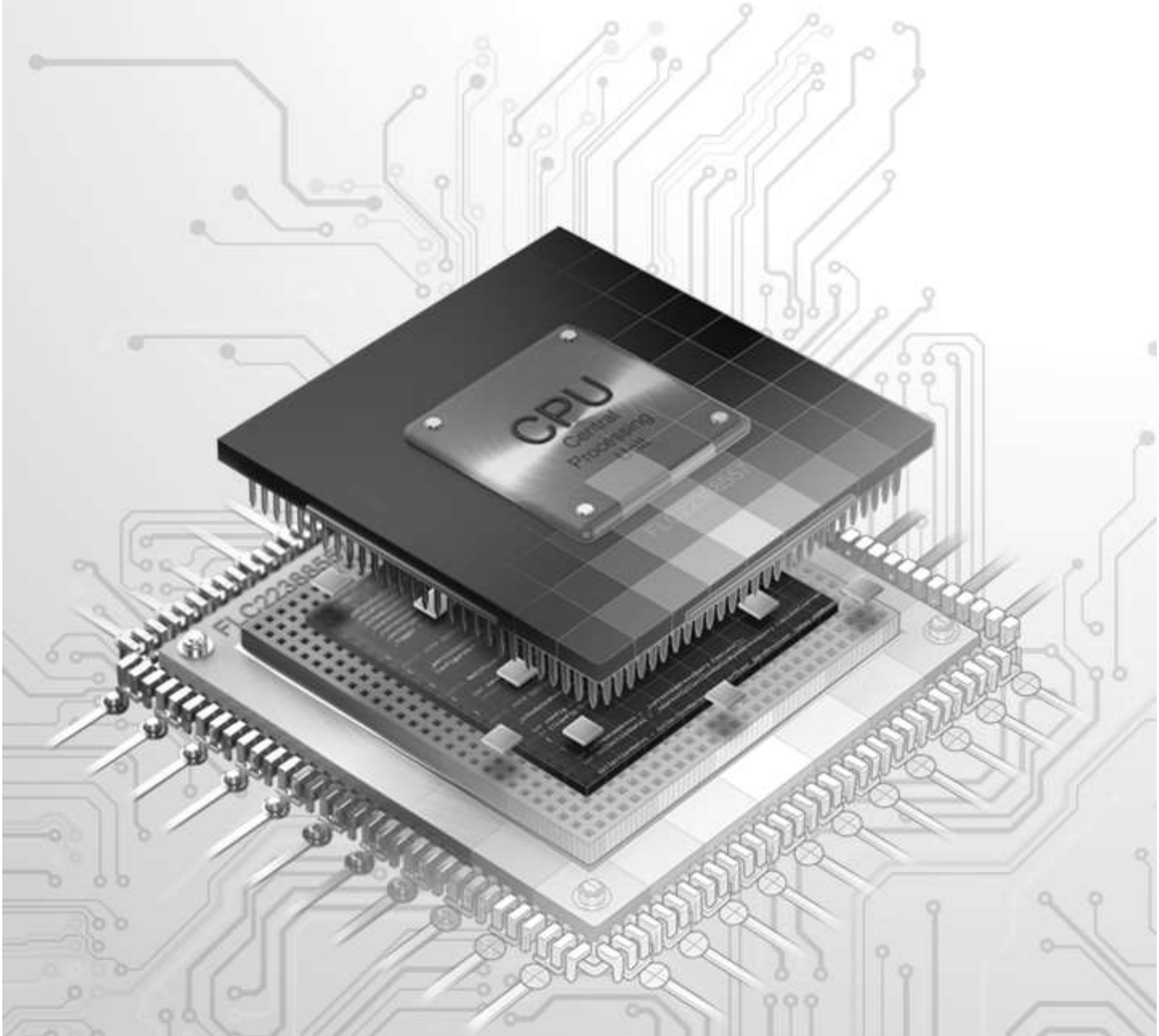
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller

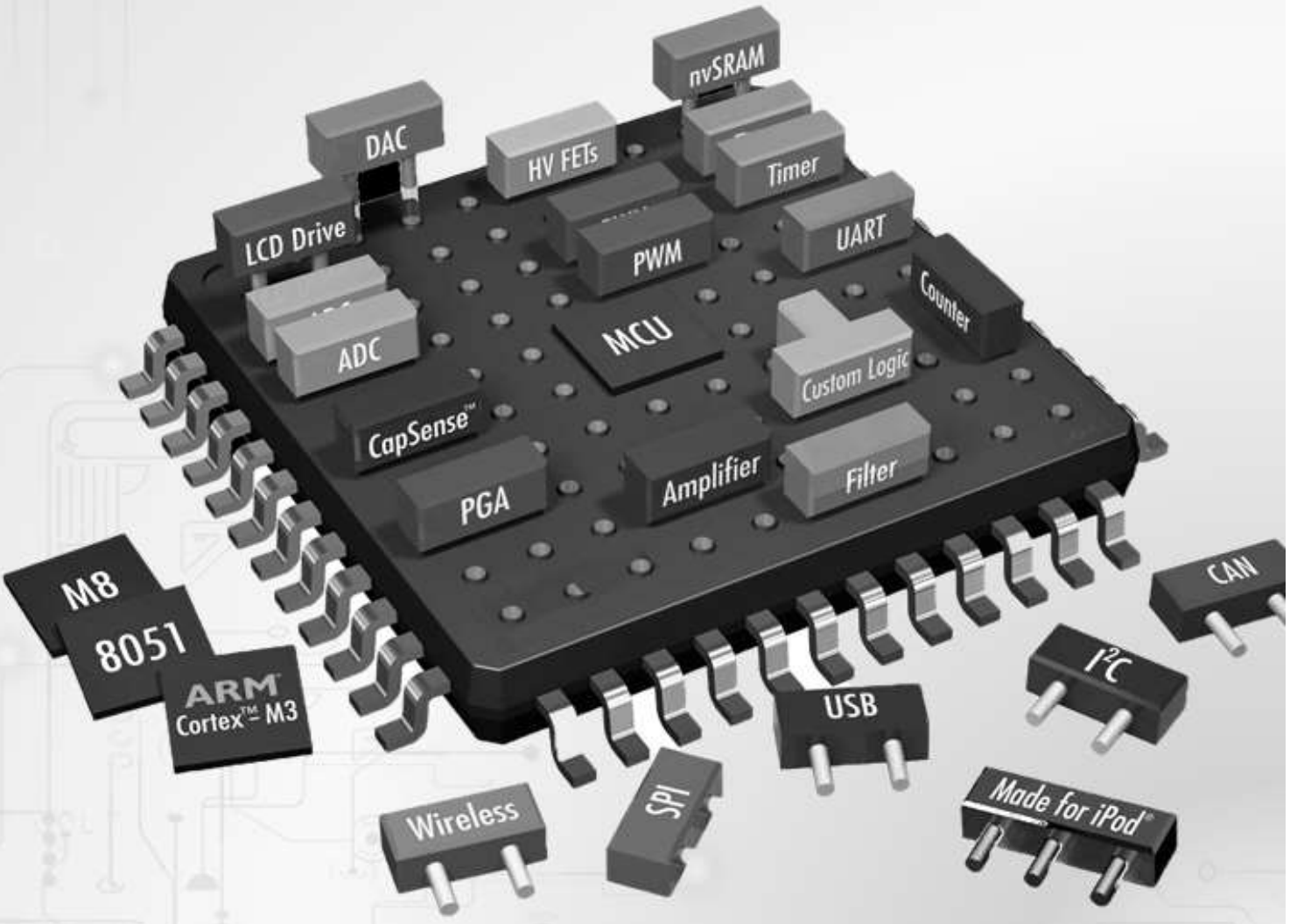


Dr. Le Trong Nhan

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

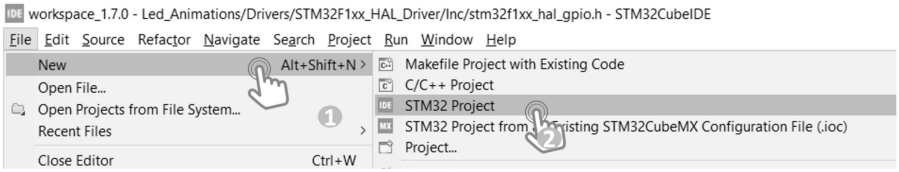
The rest of this manual consists of:

