

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT



TÀI LIỆU
HƯỚNG DẪN
THỰC HÀNH
VI ĐIỀU KHIỂN

Người biên soạn: Phan Ngọc Kỳ

-Năm 2025-

PHẦN 1 : THỰC HÀNH VI ĐIỀU KHIỂN CƠ BẢN

Bài 1. HƯỚNG DẪN CÁC MODULE CHỨC NĂNG**1.1 GIỚI THIỆU CHUNG**

Bộ thực hành vi điều khiển này được xây dựng nhằm mang đến một môi trường học tập và thực nghiệm toàn diện cho sinh viên ngành kỹ thuật, đặc biệt là lĩnh vực vi điều khiển, điện tử và điều khiển tự động.

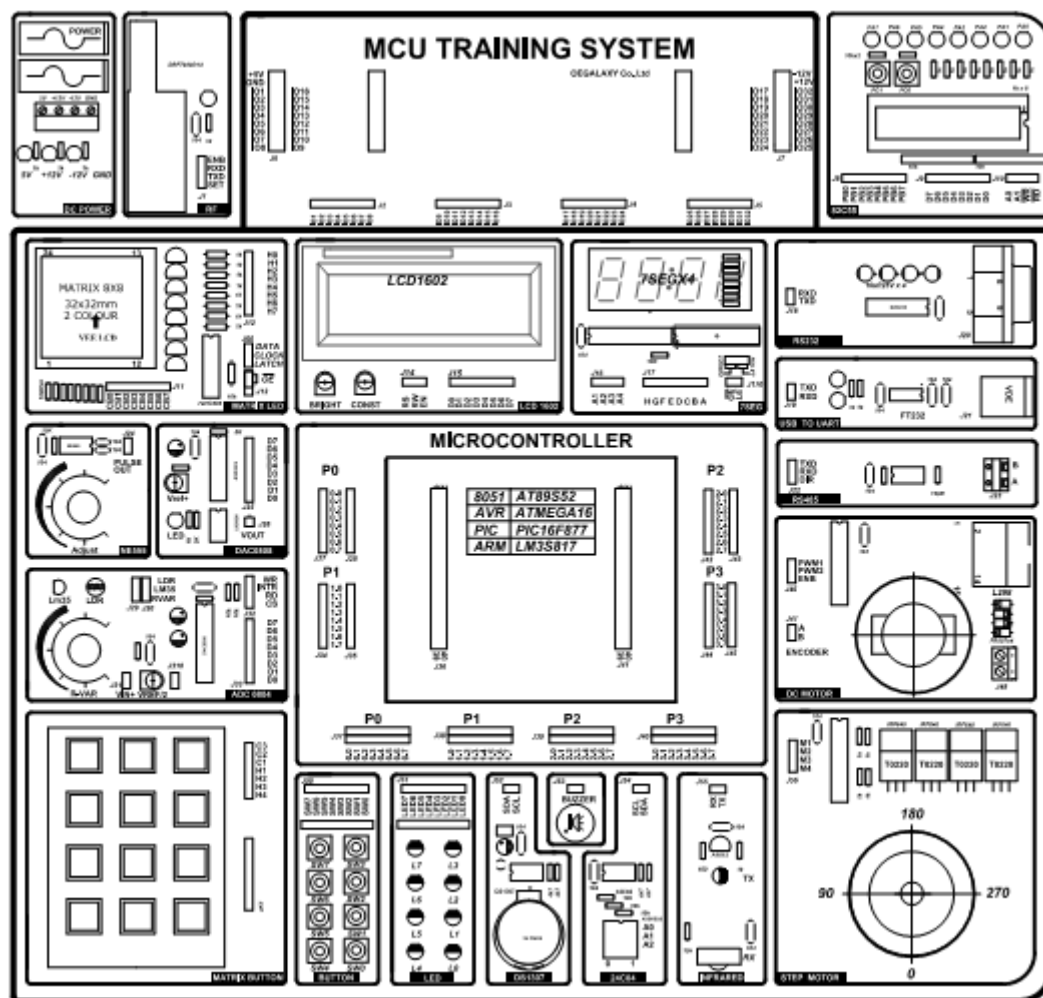
Mục đích cốt lõi là giúp người học không chỉ nắm vững kiến thức lý thuyết mà còn thực hành trực tiếp trên phần cứng thông qua nhiều module riêng biệt—từ hệ thống nguồn, LED, giao tiếp analog/digital (ADC/DAC), phím nhấn, LCD, đến các cơ cấu chấp hành như động cơ bước, động cơ DC với encoder, và các giao thức truyền thông RS-232, RS-485, I²C, RF, cùng với đồng hồ thời gian thực DS1307.

Ý nghĩa giáo dục của bộ thực hành nằm ở việc khuyến khích tư duy mô-đun hóa hệ thống, rèn kỹ năng thiết kế mạch, lập trình xử lý tín hiệu (số và tương tự), và điều khiển cơ – điện tử trong các ứng dụng thực tế như đo nhiệt độ, hiển thị thời gian, giao tiếp không dây và điều khiển động cơ có phản hồi.

1.2 SƠ ĐỒ TỔNG QUÁT CÁC MODULE

Bộ thí nghiệm gồm có 16 Module được bố trí như trên hình với MCU AT89S52

Là khối điều khiển trung tâm gồm một vi điều khiển AT89S52 8 bit, có nghĩa là nó chỉ làm với 8 bit dữ liệu tại một thời điểm. Dữ liệu lớn hơn 8bit được chia thành các dữ liệu 8 bit để xử lý. AT89S52 có 4 cổng vào-ra I/O, mỗi cổng rộng 8 bit, có 8K byte ROM, 256K byte RAM.



