



HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller



Dr. Le Trong Nhan

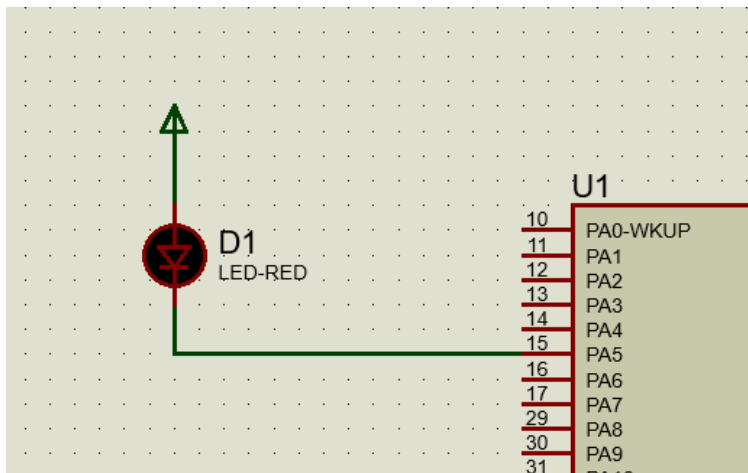
```
while(1)
{

// Các câu lệnh if

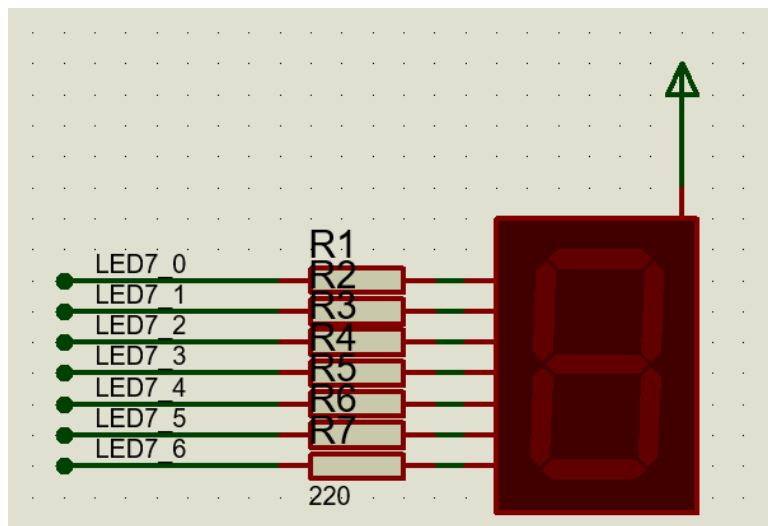
delay 10ms
}
```

Bước đầu tiên khi tiếp cận 1 hệ thống MCU là cấu hình thạch anh dao động

Nên đặt tên chân lại khi config file .ioc để dễ quản lý, dễ code hơn



cực dương LED nên lấy nguồn ngoài, cực âm mắc vào chân I/O của MCU
 => tránh lấy nguồn từ chân I/O của MCU cấp trực tiếp cho LED



Tương tự nên dùng LED7 chung anode

=> nên lắp mỗi led có 1 trở hạn dòng riêng để đảm bảo độ sáng các led đồng đều

Contents

Chapter 1. LED Animations	7
1 Introduction	8
2 First project on STM32Cube	8
3 Simulation on Proteus	9
4 Exercise and Report	11
4.1 Exercise 1	11
4.2 Exercise 2	12
4.3 Exercise 3	12
4.4 Exercise 4	12
4.5 Exercise 5	13
4.6 Exercise 6	13
4.7 Exercise 7	13
4.8 Exercise 8	14
4.9 Exercise 9	14
4.10 Exercise 10	14
Chapter 2. LED Scanning	15
Chapter 3. Buttons	17
Chapter 4. Smart Watch Project	19
Chapter 5. Internet of Things Project	21
1 Introduction	22
2 Reading sensor	22
3 Control actuators	22
4 Communications based UART	22

CHAPTER 1

LED Animations



1 Introduction

2 First project on STM32Cube

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**.

