

用单片机控制直流电机的设计

福星电子网

提供单片机学习板，开发板，最小系统板；超声波测距实验应用板，

各类器件仪表，详情请访问网站 <http://www.fxdzw.com>

用单片机控制直流电机的设计

摘要

本设计以 AT89C51 单片机为核心，以 4*4 矩阵键盘做为输入达到控制直流电机的启停、速度和方向，完成了基本要求和发挥部分的要求。在设计中，采用了 PWM 技术对电机进行控制，通过对占空比的计算达到精确调速的目的。

一、 设计方案比较与分析:

1、 电机调速控制模块:

方案一: 采用电阻网络或数字电位器调整电动机的分压, 从而达到调速的目的。但是电阻网络只能实现有级调速, 而数字电阻的元器件价格比较昂贵。更主要的问题在于一般电动机的电阻很小, 但电流很大; 分压不仅会降低效率, 而且实现很困难。

方案二: 采用继电器对电动机的开或关进行控制, 通过开关的切换对小车的速度进行调整。这个方案的优点是电路较为简单, 缺点是继电器的响应时间慢、机械结构易损坏、寿命较短、可靠性不高。

方案三: 采用由达林顿管组成的 H 型 PWM 电路。用单片机控制达林顿管使之工作在占空比可调的开关状态, 精确调整电动机转速。这种电路由于工作在管子的饱和截止模式下, 效率非常高; H 型电路保证了可以简单地实现转速和方向的控制; 电子开关的速度很快, 稳定性也极佳, 是一种广泛采用的 PWM 调速技术。

鉴于方案三调速特性优良、调整平滑、调速范围广、过载能力大, 因此本设计采用方案三。

2、 PWM调速工作方式:

方案一: 双极性工作制。双极性工作制是在一个脉冲周期内, 单片机两控制口各输出一个控制信号, 两信号高低电平相反, 两信号的高电平时差决定电动机的转向和转速。

方案二: 单极性工作制。单极性工作制是单片机控制口一端置低电平, 另一端输出 PWM 信号, 两口的输出切换和对 PWM 的占空比调节决定电动机的转向和转速。

由于单极性工作制电压波开中的交流成分比双极性工作制的小, 其电流的最大波动也比双极性工作制的小, 所以我们采用了单极性工作制。

3、 PWM调脉宽方式:

调脉宽的方式有三种: 定频调宽、定宽调频和调宽调频。我们采用了定频调宽方式, 因为采用这种方式, 电动机在运转时比较稳定; 并且在采用单片机产生 PWM 脉冲的软件实现上比较方便。

4、 PWM软件实现方式:

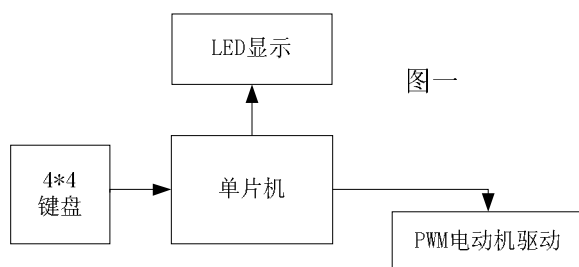
方案一: 采用定时器做为脉宽控制的定时方式, 这一方式产生的脉冲宽度极其精确, 误差只在几个 us。

方案二: 采用软件延时方式, 这一方式在精度上不及方案一, 特别是在引入中断后, 将有一定的误差。但是基于不占用定时器资源, 且对于直流电机, 采用软件延时所产生的定时误差在允许范围, 故采用方案二。

二、 系统分析与设计:

总体设计方案的硬件部分详细框图如图一所示。

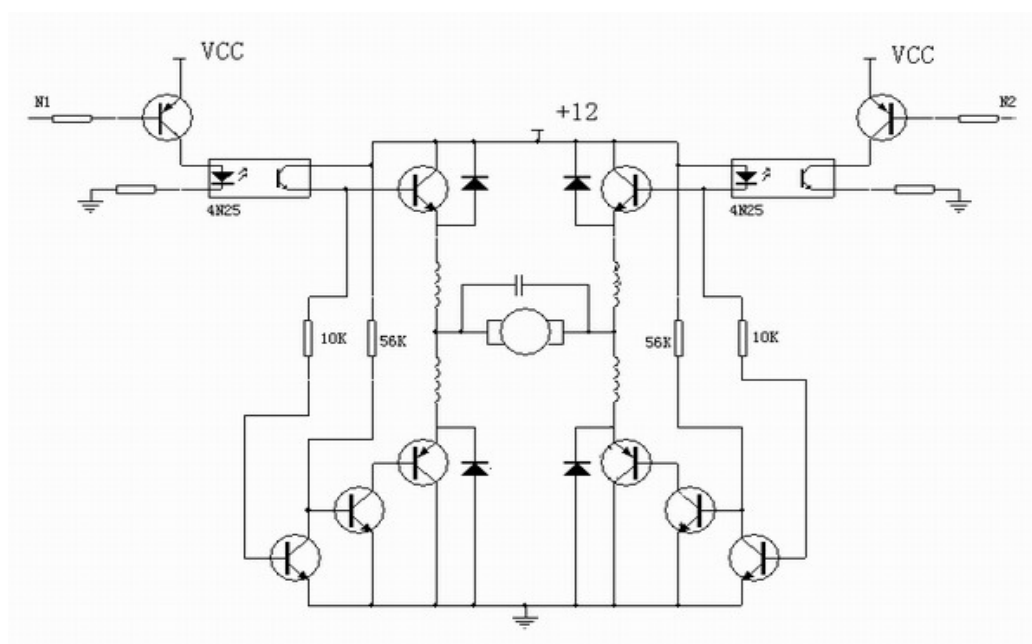
键盘向单片机输入相应控制指令，由单片机通过 P2.0 与 P2.1 其中一口输出与转速相应的 PWM 脉冲，另一口输出低电平，经过信号放大、光耦传递，驱动 H 型桥式电动机控制电路，实现电动机转向与转速的控制。电动机的运转状态通过 LED 显示出来。电动机所处速度级以速度档级数显示。正转时数字向右移动，反转时数字向左移动。移动速度分 7 档，快慢与电动机所处速度级快慢一一对应。每次电动机启动后开始计时，停止时 LED 显示出本次运转所用时间，时间精确到 0.1s。



