

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



**ISO 9001:2015**

**THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ  
DỤNG VI ĐIỀU KHIỂN 8051 – PHƯƠNG ÁN ĐIỀU  
KHIỂN TỰ ĐỘNG MÁY CÀ**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY  
NGÀNH ĐIỆN TỬ VIỄN THÔNG**

**HẢI PHÒNG - 2019**

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



**ISO 9001:2015**

**THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ  
DỤNG VI ĐIỀU KHIỂN 8051 PHƯƠNG ÁN ĐIỀU  
KHIỂN TỰ ĐỘNG MÁY CÀ**

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

**NGÀNH ĐIỆN TỬ VIỄN THÔNG**

**Sinh viên: NGUYỄN NHẬT NAM**

**Người hướng dẫn: TS. MAI VĂN LẬP**

**HẢI PHÒNG - 2019**

Cộng hoà xã hội chủ nghĩa Việt Nam  
**Độc lập – Tự Do – Hạnh Phúc**  
-----o0o-----  
BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

## **NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên : NGUYỄN NHẬT NAM – MSV : 1412103010  
Lớp : ĐT1801 - Ngành Điện Tử Viễn Thông  
Tên đề tài : Thiết Kế Hệ Thống Điều Khiển Tự Động Sử Dụng  
Vi Điều Khiển 8051

## 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 : Mai Văn Lập  
Học hàm, học vị : Tiến sĩ  
Cơ quan công tác : Trường Đại học dân lập Hải Phòng  
Nội dung hướng dẫn : Toàn bộ đề tài

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 2018.

Yêu cầu phải hoàn thành xong trước ngày ..... tháng ..... năm 2019.

Đã nhận nhiệm vụ Đ.T.T.N  
Sinh viên

Đã giao nhiệm vụ Đ.T.T.N  
Cán bộ hướng dẫn Đ.T.T.N

Nguyễn Nhật Nam

TS. Mai Văn Lập

Hải Phòng, ngày.....tháng.....năm 2019

HIỆU TRƯỞNG

GS.TS.NGƯT TRẦN HỮU NGHỊ

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**Độc lập - Tự do - Hạnh phúc**

**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP**

Họ và tên giảng viên: .....

Đơn vị công tác: .....

Họ và tên sinh viên: ..... Chuyên ngành: .....

Nội dung hướng dẫn: .....

.....

**Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp**

.....

.....

.....

.....

**1. Đánh giá chất lượng của đề án/khóa luận (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 số liệu...)**

.....

.....

.....

.....

.....

**3. Ý kiến của giảng viên hướng dẫn tốt nghiệp**

Được bảo vệ  Không được bảo vệ  Điểm hướng dẫn

*Hải Phòng, ngày ... tháng ... năm .....*

**Giảng viên hướng dẫn**

*(Ký và ghi rõ họ tên)*

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**Độc lập - Tự do - Hạnh phúc**

---

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN CHẤM PHẢN BIỆN**

Họ và tên giảng viên: .....

Đơn vị công tác: .....

Họ và tên sinh viên: ..... Chuyên ngành: .....

Đề tài tốt nghiệp: .....

**1. Phần nhận xét của giáo viên chấm phản biện**

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

**2. Những mặt còn hạn chế**

.....  
.....

**3. Ý kiến của giảng viên chấm phản biện**

Được bảo vệ  Không được bảo vệ  Điểm hướng dẫn

*Hải Phòng, ngày ... tháng ... năm .....*

**Giảng viên chấm phản biện**

*(Ký và ghi rõ họ tên)*

## MỤC LỤC

<b>LỜI CẢM ƠN</b> .....	<b>3</b>
<b>CHƯƠNG I: GIỚI THIỆU VI ĐIỀU KHIỂN 8051 (AT89S52)</b> .....	<b>4</b>
1.1 GIỚI THIỆU .....	4
1.2 LỊCH SỬ PHÁT TRIỂN CỦA CÁC LOẠI VI ĐIỀU KHIỂN.....	4
1.3 CẤU TRÚC BÊN TRONG CỦA AT89S52.....	5
1.4 TÓM TẮT PHẦN CỨNG: .....	6
1.5 MẠCH CƠ BẢN ĐỂ AT89S52 HOẠT ĐỘNG. ....	11
<b>CHƯƠNG 2: GIỚI THIỆU VỀ TAY ĐIỀU KHIỂN TỪ XA HULVA 6 KÊNH 2.4GHz ( TX VÀ RX)</b> .....	<b>12</b>
2.1 TX (viết tắt của từ Transmitter) có nghĩa là Máy phát sóng .....	12
<b>CHƯƠNG 3: GIỚI THIỆU MODUL ULN2003 &amp; MOTOR STEP</b> .....	<b>15</b>
3.1 GIỚI THIỆU .....	15
3.2 MODUL UNL2003 .....	16
3.3 ĐỘNG CƠ BƯỚC 28BYJ-48 5VDC .....	16
3.4 KẾT NỐI MODULE VÀ MOTOR STEP .....	18
<b>CHƯƠNG 4: THIẾT KẾ VÀ LẬP TRÌNH</b> .....	<b>19</b>
4.1 THIẾT KẾ .....	19
4.1.1 Sơ đồ khối .....	19
4.1.2 Sơ đồ mạch điều khiển trung tâm .....	19
4.1.3 Mô phỏng trên phần mềm Proteus .....	20
4.1.4 Vẽ mạch trên phần mềm Altium.....	20
4.1.5 Khối xử lí đóng cắt relay .....	21
4.2 THỰC HIỆN MẠCH THỰC TẾ VÀ LẮP RÁP MODULE .....	23
4.2.1 Các mạch thiết kế sau khi hoàn thiện .....	23
4.2.2 Thực hiện lắp ráp và ghép nối các mạch và Module.....	25
4.3 LẬP TRÌNH .....	27
4.3.1 Kết Cấu Của Chương Trình.....	27



4.3.2 Lập Trình.....	27
<b>KẾT LUẬN</b> .....	<b>38</b>
<b>TÀI LIỆU THAM KHẢO</b> .....	<b>39</b>

## LỜI CẢM ƠN

Khi em nghiên cứu đề tài, trong quá trình thực hiện đề án này ngoài sự nỗ lực, cố gắng của bản thân thì em đã nhận được sự hướng dẫn, giúp đỡ, động viên không nhỏ từ phía thầy giáo, cô giáo và bạn bè. Em xin gửi lời cảm ơn trân thành đến:

Thầy giáo TS. Mai Văn Lập đã trực tiếp giúp em định hướng đề tài đề án cũng như tận tình hướng dẫn, giải đáp những thắc mắc. Thầy cũng chia sẻ những kiến thức chuyên môn sâu và những kinh nghiệm quý báu giúp em hoàn thành đề án này.

Đồng thời em xin cảm ơn đến các thầy giáo, cô giáo trong bộ môn và các bạn trong lớp ĐT1801 đã nhiệt tình chia sẻ, giúp đỡ và động viên trong suốt quá trình làm đề án.

Cho dù em đã rất cố gắng, nỗ lực trong quá trình thực hiện nhưng đề án này có nhiều kiến thức mới. Cho nên sẽ không thể tránh khỏi những thiếu sót và những câu văn dịch từ tiếng anh không được rõ nghĩa lắm. Em rất mong nhận được sự góp ý, chỉ bảo tận tình của quý thầy giáo, cô giáo và các bạn đồng môn.

*Hải Phòng, tháng 1 năm 2019*

Sinh viên thực hiện

**NGUYỄN NHẬT NAM**

# **CHƯƠNG I: GIỚI THIỆU VI ĐIỀU KHIỂN 8051 (AT89S52)**

## **1.1 GIỚI THIỆU**

Bộ vi điều khiển viết tắt là Micro-controller, là mạch tích hợp trên một chip có thể lập trình được, dùng để điều khiển hoạt động của một hệ thống. Theo các tập lệnh của người lập trình, bộ vi điều khiển tiến hành đọc, lưu trữ thông tin, xử lý thông tin, đo thời gian và tiến hành đóng mở một cơ cấu nào đó.

Trong các thiết bị điện, điện và điện tử dân dụng, các bộ vi điều khiển, điều khiển hoạt động của TV, máy giặt, đầu đọc laser, điện thoại, lò vi-ba ... Trong hệ thống sản xuất tự động, bộ vi điều khiển được sử dụng trong Robot, dây chuyền tự động. Các hệ thống càng “thông minh” thì vai trò của hệ vi điều khiển càng quan trọng.

## **1.2 LỊCH SỬ PHÁT TRIỂN CỦA CÁC LOẠI VI ĐIỀU KHIỂN.**

Bộ vi điều khiển thực ra, là một loại vi xử lý trong tập hợp các bộ vi xử lý nói chung. Bộ vi điều khiển được phát triển từ bộ vi xử lý, từ những năm 70 do sự phát triển và hoàn thiện về công nghệ vi điện tử dựa trên kỹ thuật MOS (Metal-Oxide-Semiconductor), mức độ tích hợp của các linh kiện bán dẫn trong một chip ngày càng cao.

Năm 1971 xuất hiện bộ vi xử lý 4 bit loại TMS1000 do công ty Texas Instruments vừa là nơi phát minh vừa là nhà sản xuất. Nhìn tổng thể thì bộ vi xử lý chỉ có chứa trên một chip những chức năng cần thiết để xử lý chương trình theo một trình tự, còn tất cả bộ phận phụ trợ khác cần thiết như : bộ nhớ dữ liệu, bộ nhớ chương trình, bộ chuyển đổi A/D, khối điều khiển, khối hiển thị, điều khiển máy in, khối đồng hồ và lịch là những linh kiện nằm ở bên ngoài được nối vào bộ vi xử lý.

Mãi đến năm 1976 công ty INTEL (Intelligent-Electronics). Mới cho ra đời bộ vi điều khiển đơn chip đầu tiên trên thế giới với tên gọi 8048. Bên cạnh bộ xử lý trung tâm 8048 còn chứa bộ nhớ dữ liệu, bộ nhớ chương trình, bộ đếm và phát thời gian các công vào và ra Digital trên một chip.

Các công ty khác cũng lần lượt cho ra đời các bộ vi điều khiển 8bit tương tự như 8048 và hình thành họ vi điều khiển MCS-48 (Microcontroller-sustem-48).

Đến năm 1980 công ty INTEL cho ra đời thế hệ thứ hai của bộ vi điều khiển đơn chip với tên gọi 8051. Và sau đó hàng loạt các vi điều khiển cùng loại với 8051 ra đời và hình thành họ vi điều khiển MCS-51 .

Đến nay họ vi điều khiển 8 bit MCS51 đã có đến 250 thành viên và hầu hết các công ty hàng dẫn hàng đầu thế giới chế tạo. Đứng đầu là công ty INTEL và rất nhiều công ty khác như : AMD, SIEMENS, PHILIPS, DALLAS, OKI..

Ngoài ra còn có các công ty khác cũng có những họ vi điều khiển riêng như:

Họ 68HCOS của công ty Motorola

Họ ST62 của công ty SGS-THOMSON

Họ H8 của công ty Hitachi

Họ PIC của công ty Microchip

- Các đặc điểm của AT89S52 được tóm tắt như sau:

4 KB ROM

4 KB EPROM bên trong.

128 Byte RAM nội. 4 Port xuất nhập I/O 8 bit.

2 bộ định thời 16 bit

Mạch giao tiếp nối tiếp.

64 KB vùng nhớ mã ngoài

64 KB vùng nhớ dữ liệu ngoài

Xử lý Boolean (hoạt động trên bit đơn).

210 vị trí nhớ có thể định vị bit.

4s cho hoạt động nhân hoặc chia.

