

## 舵机原理应用程序详解

### 1、舵机实物图片



图 1 普通航模用舵机



图 2 微型舵机

### 2、舵机介绍

舵机英文叫 Servo，也称伺服机。其特点是结构紧凑、易安装调试、控制简单、大扭力、成本较低等。舵机的主要性能取决于最大力矩和工作速度（一般是以秒/60°为单位）。它是一种位置伺服的驱动器，适用于那些需要角度不断变化并能够保持的控制系统。在机器人机电控制系统中，舵机控制效果是性能的重要影响因素。舵机能够在微机电系统和航模中作为基本的输出执行机构，其简单的控制和输出使得单片机系统很容易与之接口。

### 3、舵机的工作原理

标准的舵机有 3 条引线，分别是：电源线 Vcc、地线 GND 和控制信号线，如图 3 所示。

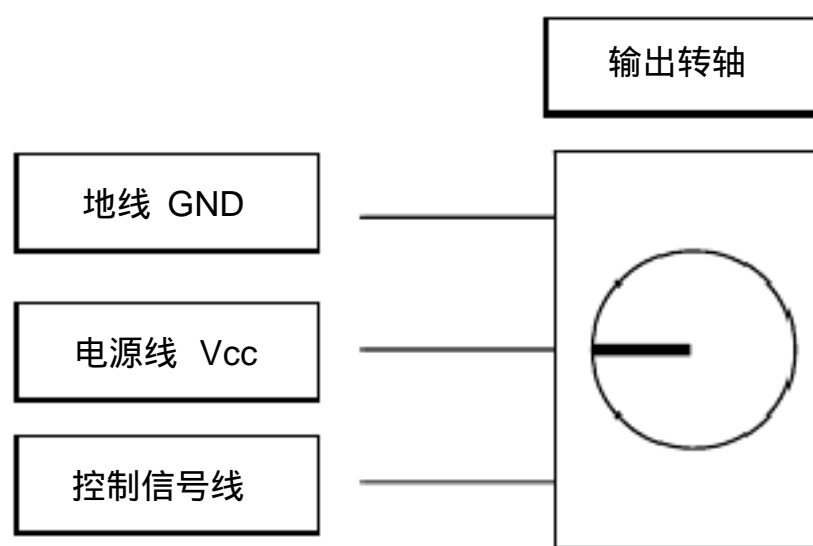


图 3 标准舵机引线示意图

在航模遥控系统中，控制信号由接收机的通道进入信号调制芯片，获得直流偏置电压。他内部有一个基准电路，产生周期为 20ms，宽度为 1.5ms 的基准信号，将获得的直流偏置电压和电位器的电压比较，获得电压差输出。最后，电压差的正负输出到电机驱动芯片决定电机的正反转。当电机转速一定时，通过级联减速齿轮带动电位器旋转，使得电压差为 0，电机停止转动。其实我们可以不用去了解它内部的具体工作原理，知道它的控制原理就够了。就象我们使用三极管一样，知道可以拿它来做开关管或放大管就行了，至于管内的电子具体怎么流动是可以完全不用去考虑的。舵机的控制信号也是 PWM 信号，利用占空比的变化改变舵机的位置。图 4 为舵机输出转角与输入信号脉冲宽度的关系，其脉冲宽度在 0.5 ~ 2.5ms 之间变化时，舵机输出轴转角在 0° ~ 180° 之间变化。

