

BÁO CÁO THỰC HÀNH BÀI 1

Môn học: **CHUYÊN ĐỀ THIẾT KẾ HỆ THỐNG NHÚNG 1**- Mã lớp: **CE437.N11**
Giảng viên hướng dẫn thực hành: **Phạm Minh Quân**

Thông tin các sinh viên	Mã số sinh viên	Họ và tên sinh viên
	19520887	Phạm Trung Quốc
	19521651	Phạm Trọng Huỳnh
	19520928	Viên Minh Tân
	19520036	Phạm Quốc Đăng
Link các tài liệu tham khảo <i>(nếu có)</i>		
Đánh giá của giảng viên: + <i>Nhận xét</i> + <i>Các lỗi trong chương trình</i> + <i>Gợi ý</i>		

Mục lục

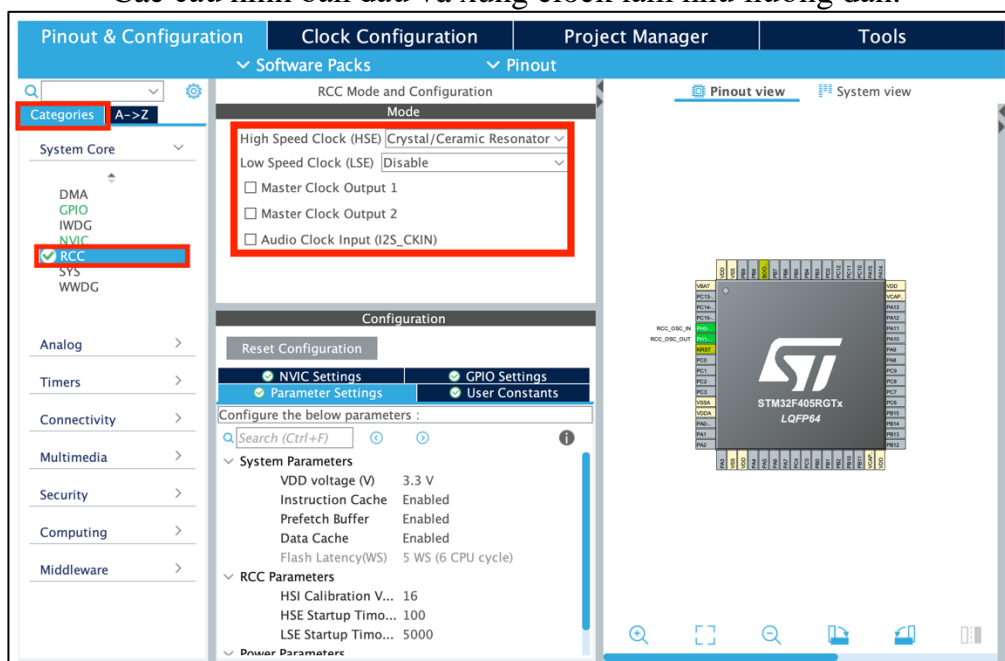
BÁO CÁO THỰC HÀNH BÀI 1	1
Yêu cầu: Lập trình điều khiển GPIO trên KIT Open405R-C Package A, STM32F4 Development Board.....	3
1. Thiết lập các cấu hình liên quan.....	3
2. Sơ đồ khối.....	7
3. Mã nguồn và giải thích.....	8
4. Video demo.....	10

Yêu cầu: Lập trình điều khiển GPIO trên KIT Open405R-C Package A, STM32F4 Development Board

