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

Vi xử lý - Vi điều khiển (TN)

Lab 1: Led Animation

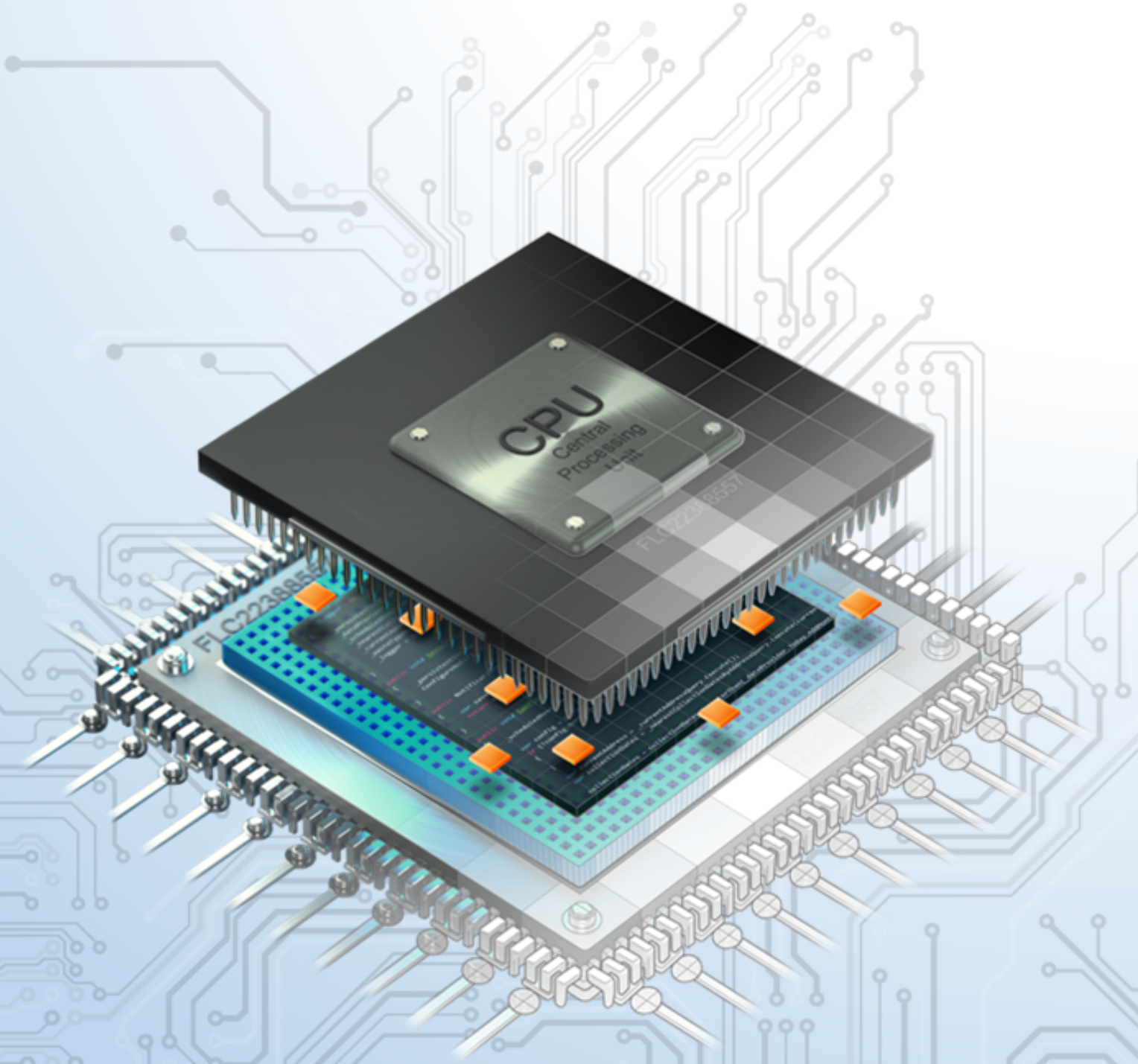
Giảng viên hướng dẫn: thầy Tôn Huỳnh Long
Nhóm lớp: L02

