

MỤC LỤC

Lời Nói Đầu

Vấn đề 1:

KHÁI LƯỢC VỀ HỘ ATMEL AT80C51

----Trang 3

PHẦN 1: HƯỚNG DẪN LẮP RÁP CÁC LOẠI MẠCH NẠP VI XỬ LÝ

Vấn đề 2:

MẠCH NẠP :
AT89C51 - QUA CỔNG COM

----Trang 6

Vấn đề 3:

MẠCH NẠP :
AT89C2051 - QUA CỔNG COM

----Trang 16

Vấn đề 4:

MẠCH NẠP:
AT89C2051 QUA CỔNG MÁY IN

----Trang 26

Vấn đề 5:

MẠCH NẠP TỔNG HỢP:
AT89C51 & AT89C2051 QUA CỔNG MÁY IN

----Trang 32

PHẦN 2: HƯỚNG DẪN LÀM BO MẠCH IN CHẤT LƯỢNG CAO

Vấn đề 6:

HƯỚNG DẪN THIẾT KẾ MẠCH IN TRÊN MÁY TÍNH
BẰNG PHẦN MỀM : **Eagle 4.01**

----Trang 41

Vấn đề 7:

PHƯƠNG PHÁP IN BẢN VẼ LÊN BO ĐỒNG

----Trang 41

PHẦN 3: HƯỚNG DẪN SỬ DỤNG CÁC PHẦN MỀM VI XỬ LÝ

Vấn đề 8:

MỘT SỐ PHẦN MỀM LẬP TRÌNH / SOẠN THẢO **ASSEMBLY**

----Trang 44

Vấn đề 9:

PHẦN MỀM BIÊN DỊCH RA FILE HEX : **ASM / TASM**

----Trang 46

Vấn đề 10 :

CÁC PHẦN MỀM MÔ PHỎNG VI XỬ LÝ

----Trang 50

Vấn đề 11 :

HƯỚNG DẪN SỬ DỤNG **Ezdl4.0** CHO CÁC MẠCH NẠP

----Trang 52

PHẦN 4: CÁC MẠCH ĐIỆN VI XỬ LÝ - LÝ THÚ- ĐƠN GIẢN CHO NGƯỜI MỚI BẮT ĐẦU HỌC LẬP TRÌNH CÙNG VI XỬ LÝ

Vấn đề 12:

HÃY LÀM MẠCH ĐÈN CHỚP VỚI 89C51

----Trang 53

Vấn đề 13:

HÃY LÀM BỘ ĐIỀU KHIỂN NHÚNG ĐƠN GIẢN

----Trang 57

Vấn đề 14:

LÀM CÒI HỤ VỚI HAI MỨC ÂM ĐIỆN KHÁC NHAU

----Trang 64

Vấn đề 15 :

LÀM ĐỒNG HỒ BÁO GIỜ ĐIỆN TỬ

----Trang 67

LỜI KẾT.

Lời nói đầu

Ngày nay , nền công nghệ ứng dụng vi xử lý vào đời sống đang ngày một trở nên khá phổ biến và không ngừng được phát triển , hiện nay nó đã được phổ cập ở nhiều trường đại học , cao đẳng và cả các trường trung cấp.

Bên cạnh việc học lý thuyết ở trường , để các bạn có thể hiểu sâu hơn nữa về vi xử lý và khám phá các ứng dụng , các tính năng của nó bằng thực tế , tôi xin hân hạnh giới thiệu với các bạn quyển sách này “*Hướng dẫn lắp ráp các mạch nạp Vi Xử Lý* ” ngõ hầu sẽ giúp các bạn tự làm cho riêng mình một mạch nạp , để nạp chương trình mà bạn đã viết vào cho con vi xử lý để bạn có thể biết được chương trình mà bạn đã viết ấy - thực tế nó sẽ hoạt động như thế nào- đó chính là mục đích mà tôi đã viết quyển sách này cho các bạn.

Các bạn có thể tìm thấy các loại mạch nạp cho vi xử lý (đúng ra phải gọi là vi điều khiển) rất đơn giản , dễ lắp ráp , linh kiện dễ tìm kiếm , cho nên xác suất thành công rất cao nên ai ai cũng có thể tự làm lấy cho riêng mình một mạch nạp để phục vụ cho việc học , nghiên cứu và nhất là để làm các đồ án , luận văn liên quan đến vi xử lý !

Ngoài ra tôi còn bổ sung thêm phần hướng dẫn làm bo mạch in chất lượng rất cao để các bạn tự làm ở nhà , tiết kiệm rất nhiều chi phí rất thích hợp cho sinh viên nhưng chất lượng như đã nói là rất cao . Đồng thời tôi cũng có hướng dẫn cho bạn cách sử dụng các phần mềm về vi xử lý phục vụ cho việc lập trình để bạn không phải mất nhiều thời gian tìm kiếm-và sử dụng các phần mềm này !

Sau cùng là các ví dụ cụ thể sẽ giúp bạn bước đầu làm quen cùng vi xử lý , các ứng dụng này dành riêng cho người mới bắt đầu học lập trình vi xử lý ! Nếu các bạn nào đã thành thạo rồi thì xin hãy đọc quyển 2 “**Hướng dẫn giao tiếp vi xử lý với máy tính**” ở đây bạn sẽ biết cách giao tiếp qua lại giữa chúng . đồng thời tôi cũng có nhiều ví dụ thiết thực như “**Máy đo nhiệt độ bằng Led , máy đo nhiệt độ giao tiếp máy tính , bảng đèn chữ chạy đơn giản , bảng đèn chữ chạy giao tiếp máy tính , đo điện áp qua máy tính** “ và một số mạch điện ứng dụng vi xử lý khác sẽ làm cho bạn cảm thấy thích thú hơn khi học về vi xử lý

Tập sách này được viết không ngoài mục đích góp phần làm phát triển việc ứng dụng vi xử lý ngày một hơn , đồng thời nó cũng sẽ giúp bạn tiết kiệm rất nhiều chi phí , hiện tại giá của mỗi mạch nạp Vi xử lý theo tôi được biết ở Nhật Tảo bán khoảng 250.000đ , ngoài ra cũng có một số bạn khác chỉ bán giá 130.000đ tuy nhiên mạch nạp này lại quá tiết kiệm vì chỉ ráp toàn các linh kiện rẻ tiền . Nhưng nếu bạn tự ráp không những chỉ tiết kiệm khá nhiều tiền của , **chỉ cần bạn tốn khoảng 70.000đ là bạn có ngay cho riêng mình một mạch nạp chất lượng vô cùng** vì linh kiện là do chính tay bạn mua lấy . Ở đây vấn đề không phải là chuyện tiền nong , song vấn đề là mạch nạp ấy là do chính bạn đã lắp ráp nó , do đó bạn sẽ cảm thấy thích thú hơn ,hài lòng hơn với sản phẩm của mình .

Cuối cùng tôi xin chân thành cảm ơn các bạn đang học cùng tôi đã động viên khích lệ tinh thần , cùng có lời tri ơn sâu sắc đến các thầy trong trường đã giúp đỡ cho tôi rất nhiều để quyển sách này hoàn thành đúng lúc các bạn đang học về vi xử lý . Trong điều kiện ấy , việc thiếu sót sẽ là điều không thể tránh khỏi , cho nên sự góp ý từ phía các bạn độc giả cũng như những lời góp ý của quý thầy cô sẽ là niềm vinh hạnh cho tôi , tôi xin lắng nghe , sửa chữa để lần sau có dịp thì quyển sách này sẽ hoàn thiện hơn , ngõ hầu sẽ giúp ích cho các bạn trong việc học tập cùng việc nghiên cứu đạt được nhiều lợi lạc !

Nhân đây tôi xin hân hạnh mời bạn đón đọc tập 2:” Hướng dẫn giao tiếp vi xử lý với máy tính ” bằng Visual Basic mà tôi sắp viết xong !

Một lần nữa tôi xin cảm ơn các bạn đã đọc quyển sách này !

Lê Duy Phi

Vấn đề 1:

KHÁI LƯỢC VỀ AT89C51 & AT89C2051



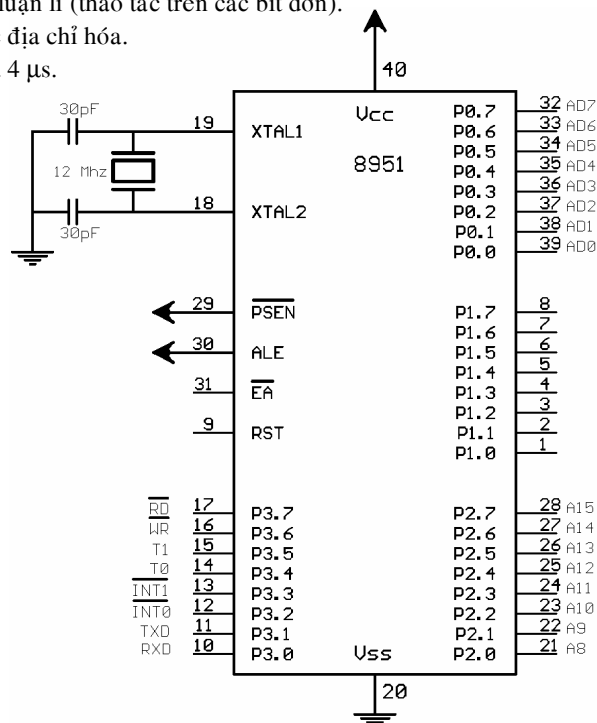
Tước khi đi vào lắp ráp bo mạch, tôi xin các bạn dành ít thời gian xem qua sơ đồ chân và một số các đặc điểm cũng như một số các đặt tính của đối tượng mà ta sẽ làm việc rất nhiều với nó đó là con vi điều khiển AT89C51 và AT89C2051.

I. MÔ TẢ CẤU TRÚC PHẦN CỨNG CỦA VI ĐIỀU KHIỂN 8951.

1/ Giới thiệu họ MCS51:

MCS51 là một họ IC vi điều khiển (Microcontroller) do hãng Intel sản xuất. Các IC tiêu biểu cho họ MCS51 là **8051** và **8031**. Đặc biệt, vi điều khiển **8951** được sản xuất gần đây mang các đặc điểm sau:

- § 4 Kbytes EEPROM.
- § 128 bytes RAM.
- § 4 ports I/O (Input/Output).
- § 2 bộ định thời (timer) 16 bits.
- § Giao tiếp nối tiếp.
- § 64 Kbytes không gian bộ nhớ chương trình mở rộng.
- § 64 Kbytes không gian bộ nhớ dữ liệu mở rộng.
- § Một bộ xử lý luận lý (thao tác trên các bit đơn).
- § 210 bits được địa chỉ hóa.
- § Bộ nhân chia 4 μ s.



2. HỆ THỐNG GIAO TIẾP PORT:

a/ **Port 0:** Port 0 là một port hai chức năng trên các chân 32 – 39.

Hãy nhớ rằng : Trên các chân này chưa có điện trở kéo dương , do đó khi cần chúng ta phải nhớ đến đặc điểm này.

b/ **Port 1:** Port 1 là một port I/O trên các chân 1 – 8.

c/ **Port 2:** Port 2 là một port công dụng kép trên các chân 21 – 28 .

d/ **Port 3:** Port 3 là một port công dụng kép trên các chân 10 – 17. Các chân của port này có nhiều chức năng, các công dụng chuyển đổi có liên hệ với các đặc tính đặc biệt của 8951 như ở bảng sau:

	Tên	Chức năng chuyển đổi
P3.0	RXD	Dữ liệu nhận cho port nối tiếp
P3.1	TXD	Dữ liệu phát cho port nối tiếp
P3.2	INT0\	Ngắt 0 bên ngoài
P3.3	INT1\	Ngắt 1 bên ngoài
P3.4	T0	Ngõ vào của Timer/counter 0
P3.5	T1	Ngõ vào của Timer/counter 1
P3.6	WR\	Xung ghi bộ nhớ dữ liệu ngoài
P3.7	RD\	Xung đọc bộ nhớ dữ liệu ngoài

3 - CÁC TÍN HIỆU ĐIỀU KHIỂN:

Chip AT8951 có các tín hiệu điều khiển cần phải lưu ý sau:

A- Chân EA\ (External Access):

Tín hiệu vào **EA** trên chân 31 thường được mắc lên mức cao (+5v) hoặc mức thấp (GND).

Nếu ở mức cao, **8951** thi hành chương trình từ ROM nội trong khoảng địa chỉ thấp (4K hoặc tối đa 8k đối với AT89C52).

Nếu ở mức thấp, chương trình được thi hành từ bộ nhớ ngoài mở rộng (Tối đa đến 64K).

Ngoài ra Người ta còn dùng EA\ làm chân cấp điện áp 12V khi lập trình cho EEPROM trong 8051.

B- CHÂN PSEN\ (PROGRAM STORE ENABLE):

PSEN\ là tín hiệu ra trên chân 29. Nó là tín hiệu điều khiển cho phép bộ nhớ chương trình mở rộng, **PSEN** thường được nối đến chân OE (Output Enable) của một EPROM hoặc ROM để cho phép đọc các byte mã lệnh.

Hãy nhớ rằng : Bình thường chân PSEN\ sẽ được thả trống (No Connect). Chỉ khi nào chân EA\ ở mức thấp thì lúc đó:

PSEN sẽ ở mức thấp trong thời gian lấy lệnh. Các mã nhị phân của chương trình được đọc từ EPROM qua bus dữ liệu và được chốt vào thanh ghi lệnh của **8951** để giải mã lệnh.

PSEN sẽ ở mức thụ động (mức cao) nếu thi hành chương trình trong ROM nội (**8951**)

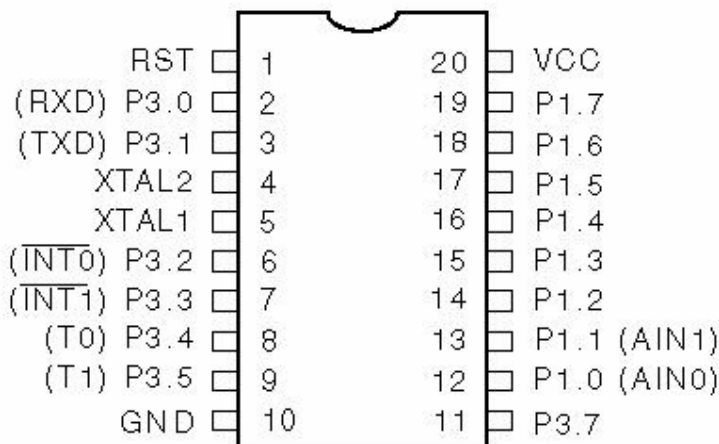
C- CÁC CHÂN NGUỒN:

AT8951 hoạt động với nguồn đơn +5V. Vcc được nối vào chân 40 và Vss (GND) được nối vào chân 20.

II. MÔ TẢ CẤU TRÚC PHẦN CỨNG CỦA VI ĐIỀU KHIỂN AT89C2051.

AT89C2051 là một vi điều khiển thu gọn chỉ có 20 chân, với đầy đủ các tính năng như AT89C51. Duy chỉ khác một điểm cần lưu ý đó là Flash ROM của nó chỉ có 2K mà thôi và một vài Port không được hỗ trợ đủ, do đó tùy theo mục đích mà ta sẽ cần đến nó.

Hình : Sơ đồ chân AT89C2051



MỘT SỐ CÁC ĐẶC TÍNH CỦA AT89C2051 LÀ:

- ~ 2 Kbytes of Flash (Có nạp vào hoặc xóa đi 1000 lần)
- ~ 128 bytes of RAM
- ~ 15 I/O lines (15 đường xuất/ nhập)
- ~ two 16-bit timer/counters (hai bộ định thì/ đếm 16 bit)
- ~ five vector, two-level interrupt architecture (có 5 vectơ ngắt-2 mức)
- ~ full duplex serial port (có cổng nối tiếp)
- ~ precision analog comparator
- ~ on chip oscillator and clock circuitry (có mạch dao động và tạo xung bên trong chip)

Hãy nhớ rằng: Mức điện áp hoạt động thấp (2,7V ÷ 6V, chúng ta thường dùng nó ở 5V)

Tần số dao động thạch anh tối đa 24Mhz (thường ta dùng 11.0592Mhz để hạn chế lỗi đến mức nhỏ nhất trong quá trình nó làm việc)



Vấn đề 2:

MẠCH NẠP : AT89C51 - QUA CỔNG COM



Hãy làm riêng cho bạn một nạp chương trình- để nạp file HEX cho các vi điều khiển 89C51 , 89C52 , 89C53 và 89C55 .Ở đây có sẵn bo mạch in , sơ đồ và cả phần mềm cùng hướng dẫn hết sức dễ dàng để làm và sử dụng nó.

I- ĐÔI LỜI GIỚI THIỆU :

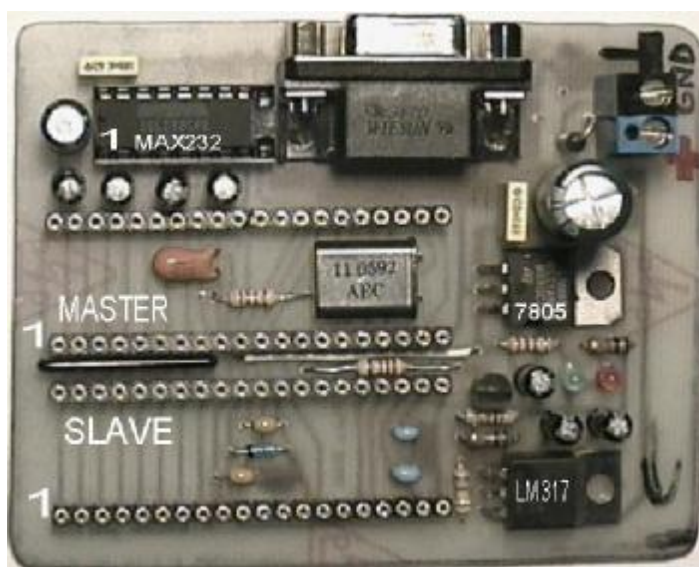
Nếu bạn muốn vừa học Vi Xử Lý đồng thời cũng muốn khám phá nó qua các ứng dụng cụ thể, qua các dự án thực tế để phát triển 89C51, 89C52...với ngôn ngữ lập trình Asembly thì “ **Mạch Nạp AT89C51** ” chính là câu trả lời . Nó rất rẻ và mọi người đều có thể tự làm được cho riêng mình một nạp .

Mạch nạp này không những chỉ nạp được cho AT89C51(4k) mà còn nạp được cho cả 89C52(8k) và 89C55(20k) .

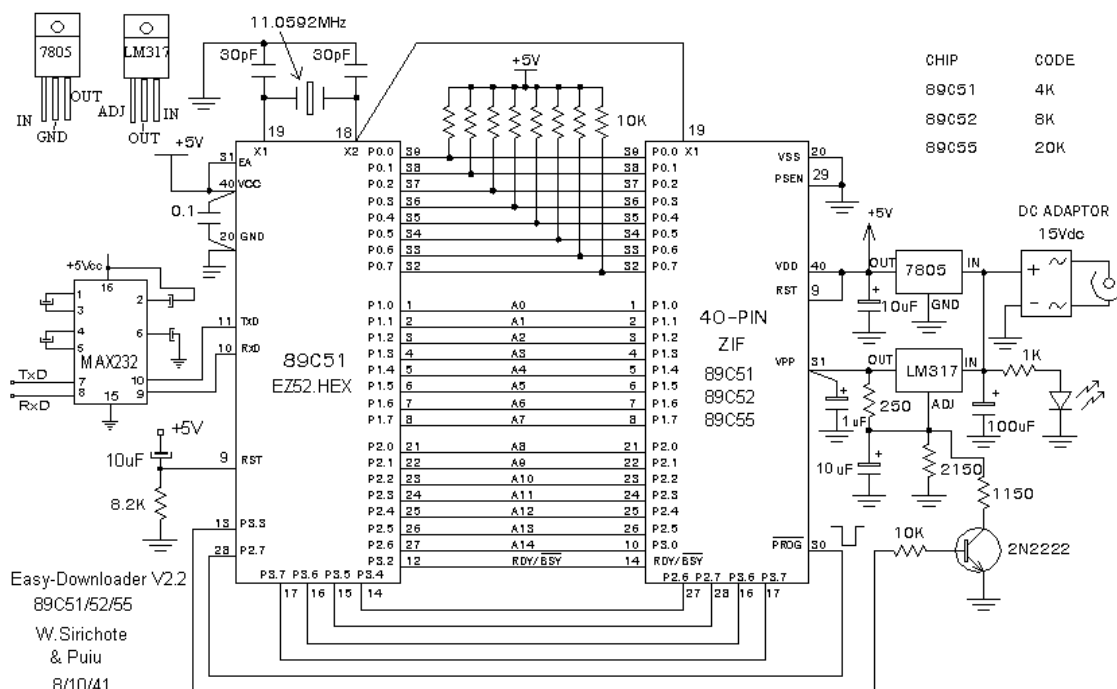
Sau khi bạn lập trình với Asembly hoặc C xong ,rồi bạn hãy dịch nó ra file HEX (nếu bạn chưa biết cách dịch thì bạn hãy xem mục hướng dẫn các phần mềm Vi Xử Lý ở đằng sau) sau đó bạn hãy nạp nó vào cho con Chip bằng cách dùng “Mạch Nạp AT89C51”.

II- PHẦN CỨNG : SƠ ĐỒ VÀ BO MẠCH IN

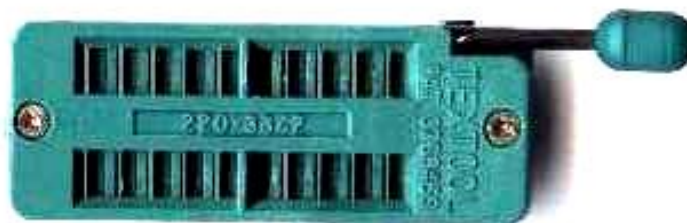
1). HÌNH ẢNH MẠCH NẠP HOÀN CHỈNH :



Hình 1

2) SƠ ĐỒ MẠCH NẠP AT89C51 :**Hình 2:** Đây chính là sơ đồ của **Mạch Nạp AT89C51**.**TRÊN SƠ ĐỒ BẠN HÃY LƯU Ý HAI ĐIỂM SAU :**

- Kí hiệu IC ghi **89C51 EZ52.HEX** gọi là con **Chip chính (ChipMaster)**
- Kí hiệu IC ghi **40 -PIN ZIF 89c51/ 89c52/ 89c55** đây là **Socket** hay là chân để cắm IC (còn gọi là **đế cắm IC 40 Chân**) , hay gọi là **con chip phụ (Chip Slave)**



Đây là hình dáng thật của Socket (để để gắn IC vào) tuy nhiên đây là cái Sôket 20 chân thôi ,còn loại 40 cũng giống y hệt như vậy nhưng nó dài và to hơn thôi.

Hãy nhớ rằng :

Đầu tiên bạn cần phải nạp file EZ52.HEX vào cho con *ChipMaster* . Nhưng khó nổi là lấy đâu ra mạch nạp để nạp file EZ52.Hex này cho con *Chip Master* ?? Bạn đừng lo , bạn có thể mua con *Chip Master* –đã nạp sẵn này tại quầy photo B1 tại trường đại học Bách Khoa , hoặc bạn cũng có thể mua đĩa mềm 1,4Mb tôi có chép sẵn file EZ52.HEX trong đó luôn rồi , bạn có thể đem ra các tiệm điện ở chợ Điện Tử Nhật Tảo , ở đó họ có sẵn máy nạp họ sẽ nạp cho bạn .với giá cho mỗi lần nạp là 2.000đ

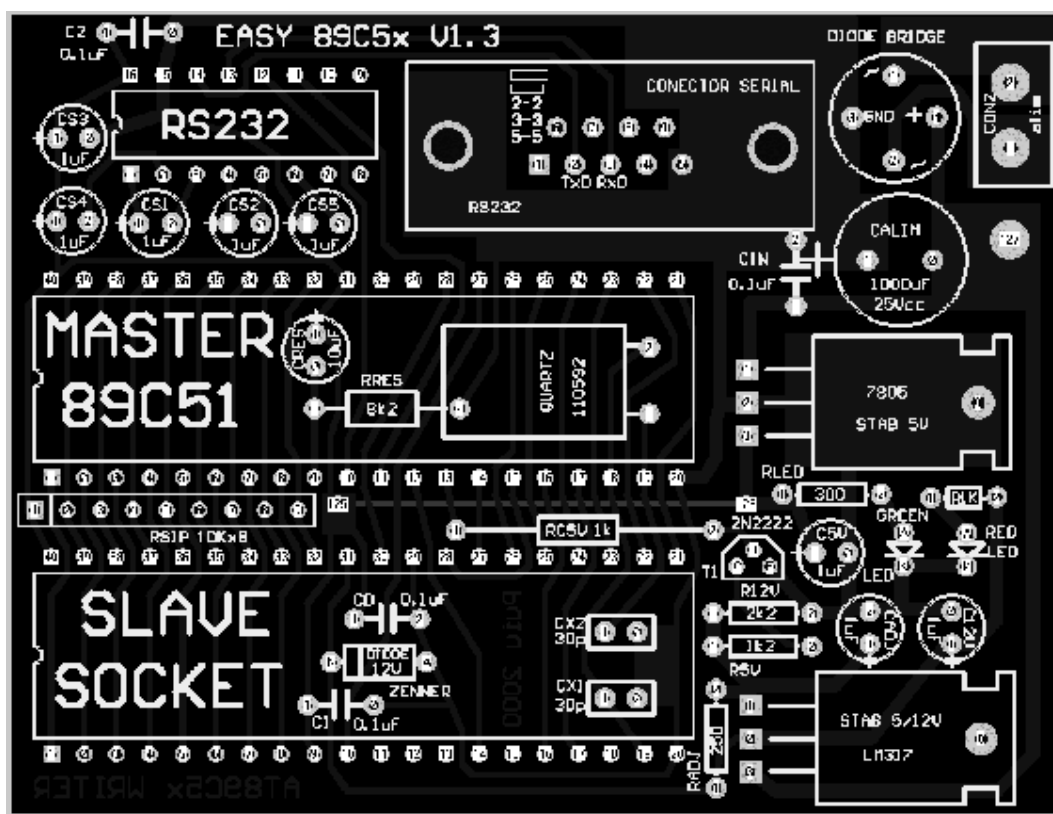
Và tôi cũng xin nói thêm là con Chip Master cũng là một con AT89C51. Còn cái Socket là nơi để bạn đặt con Chip 89C51 Slave - mà bạn cần nạp bất kỳ file .HEX nào đó vào cho nó.

3) NGUỒN ĐIỆN CẤP CHO MẠCH NẠP:

Hãy lưu ý :Tại chân Vpp của Mạch nạp có hai mức điện áp khác nhau là 5V và 12V được điều khiển bởi chân P3.3 (chính là đầu ra của LM317). Bình thường thì đầu ra của LM317 chỉ có 5V thôi, nhưng khi bắt đầu nạp chương trình thì đầu ra của nó lên tới 12V. Tôi đã lắp ráp, kiểm tra và sử dụng –mạch này rất nhiều nhưng chưa hề thấy sự cố nào quá nghiêm trọng cả.

4) BỘ MẠCH IN

Hiện nay trên chợ Điện Tử Nhật Tạo bán rất nhiều loại Bo mạch nạp này, và hiện đã trở nên quá quen thuộc, quen thuộc đến nỗi bạn chỉ cần nói bán cho bạn “Mạch in –của mạch nạp Vi Xử Lý” thì người sẽ đưa cho bạn ngay một bản mạch như sau với giá chỉ có 10.000 đ mà thôi :

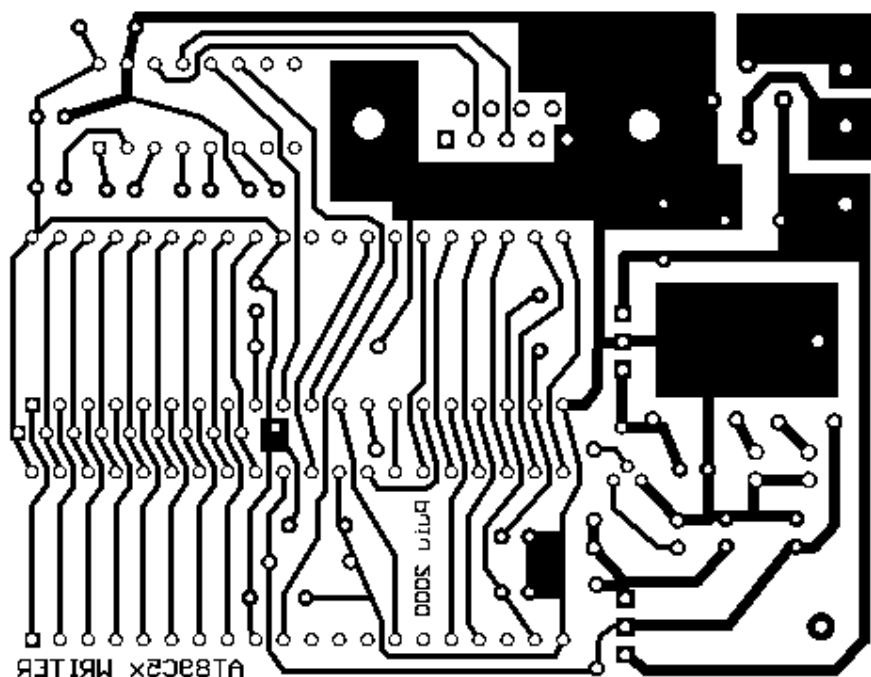


Hình 3

Đây là một loại mạch nạp đang phổ biến, rất được nhiều người sử dụng và đang thịnh hành trên Chợ Nhật Tảo. Hình này là “Mạch In Hướng Dẫn Lắp Ráp Linh Kiện” lên bản mạch đồng.

