

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



**ISO 9001:2008**

**THIẾT KẾ XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN VÀ**  
**GIÁM SÁT MẠCH NẠP AC QUY TỰ ĐỘNG SỬ DỤNG**  
**VI ĐIỀU KHIỂN AVR, ĐI SÂU THIẾT KẾ PHẦN MỀM**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**  
**NGÀNH ĐIỆN CÔNG NGHIỆP**

**HẢI PHÒNG - 2016**

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



**ISO 9001:2008**

**THIẾT KẾ XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN VÀ  
GIÁM SÁT MẠCH NẠP AC QUY TỰ ĐỘNG SỬ DỤNG  
VI ĐIỀU KHIỂN AVR, ĐI SÂU THIẾT KẾ PHẦN MỀM**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY  
NGÀNH ĐIỆN CÔNG NGHIỆP**

Sinh viên: Nguyễn Mạnh Hùng

Người hướng dẫn: T.S Nguyễn Trọng Thắng

**HẢI PHÒNG - 2016**

Cộng hoà xã hội chủ nghĩa Việt Nam  
**Độc lập – Tự Do – Hạnh Phúc**  
-----oO-----  
BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

## **NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên: Nguyễn Mạnh Hùng      Mã sv: 1513102014  
Lớp: ĐCL901      Ngành Điện Tự động công nghiệp  
Tên đề tài: Thiết kế xây dựng hệ thống điều khiển và giám sát mạch nạp acquy tự động sử dụng vi điều khiển AVR, đi sâu thiết kế phần mềm.

## NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp: .....

.....

## CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên : Nguyễn Trọng Thắng

Học hàm, học vị : Tiến Sĩ

Cơ quan công tác : Trường Đại học dân lập Hải Phòng

Nội dung hướng dẫn : Toàn bộ đề án

Người hướng dẫn thứ hai:

Họ và tên :

Học hàm, học vị :

Cơ quan công tác :

Nội dung hướng dẫn :

Đề tài tốt nghiệp được giao ngày ..... tháng ..... năm 2016.

Yêu cầu phải hoàn thành xong trước ngày..... tháng.....năm 2016

Đã nhận nhiệm vụ Đ.T.T.N

Sinh viên

Nguyễn Mạnh Hùng

Đã giao nhiệm vụ Đ.T.T.N

Cán bộ hướng dẫn Đ.T.T.N

T.S Nguyễn Trọng Thắng

Hải Phòng, ngày.....tháng.....năm 2016

HIỆU TRƯỞNG

GS.TS. NGUYỄN TRẦN HỮU NGHỊ

## PHÂN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần, thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp.

.....  
.....  
.....

2. Đánh giá chất lượng của Đ.T.T.N. (*so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N trên các mặt lý luận thực tiễn, tính toán giá trị sử dụng, chất lượng các bản vẽ ...*)

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

3. Cho điểm của cán bộ hướng dẫn:

*(Điểm ghi bằng số và chữ)*

Ngày .... tháng ..... năm 2016

Cán bộ hướng dẫn chính

*(Ký và ghi rõ họ tên)*



# MỤC LỤC

<b>LỜI MỞ ĐẦU</b> .....	1
<b>CHƯƠNG 1. GIỚI THIỆU VỀ CÔNG NGHỆ NẠP ẮC QUY</b> .....	2
<b>1.1. CẤU TRÚC CỦA MỘT BÌNH ẮC QUY</b> .....	2
<b>1.2. QUÁ TRÌNH BIẾN ĐỔI NĂNG LƯỢNG</b> .....	4
<b>1.3. PHÂN LOẠI ẮC QUY</b> .....	6
<b>1.4. CÁC ĐẶC TÍNH CƠ BẢN CỦA ẮC QUY</b> .....	7
1.4.1. Sức điện động của ắc quy.....	7
1.4.2. Dung lượng phóng của ắc quy .....	7
1.4.3. Dung lượng nạp của ắc quy.....	7
1.4.4 Đặc tính phóng của ắc quy. ....	8
1.4.5. Đặc tính nạp của ắc quy .....	9
<b>1.5. CÁC THÔNG SỐ CƠ BẢN CỦA ẮC QUY</b> .....	10
1.5.1. Dung lượng.....	10
1.5.2. Điện áp. ....	10
1.5.3. Điện trở trong. ....	10
<b>1.6. CÁC PHƯƠNG PHÁP NẠP ẮC QUY.</b> .....	11
1.6.1. Nạp với dòng điện không đổi.....	11
1.6.2. Nạp với điện áp không đổi .....	11
1.6.3. Phương pháp nạp dòng áp.....	12
<b>CHƯƠNG 2. TÍNH TOÁN THIẾT KẾ MẠCH CÔNG SUẤT</b> .....	14
<b>2.1. CẤU TRÚC MẠCH NẠP ẮC QUY</b> .....	14
<b>2.2. CHỌN MẠCH CHỈNH LƯU</b> .....	14
2.2.1. Chỉnh lưu cầu ba pha đối xứng dùng thyristor.....	14
2.2.2 Mạch chỉnh lưu cầu 3 pha dùng diot.....	18
<b>2.3 TÍNH CHỌN TIRISTOR</b> .....	20
<b>2.4 TÍNH TOÁN MÁY BIẾN ÁP</b> .....	21



<b>CHƯƠNG 3: XÂY DỰNG VÀ THIẾT KẾ BỘ ĐIỀU KHIỂN.....</b>	<b>24</b>
<b>3.1 VI ĐIỀU KHIỂN AVR .....</b>	<b>24</b>
<b>3.2 PHẦN CỨNG HỆ THỐNG.....</b>	<b>25</b>
<b>3.3 PHẦN MỀM HỆ THỐNG.....</b>	<b>30</b>
3.3.1 PHÂN MỀM MẠCH ĐIỀU KHIỂN.....	30
3.3.2. PHÂN MỀM MẠCH GIÁM SÁT .....	41
<b>KẾT LUẬN .....</b>	<b>59</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>60</b>

## LỜI MỞ ĐẦU

Đất nước ta đang bước trên con đường công nghiệp hoá và hiện đại hóa. Là một nước đang phát triển và đang dần tiếp cận với khoa học kỹ thuật hiện đại thì nhu cầu tự động hóa trong quá trình sản xuất ngày một được đề cao. Ngày nay trong công nghiệp, các mạch điều khiển được kỹ thuật số với các chương trình phần mềm đơn giản, linh hoạt và dễ dàng thay đổi được cấu trúc tham số hoặc các luật điều khiển. Nó làm tăng tốc độ xử lý, tính tác động nhanh và có độ chính xác cao dẫn đến nâng cao độ chuẩn hoá các hệ thống truyền động điện và các bộ điều khiển tự động. Trong xu thế đó thì việc áp dụng vào mạch nạp ắc quy tự động đang được sử dụng rộng rãi và có những đặc tính rất ưu việt. Như chúng ta đã biết thì ắc quy là thiết bị cấp nguồn một chiều được sử dụng phổ biến trong công nghiệp cũng như các lĩnh vực khác. Chính vì vậy việc nghiên cứu, chế tạo ắc quy và nguồn nạp ắc quy là hết sức cần thiết, nó ảnh hưởng rất lớn tới dung lượng và độ bền của ắc quy.

Từ yêu cầu về tính cấp thiết của đề tài hướng tới mục đích là thiết kế bộ nạp ắc quy tự động. Theo đó người sử dụng có thể hoàn toàn tự động hóa quá trình nạp ắc quy, cụ thể hơn là ắc quy sẽ có thể tự động được nạp khi điện áp thấp hơn so với yêu cầu hay có thể tự động ngắt mạch nạp khi đã đủ điện. Cùng với đó thì các thông số của quá trình nạp như điện áp, dòng điện cũng được điều khiển tự động để đảm bảo an toàn cho quá trình nạp cũng như tăng độ bền cho ắc quy.

Vì lý do đó, em đã chọn đề tài tốt nghiệp :“***Thiết kế xây dựng hệ thống điều khiển và giám sát mạch nạp ắc quy tự động sử dụng vi điều khiển AVR, đi sâu thiết kế phần mềm***” do thầy giáo TS Nguyễn Trọng Thắng hướng dẫn.

Đề án gồm 4 chương:

Chương 1. Giới thiệu về công nghệ nạp Ắc quy

Chương 2. Tính toán thiết kế mạch công suất

Chương 3: Xây dựng và thiết kế bộ điều khiển

# CHƯƠNG 1. GIỚI THIỆU VỀ CÔNG NGHỆ NẠP ẮC QUY

## 1.1. CẤU TRÚC CỦA MỘT BÌNH ẮC QUY

Ắc quy là nguồn điện hoá, sức điện động của ắc quy phụ thuộc vào vật liệu cấu tạo bản cực và chất điện phân. Với ắc quy chì axit sức điện động của một ắc quy đơn là 2,1V. Muốn tăng khả năng dự trữ năng lượng của ắc quy người ta phải tăng số lượng các cặp bản cực dương và âm trong mỗi ắc quy đơn. Để tăng giá trị sức điện động của nguồn người ta ghép nối nhiều ắc quy đơn thành một bình ắc quy. Bình ắc quy được làm từ số những tế bào (cell) đặt trong một vỏ bọc bằng cao su cứng hay nhựa cứng. Những đơn vị cơ bản của mỗi tế bào là những bản cực dương và bản cực âm. Những bản cực này có những vật liệu hoạt hoá nằm trong các tấm lưới phẳng. Bản cực âm là chì xốp sau khi nạp có màu xám. Bản cực dương sau khi nạp là  $PbO_2$  có màu nâu.

Cấu trúc của một ắc quy đơn gồm có: phân khối bản cực dương, phân khối bản cực âm, các tấm ngăn. Phân khối bản cực do các bản cực cùng tên ghép lại với nhau. Cấu tạo của một bản cực trong ắc quy gồm có phần khung xương và chất tác dụng trát lên nó. Khung xương của bản cực dương và âm có cấu tạo giống nhau. Chúng được đúc từ chì có pha thêm (5 ÷ 8%) Sb và tạo hình dạng mặt lưới. Phụ gia Sb thêm vào chì sẽ làm tăng thêm độ dẫn điện và cải thiện tính đúc. Trong thành phần của chất tác dụng còn có thêm khoảng 3% chất nở (các muối hữu cơ) để tăng độ xốp, độ bền của lớp chất tác dụng. Nhờ tăng độ xốp, dung dịch điện phân dễ thấm sâu vào trong lòng bản cực, đồng thời diện tích thực tế tham gia phản ứng hoá học của các bản cực cũng được tăng thêm.

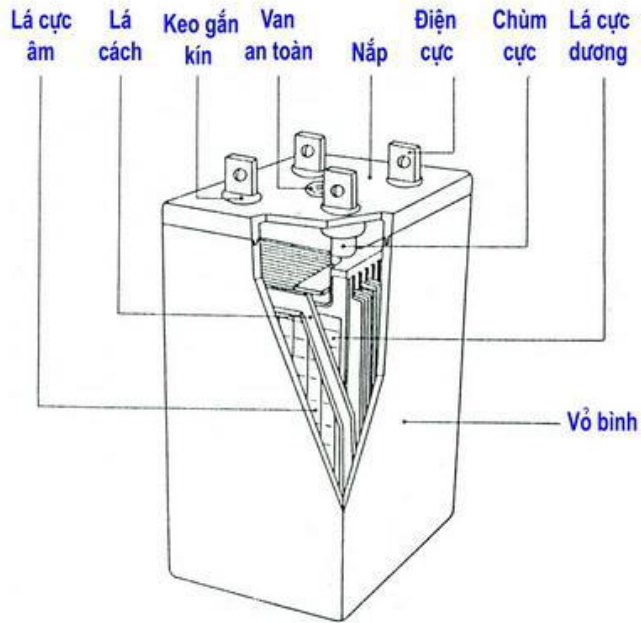
Phần đầu mỗi bản cực có vấu, các bản cực dương của mỗi ắc quy đơn được hàn với nhau tạo thành phân khối bản cực dương. Các bản cực âm hàn với nhau tạo thành phân khối bản cực âm. Số lượng các cặp bản cực trong mỗi ắc quy đơn thường từ 5 đến 8 cặp. Bề dày tấm bản cực dương của các ắc quy trước đây thường khoảng 2mm. Ngày nay với các công nghệ tiên tiến đã giảm xuống còn ( 1,3 ÷ 1,5 ) mm. Bản cực âm thường mỏng hơn ( 0,2 ÷ 0,3 ) mm. Số bản cực âm trong ắc quy đơn nhiều hơn số bản cực dương một bản nhằm tận dụng

triệt để diện tích tham gia phản ứng của các bản cực dương, do đó bản cực âm nằm ra bên ngoài nhóm bản cực. Tấm ngăn được bố trí giữa bản cực âm và bản cực dương là một tấm ngăn xếp có tác dụng ngăn cách và tránh va đập giữa các bản cực. Những tấm ngăn xếp cho phép dung dịch chất điện phân đi quanh các bản cực vì trên bề mặt của nó có lỗ. Tấm ngăn làm bằng vật liệu pôliclovinyl có bề dày ( 0,8 ÷ 1,2 ) mm và có dạng lượn sóng. Một bộ những sắp xếp như vậy gọi là một phần tử. Sau khi đã sắp xếp một bộ phận như trên, nó được đặt vào một ngăn trong vỏ bình ắc quy. Ở bình ắc quy có nắp đậy mềm, các nắp đậy tế bào được đặt lên sau đó những phiến nối được hàn vào để nối các cực liên tiếp của tế bào. Trong cách nối này các tế bào được nối liên tiếp. Cuối cùng nắp đậy bình ắc quy được hàn vào. Bình ắc quy có nắp đậy cứng , có một nắp đậy chung làm giảm được sự ăn mòn trên vỏ bình. Những bình ắc quy này có bản nối cực đi xuyên qua tấm ngăn cách từng tế bào. Tấm ngăn cách không cho dung dịch điện phân qua lại các tế bào. Điều này làm bình ắc quy vận hành tốt hơn vì bản nối ngắn và được đậy kín. Đầu nối chính của ắc quy là cọc dương và cọc âm. Cọc dương lớn hơn cọc âm để tránh nhầm điện cực. Người ta thường nối dây màu đỏ với cực dương và dây màu đen với cực âm. Dây cực âm được nối với lốc máy hay bộ phận kim loại. Dây cực dương được nối với bộ phận khởi động. Nắp thông hơi được đặt trên nắp mỗi tế bào. Những nắp này có hai mục đích:

Để đậy kín tế bào ắc quy, khi cần kiểm tra nước hay cho thêm nước thì ta sẽ mở nắp đậy này.

Khi nạp bình người ta cũng mở nắp đậy để chất khí hình thành có lối thoát ra. Mỗi tế bào ắc quy có điện thế khoảng 2 vôn. Ắc quy 6V có 3 tế bào mắc nối tiếp. Ắc quy 12V có 6 tế bào mắc nối tiếp. Muốn có điện thế cao hơn người ta mắc nối tiếp các bình ắc quy với nhau. Hai ắc quy 12V mắc nối tiếp sẽ tạo ra một hệ thống 24V.

Nồng độ dung dịch điện phân  $H_2SO_4$  là  $\gamma = 1,1 \div 1,3$  g/cm<sup>3</sup>. Nồng độ dung dịch điện phân có ảnh hưởng lớn đến sức điện động của ắc quy.



**Hình 1.1.** Cấu trúc bình ắc quy



**Hình 1.2.** Bình ắc quy trong thực tế

## 1.2. QUÁ TRÌNH BIẾN ĐỔI NĂNG LƯỢNG

Bình ắc quy là bình chứa năng lượng cho hệ thống điện. Khi cần bình ắc quy sẽ tạo ra dòng điện một chiều đi qua các thiết bị nối với các cực của nó. Dòng điện trong bình ắc quy tạo ra do phản ứng hoá học hoặc giữa những vật liệu trên bản cực và axit  $H_2SO_4$  trong bình hay còn gọi là chất điện giải. Sau một thời gian sử dụng bình ắc quy bị hết điện. Tuy nhiên nó có thể được nạp lại bằng cách cho một dòng điện bên ngoài đi qua nó theo chiều ngược với chiều phát điện của bình.

Trong điều kiện bình thường ắc quy được nạp do dòng điện từ máy phát điện. Để hoạt động tốt bình phải làm ba việc:

Cung cấp dòng điện khởi động động cơ.

Cung cấp điện khi hệ thống cần có mức điện lớn hơn hệ thống sạc có thể cung cấp.

