ về cách tùy biến và sử dụng các header giao tiếp để kết nối theo nhu cầu riêng trên EasyPIC\_PRO

Như đã trình bày trong mục 4.12 (Digital 7- Segment display Module), lấy ví dụ Module LED 7 đoạn có bit điều khiển mặc định là PORTA0~A5 thông qua Coding switch. Giả sử bạn muốn sử dụng PORT khác để làm bit điều khiển? chúng ta có thể hoàn toàn thực hiện được trên EasyPIC\_PRO bằng cách như sau:

1. Ngắt kết nối với PORTA bằng cách OFF coding switch (SW12)
2. Dùng dupont wire nối header giao tiếp CN14 của module 7-segment với Header PORTB của module Direct PORT Access. Kết quả như hình 4-39



Như vậy việc chuyển đổi kết nối đã hoàn tất, việc còn lại của bạn là sửa đổi chương trình cho phù hợp với nhu cầu của mình.

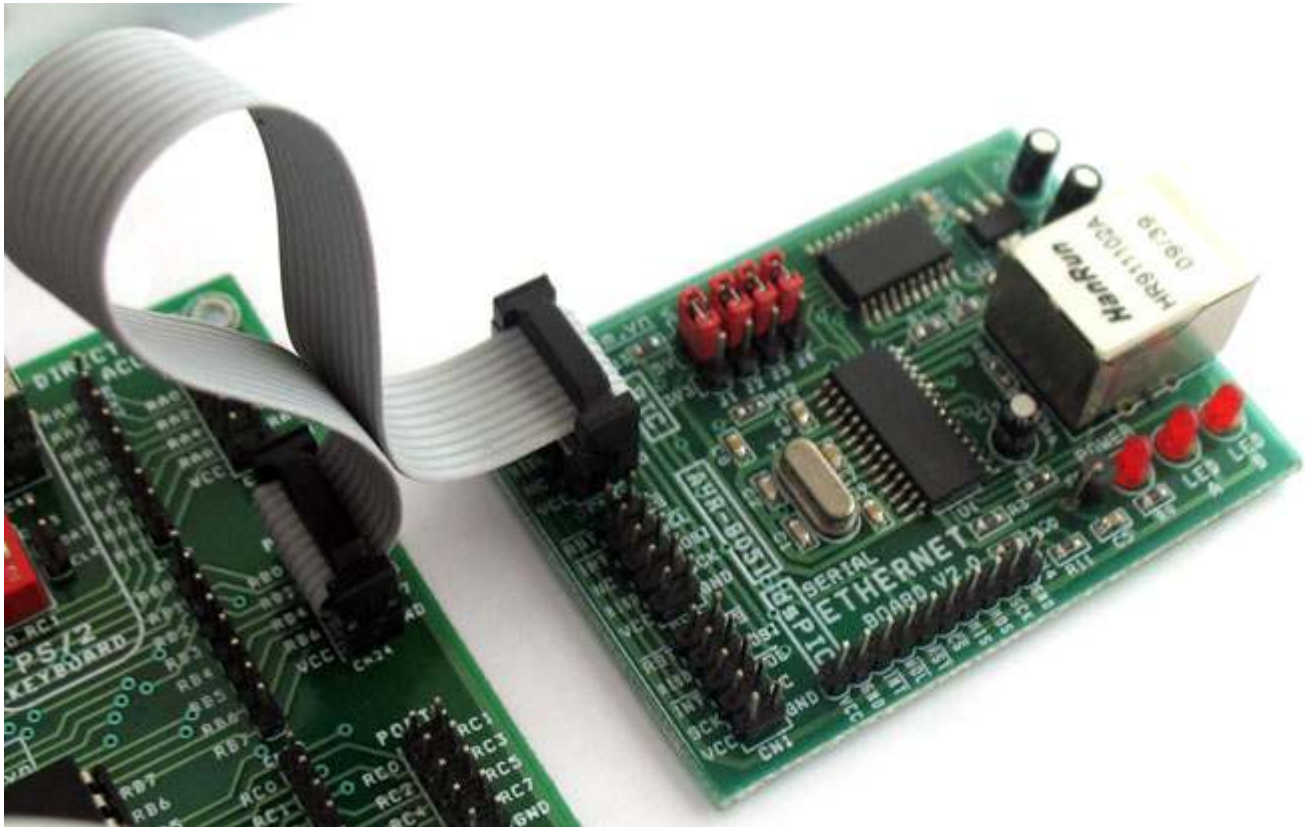


Hình 4-49 sử dụng Dupont wire và header giao tiếp

#### Sử dụng Header 5x2 giao tiếp với các module mở rộng bên ngoài:

Khả năng mở rộng ra bên ngoài của EasyPIC\_PRO là vô hạn, bạn có thể phát triển thêm các thành phần ngoại vi khác theo kiểu module và kết nối với EasyPIC\_PRO để tận dụng hết tính năng linh hoạt của nó, đáp ứng nhu cầu thử nghiệm nhanh và hiệu quả

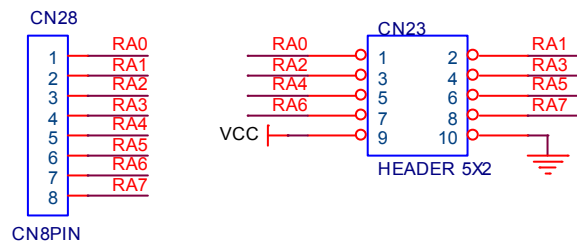
Hình 4-50 minh họa kết nối giữa EasyPIC\_PRO với module mở rộng [SPI Ethernet](#). Rất tiện lợi và gọn gàng



