

AVR单片机课程设计报告

**题目 娱乐按键时钟**

组长 李明江

组内成员 李明江，鲁博文

学号 2018107243，2018171107

专业 物联网工程

班级 20181711

完成日期 2021年 1月 3日

目 录

[1．项目概要设计](#_Toc423899749)

[1.1 架构设计及功能](#_Toc423899750)

[2．项目详细设计](#_Toc423899753)

[2.1 硬件设计](#_Toc423899754)

[2.1.1 功能性能描述](#_Toc423899760)

[3.2 软件设计](#_Toc423899759)

[3.2.1 功能性能描述](#_Toc423899760)

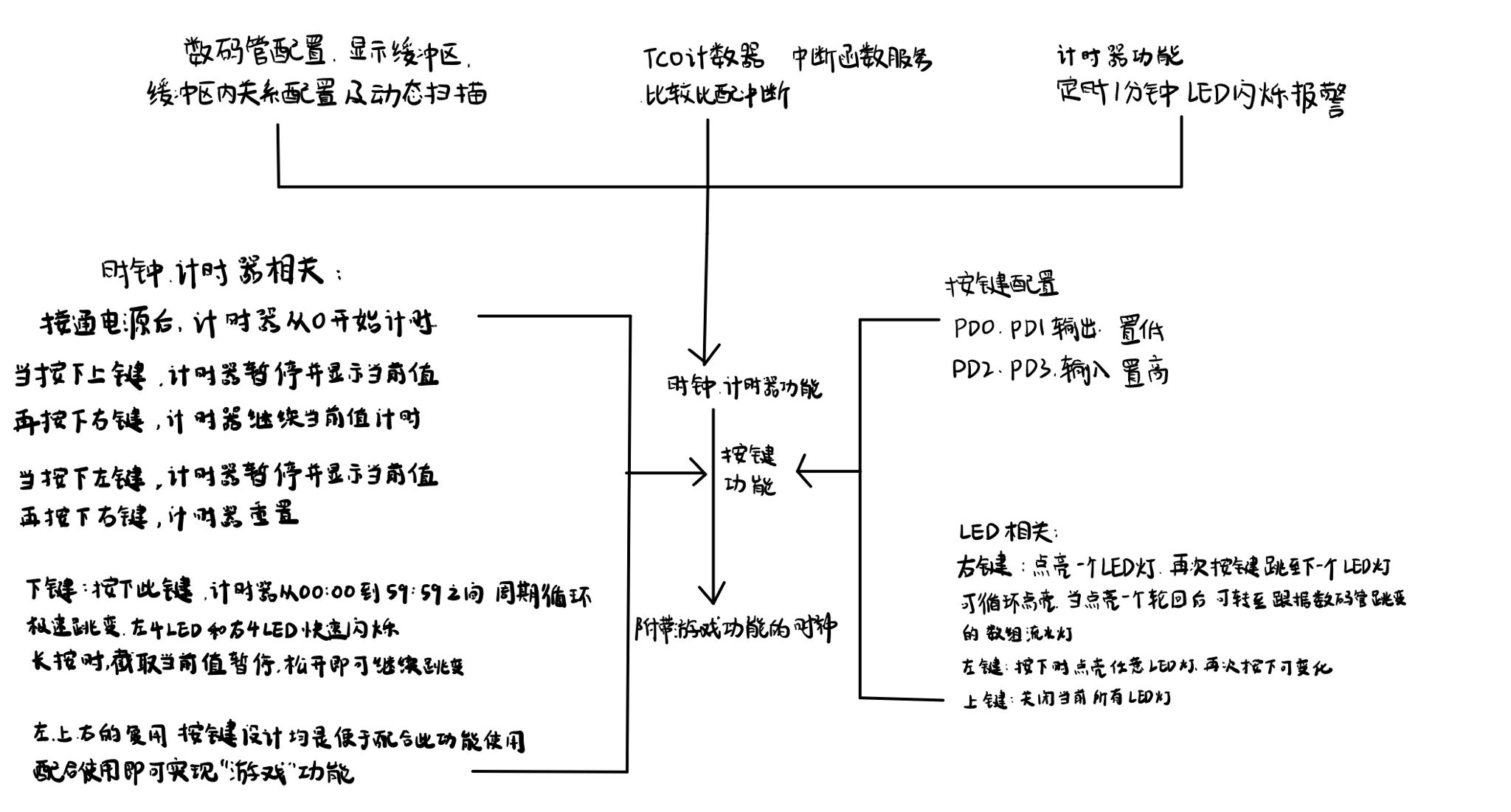
[4．项目实现与测试](#_Toc423899764)

