



HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller



Dr. Le Trong Nhan

Mục lục

Chapter 1. GIỮA KÌ 2022	7
1 Giới thiệu	8
2 Hiện thực và Report	9
2.1 Sơ đồ nguyên lý trên Proteus - 1 điểm	9
2.2 State machine Step 1 - 2 điểm	10
2.3 State machine Step 2 - 2 điểm	11
2.4 State machine Step 3 - 2 points	14
2.5 Led Blinky for Debugging - 1 điểm	15
2.6 Github và Demo	15
3 Bài tập thêm - Engineer mindset - 1 điểm	16

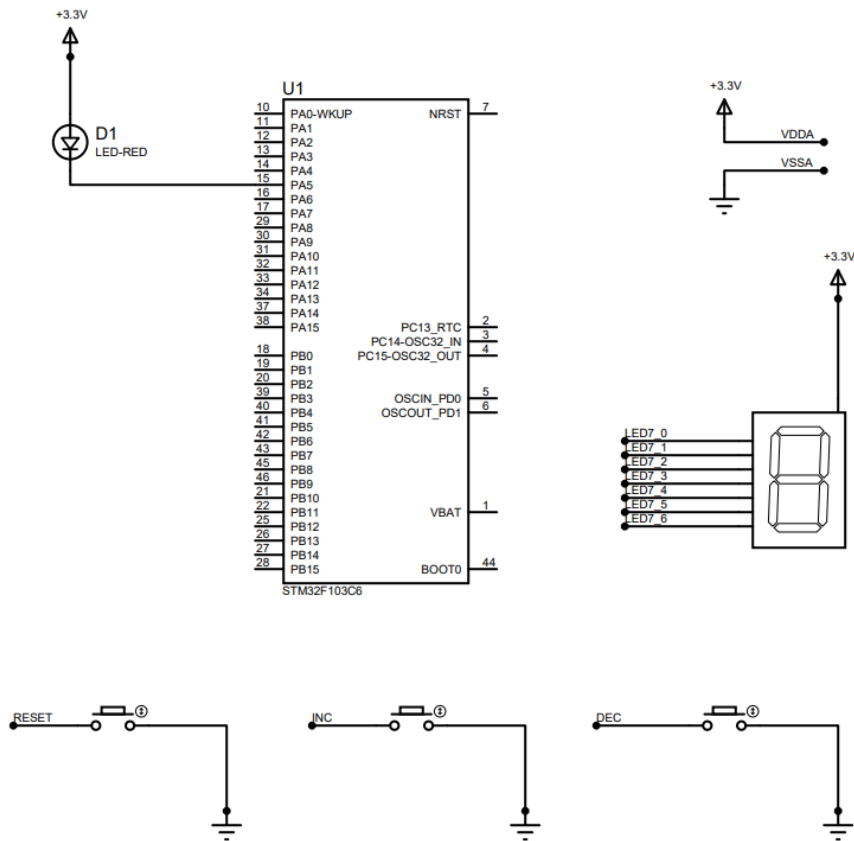
CHƯƠNG 1

GIỮA KÌ 2022



1 Giới thiệu

Trong project giữa kì này, một hệ thống đếm lùi sẽ được hiện thực trên phần mềm mô phỏng Proteus. Như được trình bày ở Hình 1.1, các thành phần chính của hệ thống bao gồm vi điều khiển STM32F103C6, một đèn LED, một LED 7 đoạn và 3 nút nhấn đơn.



Hình 1.1: Proteus schematic for count-down system

Một số tính năng chính của hệ thống được trình bày như sau:

- LED 7 đoạn dùng để hiển thị giá trị của counter, có giá trị từ 0 đến 9.
- Nút **RESET** được dùng để reset giá trị của counter về 0. Trong khi đó, nút nhấn **INC** và **DEC** được dùng để tăng hoặc giảm giá trị của counter. Có 2 sự kiện cần phải xử lý cho các nút nhấn, là nhấn thường và nhấn giữ. Trong dự án này, một nút nhấn được coi là nhấn giữ, nếu nó giữ nguyên trạng thái đó trong 3 giây liên tiếp.
- Đèn LED D1 được dùng để theo dõi hoạt động của hệ thống, nó sẽ luân phiên chớp tắt mỗi giây.

