

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



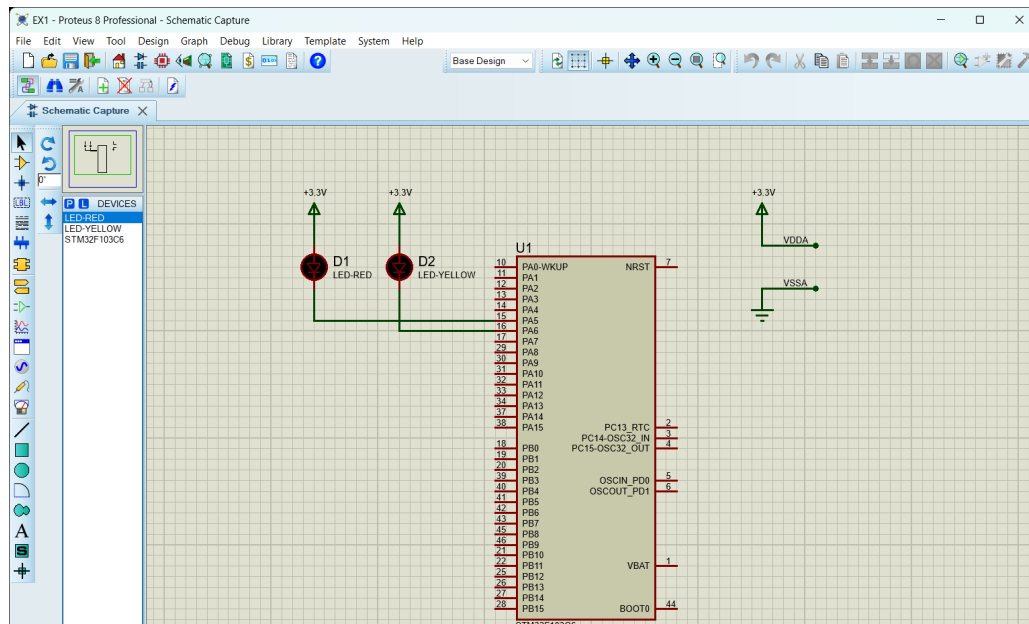
BÁO CÁO BÀI TẬP LỚN
MÔN: VI XỬ LÝ - VI ĐIỀU KHIỂN

LAB 1

Lớp:	L01
GVHD:	Phan Văn Sỹ
Họ và tên:	Phạm Đình Phong
MSSV:	2312628

1 Exercise 1

The schematic from Proteus simulation:



Code:

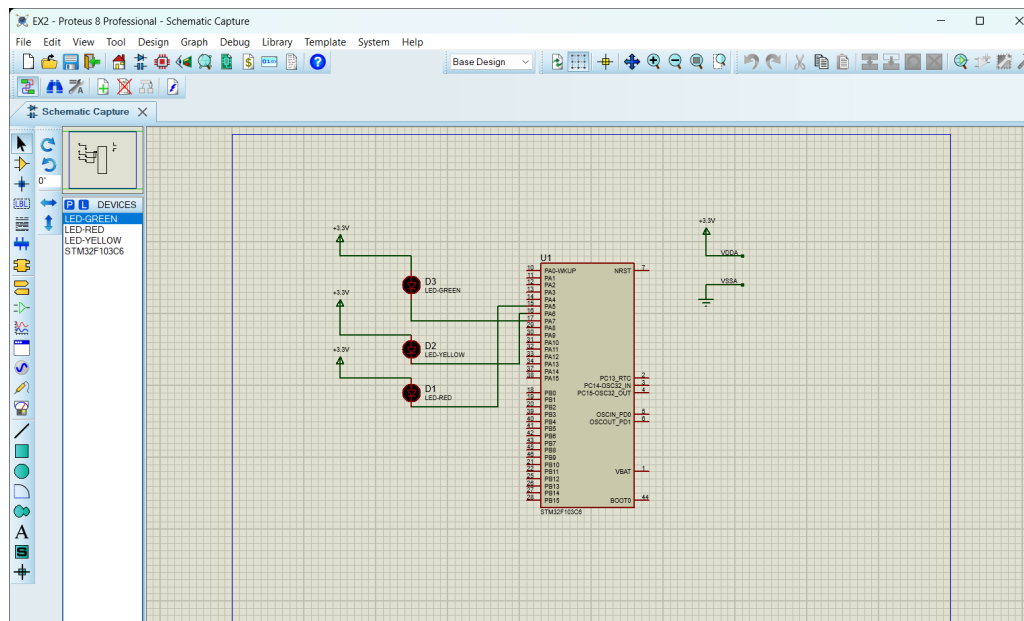
```
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    HAL_GPIO_WritePin ( LED_RED_GPIO_Port , LED_RED_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW_GPIO_Port , LED_YELLOW_Pin ,GPIO_PIN_RESET ) ;
    HAL_Delay (2000) ;
    HAL_GPIO_WritePin ( LED_RED_GPIO_Port , LED_RED_Pin ,GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW_GPIO_Port , LED_YELLOW_Pin , GPIO_PIN_SET ) ;
    HAL_Delay (2000) ;

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```

2 Exercise 2:

Report 1: Present the schematic:



Report 2: Present the source code in while:

