

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG



**ISO 9001 : 2008**

**ỨNG DỤNG DS1307  
XÂY DỰNG HỆ THỐNG THỜI GIAN THỰC**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP**

**Sinh viên : Nguyễn Ngọc Trung**

**Giáo viên hướng dẫn: Th.S Nguyễn Trọng Thắng**

**Hải Phòng - 2011**

# MỤC LỤC

	<i>Trang</i>
<b>LỜI MỞ ĐẦU</b> .....	1
<b>CHƯƠNG 1. TỔNG QUAN HỆ THỐNG THỜI GIAN THỰC</b> .....	2
1.1. GIỚI THIỆU CHUNG.....	2
1.1.1. Giới thiệu về hệ thống thời gian thực .....	2
1.1.2. Khái niệm về hệ thống thời gian thực .....	2
1.1.3. Các loại hệ thống thời gian thực .....	3
1.2. TỔNG QUAN HỆ THỐNG .....	6
<b>CHƯƠNG 2. CÁC PHẦN TỬ TRONG HỆ THỐNG</b> .....	7
2.1. SƠ ĐỒ TỔNG THỂ .....	7
2.1.1. Sơ đồ đặc tả các khối.....	8
<b>2.1.1.1.</b> Khối nguồn .....	8
<b>2.1.1.2.</b> Khối Reset .....	8
<b>2.1.1.3.</b> Khối điều khiển .....	9
<b>2.1.1.4.</b> Khối tạo xung dao động .....	10
<b>2.1.1.5.</b> Khối hiển thị .....	10
<b>2.1.1.6.</b> Khối tạo thời gian thực .....	11
2.2 LỰA CHỌN GIẢI PHÁP .....	12
2.2.1 Giải pháp công nghệ.....	12
2.2.2 Giải pháp thiết kế .....	13
2.2.2.1 Quy trình Top-down .....	13
2.2.2.2 Quy trình Bottom-Up .....	15
2.2.3 Sơ đồ Call graph.....	17
2.2.4 Các yêu cầu và giới hạn cho hệ thống.....	17
2.2.4.1 Các yêu cầu .....	17

2.2.4.2. Giới hạn cho hệ thống.....	17
2.3. LỰA CHỌN TỔNG QUAN VỀ LINH KIỆN .....	18
2.3.1. IC thời gian thực DS1307 .....	18
2.3.1.1. Giới thiệu chung về DS1307.....	18
2.3.1.2. Cơ chế hoạt động và chức năng của DS1307 .....	19
2.3.1.3. Mô tả hoạt động của các chân.....	20
2.3.2. Vi Điều Khiển AT89C52 .....	27
2.3.2.1. Cấu tạo và chức năng các khối của AT89C52.....	27
2.3.2.2. Chức năng các chân của AT89C52.....	28
2.3.3. Led 7 thanh.....	29
2.3.3.1. Các khái niệm cơ bản.....	29
2.3.3.2. Sơ đồ vị trí các led .....	30
2.3.3.3. Kết nối với vi điều khiển .....	31
2.3.3.4. Bảng mã của Led Anode chung.....	31
2.3.4. IC giải mã 74HC138.....	32
2.4. LINH KIỆN KHÁC.....	33
2.4.1 IC nguồn 7805 .....	33
2.4.2 Tụ điện.....	34
2.4.3 Điện trở.....	35
<b>2.4.4. Nút bấm button .....</b>	<b>35</b>
<b>2.4.5. Pin CMOS 3V.....</b>	<b>36</b>
2.4.6. Thạch anh .....	36
<b>CHƯƠNG 3. THIẾT KẾ VÀ THI CÔNG HỆ THỐNG HIỂN THỊ THỜI</b>	
<b>GIAN THỰC .....</b>	<b>38</b>
3.1. THIẾT KẾ PHẦN CỨNG.....	38
3.1.1. Sơ đồ nguyên lý mạch điều khiển.....	38
3.1.2. Sơ đồ nguyên lý mạch hiển thị .....	39
3.1.3. Sơ đồ mạch in mạch điều khiển.....	40

3.1.4. Sơ đồ mạch in mạch hiển thị .....	
<b>3.2. THIẾT KẾ PHẦN MỀM .....</b>	<b>43</b>
3.2.1 Lưu đồ thuật toán.....	43
3.2.2. Lưu đồ hiển thị thời gian thực lên LED 7 thanh.....	43
3.2.3. Lưu đồ chỉnh sửa thời gian .....	45
<b>3.3. MÃ NGUỒN CHƯƠNG TRÌNH .....</b>	<b>46</b>
<b>KẾT LUẬN.....</b>	<b>59</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>60</b>
<b>PHỤ LỤC .....</b>	<b>61</b>

## LỜI MỞ ĐẦU

Ngày nay việc ứng dụng vi điều khiển, vi xử lý đang ngày càng phát triển rộng rãi và thâm nhập ngày càng nhiều vào các lĩnh vực kỹ thuật và đời sống xã hội. Tuy nhiên ứng dụng cho các hệ thống nhúng ngày nay không đơn giản chỉ dừng lại ở điều khiển đèn nhấp nháy, đếm số người vào/ra, hiển thị dòng thông báo trên matrix led hay điều khiển ON-OFF của động cơ... mà nó ngày càng trở nên phức tạp.

Với sự phát triển mạnh mẽ của khoa học, đặc biệt là ngành điện tử, sự phát minh ra các linh kiện điện tử đã và đang ngày càng đáp ứng được yêu cầu của các hệ thống. Ưu điểm của việc sử dụng các linh kiện điện tử làm cho các hệ thống linh hoạt và đa dạng hơn, giá thành thấp hơn và độ chính xác cao hơn.

Sau gần 2 năm học tập và nghiên cứu ở trường, em đã được làm quen với các môn học chuyên ngành. Để áp dụng lý thuyết với thực tế học kỳ này em được giao đề án tốt nghiệp với tên đề tài “**Ứng dụng DS1307 xây dựng hệ thống thời gian thực**”

Tuy nhiên do kiến thức chuyên môn còn hạn chế, tài liệu tham khảo có giới hạn nên còn xảy ra nhiều sai sót. Em rất mong thầy, cô và các bạn góp ý bổ sung để bản đề án của em được hoàn thiện hơn và giúp em hiểu biết hơn trong quá trình học tập tiếp theo.

**Em xin chân thành cảm ơn!**

Sinh viên thực hiện

Nguyễn Ngọc Trung

## **CHƯƠNG 1.**

# **TỔNG QUAN HỆ THỐNG THỜI GIAN THỰC**

## **1.1. GIỚI THIỆU CHUNG.**

### **1.1.1. Giới thiệu về hệ thống thời gian thực.**

Trong những năm gần đây, các hệ thống điều khiển theo thời gian thực là một trong những lĩnh vực thu hút nhiều sự chú ý trong giới khoa học nghiên cứu về khoa học máy tính. Trong đó, vấn đề điều hành thời gian thực và vấn đề lập lịch là đặc biệt quan trọng. Một trong ứng dụng quan trọng của hệ thống thời gian thực (RTS) đã và đang được ứng dụng rộng rãi hiện nay là các dây chuyền sản xuất tự động, robot, điều khiển các thí nghiệm tự động, trong thiết kế đồng hồ hiển thị thời gian thực... Thế hệ ứng dụng tiếp theo của hệ thống này sẽ điều khiển robot giống con người, hệ thống kiểm soát thông minh trong các nhà máy công nghiệp, điều khiển các trạm không gian...

### **1.1.2. Khái niệm về hệ thống thời gian thực.**

Một hệ thống thời gian thực (RTC) có thể hiểu như là một mô hình xử lý mà tính đúng đắn của hệ thống không chỉ phụ thuộc vào kết quả tính toán logic mà còn phụ thuộc vào thời gian do kết quả này phát sinh ra.

Hệ thống thời gian thực thiết kế nhằm cho phép trả lời lại các yếu tố kích thích phát sinh từ các thiết bị phân cứng trong một ràng buộc thời gian xác định. Ở đây ta có thể hiểu thế nào là một RTS (real time systems) bằng cách hiểu thế nào là một tiến trình, một công nghệ thời gian thực. Nhìn chung trong những RTS chỉ có một số công việc được gọi là công việc thời gian thực, các công việc này có một mức độ khẩn cấp riêng phải hoàn tất. Sự thay đổi của sự kiện trong thế giới thực xảy ra rất nhanh, mỗi tiến trình giám sát sự kiện này phải thực hiện việc xử lý trong một khoảng thời gian ràng buộc gọi là deadline, khoảng thời gian ràng buộc này được xác định bởi thời gian bắt đầu và thời gian hoàn tất công việc. Trong thực tế, các yếu tố kích thích này

xảy ra trong thời gian rất ngắn vào khoảng vài mili giây, thời gian mà hệ thống trả lại yếu tố kích thích đó tốt nhất vào khoảng dưới một giây, thường vào khoảng vài chục mili giây, khoảng thời gian này bao gồm thời gian tiếp nhận kích thích, xử lý thông tin và trả lời kích thích. Một yếu tố khác cần quan tâm trong RTS là những công việc thời gian thực này có tuần hoàn hay không? Công việc tuần hoàn thì ràng buộc thời gian ấn định trong từng chu kỳ xác định, công việc không tuần hoàn xảy ra với ràng buộc thời gian vào lúc bắt đầu và kết thúc công việc, ràng buộc này chỉ được xác định vào lúc bắt đầu công việc. Các biến cố kích hoạt công việc không tuần hoàn thường dựa trên kỹ thuật xử lý ngắt của hệ thống phần cứng.

Về cấu tạo, RTS thường được cấu thành từ các thành tố chính sau:

- Đồng hồ thời gian thực: Cung cấp thông tin thời gian thực.
- Bộ điều khiển ngắt: Quản lý các biến cố không theo chu kỳ.
- Bộ định hiệu: Quản lý các quá trình thực hiện.
- Bộ quản lý tài nguyên: Cung cấp các tài nguyên máy tính.
- Bộ điều khiển thực hiện: Khởi động các tiến trình.

Các thành tố trên có thể được phân định là thành phần cứng hay phần mềm tùy thuộc vào hệ thống và ý nghĩa sử dụng. Thông thường các RTS được kết hợp vào phần cứng có khả năng tốt hơn so với phần mềm có chức năng tương ứng và tránh được chi phí quá đắt cho việc tối ưu hóa phần mềm. Ngày nay chi phí phần cứng ngày càng rẻ, chọn lựa ưu tiên phần cứng là một xu hướng chung.

### **1.1.3. Các loại hệ thống thời gian thực.**

Các RTS thường được phân thành hai loại Hệ thống thời gian thực cứng (Hard realtime system) và Hệ thống thời gian thực mềm (Soft realtime system):

Hệ thống thời gian thực cứng là hệ thống mà các hành động của nó phải không bao giờ vi phạm các ràng buộc thời gian trong đó có thời hạn lập lịch, hệ thống phải tiếp nhận và nắm bắt được thời hạn lập lịch của nó tại mọi thời

điểm. Hệ thống có lỗi hoặc sai sót trong việc tiếp nhận thời hạn sẽ gây ra hậu quả nghiêm trọng, thiệt hại về vật chất, gây ảnh hưởng sâu đến sức khỏe, đời sống con người, thậm chí chết người. Với hệ thống thời gian thực cứng dữ liệu trễ là không tốt. Một ví dụ về hệ thống thời gian thực cứng là hệ thống kiểm soát không lưu. Trong hệ thống này, một phân phối đường bay, thời gian cất cánh, thời gian hạ cánh không hợp lý, không đúng lúc có thể gây ra tai nạn máy bay mà hậu quả của nó khó mà lường trước được.

Ngược lại, hệ thống thời gian thực mềm thời gian trả về của hệ thống cho các yếu tố kích thích quan trọng, tuy nhiên trong trường hợp ràng buộc này bị vi phạm, tức là thời gian trả về của hệ thống vượt quá giới hạn trễ cho phép, hệ thống vẫn cho phép tiếp tục hoạt động bình thường, không quan tâm đến các tác hại do sự vi phạm này gây ra.

Trong cả hai loại này, máy tính thường can thiệp trực tiếp hoặc gián tiếp đến các thiết bị vật lý để kiểm soát cũng như điều khiển sự hoạt động của thiết bị này. Đứng trên góc độ này người ta chia các RTS ra làm hai loại sau:

- Embedded system: Bộ xử lý điều khiển là một phần trong toàn bộ thiết bị, nó được sản xuất trọn gói từ yếu tố cứng từ nhà máy, người ta sử dụng không biết về chi tiết của nó mà thông qua các nút điều khiển, các bảng số. Với hệ thống này, ta không thấy được các thiết bị như trong máy tính bình thường như bàn phím, màn hình... mà thay vào đó là các nút điều khiển, các bảng số, các bảng số hay các màn hình chuyên dụng đặc trưng cho các hệ thống, máy giặt là một ví dụ. Người sử dụng chỉ việc bấm nút chọn chương trình giặt, xem kết quả qua hệ thống đèn tín hiệu... Bộ vi xử lý trong Embedded system này đã được lập trình trước và gắn chặt vào ngay từ khi sản xuất và không thể lập trình lại những chương trình này hoạt động độc lập, không có sự giao tiếp với hệ điều hành cũng như không cho phép người sử dụng can thiệp vào.



- Loại thứ hai là bao gồm những hệ thống có sự can thiệp của máy tính thông thường. Thông qua máy tính ta hoàn toàn có thể kiểm soát cũng như điều khiển mọi hoạt động của thiết bị phần cứng của hệ thống này. Những chương trình điều khiển này có rất nhiều loại, phục vụ cho nhiều mục đích khác nhau và có thể viết lại cho phù hợp với yêu cầu thực tế. Hiển nhiên thì loại hệ thống này hoạt động được phải cần một hệ điều hành(HĐH) điều khiển máy tính. HĐH này phải có khả năng nhận biết được thiết bị phần cứng, có khả năng hoàn tất công việc trong giới hạn thời gian nghiêm ngặt. HĐH này phải là HĐH hỗ trợ xử lý thời gian thực Realtime operation system (RTOS)



**Hình 1.1:** GPS-tracker- thời gian thực trong các phương tiện.



**Hình 1.2:** Đồng hồ điện tử hiển thị thời gian thực

## 1.2. TỔNG QUAN HỆ THỐNG

Đây là ứng dụng sử dụng vi điều khiển để thiết kế một đồng hồ thời gian thực hiện thị trên led 7 thanh, với yêu cầu đảm bảo về:

- + Tính thực thi cao, có khả năng phát triển.
- + Đảm bảo về chất lượng, độ chính xác cao, làm việc lâu dài, bền bỉ.
- + Tiết kiệm chi phí, linh kiện dễ kiếm dễ sử dụng và dễ dàng thay thế khi xảy ra sự cố.
- + Giảm thiểu chi phí, thời gian vận hành, bảo dưỡng và sửa chữa.
- + Có thể sử dụng riêng (đồng hồ vạn niên xem giờ, ngày tháng năm) hay có thể sử dụng chung (lắp vào các hệ thống mẹ).

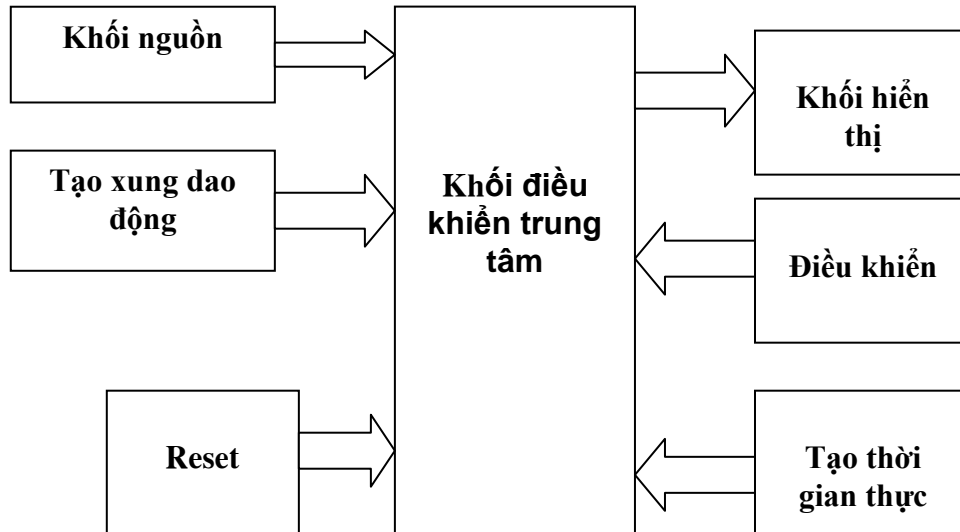
Từ các yêu cầu trên về hệ thống chúng ta thiết kế hệ thống theo hướng sử dụng hệ thống thời gian thực cứng với các ưu điểm của nó. Sau đây là một vài điểm giới thiệu sơ lược:

- Sử dụng LED 7 đoạn để hiển thị vì giá thành rẻ, dễ tìm kiếm.
- Sử dụng IC thời gian thực DS1307. IC này có tác dụng tạo ra thời gian thực tương đối chính xác, bao gồm giờ, phút, giây, thứ, ngày, tháng, năm
- Sử dụng họ vi điều khiển MCS-51(Atmel).
- Sử dụng IC ghi dịch 74HC138 để tăng số lượng chân điều khiển cho vi điều khiển.

## CHƯƠNG 2.

### CÁC PHẦN TỬ TRONG HỆ THỐNG

#### 2.1. SƠ ĐỒ TỔNG THỂ.

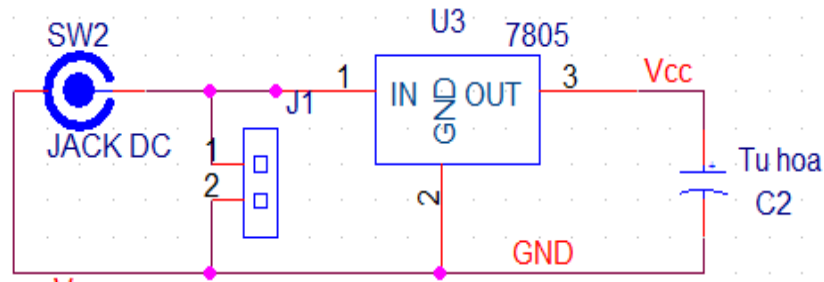


**Hình 2.1** : Sơ đồ tổng quát của hệ thống

Nguyên lý hoạt động của sơ đồ tổng thể : Khi cho điện áp qua khối nguồn cho vi điều khiển, khi đó chương trình trong vi điều khiển sẽ làm việc, đồng thời bộ tạo xung dao động tạo xung nhịp với tần số 12MHz cho VDK hoạt động. Chế độ ghi và nhận dữ liệu của IC thời gian thực đưa tới vi điều khiển, các điều kiện START và STOP được nhận dạng khi bắt đầu hoặc kết thúc truyền một chuỗi, lúc này các thanh ghi của IC thời gian thực nhận giá trị thời gian thực (giờ, phút, giây, thứ, ngày, tháng, năm) và gửi đến vi điều khiển đồng thời lúc này vi điều khiển sẽ gán một giá trị tương đương giá trị thời gian thực rồi gửi ra khối hiển thị. Lúc này IC ghi dịch trong khối điều khiển sẽ gửi tín hiệu đến khối hiển thị. Các nút ấn trong khối điều khiển có nhiệm vụ điều chỉnh thời gian. Khối Reset có nhiệm vụ đưa hệ thống về trạng thái ban đầu.