Sinh viên sẽ hiện thực dự án của mình theo từng bước yêu cầu ở phần bên dưới. Trong mỗi phần, sinh viên cần trình bày các yêu cầu về report. Một số lưu ý quan trọng cho phần hiện thực như sau:

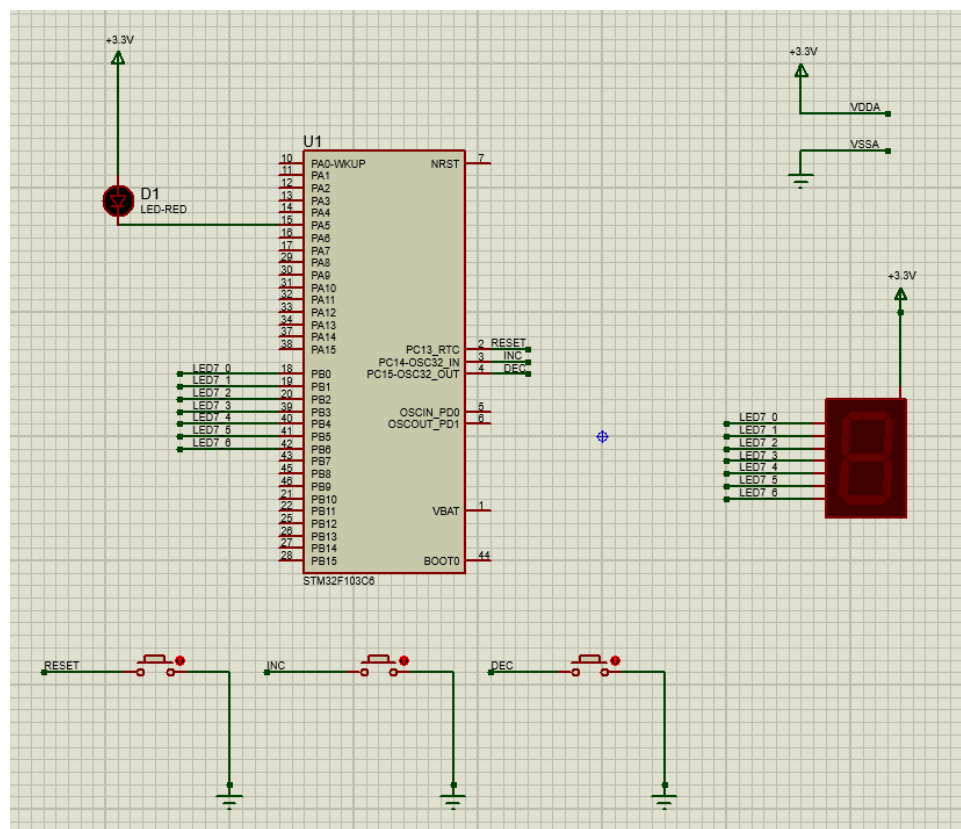
- Chu kỳ của ngắt timer là 10ms. Giá trị của counter khi cài đặt timer có thể là 9 hoặc 10 đều được chấp nhận (trong trường hợp này, prescaler là 7999).
- Tất cả các nút nhấn đều được xử lý chống rung bằng ngắt timer 10ms. Một nút nhấn sẽ được xem là nhấn đè nếu nó được giữ liên tục trong 3 giây.
- Không sử dụng HAL_Delay() trong việc hiện thực. Tất cả các hiệu ứng thời gian đều phải được hiện thực dựa trên software timer.
- Report này cần được nộp lại kèm theo các câu trả lời của sinh viên.
- Link github và video demo trong report này được chỉnh quyền truy cập public.

