

# 基于C8051F020的智能控制系统设计与实现

作者：自动化96 薛荣坤

学号：2196113513

## 1 智能系统硬件部分

1.1 智能控制器系统结构

1.2 单片机

## 2 系统各个模块讲解

2.1 按键

2.2 液晶显示电路

2.3 数码管

2.4 显示音乐

2.5 时间显示

## 3 搭建系统主函数

3.1 引脚初始化

3.2 准备问题

3.3 初始化硬件和变量

3.4 回答问题模块

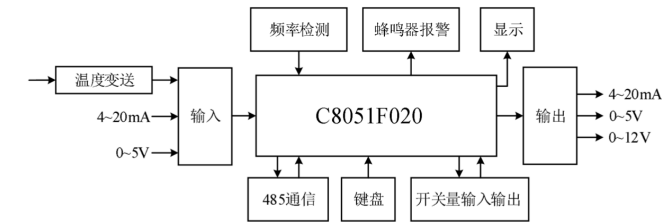
3.5 得分结果

## 4 实验分析和总结

# 1 智能系统硬件部分

## 1.1 智能控制器系统结构

智能控制器是以 C8051F020 单片机为核心部件，增加了按键模块、数码管和液晶屏显示模块、AD 和 DA 输入输出信号调理 电路等。智能控制器系统框图如图 1-1 所示



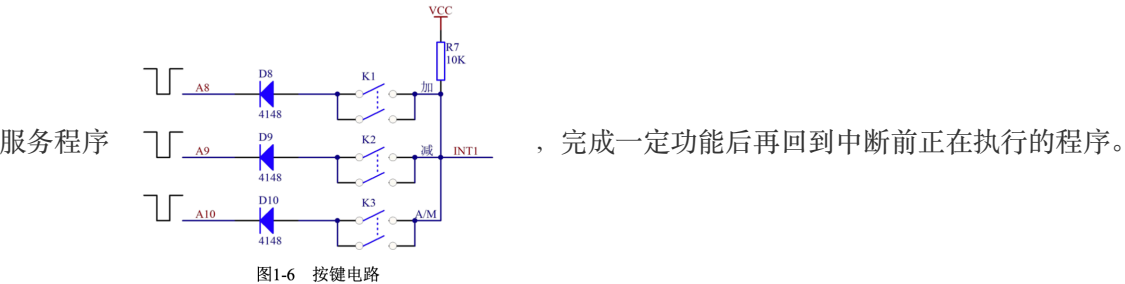
## 1.2 单片机

基于C8051F020单片机的特性，智能控制器在设计时，选取该单片机作为控制、计算、显示的核心部件。C8051F020单片机有100个引脚，封装为TQFP-100C8051F020单片机低端口(P0、P1、 P2、P3)既可以按位寻址，也可以按字节寻址，高端口(P4、P5、P6、P7)只能按字节寻址，所有引脚都 可以被配置为开漏或推挽输出方式。C8051F020单片机有大量的数字资源需要通过P0、P1、P2和P3端口才 能使用。P0、P1、P2和P3中的每个引脚即可定义为通用的I/O端口引脚，也可以分配给一个数字外设或功能 (例如:UART0或INT1)。这种资源分配的灵活性是通过使用优先权交叉开关实现的。

# 2 系统各个模块讲解

## 2.1 按键

智能控制器的按键电路，如图1-6所示。三个按键信号A8、A9、A10分别接在C5051F020单片机的P5.0、P5.1、P5.2 引脚，中断信号INT1接在P0.3引脚。按键由外部中断信号触发，低电平有效，按键按下触发中断， 进入按键中断



假设某一按键按下，P5 对应的位置为逻辑‘0’，此时这条线路形成 通路，才能将低电平信号和 INT1 连通。如果在 按键扫描时，让 P5 的三个端口状态轮流为逻辑‘0’，读取 P5 端口的值，就可以建立 P5 的端口和三个按键的一一 对应关系。共阳极数码管显示的位选是轮流使能的过 程，故将数码管的位选端口与 INT1 的端口复用。

```
1 //在按键触发中断后，不同按键会使能flag的标志。
2 void INT1_ISR(void) interrupt 2
3 {
4     Delay_ms(1);
5     answerf=1;
6     switch(P5){
7         case 0xfb:
8             flag=1;
```

```

9         break;
10    case 0xfd:
11        flag=2;
12        break;
13    case 0xfe:
14        flag=3;
15        break;
16    }
17 }
18
19

