# Microprocessor & Embedded System Lab Report

*by*

**Md. Shakibul Islam**

CSE 078 08427

**Supervised by:** Tarikuzzaman Emon

Assistant Professor

Department of Computer Science and Engineering

STAMFORD UNIVERSITY BANGLADESH


*20 May 2024*

# Experiment - 1

## Name of the Experiment: Display of Seven Segment.

### Theory:

A seven-segment display (SSD) is a widely used electronic visual display that presents decimal numerals through the illumination of seven segments. Each segment is composed of light-emitting diodes (LEDs) or other display technologies arranged in a specific pattern. The arrangement forms a rectangular structure with two vertical segments on each side and one horizontal segment on the top, middle, and bottom. Additionally, a seventh segment bisects the rectangle horizontally, providing a distinctive shape for each numeral.

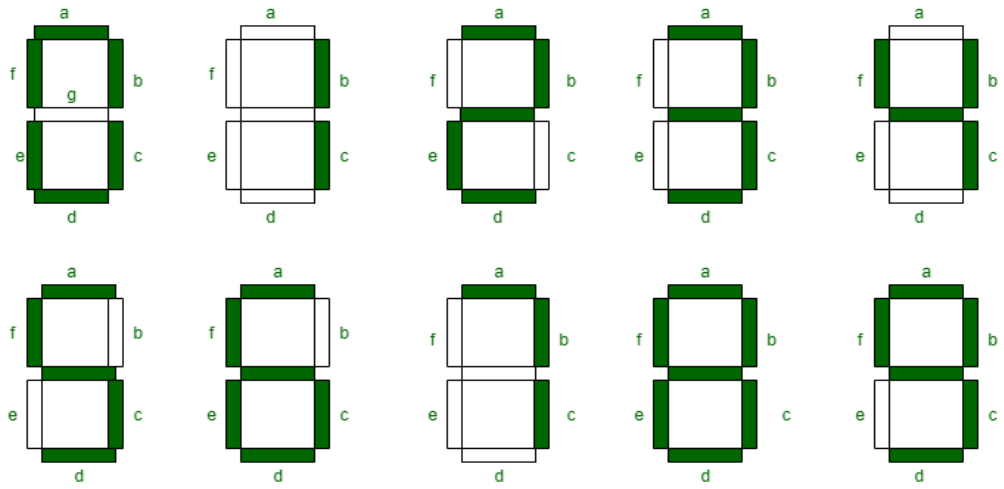


*Figure 1: 7 segment display*

### Hexadecimal encoding to display the digits 0 to 9:

| Digits | Hex. Value | H | G | F | E | D | C | B | A |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0C0h | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0F9h | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 2 | 0A4h | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0B0h | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 4 | 099h | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 5 | 092h | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 6 | 082h | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 7 | 0F8h | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 8 | 080h | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 090h | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

