

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC HÀNH MÔN VI XỬ LÝ – VI ĐIỀU KHIỂN
BÀI THỰC HÀNH SỐ 3: LÀM QUEN VỚI PROTEUS VÀ
HỌ VI ĐIỀU KHIỂN 8051

Sinh viên thực hiện:

Trần Ngọc Ánh

22520077

Giảng viên hướng dẫn: Phạm Minh Quân

Mã lớp: CE103.O22

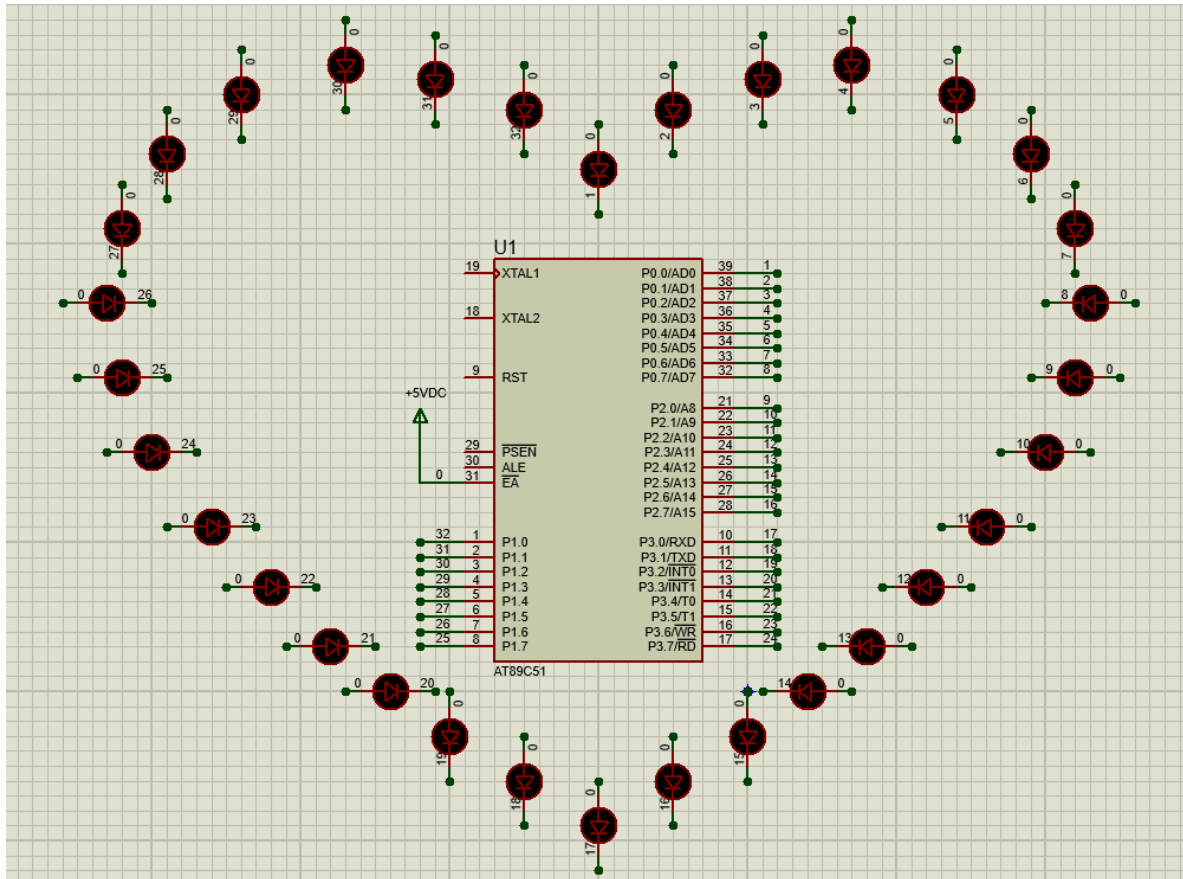
TP. HỒ CHÍ MINH, 8 THÁNG 4 NĂM 2024

MỤC LỤC

I. Thực hiện thiết kế 1 mạch led trái tim gồm 32 led và được điều khiển bởi AT89C51....	3
II. Lập trình Assembly để mạch có thể chạy liên tục ít nhất 3 hiệu ứng, mỗi hiệu ứng chạy trong vòng 05 giây.....	3
1. Source code:	3
2. Kết quả:	14
III. Giải thích nguyên lí hoạt động của từng hiệu ứng:	14
Hiệu ứng 1:	14
Hiệu ứng 2:	15
Hiệu ứng 3:	16
Hiệu ứng 4:	17
Hiệu ứng 5:	18
Hiệu ứng 6:	19
Hiệu ứng 7:	20
Hiệu ứng 8:	22
Hiệu ứng 9:	24
Hiệu ứng 10:	24
III. Tìm hiểu và viết các bước thực hiện mạch in từ bước thiết kế trên proteus đến bước ra được mạch in trong thực tế.....	26

I. Thực hiện thiết kế 1 mạch led trái tim gồm 32 led và được điều khiển bởi AT89C51.

Bên dưới là schematic được tạo từ 32 led *màu đỏ* và vi điều khiển AT89C51. Các linh kiện (led) được đánh số từ 1 đến 32 tương ứng với các chân của vi điều khiển và một nguồn điện một chiều 5V.



