PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS

Instructor: The Tung Than

Student's name: Nguyen Quoc Truong An

Student code: 21521810

PRACTICE REPORT NO 1

LAB1: ACQUAINTANCE WITH PROTEUS AND THE 8051 MICROCONTROLLER FAMILY

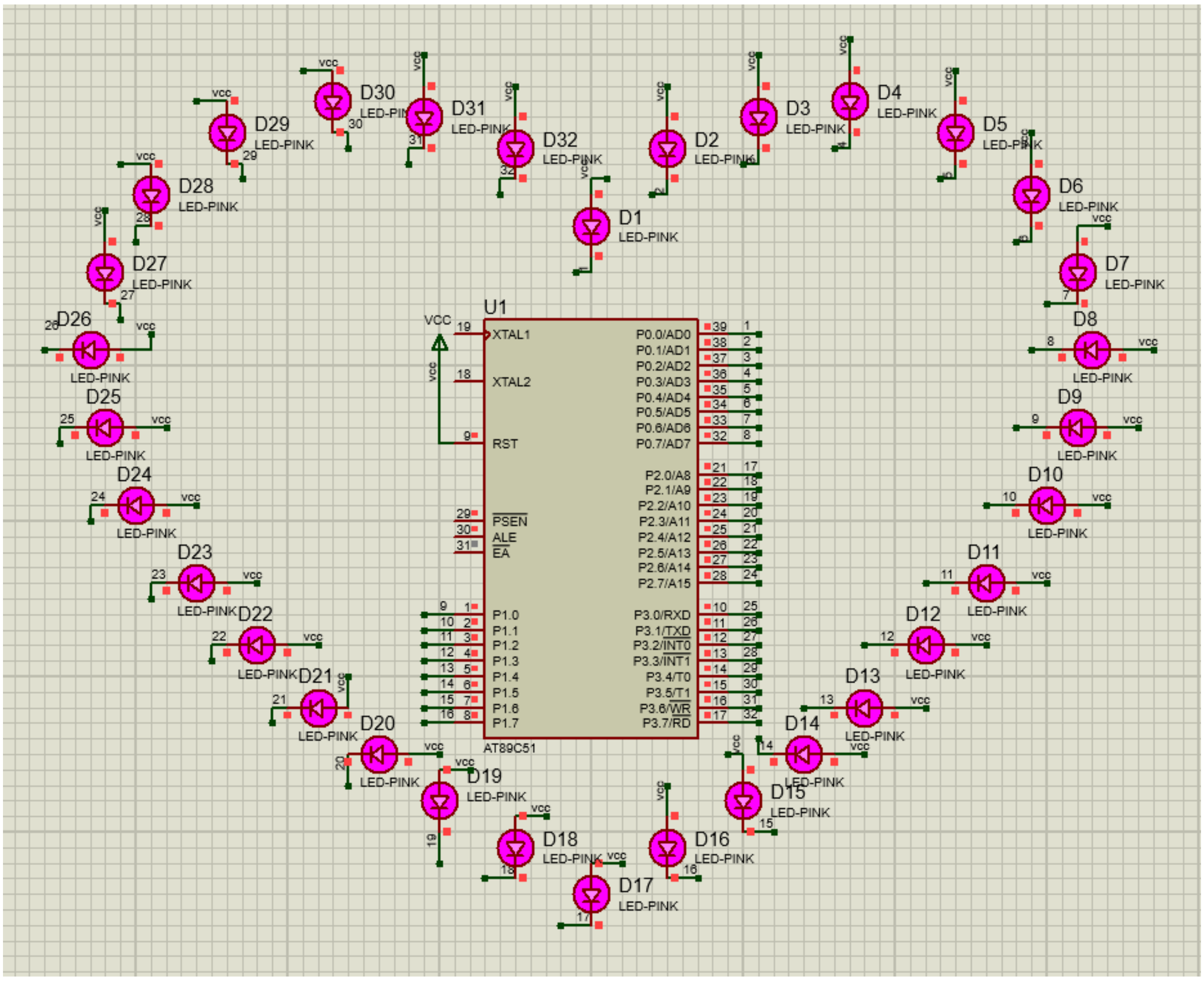
I. Student preparation

– Knowledge of Assembly programming on 8051. Download and preview "Assembly instruction set of 8051".

– Download and install Proteus software version 8.6.

