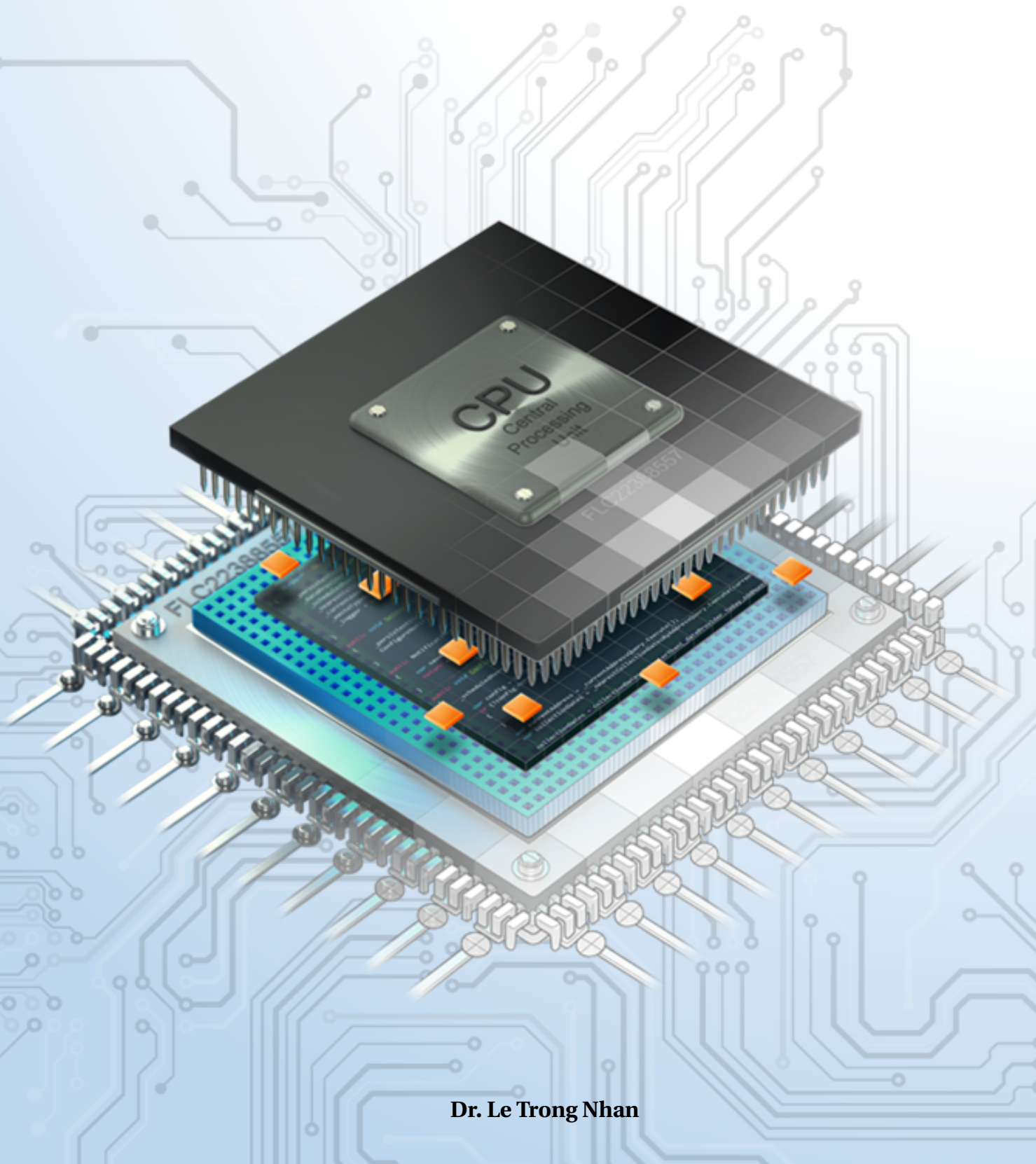




HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY  
COMPUTER ENGINEERING

# Microcontroller



Dr. Le Trong Nhan







---

# Mục lục

---

<b>Chapter 1. Bài Tập Giữa Kỳ</b>	<b>7</b>
1 Giới thiệu . . . . .	8
2 Nộp bài . . . . .	8
3 Report . . . . .	9
3.1 Mô phỏng trên Proteus . . . . .	9
3.2 Thiết kế máy trạng thái . . . . .	9
3.3 Lập trình trên STM32Cube . . . . .	10
3.4 Module Timer . . . . .	10
3.4.1 File timer.h . . . . .	10
3.4.2 File timer.c . . . . .	11
3.5 Module buttonScan . . . . .	11
3.5.1 File buttonScan.h . . . . .	11
3.5.2 File buttonScan.c . . . . .	12
3.6 Module clockFSM . . . . .	14
3.6.1 File clock_12LEDs.h . . . . .	14
3.6.2 File clock_12LEDs.c . . . . .	14
4 Video demo . . . . .	16