II. Lập trình Assembly để mạch có thể chạy liên tục ít nhất 3 hiệu ứng, mỗi hiệu ứng chạy trong vòng 05 giây.

1. Source code:

```
ORG 00H
MAIN:
CALL EFFECT1
CALL EFFECT2
CALL EFFECT3
CALL EFFECT4
CALL EFFECT5
CALL EFFECT6
CALL EFFECT7
```

```

CALL EFFECT8
CALL EFFECT9
CALL EFFECT10
JMP MAIN

EFFECT1:      ; cháy đơn
MOV A,#00000011B
LOOP0:
MOV P0,A
CALL DELAY
RL A
CJNE A,#00000011B,LOOP0
MOV A,#00000011B
LOOP1:
MOV P2,A
CALL DELAY
RL A
CJNE A,#00000011B,LOOP1
MOV A,#00000011B
LOOP2:
MOV P3,A
CALL DELAY
RL A
CJNE A,#00000011B,LOOP2
MOV A,#00000011B
LOOP3:
MOV P1,A
CALL DELAY
RL A
CJNE A,#00000011B,LOOP3
RET

EFFECT2:      ; sáng dần
MOV A,#00H
LOOP4:
SETB C
RLC A
MOV P0,A
CALL DELAY
CJNE A,#0FFH,LOOP4
MOV A,#00H
LOOP5:

```

```

SETB C
RLC A
MOV P2,A
CALL DELAY
CJNE A,#0FFH,LOOP5
MOV A,#00H
LOOP6:
SETB C
RLC A
MOV P3,A
CALL DELAY
CJNE A,#0FFH,LOOP6
MOV A,#00H
LOOP7:
SETB C
RRC A
MOV P1,A
CALL DELAY
CJNE A,#0FFH,LOOP7
RET

EFFECT3:    ; tắt dần
MOV A,#0FFH
LOOP8:
CLR C
RLC A
MOV P0,A
CALL DELAY
CJNE A,#00H,LOOP8
MOV A,#0FFH
LOOP9:
CLR C
RLC A
MOV P2,A
CALL DELAY
CJNE A,#00H,LOOP9
MOV A,#0FFH
LOOP10:
CLR C
RLC A
MOV P3,A
CALL DELAY

```

```

    CJNE A,#00H,LOOP10
    MOV A,#0FFH
LOOP11:
    CLR C
    RRC A
    MOV P1,A
    CALL DELAY
    CJNE A,#00H,LOOP11
    RET

EFFECT4:    ; sáng từ trên xuống
    MOV A,#00H
LOOP12:
    SETB C
    RLC A
    MOV P0,A
    MOV P1,A
    CALL DELAY
    CJNE A,#0FFH,LOOP12
    MOV R6,#00H
    MOV R7,#00H
LOOP13:
    SETB C
    MOV A,R6
    RRC A
    MOV R6,A
    MOV R6,A
    MOV P3,R6
    CALL DELAY
    SETB C
    MOV A,R7
    RLC A
    MOV R7,A
    MOV P2,R7
    CALL DELAY
    CJNE R7,#0FFH,LOOP13
    RET

EFFECT5:    ; tắt trái tim
    MOV R6,#0FFH
    MOV R7,#0FFH

```

LOOP14:

```
CLR C
MOV A,R6
RLC A
MOV R6,A
MOV P3,R6
CALL DELAY
CLR C
MOV A,R7
RRC A
MOV R7,A
MOV P2,R7
CALL DELAY
CJNE R7,#00H,LOOP14
MOV A,#0FFH
```

LOOP15:

```
CLR C
RRC A
MOV P0,A
MOV P1,A
CALL DELAY
CJNE A,#00H,LOOP15
RET
```

EFFECT6: ; tim sáng lên xuống

```
MOV A,#00H
```

LOOP16:

```
SETB C
RLC A
MOV P0,A
MOV P1,A
```

```
CALL DELAY
CJNE A,#0FFH,LOOP16
MOV R6,#00H
MOV R7,#00H
```

LOOP17:

```
SETB C
MOV A,R6
RLC A
MOV R6,A
MOV P3,R6
CALL DELAY
SETB C
MOV A,R7
RRC A
MOV R7,A
MOV P2,R7
CALL DELAY
CJNE R7,#0FFH,LOOP17
RET
```

EFFECT7: ; sáng quay trái

```
MOV R6,#00H
MOV R7,#00H
```

LOOP18:

```
SETB C
MOV A,R6
RLC A
MOV R6,A
MOV P1,R6
CALL DELAY
SETB C
```



```

MOV A,R7
RRC A
MOV R7,A
MOV P2,R7
CALL DELAY
CJNE R7,#0FFH,LOOP18
MOV A,#00H
LOOP19:
SETB C
RRC A
MOV P0,A
MOV P3,A
CALL DELAY
CJNE A,#0FFH,LOOP19
MOV R6,#0FFH
MOV R7,#0FFH
LOOP20:
CLR C
MOV A,R6
RLC A
MOV R6,A
MOV P1,R6
CALL DELAY
CLR C
MOV A,R7
RRC A
MOV R7,A
MOV P2,R7
CALL DELAY
CJNE R7,#00H,LOOP20
MOV A,#0FFH

