HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Microcontrollers

(CO3117)

Report Lab 1

LED ANIMATIONS

Advisor: TS. Lê Trọng Nhân

Student: Nguyễn Chí Thanh 2313078

Ho Chi Minh City, September 23, 2025

Mục lục

1 Introduction

In this lab, we will explore various LED animations using a microcontroller. The goal is to understand how to control LEDs and create visually appealing effects. This report consists of:

- Source code C language for LED animations in STM32CubeIDE.
- Simulate LED animations using Proteus software.

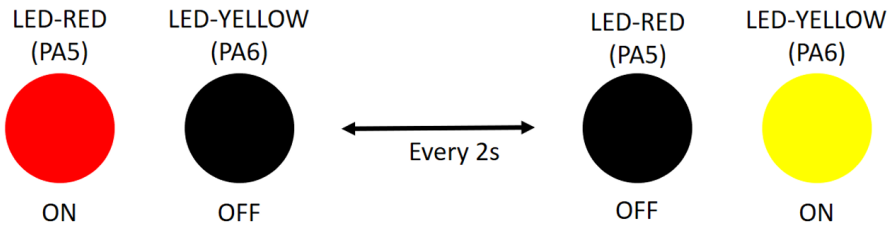
For more details about source code, visit my GitHub repository.

2 Exercises

2.1 Exercise 1

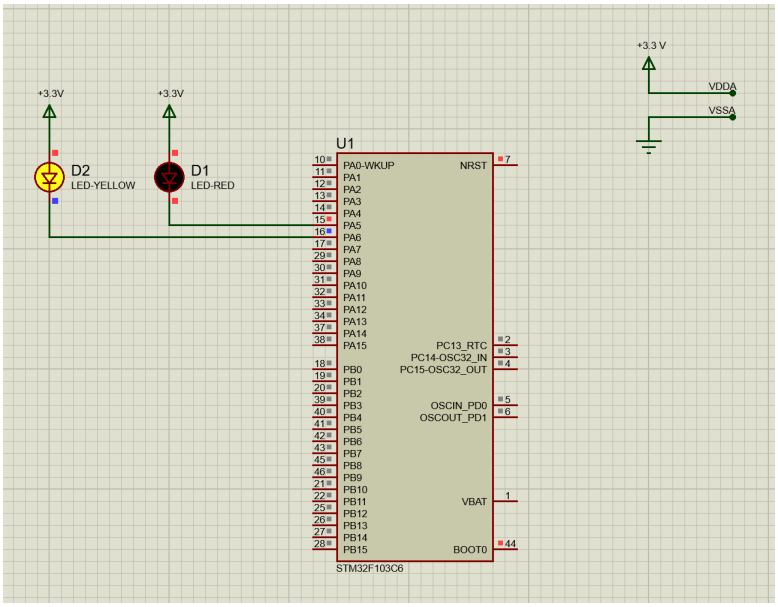
From the simulation on Proteus, an additional LED is connected to pin **PA6** of the STM32 microcontroller (the negative pin of the LED is connected to PA6). The recommended component for this exercise is **LED-YELLOW**, which can be found in the device list.

In this exercise, the states of the two LEDs are toggled every 2 seconds, as illustrated in the figure below.



Hình 2.1: State transitions for 2 LEDs

Schematics from Protus simulation:



Hình 2.2: Download Proteus project file

Source code in STM32CubeIDE:

Listing 1: Source code for Exercise 1

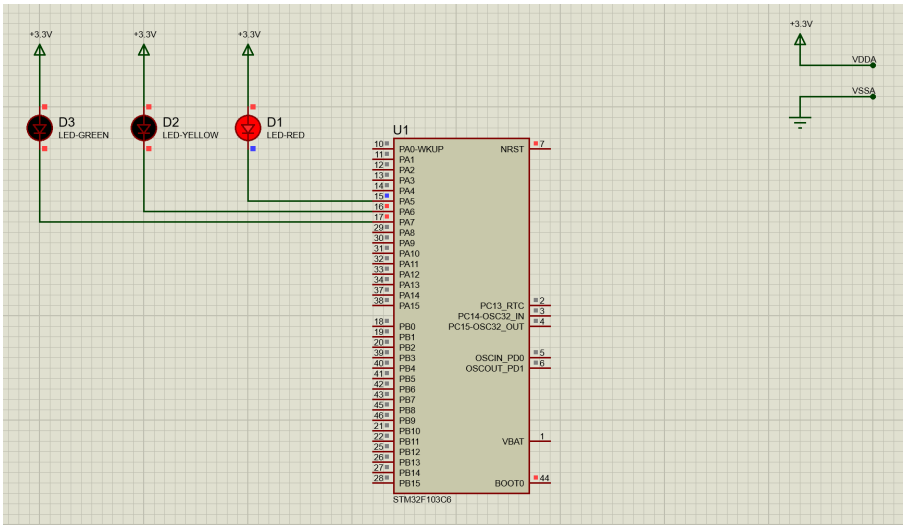
```
 int32_t counter = 400;
 int8_t state = -1;
 while (1){
 if(counter == 400 && state != 0){
      HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, 1);
      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, 0);
      state = 0;
}
else if(counter == 200 && state != 1){
      HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, 0);
      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, 1);
      state = 1;
}
else if(counter <= 1)
      counter = 401;
        counter--;
        HAL_Delay(10);
 }
```