**Hình 4-50** minh họa kết nối giữa EasyPIC\_PRO với module mở rộng [SPI Ethernet](#)

Header 5x2 của Direct PORT Access đã có sẵn nguồn 5V đưa ra bên ngoài. Vì vậy với các module mở rộng sử dụng nguồn VDD = 5V, bạn không cần phải cấp nguồn riêng cho chúng, Cần lưu ý nguồn cung cấp từ các cổng USB thường có dòng nhỏ, do đó trong một vài trường hợp bạn cần phải dùng nguồn bên ngoài cung cấp cho EasyPIC\_PRO và các module mở rộng qua JACK DC trên module Power.

Hình 4-51 minh họa sơ đồ nguyên lý của các header PORTA, các PORT khác cũng tương tự



**Hình 4-51** I/O Header PORTA



## Chương 5: thực hành thực tế board phát triển EASYPIC\_PRO

### Mục đích thực hành thực tế:

Chương này sử dụng "module bàn phím ma trận 4x4" và "module 6 LED 7 đoạn" làm cơ sở, và đã giới thiệu về việc sử dụng Board phát triển EASYPIC\_PRO, bao gồm cả việc sử dụng MPLAB soạn thảo, biên dịch mã nguồn và dùng PICKIT 2 gỡ lỗi kết quả.

Bố trí phần cứng:

- 1) Coding SW SW5 cho Bàn phím ma trận 4x4 tất cả ON.
- 2) Các bit 5 và 6 của coding Switch cho LED 7 ở vị trí ON, trong khi các bit khác là off (chúng ta chỉ sử dụng LED 5 và 6 để hiển thị.)
- 3) Coding Switch điều khiển Segment của Led 7 đoạn ở vị trí ON.

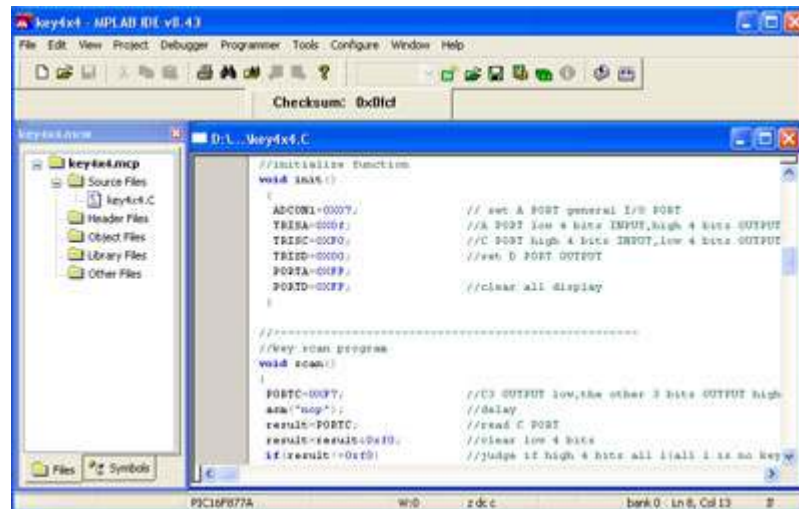
### Lập kế hoạch phần mềm:

- 1) Trong phần mềm này chúng ta bỏ qua hiện tượng nảy phím, khi phần mềm phát hiện mức điện áp thấp (mức 0), nó sẽ cho rằng có một phím được nhấn.
- 2) Trong phần mềm này, chúng ta không xem xét tình huống có nhiều phím được nhấn cùng lúc. Khi có một số phím cùng lúc được nhấn, chúng ta xem như là phím nhỏ nhất được nhấn. ví dụ, khi K10 và K11 cùng lúc được nhấn, chúng ta chỉ xem rằng K10 được nhấn.
- 3) Khi không có nút nào được nhấn, LED 7 đoạn sẽ hiển thị "FF", khi có một phím được nhấn, LED 7 đoạn sẽ hiển thị số của phím được nhấn đó. Ví dụ khi K10 được nhấn LED's 7 đoạn sẽ hiển thị "10", khi K25 được nhấn, LED's 7 đoạn hiển thị "25."


### Chuẩn bị và biên dịch mã nguồn:

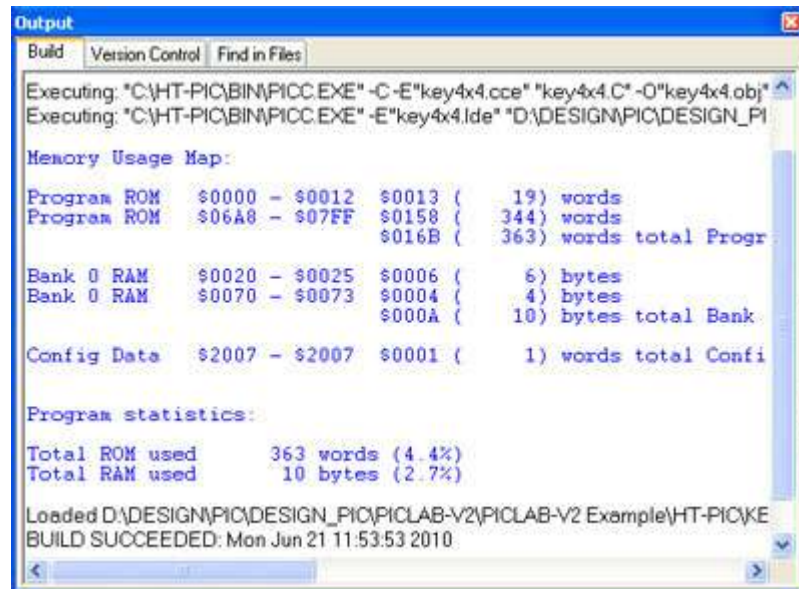
Nhấn đúp vào biểu tượng MPLAB trên Desktop để chạy môi trường lập trình MPLAB.

- 1) Theo phương pháp được giới thiệu trong 2.2.1, soạn thảo mã nguồn mới, và lưu nó như là "KEY4x4.C".
- 2) Theo phương pháp được giới thiệu trong 2.2.1, thiết lập một Project mới, và hoàn thành việc cài đặt như dưới đây: PIC16F877A chọn trong bước thứ hai như là chip đích; chọn "HI-TECH PICC Tool suite" là công cụ biên dịch trong bước thứ ba; chọn "KEY4x4" là tên của Project (các hậu tố có thể được bỏ qua), và thư mục để lưu project tương tự như của mã nguồn. Sau khi chỉnh sửa hoặc soạn thảo mã nguồn và thiết lập project, giao diện MPLAB sẽ được hiển thị như Hình 5-1.



