**第七章 数码管实验**

**一、实验内容**

基于远程云端硬件实验平台编程实现：数码管显示控制。

学生在远程云端硬件实验平台上，利用平台现有资源，设计一个数码管显示控制功能电路

**二、实验目的**

了解KEIL软件和远程云端实验界面的操作方法。

学习用KEIL软件编写程序和使用远程云端绘制电路以及下载程序的操作。

掌握单片机I/O口的应用。

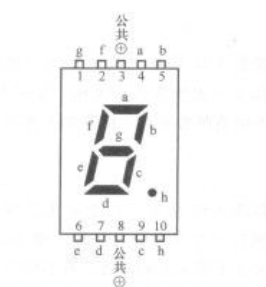
了解发光数码管的使用。

**三、实验原理**

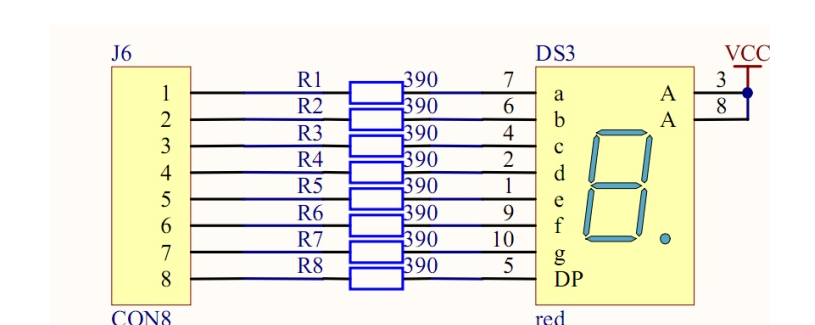
实验器件简介

数码管也称作辉光管，是一种可以显示数字和其他信息的[电子](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)设备。[玻璃](https://baike.baidu.com/item/%E7%8E%BB%E7%92%83" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)管中包括一个金属丝网制成的[阳极](https://baike.baidu.com/item/%E9%98%B3%E6%9E%81" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)和多个[阴极](https://baike.baidu.com/item/%E9%98%B4%E6%9E%81" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)。大部分数码管阴极的形状为[数字](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)。管中充以低压气体，通常大部分为[氖](https://baike.baidu.com/item/%E6%B0%96" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)加上一些[汞](https://baike.baidu.com/item/%E6%B1%9E" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)和/或[氩](https://baike.baidu.com/item/%E6%B0%A9" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)。给某一个阴极充电，数码管就会发出颜色光，视乎管内的气体而定，一般都是橙色或绿色。

按发光二极管单元连接方式可分为共阳极数码管和[共阴极数码管](https://baike.baidu.com/item/%E5%85%B1%E9%98%B4%E6%9E%81%E6%95%B0%E7%A0%81%E7%AE%A1/18884048" \t "https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%AE%A1/_blank)。共阳数码管是指将所有发光二极管的阳极接到一起形成公共阳极(COM)的数码管，共阳数码管在应用时应将公共极COM接到+5V，当某一字段发光二极管的阴极为低电平时，相应字段就点亮，当某一字段的阴极为高电平时，相应字段就不亮。共阴数码管是指将所有发光二极管的阴极接到一起形成公共阴极(COM)的数码管，共阴数码管在应用时应将公共极COM接到地线GND上，当某一字段发光二极管的阳极为高电平时，相应字段就点亮，当某一字段的阳极为低电平时，相应字段就不亮



实物参考电路分析



实验所用模块配置与分析

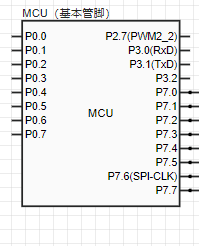
# 数码管

本数码管器件为有源模块，数码管的亮段当成7个发光二极管。根据内部发光二极管的共链接端不同，可以分为共阴极接法和共阳极接法，共阳极接法是七个发光二极管的正极共同接电源VCC，通过控制MCUgpio口的输出状态决定每个发光二极管的负极是否接地来显示字母。共阴极接法是每个发光二极管的负极共同接地GND，通过控制MCUgpio，的输出状态决定每个发光二极管的正极对否接电源来显示字母。4位数码管控制原理和实物类多位数码管相同，通过控制（a-dp）8个管脚高低电平控制单个数码管显示，通过1-4号管脚控制4位数码管显示。