```

## 2.2 液晶显示电路

液晶显示电路中，智能控制器选用HS12864-15B汉字图形型液晶，带中文字库。液晶显示采用串口通信模式，可以显示字母、数字符号、中文字型及图形，具有绘图及文字画面混合显示功能。该液晶共有20个引脚，E、RW、RS分别接在单片机的P1.3、P1.4、P1.5引脚，引脚说明如表1-1所示，没有列出的引脚是空接状态。P1.3、P1.4、P1.5引脚在系统端口初始化时被设置为推挽模式，一旦端口在初始化时进行了交叉开关配置，则在程序运行过程中，端口不能进行修改。RW引脚(P1.4)在端口初始化时被配置成推挽输出，则不能读取LCD返回的数据。

```

1  //在这个例子中，我们只需要掌握在屏幕上打字即可。
2  uchar code str0[]="抑郁症初步测试";
3  uchar code str1[]="结果仅参考";
4  uchar code str2[]="小鱼出品";
5  uchar code str3[]="任意键跳过";
6  WriteStr(0, 0, str0);
7  WriteStr(1, 0, str1);
8  WriteStr(2, 0, str2); WriteStr(3, 0, str3);

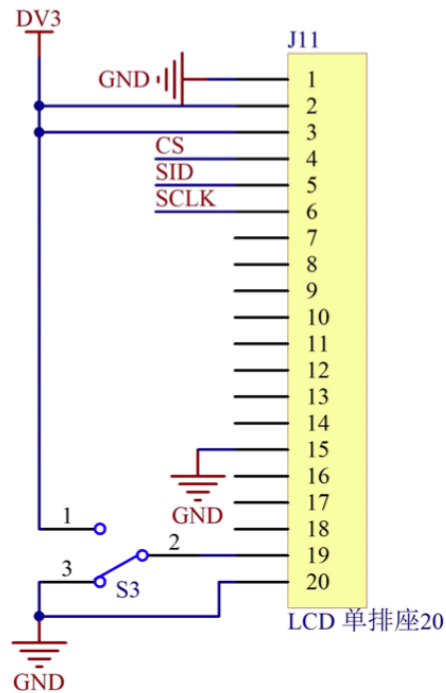
```

## 2.3 数码管

智能控制器设置了三组四位数码管，其中一组数码管的显示电路，数码管采用动态扫描显示方式，数码管为共阳极接法，位选信号为逻辑‘0’表示该位对应的数码管被选中，数码管显示内容由段选信号决定，利用余辉效应可以分时复用P7端口，来点亮特定的数码管，显示对应的数字。在实际应用中，采用两种方式可以达成余辉的效果。一种方式是在程序中设置延时，来制造余辉的效果。另一种方式是程序在一次循环中需要执行很多指令的情况下，这

样即使设置了延迟，

程序执行的时 间往往远超过期望的延迟，则余晖的



效果会变成闪烁，严重影响数码管的显示。在这种情况下，可以考虑 采用定时器中断进行特定周期的触发以达成稳定显示。

```
1 void ResDispNum() //数码管显示数字
2 {
3     uchar temp2[4];
4     uchar temp3[4];
5
6     temp2[0] = score%10;
7     temp2[1] = score%100/10;
8     temp2[2] = score%1000/100;
9     temp2[3] = score/1000;
10
11     temp3[0] = time%10;
12     temp3[1] = time%100/10;
13     temp3[2] = time%1000/100;
14     temp3[3] = time/1000;
15
16     // 最上边一行显示
17     select(4);display(0); Delay(500); P7 = 0xff;
18     select(3);display(0); Delay(500); P7 = 0xff;
19     select(2);display(0); Delay(500); P7 = 0xff;
20     select(1);display(0); Delay(500); P7 = 0xff;
21
22     // 中间一行
23     select(8);display(temp2[0]); Delay(500); P7 = 0xff;
24     select(7);display(temp2[1]); Delay(500); P7 = 0xff;
25     select(6);display(temp2[2]); Delay(500); P7 = 0xff;
```

```

26     select(5);display(temp2[3]); Delay(500); P7 = 0xff;
27
28     // 下边一行
29     select(12);display(temp3[0]); Delay(500); P7 = 0xff;
30     select(11);display(temp3[1]); Delay(500); P7 = 0xff;
31     select(10);display(temp3[2]); P7 = P7 & ~0x80; Delay(500); P7 = 0xff;
32     select(9) ;display(temp3[3]); Delay(500); P7 = 0xff;
33
34 }

```

## 2.4 显示音乐

```

1 //这是一段经典的代码用于播放音乐
2 void PlayMusic()
3 {
4     uint i=0,j,k;
5     while(SONG_LONG[i]!=0||SONG_TONE[i]!=0){
6         for(j=0;j<SONG_LONG[i]*20;j++){
7             if(P4==0x00) P4=0xff;
8             else P4=0x00;
9             for(k=0;k<SONG_TONE[i]/3;k++);
10        }
11        Delay_ms(10);
12        i++;
13    }
14 }

```

## 2.5 时间显示

```

1 //通过计时器不断溢出产生中断实现给数码管不断记数的原理
2 void Timer0_ISR (void) interrupt 1
3 {
4     TH0 = TIMERO_RELOAD_HIGH;           // Reinit Timer0 High register
5     TLO = TIMERO_RELOAD_LOW;
6     Time_num++;
7     if(Time_num >= 6000) Time_num = 0;
8

```

## 3 搭建系统主函数

注：代码这部分相对容易，直接讲解

### 3.1 引脚初始化

```
1  int mm;
2  Question Que[15];
3  Init_Device();
4  Timer0_Init();
5  INT1 = 1;
6  LcdInit();
7  //初始化硬件并开启中断。
```

