4.5独立式键盘控制模块

4.5.1键盘的功能及分类

键盘是一种最常用的输入设备,它是一组按键的集合，从功能上可分为数字键和功能键两种，作用是输入数据与命令，查询和控制系统的工作状态，实现简单的人机对话。

键盘的分类

（a）键盘按照接口原理可分为编码键盘与非编码键盘两类。这两类键盘的主要区别是识别键符及给出相应键码的方法。

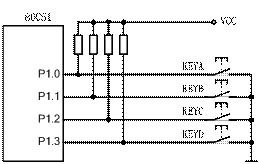
    编码键盘主要是用硬件来实现对键的识别；

    非编码键盘主要是由软件来实现键盘的定义与识别。

（b）键盘按照其结构可分为独立式键盘与矩阵式键盘两类。

    独立式键盘主要用于按键较少的场合,矩阵式键盘主要用于按键较多的场合，也称行列式键盘

4.5.2独立式键盘

独立式键盘的按键相互独立，每个按键接一根I/O口线，一根I/O口线上的按键工作状态不会影响其它I/O口线的工作状态。因此，通过检测I/O口线的电平状态，即可判断键盘上哪个键被按下

4.5.3独立式键盘与单片机的链接

2 系统论述

2．1设计背景

近年来，随着科技的进步，电力电子技术得到了迅速的发展，直流电机得到了越来越广泛的应用。直流它具有优良的调速特性,调速平滑、方便,调速范围广;过载能力大,能承受频繁的冲击负载,可实现频繁的无级快速起动、制动和反转;需要能满足生产过程自动化系统各种不同的特殊运行要求，从而对直流电机的调速提出了较高的要求，改变电枢回路电阻调速，改变电枢电压调速等技术已远远不能满足要求，这时通过PWM方式控制直流电机调速的方法应运而生。

2．2 设计思路

直流电机PWM控制系统的主要功能包括：实现对直流电机的加速、减速以及电机的正转、反转和急停，并且可以调整电机的转速，能够很方便的实现电机的智能控制。

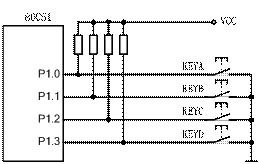
主体电路：即直流电机PWM控制模块。这部分电路主要由AT89S52单片机的I/O端口、定时计数器、外部中断扩展等控制直流电机的加速、减速以及电机的正转和反转，并且可以调整电机的转速，能够很方便的实现电机的智能控制。其间是通过AT89S52单片机产生脉宽可调的脉冲信号并输入到L298驱动芯片来控制直流电机工作的。该直流电机PWM控制系统由以下电路模块组成：

设计输入部分：这一模块主要是利用带中断的独立式键盘来实现对直流电机的加速、减速以及电机的正转、反转和急停控制。

设计控制部分：主要由AT89S52单片机的外部中断扩展电路组成。直流电机PWM控制实现部分主要由一些二极管、电机和L298直流电机驱动模块组成。

设计显示部分： LED数码显示部分，实现对PWM脉宽调制占空比的实时显示。

2．3 系统框架设计



直流电机PWM调速方案

方案说明：直流电机PWM调速系统以AT89S52单片机为控制核心，由命令输入模块、LED显示模块及电机驱动模块组成。采用带中断的独立式键盘作为命令的输入，单片机在程序控制下，定时不断给L298直流电机驱动芯片发送PWM波形，H型驱动电路完成电机正，反转和急停控制；同时单片机不停的将PWM脉宽调制占空比送到LED数码管完成实时显示。

3 PWM脉宽调制原理

3．1 PWM调速原理

PWM（脉冲宽度调制）是通过控制固定电压的直流电源开关频率，改变负载两端的电压，从而达到控制要求的一种电压调整方法。PWM可以应用在许多方面，比如：电机调速、温度控制、压力控制等等[7]。

在PWM驱动控制的调整系统中，按一个固定的频率来接通和断开电源，并且根据需要改变一个周期内“接通”和“断开”时间的长短。通过改变直流电机电枢上电压的“占空比”来达到改变平均电压大小的目的，从而来控制电动机的转速。也正因为如此，PWM又被称为“开关驱动装置”。

