

(SKU:DRI0029)Veyron Servo Driver 24-Channel

来自DFRobot Product Wiki

目录

- 1 简介
- 2 技术参数
- 3 性能描述
- 4 引脚定义
- 5 控制器测试的基本示例
- 6 Arduino与Veyron_Servo_Driver_24-Channel串口通讯实例
- 7 Arduino&XBee与Veyron_Servo_Driver_24-Channel无线通讯实例
- 8 Arduino&APC220与Veyron_Servo_Driver_24-Channel无线通讯实例
- 9 Veyron_Servo_Driver_24-Channel 软件使用说明
 - 9.1 1.快速参考图
 - 9.2 2.软件界面说明
 - 9.3 3.一个简单的例程
- 10 舵机角度与PWM信号的关系
- 11 控制器命令集
- 12 疑难解答
- 13 相关文档

简介



- 在没有认真阅读本说明之前，请勿给模块加电！错误接线将导致模块永久性损坏或烧毁微控制器；
- 请认真查看引脚功能说明，正确接线！请勿将电源反接，否则将导致模块永久性损坏；
- 请勿使用超出额定电压的电源！保证电源的稳定，如果出现高压脉冲，可能会导致微控制器永久性损坏；
- 本产品无防水防潮功能，请在干燥环境下保存或使用！不可将重物堆积在上面；
- **请勿给Maple RET6芯片烧写任何程序**，否则预装的程序将会丢失，无法读取任何指令。

Veyron Servo Driver 24-Channel是专为人形机器人、蜘蛛机器人、机械手等多舵机使用而量身定做的多路舵机控制器。控制器还增加了无线数传接口，兼容DFRobot出品的蓝牙模块和APC220无线数传以及Xbee模块，可实现无线远程控制。Veyron Servo Driver 24-Channel控制方式包括实时、定时、定速控制等。

