标签：

DIY 基于 51 单片机的旋转 LED 数字电子钟 (红外线遥控调时 )

在网上看到不少老外做的各种旋转ＬＥＤ显示屏， 非常ＣＯＯＬ， 我也动手用洞洞板试做了一个类似的显示屏， 结果感觉还不错。 于是再接再励继续努力，将作品进一步改进，完善后制成如今这个样子。

由于刚学 51 单片机，加上制作电路板软件也是从零开始，的确花了我不少的时间和精力。 不过也就是在这艰难的独立制作中， 真正学到了不少实在的东西。

本项目的关键是如何解决高速旋转的电路板如何供电， 如何调时的问题。我采用电机电刷的原理，将旋转轴钻空，通过一只插头将电源的 从反面引到前面的电路板上， 而这个旋转的插头又与固定在背板上的两 个铜片接触的。调时的问题有些困难， 一是让电路板在旋转前与ＰＣ机相接，由电脑传送调时数据，这虽然可行但不方便。还有就是用遥控方 法，但此方案在调试方面有很大的困难。

显示方式上， 我采用平衡式的两排ＬＥＤ，这除了在旋转时能较好的保持平衡外，主要能利用两边交替显示方式，比单排要快一倍。

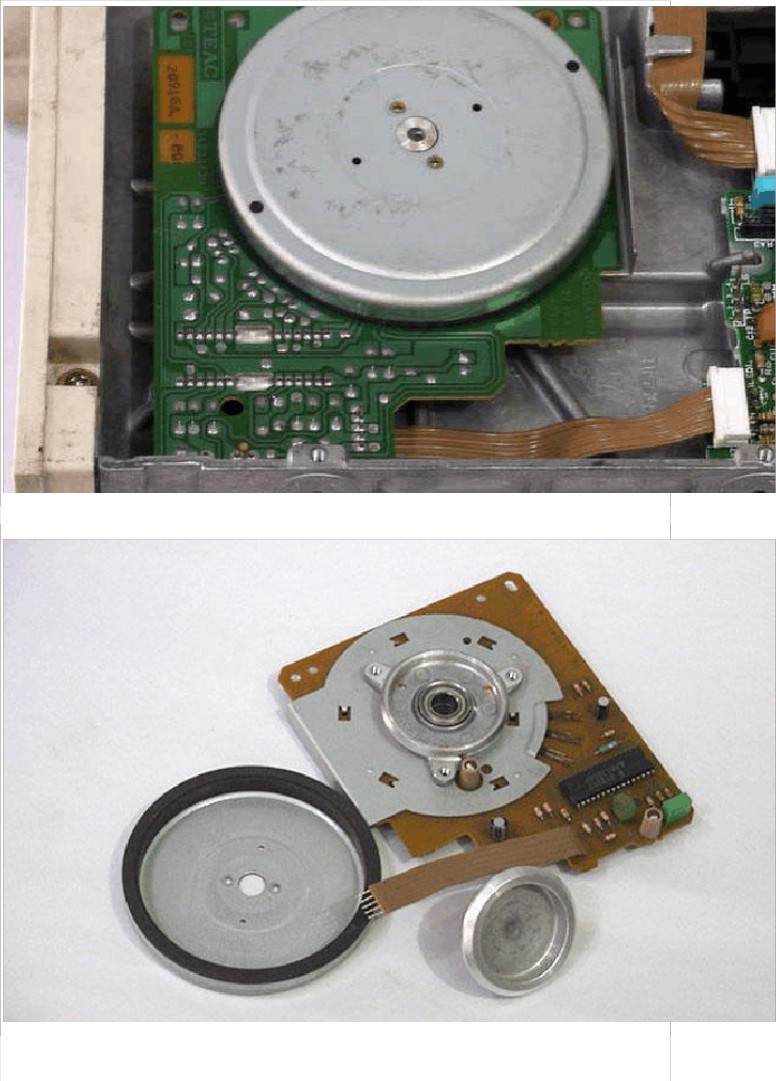
本装置不仅是一个时钟， 它还可以动态显示汉字及图案， 这就看如何发挥了。

其具体制作过程如下：

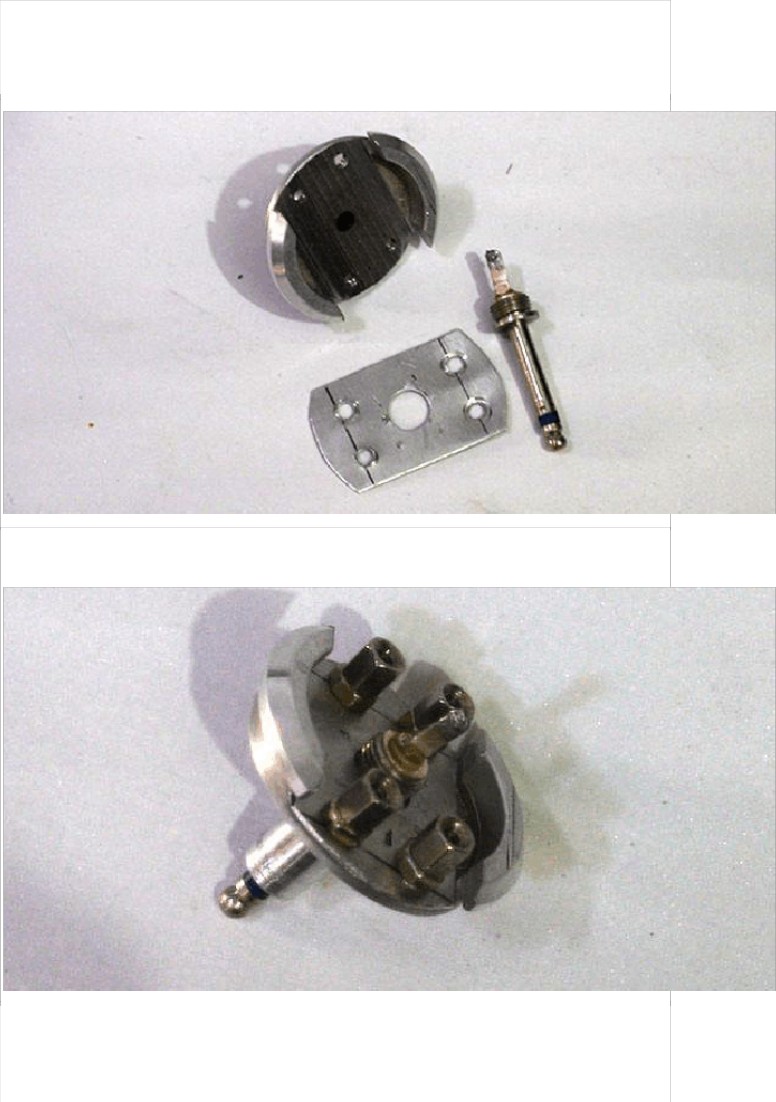
一。旋转电机的制作

从制作成本与方便考虑， 选用旧电脑用的大软驱上的直流无刷电机， 只是对局部进行改造。

就是这种古董软驱

软驱上的直流无刷电机

拆开后的电机



仔细拆开直流电机， 将带圆盘的铝轴从中开孔，让它刚好能插入一个插头。

将旋转轴加工成这样

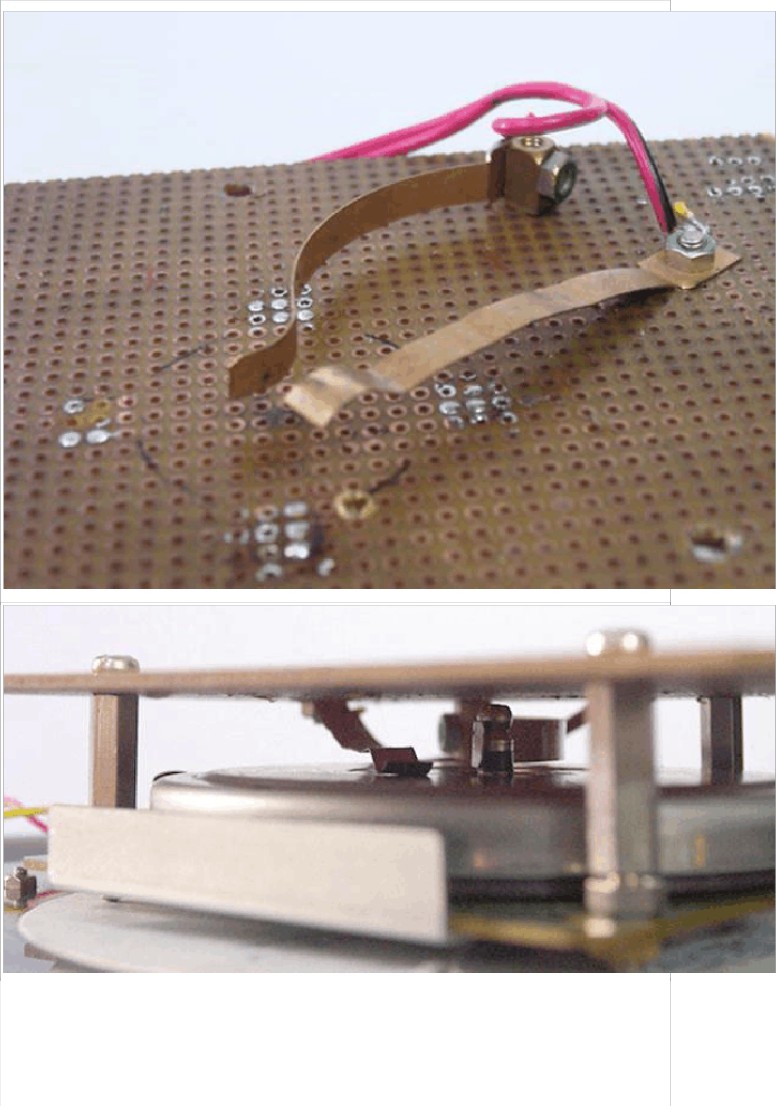
装配好以后

按拆开时的顺序，反序将轴安装直流电机上。

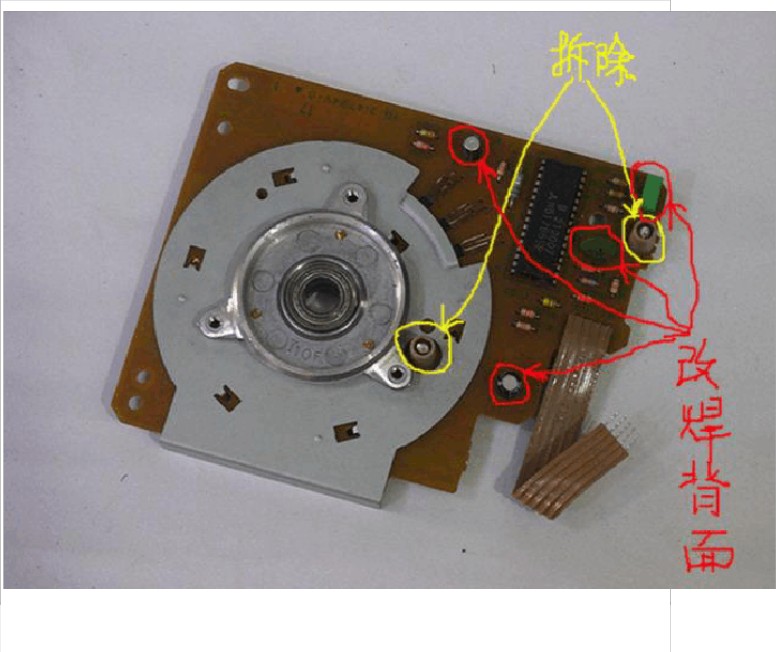


电机装配完成后