II. Practice content

1) **Design a heart-led circuit consisting of 32 LEDs controlled by AT89C51 (2p)**



2) **Build an Assembly program so that circuit can run continously with at least 3 effects, each effect runs within 5 seconds. (2p)**

```
;====================================================================
; Main.asm file generated by New Project wizard
;
; Created: Mon Mar 20 2023
; Processor: AT89C51
; Compiler: ASEM-51 (Proteus)
;====================================================================

$NOMOD51
$INCLUDE (8051.MCU)

;====================================================================
; DEFINITIONS
;====================================================================

;====================================================================
; VARIABLES
;====================================================================

;====================================================================
; RESET and INTERRUPT VECTORS
;====================================================================
; Reset Vector
ORG 0000H
JMP Start

;====================================================================
; CODE SEGMENT
;====================================================================
ORG 0100H

Start:
MOV R3, #10                 ; initial number of repetitions for effect 1

Loop:
    CALL Loop_effect_1      ; call effect 1 function
    CALL Loop_effect_2      ; call effect 2 function
    CALL Loop_effect_3      ; call effect 3 function

JMP Loop                    ; repeat all the effect

Delay_1:                    ; Delay function with delay time: 2*250*250*2us = 0.25s
    MOV R0, #2
    Loop1:
        MOV R1, #250
    Loop2:
        MOV R2, #250
        DJNZ R2, $
        DJNZ R1, Loop2
        DJNZ r0, Loop1
RET
```

