

PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS

Instructor: The Tung Than

Student's name: Nguyen Quoc Truong An

Student code: 21521810

PRACTICE REPORT NO 3

LAB3: USING INTERRUPT

I. Student preparation

- Knowledge of how to install and use Interrupt.

II. Practice content

- 1) Present and draw a flowchart to handle 2 buttons with the following functions:

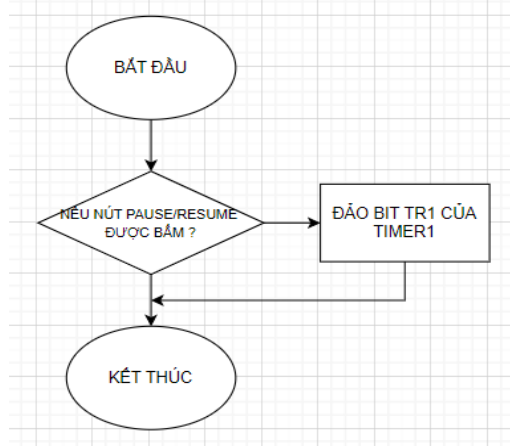
- Button A: Pause/Resume stopwatch
- Button B: Reset the stopwatch.

- Với nút A, ta nối với ngắt ngoài 0 (INT0) sử dụng ngắt ở chế độ ngắt cạnh. Ta thực hiện tính toán thời gian (giây, phần trăm giây) trong hàm ngắt của Timer1 khi giá trị Timer1 tràn và thực hiện quét led để hiển thị sử dụng hàm delay bằng vòng lặp.

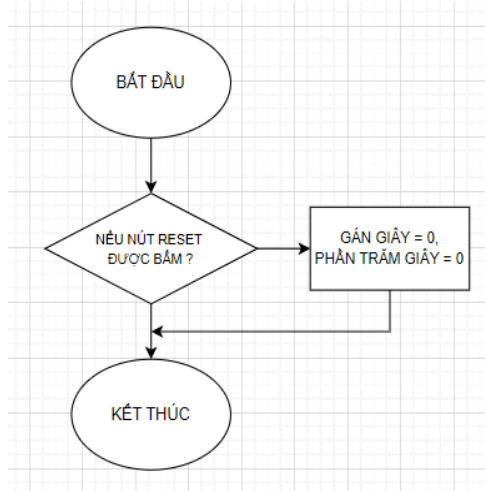
Khi nút A được bấm, tức ngắt ngoài 0 (INT0) được hoạt động. Lúc này hàm xử lý ngắt ISR_INT0 sẽ được tự động gọi. Trong hàm này ta thực hiện đảo bit TR1 để bật tắt Timer1 tương ứng với khi Resume và Pause.

- Với nút B, ta nối với ngắt ngoài 1 (INT1) sử dụng ngắt ở chế độ ngắt cạnh. Khi nút B được bấm, hàm xử lý ngắt ISR_INT1 được tự động gọi. Trong hàm này ta thực hiện nạp lại giá trị của 2 thanh ghi lưu giây và lưu phần trăm giây $R1 = 0$, $R0 = 0$ ứng với chức năng reset.

- Lưu đồ thuật toán xử lý khi bấm nút A:

