



合肥工业大学

计算机与信息学院

实验报告

课 程： 单片机原理与应用

专业班级： 物联网工程 19-2 班

学 号： 2019217819

姓 名： 付炎平

老 师： 欧 阳 一 鸣

2022 年 5 月 22 日

目录

一、 实验任务	2
二、 实验原理	2
● 数码管	2
● 数码管工作原理	2
● 段码表	2
● 74LS138	2
● 74HC164	3
显示部分:	3
中断部分:	3
三、 实验分析及思路	3
1. 38 译码器	4
2. 74HC164	5
3. 数码管显示	6
● 单个数码管显示	6
● 多数码管同时显示	6
● 串并转换时数码管显示干扰的消除	7
四、 实验结果截图	7
附录: 实验源码	9
延时函数	9
数码管	10
主函数	12

一、实验任务

- (1) 使用数码管显示学号后 8 位。
- (2) 使用串转并电路控制数码管的段码。

二、实验原理

● 数码管

数码管结构：数码管由 8 个发光二极管(以下简称字段)构成，通过不同的组合可用来显示数字 0~9，字符 A~F、H、L、P、R、U、Y 等符号及小数点“.”。数码管又分为共阴极和共阳极两种类型。

● 数码管工作原理

共阳极数码管中 8 个发光二极管的阳极(二极管正端)连接一起,即为共阳极接法,简称共阳数码管。通常,公共阳极接高电平(一般接电源),其它管脚接段驱动电路输出端。当某段驱动电路的输入端为低电平时,该端所连接的字段导通并点亮。根据发光字段的不同组合可显示出各种数字或字符。此时,要求段驱动电路能吸收额定的段导通电流,还需根据外接电源及额定段导通电流来确定相应的限流电阻。共阴极数码管中 8 个发光二极管的阴极(二极管负端)连接在一起,即为共阴极接法,简称共阴数码管。通常,共阴极接低电平(一般接地),其它管脚接段驱动电路输出端。当某段驱动电路的输出端为高电平时,该端所连接的字符导通并点亮,根据发光字段的不同组合可显示出各种数字或字符。同样,要求段驱动电路能提供额定的段导通电流,还需根据外接电源及额定段导通电流来确定相应的限流电阻。

● 段码表

单片机 LED 共阳极段码表[0-F]

DB 0c0H, 0f9H, 0a4H, 0b0H, 99H, 92H, 82H, 0f8H [0-7]

DB 80H, 90H, 88H, 83H, 0c6H, 0a1H, 86H, 8eH [8-F]

单片机 LED 共阴极段码表[0-F]

DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H [0-7]

DB 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 76H [8-F]

为了满足某些特殊用户需要有时也会用到以下特别字符的段码表单片机 LED 共阳极段码表[HLPRUY-. 熄灭]

DB 89H, 0C7H, 8CH, 0CEH, 0C1H, 91H, 0BFH, 7FH, 0FFH[HLPRUY-. 熄灭]

单片机 LED 共阴极段码表[HLPRUY-. 熄灭]

DB 76H, 38H, 73H, 31H, 3EH, 6EH, 40H, 80H, 00H [HLPRUY-. 熄灭]

● 74LS138

74LS138 为 3 线—8 线译码器,共有 54/74S138 和 54/74LS138 两种线路结构型式,其 74LS138 工作原理如下: 当一个选通端 (G1) 为高电平,另两个选通端 (G2A 和 G2B) 为低电平时,可将地址端 (A、B、C) 的二进制编码在一个对应的输出端以低电平译出。下图是它的原理结构图以及真值表:

74HC138										
A	B	C	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
H	H	H	H	H	H	H	H	H	H	L
H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	H	H	L	H	H	H
L	H	H	H	H	H	L	H	H	H	H
L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	H	H

74LS138 真值表

● 74HC164

74HC164 是高速硅门 CMOS 器件,与低功耗肖特基型 TTL (LSTTL) 器件的引脚兼容。为 8 位边沿触发式移位寄存器,串行输入数据,然后并行输出。数据通过两个输入端 (DSA 或 DSB) 之一串行输入;任一输入端可以用作高电平使能端,控制另一输入端的数据输入。两个输入端或者连接在一起,或者把不用的输入端接高电平,一定不要悬空。

74hc164 工作原理即等同于移位寄存器:

F0、F1、F2、F3 是四个边沿触发的 D 触发器,每个触发器的输出端 Q 接到右边一个触发器的输入端 D。因为从时钟信号 CP 的上升沿加到触发器上开始到输出端新状态稳定地建立起来有一段延迟时间,所以当时钟信号同时加到四个触发器上时,每个触发器接收的都是左边一个触发器中原来的数据 (F0 接收的输入数据 D1)。寄存器中的数据依次右移一位。

显示部分:

将需要显示的数值送入 A->查表求得显示段码->将段码逐位移入 164->8 位移完后点亮数码管->延时->返回第一步执行

中断部分:

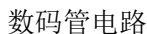
进入中断->保存现场->重置 TCC->够 1 秒钟将需要显示的数据+1,并重置,不够就退出;

三、实验分析及思路

实验要求使用数码管显示学号的后八位。多个数码管的显示实际上是利用人眼的视觉暂留现象在短时间循环控制每个数码管的显示,从而实现多个数码管的同时显示,即循环扫描。

数码管我选择了共阴极的控制方式，段选的驱动电路选择了 38 译码器，段码的驱动电路使用了实验要求的串转并电路。根据实验箱的电路原理图，进行了如下的实验连线。

数码管的段选使用的 38 译码器。



38 译码器的 A、B、C 分别连接 MCU 的 P1.3, P1.4, P1.5, 即 G2 区的 JP32 的 1, 2, 3 排针连接 A3 区的 P0.3~P0.5 的对应排针。JP25 的跳线帽接下, 为 32 译码器供电。JP35 连接 F4 区的 JP41, 将 32 译码器的输出作为段选。

```
sbit A1 = P1^3;
sbit B1 = P1^4;
sbit C1 = P1^5;
```

38 译码器的引脚连接

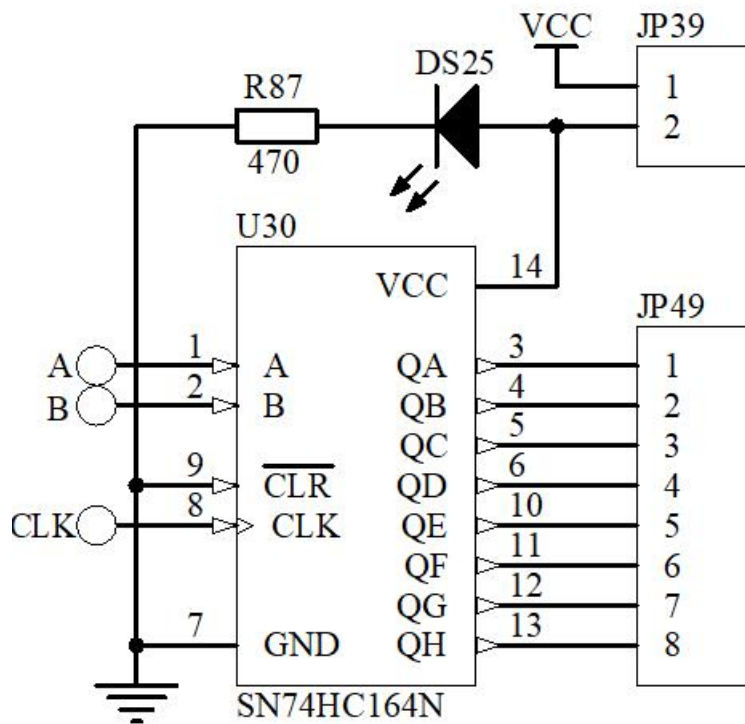
通过 CBA 对应的二进制的值我们可以实现 Y0~Y7 为地电平的选择，从而控制共阴极的段选。

```
A1 = (i >> 0) & 0x01;  
B1 = (i >> 1) & 0x01;  
C1 = (i >> 2) & 0x01;
```

段选的代码实现

2. 74HC164

74HC164N 本质上一个上跳沿触发的移位寄存器，通过它，我们可以实现串行输入的数据转换为并行的输出。



74HC164

A、B 接口分别连接 MCU 的 P1.0, P1.1, A, B 是数据的输入端，输入数据到移位寄存器的数据是根据二者相与的结果得到的，所以在代码编写的时候将 B 一直置为 1，A 输入的数据即为移位寄存器的输入值。

CLK 输入的是移位控制信号，在每个 CLK 输入的上跳将触发移位寄存器向右移动移位，从将输入的值不断移位，使得 QA~QH 填满输入的值。所以，我们将 clk 与 P1.2 连接，通过控制 P1.2 的高低电平的变化，实现 A 的输入值填入移位寄存器。

```
sbit inA = P1^0;
sbit inB = P1^1;
sbit clk = P1^2;
```

