

Bài viết này viết cho IRF.

Grenoble, 12/05/2005

Falleaf@irfvn.com

Chào các bạn,

Có thể nói, hiện nay vi điều khiển đã rất phổ biến ở Việt Nam, và được ứng dụng rất nhiều. Những sinh viên ngành Điện, Điện Tử, Cơ Điện Tử, Tin Học, Viễn Thông... hầu như ai cũng biết cách để làm việc với vi điều khiển. Tuy nhiên, để viết một bài viết tổng quát về vi điều khiển và việc lập trình cho vi điều khiển, quả thật là quá khó khăn đối với tôi.

Do vậy, tôi đặt ngay vấn đề là bài viết này sẽ hướng vào những bạn sinh viên mới bắt đầu tiếp cận với vi điều khiển. Bài viết sẽ được chia thành 3 nội dung chính. Nội dung thứ nhất, tôi trình bày khái quát về vi điều khiển. Nội dung này được đề cập trong rất nhiều tài liệu, và dường như là vấn đề cơ bản, do vậy, tôi chỉ trình bày những ý kiến cá nhân của tôi sao cho thật phù hợp với các bạn sinh viên mới tiếp cận với vi điều khiển. Nội dung thứ hai, tôi trình bày về cách tiếp cận một họ vi điều khiển, hay cụ thể hơn là các bạn cần gì để bắt đầu học vi điều khiển. Nội dung cuối cùng, tôi sẽ liệt kê các bước học vi điều khiển một cách tương đối rõ ràng, mà cá nhân tôi cho rằng nó hữu ích cho các bạn. Bởi vì những bước này tôi cũng tham khảo ở một số tài liệu hướng dẫn, và khi học thì tôi cũng phải lần mò và đi theo con đường đó.

1) Vi điều khiển

Trong rất nhiều tài liệu cơ bản đều đề cập đến khái niệm vi điều khiển là gì. Do vậy, tôi lướt qua các khái niệm cơ bản này. Tôi chỉ đề cập đến các khái niệm, mà các bạn cần quan tâm khi bắt đầu học, hoặc lựa chọn một vi điều khiển cho một ứng dụng nào đó của các bạn. Từ những khái niệm tôi đề cập, các bạn có thể tham khảo từ các tài liệu chi tiết hơn, và rồi các bạn sẽ nắm được toàn bộ những vấn đề cơ bản của một họ vi điều khiển nào đó.

a) Kiến trúc vi điều khiển

Thực ra vi điều khiển cũng là một cấu trúc siêu nhỏ, gồm các linh kiện điện tử ở kích thước micro hoặc nano, các linh kiện này được kết hợp với nhau và được nối với các thiết bị bên ngoài qua các chân vi điều khiển. Chính vì vậy, hiểu rõ được kiến trúc vi điều khiển, các bạn sẽ hiểu rõ được mình đang làm việc với cái gì, và cái đó làm việc như thế nào.

Kiến trúc máy tính hay kiến trúc vi điều khiển cũng tương tự nhau. Do đó, các bạn có thể tìm hiểu về kiến trúc máy tính, để hiểu rõ về kiến trúc vi điều khiển. Hai kiến trúc vi điều khiển phổ biến hiện nay, là kiến trúc Von Neumann và kiến trúc Harvard. Sự khác biệt chủ yếu giữa hai kiến trúc này, chính là việc tổ chức bộ nhớ dữ liệu và bộ nhớ chương trình. Kiến trúc Von Neumann tổ chức bộ nhớ dữ liệu và bộ nhớ chương trình chung với nhau, chính vì vậy, đường truyền (bus) của kiến trúc Von Neumann là đường truyền chung. Trong khi đó, kiến trúc Harvard tách rời bộ nhớ dữ liệu và bộ nhớ chương trình.