- 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

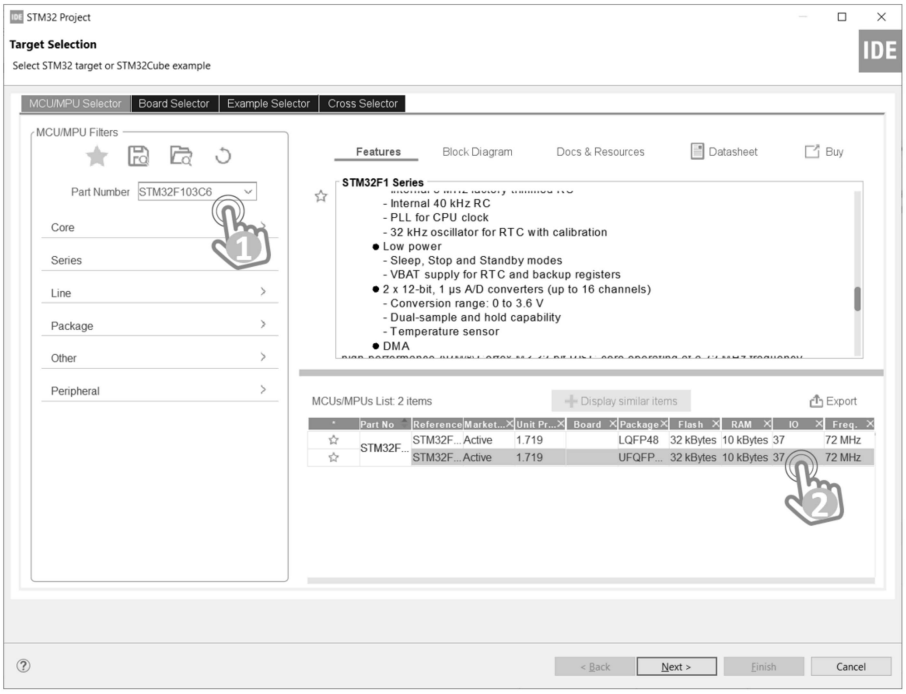
Step 1: Launch STM32CubeIDE, from the menu **File**, select **New**, then chose **STM32 Project**



Hình 1.2: Create a new project on STM32CubeIDE

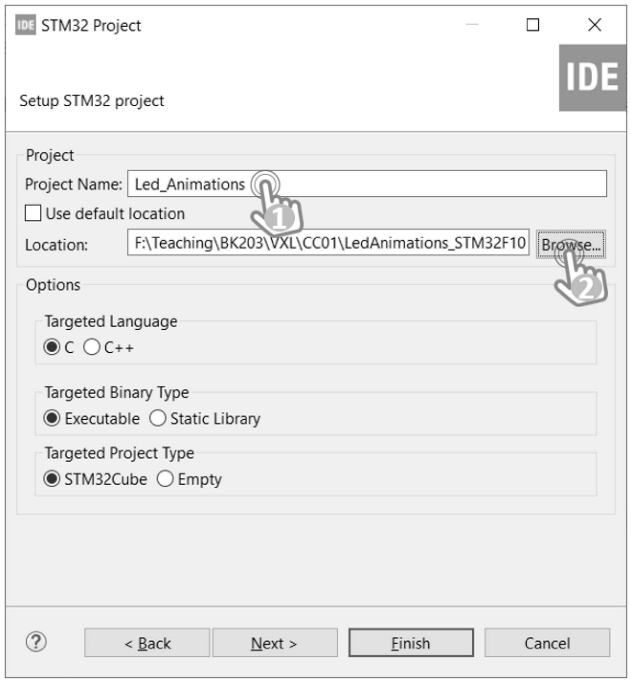
The IDE needs to download some packages, which normally takes time in this first time a project is created.

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.

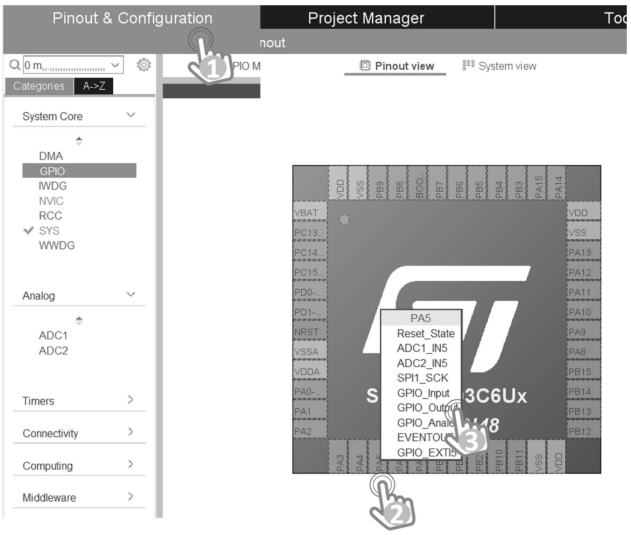
Step 6: Right click on PA5 and select **Enter user lable**, and provide the name for this pin (e.g. **LED_RED**). This step helps programming afterward more memorable.

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

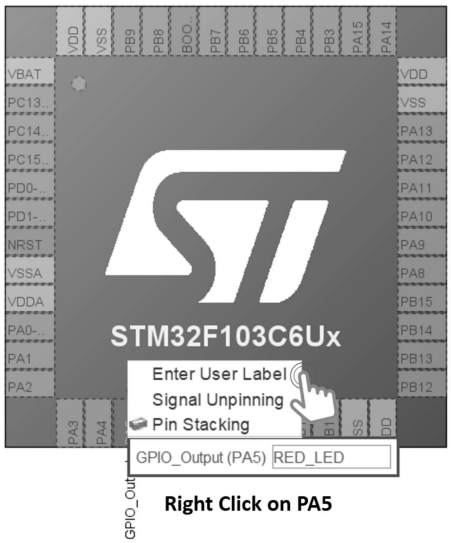
Step 7: Implement the first blinky project in the main function as follow:

```
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*

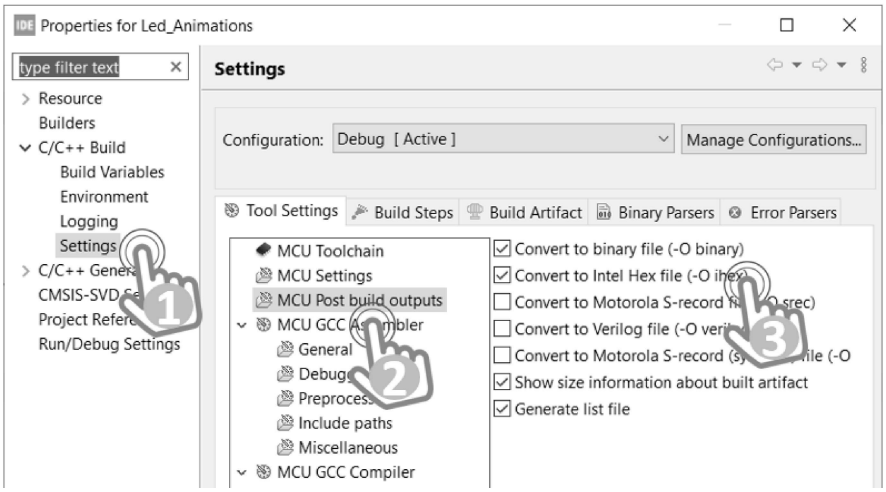
```
6
7   /* MCU Configuration
     ----------------------------------------------------------
     */
8
9   /* Reset of all peripherals, Initializes the Flash
     interface and the Systick. */
10  HAL_Init();
11
12  /* USER CODE BEGIN Init */
13
14  /* USER CODE END Init */
15
16  /* Configure the 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

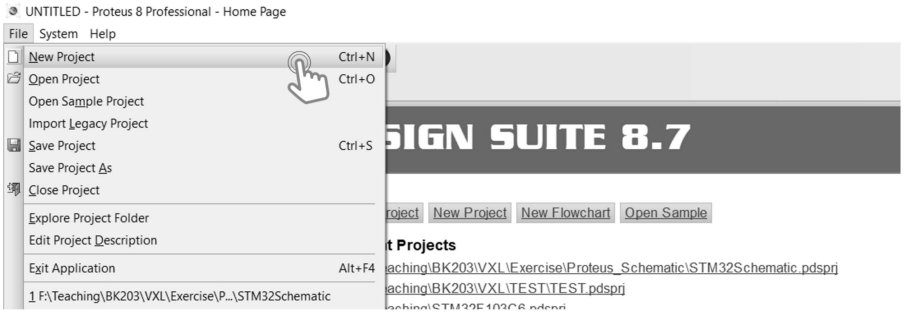
The hex file is located under the **Debug** folder of your project, which is used for the simulation in Proteus afterward. In the case a development kit is connected to your PC, from menu **Run**, select **Run** to download the program to the hardware platform.

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

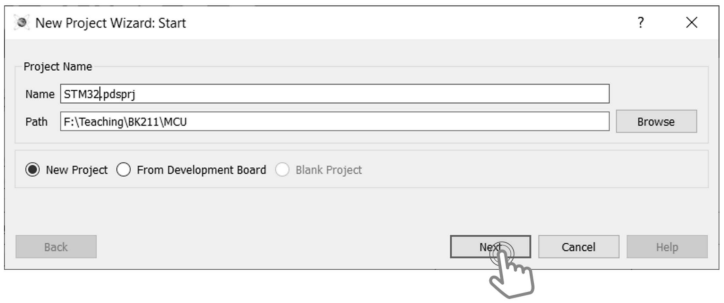
For an online training, a simulation on Proteus can be used. The details to create an STM32 project on Proteus are described bellow.

