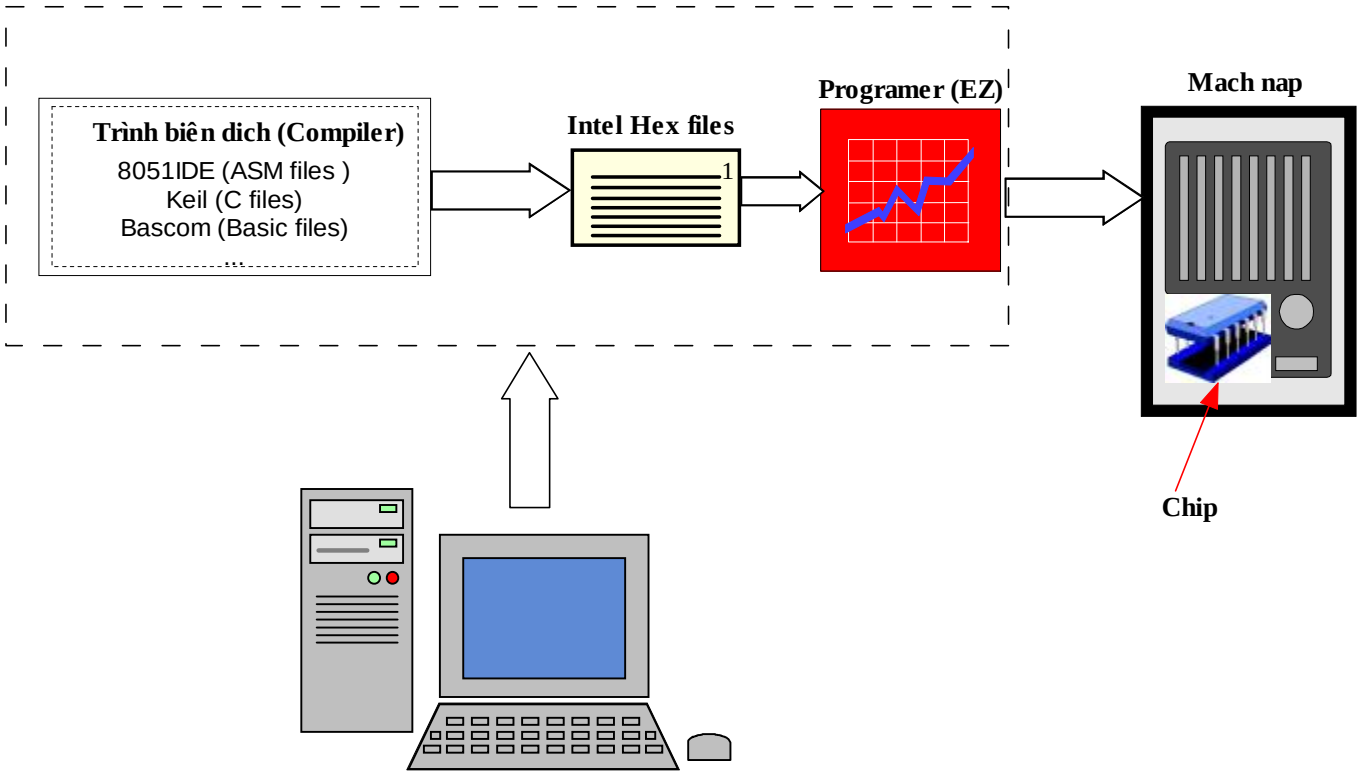
BẮT ĐẦU LÀM QUEN VỚI VI ĐIỀU KHIỂN

Bạn đang có ý định tìm hiểu về vi điều khiển để tự mình tạo những ứng dụng nho nhỏ, tôi nghĩ tài liệu này có thể giúp bạn!

Vi điều khiển thì có rất nhiều loại, nhiều kích cỡ, nhiều mức giá…tài liệu này chỉ giới thiệu cho bạn một loại vi điều khiển thuộc vào dạng "phổ thông" nhất, vi điều khiển AT89C51 của Atmel. Tôi cũng không có ý định giới thiệu cấu trúc của vi điều khiển này mà chỉ giúp bạn biết bạn nên bắt đầu như thế nào, tôi sẽ hướng dẫn từng bước vì mục đích của tài liệu là giúp những bạn "chưa biết gì". Bạn có thể làm theo những hướng dẫn này trước rồi sao đó tìm hiểu chi tiết về AT89C51 sau hay ngược lại đều được .