## Requirements:

1. Windows pc
2. Wincom Software
3. Masm Software
4. Text Editor
5. MDA 8086

## Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: A
ORG 1000H

S:
 MOV AL,80H
 OUT 1FH,AL

 MOV AL,082H
 OUT 19H,AL

A ENDS
END S
```

## DOS Command:

The following commands are given in the command prompt in order to connect the code segment with the WINCOM.

- ✓ Cd \
- ✓ Cd MDA
- ✓ Cd 8086
- ✓ Cd ASM8086
- ✓ MASM
- ✓ File_name
- ✓ File_name
- ✓ File_name

- ✓ LOD186
- ✓ File_name
- ✓ File_name

Now execute the WINCOM software and access the tool kit to initiate a RESET. Upon pressing the RESET button, the PC screen displays machine-generated information. Subsequently, issue the following commands:

- ✓ L
- ✓ F3
- ✓ File_name
- ✓ G

Following the execution of the commands, the desired output becomes visible on the tool kit, indicating successful processing or attainment of the intended result.

**Discussion:**

This experiment focused on utilizing seven-segment displays to exhibit decimal numbers and successfully achieved this goal. By maintaining a straightforward file placement strategy and avoiding any complications, this experience was seamless. It executed the WINCOM software effortlessly after providing the necessary DOS commands, resulting in the desired output without encountering any notable issues.

# Experiment - 2

Name of the Experiment: Turn on the LED.

## Theory:

This experiment aims to turn on the LED using Segment B (1BH) in a MDA 8086 microcontroller. The objective is to illuminate four LEDs with distinct colors: Red (PB 0), Green (PB 1), Yellow (PB 2), and Blue (PB 3). The perpetual loop is designed to continuously control the LEDs, maintaining the desired lighting configuration.

## Hexadecimal encoding to display the digits 0 to 9:

| LED | Hexadecimal Value | R | Y | G | R |
|---|---|---|---|---|---|
| Red | 01H | 0 | 0 | 0 | 1 |
| Green | 02H | 0 | 0 | 1 | 0 |
| Yellow | 04H | 0 | 1 | 0 | 0 |
| red | 08H | 1 | 0 | 0 | 0 |

## Requirements:

1. Windows pc
2. Wincom Software
3. Masm Software
4. Text Editor
5. MDA 8086

## Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS:A
ORG 1000H

S:
    MOV AL,80H
    OUT 1FH,AL
    MOV AL,0FFH ; Disable 7-Segment Display
    OUT 19H,AL
```

```
L:
    MOV AL,01H ; LED
    OUT 1BH,AL
    JMP L
A ENDS
END S
```

## DOS Command:

The following commands are given in the command prompt in order to connect the code segment with the WINCOM.

- ✓ Cd \
- ✓ Cd MDA
- ✓ Cd 8086
- ✓ Cd ASM8086
- ✓ MASM
- ✓ File_name
- ✓ File_name
- ✓ File_name
- ✓ LOD186
- ✓ File_name
- ✓ File_name

Now execute the WINCOM software and access the tool kit to initiate a RESET. Upon pressing the RESET button, the PC screen displays machine-generated information. Subsequently, issue the following commands:

- ✓ L
- ✓ F3
- ✓ File_name
- ✓ G

Following the execution of the commands, the desired output becomes visible on the tool kit, indicating successful processing or attainment of the intended result.

## Discussion:

The primary objective of the experiment was to illuminate an LED through precise control of input/output ports. Port 1FH is likely instrumental in configuring the LED display, while port 19H receives a command (0FFH) to disable the 7-segment

display, hinting at a potential connection between the LED and this display. The crux of the code lies in an infinite loop labeled 'L,' where the constant output of the value 01H to port 1BH suggests a command to turn on the LED, resulting in a continuous state of illumination. The persistent loop, facilitated by the JMP instruction, implies an enduring state of LED activation. However, a comprehensive understanding of the experiment's success and implications necessitates a deeper insight into the specific hardware configuration, the intricate interplay between the assembly code and the LED display, and the broader operational context in which this code is intended to function. In the absence of such details, the precise nature and significance of the experiment remain subject to interpretation.

# Experiment - 3

Name of the Experiment: Running the LED clock wise.

## Theory:

The experiment focuses on controlling LEDs using Segment B (1BH) in an MDA 8086 microcontroller. The goal is to illuminate four LEDs with distinct colors: Red (PB 0), Green (PB 1), Yellow (PB 2), and Blue (PB 3). The code employs a reverse logic approach, likely activating specific bits in the PB register to turn on the corresponding LEDs. The perpetual loop 'L' ensures continuous LED control, maintaining the desired lighting configuration.