2 Hiện thực và Report

2.1 Sơ đồ nguyên lý trên Proteus - 1 điểm

Trong phần này, sinh viên đề xuất các kết nối của LED 7 đoạn và 3 nút nhấn với STM32F103C6.

Report: Trình bày sơ đồ nguyên lý tại đây. Sinh viên có thể chụp màn hình Proteus cho phần sơ đồ nguyên lý.



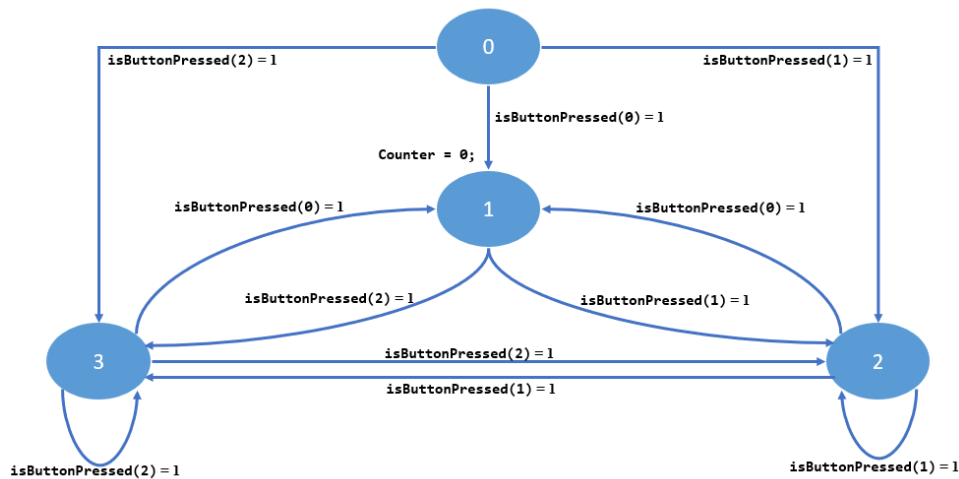
2.2 State machine Step 1 - 2 điểm

Một máy trạng thái sẽ được thiết kế để thực hiện các chức năng với trạng thái nhấn bình thường (normal-press) cho 3 nút nhấn, như sau:

- Khi nút RESET is được nhấn, giá trị counter sẽ là 0.
- Khi INC được nhấn, giá trị của counter tăng 1 đơn vị. Khi nó đến 9, giá trị tiếp theo là 0.
- Khi DC được nhấn, giá trị của counter giảm 1 đơn vị. Khi giá trị của nó là 0, giá trị tiếp theo là 9.

Giá trị của biến counter được hiển thị lên LED 7 đoạn.

Report: Trình bày máy trạng thái của hệ thống.



Trạng thái 0: chưa có nút nào được nhấn, counter có giá trị xác định.

Trạng thái 1: nếu nút RESET được nhấn, chuyển sang trạng thái 1, giá trị counter bằng 0.

Trạng thái 2: nếu nút INC được nhấn, chuyển sang trạng thái 2, giá trị counter bằng giá trị counter trước đó cộng thêm 1. Nếu giá trị counter bằng 9, thì giá trị counter tiếp theo bằng 0.

Trạng thái 3: nếu nút DEC được nhấn, chuyển sang trạng thái 3, giá trị counter bằng giá trị counter trước đó trừ đi 1. Nếu giá trị counter bằng 0, thì giá trị counter tiếp theo bằng 9.

Your report: Trình bày hiện thực của hàm dùng để hiển thực máy trạng thái ở trên. Thông thường, hàm này sẽ được gọi trong main().