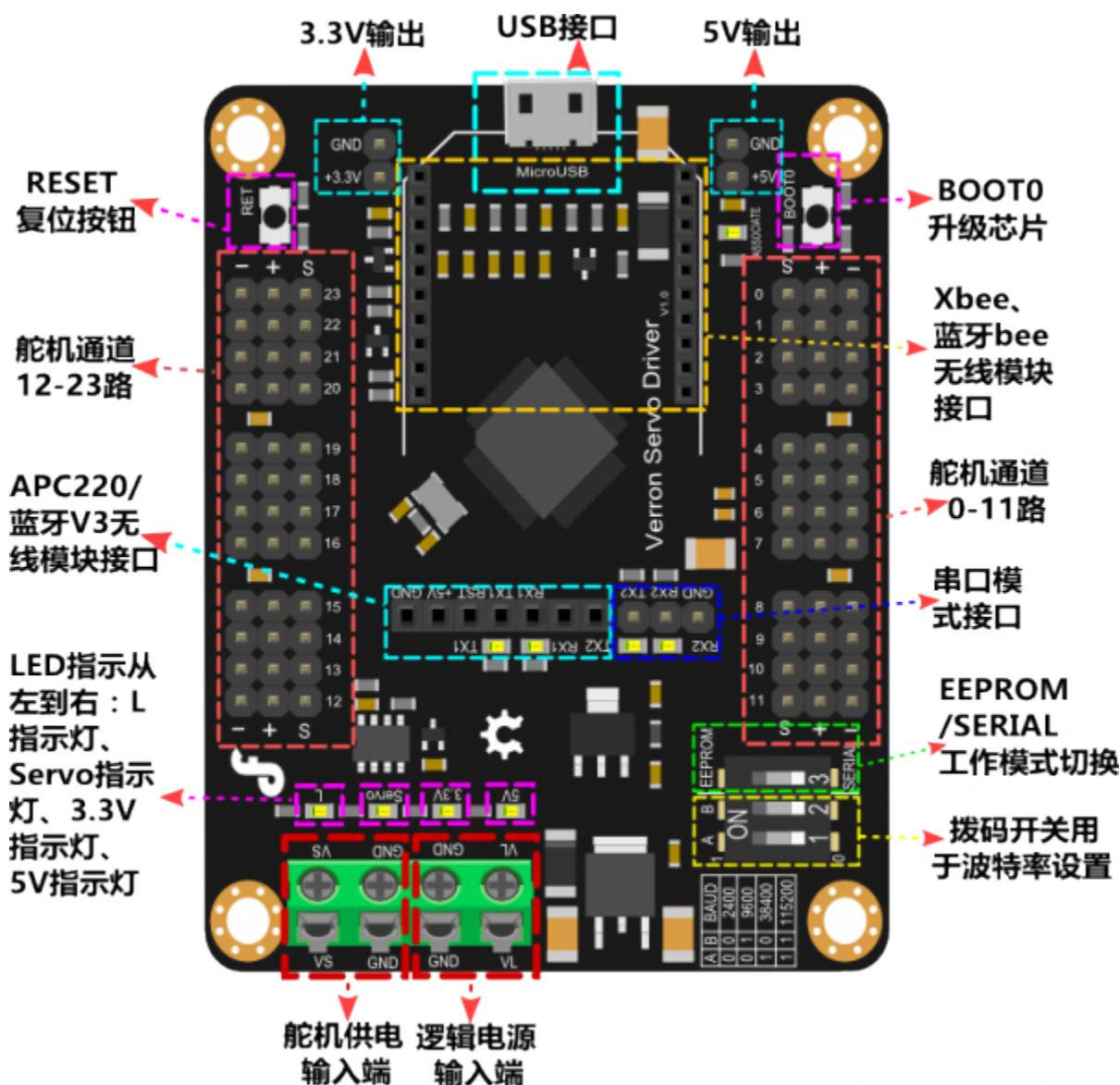
技术参数

- 输出通道：24路（脉冲调制输出或TTL电平输出）；
- 舵机供电：DC4.8V~6V；
- 逻辑供电：DC6V~12V或USB供电（具有自恢复保险丝，调试时使用）；
- 驱动分辨率：1 μ S, 0.09°；
- 驱动速度分辨率：1 μ S/秒，0.09°/秒；
- 通讯接口：MicroUSB/TTL串口接口；
- 串口波特率：2400、9600、38.4k、115.2k可设置；
- 电路板尺寸：57.3 x 72.3mm；
- 重量：26g

性能描述

- 目前市面上功能最强、可靠性最高的MicroUSB版Veyron Servo Driver 24-Channel舵机控制器，核心控制部件采用了高性能、低功耗的STM32F103单片机，具有功能强大、命令执行速度快、控制精度极高、I/O 端口驱动能力强等特点；
- 电路布局紧凑合理，采用贴片式设计使电路板体积大大缩小，同时具有很强的抗干扰性能；
- 目前市面上的电脑都很少提供串口，而较多的是USB接口，所以本产品采用了USB接口，方便没有串口的台式电脑和笔记本使用；
- 使用2位DIP拨码开关设置波特率；
- 提供蓝牙模块、Xbee模块和APC220无线数传接口，可实现远程控制；
- 支持Futaba、Hitec、辉盛以及大部分常见舵机；
- 舵机可控范围：0~180°（360°连续旋转需对舵机进行改装）；
- 舵机模式：单舵机控制，群舵机控制，群控制中，被编入同一群的舵机可以在旋转角度大小不同的时候能做到自动协调速度，同启同止，此功能在设计多自由度仿生机器人时非常实用，可使动作平滑、流畅；
- 控制方式包括：即时、定时、定速控制等。

引脚定义



控制器测试的基本示例

这一个章节注重于Veyron Servo Driver 24-Channel的基本功能。通过这个示例，你可以很方便的搭建你自己的舵机控制应用。

准备材料

- Veyron Servo Driver 24-Channel ×1
- Micro USB 线 ×1
- TowerPro SG90舵机 ×3
- 5V舵机供电电源 ×1
- 串口工具软件 ×1

