THE MICROPROCESSORS & MICROCONTROLLERS

Instructor: The Tung Than

Student: Bach Hoang Phan Thanh

ID : 21520599

PRACTICE EXERCISE #1:

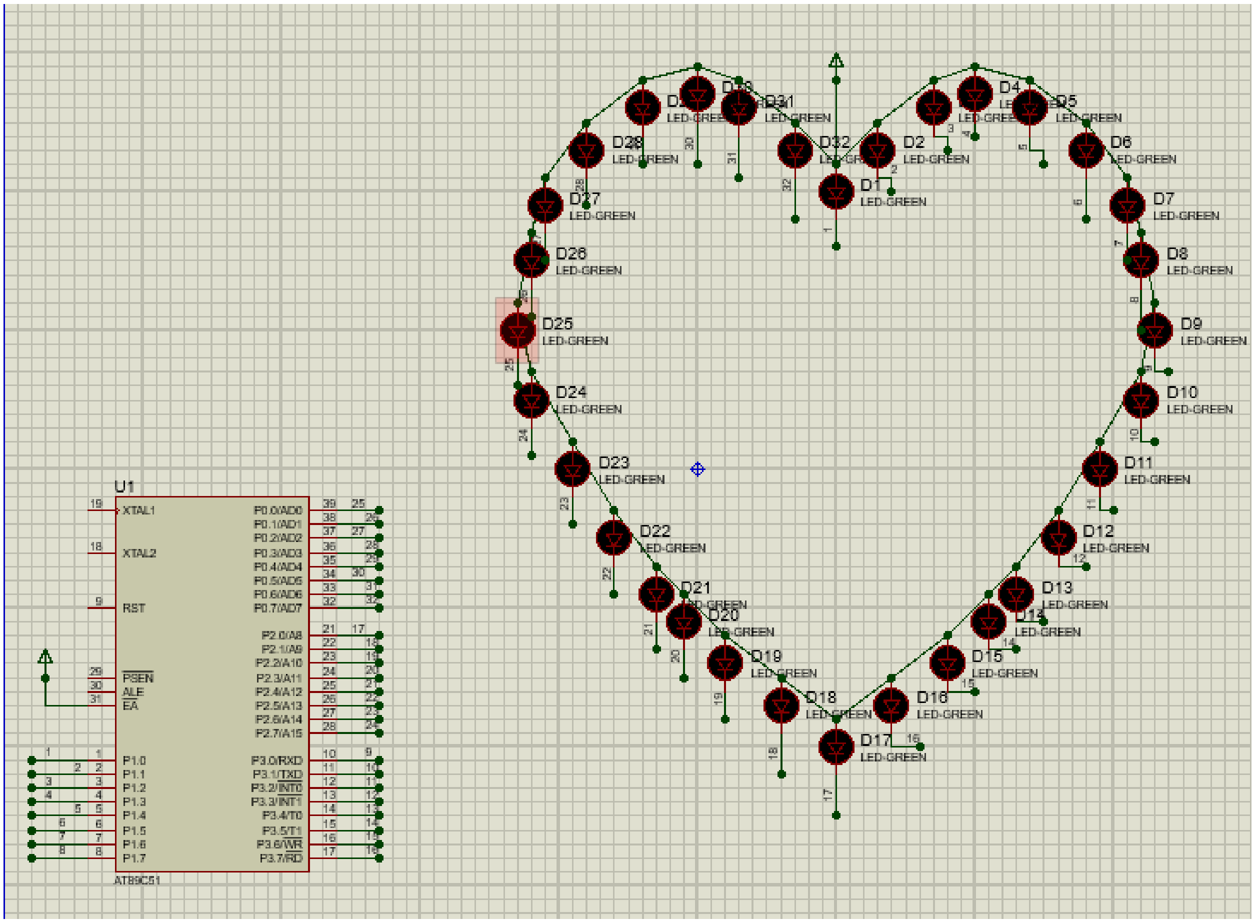
ACQUAINTANCE WITH PROTEUS AND THE 8051 MICROCONTROLLER FAMILY

Table of contents

1. Design Result.. 3
2. Explain the operating principle of the effects 4
3. Presentation of the steps to implements the printed circuit.................... 14

1. Design Result

Design a heart-led circuit consisting of 32 LEDs controlled by AT89C51.



2. Explain the operating principle of the effects

Google Drive link: https://tinyurl.com/54x9x4xe

Source code, include explanation:

Note: When copy from Proteus to Word, the color for each instruction, initial number has been changed to black. So I try to change the color like the same which it display on Proteus.

