1 引言

机械手是近几十年发展起来的，由于它可以模仿人的手臂进行一些工作并且机械手作业的准确性和各种环境中完成工作的能力，在各个领域有广阔的发展前景。它可以代替人的繁重劳动以实现生产的机械化保护人身安全。机械手有四种驱动方式，其中电气驱动是最灵活的，而且可以实现机械手的闭环控制，本设计是关于电动式机械手的一个自由度的控制，结合DDC（直接数字控制系统），便可以实现机械手多个自由度的控制和协调。

2 机械手的驱动方式

2.1 液压驱动式

液压驱动式机械手通常由液动机（各种油缸、油马达）、伺服阀、油泵、油箱等组成驱动系统，由驱动机械手执行机构进行工作。通常它的具有很大的抓举能力（高达几百千克以上），其特点是结构紧凑、动作平稳、耐冲击、耐震动、防爆性好，但液压元件要求有较高的制造精度和密封性能，否则漏油将污染环境。

2.2 气压驱动式

其驱动系统通常由气缸、气阀、气罐和空压机组成，其特点是气源方便、动作迅速、结构简单、造价较低、维修方便。但难以进行速度控制，气压不可太高，故抓举能力较低。

2.3 电气驱动式

电力驱动是机械手使用得最多的一种驱动方式。其特点是电源方便，响应快，驱动力较大（关节型的持重已达400kg），信号检测、传动、处理方便，并可采用多种灵活的控制方案。

2.4 机械驱动式

机械驱动只用于动作固定的场合。一般用凸轮连杆机构来实现规定的动作。其特点是动作确实可靠，工作速度高，成本低，但不易于调整。其他还有采用混合驱动，即液-气或电-液混合驱动。

3 总体方案的设计

3.1 DDC 系统简介

DDC 系统是用一台工业计算机配以适当的输入输出设备，从生产过程中经输入通道获取信息，按照预先规定的控制算法计算控制量，并通过输出通道，直接作用在执行结构上，实现对整个生产，实验过程的闭环控制，可以有多个控制回路。

3.2 用 DDC 实现对机械手多个自由度的控制方法

利用单片机等元件构造一个闭环系统控制一个自由度，这样多个自由度就可以用多个这样的闭环系统进行控制 ， 然后利用单片机的通信功能，实现每个单片机通过串口与计算机进行通信，由上位机来发送命令，实现多个闭环系统的控制来达到控制机械手的多个自由度。在本次毕业设计中仅研究一个闭环控速系统。

3.3 实施方案设计

采用 STC89C52 单片机作为控制器，由于 STC89C52 单片机 I/O 口输出电流能力很小，所以用 L298N 芯片用以驱动直流电机，并且 L298N 内部的 H 桥可以实现单片机的直接正反转切换以及该芯片很容易配合单片机进行 PWM 调速。两通道高分辨率光学增量编码器模块并配有 888 线的光栅码盘实现对直流电机的精确测速，采用 12864 液晶来显示直流电机的速度和运行状态。

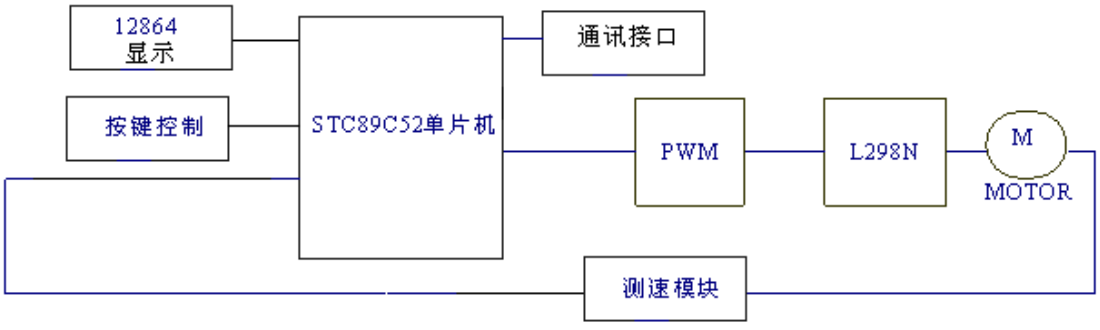


图 3-1

3.4 简单阐述控制机械手自由度和末端位置的原理

机械手的自由度由电机来驱动，那么控制电机的转速便可以控制这个自由度。当直流电机处于匀速运动的状态时，由速度和时间就可以算出它转了多少角度，然后根据机械装置的结构，便可知道末端运动了多少位移，就可以得到，机械手的所处的位置。

4 测速方案的比较和选定

4．1 方案一 测速发电机测速

测速发电机是输出电动势与转速成比例的微特电机。测速发电机的绕组和磁路经精确设计，其输出电动势 E 和转速 n 成线性关系，即 E=Kn ,K 是常数。改变旋转方向时输出电动势的极性即相应改变。在被测机构与测速发电机同轴联接时，只要检测出输出电动势，就能获得被测机构的转速，故又称速度传感器。

若采用测速发电机测速，此时需要 AD 芯片将模拟量转换为数字量给计算机进行处理。并且测速发电机的价格比较贵。

4．2 方案二 霍尔元件测速

只要在转轴的圆周上粘上一粒磁钢，让霍尔开关靠近磁钢，就有信号输出，转轴旋转时，就会不断地产生脉冲信号输出。如果在圆周上粘上多粒磁钢，可以实现旋转一周，获得多个脉冲输出。这样通过在一定的时间内统计脉冲的个数可以得到电机的转速。

该方案的缺点是：长期使用磁钢的磁性会下降。

4．3 方案三 光电编码器（该设计选用方案）

由于现在的码盘精度可以做的很高，测速非常的精准，所以光电编码器测速用的比较广泛。本设计选用的是两通道高分辨光学增量编码器模块，码盘的精度为 888 线。它的详细工作原理在第四章元件资料介绍中进行详细说明。测速的方法：把其中一个通道连接到 52 单片机的 T0 口。假设 1S 得到的脉冲个数为 N. 那么电机的转速为 N/888*60 (r/min)

5 元件资料介绍

5．1 电源模块

该设计采用了 6-12V 的直流稳压可调电源（主要用于给 L298N 的 4 号脚提供 9V 的电压，用来驱动电机） 和 5V 的 USB 供电。

5．2 直流电机

本次用的直流电机是日本万宝至制造的电机。6-12V 是可用的电压范围，本次设计用的带测速，减速用的直流电机模块的原设计是 HP 扫描仪上的。

5．3 STC89C52RC 简介

STC89C52RC 是一款超强抗干扰的单片机，无需看门狗电路，是 51 的增强型，多了一个定时器 T2，程序存储器是 8KB.

