

同济大学实验报告纸

软件工程 专业 22 届 1 班 2025170 姓名 刘淑仪 第 组 同组人员 林继申

课程名称 嵌入式系统导论 实验名称 键盘模块控制实验 实验日期 2024 年 10 月 29 日

实验目的

1. 通过实验掌握键盘控制与设计方法
2. 熟练编写 IIC 通信处理程序

实验设备

- 硬件: Embest Edukit-IV 平台, JTAG 线, 串口线, 键盘模块 (含 5x3 数字键盘和 8 位 16 段数码管), PC 机;
- 软件: Windows 7, Hyper Terminal for Win7, μ Vision IDE for ARM 集成开发环境

实验原理

1. 行列键盘的三种读键方法

· 中断式: 按键按下时产生一个外部中断通知 CPU, 中断处理程序根据地址读取数据线状态确定按键。优点是抗干扰强, 但占用一个外部中断源, 也可用查询替代, 即不断查询 /INT 引脚判断按键状态, 节省 I/O 口线, 但 IIC 总线活动增加功耗且抗干扰性较差

· 扫描法: 在键盘某一行发送低电平, 其他行为高电平, 读取列值。若某列出现低电平, 表明该行对应列的按键被按下; 否则扫描下一行。

· 反转法: 所有行扫描线先输出低电平, 读取列值, 若有低电平表明有按键按下; 接着所有列扫描线输出低电平, 再读取行值。通过组合行列值来查表确定键码。

2. IIC 总线:

IIC (I²C, Inter-Integrated Circuit) 总线是由飞利浦公司开发的两线式串行总线, 用于连接并微控制器及其外围设备, 广泛应用于微电子通信控制领域。作为同步通信的一种特殊形式, I²C 总线具有接口线少、控制简单、器



件封装小、通信速率高等优点。其两根双向信号线分别为数据线 SDA 和时钟线 SCL。

输入到 SDA 线上的每个字节必须是 8 位，每次传输的字节不受限制，但每个字节必须要有一个应答 Ack。

3. ZLG7290 寄存器说明：

- 系统寄存器 (System Reg)：地址 00H，复位值 11110000B，系统寄存器存储 ZLG7290 的系统状态，并可配置置系统运行状态

- KeyAvi (System Reg. D)：置 1 表示有效按键动作（如单击、连击、功能键状态变化），INT 引脚信号有效（低电平）；清 0 表示无按键动作，INT 引脚无效（高阻态），有效按键动作消失或读取 Key 后 KeyAvi 位自动清 0。

- 键值寄存器 (Key)：地址 01H，复位值 00H，存储被按下的按键的键值，Key=0 表示无按键按下

- 连击次数计数器 (RepeatCnt)：地址 02H，复位值 0FFH，功能键被按下时，对应位值为 0 (FunctionKey. 7 至 FunctionKey. 0 对应 564 至 557)。

- 命令缓冲区 (CmdBuf0 ~ CmdBuf1)：地址 07H~08H，复位值 00H~00H，用于传输指令

- 闪烁控制寄存器 (FlashCtrl)：地址 0CH，复位值 0111B1011B，高 4 位控制亮时间，低 4 位控制灭时间，调整值可改变闪烁频率和亮灭占空比，1 单位约为 150~250 ms，所有像素闪烁频率和占空比相同。

4. 键盘控制电路：

键盘控制电路由芯片 ZLG7290 控制，当键盘按下时，ZLG7290 系统寄存器的最低位 KeyAvi 置 1（通过 CPU 查询方式得到），CPU 通过 IIC 总线读取芯片 ZLG7290 键值寄存器 Key (01H) 中保存的键值；

[实验步骤]

1. 准备实验环境

2. 串口接收设置



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

3. 打开实验例程

1) 运行软件, 打开 b.2-keyboard-Test 子目录下的 keyboard-Test.uv2 工程。
2) 下载到 SDRAM 中调试运行: 在 Select Target 下拉框中选择 Keyboard-Test

IN RAM

3) 在 Project 中选择 "Build target" 或 "Rebuild all target files" 编译整个工程

4) 编译完成后, 若显示 "0 Error(s)" 即表示编译成功。

5) 给实验平台上电, 点击 "Debug → Start / Stop Debug Session" 将编译出来的映像文件下载到 SDRAM 中。

6) 下载完成后, 点击 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, 此时用户可以按下模块上的键盘, 可以在超级终端上显示所对应的键值。

4. 观察实验结果:

Keyboard Test Example

8-Segment Digit LED Test Example (Please look at LED)

press key 0

...

press key 9

press key A

...

press key E

[实验代码]

```
void keyboard-test(void)
```

```
{
```

```
    U8INT8T ucChar;
```




```
uart-printf ("Keyboard Test Example\n")  
keyboard-init();
```

```
while (1) {
```

```
    // 等待按键被按下
```

```
    while (1) {
```

```
        iic-read-keybd (0x70, 0x00, &ucChar);
```

```
        if (ucChar & 0x01) break;
```

```
    }
```

```
    // 读取按键数据
```

```
    iic-read-keybd (0x70, 0x01, &ucChar);
```

```
    if (ucChar != 0) {
```

```
        ucChar = key-set (ucChar); // 键值映射
```

```
    // 映射到 ASCII 码
```

```
    if (ucChar < 10)
```

```
        ucChar += '0'; // 映射字符 0~9
```

```
    else if (ucChar < 15)
```

```
        ucChar += 'A' - 10; // 映射字符 A~E
```

```
    uart
```

```
    uart-printf ("You have pressed key <%c>\n", ucChar);
```

```
}
```

```
}
```

```
}
```

```
UINT8T key-set (UINT8T ucChar) {
```

```
    switch (ucChar) {
```

```
        case 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

case 5:

ucChar -= 1;

break;

case 9:

...

case 13:

ucChar -= 4;

break;

case 17:

...

case 21:

ucChar -= 7;

break;

default: ucChar = 0;

};

return ucChar;

优化说明:

- ① 移除了不必要的重复代码
- ② 简化了 ASCII 映射
- ③ 逻辑清晰, 便于后续维护

实验小结

在本次实验中, 我们成功实现了键盘输入的检测与处理, 通过 I2C 通信从 ZLG7290 芯片读取按键信息。实验中, 我们验证了键盘的功能, 能够准确



识别并显示用户按下的键，并对其进行了ASCII码转换，确保输入的字符能够正确显示。

通过优化代码结构，我们提高了程序的运行效率，减少了不必要的重复操作，增加了代码的可读性。这一过程不仅加深了我们对键盘模块工作原理的理解，也为未来嵌入式系统开发提供基础。

