

# MỤC LỤC

<b>LỜI MỞ ĐẦU</b> .....	1
<b>CHƯƠNG 1. GIỚI THIỆU CHUNG</b> .....	2
1.1. MỞ ĐẦU .....	2
1.2. HỆ THỐNG ĐIỀU KHIỂN TỪ XA .....	2
1.3. TỔNG QUÁT VI ĐIỀU KHIỂN 8051 .....	4
1.3.1. Giới thiệu lịch sử của 8051 .....	4
1.3.2. Các phiên bản của 8051.....	4
1.3.3. Các hãng sản xuất .....	5
1.3.3.1. Hãng Atmel .....	5
1.3.3.2. Hãng Philips.....	6
1.3.3.3. Hãng Dallas Semiconductor .....	6
1.3.4. Cấu trúc vi điều khiển 8051 .....	7
1.3.4.1. Cấu trúc phần cứng giao tiếp bên ngoài.....	7
1.3.4.2. Cấu trúc bên trong vi điều khiển 8051 .....	10
1.4. TỔNG QUÁT VỀ TÍN HIỆU HỒNG NGOẠI.....	22
1.4.1. Khái niệm.....	22
1.4.2. Nguồn phát tia hồng ngoại .....	23
1.4.3. Bộ phát tín hiệu hồng ngoại .....	24
1.4.4. Bộ thu tín hiệu hồng ngoại .....	25
1.5. TỔNG QUÁT BỘ ĐIỀU KHIỂN REMOTE PHÁT TIA IR .....	25
1.5.1. Giới thiệu.....	25
1.5.2. Giải mã tín hiệu điều khiển TV SHARP .....	26
<b>CHƯƠNG 2. XÂY DỰNG MÔ HÌNH PHẦN CỨNG HỆ THỐNG</b> ....	36
2.1. MỞ ĐẦU .....	36
2.2. SƠ ĐỒ KHỐI HỆ THỐNG .....	36
2.3. CHỨC NĂNG VÀ HOẠT ĐỘNG CÁC THÀNH PHẦN.....	37
2.3.1. Khối nguồn.....	37
2.3.2. Khối keyboard kiểm tra.....	38
2.3.3. Khối phát tín hiệu IR .....	38
2.3.4. Khối thu tín hiệu IR.....	39
2.3.5. Khối xử lý tín hiệu.....	40

2.3.6. Khối đệm dòng .....	40
2.3.7. Khối chấp hành .....	41
2.3.8. Khối hiển thị.....	42
2.4. LIỆT KÊ, TÍNH TOÁN CÁC LINH KIỆN TRÊN MẠCH .....	42
2.4.1. Liệt kê linh kiện .....	42
2.4.2. Tính toán lựa chọn các linh kiện.....	42
2.5. SƠ ĐỒ NGUYÊN LÝ VÀ MẠCH IN.....	43
2.6. NGUYÊN LÝ HOẠT ĐỘNG HỆ THỐNG .....	47
2.7. ẢNH CHỤP MÔ HÌNH THỰC .....	48
NHẬN XÉT .....	48
<b>CHƯƠNG 3. XÂY DỰNG CHƯƠNG TRÌNH PHẦN MỀM ĐIỀU</b>	
<b>KHIỂN .....</b>	<b>49</b>
3.1. MỞ ĐẦU .....	49
3.2. HỢP NGỮ ASSEMBLY .....	49
3.3. LƯU ĐỒ THUẬT GIẢI .....	50
3.3.1. Lưu đồ thuật giải Keyboard .....	50
3.3.1. Lưu đồ thuật giải điều khiển bằng Remote TV Sharp .....	51
3.4. SOẠN THẢO VÀ BIÊN DỊCH CHƯƠNG TRÌNH.....	51
3.5. CHƯƠNG TRÌNH ĐIỀU KHIỂN .....	54
3.5.1. Chương trình kiểm tra keyboard trên mạch .....	54
3.5.2. Kết quả chạy kiểm tra mạch.....	56
3.5.3. Chương trình khi điều khiển bằng Remote TV SHARP .....	56
3.5.4. Kết quả chạy trên mô hình thực .....	60
NHẬN XÉT .....	63
<b>KẾT LUẬN.....</b>	<b>63</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>65</b>

## LỜI MỞ ĐẦU

Vào năm 1980 Intel công bố chip 8051(80C51), bộ vi điều khiển đầu tiên của họ vi điều khiển MCS-51. Nó bao gồm 4KB ROM, 128 byte RAM, 32 đường xuất nhập, 1 port nối tiếp và 2 bộ định thời 16 bit. Tiếp theo sau đó là sự ra đời của chip 8052,8053,8055 với nhiều tính năng được cải tiến , và được ứng dụng trong nhiều lĩnh vực như điều khiển tự động ,máy tính , cảm biến, quảng cáo ...

Ngày nay, với những ứng dụng khoa học tiên tiến, thế giới đang thay đổi từng ngày, cũng như sự thay đổi của kỹ thuật điện tử đã tạo ra hàng loạt những thiết bị thông minh có tính nổi bật nhanh và độ chính xác cao gọn nhẹ, phù hợp với xu thế phát triển hiện nay.

Một trong những ứng dụng rất hay của chip vi điều khiển AT89C51 là điều khiển các thiết bị dùng sóng mang. Do vậy xuất phát từ ứng dụng quan trọng của khoa học kỹ thuật , em đã chọn đề tài **“Thiết Kế Mạch Thu Phát Hồng Ngoại Điều Khiển Thiết Bị Từ Xa Sử Dụng Vi Điều Khiển 8051”** làm đề án tốt nghiệp của mình. Đề án gồm các nội dung sau:

Chương 1: Giới thiệu chung.

Chương 2: Xây dựng mô hình phần cứng của hệ thống.

Chương 3: Xây dựng chương trình phần mềm điều khiển.

Kết quả của đề án là một bộ sản phẩm thiết bị điều khiển từ xa cho hệ thống bốn đèn chiếu sáng.

Em xin chân thành cảm ơn các thầy cô trong Khoa Điện – Điện tử đã tạo mọi điều kiện thuận lợi cho em trong quá trình học tập cũng như khi thực hiện đề án tốt nghiệp này.

Hải Phòng, ngày 20 tháng 5 năm 2013

**Vũ Văn Duy**

## CHƯƠNG 1.

# GIỚI THIỆU CHUNG

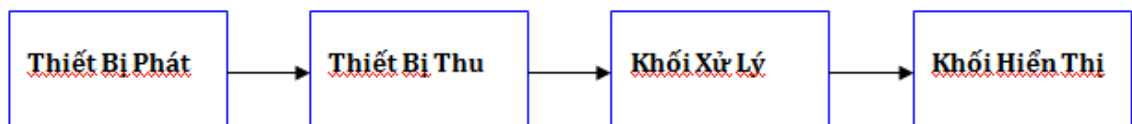
### 1.1. MỞ ĐẦU

Hiện nay có rất nhiều họ Vi điều khiển trên thị trường với nhiều ứng dụng khác nhau, trong đó họ Vi điều khiển họ MCS-51 được sử dụng rất rộng rãi trên thế giới và ở Việt nam.. VĐK8051 có khả năng tương tự như khả năng của vi xử lý, nhưng cấu trúc phần cứng dành cho người dùng đơn giản hơn nhiều. Vi điều khiển ra đời mang lại sự tiện lợi đối với người dùng, họ không cần nắm vững một khối lượng kiến thức quá lớn, kết cấu mạch điện dành cho người dùng cũng trở nên đơn giản hơn nhiều và có khả năng giao tiếp trực tiếp với các thiết bị bên ngoài.

### 1.2. HỆ THỐNG ĐIỀU KHIỂN TỪ XA

Điều khiển từ xa là một hệ thống điều khiển cho phép ta điều khiển các thiết bị từ khoảng cách xa: như điều khiển bằng sóng vô tuyến, qua mạng, điều khiển FM ,tia hồng ngoại ...

Sơ đồ cấu trúc hệ thống :



**Hình 1.1:** Cấu trúc chung hệ thống.

**Thiết bị phát:** Biến đổi lệnh điều khiển thành tín hiệu tương tự và truyền đi.

**Thiết bị thu:** Sẽ đưa tín hiệu từ bên phát sang bên thu.

**Khối xử lý:** Nhận tín hiệu từ đường truyền qua quá trình biến đổi chuyển đến cơ cấu chấp hành.

**Khối hiển thị:** Nhận lệnh từ khối xử lý tín hiệu và thực hiện công việc điều khiển.

Nhưng trong cuộc sống hiện đại hàng ngày, điều khiển từ xa bằng tia hồng ngoại trở lên thông dụng và cần thiết đối với các thiết bị trong gia đình như : ti vi, quạt, máy điều hòa, đầu đĩa, các thiết bị chiếu sáng ...



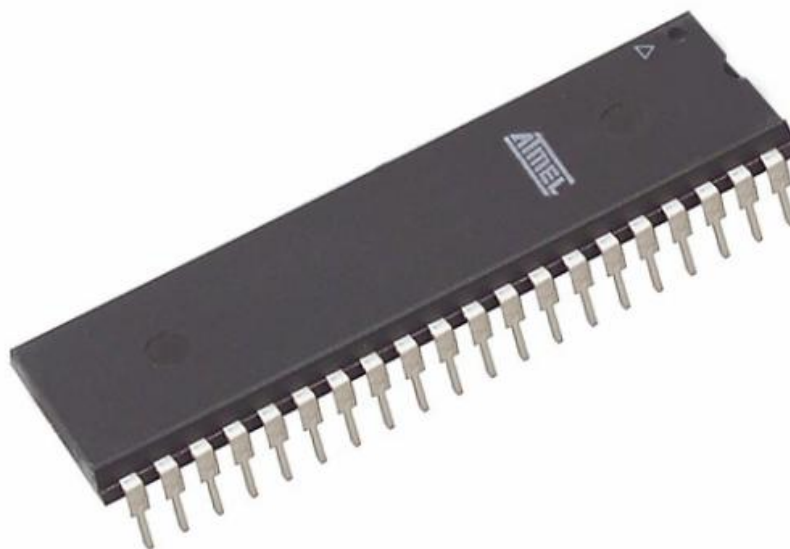
**Hình 1.2:** Ứng dụng điều khiển.

Thay vì phải đứng dậy để bật hay tắt những thiết bị đó, chúng ta chỉ việc ngồi tại chỗ với chiếc điều khiển từ xa trong tay, ta có thể tắt mở những dụng cụ theo ý muốn. Với các thiết bị điều khiển từ xa, ta có thể làm được rất nhiều việc mà không phải mất nhiều công sức, điều này càng có ý nghĩa khi ta mỏi mệt, không tiện đi lại hay đang cần tập trung hết mức vào một công việc nào đó, cũng đơn giản là bạn muốn có cảm giác thực sự làm chủ những thiết bị phục vụ cuộc sống của mình. Đó là một ứng dụng rất thú vị và tiện ích trong cuộc sống hàng ngày.

### **1.3. TỔNG QUÁT VI ĐIỀU KHIỂN 8051**

#### **1.3.1. Giới thiệu lịch sử của 8051**

Vào năm 1980. Hãng Intel giới thiệu một bộ vi điều khiển được gọi là 8051. Bộ vi điều khiển này có 128 byte RAM, 4K byte ROM trên chip, hai bộ định thời, một cổng nối tiếp và 4 cổng vào - ra I/O. Lúc ấy nó được coi là một “hệ thống trên chip”. 8051 là một bộ xử lý 8 bit có nghĩa là CPU chỉ có thể làm việc với 8 bit dữ liệu tại một thời điểm. Dữ liệu lớn hơn 8 bit được chia ra thành các dữ liệu 8 bit để cho xử lý. Tiếp theo sau đó là sự ra đời của chip 8052,8053,8055 với nhiều tính năng được cải tiến.



**Hình 1.3:** Vi Điều Khiển 8051.

VĐK8051 đã trở nên phổ biến sau khi Intel cho phép các nhà sản xuất khác cùng nghiên cứu sản xuất các phiên bản của 8051. Điều này dẫn đến sự ra đời nhiều phiên bản của 8051 với các tốc độ khác nhau và dung lượng ROM trên chip khác nhau. Nhưng tất cả chúng đều tương thích với 8051 ban đầu về tập lệnh.

#### **1.3.2. Các phiên bản của 8051**

Bộ vi điều khiển đầu tiên của họ vi điều khiển MCS-51 được trang bị 4KB ROM, 128 byte RAM, 32 đường xuất nhập, 1 port nối tiếp và 2 bộ

định thời 16 bit. Tiếp theo sau đó là sự ra đời của chip 8052,8053, 8055, được sử dụng rộng rãi trên thế giới cũng như ở Việt Nam.

Hãng Atmel có các chip Vi điều khiển có tính năng tương tự như chip MCS-51 của Intel, các mã số chip được thay đổi chút ít, mã số 80 chuyển thành 89, chẳng hạn 80C51 của Intel khi sản xuất ở Atmel mã số thành 89C51. Tương tự 8052, 8053, 8055 có mã số tương đương ở Atmel là: 89C52, 89C53, 89C55.

**Bảng 1.1:** Các phiên bản của 8051.

<b>Phiên bản</b>	<b>Dung lượng RAM</b>	<b>Dung lượng ROM</b>
89C51	128 byte	4 Kbyte
89C52	128 byte	8 Kbyte
89C53	128 byte	12 Kbyte
89C55	128 byte	20 Kbyte

### **1.3.3. Các hãng sản xuất**

#### **1.3.3.1. Hãng Atmel**

Chip 8051 hãng này có ROM trên chip ở dạng bộ nhớ Flash. Điều này là lý tưởng đối với những phát triển nhanh vì bộ nhớ Flash có thể được xóa trong vài giây. Vì lý do này mà AT89C51 để phát triển một hệ thống dựa trên bộ vi điều khiển yêu cầu một bộ đốt ROM mà có hỗ trợ bộ nhớ Flash. Trong bộ nhớ Flash ta phải xóa toàn bộ nội dung của ROM nhằm để lập trình lại cho nó. Việc xóa bộ nhớ Flash được thực hiện bởi chính bộ đốt ROM.

Chữ “C” trong ký hiệu AT89C51 là CMOS, “12” ký hiệu cho 12 MHZ, “P” là kiểu đóng vỏ DIP và chữ “C” cuối cùng là ký hiệu cho thương mại. AT89C51 - 12PC rất phù hợp cho các đề tài nghiên cứu của sinh viên

**Bảng 1.2:** 8051 của hãng Atmel.

<i>Số linh kiện</i>	<i>ROM</i>	<i>RAM</i>	<i>Chân I/O</i>	<i>Timer</i>	<i>Ngắt</i>	<i>Vcc</i>	<i>Đóng vỏ</i>
AT89C51	4K	128	32	2	6	5V	40
AT89LV51	4K	128	32	2	6	3V	40
AT89C1051	1K	64	15	1	3	3V	20
AT89C2051	2K	128	15	2	6	3V	20
AT89C52	8K	128	32	3	8	5V	40
AT89LV52	8K	128	32	3	8	3V	40

### **1.3.3.2. Hãng Philips**

Một nhà sản xuất của họ 8051 khác nữa là Philips, hãng này có một dải lựa chọn rộng lớn cho các bộ VĐK 8051. Nhiều sản phẩm của hãng đã có kèm theo các đặc tính như các bộ chuyển đổi ADC, DAC, cổng I/O mở rộng.

### **1.3.3.3. Hãng Dallas Semiconductor**

Một phiên bản phổ biến khác nữa của 8051 là DS5000 của hãng Dallas Semiconductor. Bộ nhớ ROM trên chip của DS5000 ở dưới dạng NV-RAM. Khả năng đọc/ ghi của nó cho phép chương trình được nạp vào ROM trên chip trong khi nó vẫn ở trong hệ thống (không cần phải lấy ra). Điều này còn có thể được thực hiện thông qua cổng nối tiếp của máy tính IBM PC.

Một ưu việt của NV-RAM là khả năng thay đổi nội dung của ROM theo từng byte tại một thời điểm. Điều này tương phản với bộ nhớ Flash và EPROM mà bộ nhớ của chúng phải được xoá sạch trước khi lập trình lại cho chúng.



**Bảng 1.3:** 8051 của hãng Dallas Semiconductor.

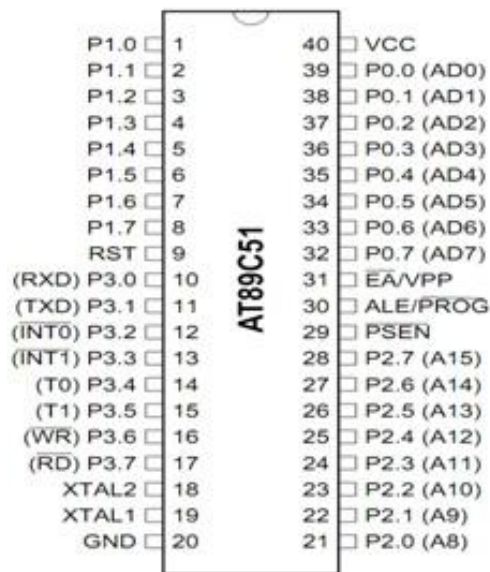
<i>Mã linh kiện</i>	<i>ROM</i>	<i>RAM</i>	<i>Chân I/O</i>	<i>Timer</i>	<i>Ngắt</i>	<i>Vcc</i>	<i>Đóng vỏ</i>
DS5000-8	8K	128	32	2	6	5V	40
DS5000-32	32K	128	32	2	6	5V	40
DS5000T-8	8K	128	32	2	6	5V	40
DS5000T-8	32K	128	32	2	6	5V	40

Chữ “T” đứng sau 5000 là có đồng hồ thời gian thực RTC. RTC tạo và giữ thời gian 1 phút, giờ, ngày, tháng, năm kể cả khi tắt nguồn.

### 1.3.4. Cấu trúc vi điều khiển 8051

#### 1.3.4.1. Cấu trúc phần cứng giao tiếp bên ngoài

**a. Sơ đồ chân.** Hầu hết các hãng sản xuất đóng vỏ 40 chân với hai hàng chân.



**Hình 1.4:** Sơ đồ chân 8051

**Chân VCC:** Chân số 40 là VCC cấp điện áp +5V cho Vi điều khiển.

**Chân GND:** Chân số 20 nối GND.

**Port 0 (P0):** Gồm 8 chân (từ chân 32 đến 39) có hai chức năng:

Chức năng xuất/nhập: Các chân này được dùng để nhận tín hiệu từ bên ngoài vào để xử lý, hoặc dùng để xuất tín hiệu ra bên ngoài.

Chức năng là bus dữ liệu và bus địa chỉ (AD7-AD0) : 8 chân này làm nhiệm vụ lấy dữ liệu từ ROM hoặc RAM.

**Port 1 (P1):** Gồm 8 chân (từ chân 1 đến chân 8), chỉ có chức năng làm các đường xuất/nhập, không có chức năng khác.

**Port 2 (P2):** Gồm 8 chân (từ chân 21 đến chân 28) có hai chức năng:

Chức năng xuất/nhập và chức năng là bus địa chỉ cao (A8-A15): Khi kết nối với bộ nhớ ngoài có dung lượng lớn.

**Port 3 (P3):** Gồm 8 chân (từ chân 10 đến 17):

Chức năng xuất/nhập và chức năng riêng thứ hai như trong bảng sau:

**Bảng 1.4:** Các chức năng riêng của P3.

Bit	Bit	Chức năng
P3.0	RxD	Ngõ vào nhận dữ liệu nối tiếp
P3.1	TxD	Ngõ xuất dữ liệu nối tiếp
P3.2	INT0\	Ngõ vào ngắt cứng thứ 0
P3.3	INT1\	Ngõ vào ngắt cứng thứ 1
P3.4	T0	Ngõ vào của Timer/Counter thứ 0
P3.5	T1	Ngõ vào của Timer/Counter thứ 1
P3.6	WR\	Ngõ điều khiển ghi dữ liệu lên bộ nhớ ngoài
P3.7	RD\	Ngõ điều khiển đọc dữ liệu từ bộ nhớ bên ngoài

#### **Chân RESET (RST) (Chân 9)**

Ngõ vào RST ở chân 9 là ngõ vào Reset dùng để thiết lập trạng thái ban đầu cho vi điều khiển. Hệ thống sẽ được thiết lập lại các giá trị ban đầu nếu ngõ này ở mức 1.