```
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */
    HAL_GPIO_WritePin ( LED_GREEN_GPIO_Port , LED_GREEN_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_RED_GPIO_Port , LED_RED_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW_GPIO_Port , LED_YELLOW_Pin , GPIO_PIN_SET ) ;
    HAL_Delay(3000) ;
    HAL_GPIO_WritePin ( LED_GREEN_GPIO_Port , LED_GREEN_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED_GPIO_Port , LED_RED_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW_GPIO_Port , LED_YELLOW_Pin ,GPIO_PIN_RESET ) ;
    HAL_Delay (2000) ;
    HAL_GPIO_WritePin ( LED_GREEN_GPIO_Port , LED_GREEN_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED_GPIO_Port , LED_RED_Pin ,GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW_GPIO_Port , LED_YELLOW_Pin , GPIO_PIN_SET ) ;
    HAL_Delay (5000) ;
    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```

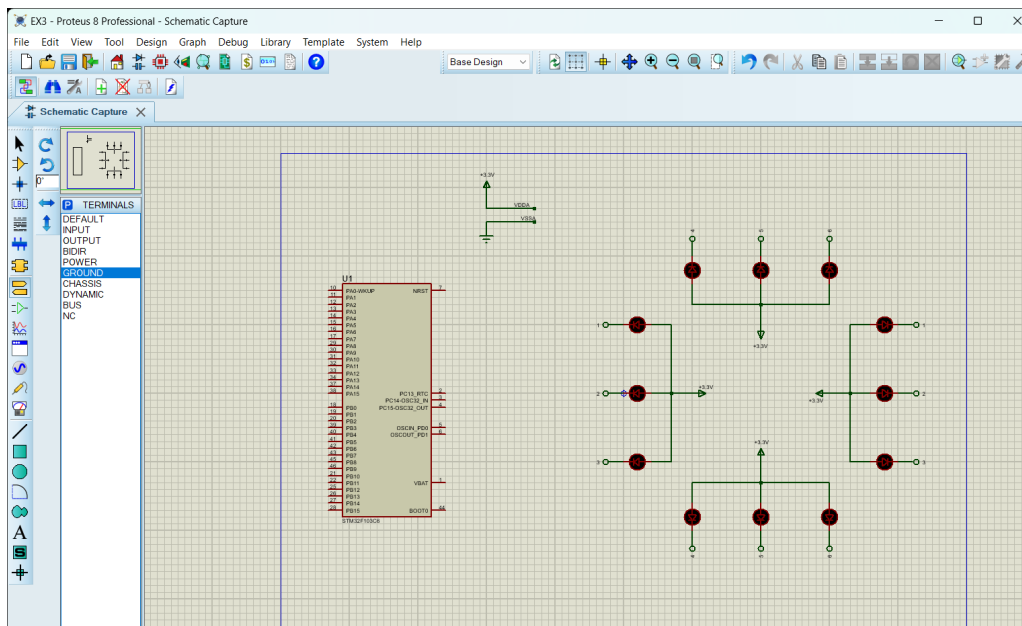
Link:

Source Code

Schematic

3 Exercise 3

Reference design for a 4 way traffic light:



Source code:

```
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    HAL_GPIO_WritePin ( LED_GREEN1_GPIO_Port , LED_GREEN1_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_RED1_GPIO_Port , LED_RED1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW1_GPIO_Port , LED_YELLOW1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_GREEN2_GPIO_Port , LED_GREEN2_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED2_GPIO_Port , LED_RED2_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW2_GPIO_Port , LED_YELLOW2_Pin , GPIO_PIN_SET ) ;
    HAL_Delay(3000) ;
    HAL_GPIO_WritePin ( LED_GREEN1_GPIO_Port , LED_GREEN1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED1_GPIO_Port , LED_RED1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW1_GPIO_Port , LED_YELLOW1_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_GREEN2_GPIO_Port , LED_GREEN2_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED2_GPIO_Port , LED_RED2_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW2_GPIO_Port , LED_YELLOW2_Pin , GPIO_PIN_SET ) ;
    HAL_Delay (2000) ;
    HAL_GPIO_WritePin ( LED_GREEN1_GPIO_Port , LED_GREEN1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED1_GPIO_Port , LED_RED1_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW1_GPIO_Port , LED_YELLOW1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_GREEN2_GPIO_Port , LED_GREEN2_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_RED2_GPIO_Port , LED_RED2_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW2_GPIO_Port , LED_YELLOW2_Pin , GPIO_PIN_SET ) ;
    HAL_Delay (3000) ;
    HAL_GPIO_WritePin ( LED_GREEN1_GPIO_Port , LED_GREEN1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED1_GPIO_Port , LED_RED1_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW1_GPIO_Port , LED_YELLOW1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_GREEN2_GPIO_Port , LED_GREEN2_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED2_GPIO_Port , LED_RED2_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW2_GPIO_Port , LED_YELLOW2_Pin , GPIO_PIN_RESET ) ;
    HAL_Delay (2000) ;
    /* USER CODE BEGIN 3 */
}
```

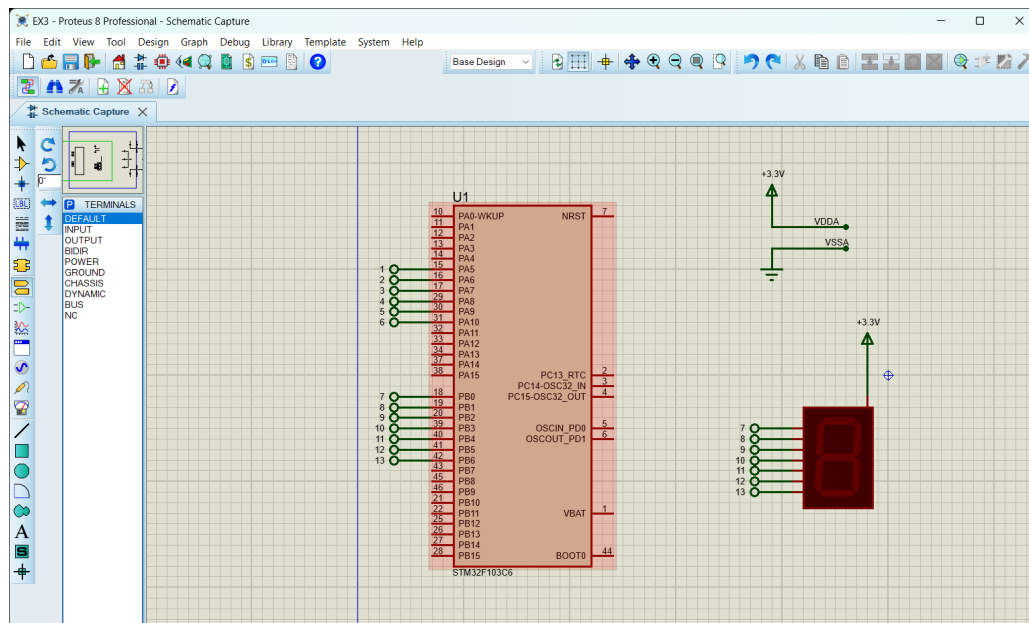
```
}
/* USER CODE END 3 */
```

Link:

Source Code

4 Exercise 4

Report 1: Present the schematic:



Report 2: Present the source code for display7SEG function:

```
void display7SEG(int num){
    switch(num){
        case 0:
            HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_RESET ) ;
            HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_RESET ) ;
            HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
            HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_RESET ) ;
            HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_RESET ) ;
            HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_RESET ) ;
            HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_SET ) ;
            break;
        case 1:
            HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_SET ) ;
            HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_RESET ) ;
            HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
            HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_SET ) ;
            HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_SET ) ;
            HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_SET ) ;
            HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_SET ) ;
            break;
        case 2:
```

```
HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_RESET ) ;
break;
case 3:
HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_RESET ) ;
break;
case 4:
HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_RESET ) ;
break;
case 5:
HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_RESET ) ;
break;
case 6:
HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_RESET ) ;
break;
case 7:
HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_SET ) ;
break;
```

```
case 8:
    HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_RESET ) ;
    break;
case 9:
    HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_RESET ) ;
    break;
default:
    HAL_GPIO_WritePin ( a_GPIO_Port , a_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( b_GPIO_Port , b_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( c_GPIO_Port , c_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( d_GPIO_Port , d_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( e_GPIO_Port , e_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( f_GPIO_Port , f_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( g_GPIO_Port , g_Pin , GPIO_PIN_SET ) ;
    break;
}
}
```