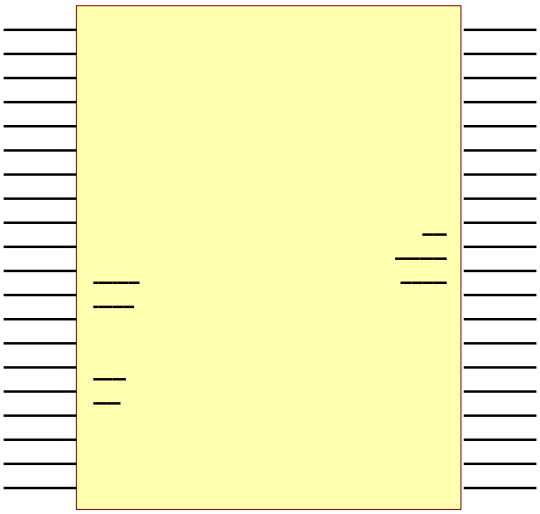


图 5-3

5.3.1 定时器/计数器 0 和 1

定时器/计数器 0 和 1 是 51 系列单片机的两个通用定时器/计数器，简称为 T0 和 T1 。T0 和 T1 都具有定时和计数的功能和四种工作模式，可以通过特殊功能寄存器来选择。

定时器/计数器的结构的和核心是一个 16 位的加 1 计数器，这个 16 位的计数器是由两个 8 位计数器组成的，定时器/计数器 T0 由 TH0 和 TL0 构成，T1 由 TH1 和 TL1 构成。TMOD 和 TCON 是定时器/计数器的控制寄存器。

1. 计数

计数就是对脉冲进行计数。其中，计数脉冲来自相应的外部输入引脚 P3.4（T0）或 P3.5（T1） 。 当该引脚的输入信号由高电平至低电平的负跳变时，计数器加 1.

2. 定时

定时是对时间进行统计。定时器/计数器的定时功能其实也是通过计数实现的，与计数功能不同的是，此时计数脉冲来自于单片机的内部时钟脉冲。

3．定时器 T0/T1 的控制寄存器 TMOD

不按位寻址，地址 89H

B7	B6	B5	B4	B3	B2	B1	B0
GATE	C/T	M1	M0	GATE	C/T	M1	M

表 5-3-1

GATE ：定时操作开关控制位，当 GATE=1 时，INT0 或 INT1 引脚为高电平，同时 TCON 中的 TR0 或 TR1 控制位为 1 时，计时/计数器 0 或 1 才开始工作。若 GATE=0，则只要将 TR0 或 TR1 控制位设为 1，计时/计数器 0 或 1 就开始工作。

C/T ：定时器或计数器功能的选择位。C/T=1 为计数器，通过外部引脚 T0 或 T1 输入计数脉冲。C/T=0 时为定时器，由内部系统时钟提供计时工作脉冲。

M1 ：模式选择位高位

M0 ：模式选择位低位

M1	M0	工作模式
0	0	13 位计数/计时器
0	1	16 位计数/计时器
1	0	8 位自动加载计数/计时器
1	1	定时器 1 停止工作,定时器 0 分为两个独立的 8 位定时器 TH0 及 TL0

表 5-3-2

4．定时器 T0/T1 的控制寄存器 TCON

按位寻址，地址位 88H B7	B6	B5	B4	B3	B2	B1	B0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

表 5-3-3-3

寄存器 TCON 的功能是在定时器/计数器溢出时设定标志位，并控制定时器的运行、停止和中断请求。其包含三个部分，TF1 和 TR1 位用与控制 T1,TF0 和 TR0 位用于控制好 T0,其他为中断控制。下面分别介绍各种控制位的含义。

溢出标志位 TF1/TF0

当定时器 T1/T0 溢出时，硬件自动将 TF1/TF0 置 1，并申请中断。当进入中断服务程序时，硬件又将自动清零 TF1/TF0 。

启/停控制位 TR1/TR0

该位由软件置位(1)和复位(0)。当 GATE 为 0 时，TR1/TR0 置位为 1 时，T1/T0 开始计数,TR1/TR0 复位为 0 时 T1/T0 停止计数;当 GATE 为 1 时,TR1/TR0 为 1 且 INT0/INT1 输入为高电平时，T0/T1 开始计数。

IE1：外部中断 1 请求标志。

IT1：外部中断 1 触发方式选择位。

IE0：外部中断 0 请求标志。

IT0：外部中断 0 触发方式选择位。

寄存器 TCON 的低 4 位与外部中断有关，由于该课题未使用外部中断功能，所以在这里不予详细说明。

5.3.2 定时器/计数器 2

定时器/计数器 2 是一个 16 位的具有自动重装载和捕获能力的定时器/计数器，在 52 系列单片机中提供。52 子系列单片机是对 51 子系列的扩展，在其特殊功能寄存器组中与 T2 有关的寄存器分别为控制寄存器 T2CON 和 T2MOD、捕获寄存器 RCAP2H 和 RCAP2L、定时器高低字节 TH2 和 TL2 。

该设计中采用了 T2 的重新再装入模式

DCEN=0；T2 为加 1 在装入方式。EXEN2=0，当 T2 计满回 0 溢出，将中断申请标志位 TF2 置位为 1，同时又将 RCAP2L、RCAP2H 中的初值分别重新再装入 TL2 和 TH2 中，继续下一轮计数。这种工作方式的功能与 T0/T1 的模式 2 相同，只是 T2 为 16 位，计数范围更大。注意：T2 申请中断当进入中断服务程序时，硬件不会自动将 TF2 清零，所以必须在中断服务程序中加入 TF2=0 。

5.3.3 定时器初值的计算方法

在这里以 T1 定时器的工作模式 1 为例进行说明。

T1 是 16 位的计数器，那么 2 的 16 次方是 65536， 那么初值的计算公式如下：

（65536-X）*机器周期=定时时间， X 为初值，机器周期=1/晶振的频率，不同的工作模式计数的总数是不同的，也就是公式中的 65536 这个值需要改变。

5.3.4 中断优先级

响应次序为：定时器 0－＞外中断 1－＞外中断 0－＞定时器 1－＞串行中断>定时器 2。

在微处理器的中断系统中，每个中断源赋予不同的优先级，根据优先级的不同来执行中断请求。解决的原则如下：

1. 在同一时刻，有两个中断同时提出请求时，中断系统按照中断源优先级的高低进行逐次响应。即优先级高的中断优先处理，处理完毕后，再处理优先级低的中断。这个过程称为中断优先级排队
2. 在一个中断得到请求并进行处理的进程中，如果另外一个中断发生，则判断其优先级的高低。如果新中断的优先级高于原中断，则在原中断服务程序中产生新断点，并转而执行新的中断服务子程序，然后逐级返回。这个过程称为中断嵌套。如果新的中断优先级低于原中断，而继续保持原中断的执行，直至完毕后返回再执行新中断请求。

如果想要改变中断的优先级，可以在控制寄存器 IP 中进行设置。

5.4 L298N 简介

L298N 是 ST 公司生产的一种高电压、大电流电机驱动芯片。该芯片采用 15 脚封装。主要特点是：工作电压高，最高工作电压可达 46V；输出电流大，瞬间峰值电流可达 3A，持续工作电流为 2A；额定功率 25W。内含两个 H 桥的高电压大电流全桥式驱动器，可以用

来驱动直流电动机和步进电动机、继电器线圈等感性负载；采用标准逻辑电平信号控制；具有两个使能控制端，在不受输入信号影响的情况下允许或禁止器件工作有一个逻辑电源输入端，使内部逻辑电路部分在低电压下工作；可以外接检测电阻，将变化量反馈给控制电路。使用 L298N 芯片驱动电机，该芯片可以驱动一台两相步进电机或四相步进电机，也可以驱动两台直流电机。 管脚如图：