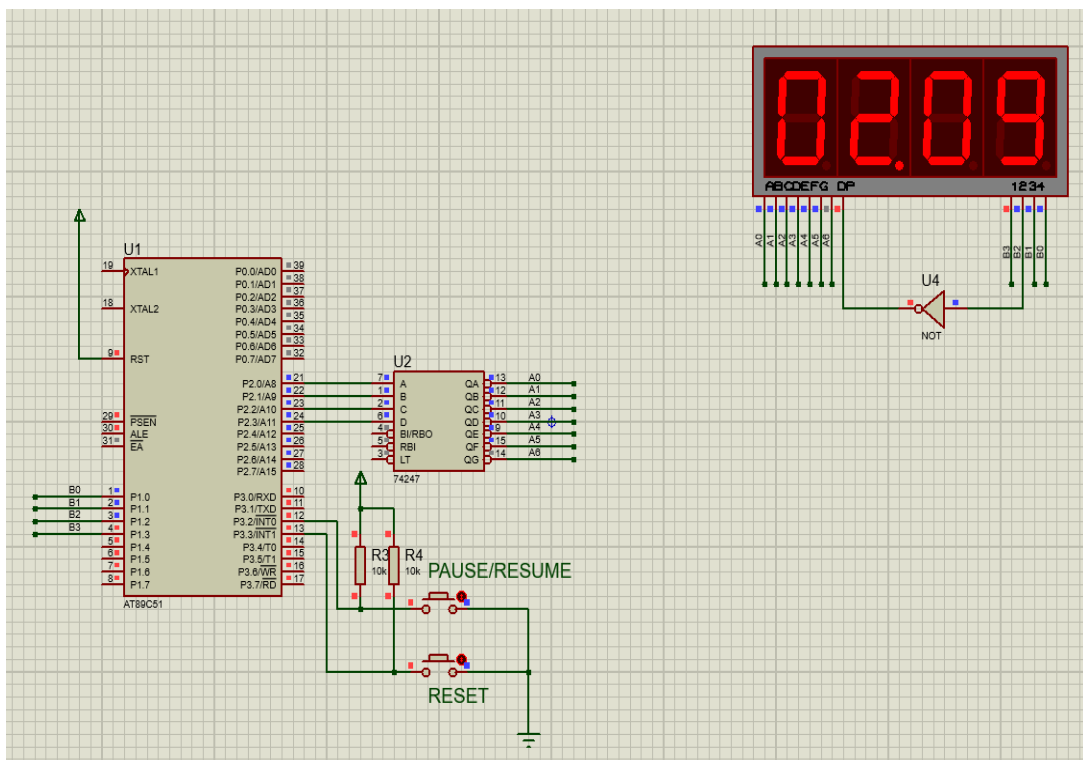


- Lưu đồ thuật toán xử lý khi bấm nút B:



2) Using AT89C51/AT89C52 in combination with 4 7-Segment LED modules and 2 buttons above, design a Sport clock circuit with the ability to count accurately to 1% of seconds, counting range from 00.00 seconds to 99.99 seconds and has 2 buttons to control Pause/Resume and Reset.

-Thiết kế:



-Code:

```
$NOMOD51
$INCLUDE (8051.MCU)

; RESET VECTOR
ORG 0000H
JMP START

; VECTOR NGAT TIMER1
ORG 0001BH
LJMP ISR_TIMER1

; VECTOR NGAT INT0
ORG 0003H
LJMP ISR_INT0

; VECTOR NGAT INT1
ORG 0013H
LJMP ISR_INT1

ORG 0100H
START:
    MOV TMOD, #10H ; CHON TIME1 CHE DO 1
    MOV TL1, #0F0H ; NAP GIA TRI CHO TIMER1
    MOV TH1, #0D8H
    SETB TR1 ; BAT TIMER1

    MOV IE, #10001101B ; CHO PHEP NGAT TIMER1, NGAT INT0, INT1
    SETB TCON.0 ; CHO PHEP NGAT THEO CANH INT0
    SETB TCON.2 ; CHO PHEP NGAT THEO CANH INT1

    MOV R0, #0 ; LUU 1%S
    MOV R1, #0 ; LUU GIAY

LOOP:
    CALL DISPLAY_LOOP

JMP LOOP

DISPLAY_LOOP: ; HAM HIEN THI QUET LED
    SETB P1.0 58 MOV A, R0
    MOV B, #10
    DIV AB
    MOV P2, B
    CALL DELAY_5MS
    CLR P1.0

    SETB P1.1

    MOV A, R0

    MOV B, #10
    DIV AB
    MOV P2, A
    CALL DELAY_5MS
    CLR P1.1
```

```
SETB P1.2 74 MOV A, R1
MOV B, #10
DIV AB
MOV P2, B
CALL DELAY_5MS
CLR P1.2
```

```
SETB P1.3 82 MOV A, R1
MOV B, #10
DIV AB
MOV P2, A
CALL DELAY_5MS
CLR P1.3
```

```
RET
```

```
DELAY_10MS:      ; HAM DELAY 10ms BANG TIMER1, DUNG DE TANG GIA TRI PHAN TRAM GIAY
MOV TL1, #0F0H 93 MOV TH1, #0D8H
SETB TR1
JNB TF1, $
```

```
RET
```

```
DELAY_5MS:      ; HAM DELAY 5ms BANG VONG LAP, DUNG DE QUET LED
MOV R4, #10
DELAY:
MOV R5, #250
DJNZ R5, $
DJNZ R4, DELAY
```

```
RET
```

```
ISR_TIMER1:      ; HAM XU LY BO DEM THOI GIAN KHI TIMER1 TRAN
                  ; XU LY TANG DON VI THOI GIAN
```

```
INC R0
MOV A, R0
MOV B, #100
CJNE A, B, NOT_EQUAL
MOV R0, #0
INC R1
```

```
MOV A, R1
MOV B, #100
CJNE A, B, NOT_EQUAL
MOV R1, #0
```

```
NOT_EQUAL:      ; NAP LAI GIA TRI CHO TIMER1
MOV TL1, #0F0H 123 MOV TH1, #0D8H
RETI
```

```
ISR_INT0:      ; HAM XU LY NUT A
CPL TR1      ; TAT TIMER1
RETI
```

```
ISR_INT1:      ; HAM XU LY NUT B
MOV R0, #0      ; GAN PHAN TRAM GIAY = 0
MOV R1, #0      ; GAN GIAY = 0
```

```
RETI
```

