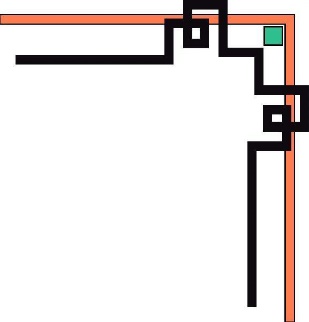
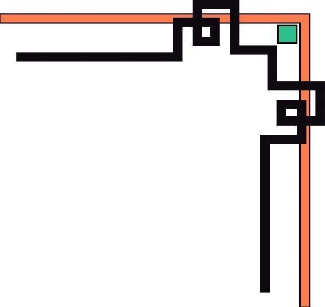
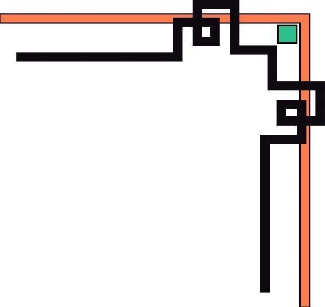
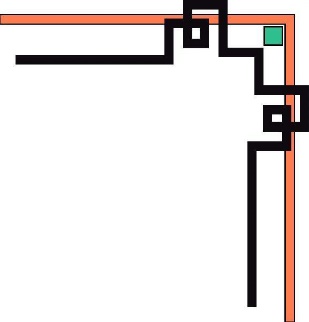
**TRƯỜNG ĐẠI HỌC CNTT VÀ TRUYỀN THÔNG**



**KHOA CÔNG NGHỆ ĐIỆN TỬ VÀ TRUYỀN THÔNG**

**------00O**\*\* **🕮**\*\* **O00------**

****

**BÀI TẬP LỚN**

**MÔN: THỰC HÀNH VI ĐIỀU KHIỂN VÀ VI XỬ LÝ**

# *Đề tài:*

**Thiết kế mạch điều khiển thiết qua sử dụng module Sim800L**

**Giảng viên hướng dẫn: Nguyễn Văn Thao**

**Nhóm sinh viên: Nguyễn Trọng Dũng**

**Lớp: KTMT K15A**

***Thái Nguyên, tháng 4 năm 2018***

# MỤC LỤC

# LỜI NÓI ĐẦU

Việt Nam ta ngày phát triển và giàu mạnh. Một trong những thay đổi đáng kể là Việt Nam đã gia nhập “WTO”, một bước ngoặt quan trọng để đất nước thay đổi bộ mặt nghèo nàn của mình, để chúng ta con người Việt có cơ hội nắm bắt nhiểu thành tựu vĩ đại của thế giới, đặc biệt là về các lĩnh vực khoa học kĩ thuật nói chung và ngành Điện Tử nói riêng.

Thế hệ trẻ chúng ta không tự mình phấn đấu học hỏi không ngừng thì chúng ta sẽ sớm lạc hậu và nhanh chóng thụt lùi. Nhìn ra được điều đó Trường “ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG THÁI NGUYÊN” đã sớm chủ trương hình thức đào tạo sâu rộng, từ thấp đến cao. Để tăng chất lượng học tập của sinh viên nhà trường nói chung và trung tâm Điện Tử nói riêng đã tổ chức cho sinh viên làm các bài tập lớn môn học nhằm tạo nền tảng vững chắc cho sinh viên khi ra trường, đáp ứng nhu cầu tuyển dụng việc làm.

Sau thời gian học tập được thầy giáo trong khoa giảng dạy về các kiến thức chuyên môn, đồng thời được sự giúp đỡ nhiệt tình của thầy Nguyễn Văn Thao cùng với sự nỗ lực của bản thân em đã hoàn thiện được bài Báo cáo: **“Thiết kế mạch điều khiển thiết qua sử dụng module Sim800L*”***.

Việt Nam ta ngày phát triển và giàu mạnh. Một trong những thay đổi đáng kể là Việt Nam đã gia nhập “WTO”, một bước ngoặc quan trọng để đất nước thay đổi bộ mặt nghèo nàn của mình, để chúng ta con người Việt có cơ hội nắm bắt nhiểu thành tựu vĩ đại của thế giới, đặc biệt là về các lĩnh vực khoa học kĩ thuật nói chung và ngành Điện Tử nói riêng.

Trong quá trình làm Báo cáo do thời gian, kiến thức và kinh nghiệm của em còn có hạn nên sẽ không thể tránh khỏi những sai sót. Em rất mong được sự giúp đỡ và tham khảo ý kiến của thầy cô và các bạn nhằm đóng góp phát triển thêm cho đề tài này.

*Thái Nguyên, ngày 02 tháng 04 năm 2018*

Nhóm sinh viên thực hiện

# CHƯƠNG I. TỔNG QUAN VỀ MẠCH ĐIỀU KHIỂN THIẾT BỊ

**1.1 ĐẶT VẤN ĐỀ**

Ngày nay việc nghiên cứu ứng dụng vi điều khiển vào các ứng dụng cuộc sống khá phổ biến. Ở trong công nghiệp cũng như các lĩnh vực liên quan tới điều khiển từ xã đã phát huy được lợi thế khi sử dụng vi điều khiển kết hợp với các kiểu truyền thông không dây như bluetooth, wifi, sóng điện thoại,… đã trở nên đơn giản hơn giảm bớt được nhiều sức lao động và thời gian. Được sự gợi ý của thầy Nguyễn Văn Thao và qua tìm hiểu của các thành viên trong nhóm nên chúng em đã quyết định chọn đề tài điều khiển bật tắt thiết bị sử dụng module Sim800L và ứng dụng của họ vi điều khiển 8051.

* 1. **GIỚI HẠN ĐỀ TÀI:**

Chúng em đang thi công điều khiển thiết bị sử dụng module Sim800L với một số yêu cầu sau:

* Khi có cuộc gọi tới thì bật/tắt thiết bị là 2 Rơ le.

**1.3 MỤC ĐÍCH NGHIÊN CỨU**

Khi thực hiện đề tài này, mục đích trước hết là hoàn tất chương trình môn học. Mục đích thứ hai là bổ sung thêm kinh nghiệm cho công việc chúng em sau này.

**1.4 ĐỐI TƯỢNG NGHIÊN CỨU**

- Tìm hiểu về vi xử lý sử dụng VĐK 8051.

- Tìm hiểu về module Sim800L.

- Tìm hiểu về cách truyền nhận dữ liệu thông qua cổng truyền thông nối tiếp UART.

# CHƯƠNG 2. CÁC LINH KIỆN SỬ DỤNG.

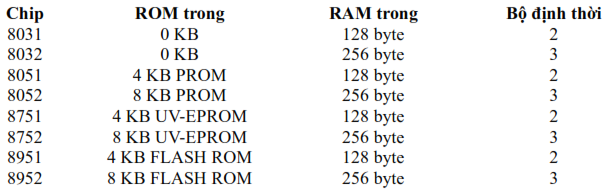
# 2.1. Vi điều khiển AT89S51

## 2.1.1. Giới thiệu chung về họ VĐK 8051

MCS-51 là họ vi điều khiển của hãng Intel. Vi mạch tổng quát của họ MCS-51 là chip 8051. Chip 8051 có một số đặc trưng cơ bản sau:

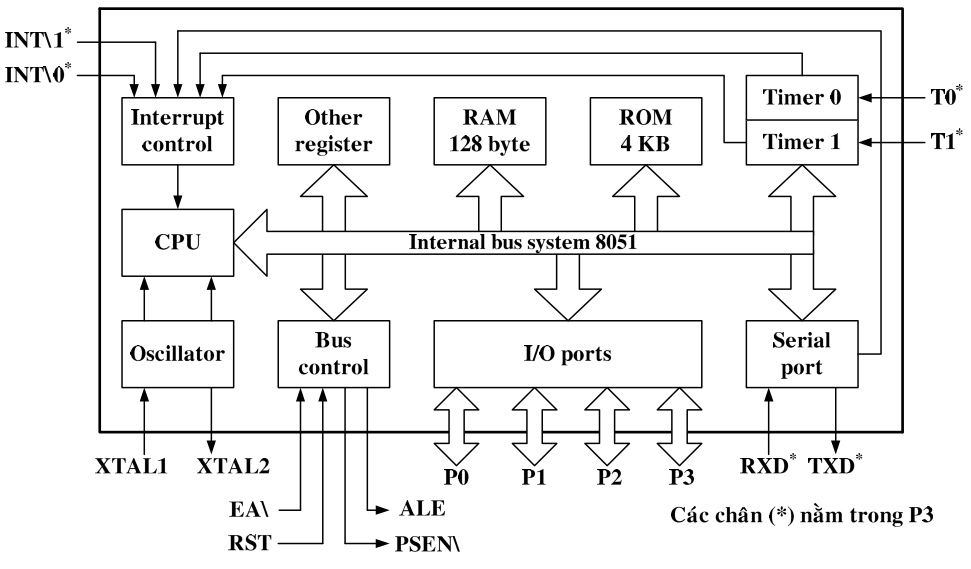
* Bộ nhớ chương trình bên trong: 4 KB (ROM).
* Bộ nhớ dữ liệu bên trong: 128 byte (RAM).
* Bộ nhớ chương trình bên ngoài: 64 KB (ROM).
* Bộ nhớ dữ liệu bên ngoài: 64 KB (RAM).
* 4 port xuất nhập (I/O port) 8 bit.
* 2 bộ định thời 16 bit.
* Mạch giao tiếp nối tiếp.
* Bộ xử lý bit (thao tác trên các bit riêng lẻ).
* 210 vị trí nhớ được định địa chỉ, mỗi vị trí 1 bit.
* Nhân / Chia trong 4 µs.