# CHƯƠNG 1

---

## Bài Tập Giữa Kỳ

---



# 1 Giới thiệu

Trong yêu cầu của bài giữa kì, một đồng hồ analog với 12 bóng đèn hiển thị, trưng trưng cho 12 số trên đồng hồ. Bên cạnh đó, sẽ có 3 nút nhấn: nút MENU, INC và DEC (để tăng và giảm thông tin)

Để tiện lợi trong quá trình demo, đồng hồ sẽ có 12 giờ. Tuy nhiên, mỗi phút sẽ chỉ có 12 giây, và mỗi giờ cũng sẽ có 12 phút.

Đồng hồ có 3 chế độ hoạt động như sau:

- Mode 0: Đây là chế độ mặt định mỗi khi bật nguồn hoặc khởi động lại hệ thống. Thông tin giờ phút và giây sẽ được hiển thị trên màn hình. Tại một thời điểm, chỉ có tối đa 3 đèn được hiển thị. Nếu 2 thông tin (hoặc 3 thông tin) trùng nhau, thì chỉ 1 đèn sẽ được hiển thị. Khi đang ở chế độ này, thông tin về giờ phút giây sẽ được cập nhật theo đúng quy luật của đồng hồ.
- Mode 1: Khi đang ở Mode 0 và nhấn vào nút MENU, đồng hồ sẽ chuyển sang chế độ này để chỉnh giờ. **Thông tin về giờ phút giây sẽ ngừng cập nhật để người dùng chỉnh giờ.** Chỉ một thông tin về giờ hiện tại sẽ được hiển thị. Khi nhấn nút INC và DEC, thông tin giờ sẽ được cộng thêm, hoặc trừ đi. Đồng thời, đèn hiển thị cũng sẽ được cập nhật trên mặt của đồng hồ. Lưu ý rằng, khi đang ở vị trí số 1, và nhấn nút trừ (DEC), thì thông tin giờ sẽ đếm vòng sang 12. Mỗi lần nhấn nút INC hoặc DEC, thông tin mới về giờ sẽ được lưu lại ngay lập tức.
- Mode 2: Khi đang ở Mode 1 và nhấn vào nút MENU, đồng hồ sẽ chuyển sang chế độ chỉnh phút. Chỉ một thông tin về phút hiện tại sẽ được hiển thị trên màn hình. Việc chỉnh phút cũng được thực hiện tương tự như chỉnh giờ.

**Khi đang ở Mode 1 hoặc Mode 2, người dùng không tương tác vào bất cứ nút nào (MENU, INC hay DEC) trong vòng 5 giây, hệ thống tự động chuyển về Mode 0.**

Hệ thống sẽ được hiện thực trên phần mềm mô phỏng Proteus. Vi điều khiển STM32103C6 sẽ được sử dụng. Sinh viên không bắt buộc phải sử dụng ngắt timer. Nút nhấn chỉ tích cực mỗi khi nhấn xuống và **không kiểm tra trường hợp nhấn đè**. Ví dụ đang ở trạng thái chỉnh giờ, nút INC được nhấn xuống, giờ ngay lập tức sẽ được tăng lên 1 đơn vị. Tuy nhiên nếu cứ giữ đè, thì giờ vẫn không tăng. Nút nhấn được chống rung bằng cách đọc 2 lần liên tiếp giống nhau, mỗi lần cách nhau 10ms.

## 2 Nộp bài

Sinh viên hoàn thiện file report này, với các nội dung được yêu cầu bên dưới, file main.c, tất cả các file .h và .c sinh viên hiện thực thêm sẽ được nén lại với MSSV và nộp lên hệ thống.

