



单片机应用技术（C51）

主讲：陈丽娟

步进电机的控制

- 一、步进电机的结构、驱动方式
- 二、应用8x51控制步进电机
- 三、步进电机的1相、2相驱动应用

一、步进电机的结构

步进电机（stepping motor）：

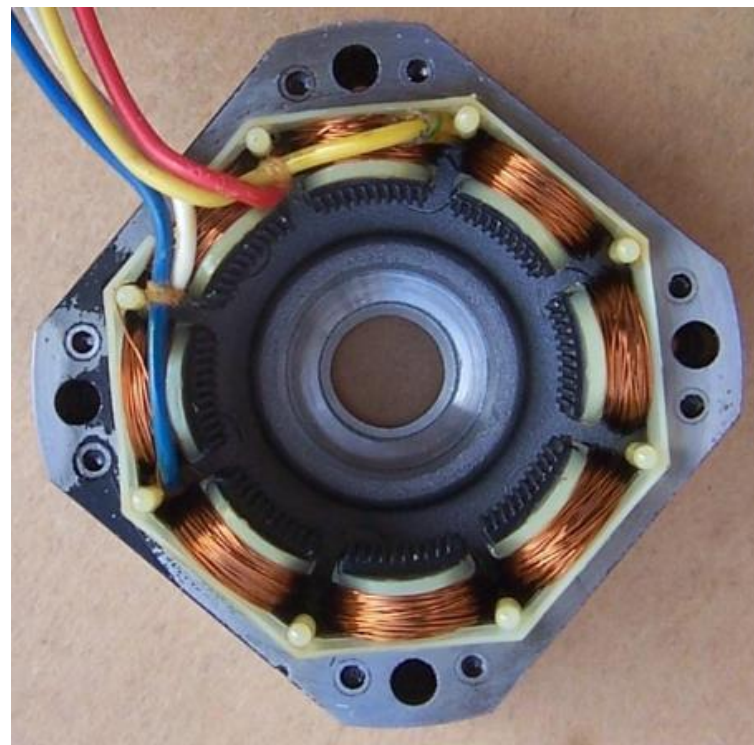
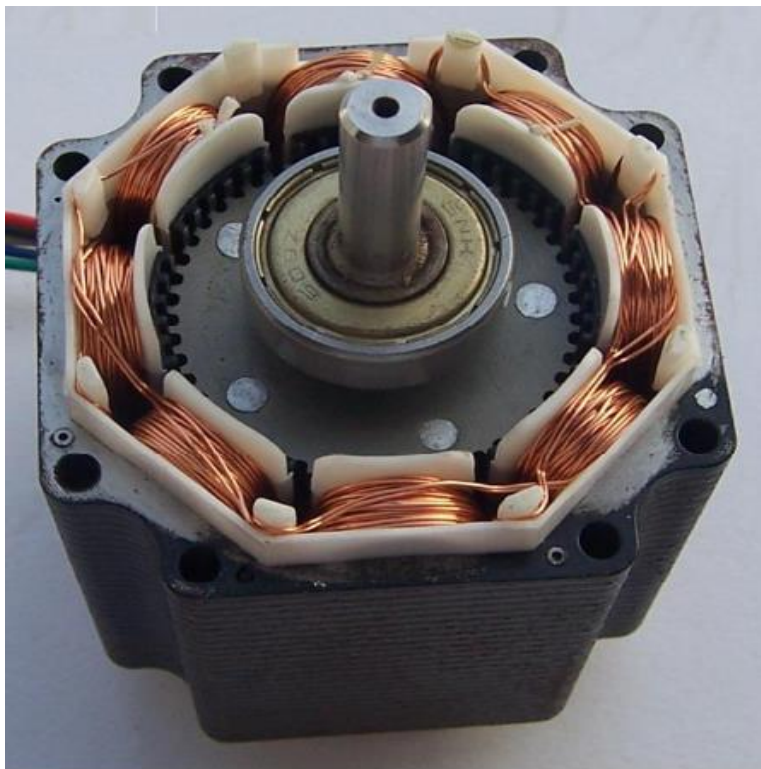
是一种以脉冲控制的转动器件。

