

PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS

Instructor: The Tung Than

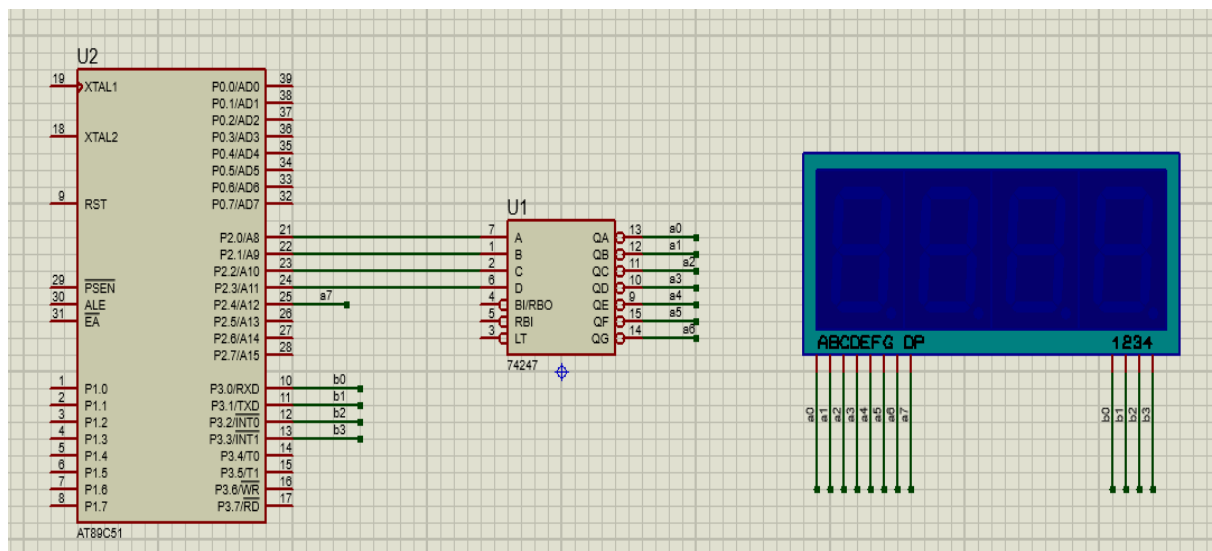
Student's name: Trần Lê Minh Đăng

Student code: 21520684

PRACTICE REPORT NO 2

COMMUNICATION WITH 7-SEGMENT LED AND TIMER

I. Present and draw a flowchart of the LED scanning algorithm applied to display 7-segment led.



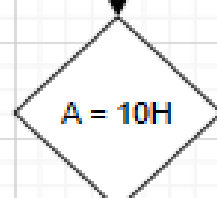
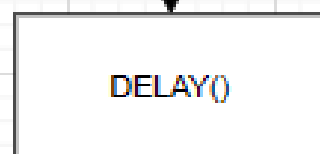
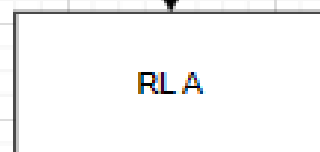
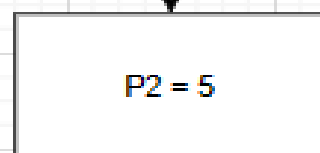
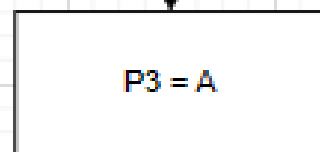
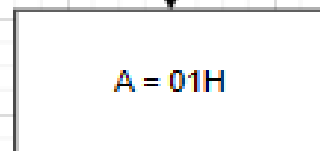
- Giải thuật quét LED từ trái sang phải
- + B1: Công tắc 1 bật, các công tắc khác tắt: $P3.0 = 1, P3.1 = P3.2 = P3.3 = 0$. Tức là $P3 = 01H$. Chuyển số cần hiển thị vào P2
- + B2: Công tắc 2 bật, các công tắc khác tắt: $P3.1 = 1, P3.0 = P3.2 = P3.3 = 0$. Tức là $P3 = 02H$. Chuyển số cần hiển thị vào P2
- + B3: Công tắc 3 bật, các công tắc khác tắt: $P3.2 = 1, P3.0 = P3.1 = P3.3 = 0$. Tức là $P3 = 04H$. Chuyển số cần hiển thị vào P2
- + B4: Công tắc 1 bật, các công tắc khác tắt: $P3.3 = 1, P3.0 = P3.1 = P3.2 = 0$. Tức là $P3 = 08H$. Chuyển số cần hiển thị vào P2

Chúng ta có thể gán $A = 01H$. Rồi gán $P3 = A$. Sau đó thực hiện lệnh quay trái A đến khi $A = 10H$ thì kết thúc.