A、B、clk 的引脚连接

```
void series_to_parallel(uchar dat)
{
    int i = 0;
    clk = 0;
    for(i = 0; i < 8; i++)
    {
        inA = (dat >>(7 - i)) & 0x01;
        clk = 1;
        clk = 0;
    }
}
```

串转并的代码实现

3. 数码管显示

七段式数码管段码从低位到高位分辨对应某一数码管的 a~g 以及数码管右下方的小数点的两面，我们使用的是共阴极，段码对应的位是高电平时，该段码管就会亮。

● 单个数码管显示

某一个数码管的显示的逻辑如下，首先我们将段码通过串转并电路置为全灭的转态，即向 A 填入 0x00。接着我们通过 38 译码器的 A、B、C 选择码段，选择某一数码管后对应的 Y 会输出低电平，使得该数码管的共极为低电平，紧接着我们就可以通过串转并电路填入数码管需要显示的数值的段码。完成段码的填入后我们还需要一个延时函数，这是为了保证这个数码管的显示时间，确保循环扫描数码管时的视觉暂留效果。

```
27
28 void digitron_display(uchar dat, uchar i)
29 {
30     series_to_parallel(0x00);
31     g2 = 1;
32     A1 = (i >> 0) & 0x01;
33     B1 = (i >> 1) & 0x01;
34     C1 = (i >> 2) & 0x01;
35     g2 = 0;
36     series_to_parallel(digitron_data[dat]);
37     delay_ms(2);
38 }
```

点亮一个数码管的代码实现

● 多数码管同时显示

完成了一个数码管后，我们对于每一个数码管采用循环扫描的方式进行点亮，即可实现同时点亮多个数码管。

```

void digitron_dis(char* dat, uchar num)
{
    int i;
    for(i = 0; i < num; i++)
    {
        digitron_display(dat[i], num - 1 - i);
    }
}

```

同时点亮多个数码管的代码实现

● 串并转换时数码管显示干扰的消除

这里需要注意的是，串转并电路将串行数据通过移位寄存器转换为并行数据需要时间，在移位的过程中，可能会出现显示干扰的情况，为了消除这个干扰，我们在进行串并转换的时候，需要先将 38 译码器的 G2A 和 G2B 置为高电平，此时 38 译码器所有的输出都是高电平，由于我们使用的是共阴极数码管，所以相当于没有选择任何数码管。此时，我们进行段码的串转并，就不会存在显示干扰的情况了。

```

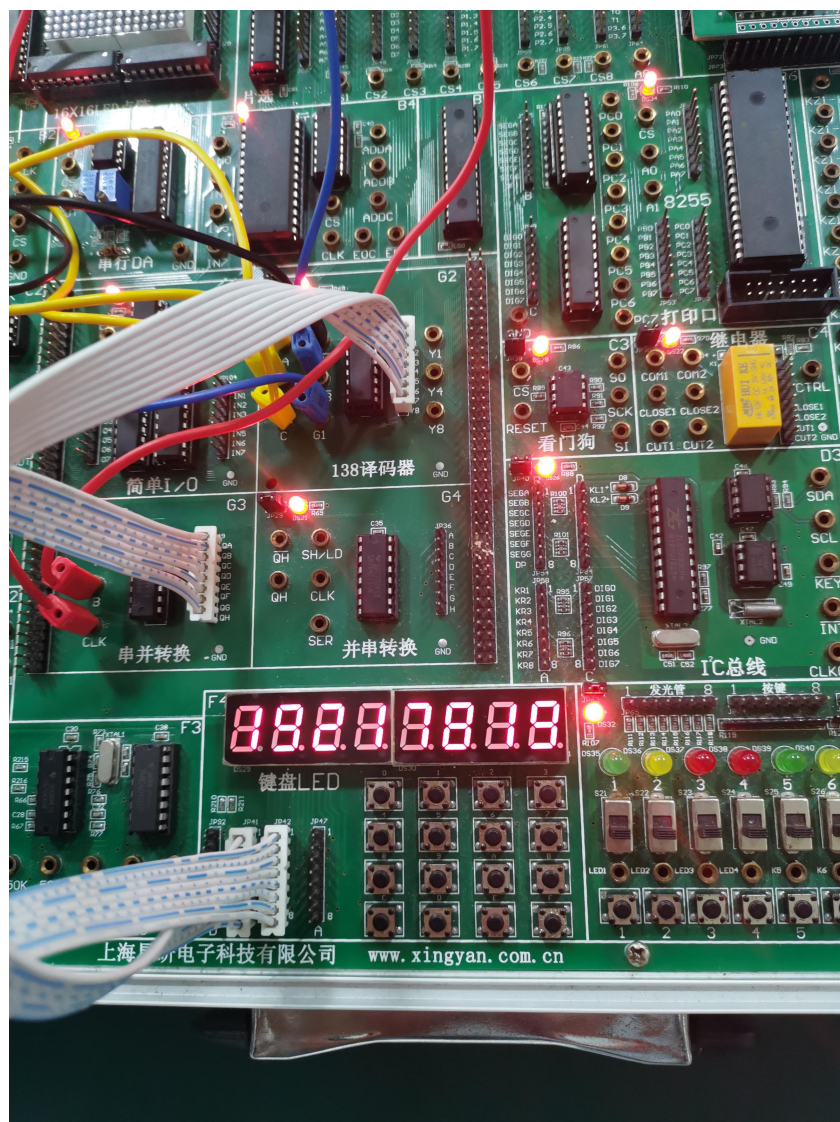
27
28 void digitron_display(uchar dat, uchar i)
29 {
30     series_to_parallel(0x00);
31     g2 = 1;
32     A1 = (i >> 0) & 0x01;
33     B1 = (i >> 1) & 0x01;
34     C1 = (i >> 2) & 0x01;
35     g2 = 0;
36     series_to_parallel(digitron_data[dat]);
37     delay_ms(2);
38 }

