实验 6　　外部中断实验

1. 实验目的

●掌握向量中断控制器（VIC）的设置方法。
●掌握外部中断引脚功能设置及外部中断的工作模式设置。
●掌握中断服务函数的编写。
●熟悉 LPC2000 系列 ARM7 微控制器的 GPIO 控制。

2. 实验设备

●硬件：计算机一台、EasyARM2103 开发板 一套
●软件：Windows XP 系统，ADS1.2 集成开发环境

3. 实验内容

使用 P0.16 引脚为 EINT0 功能，初始化为非向量中断，并设置为电平触发模式，然后等待外部中断。短接 JP3 的 P0.16 端口，当按键 KEY1 按下后， P0.16 输入低电平触发外部中断，中断服务程序将 LED 灯控制信号取反，取反 LED，观察 LED 现象。然后等待中断信号的撤销，最后清除中断标志并退出中断。

4. 实验预习要求

（1）仔细阅读《ARM 嵌入式系统基础教程（第 2 版）》 第 4 章 4.10 小节外部中断输入说明以及 4.9 小节中断控制器的说明。
（2）仔细阅读《EasyARM2103》手册第 4 章的内容，熟悉 GIPO 的设置。

5. 实验步骤

（1） 启动 ADS 1.2，使用 ARM Executable Image for lpc2103 工程模板建立一个工程 VICDef_C。
（2） 在工程的 user 的 main.c 中编写实验程序，然后调试。
（3） 选用 DebugInRAM 生成目标，然后编译连接工程。
（4） 将 EasyARM2103 开发板上的 JP3 的 P0.16 端口短接。
（5） 选择【Project】->【Debug】，启动 AXD 进行 JTAG 仿真调试。
（6） 在中断服务程序中设置断点，全速运行程序，观察现象。
（7） 单步/全速运行程序，观察程序是否正确运行。

6. 实验参考程序

外部中断实验的参考程序见程序清单 6。

程序清单 6　　P0.16 低电平触发外部中断 0

```
#include "config.h"
#define   LED        1 << 17        /*设置 P0.17 为 GPIO 功能                */
/*****************************************************************************
**   函数名称：Eint0IRQ
**   功能描述：外部中断 0 服务程序
**   入口参数：无          **   出口参数：无
*****************************************************************************/
void __irq   Eint0IRQ (void)
{
```

```
    if ((IO0PIN & (1 << 17)) == 0)
    {       IO0SET = 1 << 17;                /* 熄灭发光二极管                    */
     }
    else {   IO0CLR = 1 << 17;          /* 点亮发光二极管                    */
      }
    while((IO0PIN & (1 << 16)) == 0);      /* 等待按键松开                      */
    EXTINT = 0x01;                         /* 清中断标志                        */
    VICVectAddr = 0x00;                    /* 通知 VIC 中断处理结束             */
}
/*****************************************************************************
**  函数名称：main
**  功能描述：P0.16 低电平触发外部中断主函数
**  入口参数：无          **  出口参数：无
*****************************************************************************/
int main (void)
{
    PINSEL1 = PINSEL1 & (~0x03);
    PINSEL1 = PINSEL1 | 0x01;              /* 设置 P0.16 为外部中断 0 管脚      */
    PINSEL1 = PINSEL1&(~(0x03 << 2));      /*设置 P0.17 为 GPIO 功能           */
    IO0DIR  =   LED;                       /* 设置 P0.17 为输出                 */
    IO0SET  =   LED;                       /* 设置输出为高电平                  */

    IRQEnable();                           /* IRQ 中断使能                      */

    EXTMODE       = 0x00;                  /* 设置外部中断为低电平触发          */
    VICIntSelect  = 0 << 14;               /* 选择 EINT0 为 IRQ 中断            */
    VICVectCntl0  = 0x20 | 14;             /* 将外部中断 0 分配给向量中断 0     */
    VICVectAddr0  = (uint32)Eint0IRQ;      /* 设置中断服务程序地址              */
    VICIntEnable = 1 << 14;                /* 使能 EINT0 中断                   */
    EXTINT        = 0x01;                  /* 清除 EINT0 中断标志               */

    while(1);
    return 0;
}
/*****************************************************************************************
**                                End  Of  File
*****************************************************************************************/
```

7. 思考题

（1）中断服务函数为什么要使用__irq 修饰？(提示：IRQ 中断时微控制器切换到了 IRQ 模式，所以中断返回需要同时恢复 CPSR 寄存器，比如使用 “SUBS PC,R14,#4”)

（2） 在 VIC 中，如何禁止某一个中断？假设系统使用了几个 IRQ 中断，如何一次全部禁止所有中断？

☺（3）编写程序，实现按键控制流水灯，4 个 LED 灯轮流点亮。当按键按下并松开时，LED1 亮，1 秒后 LED2 亮直到 4 个灯全亮，之后全部熄灭，重新循环。当再次按下按键时停止流水灯显示，并全部熄灭。（选做题）