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

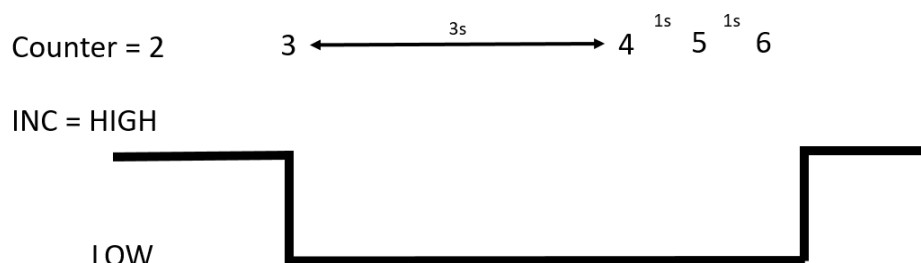
Program 1.1: Hiện thực máy trạng thái

```
1 void fsm_simple_buttons_run() {  
2     if(isButtonPressed(0)==1){  
3         counter = 0;  
4     }  
5     if(isButtonPressed(1)==1){  
6         counter++;  
7         if(counter>9) counter=0;  
8     }  
9     if(isButtonPressed(2)==1){  
10        counter--;  
11        if(counter<0) counter=9;  
12    }  
13    display7SEG(counter);  
14 }
```

2.3 State machine Step 2 - 2 điểm

Trong phần này, việc nhấn giữ (long-press) cho nút INC và DEC được thêm vào trong dự án. Đối với nút nhấn, sự kiện long-press xảy ra khi nó được nhấn giữ liên tục sau 3 giây.

Khi một sự kiện nhấn giữ được phát hiện, giá trị của counter liên tục thay đổi mỗi 1 giây. Ví dụ, giá trị hiện tại của counter là 2. Khi nút INC được nhấn, ngay lập tức giá trị của nó sẽ tăng lên 3. Tuy nhiên, khi INC tiếp tục được giữ, 3 giây sau, sự kiện long-press xảy ra, giá trị của counter tăng lên 4. Từ lúc này, giá trị của counter sẽ tăng lên 1 mỗi giây, cho đến khi nút INC được thả ra. Giá trị của biến counter này được minh họa như hình bên dưới.

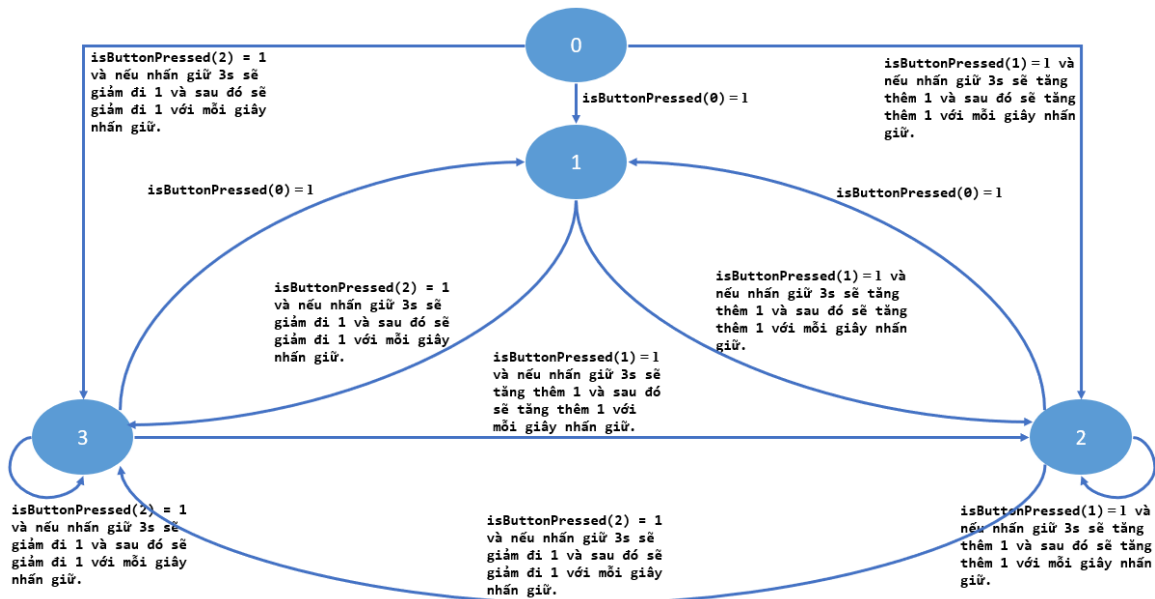


Hình 1.2: Long press behavior for INC button

Hành vi của nút DEC sẽ ngược lại với nút INC. Giá trị của biến counter sẽ xoay

vòng khi nó đến 0 hoặc 9.

