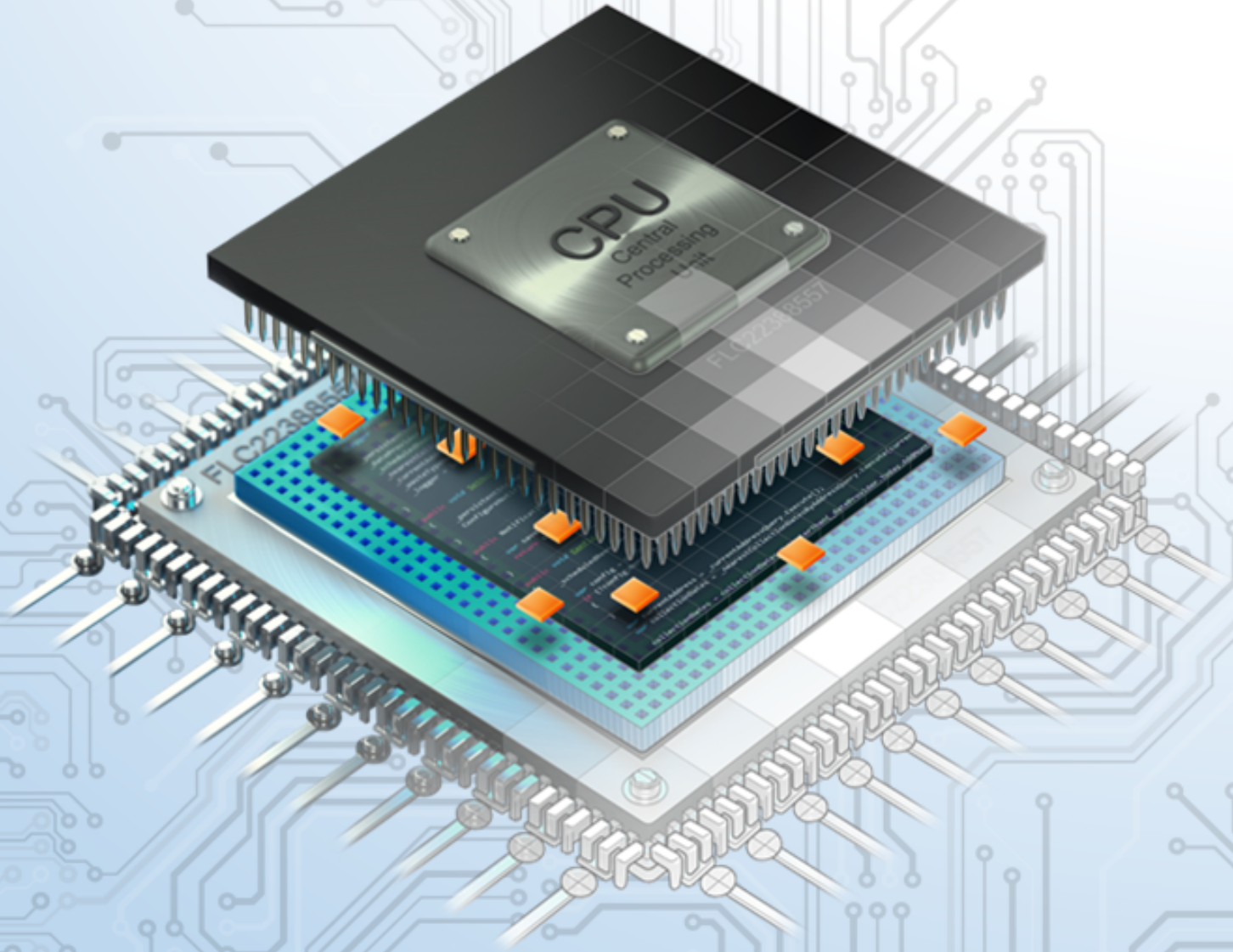
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller

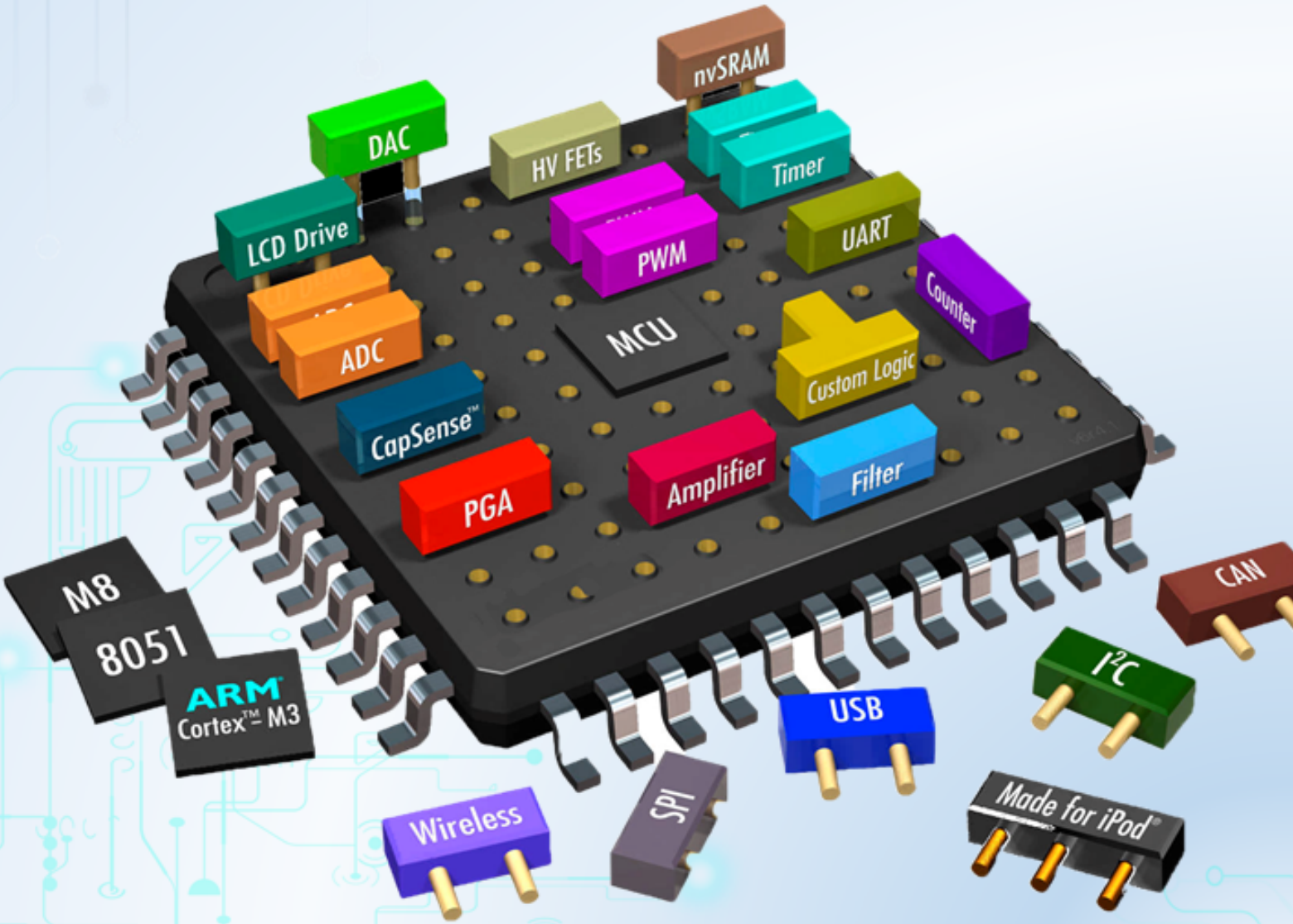


Dr. Le Trong Nhan

Contents

CHAPTER 1

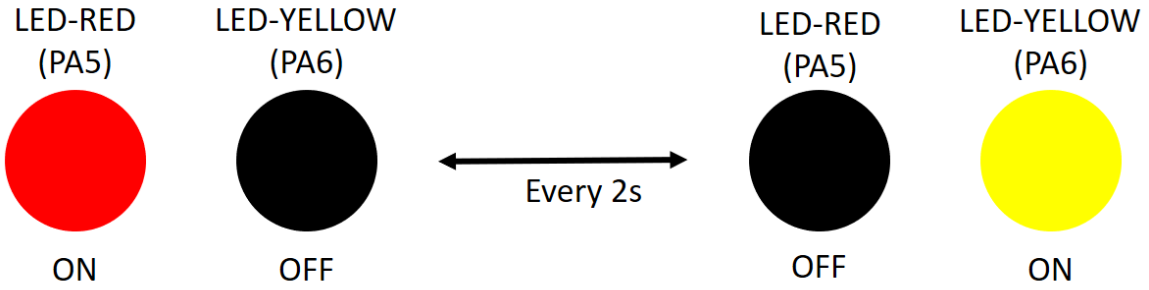
LED Animations



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.



Report 1: Depict the schematic from Proteus simulation in this report.

