

## 舵机的相关原理与控制原理

## 什么是舵机：

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。舵机可以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非常容易与之接口。舵机是一种位置（角度）伺服的驱动器，适用于那些需要角度不断变化并可以保持的控制系统。目前在高档遥控玩具，如航模，包括飞机模型，潜艇模型；遥控机器人中已经使用得比较普遍。舵机是一种俗称，其实是一种伺服马达。



## 工作原理：

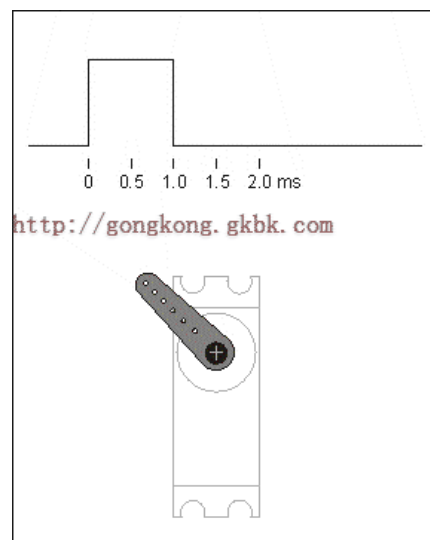
控制信号由接收机的通道进入信号调制芯片，获得直流偏置电压。它内部有一个基准电路，产生周期为20ms，宽度为1.5ms的基准信号，将获得的直流偏置电压与电位器的电压比较，获得电压差输出。最后，电压差的正负输出到电机驱动芯片决定电机的正反转。当电机转速一定时，通过级联减速齿轮带动电位器旋转，使得电压差为0，电机停止转动。当然我们可以不用去了解它的具体工作原理，知道它的控制原理就够了。就象我们使用晶体管一样，知道可以拿它来做开关管或放大管就行了，至于管内的电子具体怎么流动是可以完全不用去考虑的。

## 舵机的控制：

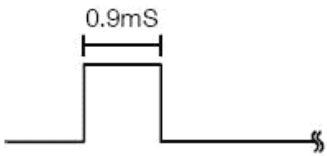

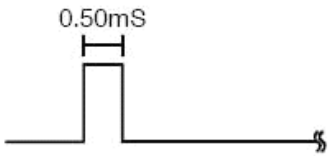

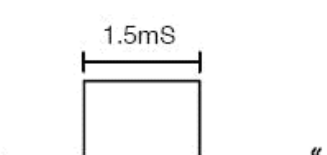

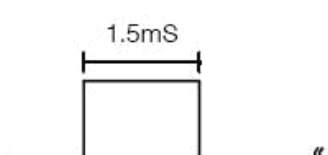

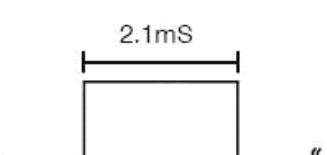

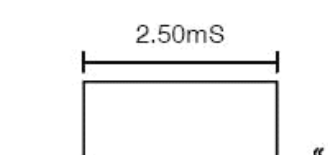

舵机的控制一般需要一个20ms左右的时基脉冲，该脉冲的高电平部分一般为0.5ms~2.5ms范围内的角度控制脉冲部分。以180度角度伺服为例，那么对应的控制关系是这样的：