#### **Chân XTAL1 và XTAL2 (Chân 18 và 19)**

Hai chân này có vị trí chân là 18 và 19 được sử dụng để nhận nguồn xung clock từ bên ngoài để hoạt động, thường được ghép nối với thạch anh và các tụ để tạo nguồn xung clock ổn định.

**Chân cho phép bộ nhớ chương trình PSEN\:** (Chân 29)

Dùng để truy xuất bộ nhớ chương trình ngoài. Chân này thường được nối với chân OE\ (outputenable) của ROM ngoài.

Khi vi điều khiển làm việc với bộ nhớ chương trình ngoài, chân này kích hoạt ở mức 0.

Khi thực thi một chương trình ở ROM nội, chân này được duy trì ở mức logic không tích cực (logic 1).

**Chân ALE:** (Chân 30)

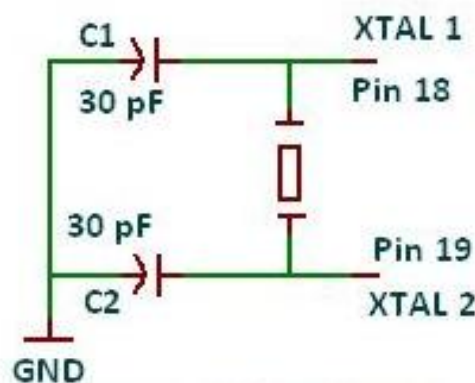
Là chân cho phép chốt địa chỉ khi Vi điều khiển truy xuất bộ nhớ từ bên ngoài. Tín hiệu ở chân ALE dùng làm tín hiệu điều khiển để giải đa hợp các đường địa chỉ và các đường dữ liệu khi kết nối chúng với IC chốt.

**Chân EA\:** (Chân 31)

Là chân dùng để xác định chương trình thực hiện được lấy từ ROM nội hay ROM ngoài. Khi EA nối với logic 1(+5V) thì Vi điều khiển thực hiện chương trình lấy từ bộ nhớ nội. Khi EA nối với logic 0(0V) thì Vi điều khiển thực hiện chương trình lấy từ bộ nhớ ngoài.

**b. Kết nối phần cứng của XTAL1 và XTAL2**

Mạch dao động được đưa vào hai chân này thông thường được kết nối với dao động thạch anh như sau:

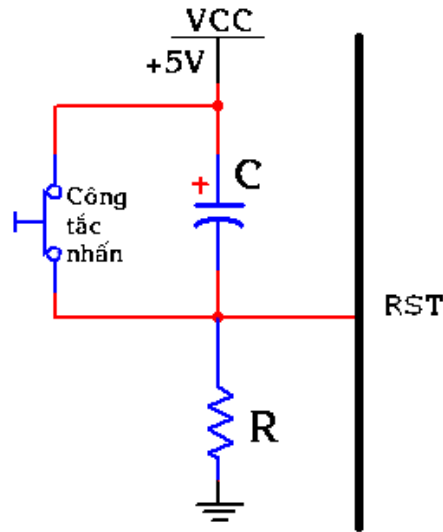


**Hình 1.5:** Kết nối bộ dao động thạch anh.

C1=C2= 30pF dùng ổn định dao động cho thạch anh.

### c. Kết nối phần cứng của chân RESET

Việc kết nối chân RESET đảm bảo hệ thống bắt đầu làm việc khi Vi điều khiển được cấp điện, hoặc đang hoạt động mà hệ thống bị lỗi cần tác động cho Vi điều khiển hoạt động trở lại, hoặc do người sử dụng muốn quay về trạng thái hoạt động ban đầu. Vì vậy chân RESET được kết nối như sau:

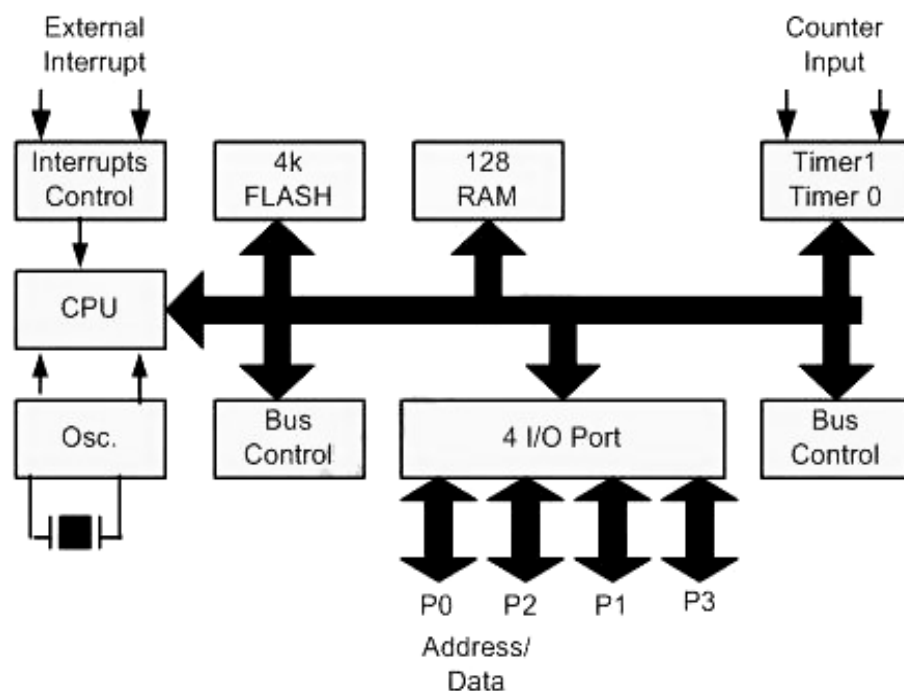


*Hình 1.6:* Kết nối bộ Reset.

Vi điều khiển sử dụng thạch anh có tần số  $f_{xtal} = 12\text{MHz}$ ,  $C=10\mu F$  và  $R=10K\Omega$ .

#### 1.3.4.2. Cấu trúc bên trong vi điều khiển 8051

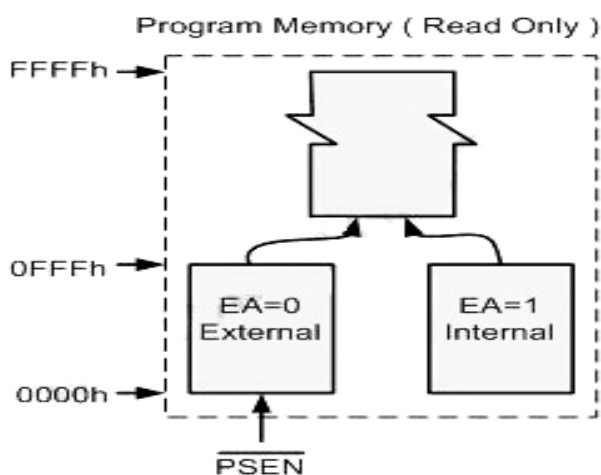
Tất cả các bộ Flash Microcontrollers của Atmel đều tổ chức các vùng địa chỉ tách biệt đối với bộ nhớ chương trình và bộ nhớ dữ liệu, được mô tả trong hình sau:



**Hình 1.7:** Cấu trúc bên trong VDK 8051.

**b. Bộ nhớ chương trình- ROM**

Bộ nhớ ROM dùng để lưu chương trình do người viết chương trình viết ra. Chương trình là tập hợp các câu lệnh thể hiện các thuật toán để giải quyết các công việc cụ thể, chương trình được viết sau đó được đưa vào lưu trong ROM của vi điều khiển, khi hoạt động vi điều khiển truy xuất từng câu lệnh trong ROM để thực hiện chương trình. Trong quá trình hoạt động nội dung ROM là cố định, không thể thay đổi, nội dung ROM chỉ thay đổi khi ROM ở chế độ xóa hoặc nạp chương trình.



**Hình 1.8:** Cấu trúc bộ nhớ chương trình

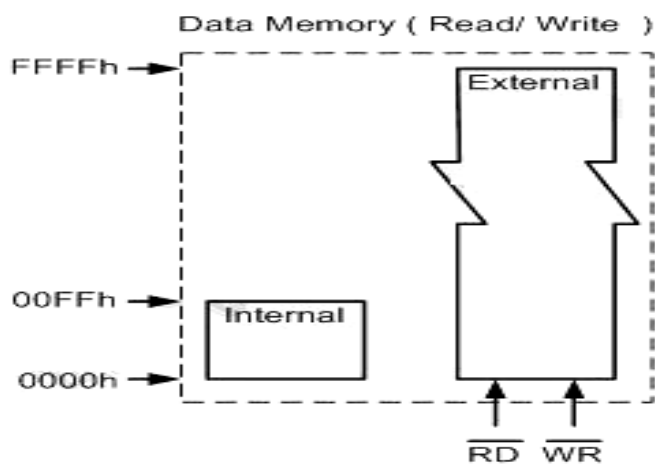
Bộ nhớ ROM được định địa chỉ theo từng Byte, các byte được đánh địa chỉ theo hệ số hexa.

Bộ nhớ ROM của Vi điều khiển có dung lượng tùy vào chủng loại cần dùng, chẳng hạn đối với 89S51 là 4KByte, với 89S52 là 8Kbyte, với 89S53 là 12Kbyte. Ngoài ra có khả năng mở rộng bộ nhớ ROM với việc giao tiếp với bộ nhớ ROM bên ngoài lên đến 64Kbyte. Khi khởi động, CPU bắt đầu thực hiện chương trình ở vị trí 0000H.

### b. Bộ nhớ dữ liệu-RAM

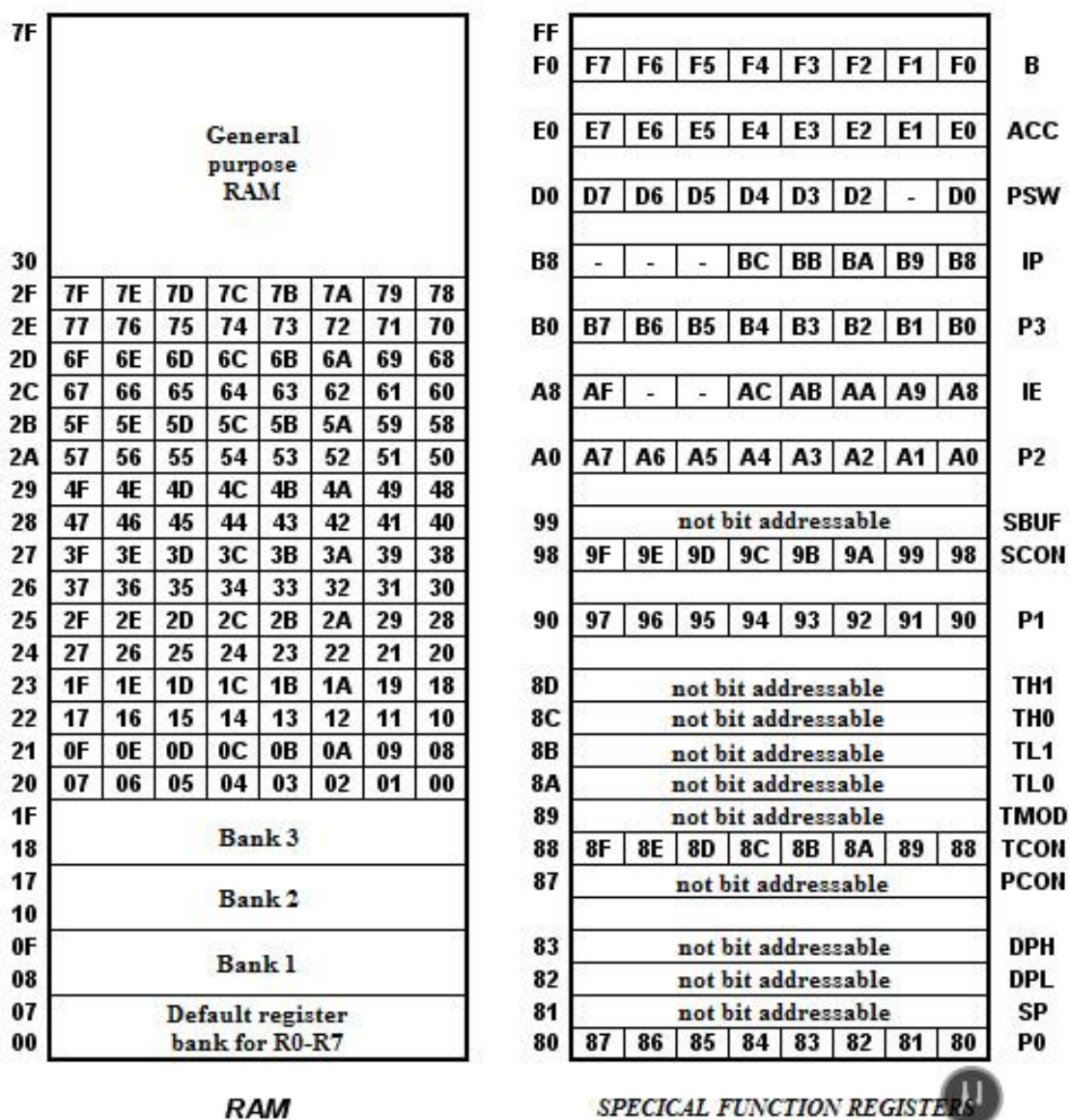
Bộ nhớ RAM dùng làm môi trường xử lý thông tin, lưu trữ các kết quả trung gian và kết quả cuối cùng của các phép toán, xử lý thông tin.

Bộ nhớ dữ liệu (RAM) tồn tại độc lập so với bộ nhớ chương trình. Họ vi điều khiển 8051 có bộ nhớ dữ liệu tích hợp trên chip nhỏ nhất là 128byte địa chỉ từ 00h đến 7Fh. Phạm vi địa chỉ từ 80h đến FFh dành cho SFR. VĐK có thể mở rộng với bộ nhớ dữ liệu ngoài lên tới 64kByte. Khi sử dụng RAM ngoài, CPU sẽ dùng đến các chân RD và WR khi truy cập đến bộ nhớ dữ liệu ngoài. CPU đọc và ghi dữ liệu nhờ tín hiệu trên các chân RD và WR.



**Hình 1.9:** Cấu trúc bộ nhớ dữ liệu.

Cấu trúc bộ nhớ dữ liệu RAM trong chip, được chia thành 128 byte thấp, 128 byte cao. Chi tiết được mô tả trong hình sau:



**RAM**

**SPECIAL FUNCTION REGISTERS**

ViDieukhien.Net

**Hình 1.10:** Bộ nhớ dữ liệu RAM.

### Vùng nhớ 128 Byte thấp

Địa chỉ từ 00đến 7Fh, chia thành 3 vùng:

Địa chỉ từ (00-1F) có độ lớn 32 Byte là 4 băng thanh ghi (bank 0-banh 3) mỗi bank có 8 thanh ghi 8 bit. Các thanh ghi trong mỗi bank có tên gọi từ R0-R7.

Địa chỉ từ (20-2F) có độ lớn 16Byte ,16 byte x 8 bit = 128 bit, cho phép truy cập địa chỉ trực tiếp bằng địa chỉ mức bit.

Địa chỉ từ (30-7F) có độ lớn 80Byte, được dùng cho người dùng để lưu trữ dữ liệu. Đây có thể xem là vùng RAM đa mục đích, có thể truy cập vùng nhớ này bằng địa chỉ trực tiếp hoặc gián tiếp.

**Vùng nhớ 128 Byte cao dành cho SFR**

Được định địa chỉ từ 80 đến FFh gồm các thanh ghi chức năng đặc biệt sau:

**Bảng 1.5:** Địa chỉ thanh ghi chức năng đặc biệt SFR.

F8H									FFH
F0H	B								
E0H	ACC								
D0H	PSW								
B8H	IP								
B0H	P3								
A8H	IE								
A0H	P2								
98H	SCON	SBUF							
90H	P1								
88H	TCON	TMOD	TL0	TL1	TH0	TH1			8FH
80H	P0	SP	DPL	DPH					87H

**Thanh ghi A:** Là thanh ghi tích lũy, dùng để lưu trữ các toán hạng và kết quả của máy tính. ACC (Accumulator) ở địa chỉ 0E0H có độ dài 8.

**Thanh ghi B:** Là thanh ghi tính toán phụ, dùng để thực hiện các phép toán nhân, chia. Thanh ghi B ở địa chỉ 0F0H, có độ dài 8 bit.

**Thanh ghi ngăn xếp SP (Stack Pointer):** Là thanh ghi 8 bit ở địa chỉ 81H. SP chứa địa chỉ của dữ liệu hiện đang ở đỉnh của stack. Giá trị của nó tự động tăng lên khi thực hiện lệnh PUSH trước khi ghi dữ liệu được lưu trữ trong ngăn xếp. Giá trị của nó tự động giảm xuống khi thực hiện lệnh POP. Ngăn xếp có thể đặt ở bất kì vị trí nào trong RAM nhưng khi khởi động lại hệ thống thì con trỏ mặc định sẽ trở tới địa chỉ 07h.



**Thanh ghi trạng thái chương trình (PSW):** Là thanh ghi mô tả toàn bộ trạng thái chương trình đang hoạt động của hệ thống. Địa chỉ là D0H.

**Bảng 1.6:** Mô tả các bit trong thanh ghi PSW.

Bit	Ký Hiệu	Địa Chỉ	Mô tả Bit
PSW.7	CY	D7H	<b>Cờ nhớ</b> (Carry Flag): được Set nếu có Bit nhớ từ Bit 7 trong phép cộng hoặc có Bit mượn cho Bit 7 trong phép trừ.
PSW.6	AC	D6H	<b>Cờ nhớ phụ:</b> được Set trong phép cộng nếu có Bit nhớ từ Bit 3 sang Bit 4 hoặc kết quả trong 4 Bit thấp nằm trong khoảng 0AH->0FH.
PSW.5	FO	D5H	<b>Cờ O:</b> dành cho người sử dụng.
PSW.4	RS1	D4H	<b>Chọn dây thanh ghi</b> (Bit 1)
PSW.3	RS0	D3H	<b>Chọn dây thanh ghi</b> (Bit 0) 00=Bank 0: Địa chỉ 00H->07H 01=Bank 1: Địa chỉ 08H->0FH 10=Bank 2: Địa chỉ 10H->17H 11=Bank 3: Địa chỉ 18H->1FH
PSW.2	OV	D2H	<b>Cờ tràn</b> (Overflow Flag): được Set khi phép toán có dấu có kết quả > +127 hoặc < -128.
PSW.1	-	D1H	Chưa dùng
PSW.0	P	D0H	<b>Cờ kiểm tra chẵn lẻ:</b> được Set hoặc Clear bởi phần cứng sau mỗi 1 chu kỳ lệnh, để chỉ ra rằng có 1 số chẵn hoặc số lẻ Bit 1 trong thanh chứa.

**Con trỏ dữ liệu DPTR:** DPTR là một thanh ghi 16 bit có địa chỉ là 82H (DPL, byte thấp) và 83H (DPH, byte cao), dùng để truy xuất bộ nhớ chương trình ngoài hoặc bộ nhớ dữ liệu ngoài.

**Thanh ghi cổng P0-P3:** Các port xuất/nhập của 8051 bao gồm Port 0 tại địa chỉ 80H, Port 1 tại địa chỉ 90H, Port 2 tại địa chỉ A0H và Port 3 tại địa chỉ B0H. Tất cả các port đều được định địa chỉ từng bit nhằm cung cấp các khả năng giao tiếp mạnh.