2.2 Exercise 2

Extend the first exercise to simulate the behavior of a traffic light. A third LED, named **LED-GREEN** is added to the system, which is connected to pin **PA7**. A cycle in this traffic light is 5 seconds for the RED, 2 seconds for the YELLOW and 3 seconds for the GREEN. The **LED-GREEN** is also controlled by its negative pin.

Similarly, the report in this exercise includes the schematic of the circuit and the source code in the while loop.

Schematics from Protus simulation:



Hình 2.3: Download Proteus project file

Source code in STM32CubeIDE:

Listing 2: Source code for Exercise 2

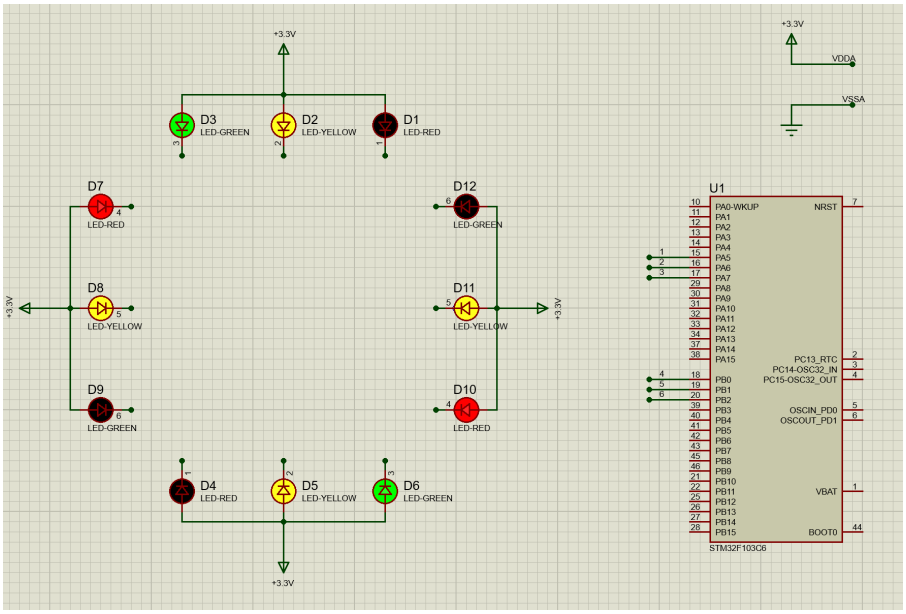
```
int32_t counter = 1000;
int8_t state = -1;
while (1){
  if(counter == 1000 && state != 0){
       HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, 0);
       HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, 1);
       HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, 1);
       state = 0;
 }
 else if(counter == 500 && state != 1){
       HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, 1);
       HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, 0);
```

```
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, 1);
        state = 1;
    }
    else if(counter == 200 && state != 2){
        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, 1);
        HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, 1);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, 0);
        state = 2;
    }
    else if(counter <= 1)
        counter = 1001;
    counter--;
    HAL_Delay(10);
  }
```

2.3 Exercise 3

Extend to the 4-way traffic light. Arrange 12 LEDs in a nice shape to simulate the behaviors of a traffic light. A reference design can be found in the figure below.

Schematics from Protus simulation:



Hình 2.4: Download Proteus project file

Source code in STM32CubeIDE:

Listing 3: Source code for Exercise 3

```
//CONFIGE TIMER FOR TRAFFIC LIGHT
//NOTE: SUM OF TIME 3 LED OF TRAFFIC LIGHT MUST BE EQUAL SUM_TIME_LED
//Real Time = Time x 10 ms
//YELLOW_TIME + GREEN_TIME == RED_TIME_LED OTHER
  #define SUM_TIME_LED 1000

  #define TIME_LED_RED_1 500
  #define TIME_LED_GREEN_1 300
  #define TIME_LED_YELLOW_1 200

  #define TIME_LED_RED_2 500
  #define TIME_LED_GREEN_2 300
  #define TIME_LED_YELLOW_2 200
```

