

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



**BÁO CÁO THỰC HÀNH**  
**VI XỬ LÝ - VI ĐIỀU KHIỂN**  
**LAB 1: LED ANIMATIONS**

Lớp – Nhóm:

L03 – L05

Giảng viên hướng dẫn:

Lê Trọng Nhân

Cao Tiến Đạt

Sinh viên thực hiện:

Ngô Quang Tùng

2213869

Thành phố Hồ Chí Minh, 10/2024

# Mục lục

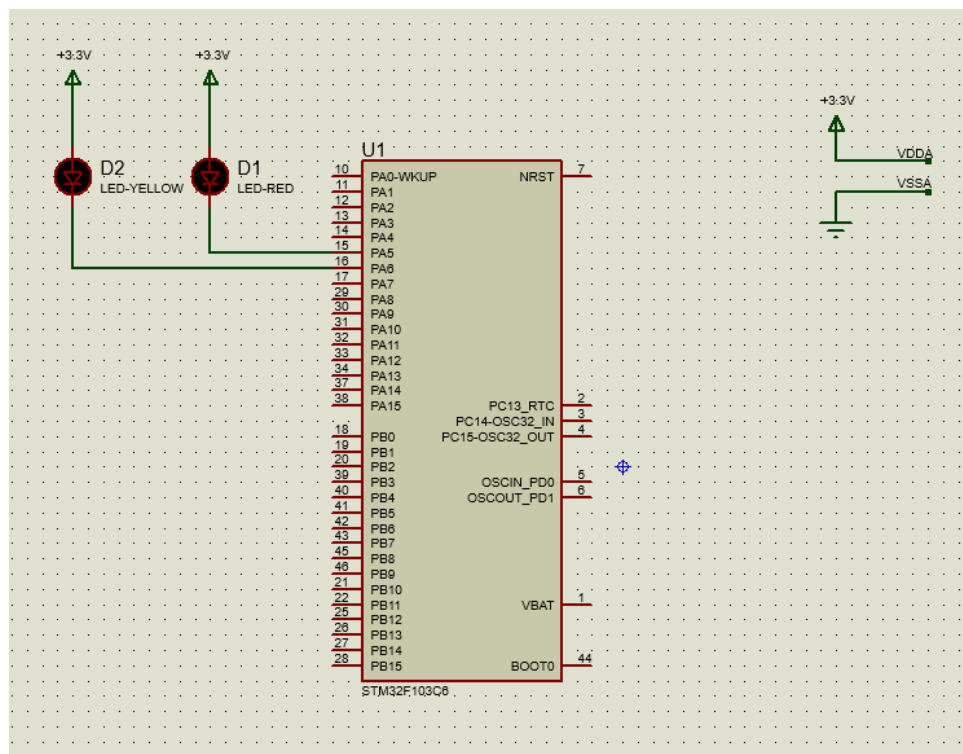
<b>1</b>	<b>Shortest Path Algorithm</b>	<b>2</b>
1.1	Bài 1 . . . . .	2
1.1.1	Sơ đồ nguyên lý . . . . .	2
1.1.2	Source code . . . . .	2
1.2	Bài 2 . . . . .	3
1.2.1	Sơ đồ nguyên lý . . . . .	3
1.2.2	Source code . . . . .	3
1.3	Bài 3 . . . . .	4
1.3.1	Sơ đồ nguyên lý . . . . .	4
1.3.2	Source code . . . . .	4
1.4	Bài 4 . . . . .	6
1.4.1	Sơ đồ nguyên lý . . . . .	6
1.4.2	Source code . . . . .	6
1.5	Bài 5 . . . . .	7
1.5.1	Sơ đồ nguyên lý . . . . .	7
1.5.2	Source code . . . . .	7
1.6	Bài 6 . . . . .	9
1.6.1	Source code . . . . .	9
1.7	Bài 7 . . . . .	9
1.7.1	Source code . . . . .	9
1.8	Bài 8 . . . . .	10
1.8.1	Source code . . . . .	10
1.9	Bài 9 . . . . .	10
1.9.1	Source code . . . . .	10
1.10	Bài 10 . . . . .	10
1.10.1	Source code . . . . .	10