5 Exercise 5

Source Code:

```
int counter = 0;
int counter2 = 0;
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */
    HAL_GPIO_WritePin ( LED_GREEN1_GPIO_Port , LED_GREEN1_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_RED1_GPIO_Port , LED_RED1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW1_GPIO_Port , LED_YELLOW1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_GREEN2_GPIO_Port , LED_GREEN2_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_RED2_GPIO_Port , LED_RED2_Pin , GPIO_PIN_RESET ) ;
    HAL_GPIO_WritePin ( LED_YELLOW2_GPIO_Port , LED_YELLOW2_Pin , GPIO_PIN_SET ) ;
    counter = 5;
    counter2 = 9;
```

```
while (counter >= 0){
    display7SEG( counter --) ;
    display7SEG2( counter2 --) ;
    HAL_Delay (1000) ;
}
HAL_GPIO_WritePin ( LED_GREEN1_GPIO_Port , LED_GREEN1_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_RED1_GPIO_Port , LED_RED1_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_YELLOW1_GPIO_Port , LED_YELLOW1_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( LED_GREEN2_GPIO_Port , LED_GREEN2_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_RED2_GPIO_Port , LED_RED2_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( LED_YELLOW2_GPIO_Port , LED_YELLOW2_Pin , GPIO_PIN_SET ) ;
counter = 3;
while (counter >= 0){
    display7SEG( counter --) ;
    display7SEG2( counter2 --) ;
    HAL_Delay (1000) ;
}
HAL_GPIO_WritePin ( LED_GREEN1_GPIO_Port , LED_GREEN1_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_RED1_GPIO_Port , LED_RED1_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( LED_YELLOW1_GPIO_Port , LED_YELLOW1_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_GREEN2_GPIO_Port , LED_GREEN2_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( LED_RED2_GPIO_Port , LED_RED2_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_YELLOW2_GPIO_Port , LED_YELLOW2_Pin , GPIO_PIN_SET ) ;
counter = 9;
counter2 = 5;
while (counter2 >= 0){
    display7SEG( counter --) ;
    display7SEG2( counter2 --) ;
    HAL_Delay (1000) ;
}
HAL_GPIO_WritePin ( LED_GREEN1_GPIO_Port , LED_GREEN1_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_RED1_GPIO_Port , LED_RED1_Pin , GPIO_PIN_RESET ) ;
HAL_GPIO_WritePin ( LED_YELLOW1_GPIO_Port , LED_YELLOW1_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_GREEN2_GPIO_Port , LED_GREEN2_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_RED2_GPIO_Port , LED_RED2_Pin , GPIO_PIN_SET ) ;
HAL_GPIO_WritePin ( LED_YELLOW2_GPIO_Port , LED_YELLOW2_Pin , GPIO_PIN_RESET ) ;
counter2 = 3;
while (counter >= 0){
    display7SEG( counter --) ;
    display7SEG2( counter2 --) ;
    HAL_Delay (1000) ;
}
/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```

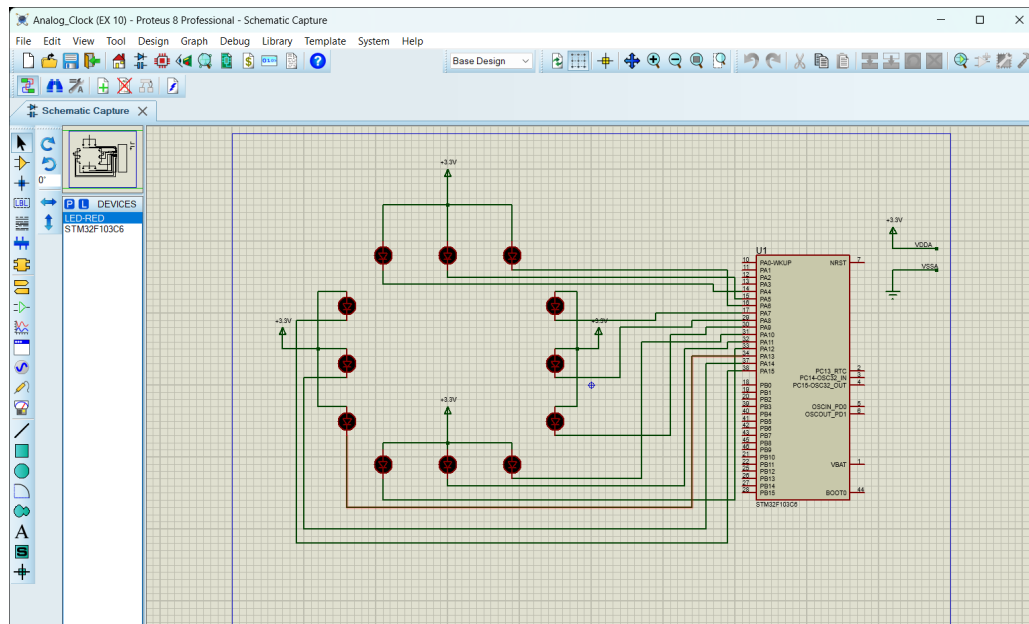
Link:

Source Code

Schematic

6 Exercise 6

Report 1: Present the schematic:



Report 2: Implement a simple program to test the connection of every single LED. This testing program should turn every LED in a sequence.

