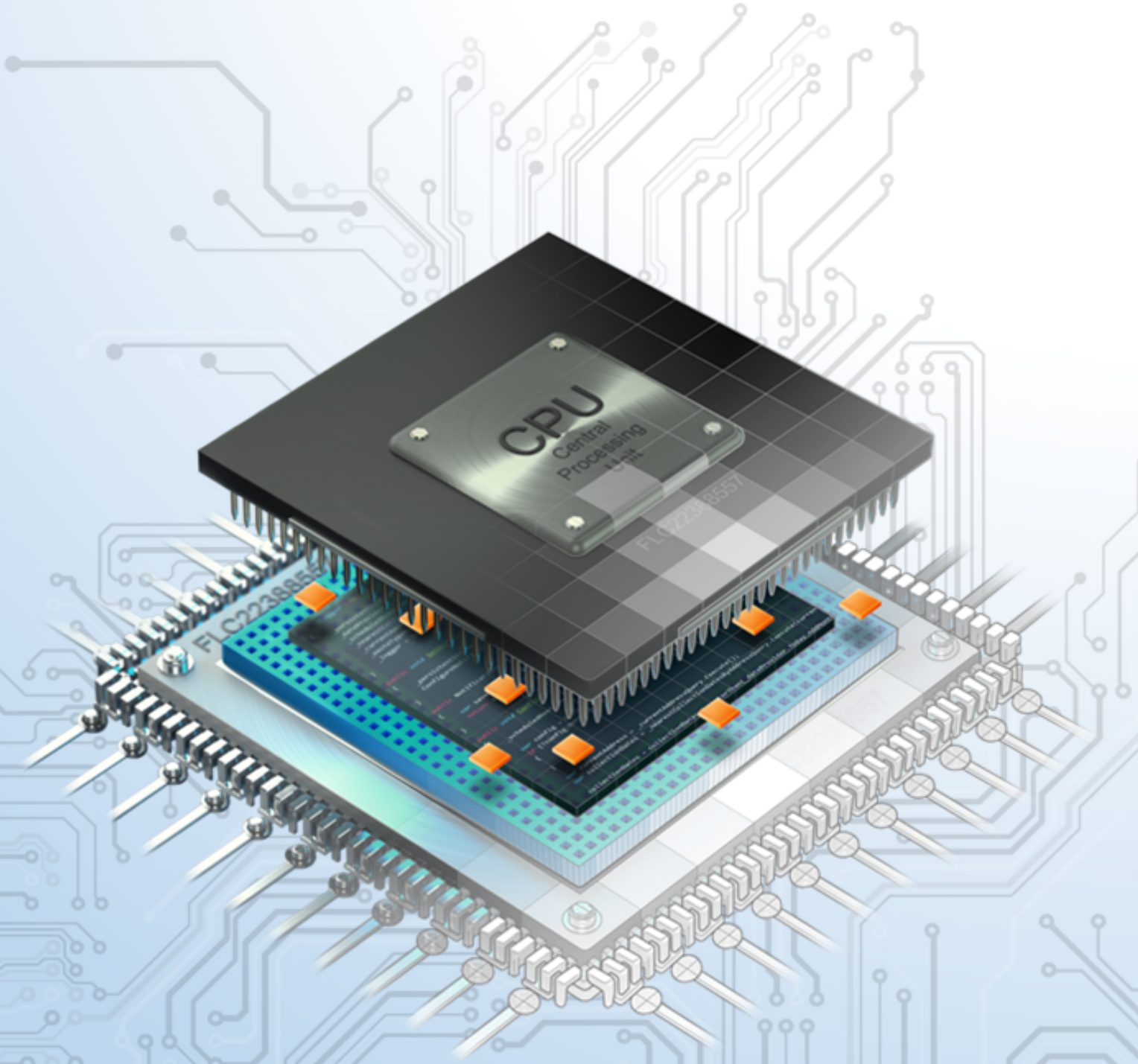
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller



Dr. Le Trong Nhan

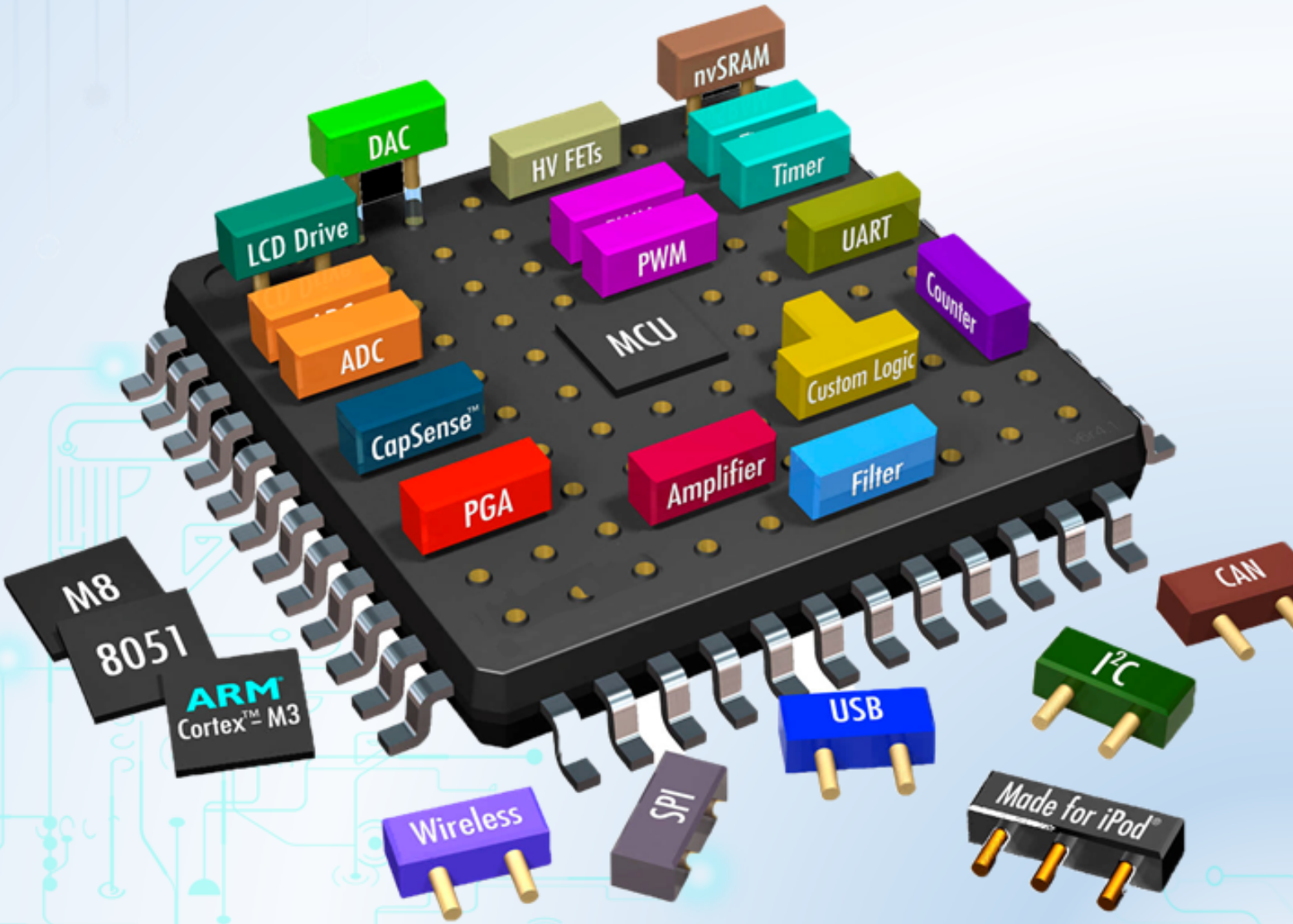
Dư Văn An

MSSV: 2012545

Mục lục

CHƯƠNG 1

LED Animations



1 Introduction

In this manual, the STM32CubeIDE is used as an editor to program the ARM microcontroller. STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors.

Hình 1.1: *STM32Cube IDE for STM32 Programming*

The most interest of STM32CubeIDE is that after the selection of an empty STM32 MCU or MPU, or preconfigured microcontroller or microprocessor from the selection of a board, the initialization code generated automatically. At any time during the development, the user can return to the initialization and configuration of the peripherals or middleware and regenerate the initialization code with no impact on the user code. This feature can simplify the initialization process and speedup the development application running on STM32 micro-controller. The software can be downloaded from the link bellow:

https://ubc.sgp1.digitaloceanspaces.com/BKU_Softwares/STM32/stm32cubeide_1.7.0.zip

Moreover, for a hangout class, the program is firstly simulated on Proteus. Students are also supposed to download and install this software as well:

https://ubc.sgp1.digitaloceanspaces.com/BKU_Softwares/STM32/Proteus_8.10_SP0_Pro.exe

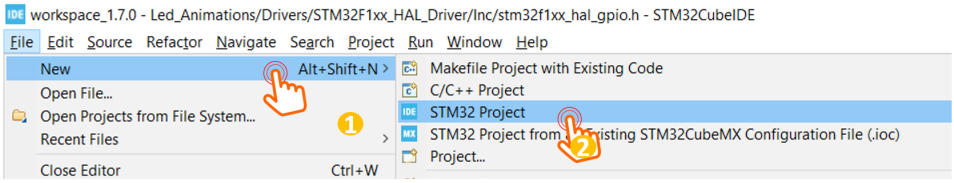
The rest of this manual consists of:

- Create a project on STM32Cube IDE
- Create a project on Proteus
- Simulate the project on Proteus

Finally, students are supposed to finish 10 different projects.

