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

VI XỬ LÝ - VI ĐIỀU KHIỂN

Lab 1

LED ANIMATIONS

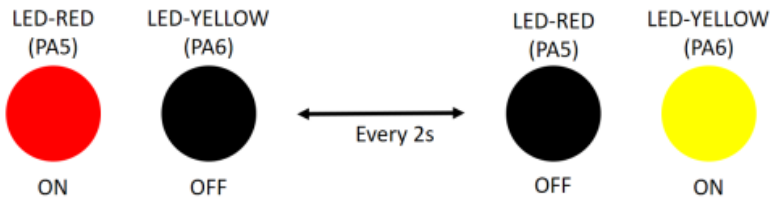
GVHD: **Huỳnh Phúc Nghị**
Thành viên : Đỗ Xuân Thơ 1713365

TP. HỒ CHÍ MINH, THÁNG 09/2021

Mục lục

1 Execises 1

From the simulation on Proteus, one more LED is connected to pin PA6 of the STM32 (negative pin of the LED is connected to PA6). The component suggested in this exercise is LED-YELLOW, which can be found from the device list. In this exercise, the status of two LEDs are switched every 2 seconds, as demonstrated in the figure bellow.

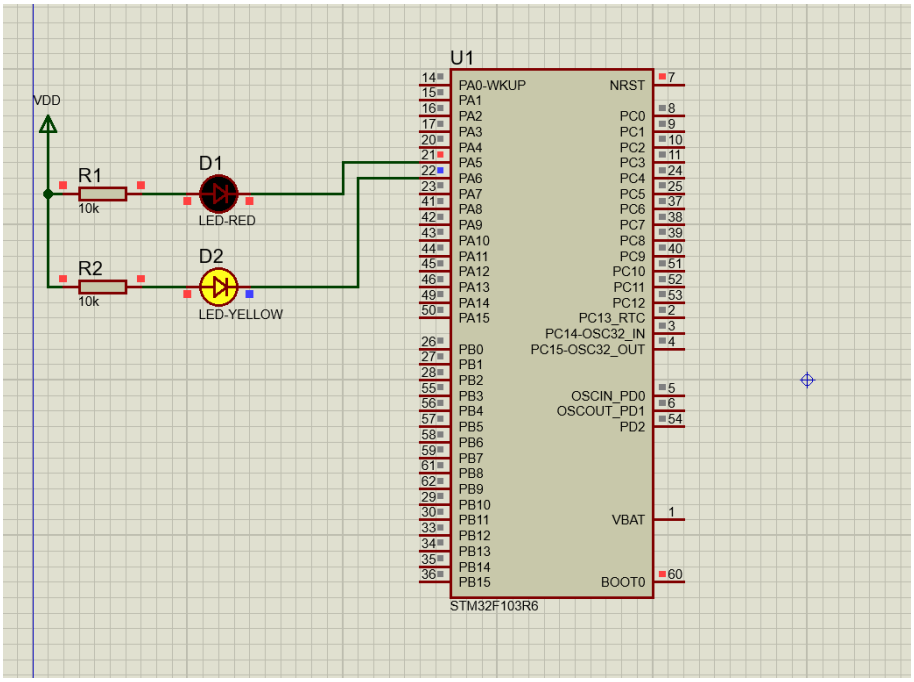


Hình 1: State transition for 2 LEDs

1.1 Source code

```
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
while (1)
{
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
   HAL_Delay(2000);
}
```