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



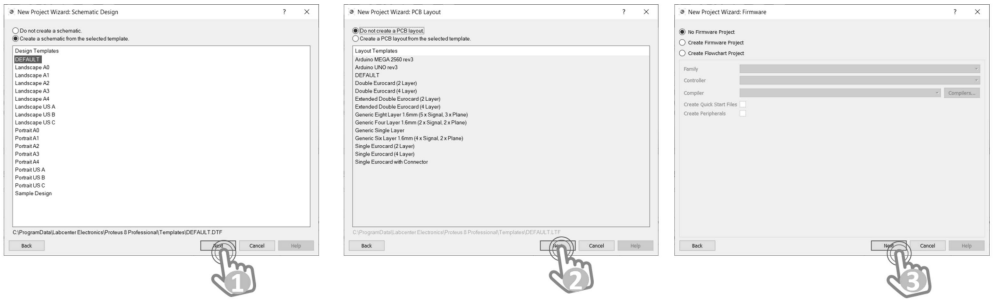
Hình 1.10: *Create a new project on Proteus*

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



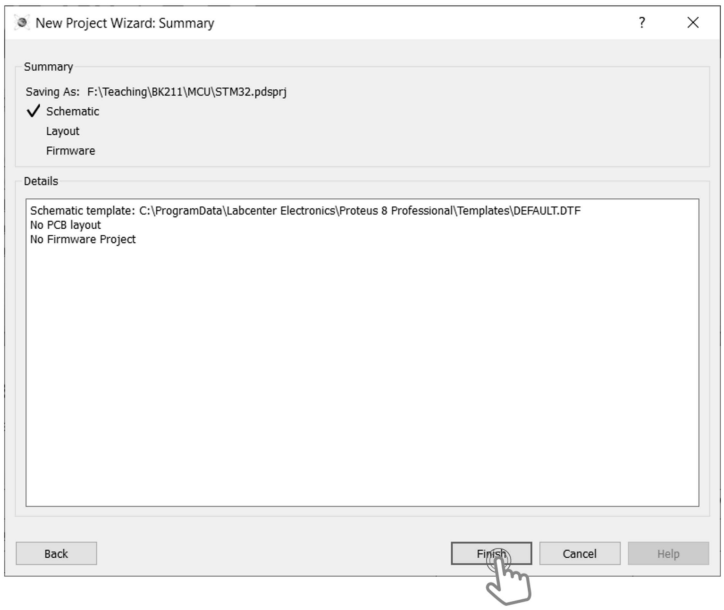
Hình 1.11: *Provide project name and location*

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



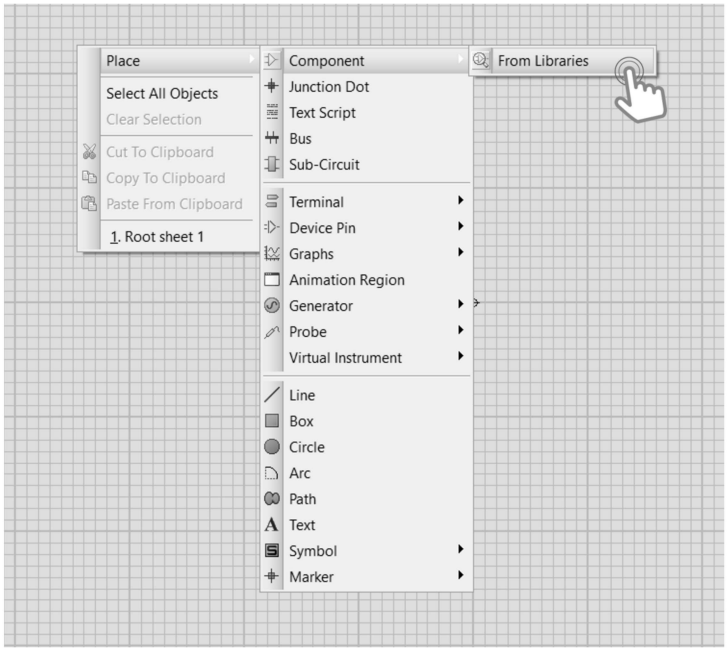
Hình 1.12: *Keep the default options by clicking on Next*

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



Hình 1.13: *Finish the project wizard*

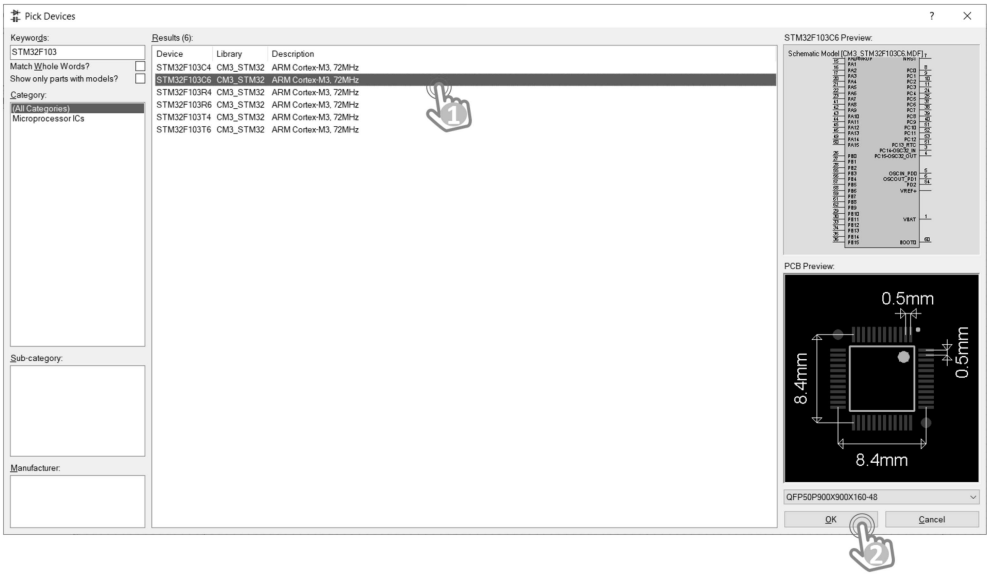
Step 5: On the main page of the project, right click to select **Place, Components, From Libraries**, as follows:



Hình 1.14: *Select a component from the library*

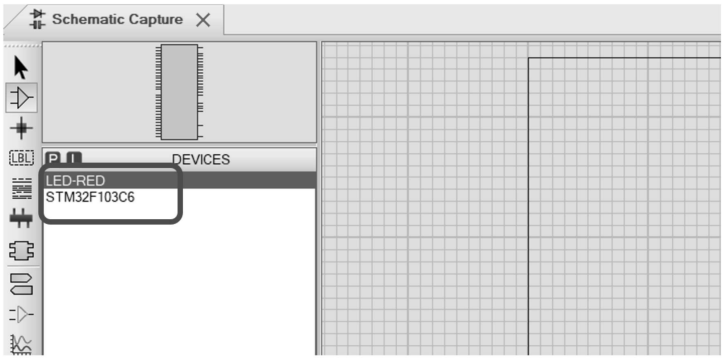
If there is an error with no library found, please restart the Proteus software with Run as administrator option.

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



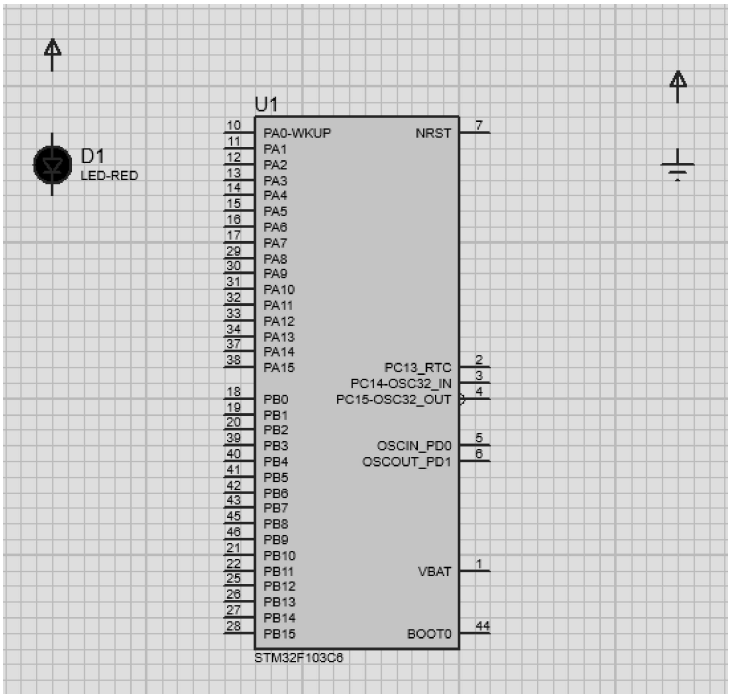
Hình 1.15: *Select STM32F103C6*

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:



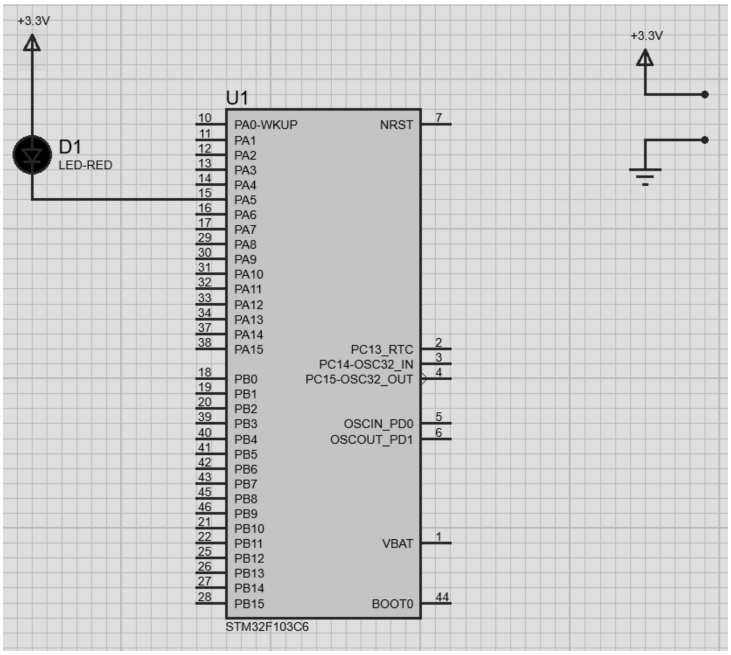
Hình 1.16: *STM32 and an LED in the project*

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*

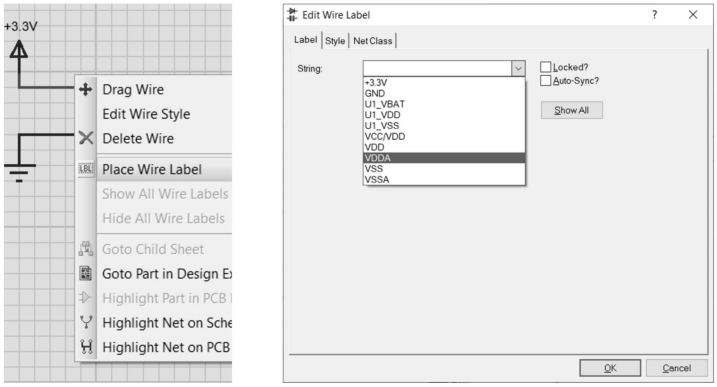
Step 8: Start wiring the circuit. The negative pin of the LED is connected to PA5 while its positive pin is connected to the power supply. For the power and the ground on the right, just make a short wire, which will labeled in the next step.



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

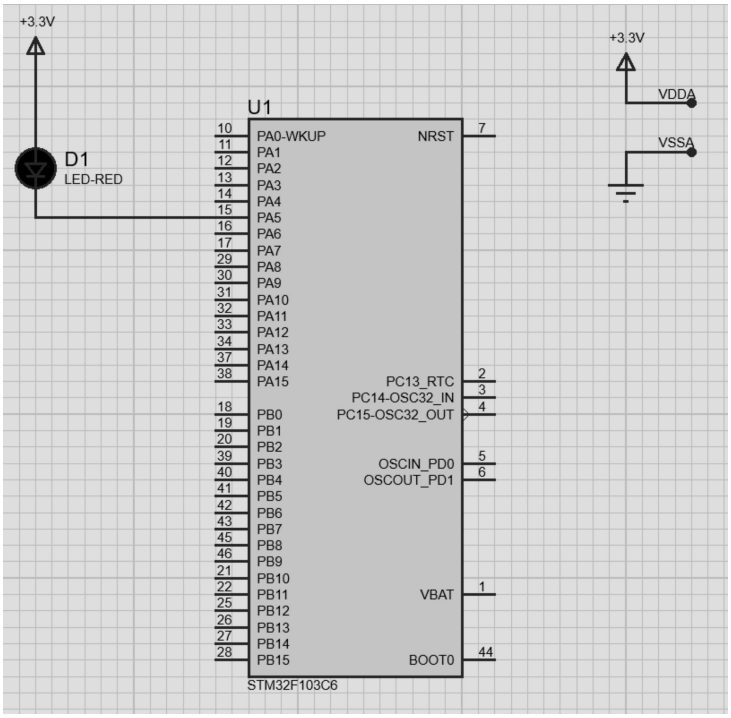
In this step, also double click on the power supply in order to provide the String property to **+3.3V**.

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



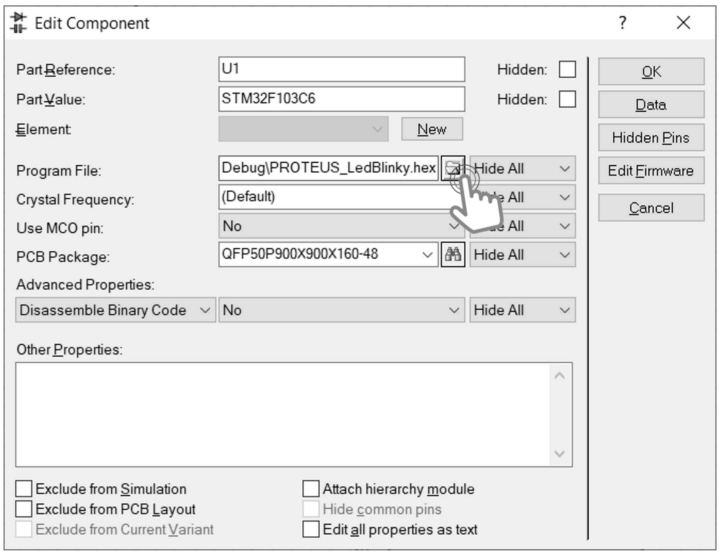
Hình 1.19: *Place label for Power and Ground*

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:



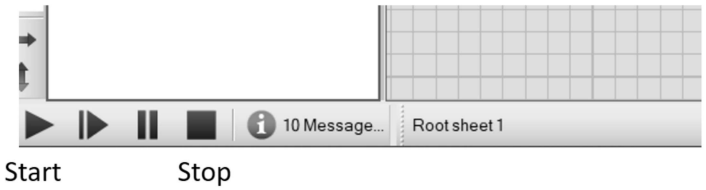
Hình 1.20: *Finalize the schematic*

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:

```
while (1){
  HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
   GPIO_PIN_SET);
  HAL_Delay(1000);
  HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
   GPIO_PIN_RESET);
  HAL_Delay(1000);
}
```

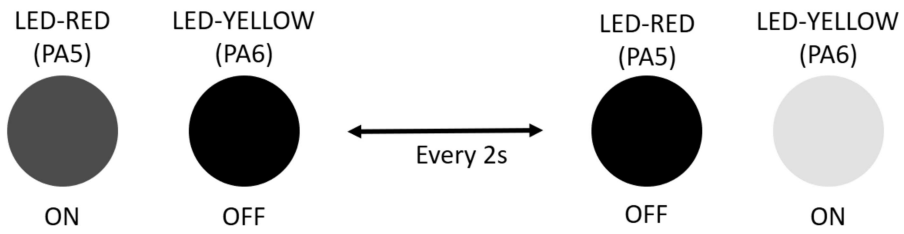
Program 1.2: An example for LED blinky

4 Exercise and Report

4.1 Exercise 1

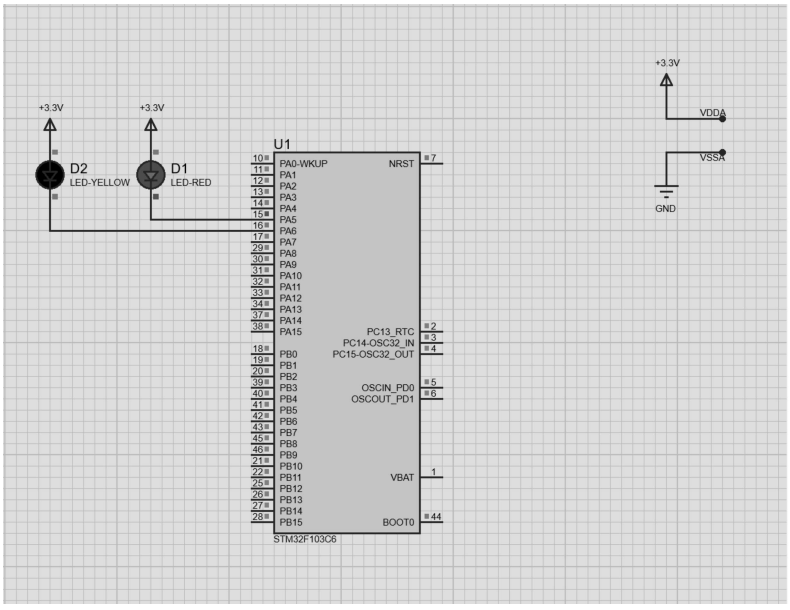
From the simulation on Proteus, one more LED is connected to pin **PA6** of the STM32 (negative pin of the LED is connected to PA6). The component suggested in this exercise is **LED-YELLOW**, which can be found from the device list.

