

基于 P87LPC762 单片机的电子时间继电器

摘要：本文详细介绍了利用 P87LPC762 单片机设计电子时间继电器的方法。

该电子时间继电器在接线兼容传统的时间继电器，又可工作于有源模式，在微小定时上精度较传统时间继电器有很大提高。延时范围很大(超过 48 小时)，最小延时时间为 1ms。

一、硬件电路设计

(一) 系统结构分析

该电子时间继电器是以 P87LPC762 单片机为核心的应用系统。整个系统包括如下几部分：处理器 P87LPC762、继电器、电源和拨码盘。系统结构如图 1。

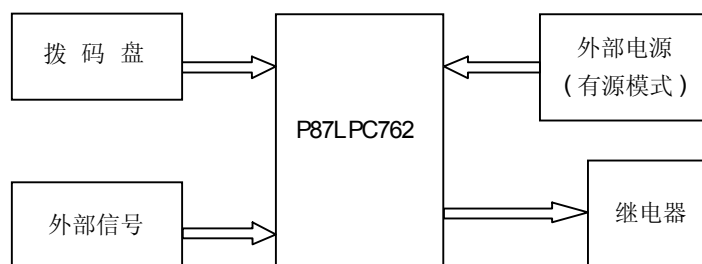


图 1 系统结构框图

(二) 电路原理图设计

根据系统结构，可把整个电路分成 3 部分：CPU 部分(包括拨码盘输入)、电源和继电器电路。分别作如下介绍。

1. CPU 部分

这部分包括单片机所需要的外围电路和拨码盘输入电路，如图 2 所示。

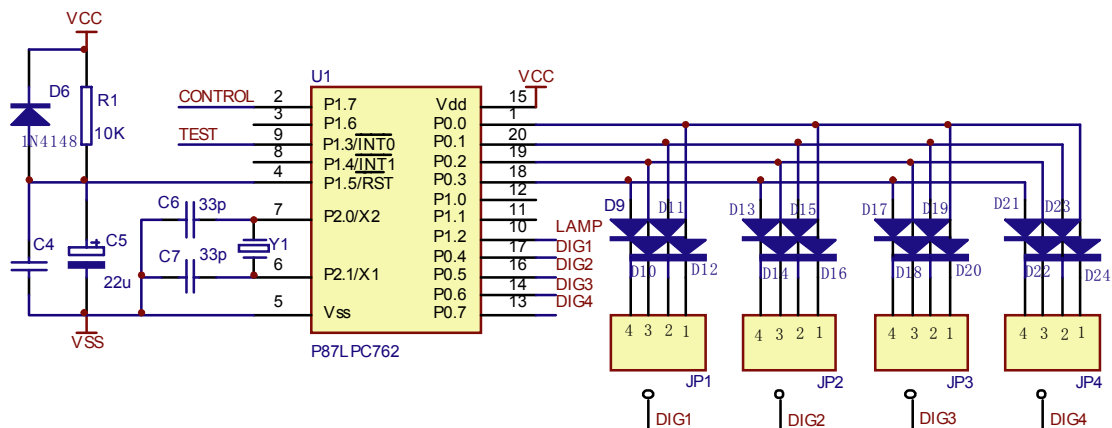


图 2 CPU 部分电路图

单片机 I/O 所能提供的输出电流较弱，而所能吸收的灌电流较大，为了系统工作的稳定，拨码盘采用了低电平选通，同时在拨码盘的输出口线上接了二极管，是为了防止各个拨码盘之间的干扰，确保能够准确读出每位的值。