### 2.1.1. Sơ đồ đặc tả các khối

#### 2.1.1.7. Khối nguồn



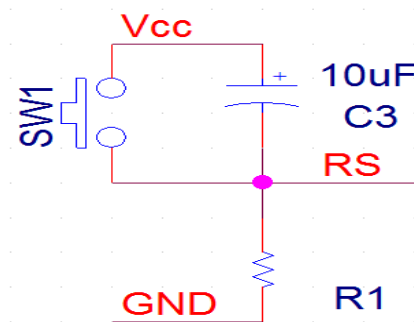
**Hình 2.2 :** Sơ đồ khối nguồn

Đây là mạch dùng để tạo ra nguồn điện áp chuẩn +5V cấp cho khối điều khiển trung tâm sử dụng IC7805.

Đầu vào là điện áp xoay chiều sau khi được biến đổi qua máy biến thế, đưa vào bộ Diode cầu để cho ra dòng điện một chiều( lúc này điện áp nằm trong khoảng từ 7-10V). Sau khi đi qua IC ổn áp 7805 sẽ tạo ra nguồn điện áp chuẩn +5V cung cấp cho mạch.

IC ổn áp 7805: đầu vào > 7V đầu ra 5V, 500 mA. Mạch ổn áp: cần cho vi điều khiển vì nếu nguồn cho vi điều khiển không ổn định thì sẽ treo VĐK, không chạy đúng hoặc reset liên tục thậm chí là chết chip.

#### 2.1.1.8. Khối Reset

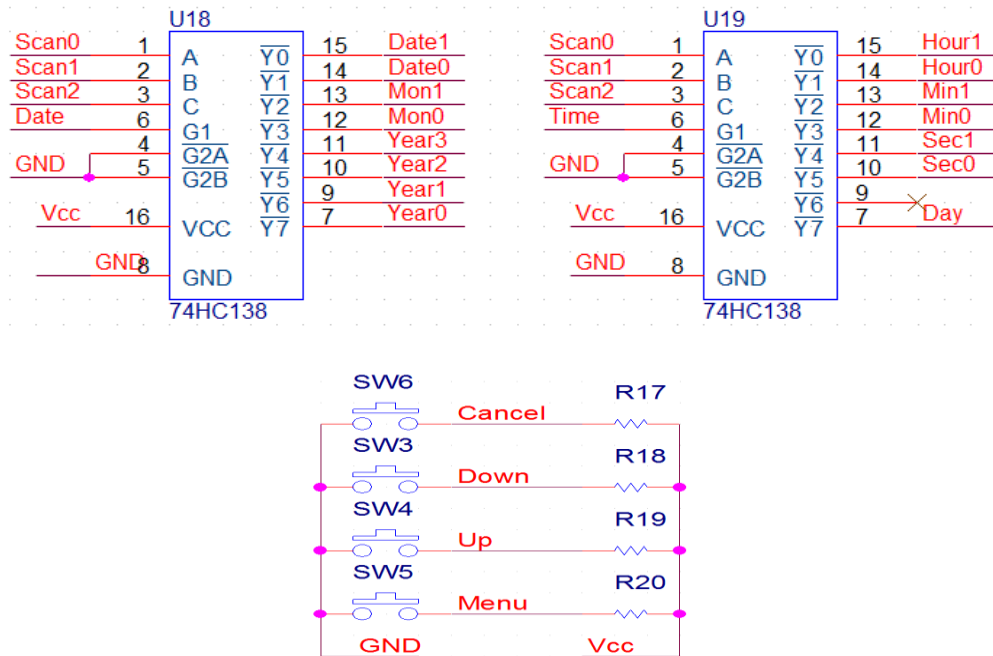


**Hình 2.2 :** Sơ đồ khối reset

Khối RESET có tác dụng đưa vi điều khiển về trạng thái ban đầu. Khi nút Reset được ấn điện áp +5V từ nguồn được nối vào chân Reset của vi điều khiển được chạy thẳng xuống đất lúc này điện áp tại chân vi điều khiển thay

đổi đột ngột về 0. Khối điều khiển nhận biết được sự thay đổi này và khởi động lại trạng thái ban đầu cho hệ thống.

### 2.1.1.9. Khối điều khiển.



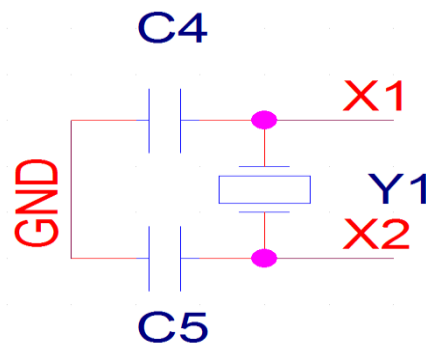
**Hình 2.3 :** Sơ đồ khối điều khiển

Gồm 4 nút ấn:cancel, down, up, menu. Khi 1 nút ấn được tác động làm thay đổi điện áp trên chân nối với vi điều khiển từ +5V xuống 0V. Lúc này vi điều khiển nhận biết được sự thay đổi và làm thay đổi giá trị đầu ra:

- Nút menu: Để chuyển chế độ chỉnh thời gian.
- Nút up: Tăng giá trị cần điều chỉnh ++1.
- Nút down: Giảm giá trị cần điều chỉnh –1.
- Cancel: thoát trạng thái điều chỉnh.

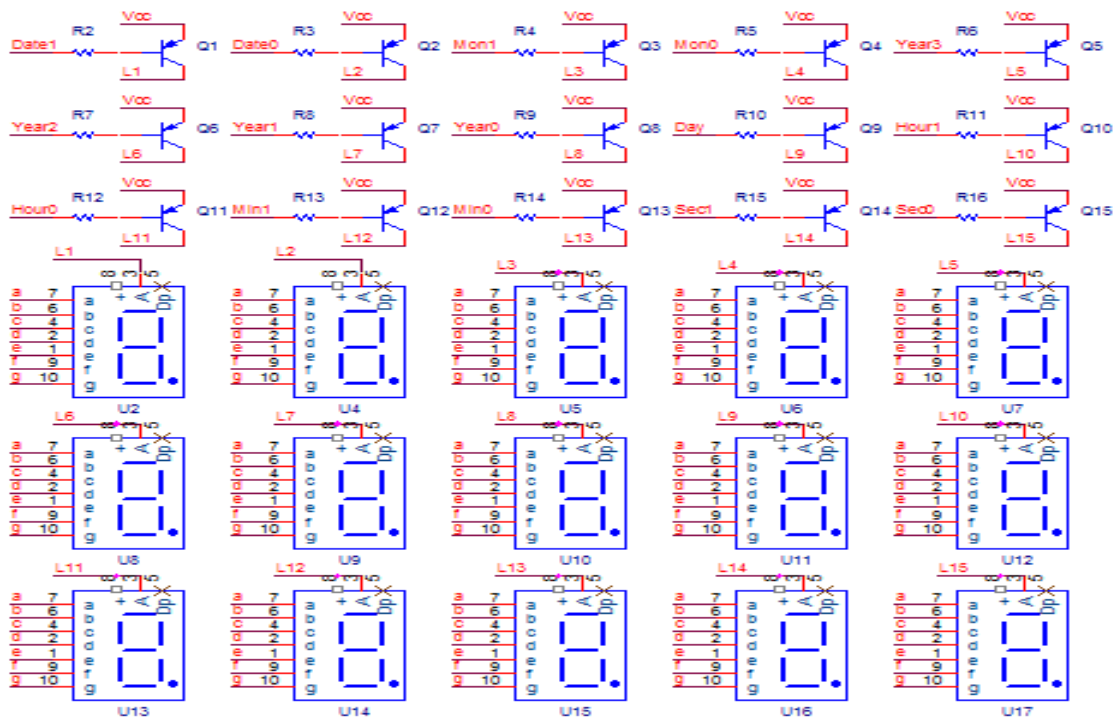
IC 74HC138 là bộ giả mã địa chỉ với 3 đầu vào ( A,B,C) và 8 đầu ra phủ định (Y0 đến Y7 ). Nó có 3 đầu vào cho phép: 2 đầu vào tích cực thấp (G2A,G2B) và một đầu vào tích cực mức cao (G1). Tất cả các đầu ra của 74HC138 sẽ ở mức cao trừ khi G2A ở mức thấp và G1 ở mức cao. Khi các đầu vào G2A,G2B ở mức thấp và G1 ở mức cao thì đầu ra của 74HC138 sẽ được quyết định bởi đầu vào .

### 2.1.1.10. Khối tạo xung dao động.



Đây là bộ dao động thạch anh có tác dụng tạo xung nhịp với tần số 12MHz cho VĐK hoạt động. Hai đầu này được nối vào 2chân XTAL1 và XTAL2 của VĐK.

### 2.1.1.11. Khối hiển thị:

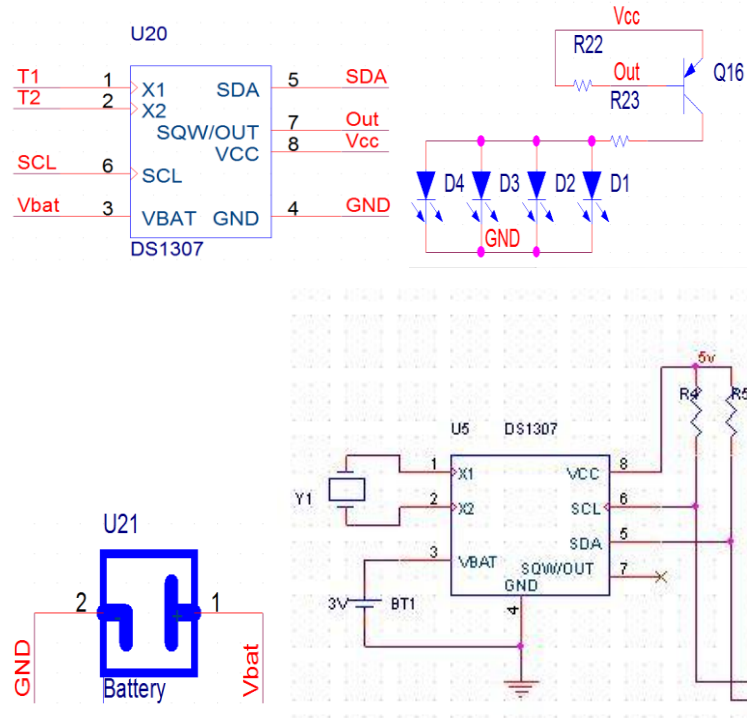


**Hình 2.4 :** Sơ đồ khối hiển thị

Khối hiển thị bao gồm các LED 7 thanh đơn (Anode chung) có các đầu vào a,b,c,d,e,f,g của các LED được nối song song với nhau và nối với các chân của VĐK (từ chân P0-P3) có tác dụng làm cho LED hiển thị dạng số mong muốn. Và đầu còn lại của 15 LED 7 thanh được nối với 15 chân C của transistor thuận và chân B của transistor nối với các PORT của VĐK (từ P0-

>P3), chân E của transistor được nối với +5V. VĐK làm nhiệm vụ điều khiển IC 74HC138 làm cho từng LED sáng trong khoảng thời gian nhất định.

### 2.1.1.12. Khối tạo thời gian thực.



**Hình 2.5 :** Sơ đồ khối tạo thời gian thực

DS1307 là một IC thời gian thực với nguồn cung cấp nhỏ, dùng để cập nhật thời gian và ngày tháng với 56 bytes SRAM. Địa chỉ và dữ liệu được truyền nối tiếp qua 2 đường bus 2 chiều. Nó cung cấp thông tin về giờ, phút, giây, thứ, ngày, tháng, năm. Ngày cuối tháng sẽ tự động được điều chỉnh với các tháng nhỏ hơn 31 ngày, bao gồm cả việc tự động nhảy năm. Đồng hồ có thể hoạt động ở dạng 24h hoặc 12h với chỉ thị AM/PM.

Để không phải điều chỉnh lại thời gian vào những lúc bị mất nguồn, có thể nối thêm 1 pin khoảng 3V vào chân SQW/OUT của IC DS1307 (sao cho chân + của pin nối vào IC và chân – của pin nối xuống đất). Hai chân 1 và 2 (X1,X2) của DS1307 được nối vào bộ dao động thạch anh có tần số 32,768KHz để tạo dao động cho IC hoạt động.

## 2.2. LỰA CHỌN GIẢI PHÁP.

### 2.2.1. Giải pháp công nghệ.

- Dựa vào yêu cầu : “**Ứng dụng DS1307 xây dựng hệ thống thời gian thực**” và các kiến thức đã học trong chương trình.

+ Sử dụng LED 7 thanh thích hợp nhất với mục đích hiển thị của RTC khi dùng làm đồng hồ, với yêu cầu hiển thị thời gian với những dãy số không đòi hỏi quá phức tạp. LCD hiển thị linh hoạt hơn song LED 7 thanh có nhiều ưu thế riêng như ít chịu ảnh hưởng của nhiệt độ, góc nhìn rộng, đơn giản trong cấu tạo và sử dụng, rẻ hơn, dễ tạo sự chú ý.

+ Sử dụng IC thời gian thực DS1307 phù hợp với yêu cầu bài toán. IC này có tác dụng tạo ra thời gian thực chính xác cao, bao gồm giờ, phút, giây, thứ, ngày, tháng, năm.

+ Sử dụng vi điều khiển AT89C52 cho các thao tác truy cập thời gian thực, hiển thị giờ và chỉnh giờ phù hợp với phạm vi bài toán nhỏ không phức tạp. Ta cũng có thể sử dụng PIC nhưng do yêu cầu bài toán chỉ cần sử dụng chức năng I/O mà không cần sử dụng chức năng phụ nào khác của vi điều khiển ngoài nên việc dùng PIC là lãng phí. Với việc lần đầu sử dụng thì dùng AT89C52 sẽ đơn giản hơn, tránh được hỏng hóc nhiều hơn so với sử dụng PIC.

+ Sử dụng IC ghi dịch 74HC138. Đây là IC ghi dịch nối tiếp song song. IC được ứng dụng để tăng số lượng chân output cho vi điều khiển. Có nhiều phương pháp để tăng số lượng chân như dùng IC giải mã, tuy nhiên IC ghi dịch 74HC138 được lựa chọn với các nguyên nhân sau:

Đầu vào 3 chân có thể điều khiển được 8 chân đầu ra.

Cho phép điều khiển linh hoạt và ổn định hơn: giữa các thanh ghi dịch và ngõ ra có một “chốt”. Điều này cho phép thay đổi linh hoạt dữ liệu trong các thanh ghi dịch và ổn định trạng thái logic ngõ ra.

Giá thành rẻ, dễ kiếm.



### **2.2.2. Giải pháp thiết kế.**

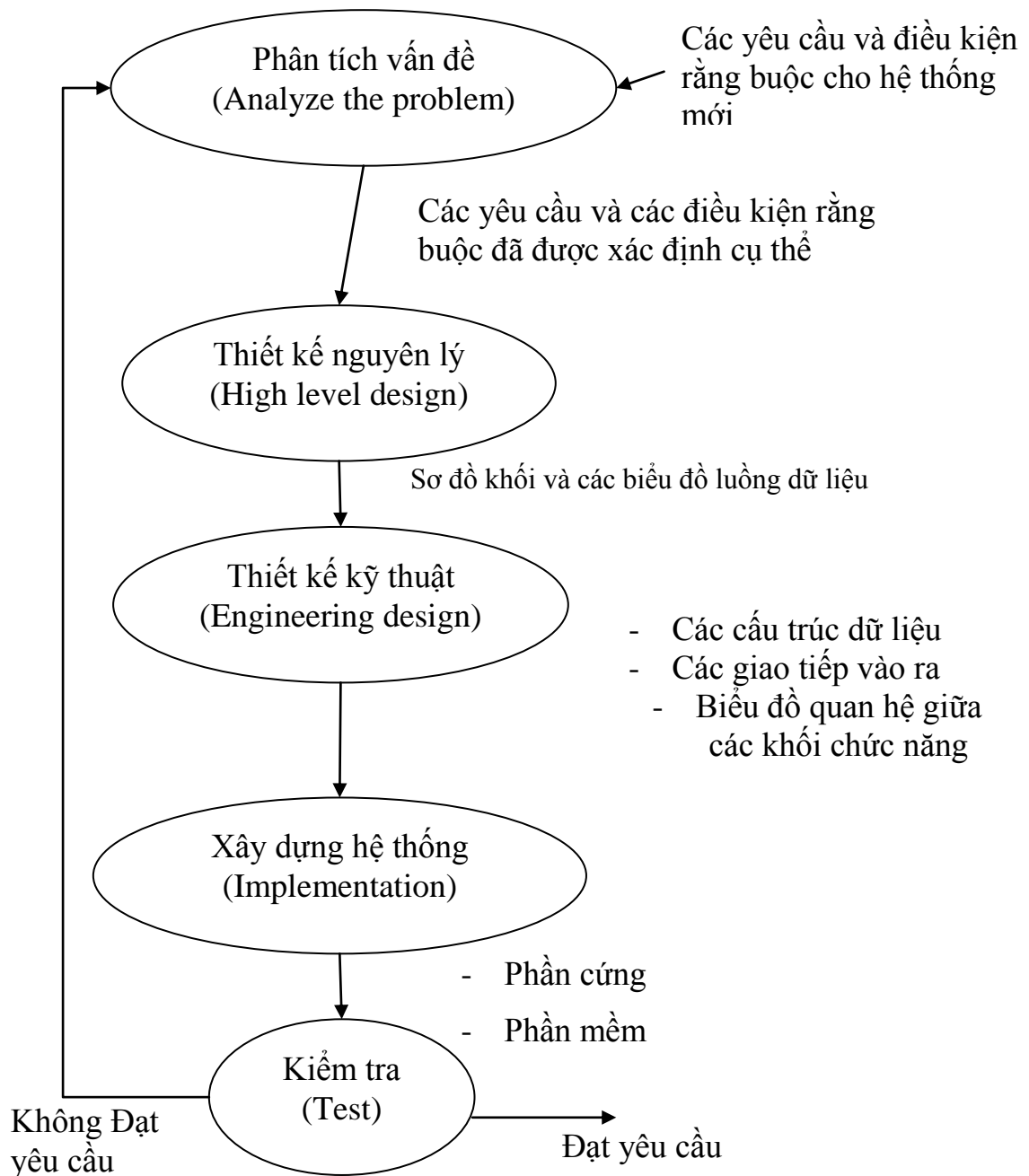
Thiết kế hệ thống có vai trò rất quan trọng. Chất lượng của phần mềm phụ thuộc rất nhiều vào bản thiết kế. Một bản thiết kế tốt còn giúp cho việc thực hiện các giai đoạn khác dễ dàng hơn, giúp cho người thực hiện hoàn thành chính xác hơn công việc của mình. Các quy trình thiết kế thường được sử dụng như: Top-Down, Bottom-Up hoặc kết hợp cả hai quy trình trên.

