

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。其工作原理是：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。

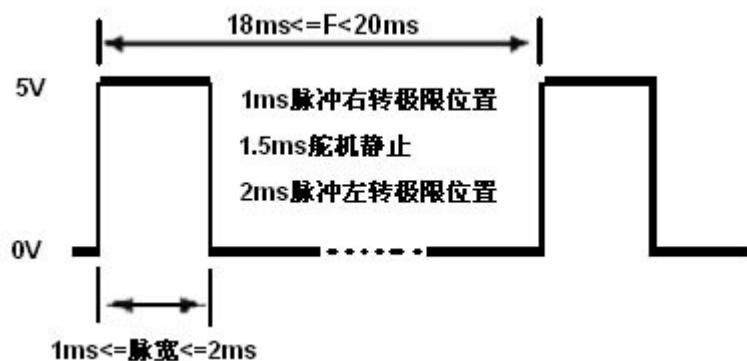


图 1 舵机的控制要求

舵机的控制信号是 **PWM** 信号，利用占空比的变化改变舵机的位置。一般舵机的控制要求如图 1 所示。

单片机实现舵机转角控制

可以使用 **FPGA**、模拟电路、单片机来产生舵机的控制信号，但 **FPGA** 成本高且电路复杂。对于脉宽调制信号的脉宽变换，常用的一种方法是采用调制信号获取有源滤波后的直流电压，但是需要 50Hz(周期是 20ms) 的信号，这对运放器件的选择有较高要求，从电路体积和功耗考虑也不易采用。5mV 以上的控制电压的变化就会引起舵机的抖动，对于机载的测控系统而言，电源和其他器件的信号噪声都远大于 5mV，所以滤波电路的精度难以达到舵机的控制精度要求。

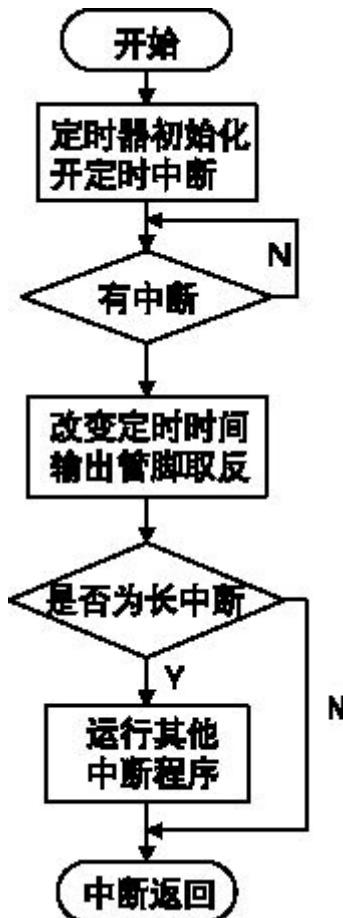
也可以用单片机作为舵机的控制单元，使 **PWM** 信号的脉冲宽度实现微秒级的变化，从而提高舵机的转角精度。单片机完成控制算法，再将计算结果转化为 **PWM** 信号输出到舵机，由于单片机系统是一个数字系统，其控制信号的变化完全依靠硬件计数，所以受外界干扰较小，整个系统工作可靠。

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，本设计是产生 20ms 的周期信号；其次是脉宽的调整，即单片机模拟 **PWM** 信号的输出，并且调整占空比。

当系统中只需要实现一个舵机的控制，采用的控制方式是改变单片机的一个定时器中断的初值，将 **20ms** 分为两次中断执行，一次短定时中断和一次长定时中断。这样既节省了硬件电路，也减少了软件开销，控制系统工作效率和控制精度都很高。

具体的设计过程：例如想让舵机转向左极限的角度，它的正脉冲为 **2ms**，则负脉冲为 **20ms-2ms=18ms**，所以开始时在控制口发送高电平，然后设置定时器在 **2ms** 后发生中断，中断发生后，在中断程序里将控制口改为低电平，并将中断时间改为 **18ms**，再过 **18ms** 进入下一次定时中断，再将控制口改为高电平，并将定时器初值改为 **2ms**，等待下次中断到来，如此往复实现 **PWM** 信号输出到舵机。用修改定时器中断初值的方法巧妙形成了脉冲信号，调整时间段的宽度便可使伺服机灵活运动。

为保证软件在定时中断里采集其他信号，并且使发生 **PWM** 信号的程序不影响中断程序的运行(如果这些程序所占用时间过长，有可能会发生中断程序还未结束，下次中断又到来的后果)，所以需要将采集信号的函数放在长定时中断过程中执行，也就是说每经过两次中断执行一次这些程序，执行的周期还是 **20ms**。软件流程如图 2 所示。



如图 2 产生 PWM 信号的软件流程

如果系统中需要控制几个舵机的准确转动，可以用单片机和计数器进行脉冲计数产生 PWM 信号。

脉冲计数可以利用 51 单片机的内部计数器来实现，但是从软件系统的稳定性和程序结构的合理性看，宜使用外部的计数器，还可以提高 CPU 的工作效率。实验后从精度上考虑，对于 FUTABA 系列的接收机，当采用 1MHz 的外部晶振时，其控制电压幅值的变化为 0.6mV，而且不会出现误差积累，可以满足控制舵机的要求。最后考虑数字系统的离散误差，经估算误差的范围在±0.3% 内，所以采用单片机和 8253、8254 这样的计数器芯片的 PWM 信号产生电路

