嵌入式系统与开发

实验报告

|  |  |
| --- | --- |
| 实验名称 | 串口按键实验 |
| 实验次数 | 实验12 |
| 学生姓名 | 蒋文涛 |
| 所在专业 | 计算机科学与技术 |
| 所在班级 | 214班 |
| 指导教师 | 李剑 |
| 地点 | 学10-509 |
| 时间 | 2024年5月 |

实验 12 按键中断控制 LED 实验

一、实验目的

1. 熟悉中断控制主程序需要进行的步骤

2. 熟悉中断服务程序需要进行的步骤

二、实验内容

实验12的示例源代码为：用按键SW5实现按键中断控制LED。按下按键SW5，4个

LED全亮，再次按下按键，4个LED全灭。

实验12的实验基础内容为：按下SW4，程序里的随机函数让四个LED灯中的一个随机亮。

实验12提升奖励内容：SW5(EINT3)~SW8（EINT9）分别对应1个LED灯，当其中一个灯亮的时

候，按下相对应的按键，则全部灯亮；如果按错，则重新随机产生一个亮灯。

5个按键，全部都在中断服务程序中实现。

三、实验原理

1．实验箱按键电路

按键使用GPIO接口，但按键本身需要外部的输入。按键硬件驱动原理图如图1所示。在下图

的1×6矩阵按键（SW4~SW9）电路中，使用6个输入（EINT2、EINT3、EINT4、EINT8EINT9 、

和EINT11）。

图 1 按键电路

按键入口对应的核心板接口电路如图2所示。

图 2 核心板按键接口电路

按键入口对应于核心板的GPH0接口，当其中一个SW按键被按下，通过查询方式就可以

检测到是哪一个接口有输入信号，从而控制相应的操作。前述的按键工作原理都是在按键

的理想状态下进行的，实际的按键动作会在短时间（几毫秒至几十毫秒）内产生信号抖动。

例如，当按键被按下时，其动作就像弹簧的若干次往复运动，将产生几个脉冲信号。一次按

键操作将会产生若干次按键中断，从而会产生抖动现象。因此驱动程序中必须要解决去除抖

动所产生的毛刺信号的问题。

2．实验箱LED电路

LED使用GPIO接口，对应于核心板的GPJ2接口，其接口电路如图3所示。当GPJ2接口

输出低电平时，LED亮；输出高电平时，LED灭。

图 3 LED 接口电路

3．寄存器说明

按键对应的GPH0相关寄存器以及LED对应的GPJ2相关寄存器功能如下图所示。

图 4 GPH0CON 寄存器功能

图 5 GPJ2CON 和 GPJ2DAT 寄存器地址

图 6 GPJ2CON 寄存器功能

图 7 GPJ2DAT 寄存器功能

四、实验步骤：

1．编写键控制 LED 代码，将代码编译为二进制文件

1）在 ubuntu 系统中，进入共享文件夹 forlinux，新建 key 文件夹。

2）进入 key 文件夹，新建启动文件 start.S，并添加启动代码（参考附录 1）。

3）新建按键控制代码文件 key.c，自行编写 c 语言代码。

4）新建 Makefile 文件，添加编译命令（参考附录 2），由以下命令编译生成二进制文件

210.bin。

$ make clean

$ make

2．安装 USB 驱动

将实验箱中的拨码开关 2 拨到 on，长按 Power 键直至电脑提示安装驱动。打开计算机

设备管理器，选择下图所示硬件安装驱动。

图 8 设备管理器界面

右键选择更新驱动程序，手动添加 USB 驱动程序路径“D:\新 509\04-常用工具\DNW”。

在图 9 所示驱动安装过程中，关闭开发板的电源，将拨码开关 2 重新置为 OFF 状态。

图 9 驱动程序安装

驱动安装完成提示如图 10 所示。

图 10 驱动安装成功

3. 使用 DNW 软件下载裸机程序至 SRAM 中运行

1）用 USB device 线连接电脑和开发板，设置开发板为 nandflash 启动(拨码开关全部拨

至 OFF 状态)。

2）在目录“D:\新 509\04-常用工具\DNW”中打开 DNW.exe。设置串口：波特率为 115200，

USB Port 为 Download，Address 为 0xd0020010。

3）在菜单栏开启 DNW 串口连接（Serial Port ->Connect）。启动开发板后立即在 DNW 窗

口迅速敲击空格键进入 Uboot 状态，可见图 11 启动界面

图 11 开发板 Uboot 启动界面

4）在 DNW 窗口中输入“dnw 0xd0020010”设置下载地址。如果 DNW 驱动安装失

败或首次使用 DNW，会提示安装驱动，请正确安装驱动，等到提示硬件可使用从进行下一

步。

5）在 DNW 菜单中，选择 usbport->Transmit->Transmit 发送生成的 key.bin 文件， DNW

自动下载 key.bin 文件至开发板。

6）在 DNW 窗口中输入“go 0xd0020010”，即可开始运行 key.bin 程序。

注意，将二进制文件下载到 SRAM 中不会破坏开发板中现有文件与程序，但是掉电后

所下载的文件将丢失。

代码：

#define GPJ2CON (\*(volatile unsigned long \*) 0xE0200280)