Có thể hiểu đơn giản vi điều khiển là những vi mạch điện tử tích hợp (chip) mà bạn có thể lập trình để nó thực hiện những nhiệm vụ mà bạn mong muốn. Khác với vi xử lí, các vi điều khiển ngoài chức năng xử lí dữ liệu, thuật toán…còn được tích hợp các bộ chức năng đặc biệt khác, các vi điều khiển có các ngõ vào/ra để nhận và xuất dữ liệu, các bộ timer xử lí thời gian, các bộ trao đổi dữ liệu theo một số chuẩn giao tiếp, thậm chí một số loại vi điều khiển còn có cả bộ chuyển đổi AD, bộ điều khiển động cơ…nói chung chúng ta có thể làm được rất nhiều việc với vi điều khiển!

Bây giờ chúng ta bắt đầu với vi điều khiển AT89C51.



Hình 1

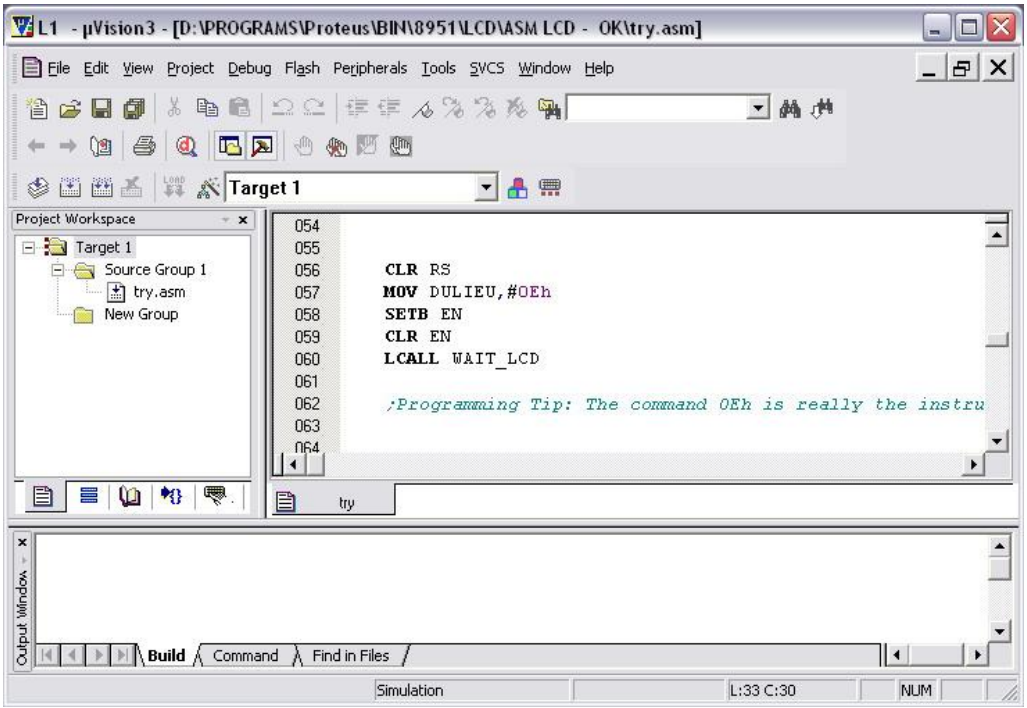
Trước hết bạn hãy xem xét kỹ hình 1 và chúng ta tìm hiểu một số khái niệm:

COMPILER: Chương trình cho vi điều khiển được chúng ta lập trình bằng một phần mềm trên máy tính, sau đó chúng ta sử dụng chính phần mềm này để biên dịch chương trình của chúng ta thành các định dạng mà có thể đổ vào vi điều khiển như các file có phần mở rộng **.hex** hay **.bin**. Phần mềm nói