步进电机的内部结构：

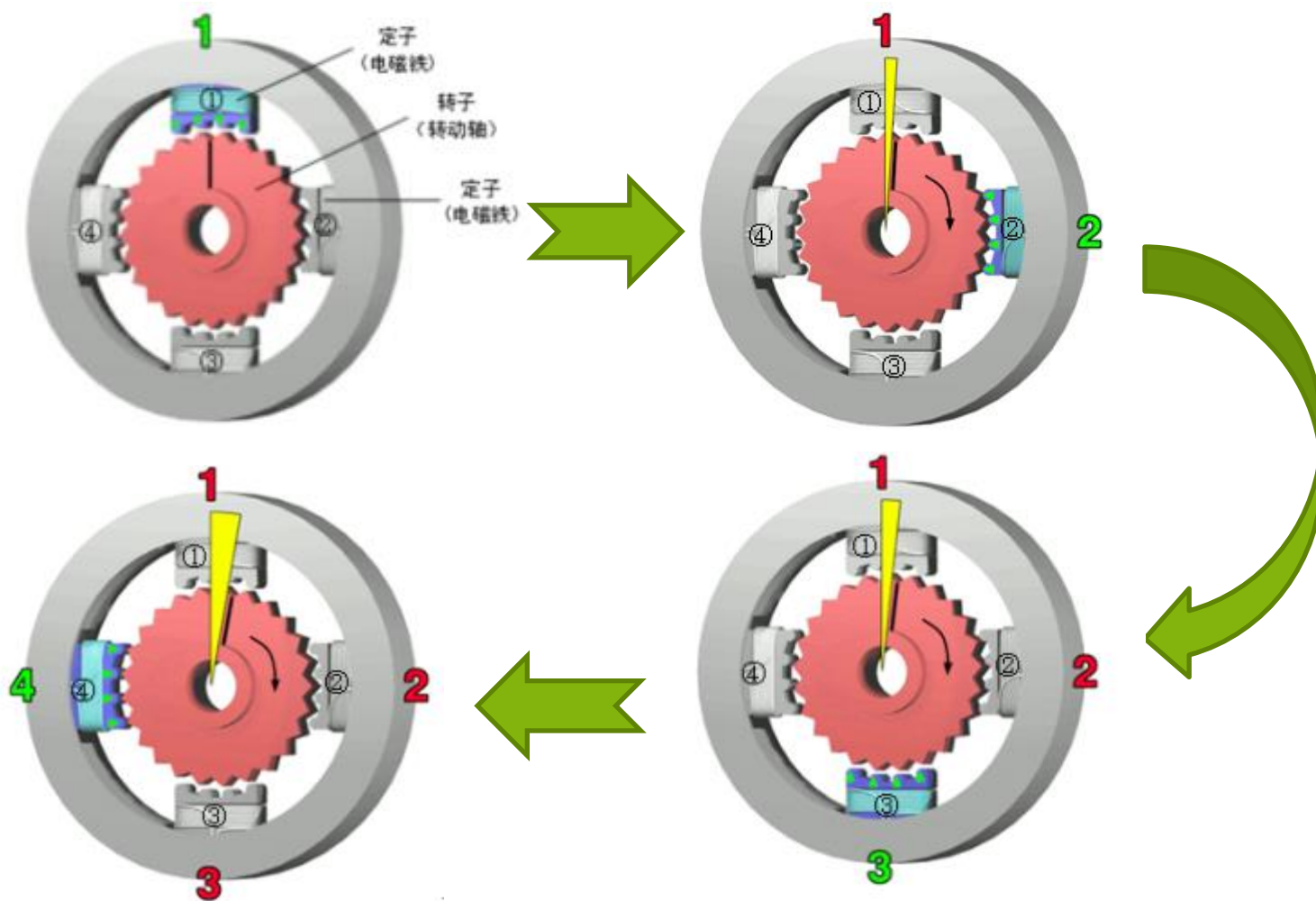
中央有一个齿轮状的转子，转动轴就固定在转子上。转子周围是4个多齿状的电磁铁作为定子（与转子不接触），电磁铁由管脚供电，于是可以被外部电路驱动并控制。



一、步进电机的结构



二、步进电机的驱动方式



二、步进电机的驱动方式

第一步，给电磁铁①通电，于是转子被电磁铁①吸引并转动一个很小的角度，当转子上的齿与电磁铁①的齿对齐后转子停止，步进电机转动了1步。

第二步，当转子上的齿与电磁铁①的齿对齐后，转子上的齿与电磁铁②之间偏离了一个很小的角度。这时给电磁铁②通电而关闭电磁铁①，于是转子被电磁铁②吸引转动一个很小的角度，当转子上的齿与电磁铁②的齿对齐后转子停止，步进电机又转动了1步。