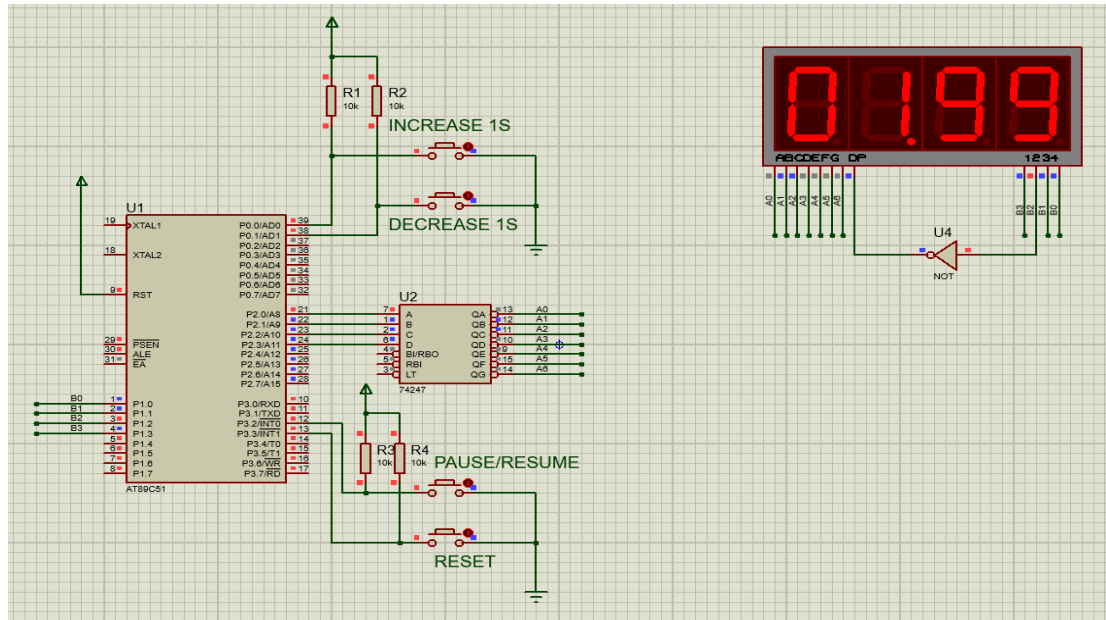
```
END
```

III. Exercises

Add 2 buttons to the Sport watch with the following function:

- Button C: Increase the number of seconds counting to 1 second
- Button D: Decrease the number of seconds to 1 second

-Thiết kế:



-Code: Ta chỉ cần xử lý thêm một đoạn code ở vòng lặp Loop

```

LOOP:
JNB P0.0, INCREASE          ; NEU NUT INCREASE ĐƯỢC BẮM THÌ NHAY TOI HAM INCREASE
JNB P0.1, DECREASE          ; NEU NUT DECREASE ĐƯỢC BẮM THÌ NHAY TOI HAM DECREASE
JMP NO_PRESS                ; KHONG NUT NAO BAM

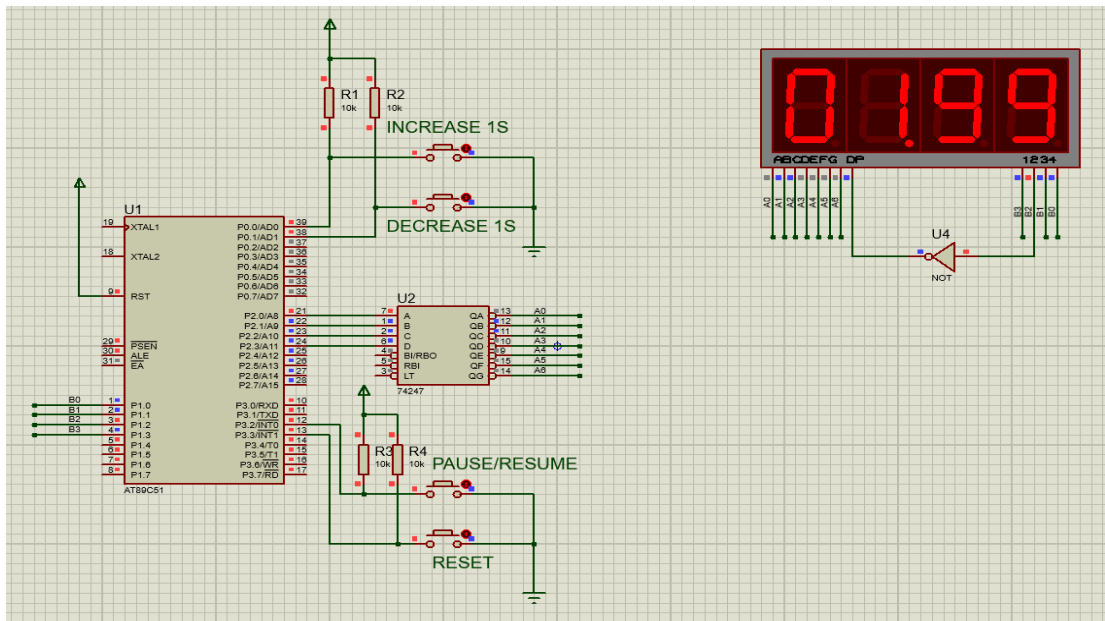
INCREASE:                    ; HAM XU LY TANG 1s
CALL DISPLAY_LOOP
CLR TR1                      ; TAT TIMER1
JNB P0.0, INCREASE          ; NHAY TAI CHO DE KIEM TRA CO GIU NUT NHAN
INC R1                       ; TANG 1 GIAY
SETB TR1                     ; BAT TIMER1
JMP NO_PRESS                ; NHAY TOI NHAN NO_PRESS
DECREASE:                    ; HAM XU LY GIAM 1s
CALL DISPLAY_LOOP
CLR TR1                      ; TAT TIMER1
JNB P0.1, DECREASE          ; NHAY TAI CHO DE KIEM TRA CO GIU NUT NHAN
DEC R1                       ; GIAM 1 GIAY
SETB TR1                     ; BAT TIMER1

NO_PRESS:
CALL DISPLAY_LOOP            ; GOI HAM QUET LED
JMP LOOP

```

IV. Report

- 1) Design result (screenshot and pasted in the report). (1 point)