2. 电源部分

为了使该时间继电器能够在有源和无源模式下工作，我们采用如图 3 所示的电源电路。

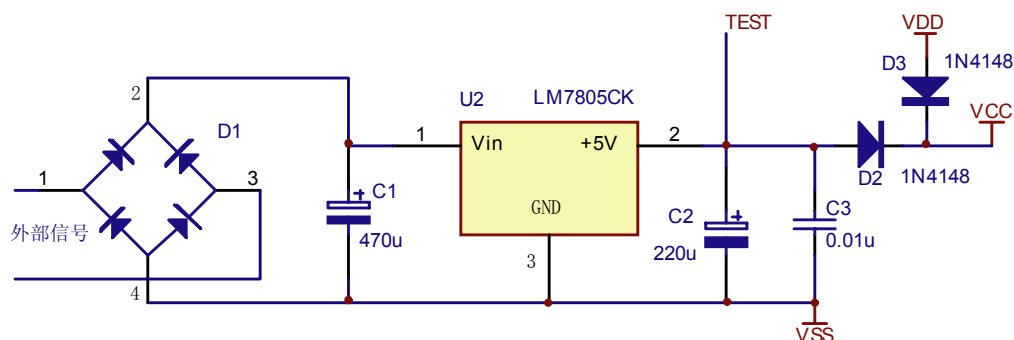


图 3 电源电路

外部信号接到电桥 D1 的 1、3 脚。需要把外部信号降压(5~10V)才能接入。VDD 是有源模式时的外部电源。

3. 继电器电路

这是该时间继电器的输出部分，电路如图 4。

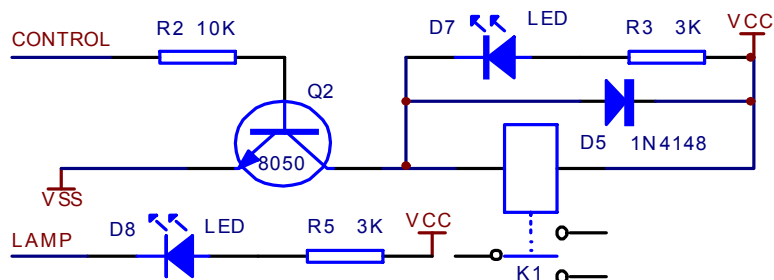


图 4 继电器电路

其中 D5 是为了保护继电器；D7 是继电器工作状态指示灯，继电器工作时，D7 亮。

为了用户能够直观的感到系统在计数，我们加上了指示灯 D8 做为计数指示灯，当系统在计数的时候，D8 会不停的闪烁，计数完毕，D8 停止闪烁。

二、软件设计

(一) 程序流程分析

1. 整体分析

先看有源模式下的情况。系统一旦上电，则读入用户所设定的值，等待外部信号的到来；延时时间到，则关闭继电器；若外部信号离开，则系统复位，重新等待外部信号。工作流程如图 5。

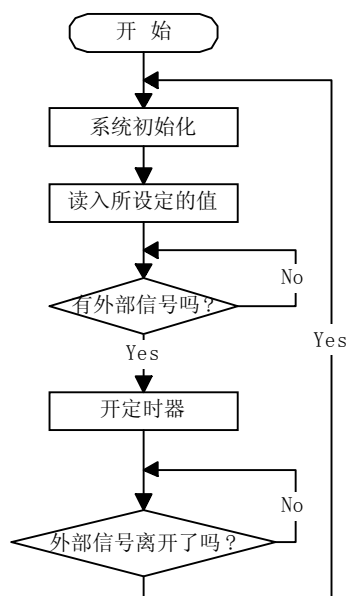


图 5 有源模式下工作流程图

至于无源模式下，工作流程更简单(仅是有源模式下的一部分)，外部信号充当系统的电源，一旦外部信号到来，系统就初始化，同时开始计时，时间到则关闭继电器。

2. 中断服务程序

这是一个计时系统，我们安排定时器 0 来计时。定时器开启后，每隔一定时间(程序中设定为 1ms)把计数变量加 1，当计数到和用户所设定的值相同时，则停止计数，同时关闭继电器。流程图如图 6。

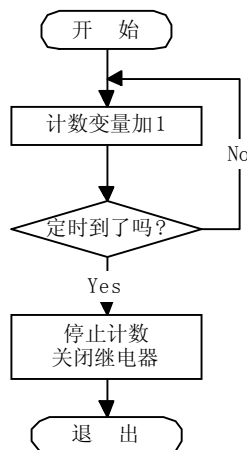


图 6 定时器 0 计时流程图

3. 读数处理

为了扩大延时范围，我们采用指数的形式。一共安排了 4 个拨码盘 D0~D3，我们设前位数据 D0~D2 为基数，第 4 位数据 D3 为指数，所设定的数据则为 $D_0D_1D_2 \times 10^{D_3}$ ，如分别设定 2、3、1、5，则所设定的延时时间为 $231 \times 10^5 \text{ms}$ 。最小延时单位为 1ms，最大可到 $999 \times 10^9 \text{ms}$ 。