Ổn định điện thế trong khi máy đang hoạt động. Ắc quy là nguồn năng lượng có tính thuận nghịch. Nó tích trữ năng lượng dưới dạng hoá năng và giải phóng năng lượng dưới dạng điện năng. Quá trình ắc quy cung cấp điện cho mạch ngoài gọi là quá trình phóng điện. Quá trình ắc quy được dự trữ năng lượng gọi là quá trình nạp điện.

**Bảng 1.1.** Quá trình biến đổi trên các điện cực

Trạng thái ắc quy	Bản cực dương	Dung dịch điện phân	Bản cực âm
Nạp đầy	PbO <sub>2</sub>	H <sub>2</sub> SO <sub>4</sub>	Pb
↓↑	↓↑	↓↑	↓↑
phóng điện hết	PbSO <sub>4</sub>	H <sub>2</sub> O	PbSO <sub>4</sub>

Trong quá trình phóng nạp, nồng độ dung dịch điện phân của ắc quy thay đổi. Khi ắc quy phóng điện, nồng độ dung dịch điện phân giảm dần. Khi được nạp điện, nồng độ dung dịch điện phân tăng dần. Do đó ta có thể căn cứ vào nồng độ dung dịch điện phân để đánh giá trạng thái tích điện của ắc quy.

**Bảng 1.2.** Tỷ trọng chất điện phân của bình ắc quy

Loại bình ắc quy	Tỷ trọng chất điện phân
Bình ắc quy làm việc ở chế độ tải nặng ví dụ các xe tải điện công nghiệp lớn.	1.275
Bình ắc quy dùng cho xe ô tô, phi cơ.	1.260
Bình ắc quy dùng cho tải không nặng lắm: ví dụ chiếu sáng tàu điện, khởi động các động cơ...	1.245
Bình ắc quy tĩnh, hoặc dùng cho các ứng dụng dự phòng.	1.215

### 1.3. PHÂN LOẠI ẮC QUY

Cho đến nay có rất nhiều loại ắc quy khác nhau được sản xuất tùy thuộc vào những điều kiện yêu cầu cụ thể của từng loại máy móc, dụng cụ, điều kiện làm việc. Cũng như những tính năng kinh tế kỹ thuật của ắc quy có thể liệt kê một số loại sau:

- Ắc quy chì (ắc quy axit)
- Ắc quy kiềm
- Ắc quy không lamen và ắc quy kiềm
- Ắc quy kẽm-bạc

Tuy nhiên trên thực tế ắc quy axit và ắc quy kiềm được sử dụng nhiều hơn.

**Bảng 1.3.** So sánh ắc quy kiềm và ắc quy axit

Ắc quy axit	Ắc quy kiềm
<ul style="list-style-type: none"><li>- Khả năng quá tải không cao, dòng nạp lớn nhất đạt được khi quá tải là <math>I_{max} = 20\% Q_{10}</math></li><li>- Hiện tượng tự phóng lớn, ắc quy nhanh hết điện ngay cả khi không sử dụng.</li><li>- Sử dụng rộng rãi trong đời sống công nghiệp, ở những nơi có nhiệt độ cao và đập lớn nhưng đòi hỏi công suất và quá tải vừa phải.</li><li>- Dùng trong xe máy, ô tô, các động cơ máy nổ công suất vừa và nhỏ.</li><li>- Giá thành thấp.</li><li>- Tuổi thọ thấp.</li></ul>	<ul style="list-style-type: none"><li>- Khả năng quá tải rất lớn, dòng điện áp nạp lớn nhất khi đó có thể đạt tới <math>50\% Q_{10}</math></li><li>- Hiện tượng tự phóng nhỏ</li><li>- Với khả năng trên thì ắc quy kiềm thường được sử dụng ở những nơi yêu cầu công suất cao và quá tải thường xuyên.</li><li>- Dùng trong công nghiệp hàng không, hàng hải và quốc phòng.</li><li>- Giá thành cao.</li><li>- Tuổi thọ cao.</li></ul>

Nhưng thông dụng nhất từ trước đến nay vẫn là ắc quy axit. Vì so với ắc quy kiềm nó có một vài tính năng tốt hơn như: sức điện động của mỗi bản "cặp bản" cực cao hơn, có điện trở trong nhỏ. Vì vậy, trong đồ án này ta chọn loại ắc quy axit để nghiên cứu và thiết kế.

## 1.4. CÁC ĐẶC TÍNH CƠ BẢN CỦA ẮC QUY

### 1.4.1. Sức điện động của ắc quy

Sức điện động của ắc quy chì axit phụ thuộc vào nồng độ dung dịch điện phân:

$$E_0 = 0,85 + \gamma \text{ (V)}$$

Trong đó:  $E_0$  là sức điện động tĩnh của ắc quy đơn, tính bằng V,  $\gamma$  là nồng độ dung dịch điện phân ở nhiệt độ 150°C tính bằng g/cm<sup>3</sup>.

Trong quá trình phóng điện, sức điện động của ắc quy được tính bằng công thức:

$$E_p = U_p + I_p \cdot r_{aq}$$

Trong đó:  $E_p$  : là sức điện động của ắc quy phóng điện,  $U_p$  : là điện áp đo trên các cực của ắc quy khi phóng điện,  $I_p$  : là dòng điện phóng,  $r_{aq}$  : là điện trở trong của ắc quy khi phóng điện. Sức điện động  $E_n$  của ắc quy được tính như sau:

$$E_n = U_n - I_n \cdot r_{aq}$$

Trong đó:  $E_n$ : sức điện động của ắc quy nạp điện,  $I_n$ : dòng điện nạp,  $U_n$ : điện áp đo trên các cực của ắc quy khi nạp điện,  $r_{aq}$ : điện trở trong của ắc quy khi nạp điện.

### 1.4.2. Dung lượng phóng của ắc quy

Dung lượng phóng của ắc quy là đại lượng đánh giá khả năng cung cấp năng lượng của ắc quy cho phụ tải, được tính theo công thức:

$$C_p = I_p \cdot t_p$$

Trong đó :  $C_p$  : dung lượng thu được trong quá trình phóng điện, tính bằng Ah,  $I_p$  : dòng điện phóng ổn định trong thời gian phóng điện  $t_p$

### 1.4.3. Dung lượng nạp của ắc quy

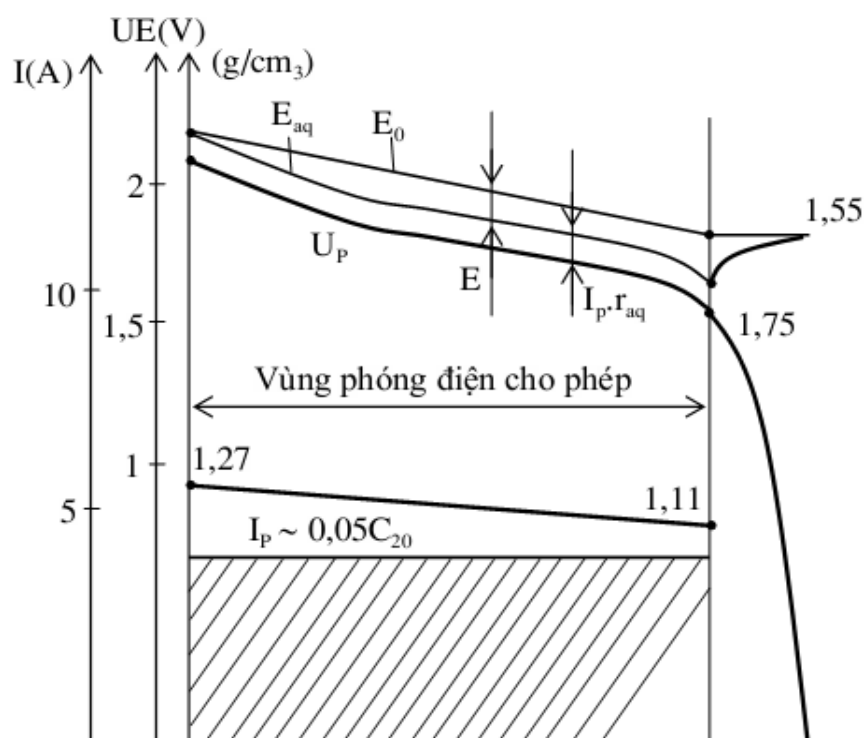
Dung lượng nạp của ắc quy là đại lượng đánh giá khả năng tích trữ năng lượng của ắc quy, được tính theo công thức:

$$C_n = I_n \cdot t_n$$

Trong đó:  $C_n$  - dung lượng thu được trong quá trình nạp điện, tính bằng Ah  $I_n$  - dòng điện nạp ổn định trong thời gian nạp điện  $t_n$



### 1.4.4 Đặc tính phóng của ắc quy.



**Hình 1.3.** Đặc tính phóng của ắc quy

Đặc tính phóng của ắc quy là đồ thị biểu diễn quan hệ phụ thuộc của sức điện động, điện áp ắc quy và nồng độ dung dịch điện phân theo thời gian phóng khi dòng điện phóng không thay đổi.

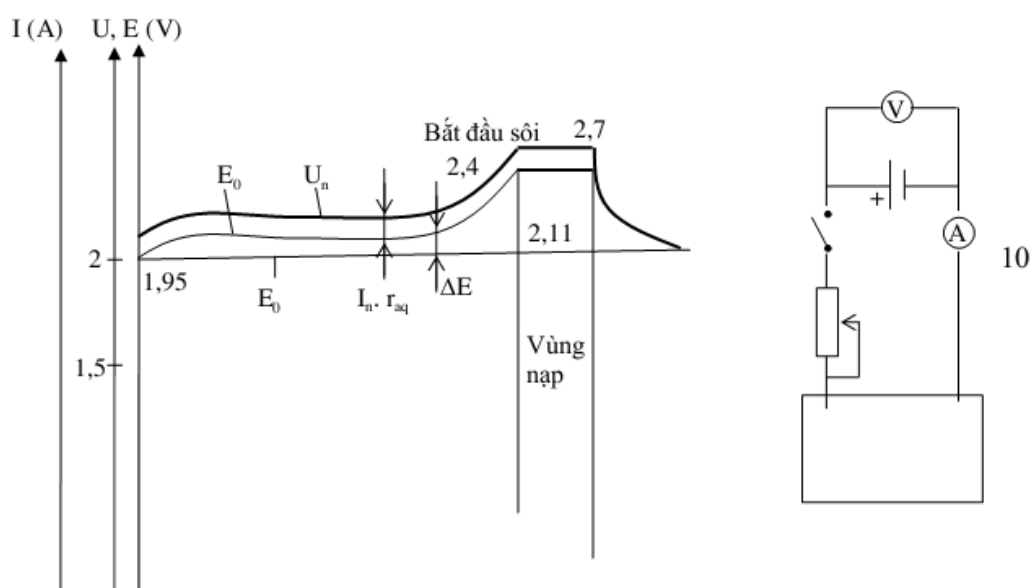
Từ đồ thị ta có nhận xét: Trong khoảng thời gian phóng từ  $t_p = 0$  đến  $t_p = t_{gh}$  sức điện động, điện áp, nồng độ dung dịch điện phân giảm dần. Tuy nhiên trong khoảng thời gian này độ dốc của các đồ thị không lớn, ta gọi đó là giai đoạn phóng ổn định hay thời gian cho phép tương ứng với mỗi chế độ phóng điện (dòng điện) của ắc quy.

Từ thời điểm  $t_{gh}$  trở đi độ dốc các đồ thị thay đổi đột ngột. Nếu tiếp tục cho ắc quy phóng điện sau  $t_{gh}$  thì sức điện động, điện áp của ắc quy sẽ giảm rất nhanh. Mặt khác các tinh thể Sunfat chì ( $PbSO_4$ ) tạo thành trong phản ứng sẽ có dạng thô, rắn rất khó hoà tan (biến đổi hoá học) trong quá trình nạp điện trở lại cho ắc quy sau này. Thời điểm  $t_{gh}$  gọi là giới hạn phóng điện cho phép của ắc quy, các giá trị  $E_p$ ,  $U_p$ ,  $\gamma$  tại  $t_{gh}$  gọi là các giá trị giới hạn phóng điện cho ắc quy.

Sau khi đã ngắt mạch phóng một khoảng thời gian, các giá trị sức điện động, điện áp của ắc quy, nồng độ dung dịch điện phân lại tăng lên, ta gọi đây là thời gian hồi phục hay khoảng nghỉ của ắc quy. Thời gian phục hồi này phụ thuộc vào chế độ phóng điện của ắc quy. Để đánh giá khả năng cung cấp điện của ắc quy có cùng điện áp danh nghĩa, người ta quy định so sánh dung lượng phóng điện thu được của các ắc quy khi tiến hành thí nghiệm ở chế độ phóng điện cho phép là 20h. Dung lượng phóng trong trường hợp này được kí hiệu là C20.

Thời gian phóng điện cho phép, các giá trị giới hạn phóng điện của ắc quy phụ thuộc vào dòng điện phóng. Sự phụ thuộc của dung lượng phóng vào dòng điện phóng của ắc quy có dung lượng phóng định mức C20 (dung lượng phóng thu được ở chế độ 20h) là 60Ah.

#### 1.4.5. Đặc tính nạp của ắc quy



**Hình 1.4.** Sơ đồ đặc tính nạp

Đặc tính nạp của ắc quy là đồ thị biểu diễn quan hệ phụ thuộc của sức điện động, điện áp ắc quy và nồng độ dung dịch điện phân theo thời gian nạp khi trị số dòng điện nạp không thay đổi. Từ đồ thị đặc tính nạp ta có nhận xét: Trong khoảng thời gian nạp từ 0 đến  $t = t_s$ , sức điện động, điện áp, nồng độ dung dịch điện phân tăng dần. Tới thời điểm  $t_s$  trên bề mặt các bản cực âm xuất hiện các

bọt khí (còn gọi là hiện tượng sôi) lúc này hiệu điện thế giữa các cực của ắc quy đơn tăng tới giá trị 2,4V. Nếu vẫn tiếp tục nạp, giá trị này nhanh chóng tăng tới 2,7V và giữ nguyên. Thời gian này gọi là thời gian nạp no, có tác dụng làm cho phần các chất tác dụng ở sâu trong lòng các bản cực được biến đổi hoàn toàn, nhờ đó sẽ làm tăng thêm dung lượng phóng điện của ắc quy. Trong sử dụng thời gian nạp no cho ắc quy kéo dài từ ( 2 ÷ 3 ) h, trong suốt thời gian đó hiệu điện thế trên các cực của ắc quy và nồng độ dung dịch điện phân không thay đổi. Như vậy dung lượng thu được khi ắc quy phóng điện luôn nhỏ hơn dung lượng cần thiết để nạp no ắc quy. Sau khi ngắt mạch nạp, điện áp, sức điện động của ắc quy, nồng độ dung dịch điện phân giảm xuống và ổn định. Thời gian này cũng gọi là khoảng nghỉ của ắc quy sau khi nạp. Trị số dòng điện nạp ảnh hưởng rất lớn đến chất lượng và tuổi thọ của ắc quy.

Dòng điện nạp định mức đối với ắc quy qui định bằng  $0,05C_{20}$ .

## **1.5. CÁC THÔNG SỐ CƠ BẢN CỦA ẮC QUY**

### **1.5.1. Dung lượng.**

Là điện lượng của ắc quy đã được nạp đầy, rồi đem cho phóng điện liên tục với dòng điện phóng 1A tới khi điện áp của ắc quy giảm xuống đến trị số giới hạn quy định ở nhiệt độ quy định. Dung lượng của ắc quy được tính bằng ampe-giờ (Ah).

### **1.5.2. Điện áp.**

Tùy thuộc vào nồng độ chất điện phân và nguồn nạp cho ắc quy mà điện áp ở mỗi ngăn của ắc quy khi nó được nạp đầy sẽ đạt 2,6V đến 2,7V (để hở mạch), và khi ắc quy đã phóng điện hoàn toàn là 1,7V đến 1,8V.

Điện áp của ắc quy không phụ thuộc vào số lượng bản cực của ắc quy nhiều hay ít.

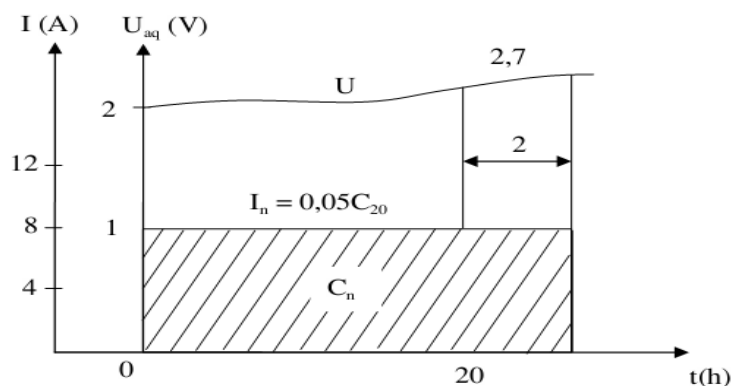
### **1.5.3. Điện trở trong.**

Là trị số điện trở bên trong của ắc quy, bao gồm điện trở các bản cực, điện trở dung dịch điện phân có xét đến sự ngăn cách của các tấm ngăn giữa các

bản cực. Thường thì trị số điện trở trong của ắc quy khi đã nạp đầy điện là  $(0,001-0,0015)\Omega$  và khi ắc quy đã phóng điện hoàn toàn là  $(0,02-0,025)\Omega$ .