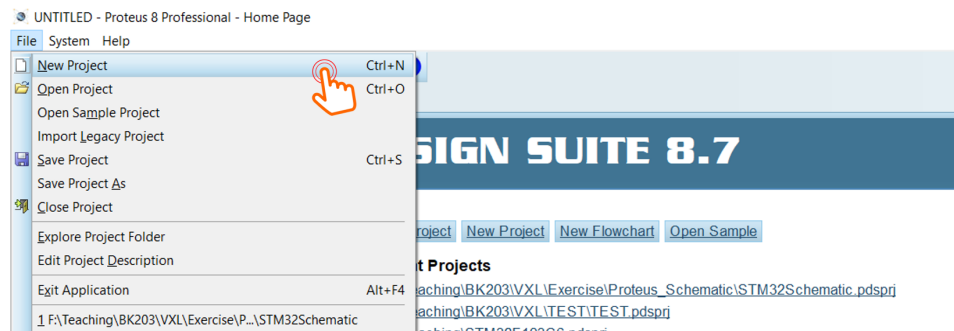


Figure 1.1: Create a new project on Proteus

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

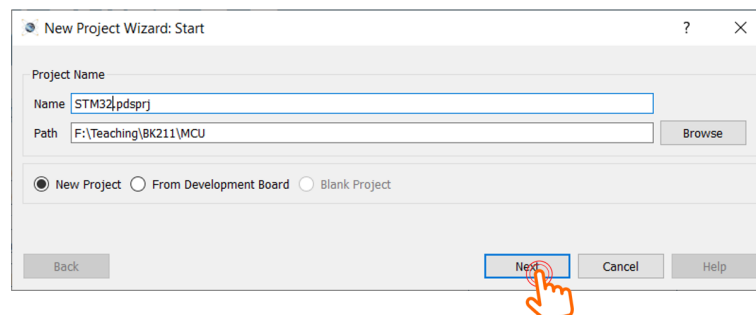


Figure 1.2: Provide project name and location

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

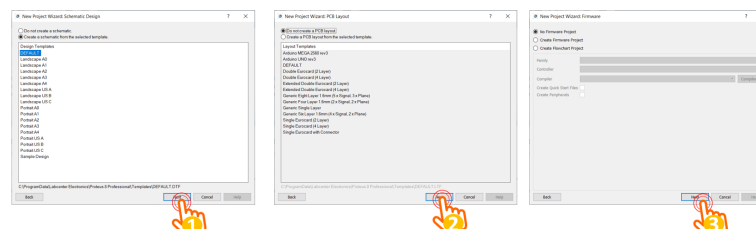


Figure 1.3: Keep the default options by clicking on Next

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

