BÁO CÁO THỰC HÀNH BÀI Lab1_GPIO_Timer

Môn học: **Chuyên đề thiết kế hệ thống nhúng 1** - Mã lớp: **CE437.P11**
Giảng viên hướng dẫn thực hành: **Phạm Minh Quân**

Thông tin sinh viên	**Mã số sinh viên**	**Họ và tên**
	22521472	**Phạm Quốc Tiến**
	22521570	**Trịnh Thành Trung**
	22521564	**Nguyễn Đức Trung**
Link các tài liệu tham khảo *(nếu có)*		
Đánh giá của giảng viên: *+ Nhận xét* *+ Các lỗi trong chương trình* *+ Gợi ý*		

[Báo cáo chi tiết các thao tác, quy trình sinh viên đã thực hiện trong quá trình làm bài thực hành. Chụp lại hình ảnh màn hình hoặc hình ảnh kết quả chạy trên sản phẩm. Mô tả và giải thích chương trình tương ứng để cho ra kết quả như hình ảnh đã trình bày.]

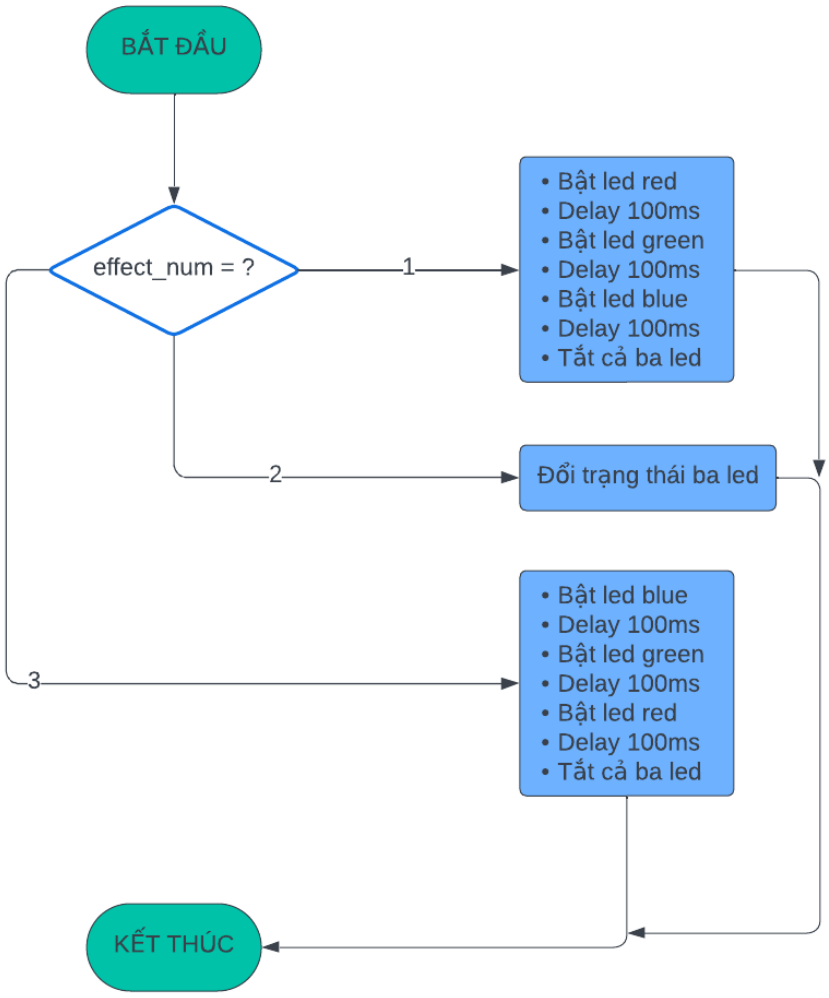
Mục lục

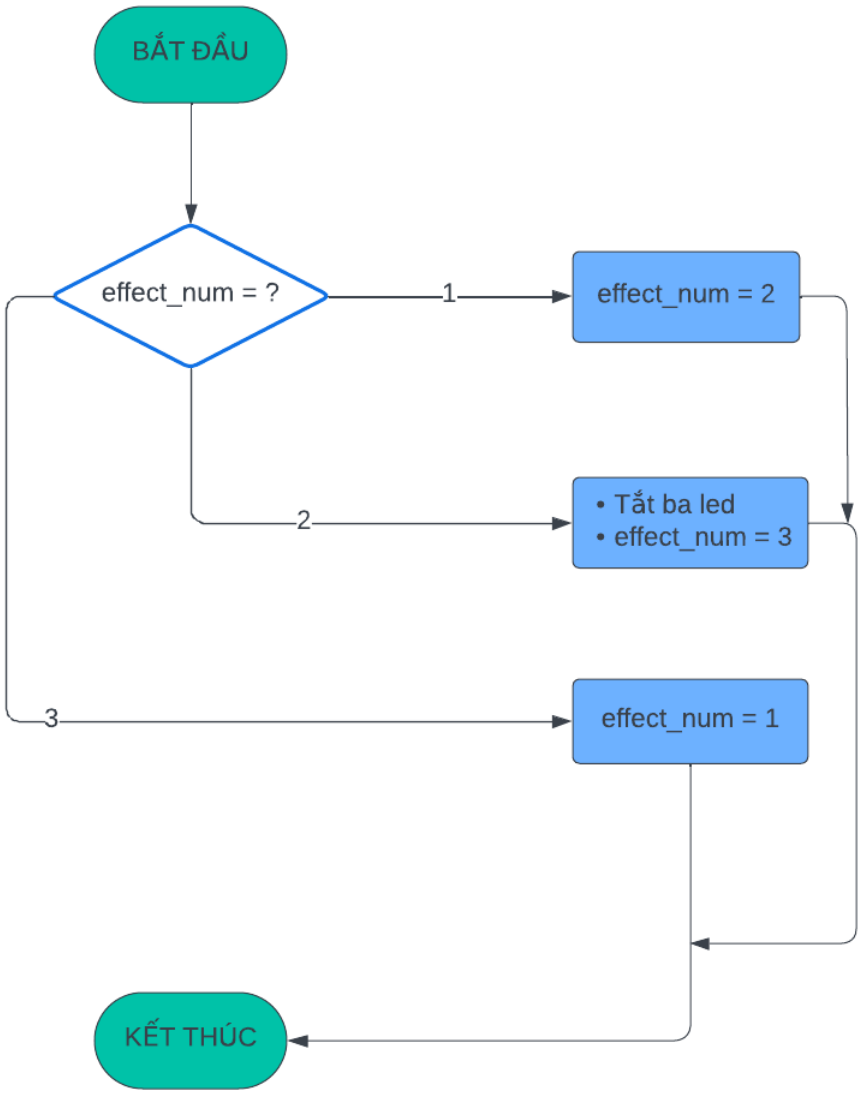
1) Ý tưởng

Với yêu cầu của đề bài thì nhóm sẽ sử dụng ba bộ timer, một dành cho thời gian hiệu ứng chớp tắt và hai bộ còn lại để dành cho việc xử lý nút bấm.

