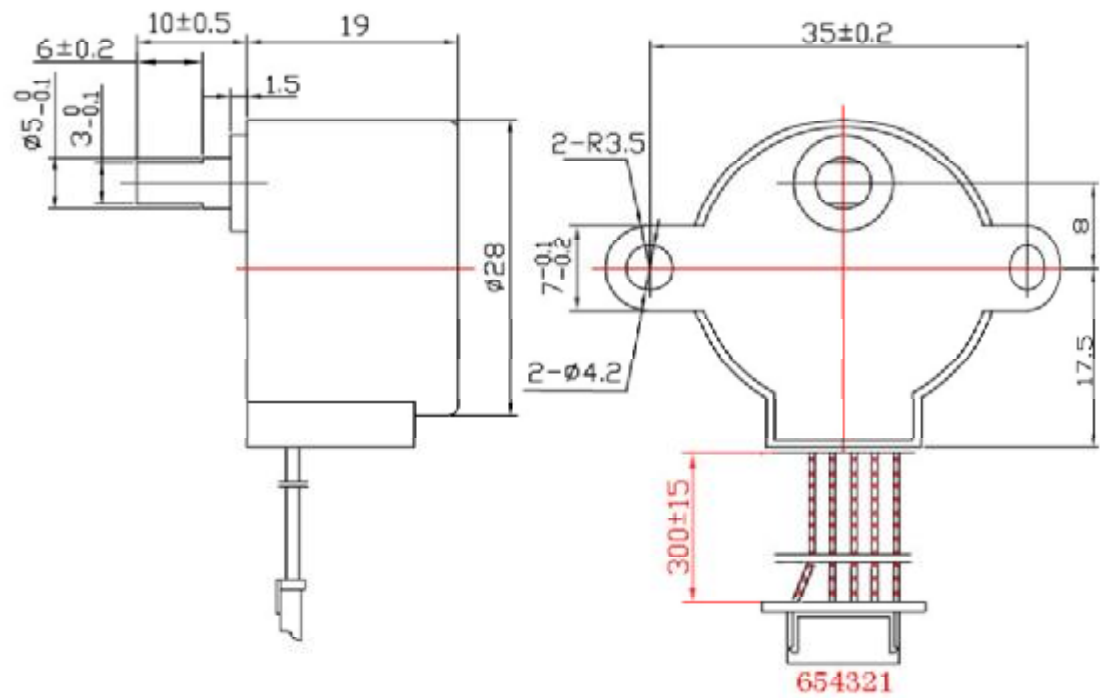


24BYJ48 步进电机使用手册

外型尺寸

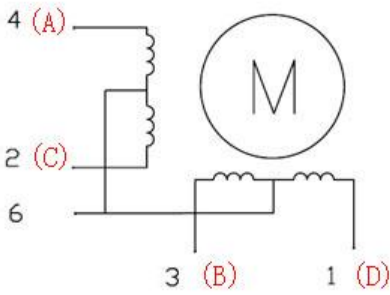


驱动方法及参数

驱动方式: (4-1-2相驱动)

导线颜色	1	2	3	4	5	6	7	8
6 红	+	+	+	+	+	+	+	+
4 橙	-	-	-	-	-	-	-	-
3 黄	-	-	-	-	-	-	-	-
2 粉	-	-	-	-	-	-	-	-
1 蓝	-	-	-	-	-	-	-	-