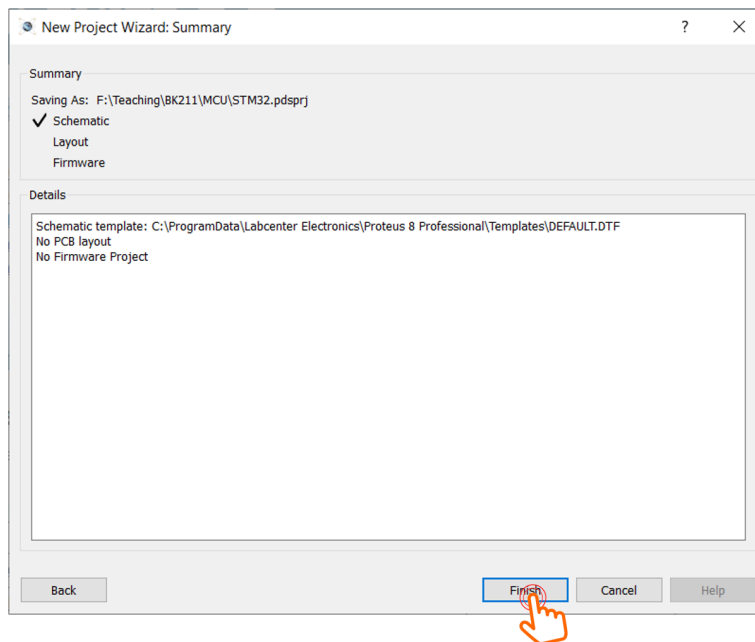


Figure 1.4: Finish the project wizard

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

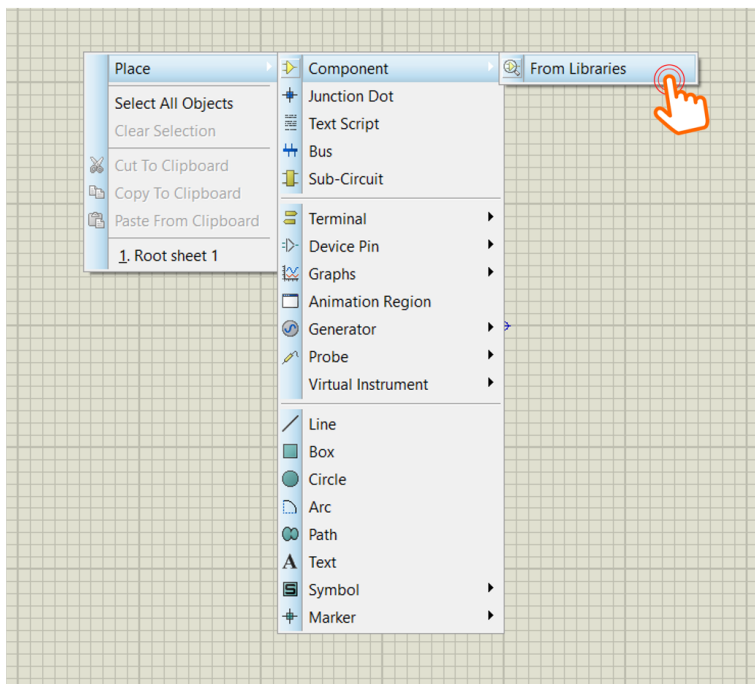


Figure 1.5: Select a component from the library

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:

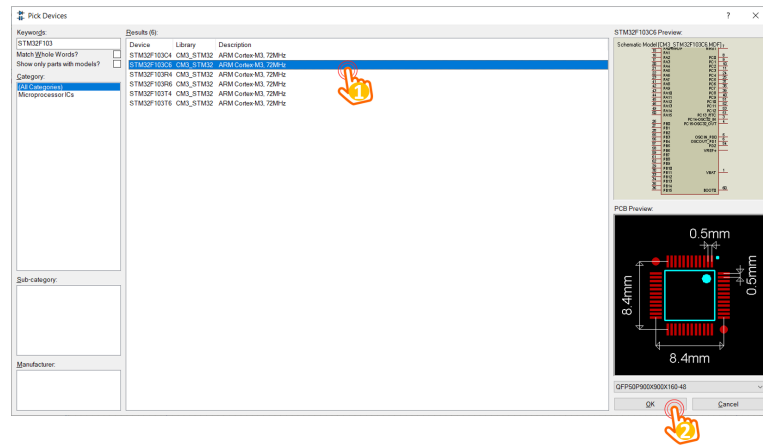


Figure 1.6: Select STM32F103C6

Repeat step 5 and 6 to select an LED, named **LED-RED** in Proteus.

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.