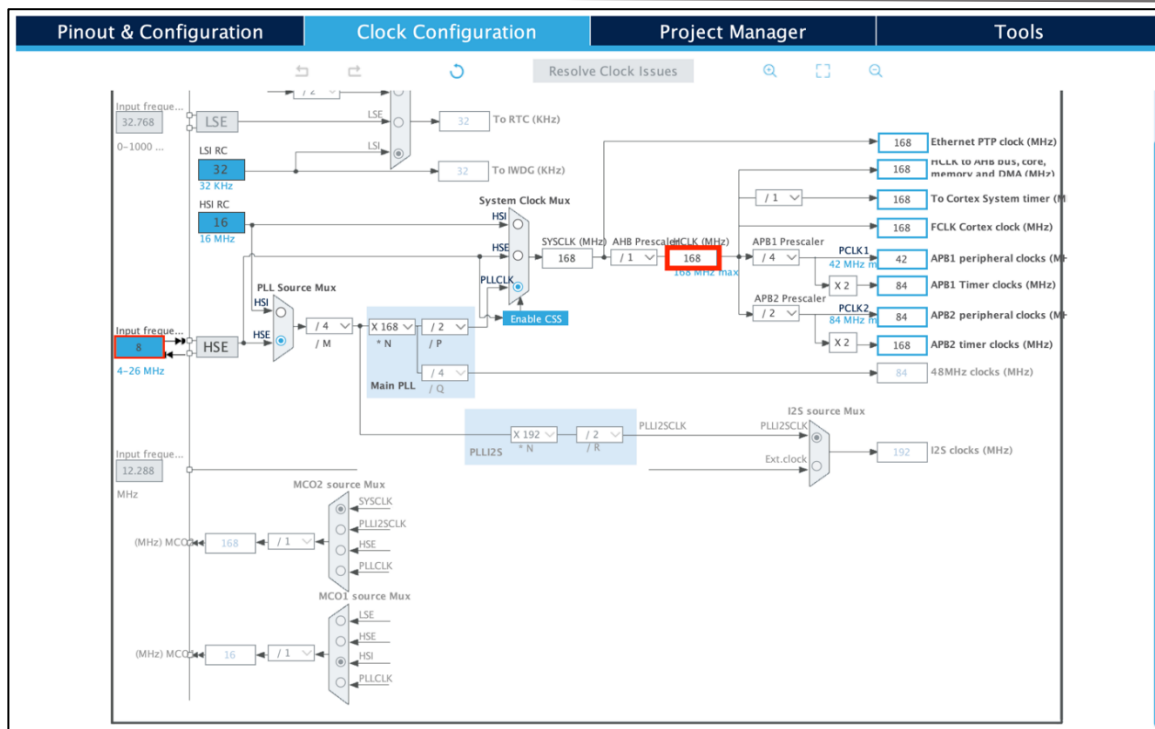
- Viết chương trình để tạo ra ít nhất 2 hiệu ứng chớp/tắt trên các LED sử dụng timer để điều khiển
- Đọc trạng thái nút nhấn tại chân PA0: nếu nút PA0 được nhấn thì sinh viên thực hiện giảm thời gian chu kỳ của các hiệu ứng LED ở trên.
- Đọc trạng thái nút nhấn tại chân PA1: nếu nút PA1 được nhấn thì sinh viên thực hiện tăng thời gian chu kỳ của các hiệu ứng LED ở trên
- Nếu cả 2 nút cùng được nhấn thì sinh viên sẽ thay đổi hiệu ứng LED mới sau khi cả 2 nút nhấn được thả ra

1. Thiết lập các cấu hình liên quan

- Các cấu hình ban đầu và xung clock làm như hướng dẫn.

