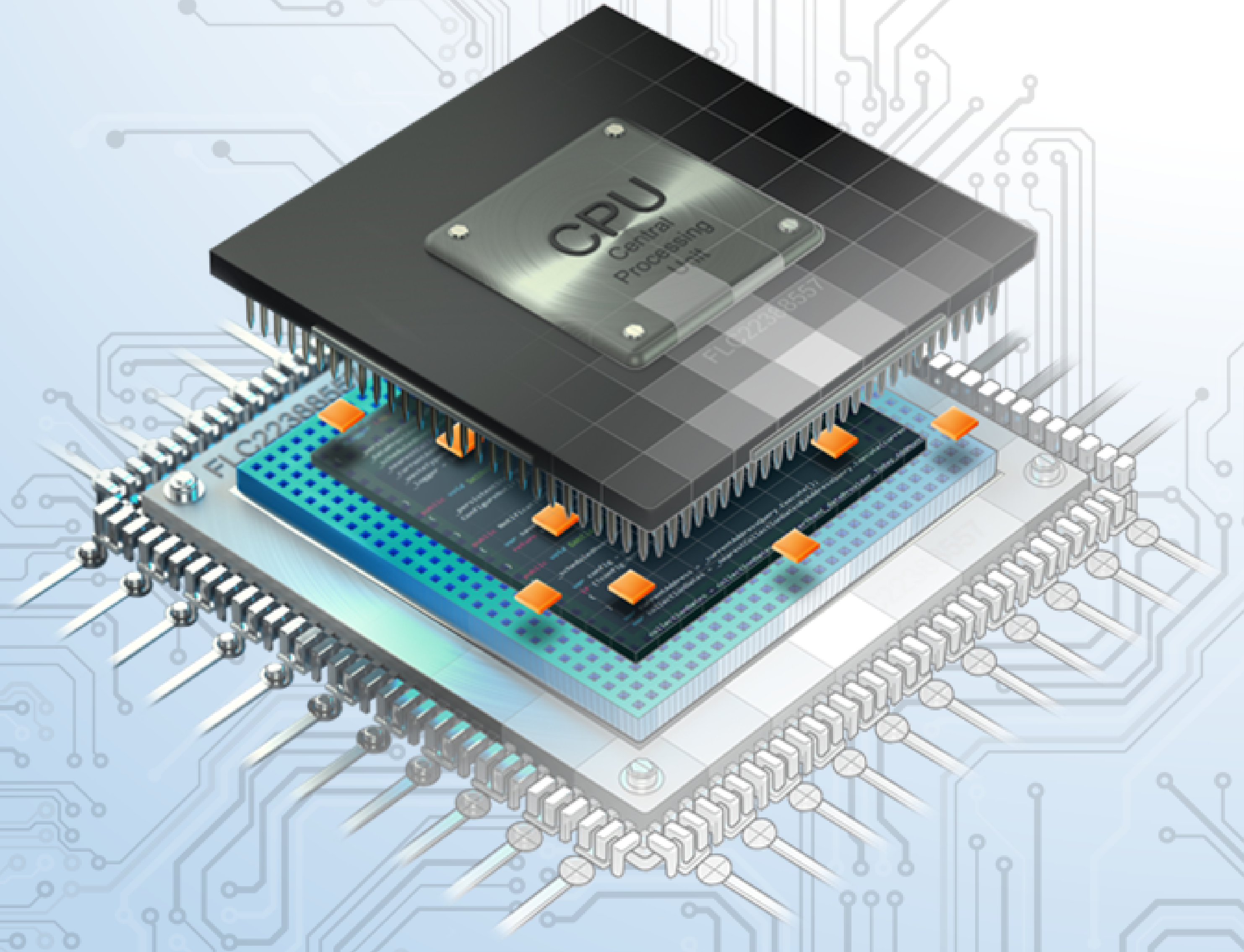
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller



Dr. Le Trong Nhan

Mục lục

Chapter 1. LED Animations **5**

1 Exercise and Report 6

1.1 Exercise 1 6

1.2 Exercise 2 8

1.3 Exercise 3 11

1.4 Exercise 4 11

1.5 Exercise 5 13

1.6 Exercise 6 15

1.7 Exercise 7 17

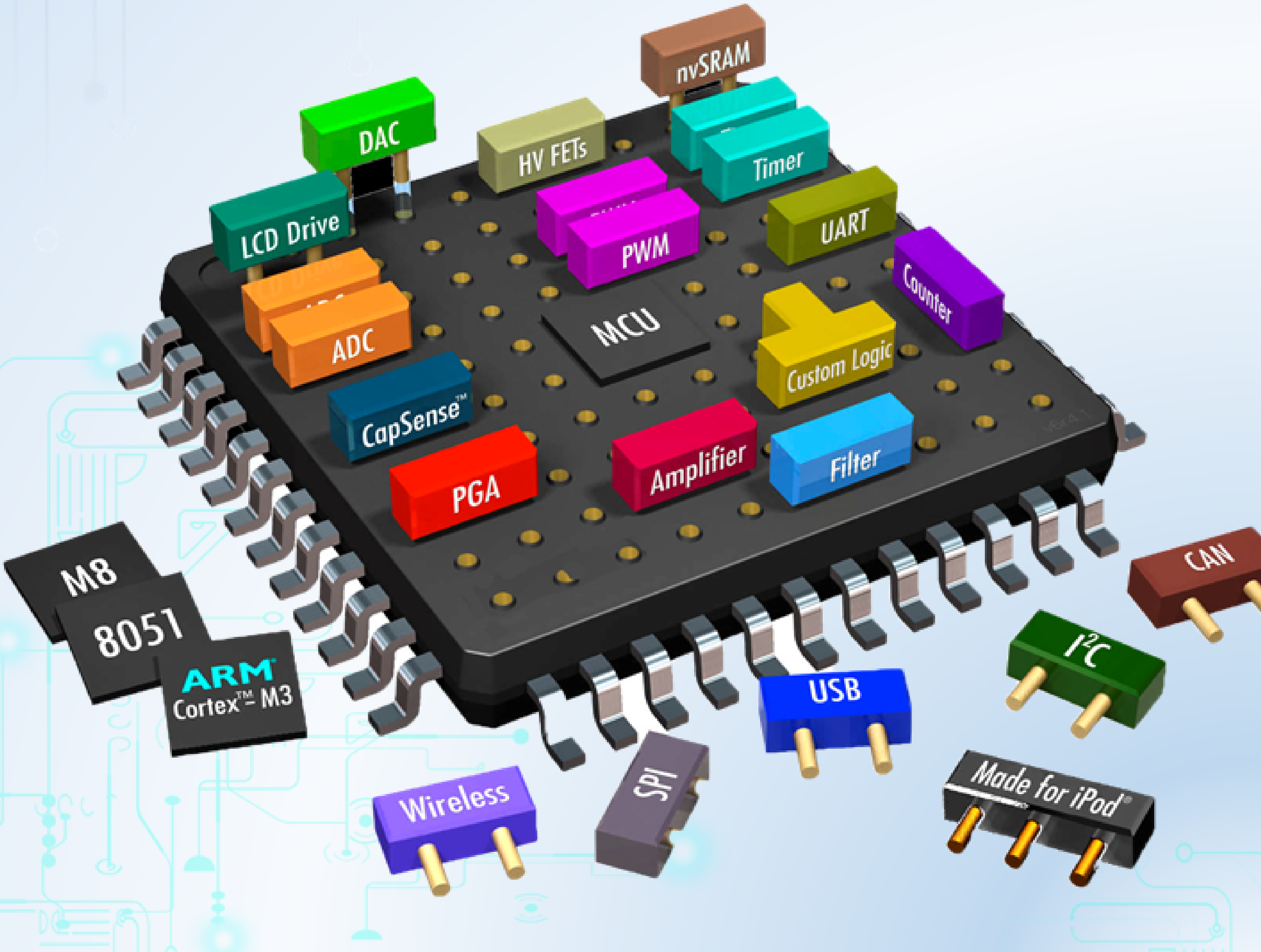
1.8 Exercise 8 18

1.9 Exercise 9 18

1.10 Exercise 10 18

CHƯƠNG 1

LED Animations

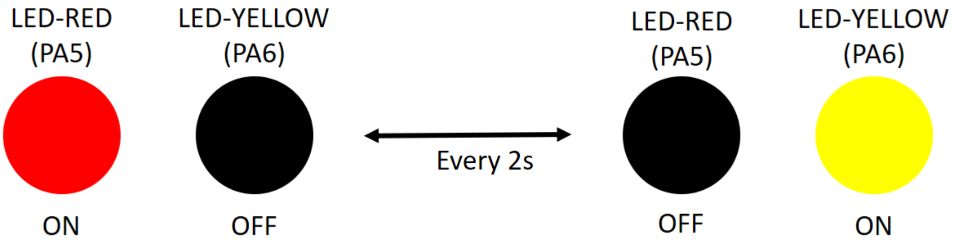


1 Exercise and Report

1.1 Exercise 1

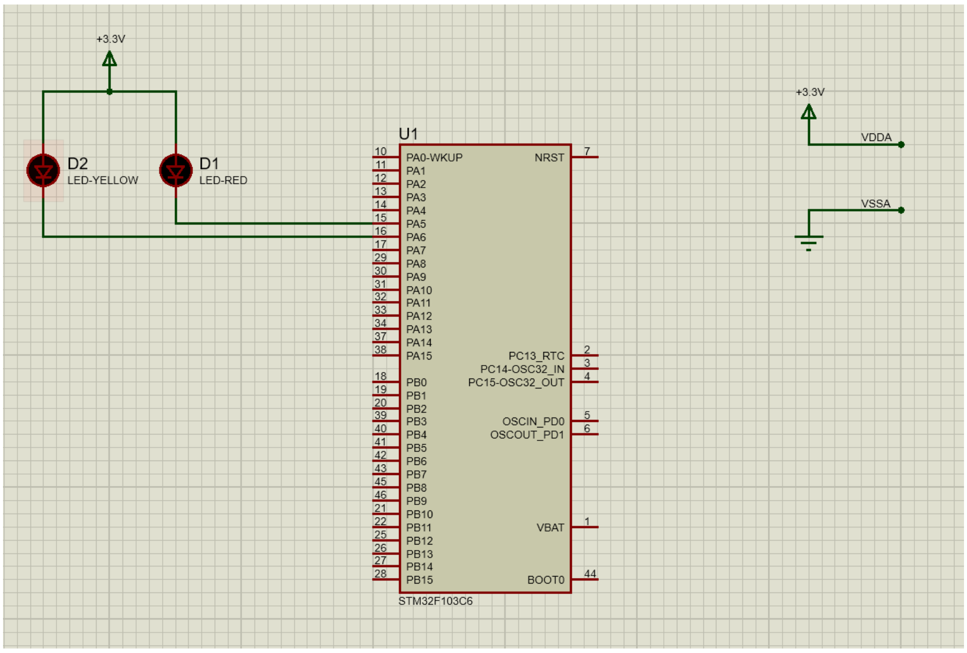
From the simulation on Proteus, one more LED is connected to pin **PA6** of the STM32 (negative pin of the LED is connected to PA6). The component suggested in this exercise is **LED-YELLOW**, which can be found from the device list.

In this exercise, the status of two LEDs are switched every 2 seconds, as demonstrated in the figure bellow.



Hình 1.1: State transitions for 2 LEDs

Report 1: Depict the schematic from Proteus simulation in this report. The caption of the figure is a downloadable link to the Proteus project file (e.g. a github link).



Hình 1.2: Schematic of exercise 1 - Source:

`https://github.com/justkh29/VXL_Lab1/blob/ex1/Ex1.pdsprj`

Report 2: Present the source code in the infinite loop while of your project. If a user-defined functions is used, it is required to present in this part. A brief description can be added for this function (e.g. using comments). A template to present your source code is presented bellow.

```
 int led_status = 0;
 int count = 0;
 while (1)
 {
     switch(led_status)
     {
         case 0:
         {
             HAL_GPIO_WritePin(LED_RED_GPIO_Port,
LED_RED_Pin, SET);
             HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
LED_YELLOW_Pin, RESET);
             if (count >= 2)
             {
                 HAL_GPIO_WritePin(LED_RED_GPIO_Port,
LED_RED_Pin, RESET);
                 HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
 LED_YELLOW_Pin, SET);
                 count = 0;
                 led_status = 1;
             }
             else
             {
                 count++;
             }
             break;
         }
         case 1:
         {
             HAL_GPIO_WritePin(LED_RED_GPIO_Port,
LED_RED_Pin, RESET);
             HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
LED_YELLOW_Pin, SET);
             if (count >= 2)
             {
                 HAL_GPIO_WritePin(LED_RED_GPIO_Port,
LED_RED_Pin, SET);
                 HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
 LED_YELLOW_Pin, RESET);
                 count = 0;
                 led_status = 0;
             }
             else
             {
                 count++;
             }
             break;
```

```
            }
        }
        HAL_Delay(1000);
    }
```