Listing 4: Source code for Exercise 3

```
   int32_t counter = SUM_TIME_LED;
   int8_t state = -1;
   while(1){
       if(counter == SUM_TIME_LED && state != 0){
            HAL_GPIO_WritePin(LED_RED_1_GPIO_Port, LED_RED_1_Pin, 0);
            HAL_GPIO_WritePin(LED_GREEN_1_GPIO_Port, LED_GREEN_1_Pin, 1);
            HAL_GPIO_WritePin(LED_YELLOW_1_GPIO_Port, LED_YELLOW_1_Pin, 1);
            HAL_GPIO_WritePin(LED_RED_2_GPIO_Port, LED_RED_2_Pin, 1);
            HAL_GPIO_WritePin(LED_GREEN_2_GPIO_Port, LED_GREEN_2_Pin, 0);
            HAL_GPIO_WritePin(LED_YELLOW_2_GPIO_Port, LED_YELLOW_2_Pin, 1);
            state = 0;
       }
    else if(counter == SUM_TIME_LED - TIME_LED_GREEN_2 && state != 1){
            HAL_GPIO_WritePin(LED_RED_1_GPIO_Port, LED_RED_1_Pin, 0);
            HAL_GPIO_WritePin(LED_GREEN_1_GPIO_Port, LED_GREEN_1_Pin, 1);
            HAL_GPIO_WritePin(LED_YELLOW_1_GPIO_Port, LED_YELLOW_1_Pin, 1);
            HAL_GPIO_WritePin(LED_RED_2_GPIO_Port, LED_RED_2_Pin, 1);
            HAL_GPIO_WritePin(LED_GREEN_2_GPIO_Port, LED_GREEN_2_Pin, 1);
            HAL_GPIO_WritePin(LED_YELLOW_2_GPIO_Port, LED_YELLOW_2_Pin, 0);
            state = 1;
```

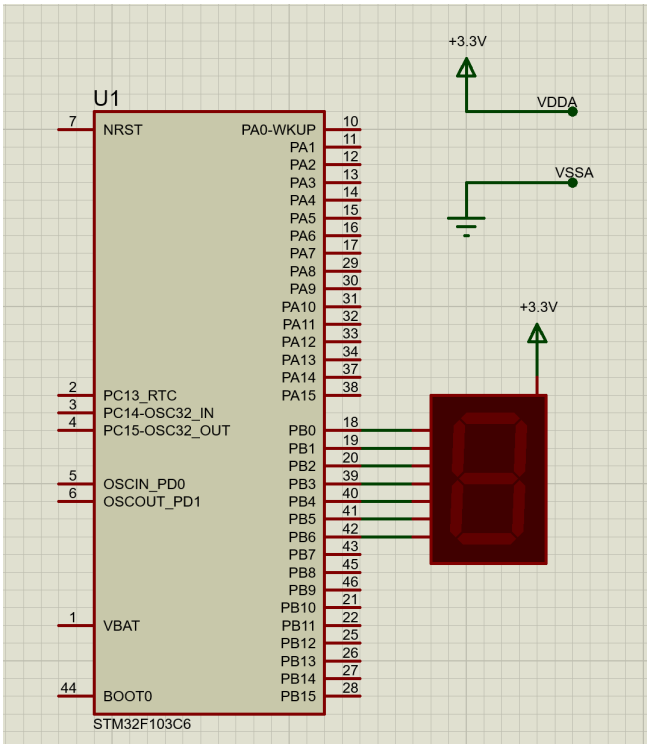
```
    }
    else if(counter == TIME_LED_RED_2 && state != 2){
              HAL_GPIO_WritePin(LED_RED_1_GPIO_Port, LED_RED_1_Pin, 1);
              HAL_GPIO_WritePin(LED_GREEN_1_GPIO_Port, LED_GREEN_1_Pin, 0);
              HAL_GPIO_WritePin(LED_YELLOW_1_GPIO_Port, LED_YELLOW_1_Pin, 1);
              HAL_GPIO_WritePin(LED_RED_2_GPIO_Port, LED_RED_2_Pin, 0);
              HAL_GPIO_WritePin(LED_GREEN_2_GPIO_Port, LED_GREEN_2_Pin, 1);
              HAL_GPIO_WritePin(LED_YELLOW_2_GPIO_Port, LED_YELLOW_2_Pin, 1);
              state = 2;
       }
    else if(counter == TIME_LED_YELLOW_1 && state != 3){
                 HAL_GPIO_WritePin(LED_RED_1_GPIO_Port, LED_RED_1_Pin, 1);
                 HAL_GPIO_WritePin(LED_GREEN_1_GPIO_Port, LED_GREEN_1_Pin, 1);
                 HAL_GPIO_WritePin(LED_YELLOW_1_GPIO_Port, LED_YELLOW_1_Pin, 0);
              HAL_GPIO_WritePin(LED_RED_2_GPIO_Port, LED_RED_2_Pin, 0);
              HAL_GPIO_WritePin(LED_GREEN_2_GPIO_Port, LED_GREEN_2_Pin, 1);
              HAL_GPIO_WritePin(LED_YELLOW_2_GPIO_Port, LED_YELLOW_2_Pin, 1);
                 state = 3;
       }
    else if(counter <= 1)
            counter = SUM_TIME_LED+1;
       counter--;
       HAL_Delay(10);
  }
```

2.4 Exercise 4

Add **only one 7 led segment** to the schematic in Exercise 3. This component can be found in Proteus by the keyword **7SEG-COM-ANODE**. For this device, the common pin should be connected to the power supply and other pins are supposed to be connected to PB0 to PB6. Therefore, to turn-on a segment in this 7SEG, the STM32 pin should be in logic 0 (0V).

