同济大学实验报告纸

软件工程 专业 22 届 1 班 2351730 姓名 刘淑仪 第 9 组 同组人员 _____ _____ _____

课程名称 嵌入式系统导论 实验名称 矩阵扫描键盘实验 实验日期 2024 年 10 月 22 日

[实验目的]

1. 掌握 Embest Edu板 - IV 实验箱扩展输入输出端口的原理和应用；
2. 掌握矩阵式键盘原理及编程方法。

[实验设备]

1. 硬件：Embest Edu板 - IV 平台，JTAG 线，串口线线，4×5 矩阵键盘，PC 机。
2. 软件：Windows 7，Hyper Terminal for Win7，μVision IDE for ARM 集成开发环境

[实验原理]

1. 矩阵键盘：矩阵键盘的行线与列线相互交叉但不直接连接，而是通过按键进行连接，在硬件设计上，列线通过电阻连接至电源，使其处于高电平状态，而行线则输出低电平。如果没有按键被按下，所有列线保持高电平；一旦按下某个按键，相应的行线与列线被接通，导致该列线的电平从高电平变为低电平。

2. 列线电平状态的获取：

标号为 U32 的芯片 74VHC541DT 连接 6 根列线 KR1 ~ KR6，并与寄存器 0x21080000 的 D1 ~ D6 位一一相连。因此，可以通过读取 8 位只读寄存器 0x21080000 来获取列线 KR1 ~ KR6 的电平状态。KR1 ~ KR6 的状态由硬件决定，初始状态均为高电平。

3. ~~寄存器~~ 行线电平状态设置：

寄存器 0x21140000 的 D2 ~ D7 位通过标号为 U34（实际为 U14 ?）的芯片

74VHC573D门控制行线KL1~KL6，通过向8位只写寄存器0x2114 0000赋值，可以调节行线KL1~KL6。通过向8位只写寄存器0x2114 0000赋值，可以调节行线KL1~KL6的电平状态。行线的电平状态由程序循环控制，当某行线被设为低电平时，如果该行线上的某个按键被按下，相关的列线将变为低电平，

【实验步骤】

1. 准备实验环境

2. 串口接收设置

3. 打开实验例程

1) 运行 μVision IDE for ARM，打开 6.1_keypad_Test 子目录下的 keypad_Test.Uv2 工程

2) 下载到SDRAM中调试运行。故在Select Target下拉框中选择 Keypad_Test IN RAM.

3) 在菜单栏"Project"选择"Build Target"或"Rebuild all target files"编译

4) 编译完毕后看到输出窗口显示"0 Error(s)"即表示编译成功，

5) 拨动实验平台电源开关，点击菜单栏 Debug → Start/Stop Debug Session 项将编译出来的映像文件下载到SDRAM中。

6) 下载完成后，点击菜单栏 Debug → Run 项运行程序

7) 全速运行后，用户可以在超级终端看到程序运行的信息，此时，用户可单击键盘上的按键，会在超级终端上显示所按下的键值。

4. 观察实验结果：

在执行到第6)步时，可以看到超级终端上输出"等待输入字符：

Please press some keys on the keypad.

此时用户可在键盘上按下任意键，会在超级终端上显示所按下的键，如：

You have pressed key <0>

You have pressed key <1>

同济大学实验报告纸

专业____届____班____姓名____第____组 同组人员____ ____ ____

课程名称____ 实验名称____ 实验日期____年____月____日

...

You have pressed key <F>

You have pressed key <*>

You have pressed key <FUN>

[实验代码]

1. 寄存器变量和按键定位变量

```
//寄存器变量和按键定位变量。
#define KPLAddr (*(volatile unsigned char*) 0x21140000)
#define KPRAddr (*(volatile unsigned char*) 0x21080000)

//volatile 是类型修饰符，用于修饰被不同线程访问和修改的变量
UNINT16TKeyNo;
UNINT8T KPRData;

Switch (KeyNo){
    case 0x0000  keychar = 'U'; break;
    case 0x0001  keychar = 'D'; break;
    case 0x0002  keychar = '_'; break;
    //其他按键同理
    ...
}
```

2. 扫描设计

```
UINT8T keyscan(void){
```

```
UINT8T i,j;

//扫描每一行
for(i=0; i<4; i++){
  KPLAddr = ~(0x4<<i);

  //检查每一类
  for(j=0; j<5; j++){
    KPRData = KPRAddr;
    if(!(KPRData & (0x2<<j))){ //为了避免在按下过程中输出多次按下
      delay(100);  /*...*/①      的信息,在按钮放开之前,对按下的
      KeyNo = (i<<8)|j;          信息进行屏蔽
      return 1;
    }
  }
}
return 0;
}
```

关键代码分析:

① …

```
if(!(KPRData & (0x2<<j))){
  delay(100);
  do{
    KPRData = KPRAddr; }                    钮被放开才退出循环
  while(!(KPRData & (0x2<<j)));  //检查KPRData的值有无改变,直到按
  KeyNo = i<<8|j;  //记录是哪个按键被按下
  return 1;        //有按钮被放下,返回1
}
```

同济大学实验报告纸

专业____届____班____姓名____第____组 同组人员____ ____ ____

课程名称____ 实验名称____ 实验日期____年____月____日

[实验小结]

在本次实验中，我深入理解了矩阵键盘的工作原理，尤其是通过行列扫描来检测按键按下的位置。在代码实现过程中，通过硬件寄存器控制行电平和读取列状态，我熟悉了如何高效地进行按键检测。同时，通过消抖处理，有效避免了由于按键抖动引起的误操作，确保了按键输入的准确性。

本实验让我对嵌入式系统中外设交互的设计与优化有了更深的体会，尤其在如何结合硬件特性和软件逻辑来实现稳定可靠的输入检测上，积累了宝贵经验。