Ngoài ra, trong họ MCS-51 còn có một số chip vi điều khiển khác có cấu trúc tương đương như:



Hình 2.1. Cấu hình một số loại vi điều khiển

## 2.1.2. Các chân của họ VĐK 8051



Hình 2.2. Sơ đồ các chân của IC 8051

- CPU (Central Processing Unit): Đơn vị xử lý trung tâm tính toán và điều khiển quá trình hoạt động của hệ thống.

- OSC (Oscillator): Mạch dao động tạo tín hiệu xung clock cung cấp cho các khối trong chip hoạt động.

- Interrupt control: Điều khiển ngắt nhận tín hiệu ngắt từ bên ngoài (INT0\, INT1\), từ bộ định thời (Timer 0, Timer 1) và từ cổng nối tiếp (Serial port), lần luợt đua các tín hiệu ngắt này đến CPU để xử lý.

- Other registers: Các thanh ghi khác : Lưu trữ dữ liệu của các port xuất/nhập, trạng thái làm việc của các khối trong chip trong suốt quá trình hoạt động của hệ thống.

- RAM (Random Access Memory): Bộ nhớ dữ liệu trong chip lưu trữ các dữ liệu.

- ROM (Read Only Memory): Bộ nhớ chương trình trong chip lưu trữ chương trình hoạt động của chip.

- I/O ports (In/Out ports): Các port xuất/nhập điều khiển việc xuất nhập dữ liệu duới dạng song song giữa trong và ngoài chip thông qua các port P0, P1, P2, P3.

- Serial port: Port nối tiếp điều khiển việc xuất nhập dữ liệu dưới dạng nối tiếp giữa trong và ngoài chip thông qua các chân TxD, RxD.

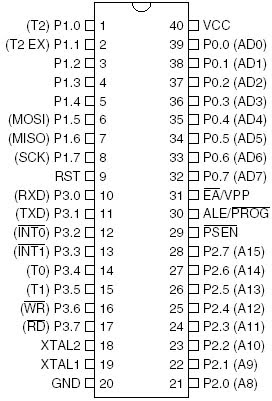
- Timer 0, Timer 1: Bộ định thời 0, 1 dùng để định thời gian hoặc đếm sự kiện (đếm xung) thông qua các chân T0, T1.

- Bus control: Điều khiển bus điều khiển hoạt động của hệ thống bus và việc di chuyển thông tin trên hệ thống bus.

- Bus system: Hệ thống bus liên kết các khối trong chip lại với nhau.

## 2.1.3. Vi điều khiển AT89S51

Vi điều khiển AT89S51 là một vi điều khiển thuộc họ 8051, loại CMOS, có tốc độ cao và công suất thấp với bộ nhớ Flash có thể lập trình được. AT89S51 có 40 chân.

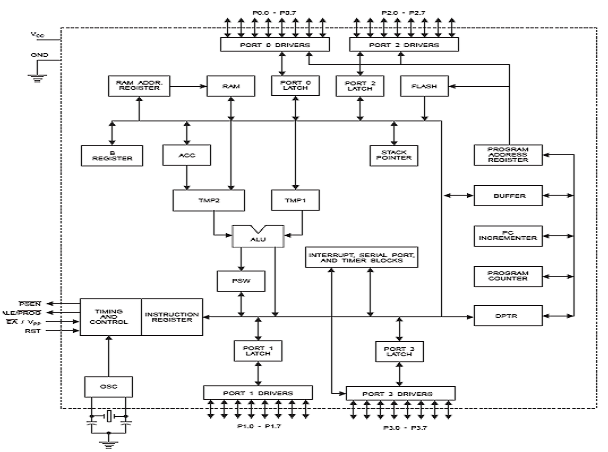
****

Hình 2.3. Hình minh họa thực tế và sơ đồ chân ra của IC AT89S51

VĐK AT89S51 có các đặc tính sau:

* 4 KB EPROM (Flash Erasable and Programmable Read Only Memory), có khả năng tới 1000 chu kỳ ghi/xoá.
* Tần số hoạt động từ: 0Hz đến 24 MHz.
* Có 3 mức khóa bộ nhớ lập trình.
* 128 Byte RAM nội.
* 4 Port xuất /nhập I/O 8 bit.
* 2 bộ Timer/counter 16 Bit.
* 6 nguồn ngắt.
* Giao tiếp nối tiếp điều khiển bằng phần cứng.
* 64 KB vùng nhớ mã ngoài
* 64 KB vùng nhớ dữ liệu ngoài.
* Cho phép xử lý bit.
* 210 vị trí nhớ có thể định vị bit.

**a. Sơ đồ khối:**



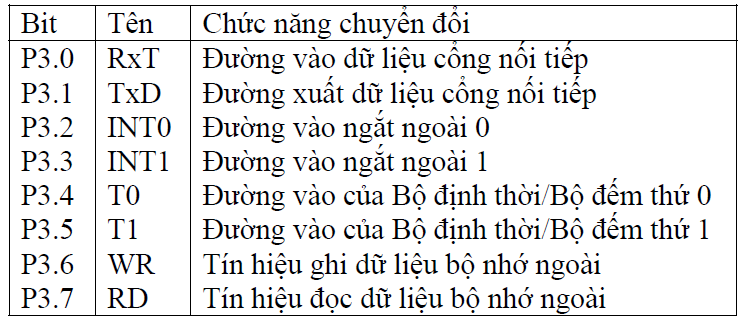
Hình 2.4. Biểu diễn sơ đồ khối chức năng của bộ vi điều khiển AT89S51.

**b. Chức năng các chân AT89S51:**

**Các cổng vào/ra song song**

8051 có 4 cổng vào/ra song song 8 bit là Port 0, Port 1, Port 2, Port 3. Các cổng này có thể sử dụng như là cổng vào hoặc cổng ra.

* Cổng Port 0 (các chân 32÷39): là cổng vào/ra song song có hai chức năng.
* Cổng Port 1 (các chân 1÷8): là cổng vào/ra song song. Các chân được ký hiệu P1.0, P1.1, P1.2…có thể dùng cho giao tiếp với các thiết bị ngoài nếu cần.
* Cổng Port 2 (các chân 21÷28): là một cổng vào/ra song song có tác dụng kép, được dùng như các đường xuất nhập hoặc là byte của bus địa chỉ 16 bitđối với các thiết bị dùng bộ nhớ mở rộng.
* Cổng Port 3 (các chân 10÷17): là cổng vào/ra song song có tác dụng kép. Bảng sau cho ta chức năng của các chân cổng Port3



**Nguồn:**

Chân 40: VCC = 5V ± 20%

Chân 20: GND

**Các chân tín hiệu điều khiển**

- Chân cho phép bộ nhớ chương trình PSEN (Program Storage Enable):

+Tín hiệu PSEN là tín hiệu ra ở chân 29 có tác dụng kép.

+Cho phép đọc bộ nhớ chương trình ngoài, thường được nối đến chân OE (Output Enable) của EPROM cho phép đọc các byte mã lệnh.

+Khi vi điều khiển thi hành chương trình trong ROM nội PSEN sẽ ở mức logic 1.

- Chân cho phép chốt địa chỉ ALE/PROG (Address Latch Enable):

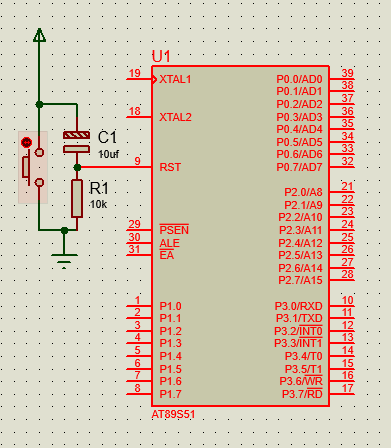
+Chân tín hiệu ALE (chân 30) đưa ra xung điều khiển cho phép chốt byte thấp của địa chỉ khi vi điều khiển truy xuất bộ nhớ ngoài. Chân này cũng là đầu vào của xung lập trình khi lập trình cho FLASH, khi đó chân tín hiệu ở mức 0.

- Chân tín hiệu truy xuất ngoài EA (External Access):

+Tín hiệu vào EA (chân 31) được nối với 5V (mức logic 1) hoặc với GND (mức 0). Nếu ở mức 1, vi điều khiển thi hành chương trình từ ROM nội. Nếu ở mức 0, vi điều khiển sẽ thi hành chương trình ở bộ nhớ mở rộng.

- Chân thiết lập lại RST (Reset):

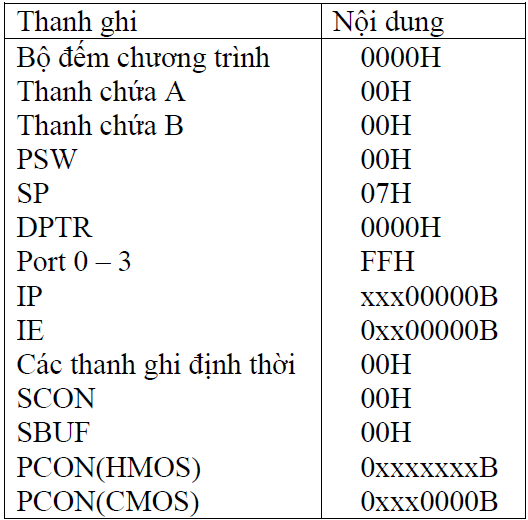
+Chân RST (chân 9) là đường vào xóa chính của vi điều khiển dùng để thiết lập lại hệ thống. Khi chân tín hiệu này đưa lên mức cao, các thanh ghi bên trong được nạp những giá trị thích hợp để khởi động hệ thống. RST có thể được kích khi cấp điện dùng một mạch R-C. Mạch này như sau:



Hình 2.5. Mạch reset cho AT89S51

+Trạng thái các thanh ghi của vi điều khiển được tóm tắt trong bảng sau. Quan trọng nhất trong các thanh ghi trên là thanh ghi bộ đếm chương trình (PC – Program Counter). Sau khi thiết lập lại (RST trở về mức thấp), thanh ghi PC có giá trị 0000H, tức là chương trình luôn bắt đầu tại địa chỉ đầu tiên trong bộ nhớ chương trình. Nội dung của RAM trên chip không bị thay đổi khi thiết lập lại.