**Thanh ghi SBUF:** Là thanh ghi đệm truyền thông nối tiếp được chia thành 2 thanh ghi riêng biệt, thanh ghi đệm phát và thanh ghi đệm thu. Khi dữ liệu được chuyển vào thanh ghi SBUF, dữ liệu sẽ được chuyển vào bộ đệm phát và sẽ được lưu giữ ở đó để biến thành dạng truyền tin nối tiếp. Khi thực hiện việc chuyển dữ liệu từ SBUF ra ngoài, nó sẽ đi ra từ bộ đệm thu.

**Thanh ghi Timer:** 8051 có 2 bộ đếm/định thời để định các khoảng thời gian hoặc để đếm các sự kiện. Các cặp thanh ghi (TH0, TL0) và (TH1, TL1) là các thanh ghi 16 bit tương ứng với các bộ Timer/Counter 0 và 1.

**Thanh ghi chế độ timer TMOD:** Gồm 8 bit chia thành 2 nhóm: 4 bit thấp đặt chế độ hoạt động cho Timer 0 và 4 bit cao đặt chế độ hoạt động cho Timer 1.

**Bảng 1.7:** Mô tả các bit trong thanh ghi TMOD.

Bit	Tên	Time	Mô tả
TMOD.7	GATE	1	Bit điều khiển cổng. Khi GATE =1 bộ time/count 1 chạy khi INTx ở mức cao. Khi GATE =0 bộ time/count 1 chạy khi TRx ở mức cao.
TMOD.6	C/T	1	Bit chọn chế độ timer/counter. 1: Bộ đếm sự kiện. 0: Bộ định khoảng thời gian.
TMOD.5	M1	1	Bit 1 của chế độ ( Mode )
TMOD.4	M0	1	Bit 0 của chế độ 00: Chế độ 0: time 13 bit. 01: Chế độ 1: time 16 bit. 10: Chế độ 2: time tự động nạp lại. 11: Chế độ 3: tách time
TMOD.3	GATE	0	Nt
TMOD.2	C/T	0	Nt
TMOD.1	M1	0	Nt
TMOD.0	M0	0	Nt

**Thanh ghi điều khiển Timer TCON:** Gồm các bit trạng thái và các bit điều khiển bởi Timer 0 và Timer 1.

**Bảng 1.8:** Mô tả các bit trong thanh ghi TCON.

Bit	Ký hiệu	Địa chỉ	Mô tả
TCON.7	TF1	8FH	Cờ báo tràn TIMER 1. Được đặt bởi phần cứng, khi tràn được xóa bằng phần mềm, hay phần cứng khi bộ xử lý chỉ đến chương trình phục vụ ngắt.
TCON.6	TR1	8EH	Bit điều khiển timer 1 chạy, đặt và xóa bằng phần mềm.

TCON.5	TF0	8DH	Cờ báo tràn timer 0, tương tự như timer 1
TCON.4	TR0	8CH	Bít điều khiển chạy cho timer 0, đặt xóa bằng phần mềm.
TCON.3	IE1	8BH	Cờ báo ngắt 1 bên ngoài.
TCON.2	IT1	8AH	Cờ ngắt do timer 1.
TCON.1	IE0	89H	Cờ báo ngắt do timer 0.
TCON.0	IT0	88h	Cờ ngắt do timer 0.

**Thanh ghi IE:** Là thanh ghi cho phép ngắt.

**Bảng 1.9:** Mô tả các bit trong thanh ghi IE.

Bít	Kí hiệu	Địa chỉ bit	Mô tả (1: cho phép, 0: cấm)
IE.7	EA	AFH	Cho phép hoặc cấm toàn bộ
IE.6	EA	AEH	Không được định nghĩa
IE.5	ET2	ADH	Cho phép ngắt từ Time 2
IE.4	ES	ACH	Cho phép ngắt Port nối tiếp
IE.3	ET1	ABH	Cho phép ngắt từ Time 1
IE.2	EX1	AAH	Cho phép ngắt ngoài 1
IE.1	ET0	A9H	Cho phép ngắt từ Time 0
IE.0	EX0	A8H	Cho phép ngắt ngoài 0

### **Các chế độ hoạt động của Timer/Counter**

VĐK 8051 có 4 chế độ hoạt động được mô tả như sau:

**Chế độ 0:** Là chế độ 13 bit bao gồm 8 bit của thanh ghi THx và 5 bit của thanh ghi TLx còn 3 bit cao của thanh ghi TLx không sử dụng. Mỗi lần có xung đếm, giá trị trong thanh ghi 13 bit tăng lên 1. Khi giá trị đếm thay đổi từ 8192 về 0 thì bộ đếm tràn làm cho TFx được đặt lên mức 1.

**Chế độ 1:** Là chế độ 16 bit bao gồm 8 bit của THx và 8 bit của TLx.. Chế độ 0 và chế độ 1 giống nhau nhưng chỉ khác ở số bit đếm. Khi bộ đếm thay đổi từ 65536 về 0, cờ tràn TFx được set lên mức 1. Khi timer tràn, giá

trị của các thanh ghi đếm là 0 ( $THx = 0$  và  $TLx = 0$ ) nên nếu muốn timer hoạt động tiếp thì phải nạp lại giá trị cho các thanh ghi  $THx$  và  $TLx$ .

**Chế độ 2:** Là chế độ 8 bit trong đó sử dụng thanh ghi  $TLx$  để chứa giá trị đếm còn thanh ghi  $THx$  chứa giá trị nạp. Mỗi khi giá trị trong thanh ghi  $TLx$  thay đổi từ 256 về 0 thì cờ  $TFx$  được set lên mức 1 đồng thời giá trị trong thanh ghi  $THx$  được chuyển vào thanh ghi  $TLx$ . Giá trị đếm trong  $TLx$  và  $THx$  chỉ được nạp một lần khi khởi động timer.

**Chế độ 3:** Là chế độ sử dụng các thanh ghi  $TL0$  và  $TH0$  như các bộ định thời độc lập trong đó  $TL0$  điều khiển bằng các thanh ghi của timer 0 và  $TH0$  điều khiển bằng các thanh ghi của timer 1. Khi  $TL0$  đếm tràn thì  $TF0$  thiết lập ở mức 1. Còn  $TH0$  khi có tràn thì  $TF1$  được đặt lên mức 1.

### 1.3.5. Tập lệnh VDK8051

Các lệnh của AT89C51 được chia thành 5 nhóm lệnh:

- Nhóm lệnh di chuyển dữ liệu.
- Nhóm lệnh số học.
- Nhóm lệnh logic.
- Nhóm lệnh rẽ nhánh chương trình.
- Nhóm lệnh điều khiển biến logic.

#### *Các quy ước trong câu lệnh và địa chỉ*

+  $Rn$ : Thanh ghi  $R0-R7$  của bảng thanh ghi hiện hành đang được chọn để định địa chỉ thanh ghi.

+ Direct: Địa chỉ 8 bit của ô nhớ dữ liệu nội trú, nó có thể là ô nhớ trong RAM nội hoặc SFR. (00h-FFh)

+ @ $Ri$ : Ô nhớ 8 bit của RAM nội được định địa chỉ gián tiếp thông qua thanh ghi  $R0$  hoặc  $R1$ .

+ Source (Src): toán hạng nguồn, có thể là  $Rn$  hoặc direct hoặc @ $Ri$ .

- + Dest: Toán hạng đích, có thể là Rn hoặc direct hoặc @Ri.
- + Bit: Bit được định địa chỉ trực tiếp trong RAM nội trú hoặc SFR.
- + Rel: Offset 8 bit có dấu (từ -128 đến +127)
- + Addr11: địa chỉ 11 bit của bộ nhớ chương.
- + Addr16: địa chỉ 16 bit của 64Kb bộ nhớ chương trình.

- ***Nhóm lệnh di chuyển dữ liệu***

Lệnh MOV dạng Byte:

**MOV <dest-byte>, <src-byte>**

Lệnh MOV dạng Bit:

**MOV <dest-bit>, <scr-bit>**

Lệnh MOV dạng Word:

**MOV DPTR, #data16**

Lệnh chuyển byte mã lệnh:

**MOVC A, @A + <thanh ghi cơ sở>**

Lệnh chuyển dữ liệu ra ngoài:

**MOVX <dest-byte>, <src-byte>**

Lệnh chuyển số liệu vào ngăn xếp:

**PUSH direct**

Lệnh chuyển số liệu vào ngăn xếp:

**PUSH direct**

Hoán chuyển dữ liệu:

**XCH A, <byte>**

Hoán chuyển 4 bit thấp:

**XCHD A, @Ri**

- ***Nhóm lệnh tính toán số học***

Lệnh cộng có nhớ.

**ADDC A, <scr-byte>**

Lệnh trừ có mượn.

**SUBB A, <scr-byte>**

Lệnh tăng lên 1 đơn vị.

**INC <byte>**

Lệnh giảm 1 đơn vị.

**DEC <byte>**

Lệnh tăng con trỏ dữ liệu.

**INC DPTR**

Lệnh thực hiện phép nhân.

**MUL AB**

Lệnh thực hiện phép chia.

**DIV AB**

Hiệu chỉnh số thập phân.

**DA A**

- **Nhóm lệnh tính toán logic**

Lệnh AND cho các biến 1 byte.

**ANL <dest-byte>, <src-byte>**

Lệnh AND cho các biến 1 bit

**C, <src-bit>**

Lệnh OR cho các biến 1 byte

**ORL <dest-byte>, <src-byte>**

Lệnh X-OR cho các biến 1 byte

**XRL <dest-byte>, <src-byte>**

Lệnh dịch trái thành ghi A

**RL A**

Lệnh dịch trái thành ghi A cùng với cờ nhớ

**RLC A**

Lệnh dịch phải thành ghi A.

**RR A**

Lệnh dịch phải thành ghi A cùng với cờ nhớ

**RRC A**

Lệnh trao đổi nội dung hai nửa byte của A

**SWAP A**

- **Nhóm lệnh rẽ nhánh chương trình**

Lệnh gọi tuyệt đối.

**ACALL addr11**

Lệnh gọi dài.

**LCALL addr16**

Lệnh quay trở lại từ chương trình con.

**RET**

Lệnh quay trở lại từ ngắt.

**RETI**

Lệnh nhảy gián tiếp.

**JMP @A+DPTR**

Lệnh nhảy nếu 1 bit được thiết lập.

**JB bit, rel**  
 Lệnh nhảy nếu 1 bit không được thiết lập.

**JNB bit, rel**  
 Lệnh nhảy nếu 1 bit được thiết lập và xoá bit đó.

**JBC bit, rel**  
 Lệnh nhảy nếu cờ nhớ được thiết lập.

**JC rel**  
 Lệnh nhảy nếu cờ nhớ không được thiết lập.

**JNC rel**  
 Lệnh nhảy nếu thanh ghi A bằng 0.

**JZ rel**  
 Lệnh nhảy nếu thanh ghi A khác 0.

**JNZ rel**  
 Lệnh nhảy khi so sánh 2 toán hạng.

**CJNE <dest-byte>, <src-byte>, rel**  
 Lệnh giảm và nhảy.

**DJNZ <byte>, <rel-address>**  
 Lệnh tạm ngừng hoạt động.

**NOP**

- **Nhóm lệnh điều khiển biến logic**

Lệnh xoá bit

**CLR bit**

Lệnh xoá thanh ghi tích lũy

**CLR A**

Lệnh thiết lập bit

**SETB bit**

Lệnh lấy bù của bit

**CPL <bit>**

Lệnh lấy bù của thanh ghi tích lũy

**CPL A**

## **1.4. TỔNG QUÁT VỀ TÍN HIỆU HỒNG NGOẠI**

### **1.4.1. Khái niệm**

Ánh sáng hồng ngoại (infrared)-IR là ánh sáng không thể nhìn thấy được bằng mắt thường, có bước sóng khoảng từ 0.86 $\mu$ m đến 0.98 $\mu$ m. Tia hồng ngoại có vận tốc truyền bằng vận tốc ánh sáng.



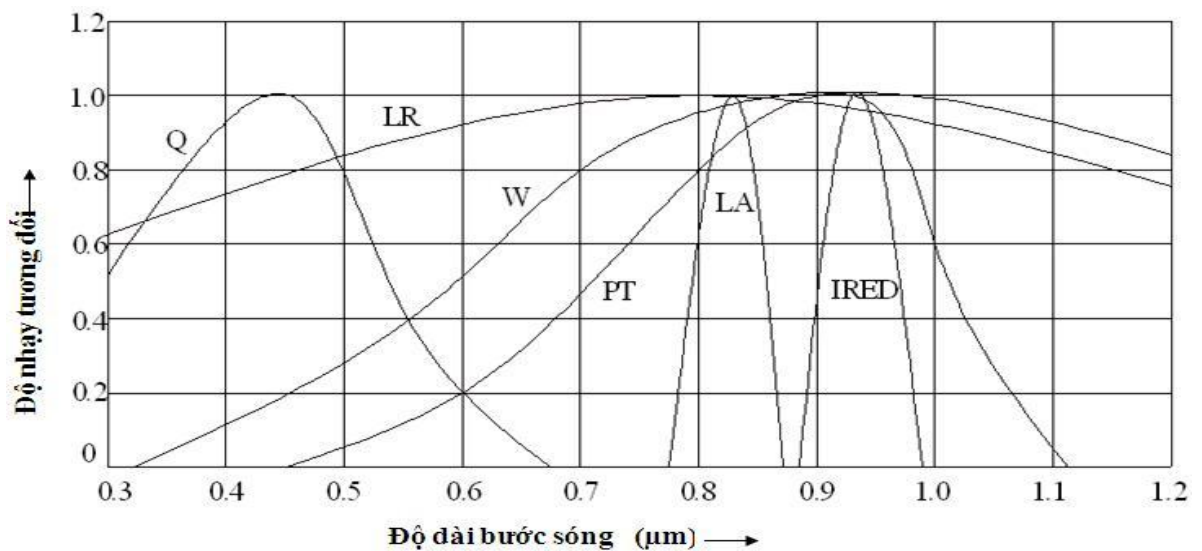
Tia hồng ngoại có thể truyền đi được nhiều kênh tín hiệu. Nó được ứng dụng rộng rãi trong công nghiệp. Lượng thông tin được truyền đi với ánh sáng hồng ngoại lớn gấp nhiều lần so với sóng điện từ có thể đạt 3 Mbit /s.

#### 1.4.2. Nguồn phát tia hồng ngoại

+ Các nguồn dùng phát ra tia hồng ngoại như:

Mặt Trời là một nguồn phát tia hồng ngoại mạnh, cơ thể con người có nhiệt độ bình thường là 37°C nên là một nguồn phát ra tia hồng ngoại với bước sóng khoảng 9  $\mu m$ . Các vật có nhiệt độ lớn hơn 0 độ K đều có phát ra tia hồng ngoại. Đèn dây tóc, bếp gas, lò sưởi là những nguồn phát ra tia hồng ngoại khá mạnh.

+ Bước sóng của các nguồn hồng ngoại.



**Hình 1.11.** Bước sóng của các nguồn hồng ngoại.

IRED: Diode hồng ngoại

LA : Laser bán dẫn

LR : Đèn huỳnh quang

Q : Đèn thủy tinh

W : Bóng đèn điện với sợi wolfram

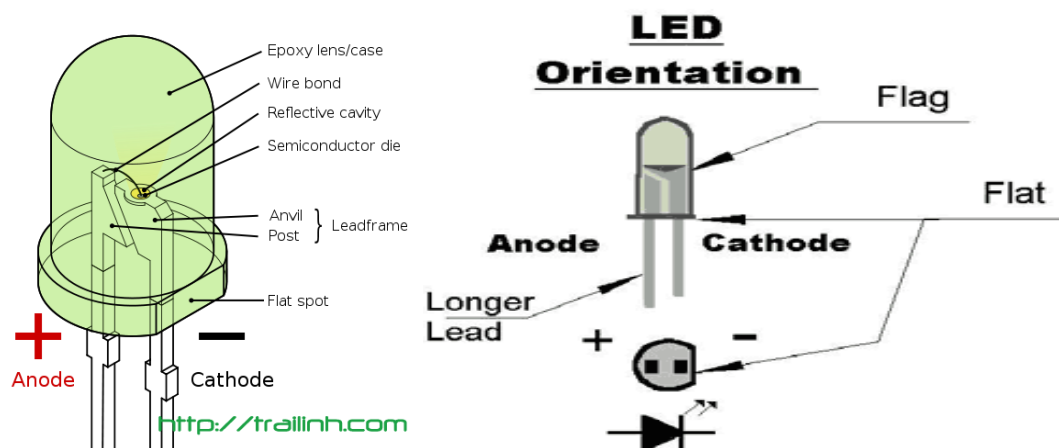
PT : Phototransistor

Sóng hồng ngoại có những đặc tính quan trọng giống như ánh sáng ( sự hội tụ qua thấu kính, tiêu cự..

Ánh sáng và sóng hồng ngoại khác nhau rất rõ trong sự xuyên suốt qua vật chất, có những vật mà ta thấy sự phản chiếu sáng nhưng đối với tia hồng ngoại là những vật phản chiếu tối. Vật liệu bán dẫn trong suốt đối với ánh sáng hồng ngoại, tia hồng ngoại không bị yếu đi khi nó xuyên qua các lớp bán dẫn để ra ngoài.

### 1.4.3. Bộ phát tín hiệu hồng ngoại

Để phát chuỗi mã lệnh điều khiển ra ngoài không gian ta dùng led phát hồng ngoại, về cấu tạo giống như led phát quang bình thường có phân cực anot và katot nhưng được chế tạo từ vật liệu bán dẫn Gallium Arsenide năng lượng phát ra là ánh sáng hồng ngoại không nhìn thấy được và dùng trong các mạch báo động, điều khiển từ xa...



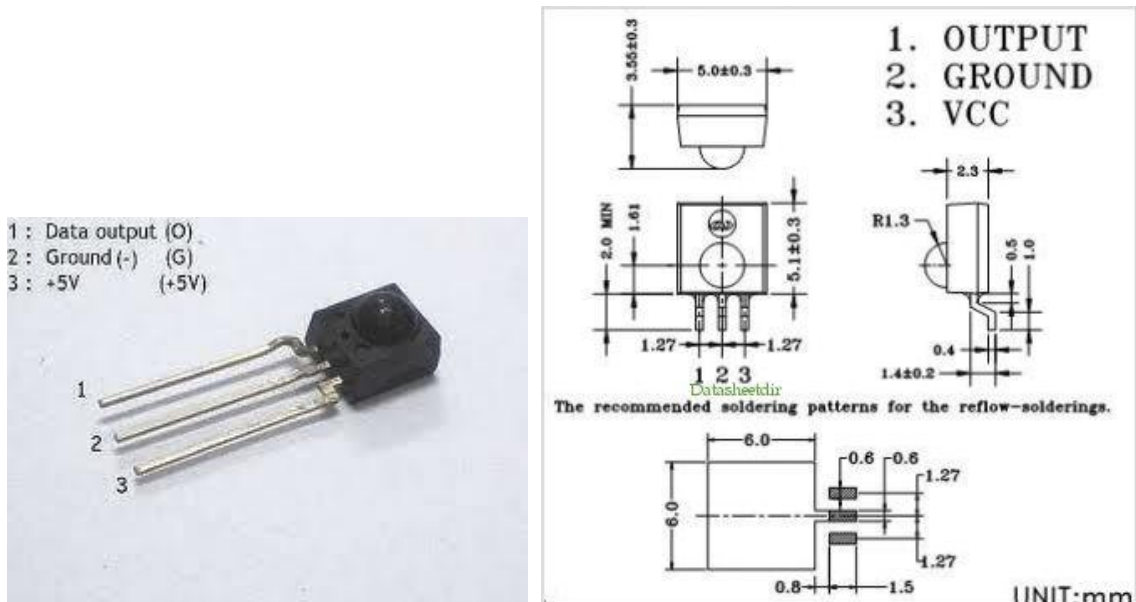
Hình 1.12: Led phát tia IR.

