LED Sequence V3.0

By Omar Ashraf Taha

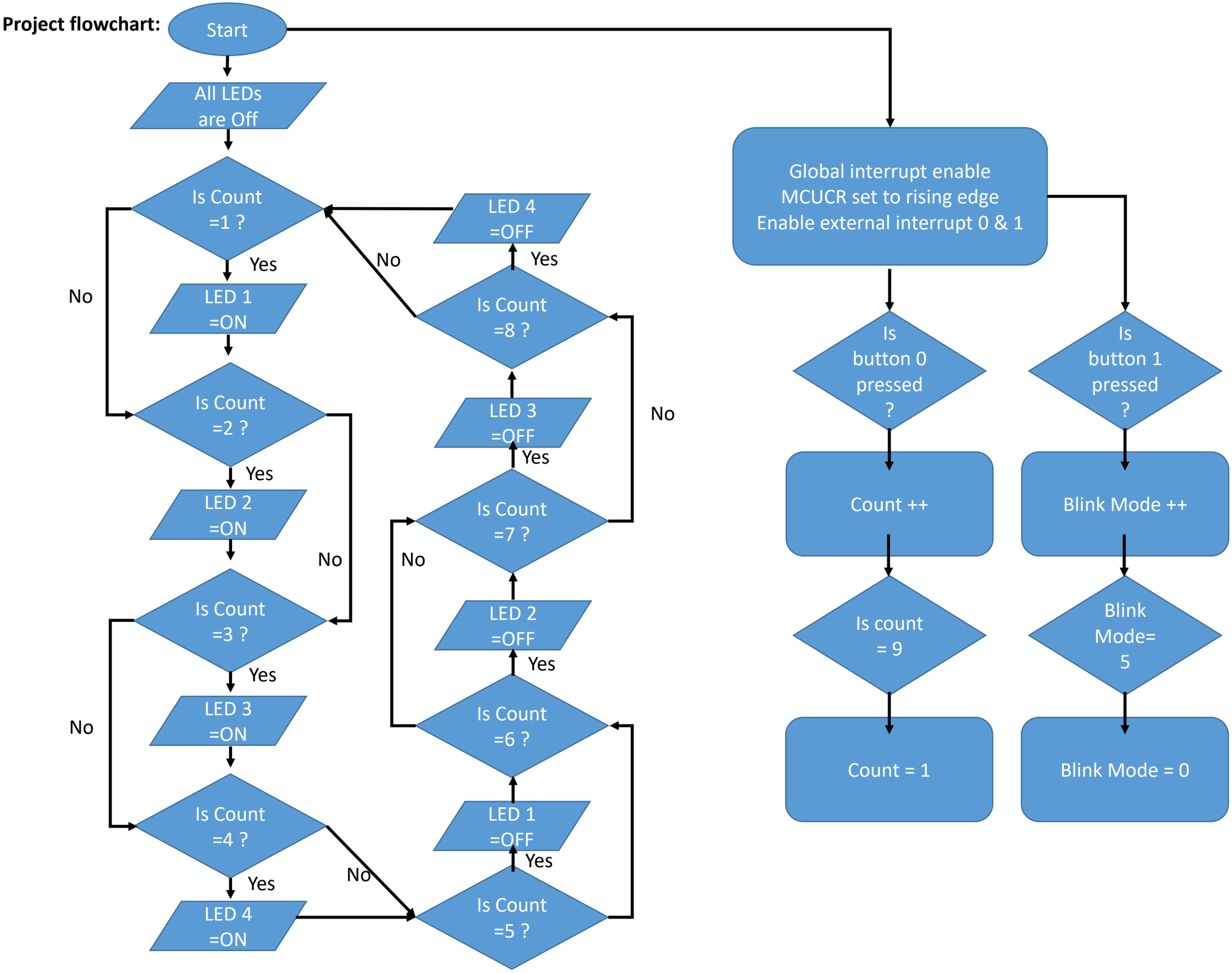
LED Sequence V3.0

Description :