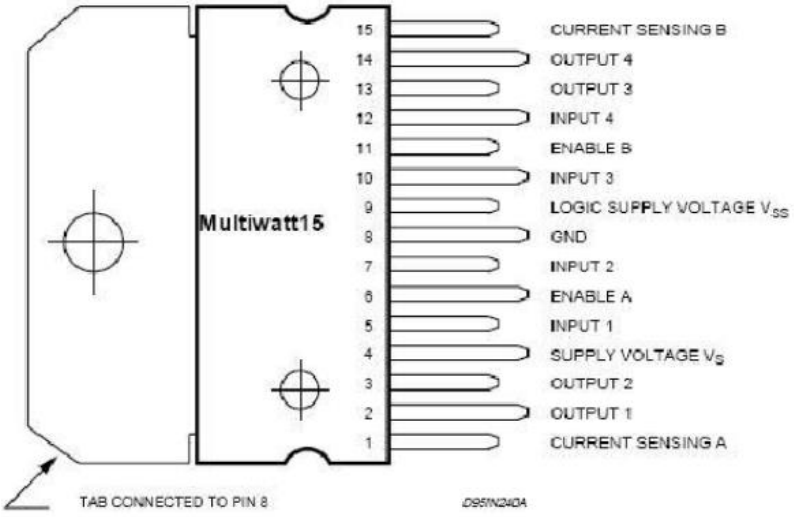


图 5-4

Enable A	Input 1	Input 2	电机运行状态
1	1	0	正转
	0	1	反转
	0	0	不转
	1	1	不转
0	X	X	停止

表 5-4-1

Enable B 对应 input 3 和 input 4，和上表是完全一样的。

5.4.1 H 桥驱动电机正反转的原理：

如图 5-4-1 所示，H 桥式电机驱动电路包括 4 个三极管和一个电机。要使电机运转，必须同时导通 Q1 和 Q4 或者 Q2 和 Q3 其中一对三极管。根据不同三极管对的导通情况，电流可能会从左至右或从右至左流过电机，从而控制电机的转向。

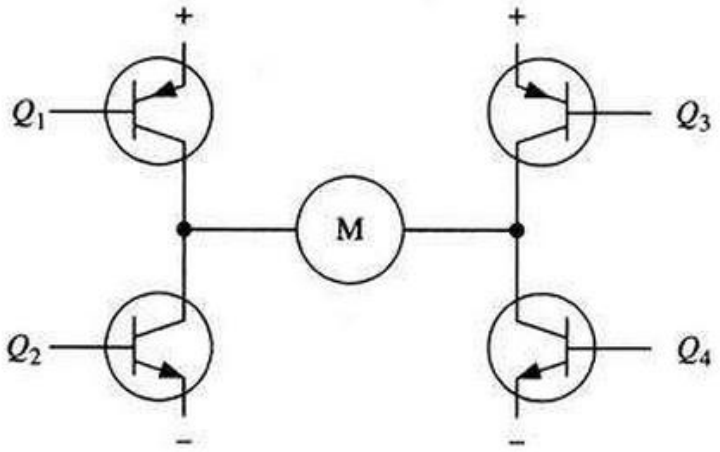


图 5-4-1

如图5-4-2所示，当 Q1管和 Q4管导通时，电流就从电源正极经 Q1从左至右穿过电机，然后再经 Q4回到电源负极。按图中电流箭头所示，该流向的电流将驱动电机顺时针转动。当三极管 Q1和 Q4导通时，电流将从左至右流过电机，从而驱动电机按特定的方向转动。

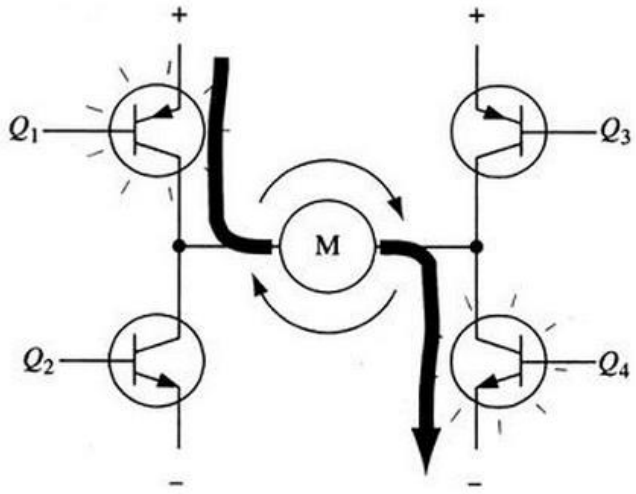


图5-4-2

图5-4-3 所示为另一对三极管 Q2和 Q3导通的情况，电流将从右至左流过电机。当三极管 Q2和 Q3导通时，电流将从右至左流过电机，从而驱动电机沿另一方向转动。

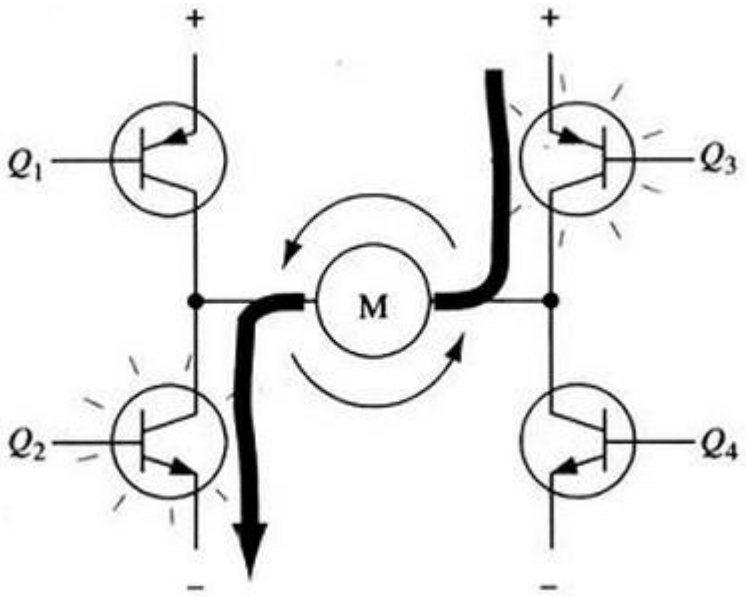


图 5-4-3

5.5 两通道高分辨率光学增量编码器模块 HEDS9000(惠普 美国计算机公司)