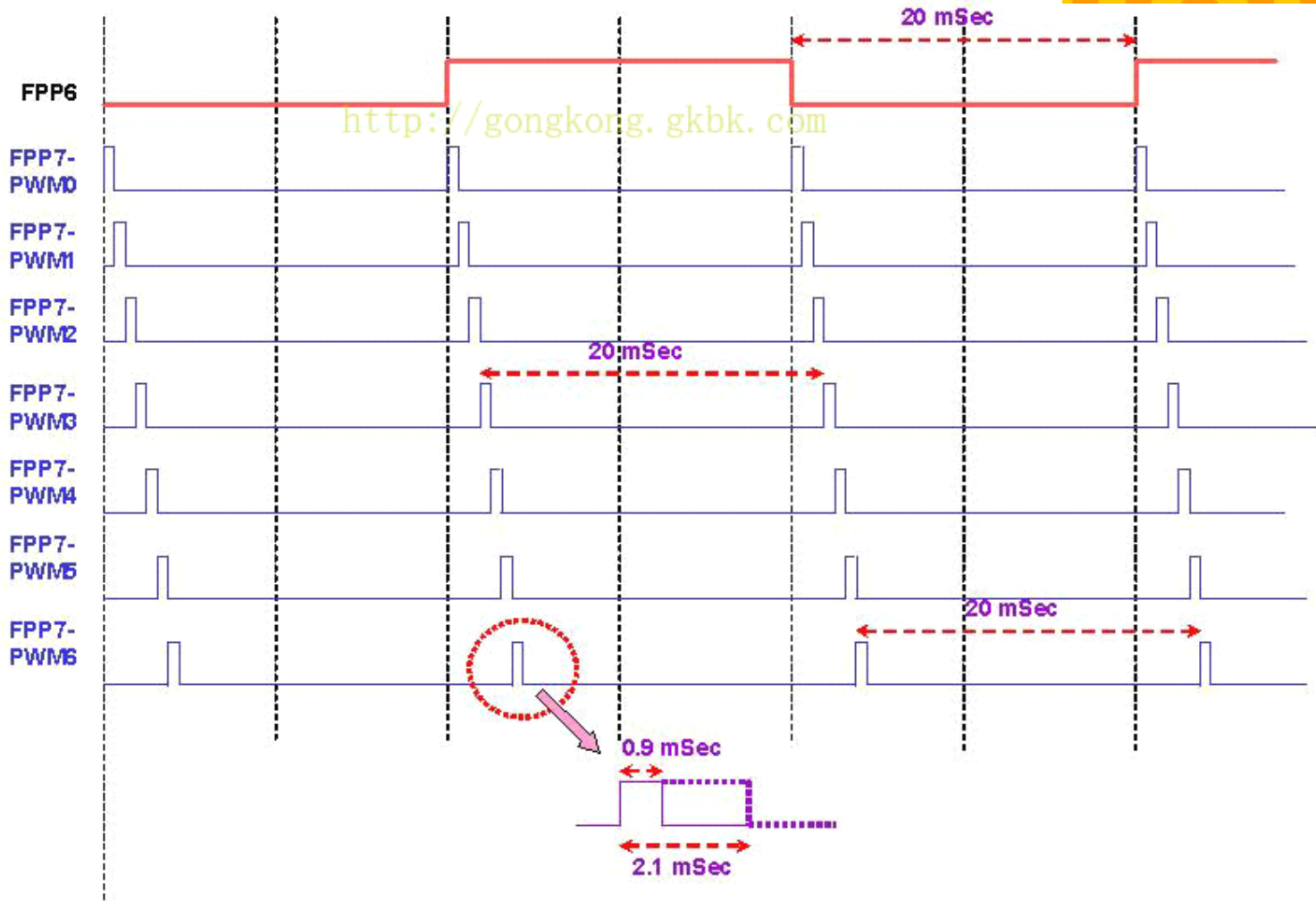
0.5ms-----0度；  
1.0ms-----45度；  
1.5ms-----90度；  
2.0ms-----135度；  
2.5ms-----180度；

这只是一种参考数值，具体的参数，请参见舵机的技术参数。小型舵机的工作电压一般为4.8V或6V，转速也不是很快，一般为0.22/60度或0.18/60度，所以假如你更改角度控制脉冲的宽度太快时，舵机可能反应不过来。如果需要更快速的反应，就需要更高的转速了。要精确的控制舵机，其实没有那么容易，很多舵机的位置等级有1024个，那么，如果舵机的有效角度范围为180度的话，其控



制的角度精度是可以达到180/1024度约0.18度了，从时间上看其实要求的脉宽控制精度为2000/1024us约2us。如果你拿了一个舵机，连控制精度为1度都达不到的话，而且还看到舵机在发抖。在这种情况下，只要舵机的电压没有抖动，那抖动的就是你的控制脉冲了。而这个脉冲为什么会抖动呢？当然和你选用的脉冲发生器有关了。一些前辈喜欢用555来调舵机的驱动脉冲，如果只是控制几个点位置伺服好像是可以这么做的，可以多用几个开关引些电阻出来调占空比，这么做简单吗，应该不会啦，调试应该是非常麻烦而且运行也不一定可靠的。其实主要还是他那个年代，单片机这东西不流行呀，哪里会哟！使用传统单片机控制舵机的方案也有很多，多是利用定时器和中断的方式来完成控制的，这样的方式控制1个舵机还是相当有效的，但是随着舵机数量的增加，也许控制起来就没有那么方便而且可以达到约2微秒的脉宽控制精度了。听说AVR也有控制32个舵机的试验板，不过精度能不能达到2微秒可能还是要泰克才知道了。其实测试起来很简单，你只需要将其控制信号与示波器连接，然后让试验板输出的舵机控制信号以2微秒的宽度递增。为什么FPPA就可以很方便地将脉宽的精度精确地控制在2微秒甚至2微秒一下呢。主要还是 `delay memory` 这样的具有创造性的指令发挥了功效。该指令的延时时间为数据单元中的立即数的值加1个指令周期（数据0出外，详情请参见`delay`指令使用注意事项）因为是8位的数据存储单元，所以`memory`中的数据为（0~255），记得前面有提过，舵机的角度级数一般为1024级，所以只用一个存储空间来存储延时参数好像还不够用的，所以我们可以采用2个内存单元来存放舵机的角度伺服参数了。所以这样一来，我们可以采用这样的软件结构了：

