

用单片机控制步进电机

步进电机是机电控制中一种常用的执行机构，它的用途是将电脉冲转化为角位移，通俗地说：当步进驱动器接收到一个脉冲信号，它就驱动步进电机按设定的方向转动一个固定的角度（及步进角）。通过控制脉冲个数即可以控制角位移量，从而达到准确定位的目的；同时通过控制脉冲频率来控制电机转动的速度和加速度，从而达到调速的目的。

一、步进电机常识

常见的步进电机分三种：永磁式（PM），反应式（VR）和混合式（HB），永磁式步进一般为两相，转矩和体积较小，步进角一般为 7.5 度 或 15 度；反应式步进一般为三相，可实现大转矩输出，步进角一般为 1.5 度，但噪声和振动都很大。在欧美等发达国家 80 年代已被淘汰；混合式步进是指混合了永磁式和反应式的优点。它又分为两相和五相：两相步进角一般为 1.8 度而五相步进角一般为 0.72 度。这种步进电机的应用最为广泛。

二、永磁式步进电机的控制

下面以电子爱好者业余制作中常用的永磁式步进电机为例，来介绍如何用单片机控制步进电机。

图 1 是 35BY 型永磁步进电机的外形图，图 2 是该电机的接线图，从图中可以看出，电机共有四组线圈，四组线圈的一个端点连在一起引出，这样一共有 5 根引出线。要使用步进电机转动，只要轮流给各引出端通电即可。将 COM 端标识为 C，只要 AC、 $\overline{A}C$ 、BC、 $\overline{B}C$ ，轮流加电就能驱动步进电机运转，加电的方式可以有多种，如果将 COM 端接正电源，那么只要用开关元件（如三极管），将 A、 \overline{A} 、B、 \overline{B} 轮流接地。

下表列出了该电机的一些典型参数：

表 1 35BY48S03 型步机电机参数								
型号	步距角	相数	电压	电流	电阻	最大静转距	定位转距	转动惯量
35BY48S03	7.5	4	12	0.26	47	180	65	2.5

有了这些参数，不难设计出控制电路，因其工作电压为 12V，最大电流为 0.26A，因此用一块开路输出达林顿驱动器（ULN2003）来作为驱动，通过 P1.4~P1.7 来控制各线圈的接通与切断，电路如图 3 所示。开机时，P1.4~P1.7 均为高电平，依次将 P1.4~P1.7 切换为低电平即可驱动步进电机运行，注意在切换之前将前一个输出引脚变为高电平。如果要改变电机的转动速度只要改变两次接通之间的时间，而要改变电机的转动方向，只要改变各线圈接通的顺序。