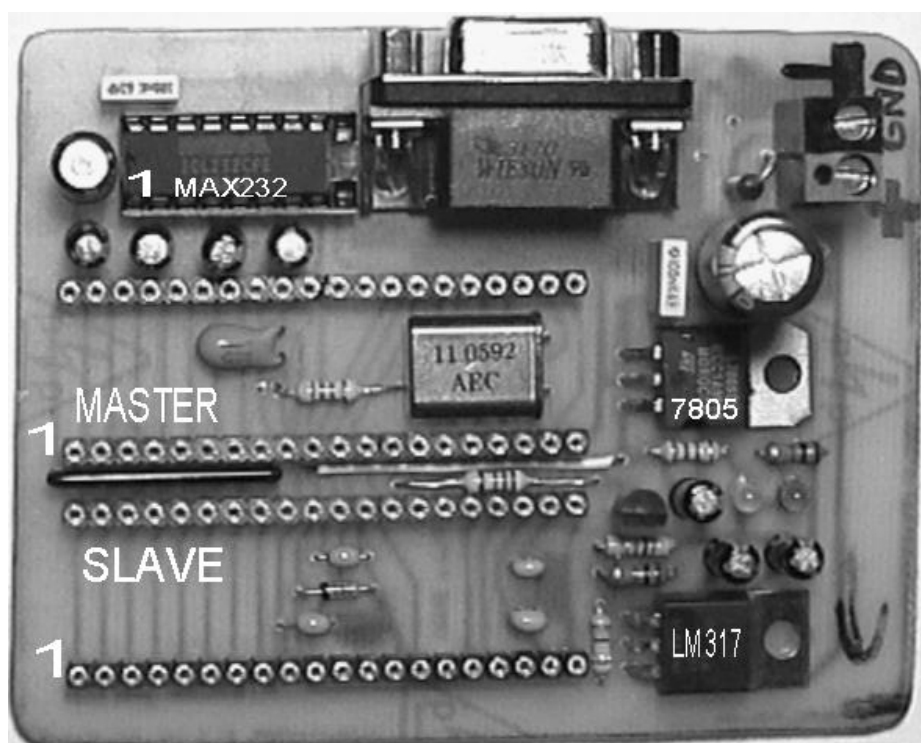
Và tôi cũng đã chép luôn file chứa các mạch in ở hình 2 trong đĩa mềm với dung lượng 36,4kb để bạn có thể xem qua để có thể tự làm lấy bo mạch cho mình.

Và sau đây là bo mạch đồng của nó , bạn có thể cắt hình sau để tự làm cho mình một mạch đồng rất đẹp , giống y như thế theo sự hướng dẫn ở phần 2 của quyển sách này.



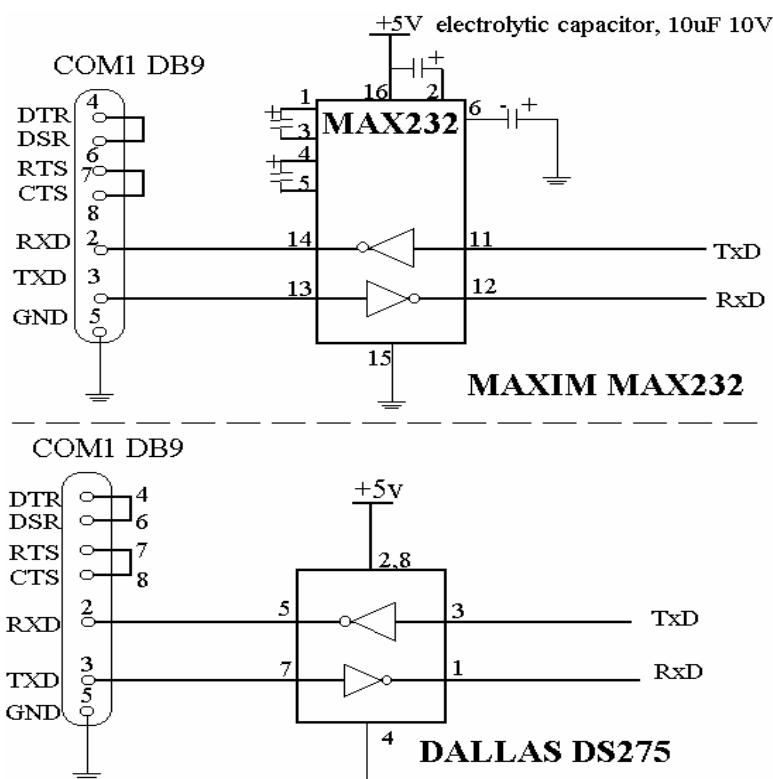
Đây là Mạch đồng của mạch nạp vi xử lý đang thịnh hành trên chợ Nhật Tảo hiện nay .Giá một mạch chỉ có 10.000 đ (Bạn có thể cắt hình này ra để làm mạch đồng hoặc mua riêng ở quây Photo B1 / dh.BK)

Và đây là những gì mà bạn có thể thấy sau khi bạn đã lắp ráp hoàn chỉnh



Hãy nhớ rằng : MAX 232 – 16 chân còn có thể được thay thế bằng con chip DS257 hết sức đơn giản chỉ có 8 chân thôi . Khi đó bo mạch in của bạn sẽ nhỏ gọn hơn .

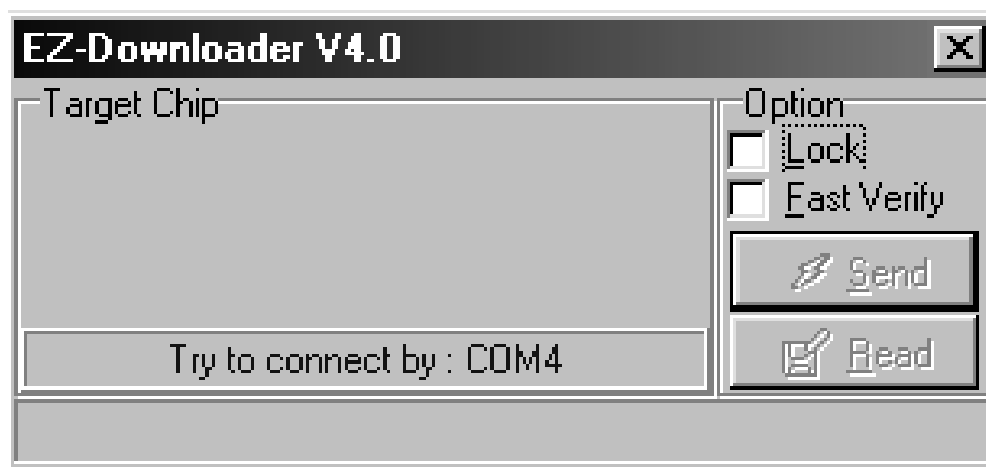
Sơ đồ thay thế như sau:



IV- PHẦN MỀM –CHO MẠCH NẠP AT89C51

Mạch nạp AT89C51 này giao tiếp với máy tính thông qua phần mềm EZ4.0. Hiện thời đây là phần mềm mới nhất và hoàn toàn miễn phí chạy trên nền Windows rất đẹp và rất dễ dàng sử dụng .

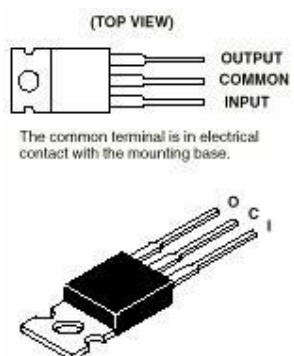
Giao diện của nó như sau:



Nếu bạn chưa rõ cách dùng nó thì xin bạn hãy xem phần 3 “ Hướng dẫn sử dụng một số phần mềm Vi Xử Lý thì sẽ rõ .

V- MỘT SỐ LƯU Ý VỀ CÁC LINH KIỆN CẦN DÙNG:

1- CÁCH XÁC ĐỊNH CÁC CHÂN ĐIỆN ÁP VÀO RA CỦA 7805



Output (O):= Chân điện áp ra 5V

Command(C):= Chân này hàn xuống Mass (GND)

Input (I):= Điện áp vào 9V à 12V

Hãy nhớ rằng: Chân Command rất quan trọng, chúng ta cần hàn nó xuống mass(GND) một cách hết sức chắc chắn nếu như chân này hàn không khéo có làm cho điện áp ở ngõ ra tăng vọt lên rất cao gần bằng điện áp ngõ vào (9-12V) sẽ làm hỏng các chip.

2- CÁCH XÁC ĐỊNH CHÂN B-C-E CỦA TRANSISTOR

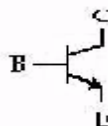
Lưu ý: Các Transistor BC239-được thay bằng C1815

Và Transistor BC328 được thay bằng A1015

Sau đây là cách xác định chân B_C_E cho transistor A1015 và C1815



Hình dáng thực tế

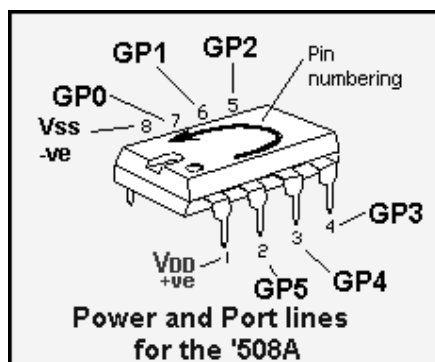


Kí hiệu của C 1815/ A 1015

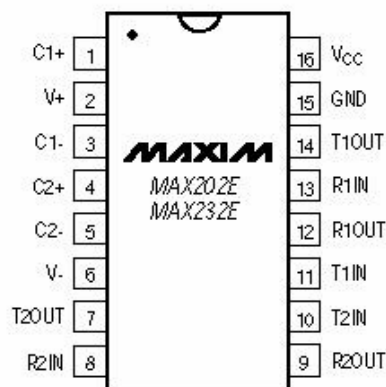
3- CÁCH XÁC ĐỊNH CHÂN CỦA IC

Hãy nhớ rằng: Hầu hết trên các IC đều có một **dấu chấm** ở một góc IC hay **một vạt bán nguyệt** ở trên một cạnh của IC, và quy tắc đếm số thứ tự chân của IC luôn theo chiều ngược với kim đồng hồ.

Chân số 1 của IC chính là chân đầu tiên nằm trên cạnh bên trái của dấu Chấm hay Vạt bán nguyệt. **Với quy ước là vạt bán nguyệt hay dấu chấm phải được đặt bên tay trái!**



SƠ ĐỒ CHÂN CỦA MAX232



4- CÁCH ĐỌC GIÁ TRỊ ĐIỆN TRỞ**CÁCH TÌM GIÁ TRỊ ĐIỆN TRỞ
THÔNG QUA CÁC VẠCH MÀU**

Để tính toán giá trị của điện trở thông qua các mã vạch màu (color coded stripes) trên điện trở, thường theo các bước sau:

Bước 1 : Đặt con điện trở sao cho Vạch màu Vàng kim hay nhũ bạc của điện trở -phải ở góc bên phía tay phải của bạn.

Bước 2 : Bạn hãy nhìn kỹ 2 vạch màu đầu tiên ở phía bên trái (the first two stripes on the left end). Tiếp theo bạn hãy chuyển đổi các màu mà bạn đọc được ra thành số theo bảng chuyển đổi cho sẵn ở dưới đây.

Bước 3 : Xong, bạn hãy nhìn tiếp vạch thứ 3. Vạch này chính là hệ số nhân.

Bạn hãy chuyển đổi vạch màu thứ 3 này ra giá trị theo bảng cho sẵn phía sau.

Bước 4 : Đến đây bạn chỉ việc lấy 2 con số ở bước 2 nhân với con số ở bước 3 là bạn đã tìm được giá trị của điện trở. Giá trị của điện trở mà bạn vừa tìm sẽ mang **đơn vị là ohms**

Vạch màu nhũ kim hay nhũ bạc (tức là vạch thứ 4 tính từ trái sang) sẽ cho chúng ta biết sai số của nó so với giá trị ghi trên điện trở đó. Theo tôi thì chúng ta không nên phải bận tâm đối với điều này, có nghĩa là chúng ta không cần biết nó trong khi thực hành.

Hãy nhớ rằng: Đối với điện trở chúng ta cần hết sức lưu ý đến một thông số vô cùng quan trọng đó là công suất chịu đựng của nó. Thông thường công suất điện trở bán ở thị trường sẽ có các mức sau : 1/4 Watt, 0.5 Watt, 1W, 5W, 10W..... Nhưng thông dụng nhất vẫn là điện trở 1/4 Watt- **cho nên khi bạn ra tiệm mua phải nói rõ điện trở cần mua là bao nhiêu Watt- thế mới là dân điện chuyên nghiệp !**. Giá hiện thời 100 con khoảng 2.500 ĐVN

BẢNG MÃ MÀU ĐIỆN TRỞ.**(Với vạch nhũ kim hay bạc nằm bên phải)**

Màu	Vạch 1	Vạch 2	Vạch 3	Vạch 4
Đen	0	0	x1	
Nâu	1	1	x10	
Đỏ	2	2	x100	
Cam	3	3	x1,000	
Vàng	4	4	x10,000	
Lục (Green)	5	5	x100,000	
Lam (Blue)	6	6	x1,000,000	
Tím	7	7		
Xám	8	8		
Trắng	9	9		
Nhũ Kim				5%
Bạc				10%

Ví dụ:

Cho một điện trở có các vạch màu tính từ trái qua như sau :

Xanh da trời(lam-blue) ,Xám , Đỏ, Nhũ vàng .Hãy tìm giá trị điện trở của nó.

Bước 1.: Vạch màu nhũ vàng –phải được đặt bên phải , rồi đến Bước2

Bước 2: Vạch đầu tiên là màu xanh da trời(blue) –tra bảng thì thấy nó tương ứng với số 6. Vạch thứ 2 là màu xám và cũng làm tương tự ta có được số 8. Như vậy 2 con số đầu tiên của điện trở có giá trị là 68.

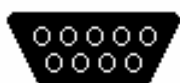
Bước3: Vạch thứ 3 là màu đỏ -có nghĩa là x 100.

Bước4: Giá trị của điện trở tìm được sẽ là : $68 \times 100 = 6800 \text{ ohms}$

(6800 ohms = 6.8 kohms).

Vạch màu nhũ đồng có nghĩa là giá trị chính xác của điện trở sẽ sai khác 5% so với thực tế . Giá trị của nó có thể sẽ nằm giữa 6,460 Kohms and 7,140 Kohms. (Do sai số 5% của $6,800 = 0.05 \times 6,800 = 340$)

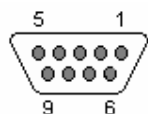
5) SƠ ĐỒ CHÂN CỦA CỔNG COM DB9:



(9p D-SUB male at PC (DTE))

Hãy nhớ rằng : Cổng com gắn trên máy tính là cổng “Đực”(male)

Để nhận dạng cổng COM đực , bạn chỉ cần nhìn vào thì bạn sẽ thấy có các “ chấu “ ở bên trong



(9p D-SUB female at cable / mouse (DCE))

Hãy nhớ rằng : Cổng mà ta cần dùng để lắp ráp mạch nạp là cổng COM này , nó là cổng “cái “.

Để nhận ra cổng COM cái , bạn chỉ cần nhìn vào thì sẽ thấy chỉ toàn là” các “ lỗ”.

Cần lưu ý

: Rất nhiều bạn mua lộn giữa Cổng đực và cổng Cái , cho nên khi đi mua bạn cần nói rõ là mua cổng nào , và cũng nhớ là ở trong máy tính của chúng ta đã sẵn có cổng đực rồi . Giá mỗi cổng COM -DB9 khoảng 5.000đ mà thôi .Nhắc lại là phải hết sức lưu ý kiểm tra cho kỹ trước khi hàn cổng COM lên mạch nạp , và trên mạch nạp luôn là cổng Cái

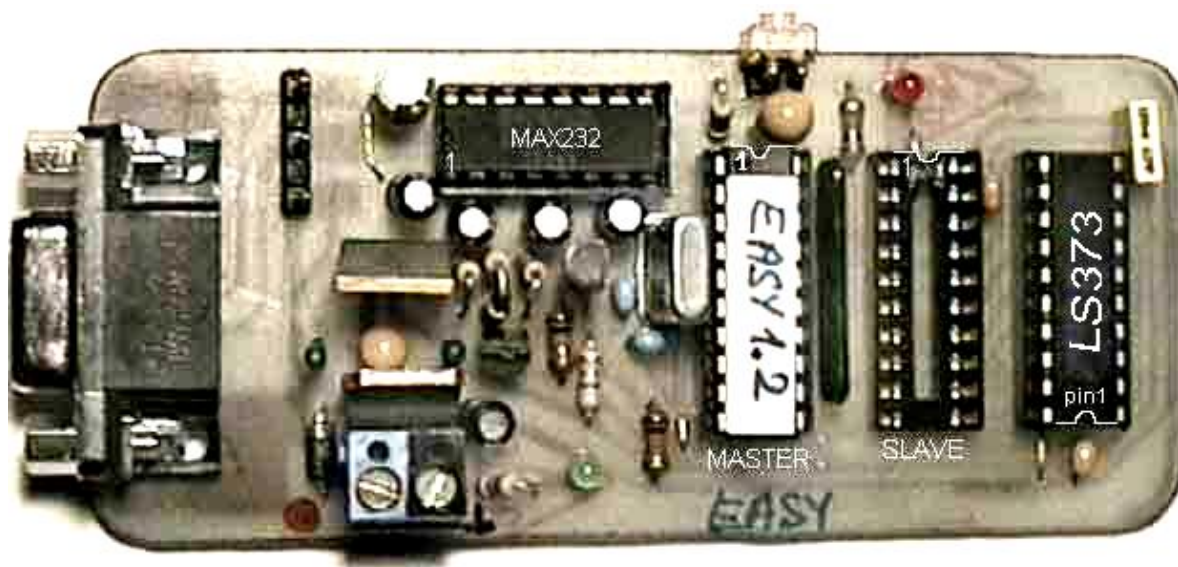
VI- MỘT SỐ CÁC LƯU Ý :

- Biến thế điện có thể dùng loại 450mA cho đến 1Ampe . Với ngõ ra AC (xoay chiều) tối thiểu là 12V –AC . Tốt nhất là nên mua biến thế 1 Ampe có ngõ ra 15 V giá khoảng 13.000đ .
- IC ổn áp 7805 bạn không nên mua loại có chữ màu trắng –loại này đa số là của Trung Hoa sản xuất nên không khi nào đạt đúng 5V như con số đã ghi cả, bạn nên mua loại chữ 7805 chìm , loại này chịu nhiệt cao và rất chuẩn volt . Giá của nó khoảng 3000đ , nếu mua hàng cũ thì mỗi con chỉ hết 500 đ rất rẻ(bán rất nhiều ở chợ Nhật Tảo) .!
- Tương tự cho LM317 cũng nên mua loại chữ chìm và giá cả cũng như 7805 .
- Transistor 2N2222 bạn có thể thay bằng C1815 giá 300đ/c
- Còn các linh kiện đều có bán ở chợ Nhật Tảo

Trong quá trình lắp ráp nếu bạn có gặp trục trặc hay bất kỳ một sự cố nào sau khi chính bản thân bạn đã cố sửa nhưng vẫn không được thì xin hãy email cho chúng tôi qua địa chỉ sau : duong_phi_vxl@yahoo.com / duyphi@yahoo.com ,Nếu các bạn ở xa thì viết thư cho tôi qua địa chỉ : 67/30/10/25 Đường Phú Trung –P10- Q.Tân Bình- TpHCM hoặc gọi trực tiếp qua điện thoại 0914007477 gặp Dương , bằng kinh nghiệm đã từng lắp ráp loại này , chúng tôi hy vọng sẽ đem đến cho bạn những niềm vui bằng một tấm lòng đầy nhiệt huyết mà bạn không phải tốn một khoảng chi phí nào ! Nhân đây chúng tôi cũng xin nói luôn là chúng tôi làm việc này trên cơ sở hoàn toàn tự nguyện vì sự phát triển nền kỹ thuật của nước nhà ngày mỗi một hơn mà thôi!.

_____ < Í O Í > _____

Mời các bạn tiếp tục xem-và hãy tự lắp ráp cho mình một mạch nạp cho chip AT89c2051 - 20 chân- nhỏ gọn- dễ lắp –chạy ngay sau khi lắp:



Vấn đề 3:

MẠCH NẠP : AT89C2051 - QUA CỔNG COM



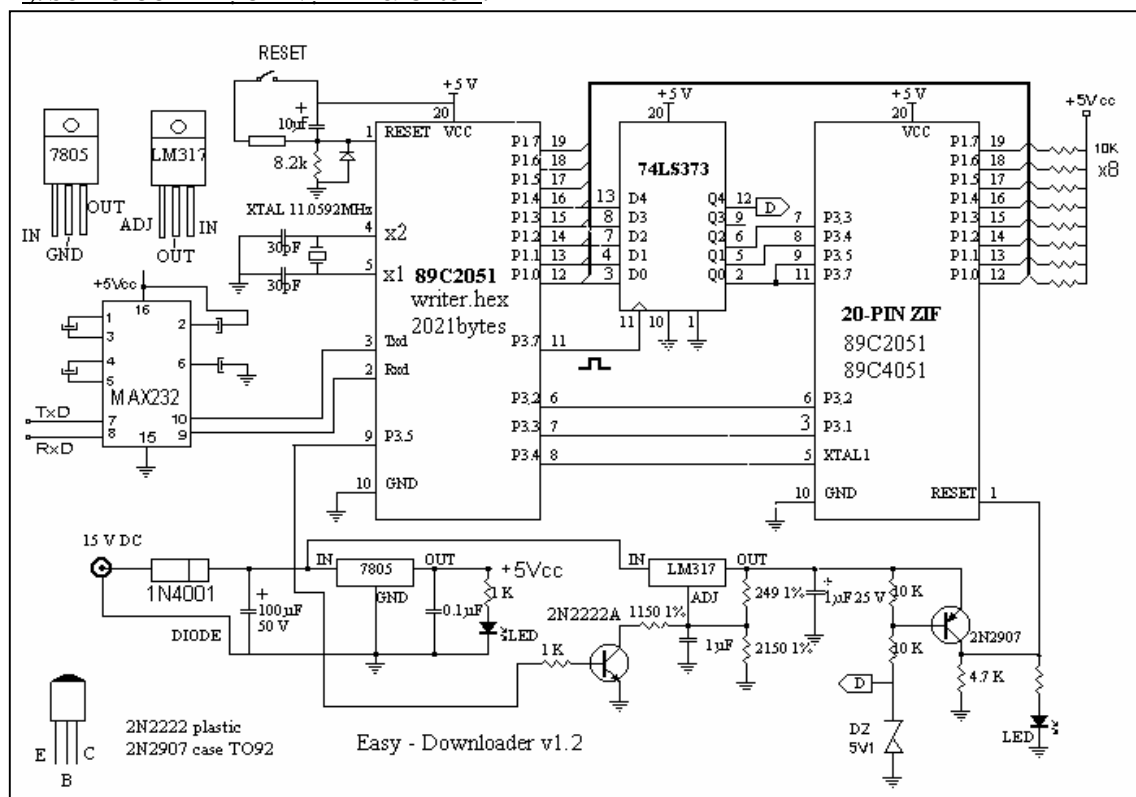
Nếu bạn không muốn dùng con Chip 89c51- 40 chân vừa to , lại đắt tiền thì bạn có thể dùng con chip 89c2051 –20 chân vừa nhỏ gọn , vừa tiết kiệm tiền bạc nhưng vẫn đầy đủ các tính năng như con chip 89c51.Và sau đây là mạch nạp chương trình con chip 89c2051 này ,ngoài ra mạch nạp AT89C2051 còn nạp được cho con chip 89c4051 luôn.! Rất dễ lắp ráp .

I- ĐÔI LỜI GIỚI THIỆU :

Con chip 89c2051 rất nhỏ gọn nên nó được sử dụng rất nhiều trong các ứng dụng nhỏ. Nếu bạn muốn vừa học Vi Xử Lý đồng thời cũng muốn khám phá nó qua các ứng dụng cụ thể, qua các dự án thực tế để phát triển 89C2051 , 89c4051...với ngôn ngữ lập trình Assembly thì “ **Mạch Nạp AT89C2051** ” chính là câu trả lời . Nó rất rẻ và mọi người đều có thể tự làm được .

II- PHẦN CỨNG : SƠ ĐỒ VÀ BƠ MẠCH IN

1). SƠ ĐỒ CỦA MẠCH NẠP AT89C2051 :



Hình 4

TRÊN SƠ ĐỒ BẠN HÃY LƯU Ý HAI ĐIỂM SAU :

- Kí hiệu IC ghi **89C2051 Writer.hex 2021 bytes** gọi là con **Chip chính** (**Chip Master**)
- Kí hiệu IC ghi **20 -PIN ZIF Socket 89c2051/ 89c4051** đây là cái **Socket** hay là chân đế để cắm IC (còn gọi là **đế cắm IC 20 Chân**) , hay gọi là **con chip phụ** (**Chip Slave**)



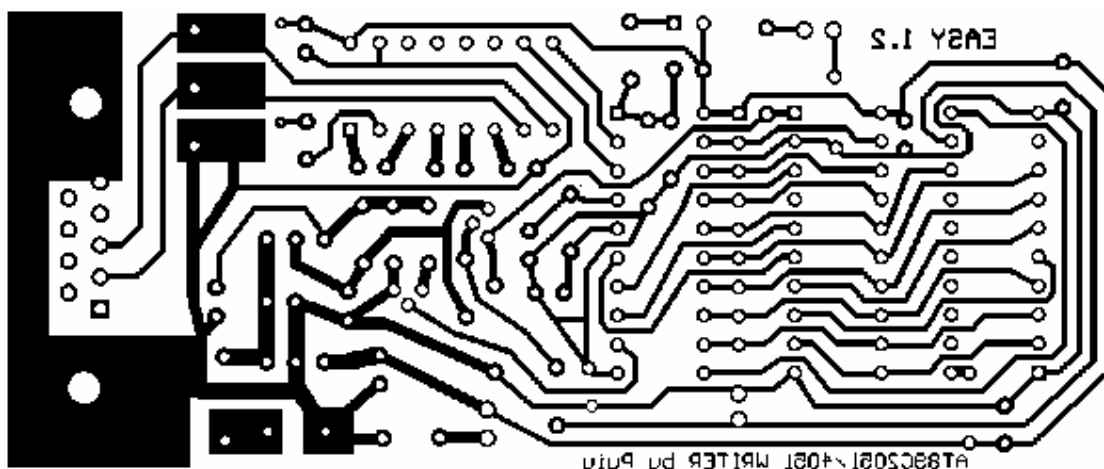
Đây là hình dáng thật của Socket (đế để gắn IC vào) 20 chân .

Hãy nhớ rằng :

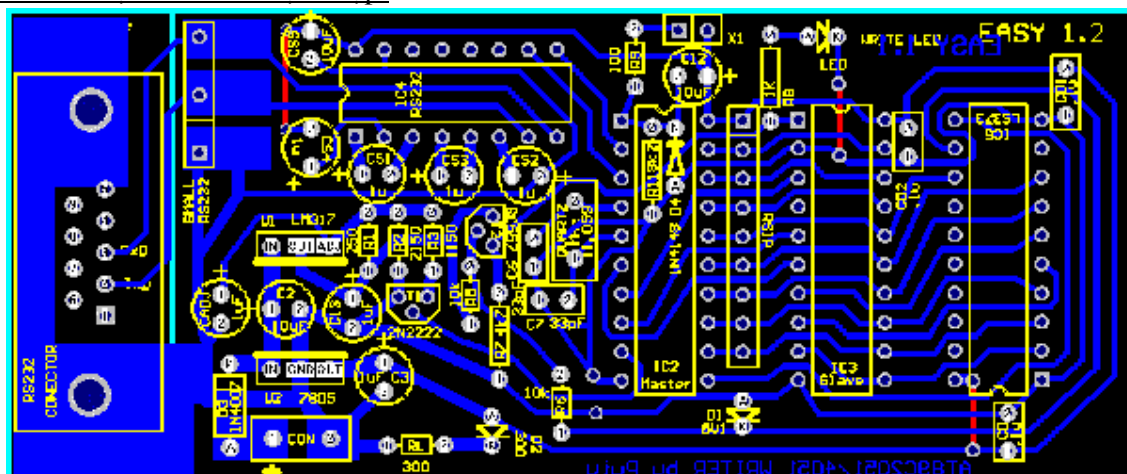
Đầu tiên bạn cần phải nạp file Writer.HEX vào cho con ChipMaster . Nhưng khổ nỗi là lấy đâu ra mạch nạp để nạp file Writer.Hex này cho con Chip Master ?? Bạn đừng lo , bạn có thể mua con Chip Master –đã nạp sẵn này tại quầy photo B1 tại trường đại học Bách Khoa hoặc bạn cũng có thể mua đĩa mềm 1,4Mb tôi có chép sẵn file Writer.hex trong đó luôn rồi , bạn có thể đem ra các tiệm điện ở chợ Điện Tử Nhật Tảo , ở đó họ có sẵn máy nạp- họ sẽ nạp cho bạn với giá mỗi lần nạp là 2.000đ.

2) BỘ MẠCH IN- CỦA MẠCH NẠP AT89C2051 :

A) Hình ảnh hướng dẫn lắp ráp linh kiện lên bộ mạch in:



B) Board mạch in- của mạch nạp:



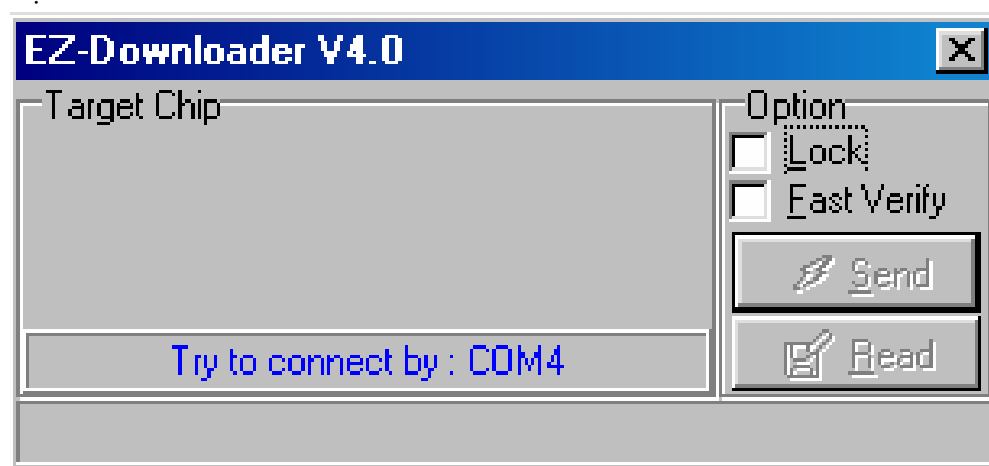
C) Hình ảnh của mạch nạp hoàn chỉnh :



III- PHẦN MỀM –CHO MẠCH NẠP AT89C2051

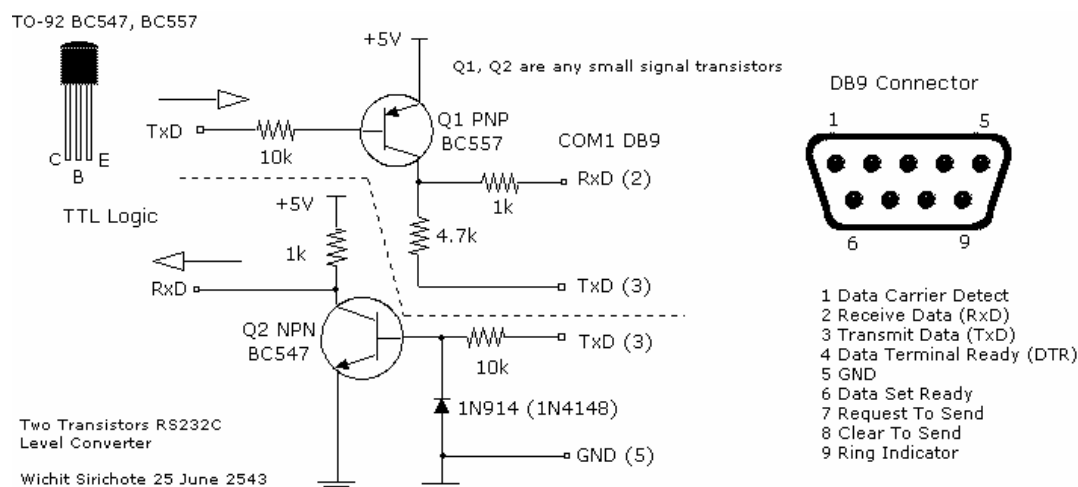
Mạch nạp AT89C2051 này giao tiếp với máy tính thông qua phần mềm EZ4.0. Hiện thời đây là phần mềm mới nhất và hoàn toàn miễn phí chạy trên nền Windows rất đẹp và rất dễ dàng sử dụng .