Report: Trình bày toàn bộ máy trạng thái cho đến bước này.



Trạng thái 0: chưa có nút nào được nhấn, counter có giá trị xác định.

Trạng thái 1: nếu nút RESET được nhấn, chuyển sang trạng thái 1, giá trị counter bằng 0.

Trạng thái 2: nếu nút INC được nhấn, chuyển sang trạng thái 2, giá trị counter bằng giá trị counter trước đó cộng thêm 1. Khi nút được nhấn thì counter tăng lên 1 và nếu nhấn giữ sau 3s, giá trị counter sẽ tăng thêm 1 đơn vị, sau đó giá trị counter sẽ tăng thêm 1 với mỗi giây nhấn giữ thêm. Nếu giá trị counter bằng 9, thì giá trị counter tiếp theo bằng 0.

Trạng thái 3: nếu nút DEC được nhấn, chuyển sang trạng thái 3, giá trị counter bằng giá trị counter trước đó trừ đi 1. Khi nút được nhấn thì counter giảm đi 1 và nếu nhấn giữ sau 3s, giá trị counter sẽ giảm đi 1 đơn vị, sau đó giá trị counter sẽ giảm đi 1 với mỗi giây nhấn giữ thêm. Nếu giá trị counter bằng 0, thì giá trị counter tiếp theo bằng 9.