Implement a function named **display7SEG(int num)**. The input for this function is from 0 to 9 and the outputs are listed as following:

Schematics from Protus simulation:



Hình 2.5: Download Proteus project file

Source code in STM32CubeIDE:

Listing 5: Source code for Exercise 4

```
void display7SEG(int counter){
   if(counter > 9 || counter < 0) return;
   HAL_GPIO_WritePin(PIN_A_GPIO_Port, PIN_A_Pin, counter==1 || counter==4);
   HAL_GPIO_WritePin(PIN_B_GPIO_Port, PIN_B_Pin, counter==5 || counter==6);
   HAL_GPIO_WritePin(PIN_C_GPIO_Port, PIN_C_Pin, counter==2);
   HAL_GPIO_WritePin(PIN_D_GPIO_Port, PIN_D_Pin, counter==1 || counter==4 ||
       counter==7);
   HAL_GPIO_WritePin(PIN_E_GPIO_Port, PIN_E_Pin, !(counter == 0 || counter == 2 ||
       counter == 6 || counter == 8));
   HAL_GPIO_WritePin(PIN_F_GPIO_Port, PIN_F_Pin, counter==1 || counter==2 ||
       counter==3 || counter==7);
   HAL_GPIO_WritePin(PIN_G_GPIO_Port, PIN_G_Pin, counter== 0 || counter==1 ||
       counter==7);
}
```

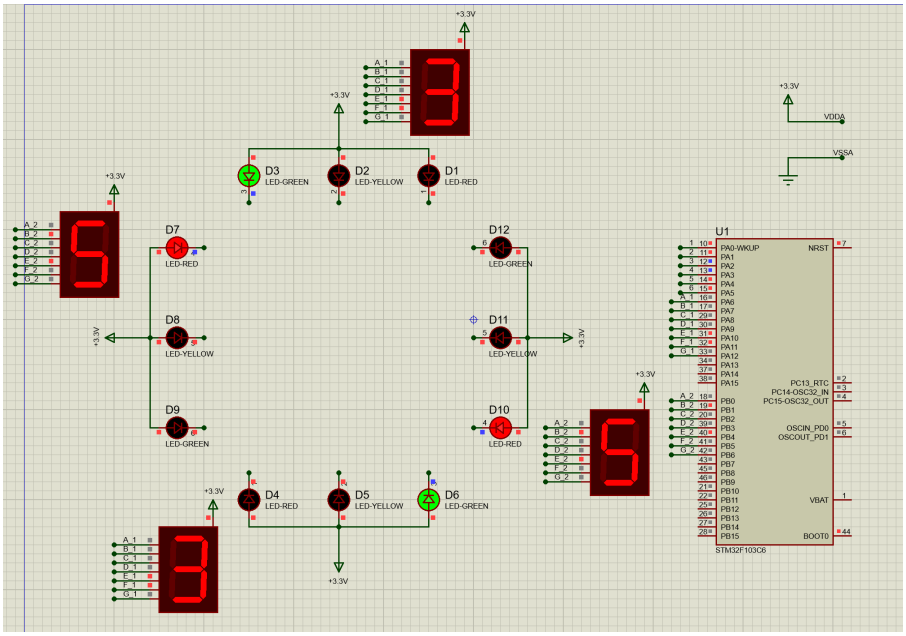
Listing 6: Source code for Exercise 4

```
 int32_t counter = 100;
uint8_t state = 0;
 while(1){
     if(state >= 10) state = 0;
   if(counter <= 0){
       display7SEG(state++);
       counter = 100;
   }
   counter--;
   HAL_Delay(10);}
```

2.5 Exercise 5

Integrate the 7SEG-LED to the 4-way traffic light. In this case, the 7SEG-LED is used to display countdown value.

In this exercise, only source code is required to present. The function **display7SEG** in previous exercise can be re-used.