```

LOOP21:

```
CLR C
RRC A
MOV P0,A
MOV P3,A
CALL DELAY
CJNE A,#00H,LOOP21
RET
```

EFFECT8: ; sáng quay phải

```
MOV A,#00H
```

LOOP22:

```
SETB C
RLC A
MOV P0,A
MOV P3,A
CALL DELAY
CJNE A,#0FFH,LOOP22
MOV R6,#00H
MOV R7,#00H
```

LOOP23:

```
SETB C
MOV A,R6
RLC A
MOV R6,A
MOV P2,R6
CALL DELAY
SETB C
MOV A,R7
RRC A
MOV R7,A
```

```
MOV P1,R7
CALL DELAY
CJNE R7,#0FFH,LOOP23
MOV A,#0FFH
```

LOOP24:

```
CLR C
RLC A
MOV P0,A
MOV P3,A
CALL DELAY
CJNE A,#00H,LOOP24
MOV R6,#0FFH
MOV R7,#0FFH
```

LOOP25:

```
CLR C
MOV A,R6
RLC A
MOV R6,A
MOV P2,R6
CALL DELAY
CLR C
MOV A,R7
RRC A
MOV R7,A
MOV P1,R7
CALL DELAY
CJNE R7,#00H,LOOP25
RET
```

EFFECT9: ; nhấp nháy

```
MOV R7,#5
```

TIMES:

```
MOV P0,#00H
MOV P1,#00H
MOV P2,#00H
MOV P3,#00H
CALL DELAY
MOV P0,#0FFH
MOV P1,#0FFH
MOV P2,#0FFH
MOV P3,#0FFH
CALL DELAY
DJNZ R7,TIMES
RET
```

EFFECT10: ; nhịp tim kép

```
MOV A,#00H
```

LOOP26:

```
SETB C
RLC A
MOV P0,A
MOV P1,A
CALL DELAY
CJNE A,#0FFH,LOOP26
MOV R6,#00H
MOV R7,#00H
```

LOOP27:

```
SETB C
MOV A,R6
RRC A
MOV R6,A
MOV P3,R6
```

```

CALL DELAY
SETB C
MOV A,R7
RLC A
MOV R7,A
MOV P2,R7
CALL DELAY
CJNE R7,#0FFH,LOOP27
MOV R6,#00H
MOV R7,#00H
LOOP28:
    SETB C
    MOV A,R6
    RLC A
    MOV R6,A
    MOV P3,R6
    CALL DELAY
    SETB C
    MOV A,R7
    RRC A
    MOV R7,A
    MOV P2,R7
    CALL DELAY
    CJNE R7,#0FFH,LOOP28
    MOV A,#00H
LOOP29:
    SETB C
    RLC A
    MOV P0,A
    MOV P1,A
    CALL DELAY

```

```

    CJNE A,#0FFH,LOOP29
    RET

DELAY:
    MOV R1,#150
LOOP30:
    MOV R0,#150
LOOP31:
    NOP
    NOP
    NOP
    DJNZ R0,LOOP31
    DJNZ R1,LOOP30
    RET
END

```

2. Kết quả:

Video chứng minh tính hoạt động của các hiệu ứng trong thiết kế trên được đính kèm link dưới đây:

<https://drive.google.com/file/d/1osdDO0YcyBY8OdSkovt9kYgNADu85Q0F/view?usp=sharing>

III. Giải thích nguyên lí hoạt động của từng hiệu ứng:

- Hiệu ứng 1:

<pre> EFFECT1: ; cháy đơn MOV A,#00000011B LOOP0: MOV P0,A CALL DELAY RL A CJNE A,#00000011B,LOOP0 MOV A,#00000011B LOOP1: MOV P2,A </pre>	<ol style="list-style-type: none"> 1. Khởi tạo: Biến A được gán giá trị 00000011B, tương đương 11 bit đầu tiên (0 đến 10) được bật. 2. Vòng lặp LOOP0: <ul style="list-style-type: none"> Gía trị của A được gán cho chân P0, điều khiển dải đèn LED đầu tiên. Hàm DELAY để tạo độ trễ cho hiệu ứng.
---	---

<pre> CALL DELAY RL A CJNE A, #00000011B, LOOP1 MOV A, #00000011B LOOP2: MOV P3, A CALL DELAY RL A CJNE A, #00000011B, LOOP2 MOV A, #00000011B LOOP3: MOV P1, A CALL DELAY RL A CJNE A, #00000011B, LOOP3 RET </pre>	<ul style="list-style-type: none"> ○ Bit 0 của A được xoay trái (RL) ○ Kiểm tra xem A có bằng 00000011B hay không. Nếu không, lặp LOOP0 để tiếp tục xoay bit đầu tiên và điều khiển dải LED đầu tiên. ○ Khi A bằng 00000011B, tất cả các bit đầu tiên được xoay trái và dải đèn LED đầu tiên sẽ tắt. <p>3. Vòng lặp LOOP1: tương tự LOOP0 nhưng điều khiển dải đèn LED thứ hai (P2).</p> <p>4. Vòng lặp LOOP2: tương tự LOOP0 nhưng điều khiển dải đèn LED thứ ba (P3).</p> <p>5. Vòng lặp LOOP3: tương tự LOOP0, nhưng điều khiển dải đèn LED thứ tư (P0).</p>
--	---