第三步，接下来接通电磁铁③而关闭电磁铁①和②，于是转子又将转动一个很小的角度，步进电机继续转动1步。

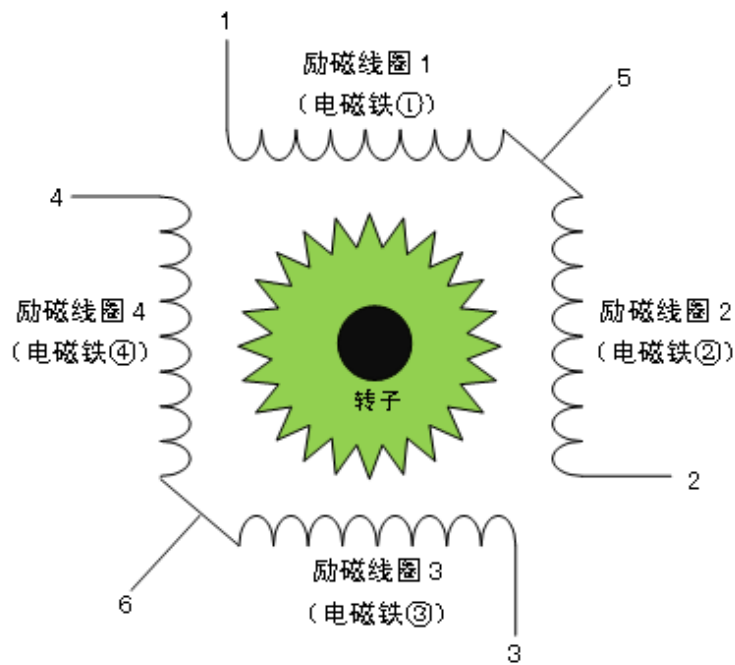
第四步，电磁铁④接通，转子又转动一个很小的角度，步进电机又转动1步。

就这样，电磁铁①、②、③、④快速依次逐一接通电源，转子不断地以很小的角度“跳动”，在转动轴上就形成了步进的表现。

三、步进电机的种类

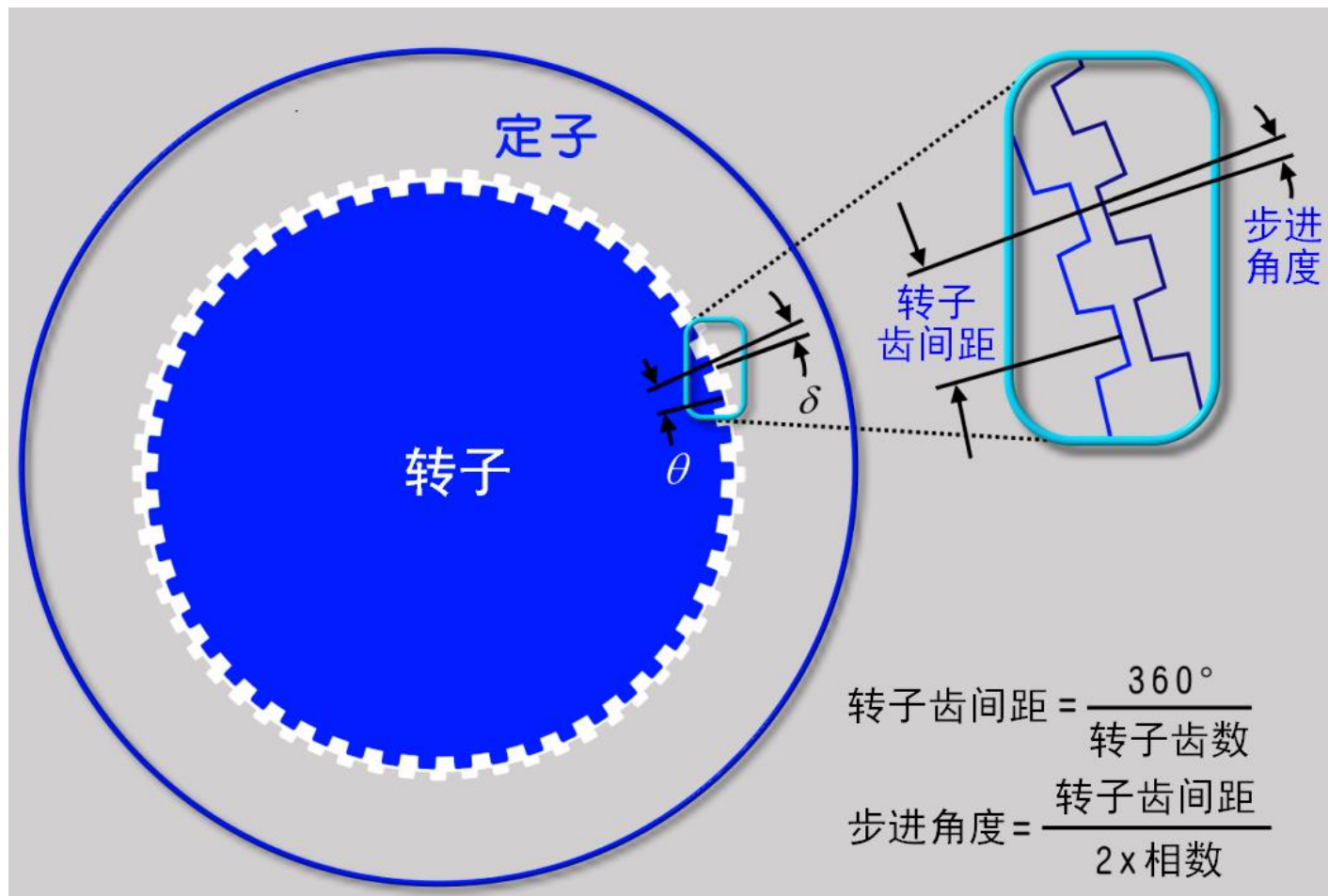
一般分为**单极步进电机**和**双极步进电机**两种。

单极步进电机一般有**5或6**个管脚，在电机内部有**4**个励磁线圈。单极步进电机每个相位上有两个励磁线圈，励磁线圈**1**和励磁线圈**2**是同一相位上的两个励磁线圈，励磁线圈**3**和励磁线圈**4**是另一个相位上的两个励磁线圈。通常，同一相位上的两个励磁线圈有一端连接在一起形成公共端，如管脚**5**和**6**，这样在一个相位上就有**3**个管脚：管脚**1**、**2**、**5**构成一个相位，管脚**3**、**4**、**6**构成另一个相位。在大多数情况下，二相步进电机只有**5**个管脚，这是因为在电机内部，两个相的公共端（管脚**5**和管脚**6**）相连形成了一个公共端。