1. MODULE NGUỒN

Công suất các đường nguồn như sau:

Tên	Điện áp	Dòng điện max
12V	$12 \pm 15\%$	3,5A
5V	$5 \pm 10\%$	2,5A
-12V	Không cố định -3V đến -15V cung cấp cho module DAC	50mA

- Có 2 cầu chì riêng biệt bảo vệ các đường 12V và 5V trên mạch
- Có các điểm đo/ lấy nguồn, trong trường hợp cần thiết phải cung cấp nguồn điện cho mạch ngoài

2. MODULE LED ĐƠN

Module gồm 8 led đơn 5ly màu đỏ được kết nối với một socket PL tại đầu anode của led

3. MODULE LED 7 ĐOẠN

Module gồm có 4 led 7 đoạn anode chung, các đường dữ liệu tác động ở mức thấp, đường điều khiển tác động ở mức cao. Các socket bao gồm:

Tên	Chức năng
J16	4 bit điều khiển led
J17	8 bit data led 7 đoạn
J170	Dành riêng điều khiển IC chốt dữ liệu 74hc573
J171	Lựa chọn điều khiển led trực tiếp hay thông qua chốt

4. MODULE LCD16X2

Gồm các socket như sau:

Tên	Chức năng
J15	8 bit data LCD
J14	3 bit điều khiển
Biến trở BRIGHT	Điều chỉnh độ sáng màn hình
Biến trở CONST	Điều chỉnh độ tương phản màn hình LCD

5. MODULE ADC0804

ADC0804 là bộ chuyển đổi tương tự sang số 8 bit, đầu vào analog được chọn lựa từ VR, LM335, quang trở thông qua các dip được bật.

Tên	Chức năng
J33	Các chân điều khiển ghi đọc
J32	8 bit ngõ ra số
J29 và J30	Lựa chọn ngõ vào M35: Ngõ vào cảm biến nhiệt LM35 DR: Ngõ vào quang trở - cảm biến ánh sáng VAR: Ngõ vào biến trở
J31	Vin+ : Đầu đo điện áp ngõ vào
Biến trở Vref/2	Điều chỉnh điện áp tham chiếu đầu vào

6. MODULE DAC0808

DAC0808 là bộ chuyển đổi số sang tương tự 8 bit, ngõ vào điện áp chuẩn được thay đổi nhờ vào điều chỉnh VR. Các socket của module là:

Tên	Chức năng
-----	-----------

J25	8 bit ngõ vào số
J26	Đầu ra điện áp biến đổi
Biến trở Vref+	Điều chỉnh dòng điện tham chiếu đầu vào

7. MODULE PHIM NHẤN ĐƠN

Gồm có 8 phím nhấn được kết nối với socket PSW , bình thường ở mức cao, khi tác động ở mức thấp, và có 8 led đơn báo khi phím được tác động.

8. MODULE MA TRẬN PHÍM

Gồm có 12 phím được kết nối thành 4 hàng, 3 cột tạo thành một ma trận 4x3 gồm 7 bit được kết nối với socket J4

9. MODULE THU PHÁT HỒNG NGOẠI

Gồm một led phát hồng ngoại và một con led thu hồng ngoại, khi thu được hồng ngoại thì đầu ra ở mức cao, ngược lại thì ở mức thấp.

Tên	Chức năng
J15	TX : đầu phát RX : đầu thu

10. MODULE ĐỘNG CƠ BƯỚC

Mức tích cực 1 cho phép nối nguồn vào các cuộn dây động cơ bước

Thao tác điều khiển động cơ bước đúng cách, khi động cơ dừng, phải chắc chắn cho tất cả các chân điều khiển về mức tích cực 0.

Nên thao tác điều khiển động cơ với tần số xung từ 100-1000Hz. Tần số xung càng thấp, dòng điện qua động cơ càng lớn gây nên tỏa nhiệt trên các mosfet công suất, trong thời gian dài có thể gây hỏng bộ nguồn cung cấp.

Do điều kiện giải nhiệt, nguồn điện không cho phép, chú ý đến nhiệt độ của các Mosfet công suất trên mạch, sau mỗi lần chạy – không quá 3 phút, phải có thời gian nghỉ giải nhiệt khoảng 3 phút.

Tên	Chức năng
J56	Thứ tự các pha A,B,C,D tương ứng với M1,M2,M3,M4

Gồm một động cơ bước và mạch công suất, các đường điều khiển tác động ở mức cao và được nối vào socket P3.

11. MODULE MỞ RỘNG 8255

Module giao tiếp song song 8255 gồm các socket sau:

Tên	Chức năng
-----	-----------

J8	Port B
J9	8bit dữ liệu
J10	Các chân điều khiển ghi/đọc
8 bóng Led	PA của 8255
2 Nút bấm	PC của 8255

12. MODULE MAX232

Tên	Chức năng
J18	TXD : chân truyền dữ liệu RXD : chân nhận dữ liệu
J19	Kết nối máy tính

13. MODULE MAX485

Tên	Chức năng
J23	Kết nối Bus485
J22	TXD : chân truyền dữ liệu RXD : chân nhận dữ liệu DIR:Chân điều khiển hướng

14. MODULE THỜI GIAN THỰC DS1307

Tên	Chức năng
J52	Chung cho các đường SDA: Chân dữ liệu SCL : Chân clock

15. MODULE RF

Tên	Chức năng
J1	TXD : chân truyền dữ liệu RXD : chân nhận dữ liệu ENB : Chân cho phép SET : Chân cho phép cài đặt

16. MODULE ĐỘNG CƠ DC

Có 2 kênh PWM đóng vai trò như nhau trong việc điều khiển động cơ. Sử dụng 1 kênh điều khiển tốc độ, kênh còn lại sẽ điều khiển chiều động cơ. Tránh việc liên tục đảo chiều động cơ.

Thao tác điều khiển động cơ đúng cách, khi động cơ dừng, phải chắc chắn cho chân ENB về mức 0.

Nên thao tác điều khiển động cơ với tần số xung <12Khz. Tần số xung càng cao, nhiệt lượng tỏa ra trên IC L298 càng cao. Liên tục kiểm tra nhiệt lượng tỏa ra trên IC này.