# Chương 1

## Shortest Path Algorithm

### 1.1 Bài 1

#### 1.1.1 Sơ đồ nguyên lý



Hình 1.1: Sơ đồ nguyên lý

#### 1.1.2 Source code

```
1 while (1)
2 {
3     /* USER CODE END WHILE */
4     HAL_GPIO_TogglePin(LED_RED_GPIO_Port , LED_RED_Pin);
5     HAL_Delay(2000);
```

```

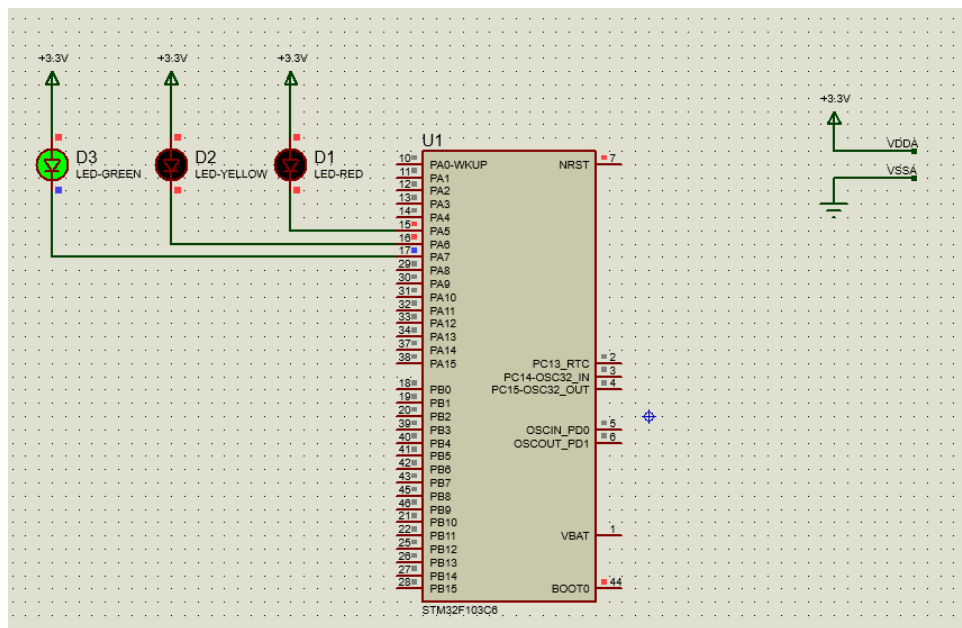
6     HAL_GPIO_TogglePin(LED_GREEN_GPIO_Port , LED_GREEN_Pin);
7
8     /* USER CODE BEGIN 3 */
9 }

```

Chương trình 1.1: Hàm chuyển đổi trạng thái cho 2 đèn LED

## 1.2 Bài 2

### 1.2.1 Sơ đồ nguyên lý



Hình 1.2: Sơ đồ nguyên lý

### 1.2.2 Source code

```

1  while (1)
2  {
3      /* USER CODE END WHILE */
4      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port , LED_GREEN_Pin ,
5                          GPIO_PIN_SET);
6      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port , LED_YELLOW_Pin ,
7                          GPIO_PIN_SET);
8      HAL_GPIO_WritePin(LED_RED_GPIO_Port , LED_RED_Pin ,
9                          GPIO_PIN_RESET);
10     HAL_Delay(5000);
11
12     HAL_GPIO_WritePin(LED_GREEN_GPIO_Port , LED_GREEN_Pin ,
13                         GPIO_PIN_RESET);