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: *State transitions for 2 LEDs*

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

```
  int flag = 0;
  while (1)
  {
    if (flag < 2)
    {
      HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port,
   YELLOW_LED_Pin, SET);
      HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin,
   RESET);
      flag++;
    }
    else
    {
      HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port,
   YELLOW_LED_Pin, RESET);
      HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin, SET
   );
      flag++;
      if (flag == 4) flag = 0;
    }
    HAL_Delay(1000);
  }
  /* USER CODE END 3 */
}
```

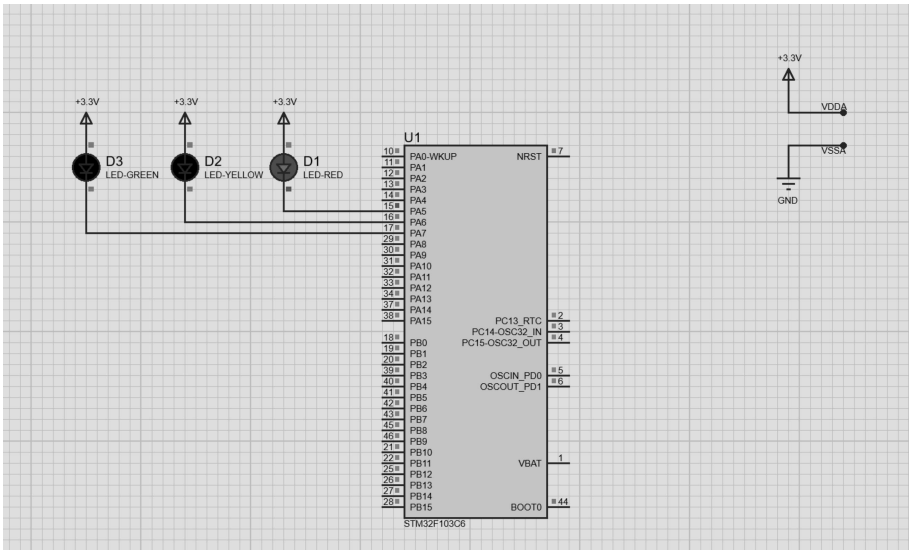
Program 1.3: Coding

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: *State transitions for 3 LEDs*

Report 2: Present the source code in while.

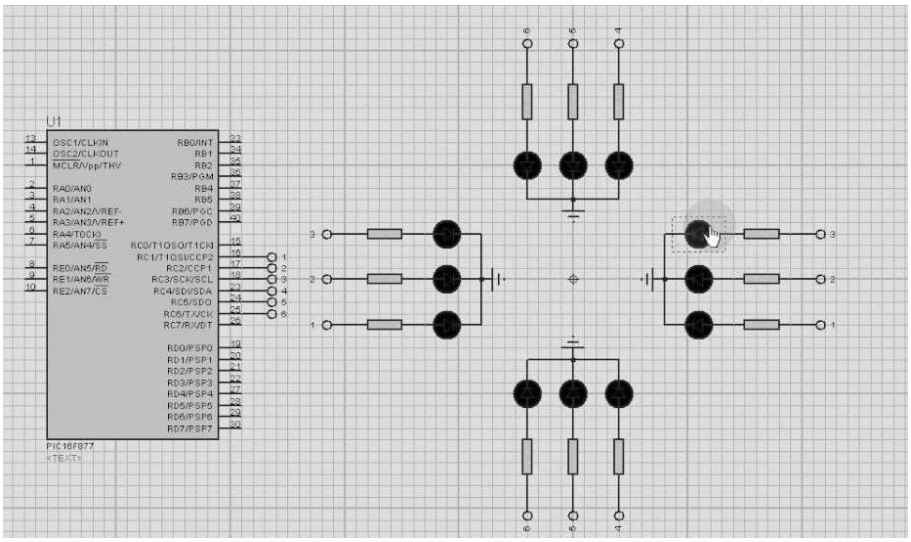
```
int flag_red = 0;
int flag_yellow = 0;
int flag_green = 0;
while (1)
{
//Red Led turns on
  if (flag_red <= 5 && flag_yellow == 0 && flag_green == 0)
  {
    HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin, RESET);
    HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port, YELLOW_LED_Pin, SET);
    HAL_GPIO_WritePin(GREEN_LED_GPIO_Port, GREEN_LED_Pin, SET);
    flag_red++;
  }

  //Green Led turns on
  if (flag_red > 5 && flag_yellow == 0 && flag_green <=3)
  {
    HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin, SET);
    HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port, YELLOW_LED_Pin, SET);
    HAL_GPIO_WritePin(GREEN_LED_GPIO_Port, GREEN_LED_Pin, RESET);
    flag_green++;
  }

```

```
    //Yellow Led turns on
    if (flag_red > 5 && flag_yellow <= 2 && flag_green > 3)
    {
      HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin, SET
  );
      HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port,
   YELLOW_LED_Pin, RESET);
      HAL_GPIO_WritePin(GREEN_LED_GPIO_Port, GREEN_LED_Pin,
    SET);
      flag_yellow ++;
        if (flag_red > 5 && flag_yellow > 2 && flag_green >
    3)
        {
          flag_red = 0;
          flag_yellow = 0;
          flag_green = 0;
        }
    }
    HAL_Delay(1000);
  }
```

Program 1.4: Coding

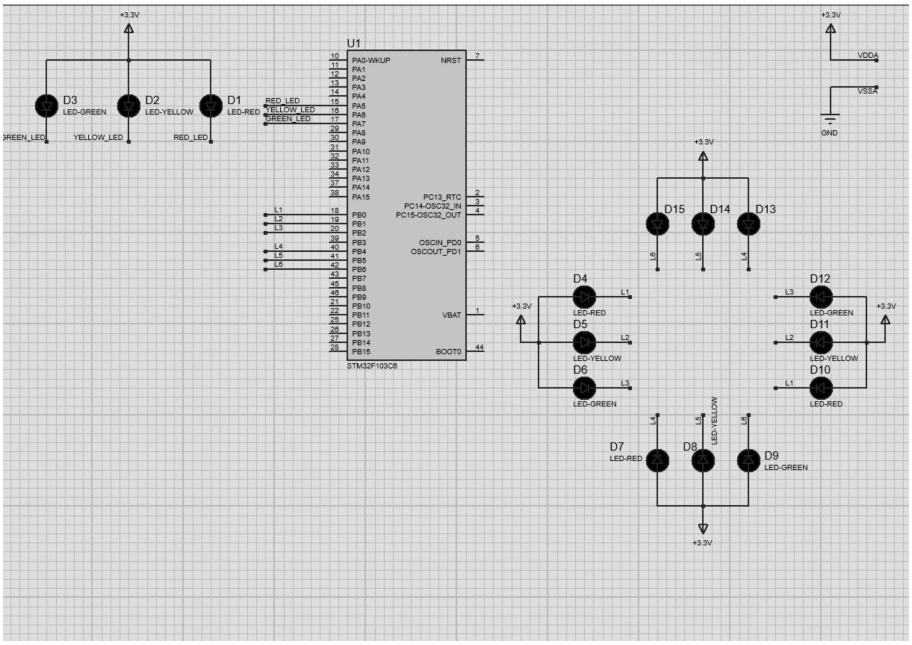
4.3 Exercise 3

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*

Report 1: Present the schematic.



Hình 1.27: *State transitions for 3 LEDs*

Report 2: Present the source code in while.

```
#define RED     0
#define YELLOW  1
#define GREEN   2

int flag = 0;
int led_color = RED;

while (1)
{
    switch(led_color)
    {
        case RED:
            HAL_GPIO_WritePin(RED_1_GPIO_Port, RED_1_Pin, RESET);
            HAL_GPIO_WritePin(YELLOW_1_GPIO_Port, YELLOW_1_Pin, SET);
            HAL_GPIO_WritePin(GREEN_1_GPIO_Port, GREEN_1_Pin, SET);

            HAL_GPIO_WritePin(RED_2_GPIO_Port, RED_2_Pin, SET);
            HAL_GPIO_WritePin(YELLOW_2_GPIO_Port, YELLOW_2_Pin, SET);
            HAL_GPIO_WritePin(GREEN_2_GPIO_Port, GREEN_2_Pin, RESET);
```

```
20              flag++;
21
22              if(flag > 3)
23              {
24                  HAL_GPIO_WritePin(RED_2_GPIO_Port,
    RED_2_Pin, SET);
25                  HAL_GPIO_WritePin(YELLOW_2_GPIO_Port,
    YELLOW_2_Pin, RESET);
26                  HAL_GPIO_WritePin(GREEN_2_GPIO_Port,
    GREEN_2_Pin, SET);
27              }
28
29              if (flag == 5)
30              {
31                  led_color = GREEN;
32                  flag = 0;
33              }
34              break;
35
36          case YELLOW:
37              HAL_GPIO_WritePin(RED_1_GPIO_Port, RED_1_Pin,
    SET);
38              HAL_GPIO_WritePin(YELLOW_1_GPIO_Port,
    YELLOW_1_Pin, RESET);
39              HAL_GPIO_WritePin(GREEN_1_GPIO_Port,
    GREEN_1_Pin, SET);
40
41              HAL_GPIO_WritePin(RED_2_GPIO_Port, RED_2_Pin,
    RESET);
42              HAL_GPIO_WritePin(YELLOW_2_GPIO_Port,
    YELLOW_2_Pin, SET);
43              HAL_GPIO_WritePin(GREEN_2_GPIO_Port,
    GREEN_2_Pin, SET);
44              flag++;
45
46              if (flag == 2)
47              {
48                  led_color = RED;
49                  flag = 0;
50              }
51              break;
52
53          case GREEN:
54              HAL_GPIO_WritePin(RED_1_GPIO_Port, RED_1_Pin,
    SET);
55              HAL_GPIO_WritePin(YELLOW_1_GPIO_Port,
    YELLOW_1_Pin, SET);
56              HAL_GPIO_WritePin(GREEN_1_GPIO_Port,
    GREEN_1_Pin, RESET);
```