Tên	Chức năng
J46	PWM1, PWM2 các chân điều khiển tốc độ/chiều ENB : chân cho phép
J47	2 kênh encoder A, B Encoder 500 xung/vòng

17. MODULE LED MA TRẬN

Gồm một led ma trận 8x8 2 màu. Màu xanh dành riêng cho quét ma trận led bằng IC ghi dịch 74hc595

Tên	Chức năng
J11	Điều khiển cột(mức 0)
J12	Điều khiển hàng (mức 1)
J60	Dành riêng cho quét ma trận led bằng 74hc595
J13	Lựa chọn chân cho phép đầu ra của IC 74HC595

Bài 2. HƯỚNG DẪN CÀI ĐẶT VÀ LÀM VIỆC VỚI MÔI TRƯỜNG PHÁT TRIỂN

2.1 Mục đích yêu cầu:

Giúp sinh viên làm quen với thiết bị

Sinh viên tìm hiểu cấu trúc chương trình điều khiển bằng ngôn ngữ ASM và ngôn ngữ C

Quy trình thực hiện viết chương trình, mô phỏng, nạp và thực hành trên phần cứng.

Quy trình thực hiện Debug và cải tiến.

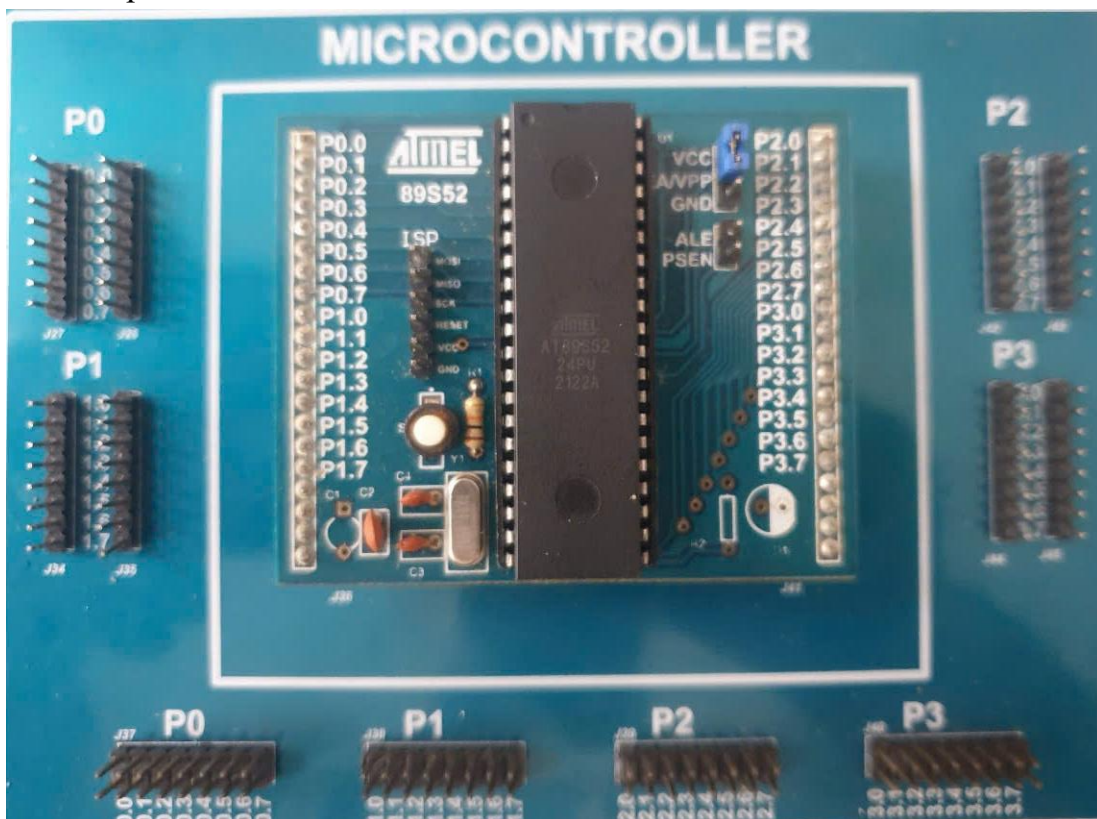
2.2 Chuẩn bị thí nghiệm:

Bộ thực hành vi điều khiển

Máy tính

Phần mềm KeiC (phiên bản sau 4.X), Protues (phiên bản sau 8.X)

Mạch nạp



2.3 Tiến trình thực hành:

2.3.1 Cài đặt phần mềm KeiC (8051), Protues

2.3.2 Cài đặt driver mạch nạp SP200-SE

2.3.3 Thực hiện biên dịch, mô phỏng, nạp và kiểm chứng trên phần cứng

2.3.4 Thử với chương trình trình mẫu ASM

MAIN: MOV A,#01H

BEGIN: MOV P3,A

RL A

CALL DELAY


```

    LJMP BEGIN
    DELAY: PUSH 06
    PUSH 07 ;
    MOV R6,#255
    LAP: MOV R7,#255
    DJNZ R7,$      ; hoặc X: DJNZ R7,X
    DJNZ R6, LAP
    POP 07
    POP 06
    RET
END

```

1. Tạo project ASM từ KeilC
2. Soạn thảo chương trình như đã giới thiệu ở trên
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp và thử nghiệm trên bộ thực hành

1.3.3.1 Thử với chương trình mẫu C

```

#include<AT89x51.h>
#define tat 1
#define sang 0
/*=====khai bao bien=====*/
sbit Led = P1^0; // Khai bao bien Led kieu bit chan P1.0
/*-----ham chinh-----*/
void main(void){
    long i, time=25000;
    while(1){
        Led = tat;
        for (time=0; i<=25000; i++);
        Led = sang;
        while(time--);
    }
}

```

1. Tạo project TestC_tenSV và file TestC_tenSV.c từ KeilC
2. Soạn thảo chương trình như đã giới thiệu ở trên
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.
6. Tối ưu mã nguồn và nhận xét kết quả.

1.4 Bài tập về nhà

1. Phân tích ý nghĩa từng câu lệnh của 2 bài mẫu trên.

2. Phân tích và vẽ lại lưu đồ thuật toán của 2 bài mẫu trên
3. Từ bài mẫu chương trình C, sinh viên tối ưu chương trình theo hướng xây dựng Hàm tạo trẻ.
4. Tạo trước file nguyên lý của bài số 2 trên Protues

Bài 3. Lập trình truy xuất I/O điều khiển LED đơn

3.1 Mục đích

Hiểu nguyên lý điều khiển LED đơn và LED 7 đoạn.