详细教程

1.初次使用Veyron Servo Driver 24-Channel，都需要为其安装DFU和虚拟串口两个驱动程序。首先按照如下步骤安装DFU驱动程序：

在windos环境下安装：

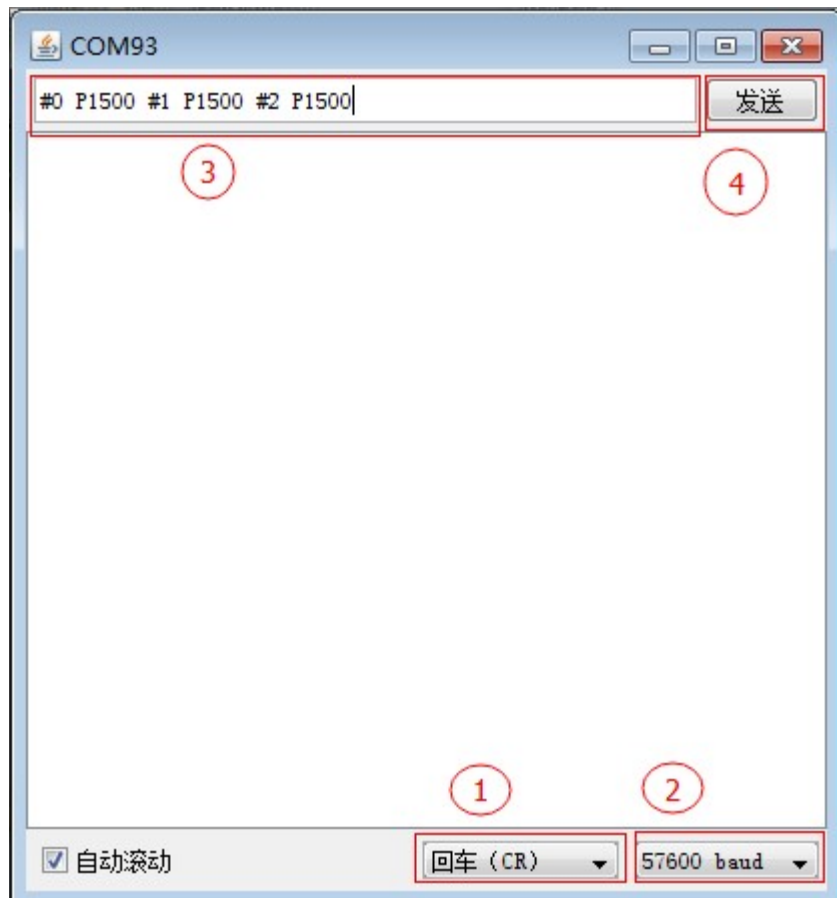
- 1) 将Veyron Servo Driver 24-Channel连接到计算机的USB接口。
- 2) 按下左下角的复位键（印有“RET”的按键），这时蓝色指示灯会快速闪烁6次，然后慢闪烁几次。
- 3) 再次按下复位键，这次在蓝色指示灯快速闪烁6次期间按下另一个键（右上角，印有“BOOT0”）并保持不放，直到指示灯开始慢闪烁时放开。蓝色指示灯会一直闪烁，使你有机会安装DFU驱动程序。
- 4) Windows会提示你需要驱动程序，人工指定驱动程序所在目录位置，选择DFservo文件夹中的mapleDrv\dfu。下一步安装虚拟串口驱动程序：
- 5) 复位Veyron Servo Driver 24-Channel的RET按键，等待蓝色指示灯停止闪烁。此时Windows会提示安装驱动程序，同样，人工指定驱动程序所在目录为DFservo文件夹中的mapleDrv\serial。

在Linux环境下安装步骤如下：

- 1) 首先确认是否已经安装了JRE，如果没有，可以通过如下命令安装：`sudo aptitude install openjdk-6-jre`
- 2) 解压压缩文件到合适的位置（例如桌面）。
- 3) 打开解压压缩文件得到的文件夹，运行其中的install-udev-rules.sh，出现要求输入用于获得管理员权限的密码的提示。
- 4) 用“`sudo restart udev`”命令重启udev即可。

