哈尔滨理工大学

《单片微型计算机原理与嵌入式系统》

项目报告

项目2 数字钟的设计与实现

班级： 测控23-5

姓名： 郭晟芃

学号： 2305010504

# 项目目的

1. 掌握四位共阴极数码管的使用方法
2. 掌握独立按键与外部中断的使用方法
3. 掌握51单片机定时器的操作

# 项目要求

1.说明如何使用51单片机定时器的工作流程

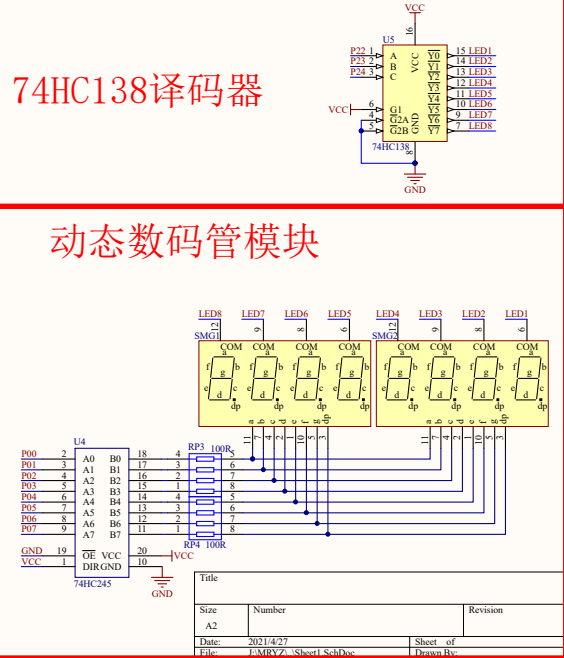
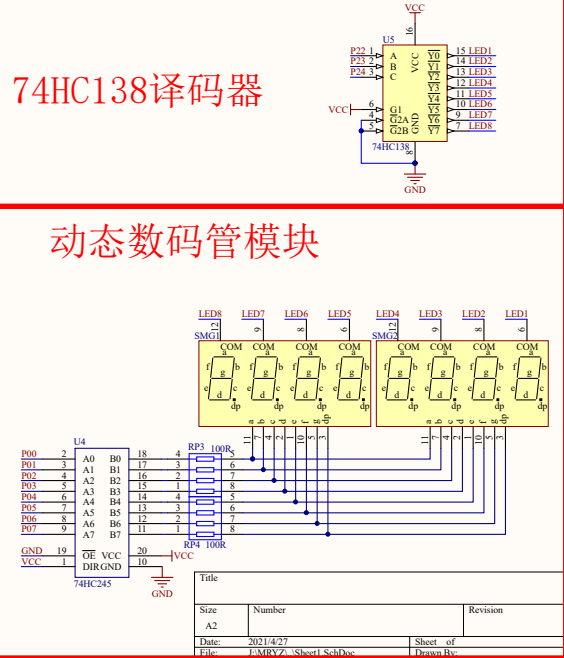
2.说明如何使用51单片机外部中断的工作流程

3.基于51单片机的定时器功能编写一个使用八位数码管显示时间的秒表

# 项目原理

## 数码管动态显示

本次实验使用的是共阴极数码管，其 8 个发光二极管的阴极在数码管内部全部连接在一起，所以称“共阴”，，而它们的阳极是独立的，通常在设计电路时一般把阴极接地。当我们给数码管的任意一个阳极加一个高电平时，对应的这个发光二极管就点亮了，用人眼视觉暂留和数码管显示的余晖的原理，通过对显示的数字进行延迟就能达到显示多个数字的效果。

****

原理图 1

如原理图 1所示，多位数码管的段选端由P2.0、P2.1、P2.2通过一个74HC138芯片控制，位选端由P0口通过一个74HC245芯片控制。

## 51单片机定时器/计数器

定时/计数器实质上是加一的计数器。内部/外部脉冲让它不断加一，当定时/计数器溢出时，定时/计数器的溢出相应的中断标志位置1，并向CPU发出中断请求，进入中断服务函数。如果定时/计数器工作为定时模式，则表示时间已到；如果工作于计数模式，则表示计数值已满。