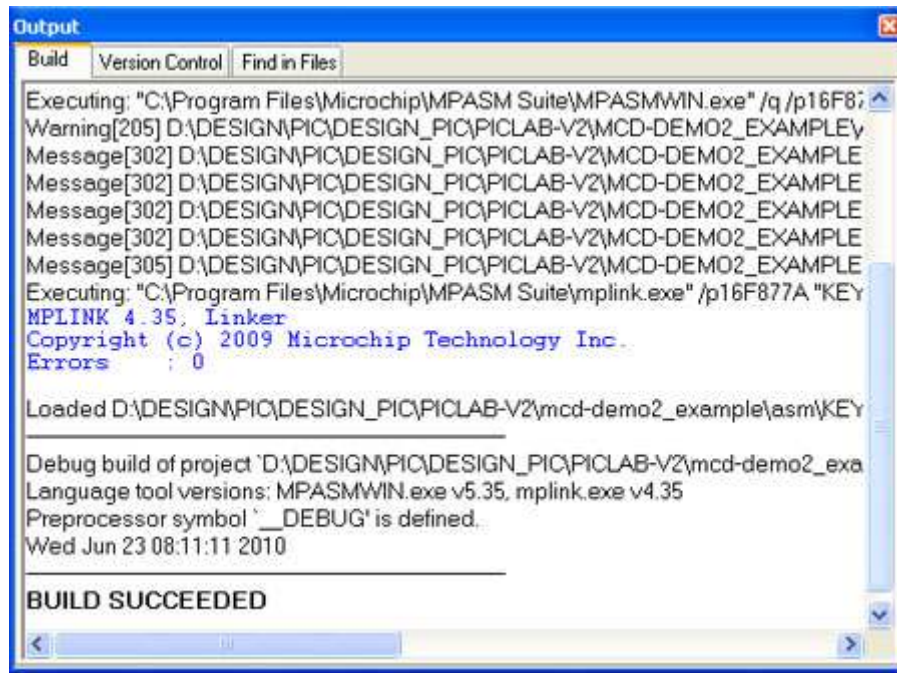
Hình 5-1 Giao diện MPLAB sau khi thiết lập project và soạn thảo mã nguồn

- 3) Thực hiện các menu lệnh Project → Build ALL hoặc biểu tượng  trên thanh công cụ để biên dịch mã nguồn. kết quả biên dịch sẽ được hiển thị như Hình 5-2. chúng ta có thể nhìn thấy trên hình là "BUILD SUCCEEDED" nghĩa là “biên dịch thành công”, và nó đã tạo ra một file KEY4x4.HEX trong thư mục của Project (chỉ biên dịch thành công thì mới tạo ra file HEX).



Hình 5-2 Kết quả sau khi biên dịch mã nguồn

Có thể trong một số trường hợp kết quả sẽ như sau:



Hình 5-3 Kết quả sau khi biên dịch mã nguồn

Có thể một vài trường hợp cho thấy như Hình 5-3 rằng sau khi biên dịch có một vài tin nhắn ví dụ như:

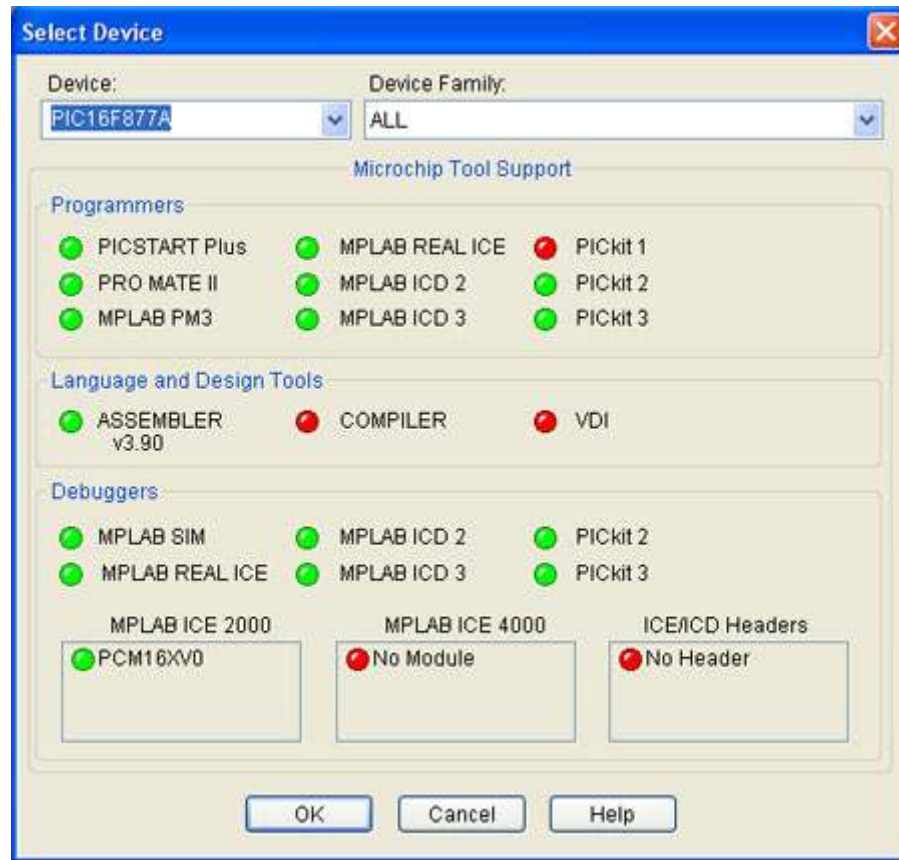
Message[302]D:\DESIGN\PIC\DESIGN\_PIC\EASYPIC\_PRO\ASM\KEY4X4\ KEY4X4.ASM 49 : Register in operand not in bank 0. Ensure that bank bits are correct.