如图1所示：

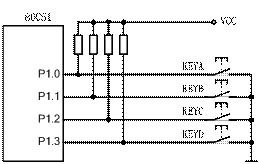
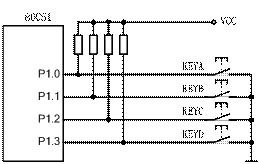


图1  PWM信号的占空比

设电机始终接通电源时，电机转速最大为Vmax，设占空比为D= t1 / T，则电机的平均速度为Va = Vmax \* D，其中Va指的是电机的平均速度；Vmax 是指电机在全通电时的最大速度；D = t1 / T是指占空比。

由上面的公式可见，当我们改变占空比D=t1/T时，就可以得到不同的电机平均速度Vd,从而达到调速的目的。严格来说，平均速度Vd与占空比D并非严格的线性关系，但是在一般的应用中，我们可以将其近似的看成是线性关系。



致谢

经过近半学期的忙碌和工作，本次毕业设计已经接近尾声，作为一个本科生的毕业设计，由于经验的匮乏，难免有许多考虑不周全的地方，如果没有导师的督促指导，以及一起学习的同学们的支持，想要完成这个设计是难以想象的。在这里首先要感谢我的指导老师陈老师。陈老师平日里工作繁多，但在我做毕业设计的每个阶段，从查阅资料到设计草案的确定和修改，中期检查，后期详细设计，装配草图等整个过程中都给予了我悉心的指导。我的设计较为复杂烦琐，但是陈老师仍然细心地纠正设计内容中的错误。除了敬佩陈老师的专业水平外，他的治学严谨和科学研究的精神也是我永远学习的榜样，他的循循善诱的教导和不拘一格的思路也给予我无尽的启迪。并将积极影响我今后的学习和工作。其次要感谢我的同学对我无私的帮助，特别是在直流电机控速技术的相关信息方面，正因为如此我才能顺利的完成设计。最后我要感谢我的母校——徐州工程学院，是母校给我们提供了优良的学习环境；另外，我还要感谢那些曾给我授过课的每一位老师，是你们教会我专业知识。在此，我再说一次谢谢！谢谢大家！！！。

参考文献

[1] 林志琦.基于Proteus的单片机可视化软硬件仿真[M].北京:北京航空航天大学出版    社,2006.9

[2] 周润景,张丽娜.基于PROTEUS的电路及单片机系统设计与仿真[M].北京:北京航空航天大学出版社,2006.5

[3] 张靖武,周灵彬.单片机系统的PROTEUS设计与仿真[M].北京:电子工业出版社,2007.4

[4] 周润景,张丽娜.PROTEUS入门实用教程[M].北京:机械工业出版社,2007.9

[5] 楼然苗,李光飞.51系列单片机设计实例[M].北京:北京航空航天大学出版社,2003.3

[6] 楼然苗,李光飞.单片机课程设计指导[M].北京:北京航空航天大学出版社,2007.7

[7] 贾东耀，汪仁煌. 数字温度传感器在仓库温度检测系统的应用[J]. 传感器世界，2001

[8] DALLAS DS18B20数据手册［Z］.http：//www.maximic.com、

[9］李朝青．单片机原理及接口技术［M］．北京：北京航空航天大学出版社，2006．

[10］康华光，陈大钦．电子技术基础（模拟部分）［M］．北京：高等教育出版社，1999．

[11］康华光，邹寿彬．电子技术基础（数字部分）［M］．北京：高等教育出版社，2000．

[12］胡祝兵．基于单片机控制的直流电机调速系统的设计［J］．承德石油高等专科学校学报，2008（1）．

[13］ 卢春华，姚海燕，张莉. 基于单片机的直流电机调速系统设计［J］.硅谷，2009（20）.

[14] 吴守箴，戚英杰.电气传的脉宽调制控制技术.北京：机械工业出版社.

[15] 贾玉瑛，王臣.基于单片机控制的PWM直流调速系统.包头钢铁学院学报，2005年.

[16] 康华光，邹寿彬.电子技术基础(数字部分第四版).北京：高等教育出版社，2004年.

[17] 三恒星科技.MCS 51单片机原理与应用实例.北京：电子工业出版社，2008年1月.