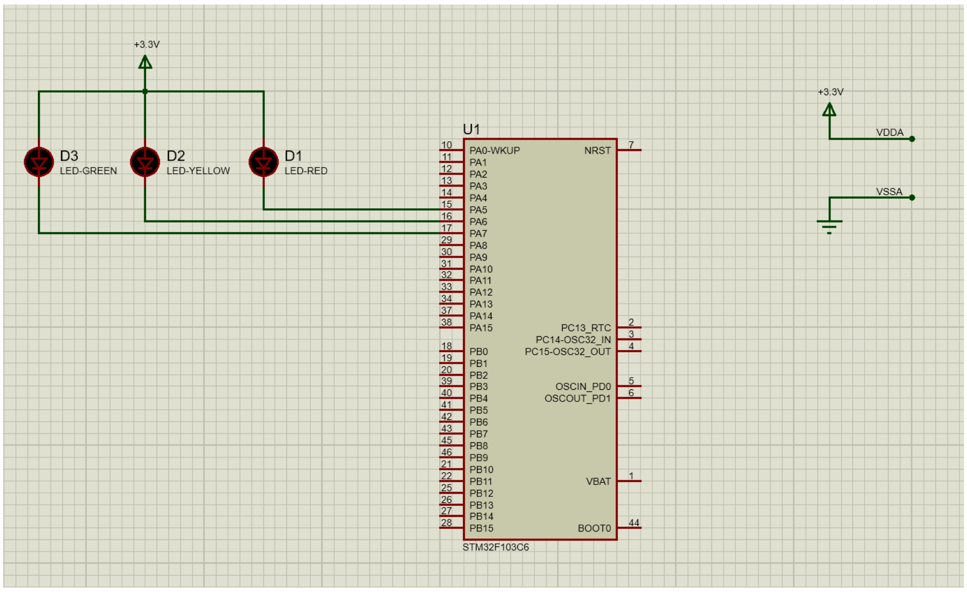
Program 1.1: Source code of exercise 1:
`https://github.com/justkh29/VXL_Lab1/tree/ex1`

1.2 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 **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.

Report 1: Present the schematic.



Hình 1.3: Schematic of exercise 2 - Source:
`https://github.com/justkh29/VXL_Lab1/blob/ex2/Ex2.pdsprj`

Report 2: Present the source code in while.