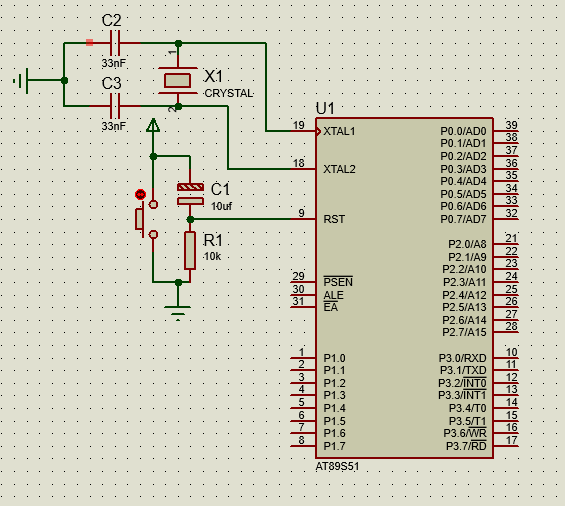
Bảng 2.1. Các thanh ghi của 89S52



Bảng trạng thái các thanh ghi sau khi Reset

- Các chân XTAL1, XTAL2:

+Các chân này (chân 18, 19) nối với bộ tạo dao động trên chip. Mạch tạo dao động như sau:



Hình 2.6. Mạch tạo dao động cho bộ tạo dao động của AT89S51

+Tần số của dao động thường là 12MHz. Khi đó tụ có giá trị 33pF. Chân VCC nối đến +5V của nguồn cấp, chân GND nối đất.

**Bộ nhớ trên chip**

**RAM trong:**

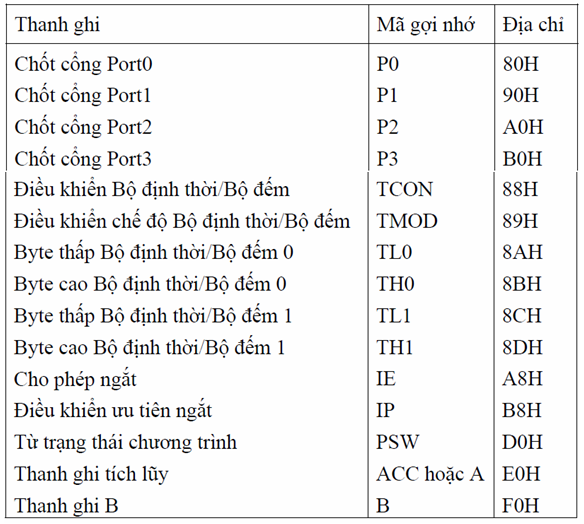
+Bộ vi điều khiển 8051 có 128 byte RAM trong bao gồm 32 byte đầu tiên (00H đến 1FH) dành cho các thanh ghi, 16 byte tiếp theo (20H đến 2FH) là vùng RAM định địa chỉ theo bit, sau đó là 80 byte RAM nháp.

+Vùng thanh ghi có 32 byte, chia thành 4 khối (bank 0 đến bank 3), mỗi khối có 8 thanh ghi (từ R0 đến R7).

+Ở vùng RAM định địa chỉ theo bit, các bit được dánh địa chỉ từ 00H đến 7FH. Các thanh ghi chuyên dụng (SFRs – Special Function Registers).

+Các thanh ghi này có địa chỉ từ 80H đến FFH. Chúng chứa nội dung của các thanh ghi điều khiển.

+Sau đây là một số thanh ghi chuyên dụng:



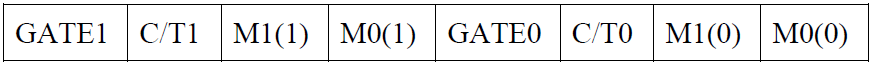
**ROM:**

+Bộ vi điều khiển AT89S51 có 4KB FLASH lập trình được.

+ROM luôn chiếm vùng địa chỉ thấp nhất trong bộ nhớ chương trình.

**Các Bộ định thời/Bộ đếm**

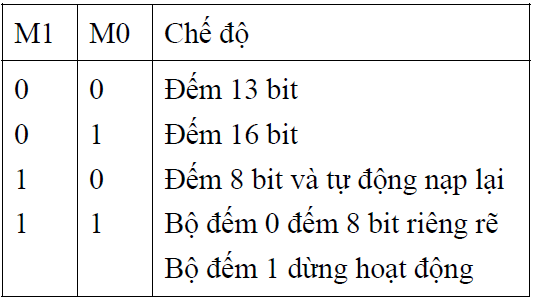
Bộ vi điều khiển 8051 có 2 Bộ định thời/Bộ đếm là Bộ định thời/Bộ đếm 0 và Bộ định thời/Bộ đếm 1. Chúng có thể hoạt động như là bộ định thời hoặc bộ đếm. Chế độ hoạt động của các Bộ định thời/Bộ đếm được cất trong thanh ghi TMOD:



+Nếu bit GATE xóa, các Bộ định thời/Bộ đếm được phép hoạt động khi bit TR# tương ứng trong thanh ghi TCON thiết lập. Ngược lại, nếu GATE thiết lập thì các Bộ định thời/Bộ đếm chỉ hoạt động khi các chân INT# tương ứng tích cực (mức thấp).

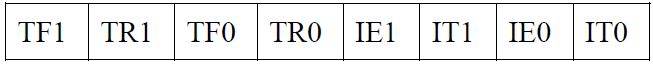
+Bit C/T# dùng để lựa chọn chế độ hoạt động bộ đếm hay bộ định thời. Nếu bị xóa, thì nó hoạt động theo chế độ định thời với nguồn xung là xung tạo ra từ bộ tạo dao động trên chip sau khi chia 12.

+Các bit M0, M1 dùng để xác định chế độ đếm cho các bộ đếm:



Các chế độ hoạt động của các bộ đếm

Sự hoạt động của các Bộ định thời/Bộ đếm được điều khiển bởi thanh ghi TCON:



+Các bit TR# cho phép Bộ định thời/Bộ đếm hoạt động (nếu được thiết lập) hoặc không cho phép chúng hoạt động (nếu bị xóa).

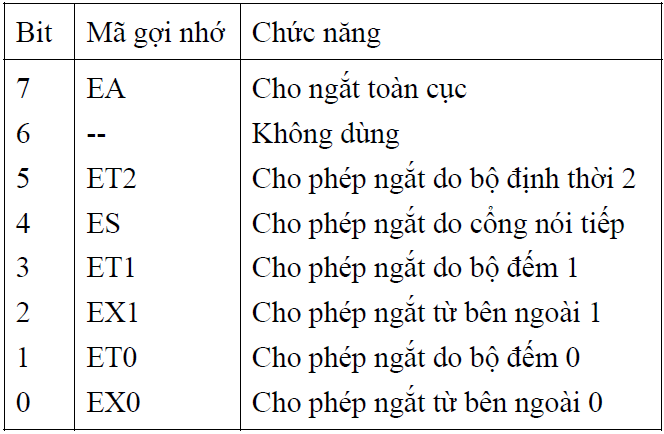
+Các bit TF# là các cờ tràn tương ứng với các Bộ định thời/Bộ đếm. Chúng được thiết lập khi xảy ra tràn và được xóa bằng phần cứng nếu khi đó bộ xử lý rẽ nhánh đến chương trình phục vụ ngắt tương ứng.

+Các bit IT# là các bit ngắt.

+Các bit IE# là các cờ ngắt cạnh, được thiết lập khi dò thấy ngắt cạnh.

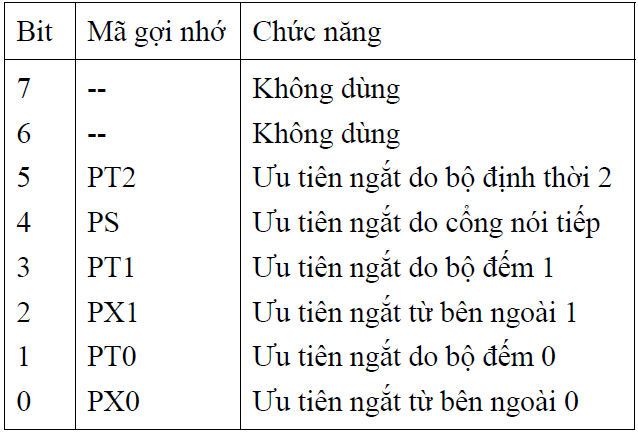
**Điều khiển ngắt**

Bộ vi điều khiển 8051 có 5 nguồn ngắt: TF0, TF1, INT0, INT1 và ngắt do cổng nối tiếp. Sự điều khiển hoạt động ngắt được cất trong 2 thanh ghi là thanh ghi cho phép ngắt IE (Interrupt Enable) và thanh ghi xác định thứ tự ưu tiên ngắt IP (Interrupt Priority). Các bit và chức năng của chúng trong thanh ghi IE như sau (thiết lập là cho phép, xóa là cấm):



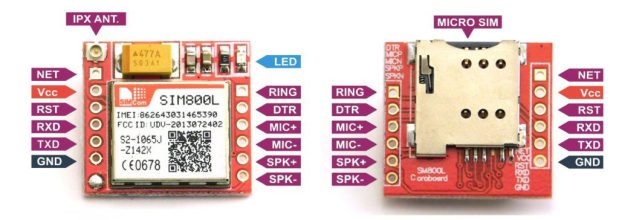
Các bit và chức năng của nó trong thanh ghi IE.