Source Code (Include English Explanation)

```
;===========================================================
==============
; Main.asm file generated by New Project wizard
;
; Created:   Wed Mar 15 2023
; Processor: 80C31
; Compiler:  ASEM-51 (Proteus)
;===========================================================
==============

$NOMOD51
$INCLUDE (8051.MCU)

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


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


;===========================================================
==============
; RESET and INTERRUPT VECTORS
```

```
;=======================================================
=============

    ; Reset Vector
    org   0000h
    jmp   Start

;=======================================================
=============
; CODE SEGMENT
;=======================================================
=============

    org   0100h  ;set the starting address
Start:
    ; Write your code here
    org 00h   ; set origin .address, turn off all leds

main:
    mov r0, #4          ; Repeat Mode1 for 4 times
l1:
    call Mode1
    djnz r0, l1

    mov r0, #4          ; Repeat Mode2 for 4 times
l2:
    call Mode2
    djnz r0, l2

    mov r0, #8          ; Repeat Mode3 for 8 times
l3:
    call Mode3
    djnz r0, l3

jmp main
;; in each mode, when bit is set to 1 mean the led turn off
;; when bit is set to 0, the led turn on
;; Mode1 to turn on the bulbs from leg 0 to 8 of each port
Mode1:
            mov P0, #11111111B
mov P1, #11111111B
mov P2, #11111111B
mov P3, #11111111B
call Delay
mov P0, #11111110B
mov P1, #11111110B
mov P2, #11111110B
mov P3, #11111110B
call Delay
mov P0, #11111100B
mov P1, #11111100B
mov P2, #11111100B
mov P3, #11111100B
call Delay
mov P0, #11111000B
mov P1, #11111000B
mov P2, #11111000B
mov P3, #11111000B
call Delay
mov P0, #11110000B
mov P1, #11110000B
mov P2, #11110000B
mov P3, #11110000B
call Delay
mov P0, #11100000B
mov P1, #11100000B
mov P2, #11100000B
mov P3, #11100000B
call Delay
mov P0, #11000000B
mov P1, #11000000B
mov P2, #11000000B
mov P3, #11000000B
call Delay
mov P0, #10000000B
mov P1, #10000000B
mov P2, #10000000B
mov P3, #10000000B
call Delay
mov P0, #00000000B
```