[5．心得体会](#_Toc423899765)

（三峡大学 计算机与信息学院）

**1．项目概要设计**

**1.1 架构设计及功能**



接通电源后，计时器开始计时，计时一分钟，灯光开始闪烁报警。

LED 相关：  
右键：点亮一个LED灯，再次按键跳至下一个LED灯  
可循环点亮,当点亮一个轮回后可转至跟据数码管时序跳变的数组流水灯  
左键：按下时点亮任意LED灯，再次按下可变化（根据数组变化）  
上键：关闭当前所有 LED 灯

数码管相关：

当按下上键，计时器暂停并显示当前值  
再按下右键，计时器继续当前值计时  
当按下左键，计时器暂停并显示当前值  
再按下右键，计时器重置

下键:按下此键,计时器从00:00到59:59之间周期循环  
极速跳变, LED 快速闪烁  
左上的复用按键设计均是便于配合此功能使用  
配合使用即可实现"游戏功能

即：但数码管急速跳变时，按下左键，数码管暂停并显示当前数值，此时按下其它任意键即可清零重置，但因为其它三个键中的上下键都有控制LED相关的功能，因此一般选择按下右键，来重置数码管

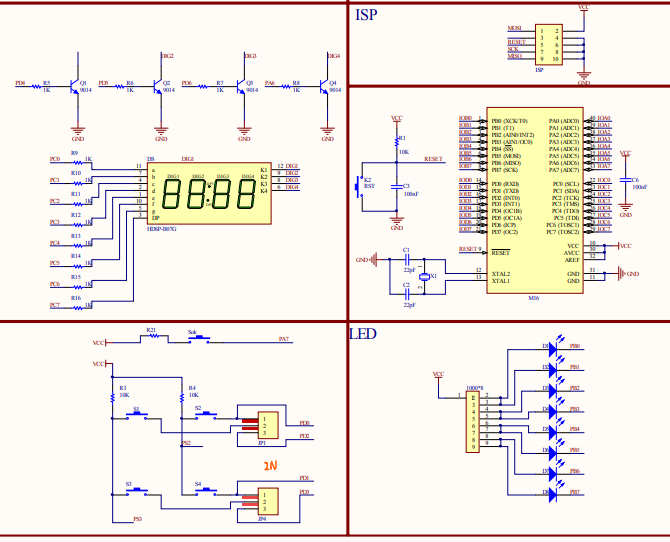
