



BÁO CÁO LAB 1

Họ và tên sinh viên: Nguyễn Tấn Thanh

MSSV: 1915095

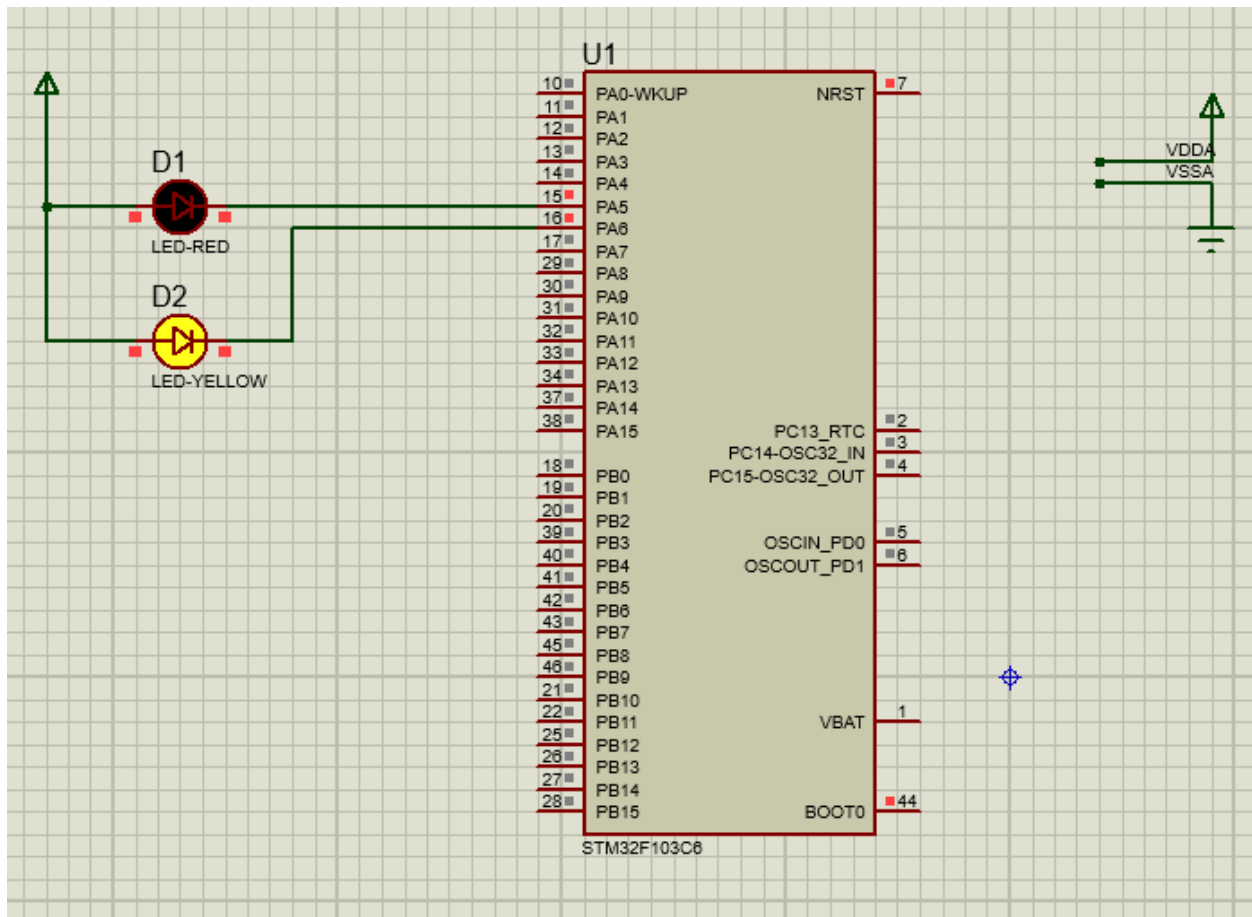
GVHD: Huỳnh Phúc Nghi

Exercise 1:

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).