1.2 Schematic



Hình 2: Execise 1 - Proteus project file

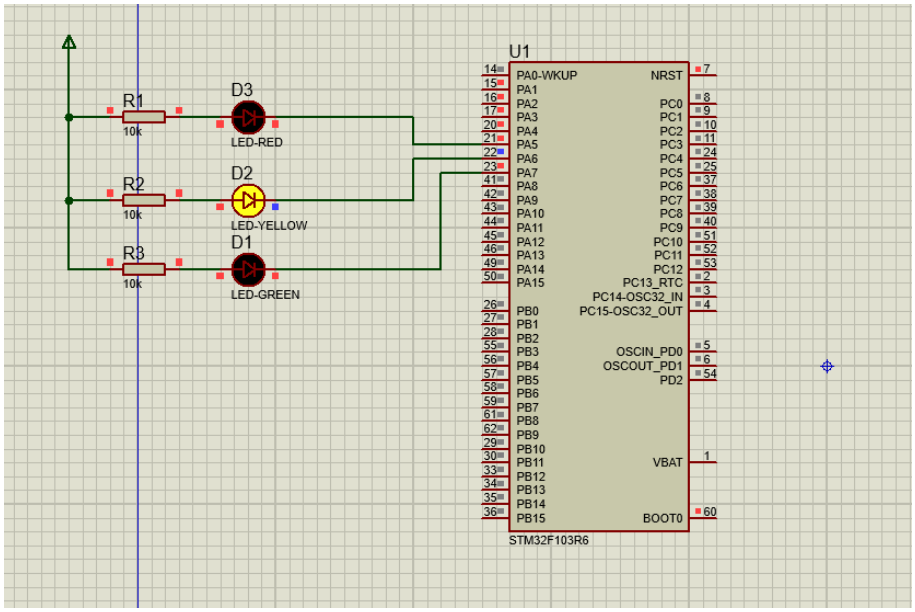
2 Execises 2

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 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 your circuit and a your source code in the while loop.

2.1 Source code

```
while (1)
{
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
   HAL_Delay(5000);
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
   HAL_Delay(2000);
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_7);
   HAL_Delay(3000);
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_7);
}
```

2.2 Schematic



Hình 3: Execise 2 - Proteus project file

3 Execises 3

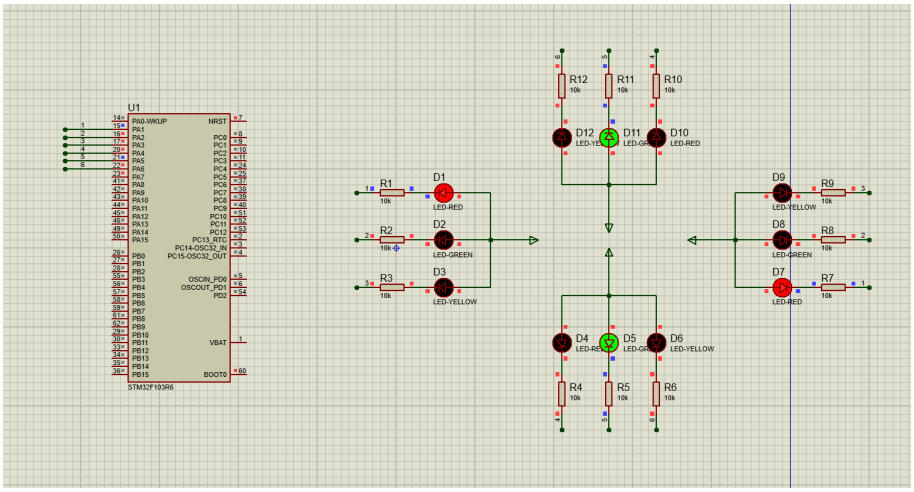
Extend to the 4-way traffic light. Arrange 12 LEDs in a nice shape to simulate the behaviors of a traffic light. A reference design can be found in the figure bellow

3.1 Source code

```
 /*   state
 *    - 0: Red - Green  //Delay 3s
 *    - 1: Red - Yellow //Delay 2s
 *    - 2: Green - Red //Delay 3s
 *    - 3: Yellow - Red //Delay 2s
 */
// Note GPIO_PIN_RESET make LED to light.
uint8_t state = 0;
 while (1)
 {
  switch (state) {
   case 0:
      // Turn off Yellow - Red
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
      // Turn on Red - Green
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
      HAL_Delay(3000);
      break;
   case 1:
      // Turn off Red - Green
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
      // Turn on Red - Yellow
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
      HAL_Delay(2000);
      break;
   case 2:
      // Turn off Red - Yellow
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
      // Turn on Green - Red
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
      HAL_Delay(3000);
      break;
   case 3:
      // Turn off Green - Red
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
      // Turn on Yellow - Red
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
```

```
        HAL_Delay(2000);
        break;
    default:
        break;
     }
   state = (state+1)%4;
}
```