- Hiệu ứng 2:

<pre> EFFECT2: ; sáng dần MOV A, #00H LOOP4: SETB C RLC A MOV P0, A CALL DELAY CJNE A, #0FFH, LOOP4 MOV A, #00H LOOP5: SETB C RLC A MOV P2, A CALL DELAY CJNE A, #0FFH, LOOP5 MOV A, #00H LOOP6: SETB C RLC A MOV P3, A CALL DELAY CJNE A, #0FFH, LOOP6 </pre>	<p>1. Khởi tạo: Biến A được gán giá trị 00H, tương đương với tắt cả các bit đều tắt.</p> <p>2. Vòng lặp LOOP4:</p> <ul style="list-style-type: none"> ○ Kiểm tra trạng thái của cờ Carry (C). Nếu cờ không được bật, nó sẽ được bật bằng lệnh SETB C. ○ Bit cuối cùng của A được xoay sang trái kèm theo cờ Carry (RLC). ○ Giá trị mới của A được gán cho chân P0, điều khiển dải LED đầu tiên. ○ Hàm DELAY để tạo độ trễ cho hiệu ứng. ○ Kiểm tra xem A có bằng 0FFH (tất cả các bit đều bật) hay không. Nếu không bằng, lặp lại LOOP4 để tiếp tục xoay bit cuối cùng và bật dải đèn LED đầu tiên dần dần.
---	--

<pre> MOV A,#00H LOOP7: SETB C RRC A MOV P1,A CALL DELAY CJNE A,#0FFH,LOOP7 RET </pre>	<ul style="list-style-type: none"> ○ Khi A = 0FFH, dải LED đầu tiên sẽ sáng hoàn toàn. <p>3. Vòng lặp LOOP5: tương tự LOOP4, nhưng điều khiển dải LED thứ hai (P2).</p> <p>4. Vòng lặp LOOP6: tương tự LOOP4, nhưng điều khiển dải LED thứ hai (P3).</p> <p>5. Vòng lặp LOOP7:</p> <ul style="list-style-type: none"> ○ Sử dụng lệnh xoay sang phải kèm theo cờ Carry (RRC) thay vì xoay sang trái. Điều này làm cho các bit được tắt dần bắt đầu từ bit cuối cùng. ○ Nó điều khiển dải đèn LED thứ tư (P1).
--	--

- Hiệu ứng 3:

<pre> EFFECT3: ; tắt dần MOV A,#0FFH LOOP8: CLR C RLC A MOV P0,A CALL DELAY CJNE A,#00H,LOOP8 MOV A,#0FFH LOOP9: CLR C RLC A MOV P2,A CALL DELAY CJNE A,#00H,LOOP9 MOV A,#0FFH LOOP10: CLR C RLC A MOV P3,A CALL DELAY CJNE A,#00H,LOOP10 MOV A,#0FFH LOOP11: CLR C </pre>	<p>1. Khởi tạo: Biến A được gán giá trị 0FFH, tương đương với tắt cả các bit đều bật.</p> <p>2. Vòng lặp LOOP8:</p> <ul style="list-style-type: none"> ○ Nếu cờ không được bật, nó tắt bằng lệnh CLR C. ○ Bit cuối cùng của A được xoay sang trái kèm theo cờ Carry (RLC). ○ Giá trị mới của A được gán cho chân P0, điều khiển dải LED đầu tiên. ○ Hàm DELAY để tạo độ trễ cho hiệu ứng. ○ Kiểm tra xem A có bằng 00H (tắt cả các bit đều tắt) hay không. Nếu không bằng, lặp lại LOOP8 để tiếp tục xoay bit cuối cùng và tắt dải đèn LED đầu tiên dần dần. ○ Khi A = 0FFH, dải LED đầu tiên sẽ tắt hoàn toàn. <p>3. Vòng lặp LOOP9: tương tự LOOP8, nhưng điều khiển dải LED thứ hai (P2).</p>
---	---

<pre> RRC A MOV P1,A CALL DELAY CJNE A,#00H,LOOP11 RET </pre>	<p>4. Vòng lặp LOOP10: tương tự LOOP8, nhưng điều khiển dải LED thứ hai (P3).</p> <p>5. Vòng lặp LOOP11:</p> <ul style="list-style-type: none"> ○ Sử dụng lệnh xoay sang phải kèm theo cờ Carry (RRC) thay vì xoay sang trái. Điều này làm cho các bit được tắt dần bắt đầu từ bit cuối cùng. ○ Nó điều khiển dải đèn LED thứ tư (P1)
---	---

- Hiệu ứng 4:

<pre> EFFECT4: MOV A,#00H LOOP12: SETB C RLC A MOV P0,A MOV P1,A CALL DELAY CJNE A,#0FFH,LOOP12 MOV R6,#00H MOV R7,#00H LOOP13: SETB C MOV A,R6 RRC A MOV R6,A MOV R6,A MOV P3,R6 CALL DELAY SETB C MOV A,R7 RLC A MOV R7,A MOV P2,R7 CALL DELAY CJNE R7,#0FFH,LOOP13 RET </pre>	<p>1. Khởi tạo:</p> <ul style="list-style-type: none"> ○ Biến A được gán 00H (tắt cả các bit đều tắt). ○ Biến R6, R7 được gán 00H (dùng để lưu trữ trạng thái bật/tắt của LED 3, 4) <p>2. Vòng lặp LOOP12:</p> <ul style="list-style-type: none"> ○ Nếu cờ Carry không được bật, nó được bật bằng lệnh SETB C. ○ Bit cuối cùng của A được xoay sang trái kèm theo cờ Carry (RLC). ○ Giá trị mới của A được gán cho cả chân P0 và P1, điều khiển đồng thời 2 dải đèn LED trên cùng. ○ Hàm DELAY được gọi để tạo độ trễ cho hiệu ứng. ○ Kiểm tra xem A có bằng 0FFH hay không. Nếu không bằng, lặp lại vòng lặp LOOP12 để tiếp tục xoay bit và bật sáng dần 2 dải đèn LED trên cùng. <p>3. Vòng lặp LOOP13:</p> <ul style="list-style-type: none"> ○ Nếu cờ Carry không bật, nó bật bằng lệnh SETB C. ○ Xử lý dải đèn LED thứ 3 (P3): <ul style="list-style-type: none"> ▪ Sao chép giá trị của R6 vào A.
--	---

	<ul style="list-style-type: none"> ▪ Bit cuối cùng của A được xoay sang phải kèm theo cờ Carry (RRC). ▪ Giá trị mới của A được gán cho R6 và sau đó gán luôn cho chân P3 để điều khiển dải đèn LED thứ 3. ○ Hàm DELAY được gọi để tạo độ trễ cho hiệu ứng. ○ Bật lại cờ Carry (nếu cần) bằng SETB C. ○ Xử lý dải đèn LED thứ 4 (P2): <ul style="list-style-type: none"> ▪ Sao chép giá trị của R7 vào A. ▪ Bit cuối cùng của A được xoay sang trái kèm theo cờ Carry (RLC). ▪ Giá trị mới của A được gán cho R7 và sau đó gán luôn cho chân P2 để điều khiển dải đèn LED thứ 4. ○ Hàm DELAY được gọi để tạo độ trễ cho hiệu ứng. ○ Kiểm tra xem R7 có bằng 0FFH hay không. Nếu chưa bằng, lặp lại vòng lặp LOOP13 để tiếp tục xử lý bật sáng dải đèn LED thứ 3 và 4.
--	---

- Hiệu ứng 5:

<p>EFFECT5:</p> <pre> MOV R6,#0FFH MOV R7,#0FFH </pre> <p>LOOP14:</p> <pre> CLR C MOV A,R6 RLC A MOV R6,A MOV P3,R6 CALL DELAY CLR C MOV A,R7 </pre>	<p>1. Khởi tạo: Biến R6, R7 được gán 0FFH cho hai dải đèn P2 & P3.</p> <p>2. Vòng lặp LOOP14:</p> <ul style="list-style-type: none"> ○ Nếu cờ Carry bật, nó được tắt bằng lệnh CLR C. ○ Xử lý dải đèn LED thứ 3 (P3): <ul style="list-style-type: none"> ▪ Sao chép giá trị của R6 vào A. ▪ Bit cuối cùng của A được xoay sang trái kèm theo cờ Carry (RLC). ▪ Giá trị mới của A được gán cho R6 và sau đó gán luôn cho chân P3 để điều khiển dải đèn LED thứ 3. ○ Hàm DELAY được gọi để tạo độ trễ cho hiệu ứng.
--	--

<pre> RRC A MOV R7,A MOV P2,R7 CALL DELAY CJNE R7,#00H,LOOP14 MOV A,#0FFH LOOP15: CLR C RRC A MOV P0,A MOV P1,A CALL DELAY CJNE A,#00H,LOOP15 RET </pre>	<ul style="list-style-type: none"> ○ Nếu cờ Carry bật, nó được tắt bằng lệnh CLR C. ○ Xử lý dải đèn LED thứ 4 (P2): <ul style="list-style-type: none"> ▪ Sao chép giá trị của R7 vào A. ▪ Bit cuối cùng của A được xoay sang phải kèm theo cờ Carry (RRC). ▪ Giá trị mới của A được gán cho R7 và sau đó gán luôn cho chân P2 để điều khiển dải đèn LED thứ 4. ○ Hàm DELAY được gọi để tạo độ trễ cho hiệu ứng. ○ Kiểm tra xem R7 có bằng 00H hay không. Nếu chưa bằng, lặp lại vòng lặp LOOP14 để tiếp tục xử lý tắt dần dải đèn LED thứ 3 và 4 theo thứ tự. <p>3. Vòng lặp LOOP15:</p> <ul style="list-style-type: none"> ○ Nếu cờ Carry bật, nó được tắt bằng lệnh CLR C. ○ Bit cuối cùng của A (được gán giá trị 0FFH trước vòng lặp) được xoay sang phải (RRC). ○ Giá trị mới của A được gán đồng thời cho cả chân P0 và P1, điều khiển 2 dải đèn LED trên cùng. ○ Hàm DELAY được gọi để tạo độ trễ cho hiệu ứng. ○ Kiểm tra xem A có bằng 00H hay không. Nếu chưa bằng, lặp lại vòng lặp LOOP15 để tiếp tục tắt dần 2 dải đèn LED trên cùng theo từng bit.
--	--