Link GitHub: <https://github.com/thanhnguyentan131/Vx1-Lab-1.git>

Hình Proteus:



Report 2: Present the source code

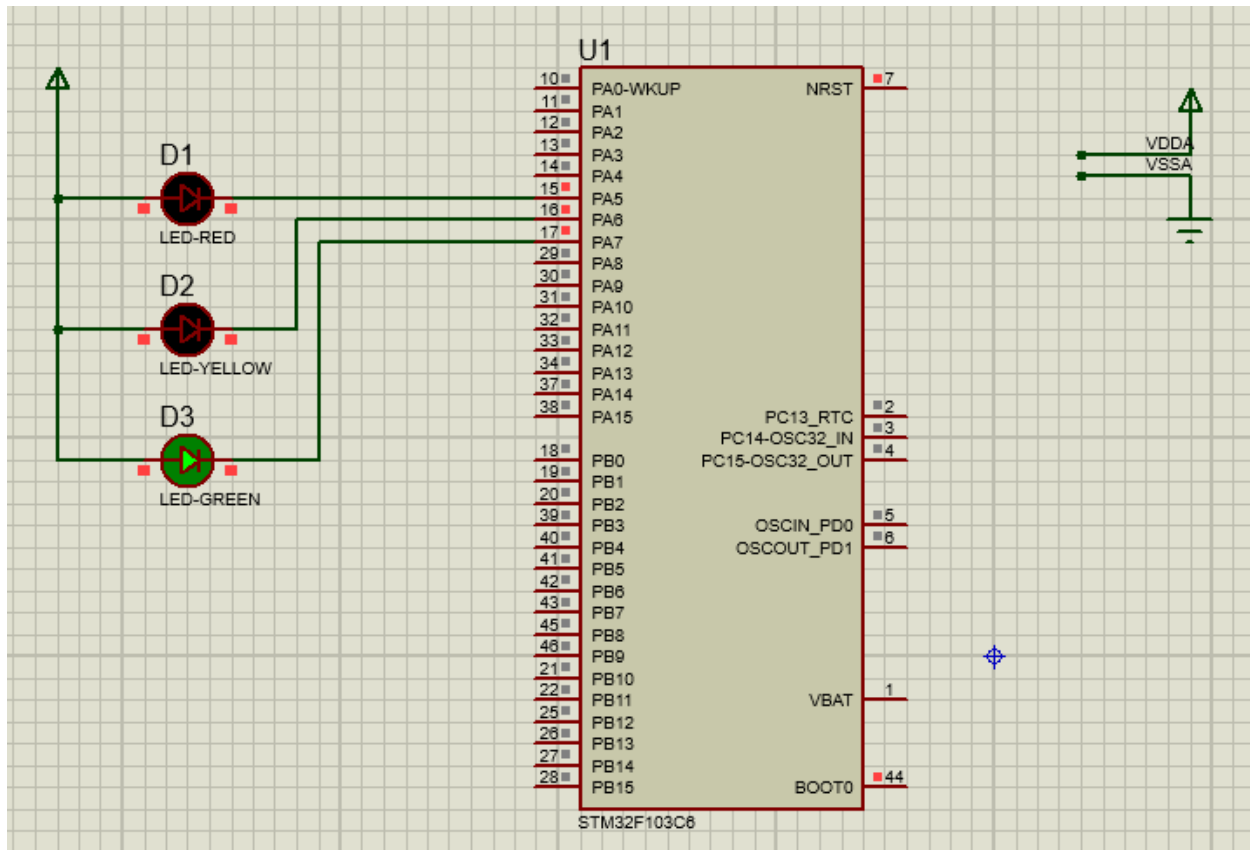
```
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);  
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
```

```
HAL_Delay(2000);  
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);  
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
```

```
HAL_Delay(2000);
```

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.