#### **2.2.2.1. Quy trình Top-down.**

Quy trình này tiếp cận bài toán theo hướng xem xét bài toán từ các khía cạnh chi tiết và sau đó mới tổng quát lên. Quy trình Top-Down thường được áp dụng cho các bài toán đã có giải pháp công nghệ cả về phần mềm cũng như phần cứng. Các giải pháp này đã được phát triển trước đó ở các ứng dụng khác, và đã được kiểm định.

Trong thực tế chúng ta sẽ thấy, bản chất hay mấu chốt của quy trình là vấn đề tìm hiểu và xác định bài toán, làm sao để xác định được chính xác và đầy đủ nhất các yêu cầu cũng các ràng buộc mà hệ thống phải đạt được.

Sơ đồ khối quy trình kế top-down ở hình dưới



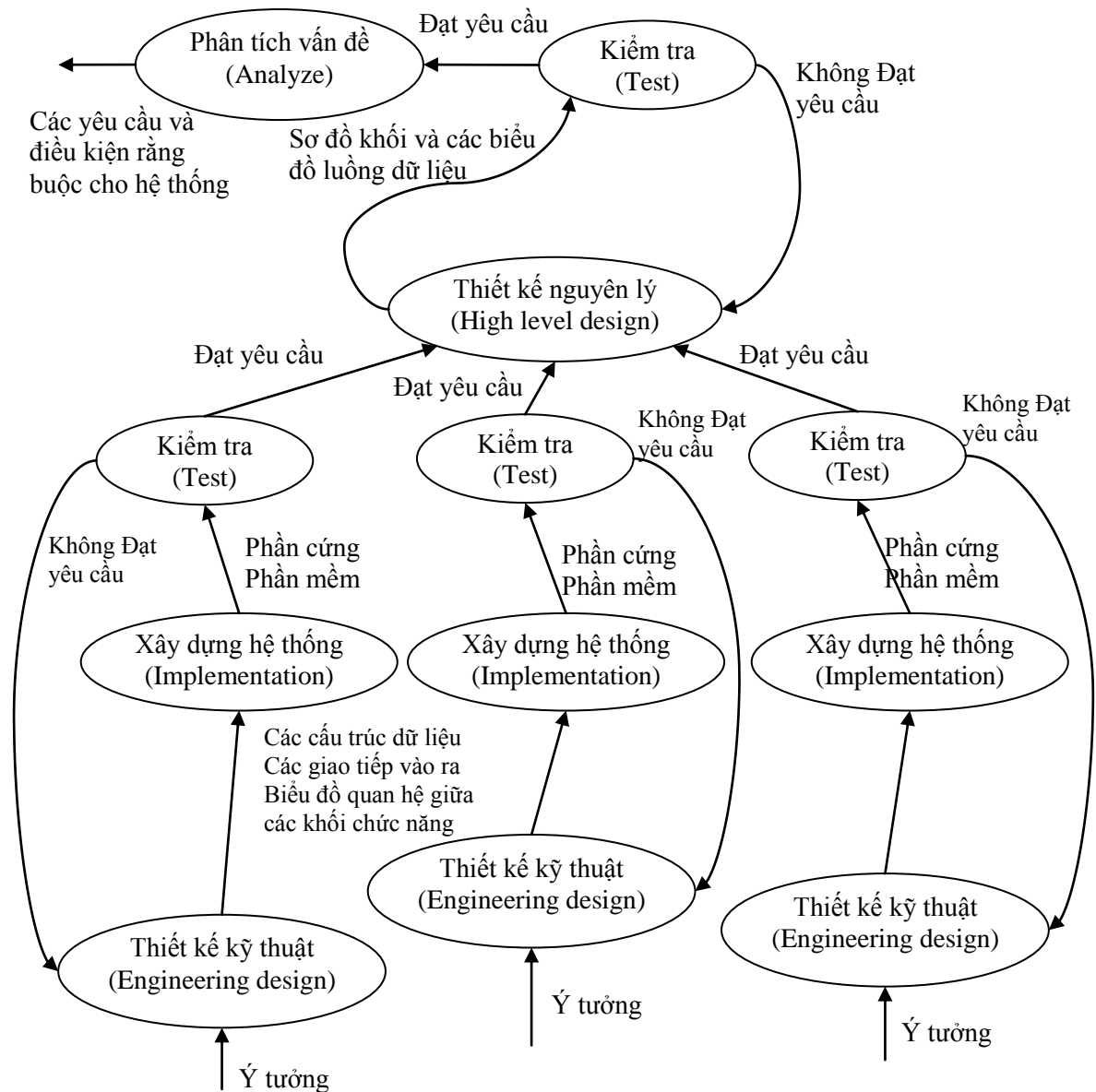
**Hình 2.6:** Sơ đồ khối quy trình Top-Down

### **2.2.2.2. Quy trình Bottom-Up.**

Quy trình Bottom-Up trong thực tế thường áp dụng trong các bài toán chưa lựa chọn hay chưa tìm ra được giải pháp công nghệ. Mấu chốt của quy trình tập trung chủ yếu và quá trình thử nghiệm với hệ thống và tín hiệu thực, từ đó chọn ra giải pháp công nghệ và linh kiện phù hợp nhất cho bài toán. Sơ đồ tổng quát của quy trình như hình bên dưới.

Quy trình Bottom-Up bắt đầu từ các ý tưởng đơn lẻ, sau đó xây dựng luôn thiết kế kỹ thuật. Như ta thấy quy trình hoàn toàn ngược so với Top-Down. Quy trình này thường áp dụng có các bài toán chưa nắm chắc về lời giải, người thiết kế mới chỉ có ý tưởng về một vấn đề nào đó và muốn tìm một giải pháp hoặc giải pháp tốt nhất để giải quyết vấn đề. Việc giải quyết các ý tưởng có thể một hoặc nhiều để có một sản phẩm hoàn chỉnh. Ở quy trình này ta cần chú ý có 2 khâu test nhằm kiểm định chính xác lại các thiết kế kỹ thuật và thiết kế nguyên lý trước khi lựa chọn một giải pháp tối ưu nhất.

Chính từ việc thí nghiệm và thiết kế thử hệ thống trước, sau đó mới có thể phân tích nguyên lý để chọn các đặc tính mới, ràng buộc mới cho một hệ thống mới. Với quy trình này khâu thiết kế kỹ thuật và Test sau khi xây dựng hệ thống là quan trọng nhất. Vì với Top-Down việc xây dựng một sản phẩm là theo nhu cầu của người dùng và môi trường đặt hệ thống. Còn với Bottom-Up có thể người ta còn chưa tìm ra cách để thiết kế ra sản phẩm đó, hoặc sản phẩm đó chưa hề có trên thị trường, khi đó cả người dùng và người thiết kế chưa thể có thông tin gì về các yêu cầu cho sản phẩm hay các đặc tính kỹ thuật của sản phẩm, vì vậy khâu thiết kế kỹ thuật và Test sau thực thi các kỹ sư phải tìm ra các đặc tính đó, nhằm xác định được các ưu việt cũng như các hạn chế của sản phẩm mới.

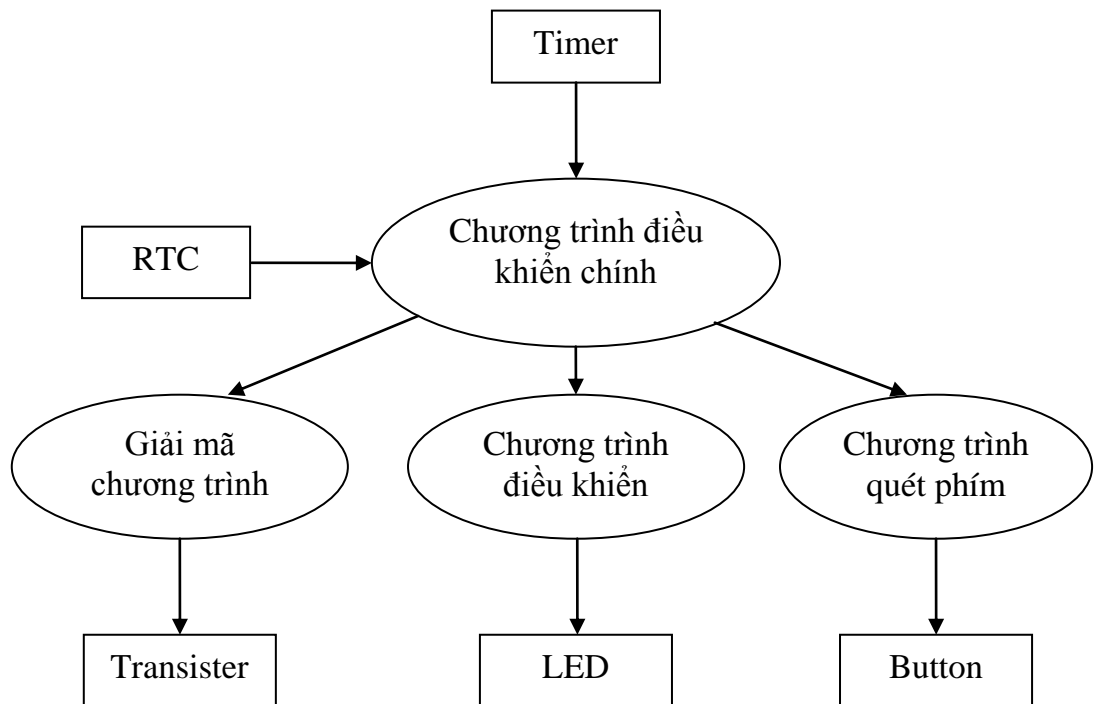


**Hình 2.7:** Sơ đồ khối quy trình Bottom-Up

Trong thực tế có nhiều chương trình được kết hợp cả hai quy trình thiết kế Top-Down và Bottom-Up, phương pháp này tận dụng được các ưu điểm và loại bỏ một số khuyết điểm của cả 2 phương pháp trên.

Qua những phân tích và nhận định ở trên kết hợp với những yêu cầu của đề tài thiết kế chúng em lựa chọn quy trình top-Down để thiết kế cho đề tài này.

### 2.2.3. Sơ đồ Call graph.



*Hình 2.8:* Sơ đồ Call graph

### 2.2.4. Các yêu cầu và giới hạn cho hệ thống

#### 2.2.4.1. Các yêu cầu.

- Giao diện rõ ràng để hiển thị thời gian đang chạy.
- Hiển thị toàn màn hình khi chạy chương trình.
- Tính chính xác cao về thời gian.
- Không mất dữ liệu khi mất điện đột ngột (do sử dụng nguồn dự trữ 3V trong DS1307)

#### 2.2.4.2. Giới hạn cho hệ thống.

- Dùng led 7 thanh để hiển thị thời gian.
- Chỉ hiển thị giờ, phút, giây, thứ, ngày, tháng, năm.
- Kích thước nhỏ gọn cho người sử dụng.
- Có phím điều chỉnh thời gian.

## 2.3. LỰA CHỌN TỔNG QUAN VỀ LINH KIỆN.

Linh kiện trong đề tài gồm :

- \* IC thời gian thực DS1307
- \* VĐK AT89C52
- \* LED 7 đoạn Anode chung
- \* IC giải mã 74HC138
- \* Tụ điện (capacitor)
- \* Điện trở(resistors)
- \* Nút ấn(button)

### 2.3.1. IC thời gian thực DS1307.

#### 2.3.1.1. Giới thiệu chung về DS1307.



**Hình 2.9:** Hình ảnh IC DS1307

IC thời gian thực là họ vi điều khiển của hãng Dalat. DS1307 có một số đặc trưng cơ bản sau:

- + DS1307 là IC thời gian thực với nguồn cung cấp nhỏ dùng để cập nhật thời gian và ngày tháng
- + SRAM: 56bytes
- + Địa chỉ và dữ liệu được truyền nối tiếp qua 2 đường bus 2 chiều
- + DS1307 có một mạch cảm biến điện áp dùng để dò các điện áp lỗi và tự động đóng ngắt với nguồn pin cung cấp 3V:

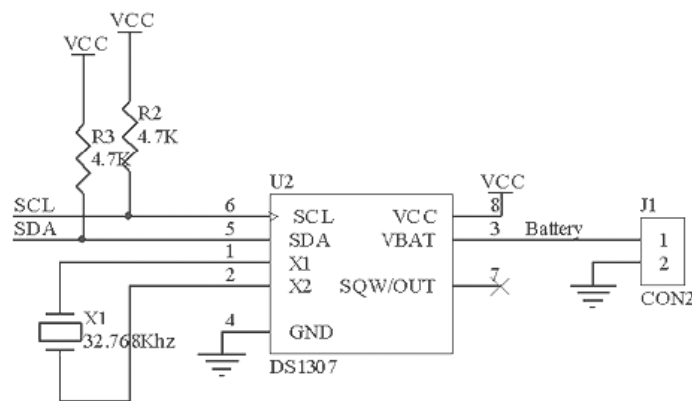
+ DS1307 có 7 byte dữ liệu nằm từ địa chỉ 0x00 tới 0x06, 1 byte điều khiển, và 56 byte lưu trữ ( dành cho người sử dụng ).

+ Khi xử lý dữ liệu từ DS1307, họ đã tự chuyển cho ta về dạng số BCD, ví dụ như ta đọc được dữ liệu từ địa chỉ 0x04 ( tương ứng với ngày trong tháng ) và tại 0x05 ( tháng ) là 0x15, 0x11.

+ Lưu ý đến vai trò của chân SQW/OUT. Đây là chân cho xung ra của DS1307 có 4 chế độ 1Hz, 4.096HZ, 8.192Hz, 32.768Hz... các chế độ này được quy định bởi các bit của thanh ghi Control Register ( địa chỉ 0x07 ).

+ Địa chỉ của DS1307 là 0xD0.

### 2.3.1.2. Cơ chế hoạt động và chức năng của DS1307.



**Hình 2.10** Sơ đồ chân DS1307

Vcc: nối với nguồn

X1,X2: nối với thạch anh 32,768 kHz

Vbat: đầu vào pin 3V

GND: đất

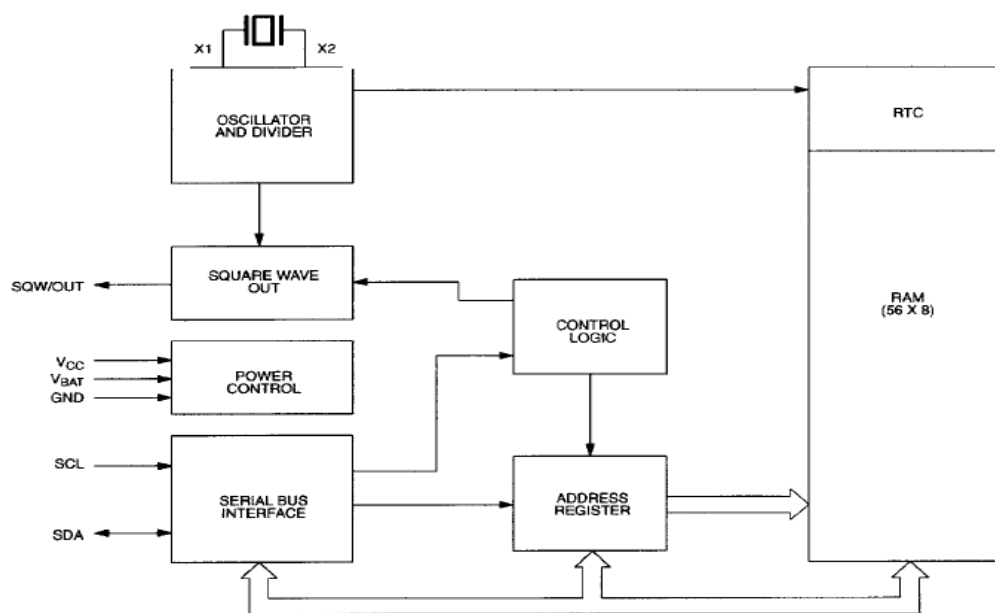
SDA: chuỗi data

SCL: dây xung clock

SQW/OUT: xung vuông/đầu ra driver

- DS1307 là một IC thời gian thực với nguồn cung cấp nhỏ, dùng để cập nhật thời gian và ngày tháng với 56 bytes SRAM. Địa chỉ và dữ liệu được truyền nối tiếp qua 2 đường bus 2 chiều. Nó cung cấp thông tin về giờ, phút, giây, thứ, ngày, tháng, năm. Ngày cuối tháng sẽ tự động được điều chỉnh với các tháng nhỏ hơn 31 ngày, bao gồm cả việc tự động nhảy năm. Đồng hồ có thể hoạt động ở dạng 24h hoặc 12h với chỉ thị AM/PM. DS1307 có một mạch cảm biến điện áp dùng để dò các điện áp lỗi và tự động đóng ngắt với nguồn pin cung cấp.

- DS1307 hoạt động với vai trò slave trên đường bus nối tiếp. Việc truy cập được thi hành với chỉ thị START và một mã thiết bị nhất định được cung cấp bởi địa chỉ các thanh ghi. Tiếp theo đó các thanh ghi sẽ được truy cập liên tục đến khi chỉ thị STOP được thực thi.



**Hình 2.11** : Sơ đồ khối của DS1307

### 2.3.1.3. Mô tả hoạt động của các chân.

- Vcc, GND: nguồn một chiều được cung cấp tới các chân này. Vcc là đầu vào 5V. Khi 5V được cung cấp thì thiết bị có thể truy cập hoàn chỉnh và dữ liệu có thể đọc và viết. Khi pin 3 V được nối tới thiết bị này và Vcc nhỏ hơn 1,25Vbat thì quá trình đọc và viết không được thực thi, tuy nhiên chức



năng timekeeping không bị ảnh hưởng bởi điện áp vào thấp. Khi Vcc nhỏ hơn Vbat thì RAM và timekeeper sẽ được ngắt tới nguồn cung cấp trong (thường là nguồn 1 chiều 3V).

- Vbat: Đầu vào pin cho bất kỳ một chuẩn pin 3V. Điện áp pin phải được giữ trong khoảng từ 2,5 - 3V để đảm bảo cho sự hoạt động của thiết bị.

- SCL(serial clock input): SCL được sử dụng để đồng bộ sự chuyển dữ liệu trên đường dây nối tiếp.

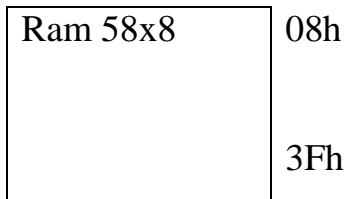
- SDA(serial data input/output): Là chân vào ra cho 2 đường dây nối tiếp. Chân SDA thiết kế theo kiểu cực máng hở, đòi hỏi phải có một điện trở kéo trong khi hoạt động.

- SQW/OUT(square wave/output driver)- khi được kích hoạt thì bit SQWE được thiết lập 1 chân SQW/OUT phát đi 1 trong 4 tần số (1Hz,4kHz,8kHz,32kHz). Chân này cũng được thiết kế theo kiểu cực máng hở vì vậy nó cũng cần có một điện trở kéo trong. Chân này sẽ hoạt động khi cả Vcc và Vbat được cấp.