## 1.6. CÁC PHƯƠNG PHÁP NẠP ẮC QUY.

### 1.6.1. Nạp với dòng điện không đổi

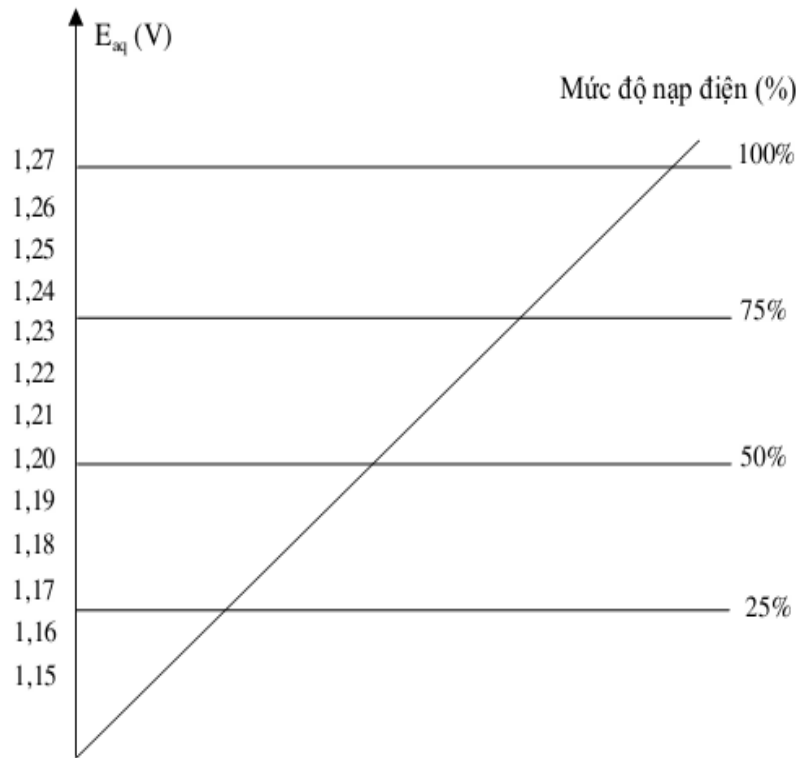


**Hình 1.5.** Sơ đồ đặc tính nạp với dòng điện không đổi

Nhược điểm của phương pháp nạp với dòng không đổi là thời gian nạp kéo dài và yêu cầu các ắc quy đưa vào nạp có cùng cỡ dung lượng định mức. Để khắc phục nhược điểm thời gian nạp kéo dài người ta sử dụng phương pháp nạp với dòng điện nạp thay đổi hai hay nhiều nấc. Trong trường hợp nạp hai nấc, dòng điện nạp ở nấc thứ nhất chọn bằng  $(0,3 \div 0,5) C_{20}$  và kết thúc nạp ở nấc một khi ắc quy bắt đầu sôi. Dòng điện nạp ở nấc thứ hai bằng  $0,05 C_{20}$ .

### 1.6.2. Nạp với điện áp không đổi

Phương pháp nạp với điện áp nạp không đổi yêu cầu các ắc quy được mắc song song với nguồn nạp. Hiệu điện thế của nguồn nạp không đổi và được tính bằng  $(2,3 \div 2,5) V$  cho một ngăn ắc quy đơn. Đây là phương pháp nạp điện cho ắc quy lắp trên ô tô. Phương pháp nạp với điện áp nạp không đổi có thời gian nạp ngắn, dòng điện nạp tự động giảm theo thời gian. Tuy nhiên dùng phương pháp này ắc quy không được nạp no, vậy nạp với điện áp không đổi chỉ là phương pháp nạp bổ sung cho ắc quy trong quá trình sử dụng.



**Hình 1.6.** Sơ đồ đặc tính nạp với điện áp nạp không đổi

### 1.6.3. Phương pháp nạp dòng áp

Đây là phương pháp tổng hợp của hai phương pháp trên. Nó tận dụng được những ưu điểm của mỗi phương pháp. Đối với yêu cầu của đề tài là nạp ắc qui tự động tức là trong quá trình nạp mọi quá trình biến đổi và chuyển hoá được tự động diễn ra theo một trình tự đã đặt sẵn thì ta chọn phương pháp nạp ắc qui là phương pháp dòng áp. Đối với ắc qui axit : Để đảm bảo cho thời gian nạp cũng như hiệu suất nạp thì trong khoảng thời gian  $t_n = 8$  giờ tương ứng với  $(75 \div 80)$  % dung lượng ắc qui ta nạp với dòng điện không đổi là  $I_n = 0,1 C_{10}$  . Vì theo đặc tính nạp của ắc qui trong đoạn nạp chính thì khi dòng điện không đổi thì điện áp, sức điện động tải ít thay đổi, do đó bảo đảm tính đồng đều về tải cho thiết bị nạp. Sau thời gian 8 giờ ắc qui bắt đầu sôi lúc đó ta chuyển sang nạp ở chế độ ổn áp. Khi thời gian nạp được 10 giờ thì ắc qui bắt đầu no, ta nạp bổ sung thêm 2-3 giờ.

Đối với ắc qui kiềm: Trình tự nạp cũng giống như ắc qui axit nhưng do khả năng quá tải của ắc qui kiềm lớn nên lúc ổn dòng ta có thể nạp với dòng nạp

$I_n = 0,2 C10$  hoặc nạp cường bức để tiết kiệm thời gian với dòng nạp  $I_n = 0,5 C10$

Các quá trình nạp ắc quy tự động kết thúc khi bị cắt nguồn nạp hoặc khi nạp ổn áp với điện áp bằng điện áp trên 2 cực của ắc quy, lúc đó dòng nạp sẽ từ từ giảm về không. Kết luận: Vì ắc quy là tải có tính chất dung kháng kèm theo sức phản điện động cho nên khi ắc quy đói mà ta nạp theo phương pháp điện áp thì dòng điện trong ắc quy sẽ tự động dâng lên không kiểm soát được sẽ làm sôi ắc quy dẫn đến hỏng hóc nhanh chóng. Vì vậy trong vùng nạp chính ta phải tìm cách ổn định dòng nạp trong ắc quy.

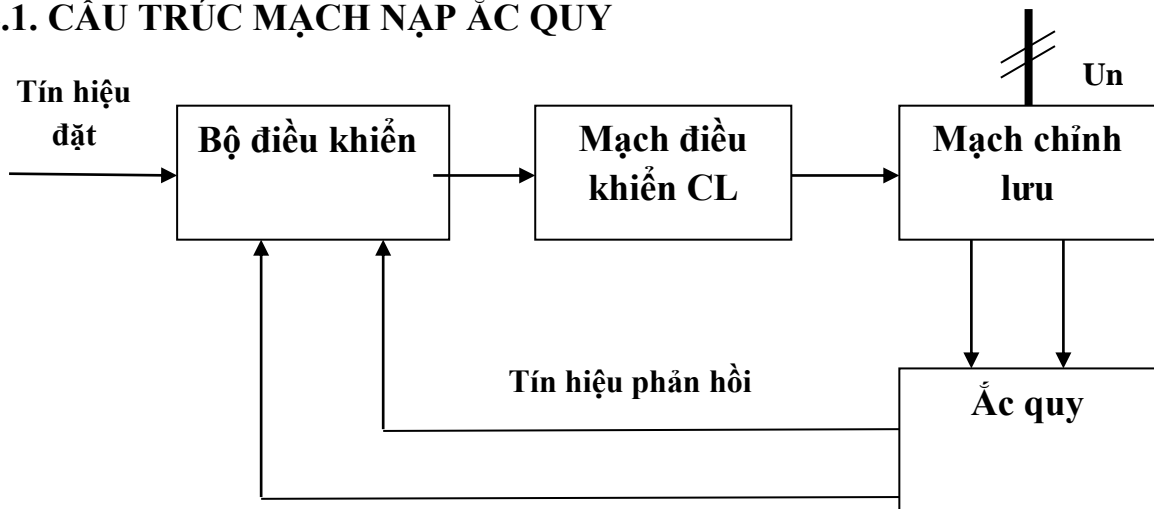
Khi dung lượng của ắc quy dâng lên đến 80% lúc đó nếu ta cứ tiếp tục giữ ổn định dòng nạp thì ắc quy sẽ sôi và làm cạn nước. Do đó đến giai đoạn này ta lại phải chuyển chế độ nạp cho ắc quy sang chế độ ổn áp. Chế độ ổn áp được giữ cho đến khi ắc quy đã thực sự no. Khi điện áp trên các bản cực của ắc quy bằng điện áp nạp thì lúc đó dòng nạp sẽ tự động giảm về không, kết thúc quá trình nạp. Tùy theo loại ắc quy mà ta nạp với dòng điện nạp khác nhau :

Ắc quy axit: dòng nạp  $I_n = 0,1 C10$ . Nạp cường bức với dòng điện nạp  $I_n = 0,2 C10$ .

\* Ắc quy kiềm dòng nạp  $I_n = 0,2 C10$ . Nạp cường bức  $I_n = 0,5 C10$ .

## CHƯƠNG 2. TÍNH TOÁN THIẾT KẾ MẠCH CÔNG SUẤT

### 2.1. CẤU TRÚC MẠCH NẠP ẮC QUY

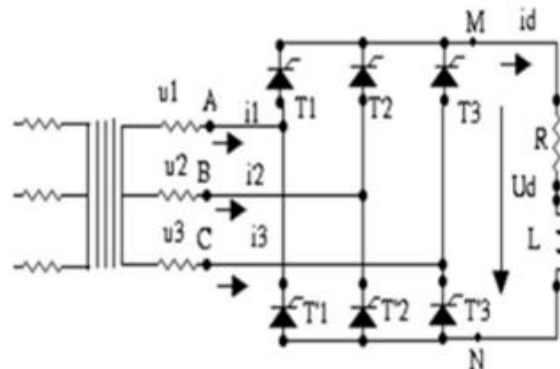


**Hình 2.1:** Cấu trúc mạch nạp ắc quy

Cấu trúc mạch nạp ắc quy được thực hiện theo một mạch vòng khép kín như sau: bộ điều khiển cùng lúc nhận được hai tín hiệu đó là tín hiệu đặt (do người thiết kế yêu cầu) cùng với tín hiệu phản hồi về các trạng thái ắc quy (tín hiệu dòng, điện áp nạp), hai tín hiệu này sẽ được so sánh và tùy thuộc vào kết quả bộ điều khiển sẽ xử lý và phát tín hiệu điều khiển tới mạch điều khiển chỉnh lưu để trực tiếp điều khiển các van công suất. Qua đó điện áp và dòng nạp vào ắc quy được điều khiển hoàn toàn tự động. Quá trình lặp đi lặp lại như vậy cho tới khi bộ nạp ngừng hoạt động.

### 2.2. CHỌN MẠCH CHỈNH LƯU

#### 2.2.1. Chỉnh lưu cầu ba pha đối xứng dùng thyristor



**Hình 2.2:** Mạch chỉnh lưu cầu 3 pha đối xứng

a) Sơ đồ nguyên lý

- Mạch chỉnh lưu cầu 3 pha đối xứng dùng thyristor có thể làm việc ở 2 chế độ

- + Chế độ chỉnh lưu: Khi góc mở chậm của Thyristor  $\alpha < \pi/2$ .

- + Chế độ nghịch lưu: Khi góc mở chậm của Thyristor  $\alpha < \pi/2$  và phụ

tải chứa sức điện động E' cùng chiều với dòng điện phụ tải

- Trong một chu kì của điện áp xoay chiều, điện áp Ud sẽ hình thành từ 6 đoạn điện áp dây của nguồn xoay chiều. Điện áp trung bình nhận được trên tải, điện áp ngược lớn nhất trên van và dòng trung bình chảy qua các van

- **Ưu điểm :**

- + Số xung áp chỉnh lưu trong một chu kì lớn

- + Giá trị trung bình của dòng điện chạy qua mỗi van trong một chu kì thấp, chỉ bằng 1/3 dòng chỉnh lưu.

- + Do sơ đồ là đối xứng nên không làm việc lệch pha lưới điện

- + Sơ đồ có thể làm việc chế độ nghịch lưu

- **Nhược điểm:**

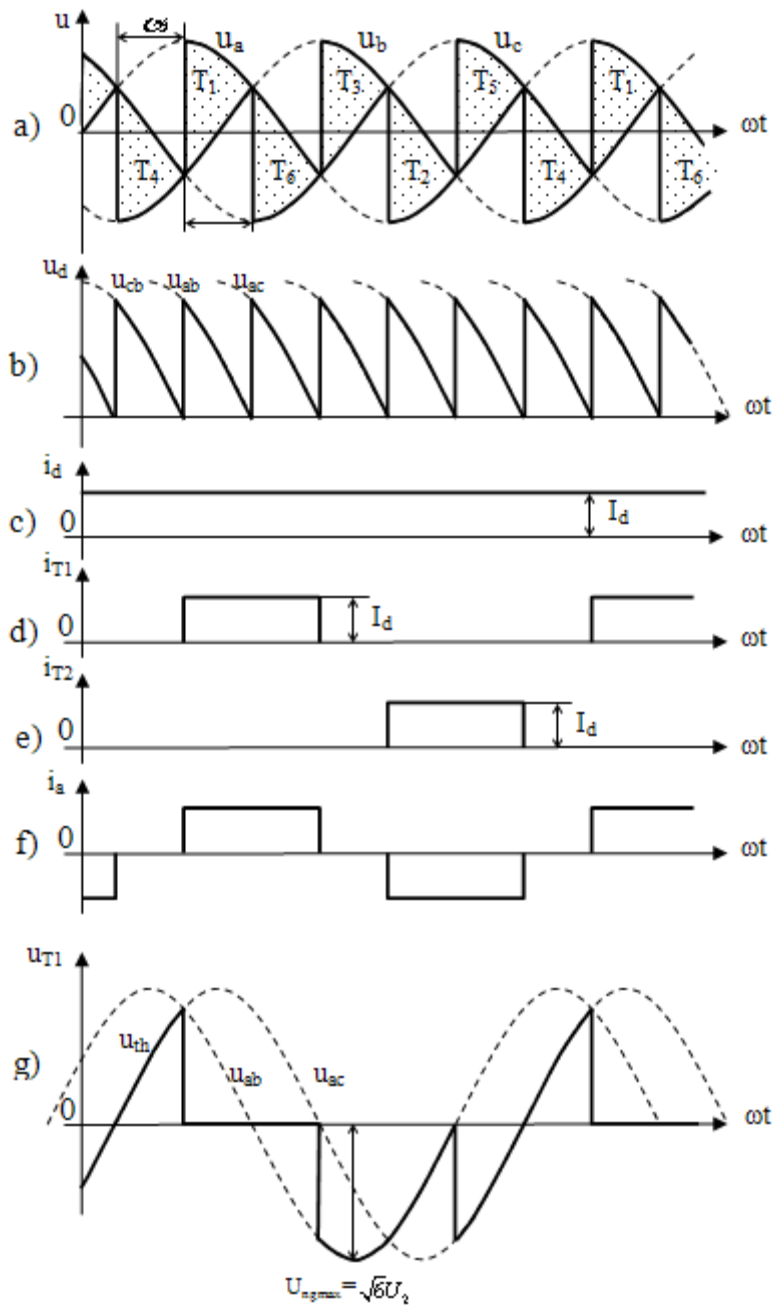
- + Giá thành thiết bị cao, sử dụng số van lớn