当按下上键时，暂停。同理，按下右键即可继续当前数值开始转为正常时钟状态

当按下下键，数码管急速跳变时，长按下键时，可显示当前值，松开即可继续当前值跳变

**2．项目详细设计**

**2.1 硬件设计**

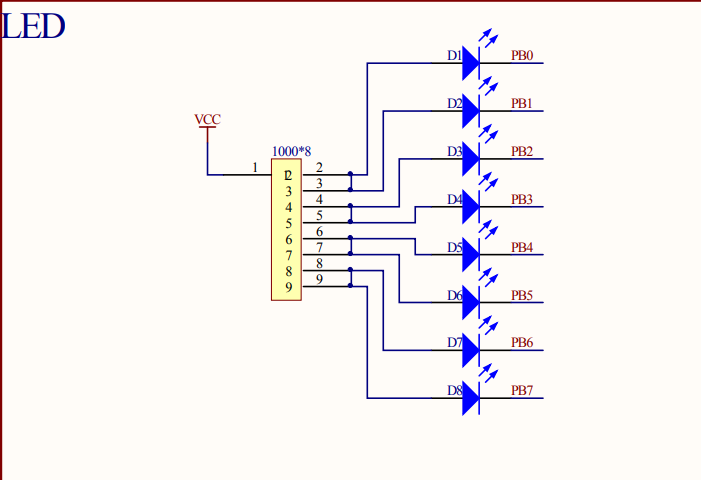
**总体电路图如下：**



2.1.1 功能性能描述

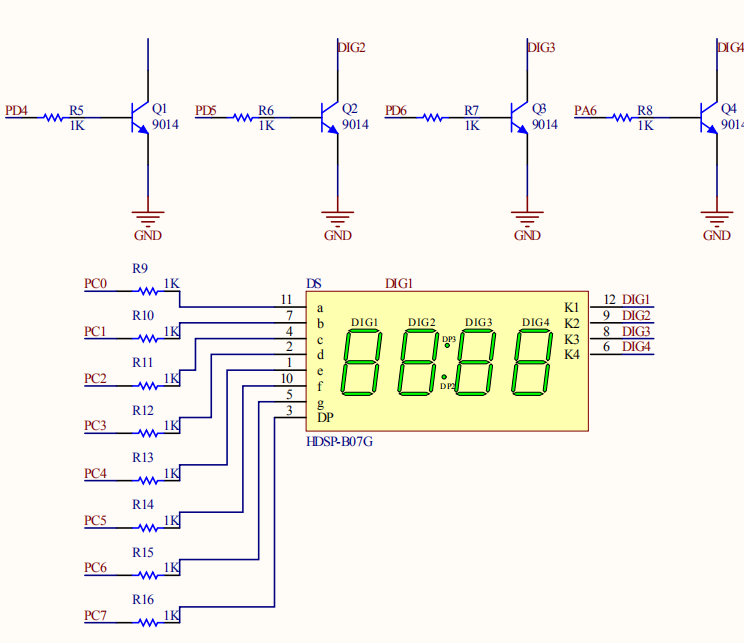
（1）LED模块

将PB口均置为输出，将PB某个口置为低电平时，LED灯发光。

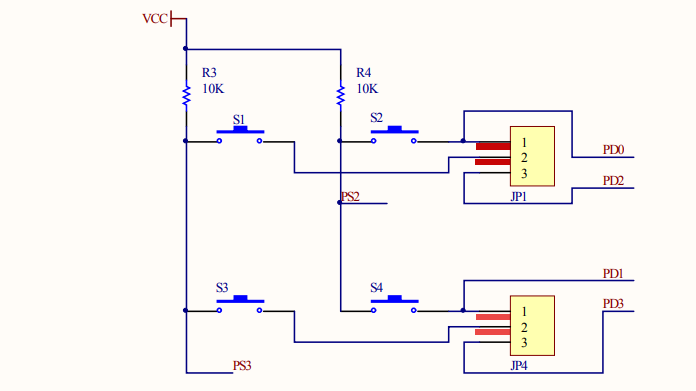


（2）数码管模块

PD4,PD5,PD6,PA6片选数码管和点，PC0-PC7控制数码管和点的亮暗，其电路图如下：



（3）Key模块



扫描时，将PD0、PD1置为输出，PD2、PD3置为输入，并将PD2、PD3置为高电平，PD0、PD0置低电平。按键按下时，置PD2、PD3为输出，置PD1、PD0为输入，通过读取PD输入的状态从而判断当前是哪个按键按下.