Mỗi kiến trúc này có một lợi điểm riêng rẽ khác nhau. Kiến trúc Von Neumann tận dụng được tài nguyên bộ nhớ, trong khi đó kiến trúc Harvard sẽ đạt tốc độ xử lý cao hơn, mặt khác đường truyền dữ liệu và đường truyền lệnh điều khiển (chương trình) có thể có dung lượng khác nhau.

b) Tập lệnh

Tập lệnh ở đây được coi là tập mã lệnh nhị phân, và chúng ta chưa nên vội đề cập đến ngôn ngữ lập trình. Bản chất của tập lệnh là một tập hợp các mã nhị phân, mà từ đó các đơn vị xử lý trung tâm (CPU) nhận biết và thực hiện. Dữ liệu được CPU xử lý là các số nhị phân. Chính vì vậy, tập lệnh dù thế nào đi nữa cũng sẽ thực hiện mấy việc chính sau đây:

- tính toán các con số nhị phân.
- các lệnh để chuyển các giá trị ra thành tín hiệu điện tử ở chân linh kiện
- các lệnh di chuyển các giá trị giữa các thanh ghi
- các lệnh điều khiển con trỏ chương trình

Những điều này tưởng chừng đơn giản, nhưng thiết nghĩ nó sẽ giúp ích cho các bạn khi học và hiểu một tập lệnh, các bạn có thể phân loại chúng đúng cách.

Tập lệnh hiện nay được phân làm hai loại, tập lệnh RISC và tập lệnh CISC. Các bạn có thể tìm hiểu sâu hơn qua các bài viết trên các diễn đàn điện tử, tin học, hoặc những tài liệu tin học khác.

Như vậy, từ kiến thức này, những khó khăn mà rất nhiều người gặp phải khi thực hiện lệnh tác động lên những thanh ghi lệnh (vì nó cũng chỉ là dãy số nhị phân) sẽ được giảm bớt đi khá nhiều.

c) Chức năng

Câu hỏi lớn nhất đặt ra khi cầm một con vi điều khiển trên tay, chính là chúng ta sẽ làm gì với nó? Hiện nay rất nhiều loại vi điều khiển ra đời, và rất nhiều tính năng được tích hợp vào trong vi điều khiển dưới dạng phần cứng. Tuy nhiên, tựu chung lại thì mọi việc cũng đều nằm ở việc điều khiển động cơ và đọc cảm biến. Một cánh cửa tự động là một cái cảm biến hồng ngoại và một cái động cơ. Đại đa số những gì tự động đều có dính đến động cơ trong đó, vì nếu không có động cơ thì làm sao nó biến đổi điện năng thành cơ năng được? Mà cái gì có đi thì cũng có lại, đã có động cơ làm việc, thì phải có cảm biến để quan sát.

Cái lý luận này sâu cũng không sâu, mà nông cũng không nông, nhưng nó lại là cơ sở để những nhà sản xuất tung ra các sản phẩm vi điều khiển của mình. Và cũng từ đó, các bạn cũng dễ phân loại để lựa chọn vi điều khiển cho mình dùng. Ngoại trừ hai loại này, thì những loại khác chúng ta cứ xem là loại thứ ba.

Chúng ta cần nhóm loại để làm gì? Bởi vì tính năng của vi điều khiển, có liên quan trực tiếp đến số chân của vi điều khiển. Cho nên việc phân loại và hình dung ra một bản đồ đầy đủ các loại, các chức năng của vi điều khiển, sẽ giúp các bạn lựa chọn vi điều khiển tốt hơn. Một người mới học, để có một tầm nhìn rộng về vi điều khiển mà lựa chọn cũng thật là khó, nhưng nếu tìm tòi,

hiểu và phân loại được một số loại vi điều khiển, tự nhiên sẽ hiểu rõ được cái bản đồ tương đối phức tạp này.

Làm thế nào để biết? Đơn giản thôi, trong tất cả các datasheet được cung cấp cho từng loại vi điều khiển, đều có mô tả rõ tính năng của từng loại. Các bạn tự mình tìm hiểu và phân loại một số vi điều khiển thông dụng (mà các bạn có thể mua) rồi tự đánh giá chúng về mặt tính năng qua các datasheet. Và để đọc hiểu được các tính năng này, thì các bạn phải có một kiến thức nền, ví dụ như timer là gì? pwm là gì? capture là gì? i2c là gì? usart là gì? giao tiếp CAN là gì?... Những cái đó, bạn không cần học từ nhiều con vi điều khiển, vì những tính năng đó hoàn toàn giống nhau, chỉ có cách để làm cho nó hoạt động là khác nhau thôi. Vậy nên, các bạn cứ lấy một datasheet của một con vi điều khiển bất kỳ nào đó, đọc và cố gắng hiểu mục đích của các tính năng đó, và chúng ta có thể dùng kiến thức này cho bất kỳ con vi điều khiển nào khác.