Cho thấy registers sử dụng không thuộc BANK chính xác, hãy cẩn thận kiểm tra các chương trình để đảm bảo rằng tất cả các registers phải trong BANK (ngay cả khi hoàn toàn chính xác, thông tin sẽ vẫn xuất hiện, nhưng nó không ảnh hưởng đến việc thực hiện các kết quả).

Nếu có ERROR[num] hoặc WARNING[num] trong kết quả biên dịch (num là số mã của error hoặc warning), Bạn có thể cho con trỏ nhảy đến vị trí lỗi bằng cách nhấn đúp vào tin nhắn này và sửa chữa mã nguồn và lặp lại thao tác trên để tiếp tục sửa chữa các lỗi khác cho đến khi không còn lỗi.

### Sử dụng PICKit 2 để debug mã nguồn

Tương tự như PICKit 2, bạn có thể sử dụng ICD2 hoặc PICKit 3 cho việc debug. Để biết các công cụ nào hỗ trợ debug cho chip của bạn, bạn có thể kiểm tra trong MPLAB bằng cách nhấn vào menu lệnh Configure → Select device...và chọn chip của bạn trong ô Device. Bạn sẽ có kết quả như Hình 5-4



**Hình 5-4** Minh họa các công cụ hỗ trợ Programmer và Debugger cho PIC16F877A

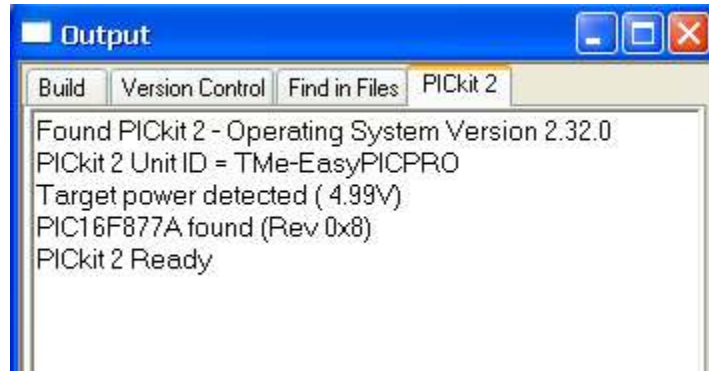
Trên hình 5-4, chú ý các nút tròn trên hình ta sẽ biết được các tools nào có hỗ trợ cho PIC MCU của mình bằng cách phân biệt các màu như sau

- Màu xanh lá (Green): có hỗ trợ
- Màu đỏ (Red) : không hỗ trợ
- Màu vàng (Yellow) : chưa hỗ trợ ( nghĩa là sẽ hỗ trợ trong tương lai)

Quay lại vấn đề, sử dụng PICKIT 2 để Debug cho mã nguồn, ICD2 và PICKit 3 cũng theo phương pháp tương tự. ở đây dùng PICKIT 2 làm điển hình