图 1 35BY48S03 型步进电机外形图

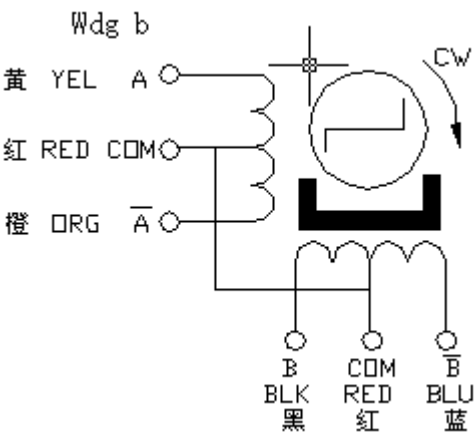


图 2 35BY48S03 型步进电机的接线图

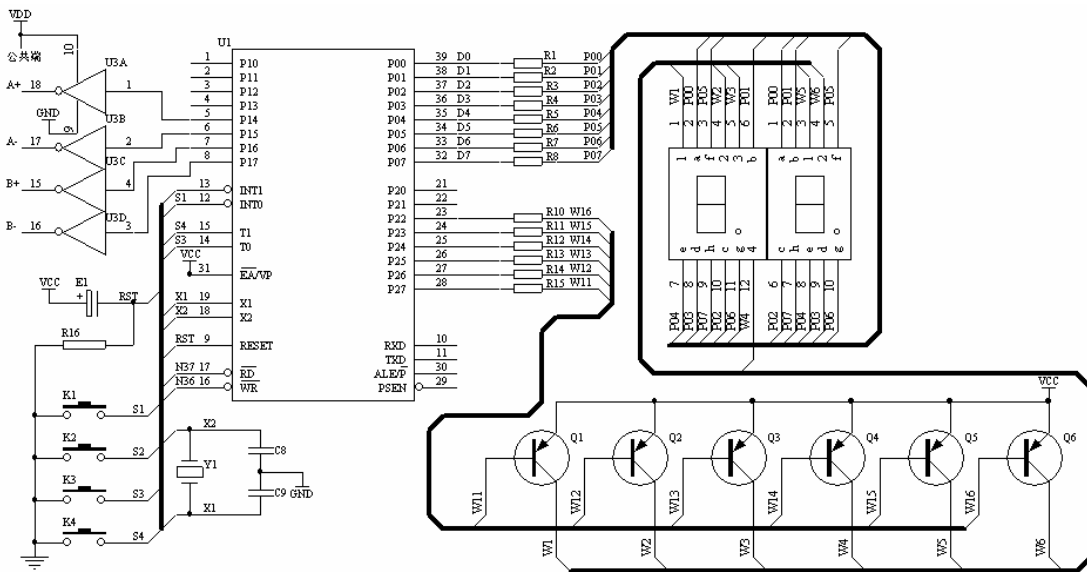


图 3 单片机控制 35BY48S03 型步进电机的电路原理图

三、步进电机的驱动实例

要求：控制电路如图 3 所示，开机后，电机不转，按下启动键，电机旋转，速度为 25 转/分，按下加 1 键，速度增加，按下减 1 键，速度降低，最高速度为 100 转/分，最低转带为 25 转/分，按下停止键，电机停转。速度值要求在数码管上显示出来。

1. 要求分析

按上面的分析，改变转速，只要改变 P1.0~P1.3 轮流变低电平的时间即可达到要求，这个时间不应采用延时来实现，因为会影响到其他功能的实现。这里以定时的方式来实现。下面首先计算一下定时时间。

按要求，最低转速为 25 转/分，而上述步进电机的步距角为 7.5，即每 48 个脉冲为 1 周，即在最低转速时，要求为 1200 脉冲/分，相当于 50ms/脉冲。而在最高转速时，要求为 100 转/分，即 48000 脉冲/分，相当于 12.5ms/脉冲。可以列出下表

表 1 步进电机转速与定时器定时常数关系

速度	单步时间(us)	TH1	TL1	实际定时(us)
25	50000	76	0	49996.8
26	48077	82	236	48074.18
27	46296	89	86	46292.61
28	44643	95	73	44640.155
...
100	12500	211	0	12499.2

表中不仅计算出了 TH1 和 TL1，而且还计算出了在这个定时常数下，真实的定时时间，可以根据这个计算值来估算真实速度与理论速度的误差值。

表中 TH1 和 TL1 是根据定时时间算出来的定时初值，这里用到的晶振是 11.0592M。有了上述表格，程序就不难实现了，使用定时/计数器 T1 为定时器，定时时间到后切换输出脚即可。

2. 程序实现

定义 DSB-1A 实验板的 S1 为启动键，S2 为停止键，S3 为加 1 键，S4 为减 1 键，程序如下：

StartEnd bit 01H ;起动及停止标志	MOV DispBuf+1,A
MinSpd EQU 25 ;起始转动速度	MOV DispBuf+2,A
MaxSpd EQU 100 ;最高转动速度	
Speed DATA 23H ;流动速度计数	MOV DjCount,#11110111B
DjCount DATA 24H ;控制电机输出的一个值，初始为 11110 111	MOV SPEED,#MinSpd ;起始转动速度送入计数器
Hidden EQU 10H ;消隐码	CLR StartEnd ;停转状态
Counter DATA 57H ;显示计数器	MOV TMOD,#00010001B ;
DISPBUF DATA 58H ;显示缓冲区	MOV TH0,#HIGH(65536-3000)
	MOV TL0,#LOW(65536-3000)
	MOV TH1,#0FFH;
	MOV TL1,#0FFH
	SETB TR0
	SETB EA
	SETB ET0
	SETB ET1
ORG 0000H	LOOP: ACALL KEY ;键盘程序
AJMP MAIN	JNB F0,m_NEXT1 ;无键继续
ORG 000BH	ACALL KEYPROC ;否则调用键盘
JMP DISP	处理程序
ORG 001BH	m_NEXT1:
JMP DJZD	
ORG 30H	
MAIN:	
MOV SP,#5FH	
MOV P1,#0FFH	
MOV A,#Hidden	
MOV DispBuf,A	

```

MOV    A,Speed
MOV    B,#10
DIV    AB
MOV    DispBuf+5,B ;最低位
MOV    B,#10
DIV    AB
MOV    DispBuf+4,B
MOV    DispBuf+3,A
JB     StartEnd,m_Next2
CLR    TR1      ;关闭电机
JMP    LOOP
ORL    P1,#11110000B
m_Next2:
    SETB    TR1      ;启动电机
    AJMP    LOOP      ;主程序结束
;-----
D10ms:
.....
;-----延时程序,键盘处理中调用
KEYPROC:
    MOV     A,B      ;获取键值
    JB      ACC.2,StartStop ;分析键的代码,某位被按下,则该位为 1
    JB      ACC.3,KeySty
    JB      ACC.4,UpSpd
    JB      ACC.5,DowSpd
    AJMP    KEY_RET
StartStop:
    SETB    StartEnd ;启动
    AJMP    KEY_RET
KeySty:
    CLR     StartEnd; ;停止
    AJMP    KEY_RET
UpSpd:
    INC     SPEED;
    MOV     A,SPEED
    CJNE    A,#MaxSpd,K1 ;到了最多的次数?
    DEC     SPEED ;是则减去 1, 保证下次仍为该值
K1:
    AJMP    KEY_RET
DowSpd:
    DEC     SPEED