图一

1、系统的硬件电路设计与分析

电动机 PWM 驱动模块的电路设计与实现具体电路见下图二。本电路采用的是基于 PWM 原理的 H 型桥式驱动电路。



图二

PWM 电路由四个大功率晶体管组成 H 型桥式电路构成，四部分晶体管以对角组合分为两组：根据两个输入端的高低电平决定晶体管的导通和截止。4 个二极管在电路中起防止晶体管产生反向电压的保护作用。4 个电感在电路中是起防止电动机两端的电流和晶体管上的电流过大的保护作用。

在实验中的控制系统电压统一为 5v 电源，因此若达林顿管基极由控制系统直接控制，则控制电压最高为 5V，再加上三极管本身压降，加到电动机

两端的电压就只有 4V 左右，严重减弱了电动机的驱动力。基于上述考虑，我们运用了 4N25 光耦集成块，将控制部分与电动机的驱动部分隔离开来。输入端各通过一个三极管增大光耦的驱动电流；电动机驱动部分通过外接 12V 电源驱动。这样不仅增加了各系统模块之间的隔离度，也使驱动电流得到了大大的增强。

在电动机驱动信号方面，我们采用了占空比可调的周期矩形信号控制。脉冲频率对电动机转速有影响，脉冲频率高连续性好，但带带负载能力差脉冲频率低则反之。经实验发现，脉冲频率在 40Hz 以上，电动机转动平稳，但加负载后，速度下降明显，低速时甚至会停转；脉冲频率在 10Hz 以下，电动机转动有明显跳动现象。实验证明，脉冲频率在 15Hz-30Hz 时效果最佳。而具体采用的频率可根据个别电动机性能在此范围内调节。通过 N1 输入信号，N2 输入低电平与 N1 输入低电平，N2 输入信号分别实现电动机的正转与反转功能。通过对信号占空比的调整来对车速进行调节。速度分 7 档控制，从高电平（第 6 档）到低电平（第 0 档）中间占空比以 20%逐极递减。速度微调方面，可以通过对占空比以 1%的跨度逐增或逐减分别实现对速度的逐加或逐减。

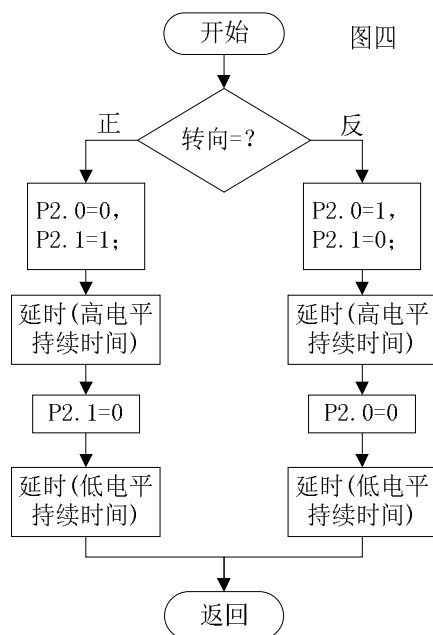
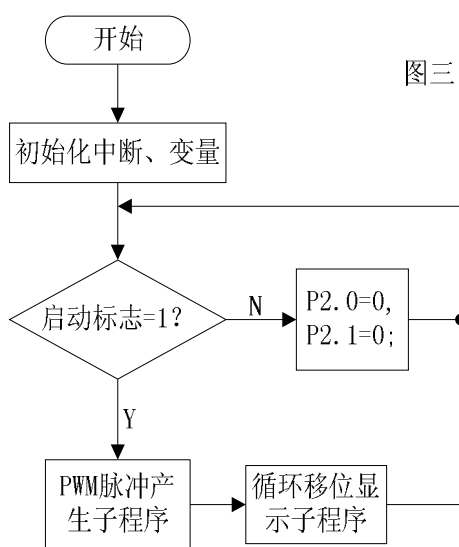
2、系统的软件设计

