VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

REPORT

LAB 2

Class: Microprocessors - Microcontrollers – CC04

Lecturer: PHAN VĂN SỸ

No.	Full Name	ID Student
1.	**Lê Mạnh Quân**	**2352993**

Ho Chi Minh City, September 2025

Contents

1 Exercise 1

1.1 Report 1

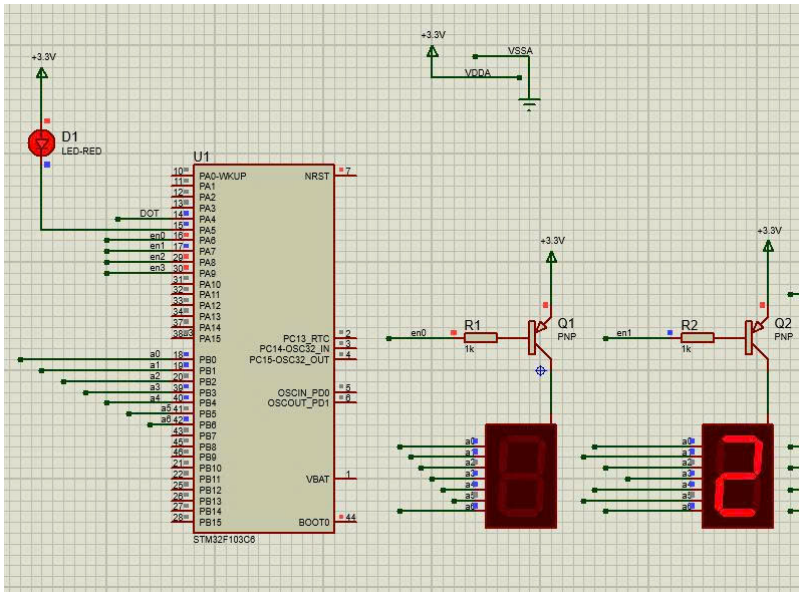

Figure 1: The schematic of exercise 1

1.2 Report 2

```
int counter = 50;  // 50*10ms = 0.5s
int state = 0;     // 0: LED0, 1: LED1

void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef *htim) {
        counter--;
        if (counter <= 0) {
            counter = 50;
            if (state == 0) {
                display7SEG(1);
                HAL_GPIO_WritePin(en0_GPIO_Port, en0_Pin,
                    GPIO_PIN_RESET);
                HAL_GPIO_WritePin(en1_GPIO_Port, en1_Pin,
                    GPIO_PIN_SET);
                state = 1;
            } else {
                display7SEG(2);
                HAL_GPIO_WritePin(en0_GPIO_Port, en0_Pin,
                    GPIO_PIN_SET);   \
                HAL_GPIO_WritePin(en1_GPIO_Port, en1_Pin,
                    GPIO_PIN_RESET);
                state = 0;
            }
        }
```

```
}
```