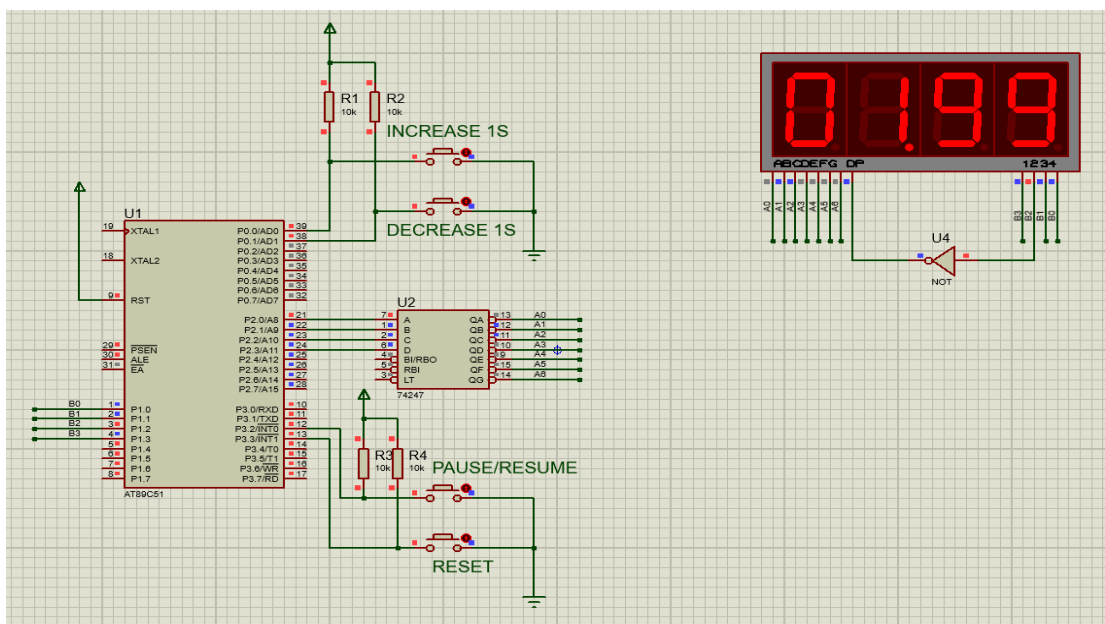


- 2) Explain the operating principle of the effects, accompanied by a video (send a Google Drive link) to demonstrate the circuit operation in case the instructor cannot run the design file. (3 points)

* Google Drive link:

<https://drive.google.com/drive/folders/1AAAtaz9caai1kAWlxRj6GhLg12cMjXCMZ?usp=sharing>

**Nguyên tắc hoạt động của mạch trên proteus



- Ta sử dụng điện trở kéo lên ở các nút bấm tại pin P0.0, P0.1, P3.2, P3.3. Khi không bấm, các chân này ở mức cao, khi các nút được bấm chân ở mức thấp.
- Pin P0.0 và P0.1 nối với các nút bấm tăng 1s, giảm 1s, được xử lý bằng phần mềm.
- Pin P3.2, P3.3 nối với các nút bấm Pause/Resume, Reset, được xử lý bằng ngắt ngoài 0 và ngắt ngoài 1.
- Sử dụng IC giải mã 74247 led-7-segments để hỗ trợ hiển thị led 7 đoạn.
- Các pin P1.0 -> P1.3 dùng để cấp cực dương cho các led.

*****Chương trình đầy đủ hỗ trợ 4 nút bấm: PAUSE/RESUME, RESET, INCREASE, DECREASE**

```

$NOMOD51
$INCLUDE (8051.MCU)
; RESET VECTOR
ORG 0000H
JMP START

; VECTOR NGAT TIMER1
ORG 001BH
LJMP ISR_TIMER1

; VECTOR NGAT INTO
ORG 0003H
LJMP ISR_INT0

; VECTOR NGAT INT1
ORG 0013H
LJMP ISR_INT1

```

ORG 0100H

START:

```
MOV TMOD, #10H           ; CHON TIME1 CHE DO 1
MOV TL1, #0F0H           ; NAP GIA TRI CHO TIMER1
MOV TH1, #0D8H
SETB TR1                 ; BAT TIMER1

MOV IE, #10001101B       ; CHO PHEP NGAT TIMER1, NGAT INTO, INT1
SETB TCON.0              ; CHO PHEP NGAT THEO CANH INTO
SETB TCON.2              ; CHO PHEP NGAT THEO CANH INT1

MOV R0, #0               ; LUU 1%S
MOV R1, #0               ; LUU GIAY
```

