

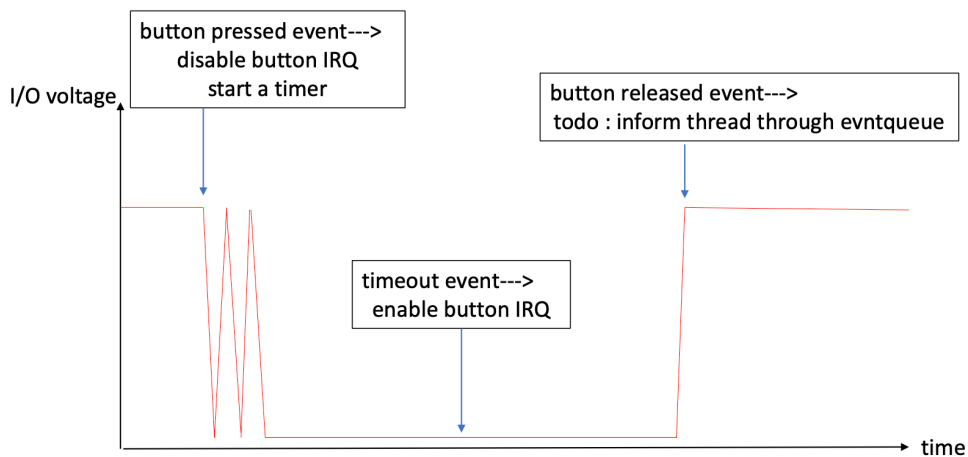
# Embedded System Lab 6

## Interrupt Service Routine and EventQueue

b07901095 吳宜庭

Github: <https://github.com/yitingwu31/eslab2021/tree/master/lab6>

### I. Code Design



1. `button_pressed()`:  
Use `disable_irq()` to disable button interrupts.  
When the button is pressed, a timer will be started. The timer `attach()` method calls the `button_release_detecting()` function after the timer's time is up (set to 3 seconds here).
2. `button_released()`:  
Flip the led signal and print out a message indicating button rise.
3. `button_release_detecting()`:  
This function is called after the timer set in `button_pressed()` is up, i.e. roughly 3 seconds after `button_pressed()` is executed. It will use the `enable_irq()` method to enable button interrupts.

### II. Results

```
fall_handler in context 0x10001904
Start timer...
Timer ends
fall_handler in context 0x10001904
Start timer...
Timer ends
rise_handler in context 0x10002168
```

1. Line 1-3:  
The button is released immediately after pressed. Since the timer is still counting down by the time the button is released, the `button_release_detecting` function is still on wait and the button IRQ has not been enabled yet. Thus, the button release IRQ is not recorded and `button_released()` is not called.
2. Line 4-7:  
The button is held for more 3 seconds and then released after pressed. After 3 seconds, `button_release_detecting()` is called and the button IRQ is enabled. Subsequently, the button release IRQ is recorded and we see from line 7 that the rise handler is called.