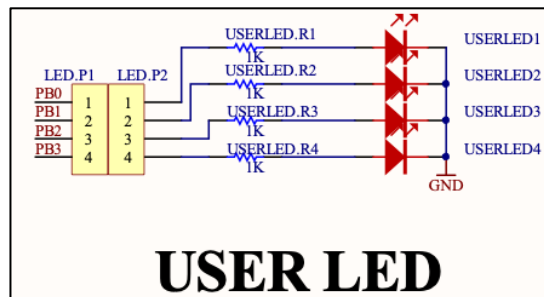


Hình 1-1: Chọn nguồn xung clock cho vi điều khiển

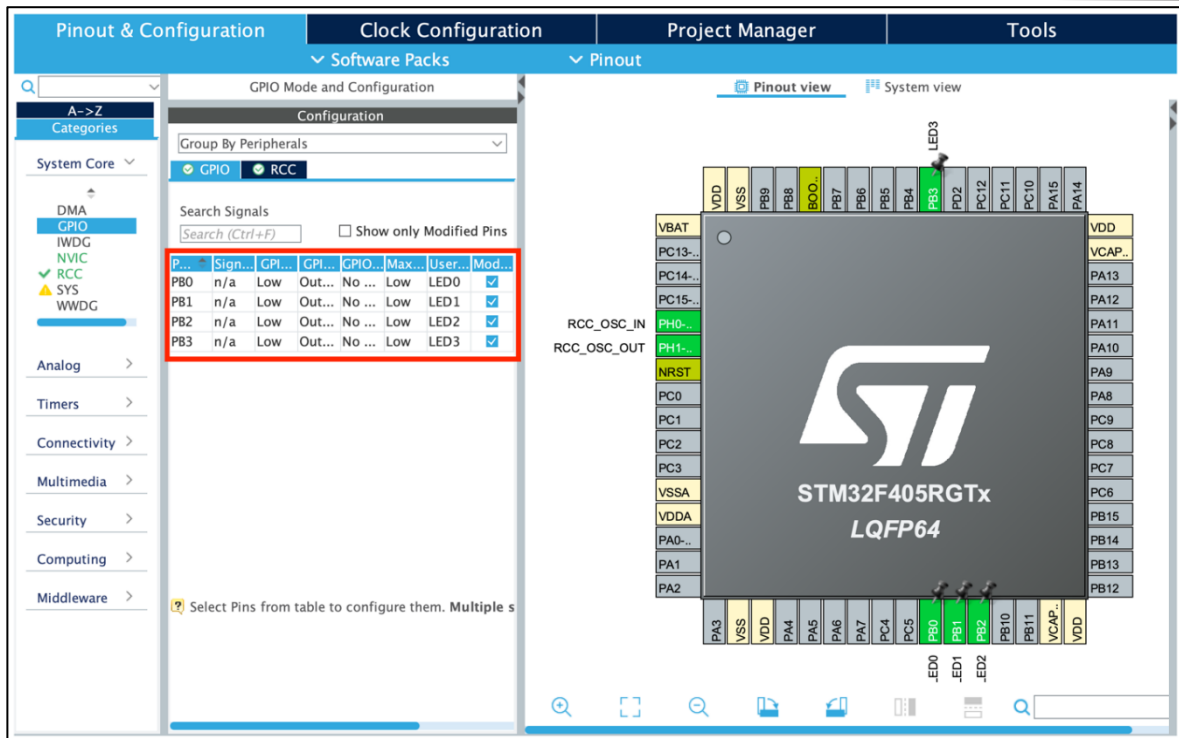


Hình 1-2: Cấu hình xung clock cho vi điều khiển

- Thực hiện cấu hình cho 4 User LEDs trên KIT Open405R-C Package A, STM32F4 Development Board được kết nối sẵn với các chân PB0, PB1, PB2, PB3 của vi điều khiển:

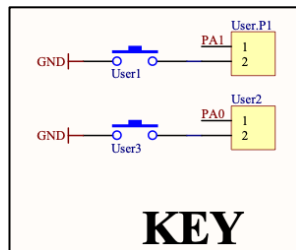


Hình 1-3: Sơ đồ nguyên lý các USER LED

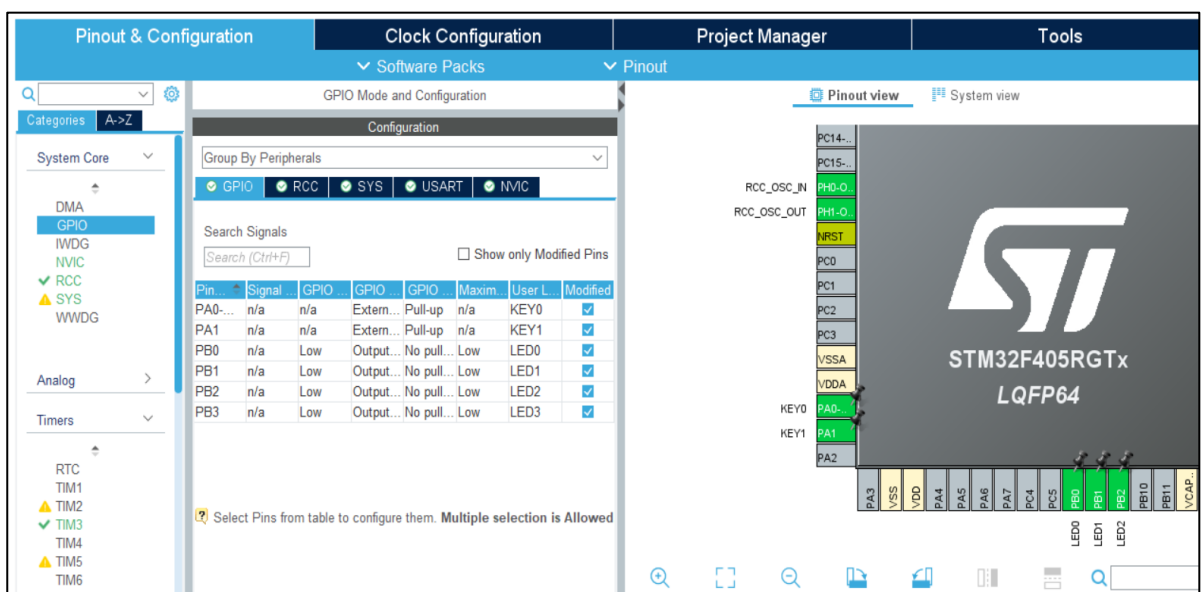


Hình 1-4: Cấu hình và đặt tên cho các chân USER LED

- Thực hiện cấu hình cho 2 User Key trên KIT Open405R-C Package A, STM32F4 Development Board được kết nối sẵn với các chân PA0, PA1 của vi điều khiển