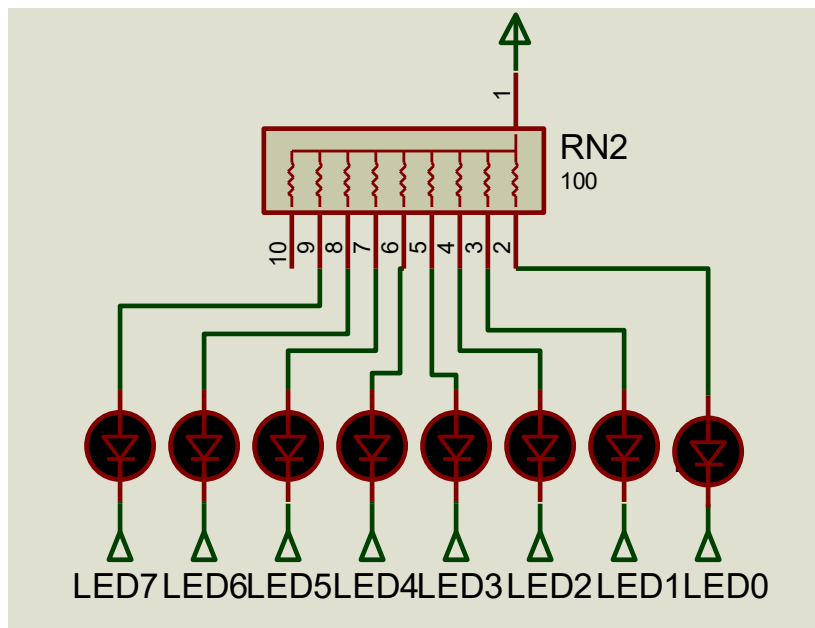
Rèn luyện kỹ năng lập trình xuất tín hiệu điều khiển LED.

2.2 Yêu cầu.

Điều khiển LED đơn sáng/tắt theo chu kỳ định sẵn.

Điều khiển LED đơn sáng/tắt theo hiệu ứng.

2.3 Sơ đồ mạch



2.4 Các bước thực hành:

1. Tạo project từ KeilC
2. Viết chương trình điều khiển 8 LED lần lượt sáng tắt
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.
6. Tối ưu mã nguồn và nhận xét kết quả.

2.4 Bài tập về nhà

1. Phân tích tính và chọn lựa điện trở hạn dòng phù hợp cho LED màu vàng có công suất 1W.

2. Phân tích viết lại chương trình để các LED sáng tắt lần lượt với chu kỳ sáng/tắt khác nhau.

3. Tạo trước file nguyên lý của bài số 3 trên Protues

4. Mô phỏng hoạt động của IC 74573 và IC 74245

Bài 3. Lập trình điều khiển LED 7 đoạn đơn và module LED 7 đoạn 4 số (4 digit)

3.2 Mục đích

Hiểu nguyên lý điều khiển LED 7 đoạn đơn và module LED 7 đoạn 4 số (4 digit).

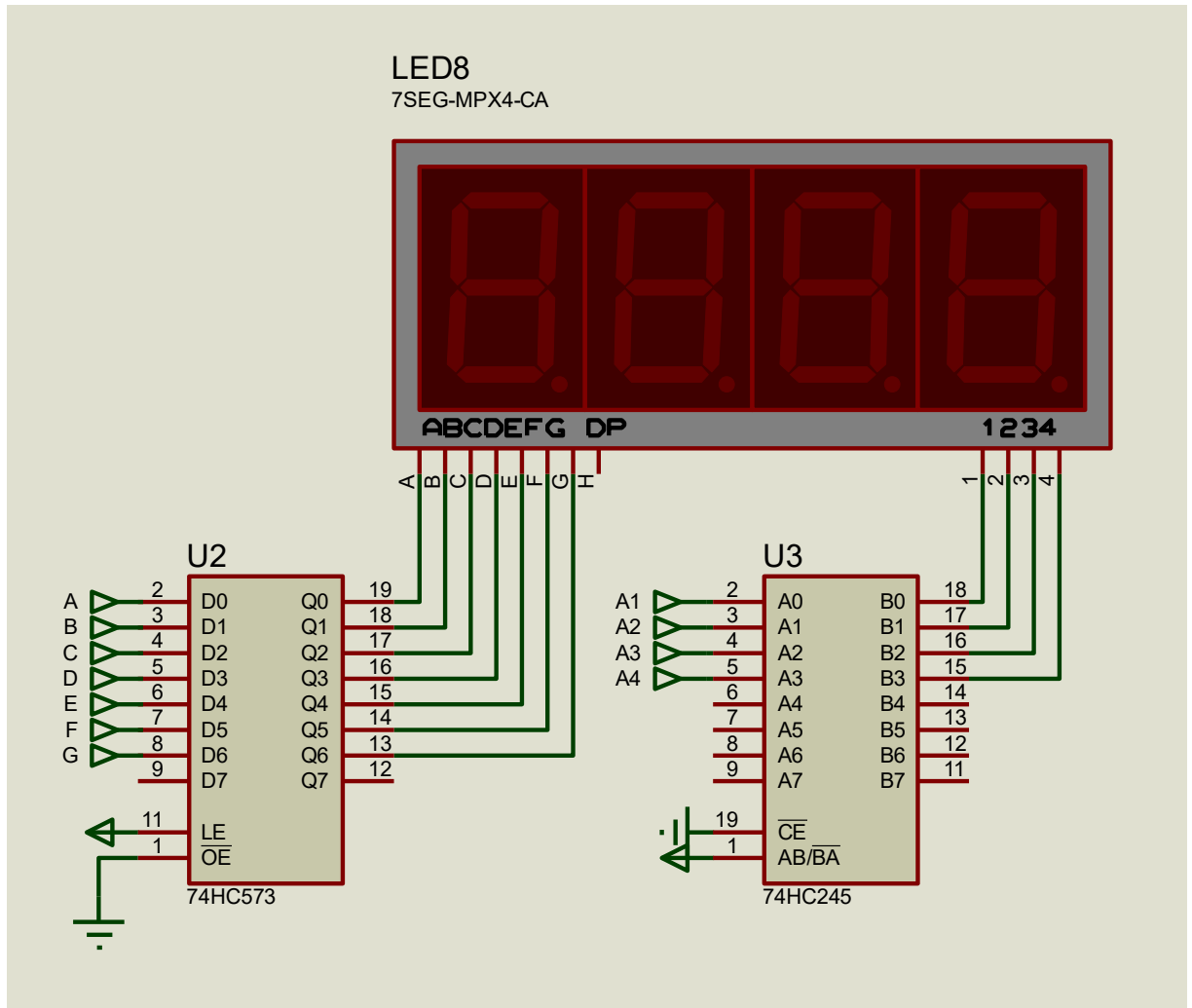
Rèn luyện kỹ năng lập trình xuất tín hiệu điều khiển LED.

