```
void t0Server() interrupt 1
{
  static uint keyTimeCnt;

  //每30ms矩阵键盘扫描
  if(++keyTimeCnt==30)
  {
    keyTimeCnt=0;
    keyscan();
  }

  //系统状态
  switch(sysState)
  {
    case 0: //上电设定时间
    {
      static uint cursor=0; //光标位置

      if(keyVal!=55)
      {
        switch(keyVal)
        {
          case 12: timeNow[cursor]=1;keyVal=55;cursor++;break;
          case 13: timeNow[cursor]=2;keyVal=55;cursor++;break;
          case 14: timeNow[cursor]=3;keyVal=55;cursor++;break;
          case 22: timeNow[cursor]=4;keyVal=55;cursor++;break;
          case 23: timeNow[cursor]=5;keyVal=55;cursor++;break;
          case 24: timeNow[cursor]=6;keyVal=55;cursor++;break;
          case 32: timeNow[cursor]=7;keyVal=55;cursor++;break;
          case 33: timeNow[cursor]=8;keyVal=55;cursor++;break;
          case 34: timeNow[cursor]=9;keyVal=55;cursor++;break;
          case 43: timeNow[cursor]=0;keyVal=55;cursor++;break;
          default:break;
        }
      }
      if(cursor==6)
      {
        hh=timeNow[0]*10+timeNow[1];
        mm=timeNow[2]*10+timeNow[3];
        ss=timeNow[4]*10+timeNow[5];
        cursor=0;sysState=1;
        Write_Ds1302(0x8e,0);  //去写保护
        Write_Ds1302(0x84,timeNow[0]*16+timeNow[1]);
        Write_Ds1302(0x82,timeNow[2]*16+timeNow[3]);
        Write_Ds1302(0x80,timeNow[4]*16+timeNow[5]);  //写入时分秒
      }else
      {
        ledBuff[0]=ledChar[timeNow[0]]; ledBuff[1]=ledChar[timeNow[1]];
        ledBuff[2]=ledChar[10];
        ledBuff[3]=ledChar[timeNow[2]]; ledBuff[4]=ledChar[timeNow[3]];
        ledBuff[5]=ledChar[10];
        ledBuff[6]=ledChar[timeNow[4]]; ledBuff[7]=ledChar[timeNow[5]];
      }
    }break;

    case 1: //待机显示时间
    {
      Write_Ds1302(0x8e,0);  //去写保护
      hh=Read_Ds1302(0x85);
      mm=Read_Ds1302(0x83);
      ss=Read_Ds1302(0x81);  //读时分秒
      timeNow[0]=hh/16; timeNow[1]=hh%16;
      timeNow[2]=mm/16; timeNow[3]=mm%16;
      timeNow[4]=ss/16; timeNow[5]=ss%16;

      //刷新时间显示
      ledBuff[0]=ledChar[timeNow[0]]; ledBuff[1]=ledChar[timeNow[1]];
      ledBuff[2]=ledChar[10];
      ledBuff[3]=ledChar[timeNow[2]]; ledBuff[4]=ledChar[timeNow[3]];
      ledBuff[5]=ledChar[10];
      ledBuff[6]=ledChar[timeNow[4]]; ledBuff[7]=ledChar[timeNow[5]];

      //切换状态
      if(keyVal==13) {sysState=2;keyVal=0;}
      if(keyVal==14) {sysState=3;keyVal=0;}
      if(keyVal==22) {sysState=4;keyVal=0;}
      if(keyVal==44)
      {
        //时间重设
        sysState=0;keyVal=0;
        timeNow[0]=timeNow[1]=timeNow[2]=
            timeNow[3]=timeNow[4]=timeNow[5]=10;
      }
    }break;

    case 2: //显示温度
    {
      init_ds18b20();
      Write_DS18B20(0xcc);  //跳过ROM检测
      Write_DS18B20(0x44);  //启动t转换

      init_ds18b20();
      Write_DS18B20(0xcc);  //跳过ROM检测
      Write_DS18B20(0xbe);  //写读取指令

      tLSB=Read_DS18B20();  //先读取低位
      tMSB=Read_DS18B20();
      tempreatureGet=((tMSB<<8)+tLSB)>>4;
      if(tempreatureGet==85) {tempreatureGet=0;} //温度合法性检测

      //温度报警
      if(tempreatureGet>=25) {P2=0xa0;P0=0xf0;P2=0;}
      else {P2=0xa0;P0=0;P2=0;}
      led_set(tempreatureGet);

      //切换状态
      if(keyVal==12) {sysState=1;keyVal=0;}
      if(keyVal==14) {sysState=3;keyVal=0;}
      if(keyVal==22) {sysState=4;keyVal=0;}
    }break;

    case 3: //显示ADC
    {
      //初始化PCF8591
      IIC_Start();
      IIC_SendByte(0x90); //寻址
      IIC_WaitAck();
      IIC_SendByte(0x01); //AIN3
      IIC_WaitAck();

      //读取ADC值
      IIC_Start();
      IIC_SendByte(0x91); //寻址读
      IIC_WaitAck();
      voltageGet=IIC_RecByte();  //读取读数
      IIC_Stop();

      led_set(voltageGet);

      //切换状态
      if(keyVal==12) {sysState=1;P2=0x80;P0=0xff;P2=0;keyVal=0;}
      if(keyVal==13) {sysState=2;P2=0x80;P0=0xff;P2=0;keyVal=0;}
      if(keyVal==22) {sysState=4;P2=0x80;P0=0xff;P2=0;keyVal=0;}
    }break;

    case 4: //EEPROM记录切换模式次数
    {
      static uchar record;
      static uint i,j;

      //初始化AT24C02
      IIC_Start();
      IIC_SendByte(0xa0); //寻址
      IIC_WaitAck();
      IIC_SendByte(0x00);
      IIC_WaitAck();

      //读取
      IIC_Start();
      IIC_SendByte(0xa1); //寻址读
      IIC_WaitAck();
      record=IIC_RecByte();
      IIC_Ack(0);
      IIC_Stop();

      led_set(record);

      if(++i==100)  //0.1s写入延迟，最低写入间隔10ms
      {
        i=0;record++;
        //写入
        IIC_Start();
        IIC_SendByte(0xa0); //寻址
        IIC_WaitAck();
        IIC_SendByte(0x00);
        IIC_WaitAck();
        IIC_SendByte(record);
        IIC_WaitAck();
        IIC_Stop();
      }
```