- Hiệu ứng 6:

<pre> EFFECT6: MOV A,#00H LOOP16: SETB C RLC A MOV P0,A </pre>	<p>1. Khởi tạo:</p> <ul style="list-style-type: none"> ○ Biến A được gán giá trị 00H (tắt cả các bit đèn tắt). ○ Biến R6 và R7 được gán giá trị 00H (dùng để lưu trữ trạng thái bật/tắt của dải đèn LED thứ 3 và 4).
--	---

<pre> MOV P1,A CALL DELAY CJNE A,#0FFH,LOOP16 MOV R6,#00H MOV R7,#00H LOOP17: SETB C MOV A,R6 RLC A MOV R6,A MOV P3,R6 CALL DELAY SETB C MOV A,R7 RRC A MOV R7,A MOV P2,R7 CALL DELAY CJNE R7,#0FFH,LOOP17 RET </pre>	<p>2. Vòng lặp LOOP16: giống hệt vòng lặp LOOP12 của hiệu ứng 4. Nó sẽ điều khiển 2 dải đèn LED trên cùng (P0, P1) bật sáng dần theo từng bit.</p> <p>3. Vòng lặp LOOP17:</p> <ul style="list-style-type: none"> ○ Giống hệt vòng lặp LOOP13 của hiệu ứng 4. Nó sử dụng các biến R6 và R7 để điều khiển 2 dải đèn LED dưới (P2 & P3) bật sáng từng bit theo thứ tự từ dưới lên (P3 trước, P2 sau). ○ Có thể thêm một đoạn code sau vòng lặp LOOP17 để giữ cho hình trái tim sáng trong một khoảng thời gian nhất định. ○ Sử dụng lại các vòng lặp LOOP14 và LOOP15 từ hiệu ứng 5 để tắt dần hình trái tim theo thứ tự từ trên xuống (P0 & P1 trước, P2 & P3 sau).
---	--

- Hiệu ứng 7:

<pre> EFFECT7: MOV R6,#00H MOV R7,#00H LOOP18: SETB C MOV A,R6 RLC A MOV R6,A MOV P1,R6 </pre>	<p>1. Khởi tạo: Biến R6, R7 được gán giá trị 00H.</p> <p>2. Vòng lặp LOOP18:</p> <ul style="list-style-type: none"> ○ Nếu cờ Carry không bật, nó bật bằng lệnh SETB C. ○ Xử lý dải đèn LED thứ 2 (P2): <ul style="list-style-type: none"> ▪ Sao chép giá trị của R6 vào A. ▪ Bit cuối cùng của A được xoay sang trái kèm theo cờ Carry (RLC).
--	--

<pre> CALL DELAY SETB C MOV A,R7 RRC A MOV R7,A MOV P2,R7 CALL DELAY CJNE R7,#0FFH,LOOP18 MOV A,#00H LOOP19: SETB C RRC A MOV P0,A MOV P3,A CALL DELAY CJNE A,#0FFH,LOOP19 MOV R6,#0FFH MOV R7,#0FFH LOOP20: CLR C MOV A,R6 RLC A MOV R6,A MOV P1,R6 CALL DELAY CLR C MOV A,R7 RRC A MOV R7,A MOV P2,R7 </pre>	<ul style="list-style-type: none"> ▪ Giá trị mới của A được gán cho R6 và sau đó gán luôn cho chân P2 để điều khiển dải đèn LED thứ 2. ○ Hàm DELAY được gọi để tạo độ trễ cho hiệu ứng. ○ Bật lại cờ Carry (nếu cần) bằng SETB C. ○ Xử lý dải đèn LED thứ 3 (P1): <ul style="list-style-type: none"> ▪ Sao chép giá trị của R7 vào A. ▪ Bit cuối cùng của A được xoay sang phải kèm theo cờ Carry (RRC). ▪ Giá trị mới của A được gán cho R7 và sau đó gán luôn cho chân P1 để điều khiển dải đèn LED thứ 3. ○ Hàm DELAY được gọi để tạo độ trễ cho hiệu ứng. ○ Kiểm tra xem R7) có bằng 0FFH hay không. Nếu chưa bằng, lặp lại vòng lặp LOOP18 để tiếp tục xử lý bật sáng dải đèn LED thứ 2 và 3 theo thứ tự từ phải sang trái. <p>3. Vòng lặp LOOP19:</p> <ul style="list-style-type: none"> ○ Nếu cờ Carry bật, nó tắt bằng lệnh CLR C. ○ Bit cuối cùng của A được xoay phải (RRC). ○ Giá trị mới A được gán đồng thời cho cả chân P0, P3 điều khiển 2 dải đèn LED ngoài cùng. ○ Hàm DELAY để tạo độ trễ cho hiệu ứng. ○ Kiểm tra xem A có bằng 00H hay không. Nếu chưa bằng, lặp lại vòng lặp LOOP19 để tiếp tục tắt dần 2 dải đèn LED ngoài cùng theo thứ tự từ trái sang phải.
--	--