### **1.3 CẤU TRÚC BÊN TRONG CỦA AT89S52.**

- Phần chính của vi điều khiển 8051 / 8031 là bộ xử lý trung tâm (CPU: central processing unit ) bao gồm :

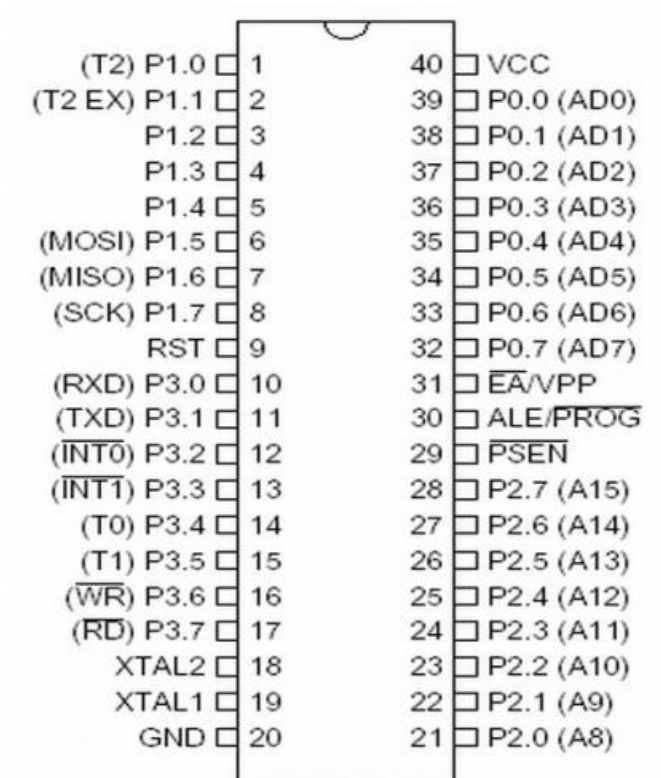
- Thanh phi tích lũy A

- Thanh ghi tích lũy phụ B, dùng cho phép nhân và phép chia
- Đơn vị logic học (ALU : Arithmetic Logical Unit )
- Từ trạng thái chương trình (PSW : Prorgam Status Word)
- Bốn băng thanh ghi
- Con trỏ ngăn xếp

Ngoài ra còn có bộ nhớ chương trình, bộ giải mã lệnh, bộ điều khiển thời gian và logic.

#### 1.4 TÓM TẮT PHẦN CỨNG:

- AT89S52 có tất cả 40 chân có chức năng như các đường xuất nhập . Trong đó có 24 chân có tác dụng kép (có nghĩa là một chân có hai chức năng), mỗi đường có thể hoạt động như đường xuất nhập hoặc như đường điều khiển hoặc là thành phần của các bus dữ liệu và bus địa chỉ.





Hình ảnh thực tế

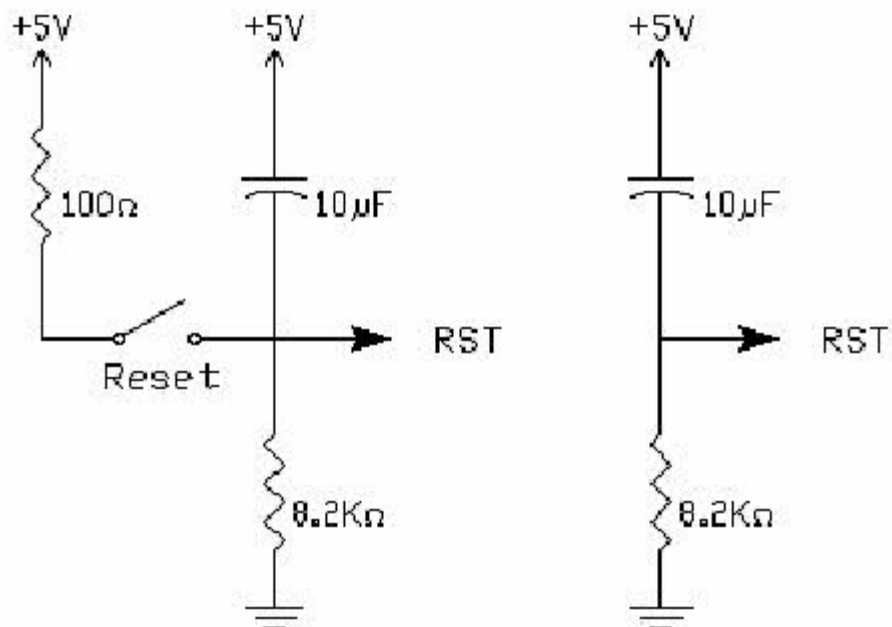
- Các công vào ra
- Port 0 (Chân 32-39): là cổng hai chiều dùng 8 bit để mở, như là công ra, Port 0 có những cấu hình công đường dẫn địa chỉ, dữ liệu để truy xuất tới chương trình ngoài và bộ nhớ dữ liệu. yêu cầu bên ngoài dùng lại trong lúc kiểm tra chương trình.
- Port 1 (Chân 1-8): có cổng hai chiều 8bit, trong phép cộng P1.0 và P1.1 có thể thực hiện để đi tới bộ định thời/bộ đếm bên trong đếm ngõ vào(P1.0/T2) và hai bộ định thời/bộ đếm truy xuất ngõ vào(P1.1/T2EX).
- Port 2 (Chân 21-28): có công hai chiều 8bit, phát ra những địa chỉ byte cao khác trong lúc tìm về từ bộ nhớ chương trình bên ngoài và truy xuất từ bộ nhớ dữ liệu bên ngoài việc đó sử dụng 8bit địa chỉ. Port 2 phát ra những nội dung của thanh ghi có chức năng đặc biệt P2
- Port 3 (Chân 10-17):

- P3.0 RxD Chân phát dữ liệu của Port nối tiếp
- P3.1 TxD Chân thu dữ liệu của Port nối tiếp
- P3.2 INT0 Ngõ vào ngắt ngoài 0
- P3.3 INT1 Ngõ vào ngắt ngoài 1
- P3.4 T0 Ngõ vào bộ định thời đếm 0
- P3.5 T1 Ngõ vào bộ định thời đếm 1
- P3.6 WR Điều khiển ghi dữ liệu vào RAM ngoài
- P3.7 RD Điều khiển đọc dữ liệu từ RAM ngoài

- Reset (Chân 9):

Chân reset có tác dụng reset cho chip, mức tích cực của chân này là mức 1, để reset ta phải đưa mức 1 (5v) đến chân này với thời gian tối thiểu 2 chu kỳ máy ( tương đương 2 $\mu$ s — tương đương với thạch anh 12Mhz ).

Sau đây là mạch reset.



Manual reset

Power-on reset

Trạng thái của các thanh ghi khi reset, khi reset thì trạng thái của RAM nội không bị thay đổi

- RxD :nhận tín hiệu kiểu nối tiếp.

- TxD :truyền tín hiệu kiểu nối tiếp.
- /INT0: ngắt ngoài 0.

-/INT1: ngắt ngoài 1.

- T0: chân vào 0 của bộ timer/counter 0.
- T1: chân vào 0 của bộ timer/counter 1.
- /WR: ghi dữ liệu vào bộ nhớ ngoài. - /Rd: đọc dữ liệu từ bộ nhớ ngoài.
- XTAL1: chân vào mạch khuếch đại dao động.
- XTAL2: chân ra từ mạch khuếch đại dao động.
- /PSEN: chân cho phép đọc chương trình ngoài (Rom ngoài).
- Chân cho phép chốt địa chỉ (ALE/PROG)

Chân ALE có xung ở ngõ ra để chốt địa chỉ Byte thấp trong thời gian truy xuất bộ

nhớ ngoài. Chân này có chương trình xung ở ngõ vào trong khi tín hiệu điện đang chạy.

Trong điều khiển bình thường, chân ALE được xuất ra với một giá trị bằng 1/6 tần số của mạch dao động và có thể được sử dụng cho việc quy định thời gian bên ngoài hoặc mục đích đếm thời gian. Ghi nhớ, một xung ALE được ngắt quãng trong khi mỗi truy xuất từ dữ liệu bộ nhớ ngoài. Nếu ra lệnh, bình thường ALE có thể bị hủy bởi việc cài đặt bit 0 của SFR

được định vị trí 8EH

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

PSEN được đọc xung nhọn tới bộ nhớ chương trình ngoài. Khi AT91S52RC đang thực hiện mã từ bộ nhớ chương trình ngoài, PSEN được thực hiện với chu kỳ máy tính gấp đôi, trừ phi hai hoạt động PSEN đó được ngắt quãng trong thời gian truy xuất tới bộ nhớ dữ liệu bên ngoài.

- Chân truy xuất ngoài (EA/VPP)



Kích hoạt truy xuất ngoài, chân EA phải được nối với GND khi sử dụng các thiết bị từ mã truy cập từ bộ nhớ chương trình ngoài được định vị trí từ 0000H tới FFFFH.

- Chân tinh thể thạch anh XTAL

XTALI: ngõ vào tới mạch dao động khuếch đại ngược và tới mạch điện khóa điều khiển bên trong.

XTAL2: ngõ ra từ mạch dao động khuếch đại ngược.

- Thanh ghi các chức năng đặc biệt (SFR)

Một ánh xạ trên bề mặt diện tích của bộ nhớ chip được gọi là thanh ghi các chức năng đặc biệt.

Ghi chú, đó không phải là tất cả các địa chỉ đã được sử dụng, và những địa chỉ không được sử dụng có thể không được bổ sung vào chip. Đọc truy xuất tới các địa chỉ đó sẽ được tổng hợp đầy đủ vào dữ liệu ngẫu nhiên, và truy xuất được ghi sẽ có hiệu ứng mờ.

- Thanh ghi bộ định thời 2:

Điều khiển và trạng thái các bit được chứa đựng vào thanh ghi T2CON và T2MOD.

- Thanh ghi ngắt:

Khởi động những bit ngắt riêng biệt được thực hiện bởi thanh ghi IE.

TF2: Dấu hiệu cờ tràn 2 bộ định thời đặt bởi 2 bộ định thời cờ tràn và phải được xóa bởi phần mềm. TF2 sẽ không được đặt khi RCLK = 1 hoặc TCLK = 1

EXF2: 2 Bộ định thời dấu hiệu ngoài khi một cái được giữ lại hoặc chạy lại bởi một từ chối chuyển tiếp trên T2EX và EXEN2 = 11. khi 2 bộ định thời trong được kích hoạt, EXF2 = 1 sẽ là nguyên nhân để CPU tới vector tới thủ tục 2 bộ định thời trong. XEN2 phải được xóa bởi phần mềm. EXF2 không phải nguyên nhân gây 9 ngắt trong bộ đếm lên/xuống (DCEN = 1). RCLK Kích hoạt xung nhận, khi điều chỉnh, nguyên nhân công nối tiếp được

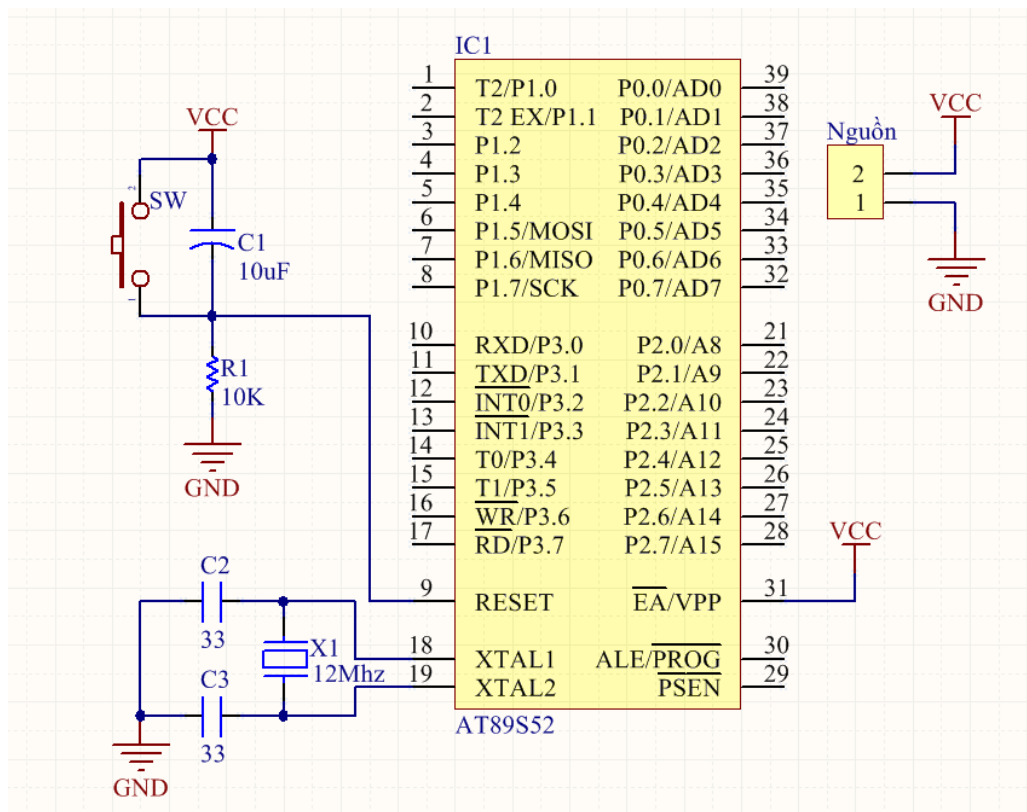
sử dụng 2 bộ định thời cờ tràn tạo xung cho xung nhận trong công nối tiếp cho dạng 1 và 3. RCLK = 0 là nguyên nhân cờ tràn một bộ định thời được sử dụng cho việc nhận xung.