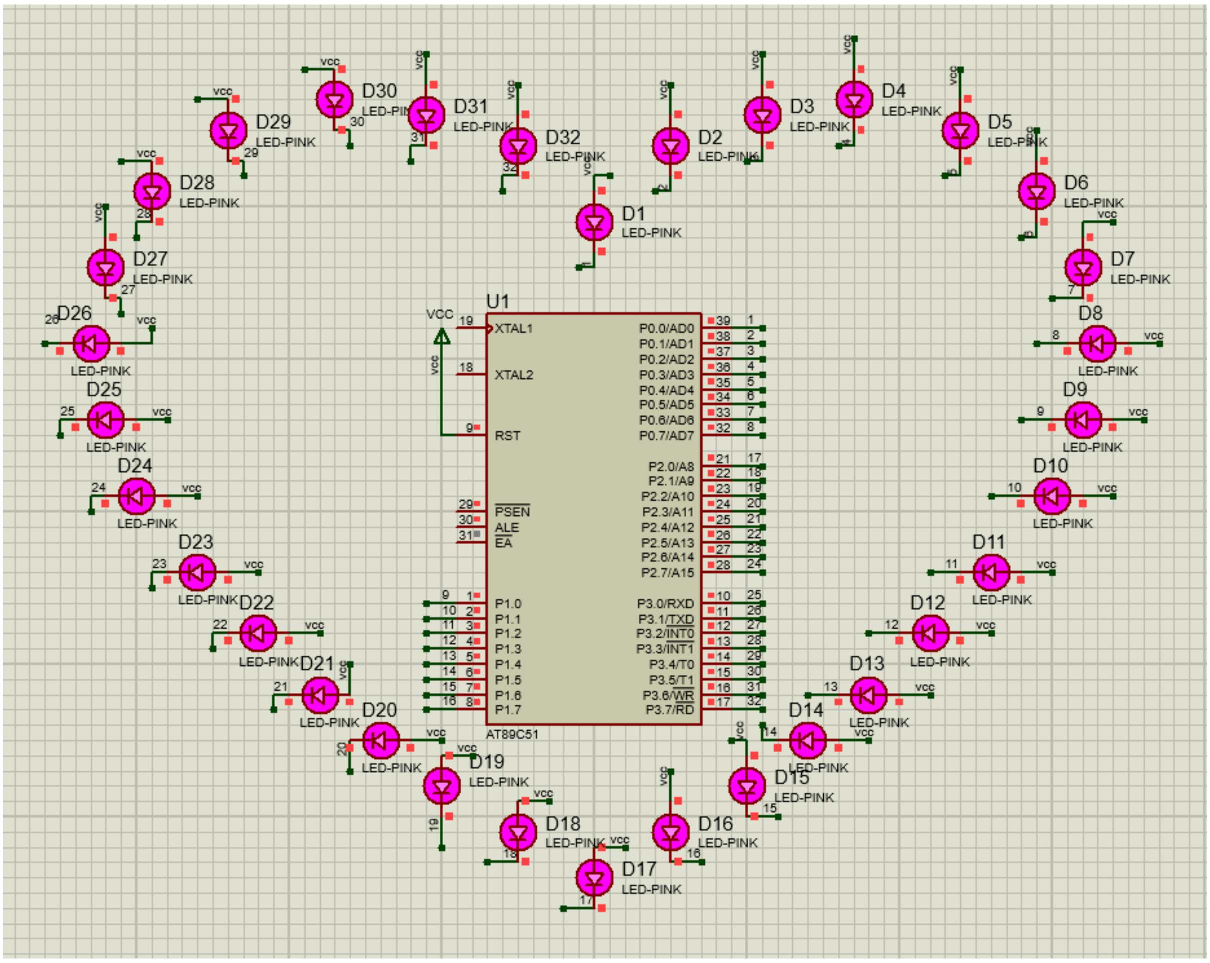
```
Delay_2:                    ; Delay fuction with delay time: 125*125*2us = 0.03125s
    MOV R0, #125
    Loop3:
        MOV R1, #125
        DJNZ R1, $
        DJNZ R0, Loop3
RET

Loop_effect_1:              ; Effect 1 function
    MOV A, #0               ; Assign A = 0
    MOV P0, A               ; Assign P0 = A, turn on all Leds in port P0
    MOV P2, A               ; Similar to P0
    MOV P1, A               ; Similar to P0
    MOV P3, A               ; Similar to P0
    CALL Delay_1            ; Delay 0.25s
    MOV A, #0FFH            ; Assign A = FFH
    MOV P1, A               ; Assign P1 = A, turn off all Leds in port P1
    MOV P3, A               ; Similar to P1
    MOV P0, A               ; Similar to P1
    MOV P2, A               ; Similar to P1
    CALL Delay_1            ; Delay 0.25s
    DJNZ R3, Loop_effect_1  ; Repeat Loop_effect_1 10 times
    MOV R3, #10             ; reassign R3 = 10 that mean the next effect will be repeated 10 times
RET

Loop_effect_2:              ; Effect 2 function
    MOV A, #10101010b       ; Assign A = 10101010b
    MOV P0, A               ; Assign P0 = A, turn on Leds at position 0,2,4,6, the rest are off
    MOV P1, A               ; Similar to P0
    MOV P2, A               ; Similar to P0
    MOV P3, A               ; Similar to P0

    CALL Delay_1            ; Delay 0.25s
    MOV A, #01010101b       ; Assign A = 01010101b
    MOV P0, A               ; Assign P0 = A, turn on Leds at position 1,3,5,7, the rest are off
    MOV P1, A               ; Similar to P0
    MOV P2, A               ; Similar to P0
    MOV P3, A               ; Similar to P0

    CALL Delay_1            ; Delay 0.25s
    DJNZ R3, Loop_effect_2  ; Repeat Loop_effect_2 10 times
    MOV R3, #5              ; reassign R3 = 5 sothe next effect will be repeated 5 times
RET

Loop_effect_3:              ; Effect 3 function
    MOV R4, #8              ; Assign R4 = 8
    MOV R5, #0              ; Assign R5 = 0
    MOV A, #0FFH            ; Assign A = FFH
    MOV P0, A               ; Assign P0 = A, turn off all Leds in port P0
    MOV P1, A               ; Similar to P0
    MOV P2, A               ; Similar to P0
    MOV P3, A               ; Similar to P0

    LAP1:                   ; Create effect for P1, P2
        MOV DPTR, #MALED1   ; Move the first address in MALED1 array into DPTR register
        MOV A, R5           ; Assign A = R5
        MOVC A, @A+DPTR     ; Get the value at address A+DPTR assign to A
        MOV P1, A           ; Assign P1 = A, turn on the LED at respective position
        MOV DPTR, #MALED2   ; Move the first address in MALED2 array into DPTR register
        MOV A, R5          ; Assign A = R5
        MOVC A, @A+DPTR    ; Get the value at address A+DPTR assign to A
        MOV P2, A          ; Assign P2 = A, turn on the LED at respective position
        INC R5             ; Increase R5, use the next element in MALED1 and MALED2 array
        CALL Delay_2       ; Delay 0.03125s
        DJNZ R4, LAP1      ; Repeat LAP1 8 times because R4 = 8 at the begining
        MOV R4,#8          ; reassign R4 = 8 so LAP2 will be repeated 8 times
        MOV R5,#0          ; Assign R5 = 0, return the first element in array

    LAP2:                  ; Create effect for P3, P0
        MOV DPTR, #MALED2  ; The idea is similar to LAP1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P3, A
        MOV DPTR, #MALED1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P0, A
        INC R5
        CALL Delay_2
        DJNZ R4, LAP2
        MOV R4,#8
        MOV R5,#0

    LAP3:                  ; Create effect for P0, P3
        MOV DPTR, #MALED2  ; The idea is similar to LAP1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P0, A
        MOV DPTR, #MALED1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P3, A
        INC R5
        CALL Delay_2
        DJNZ R4, LAP3
        MOV R4,#8
        MOV R5,#0

    LAP4:                  ; Create effect for P2, P1
        MOV DPTR, #MALED1  ; The idea is similar to LAP1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P2, A
        MOV DPTR, #MALED2
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P1, A
        INC R5
        CALL Delay_2
        DJNZ R4, LAP4
    DJNZ R3, Loop_effect_3 ; Repeat Loop_effect_3 5 times
    MOV R3, #10            ; reassign R3 = 10 so the next effect will be repeated 10 times
RET
                           ; MALED1 and MALED2 Array, store values to make LEDs effect
MALED1: DB 01111111b,00111111b,00011111b,00001111b,00000111b,00000011b,00000001b,00000000b
MALED2: DB 11111110b,11111100b,11111000b,11110000b,11100000b,11000000b,10000000b,00000000b
;====================================================================
END
```