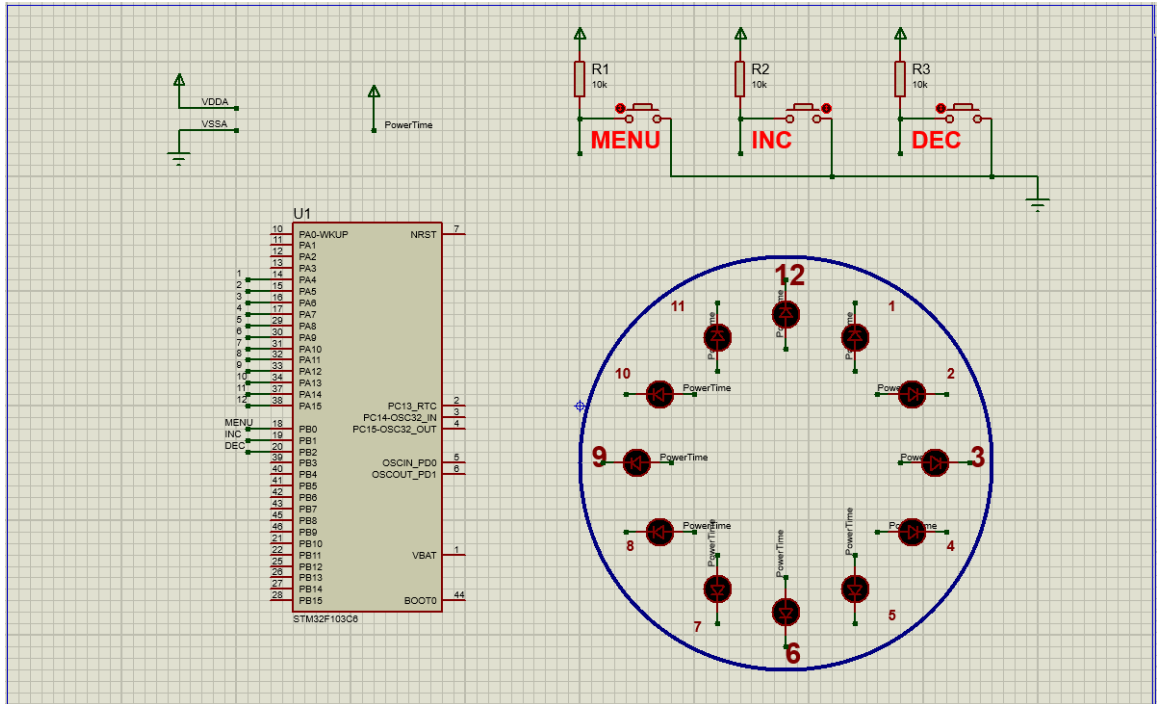
Sinh viên quay màn hình phần demo trên Proteus và tải lên Drive của mình (ở chế độ public). Link của video demo sẽ được trình bày ở phần cuối của Report.



## 3 Report

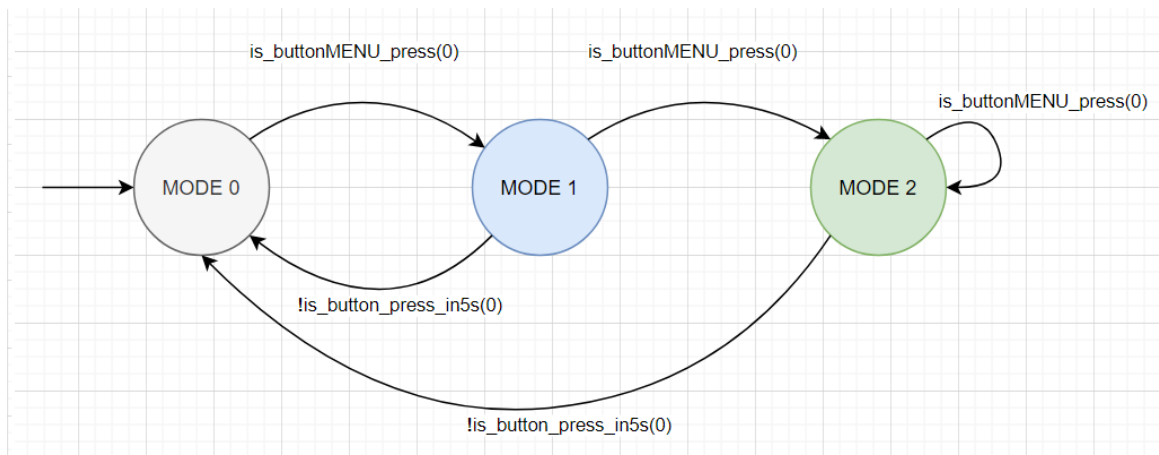
### 3.1 Mô phỏng trên Proteus

Thiết kế sơ đồ mạch điện trên Proteus, bao gồm 12 chân đèn LED và 3 nút nhấn. Để đơn giản, sinh viên có thể lược bỏ điện trở hạn dòng cho đèn LED.



Hình 1.1: Mô phỏng Proteus.

### 3.2 Thiết kế máy trạng thái



Hình 1.2: Máy trạng thái.

Thiết lập cờ **Flag\_5s=0** và biến đếm **counter\_Flag5s=0**.