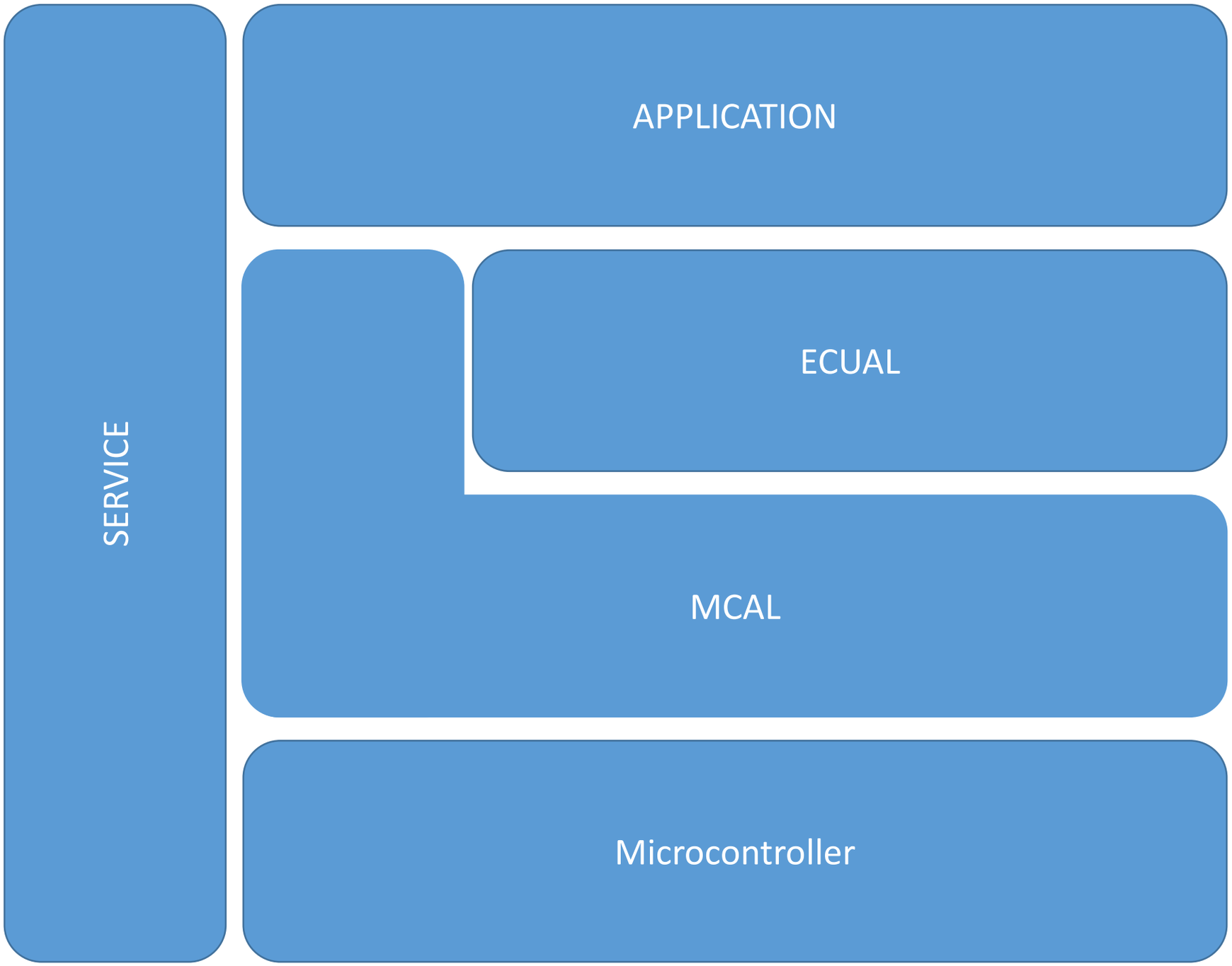
1. ***Hardware Requirements***
    1. Four LEDs (LED0, LED1, LED2, LED3)
    2. Two buttons (BUTTON0 and BUTTON1)
2. ***Software Requirements***
    1. Initially, all LEDs are OFF
    2. Once BUTTON0 is pressed, LED0 will blink with BLINK_1 mode
    3. Each press further will make another LED blinks BLINK_1 mode
    4. At the fifth press, LED0 will changed to be OFF
    5. Each press further will make only one LED is OFF
    6. This will be repeated forever
    7. The sequence is described below
        1. Initially (OFF, OFF, OFF, OFF)
        2. Press 1 (BLINK_1, OFF, OFF, OFF)
        3. Press 2 (BLINK_1, BLINK_1, OFF, OFF)
        4. Press 3 (BLINK_1, BLINK_1, BLINK_1, OFF)
        5. Press 4 (BLINK_1, BLINK_1, BLINK_1, BLINK_1)
        6. Press 5 (OFF, BLINK_1, BLINK_1, BLINK_1)
        7. Press 6 (OFF, OFF, BLINK_1, BLINK_1)
        8. Press 7 (OFF, OFF, OFF, BLINK_1)
        9. Press 8 (OFF, OFF, OFF, OFF)
        10. Press 9 (BLINK_1, OFF, OFF, OFF)
    8. When BUTTON1 has pressed the blinking on and off durations will be changed
        1. No press → **BLINK_1** mode (**ON**: 100ms, **OFF**: 900ms)
        2. First press → **BLINK_2** mode (**ON**: 200ms, **OFF**: 800ms)
        3. Second press → **BLINK_3** mode (**ON**: 300ms, **OFF**: 700ms)
        4. Third press → **BLINK_4** mode (**ON**: 500ms, **OFF**: 500ms)
        5. Fourth press → **BLINK_5** mode (**ON**: 800ms, **OFF**: 200ms)
        6. Fifth press → **BLINK_1** mode
    9. USE EXTERNAL INTERRUPTS

•

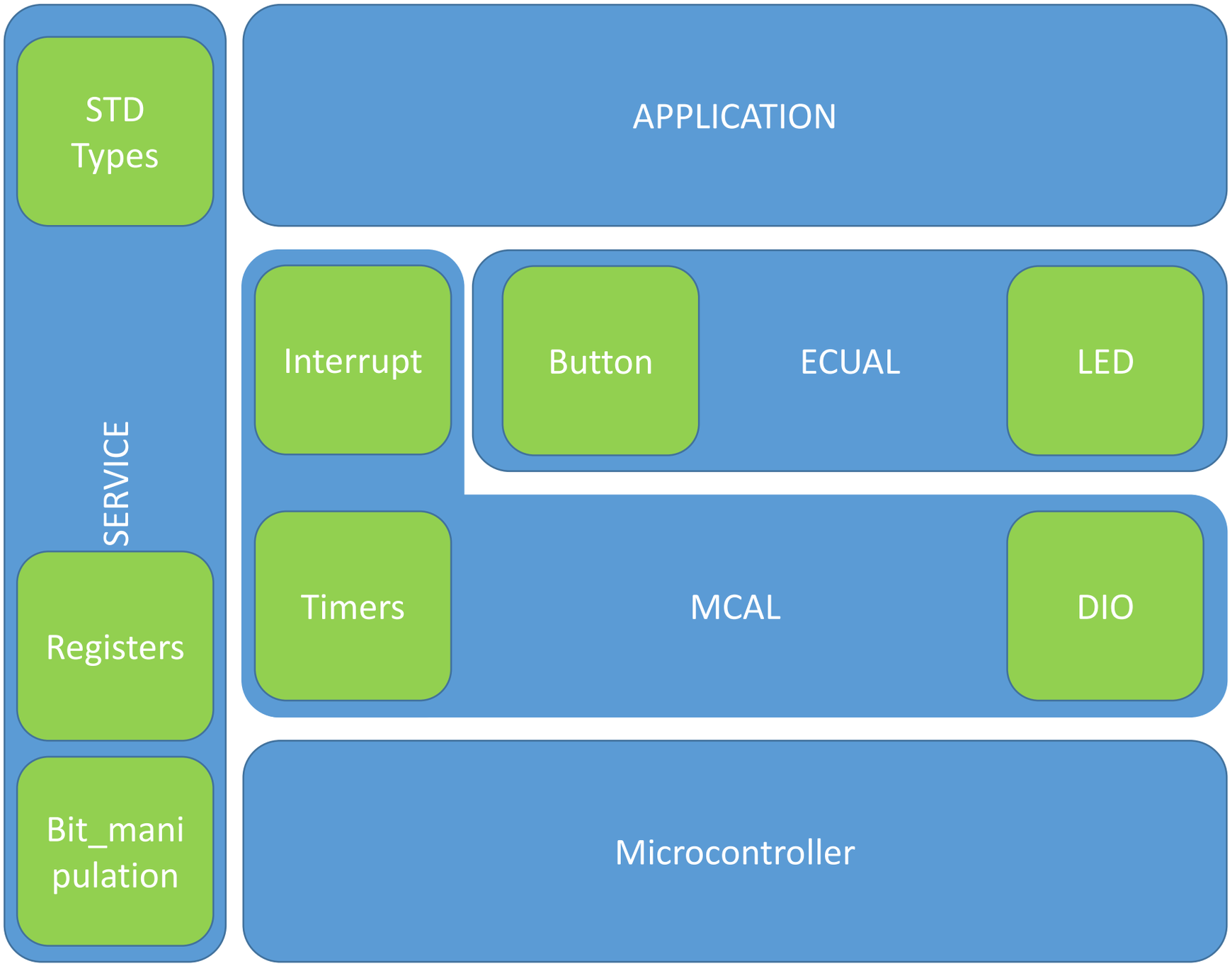
Project flowchart:



Layered Architecture:



Modules/Drivers:



APIs:

Button Driver:

- err_state BUTTON_init(uint8_t pinNumber, uint8_t portNumber);
- err_state BUTTON_read(uint8_t pinNumber, uint8_t portNumber, uint8_t *value);

LED Driver:

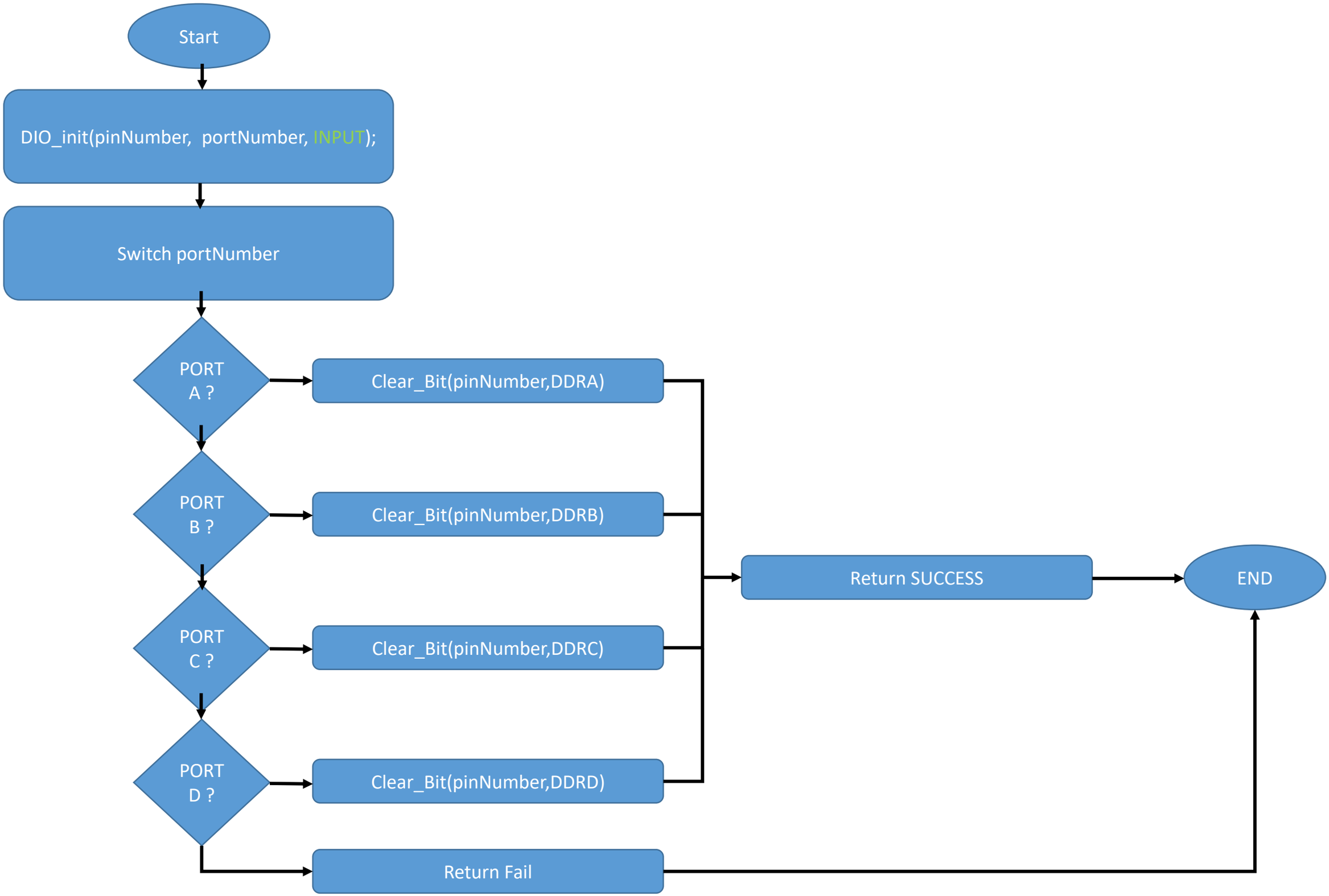
- err_state LED_init(uint8_t ledPin, uint8_t ledPort);
- err_state LED_on(uint8_t ledPin, uint8_t ledPort);
- err_state LED_off(uint8_t ledPin, uint8_t ledPort);
- err_state LED_toggle(uint8_t ledPin, uint8_t ledPort);
- err_state LED_blink(uint8_t ledPin, uint8_t ledPort,float on_time, float off_time);
- err_state LED_array_blink(uint8_t mask, uint8_t ledPort,float on_time, float off_time);
- err_state LED_array_on(uint8_t mask, uint8_t ledPort);
- err_state LED_array_off(uint8_t mask, uint8_t ledPort);