2 First project on STM32Cube

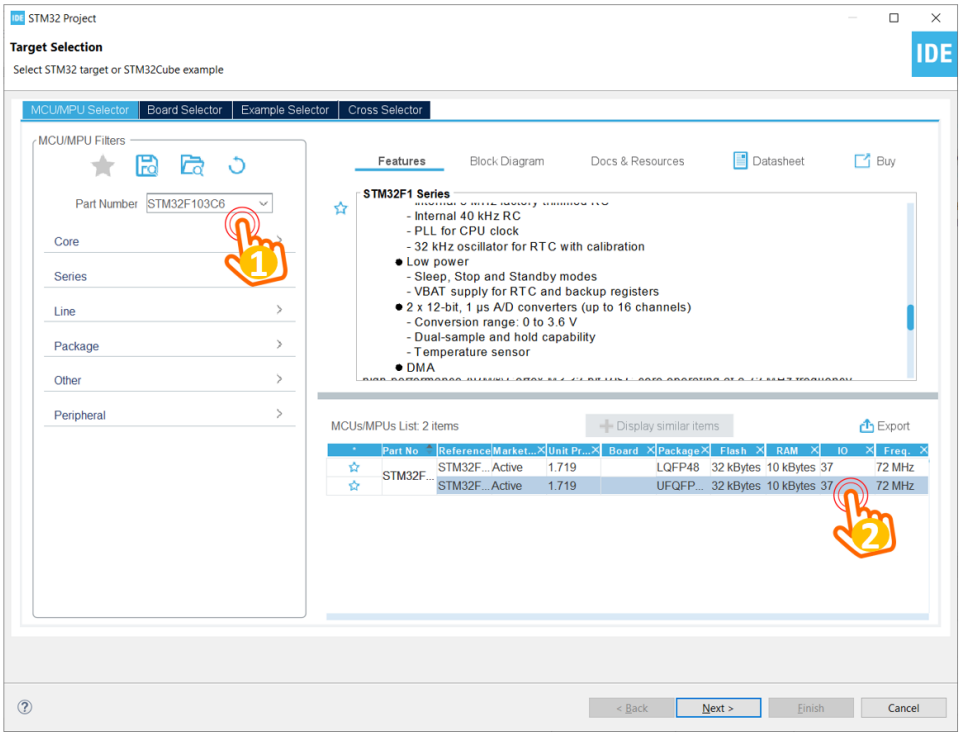
Step 1: Launch STM32CubeIDE, from the menu **File**, select **New**, then chose **STM32 Project**



Hình 1.2: Create a new project on STM32CubeIDE

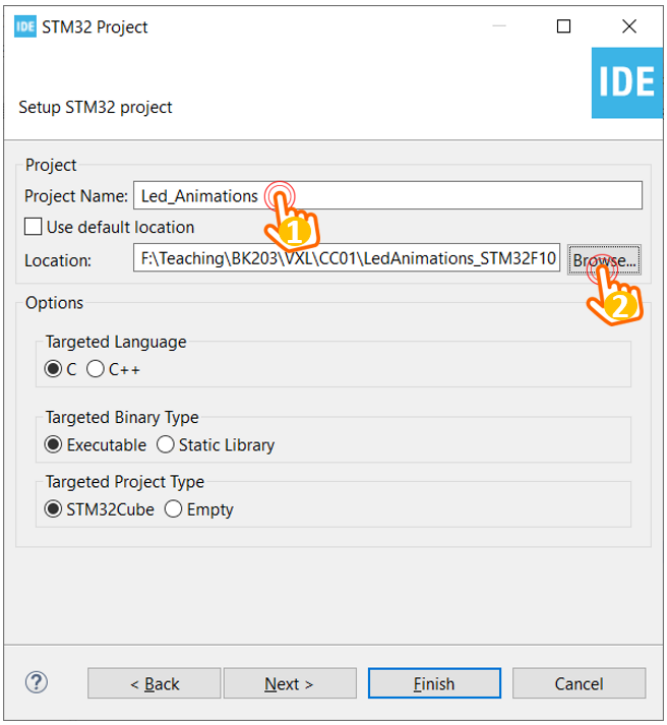
The IDE needs to download some packages, which normally takes time in this first time a project is created.

Step 2: Select the STM32F103C6 in the following dialog, then click on **Next**



Hình 1.3: Select the target device

Step 3: Provide the **Name** and the **Location** for the project.



Hình 1.4: Select the target device

It is important to notice that the **Targeted Project Type** should be **STM32Cube**. In the case this option is disable, step 1 must be repeated. The location path should not contain special characters (e.g. the space). Finally, click on the **Next** button.

Step 4: On the last dialog, just keep the default firmware version and click on **Finish** button.

Step 5: The project is created and the wizard for configuration is display. This utility from CubeIDE can simplify the configuration process for an ARM microcontroller like the STM32.

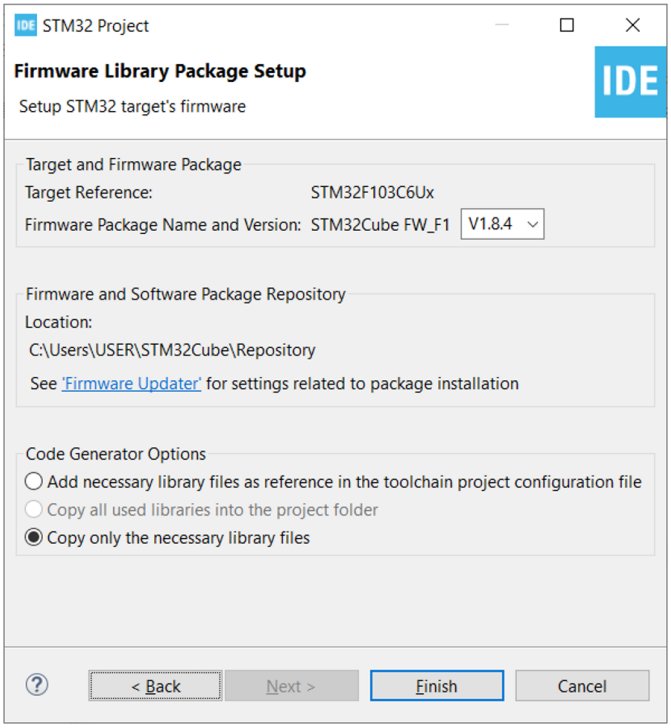
From the configuration windows, select **Pin configuration**, select the pin **PA5** and set to **GPIO Output** mode, since this pin is connected to an LED in the STM32 development kit.

Step 6: Right click on PA5 and select **Enter user lable**, and provide the name for this pin (e.g. **LED_RED**). This step helps programming afterward more memorable.

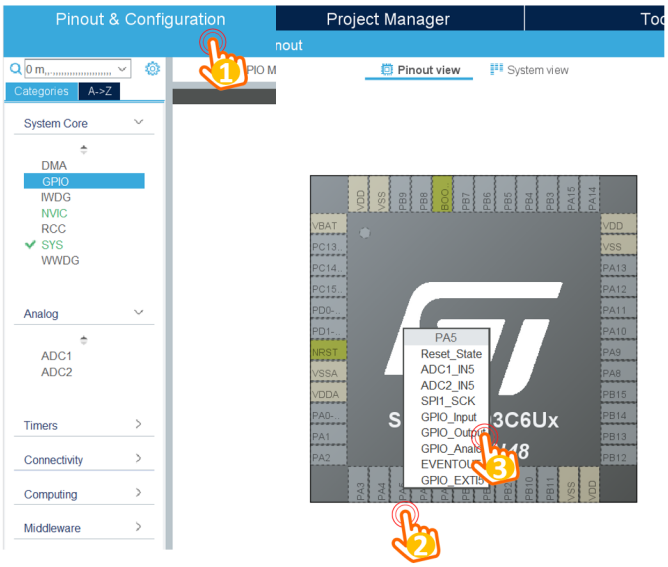
Finally, save the configuration process by pressing **Ctrl + S** and confirm this step by clicking on **OK** button. The code generation is started.

Step 7: Implement the first blinky project in the main function as follow:

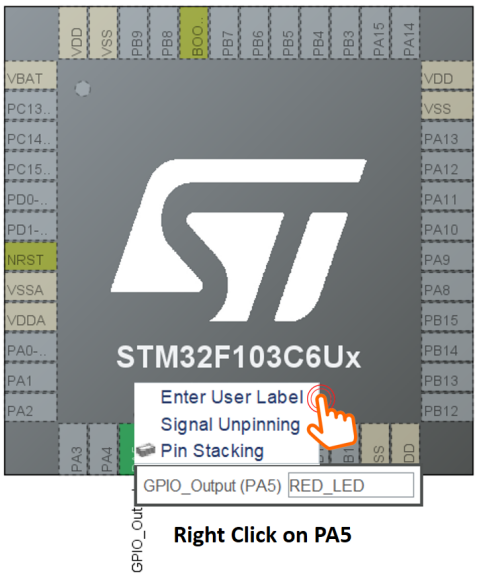
```
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */
```