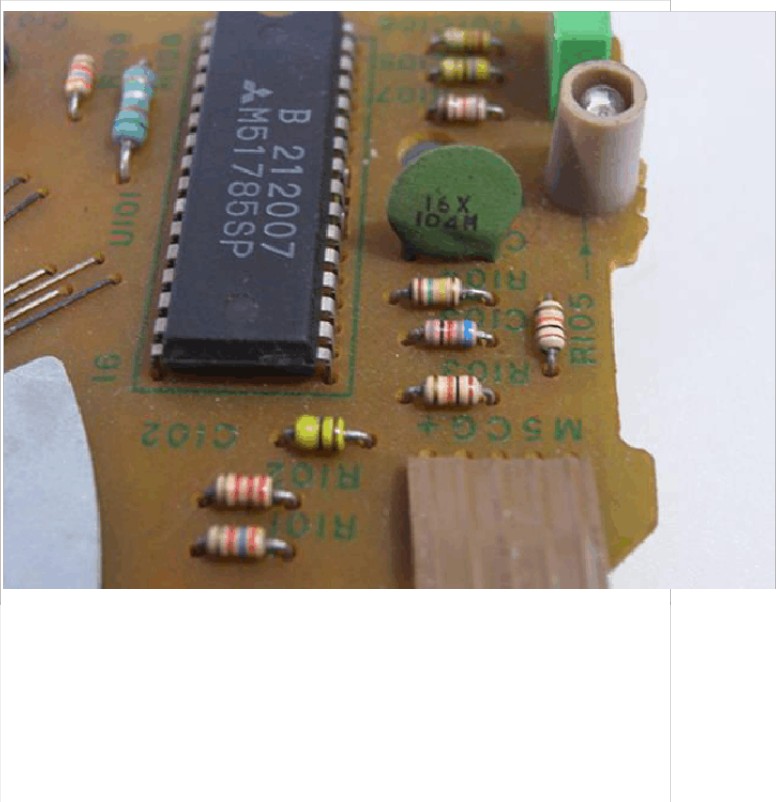
用两片铜片做的电刷



电刷装好后的侧面图将电路板上较突出的元件改焊在反面，



电机的电源接法。



从电路板标注的符号看，“＋”为电源正，“Ｇ”为电源负，

“Ｃ”与“Ｍ”端分别与电源正相连匀可使电机运转

将一张旧唱片按电机座的位置开孔，

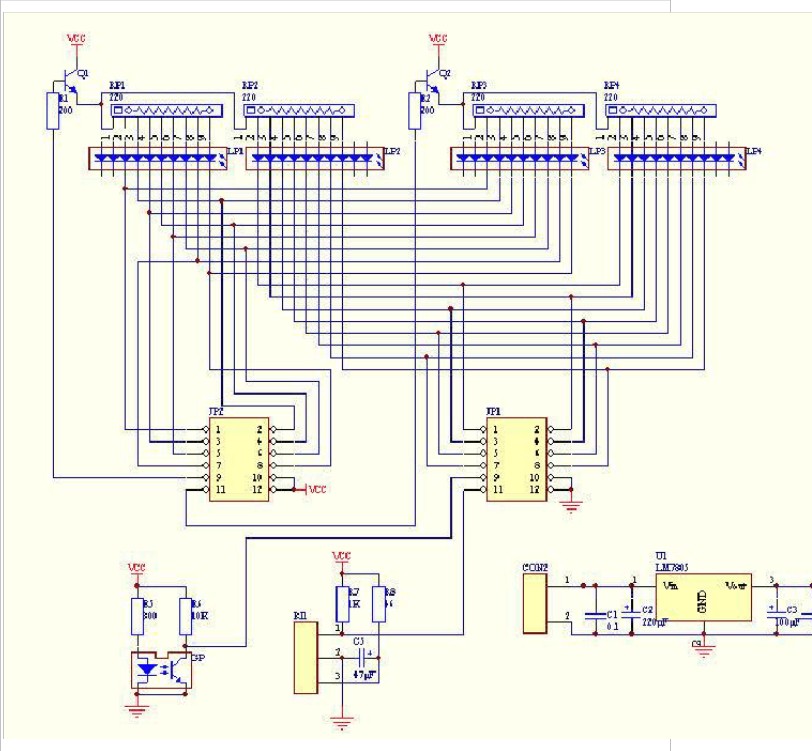
而定位用的挡光板应根据电路板上

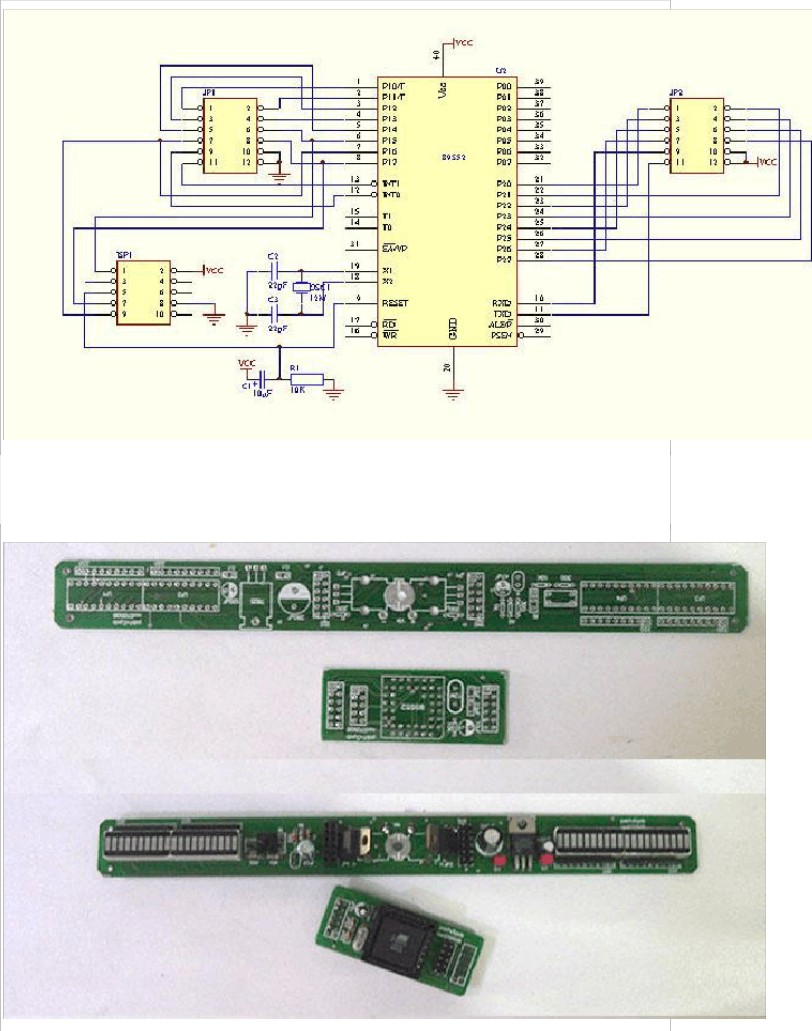
感光组件的位置确定。

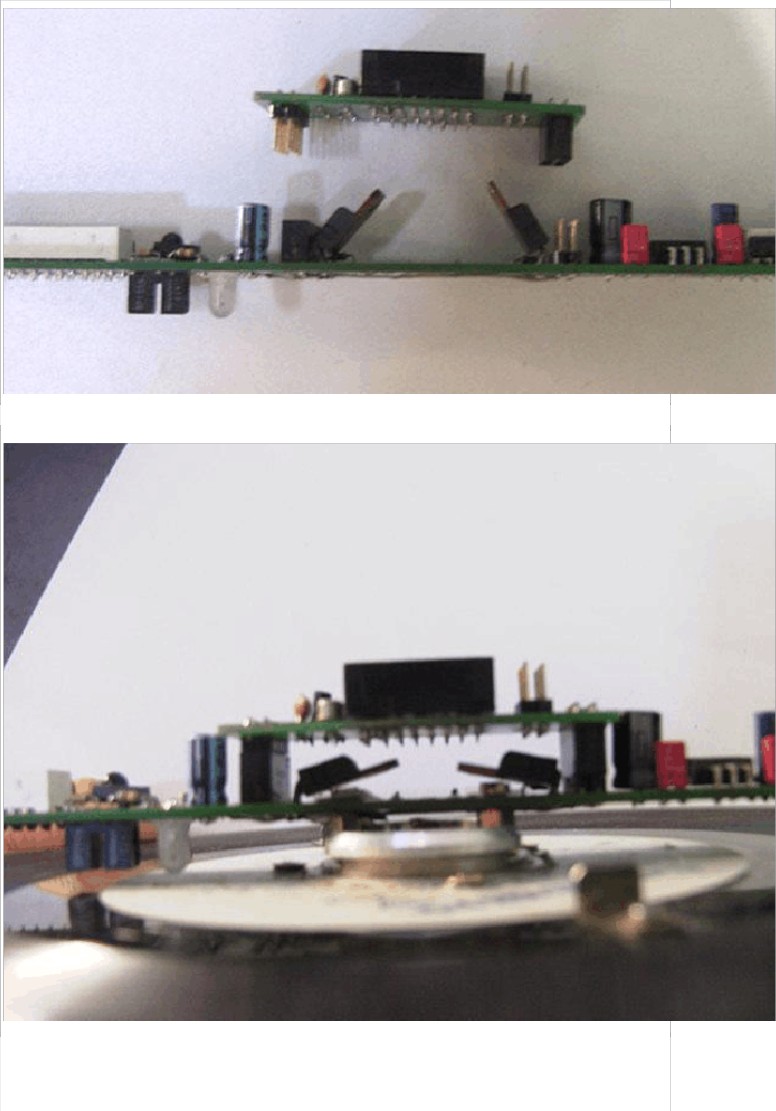


二。电路板的制作

本制作品用５１单片机控制，具体电原理图如下：



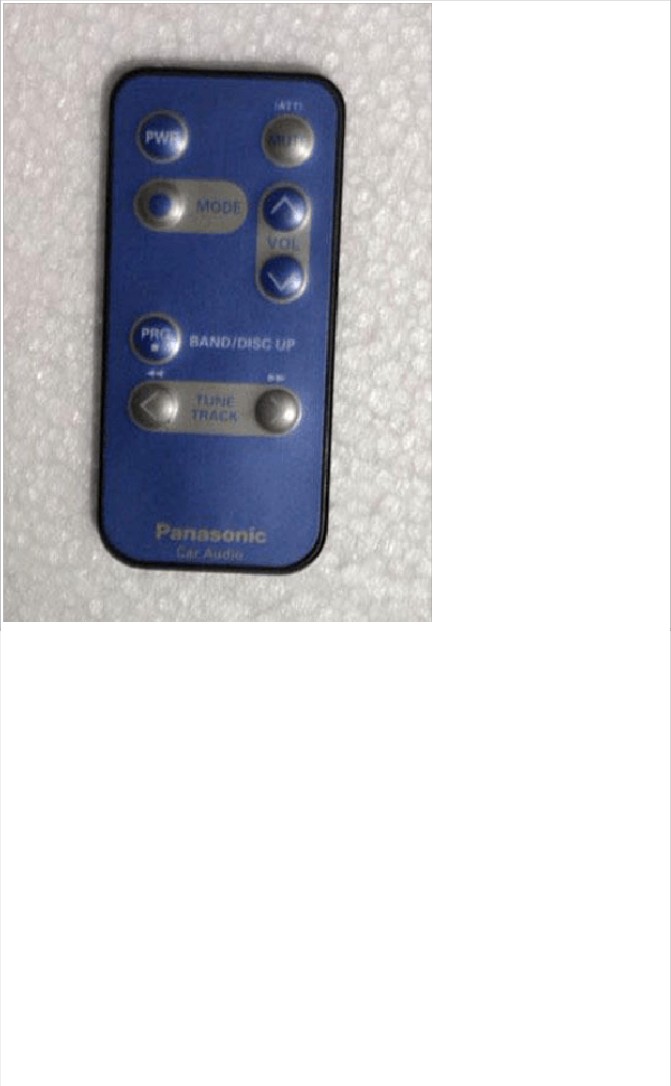
用 Protel 99 设计制作了电路板。



最后得到完成的作品。