```

```

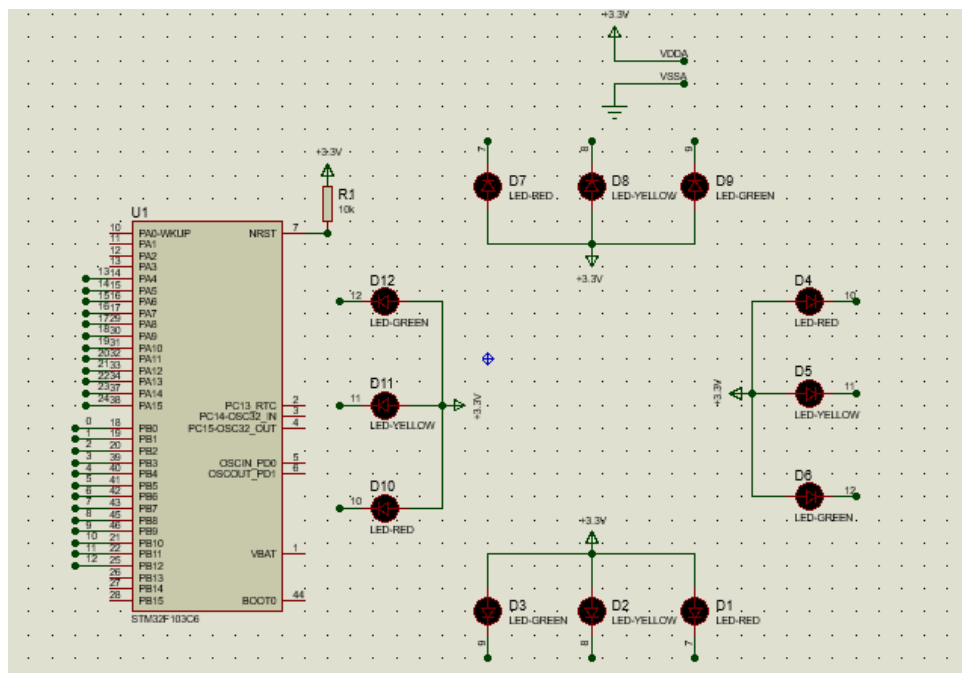
10     HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
        GPIO_PIN_SET);
11     HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
        GPIO_PIN_SET);
12     HAL_Delay(3000);
13
14     HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin,
        GPIO_PIN_SET);
15     HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
        GPIO_PIN_RESET);
16     HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
        GPIO_PIN_SET);
17     HAL_Delay(2000);
18     /* USER CODE BEGIN 3 */
19 }

```

Chương trình 2.1: Hàm chuyển đổi trạng thái cho 3 đèn LED

## 1.3 Bài 3

### 1.3.1 Sơ đồ nguyên lý



Hình 1.3: Sơ đồ nguyên lý

### 1.3.2 Source code

```

1     while (1)
2     {

```

```

3  /* USER CODE END WHILE */
4  HAL_GPIO_WritePin(LED_GREEN_GPIO_Port,LED_GREEN_Pin,
5  GPIO_PIN_RESET);
6  HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
7  GPIO_PIN_SET);
8  HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
9  GPIO_PIN_SET);
10 HAL_GPIO_WritePin(GPIO_PIN9_GPIO_Port, GPIO_PIN9_Pin,
11 GPIO_PIN_RESET);
12 HAL_GPIO_WritePin(GPIO_PIN10_GPIO_Port, GPIO_PIN10_Pin,
13 GPIO_PIN_SET);
14 HAL_GPIO_WritePin(GPIO_PIN11_GPIO_Port, GPIO_PIN11_Pin,
15 GPIO_PIN_SET);
16
17 for (countdown = 5; countdown > 0; countdown--)
18 {
19     if (countdown == 3)
20     {
21         HAL_GPIO_WritePin(LED_GREEN_GPIO_Port,
22         LED_GREEN_Pin, GPIO_PIN_SET);
23         HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
24         LED_YELLOW_Pin, GPIO_PIN_RESET);
25     }
26     HAL_Delay(1000);
27 }
28 HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
29 GPIO_PIN_SET);
30 HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
31 GPIO_PIN_RESET);
32 HAL_GPIO_WritePin(GPIO_PIN9_GPIO_Port, GPIO_PIN9_Pin,
33 GPIO_PIN_SET);
34 HAL_GPIO_WritePin(GPIO_PIN11_GPIO_Port, GPIO_PIN11_Pin,
35 GPIO_PIN_RESET);
36
37 for (countdown = 3; countdown > 0; countdown--)
38 {
39     HAL_Delay(1000);
40 }
41
42 HAL_GPIO_WritePin(GPIO_PIN11_GPIO_Port, GPIO_PIN11_Pin,
43 GPIO_PIN_SET);
44 HAL_GPIO_WritePin(GPIO_PIN10_GPIO_Port, GPIO_PIN10_Pin,
45 GPIO_PIN_RESET);
46
47 for (countdown = 2; countdown > 0; countdown--)
48 {
49     HAL_Delay(1000);
50 }