Với thanh ghi IP:



Các bit và chức năng của chúng trong thanh ghi IP.

# 2.2. Module Sim800L



Hình 2.5 Module Sim800L

Module Sim800L có khả năng nhắn tin SMS, nghe, gọi, GPRS, … như một điện thoại nhưng có kích thước nhỏ nhất trong các loại module SIM (25 mm x 22 mm). Điều khiển module sử dụng bộ tập lệnh AT dễ dàng, chân kết nối dùng rào đực thông dụng (male header) chuẩn 100mil.

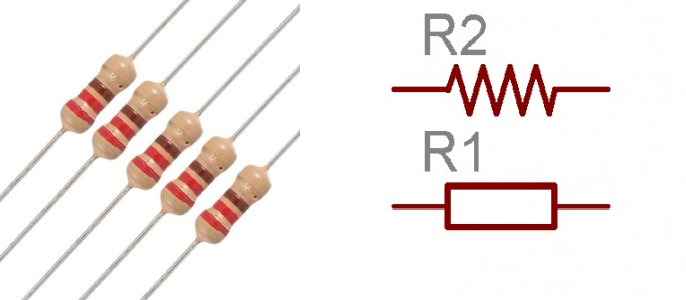
Đặc điểm:

* Điện áp hoạt động : 3.7 - 4.2V
* Dòng khi ở chế độ chờ: 10 mA
* Dòng khi hoạt động: 100 mA đến 1A.
* Khe cắm SIM : MICROSIM
* Hỗ trợ 4 băng tần : GSM850MHz, EGSM900MHz, DSC1800Mhz, PCS1900MHz
* Có thể giao tiếp với vi điều khiển qua TTL không cần MAX232
* 1 led báo tín hiệu

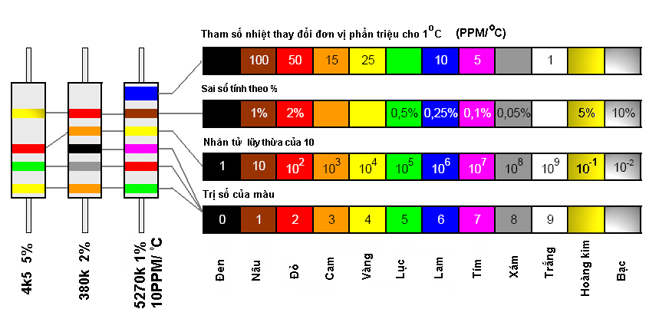
# 2.3. Điện trở

Trong mạch này em sử dụng điện trở than, là loại điện trở sử dụng nhiều nhất trong mạch điện. Điện trở than được cấu tạo bởi hỗn hợp của bột than và các chất khác, tùy theo tỷ lệ pha trộn mà điện trở có trị số lớn hay nhỏ. Bên ngoài điện trở được bọc bằng lớp cách điện.

Trong mạch điện thì điện trở có tác dụng điều chỉnh điện áp, hạn chế dòng điện, chia điện áp, điều chỉnh hệ số khuếch đại, ổn định nhiệt…..



Hình 2.8. Điện trở và kí hiệu thường gặp.



Hình 2.9. Bảng màu điện trở.

# 2.4. Tụ điện

Công dụng:

* Cho điện áp xoay chiều đi qua và ngăn điện áp một chiều lại, tụ được sử dụng để truyền tín hiệu giữa các tầng khuếch đại có chênh lệch về điện áp một chiều.
* Lọc điện áp xoay chiều sau khi đã được chỉnh lưu (loại bỏ pha âm) thành điện áp một chiều bằng phẳng. Đó là nguyên lý của các tụ lọc nguồn.
* Với điện AC (xoay chiều) thì tụ dẫn điện còn với điện DC (một chiều) thì tụ lại trở thành tụ lọc.

\*Tụ gốm:



Hình 2.10. Tụ gốm 103.

Đặc điểm của tụ gốm là kích thước nhỏ, điện dung lớn, có tính ổn định rất tốt, có thể làm việc lâu dài mà không lão hoá.

Cách đọc: lấy hai chữ số đầu nhân với 10(mũ là số thứ 3)

\*Ví dụ: Tụ trên hình có trị số là 474K nghĩa là

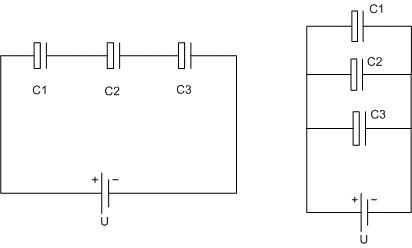
Giá trị của tụ= 47 x 104 = 470000 (pF) = 0,47 (µF)

Chữ cái K hoặc J ở cuối là chỉ sai số 5% hay 10% của tụ.

Dòng bên dưới ghi điện áp cực đại (Umax) mà tụ chịu được.

**Nguyên lí hoạt động:** Tụ gốm là tụ không phân cực, không quy định cực tính, đấu nối thoải mái và các mạng AC hay DC. Thường sử dụng trong các mạch có tần số cao hay mạch lọc nhiễu.

Tụ có kí hiệu là C (Capacitor)



Tụ mắc nối tiếp. Tụ mắc song song.

Hình 2.11. Tụ trong mạch nối tiếp và song song.

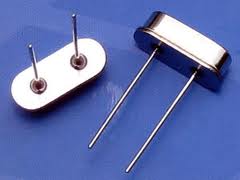
\*Tụ hóa :



Hình 2.12. Hình ảnh thực tế tụ hóa.

Tụ hóa là tụ có phân cực âm và dương. Tụ được chế tạo từ bản cực nhôm và cực dương có bề mặt hình thành lớp Oxit nhôm và lớp bọt khí có tính cách điện để làm chất điện môi. Lớp Oxit nhôm rất mỏng nên điện dung của tụ lớn từ 1µF đến 10.000µF.

# 2.5. Thạch anh



Hình 2.13.Hình dạng thạch anh

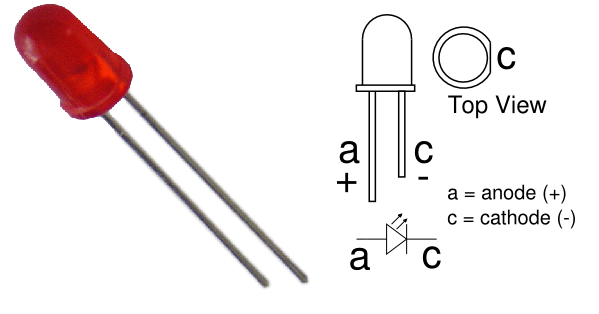
Thạch anh là một vật chất cứng, trong suốt, có trọng lượng riêng 2.649 kg/m3 (1.531 oz/in3), nhiệt độ nóng chảy ở 1750 °C (3182 °F).

Linh kiện thạch anh làm bằng tinh thể đá thạch anh được mài phẳng và chính xác. Linh kiện thạch anh làm việc dựa trên hiệu ứng áp điện. Hiệu ứng này có tính thuận nghịch. Khi áp một điện áp vào 2 mặt của thạch anh, nó sẽ bị biến dạng. Ngược lại, khi tạo sức ép vào 2 bề mặt đó, nó sẽ phát ra điện áp.

Tác dụng của thạch anh trong mạch dùng để tạo dao động cho ra tần số ổn định hơn.

Ở trong mạch này ta dùng thạch anh 11.0592 MHZ.

# 2.6.LED

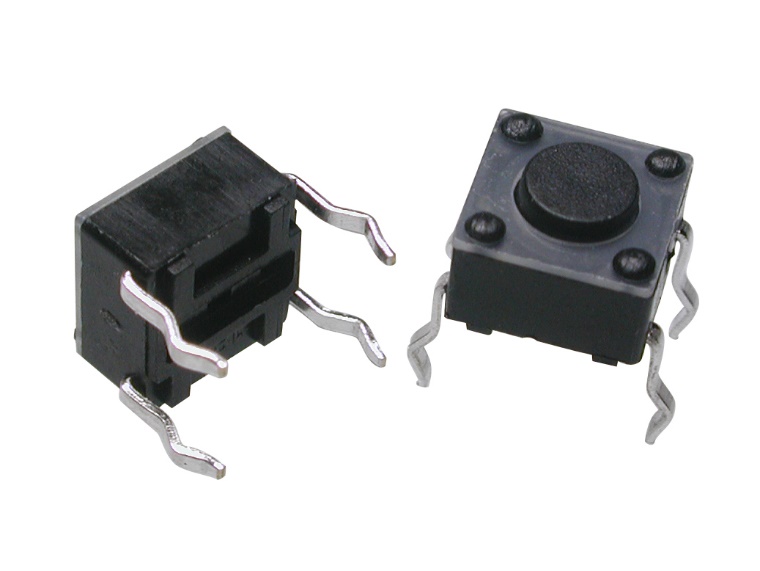


Hình 2.14. Hình ảnh thực tế led, kí hiệu và phân cực.

