

Lab 5 Timer 實習

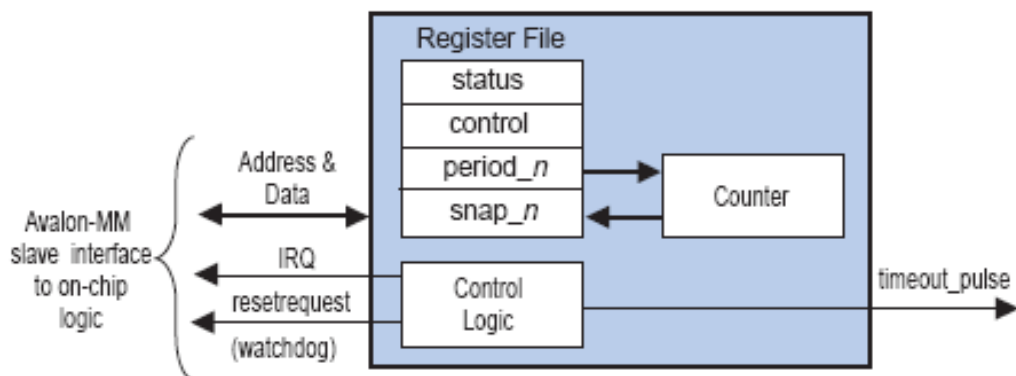
銘傳大學電腦與通訊工程學系

陳慶逸

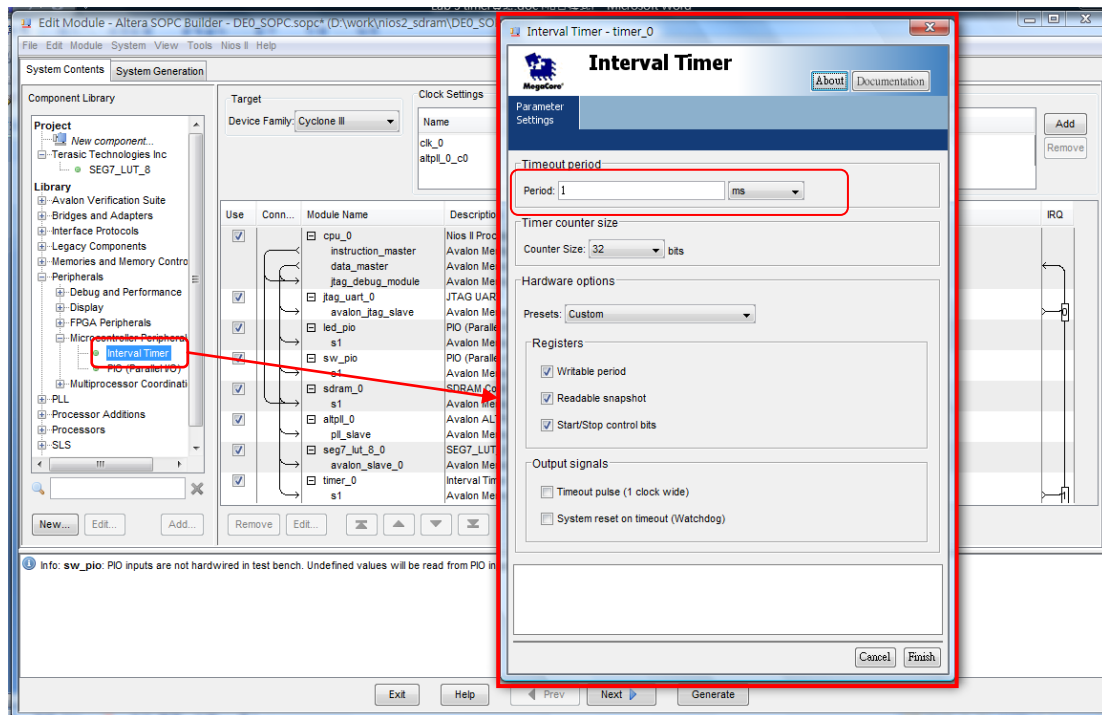
一、背景知識

Timer：顧名思義就是計時器。用來產生週期性的脈衝與作為系統監視的計時器。

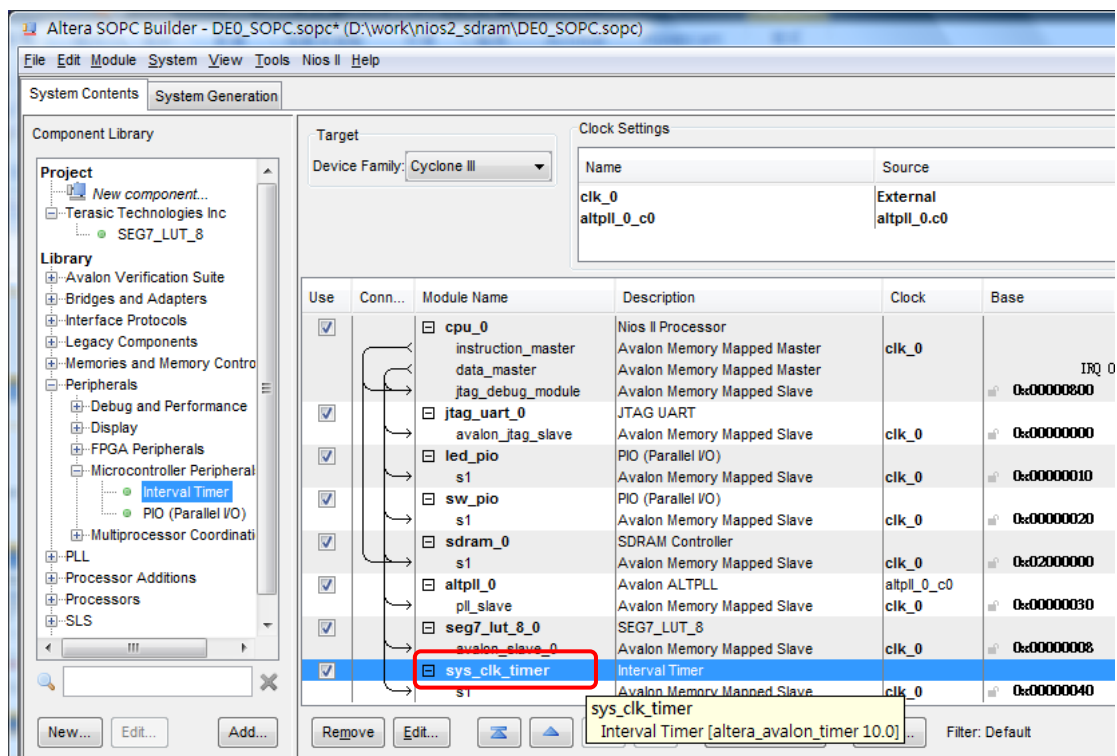
Nios Timer module 是一個32-Bit 的計時器，軟體藉由寫入暫存器要求快速且連續的讀進內部的計數值來控制計時器，以此產生一個中斷要求輸出。另一個要注意的重點是Timer 所用的是16Bits 的系統，所有的指令皆必須要是16Bits 長；而Nios CPU 是32Bits 的指令系統，所以32Bits 的指令需要被拆成兩組16Bits 的指令才能被Timer 所執行。



Interval Timer Core Block Diagram



Rename as sys_clk_timer:



計時器暫存器

<i>Table 25. Timer Register Map</i>							
A2..A0	Register Name	R/W	Description/Register Bits				
			15	...	3	2	1 0
0	status	RW					run to
1	control	RW			stop	start	cont ito
2	periodl	RW	Timeout Period – 1 (bits 15..0)				
3	periodh	RW	Timeout Period – 1 (bits 31..16)				
4	snapl ¹	RW	Timeout Counter Snapshot (bits 15..0)				
5	snaph ¹	RW	Timeout Counter Snapshot (bits 31..16)				