Hình 1.5: Keep default firmware version



Hình 1.6: Set PA5 to GPIO Output mode



Hình 1.7: Provide a name for PA5

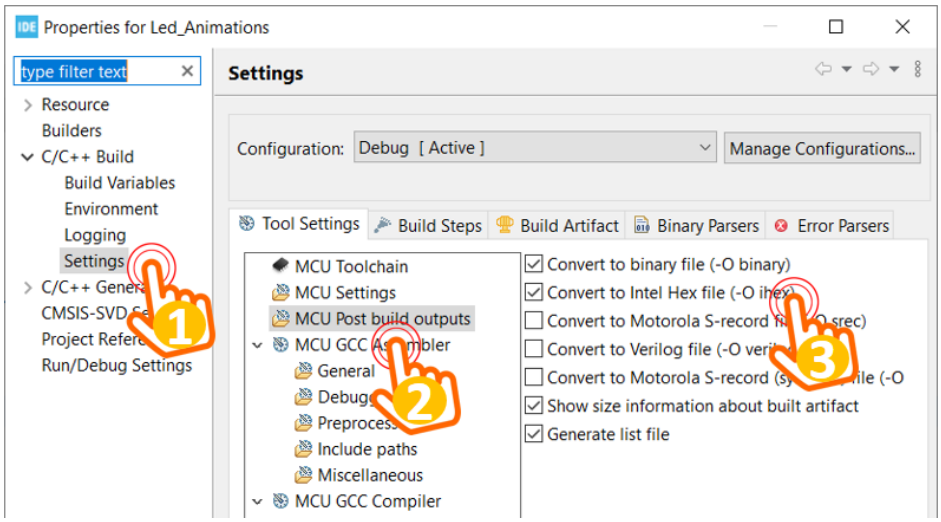
```
6
7   /* MCU Configuration
     --------------------------------------------------------
     */
8
9   /* Reset of all peripherals, Initializes the Flash
     interface and the Systick. */
10  HAL_Init();
11
12  /* USER CODE BEGIN Init */
13
14  /* USER CODE END Init */
15
16  /* Configure the system clock */
17  SystemClock_Config();
18
19  /* USER CODE BEGIN SysInit */
20
21  /* USER CODE END SysInit */
22
23  /* Initialize all configured peripherals */
24  MX_GPIO_Init();
25  /* USER CODE BEGIN 2 */
26
27  /* USER CODE END 2 */
28
29  /* Infinite loop */
30  /* USER CODE BEGIN WHILE */
31
32  while (1)
33  {
```

```
34      HAL_GPIO_TogglePin(LED_RED_GPIO_Port, LED_RED_Pin);
35      HAL_Delay(1000);
36      /* USER CODE END WHILE */
37
38      /* USER CODE BEGIN 3 */
39    }
40    /* USER CODE END 3 */
41  }
```

Program 1.1: First blinky LED project

Actually, what is added to the main function is line number 34 and 35. Please put your code in a right place, otherwise it can be deleted when the code is generated (e.g. change the configuration of the project). When coding, frequently use the suggestions by pressing **Ctrl+Space**.

Step 7: Due to the simulation on Proteus, the hex file should be generated from STM32Cube IDE. From menu **Project**, select **Properties** to open the dialog bellow:



Hình 1.8: Config for hex file output

Navigate to **C/C++ Build**, select **Settings, MCU Post build outputs**, and check to the **Intel Hex file**.

Step 8: Build the project by clicking on menu **Project** and select **Build Project**. Please check on the output console of the IDE to be sure that the hex file is generated, as follow:

```
22:36:06 **** Incremental Build of configuration Debug for project Led_Animations ****
make -j8 all
arm-none-eabi-size   Led_Animations.elf
   text    data     bss     dec     hex filename
   4596      20    1572    6188    182c Led_Animations.elf
Finished building: default.size.stdout


22:36:06 Build Finished. 0 errors, 0 warnings. (took 272ms)
```

Hình 1.9: Compile the project and generate Hex file

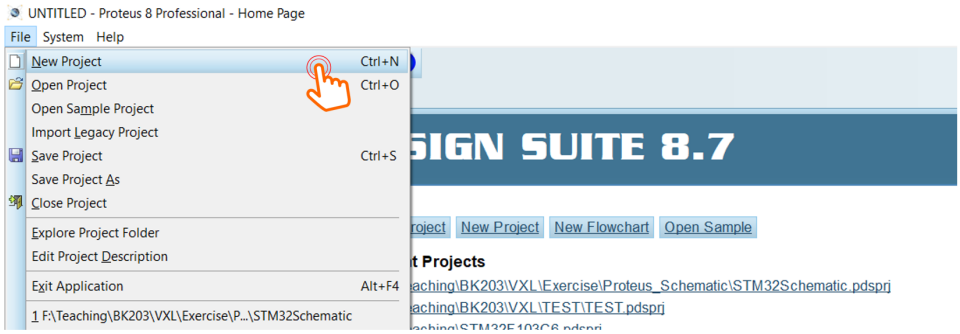
The hex file is located under the **Debug** folder of your project, which is used for the simulation in Proteus afterward. In the case a development kit is connected to your PC, from menu **Run**, select **Run** to download the program to the hardware platform.

In the case there are multiple project in a work-space, double click on the project name to activate this project. Whenever a project is built, check the output files to make sure that you are working in a right project.

3 Simulation on Proteus

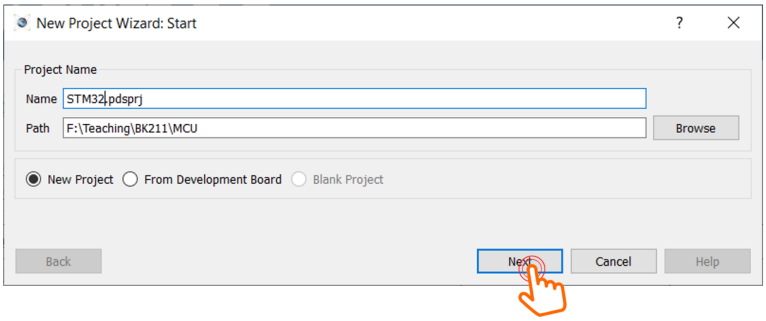
For an online training, a simulation on Proteus can be used. The details to create an STM32 project on Proteus are described bellow.

Step 1: Launch Proteus (**with administration access**) and from menu **File**, select **New Project**.



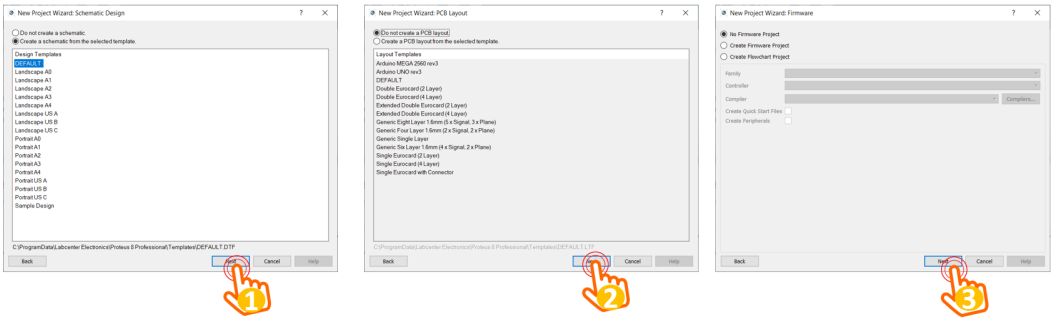
Hình 1.10: Create a new project on Proteus

Step 2: Provide the name and the location of the project, then click on **Next** button.



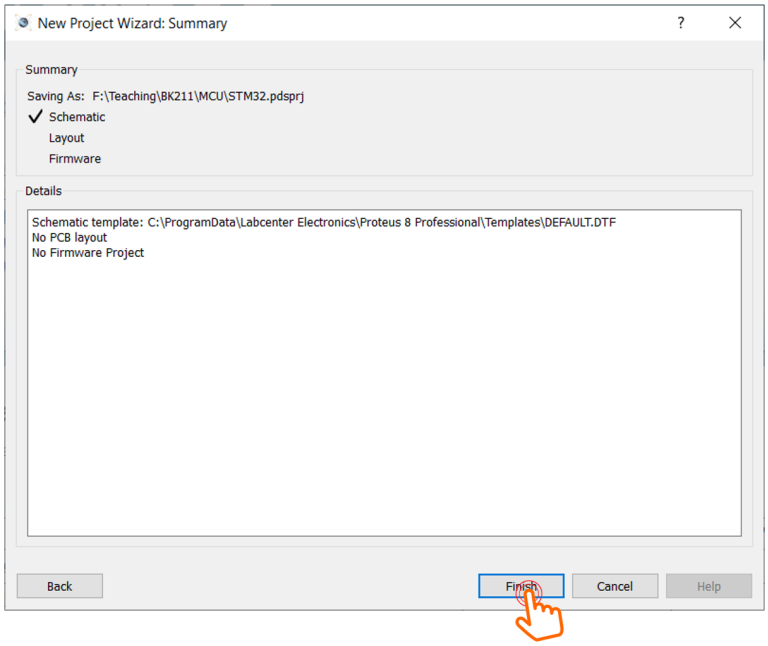
Hình 1.11: Provide project name and location

Step 3: For following dialog, just click on **Next** button as just a schematic is required for the lab.



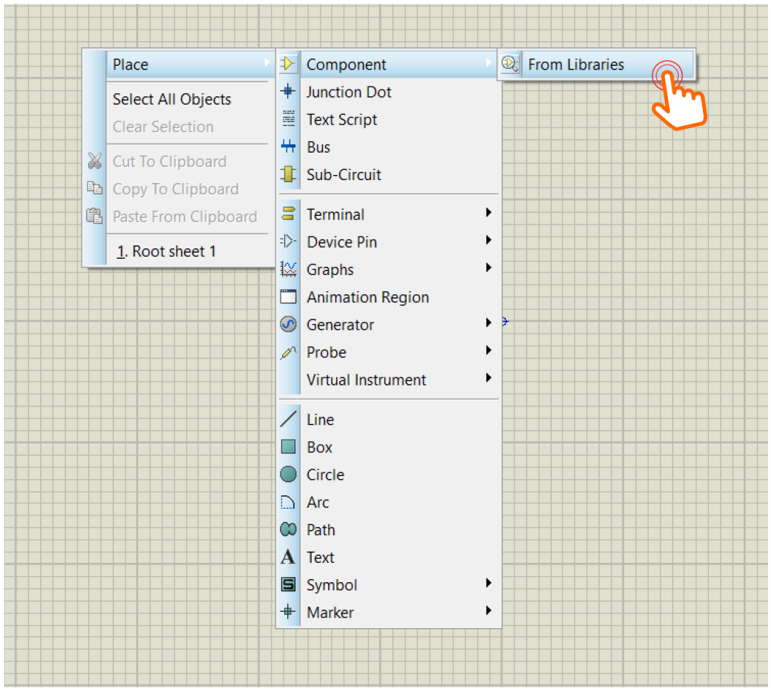
Hình 1.12: Keep the default options by clicking on Next

Step 4: Finally, click on **Finish** button to close the project wizard.