Họ và tên	**MSSV**
Phạm Trần Đức Hạnh	2310896

Thành phố Hồ Chí Minh, 9/2025

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller

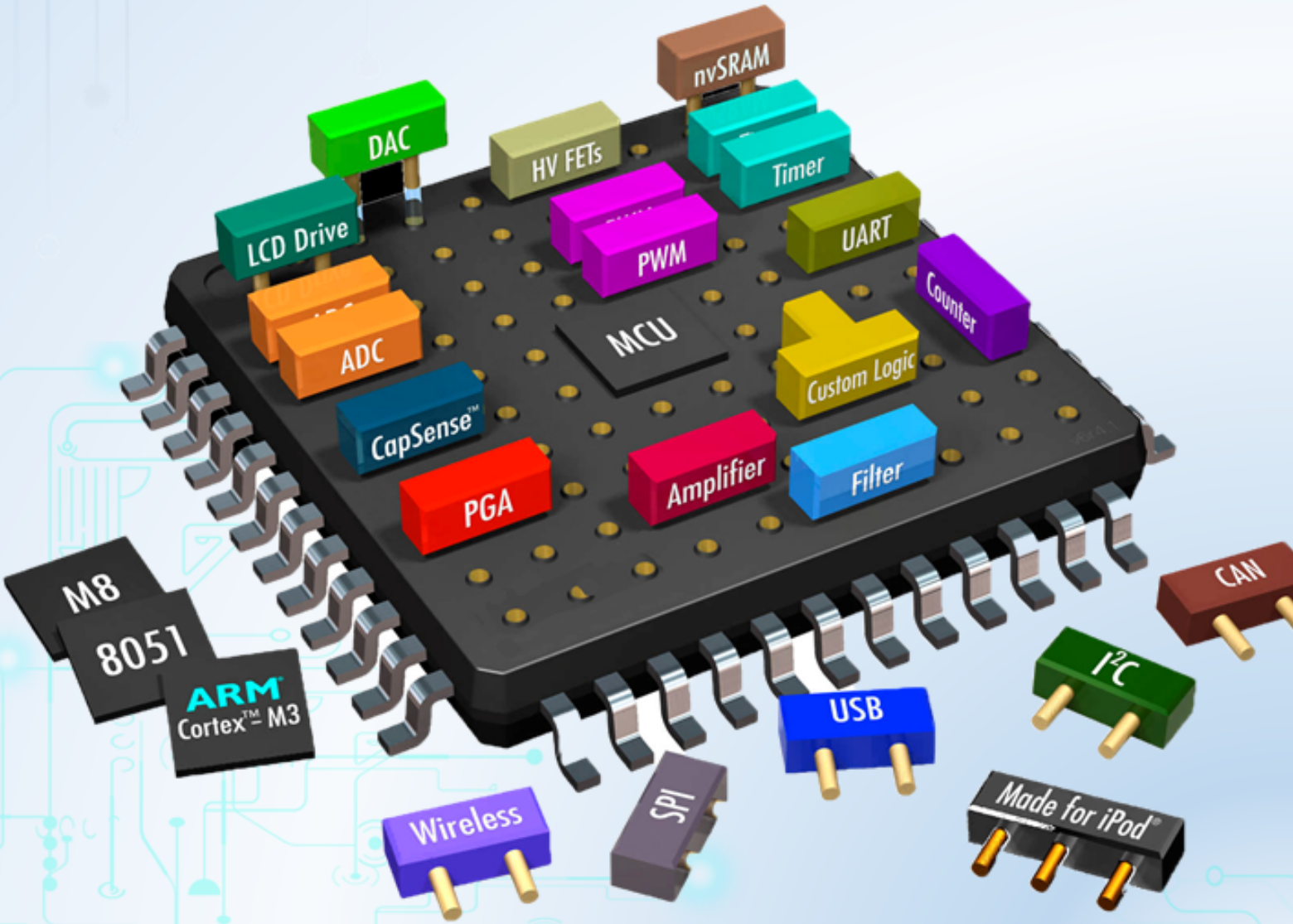


Dr. Le Trong Nhan

Mục lục

CHƯƠNG 1

LED Animations



1 Github link and video demos

1.1 Github repository

The Github repository for this project is kept up-to-date and contains all relevant source code and resources. In the `README.md` file, you will find comprehensive information including:

- The `.ioc` configuration image for STM32CubeMX.
- An explanation of the project idea and implementation.
- The structure of the `Src` directory and main source files.
- Detailed instructions on how to build and run the demo code.