- X1, X2: được nối với một thạch anh tần số 32,768kHz. Là một mạch tạo dao động ngoài, để hoạt động ổn định thì phải nối thêm 2 tụ 33pF. Cũng có DS1307 với bộ tạo dao động trong tần số 32,768kHz, với cấu hình này thì chân X1 sẽ được nối vào tín hiệu dao động trong còn chân X2 thì để hở.

a) Sơ đồ địa chỉ RAM và RTC:

Seconds	00h
Minutes	
Hours	
Day	
Date	
Month	
Year	
Control	07h



- Thông tin về thời gian và ngày tháng được lấy ra bằng cách đọc các byte thanh ghi thích hợp. thời gian và ngày tháng được thiết lập cũng thông qua các byte thanh ghi này bằng cách viết vào đó những giá trị thích hợp. Nội dung của các thanh ghi dưới dạng mã BCD(binary coded decreaseimal). Bit 7 của thanh ghi seconds là bit clock halt( CH ), khi bit này được thiết lập 1 thì dao động disable, khi nó được xoá về 0 thì dao động được enable.

**Bảng 2.1:** Thanh ghi thời gian thực

BIT7								BIT 0
00H	CH	10 SECENDS			SECENDS			00-59
	X	10MINUTES			MINUTES			00-59
	X	12	10HR	10H R	HOURS			01-12
		24	A/P					00-23
	X	X	X	X	X	DAY		1-7
	X	X	10 DATE		DAY			01-28/29 01-30 01-31
	X	X	X	10 MONTH	MONTH			01-12
		10YEAR			YEAR			00-59
07H	OUT	X	X	SOWE	X	X	RS1	R S0

- DS1307 có thể chạy ở chế độ 24h cũng như 12h. Bit thứ 6 của thanh ghi hours là bit chọn chế độ 24h hoặc 12h. khi bit này ở mức cao thì chế độ 12h được chọn. ở chế độ 12h thì bit 5 là bit AM/PM với mức cao là là PM. ở chế độ 24h thì bit 5 là bit chỉ 20h (từ 20h đến 23h).

- Trong quá trình truy cập dữ liệu, khi chỉ thị START được thực thi thì dòng thời gian được truyền tới một thanh ghi thứ 2, thông tin thời gian sẽ được đọc từ thanh ghi thứ cấp này, trong khi đó đồng hồ vẫn tiếp tục chạy. Trong DS1307 có một thanh ghi điều khiển để điều khiển hoạt động của chân SQW/OUT :

Bit 7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OUT	X	X	SQWE	X	X	RS1	RS0

- OUT(output control): bit này điều khiển mức ra của chân SQW/OUT khi đầu ra xung vuông là disable. Nếu SQWE = 0 thì mức logic ở chân SQW/OUT sẽ là 1 nếu OUT=1 và OUT = 0 nếu OUT = 0 .

- SQWE(square wave enable): bit này được thiết lập 1 sẽ enable đầu ra của bộ tạo dao động. Tần số của đầu ra sóng vuông phụ thuộc vào giá trị của RS1 và RS0.

**Bảng 2.2 : Tần số DS1307**

RS1	RS0	Tần số đầu ra SQW
0	0	1 Hz
0	1	4.096 kHz
1	0	8.192kHz
1	1	32.768 kHz

DS1307 hỗ trợ bus 2 dây 2 chiều và giao thức truyền dữ liệu. thiết bị gửi dữ liệu lên bus được gọi là bộ phát và thiết bị nhận gọi là bộ thu. thiết bị điều khiển quá trình này gọi là master. Thiết bị nhận sự điều khiển của master gọi

là slave. Các bus nhận sự điều khiển của master, là thiết bị phát ra chuỗi xung clock(SCL), master sẽ điều khiển sự truy cập bus, tạo ra các chỉ thị START và STOP.

b) Sự truyền nhận dữ liệu trên chuỗi bus 2 dây :

Tùy thuộc vào bit R/ w mà 2 loại truyền dữ liệu sẽ được thực thi:

- Truyền dữ liệu từ master truyền và slave nhận: Master sẽ truyền byte đầu tiên là địa chỉ của slave. Tiếp sau đó là các byte dữ liệu. Slave sẽ gửi lại bit thông báo đã nhận được (bit acknowledge) sau mỗi byte dữ liệu nhận được. dữ liệu sẽ truyền từ bit có giá trị nhất (MSB).

- Truyền dữ liệu từ slave và master nhận: byte đầu tiên (địa chỉ của slave) được truyền tới slave bởi master. Sau đó slave sẽ gửi lại master bit acknowledge. Tiếp theo đó slave sẽ gửi các byte dữ liệu tới master. Master sẽ gửi cho slave các bit acknowledge sau mỗi byte nhận được trừ byte cuối cùng, sau khi nhận được byte cuối cùng thì bit acknowledge sẽ không được gửi .

Master phát ra tất cả các chuỗi xung clock và các chỉ thị START và STOP. sự truyền sẽ kết thúc với chỉ thị STOP hoặc chỉ thị quay vòng START. Khi chỉ thị START quay vòng thì sự truyền chuỗi dữ liệu tiếp theo được thực thi và các bus vẫn chưa được giải phóng. Dữ liệu truyền luôn bắt đầu bằng bit MSB.

c) DS1307 có thể hoạt động ở 2 chế độ sau:

- Chế độ slave nhận( chế độ DS1307 ghi): Chuỗi dữ liệu và chuỗi xung clock sẽ được nhận thông qua SDA và SCL. Sau mỗi byte được nhận thì 1 bit acknowledge sẽ được truyền. các điều kiện START và STOP sẽ được nhận dạng khi bắt đầu và kết thúc một truyền 1 chuỗi. Nhận dạng địa chỉ được thực hiện bởi phần cứng sau khi chấp nhận địa chỉ của slave và bit chiều. Byte địa chỉ là byte đầu tiên nhận được sau khi điều kiện START được phát ra từ master. Byte địa chỉ có chứa 7 bit địa chỉ của DS1307, là 1101000, tiếp theo đó là bit chiều (R/ w) cho phép ghi khi nó bằng 0. Sau khi nhận và giải mã

byte địa chỉ thì thiết bị sẽ phát đi 1 tín hiệu acknowledge lên đường SDA. Sau khi DS1307 nhận dạng được địa chỉ và bit ghi thì master sẽ gửi một địa chỉ thanh ghi tới DS1307, tạo ra một con trỏ thanh ghi trên DS1307 và master sẽ truyền từng byte dữ liệu cho DS1307 sau mỗi bit acknowledge nhận được. Sau đó master sẽ truyền điều kiện STOP khi việc ghi hoàn thành.

	Slave Address	R/W		Word Address (n)		Data (n)		Data (n+1)		Data (n+X)		
S	1101000	0	A	XXXXXXXX	A	XXXXXXXX	A	XXXXXXXX	A	XXXXXXXX	A	P

**S-START**

**A-ACKNOWLEDGE**

**DATA TRANSFERRED(X+1 BYTE + ACKNOWLEDGE)**

**P-STOP**

**R/W-READ/WRITE OR DIRECTION BIT**

**ADDRESS = D0h**

- Chế độ slave phát ( chế độ DS1307 đọc): byte đầu tiên slave nhận được tương tự như chế độ slave ghi. Tuy nhiên trong chế độ này thì bit chiều lại chỉ chiều truyền ngược lại. Chuỗi dữ liệu được phát đi trên SDA bởi DS 1307 trong khi chuỗi xung clock vào chân SCL. Các điều kiện START và STOP được nhận dạng khi bắt đầu hoặc kết thúc truyền một chuỗi. Byte địa chỉ nhận được đầu tiên khi master phát đi điều kiện START. Byte địa chỉ chứa 7 bit địa chỉ của slave và 1 bit chiều cho phép đọc là 1. Sau khi nhận và giải mã byte địa chỉ thì thiết bị sẽ nhận 1 bit acknowledge trên đường SDA. Sau đó DS1307 bắt đầu gửi dữ liệu tới địa chỉ con trỏ thanh ghi thông qua con trỏ thanh ghi. Nếu con trỏ thanh ghi không được viết vào trước khi chế độ đọc được thiết lập thì địa chỉ đầu tiên được đọc sẽ là địa chỉ cuối cùng chứa trong con trỏ thanh ghi. DS1307 sẽ nhận được một tín hiệu Not Acknowledge khi kết thúc quá trình đọc. Đọc dữ liệu-chế độ slave phát.

	Slave Address	R/W		Data (n)		Data (n+1)		Data (n+2)		Data (n+X)		
S	1101000	0	A	XXXXXXXX	A	XXXXXXXX	A	XXXXXXXX	A	XXXXXXXX	A	P

S-START

A-ACKNOWLEDGE

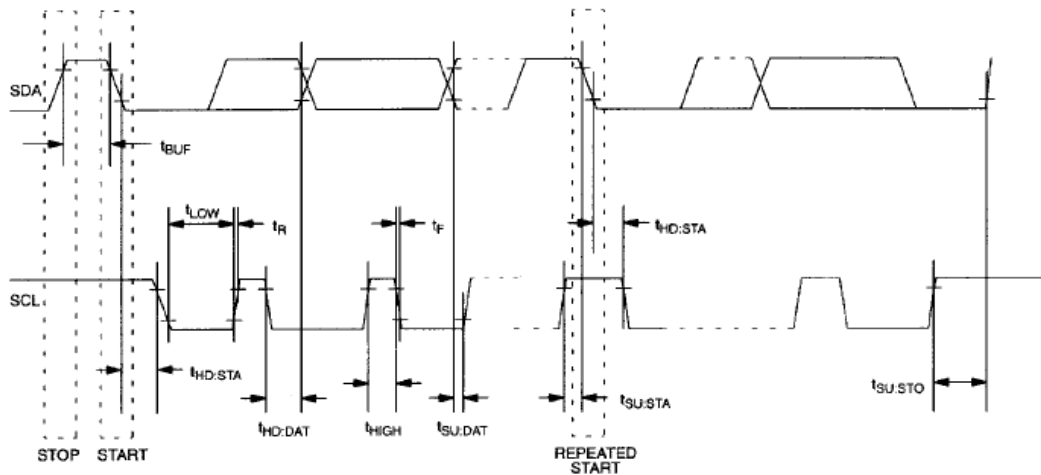
DATA TRANSFERRED(X+1 BYTE + ACKNOWLEDGE)

P-STOP

R/W-READ/WRITE OR DIRECTION BIT

ADDRESS = D0h

- Thời gian thực hiện việc đọc, ghi dữ liệu của DS1307: sơ đồ đồng bộ:



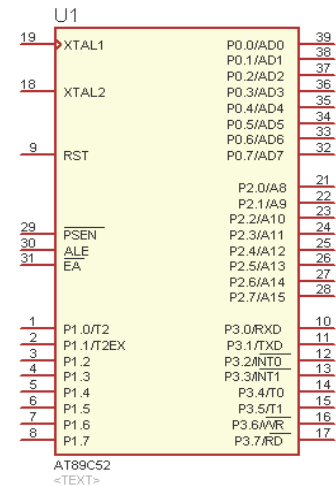
**Hình 2.12** : Sơ đồ đồng bộ đọc ghi dữ liệu của DS1307

(0°C to 70 °C or -40 °C to +85 °C; V<sub>cc</sub>=4.5V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTE
SCL Clock Frequency	f <sub>SCL</sub>	0		100	kHz	
Bus free Time Between a STOP and START Condition	t <sub>BUF</sub>	4.7			μs	
Hold Time (Repeated) START Condition	t <sub>HD:STA</sub>	4.0			μs	5
LOW Period of SCL Clock	t <sub>LOW</sub>	4.7			μs	
HIGH Period of SCL Clock	t <sub>HIGH</sub>	4.0			μs	
Set-up Time for a Repeated START Condition	t <sub>SU:STA</sub>	4.7			μs	
Data Hold Time	t <sub>HD:DAT</sub>	0			ns	6.7
Data Set-up Time	t <sub>SU:DAT</sub>	250			ns	
Rise Time of Both SDA and SCL Signals	t <sub>R</sub>			1000	ns	
Fall Time of Both SDA and SCL Signals	t <sub>F</sub>			300	ns	
Set-up Time for STOP Condition	t <sub>SU:STO</sub>				ns	

Capacitive Load for each Bus Line	$C_B$			400	pF	8
I/O Capacitance	$C_{IO}$		10		pF	
Crystal Specified Load Capacitance			12.5		pF	

### 2.3.2. Vi Điều Khiển AT89C52



**Hình 2.13:** Hình ảnh và sơ đồ chân AT9C52.

#### 2.3.2.1. Cấu tạo và chức năng các khối của AT89C52.

- CPU( *CPU centralprocessing unit*) bao gồm:
  - Thanh ghi tích lũy A
  - Thanh ghi tích lũy phụ B
  - Đơn vị logic học (ALU)
  - Thanh ghi từ trạng thái chương trình
  - Bốn băng thanh ghi
  - Con trỏ ngăn xếp
- Bộ nhớ chương trình( ROM) gồm 8Kbyte Flash.
- Bộ nhớ dữ liệu( RAM) gồm 256 byte.
- Bộ UART, có chức năng truyền nhận nối tiếp.
- 3 bộ Timer/Counter 16 bit thực hiện chức năng định thời và đếm sự kiện.
- Khối điều khiển ngắt với 2 nguồn ngắt ngoài và 4 nguồn ngắt trong.

- Bộ lập trình ( ghi chương trình lên Flash ROM) cho phép người sử dụng có thể nạp các chương trình cho chip mà không cần các bộ nạp chuyên dụng.
- Bộ chia tần số với hệ số chia là 12.
- 4 cổng xuất nhập với 32 chân.

### 2.3.2.2. Chức năng các chân của AT89C52.

#### ❖ Port 0( P0.0-P0.7)

Port 0 gồm 8 chân, ngoài chức năng xuất nhập, port 0 còn là bus đa hợp dữ liệu và địa chỉ ( AD0-AD7), chức năng này sẽ được sử dụng khi 89c52 giao tiếp với các thiết bị ngoài có kiến trúc Bus như các vi mạch nhớ, mạch PIO...

#### ❖ Port 1( P1.0-P1.7)

Chức năng duy nhất của Port 1 là chức năng xuất nhập cũng như các Port khác. Port 1 có thể xuất nhập theo bit và theo byte.

#### ❖ Port 2( P2.0=>P2.7)

Port 2 ngoài chức năng là cổng vào/ra như Port 0 và 1 còn là byte cao của bus địa chỉ khi sử dụng bộ nhớ ngoài.

#### ❖ Port 3(P3.0=>P3.7)

Mỗi chân trên Port 3 ngoài chức năng xuất nhập còn có một chức năng riêng, cụ thể như sau:

**Bảng 2.3 : Chức năng chân port 3**

Bit	Tên	Chức năng
P3.0	RXD	Dữ liệu nhận cho Port nối tiếp
P3.1	TXD	Dữ liệu truyền cho Port nối tiếp
P3.2	INT0	Ngắt bên ngoài 0
P3.3	INT1	Ngắt ngoài 1
P3.4	TO	Ngõ vào của Timer/counter0
P3.5	T1	Ngõ vào của Timer/counter1
P3.6	/WR	Xung ghi bộ nhớ dữ liệu ngoài.
P3.7	/RD	Xung đọc bộ nhớ dữ liệu ngoài.



❖ Chân /PSEN : là chân điều khiển đọc chương trình ở bộ nhớ ngoài.

❖ Chân ALE.

ALE là tín hiệu điều khiển chốt địa chỉ có tần số bằng 1/6 tần số dao động của vi điều khiển. Tín hiệu ALE được dùng để cho phép vi mạch chốt bên ngoài như 7473.

❖ Chân /EA.

Tín hiệu /EA cho phép chọn bộ nhớ chương trình là bộ nhớ trong hay ngoài. EA=1 thì thực hiện chương trình trong RAM nội. EA=0 thực hiện ở RAM ngoài.

❖ RST( reset)

Ngõ vào reset trên chân số 9. Khi RST=1 thì bộ vi điều khiển sẽ được khởi động lại thiết lập ban đầu.

❖ XTAL1, XTAL2

2 chân này được nối song song với thạch anh tần số max=33 Mhz. Để tạo dao động cho bộ vi điều khiển.

❖ Vcc, GND : cung cấp nguồn nuôi cho bộ vi điều khiển. Cấp qua chân 20 và 40.

### **2.3.3. Led 7 thanh.**

#### **2.3.3.1. Các khái niệm cơ bản.**

Trong các thiết bị, để báo trạng thái hoạt động của thiết bị đó cho người sử dụng với thông số chỉ là các dãy số đơn thuần, thường người ta sử dụng led 7 thanh. Led 7 thanh được sử dụng khi các dãy số không đòi hỏi quá phức tạp, chỉ cần hiển thị số là đủ, chẳng hạn led 7 thanh được dùng để hiển thị nhiệt độ phòng, trong các đồng hồ treo tường bằng điện tử, hiển thị số lượng sản phẩm được kiểm tra sau một công đoạn nào đó...

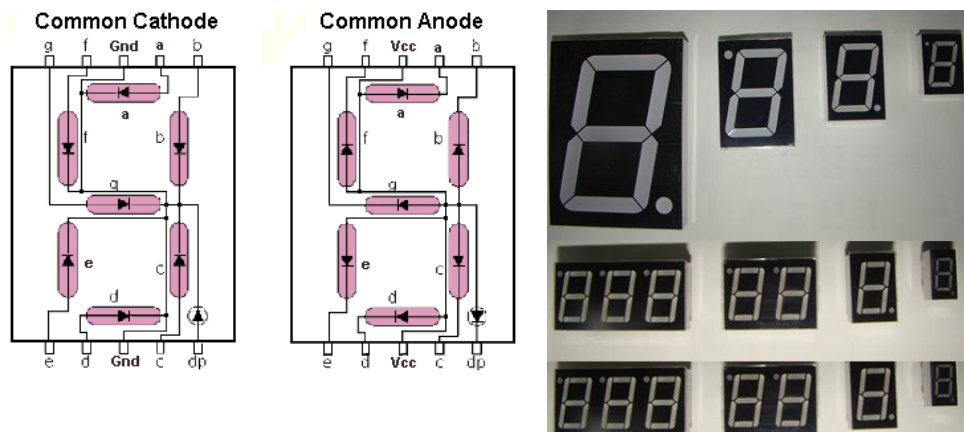
Led 7 thanh có cấu tạo bao gồm 7 led đơn có dạng thanh xếp theo hình và có thêm. Một led đơn hình tròn nhỏ thể hiện dấu chấm tròn ở góc dưới, bên phải của led 7 thanh. 8 chân trên led 7 thanh có Anode (cực +) hoặc Cathode (cực -) được nối chung với nhau vào một điểm, được đưa chân ra ngoài để kết

nối với mạch điện. 8 cực còn lại trên mỗi led đơn được đưa thành 8 chân riêng, cũng được đưa ra ngoài để kết nối với mạch điện. Nếu led 7 thanh có Anode (cực +) chung, đầu chung này được nối với +Vcc, các chân còn lại dùng để điều khiển trạng thái sáng tắt của các led đơn, led chỉ sáng khi tín hiệu đặt vào các chân này ở mức 0. Nếu led 7 thanh có Cathode (cực -) chung, đầu chung này được nối xuống Ground (hay Mass), các chân còn lại dùng để điều khiển trạng thái sáng tắt của các led đơn, led chỉ sáng khi tín hiệu đặt vào các chân này ở mức 1.