遥控器用的是松下车载机的，只用了其中的六个键。

三。软件编程

因学的是Ｃ５１（不懂汇编）这里只提供Ｃ语言源程序，并限制为无遥控功能。也可提供带遥控功能的 hex 文件。

#include <reg51.h> sbit gate11=P3^0; sbit gate12=P3^1;



#define unit unsigned int #define uchar unsigned char

uchar data BUFFER[]={0,0,0,0,1,1,7};

uchar data M[]={31,28,31,30,31,30,31,31,30,31,30, 31};

uchar code NUM1[] =

{ 0x80,0x7F,0xC0,0xFF,0x40,0x90,0x40,0x8

C, // -0-

0x40,0x82,0xC0,0xFF,0x80,0x7F,0x00,0x00,

0x00,0x00,0x00,0x81,0x80,0x81,0xC0,0xF F, // -1-

0xC0,0xFF,0x00,0x80,0x00,0x80,0x00,0x00,

0x80,0xE0,0xC0,0xF0,0x40,0x98,0x40,0x8 C, // -2-

0x40,0x86,0xC0,0xC3,0x80,0xC1,0x00,0x00,

0x80,0x40,0xC0,0xC0,0x40,0x84,0x40,0x8

4, // -3-

0x40,0x84,0xC0,0xFF,0x80,0x7B,0x00,0x00,

0x00,0x0C,0x00,0x0E,0x00,0x0B,0x80,0x8

9, // -4-

0xC0,0xFF,0xC0,0xFF,0x00,0x88,0x00,0x00,



0xC0,0x47,0xC0,0xC7,0x40,0x84,0x40,0x8

4, // -5-

0x40,0x8C,0x40,0xFC,0x40,0x78,0x00,0x00,

0x00,0x7F,0x80,0xFF,0xC0,0x84,0x40,0x8

4, // -6-

0x40,0x84,0x00,0xFC,0x00,0x78,0x00,0x00,

0xC0,0x00,0xC0,0x00,0x40,0xF0,0x40,0xF

8, // -7-

0x40,0x0C,0xC0,0x07,0xC0,0x03,0x00,0x00,

0x80,0x7B,0xC0,0xFF,0x40,0x84,0x40,0x8

4, // -8-

0x40,0x84,0xC0,0xFF,0x80,0x7B,0x00,0x00,

0x80,0x03,0xC0,0x87,0x40,0x84,0x40,0x8

4, // -9-

0x40,0xC4,0xC0,0x7F,0x80,0x3F,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x6