`https://github.com/PhamTranDucHanh/MCU_LAB1_2310896`

1.2 Demo video (Google Drive link)

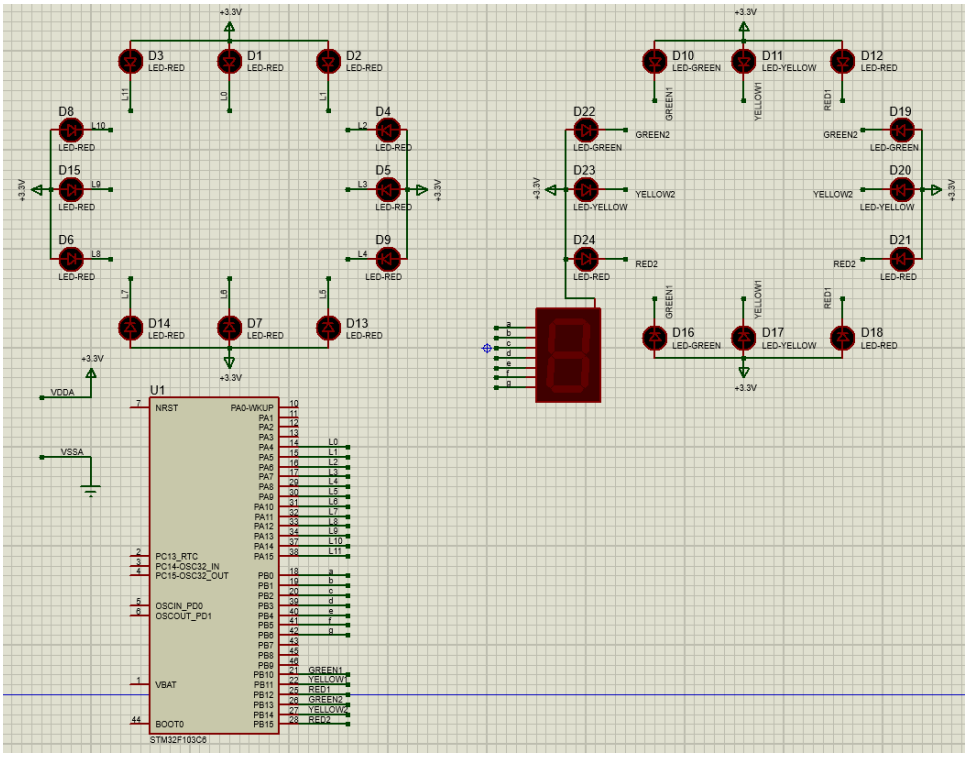
A screen-recorded demo video of the project running is provided via the following Google Drive link. The video demonstrates the execution of all required tasks and features. Please refer to the attached link for visual verification and step-by-step demonstration of the project in action.

`https://drive.google.com/drive/folders/1PykEqH20q_WXJJ6iM6BcTcNUmLXYFylr?usp=drive_link`

2 Exercise and Report

2.1 Proteus schematic (for all 10 exercises)

The schematic below is fully pre-wired and designed to support all 10 exercises in this lab. It provides the necessary connections for the STM32 microcontroller and peripheral components, allowing each exercise to be tested without modifying the circuit.



2.2 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.

Report: Source code in the infinite loop while of my project:

```
  HAL_GPIO_WritePin(GPIOB, RED1_Pin, 0);
  HAL_GPIO_WritePin(GPIOB, YELLOW1_Pin, 1);
  int cnt = 0;
  while(1){
    if (cnt == 2){
      HAL_GPIO_TogglePin(GPIOB, RED1_Pin);
      HAL_GPIO_TogglePin(GPIOB, YELLOW1_Pin);
      cnt = 0;
    }
    cnt++;
    HAL_Delay(1000);
  }
```

2.3 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: Present the source code in while.

```
int cnt1 = 0;

HAL_GPIO_WritePin(GPIOB, GREEN1_Pin, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOB, YELLOW1_Pin, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOB, RED1_Pin, GPIO_PIN_SET);
while (1)
{
  if (cnt1 == 3){
     HAL_GPIO_WritePin(GPIOB, GREEN1_Pin, GPIO_PIN_SET);
     HAL_GPIO_WritePin(GPIOB, YELLOW1_Pin, GPIO_PIN_RESET);
  }
  if (cnt1 == 5){
      HAL_GPIO_WritePin(GPIOB, YELLOW1_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, RED1_Pin, GPIO_PIN_RESET);
  }
  if (cnt1 == 10){
    HAL_GPIO_WritePin(GPIOB, RED1_Pin, GPIO_PIN_SET);
   HAL_GPIO_WritePin(GPIOB, GREEN1_Pin, GPIO_PIN_RESET);
    cnt1 = 0;
  }
  cnt1++;
  HAL_Delay(1000);
}
```

2.4 Exercise 3

Extend to the 4-way traffic light. Arrange 12 LEDs in a nice shape to simulate the behaviors of a traffic light.