LED (viết tắt của *Light Emitting Diode*, có nghĩa là điốt phát quang) là các [điốt](http://vi.wikipedia.org/wiki/%C4%90i%E1%BB%91t) có khả năng phát ra [ánh sáng](http://vi.wikipedia.org/wiki/%C3%81nh_s%C3%A1ng)hay tia [hồng ngoại](http://vi.wikipedia.org/wiki/Tia_h%E1%BB%93ng_ngo%E1%BA%A1i), [tử ngoại](http://vi.wikipedia.org/wiki/T%E1%BB%AD_ngo%E1%BA%A1i). Cũng giống như điốt, LED được cấu tạo từ một khối [bán dẫn loại p](http://vi.wikipedia.org/wiki/Ch%E1%BA%A5t_b%C3%A1n_d%E1%BA%ABn) ghép với một khối [bán dẫn loại n](http://vi.wikipedia.org/wiki/Ch%E1%BA%A5t_b%C3%A1n_d%E1%BA%ABn).

LED thường được dùng trong các mạch báo hiệu, chỉ thị trạng thái của mạch như báo nguồn, trạng thái thuận hay ngược.

# 2.7. Nút bấm

****

Hình 2.15. Hình dạng nút bấm

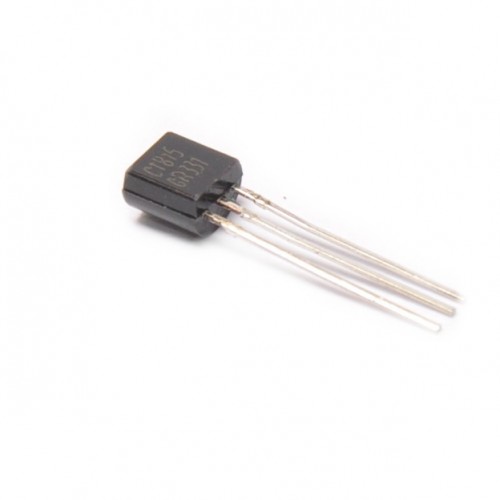
Đặc điểm:

* Kiểu chân: DIP(4)
* Kích thước: 6x6x5 mm
* Điện áp hoạt động: DC 12V/0.5A

Tác dụng trong mạch dùng để:

* RESET chương trình về ban đầu

2.8. **Transistor C1815**

****

Hình 2.7 Transistor C1815

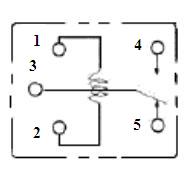
Transistor C1815 là transistor thuộc loại transistor NPN. C1815 có Uc cực đại bằng 50V dòng Ic cực đại bằng 150mA.

Ứng dụng của Transitor C1815: Điều khiển tần số âm chung cho các mạch ứng dụng khuếch đại, kích điện áp, đóng mở như công tắc điện tử.

Kiểu chân và kích thước Transitor C1815 :  
- C1815 có kiểu chân là T092 có nghĩa là kiểu chân cắm.  
- C1815 la dòng transistor ngược NPN .

Thông số kỹ thuật Transitor C1815 :  
- Điện áp cực đại : 50V.  
- Dòng cực đại : 150mA.  
- Hệ số khuếch đại ~ 25-100.  
- Khối lượng : 0.21 g.

# 2.9. Rơ le



Hình 2.6 Hình dáng rơ le và sơ đồ chân

Rơle (relay) là một chuyển mạch hoạt động bằng điện. Dòng điện chạy qua cuộn dây của rơle tạo ra một từ trường hút lõi sắt non làm thay đổi công tắc chuyển mạch. Dòng điện qua cuộn dây có thể được bật hoặc tắt vì thế rơle có hai vị trí chuyển mạch qua lại.

**2.10 Đầu cắm nguồn 5V**



Hình 2.8 Đầu cắm nguồn 5V

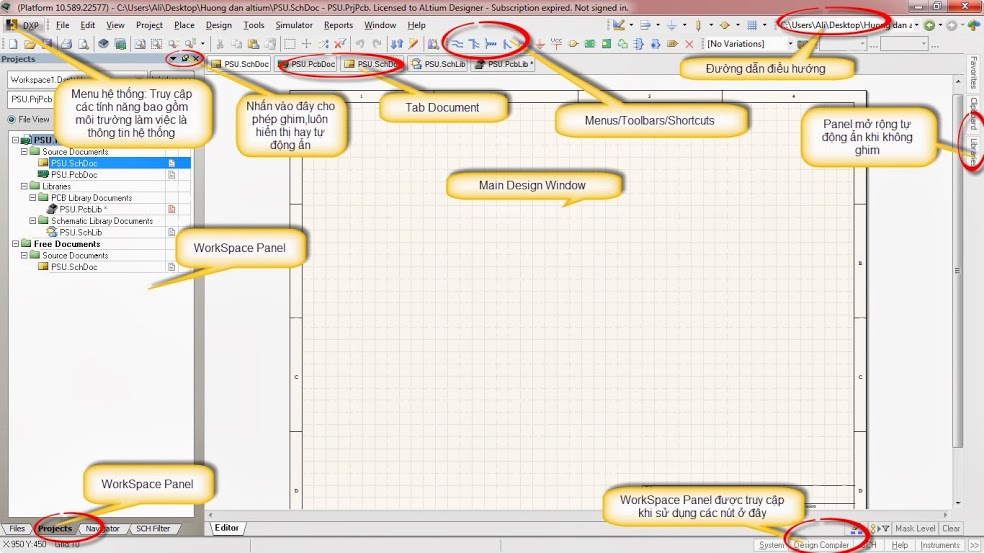
Đặc điểm:

* Lõi 2.1 mm
* Đường kính lỗ: 5.5mm

# CHƯƠNG III. THIẾT KẾ VÀ THI CÔNG MẠCH THỰC TẾ

# 3.1. Giới thiệu phần mềm Altium

**Altium Designer** là một trong những phần mềm tự động hóa thiết kế PCB và điện tử cho các bo mạch in tốt và phổ biến nhất hiện nay. Nó được phát triển bởi công ty phần mềm Altium Limited có trụ sở tại Úc.



Hình 3.1 Giao diện phần mềm altium

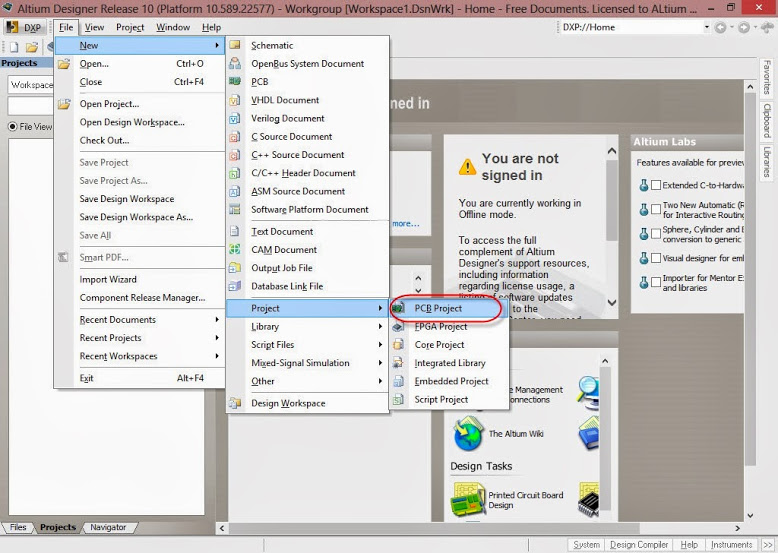
* Altium cung cấp một môi trường làm việc với số lượng phím tắt tương đối lớn và hoàn chỉnh. Để ý các phím tắt là cái kí tự có gạch chân trong các MenuBar. Tận dụng điều đó giúp thiết kế nhanh và chuyên nghiệp hơn.
* Chú ý một số khái niệm đã nêu trong bài tổng quan về mạch in và các tiêu chuẩn khi thiết kế mạch in.
* Để làm quen dần với phím tắt, các phím tắt được trình bày sau mỗi thao tác, xuất hiện trong dấu ngoặc đơn. Tập sử dụng các phím tắt và ghi nhớ, các mạch sau này thực hiện dễ dàng và thao tác nhanh hơn rất nhiều.
* Mỗi linh kiện sẽ có hình dáng và kích thước riêng, tùy loại linh kiện mà cấu tạo Footprint sẽ khác nhau.Ví dụ linh kiện hàn xuyên mạch vẽ trên Multiplayer để hiện đầy đủ các lớp, kích thước lỗ khoan. Đối với linh kiện dán thì xuất hiện lớp Top Paster, được thực hiện trên một mặt của mạch.
* Các layer:
  + Top layer: Lớp đường mạch ở mặt trên.
  + Bottom layer: Lớp đường mạch ở mặt dưới.
  + Top overlay: Hiển thị thông tin linh kiện ở mặt trên.
  + Bottom overlay: Hiển thị thông tin linh kiện ở mặt dưới.
  + Top paste: được tạo ra bởi các chân của linh kiển dán ở mặt trên.
  + Bottom paste: được tạo ra bởi các chân của linh kiển dán ở mặt dưới.
  + Top solder: Lớp Solder Mask ở mặt trên, xem thêm trong bài tổng quan về mạch in.
  + Bottom solder:Lớp Solder Mask ở mặt dưới, xem thêm trong bài tổng quan về mạch in.
  + Keep out layer: đường giới hạn board, board được cắt theo nhưng đường vẽ trên lớp này.