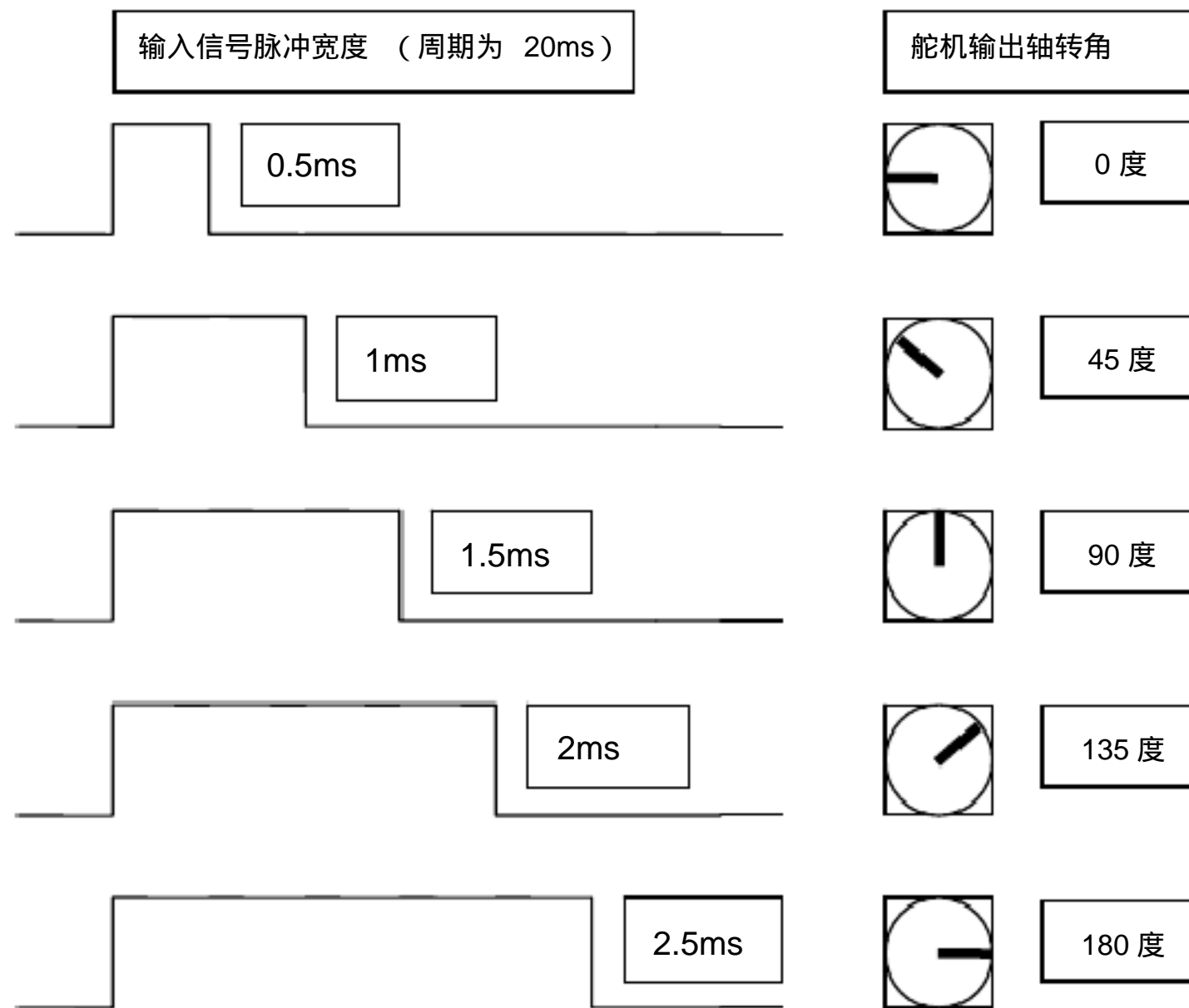


图 4 舵机输出转角与输入信号脉冲宽度的关系

#### 4、用单片机实现舵机转角控制

单片机系统实现对舵机输出转角的控制，必须首先完成两个任务：首先是产生基本的 PWM 周期信号，即产生 20ms 的周期信号；其次是脉宽的调整，即单片机调节 PWM 信号的占空比。

单片机作为舵机的控制部分，能使 PWM 信号的脉冲宽度实现微秒级的变化，从而提高舵机的转角精度。单片机完成控制算法，再将计算结果转化为 PWM 信号输出到舵机，由于单片机系统是个数字系统，其控制信号的变化完全依靠硬件计数，所以受外界干扰较小，整个系统工作可靠。