二、MCU

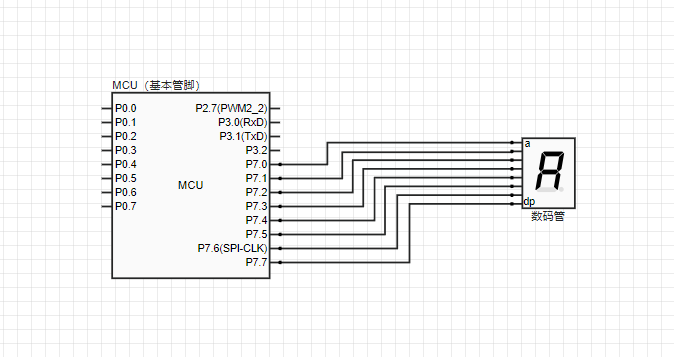
基本管脚（MCU）、自定义管脚（MCU）均为逻辑器件，其中基本管脚包含时钟复位端，8位单比特输入信号（IN0-IN7）、8位单比特输出信号（OUT0-OUT7），自定义管脚的逻辑器件模型则可以通过页面选择控制时钟复位端口是否显示，根据设计需求更改输入输出数据线位宽等操作。



GPIO端口对应

|  |
| --- |
| **数码管** |
| **P7** |

**四、实验参考仿真电路**



**五、实验步骤**

1.打开编程软件。

2. 点击“Project”,选择：“Open Project”调用对应工程下的工程文件。

3. 双击工程界面的“main.C”文件，查看主程序，点左上角Translate图标进行程序编译。

4. 打开远程云端中实验面板按照仿真参考电路进行绘制，

5.绘制完成将KEIL生成的Hex文件烧录到电路，观察实验现象。

**六、参考代码与分析**

**数码管端口配置初始化**

void ONELED\_GPIO\_config(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure; //结构定义

GPIO\_InitStructure.Mode = GPIO\_OUT\_PP; //指定IO的输入或输出方式,GPIO\_PullUp,GPIO\_HighZ,GPIO\_OUT\_OD,GPIO\_OUT\_PP

GPIO\_InitStructure.Pin = GPIO\_Pin\_All; //指定要初始化的IO, GPIO\_Pin\_7, 或操作

GPIO\_Inilize(GPIO\_P7, &GPIO\_InitStructure); //初始化

}

code u8 one\_mum[16]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};

**主函数：**

Special\_IO\_Init(); //特殊端口初始化函数

WBUZZER\_GPIO\_config();

ONELED\_GPIO\_config();

while(1) //循环

{

for(i=0;i<16;i++)

{

P7=one\_mum[i];

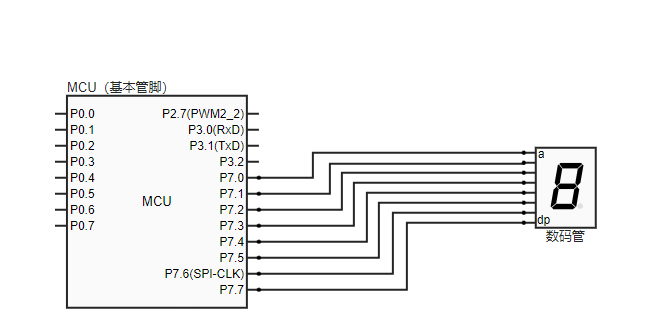
delay\_ms(500);

}

}

**七、实验结果**

运行实验后数码管循环显示0-9，A-F。



**八、拓展思考**

 **使用矩阵按键控制显示内容**