

In this lab, we learn interrupts, timer and OS in mbed system.

(1) Lab6 1 Interrupt

This part uses one button to trigger interrupt on its rise edge, introduce how interrupt works in mbed system.

(2) Lab6 2 Simple Timer

In this part we use timer to count the duration to print "Hello World!", and use `std::chrono` to cast time into seconds and milliseconds, besides that we know that `timer.elapsed_time().count()` return data in microseconds.

(3) Lab6 3 Multiple Timer

In this part, we see that we can manipulate multiple timer, one is 1 second in this lab and the other one is 2 second, also timer need reset to run forever or it will only run once.

(4) Lab6 4 Simple Timeout

By attaching **Timeout** to an event, it will call event once after a desired period.

(5) Lab6 5 Simple Ticker

By attaching **Ticker** to an event, it will call event repeatedly after a desired period.

(6) Lab6 6 Debounce

In this part, timer ensure that after one event happened, it needs some time for system to recall it again. This will cancel the mechanical bouncing in real button structure as we learned in Logic Design Lab.

(7) Lab6 7 Multi Thread

This method allows as to implement "parallel" system (though it's not in reality.) Two LEDs will blink and the interference between them are negligible.

(8) Lab6 8 EventQueue

This is the most important in Lab6, it use the concept of OS, if we trigger the interrupt, long duration such as `printf()` or ADC/DAC are forbidden, so we need to schedule the task into our eventqueue to wait for OS to do the job.

(9) Lab6 9 Single Thread

In this part, we combine thread and Eventqueue.

(10) Lab6 10 Multi Thread

In this part, we apply to thread in one system by using eventqueue, however we encounter some problems, Interrupt should not have printf() function. To deal with it, we divided the task by two, changing LED state in ISR and printf() in eventqueue. This will solve our problem.

(11) 6 11 OS Priority

In this part we see we can change thread's priority, which can make OS work better.

(12) Discussion

We use bare-metal method in Microprocessor lecture, but in this lecture we use RTOS to implement different task, also introducing timer, ticker and ISR for more method to implement our task.