Vì led 7 thanh chứa bên trong nó các led đơn, do đó khi kết nối cần đảm bảo dòng qua mỗi led đơn trong khoảng 10mA-20mA để bảo vệ led. Nếu kết nối với nguồn 5V có thể hạn dòng bằng điện trở 330Ω trước các chân nhận tín hiệu điều khiển.

### 2.3.3.2. Sơ đồ vị trí các led

Các điện trở 330Ω là các điện trở bên ngoài được kết nối để giới hạn dòng điện qua led nếu led 7 đoạn được nối với nguồn 5V.



**Hình 2.14:**

*Sơ đồ chân 7 thanh cathode, anode chung và hình ảnh minh họa.*

### 2.3.3.3. Kết nối với Vi điều khiển

Ngõ nhận tín hiệu điều khiển của led 7 đoạn có 8 đường, vì vậy có thể dùng 1 Port nào đó của Vi điều khiển để điều khiển led 7 đoạn. Như vậy led 7 đoạn nhận một dữ liệu 8 bit từ Vi điều khiển để điều khiển hoạt động sáng tắt của từng led đơn trong nó, dữ liệu được xuất ra điều khiển led 7 đoạn thường được gọi là "mã hiển thị led 7 đoạn". Có hai kiểu mã hiển thị led 7 đoạn: mã dành cho led 7 đoạn có Anode(cực +) chung và mã dành cho led 7 đoạn có Cathode (cực -) chung. Chẳng hạn, để hiển thị số 1 cần làm cho các led ở vị trí b và c sáng, nếu sử dụng led 7 đoạn có Anode chung thì phải đặt vào hai chân b và c điện áp là 0V(mức 0) các chân còn lại được đặt điện áp là 5V(mức 1), nếu sử dụng led 7 đoạn có Cathode chung thì điện áp(hay mức logic) hoàn toàn ngược lại, tức là phải đặt vào chân b và c điện áp là 5V(mức 1).Phần cứng được kết nối với 1 Port bất kì của Vi điều khiển, để thuận tiện cho việc xử lí về sau phần cứng nên được kết nối như sau: Px.0 nối với chân a, Px.1 nối với chân b, lần lượt theo thứ tự cho đến Px.7 nối với chân h.

### 2.3.3.4. Bảng mã của Led Anode chung.

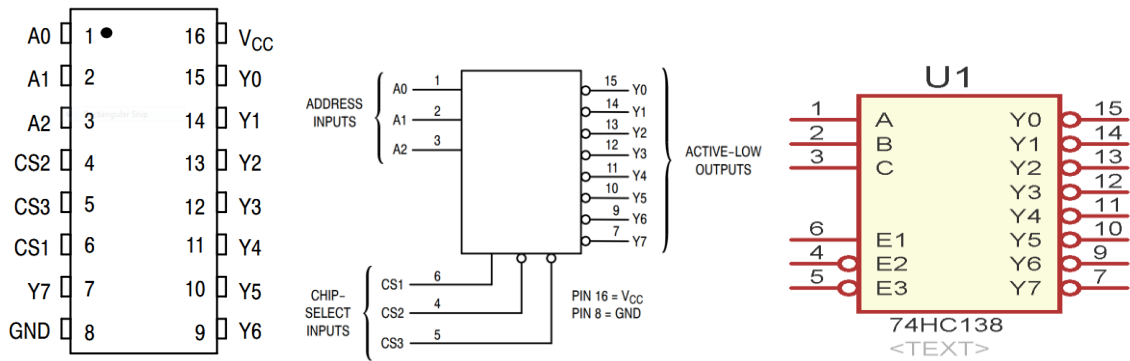
**Bảng 2.4:** Bảng mã cho Led Anode chung (a là MSB, dp là LSB):

Số	a	b	c	d	e	f	g	dp	Mã Hex
0	0	0	0	0	0	0	1	1	03h
1	1	0	0	1	1	1	1	1	9Fh
2	0	0	1	0	0	1	0	1	25h
3	0	0	0	0	1	1	0	1	0Dh
4	1	0	0	1	1	0	0	1	99h
5	0	1	0	0	1	0	0	1	49h
6	0	1	0	0	0	0	0	1	41h
7	0	0	0	1	1	1	1	1	1Fh
8	0	0	0	0	0	0	0	1	01h
9	0	0	0	0	1	0	0	1	09h

**Bảng 2.5:** Bảng mã cho Led Anode chung (a là LSB, dp là MSB):

Số	dp	g	f	e	d	c	b	a	Mã Hex
0	1	1	0	0	0	0	0	0	C0h
1	1	1	1	1	1	0	0	1	F9h
2	1	0	1	0	0	1	0	0	A4h
3	1	0	1	1	0	0	0	0	B0h
4	1	0	0	1	1	0	0	1	99h
5	1	0	0	1	0	0	1	0	92h
6	1	0	0	0	0	0	1	0	82h
7	1	1	1	1	1	0	0	0	F8h
8	1	0	0	0	0	0	0	0	80h
9	1	0	0	1	0	0	0	0	90h

### 2.3.4. IC giải mã 74HC138.



**Hình 2.15:** Sơ đồ chân của 74HC138

IC 74HC138 là bộ giả mã địa chỉ với 3 đầu vào ( A0,A1,A2) và 8 đầu ra phủ định (Y0 đến Y7 ). Nó có 3 đầu vào cho phép: 2 đầu vào tích cực thấp (1E,2E) và một đầu vào tích cực mức cao (E3). Tất cả các đầu ra của 74HC138 sẽ ở mức cao trừ khi E1 ở mức thấp và E3 ở mức cao. Khi các đầu vào 1E, 2E ở mức thấp và E3 ở mức cao thì đầu ra của 74HC138 sẽ được quyết định bởi đầu vào .

**Bảng 2.6:** Bảng chức năng của 74HC138

Inputs						Outputs							
CS1	CS2	CS3	A2	A1	A0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	X	H	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	H	H	H	H	H	L	H	H	H
H	L	L	H	H	L	H	H	H	H	H	L	H	H
H	L	L	H	H	H	H	H	H	H	H	H	L	H

H : Mức cao.      L : Mức thấp.      X : Không làm việc.

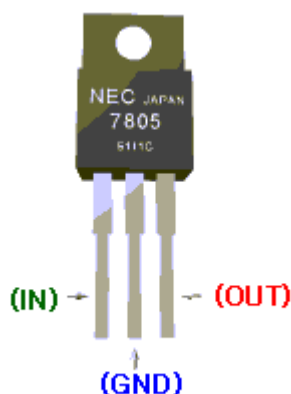
- Đặc điểm:

- + Số drive đầu ra: 10 LSTTL Loads
- + Các giao diện đầu ra tới CMOS, NMOS and TTL
- + Dải điện áp hoạt động : 2.0 to 6.0 V
- + Dòng điện vào thấp: 1.0  $\mu$ A
- + Khả năng loại trừ tạp âm cao trong CMOS Devices
- + Phù hợp với những yêu cầu đã được định nghĩa bởi JEDEC theo chuẩn 7A
- + Đặc trưng ESD : HBM >2000 V; machine Model > 200 V
- + Độ phức tạp của chip: 100 FETs or 29 cổng tương đương

## 2.4 LINH KIỆN KHÁC.

### 2.4.1 IC nguồn 7805.

Với những mạch điện không đòi hỏi độ ổn định của điện áp quá cao, sử dụng IC ổn áp thường được người thiết kế sử dụng vì mạch điện khá đơn giản. Các loại ổn áp thường được sử dụng là IC 78xx, với xx là điện áp cần ổn áp. Ví dụ 7805 ổn áp 5V, 7812 ổn áp 12V. Việc dùng các loại IC ổn áp 78xx tương tự nhau, dưới đây là minh họa cho IC ổn áp 7805.



**Hình 2.16:** Hình ảnh IC nguồn 7805

Sơ đồ phía dưới IC 7805 có 3 chân:

- \* Chân số 1 là chân IN.
- \* Chân số 2 là chân GND.
- \* Chân số 3 là chân OUT.

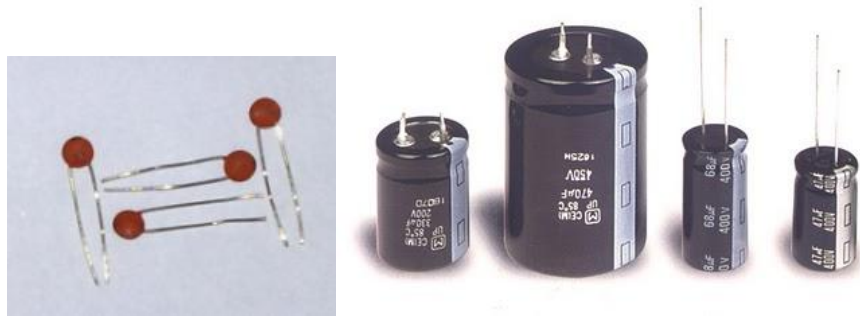
Ngõ ra OUT luôn ổn định ở 5V dù điện áp từ nguồn cung cấp thay đổi. Mạch này dùng để bảo vệ những mạch điện chỉ hoạt động ở điện áp 5V (các loại IC thường hoạt động ở điện áp này). Nếu nguồn điện có sự cố đột ngột: điện áp tăng cao thì mạch điện vẫn hoạt động ổn định nhờ có IC 7805 vẫn giữ được điện áp ở ngõ ra OUT 5V không đổi.

Mạch trên lấy nguồn một chiều từ một máy biến áp với điện áp từ 7V đến 9V để đưa vào ngõ IN. Khi kết nối mạch điện, do nhiều nguyên nhân, người dùng dễ nhầm lẫn cực tính của nguồn cung cấp khi đấu nối vào mạch, trong trường hợp này rất dễ ảnh hưởng đến các linh kiện trên board mạch. Vì lí do đó một diode được lắp thêm vào mạch, diode đảm bảo cực tính của nguồn cấp cho mạch theo một chiều duy nhất, và người dùng cũng không cần quan tâm đến cực tính của nguồn khi nối vào ngõ IN nữa.

#### **2.4.4 Tụ điện.**

Tụ điện là linh kiện điện tử thụ động được sử dụng rất rộng rãi trong các mạch điện tử, chúng được sử dụng trong các mạch lọc nguồn, lọc nhiễu,

mạch truyền tín hiệu xoay chiều, mạch tạo dao động. Trong mạch này ta sẽ sử dụng một tụ hóa cho khối reset, và hai tụ gốm cho khối dao động.



*Tụ gốm*

*Tụ hóa*

**Hình 2.17:** Hình ảnh tụ điện

### 2.4.3. Điện trở.

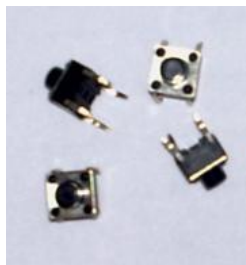
Trong thiết bị điện tử điện trở là một linh kiện quan trọng, chúng được làm từ hợp chất cacbon và kim loại tùy theo tỷ lệ pha trộn mà người ta tạo ra được các loại điện trở có trị số khác nhau.



**Hình 2.18:** Hình ảnh điện trở

### 2.4.4. Nút bấm button.

Trong mạch này ta sử dụng 3 nút bấm để điều khiển menu, tăng, giảm cho việc cài đặt giờ, ngày, tháng, năm.



**Hình 2.19:** Nút bấm

### 2.4.5. Pin CMOS 3V.

Ta sẽ sử dụng một ổ và pin CMOS 3V để làm nguồn nuôi cho DS1307 để nó có thể lưu được giờ khi mất điện nguồn cung cấp cho mạch.



**Hình 2.20:** Pin CMOS

### 2.4.6. Thạch anh.

Trong mạch ta sẽ sử dụng hai thạch anh, một loại 12Mhz để tạo dao động cho AT89C51, một loại 32,768 Mhz để tạo dao động cho DS1307.



Thạch anh 12 Mhz



Thạch anh 32,768 Mhz

**Hình 2.21:** Thạch anh

### Kết luận

RTC sử dụng IC chuyên dụng cho ta độ chính xác tuyệt đối về thời gian thực mà không cần lo lắng đến việc có sai số như khi thực hiện RTC bằng Timer. Đây là ưu điểm vượt trội của RTC thời gian thực và dựa trên đó, ta có thể phát triển rộng rãi hơn các ứng dụng cho phép sử dụng được trong thực tế.



Định hướng phát triển cho các phiên bản sau này:

- Kết hợp với cảm biến nhiệt độ để đo nhiệt độ ngoài trời
- Thay đổi phương pháp hiển thị sao cho đẹp và có mỹ thuật hơn (sử dụng ma trận LED).
- Dựa vào ngày tháng năm dương lịch để tính toán ngày tháng năm âm lịch và hiển thị.
- Trang bị thêm chức năng hẹn giờ.
- Hạ giá thành sản phẩm (sử dụng vi điều khiển PIC rẻ hơn).

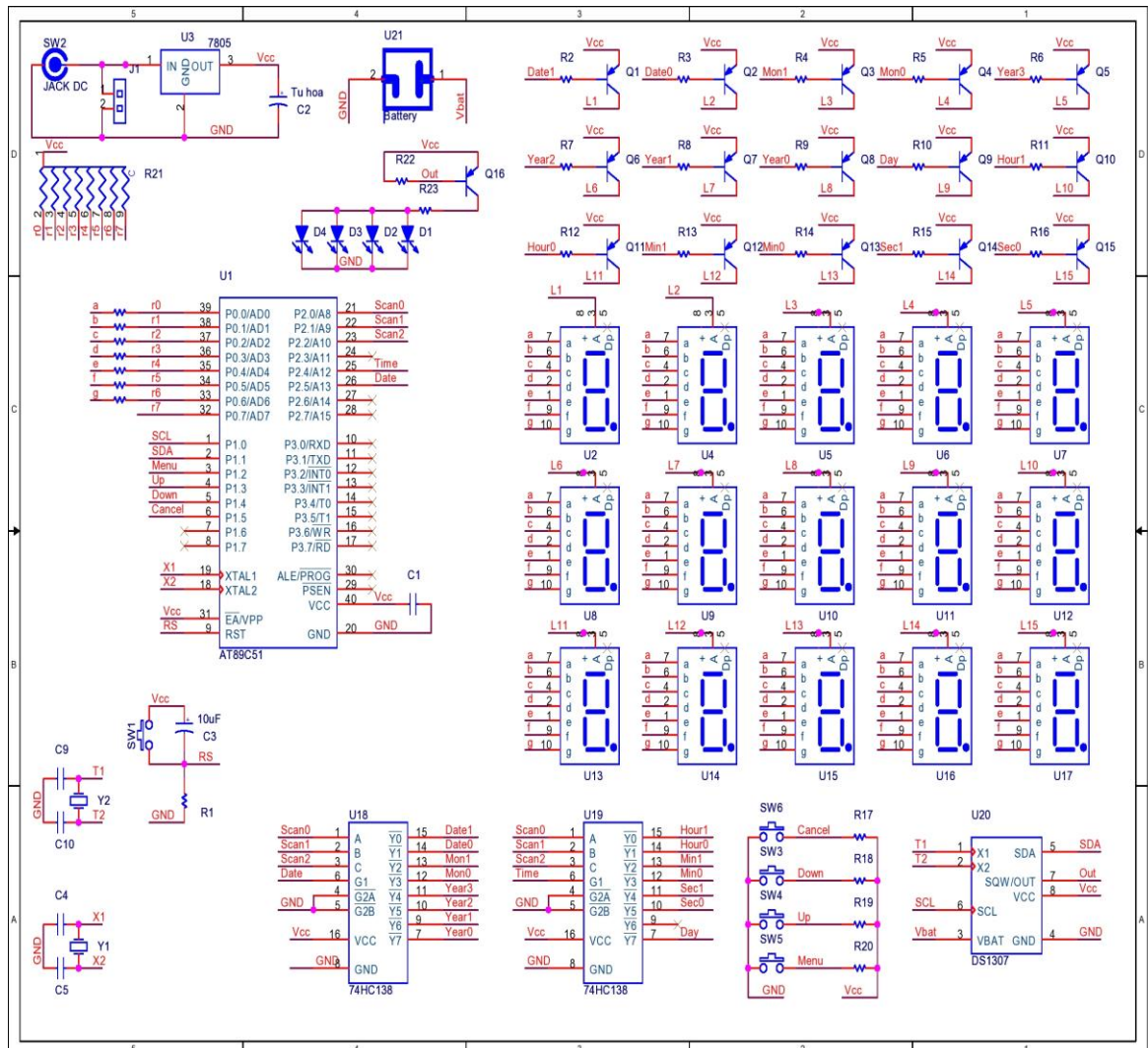
# CHƯƠNG 3.

## THIẾT KẾ VÀ THI CÔNG

### HỆ THỐNG HIỂN THỊ THỜI GIẠN THỰC

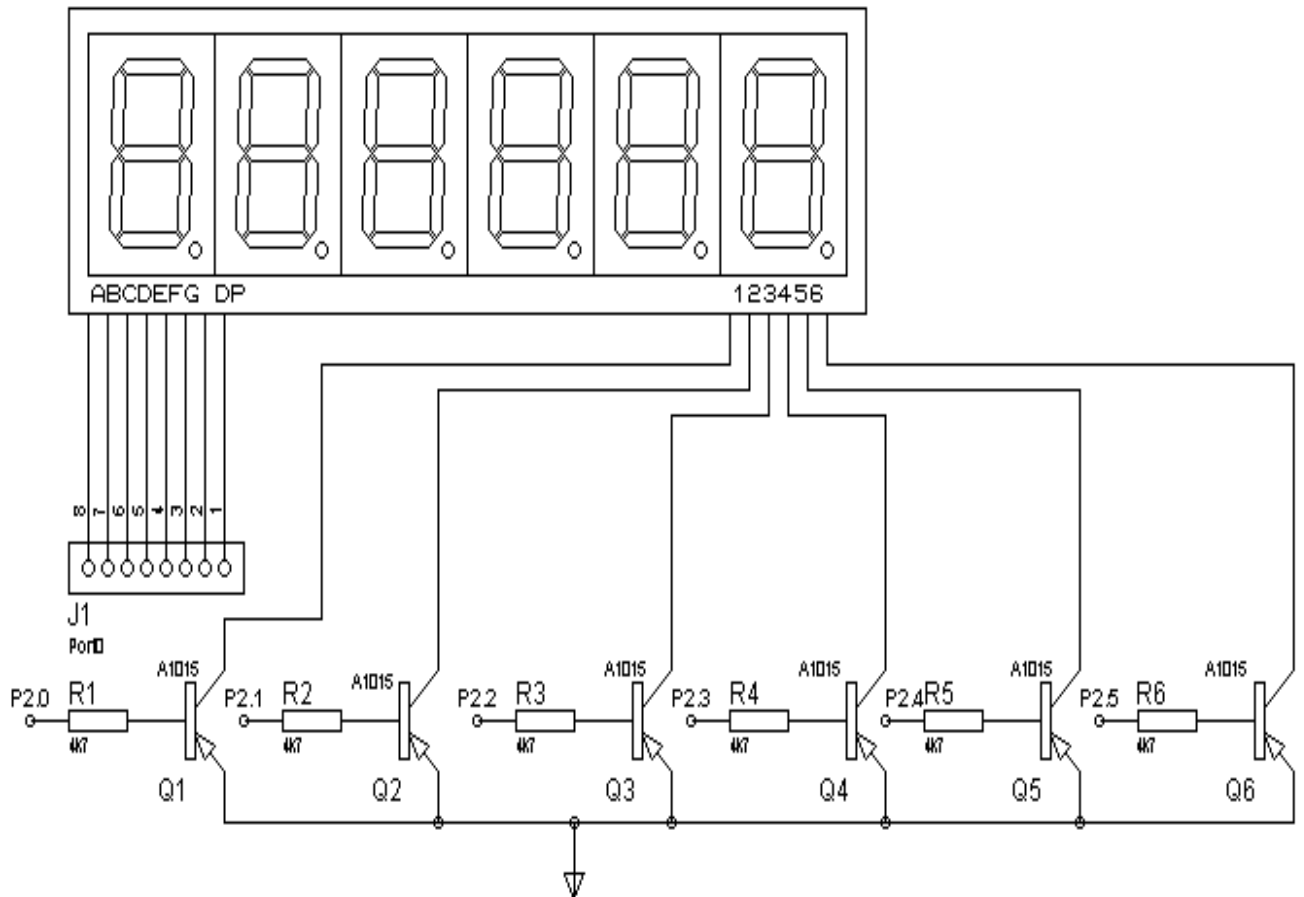
#### 3.2. THIẾT KẾ PHẦN CỨNG.