trên gọi là Compiler (trình biên dịch). Có rất nhiều Compiler được dùng để lập trình cho vi điều khiển AT89C51, thông thường mỗi trình biên dịch cho phép bạn lập trình bằng một loại ngôn ngữ lập trình nhất định, bao gồm ngôn ngữ Assembly, C, Pascal hay Basic. Trong các loại ngôn ngữ lập trình kể trên. **Assembly** là ngôn ngữ cơ bản nhất cho vi điều khiển (tất nhiên là assembly cho vi điều khiển sẽ khác assembly cho máy tính), để lập trình được bằng assembly đòi hỏi bạn phải hiểu cấu trúc của vi điều khiển, vì vậy lập trình bằng ngôn ngữ này có khó khăn đôi chút, tuy nhiên tôi khuyên bạn nên tìm hiểu thật kỹ cấu trúc vi điều khiển và sử dụng ngôn ngữ này vì có như vậy bạn mới thật sự khám phá được những điều thú vị của vi điều khiển, ưu điểm của lập trình bằng assembly là chương trình của chúng ta sau khi biên dịch sẽ rất nhỏ gọn, tiết kiệm được bộ nhớ cho vi điều khiển. Bạn có thể sử dụng chương trình **8051IDE** hoặc **Keil** để lập trình bằng ngôn ngữ assembly (bạn sẽ tìm thấy các phần mềm này trong thư mục "Compiler" tôi gởi kèm). Sau khi bạn đã hiểu tường tận về vi điều khiển và bắt đầu xây dựng những ứng dụng phức tạp bạn có thể sử dụng các ngôn ngữ lập trình cấp cao như C (phần mềm Keil) hay Basic (Phần mềm **Bascom**)…

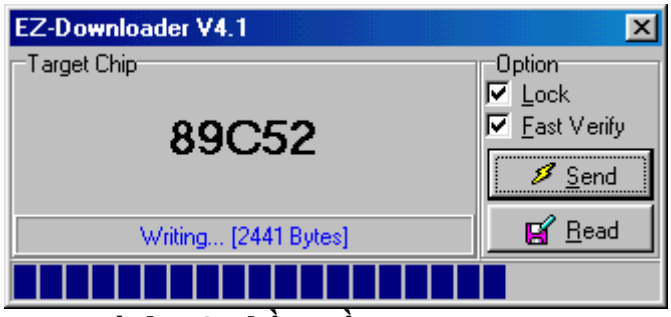
Ví dụ bạn lập trình bằng 8051IDE, bạn sẽ lưu chương trình bạn với tên **INTERRUPT.asm** trong đó phần mở rộng **.asm** là phần mợ rộng của các file assembly. Nếu bạn lập trình bằng Keil chương trình của bạn sẽ có tên là **INTERRUPT.c** còn lập trình bằng Bascom thì là **INTERRUPT.bas.** Tất các các file nói trên đều không thể đổ trực tiếp vào vi điều khiển mà chúng phải được biên dịch thành các file **hex** hay **bin**, ví dụ **INTERRUPT.hex, INTERRUPT.bin.** Các file này là các file tiêu chuẩn có thể được dùng để đổ vào chip.

- Tóm lại với một Compiler chúng ta có thể lập trình và biên dịch chương trình thành các file **hex** hay **bin**, đây là bước chuẩn bị đầu tiên.
- Bạn có thể tìm thấy các Compiler tôi đề cập trên trong thư mục Compiler tôi gởi kèm, nhưng chú ý các chương trình này chỉ là bản Demo hay evaluation nên sẽ có một số giới hạn nhất định.



Hình 1.1 một trình biên tập và biên dịch, Keil

PROGRAMER : Sauk hi đã có file **hex** cái tiếp theo chúng ta cần là 1 chương trình (gọi là Programer) đổ file hex của bạn lên chip thông qua một mạch nạp, tuỳ theo loại mạch nạp mà chương trình nạp cũng khác nhau. (có những trình biên dịch tích hợp cả trình nạp). Ở đây tôi giới thiệu cho bạn một chương trình nạp "ai cũng biết" đó là EZ V4.1. Chương trình nạp có chức năng đổ file hex vào chip của bạn.