在Mac环境下安装步骤如下：

- 1) 双击你所下载的DMG文件使该镜像被加载即可。
- 2.连接Micro USB电缆到Veyron Servo Driver 24-Channel，即使用USB电源给控制器逻辑部分供电，这是控制器上的逻辑电源指示灯5V/3.3V和舵机控制器指示灯L都会亮。将TowerPro SG90舵机接上0、1、2号口，舵机供电使用外部5V舵机供电电源接到VS与GND上，拨码开关3拨到SERIAL处，USB波特率已默认为57600（不可更改），需要和软件一致。
- 3.安装 Arduino IDE (<http://arduino.cc/en/Main/Software>) . (Arduino 1.0.x 或 Arduino 1.5.x)。我们需要一个串口工具用来发送指令来控制舵机，常用的串口工具很多如Arduino IDE 串口监视器、CoolTerm等，这里我们选择Arduino IDE 串口监视器来演示打开串口助手软件，在菜单-》工具-》串口中选择正确的设备,开启串口监视器 (点击窗口的右上角的按键)。连接成功后，测试就正式开始了，在发送区框中输入命令：



#0 P1500 #1 P1500 #2 P1500<cr> <cr>表示回车

回车后，你会看见L指示灯会闪烁，同时通道0，1，2上的舵机会转动到中间位置。

#0 P750 #1 P1000 #2 P2000 T3000 <cr> <cr>表示回车

回车后，你会看见通道0比通道1和2运动的慢一些，但不管他们的初始位置相差多远，都会同时到达指定位置。

现在来测试查询运动状态，在发送区框中输入命令：

#0 P750 <cr>

回车后，通道0舵机转到接近最小位置，接着输入下面的命令，使通道0舵机花10秒转到接近最大位置。

#0 P2250 T10000 <cr>

舵机正在运动时，输入下面的命令：

Q <cr>

如果舵机正在运动，将返回“+”，如果舵机到达指定位置将返回“。”。

接下来测试舵机的运动速度，在黑色框中输入命令：

```
#0 P750 S1000 <cr>
```

这条命令将使舵机从2250us移动到750us（大概170度），一共耗时1.5秒。

$$\frac{2250us - 750us(\text{运动距离})}{1000us/\text{秒}(\text{速度})} = 1.5\text{秒}$$