- TCLK Kích hoạt truyền xung, khi điều chỉnh, nguyên nhân công nối tiếp được dùng cờ tràn xung bộ định thời 2 cho việc phát xung trong công nối tiếp cho dạng 1 và 3.

TCLK = 0 nguyên nhân cờ tràn bộ định thời 1 đã được dùng để phát xung.

- EXEN2 Kích hoạt bộ định thời ngoài 2, một cái được giữ lại hoặc chạy lại để xuất như là một kết quả của một từ chối chuyển tiếp trên T2EX neul bộ định thời 2 không được sử dụng để tạo xung cho công nối tiếp.

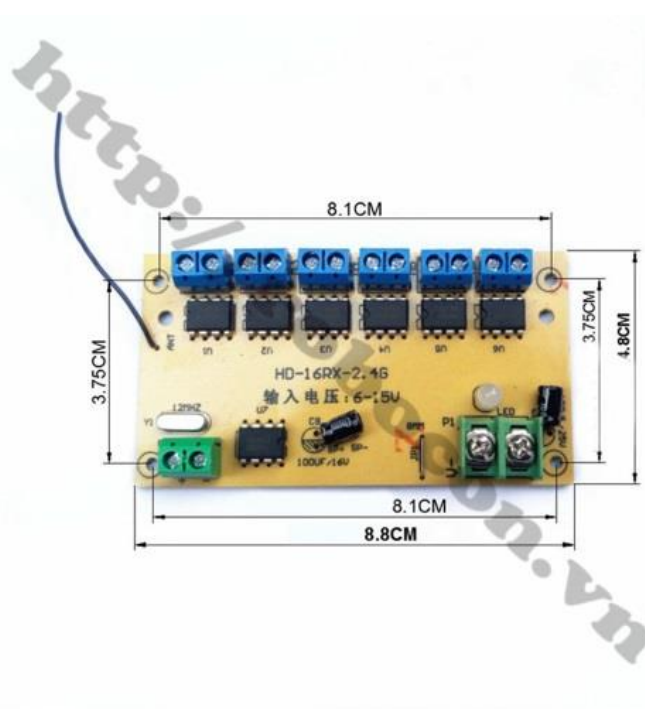
### 1.5 MẠCH CƠ BẢN ĐỂ AT89S52 HOẠT ĐỘNG.



## CHƯƠNG 2: GIỚI THIỆU VỀ TAY ĐIỀU KHIỂN TỪ XA HULVA 6 KÊNH 2.4GHz ( TX VÀ RX)

Tay điều khiển ( TX: mạch phát sóng)

Mạch thu sóng(RX)



### 2.1 TX (viết tắt của từ Transmitter) có nghĩa là Máy phát sóng

Máy phát có nhiệm vụ mã hóa vị trí của các cần điều khiển (stick) thành một dãy các tín hiệu điện (singal) và phát tín hiệu này ra không gian.

Tx có một số khái niệm như sau:

- Channel:

Đó là số kênh, số lệnh hay đơn giản nhất là số "servo" mà nó điều khiển được. Tùy vào Tx dùng cho mục đích gì mà số kênh có thể từ 1 đến 14 hay nhiều hơn nữa. Trong RC thì thông dụng có từ 2 đến 14 kênh.

- AM và FM:

Tất cả các Tx đều sử dụng radio để truyền tín hiệu ra không gian, tần số của sóng được xác định bởi thạch anh (crystal). Sóng radio đơn thuần chỉ là sóng mang (carrier frequency), một công cụ truyền dẫn, do đó để có thể

truyền tín hiệu đến máy thu (Rx), sóng radio cần phải được điều chế (modulation) trước khi phát đi! Có 2 dạng điều chế là AM và FM

- AM (amplitude modulation) điều biên: là tín hiệu được điều chế vào sóng mang dưới dạng thay đổi biên độ của sóng mang.

- FM (frequency modulation) điều tần: là tín hiệu được điều chế vào sóng mang dưới dạng thay đổi tần số sóng mang. Tất cả các máy phát dùng cơ chế mã hóa PCM đều dùng sóng mang là FM.

- Sóng FM nếu so sánh với sóng AM thì có khả năng chống nhiễu cao hơn hẳn. Với AM thì các thiết bị điện thông dụng đều là nguồn gây nhiễu cho sóng AM, trong khi đó với

FM thì các nguồn này không thể gây nhiễu trừ trường hợp các thiết bị đó có tần số gần hoặc bằng với tần số mà ta đang dùng.

- PPM và PCM:

Đây là cơ chế mã hóa tín hiệu trước khi phát ra của Tx

- PPM vị trí của servo được quyết định bởi thời gian của 2 xung tín hiệu liên tiếp, xét theo hình thức làm việc có thể xem nó thuộc nhóm Analog.

- PCM vị trí max & min của servo được chia ra thành nhiều khoảng nhỏ và được đánh số (VD với PCM1028 thì từ min tới max của servo được chia ra thành 1028 vị trí...) Và tùy theo vị trí của tay điều khiển mà Tx gửi đi 1 con số ứng với vị trí đó.

Module RF:

Với một số máy phát chất lượng cao, phần phát sóng được tách rời và người dùng có thể thay đổi dễ dàng. Khi đó với cùng một bộ điều khiển người dùng có thể dùng được ở nhiều băng tầng khác nhau bằng cách thay đổi module cho tần số tương ứng.

Spektrum:

Cũng là một loại sóng radio nhưng dùng tần số 2.4G và dùng kỹ thuật tương tự như các thiết bị Wifi của máy tính để tự điều chỉnh tần số. Do đó về lý thuyết Spektrum không bị trùng tần số như AM hay Fm thông thường.  
eCCPM: hay CCPM 120

Chức năng này chỉ dành riêng cho máy bay trực thăng. Có nghĩa là Tx có khả năng phối hợp 3 servo để điều khiển đồng thời các lệnh pith, airleron, elevator. Một số heli dùng cơ chế lái trực tiếp (mCCPM) thì không dùng đến tính năng này.

## **2.2 RX** (viết tắt của từ Receiver) hay Máy thu sóng

Có chức năng nhận sóng radio từ Tx và giải mã các tín hiệu thành tín hiệu điều khiển cho từng servo.

Tùy theo bạn dùng Tx gì mà chọn Rx theo Tx đó, có một số thông số như sau:

Tần số: Đương nhiên là phải cùng tần số với máy phát rồi

Số kênh (channel): Tùy vào nhu cầu mà bạn chọn Rx có số kênh tương ứng

PPM hay PCM: Đương nhiên 2 loại này có chất lượng khác nhau, nhưng khi chọn lựa có một số lưu ý như sau. Các máy dùng chế độ PCM thông thường đều có chế độ PPM. Nhưng những máy phát dùng chế độ PPM chưa chắc có chế độ PCM.

Single Conversion hay Dual Conversion: Nhằm tăng chất lượng nhận sóng và khả năng kháng nhiễu các mạch thu thường chuyển tần số sóng mang (cao tần) xuống tần số thấp hơn (trung tần) để khuếch đại và giải mã.

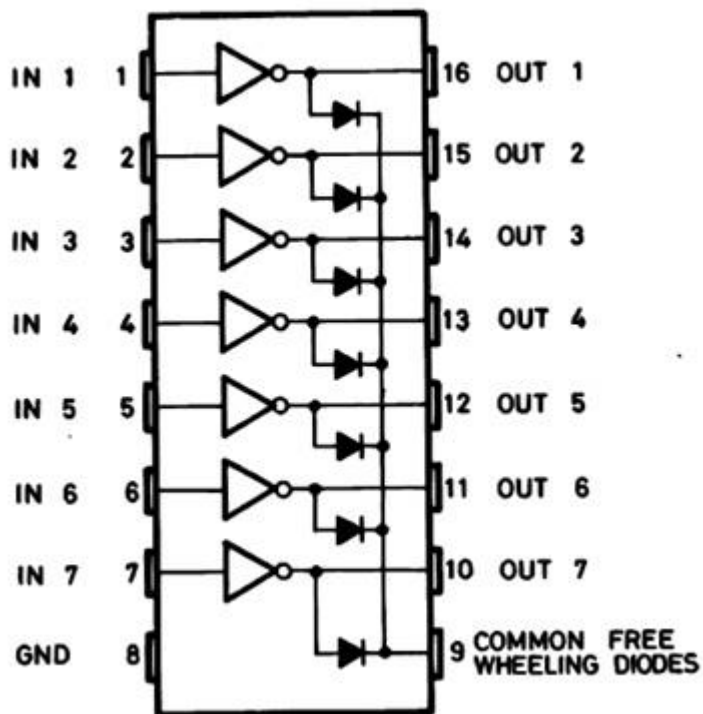
# CHƯƠNG 3: GIỚI THIỆU MODUL ULN2003 & MOTOR STEP

## 3.1 GIỚI THIỆU

- ULN2003 là tổ hợp của 7 mạch Darlington được tích hợp lại. Nó được dùng rộng rãi

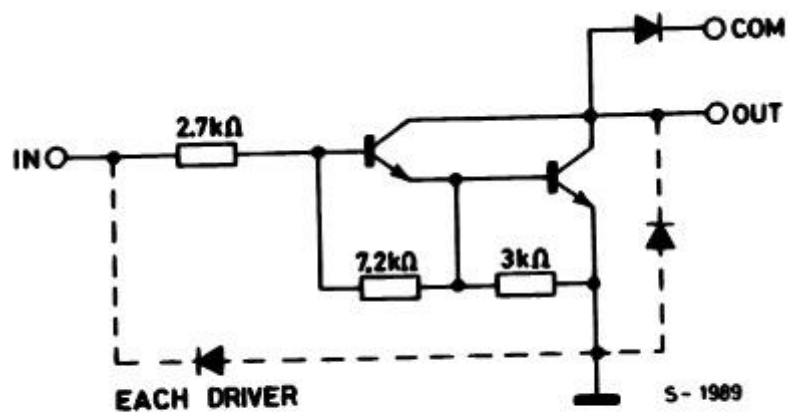
trong các ứng dụng điều khiển động cơ, Led,...

- Mỗi một kênh của UNL2003 có thể cho dòng đi qua tối đa 0.5A



- Sơ đồ chân

- Sơ đồ mạch của mỗi kênh.



Series ULN-2003A  
(each driver)

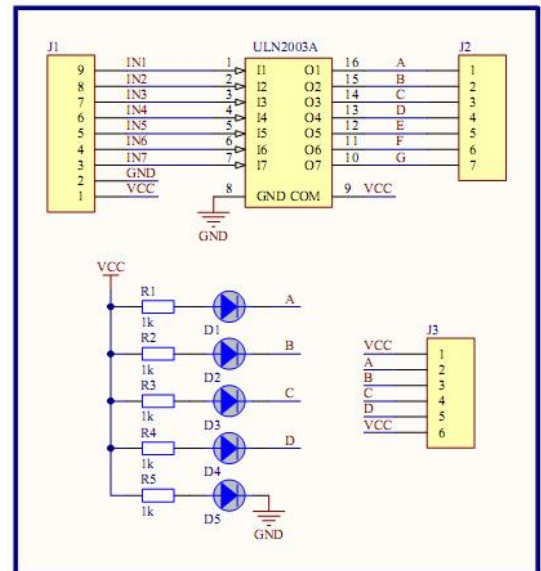
- Chú ý:

o Nếu nhìn thoáng qua sơ đồ chân của UNL2003 thì ta nhầm sang đây là IC đảo ( NOT). Nhưng thực tế nó không phải vậy.

o Đây là 1 mạch đệm công suất. Khi Input = 1; thì Output =0; còn khi Input =0, Output không xác định. Nó dùng tăng buffer cho VĐK.