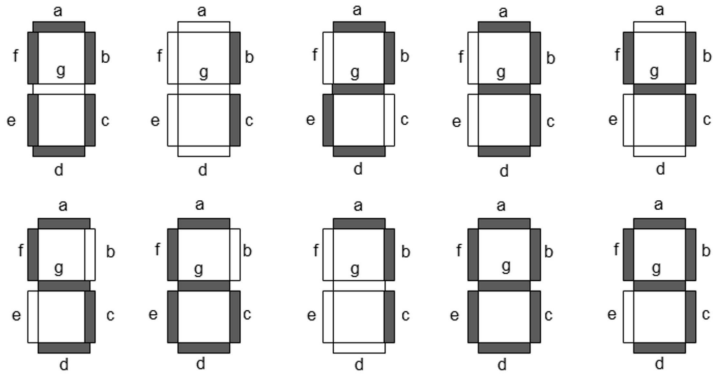
```
57
58              HAL_GPIO_WritePin(RED_2_GPIO_Port, RED_2_Pin,
    RESET);
59              HAL_GPIO_WritePin(YELLOW_2_GPIO_Port,
    YELLOW_2_Pin, SET);
60              HAL_GPIO_WritePin(GREEN_2_GPIO_Port,
    GREEN_2_Pin, SET);
61              flag++;
62
63              if (flag == 3)
64              {
65                  led_color = YELLOW;
66                  flag = 0;
67              }
68              break;
69
70          default:
71              break;
72      }
73      HAL_Delay(1000);
74  }
```

Program 1.5: Coding

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:



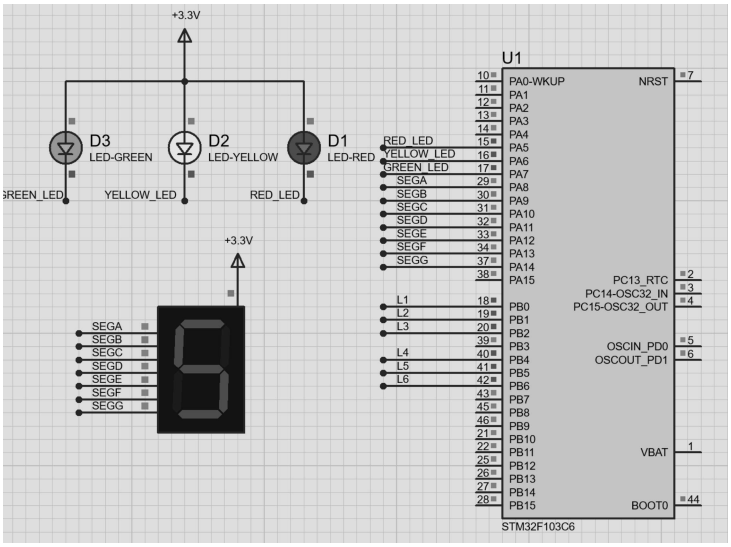
Hình 1.28: Display a number on 7 segment LED

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 1.6: An example for your source code

Report 1: Present the schematic.



Hình 1.29: Display a number on 7 segment LED

Report 2: Present the source code for display7SEG function.

```
#include "7SEG.h"
int counter = 0;
void display_7SEG (int num)
{
    //Turn off all of the segment first
    HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin, SET);
    HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin, SET);
    HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin, SET);
    HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin, SET);
    HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET);
    HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET);
    HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin, SET);

    switch(counter)
    {
        case 0:
```

```
17            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
    RESET);
18            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
19            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
20            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
    RESET);
21            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin,
    RESET);
22            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
    RESET);
23            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin, SET
    );
24            break;
25
26          case 1:
27            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin, SET
    );
28            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
29            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
30            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin, SET
    );
31            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
    );
32            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET
    );
33            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin, SET
    );
34            break;
35
36          case 2:
37            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
    RESET);
38            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
39            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin, SET
    );
40            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
    RESET);
41            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin,
    RESET);
42            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET
    );
43            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
    RESET);
44            break;
```

```
45
46          case 3:
47            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
    RESET);
48            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
49            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
50            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
    RESET);
51            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
    );
52            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET
    );
53            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
    RESET);
54            break;
55
56          case 4:
57            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin, SET
    );
58            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
59            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
60            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin, SET
    );
61            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
    );
62            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
    RESET);
63            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
    RESET);
64            break;
65
66          case 5:
67            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
    RESET);
68            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin, SET
    );
69            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
70            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
    RESET);
71            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
    );
72            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
    RESET);
73            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
```

```
RESET);
74            break;
75
76          case 6:
77            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
RESET);
78            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin, SET
);
79            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
RESET);
80            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
RESET);
81            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin,
RESET);
82            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
RESET);
83            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
RESET);
84            break;
85
86          case 7:
87            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
RESET);
88            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
RESET);
89            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
RESET);
90            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin, SET
);
91            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
);
92            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET
);
93            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin, SET
);
94            break;
95
96          case 8:
97            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
RESET);
98            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
RESET);
99            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
RESET);
100           HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
RESET);
101           HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin,
RESET);
102           HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
```

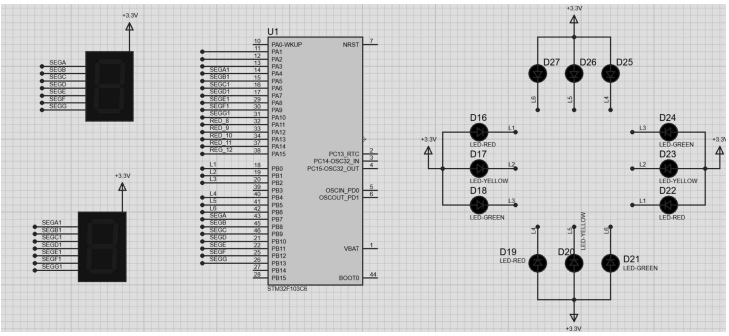
```
   RESET);
103          HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
   RESET);
104          break;
105
106        case 9:
107          HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
   RESET);
108          HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
   RESET);
109          HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
   RESET);
110          HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
   RESET);
111          HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
   );
112          HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
   RESET);
113          HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
   RESET);
114          break;
115
116        default:
117          break;
118    }
119
120 }
121
122   /* USER CODE BEGIN WHILE */
123   while (1)
124   {
125     if (counter >= 10) counter = 0;
126     display_7SEG(counter++);
127     HAL_Delay(1000);
128   }
129   /* USER CODE END 3 */
130 }
```

Program 1.7: An example for your source code

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.

Report 1: Present the schematic.