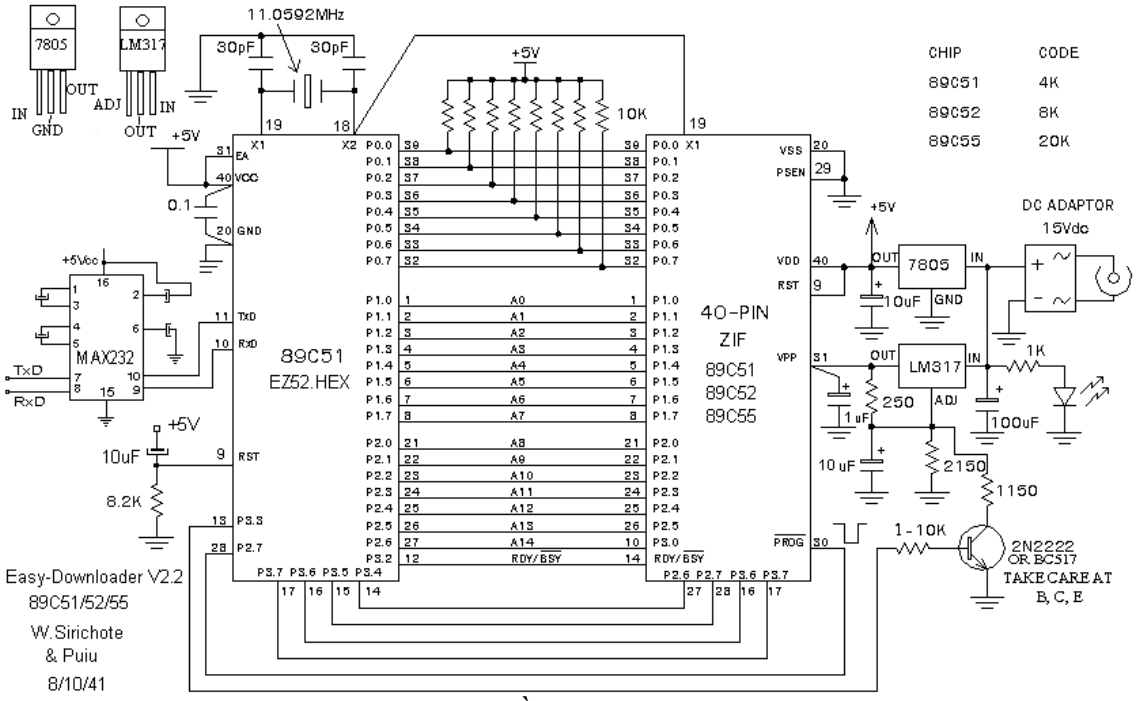


Hình 1.2 Phần mềm nạp EZ V4.1

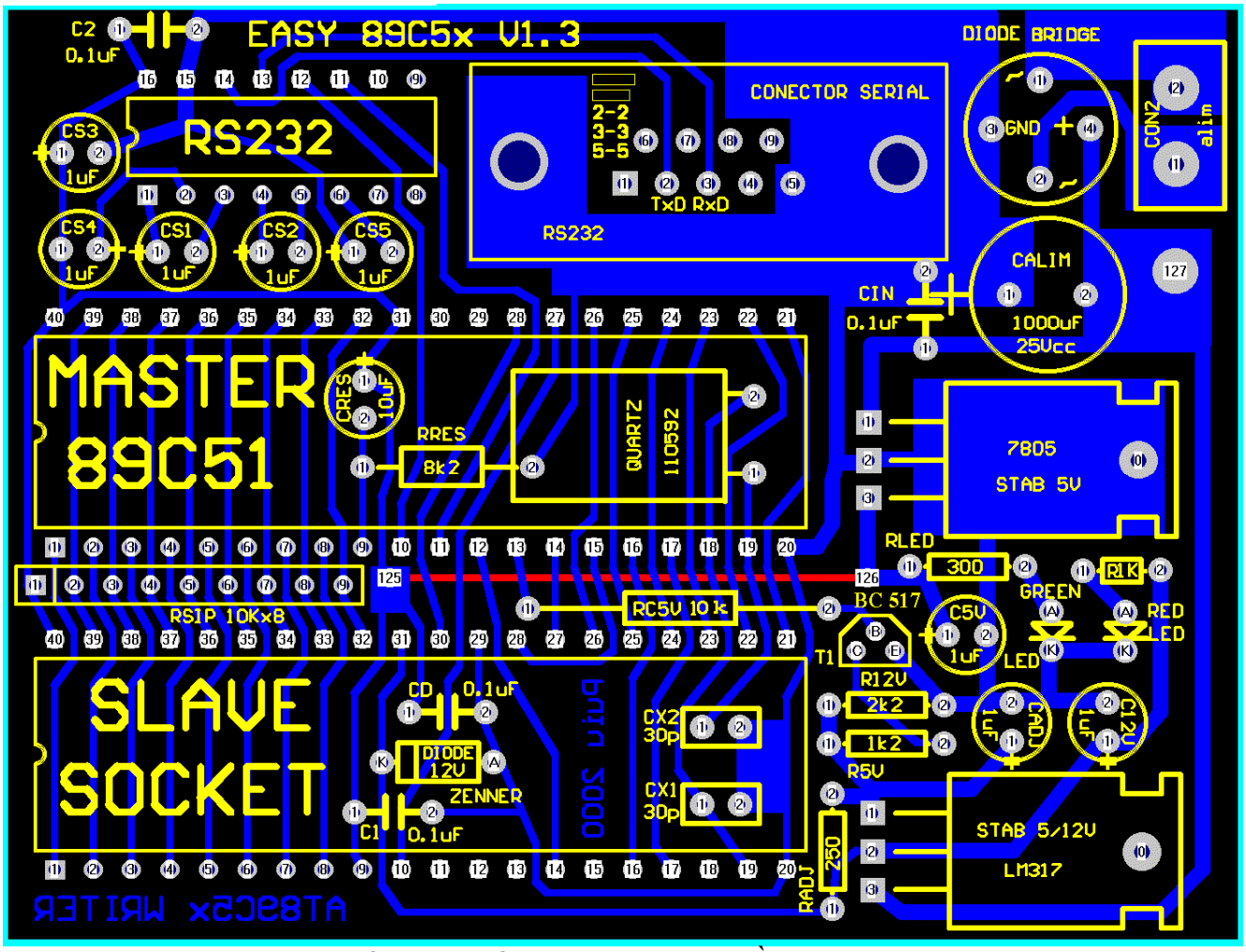
MẠCH NẠP: Bạn cũng cần một mạch điện gọi là mạch nạp để nạp chương trình vào chip, tất nhiên có nhiều loại mạch nạp khác nhau cho chip AT89C51. Ở đây tôi giới thiệu bạn mạch **Easy Download V2.2 (Thailand),** mạch này dùng với phần mềm nạp EZ V4.1 tôi giới thiệu trên.

Có thể nói mạch nạp là vấn đề khó khăn nhất cho những người mới tìm hiểu AT89C51, sau đây chúng ta sẽ bắt đầu đi làm mạch nạp:

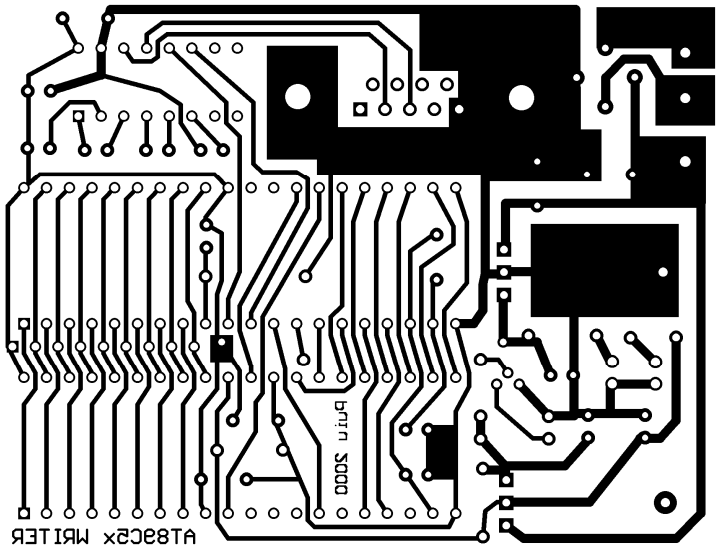
- Trước hết bạn hãy xem xét các mạch nguyên lý và mạch in :