2) Tiếp cận với vi điều khiển

Về mặt khoa học mà nói, chúng ta có hai hướng tiếp cận chính với một đối tượng, một là nghiên cứu để phát triển đối tượng đó, hai là sử dụng đối tượng đó. Đối với vi điều khiển, ở thời điểm hiện tại năm 2005, tôi khuyên rằng chúng ta không nên nghiên cứu để phát triển vi điều khiển, mà nên học để sử dụng nó.

Vậy chúng ta cần những gì để tiếp cận và học vi điều khiển?

Trước tiên, nên hiểu quá trình thao tác từ đầu đến cuối để cho một con vi điều khiển bất kỳ hoạt động đó là:

a) Chúng ta cần làm cái gì?

Khi đặt câu hỏi này, chúng ta nghĩ ngay đến tính năng, số chân, và kích thước cần thiết của vi điều khiển. Và chúng ta phải lựa chọn được con vi điều khiển chúng ta cần dùng, tất nhiên kèm theo ngay sau đó là chúng ta có thể mua được nó nữa.

Tôi nói rằng chúng ta phải biết rằng chúng ta có thể mua được nó, chứ đừng đi mua nó vội, mà hãy xem những vấn đề tiếp theo.

b) Lập trình

Tất nhiên công việc của bạn là công việc lập trình, để làm cho con vi điều khiển đó hoạt động theo ý bạn muốn trong giới hạn các tính năng của nó. Ở đây có một điểm rất vui đó là nếu bạn xét lại từ thườ "khai sinh lập địa" của con vi điều khiển, chức năng và tập lệnh của nó thật đơn giản, nhưng những công việc như bây giờ nó vẫn hoàn toàn làm được. Đó là vì sao? Là vì bản chất của nó vẫn chỉ là thao tác với những con số nhị phân. Do vậy, nếu một con vi điều khiển không hỗ trợ một tính năng nào đó, thì chúng ta cũng đều có thể thực hiện tính năng đó bằng phần mềm và một vài phần cứng hỗ trợ bên ngoài.

Nhưng tạm dừng chuyện đó lại, các bạn muốn con vi điều khiển hiểu được những chữ là chữ mà các bạn sẽ dùng để lập trình, thì các bạn phải biến đổi các chữ đó thành ra các con số nhị phân, thể hiện dưới dạng file .HEX

Muốn làm được điều này, các bạn cần có một chương trình dịch.

Việc sau đó là bạn phải biết ngôn ngữ lập trình nào đó, thông thường khi mới bắt đầu, người ta dùng ASM

c) Nạp chương trình

Bạn viết chương trình trên máy tính, bạn đã dịch ra được file HEX để vi điều khiển khi nhận được có thể hiểu được bạn muốn làm gì. Vậy làm sao để đưa nội dung đó vào cho vi điều khiển?

Các bạn cần có một mạch nạp và một chương trình nạp phù hợp với mạch nạp đó. Công việc nạp được cụ thể hoá bằng việc cắm mạch nạp vào máy tính, bật chương trình nạp, load file .HEX vào chương trình nạp, lựa chọn vi điều khiển cần nạp, cài đặt các thông số nạp, nhấn nút Program trên chương trình nạp. Sau khi nạp xong, chương trình nạp nào cũng thông báo kết quả nạp được hay không nạp được. Nếu không có gì sai, nhấn OK là xong.

d) Mạch chạy vi điều khiển

Một mạch chạy vi điều khiển tối thiểu cần có nguồn cấp điện cho vi điều khiển hoạt động. Nguồn này phải cấp điện áp từ 2.5V đến 5.5V và dòng từ 150mA đến 300mA tùy theo mỗi loại vi điều khiển. Nguồn này chỉ cung cấp cho vi điều khiển hoạt động, không cung cấp cho các thiết bị ngoại vi. Mạch reset để reset hoạt động của vi điều khiển. Mạch dao động (có hoặc không có) có thể được tích hợp sẵn trong vi điều khiển, chế độ này gọi là dao động nội. Nếu không có mạch dao động nội, phải dùng thạch anh để tạo dao động cho vi điều khiển. Thạch anh thường dùng từ 38KHz cho đến 40MHz tùy theo từng loại vi điều khiển.