Thông số kỹ thuật:

- Nguồn cấp 3,8V-12V
- Dòng điện làm việc 15mA đến 25mA
- Tần số sóng mang 36-38KHz
- Nhiệt độ làm việc -40<sup>o</sup>c đến + 85<sup>o</sup>c

#### 1.4.4. Bộ thu tín hiệu hồng ngoại

Để thu được tín hiệu hồng ngoại do thiết bị phát phát đi có chứa chuỗi lệnh điều khiển ta dùng một thiết bị thu về đưa qua khuếch đại, so sánh, phân tích thành tín hiệu số rồi đưa vào vi điều khiển. PIC 1018SCL - Là thiết bị thực hiện chức năng đó.



**Hình 1.13:** Mắt thu IR PIC 1018SCL.

Các thông số kỹ thuật:

- Supply Voltage là nguồn cấp 2.5-5V thường chọn 5V
- Current Max: dòng tiêu thụ cực đại  $I_{cc}=1.5\text{mA}$
- B.P.F Center Frequency: tần số dao động  $F_0=37.9\text{KHz}$
- High Lever Output Voltage : điện áp ra mức cao  $V_{OH}=V_{CC}-0.5\text{V}$
- Low Lever Output Voltage : điện áp ra mức thấp  $V_{OL}=0.2\text{V}$
- Temperature Storate : là nhiệt độ hoạt động từ  $-10^\circ\text{C}$  đến  $+60^\circ\text{C}$ .

### 1.5. TỔNG QUÁT BỘ ĐIỀU KHIỂN REMOTE PHÁT TIA IR

#### 1.5.1. Giới thiệu

Bộ điều khiển Remote là một thiết bị dùng để phát ra các chuỗi mã lệnh khác nhau. Mỗi nhà sản xuất khác nhau có quy định chuỗi mã lệnh khác nhau. Chuỗi mã lệnh có cấu trúc chung bao gồm: ( bit Start, bit Địa chỉ + bit Dữ liệu + bit Stop ).

Chuỗi mã được phát ra qua bộ phát hồng ngoại khi ta nhấn một nút trên Remote và bên thiết bị thu mã lệch sẽ nhận được tín hiệu đưa đến bộ xử lý nhằm mục đích thực hiện các chức năng điều khiển. Sau đây là các loại Remote của các hãng sản xuất: Sharp, Sony, Samsung, Hitachi, Toshiba v.v.



*Hình 1.14:* Các bộ điều khiển TV các hãng.

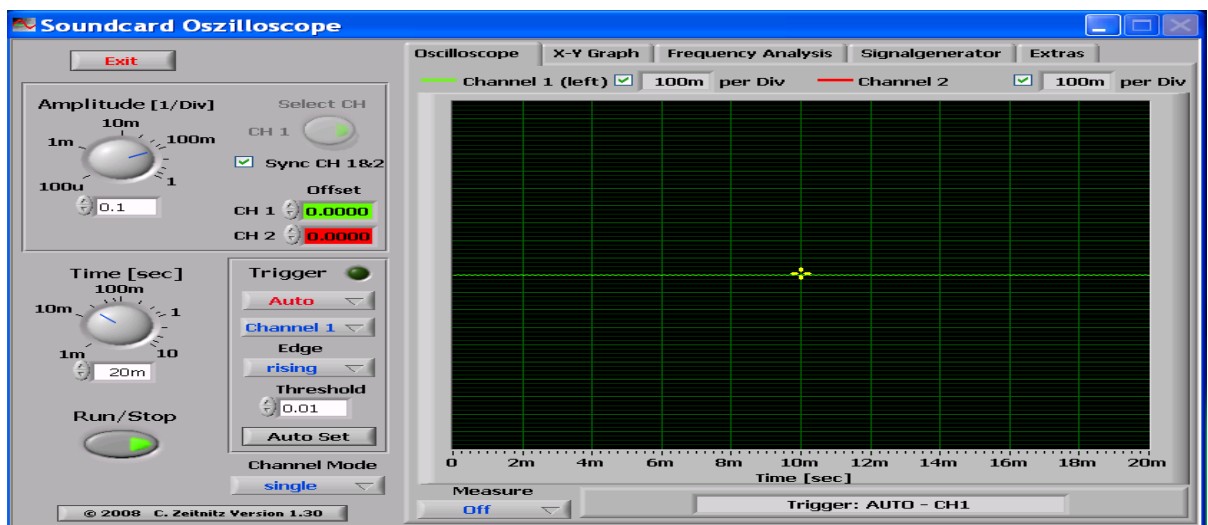
### **1.5.2. Giải mã tín hiệu điều khiển TV SHARP**

Remote TV Sharp là một bộ điều khiển tivi Sharp của nhật bản bao gồm có 25 phím chức năng dùng để chọn các kênh, cài đặt âm lượng, cài đặt màu sắc v.v.



**Hình 1.15:** Bộ điều khiển TV Sharp TC-987A.

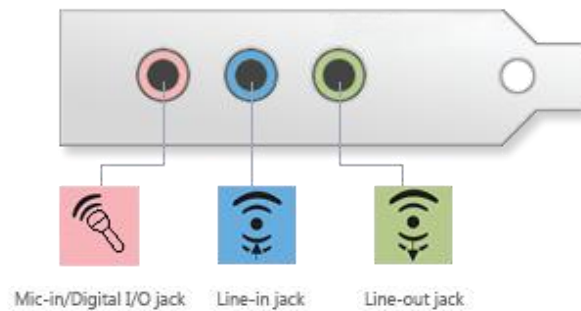
Việc giải mã tín hiệu mã bit ta dùng thiết bị Oscilloscope. Ta cũng có thể dùng một phần mềm **Scope 1.30** hỗ trợ cài đặt chạy trên môi trường window để tiến hành giải mã. Sau khi cài đặt thành công phần mềm có giao diện như sau:



**Hình 1.16:** Giao diện phần mềm Scope 1.30.

Phần mềm có chức năng tương tự như một Oscilloscope, có 2 kênh đo tín hiệu CH1 và CH2, có thể chỉnh độ cao của xung tín hiệu trong (Amplitude[1/div] và độ rộng trong (Time[sec])). Để đưa tín hiệu vào cho phần mềm phân tích được thì phải kết nối giữa bộ remote điều khiển và

máy tính thông qua cổng **Line IN** trên main board máy tính bằng dây kết nối Audio.



**Hình 1.17:** Các cổng on Main máy tính.

- (1) Là cổng **Line Out** (tín hiệu ra loa).
- (2) Là cổng **Line In** (là cổng ta dùng để kết nối).
- (3) Là cổng **Microphone**.

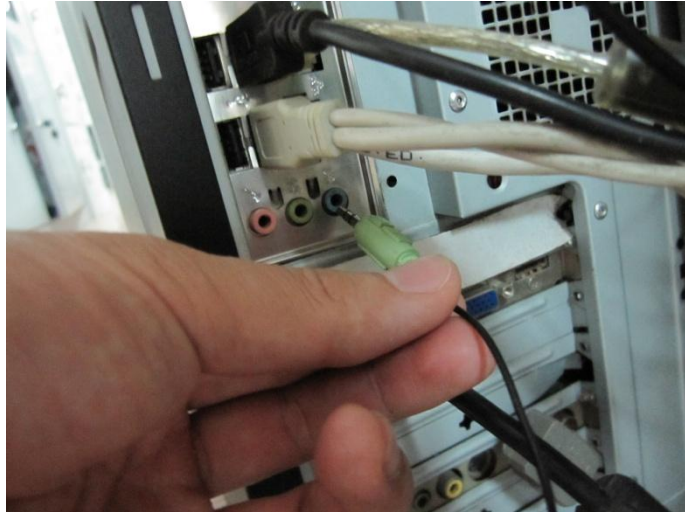
Dây kết nối ta dùng loại dây Audio:



**Hình 1.18:** Dây Audio.

Một đầu của dây audio được cắm vào cổng **Line in**. Đầu dây kia tách ra làm 2 nhánh, sau đó hàn vào nơi phát tín hiệu hồng ngoại, chính là mắc vào 2 chân ( -, + ) của led phát hồng ngoại trong bộ Remote Sharp.

Tiếp theo ta thực hiện việc giải mã tín hiệu của từng phím.

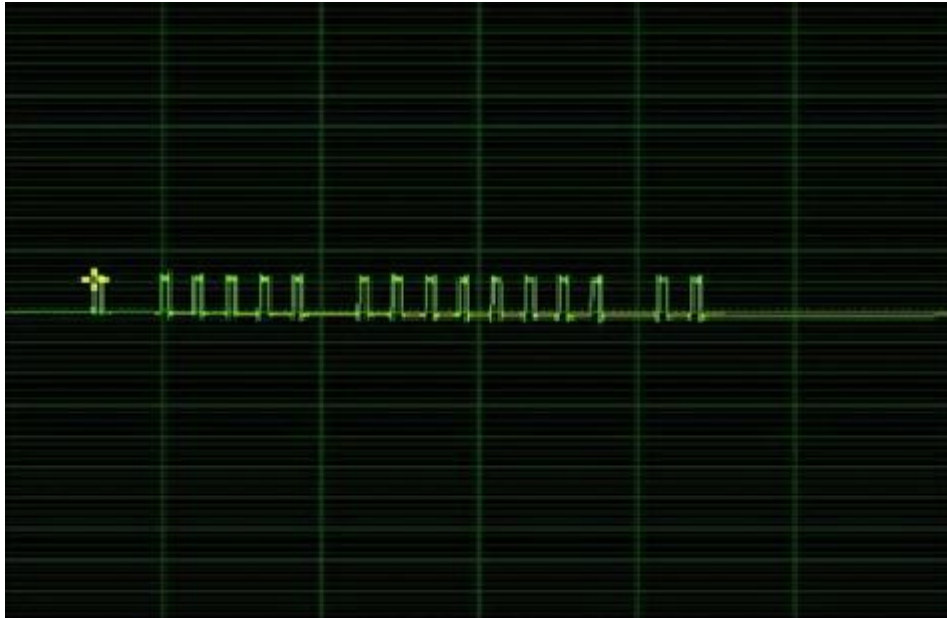


**Hình 1.19:** Thao tác kết nối.

Khởi động chương trình **Scope 1.30.**, chỉnh các thông số (Amplitude[1/div] =0.5 , (Time[sec])=30ms.

Khi nhấn các phím của Remote TV thì trên giao diện **Scope 1.30** có xuất hiện các chuỗi bit tín hiệu như sau:

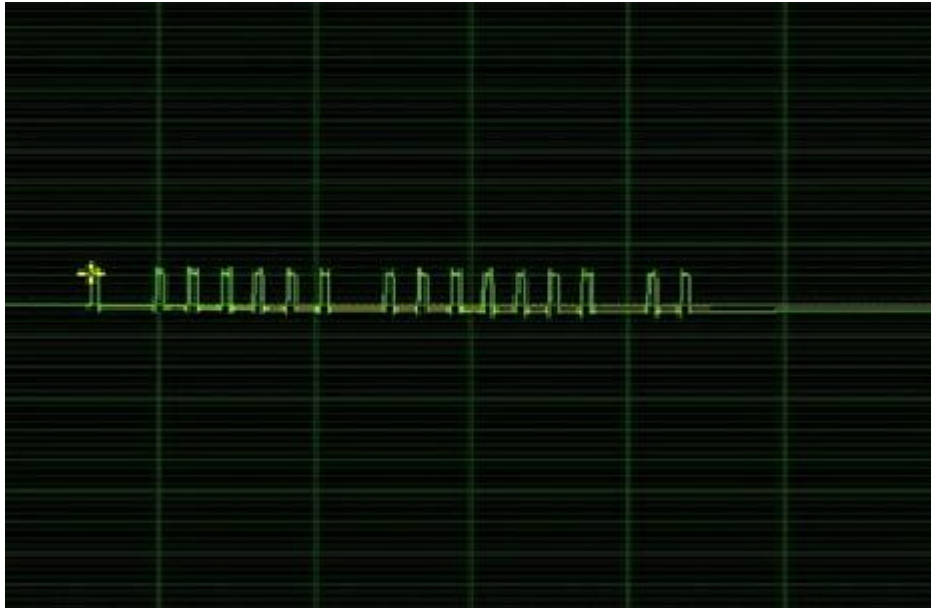
Phím 1:



**Hình 1.20:** Tín hiệu điều khiển phím 1.

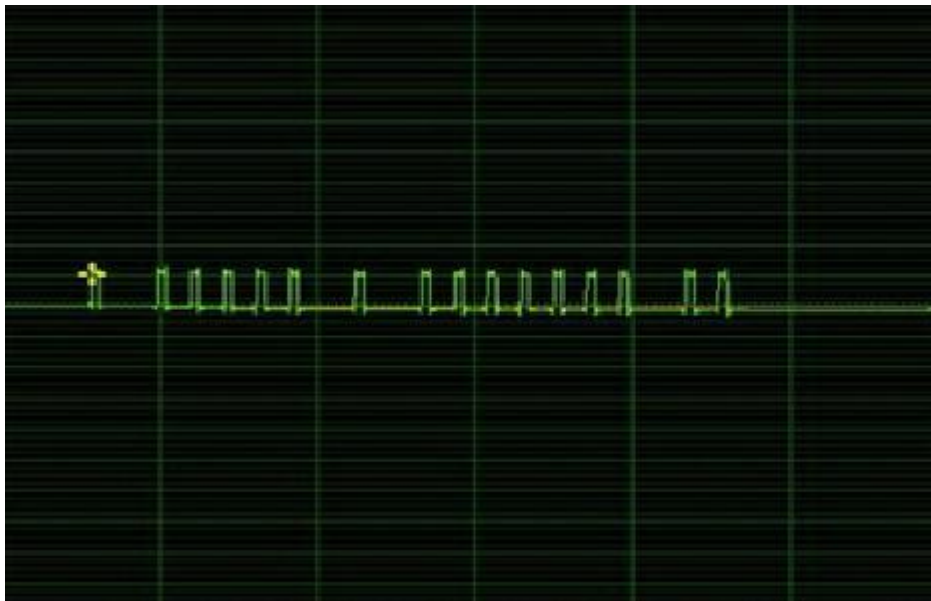


Phím 2:



*Hình 1.21:* Tín hiệu điều khiển phím 2.

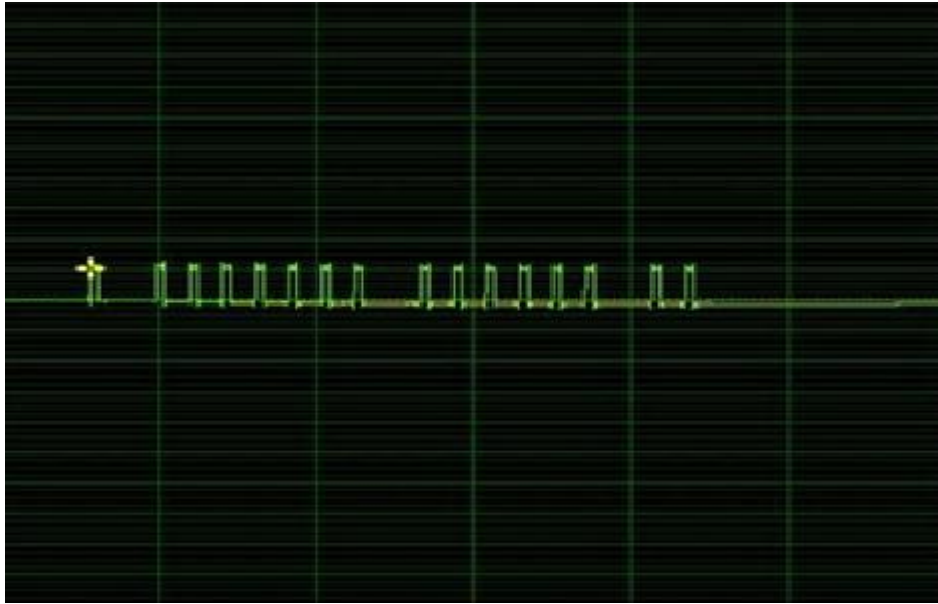
Phím 3:



*Hình 1.22:* Tín hiệu điều khiển phím 3.

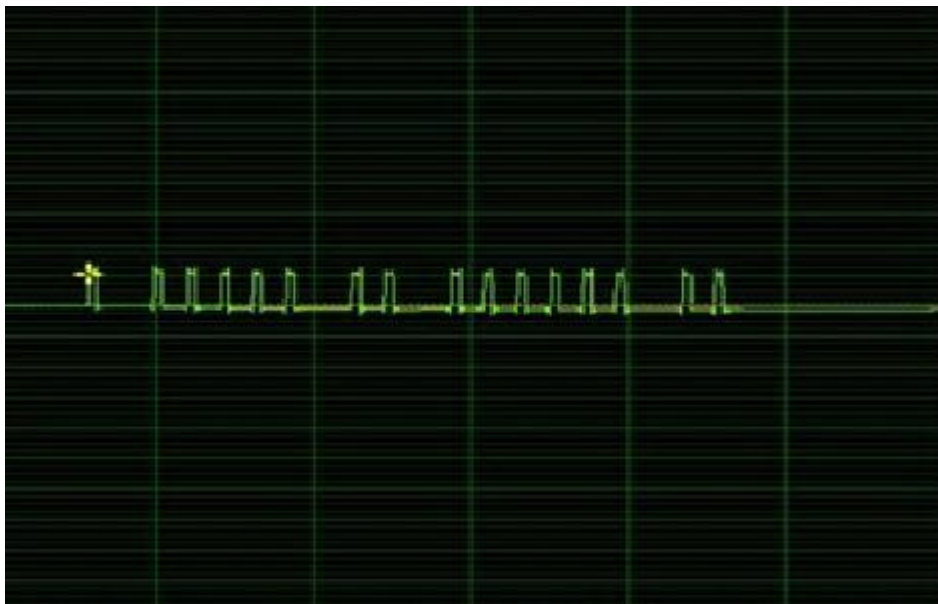


Phím 4:



*Hình 1.23:* Tín hiệu điều khiển phím 4.

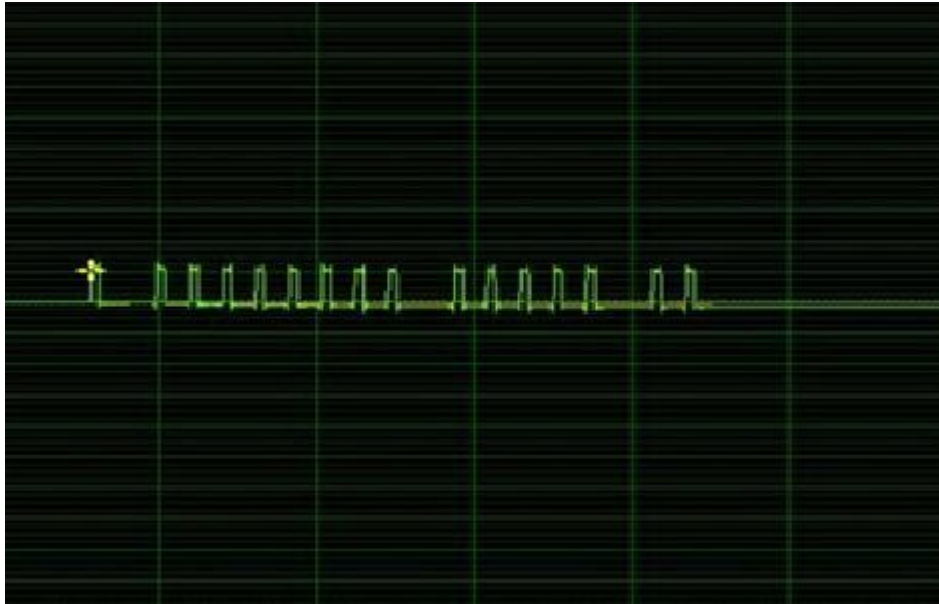
Phím 5:



*Hình 1.24:* Tín hiệu điều khiển phím 5.