```
while (1)
{
    HAL_GPIO_WritePin ( LED_1_GPIO_Port , LED_1_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_2_GPIO_Port , LED_2_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_3_GPIO_Port , LED_3_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_4_GPIO_Port , LED_4_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_5_GPIO_Port , LED_5_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_6_GPIO_Port , LED_6_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_7_GPIO_Port , LED_7_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_8_GPIO_Port , LED_8_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_9_GPIO_Port , LED_9_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_10_GPIO_Port , LED_10_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_11_GPIO_Port , LED_11_Pin , GPIO_PIN_SET ) ;
    HAL_GPIO_WritePin ( LED_12_GPIO_Port , LED_12_Pin , GPIO_PIN_SET ) ;
    HAL_Delay(100);
    HAL_GPIO_WritePin ( LED_1_GPIO_Port , LED_1_Pin , GPIO_PIN_RESET ) ;
    HAL_Delay(100);
    HAL_GPIO_WritePin ( LED_2_GPIO_Port , LED_2_Pin , GPIO_PIN_RESET ) ;
    HAL_Delay(100);
    HAL_GPIO_WritePin ( LED_3_GPIO_Port , LED_3_Pin , GPIO_PIN_RESET ) ;
    HAL_Delay(100);
    HAL_GPIO_WritePin ( LED_4_GPIO_Port , LED_4_Pin , GPIO_PIN_RESET ) ;
    HAL_Delay(100);
    HAL_GPIO_WritePin ( LED_5_GPIO_Port , LED_5_Pin , GPIO_PIN_RESET ) ;
    HAL_Delay(100);
    HAL_GPIO_WritePin ( LED_6_GPIO_Port , LED_6_Pin , GPIO_PIN_RESET ) ;
    HAL_Delay(100);
    HAL_GPIO_WritePin ( LED_7_GPIO_Port , LED_7_Pin , GPIO_PIN_RESET ) ;
    HAL_Delay(100);
}
```

```
HAL_GPIO_WritePin ( LED_8_GPIO_Port , LED_8_Pin , GPIO_PIN_RESET ) ;  
    HAL_Delay(100);  
HAL_GPIO_WritePin ( LED_9_GPIO_Port , LED_9_Pin , GPIO_PIN_RESET ) ;  
    HAL_Delay(100);  
HAL_GPIO_WritePin ( LED_10_GPIO_Port , LED_10_Pin , GPIO_PIN_RESET ) ;  
    HAL_Delay(100);  
HAL_GPIO_WritePin ( LED_11_GPIO_Port , LED_11_Pin , GPIO_PIN_RESET ) ;  
    HAL_Delay(100);  
HAL_GPIO_WritePin ( LED_12_GPIO_Port , LED_12_Pin , GPIO_PIN_RESET ) ;  
    HAL_Delay(100);  
}
```

7 Exercise 7

Present the source code:

```
void clearAllClock(void){  
    HAL_GPIO_WritePin ( LED_1_GPIO_Port , LED_1_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_2_GPIO_Port , LED_2_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_3_GPIO_Port , LED_3_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_4_GPIO_Port , LED_4_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_5_GPIO_Port , LED_5_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_6_GPIO_Port , LED_6_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_7_GPIO_Port , LED_7_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_8_GPIO_Port , LED_8_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_9_GPIO_Port , LED_9_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_10_GPIO_Port , LED_10_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_11_GPIO_Port , LED_11_Pin , GPIO_PIN_SET ) ;  
    HAL_GPIO_WritePin ( LED_12_GPIO_Port , LED_12_Pin , GPIO_PIN_SET ) ;  
}
```

8 Exercise 8

Present the source code:

```
void setNumberOnClock(int num){  
    if (num == 0){  
        HAL_GPIO_WritePin ( LED_2_GPIO_Port , LED_2_Pin , GPIO_PIN_RESET ) ;  
    }  
    else if (num == 1){  
        HAL_GPIO_WritePin ( LED_3_GPIO_Port , LED_3_Pin , GPIO_PIN_RESET ) ;  
    }  
    else if (num == 2){  
        HAL_GPIO_WritePin ( LED_4_GPIO_Port , LED_4_Pin , GPIO_PIN_RESET ) ;  
    }  
    else if (num == 3){  
        HAL_GPIO_WritePin ( LED_5_GPIO_Port , LED_5_Pin , GPIO_PIN_RESET ) ;  
    }  
    else if (num == 4){
```

