单片机控制舵机

我们知道，舵机和步进电机、直流电机等都是感性负载，单片机的驱动电流较小，我们驱动直流电机、步进电机的时候都使用了驱动模块，也就是功率放大器件。那驱动舵机的时候是否需要呢？因为舵机内部集成了驱动电路，可以对我们输入的 PWM 信号直接采样，所以，控制舵机的时候，用一个单片机的 PWM 输出引脚即可，这大大精简了电路的设计。

1.舵机供电电压和电流

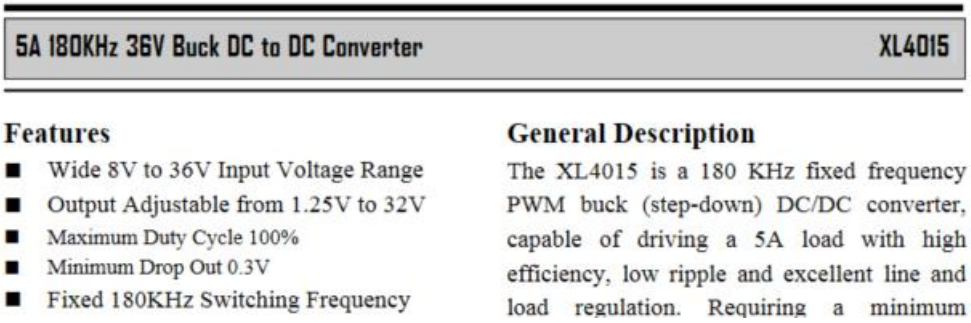
要使舵机工作在额定功率下，电路方面需要满足舵机的要求，包括电流和电压，这个我们可以根据舵机的具体参数选择，比如某款舵机参数如下：

- 扭力：13kg/cm(at 4.8V) 15kg/cm(at 6V)
- 速度：0.18sec/60°(at 4.8V) 0.15sec/60°(at 6V)
- 工作电压：4.8v-6v

根据以上的信息，我们最好能够提供 6V 的电压，我们知道，设备的电流是由负载决定的，比如舵机空载控制的时候一般电流是不大于 400mA，但是带负载的时候可能大于 1A，然后我们设计机械臂的时候有 5 个舵机，因为处于不同的关节，所以实际使用中不会每个舵机都同时达到最大电流，那这里就可以选择 6V 5A 的电源。

要输出这么大的电流，一般的 LDO(线性稳压器)是无法满足的了，需要选择开关稳压芯片，而一般的芯片也没有固定 5V 输出的，需要选择可调的版本，通过电阻调节输出电压至 6V。

这里我们选择 XL4015,根据手册，这款芯片可以满足我们的要求，如图 1。



5A 180KHz 36V Buck DC to DC Converter XL4015

Features

- Wide 8V to 36V Input Voltage Range
- Output Adjustable from 1.25V to 32V
- Maximum Duty Cycle 100%
- Minimum Drop Out 0.3V
- Fixed 180KHz Switching Frequency

General Description

The XL4015 is a 180 KHz fixed frequency PWM buck (step-down) DC/DC converter, capable of driving a 5A load with high efficiency, low ripple and excellent line and load regulation. Requiring a minimum

图 1　XL4015 芯片手册

下面是使用的应用电路图，如图 2。

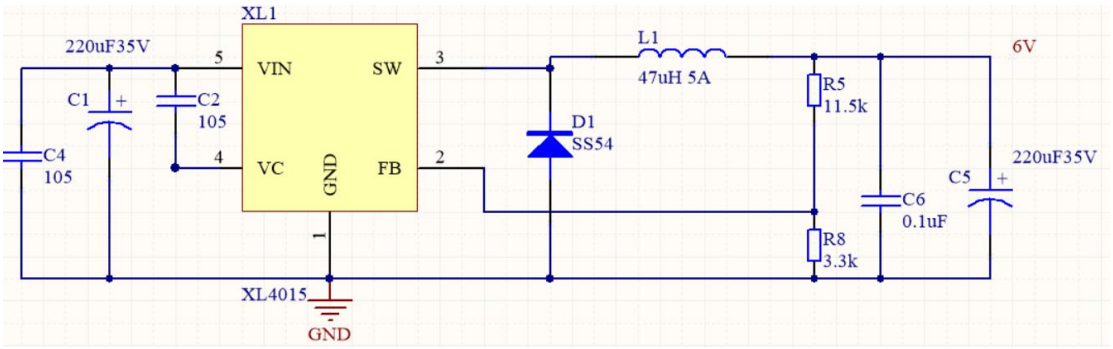


图 2　XL4015 应用电路

2. 舵机的速度控制

舵机的驱动是比较容易的，当我们使用单片机控制的时候，通过输出 50HZ(20ms 的周期)的 PWM，控制 PWM 的脉宽调节舵机的转角。为节约篇幅，冗长的 PWM 初始化的代码就不贴出来了，大家翻看我们的程序即可。前面的章节有说明：舵机的转角和脉宽(高电平长度)存在一一对应关系，如果要控制舵机转到某一角度，就改变输出的脉宽即可，比如从 1ms 到 1.5ms，显然，很容易就实现了舵机的位置控制，但是我们如何进行舵机的速度控制呢？

这里我们引入了位置 PID 算法，下面先看一下程序

```
Velocity1=Position_PID1(Position1,Target1);
Position1+=velocity1;
TIM4->CCR1=Position1;
```

其中我们使用 Velocity1 用于代表舵机的速度，这个值根据目标值和舵机的实际位置计算得到，然后通过累积的方法，赋值给相关的寄存器作用到舵机。这样我们就把舵机的速度调节变成了 PID 参数大小的调节。另外，在接近目标位置的时候还可以实现减速，防止因为惯性的问题造成舵机齿轮减速箱的损坏。下面我们看一下 Position_PID1 这个函数

```
/*函数功能：位置式 PID 控制器
入口参数：位置信息，目标位置
返回　值：控制量
根据位置式离散 PID 公式
pwm=Kp*e(k)+Ki*∑e(k)+Kd[e（k）-e(k-1)]
```

```
e(k)代表本次偏差
e(k-1)代表上一次的偏差
∑e(k)代表 e(k)以及之前的偏差的累积和;其中 k 为 1,2,,k;
pwm 代表输出*/
float Position_PID1 (float Encoder,float Target)
{
     static float Bias,Pwm,Integral_bias,Last_Bias;
     Bias=Target-Encoder; //计算偏差
     Integral_bias+=Bias;//求出偏差的积分
    Pwm=Position_KP*Bias/100+Position_KI*Integral_bias/100+Position_KD*(Bia
    s-Last_Bias)/100; //位置式 PID 控制器
     Last_Bias=Bias;      //保存上一次偏差
     return Pwm;          //控制量输出
}
```

PID 参数是这样设定的，Position_KP=5,Position_KI=0,Position_KD=2;这里没使用 I 控制，因为 PD 控制已经可以满足要求，关于 PID 调试可以结合之前的 PID 位置控制参数整定章节进一步学习了解。