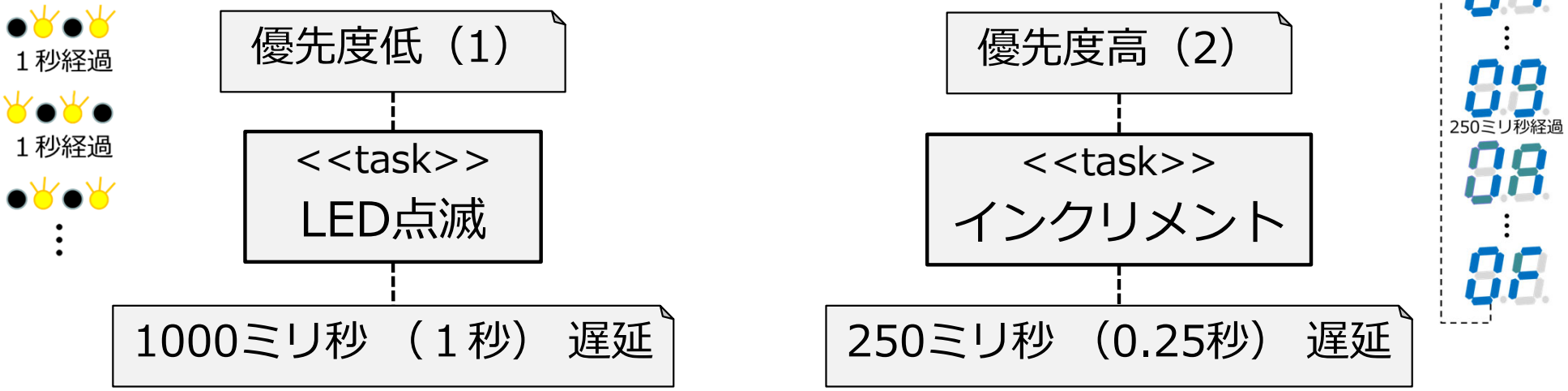
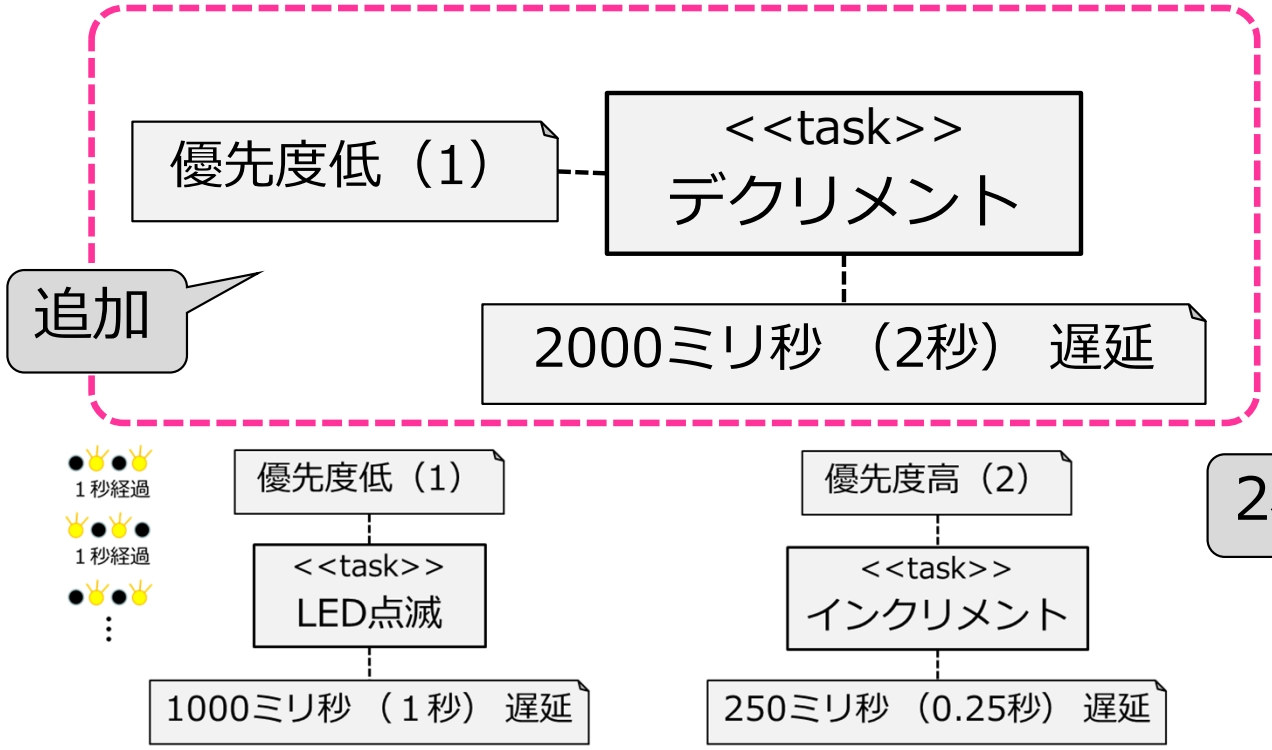
演習4　複数のタスク