***Khởi tạo Project và tùy chỉnh giao diện làm việc***

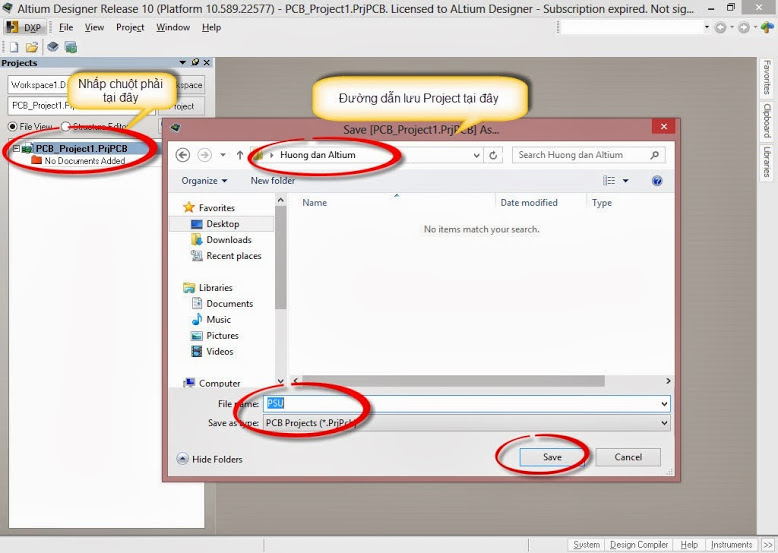
Để thiết kế mạch in với Altium cần khởi tạo các cơ bản như sau:

Khởi tạo một dự án PCB trong đó cơ bản bao gồm:

* Schematic : Sơ đồ mạch nguyên lý
* PCB : Sơ đồ mạch in
* Schematic Library : Thư viện dùng cho mạch nguyên lý
* PCB Library: Thư viện dùng cho mạch in.
* Tạo project mới
  + Từ giao diện chính của chương trình Chọn File->New->Project->PCB Project.

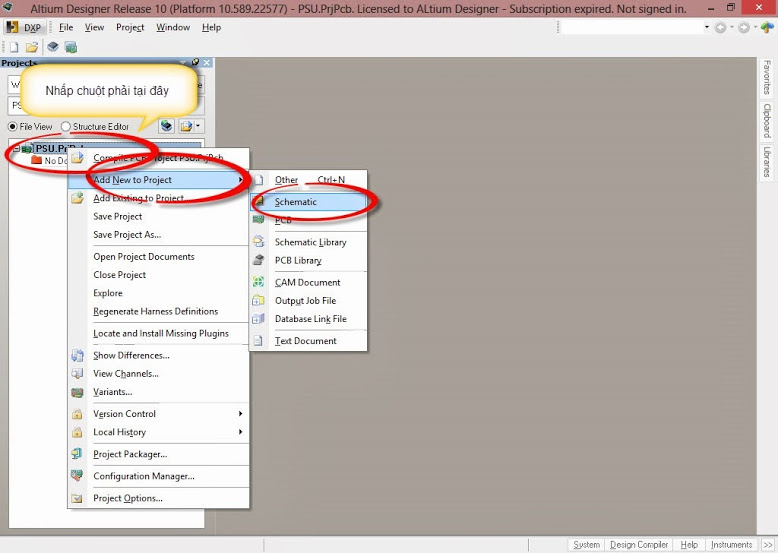
Hình 3.2 Tạo project mới trong Altium

* + Trên Workspace nhấn chuột phải vào PCB\_Project1 đã tạo chọn Save Project As…
  + Tại đây chọn thư mục để lưu Project, ở đây lưu với mạch tên mà bận muốn. Lưu ý việc lưu tên và đường dẫn project sẽ thuận tiện làm việc sau này.



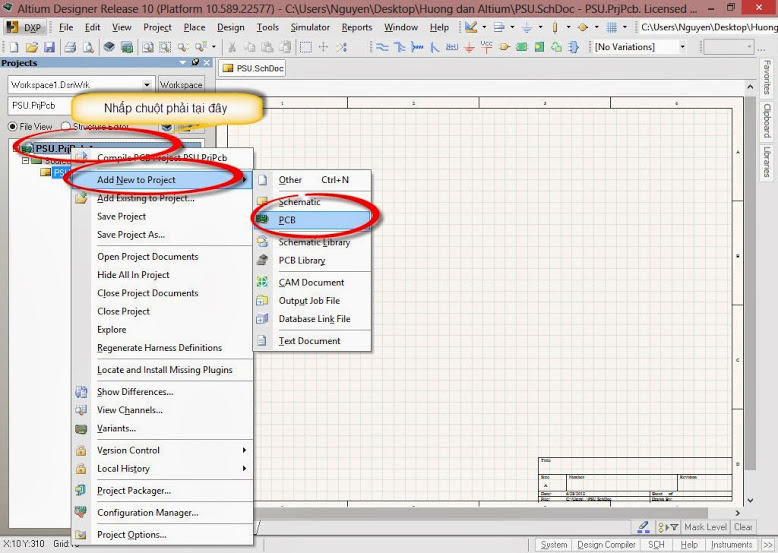
Hình 3.2 Lưu project mới tạo

* ***Tạo mới 1 schematic:***
  + Nhấp chuột phải trong giao diện Workspace chọn Add New To Project -> Schematic.



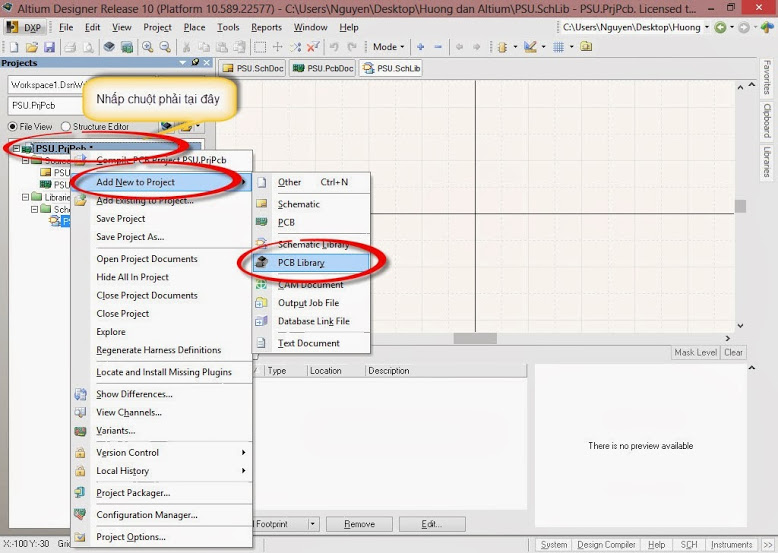
Hình 3.3 Tạo file schematic

* Sau đó lưu lại file Schematic tương tự như Project, lưu trong cùng một thư mục để dễ quản lý. Nhấp chuột phải vào Sheet1.SchDoc trong giao diện WorkSpace chọn Save As, lưu lại .
* Lưu ý rằng nếu bạn không lưu để dễ quản lý thì sau này khó làm việc.
* Tạo mới 1 PCB (sơ đồ thực hiện trên mạch in).
  + Nhấp chuột phải trong giao diện Workspace chọn Add New To Project -> PCB.



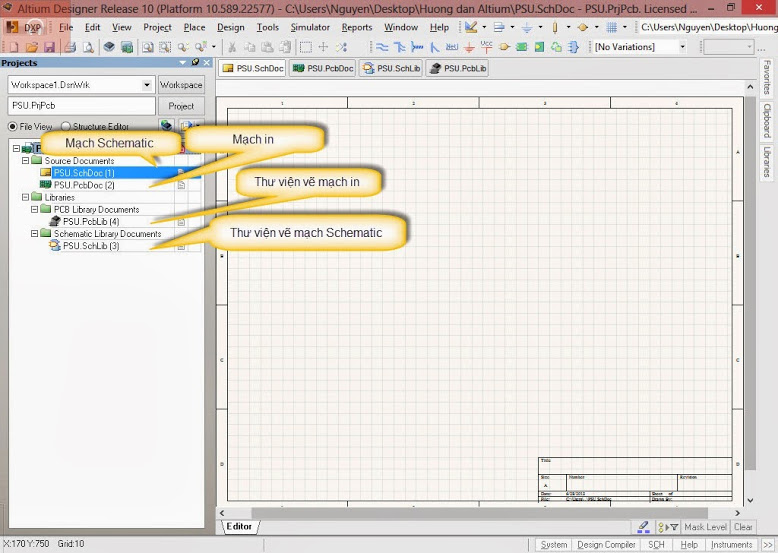
Hình 3.4 Tạo file PCB

* + Sau đó lưu lại file Schematic tương tự như Project, lưu trong cùng một thư mục để dễ quản lý. Nhấp chuột phải vào PCB1.PcbDoc trong giao diện WorkSpace chọn Save As, sau đó lưu lại.
  + Lưu ý là nếu chưa lưu các bạn sẽ không Update được từ sơ đồ mạch nguyên lý sang mạch in.
* Tạo thư viện vẽ Footprint để thực hiện trên mạch in.
  + Trong giao diện Workspace nhấp chuột phải chọn Add New To Project -> PCB Libray.