单极步进电机

四、步进电机的齿间距



五、步进电机的动作

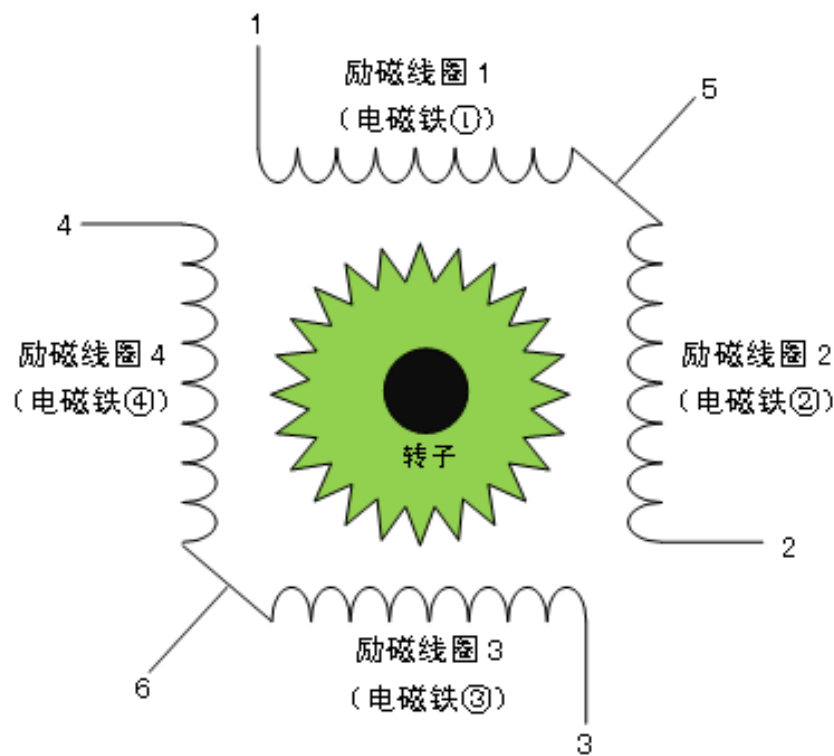
控制方法：想让步进电机运转起来，需要依次给励磁线圈1、2、3、4通电。也就是依次在控制线A、B、C、D上出现高电平，如表示。如果反顺序给励磁线圈通电，则步进电机反转。

三种驱动信号：

1相驱动

2相驱动

1-2相驱动（半步驱动）



五、步进电机的动作

1、1相驱动

——每一时刻只有一个励磁线圈通电，消耗小但转矩小、抖动大。

顺 序	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

单极步进电机正转操作顺序表（1相驱动）

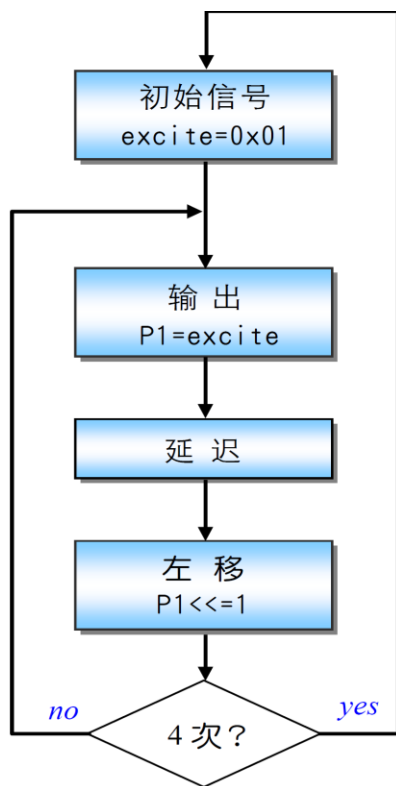
1000 → 0100 → 0010 → 0001 → 1000... (正转)

1000 → 0001 → 0010 → 0100 → 1000... (反转)