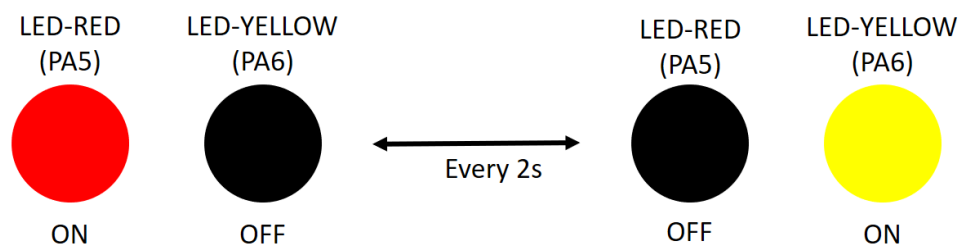


Figure 1.7: State transitions for 2 LEDs

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) {
2     HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
3     HAL_Delay(1000);
```

Program 1.1: An example for your source code

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.

Report 2: Present the source code in while.

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.

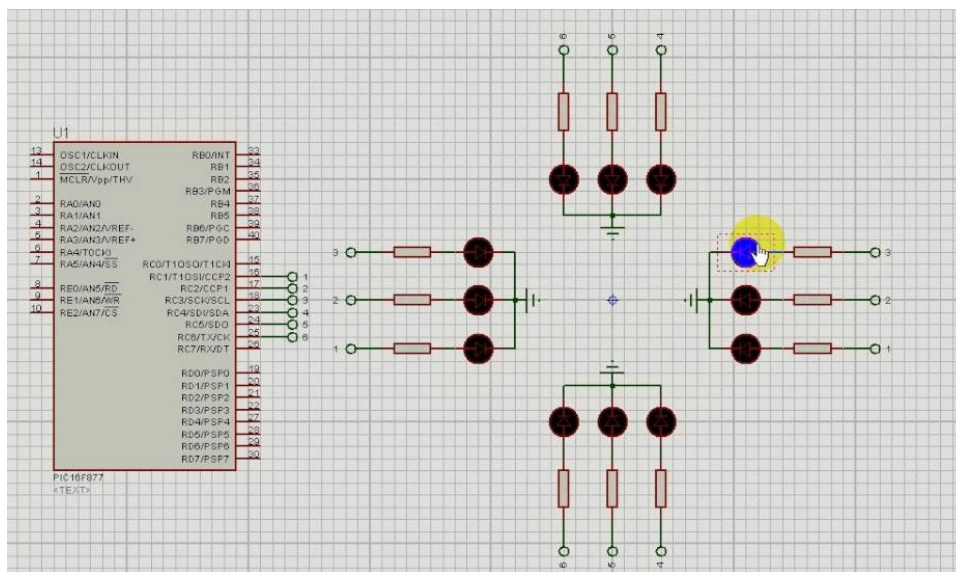


Figure 1.8: Reference design for a 4 way traffic light

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 be 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:

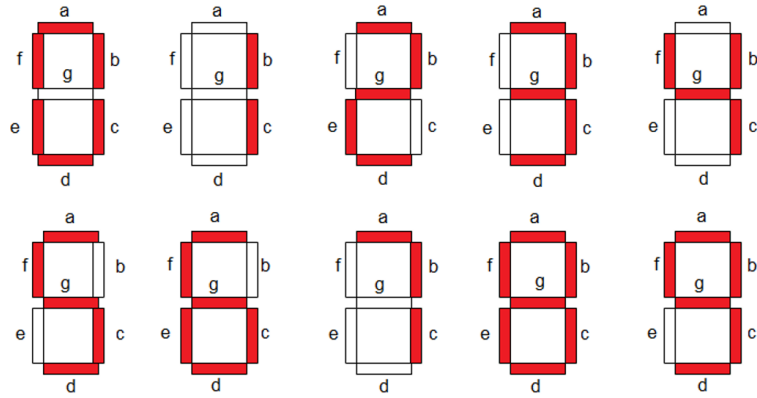


Figure 1.9: Display a number on 7 segment LED

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

```

1 int counter = 0;
2 while (1){
3     if(counter >= 10) counter = 0;
4     display7SEG(counter++);
5     HAL_Delay(1000);
6
7 }
```

Program 1.2: An example for your source code

Report 1: Present the schematic.

Report 2: Present the source code for display7SEG function.

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.

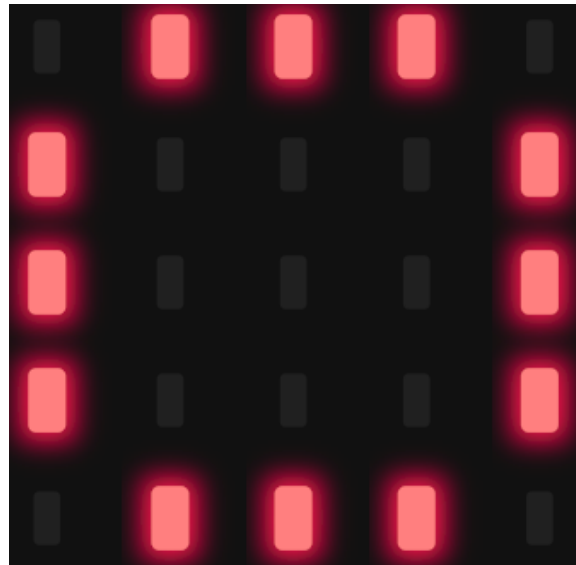
4.6 Exercise 6

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.

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

4.7 Exercise 7

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



kim giờ giữ nguyên
kim phút blink
kim giây thì nhảy ào ào rồi

Figure 1.10: 12 LEDs for an analog clock