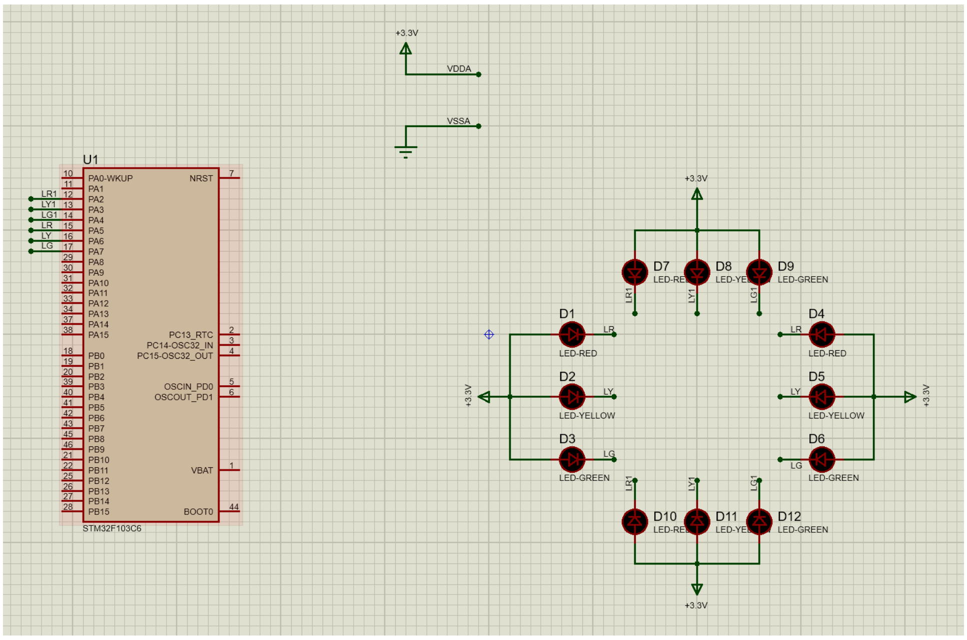
```
    while (1)
    {
    switch(led_status)
    {
      case 0:
      {
        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
  RESET);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
  LED_YELLOW_Pin, SET);
```

```
      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, SET);
      if (count >= 5)
      {
        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, SET);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, SET);
        HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, RESET);
        count = 0;
        led_status = 2;
      }
      else
      {
        count++;
      }
      break;
    }
    case 1:
    {
      HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, SET);
      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, RESET);
      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, SET);
      if (count >= 2)
      {
        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, RESET);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, SET);
        HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, SET);
        count = 0;
        led_status = 0;
      }
      else
      {
        count++;
      }
      break;
    }
    case 2:
    {
      HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, SET);
      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
LED_YELLOW_Pin, SET);
      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port,
LED_GREEN_Pin, RESET);
      if (count >= 3)
      {
        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
 SET);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
LED_YELLOW_Pin, RESET);
        HAL_GPIO_WritePin(LED_GREEN_GPIO_Port,
LED_GREEN_Pin, SET);
        count = 0;
        led_status = 1;
      }
      else
      {
        count++;
      }
      break;
    }
  }
  HAL_Delay(1000);

  /* USER CODE END WHILE */

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

Program 1.2: Source code of exercise 2:
https://github.com/justkh29/VXL_Lab1/tree/ex2

1.3 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 bellow.

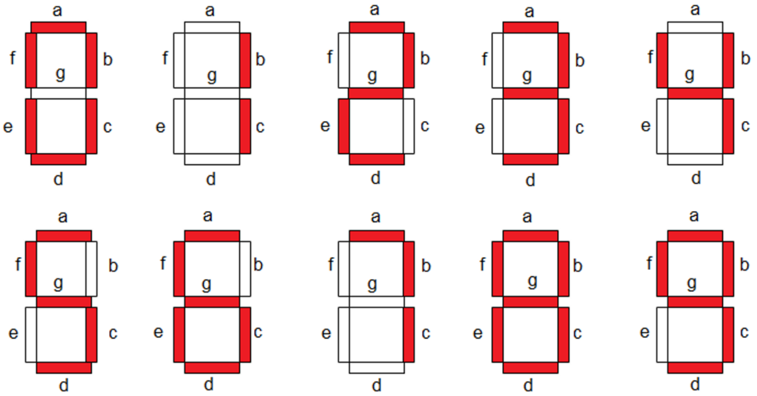


Hình 1.4: Reference design for a 4 way traffic light

1.4 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 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:



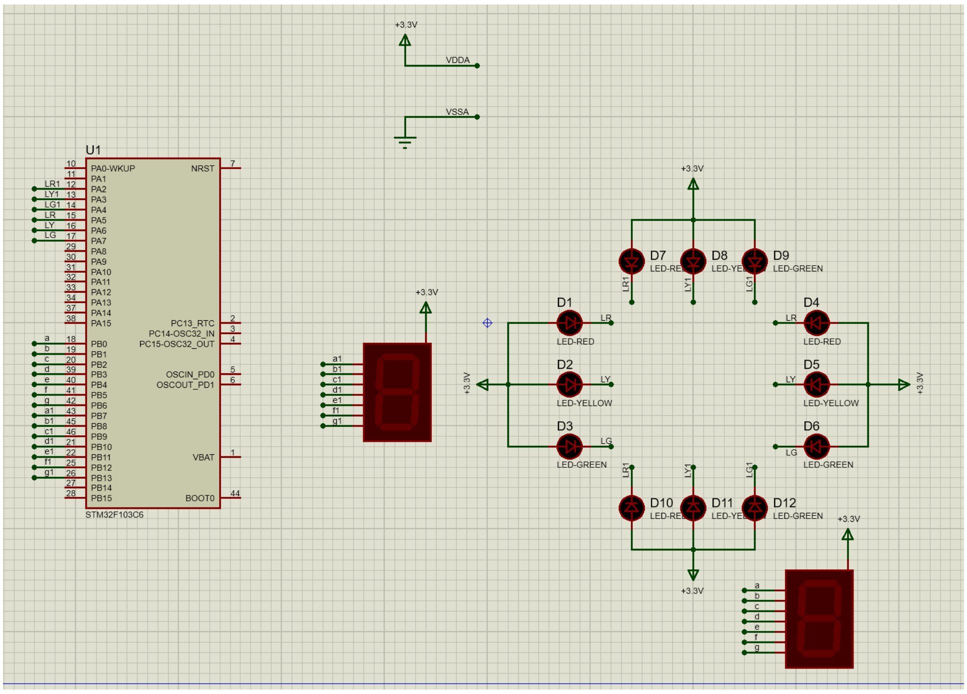
Hình 1.5: Display a number on 7 segment LED

This function is invoked in the while loop for testing as following:

```
int counter = 0;
while (1){
    if(counter >= 10) counter = 0;
    display7SEG(counter++);
    HAL_Delay(1000);

}
```

Program 1.3: An example for your source code

Report 1: Present the schematic.



Hình 1.6: Schematic of exercise 4+5 - Source:
https://github.com/justkh29/VXL_Lab1/blob/ex4+5/traffic.pdsprj

Report 2: Present the source code for display7SEG function.