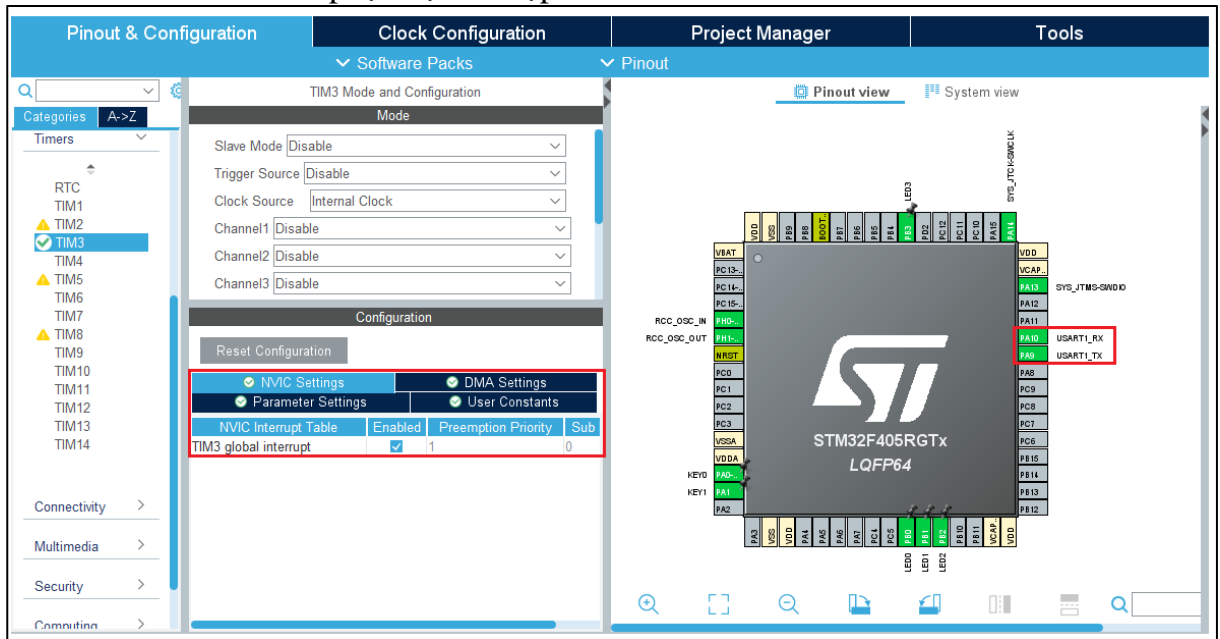


Hình 1-5: Sơ đồ nguyên lý các USER KEY

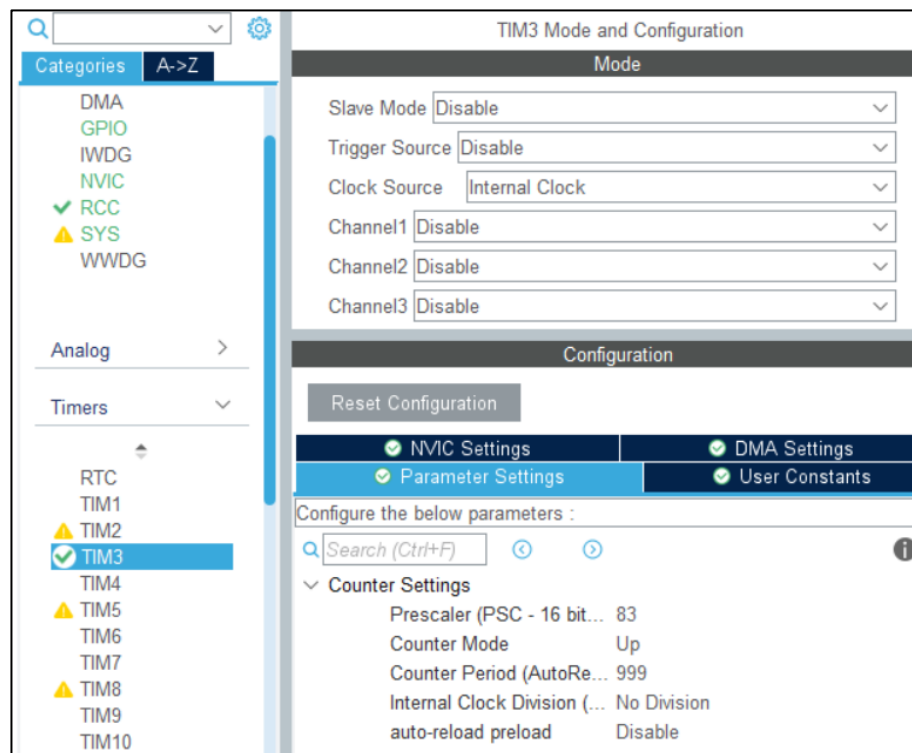


Hình 1-6: Cấu hình và đặt tên cho các USER KEY

- Thực hiện cấu hình Timer (TIM3) phục vụ cho việc thay thế HAL_Delay và USART1 phục vụ cho nạp code.



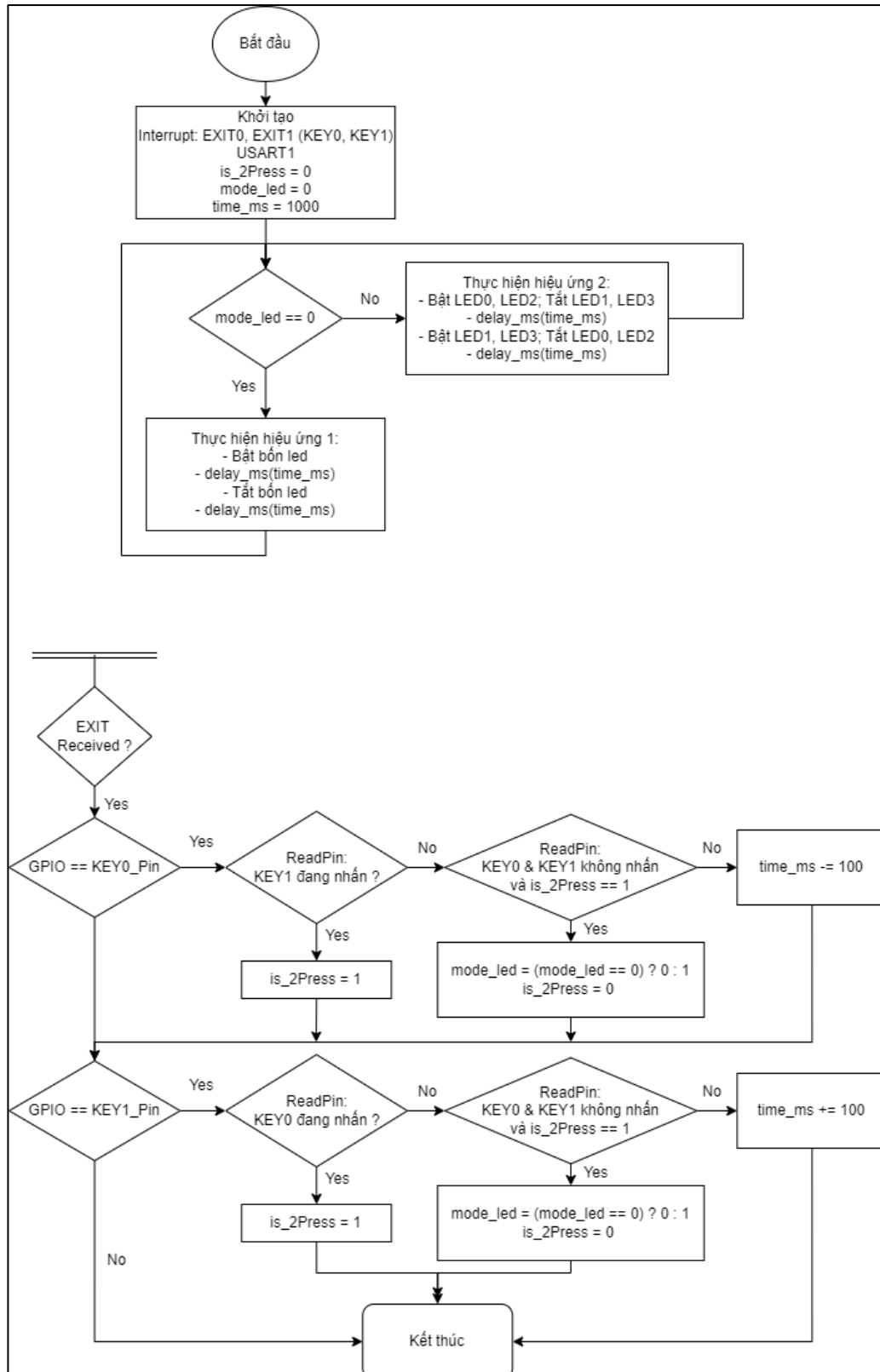
Hình 1-7: Cấu hình cho phép Interrupt timer3