Hình 2.6: Download Proteus project file

Source code in STM32CubeIDE:

Listing 7: Source code for Exercise 5

```
//CONFIGE TIMER FOR TRAFFIC LIGHT
//NOTE: SUM OF TIME 3 LED OF TRAFFIC LIGHT MUST BE EQUAL SUM_TIME_LED
//Real Time = Time x 10 ms and < 10 s
//YELLOW_TIME + GREEN_TIME == RED_TIME_LED OTHER
#define SUM_TIME_LED 1000
#define TIME_LED_RED_1 500
#define TIME_LED_GREEN_1 300
#define TIME_LED_YELLOW_1 200
#define TIME_LED_RED_2 500
#define TIME_LED_GREEN_2 300
#define TIME_LED_YELLOW_2 200
void display7SEG(int counter, int i){
   if(counter > 9 || counter < 0) return;
   if(i == 1){
   HAL_GPIO_WritePin(PIN_A_1_GPIO_Port, PIN_A_1_Pin, counter==1 || counter==4);
   HAL_GPIO_WritePin(PIN_B_1_GPIO_Port, PIN_B_1_Pin, counter==5 || counter==6);
   HAL_GPIO_WritePin(PIN_C_1_GPIO_Port, PIN_C_1_Pin, counter==2);
   HAL_GPIO_WritePin(PIN_D_1_GPIO_Port, PIN_D_1_Pin, counter==1 || counter==4 ||
       counter==7);
   HAL_GPIO_WritePin(PIN_E_1_GPIO_Port, PIN_E_1_Pin, !(counter == 0 || counter ==
       2 || counter == 6 || counter == 8));
   HAL_GPIO_WritePin(PIN_F_1_GPIO_Port, PIN_F_1_Pin, counter==1 || counter==2 ||
       counter==3 || counter==7);
   HAL_GPIO_WritePin(PIN_G_1_GPIO_Port, PIN_G_1_Pin, counter== 0 || counter==1 ||
       counter==7);
   }
   else{
   HAL_GPIO_WritePin(PIN_A_2_GPIO_Port, PIN_A_2_Pin, counter==1 || counter==4);
   HAL_GPIO_WritePin(PIN_B_2_GPIO_Port, PIN_B_2_Pin, counter==5 || counter==6);
   HAL_GPIO_WritePin(PIN_C_2_GPIO_Port, PIN_C_2_Pin, counter==2);
   HAL_GPIO_WritePin(PIN_D_2_GPIO_Port, PIN_D_2_Pin, counter==1 || counter==4 ||
       counter==7);
   HAL_GPIO_WritePin(PIN_E_2_GPIO_Port, PIN_E_2_Pin, !(counter == 0 || counter ==
       2 || counter == 6 || counter == 8));
   HAL_GPIO_WritePin(PIN_F_2_GPIO_Port, PIN_F_2_Pin, counter==1 || counter==2 ||
       counter==3 || counter==7);
   HAL_GPIO_WritePin(PIN_G_2_GPIO_Port, PIN_G_2_Pin, counter== 0 || counter==1 ||
       counter==7);
```

```
    }
}
```