Chính vì những đặc điểm này, nếu chúng ta không tính đến tất cả các thiết bị ngoại vi cần điều khiển, chúng ta hãy mạnh dạn thiết kế một mạch chuẩn cho vi điều khiển, cung cấp điện áp 5V, dòng 300mA, mạch reset và mạch dao động. Khi dùng bất kỳ loại vi điều khiển nào, cũng có thể dùng mạch này.

Nguồn cho thiết bị ngoại vi, tôi khuyên rằng nên thiết kế riêng cho từng thiết bị ngoại vi, và cho rằng thiết bị ngoại vi là bất kỳ thiết bị nào nối trực tiếp với mạch vi điều khiển nêu trên.

Ngoài ra, cần lưu ý rằng, vi điều khiển nhận và xuất tín hiệu điện trên các chân từ 2.5V đến 5.5V (phụ thuộc vào nguồn), và từ 12mA đến 20mA (tùy loại). Vì vậy, các thiết bị ngoại vi phải được thiết kế sao cho khi giao tiếp với vi điều khiển, trực tiếp nối vào các chân vi điều khiển, và có dòng xuất nhập, cũng như điện áp logic như trên.

e) Kết luận

Trong tất cả những vấn đề cần có này, chúng ta thấy rằng các mạch nguồn, reset và dao động là khá đơn giản, và hầu như được cung cấp một cách chi tiết bởi nhà sản xuất. Chính vì thế, chỉ cần lật datasheet ra và thiết kế giống hệt.

Đối với thiết bị ngoại vi, thì tùy theo ứng dụng mà chúng ta thiết kế. Tôi chưa đề cập đến ở đây.

Đối với chương trình nạp và mạch nạp, chúng ta hoàn toàn có thể lên các diễn đàn điện tử trong và ngoài nước, hoặc tìm các nguồn cung cấp mạch nạp cho từng loại vi điều khiển. Đa số các loại vi điều khiển đều có những người thiết kế mạch nạp làm sẵn, chỉ việc tìm và sử dụng. Chúng ta tuyệt đối không nên quan tâm đến việc làm sao để làm ra mạch nạp, hay viết ra được chương trình nạp, hoặc giả nguyên lý nạp như thế nào.

Những người tìm hiểu sâu với mục đích giảng dạy thì cần thiết làm việc này, để giúp sinh viên hiểu rõ hơn vấn đề, hoặc phòng khi sinh viên đặt câu hỏi. Nhưng những người dùng, không cần quan tâm đến việc này. Chúng ta chỉ cần tìm mạch nguyên lý, làm mạch, và từ đó về sau cứ thế là dùng. Thông thường, và gần như là mặc định, bất cứ người nào cung cấp mạch nạp, cũng đều cung cấp theo đó chương trình nạp. Vậy kể từ đây, chúng ta cũng không cần quan tâm đến mạch nạp và chương trình nạp nữa.

Chương trình dịch, hầu hết các nhà sản xuất cung cấp chương trình dịch cho ngôn ngữ ASM là miễn phí. Có một số nhà sản xuất cung cấp cả môi trường soạn thảo miễn phí luôn (như Microchip PIC chẳng hạn). Các chương trình dịch từ ngôn ngữ cấp cao C, Pascal, Basic, ... thường được bán với giá khá cao. Tuy nhiên, ở Việt Nam, thực tế là chúng ta dùng rất nhiều những chương trình không có bản quyền. Tôi không cổ vũ cho việc dùng trình dịch không có bản quyền, nhưng thực tế là thực tế, bây giờ hầu như các chương trình dịch ngôn ngữ cấp cao cho hầu hết các loại vi điều khiển đều được cung cấp đầy đủ trên các diễn đàn của cả Việt Nam lẫn nước ngoài. Vì vậy, về phần chương trình dịch, các bạn cũng chẳng phải lo lắng gì nữa.