3.2 Yêu cầu:

Hiển thị số đếm trên LED 7 đoạn từ 0 đến 9.

Sử dụng phương pháp quét để hiển thị số trên module LED 7 đoạn 4 số.

3.3 Sơ đồ mạch



3.4 Các bước thực hành:

3.4.1 Hiện thị số đếm trên LED 7 đoạn từ 0 đến 9.

1. Tạo project từ KeilC
2. Viết chương trình điều khiển LED thanh hiện thị số từ 0->9 với chu kỳ cố định.
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.
6. Tối ưu mã nguồn và nhận xét kết quả.

3.4.1 Hiện thị số đếm trên LED 7 đoạn từ 0000 đến 9999.

1. Tạo project từ KeilC
2. Viết chương trình điều khiển LED thanh hiện thị số từ 0000 đến 9999 với chu kỳ cố định.
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.
6. Tối ưu mã nguồn và nhận xét kết quả.

3.4 Bài tập về nhà

1. Phân tích tính và chọn lựa điện trở hạn dòng phù hợp cho LED màu vàng có công suất 1W.
2. Phân tích viết lại chương trình để các LED sáng tắt lần lượt với chu kỳ sáng/tắt khác nhau.
3. Tạo trước file nguyên lý của bài số 4 trên Protues

Bài 4. Lập trình đọc tín hiệu từ phím nhấn và điều khiển ngõ ra

3.3 Mục đích

Hiểu cách kết nối và đọc tín hiệu từ phím nhấn.

Thực hành điều khiển LED hoặc thiết bị khác dựa vào tín hiệu phím nhấn

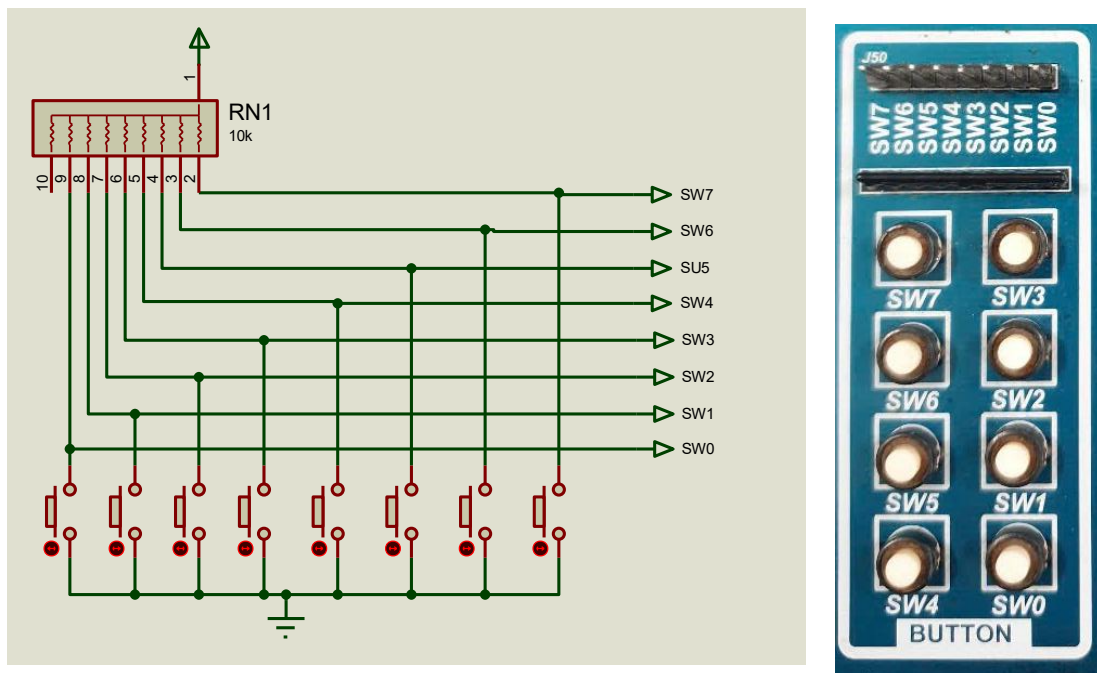
4.2 Yêu cầu.

Khi nhấn nút, LED sáng; khi nhả nút, LED tắt.

Tạo hiệu ứng bật/tắt LED theo từng lần nhấn.

Khắc phục các ảnh hưởng do hiện tượng nảy phím (Debounce) trong thực tế.

4.3 Sơ đồ mạch



4.4 Các bước thực hành:

4.4.1 Viết chương trình điều khiển 8 LED lần lượt sáng tắt

1. Tạo project từ KeilC
2. Viết chương trình điều khiển 8 LED lần lượt sáng tắt khi nhấn nút
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.
6. Tối ưu mã nguồn và nhận xét kết quả.

4.4.2 Viết chương trình điều khiển LED 7 đoạn đếm lên, đếm xuống

1. Tạo project từ KeilC
2. Viết chương trình điều khiển LED 7 đoạn đếm lên tăng/giảm 01 đơn vị mỗi khi nhấn nút.
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.
6. Tối ưu mã nguồn và nhận xét kết quả.