两通道高分辨率光学增量编码器模块 HEDS9000，配有 888 线的光栅码盘，即转一圈

输出 888 个脉冲，可以实现电机的精确测速。

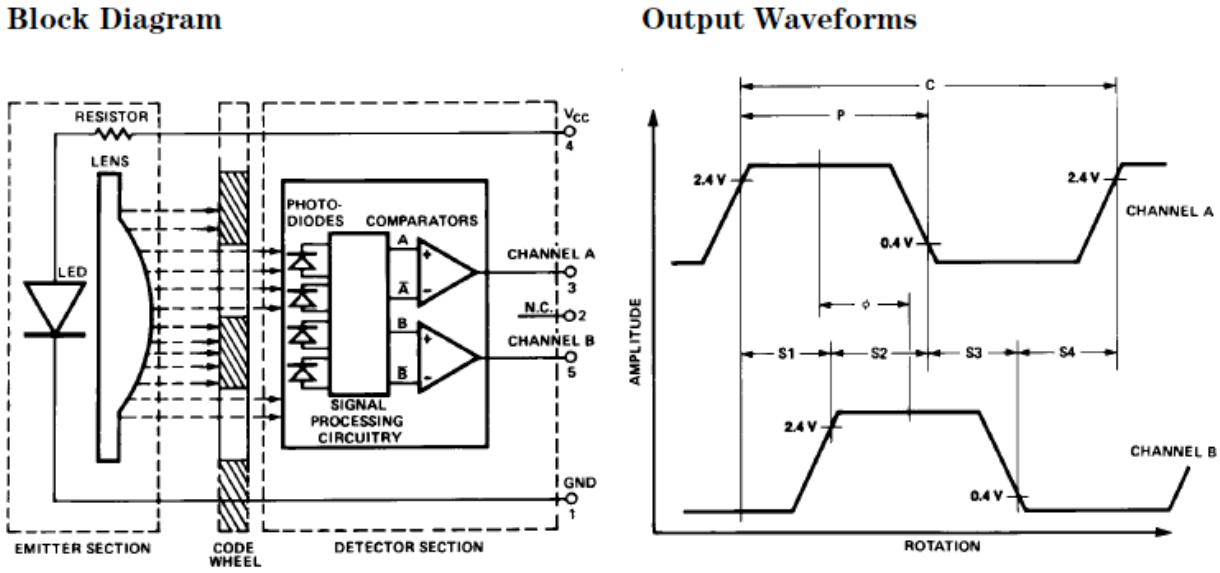


图 5-5

如图 5-5 所示，左边的虚线方框是发射器部分，这个部分由一个 LED 光源和透镜（透镜的材料是一种高分子聚碳酸酯 ，它的作用可以将光线变成平行光射出）； 中间是码盘，右边的虚线方框部分是集成的探测器电路，这个集成的探测器电路是由多组光电探测器和一个处理电路的组成，目的是为了产生数字波形。

工作过程：LED 通过透镜发出平行光，透过码盘的光束被一组光电二极管检测到（与此同时，另一组光电二极管接收不到光束），光电二极管的输出通过处理电路产生 A ,\A ,B ,\B,比较器接受这些信号并且产生了最终的输出 A 通道和 B 通道。由于集成的调相技术，使得 A、B 通道的数字输出正交 90 度。

5.6 12864 液晶

工作温度：-20C ---- +70C

人机界面采用了 12864 液晶，该液晶总共可显示 4 行，每行最多可以显示 8 个汉字或 16 个字符。

5.7 非编码按键（独立按键）

按键采用了非编码的独立按键，按键的抖动用软件延时方法。实现软件“消抖”（注：软件消抖并不是真正消除了按键在按下和松开时的抖动，而是在按键抖动的期间采用软件延时，来解决）

6. 硬件电路设计

6.1 总电路图

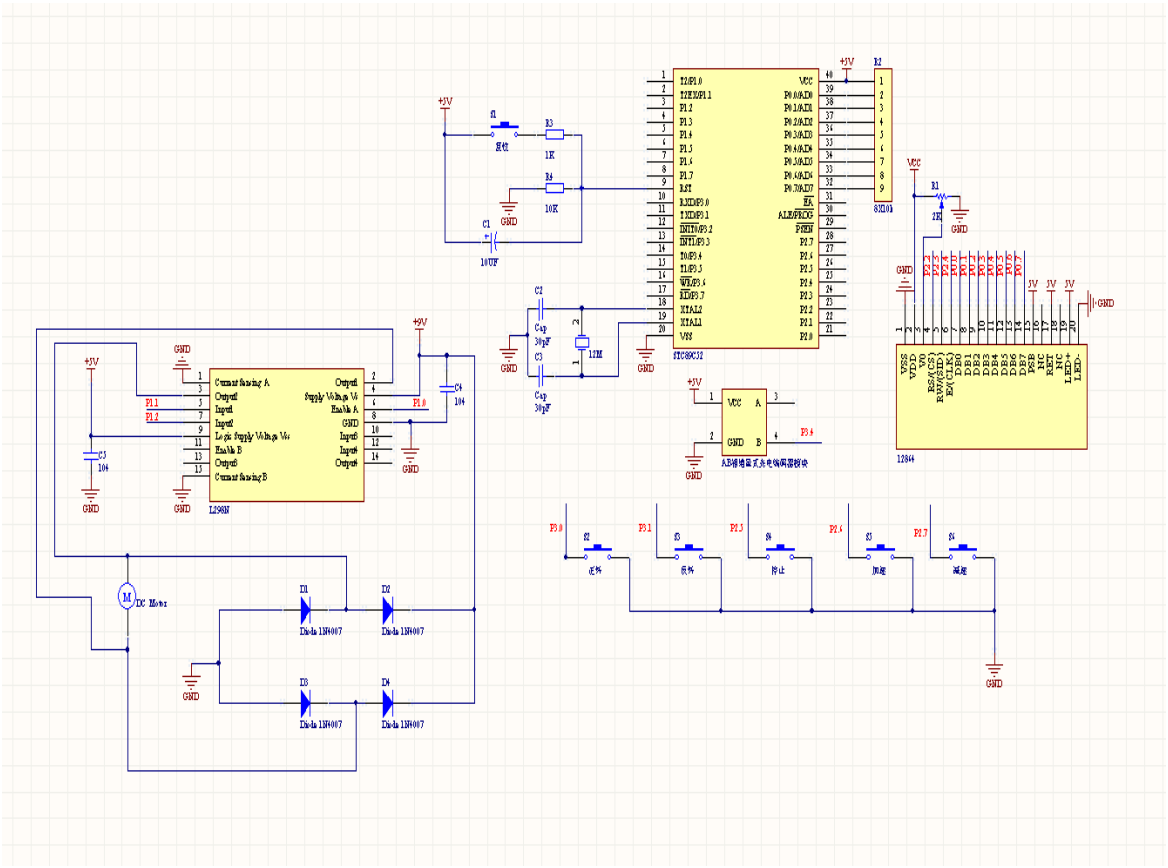


图 6-1

6.2 STC89C52 单片机最小系统

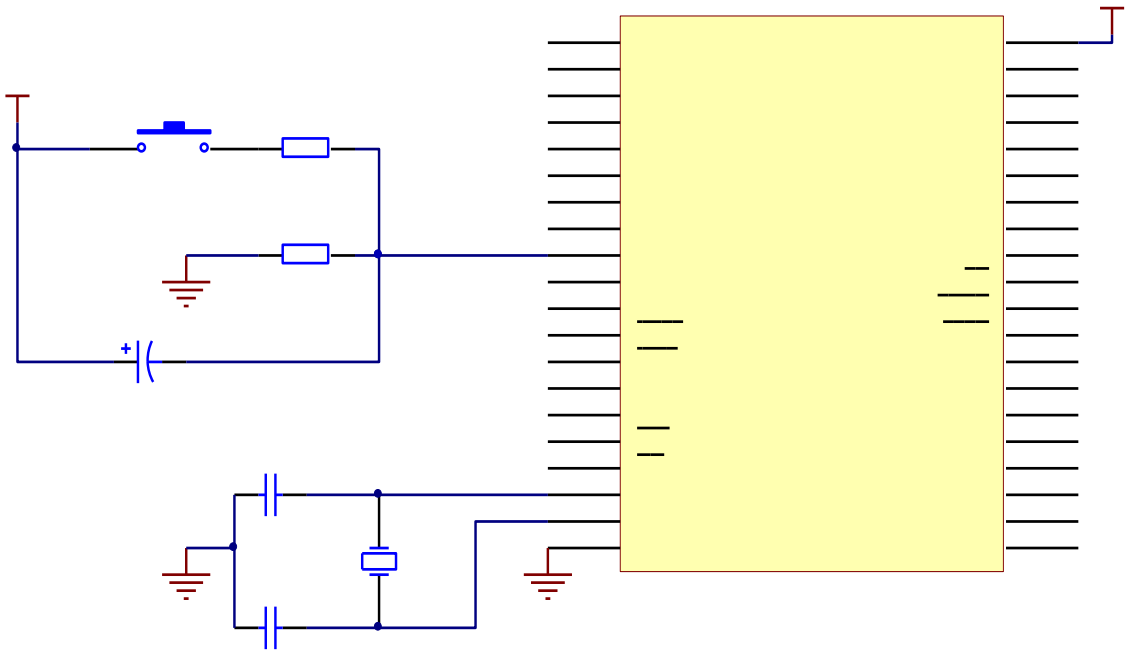


图 6-2

STC89C52 单片机加上复位电路，晶振电路和电源便可构成最小系统，使该单片机中的程序正常运行。复位电路是在程序跑飞时，通过复位按键可以给系统复位。晶振电路是用来给单片机提供时间的。