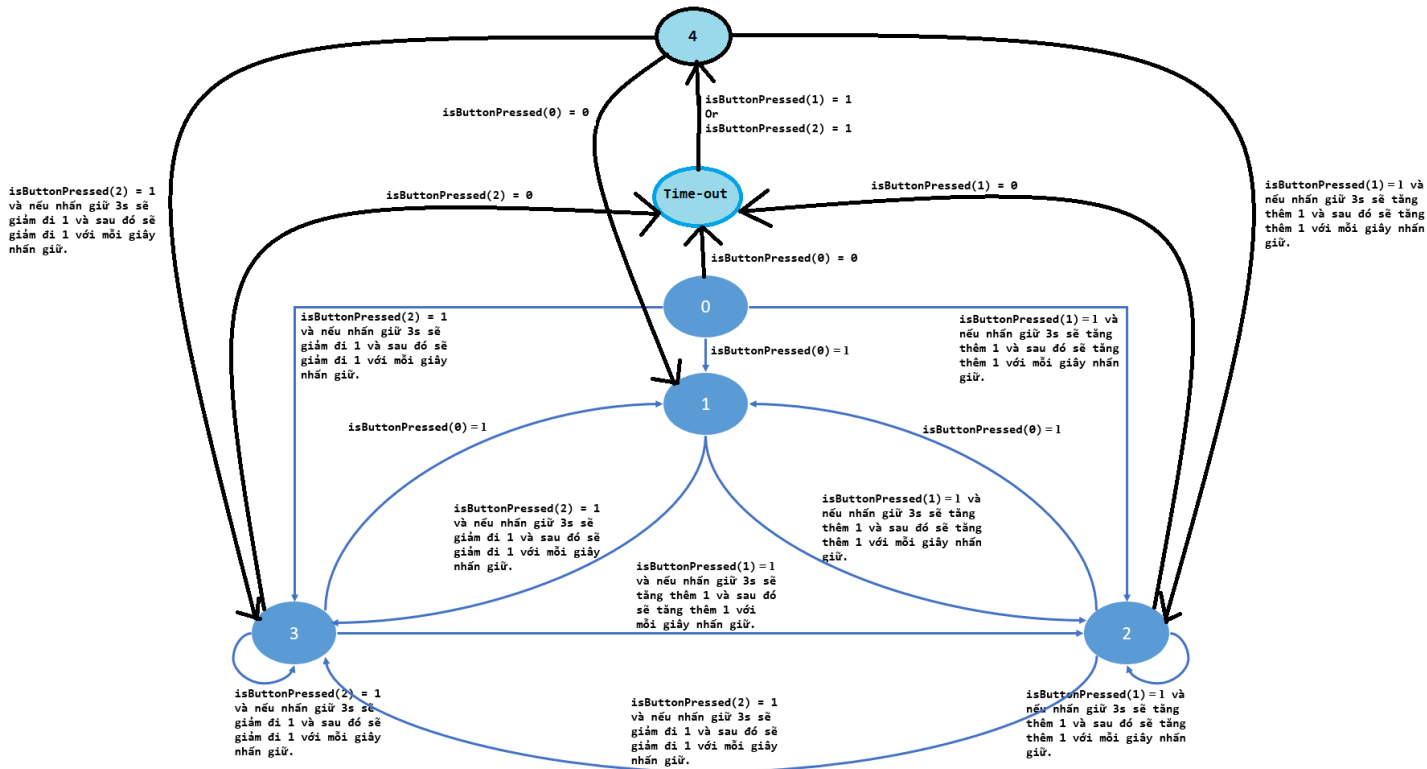
Report: Sinh viên trình bày một hàm chính dùng để hiện thực các trạng thái mới. Các thay đổi trên máy trạng thái cũ không cần phải trình bày ở đây.

```
1 // em không thêm một hàm mới mà em thêm một vài dòng code
   vào file button.c để thêm chức năng nhấn để luôn.
2 // ở đây em tạo thêm mảng thời gian nhấn để.
3 int TimerForKeyPress[3];
4 void getKeyInput(){
5     for(int i=0;i<3;i++){
6         keyReg0[i] = keyReg1[i];
7         keyReg1[i] = keyReg2[i];
8         if(i==0){
9             keyReg2[i] = HAL_GPIO_ReadPin(
10 Button_RESET_GPIO_Port, Button_RESET_Pin);
11         }
12         if(i==1){
13             keyReg2[i] = HAL_GPIO_ReadPin(Button_INC_GPIO_Port,
14 Button_INC_Pin);
15         }
16         if(i==2){
17             keyReg2[i] = HAL_GPIO_ReadPin(Button_DEC_GPIO_Port,
18 Button_DEC_Pin);
19         }
20         if((keyReg0[i]==keyReg1[i])&&(keyReg1[i]==keyReg2[i])){
21             if(keyReg3[i] != keyReg2[i]){
22                 keyReg3[i] = keyReg2[i];
23                 if(keyReg2[i]==PRESSED_STATE){
24                     subKeyProcess(i);
25 //nếu trạng thái trước khác với trạng thái nhấn để thì em
26 //se reset thời gian nhấn để là 3s.
27                 TimerForKeyPress[i] = 300;
28             }
29         }
30         else{
31             TimerForKeyPress[i]--;
32             if(TimerForKeyPress[i]<0){
33                 if(keyReg2[i]==PRESSED_STATE){
34                     subKeyProcess(i);
35 //nếu như trạng thái trước giống trạng thái nhấn để thì sau
36 //3s counter sẽ tăng lên 1, nếu nhấn lâu hơn 3s thì cứ
37 //sau 1s giá trị counter sẽ tăng thêm 1.
38                 TimerForKeyPress[i] = 100;
39             }
40         }
41     }
42 }
43 }
44 }
```

2.4 State machine Step 3 - 2 points

Cuối cùng, khi không có nút nào được nhấn, hệ thống tự động đếm lùi biến counter và dừng lại khi nó bằng 0. Sau đó, nếu nút INC hoặc DEC được nhấn, trạng thái của hệ thống lại quay về một trong các trạng thái đã thiết kế trước đó.

Your report: Trình bày toàn bộ máy trạng thái, khi sự kiện time-out 10s được thêm vào.



Trạng thái 0: chưa có nút nào được nhấn, counter có giá trị xác định.