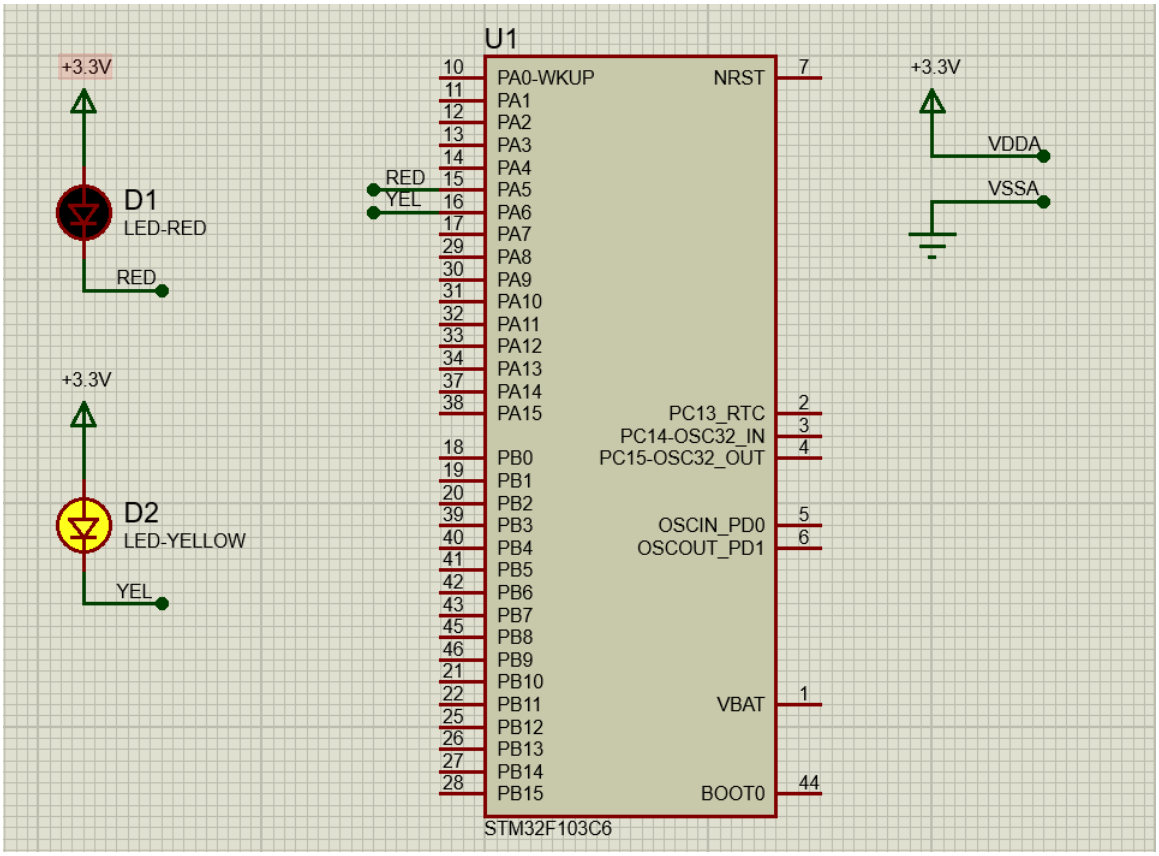


Figure 1.1: Schematic Exercise 1

Report 2: Present the source code in the infinite loop while.

```
int a=0;
while (1){
    //At the first state, red led on, yellow led off.
    if(a==0) HAL_GPIO_TogglePin(YEL_GPIO_Port, YEL_Pin);
    //Then, each led changes their state every 2 second
    else{
        HAL_GPIO_TogglePin(RED_GPIO_Port, RED_Pin);
        HAL_GPIO_TogglePin(YEL_GPIO_Port, YEL_Pin);
    }
    a++;
    HAL_Delay(2000); //Set cycle time to 2 seconds
}
```

Program 1.1: Source code exercise 1

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.

Report 1: Present the schematic.