Giao diện của nó như sau:

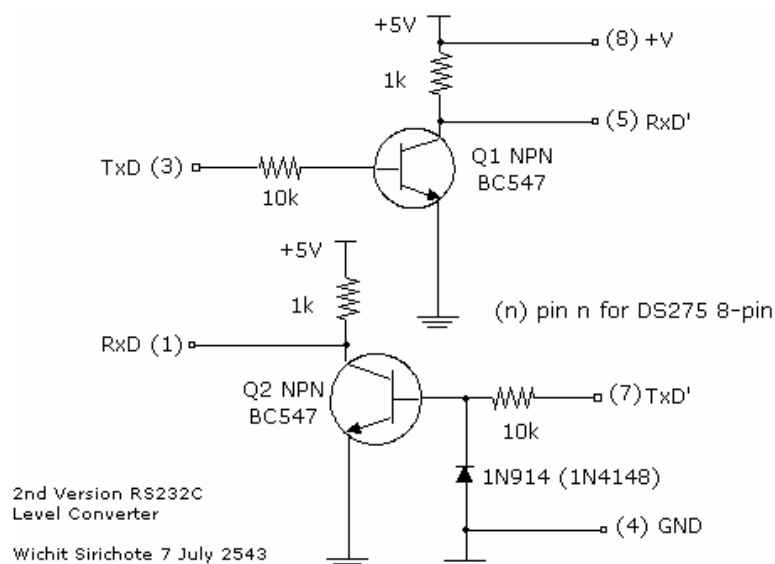


Hãy nhớ rằng : MAX 232 có thể được thay thế bằng 2 con transistor hết sức đơn giản . Khi đó bo mạch của bạn sẽ nhỏ gọn hơn, đồng thời bạn cũng có thể tiết kiệm một khoảng tiền rất đáng kể .

Sơ đồ thay thế như sau:

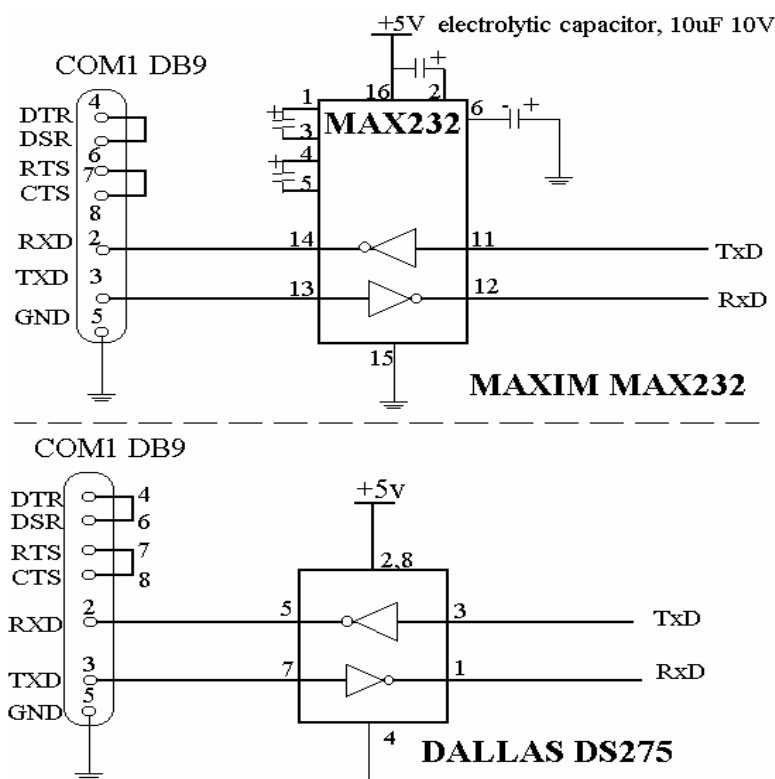


Một kiểu khác:



Hãy nhớ rằng : MAX 232 – 16 chân còn có thể được thay thế bằng con chip DS257 hết sức đơn giản chỉ có 8 chân thôi . Khi đó bo mạch in của bạn sẽ nhỏ gọn hơn .

Sơ đồ thay thế như sau:

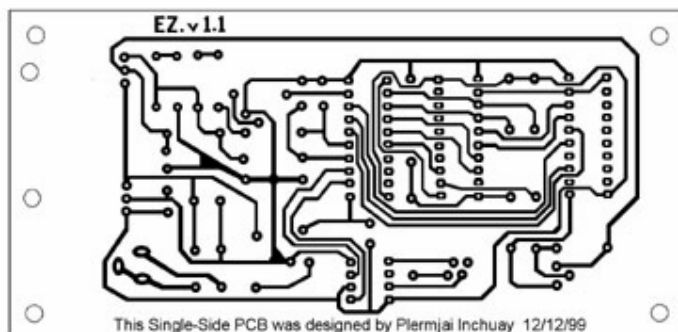


VÀ SAU ĐÂY XIN MỜI BAN XEM MỘT LOẠI BO MẠCH IN KHÁC CŨNG LẮP RÁP THEO Y NHƯ SƠ ĐỒ MẠCH NẠP AT89C2051 Ở TRÊN:

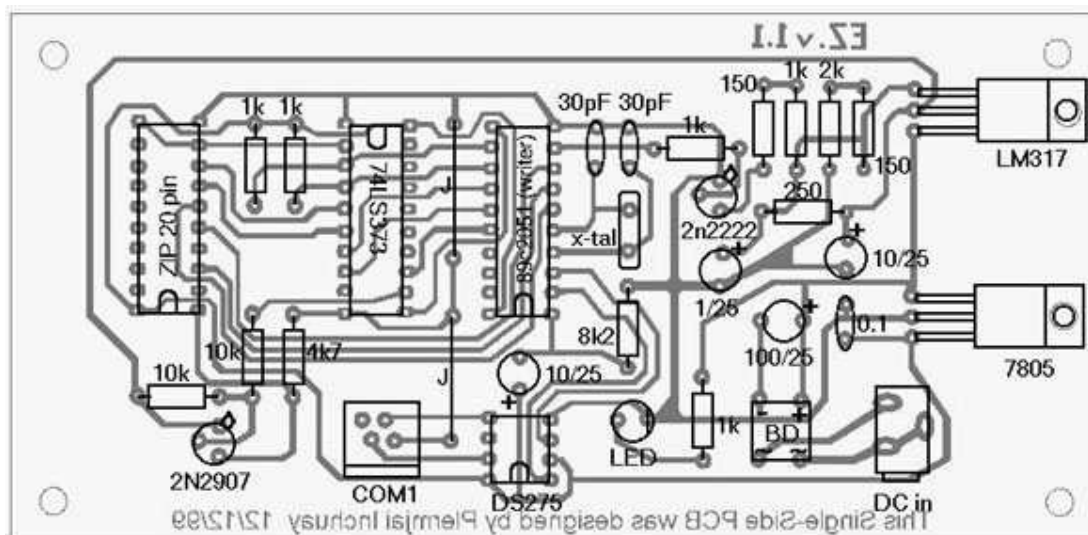
A) Hình ảnh của mạch nạp AT89C2051 hoàn chỉnh:



b) Bo mạch đồng :

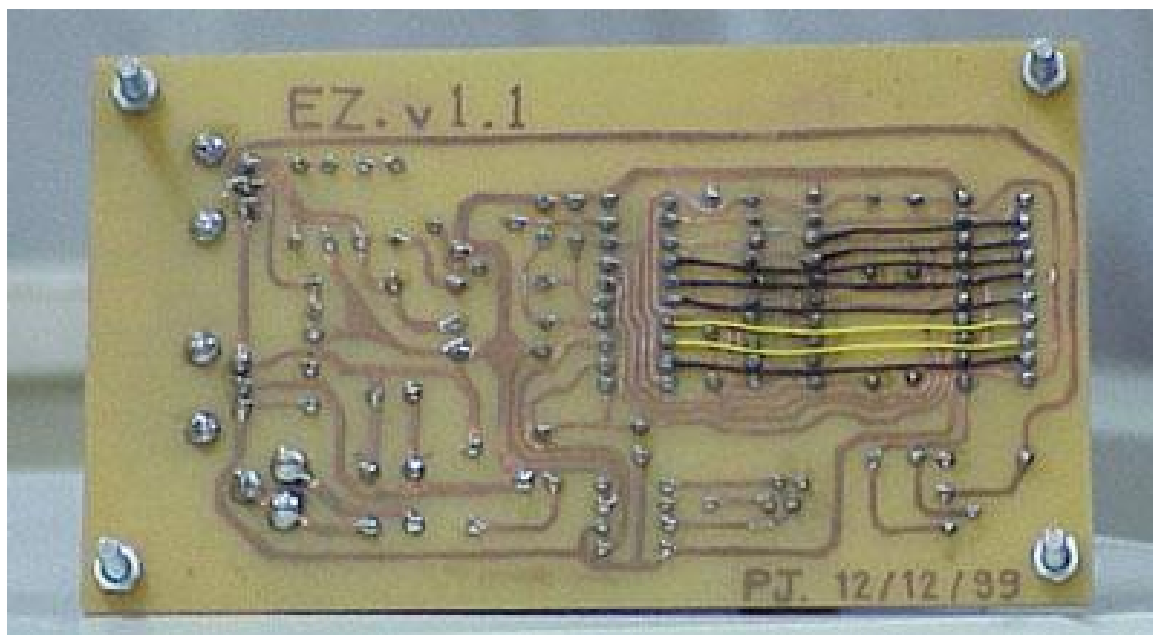


c) hình hướng dẫn lắp ráp linh kiện lên bo đồng:



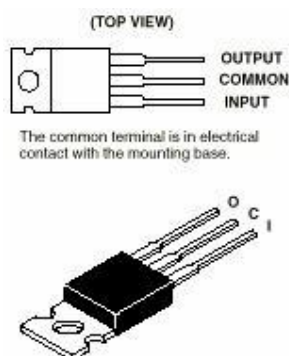
d) Bo mạch được làm bằng thủ công rất đơn giản :

Bảng mạch này được làm bằng tay hết sức giản đơn, nhưng bạn phải hết sức lưu ý nếu bạn muốn làm theo mạch này là: Bạn phải hàn các dây từ chân 12 đến 19 của con chip 89c2051 Master đến các chân 12 đến 19 của ZIF Socket như trong hình dưới đây:



IV- MỘT SỐ LƯU Ý VỀ CÁC LINH KIỆN CẦN DÙNG:

1- CÁCH XÁC ĐỊNH CÁC CHÂN ĐIỆN ÁP VÀO RA CỦA 7805



Output (O):= Chân điện áp ra 5V

Command(C):= Chân này hàn xuống Mass (GND)

Input (I):= Điện áp vào 9V à 12V

Hãy nhớ rằng: Chân Command rất quan trọng, chúng ta cần hàn nó xuống mass(GND) một cách hết sức chắc chắn nếu như chân này hàn không khéo có làm cho điện áp ở ngõ ra tăng vọt lên rất cao gần bằng điện áp ngõ vào (9-12V) sẽ làm hỏng các chip.

Hãy nhớ rằng : MAX 232 có thể được thay thế bằng 2 con transistor hết sức đơn giản. Khi đó bo mạch của bạn sẽ nhỏ gọn hơn, đồng thời bạn cũng có thể tiết kiệm một khoảng tiền rất đáng kể.

Sơ đồ thay thế như sau:

2- CÁCH XÁC ĐỊNH CHÂN B-C-E CỦA TRANSISTOR

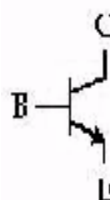
Lưu ý : Các Transistor 2N222- được thay bằng C1815

Và Transistor 2N2907 được thay bằng A1015

Sau đây là cách xác định chân B_C_E cho transistor A1015 và C1815

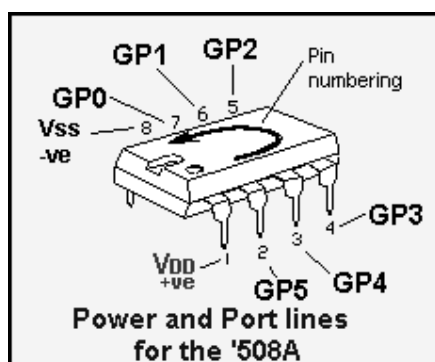


Hình dáng thực tế C1815
/ A1015



Kí hiệu của C 1815/ A 1015

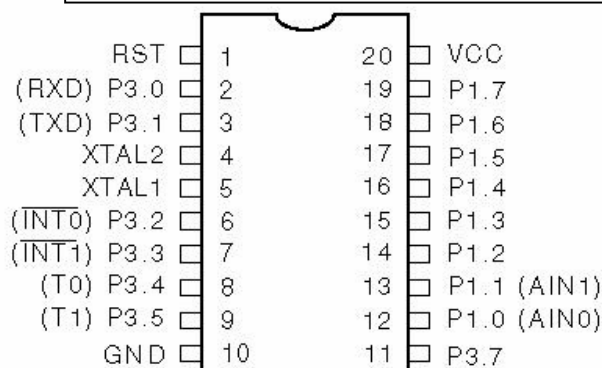
3- CÁCH XÁC ĐỊNH CHÂN CỦA IC



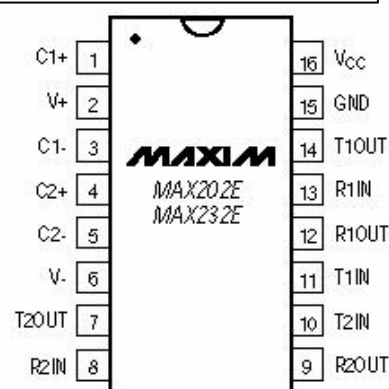
Hãy nhớ rằng : Hầu hết trên các IC đều có một dấu chấm ở một góc IC hay một vật bán nguyệt ở trên một cạnh của IC, và quy tắc đếm số thứ tự chân của IC luôn theo chiều ngược với kim đồng hồ.

Chân số 1 của IC chính là chân đầu tiên nằm trên cạnh bên trái của dấu Chấm hay Vật bán nguyệt. Với quy ước là vật bán nguyệt hay dấu chấm phải được đặt bên tay trái!.

SƠ ĐỒ CHÂN CỦA CHIP AT89C2051



SƠ ĐỒ CHÂN CỦA MAX232



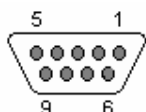
4) SƠ ĐỒ CHÂN CỦA CỔNG COM DB9:



(9p D-SUB male at PC (DTE))

Hãy nhớ rằng : Cổng com gắn trên máy tính là cổng “Đực”(male)

Để nhận dạng cổng COM đực , bạn chỉ cần nhìn vào thì bạn sẽ thấy có các “ chấu “ ở bên trong



(9p D-SUB female at cable / mouse (DCE))

Hãy nhớ rằng : Cổng mà ta cần dùng để lắp ráp mạch nạp là cổng COM này , nó là cổng “cái “.

Để nhận ra cổng COM cái , bạn chỉ cần nhìn vào thì sẽ thấy chỉ toàn là” các “ lỗ”.

Cần lưu ý

: Rất nhiều bạn mua lộn giữa Cổng đực và cổng Cái , cho nên khi đi mua bạn cần nói rõ là mua cổng nào , và cũng nhớ là ở trong máy tính của chúng ta đã sẵn có cổng đực rồi . Giá mỗi cổng COM -DB9 khoảng 5.000đ mà thôi .Nhắc lại là phải hết sức lưu ý kiểm tra cho kỹ trước khi hàn cổng COM lên mạch nạp , và trên mạch nạp luôn là cổng Cái

5) CÁCH ĐỌC GIÁ TRỊ ĐIỆN TRỞ

CÁCH TÌM GIÁ TRỊ ĐIỆN TRỞ THÔNG QUA CÁC VẠCH MÀU

Để tính toán giá trị của điện trở thông qua các mã vạch màu (color coded stripes) trên điện trở , thường theo các bước sau:

Bước 1 : Đặt con điện trở sao cho Vạch màu Vàng kim hay nhũ bạc của điện trở -phải ở góc bên phía tay phải của bạn.

Bước 2 : Bạn hãy nhìn kỹ 2 vạch màu đầu tiên ở phía bên trái (the first two stripes on the left end). Tiếp theo bạn hãy chuyển đổi các màu mà bạn đọc được ra thành số theo bảng chuyển đổi cho sẵn ở dưới đây.

Bước 3 : Xong , bạn hãy nhìn tiếp vạch thứ 3 . Vạch này chính là hệ số nhân .

Bạn hãy chuyển đổi vạch màu thứ 3 này ra giá trị theo bảng cho sẵn phía sau.

Bước 4 : Đến đây bạn chỉ việc lấy 2 con số ở bước 2 nhân với con số ở bước 3 là bạn đã tìm được giá trị của điện trở . Giá trị của điện trở mà bạn vừa tìm sẽ mang **đơn vị là ohms**

Vạch màu nhũ kim hay nhũ bạc (tức là vạch thứ 4 tính từ trái sang) sẽ cho chúng ta biết sai số của nó so với giá trị ghi trên điện trở đó. Theo tôi thì chúng ta không nên phải bận tâm đối với điều này, có nghĩa là chúng ta không cần biết nó trong khi thực hành.

Hãy nhớ rằng: Đối với điện trở chúng ta cần hết sức lưu ý đến một thông số vô cùng quan trọng đó là công suất chịu đựng của nó. Thông thường công suất điện trở bán ở thị trường sẽ có các mức sau : 1/4 Watt , 0.5 Watt , 1W , 5W , 10W..... Nhưng thông dụng nhất vẫn là điện trở 1/4 Watt- **cho nên khi bạn ra tiệm mua phải nói rõ điện trở cần mua là bao nhiêu Watt- thế mới là dân điện chuyên nghiệp !**. Giá hiện thời 100 con khoảng 2.500 ĐVN

BẢNG MÃ MÀU ĐIỆN TRỞ .
(Với vạch nhũ kim hay bạc nằm bên phải)

Màu	Vạch 1	Vạch 2	Vạch 3	Vạch 4
Đen	0	0	x1	
Nâu	1	1	x10	
Đỏ	2	2	x100	
Cam	3	3	x1,000	
Vàng	4	4	x10,000	
Lục (Green)	5	5	x100,000	
Lam (Blue)	6	6	x1,000,000	
Tím	7	7		
Xám	8	8		
Trắng	9	9		
Nhũ Kim				5%
Bạc				10%

Ví dụ:

Cho một điện trở có các vạch màu tính từ trái qua như sau :

Xanh da trời(lam-blue) ,Xám , Đỏ, Nhũ vàng .Hãy tìm giá trị điện trở của nó.

Bước 1.: Vạch màu nhũ vàng –phải được đặt bên phải , rồi đến Bước 2

Bước 2: Vạch đầu tiên là màu xanh da trời(blue) –tra bảng thì thấy nó tương ứng với số 6. Vạch thứ 2 là màu xám và cũng làm tương tự ta có được số 8. Như vậy 2 con số đầu tiên của điện trở có giá trị là 68.

Bước 3: Vạch thứ 3 là màu đỏ -có nghĩa là x 100.

Bước 4: Giá trị của điện trở tìm được sẽ là : $68 \times 100 = 6800$ ohms

(6800 ohms = 6.8 kohms).

Vạch màu nhũ đồng có nghĩa là giá trị chính xác của điện trở sẽ sai khác 5% so với thực tế . Giá trị của nó có thể sẽ nằm giữa 6,460 Kohms and 7,140 Kohms. (Do sai số 5% của $6,800 = 0.05 \times 6,800 = 340$)

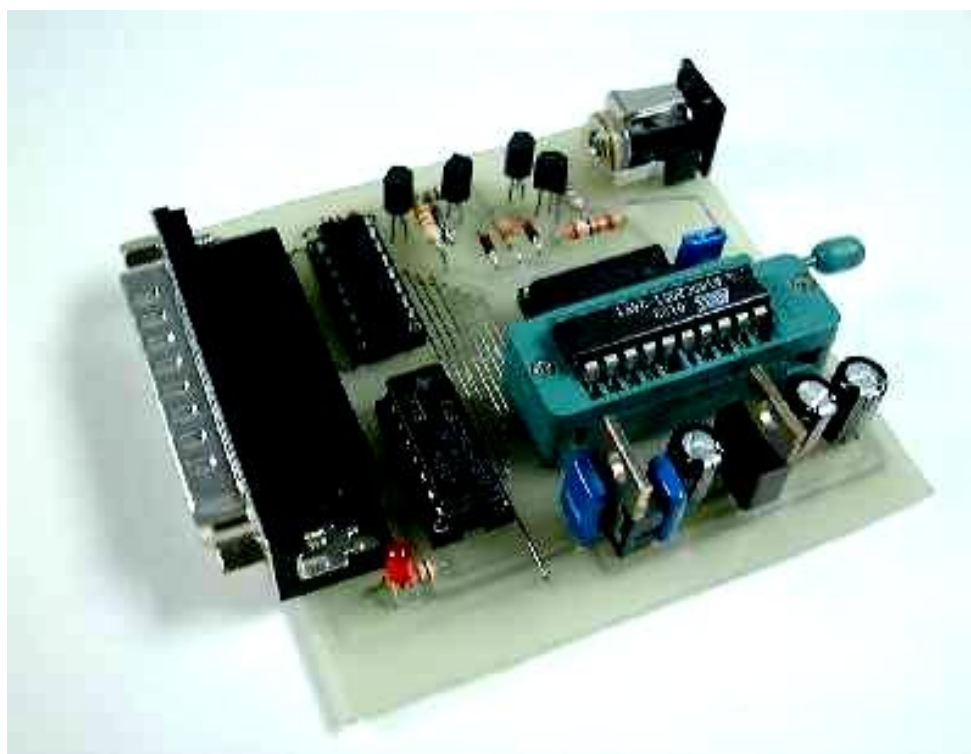
VI- MỘT SỐ CÁC LƯU Ý:

- Biến thế điện có thể dùng loại 450mA cho đến 1Ampe . Với ngõ ra AC (xoay chiều) tối thiểu là 12V –AC . Tốt nhất là nên mua biến thế 1 Ampe có ngõ ra 15 V giá khoảng 13.000đ .
- IC ổn áp 7805 bạn không nên mua loại có chữ màu trắng –loại này đa số là của Trung Hoa sản xuất nên không khi nào đạt đúng 5V như con số đã ghi cả, bạn nên mua loại chữ 7805 chìm , loại này chịu nhiệt cao và rất chuẩn volt . Giá của nó khoảng 3000đ , nếu mua hàng cũ thì mỗi con chỉ hết 500 đ rất rẻ(bán rất nhiều ở chợ Nhật Tảo) .!
- Tương tự cho LM317 cũng nên mua loại chữ chìm và giá cả cũng như 7805 .
- Transistor 2N2222 bạn có thể thay bằng C1815 giá 300đ/c
- Còn các linh kiện đều có bán ở chợ Nhật Tảo

Trong quá trình lắp ráp nếu bạn có gặp trục trặc hay bất kỳ một sự cố nào sau khi chính bản thân bạn đã cố sửa nhưng vẫn không được thì xin hãy email cho chúng tôi qua địa chỉ sau : duong_phi_vxl@yahoo.com / duyphi@yahoo.com ,hoặc gọi trực tiếp qua điện thoại 0914007477 hay 08.8616115

**XIN MỜI BẠN XEM TIẾP MỘT LOẠI MẠCH NẠP VI XỬ LÝ KHÁC
BẰNG CỔNG MÁY IN VỚI TỐC ĐỘ CỰC KỲ NHANH -LẠI RẤT ĐƠN GIẢN !**

Đây là hình ảnh hoàn chỉnh của mạch nạp 89c2051 qua cổng máy in:



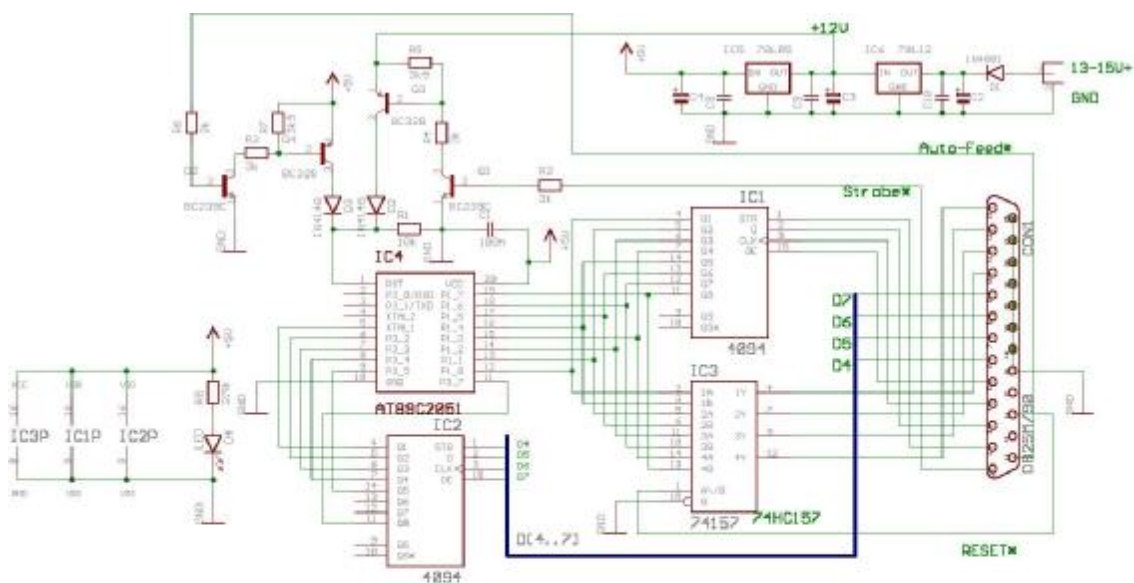
Vấn đề 4
**MẠCH NẠP:
AT89C2051 QUA CỔNG MÁY IN**


Sau đây là một loại “**mạch nạp chương trình**” cho AT89C2051 thông qua cổng máy in (Parallel Printer Port), loại “Mạch nạp” này còn gọi là “**mạch nạp song song**”.

Ưu điểm: lớn nhất của loại này là thời gian nạp chương từ máy tính dưới dạng “file Hex” sang vi điều khiển cực kỳ nhanh.

Hãy nhớ rằng: Các bạn muốn nạp chương trình mà bạn đã viết vào cho con chip vi khiển, thì trước hết bạn phải biên dịch nó ra file Hex trước đã, rồi từ đó bạn mới dùng mạch nạp này để nạp file Hex mà bạn vừa biên dịch vào cho con chip AT89C2051.

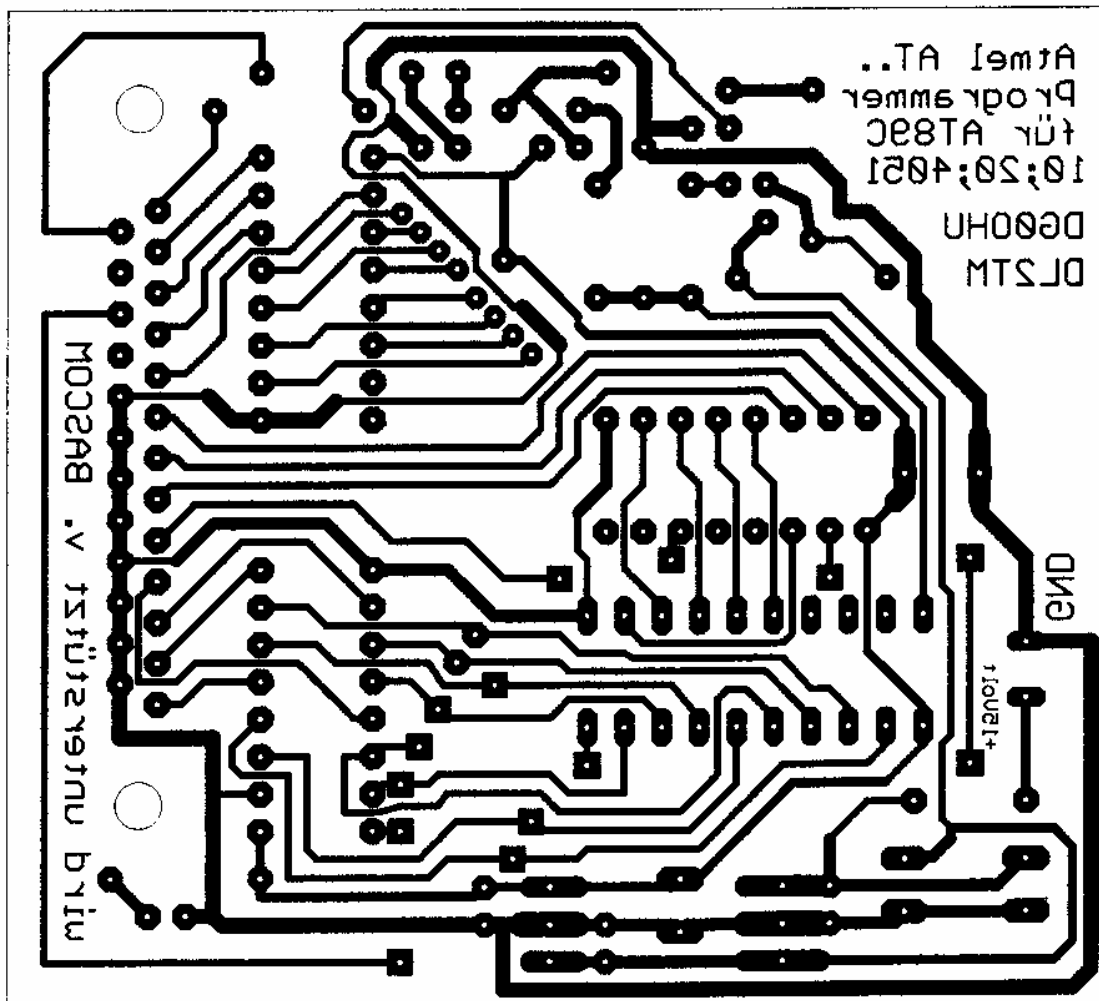
Nếu Các bạn muốn biết cách làm sao để biên dịch ra file Hex từ một file ASM (được viết bằng ngôn ngữ Assembly) thì các bạn hãy xem Phần 3 “**Hướng Dẫn Sử Dụng Các Phần Mềm Vi Xử Lý**” của quyển sách này.