Hình 1.13: Finish the project wizard

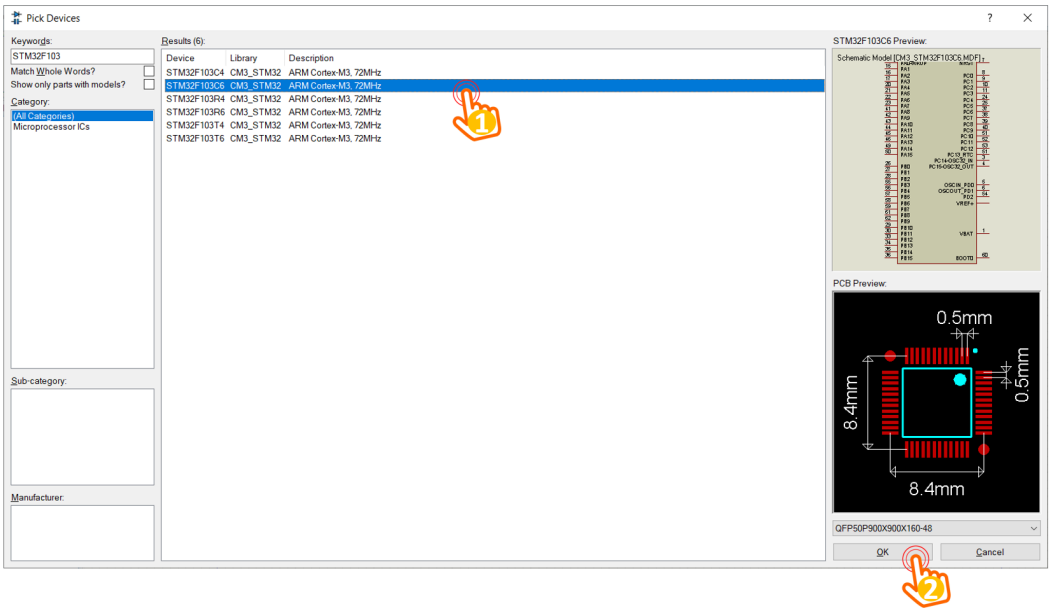
Step 5: On the main page of the project, right click to select **Place, Components, From Libraries**, as follows:



Hình 1.14: Select a component from the library

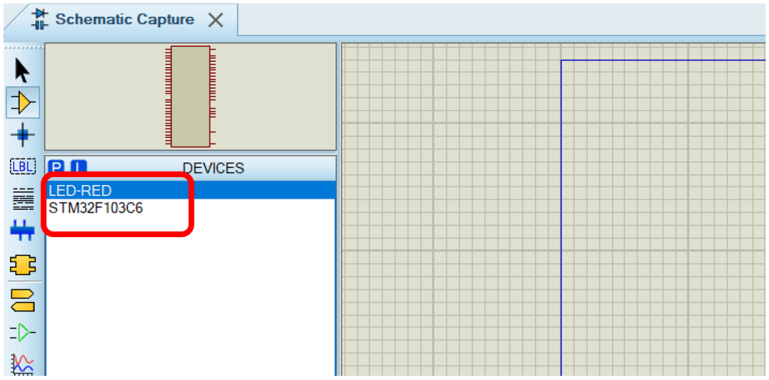
If there is an error with no library found, please restart the Proteus software with Run as administrator option.

Step 6: From the list of components in the library, select STM32F103C6, as follows:



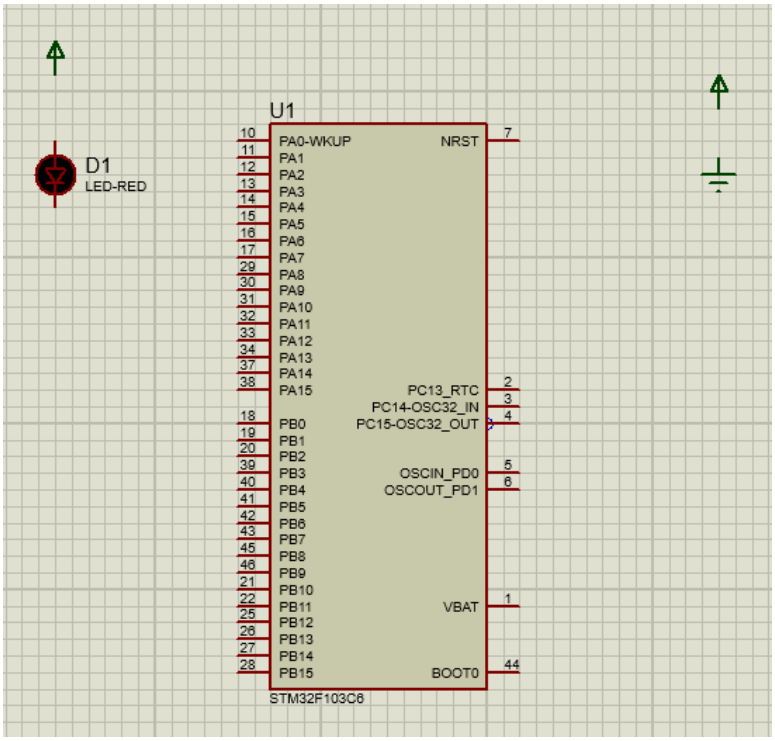
Hình 1.15: Select STM32F103C6

Repeat step 5 and 6 to select an LED, named **LED-RED** in Proteus. Finally, these components are appeared on the DEVICES windows, which is on left hand side as follows:



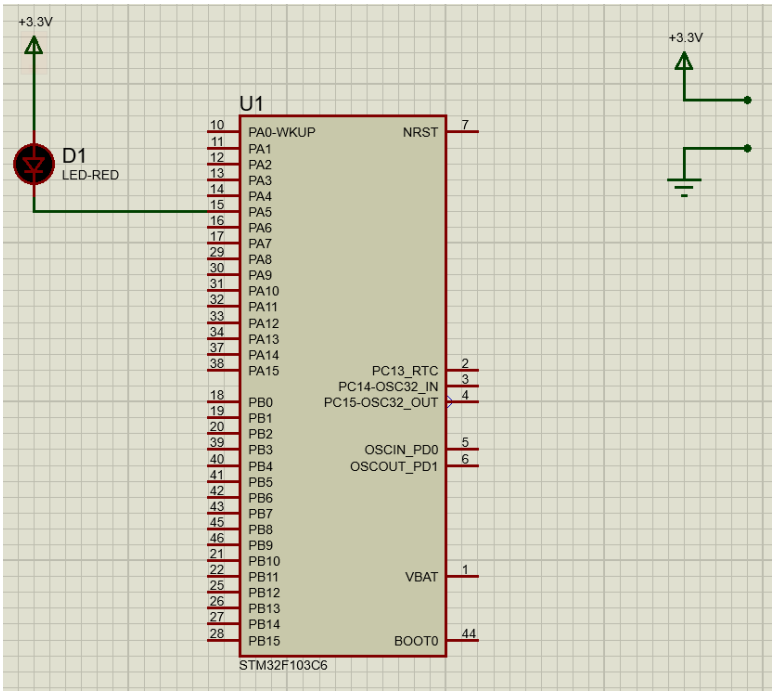
Hình 1.16: STM32 and an LED in the project

Step 7: Place the components to the project: right click on the main page, select on **Place, Component**, and select device added in Step 6. To add the Power and the Ground, right click on the main page, select on **Place, Terminal**. The result in this step is expected as follows:



Hình 1.17: *Place components to the project*

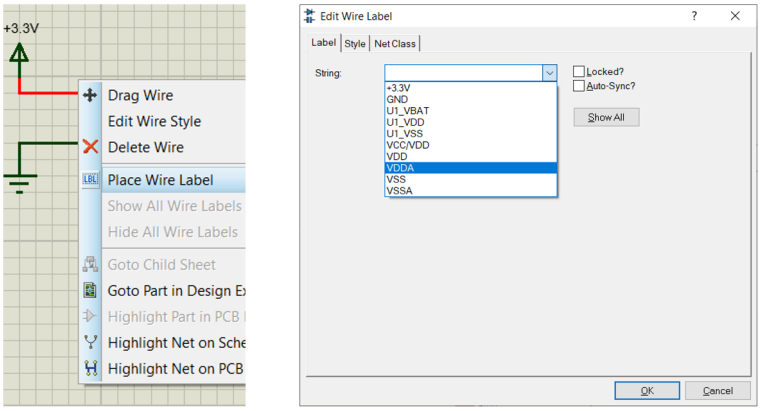
Step 8: Start wiring the circuit. The negative pin of the LED is connected to PA5 while its positive pin is connected to the power supply. For the power and the ground on the right, just make a short wire, which will labeled in the next step.