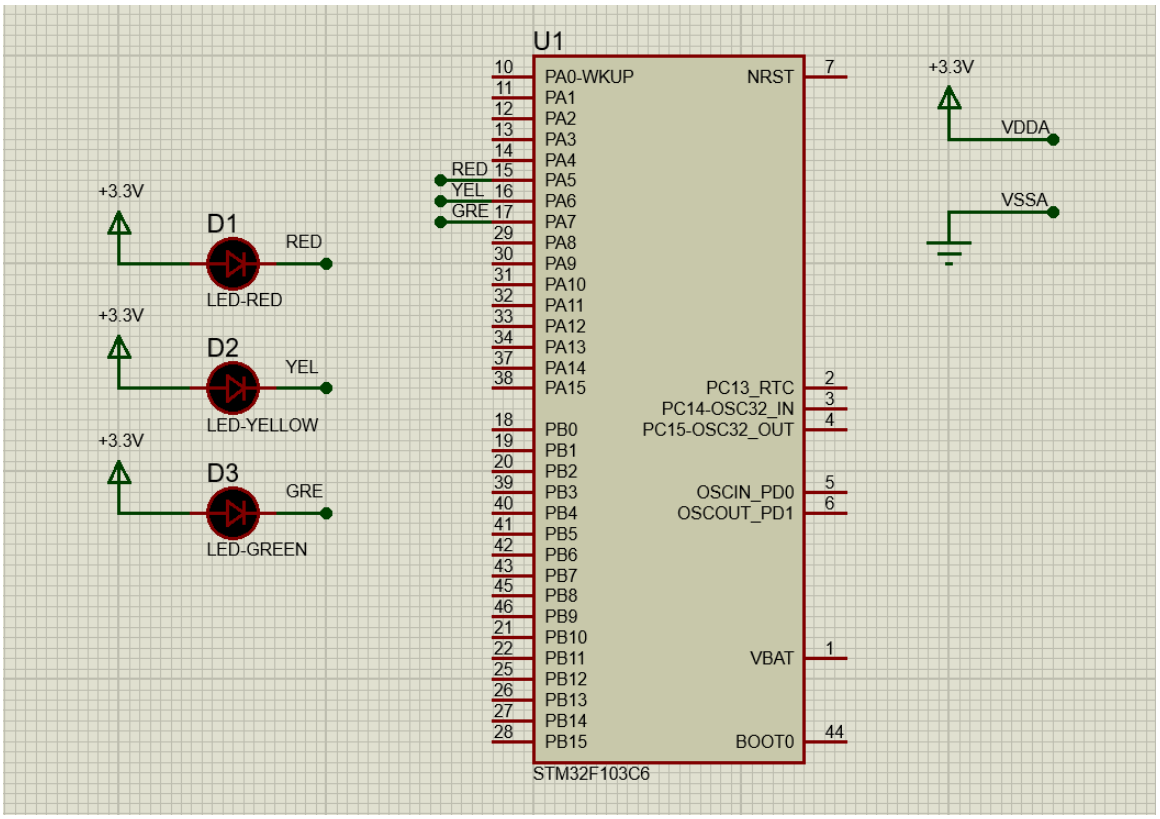


Figure 1.2: Schematic Exercise 2

Report 2: Present the source code in while.

```
int count=0;
while (1){
    //The order is Red(5s) -> Green(3s) -> Yellow(2s)
    //At first all leds are on, we need to turn off 2 leds
    //Then the next states, we need to turn off the current
    led and turn on the next led.
    //RED
    if(count==0){
        HAL_GPIO_TogglePin(YEL_GPIO_Port, YEL_Pin);
        HAL_GPIO_TogglePin(GRE_GPIO_Port, GRE_Pin);
    }
    //YEL
    if(count==5){
        HAL_GPIO_TogglePin(RED_GPIO_Port, RED_Pin);
        HAL_GPIO_TogglePin(YEL_GPIO_Port, YEL_Pin);
```

```
    }
    //GREEN
    if(count==7){
        HAL_GPIO_TogglePin(YEL_GPIO_Port, YEL_Pin);
        HAL_GPIO_TogglePin(GRE_GPIO_Port, GRE_Pin);
    }
    //RED
    if(count==10){
        HAL_GPIO_TogglePin(GRE_GPIO_Port, GRE_Pin);
        HAL_GPIO_TogglePin(RED_GPIO_Port, RED_Pin);
    }
    count++;
    if(count==15) count=5;
    HAL_Delay(1000); //Set cycle time to 1 second
    }
}
```

Program 1.2: Source code exercise 2

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.

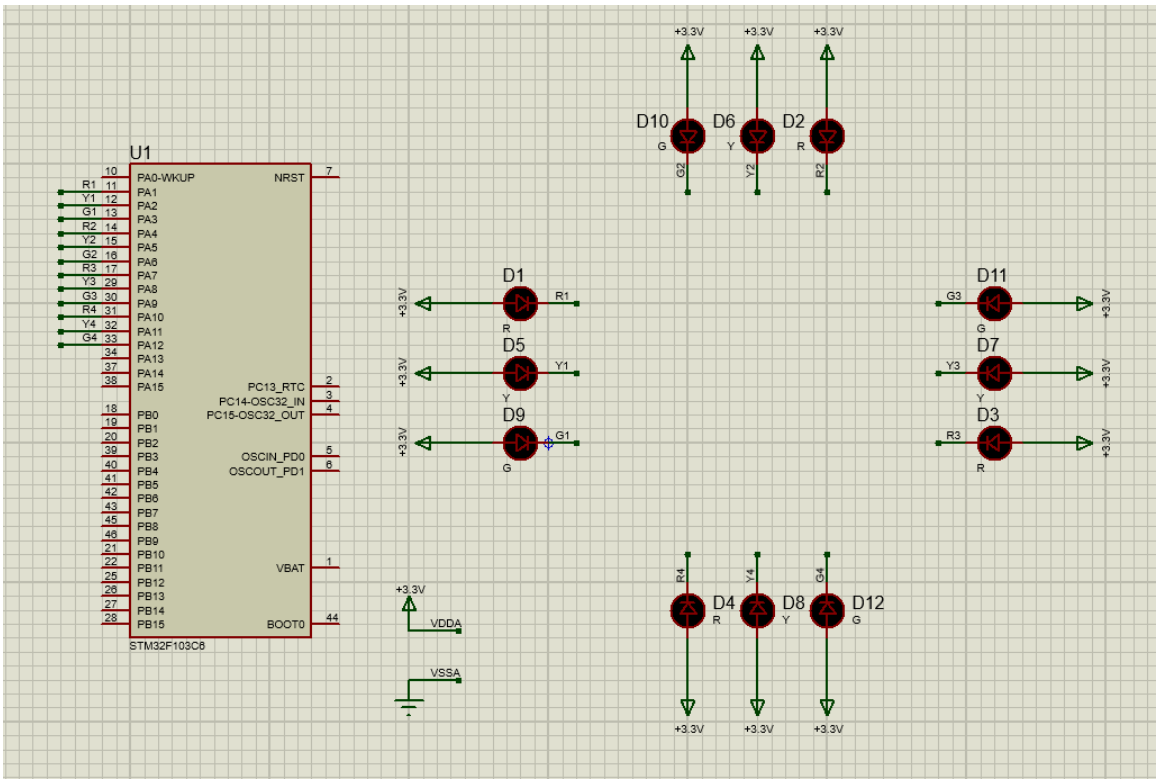


Figure 1.3: Reference design for a 4 way traffic light

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)**. 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.