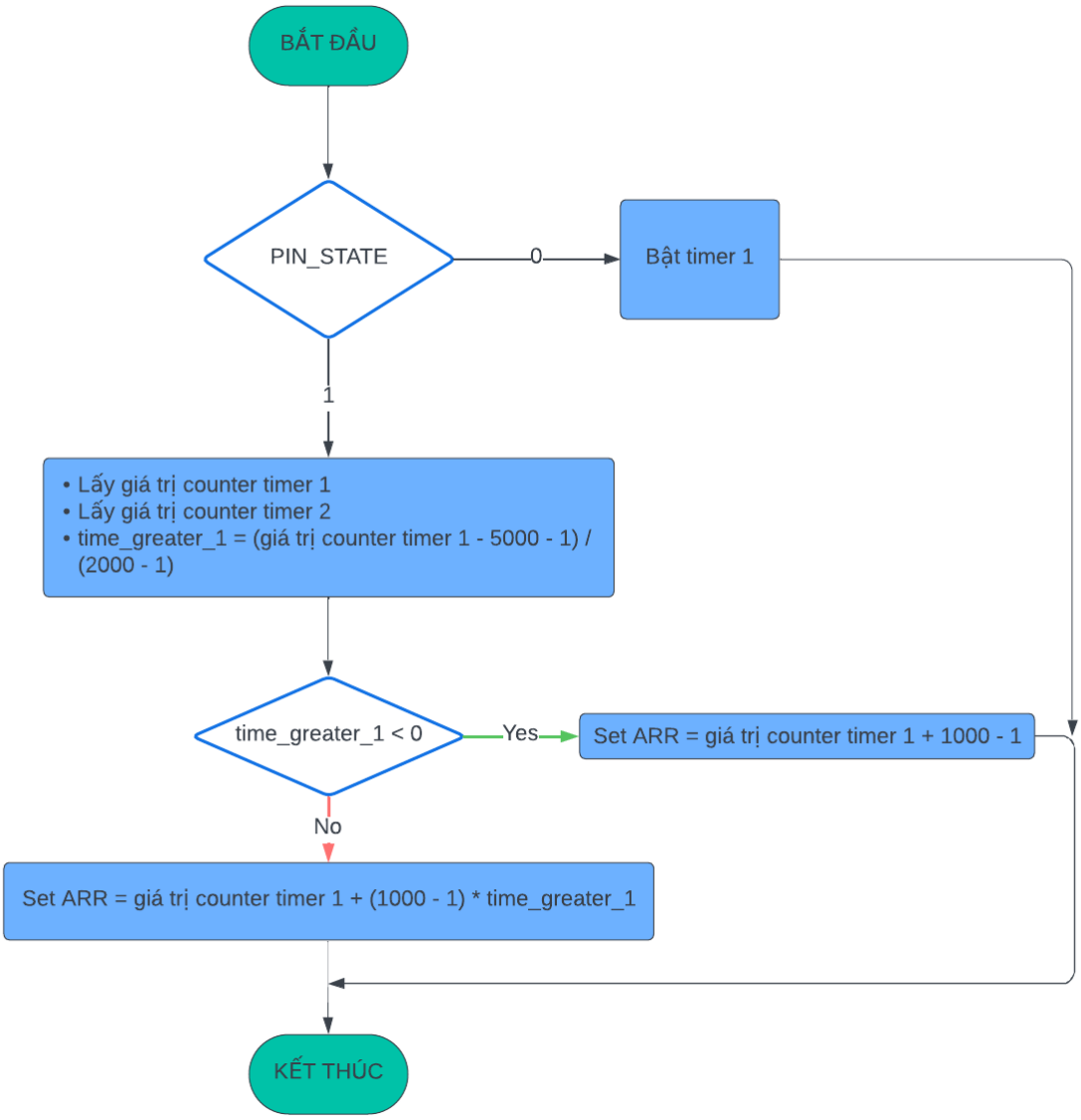
Việc xử lý nút bấm nhóm sẽ sử dụng interrupt cạnh lên và cạnh xuống. Lúc được nhấn thì sẽ bật timer lên và khi nút nhấn được nhả ra thì sẽ lấy giá trị thanh ghi counter của bộ timer và dừng bộ timer đó. Sau đó nhóm sẽ lấy giá trị thanh ghi counter trừ cho 500 và chia cho 200, việc trừ cho 500 là để so sánh xem liệu nút bấm có được giữ lâu hơn 500ms không còn việc chia cho 200 là để xác định số lần thời gian 200ms trôi qua kể từ khi nút được giữ đạt 500ms. Nếu giá trị tính được nhỏ hơn 0 thì sẽ thực hiện chức năng một của nút bấm, nếu không thì sẽ thực hiện chức năng thứ hai với giá trị tính được là số lần thời gian 200ms trôi qua.