```

```

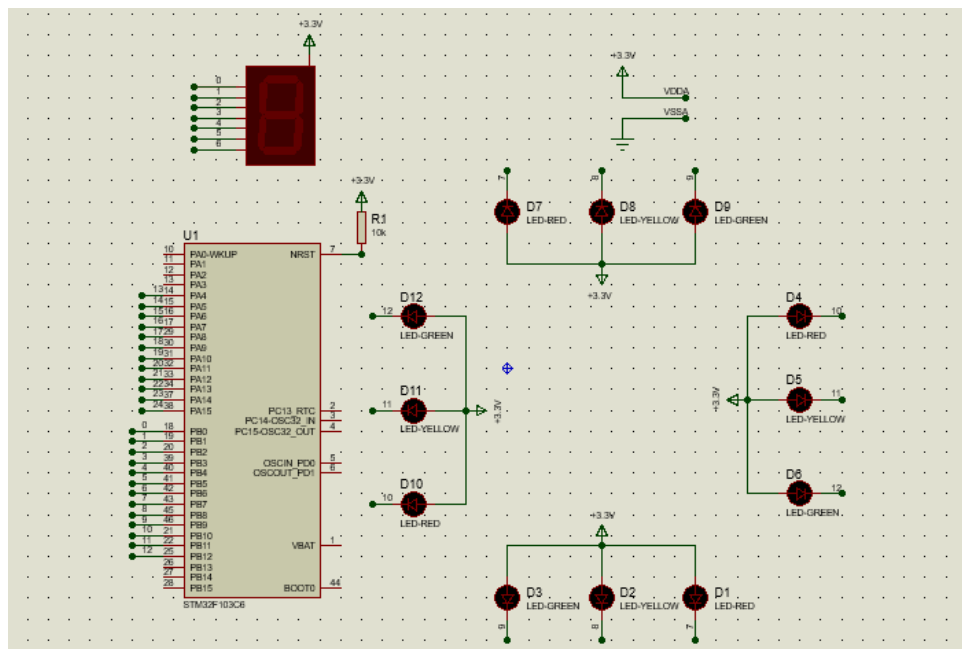
37     }
38
39     HAL_GPIO_WritePin(GPIO_PIN10_GPIO_Port, GPIO_PIN10_Pin,
40                       GPIO_PIN_SET);
41     /* USER CODE BEGIN 3 */
42 }

```

Chương trình 3.1: Hàm chuyển đổi đèn giao thông 4 chiều

## 1.4 Bài 4

### 1.4.1 Sơ đồ nguyên lý



Hình 1.4: Sơ đồ nguyên lý

### 1.4.2 Source code

```

1 void display7SEG(int num)
2 {
3     uint8_t segment_map[10] = {
4         0b11000000, // 0: a, b, c, d, e, f
5         0b11111001, // 1: b, c
6         0b10100100, // 2: a, b, d, e, g
7         0b10110000, // 3: a, b, c, d, g
8         0b10011001, // 4: b, c, f, g
9         0b10010010, // 5: a, c, d, f, g
10        0b10000010, // 6: a, c, d, e, f, g
11        0b11111000, // 7: a, b, c

```

```

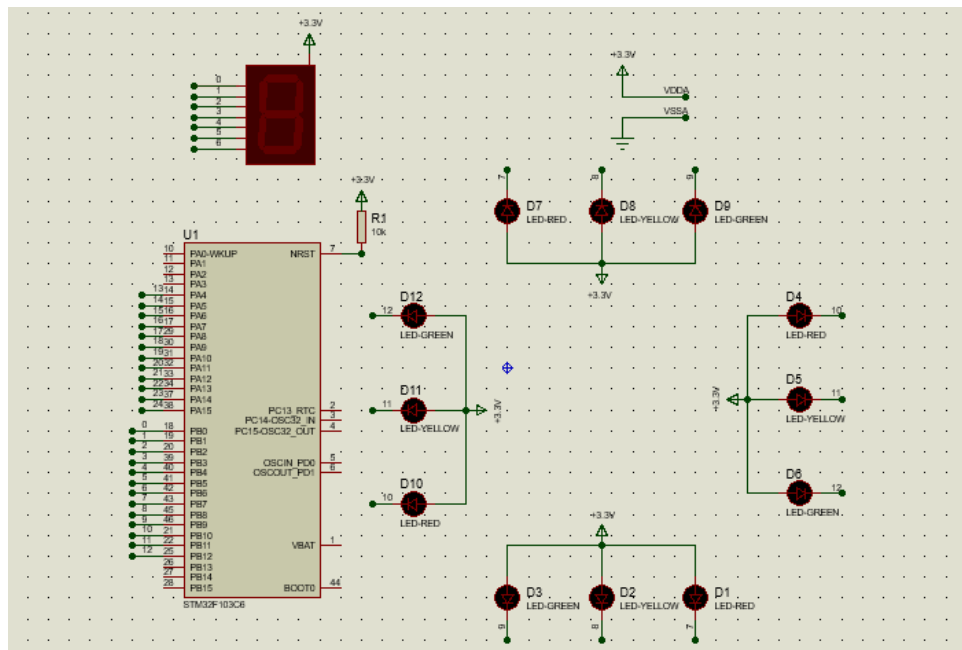
12         0b10000000, // 8: a, b, c, d, e, f, g
13         0b10010000 // 9: a, b, c, d, f, g
14     };