单片机控制单个舵机是比较简单的，利用一个定时器即可，假设仅控制舵机 5 个角度转动，其控制思路如下：只利用一个定时器 T0，定时时间为 0.5ms，定义一个角度标识，数值可以为 1、2、3、4、5，实现 0.5ms、1ms、1.5ms、2ms、2.5ms 高电平的输出，再定义一个变量，数值最大为 40，实现周期为 20ms。每次进入定时中断，判断此时的角度标识，进行相应的操作。比如此时为 5，则进入的前 5 次中断期间，信号输出为高电平，即为 2.5ms 的高电平。剩下的 35 次中断期间，信号输出为低电平，即为 17.5ms 的低电平。这样总的周期是 20ms，为一个周期。

当用单片机系统控制多个舵机工作时，可以参考下以方法：以驱动 8 路舵机为例，假设使用的舵机工作周期均为 20ms 时，那么用单片机定时器产生的多路 PWM 波的周期也相同。使用单片机的内部定时器产生脉冲计数，一般来说，舵机工作正脉冲宽度小于周期的 1/8，这样能够在 1 个周期内分时启动各路 PWM 波的上升沿，再利用定时器中断 T0 确定各路 PWM 波的输出宽度，定时器中断 T1 控制 20ms 的基准时间。第 1 次定时器中断 T0 按 20ms 的 1/8 配置初值，并配置输出 I/O 口，第 1 次 T0 定时中断响应后，将当前输出 I/O 口对应的引脚输出置高电平，配置该路输出正脉冲宽度，并启动第 2 次定时器中断，输出 I/O 口指向下一个输出口。第 2 次定时器定时时间结束后，将当前输出引脚置低电平，配置此中断周

期为 20ms 的 1/8 减去正脉冲的时间，此路 PWM 信号在该周期中输出完毕，往复输出。在每次循环的第 16 次( $2 \times 8=16$ )中断实行关定时中断 T0 的操作，最后就能够实现 8 路舵机控制信号的输出。

### 5、舵机与单片机连接原理图

在用单片机驱动舵机之前，要先确定相应舵机的功率，然后选择足够功率的电源为舵机供电，控制端无需大电流，直接用单片机的 I/O 口就可操作，扩展板上舵机信号线接单片机的 P1.7，舵机与单片机连接原理图如图 5 所示。