3, // -:-

0x00,0x63,0x00,0x00,0x00,0x00,0x00,0x00,

};

uchar code NUM2[]=

{

0x00,0x00,0x7f,0x80,0xff,0xc0,0x82,0x4

0, // -0-

0x8c,0x40,0x90,0x40,0xff,0xc0,0x7f,0x8

0,

0x00,0x00,0x00,0x00,0x20,0x40,0x60,0

x40, // -1-

0xff,0xc0,0xff,0xc0,0x00,0x40,0x00,0x4

0,

0x00,0x00,0x41,0xc0,0xc3,0xc0,0x86,0

x40, // -2-

0x8c,0x40,0x98,0x40,0xf0,0xc0,0x60,0x

c0,

0x00,0x00,0x40,0x80,0xc0,0xc0,0x88,0

x40, // -3-

0x88,0x40,0x88,0x40,0xff,0xc0,0x77,0x

80,

0x00,0x00,0x0c,0x00,0x1c,0x00,0x34,0

x00, // -4-

0x64,0x40,0xff,0xc0,0xff,0xc0,0x04,0x4

0,

0x00,0x00,0xf8,0x80,0xf8,0xc0,0x88,0x

40, // -5-

80,

0x88,0x40,0x8c,0x40,0x8f,0xc0,0x87,0x

0x00,0x00,0x3f,0x80,0x7f,0xc0,0xc8,0x

40, // -6-

0x88,0x40,0x88,0x40,0x0f,0xc0,0x07,0x

80,

0x00,0x00,0xc0,0x00,0xc0,0x00,0x83,0

xc0, // -7-

0x87,0xc0,0x8c,0x00,0xf8,0x00,0xf0,0x

00,

0x00,0x00,0x77,0x80,0xff,0xc0,0x88,0x

40, // -8-

0x88,0x40,0x88,0x40,0xff,0xc0,0x77,0x

80,

0x00,0x00,0x70,0x00,0xf8,0x40,0x88,0

x40, // -9-

0x88,0x40,0x88,0xc0,0xff,0x80,0x7f,0x0

0,

0x00,0x00,0x00,0xc0,0x01,0x80,0x03,0

x00, // -/-

0x06,0x00,0x0c,0x00,0x18,0x00,0x30,0

x00,

};

unit disp1,disp2,key1,key2; unit ii,jj;

unit i,sw,xz;

void Delay(unit ms){

ms="ms"\*3; while(--ms);

}

void num\_led(int kk, int tt)

{

int jj; for(jj=0;jj<8;jj++){

gate11=key1; gate12=key2;

P2=~NUM1[kk+jj\*2];P1=~NUM1[kk+1+jj\*

2];

j\*2];

Delay(20); P1=0xff;P2=0xff;

gate11=key2; gate12=key1; P2=~NUM2[tt+15-jj\*2];P1=~NUM2[tt+14-j

Delay(20); P1=0xff;P2=0xff;

}

}

void display\_clock(void)

{

key1=!key1;key2=!key2; disp1=BUFFER[3]/10;disp2=BUFFER[4]-

(BUFFER[4]/10)\*10;

ii="disp1"\*16;jj=disp2\*16; num\_led(ii,jj); P2=0xff;P1=0xff;Delay(60);

ER[4]/10;

disp1=BUFFER[3]-disp1\*10;disp2=BUFF

ii="disp1"\*16;jj=disp2\*16; num\_led(ii,jj); P2=0xff;P1=0xff;Delay(60); ii="160";jj=160;

num\_led(ii,jj); P2=0xff;P1=0xff;Delay(60);

disp1=BUFFER[2]/10;disp2=BUFFER[5]- (BUFFER[5]/10)\*10;

ii="disp1"\*16;jj=disp2\*16; num\_led(ii,jj); P2=0xff;P1=0xff;Delay(60);

ER[5]/10;

disp1=BUFFER[2]-disp1\*10;disp2=BUFF

ii="disp1"\*16;jj=disp2\*16; num\_led(ii,jj); P2=0xff;P1=0xff;Delay(60); ii="160";jj=160;

num\_led(ii,jj); P2=0xff;P1=0xff;Delay(60); disp1=BUFFER[1]/10;disp2=BUFFER[6]-

(BUFFER[6]/10)\*10;

ii="disp1"\*16;jj=disp2\*16; num\_led(ii,jj); P2=0xff;P1=0xff;Delay(60);

disp1=BUFFER[1]-disp1\*10;disp2=BUFF

ER[6]/10;

ii="disp1"\*16;jj=disp2\*16; num\_led(ii,jj); P2=0xff;P1=0xff;Delay(60);

}

void timer0(void) interrupt 1 using 1

{

TMOD="0x11"; TH0=-(5000/256); TL0=-(5000%256); TR0=1;

BUFFER[0]=BUFFER[0]+1;

if (BUFFER[6]%4==0) M[1]=M[1]+1; if (BUFFER[0]>201+xz){

BUFFER[0]=0;