```

Chương trình 4.1: Hàm display7SEG

## 1.5 Bài 5

### 1.5.1 Sơ đồ nguyên lý



Hình 1.5: Sơ đồ nguyên lý

### 1.5.2 Source code

```

1 void display7SEG(int num);
2 int countdown;
3 while (1)
4 {
5     /* USER CODE END WHILE */
6     HAL_GPIO_WritePin(GPIO_PIN_12_GPIO_Port,
7         GPIO_PIN_12_Pin, GPIO_PIN_RESET);
8     HAL_GPIO_WritePin(GPIO_PIN_11_GPIO_Port,
9         GPIO_PIN_11_Pin, GPIO_PIN_SET);
10    HAL_GPIO_WritePin(GPIO_PIN_10_GPIO_Port,
11        GPIO_PIN_10_Pin, GPIO_PIN_SET);
12
13    HAL_GPIO_WritePin(GPIO_PIN_7_GPIO_Port, GPIO_PIN_7_Pin,
14        GPIO_PIN_RESET);

```



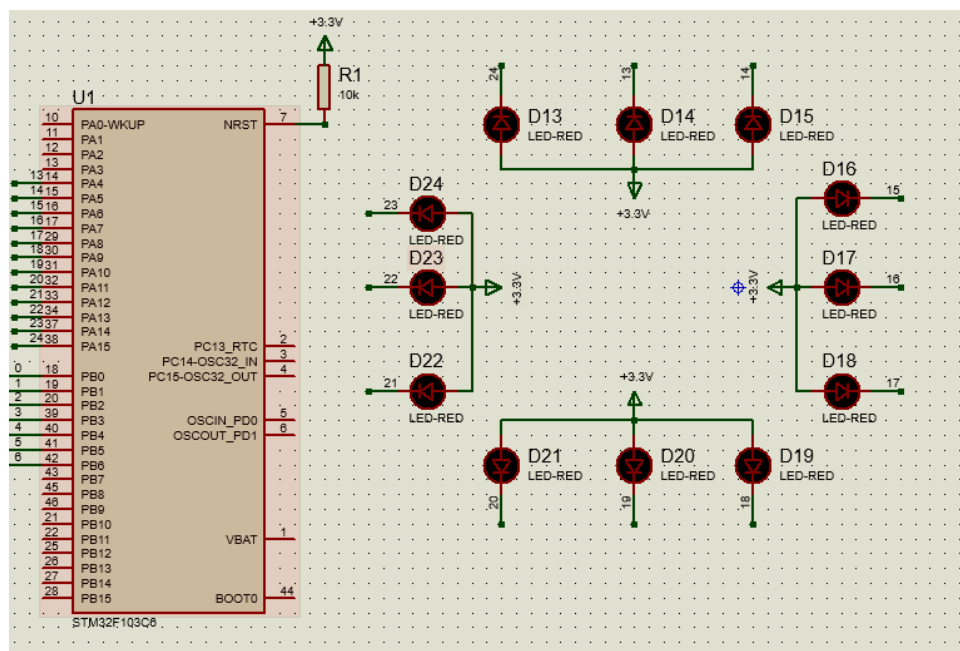
```

11     HAL_GPIO_WritePin(GPIO_PIN_8_GPIO_Port, GPIO_PIN_8_Pin,
12                        GPIO_PIN_SET);
13
14     HAL_GPIO_WritePin(GPIO_PIN_9_GPIO_Port, GPIO_PIN_9_Pin,
15                        GPIO_PIN_SET);
16
17     for (countdown = 5; countdown > 0; countdown--)
18     {
19         display7SEG(countdown);
20         if (countdown == 3)
21         {
22             HAL_GPIO_WritePin(GPIO_PIN_12_GPIO_Port,
23                                GPIO_PIN_12_Pin, GPIO_PIN_SET);
24             HAL_GPIO_WritePin(GPIO_PIN_11_GPIO_Port,
25                                GPIO_PIN_11_Pin, GPIO_PIN_RESET);
26         }
27         HAL_Delay(1000);
28     }
29
30     HAL_GPIO_WritePin(GPIO_PIN_11_GPIO_Port,
31                        GPIO_PIN_11_Pin, GPIO_PIN_SET);
32     HAL_GPIO_WritePin(GPIO_PIN_10_GPIO_Port,
33                        GPIO_PIN_10_Pin, GPIO_PIN_RESET);
34     HAL_GPIO_WritePin(GPIO_PIN_7_GPIO_Port, GPIO_PIN_7_Pin,
35                        GPIO_PIN_SET);
36     HAL_GPIO_WritePin(GPIO_PIN_9_GPIO_Port, GPIO_PIN_9_Pin,
37                        GPIO_PIN_RESET);
38
39     for (countdown = 3; countdown > 0; countdown--)
40     {
41         display7SEG(countdown);
42         HAL_Delay(1000);
43     }
44
45     HAL_GPIO_WritePin(GPIO_PIN_8_GPIO_Port, GPIO_PIN_8_Pin,
46                        GPIO_PIN_RESET);
47     HAL_GPIO_WritePin(GPIO_PIN_9_GPIO_Port, GPIO_PIN_9_Pin,
48                        GPIO_PIN_SET);
49
50     for (countdown = 2; countdown > 0; countdown--)
51     {
52         display7SEG(countdown);
53         HAL_Delay(1000);
54     }
55
56     HAL_GPIO_WritePin(GPIO_PIN_8_GPIO_Port, GPIO_PIN_8_Pin,
57                        GPIO_PIN_SET);
58     /* USER CODE BEGIN 3 */
59 }