Hình 1.30: *Display a number on 7 segment LED*

Report 2: Present the source code for display7SEG function.

```
void display_7SEG (int num)
{
    //Turn off all of the segment first
    HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin, SET);
    HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin, SET);
    HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin, SET);
    HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin, SET);
    HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET);
    HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET);
    HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin, SET);

    switch(num)
    {
        case 0:
          HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
   RESET);
          HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
   RESET);
          HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
   RESET);
          HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
   RESET);
          HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin,
   RESET);
          HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
   RESET);
          HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin, SET
   );
          break;

        case 1:
```

```
25          HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin, SET
    );
26          HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
27          HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
28          HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin, SET
    );
29          HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
    );
30          HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET
    );
31          HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin, SET
    );
32          break;
33
34        case 2:
35          HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
    RESET);
36          HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
37          HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin, SET
    );
38          HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
    RESET);
39          HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin,
    RESET);
40          HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET
    );
41          HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
    RESET);
42          break;
43
44        case 3:
45          HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
    RESET);
46          HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
47          HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
48          HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
    RESET);
49          HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
    );
50          HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET
    );
51          HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
    RESET);
52          break;
```

```
53
54          case 4:
55            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin, SET
    );
56            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
    RESET);
57            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
58            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin, SET
    );
59            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
    );
60            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
    RESET);
61            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
    RESET);
62            break;
63
64          case 5:
65            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
    RESET);
66            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin, SET
    );
67            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
68            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
    RESET);
69            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
    );
70            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
    RESET);
71            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
    RESET);
72            break;
73
74          case 6:
75            HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
    RESET);
76            HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin, SET
    );
77            HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
    RESET);
78            HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
    RESET);
79            HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin,
    RESET);
80            HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
    RESET);
81            HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
```

```
  RESET);
82          break;
83
84        case 7:
85          HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
  RESET);
86          HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
  RESET);
87          HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
  RESET);
88          HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin, SET
  );
89          HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
  );
90          HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin, SET
  );
91          HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin, SET
  );
92          break;
93
94        case 8:
95          HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
  RESET);
96          HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
  RESET);
97          HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
  RESET);
98          HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
  RESET);
99          HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin,
  RESET);
100         HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
  RESET);
101         HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
  RESET);
102         break;
103
104       case 9:
105         HAL_GPIO_WritePin(SEG_A_GPIO_Port, SEG_A_Pin,
  RESET);
106         HAL_GPIO_WritePin(SEG_B_GPIO_Port, SEG_B_Pin,
  RESET);
107         HAL_GPIO_WritePin(SEG_C_GPIO_Port, SEG_C_Pin,
  RESET);
108         HAL_GPIO_WritePin(SEG_D_GPIO_Port, SEG_D_Pin,
  RESET);
109         HAL_GPIO_WritePin(SEG_E_GPIO_Port, SEG_E_Pin, SET
  );
110         HAL_GPIO_WritePin(SEG_F_GPIO_Port, SEG_F_Pin,
```

```
   RESET);
111           HAL_GPIO_WritePin(SEG_G_GPIO_Port, SEG_G_Pin,
   RESET);
112           break;
113
114         default:
115           break;
116     }
117
118 }
119
120 void display_7SEG1 (int num1)
121 {
122     //Turn off all of the segment first
123     HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, SET);
124     HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, SET);
125     HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, SET);
126     HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, SET);
127     HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET);
128     HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET);
129     HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, SET);
130
131     switch(num1)
132     {
133         case 0:
134           HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin,
   RESET);
135           HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin,
   RESET);
136           HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
   RESET);
137           HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin,
   RESET);
138           HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin,
   RESET);
139           HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin,
   RESET);
140           HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, SET
   );
141           break;
142
143         case 1:
144           HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, SET
   );
145           HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin,
   RESET);
146           HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
   RESET);
147           HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, SET
```

```
);
148           HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET
);
149           HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET
);
150           HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, SET
);
151           break;
152
153
154         case 2:
155           HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin,
RESET);
156           HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin,
RESET);
157           HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, SET
);
158           HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin,
RESET);
159           HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin,
RESET);
160           HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET
);
161           HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin,
RESET);
162           break;
163
164         case 3:
165           HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin,
RESET);
166           HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin,
RESET);
167           HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
RESET);
168           HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin,
RESET);
169           HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET
);
170           HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET
);
171           HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin,
RESET);
172           break;
173
174         case 4:
175           HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, SET
);
176           HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin,
RESET);
```

```
177           HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
    RESET);
178           HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, SET
    );
179           HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET
    );
180           HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin,
    RESET);
181           HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin,
    RESET);
182           break;
183
184
185         case 5:
186           HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin,
    RESET);
187           HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, SET
    );
188           HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
    RESET);
189           HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin,
    RESET);
190           HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET
    );
191           HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin,
    RESET);
192           HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin,
    RESET);
193           break;
194
195         case 6:
196           HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin,
    RESET);
197           HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, SET
    );
198           HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
    RESET);
199           HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin,
    RESET);
200           HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin,
    RESET);
201           HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin,
    RESET);
202           HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin,
    RESET);
203           break;
204
205         case 7:
206           HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin,
```

```
  RESET);
207         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin,
  RESET);
208         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
  RESET);
209         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, SET
  );
210         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET
  );
211         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET
  );
212         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, SET
  );
213         break;
214
215       case 8:
216         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin,
  RESET);
217         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin,
  RESET);
218         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
  RESET);
219         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin,
  RESET);
220         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin,
  RESET);
221         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin,
  RESET);
222         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin,
  RESET);
223         break;
224
225       case 9:
226         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin,
  RESET);
227         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin,
  RESET);
228         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin,
  RESET);
229         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin,
  RESET);
230         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET
  );
231         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin,
  RESET);
232         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin,
  RESET);
233         break;
234
```

```
235         default:
236           break;
237     }
238 
239 }
240 
241 
242 #define RED     0
243 #define YELLOW  1
244 #define GREEN   2
245 
246 int flag_1 = 0;
247 int flag_2 = 0;
248 int countdown_1 = 0;
249 int countdown_2 = 0;
250 int led_color_1 = RED;
251 int led_color_2 = GREEN;
252 
253 while (1)
254 {
255     //  B     n   1
256     switch(led_color_1)  //  K i m  tra  t r ng  th i  c a
        n   1
257     {
258         case RED:
259             HAL_GPIO_WritePin(RED_1_GPIO_Port, RED_1_Pin,
    RESET);
260             HAL_GPIO_WritePin(YELLOW_1_GPIO_Port,
    YELLOW_1_Pin, SET);
261             HAL_GPIO_WritePin(GREEN_1_GPIO_Port,
    GREEN_1_Pin, SET);
262 
263             countdown_1 = 5 - flag_1;
264             display_7SEG(countdown_1);
265 
266             flag_1++;
267             if(flag_1 == 5)
268             {
269                 led_color_1 = GREEN;
270                 flag_1 = 0;
271             }
272             break;
273 
274         case YELLOW:
275             HAL_GPIO_WritePin(RED_1_GPIO_Port, RED_1_Pin,
    SET);
276             HAL_GPIO_WritePin(YELLOW_1_GPIO_Port,
    YELLOW_1_Pin, RESET);
277             HAL_GPIO_WritePin(GREEN_1_GPIO_Port,
```

```
GREEN_1_Pin, SET);

            countdown_1 = 2 - flag_1;
            display_7SEG(countdown_1);

            flag_1++;
            if (flag_1 == 2)
            {
                led_color_1 = RED;
                flag_1 = 0;
            }
            break;

        case GREEN:
            HAL_GPIO_WritePin(RED_1_GPIO_Port, RED_1_Pin,
SET);
            HAL_GPIO_WritePin(YELLOW_1_GPIO_Port,
YELLOW_1_Pin, SET);
            HAL_GPIO_WritePin(GREEN_1_GPIO_Port,
GREEN_1_Pin, RESET);

            countdown_1 = 3 - flag_1;
            display_7SEG(countdown_1);

            flag_1++;
            if (flag_1 == 3)
            {
                led_color_1 = YELLOW;
                flag_1 = 0;
            }
            break;

        default:
            break;
     }


     switch(led_color_2)
     {
         case RED:
            HAL_GPIO_WritePin(RED_2_GPIO_Port, RED_2_Pin,
RESET);
            HAL_GPIO_WritePin(YELLOW_2_GPIO_Port,
YELLOW_2_Pin, SET);
            HAL_GPIO_WritePin(GREEN_2_GPIO_Port,
GREEN_2_Pin, SET);

            countdown_2 = 5 - flag_2;
            display_7SEG1(countdown_2);
```

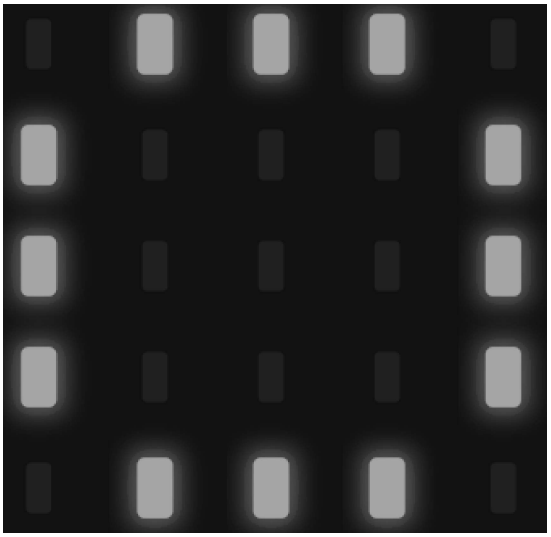
```
320
321             flag_2++;
322             if(flag_2 == 5)
323             {
324                 led_color_2 = GREEN;
325                 flag_2 = 0;
326             }
327             break;
328
329         case YELLOW:
330             HAL_GPIO_WritePin(RED_2_GPIO_Port, RED_2_Pin,
    SET);
331             HAL_GPIO_WritePin(YELLOW_2_GPIO_Port,
    YELLOW_2_Pin, RESET);
332             HAL_GPIO_WritePin(GREEN_2_GPIO_Port,
    GREEN_2_Pin, SET);
333
334             countdown_2 = 2 - flag_2;
335             display_7SEG1(countdown_2);
336
337             flag_2++;
338             if (flag_2 == 2)
339             {
340                 led_color_2 = RED;
341                 flag_2 = 0;
342             }
343             break;
344
345         case GREEN:
346             HAL_GPIO_WritePin(RED_2_GPIO_Port, RED_2_Pin,
    SET);
347             HAL_GPIO_WritePin(YELLOW_2_GPIO_Port,
    YELLOW_2_Pin, SET);
348             HAL_GPIO_WritePin(GREEN_2_GPIO_Port,
    GREEN_2_Pin, RESET);
349
350             countdown_2 = 3 - flag_2;
351             display_7SEG1(countdown_2);
352
353             flag_2++;
354             if (flag_2 == 3)
355             {
356                 led_color_2 = YELLOW;
357                 flag_2 = 0;
358             }
359             break;
360
361         default:
362             break;
```

```
363     }
364
365     HAL_Delay(1000);
```