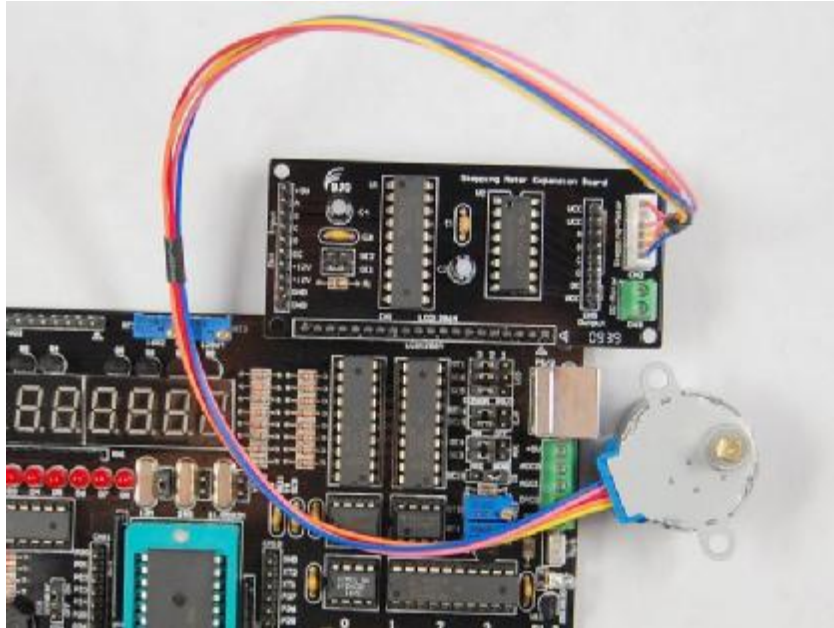
→ CCW 方向旋转 (轴伸端视)



主要技术参数

电机型号	电压 V	相数	相电阻 Ω $\pm 10\%$	步距角度	减速比	起动转矩 100P. P. S g. cm	起动频率 P. P. S	定位转矩 g. cm	摩擦转矩 g. cm	嘈声 dB	绝缘介 电强度
28BYJ- 48	5	4	300	5.625/64	1:64	≥ 300	≥ 550	≥ 300	—	≤ 35	600VAC 1S

插入到开发板的方法：



直接插入到 LCD12864 位置

一、前言：

步进电机是将电脉冲信号转变为角位移或线位移的开环控制元件。在非超载的情况下，电机的转速、停止的位置只取决于脉冲信号的频率和脉冲数，而不受负载变化的影响，即给电机加一个脉冲信号，电机则转过一个步距角。这一线性关系的存在，加上步进电机只有周期性的误差而无累积误差等特点。使得在速度、位置等控制领域用步进电机来控制变的非常的简单。为此，黑金刚开发板套件中首次引入了步进电机技术，采用扩展的方式，方便用户应用掌握。

虽然步进电机已被广泛地应用，但步进电机并不能象普通的直流电机，交流电机在常规下使用。它必须由双环形脉冲信号、功率驱动电路等组成控制系统方可使用。因此用好步进电机却非易事，它涉及到机械、电机、电子及计算机等许多专业知识。

步进电机的主要特性：

1、 步进电机必须加驱动才可以运转，驱动信号必须为脉冲信号，没有脉冲的时候，步进电机静止，如果加入适当的脉冲信号，就会以一定的角度（称为步角）转动。转动的速度和脉冲的频率成正比。

2、 黑金刚配套的是 28BYJ48 5V 驱动的 4 相 5 线的步进电机，而且是减速步进电机，减速比为 1：64，步进角为 $5.625/64$ 度。如果需要转动 1 圈，那么需要 $(360/5.625)*64=4096$ 个脉冲信号。