III. Exercises

Research and write the steps of printed circuits from the design step on the proteus to the step of completing the printed circuit in practice.

IV. Report

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



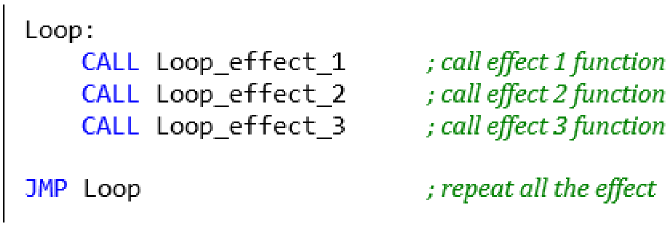
2) Explain the operating principle of the effects, accompanied by a video (send a Google Drive link) to demonstrate the circuit operation in case the instructor can’t run design file. (3p)

***** Google Drive link:**

https://drive.google.com/drive/folders/1qeNdhYbpOr2WTPzGTl4xiMCXLU7JjLt7?usp=sharing

*Explain the operating principle of the effects:

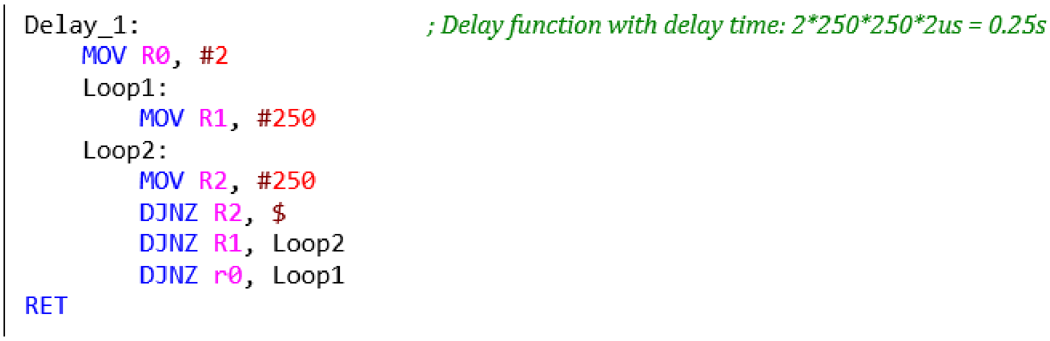
*Loop (Main):



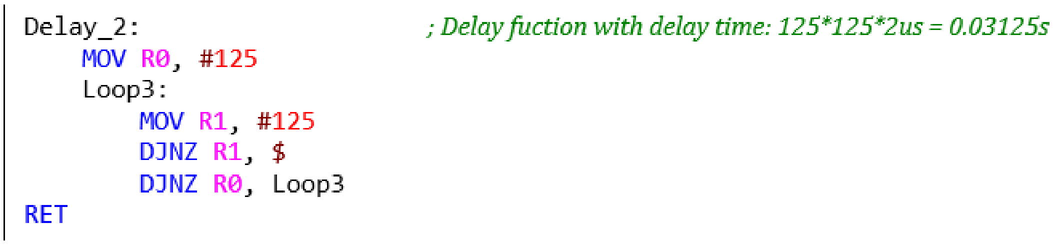
```
Loop:
    CALL Loop_effect_1        ; call effect 1 function
    CALL Loop_effect_2        ; call effect 2 function
    CALL Loop_effect_3        ; call effect 3 function

JMP Loop                      ; repeat all the effect
```

-Call and repeat all the effect

*Delay function:



```
Delay_1:                         ; Delay function with delay time: 2*250*250*2us = 0.25s
    MOV R0, #2
    Loop1:
        MOV R1, #250
    Loop2:
        MOV R2, #250
        DJNZ R2, $
        DJNZ R1, Loop2
        DJNZ r0, Loop1
RET
```



```
Delay_2:                         ; Delay fuction with delay time: 125*125*2us = 0.03125s
    MOV R0, #125
    Loop3:
        MOV R1, #125
        DJNZ R1, $
        DJNZ R0, Loop3
RET
```