```
1 void clearAllClock(){  
2     //TODO  
3 }
```

Program 1.3: 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.

4.9 Exercise 9

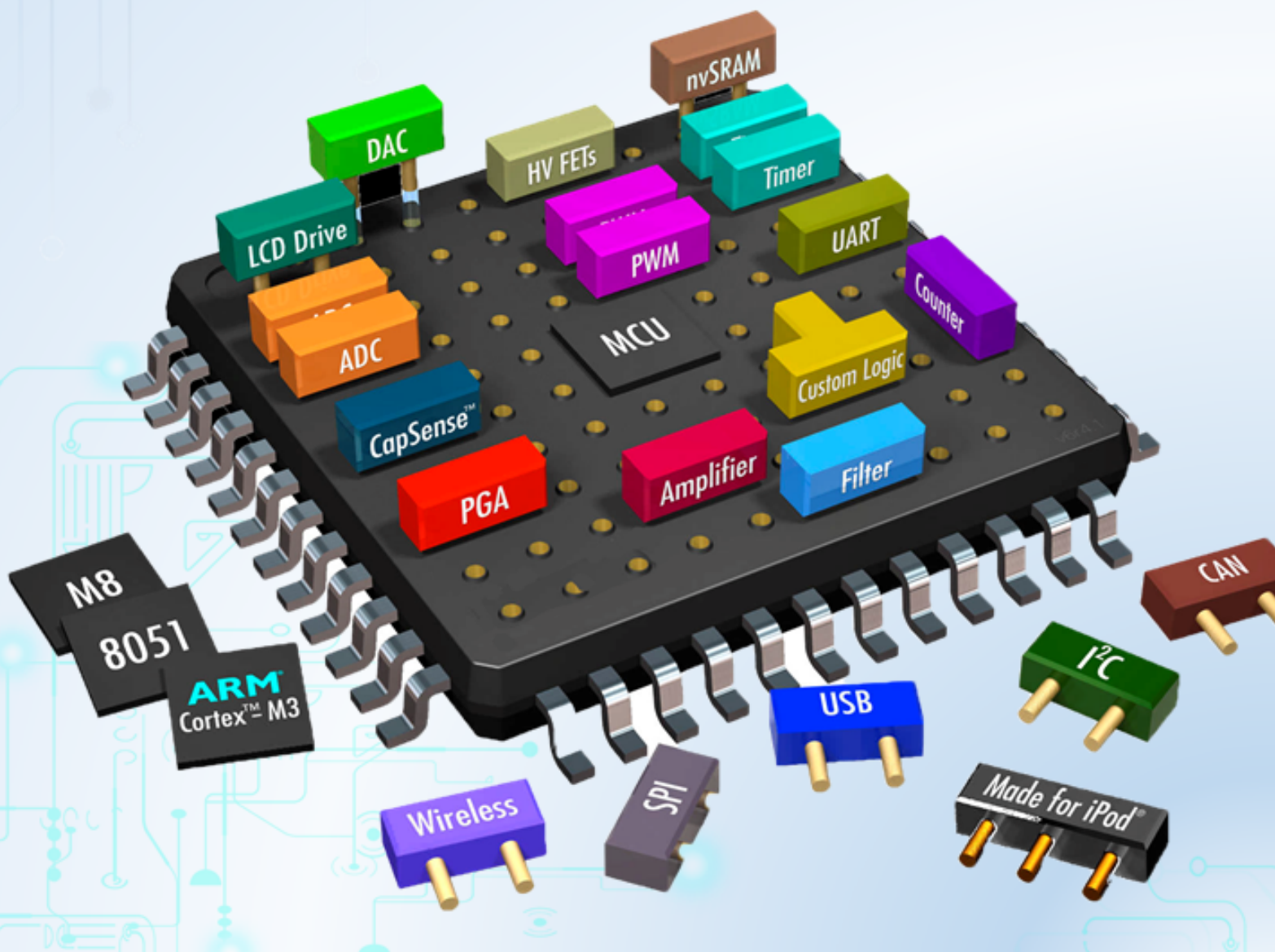