- MODE 0: Trạng thái mặc định.

- MODE 1: Khi bấm nút MENU thì chuyển sang trạng thái MODE 2.

Cho biến **counter\_Flag5s** bắt đầu đếm giờ. Mỗi khi nhấn nút bất kỳ thì reset biến **counter\_Flag5s** về 0. Khi **counter\_Flag5s**  $\geq$  5s thì bật biến cờ **Flag\_5s=1**. Lúc này hệ thống sẽ chuyển lại MODE 0.

- MODE 2: Giống như MODE 1.

### 3.3 Lập trình trên STM32Cube

Bài làm KHÔNG sử dụng ngắt timer.

```
1 int main(void)
2 {
3     HAL_Init();
4     SystemClock_Config();
5
6     MX_GPIO_Init();
7     MODE=0;
8     setTimer0(1000); //delay 1s for power on clock
9     while (1){
10         clock_fsm();
11         if (MODE==0) timer_run();
12         else timer_run_Reset();
13         button_scan();
14         HAL_Delay(10);
15     }
16 }
```

Program 1.1: Cấu trúc chương trình trên main

### 3.4 Module Timer

Đặc tả ngắn gọn về module này

#### 3.4.1 File timer.h

```
1 # ifndef _TIMER_H_
2 # define _TIMER_H_
3
4 extern int timer0_flag ;
5 extern int MODE;
6
7 void setTimer0 (int duration );
8 void timer_run ();
9 void setTimerReset();
10 void timer_run_Reset();
11
```

```
12 # endif
```

Program 1.2: Mã nguồn file .h

### 3.4.2 File timer.c

```
1 # include "timer.h"
2
3 int MODE=0;
4 int timer0_counter = 0;
5 int timer0_flag = 0;
6 int timerReset_counter = 0;
7 int TIMER_CYCLE = 10;
8
9 void setTimer0 (int duration )
10 {
11     timer0_counter = duration / TIMER_CYCLE ;
12     timer0_flag = 0;
13 }
14
15 void timer_run ()
16 {
17     if( timer0_counter > 0)
18     {
19         timer0_counter --;
20         if( timer0_counter <= 0) timer0_flag = 1;
21     }
22 }
23
24 //RESET: MODE 1,2 --> MODE 0. (after 5s no have action)
25 void setTimerReset()
26 {
27     timerReset_counter = 500; //5s
28 }
29
30 void timer_run_Reset()
31 {
32     if( timerReset_counter > 0)
33     {
34         timerReset_counter --;
35         if( timerReset_counter <= 0) MODE=0;
36     }
37 }
```

Program 1.3: Mã nguồn file .c

## 3.5 Module buttonScan

### 3.5.1 File buttonScan.h

```

1 #ifndef INC_BUTTONSCAN_H_
2 #define INC_BUTTONSCAN_H_
3 void button_scan();
4 #endif /* INC_BUTTONSCAN_H_ */

```

Program 1.4: Mã nguồn buttonScan.h

### 3.5.2 File buttonScan.c

```

1 #include "main.h"
2 #include "clock_12LEDs.h"
3 #include "buttonScan.h"
4 #include "timer.h"
5
6 //previous state
7 static GPIO_PinState MENU    =GPIO_PIN_SET;
8 static GPIO_PinState INC     =GPIO_PIN_SET;
9 static GPIO_PinState DEC     =GPIO_PIN_SET;
10
11 //flag hold MENU / INC / DEC
12 static int HoldMENU =0;
13 static int HoldINC  =0;
14 static int HoldDEC  =0;
15
16
17 // this function only scan button and change MODE, hour,
18 // minute.
19 void button_scan()
20 {
21     //all button will read to "temp" (current state)
22     //scan MENU button
23     GPIO_PinState temp=HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_0);
24
25     //check hold button, if not -> increase MODE by 1
26     if (temp==GPIO_PIN_RESET &&
27         MENU==GPIO_PIN_RESET && HoldMENU==0)
28     {
29         MODE++;
30         if (MODE>2) MODE=2;
31         HoldMENU=1;
32     }
33     else
34     {
35         //after release button, turn off flag hold.
36         if (temp==GPIO_PIN_SET && MENU==GPIO_PIN_RESET)
37             HoldMENU=0;
38     }
39     MENU=temp;