Notes

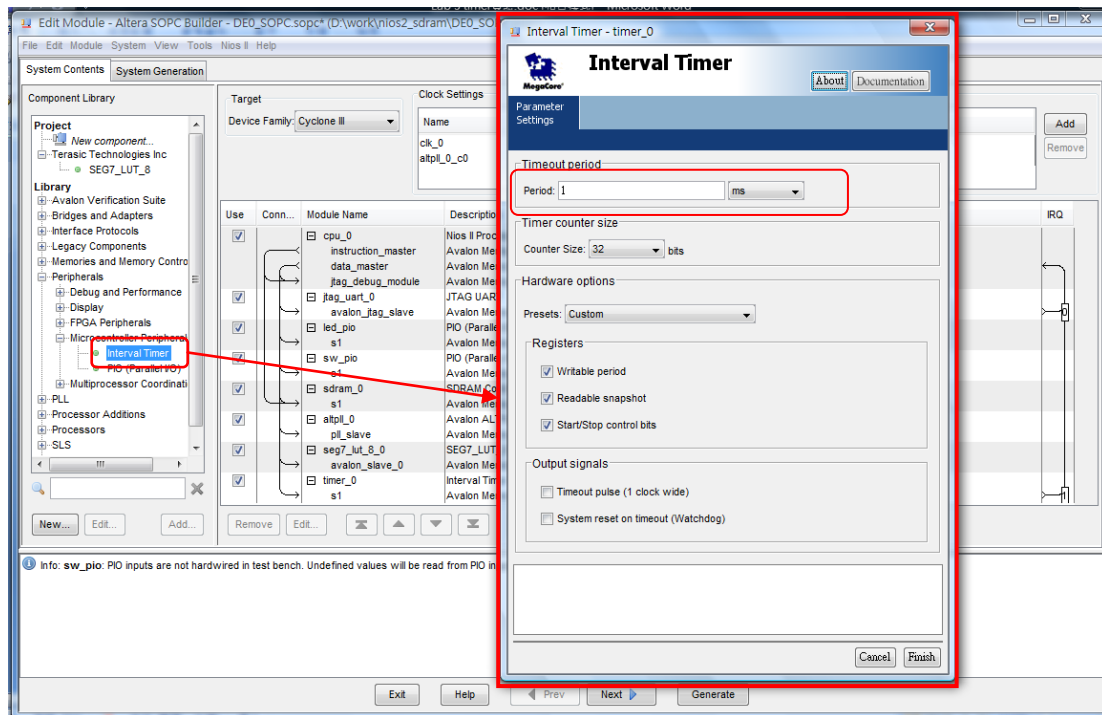
- (1) A write operation to either the `snapl` or `snaph` registers updates both registers with a coherent snapshot of the current internal counter value.