Hình 3.5 Tạo thư viện

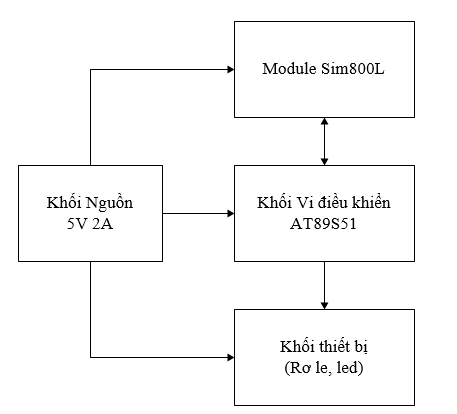
* Sau đó các lưu lại thư viện
* Và đây là giao diện cuối cùng sau khi đã tạo xong một Project.



Hình 3.6 Project hoàn chỉnh

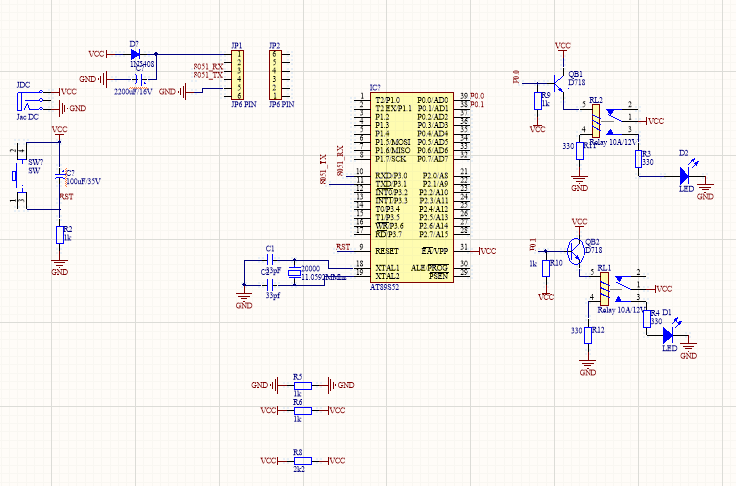
# 3.2. Thiết kế mạch

## 3.2.1. Thiết kế mô hình hệ thống



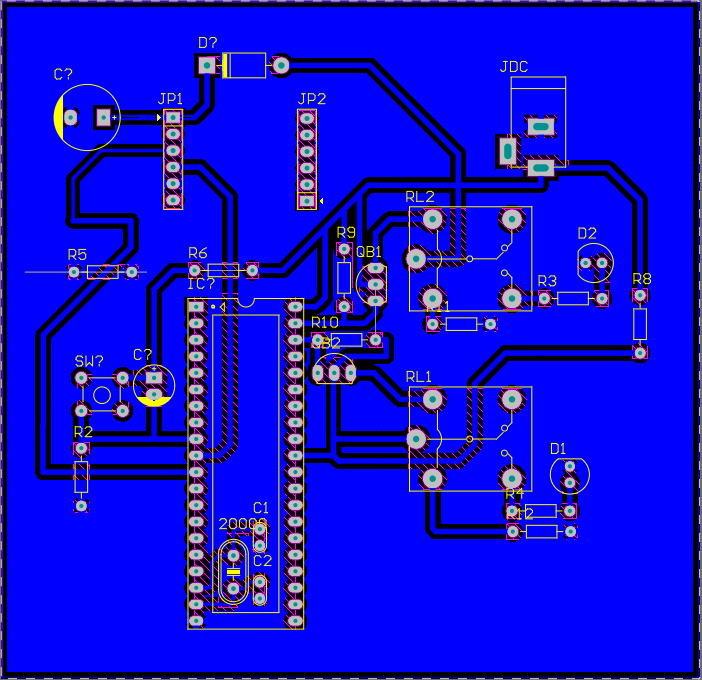
Hình 3.1. Sơ đồ khối hệ thống

**3.2.2 Thiết kế Mạch nguyên lý**

****

Hình 3.2 Sơ đồ mạch nguyên lý

## 3.2.3. Thiết kế mạch in



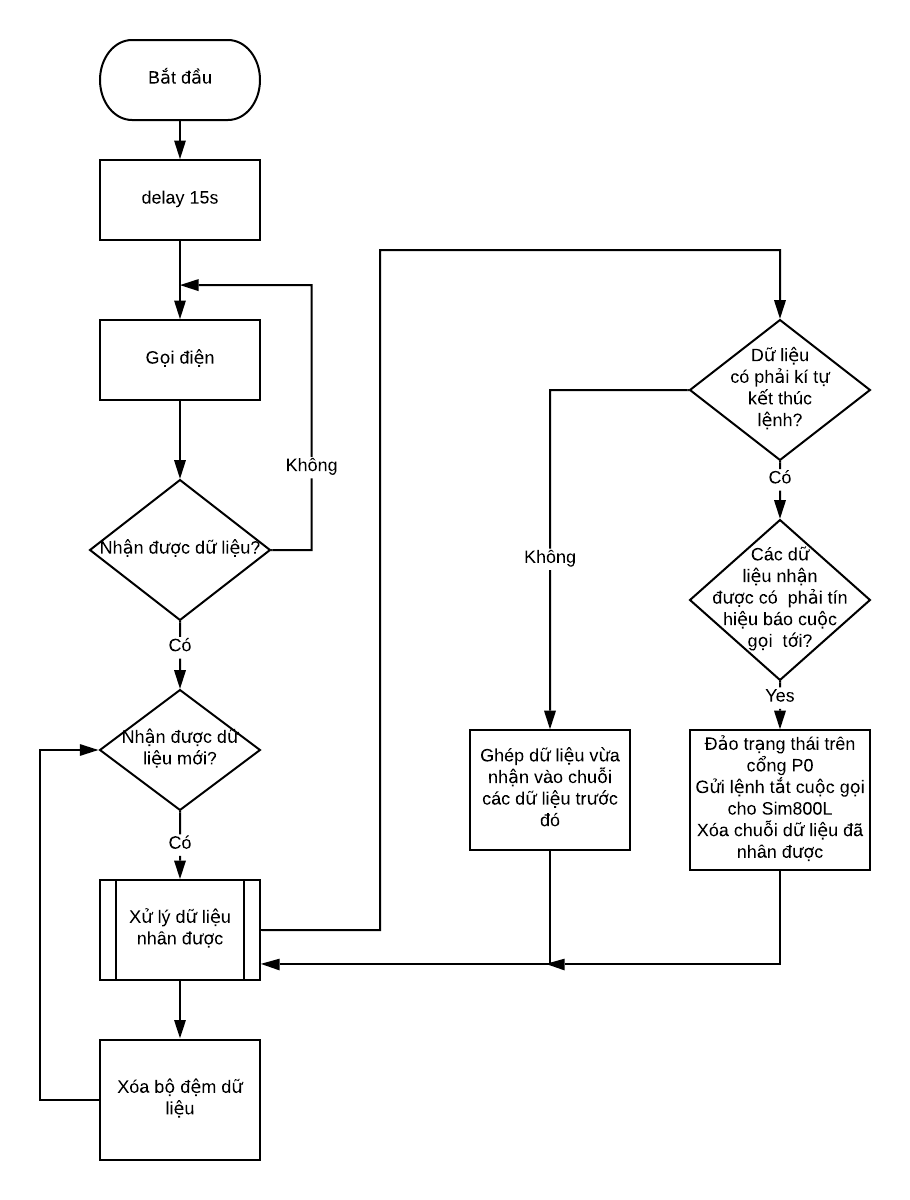
Hình 3.3 Sơ đồ mạch in

## 3.2.4. Nguyên lý hoạt động

- Khi vừa được cấp nguồn, hệ thống sẽ đợi cho tới khi ổn định, sau đó thực hiện gọi tới 1 số điện thoại được cài đặt trước để kiểm tra việc kết nối với module Sim800L.

- Sau khi thực hiện được cuộc gọi, ta có thể bắt đầu điều khiển bật tắt thiết bị bằng việc gọi đến số điện thoại của sim được gắn trên module Sim800L. Gọi lần 1 thiết bị sẽ được bật, gọi lần tiếp theo thiết bị sẽ tắt.

## 3.2.5. Lưu đồ thuật toán

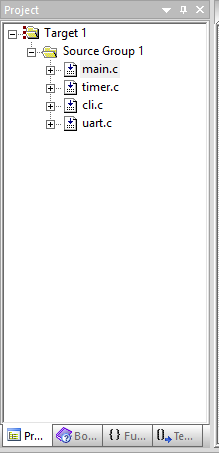


2.13 Hình ảnh lưu đồ thuật toán

## 3.2.6 Mã code chương trình cho vi điều khiển AT89S51

Chương trính được lập trình bằng ngôn ngữ lập trình C, sử dụng IDE KeilC V4 để lập biên dịch chương trình.

Tổ chức chương trình gồm 4 phần: main.c, timer.c, uart.c, cli.c như sau:



Mỗi phần tương ứng với một công việc riêng:

* Main.c: chứa hàm main của chương trình, chương trình bắt đầu chạy từ đây.
* Timer.c: chứa source code cầu hình timer0 và thiết lập hàm delay.
* Uart.c: chứa source code cầu hình uart và các nhận truyền nhận dữ liệu giứa vi điều khiển và module Sim800L.
* Cli.c: chứa source code xử lý dữ liệu nhận được từ cổng truyền thông nối tiếp.