LOOP:

```
JNB P0.0, INCREASE
JNB P0.1, DECREASE
JMP NO_PRESS
```

INCREASE:

```
CLR TR1
JNB P0.0, $
INC R1
SETB TR1
JMP NO_PRESS
```

DECREASE:

```
CLR TR1
JNB P0.1, $
DEC R1
SETB TR1
```

NO_PRESS:

```
ALL DISPLAY_LOOP
```

JMP LOOP

DISPLAY_LOOP: ; HAM HIEN THI QUET LED

```
SETB P1.0
MOV A, R0
MOV B, #10
DIV AB
MOV P2, B
CALL DELAY_5MS
CLR P1.0
```

```
SETB P1.1
MOV A, R0
MOV B, #10
DIV AB
MOV P2, A
CALL DELAY_5MS
CLR P1.1
```

```
SETB P1.2
MOV A, R1
MOV B, #10
DIV AB
MOV P2, B
CALL DELAY_5MS
CLR P1.2
```



```
SETB P1.382 MOV A, R1
MOV B, #10
DIV AB
MOV P2, A
CALL DELAY_5MS
CLR P1.3
```

```
RET
```

```
DELAY_10MS: ; HAM DELAY 10ms BANG TIMER1, DUNG DE TANG GIA TRI PHAN TRAM GIAY
```

```
MOV TL1, #0F0H
MOV TH1, #0D8H
SETB TR1
JNB TF1, $
```

```
RET
```

```
DELAY_5MS: ; HAM DELAY 5ms BANG VONG LAP, DUNG DE QUET LED
```

```
MOV R4, #10
DELAY:
MOV R5, #250
DJNZ R5, $
DJNZ R4, DELAY
```

```
RET
```

```
ISR_TIMER1: ; HAM XU LY BO DEM THOI GIAN KHI TIMER1 TRAN
; XU LY TANG DON VI THOI GIAN
; TANG PHAN TRAM GIAY
```

```
INC R0
MOV A, R0
MOV B, #100
CJNE A, B, NOT_EQUAL ; KIEM TRA NEU PHAN TRAM GIAY = 100 THI RESET PHAN TRAM GIAY VE 0
MOV R0, #0
INC R1 ; TANG GIAY
```

```
MOV A, R1
MOV B, #100
CJNE A, B, NOT_EQUAL ; KIEM TRA NEU GIAY = 100 THI RESET GIAY VE 0
MOV R1, #0
```

```
NOT_EQUAL: ; NAP LAI GIA TRI CHO TIMER1
MOV TL1, #0F0H
MOV TH1, #0D8H
```

```
RETI
```

```
ISR_INT0: ; HAM XU LY NUT A
CPL TR1 ; TAT TIMER1
```

```
RETI
```

```
ISR_INT1: ; HAM XU LY NUT B
MOV R0, #0 ; GAN PHAN TRAM GIAY = 0
MOV R1, #0 ; GAN GIAY = 0
```

```
RETI
```

```
END
```

***Giải thích chi tiết code:

```
; VECTOR NGAT TIMER1
ORG 001BH
LJMP ISR_TIMER1

; VECTOR NGAT INT0
ORG 0003H
LJMP ISR_INT0

; VECTOR NGAT INT1
ORG 0013H
LJMP ISR_INT1
```

-Khai báo vector ngắt của Timer1 tại địa chỉ 1BH

-Khai báo vector ngắt ngoài 0, ngắt ngoài 1 lần lượt tại địa chỉ 03H và 13H

```
START:
MOV TMOD, #10H ; CHON TIME1 CHE DO 1
MOV TL1, #0F0H ; NAP GIA TRI CHO TIMER1
MOV TH1, #0D8H
SETB TR1 ; BAT TIMER1

MOV IE, #10001101B ; CHO PHEP NGAT TIMER1, NGAT INT0, INT1
SETB TCON.0 ; CHO PHEP NGAT THEO CANH INT0
SETB TCON.2 ; CHO PHEP NGAT THEO CANH INT1

MOV R0, #0 ; LUU 1%S
MOV R1, #0 ; LUU GIAY
```

-Để tạo hàm delay 10ms thì cần nạp giá trị cho Timer1 là D8F0H và bật Timer1 bằng bit TR1

-Set thanh ghi IE với giá trị 10001101B, ứng với bật cho phép ngắt, cho phép ngắt Timer1, cho phép ngắt ngoài INT1, INT0

-Set bit TCON.0, TCON.2 để chọn chế độ ngắt theo cạnh cho ngắt ngoài 0 và 1

-Khởi tạo giá trị R0 và R1 lần lượt lưu giây và phần trăm giây

```

LOOP:
JNB P0.0, INCREASE           ; NEU NUT INCREASE ĐƯỢC BẮM THÌ NHAY TOI HAM INCREASE
JNB P0.1, DECREASE          ; NEU NUT DECREASE ĐƯỢC BẮM THÌ NHAY TOI HAM DECREASE
JMP NO_PRESS                 ; KHONG NUT NAO BAM

INCREASE:                     ; HAM XU LY TANG 1s
CALL DISPLAY_LOOP
CLR TR1                       ; TAT TIMER1
JNB P0.0, INCREASE           ; NHAY TAI CHO DE KIEM TRA CO GIU NUT NHAN
INC R1                        ; TANG 1 GIAY
SETB TR1                      ; BAT TIMER1
JMP NO_PRESS                 ; NHAY TOI NHAN NO_PRESS
DECREASE:                     ; HAM XU LY GIAM 1s
CALL DISPLAY_LOOP
CLR TR1                       ; TAT TIMER1
JNB P0.1, DECREASE           ; NHAY TAI CHO DE KIEM TRA CO GIU NUT NHAN
DEC R1                        ; GIAM 1 GIAY
SETB TR1                      ; BAT TIMER1

NO_PRESS:
CALL DISPLAY_LOOP            ; GOI HAM QUET LED
JMP LOOP

