

Name : Phạm Văn Bách

Student ID : 2252057

Class : CC01

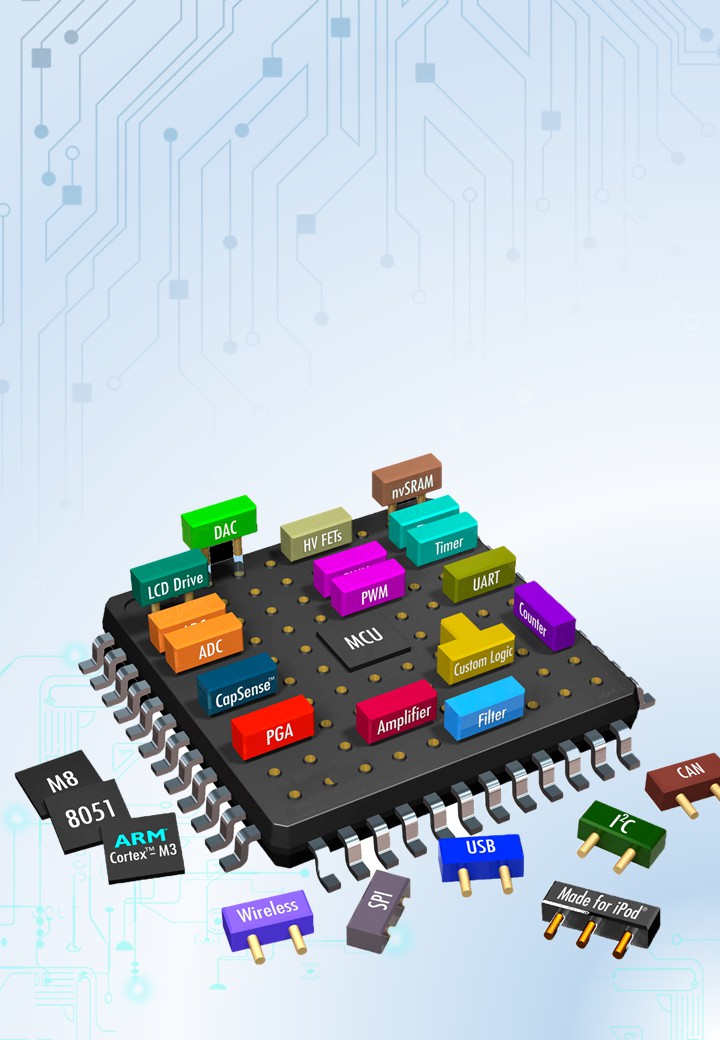
##### Dr. Le Trong Nhan

**Contents**

[Chapter 1. LED Animations](#_bookmark0) 7

1. [Introduction](#_bookmark1) 8
2. [First project on STM32Cube](#_bookmark2) 9
3. [Simulation on Proteus](#_bookmark3) 14
4. [Exercise and Report](#_bookmark4) 20

|  |  |  |  |
| --- | --- | --- | --- |
| [4.1](#_bookmark5) | [Exercise 1](#_bookmark5) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
| [4.2](#_bookmark6) | [Exercise 2](#_bookmark6) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
| [4.3](#_bookmark7) | [Exercise 3](#_bookmark7) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 21 |
| [4.4](#_bookmark8) | [Exercise 4](#_bookmark8) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 21 |
| [4.5](#_bookmark9) | [Exercise 5](#_bookmark9) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 22 |
| [4.6](#_bookmark10) | [Exercise 6](#_bookmark10) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 22 |
| [4.7](#_bookmark11) | [Exercise 7](#_bookmark11) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 23 |
| [4.8](#_bookmark12) | [Exercise 8](#_bookmark12) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 23 |
| [4.9](#_bookmark13) | [Exercise 9](#_bookmark13) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 23 |
| [4.10](#_bookmark14) | [Exercise 10](#_bookmark14) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 23 |