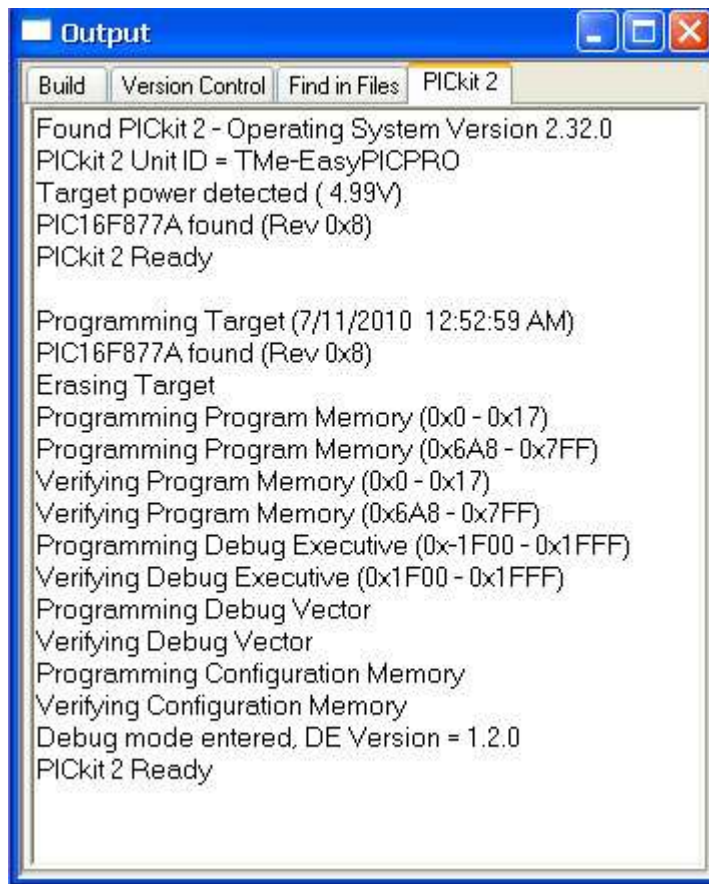
- 1) Theo phương pháp thể hiện trong Hình 3-1, Kết nối PICKIT 2 với PC và EASYPIC\_PRO development board.
- 2) Thực hiện menu lệnh Debugger → Select Tool để chọn PICKIT 2 làm công cụ debugging. MPLAB sẽ tự động kết nối với PICKit 2 và nhận dạng chip đích. Kết quả kết nối sẽ hiển thị trong cửa sổ output của MPLAB, Nó thể hiện như Hình 5-5.
- 3) Nếu không nhìn thấy kết quả kết nối như hình 5-5, hãy thực hiện menu Debugger → Connect hoặc nhấn icon trên tool bar để kết nối PICKIT 2 và chip đích trên EASYPIC\_PRO development board.





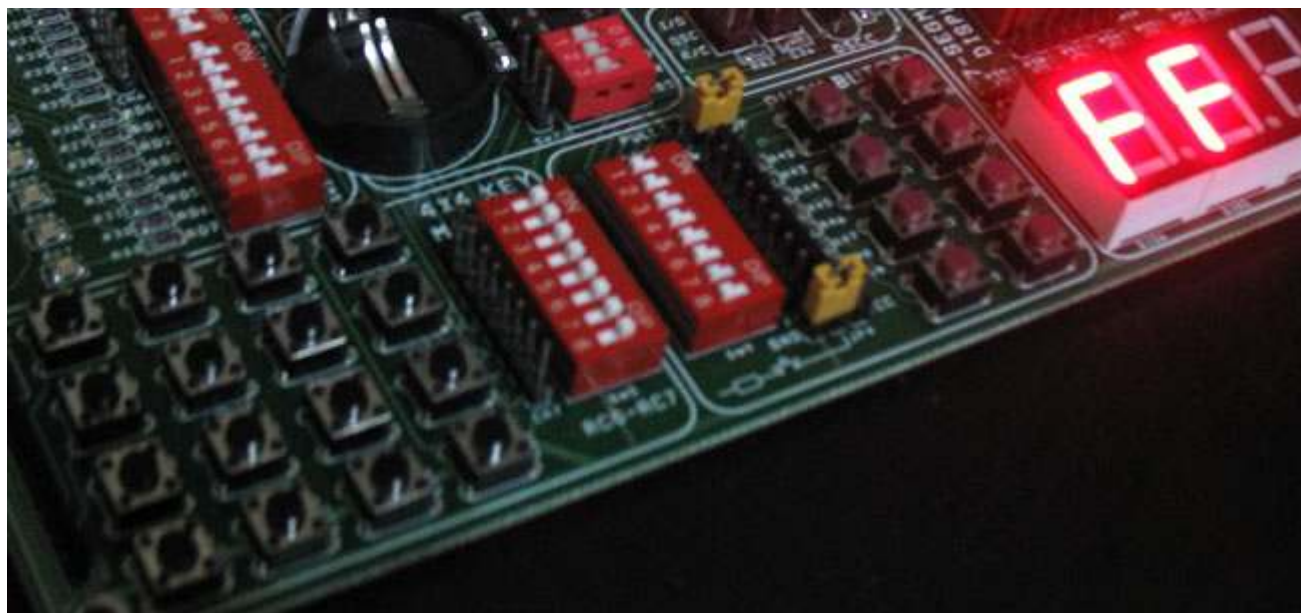
**Hình 5-5** Kết quả sau khi chọn PICKit 2 làm công cụ debug