狀態暫存器

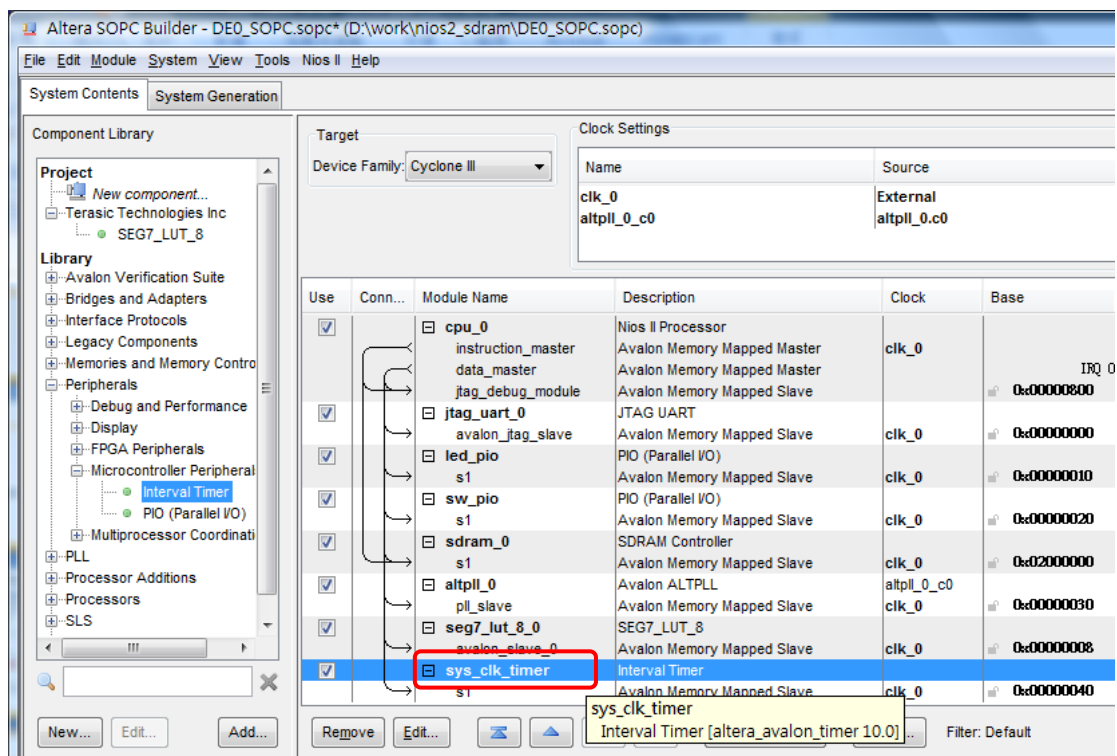
<i>Table 26. status Register Bits</i>		
Bit Number	Bit Name	Description
0	to	Timer timed out
1	run	Timer is running

二、硬體設計(Nios II Hardware Development)

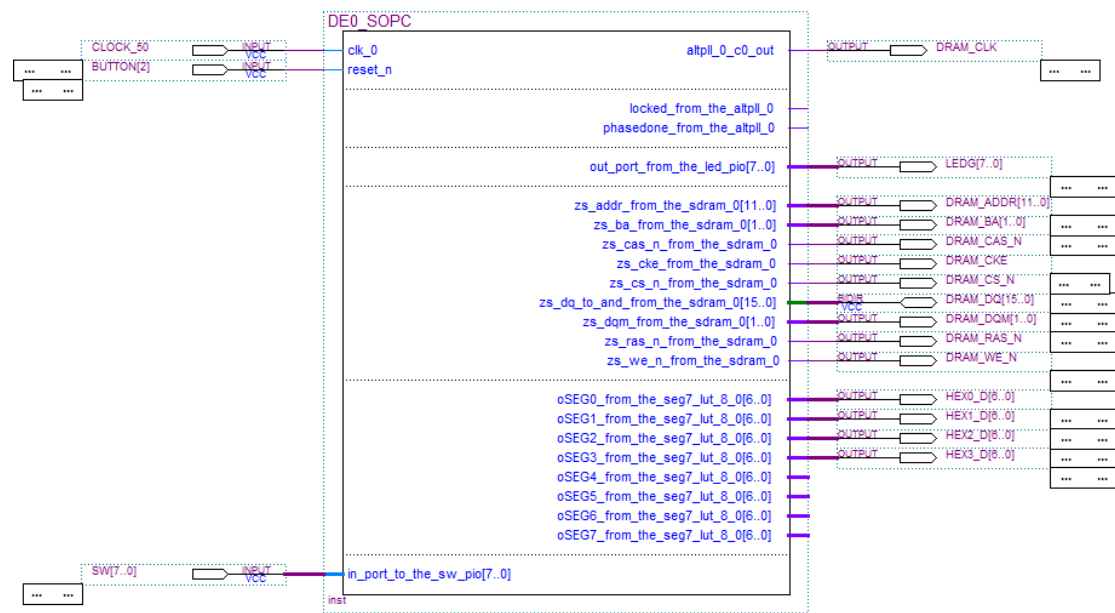
1. Add Interval Timer



Rename as sys_clk_timer:



2. Quartus II 整合設計



三、軟體設計

1. Set timer running, no looping, no interrupts (IOWR(SYS_CLK_TIMER_BASE,1,0x04);)

說明：在 system.h 中： `#define SYS_CLK_TIMER_FREQ 50000000`

```
#include <stdio.h>
#include <system.h>
#include <io.h>

int main()
{
    // Set timer for 1 second
    IOWR(SYS_CLK_TIMER_BASE,2,(short)(SYS_CLK_TIMER_FREQ & 0x0000ffff));
    IOWR(SYS_CLK_TIMER_BASE,3,(short)((SYS_CLK_TIMER_FREQ >> 16) & 0x0000ffff));
    // Set timer running, no looping, no interrupts
    IOWR(SYS_CLK_TIMER_BASE,1,0x04);

    // Poll timer forever, print once per second
    while(1)
    {
        if( IORD(SYS_CLK_TIMER_BASE,0) & 0x01)
        {IOWR(SYS_CLK_TIMER_BASE,0,0);
         printf("A second passed! \n") ; }
    }
```

```
}  
  
return 0;  
  
}
```

結果：只顯示一次 “A second passed!”

2. Set timer running, looping, no interrupts (IOWR(SYS_CLK_TIMER_BASE,1,0x06);)

```
#include <stdio.h>  
#include <system.h>  
#include <io.h>  
  
int main()  
{  
    // Set timer for 1 second  
    IOWR(SYS_CLK_TIMER_BASE,2,(short)(SYS_CLK_TIMER_FREQ & 0x0000ffff));  
    IOWR(SYS_CLK_TIMER_BASE,3,(short)((SYS_CLK_TIMER_FREQ >> 16) & 0x0000ffff));  
    // Set timer running, looping, no interrupts  
    IOWR(SYS_CLK_TIMER_BASE,1,0x06);  
  
    // Poll timer forever, print once per second  
    while(1)
```



```
{  
    if( IORD(SYS_CLK_TIMER_BASE,0) & 0x01)  
    {IOWR(SYS_CLK_TIMER_BASE,0,0);  
    printf("A second passed! \n") ; }  
}  
return 0;  
}
```

結果：每秒顯示一次 “A second passed!”

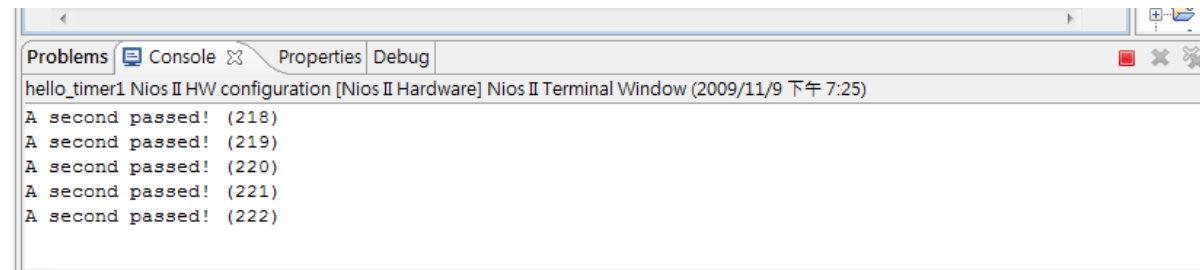
3. Set timer running, looping, no interrupts; 顯示計數秒數。

```
#include <stdio.h>  
#include <system.h>  
#include <io.h>  
  
int main()  
{  
    int t=0;  
    // Set timer for 1 second
```

```
IOWR(SYS_CLK_TIMER_BASE,2,(short)(SYS_CLK_TIMER_FREQ & 0x0000ffff));
IOWR(SYS_CLK_TIMER_BASE,3,(short)((SYS_CLK_TIMER_FREQ >> 16) & 0x0000ffff));
// Set timer running, looping, no interrupts
IOWR(SYS_CLK_TIMER_BASE,1,0x06);

// Poll timer forever, print once per second
while(1)
{
    if( IORD(SYS_CLK_TIMER_BASE,0) & 0x01)
    {
        IOWR(SYS_CLK_TIMER_BASE,0,0);    // Clear the to (timeout) bit
        printf("A second passed! (%d)\n",t);
        IOWR(LED_PIO_BASE,0,t);
        t++;
    }
}

return 0;
}
```