```

```

MOV     A,SPEED
CJNE    A,#MAXSPD,KEY_RET ;不
等（未到最大值），返回
    MOV     SPEED,#MinSpd;
KEY_RET:
    RET

KEY:
.....获取键值的程序
    RET

DjZd:    ;定时器 T1 用于电机转速控制
    PUSH    ACC
    PUSH    PSW
    MOV     A,Speed
    SUBB    A,#MinSpd ;减基准数
    MOV     DPTR,#DjH
    MOVC    A,@A+DPTR
    MOV     TH1,A
    MOV     A,Speed
    SUBB    A,#MinSpd
    MOV     DPTR,#DjL
    MOVC    A,@A+DPTR
    MOV     TL1,A
    MOV     A,DjCount
    CPL     A
    ORL     P1,A
    MOV     A,DjCount
    JNB     ACC.7,d_Next1
    JMP     d_Next2
d_Next1:
    MOV     DjCount,#11110111B
d_Next2:
    MOV     A,DjCount
    RL      A
    MOV     DjCount,A ;回存
    ANL     P1,A
    POP     PSW
    POP     ACC
    RETI

DjH:     DB
        76,82,89,95,100,106,110,115,119,123,12
.....

```

DjL:DB 0,236,86,73,212,0,214,96,163,165	7Fh,0BFH,0DFH,0EFH,0F7H,0FBH
.....	DISPTAB:DB
DISP: ;显示程序	0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H
POP PSW	,80H,90H,88H,83H,0C6H,0A1H,86H,8EH,0F
POP ACC	FH
.....	END
RETI	
BitTab: DB	

3. 程序分析

本程序主要由键盘程序、显示器程序、步进电机驱动程序三部份组成，主程序首先初始化各变量，将显示器的高3位消隐，步进电机驱动的各引脚均输出高电平，然后调用键盘程序，并作判断，如果有键按下，则调用键盘处理程序，否则直接转下一步。下一步是将当前的转速值转换为BCD码，送入显示缓冲区；接着判断StartEnd这个位变量，是“1”还是“0”，如果是“1”，则开启定时器T1，否则关闭定时器T1，为防止关闭时某一相线圈长期通电，因此，在关闭定时器T1时，将P1.0~P1.3均置高。至此，主程序的工作即结束。这里为简便起见，这里没有做高位“0”消隐的工作，即如果速度为10转/分，则显示值“010”，读者可以自行加入相关的代码来处理这一工作。

步进电机的驱动工作是在定时器T1的中断服务程序中实现的，由前述分析，每次的定时时间到达以后，需要将P1.0~P1.3依次接通，程序中用了一个变量DjCntr来实现这一功能，在主程序初始化时，该变量被赋予初值11110111B，进入到定时中断以后，将该变量取出送ACC累加器，并在累加器中进行左移，这样，该数值就变为1110 1111，然后将该数与P1相“与”，此时，P1.4即输出低电平，第二次进入中断时，先将该数取反，成为0001 0000，然后将该数与P1相“或”，这样，P1.4即输出高电平，关断了相应的线圈，然后将该数重新取出，并作左移，即1110, 1111右移成为1101 1111，将该数与P1相“与”，这样P1.5即输出低电平，依次类推，P1.7~P1.4即循环输出低电平。当这一数据变为0111 1111后，需要作适当的改动，将数据重新变回1111 0111，进行第二次循环，相关代码，请读者自行分析。

定时时间又是如何确定的呢？这里用的是查表的方法，首先用Excel计算得出在每一种转速下的TH值和TL值，然后，分别放入DjH和DjL表中，在进入T1中断程序之后，将速度值变量Speed送入累加器ACC，然后减去基数25，使其基数从0开始计数，然后分别查表，送入TH1和TL1，实现重置定时初值的目的。

看完这一部份内容以后，请读者自行完成以下工作：

1. 更改程序，将S1定义为“启动/停止”，而S2定义为“方向”，按下S2，切换电机旋转方向。
2. 更改程序，要求转速从1到100。
3. 更改程序，实现首位无效零消隐。