I. SƠ ĐỒ LẮP RÁP :

AT89C2051 Programmer by Henry Arndt, DL2TH nach Funkamateu 4/96 Seite 116	
TITLE: Atmel-Programmer DL2TH_Ver.01U	
Document Number:	REV:
Dates 28.11.2001 21:04:38	Sheet: 1/1

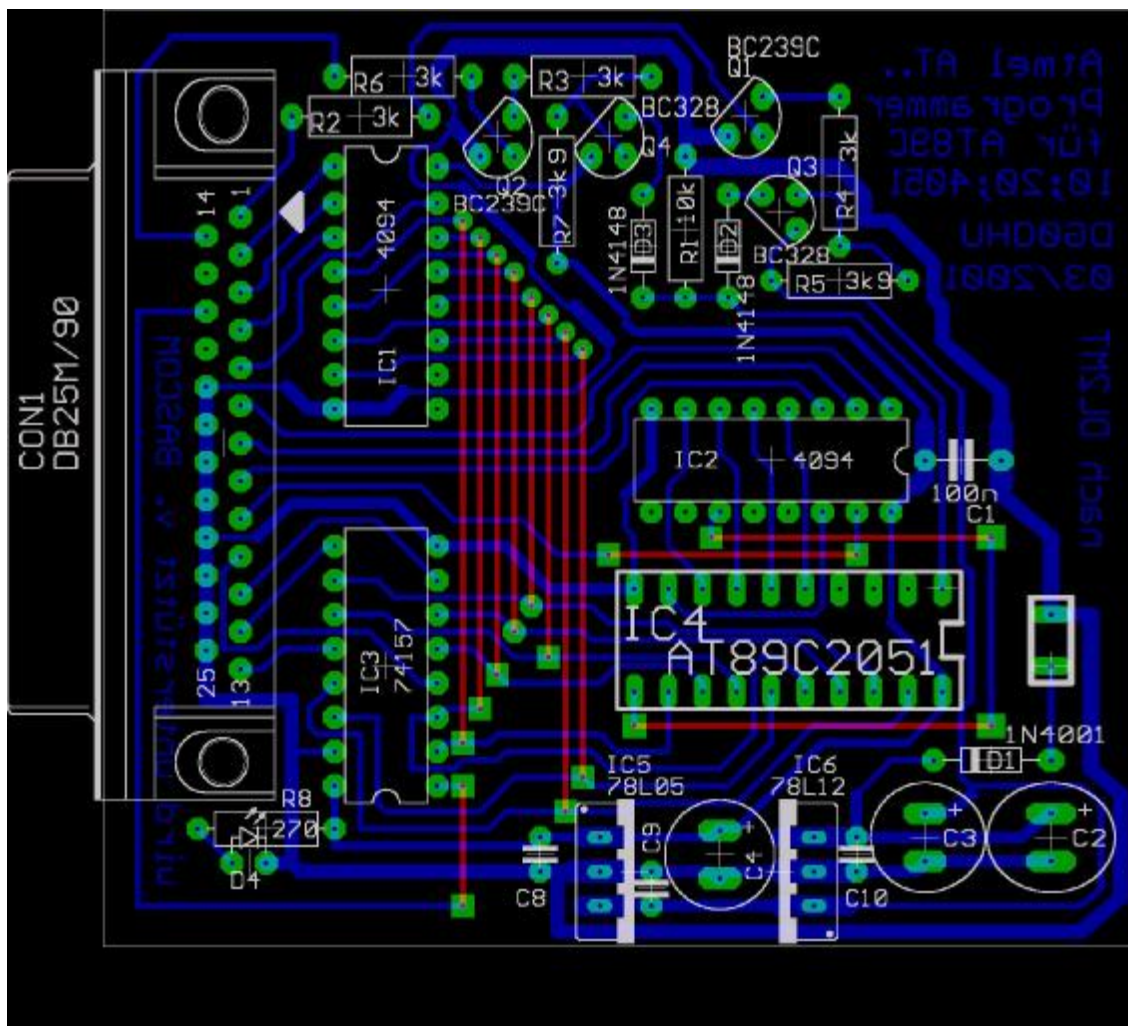
Đây là Board mạch in để gắn linh kiện , các bạn có thể liên hệ với phòng Photo B1 trường đh.Bách Khoa (nó nằm ở ngay dưới cầu thang dây B1) để lấy bản vẽ này đã được in sẵn bằng máy in Laser , sau đó bạn hãy dùng “bàn la” (hay còn gọi: “bàn ủi” để đưa nó lên bo đồng).

Lưu ý: Nếu các bạn muốn biết cách “Thiết Kế Bô Mạch In “ và cách tạo ra một bản mạch in chất lượng cao thì bạn hãy xem **Phần 2 “HƯỚNG DẪN LÀM BÔ MẠCH IN CHẤT LƯỢNG CAO”** trong quyển sách này.

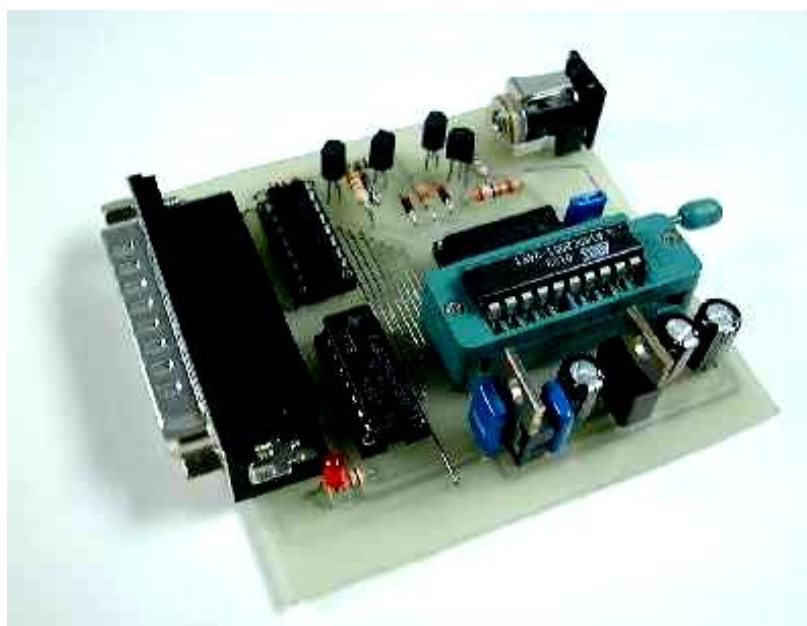


Hãy lưu ý rằng: Bạn có thể download phần mềm cùng sơ đồ và board mạch in và nhiều vấn đề khác rất hay tại website : **www.qsl.net/dg5dbz/main_eng.html**

III. HƯỚNG DẪN LẮP RÁP LINH KIỆN LÊN BƠ MẠCH IN:

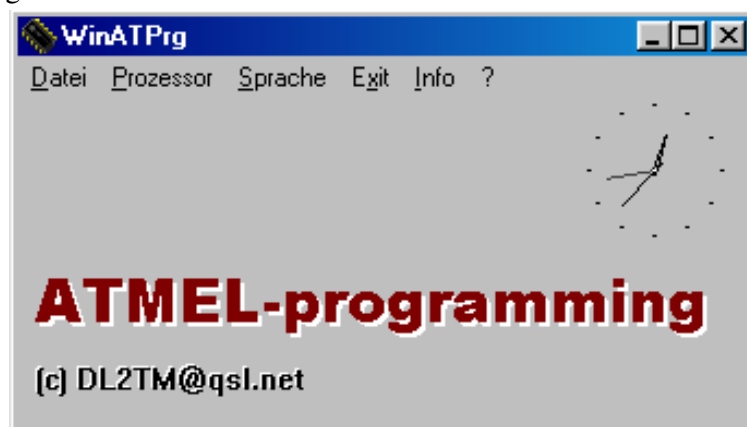


IV- MẠCH HOÀN CHỈNH SAU KHI ĐÃ LẮP RÁP XONG:



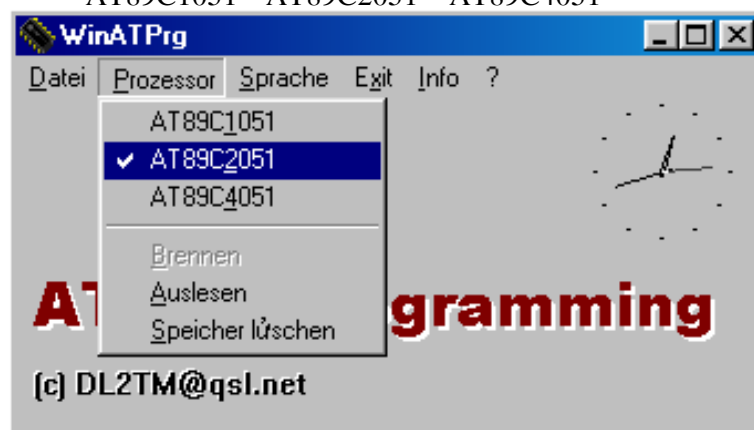
V-PHẦN MỀM-CHO MẠCH NẠP 89C2051 QUA CỔNG MÁY IN:

Mạch nạp này sử dụng phần WinATProg chạy trên môi trường windows 98/ Me dung lượng 273kb . Giao diện như sau :

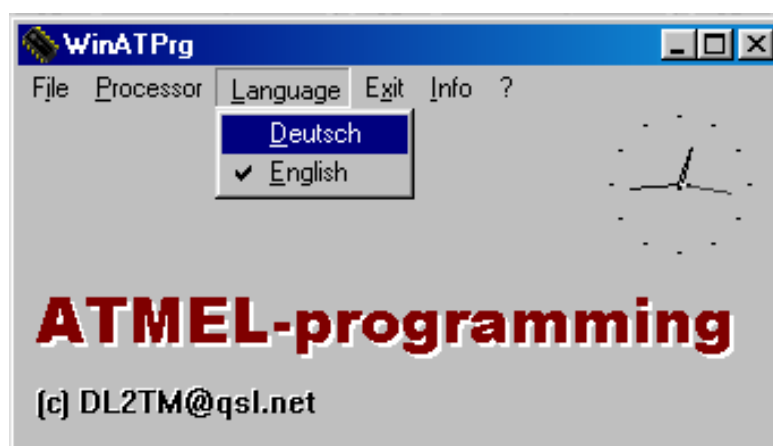


Bạn có chọn loại Chip để nạp trong mục menu Prozessor :

* AT89C1051 * AT89C2051 * AT89C4051



Bạn cũng có thể chọn lại ngôn ngữ hiển thị trên giao diện là tiếng anh :



Hãy lưu ý rằng: Bạn có thể download phần mềm này cùng sơ đồ board mạch in và nhiều vấn đề khác rất hay tại website :

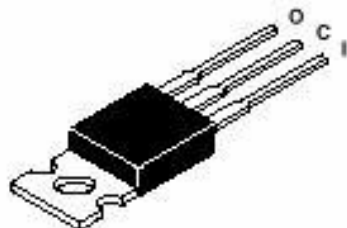
www.qsl.net/dg5dbz/main_eng.html

IV- MỘT SỐ LƯU Ý VỀ CÁC LINH KIỆN CẦN DÙNG:

1- CÁCH XÁC ĐỊNH CÁC CHÂN ĐIỆN ÁP VÀO RA CỦA 7805



The common terminal is in electrical contact with the mounting base.



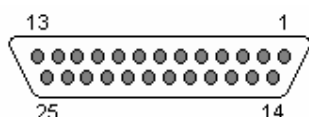
Output (O):= Chân điện áp ra 5V

Command(C):= Chân này hàn xuống Mass (GND)

Input (I):= Điện áp vào 9V ã 12V

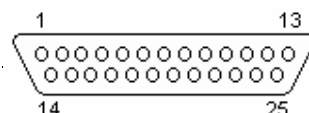
Hãy nhớ rằng: Chân Command rất quan trọng , chúng ta cần hàn nó xuống mass(GND) một cách hết sức chắc chắn nếu như chân này hàn không khéo có làm cho điện áp ở ngõ ra tăng vọt lên rất cao gần bằng điện áp ngõ vào (9-12V) sẽ làm hỏng các chip .

2- SƠ ĐỒ CHÂN CỦA CỔNG MÁY IN



—25p D-SUB female at PC

Hãy nhớ rằng : Cổng **Máy in** gắn trên máy tính là cổng “Cái”(female)
Để nhận dạng cổng Máy in Cái , bạn chỉ cần nhìn vào thì bạn sẽ thấy có các “ Lỗ ” ở bên trong



—25p D-SUB male at parallel-cable

Hãy nhớ rằng : Cổng mà ta cần dùng để lắp ráp mạch nạp là cổng Máy in này , nó là cổng “ Đực “. Để nhận ra cổng COM cái , bạn chỉ cần nhìn vào thì sẽ thấy chỉ toàn là” các “ Chấu”.

Cần lưu ý

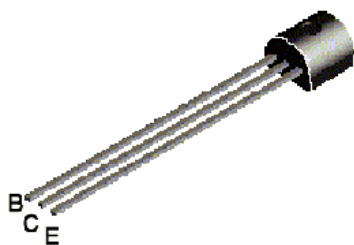
: Rất nhiều bạn mua lộn giữa Cổng đực và cổng Cái , cho nên khi đi mua bạn cần nói rõ là mua cổng nào , và cũng nhớ là ở trong máy tính của chúng ta đã sẵn có cổng đực rồi . Giá mỗi cổng máy in –DB25 khoảng 5.000đ mà thôi .Nhắc lại là phải hết sức lưu ý kiểm tra cho kỹ trước khi hàn cổng máy in lên mạch nạp , và trên mạch nạp luôn là **cổng Đực**

3- CÁCH XÁC ĐỊNH CHÂN B-C-E CỦA TRANSISTOR

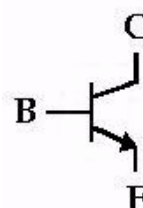
Lưu ý: Các Transistor BC239-được thay bằng C1815

Và Transistor BC328 được thay bằng A1015

Sau đây là cách xác định chân B_C_E cho transistor A1015 và C1815



Hình dáng thực tế C1815 / A1015



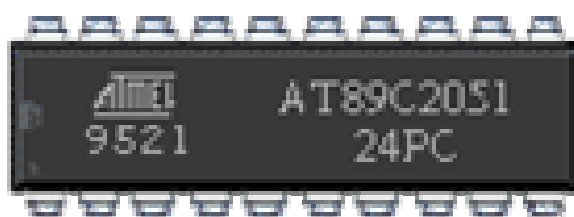
Kí hiệu của C 1815/ A 1015

4- CÁCH XÁC ĐỊNH CHÂN CỦA IC

RST	1	20	VCC
(RXD) P3.0	2	19	P1.7
(TXD) P3.1	3	18	P1.6
XTAL2	4	17	P1.5
XTAL1	5	16	P1.4
(INT0) P3.2	6	15	P1.3
(INT1) P3.3	7	14	P1.2
(T0) P3.4	8	13	P1.1 (AIN1)
(T1) P3.5	9	12	P1.0 (AIN0)
GND	10	11	P3.7

Hãy nhớ rằng: Hầu hết trên các IC đều có một **dấu chấm** ở một góc IC hay **một vạt bán nguyệt** ở trên một cạnh của IC, và quy tắc đếm số thứ tự chân của IC luôn theo chiều ngược với kim đồng hồ.

Chân số 1 của IC chính là chân đầu tiên nằm trên cạnh bên trái của dấu Chấm hay Vạt bán nguyệt.



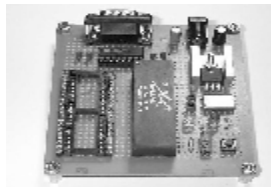
5- CÁCH ĐỌC GIÁ TRỊ ĐIỆN TRỞ

CÁCH TÌM GIÁ TRỊ ĐIỆN TRỞ THÔNG QUA CÁC VẠCH MÀU

Ở trên đã hướng dẫn, xin xem lại.

Vấn đề 5:

**MẠCH NẠP TỔNG HỢP:
89C51 & 89C2051 - QUA CỔNG MÁY IN**



Nếu bạn không muốn lắp ráp nhiều mạch nạp cho từng loại vi xử lý và bạn muốn nạp thật nhanh nhiều loại vi xử lý, thì đây chính là câu trả lời. Với bản mạch in hai mặt đang có bán sẵn – bạn có thể mua về tự ráp lấy ! Rất đơn giản – dễ làm.

I- ĐÔI LỜI GIỚI THIỆU :

Đây là mạch nạp tổng hợp nhiều loại vi xử lý khác nhau, tốc độ rất nhanh, hiện tại được nhiều người sử dụng, đa số vẫn là sinh viên. Mạch điện đơn giản, bên cạnh đó còn có sẵn bo mạch hai lớp với giá rất rẻ khoảng 50.000đ cho 1 mạch. Loại này giao tiếp qua cổng máy in.

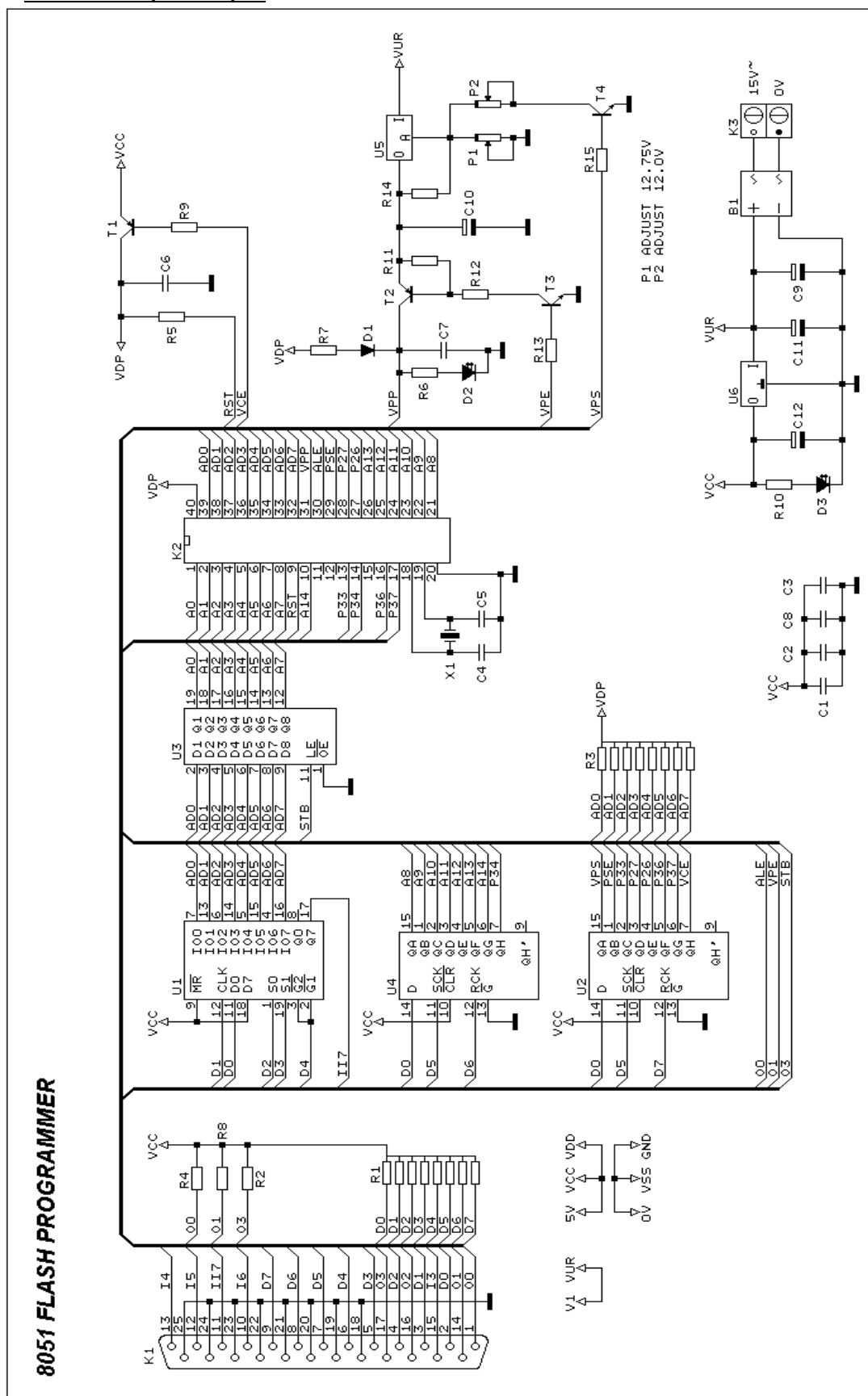
Hiện tại mạch nạp tổng hợp này có thể nạp chương trình cho các vi xử lý sau:

- AT89C51
- AT89C52
- AT89C55
- AT89C1051
- AT89C1051U
- AT89C2051
- AT89C4051

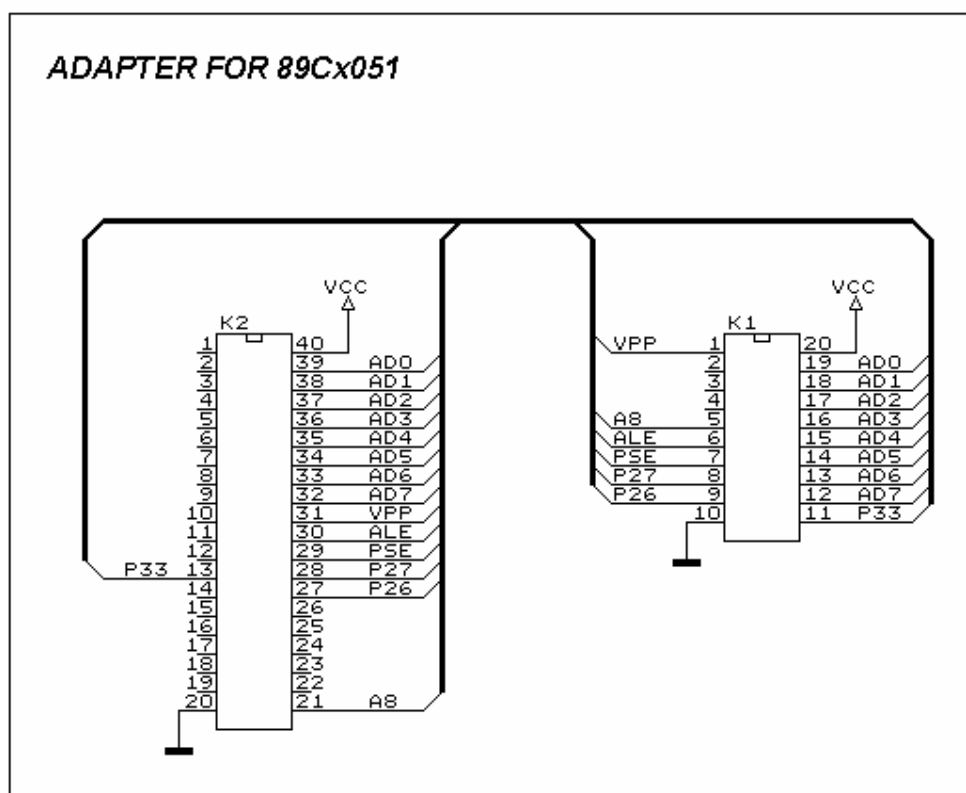
Hãy lưu ý là : Đối với loại chip 20 chân thì ta phải dùng một “**Mạch Giao tiếp 20 Chân**” để dùng thì mới dùng được . “**Mạch Giao tiếp 20 Chân**” này như thế nào thì xin bạn hãy xem các phần tiếp theo thì sẽ rõ.!

Bạn có thể mua Bo mạch nạp này tại chỗ làm mạch “ Kim Sơn – Quận 1-Tp.HCM - đường Nguyễn thị Minh Khai- hẻm số 17-19 gần cầu Thị Nghè ” giá 50.000đ/một mạch

I-SƠ ĐỒ MẠCH NAP:



Hãy lưu ý là : Đối với loại chip 20 chân thì ta phải dùng một “**Mạch Giao tiếp 20 Chân**” để dùng thì mới dùng được . sơ đồ “**Mạch Giao tiếp 20 Chân**” như sau:



Hình 6

Bạn hãy nhìn lên hình 5 thì thấy kí hiệu “ **K2** ” và bạn hãy nhìn ở hình 6 cũng thấy kí hiệu “ **K2** ”. Điều đó có nghĩa là :Các con vi xử lý 20 chân sẽ được gắn vào socket “**K1**”, socket **K1** sẽ được gắn lên **K2**. Cũng có nghĩa là Bo mạch **K1** (hình 6) là một bo rời , nó phải được ráp riêng.

II - nguồn cung cấp cho mạch nạp:

Nguồn cung cấp cho mạch nạp sẽ thông qua U6-LM7805, nên ngõ ra chỉ còn 5V cung cấp cho mạch nạp sử dụng . Biến thế nguồn nên dùng loại có ngõ ra 15 V- đến 18v AC (xoay chiều) là tốt nhất .

III - Lưu ý cách điều chỉnh hai biến trở P1 và P2 như sau:

1. Đầu tiên chỉnh P1 để được 12.75V tại ngõ ra của con ổn áp LM317 . Hãy làm cho chắc chắn là con transistor T4 ở trạng thái tắt hoặc nối chân B (base) của T4 xuống mass GND.
2. Bây giờ hãy ngắn mạch chân C của transistor T4 xuống mass ground.
3. Điều chỉnh P2 để được 12.0V tại ngõ ra của LM317 .

IV- Danh sách linh kiện cần dùng cho mạch nạp:

Thứ tự	Số lượng	Kí hiệu trên sơ đồ	Giá trị
1	1	B1	Diode cầu(BRIDGE) loại: 1A - 100v
2	5	C1, C2, C3, C6, C8	100nF
3	2	C4, C5	33pF
4	1	C7	2.2nF (Tu pi: 222 hay 0,0022pF)

5	1	C9	1000uF 25V
6	3	C10,C11,C12	10uF 25V
7	1	D1	1N4148
8	1	D2	LED RED 3mm
9	1	D3	LED GREEN 3mm
10	1	K1	DB25M-R/A CONNECTOR PCB TYPE
11	1	K2	ZIF SOCKET 40 WAY
12	1	K3	(Bo mạch in)PCB TERMINAL BLOCK 2 WAY
13	1	P1	5K bien tro
14	1	P2	50K Bien tro
15	2	R1,R3	Điện trở thành: 4K7 x 8
16	7	R2,R4,R8,R11,R12,R13,R15	4K7
17	2	R7,R5	1K
18	1	R6	3K3
19	1	R9	2K7
20	1	R10	680R
21	1	R14	220R
22	2	T2,T1	2N3906
23	2	T3,T4	2N3904
24	1	U1	74HC299
25	2	U4,U2	74HC595
26	1	U3	74HC573
27	1	U5	LM317LZ
28	1	U6	LM7805
29	1	X1	Thạch anh(CRYSTAL):4.0 MHz

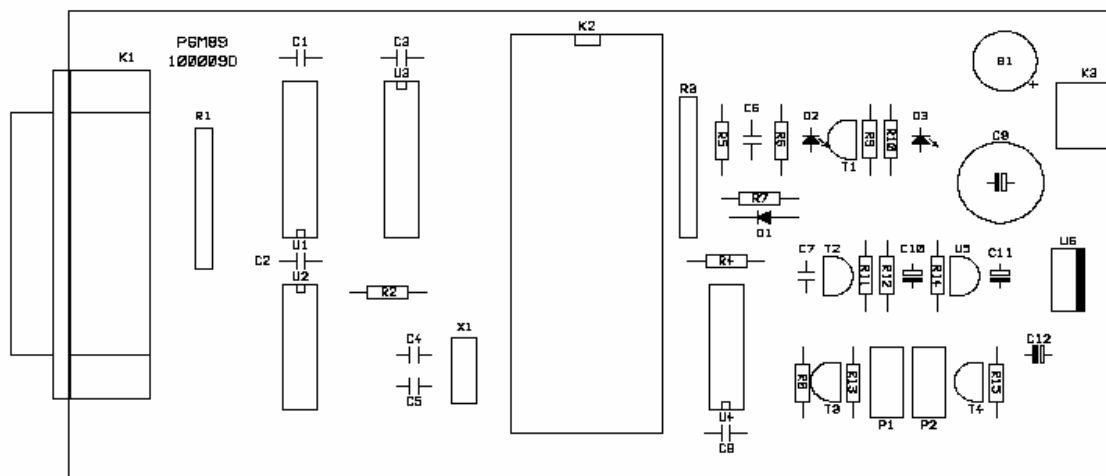
ADAPTER FOR 89CX051

1	1	K1	ZIF SOCKET 20 WAY
2	2	K2	20 WAY WW SOCKET STRIP

V- BO MACH IN:

Bạn có thể mua Bo mạch nạp này tại chỗ làm mạch “ Kim Sơn – Quận 1-Tp.HCM - đường Nguyễn thị Minh Khai- hẻm số 17-19 gần cầu Thị Nghè ” giá 50.000đ/một mạch

Tôi nghĩ mạch nạp này chúng ta nếu tự làm sẽ rất tốn kém , giá tiền mà bạn tự làm có thể hơn 100.000 đ .Do đó theo tôi đề nghị bạn nên mua là hơn. Tuy nhiên tôi cũng in ra đây các bản mạch in của nó cho bạn xem .