Listing 1: Source code in the HAL TIM PeriodElapsedCallback function

```
    void clear7SEG() {
    HAL_GPIO_WritePin(a0_GPIO_Port, a0_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(a1_GPIO_Port, a1_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(a2_GPIO_Port, a2_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(a3_GPIO_Port, a3_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(a4_GPIO_Port, a4_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(a5_GPIO_Port, a5_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(a6_GPIO_Port, a6_Pin, GPIO_PIN_SET);
}

void display7SEG (int num) {
    clear7SEG();
    switch (num) {
        case 0: HAL_GPIO_WritePin(a0_GPIO_Port,a0_Pin,
           GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a1_GPIO_Port,a1_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a3_GPIO_Port,a3_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a4_GPIO_Port,a4_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a5_GPIO_Port,a5_Pin,
                   GPIO_PIN_RESET);
                break;
        case 1: HAL_GPIO_WritePin(a1_GPIO_Port,a1_Pin,
           GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                   GPIO_PIN_RESET);
                break;
        case 2: HAL_GPIO_WritePin(a0_GPIO_Port,a0_Pin,
           GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a1_GPIO_Port,a1_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a3_GPIO_Port,a3_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a4_GPIO_Port,a4_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a6_GPIO_Port,a6_Pin,
                   GPIO_PIN_RESET);
                break;
        case 3: HAL_GPIO_WritePin(a0_GPIO_Port,a0_Pin,
           GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a1_GPIO_Port,a1_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a3_GPIO_Port,a3_Pin,
                   GPIO_PIN_RESET);
                HAL_GPIO_WritePin(a6_GPIO_Port,a6_Pin,
                   GPIO_PIN_RESET);
```

```
35                  break;
36          case 4: HAL_GPIO_WritePin(a1_GPIO_Port,a1_Pin,
                GPIO_PIN_RESET);
37                  HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                        GPIO_PIN_RESET);
38                  HAL_GPIO_WritePin(a5_GPIO_Port,a5_Pin,
                        GPIO_PIN_RESET);
39                  HAL_GPIO_WritePin(a6_GPIO_Port,a6_Pin,
                        GPIO_PIN_RESET);
40                  break;
41          case 5: HAL_GPIO_WritePin(a0_GPIO_Port,a0_Pin,
                GPIO_PIN_RESET);
42                  HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                        GPIO_PIN_RESET);
43                  HAL_GPIO_WritePin(a3_GPIO_Port,a3_Pin,
                        GPIO_PIN_RESET);
44                  HAL_GPIO_WritePin(a5_GPIO_Port,a5_Pin,
                        GPIO_PIN_RESET);
45                  HAL_GPIO_WritePin(a6_GPIO_Port,a6_Pin,
                        GPIO_PIN_RESET);
46                  break;
47          case 6: HAL_GPIO_WritePin(a0_GPIO_Port,a0_Pin,
                GPIO_PIN_RESET);
48                  HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                        GPIO_PIN_RESET);
49                  HAL_GPIO_WritePin(a3_GPIO_Port,a3_Pin,
                        GPIO_PIN_RESET);
50                  HAL_GPIO_WritePin(a4_GPIO_Port,a4_Pin,
                        GPIO_PIN_RESET);
51                  HAL_GPIO_WritePin(a5_GPIO_Port,a5_Pin,
                        GPIO_PIN_RESET);
52                  HAL_GPIO_WritePin(a6_GPIO_Port,a6_Pin,
                        GPIO_PIN_RESET);
53                  break;
54          case 7: HAL_GPIO_WritePin(a0_GPIO_Port,a0_Pin,
                GPIO_PIN_RESET);
55                  HAL_GPIO_WritePin(a1_GPIO_Port,a1_Pin,
                        GPIO_PIN_RESET);
56                  HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                        GPIO_PIN_RESET);
57                  break;
58          case 8: HAL_GPIO_WritePin(a0_GPIO_Port,a0_Pin,
                GPIO_PIN_RESET);
59                  HAL_GPIO_WritePin(a1_GPIO_Port,a1_Pin,
                        GPIO_PIN_RESET);
60                  HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                        GPIO_PIN_RESET);
61                  HAL_GPIO_WritePin(a3_GPIO_Port,a3_Pin,
                        GPIO_PIN_RESET);
62                  HAL_GPIO_WritePin(a4_GPIO_Port,a4_Pin,
                        GPIO_PIN_RESET);
63                  HAL_GPIO_WritePin(a5_GPIO_Port,a5_Pin,
                        GPIO_PIN_RESET);
64                  HAL_GPIO_WritePin(a6_GPIO_Port,a6_Pin,
                        GPIO_PIN_RESET);
65                  break;
```

```
66          case 9: HAL_GPIO_WritePin(a0_GPIO_Port,a0_Pin,
                GPIO_PIN_RESET);
67                  HAL_GPIO_WritePin(a1_GPIO_Port,a1_Pin,
                        GPIO_PIN_RESET);
68                  HAL_GPIO_WritePin(a2_GPIO_Port,a2_Pin,
                        GPIO_PIN_RESET);
69                  HAL_GPIO_WritePin(a3_GPIO_Port,a3_Pin,
                        GPIO_PIN_RESET);
70                  HAL_GPIO_WritePin(a5_GPIO_Port,a5_Pin,
                        GPIO_PIN_RESET);
71                  HAL_GPIO_WritePin(a6_GPIO_Port,a6_Pin,
                        GPIO_PIN_RESET);
72                  break;
73          default: clear7SEG(); break;
74      }
75  }
76  void initState() {
77      clear7SEG();
78      HAL_GPIO_WritePin(en0_GPIO_Port, en0_Pin, GPIO_PIN_SET);
79      HAL_GPIO_WritePin(en1_GPIO_Port, en1_Pin, GPIO_PIN_SET);
80      HAL_GPIO_WritePin(en2_GPIO_Port, en2_Pin, GPIO_PIN_SET);
81      HAL_GPIO_WritePin(en3_GPIO_Port, en3_Pin, GPIO_PIN_SET);
82      HAL_GPIO_WritePin(DOT_GPIO_Port, DOT_Pin, GPIO_PIN_RESET);
83      HAL_GPIO_WritePin(led_GPIO_Port, led_Pin, GPIO_PIN_RESET);
84  }
85
86  void clearEN() {
87      HAL_GPIO_WritePin(en0_GPIO_Port, en0_Pin, GPIO_PIN_SET);
88      HAL_GPIO_WritePin(en1_GPIO_Port, en1_Pin, GPIO_PIN_SET);
89      HAL_GPIO_WritePin(en2_GPIO_Port, en2_Pin, GPIO_PIN_SET);
90      HAL_GPIO_WritePin(en3_GPIO_Port, en3_Pin, GPIO_PIN_SET);
91  }
```