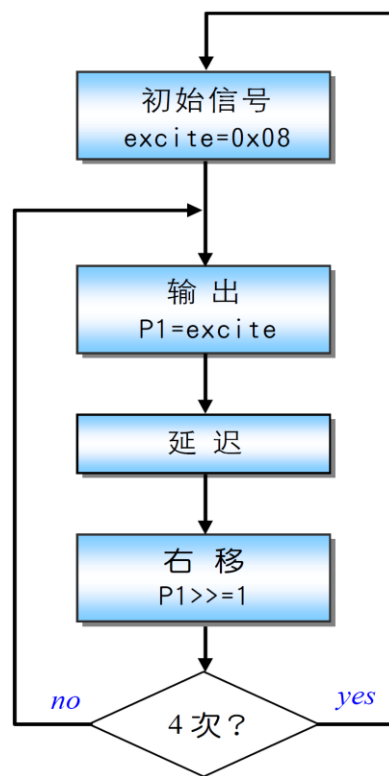
单极步进电机正转、反转驱动方式（1相驱动）

五、步进电机的动作

1、1相驱动



反转



正转

五、步进电机的动作

2、2相驱动

——每一时刻有两组线圈同时被激磁，转矩大、抖动小。

顺 序	A	B	C	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

单极步进电机正转操作顺序表（2相驱动）

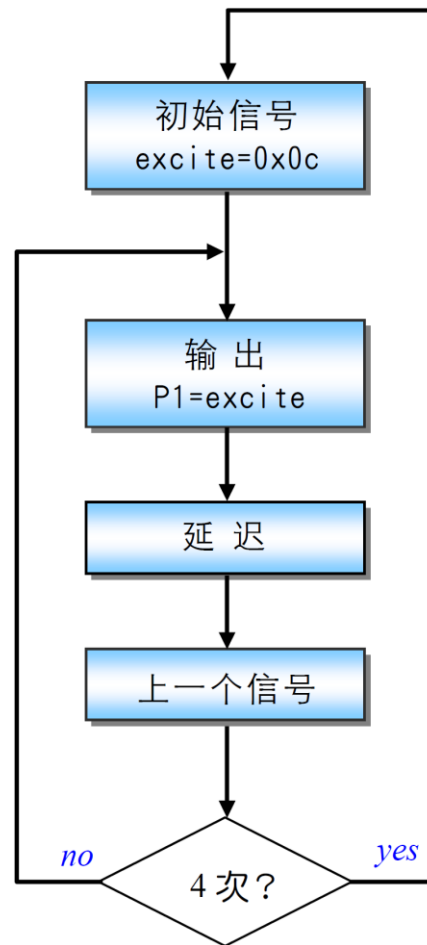
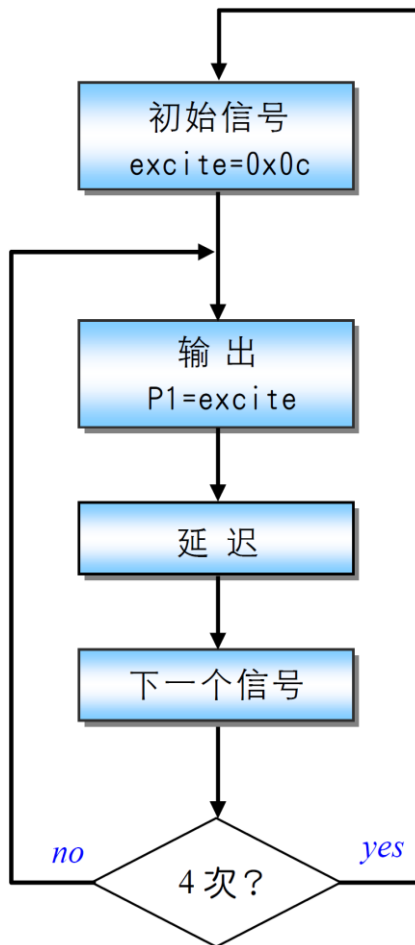
1100 → 0110 → 0011 → 1001 → 1100... (正转)

1100 → 1001 → 0011 → 0110 → 1100... (反转)

单极步进电机正转、反转驱动方式（2相驱动）

五、步进电机的动作

2、2相驱动



五、步进电机的动作

3、1-2相驱动

——前面两种方法的结合，每个驱动信号只驱动半步。
电机运转平滑，步进小。

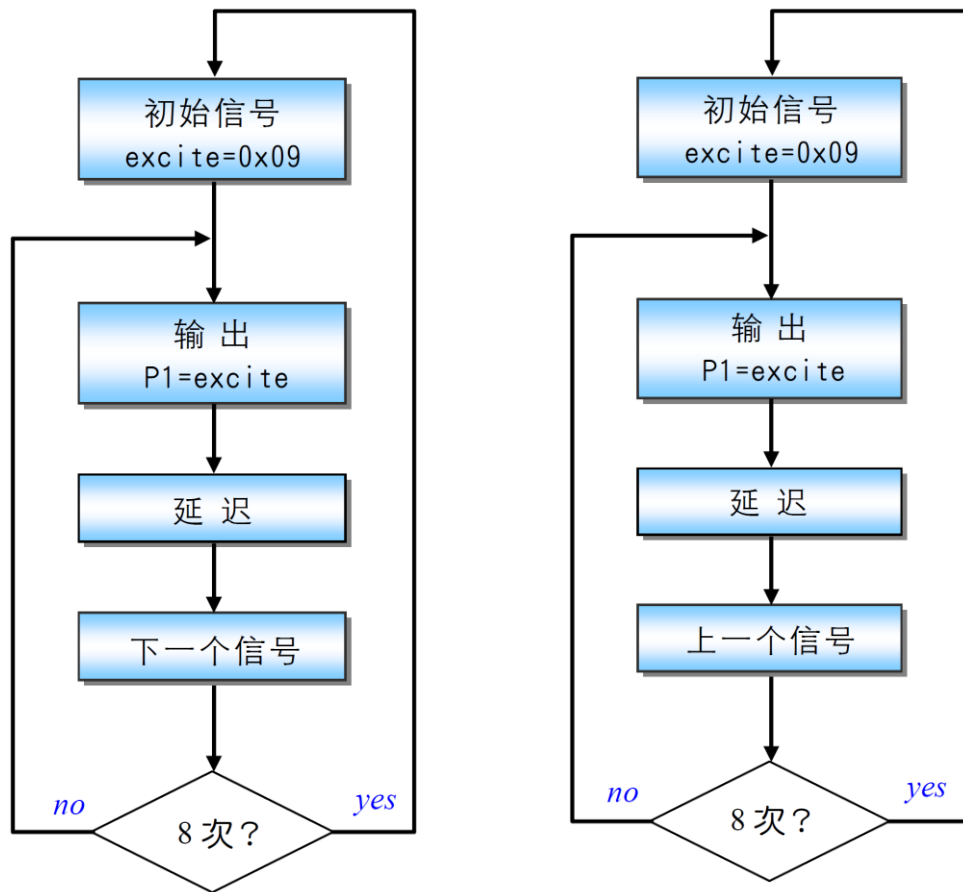
1001 → 1000 → 1100 → 0100 → 0110 → 0010 → 0011 → 0001(正转)

1001 → 0001 → 0011 → 0010 → 0110 → 0100 → 1100 → 1000(反转)