Report: Source code.

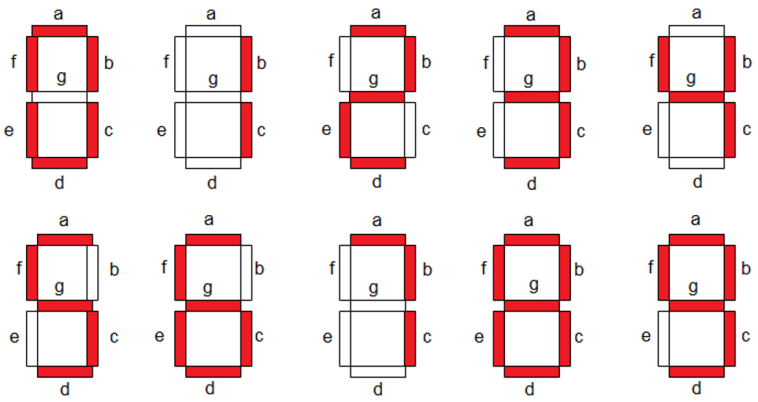
```
int cnt1 = 0;
int cnt2 = 0;

HAL_GPIO_WritePin(GPIOA, GREEN1_Pin, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOA, YELLOW1_Pin, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, RED1_Pin, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GREEN2_Pin, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, YELLOW2_Pin, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, RED2_Pin, GPIO_PIN_RESET);
while (1)
{
 if (cnt1 == 3){
   HAL_GPIO_WritePin(GPIOA, GREEN1_Pin, GPIO_PIN_SET);
   HAL_GPIO_WritePin(GPIOA, YELLOW1_Pin, GPIO_PIN_RESET);
 }
 if (cnt1 == 5){
   HAL_GPIO_WritePin(GPIOA, YELLOW1_Pin, GPIO_PIN_SET);
   HAL_GPIO_WritePin(GPIOA, RED1_Pin, GPIO_PIN_RESET);
 }
 if (cnt1 == 10){
   HAL_GPIO_WritePin(GPIOA, RED1_Pin, GPIO_PIN_SET);
   HAL_GPIO_WritePin(GPIOA, GREEN1_Pin, GPIO_PIN_RESET);
   cnt1 = 0;
 }
 if(cnt2 == 5){
   HAL_GPIO_WritePin(GPIOA, RED2_Pin, GPIO_PIN_SET);
   HAL_GPIO_WritePin(GPIOA, GREEN2_Pin, GPIO_PIN_RESET);
 }
 if(cnt2 == 8){
   HAL_GPIO_WritePin(GPIOA, GREEN2_Pin, GPIO_PIN_SET);
   HAL_GPIO_WritePin(GPIOA, YELLOW2_Pin, GPIO_PIN_RESET);
 }
 if(cnt2 == 10){
   HAL_GPIO_WritePin(GPIOA, YELLOW2_Pin, GPIO_PIN_SET);
   HAL_GPIO_WritePin(GPIOA, RED2_Pin, GPIO_PIN_RESET);
   cnt2 = 0;
 }
 HAL_Delay(1000);
 cnt1++;
 cnt2++;
}
```

2.5 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 1.1: *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 1.1: An example for your source code

Report: Present the source code for display7SEG function.

```
  uint16_t segPins[7] = {GPIO_PIN_0, GPIO_PIN_1, GPIO_PIN_2
   , GPIO_PIN_3, GPIO_PIN_4, GPIO_PIN_5, GPIO_PIN_6};

  const uint8_t digitCode[10][7] = {
    {0,0,0,0,0,0,1}, // 0
    {1,0,0,1,1,1,1}, // 1
    {0,0,1,0,0,1,0}, // 2
    {0,0,0,0,1,1,0}, // 3
    {1,0,0,1,1,0,0}, // 4
    {0,1,0,0,1,0,0}, // 5
    {0,1,0,0,0,0,0}, // 6
```

```
    {0,0,0,1,1,1,1}, // 7
    {0,0,0,0,0,0,0}, // 8
    {0,0,0,0,1,0,0}  // 9
  };

  void display7SEG(int num) {
    if (num > 9) return;
    for (int i = 0; i < 7; i++) {
      HAL_GPIO_WritePin(GPIOB, segPins[i], digitCode[num][i
   ] ? GPIO_PIN_SET : GPIO_PIN_RESET);
    }
  }
```

2.6 Exercise 5

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

Report: Source code (function *display7SEG(int num)* is from exercise 4).