Listing 2: Function code in exercise1

1.3 Short question

Short Answer: The frequency of the scanning process is 1Hz

2 Exercise 2

2.1 Report 1

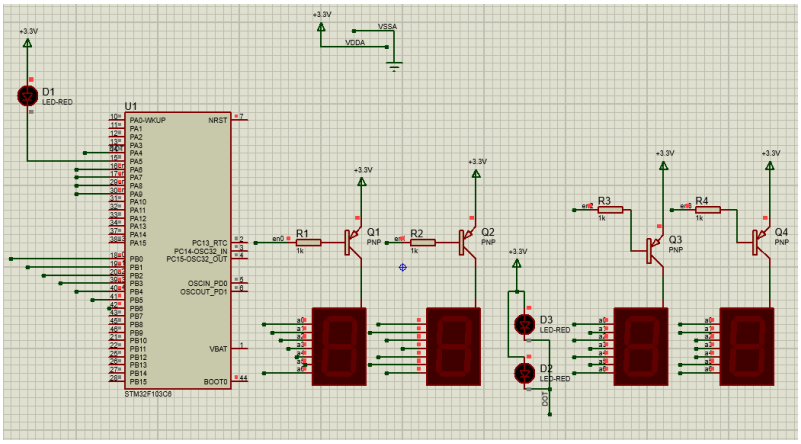

Figure 2: The schematic of exercise 2

2.2 Report 2

```
int counter = 100;
int idx = 0;
int buffer[4] = {1, 2, 3, 0};
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
        counter--;

        if (counter == 50 || counter == 0) {
            if (counter == 0) {

                counter = 100;
                HAL_GPIO_TogglePin(GPIOA, led_Pin | DOT_Pin); // nh y LED
            }
            clearEN();
            display7SEG(buffer[idx]);
            switch (idx) {
                case 0: HAL_GPIO_WritePin(en0_GPIO_Port, en0_Pin, GPIO_PIN_RESET); break;
                case 1: HAL_GPIO_WritePin(en1_GPIO_Port, en1_Pin, GPIO_PIN_RESET); break;
                case 2: HAL_GPIO_WritePin(en2_GPIO_Port, en2_Pin, GPIO_PIN_RESET); break;
                case 3: HAL_GPIO_WritePin(en3_GPIO_Port, en3_Pin, GPIO_PIN_RESET); break;
            }
            idx = (idx + 1) % 4;
        }

```

```
24 }
```

Listing 3: Source code in the HAL TIM PeriodElapsedCallback function

2.3 Short question

Short Answer: The frequency of the scanning process is 2Hz

3 Exercise 3

3.1 Report 1

```
const int MAX_LED = 4;
int counter = 100;
int index_led = 0;
void update7SEG(int index) {
        int led_buffer[4] = {1, 2, 3, 4};
        clearEN();
    switch (index) {
        case 0:
            display7SEG(led_buffer[0]);
            HAL_GPIO_WritePin(en0_GPIO_Port, en0_Pin,
               GPIO_PIN_RESET);
            break;
        case 1:
            display7SEG(led_buffer[1]);
            HAL_GPIO_WritePin(en1_GPIO_Port, en1_Pin,
               GPIO_PIN_RESET);
            break;
        case 2:
            display7SEG(led_buffer[2]);
            HAL_GPIO_WritePin(en2_GPIO_Port, en2_Pin,
               GPIO_PIN_RESET);
            break;
        case 3:
            display7SEG(led_buffer[3]);
            HAL_GPIO_WritePin(en3_GPIO_Port, en3_Pin,
               GPIO_PIN_RESET);
            break;
        default:
            break;
    }
}
```