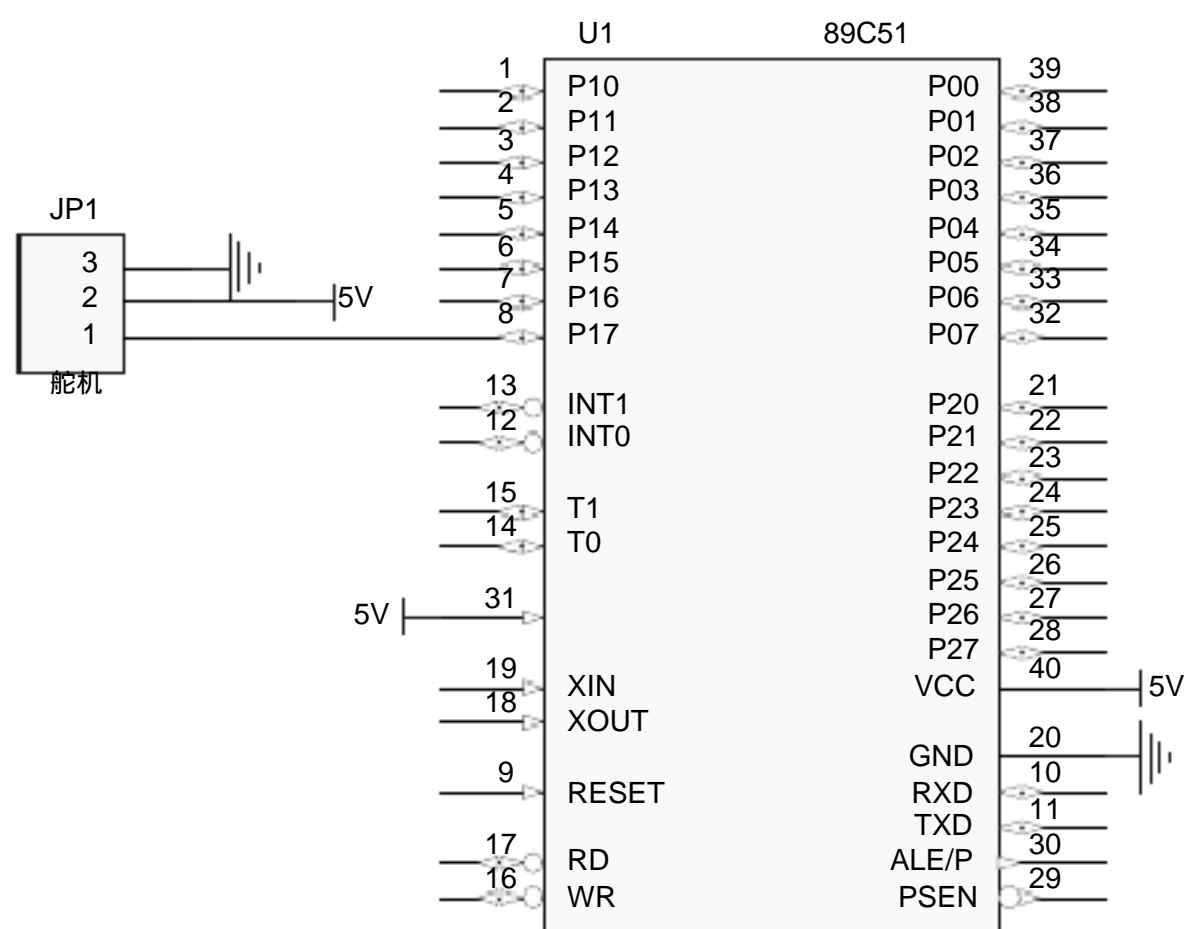


图5 舵机和单片机连接原理图

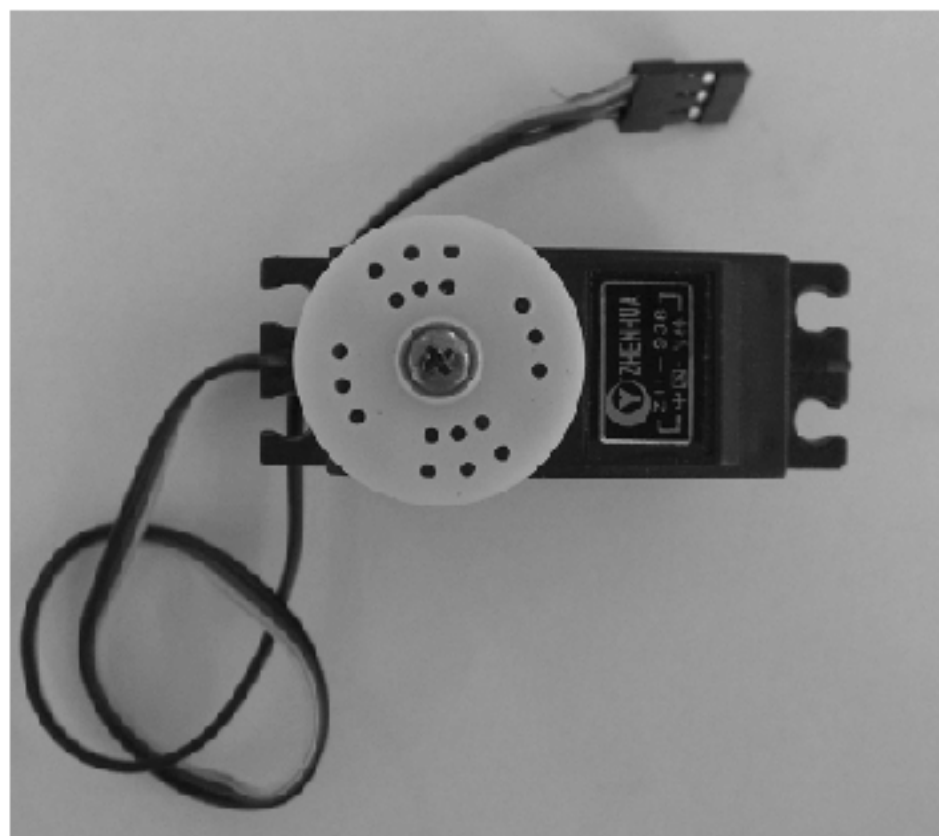


图6 本实验使用的舵机

本实验中使用的舵机参数如下：工作电压：4.8V ~ 6V；电流：10 mA(静态)；力矩：3kg/cm；外型尺寸：41×42×20 mm；重量：48g；转速:0.22ms/60°。