**CHAPTER 1**

**LED Animations**

# Introduction

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



*Figure 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

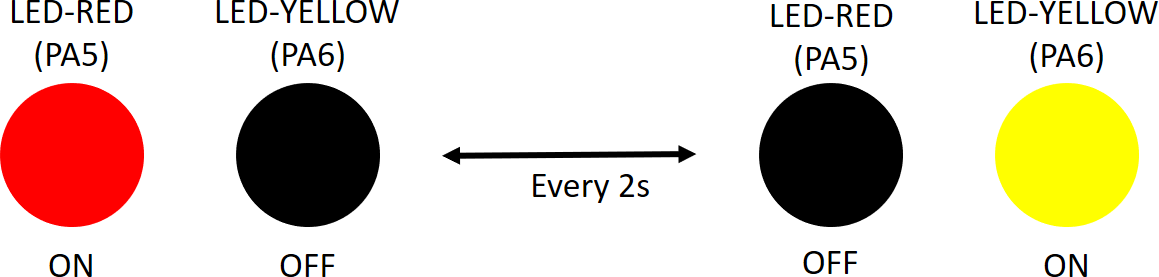
Finally, students are supposed to finalize 10 different proje

# Exercise and Report

## 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.



*Figure 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).

**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.

1

while (1) {

HAL\_GPIO\_Toggle Pin ( GPIOA , GPIO\_PIN\_ 5 );

HAL\_Delay ( 1000) ;