### 3.2 MODUL UNL2003

Your Cee

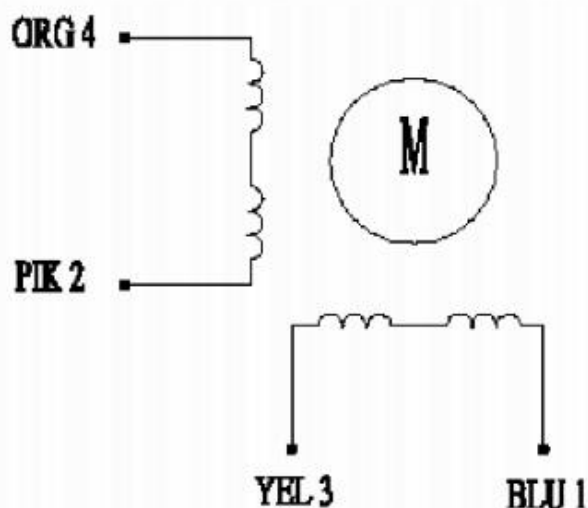


### 3.3 ĐỘNG CƠ BƯỚC 28BYJ-48 5VDC

- Đây là động cơ bước 5V. Nó có số bước là 64

- Sơ đồ cuộn dây

#### WIRING DIAGRAM





- Cấu tạo gồm 5 chân (đỏ). Trong đó chân thứ 5 là chân chung.
  - Trong 1 số trường hợp nếu ko rõ chân nào là chân chung và đâu là 2 chân của 1 cuộn dây ta có thể dùng đồng hồ để xác định
    - o Đo điện trở của cuộn dây để xác định
    - o Nếu 2 đầu đo có điện trở thì chúng trên 1 cuộn dây
    - o Điện trở của chân chung và 1 chân của cuộn dây bằng 1 nửa giá trị điện trở giữa 2 đầu 1 cuộn dây.
  - Kết nối động cơ bước và modul UNL2003
    - o Chân chung đấu vào VCC. Chú ý cấp điện áp VCC cho modul là 5V
    - o Các chân A,B,C,D nối vào các chân còn lại (1,2,3,4)
  - Điều khiển động cơ bước theo 2 chế độ:
    - o Chế độ nửa bước.
- § Trong chế độ nửa bước, mỗi một nhịp động cơ chỉ quay 1 nửa bước, tức là quay 128 lần thì hết 1 vòng.



§ Sơ đồ cấp điện

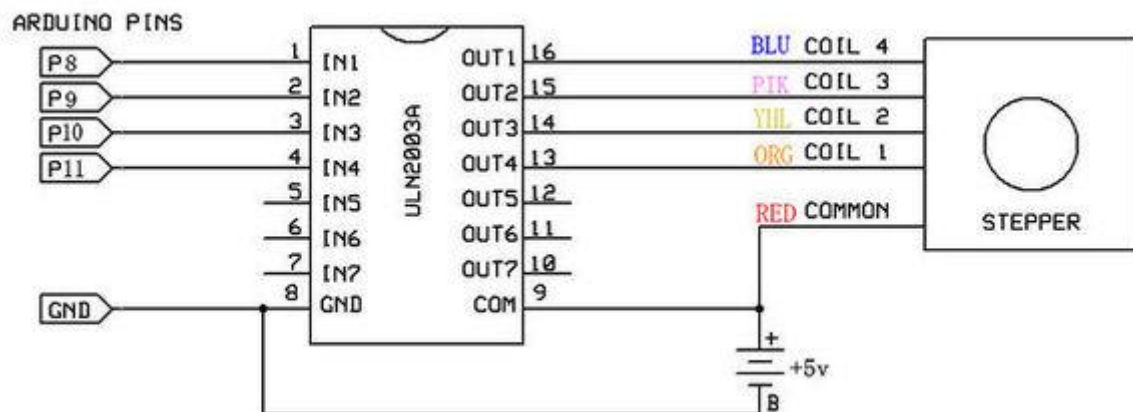
## SWITCHING SEQUENCE

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

o Chế độ 1 bước

§ Trong chế độ 1 bước, tại mỗi nhịp ta chỉ cấp điện cho 1 cuộn dây. Sau 64 nhịp động cơ quay được 1 vòng.

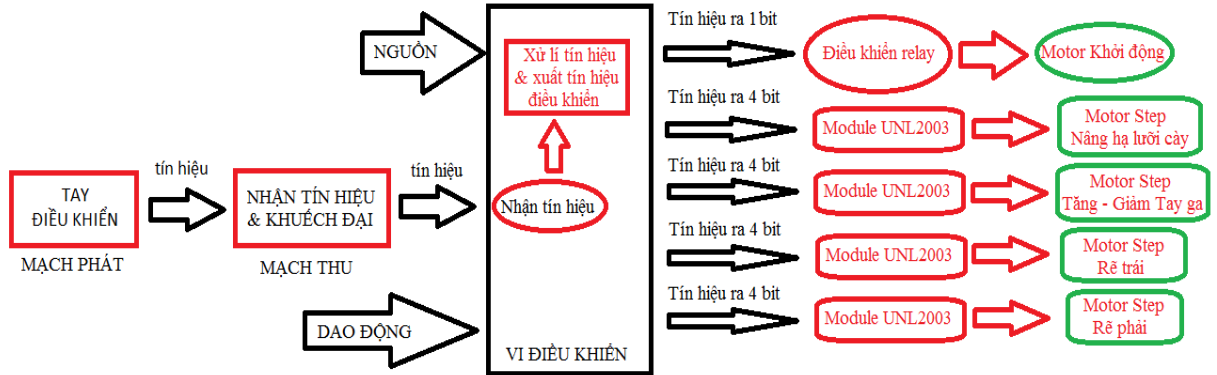
### 3.4 KẾT NỐI MODULE VÀ MOTOR STEP



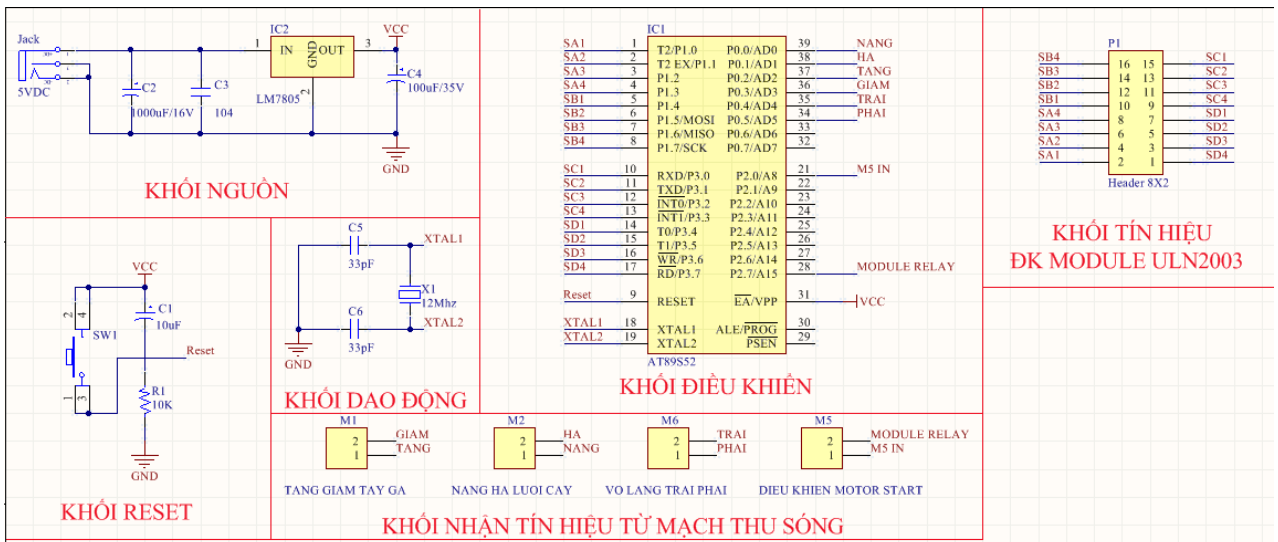
# CHƯƠNG 4: THIẾT KẾ VÀ LẬP TRÌNH

## 4.1 THIẾT KẾ

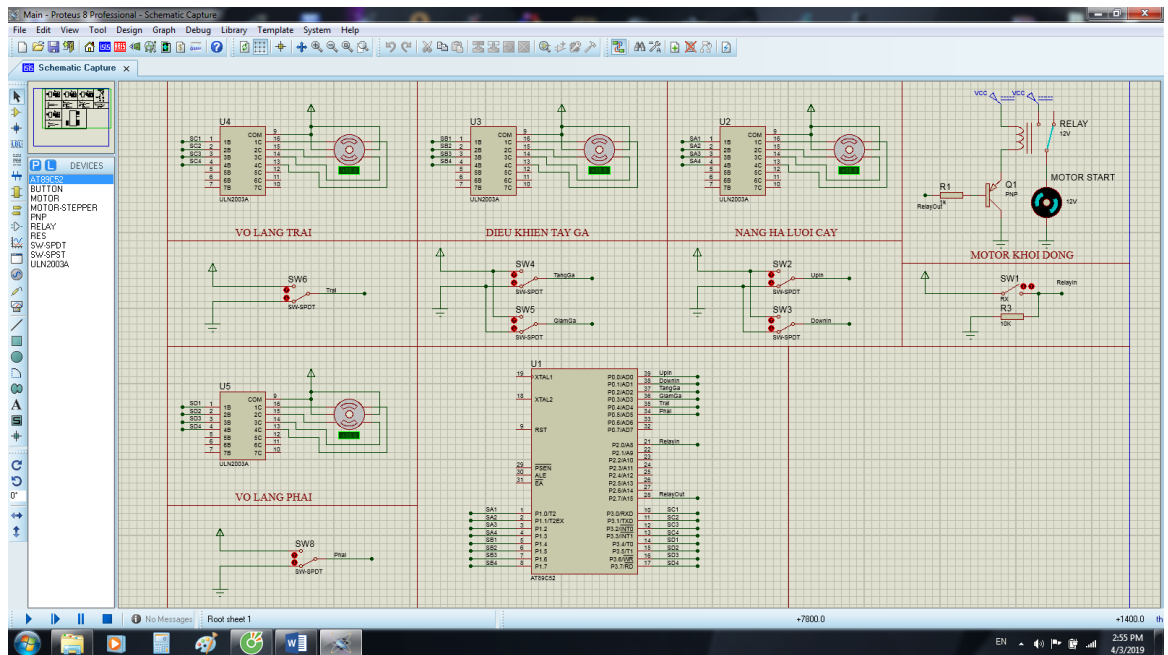
### 4.1.1 Sơ đồ khối



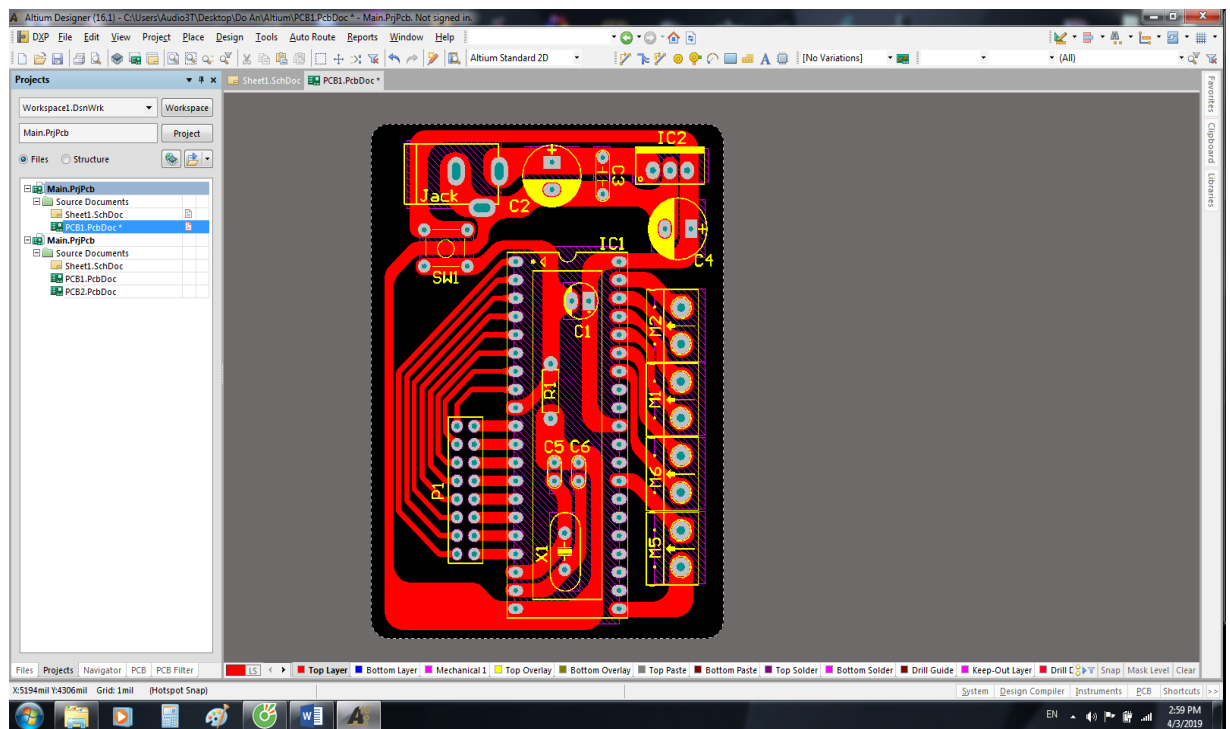
### 4.1.2 Sơ đồ mạch điều khiển trung tâm