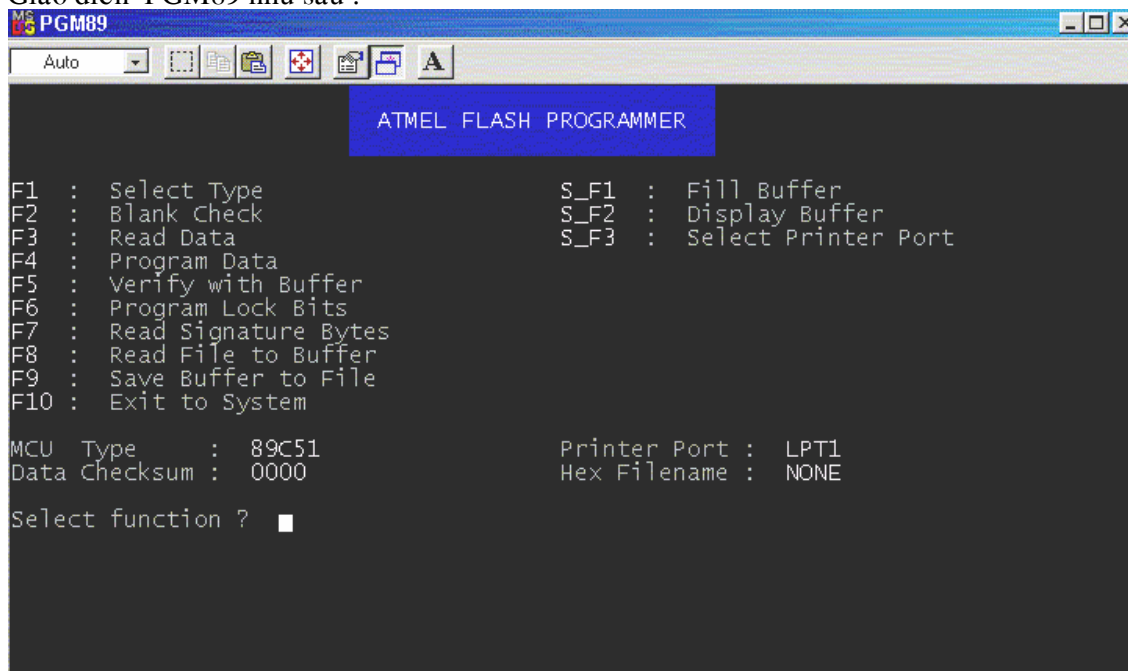
VI- PHẦN MỀM CHO MẠCH NẠP

Hiện tại dạng mạch này mới chỉ có phần mềm chạy trên môi trường DOS , tên của chương trình này là **PGM89** ,bạn có thể dễ dàng dowload tại địa chỉ website sau :

<http://chaokhun.kmitl.ac.th/%7Ekswichit/personal>

Dung lượng 51,9kb ,hoặc bạn cũng có thể liên hệ phòng Photo B1- trường đại học Bách Khoa ,tôi có ghi sẵn trong đĩa mềm 1,44mb.

Giao diện PGM89 như sau :



Các đặc tính của phần mềm này là:

Read and Write Intel Hex File (đọc và nạp các file dạng .Hex)

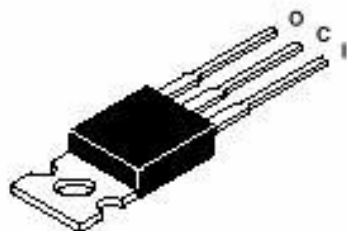
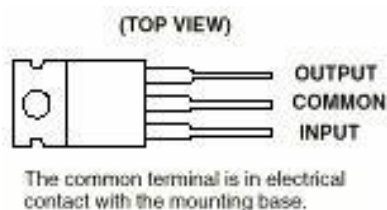
Read Chip info (đọc các thông tin có trong Chip)

Verify from hex file (Kiểm tra file HEX sau khi nạp xong)

Selection LPT1, 2 & 3 (Được lựa chọn cổng máy in)

Remember the previous settings of the programmer

(Có thể lưu/nhớ - ưu tiên cho các file đã nạp)

1- CÁCH XÁC ĐỊNH CÁC CHÂN ĐIỆN ÁP VÀO RA CỦA 7805

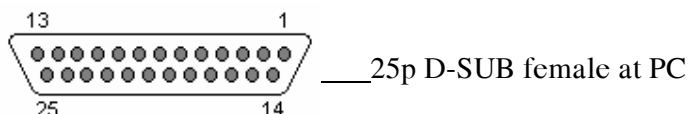
Output (O):= Chân điện áp ra 5V

Command(C):= Chân này hàn xuống Mass (GND)

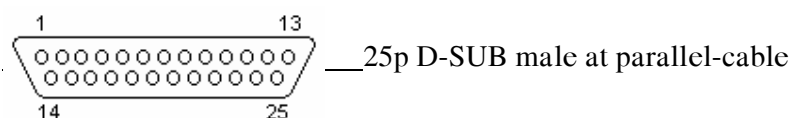
Input (I):= Điện áp vào 9V à 12V

Hãy nhớ rằng: Chân Command rất quan trọng , chúng ta cần hàn nó xuống mass(GND) một cách hết sức chắc chắn nếu như chân này hàn không khéo có làm cho điện áp ở ngõ ra tăng vọt lên rất cao gần bằng điện áp ngõ vào (9-12V) sẽ làm hỏng các chip .

2- SƠ ĐỒ CHÂN CỦA CỔNG MÁY IN



Hãy nhớ rằng : Cổng **Máy in** gắn trên máy tính là cổng “Cái”(female)
Để nhận dạng cổng Máy in Cái , bạn chỉ cần nhìn vào thì
bạn sẽ thấy có các “ Lỗ ” ở bên trong



Hãy nhớ rằng : Cổng mà ta cần dùng để lắp ráp mạch nạp là cổng Máy in
này , nó là cổng “ Đực “. Để nhận ra cổng COM cái , bạn
chỉ cần nhìn vào thì sẽ thấy chỉ toàn là” các “ Chấu”.

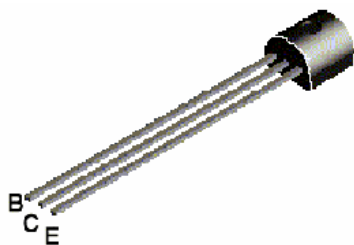
Cần lưu ý : Rất nhiều bạn mua lộn giữa Cổng đực và cổng Cái ,
cho nên khi đi mua bạn cần nói rõ là mua cổng nào , và
cũng nhớ là ở trong máy tính của chúng ta đã sẵn có
cổng đực rồi . Giá mỗi cổng máy in –DB25 khoảng
5.000đ mà thôi .Nhắc lại là phải hết sức lưu ý kiểm tra
cho kỹ trước khi hàn cổng máy in lên mạch nạp , và trên
mạch nạp luôn là **cổng Đực**

3- CÁCH XÁC ĐỊNH CHÂN B-C-E CỦA TRANSISTOR

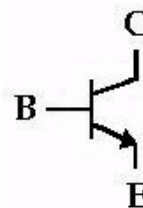
Lưu ý : Các Transistor BC239-được thay bằng C1815

Và Transistor BC328 được thay bằng A1015

Sau đây là cách xác định chân B_C_E cho transistor A1015 và C1815



Hình dáng thực tế C1815 / A1015



Kí hiệu của C 1815/ A 1015

PHẦN 2:

HƯỚNG DẪN LÀM BOMẠCH IN CHẤT LƯỢNG CAO

NỘI DUNG :

Vấn đề 6:

HƯỚNG DẪN THIẾT KẾ MẠCH IN TRÊN MÁY TÍNH
BẰNG PHẦN MỀM : **Eagle 4.01**

Vấn đề 7:

PHƯƠNG PHÁP IN BẢN VẼ LÊN BOMẠCH ĐỒNG
NHANH - ĐƠN GIẢN - HIỆU QUẢ

VẤN ĐỀ 6:

HƯỚNG DẪN THIẾT KẾ MẠCH IN TRÊN MÁY TÍNH BẰNG PHẦN MỀM : **Eagle 4.01**

I-GIỚI THIỆU :

Hiện nay việc ứng dụng máy tính để thiết kế mạch in ngày một trở nên khá phổ biến như Orcad, eagle, Protel.... Tuy nhiên dễ sử dụng hơn cả và hiệu quả nhất đó là phần mềm Eagle , nó rất đơn giản , giao diện đẹp , không quá phức tạp trong khi thiết kế mạch in.... Chính vì những đặc điểm này mà đa số các sinh viên hiện nay nhất là sinh viên trường kỹ thuật đang sử dụng phần mềm này .

II-NỘI DUNG :

Về nội dung hướng dẫn thiết kế mạch in bằng phần mềm Eagle 4.01 tôi cũng đã soạn xong và làm thành một quyển sách riêng khoảng gần 20 trang . Nếu các bạn muốn xem thì có thể liên hệ với Phòng Photo B1 trường Đh.Bách Khoa hoặc các bạn cũng có thể xem nội dung của quyển sách này trên mạng theo địa chỉ sau:

www.vixuly.cjp.net

VẤN ĐỀ 7:

PHƯƠNG PHÁP IN BẢN VẼ LÊN BOMẠCH ĐỒNG

I-GIỚI THIỆU:

Thông thường bạn cần in 1 bản mạch mà bạn tự thiết kế lên bomạch đồng(Cu) bạn thường phải ra tiệm đặt làm dùm với giá không được rẻ cho lắm (thường là 100.000đ trở lên). Vậy thì tại sao bạn lại không tự mình làm lấy với giá khoảng là 6.000đ nhỉ ? Sau đây là 1 quá trình rút kinh nghiệm của chúng tôi trong quá trình thực hiện khi đưa bản mạch từ máy tính lên bản Cu

Lưu ý: ở đây chúng tôi chỉ hướng dẫn bạn làm mạch in một mặt mà thôi . Nếu bạn muốn làm mạch in 2 mặt thì hãy liên hệ với chúng tôi hoặc bạn có thể liên hệ với cơ sở làm mạch sau :

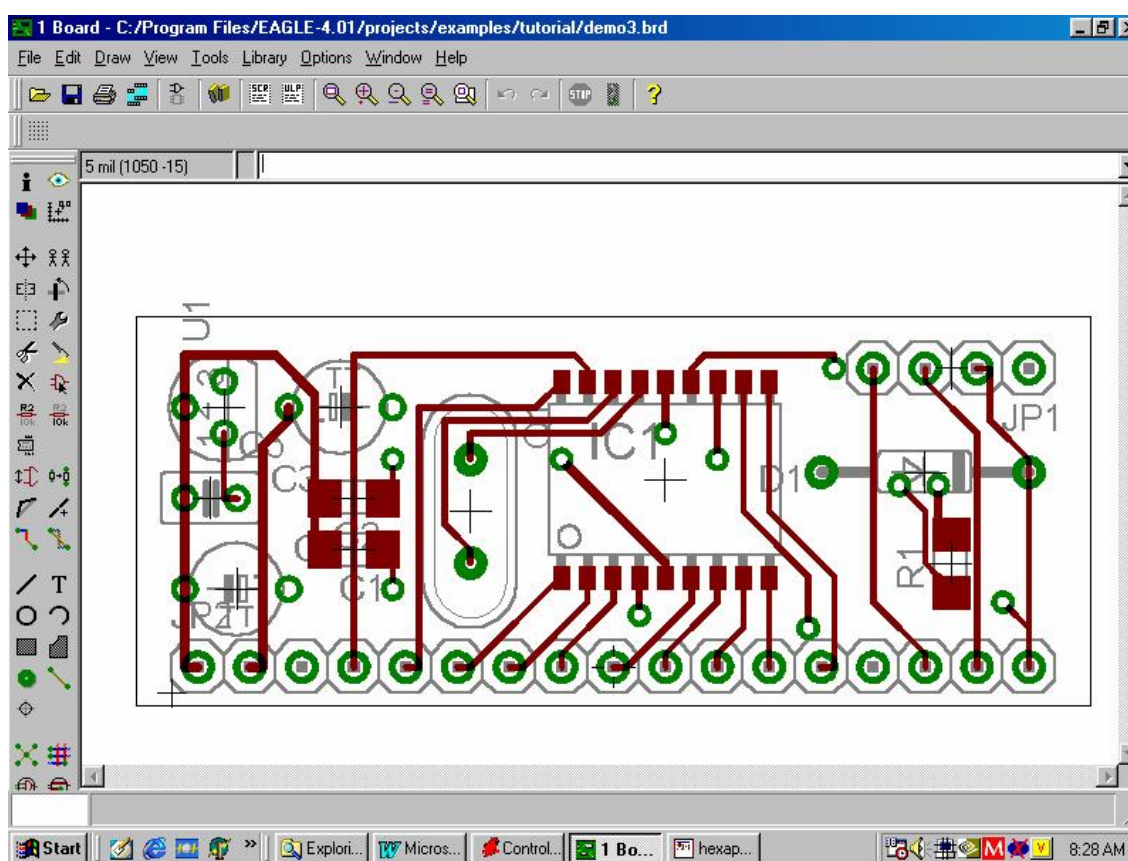
“ Kim Sơn – Quận 1-Tp.HCM - đường Nguyễn thị Minh Khai- hẻm số 17-19 gần cầu Thị Nghè ”

II-TIẾN TRÌNH LÀM MẠCH IN :

Bước 1:

Đầu tiên bạn dùng máy tính để thiết kế bảng mạch in mà bạn muốn thực hiện bằng các phần mềm thông dụng sau :Eagle, Orcad, Protel, Traxmat...v.v.v. (các bạn có thể xem hướng dẫn tự thiết kế mạch in bằng phần mềm Eagle Version4.01 như hướng dẫn ở mục 2.6) .

Giao diện thiết kế mạch in của phần mềm Eagle 4.01 :



Sau khi bạn thiết kế , hãy lưu nó lại trong đĩa mềm và in ra bằng mực laze lên giấy photo thường , mỗi tờ photo giá khoảng 300đ .

Hãy lưu ý: Phải in bằng mực in **Laser** .

Không nên in lên giấy scan vì giấy scan mỏng quá khi ủi sẽ bị cháy , không in lên giấy kiếng vì mực dễ bị lem & rất tốn kém (1500đ cho 1 tờ giấy) .

Nơi in giá sinh viên:Cổng thứ 3 của trường ĐH BÁCH KHOA –Tp HCM nhìn qua bên kia đường

Bước 2:

Đặt tờ giấy có in mạch lên miếng Cu. Sau đó ủi lên tờ giấy(trong khoảng từ 10à 15 phút) nhớ là phải ủi đều từ giữa ra đến các đường viền (bàn ủi phải nóng)

+Sau khoảng thời gian đó tờ giấy có vẻ bị nám lại .Ủi xong thì bỏ bảng Cu vào nước à để khoảng 15 phút trở lên cho lớp giấy mực ra rồi bóc nhẹ lớp giấy ra .

Khi đó mực laze sẽ in lên bảng Cu toàn bộ như mình đã thiết kế mạch

Bước 3:

Rửa mạch bằng bột FeCl_3 pha vào nước , tùy vào lượng bột bạn cho vào (thường là 2 bịch khi bạn muốn làm nhanh và rửa nhiều mạch cùng lúc).

Sau khi rửa mạch thì lớp Cu không được mực laze bảo vệ sẽ bị tan hết để lại những đường mực laze (ở dưới có lớp Cu).

Sau đó bạn hãy rửa sạch bảng mạch bằng nước lạnh thật kỹ lưỡng nhiều lần , rồi dùng giấy nhám chà sơ sơ lên mạch Cu để làm bay hết lớp mực laze , như vậy bạn đã làm được bản mạch mà bạn đã tự thiết kế .

Hãy lưu ý : Làm lần đầu thường hay bị hỏng do ủi không đều , khi đó bạn hãy lấy giấy nhám chà cho bay lớp mực rồi làm lại nhé!

III- LỜI KẾT :

Đây chỉ là những kinh nghiệm của chúng tôi trong quá trình làm mạch ,cho nên sai sót cũng là điều có thể xảy ra .Do đó kính mong cùng bạn đọc góp ý & bổ sung để tập sách này ngày một hoàn thiện hơn .

PHẦN 3:

HƯỚNG DẪN SỬ DỤNG CÁC PHẦN MỀM VI XỬ LÝ

NỘI DUNG

Vấn đề 8:

MỘT SỐ PHẦN MỀM LẬP TRÌNH / SOẠN THẢO ASSEMBLY

Vấn đề 9:

CÁC PHẦN MỀM BIÊN DỊCH FILE .ASM RA FILE .HEX

Vấn đề 10 :

CÁC PHẦN MỀM MÔ PHỎNG VI XỬ LÝ

Vấn đề 11 :

HƯỚNG DẪN SỬ DỤNG Ezdl4.0 CHO CÁC MẠCH NẠP

Vấn đề 8:

MỘT SỐ PHẦN MỀM LẬP TRÌNH - SOẠN THẢO ASSEMBLER

I-GIỚI THIỆU:

Nhân tiện viết tập sách này ,tôi xin giới thiệu cùng các bạn sơ lược về một số phần mềm thông dụng giành cho Vi Xử Lý , các phần mềm mà tôi giới thiệu hiện đang có rất nhiều tại các tiệm bán software , các bạn có thể mua ở đó hoặc lên mạng vào trang web :

“ vixuly.cjp.net ” để download về sử dụng hoàn toàn miễn phí .

Tôi cũng xin nói trước là tất cả những gì mà tôi hướng dẫn ở đây hoàn toàn là tôi đã tải nghiệm bằng thực tế rồi nên các bạn cứ yên tâm mà sử dụng , ngoài ra có một số phần mềm mới khác rất hay nhưng tôi chưa kịp cập nhật kịp thì cũng mong các bạn thông cảm bỏ qua cho tôi .

II-CÁC PHẦN MỀM LẬP TRÌNH- SOẠN THẢO VI XỬ LÝ ASSEMBLER:

1) LẬP TRÌNH ASSEMBLER BẰNG NOTEPAD:

Từ menu Star bạn hãy vào mục Program để mở cửa sổ NOTEPAD như hình sau.

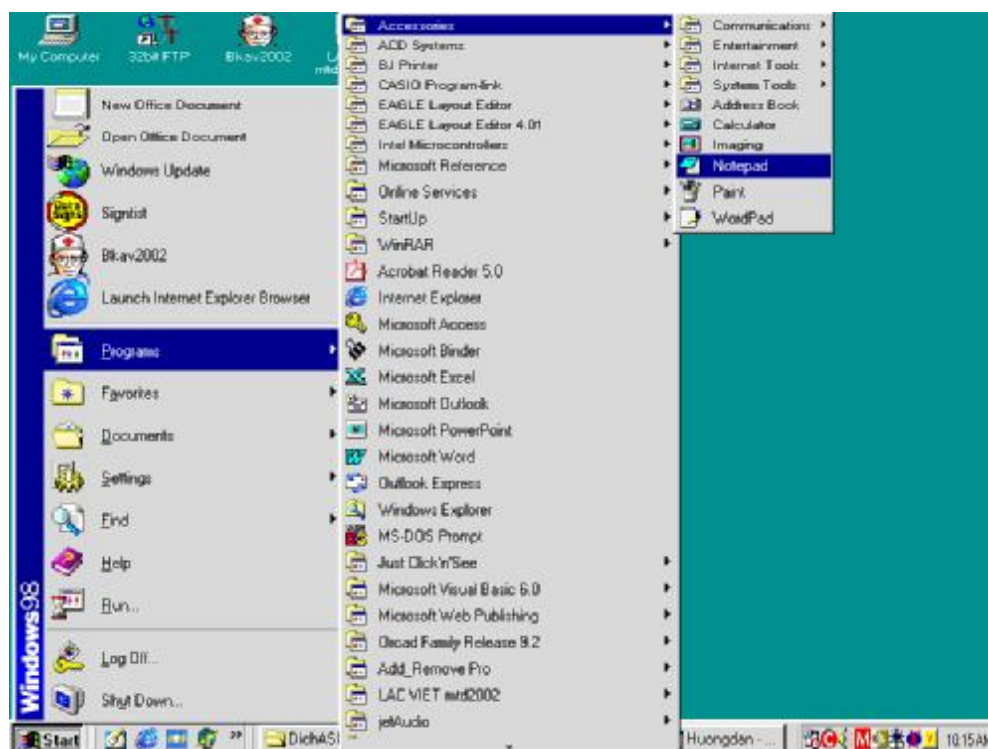
Sau khi bạn đã lập trình xong thì lưu lại với đuôi là “chấm-asm” như sau:

*****.asm**

Lưu ý rằng : Dấu *** là nơi để cho bạn đặt tên chương trình của bạn .

Tên mà bạn đặt dài **tối đa là 8 kí tự** mà thôi .

Ví dụ tên hợp lệ: ledtest.asm , denchop.asm , leduyphi.asm



Hình : Mở cửa sổ NOTEPAD để lập trình ASM

2) LẬP TRÌNH ASSEMBLER BẰNG NC-EDIT (NORTONCOMMANDER):

Ngoài Notepad ra , bạn còn có thể lập trình Assembler bằng chức năng soạn thảo Edit của Norton Commander chạy trên Windows hoặc Dos cũng được .

Để mở chức năng này của NC bạn hãy nhấn hai phím: **“SHIFT + F4”**

Sau đó bạn cũng lưu lại với tên là: *****.ASM**

VẤN ĐỀ 9:

CÁC PHẦN MỀM BIÊN DỊCH FILE .ASM RA FILE .HEX

I-GIỚI THIỆU:

Như chúng ta biết, muốn nạp được nội dung chương trình mà chúng ta đã lập trình vào cho con vi xử lý để nó hoạt động thì chúng ta phải nạp vào cho nó các file có dạng là file .HEX hoặc là file .BIN, nhưng đa số người ta thường nạp cho nó các file có dạng là file .HEX. Do đó tôi xin giới thiệu cùng với các bạn các phần mềm thường dùng để chuyển các file có dạng là file .Asm sang dạng file .Hex đó là ASM51 và TASM.

II_CÁC PHẦN MỀM PHIÊN DỊCH .ASM SANG .HEX:**(A) PHẦN MỀM ASM51:****1-CÁC FILE CẦN CÓ TRONG BỘ ASM51:**

Trong bộ ASM51 gồm có các file sau:

ASM51.EXE	The Cross Assembler program itself
MOD152	Source file for the \$MOD152 control
MOD154	Source file for the \$MOD154 control
MOD252	Source file for the \$MOD252 control
MOD44	Source file for the \$MOD44 control
MOD451	Source file for the \$MOD451 control
MOD452	Source file for the \$MOD452 control
MOD51	Source file for the \$MOD51 control
MOD512	Source file for the \$MOD512 control
MOD515	Source file for the \$MOD515 control
MOD517	Source file for the \$MOD517 control
MOD52	Source file for the \$MOD52 control
MOD521	Source file for the \$MOD521 control
MOD552	Source file for the \$MOD552 control
MOD652	Source file for the \$MOD652 control
MOD751	Source file for the \$MOD751 control
MOD752	Source file for the \$MOD752 control
MOD851	Source file for the \$MOD851 control

Tổng dung lượng : 153Kb

2) CÁCH SỬ DỤNG:

- Sau khi bạn lập trình xong (bằng Notepad hay NC-Edit như đã nói ở trên) bạn hãy lưu nó lại với tên là: *****.asm**, tôi giả sử rằng là : **vidu.asm** và file này bạn lưu trên ổ đĩa C:\ chẳng hạn, đồng thời tôi cũng giả sử rằng bạn cũng để bộ ASM51 này trên ổ đĩa C:\ thì khi đó chúng ta có như sau:

C:\ vidu.asm

ASM51.EXE	The Cross Assembler program itself
MOD152	Source file for the \$MOD152 control
MOD154	Source file for the \$MOD154 control
MOD252	Source file for the \$MOD252 control
MOD44	Source file for the \$MOD44 control
MOD451	Source file for the \$MOD451 control
MOD452	Source file for the \$MOD452 control
MOD51	Source file for the \$MOD51 control
MOD512	Source file for the \$MOD512 control
MOD515	Source file for the \$MOD515 control
MOD517	Source file for the \$MOD517 control
MOD52	Source file for the \$MOD52 control
MOD521	Source file for the \$MOD521 control
MOD552	Source file for the \$MOD552 control
MOD652	Source file for the \$MOD652 control
MOD751	Source file for the \$MOD751 control
MOD752	Source file for the \$MOD752 control

MOD851 Source file for the \$MOD851 control

Sau đó bạn hãy **mở cửa sổ MS-DOS** và gõ dòng lệnh y như sau thì bạn sẽ được một file có tên là: **vidu.hex**, dòng lệnh như sau :

C:\asm51 vidu.asm hoặc C:\asm51 vidu

Như vậy là bạn đã hoàn thành xong công việc biên dịch file >ASM sang file >HEX rồi , bây giờ bạn có thể lấy file này nạp vào cho con chip vi xử lý của bạn được rồi đấy .

3) Phần Mềm Này Kiểm Ở Đâu :

Các bạn có thể vào trang Web của tôi để tải về sử dụng miễn phí hoàn toàn : <http://vixuly.cjp.net> hay bạn cũng có thể liên hệ tại phòng **Photo B1 (dưới cầu thang)** trường ĐH.Bách Khoa Tp.HCM tôi có chép sẵn trong đĩa mềm 1,44Mb.

B) PHẦN MỀM TASM

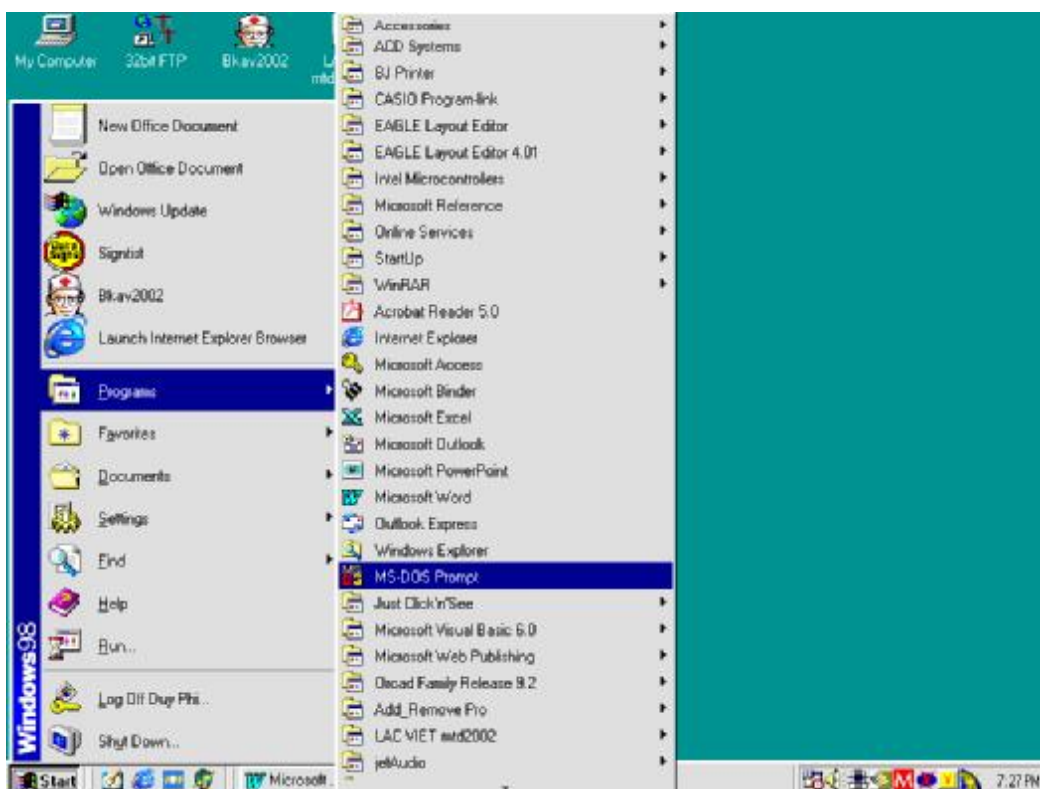
Tiếp theo tôi xin giới thiệu cùng các bạn một phần mềm khác cũng biên dịch file .ASM sang file.HEX rất hay khác đó là phần mềm : **TASM**

1) Nội Dung Các File Có Trong Bộ TASM

Bộ Tasm chứa trong một file có tên là Tasminst có nghĩa là Tasm instal (tệp tin để cài đặt) bạn click chuột lên tệp tin này thì tự động nó sẽ tự giải nén và cài đặt luôn dung lượng của tệp tin Tasminst là 126Kb , các bạn có thể vào trang Web của tôi để tải về sử dụng miễn phí hoàn toàn : <http://vixuly.cjp.net> hay bạn cũng có thể liên hệ tại phòng **Photo B1 (dưới cầu thang)** trường ĐH.Bách Khoa Tp.HCM

2) CÁCH SỬ DỤNG :

Bạn hãy thoát khỏi windows –ra môi trường DOS bằng cách từ môi trường Windows98 , bạn click chuột vào Star menu à chọn mục Program à rồi chọn tiếp **MS-DOS prompt** , như vậy là bạn đã **ra ngoài môi trường DOS** rồi đó .

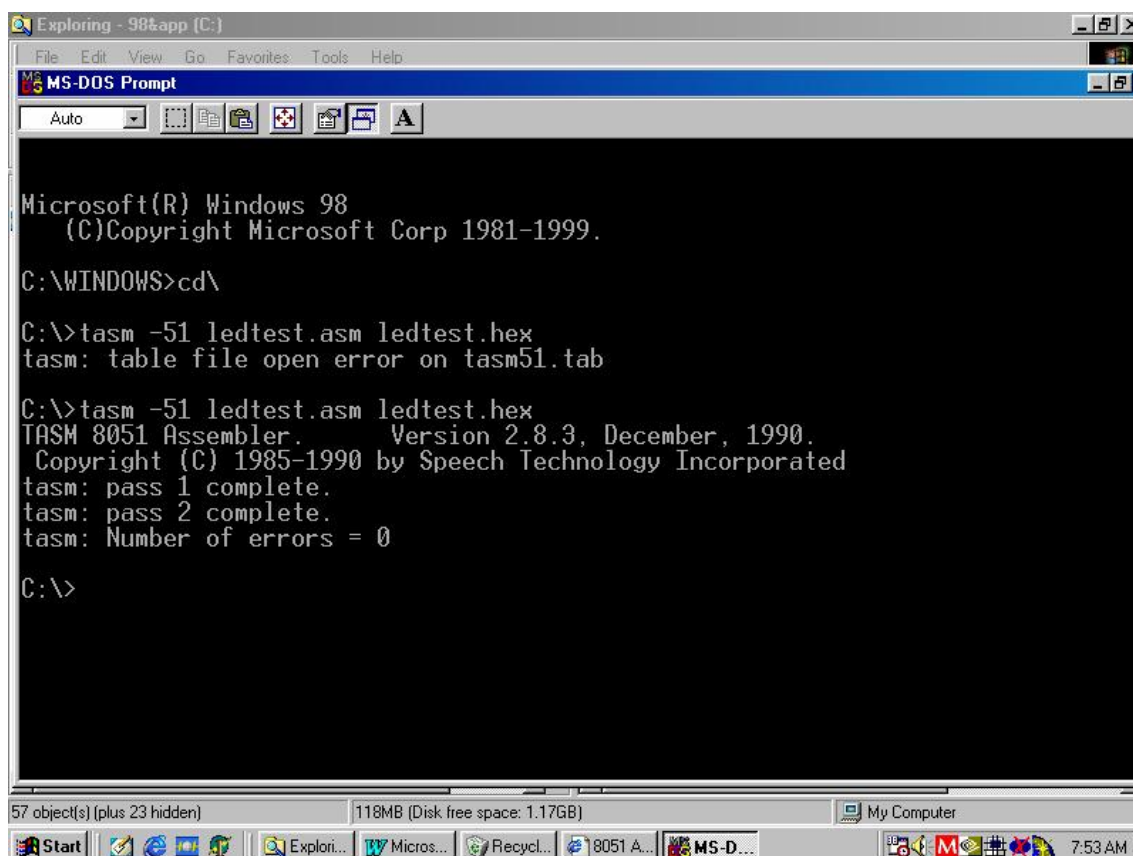


Sau đó bạn hãy thay đổi lại đường dẫn , cùng thư mục để đến nơi chứa file TASM để bắt đầu biên dịch bằng cách sử dụng dòng lệnh sau :

```
tasm -51 *.asm *.hex
```

Tôi giả sử , bạn để file phần mềm TASM và file ledtest.asm tại ổ đĩa C:\ chẳng hạn , thì khi đó bạn sẽ gõ đúng dòng lệnh sau để biên dịch file ledtest.asm :

```
C:\ tasm -51 ledtest.asm ledtest.hex
```



Kết quả của việc làm này bạn sẽ tạo ra được file : **ledtest.hex**.