单极步进电机正转、反转驱动方式（1-2相驱动）

五、步进电机的动作

3、1-2相驱动



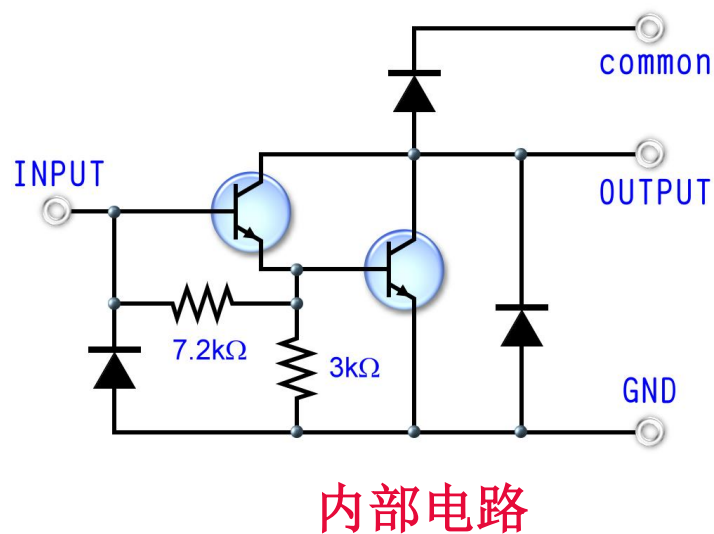
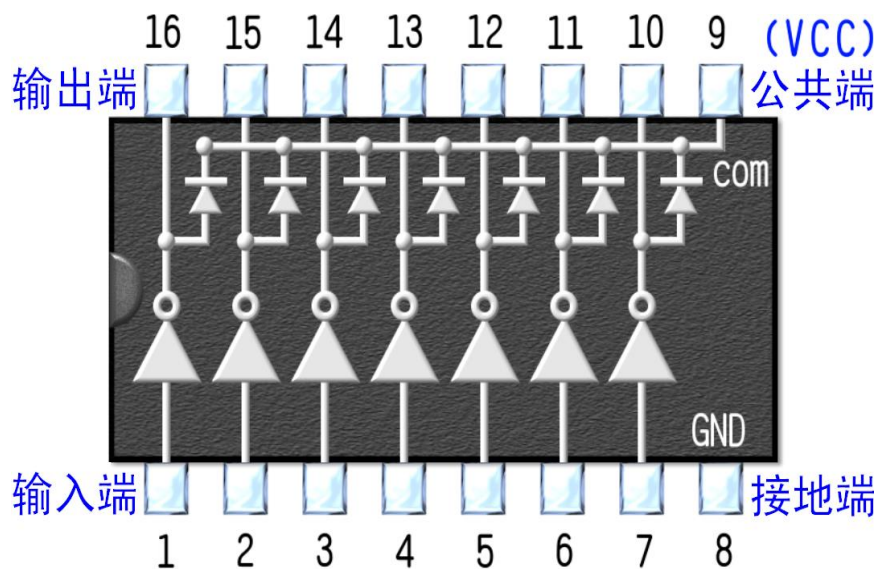
把8个驱动信号放入数组