DIO Driver:

- err_state DIO_init(uint8_t pinNumber, uint8_t portNumber, pin_dir direction);
- err_state DIO_write(uint8_t pinNumber, uint8_t portNumber, pin_state value);
- err_state DIO_toggle(uint8_t pinNumber, uint8_t portNumber);
- err_state DIO_read(uint8_t pinNumber, uint8_t portNumber, uint8_t *value);
- err_state DIO_array_write(uint8_t mask, uint8_t portNumber, pin_state value);

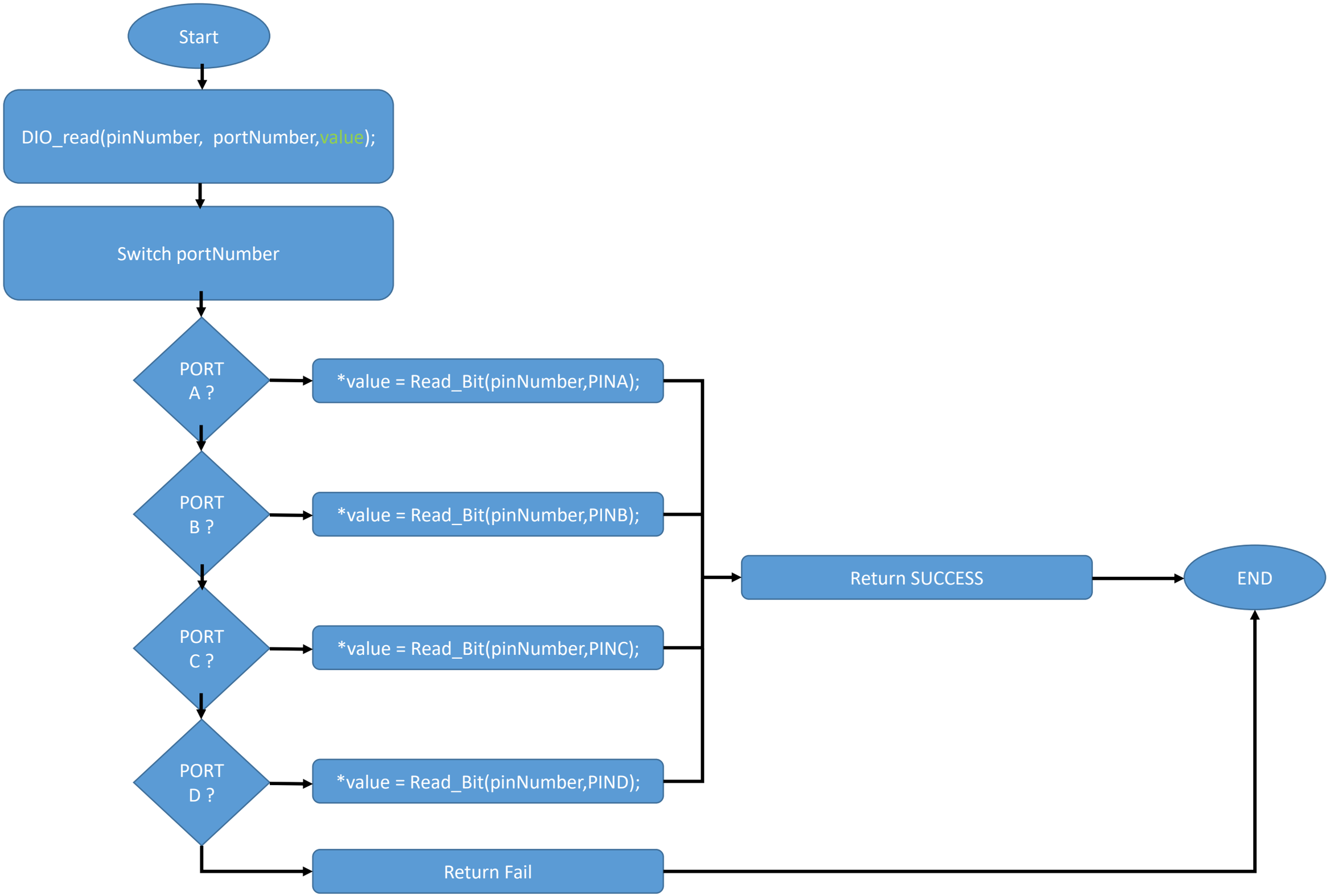
APIs flowchart:

err_state BUTTON_init(uint8_t pinNumber, uint8_t portNumber);