OurRobot      Normal Range	Extended Range
  <div>Servo -45°</div>	  <div>Servo -90°</div>
  <div>Servo Centered</div>	  <div>Servo Centered</div>
  <div>Servo +45°</div>	  <div>Servo +90°</div>



舵机驱动的应用场合：

高档遥控仿真车，至少得包括左转和右转功能，高精度的角度控制，必然给你最真实的驾车体验。

多自由度机器人设计，为什么日本人设计的机器人可以上万RMB的出售，而国内设计的一些两三千块也卖不出去呢，还是一个品质的问题。

多路伺服航模控制，电动遥控飞机，油动遥控飞机，航海模型等



[www.HiModel.com](http://www.HiModel.com)



[www.HiModel.com](http://www.HiModel.com)

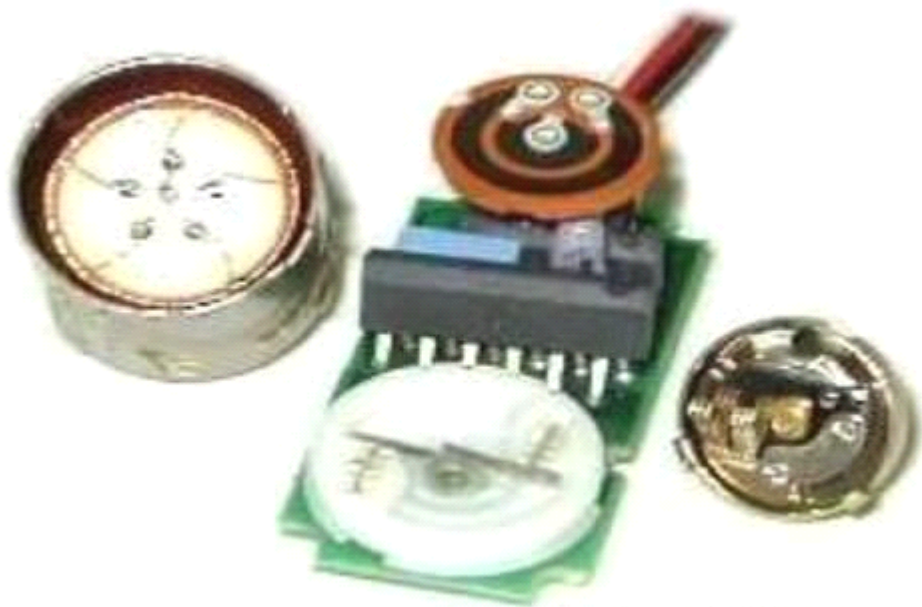


[www.HiModel.com](http://www.HiModel.com)