4. Set timer running, no looping, no interrupts; 顯示計數秒數。

```
#include <stdio.h>
#include <system.h>
#include <io.h>

int main()
{
    int t=0;

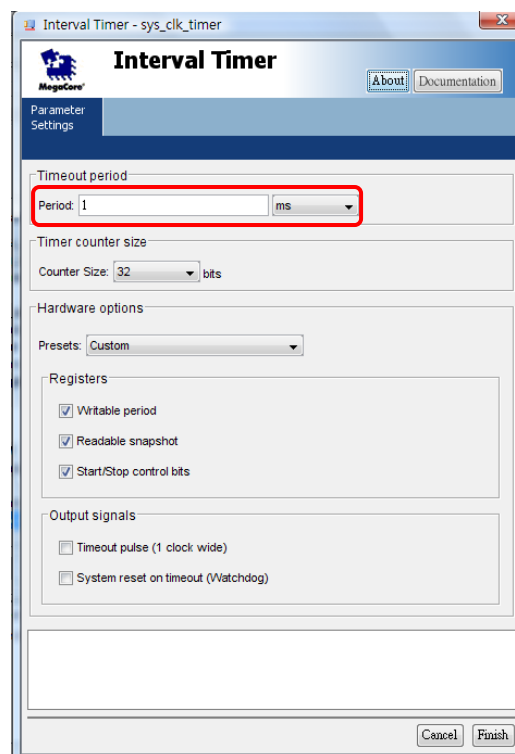
    // Set timer for 1 second
    IOWR(SYS_CLK_TIMER_BASE,2,(short)(SYS_CLK_TIMER_FREQ & 0x0000ffff));
    IOWR(SYS_CLK_TIMER_BASE,3,(short)((SYS_CLK_TIMER_FREQ >> 16) & 0x0000ffff));
    IOWR(SYS_CLK_TIMER_BASE,1,0x04); // Set timer running, no looping, no interrupts

    // Poll timer forever, print once per second
    while(1)
```

```
{  
    if( IORD(SYS_CLK_TIMER_BASE,0) & 0x01)  
    {  
        IOWR(SYS_CLK_TIMER_BASE,0,0);    // Clear the to (timeout) bit  
        IOWR(SYS_CLK_TIMER_BASE,1,0x04);  
        printf("A second passed! (%d)\n",t);  
        IOWR(LED_PIO_BASE,0,t);  
        t++;  
    }  
}  
return 0;  
}
```