Hình 1.18: *Connect components and set the power to 3.3V*

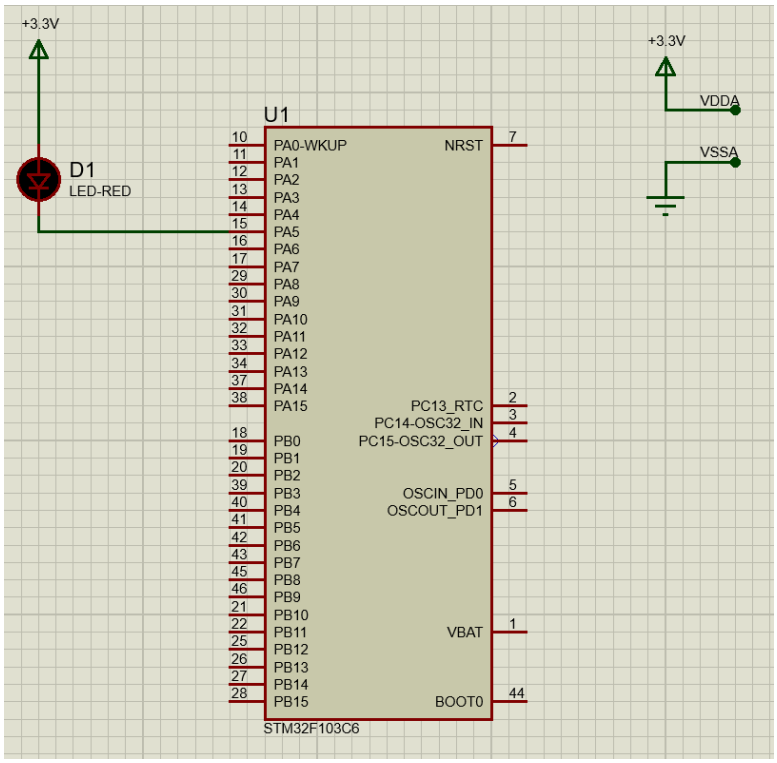
In this step, also double click on the power supply in order to provide the String property to **+3.3V**.

Step 8: Right click on the wire of the power supply and the ground, and select **Place wire Label**



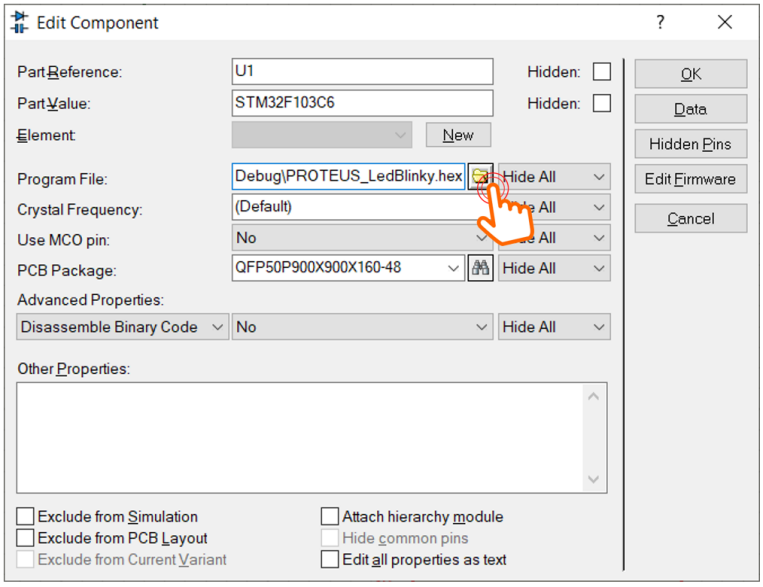
Hình 1.19: *Place label for Power and Ground*

This step is required as VDDA and VSSA of the STM32 must be connected to provide the reference voltage. Therefore, VDDA is connected to 3.3V, while the VSSA is connected to the Ground. Finally, the image of our schematic is shown bellow:



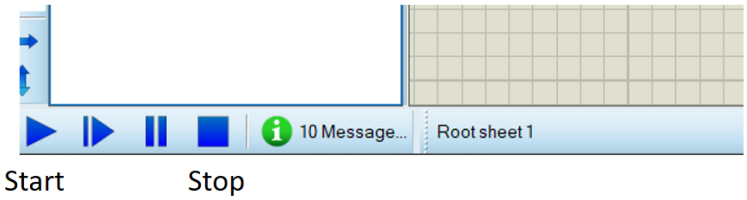
Hình 1.20: *Finalize the schematic*

Step 9: Double click on the STM32, and set the **Program File** to the Hex file, which is generated from Cube IDE, as following:



Hình 1.21: *Set the program of the STM32 to the hex file from Cube IDE*

From now, the simulation is ready to start by clicking on the menu **Debug**, and select on **Run simulation**. To stop the simulation, click on **Debug** and select **Stop VMS Debugging**. Moreover, there are some quick access bottom on the left corner of the Proteus to start or stop the simulation, as shown following:



Hình 1.22: *Quick access buttons to start and stop the simulation*

If everything is success, students can see the LED is blinking every second. Please stop the simulation before updating the project, either in Proteus or STM32Cube IDE. However, the step 9 (set the program file for STM32 in Proteus) is required to do once. Beside the toggle instruction, student can set or reset a pin as following:

```
1 while (1){
2   HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
     GPIO_PIN_SET);
3   HAL_Delay(1000);
4   HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
     GPIO_PIN_RESET);
5   HAL_Delay(1000);
6 }
```

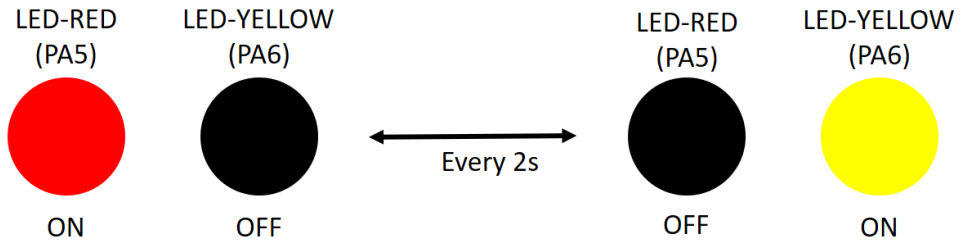
Program 1.2: An example for LED blinky

4 Exercise and Report

4.1 Exercise 1

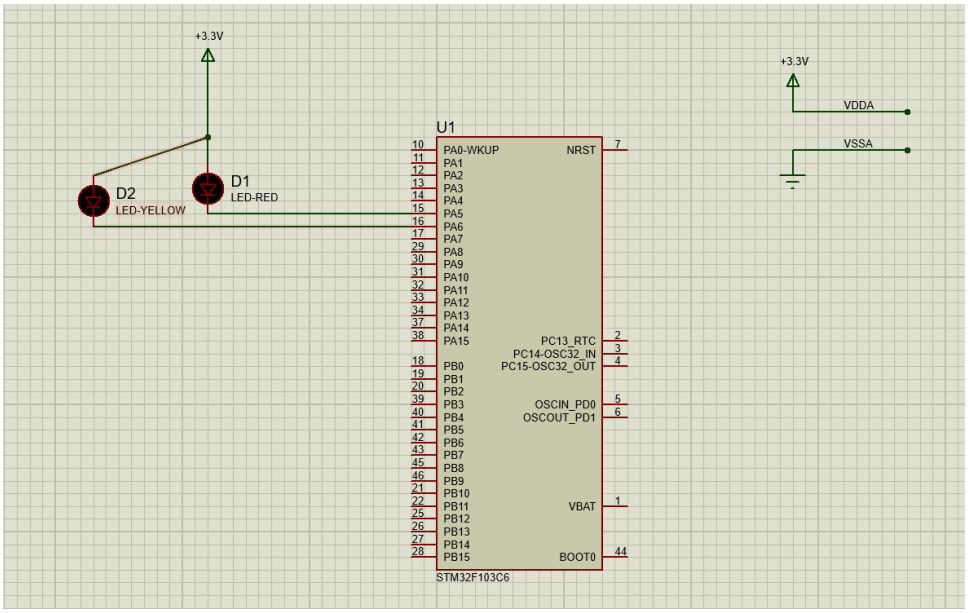
From the simulation on Proteus, one more LED is connected to pin **PA6** of the STM32 (negative pin of the LED is connected to PA6). The component suggested in this exercise is **LED-YELLOW**, which can be found from the device list.

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



Hình 1.23: 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 1.24: https://github.com/duvanan13/Lab1.git

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.

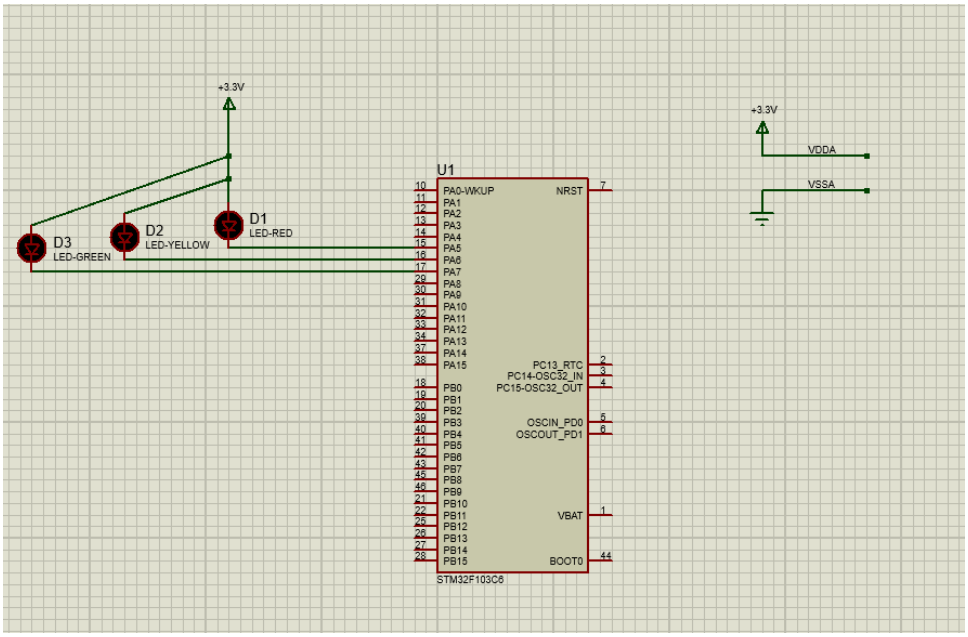
```
while (1){
  if(counter>1){
        HAL_GPIO_WritePin(LED_RED_GPIO_Port,
        LED_RED_Pin, RESET);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
        LED_YELLOW_Pin, SET);
    }
    else{
        HAL_GPIO_WritePin(LED_RED_GPIO_Port,
        LED_RED_Pin, SET);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
        LED_YELLOW_Pin, RESET);
    }
    counter--;
    if(counter==0){
       counter=2;
    }
    HAL_Delay(2000);
}
```

4.2 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.