```
void display7SEG(int num, uint32_t GPIO_Pin)
{
    char segNumber[10] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0
  x92, 0x82, 0xF8, 0x80, 0x90};
    for (int i = 0; i < 7;i++)
    {
        HAL_GPIO_WritePin(GPIOB, GPIO_Pin << i, (segNumber[
  num] >> i) & 1);
    }
}
```

Program 1.4: Source code of exercise 4:
https://github.com/justkh29/VXL_Lab1/tree/ex4+5

1.5 Exercise 5

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

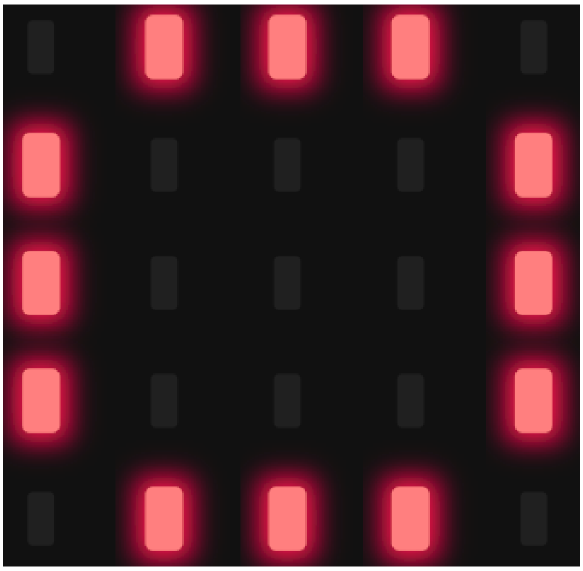
```c
int countRed = 5, countYellow = 2, countGreen = 3, cross = 1;
while (1)
{
    switch (cross)
    {
        case 1:
        {
            if (countRed > 0)
            {
                display7SEG(countRed, GPIO_PIN_7);
                switch (countRed)
                {
                    case 1 ... 2:
                    {
                        HAL_GPIO_WritePin(GPIOA, LED_RED_Pin, RESET);
                        HAL_GPIO_WritePin(GPIOA, LED_YELLOW_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_GREEN_Pin, SET);

                        HAL_GPIO_WritePin(GPIOA, LED_RED1_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_YELLOW1_Pin, RESET);
                        HAL_GPIO_WritePin(GPIOA, LED_GREEN1_Pin, SET);

                        display7SEG(countYellow, GPIO_PIN_0);
                        countYellow--;
                        break;
                    }
                    case 3 ... 5:
                    {
                        HAL_GPIO_WritePin(GPIOA, LED_RED_Pin, RESET);
                        HAL_GPIO_WritePin(GPIOA, LED_YELLOW_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_GREEN_Pin, SET);

                        HAL_GPIO_WritePin(GPIOA, LED_RED1_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_YELLOW1_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_GREEN1_Pin, RESET);

                        display7SEG(countGreen, GPIO_PIN_0);
                        countGreen--;
                        break;
                    }
                }
                countRed--;
            }
            if (countYellow == 0) countYellow = 2;
            if (countGreen == 0) countGreen = 3;
            if (countRed == 0)
            {
                countRed += 5;
                cross = 2;
            }
            break;
        }
        case 2:
        {
            if (countRed > 0)
            {
                display7SEG(countRed, GPIO_PIN_0);
                switch (countRed)
                {

                    case 1 ... 2:
                    {
                        HAL_GPIO_WritePin(GPIOA, LED_RED1_Pin, RESET);
                        HAL_GPIO_WritePin(GPIOA, LED_YELLOW1_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_GREEN1_Pin, SET);

                        HAL_GPIO_WritePin(GPIOA, LED_RED_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_YELLOW_Pin, RESET);
                        HAL_GPIO_WritePin(GPIOA, LED_GREEN_Pin, SET);

                        display7SEG(countYellow, GPIO_PIN_7);
                        countYellow--;
                        break;
                    }
                    case 3 ... 5:
                    {
                        HAL_GPIO_WritePin(GPIOA, LED_RED1_Pin, RESET);
                        HAL_GPIO_WritePin(GPIOA, LED_YELLOW1_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_GREEN1_Pin, SET);

                        HAL_GPIO_WritePin(GPIOA, LED_RED_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_YELLOW_Pin, SET);
                        HAL_GPIO_WritePin(GPIOA, LED_GREEN_Pin, RESET);
                        display7SEG(countGreen, GPIO_PIN_7);
                        countGreen--;
                        break;
                    }

                }
                countRed--;
            }
            if (countYellow == 0) countYellow = 2;
            if (countGreen == 0) countGreen = 3;
            if (countRed == 0)
            {
                countRed += 5;
                cross = 1;
            }
            break;
        }
    }
    HAL_Delay(1000);
/* USER CODE END WHILE */

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

Program 1.5: Source code of exercise 5:
`https://github.com/justkh29/VXL_Lab1/tree/ex4+5`