是可靠的。图 3 是硬件连接图。

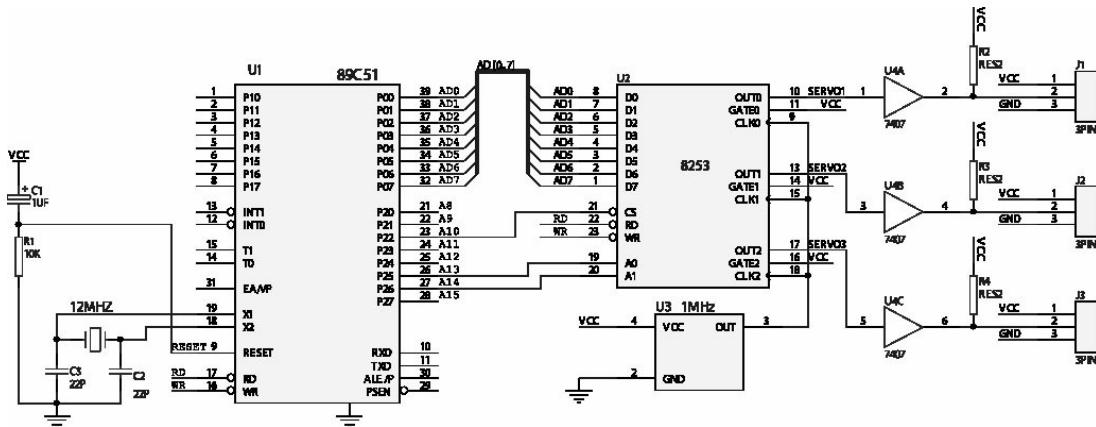


图 3 PWA 信号的计数和输出电路

基于 8253 产生 PWM 信号的程序主要包括三方面内容：一是定义 8253 寄存器的地址，二是控制字的写入，三是数据的写入。软件流程如图 4 所示，具体代码如下。

//关键程序及注释：

//定时器 T0 中断，向 8253 发送控制字和数据

```
void T0Int() interrupt 1
```

```
{
```

```
TH0 = 0xB1;
```

```
TL0 = 0xE0; //20ms 的时钟基准
```

```
//先写入控制字，再写入计数值
```

```
SERVO0 = 0x30; //选择计数器 0，写入控制字
```

```
PWM0 = BUF0L; //先写低，后写高
```

```
PWM0 = BUF0H;
```

```
SERVO1 = 0x70; //选择计数器 1，写入控制字
```

```

PWM1 = BUF1L;
PWM1 = BUF1H;
SERVO2 = 0xB0; //选择计数器 2, 写入控制字
PWM2 = BUF2L;
PWM2 = BUF2H;
}

```

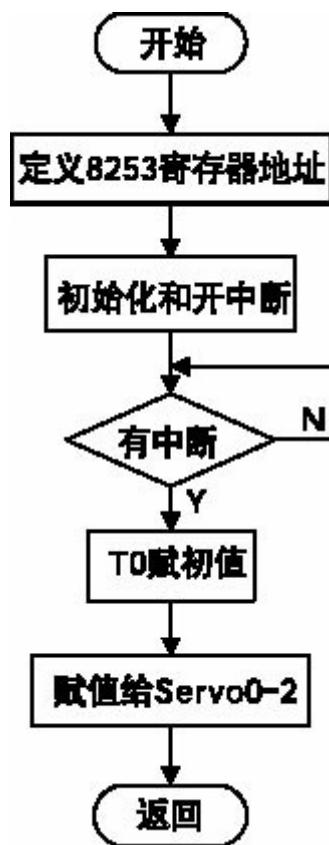


图 4 基于 8253 产生 PWA 信号的软件流程

当系统的主要工作任务就是控制多舵机的工作，并且使用的舵机工作周期均为 20ms 时，要求硬件产生的多路 PWM 波的周期也相同。使用 51 单片机的内部定时器产生脉冲计数，一般工作正脉冲

宽度小于周期的 **1/8**, 这样可以在 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** 设置初值, 并设置输出 **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, 设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, 设置此中断周期为 **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, 最后就可以实现 **8** 路舵机控制信号的输出。

也可以采用外部计数器进行多路舵机的控制, 但是因为常见的 8253、8254 芯片都只有 3 个计数器, 所以当系统需要产生多路 **PWM** 信号时, 使用上述方法可以减少电路, 降低成本, 也可以达到较高的精度。调试时注意到由于程序中脉冲宽度的调整是靠调整定时器的初值, 中断程序也被分成了 **8** 个状态周期, 并且需要严格的周期循环, 而且运行其他中断程序代码的时间需要严格把握。

在实际应用中, 采用 **51** 单片机简单方便地实现了舵机控制需要的 **PWM** 信号。对机器人舵机控制的测试表明, 舵机控制系统工作稳定, **PWM** 占空比 (**0.5~2.5ms** 的正脉冲宽度)和舵机的转角(**-90°~90°**)线性度较好。

天猫店铺网址: <https://zave.tmall.com/>



天猫店铺网址: <https://zave.tmall.com/>