## 6、舵机 C语言程序实例

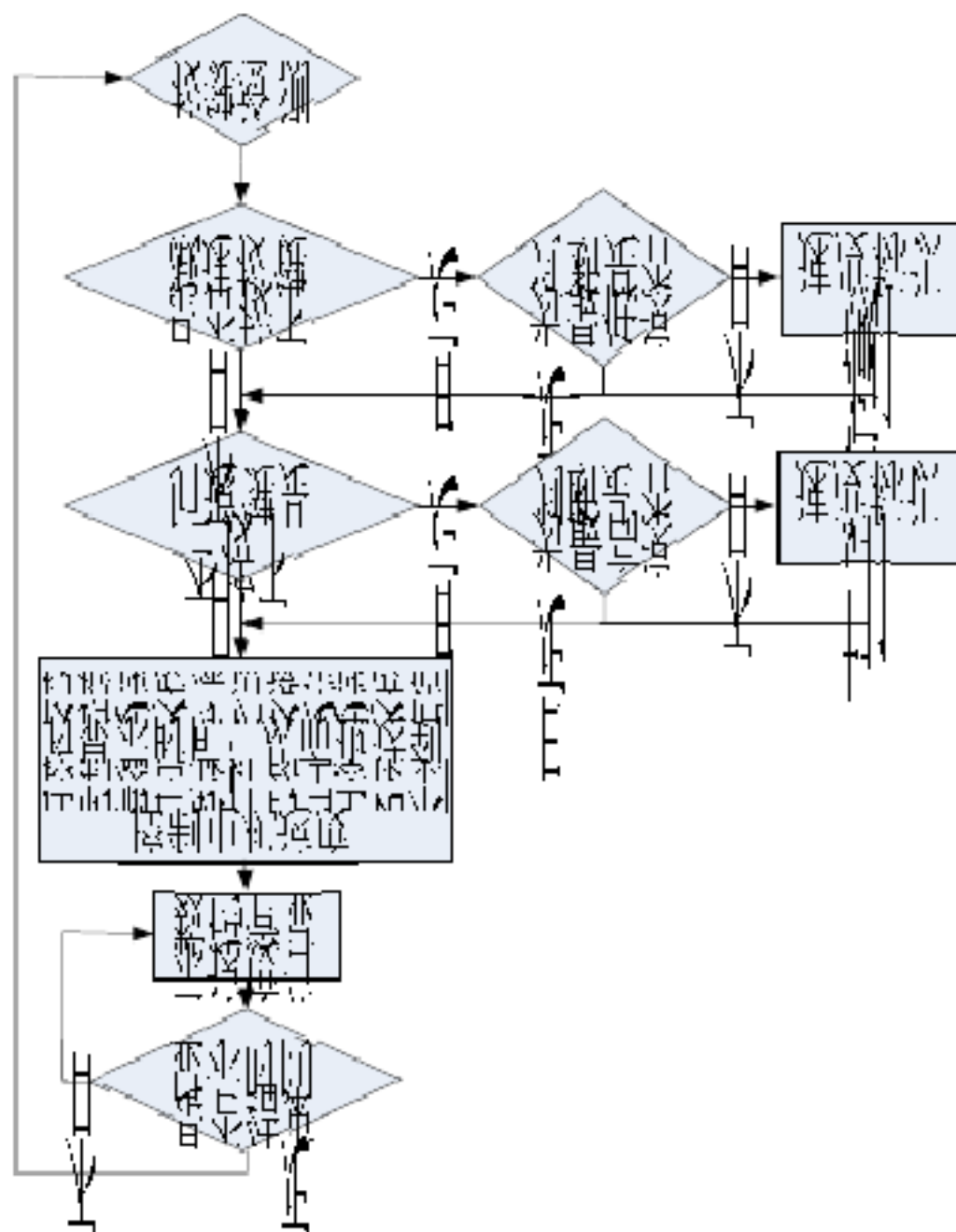


图8 程序流程图

实验说明：开机时舵机角度自动转为 0度，通过实验板上的独立按键调节舵机的角度转动，并且在实验板上数码管上显示出相应的角度， 本实验仅演示 5个角度的控制，若想实验任意角度控制请大家自行编程实验。

```
#include "reg52.h"
unsigned char count;    //0.5ms    次数标识
sbit pwm =P1^7;        //PWM 信号输出
sbit jia =P3^7;         //角度增加按键检测 IO 口
sbit jian =P3^6;        //角度减少按键检测 IO 口
unsigned char jd;       //    角度标识
sbit dula=P2^6;
sbit wela=P2^7;
unsigned char code table[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,
                             0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
void delay(unsigned char i)// 延时
{
    unsigned char j,k;
    for(j=i;j>0;j--)
        for(k=125;k>0;k--);
}
void Time0_Init()        //定时器初始化
{
    TMOD = 0x01;         //    定时器 0工作在方式 1
    IE=    0x82;
```

```
    TH0=    0xfe;
    TL0= 0x33;          //11.0592MHz 晶振, 0.5ms
    TR0=1;              //定时器开始
}
void Time0_Int() interrupt 1 // 中断程序
{
    TH0  = 0xfe;          //重新赋值
    TL0  = 0x33;
    if(count<jd)          //判断 0.5ms次数是否小于角度标识
        pwm=1;            //确实小于, PWM 输出高电平
    else
        pwm=0;            //大于则输出低电平
        count=(count+1);  //0.5ms 次数加 1
        count=count%40;   // 次数始终保持为 40 即保持周期为 20ms
}
void keyscan()           //按键扫描
{
    if(jia==0)            //角度增加按键是否按下
    {
        delay(10);        //按下延时, 消抖
        if(jia==0)        //确实按下
        {
            jd++;          //角度标识加 1