Trạng thái 1: nếu nút RESET được nhấn, chuyển sang trạng thái 1, giá trị counter bằng 0.

Trạng thái 2: nếu nút INC được nhấn, chuyển sang trạng thái 2, giá trị counter bằng giá trị counter trước đó cộng thêm 1. Khi nút được nhấn thì counter tăng lên 1 và nếu nhấn giữ sau 3s, giá trị counter sẽ tăng thêm 1 đơn vị, sau đó giá trị counter sẽ tăng thêm 1 với mỗi giây nhấn giữ thêm. Nếu giá trị counter bằng 9, thì giá trị counter tiếp theo bằng 0.

Trạng thái 3: nếu nút DEC được nhấn, chuyển sang trạng thái 3, giá trị counter bằng giá trị counter trước đó trừ đi 1. Khi nút được nhấn thì counter giảm đi 1 và nếu nhấn giữ sau 3s, giá trị counter sẽ giảm đi 1 đơn vị, sau đó giá trị counter sẽ giảm đi 1 với mỗi giây nhấn giữ thêm. Nếu giá trị counter bằng 0, thì giá trị counter tiếp theo bằng 9.

Trạng thái Time-out: nếu không có nút nhấn nào được nhấn trong 10s thì counter sẽ tự đếm xuống và dừng lại khi nó bằng 0.

Trạng thái 4: nếu trong khoảng thời gian đếm xuống 0, khi nhấn INC hoặc DEC, giá trị counter sẽ quay trở lại giá trị counter trước đó.

Report: Sinh viên trình bày một hàm chính dùng để hiện thực các trạng thái mới. Các thay đổi trên máy trạng thái cũ không cần phải trình bày ở đây.

2.5 Led Blinky for Debugging - 1 điểm

Cuối cùng, trong nhiều dự án dựa trên vi điều khiển, có 1 LED luôn luôn chớp tắt mỗi giây, để giám sát hoạt động của hệ thống. Trong project này, LED nối với chân PA5 sẽ được dùng để hiện thực tính năng này.

Report: Sinh viên trình bày giải pháp của mình cho tính năng này. Giải pháp có thể rất đơn giản với vài dòng code hoặc thiết kế máy trạng thái và lập trình cho nó. Trong trường hợp thứ 2, sinh viên trình bày máy trạng thái trước khi trình bày phần hiện thực mã nguồn.

```
1 setTimer1(100);
2 while(1){
3     if(isButtonPressed(0)==1 || isButtonPressed(1)==1 ||
4        isButtonPressed(2)==1){
5         HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, 0);
6         if(timer1_flag==1){
7             HAL_GPIO_TogglePin(LED_RED_GPIO_Port, LED_RED_Pin);
8             setTimer1(100);}
9         }
10        else{
11            if(timer1_flag==1){
12                HAL_GPIO_TogglePin(LED_RED_GPIO_Port, LED_RED_Pin);
13                setTimer1(100);
14            }
15        }
```

2.6 Github và Demo

Link github cho dự án của sinh viên được trình bày ở đây, bao gồm tất cả các files trong dự án (configurations, header and source files). Link này là lần commit sau cùng trong project giữa kì của sinh viên.

[Link github](#)

Link cho 1 video demo cũng sẽ được trình bày ở phần này.

[Link video mô phỏng](#)

3 Bài tập thêm - Engineer mindset - 1 điểm

In this course, we encourage you to obtain an innovative mindset to solve daily problem. In this question, we would expect you to write a C program to solve the following problem.

Suffix with Unit

EXample:

1 suffixWithUnit(123) => 123

2 suffixWithUnit(1234) => 1.234 Kilo

3 suffixWithUnit(12345) => 12.345 Kilo

4 suffixWithUnit(1234567) => 1.234567 Mega

5 suffixWithUnit(12345678) => 12.345678 Mega

Prototype

```
1 string suffixWithUnit(double number) {  
2 }
```

How would you solve them? Please share your thinking to solve this problem and provide your answer.