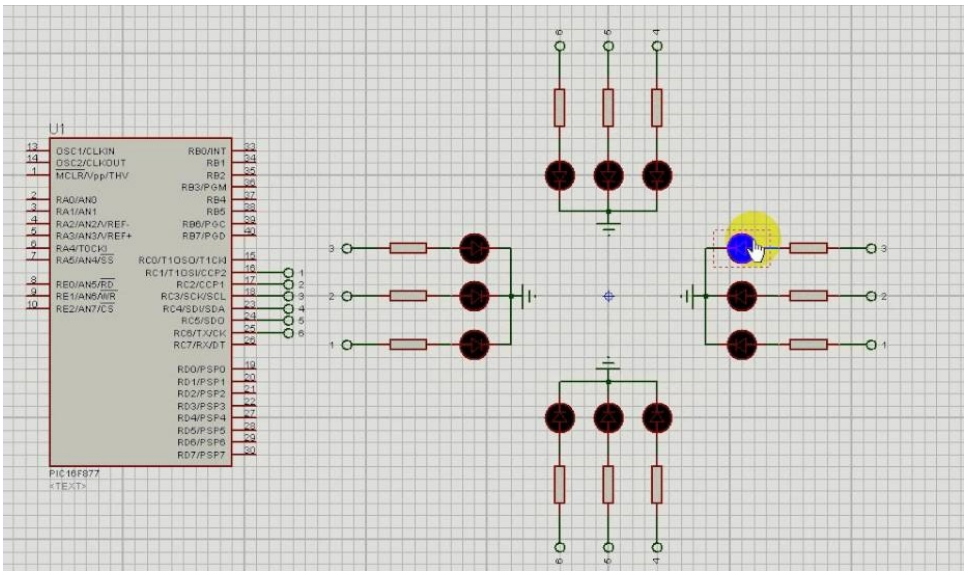
Hình 1.25: *https://github.com/duvanan13/Lab1.git*

Report 2: Present the source code in while.

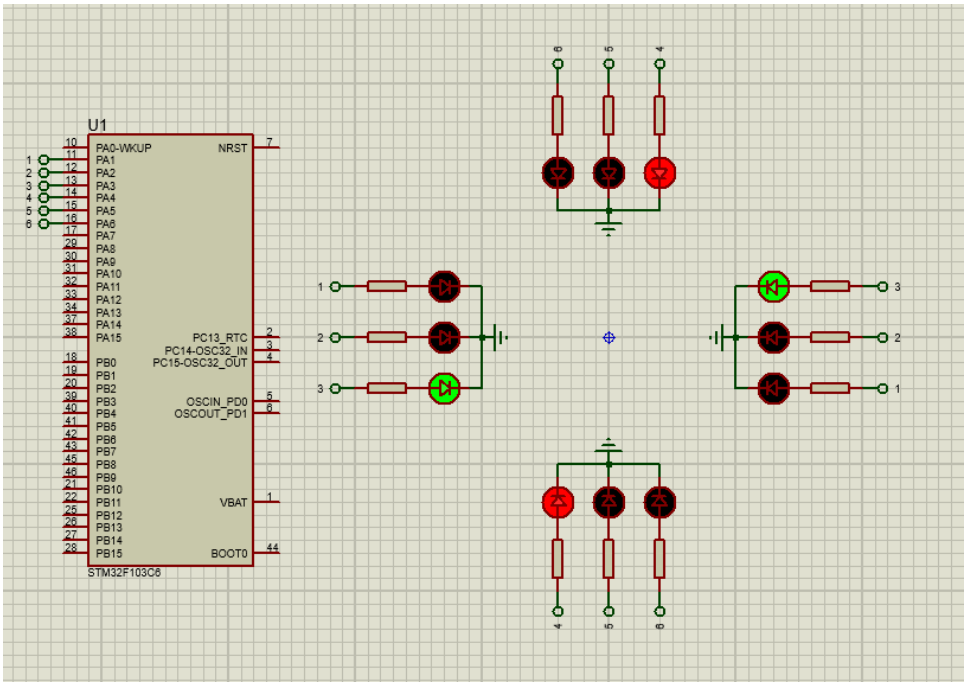
```
int counter = 10;
while (1){
    if (counter > 5) {
      HAL_GPIO_WritePin(LED_RED_GPIO_Port,
      LED_RED_Pin, RESET);
      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port,
      LED_GREEN_Pin, SET);
      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
      LED_YELLOW_Pin, SET);
    } else if(counter > 3){
      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
      LED_YELLOW_Pin, RESET);
      HAL_GPIO_WritePin(LED_RED_GPIO_Port,
      LED_RED_Pin, SET);
      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port,
      LED_GREEN_Pin, SET);
    } else {
      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port,
      LED_GREEN_Pin, RESET);
      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
      LED_YELLOW_Pin, SET);
      HAL_GPIO_WritePin(LED_RED_GPIO_Port,
      LED_RED_Pin, SET);
    }
    counter--;
    if(counter==0) counter = 10;
    HAL_Delay(1000);
}
```

4.3 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 1.26: *Reference design for a 4 way traffic light*



Hình 1.27: *https://github.com/duvanan13/Lab1.git*

Present the source code in while.

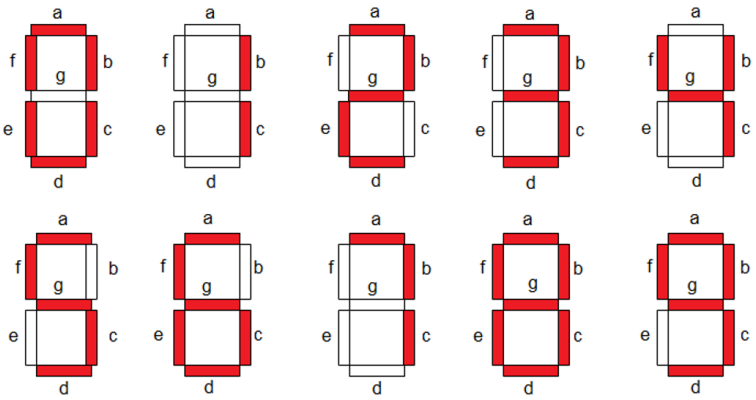
```
int counter0 = 10;
while (1){
  if (counter0 > 5) {
    HAL_GPIO_WritePin(LED_RED1_GPIO_Port,LED_RED1_Pin,SET);
```

```
    HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,
   LED_YELLOW1_Pin, RESET);
    HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port, LED_GREEN1_Pin,
    RESET);
    if (counter0 > 7) {
    HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
   RESET);
    HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,
   LED_YELLOW2_Pin, RESET);
    HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin,
    SET);
    } else {
    HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
   RESET);
    HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,
   LED_YELLOW2_Pin, SET);
    HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin,
    RESET);
    }
} else{
    HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin, SET
   );
    HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,
   LED_YELLOW2_Pin, RESET);
    HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin,
    RESET);
    if (counter0 > 2) {
    HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
   RESET);
    HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,
   LED_YELLOW1_Pin, RESET);
    HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port, LED_GREEN1_Pin,
    SET);
    } else{
    HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
   RESET);
    HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,
   LED_YELLOW1_Pin, SET);
    HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port, LED_GREEN1_Pin,
    RESET);
    }
}
    counter0--;
    if (counter0 == 0) counter0 = 10;
    HAL_Delay(1000);;
}
```

4.4 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:



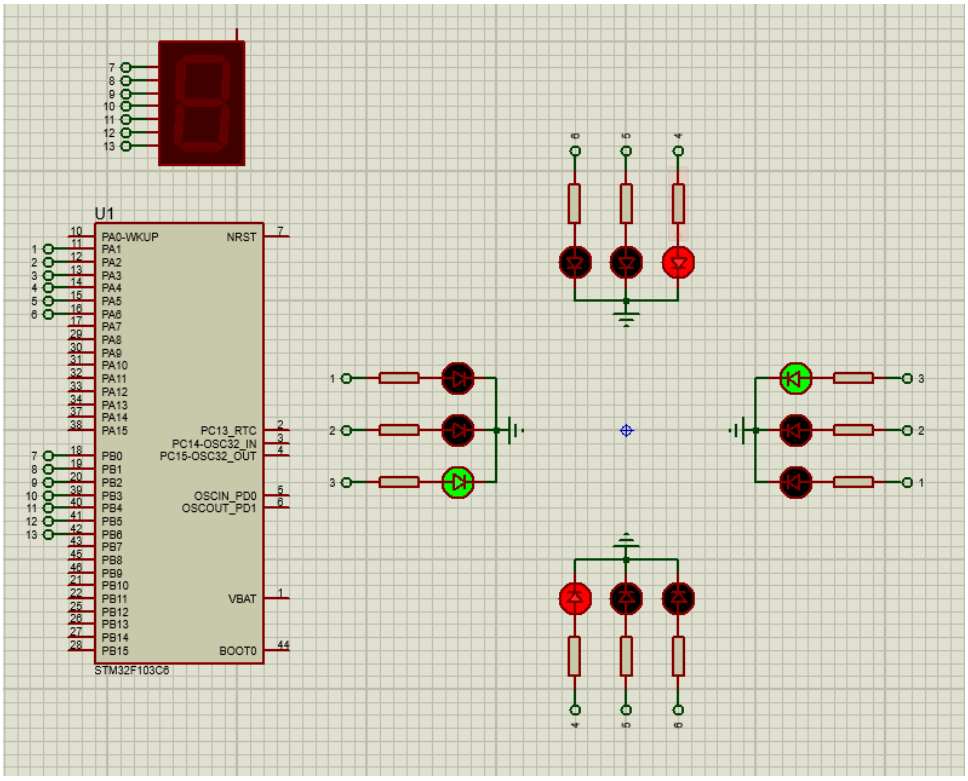
Hình 1.28: *Display a number on 7 segment LED*

