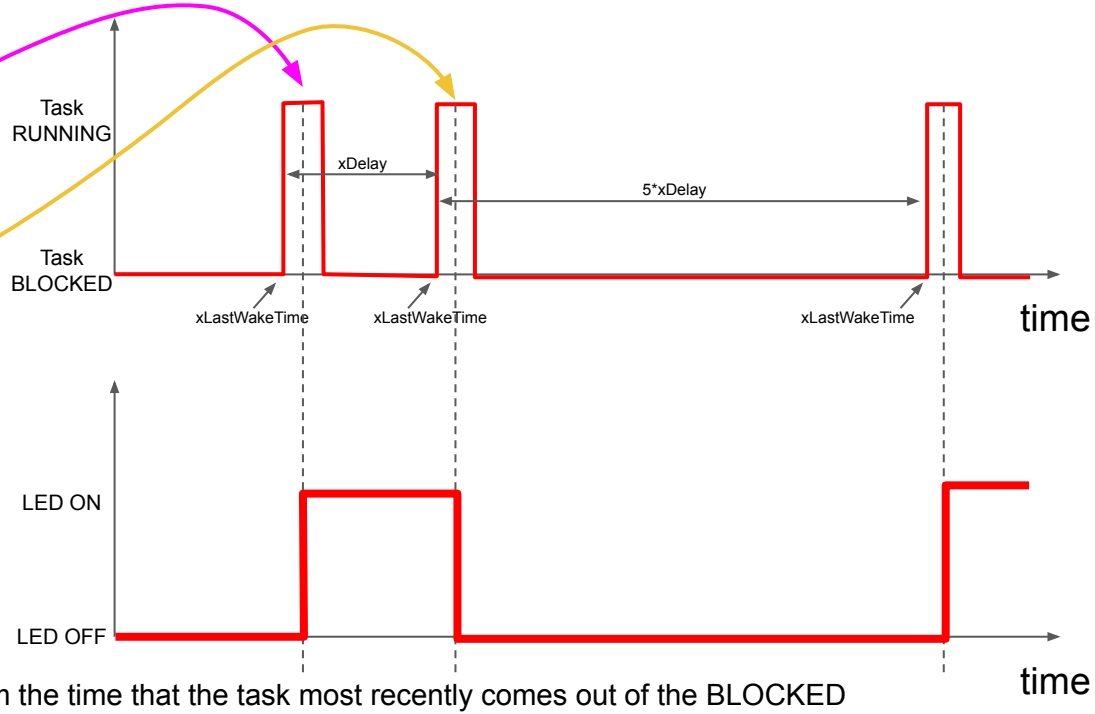
vTaskDelayUntil() explained

ese3025

how vTaskDelayUntil() works...

```
/* LED1 toggle thread */
static void vLEDTask1(void *pvParameters) {

    portTickType xLastWakeTime = xTaskGetTickCount();
    while (1) {
        Board_LED_Set(red, true);
        vTaskDelayUntil(&xLastWakeTime, xDelay);
        Board_LED_Set(red, false);
        vTaskDelayUntil(&xLastWakeTime, 5*xDelay);
    }
}
```

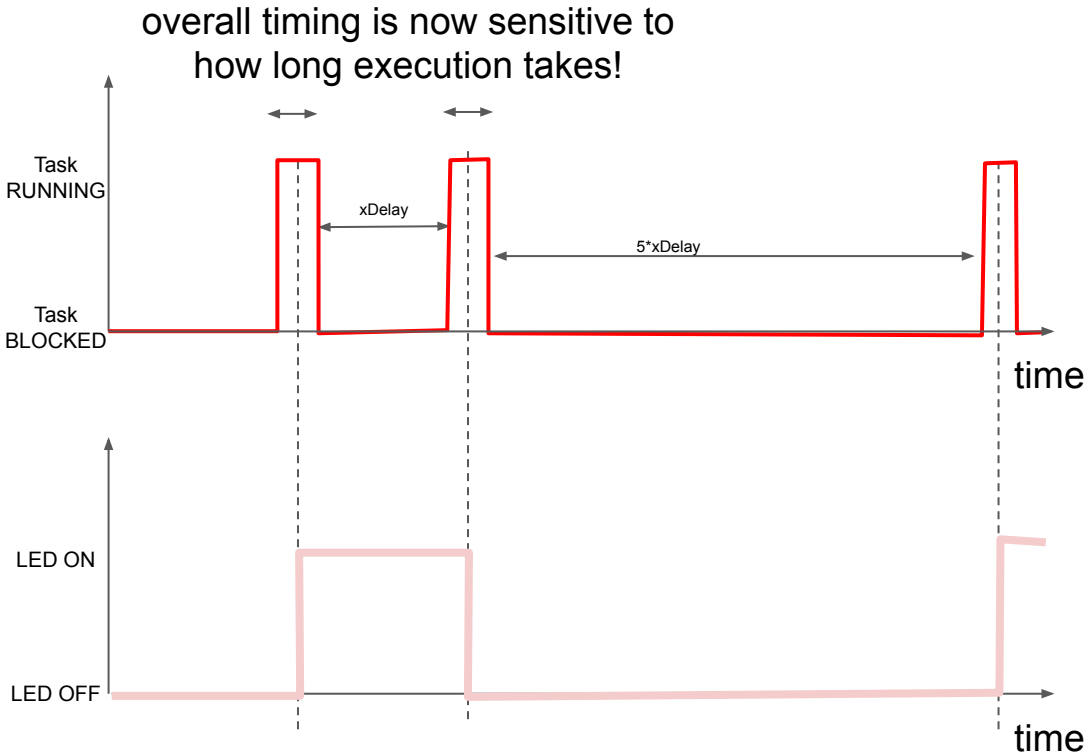


In summary, the delay is measured from the time that the task most recently comes out of the BLOCKED state; whereas vTaskDelay() simply starts the delay *from the time vTaskDelay() is called*, so the timing with vTaskDelay() is less definite overall...

how vTaskDelay() compares

```
/* LED1 toggle thread */
static void vLEDTask1(void *pvParameters) {

    while (1) {
        Board_LED_Set(red, true);
        vTaskDelay(xDelay);
        Board_LED_Set(red, false);
        vTaskDelayUntil(5*xDelay);
    }
}
```



if there are unavoidable delays in execution, for example, preemption delays caused by higher priority tasks or I/O latencies, our overall timing can be thrown off