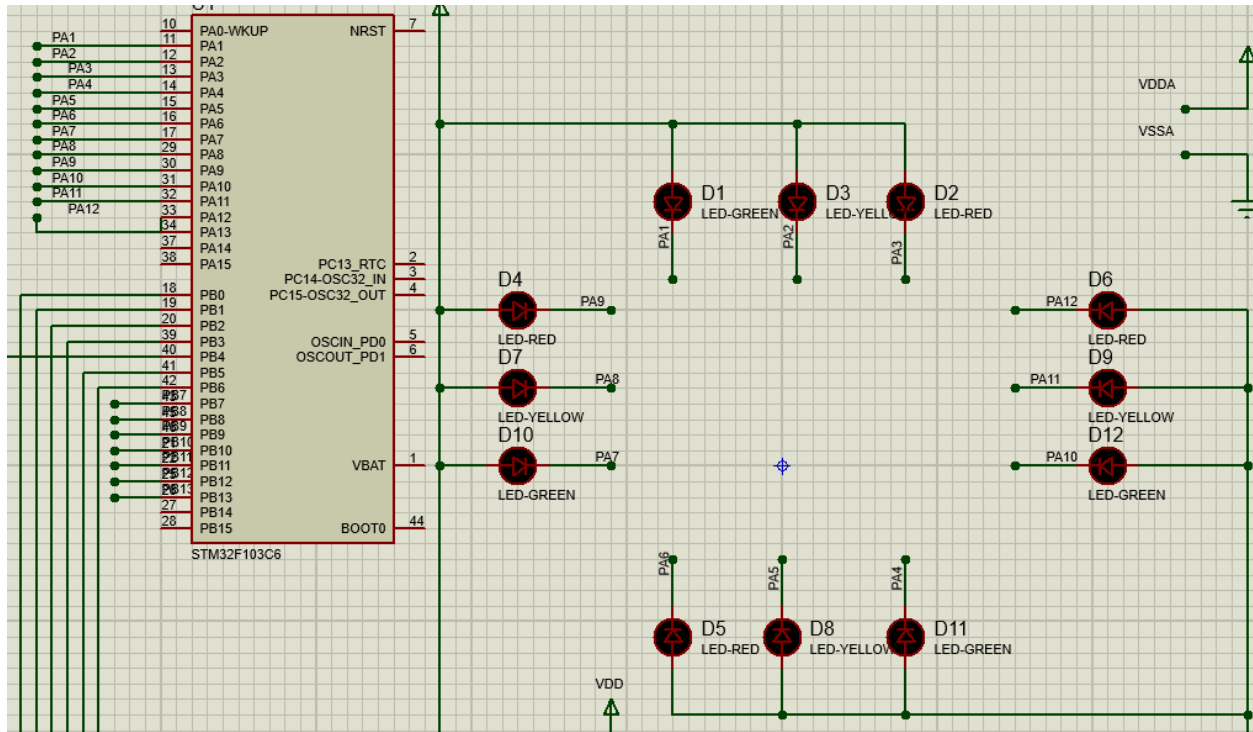
Report 1: Present the schematic



Report 2: Present the source code in while

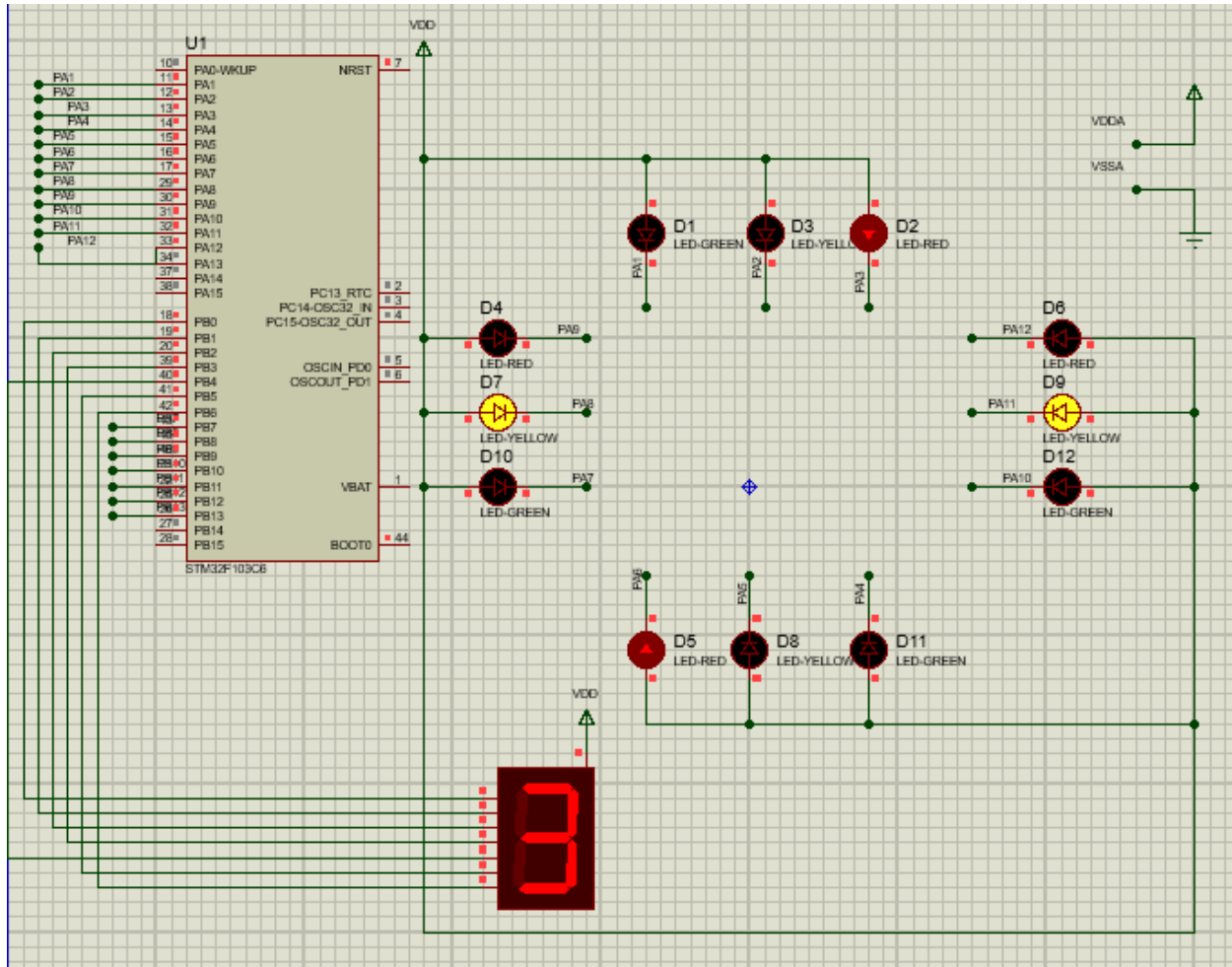
```
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
HAL_Delay(3000);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
HAL_Delay(2000);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
HAL_Delay(5000);
```

