

THÍ NGHIỆM: VI XỬ LÝ – VI ĐIỀU KHIỂN

BÁO CÁO

LAB1

Giáo viên hướng dẫn	Huỳnh Phúc Nghị
Sinh viên thực hiện	Trương Phi Trường
Mssv	1915749

Link cho các bài mô phỏng:

https://github.com/TRUONGTRUONG2304/Simulation_Proteus_LAB1

Exercise 1: In this exercise, the status of two LEDs are switched every 2 seconds, as demonstrated in the figure bellow.

Source code:

```
while (1)
{
    /* USER CODE END WHILE */
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
    HAL_Delay(2000);
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
    HAL_Delay(2000);
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
    /* USER CODE BEGIN 3 */
}
```

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.

Source code:

```
int state = 1;
while (1)
```

```

{
/* USER CODE END WHILE */
//  Cách 1
//  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
//  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_7);
//  HAL_Delay(5000);
//  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
//  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
//  HAL_Delay(5000);
//  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
//  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_7);
//  HAL_Delay(5000);
//  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
//  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
//  Cách 2
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
switch(state){
case 1:
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
    break;
case 2:
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
    break;
case 3:
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
    break;
}
HAL_Delay(1000);
state += 1;
if(state > 3) state = 1;
/* USER CODE BEGIN 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.

Source code:

```
int state = 1;
while (1)
{
    /* USER CODE END WHILE */
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_SET);
    switch(state){
        case 1:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_RESET);
            break;
        case 2:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, GPIO_PIN_RESET);
            break;
        case 3:
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, GPIO_PIN_RESET);
            break;
    }
    state++;
    if(state > 3) state = 1;
    HAL_Delay(1000);
    /* USER CODE BEGIN 3 */
}
```

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).

Source code:

```
void display7SEG(int count){
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
    switch(count){
case 0:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
        break;
case 1:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
        break;
case 2:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
        break;
case 3:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
        break;
case 4:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
        break;
case 5:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
```

```
        break;
    case 6:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
        break;
    case 7:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
        break;
    case 8:
        break;
    case 9:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
        break;
}
}
```