Hình 1-8: Setting các Parameter của timer3

- Setting tim3:
 - Với HCLK = 168Mhz => APB1 Timer1 = 84 Mhz. Đặt Prescaler = 83 (84 - 1)
 - Counter Period = 999 (1000 – 1)
 ⇒ Timer/Counter = 1Mhz => Timer sẽ ticks mỗi 1 microsecond

2. Sơ đồ khối



Hình 2-1: Sơ đồ khối của Yêu cầu lab 1

Khi khởi động bốn LED sẽ hoạt động theo hiệu ứng 1 (mode_led được khởi tạo là 0)

Khi hai nút nhấn nhận được tương tác thì EXIT Callback được gọi (giải thích chi tiết ở phần sau)

3. Mã nguồn và giải thích

```
41 /* Private variables -----*/
42 TIM_HandleTypeDef htim3;
43
44 UART_HandleTypeDef huart1;
45
46 uint16_t time_ms = 1000;
47 int state_0 = 0 , state_1 = 0, mode_led = 0, is_2Press = 0;
48
```

Khởi tạo các biến:

- htim3, huart1
- state_0 , state_1 (phục vụ kiểm tra trạng thái nút KEY0, KEY1)
- mode_led
- is_2Press (phục vụ kiểm tra trạng thái nhấn đồng thời của hai nút KEY0, KEY1)

```
53 /* Private function prototypes -----*/
54 void SystemClock_Config(void);
55 static void MX_GPIO_Init(void);
56 static void MX_TIM3_Init(void);
57 static void MX_USART1_UART_Init(void);
```

Gọi các hàm khởi tạo cần thiết cho hệ thống

```
64 void delay_1ms (void)
65 {
66     __HAL_TIM_SET_COUNTER(&htim3, 0); // set the counter value a 0
67     while (__HAL_TIM_GET_COUNTER(&htim3) < 999); // wait for the counter
68 }
69
70 void delay_ms(int time)
71 {
72     int i = 0;
73     for(i = 0; i < time; i++)
74     {
75         delay_1ms();
76     }
77 }
```

Hàm delay_1ms: sử dụng htim3 và chờ đếm đủ 1ms (với 1 ticks htim3 = 1 microsecond => 1000 lần)