6.3 L298N 直流电机驱动电路

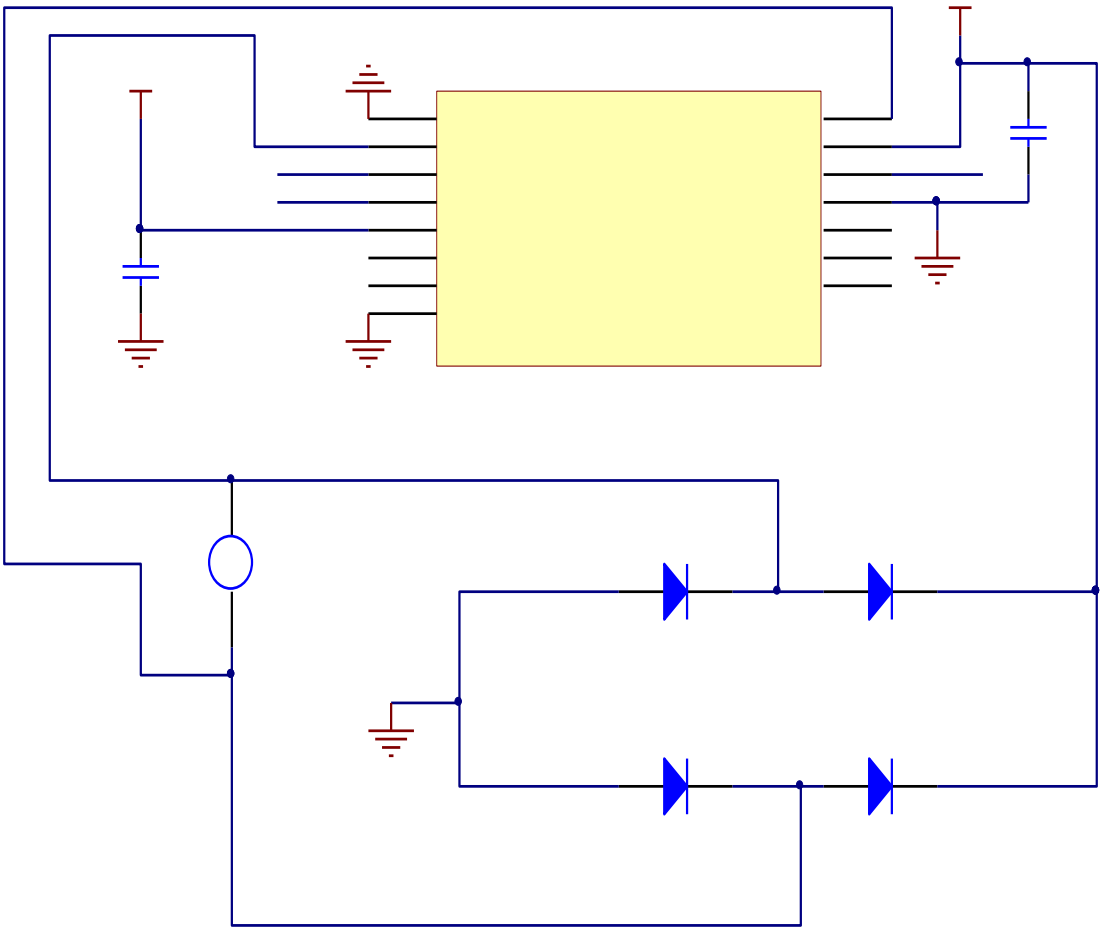


图 6-3

该图中的两个 104 是用来稳压的，实际证明效果不错。这个芯片的 4 号脚是接驱动电机的电压，9 号脚是该芯片的电压。

6.4 两通道高分辨率光学增量编码器

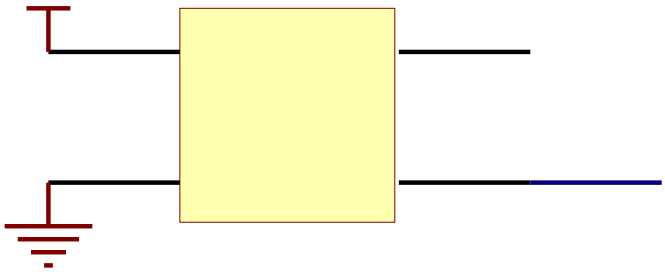


图 6-4

A 相和 B 相输出的脉冲是完全一样的，由于调相技术使得相位相差 90 度。

6.5 12864 液晶接口电路

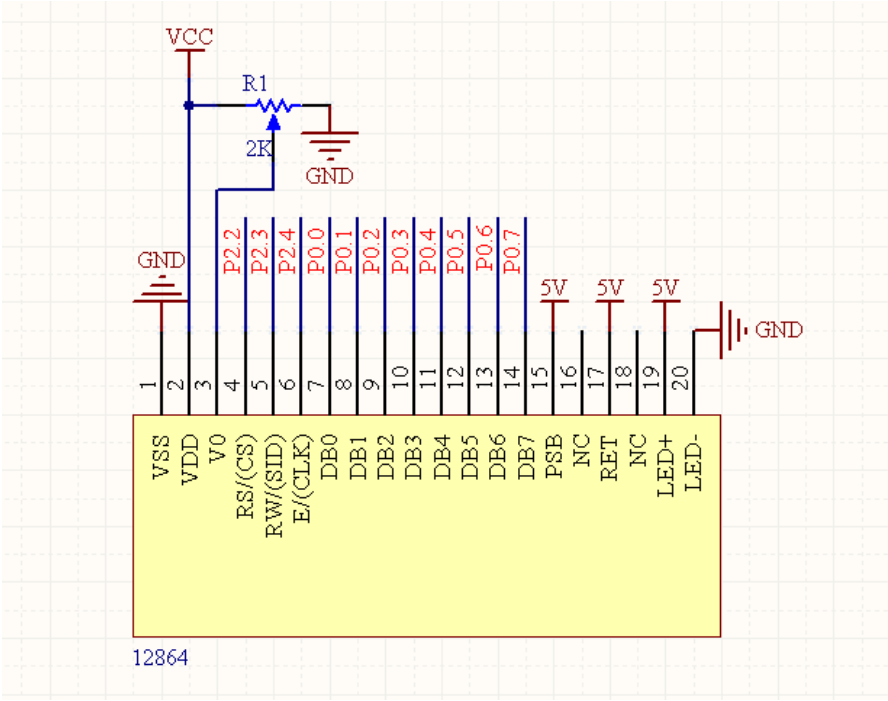


图 6-5

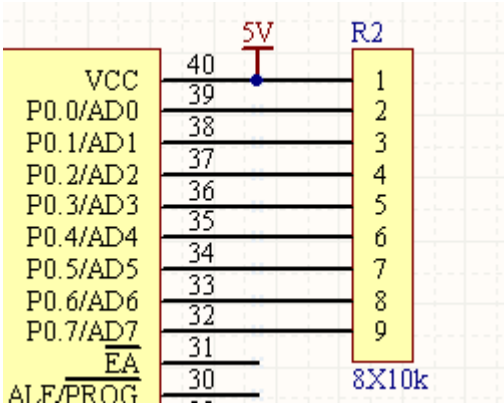


图 6-6

如图 6-4-2 所示，由于 STC89C52 单片机的 P0 口是开漏极，所以接 12864 液晶需要加上 8 个 10K 的上拉电阻。

6.6 独立键盘电路

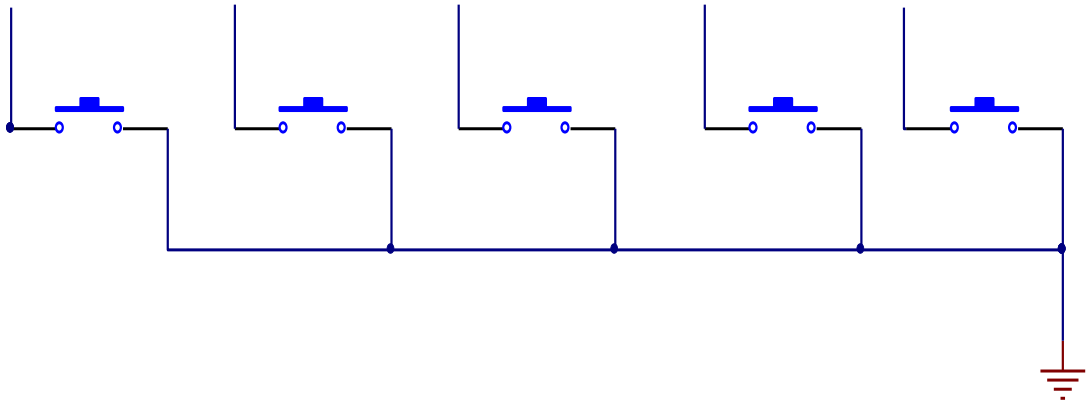


图 6-6

由于按键按下和松开是存在抖动的，本次设计是在软件中加入了延时去抖的方法。

7. PWM 调速原理