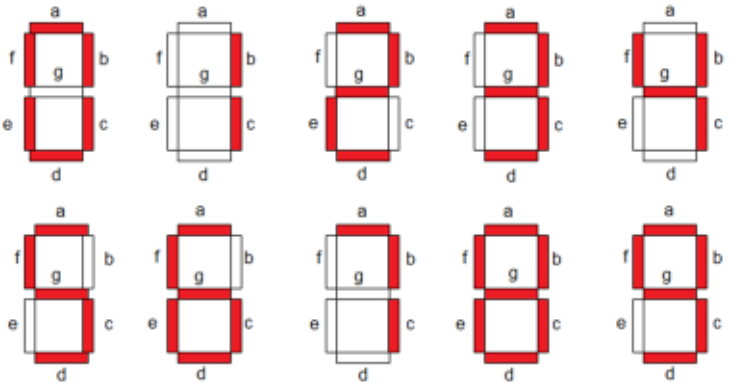
3.2 Schematic



Hình 4: Execise 3 - Proteus project file

4 Execises 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 connected to PB0 to PB6. Therefore, to turn-on a segment in this 7SEG, the STM32 pin should be in logic 0 (0V).



Hình 5: Display a number on 7 segment LED

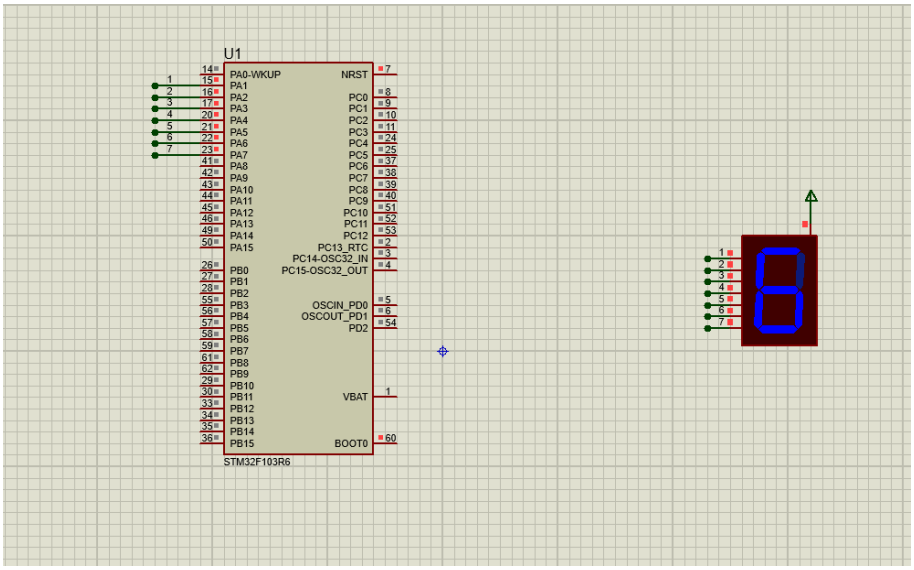
4.1 Source code

```
void display7SEG(uint8_t number){
   switch (number) {
      case 0:
         // Clear LED7SEG
         HAL_GPIO_WritePin(GPIOA,
             GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
             GPIO_PIN_SET);
         /*
          * b'abcdefg = b'1111110
          */
         HAL_GPIO_WritePin(GPIOA,
             GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6, GPIO_PIN_RESET);
         break;
      case 1:
         // Clear LED7SEG
         HAL_GPIO_WritePin(GPIOA,
             GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
             GPIO_PIN_SET);
         /*
          * b'abcdefg = b'0110000
          */
         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2|GPIO_PIN_3, GPIO_PIN_RESET);
         break;
```

```
case 2:
  // Clear LED7SEG
  HAL_GPIO_WritePin(GPIOA,
      GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
      GPIO_PIN_SET);
  /*
   * b'abcdefg = b'1101101
   */
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_7,
      GPIO_PIN_RESET);
  break;
case 3:
  // Clear LED7SEG
  HAL_GPIO_WritePin(GPIOA,
      GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
      GPIO_PIN_SET);
  /*
   * b'abcdefg = b'1111001
   */
  HAL_GPIO_WritePin(GPIOA,
      GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_7, GPIO_PIN_RESET);
  break;
case 4:
  // Clear LED7SEG
  HAL_GPIO_WritePin(GPIOA,
      GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
      GPIO_PIN_SET);
  /*
   * b'abcdefg = b'0110011
   */
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_6|GPIO_PIN_7, GPIO_PIN_RESET);
  break;
case 5:
  // Clear LED7SEG
  HAL_GPIO_WritePin(GPIOA,
      GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
      GPIO_PIN_SET);
  /*
   * b'abcdefg = b'1011011
   */
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_6|GPIO_PIN_7,
      GPIO_PIN_RESET);
  break;
case 6:
  // Clear LED7SEG
  HAL_GPIO_WritePin(GPIOA,
      GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
      GPIO_PIN_SET);
  /*
   * b'abcdefg = b'1011111
   */
  HAL_GPIO_WritePin(GPIOA,
      GPIO_PIN_1|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7, GPIO_PIN_RESET);
```

```
        break;
    case 7:
        // Clear LED7SEG
        HAL_GPIO_WritePin(GPIOA,
            GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
            GPIO_PIN_SET);
        /*
         * b'abcdefg = b'1110000
         */
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, GPIO_PIN_RESET);
        break;
    case 8:
        // Clear LED7SEG
        HAL_GPIO_WritePin(GPIOA,
            GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
            GPIO_PIN_SET);
        /*
         * b'abcdefg = b'1111111
         */
        HAL_GPIO_WritePin(GPIOA,
            GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
            GPIO_PIN_RESET);
        break;
    case 9:
        // Clear LED7SEG
        HAL_GPIO_WritePin(GPIOA,
            GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7,
            GPIO_PIN_SET);
        /*
         * b'abcdefg = b'1111011
         */
        HAL_GPIO_WritePin(GPIOA,
            GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_6|GPIO_PIN_7, GPIO_PIN_RESET);
        break;
    default:
        break;
  }
}
```