##### 3.2.1. Sơ đồ nguyên lý mạch điều khiển.



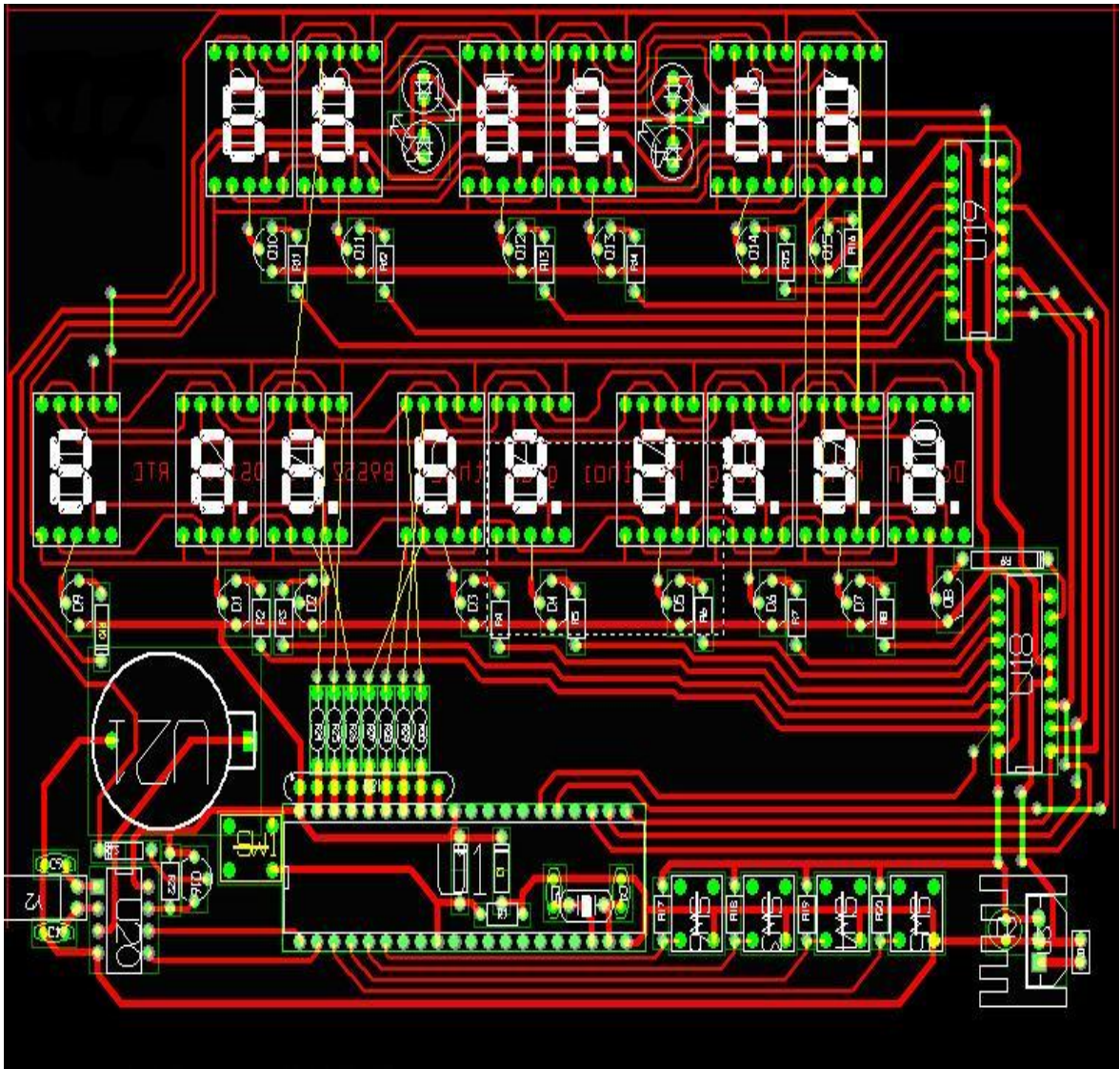
Hình 3.1 : Sơ đồ nguyên lý mạch điều khiển

### 3.1.2. Sơ đồ nguyên lý mạch hiển thị.

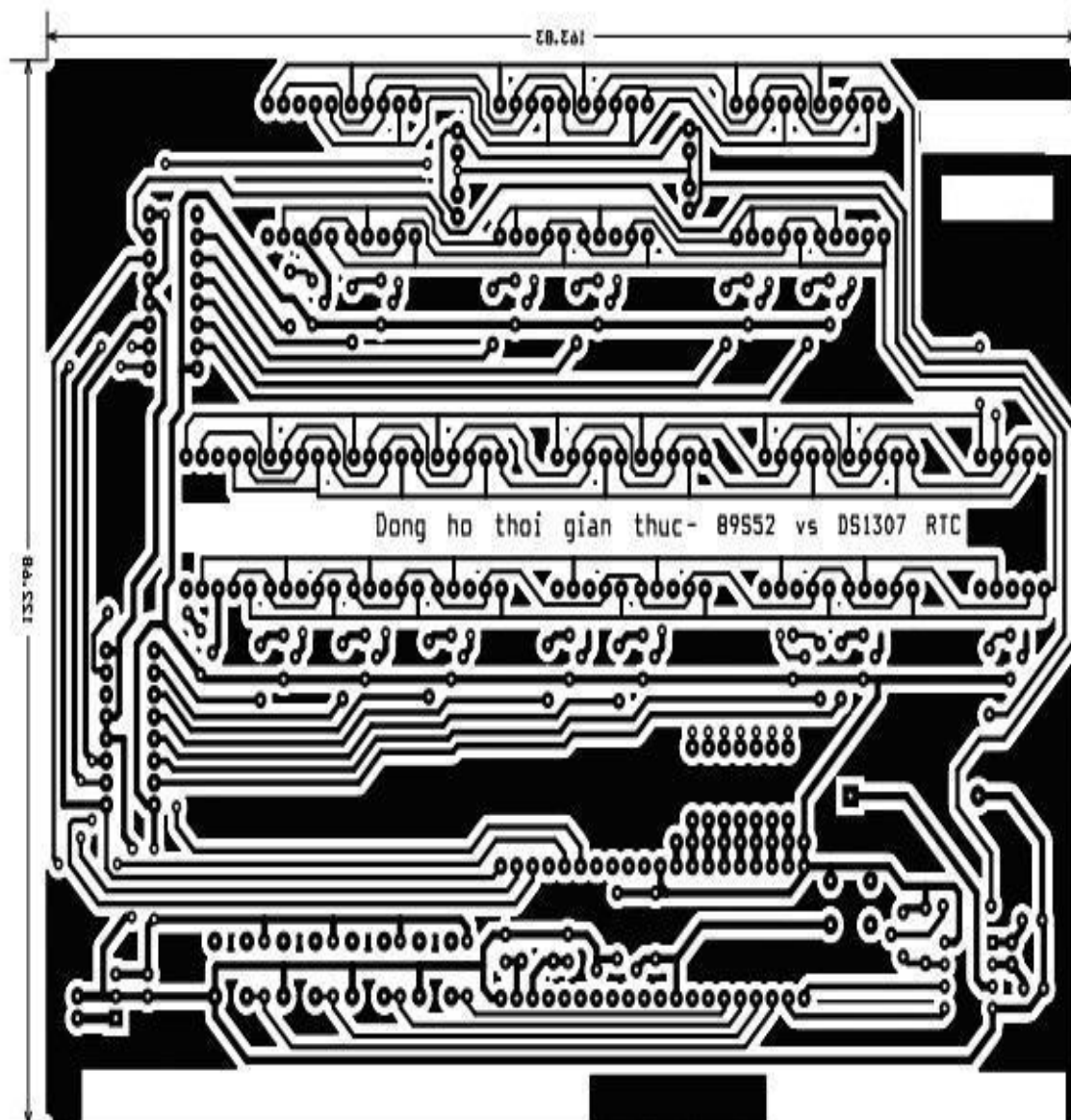


**Hình 3.2** : Sơ đồ nguyên lý mạch hiển thị

### 3.1.3. Sơ đồ mạch in mạch điều khiển.

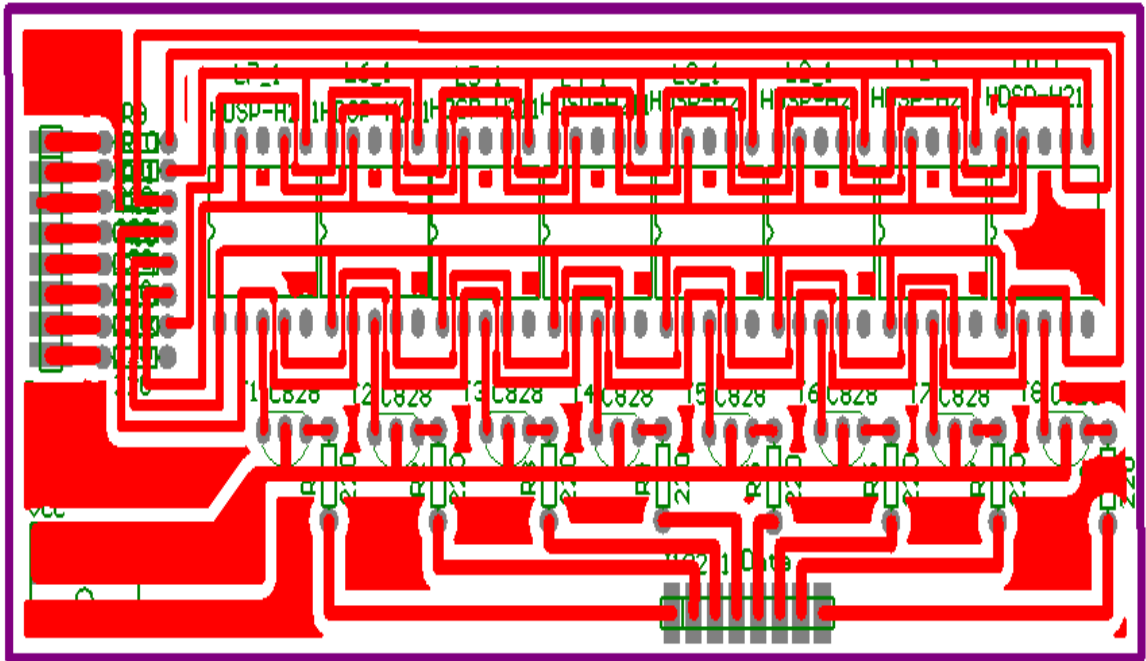


*Hình 3.3* : Mạch điều khiển vẽ trên ORCAD

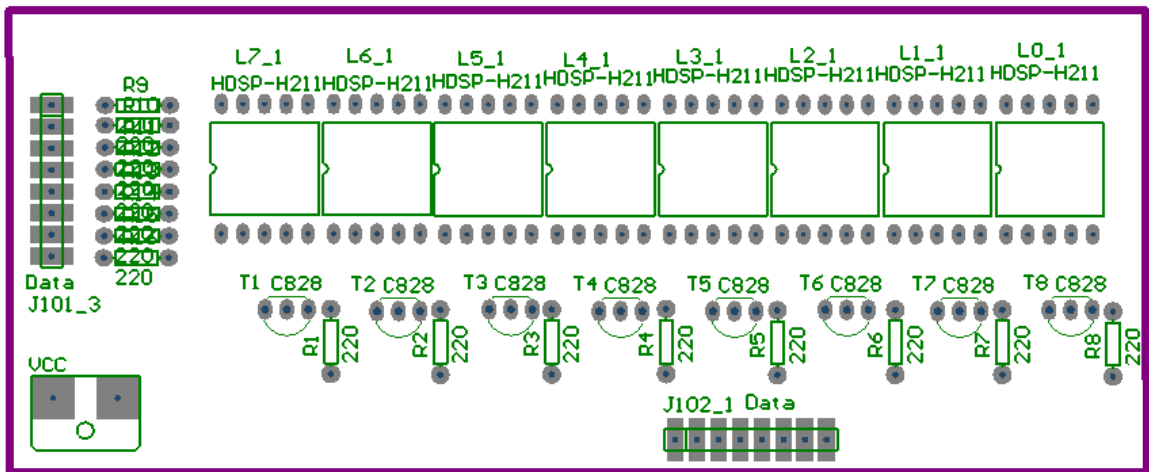


*Hình 3.4* : Mạch điều khiển in một lớp

### 3.3.4. Sơ đồ mạch in mạch hiển thị.



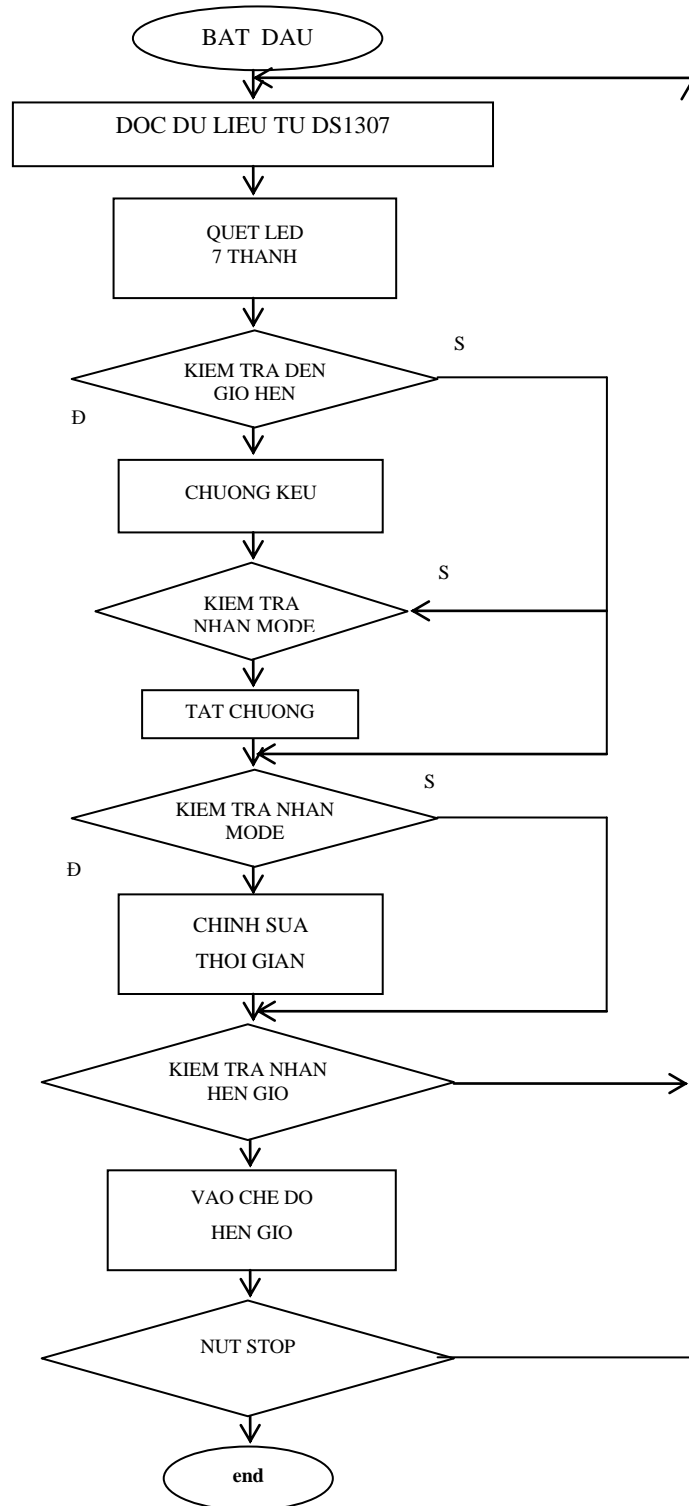
Hình 3.5: Mạch hiển thị vẽ trên Protel.



Hình 3.6: Mạch hiển thị in một lớp.