## Hexadecimal encoding to display the digits 0 to 9:

| LED | Hexadecimal Value | R | Y | G | R |
|---|---|---|---|---|---|
| Red | 01H | 0 | 0 | 0 | 1 |
| Green | 02H | 0 | 0 | 1 | 0 |
| Yellow | 04H | 0 | 1 | 0 | 0 |
| red | 08H | 1 | 0 | 0 | 0 |

## Requirements:

1. Windows pc
2. Wincom Software
3. Masm Software
4. Text Editor
5. MDA 8086

## Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: A
ORG 1000H
S:
    MOV AL,80H
    OUT 1FH,AL
    MOV AL,0FFH  ;Disable 7 Segment Display [port A]
    OUT 19H,AL
L:
    MOV AL,01H  ;Turn on Red LED
```

```
    OUT 1BH,AL
    MOV CX,0FFFFH   ;Delay for 1 second
    L1: LOOP L1
    MOV AL,00H  ;Turn LED
    OUT 1BH,AL

    MOV AL,02H  ;Turn on Green LED
    OUT 1BH,AL
    MOV CX,0FFFFH   ;Delay for 1 second
    L2: LOOP L2
    MOV AL,00H  ;Turn off LED
    OUT 1BH,AL

    MOV AL,08H  ;Turn on Red LED
    OUT 1BH,AL
    MOV CX,0FFFFH   ;Delay for 1 second
    L3: LOOP L3
    MOV AL,00H  ;Turn off LED
    OUT 1BH,AL

    MOV AL,04H  ;Turn on Yellow LED
    OUT 1BH,AL
    MOV CX,0FFFFH   ;Delay for 1 second
    L4: LOOP L4
    MOV AL,00H  ;Turn off LED
    OUT 1BH,AL

    JMP L
A ENDS
END S
```

## DOS Command:

The DOS command remains the same as before.

## Discussion:

The experiment aimed to activate LEDs in an anti-clockwise manner using an LED display. Its goal was to emphasize port utilization and the importance of understanding hardware for proper port selection (1FH for the 7-segment display and 1BH for LEDs).

# Experiment - 4

Name of the Experiment: Running the LED anti-clock wise.

## Theory:

This experiment focuses on controlling LEDs using Segment B (1BH) in an MDA 8086 microcontroller. The goal is to illuminate four LEDs with distinct colors: Red (PB 0), Green (PB 1), Yellow (PB 2), and Blue (PB 3). The code likely employs a reverse logic approach, activating specific bits in the PB register to illuminate the corresponding LEDs.

## Hexadecimal encoding to display the digits 0 to 9:

| LED | Hexadecimal Value | R | Y | G | R |
|---|---|---|---|---|---|
| Red | 01H | 0 | 0 | 0 | 1 |
| Green | 02H | 0 | 0 | 1 | 0 |
| Yellow | 04H | 0 | 1 | 0 | 0 |
| red | 08H | 1 | 0 | 0 | 0 |

## Requirements:

1. Windows pc
2. Wincom Software
3. Masm Software
4. Text Editor
5. MDA 8086

## Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: A
ORG 1000H
S:
    MOV AL,80H
    OUT 1FH,AL
    MOV AL,0FFH  ;Disable 7 Segment Display [port A]
    OUT 19H,AL

L:
    MOV AL,04H  ;Turn on Yellow LED
```

```
    OUT 1BH,AL
    MOV CX,0FFFFH   ;Delay for 1 second
    L1: LOOP L1
    MOV AL,00H  ;Turn off LED
    OUT 1BH,AL

    MOV AL,08H  ;Turn on Red LED
    OUT 1BH,AL
    MOV CX,0FFFFH   ;Delay for 1 second
    L2: LOOP L2
    MOV AL,00H  ;Turn off LED
    OUT 1BH,AL

    MOV AL,02H  ;Turn on Green LED
    OUT 1BH,AL
    MOV CX,0FFFFH   ;Delay for 1 second
    L3: LOOP L3
    MOV AL,00H  ;Turn off LED
    OUT 1BH,AL

    MOV AL,01H  ;Turn on Red LED
    OUT 1BH,AL
    MOV CX,0FFFFH   ;Delay for 1 second
    L4: LOOP L4
    MOV AL,00H  ;Turn off LED
    OUT 1BH,AL

    JMP L
A ENDS
END S
```

## DOS Command:

The DOS command remains the same as before.

## Discussion:

The experiment aimed to activate LEDs in an anti-clockwise manner using an LED display. Its goal was to emphasize port utilization and the importance of understanding hardware for proper port selection (1FH for the 7-segment display and 1BH for LEDs).