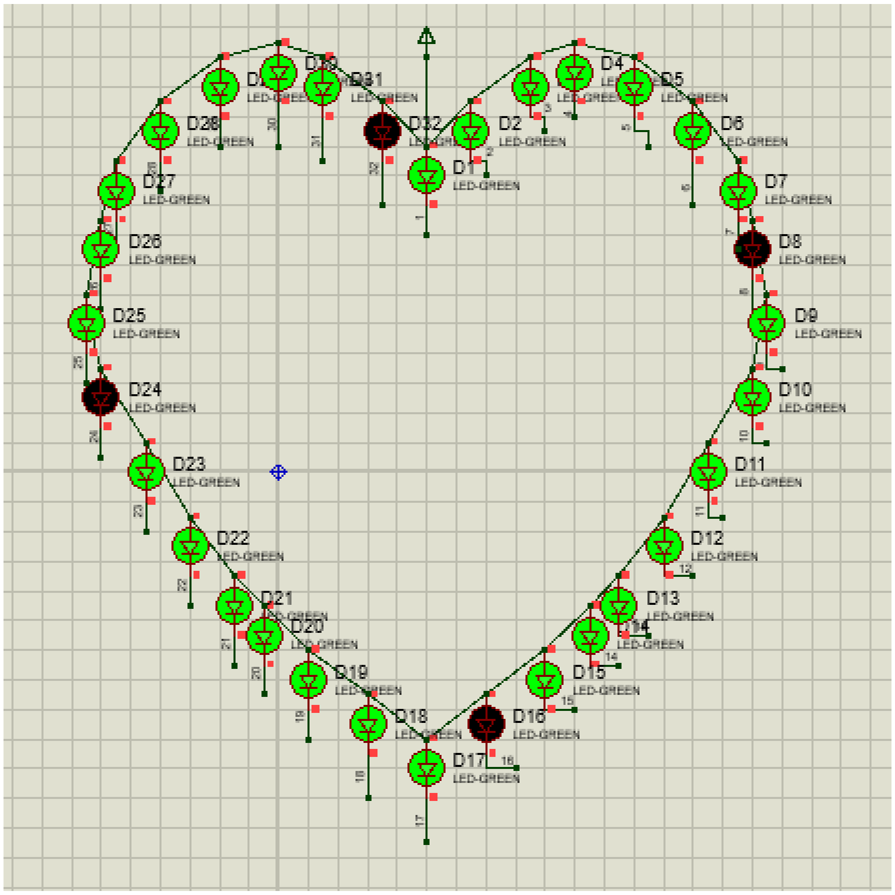
```
                mov P1, #00000000B
                mov P2, #00000000B
                mov P3, #00000000B
                call Delay

ret
;;Mode2 to turn off the bulbs from leg 8 to 0 of each port
Mode2:
                mov P0, #10000000B
                mov P1, #10000000B
                mov P2, #10000000B
                mov P3, #10000000B
                call Delay
                mov P0, #11000000B
                mov P1, #11000000B
                mov P2, #11000000B
                mov P3, #11000000B
                call Delay
                mov P0, #11100000B
                mov P1, #11100000B
                mov P2, #11100000B
                mov P3, #11100000B
                call Delay
                mov P0, #11110000B
                mov P1, #11110000B
                mov P2, #11110000B
                mov P3, #11110000B
                call Delay
                mov P0, #11111000B
                mov P1, #11111000B
                mov P2, #11111000B
                mov P3, #11111000B
                call Delay
                mov P0, #11111100B
                mov P1, #11111100B
                mov P2, #11111100B
                mov P3, #11111100B
                call Delay
                mov P0, #11111110B
                mov P1, #11111110B
```

```
        mov P2, #11111110B
        mov P3, #11111110B
        call Delay
        mov P0, #11111111B
        mov P1, #11111111B
        mov P2, #11111111B
        mov P3, #11111111B
        call Delay
ret
;;Mode 3 to turn off two consecutive bulbs, the turn on them and turn off two next consecutive bulbs, end up with turn off all bulbs
Mode3:
        mov P0, #00000000B
        mov P1, #00000000B
        mov P2, #00000000B
        mov P3, #00000000B
        call Delay
        mov P0, #00000011B
        mov P1, #00000011B
        mov P2, #00000011B
        mov P3, #00000011B
        call Delay
        mov P0, #00001100B
        mov P1, #00001100B
        mov P2, #00001100B
        mov P3, #00001100B
        call Delay
        mov P0, #00110000B
        mov P1, #00110000B
        mov P2, #00110000B
        mov P3, #00110000B
        call Delay
        mov P0, #11000000B
        mov P1, #11000000B
        mov P2, #11000000B
        mov P3, #11000000B
        call Delay
        mov P0, #11111111B
        mov P1, #11111111B
        mov P2, #11111111B
```

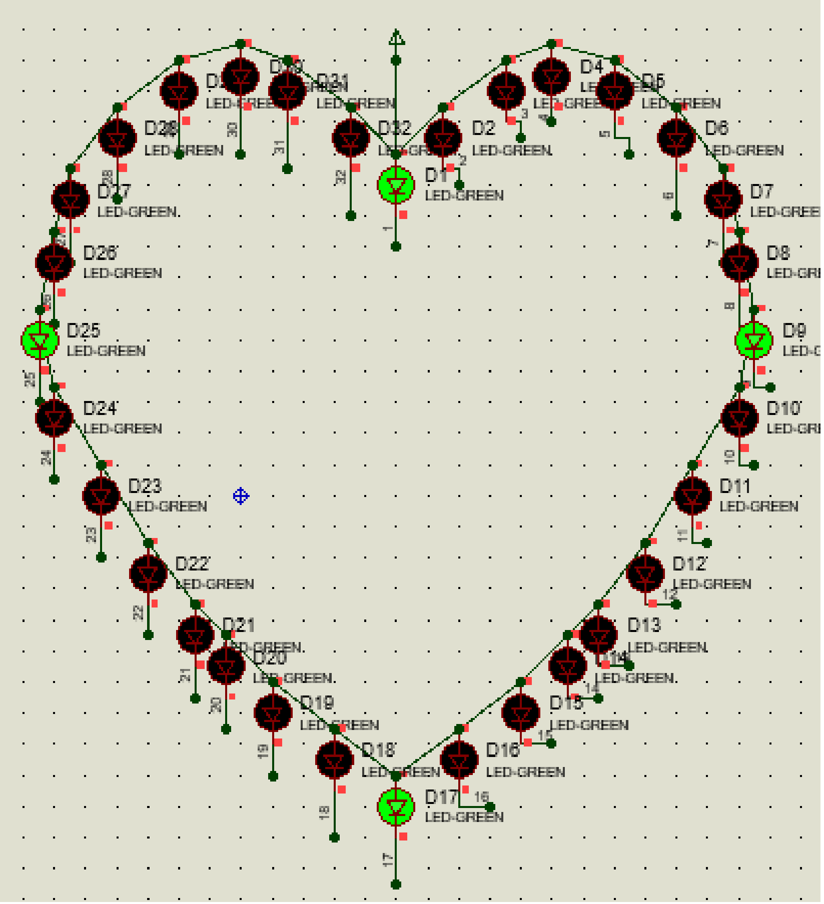
```
                mov P3, #11111111B
                call Delay
ret


;;create 3 nested loops that each count down from an initial value
;;each time a loop decrements its counter, it jump back to the beginning of that loop until the counter reaches zero
;;will create a delay - the processor must spend time executing these loops before moving on to the next instruction
Delay:
   mov r1, #30
; Set the initial loop of r1 for 15
Loop1:
   mov r2, #50
; Set the initial loop of r2 for 50
Loop2:
   mov r3, #50
; Set the initial loop of r3 for 50
   djnz r3, $
; decrements the value of r3 by 1 and jumps back to the current location - '$' symbol if it is not zero
   djnz r2, Loop2
; decrements the value of r2 by 1 and jumps back to Loop2 it is not zero
   djnz r1, Loop1
; decrements the value of r3 by 1 and jumps back to Loop1 if it is not zero
ret
;; djnz instruction takes 2 clock cylcles
;;for r1=30, r2=r3=50, the total time of delay is 15*50*50*2µs = 0.15s
end
```