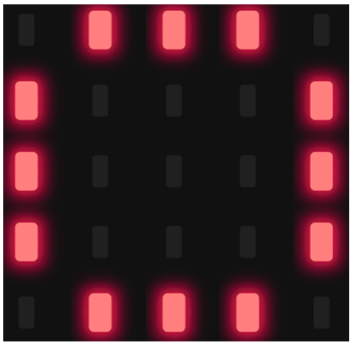
Listing 8: Source code for Exercise 5

```
  int32_t counter = SUM_TIME_LED;
  int8_t state = -1;
  int32_t counter_7LED = 0;
while(1){
//CONTROL LED TRAFFIC
    if(counter == SUM_TIME_LED && state != 0){
                HAL_GPIO_WritePin(LED_RED_1_GPIO_Port, LED_RED_1_Pin, 0);
                HAL_GPIO_WritePin(LED_GREEN_1_GPIO_Port, LED_GREEN_1_Pin, 1);
                HAL_GPIO_WritePin(LED_YELLOW_1_GPIO_Port, LED_YELLOW_1_Pin, 1);
                HAL_GPIO_WritePin(LED_RED_2_GPIO_Port, LED_RED_2_Pin, 1);
                HAL_GPIO_WritePin(LED_GREEN_2_GPIO_Port, LED_GREEN_2_Pin, 0);
                HAL_GPIO_WritePin(LED_YELLOW_2_GPIO_Port, LED_YELLOW_2_Pin, 1);
                state = 0;
          }
       else if(counter == SUM_TIME_LED - TIME_LED_GREEN_2 && state != 1){
                HAL_GPIO_WritePin(LED_RED_1_GPIO_Port, LED_RED_1_Pin, 0);
                HAL_GPIO_WritePin(LED_GREEN_1_GPIO_Port, LED_GREEN_1_Pin, 1);
                HAL_GPIO_WritePin(LED_YELLOW_1_GPIO_Port, LED_YELLOW_1_Pin, 1);
                HAL_GPIO_WritePin(LED_RED_2_GPIO_Port, LED_RED_2_Pin, 1);
                HAL_GPIO_WritePin(LED_GREEN_2_GPIO_Port, LED_GREEN_2_Pin, 1);
                HAL_GPIO_WritePin(LED_YELLOW_2_GPIO_Port, LED_YELLOW_2_Pin, 0);
                state = 1;
       }
       else if(counter == TIME_LED_RED_2 && state != 2){
                   HAL_GPIO_WritePin(LED_RED_1_GPIO_Port, LED_RED_1_Pin, 1);
                   HAL_GPIO_WritePin(LED_GREEN_1_GPIO_Port, LED_GREEN_1_Pin, 0);
                   HAL_GPIO_WritePin(LED_YELLOW_1_GPIO_Port, LED_YELLOW_1_Pin, 1);
                   HAL_GPIO_WritePin(LED_RED_2_GPIO_Port, LED_RED_2_Pin, 0);
                   HAL_GPIO_WritePin(LED_GREEN_2_GPIO_Port, LED_GREEN_2_Pin, 1);
                   HAL_GPIO_WritePin(LED_YELLOW_2_GPIO_Port, LED_YELLOW_2_Pin, 1);
                   state = 2;
          }
       else if(counter == TIME_LED_YELLOW_1 && state != 3){
                      HAL_GPIO_WritePin(LED_RED_1_GPIO_Port, LED_RED_1_Pin, 1);
```

```
                    HAL_GPIO_WritePin(LED_GREEN_1_GPIO_Port, LED_GREEN_1_Pin, 1);
                    HAL_GPIO_WritePin(LED_YELLOW_1_GPIO_Port, LED_YELLOW_1_Pin, 0);
                HAL_GPIO_WritePin(LED_RED_2_GPIO_Port, LED_RED_2_Pin, 0);
                HAL_GPIO_WritePin(LED_GREEN_2_GPIO_Port, LED_GREEN_2_Pin, 1);
                HAL_GPIO_WritePin(LED_YELLOW_2_GPIO_Port, LED_YELLOW_2_Pin, 1);
                    state = 3;
          }
       else if(counter <= 1)
              counter = SUM_TIME_LED+1;
//CONTROL 7 7SEG-LED
        if(counter_7LED <= 0){
           if(state == 0){
              display7SEG((TIME_LED_RED_1 - (SUM_TIME_LED - counter))/100, 1);
              display7SEG((TIME_LED_GREEN_2 - (SUM_TIME_LED - counter))/100, 2);
           }
           else if(state == 1){
              display7SEG((TIME_LED_RED_1 - (SUM_TIME_LED - counter))/100, 1);
              display7SEG((TIME_LED_YELLOW_2 - (SUM_TIME_LED - counter -
                  TIME_LED_GREEN_2))/100, 2);
           }
           else if(state == 2){
              display7SEG((TIME_LED_GREEN_1 - (SUM_TIME_LED - counter -
                  TIME_LED_RED_1))/100, 1);
              display7SEG(counter/100, 2);
           }
           else if(state == 3){
              display7SEG(counter/100, 1);
              display7SEG(counter/100, 2);
           }
            counter_7LED = 100;
       }
         counter_7LED--;
         counter--;
         HAL_Delay(10);
}
```

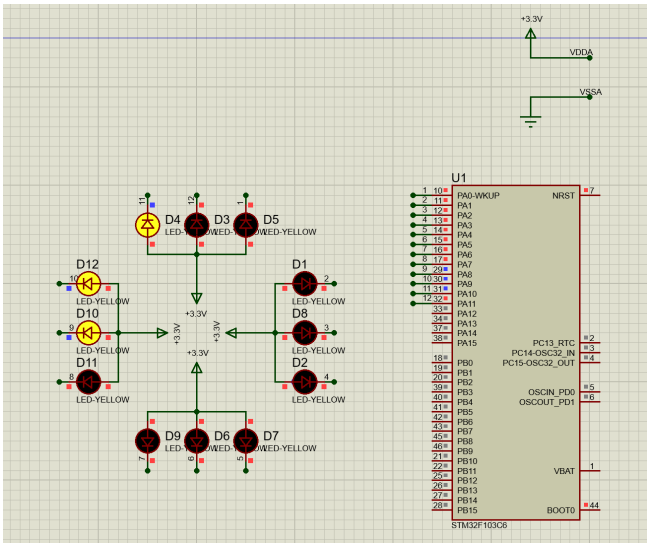
2.6 Exercise 6

In this exercise, a new Proteus schematic is designed to simulate an analog clock, with 12 different numbers. The connections for 12 LEDs are supposed to be from PA4 to PA15 of the STM32. The arrangement of 12 LEDs is depicted as follows.



Hình 2.7: 12 LEDs for an analog clock

Report 1: Present the schematic.