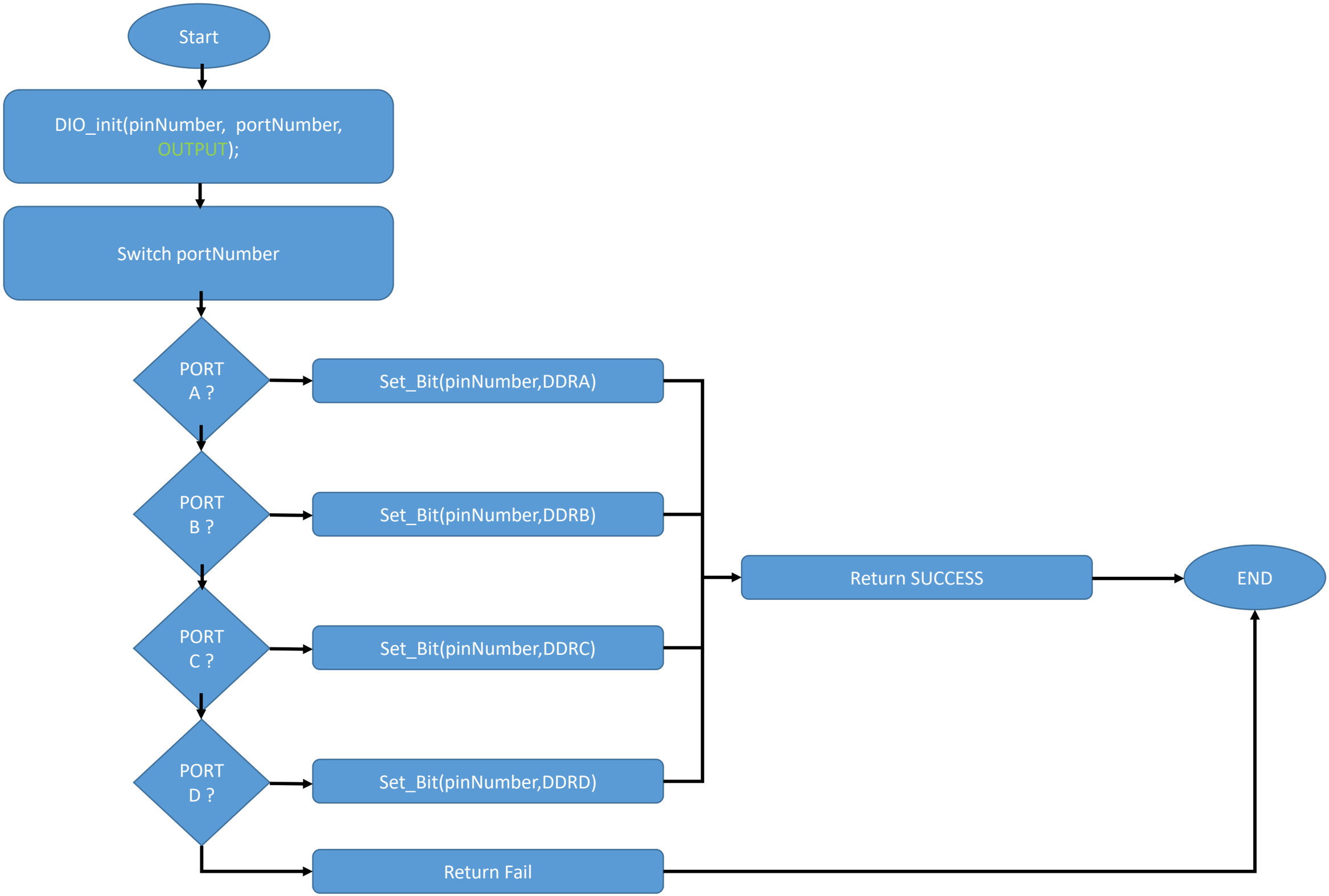
APIs flowchart:

err_state BUTTON_read(uint8_t pinNumber, uint8_t portNumber, uint8_t *value);



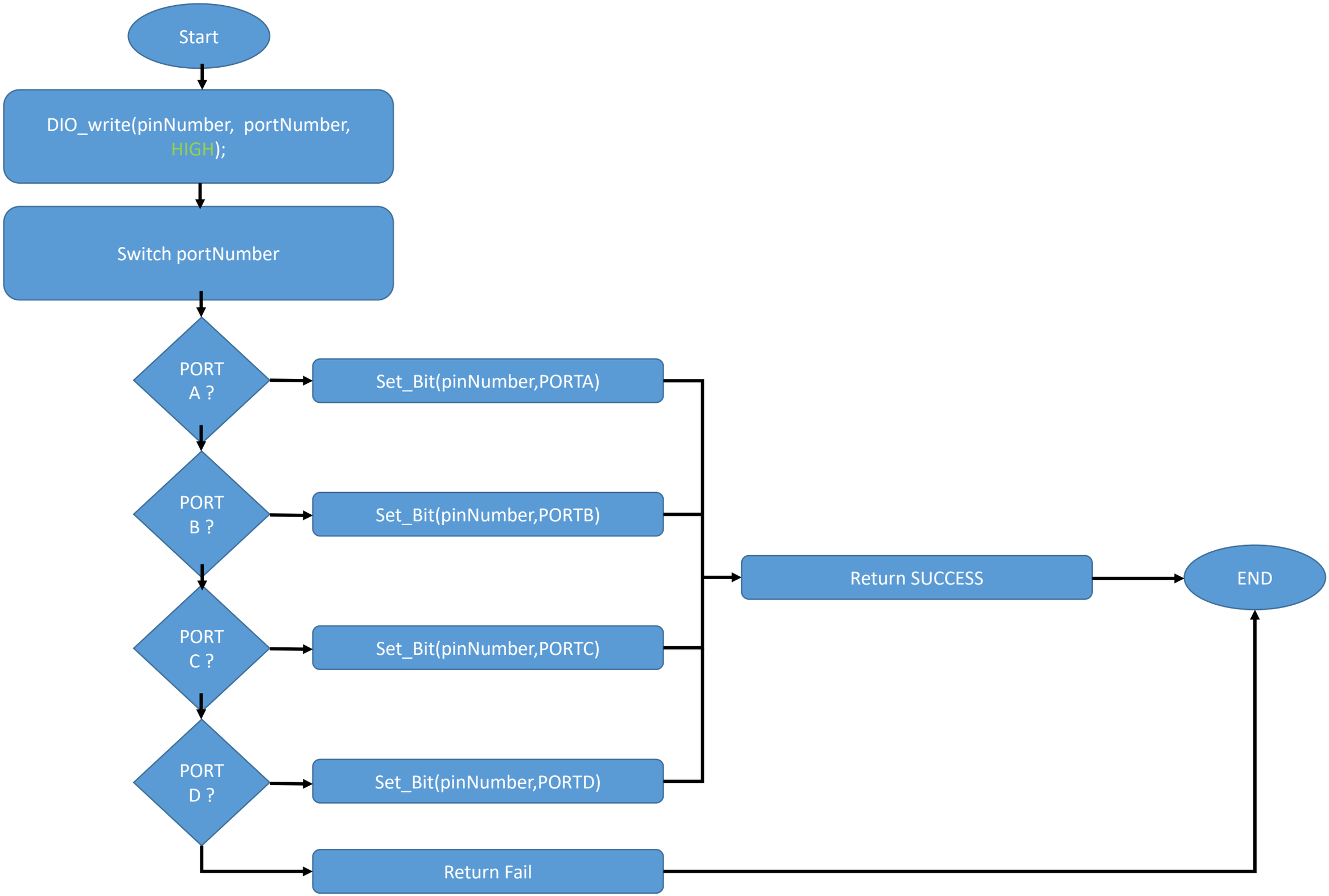
APIs flowchart:

err_state LED_init(uint8_t ledPin, uint8_t ledPort);