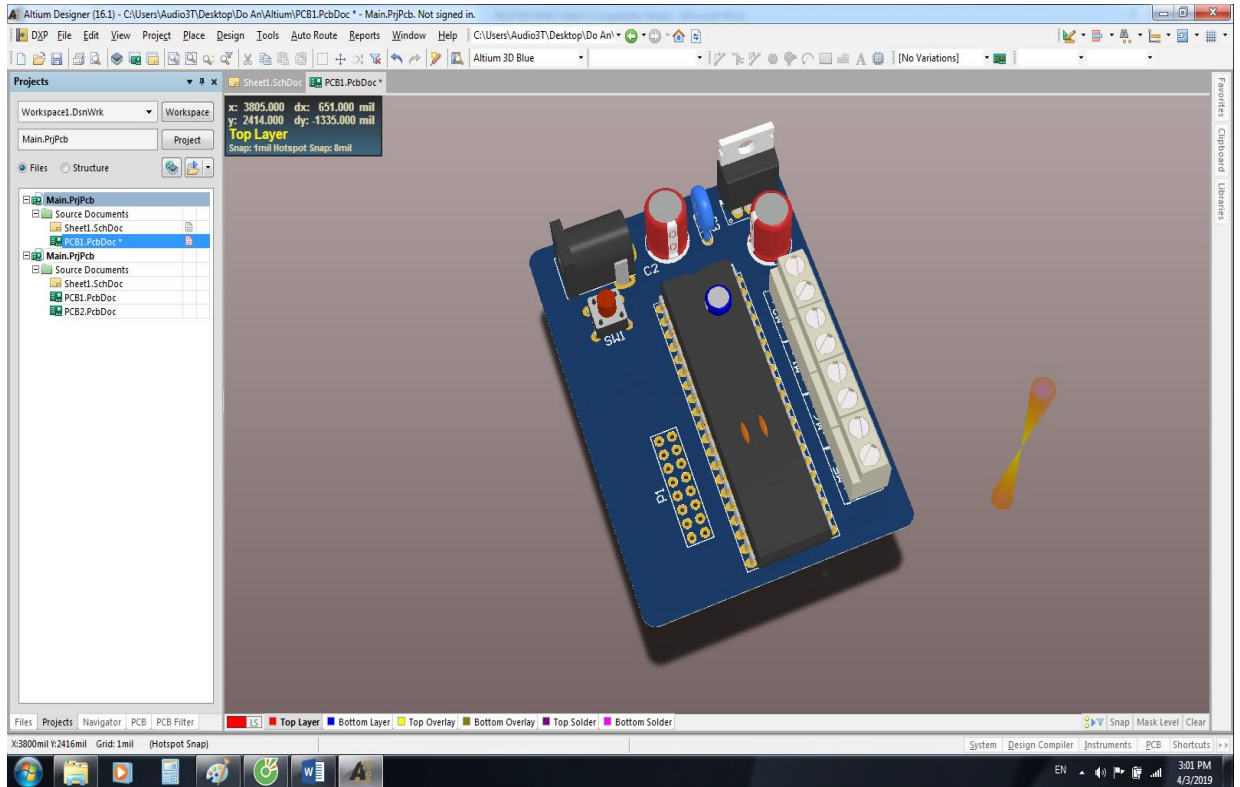


### 4.1.3 Mô phỏng trên phần mềm Proteus



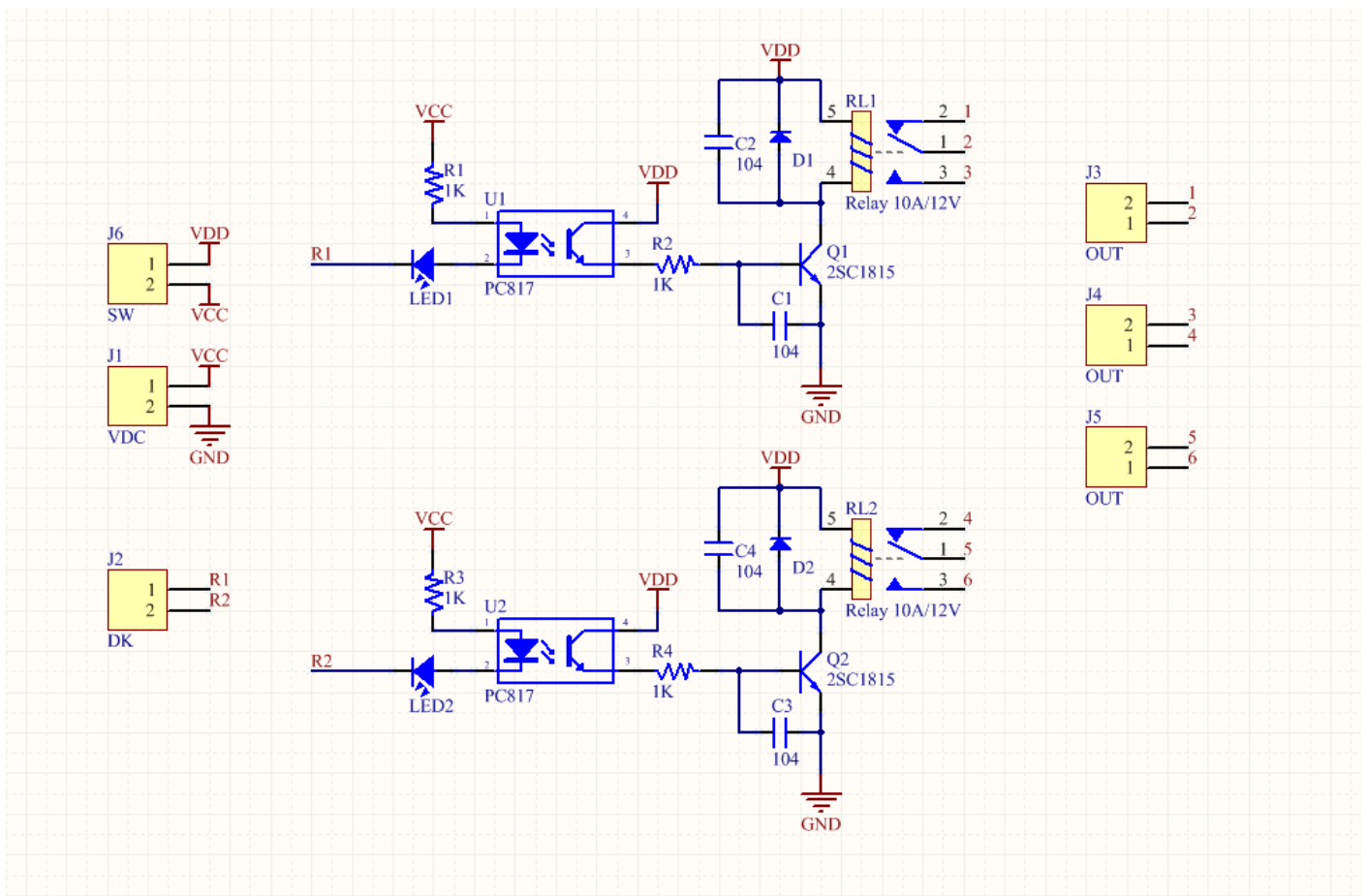
### 4.1.4 Vẽ mạch trên phần mềm Altium



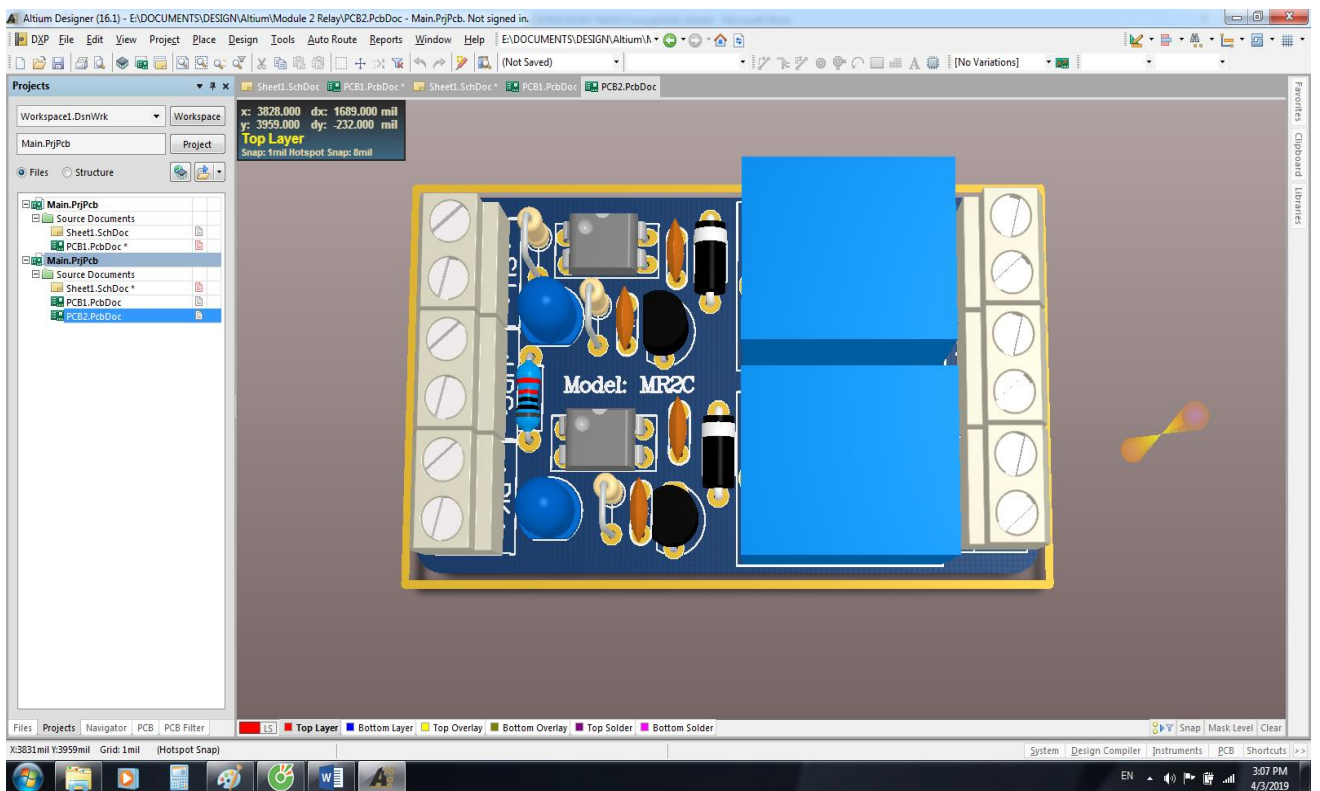
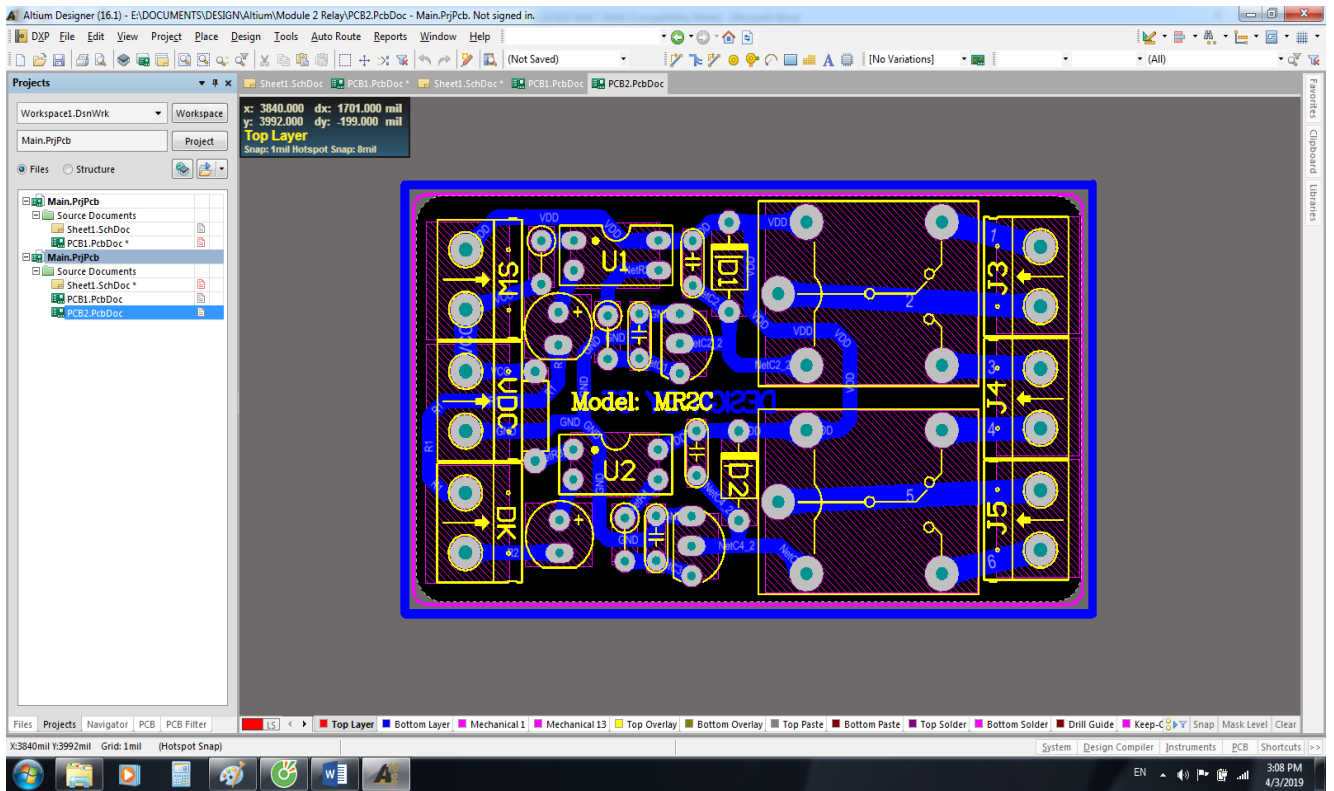