- + Điều khiển phức tạp đối với các cơ cấu trong mạch điều khiển

b) Nguyên lý hoạt động

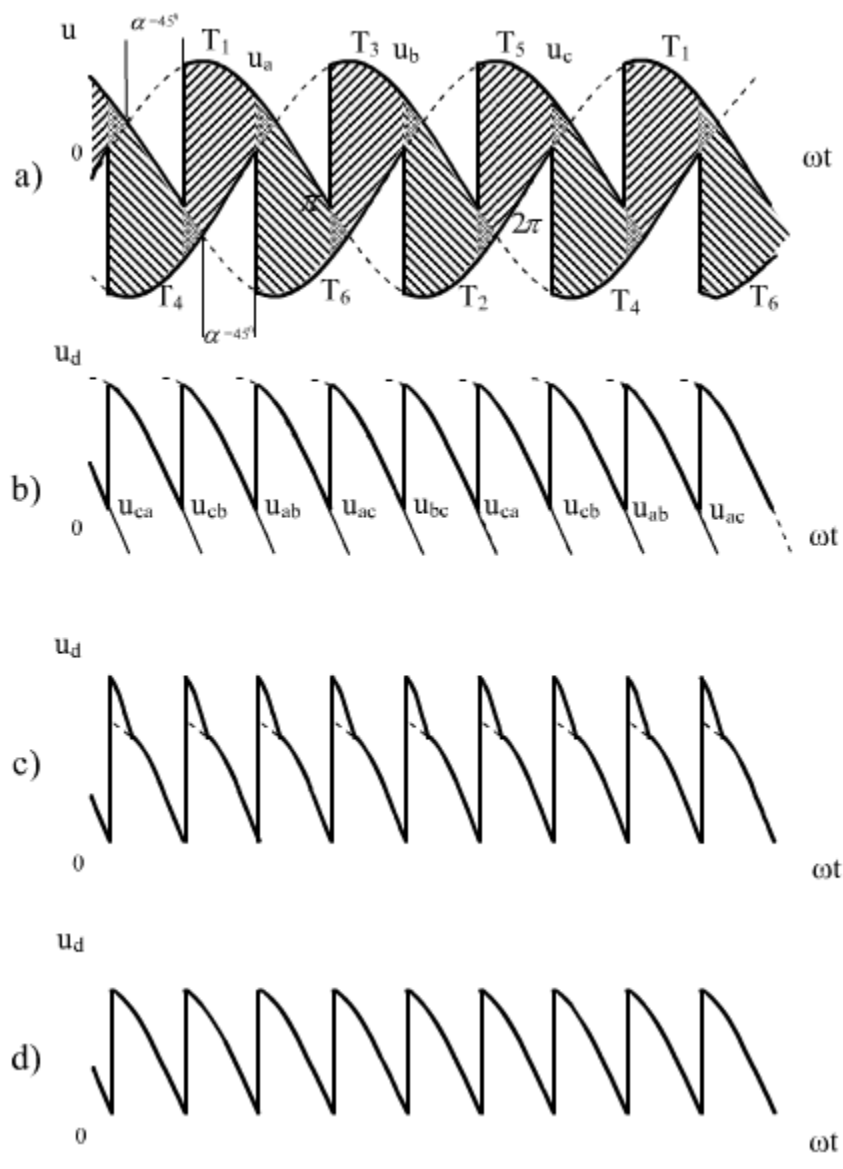
- Nguyên lý hoạt động của mạch chỉnh lưu cầu ba pha dùng thyristo ứng với  $\alpha=60^0$ , Tải điện cảm được thể hiện như hình 2.3:





**Hình 2.3:** Nguyên lý hoạt động và dạng điện áp trên tải khi  $\alpha=60^\circ$

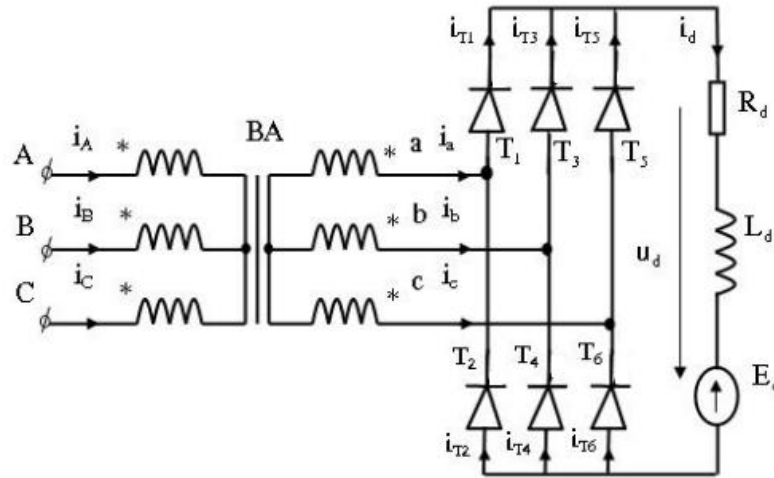
- Nguyên lý hoạt động của mạch chỉnh lưu cầu ba pha dùng thyristo ứng với  $\alpha=45^\circ$ , tải điện cảm được thể hiện như hình 2.4:



**Hình 2.4:** Nguyên lý hoạt động và dạng điện áp trên tải khi  $\alpha=45^\circ$

## 2.2.2 Mạch chỉnh lưu cầu 3 pha dùng diot

a) Sơ đồ mạch



**Hình 2.5:** Sơ đồ mạch chỉnh lưu cầu 3 pha dùng diot

Trong một chu kì của điện áp xoay chiều, điện áp  $U_d$  sẽ hình thành từ 6 đoạn điện áp dây của nguồn xoay chiều. Điện áp trung bình nhận được trên tải, điện áp ngược lớn nhất trên van và dòng trung bình chảy qua các van.

- Ưu điểm:
  - + Số xung áp chỉnh lưu trong một chu kì lớn
  - + Giá trị trung bình của dòng điện chạy qua mỗi van trong một chu kì thấp, chỉ bằng  $1/3$  dòng chỉnh lưu.
- Nhược điểm :
  - + Sử dụng số van lớn
  - + Dùng diot nên không thể điều khiển
  - + Không thể làm việc ở chế độ nghịch lưu



- Qua phân tích các ưu nhược điểm của các sơ đồ chỉnh lưu và yêu cầu của công nghệ, em quyết định chọn sơ đồ mạch chỉnh lưu điều khiển cầu 3 pha dùng thyristo.
- Ưu điểm :
  - + Số xung áp chỉnh lưu trong một chu kì lớn
  - + Giá trị trung bình của dòng điện chạy qua mỗi van trong một chu kì thấp, chỉ bằng 1/3 dòng chỉnh lưu.
  - + Do sơ đồ là đối xứng nên không làm việc lệch pha lưới điện
  - + Sơ đồ có thể làm việc chế độ nghịch lưu

### 2.3 TÍNH CHỌN TIRISTOR

Tính chọn thyristor dựa vào các yếu tố cơ bản dòng tải, sơ đồ đã chọn, điều kiện tải nhiệt, điện áp làm việc. Ta tính toán chọn thiết bị cho mạch nạp ac quy 24V với dòng nạp là 10A, các thông số cơ bản của van được tính như dưới đây:

Điện áp ngược lớn nhất mà thyristor phải chịu:

$$U_{nmax} = K_{nu} \cdot U_2 = K_{nu} \cdot U_d / K_u$$

Trong đó:

- +  $U_d, U_2, U_{nmax}$ : Điện áp tải, nguồn xoay chiều, ngược của van.
- +  $K_{nu}, K_u$ : Các hệ số điện áp ngược, điện áp tải:

$$K_{nu} = \sqrt{2}; K_u = 2\sqrt{2} / \pi$$

$$U_{nmax} = \pi \cdot 24 / 2 = 37,68V$$

Điện áp ngược của van cần chọn:

$$U_{nv} = K_{dtu} \cdot U_{nmax}$$

Trong đó:

$K_{dtu}$ : Hệ số dự trữ điện áp, chọn  $K_{dtu} = 1,8$

$$U_{nv} = 1,8 \cdot 37,68 = 67,82 V$$

Dòng điện làm việc của van được tính theo dòng hiệu dụng:

$$I_h = I_{hd} = K_{hd} \cdot I_d$$

Trong đó:  $K_{hd}$ : Hệ số dòng điện hiệu dụng cầu 1 pha.  $K_{hd} = 1/\sqrt{2} = 0,71$

$$I_h = 10 \cdot 0,71 = 7,1A$$

Chọn điều kiện làm việc của van là có cánh tản nhiệt và đủ diện tích tản nhiệt, không có quạt đối lưu không khí, với điều kiện đó dòng điện định mức của van cần chọn:

$$I_{dmv} = K_i \cdot I_h = 1,4 \cdot 7,1 = 9,94A$$

Với  $K_i$  là hệ số dự trữ dòng điện, chọn  $K_i = 1,4$

Từ các thông số  $U_{nv}$ ,  $I_{dmv}$  ta chọn 2 thyristor loại 5P4M có các thông số sau:

$$U_n = 80V \quad \text{Dòng điện ngược cực đại của van}$$

$$I_{dm} = 13A \quad \text{Dòng điện định mức của van}$$

$$I_{pk} = 250A \quad \text{Điểm xung dòng điện}$$

$$I_{dk} = 0,04A \quad \text{Dòng điện của xung điều khiển}$$

$$U_{gmax} = 3V \quad \text{Điện áp xung điều khiển}$$

$$I_r = 0,01A \quad \text{Dòng điện rò}$$

$$\Delta u = 2,1V \quad \text{Sụt áp lớn nhất của thyristor ở trạng thái dẫn}$$

$$T_{max} = 125^{\circ}C \quad \text{Nhiệt độ làm việc cực đại cho phép.}$$

Chọn 2 Diôt S6460P-G có các thông số như sau:

$$\text{Dòng điện chỉnh lưu cực đại: } I_{max} = 30A.$$

$$\text{Điện áp cực của Diôt: } U_{ng} = 500V.$$

$$\text{Tổn hao điện áp ở trạng thái mở của Diot: } \Delta U = 1,2V.$$

## 2.4 TÍNH TOÁN MÁY BIẾN ÁP

Điện áp pha sơ cấp máy biến áp:

$$U_1 = 220V$$

Tính điện áp pha thứ cấp máy biến áp như sau:

Phương trình cân bằng điện áp khi có tải:

$$U_{do} \cdot \cos \alpha_{min} = U_d + 2\Delta U + \Delta U_{dm} + \Delta U_{ba}$$

Trong đó:

$$U_d: \text{ Điện áp chỉnh lưu, } U_d = 24V$$

$$\Delta U_v: \text{ Sụt áp trên các van, } \Delta U_v = 2,1$$

$\Delta U_{ba}$ : Sụt áp trên các máy biến áp khi có tải bao gồm sụt áp trên điện trở và sụt áp trên các điện cảm. Lấy sơ bộ:

$$\Delta U_{ba} = 6\% \cdot U_d = 0,06 \cdot 24 = 1,44 \text{ V}$$

$$\alpha_{\min} = 0^\circ.$$

Từ phương trình của điện áp khi có tải ta có:

$$U_{do} = \frac{U_d + 2 \cdot \Delta U_v + \Delta U_{dm} + \Delta U_{ba}}{\cos \alpha_{\min}}$$

$$U_{do} = \frac{24 + 2 \cdot 2,1 + 0 + 1,44}{1} = 27,65 \text{ V}$$

Vậy điện áp phía thứ cấp máy biến áp là:

$$U_\alpha = U_{do} / K_u = \pi \cdot 27,65 / 2 \sqrt{2} = 30,71 \text{ V}$$

Với  $K_u = 2\sqrt{2}$ : Hệ số điện áp chỉnh lưu





## CHƯƠNG 3: XÂY DỰNG VÀ THIẾT KẾ BỘ ĐIỀU KHIỂN

### 3.1 VI ĐIỀU KHIỂN AVR

AVR là một họ vi điều khiển do hãng Atmel sản xuất (Atmel cũng là nhà sản xuất dòng vi điều khiển 89C51 mà có thể chúng ta đã từng nghe đến). AVR là chip vi điều khiển 8 bits với cấu trúc tập lệnh đơn giản hóa-RISC(Reduced Instruction Set Computer), một kiểu cấu trúc đang thể hiện ưu thế trong các bộ xử lí.

So với các chip vi điều khiển 8 bits khác, AVR có nhiều đặc tính hơn hẳn, hơn cả trong tính ứng dụng (dễ sử dụng) và đặc biệt là về chức năng:

Gần như chúng ta không cần mắc thêm bất kỳ linh kiện phụ nào khi sử dụng AVR, thậm chí không cần nguồn tạo xung clock cho chip (thường là các khối thạch anh).

Thiết bị lập trình (mạch nạp) cho AVR rất đơn giản, có loại mạch nạp chỉ cần vài điện trở là có thể làm được. một số AVR còn hỗ trợ lập trình on – chip bằng bootloader không cần mạch nạp...

Bên cạnh lập trình bằng ASM, cấu trúc AVR được thiết kế tương thích C.

Nguồn tài nguyên về source code, tài liệu, application note...rất lớn trên internet.

Hầu hết các chip AVR có những tính năng (features) sau:

- Có thể sử dụng xung clock lên đến 16MHz, hoặc sử dụng xung clock nội lên đến 8 MHz (sai số 3%)

- Bộ nhớ chương trình Flash có thể lập trình lại rất nhiều lần và dung lượng lớn, có SRAM (Ram tĩnh) lớn, và đặc biệt có bộ nhớ lưu trữ lập trình được EEPROM.

- Nhiều ngõ vào ra (I/O PORT) 2 hướng (bi-directional).

8 bits, 16 bits timer/counter tích hợp PWM.

- Các bộ chuyển đổi Analog – Digital phân giải 10 bits, nhiều kênh.

- Chức năng Analog comparator.

- Giao diện nối tiếp USART (tương thích chuẩn nối tiếp RS-232).
- Giao diện nối tiếp Two –Wire –Serial (tương thích chuẩn I2C) Master và Slaver.
- Giao diện nối tiếp Serial Peripheral Interface (SPI).

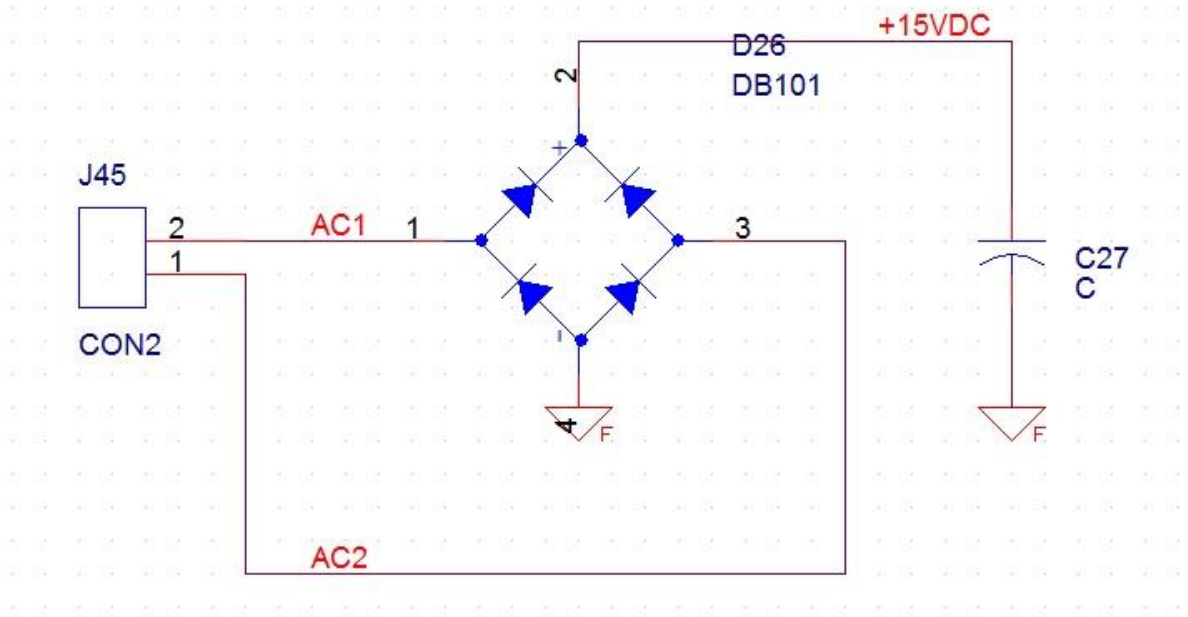
Một số chip AVR thông dụng:

- AT90S1200
- AT90S2313
- AT90S2323 and AT90S2343
- AT90S2333 and AT90S4433
- AT90S4414 and AT90S8515
- AT90S4434 and AT90S8535
- AT90C8534
- ATtiny10, ATtiny11 and ATtiny12
- ATtiny15
- ATtiny22
- ATtiny26
- ATtiny28
- ATmega8/8515/8535
- ATmega16

### **3.2PHẦN CỨNG HỆ THỐNG**

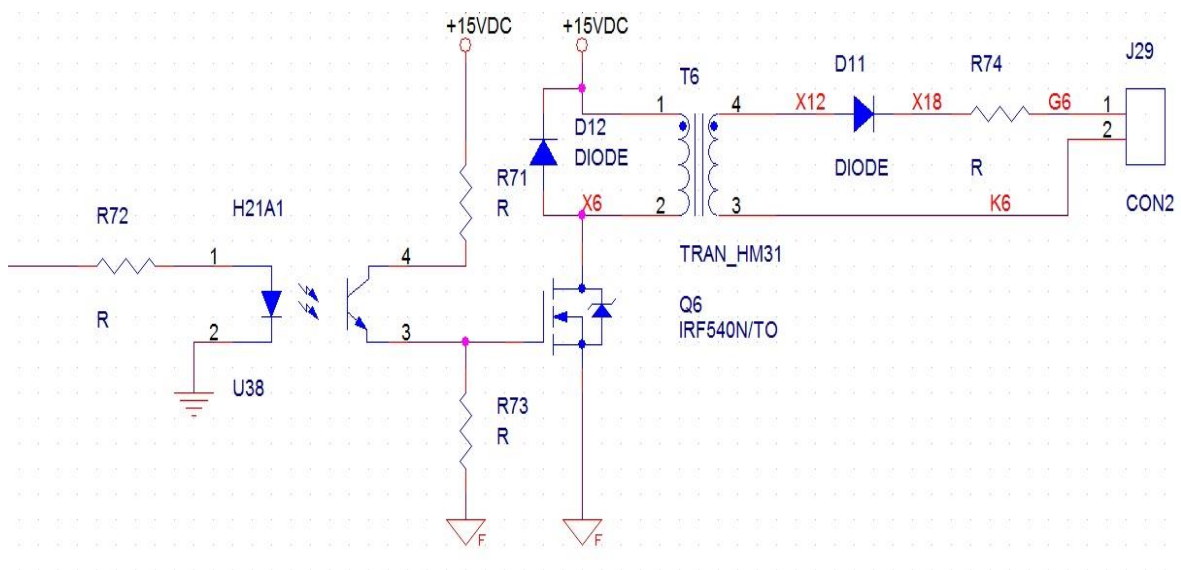
Trong hệ thống hệ phần cứng của chúng ta có khá nhiều các mạch điều khiển bao gồm các mạch sau:

Mạch chỉnh lưu từ điện áp xoay chiều sang một chiều để cấp nguồn cho hệ thống mạch điều khiển được thể hiện ở hình 3.1.