*Sau đây là source code của từng phần:*

**Main.c:**

#include <REGX51.H>

#include "./uart.h"

#include "./cli.h"

unsigned char i = 0;

void main()

{

P0 = 0x00;

P1 = 0x00;

P2 = 0xff;

uart\_init();

timer0\_init();

write\_index = 0;

memset(buffer,0,BUFFER\_SIZE);

\_data = 0;

delay(15000);

do{

uart\_send\_str("ATD+84982634613;\r\n");

}while(\_data !=0);

\_data = 0;

while(1)

{

if(\_data != 0)

{

prepare\_cli(\_data));

\_data = 0;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INTERRUPT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer0\_ISR(void) interrupt 1

{

tick++;

TH0 = 0xFC;

TL0 = 0x66;

}

/\* UART receive data interrupt \*/

void uart\_ISR(void) interrupt 4

{

if(RI == 1)

{

\_data = SBUF;

RI = 0;

}

}

**Timer.c: bao gồm 2 file timer.h và timer.c**

**File timer.h:**

#ifndef \_TIMER\_H\_

#define \_TIMER\_H\_

#include <REGX51.H>

extern volatile unsigned long tick;

void timer0\_init(void);

void delay(unsigned long time);

#endif

**File timer.c:**

#include "timer.h"

volatile unsigned long tick = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TIMER \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer0\_init() // 1ms

{

TMOD &= 0xF0;

TMOD |= 0x01; // 16 bit

TH0 = 0xFC;

TL0 = 0x67;

IE |= 0x82; // enable timer0 interrupt

TR0 = 1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* delay function \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void delay(unsigned long time)

{

unsigned long current\_tick = tick;

while(!(tick - current\_tick == time));

}

**Uart.c:** *gồm 2 file là uart.h và uart.c*

**File uart.h:**

#ifndef \_UART\_H\_

#define \_UART\_H\_

#include <REGX51.H>

#define BUFFER\_SIZE 50

extern volatile char \_data;

extern char buffer[BUFFER\_SIZE];

extern unsigned char write\_index;

void uart\_init(void);

void uart\_send\_byte(char data\_);

void uart\_send\_str(char\* str);

#endif

**File uart.c:**

#include "uart.h"

volatile char \_data = 0;

char buffer[BUFFER\_SIZE] = "";

unsigned char write\_index = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UART \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void uart\_init()

{

TMOD &= 0x0F;

TMOD |= 0x20; // timer1 mode 2 auto update

SCON = 0x50; // uart 8bit baudrate is selected by timer1

TH1 = 0xFD; //9600

IE |= 0x90; // enable interrupt uart

TR1 = 1;

}

void uart\_send\_byte(char data\_)

{

SBUF = data\_;

while(TI ==0){}

TI = 0;

}

/\* UART send string \*/

void uart\_send\_str(char\* str)

{

while(\*str)

{

uart\_send\_byte(\*str);

str++;

}

}

**cli.c:** *gồm 2 file cli.h và cli.c*

**cli.h:**

#ifndef \_CLI\_H\_

#define \_CLI\_H\_

#include <REGX51.H>

#include <stdlib.h>

#include <string.h>

#include "uart.h"

#include "timer.h"

#define DELIMITED "\_ \n\r"

char prepare\_cli(char rx\_data);

void parse\_cli(unsigned char\* buffer, unsigned char len\_buffer);

#endif

**cli.c:**

#include "cli.h"

/\* function : prepare\_cli \*/

char prepare\_cli(char rx\_data)

{

char result = 0;

if((rx\_data == 13 || rx\_data == 10))

{

parse\_cli(buffer,write\_index);

write\_index = 0;

memset(buffer,0,BUFFER\_SIZE);

result = 1;

}

else

{

buffer[write\_index] = rx\_data;

write\_index++ ;

}

return result;

}

/\* function : parse\_cli \*/

void parse\_cli(unsigned char\* buffer, unsigned char len\_buffer)

{

unsigned char\* temp\_cmd = NULL;

temp\_cmd = strtok(buffer,DELIMITED);

while(temp\_cmd != NULL)

{

if(!strcmp(temp\_cmd,"Ring") || !strcmp(temp\_cmd,"RING"))

{

uart\_send\_str("ATH\r\n");

P0 = ~P0;

break;

}

else

{

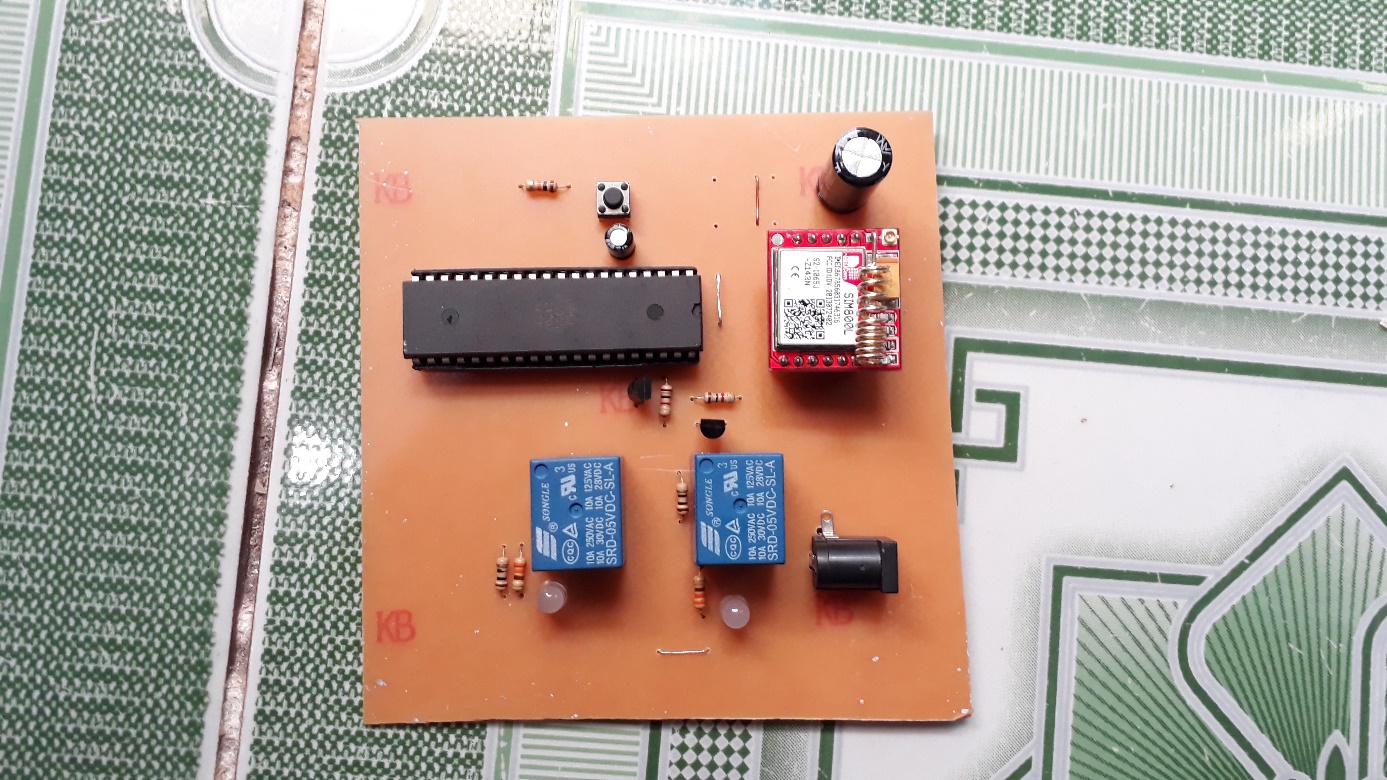
temp\_cmd = strtok(NULL,DELIMITED);

}

}

}

3.2.8. Mô hình mạch thực tế



Hình 3.8. Mô hình mạch thực tế.

# KẾT LUẬN

Sau một thời gian tìm hiểu và nghiên cứu về đề tài**:*“Mạch điều khiển thiết bị sử dụng module Sim800L”***. Quá trình hoàn thành không thể tránh khỏi những sai sót. Cùng với sự giúp đỡ nhiệt tình của cô em đã hoàn thành bài báo cáo.

Qua đây một phần nào cũng giúp chúng em hiểu rõ về ứng dụng và những kiến thức cơ bản của kỹ thuật vi xử lý trong thực tế, cũng như chức năng của các vi mạch. Trong những năm gần đây công nghệ vi mạch điện tử phát triển rất mạnh mẽ cho ra đời các vi mạch với mọi kích thước, đa dạng về chức năng với giá thành hợp lý, khả năng lập trình ngày càng cao đã mang lại những thay đổi sâu sắc trong ngành kĩ thuật điện tử, mạch số ở những mức độ khác nhau đã và đang thâm nhập vào tất cả các thiết bị điện tử thông dụng, chuyên dụng. Vì vậy sự ra đời kỹ thuật vi xử lý và kĩ thuật số là cơ sở cho ngành kỹ thuật điện tử, kỹ thuật máy tính, tin học… bài báo cáo này nhằm giúp và đáp ứng thêm nhu cầu tiếp cận với vi mạch số. Mục đích ứng dụng trong thực tế.

Qua quá trình thiết kế, chế tạo không thể tránh khỏi những sai sót và khuyết điểm. Vì vậy em rất mong được sự đóng góp ý kiến của thầy và các bạn, từ đó em có thể rút ra được những kinh nghiệm cho bản thân đồng thời tìm ra những nhược điểm của bài viết này. Qua đó sẽ giúp em hoàn chỉnh hơn, tối ưu hơn bài báo cáo này.

# TÀI LIỆU THAM KHẢO

* [http://www.alldatasheet.com](http://www.alldatasheet.com/)
* [http://www.luanvan.net.vn](http://www.luanvan.net.vn/)
* [http://www.dientuvietnam.net](http://www.dientuvietnam.net/)
* [http://www.tailieu.vn](http://www.tailieu.vn/)

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

*Thái Nguyên, Ngày tháng năm 2018*

**Giảng viên hướng dẫn**