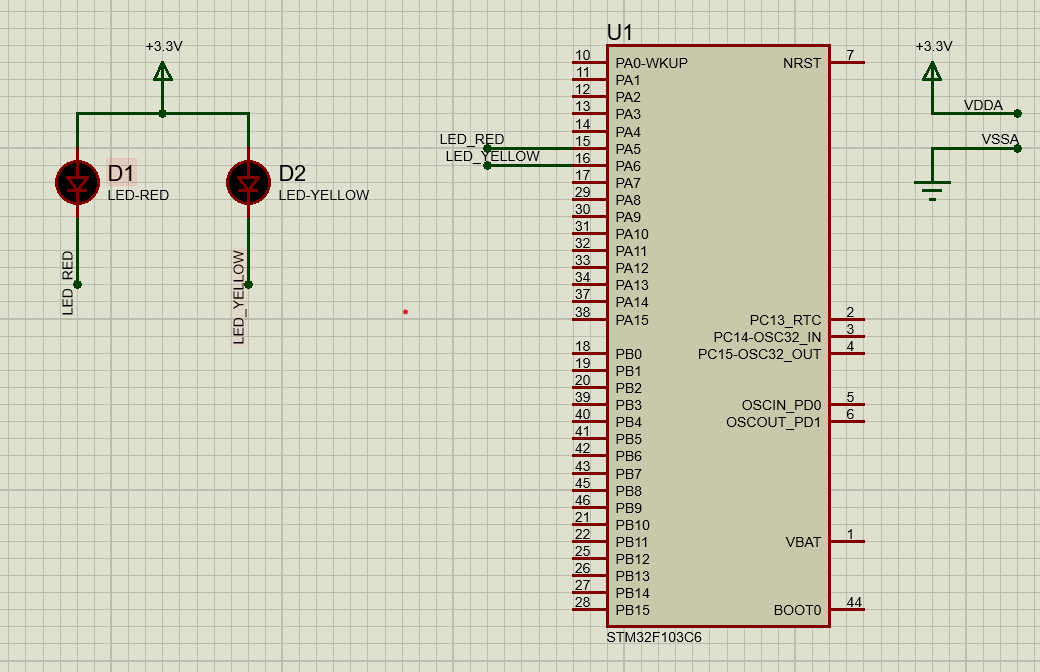
}

2

3

4

Program 1.3: An example for your source code

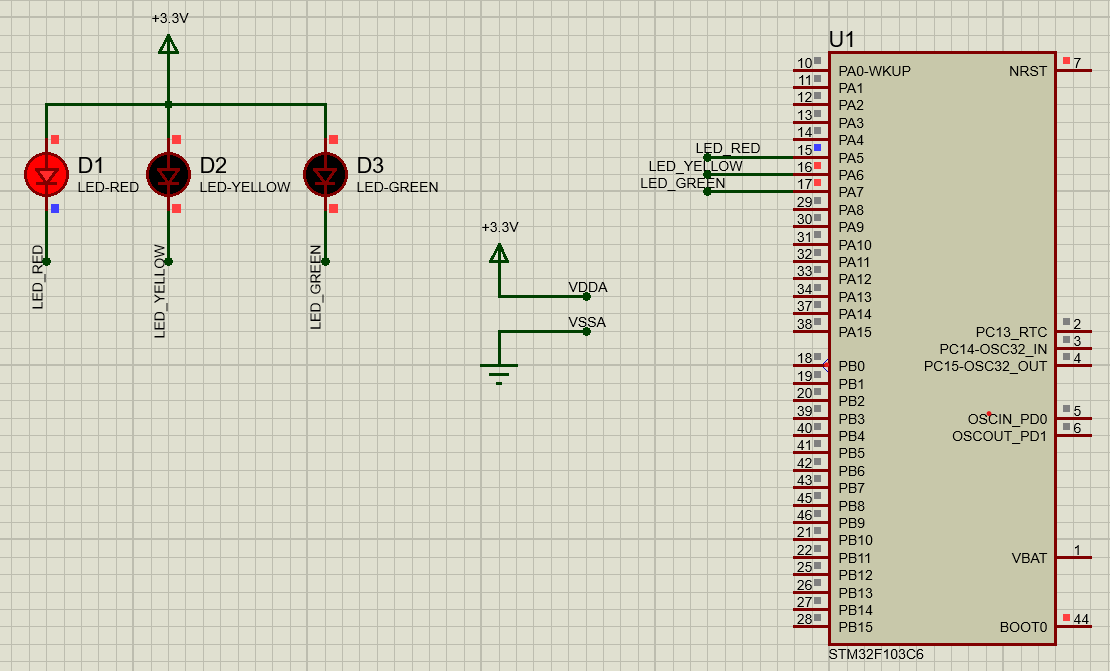


## Exercise 2

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.



**Report 2:** Present the source code in while.

while (1)

{

/\* USER CODE END WHILE \*/

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, GPIO\_PIN\_SET);

HAL\_Delay(5000);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, GPIO\_PIN\_SET);

HAL\_Delay(2000);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5,GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6,GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7,GPIO\_PIN\_RESET);

HAL\_Delay(3000);

/\* USER CODE BEGIN 3 \*/

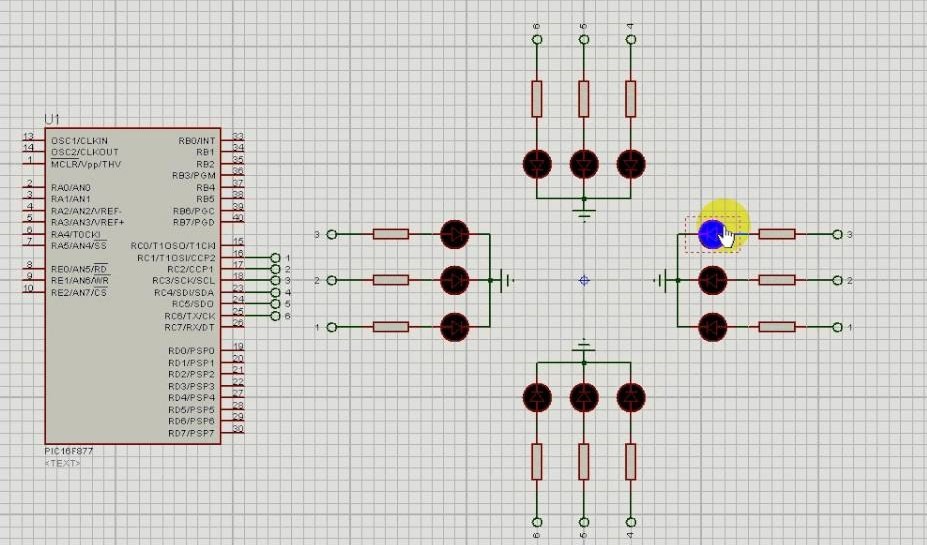
}

/\* USER CODE END 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.