Implement a function named **clearNumberOnClock(int num)**. The input for this function is from **0 to 11** and an **appropriate LED is turn off**.

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.**

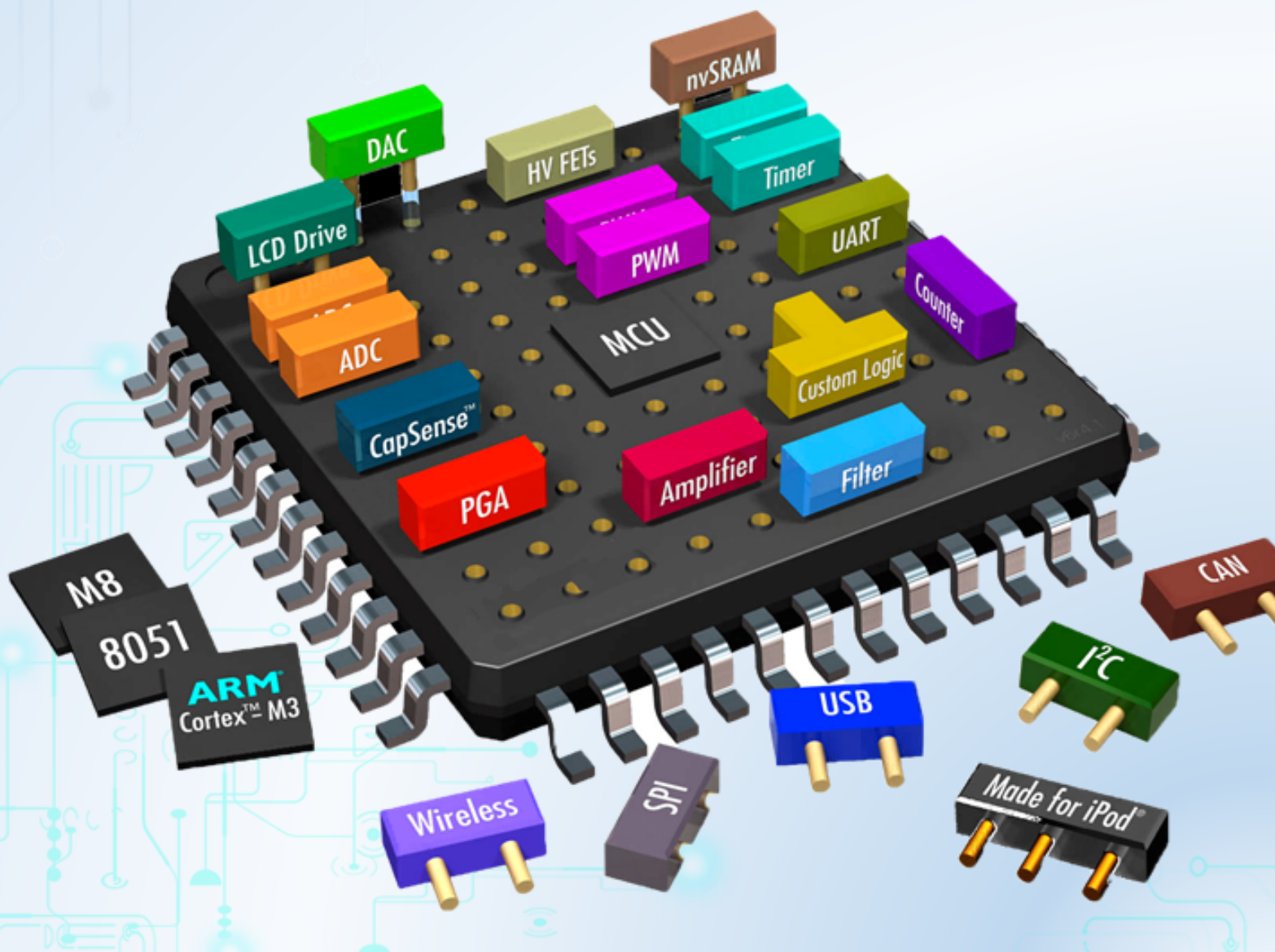
CHAPTER 2

LED Scanning



CHAPTER 3

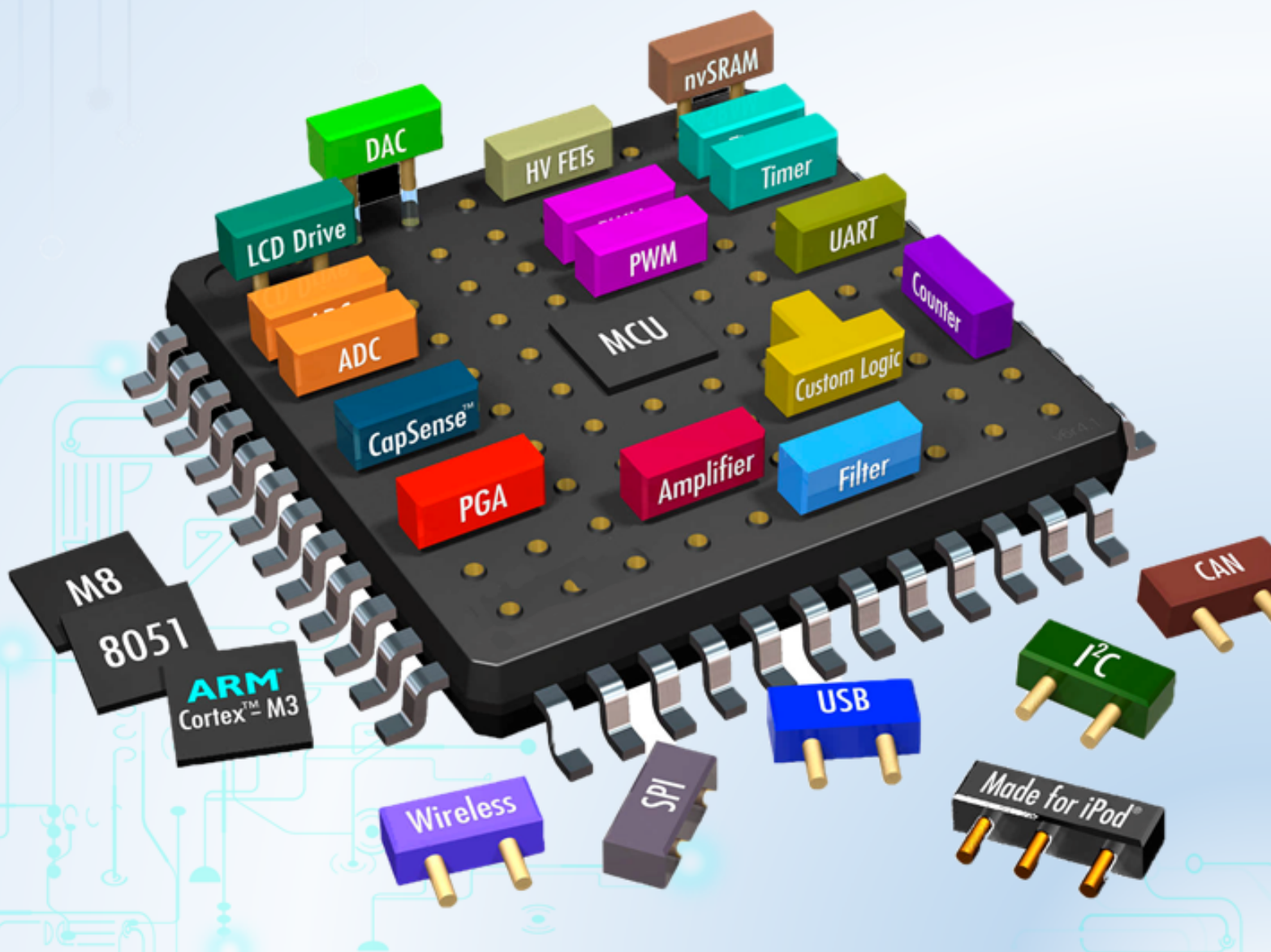
Buttons



CHAPTER 4

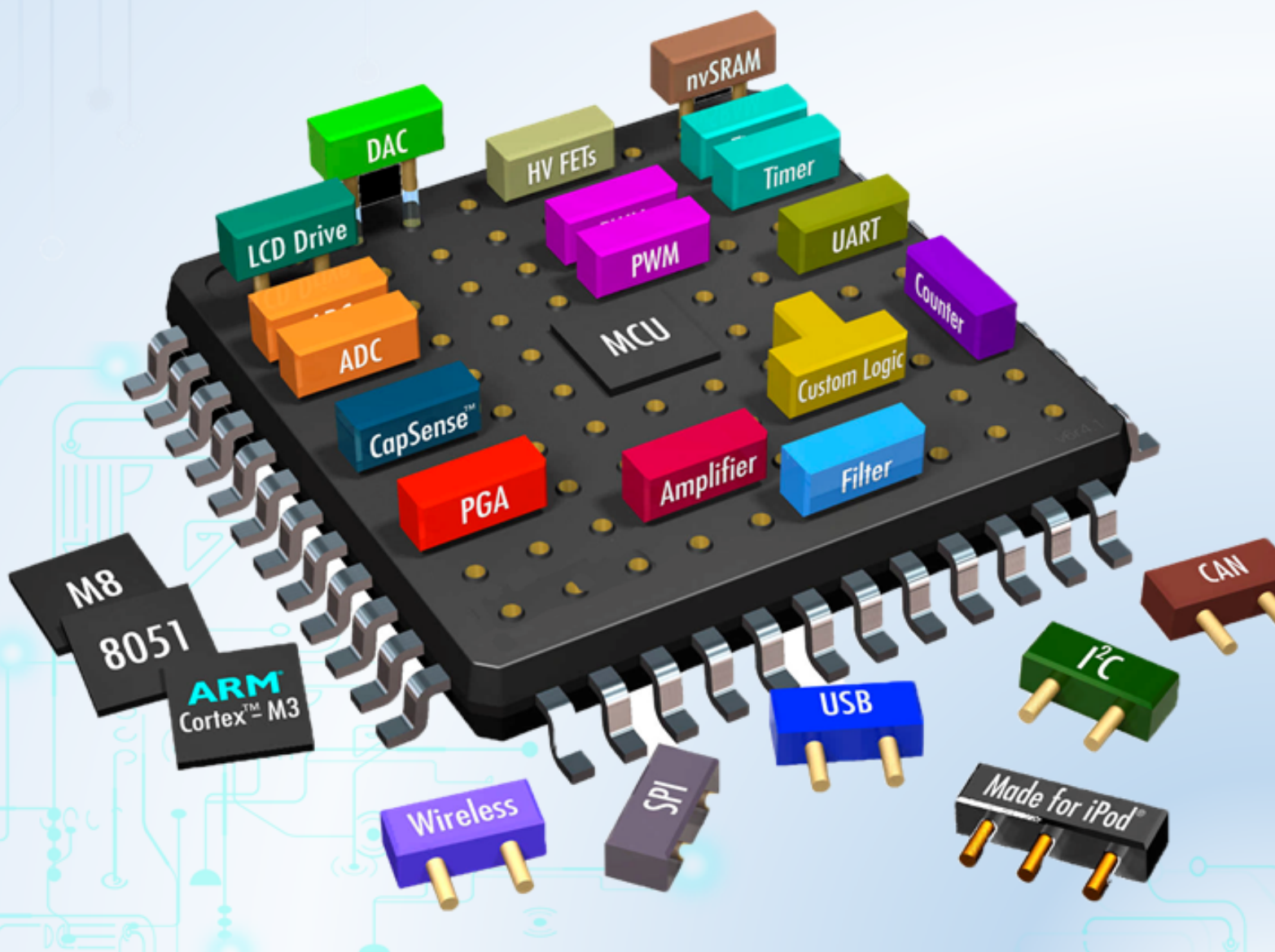
Smart Watch Project

làm 1 đồng hồ casio, có nút nhấn chỉnh giờ phút giây



CHAPTER 5

Internet of Things Project



1 Introduction

2 Reading sensor

analogRead

3 Control actuators

analogWrite PWM

4 Communications based UART