<pre> CALL DELAY CJNE R7, #00H, LOOP20 MOV A, #0FFH LOOP21: CLR C RRC A MOV P0, A MOV P3, A CALL DELAY CJNE A, #00H, LOOP21 RET </pre>	
--	--

- Hiệu ứng 8:

<pre> EFFECT8: MOV A, #00H LOOP22: SETB C RLC A MOV P0, A MOV P3, A CALL DELAY CJNE A, #0FFH, LOOP22 MOV R6, #00H MOV R7, #00H LOOP23: SETB C MOV A, R6 RLC A MOV R6, A MOV P2, R6 CALL DELAY </pre>	<ol style="list-style-type: none"> 1. Khởi tạo: Biến A được gán giá trị 00H. 2. Vòng lặp LOOP22: Giống hệt vòng lặp LOOP19 của hiệu ứng 7. Nó sẽ điều khiển 2 dải LED ngoài cùng (P0, P3) bật sáng dần theo thứ tự từ trái sang phải, 3. Vòng lặp LOOP23: <ul style="list-style-type: none"> ○ Biến R6, R7 được gán giá trị 00H. ○ Giống hệt vòng lặp LOOP18 của hiệu ứng 7 nhưng sử dụng SETB C thay vì CLR C trước khi xoay bit. Điều này sẽ làm cho các bit được bật sáng theo thứ tự từ dưới lên trên (P2 trước, P1 sau). 4. Vòng lặp LOOP24: <ul style="list-style-type: none"> ○ Biến A được gán giá trị 0FFH. ○ Giống hệt vòng lặp LOOP19 của hiệu ứng 7 nhưng sử dụng lệnh RLC thay vì RRC. Điều này sẽ tắt dần các bit theo thứ tự từ phải sang trái. 5. Vòng lặp LOOP25: <ul style="list-style-type: none"> ○ Biến R6, R7 được gán giá trị 0FFH.
--	--

<pre> SETB C MOV A,R7 RRC A MOV R7,A MOV P1,R7 CALL DELAY CJNE R7,#0FFH,LOOP23 MOV A,#0FFH LOOP24: CLR C RLC A MOV P0,A MOV P3,A CALL DELAY CJNE A,#00H,LOOP24 MOV R6,#0FFH MOV R7,#0FFH LOOP25: CLR C MOV A,R6 RLC A MOV R6,A MOV P2,R6 CALL DELAY CLR C MOV A,R7 RRC A MOV R7,A MOV P1,R7 CALL DELAY </pre>	<ul style="list-style-type: none"> ○ Sử dụng CLR C thay vì SETB C trước khi xoay bit. Điều này sẽ tắt dần các bit theo thứ tự từ dưới lên trên (P2 trước, P1 sau).
---	---

CJNE R7,#00H,LOOP25 RET	
--------------------------------	--

- Hiệu ứng 9:

EFFECT9: ; nhấp nháy MOV R7,#5 TIMES: MOV P0,#00H MOV P1,#00H MOV P2,#00H MOV P3,#00H CALL DELAY MOV P0,#0FFH MOV P1,#0FFH MOV P2,#0FFH MOV P3,#0FFH CALL DELAY DJNZ R7,TIMES RET	<ul style="list-style-type: none"> ○ Biến R7 được gán giá trị 5 (số lần nhấp nháy). ○ Tắt tất cả dải đèn LED: <ul style="list-style-type: none"> ▪ Gán giá trị 00H cho các chân P0, P1, P2, và P3. ▪ Gọi hàm DELAY để tạo độ trễ cho hiệu ứng tắt. ○ Bật tất cả các dải đèn LED. <ul style="list-style-type: none"> ▪ Gán giá trị 0FFH cho các chân P0, P1, P2, và P3. ▪ Gọi hàm DELAY để tạo độ trễ cho hiệu ứng bật. ○ Giảm giá trị của biến R7 đi 1. ○ Kiểm tra xem R7 có bằng 0 không. Nếu có, thoát khỏi vòng lặp TIMES. ○ Nếu chưa, lặp lại vòng lặp TIMES để tiếp tục nhấp nháy các dải đèn LED.
---	---

- Hiệu ứng 10:

EFFECT10: MOV A,#00H LOOP26: SETB C RLC A MOV P0,A MOV P1,A CALL DELAY CJNE A,#0FFH,LOOP26 MOV R6,#00H MOV R7,#00H	<ol style="list-style-type: none"> 1. Khởi tạo: <ul style="list-style-type: none"> ○ Biến A được gán giá trị 00H. ○ Biến R6, R7 được gán giá trị 00H. 2. Vòng lặp LOOP26: Giống vòng lặp LOOP12 của hiệu ứng 4, nó sẽ điều khiển 2 dải đèn LED trên cùng (P0, P1) bật sáng dần theo từng bit. 3. Vòng lặp LOOP27: Giống vòng lặp LOOP13 của hiệu ứng 4, nó sử dụng các biến R6 và R7 để điều khiển 2 dải đèn LED dưới (P2, P3) bật sáng từng bit theo thứ tự từ dưới lên (P3 trước P2).
--	--