1.6 Exercise 6

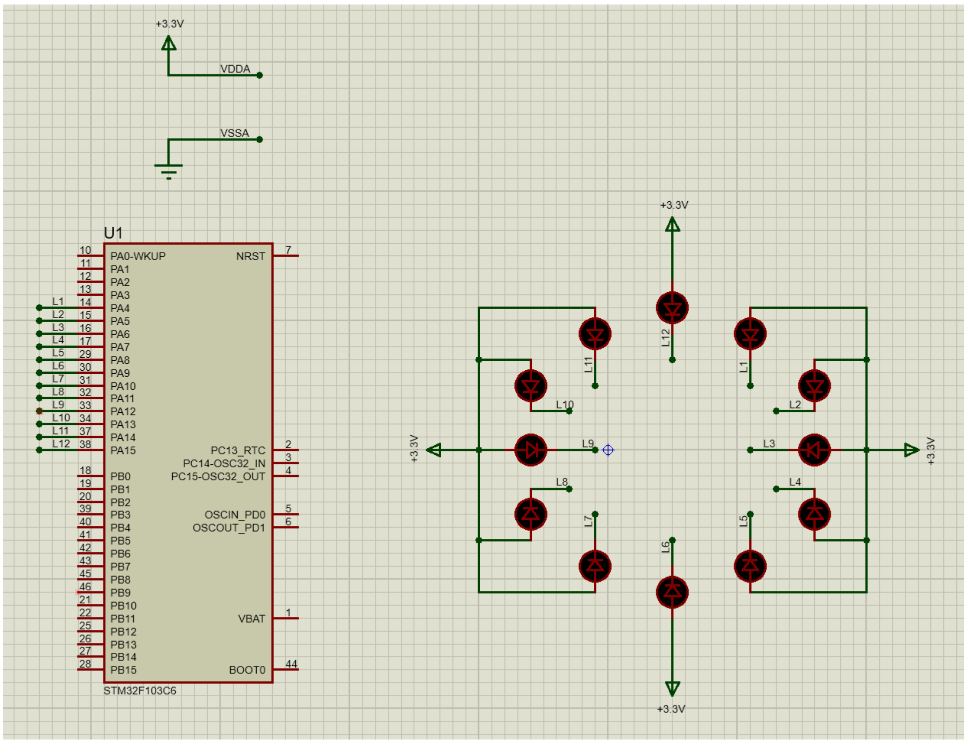
In this exercise, a new Proteus schematic is designed to simulate an analog clock, with 12 different number. The connections for 12 LEDs are supposed from PA4 to

PA15 of the STM32. The arrangement of 12 LEDs is depicted as follows.



Hình 1.7: 12 LEDs for an analog clock

Report 1: Present the schematic.



Hình 1.8: Schematic of exercise 6 - Source:
`https://github.com/justkh29/VXL_Lab1/blob/ex6/clock.pdsprj`

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

```
uint16_t led_pin [12] = {LED_1_Pin, LED_2_Pin, LED_3_Pin,
    LED_4_Pin, LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin,
    LED_9_Pin, LED_10_Pin, LED_11_Pin, LED_12_Pin};
```

```
void turnLED(int n)
{
    HAL_GPIO_WritePin(GPIOA , LED_1_Pin | LED_2_Pin |
  LED_3_Pin | LED_4_Pin | LED_5_Pin | LED_6_Pin |
  LED_7_Pin | LED_8_Pin | LED_9_Pin | LED_10_Pin |
  LED_11_Pin | LED_12_Pin , SET );
    HAL_GPIO_WritePin(GPIOA, led_pin[n], RESET);
}
int count = 0;
while (1)
{
    turnLED(count);
    count++;
    if (count >= 12) count = 0;
    HAL_Delay(1000);
/* USER CODE END WHILE */

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

Program 1.6: Source code of exercise 5:
https://github.com/justkh29/VXL_Lab1/tree/ex6

1.7 Exercise 7

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

```
void clearAllClock(){
    HAL_GPIO_WritePin(GPIOA , LED_1_Pin | LED_2_Pin |
  LED_3_Pin | LED_4_Pin | LED_5_Pin | LED_6_Pin |
  LED_7_Pin | LED_8_Pin | LED_9_Pin | LED_10_Pin |
  LED_11_Pin | LED_12_Pin , SET );
}
```

Program 1.7: Function Implementation:
https://github.com/justkh29/VXL_Lab1/tree/ex7

1.8 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.

```c
void setNumberOnClock(int num)
{
    HAL_GPIO_WritePin(GPIOA, led_pin[num], RESET);
}
```

Program 1.8: Function Implementation:
`https://github.com/justkh29/VXL_Lab1/tree/ex8`

1.9 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.