对于电机的转速调整，是采用脉宽调制（PWM）办法，控制电机的时候，电源并非连续地向电机供电，而是在一个特定的频率下以方波脉冲的形式提供电能。不同占空比的方波信号能对电机起到调速作用，这是因为电机实际上是一个大电感，它有阻碍输入电流和电压突变的能力，因此脉冲输入信号被平均分配到作用时间上，这样，改变在 L298N 始能端上输入方波的占空比就能改变加在电机两端的电压大小，从而改变了转速。

此电路中用 STC89C52 来实现脉宽调制。

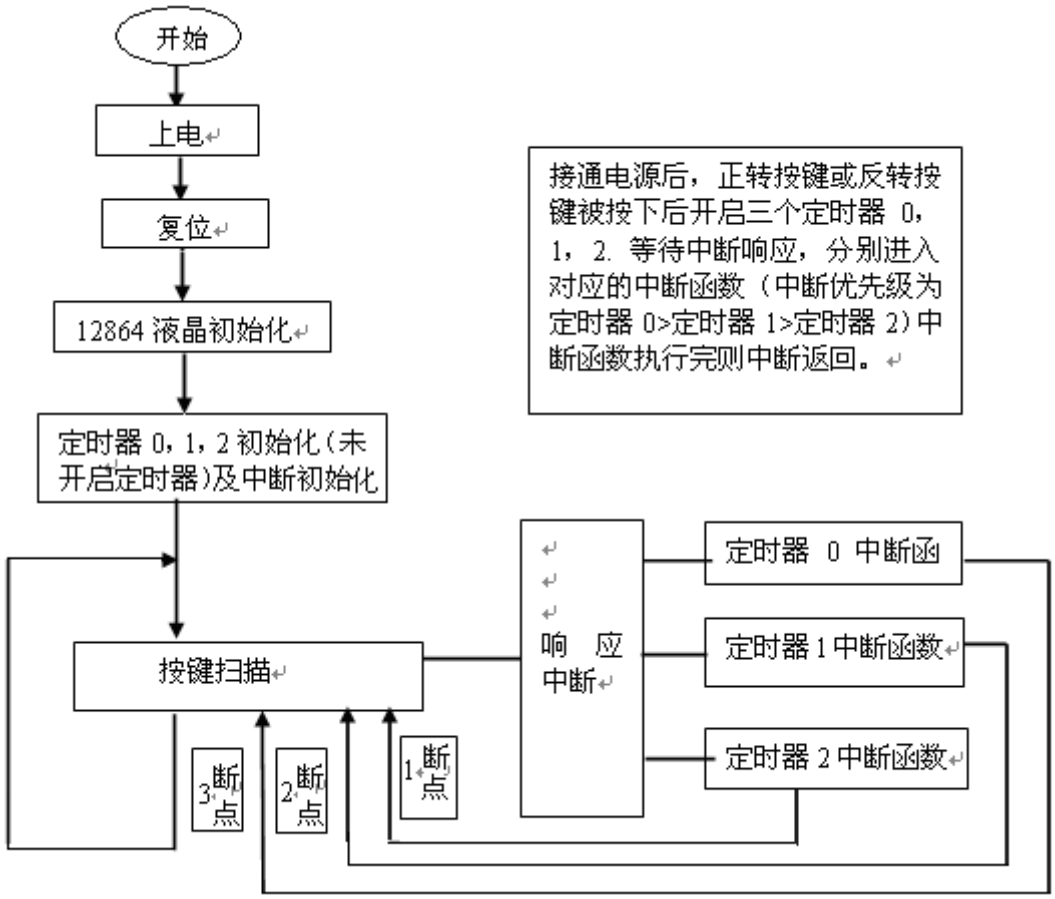
用软件方式来实现，即通过使用 52 单片机的定时器/计数器 2 进行定时，使得周期一定，改变 L298N 使能端高电平和低电平的时间，即改变占空比，来对直流电机进行调速。此方法叫定频调宽法。

8 软件设计

软件用的语言是 C51，单片机上电后，首先进入复位状态，然后程序开始执行，大致的思路是，先进行 12864 液晶初始化及显示一些数据，然后是三个定时器/计数器 T0，T1，T2 的初始化，然后不断扫描按键函数，当正转或反转按键按下时，才开启三个定时器/计数器，T0，T1，T2.等待中断发生执行相应函数，然后返回断点处继续执行。