[18] 浦龙梅，李私.单片机控制的直流PWM调速装置的研究.变频器世界，2006年3月

[19] 张波.SG3525A控制的直流电机脉宽调速装置.电气制造，2006年第七期.

附录

附录1 C程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                                                 直流电机PWM调速

功能描述：使用单片机T0定时器产生(xxxHz)脉宽调制信号，输到L298N的EN\_A

                实现双直流电机的速度调控。

修订日期：2011-5-7

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<reg52.h>

#include<intrins.h>

#define uchar unsigned char

#define uint unsigned int

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                                                 L298n接口定义

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

sbit MOTOR\_A\_2=P3^6;

sbit MOTOR\_A\_1=P3^7;

sbit EN\_MOTOR\_A =P3^4;

sbit k1=P1^5;        //定义k1为p1.5口

sbit k2=P3^1;   //定义k2为p3.1口

sbit k3=P3^2;   //定义k3为p3.2口

sbit k4=P3^3;   //定义k4为p3.3口

uchar T=0;         //定时标记

uchar W=0;       //脉宽值   0~100

uchar A=0;        //方向标记 0，1

uchar k=0;         //按键标记

uchar i=0;            //计数变量

uchar code table1[]={

0x3f,0x06,0x5b,0x4f,

0x66,0x6d,0x7d,0x07,

0x7f,0x6f,0x77,0x7c,

0x39,0x5e,0x79,0x71};

uchar code table2[]={0xef,0xbf,0xdf,0x7f};

void delayms(uint t);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                                          数码管显示

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void disp(void)

{

       P0=table2[3];

       P2=table1[W%10];               //显示占空比个位

       delayms(1);                                //延时1ms

       P0=0xff;                                           //P0清1

       P0=table2[2];

       P2=table1[W/100];              //显示占空比百位

       delayms(1);                                //延时1ms

       P0=0xff;                                           //P0清1

       P0=table2[1];

       P2=table1[W/10%10];  //显示占空比十位

       delayms(1);                          //延时1ms

       P0=0xff;                                   //P0清1

       P0=table2[0];

       P2=table1[A];                    //显示方向

       delayms(1);                         //延时1ms

       P0=0xff;                                    //P0清1

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                                                 定时器变量定义

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void init(void)

{

       //启动中断

       TMOD=0x01;

       EA=1;

       ET0=1;

       TR0=1;

       //设置定时时间

       TH0=0xfd;

       TL0=0x17;

}

void timer0() interrupt 1

{

       //重置定时器时间

       TL0=0x17;

       TH0=0xff;

       T++;                   //定时标记加1

       disp();            //数码管显示

       if(T>W)

              EN\_MOTOR\_A =1;

       else

              EN\_MOTOR\_A =0;

       if(T==100)

              T=0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                                                 延时1ms

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void delayms(uint t)

{

       uchar j;

       while(t--)

       {

              for(j=0;j<250;j++)               //循环250次

              {

                      \_nop\_();             //系统延时

       \_nop\_();                                              //系统延时

       \_nop\_();                                              //系统延时

       \_nop\_();                                              //系统延时

              }

       }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                                                 独立按键检测

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void key(void)           //按键判断程序

{

       if(k1==0)                                                 //按键1按下

       {

              while(k1==0);                      //按键1抬起

              if(W==100)                                //如果脉宽为100

                     W=0;                                                //脉宽置0

              else

                     W+=1;                                              //否则加1

       }

       else if(k2==0)                      //按键2按下

       {

              while(k2==0);                      //按键2抬起

              if(W==0)                                          //如果脉宽为0

                     W=100;                                      //脉宽设置成100

              else

                     W-=1;                                               //否则减1

       }

       else if(k3==0)                      //按键3按下

       {

              while(k3==0);                      //按键3抬起

              A=!A;                                                      //方向标记取反

              k=0;                                                  //按键标记清0

       }

       else if(k4==0)                      //按键4按下

       {

              while(k4==0);                      //按键4抬起

              W=0;                                                       //脉宽清0

              k=0;                                                  //按键标记清0

       }

}

void main(void)

{

       init();                         /////////系统初始化

       while(1)

              key();                    ////////查询按键

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                                                   END

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

附录2：原理图