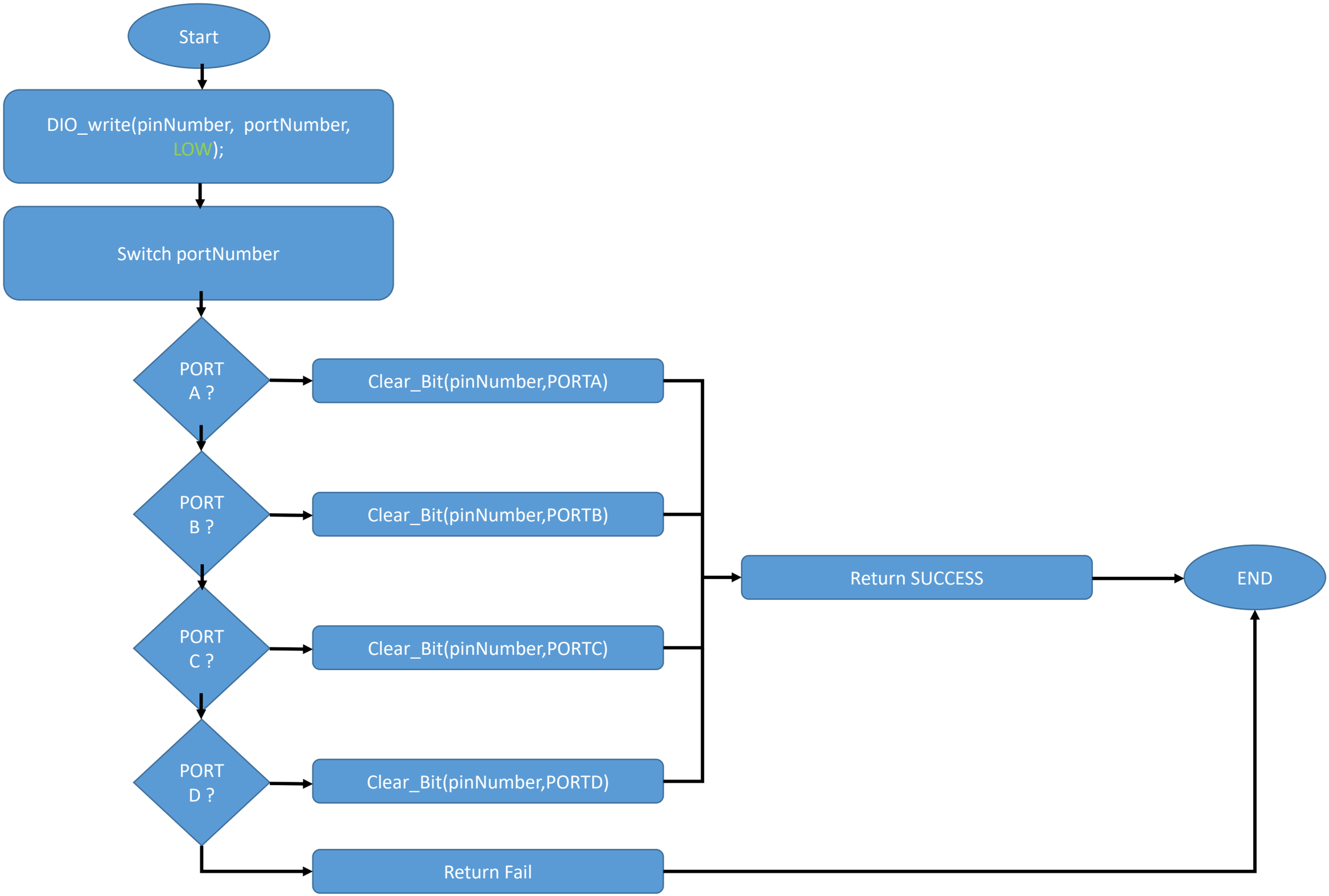
APIs flowchart:

err_state LED_on(uint8_t ledPin, uint8_t ledPort);