# Experiment - 5

## Name of the Experiment: Traffic Light

**Theory:**

Generally, the sequence of a traffic signal is the red light turning on first, followed by the yellow light, then the green light. Accordingly, the experiment's steps involve initially activating the red light and then displaying numbers 0-9 backwards on the seven-segment display. Subsequently, the yellow and green lights are activated, followed by the red light again.

**Hexadecimal encoding to display the digits 0 to 9:**

| LED | Hexadecimal Value | R | Y | G | R |
|---|---|---|---|---|---|
| Red | 01H | 0 | 0 | 0 | 1 |
| Green | 02H | 0 | 0 | 1 | 0 |
| Yellow | 04H | 0 | 1 | 0 | 0 |
| red | 08H | 1 | 0 | 0 | 0 |

**Hexadecimal encoding to display the digits 0 to 9:**

| Digits | Hex. Value | H | G | F | E | D | C | B | A |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0C0h | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0F9h | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 2 | 0A4h | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0B0h | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 4 | 099h | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 5 | 092h | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 6 | 082h | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 7 | 0F8h | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 8 | 080h | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 090h | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

**Requirements:**

1. Windows pc
2. Wincom Software
3. Masm Software
4. Text Editor

5. MDA 8086

## Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: A
ORG 1000H
S:
MOV AL, 80H ;set control register
OUT 1FH, AL
MOV AL, 0FFH ;stop port A
OUT 19H, AL

L:
MOV AL, 01H ;turn Red LED on
OUT 1BH, AL
MOV CX, 0FFFFH ;delay for 1 second

;display 9 on seven segment
L1: LOOP L1
MOV AL, 090H ;display 9 on seven segment
OUT 19H, AL
MOV CX,0FFFFH ;delay for 1 second

LA: LOOP LA
MOV AL, 090H
OUT 19H, AL
MOV CX,0FFFFH

;display 8 on seven segment
L2: LOOP L2
MOV AL, 0FFH ;turn off segment
OUT 19H, AL
MOV AL, 080H ;display 8 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L2A: LOOP L2A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 080H
OUT 19H, AL
MOV CX, 0FFFFH

;display 7 on seven segment
L3: LOOP L3
```

```
MOV AL, 0FFH ;turn off segment
OUT 19H, AL
MOV AL, 0D8H ;display 7 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L3A: LOOP L3A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0D8H
OUT 19H, AL
MOV CX, 0FFFFH

;display 6 on seven segment
L4: LOOP L4
MOV AL, 0FFH ;turn off segment
OUT 19H, AL
MOV AL, 082H ;display 6 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L4A: LOOP L4A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 082H
OUT 19H, AL
MOV CX, 0FFFFH

;display 5 on seven segment
L5: LOOP L5
MOV AL, 0FFH ;turn off 7 segment
OUT 19H, AL
MOV AL, 092H ;display 5 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L5A: LOOP L5A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 092H
OUT 19H, AL
MOV CX, 0FFFFH

;display 4 on seven segment
L6: LOOP L6
MOV AL, 0FFH ;turn off 7 segment
OUT 19H, AL
MOV AL, 099H ;display 4 on seven segment
```

```
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L6A: LOOP L6A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 099H
OUT 19H, AL
MOV CX, 0FFFFH

;display 3 on seven segment
L7: LOOP L7
MOV AL, 0FFH ;turn off 7 segment
OUT 19H, AL
MOV AL, 0B0H ;display 3 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L7A: LOOP L7A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0B0H
OUT 19H, AL
MOV CX, 0FFFFH

;display 2 on seven segment
L8: LOOP L8
MOV AL, 0FFH ;turn off 7 segment
OUT 19H, AL
MOV AL, 0A4H ;display 2 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L8A: LOOP L8A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0A4H
OUT 19H, AL
MOV CX, 0FFFFH

;display 1 on seven segment
L9: LOOP L9
MOV AL, 0FFH ;turn off 7 segment
OUT 19H, AL
MOV AL, 0F9H ;display 1 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L9A: LOOP L9A
```