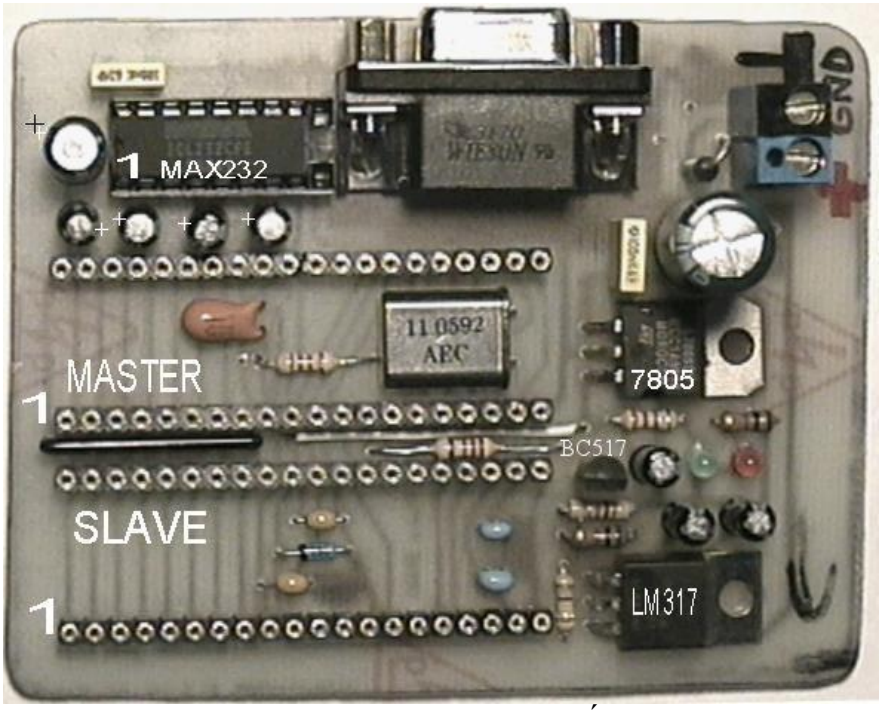
Hình 2 : Sơ đồ nguyên lý mạch nạp



Hình 3 : Toàn bộ mạch in (vẽ bằng Protel)