2) Lưu đồ giải thuật

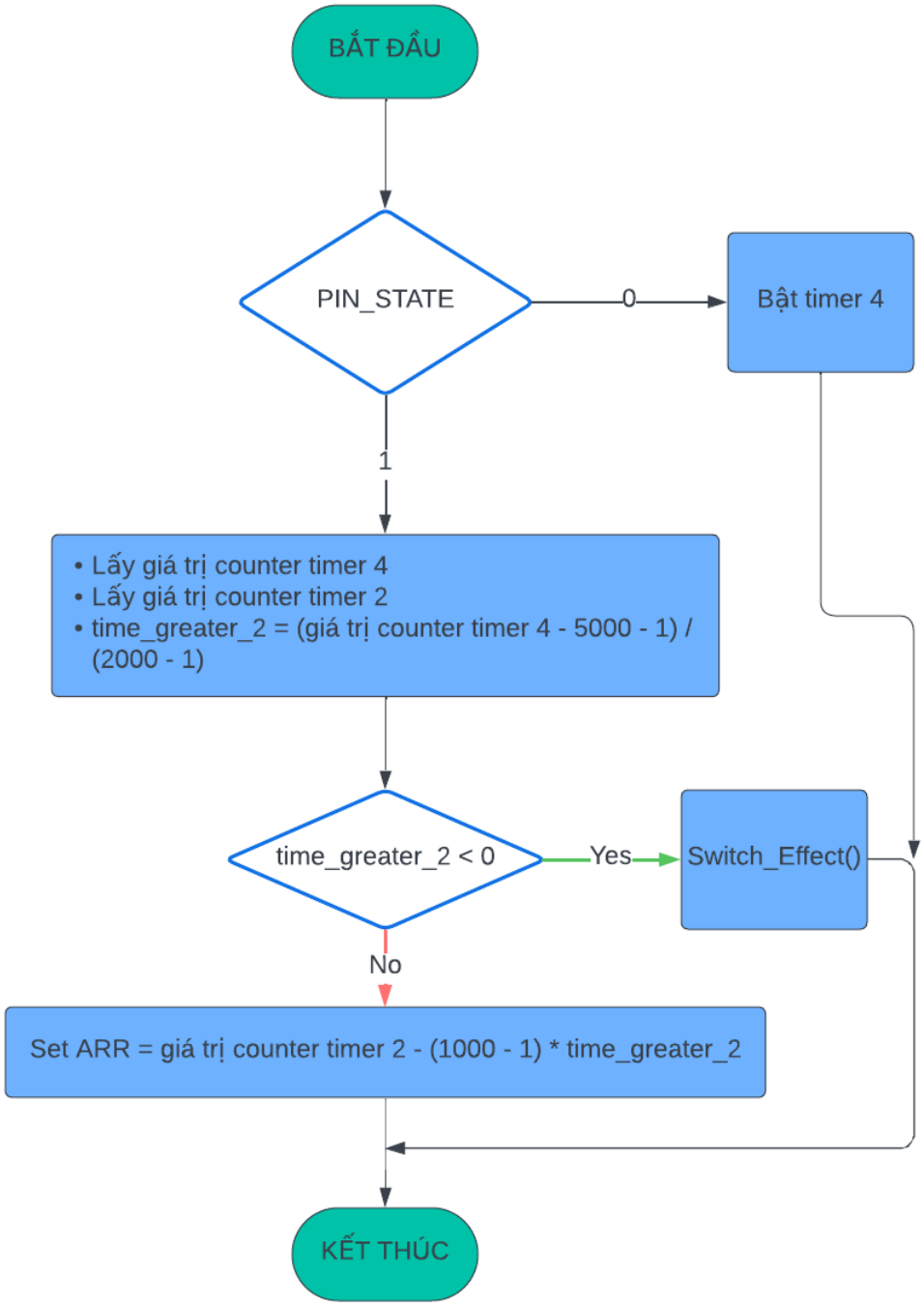




Hình 1: Lưu đồ giải thuật hàm Switch_Effect()