5. 使用預設值

重新燒錄硬體檔，以清除之前實驗所載入的計數值。



```
#include <stdio.h>
#include <system.h>
#include <io.h>

int main()
{
    int t=0;
    IOWR(SYS_CLK_TIMER_BASE,1,0x04);

    // Poll timer forever, print once per 1ms
    while(1)
    {
        if( IORD(SYS_CLK_TIMER_BASE,0) & 0x01)
        {
            IOWR(SYS_CLK_TIMER_BASE,0,0);    // Clear the to (timeout) bit
            IOWR(SYS_CLK_TIMER_BASE,1,0x04);
            printf("1ms passed! (%d)\n",t);
            IOWR(LED_PIO_BASE,0,t);
            t++;
        }
    }

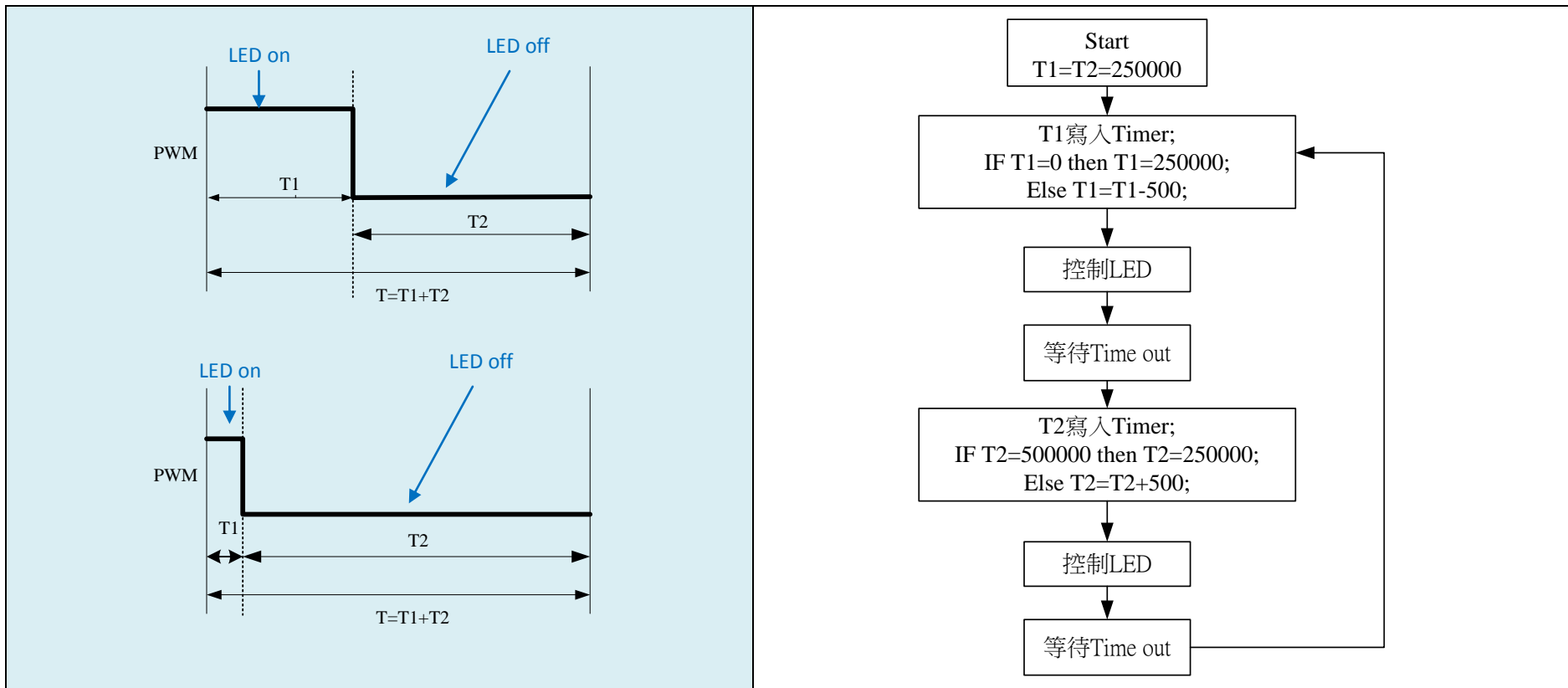
    return 0;
}
```

```
}
```

四、隨堂作業

1、PWM 功能設計：利用 Timer 製造不同寬度的正電位脈波，讓 LED 作從明亮到暗的漸近變化。

功能說明：工作週期由 50% 遞減至 0%，再重新開始。



[illegible]

```
    }  
}  
return 0;  
}
```

2、透過 SWITCH 調整改變 LED 燈的閃爍頻率