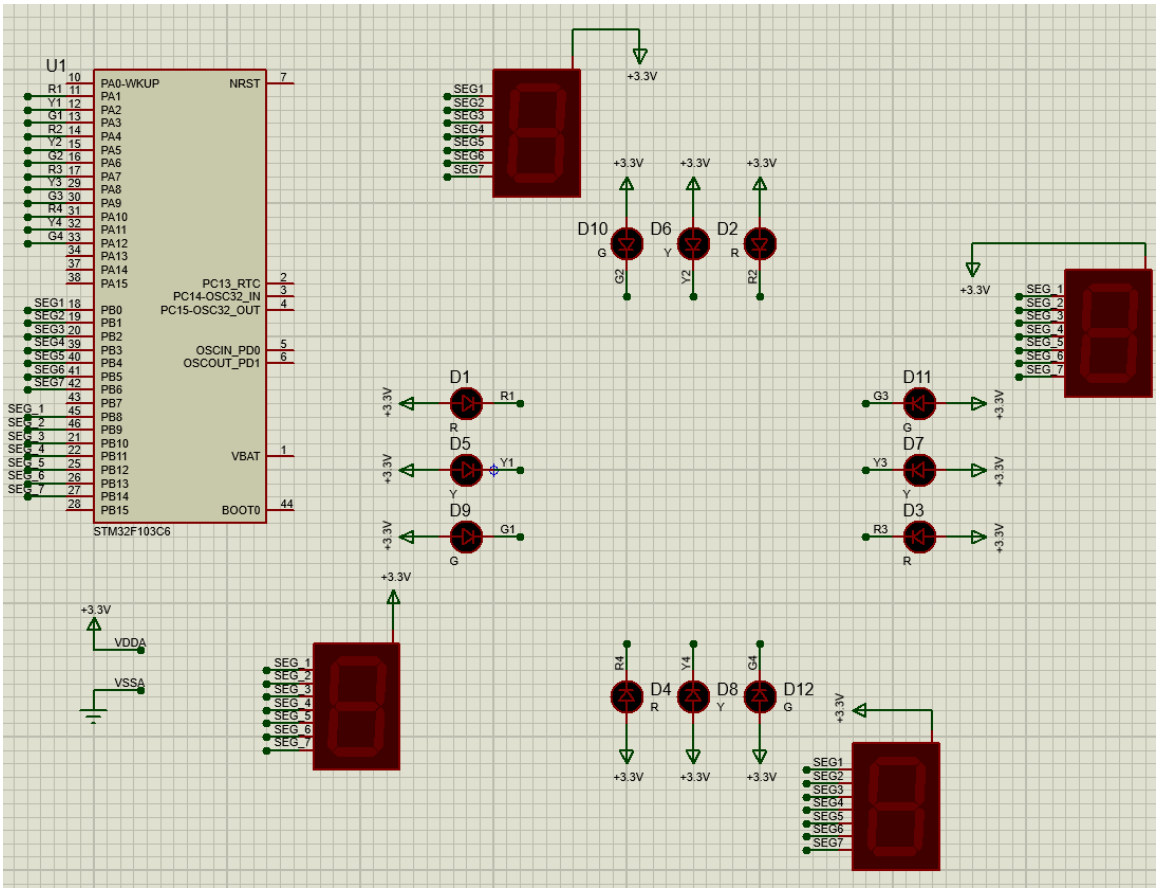


Figure 1.4: Display a number on 7 segment LED

Report 2: Present the source code for display7SEG function.

```
void display7SEG(int num){
//Make 10 case for 7SEG led to display from 0 to 9
  switch(num){
  case 0:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, RESET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, RESET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, RESET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, RESET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, RESET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, SET);
        break;
  case 1:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, SET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, RESET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, SET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, SET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, SET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, SET);
        break;
  case 2:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, RESET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, RESET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, SET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, RESET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, RESET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, SET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, RESET);
        break;
  case 3:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, RESET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, RESET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, RESET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, SET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, SET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, RESET);
        break;
  case 4:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, SET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, RESET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, SET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, SET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, RESET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, RESET);
        break;
```

```
  case 5:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, RESET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, SET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, RESET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, SET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, RESET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, RESET);
        break;
  case 6:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, RESET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, SET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, RESET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, RESET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, RESET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, RESET);
        break;
  case 7:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, RESET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, RESET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, SET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, SET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, SET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, SET);
        break;
  case 8:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, RESET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, RESET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, RESET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, RESET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, RESET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, RESET);
        break;
  case 9:
        HAL_GPIO_WritePin(SEG1_GPIO_Port, SEG1_Pin, RESET);
        HAL_GPIO_WritePin(SEG2_GPIO_Port, SEG2_Pin, RESET);
        HAL_GPIO_WritePin(SEG3_GPIO_Port, SEG3_Pin, RESET);
        HAL_GPIO_WritePin(SEG4_GPIO_Port, SEG4_Pin, RESET);
        HAL_GPIO_WritePin(SEG5_GPIO_Port, SEG5_Pin, SET);
        HAL_GPIO_WritePin(SEG6_GPIO_Port, SEG6_Pin, RESET);
        HAL_GPIO_WritePin(SEG7_GPIO_Port, SEG7_Pin, RESET);
        break;
  }
}
```