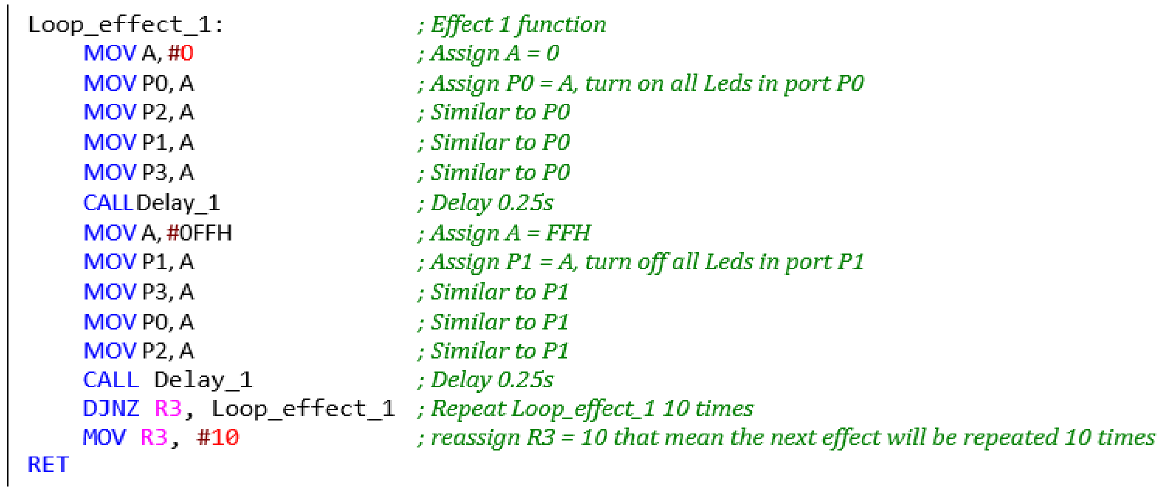
-The delay function is used to make sure people can see the effect change in specific time period.

-Delay_1 and Delay_2 include two or three nested loops, the number of loops stored in R0, R1, R2.

-DJNZ instruction completed in two clock cycles. The clock frequency is 12MHz/12, so the clock cycle is 1us.

-So delay times of Delay_1 function equal to 2*250*250*2us = 0.25s and the delay times of Delay_2 function equal to 125*125*2us = 0.03125s

*First effect:

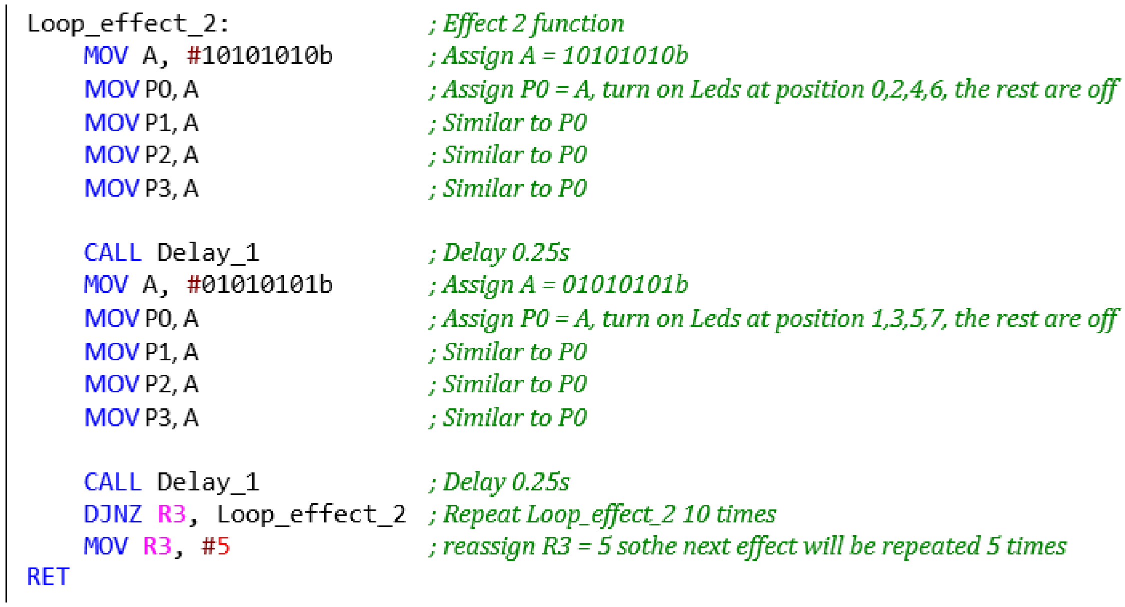


```
Loop_effect_1:                  ; Effect 1 function
        MOV A, #0               ; Assign A = 0
        MOV P0, A               ; Assign P0 = A, turn on all Leds in port P0
        MOV P2, A               ; Similar to P0
        MOV P1, A               ; Similar to P0
        MOV P3, A               ; Similar to P0
        CALL Delay_1            ; Delay 0.25s
        MOV A, #0FFH            ; Assign A = FFH
        MOV P1, A               ; Assign P1 = A, turn off all Leds in port P1
        MOV P3, A               ; Similar to P1
        MOV P0, A               ; Similar to P1
        MOV P2, A               ; Similar to P1
        CALL  Delay_1           ; Delay 0.25s
        DJNZ  R3, Loop_effect_1 ; Repeat Loop_effect_1 10 times
        MOV  R3,  #10           ; reassign R3 = 10 that mean the next effect will be repeated 10 times
RET
```