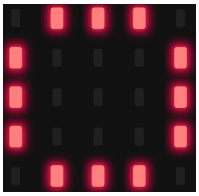
```
  int cnt1 = 0;
  int cnt2 = 0;

  HAL_GPIO_WritePin(GPIOA, GREEN1_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, YELLOW1_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA, RED1_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA, GREEN2_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA, YELLOW2_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA, RED2_Pin, GPIO_PIN_RESET);
  int counter = 3;
  while (1)
  {
   if (cnt1 == 3){
     HAL_GPIO_WritePin(GPIOA, GREEN1_Pin, GPIO_PIN_SET);
     HAL_GPIO_WritePin(GPIOA, YELLOW1_Pin, GPIO_PIN_RESET);
     counter = 2;
   }
   if (cnt1 == 5){
     HAL_GPIO_WritePin(GPIOA, YELLOW1_Pin, GPIO_PIN_SET);
     HAL_GPIO_WritePin(GPIOA, RED1_Pin, GPIO_PIN_RESET);
     counter = 5;
   }
   if (cnt1 == 10){
     HAL_GPIO_WritePin(GPIOA, RED1_Pin, GPIO_PIN_SET);
     HAL_GPIO_WritePin(GPIOA, GREEN1_Pin, GPIO_PIN_RESET);
     cnt1 = 0;
     counter = 3;
   }
```

```
    if(cnt2 == 5){
      HAL_GPIO_WritePin(GPIOA, RED2_Pin, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOA, GREEN2_Pin, GPIO_PIN_RESET);
    }
    if(cnt2 == 8){
      HAL_GPIO_WritePin(GPIOA, GREEN2_Pin, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOA, YELLOW2_Pin, GPIO_PIN_RESET);
    }
    if(cnt2 == 10){
      HAL_GPIO_WritePin(GPIOA, YELLOW2_Pin, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOA, RED2_Pin, GPIO_PIN_RESET);
      cnt2 = 0;
    }
    display7SEG(counter);
    HAL_Delay(1000);
    cnt1++;
    cnt2++;
    counter--;
  }
```

2.7 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 1.2: 12 LEDs for an analog clock

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

```
uint16_t clockPins[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};
for (int i = 0; i < 12; i++){
  HAL_GPIO_WritePin(GPIOA, clockPins[i], GPIO_PIN_SET);
}
int cnt = 0;
while (1)
{
if (cnt >= 12) cnt = 0;
HAL_GPIO_TogglePin(GPIOA, clockPins[cnt]);
```

```
  cnt++;
  HAL_Delay(500);
  }
```

Code explaination: This code sequentially turns on each LED connected to pins 0 to 11 of port A, with each LED being activated every 0.5 seconds. After reaching the last LED, the process repeats in a loop, and each time through the loop, the code toggles (switches the state of) each LED in order, creating a running light effect.

2.8 Exercise 7

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

```
uint16_t clockPins[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};

void clearAllClock(){
  for (int i = 0; i < 12; i++){
    HAL_GPIO_WritePin(GPIOA, clockPins[i], GPIO_PIN_SET);
  }
}
```

2.9 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.

```
uint16_t clockPins[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};

void setNumberOnClock(int num){
  if (num >= 12 || num < 0) return;
  HAL_GPIO_WritePin(GPIOA, clockPins[num], GPIO_PIN_RESET
 );
}
```

2.10 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.

```
uint16_t clockPins[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};
void clearNumberOnClock(int num){
  if (num >= 12 || num < 0) return;
  HAL_GPIO_WritePin(GPIOA, clockPins[num], GPIO_PIN_SET);
}
```

2.11 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.

Report: Source code in while (1) loop.

```
int sec = 0; int secCount = 0; int hour = 0;
int min = 0; int minCount = 0; int hourCount = 0;
while(1){
  clearAllClock();
  if (secCount == 5){
    sec++;
    secCount = 0;
  }
  if (sec == 12){
    sec = 0;
    minCount++;
  }
  if (minCount == 5){
    min++;
    minCount = 0;
  }
  if (min == 12){
    min = 0;
    hourCount++;
  }
  if (hourCount == 1){
    hour++;
    hourCount = 0;
  }
  if (hour == 12){
    hour = 0;
  }
  setNumberOnClock(sec);
  setNumberOnClock(min);
  setNumberOnClock(hour);
  HAL_Delay(1000);
  secCount++;
}
```