```
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0F9H
OUT 19H, AL
MOV CX, 0FFFFH

;display 0 on seven segment
L10: LOOP L10
MOV AL, 0FFH ;turn off 7 segment
OUT 19H, AL
MOV AL, 0C0H ;display 0 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L10A: LOOP L10A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0C0H
OUT 19H, AL
MOV CX, 0FFFFH

;turn on Red LED
L11: LOOP L11
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 00H ;turn off Red LED
OUT 1BH, AL
MOV CX, 0FFFFH ;delay for 1 second

;turn on Yellow LED
L12: LOOP L12
MOV AL, 04H
OUT 1BH, AL
L12A: LOOP L12A
MOV AL, 04H
OUT 1BH, AL
L12B: LOOP L12B
MOV AL, 04H
OUT 1BH, AL
L12C: LOOP L12C
MOV AL, 04H
OUT 1BH, AL
MOV CX, 0FFFFH ;delay for 1 second
;END

L13: LOOP L13
```

```
MOV AL, 00H ;turn off Yellow LED
OUT 1BH, AL
MOV CX, 0FFFFH

L14: LOOP L14
MOV AL, 02H ;turn on Green LED
OUT 1BH, AL
MOV CX, 0FFFFH

;display 9 on seven segment
L15: LOOP L15
MOV AL, 090H ;display 9 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L15A: LOOP L15A
MOV AL, 090H
OUT 19H, AL
MOV CX, 0FFFFH

;display 8 on seven segment
L16: LOOP L16
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
MOV AL, 080H ;display 8 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L16A: LOOP L16A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 080H
OUT 19H, AL
MOV CX, 0FFFFH

;display 7 on seven segment
L17: LOOP L17
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
MOV AL, 0D8H ;display 7 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L17A: LOOP L17A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0D8H
OUT 19H, AL
```

```
MOV CX, 0FFFFH

;display 6 on seven segment
L18: LOOP L18
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
MOV AL, 082H ;display 6 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L18A: LOOP L18A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 082H
OUT 19H, AL
MOV CX, 0FFFFH

;display 5 on seven segment
L19: LOOP L19
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
MOV AL, 092H ;display 5 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L19A: LOOP L19A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 092H
OUT 19H, AL
MOV CX, 0FFFFH

;display 4 on seven segment
L20: LOOP L20
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
MOV AL, 099H ;display 4 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L20A: LOOP L20A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 099H
OUT 19H, AL
MOV CX, 0FFFFH

;display 3 on seven segment
```

```
L21: LOOP L21
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
MOV AL, 0B0H ;display 3 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L21A: LOOP L21A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0B0H
OUT 19H, AL
MOV CX, 0FFFFH

;display 2 on seven segment
L22: LOOP L22
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
MOV AL, 0A4H ;display 2 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L22A: LOOP L22A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0A4H
OUT 19H, AL
MOV CX, 0FFFFH

;display 1 on seven segment
L23: LOOP L23
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
MOV AL, 0F9H ;display 1 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH
L23A: LOOP L23A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0F9H
OUT 19H, AL
MOV CX, 0FFFFH

;display 0 on seven segment
L24: LOOP L24
MOV AL, 0FFH ;turn off 7 segment display
OUT 19H, AL
```

```
MOV AL, 0C0H ;display 0 on seven segment
OUT 19H, AL
MOV CX, 0FFFFH ;delay for 1 second
L24A: LOOP L24A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0C0H
OUT 19H, AL
MOV CX, 0FFFFH

L45: LOOP L45
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 00H ;turn off Red LED
OUT 1BH, AL
MOV AL, 01H
OUT 1BH, AL
MOV CX, 0FFFFH ;delay for 1 second

L46: LOOP L46
JMP L ;go back to start
A ENDS
END S
```

## DOS Command:

The DOS command remains the same as before.

## Discussion:

The experiment's code was lengthy, requiring multiple runs and loading times, resulting in delayed achievement of the experiment's outcome. After several code checks, the goal was eventually attained. Then, the WINCOM software was executed by providing the necessary DOS commands, and the desired output was obtained.