### 4.1.5 Khối xử lí đóng cắt relay

Sơ đồ nguyên lí:



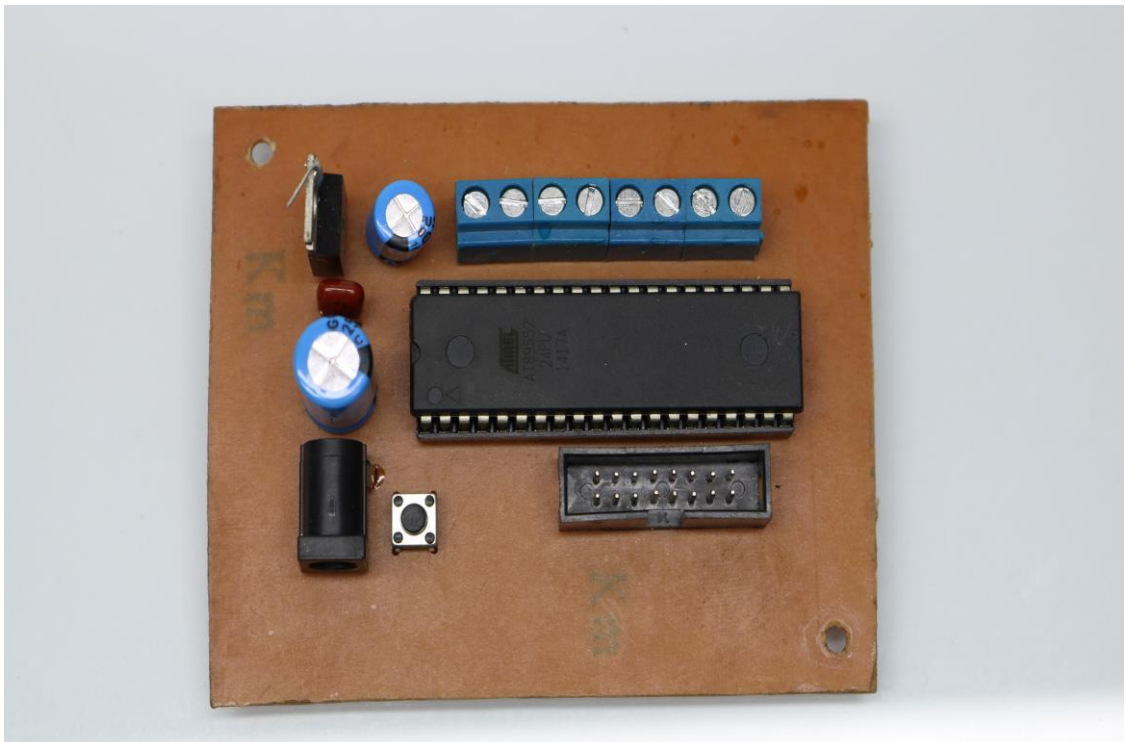
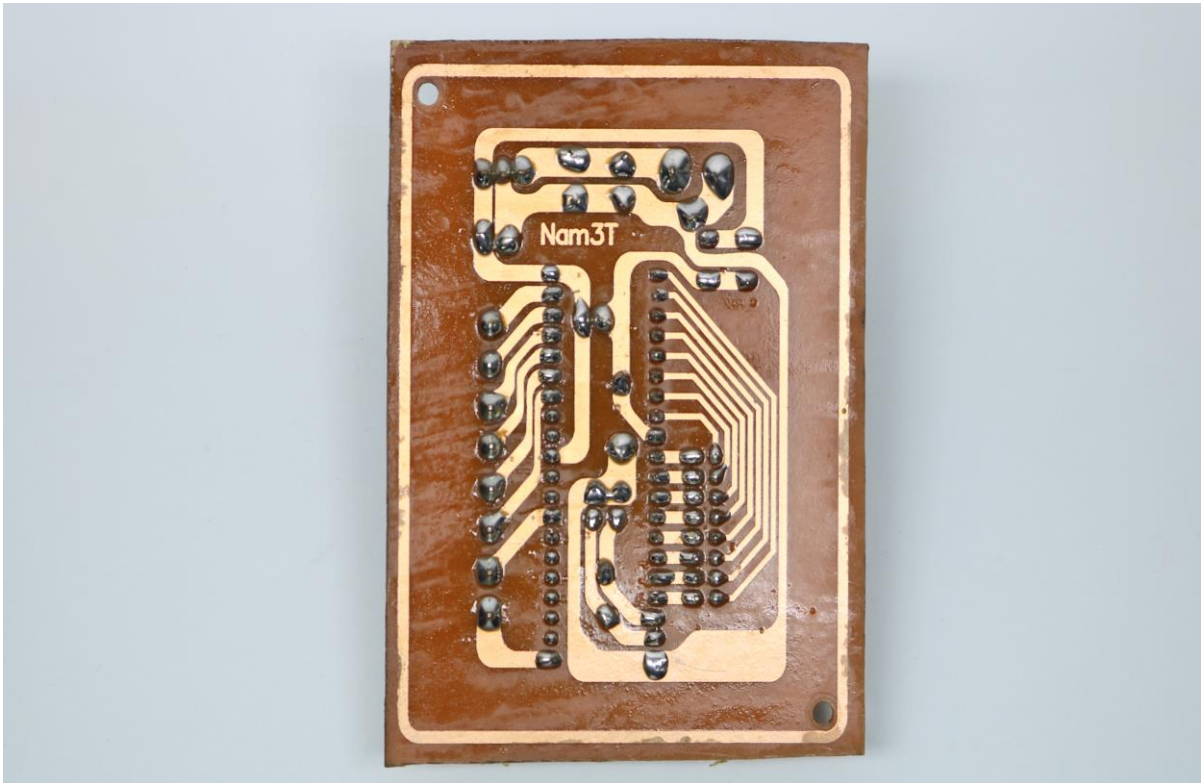
## Vẽ mạch trên Altium :



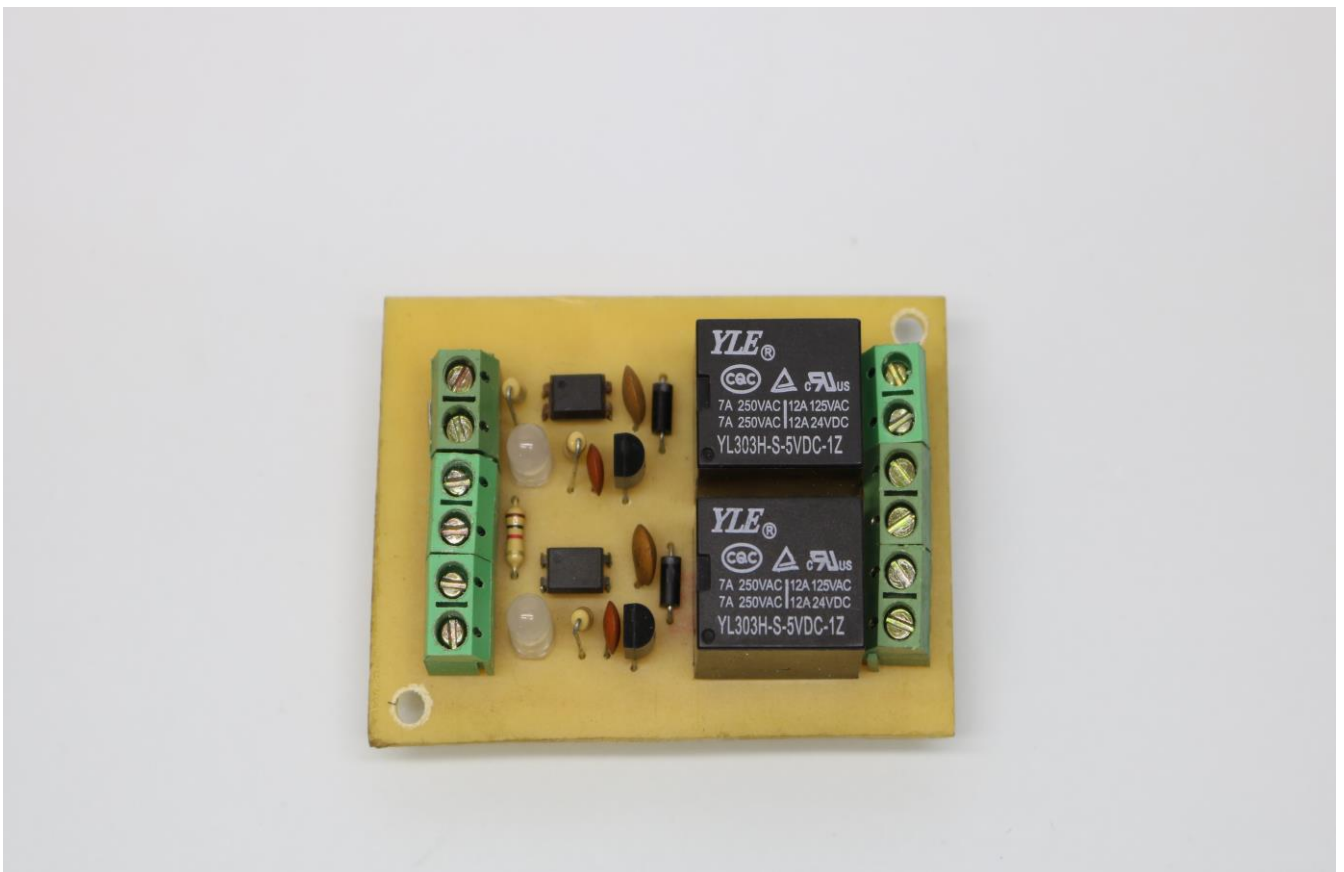
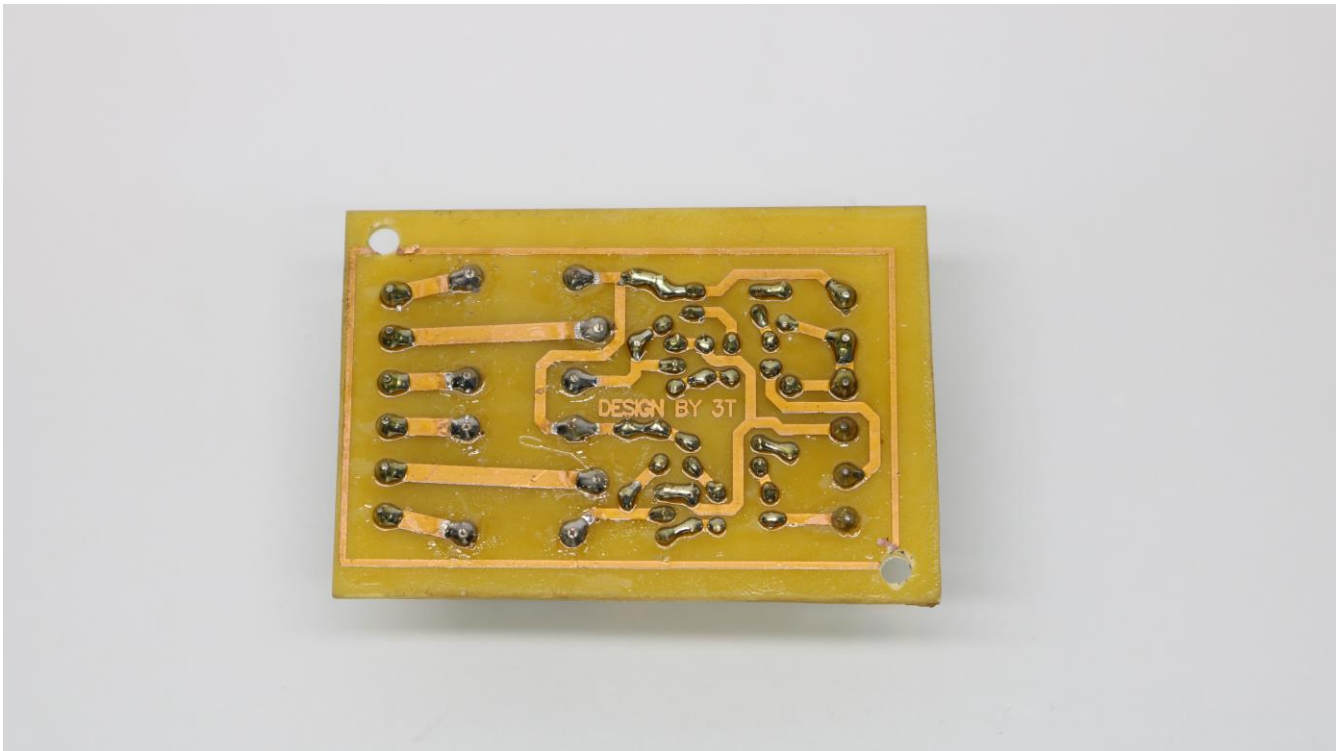
## 4.2 THỰC HIỆN MẠCH THỰC TẾ VÀ LẮP RÁP MODULE