```

```

39 // scan INC, DEC button as same as MENU button
40 if (MODE!=0)
41 {
42     //scan INC button
43     temp=HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_1);
44     if (temp==GPIO_PIN_SET && INC==GPIO_PIN_SET);
45     else
46     {
47         if (temp==GPIO_PIN_RESET && INC==GPIO_PIN_RESET &&
48 HoldINC==0)
49         {
50             if (MODE==1)
51             {
52                 hour--;
53                 if (hour<=0) hour=12;
54             }
55             else //MODE 2
56             {
57                 minute--;
58                 if(minute<=0) minute=12;
59             }
60             HoldINC=1;
61         }
62         else
63         {
64             if (temp==GPIO_PIN_SET && INC==GPIO_PIN_RESET)
65             {
66                 setTimerReset();
67                 HoldINC=0;
68             }
69         }
70     }
71     INC=temp;
72     //scan DEC button
73     temp=HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_2);
74     if (temp==GPIO_PIN_SET && DEC==GPIO_PIN_SET);
75     else
76     {
77         if (temp==GPIO_PIN_RESET && DEC==GPIO_PIN_RESET &&
78 HoldDEC==0)
79         {
80             if (MODE==1)
81             {
82                 hour++;
83                 if (hour>=13) hour=1;
84             }
85             else
86             {
87                 minute++;

```

```

86         if (minute >= 13) minute = 1;
87     }
88     HoldDEC = 1;
89 }
90 else
91 {
92     if (temp == GPIO_PIN_SET && DEC == GPIO_PIN_RESET)
93     {
94         setTimerReset();
95         HoldDEC = 0;
96     }
97 }
98 }
99 DEC = temp;
100 }
101 }

```

Program 1.5: Mã nguồn buttonScan.c

## 3.6 Module clockFSM

### 3.6.1 File clock\_12LEDs.h

```

1 #ifndef INC_CLOCK_12LEDS_H_
2 #define INC_CLOCK_12LEDS_H_
3
4 extern int hour;
5 extern int minute;
6
7 void setNumberOnClock(int num);
8 void clearNumberOnClock(int num);
9 void clock_fsm();
10
11 #endif /* INC_CLOCK_12LEDS_H_ */

```

Program 1.6: Mã nguồn clock\_12LEDs.h

### 3.6.2 File clock\_12LEDs.c

```

1 #include "main.h"
2 #include "clock_12LEDs.h"
3 #include "timer.h"
4
5 // Count down from 12
6 int hour = 12;
7 int minute = 12;
8 int second = 12;
9
10 // this function is reuse from Lab 1

```

```

11 void clearAllClock()
12 {
13     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6
14         |GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9
15         |GPIO_PIN_10|GPIO_PIN_11|GPIO_PIN_12|
16         GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15, GPIO_PIN_SET);
17 }
18 // this function is reuse from Lab 1
19 void setNumberOnClock(int num)
20 {
21     switch (num)
22     {
23         case 0: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15,
24             GPIO_PIN_RESET); break;
25         case 1: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4,
26             GPIO_PIN_RESET); break;
27         case 2: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5,
28             GPIO_PIN_RESET); break;
29         case 3: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6,
30             GPIO_PIN_RESET); break;
31         case 4: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7,
32             GPIO_PIN_RESET); break;
33         case 5: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8,
34             GPIO_PIN_RESET); break;
35         case 6: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9,
36             GPIO_PIN_RESET); break;
37         case 7: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10,
38             GPIO_PIN_RESET); break;
39         case 8: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11,
40             GPIO_PIN_RESET); break;
41         case 9: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12,
42             GPIO_PIN_RESET); break;
43         case 10: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13,
44             GPIO_PIN_RESET); break;
45         case 11: HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14,
46             GPIO_PIN_RESET); break;
47         default: break;
48     }
49 }
50 void clock_fsm()
51 {
52     clearAllClock();
53     if (MODE==0)
54     {
55         setNumberOnClock(12-second);
56         setNumberOnClock(12-minute);
57         setNumberOnClock(12-hour );

```

```

58
59     if (timer0_flag==1)
60     {
61         // 1h = 12p    // 1p=12s
62         second--;
63         if (second <= 0)
64         {
65             second=12;
66             minute--;
67             if (minute<=0)
68             {
69                 minute=12;
70                 hour--;
71                 if (hour<=0) hour=12;
72             }
73         }
74         // calculate per 1s
75         setTimer0(1000);
76     }
77 }
78 else
79 {
80     //Mode 1: only display LEDhour
81     //Mode 2: only display LEDminute
82     if (MODE==1) setNumberOnClock(12-hour);
83     else setNumberOnClock(12-minute);
84 }
85 }

```

Program 1.7: Mã nguồn clock\_12LEDs.c

## 4 Video demo

Đường link cho video demo:

<https://drive.google.com/file/d/1NRC<sub>P</sub>tZ7yydOqo-aQqzRfQp<sub>J</sub>W1hGBRj/view?usp=sharing>