-The idea of the first effect is turn on all LEDs then delay 0.25s and then turn off and delay 0.25s. Total delay in one loop is 0.5s.

-Each loop of Loop_effect_1 takes 0.5s, so it takes 10 loops within 5s.

*Second effect:



```
Loop_effect_2:                   ; Effect 2 function
        MOV  A,  #10101010b      ; Assign A = 10101010b
        MOV P0, A                ; Assign P0 = A, turn on Leds at position 0,2,4,6, the rest are off
        MOV P1, A                ; Similar to P0
        MOV P2, A                ; Similar to P0
        MOV P3, A                ; Similar to P0

        CALL  Delay_1            ; Delay 0.25s
        MOV  A,  #01010101b      ; Assign A = 01010101b
        MOV P0, A                ; Assign P0 = A, turn on Leds at position 1,3,5,7, the rest are off
        MOV P1, A                ; Similar to P0
        MOV P2, A                ; Similar to P0
        MOV P3, A                ; Similar to P0

        CALL  Delay_1            ; Delay 0.25s
        DJNZ  R3,  Loop_effect_2 ; Repeat Loop_effect_2 10 times
        MOV  R3,  #5             ; reassign R3 = 5 sothe next effect will be repeated 5 times
RET
```

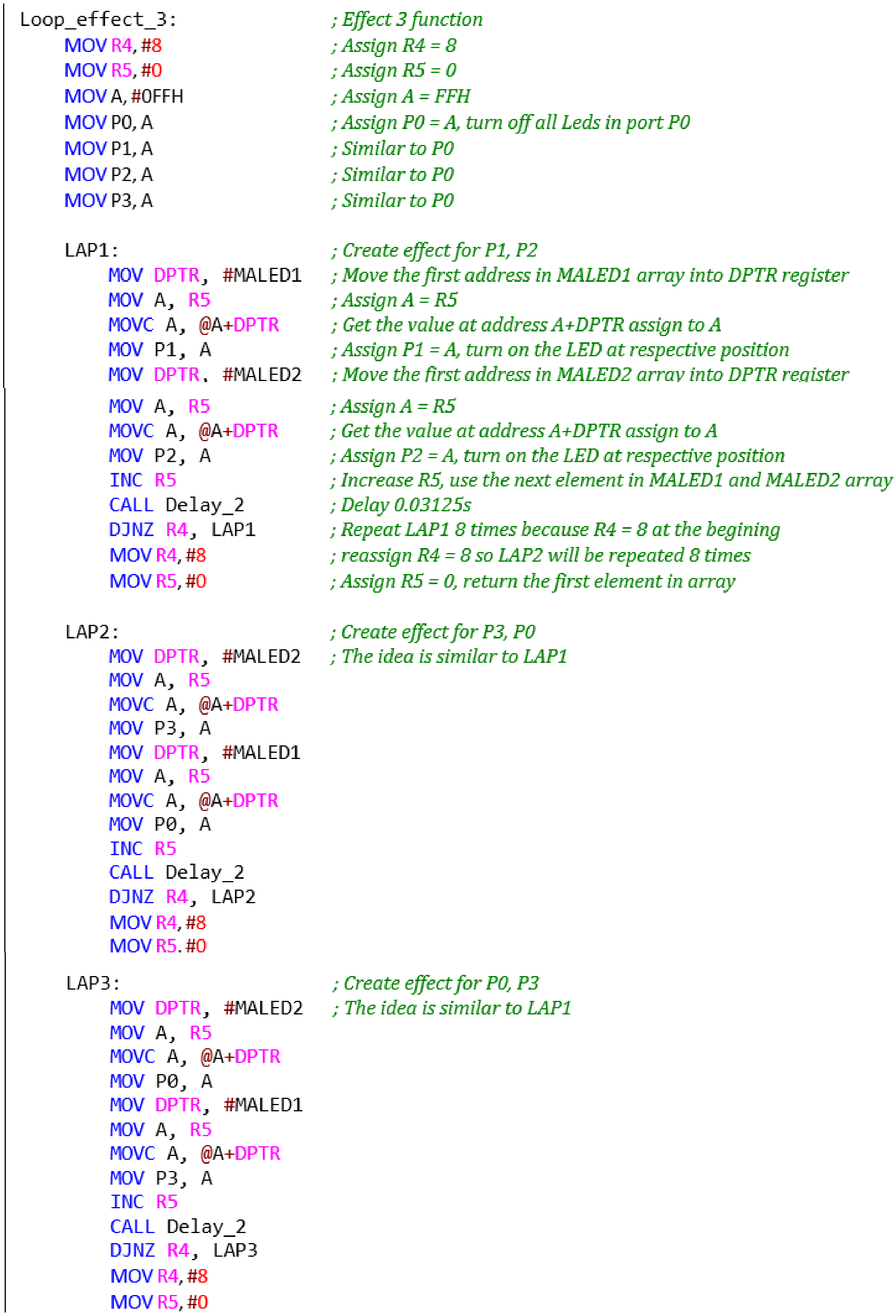
-The idea of the second effect is to turn on and turn off the LEDs alternately.