Vậy điều quan trọng nhất, các bạn cần phải học, đó là học cách viết chương trình. Tôi đã phân tích vấn đề này ở diễn đàn điện tử www.diendandientu.com và sẽ đăng lại bài viết này tại www.picvietnam.com (diễn đàn mà tôi sắp mở ra). Để viết được chương trình, các bạn phải hiểu rằng, bạn viết chương trình cho một vi điều khiển, tức là sử dụng các tính năng của vi điều khiển. Vậy các bạn phải theo các bước sau:

- tính năng đó hoạt động như thế nào? tại sao chúng ta dùng tính năng đó?
- làm sao để kích hoạt cho vi điều khiển hiểu rằng chúng ta cần dùng tính năng đó?
- khi làm việc với một tính năng bất kỳ, chúng ta phải tương tác với những thanh ghi đặc biệt nào trong vi điều khiển?
- một số thuật toán để thực hiện công việc

Như vậy, các bạn sẽ nắm ngay được rằng, để điều khiển một thiết bị ngoại vi nào đó, chúng ta phải xác định công việc cần phải làm, xác định xem dùng tính năng nào của vi điều khiển để giải quyết công việc, khi đã nắm rõ hết các tính năng của vi điều khiển, phần còn lại là một chút logic để sắp xếp chương trình chạy cho có hệ thống.

Chính vì vậy, các bạn sẽ thấy rằng có rất nhiều thuật toán kinh điển đã được xây dựng, các thư viện code... Bởi vì thực tế, công việc cũng chỉ quanh đi quẩn lại có bao nhiêu đó. Đầu óc logic và khả năng lập trình một phần là do thiên phú, một phần là do sự rèn luyện và tích lũy mà có.

Chính vì vậy, cùng một việc làm, người này làm tốt hơn người kia, chủ yếu là sự logic và biết nhiều thuật toán xử lý công việc.

Tóm lại, điều quan trọng nhất các bạn phải học là cách thực hiện một chương trình sao cho thật logic, hiệu quả, nhờ sự hướng dẫn của mọi người để tìm đến các thư viện code, những bài tập mẫu để đỡ mất công suy nghĩ những vấn đề bé nhỏ mà người khác đã giải quyết tốt.

Rất nhiều người tự tin vào khả năng sáng tạo của mình, mà thiếu sự rèn luyện và tích lũy kinh nghiệm lập trình, cho nên cho rằng bài viết của tôi có phần dở hơi, vì không cung cấp thêm các kỹ thuật gì cho họ. Tuy nhiên tôi lại cho rằng khi hiểu mình *cần phải làm gì, học gì, tích lũy gì, giảm bớt cái gì*, nó quan trọng hơn rất nhiều so với việc biết thêm một thứ kiến thức nhỏ mọn từ người khác. Cá nhân tôi khuyên các sinh viên mới bắt đầu làm việc với vi điều khiển một lời khuyên chân thành là các bạn hãy theo những bước tôi trình bày, học tập và tự mình đánh giá các vấn đề, hình thành một bản đồ kiến thức, rồi tự thực hiện một đề tài bất kỳ. Sau khi làm xong một đề tài bất kỳ, kể cả đề tài đó chỉ là làm nhấp nháy một cái đèn led, các bạn sẽ thấy rằng, các đề tài khác rồi

cũng tương tự. Nhưng vấn đề quan trọng là phải xác định được những bước thực hiện như tôi đề cập trên kia và tiếp sau đây.

3) Các bước thực hiện một đề tài với vi điều khiển

Tôi viết ra đây các bước để một người mới học có thể theo và tiếp cận ngay được với vi điều khiển

a) Tìm hiểu một số khái niệm cần thiết:

- Hiện nay có 3 họ vi điều khiển nổi tiếng là 8051 (bao gồm AVR), PIC và Motorola. Nếu bắt đầu học, chọn một trong 3 loại này để bắt đầu.
- Tìm hiểu về kiến trúc máy tính, hiểu cơ bản về kiến trúc Harvard và kiến trúc Von Neumann
- Tìm hiểu khái niệm RISC và CISC
- Khái niệm thanh ghi
- Khái niệm Stack
- Khái niệm con trỏ (pointer)
- Khái niệm địa chỉ và địa chỉ gián tiếp của một thanh ghi
- Khái niệm timer
- Khái niệm ngắt