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



Exercise 4: Implement a function named `display7SEG(int num)`. The input for this function is from 0 to 9

Report 1: Present the schematic



Report 2: Present the source code for display7SEG function

```
void display7SEG (int x)
{
    switch (x)
    {
        case 0:
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_RESET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_RESET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_RESET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_SET);
            break;
        case 1:
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_RESET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_SET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_SET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_SET);
            HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_SET);
    }
}
```

[illegible]

```

case 8:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
break;
default:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
}
}

```

Exercise 5: Integrate the 7SEG-LED to the 4 way traffic light. In this case, the 7SEG-LED is used to display countdown value.

Source code:

```

void display7SEG(int x)
{
switch (x)
{
case 1:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_SET);
break;
case 2:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
break;
case 3:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);

```

```

break;
case 4:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
break;
default:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_1, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_3, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_4, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
}
}
void display7SEG_left_right(int x)
{
switch (x)
{
case 1:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_7, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_8, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_9, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_10, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_12, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_13, GPIO_PIN_SET);
break;
case 2:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_8, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_9, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_10, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_11, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_12, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_13, GPIO_PIN_RESET);
break;
case 3:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_8, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_9, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_10, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_12, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_13, GPIO_PIN_RESET);
break;
case 4:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_7, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_8, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_9, GPIO_PIN_RESET);

```



```

HAL_GPIO_WritePin (GPIOB, GPIO_PIN_10, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_12, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_13, GPIO_PIN_RESET);
break;
default:
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_8, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_9, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_10, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_12, GPIO_PIN_RESET);
HAL_GPIO_WritePin (GPIOB, GPIO_PIN_13, GPIO_PIN_RESET);
}
}
void traffic_led(int x)
{
if(x == 1 || x == 2 || x == 3)
{
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET); // green_left_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET); // yellow_left_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET); // red_left_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET); // green_right_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET); // yellow_right_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET); // red_right_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET); // red_up_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET); // yellow_up_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET); // green_up_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET); // red_down_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET); // yellow_down_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET); // green_down_off
}
else if(x == 4 || x == 5)
{
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET); // green_left_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET); // yellow_left_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET); // green_right_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET); // yellow_right_on
}
else if(x == 6 || x == 7)
{
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET); // red_up_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET); // green_up_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET); // red_down_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET); // green_down_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET); // yellow_left_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET); // red_left_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET); // yellow_left_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET); // red_left_on
}
else if(x == 8)
{
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET); // yellow_up_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET); // green_up_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET); // yellow_down_off