六、步进电机的驱动电路

- 1、小型步进电机的驱动电路（电流小于0.5A）
- 2、达林顿晶体管驱动电路（电流约1~3A）
- 3、FT5754驱动电路（功能同上，但结构简单）

六、步进电机的驱动电路

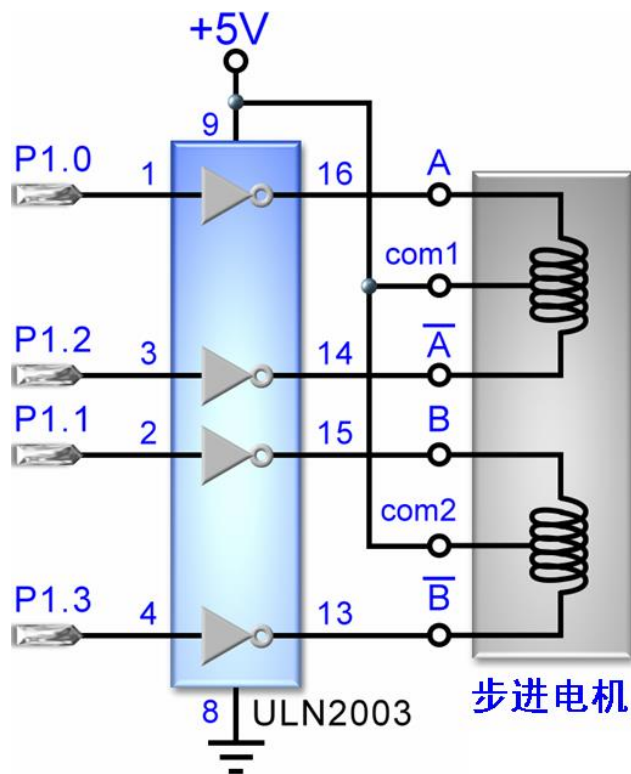
小型步进电机的驱动电路



2001/2002/2003/2004系列驱动IC的引脚图

六、步进电机的驱动电路

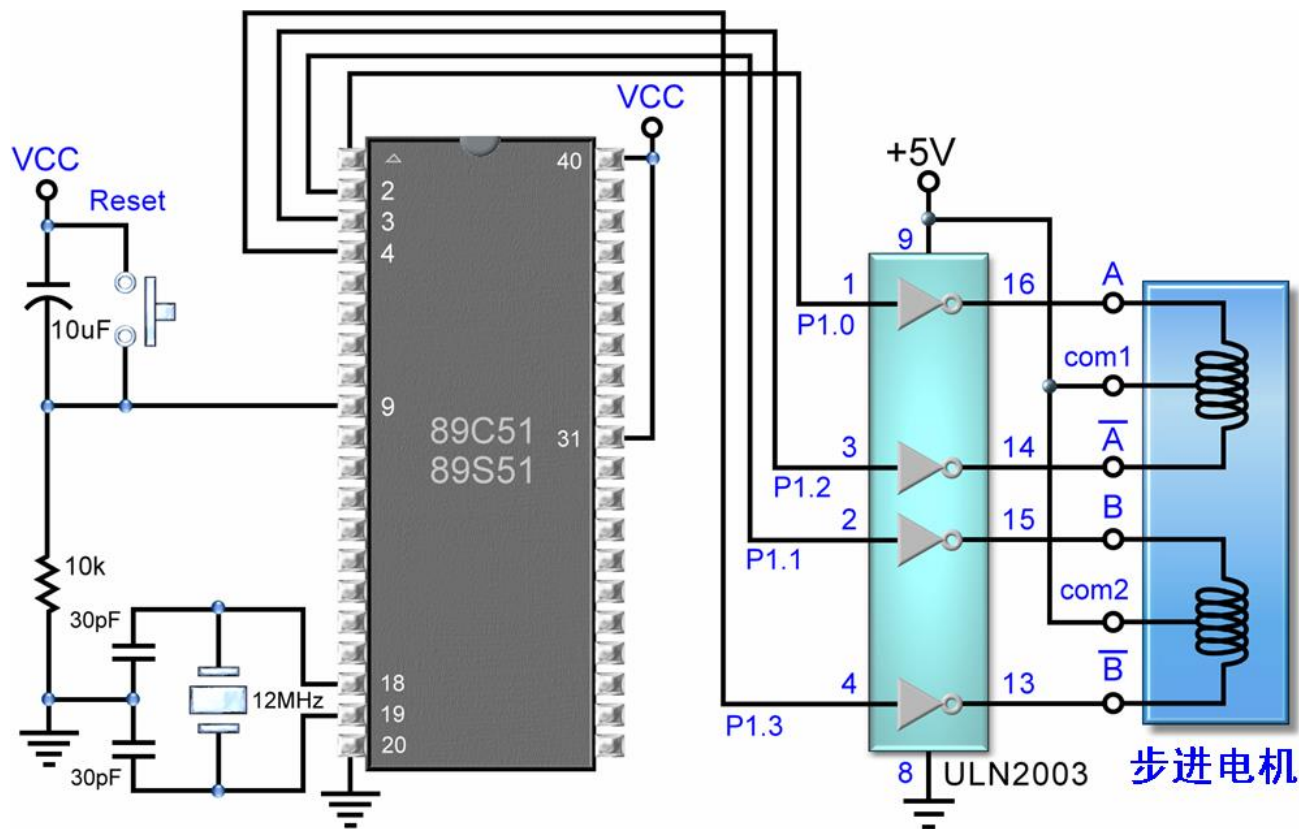
小型步进电机的驱动电路



2003驱动电路

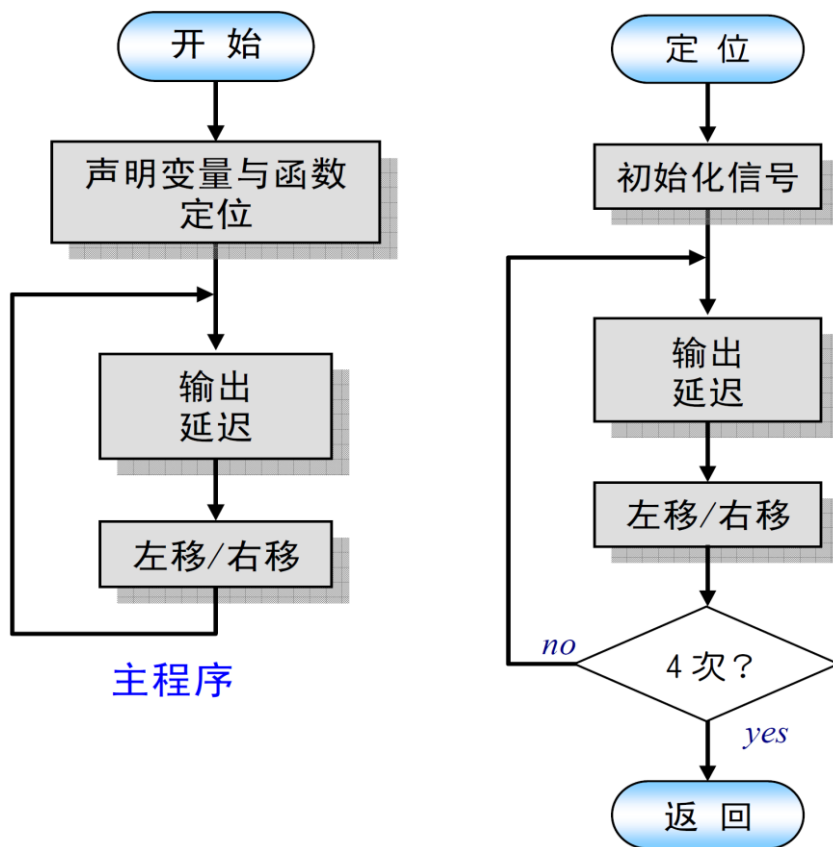
任务一：1相驱动实验

1、电路设计



任务一：1相驱动实验

2、程序设计



任务一：1相驱动实验

```
/* ch10-3-1.c - 1相驱动实验 */
//利用delay5mDELAY子程序(5ms×times),产生驱动信号
//由P1输出    速度为 1/(5ms×timers) 步/秒
#include <reg51.h>                // 包含reg51.h文件
#define OUTPUT    P1              // 定义输出口为P1
unsigned int times=50;            // 声明延迟时间变量(×5ms)
unsigned char excite;             // 声明激磁变量
void step_rst(void);             // 声明定位函数
void delay5ms(int);              // 声明延迟函数
//=====主程序=====
main()                            // 主程序开始
{ OUTPUT=0;                      // 关闭输出
  step_rst();                    // 定位
  while (1)                     // while循环
  { step_rst();                 // 运转
  }                             // while循环结束
}
```

任务一：1相驱动实验

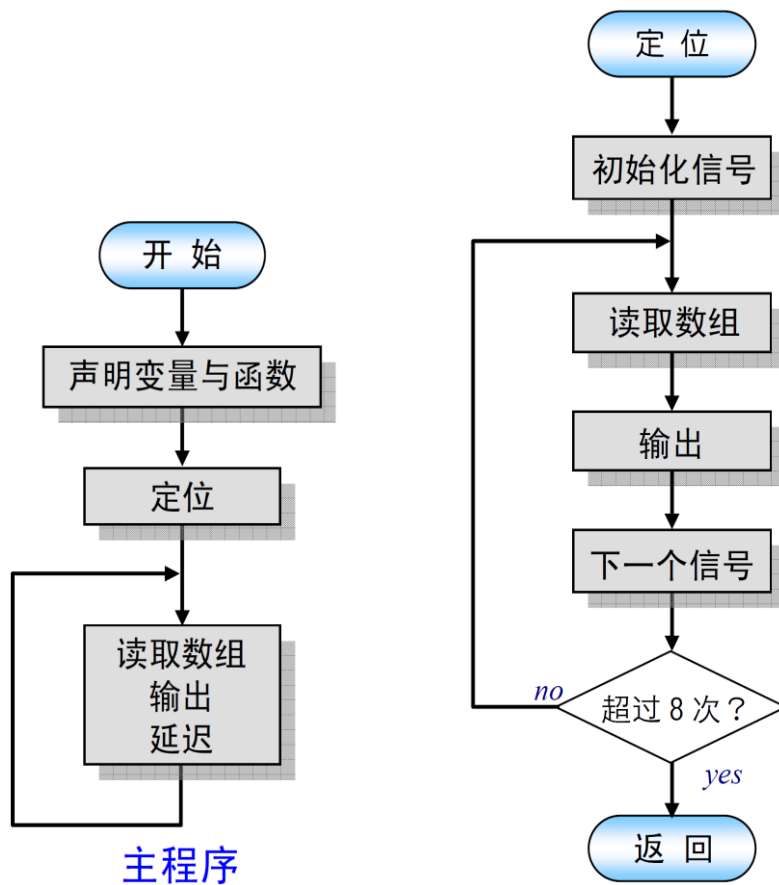
```
//=====定位函数=====
void step_rst(void)
{ char i;
  excite=1;
  for(i=0;i<4;i++)
  { OUTPUT=excite;
    delay5ms(times);