- 4) Thực hiện menu Debugger → Program or nhấn icon trên tool bar để nạp /ghi objective codes vào MCU chip đích (Lưu ý: bây giờ MCU không thể chạy ở chế độ offline mà phải được chạy ở chế độ gỡ lỗi - debugging mode), sau khi nạp nó sẽ hiển thị thông tin quá trình như trên Hình 5-6.

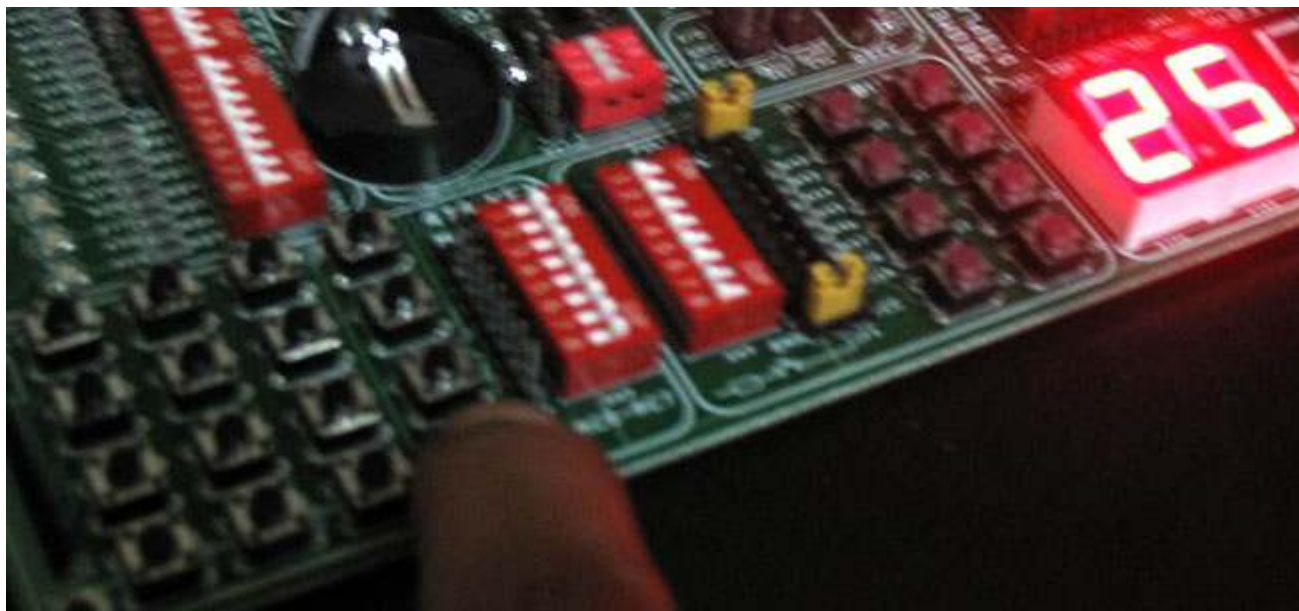


**Hình 5-6** Kết quả sau khi nhấn Program

- 5) Thực hiện menu lệnh Debugger → Run hoặc nhấn icon trên tool bar, chúng ta sẽ thấy LED 7 đoạn hiển thị "FF" trên EASYPIC\_PRO development board, như Hình 5-7; bây giờ nếu nhấn một phím LED 7 đoạn sẽ hiển thị số của phím đó, ví dụ nếu K25 được nhấn, LED sẽ hiển thị "25", trình bày như hình Hình 5-8.