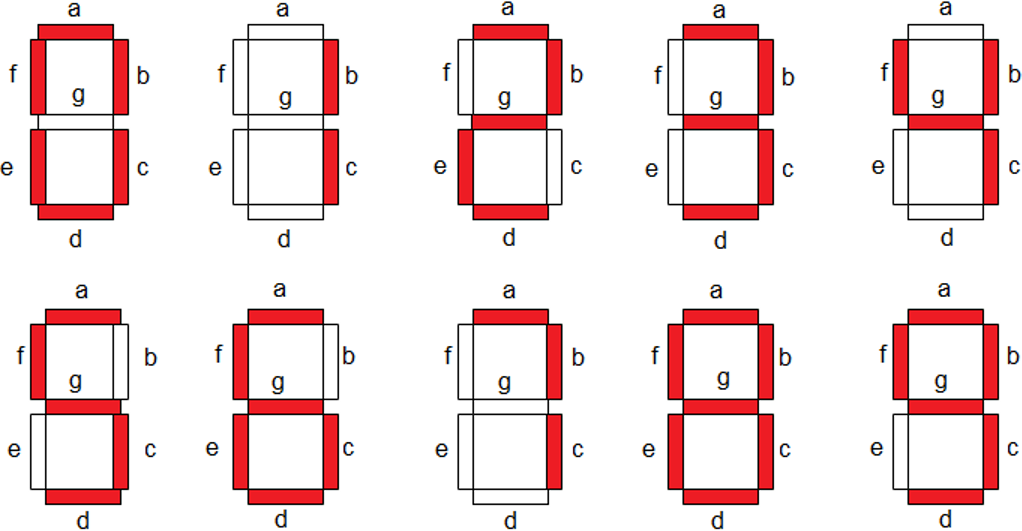


*Figure 1.24*: *Reference design for a 4 way traffic light*

## 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:



*Figure 1.25*: *Display a number on 7 segment LED*

This function is invoked in the while loop for testing as following:

1

int counter = 0;

while (1) {

if( counter >= 10) counter = 0; display 7 SEG ( counter ++) ; HAL\_Delay ( 1000) ;

}

2

3

4

5

6

7

Program 1.4: An example for your source code

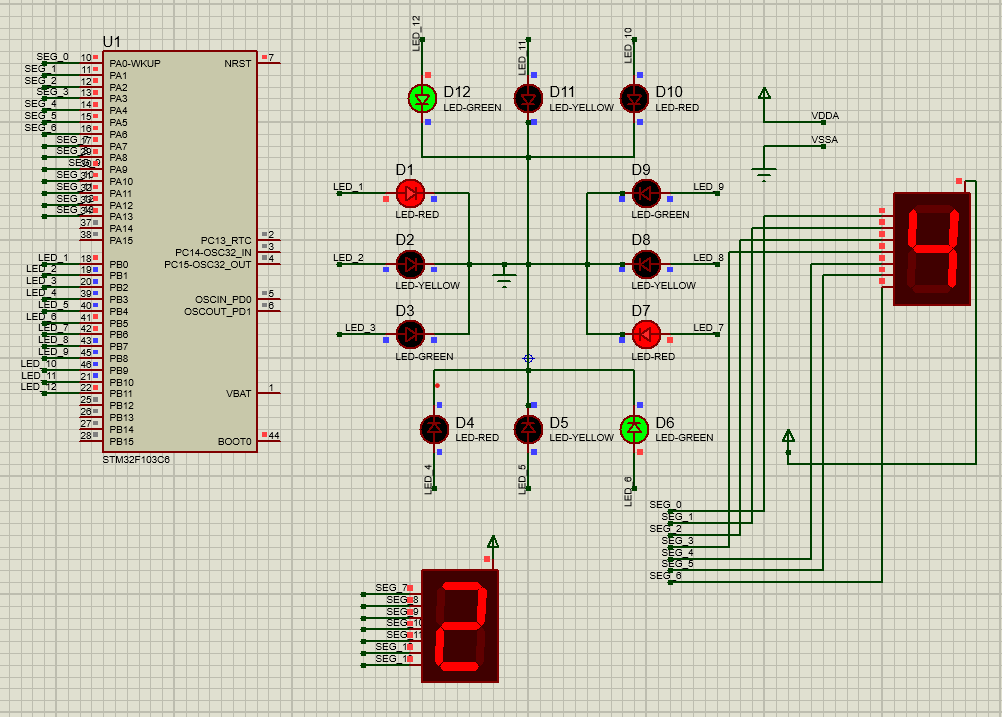
**Report 1:** Present the schematic.