Vì sao cần những khái niệm này, bởi vì những khái niệm này sẽ là nền tảng để các bạn hiểu sâu hơn về tính năng và phương thức làm việc của một vi điều khiển. Hơn nữa, những khái niệm này đều là những khái niệm tổng quát, và đều bắt nguồn từ kiến trúc máy tính mà ra. Do vậy, việc tìm kiếm tài liệu cũng không khó khăn gì.

b) Hiểu các tính năng của vi điều khiển:

Các bạn download ngay một datasheet bất kỳ loại vi điều khiển nào, tôi lấy thí dụ download datasheet của PIC16F877A, cho dù là các bạn học AVR hay Motorola hay 89C51.... Các bạn cứ download datasheet này về.

Trong đó, các bạn không cần chú ý chi tiết đến kiến trúc của vi điều khiển đó, mà các bạn chỉ cần xem phần mô tả tính năng. Các bạn sẽ thấy trang đầu tiên của datasheet luôn liệt kê đầy đủ các tính năng của vi điều khiển. Một số tính năng như: timer 8 bit, timer 16 bit, có bao nhiêu ngắt, tính năng ADC, DAC, tính năng PWM, Comparator, Capture, giao tiếp USB, CAN, I2C, UASRT, Parallel, ... Đây là các tính năng phổ biến của vi điều khiển hiện nay. Các bạn lật vào trong, đọc để hiểu những tính năng này làm những gì.

Vd: Chỉ cần hiểu ADC (Analog to Digital Converter) dùng để biến đổi một tín hiệu điện áp Analog vào một chân nào đó của vi điều khiển, biến đổi nó qua giá trị số (Digital) bằng cách so sánh với một điện áp tham chiếu (Reference Voltage). Điện áp tham chiếu có thể là điện áp VDD (điện áp nguồn) hoặc điện áp tham chiếu được đưa vào một chân khác. Nguyên lý biến đổi này được thực hiện như thế nào? Thế nào là ADC 10 bit, 12 bit, 8 bit...

Chỉ cần như vậy thôi, các bạn không cần hiểu nhiều hơn nữa.

Nói điều này cũng thật buồn cười cho các bạn đã học lâu rồi, nhưng đây là các bạn mới học, thì việc này vô cùng quan trọng. Vì rất nhiều bạn lên mạng và hỏi rằng PWM là cái gì? Thế nào là Duty cycle?

Do vậy, những khái niệm này, nếu các bạn nắm và hiểu được rồi, thì vấn đề chỉ còn là ứng dụng sử dụng nó thông qua việc lập trình trên vi điều khiển mà bạn chọn nữa thôi. Mà các tính năng này thì vi điều khiển nào cũng giống nhau, cho nên chỉ cần đọc một datasheet thì sẽ biết hết.

c) Tìm dụng cụ học tập

Tất nhiên, đi học phải có dụng cụ học tập. Bạn muốn học con vi điều khiển A, thì buổi này không phải là thời buổi bao cấp, phát phiếu đi lĩnh vi điều khiển về xài. Cho nên, tốt nhất là bạn làm cách nào đó muốn học thì phải tìm mua bằng được nó rồi tính chuyện học gì rồi học.

Cầm con vi điều khiển trong tay rồi mới tính chuyện học, lỡ học không được thì sao? Thì kiếm cái búa phang mạnh một cái vào con vi điều khiển, mở ruột nó ra xem coi nó có cái gì trong đó, thấy nó đen thui, chẳng có gì. Một cái cục

đen thui mà mình phải học về nó mà làm gì? Thôi vứt mẹ nó đi. Cách đó là hay nhất.

Tiếp tục trang bị dụng cụ học tập, như tôi đã nói, chúng ta cần có: chương trình dịch, mạch nạp và chương trình nạp. Hiện nay diễn đàn điện tử www.diendandientu.com và diễn đàn tôi sắp mở www.picvietnam.com có cung cấp đủ các công cụ này. Muốn tìm hiểu thêm, các bạn lên các diễn đàn nước ngoài, họ cung cấp không thiếu một thứ gì. Thậm chí không cần lên diễn đàn, đi hỏi thẳng Google là có ngay.