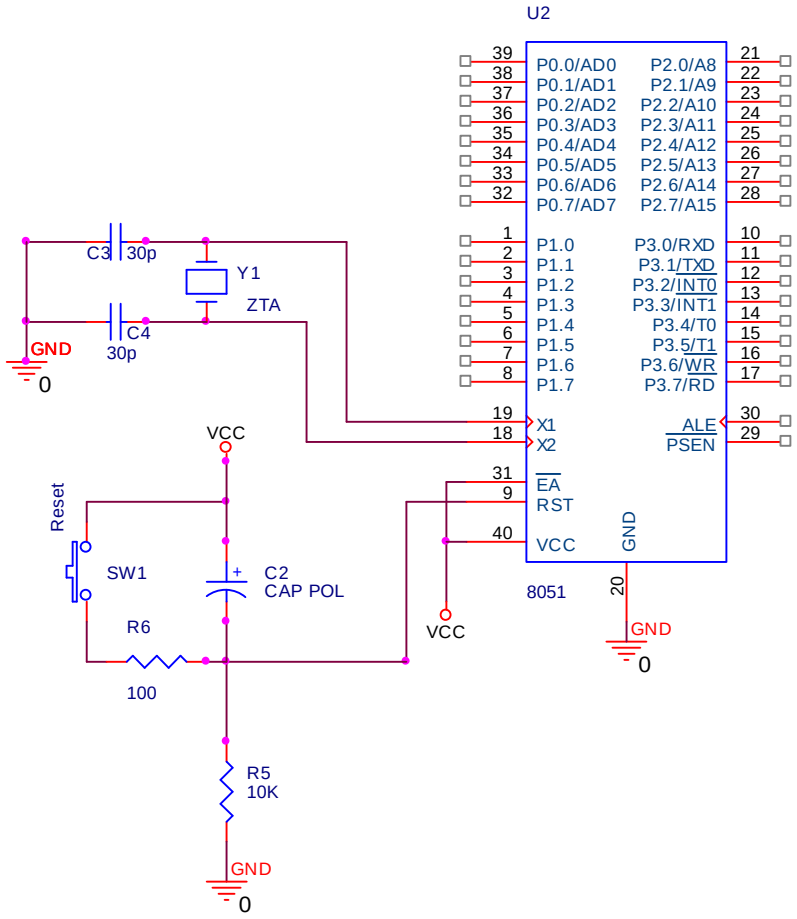
Hình 4: Mạch in



Hình 5: Mạch đã được chế tạo

- Bước 1 làm mạch in: có 2 cách để làm mạch in cho mạch nạp. Cách thứ nhất bạn tự chế tạo (giả sử bạn biết làm mạch in) từ hình 4 và cách thứ hai, cách bạn nên làm, là bạn hãy đến cửa làm làm mạch **KIM SƠN** mua lấy một bản mạch in đã được chế tạo sẵn, nhìn chung không đắt lắm, khoảng 10K/ mạch.
- Bước 2 mua linh kiện: bạn in mạch hình 2 để tham khảo (hoặc xem hình 3 cũng được), ra chợ nhật tảo mua lấy tất cả các linh kiện trong mạch, nhớ mua cả dây cab kết nối với máy tính nữa (tất cả khoảng hơn 100K đấy), nếu là lần đầu tiên "đi chợ" bạn nên đi theo 1 người đã biết thì hay hơn ! Nếu như bạn không biết phải mua thế nào tôi sẽ chỉ bạn sau.
- Bước 3 hàn mạch: sau khi có đủ mạch in, linh kiện và cả dụng cụ làm mạch bạn hãy hàn mạch dựa theo hình 4.
- Bước 4 sử dụng mạch: nói là bước sử dụng mạch nhưng thực tế bạn chưa sử dụng mạch được đâu, lý do bạn hãy để trên mạch nguyên lý có 1 chip 89C51 MASTER, chip này nằm cố định trên mạch nạp để điều khiển quá trình nạp, như vậy bản thân nó phải chứa một chương trình gọi là một Firmware. Như vậy bạn phải nhờ một mạch khác để nạp firmware vào chip Master của bạn, firmware cho chip này tôi cũng gởi kèm theo với tên **ez52.hex**
- Sau khi đã có mạch nạp, bạn có thể test mạch bằng cách kết nối mạch của bạn với cổng COM của máy tính thông qua cab nối (tất nhiên là đã có cấp nguồn cho mạch, nguồn nên lớn hơn 15V) , bạn hãy mở phần mềm EZ V4.1 lên, nếu bạn thấy các nút "**send**",**"read"** sáng cho phép bạn click vào thì mạch bạn làm đã thành công, nếu như thế bạn hãy tiếp tục nhấn vào nút "**send**", bạn tìm đến file "DELAY500.hex" tôi đính kèm, bạn sẽ thấy quá trình nạp file vào chip diễn ra, XIN CHÚC MỪNG.

MẠCH ỨNG DỤNG : tất nhiên để sử dụng con chip vừa được nạp chương trình và kiểm tra chương trình bạn cần một mạch ứng dụng. Tuỳ theo nhu cầu ứng dụng mà mạch ứng dụng sẽ khác nhau rất nhiều về mức độ phức tạp, tuy nhiên do bạn là người mới bắt đầu nên chúng ta chỉ nên sử dụng các **project board** để làm mạch thử, bạn ghim chip lên project board rồi gắn thêm các đèn led chẳng hạn để thử chương trình đã nạp. Về cơ bản để một chip sau khi được nạp chương trình chạy được thì mạch ứng dụng của bạn ít nhất phải được mắc như sau:



Hình 6: mạch ứng dụng cơ bản.

Trên đây tôi đã giới thiệu bạn cách sử dụng cơ bản nhất vi điều khiển AT89C51, chủ yếu là xây dựng phần cứng, vấn để phần mềm, lập trình cho vi điều khiển tôi sẽ giới thiệu ở bài khác hoặc các bạn có thể tìm hiểu ở các tài liệu hướng dẫn vi điều khiển AT89C51.

CHÚ Ý: Nếu như bạn đã biết chút ít về lập trình cho vi điều khiển 89C51 (bằng bất kỳ compiler nào) nhưng chưa co điều khiển xây dựng phần cứng như trên bạn có thể sử dụng phần mềm mô phỏng mạch điện "proteus", đây là một phần mềm cực hay để bạn thử các khả năng lập trình của mình, phần hướng dẫn mô phỏng bằng Proteus tôi cũng sẽ trình bày trong một tài liệu khác.