Xong , bây giờ thì bạn hãy đóng cửa sổ: DOS prompt để trở lại với windows.

Hãy lưu ý rằng : Bạn cũng có thể tạo ra tệp tin kiểm tra lỗi lập trình trước khi tạo ra tệp tin nạp cho vi xử lý , bằng cách bạn tạo ra file có đuôi là **.Lst** , tệp tin này sẽ kiểm tra lỗi cú pháp lệnh trong chương trình của bạn , nó sẽ chỉ cho bạn thấy những lỗi sai để bạn kịp thời sửa chữa .Khi đó bạn hãy gõ câu lệnh sau:

```
tasm -51 *.asm *.obj
```

Kết quả ta sẽ được 2 tệp tin :*.Obj và *.Lst . Cũng với giả sử như trên ta sẽ gõ như sau :

```
C:\ tasm -51 ledtest.asm ledtest.obj
```

Sau đó bạn được hai tệp tin : ledtest.obj và ledtest.Lst bạn hãy mở file Ledtest.Lst này bằng Notepad hay Word để xem kết quả kiểm tra.!

Ngoài ra phần mềm Tasm này còn hỗ trợ cho một số loại Vi Xử Lý khác nữa , khi đó tùy theo loại mà bạn sẽ gõ các dòng lệnh cho phù hợp , tôi xin chép nguyên bản tiếng anh như sau :

TASM - Table Driven Assembler Version 2.8

assemble the code in a file called source.asm, one would enter:

tasm -48	source.asm	for an 8048 assembly
tasm -65	source.asm	for a 6502 assembly
tasm -51	source.asm	for an 8051 assembly.
tasm -85	source.asm	for an 8085 assembly.
tasm -80	source.asm	for a Z80 assembly.
tasm -05	source.asm	for a 6805 assembly.
tasm -68	source.asm	for a 6800/6801 assembly.
tasm -70	source.asm	for a TMS7000 assembly.
tasm -32	source.asm	for a TMS32010 assembly.

The file name that the tables are read from is formed by taking the digits specified after the '-' and appending it to 'TASM' then appending the '.TAB' extension. Thus, the '-48' flag would cause the tables to be read from the file 'TASM48.TAB'.

It is possible to designate tables by non numeric part numbers if the '-t' flag is used. For example, if a user built a table called TASMf8.TAB then TASM could be invoked as follows:

```
tasm -tf8 source.asm
```

Each option flag must be preceded by a dash. Options need not precede the file names, however. The various options are described below:

C- MỘT PHẦN MỀM BIÊN DỊCH KHÁC :

Trong đĩa SoftWare cơ điện 2003 của bộ môn cơ điện tử tôi thấy có giới thiệu một bộ dịch –nhưng không biết là tên gì , nhưng có đầy đủ các tệp tin cần dùng trong thư mục

DichASM , các bạn có thể mua và sử dụng nó :

Cách sử dụng :

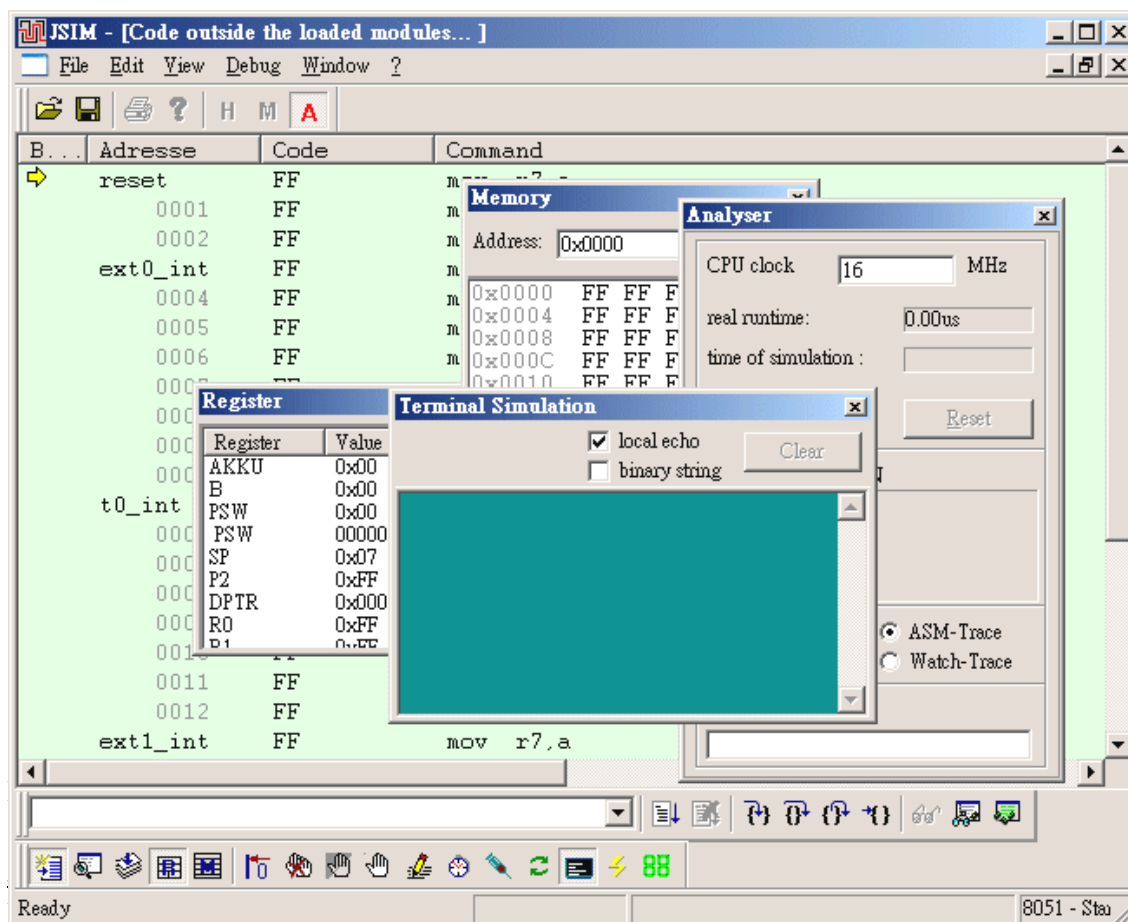
Day la mot so phan mem de bien dich asm sang hex hoac bin
 Cho ho ic intel. Chay tren moi truong dos
 Cac ban phai su dung ket hop cac phan mem voi nhau.
 1/ ban lap trinh assembler bang notepad hoac nc-edit (nhan shift +f4).
 2/ sau do luu lai thanh file : *.asm
 3/ su dung "x8051" de kiem tra loi va bien dich thanh : *.obj
 4/ su dung "link" de bien dich *.obj thanh *.hex
 5/ nhu vay ban da co file *.hex de nap vao ic hay rom.
 Neu may nap cua ban can file *.bin thi ban dung "hexbin2" de chuyen doi.

Ngoài ra còn phần mềm Assemblers vs dung lượng chỉ 40 kb và nhiều phần mềm biên dịch rất hay khác tôi sẽ giới thiệu cùng các bạn trong tập 2” Hướng dẫn giao tiếp vi xử lý với máy tính “ , xin mời các bạn đón đọc!

**Xin chuộc càc bảin
vảin sõi thaønh
côang !**

VẤN ĐỀ 10:**CÁC PHẦN MỀM MÔ PHỎNG VI XỬ LÝ**

1) Phần mềm mô phỏng 8051 [JSIM-51](#) miễn phí hoàn toàn bởi [Jens Altmann](#).

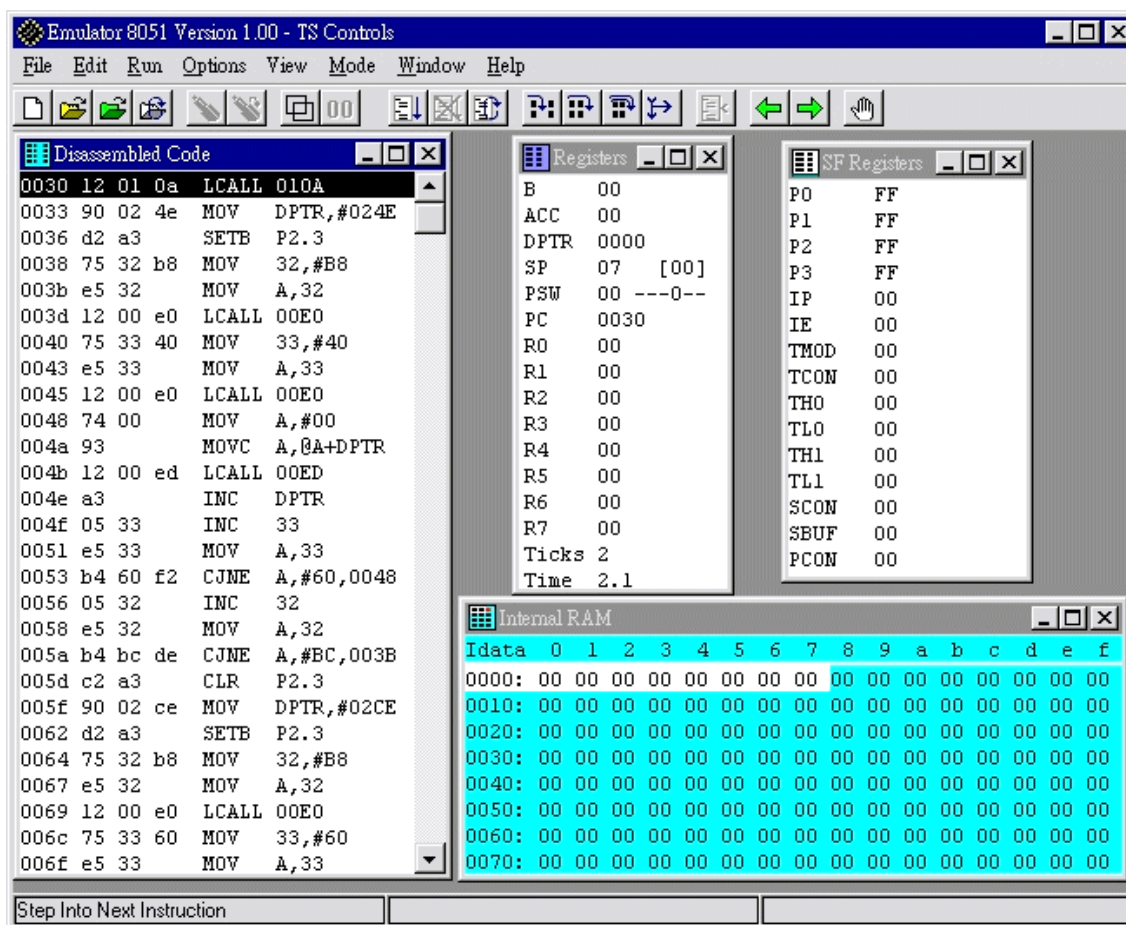


Hiện nay nó có nhiều trên các đĩa Software điện tử, các bạn có thể mua nó, thông thường nó có tên là Ts_Contols 8051.

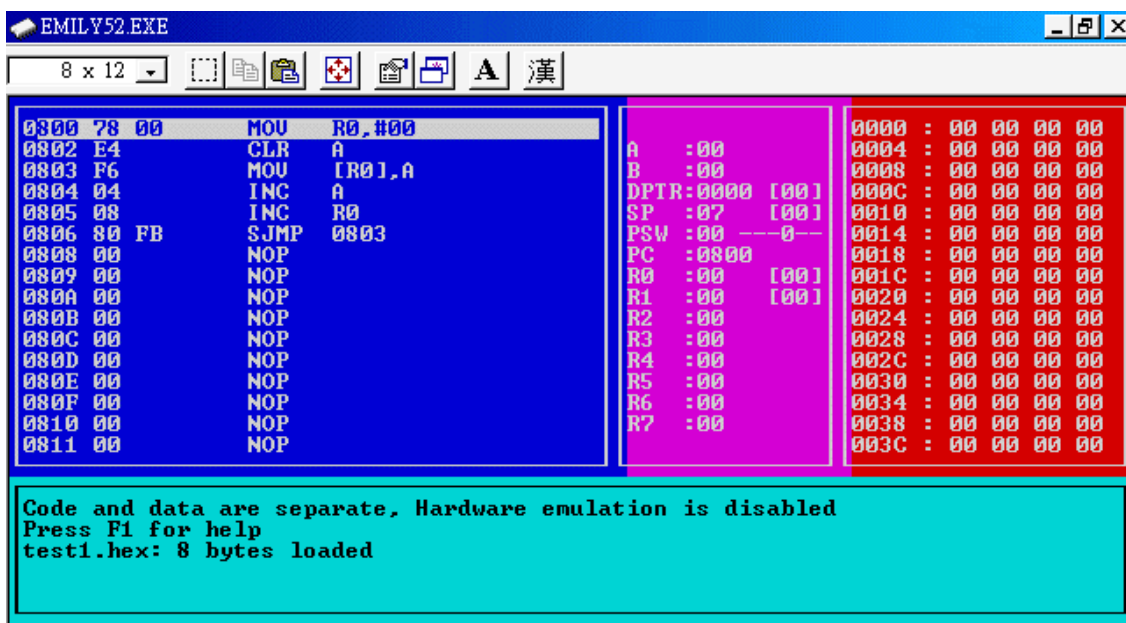
Dung lượng: 963kb

Lưu ý : Một số phần mềm lớn tôi không thể đưa lên mạng , và một số phần mềm đã có bán ở các tiệm Software thì tôi cũng chỉ giới thiệu , các bạn hãy chịu khó mua , hoặc có thể liên hệ riêng với tôi nếu được tôi sẽ gửi cho các bạn !

Giao diện của phần mềm này như sau :



3) Ngoài ra còn một phần mềm mô phỏng khác chạy trên môi trường Dos như sau:



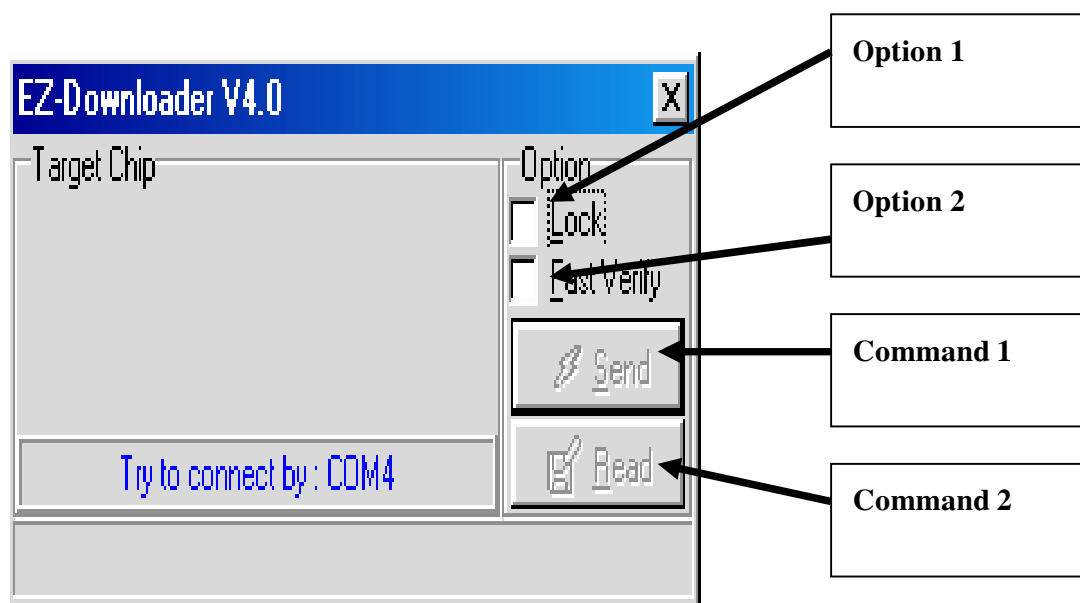
[Emily 8051/8052 V1.7](#) của [Dunfield Development Systems](#) là một phần mềm mô phỏng cho 8051 for DOS.

Dung lượng : 357 kb

Vấn đề 11:

HƯỚNG DẪN SỬ DỤNG PHẦN MỀM EZ4.0 CHO MẠCH NẠP VI XỬ LÝ

I-Giao diện của EZ4.0 như sau:



II - CÁCH SỬ DỤNG:

Option 1: Khi bạn nút này thì con chip của bạn nó sẽ khoá lại , sau này bạn không thể nào nạp lại được nữa do đó rất ít ai chọn option này , theo tôi là không chọn thì tốt hơn để có thể nạp nhiều lần !

Option 2: Khi bạn nút này việc kiểm tra sau khi nạp xong sẽ diễn ra rất nhanh , bạn có chọn hay không tùy ý chẳng hề hấn gì cả !

Command 1: Chức năng của Command này là nạp chương trình file.Hex vào cho vi xử Lý .Bạn hãy nhấn vào command này để chỉ đến nơi chứa file ,Hex cần nạp để tiến hành quá trình nạp .

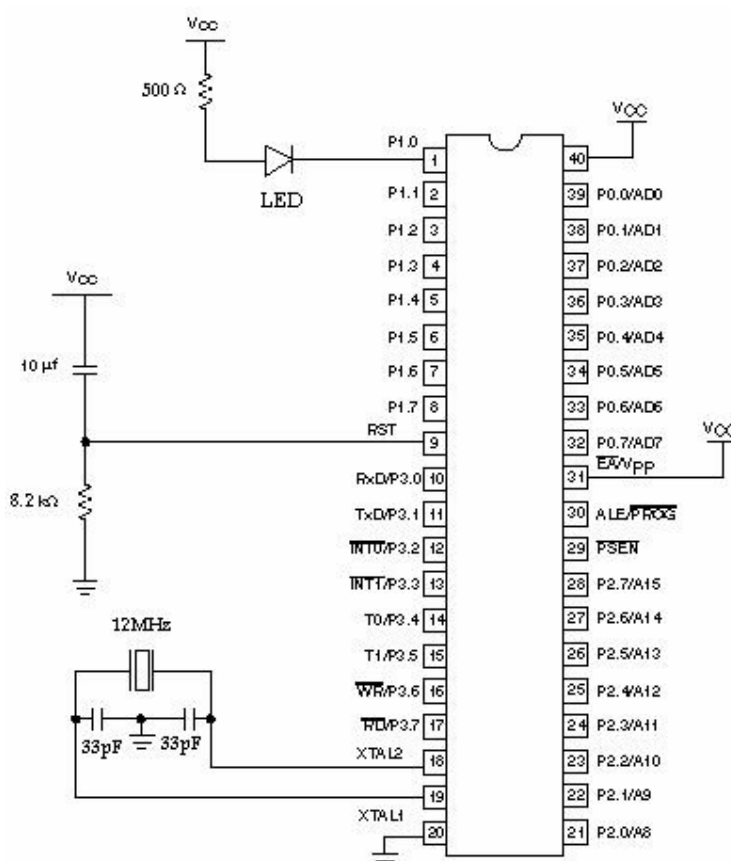
Lưu ý là : Khi bạn chọn Command này thì điều đầu tiên nó sẽ xoá nội dung cũ trên con vi xử lý nếu có , để dọn đường chuẩn bị nạp nội dung mới vào , công việc này được nó làm tự động hoàn toàn .

Command 2: Chức năng này sẽ đọc ngược nội dung có trong con vi xử lý của bạn ra ngoài lại dưới dạng file .Hex .

Hãy nhớ rằng : Phần mềm này dùng chung cho cả hai mạch nạp AT89C51 và At89C2051 , việc nhận dạng sẽ được nó làm tự động hoàn toàn .

PHẦN 4:**CÁC MẠCH ĐIỆN VI XỬ LÝ - LÝ THỨ- ĐƠN GIẢN
CHO NGƯỜI MỚI BẮT ĐẦU HỌC LẬP TRÌNH CÙNG VI XỬ LÝ****Vấn đề 12 – MẠCH 1:****Hãy Làm Một Đèn Led Chớp Với- 8951****Bước 1)**

Bước đầu tiên là làm mạch điện . Tại thời điểm này bạn phải làm quen với việc sử dụng các linh kiện .(Ở đây có 2 điện trở , 3 cái tụ , 1 thạch anh 12Mhz và 1 đèn Led , tất cả đều có bán ở chợ Nhật Tảo cả). Bạn có thể đặt các linh kiện này lên trên một breakboard rồi dùng dây điện nối chúng lại (breakboard bày bán rất nhiều ở chợ Nhật Tảo bạn hãy mua một cái để làm các thí nghiệm !) . Sự thiết kế này dùng với con vi điều khiển 89c51 .



Điều duy nhất mà chúng tôi muốn làm mạch này là :Làm cho Led chớp . Bằng việc làm này bạn sẽ có thể được học một chương trình xử lý được viết bằng ngôn ngữ Assembly và sau đó bạn sẽ biết cách nạp file chương trình vào cho con Chip vi điều khiển .

Lần Đầu tiên này chúng tôi sẵn sàng viết sẵn các dòng code cho bạn bằng ngôn ngữ Assembly , file này sẽ có tên là : **ledtest.asm** , chúng ta sẽ sử dụng nó cho mạch này .

Mã code như sau :

```

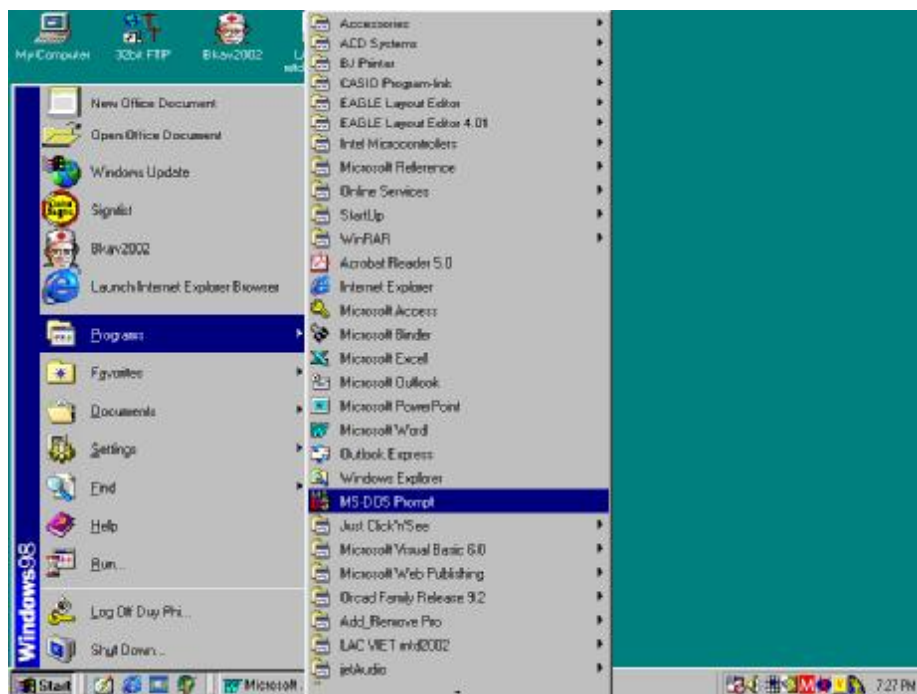
ORG 0030H
LOOP:
    CLR  P1.0      ;P1.0 SANG
    CALL DELAY_1S
    SETB P1.0      ;P1.0 TAT
    CALL DELAY_1S
    LJMP LOOP
DELAY_1S:
    MOV  R1,#20
    LOOP1:  MOV  R2,#200
    LOOP2:  MOV  R3,#255
    LOOP3:  DJNZ R3,LOOP3
    DJNZ R2,LOOP2
    DJNZ R1,LOOP1
    RET
END

```

Hãy nhớ rằng : Dòng Code nào bắt đầu bằng dấu “ ; ” thì đó là dòng chú thích , do đó chúng ta có thể bỏ qua mà không ảnh hưởng gì đến chương trình cả.
Sau khi viết xong bạn hãy lưu lại với tên là : ledtest.asm

BƯỚC 2: BIÊN DỊCH CHƯƠNG TRÌNH

Hãy MOVE file ledtest.asm này đến thư mục – nơi mà bạn đã chứa phần mềm TASM . Tiếp theo bạn hãy thoát khỏi windows –ra môi trường DOS bằng cách từ môi trường Windows98 , bạn click chuột vào Star menu **à** chọn mục Program **à** rồi chọn tiếp **MS-DOS prompt** , như vậy là bạn đã **ra ngoài môi trường DOS** rồi đó .

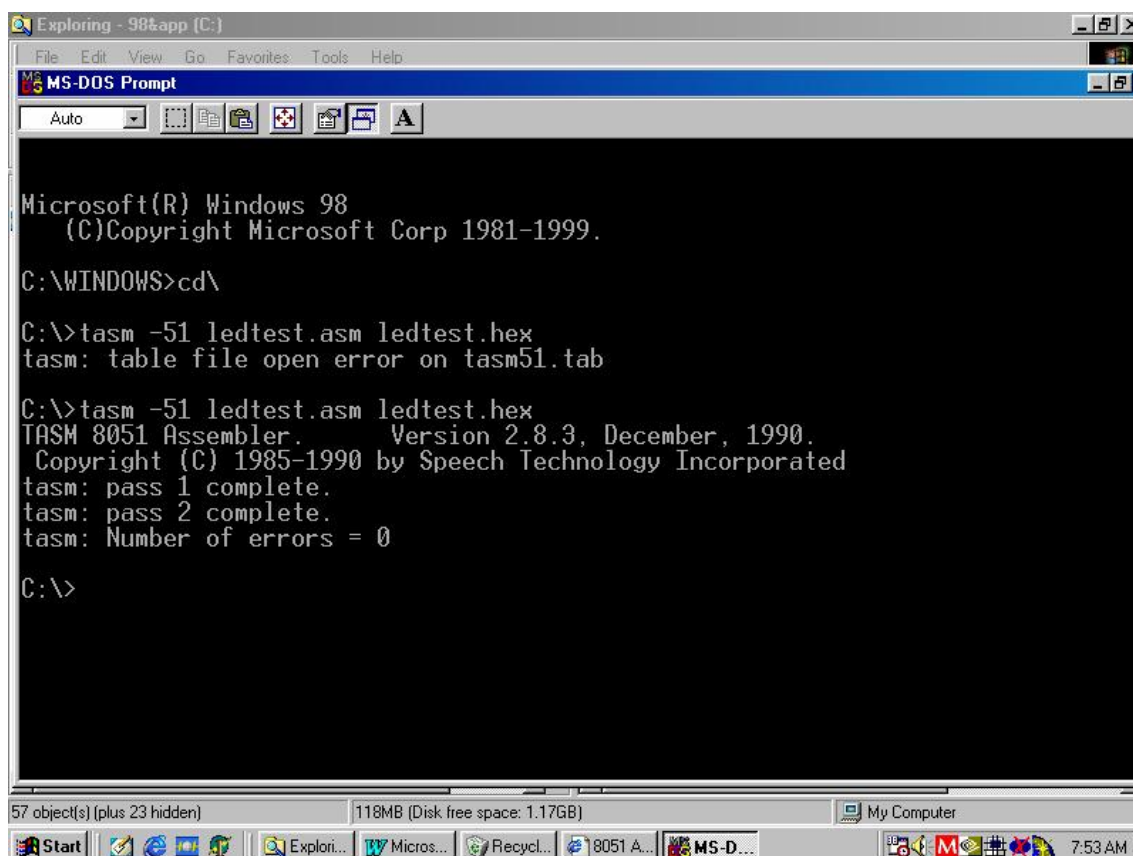


Sau đó bạn hãy thay đổi lại đường dẫn , cùng thư mục để đến nơi chứa file TASM để bắt đầu biên dịch bằng cách sử dụng dòng lệnh sau :

```
tasm -51 ledtest.asm ledtest.hex
```

Tôi giả sử , bạn để file phần mềm TASM và file ledtest.asm tại ổ đĩa C:\ chẳng hạn , thì khi đó bạn sẽ gõ đúng dòng lệnh sau để biên dịch file ledtest.asm :

```
C:\ tasm -51 ledtest.asm ledtest.hex
```



```
Exploring - 98&app (C:)
File Edit View Go Favorites Tools Help
MS-DOS Prompt
Auto
Microsoft(R) Windows 98
(C)Copyright Microsoft Corp 1981-1999.
C:\WINDOWS>cd\
C:\>tasm -51 ledtest.asm ledtest.hex
tasm: table file open error on tasm51.tab
C:\>tasm -51 ledtest.asm ledtest.hex
TASM 8051 Assembler.      Version 2.8.3, December, 1990.
Copyright (C) 1985-1990 by Speech Technology Incorporated
tasm: pass 1 complete.
tasm: pass 2 complete.
tasm: Number of errors = 0
C:\>
```

Kết quả của việc làm này bạn sẽ tạo ra được file : **ledtest.hex**.

Xong , bây giờ thì bạn hãy đóng cửa sổ: DOS prompt để trở lại với windows.

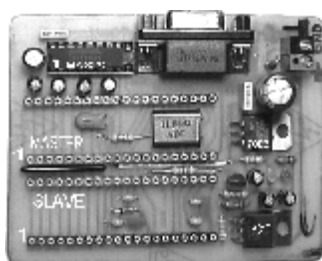
Bước 3) Nạp chương trình vào cho con Chip Vi Xử Lý

Bây giờ bạn hãy đem “**Mạch Nạp AT89C51**” ra, nạp file ledtest.hex vào cho con vi xử lý .