    excite<<=1;
  }
}
//=====延迟函数=====
void delay5ms(int x)
{ int i,j;
  for(i=0;i<x;i++)
    for(j=0;j<600;j++); }

// 定位函数开始
// 声明变量
// 激磁初值
// 输出四个信号
// 输出激磁信号
// 延迟0.25秒
// 左移，下一个激磁
// 结束

// 延迟函数开始
// 声明变量
// i 循环
// j 循环
```

任务二：1-2相驱动实验

2、程序设计



任务二：1-2相驱动实验

```
/* ch10-3-3.c - 1-2 相驱动实验 */
#include <reg51.h> // 包含reg51.h 件
#define OUTPUT P1 // 定义输出口为P1
/* 1-2 相驱动激励信号数组 */
char code excite[]={0x01, 0x03, 0x02, 0x06, 0x04, 0x0c, 0x08, 0x09};
unsigned int times=50; // 声明延迟时间变量(×5ms)
void step_rst(void); // 声明定位函数
void delay5ms(int); // 声明延迟函数
main() // 主程序
{ OUTPUT=0; // 关闭输出
  step_rst(); // 定位
  delay5ms(200); // 暂停1秒钟
  while (1) // while循环
  { step_rst(); // 运转
  } // 结束主程序
```

任务二：1-2相驱动实验

2、程序设计

```
//=====
void step_rst(void)                // 定位函数开始
{ char i;                          // 声明变量
  for(i=0;i<8;i++)                 // 输出8个信号
  { OUTPUT=excite[i];              // 输出激磁信号
    delay5ms(times);               // 延迟0.25秒
  }                                 // 结束
}
//=====
void delay5ms(int x)                // 延迟函数开始
{ int i,j;                          // 声明变量
  for(i=0;i<x;i++)                 // i 循环
    for(j=0;j<600;j++);           // j 循环
}
```