- Sơ đồ thuật toán



BẮT ĐẦU



S

Đ



- Code minh họa

```
ORG 0H
MAIN:
    CALL HIENTHI
    JMP MAIN
HIENTHI:
    MOV A,#01H
HT:
    MOV P3,A
    MOV P2,#5
    RL A
    CALL DELAY5MS
    CJNE A,#10H,HT
    RET
DELAY5MS:
    MOV TMOD,#01H
    MOV TH0,#HIGH(-5000)
    MOV TL0,#LOW(-5000)
    SETB TR0
    JNB TF0,$
    CLR TR0
    CLR TF0
    RET
END
```

II. Using the 8051 microcontroller's Timer, design a clock circuit with 24h format with the initial time set in the source code.

-Link:

https://drive.google.com/drive/folders/1ahfHLvv4_2LD8m3M5XengMgsVvsRp9Ae?usp=sharing

- Ý tưởng thuật toán

+ Sử dụng thanh ghi R1, R2, R3 để lưu giá trị của giờ, phút, giây

+ Giây tăng thêm 1, nếu bằng 60 thì phút cộng 1 và giây reset về 0. Nếu giây không bằng 60 thì tiếp tục tăng giây thêm 1, giờ, phút giữ nguyên.

+ Trường hợp phút tăng thêm 1, nếu bằng 60 thì giờ cộng 1 và phút reset về 0. Nếu phút không bằng 60 thì tiếp tục quay lại tăng giây, giờ giữ nguyên.

+ Trường hợp giờ tăng thêm 1, nếu bằng 24 thì giờ reset về lại 0

- Giải thích hàm HIEN THI:

+ Tách riêng 2 chữ số của Giờ

+ Gán $A = R1 = \text{Giờ}$

+ Gán $B = 10$

+ DIV AB: Lấy A chia B, A chứa thương của A và B, B chứa số dư khi chia A cho B. Khi đó A là chữ số hàng chục, B là chữ số hàng đơn vị của Giờ

+ Bật công tắc 1, $P2 = A$

+ Bật công tắc 2, $P2 = B$

+ Vì ta chỉ sử dụng dấu chấm ở LED thứ 2 nên khi công tắc 2 bật, ta cho dấu chấm hiển thị, tức là $P2.4 = 0$

+ Tách riêng 2 chữ số của Phút : Tương tự của Giờ

+ Hàm hiển thị chứa 4 câu lệnh CALL DELAY5MS nên thời gian delay của hàm là 20ms. Ta muốn thời gian delay là 1s nên tiến hành lặp lại 50 lần.

- Giải thích hàm delay5ms:

+ Đầu ta chọn Timer 0 chế độ 1 bằng câu lệnh MOV TMOD, #01H

+ Tiếp đến gán giá trị cần nạp vào thanh ghi TH0, TL0. Ở đây ta sử dụng tần số thạch anh là 12MHz nên ta có công thức tính thời gian delay:

Tần số bộ định thời: $12\text{MHz}/12 = 1\text{MHz}$

Thời gian 1 chu kỳ bộ định thời: $1/1\text{MHz} = 10^{-6}\text{s}$

Muốn tạo delay 5ms thì số chu kỳ trễ: $5 \times 10^{-3}\text{s} / 10^{-6}\text{s} = 5000$

Khi đó ta nạp giá trị -5000 để timer đếm từ 65536 – 5000 đến 65536

+ Bật bộ định thời bằng lệnh SETB TR0

+ Lệnh JNB TF0, \$ có tác dụng nhảy tại chỗ cho đến khi TF0 bật lên 1

- Source code:

```
ORG 0H
    MOV R1,#16;GIO
    MOV R2,#01;PHUT
    MOV R3,#55;GIAY
MAIN:
    CALL SET_CLOCK
    JMP MAIN
SET_CLOCK:
    CALL HIENTHI
    INC R3
    CJNE R3,#60,SET_CLOCK
    INC R2
    MOV R3,#0
    CJNE R2,#60,SET_CLOCK
    INC R1
    MOV R2,#0
    CJNE R1,#24,SET_CLOCK
    MOV R1,#0
    RET
HIENTHI:
    MOV R4,#50
HIEN_THI:
    MOV A,R1
    MOV B,#10
    DIV AB
```