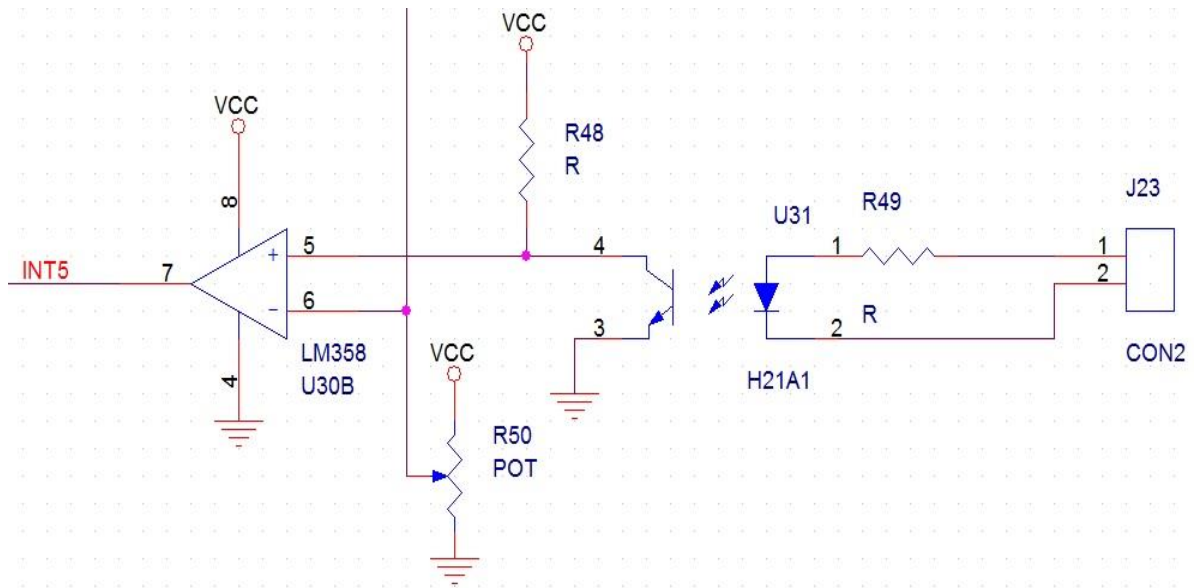
**Hình 3.1:** Mạch chỉnh lưu

Phần chi tiết mạch tạo xung để điều khiển thyristor được thể hiện ở hình 3.2



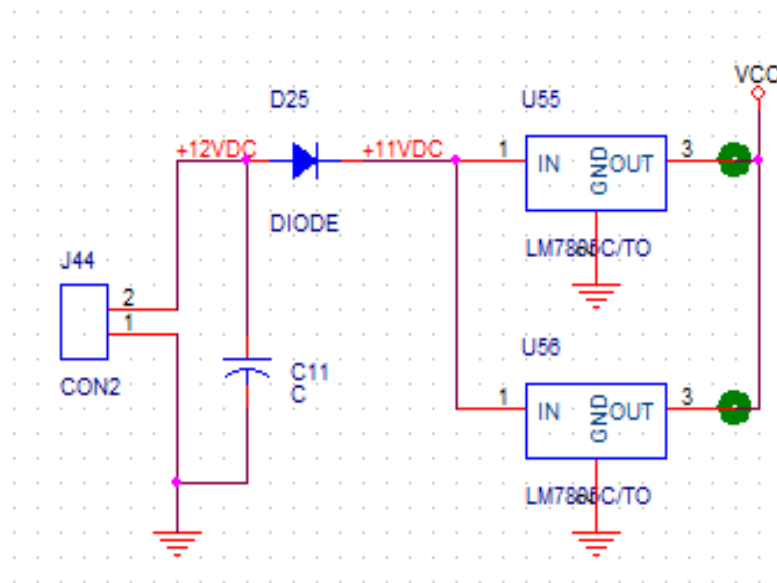
**Hình 3.2:** Mạch tạo xung điều khiển thyristor

Phần chi tiết mạch phát hiện thời điểm chuyển dấu điện áp lưới ( $\alpha=0^0$ ) được thể hiện ở hình 3.3. Tín hiệu này sẽ sinh ra ngắt đưa về chân INT5 của vi điều khiển AVR.



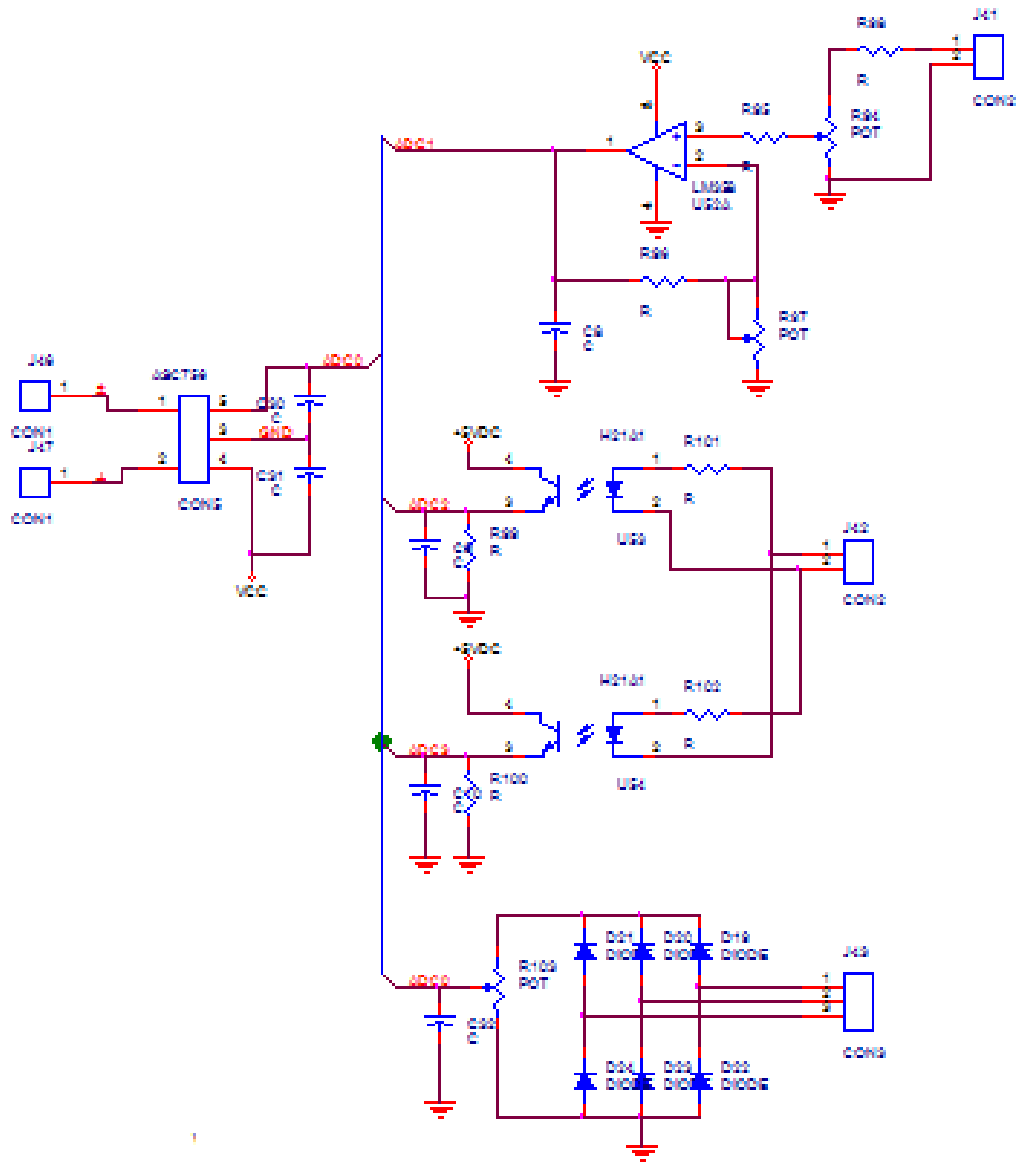
**Hình 3.3:** Mạch tạo ngắt phát hiện thời điểm  $\alpha=0^0$

Mạch nguồn có chức năng tạo điện áp 5V để cấp nguồn cho vi điều khiển và một số linh kiện được thể hiện ở hình 3.4.



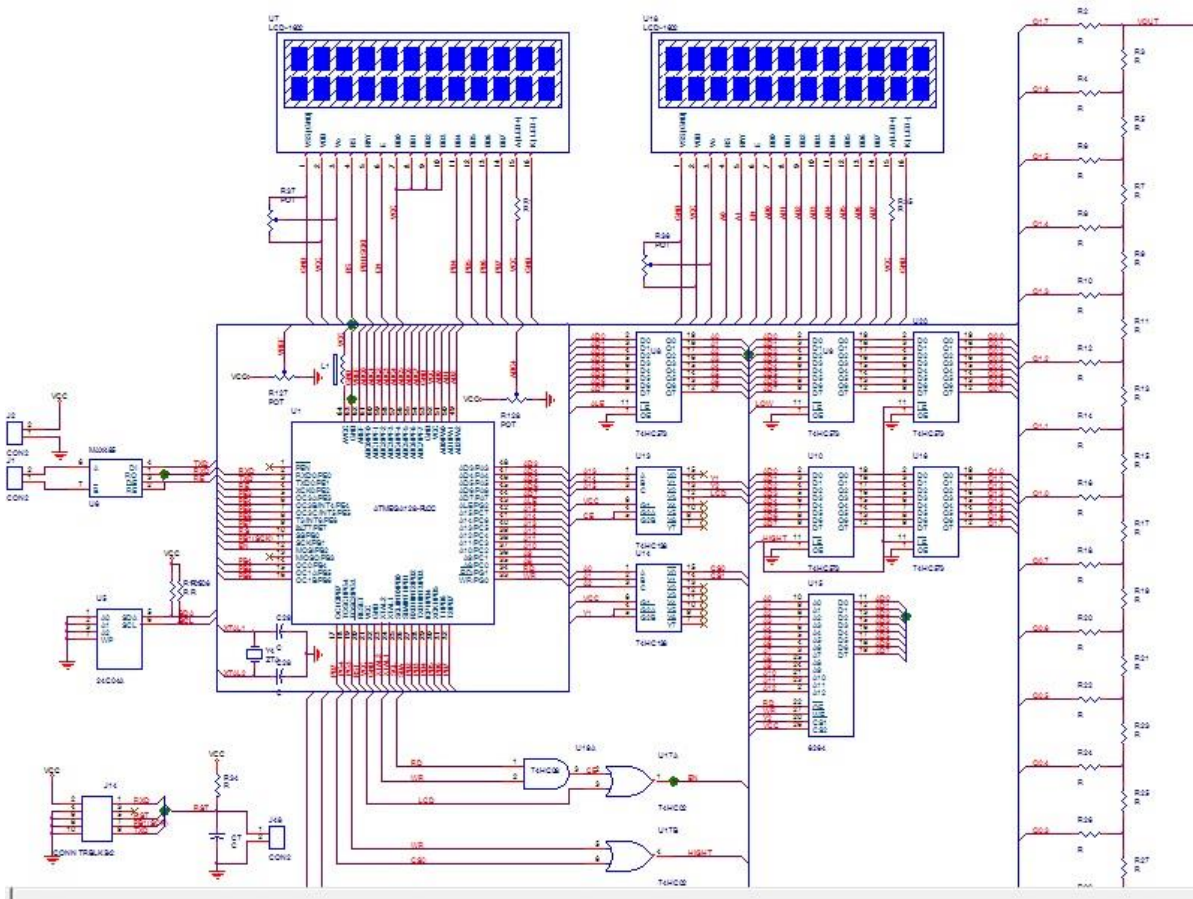
**Hình 3.4:** Mạch nguồn

Phần chi tiết mạch đo dòng điện nạp vào ac quy và điện áp của ac quy được thể hiện ở hình 3.5. Các tín hiệu đo này sẽ được đưa vào các chân ADC của vi điều khiển, từ đó vi điều khiển sẽ tính toán chuyển đổi sang các giá trị dòng điện và điện áp thực tế của ac quy.