4.4 Bài tập về nhà

1. Phân tích tính và chọn lựa giải pháp chống hiện tượng Debounce ở bàn phím bằng phần cứng.
2. Phân tích viết lại chương trình để khi hoàn tất nhấn (nhấn và nhả phím) các LED thay đổi hiệu ứng.
3. Tạo trước file nguyên lý của bài số 3 trên Protues

Bài 4. Lập trình điều khiển LCD 16x2

4.1 Mục đích

Biết cách giao tiếp và hiển thị dữ liệu lên màn hình LCD 16x2 thông qua việc phân tích Timing Diagram.

Thực hành xuất thông tin lên màn hình dưới dạng ký tự và số.

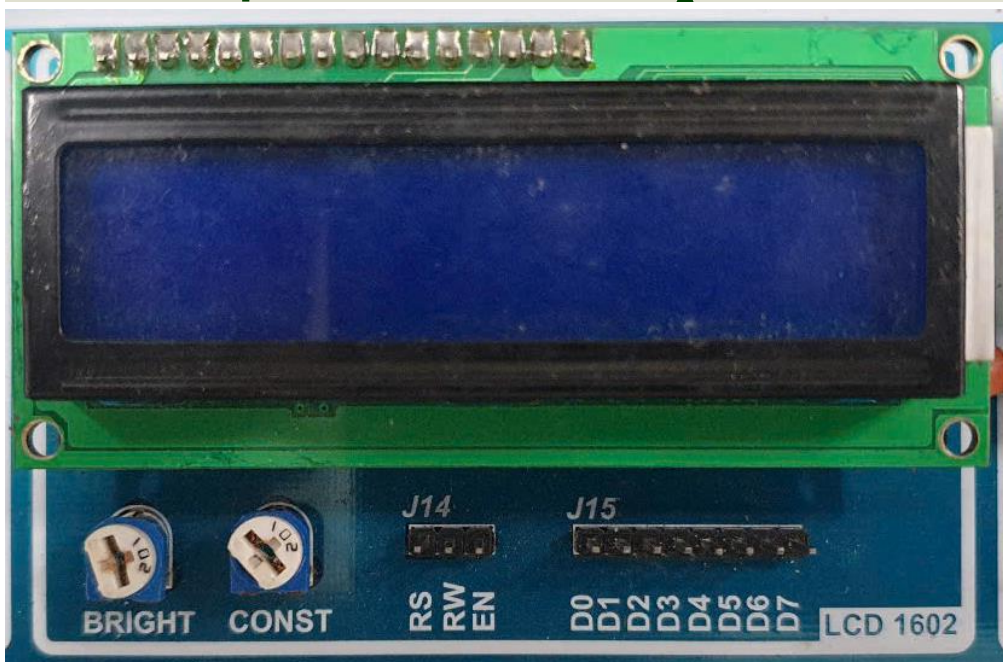
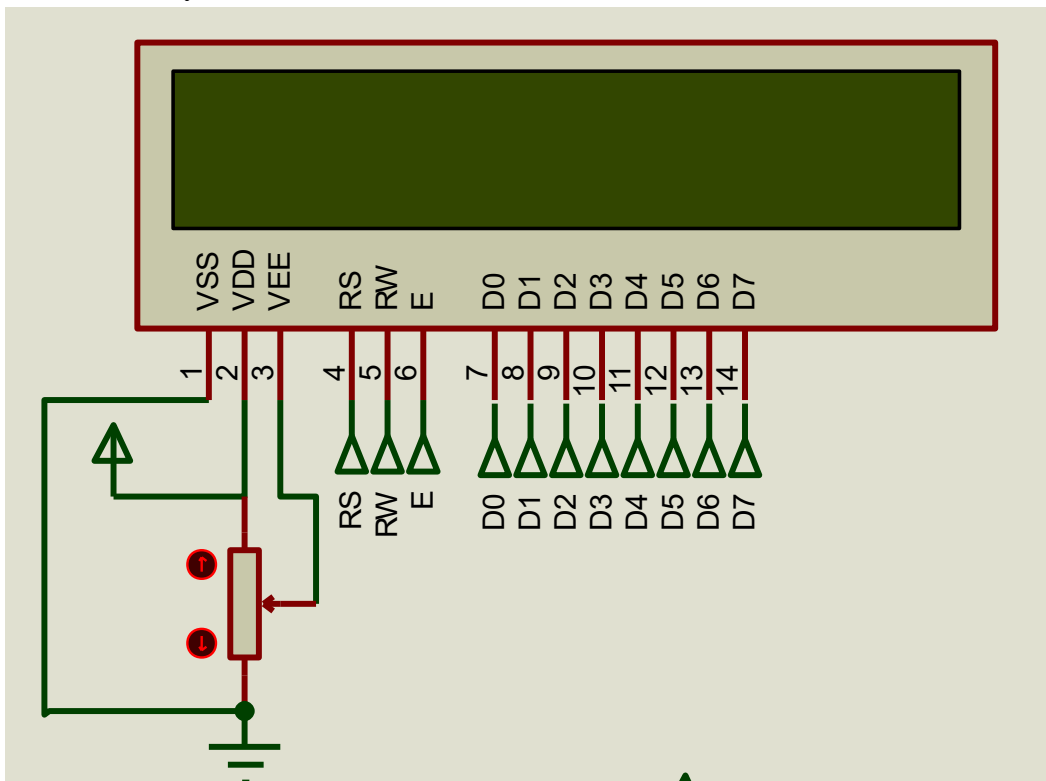
5.2 Yêu cầu.

Đọc và phân tích Timing Diagram của LCD.

Xây dựng các hàm khởi tạo, hàm ghi dữ liệu lên LCD.