            count=0;        // 按键按下 则 20ms周期从新开始
            if(jd==6)
                jd=5;      //已经是 180度, 则保持
                while(jia==0); //等待按键放开
        }
    }
    if(jan==0)            //角度减小按键是否按下
    {
        delay(10);
        if(jan==0)
        {
            jd--;          //角度标识减 1
            count=0;
            if(jd==0)
                jd=1;      //已经是 0度, 则保持
                while(jan==0);
        }
    }
}
void display()           //数码管显示函数
{

```



```
unsigned char bai,shi,ge;
switch(jd)          //根据角度标识显示相应的数值
{
    case 1:          //为 1，角度为 0，前 3 个数码管显示 000
        bai=0;
        shi=0;
        ge=0;
        break;
    case 2:
        bai=0;
        shi=4;
        ge=5;
        break;
    case 3:
        bai=0;
        shi=9;
        ge=0;
        break;
    case 4:
        bai=1;
        shi=3;
        ge=5;
        break;
    case 5:          //为 5，角度为 180，前 3 个数码管显示 180
        bai=1;
        shi=8;
        ge=0;
        break;
}
dula=0;
P0=table[bai];
dula=1;
dula=0;
wela=0;
P0=0xfe;
wela=1;
wela=0;
delay(5);
P0=table[shi];
dula=1;
dula=0;
P0=0xfd;
wela=1;
wela=0;
```

```
    delay(5);
    P0=table[ge];
    dula=1;
    dula=0;
    P0=0xfb;
    wela=1;
    wela=0;
    delay(5);
}
void main()
{
    jd=1;
    count=0;
    Time0_Init();
    while(1)
    {
        keyscan();           // 按键扫描
        display();
    }
}
```