

# 同济大学实验报告纸

软件工程 专业 22 届 1 班 学号 姓名 刘敬仪 第 组 同组人员 林健申  
课程名称 嵌入式系统导论 实验名称 数码管显示实验 实验日期 2024 年 11 月 日

## 实验目的

1. 通过实验掌握 LED 的显示控制方法
2. 通过实验加深对 IIC 总线工作原理的掌握

## 实验设备

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

## 实验原理

1. 八段数码管由八个发光二极管组成, 其中七个长条形的发光管排列成“日”字形, 右下角一个 L 形的发光管作为显示小数点用。八段数码管能显示所有数字及部分英文字母

### 2. 八段数码管的基本工作原理:

以共阳极八段管为例, 当某一段发光二极管的信号为低电平时, 该段二极管会点亮。要显示某个特定字符, 需要点亮该字符对应的所有发光二极管。与此相对, 共阴极数码管的工作原理是, 当控制信号为高电平时, 相应的发光二极管点亮。

每个字符的显示由特定的电平组合来控制, 这些信号按 a, b, c, d, e, f, g 和 dp 的顺序排列形成一个特定的段码。常用字符及其对应的段码见后表。

### 3. 八段数码管的显示方式





### 10 静态显示:

在静态显示模式下,当数码管显示一个字符时,该字符对应段的发光二极管控制信号始终保持有效状态,这种方式下,字符持续亮起,显示效果稳定,不依赖外部控制周期。

### 11 动态显示:

在动态显示模式下,数码管显示某个字符时,各段的发光二极管是按一定周期轮流点亮的,每段的点亮时间非常短(约1ms)。由于视觉暂留现象和发光二极管的余辉效应,虽然实际每段是间歇点亮的,但在观察者眼中,字符依然呈现出稳定的显示效果。

## 3. ZLG7290 寄存器说明

### 11 扫描位寄存器 (Scan/Num)

• 地址: 0DH

• 复位值: 7

此寄存器用于控制最大扫描显示的位数,取值范围为18位的显示。通过减少扫描位数,可以提高每个位的扫描时间占比,从而提升LED亮度。未参与扫描的显示缓存寄存器内容将保持不变,如,当Scan/Num为3时,仅显示DpRam 0至DpRam 3的内容。

### 12 显示缓存寄存器 (DpRam 0~DpRam 7)

• 地址: 10H~17H

• 复位值: 00H (每个寄存器复位值均为00H)

这些寄存器用于存储显示内容,其中位置1表示对应像素点亮。DpRam 7至DpRam 0的显示内容分别对应Dig 7至Dig 0引脚。

## 4. 软件程序设计:

### 11 字符段码的存放:

八段数码管上显示的字符段码存储在数组f-segDigit[17]中,通过IIC总线依次将他们写入到ZLG7290芯片的显示缓存寄存器(DpRam 0~DpRam 7)中即可完成字符在八段数码管上的显示。

码





# 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

unsigned char f-szDigital[10]

= {0xFC, 0x6D, 0xDA, 0xF2, 0x66, 0xB6, 0xBE, 0xED, 0xFE, 0xF6};

2). 函数 `zic-init-8led()`

· 中断方式处理LED段码传输:

由于LED的段码通过IIC总线传输, 因此需要使用中断方式来检测每个字节的传输. 为此, 必须定义中断处理程序的入口并启用中断. 具体实现如下:

`rINTMSK &= ~BIT-IIC;`

`PIR-IIC = (unsigned) zic-int-8led;`

· IIC 接口初始化:

IIC 接口的初始化包括设置相关寄存器的初始值. 具体配置如下:

`MUCADD = 0x10;` // 设置S3C2410X的从设备地址

`MUCCON = 0xef;` // 启用Ack和IIC总线中断, 设置IIC时钟, 并清除pending<sup>位</sup>

`MUCSTAT = 0x10;` // 启用发送/接收中断

3). 函数 `zic-write-8led()` 函数:

· 函数原型: `void zic-write-8led (UINT32T unSlaveAddr, UINT32T unAddr, UINT8T ucData)`

· 参数说明: `unSlaveAddr` - 输入, IIC从设备地址

`unAddr` - 输入, 数据地址

`ucData` - 输入, 表示写入的数据值

· 函数返回值: 无 (NULL)

· 功能描述: 调用 `zic-write-8led (0x70, 0x10 + i, f-szDigital[i])` 可实现以下功能: 将数组 `f-szDigital` 中第 `k+1` 个元素的段码数据写入到第 `i+1` 个8段数码管进行显示.





## [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 6.3-8LED-Test 子目录下的 8LED-Test.UV2-工程

2) 在 Select Target 下拉框选择 8LED-Test-IN-RAM

3) 编译整个工程, 显示 "0 Errors" 则编译成功

4) 拨动实验平台电源开关, 给实验平台上电

5) 全速运行

6) 全速运行后, 用户可看到 0~9 这 10 个数字在 8 段数码管上移动显示

4. 观察实验结果

## [实验代码]

```
void display_number_sequence (int iterations, int delay_time_ms) {  
    iic-init-8leds();  
    for (int m=0; m<iterations; m++) {  
        for (int j=0; j<10; j++) {  
            update-8led-display(j);  
            delay(delay_time_ms);  
        }  
    }  
    clear-8led-display();  
}  
  
void update-8led-display (int offset) {  
    for (int i=0; i<8; i++) {  
        int digit = 9 - (i+offset)%10;  
        iic-write-8led(0x70, 0x10+i, f-8-Digital(digit));  
    }  
}
```





# 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

```
void clear_8led_display() {  
    for (int i=0; i<8; i++) {  
        iic_write_8led(0x70, 0x10+i, 0x00);  
    }  
}
```

优化点:

1. 函数封装: 将显示逻辑封装为 `update_8led_display()`, 独立函数 `数字计数与显示逻辑`; 将清除显示封装为 `clear_8led_display()`, 简化代码。
2. 参数化设计: 主函数 `display-number-sequence()` 使用参数控制循环次数和延迟时间, 增加灵活性。
3. 代码简化: 使用了更为清晰的变量名称。

## [实验小结]

本实验通过八段数码管和IIC总线的结合, 实现了对多位数码管的动态显示控制。八段数码管通过段码控制每段的亮灭以显示字符, ZLG7290芯片作为驱动器, 通过IIC总线接收段码数据并分配给各数码管。本实验中, 我们利用IIC通信的可靠性, 完成了主设备与ZLG7290之间的数据传输, 并通过动态扫描的方式实现数字的滚动显示, 同时通过适当的延时确保显示效果稳定清晰。

实验进一步加深了对IIC通信协议和数码管工作原理的理解, 特别是在中断机制的应用中提高了通信效率和系统实时性。此外, 通过代码模块化设计和功能封装, 我优化了程序的可读性和复用性。