Program 1.8: Source code

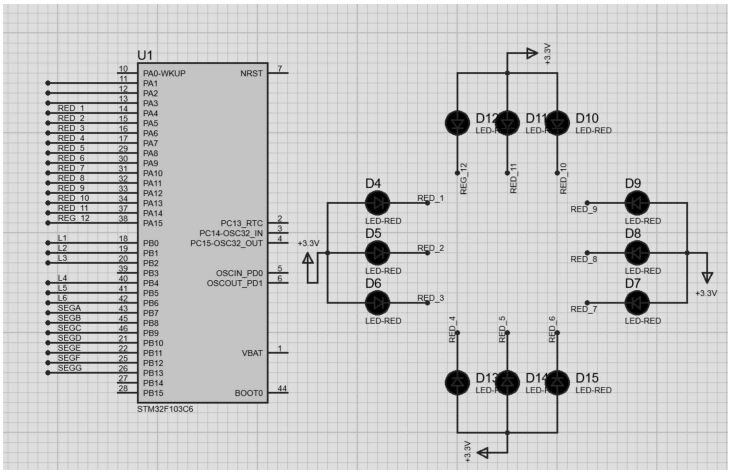
4.6 Exercise 6

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 1.31: *12 LEDs for an analog clock*

Report 1: Present the schematic.



Hình 1.32: *Display a number on 7 segment LED*

Report 2: Present the source code

```
int main(void)
{
   HAL_Init();
   SystemClock_Config();
   MX_GPIO_Init();
  while (1)
  {
   for (int pin = 4; pin <= 15; pin++)
    {
          // Turn on the current LED (PA pin)
          HAL_GPIO_WritePin(GPIOA, 1 << pin, GPIO_PIN_RESET
   );
          HAL_Delay(1000);  // 1s delay to observe the LED

          // Turn off the current LED
          HAL_GPIO_WritePin(GPIOA, 1 << pin, GPIO_PIN_SET);
    }
  }
}



static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
/* USER CODE BEGIN MX_GPIO_Init_1 */
/* USER CODE END MX_GPIO_Init_1 */

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOA_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOA, RED_1A_Pin|RED_2A_Pin|RED_3_Pin|
   RED_4_Pin
                          |RED_5_Pin|RED_6_Pin|RED_7_Pin|
   RED_8_Pin
                          |RED_9_Pin|RED_10_Pin|RED_11_Pin|
   RED_12_Pin, GPIO_PIN_SET);       //Initial State is OFF.

  /*Configure GPIO pins : RED_1A_Pin RED_2A_Pin RED_3_Pin
   RED_4_Pin
                           RED_5_Pin RED_6_Pin RED_7_Pin
   RED_8_Pin
                           RED_9_Pin RED_10_Pin RED_11_Pin
   RED_12_Pin */
  GPIO_InitStruct.Pin = RED_1A_Pin|RED_2A_Pin|RED_3_Pin|
   RED_4_Pin
                          |RED_5_Pin|RED_6_Pin|RED_7_Pin|
```

```
    RED_8_Pin
                            |RED_9_Pin|RED_10_Pin|RED_11_Pin|
    RED_12_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/* USER CODE BEGIN MX_GPIO_Init_2 */
/* USER CODE END MX_GPIO_Init_2 */
}
```

Program 1.9: Source code

4.7 Exercise 7

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

```
  void clearAllClock()
  {
    for (int pin = 4; pin <= 15; pin++)
    {
              // Turn off the current LED (PA pin)
              HAL_GPIO_WritePin(GPIOA, 1 << pin,
    GPIO_PIN_SET);
              HAL_Delay(1000);  // 1s delay to observe the
    LED
    }
  }


  while (1)
  {
   clearAllClock();
  }
}
```

Program 1.10: Function Implementation

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.

```
    int pin = 4;
    void SetNumberOnClock(int num)
    {
```

```
    pin = 4 + num;  // Map num to the corresponding GPIO pin (PA4 to PA15)
    HAL_GPIO_WritePin(GPIOA, 1 << pin, GPIO_PIN_RESET); // Turn on the corresponding LED
    HAL_Delay(1000);  // Delay to observe the LED
  }
 while (1)
 {
  for (int num = 0; num < 12; num++)
  {
    SetNumberOnClock(num);
  }
 }
```

Program 1.11: Function Implementation

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.

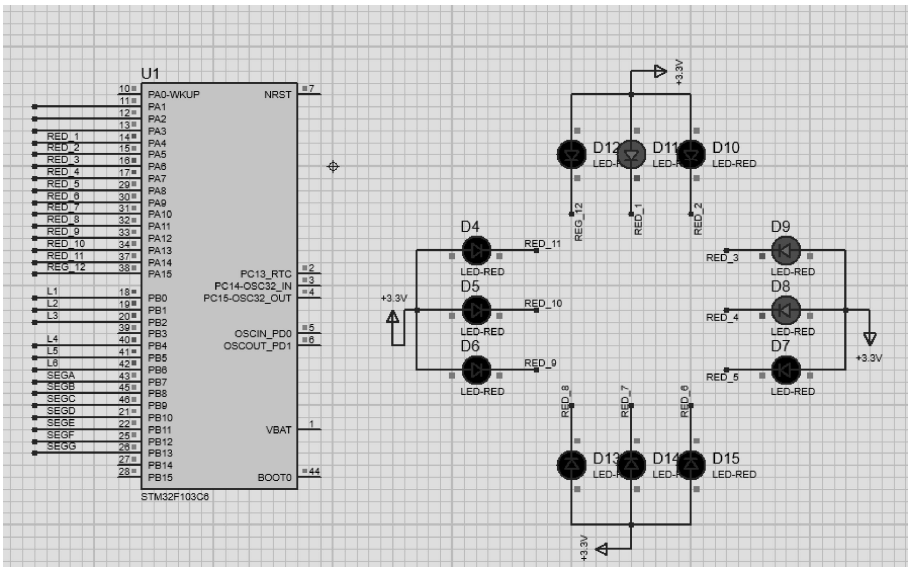
```
  int pin = 4;
  void clearNumberOnClock(int num)
  {
  pin = 4 + num;  // Map num to the corresponding GPIO pin (PA4 to PA15)
  HAL_GPIO_WritePin(GPIOA, 1 << pin, GPIO_PIN_SET);  // Turn off the corresponding LED
    }
  while (1)
  {
  for (int num = 0; num < 12; num++)
  {
    clearNumberOnClock(num);
  }
  }
```

Program 1.12: Function Implementation

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.

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 1.33: State transitions for 2 LEDs

Report 2: Present the source code

```
// Global variable for tracking the current LED position (used for simplicity)
int pin = 4;  // Start with pin 4 (PA4)

void SetNumberOnClock(int num)
{
   pin = 4 + num;  // Map num to the corresponding GPIO pin (PA4 to PA15)
    HAL_GPIO_WritePin(GPIOA, 1 << pin, GPIO_PIN_RESET);  // Turn on the corresponding LED
}

void clearNumberOnClock(int num)
{
    pin = 4 + num;  // Map num to the corresponding GPIO pin (PA4 to PA15)
    HAL_GPIO_WritePin(GPIOA, 1 << pin, GPIO_PIN_SET);  // Turn off the corresponding LED
}

// Function to map a value (hour, minute, or second) to a 12-position clock
int mapToClockPosition(int value, int maxValue) {
```

```
21          return (value * 12) / maxValue;  // Maps 0-maxValue
      to 0-11
22      }
23
24    // Function to turn on appropriate LEDs for hour, minute,
      and second
25    void displayTime(int hour, int minute, int second) {
26        // Clear all LEDs first
27        for (int i = 0; i < 12; i++) {
28            clearNumberOnClock(i);  // Turn off all LEDs
29        }
30
31        // Map hour, minute, and second to clock positions
     (0-11)
32        int hourPosition = mapToClockPosition(hour, 12);
33        int minutePosition = mapToClockPosition(minute, 60);
34        int secondPosition = mapToClockPosition(second, 60);
35
36        // Set the LEDs for hour, minute, and second
37        SetNumberOnClock(hourPosition);
38        SetNumberOnClock(minutePosition);
39        SetNumberOnClock(secondPosition);
40    }
41
42    int hour = 0;
43    int minute = 0;
44    int second = 0;
45    while (1)
46    {
47      // Update the display with the current time
48      displayTime(hour, minute, second);
49      // Simulate time progression
50              HAL_Delay(1000);  // 1 second delay
51              second++;
52
53              if (second >= 60) {
54                  second = 0;
55                  minute++;
56              }
57              if (minute >= 60) {
58                  minute = 0;
59                  hour++;
60              }
61              if (hour >= 12) {
62                  hour = 0;
63              }
64    }
65  }
```

Program 1.13: Source Code

- **Name**: HOANG VAN PHI
- **MSSV**: 2252608
- **Class**: CO3010_CC01
- **Teacher**: NGUYEN THIEN AN
- **Lab**: 1
- **Link GitHub**: https://github.com/PhiuTheWind/LAB1$_VXL.git$