(二) 程序清单

```

#include <REG764.H>
#define uchar unsigned char
  
```

```

#define CLOCK_1MS (65536-1000)

sbit CONTROL=P1^7;           //继电器控制口
sbit TEST=P1^3;             //外部信号测试口
sbit LAMP=P1^2;             //指示灯
bit stop=0;                 //停止标志

float base;                  //基数
uchar number[4];            //读入数据存放缓冲区
uchar pos[]={0xef,0xdf,0xbf,0x7f}; //拨码盘位选控制字

float TimeSet;               //所设定的定时时间
float TimeCount;            //定时器计数
int count;                   //指示灯计数器

/*****//
//      初始化函数      //
/*****//
void Init()
{
//I/O 口线设置=====
//P0 口准双向口
P0M1=0x00;                  //P0M1=00000000B
P0M2=0xf0;                  //P0M2=11110000B
//P1.2,P1.7 上拉输出
P1M1=0x08;                  //P1M1=00001000B
P1M2=0x84;                  //P1M2=10000100B
P2M1=0x40;                  //使能 P1 口施密特输入,(P1.3 施密特输入)
//中断设置=====
EA=1;                        //总中断允许
ET0=1;                       //定时器 0 中断允许
//变量初始化=====
P0=0x00;
TimeCount=0;                 //定时器计数初值为 0
CONTROL=0;
}

/*****//
//      定时器 0 中断函数,计数      //
/*****//
void Timer0() interrupt 1
{
TL0=CLOCK_1MS&0xff;         //定时器常数重载
TH0=CLOCK_1MS>>8;

```

```

TimeCount++;          //定时计数器
if(TimeCount>=TimeSet)
{
    CONTROL=0;        //关闭继电器
    TR0=0;            //停止计数
    TimeCount=0;      //定时计数恢复为 0
    LAMP=1;           //熄灭指示灯
    stop=1;
}

count++;             //指示灯计数器
if(count>=1000&&LAMP==0) //指示灯闪烁,表示正在计数中
    { LAMP=1;count=0; }
if(count>=1000&&LAMP==1)
    { LAMP=0;count=0; }
}

/*****//
//a*10^b 运算函数,float 型,返回值 y //
/*****//
float power(float base,uchar power)
{
    uchar i;
    float y;

    if(power==0)          //幂为 0 则等于基数
        y=base;

    else
    { y=base*10;          //否则乘 10
      for(i=1;i<power;i++) //循环
          y=y*10; }

    return(y);           //返回
}

/*****//
//      读入数据函数 //
/*****//
void ReadNum()
{
    uchar temp,i;

    for(i=0;i<4;i++)     //读入所设定的数据

```

```

    {
        P0=0x00;
        P0=pos[i];           //选中拨码盘
        temp=P0;
        number[i]=(~temp)&0x0f; //存放在 number[]中
    }

    base=number[0]*100+number[1]*10+number[2]; //计算基数
    TimeSet=power(base,number[3]);           //计算出所设定的时间
}

//*****//
//          主函数                      //
//*****//
main()
{
    Init();           //系统初始化

    ReadNum();       //读入所设定的数据

    while(1)
    {
        if((TEST==1)&&(stop==0)) //如果有外部信号而没有终止信号,就开始计数
        {
            CONTROL=1;           //打开继电器
            if(!TR0)             //如果计数器还没有打开
                { TR0=1; }       //打开计数器,开始计数
        }
        if(TEST==0)
        {
            TR0=0;               //外部信号离开,停止计数
            stop=0;              //清除正常停止信号
            CONTROL=0;
            TimeCount=0;         //计数恢复为 0

            ReadNum();           //重新读入所设定数据
        }
    }
}

```

三、测试现象及使用说明

(一) 测试现象

1. 有源模式

当外部信号到来的时候，立即开启继电器，计数指示灯闪烁，到了所设定的延时后，立即关闭继电器。若此时去掉外部信号之后再接入外部信号，系统重新工作。在这过程中，用户可以重新设定延时时间。

2. 无源模式

当外部信号到来，系统工作，继电器开启，计数指示灯闪烁，到所设定时间后停止继电器。

(二) 使用说明

无论在工作在何种模式下，必须先设定好延时时间才能接入外部信号。

若需要进行微小延时，最好工作在有源模式下，以提高延时精度；若延时时间比较长，则两种模式下均可。