<pre> LOOP27: SETB C MOV A,R6 RRC A MOV R6,A MOV P3,R6 CALL DELAY SETB C MOV A,R7 RLC A MOV R7,A MOV P2,R7 CALL DELAY CJNE R7,#0FFH,LOOP27 MOV R6,#00H MOV R7,#00H LOOP28: SETB C MOV A,R6 RLC A MOV R6,A MOV P3,R6 CALL DELAY SETB C MOV A,R7 RRC A MOV R7,A MOV P2,R7 CALL DELAY </pre>	<p>4. Vòng lặp LOOP28:</p> <ul style="list-style-type: none"> ○ Có thêm một đoạn code sau vòng lặp LOOP27 để giữ cho hình trái tim sáng trong một khoảng thời gian nhất định. ○ Lặp lại giống LOOP27, điều này tạo ra hiệu ứng nhấp nháy nhanh hình trái tim sáng từ dưới lên. <p>5. Vòng lặp LOOP29: Giống vòng lặp LOOP14 của hiệu ứng 5. Nó sẽ tắt dần 2 dải đèn LED trên cùng (P0 & P1) theo từng bit.</p>
--	--

<pre> CJNE R7,#0FFH,LOOP28 MOV A,#00H LOOP29: SETB C RLC A MOV P0,A MOV P1,A CALL DELAY CJNE A,#0FFH,LOOP29 RET </pre>	
--	--

III. Tìm hiểu và viết các bước thực hiện mạch in từ bước thiết kế trên proteus đến bước ra được mạch in trong thực tế.

1. Chuẩn bị:

- Phần mềm: Proteus 8 Professional (phiên bản 8.6 trở lên)
- Phần cứng: Máy tính, máy in laser, máy in ỉ (bàn ỉ nhiệt), giấy in nhiệt, đồng clad (đồng ép phíp), dung dịch ăn mòn, dung dịch trung hòa, bể chứa dung dịch, găng tay cao su, kìm, dao cắt,...

2. Thiết kế mạch in trên Proteus:

- Mở Proteus và tạo một project mới.
- Sử dụng các công cụ vẽ trong Proteus để vẽ sơ đồ nguyên lý và layout mạch in.
- Kiểm tra kỹ lưỡng sơ đồ nguyên lý và layout để đảm bảo không có lỗi.
- Chạy DRC (Design Rule Check) để phát hiện các vi phạm quy tắc thiết kế.
- Lưu file thiết kế ở định dạng *.sch và *.brd.

3. In ấn layout mạch in:

- Mở file layout mạch in (*.brd) trong Proteus.

- Cài đặt các thông số in ấn như kích thước giấy, hướng in, tỉ lệ in,...
- In layout mạch in lên giấy in nhiệt bằng máy in laser.
- Lưu ý: Sử dụng giấy in nhiệt chuyên dụng cho máy in ulti.

4. Chuyển mạch in lên đồng clad:

- Cắt bỏ phần giấy thừa xung quanh bản in.
- Đặt bản in nhiệt lên đồng clad, mặt có hình ảnh úp xuống.
- Cố định bản in bằng băng dính chịu nhiệt.
- Sử dụng máy in ulti để ulti bản in lên đồng clad.
- Cài đặt nhiệt độ và thời gian ulti phù hợp với loại đồng clad và giấy in nhiệt.
- Miết nhẹ bằng tay hoặc dụng cụ chuyên dụng trong quá trình ulti để đảm bảo bản in bám chặt vào đồng clad.

5. Ăn mòn đồng:

- Đeo găng tay cao su và kính bảo hộ.
- Chuẩn bị dung dịch ăn mòn theo hướng dẫn của nhà sản xuất.
- Cho đồng clad đã ulti vào bể chứa dung dịch ăn mòn.
- Thời gian ăn mòn phụ thuộc vào độ dày của đồng clad và loại dung dịch ăn mòn.
- Quan sát quá trình ăn mòn và khuấy đều dung dịch để đảm bảo ăn mòn đồng đều.

6. Trung hòa dung dịch ăn mòn:

- Cho đồng clad đã ăn mòn vào dung dịch trung hòa để loại bỏ axit dư thừa.
- Rửa sạch đồng clad bằng nước và lau khô.

7. Kiểm tra mạch in:

- Sử dụng đồng hồ đo điện để kiểm tra các đường mạch và pads trên mạch in.

- Đảm bảo các đường mạch không bị hở mạch hoặc chạm mạch.
- Các pads phải có đủ diện tích tiếp xúc để hàn linh kiện.

8. Lắp ráp linh kiện:

- Hàn các linh kiện điện tử lên pads trên mạch in theo đúng sơ đồ nguyên lý.
- Sử dụng mỏ hàn và chì hàn phù hợp.
- Hàn các linh kiện cẩn thận để tránh làm hỏng mạch in.

9. Kiểm tra và hoàn thiện:

- Kiểm tra lại hoạt động của mạch in bằng đồng hồ đo điện hoặc dụng cụ chuyên dụng.
- Sửa chữa nếu có bất kỳ lỗi nào.
- Vệ sinh sạch sẽ mạch in.
- Lắp đặt mạch in vào vỏ hoặc khung theo thiết kế.