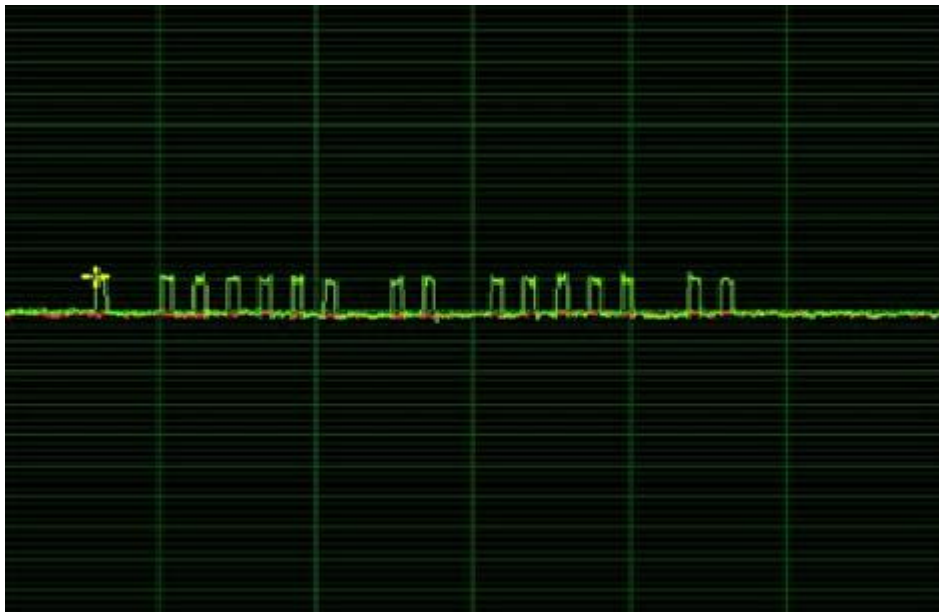


Phím 8:



*Hình 1.27:* Tín hiệu điều khiển phím 8.

Phím 10:



*Hình 1.28:* Tín hiệu điều khiển phím 10.

Phím 11:



**Hình 1.29:** Tín hiệu điều khiển phím 11.

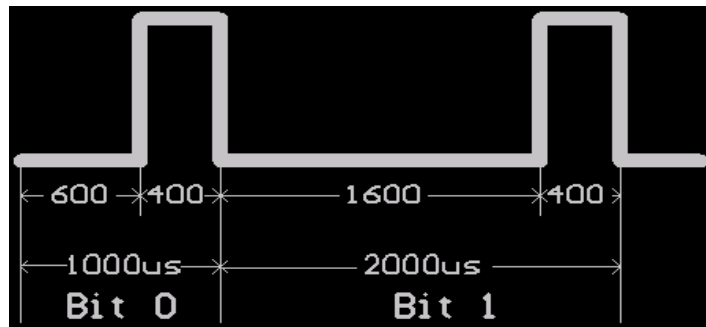
Một khung truyền tín hiệu Remote Sharp gồm 5 bit đầu tiên là bit Start và bit địa chỉ, 8 bit tiếp theo là dữ liệu và 2 bit cuối là Stop. Bit địa chỉ của tất cả các phím ấn là giống nhau còn 8 bit dữ liệu của mỗi phím là hoàn toàn khác nhau. Vấn đề là phải tách được chuỗi 8 bit dữ liệu này đưa vào cho vi điều khiển phân tích và đưa ra lệnh điều khiển. Chuỗi mã lệnh được truyền nhờ sóng mang hồng ngoại với tần số 36 KHz thể hiện trong hình sau:



**Hình 1.30:** Chuỗi mã bit của một phím ấn.

Thời gian để truyền hết một chuỗi bit khi ta nhấn phím là 20ms. Bit 0 và bit 1 có chu kì giống nhau nhưng có độ rộng bit là khác nhau, ở bit 0

truyền mất thời gian 1000 micro giây còn bit 1 là 2000 micro giây. Chi tiết như sau:



**Hình 1.31:** Thời gian truyền của bit 0 và bit 1.

Bộ điều khiển Remote TV Sharp được giải mã cụ thể như sau:

- 1: 100001000000010 = 80h
- 2: 100000100000010 = 40h
- 3: 100001100000010 = 0c0h
- 4: 100000010000010 = 20h
- 5: 100001010000010 = 0a0h
- 6: 100000110000010 = 60h
- 7: 100001110000010 = 0e0h
- 8: 100000001000010 = 10h
- 10: 100000101000010 = 50h
- 11 : 100001101000010 = 0D0h

## CHƯƠNG 2.

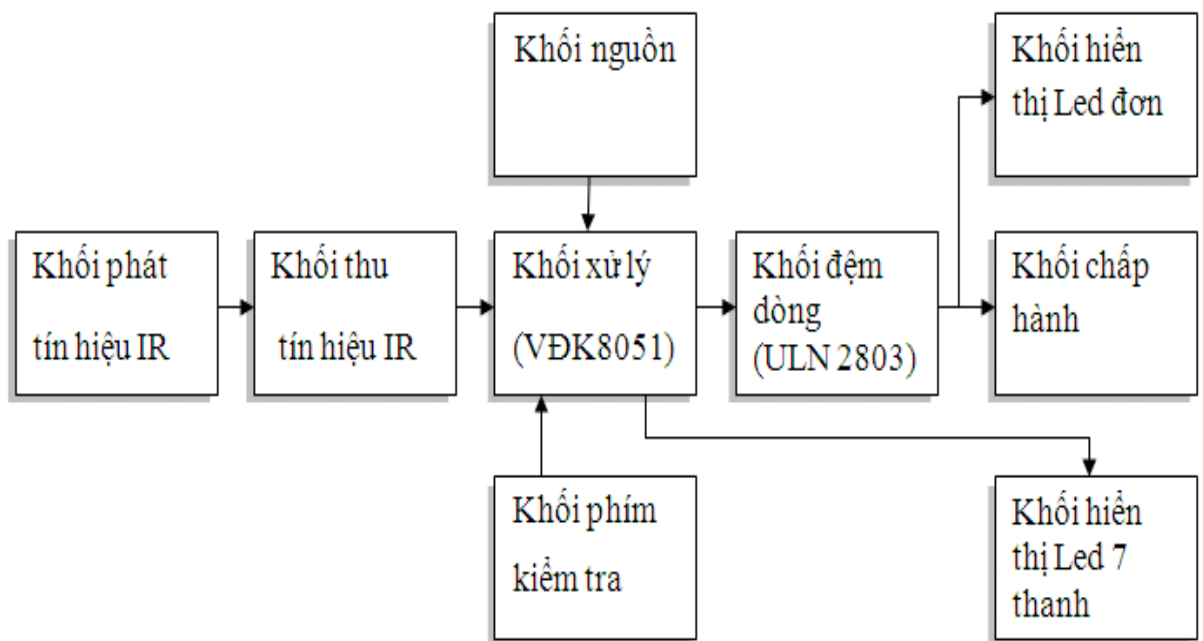
### XÂY DỰNG MÔ HÌNH PHẦN CỨNG HỆ THỐNG

#### 2.1. MỞ ĐẦU

Trong mô hình điều khiển các thiết bị điện từ xa sau đây là một mô hình nhỏ gọn, điều khiển minh họa 4 bóng đèn. Nhưng thực tế ta có thể mở rộng điều khiển lên đến 31 thiết bị dân dụng. Mô hình được xây dựng lên có tính ứng dụng cao vào thực tế. Thiết bị dễ tìm, linh kiện rẻ, độ chính xác cao và an toàn.

#### 2.2. SƠ ĐỒ KHỐI HỆ THỐNG

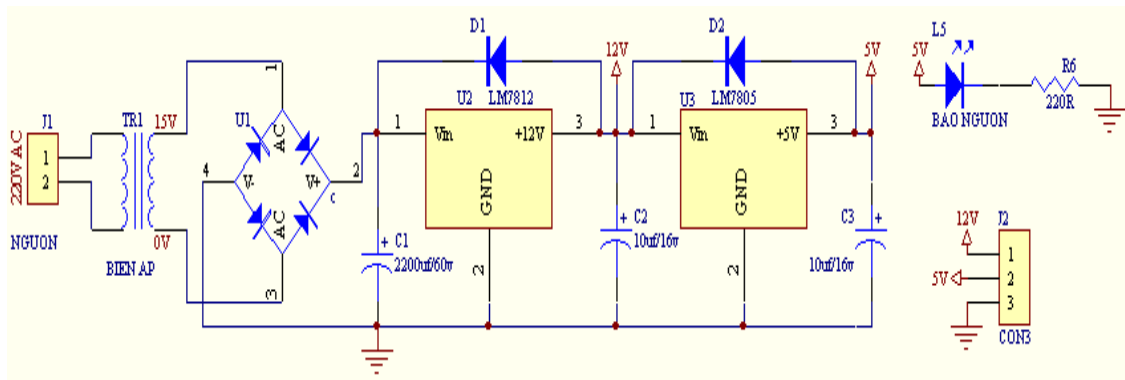
Sơ đồ khối của hệ thống điều khiển được mô tả tổng quát qua các khối chức năng sau:



*Hình 2.1:* Sơ đồ khối hệ thống.

## 2.3. CHỨC NĂNG VÀ HOẠT ĐỘNG CÁC THÀNH PHẦN

### 2.3.1. Khối nguồn

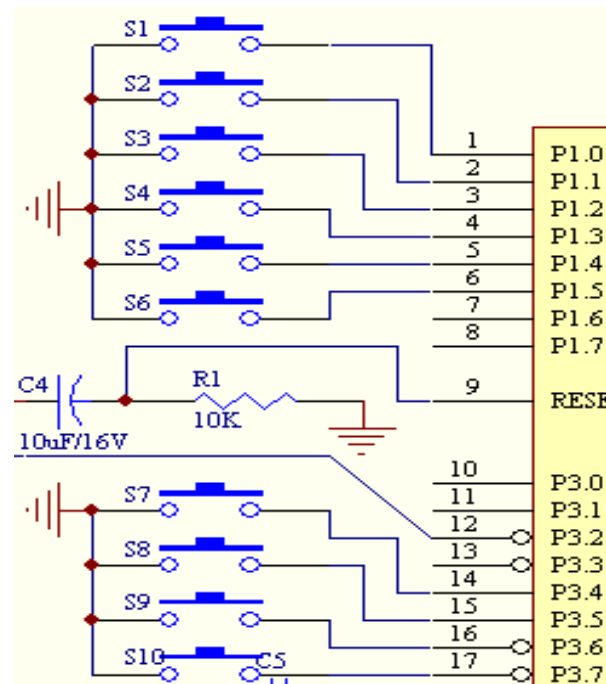


**Hình 2.2:** Khối nguồn.

+ Chức năng: Chuyển đổi điện 220VAC sang 12VDC, 5VDC cấp cho mạch.

+ Hoạt động: Nguồn điện 220VAC vào biến áp 3A biến đổi thành các mức điện áp xoay chiều nhỏ: 24V, 18V, 15V, 12V, 9V, 6V, 0V. Vì mạch cần mức điện áp 12VDC và 5VDC nên lấy mức 15V và 0V sẽ đưa vào bộ chỉnh lưu cầu nắn thành điện áp 1 chiều. Ta dùng IC LM7812 và LM7805 để lấy ra mức điện áp chuẩn 12VDC và 5VDC. Tụ C=2200uf/50v ban đầu lớn tác dụng làm cho điện áp 1 chiều chuẩn hơn. Các tụ 10uf tác dụng lọc lại điện áp ngõ ra tại chân 3 của IC LM7812 và LM7805.

### 2.3.2. Khối keyboard kiểm tra



**Hình 2.3:** Khối keyboard.

+ Chức năng: Kiểm tra phần cứng board mạch trước khi sử dụng Remote.

+ Hoạt động: Viết một chương trình kiểm tra mạch, nếu phím nào được nối GND thì sẽ thực hiện bật tắt các bit cổng P0. Từng bit của P0 được hiển thị qua led đơn và Role

### 2.3.3. Khối phát tín hiệu IR

+ Chức năng: Điều khiển các thiết bị điện.

+ Hoạt động: Mỗi bộ điều khiển TV Remote được mã hóa có cấu trúc mã lệnh khác nhau tùy thuộc vào nhà sản xuất

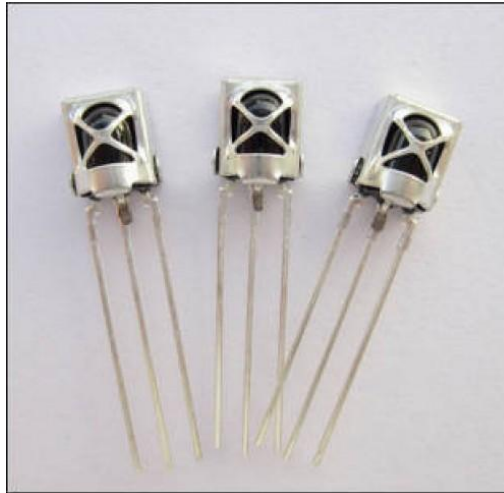


**Hình 2.4:** Khối phát tín hiệu IR.



Khi nhấn một phím trên bộ Remote thì một chuỗi mã lệnh được Led phát IR phát đi với tần số 36KHz đến các thiết bị thu.

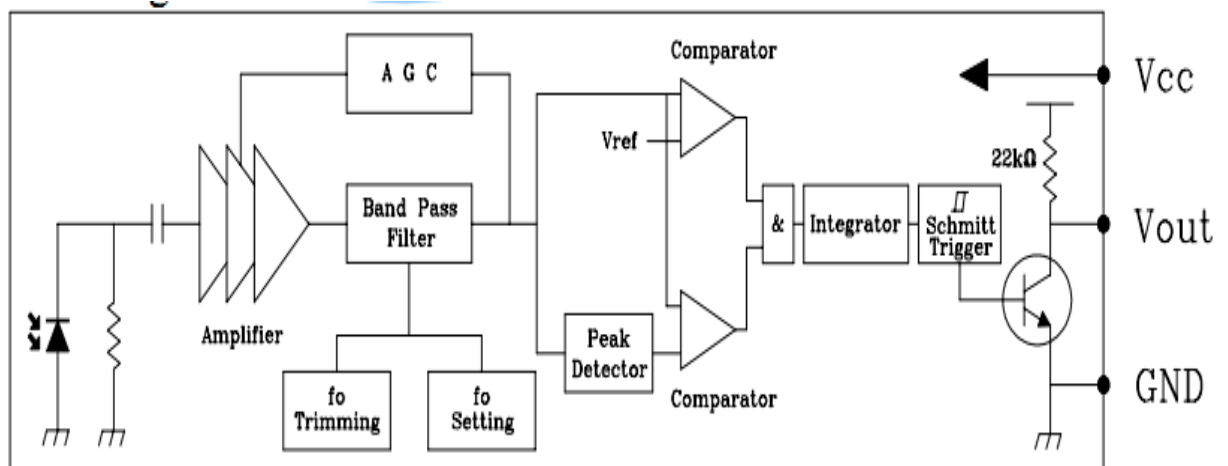
#### 2.3.4. Khối thu tín hiệu IR



**Hình 2.5:** Khối thu tín hiệu IR.

+ Chức năng: Thu nhận tín hiệu IR đưa về cho VDK 8051 xử lý.

+ Hoạt động:

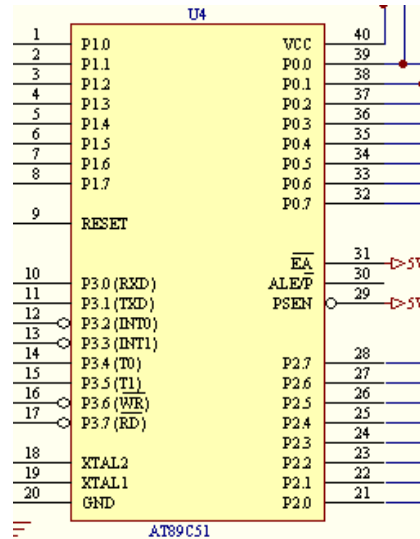


**Hình 2.6:** Cấu trúc bên trong PIC 1018SCL

Tín hiệu IR từ nguồn phát đến mạch thu được led hồng ngoại nhận rồi đưa qua ba tầng khuếch đại (Amplifier). Sau đó tín hiệu này được qua mạch lọc băng thông (Band Pass Filter) để chọn dải băng thông thích hợp. Ở ngõ ra tín hiệu này được ngược lại mạch khuếch đại (AGC) để tăng độ khuếch đại nếu cần thiết. Tiếp tục tín hiệu đưa đến bộ so sánh và phân tích

trước khi vào mạch Schmitt Trigger. Nhiệm vụ chủ yếu của mạch Schmitt Trigger là đổi tín hiệu liên tục thành tín hiệu xung vuông đưa vào VĐK.

### 2.3.5. Khối xử lý tín hiệu

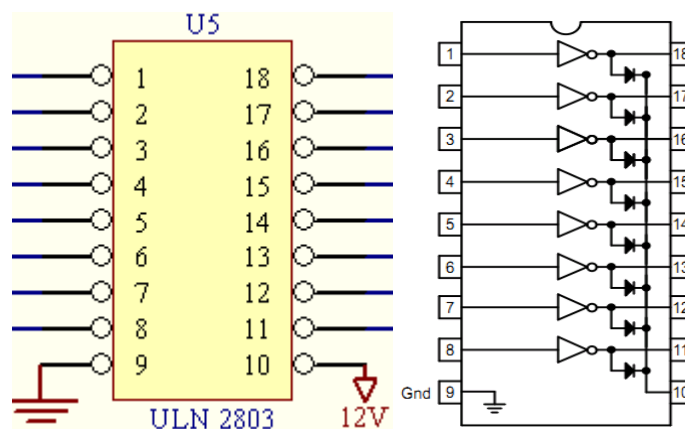


**Hình 2.7:** Khối xử lý tín hiệu.

+ Chức năng: Xử lý tín hiệu IR.

+ Hoạt động: VĐK sẽ nhận mã lệnh do Led thu IR PIC 1018SCL đưa về và so sánh với bộ mã lệnh mà người viết chương trình cung cấp. Khi mã phím Remote nào giống với mã của VĐK thì thực hiện công việc tương ứng với phím đó.

### 2.3.6. Khối đệm dòng



**Hình 2.8:** Khối đệm dòng.

ULN 2803 là một vi mạch đệm, có chứa 8 bộ đệm. Mỗi bộ đệm có một diode với đầu anod được nối với ngõ ra của ic còn catod được nối chung với catot của các diode còn lại. Ngõ ra của vi mạch là các cực góp hở. Tín hiệu ngõ vào sẽ bị đảo chiều so với tín hiệu ngõ ra. Nguồn cấp <math><50V</math>, dòng làm việc <math><500\text{ mA}</math>.

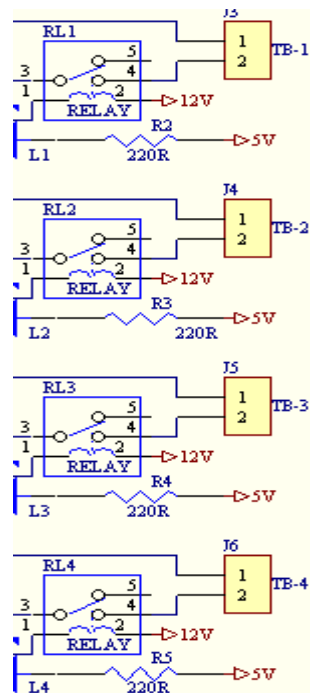
+ Chức năng: Đệm dòng.

+ Hoạt động:

- Nếu các chân đầu vào I1 ÷ I8 là mức 0 thì ngõ ra không tác dụng.

- Nếu các chân đầu vào I1 ÷ I8 là mức 1 thì ngõ ra ở mức 0.

### 2.3.7. Khối chấp hành



**Hình 2.9:** Khối chấp hành.

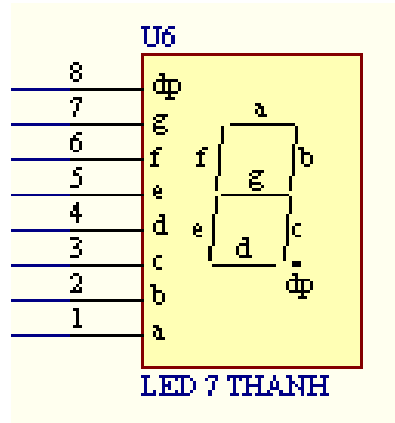
+ Chức năng: Đóng ngắt thiết bị điện

+ Hoạt động: Sự làm việc của role dựa trên nguyên lí điện từ. Khi cho dòng điện đi vào cuộn dây của nam châm điện thì nắp sẽ chịu một lực hút.