```

```

HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET); // green_down_on
}
else
{
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET); // yellow_up_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET); // green_up_off
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET); // yellow_down_on
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET); // green_down_off
}
}
int counter = 0;
int counter2 = 0;
int counter_rec = 0;
int counter_rec2 = 0;
int tmp = 1;
while (1)
{
    if(counter <= 0)
    {
        if(counter_rec == 0)
        {
            counter = 5;
        }
        else if(counter_rec == 1)
        {
            counter = 3;
        }
        else
        {
            counter = 2;
            counter_rec = -1;
        }
        counter_rec += 1;
    }
    if(counter2 <= 0)
    {
        if(counter_rec2 == 0)
        {
            counter2 = 3;
        }
        else if(counter_rec2 == 1)
        {
            counter2 = 2;
        }
        else
        {
            counter2 = 5;
            counter_rec2 = -1;
        }
        counter_rec2 += 1;
    }
    if(tmp >= 10)
    {
        tmp = 0;
    }
}

```

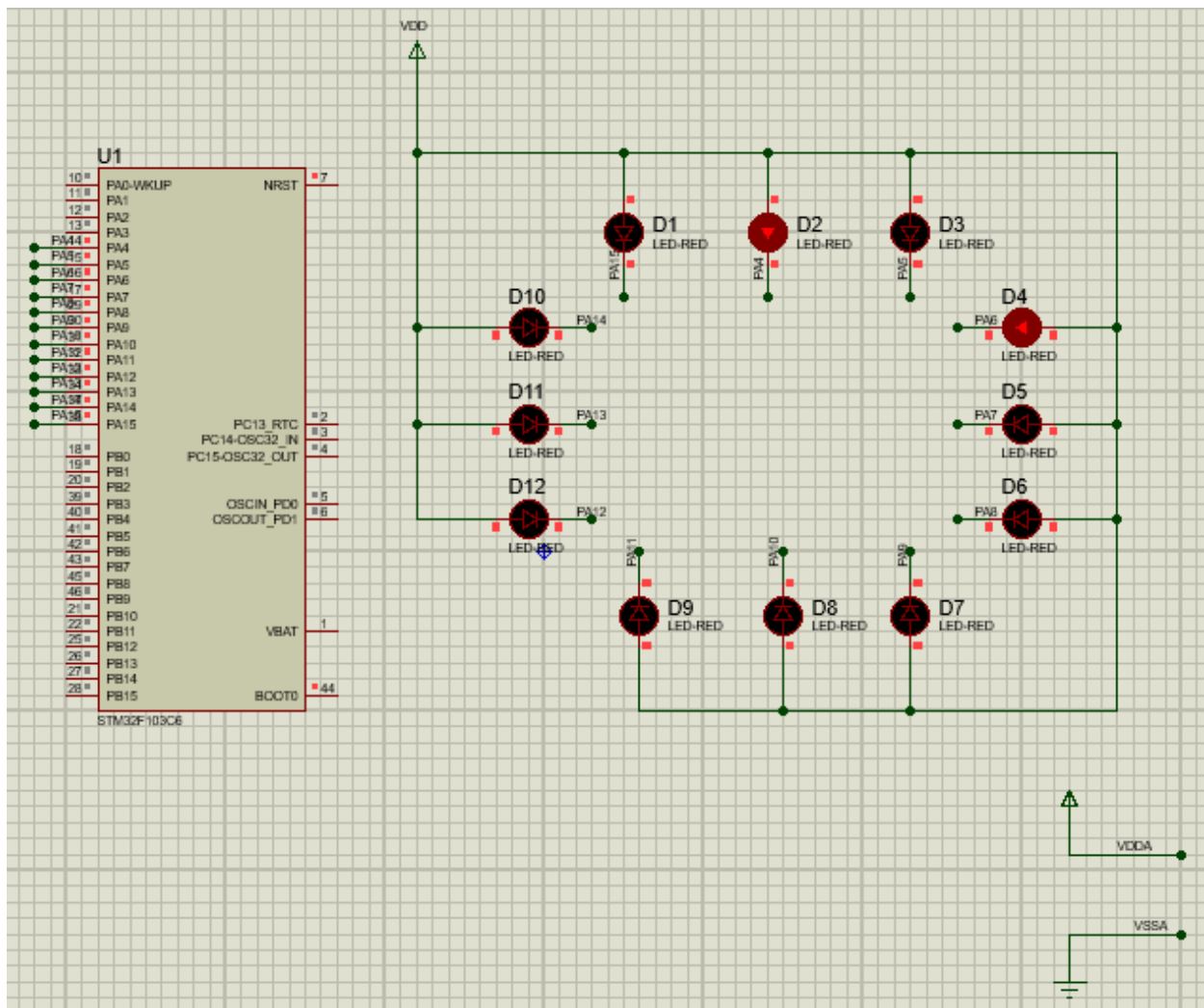
```

    display7SEG (counter--);
    display7SEG_left_right (counter2--);
    traffic_led(tmp++);
    HAL_Delay (1000);
}

```

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.