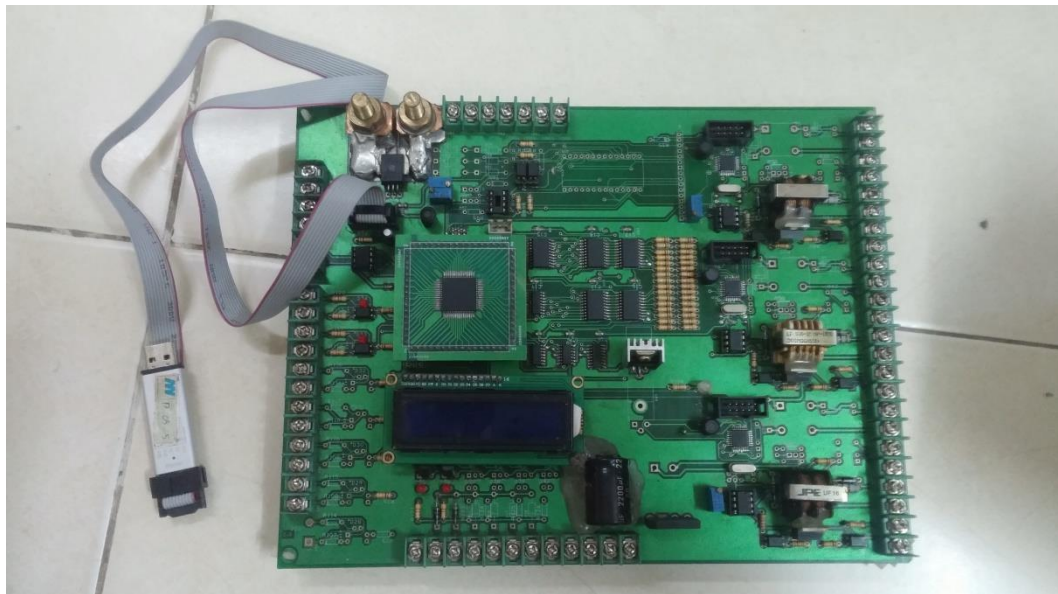
**Hình 3.5:** Mạch đo dòng và điện áp

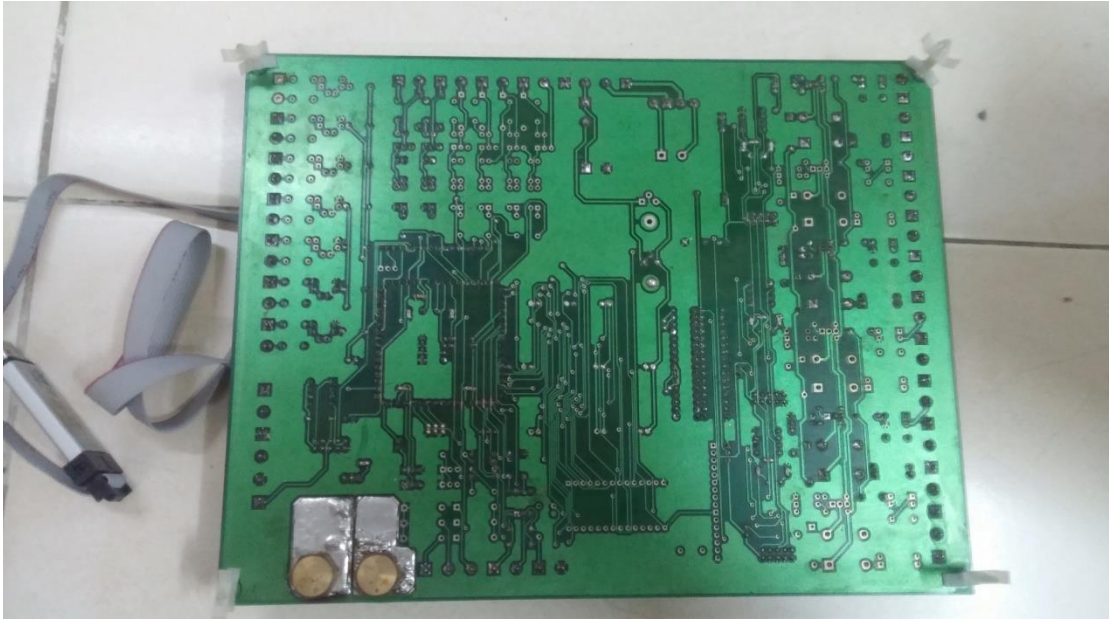
Các chi tiết phần mạch còn lại như vi điều khiển trung tâm, hệ thống hiển thị LCD... được hiển thị ở hình 3.6.



**Hình 3.6:** Mạch trung tâm và hiển thị

Trên cơ sở mạch nguyên lý đã thiết kế, tác giả thi công mạch thực tế, thể hiện ở các hình sau:





### 3.3 PHẦN MỀM HỆ THỐNG

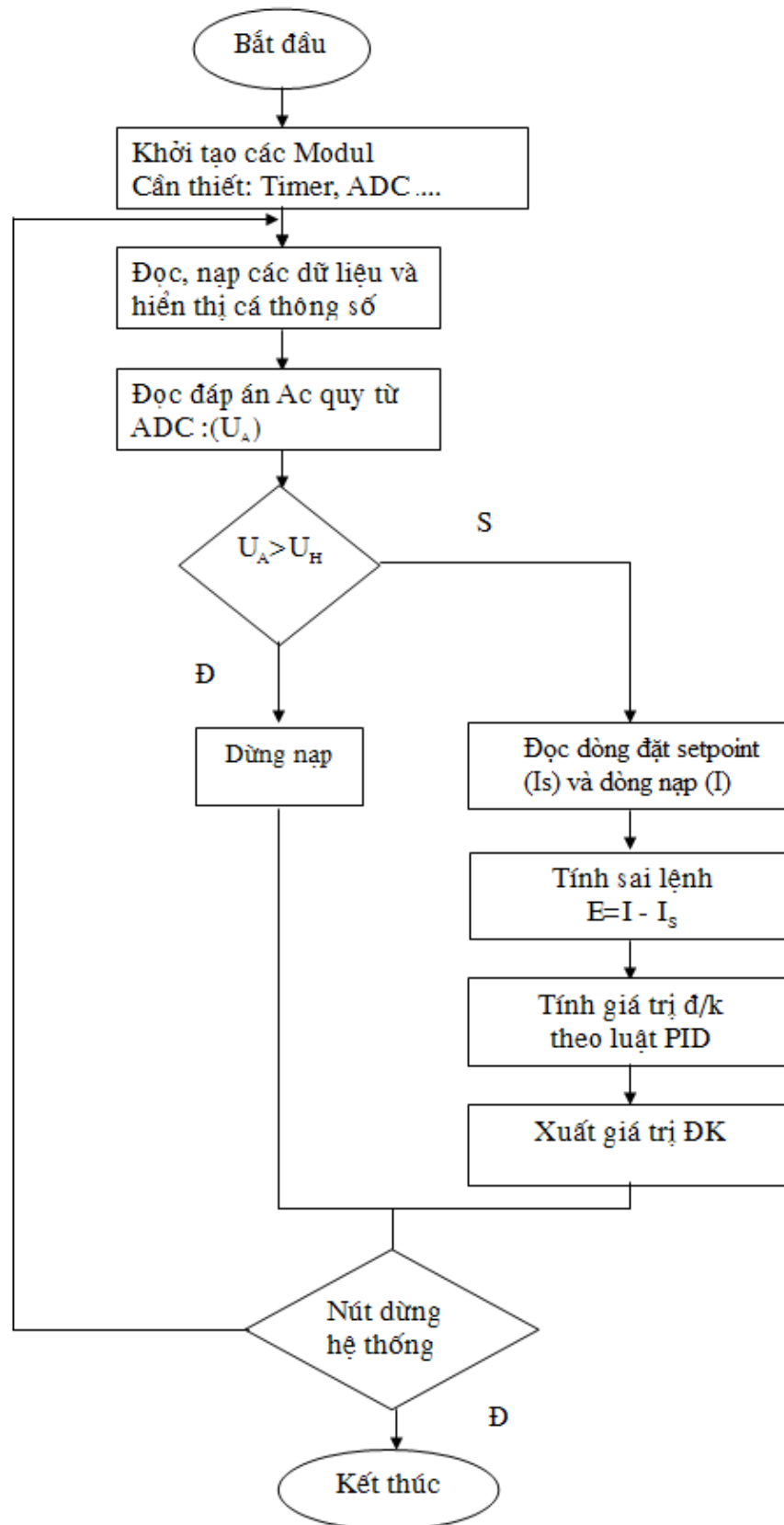
#### 3.3.1 PHẦN MỀM MẠCH ĐIỀU KHIỂN

- Chức năng chính của bộ điều khiển nhằm thực hiện một quy luật điều khiển nào đó được điều chỉnh bằng lượng đặt (phụ thuộc vào công nghệ ) và lượng phản hồi (phụ thuộc các tham số U, I, ... của tải ) để tạo ra xung điều khiển điều chỉnh góc  $\alpha$  để không chế năng lượng ra tải theo yêu cầu.

Đề cập hơn về chức năng của vi điều khiển chúng ta sẽ cùng quan sát lưu đồ thuật toán sau:

## Lưu đồ thuật toán

Hàm main (Mạch điều khiển)





- Bắt đầu khởi động vi điều khiển khởi tạo các modul và cài đặt các tham số cần thiết như Bộ đếm(Counter), Bộ thời gian (Timer), bộ biến đổi tương tự số (ADC)...

- Vi điều khiển đọc và nạp các dữ liệu và đưa ra để hiển thị các thông số, đọc điện áp ác quy từ mạch chỉnh lưu (ADC):  $U_a$

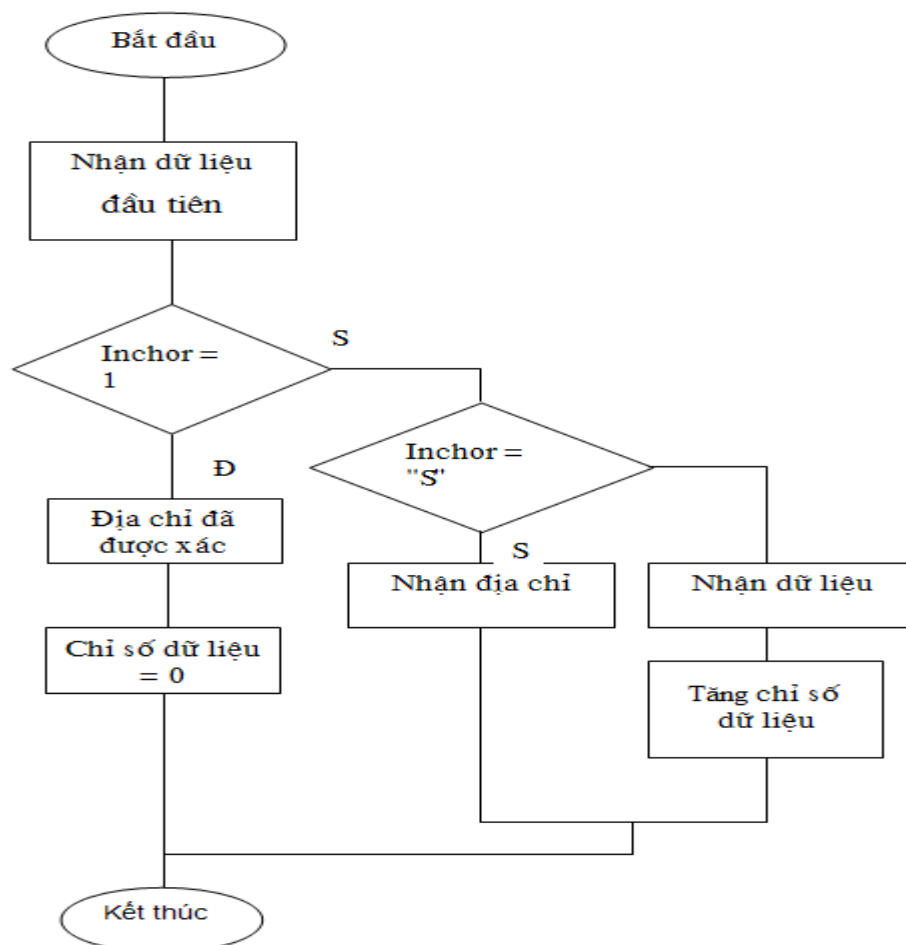
- Với điều kiện  $U_a > U_h$

+ Nếu đúng thì dừng nạp, nhân nút dừng hệ thống và kết thúc

+ Nếu sai, vi điều khiển sẽ đọc dòng đặt Set point( $I_s$ ) và dòng nạp thực tế, sau đó tính sai lệch điện áp  $E = I - I_s$  và tính giá trị điều khiển, đưa vào bộ điều khiển theo luật PID và bắt đầu xuất giá trị điều khiển để điều khiển xung mở thyristor nạp cho ác quy đến khi  $U_a > U_h$  thì dừng hệ thống.

### Hàm ngắt nhận dữ liệu:

Hàm này có chức năng nhận các thông số từ mạch giám sát để làm các tham số cho bộ điều khiển.



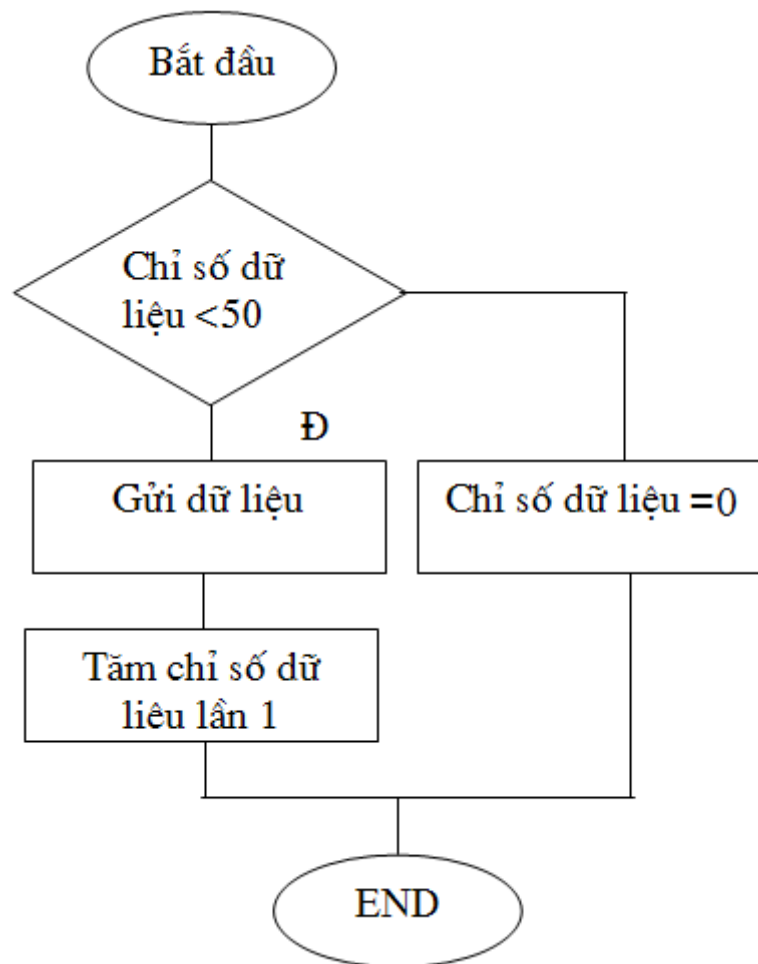
- Mạch điều khiển nhận những dữ liệu đầu tiên, nếu thông số Inchor =1 thì địa chỉ đã được xác định, gán chỉ số dữ liệu bằng 0 rồi kết thúc chương trình.

- Nếu Inchor=1 là sai thì so sánh Inchor='S' Nếu đúng thì nhận dữ liệu và tăng chỉ số của dữ liệu rồi kết thúc chương trình.

- Nếu Inchor='S' là sai thì nhân địa chỉ của nơi truyền dữ liệu và kết thúc chương trình.

### Hàm truyền dữ liệu:

Hàm có nhiệm vụ truyền các tham số của mạch điều khiển đến mạch giám sát



- Khi bắt mạch bắt đầu hoạt động, các modul so sánh chỉ số dữ liệu như sau:

+ Nếu Chỉ số dữ liệu < 50 thì bắt đầu gửi dữ liệu đi sau đó tăng dần dần chỉ số dữ liệu lần 1 rồi kết thúc

+Nếu Chỉ số dữ liệu >50 là sai thì gán chỉ số dữ liệu =0 rồi kết thúc chương trình

### Mã nguồn chương trình mạch điều khiển:

```
/*
Chip type      : ATmega128L
Program type   : Application
AVR Core Clock frequency: 11.059200 MHz
Memory model   : Small
External RAM size  : 65536
Ext. SRAM wait state : 0
Data Stack size  : 1024
*/
#include <mega128.h>
#include <mega128_bits.h>
#include <stdlib.h>
#asm
.equ __i2c_port=0x12 ;PORTD
.equ __sda_bit=1
.equ __scl_bit=0
#endasm
#include <i2c.h>
#asm
.equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>
#define scan_max 50
#define PID_MAX 1020
#define PID_MIN 0
#define ADC_VREF_TYPE 0x00
#define RE PORTE.2
#define out_low (*(unsigned char *) 0x2001)
#define out_hig (*(unsigned char *) 0x2000)
bit ADDR_OK;
bit SEND_EN;
char recv[60];
char send[60];
char recv_index;
char send_index;
char in_char;
unsigned int Curent,Vp,Vn;
int Visai;
bit PID_RUN,OV_Curent;
float sai_lech;
float Is,Im,Vh,VI;
float dong;
float dong_thuc;
float Ap;
float U;
float Ty_le;
float Tich_phan;
float Vi_phan;
float New_control;
float old_sai_lech;
float Kp,Ki,Kd;
unsigned int out;
```

```

unsigned long int Sum1,Sum2;
char scan;
unsigned char T[4];
char in_put;
bit charg;
unsigned int h;
float    VOLT[16]  ={
0,      // y0
2.33,   // y1
3.25,   // y2
5.28,   // y3
9.83,   // y4
19.03,  // y5
20.93,  // y6
22.58,  // y7
23.20,  // y8
25.53,  // y9
27.01,  // y10
28.00,  // y11
30.19,  // y12
32.94,  // y13
35.39,  // x14
40.00   // x15
};
float    ADC[16]  ={
0,      // x0
48,     // x1
72,     // x2
120,    // x3
220,    // x4
400,    // x5
440,    // x6
470,    // x7
486,    // x8
520,    // x9
644,    // x10
692,    // x11
713,    // x12
758,    // x13
760,    // y14
800     // y15
};
// USART0 Receiver interrupt service routine
interrupt [USART0_RXC] void usart0_rx_isr(void)
{
in_char=UDR0;
if (in_char==0x01)
{
ADDR_OK=1;
recv_index=0;
}
else
if ((in_char=='S')&&ADDR_OK)
{
RE=1;
SEND_EN=1;
ADDR_OK=0;    // cho phep truyen
for (h=0;h<500;h++)
{
}
}
}