4.2 Schematic



Hình 6: Execise 4 - Proteus project file

5 Execises 5

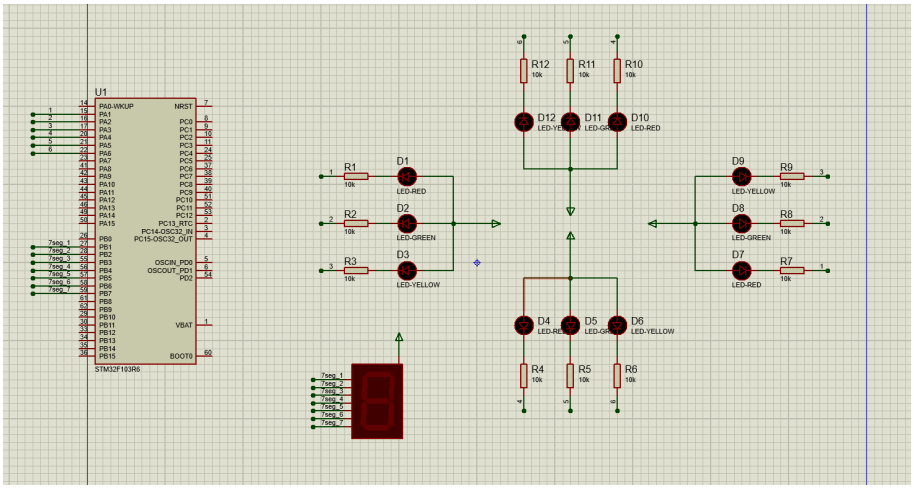
Integrate the 7SEG-LED to the 4 way traffic light. In this case, the 7SEG-LED is used to display countdown value. In this exercise, only source code is required to present. The function display7SEG in previous exercise can be re-used.

5.1 Source code

```
/* state
    *    - 0: Red - Green  //Delay 3s
    *    - 1: Red - Yellow //Delay 2s
    *    - 2: Green - Red //Delay 3s
    *    - 3: Yellow - Red //Delay 2s
    */
   // Note GPIO_PIN_RESET make LED to light.
   uint8_t state = 0;
   uint8_t counter = 0;
   uint8_t count_down;
  while (1)
  {
     switch (state) {
      case 0:
         // Turn off Yellow - Red
         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
         // Turn on Red - Green
         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
```

```
        count_down = 3 - counter;
        if(counter == 3){
            counter = 0;
            state++;
        }
        break;
    case 1:
        // Turn off Red - Green
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
        // Turn on Red - Yellow
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
        count_down = 2 - counter;
        if(counter == 2){
            counter = 0;
            state++;
        }
        break;
    case 2:
        // Turn off Red - Yellow
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
        // Turn on Green - Red
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
        count_down = 3 - counter;
        if(counter == 3){
            counter = 0;
            state++;
        }
        break;
    case 3:
        // Turn off Green - Red
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
        // Turn on Yellow - Red
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
        count_down = 2 - counter;
        if(counter == 2){
            counter = 0;
            state++;
        }
        break;
    default:
        break;
 }
   HAL_Delay(1000);
   display7SEG(count_down);
   counter++;
}
```