Keyword: PIC programmer, PIC bootloader, AVR programmer, free pcb programmer,... nhiều lắm đánh cái gì vào cũng tìm ra được hết. Nhưng các bạn nên tìm cái nào miễn phí. Sau khi tìm xong nhớ quay lại share cái link đó cho mọi người để mọi người đỡ mất công tìm kiếm.

Keyword: PIC C compiler, AVR C compiler... cứ vậy mà tìm tới tới

Sau khi tìm được sơ đồ nguyên lý mạch nạp, chương trình nạp cho con vi điều khiển của mình, các bạn nhanh chóng làm ngay mạch nạp. Load một chương trình mẫu bất kỳ đơn giản vào vi điều khiển để kiểm tra mạch nạp hoạt động tốt hay không. Nếu tốt thì oki, nếu không thì lại bò lên các diễn đàn để hỏi. Những thằng rảnh rỗi như tôi sẵn sàng trả lời cho các bạn.

d) Học viết chương trình

Học viết chương trình có thể chia cụ thể làm 2 cách học, học để viết cho vi điều khiển chạy được thì nhảy thẳng vào học ngôn ngữ cấp cao, học cấp tốc để đối phó với một đồ án nào đó. Thậm chí vài dòng lệnh ban đầu viết sai, nhảy lên diễn đàn hỏi, tìm thầy hướng dẫn hỏi, tìm bạn hỏi... cuối cùng rồi cũng viết xong cho nó chạy được. Cái nào thấy khó khăn, không biết làm thì tìm code mẫu bằng ngôn ngữ cấp cao, thế là xong.

Cách thứ hai rườm rà hơn, đó là học viết ASM. Học ASM khá công phu, phải hiểu chi tiết cấu trúc vi điều khiển, phải điều khiển từng thanh ghi...

Tuy nhiên, tôi khuyên khi mới học, các bạn nên bắt đầu bằng việc học ASM. Khi nắm vững cách viết ASM cho một con vi điều khiển nào đó, chuyển sang con vi điều khiển khác không mấy khó khăn, và các bạn sẽ giống như một chuyên gia về nó một khi các bạn đã hiểu tường tận mọi vấn đề.

Ban đầu có vẻ sẽ hơi khó xoi và mệt mỏi, nhưng sau này thì các bạn sẽ thấy hoàn toàn tự tin vào kiến thức mình thu thập được.

e) Làm các bài ứng dụng

Sau khi có chương trình nạp và mạch nạp, các bạn thường hay làm ngay bài tập nhấp nháy đèn led là bài tập khá kinh điển. Thực ra đó là gì, các bạn thực hiện một mạch chạy cho vi điều khiển, và thiết bị ngoại vi là cái đèn led. Nhớ lại là thông thường vi điều khiển xuất tín hiệu 5V, 12mA.

Nhưng cái đèn led chỉ dùng ở khoảng 2.5V 12mA, nên chúng ta nối một cái điện trở nối tiếp với cái đèn led.

Như vậy, chúng ta xác định rõ, chúng ta phải làm một mạch chạy cho vi điều khiển, bao gồm nguồn, mạch reset, mạch dao động. Các chân còn lại được nối với dây chân cắm để nối ra ngoài. Cái này, chúng ta sẽ gọi là mạch test. Và cho đến khi chúng ta học thành thạo một con vi điều khiển nào đó, thậm chí sau này dùng vào việc thực hiện những dự án lớn với con vi điều khiển đó, mạch test này vẫn có giá trị của nó.

Làm một mạch test là điều thứ tư phải làm.

