# 实验 04\_Timer中断的使用

【实验目的】：

1、通过本实验复习LPC11C14 的32 位Timer 的使用方法

2、掌握LPC11C14 中断的使用

【实验环境】：

1、FS\_11C14 开发板

2、FS\_Colink V2.0

3、RealView MDK（Keil uVision4）

【实验步骤】：

1、在Interrupt 文件夹下找到并打开project.uvproj 文件；

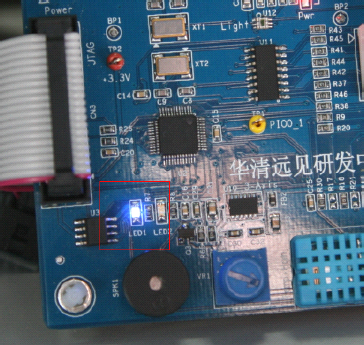
2、编译此工程；

3、通过FS\_Colink 下载编译好的工程到FS\_11C14 开发板；

4、按Reset 键复位开发板，观察LED1 现象；

【实验现象】

LED1 循环一亮一灭（如图）



【实验分析】



硬件分析：

在硬件上，我们将PIO3\_0 引脚连接到LED1 的控制引脚，如下图：

由图可以看出当PIO3\_0 输出低电平时，LED1 亮。当PIO3\_0 输出低高电平时，

LED1 灭。在程序中我们通过函数void GPIOSetValue( uint32\_t portNum, uint32\_t

bitPosi, uint32\_t bitVal )设置PIO3\_0 的输出值来控制LED 灯的亮灭。

软件分析：

中断处理函数：

void TIMER32\_0\_IRQHandler(void)

{

if ( LPC\_TMR32B0->IR & 0x01 )

{

LPC\_TMR32B0->IR = 1; /\* clear interrupt flag \*/

timer32\_0\_counter++;

}

if ( LPC\_TMR32B0->IR & (0x1<<4) )

{

LPC\_TMR32B0->IR = 0x1<<4; /\* clear interrupt flag \*/

timer32\_0\_capture++;

}

return;

}

由中断处理函数可知：当定时/计数器发生一次中断时，timer32\_0\_capture++;

会执行一次，再看主程序。

主程序：

while(1)

{

if(timer32\_0\_counter < 50)

{

GPIOSetValue(PORT3, 0, 0); //点亮LED1

}

else if((timer32\_0\_counter >= 50) && (timer32\_0\_counter < 100))

{

GPIOSetValue(PORT3, 0, 1); //熄灭LED1

}

else

{

timer32\_0\_counter = 0; //计数器清零

}

}

当timer32\_0\_capture 的值小于50 时，LED1 亮，当timer32\_0\_capture 的值

大于等于50 并且小于100 时LED1 灭，否则timer32\_0\_capture 清零。