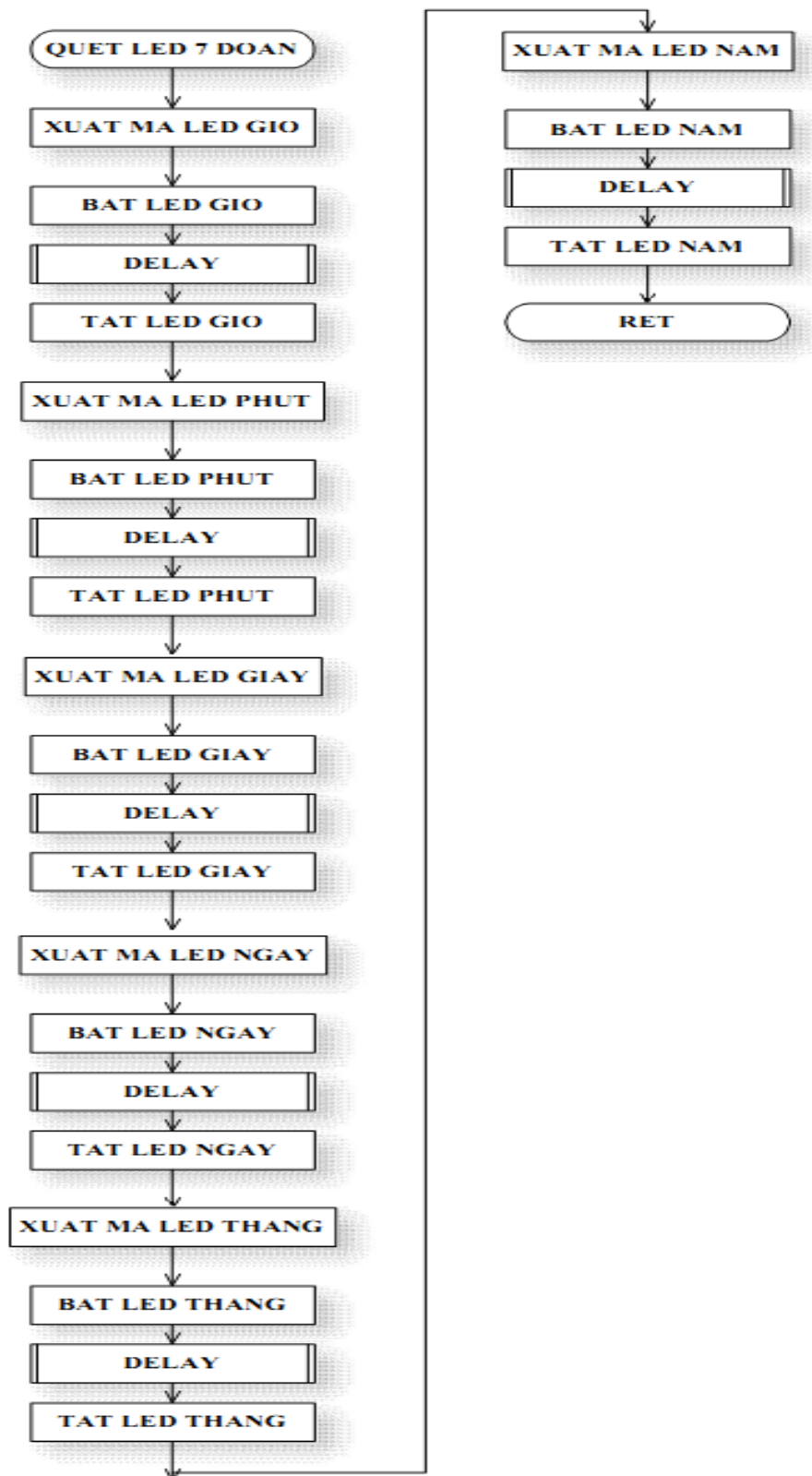
### 3.4. Thiết kế phần mềm.

#### 3.2.1 Lưu đồ thuật toán.





3.2.2. Lưu đồ hiển thị thời gian thực lên LED 7 thanh.



Hình 3.7: Lưu đồ hiển thị thời gian thực trên LED 7 thanh.



Đây là lưu đồ chuyển mã thời gian thực đọc được từ led 7 đoạn và hiển thị, đề hiển thị giá trị thời gian lên led 7 đoạn, trước hết phải chuyển giá trị muốn hiển thị thành mã led 7 đoạn bằng cách sử dụng bảng mã led 7 đoạn anode chung. Tiếp theo, xuất mã led ra và bật led tương ứng với mã đó (ví dụ: mã giây tương ứng với led giây), delay 1 khoảng thời gian sau đó tắt led đó đi. Nhờ hiện tượng lưu ảnh của mắt khi vi điều khiển quét led với tần số cao trên 60Hz thì mắt sẽ thấy các led sáng cùng lúc.

### 3.2.3. Lưu đồ chỉnh sửa thời gian.

Để cài đặt thì các biến giờ, phút, giây, ngày, tháng, năm sẽ được tăng giảm thông qua việc kiểm tra các nút “TANG” và “GIAM”. Việc đổi cài đặt sang giá trị nào được thực hiện qua việc kiểm tra nút “MODE”. Sau đó dữ liệu được lưu vào DS1307 để nó cập nhật giá trị thời gian mới.



Hình 3.8: Lưu đồ chỉnh sửa thời gian

### 3.2.3. Mã nguồn chương trình.

#### CHƯƠNG TRÌNH

```
;P0 DATA Led 7 đoạn
;P2 QUÉT LED

TBI1 BIT P3.1 ; Đặt Bit P3.1 điều khiển thiết bị 1.
TBI2 BIT P3.3 ; Đặt Bit P3.3 điều khiển thiết bị 2.
$INCLUDE(REG51.INC) ; Khai báo thư viện sử dụng.
ORG 0000H ; Nhập giá trị ban đầu.

;THOI GIAN ON THIET BI 1
MOV 20H,#00 ;D/VI GIAY
MOV 21H,#00 ;CHUC GIAY
MOV 22H,#00 ;D/VI PHUT
MOV 23H,#00 ;CHUC PHUT
MOV 24H,#00 ;D/VI GIO
MOV 25H,#00 ;CHUC GIO

;THOI GIAN OFF THIET BI 1
MOV 26H,#00 ;D/VI GIAY
MOV 27H,#00 ;CHUC GIAY
MOV 28H,#01 ;D/VI PHUT
MOV 29H,#00 ;CHUC PHUT
MOV 2AH,#00 ;D/VI GIO
MOV 2BH,#00 ;CHUC GIO

;THOI GIAN ON THIET BI 2
MOV 30H,#00 ;D/VI GIAY
MOV 31H,#00 ;CHUC GIAY
MOV 32H,#00 ;D/VI PHUT
MOV 33H,#00 ;CHUC PHUT
MOV 34H,#00 ;D/VI GIO
MOV 35H,#00 ;CHUC GIO

;THOI GIAN OFF THIET BI 2
```

```

MOV 36H,#00 ;D/VI GIAY
MOV 37H,#00 ;CHUC GIAY
MOV 38H,#02 ;D/VI PHUT
MOV 39H,#00 ;CHUC PHUT
MOV 3AH,#00 ;D/VI GIO
MOV 3BH,#00 ;CHUC GIO
CLR TBI1_OUT ; Xóa bit P3.1
CLR TBI2_OUT ; Xóa bit P3.3

LCALL KTAODS ; Gọi chương trình con KTAODS.
LB: LCALL DOCGPG ; Gọi chương trình con ĐọcGiờ Phút Giây.
LCALL HEXBCD ; Gọi chương trình con HEXBCD.
LCALL BCD7DOAN ; Gọi chương trình con BDC7Đoạn.
LCALL HIENHTHI ; Gọi chương trình con Hiện Thị.
LCALL QUETPHIM ; Gọi chương trình con Quét Phím.
LCALL KTPHIM ; Gọi chương trình con Khởi tạo phím.
CJNE A,#0AH,LB1 ; So sánh #0AH với A và nhảy đến LB1 nếu không bằng.
LCALL SETTING ; Gọi chương trình con Setting.

LB1: LCALL KTTBI ; Gọi chương trình con Khởi Tạo Thiết Bị.
SJMP LB ; Nhảy đến LB
KTTBI:MOV A,55H ; Chuyển dữ liệu ô 55H vào A.
CJNE A,25H,KT1 ; So sánh ô 25H với A và nhảy đến KT1 nếu không bằng.
MOV A,54H ; Chuyển dữ liệu ô 54H vào A.
CJNE A,24H,KT1 ; So sánh ô 24H với A và nhảy đến KT1 nếu không bằng.
MOV A,53H ; Chuyển dữ liệu ô 53H vào A.
CJNE A,23H,KT1 ; So sánh ô 23H với A và nhảy đến KT1 nếu không bằng.
MOV A,52H ; Chuyển dữ liệu ô 52H vào A.
CJNE A,22H,KT1 ; So sánh ô 22H với A và nhảy đến KT1 nếu không bằng.
MOV A,51H ; Chuyển dữ liệu ô 51H vào A.
CJNE A,21H,KT1 ; So sánh ô 21H với A và nhảy đến KT1 nếu không bằng.
MOV A,50H ; Chuyển dữ liệu ô 50H vào A.
CJNE A,20H,KT1 ; So sánh ô 20H với A và nhảy đến KT1 nếu không bằng.
SETB TBI1 ; Bật Thiết Bị 1.

KT1: MOV A,55H ; Chuyển dữ liệu ô 55H vào A.
CJNE A,2BH,KT2 ; So sánh ô 2BH với A và nhảy đến KT2 nếu không bằng.

```

MOV A,54H ; Chuyển dữ liệu ô 54H vào A.  
 CJNE A,2AH,KT2 ; So sánh ô 2AH với A và nhảy đến KT2 nếu không bằng.  
 MOV A,53H ; Chuyển dữ liệu ô 53H vào A.  
 CJNE A,29H,KT2 ; So sánh ô 29H với A và nhảy đến KT2 nếu không bằng.  
 MOV A,52H ; Chuyển dữ liệu ô 52H vào A.  
 CJNE A,28H,KT2 ; So sánh ô 28H với A và nhảy đến KT2 nếu không bằng.  
 MOV A,51H ; Chuyển dữ liệu ô 51H vào A.  
 CJNE A,27H,KT2 ; So sánh ô 27H với A và nhảy đến KT2 nếu không bằng.  
 MOV A,50H ; Chuyển dữ liệu ô 50H vào A.  
 CJNE A,26H,KT2 ; So sánh ô 26H với A và nhảy đến KT2 nếu không bằng.  
 CLR TBI1 ;Tắt Thiết Bị 1.

KT2: MOV A,55H ; Chuyển dữ liệu ô 55H vào A.  
 CJNE A,35H,KT3 ;So sánh ô 35H với A và nhảy đến KT3 nếu không bằng.  
 MOV A,54H ; Chuyển dữ liệu ô 54H vào A.  
 CJNE A,34H,KT3 ;So sánh ô 34H với A và nhảy đến KT3 nếu không bằng.  
 MOV A,53H ; Chuyển dữ liệu ô 53H vào A.  
 CJNE A,33H,KT3 ;So sánh ô 33H với A và nhảy đến KT3 nếu không bằng.  
 MOV A,52H ; Chuyển dữ liệu ô 52H vào A.  
 CJNE A,32H,KT3 ;So sánh ô 32H với A và nhảy đến KT3 nếu không bằng.  
 MOV A,51H ; Chuyển dữ liệu ô 51H vào A.  
 CJNE A,31H,KT3 ;So sánh ô 31H với A và nhảy đến KT3 nếu không bằng.  
 MOV A,50H ; Chuyển dữ liệu ô 50H vào A.  
 CJNE A,30H,KT3 ;So sánh ô 30H với A và nhảy đến KT3 nếu không bằng.  
 SETB TBI2 ; Bật Thiết Bị 2.

KT3: MOV A,55H ; Chuyển dữ liệu ô 55H vào A.  
 CJNE A,3BH,KT4 ;So sánh ô 3BH với A và nhảy đến KT4 nếu không bằng.  
 MOV A,54H ; Chuyển dữ liệu ô 54H vào A.  
 CJNE A,3AH,KT4 ;So sánh ô 3AH với A và nhảy đến KT4 nếu không bằng.  
 MOV A,53H ; Chuyển dữ liệu ô 53H vào A.  
 CJNE A,39H,KT4 ;So sánh ô 39H với A và nhảy đến KT4 nếu không bằng.  
 MOV A,52H ; Chuyển dữ liệu ô 52H vào A.  
 CJNE A,38H,KT4 ;So sánh ô 38H với A và nhảy đến KT4 nếu không bằng.  
 MOV A,51H ; Chuyển dữ liệu ô 51H vào A.  
 CJNE A,37H,KT4 ;So sánh ô 37H với A và nhảy đến KT4 nếu không bằng.  
 MOV A,50H ; Chuyển dữ liệu ô 50H vào A.

```

CJNE A,36H,KT4 ;So sánh ô 36H với A và nhảy đến KT4 nếu không bằng.
CLR TBI2 ; Tắt Thiết Bị 2.
KT4: RET ; Thoát khỏi ct con Khởi tạo thiết bị.

```

```

SETTING: ; Chương trình con Setting.

```

```

X4: MOV 50H,#0FFH ;TAT LED
MOV 51H,#0FFH ;TAT LED
MOV 52H,#0FFH ;TAT LED
MOV 53H,#0FFH ;TAT LED
MOV 54H,#0FFH ;TAT LED
MOV 55H,#0FFH ;TAT LED
LCALL BCD7DOAN ; Gọi chương trình con BCD7 Đoạn.
LCALL HIEN THI ; Gọi chương trình con Hiện Thị.
LCALL QUET PHIM ; Gọi chương trình con Quét Phím.
LCALL KTPHIM ; Gọi chương trình con Khởi Tạo Phím.
CJNE A,#0CH,X1 ; T/BI1
MOV 71H,#00H ; Chuyển 00H vào ô 71H
LCALL TGIAN1 ; Gọi chương trình con TGian1.
SJMP X4 ; Nhảy đến X4
X1: CJNE A,#0DH,X2 ; T/BI2
MOV 71H,#00H ; Chuyển 00H vào ô 71H
LCALL TGIAN2 ; Gọi chương trình con TGian2.
SJMP X4 ; Nhảy đến X4
X2: CJNE A,#0EH,X3 ; T/BI3
MOV 71H,#00H ; Chuyển 00H vào ô 71H
LCALL TGIAN3 ; Gọi chương trình con TGian3.
SJMP X4 ; Nhảy đến X4
X3: CJNE A,#0AH,X4 ; SET
RET ; Thoát khỏi chương trình con.
TGIAN1: ;Chương trình con TGian1.
7: LCALL BCD7DOAN ; Gọi chương trình con BCD7 Đoạn.
LCALL HIEN THI ; Gọi chương trình con Hiện Thị.
LCALL QUET PHIM ; Gọi chương trình con Quét Phím.
LCALL KTPHIM ; Gọi chương trình con Khởi tạo Phím.
CJNE A,#0BH,X5 ; PHIM ON
MOV 20H,50H ; Chuyển dữ liệu ô 50H vào ô 20H.
MOV 21H,51H ; Chuyển dữ liệu ô 51H vào ô 21H.

```

```

MOV 22H,52H ; Chuyển dữ liệu ô 52H vào ô 22H.
MOV 23H,53H ; Chuyển dữ liệu ô 53H vào ô 23H.
MOV 24H,54H ; Chuyển dữ liệu ô 54H vào ô 24H.
MOV 25H,55H ; Chuyển dữ liệu ô 55H vào ô 250H.
SJMP X8 ; Nhảy đến X8

```

```

X5: CJNE A,#0FH,X6 ;PHIM OFF
MOV 26H,50H ; Chuyển dữ liệu ô 50H vào ô 26H.
MOV 27H,51H ; Chuyển dữ liệu ô 51H vào ô 27H.
MOV 28H,52H ; Chuyển dữ liệu ô 52H vào ô 28H.
MOV 29H,53H ; Chuyển dữ liệu ô 53H vào ô 29H.
MOV 2AH,54H ; Chuyển dữ liệu ô 54H vào ô 2AH.
MOV 2BH,55H ; Chuyển dữ liệu ô 55H vào ô 2BH.
SJMP X8 ; Nhảy đến X8

```

```

X6: MOV A,71H ; Chuyển dữ liệu ô 71H vào thanh ghi A.
CJNE A,#0FFH,X7 ; So sánh FFH với A và nhảy đến X7 nếu không bằng.
MOV 55H,54H ; Chuyển dữ liệu ô 54H vào ô 55H.
MOV 54H,53H ; Chuyển dữ liệu ô 53H vào ô 54H.
MOV 53H,52H ; Chuyển dữ liệu ô 52H vào ô 53H.
MOV 52H,51H ; Chuyển dữ liệu ô 51H vào ô 52H.
MOV 51H,50H ; Chuyển dữ liệu ô 50H vào ô 51H.
MOV 50H,70H ; Chuyển dữ liệu ô 70H vào ô 50H.
MOV 71H,#00H ; Chuyển gtri 00H vào ô 71H
SJMP X7 ; Nhảy đến X7

```

```

X8: RET ;Thoát khỏi chương trình con.

```

```

TGIAN2: ; Chương trình con Tgian 2.
X11: LCALL BCD7DOAN ; Gọi chương trình con BCD7 Đoạn.
LCALL HIEN THI ; Gọi chương trình con Hiện Thị.
LCALL QUETPHIM ; Gọi chương trình con Quét Phím.
LCALL KTPHIM ; Gọi chương trình con Khởi tạo phím.
CJNE A,#0BH,X9 ;PHIM ON
MOV 30H,50H ; Chuyển dữ liệu ô 50H vào ô 30H.
MOV 31H,51H ; Chuyển dữ liệu ô 51H vào ô 31H.
MOV 32H,52H ; Chuyển dữ liệu ô 52H vào ô 32H.

```

```

MOV 33H,53H ; Chuyển dữ liệu ô 53H vào ô 33H.
MOV 34H,54H ; Chuyển dữ liệu ô 54H vào ô 34H.
MOV 35H,55H ; Chuyển dữ liệu ô 55H vào ô 35H.
SJMP X12 ; Nhảy đến X12.

```

```

X9: CJNE A,#0FH,X10 ;PHIM OFF
MOV 36H,50H ; Chuyển dữ liệu ô 50H vào ô 36H.
MOV 37H,51H ; Chuyển dữ liệu ô 51H vào ô 37H.
MOV 38H,52H ; Chuyển dữ liệu ô 52H vào ô 38H.
MOV 39H,53H ; Chuyển dữ liệu ô 53H vào ô 39H.
MOV 3AH,54H ; Chuyển dữ liệu ô 54H vào ô 3AH.
MOV 3BH,55H ; Chuyển dữ liệu ô 55H vào ô 3BH.
SJMP X12 ; Nhảy đến X12.

```

```

X10: MOV A,71H ; Chuyển dữ liệu ô 71H vào thanh ghi A.
CJNE A,#0FFH,X11 ; So sánh FFH với A và nhảy đến X11 nếu không
bằng.
MOV 55H,54H ; Chuyển dữ liệu ô 54H vào ô 55H.
MOV 54H,53H ; Chuyển dữ liệu ô 53H vào ô 54H.
MOV 53H,52H ; Chuyển dữ liệu ô 52H vào ô 53H.
MOV 52H,51H ; Chuyển dữ liệu ô 51H vào ô 52H.
MOV 51H,50H ; Chuyển dữ liệu ô 50H vào ô 51H.
MOV 50H,70H ; Chuyển dữ liệu ô 70H vào ô 50H.
MOV 71H,#00H ; Chuyển 00h vào ô 71H.
SJMP X11 ; Nhảy đến X11.

```

```

X12: RET ; Thoát khỏi ct con.

```

```

KTAODS: ; Chương trình Khởi tạo Dallas
MOV R0,#10 ; Chuyển giá trị 10 vào thanh ghi R0.
MOV A,#00100000B; Chuyển giá trị 00100000B vào thanh ghi A.
MOVX @R0,A ; Chuyển nội dung thanh ghi A ra dữ liệu ngoài.
MOV R0,#11 ; Chuyển giá trị 11 vào thanh ghi R0.
MOV A,#06H ; Chuyển giá trị 06H vào thanh ghi A.
MOVX @R0,A ; Chuyển nội dung thanh ghi A ra dữ liệu ngoài.
RET ; Thoát khỏi ct con.