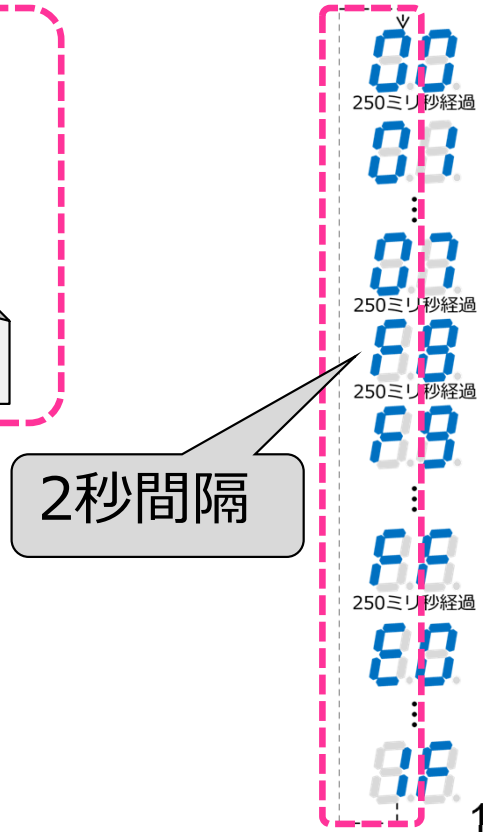
- 二つのタスクを並行動作させる
  - LED 点滅
    - 1000 ミリ秒間隔で LED を点滅する
  - インクリメント
    - 250 ミリ秒間隔で数を数え、十六進数表記で
      右の 7 セグメント LEDに表示する
      （左の 7 セグメント LED はゼロまたはデクリメント）



演習4　挑戦問題

- 更に一つタスクを生成し、並行動作させる
  - デクリメント
    - 2000 ミリ秒間隔で数を 1 ずつ減らし、十六新表記で **左の** 7 セグメント LED に表示する
      （右の 7セグメントLED はインクリメントタスクが動作）





ヒント：ファイル task.c

```
void app_main(void)
{
    BaseType_t  pass;
    // initialize devices
    initialize();
    // create tasks
    // 点滅タスクの生成（省略）
    if (pass != pdPASS) {  // エラー表示（省略）
    } else {
        pass = xTaskCreate(
                    /*** Hint:task function */&taskBlink,
                    "taskIncrement",
                    STACK_DEPTH,
                    NULL,
                    PRIORITY_INCREMENT,
                    &taskHandleIncrement
                );
        if (p     != pdPASS) {// エラー表示（省略）
        } e       // 挑戦（省略）
    return;
}
```

インクリメントタスクの生成

```
static void taskBlink(void *arg)
{
    // monitor  （省略）
    bl_init();
    for (;;) {  // closed loop
        // *** Hint:delay
        // *** Hint:blink
    }
}
```

点滅タスクのタスク関数

```
static void taskIncrement(void *arg)
{
    // monitor  （省略）
    ct_init();
    for (;;) {  // closed loop
        // *** Hint:delay
        // *** Hint:increment
    }
}
```

インクリメントタスクのタスク関数