Hình 2.8: Download Proteus project file

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

Listing 9: Source code for Exercise 6

```
int num = 0;
int counter = 100;
```

```
while(1){
if(counter <= 0){
           if(num == 1){
              HAL_GPIO_TogglePin(PIN_1_GPIO_Port, PIN_1_Pin);
           }
           else if(num == 2){
              HAL_GPIO_TogglePin(PIN_2_GPIO_Port, PIN_2_Pin);
           }
           else if(num == 3){
              HAL_GPIO_TogglePin(PIN_3_GPIO_Port, PIN_3_Pin);
           }
           else if(num == 4){
              HAL_GPIO_TogglePin(PIN_4_GPIO_Port, PIN_4_Pin);
           }
           else if(num == 5){
              HAL_GPIO_TogglePin(PIN_5_GPIO_Port, PIN_5_Pin);
           }
           else if(num == 6){
              HAL_GPIO_TogglePin(PIN_6_GPIO_Port, PIN_6_Pin);
           }
           else if(num == 7){
              HAL_GPIO_TogglePin(PIN_7_GPIO_Port, PIN_7_Pin);
           }
           else if(num == 8){
              HAL_GPIO_TogglePin(PIN_8_GPIO_Port, PIN_8_Pin);
           }
           else if(num == 9){
              HAL_GPIO_TogglePin(PIN_9_GPIO_Port, PIN_9_Pin);
           }
           else if(num == 10){
              HAL_GPIO_TogglePin(PIN_10_GPIO_Port, PIN_10_Pin);
           }
           else if(num == 11){
              HAL_GPIO_TogglePin(PIN_11_GPIO_Port, PIN_11_Pin);
           }
           else if(num == 0){
              HAL_GPIO_TogglePin(PIN_12_GPIO_Port, PIN_12_Pin);
           }
```

```
            num++;
            if(num == 12) num = 0;
            counter = 50;
        }
        counter--;
        HAL_Delay(10);
}
```

2.7 Exercise 7

Implement a function named **clearAllClock()** to turn off all 12 LEDs. Present the source code of this function.

Listing 10: Source code for the clearAllClock() function

```
void clearAllClock(){
   HAL_GPIO_WritePin(PIN_1_GPIO_Port, PIN_1_Pin, 1);
   HAL_GPIO_WritePin(PIN_2_GPIO_Port, PIN_2_Pin, 1);
   HAL_GPIO_WritePin(PIN_3_GPIO_Port, PIN_3_Pin, 1);
   HAL_GPIO_WritePin(PIN_4_GPIO_Port, PIN_4_Pin, 1);
   HAL_GPIO_WritePin(PIN_5_GPIO_Port, PIN_5_Pin, 1);
   HAL_GPIO_WritePin(PIN_6_GPIO_Port, PIN_6_Pin, 1);
   HAL_GPIO_WritePin(PIN_7_GPIO_Port, PIN_7_Pin, 1);
   HAL_GPIO_WritePin(PIN_8_GPIO_Port, PIN_8_Pin, 1);
   HAL_GPIO_WritePin(PIN_9_GPIO_Port, PIN_9_Pin, 1);
   HAL_GPIO_WritePin(PIN_10_GPIO_Port, PIN_10_Pin, 1);
   HAL_GPIO_WritePin(PIN_11_GPIO_Port, PIN_11_Pin, 1);
   HAL_GPIO_WritePin(PIN_12_GPIO_Port, PIN_12_Pin, 1);
}
```

Listing 11: Source code for the clearAllClock() function

```
//JUST USING ONLY THIS TASK OR IF ALL PORT OF PIN 1-12 ARE SAME THIS TASK.
void clearAllClock(){
   uint16_t temp = 15;
   HAL_GPIO_WritePin(PIN_1_GPIO_Port, ~(temp<<12) , 1);
}
```

2.8 Exercise 8

Implement a function named **setNumberOnClock(int num)**. The input for this function is from **0 to 11** and an appropriate LED is turn on. Present the source code of this function.