Listing 4: Source code in update7SEG(int index) function

3.2 Report 2

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
        counter--;
        if (counter == 50 || counter == 0) {
            if (counter == 0) {
                counter = 100;
                HAL_GPIO_TogglePin(GPIOA, led_Pin | DOT_Pin);
            }
            update7SEG(index_led);
            index_led = (index_led + 1) % MAX_LED;
        }
}
```

Listing 5: Source code in the $\mathrm{HAL}_{TIMPeriodElapsedCallback}function$

4 Exercise 4

4.1 Report 1

```
int counter = 50;
int idx = 0;
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
        counter--;
        if (counter == 25 || counter == 0) {
            if (counter == 0) {
                counter = 50; // Reset sau 0.5s
                HAL_GPIO_TogglePin(GPIOA, led_Pin | DOT_Pin);
            }
            update7SEG(idx);
            idx = (idx + 1) % 4;
        }
}
```

Listing 6: source code in the $\mathrm{HAL}_{TIM_{P}eriodElapsedCallback}$

5 Exercise 5

5.1 Report

```
void updateClockBuffer(int hour, int minute) {
        led_buffer[0] = hour / 10;
        led_buffer[1] = hour % 10;
        led_buffer[2] = minute / 10;
        led_buffer[3] = minute % 10;
}
```

Listing 7: Source code in the updateClockBuffer function

6 Exercise 6

6.1 Report 1

If line 1 of the code is missed, what happens after that and why?
If line 1 of the code is missed, the value timer0flag is kept at 0 and cannot be set to 1,so the LED will not blink.

6.2 Report 2

If line 1 of the code is changed to setTimer0(1), what happens after that and why?
If line 1 of the code is changed to setTimer0(1), the LED will not blink, because if duration=1,we get timer0counter=0(since timer0counter is of type int), then when executing timerrun(), the value of timer0counter cannot satisfy the if condition,thus timer0flag is kept at 0 and cannot be set to 1.

6.3 Report 3

If in line 1 of the code above is changed to setTimer0(10),what is changed compared to 2 first questions and why?
If line 1 of the code is changed to setTimer0(10),we get timer0counter=1, this value satisfy the if condition in timerrun() and the timer0flag is set to 1 right away, so the LED will be invoked and start blinking properly.

7 Exercise 7

7.1 Report

```
int hour = 15, minute = 8, second = 55;
   setTimer0(1000);
   while (1)
    {
```

```
5
6     /* USER CODE END WHILE */
7
8     /* USER CODE BEGIN 3 */
9           if (timer0_flag == 1) {
10                                      second++;
11                                          if (second >= 60) {
12                                              second = 0;
13                                              minute++;
14                                          }
15                                          if (minute >= 60) {
16                                              minute = 0;
17                                              hour++;
18                                          }
19                                          if (hour >= 24) {
20                                              hour = 0;
21                                          }
22
23                                          updateClockBuffer(hour,
                                               minute);
24                                          setTimer0(1000);
25          }
26
27    }
```

Listing 8: Source code in the while loop on main function

8 Exercise 8

8.1 Report 1

```
1     int hour = 15, minute = 8, second = 55;
2     int index_led = 0;
3     const int MAX_LED = 4;
4     int led_buffer[4] = {1, 5, 0, 8};
5     int dot_state = 0;
6     setTimer0(1000);
7     setTimer1(250);
8     while (1)
9      {
10    if (timer0_flag == 1) {
11                                      second++;
12                                          if (second >= 60) {
13                                              second = 0;
14                                              minute++;
15                                          }
16                                          if (minute >= 60) {
17                                              minute = 0;
18                                              hour++;
19                                          }
20                                          if (hour >= 24) {
21                                              hour = 0;
22                                          }
```