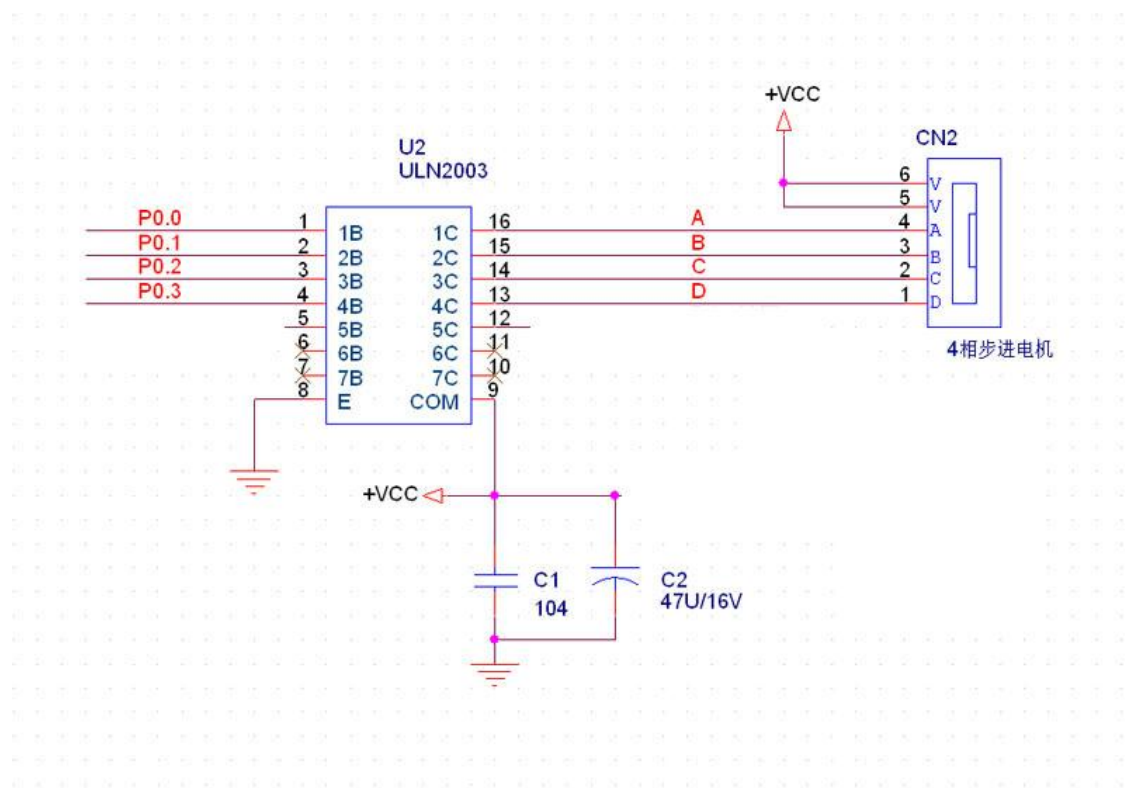
3、 步进电机具有瞬间启动和急速停止的优越特性。

4、 改变脉冲的顺序， 可以方便的改变转动的方向。

因此，目前打印机，绘图仪，机器人，等等设备都以步进电机为动力核心。

二、配套的原理图及程序：

1、 如果不考虑数据锁存功能，步进电机的扩展板电路可以简化为下图：



2、黑金刚套件采用的是 5V 步进电机，该步进电机的耗电流为 200ma 左右， 采用 uln2003 驱动，驱动端口为 P0.0 (A) , P0.1(B) , P0.2(C) , P0.3(C)。正转次序：AB 组—BC 组—CD 组—DA 组（即一个脉冲, 正转 5.625 度）；反转次序：AB 组—AD 组—CD 组—CB 组（即一个脉冲, 正转 5.625 度）, 如下表：

表 1：正转表

端口值	步数	P0.0(A)	P0.1(B)	P0.2(C)	P0.3(D)
0x03	1	1	1	0	0
0x06	2	0	1	1	0
0x0C	3	0	0	1	1
0x09	4	1	0	0	1

表 2：反转表

端口值	步数	P0.0(A)	P0.1(B)	P0.2(C)	P0.3(D)
0x03	1	1	1	0	0
0x09	2	1	0	0	1
0x0C	3	0	0	1	1
0x06	4	0	1	1	0