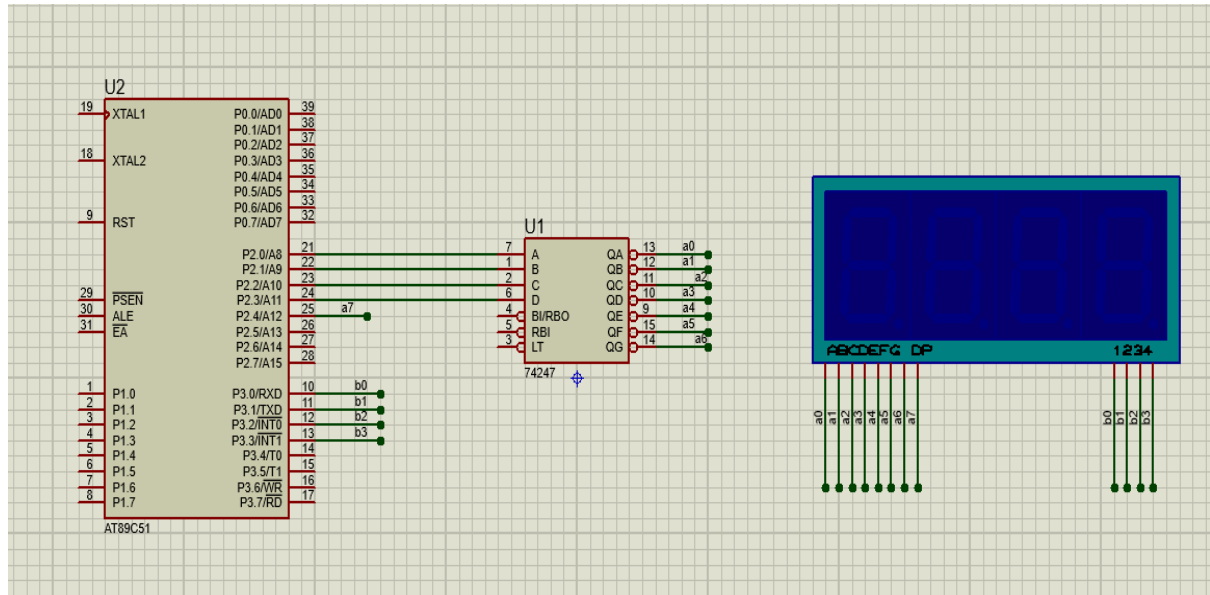
```
MOV P3,#01H
MOV P2,A
SETB P2.4
CALL DELAY5MS
MOV P3,#02H
MOV P2,B
CLR P2.4
CALL DELAY5MS
MOV A,R2
MOV B,#10
DIV AB
MOV P3,#04H
MOV P2,A
SETB P2.4
CALL DELAY5MS
MOV P3,#08H
MOV P2,B
SETB P2.4
CALL DELAY5MS
DJNZ R4,HIEN_THI
RET
```

DELAY5MS:

```
MOV TMOD,#01H
MOV TH0,#HIGH(-5000)
MOV TL0,#LOW(-5000)
SETB TR0
JNB TF0,$
CLR TR0
CLR TF0
RET
```

END

- Sơ đồ mạch trên Proteus



III. Exercise

With the above clock design, use a loop to create a delay instead of Timer. State the advantages and disadvantages of the two methods.

- Ý tưởng thuật toán: Tương tự như thuật toán bài trên
- Hàm delay 5ms viết bằng vòng lặp

DELAY5MS_0:

MOV R6,#48

DELAY5MS_1:

MOV R5,#50

DJNZ R5,\$

DELAY5MS_2:

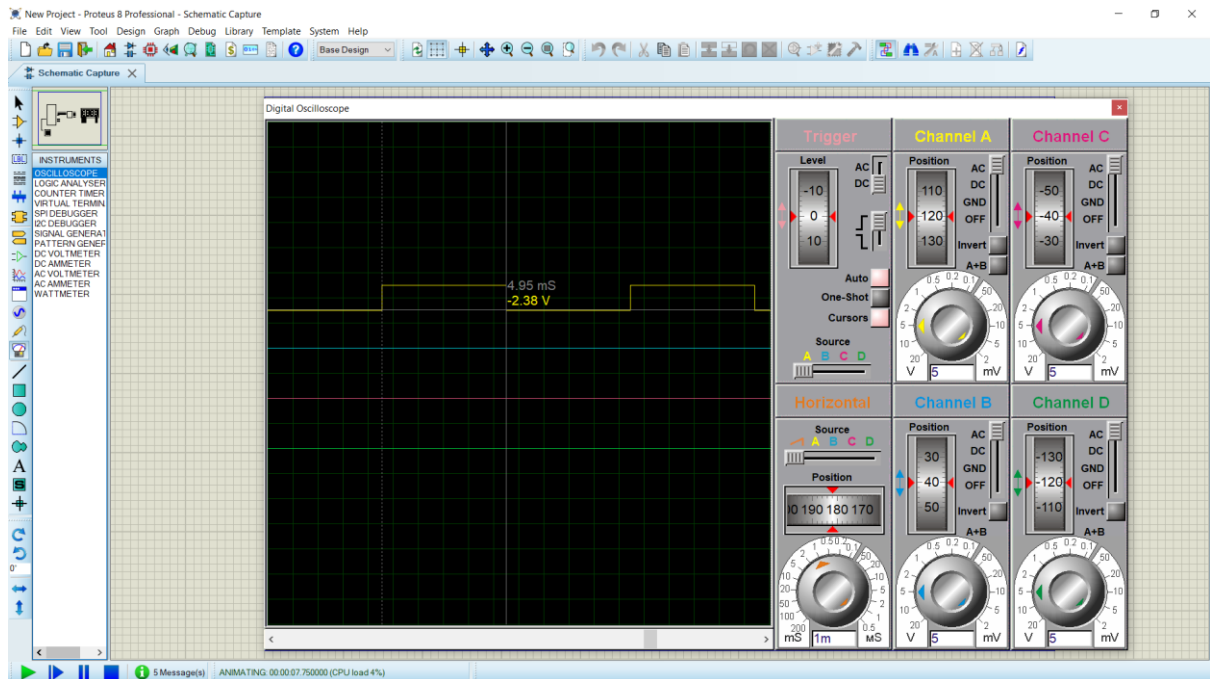
DJNZ R6,DELAY5MS_1

RET

Thời gian delay:

$$(2 \times 2 + 2 \times 50 + 2 \times 48 \times 50) \times 10^{-6} = 4.9 \text{ms}$$

Mô phỏng



- Source code

```
ORG 0H

    MOV R1,#16;GIO
    MOV R2,#01;PHUT
    MOV R3,#55;GIAY
MAIN:
    CALL SET_CLOCK
    JMP MAIN
SET_CLOCK:
    CALL HIENTHI
    INC R3
    CJNE R3,#60,SET_CLOCK
    INC R2
    MOV R3,#0
    CJNE R2,#60,SET_CLOCK
    INC R1
    MOV R2,#0
    CJNE R1,#24,SET_CLOCK
```


MOV R1,#0

RET

HIENTHI:

MOV R4,#50

HIEN_THI:

MOV A,R1

MOV B,#10

DIV AB

MOV P3,#01H

MOV P2,A

SETB P2.4

CALL DELAY5MS_0

MOV P3,#02H

MOV P2,B

CLR P2.4

CALL DELAY5MS_0

MOV A,R2

MOV B,#10

DIV AB

MOV P3,#04H

MOV P2,A

SETB P2.4

CALL DELAY5MS_0

MOV P3,#08H

MOV P2,B

SETB P2.4

CALL DELAY5MS_0

DJNZ R4,HIEN_THI

RET

DELAY5MS_0:

MOV R6,#48

```
DELAY5MS_1:
```

```
    MOV R5,#50
```

```
    DJNZ R5,$
```

```
DELAY5MS_2:
```

```
    DJNZ R6,DELAY5MS_1
```

```
    RET
```

```
END
```

- So sánh 2 phương pháp

+ Timer: Độ chính xác cao, tính toán dễ hơn, viết code phức tạp hơn

+ Vòng lặp: Độ chính xác thấp hơn, tính toán khó hơn, viết code dễ hơn