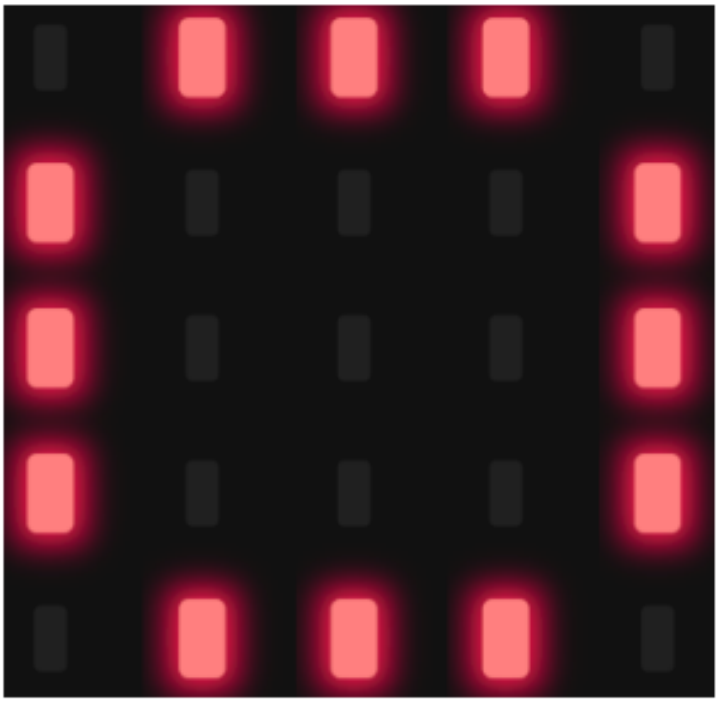
5.2 Schematic



Hình 7: Execise 5 - Proteus project file

6 Execises 6

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

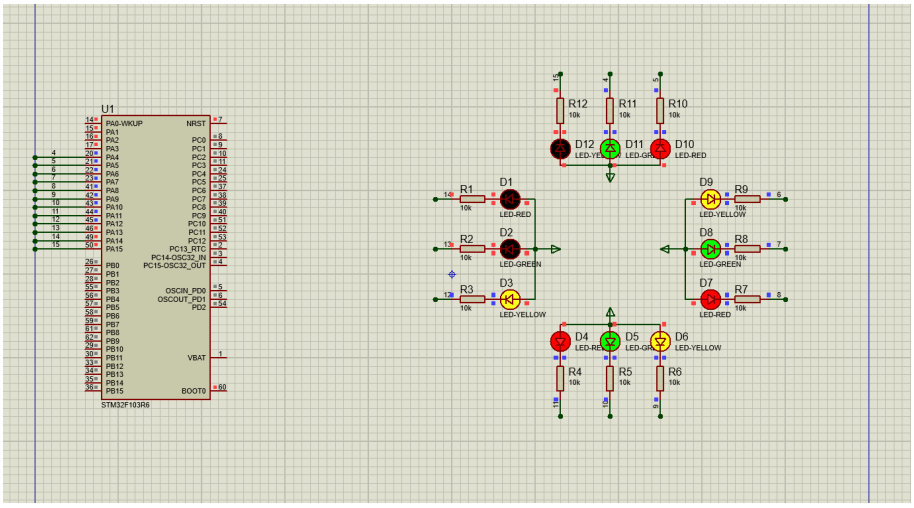


Hình 8: 12 LEDs for an analog clock

6.1 Source code

```
uint16_t led_map[12]={GPIO_PIN_4,GPIO_PIN_5,GPIO_PIN_6,GPIO_PIN_7,GPIO_PIN_8,GPIO_PIN_9,
     GPIO_PIN_10,GPIO_PIN_11,GPIO_PIN_12,GPIO_PIN_13,GPIO_PIN_14,GPIO_PIN_15};
uint8_t counter = 0;
while(1){
   HAL_GPIO_WritePin(GPIOA, led_map[counter], GPIO_PIN_RESET);
   counter = (counter+1)%12;
   HAL_Delay(1000);
}
```