Quay lại chuyện cái đèn led. Cái đèn led và điện trở, thì vai trò của cái đèn led và điện trở là thiết bị ngoại vi, cái đèn led là thiết bị cần điều khiển, cái điện trở là "mạch" giao tiếp. Điều này có nghĩa là, không cần biết bạn làm cái gì, nhưng các bạn thiết kế được một mạch ngoại vi, cắm vào vi điều khiển để điều khiển nó, thì bạn chỉ việc cắm nó vào các chân được đưa ra ngoài của mạch test (các socket, chân cắm...). Nhớ rằng Mass phải được nối chung trong tất cả các mạch điện tử. Một số mạch điện tử phức tạp dùng nhiều mass khác nhau, thì ít nhất cũng phải có một mass chung với mass của vi điều khiển.

f) Các bước học tính năng vi điều khiển

Sau khi ù ù cạc cạc làm nhấp nháy cái đèn led, theo một sự hướng dẫn nào đó, thậm chí là copy and paste, nhưng cái đèn nó nhấp nháy, chứng tỏ bạn làm tốt mọi việc từ đầu đến khi chính thức bắt đầu học vi điều khiển.

Hãy tiến hành các bài tập sau:

- Học các thanh ghi đặc biệt, bản đồ bộ nhớ (memory map) (học lý thuyết)
- Học cách xuất trên từng chân vi điều khiển (bài tập tắt mở đèn led)
- Học viết hàm delay (làm đèn nhấp nháy)
- Đèn nhấp nháy dùng timer
- Bấm một nút, đèn bật/tắt (nhập port, nhập trên từng chân, dùng ngắt ngoài)
- Điều khiển động cơ (PWM)
- Đọc encoder (Capture/Comparator)
- Đọc ADC bằng cách xoay biến trở
- Giao tiếp RS232
- Giao tiếp I2C

Sau khi các bạn học xong các bài học này, việc học tiếp các bài học khác sẽ không phải là vấn đề khó khăn với các bạn nữa, các bạn hoàn toàn có thể tham khảo datasheet để thực hiện một mình.

Các bạn cần phải phân biệt thật rõ vấn đề học các chức năng của PIC, với việc học để sử dụng một số thiết bị ngoại vi thông dụng. Ví dụ như việc điều khiển LCD, chẳng qua chỉ là việc xuất nhập PORT, hoặc như bảng điện, quang báo, led 7 đoạn, nó cũng chỉ là việc xuất nhập port. Trong các bài học này, đó là các bạn học các thiết bị ngoại vi thông thường khi dùng vi điều khiển, chứ không phải học các tính năng của vi điều khiển.

Nhưng trái lại, nếu các bạn học một bài học đơn giản vô cùng đó là dùng tính năng PWM để làm bộ biến đổi DAC (Digital to Analog Converter) thì chính là các bạn lại học cách dùng tính năng PWM.

Việc định hướng các bài học, và mục tiêu giảng dạy phụ thuộc chủ yếu vào người hướng dẫn, tuy nhiên tôi phân tích ý này ở đây để các bạn sinh viên

hiểu vì sao có thể một số bạn đã thực hành rất nhiều đề tài, nhưng vẫn cảm thấy mình không đủ kiến thức để thực hiện một đề tài lớn với vi điều khiển. Đó là vì các bạn chưa thực sự dùng hết các tính năng của vi điều khiển, mà mới chỉ dùng một vài tính năng và được rèn luyện đi rèn luyện lại.

4) Kết luận:

Tôi dành thời gian viết bài viết này, không mong muốn gì hơn là giúp các bạn sinh viên mới học, mới bắt tay làm quen với vi điều khiển có một cái nhìn tổng quát ban đầu và có một định hướng rõ ràng cho việc học tập của mình. Cá nhân tôi cũng từng là người mới học, và bất quá thì tôi chỉ học trước các bạn một vài năm. Đến bây giờ tôi không còn dùng vi điều khiển và lập trình cho vi điều khiển nữa, nhưng tôi cũng muốn chia sẻ những kinh nghiệm của tôi khi bắt đầu học đến khi làm được một số việc hữu ích bằng vi điều khiển, đến cho các bạn. Hy vọng các bạn bắt đầu học, và học tập mau tiến bộ.

Nếu các bạn có những thắc mắc gì về vi điều khiển Microchip PIC, các bạn có thể email theo địa chỉ

falleaf@irfon.com

hoặc tham dự vào diễn đàn

www.picvietnam.com

Chúc mọi điều tốt lành!

FALLEAF – IRF

Bài viết này đã gửi cho IRF, và được đăng trên www.irfon.com. Vì vậy, bài viết này là phiên bản copy từ trang web trên. Tôi chuyển dạng thành file .pdf.