**Hình 5-7** kết quả không có phím nào nhấn



**Hình 5-8** kết quả khi K25 được nhấn

Sau khi debug một số lần, chúng ta cơ bản đã thực hiện một số kết quả mong muốn và hoàn thành debug. Ngắt các kết nối

## Nạp chương trình đến chip đích

*Tương tự như trên, dùng PICKIT 2 làm công cụ điển hình. ICD2, PICKit 3 cách làm tương tự*

Các mã nguồn chuẩn bị ở trên đã thông qua quá trình debug bằng cách sử dụng PICKIT 2, và đã đạt được mục tiêu chúng ta mong muốn, bây giờ chúng ta sẽ sử dụng PICKIT 2 để nạp/ghi file KEY4x4.HEX vào chip MCU đích để nó có thể chạy ở chế độ Offline (nghĩa là sau khi nạp, MCU sẽ chạy độc lập)

- 1) Kết nối PICKIT 2 và EASYPIC\_PRO development board theo phương pháp thể hiện trong Hình 3-1.
- 2) Thực hiện menu lệnh Programmer → Select Tools để chọn PICKIT 2 làm công cụ nạp cho MCU
- 3) Thiết lập cấu hình bits và các thông số khác theo điều kiện cụ thể bằng cách nhấn menu lệnh Configure → Configuration Bits.
- 4) Thực hiện menu lệnh Programmer → Program, để nạp/ghi file KEY4x4.HEX vào MCU đích.
- 5) Sau khi nạp xong kết quả sẽ chạy ngay (Tương tự như hình 5-7 và hình 5-8).

Ngoài phương pháp nạp chương trình bằng MPLAB, chúng ta có thể sử dụng phần mềm **PICKit 2 Programmer** Đây là phần mềm sử dụng riêng cho PICKit 2 để nạp file hex vào chip. Phần này đã trình bày trước đây trong phần 3.1.3 *Sử dụng phần mềm PICKit 2 Programmer (trang 9)*

Xem thêm hướng dẫn sử dụng PICKit2 phiên bản tiếng Việt hoặc tiếng Anh kèm theo trong CD

## Phụ lục 1: Thông tin liên hệ và chính sách bảo hành

Công ty TNHH Giải Pháp Điện tử Thiên Minh (TMe)

**Địa chỉ:** 226 Tân Phước, Phường 6, Quận 10, Thành Phố Hồ Chí Minh, Việt Nam

**Điện thoại:** 08.3957.3224 **Fax:** 08.6264.3029

Web: [www.tme.com.vn](http://www.tme.com.vn) – [www.tme.vn](http://www.tme.vn)

**Email:** Bán hàng: [sale@tme.com.vn](mailto:sale@tme.com.vn) – Hỗ trợ kỹ thuật: [support@tme.com.vn](mailto:support@tme.com.vn)

**Bảo hành sản phẩm:** Chính sách bảo hành sản phẩm của chúng tôi như sau

- Bảo hành 12 tháng đối với các sản phẩm có hộp bảo vệ
- Bảo hành 03 tháng đối với sản phẩm không có hộp bảo vệ.

*Như vậy, **EasyPIC\_PRO** thuộc diện bảo hành 03 tháng*

Trong đó các thành phần sau đây không được bảo hành: Các linh kiện và phụ kiện kèm theo EasyPIC\_PRO có thể tháo rời được như: các IC gắn trên socket, LCD, PIC, và các loại cáp EasyPIC và mọi linh kiện/phụ kiện kèm theo sau khi xuất bán đều được kiểm tra kỹ lưỡng, bảo đảm hoạt động tốt khi đến tay người dùng.

Sản phẩm được hỗ trợ kỹ thuật và bảo trì (tính phí nếu thay thế linh kiện) trong suốt thời gian sử dụng.