Listing 12: Source code for the setNumberOnClock(int num) function

```
void setNumberOnClock(int num){
if(num > 11 || num < 0) return;
if(num == 1){
   HAL_GPIO_WritePin(PIN_1_GPIO_Port, PIN_1_Pin, 0);
}
else if(num == 2){
   HAL_GPIO_WritePin(PIN_2_GPIO_Port, PIN_2_Pin, 0);
}
else if(num == 3){
   HAL_GPIO_WritePin(PIN_3_GPIO_Port, PIN_3_Pin, 0);
}
else if(num == 4){
   HAL_GPIO_WritePin(PIN_4_GPIO_Port, PIN_4_Pin, 0);
}
else if(num == 5){
   HAL_GPIO_WritePin(PIN_5_GPIO_Port, PIN_5_Pin, 0);
}
else if(num == 6){
   HAL_GPIO_WritePin(PIN_6_GPIO_Port, PIN_6_Pin, 0);
}
else if(num == 7){
   HAL_GPIO_WritePin(PIN_7_GPIO_Port, PIN_7_Pin, 0);
}
else if(num == 8){
   HAL_GPIO_WritePin(PIN_8_GPIO_Port, PIN_8_Pin, 0);
}
else if(num == 9){
   HAL_GPIO_WritePin(PIN_9_GPIO_Port, PIN_9_Pin, 0);
}
else if(num == 10){
   HAL_GPIO_WritePin(PIN_10_GPIO_Port, PIN_10_Pin, 0);
}
else if(num == 11){
```

```
   HAL_GPIO_WritePin(PIN_11_GPIO_Port, PIN_11_Pin, 0);
}
else if(num == 0){
   HAL_GPIO_WritePin(PIN_12_GPIO_Port, PIN_12_Pin, 0);
}
}s
```

Listing 13: Source code for the setNumberOnClock(int num) function

```
//JUST USING ONLY THIS TASK OR IF ALL PORT OF PIN 1-12 ARE SAME THIS TASK.
void setNumberOnClock(int num){
if(num > 11 || num < 0) return;
HAL_GPIO_WritePin(PIN_1_GPIO_Port, 1<<num, 0);
}
```

2.9 Exercise 9

Implement a function named **clearNumberOnClock(int num)**. The input for this function is from **0 to 11** and an appropriate LED is turn off.

Listing 14: Source code for the clearNumberOnClock(int num) function

```
void clearNumberOnClock(int num){
   if(num > 11 || num < 0) return;
   if(num == 1){
      HAL_GPIO_WritePin(PIN_1_GPIO_Port, PIN_1_Pin, 1);
   }
   else if(num == 2){
      HAL_GPIO_WritePin(PIN_2_GPIO_Port, PIN_2_Pin, 1);
   }
   else if(num == 3){
      HAL_GPIO_WritePin(PIN_3_GPIO_Port, PIN_3_Pin, 1);
   }
   else if(num == 4){
      HAL_GPIO_WritePin(PIN_4_GPIO_Port, PIN_4_Pin, 1);
   }
   else if(num == 5){
      HAL_GPIO_WritePin(PIN_5_GPIO_Port, PIN_5_Pin, 1);
   }
```

```
  else if(num == 6){
     HAL_GPIO_WritePin(PIN_6_GPIO_Port, PIN_6_Pin, 1);
  }
  else if(num == 7){
     HAL_GPIO_WritePin(PIN_7_GPIO_Port, PIN_7_Pin, 1);
  }
  else if(num == 8){
     HAL_GPIO_WritePin(PIN_8_GPIO_Port, PIN_8_Pin, 1);
  }
  else if(num == 9){
     HAL_GPIO_WritePin(PIN_9_GPIO_Port, PIN_9_Pin, 1);
  }
  else if(num == 10){
     HAL_GPIO_WritePin(PIN_10_GPIO_Port, PIN_10_Pin, 1);
  }
  else if(num == 11){
     HAL_GPIO_WritePin(PIN_11_GPIO_Port, PIN_11_Pin, 1);
  }
  else if(num == 0){
     HAL_GPIO_WritePin(PIN_12_GPIO_Port, PIN_12_Pin, 1);
  }
}
```

Listing 15: Source code for the clearNumberOnClock(int num) function

```
//JUST USING ONLY THIS TASK OR IF ALL PORT OF PIN 1-12 ARE SAME THIS TASK.
void clearNumberOnClock(int num){
  if(num > 11 || num < 0) return;
  HAL_GPIO_WritePin(PIN_1_GPIO_Port, 1<<num, 1);
}
```

2.10 Exercise 10

Integrate the whole system and use 12 LEDs to display a clock. At a given time, there are only 3 LEDs are turn on for hour, minute and second information.

Listing 16: Source code for Exercise 10 in macro define

```
#define second_init 1; //(1 - 60)
```

```
#define minute_init 49; //(1 - 60)
#define hour_init 3; //(1-12)
```

Listing 17: Source code for Exercise 10 in while loop

```
int8_t hour = hour_init;
int8_t minute = minute_init;
int8_t second = second_init;
int32_t counter = 50;
while(1){
 if(counter <= 0){
                    second++;
                    if(second > 59){
                       minute++;
                       if(minute > 59){
                          hour++;
                          if(hour > 11){
                             hour = 0;
                          }
                          minute = 0;
                       }
                       second = 0;
                    }
                       clearAllClock();
                       setNumberOnClock(second/5);
                       setNumberOnClock(minute/5);
                       setNumberOnClock(hour);
                       counter = 100;
                 }
       counter--;
       HAL_Delay(10);
}
```

References