Khi dòng điện vào cuộn dây  $i > I_{td}$  (dòng điện tác động) làm cho lực F tăng, hút nắp dẫn đến khe hở giảm làm đóng tiếp điểm.

Khi dòng điện  $i < I_{tv}$  (dòng trở về) thì lực lò xo F lò xo giảm xuống và rơi le nhả các tiếp điểm.

### 2.3.8. Khối hiển thị



**Hình 2.10:** Khối hiển thị led đơn và led 7 thanh.

+ Chức năng: Hiển thị tín hiệu tại cổng P0, P2.

+ Hoạt động: Anot của Led đơn và chân Anot chung của led 7 thanh được nối +5V. Khi có tín hiệu âm sẽ làm cho led đơn sáng lên báo sự hoạt động của Rowle, led 7 thanh sẽ hiển thị chữ số 1, 2, 3, 4 báo là các thiết bị đang hoạt động.

## 2.4. LIỆT KÊ, TÍNH TOÁN CÁC LINH KIỆN TRÊN MẠCH

### 2.4.1. Liệt kê linh kiện

- IC: AT89C51, LM7812, LM7805, ULN2803, PIC 1018SCL, Thạch anh 11,0592 MHz.

- TỤ : 2000uf /16v, 1000uf /16v, 10uf/16v, 33p

- Role 12VDC

- Nút nhấn

- Trở thanh 10k, led đơn, led 7 thanh, điện trở 220Ω, Diode 1N4007

### 2.4.2. Tính toán lựa chọn các linh kiện

VĐK 8051 có thể được thay thế bằng 8052 để tăng bộ nhớ ROM.

Chọn Led thu hồng ngoại: Ngoài PIC 1018SCL ra ta có thể thay thế bằng TSOP1838 ( 1 VCC, 2 GND, 3 OUT ), TSOP1738 ( )

Chọn led đơn và giá trị điện trở:

Thông số kỹ thuật:  $V_{led} = 1.9 - 2.2V$ ,  $I_{led} = 15 - 25mA$ , nguồn cấp 5VDC

Chọn điện áp trung bình qua led là  $2V$ , dòng trung bình  $I = 15mA = 0.015A$ .

Công thức tính điện trở:

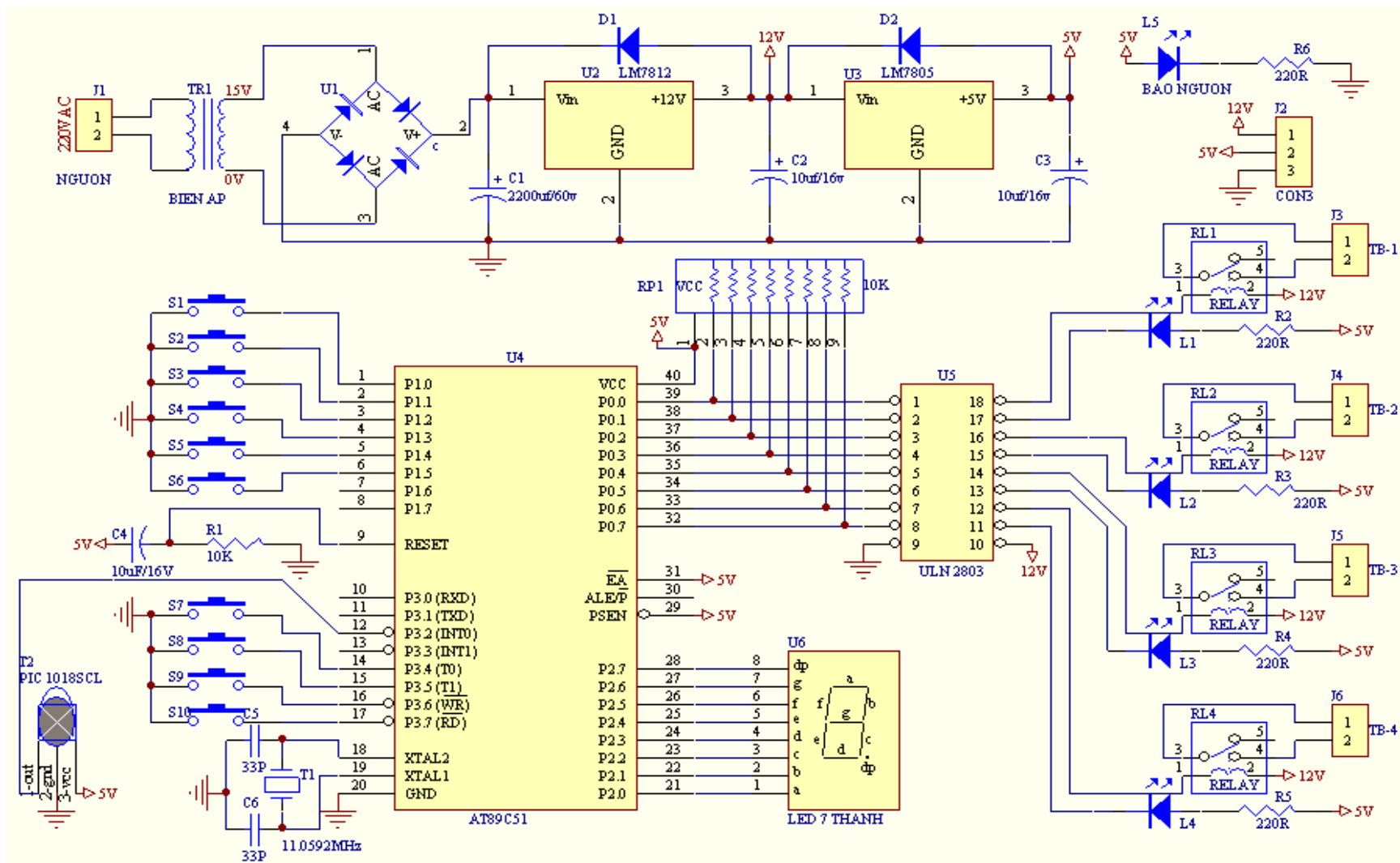
$$R = \frac{U_{nguồn} - U_{(led)}}{I_{(led)}} = \frac{5V - 2V}{0.015A} = 200\Omega$$

Chọn  $R = 220\Omega$

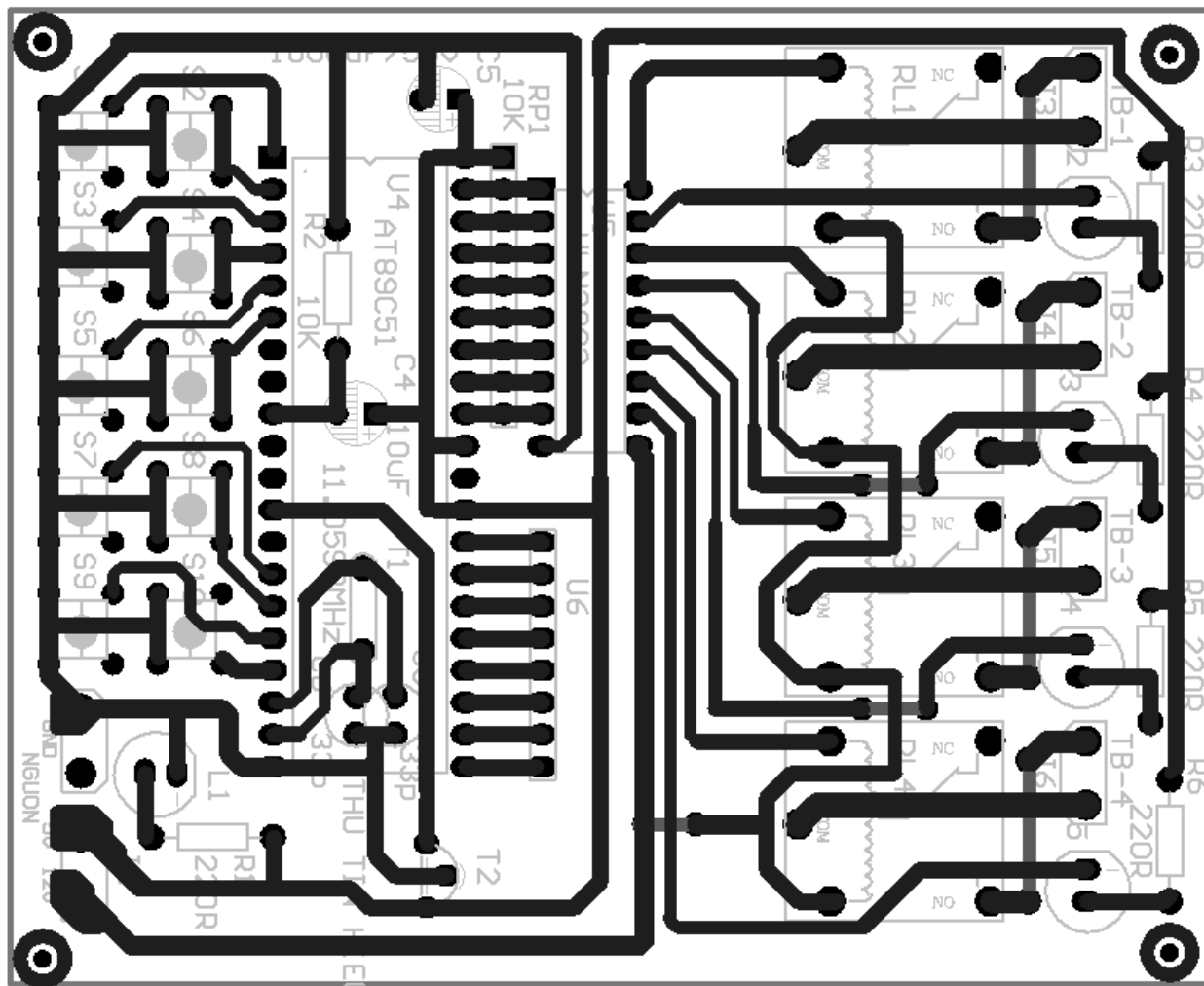
## 2.5. SƠ ĐỒ NGUYÊN LÝ VÀ MẠCH IN

Sơ đồ nguyên lý hệ thống bao gồm khối nguồn và main được thiết kế chi tiết trên phần mềm **Protel 99SE**:

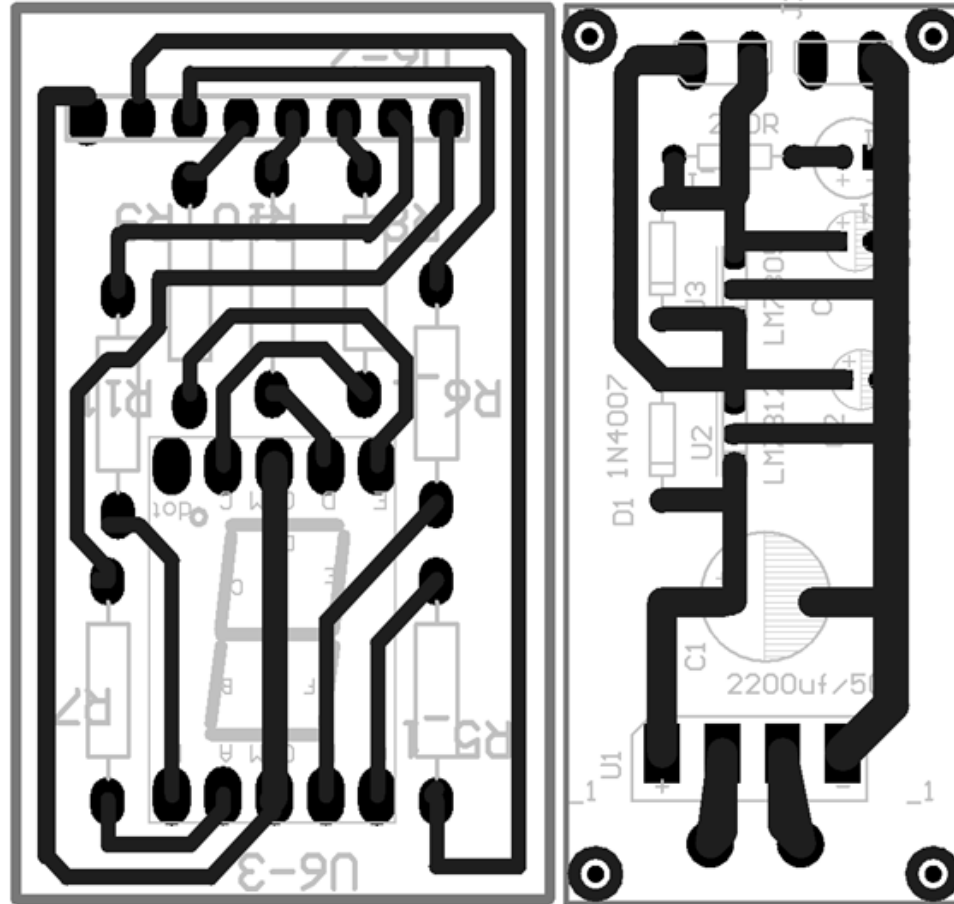
Sơ đồ mạch in được in chi tiết ở chế độ Gray Scale có hiển thị các đường *TopLayer*, *bottomLayer*, *TopOverLay*, *KeepOutLayer*



Hình 2.11: Sơ đồ nguyên lý hệ thống điều khiển từ xa.



*Hình 2.12:* Sơ đồ mạch in bo mạch chính .



**Hình 2.13:** Sơ đồ mạch in nguồn và mạch hiển thị LED 7 đoạn.

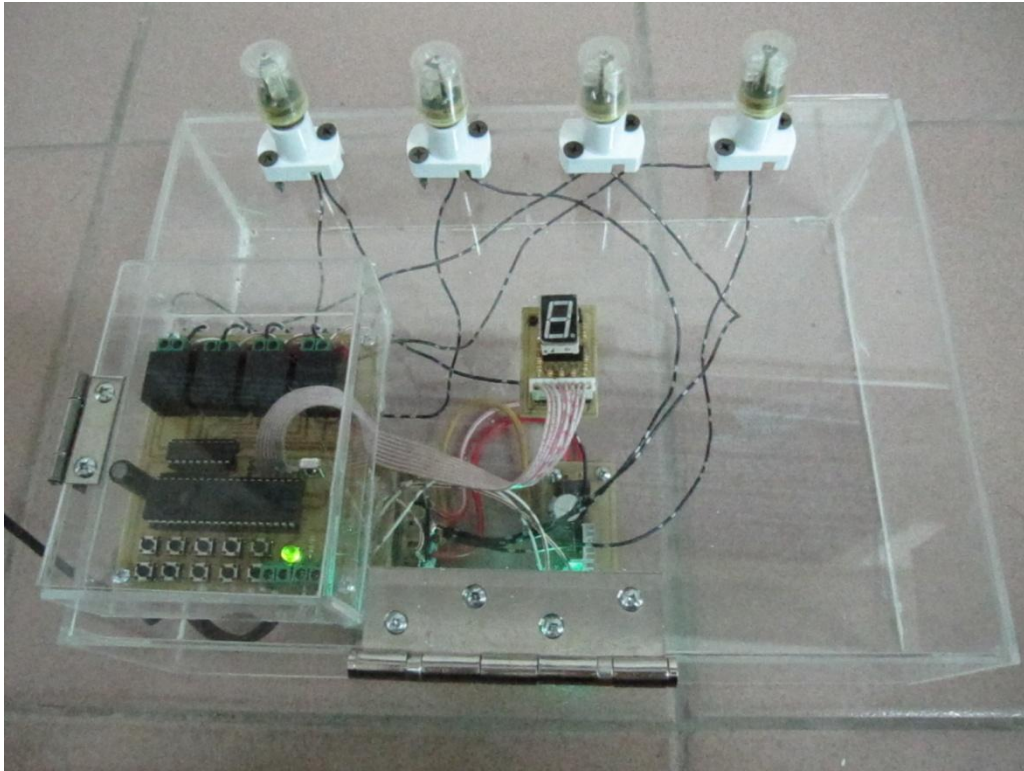


## 2.6. NGUYÊN LÝ HOẠT ĐỘNG HỆ THỐNG

PIC 1018SCL - Mắt thu tín hiệu IR sẽ nhận tín hiệu từ bộ điều khiển Remote TV. Tín hiệu được đưa qua ba tầng khuếch đại (Amplifier), qua mạch lọc băng thông (Band Pass Filter), đến bộ so sánh và phân tích vào Schmitt Trigger là đổi tín hiệu liên tục thành tín hiệu xung vuông đưa vào VĐK. Tín hiệu từ chân OUT (1) được đưa đến chân P3.2 của VĐK. 8051 sẽ thu nhận và chuyển vào thanh ghi A. Khi đã nhận đủ là 8 bit dữ liệu sẽ thực hiện so sánh với mã bit mà người viết đã cho, chuỗi mã nào nhận về mà giống với mã cho trước sẽ có tín hiệu điều khiển các bit cổng P0.

Chân p0.0, p0.2, p0.4, p0.6 điều khiển 4 rơle 12v đóng ngắt 4 thiết bị. Chân p0.1, p0.3, p0.5, p0.7 điều khiển hiển thị 4 đèn led. Khi các chân ở mức thấp (0V) qua IC đệm dòng tín hiệu là cao (+5V) đi vào cuộn hút Rơle và led hiển thị -> Rơle không hút, led tắt. Còn khi các chân ở mức cao (+5V) qua IC đệm dòng tín hiệu là mức thấp (0V) vào cuộn hút Rơle và led hiển thị -> Rơle hút, led sáng.

## 2.7. ẢNH CHỤP MÔ HÌNH THỰC



*Hình 2.13:* Mô hình thực hệ thống điều khiển.

### **NHẬN XÉT**

Mô hình phần cứng của hệ thống là tổng hợp của các hệ thống nhỏ, mỗi thành phần có chức năng riêng có mối liên kết logic chặt chẽ với độ chính xác cao. Tín hiệu ngõ ra của khối chức năng này là tín hiệu ngõ vào của khối kia.

Việc thiết kế trên phần mềm Protel 99SE phải đảm bảo khả năng chống nhiễu tốt, độ thẩm mỹ và an toàn. Sau khi mô hình đã hoàn thành ta thực hiện viết chương trình điều khiển cho VĐK.

## **CHƯƠNG 3.**

# **XÂY DỰNG CHƯƠNG TRÌNH PHẦN MỀM ĐIỀU KHIỂN**

### **3.1. MỞ ĐẦU**

Mỗi phần cứng nhất định phải có chương trình phù hợp kèm theo, do đó trước khi viết chương trình đòi hỏi người viết phải nắm bắt được cấu tạo phần cứng và các yêu cầu mà mạch điện cần thực hiện. Chương trình là tập hợp các lệnh được tổ chức theo một trình tự hợp lí để giải quyết các yêu cầu của người lập trình.