Hình 2: Lưu đồ giải thuật xử lý BTN1



Hình 3: Lưu đồ giải thuật xử lý BTN2

3) Code

link code

Khai báo trong file main.h

```
56 void Switch_Effect(void);
57 /* USER CODE END EFP */
58
59 /* Private defines -----------------------------------------------------------*/
60 #define BTN2_Pin GPIO_PIN_0
61 #define BTN2_GPIO_Port GPIOB
62 #define BTN2_EXTI_IRQn EXTI0_IRQn
63 #define BTN1_Pin GPIO_PIN_1
64 #define BTN1_GPIO_Port GPIOB
65 #define BTN1_EXTI_IRQn EXTI1_IRQn
66 #define LEDB_Pin GPIO_PIN_10
67 #define LEDB_GPIO_Port GPIOB
68 #define LEDG_Pin GPIO_PIN_11
69 #define LEDG_GPIO_Port GPIOB
70 #define LEDR_Pin GPIO_PIN_12
71 #define LEDR_GPIO_Port GPIOB
72
73 /* USER CODE BEGIN Private defines */
74 #define TIME_LIMIT_1 5000 - 1
75 #define TIME_LIMIT_2 5000 - 1
76 #define TIME_CHANGE 2000 - 1
77 #define TIME_EFFECT_ADD_OR_SUB 1000 - 1
78 #define EFFECT_NUM_1 1
79 #define EFFECT_NUM_2 2
80 #define EFFECT_NUM_3 3
81 /* USER CODE END Private defines */
```

Code trong file main.c

```
93   MX_GPIO_Init();
94   MX_TIM2_Init();
95   MX_TIM4_Init();
96   MX_TIM1_Init();
97   /* USER CODE BEGIN 2 */
98   HAL_TIM_Base_Start_IT(&htim2);
```

Khai báo trong file stm32f1xx_it.c

```
44 static int effect_num = 1;
45 /* USER CODE END PV */
81 extern TIM_HandleTypeDef htim2;
82 /* USER CODE BEGIN EV */
83 extern TIM_HandleTypeDef htim4;
84 extern TIM_HandleTypeDef htim1;
85 /* USER CODE END EV */
```

Code trong file stm32f1xx_it.c:
Hàm Switch_Effect()

```c
void Switch_Effect(void)
{
    switch(effect_num)
    {
        case EFFECT_NUM_1:
        {
            effect_num = EFFECT_NUM_2;
            break;
        }
        case EFFECT_NUM_2:
        {
            HAL_GPIO_WritePin(GPIOB, LEDR_Pin, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOB, LEDG_Pin, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOB, LEDB_Pin, GPIO_PIN_RESET);
            effect_num = EFFECT_NUM_3;
            break;
        }
        case EFFECT_NUM_3:
        {
            effect_num = EFFECT_NUM_1;
            break;
        }
    }
}
```