Program 1.4: Source code exercise 4

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.

```
//In crossroads, there 2 couple of traffic light
int count1=0; //Count for Led 1,3
int count2=0; //Count for Led 2,4
int count1_1=0; //Count for 7-SEG 1,3
int count2_1=0; //Count for 7-SEG 2,4
while (1){
//TRAFFIC LIGHT
//State 0 (TL1-3: Green, TL2-4:Red)
    if(count1==0){
      //GRE
        HAL_GPIO_TogglePin(Y1_GPIO_Port, Y1_Pin);
        HAL_GPIO_TogglePin(R1_GPIO_Port, R1_Pin);
        HAL_GPIO_TogglePin(Y3_GPIO_Port, Y3_Pin);
        HAL_GPIO_TogglePin(R3_GPIO_Port, R3_Pin);
        count1_1=3;
    }
    if(count2==0){
        //RED
        HAL_GPIO_TogglePin(Y2_GPIO_Port, Y2_Pin);
        HAL_GPIO_TogglePin(G2_GPIO_Port, G2_Pin);
        HAL_GPIO_TogglePin(Y4_GPIO_Port, Y4_Pin);
        HAL_GPIO_TogglePin(G4_GPIO_Port, G4_Pin);
        count2_1=5;
    }
//State 1 (TL1-3:Yellow, TL2-4:Green)
    if(count1==3){
      //YEL
        HAL_GPIO_TogglePin(G1_GPIO_Port, G1_Pin);
        HAL_GPIO_TogglePin(Y1_GPIO_Port, Y1_Pin);
        HAL_GPIO_TogglePin(G3_GPIO_Port, G3_Pin);
        HAL_GPIO_TogglePin(Y3_GPIO_Port, Y3_Pin);
        count1_1=2;
    }
    if(count2==5){
        //GRE
        HAL_GPIO_TogglePin(R2_GPIO_Port, R2_Pin);
        HAL_GPIO_TogglePin(G2_GPIO_Port, G2_Pin);
        HAL_GPIO_TogglePin(R4_GPIO_Port, R4_Pin);
        HAL_GPIO_TogglePin(G4_GPIO_Port, G4_Pin);
        count2_1=3;
    }
```

```
//State 2 (TL1-3:Red, TL2-4:Yellow)
    if(count1==5){
    //RED
      HAL_GPIO_TogglePin(Y1_GPIO_Port, Y1_Pin);
      HAL_GPIO_TogglePin(R1_GPIO_Port, R1_Pin);
      HAL_GPIO_TogglePin(Y3_GPIO_Port, Y3_Pin);
        HAL_GPIO_TogglePin(R3_GPIO_Port, R3_Pin);
        count1_1=5;
    }
    if(count2==8){
    //YEL
      HAL_GPIO_TogglePin(Y2_GPIO_Port, Y2_Pin);
      HAL_GPIO_TogglePin(G2_GPIO_Port, G2_Pin);
      HAL_GPIO_TogglePin(Y4_GPIO_Port, Y4_Pin);
      HAL_GPIO_TogglePin(G4_GPIO_Port, G4_Pin);
      count2_1=2;
    }
//State 3 (TL1-3:Green, TL2-4:Red)
    if(count1==10){
        //GRE
        HAL_GPIO_TogglePin(G1_GPIO_Port, G1_Pin);
        HAL_GPIO_TogglePin(R1_GPIO_Port, R1_Pin);
        HAL_GPIO_TogglePin(G3_GPIO_Port, G3_Pin);
        HAL_GPIO_TogglePin(R3_GPIO_Port, R3_Pin);
        count1_1=3;
    }
    if(count2==10){
        //RED
        HAL_GPIO_TogglePin(Y2_GPIO_Port, Y2_Pin);
        HAL_GPIO_TogglePin(R2_GPIO_Port, R2_Pin);
        HAL_GPIO_TogglePin(Y4_GPIO_Port, Y4_Pin);
        HAL_GPIO_TogglePin(R4_GPIO_Port, R4_Pin);
        count2_1=5;
    }
    display7_SEG(count1_1);
    display7SEG(count2_1);
//Setting counter variable
//Count up "traffic light" led
    count1++;
    count2++;
    if(count1>=13)count1=3;
    if(count2>=15)count2=5;
//Count down "7SEG" led
    count1_1--;
    count2_1--;
    HAL_Delay(1000); //Set cycle time to 1 second for real-
   time display
 }
```

Program 1.5: Source code exercise 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.

Report 1: Present the schematic.