**3.2 软件设计**

**3.2.1 功能性能描****述**

**（1）IO初始化**

**void IO\_init(void)**

**{**

**DDRB =0xFF; //设置PB口为输出**

**DDRD.7=1; //蜂鸣器**

**DDRC =0xFF; //设置PC口为输出**

**DDRD =0xf0; //设置PD口为输入**

**PORTD=0x0f; //设置PD上拉**

**PORTC = 0x00;**

**DDRC = 0xff;**

**PORTA = 0x00;**

**DDRA = 0x40;**

**PORTB = 0xff;**

**DDRB = 0xff;**

**（2）数码管扫描显示**

**void display() // 扫描显示函数，执行时间 12ms**

**{**

**PORTA.6=0;**

**PORTD =0x03;**

**//char i;**

**//for(i=0;i<=3;i++)**

**//{**

**PORTC = led\_7[dis\_buff[posit]];**

**if (point\_on && posit==2) PORTC |= 0x80; // (1)**

**//PORTD = position[posit];**

**//PORTA = positiona[posit];**

**if(posit==0)**

**{**

**PORTA.6=1;**

**}**

**else**

**{**

**PORTD = position[posit];**

**}**

**//delay\_ms(2); // (2)**

**//PORTC = 0xff; // (3)**

**if(++posit >= 4)**

**{**

**posit = 0;**

**}**

**//}**

**}**

**（3）匹配比较中断调用和计时器0配置**

**interrupt [TIM0\_COMP] void timer0\_comp\_isr(void)**

**{**

**display(); // 调用 LED 扫描显示**

**if (++time\_counter>=500)**

**{**

**time\_counter = 0;**

**time\_1s\_ok = 1;**

**}**

**}**

**（4）缓冲区**

**void time\_to\_disbuffer(void) // 时间值送显示缓冲区函数**

**{**

**char i,j=0;**

**for (i=0;i<=1;i++)**

**{**

**dis\_buff[j++] = time[i] % 10;**

**dis\_buff[j++] = time[i] / 10;**

**}**

**}**

**（5）按键扫描**

**void keyScan()**

**{**

**// PORTD=0x00;**

**uchar x,y;**

**DDRD =0x03; //设置PD口高4位为输入**

**PORTD=0x0c; //设置PD高4位上拉,此时PIND高4位全1**

**if((PIND&0x0c)!=0x0c) //判断是否有按键按下**

**{**

**delay(20); //延时消抖**

**if((PIND&0x0c)!=0x0c) //再次判断是否有按键按下**

**{**

**// PORTB=0x0f;**

**x=(PIND&0x0c);**

**DDRD =0x0c;**

**aa=aa%8;**

**aa++;**

**bb=bb%10;**

**bb++;**

**delay(5);**

**y=(PIND&0x03);**

**keyValue=x|y;**

**while((PIND&0x03)!=0x03);**

**}**

**//等待按键松开**

**// PORTB=0xf0;**

**}**

**}**

**（6）按键条件与配置**

**void keyHandle()**

**{**

**switch(keyValue)**

**{**

**case 0x06: //S17 上**

**{**

**PORTB=0XFF;**

**time\_1s\_ok = 0;**

**break;**

**}**

**case 0x0a: //S18 左**

**{**

**PORTB=led\_7[bb];;**

**time\_1s\_ok = 0;**

**time[1]=0;**

**time[0]=0;**

**break;**

**}**

**case 0x05: //S19 下**

**{**

**time\_1s\_ok = 1;**

**break;**

**}**

**case 0x09: //S20 右**

**{**

**PORTB=resit[aa];**

**break;**

**}**

**default:**

**// PORTB=0xf0;**

**break;**

**}**

**}**

**}**

**4.项目实现与测试**

**4.1 主函数代码代码实现**

**while(1) //死循环**

**{**

**// PORTB=0xff;**

**// PORTC=0x66;**

**/\* LED\_1=1;**

**LED\_2=1;**

**LED\_3=1;**

**LED\_4=1;\*/**

**keyScan(); //按键扫描**

**keyHandle(); //按键处理**

**display();**

**display();**

**display();**

**keyScan(); //按键扫描**

**keyHandle(); //按键处理**

**if (time\_1s\_ok) // 1 秒到**

**{**

**time\_1s\_ok = 0;**

**point\_on = ~point\_on;**

**if(i>=1)**

**{**

**if(time[0]>=31)**

**{PORTB=0xff;**

**PORTD.7=0;}**

**else if(time[0]%2==0)**

**{PORTB = 0x00;**

**PORTD.7=1;}**

**else**

**{ PORTB = 0xff;**

**PORTD.7=0;}**

**}**

**if (++time[0] >= 60) // 以下时间调整**

**{**

**time[0] = 0;**

**//PORTD.7=1;**

**i++;**

**if (++time[1] >= 60)**

**{**

**time[1] = 0;**

**//if (++time[2] >= 24) time[2] = 0;**

**}**

**}**

**time\_to\_disbuffer(); // 新调整好的时间送显示缓冲区**

**}**

**}**

**}**

1. **心得体会**

通过几周的学习，进一步的理解了单片机的部分原理，对单片机的结构、各个功能配置、计数器、中断的使用以及按键矩阵的控制有了深入地理解。在整个项目的配置中，我们也遇到过很多困难，但都在不断调试与改进中慢慢解决和克服。如刚开始数码管配置的差不多了，但在实物上显示就是不明显，发现在烧录软件中取消使能JTAG接口后，效果便明显了很多。也比如在添加按键功能的时候，按键经常失灵或者无法使用，发现是PD接口高位配置和地位配置的矛盾，因此在主函数中display（）数码管扫描前后各加一次keyScan();按键扫描和keyHandle();按键处理，按键情况便得以解决。与此同时也产生了一个问题：数码管变按。于是便在中间加入了三个数码管扫描函数，此时，数码管效果便十分明显。

之后是按键功能配置的问题，最开始的想法是让按键一个去控制片选，一个去增加数码管中某个片选的值。但发现，上面时钟实现的过程中，为了实现片选和和数码管显示不同序，使用了同一个变量posit，因此放弃了这个。之后在讨论中，理解了计数器匹配中断的过程，因此做出了如何控制数码管暂停和继续，以及重置的功能。并在尝试的过程中将书中用来作“标志位”time\_1s\_ok置为0，发现了一种快乐的时钟玩法，想到了曾经在商场中见过的一个娱乐游戏：当数码管急速跳变时，按下数字有几个6，便可中几级的奖励，也有按某个临界值下最大的数，按准10.0000s等考验反应力、预判和运气的游戏。在此基础上利用break语句实现了多状态的复用，并闲置出了可以被复用的右键，顺带着做了比较娱乐性的led灯功能，添加完LED相关按键控制后，成品完成。