

基于 RTX51 Tiny 的步进电机控制器的设计

Design of Stepping Motor Controller Based on RTX51 Tiny

冷雪锋 （常州轻工职业技术学院电子电气工程系,江苏 常州 213164）

摘 要

研究了用小型多任务实时操作系统 RTX51 Tiny 开发的步进电机控制器。给出了系统的硬件及软件设计方案。对于多任务同时运行的复杂控制系统,使用 RTX51 Tiny 更能符合思维设计习惯,降低了软件开发的复杂性,提高了软件开发的效率。同时,用实时操作系统 RTX51 Tiny 开发的步进电机控制器,在系统的实时性、稳定性和可靠性等方面都得到了提高。

关键词:实时操作系统,RTX51 Tiny,步进电机,控制器

Abstract

A stepping motor controller based on RTX51 Tiny,which is a miniature multitasking and real-time operating system is developed and studied.The system design scheme of hardware and software is given.For the complex control system in which the multitask runs at the same time,the use of RTX51 Tiny is more in line with the habit of thought,reducing the complexity of software development,improving the efficiency of software development.At the same time,applying RTX51 Tiny in the stepping motor controller,it enhances the real-time character,stability and reliability of system.

Keywords:real-time operating system,RTX51 Tiny,stepping motor,controller

步进电机是一种将电脉冲信号转换为角位移的执行机构,通过机械传动可以把角度转换成直线位移^[1]。文献[1]是基于 PC 机环境开发出用于步进电机的软控制器。文献[2-3]是用 C 语言的 C51 开发的步进电机控制器。文献[4]设计的步进电机控制器中,PC 机采用 8086/8088 汇编语言编写,AT89C51 单片机端采用 MCS-51 语言编写。而本文是用实时操作系统 RTX51 Tiny 开发的步进电机控制器,由于 RTX51 Tiny 是 Keil uVision 中自带的一个小型嵌入式 RTOS,具有小巧、速度快、系统开销小、使用方便等优点^[5],所以用 RTX51 Tiny 开发的步进电机控制器,具有较强的实时性、稳定性和可靠性,且维护方便。

1 硬件电路设计

Proteus 仿真电路如图 1 所示。主要由 4 个模块组成:单片机模块、信号隔离模块、信号放大模块、步进电机模块。

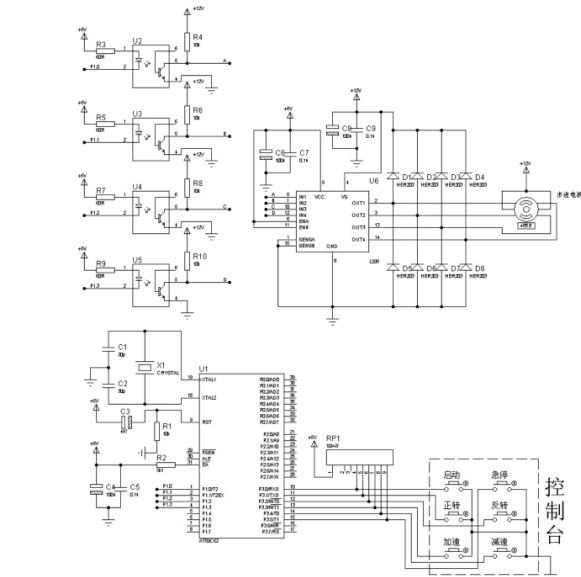


图 1 Proteus 仿真电路图

1.1 信号隔离模块

由于系统中步进电机对单片机产生干扰,不能把单片机产生的控制信号直接连在步进电机上,为了提高系统的可靠性,避免电机对数字电路的影响。在本系统中,采用四个 TLP521-1 光电耦合器。

1.2 信号放大模块

由于步进电机工作需要较大的功率,所以通常需要使用功率放大器来提供步进电机的工作电流。本设计中采用 L298 芯片将信号放大,进而驱动步进电机的各个相。

1.3 步进电机模块

在图 1 的电路图中,系统采用四相步进电机,AT89C52 单片机的 P1 口的 P1.0~P1.3 引脚输出的具有时序的方波,经光电耦合器和 L298 芯片作为步进电机四个相 A~D 的控制信号。步进电机采用单双 8 拍方式,当电机绕组通电时序为 A-AB-B-BC-C-CD-D-DA 时为正转,通电时序为 DA-D-CD-C-BC-B-BA-A 时为反转。

2 RTX51 Tiny 软件设计

为了防止步进电机在启动时产生丢步或堵转现象,启动时需要进行加速控制;为了防止在停止时产生过冲现象,在停止时需要进行减速控制^[3]。

2.1 系统可以实现功能

- 1)通过“启动”键设定电机启动模式;
- 2)通过“正转”和“反转”按键设定电机转向的模式;
- 3)通过“加速”按键步进电机缓慢加速启动,加速一段时间后,步进电机自动匀速运转;
- 4)通过“减速”按键,能缓慢减速,可减到停止;
- 5)在减速过程中,可以通过“正转”和“反转”按键切换正转或反转模式。然后再通过“加速”按键,步进电机缓慢加速,加速一段时间后,步进电机自动匀速运转;
- 6)在任一过程中,都可通过按“急停”按键,停止步进电机;
- 7)减速到停止后,或者急停后,都可通过“启动”键设定电机重新启动。