本系统编程部分工作采用 KELI-C51 语言完成，采用模块化的设计方法，与各子程序做为实现各部分功能和过程的入口，完成键盘输入、按键识别和功能、PWM脉宽控制和 LED 显示等部分的设计。

单片机资源分配如下表：

P0	显示模块接口	外部中断 0 (P3.2)	键盘中断
P1	键盘模块接口		
P2.0/P2.1	PWM电机驱动接口	内部定时器 0	系统时钟

系统主函数流程如图三：



①PWM脉宽控制：本设计中采用软件延时方式对脉冲宽度进行控制，延时程序函数如下：

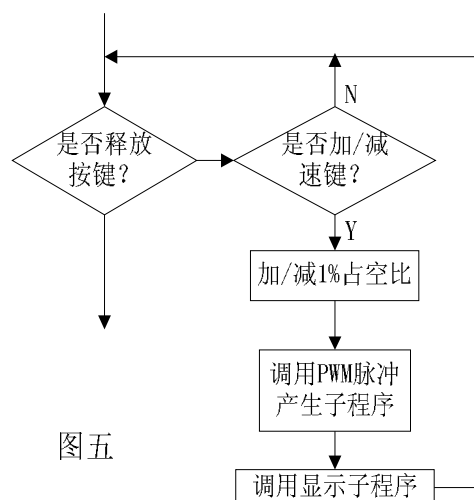
```
void delay(unsigned char dlylevel){
    int i=50*dlylevel;
    while(--i);}

```

此函数为带参数 **DLYLEVEL**，约产生 $DLYLEVEL*400\mu s$ 的延时，因此一个脉冲周期可以由高电平持续时间系数 **hlt** 和低电平持续时间系数 **llt** 组成，本设计中采用的脉冲频率为 **25Hz**，可得 $hlt+llt=100$ ，占空比为 $hlt/(hlt+llt)$ ，因此要实现定频调宽的调速方式，只需通过程序改变全局变量 **hlt**，**llt** 的值，该子程序流程图如图四。

②键盘中断处理子程序：采用中断方式，按下键，单片机 **P3.2** 脚产生一负跳沿，响应该中断处理程序，完成延时去抖动、键码识别、按键功能执行。

调速档、持续加/减速：调速档通过 (0-6) 共七档固定占空比，即相应档位相应改变 **hlt**，**llt** 的值，以实现调速档位的实现。而要实现按住加/减速键不放时恒加或恒减速直到放开停止，就需在判断是否松开该按键时，每进行一次增加/减少 1% 占空比（即 $hlt++/--$ ； $llt--/++$ ），其程序流程图如图五。



图五

③显示子程序：利用数组方式定义显示缓存区，缓存区有 8 位，分别存放各个 LED 管要显示的值。显示子程序为一带参子程序，参数为显示缓存的数组名，通过 `for(i=0; i<8; i++)` 方式对每位加上位选码，送到 **P0** 口并进行一两毫秒延时。

该显示子程序只对各个 LED 管分别点亮一次，因此在运行过程中，每秒执行的次数不应低于每秒 24 次。

④定时中断处理程序：采用定时方式 1，因为单片机使用 12M 晶振，可产生最高约为 65.5ms 的延时。对定时器置初值 3CB0H 可定时 50ms，即系统时钟精度可达 0.05s。当 50ms 定时时间到，定时器溢出则响应该定时中断处理程序，完成对定时器的再次赋值，并对全局变量 **time** 加 1，这样，通过变量 **time** 可计算出系统的运行时间。

对于一个数的显示，先应转成 BCD 码，即取出每一个位，分别送入显示缓存区，对于转 BCD 的算法，应对一个数循环除 10 取模，直至为 0，程序如下：

```
do{di spbuff[bcd_p]=bechange%10; //di spbuff 为显示缓冲区数组
    bcd_p++; }while(bechange/=10) //di sp_p 为数组指针

```

软件设计中的特点：

1、对于电机的启停，在 PWM 控制上使用渐变的脉宽调整，即开启后由停止匀

- 加速到默认速度,停止则由于当前速度逐渐降至零。这样有利于保护电机,如电机运用于小车上,在启动上采用此方式也可加大启动速度,防止打滑。
- 2、对于运行时间的计算、显示。配合传感器技术可用于计算距离,速度等重要的运行数据。
 - 3、键盘处理上采用中断方式,不必使程序对键盘反复扫描,提高了程序的效率。

三、 测试结果与分析:

结束语

本设计在硬件上采用了基于 **PWM** 技术的 **H** 型桥式驱动电路,解决了电机驱动的效率问题,在软件上也采用较为合理的系统结构及算法,提高了单片机的使用效率,且具有一定的防飞能力。但该设计也有不足之处,主要是在关于速度的反馈上,无法提供较为直观的速度表示方式,因此,有必要引入传感器技术对速度进行反馈,以 **rpm** 或 **rps** 表达当前的转速进行显示。