```

```

UDR0=0x01;    // truyen ky dia chi
}
else
if(ADDR_OK)
{
recv[recv_index]=in_char;
recv_index++;
}

}
// USART0 Transmitter interrupt service routine
interrupt [USART0_TXC] void usart0_tx_isr(void)
{
if ((send_index<50)&&SEND_EN)
{
for (h=0;h<500;h++)
{
}
UDR0=send[send_index];
send_index++;
}
else
{
SEND_EN=0;
send_index=0;
RE=0;
}
}
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
char e;
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
ADCSRA|=0x10;
return ADCW;
}
float noi_suy_ap(float bien)
{
float out;
float x0,y0,x1,y1;
unsigned char j;
j=0;
for (j=0;j<=14;j++)
{
if ((ADC[j]<=bien)&&(bien<ADC[j+1]))
{
x0=ADC[j];
y0=VOLT[j];
x1=ADC[j+1];
y1=VOLT[j+1];
out=(y1-y0)/(x1-x0)*bien+y0-x0*(y1-y0)/(x1-x0);
break;
}
}
return(out);
}
void float_to_ascc(float so_thuc)
{
unsigned int phan_nguyen;
phan_nguyen=(unsigned int)(so_thuc*10);

```

```

T[0]= phan_nguyen/1000;
T[1]=(phan_nguyen-1000*T[0])/100;
T[2]=(phan_nguyen-1000*T[0]-100*T[1])/10;
T[3]=(phan_nguyen-1000*T[0]-100*T[1]-10*T[2]);
if (T[0]==0)
{
T[0]=' ';
if (T[1]==0)
{
T[1]=' ';
if (T[2]==0)
{
T[2]=' ';
}
else
{
T[2]=T[2]+48;
}
}
else
{
T[1]=T[1]+48;
T[2]=T[2]+48;
}
}
else
{
T[0]=T[0]+48;
T[1]=T[1]+48;
T[2]=T[2]+48;
}
T[3]=T[3]+48;
}
float ascc_to_float(char recv_addr)
{
float y;
char i,j;
char U1[6];
i=0;
for(j=recv_addr;j<recv_addr+6;j++)
{
if (recv[j]==0)
{
recv[j]='0';
}
U1[i]=recv[j];
i++;
}
y=atof(U1);
return(y);
}
void main(void)
{
PORTA=0x00;
DDRA=0x00;

PORTB=0x00;
DDRB=0x00;

PORTC=0x00;

```

```

DDRC=0x00;

PORTD=0x00;
DDRD=0xFC;
PORTE=0x00;
DDRE=0x04;
PORTF=0x00;
DDRF=0x00;
PORTG=0x00;
DDRG=0x00;
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;
MCUCR=0x80;
XMCRA=0x0C;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x00;
UCSR0A=0x00;
UCSR0B=0xD8;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x23;
UCSR1B=0x00;
ACSR=0x80;
SFIO=0x00;
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

```

```

SPCR=0x00;
TWCR=0x00;
i2c_init();
lcd_init(16);
#asm("sei")
OV_Curent=1;
send[10]='Y';
send[49]='S';
lcd_gotoxy(0,0);
lcd_putchar('I');
lcd_putchar('m');
lcd_putchar(':');
lcd_gotoxy(0,1);
lcd_putchar('I');
lcd_putchar('s');
lcd_putchar(':');
while (1)
{
// Place your code here
if (scan<scan_max)
{
Curent=read_adc(0); // doc tin hieu dong dien
Vp=read_adc(2); // doc tin hieu dien ap V+
Vn=read_adc(3); // doc tin hieu dien ap V-
if (Vp>Vn)
{
Visai=Vp-Vn;
}
else
{
Visai=Vn-Vp;
}
Sum1+= Visai;
Sum2+=Curent;
scan++;
}
else
{
Ap=Sum1/scan_max;
dong=Sum2/scan_max;
Sum1=0;
Sum2=0;
scan=0;
Im=ascc_to_float(18);
if(dong<508)
{
dong_thuc=(508-dong)*0.22;
}
else
{
dong_thuc=0;
}
in_put=PING&0x10;
if (in_put==0x10)
{
PID_RUN=1;
}
else
{
PID_RUN=0;
}
}

```



```

}
if(dong_thuc>=Im)
{
OV_Curent=0;
send[10]='F';
}
else
{
if (OV_Curent)
{
send[10]='Y';
}
}
Vh=ascc_to_float(0);
Vl=ascc_to_float(6);
U=noi_suy_ap(Ap);
if (U<Vl)
{
charg=1;
}
else
if(U>Vh)
{
charg=0;
}
else
if ((Vl<=U)|| (U<=Vh))
{
if (recv[43]=='T')
{
send[11]='K';
charg=1;
PID_RUN=1;
OV_Curent=1;
}
}
if ((charg==1)&&(OV_Curent==1)&&(PID_RUN==1))
{
Is=ascc_to_float(12);
Kp=ascc_to_float(24);
Ki=ascc_to_float(30);
Kd=ascc_to_float(36);
sai_lech=(Is-dong_thuc);
Ty_le=Kp*sai_lech;
Tich_phan+=Ki*(sai_lech+old_sai_lech);
Vi_phan=Kd*(sai_lech-old_sai_lech);
New_control+=(Ty_le+Tich_phan+Vi_phan);
if ((New_control<=PID_MAX)&&(New_control>=PID_MIN))
{
out=(unsigned int)(New_control*64);
out_low=(unsigned char)(out);
out_hig=(unsigned char)(out>>8);
}
else
if (New_control>=PID_MAX)
{
New_control--;
}
else
{

```

```

New_control++;
}
old_sai_lech=sai_lech;
}
else
{
New_control=0;
old_sai_lech=0;
out_low=0x00;
out_hig=0x00;
}
}
if (recv[42]=='R')
{
OV_Curent=1;
send[10]='O';
}
lcd_gotoxy(3,0);
float_to_ascc(U);
send[0]=T[0];
send[1]=T[1];
send[2]=T[2];
send[3]='.';
send[4]=T[3];
lcd_putchar(recv[18]);
lcd_putchar(recv[19]);
lcd_putchar(recv[20]);
lcd_putchar(recv[21]);
lcd_putchar(recv[22]);
lcd_putchar(recv[23]);
float_to_ascc(dong_thuc);
send[5]=T[0];
send[6]=T[1];
send[7]=T[2];
send[8]='.';
send[9]=T[3];
lcd_gotoxy(3,1);
lcd_putchar(recv[12]);
lcd_putchar(recv[13]);
lcd_putchar(recv[14]);
lcd_putchar(recv[15]);
lcd_putchar(recv[16]);
lcd_putchar(recv[17]);
}
}

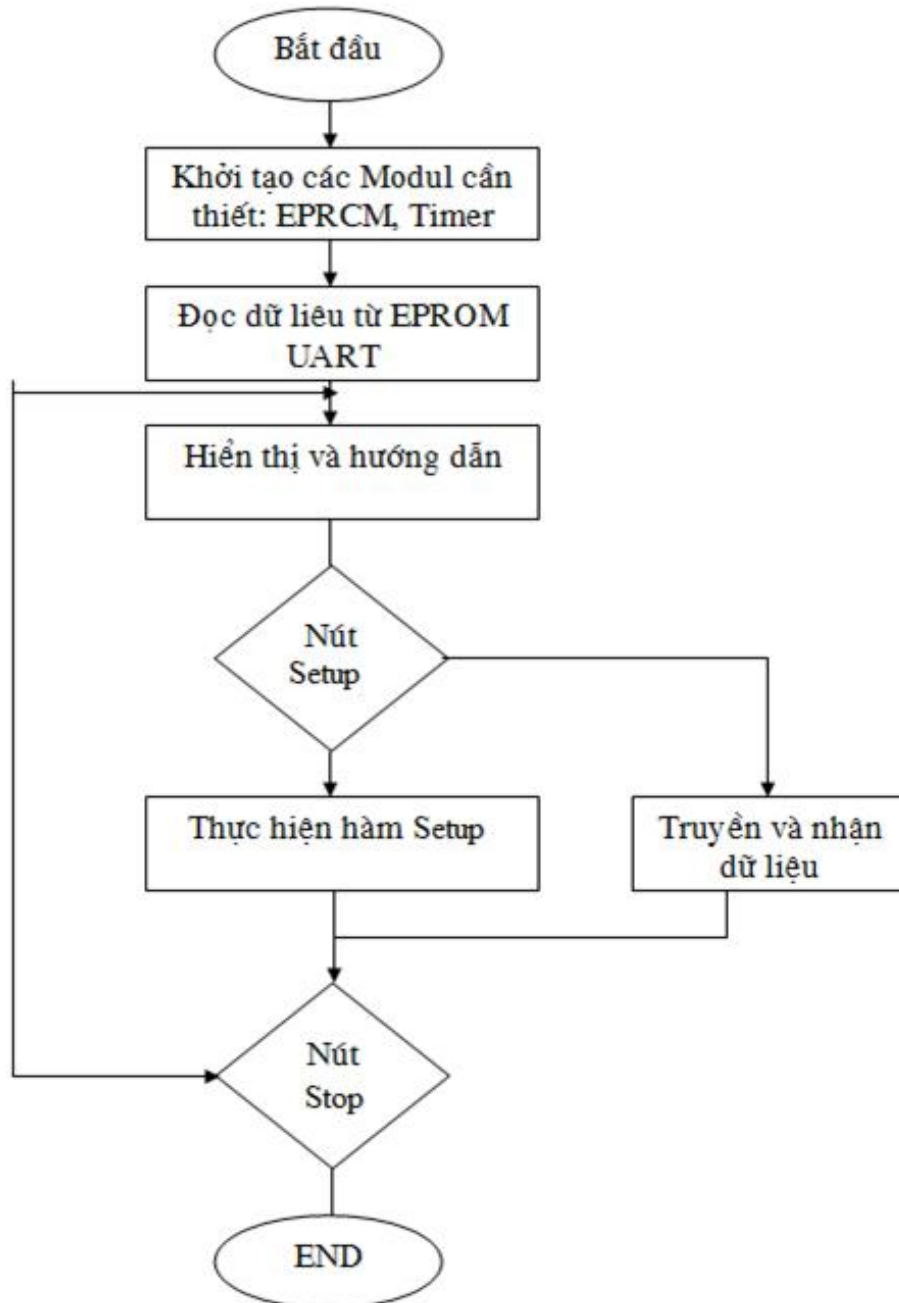
```

### 3.3.2. PHẦM MỀM MẠCH GIÁM SÁT

Mạch giám sát có nhiệm vụ nhận các thông số và trạng thái điều khiển của hệ thống từ mạch điều khiển để hiện thị. Đồng thời mạch giám sát thực hiện giao tiếp với người sử dụng để thực hiện cài đặt các thông số, sau đó truyền thông số tới mạch điều khiển để mạch điều khiển thực thi.

Chương trình mạch điều khiển giám sát cũng rất phức tạp và gồm nhiều chương trình con. Vì vậy tác giả xin chỉ trình bày hàm điển hình là hàm main của chương trình mạch giám sát.

### Lưu đồ thuật toán



-Hàm có chức năng nhận dữ liệu truyền các tham số dữ liệu của người sử dụng cài đặt ở mạch giám sát xuống mạch điều khiển đồng thời nhận các thông số từ mạch điều khiển để hiển thị và giao tiếp với người sử dụng

- Mạch bắt đầu hoạt động, khởi tạo các moodul cần thiết: Bộ nhớ EPROM, Bộ thời gian Timer

- Bắt đầu đọc dữ liệu từ bộ nhớ EPROM và UART sau đó hiển thị các giá trị đọc được và ghi các hướng dẫn cần thiết.

- Nút Stop có chức năng dừng hệ thống

- Hàm setup thực hiện các chức năng cài đặt các tham số của hệ thống.

### **Mã nguồn chương trình mạch giám sát:**

```
/******  
Chip type      : ATmega128  
Program type   : Application  
AVR Core Clock frequency: 11.059200 MHz  
Memory model   : Small  
External RAM size   : 0  
Data Stack size   : 1024  
*****/  
#include <mega128.h>  
#include <stdlib.h>  
#include <MATH.h>  
#include <alcd.h>  
#include <delay.h>           // da them  
#asm  
    .equ __i2c_port=0x12 ;PORTD  
    .equ __sda_bit=1  
    .equ __scl_bit=0  
#endasm  
#include <i2c.h>  
// DS1307 Real Time Clock functions  
#include <ds1307.h>  
// Alphanumeric LCD Module functions  
#define RE PORTE.2  
#define LOA PORTB.4  
#define CANCEL 17  
#define UP 18  
#define DOWN 19  
#define ENTER 20  
#define MENU 21  
#define DAU_PHAY 22  
#define NO_KEY 255  
// Declare your global variables here  
#define EEPROM_BUS_ADDRESS 0xa0  
#define ADC_VREF_TYPE 0x00  
#define PID_MAX 1022  
#define PID_MIN 0  
unsigned char h,m,s;  
char gio,phut,giay;  
unsigned char key_code;  
bit menu;  
unsigned char so_mu;  
int pr_index;  
unsigned char save_index;  
bit menu_ok;  
unsigned char digit_index;  
float Vh,Vl,Im,Is,Vb,Rt,Ki,Kp,Kd;
```

```

float kq,kq1,kq2;
float luy_thua;
unsigned char sig;
unsigned char digit[6];
unsigned char chuoi_so[20];
char send_ID_1[50];
char send_ID_2[50];
char send_index;
char recv1[60];
char recv2[60];
char recv_index;
bit ID1_send;
bit ID2_send;
char in_char;
bit ADDR1_OK;
bit ADDR2_OK;
unsigned int j,q,l,n;
// USART0 Receiver interrupt service routine
interrupt [USART0_RXC] void usart0_rx_isr(void)
{
    in_char=UDR0;
    if (in_char==0x01)
    {
        ADDR1_OK=1;
        recv_index=0;
    }
    else
    if (in_char==0x02)
    {
        ADDR2_OK=1;
        recv_index=0;
    }
    else
    if (ADDR1_OK) // nhan du lieu tu Slaver1
    {
        if (in_char=='S')
        {
            RE=1; // cho phep truyen
            ADDR1_OK=0;
            ID2_send=1; // cho phep truyen den Slaver2
        }
        for (q=0;q<500;q++)
        {
            UDR0=0x02; // da nhan xong du lieu tu Slaver1 truyen ky dia chi den Slaver2
        }
        else
        {
            recv1[recv_index]=in_char; // nhan du lieu tu Slaver1
            recv_index++;
        }
    }
    else
    if (ADDR2_OK)
    {
        if (in_char=='S')
        {
            RE=1; // cho phep truyen
            ADDR2_OK=0;
            ID1_send=1;
            for (q=0;q<500;q++)

```

```

    {
    }
    UDR0=0x01;    // truyen ky dia chi
    }
    else
    {
    recv2[recv_index]=in_char;
    recv_index++;
    }
}
}
// USART0 Transmitter interrupt service routine
interrupt [USART0_TXC] void usart0_tx_isr(void)
{
for (q=0;q<500;q++)
    {
    ;
    }
if (send_index<50)
    {
if (ID1_send)
    {
    UDR0=send_ID_1[send_index];
    send_index++;
    }
else
if (ID2_send)
    {
    UDR0=send_ID_2[send_index];
    send_index++;
    }
}
else
    {
    ID2_send=0;
    ID1_send=0;
    send_index=0;
    RE=0;
    }
}
// USART1 Receiver interrupt service routine
interrupt [USART1_RXC] void usart1_rx_isr(void)
{
key_code=UDR1-48;
if (key_code==21)
    {
    menu=!menu;
    }
}
// Declare your global variables here
unsigned char eeprom_read(unsigned char address)
{
    unsigned char data;
    i2c_start();
    i2c_write(EEPROM_BUS_ADDRESS);
    i2c_write(address);
    i2c_start();
    i2c_write(EEPROM_BUS_ADDRESS | 1);
    data=i2c_read(0);
    i2c_stop();
}