```

## 1.6 Bài 6

### 1.6.1 Source code



Hình 1.6: Sơ đồ nguyên lý

```

1  while(){
2      /* USER CODE END WHILE */
3      for (int i = 4; i <= 15; i++)
4      {
5          HAL_GPIO_WritePin(GPIOA, (1 << i), GPIO_PIN_RESET);
6          HAL_Delay(1000);
7          HAL_GPIO_WritePin(GPIOA, (1 << i), GPIO_PIN_SET);
8      }
9      /* USER CODE BEGIN 3 */
10 }

```

## 1.7 Bài 7

### 1.7.1 Source code

```

1  void clearAllClock(void)
2  {

```

```

3     HAL_GPIO_WritePin(GPIOA, GPIO_PIN4 | GPIO_PIN5 |
        GPIO_PIN6 | GPIO_PIN7 | GPIO_PIN8 | GPIO_PIN9 |
        GPIO_PIN10 | GPIO_PIN11 |
4     GPIO_PIN12 | GPIO_PIN13 | GPIO_PIN14 | GPIO_PIN15,
        GPIO_PIN_SET);
5 }

```

Chương trình 7.1: Hàm clearAllClock

## 1.8 Bài 8

### 1.8.1 Source code

```

1 void setNumberOnClock(int num)
2 {
3     if (num >= 0 && num < 12)
4     {
5         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4 << num,
            GPIO_PIN_RESET);
6     }
7 }

```

Chương trình 8.1: Hàm setNumberOnClock

## 1.9 Bài 9

### 1.9.1 Source code

```

1 void clearNumberOnClock(int num)
2 {
3     if (num >= 0 && num < 12)
4     {
5         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4 << num,
            GPIO_PIN_SET);
6     }
7 }

```

Chương trình 9.1: Hàm clearNumberOnClock

## 1.10 Bài 10

### 1.10.1 Source code

```

1  void setNumberOnClock(int num)
2  {
3      if (num >= 0 && num < 12)
4      {
5          HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4 << num,
6                          GPIO_PIN_RESET);
7      }
8  }

```

Chương trình 10.1: Hàm setNumberOnClock

```

1  void updateClockDisplay(int hour, int minute, int second)
2  {
3      clearAllClock();
4      setNumberOnClock(hour);
5      setNumberOnClock(minute);
6      setNumberOnClock(second);
7  }

```

Chương trình 10.2: Hàm updateClockDisplay

```

1  int hour = 0;
2  int minute = 0;
3  int second = 0;
4  while (1)
5  {
6      /* USER CODE END WHILE */
7      updateClockDisplay(hour, minute, second);
8      HAL_Delay(1000);
9      second++;
10     if (second >= 12)
11     {
12         second = 0;
13         minute++;
14         if (minute >= 12)
15         {
16             minute = 0;
17             hour++;
18             if (hour >= 12)
19             {
20                 hour = 0;
21             }
22         }
23     }
24
25     /* USER CODE BEGIN 3 */
26 }

```

### Chương trình 10.3: Hàm while trong main