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 1.29: *https://github.com/duvanan13/Lab1.git*

Report 2: Present the source code for display7SEG function.

```
void display7SEG(int num){
  if(num==0){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
  }
  if(num==1){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
  }
  if(num==2){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
```

```
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
  }
  if(num==3){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
  }
  if(num==4){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
  }
  if(num==5){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
  }
  if(num==6){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
  }
  if(num==7){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
  }
  if(num==8){
```

```
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
  }
  if(num==9){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
  }
}
```

4.5 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 this exercise, only source code is required to present. The function display7SEG in previous exercise can be re-used.

```
void display7SEG1_5(int num){     // ex5
    if(num==5){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
    }
    if(num==4){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
    }
    if(num==3){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 1);
```

```
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
    }
    if(num==2){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
    }
    if(num==1){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
    }
    if(num==0){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
    }
}
void display7SEG1_3(int num){  // ex 5
    if(num==3){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
    }
    if(num==2){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
```

```
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
    }
    if(num==1){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
    }
    if(num==0){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
    }
}
void display7SEG1_2(int num){  // ex 5
    if(num==2){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
    }
     if(num==1){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 1);
    }
    if(num==0){
    HAL_GPIO_WritePin(LED_SEG1_GPIO_Port, LED_SEG1_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG2_GPIO_Port, LED_SEG2_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG3_GPIO_Port, LED_SEG3_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG4_GPIO_Port, LED_SEG4_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG5_GPIO_Port, LED_SEG5_Pin, 0);
    HAL_GPIO_WritePin(LED_SEG6_GPIO_Port, LED_SEG6_Pin, 1);
    HAL_GPIO_WritePin(LED_SEG7_GPIO_Port, LED_SEG7_Pin, 0);
```

```
122     }
123 }
124 void display7SEG2_5(int num){     // ex5
125     if(num==5){
126     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
127     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
128     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
129     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
130     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,0);
131     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,0);
132     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,1);
133     }
134     if(num==4){
135     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 1);
136     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
137     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
138     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,1);
139     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,1);
140     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,1);
141     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,1);
142     }
143     if(num==3){
144     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
145     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
146     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,1);
147     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
148     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,0);
149     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,1);
150     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,0);
151     }
152     if(num==2){
153     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
154     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
155     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
156     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
157     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,1);
158     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,1);
159     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,0);
160     }
161     if(num==1){
162     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 1);
163     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
164     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
165     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,1);
166     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,1);
167     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,0);
168     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,0);
169     }
170     if(num==0){
```

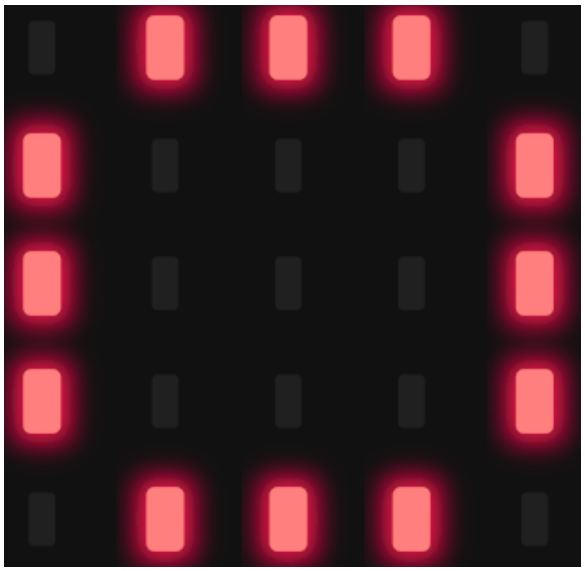
```
171     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
172     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 1);
173     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
174     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
175     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,1);
176     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,0);
177     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,0);
178     }
179 }
180 void display7SEG2_3(int num){  // ex 5
181     if(num==3){
182     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
183     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
184     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
185     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
186     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,0);
187     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,0);
188     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,1);
189     }
190     if(num==2){
191     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 1);
192     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
193     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
194     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,1);
195     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,1);
196     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,1);
197     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,1);
198     }
199     if(num==1){
200     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
201     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
202     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,1);
203     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
204     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,0);
205     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,1);
206     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,0);
207       }
208     if(num==0){
209     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
210     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
211     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
212     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
213     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,1);
214     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,1);
215     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,0);
216       }
217 }
218 void display7SEG2_2(int num){  // ex 5
219     if(num==2){
```

```
220     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
221     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
222     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
223     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
224     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,0);
225     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,0);
226     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,1);
227       }
228       if(num==1){
229     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 1);
230     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
231     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,0);
232     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,1);
233     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,1);
234     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,1);
235     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,1);
236     }
237     if(num==0){
238     HAL_GPIO_WritePin(LED_SEG8_GPIO_Port, LED_SEG8_Pin, 0);
239     HAL_GPIO_WritePin(LED_SEG9_GPIO_Port, LED_SEG9_Pin, 0);
240     HAL_GPIO_WritePin(LED_SEG10_GPIO_Port,LED_SEG10_Pin,1);
241     HAL_GPIO_WritePin(LED_SEG11_GPIO_Port,LED_SEG11_Pin,0);
242     HAL_GPIO_WritePin(LED_SEG12_GPIO_Port,LED_SEG12_Pin,0);
243     HAL_GPIO_WritePin(LED_SEG13_GPIO_Port,LED_SEG13_Pin,1);
244     HAL_GPIO_WritePin(LED_SEG14_GPIO_Port,LED_SEG14_Pin,0);
245     }
246 }
247 int counter = 0;
248 int counter0 = 0;
249 int counter1 = 0;
250 int counter2 = 0;
251 while(1){
252 if (counter0 < 5) { // ex 5
253    HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
    GPIO_PIN_SET);
254    HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,LED_YELLOW1_Pin,
    GPIO_PIN_RESET);
255    HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port, LED_GREEN1_Pin,
    GPIO_PIN_RESET);
256    if (counter0 <3) {
257    HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
    GPIO_PIN_RESET);
258    HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,LED_YELLOW2_Pin,
    GPIO_PIN_RESET);
259    HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin,
    GPIO_PIN_SET);
260    if(counter1==3) counter1=0;
261    display7SEG2_3(counter1++);
262    } else {
```

```
263     HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
        GPIO_PIN_RESET);
264     HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,LED_YELLOW2_Pin,
        GPIO_PIN_SET);
265     HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin,
        GPIO_PIN_RESET);
266     if(counter2==2) counter2=0;
267     display7SEG2_2(counter2++);
268     }
269     if(counter==5) counter=0;
270     display7SEG1_5(counter++);
271 } else {
272     HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
        GPIO_PIN_SET);
273     HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,LED_YELLOW2_Pin,
        GPIO_PIN_RESET);
274     HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin,
        GPIO_PIN_RESET);
275     if(counter==5) counter=0;
276     display7SEG2_5(counter++);
277     if (counter0 <8) {
278     HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
        GPIO_PIN_RESET);
279     HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,LED_YELLOW1_Pin,
        GPIO_PIN_RESET);
280     HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port, LED_GREEN1_Pin,
        GPIO_PIN_SET);
281     if(counter1==3) counter1=0;
282     display7SEG1_3(counter1++);
283     } else {
284     HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
        GPIO_PIN_RESET);
285     HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,LED_YELLOW1_Pin,
        GPIO_PIN_SET);
286     HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port, LED_GREEN1_Pin,
        GPIO_PIN_RESET);
287     if(counter2==2) counter2=0;
288     display7SEG1_2(counter2++);
289     }
290 }
291     counter0++;
292     if (counter0 == 10) counter0 = 0;
293     HAL_Delay(1000);
294 }
```

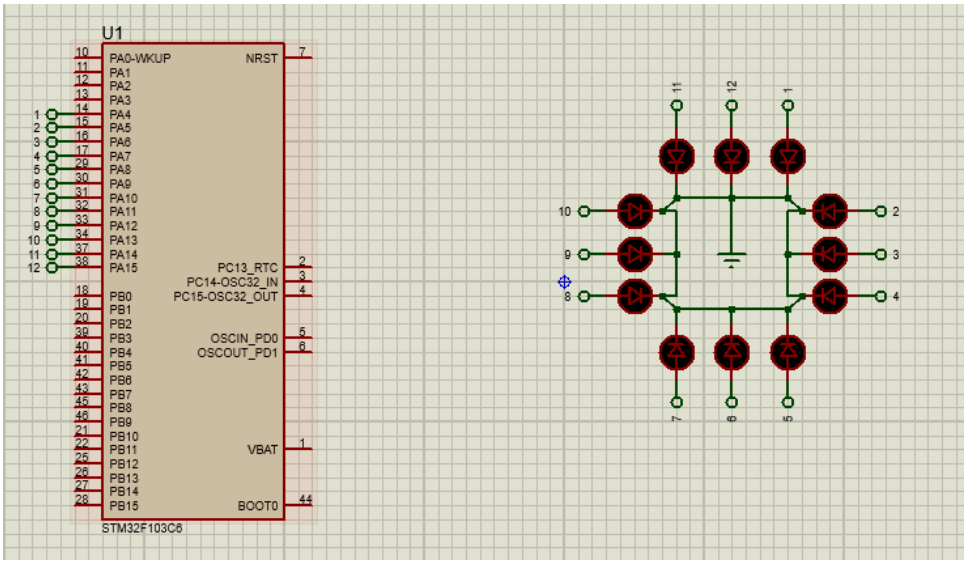
4.6 Exercise 6

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 1.30: 12 LEDs for an analog clock

Report 1: Present the schematic.