Tạo được các hiệu ứng hiển thị trên LCD

5.3 Sơ đồ mạch



5.4 Các bước thực hành:

5.4.1 Hiển thị cơ bản

1. Tạo project từ KeilC
2. Kết nối LCD 16x2 với vi điều khiển theo sơ đồ đúng chuẩn.
3. Viết chương trình khởi tạo LCD.
4. Hiển thị họ và tên ở dòng 1 và ngày tháng năm ở dòng 2.
5. Mô phỏng chương trình trên Proteus.
6. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động.
7. Tối ưu hiển thị và kiểm tra lại hoạt động của LCD

5.4.1 Hiển thị theo hiệu ứng

1. Tạo project từ KeilC
2. Kết nối LCD 16x2 với vi điều khiển theo sơ đồ đúng chuẩn.
3. Hiển thị họ và tên ở dòng 1 và ngày tháng năm ở dòng 2 dịch trái/dịch phải.
4. Mô phỏng chương trình trên Proteus.
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động.
6. Tối ưu hiển thị và kiểm tra lại hoạt động của LCD

5.4 Bài tập về nhà

1. Phân tích tính và chọn lựa giải pháp chống hiện tượng Debounce ở bàn phím bằng phần cứng.
2. Phân tích viết lại chương trình để khi hoàn tất nhấn (nhấn và nhả phím) các LED thay đổi hiệu ứng.
3. Tạo trước file nguyên lý của bài số 7 trên Protues

Bài 5. Lập trình giao tiếp với cảm biến nhiệt độ**5.1 Mục đích**

Hiểu nguyên lý hoạt động của cảm biến nhiệt độ.

Làm việc với các bộ chuyển đổi ADC

Đọc và hiển thị giá trị nhiệt độ lên LCD hoặc LED 7 đoạn.

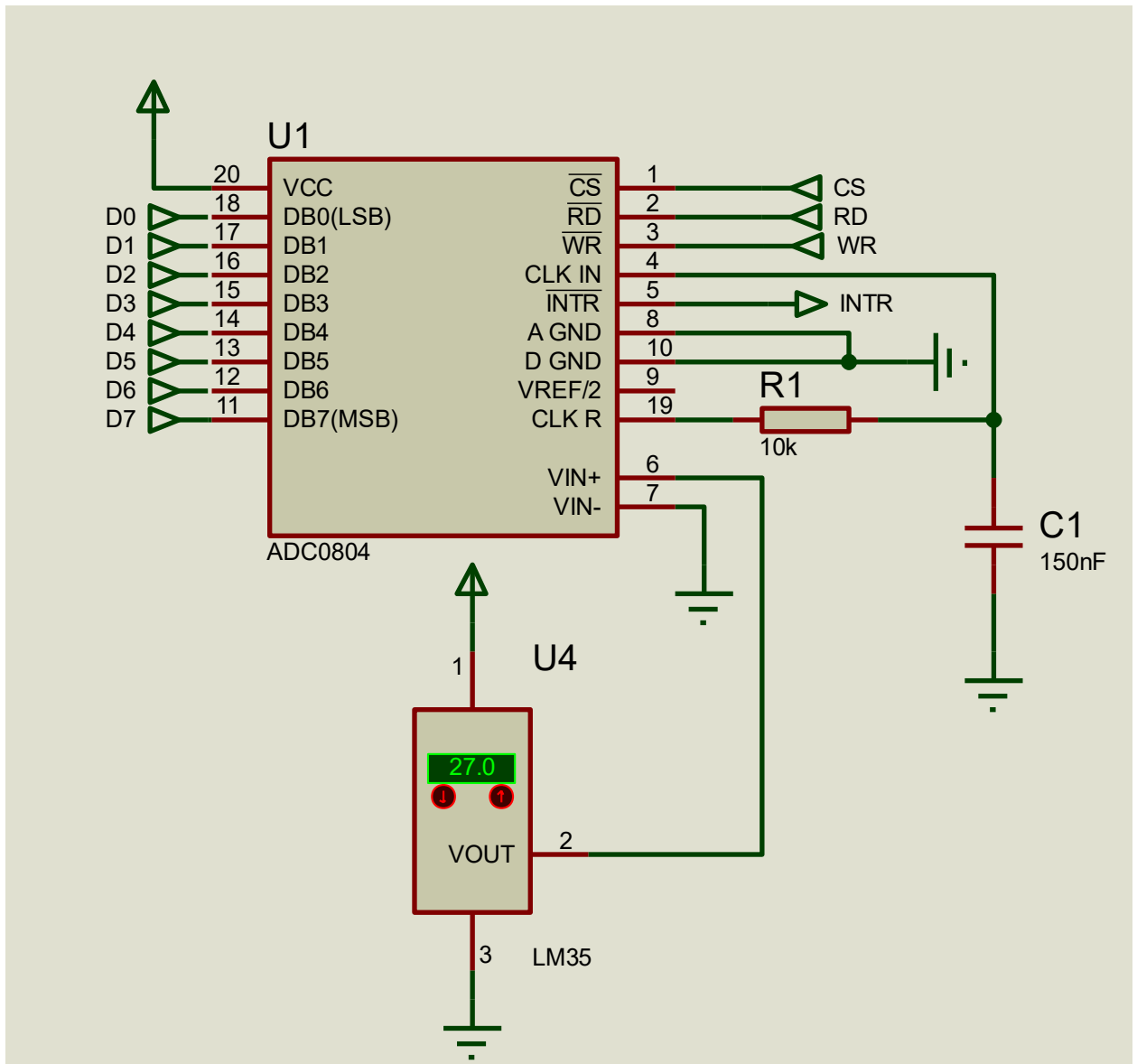
6.2 Yêu cầu.

Đọc nhiệt độ/ánh sáng từ cảm biến.

Chuyển đổi ADC

Hiển thị nhiệt độ theo thời gian thực.

6.3 Sơ đồ mạch



5.4 Các bước thực hành:

1. Tạo project từ KeilC
2. Viết chương trình điều khiển chuyển đổi giá trị nhiệt độ sang dạng hiển thị (LCD hoặc LED 7 đoạn).
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.
6. Tối ưu mã nguồn và nhận xét kết quả.

5.4 Bài tập về nhà

1. Phân tích tính và chọn lựa giải pháp chống hiện tượng Deboud ở bàn phím bằng phản cứng.
2. Phân tích viết lại chương trình để khi hoàn tất nhấn (nhấn và nhả phím) các LED thay đổi hiệu ứng.
3. Tạo trước file nguyên lý của bài số 3 trên Protues

Bài 6. Lập trình sử dụng Timer để tạo xung và định thời

6.1 Mục đích

Hiểu nguyên lý hoạt động của Timer trong vi điều khiển.
Thực hành tạo xung vuông và sử dụng định thời.