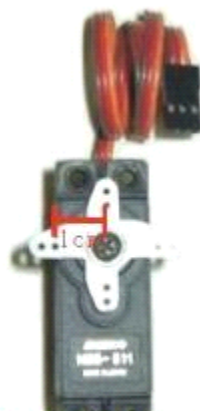




www.HiModel.com

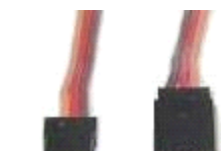


www.HiModel.com



HiModel.com

三条线分别为电源线、接地线与讯号线。电源线：红色，接地线：棕色，讯号线：橙色



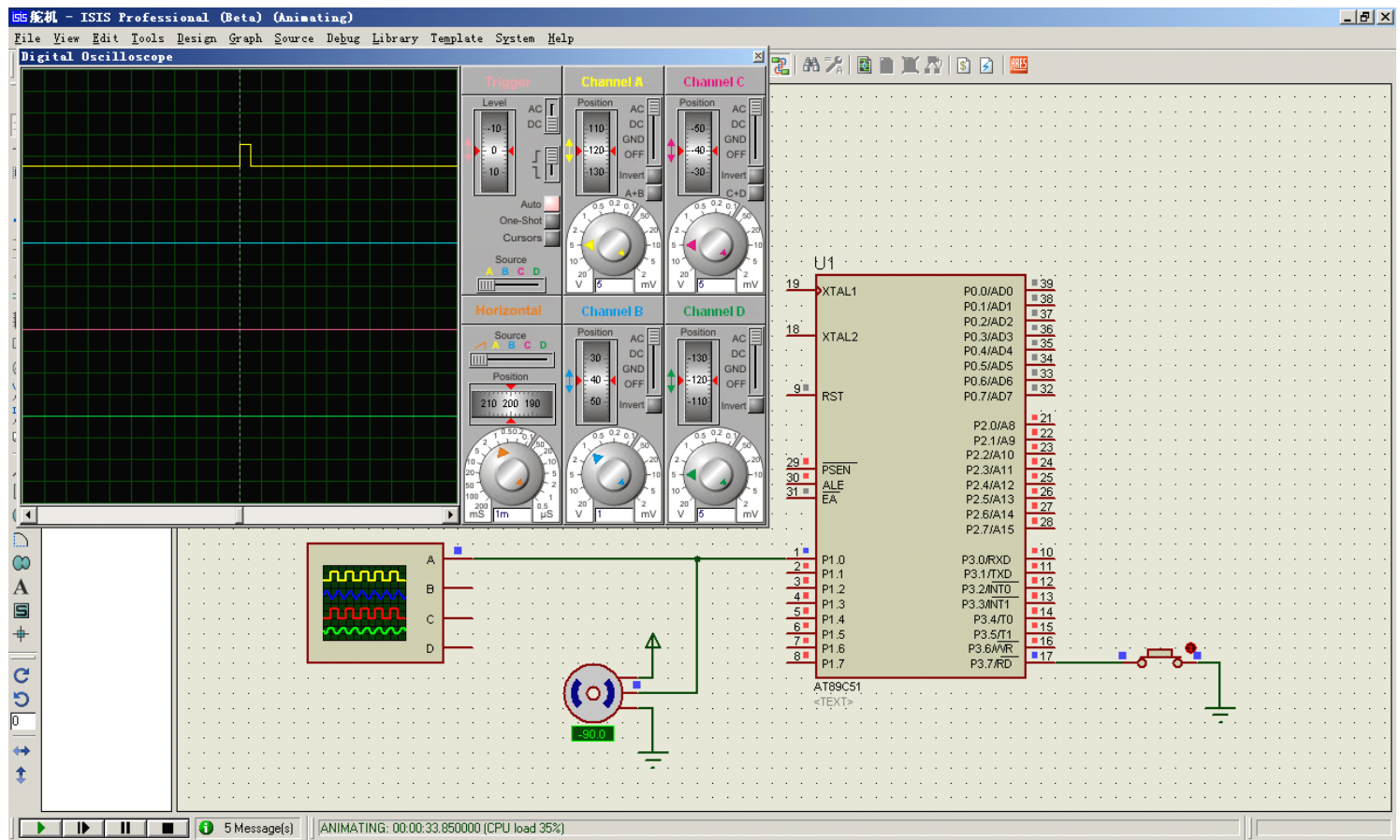
HiModel.com

三条线分别为电源线、接地线与讯号线。电源线：红色，接地线：黑色，讯号线：白色



HiModel.com

测试电路图：



源程序如下：

<duoji.h>

void delay(unsigned int t)//50ms 延时

```
{
    unsigned char j;
    for(;t>0;t--)
        for(j=19;j>0;j--);
}
```

void Degree0()//0度

```
{
    unsigned char i;
    for(i=0;i<200;i++)
    {
        P1_0=1;
        delay(10);//0.5ms
        P1_0=0;
    }
}
```

```
        delay(390);//19.5ms
    }
}
```

**void Degree45()//45度**

```
{
    unsigned char i;
    for(i=0;i<200;i++)
    {
        P1_0=1;
        delay(20);//1ms
        P1_0=0;
        delay(380);//19ms
    }
}
```

**void Degree90()//90度**

```
{
    unsigned char i;
    for(i=0;i<200;i++)
    {
        P1_0=1;
        delay(30);//1.5ms
        P1_0=0;
        delay(370);//18.5ms
    }
}
```

**void Degree135()//135度**

```
{
    unsigned char i;
    for(i=0;i<200;i++)
    {
        P1_0=1;
        delay(40);//2ms
        P1_0=0;
        delay(360);//18ms
    }
}
```

```
void Degree180()//180度
{
    unsigned char i;
    for(i=0;i<200;i++)
    {
        P1_0=1;
        delay(50);//2.5ms
        P1_0=0;
        delay(350);//17.5ms
    }
}
```

## <duoji.c>

```
#include <AT89X51.H>
#include <duoji.H>
```

```
void delay10ms(void)
{
    unsigned char i,j;
    for(i=20;i>0;i--)
    for(j=248;j>0;j--);
}
```

```
void main(void)
{
    unsigned char ID;
    while(1)
    {
        if(P3_7==0)
        {
            delay10ms();
            if(P3_7==0)
            {
                ID++;
                if(ID==5)
                {
                    ID=0;
                }
            }
        }
    }
}
```



```
}  
while(P3_7==0);
```

```
}
```

```
}
```

```
switch(ID)
```

```
{
```

```
case 0:
```

```
Degree0();
```

```
break;
```

```
case 1:
```

```
Degree180();
```

```
break;
```

```
case 2:
```

```
Degree90();
```

```
break;
```

```
case 3:
```

```
Degree180();
```

```
break;
```

```
case 4:
```

```
Degree0();
```

```
break;
```

```
}
```

```
}
```

```
}
```