```

```

    return data;
}
void eeprom_write(unsigned char address, unsigned char data)
{
    i2c_start();
    i2c_write(EEPROM_BUS_ADDRESS);
    i2c_write(address);
    i2c_write(data);
    i2c_stop();
    /* 10ms delay to complete the write operation */
    for (l=0;l<1000;l++)
        {
        }
}
void read_eeprom(char addr,char offset)
{
    char j;
    char i=0;
    for (j=addr;j<addr+offset;j++)
        {
        digit[i]=eeprom_read(j);
        i++;
        }
}
void write_eeprom(char addr,char offset)
{
    char j;
    char i=0;
    for (j=addr;j<addr+offset;j++)
        {
        eeprom_write(j,chuoi_so[i]);
        i++;
        }
}
float khoi_phuc_so ()
{
    float number;
    number=atof(digit);
    return (number);
}
void hien_thi_so_thuc(float so_thuc)
{
    char str[20];
    ftoa(so_thuc, 3, str);
    lcd_puts(str);
}
void clear(void)
{
    lcd_gotoxy(3,0);
    lcd_puts(" ");
    lcd_gotoxy(3,0);
}
void caculation( char key_num )
{
    if(digit_index==0)
        {
        clear();
        kq=0;
        kq1=0;
        }
}

```

```

    if (digit_index<5)
    {
        kq=10*kq+key_num;
        lcd_putchar(key_num+48);
        digit_index++;
    }
}
void char_to_assc( unsigned char so)
{
    char K[2];
    if (so<10)
    {
        lcd_putchar('0');
        lcd_putchar(so+48);
    }
    else
    {
        itoa(so,K);
        lcd_puts(K);
    }
}
void send_data1(float data,char start_addr) // day du lieu vao vung nho truyen thong
{
    char var[10];
    char t;
    char p=0;
    ftoa(data,5,var);
    for(t=start_addr;t<start_addr+6;t++)
    {
        send_ID_1[t]=var[p];
        p++;
    }
}
void send_data2(float data,char start_addr) // day du lieu vao vung nho truyen thong
{
    char var[10];
    char t;
    char p=0;
    ftoa(data,5,var);
    for(t=start_addr;t<start_addr+6;t++)
    {
        send_ID_2[t]=var[p];
        p++;
    }
}
void display1 (char addr) // hien thi du lieu tu vung nho truyen thong slver1 ra LCD
{
    char j;
    for (j=addr;j<addr+5;j++)
    {
        lcd_putchar(recv1[j]);
    }
}
void main(void)
{
    PORTA=0x00;
    DDRA=0x00;
    PORTB=0x00;
    DDRB=0x10;
    PORTC=0x00;
}

```



```

DDRC=0x00;
PORTD=0x00;
DDRD=0x00;
PORTE=0x04;
DDRE=0x04;
PORTF=0x00;
DDRF=0x04;
PORTG=0x00;
DDRG=0x00;
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x00;
UCSR0A=0x00;
UCSR0B=0xD8;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x23;
UCSR1A=0x00;
UCSR1B=0x90;
UCSR1C=0x06;
UBRR1H=0x00;
ACSR=0x80;
SFIOR=0x00;
ADCSRA=0x00;
SPCR=0x00;

```

```

TWCR=0x00;
lcd_init(16);
read_eeprom(0,6);
Vh=khoi_phuc_so();
read_eeprom(6,6);
Vl=khoi_phuc_so();
read_eeprom(12,6);
Is=khoi_phuc_so();
read_eeprom(18,6);
Im=khoi_phuc_so();
read_eeprom(24,6);
Vb=khoi_phuc_so();
read_eeprom(30,6);
Rt=khoi_phuc_so();
read_eeprom(36,6);
Kp=khoi_phuc_so();
read_eeprom(42,6);
Ki=khoi_phuc_so();
read_eeprom(48,6);
Kd=khoi_phuc_so();
lcd_gotoxy(0,0);
lcd_puts("Time:");
lcd_gotoxy(0,1);
lcd_puts("Vchg:");
lcd_gotoxy(0,2);
lcd_puts("Ichg:");
lcd_gotoxy(0,3);
lcd_puts("Status:");
// Global enable interrupts
#asm("sei")
delay_ms(100);
ID1_send=1; // cho phep truyen toi ID1
send_ID_1[49]='S'; // ma ket thuc
send_ID_2[49]='S'; // ma ket thuc
send_index=0;
UDR0=0x01; // truyen khoi tao byte dia chi
while (1)
{
    // Place your code here
    rtc_get_time(&h,&m,&s);
    send_data1(Vh,0);
    send_data1(Vl,6);
    send_data1(Is,12);
    send_data1(Im,18);
    send_data1(Kp,24);
    send_data1(Ki,30);
    send_data1(Kd,36);
    send_data2(Vb,0);
    send_data2(Rt,6);
    if (!menu)
    {
        if(menu_ok)
        {
            lcd_clear();
            lcd_gotoxy(0,0);
            lcd_puts("Time:");
            lcd_gotoxy(0,1);
            lcd_puts("Vchg:");
            lcd_gotoxy(0,2);
            lcd_puts("Ichg:");
        }
    }
}

```

```

lcd_gotoxy(0,3);
lcd_puts("Status:");
menu_ok=0;
}
if (key_code==CANCEL)
{
send_ID_1[42]='R';
}
else
if(key_code==ENTER)
{
send_ID_1[43]='T';
}
key_code=NO_KEY;

lcd_gotoxy(5,0);
char_to_assc(h);
lcd_putchar(':');
char_to_assc(m);
lcd_putchar(':');
char_to_assc(s);
lcd_gotoxy(5,1);
display1(0);
lcd_gotoxy(5,2);
display1(5);
if (recv1[10]=='F')
{
lcd_gotoxy(7,3);
lcd_putchar('O');
lcd_putchar('v');
lcd_putchar('_');
lcd_putchar('C');
lcd_putchar('u');
lcd_putchar('r');
lcd_putchar('e');
lcd_putchar('n');
lcd_putchar('t');
LOA=1;
for (n=0;n<30000;n++)
{
}
LOA=0;
for (n=0;n<30000;n++)
{
}
}
if ((recv1[10]=='Y')||(recv1[10]=='O'))
{
lcd_gotoxy(7,3);
lcd_putchar('N');
lcd_putchar('o');
lcd_putchar('_');
lcd_putchar('E');
lcd_putchar('r');
lcd_putchar('r');
lcd_putchar('o');
lcd_putchar('r');
lcd_putchar(' ');
send_ID_1[42]=0x00;
}

```

```

    if (recv1[11]=='K')
    {
        send_ID_1[43]=0x00;
    }
}
else // key_pad programmer
{
    if (!menu_ok)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putchar('V');
        lcd_putchar('h');
        lcd_putchar(':');
        kq1=Vh;
        kq=0;
        hien_thi_so_thuc(Vh);
        menu_ok=1;
        pr_index=1;
        digit_index=0;
        save_index=1;
        sig=0;
    }
switch (key_code)
{
case UP :
{
    if (pr_index<12)
    {
        pr_index++;
        if (pr_index==1) // vao thong so Vh
        {
            lcd_clear();
            lcd_gotoxy(0,0);
            lcd_putchar('V');
            lcd_putchar('h');
            lcd_putchar(':');
            kq1=Vh;
            hien_thi_so_thuc(Vh);
        }
        else
        if (pr_index==2)
        {
            lcd_clear();
            lcd_gotoxy(0,0);
            lcd_putchar('V');
            lcd_putchar('l');
            lcd_putchar(':');
            kq1=Vl;
            hien_thi_so_thuc(Vl);
        }
        else
        if (pr_index==3)
        {
            lcd_clear();
            lcd_gotoxy(0,0);
            lcd_putchar('l');
            lcd_putchar('s');
            lcd_putchar(':');
            kq1=Is;
        }
    }
}
}
}

```

```

hien_thi_so_thuc(Is);
}
else
if (pr_index==4)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('I');
lcd_putchar('m');
lcd_putchar(':');
kq1=Im;
hien_thi_so_thuc(Im);
}
else
if (pr_index==5)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('V');
lcd_putchar('b');
lcd_putchar(':');
kq1=Vb;
hien_thi_so_thuc(Vb);
}
if (pr_index==6)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('R');
lcd_putchar('t');
lcd_putchar(':');
kq1=Rt;
hien_thi_so_thuc(Rt);
}
else
if (pr_index==7)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('K');
lcd_putchar('p');
lcd_putchar(':');
kq1=Kp;
hien_thi_so_thuc(Kp);
}
else
if (pr_index==8)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('K');
lcd_putchar('i');
lcd_putchar(':');
kq1=Ki;
hien_thi_so_thuc(Ki);
}
else
if (pr_index==9)
{
lcd_clear();

```

```

        lcd_gotoxy(0,0);
        lcd_putchar('K');
        lcd_putchar('d');
        lcd_putchar(':');
        kq1=Kd;
        hien_thi_so_thuc(Kd);
    }
    else
    if (pr_index==10)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putchar('H');
        lcd_putchar('r');
        lcd_putchar(':');
    }
    else
    if (pr_index==11)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putchar('M');
        lcd_putchar('i');
        lcd_putchar(':');
    }
    else
    if (pr_index==12)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putchar('S');
        lcd_putchar('e');
        lcd_putchar(':');
    }
}
    kq=0;
    save_index=pr_index;
digit_index=0;
sig=0;
}
break;
}
case DOWN:
{
if (pr_index>1)
{
pr_index--;
if (pr_index==12)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('S');
lcd_putchar('e');
lcd_putchar(':');
}
if (pr_index==11)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('M');
lcd_putchar('i');
}
}
}
}

```

```

lcd_putchar(':');
}
else
if (pr_index==10)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('H');
lcd_putchar('r');
lcd_putchar(':');
}
else
if (pr_index==9)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('K');
lcd_putchar('d');
lcd_putchar(':');
hien_thi_so_thuc(Kd);
kq1=Kd;
}
else
if (pr_index==8)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('K');
lcd_putchar('i');
lcd_putchar(':');
hien_thi_so_thuc(Ki);
kq1=Ki;
}
else
if (pr_index==7)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('K');
lcd_putchar('p');
lcd_putchar(':');
hien_thi_so_thuc(Kp);
kq1=Kp;
}
if (pr_index==6)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('R');
lcd_putchar('t');
lcd_putchar(':');
hien_thi_so_thuc(Rt);
kq1=Rt;
}
else
if (pr_index==5)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putchar('V');

```

```

        lcd_putchar('b');
        lcd_putchar(':');
        hien_thi_so_thuc(Vb);
        kq1=Vb;
    }
    else
    if (pr_index==4)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putchar('I');
        lcd_putchar('m');
        lcd_putchar(':');
        hien_thi_so_thuc(Im);
        kq1=Im;
    }
    else
    if (pr_index==3)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putchar('I');
        lcd_putchar('s');
        lcd_putchar(':');
        hien_thi_so_thuc(Is);
        kq1=Is;
    }
    else
    if (pr_index==2)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putchar('V');
        lcd_putchar('l');
        lcd_putchar(':');
        hien_thi_so_thuc(Vl);
        kq1=Vl;
    }
    else
    if (pr_index==1)    // vao thong so Vh
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putchar('V');
        lcd_putchar('h');
        lcd_putchar(':');
        hien_thi_so_thuc(Vh);
kq1=Vh;
    }
    kq=0;
    save_index=pr_index;
digit_index=0;
    sig=0;
    }
    break;
    }
case ENTER:
    {
        if(save_index==1)    // save gia tri Vh

```



```

        {
        luy_thua=(float)(digit_index-so_mu)*sig;
        kq2=kq1+kq/pow(10,luy_thua);
        ftoa( kq2, 5, chuoi_so);
        write_eeprom(0,6);
Vh=kq2;      // tra lai gia tri tuc thoi cho Vh
    }

    else
    if (save_index==2)      // luu gia tri V1
    {
        luy_thua=(float)(digit_index-so_mu)*sig;
        kq2=kq1+kq/pow(10,luy_thua);
        ftoa( kq2, 5, chuoi_so);
        write_eeprom(6,6);
V1=kq2;      // tra lai gia tri tuc thoi cho V1
    }

    else
    if (save_index==3)      // luu gia tri Is
    {
        luy_thua=(float)(digit_index-so_mu)*sig;
        kq2=kq1+kq/pow(10,luy_thua);
        ftoa( kq2, 5, chuoi_so);
        write_eeprom(12,6);
        Is=kq2;      // tra lai gia tri tuc thoi cho Is
    }

    else
    if (save_index==4)      // luu gia tri Imax
    {
        luy_thua=(float)(digit_index-so_mu)*sig;
        kq2=kq1+kq/pow(10,luy_thua);
        ftoa( kq2,5, chuoi_so);
        write_eeprom(18,6);
Im=kq2;      // tra lai gia tri tuc thoi cho Imax
    }

    else
    if (save_index==5)      // luu gia tri Vb
    {
        luy_thua=(float)(digit_index-so_mu)*sig;
        kq2=kq1+kq/pow(10,luy_thua);
        ftoa( kq2,5, chuoi_so);
        write_eeprom(24,6);
Vb=kq2;      // tra lai gia tri tuc thoi cho Vb
    }

    else
    if (save_index==6)      // luu gia tri Rt
    {
        luy_thua=(float)(digit_index-so_mu)*sig;
        kq2=kq1+kq/pow(10,luy_thua);
        ftoa( kq2,5, chuoi_so);
        write_eeprom(30,6);
Rt=kq2;      // tra lai gia tri tuc thoi cho Vb
    }

    else
    if (save_index==7)      // luu gia tri Kp
    {
        luy_thua=(float)(digit_index-so_mu)*sig;
        kq2=kq1+kq/pow(10,luy_thua);
        ftoa( kq2,5, chuoi_so);
        write_eeprom(36,6);
Kp=kq2;      // tra lai gia tri tuc thoi cho Kp
    }

```

```

}
else
if (save_index==8)    // luu gia tri Ki
{
luy_thua=(float)(digit_index-so_mu)*sig;
kq2=kq1+kq/pow(10,luy_thua);
ftoa( kq2,5, chuo_i_so);
write_eeprom(42,6);
Ki=kq2;    // tra lai gia tri tuc thoi cho Ki
}

else
if (save_index==9)    // luu gia tri Kd
{
luy_thua=(float)(digit_index-so_mu)*sig;
kq2=kq1+kq/pow(10,luy_thua);
ftoa( kq2,5, chuo_i_so);
write_eeprom(48,6);
Kd=kq2;    // tra lai gia tri tuc thoi cho Kd
}

else
if (save_index==10)
{
gio=(char)kq;
}
else
if (save_index==11)
{
phut=(char)kq;
}
else
if (save_index==12)
{
giay=(char)kq;
rtc_set_time(gio,phut,giay);
}
kq1=kq2;
kq=0;
sig=0;
break;
}
case DAU_PHAY:
{
if(digit_index==0)
{
clear();
}
if(digit_index<5)
{
so_mu=digit_index;
lcd_putchar('.');
sig=1;
}
break;
}
case CANCEL:
{
kq1=0;
kq=0;
sig=0;
}

```

```
    digit_index=0;
    clear();
    break;
}
case 0:
{
    caculation(0);
    break;
}
case 1:
{
    caculation(1);
    break;
}
case 2:
{
    caculation(2);
    break;
}
case 3:
{
    caculation(3);
    break;
}
case 4:
{
    caculation(4);
    break;
}
case 5:
{
    caculation(5);
    break;
}
case 6:
{
    caculation(6);
    break;
}
case 7:
{
    caculation(7);
    break;
}
case 8:
{
    caculation(8);
    break;
}
case 9:
{
    caculation(9);
    break;
}
}
key_code=NO_KEY;
}
```

## KẾT LUẬN

Sau thời gian thực hiện đề án dưới sự hướng dẫn tận tình của thầy giáo **TS Nguyễn Trọng Thắng** em đã hoàn thành đề tài tốt nghiệp được giao. Đề án của em đã giải quyết được những vấn đề cơ bản sau:

-Tìm hiểu được về cấu trúc, các đặc tính và thông số cơ bản của bình ắc quy

-Nắm được cấu trúc cơ bản của một mạch nạp ắc quy tự động, bước đầu xây dựng được mạch nạp ắc quy tự động trên mô hình thực tế.

Nhưng do trình độ và kiến thức thực tế còn hạn chế nên đề án của em cũng không thể tránh khỏi những sai sót, em rất mong nhận được sự đóng góp ý kiến của quý thầy cô cùng các bạn để đề án của em hoàn thiện hơn.

## TÀI LIỆU THAM KHẢO

1. Lê Văn Doanh, Nguyễn Thế Công, Trần Văn Thịnh (2005), *Điện tử công suất*, NXB Khoa học và kỹ thuật.
2. Hồ Khánh Lâm (2008), *Giáo trình kỹ thuật vi xử lý*, NXB Thông tin và truyền thông.
3. Một số trang web tham khảo : <http://codientu.org/>, <http://www.hocavr.com>, [google.com](http://google.com)