### 4.2.1 Các mạch thiết kế sau khi hoàn thiện

- Mạch điều khiển trung tâm:

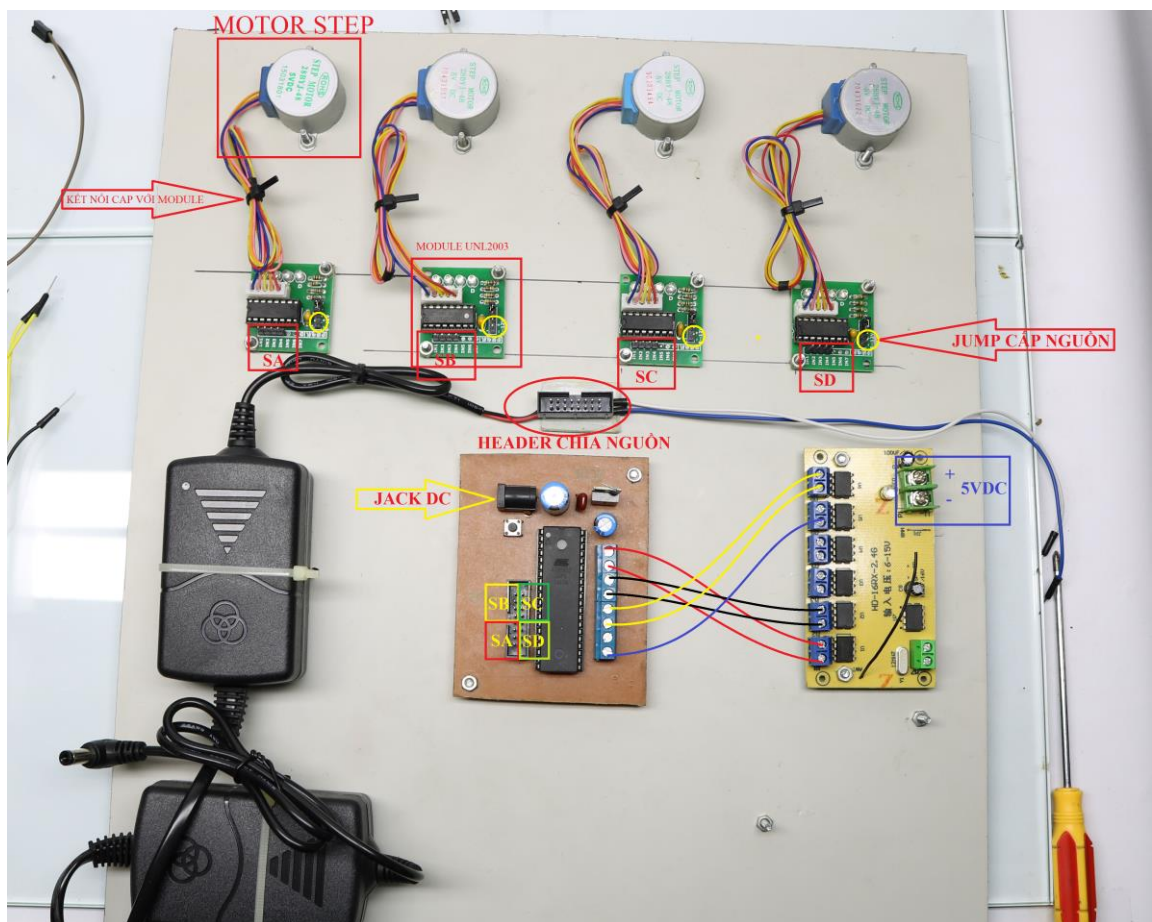


- Mạch Relay

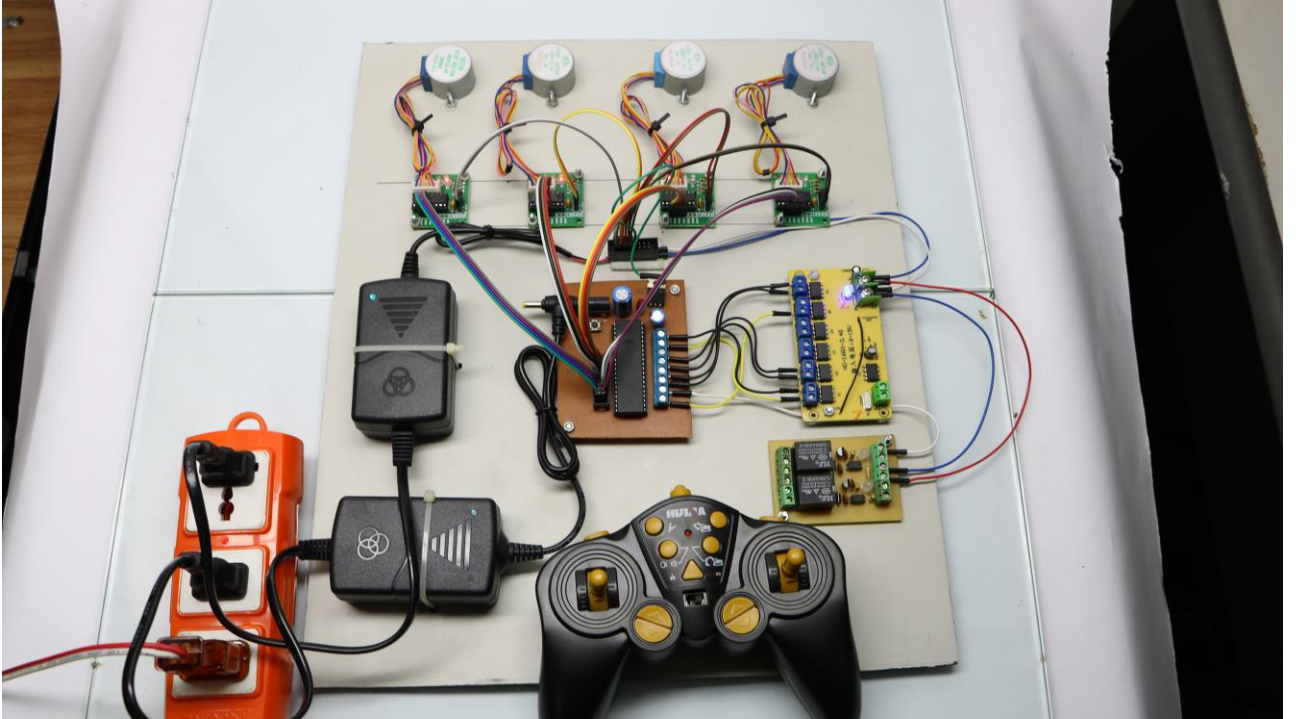
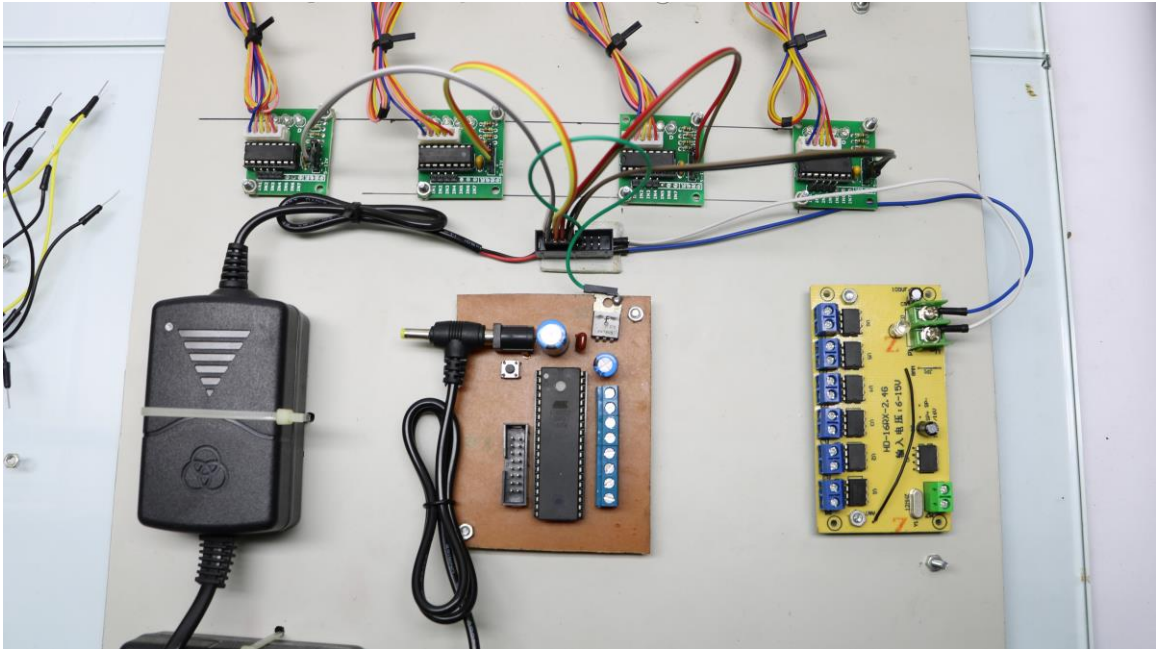


#### 4.2.2 Thực hiện lắp ráp và ghép nối các mạch và Module

- B1: Bắt tất cả các mạch, Motor step, Adapter,.. lên mặt Mica cố định
- B2: Kết nối các cáp từ Motor bước với module UNL2003
- B3: Cấp nguồn 5V cho Module vào các Jump cấp nguồn của Module, Cấp nguồn 5V vào mạch thu sóng, mạch điều khiển và mạch Relay.
- B4: Kết nối các Header tín hiệu ra từ mạch thu RX sang Header nhận tín hiệu của Mạch điều khiển
- B5: Kết nối các cáp tín hiệu điều khiển động cơ bước từ Jump xuất tín hiệu của Mạch điều khiển ra 4 Motor bước.
- B6: Kết nối tín hiệu ra điều khiển Relay từ Mạch điều khiển ra Module Relay.