6.2 Schematic



Hình 9: Execise 6 - Proteus project file

7 Execises 7

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

7.1 Source code

```
void clearAllClock(){
   uint16_t mask_gpio_pin = GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9|
        GPIO_PIN_10|GPIO_PIN_11|GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15;
   HAL_GPIO_WritePin(GPIOA, mask_gpio_pin, GPIO_PIN_SET);
}
```

8 Execises 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.

8.1 Source code

```
void setNumberOnClock(int num){
  uint16_t led_map[12]={GPIO_PIN_4,GPIO_PIN_5,GPIO_PIN_6,GPIO_PIN_7,GPIO_PIN_8,GPIO_PIN_9,
       GPIO_PIN_10,GPIO_PIN_11,GPIO_PIN_12,GPIO_PIN_13,GPIO_PIN_14,GPIO_PIN_15};
  HAL_GPIO_WritePin(GPIOA, led_map[num], GPIO_PIN_RESET);

}
```

9 Execises 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.

9.1 Source code

```
void clearNumberOnClock(int num){
   uint16_t led_map[12]={GPIO_PIN_4,GPIO_PIN_5,GPIO_PIN_6,GPIO_PIN_7,GPIO_PIN_8,GPIO_PIN_9,
        GPIO_PIN_10,GPIO_PIN_11,GPIO_PIN_12,GPIO_PIN_13,GPIO_PIN_14,GPIO_PIN_15};
   HAL_GPIO_WritePin(GPIOA, led_map[num], GPIO_PIN_SET);
}
```

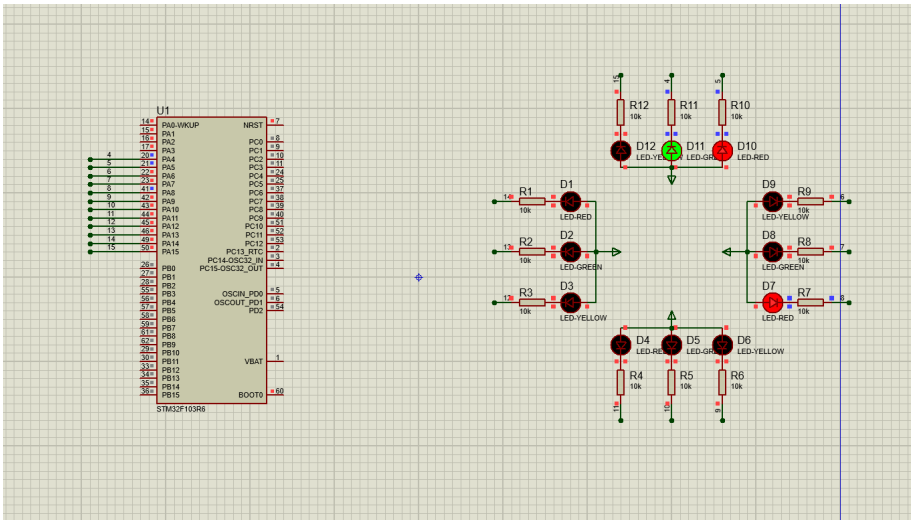
10 Execises 10

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.

10.1 Source code

```
uint8_t seconds = 0;
uint8_t minutes = 0;
uint8_t hours = 0;
while(1){
   clearAllClock();
   setNumberOnClock(seconds/5);
   setNumberOnClock(minutes/5);
   setNumberOnClock(hours);
   HAL_Delay(1000);
   seconds ++;
   if(seconds >= 60){
      seconds = 0;
      minutes++;
      if(minutes >= 60){
         minutes = 0;
         hours ++;
         if(hours >=12){
            hours = 0;
         }
      }
   }
}
```

10.2 Schematic



Hình 10: Execise 10 - Proteus project file