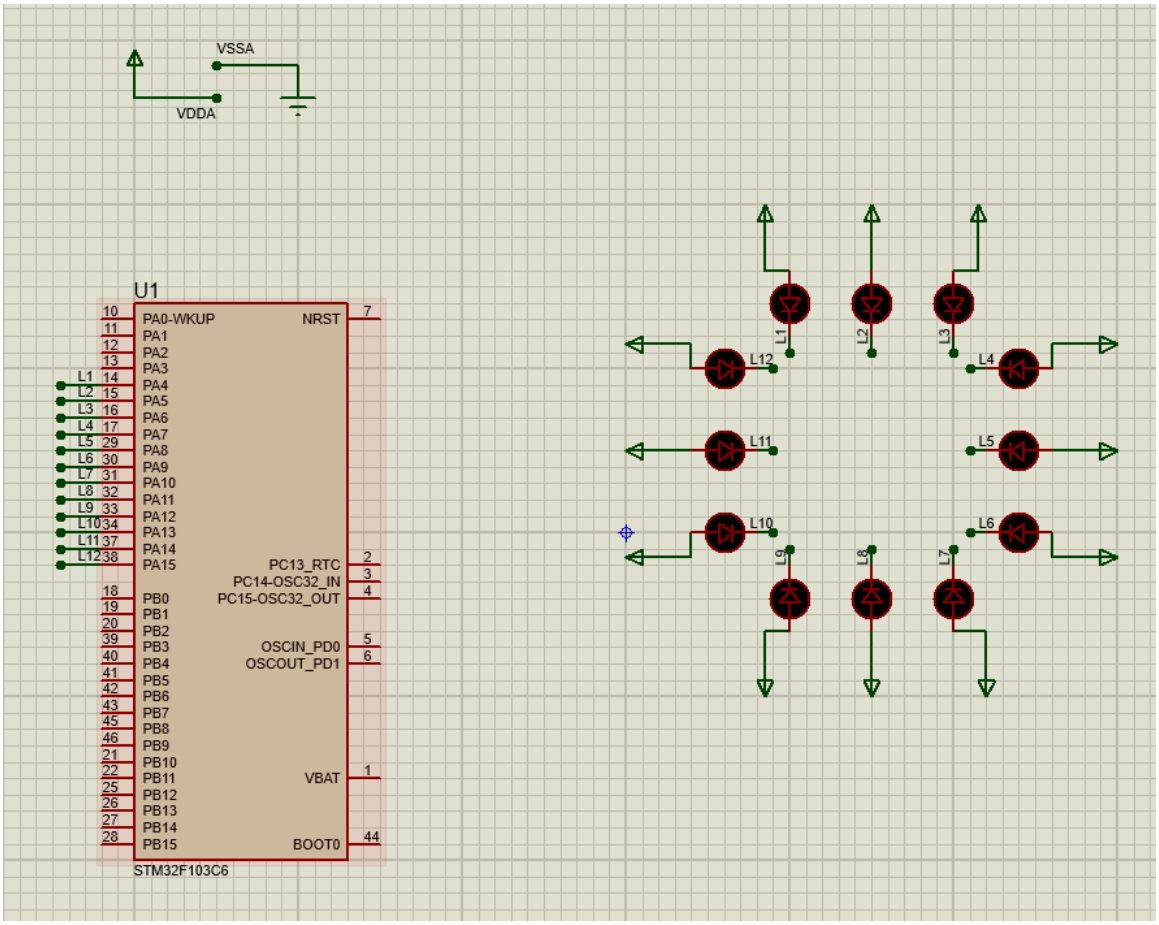


Figure 1.5: Schematic exercise 6

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

```
//Implement "Blink" function to make 12 blink continously
void Blink1(int count){
  if(count==1){HAL_GPIO_TogglePin(L1_GPIO_Port, L1_Pin);}
  if(count==2){HAL_GPIO_TogglePin(L2_GPIO_Port, L2_Pin);}
  if(count==3){HAL_GPIO_TogglePin(L3_GPIO_Port, L3_Pin);}
  if(count==4){HAL_GPIO_TogglePin(L4_GPIO_Port, L4_Pin);}
  if(count==5){HAL_GPIO_TogglePin(L5_GPIO_Port, L5_Pin);}
  if(count==6){HAL_GPIO_TogglePin(L6_GPIO_Port, L6_Pin);}
  if(count==7){HAL_GPIO_TogglePin(L7_GPIO_Port, L7_Pin);}
  if(count==8){HAL_GPIO_TogglePin(L8_GPIO_Port, L8_Pin);}
  if(count==9){HAL_GPIO_TogglePin(L9_GPIO_Port, L9_Pin);}
  if(count==10){HAL_GPIO_TogglePin(L10_GPIO_Port, L10_Pin)
   ;}
```

```
  if(count==11){HAL_GPIO_TogglePin(L11_GPIO_Port, L11_Pin);}
  if(count==12){HAL_GPIO_TogglePin(L12_GPIO_Port, L12_Pin);}
}
//Infinite loop while
int a=0;
while (1){
    Blink1(a++);
    if(a>12) a=0;
    HAL_Delay(50);
}
```