Hàm EXTI1_IRQHandler()

```c
void EXTI1_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI1_IRQn 0 */
    if(HAL_GPIO_ReadPin(GPIOB, BTN1_Pin) == 0)
    {
        HAL_TIM_Base_Start(&htim1);
    }
    else
    {
        uint32_t button_1_time = __HAL_TIM_GET_COUNTER(&htim1);
        uint32_t effect_time = __HAL_TIM_GET_COUNTER(&htim2);
        HAL_TIM_Base_Stop(&htim1);
        int time_greater_1 = (int)(button_1_time - TIME_LIMIT_1) / TIME_CHANGE;
        if(time_greater_1 < 0)
        {
            __HAL_TIM_SET_AUTORELOAD(&htim2, effect_time + TIME_EFFECT_ADD_OR_SUB);
        }
        else
        {
            __HAL_TIM_SET_AUTORELOAD(&htim2, effect_time + TIME_EFFECT_ADD_OR_SUB * (uint32_t)time_greater_1);
        }
    }

  /* USER CODE END EXTI1_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(BTN1_Pin);
  /* USER CODE BEGIN EXTI1_IRQn 1 */

  /* USER CODE END EXTI1_IRQn 1 */
}
```

Hàm EXTI0_IRQHandler()

```c
void EXTI0_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI0_IRQn 0 */
    if(HAL_GPIO_ReadPin(GPIOB, BTN2_Pin) == 0)
    {
        HAL_TIM_Base_Start(&htim4);
    }
    else
    {
        uint32_t button_2_time = __HAL_TIM_GET_COUNTER(&htim4);
        uint32_t effect_time = __HAL_TIM_GET_COUNTER(&htim2);
        HAL_TIM_Base_Stop(&htim4);
        int time_greater_2 = (int)(button_2_time - TIME_LIMIT_2) / TIME_CHANGE;
        if(time_greater_2 < 0)
        {
            Switch_Effect();
        }
        else
        {
            __HAL_TIM_SET_AUTORELOAD(&htim2, effect_time - TIME_EFFECT_ADD_OR_SUB * (uint32_t)time_greater_2);
        }
    }

  /* USER CODE END EXTI0_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(BTN2_Pin);
  /* USER CODE BEGIN EXTI0_IRQn 1 */

  /* USER CODE END EXTI0_IRQn 1 */
}
```

Hàm TIM2_IRQHandler()

```c
void TIM2_IRQHandler(void)
{
  /* USER CODE BEGIN TIM2_IRQn 0 */
    switch(effect_num)
        {
            case EFFECT_NUM_1:
            {
                HAL_GPIO_WritePin(GPIOB, LEDR_Pin, GPIO_PIN_SET);
                HAL_Delay(100);
                HAL_GPIO_WritePin(GPIOB, LEDG_Pin, GPIO_PIN_SET);
                HAL_Delay(100);
                HAL_GPIO_WritePin(GPIOB, LEDB_Pin, GPIO_PIN_SET);
                HAL_Delay(100);
                HAL_GPIO_WritePin(GPIOB, LEDR_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOB, LEDG_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOB, LEDB_Pin, GPIO_PIN_RESET);
                break;
            }
            case EFFECT_NUM_2:
            {
                HAL_GPIO_TogglePin(GPIOB, LEDR_Pin);
                HAL_GPIO_TogglePin(GPIOB, LEDG_Pin);
                HAL_GPIO_TogglePin(GPIOB, LEDB_Pin);
                break;
            }
            case EFFECT_NUM_3:
            {
                HAL_GPIO_WritePin(GPIOB, LEDB_Pin, GPIO_PIN_SET);
                HAL_Delay(100);
                HAL_GPIO_WritePin(GPIOB, LEDG_Pin, GPIO_PIN_SET);
                HAL_Delay(100);
                HAL_GPIO_WritePin(GPIOB, LEDR_Pin, GPIO_PIN_SET);
                HAL_Delay(100);
                HAL_GPIO_WritePin(GPIOB, LEDR_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOB, LEDG_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOB, LEDB_Pin, GPIO_PIN_RESET);
                break;
            }
        }

  /* USER CODE END TIM2_IRQn 0 */
  HAL_TIM_IRQHandler(&htim2);
  /* USER CODE BEGIN TIM2_IRQn 1 */
  /* USER CODE END TIM2_IRQn 1 */
}
```

4) Kết quả thực tế

video