## 4.3 LẬP TRÌNH

### 4.3.1 Kết Cấu Của Chương Trình

- Khai báo dữ liệu
- Chương trình gồm 14 chương trình con:
  - o Chương trình 1 : Tạo delay 10ms
  - o Chương trình 2 : Điều khiển đóng, cắt Relay để khởi động Motor đề khi có tín hiệu từ tay điều khiển.
  - o Chương trình 3 – 10 : Các chương trình xuất tín hiệu điều khiển 4 Motor bước quay xuôi và quay ngược.
  - o Chương trình 11 : Chương trình điều khiển Motor bước Nâng – Hạ lõi cày khi có tín hiệu từ tay điều khiển.
  - o Chương trình 12 : Chương trình điều khiển Motor bước Tăng – Giảm tay ga khi có tín hiệu từ tay điều khiển.
  - o Chương trình 13 : Chương trình điều khiển Motor bước Vô lăng trái khi có tín hiệu từ tay điều khiển.
  - o Chương trình 14 : Chương trình điều khiển Motor bước Vô lăng phải khi có tín hiệu từ tay điều khiển.
- Chương trình chính : Gọi chương trình 2,11,12,13,14

### 4.3.2 Lập Trình

#### #CODE Lập trình trên phần mềm Keil C

```
#include <REGX51.H>
```

```
#define UpIn          P0_0 //      Tin hieu vao tu RX
```

```
#define DownIn        P0_1 //      Tin hieu vao tu RX
```

```
#define TangGa         P0_2 //      Tin hieu vao tu RX
```

```
#define GiamGa         P0_3 //      Tin hieu vao tu RX
```

```
#define Trai           P0_4 //      Tin hieu vao tu RX
```

```
#define      Phai      P0_5 //      Tin hieu vao tu RX
```

```
#define RelayIn       P2_0 //      Tin hieu vao tu RX
```

```

#define RelayOut          P2_7 //      Tin hieu ra tu VDK
#define SA1                P1_0
#define SA2                P1_1
#define SA3                P1_2
#define SA4                P1_3
#define SB1                P1_4
#define SB2                P1_5
#define SB3                P1_6
#define SB4                P1_7
#define SC1                P3_0
#define SC2                P3_1
#define SC3                P3_2
#define SC4                P3_3
#define SD1                P3_4
#define SD2                P3_5
#define SD3                P3_6
#define SD4                P3_7

```

```

void Delay_10ms(unsigned char t){
    unsigned char i;
    for(i=0;i<t;i++){
        TH0=0xd8; //-10000 us
        TL0=0xf0;
        TR0=1;
        while(!TF0); //cho timer0 tran
        TF0=TR0=0;
    }
}

```

```

}
void Relay(){
    RelayIn=0;
    if(RelayIn==1){
        Delay_10ms(10);
        if(RelayIn==1){
            RelayOut=0;
        }
    }
    else RelayOut=1;
}
void XuoiSA(){
    SA1=1;SA2=0;SA3=0;SA4=0;
    Delay_10ms(20);
    SA1=1;SA2=1;SA3=0;SA4=0;
    Delay_10ms(20);
    SA1=0;SA2=1;SA3=0;SA4=0;
    Delay_10ms(20);
    SA1=0;SA2=1;SA3=1;SA4=0;
    Delay_10ms(20);
    SA1=0;SA2=0;SA3=1;SA4=0;
    Delay_10ms(20);
    SA1=0;SA2=0;SA3=1;SA4=1;
    Delay_10ms(20);
    SA1=0;SA2=0;SA3=0;SA4=1;
    Delay_10ms(20);
    SA1=1;SA2=0;SA3=0;SA4=1;

```

```

    Delay_10ms(20);
}
void NguocSA(){
    SA1=1;SA2=0;SA3=0;SA4=1;
    Delay_10ms(20);
    SA1=0;SA2=0;SA3=0;SA4=1;
    Delay_10ms(20);
    SA1=0;SA2=0;SA3=1;SA4=1;
    Delay_10ms(20);
    SA1=0;SA2=0;SA3=1;SA4=0;
    Delay_10ms(20);
    SA1=0;SA2=1;SA3=1;SA4=0;
    Delay_10ms(20);
    SA1=0;SA2=1;SA3=0;SA4=0;
    Delay_10ms(20);
    SA1=1;SA2=1;SA3=0;SA4=0;
    Delay_10ms(20);
    SA1=1;SA2=0;SA3=0;SA4=0;
    Delay_10ms(20);
}
void XuoiSB(){
    SB1=1,SB2=0,SB3=0,SB4=0;
    Delay_10ms(20);
    SB1=1,SB2=1,SB3=0,SB4=0;
    Delay_10ms(20);
    SB1=0,SB2=1,SB3=0,SB4=0;
    Delay_10ms(20);
}

```

```

    SB1=0,SB2=1,SB3=1,SB4=0;
    Delay_10ms(20);
    SB1=0,SB2=0,SB3=1,SB4=0;
    Delay_10ms(20);
    SB1=0,SB2=0,SB3=1,SB4=1;
    Delay_10ms(20);
    SB1=0,SB2=0,SB3=0,SB4=1;
    Delay_10ms(20);
    SB1=1,SB2=0,SB3=0,SB4=1;
    Delay_10ms(20);
}
void NguocSB(){
    SB1=1,SB2=0,SB3=0,SB4=1;
    Delay_10ms(20);
    SB1=0,SB2=0,SB3=0,SB4=1;
    Delay_10ms(20);
    SB1=0,SB2=0,SB3=1,SB4=1;
    Delay_10ms(20);
    SB1=0,SB2=0,SB3=1,SB4=0;
    Delay_10ms(20);
    SB1=0,SB2=1,SB3=1,SB4=0;
    Delay_10ms(20);
    SB1=0,SB2=1,SB3=0,SB4=0;
    Delay_10ms(20);
    SB1=1,SB2=1,SB3=0,SB4=0;
    Delay_10ms(20);
    SB1=1,SB2=0,SB3=0,SB4=0;

```

```

    Delay_10ms(20);
}
void XuoiSC(){
    SC1=1,SC2=0,SC3=0,SC4=0;
    Delay_10ms(20);
    SC1=1,SC2=1,SC3=0,SC4=0;
    Delay_10ms(20);
    SC1=0,SC2=1,SC3=0,SC4=0;
    Delay_10ms(20);
    SC1=0,SC2=1,SC3=1,SC4=0;
    Delay_10ms(20);
    SC1=0,SC2=0,SC3=1,SC4=0;
    Delay_10ms(20);
    SC1=0,SC2=0,SC3=1,SC4=1;
    Delay_10ms(20);
    SC1=0,SC2=0,SC3=0,SC4=1;
    Delay_10ms(20);
    SC1=1,SC2=0,SC3=0,SC4=1;
    Delay_10ms(20);
}
void NguocSC(){
    SC1=1,SC2=0,SC3=0,SC4=1;
    Delay_10ms(20);
    SC1=0,SC2=0,SC3=0,SC4=1;
    Delay_10ms(20);
    SC1=0,SC2=0,SC3=1,SC4=1;
    Delay_10ms(20);
}

```

```

    SC1=0,SC2=0,SC3=1,SC4=0;
    Delay_10ms(20);
    SC1=0,SC2=1,SC3=1,SC4=0;
    Delay_10ms(20);
    SC1=0,SC2=1,SC3=0,SC4=0;
    Delay_10ms(20);
    SC1=1,SC2=1,SC3=0,SC4=0;
    Delay_10ms(20);
    SC1=1,SC2=0,SC3=0,SC4=0;
    Delay_10ms(20);
}
void XuoiSD(){
    SD1=1,SD2=0,SD3=0,SD4=0;
    Delay_10ms(20);
    SD1=1,SD2=1,SD3=0,SD4=0;
    Delay_10ms(20);
    SD1=0,SD2=1,SD3=0,SD4=0;
    Delay_10ms(20);
    SD1=0,SD2=1,SD3=1,SD4=0;
    Delay_10ms(20);
    SD1=0,SD2=0,SD3=1,SD4=0;
    Delay_10ms(20);
    SD1=0,SD2=0,SD3=1,SD4=1;
    Delay_10ms(20);
    SD1=0,SD2=0,SD3=0,SD4=1;
    Delay_10ms(20);
    SD1=1,SD2=0,SD3=0,SD4=1;

```



```

        Delay_10ms(20);
    }
void NguocSD(){
    SD1=1,SD2=0,SD3=0,SD4=1;
    Delay_10ms(20);
    SD1=0,SD2=0,SD3=0,SD4=1;
    Delay_10ms(20);
    SD1=0,SD2=0,SD3=1,SD4=1;
    Delay_10ms(20);
    SD1=0,SD2=0,SD3=1,SD4=0;
    Delay_10ms(20);
    SD1=0,SD2=1,SD3=1,SD4=0;
    Delay_10ms(20);
    SD1=0,SD2=1,SD3=0,SD4=0;
    Delay_10ms(20);
    SD1=1,SD2=1,SD3=0,SD4=0;
    Delay_10ms(20);
    SD1=1,SD2=0,SD3=0,SD4=0;
    Delay_10ms(20);
}
void UpDownLuoiCay(){
    if(UpIn==1){
        if(UpIn==1){
            XuoiSA();
        }
    }
    if(DownIn==1){

```

```

        if(DownIn==1){
            NguocSA();
        }
    }
}

void TangGiamGa(){
    if(TangGa==1){
        if(TangGa==1){
            XuoiSB();
        }
    }
    if(GiamGa==1){
        if(GiamGa==1){
            NguocSB();
        }
    }
}

void ReTrai(){
    unsigned char i;
    if(Trai==1){
        if(Trai==1){
            for(i=0;i<10;i++){
                XuoiSC();
            }
            if(Trai==1){
                while(Trai==1){}
            }
        }
    }
}

```

```

        if(Trai==0){
            for(i=10;i>0;i--){
                NguocSC();
            }
        }
    }
}

void RePhai(){
    unsigned char i;
    if(Phai==1){
        if(Phai==1){
            for(i=0;i<10;i++){
                XuoiSD();
            }
            if(Phai==1){
                while(Phai==1){}
            }
            if(Phai==0){
                for(i=10;i>0;i--){
                    NguocSD();
                }
            }
        }
    }
}

void main(){

```

```
while(1){  
    Relay();  
    UpDownLuoiCay();  
    TangGiamGa();  
    ReTrai();  
    RePhai();  
}  
}
```

## KẾT LUẬN

Sau khi nghiên cứu qua đề tài: “Điều khiển tự động máy cày sử dụng vi điều khiển 8051” nhờ sự giúp đỡ của thầy giáo “Ths. **Mai Văn Lập**” em đã hiểu thêm nhiều về hệ thống vi điều khiển cũng như công nghệ tự động hóa.

Nhờ sự phát triển của công nghệ tự động hóa, đề tài này không chỉ giúp em hiểu sâu hơn về công nghệ tự động hóa mà còn mang lại nhiều lợi ích cho những người nông dân. Nếu dự án này được áp dụng thì con người sẽ bớt đi công sức và đạt hiệu quả công việc trong công việc cày bừa ruộng đất để trồng lúa, trồng rau màu, ....

# TÀI LIỆU THAM KHẢO

1. Phạm Quang Trí – Giáo trình vi xử lý – Trường ĐH Công Nghiệp TP. HCM.

2. Tống Văn On, Hoàng Đức Hải – Họ vi điều khiển 8051 – NXB Lao Động Và Xã Hội Hà Nội – 2001.

3. Các Webside:

[www.alldatasheets.com](http://www.alldatasheets.com)

[www.diendandientu.com](http://www.diendandientu.com)

<https://tailieu.vn/>

<https://www.youtube.com/>

<https://www.google.com.vn>