BUFFER[1]=BUFFER[1]+1; if (BUFFER[1]==60){

BUFFER[1]=0; BUFFER[2]=BUFFER[2]+1; if (BUFFER[2]==60){ BUFFER[2]=0; BUFFER[3]=BUFFER[3]+1;

if (BUFFER[3]==24){ BUFFER[3]=0;

BUFFER[4]=BUFFER[4]+1;

if (BUFFER[4]>M[BUFFER[5]-1]){ BUFFER[4]=1; BUFFER[5]=BUFFER[5]+1;

if (BUFFER[5]>12){ BUFFER[5]=1; BUFFER[6]=BUFFER[6]+1; if(BUFFER[6]>99) {

BUFFER[6]=0;

M[1]=M[1]-1;

}

}

}

}

}

}

}

}

void intersvr0(void) interrupt 0

{

sw=1;

}

void main(void)

{

xz="8";

key1=1;key2=0; EA="1";

for(;;){

}

IT0=1;EX0=1; IT1=1;ET0=1; TMOD="0x11";

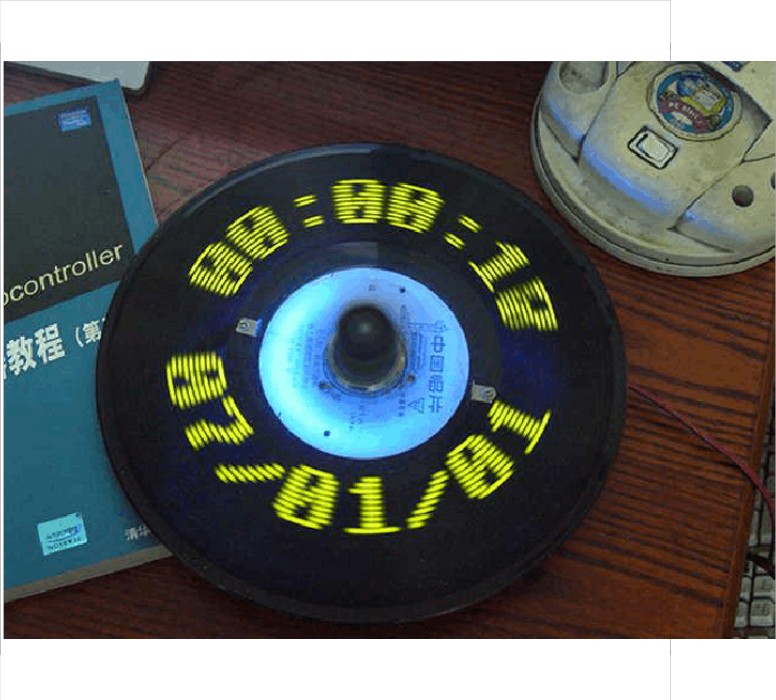
TH0=-5000/256; TL0=-5000%256; TR0=1;

Delay(10);

if(sw==1) {display\_clock();sw=0;}

}

四。运行效果

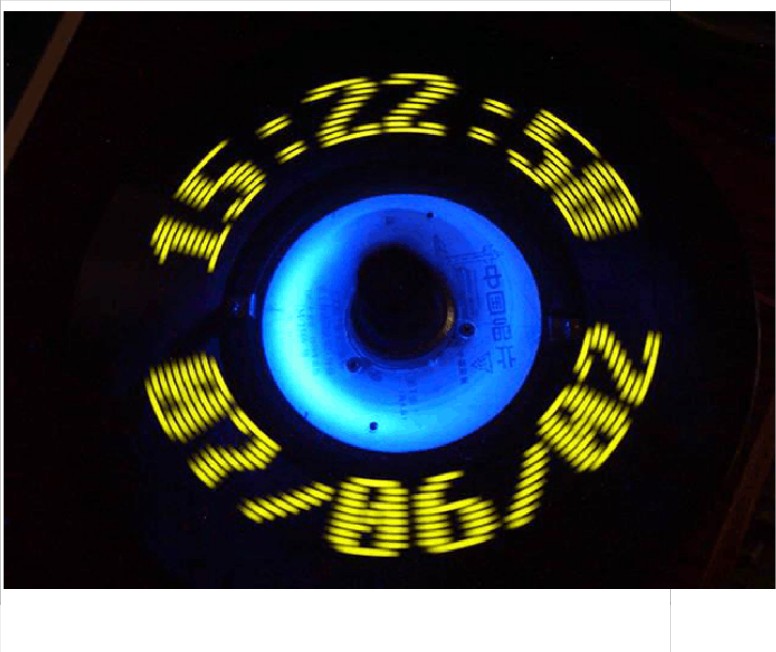


从开机，调整到最后完成。

开机时



调时状态



正常运行状态