```
    HAL_GPIO_WritePin ( LED_6_GPIO_Port , LED_6_Pin , GPIO_PIN_RESET ) ;  
}  
else if (num == 5){  
    HAL_GPIO_WritePin ( LED_7_GPIO_Port , LED_7_Pin , GPIO_PIN_RESET ) ;  
}  
else if (num == 6){  
    HAL_GPIO_WritePin ( LED_8_GPIO_Port , LED_8_Pin , GPIO_PIN_RESET ) ;  
}  
else if (num == 7){  
    HAL_GPIO_WritePin ( LED_9_GPIO_Port , LED_9_Pin , GPIO_PIN_RESET ) ;  
}  
else if (num == 8){  
    HAL_GPIO_WritePin ( LED_10_GPIO_Port , LED_10_Pin , GPIO_PIN_RESET ) ;  
}  
else if (num == 9){  
    HAL_GPIO_WritePin ( LED_11_GPIO_Port , LED_11_Pin , GPIO_PIN_RESET ) ;  
}  
else if (num == 10){  
    HAL_GPIO_WritePin ( LED_12_GPIO_Port , LED_12_Pin , GPIO_PIN_RESET ) ;  
}  
else if (num == 11){  
    HAL_GPIO_WritePin ( LED_1_GPIO_Port , LED_1_Pin , GPIO_PIN_RESET ) ;  
}  
}
```

9 Exercise 9

Present the source code:

```
void clearNumberOnClock(int num){  
    if (num == 0){  
        HAL_GPIO_WritePin ( LED_2_GPIO_Port , LED_2_Pin , GPIO_PIN_SET ) ;  
    }  
    else if (num == 1){  
        HAL_GPIO_WritePin ( LED_3_GPIO_Port , LED_3_Pin , GPIO_PIN_SET ) ;  
    }  
    else if (num == 2){  
        HAL_GPIO_WritePin ( LED_4_GPIO_Port , LED_4_Pin , GPIO_PIN_SET ) ;  
    }  
    else if (num == 3){  
        HAL_GPIO_WritePin ( LED_5_GPIO_Port , LED_5_Pin , GPIO_PIN_SET ) ;  
    }  
    else if (num == 4){  
        HAL_GPIO_WritePin ( LED_6_GPIO_Port , LED_6_Pin , GPIO_PIN_SET ) ;  
    }  
    else if (num == 5){  
        HAL_GPIO_WritePin ( LED_7_GPIO_Port , LED_7_Pin , GPIO_PIN_SET ) ;  
    }  
    else if (num == 6){  
        HAL_GPIO_WritePin ( LED_8_GPIO_Port , LED_8_Pin , GPIO_PIN_SET ) ;  
    }  
}
```

```
else if (num == 7){
    HAL_GPIO_WritePin ( LED_9_GPIO_Port , LED_9_Pin , GPIO_PIN_SET ) ;
}
else if (num == 8){
    HAL_GPIO_WritePin ( LED_10_GPIO_Port , LED_10_Pin , GPIO_PIN_SET ) ;
}
else if (num == 9){
    HAL_GPIO_WritePin ( LED_11_GPIO_Port , LED_11_Pin , GPIO_PIN_SET ) ;
}
else if (num == 10){
    HAL_GPIO_WritePin ( LED_12_GPIO_Port , LED_12_Pin , GPIO_PIN_SET ) ;
}
else if (num == 11){
    HAL_GPIO_WritePin ( LED_1_GPIO_Port , LED_1_Pin , GPIO_PIN_SET ) ;
}
}
```

10 Exercise 10

Source Code:

```
/* USER CODE BEGIN 2 */
int hour = 0;
int min = 0;
int sec = 0;
/* USER CODE END 2 */
clearAllClock();
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */
    clearNumberOnClock(sec / 5);
    sec++;
    if (sec == 60) {
        sec = 0;
        clearNumberOnClock(min / 5);
        min++;
        setNumberOnClock(min / 5);
    }
    if (min == 60) {
        min = 0;
        clearNumberOnClock(hour % 12);
        hour++;
        setNumberOnClock(hour % 12);
    }
    setNumberOnClock(hour % 12);
    setNumberOnClock(min / 5);
    setNumberOnClock(sec / 5);
    HAL_Delay(100);
    /* USER CODE BEGIN 3 */
}
```



```
}  
/* USER CODE END 3 */
```

Link:

Source Code 6 -> 10

Schematic 6 -> 10