Hàm delay_ms: gọi hàm “delay_1ms()” time lần => time (ms). Với time là đầu vào của hàm.


```

117 while (1)
118 {
119     /* USER CODE END WHILE */
120
121     /* USER CODE BEGIN 3 */
122     if (mode_led == 0)
123     {
124         HAL_GPIO_WritePin(GPIOB, LED0_Pin, GPIO_PIN_SET);
125         HAL_GPIO_WritePin(GPIOB, LED1_Pin, GPIO_PIN_SET);
126         HAL_GPIO_WritePin(GPIOB, LED2_Pin, GPIO_PIN_SET);
127         HAL_GPIO_WritePin(GPIOB, LED3_Pin, GPIO_PIN_SET);
128         delay_ms(time_ms);
129         HAL_GPIO_WritePin(GPIOB, LED0_Pin, GPIO_PIN_RESET);
130         HAL_GPIO_WritePin(GPIOB, LED1_Pin, GPIO_PIN_RESET);
131         HAL_GPIO_WritePin(GPIOB, LED2_Pin, GPIO_PIN_RESET);
132         HAL_GPIO_WritePin(GPIOB, LED3_Pin, GPIO_PIN_RESET);
133         delay_ms(time_ms);
134     }
135     else
136     {
137         HAL_GPIO_WritePin(GPIOB, LED0_Pin, GPIO_PIN_SET);
138         HAL_GPIO_WritePin(GPIOB, LED1_Pin, GPIO_PIN_RESET);
139         HAL_GPIO_WritePin(GPIOB, LED2_Pin, GPIO_PIN_SET);
140         HAL_GPIO_WritePin(GPIOB, LED3_Pin, GPIO_PIN_RESET);
141         delay_ms(time_ms);
142         HAL_GPIO_WritePin(GPIOB, LED0_Pin, GPIO_PIN_RESET);
143         HAL_GPIO_WritePin(GPIOB, LED1_Pin, GPIO_PIN_SET);
144         HAL_GPIO_WritePin(GPIOB, LED2_Pin, GPIO_PIN_RESET);
145         HAL_GPIO_WritePin(GPIOB, LED3_Pin, GPIO_PIN_SET);
146         delay_ms(time_ms);
147     }
148 }

```

Vòng lặp while trong main phục vụ kiểm tra mode_led và WritePin các LED theo mode_led

```

321 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
322 {
323     if(GPIO_Pin == KEY0_Pin)
324     {
325         if (state_1 == 0)
326         {
327             state_0 = 1;
328             if (HAL_GPIO_ReadPin(GPIOA, KEY1_Pin) == 0)
329             {
330                 is_2Press = 1;
331             }
332             else if (HAL_GPIO_ReadPin(GPIOA, KEY0_Pin) == 1 && HAL_GPIO_ReadPin(GPIOA, KEY1_Pin) == 1 && is_Press == 1)
333             {
334                 if (mode_led == 0 ) mode_led = 1;
335                 is_2Press = 0;
336             }
337             else
338             {
339                 time_ms +=100;
340             }
341             state_0 = 0;
342         }
343     }
344 }

```

```

345     if(GPIO_Pin == KEY1_Pin)
346     {
347         if (state_0 == 0)
348         {
349             state_1 = 1;
350             if (HAL_GPIO_ReadPin(GPIOA, KEY0_Pin) == 0)
351             {
352                 is_2Press = 1;
353             }
354             else if (HAL_GPIO_ReadPin(GPIOA, KEY0_Pin) == 1 && HAL_GPIO_ReadPin(GPIOA, KEY1_Pin) == 1 && is_Press == 1)
355             {
356                 if (mode_led == 0 ) mode_led = 1;
357                 is_2Press = 0;
358             }
359             else
360             {
361                 time_ms -=100;
362             }
363             state_1 = 0;
364         }
365     }
366 }
367 }

```

EXIT_Callback được gọi có tương tác với các nút nhấn KEY0, KEY1: nếu KEY0 được nhấn => Tiến hành kiểm tra trạng thái KEY1 nếu KEY1 đang được nhấn thì gán is_2Press = 1, ngược lại nếu KEY1 không được nhấn tiếp tục kiểm tra nếu KEY0, KEY1 đang không được nhấn và is_2Press = 1 thì thực hiện đổi mode_led và gán lại is_2Press = 0, Ngược lại tiến hành tăng tims_ms lên 100(ms). Tương tự với khi KEY1 được nhấn.

4. Video demo

Video demo được đặt ở đường dẫn sau:

https://drive.google.com/file/d/11PnyvCZKy3nowCu4nUqpqFXk8yHzMcdQ/view?usp=share_link