```

-Kiểm tra nút bấm nối với pin P0.0 có bằng 0, nếu bằng 0, tức được bấm, thực hiện tắt Timer1 để ngừng bộ đếm thời gian, kiểm tra nhảy tại chỗ trong trường hợp người nhấn ấn giữ không thả nút bấm, sau đó tăng 1 giây và bật lại Timer1.

-Tiếp tục kiểm tra với pin P0.1, có bằng 0, nếu bằng 0, tức được bấm, thực hiện tắt Timer1 để ngừng bộ đếm thời gian, kiểm tra nhảy tại chỗ trong trường hợp người nhấn ấn giữ không thả nút bấm, sau đó giảm 1 giây và bật lại Timer1..

-Nếu 2 nút trên không nhấn, bỏ qua nó và thực hiện quét led và hiển thị.

```

DISPLAY_LOOP:                                ; HAM HIEN THI QUET LED
    SETB P1.0
    MOV A, R0
    MOV B, #10
    DIV AB
    MOV P2, B
    CALL DELAY_5MS
    CLR P1.0

    SETB P1.1
    MOV A, R0
    MOV B, #10
    DIV AB
    MOV P2, A
    CALL DELAY_5MS
    CLR P1.1

    SETB P1.2
    MOV A, R1
    MOV B, #10
    DIV AB
    MOV P2, B
    CALL DELAY_5MS
    CLR P1.2

    SETB P1.3 82 MOV A, R1
    MOV B, #10
    DIV AB
    MOV P2, A
    CALL DELAY_5MS
    CLR P1.3

RET

```

-Thực hiện lần lượt bật tắt các led nối tại các pin P1.0 -> P1.3

-Ta dùng lệnh DIV để chia lấy phần đơn vị và phần chục của giây và phần trăm giây đưa vào port P2 để hiển thị ra led-7-segments.

```

DELAY_10MS:                                ; HAM DELAY 10ms BANG TIMER1, DUNG DE TANG GIA TRI PHAN TRAM GIAY
    MOV TL1, #0F0H
    MOV TH1, #0D8H
    SETB TR1
    JNB TF1, $
RET

```

-Hàm delay bằng Timer1, delay 10ms, khi Timer1 tràn, sẽ gọi hàm ngắt của Timer1 ở địa chỉ 1BH và thực hiện tính toán tăng giá trị giây và phần trăm giây.

```

DELAY_5MS:                ; HAM DELAY 5ms BANG VONG LAP, DUNG DE QUET LED
    MOV R4, #10
    DELAY:
        MOV R5, #250
        DJNZ R5, $
        DJNZ R4, DELAY
    RET

```

-Hàm delay bằng vòng lặp, thời gian delay là $10 \times 250 \times 2\mu s = 5ms$. Hàm này dùng để delay khi quét led.

```

ISR_TIMER1:                ; HAM XU LY BO DEM THOI GIAN KHI TIMER1 TRAN
                            ; XU LY TANG DON VI THOI GIAN
                            ; TANG PHAN TRAM GIAY
    INC R0
    MOV A, R0
    MOV B, #100
    CJNE A, B, NOT_EQUAL    ; KIEM TRA NEU PHAN TRAM GIAY = 100 THI RESET PHAN TRAM GIAY VE 0
    MOV R0, #0
    INC R1                    ; TANG GIAY

    MOV A, R1
    MOV B, #100
    CJNE A, B, NOT_EQUAL    ; KIEM TRA NEU GIAY = 100 THI RESET GIAY VE 0
    MOV R1, #0

    NOT_EQUAL:                ; NAP LAI GIA TRI CHO TIMER1
        MOV TL1, #0F0H
        MOV TH1, #0D8H
    RETI

```

-Hàm ngắt tự động được gọi của Timer1 khi giá trị timer tràn.