### 3.2 准备问题

```
1  //存储所有希望提问的句子到数组中
2  strcpy(Que[0].que, "你想要打人吗");
3      strcpy(Que[1].que, "人们对我不太友好");
4      strcpy(Que[2].que, "不大想吃东西");
5      strcpy(Que[3].que, "心里觉得苦闷");
6      strcpy(Que[4].que, "总觉得自己不如人");
7      strcpy(Que[5].que, "无法集中精力");
8      strcpy(Que[6].que, "自觉情绪低沉");
9      strcpy(Que[7].que, "任何事情觉得费力");
10     strcpy(Que[8].que, "自己的生活是失败");
11     strcpy(Que[9].que, "感到害怕");
12     for (mm=0; mm<10; mm++){
13         strcpy(Que[mm].ansA, "每周0--3次");
14         strcpy(Que[mm].ansB, "每周4--6次");
15         strcpy(Que[mm].ansC, "每周7 次以上");
16     }
17     strcpy(Que[12].que, "你是乐观的孩子");
18     strcpy(Que[12].ansA, "阳光并且积极");
19     strcpy(Que[12].ansB, "困难不会打败你");
20     strcpy(Que[12].ansC, "继续加油吧");
21
22     strcpy(Que[13].que, "你是理智的孩子");
23     strcpy(Que[13].ansA, "居安思危的处事");
24     strcpy(Que[13].ansB, "谨慎的走每一步");
25     strcpy(Que[13].ansC, "继续加油吧");
26
27     strcpy(Que[14].que, "你应当寻求帮助");
28     strcpy(Que[14].ansA, "医生或者心理辅导");
29     strcpy(Que[14].ansB, "不用担心说出");
```

```
30 strcpy(Que[14].ansC, "继续加油吧");
31
```

### 3.3 初始化硬件和变量

```
1 *****
2 //开机动画
3 *****
4 ImageShow( xjtu_image2);
5 Delay_ms(3000);
6 LcdInit();
7 WriteStr(0, 0, str0);
8 WriteStr(1, 0, str1);
9 WriteStr(2, 0, str2);
10 WriteStr(3, 0, str3);
11 Delay_ms(3000);
12 LcdInit();
13 //变量初始化
14 score=0;
15 flag=0;
16 Time_num=0;
17 time=0;
```

### 3.4 回答问题模块

```
1 //进入代码循环开始回答问题，通过按键结果给出相应
2 while(question<10){
3     LcdInit();
4     WriteStr(0, 0, Que[question].que);
5     WriteStr(1, 0, Que[question].ansA);
6     WriteStr(2, 0, Que[question].ansB);
7     WriteStr(3, 0, Que[question].ansC);
8
9     P4=0xff;
10    Delay_ms(100);
11    P4=0x00;
12    //计时开始提示音
13    TimeCounter(5);
14    P4=0xff;
15    Delay_ms(100);
16    P4=0x00;
17    //计时结束
18    if(flag==1){
19        score+=0;
20        LcdInit();
```

```

21     ImageShow(xjtu_image1);
22     Delay_ms(50);
23 }
24 if (flag==2){
25     score+=2;
26     LcdInit();
27     ImageShow(xjtu_image1);
28     Delay_ms(50);
29 }
30 if (flag==3){
31     score+=3;
32     LcdInit();
33     ImageShow(xjtu_image1);
34     Delay_ms(50);
35 }
36 flag=0;
37 answerf=0;
38 //标志位重新置为0
39 question++;
40 }

```

### 3.5 得分结果

```

1 //退出循环后给出得分
2 if(score<=5){
3     LcdInit();
4     WriteStr(0, 0, Que[12].que);
5     WriteStr(1, 0, Que[12].ansA);
6     WriteStr(2, 0, Que[12].ansB);
7     WriteStr(3, 0, Que[12].ansC);
8     P4=0x00; //关闭蜂鸣器
9     PlayMusic(); //播放音乐
10    Delay_ms(3000);
11 }
12 if(score>5&&score<=20){
13     LcdInit();
14     WriteStr(0, 0, Que[13].que);
15     WriteStr(1, 0, Que[13].ansA);
16     WriteStr(2, 0, Que[13].ansB);
17     WriteStr(3, 0, Que[13].ansC);
18     P4=0x00; //关闭蜂鸣器
19     PlayMusic(); //播放音乐
20     Delay_ms(3000);
21 }
22 if(score>20&&score<=30){

```



```
23     LcdInit();
24     WriteStr(0, 0, Que[14].que);
25     WriteStr(1, 0, Que[14].ansA);
26     WriteStr(2, 0, Que[14].ansB);
27     WriteStr(3, 0, Que[14].ansC);
28     P4=0x00;//关闭蜂鸣器
29     PlayMusic();//播放音乐
30     Delay_ms(3000);
31 }
```

## 4 实验分析和总结

实验1主要是熟悉硬件的配置，利用硬件资源为之后的使用提供经验，在这里我们因为两个板子的型号不同，所以调试了相当长的时间，以后一定要先检查硬件，再进行调试。