```
23
24                                          updateClockBuffer(hour,
                                               minute);
25                                          dot_state = !dot_state;
26                                          HAL_GPIO_WritePin(GPIOA,
                                               GPIO_PIN_4, dot_state ?
                                               GPIO_PIN_SET :
                                               GPIO_PIN_RESET);
27                                          setTimer0(1000);
28              }
29              if (timer1_flag == 1) {
30                          update7SEG(index_led);
31                          index_led = (index_led + 1) % MAX_LED;
32                          setTimer1(250);
33                      }
34      }
35
36  }
37  }
```

Listing 9: Source code in the the main function

9 Exercise 9

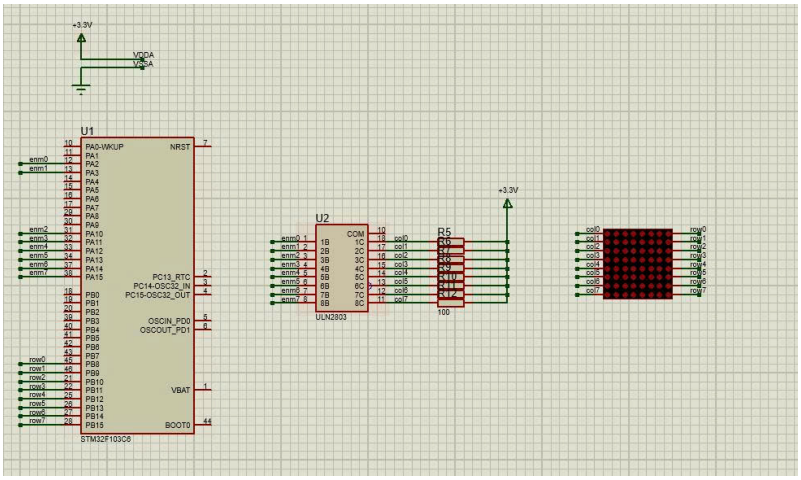

Figure 3: The schematic of exercise 1

```
const int MAX_LED_MATRIX = 8;
int index_matrix = 0;
uint8_t matrix_buffer[8] = {
    0x18,  // 00011000
    0x24,  // 00100100
    0x42,  // 01000010
    0x42,  // 01000010
    0x7E,  // 01111110
    0x42,  // 01000010
    0x42,  // 01000010
    0x00   // 00000000
};

void setRow(int row) {
    HAL_GPIO_WritePin(GPIOB, ROW0_Pin, (row == 0) ? GPIO_PIN_SET :
        GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, ROW1_Pin, (row == 1) ? GPIO_PIN_SET :
        GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, ROW2_Pin, (row == 2) ? GPIO_PIN_SET :
        GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, ROW3_Pin, (row == 3) ? GPIO_PIN_SET :
        GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, ROW4_Pin, (row == 4) ? GPIO_PIN_SET :
        GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, ROW5_Pin, (row == 5) ? GPIO_PIN_SET :
        GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, ROW6_Pin, (row == 6) ? GPIO_PIN_SET :
        GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, ROW7_Pin, (row == 7) ? GPIO_PIN_SET :
        GPIO_PIN_RESET);
}

void setColumn(int value) {
```

```
    HAL_GPIO_WritePin(GPIOA, ENM0_Pin, (value & 0x01) ?
        GPIO_PIN_SET : GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, ENM1_Pin, (value & 0x02) ?
        GPIO_PIN_SET : GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, ENM2_Pin, (value & 0x04) ?
        GPIO_PIN_SET : GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, ENM3_Pin, (value & 0x08) ?
        GPIO_PIN_SET : GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, ENM4_Pin, (value & 0x10) ?
        GPIO_PIN_SET : GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, ENM5_Pin, (value & 0x20) ?
        GPIO_PIN_SET : GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, ENM6_Pin, (value & 0x40) ?
        GPIO_PIN_SET : GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, ENM7_Pin, (value & 0x80) ?
        GPIO_PIN_SET : GPIO_PIN_RESET);
}

void updateLEDMatrix(int index) {
    setRow(index);
    setColumn(matrix_buffer[index]);
}

}
```

Listing 10: Code in private code

```
 while (1)
  {
                updateLEDMatrix(index_matrix);
                                index_matrix++;
                                HAL_Delay(1);
                                if (index_matrix >= 8) {
                                    index_matrix = 0;
                               }
}
```

Listing 11: Code in while loop

10 Exercise 10

```
void shiftLeft(uint8_t matrix_buffer[8]) {
    for (int i = 0; i < 8; i++) {
        uint8_t leftBit = (matrix_buffer[i] & 0x80) >> 7;
        matrix_buffer[i] = (matrix_buffer[i] << 1) | leftBit;
    }
}
```

Listing 12: Code in private code

```
 while (1)
  {
                        updateLEDMatrix(index_matrix);
                            index_matrix++;
```

```
5                                   HAL_Delay(5);
6                                   if (index_matrix >= 8) {
7                                   shiftLeft(matrix_buffer);
8                                 index_matrix = 0;
9                                                                       }
10 }
```

Listing 13: Code in while loop

11 Source

GG Drive Link: My Source Code Github Link: My Source Code