-Hàm này thực hiện theo thứ tự sau:

+Tăng phần trăm giây

+Kiểm tra phần trăm giây có bằng 100, nếu bằng reset phần trăm giây về 0 và tăng giây

+Kiểm tra giây có bằng 100, nếu bằng reset giây về 0 và nạp lại giá trị cho Timer1

```

ISR_INT0:                ; HAM XU LY NUT A
    CPL TR1                ; TAT TIMER1
    RETI

```

-Hàm phục vụ ngắt ngoài 0

-Trong hàm này, ta thực hiện đảo bit TR1 ứng với bật hoặc tắt Timer1, điều này sẽ giúp ta PAUSE/RESUME clock.

```

ISR_INT1:                                ; HAM XU LY NUT B
    MOV R0, #0                            ; GAN PHAN TRAM GIAY = 0
    MOV R1, #0                            ; GAN GIAY = 0

    RETI

```

-Hàm phục vụ ngắt ngoài 1

-Trong hàm này ta reset lại giá trị của R0, R1 về 0, ứng với reset giây và phần trăm giây về 0, tức reset clock về 0.

3) Exercise report. (2 points)

***Giải thích code:**

```

LOOP:
    JNB P0.0, INCREASE
    JNB P0.1, DECREASE
    JMP NO_PRESS

INCREASE:
    CLR TR1
    JNB P0.0, $
    INC R1
    SETB TR1
    JMP NO_PRESS

DECREASE:
    CLR TR1
    JNB P0.1, $
    DEC R1
    SETB TR1

NO_PRESS:
    CALL DISPLAY_LOOP

    JMP LOOP

```

-Như đã trình bày trên phần thực hành, ta thực hiện kiểm tra 2 chức năng tăng 1s và giảm 1s bằng phần mềm như sau:

+Kiểm tra nút bấm nối với pin P0.0 có bằng 0, nếu bằng 0, tức được bấm, thực hiện tắt Timer1 để ngừng bộ đếm thời gian, kiểm tra nhảy tại chỗ trong trường hợp người nhấn ấn giữ không thả nút bấm, sau đó tăng 1 giây và bật lại Timer1.

+Tiếp tục kiểm tra với pin P0.1, có bằng 0, nếu bằng 0, tức được bấm, thực hiện tắt Timer1 để ngừng bộ đếm thời gian, kiểm tra nhảy tại chỗ trong trường hợp người nhấn ấn giữ không thả nút bấm, sau đó giảm 1 giây và bật lại Timer1.

+Nếu 2 nút trên không nhấn, bỏ qua nó và thực hiện quét led và hiển thị.

- 4) So sánh sự khác nhau giữa ngắt theo cạnh (Edge) và ngắt theo mức (Level)?
- Ngắt theo cạnh (Edge): là sử dụng ngắt dựa vào kích cạnh lên hoặc kích cạnh xuống của tín hiệu ngắt. Đối với vi điều khiển 8051, cạnh xuống được dùng để kích hoạt hàm phục vụ ngắt. Khi chân INTx đang ở mức cao 1 chu kỳ máy xuống mức thấp 1 chu kỳ máy thì ngắt được xác lập trong khoảng thời gian đó và bật bit IEx lên 1 sau đó bỏ qua các sườn xuống khác.
 - Ngắt theo mức (Level): là sử dụng ngắt dựa vào mức cao hoặc mức thấp của tín hiệu ngắt. Đối với vi điều khiển 8051, tín hiệu ngắt INTx đang ở mức cao ít nhất 1 chu kỳ máy được chuyển xuống mức thấp liên tục 4 chu kỳ máy thì một ngắt theo mức sẽ được xác lập. Khi đó vi điều khiển sẽ dừng công việc đang thực hiện để thực hiện chương trình phục vụ ngắt. Tín hiệu mức thấp phải được thả ra trước khi thực hiện lệnh cuối cùng của ISR (chương trình phục vụ ngắt), nếu không thì sẽ có thêm 1 ngắt khác được tạo ra. Điều này làm chúng ta sẽ khó xác định xem phải ấn nút bao lâu để thực hiện 1 ngắt, nếu ấn chưa đủ 4 chu kỳ máy thì chưa tạo ra ngắt, nếu ấn quá lâu thì sẽ tạo nhiều ngắt xảy ra gây nên kết quả không mong muốn.

V. References

- [1] [Thuật toán tính toán thời gian clock](#)
- [2] Tống Văn On – Hoàng Đức Hải, *HỌ VI ĐIỀU KHIỂN 8051*, Nhà xuất bản Lao Động – Xã Hội