```

```

QUETPHIM:                ; Chương trình Quét Phím.
CLR  P1.4                ; Xóa bit P1.4
JB   P1.0,QUET1         ; Nhảy đến Quet1 nếu P1.0 được đặt.
Y1:  LCALL HIENTHI      ; Gọi ct con Hiền THị.
JNB  P1.0,Y1           ; Nhảy đến Y1 nếu bit p1.0 ko được đặt.
MOV  A,#0CH            ;T/BI1

QUET1:JB  P1.1,QUET2   ; Nhảy đến Quet2 nếu P1.1 được đặt.
Y2:  LCALL HIENTHI      ; Gọi ct con Hiền THị.
JNB  P1.1,Y2           ; Nhảy đến Y2 nếu bit p1.1 ko được đặt.
MOV  A,#0DH            ;T/BI2

QUET2:JB  P1.2,QUET3   ; Nhảy đến Quet3 nếu P1.2 được đặt.
Y3:  LCALL HIENTHI      ; Gọi ct con Hiền THị.
JNB  P1.2,Y3           ; Nhảy đến Y3 nếu bit p1.2 ko được đặt.
MOV  A,#0EH            ;T/BI3

QUET3:JB  P1.3,QUET4   ; Nhảy đến Quet4 nếu P1.3 được đặt.
Y4:  LCALL HIENTHI      ; Gọi ct con Hiền THị.
JNB  P1.3,Y4           ; Nhảy đến Y3 nếu bit p1.2 ko được đặt.
MOV  A,#0FH            ;PHIM OFF

QUET4:SETB P1.4         ; Đặt bit P1.4
CLR  P1.5                ; Xoá bit p1.5
JB   P1.0,QUET5         ; Nhảy đến Quet5 nếu P1.0 được đặt.
Y5:  LCALL HIENTHI      ; Gọi ct con Hiền THị.
JNB  P1.0,Y5           ; Nhảy đến Y5 Nếu P1.0 được đặt.
MOV  A,#08H            ;SO 8

QUET5:JB  P1.1,QUET6   ; Nhảy đến Quet6 nếu P1.1 được đặt.
Y6:  LCALL HIENTHI      ; Gọi ct con Hiền THị.
JNB  P1.1,Y6           ; Nhảy đến Y6 nếu P1.1 ko được đặt.
MOV  A,#09H            ;SO 9

QUET6:JB  P1.2,QUET7   ; Nhảy đến Quet7 nếu P1.2 được đặt.
Y7:  LCALL HIENTHI      ; Gọi ct con Hiền THị.

```



```

JNB P1.2,Y7 ; Nhảy đến Y7 nếu P1.2 ko được đặt.
MOV A,#0AH ;PHIM SET

QUET7:JB P1.3,QUET8 ; Nhảy đến Quet8 nếu P1.3 được đặt.
Y8: LCALL HIENTHI ; Gọi ct con Hiền THị.
JNB P1.3,Y8 ; Nhảy đến Y8 nếu P1.3 ko được đặt.
MOV A,#0BH ;PHIM ON

QUET8:SETB P1.5 ; Đặt bit P1.5
CLR P1.6 ; Xoá bit p1.6
JB P1.0,QUET9 ; Nhảy đến Quet9 nếu P1.0 được đặt.
Y9: LCALL HIENTHI ; Gọi ct con Hiền THị.
JNB P1.0,Y9 ; Nhảy đến Y9 nếu P1.3 ko được đặt.
MOV A,#04H ;SO 4

QUET9:JB P1.1,QUET10 ; Nhảy đến Quet10 nếu P1.1 được đặt.
Y10: LCALL HIENTHI ; Gọi ct con Hiền THị.
JNB P1.1,Y10 ; Nhảy đến Y10 nếu P1.1 ko được đặt.
MOV A,#05H ;SO 5

QUET10:JB P1.2,QUET11 ; Nhảy đến Quet11 nếu P1.2 được đặt.
Y11: LCALL HIENTHI ; Gọi ct con Hiền THị.
JNB P1.2,Y11 ; Nhảy đến Y11 nếu P1.2 ko được đặt.
MOV A,#06H ;SO 6

QUET11:JB P1.3,QUET12 ; Nhảy đến Quet12 nếu P1.3 được đặt.
Y12: LCALL HIENTHI ; Gọi ct con Hiền THị.
JNB P1.3,Y12 ; Nhảy đến Y12 nếu P1.3 ko được đặt.
MOV A,#07H ;SO 7

QUET12:SETB P1.6 ; Đặt bit P1.6
CLR P1.7 ; Xoá bit p1.7
JB P1.0,QUET13 ; Nhảy đến Quet13 nếu P1.0 được đặt.
Y13: LCALL HIENTHI ; Gọi ct con Hiền THị.
JNB P1.0,Y13 ; Nhảy đến Y13 nếu P1.0 ko được đặt.
MOV A,#00H ;SO 0

```

```

QUET13:JB P1.1,QUET14 ; Nhảy đến Quet14 nếu P1.1 được đặt.
Y14: LCALL HIEN THI ; Gọi ct con Hiền THị.
JNB P1.1,Y14 ; Nhảy đến Y14 nếu P1.1 ko được đặt.
MOV A,#01H ;SO 1

QUET14:JB P1.2,QUET15 ; Nhảy đến Quet15 nếu P1.2 được đặt.
Y15: LCALL HIEN THI ; Gọi ct con Hiền THị.
JNB P1.2,Y15 ; Nhảy đến Y15 nếu P1.2 ko được đặt.
MOV A,#02H ;SO 2

QUET15:JB P1.3,QUET16 ; Nhảy đến Quet16 nếu P1.3 được đặt.
Y16: LCALL HIEN THI ; Gọi ct con Hiền THị.
JNB P1.3,Y16 ; Nhảy đến Y16 nếu P1.3 ko được đặt.
MOV A,#03H ;SO 3

QUET16:SETB P1.7 ; Đặt bit p1.7
RET ; Thoat trở lại lệnh gọi ct con.

KTPHIM:CJNE A,#00H,Y17 ;SO 0
MOV 70H,#00H ; Chuyển giá trị 00H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y17: CJNE A,#01H,Y18 ;SO 1
MOV 70H,#01H ; Chuyển giá trị 01H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y18: CJNE A,#02H,Y19 ;SO 2
MOV 70H,#02H ; Chuyển giá trị 02H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y19: CJNE A,#03H,Y20 ;SO 3
MOV 70H,#03H ; Chuyển giá trị 03H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y20: CJNE A,#04H,Y21 ;SO 4
MOV 70H,#04H ; Chuyển giá trị 04H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

```

```

Y21: CJNE A,#05H,Y22 ;SO 5
MOV 70H,#05H ; Chuyển giá trị 05H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y22: CJNE A,#06H,Y23 ;SO 6
MOV 70H,#06H ; Chuyển giá trị 06H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y23: CJNE A,#07H,Y24 ;SO 7
MOV 70H,#07H ; Chuyển giá trị 07H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y24: CJNE A,#08H,Y25 ;SO 8
MOV 70H,#08H ; Chuyển giá trị 08H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y25: CJNE A,#09H,Y26 ;SO 9
MOV 70H,#09H ; Chuyển giá trị 09H vào ô 70H.
MOV 71H,#0FFH ; Chuyển giá trị FFH vào ô 71H.

Y26: CJNE A,#0AH,Y27 ;PHIM SET
Y27: CJNE A,#0BH,Y28 ;PHIM ON
Y28: CJNE A,#0CH,Y29 ;PHIM T/BI1
Y29: CJNE A,#0DH,Y30 ;PHIM T/BI2
Y30: CJNE A,#0EH,Y31 ;PHIM T/BI3
Y31: CJNE A,#0FH,Y32 ;PHIM OFF
Y32: RET ; Thoát trở lại lệnh gọi ct con.
DOCGPG: ; Chương trình Đọc giờ phút giây.
MOV R0,#00H ; Chuyển giá trị 00H vào thanh ghi R0.
MOV R1,#10H ; Chuyển giá trị 10H vào thanh ghi R1.
DS1: MOVX A,@R0 ; Chuyển dữ liệu bên ngoài vào thanh ghi A.
MOV @R1,A ; Chuyển nội dung thanh ghi A ra bên ngoài.
INC R1 ; Tăng thanh ghi R1 lên 1.
INC R0 ; Tăng thanh ghi R0 lên 1.
CJNE R0,#10,DS1 ; So sánh 10 với R0 và nhảy đến DS1 nếu ko bằng.
RET ; Thoát trở lại lệnh gọi ct con.

```

```

HEXBCD:                ; Chương trình con HexBCD
MOV  A,10H              ; Chuyển dữ liệu ô 10H vào thanh ghi A.
MOV  B,#10              ; Chuyển giá trị 10 vào thanh ghi B.
DIV  AB                 ; Chia nội dung thanh ghi A cho thanh ghi B.
MOV  50H,B              ;GIAY
MOV  51H,A              ; Chuyển nội dung thanh ghi A vào ô 51H.

MOV  A,12H              ; Chuyển dữ liệu ô 12H vào thanh ghi A.
MOV  B,#10              ; Chuyển giá trị 10 vào thanh ghi B.
DIV  AB                 ; Chia nội dung thanh ghi A cho thanh ghi B.
MOV  52H,B              ;PHUT
MOV  53H,A              ; Chuyển nội dung thanh ghi A vào ô 53H.

MOV  A,14H              ; Chuyển dữ liệu ô 14H vào thanh ghi A.
MOV  B,#10              ; Chuyển giá trị 10 vào thanh ghi B.
DIV  AB                 ; Chia nội dung thanh ghi A cho thanh ghi B.
MOV  54H,B              ;GIO
MOV  55H,A              ; Chuyển nội dung thanh ghi A vào ô 55H.

RET                     ; Thoát trở lại lệnh gọi ct con.

BCD7DOAN:              ; Chương trình con BCD 7 Đoạn.
MOV  DPTR,#500H         ; Chuyển 500H vào con trỏ dữ liệu.
MOV  A,50H              ; Chuyển dữ liệu ô 50H vào thanh ghi A.
MOVC A,@A+DPTR          ; Chuyển byte bộ nhớ ct vào thanh ghi A.
MOV  56H,A              ; Chuyển nội dung thanh ghi A vào ô 56H.

MOV  A,51H              ; Chuyển dữ liệu ô 51H vào thanh ghi A.
MOVC A,@A+DPTR          ; Chuyển byte bộ nhớ ct vào thanh ghi A.
MOV  57H,A              ; Chuyển nội dung thanh ghi A vào ô 567.

MOV  A,52H              ; Chuyển dữ liệu ô 52H vào thanh ghi A.
MOVC A,@A+DPTR          ; Chuyển byte bộ nhớ ct vào thanh ghi A.
MOV  58H,A              ; Chuyển nội dung thanh ghi A vào ô 58H.

MOV  A,53H              ; Chuyển dữ liệu ô 53H vào thanh ghi A.

```

```

MOVC A,@A+DPTR ; Chuyển byte bộ nhớ ct vào thanh ghi A.
MOV 59H,A ; Chuyển nội dung thanh ghi A vào ô 59H.

MOV A,54H ; Chuyển dữ liệu ô 54H vào thanh ghi A.
MOVC A,@A+DPTR ; Chuyển byte bộ nhớ ct vào thanh ghi A.
MOV 5AH,A ; Chuyển nội dung thanh ghi A vào ô 5AH.

MOV A,55H ; Chuyển dữ liệu ô 55H vào thanh ghi A.
MOVC A,@A+DPTR ; Chuyển byte bộ nhớ ct vào thanh ghi A.
MOV 5BH,A ; Chuyển nội dung thanh ghi A vào ô 5BH.

RET ; Thoát trở lại lệnh gọi ct con.

```

HIENTHI:

```

MOV P0,56H ; Chuyển nội dung ô 56H ra Port 0.
SETB P2.0 ; Đặt bit P2.0.
LCALL DELAY ; Gọi ct con Delay.
CLR P2.0 ; Xóa bit P2.0.

MOV P0,57H ; Chuyển nội dung ô 57H ra Port 0.
SETB P2.1 ; Đặt bit P2.1.
LCALL DELAY ; Gọi ct con Delay.
CLR P2.1 ; Xóa bit P2.1.

MOV P0,58H ; Chuyển nội dung ô 58H ra Port 0.
SETB P2.2 ; Đặt bit P2.2.
LCALL DELAY ; Gọi ct con Delay.
CLR P2.2 ; Xóa bit P2.2.

MOV P0,59H ; Chuyển nội dung ô 59H ra Port 0.
SETB P2.3 ; Đặt bit P2.3.
LCALL DELAY ; Gọi ct con Delay.
CLR P2.3 ; Xóa bit P2.3.

MOV P0,5AH ; Chuyển nội dung ô 5AH ra Port 0.
SETB P2.4 ; Đặt bit P2.4.
LCALL DELAY ; Gọi ct con Delay.

```

```

CLR  P2.4      ; Xóa bit P2.4.

MOV  P0,5BH    ; Chuyển nội dung ô 5BH ra Port 0.
SETB P2.5     ; Đặt bit P2.5.
LCALL DELAY    ; Gọi ct con Delay.
CLR  P2.5     ; Xóa bit P2.8.
RET           ; Thoát trở lại lệnh gọi ct con.

DELAY:MOV  7EH,#01    ; Chuyển giá trị 01 vào ô 7E.
DEL1:MOV  7FH,#30    ; Chuyển giá trị 30 vào ô 7F.
DJNZ  7FH,$        ; Giảm giá trị ô 7FH và nhảy nếu ko bằng.
DJNZ  7EH,DEL1     ; Giảm giá trị ô 7EH và nhảy đến Del1 nếu ko bằng.
RET              : Thoát trở lại lệnh gọi ct con.

ORG  500H        ; Địa chỉ của con trở dữ liệu

DB    0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H ;Dữ liệu Led 7 đoạn.

END              ; Kết thúc chương trình.

```

## KẾT LUẬN

Sau một thời gian tìm hiểu, dưới sự hướng dẫn nhiệt tình của thầy Ths.Nguyễn Trọng Thắng, em đã hoàn thành đề tài “Ứng dụng DS1307 xây dựng hệ thống thời gian thực”.

Đồng hồ hiển thị được thứ, ngày, tháng, năm, giờ, phút, có thể cài đặt được thời gian thông qua giao tiếp phím ấn. Báo giờ tự động mỗi ngày.

Qua đó chúng em được củng cố kiến thức đã học, kết hợp được với thực tiễn, phục vụ cho công việc tương lai rất nhiều.

Em đã cố gắng hết sức để thực hiện đề tài một cách hoàn thiện và nhanh nhất có thể, nhưng do trình độ còn hạn chế nên còn nhiều thiếu sót, đề tài mới dừng lại ở việc làm một chiếc đồng hồ hiển thị thời gian đơn giản, chỉ báo giờ được một lần trong ngày.

Cũng với đề tài này, trong tương lai chúng em sẽ phát triển nó ở mức cao hơn, có thể kết hợp chúng với các thiết bị khác để sử dụng trong thực tiễn cuộc sống như đồng hồ kèm đo nhiệt độ, hẹn giờ, chuông báo, lịch làm việc cá nhân hay tập thể,...

Em xin cảm ơn đến các thầy cô trong khoa, đặc biệt là Thạc sỹ Nguyễn Trọng Thắng, thầy đã trực tiếp hướng dẫn em nghiên cứu và thực hiện đề án tốt nghiệp.

## TÀI LIỆU THAM KHẢO

### PHỤ LỤC 1

**Bảng mã hiển thị led 7 đoạn dành cho led 7 đoạn có Anode chung  
(các led đơn sáng ở mức 0):**

<i>Số hiển thị trên led 7 đoạn</i>	<i>Mã hiển thị led 7 đoạn dạng nhị phân</i>	<i>Mã hiển thị led 7 đoạn dạng thập lục phân</i>
	<b>h g f e d c b a</b>	
<b>0</b>	<b>1 1 0 0 0 0 0 0</b>	<b>C0</b>
<b>1</b>	<b>1 1 1 1 1 0 0 1</b>	<b>F9</b>
<b>2</b>	<b>1 0 1 0 0 1 0 0</b>	<b>A4</b>
<b>3</b>	<b>1 0 1 1 0 0 0 0</b>	<b>B0</b>
<b>4</b>	<b>1 0 0 1 1 0 0 1</b>	<b>99</b>
<b>5</b>	<b>1 0 0 1 0 0 1 0</b>	<b>92</b>



<b>6</b>	<b>1 1 0 0 0 0 1 0</b>	<b>82</b>
<b>7</b>	<b>1 1 1 1 1 0 0 0</b>	<b>F8</b>
<b>8</b>	<b>1 0 0 0 0 0 0 0</b>	<b>80</b>
<b>9</b>	<b>1 0 0 1 0 0 0 0</b>	<b>90</b>
<b>A</b>	<b>1 0 0 0 1 0 0 0</b>	<b>88</b>
<b>B</b>	<b>1 0 0 0 0 0 1 1</b>	<b>83</b>
<b>C</b>	<b>1 1 0 0 0 1 1 0</b>	<b>C6</b>
<b>D</b>	<b>1 0 1 0 0 0 0 1</b>	<b>A1</b>
<b>E</b>	<b>1 0 0 0 0 1 1 0</b>	<b>86</b>
<b>F</b>	<b>1 0 0 0 1 1 1 0</b>	<b>8E</b>
<b>-</b>	<b>1 0 1 1 1 1 1 1</b>	<b>BF</b>

## PHỤ LỤC 2

**Bảng mã hiển thị led 7 đoạn dành cho led 7 đoạn có Cathode chung  
(các led đơn sáng ở mức 1):**

<i>Số hiển thị trên led 7 đoạn</i>	<i>Mã hiển thị led 7 đoạn dạng nhị phân</i>	<i>Mã hiển thị led 7 đoạn dạng thập lục phân</i>
<b>0</b>	<b>0 0 1 1 1 1 1 1</b>	<b>3F</b>
<b>1</b>	<b>0 0 0 0 0 1 1 0</b>	<b>06</b>
<b>2</b>	<b>0 1 0 1 1 0 1 1</b>	<b>5B</b>
<b>3</b>	<b>0 1 0 0 1 1 1 1</b>	<b>4F</b>
<b>4</b>	<b>0 1 1 0 0 1 1 0</b>	<b>66</b>
<b>5</b>	<b>0 1 1 0 1 1 0 1</b>	<b>6D</b>
<b>6</b>	<b>0 1 1 1 1 1 0 1</b>	<b>7D</b>
<b>7</b>	<b>0 0 0 0 0 1 1 1</b>	<b>07</b>
<b>8</b>	<b>0 1 1 1 1 1 1 1</b>	<b>7F</b>

<b>9</b>	<b>01101111</b>	<b>6F</b>
<b>A</b>	<b>01110111</b>	<b>77</b>
<b>B</b>	<b>01111100</b>	<b>7C</b>
<b>C</b>	<b>00111001</b>	<b>39</b>
<b>D</b>	<b>01011110</b>	<b>5E</b>
<b>E</b>	<b>01111001</b>	<b>79</b>
<b>F</b>	<b>01110001</b>	<b>71</b>
<b>-</b>	<b>01000000</b>	<b>40</b>