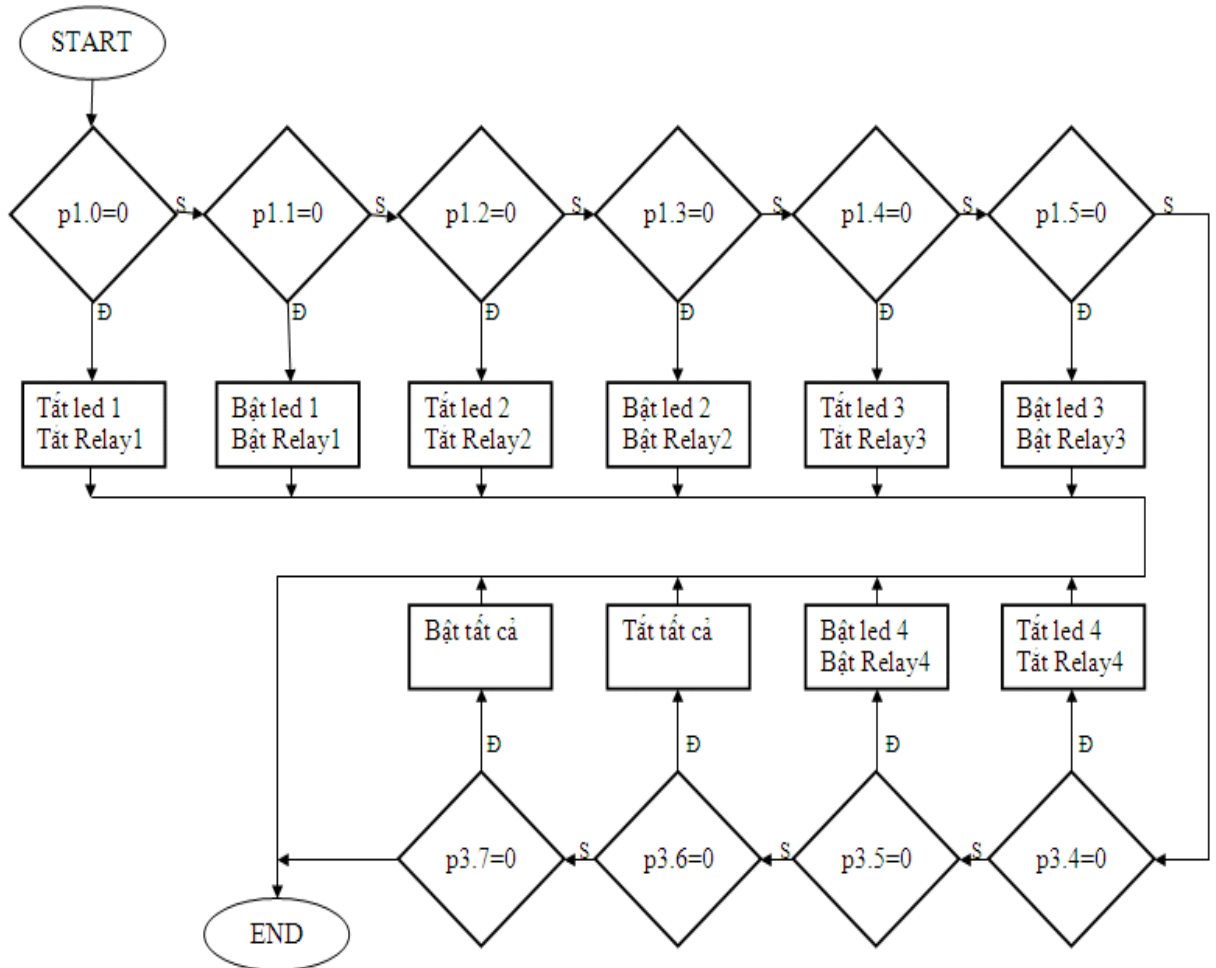
Chương trình cho Vi điều khiển 8051 có thể viết bằng C++, C, Visual Basic v.v. Tuy nhiên hợp ngữ Assembler được đa số người dùng Vi điều khiển sử dụng để lập trình.

### **3.2. HỢP NGỮ ASSEMBLY**

Ngôn ngữ assembly hay hợp ngữ assembly là một ngôn ngữ bậc thấp được dùng trong việc viết các chương trình máy tính. Ngôn ngữ assembly sử dụng các từ có tính gợi nhớ, các từ viết tắt để giúp ta dễ ghi nhớ các chỉ thị phức tạp và làm cho việc lập trình bằng assembly dễ dàng hơn. Mục đích của việc dùng các từ gợi nhớ là nhằm thay thế việc lập trình trực tiếp bằng ngôn ngữ máy được sử dụng trong các máy tính đầu tiên thường gặp nhiều lỗi và tốn thời gian. Một chương trình viết bằng ngôn ngữ assembly được dịch thành mã máy bằng một chương trình tiện ích được gọi là assembler... Ưu điểm của hợp ngữ Assembly là: Mã gọn, ít chiếm dung lượng bộ nhớ, hoạt động với tốc độ nhanh.

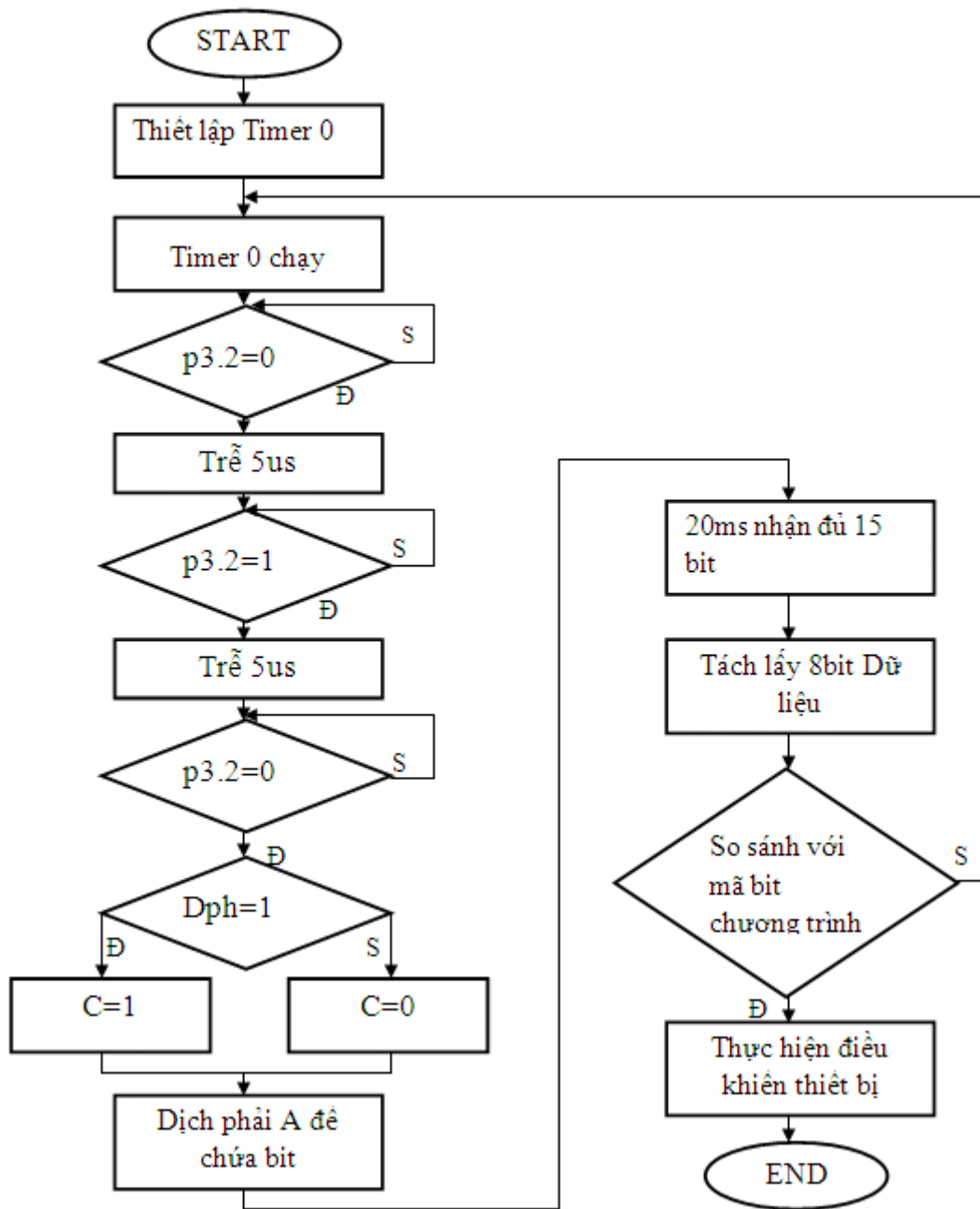
### 3.3. LƯU ĐỒ THUẬT GIẢI

#### 3.3.1. Lưu đồ thuật giải Keyboard



**Hình 2.14:** Lưu đồ thuật giải khi điều khiển bằng keyboard.

### 3.3.1. Lưu đồ thuật giải điều khiển bằng Remote TV Sharp



Hình 2.15: Lưu đồ thuật giải khi điều khiển bằng Remote TV Sharp.

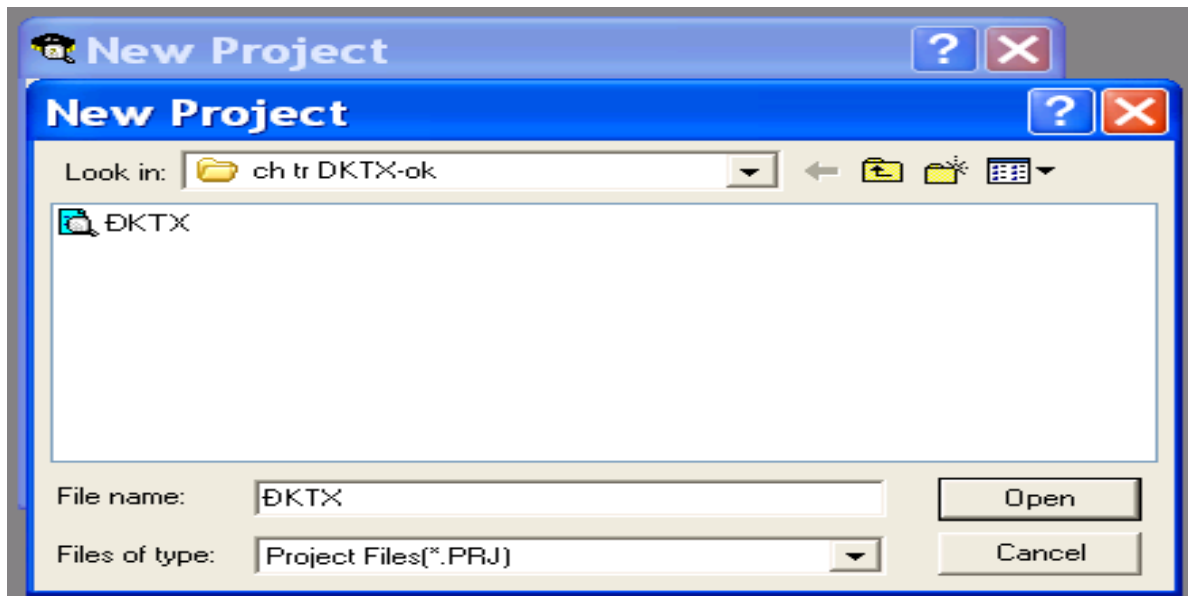
### 3.4. SOẠN THẢO VÀ BIÊN DỊCH CHƯƠNG TRÌNH

Sử dụng phần mềm **Raisonance Kit 6.1**



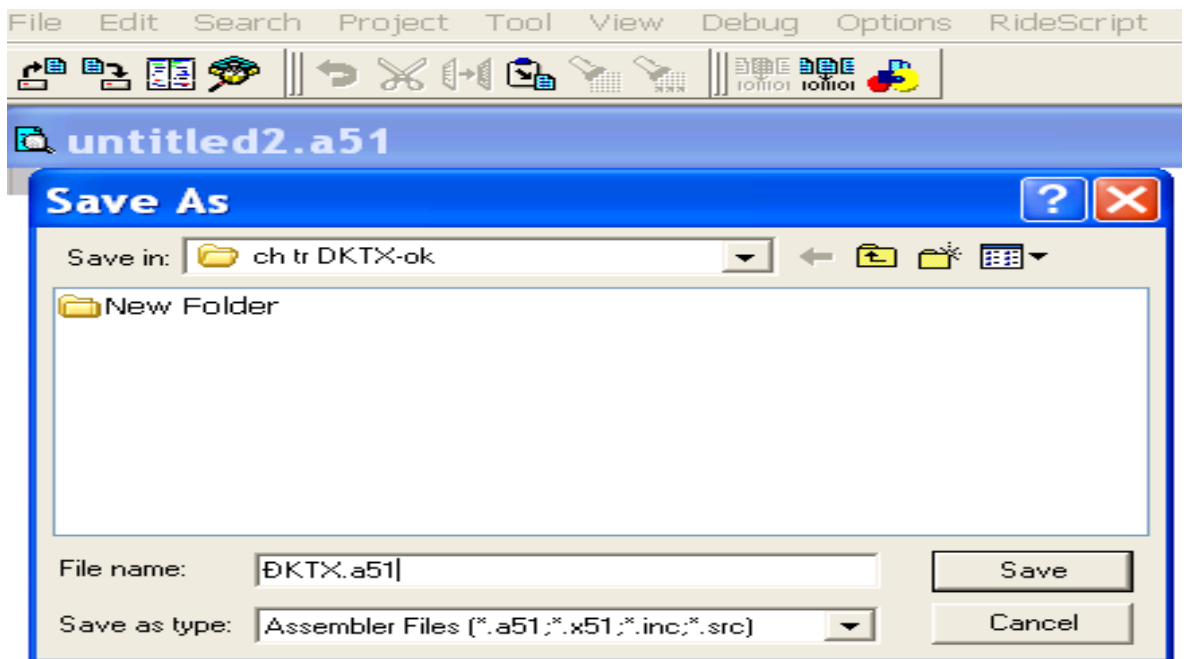
+ Tạo một môi trường làm việc Project.

Project / chọn new / Nhấn **Browse** (chọn đường dẫn tạo file Project)



+ Tạo file Assembler (.a51)

File / New / chọn Assembler (lấy tên ĐKTX.a51)



+ Soạn thảo chương trình trên Assembler (.a51).

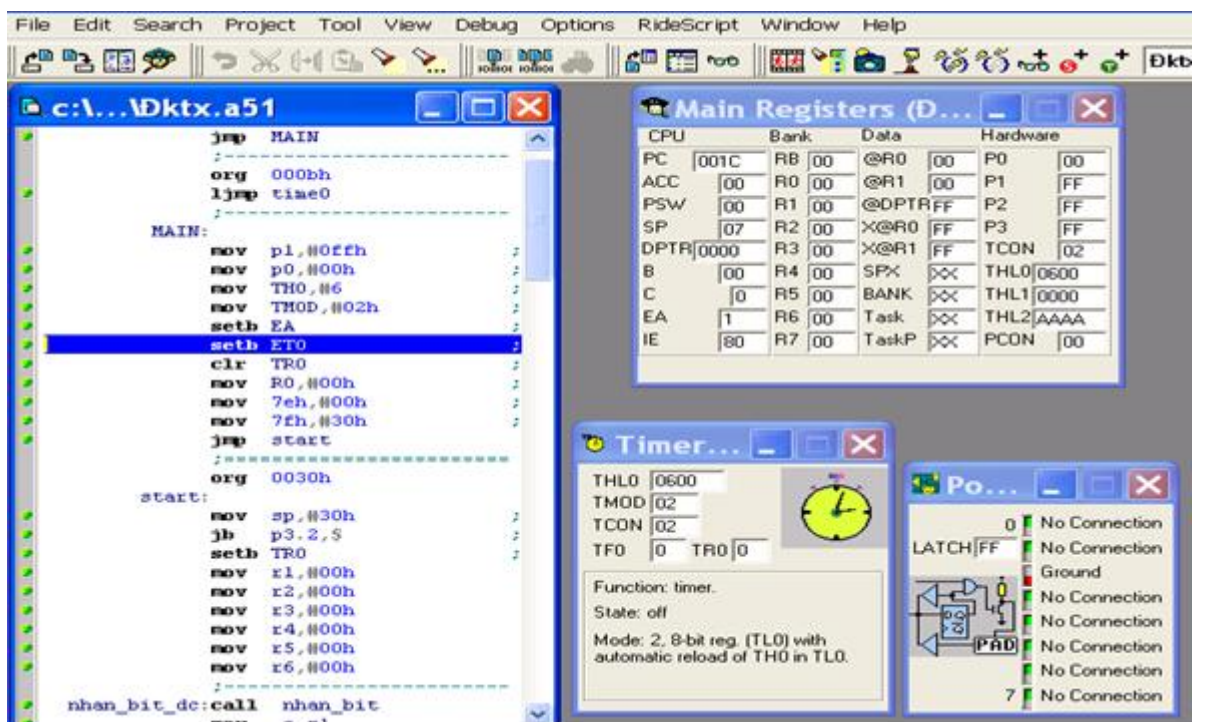
```

File Edit Search Project Tool View Debug Options RideScript Window F
[Icons]

#include (reg52.inc)
    org 0000h
    jmp MAIN
;-----
    org 000bh
    ljmp time0
;-----
MAIN:
    mov p1,#0ffh           ;bat cong p1
    mov p0,#00h           ;tat cong p0
    mov TH0,#6            ;gia tri nap lai cho TL0
    mov TMOD,#02h        ;time 0 che do 2(dem 0-255)
    setb EA              ;cho phep tat ca cac ngat hoạt dor
    setb ETO             ;cho phep ngat bo time0 hoạt động
    clr TR0              ;chua cho time0 chạy

```

+ Biên dịch chương trình Assembler.



Hình 2.16: Đang biên dịch chương trình.

## 3.5. CHƯƠNG TRÌNH ĐIỀU KHIỂN

### 3.5.1. Chương trình kiểm tra keyboard trên mạch

```
$include (reg51.inc)

    Org 0000h
    mov p0,#00h
    mov p2,#0ffh
Start:
    jnb p1.1, on_1
    jnb p1.0,off_1
    jnb p1.3, on_2
    jnb p1.2,off_2
    jnb p1.5, on_3
    jnb p1.4,off_3
    jnb p3.5, on_4
    jnb p3.4,off_4
    jnb p3.7, on_all
    jnb p3.6,off_all
    jmp start
;-----
on_1:setb p0.0
    setb p0.1
    mov p2,#0f9h
    jmp start
off_1:clr p0.0
    clr p0.1
    mov p2,#0ffh
    jmp start
on_2:setb p0.2
    setb p0.3
    mov p2,#0a4h
    jmp start
off_2:clr p0.2
    clr p0.3
    mov p2,#0ffh
    jmp start
on_3:setb p0.4
    setb p0.5
    mov p2,#0b0h
    jmp start
off_3:clr p0.4
    clr p0.5
    mov p2,#0ffh
    jmp start
on_4:setb p0.6
```





```

        setb  p0.7
        mov  p2,#99h
        jmp  start
off_4:  clr  p0.6
        clr  p0.7
        mov  p2,#0ffh
        jmp  start
on_all: mov  p0,#0ffh
        mov  p2,#00h
        jmp  start
off_all: mov p0,#00h
        mov  p2,#0ffh
        jmp  start
End

```

### 3.5.2. Kết quả chạy kiểm tra mạch

Sau khi nạp chương trình vào cho VĐK ta tiến hành nhấn từng nút trên board mạch, kết quả là phần cứng đã hoàn toàn ổn định, các linh kiện đã hoạt động, mạch điện không bị chập, đã điều khiển được đóng mở từng rơ le và các đèn led. Tiếp theo ta soạn thảo chương trình điều khiển với bộ Remoter TV.