## 按键与外部中断

当中央处理机 CPU 正在处理某件事的时候外界发生了紧急事件请求，要求 CPU 暂停当前的工作，转而去处理这个紧急事件，处理完以后，再回到原来被中断的地方，继续原来的工作，这样的过程称为中断。本次实验通过独立按键触发外部中断的形式实现按键的读取，如原理图 2所示，按键K3连接在单片机的P3.2口(外部中断0)，通过对外部中断0的操作即可进行按键的读取。

原理图 2

# 项目所用到的代码

//smg.c文件

1. #include "smg.h"
3. //共阴极数码管段码
4. u8 gsmg\_code[17]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,
5. 0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
7. u8 gsmg\_code\_dot[10] = {0xbf,0x86,0xdb,0xcf,0xe6,0xed,0xfd,0x87,
8. 0xff,0xef};
10. **void** smg(u8 dat[]){
11. u8 i;
12. **for**(i=0;i<8;i++){
13. **switch**(i) {//位选
14. **case** 0: LSC=1;LSB=1;LSA=1;**break**;
15. **case** 1: LSC=1;LSB=1;LSA=0;**break**;
16. **case** 2: LSC=1;LSB=0;LSA=1;**break**;
17. **case** 3: LSC=1;LSB=0;LSA=0;**break**;
18. **case** 4: LSC=0;LSB=1;LSA=1;**break**;
19. **case** 5: LSC=0;LSB=1;LSA=0;**break**;
20. **case** 6: LSC=0;LSB=0;LSA=1;**break**;
21. **case** 7: LSC=0;LSB=0;LSA=0;**break**;}
22. **if**(i==1||i==3||i==5){
23. SMG\_A\_DP\_PORT=gsmg\_code\_dot[dat[i]];}
24. **else** SMG\_A\_DP\_PORT=gsmg\_code[dat[i]];
25. delay\_10us(100);
26. SMG\_A\_DP\_PORT=0x00;}}//消音

//smg.h文件

1. #ifndef \_smg\_H
2. #define \_smg\_H
3. #include "public.h"
4. #define SMG\_A\_DP\_PORT   P0  //数码管段选控制
6. sbit LSA=P2^2;  //数码管位选控制
7. sbit LSB=P2^3;
8. sbit LSC=P2^4;
10. **void** smg(u8 dat[]);
12. #endif

//main.c文件

1. #include "smg.h"
2. #include "public.h"
4. u8 times[8] = {0,0,0,0,0,0,0,0};        //用于存储时间的值
5. **static** key\_value = 0;                   //保存按键按下的次数
7. **void** time\_0\_set(){                      //定时器0设置
8. EA = 1;
9. ET0 = 1;
10. TMOD |= 0x01;
11. TH0 = 0xFC;
12. TL0 = 0x66;}
14. **void** int0\_set(){                        //外部中断0设置
15. EX0 = 1;
16. EX1 = 1;
17. IT0 = 1;
18. IT1 = 1;}
20. **void** time\_change(){                      //时间进位
21. u8 i;
22. **for**(i=7;i>0;i--){
23. **if**(i==2 || i==4 ){
24. **if**(times[i] == 6){
25. times[i-1]++;
26. times[i] = 0;}}
27. **else**{
28. **if**(times[i] == 10){
29. times[i-1]++;
30. times[i] = 0;}}}}
32. **void** main(){
33. init\_set();
34. time\_0\_set();
35. **while**(1){
36. smg(times);}}
38. **void** time() interrupt 1{
39. **static** u16 i = 0;
40. TH0 = 0xFC;
41. TL0 = 0x66;
42. i++;
43. **if**(i == 10){
44. times[7]++;
45. time\_change();
46. i = 0;}
47. TF0 = 0;
48. TR0 = 1;}
50. **void** key() interrupt 0{
51. delay\_10us(1000);
52. **if**(P3^2 == 0){
53. **if**(key\_value == 0){TF0 = 0;TR0 = 1;key\_value++;IE0 = 0;}
54. **else** **if**(key\_value == 1){TF0 = 0;TR0 = 0;key\_value++;IE0 = 0;}
55. **else** **if**(key\_value == 2){
56. u8 j;
57. **for**(j=0;j<8;j++){times[j] = 0;TF0 = 0;}//数组归零
58. key\_value = 0;
59. IE0 = 0;}}}

# 实验现象

当按键第一次按下时，开始计时；第二次按下，停止计时；第三次按下，读数归零。