Hình 1.31: https://github.com/duvanan13/Lab1.git

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 sequence(int counter){    // ex 6
    if(counter<1){
    HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,SET );
    HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
    HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
    HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
    HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
    HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
    HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
    HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
    HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
    HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
    HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
    HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
    }
    else if(counter<2){
    HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
    HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
    HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
    HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
    HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
    HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
    HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
    HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
    HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
    HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
    HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
    HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, SET );
    }
    else if(counter<3){
    HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
    HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
    HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
    HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
    HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
    HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
    HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
    HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
    HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
    HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
    HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, SET );
    HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
    }
    else if(counter<4){
    HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
    HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
```

```
47      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
48      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
49      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
50      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
51      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
52      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
53      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
54      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, SET );
55      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
56      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
57      }
58      else if(counter<5){
59      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
60      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
61      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
62      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
63      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
64      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
65      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
66      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
67      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, SET );
68      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
69      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
70      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
71      }
72      else if(counter<6){
73      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
74      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
75      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
76      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
77      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
78      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
79      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
80      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, SET );
81      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
82      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
83      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
84      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
85      }
86      else if(counter<7){
87      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
88      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
89      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
90      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
91      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
92      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
93      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, SET );
94      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
95      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
```

```
96      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
97      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
98      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
99      }
100     else if(counter<8{
101     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
102     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
103     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
104     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
105     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
106     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, SET );
107     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
108     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
109     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
110     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
111     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
112     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
113     }
114     else if(counter<9){
115     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
116     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
117     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
118     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
119     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, SET );
120     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
121     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
122     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
123     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
124     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
125     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
126     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
127     }
128     else if(counter<10){
129     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
130     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
131     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
132     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, SET );
133     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
134     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
135     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
136     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
137     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
138     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
139     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
140     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
141     }
142     else if(counter<11){
143     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
144     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
```

```
145     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,SET );
146     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
147     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
148     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
149     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
150     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
151     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
152     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
153     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
154     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
155     }
156     else if(counter<12){
157     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
158     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin, SET );
159     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
160     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
161     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
162     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
163     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
164     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
165     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
166     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
167     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
168     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
169     }
170 }
```

4.7 Exercise 7

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

```
void clearAllClock(){
    //TODO
}
```

Program 1.4: Function Implementation

```
void clearAllClock(){
  HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin, RESET );
  HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin, RESET );
  HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin, RESET );
  HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
  HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
  HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
  HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
  HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
  HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
  HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
  HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
  HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
 }
```

4.8 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){
  if(num==0 ){
    HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin, SET );
  }
  else if(num==1){
    HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, SET );
  }
  else if(num==2){
    HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, SET );
  }
  else if(num==3){
     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, SET );
  }
  else if(num==4){
      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, SET );
  }
  else if(num==5){
    HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, SET );
```

```
  }
  else if(num==6){
    HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, SET );
  }
  else if(num==7){
     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, SET );
  }
  else if(num==8){
     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, SET );
  }
  else if(num==9){
     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, SET );
  }
  else if(num==10){
     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin, SET );
  }
  else if(num==11){
     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin, SET );
  }
}
```

4.9 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 setNumberOnClock(int num){
  if(num==0 ){
  HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin, RESET);
  }
  else if(num==1){
    HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
  }
  else if(num==2){
    HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
  }
  else if(num==3){
     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
  }
  else if(num==4){
    HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
  }
  else if(num==5){
    HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
  }
  else if(num==6){
    HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
  }
  else if(num==7){
```

```
24      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
25    }
26    else if(num==8){
27      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
28    }
29    else if(num==9){
30      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
31    }
32    else if(num==10){
33    HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin, RESET );
34    }
35    else if(num==11){
36    HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin, RESET );
37    }
38  }
```

4.10 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.

```
1  void second(int counter){   // ex 10
2      if(counter<1){
3      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,SET );
4      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
5      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
6      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
7      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
8      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
9      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
10     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
11     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
12     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
13     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
14     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
15     }
16     else if(counter<2){
17     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
18     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
19     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
20     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
21     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
22     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
23     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
24     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
25     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
26     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
27     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
28     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, SET );
```

```
29      }
30      else if(counter<3){
31      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
32      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
33      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
34      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
35      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
36      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
37      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
38      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
39      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
40      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
41      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, SET );
42      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
43      }
44      else if(counter<4){
45      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
46      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
47      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
48      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
49      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
50      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
51      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
52      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
53      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
54      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, SET );
55      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
56      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
57      }
58      else if(counter<5){
59      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
60      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
61      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
62      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
63      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
64      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
65      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
66      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
67      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, SET );
68      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
69      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
70      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
71      }
72      else if(counter<6){
73      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
74      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
75      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
76      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
77      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
```

```
78      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
79      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
80      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, SET );
81      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
82      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
83      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
84      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
85      }
86      else if(counter<7){
87      HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
88      HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
89      HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
90      HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
91      HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
92      HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
93      HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, SET );
94      HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
95      HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
96      HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
97      HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
98      HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
99      }
100     else if(counter<8{
101     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
102     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
103     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
104     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
105     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
106     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, SET );
107     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
108     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
109     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
110     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
111     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
112     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
113     }
114     else if(counter<9){
115     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
116     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
117     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
118     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
119     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, SET );
120     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
121     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
122     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
123     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
124     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
125     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
126     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
```

```
127     }
128     else if(counter<10){
129     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
130     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
131     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
132     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, SET );
133     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
134     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
135     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
136     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
137     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
138     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
139     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
140     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
141     }
142     else if(counter<11){
143     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
144     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin,RESET );
145     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,SET );
146     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
147     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
148     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
149     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
150     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
151     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
152     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
153     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
154     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
155     }
156     else if(counter<12){
157     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin,RESET );
158     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin, SET );
159     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin,RESET );
160     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, RESET );
161     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, RESET );
162     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, RESET );
163     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, RESET );
164     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, RESET );
165     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, RESET );
166     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, RESET );
167     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, RESET );
168     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, RESET );
169     }
170 }
171   void hour(int counter){ //ex 10
172     if(counter<5 ){
173     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin, SET );
174     }
175     else if(counter<10){
```

```
176     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, SET );
177     }
178     else if(counter<15){
179     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, SET );
180     }
181     else if(counter<20){
182     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, SET );
183     }
184     else if(counter<25){
185     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, SET );
186     }
187     else if(counter<30){
188     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, SET );
189     }
190     else if(counter<35){
191     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, SET );
192     }
193     else if(counter<40){
194     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, SET );
195     }
196     else if(counter<45){
197     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, SET );
198     }
199     else if(counter<50){
200     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, SET );
201     }
202     else if(counter<55){
203     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin, SET );
204     }
205     else if(counter<60){
206     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin, SET );
207     }
208   }
209   void minute(int counter){ // ex 10
210     if(counter<5 ){
211     HAL_GPIO_WritePin(LED_12_GPIO_Port, LED_12_Pin, SET );
212     }
213     else if(counter<10){
214     HAL_GPIO_WritePin(LED_1_GPIO_Port, LED_1_Pin, SET );
215     }
216     else if(counter<15){
217     HAL_GPIO_WritePin(LED_2_GPIO_Port, LED_2_Pin, SET );
218     }
219     else if(counter<20){
220     HAL_GPIO_WritePin(LED_3_GPIO_Port, LED_3_Pin, SET );
221     }
222     else if(counter<25){
223     HAL_GPIO_WritePin(LED_4_GPIO_Port, LED_4_Pin, SET );
224     }
```

```
225     else if(counter<30){
226     HAL_GPIO_WritePin(LED_5_GPIO_Port, LED_5_Pin, SET );
227     }
228     else if(counter<35){
229     HAL_GPIO_WritePin(LED_6_GPIO_Port, LED_6_Pin, SET );
230     }
231     else if(counter<40){
232     HAL_GPIO_WritePin(LED_7_GPIO_Port, LED_7_Pin, SET );
233     }
234     else if(counter<45){
235     HAL_GPIO_WritePin(LED_8_GPIO_Port, LED_8_Pin, SET );
236     }
237     else if(counter<50){
238     HAL_GPIO_WritePin(LED_9_GPIO_Port, LED_9_Pin, SET );
239     }
240     else if(counter<55){
241     HAL_GPIO_WritePin(LED_10_GPIO_Port, LED_10_Pin, SET );
242     }
243     else if(counter<60){
244     HAL_GPIO_WritePin(LED_11_GPIO_Port, LED_11_Pin, SET );
245     }
246 }
247 int counter0=0;
248 int counter1 =0;
249 int counter2=0;
250 while (1){
251     second(counter0++);
252     minute(counter1);
253     hour(counter2);
254     if(counter0>=12) {counter0=0; counter1++;}
255     if(counter1>=60) {counter1=0; counter2++;}
256     if(counter2>=60) counter2=0;
257     HAL_Delay(50);
258 }
```