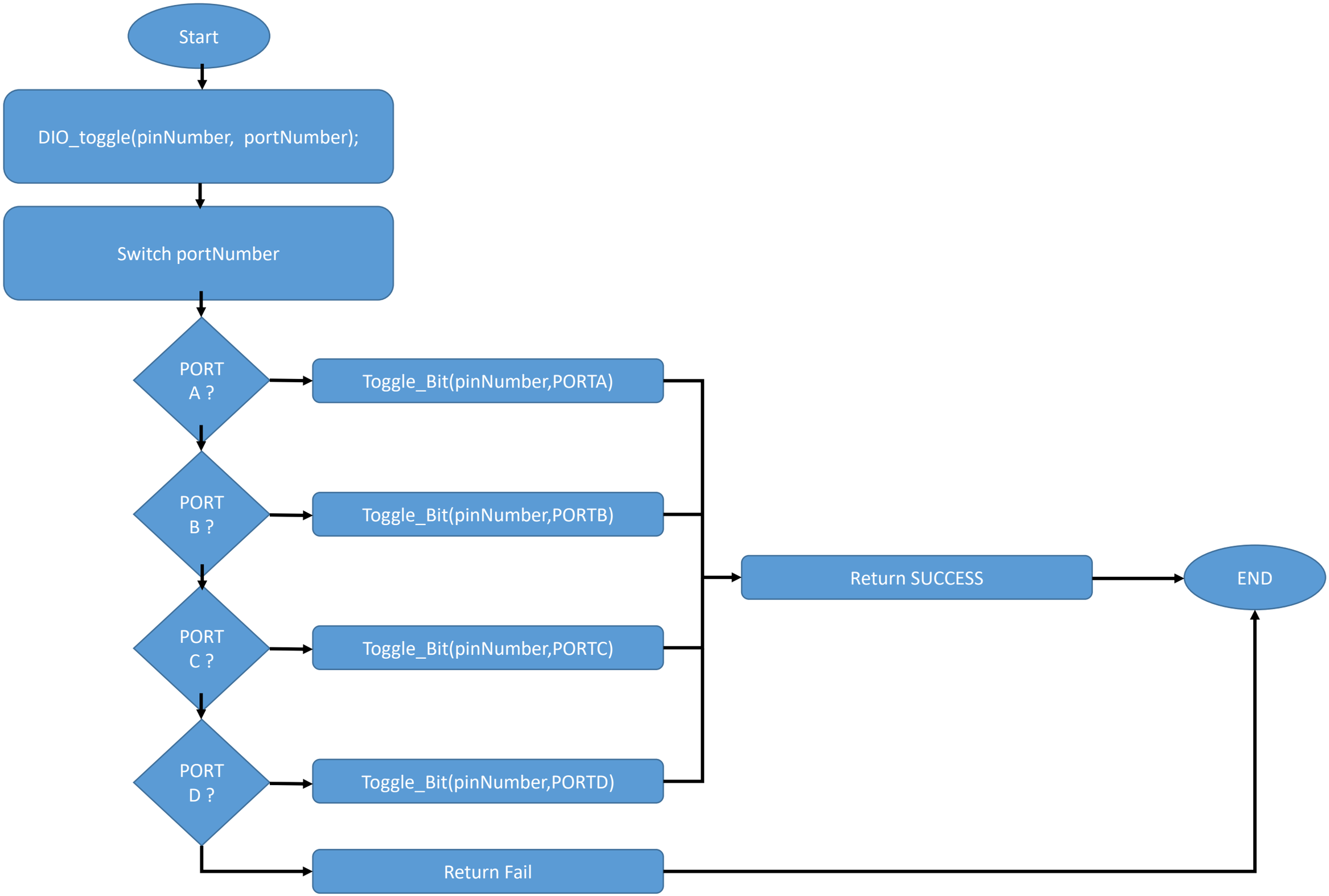
APIs flowchart:

err_state LED_off(uint8_t ledPin, uint8_t ledPort);



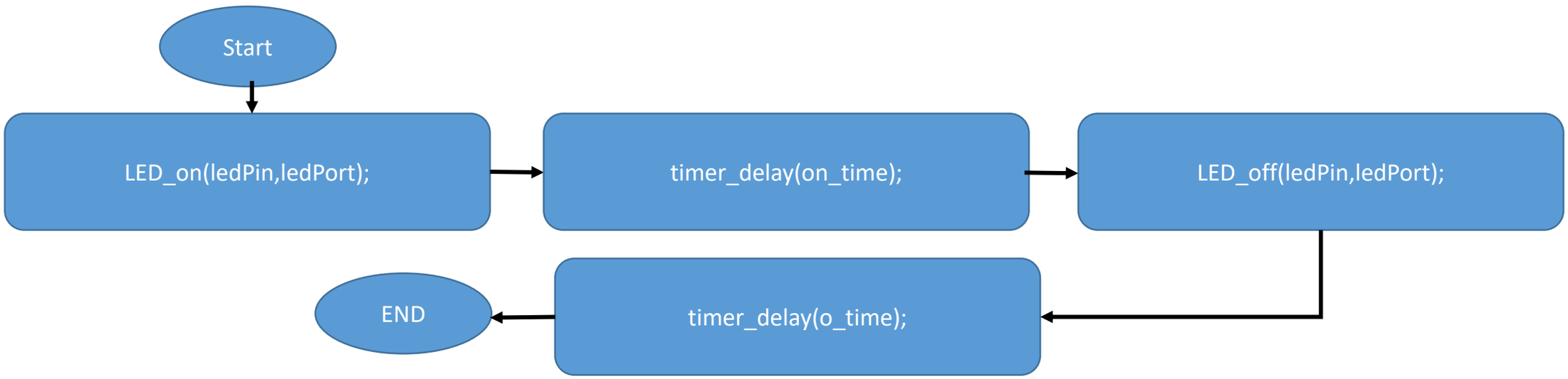
APIs flowchart:

err_state LED_toggle(uint8_t ledPin, uint8_t ledPort);



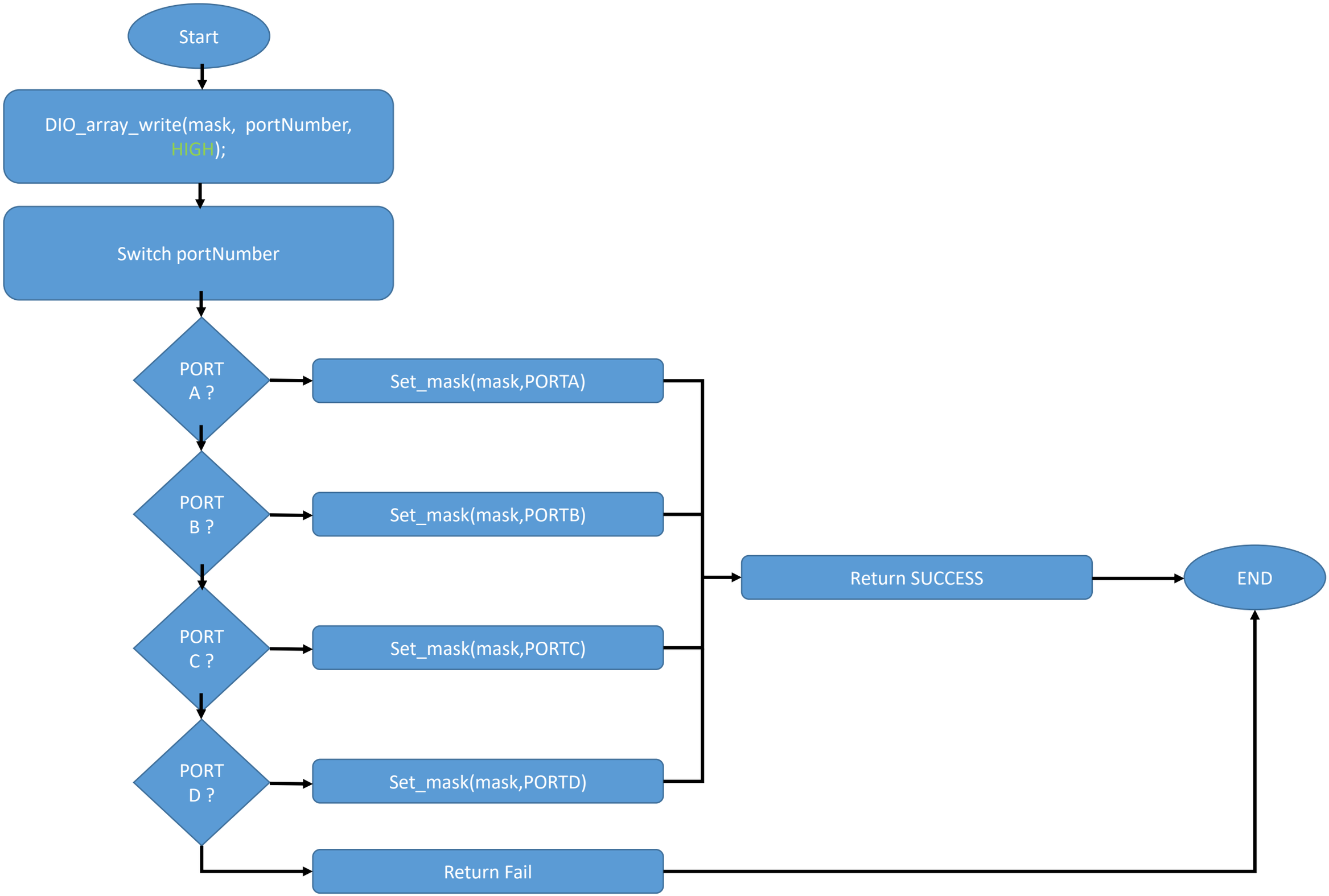
APIs flowchart:

err_state LED_blink(uint8_t ledPin, uint8_t ledPort,float on_time, float off_time);



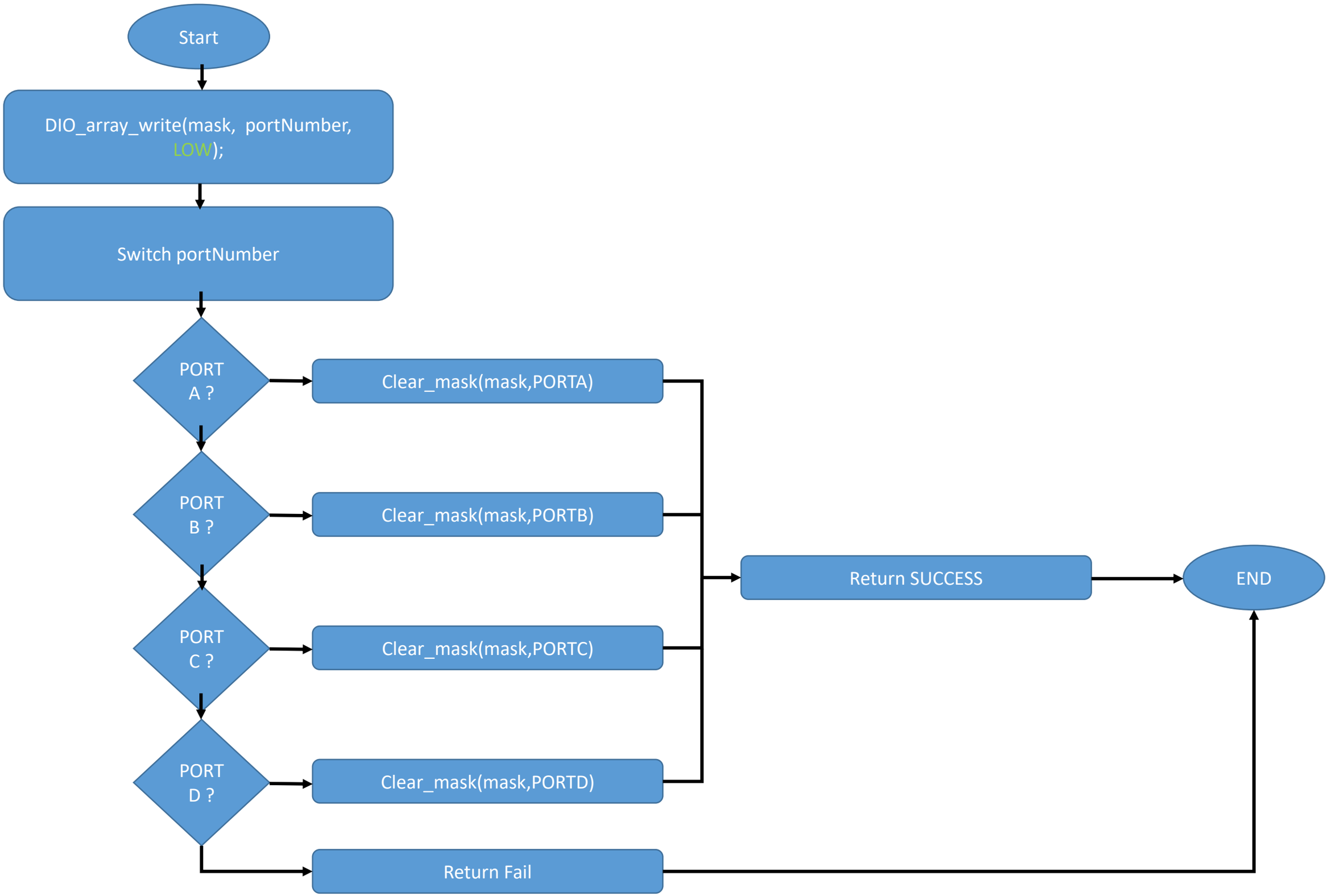
APIs flowchart:

err_state LED_array_on(uint8_t mask, uint8_t ledPort);



APIs flowchart:

err_state LED_array_off(uint8_t mask, uint8_t ledPort);



APIs flowchart:

err_state LED_array_blink(uint8_t mask, uint8_t ledPort,float on_time, float off_time);