Picture of each effect (more clearly on video):

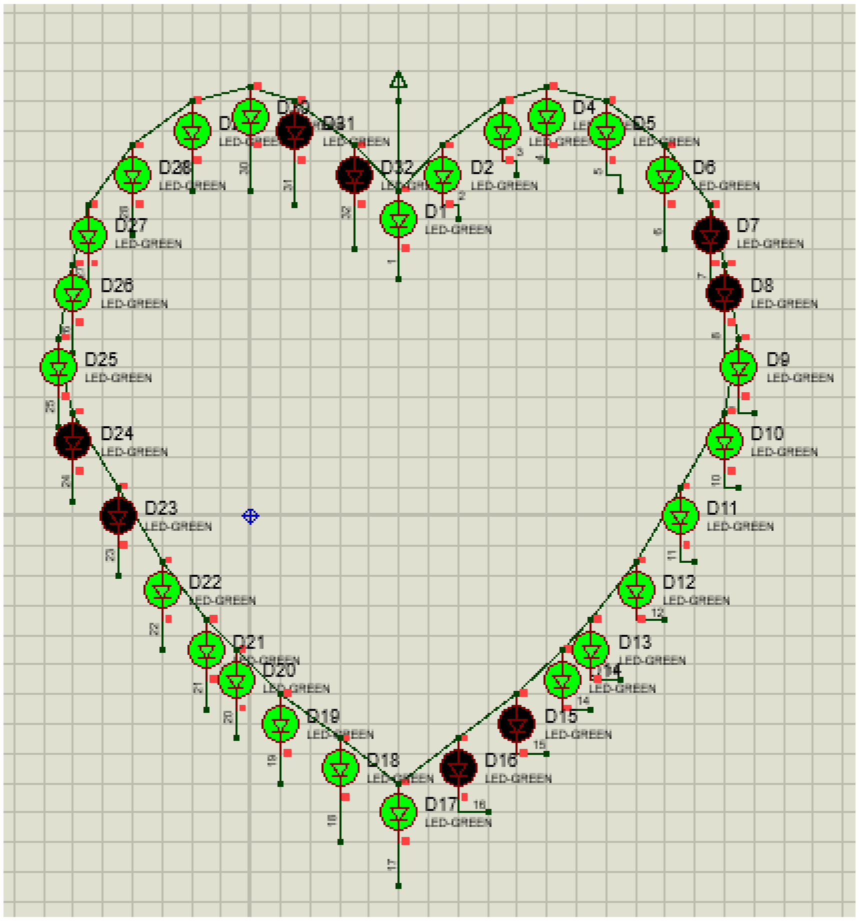
Effect 1:



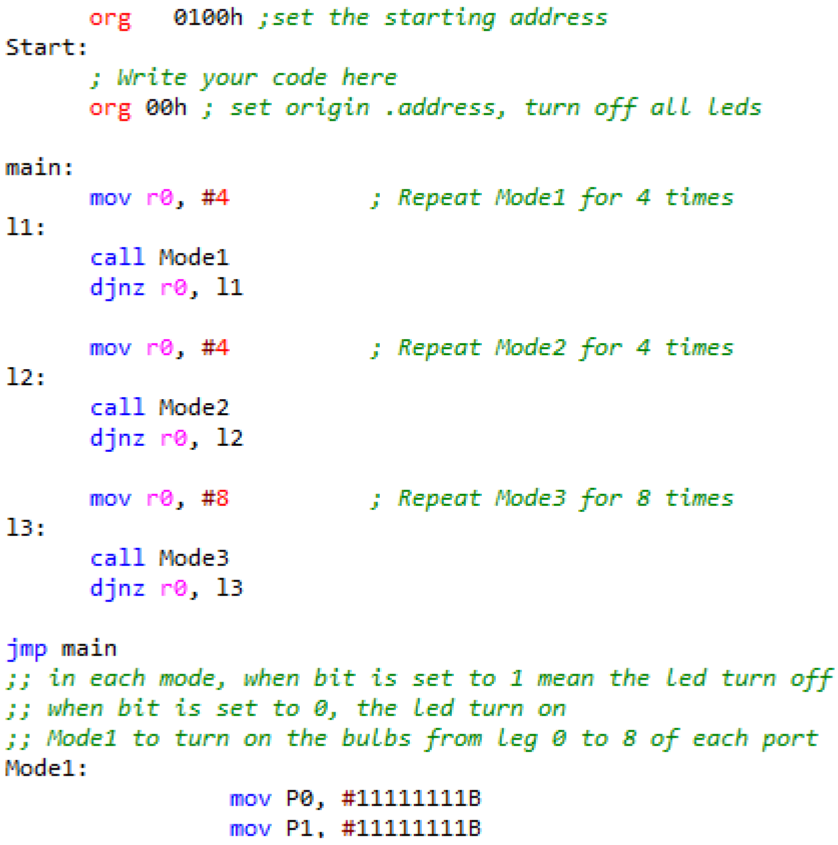
Effect 2:


D29
D30
D31
D32
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D21
D22
D23
D24
D25
D26
D27
D28
LED-GREEN

Effect 3


D31
D32
D1
D2
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D21
D22
D23
D24
D25
D26
D27
D28
LED-GREEN

Picture of source code:



```
      org   0100h ;set the starting address
Start:
      ; Write your code here
      org 00h ; set origin .address, turn off all leds

main:
      mov r0, #4          ; Repeat Mode1 for 4 times
l1:
      call Mode1
      djnz r0, l1

      mov r0, #4          ; Repeat Mode2 for 4 times
l2:
      call Mode2
      djnz r0, l2

      mov r0, #8          ; Repeat Mode3 for 8 times
l3:
      call Mode3
      djnz r0, l3

jmp main
;; in each mode, when bit is set to 1 mean the led turn off
;; when bit is set to 0, the led turn on
;; Mode1 to turn on the bulbs from leg 0 to 8 of each port
Mode1:
                mov P0, #11111111B
                mov P1, #11111111B
```

3. Presentation of the steps to implements the printed circuit

Design Step on Proteus:

Step 1: Create a new project on Proteus software

Step 2: Add the AT89C51 microcontroller to the schematic diagram and connect it to the pins of the LEDS. Put and arrange 32 LEDs to make a heart circuit.

Step 3: Add any necessary resistors or other components to the circuit

Step 4: Run a simulation to verify the circuit design and make any necessary adjustments

Programing Steps (Build an Assembly Program)

Step 1: Right click on the AT89CT51, choose Soucre Code to edit, build a new Assembly Program

Step 2: Implement the code to run different effects (Mode1, Mode2, Mode3)

Step 3:Debug and test the code to make sure that it runs correctly on the microcontroller

PCB Layout Steps:

Step 1: Switch to the PCB layout tool on Proteus

Step 2: Create the PCB layout of the circuit based on the schematic diagram.

Step 3: Arrange the components and their connections on the PCB layout.

Step 4: Add any necessary labels or notes to the PCB layout for clarity.

Step 5: Run a Design Rule Check (DRC) to verify that the PCB layout meets the necessary design rules.

Step 6: Export the PCB layout in Gerber file format for manufacturing.

Manufacturing Steps:

Step 1: Send the Gerber file to a PCB manufacturer (ex: maihungpcb, Kim Son Co PCB,…) to fabricate the printed circuit board

Step 2: Open the PCB is received, mount the components onto the PCB according to the PCB layout

Step 3: Double-check the connections and solder joints to ensure proper assembly

Step 4: Test the completed circuit to verify that it runs the program's effects correctly.