通过测得的电机速度做数据分析和实际编写程序测试，计数器 T0 不会产生中断，只有 T1 和 T2 会产生中断，并且优先级 T1>T2(这样的优先级不会影响直流电机速度的准确性)；如果电机的速度足够大，使得 1S 内，T0 的脉冲计数值超过 65536。则 T0 会产生中断，这里三个中断的优先级是 T0>T1>T2,经过思考这样的中断优先级也不会影响直流电机速度计算的准确性，需要注意的是由于实现正反转之间的任意切换，所以当已经按下正转键后再按反转键或者反过来，则需要把 T0，T1，T2（T2 可以关或者不关，不影响）关闭，把一些和电机速度有关的数据清零，重新赋计数器 T0 和定时器 T1 的初值，然后重新打开三个定时器/计数器，因为此时的直流电机是先停止然后才向反方向旋转。

8.1 软件流程图（这里只给出大致思路，具体请看软件代码）



该软件设计，通过测得的电机速度做数据分析和实验测试，定时器 0 不会产生中断。在本次设计 888 线码盘，如果电机的速度足够大，使得 1S 内，T0 的脉冲计数值超过 65536，则会产生定时器 0 中断。

计数器 0 函数中有 n++,用于记录 T0 的溢出次数，用于转速计算

定时器 1 中断函数中有 num2++,当 num2=20,即 1S 时间到做速度计算。

定时器 2 中断函数中进行 PWM 占空比设置。

8.2 调试

软件主要有，12864 液晶初始化模块，定时器 0,1,2 初始化，按键模块，三个定时器中断服务函数。 程序是分模块调试出来的，调试顺序如下：

1. 先编写和调试了 12864 液晶的显示程序，使得 12864 液晶的显示没有问题。
2. 编写测速程序并通过 12864 进行显示。（这里开启了定时器/计数器 0 和 1， T0 外部计数，T1 定时，每 1 秒作一次速度计算）
3. 编写正转，反转和停止按键程序来控制电机的正转，反转，停止，设计正转按键或反转按键按下后，才开始测速。并且要实现正转，反转和停止按键之间的任意切换功能，这里需要做细节考虑。
4. 编写加速按键和减速按键程序，此时需要开启定时器 2 来实现，并且要考虑中断优先级，这涉及电机速度计算的准确性问题。
5. 实现所有功能后，程序进行先整体后局部的再次思考和检查，确保程序没有问题。
6. 代码的优化，要得到执行效率非常高的 C51 代码。

关于 DDC 控制，只要在此基础上加入单片机通讯，实现单片机与上位机之间的通信就可以完成多个自由度的协调控制。STC89C52 单片机的 P3.0（RXD）和 P3.1（ TXD）口，是用来通信的，通信方式为全双工，即可以同时接收和发送信号。这里由于时间紧迫所以未能加入此功能。

9 结　论

通过本次的毕业设计让我对 51 单片机的硬件和 C51 编程更进一步得到提高，以及这次对 51 的汇编，有更多的了解，使得我更进一步掌握单片机的工作原理。这次设计也锻炼我综合运用知识，解决工程问题的能力。同时也提高我查阅文献资料，使用 altium designer 软件制图能力。通过对整体的掌握，对细节的斟酌处理，都使我的能力得到锻炼，经验得到了丰富并且意志力，抗压能力及耐力也都得到了不同程度的提升。

本次设计中在硬件上需要注意的是码盘精度，软件上是按键在正反切换时，需要关闭所有定时器并且把和电机速度有关的几个参数清零，并且相应的给定时器 T1 从新赋初值，外部计数值 TH0，TL0 清零，再重新开启定时器，以及涉及中断优先级的问题。

最终理论和实践成功结合，做出了实物，并且达到了比较满意的结果。这样的一个闭环直流电机的控速系统是可以用来控制机械手的一个自由度，如果要实现多自由度的控制，则可以采用多个这样的系统，然后和一个上位机通信，实现 DDC 控制。

10 元件清单

元件	价格
STC89C52	5 元
直流电机及测速模块	18 元
L298N	8 元
12864 液晶	32 元
可调直流电源	12 元

共计 75 元

致　谢

时光飞逝，大学的学习生活已经结束，回首四年，取得了些许成绩，生活中有快乐也有艰辛。感谢老师四年来对我的教导，对我成长的关心和爱护。同时也感谢我的父母，他们辛勤的工作与付出，为我的学习提供条件，感谢他们这 20 多年的抚养与培育。

最后感谢邵老师，吴老师给予我的帮助，老师们有着渊博的知识，认真工作的作风，平易近人的态度，让我获益非浅。在此，向老师表示最衷心的感谢和最诚挚的敬意！

参 考 文 献

[1] 赵建领 崔昭霞. 精通 51 单片机开发技术与应用实例（升级版）. 电子工业出版社 2012.6

[2] 杨忠宝 董晓明 C 语言程序设计 北京大学出版社 2010.2

[3] 王义军 数字电子技术基础 中国电力出版社 2007.8

[4] 王书锋 谭建豪 主编 计算机控制技术. 华中科技大学出版社 2011.8

[5] 孙立志，PWM 与数字化电动机控制技术应用.中国电力出版社 2011.7.