-First, all even position at P0, P1, P2, P3 port are turned on. Then turn it off and delay 0.25s

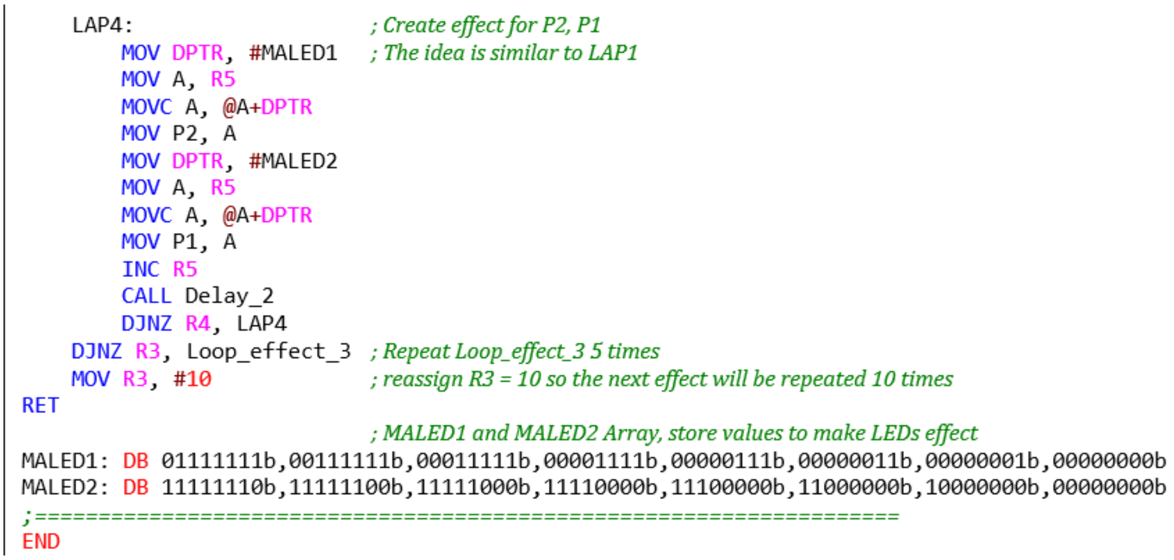
-Next, all odd position at P0, P1, P2, P3 port are turned on. Then turn it off and delay 0.25s.

-Total delay in one loop is 0.5s, so we need to repeat 10 times to do the effect within 5s.