### 3.5.3. Chương trình khi điều khiển bằng Remote TV SHARP

```

#include(reg52.inc)
        Org  0000h
        Jmp  MAIN
;-----
        Org  000bh
        ljmp Ngat_T0
;-----
MAIN:
        mov  p1,#0ffh
        mov  p0,#00h
        mov  TH0,#0
        mov  TMOD,#01h
        setb EA
        setb ET0
        clr  TR0

```

```

        mov  R0,#00h  s
        mov  7eh,#00h
        mov  7fh,#30h
        jmp  start
;=====
        Org  0030h
start:
        mov  sp,#30h
        jb   p3.2,$
        setb TR0
        mov  r1,#00h
        mov  r2,#00h
        mov  r3,#00h
        mov  r4,#00h
        mov  r5,#00h
        mov  r6,#00h
;-----
nhan_bit_dc:call  nhan_bit
        mov  a,r1
        rlc  a
        mov  r1,a
        inc  r2
        cjne r2,#05h,nhan_bit_dc
;-----
nhan_bit_dl:call  nhan_bit
        mov  a,r3
        rlc  a
        inc  r4
        cjne r4,#08h,nhan_bit_dl
;-----
nhan_bit_stop:call  nhan_bit
        call delay
        clr  TR0
;=====
        mov  a,r3
;=====
Phim1:cjne  a,#80h,Phim4
        setb p0.0
        mov  p2,#0f9h
        jmp  start

```

```

;-----
Phim4: cjne  a,#20h,Phim2
        clr   p0.0
        clr   p0.1
        mov   p2,#0ffh
        jmp   start
;-----
Phim2:cjne  a,#40h,Phim5
        setb  p0.2
        setb  p0.3
        mov   p2,#0a4h
        jmp   start
;-----
Phim5:cjne  a,#0a0h,Phim3
        clr   p0.2
        clr   p0.3
        mov   p2,#0ffh
        jmp   start
;-----
Phim3:cjne  a,#0c0h,Phim6
        setb  p0.4
        setb  p0.5
        mov   p2,#0b0h
        jmp   start
;-----
Phim6:cjne  a,#60h,Phimvideo
        clr   p0.4
        clr   p0.5
        mov   p2,#0ffh
        jmp   start
;-----
Phim7: cjne  a,#0c0h,Phim10
        setb  p0.6
        setb  p0.7
        mov   p2,#99h
        jmp   start
;-----
Phim10:cjne a,#50h,Phim8
        clr   p0.6
        clr   p0.7

```

```

        mov    p2,#0ffh
        jmp    start
        ;-----
Phim8:cjne a,#10h,Phim11
        mov    p0,#0ffh
        mov    p2,#00h
        jmp    start
        ;-----
Phim11:cjne a,#0d0h,Phim9
        mov    p0,#00h
        mov    p2,#0ffh
Phim9:jmp    start
        ;=====
nhan_bit:mov  dptr,#00h
        jb    p3.2
        call  delay
muc_0:inc    dptr
        jnb  p3.2,muc_0
        call  delay
muc_1:inc    dptr
        jb    p3.2,muc_1
        mov  a,dph
        mov  a,#01h,clr_c
        setb c
        jmp  thoat1
clr_c:clr    c
thoat1:ret
        ;=====
delay:mov    r5,#1
        djnz r5,$
        ret
        ;====ch/tr ngat=====
Ngat_T0:inc  R6
        cjne R6,#160,thoat2
        dec  sp
        dec  sp
        push 7fh
        push 7eh
thoat2:reti
        ;=====

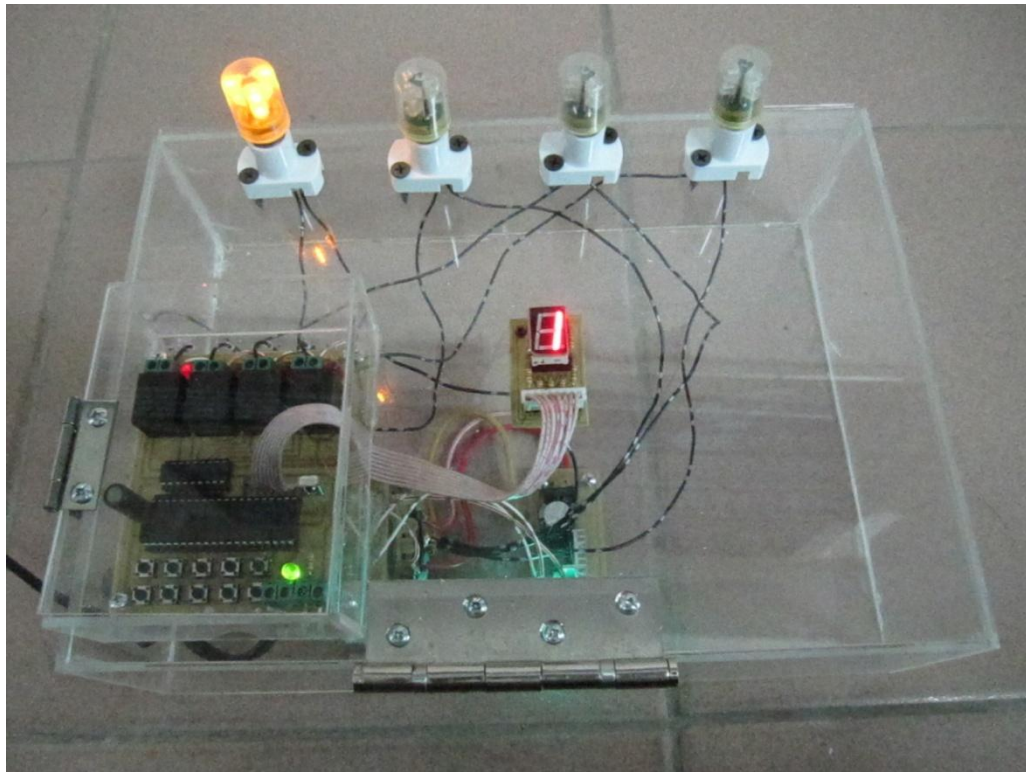
```

End

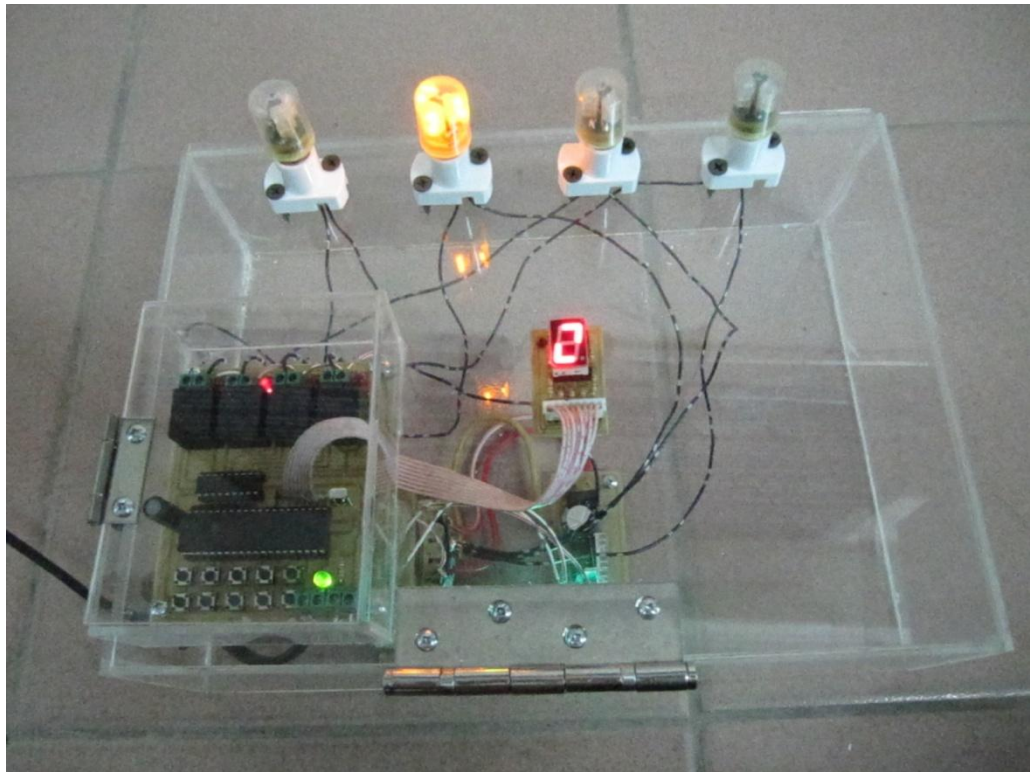
### 3.5.4. Kết quả chạy trên mô hình thực

Sau quá trình kiểm tra mạch ta tiến hành soạn thảo chương trình điều khiển và nạp chương trình vào VĐK 8051.

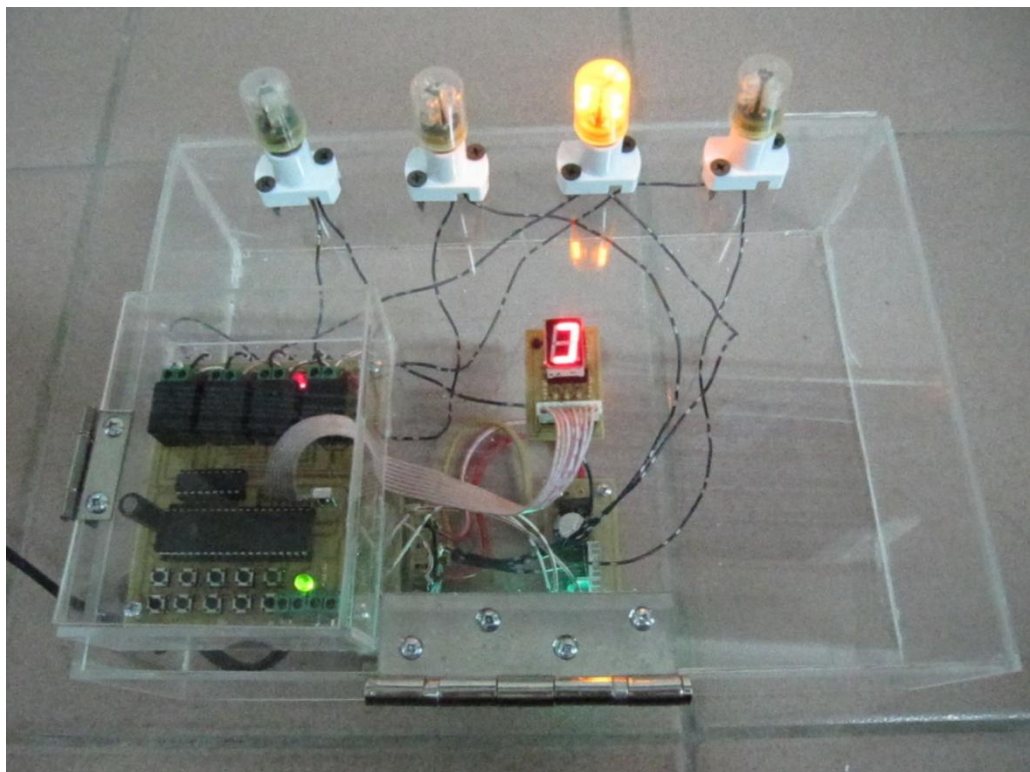
Kết quả là tín hiệu từ điều khiển TV Sharp đã được VĐK thu nhận được tín hiệu về và đã thực hiện việc điều khiển 4 thiết bị qua đóng mở 4 rơ le tương ứng. Các led đơn hiển thị tại các cổng đã sáng. Khi từng thiết bị được bật lên thì led 7 thanh cũng hiển thị tương ứng các giá trị là 1, 2, 3, 4. Chi tiết như sau:



**Hình 2.17:** Điều khiển bật thiết bị 1.

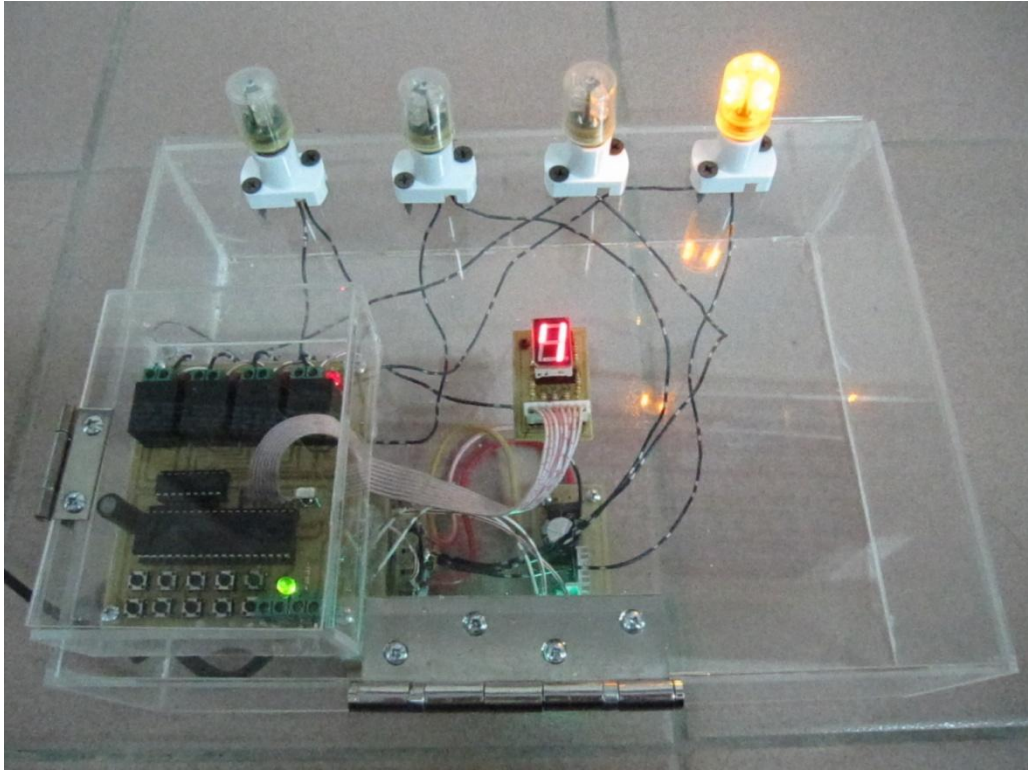


**Hình 2.18:** Điều khiển bật thiết bị 2.

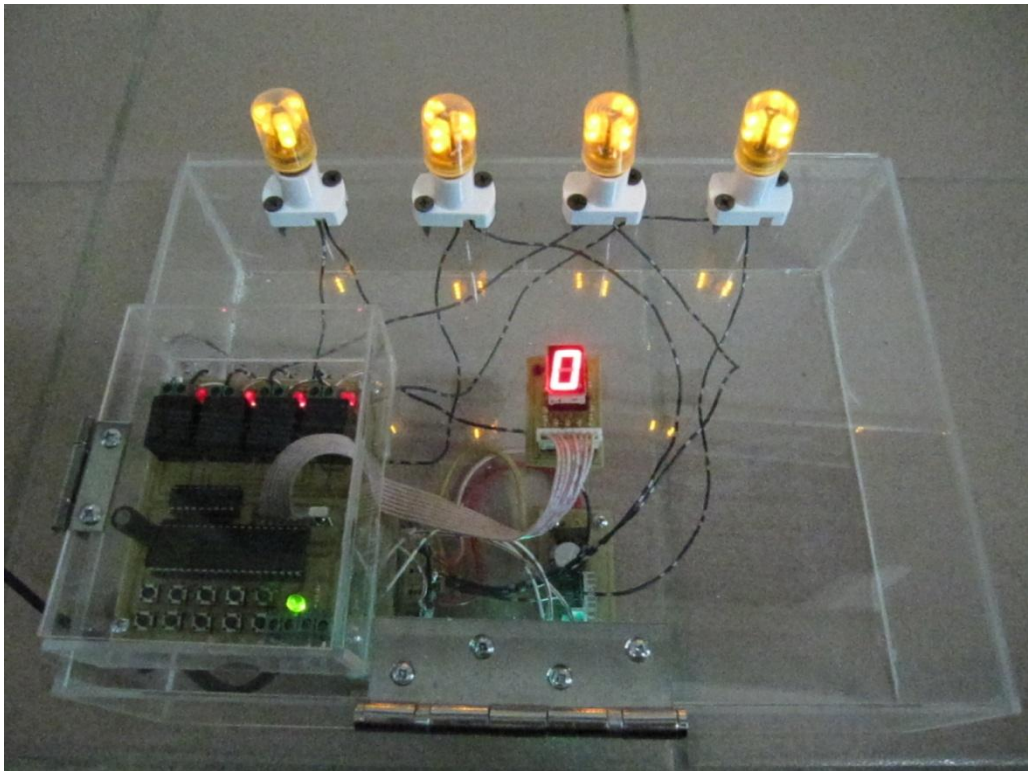


**Hình 2.19:** Điều khiển bật thiết bị 3.





**Hình 2.20:** Điều khiển bật thiết bị 4.



**Hình 2.21:** Điều khiển bật 4 thiết bị.



## NHẬN XÉT

Hợp ngữ Assembly là một ngôn ngữ bậc thấp được dùng trong việc viết các chương trình máy tính. Ngôn ngữ này sử dụng các từ có tính gợi nhớ, các từ viết tắt để giúp ta dễ ghi nhớ các chỉ thị phức tạp và làm cho việc lập trình dễ dàng hơn.

Việc chạy thử với keyboard trên bo mạch là kiểm tra có lỗi nào xảy ra với phần cứng không, nếu tất cả mạch chạy hoàn toàn ổn định ta mới cho chạy chương trình với bộ điều khiển thật.

Một số hạn chế khi điều khiển bằng bộ điều khiển Remote TV là do tín hiệu bit điều khiển được truyền đi nhờ sóng IR với tần số thấp nên hay bị nhiễu do tác động của nhiều nguồn sóng IR khác nhau gây lên. Làm cho khoảng cách điều khiển thiết bị còn ngắn không được xa (< 20m). Do vậy chỉ phù hợp với việc điều khiển các thiết bị điện dân dụng trong nhà cần khoảng cách không lớn.

## KẾT LUẬN

Trên đây em đã trình bày chi tiết các kiến thức về lý thuyết và thao tác thực hiện xoay quanh đề tài :“ **Thiết Kế Mạch Thu Phát Hồng Ngoại Điều Khiển Thiết Bị Từ Xa Sử Dụng Vi Điều Khiển 8051**”. Trong quá trình thực hiện đề tài em đã tìm hiểu được những vấn đề sau:

Việc tìm hiểu về VDK 8051, cấu trúc phần cứng bên ngoài và bên trong, chế độ hoạt động, bộ định thời. Tiếp đến là tín hiệu hồng ngoại như tính chất, đặc trưng và thiết bị phát ra tia hồng ngoại. Trong giới hạn đề tài này là tìm hiểu bộ điều khiển TV Sharp của nhật bản. Tìm hiểu về phần mềm PROTEL 99SE là công cụ chuyên thiết kế mạch nguyên lý và mạch in. Việc soạn thảo và biên dịch trình điều khiển cho VDK 8051 ta dùng phần mềm Raisonance Kit 6.1 là công cụ chuyên để soạn thảo chương trình cho VDK bằng hợp ngữ Assembly.

Quan trọng hơn cả em đã thiết kế chạy thành công mô hình thực hệ thống điều khiển mô phỏng bật tắt 4 bóng đèn. Mô hình trên được xây dựng tính ứng dụng rất cao, rất phù hợp với việc điều khiển các thiết bị điện dân dụng trong các căn hộ.

Trong thời gian làm đồ án tốt nghiệp được sự quan tâm của các Thầy, Cô trong khoa Điện tử viễn thông , đặc biệt là giáo viên hướng dẫn **Ths.ĐOÀN HỮU CHỨC** đã giúp đỡ em tận tình trong quá trình tìm hiểu và xây dựng mô hình điều khiển. Để đồ án của em hoàn thành đúng thời gian. Tuy nhiên khi thực hiện đề tài không tránh khỏi những sai sót, em mong nhận được sự góp ý đánh giá của quý thầy cô.

Em xin trân thành cảm ơn

!

## TÀI LIỆU THAM KHẢO

1. Tống Văn On - Hoàng Đức Hải (2005), *Họ vi điều khiển 8051*. Nhà xuất bản lao động – xã hội.
2. Nguyễn Bính (2007), *Giáo trình điện tử công suất*. Nhà xuất bản đại học Quốc Gia.
3. Phan Quốc Phô - Nguyễn Đức Chiến (2000), *Giáo trình cảm biến*. Nhà xuất bản khoa học và kỹ thuật.
4. Đỗ Xuân Thụ (1999), *Kỹ thuật điện tử*. Nhà xuất bản giáo dục.
5. Đỗ Xuân Tiên (2001), *Kỹ thuật Vi Xử Lý & Lập Trình Assembly cho hệ Vi Xử Lý*. Nhà xuất bản khoa học và kỹ thuật.
6. Diễn đàn <http://www.dientuvietnam.net>
7. Website <http://www.ebook.edu.vn>
8. Website <http://www.tailieu.vn>
9. Website <http://lib.hpu.edu.vn>