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.

```

void aclock(int x){
    switch(x){

case 12:
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);    //12h

```

[illegible]


```

        break;
    default:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
        //0h
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_SET);
    }
}
int tmp=0;
while (1)
{
    if(tmp>12){
        tmp=0;
    }
    aclock(tmp++);
    HAL_Delay (1000);
}

```

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, GPIO_PIN_4, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_SET);
}

```

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.


```

void setNumberOnClock(int n){
    switch(n){
        case 0:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
            break;
        case 1:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
            break;
        case 2:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
            break;
        case 3:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
            break;
        case 4:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET);
            break;
        case 5:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
            break;
        case 6:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
            break;
        case 7:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
            break;
        case 8:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
            break;
        case 9:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_RESET);
            break;
        case 10:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_RESET);
            break;
        default:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_RESET);

    }
}

```

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.

Source code:

```

void clearNumberOnClock(int n){
    switch(n){
        case 0:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
            break;
        case 1:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);

```



```

        break;
    case 2:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
        break;
    case 3:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
        break;
    case 4:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);
        break;
    case 5:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
        break;
    case 6:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
        break;
    case 7:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
        break;
    case 8:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
        break;
    case 9:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_SET);
        break;
    case 10:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_SET);
        break;
    default:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_SET);

}
}

```

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

Source code:

```

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

```

```

    }
    void setNumberOnClock(int n){
        switch(n){
            case 0:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
                break;
            case 1:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
                break;
            case 2:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
                break;
            case 3:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
                break;
            case 4:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET);
                break;
            case 5:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
                break;
            case 6:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
                break;
            case 7:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
                break;
            case 8:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
                break;
            case 9:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_RESET);
                break;
            case 10:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_RESET);
                break;
            default:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_RESET);

        }
    }

    void clearNumberOnClock(int n){
        switch(n){
            case 0:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
                break;
            case 1:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5,
GPIO_PIN_SET);
                break;
            case 2:
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6,
GPIO_PIN_SET);

```

```

        case 3:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7,
GPIO_PIN_SET);
            break;
        case 4:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8,
GPIO_PIN_SET);
            break;
        case 5:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9,
GPIO_PIN_SET);
            break;
        case 6:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10,
GPIO_PIN_SET);
            break;
        case 7:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11,
GPIO_PIN_SET);
            break;
        case 8:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12,
GPIO_PIN_SET);
            break;
        case 9:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13,
GPIO_PIN_SET);
            break;
        case 10:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14,
GPIO_PIN_SET);
            break;
        default:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15,
GPIO_PIN_SET);

```

```

    }
}
clearAllClock();
int sec=0;
int min=0;
int hour=0;
//int count_sec = 0;
//int count_min = 0;
setNumberOnClock(0);
HAL_Delay(1000); //HAL_Delay(5000);
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */
    sec++;
    if(sec>=12){
        sec=0;

```

```

        //count_sec++;
    }
    setNumberOnClock(sec);
    if(hour!=sec-1 && min!=sec-1){
        if(hour!=11 && min!=11 && sec==0){
            clearNumberOnClock(11);
        }
        if(sec !=0){
            clearNumberOnClock(sec-1);
        }
    }

    if(sec==0){
        //count_sec=0;
        min++;
        if(min>=12){
            min=0;
            //count_min++;
        }
        setNumberOnClock(min);
        if(hour!=min-1 && sec!=min-1){
            if(hour!=11 && sec!=11){
                clearNumberOnClock(11);
            }
            if(min!=0){
                clearNumberOnClock(min-1);
            }
        }
        if(min==0){
            hour++;
            if(hour>=12){
                hour=0;
            }
            setNumberOnClock(hour);
            if(min!=hour-1 && sec!=hour-1){
                if(min!=11 && sec!=11){
                    clearNumberOnClock(11);
                }
                if(hour!=0){
                    clearNumberOnClock(hour-1);
                }
            }
        }
    }
    HAL_Delay(1000);

    /* USER CODE BEGIN 3 */
}

```