Lưu ý : Hãy làm cẩn thận mọi thứ như nguồn cung cấp và dây nối cổng COM

Bạn hãy đặt con chip Vi Xử Lý 89c51 cần nạp vào **Socket – Chip Slave**

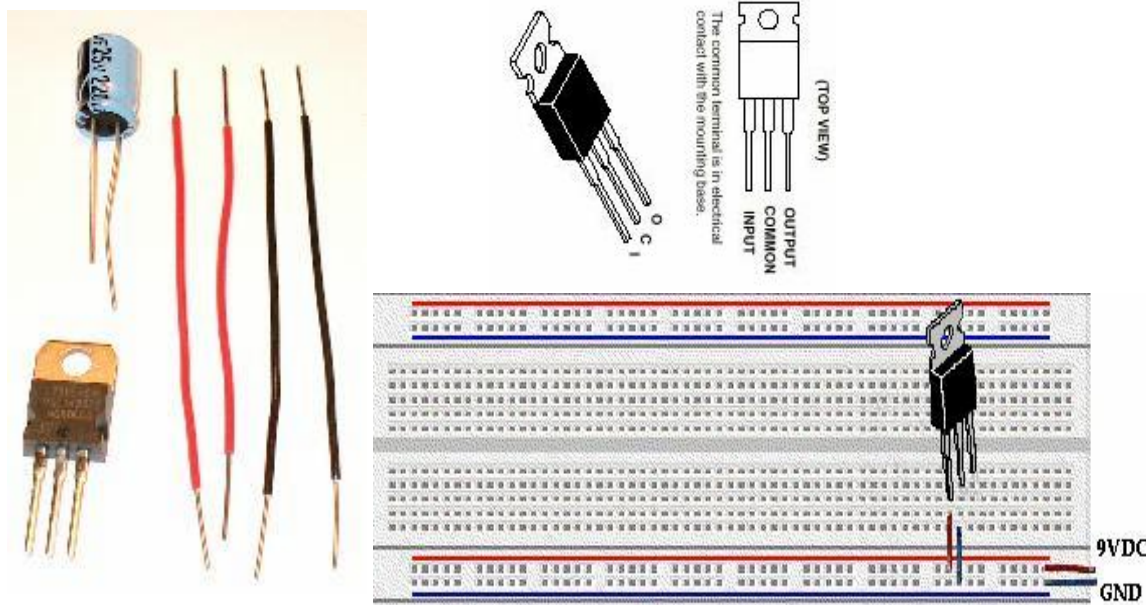
Hãy nhớ : đặt đúng chân- hãy nhìn kỹ số 1 trên hình đấy nhé, tôi cũng đã lần đặt lộn rồi đó !).



Tiếp đến , bạn mở phần mềm **EZ4.0** , phần mềm này nó sẽ tự nhận biết sự hiện diện “**Mạch Nạp AT89c51**” , rồi sau đó bạn hãy đổi đường dẫn để chỉ đến nơi chứa file **lesdtest.hex** để nạp vào cho 89c51. Tất cả mọi việc sau đó sẽ được làm tự động .

Sau khi nạp xong , bạn hãy tháo con chip ra , gắn nó vào breakboard mà hồi nãy ở ” Bước 1” bạn lắp ráp , rồi cung cấp điện cho mạch (hãy nhớ là 5V thôi nhé !). Xong bạn hãy xem hiện tượng của đèn Led như thế nào nhé ! Vâng !Rất đẹp !

Bộ nguồn 5V bạn có thể lắp ráp như sau :



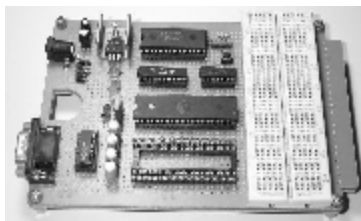
Như bạn thấy phía góc phải của breakboard là ngõ vào 9VDC .
Đến đây xem như đã xong bài thực hành đầu tiên :

**Chúc Các Bạn Làm Ngay
Được Liền Và Luôn Vui Vẻ Với
Các Mạch Điện Sau !**

Lê Duy phi

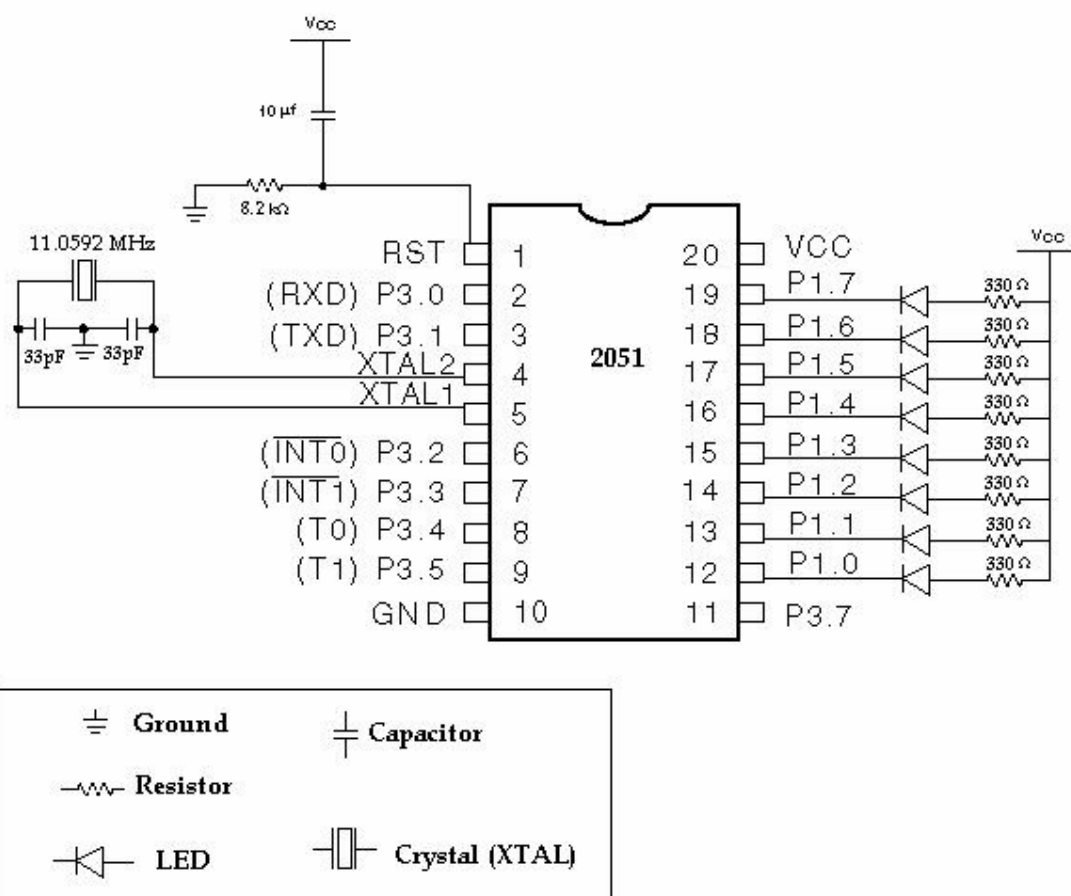
Vấn đề 13 - Mạch 2:

BỘ ĐIỀU KHIỂN NHÚNG ĐƠN GIẢN



Bước 1: Bước đầu tiên là xây dựng mạch điện

Ở đây bạn cần làm quen với một số linh kiện điện tử mà chúng ta sẽ dùng đến. Bạn có thể lắp đặt chúng lên trên một breadboard rồi dùng dây điện nối lại để tạo thành mạch điện như sau nhé. Mạch này được thiết kế dành cho con chip 89C2051-(loại 20 chân Đa số các bộ vi xử lý thì không thể nào điều khiển trực tiếp các Led ngoại trừ họ ATMEL mà tiêu biểu là 89c2051 và 89c51, đó là lý do mà tôi có thể thiết kế được mạch này hết sức đơn giản như bạn có thể thấy trong sơ đồ sau :

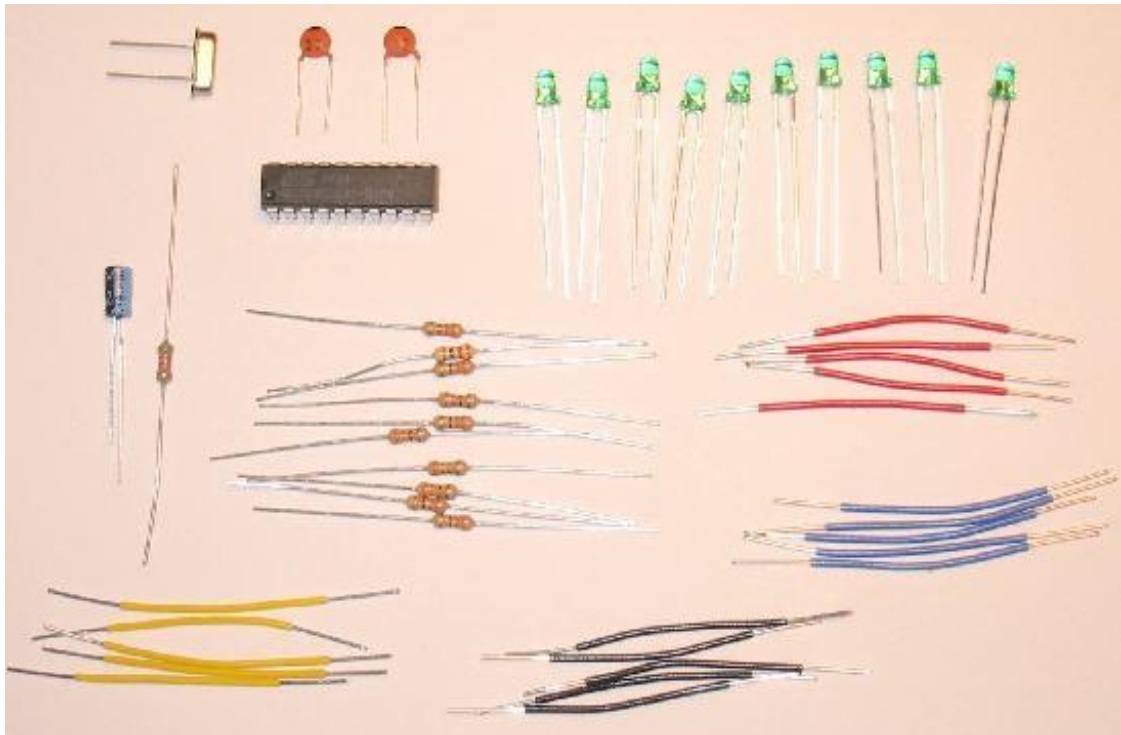


Lưu ý : Nguồn cung cấp cho mạch là **Vcc=5V** và **GND=0 V**

Các linh kiện cần dùng cho mạch này là :

- 1- AT89C2051
- 1-thạch anh 11.0592 Mhz
- 2-tụ 33p
- 1-tụ 10mF
- 1- điện trở 8k2 và 10 con 330 ohm
- 10-dây điện để nối mạch .

Đây là hình ảnh của các linh kiện như đã nói ở trên :



Mạch này sẽ làm cho 8 Led lần lượt sáng lên bắt đầu từ chân P1.0 đến P1.7 với nhiều kiểu đếm rất hay và rất lý thú sẽ làm bạn hài lòng và thích thú nó .

Sau đây là chương trình của mạch điện viết bằng Assembly:

CHƯƠNG TRÌNH VERSION 1 : SÁNG DẦN –TẮT DẦN

```

ORG 0030H
LOOP:  MOV     P1,#0FFH
        CLR    P1.7      ;P1.7 SANG
        CALL  DELAY_1S
        SETB   P1.7      ;P1.7 TAT
        CLR    P1.6
        CALL  DELAY_1S
        SETB   P1.6
        CLR    P1.5
        CALL  DELAY_1S
        SETB   P1.5
        CLR    P1.4
        CALL  DELAY_1S
    
```

```

    SETB P1.4
    CLR  P1.3
    CALL DELAY_1S
    SETB P1.3
    CLR  P1.2
    CALL DELAY_1S
    SETB P1.2
    CLR  P1.1
    CALL DELAY_1S
    SETB P1.1
    CLR  P1.0
    CALL DELAY_1S
    SETB P1.0

    CLR  P1.0      ;nguoc lai
    CALL DELAY_1S
    SETB P1.0
    CLR  P1.1
    CALL DELAY_1S
    SETB P1.1
    CLR  P1.2
    CALL DELAY_1S
    SETB P1.2
    CLR  P1.3
    CALL DELAY_1S
    SETB P1.3
    CLR  P1.4
    CALL DELAY_1S
    SETB P1.4
    CLR  P1.5
    CALL DELAY_1S
    SETB P1.5
    CLR  P1.6
    CALL DELAY_1S
    SETB P1.6
    CLR  P1.7
    CALL DELAY_1S
    SETB P1.7
    LJMP LOOP
DELAY_1S:
    MOV  R1,#20
LOOP1:   MOV  R2,#200
LOOP2:   MOV  R3,#255
LOOP3:   DJNZ R3,LOOP3
         DJNZ R2,LOOP2
         DJNZ R1,LOOP1
         RET
END

```

CHƯƠNG TRÌNH VERSION 2: SÁNG DẪN –THAY ĐỔI KIỂU RỒI TẮT

```

;*****
;*
;*                               LED Blinker
;*****
;
; RESET                               ;reset routine

        ORG      0H                               ;locate routine at 00H
        AJMP     START                           ;jump to START
;
;*****
;
INITIALIZE:                               ;set up control registers
;
        MOV      TCON,#00H
        MOV      TMOD,#00H
        MOV      PSW,#00H
        MOV      IE,#00H                       ;disable interrupts
        RET
;
;*****
;
;      Real code starts below. The first two routines are for delays so we
;      can slow down the blinking so we can see it. (Without a delay, it
;      would blink so fast it would look like it was always on.
;
;*****
;
DELAYMS:                               ;millisecond delay routine
;
        MOV      R7,#00H                       ;put value of 0 in register R7
LOOPA:
        INC      R7                             ;increase R7 by one (R7 = R7 +1)
        MOV      A,R7                         ;move value in R7 to
                                                Accumulator (also known as A)
        CJNE     A,#0FFH,LOOPA                 ;compare A to FF hex (256). If
                                                ;not equal go to LOOPA
        RET                                     ;return to the point that this
                                                ;routine was called from
;*****
;
DELAYHS:                               ;half second delay above
;millisecond delay
        MOV      R6,#00H                       ;put 0 in register R6 (R6 = 0)
LOOPB:
        INC      R6                             ;increase R6 by one (R6 = R6 +1)
        ACALL    DELAYMS                       ;call the routine above. It
will run and return to here.
        MOV      A,R6                         ;move value in R6 to A
        JNZ      LOOPB                         ;if A is not 0, go to LOOPB
        RET
;

```

```

;*****
START:                                ;main program (on power up, ;
                                      ;program starts at this point)
      ACALL INITIALIZE                ;set up control registers
LOOP:
      INC P1                          ;INCrement P1. Adds 1 to the 8
                                      ;bit register for Port 1.
      ACALL DELAYHS                   ;go to above routine that
                                      ;causes a delay
      AJMP LOOP                       ;go to LOOP(jump back to point
                                      ;labeled LOOP)
END                                   ;end program

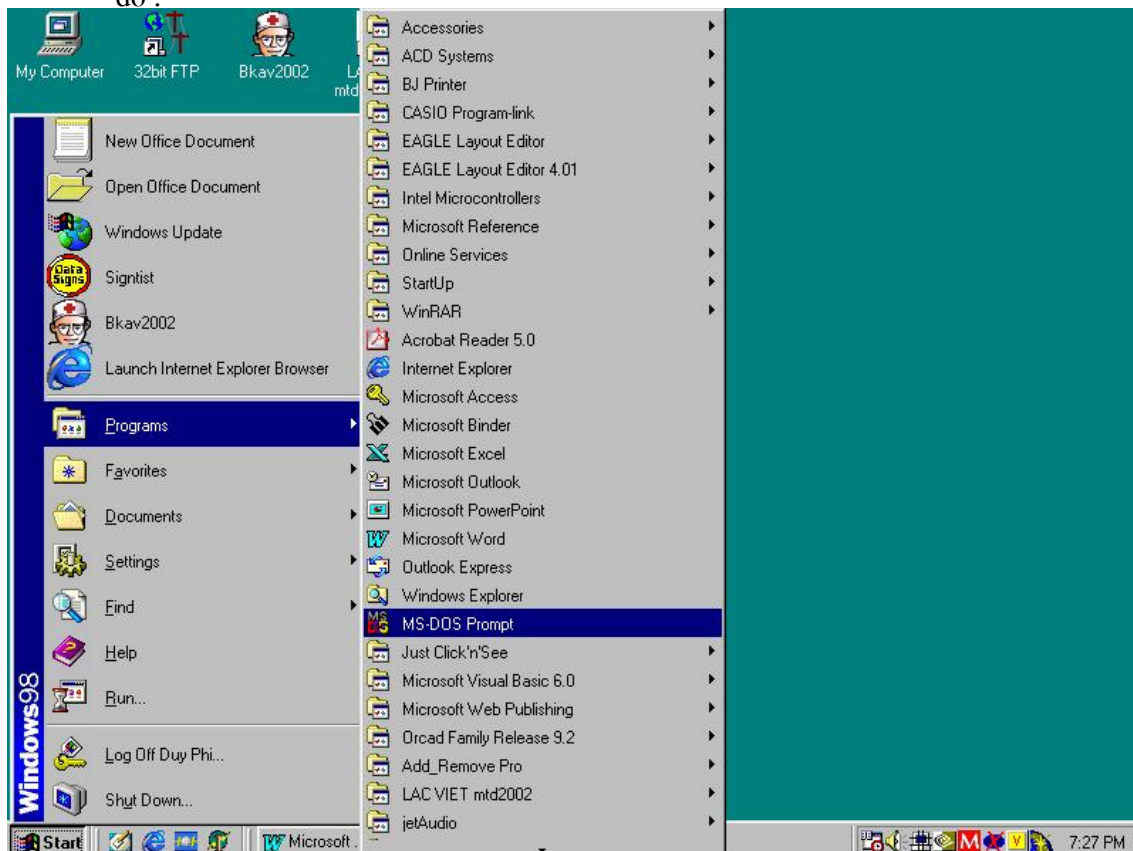
```

Hãy nhớ rằng : Dòng Code nào bắt đầu bằng dấu “ ; ” thì đó là dòng chú thích , do đó chúng ta có thể bỏ qua mà không ảnh hưởng gì đến chương trình cả.

Sau khi viết xong bạn hãy lưu lại với tên là : ledproj.asm

BƯỚC 2 : BIÊN DỊCH CHƯƠNG TRÌNH

Hãy MOVE file ledproj2.asm này đến thư mục – nơi mà bạn đã chứa phần mềm TASM . Tiếp theo bạn hãy thoát khỏi windows –ra môi trường DOS bằng cách từ môi trường Windows98 , bạn click chuột vào Star menu à chọn mục Program à rồi chọn tiếp **MS-DOS prompt** , như vậy là bạn đã ra ngoài môi trường DOS rồi đó .



Sau đó bạn hãy thay đổi lại đường dẫn , cùng thư mục để đến nơi chứa file TASM để bắt đầu biên dịch bằng cách sử dụng dòng lệnh sau :

```
tasm -51 ledproj2.asm ledproj2.hex
```

Tôi giả sử , bạn để file phần mềm TASM và file ledproj2.asm tại ổ đĩa C:\ chẳng hạn , thì khi đó bạn sẽ gõ đúng dòng lệnh sau để biên dịch file ledproj2.asm :

```
C:\ tasm -51 ledproj2.asm ledproj2.hex
```

Kết quả của việc làm này bạn sẽ tạo ra được file : **ledproj2.hex**.

Xong , bây giờ thì bạn hãy đóng cửa sổ: DOS prompt để trở lại với windows.

Bước 3) Nạp chương trình vào cho con Chip Vi Xử Lý

Bây giờ bạn hãy đem “**Mạch Nạp AT89C2051**” ra, nạp file ledtest.hex vào cho con vi xử lý .

Lưu ý : Hãy làm cẩn thận mọi thứ như nguồn cung cấp và dây nối cổng COM

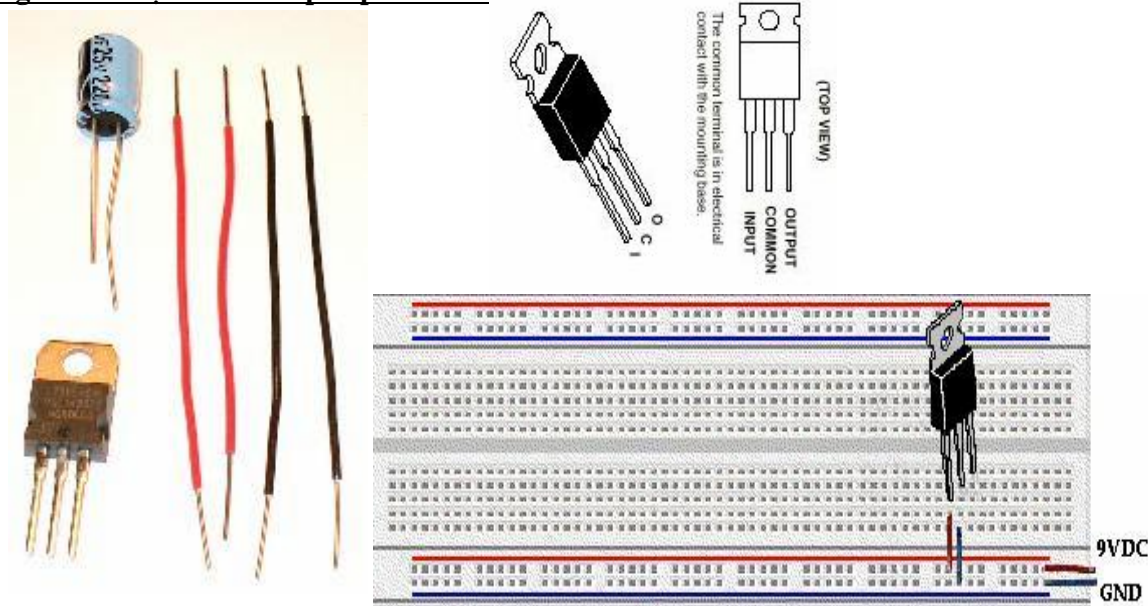
Bạn hãy đặt con chip Vi Xử Lý 89c2051 cần nạp vào **Socket – Chip Slave**

Hãy nhớ : đặt đúng chân- hãy nhìn kỹ số 1 trên hình đấy nhé , tôi cũng đã lần đặt lộn rồi đó !).



Tiếp đến , bạn mở phần mềm **EZ4.0** , phần mềm này nó sẽ tự nhận biết sự hiện diện “**Mạch Nạp AT89c2051**” , rồi sau đó bạn hãy đổi đường dẫn để chỉ đến nơi chứa file **ledtest.hex** để nạp vào cho 89c2051. Tất cả mọi việc sau đó sẽ được làm tự động .

Sau khi nạp xong , bạn hãy tháo con chip ra , gắn nó vào breakboard mà hồi nãy ở ” Bước 1 ” bạn lắp ráp , rồi cung cấp điện cho mạch (hãy nhớ là 5V thôi nhé !). Xong bạn hãy xem hiện tượng của đèn Led như thế nào nhé ! Vâng !Rất đẹp !

Bộ nguồn 5V bạn có thể lắp ráp như sau

Như bạn thấy phía góc phải của breadboard là ngõ vào 9VDC.

Đến đây xem như đã xong bài thực hành thứ 2.

Xin Chuộc Càc Bản Thaønh Công Khi Laøm Maïch Naøy

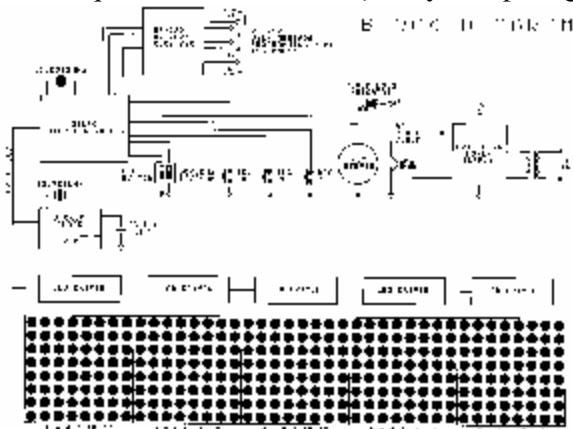
Vaø Toài Hy Voïng Noù Seõ Mang Laïi Cho Bản

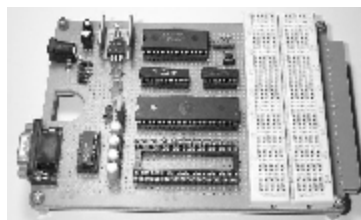
Nhieàu Nieàm Say Me à Mòuì !...

Leà Duy Phi

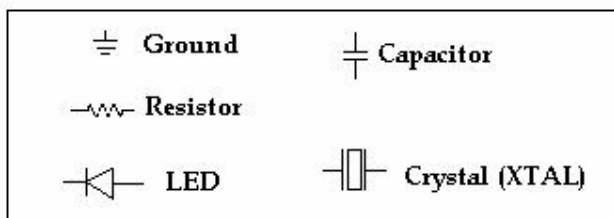
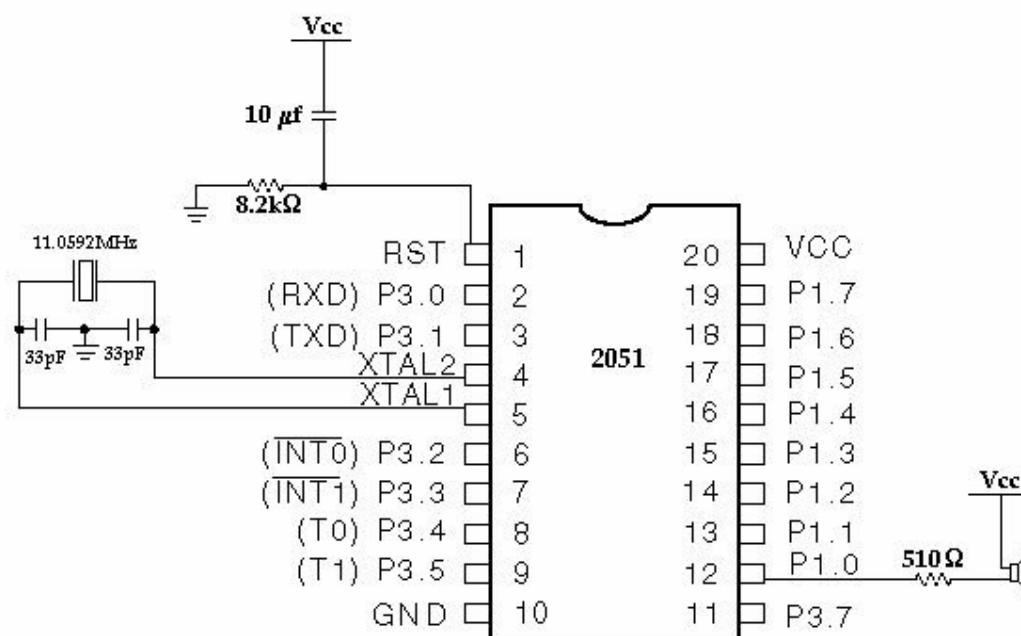
Xin mời các bạn đón đọc ví dụ này trong quyển thứ 2 sắp viết xong: “ **Hướng Dẫn Giao Tiếp Vi Xử Lý Với Máy Tính**” trong quyển này tôi sẽ giúp bạn biết cách giao tiếp giữa vi xử lý và máy tính thông qua phần mềm Visual Basic, đồng thời tôi cũng có đề cập nhiều ví dụ mạch điện rất lý thú như “ **Máy đo nhiệt độ hiển thị trên Led** , **Máy đo nhiệt độ hiển thị trên Máy Tính** , **Làm bảng đèn chữ chạy đơn giản** , **Làm bảng đèn chữ chạy giao tiếp với Máy Tính** , **Mạch đo điện áp giao tiếp máy tính** (đây là đề tài mà tôi đã đạt **giải 3 trong ” cuộc thi sinh viên sáng tạo lần 1 ”** , và những mạch điện vi xử lý khác tôi sẽ giới thiệu sau , xin các bạn đón đọc !

Các tài liệu liên quan của tôi xin các bạn hãy đến phòng Photo B1- đh.Bách Khoa.



Vấn đề 14 - MẠCH 3:**LÀM CÒI HỤ VỚI HAI MỨC ÂM ĐIỆU KHÁC NHAU**

Hãy lưu ý : Rằng , mọi thao tác tôi đã hướng dẫn kỹ ở các ví dụ trên –ở đây tôi chỉ giới thiệu cô đọng các vấn đề mà chủ yếu là sơ đồ và giải thuật cho các bạn mà thôi !