[6] 兰吉昌，单片机 C51 完全学习手册.化学工业出版社 2009

[7] 王忠飞 MCS51 单片机原理及嵌入式系统应用. 西安电子科技大学出版社 2007.1

[8][美] Andrew Koening 著 高巍 译 王昕 审校 C 陷阱与缺陷 人民邮电出版社 2008.2

附录：软件代码

```
#include<reg52.h>
#include<stdio.h>
#define uchar unsigned char
#define uint unsigned int
sbit LCD_RS=P2^2;
sbit LCD_RW=P2^3;
sbit LCD_EN=P2^4;
sbit EN_MOTOR_A=P1^0;
sbit MOTOR_A_1=P1^1;
sbit MOTOR_A_2=P1^2;
sbit key1=P3^0;//正转
sbit key2=P3^1;//反转
sbit key3=P2^5;//停止
sbit key4=P2^6;//加速
sbit key5=P2^7;//减速
 uchar code temp[]="直流电机速度";//不变的数据放在片内的"code"存储区,节省 RAM.
 uchar code temp2[]="运行状态:";
 uchar code temp3[]="正转";
 uchar code temp4[]="反转";
 uchar code temp5[]="停止";
 uchar code temp6[]="  0 (r/min)";
static uchar temp7[16];  //定义字符显示缓冲数组
/*-----------全局变量-------------------*/
uchar num=0;//用于计算定时器 T1 是否中断达到 20 次，即 1S
uchar num2=0;//用于 T2 定时器的中断次数累计
uchar flag=0;//flag=1 说明反转键被按下
uchar flag2=0;//flag2=1 说明正转键被按下
uchar flag3=0;//判断 1 秒时间到的标志位
```

```
uchar t=1;//用于 PWM，调占空比，周期 T 一定
uint a=0;//用于转速计算
uint n=0;//记录 T0 的中断次数，用于转速的计算
/*-------------延时 1ms 函数------------*/
void delay_1ms(uint xms)
    { uint i,j;
       for(i=xms;i>0;i--)
         for(j=110;j>0;j--);
    }

/*--------------12864 写指令--------------------*/
void write_zl(uchar zl)
{
   LCD_RS = 0;
   LCD_RW = 0;
   LCD_EN = 0;
    P0=zl;
    delay_1ms(5);
    LCD_EN=1;
    delay_1ms(5);
     LCD_EN = 0;
}

/*-----------------12864 写数据-------------------*/
void write_data(uchar dat)
{

   LCD_RS = 1;
   LCD_RW = 0;
   LCD_EN = 0;
```

```
    P0=dat;
    delay_1ms(5);
    LCD_EN=1;
    delay_1ms(5);
     LCD_EN = 0;
}

/*-----------------------------------------------
                写入字符串函数
-----------------------------------------------*/
 void LCD_write_string(unsigned char *s)
 {

    while (*s) // "0/"的ASII码值是0.所以当指针变量s指向temp数组的最后一次字符0/ 时 会跳出这个while循环
    {
       write_data(*s);
        s++;
    }

 }


/*-------------12864初始化函数---------------------*/
void init_12864()
{
   delay_1ms(100);
  write_zl(0x30);//功能设定，RE=0基本指令动作，G=0绘图显示关闭
  delay_1ms(5);
   write_zl(0x30);
```

```
   delay_1ms(5);
    write_zl(0x0c);//显示开，关光标
    delay_1ms(5);
   write_zl(0x01);//清除显示屏幕，把 DDRAM 位址计数器调整为 00H.
    delay_1ms(5);


}
/*------------------定时器 0,1,2 初始化----------------------*/
void init_timers()
{
 TMOD =0x15;//（0001 0101）定时器 T1 的定时，选择方式 1；定时器 T0 计数，选择方
式 1
 TH0=0x00;       //给定初值
 TL0=0x00;
 TH1=(65536-50000)/256;
 TL1=(65536-50000)%256;
 TH2=(65536-10000)/256;//定时 10ms
 TL2=(65536-10000)%256;
 EA=1;             //总中断打开
 ET0=1;    //定时器 0 中断打开
 ET1=1;    // 定时器 1 中断打开
 ET2=1;
RCAP2H=(65536-10000)/256;//重装初值
RCAP2L=(65536-10000)%256;
}
```

```
/*------------------按键检测----------------------*/
void key()
{
  if(key1==0)//正转
  {
    delay_1ms(5);
    if(key1==0)
    {
          while(!key1);
          delay_1ms(5);
          flag2=1;//正转标志位，为 1 说明按下了正转按键

        if(flag==0)
        {
          TR0=1;//启动定时器 0
          TR1=1;//启动定时器 1
           TR2=1;//启动定时器 2

        }
        else
        {
          TR0=0;//关闭定时器 0
          TR1=0;//关闭定时器 1
           TR2=0;
          n=0;
          num=0;
           num2=0;
           TH0=0x00;
          TL0=0x00;
```

```
            TH1=(65536-50000)/256;          //重新赋值 50ms
             TL1=(65536-50000)%256;
            TH2=(65536-10000)/256;//重新赋值 10ms
             TL2=(65536-10000)%256;
             TR0=1;//启动定时器 0
             TR1=1;//启动定时器 1
             TR2=1;//启动定时器 2
          }


              EN_MOTOR_A=1;
             MOTOR_A_1=1;
             MOTOR_A_2=0;
            write_zl(0x9d);
            LCD_write_string(temp3);




    }


 }


if(key2==0)//反转
 {
   delay_1ms(5);
   if(key2==0)
   { while(!key2);
      delay_1ms(5);
     flag=1;//反转标志位，flag=1 说明按下反转按键
```

```
 if(flag2==0)
 {
    TR0=1;//启动定时器 0
    TR1=1;//启动定时器 1
    TR2=1;//启动定时器 2
 }

 else
 {
  TR0=0;//关闭定时器 0
  TR1=0;//关闭定时器 1
  TR2=0;
   n=0;
   num2=0;
    num=0;
     TH0=0x00;
    TL0=0x00;
    TH1=(65536-50000)/256;          //重新赋值 50ms
      TL1=(65536-50000)%256;
       TH2=(65536-10000)/256;//重新赋值 10ms
     TL2=(65536-10000)%256;
      TR0=1;//启动定时器 0
      TR1=1;//启动定时器 1
      TR2=1;//启动定时器 2

 }

EN_MOTOR_A=1;
 MOTOR_A_1=0;
```

```
        MOTOR_A_2=1;

     write_zl(0x9d);//显示
    LCD_write_string(temp4);


   }


}


if(key3==0)//停止
{
  delay_1ms(5);
  if(key3==0)
  { while(!key3);
     delay_1ms(5);
    EN_MOTOR_A=0;
   TR0=0;//关闭定时器 0
    TR1=0;//关闭定时器 1
    TR2=0;//关闭定时器 2
    flag=0;//清反转标志位
    flag2=0;//清正转标志位
     num=0;
       n=0;
      TH0=0x00;
      TL0=0x00;
     TH1=(65536-50000)/256;          //重新赋值 50ms
      TL1=(65536-50000)%256;
```

```
        TH2=(65536-10000)/256;//重新赋值 10ms
        TL2=(65536-10000)%256;

    write_zl(0x90);//液晶第二行 行首
    LCD_write_string(temp6);

     write_zl(0x9d);
    LCD_write_string(temp5);
  }


}


if(key4==0)//加速
{
  delay_1ms(5);
  if(key4==0)
  {
    while(!key4);
    delay_1ms(5);


     if(t==0)
      t=0;
     else
     t--;
```

```
    }

  }


  if(key5==0)//减速
  {
    delay_1ms(5);
    if(key5==0)
    {
      while(!key5);
      delay_1ms(5);


       if(t==10)
        t=10;
       else
      t++;



    }

  }



}
 /*-------------速度函数--------------------*/
```

```
  void speed()
  {


      a=TH0*256+TL0;//50ms 时的脉冲个数
      a=65536*n+a/888*60;//电机的速度 r/min
      sprintf(temp7,"%4d(r/min)",a);
      write_zl(0x90);//液晶第二行的第一个位置
     LCD_write_string(temp7);//显示到液晶第二行


  }




void main()
{


     init_12864();//液晶初始化
      write_zl(0x80);//液晶第一行第一个位置
      LCD_write_string(temp);


      write_zl(0x90);//第二行第一个位置
        LCD_write_string(temp6);
```

```
    write_zl(0x98);//液晶第四行第一个位置
       LCD_write_string(temp2);

    write_zl(0x9d);
     LCD_write_string(temp5);



 init_timers();//定时器 0,1,2 初始化

while(1)
{

   key();//按键扫描
   if(flag3==1)
   {
    speed();
     flag3=0;

   }

}

}
```

```
/*-------------------------------------------------
                定时器 0 中断子程序(T0 用于外部计数）
-------------------------------------------------*/
void Timer0_isr() interrupt 1
{
 n++;

}

/*-------------------------------------------------
                定时器 1 中断子程序（T1 用于定时）
-------------------------------------------------*/
void Timer1_isr() interrupt 3
{
   num++;
    TH1=(65536-50000)/256;        //重新赋值 50ms
     TL1=(65536-50000)%256;
   if(num==20)//1s 时间到作速度计算
   {
     num=0;
     TR0=0;//停止 T0 外部脉冲计数
     TR1=0; //停止定时

      flag3=1;

    TH0=0x00;
     TL0=0x00;
    TR0=1;
```

```
    TR1=1;


   }




}

/*--------------定时器 2 中断---------------------定时用于 PWM 调速*/
void timer2() interrupt 5
{

  TF2=0;
  num2++;
      if(num2<=t)
      {
       if(EN_MOTOR_A==1)
       EN_MOTOR_A=0;
      }

      else
      {

       if(EN_MOTOR_A==0)
       EN_MOTOR_A=1;

      }
```

```
    if(num2==10)
     num2=0;


}
```