3 ASM 驱动程序：

```
*** ****步进电机的驱动*****
```

```
;FOSC = 12MHz
```

```
;
```

```
; 步进电机的驱动信号必须为脉冲信号!!! 转动的速度和脉冲的频率成正比!!!
```

```
; 本步进电机步进角为 5.625 度， 一圈 360 度， 需要 64 个脉冲完成!!!
```

```
;
```

```
; A 组线圈对应 P0.0
```

```
; B 组线圈对应 P0.1
```

```
; C 组线圈对应 P0.2
```

```
; D 组线圈对应 P0.3
```

```
; 正转次序：AB 组—BC 组—CD 组—DA 组（即一个脉冲, 正转 5.625 度）
```

```
;
```

```

ORG 0000H
    LJMP    MAIN
ORG 0100H

MAIN:
;-----正转-----
    MOV R3, #192      ;正转 3 圈共 192 个脉冲
START:
    MOV R0, #00H
START1:
    MOV P0, #00H
    MOV A, R0
    MOV DPTR, #TABLE
    MOVC    A, @A+DPTR
    JZ  START      ;对 A 的判断, 当 A=0 时则转到 START
    MOV P0, A
    LCALL  DELAY
    INC R0
    DJNZ   R3, START1
    MOV P0, #00H
    LCALL  DELAY1
;-----反转-----
    MOV R3, #128      ;反转 2 圈共 128 个脉冲
START2:
    MOV P0, #00H
    MOV R0, #05
START3:
    MOV A, R0
    MOV DPTR, #TABLE
    MOVC    A, @A+DPTR
    JZ  START2
    MOV P0, A
    CALL   DELAY
    INC R0
    DJNZ   R3, START3
    MOV P0, #00H
    LCALL  DELAY1
    LJMP  MAIN
;-----转速控制-----
DELAY: MOV R7, #10      ;步进电机的转速
M3: MOV R6, #248
    DJNZ   R6, $
    DJNZ   R7, M3
    RET
;-----延时控制-----
DELAY1:  MOV R4, #5      ;2S 延时子程序
DEL2:   MOV R3, #200
DEL3:   MOV R2, #250
    DJNZ   R2, $
    DJNZ   R3, DEL3

```

```

    DJNZ    R4, DEL2
    RET
;-----正反转表-----
TABLE:
    DB  03H, 06H, 0CH, 09H;   正转表
    DB  00;   正转结束
    DB  03H, 09H, 0CH, 06H;   反转表
    DB  00;   反转结束
END

```

4、C 语言驱动程序：

```

/*****步进电机的驱动*****/
;FOSC = 12MHz
;-----
; 步进电机的驱动信号必须为脉冲信号!!! 转动的速度和脉冲的频率成正比!!!
; 本步进电机步进角为 5.625 度 . 一圈 360 度 , 需要 64 个脉冲完成!!!
;-----
; A 组线圈对应 P0.0
; B 组线圈对应 P0.1
; C 组线圈对应 P0.2
; D 组线圈对应 P0.3
; 正转次序: AB 组--BC 组--CD 组--DA 组 (即一个脉冲, 正转 5.625 度)
;-----
*****/

/*头文件*/
#include <reg52.h>
#include <intrins.h>
#define uint unsigned int
#define uchar unsigned char
#define nop() _nop_()

uchar code table1[]={0x03, 0x06, 0x0c, 0x09}; /*正转表*/
uchar code table2[]={0x03, 0x09, 0x0c, 0x06}; /*反转表*/
#define motor P0

void delay(uchar ms) /*延时*/
{
    uchar j;
    while(ms--){
        for(j=0;j<250;j++)
            {;}
    }
}

```

```

void main()
{
    uchar i, j;
    while(1) {
        //正转 3 圈共 192 个脉冲
        j=0;
        for(i=0; i<192; i++) {
            motor = 0x00;
            motor = table1[j];
            j++;
            if(j>=4) j=0;
            delay(2);
        }
        delay(200);
        delay(200);
        delay(200);
        delay(200);
        delay(200);
        //正转 2 圈共 128 个脉冲
        j=0;
        for(i=0; i<128; i++) {
            motor = 0x00;
            motor = table2[j];
            j++;
            if(j>=4) j=0;
            delay(2);
        }
        delay(200);
        delay(200);
        delay(200);
        delay(200);
        delay(200);
    }
}

```