2.2 系统任务分配

- 1) Task1 ANYKey: 键处理任务;
- 2) Task2 motor_fr_1: 正转加速任务;
- 3) Task3 motor_fr_2: 正转匀速任务;
- 4) Task4 motor_fr_3: 正转减速任务;
- 5) Task5 motor_rr_4: 反转加速任务;
- 6) Task6 motor_rr_5: 反转匀速任务;
- 7) Task7 motor_rr_6: 反转减速任务;
- 8) 自定义函数 Key: 键判断子程序。

2.3 系统基本源程序

```
#include <REG52.h>
#include <RTX51TNY.H>
bit Start=0;           //步进电机起动变量
bit fr=0;              //步进电机正转变量
bit rr=0;              //步进电机反转变量
bit speedup=0;         //步进电机加速变量
bit slowdown=0;        //步进电机减速变量
unsigned char speed=0;  //步进电机速率变量
unsigned char KValue;   //键值变量
unsigned char ppulse [8]={0xf1,0xf3,0xf2,0xf6,0xf4,0xfc,0xf8,0xf9};
//正转脉冲
unsigned char npulse [8]={0xf9,0xf8,0xfc,0xf4,0xf6,0xf2,0xf3,0xf1};
//反转脉冲
/*****
/*          键判断子程序          */
*****/
unsigned char Key(void)
{
    .....
}
/*****
/*          任务 0: 键处理任务          */
*****/
void ANYKey(void)_task_ 0
{
    os_create_task(1);    //创建任务 1
    os_create_task(2);    //创建任务 2
    os_create_task(3);    //创建任务 3
    os_create_task(4);    //创建任务 4
    os_create_task(5);    //创建任务 5
    os_create_task(6);    //创建任务 6
    while(1)
    {
        for(;;)
        {
            KValue=Key();    //调键判断子程序,返回键值
            if(KValue!=0xff)
                break;
        }
        switch(KValue)    //判断是哪个键按下
        {
            case 0xfe:Start=1;    //启动按钮
                break;
            case 0xfd:fr=1;rr=0;    //正转按钮
                break;
            case 0xfb:rr=1;fr=0;    //反转按钮
                break;
```

```
            case 0xf7:speedup=1;slowdown=0; //加速按钮
                break;
            case 0xef:slowdown=1;speedup=0; //减速按钮
                break;
            case 0xdf:Start=0;    //急停按钮
                break;
        }
        if(Start)    //键处理
        {
            if(fr&speedup)
            {
                os_send_signal(1);    //发送 Signal 信号,激活进程 1
                os_wait(K_SIG,0,0);    //等待信号
            }
            else if(fr&slowdown)
            {
                os_send_signal(3);    //发送 Signal 信号,
                //激活进程 3
                os_wait(K_SIG,0,0);    //等待信号
            }
            else if(rr&speedup)
            {
                os_send_signal(4);    //发送 Signal
                //信号,激活进程 4
                os_wait (K_SIG,0,0);    //等待信号
            }
            else if(rr&slowdown)
            {
                os_send_signal(6);    //发送 Sig-
                //nal 信号,激活进程 7
                os_wait(K_SIG,0,0);    //等待信号
            }
            else
                P1=0xf0;    //静止
        }
        else if(Start==0)    //急停
        {
            P1=0xf0;
            fr=0;
            rr=0;
            speedup=0;
            slowdown=0;
        }
        KValue=Key();
    }
}
/*****
/*          任务 1: 正转加速任务          */
*****/
void motor_fr_1(void)_task_ 1
{
    .....
    while(1)
    {
        os_wait(K_SIG,0,0);
        .....
    }
}
```

(下转第 22 页)

外界扰动或内部设定值变动时,系统将发生动态变化,而这种变化是从系统的稳态值开始的。因此,可以先根据活性污泥过程的基本性质,选取一组各组分浓度的初值,然后在一定的输入数据和参数数值下进行运算,获得在该输入和参数的条件下系统的一组的稳态值。该组稳态值即可作为新的初值,代入方程组,然后变化输入(扰动或设定变化)或参数,对该系统进行研究。并把动态仿真结果与 Bsm1 稳态模拟计算结果进行比较。