#define GPJ2DAT (\*(volatile unsigned long \*) 0xE0200284)

#define GPH0CON (\*(volatile unsigned long \*) 0xE0200C00)

#define GPH0DAT (\*(volatile unsigned long \*) 0xE0200C04)

#define EXT\_INT\_0\_CON \*((volatile unsigned int \*)0xE0200E00)

#define EXT\_INT\_0\_MASK \*((volatile unsigned int \*)0xE0200F00)

#define EXT\_INT\_0\_PEND \*((volatile unsigned int \*)0xE0200F40)

#define VIC0INTSELECT \*((volatile unsigned int \*)0xF200000C)

#define VIC0IRQSTATUS \*((volatile unsigned int \*)0xF2000000)

#define VIC0INTENABLE \*((volatile unsigned int \*)0xF2000010)

#define VIC0VECTADDR1 \*((volatile unsigned int \*)0xF2000110)

#define VIC0VECTADDR2 \*((volatile unsigned int \*)0xF2000108)

#define VIC0VECTADDR3 \*((volatile unsigned int \*)0xF200010C)

#define VIC0ADDRESS \*((volatile unsigned int \*)0xF2000F00)

extern void key\_isr(void);

void delay(volatile unsigned int t)

{

volatile unsigned int t2 = 0xFFFF;

while (t--)

for (; t2; t2--);

}

static long long seed = 0;

#define a 25214903917

#define b 11

#define c 281474976710656

int fun()

{

seed = (((a \* seed + b) % c) + c) % c; // 计算下一个随机数

return (int)seed % 4;

}

void led\_init(void)

{

/\* 配置GPC0\_1~GPC0\_4为输出：LED1~LED4 \*/

GPJ2CON &= ~(0xFF << 0);

GPJ2CON |= ((0x01 << 0) | (0x01 << 4) | (0x01 << 8) | (0x01 << 12));

GPJ2DAT |= (0xFF << 0);

}

void key\_init(void)

{

/\* 配置GPH0\_3为外部中断：SW4 \*/

GPH0CON |= (0xF << 4\*2);

/\* 清空相应位\*/

EXT\_INT\_0\_CON &= ~(0xFF << 0);

/\* 配置EXT\_INT[2]和EXT\_INT[3] EXT\_INT[4]为下降沿触发 0b0 010 0 010 \*/

EXT\_INT\_0\_CON |=(2 << 4\*2);

/\* 取消屏蔽外部中断 EXT\_INT[4] EXT\_INT[3] EXT\_INT[2] \*/

EXT\_INT\_0\_MASK &= ~0x4;

}

void int\_init(void)

{

/\* 选择外部中断 EXT\_INT[4] EXT\_INT[3] EXT\_INT[2]为IRQ类型的中断 \*/

VIC0INTSELECT &= ~0x4;

/\* 使能外部中断 EXT\_INT[4] EXT\_INT[3] EXT\_INT[2] \*/

VIC0INTENABLE |= 0x4;

// VI

/\* 清VIC0ADDRESS \*/

VIC0ADDRESS = 0X0;

VIC0VECTADDR2 = (int)key\_isr;

}

void key\_handle(void)

{

/\* 获取键值 \*/

/\*volatile unsigned char key\_code = EXT\_INT\_0\_PEND & 0x8; \*/

volatile unsigned char key\_code = VIC0IRQSTATUS & 0x4;

int randNum = fun();

if(key\_code == 0x4)

{

GPJ2DAT = 0xFF;

if(randNum == 0)

{

GPJ2DAT ^= 1 << 0;

delay(8000);

//GPJ2DAT = 0xFF;

}

else if(randNum == 1 )

{

GPJ2DAT ^= 1 << 1;

delay(8000);

//GPJ2DAT = 0xFF;

}

else if (randNum == 2 )

{

GPJ2DAT ^= 1 << 2;

delay(2000);

//GPJ2DAT = 0xFF;

}

else if (randNum == 3 )

{

GPJ2DAT ^= 1 << 3;

delay(8000);

//GPJ2DAT = 0xFF;

}

}

if()

delay(2000);

/\* 清中断向量寄存器 \*/

VIC0ADDRESS = 0;

/\* 清中断挂起寄存器 \*/

EXT\_INT\_0\_PEND |= 0x4;

}

int main()

{

seed = 98;

key\_init();

led\_init();

int\_init();

while (1);

return 0;

}

总结：

随机数还是伪随机