7.2 Yêu cầu.

Tạo xung vuông với tần số xác định.
Định thời để LED nhấp nháy theo chu kỳ.

7.3 Sơ đồ mạch

7.4 Các bước thực hành:

1. Tạo project từ KeilC
2. Viết chương trình điều khiển 8 LED lần lượt sáng tắt khi nhấn phím
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues
5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.
6. Tối ưu mã nguồn và nhận xét kết quả.

7.4 Bài tập về nhà

1. Phân tích tính và chọn lựa giải pháp chống hiện tượng Deboud ở bàn phím bằng phản cứng.
2. Phân tích viết lại chương trình để khi hoàn tất nhấn (nhấn và nhả phím) các LED thay đổi hiệu ứng.
3. Tạo trước file nguyên lý của bài số 3 trên Protues

Bài 7. Điều khiển động cơ

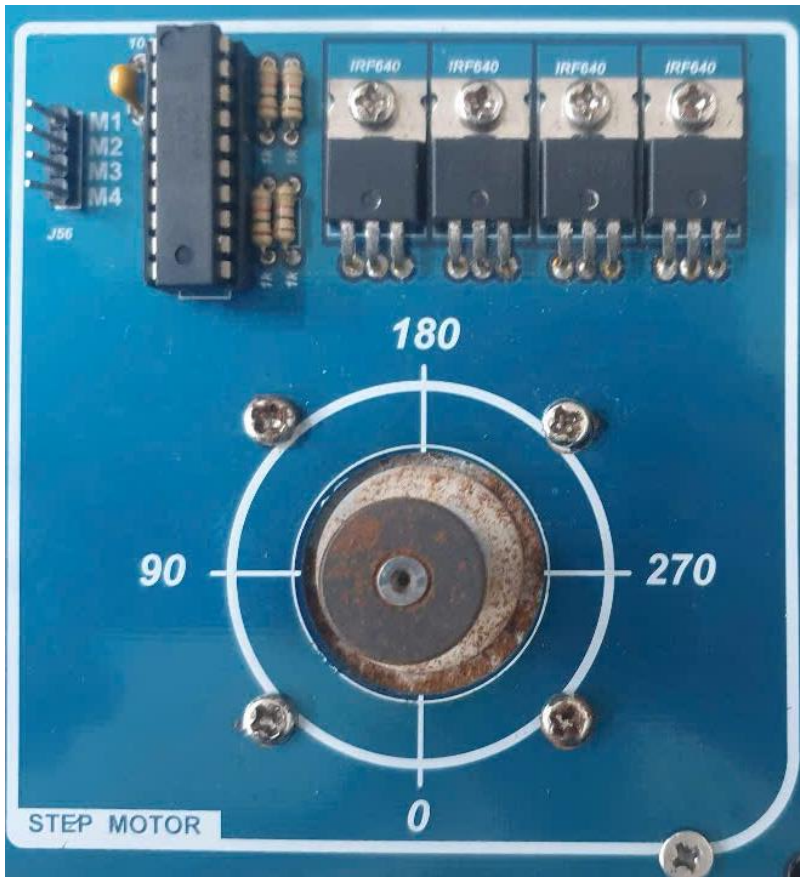
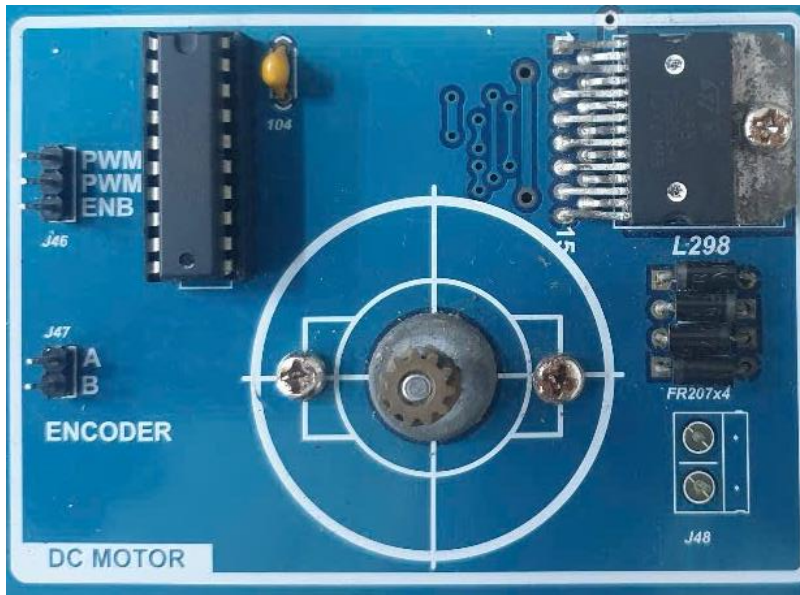
8.1 Mục đích

Hiểu nguyên lý điều khiển động cơ DC và động cơ bước.
Thực hành điều chỉnh tốc độ và hướng quay động cơ.

8.2 Yêu cầu.

Điều khiển động cơ DC chạy tiến/lùi.
Điều khiển động cơ bước quay theo góc xác định.

8.3 Sơ đồ mạch



8.4 Các bước thực hành:

1. Tạo project từ KeilC
2. **Kết nối động cơ DC và động cơ bước với vi điều khiển.**
 - **Viết chương trình điều khiển động cơ DC bằng PWM.**
 - **Viết chương trình điều khiển động cơ bước** theo từng bước hoặc theo góc quay. Viết chương trình điều khiển 8 LED lần lượt sáng tắt khi nhấn phím
3. Kiểm tra lỗi và biên dịch
4. Nạp và mô phỏng trên Protues

5. Nạp chương trình vào vi điều khiển, kiểm tra hoạt động thực tế.

6. Tối ưu mã nguồn và nhận xét kết quả.

8.5 Bài tập về nhà

1. Phân tích tính và chọn lựa giải pháp chống hiện tượng Deboud ở bàn phím bằng phần cứng.

2. Phân tích viết lại chương trình để khi hoàn tất nhấn (nhấn và nhả phím) các LED thay đổi hiệu ứng.

3. Tạo trước file nguyên lý của bài số 3 trên Protues