*Third effect:



```
Loop_effect_3:                  ; Effect 3 function
    MOV R4, #8                  ; Assign R4 = 8
    MOV R5, #0                  ; Assign R5 = 0
    MOV A, #0FFH                ; Assign A = FFH
    MOV P0, A                   ; Assign P0 = A, turn off all Leds in port P0
    MOV P1, A                   ; Similar to P0
    MOV P2, A                   ; Similar to P0
    MOV P3, A                   ; Similar to P0

    LAP1:                       ; Create effect for P1, P2
        MOV DPTR, #MALED1       ; Move the first address in MALED1 array into DPTR register
        MOV A, R5               ; Assign A = R5
        MOVC A, @A+DPTR         ; Get the value at address A+DPTR assign to A
        MOV P1, A               ; Assign P1 = A, turn on the LED at respective position
        MOV DPTR, #MALED2       ; Move the first address in MALED2 array into DPTR register
        MOV A, R5               ; Assign A = R5
        MOVC A, @A+DPTR         ; Get the value at address A+DPTR assign to A
        MOV P2, A               ; Assign P2 = A, turn on the LED at respective position
        INC R5                  ; Increase R5, use the next element in MALED1 and MALED2 array
        CALL Delay_2            ; Delay 0.03125s
        DJNZ R4, LAP1           ; Repeat LAP1 8 times because R4 = 8 at the begining
        MOV R4, #8              ; reassign R4 = 8 so LAP2 will be repeated 8 times
        MOV R5, #0              ; Assign R5 = 0, return the first element in array

    LAP2:                       ; Create effect for P3, P0
        MOV DPTR, #MALED2       ; The idea is similar to LAP1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P3, A
        MOV DPTR, #MALED1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P0, A
        INC R5
        CALL Delay_2
        DJNZ R4, LAP2
        MOV R4, #8
        MOV R5, #0

    LAP3:                       ; Create effect for P0, P3
        MOV DPTR, #MALED2       ; The idea is similar to LAP1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P0, A
        MOV DPTR, #MALED1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P3, A
        INC R5
        CALL Delay_2
        DJNZ R4, LAP3
        MOV R4, #8
        MOV R5, #0
```



```
    LAP4:                      ; Create effect for P2, P1
        MOV DPTR, #MALED1      ; The idea is similar to LAP1
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P2, A
        MOV DPTR, #MALED2
        MOV A, R5
        MOVC A, @A+DPTR
        MOV P1, A
        INC R5
        CALL Delay_2
        DJNZ R4, LAP4
    DJNZ R3, Loop_effect_3     ; Repeat Loop_effect_3 5 times
    MOV R3, #10                ; reassign R3 = 10 so the next effect will be repeated 10 times
RET
                               ; MALED1 and MALED2 Array, store values to make LEDs effect
MALED1: DB 01111111b,00111111b,00011111b,00001111b,00000111b,00000011b,00000001b,00000000b
MALED2: DB 11111110b,11111100b,11111000b,11110000b,11100000b,11000000b,10000000b,00000000b
;====================================================================
END
```

-The idea off third effect is to turn on the LEDs one after another.

-Using value in MALED1 and MALED2 array to turn on, turn off in the desired order.

-In the LAL1, we turn on the LEDs at port P1, P2 in bottom-up order.

-In the LAL2, we turn on the LEDs at port P0, P3 in bottom-up order.

-In the LAL3, we turn on the LEDs at port P0, P3 in top-down order.

-In the LAL4, we turn on the LEDs at port P1, P2 in top-down order.

-Total delay in one Loop_effect_3 loop is 4*0.03125*8 = 1s, so we need to repeat 5 times to do the effect within 5s.

3) Presentation of the steps to implement the printed circuit. (2p)

1. Schematic Design: Create a schematic of the circuit using a software like Proteus, Altium Designer or EasyEDA. Ensure that all components are correctly placed and connected.
2. PCB Layout Design: Once having a schematic, use the same software to create the physical layout of the PCB. Arrange all the components in the desired location, route the conections between them and add any necessary labels or markings.
3. Printing the layout: Once having a completed layout design, print it on a glossy paper using laser printer. Be sure to print the design in mirror image, so that it can be transferred to the copper board correctly.
4. Transfering the layout onto the board: Using the heat transfer method to transfer the layout from the glossy paper onto the copper-coated board.

Place the glossy paper with the printed layout onto the copper board and apply heat and pressure using an iron or laminator.

5. Etching: After the transfer process, etch the unwanted copper using a solution like Ferric Chloride or Ammonium Persulphate to create the traces and pads on the board.
6. Drilling: After etching board, drill holes into the board where the components will be mounted. Use a drill bit that is appropriate for the size of the components and be sure to drill the holes in the correct locations.
7. Solder the components: Once having drilled holes, solder the components onto the PCB using a soldering iron.
8. Testing and Debugging: After assembly, test the circuit to make sure it functions properly. Use a multimeter or oscilloscope to measure voltage, current, and signal levels at various points in the circuit. If any issues arise during testing, troubleshoot the circuit by checking for errors in the schematic or PCB layout design, and inspecting the physical connections for mistakes or faults.
9. Finalization: Once the circuit has been successfully tested and any errors have been resolved, finalize the PCB by applying a protective coating or enclosure to protect the components and ensure longevity of the circuit.

V. References

https://www.win.tue.nl/~aeb/comp/8051/set8051.html?fbclid=IwAR3fBgbMNw3OVxWRrqSdci2ODxwbuy2Nro_2Qo41htdCfUqGZ24UHHoIEsE#51jz

https://www.youtube.com/watch?v=FhQJzDO9Cfo&t=314s