模型求解采用五阶龙格库塔法,选用了 S_1, X_S 动态模拟结果分别显示于图 3、图 4 和表 1。

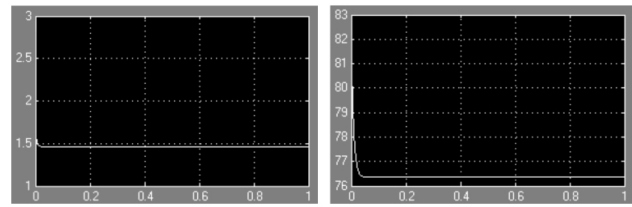


图 3 S_1 组分浓度动态响应曲线

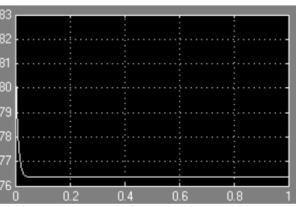


图 4 X_S 组分浓度动态响应曲线

表 1 Bsm1 与生化反应池结果比较

	Bsm1	Results	Unit		Bsm1	Results	Unit
S_1	30	30	g/m^3	S_0	0.0000631	0.00006433	g/m^3
S_2	1.46	1.458	g/m^3	S_{NO}	3.66	3.662	g/m^3
X_1	1149	1149	g/m^3	S_{NH}	8.34	8.348	g/m^3
X_2	76.4	76.36	g/m^3	S_{NO}	0.882	0.8832	g/m^3
$X_{B,II}$	2553	2554	g/m^3	X_{NO}	5.03	5.025	g/m^3
$X_{B,A}$	148	147.9	g/m^3	S_{ALK}	5.08	4.995	mol/m^3
X_P	450	449.7	g/m^3	Q_1	92230	92230	m^3/day

5.5 模型计算结果及分析

1)从图 3 和图 4 可知,反映了过程的变化。模拟流程由一个

.....

(上接第 19 页)

```
os_send_signal(2);
}
}
/*****
/*          任务 2:正转匀速任务          */
/*****
void motor_fr_2(void)task_ 2
{
    .....
    while(1)
    {
        os_wait(K_SIG,0,0);
        .....
        os_send_signal(0);
    }
}
/*****
/*          任务 3:正转减速任务          */
/*****
void motor_fr_3(void)task_ 3
{
    .....
    while(1)
    {
        os_wait(K_SIG,0,0);
```

生化反应池组成,反应产物作为二沉池进料。动态实验是对反应池的进料速率施加一阶跃性干扰,保持其他条件不变,对生化反应池进行动态仿真,模拟反应器出口浓度的动态响应行为。当出水浓度满足指定精度时,认为系统达到稳态。

2) 仿真计算结果与 Bsm1 稳态仿真结果对比如表 1 所示。模型计算结果与 Bsm1 计算结果基本吻合。

6 结束语

MATLAB 污水处理单元操作工具箱能使开发者在 Windows 环境下,通过简单的鼠标拖放操作建立较复杂的污水处理流程,模拟结果可以直接以曲线图的形式显示。本工具箱的出现为污水处理过程可视化,面向对象建模探索了一条新路。

本工具箱的成功开发证明了 MATLAB 语言可以有效应用于污水处理过程的模拟和仿真软件的开发。MATLAB 语言在仿真模型的建立、求解、模型结果的图形处理等方面有强大的功能。因此,探讨 MATLAB 在污水处理领域中的应用是非常有前途的。

参考文献

[1]张志涌,刘瑞桢,杨祖英.精通和掌握 Matlab[M].北京:北京航空航天大学出版社,1998
[2]M Henze, C P L Grady, W Gujer, et al. IAWPRC scientific and technical reports No 1: Activated sludge model No1[R]. IAWPRC:Task Group on Mathematical Modeling for design and operation of biological wastewater treatment,1986
[3]国际水协废水生物处理设计与运行数学模型课题组.活性污泥数学模型[M].张亚雷,李咏梅,译.上海:同济大学出版社,2002

[收稿日期:2009.12.9]

```
.....
os_send_signal(0);
}
.....
```

由于篇幅所限,文中仅给出部分任务的程序。

3 结束语

本文介绍了基于 RTX51 Tiny 的步进电机控制器的设计。对 51 系列单片机来说,RTX51 Tiny 是一个比较好的多任务实时操作系统,将其应用到步进电机控制器的设计中,大大提高了系统的稳定性和可靠性;同时,由于采用多任务设计,软件的模块化程度高,降低了软件开发的复杂性,提高了软件开发的效率。

参考文献

[1]麦志忠,高军礼.步进电机软控制器的设计[J].广东工业大学学报,2008,25(1):73-76
[2]张丹红,文小玲,程昌银.一种基于单片机的步进电机控制器的研制[J].武汉化工学院学报,2000,22(3):62-64
[3]李世忠,雷秀.基于单片机的步进电机加减速的控制方法[J].内蒙古科技与经济,122-124
[4]李会艳,胡荣强.基于单片机的步进电机控制系统[J].自动化技术与应用,2007,26(11):75-77
[5]邵子扬.RTX51 Tiny 中容易混淆的问题[J].单片机与嵌入式系统应用,2004:61-62

[收稿日期:2009.9.29]