**Report 2:** Present the source code for display7SEG function.

## Exercise 5

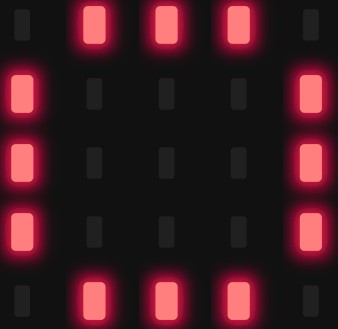
Integrate the 7SEG-LED to the 4 way traffic light. In this case, the 7SEG-LED is used to display countdown value.

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



## 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.



*Figure 1.26*: *12 LEDs for an analog clock*

Report 1: Present the schematic. Report 2: Implement a simple program to test the con- nection of every single LED. This testing program should turn every LED in a sequence.

## Exercise 7

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

1

void clear All Clock (){

// TODO

}

2

3

Program 1.5: Function Implementation

## Exercise 8

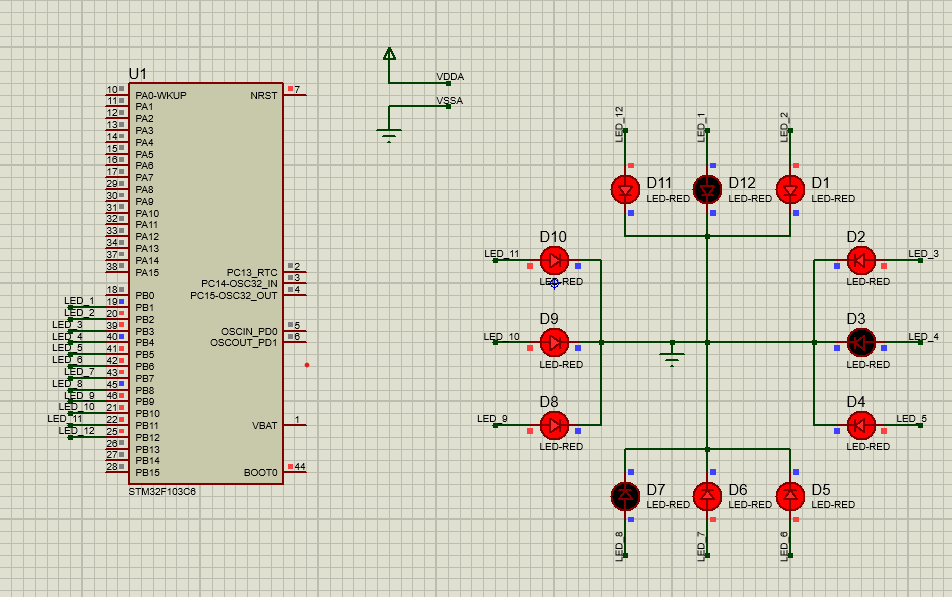
Implement a function named **setNumberOnClock(int num)**. The input for this function is from **0 to 11** and an appropriate LED is turn on. Present the source code of this function.

## Exercise 9

Implement a function named **clearNumberOnClock(int num)**. The input for this func- tion is from **0 to 11** and an appropriate LED is turn off.

## 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.



Link source code :

https://drive.google.com/drive/folders/15EuveO-GADOYun0ZEqkXsybbNwk5jx3E?usp=sharing