Program 1.6: Source code exercise 6

1.7 Exercise 7

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

```
//Set all pin at level 0
void clearAllClock() {
  HAL_GPIO_WritePin(L1_GPIO_Port, L1_Pin, SET);
  HAL_GPIO_WritePin(L2_GPIO_Port, L2_Pin, SET);
  HAL_GPIO_WritePin(L3_GPIO_Port, L3_Pin, SET);
  HAL_GPIO_WritePin(L4_GPIO_Port, L4_Pin, SET);
  HAL_GPIO_WritePin(L5_GPIO_Port, L5_Pin, SET);
  HAL_GPIO_WritePin(L6_GPIO_Port, L6_Pin, SET);
  HAL_GPIO_WritePin(L7_GPIO_Port, L7_Pin, SET);
  HAL_GPIO_WritePin(L8_GPIO_Port, L8_Pin, SET);
  HAL_GPIO_WritePin(L9_GPIO_Port, L9_Pin, SET);
  HAL_GPIO_WritePin(L10_GPIO_Port, L10_Pin, SET);
  HAL_GPIO_WritePin(L11_GPIO_Port, L11_Pin, SET);
  HAL_GPIO_WritePin(L12_GPIO_Port, L12_Pin, SET);
}
```

Program 1.7: Source code exercise 7

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.

```
void setNumberOnClock(int num){
  switch(num){
  case 0:
    HAL_GPIO_WritePin(L2_GPIO_Port, L2_Pin, RESET); break;
  case 1:
    HAL_GPIO_WritePin(L3_GPIO_Port, L3_Pin, RESET); break;
  case 2:
    HAL_GPIO_WritePin(L4_GPIO_Port, L4_Pin, RESET); break;
  case 3:
    HAL_GPIO_WritePin(L5_GPIO_Port, L5_Pin, RESET); break;
  case 4:
    HAL_GPIO_WritePin(L6_GPIO_Port, L6_Pin, RESET); break;
  case 5:
    HAL_GPIO_WritePin(L7_GPIO_Port, L7_Pin, RESET); break;
  case 6:
    HAL_GPIO_WritePin(L8_GPIO_Port, L8_Pin, RESET); break;
  case 7:
    HAL_GPIO_WritePin(L9_GPIO_Port, L9_Pin, RESET); break;
  case 8:
    HAL_GPIO_WritePin(L10_GPIO_Port, L10_Pin, RESET); break
   ;
  case 9:
    HAL_GPIO_WritePin(L11_GPIO_Port, L11_Pin, RESET); break
   ;
  case 10:
    HAL_GPIO_WritePin(L12_GPIO_Port, L12_Pin, RESET); break
   ;
  case 11:
    HAL_GPIO_WritePin(L1_GPIO_Port, L1_Pin, RESET); break;
  }
}
```

Program 1.8: Source code exercise 8

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.

```
void clearNumberOnClock(int num){
  switch(num){
  case 0:
    HAL_GPIO_WritePin(L2_GPIO_Port, L2_Pin, SET); break;
  case 1:
    HAL_GPIO_WritePin(L3_GPIO_Port, L3_Pin, SET); break;
  case 2:
    HAL_GPIO_WritePin(L4_GPIO_Port, L4_Pin, SET); break;
  case 3:
    HAL_GPIO_WritePin(L5_GPIO_Port, L5_Pin, SET); break;
  case 4:
    HAL_GPIO_WritePin(L6_GPIO_Port, L6_Pin, SET); break;
  case 5:
    HAL_GPIO_WritePin(L7_GPIO_Port, L7_Pin, SET); break;
  case 6:
    HAL_GPIO_WritePin(L8_GPIO_Port, L8_Pin, SET); break;
  case 7:
    HAL_GPIO_WritePin(L9_GPIO_Port, L9_Pin, SET); break;
  case 8:
    HAL_GPIO_WritePin(L10_GPIO_Port, L10_Pin, SET); break;
  case 9:
    HAL_GPIO_WritePin(L11_GPIO_Port, L11_Pin, SET); break;
  case 10:
    HAL_GPIO_WritePin(L12_GPIO_Port, L12_Pin, SET); break;
  case 11:
    HAL_GPIO_WritePin(L1_GPIO_Port, L1_Pin, SET); break;
  }
}
```

Program 1.9: Source code exercise 9

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.

```
// Clock start at 00:00:00
int h=0;
int m=0;
int s=0;
while (1){
//1 led represents for 5 second/ 5 minute so i display the integer part of s:5 and m:5
//The second hand need to run 5 cycles to count up the minute hand by 1
  setNumberOnClock(h);
  setNumberOnClock(m/5);
  setNumberOnClock(s/5);
  s++;
  HAL_Delay(1000); //Set cycle time to 1 second for real-time display
  clearNumberOnClock((s-1)/5); //Clear the last state of "seconds'

  if(s>59){
    s=0;
    clearNumberOnClock(m/5);
    m++;
    setNumberOnClock(m/5);
  }
  if(m>59){
    m=0;
    clearNumberOnClock(h);
    h++;
    setNumberOnClock(h);
  }
  if(h>11){
    clearNumberOnClock(h);
    h=0;
    setNumberOnClock(h);
  }
```

Program 1.10: Source code exercise 10

link github lab 1: `https://github.com/LongSmallHand/VXL_Lab1.git`

END