I-SƠ ĐỒ MẠCH:

Lưu ý : Loa mắc nối tiếp với điện trở 500 ohm bạn có thể mua loại loa Buze màu đen –tròn ở chợ Nhật Tảo

II- CHƯƠNG TRÌNH CHO MẠCH ĐIỆN :

```

;*****
;*
;*                               Speaker Sounds - Weird                               *
;*
;*****
;
;                               VARIABLES AND CONSTANTS
;
;*****
;
; RESET                               ;reset routine
;
; ORG 0H                               ;locate routine at 00H
; AJMP START                           ;jump to START
;
;*****
;
INITIALIZE:                           ;set up control registers
;
;   MOV TCON,#00H
;   MOV TMOD,#00H
;   MOV PSW,#00H
;   MOV IE,#00H                       ;disable interrupts
;   RET
;
;*****
;   Real code starts below. The first two routines are for delays.
;*****
;
DELAYMS:                               ;millisecond delay routine
;
;   MOV R7,#00H                       ;put value of 0 in register R7
LOOPA:
;   INC R7                             ;increase R7 by one (R7 = R7 +1)
;   MOV A,R7                           ;move value in R7 to Accumulator (also known
;                                       ;as A)
;   CJNE A,#0FFH,LOOPA                 ;compare A to FF hex (256). If not equal go
;                                       ;to LOOPA
;   RET                               ;return to the point that this routine was
;                                       ;called from
;
;*****
;
DELAYHS:                               ;half second delay above millisecond delay
;
;   MOV R6,#080H                       ;put 0 in register R6 (R6 = 0)
;   MOV R5,#002H                       ;put 2 in register R5 (R5 = 2)
LOOPB:
;   INC R6                             ;increase R6 by one (R6 = R6 +1)
;   ACALL DELAYMS                     ;call the routine above. It will run and
;                                       ;return to here.
;   MOV A,R6                           ;move value in R6 to A
;   JNZ LOOPB                          ;if A is not 0, go to LOOPB
;   DEC R5                             ;decrease R5 by one. (R5 = R5 -1)
;   MOV A,R5                           ;move value in R5 to A
;   JNZ LOOPB                          ;if A is not 0 then go to LOOPB.
;   RET

```

```

;
;*****
;
TONE_ONE:
;
    MOV    R4, #00H
    MOV    R3,#002H                ;put 2 in register R3
LOOPC:
    INC    R4                      ;increase R4 by one
    ACALL  DELAY                   ;Delay
    ACALL  DELAYMS
    DEC    R2                      ; This time decrement R2 to make tone go
                                ;down in frequency

    CPL    P1.0
    MOV    A,R4                   ;move value in R4 to A
    JNZ    LOOPC                  ;if A is not 0, go to LOOPC
    DEC    R3                     ;decrease R3 by one.
    MOV    A,R3                   ;move value in R3 to A
    JNZ    LOOPC                  ;if A is not 0 then go to LOOPC.
    RET

;
;*****
;
TONE_TWO:
;
    MOV    R4, #00H
    MOV    R3,#002H                ;put 2 in register R3
LOOPD:
    INC    R4                      ;increase R4 by one
    ACALL  DELAY                   ;Delay
    ACALL  DELAYMS
    INC    R2                      ; INCrement R2 to make frequency go up
    CPL    P1.0
    MOV    A,R4                   ;move value in R4 to A
    JNZ    LOOPD                  ;if A is not 0, go to LOOPD
    DEC    R3                     ;decrease R3 by one.
    MOV    A,R3                   ;move value in R3 to A
    JNZ    LOOPD                  ;if A is not 0 then go to LOOPD.
    RET

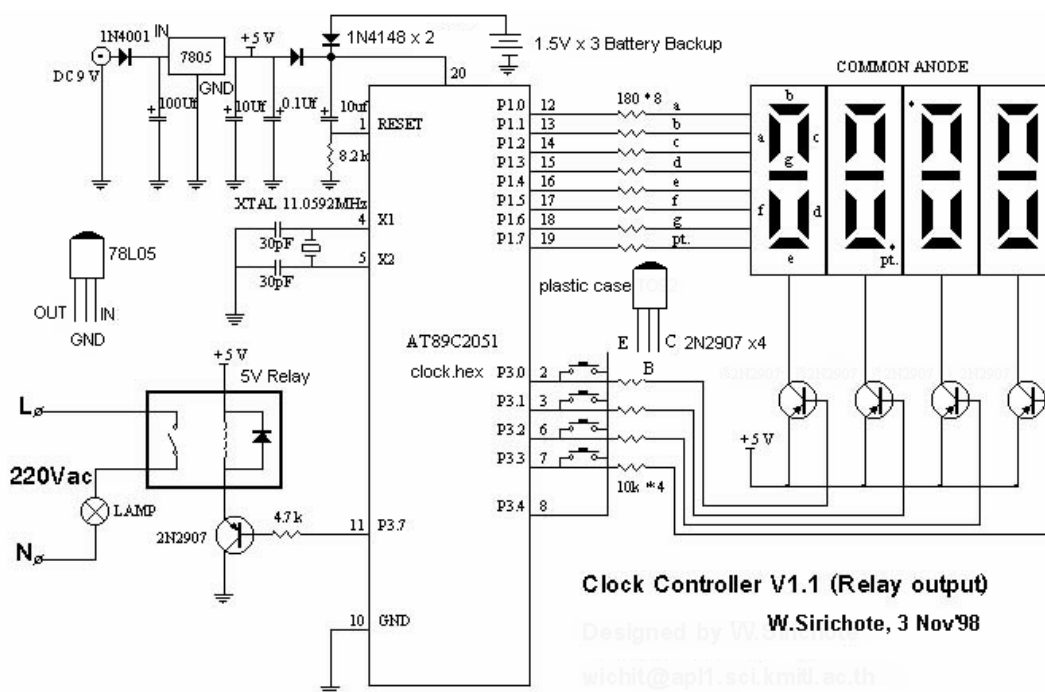
;
;*****
;
DELAY:
;
    MOV    A, R2
    MOV    R7,A                    ;put value of R2 in register R7
LOOPE:
    INC    R7                      ;increase R7 by one (R7 = R7 +1)
    MOV    A,R7                   ;move value in R7 to Accumulator (also known
                                ;as A)
    CJNE   A,#0FFH,LOOPE          ;compare A to FF hex (256). If not equal go
                                ;to LOOPA
    RET                           ;return to the point that this routine was
                                ;called from

;
;*****
;
START:
;main program (on power up, program starts
;at this point)
    ACALL  INITIALIZE             ;set up control registers
LOOP:
    ACALL  TONE_ONE
    ACALL  TONE_TWO
    AJMP   LOOP                   ;go to LOOP(always jump back to point
                                ;labeled LOOP)
END                               ;end program

```

Vấn đề 15 - MACH 4:**HÃY LÀM : ĐỒNG HỒ BÁO GIỜ ĐIỆN TỬ****I-GIỚI THIỆU:**

Hiện nay có rất nhiều dự án ứng dụng Led 7 đoạn và đang trở nên khá phổ biến , do đó tôi xin giới thiệu một trong những ứng dụng Led 7 đoạn cho bạn tham khảo mà bản thân tôi cũng đã từng làm , nó rất dễ làm nếu các bạn không ngại khó khăn .Sau khi ráp xong và kiểm tra kỹ là nó sẽ chạy được ngay .

II- SƠ ĐỒ MẠCH NHƯ SAU:

Hình này miêu tả sơ đồ mạch điện của một đồng hồ báo giờ . Ở đây Port1 (từ P1.0 đến P1.7) sẽ điều khiển các cực dương của Led 7 đoạn . Còn các chân P3.0 đến P3.3 sẽ điều khiển các cực âm chung của Led 7 đoạn .

Hãy lưu ý rằng: Cả 4 Led trên đều là loại âm chung (hay gọi là **Catod chung**). Do đó khi đi mua , bạn nhớ nói rõ là cần mua “ **Led Catod chung**” thế mới là dân chuyên nghiệp . (Đôi khi bạn nói Led âm chung sẽ có người không nhớ là Led loại gì đâu đấy nhé , hãy nhớ với Led người ta hay gọi là “Led Catod chung” hay ” Anod chung” , mặc dù hai cách gọi catod và âm cực là như nhau!) .

III- TÓM TẮT SỰ HOẠT ĐỘNG CỦA MẠCH:

Bạn thấy trên các chân P3.0 tới P3.3 có hai đường nối vào đó : Một đường nối đến transistor và một đường nối đến khoá nhấn (switch) với chân P3.4 . Điều đó có nghĩa là trong quá trình hiển thị thì đồng thời vi xử lý nó cũng kiểm tra xem có khoá nào được nhấn hay không . Nếu có thì chân P3.4 sẽ xuống mức “thấp” . Lúc đó vi xử lý sẽ xử lý theo chức của phím mà chúng ta đã nhấn . Trong 4 khoá đó có 2 khoá điều chỉnh giờ và điều chỉnh phút, một khoá dùng để tắt mở đèn mà nó được điều khiển bởi chân P3.7 thông qua cái rơ le-và điểm L và N bạn nối vào nguồn điện xoay chiều 220 V, dĩ nhiên bóng đèn cũng là loại 220 V.

IV - SAU ĐÂY LÀ CHƯƠNG TRÌNH ĐƯỢC VIẾT BẰNG NGÔN NGỮ C

```

/*
CLOCK V1.0
89C2051 + 4-digit 7-SEG led + 4-key switch
Simple Digital Clock demonstrates scanning LED and key switch.
The clock has 1-bit output port P3.7 for driving AC load through
MOC3040+triac.
The source program was compiled by Micro-C with optimize option.

```

*	* * * *			* * * *		* * * *
*	* * *	*		*	*	*
*	* * *			*	*	*
*	* * * *			*	*	*
*	* * *	*		*	*	*
*	* * *	*		*	*	*
*	* * * *			*	*	*
*	* * * *			*	*	*

set	set	time	manual
HOUR	MIN	ON/OFF	ON/OFF

Copyright (c) 1999 W.SIRICHOTE
* /

```
#include c:\mc51\8051io.h
#include c:\mc51\8051reg.h
```

```
extern register char cputick;
unsigned register char
sec100,sec,sec5,min,hour,flag1,command,ACC,temp,opto;
/* above must be defined as register for tiny model */
unsigned register char i,digit,buffer[4],onHour1,onMin1,offHour1,offMin1;
register char key,delay,count1;
char convert[10] = {0x3F,0x0c,0x76,0x5e,0x4d,0x5b,0x7b,0x0e,0x7f,0x5f};
```

```
main()
{
    opto = 0xff;
    cputick = 0;
    hour = 18;
    min = 0;
    sec = 0;
    key = -1;
    flag1 = 0;
    onHour1 = 18;    /* 18:30 turn lamp on */
    onMin1 = 01;
    offHour1 = 18;    /* 21:30 turn off */
    offMin1 = 02;
    count1 = 0;
    buffer[0] = 0x40;
    buffer[1] = 0x40;
    buffer[2] = 0x40;
```

```

buffer[3] = 0x40;

serinit(9600);      /* must be invoked for tiny model */

while(1)
{
    while ( cputick < 1)
        scanLED();

    cputick = 0;

/*----- the following tasks execute every 10ms -----*/

    time();
    timeToBuffer();
    blink();
    offmsd();
    keyexe();
    keydelay();
    comparetime();

/*-----*/
}

/* ***** change constant below for other X-tal *****/
time ()
/* update real-time clock */
{
    sec100++;
    if (sec100 >= 100)      /* 100 * 10 ms = 1 s */
    {sec100 = 0;
    flag1 |= 0x05;  /* set bit 0, bit 2 */
    temp = 50;
    sec++;
    if (sec >= 60)
    {sec = 0;
    flag1 |= 0x02; /* set bit 1 */
    min++;
    if (min >= 60)
    {min = 0;
    hour++;
    if (hour >= 24)
    {hour = 0;

    }
    }
    }
}

scanLED() /* scan 4-digit LED and 4-key switch, if key pressed key = 0-3
else key = -1 */

{
    int i;
    digit = 0x08;
    key = -1;
    for( i = 0; i < 4; i++) /* 4-DIGIT scanning */
    {
        P3 = ~digit & opto; /* send complement[digit] */
        P1 = ~buffer[i]; /* send complement[segment] */
        pause(1); /* delay a while */
        P1 = 0xff; /* off LED */
        if ((P3 & 0x10) == 0) /* if key pressed P3.4 became low */

```

```
        key = i;          /* save key position to key variable */
        digit>>=1;        /* next digit */
    }
}

timeToBuffer()
{
    buffer[0] = convert[min%10];
    buffer[1] = convert[min/10];
    buffer[2] = convert[hour%10];
    buffer[3] = convert[hour/10];
}

blink()
{
    if((flag1 & 0x04) != 0) /* check bit 2 if set decrement temp until zero
*/
    {temp--;
        if (temp != 0)
        {
            buffer[1] |= 0x80;
            buffer[2] |= 0x80;
        }
        else( flag1 &= ~0x04);
    }
}

keyexe()
{
    if (key != -1)
    {
        if ((flag1 & 0x80) == 0) /* within 0.5 sec after 1st press
the following execution is not allowed
*/
        {
            flag1 |= 0x80;
            delay = 50;

            switch(key){
                case (0): /* key position 0 */
                    manualOnOff(); /* service key 0 */
                    break;
                case (1): /* key position 1 */
                    savetimeOnOff1(); /* service key 1 */
                    break;
                case (2): /* key position 2 */
                    setmin(); /* service key 2 */
                    break;
                case (3): /* key position 3 */
                    sethour();
            }
        }
    }
}

sethour()
{
    hour++;
    if ( hour== 24)
        hour = 0;
}
```

```
}

setmin()
{
    min++;
    sec = 0;
    if( min == 60 )
        min = 0;
}

savetimeOnOff1()
{
    count1++;
    if (count1 == 1)
    {
        onHour1 = hour;
        onMin1 = min;
        buffer[0] = 0x00;
        buffer[1] = 0x68;
        buffer[2] = 0x78;
        buffer[3] = 0x71;
        showOnce();
    }
    else
    {
        count1 = 0;
        savetimeOff1();
    }
}

savetimeOff1()
{
    offHour1 = hour;
    offMin1 = min;
    buffer[0] = 0x63;
    buffer[1] = 0x63;
    buffer[2] = 0x78;
    buffer[3] = 0x71;
    showOnce();
}

manualOnOff()
{
    opto= ~opto | 0x7f; /* complement bit 7 which in turn activates P3.7
*/
    if ((opto & 0x80) == 0)
    {
        buffer[0] = 0;
        buffer[1] = 0;
        buffer[2] = 0x68;
        buffer[3] = 0x78;
        showOnce();
    }
    else
    {
        buffer[0] = 0;
        buffer[1] = 0x63;
        buffer[2] = 0x63;
        buffer[3] = 0x78;
        showOnce();
    }
}

showOnce()
{
```



```

    int i;
    for(i=0;i<500;i++)
        scanLED();
}
keydelay()
{
    if ((flag1 & 0x80) !=0)
    {
        delay--;
        if(delay == 0)
            flag1 &= ~0x80;
    }
}

comparetime()
{
    if((flag1 & 0x01) != 0 )
    {
        flag1 &= ~0x01;
        if(hour == onHour1 && min == onMin1)
            opto = 0x7f; /* clear P3.7 turning opto on */
        if(hour == offHour1 && min == offMin1)
            opto = 0xff; /* set bit P3.7 turning opto off */
    }
}
offmsd()

{
    if (buffer[3] == 0x3f) /* if msd = '0' then put blank unstead */
        buffer[3] = 0x00;
}

pause(j)
int j;
{
    int i;
    for (i = 0; i < j; i++)
        ;
}

```

V- MÃ FILE HEX CỦA NÓ -SAU KHI ĐÃ ĐƯỢC BIÊN DỊCH

Nếu các bạn rảnh rỗi thì hãy gõ y nguyên mã HEX này trong NotePad rồi lưu lại với đuôi là .HEX để nạp cho 89c2051 hoặc bạn có thể tới phòng Photo B1 trường đh.Bách Khoa để lấy file này về nạp chạy thử (tôi có chép sẵn trong đĩa mềm 1,44Mb).

```

:0300000002000EED
:0300030002019F58
:20000B0002019675811FD2AFD2A91201A280FE12001B80FBC82581C822C92
581C97A002224
:20002B00D083D082CF2581F581CFC082C08322CF2581F581CF22FBE493CB2
2FCE493FB7492
:20004B000193CC22FAE493F9740193CA2229F9E5F03AFA227C00CB30E7011
CCB2275F0009B
:20006B0030E70215F022FDE493FF7401934F7004A3A3800C740293B505107
40393B5F00A99
:20008B00E493C0E0740193C0E022A3A3A3A380D7FEEBA4FD8EF0ECA4FCEB8
EF0A4C5F02C0F
:2000AB002DC5F02211B58EF0ED22C002C001AAF0F97E007D007F11C3E933F
9EA33FADF0867
:2000CB00F5F0E9D001D00222ED33FDEE33FEC3ED9BF5F0EE9C40E0ADF0FED
380DBBB0001E7

```

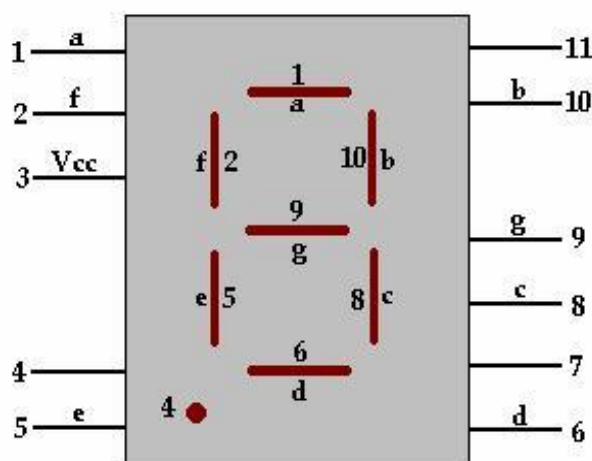
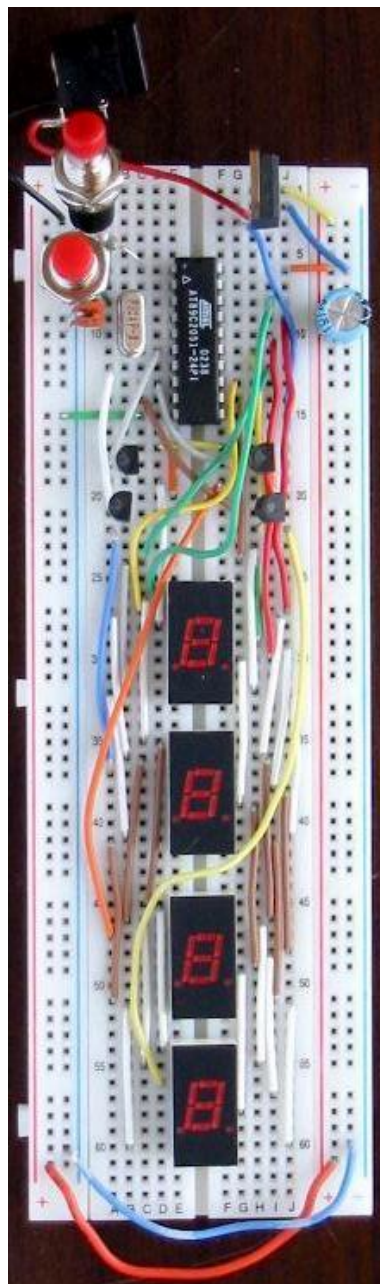
:2000EB0022C333C5F033C5F0DBF722BB000122C3C5F013C5F013DBF722240
280022401C595
:20010B00F03400C5F022C394028003C39401C5F09400C5F0223116F4C5F0F
4C5F02245F095
:20012B007022042231784024801A3178401E7014801A3178400E7014800A3
178500E8004A0
:20014B00318C6008E4F5F022318C60F8E4F5F00422318C40F780ED318C40F
170E780ED313C
:20016B008C40E170E780DD318C50E180D7C39BC5F09C20E70720D207C345F
02220D2F9D340
:20018B0022C5F0C39C7003E5F09B22758CDC758A0005083205083274FFF51
2E4F5087412DE
:2001AB00F50DE4F50CE4F50A74FFF51DE4F50E7412F5197401F51A7412F51
B7402F51CE4E5
:2001CB00F51FE4F5F079157A001200587440F7740175F00079157A0012005
87440F77402AD
:2001EB0075F00079157A001200587440F7740375F00079157A00120058744
0F7748075F020
:20020B0025C0E0C0F012079E15811581740175F00045F07003020256E5081
200687B0112AA
:20022B00005F12012F700302023B1202C8020223E4F50812025712036F120
42612072A12FC
:20024B0004781206961206C602021722E5090509E50975F0007B647C00120
17270030202A8
:20026B00C7E4F509E50E4405F50E7432F511E50A050AE50A75F0007B3C7C0
0120172700362
:20028B000202C7E4F50AE50E4402F50EE50C050CE50C75F0007B3C7C00120
17270030202E2
:2002AB00C7E4F50CE50D050DE50D75F0007B187C0012017270030202C7E4F
50D22058105C7
:2002CB00817408F51474FFF51DE4F5F078FD12001FF608A6F078FD12001FE
60886F07B04FC
:2002EB007C0012012F701902036A78FD12001FE60886F012010818F608A6F
012011680D5EE
:20030B00E514F45512F5B078FD12001FE60886F079157A00120058E7F4F59
0740175F00023
:20032B00C0E0C0F01207551581158174FFF590E5B054101200687B0012005
F12014B7003A0
:20034B0002035978FD12001FE60886F0F51DE51475F0007B017C001200F6F
5140202F515A3
:20036B0081158122E4F5F079157A00120058E50C75F0007B0A7C001200AFC
001C00279C129
:20038B007A0712005889828A83E493D007D000F6740175F00079157A00120
058E50C75F099
:2003AB00007B0A7C001200B5C001C00279C17A0712005889828A83E493D00
7D000F6740220
:2003CB0075F00079157A00120058E50D75F0007B0A7C001200AFC001C0027
9C17A071200D2
:2003EB005889828A83E493D007D000F6740375F00079157A00120058E50D7
5F0007B0A7CC8
:20040B00001200B5C001C00279C17A0712005889828A83E493D007D000F62
2E50E54047554
:20042B00F0007B007C001201537003020477E5111511E51175F0007B007C0
012015370032D

:20044B00020471740175F00079157A00120058E74480F7740275F00079157
A00120058E7F8
:20046B004480F7020477E50E54FBF50E22E51D1200687BFF12005F1201537
0030204DEE5C9
:20048B000E548075F0007B007C0012014B70030204DEE50E4480F50E7432F
51EE51D1200D7
:2004AB00689004CA0200711205BD0204DE1205140204DE1204F80204DE120
4DF0204DEC4A2
:2004CB00040300BE040200B8040100B20400000000DE0422E50D050DE50D7
5F0007B187C65
:2004EB000012014B70030204F7E4F50D22E50C050CE4F50AE50C75F0007B3
C7C0012014B4F
:20050B007003020513E4F50C22E51F051FE51F1200687B0112005F12014B7
003020571E57B
:20052B000DF519E50CF51AE4F5F079157A00120058E4F7740175F00079157
A0012005874BF
:20054B0068F7740275F00079157A001200587478F7740375F00079157A001
200587471F7D7
:20056B00120650020577E4F51F12057822E50DF51BE50CF51CE4F5F079157
A00120058742E
:20058B0063F7740175F00079157A001200587463F7740275F00079157A001
200587478F7AC
:2005AB00740375F00079157A001200587471F7020650E512F4447FF512548
075F0007B0045
:2005CB007C0012014B7003020613E4F5F079157A00120058E4F7740175F00
079157A00129E
:2005EB000058E4F7740275F00079157A001200587468F7740375F00079157
A0012005874DB
:20060B0078F712065002064FE4F5F079157A00120058E4F7740175F000791
57A001200583F
:20062B007463F7740275F00079157A001200587463F7740375F00079157A0
012005874788C
:20064B00F71206502205810581E4F5F078FD12001FF608A6F078FD12001FE
60886F07BF486
:20066B007C0112012F701902069178FD12001FE60886F012010818F608A6F
012011680D53F
:20068B001202C80206751581158122E50E548075F0007B007C00120153700
30206C5E51EDC
:2006AB00151EE51E1200687B0012005F12014B70030206C5E50E547FF50E2
2E50E5401754D
:2006CB00F0007B007C001201537003020729E50E54FEF50EE50D75F000AB1
97C0012014BE0
:2006EB00600CE50C75F000AB1A7C0012014B45F07003020704747FF512E50
D75F000AB1BC2
:20070B007C0012014B600CE50C75F000AB1C7C0012014B45F070030207297
4FFF5122274A7
:20072B000375F00079157A00120058E775F0007B3F7C0012014B700302075
4740375F00048
:20074B0079157A00120058E4F72205810581E4F5F078FD12001FF608A6F07
8FD12001FE684
:20076B000886F078F912001F860308860412012F701902079978FD12001FE
60886F01201A9
:20078B000818F608A6F012011680CF020780158115812278FB12001F86030
8860474807523

:2007AB00F0701200B5120120758921F58DF58B758859759852223F0C765E4
D5B7B0E7F5FAE
:00000001FF

V-HÌNH ẢNH CỦA ĐỒNG HỒ BÁO GIỜ ĐÃ ĐƯỢC RÚT GỌN

Ở đây tôi chỉ dùng hai khoá : Chỉnh giờ và chỉnh phút thôi. Và tôi cũng bỏ luôn phần điều khiển bóng đèn –rờ le, và một vài thứ không cần thiết khác .



Rất đơn giản quá phải không bạn !?

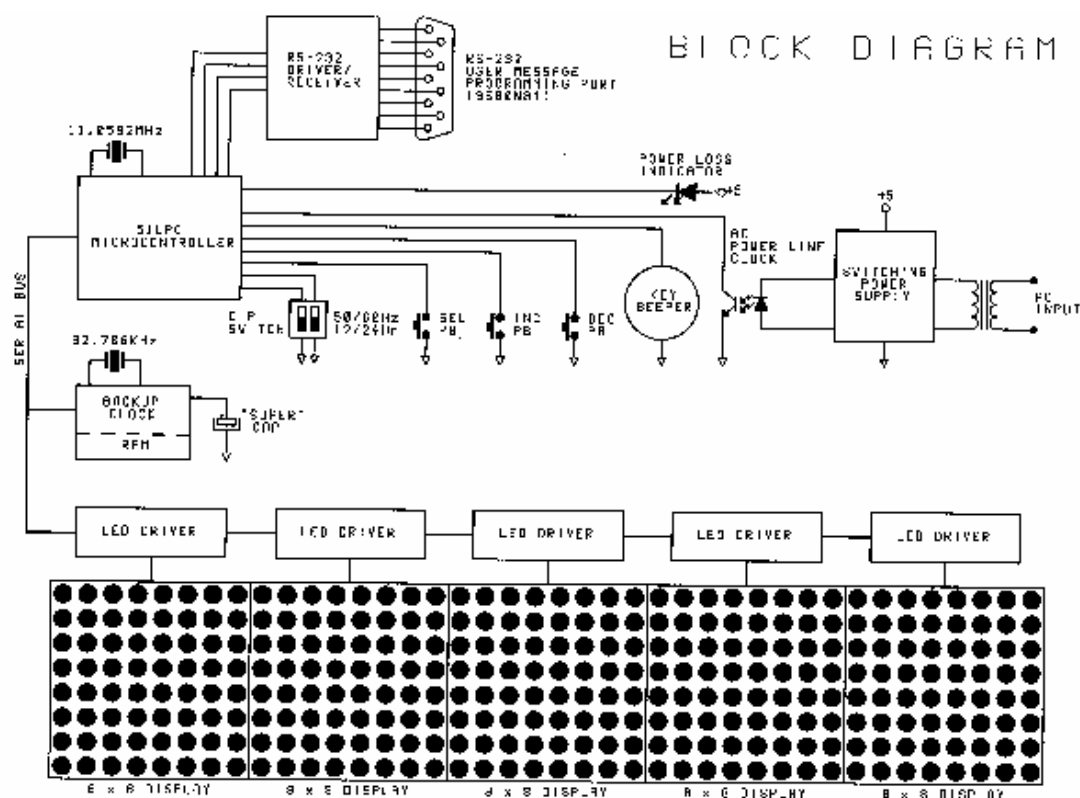
Vâng ! đồng hồ -giờ rất dễ làm thôi!

Sau đây là mộ loại đồng hồ báo giờ khác , nó có thể chạy từ phải sang trái kèm theo một dòng chữ rất xinh xắn chẳng hạn như bây”BAY GIO LA 1:23 “.

Xin mời các bạn đón đọc ví dụ này trong quyển thứ 2 sắp viết xong: “ **Hướng Dẫn Giao Tiếp Vi Xử Lý Với Máy Tính**” trong quyển này tôi sẽ giúp bạn biết cách giao tiếp

giữa vi xử lý và máy tính thông qua phần mềm Visual Basic , đồng thời tôi cũng có đề cập nhiều ví dụ mạch điện rất lý thú như “ Máy đo nhiệt độ hiển thị trên Led , Máy đo nhiệt độ hiển thị trên Máy Tính , Làm bảng đèn chữ chạy đơn giản , Làm bảng đèn chữ chạy giao tiếp với Máy Tính , Mạch đo điện áp giao tiếp máy tính (đây là đề tài mà tôi đã đạt giải 3 trong ” cuộc thi sinh viên sáng tạo lần 1 ” , và những mạch điện vi xử lý khác tôi sẽ giới thiệu sau , xin các bạn đón đọc !

Các tài liệu liên quan của tôi xin các ban hãy đến phòng Photo B1- đh.Bách Khoa.



Lời kết :

Quyển sách cuối cùng đã hoàn thành đúng vào lúc các bạn đang học về vi xử lý . Như vậy sau gần hơn một tháng biên soạn-kể từ lúc khởi thảo , trong thời gian ngắn ấy , tập sách chỉ chú trọng phần nhiều vào mặt thực hành , không đề cập quá sâu vào lý thuyết .

Mục đích mà tôi viết quyển sách này không mong gì hơn là nó sẽ góp phần phục vụ cho các bạn trong việc học vi xử lý đạt được nhiều lợi ích và đam mê hơn bởi sự kết hợp hài hoà giữa lý thuyết và thực hành !

Trong quá trình tự lắp ráp , nếu các bạn gặp sự cố thì hãy liên hệ với chúng tôi, bằng kinh nghiệm đã từng lắp ráp loại này tôi nghĩ sẽ có những lời phức đáp rất thiết thực , ngõ hầu sẽ đem lại niềm vui cho các bạn , địa chỉ liên hệ : duyphi@yahoo.com hoặc duong_phi_vxl@yahoo.com , nếu được các bạn có thể viết thư cho tôi theo địa chỉ 67/30/10/25 đường Phú Trung –F10-Q.Tân Bình Tp.HCM .

Các bạn cũng có thể xem nội dung của quyển sách này , cùng các tài liệu khác không những về vi xử lý , mà còn nhiều môn khác như điện tử , kỹ thuật số , điện công nghiệp , điện tử cơ bảntại địa chỉ website sau của tôi :

<http://vixuly.cjp.net>

Dĩ nhiên đây là lần đầu tôi làm công việc này , nên sai sót và nhiều khiếm khuyết là điều sẽ có thể xảy ra , tôi xin trân trọng lắng nghe sự đóng góp ý kiến của các bạn cùng của các sư huynh và của thầy cô trên tinh thần góp ý , sửa chữa để ngày một cùng nhau hoàn thiện hơn ! Trên tinh thần ấy tôi xin vô cùng cảm ơn quý vị , và tôi sẽ cố gắng hơn nữa để sớm giới thiệu với bạn đọc tập 2:

“Hướng dẫn giao tiếp vi xử lý với máy tính ” bằng Visual

Basic

Cuối cùng tôi xin có lời tri ơn sâu sắc đến các bạn trong trường , các sư huynh cùng quý thầy cô đã có lời động viên , có lời góp ý trong suốt thời gian tôi viết quyển sách này .

Một lần nữa tôi xin rất cảm ơn các bạn đã đọc quyển sách này !

Xin chúc các bạn luôn vui vẻ và gặt hái được nhiều thành công từ quyển sách nhỏ này!

Sinh viên thực hiện:

Lê Duy Phi

10-02-2004 Tp.HCM