```

消除串并转换过程中的数码管显示干扰

四、实验结果截图

显示学号 “2019217819”



学号的显示

附录：实验源码

延时函数

```
#ifndef _DELAY_H_
#define _DELAY_H_

#include "reg52.h"
#include "main.h"
#include "intrins.h"

void delay_ms(uchar ms);
void delay_us(uchar us);
void delay_5us(void);

#endif

#include "delay.h"

void delay_ms(uchar ms)
{
    uchar i, j;
    while(ms--)
    {
        i = 2;
        j = 239;
        do
        {
            while(--j);
        }while(--i);
    }
}

void delay_us(uchar us)
{
    while(us--);
}
```

```

void delay_5us(void)
{
    _nop_();
}

```

数码管

```

#ifndef _DIGITRON_H_
#define _DIGITRON_H_

#include "reg52.h"
#include "main.h"
#include "delay.h"

sbit inA = P1^0;
sbit inB = P1^1;
sbit clk = P1^2;

sbit A1 = P1^3;
sbit B1 = P1^4;
sbit C1 = P1^5;
sbit g2 = P1^6;

void series_to_parallel(uchar dat);
void digitron_init(void);
void digitron_display(uchar dat, uchar i);
void digitron_dis(char* dat, uchar num);

#endif

//共阴极, 采用共阴极的驱动方式
//uchar digitron_data[] =
{0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F, 0x77, 0x7C, 0x39, 0x5
E, 0x79, 0x71, 0x00};
uchar digitron_data[] =
{0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};

```

```

//串转并
void series_to_parallel(uchar dat)
{
    int i = 0;
    clk = 0;
    for(i = 0; i < 8; i++)
    {
        inA = (dat >>(7 - i)) & 0x01;
        clk = 1;
        clk = 0;
    }
}

void digitron_init(void)
{
    inB = 1;
    g2 = 0;
    clk = 0;
    series_to_parallel(0x00);
}

void digitron_display(uchar dat, uchar i)
{
    series_to_parallel(0x00);
    g2 = 1;
    A1 = (i >> 0) & 0x01;
    B1 = (i >> 1) & 0x01;
    C1 = (i >> 2) & 0x01;
    series_to_parallel(digitron_data[dat]);
    g2 = 0;
    delay_ms(2);
}

void digitron_dis(char* dat, uchar num)
{
    int i;
    for(i = 0; i < num; i++)
    {

```

```

        digitron_display(dat[i], num - 1 - i);
    }
}

```

主函数

```

#ifndef _MAIN_H_
#define _MAIN_H_

#define uchar unsigned char
#define uint unsigned int
#define uint32_t unsigned long

#endif

#include "reg52.h"
#include "main.h"
#include "digitron.h"

uchar snumber[] = {1, 9, 2, 1, 7, 8, 1, 9};
extern uchar uart_cache[];
extern uchar rx_pointer;

void uart_rx_proc(void);

void main()
{
    digitron_init();
    while(1)
    {
        digitron_dis(snumber, 8);
    }
}

```