```c
void clearNumberOnClock(int num)
{
    HAL_GPIO_WritePin(GPIOA, led_pin[num], SET);
}
```

Program 1.9: Function Implementation:
`https://github.com/justkh29/VXL_Lab1/tree/ex9`

1.10 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.

```c
uint16_t led_pin [12] = {LED_1_Pin, LED_2_Pin, LED_3_Pin,
   LED_4_Pin, LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin,
   LED_9_Pin, LED_10_Pin, LED_11_Pin, LED_12_Pin};

void clearAllClock()
{
  HAL_GPIO_WritePin(GPIOA , LED_1_Pin | LED_2_Pin |
   LED_3_Pin | LED_4_Pin | LED_5_Pin | LED_6_Pin |
   LED_7_Pin | LED_8_Pin | LED_9_Pin | LED_10_Pin |
   LED_11_Pin | LED_12_Pin , SET );
}
void setNumberOnClock(int num)
{
  HAL_GPIO_WritePin(GPIOA, led_pin[num], RESET);
}
void clearNumberOnClock(int num)
{
  HAL_GPIO_WritePin(GPIOA, led_pin[num], SET);
```

```
}
int second = 0;
int minute = 0;
int hour = 0;
while (1)
{
  if (second >= 60)
  {
      second = 0;
      minute++;
  }
  if (minute >= 60)
  {
      second = 0;
      minute = 0;
      hour++;
  }
  if (hour >= 12)
  {
      second = 0;
      minute = 0;
      hour = 0;
  }
  clearAllClock();
  switch(second)
  {
      case 0 ... 4:
      {
          setNumberOnClock(11);
          break;
      }
      case 5 ... 9:
      {
          setNumberOnClock(0);
          break;
      }
      case 10 ... 14:
      {
          setNumberOnClock(1);
          break;
      }
      case 15 ... 19:
      {
          setNumberOnClock(2);
          break;
      }
      case 20 ... 24:
      {
          setNumberOnClock(3);
```

```
        break;
    }
    case 25 ... 29:
    {
        setNumberOnClock(4);
        break;
    }
    case 30 ... 34:
    {
        setNumberOnClock(5);
        break;
    }
    case 35 ... 39:
    {
        setNumberOnClock(6);
        break;
    }
    case 40 ... 44:
    {
        setNumberOnClock(7);
        break;
    }
    case 45 ... 49:
    {
        setNumberOnClock(8);
        break;
    }
    case 50 ... 54:
    {
        setNumberOnClock(9);
        break;
    }
    case 55 ... 59:
    {
        setNumberOnClock(10);
        break;
    }
}

switch(minute)
{
    case 0 ... 4:
    {
        setNumberOnClock(11);
        break;
    }
    case 5 ... 9:
    {
        setNumberOnClock(0);
```

```
            break;
        }
        case 10 ... 14:
        {
            setNumberOnClock(1);
            break;
        }
        case 15 ... 19:
        {
            setNumberOnClock(2);
            break;
        }
        case 20 ... 24:
        {
            setNumberOnClock(3);
            break;
        }
        case 25 ... 29:
        {
            setNumberOnClock(4);
            break;
        }
        case 30 ... 34:
        {
            setNumberOnClock(5);
            break;
        }
        case 35 ... 39:
        {
            setNumberOnClock(6);
            break;
        }
        case 40 ... 44:
        {
            setNumberOnClock(7);
            break;
        }
        case 45 ... 49:
        {
            setNumberOnClock(8);
            break;
        }
        case 50 ... 54:
        {
            setNumberOnClock(9);
            break;
        }
        case 55 ... 59:
        {
            setNumberOnClock(10);
            break;
        }
    }

    switch(hour)
    {
        case 0:
        {
            setNumberOnClock(11);
            break;
        }
        case 1:
        {
            setNumberOnClock(0);
            break;
        }
        case 2:
        {
            setNumberOnClock(1);
            break;
        }
        case 3:
        {
            setNumberOnClock(2);
            break;
        }
        case 4:
        {
            setNumberOnClock(3);
            break;
        }
        case 5:
        {
            setNumberOnClock(4);
            break;
        }
        case 6:
        {
            setNumberOnClock(5);
            break;
        }
        case 7:
        {
            setNumberOnClock(6);
            break;
        }
        case 8:
        {
                setNumberOnClock(7);
                break;
            }
            case 9:
            {
                setNumberOnClock(8);
                break;
            }
            case 10:
            {
                setNumberOnClock(9);
                break;
            }
            case 11:
            {
                setNumberOnClock(10);
                break;
            }
        }
        second++;

        HAL_Delay(1000);
/* USER CODE END WHILE */

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

Program 1.10: Function Implementation:
https://github.com/justkh29/VXL_Lab1/tree/ex10