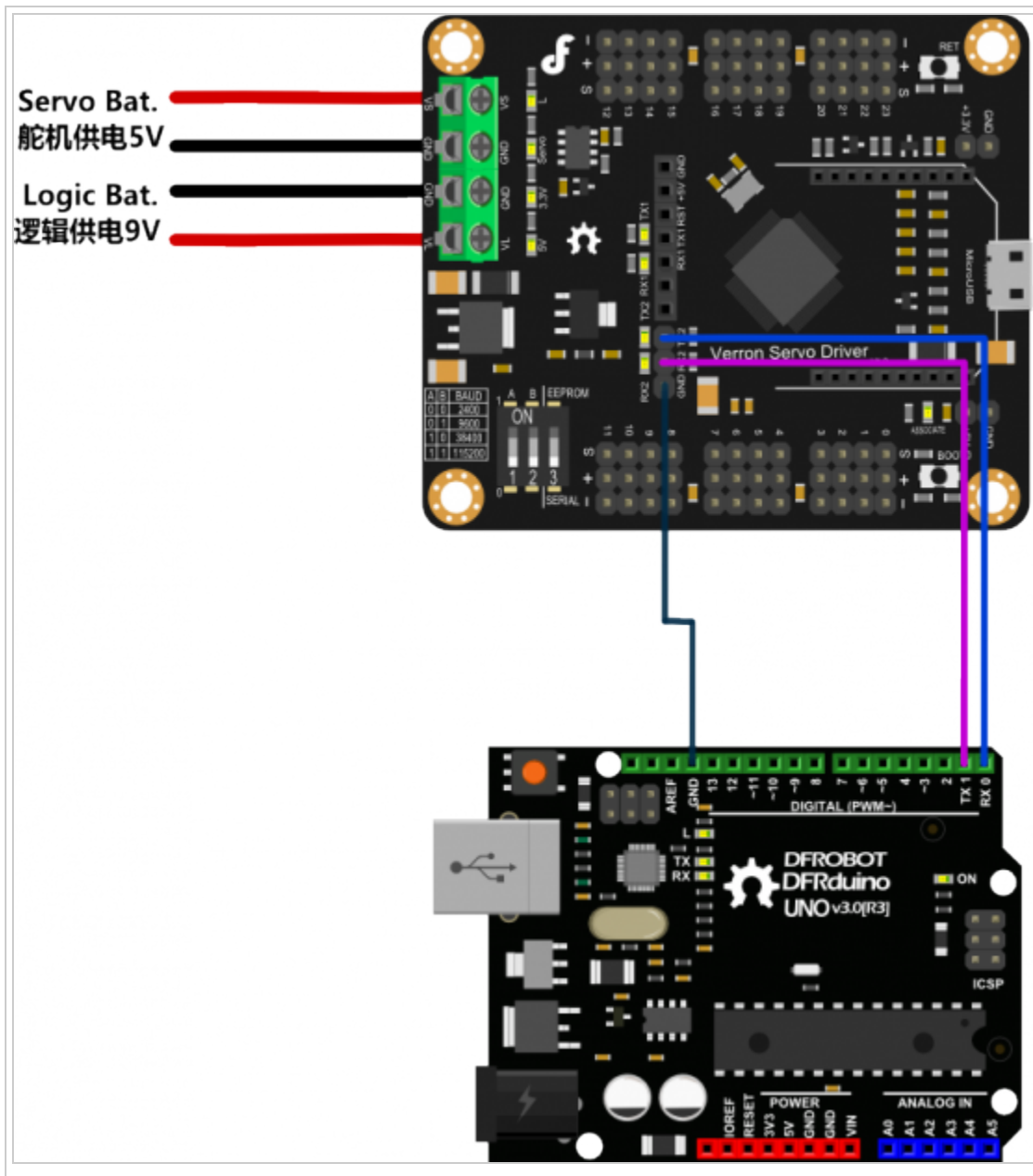
接着输入后面的命令：

```
#0 P2250 S750 <cr>
```

这条命令将使舵机从750us移动到2250us（大概170度），一共耗时2秒。

$$\frac{2250us - 750us(\text{运动距离})}{750us/\text{秒}(\text{速度})} = 2\text{秒}$$

Arduino与Veyron_Servo_Driver_24-Channel串口通讯实例



使用Arduino的串口和Veyron_Servo_Driver_24-Channel的串口通讯，实现舵机动作控制。

准备材料

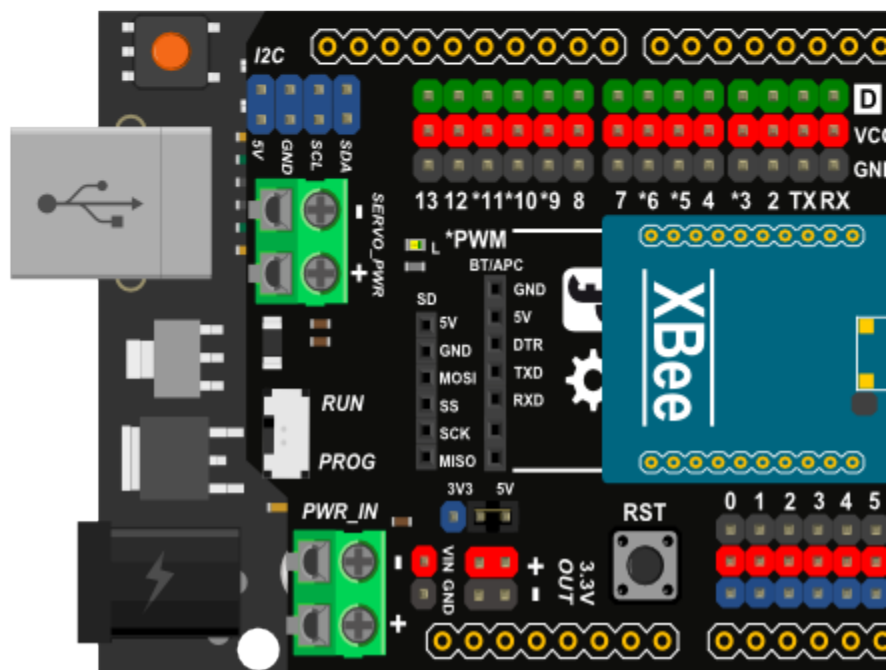
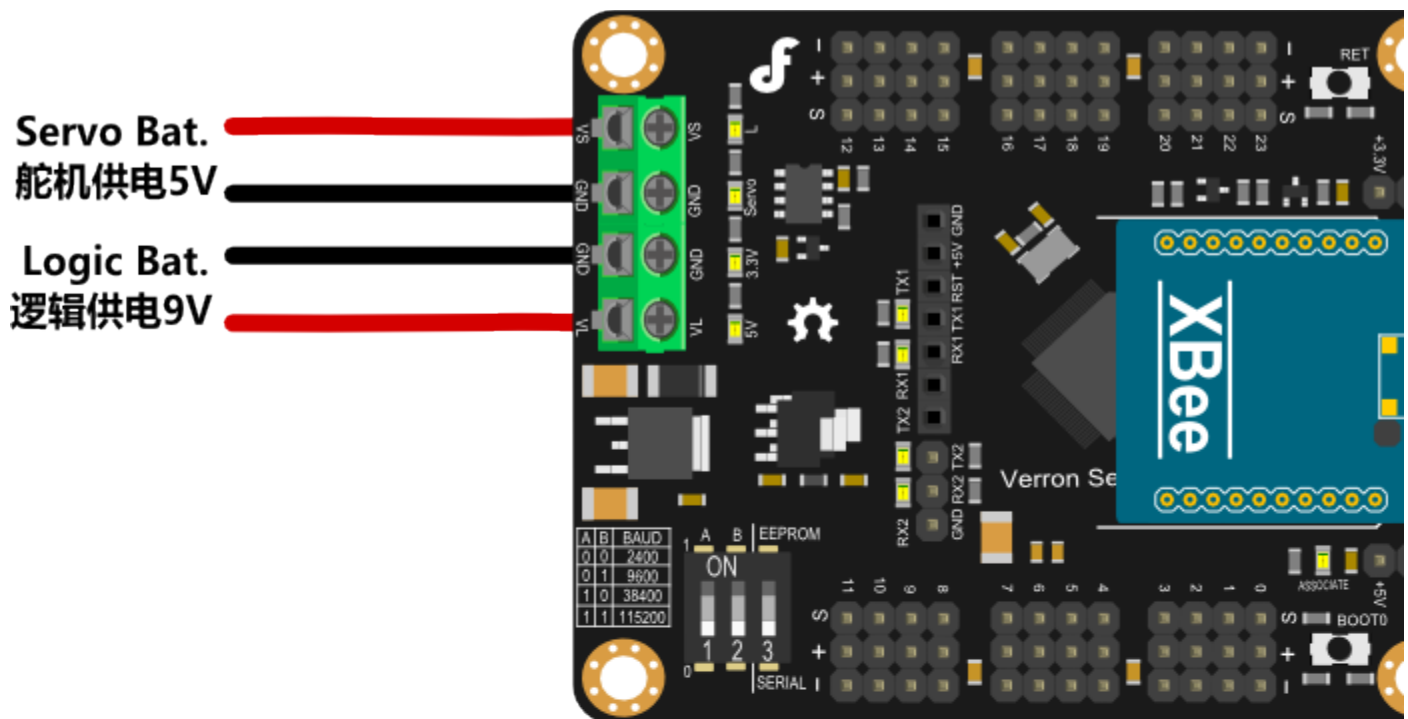
- DFRduino UNO R3 (<http://www.dfrobot.com.cn/goods-521.html>) ×1
- USB电缆 (<http://www.dfrobot.com.cn/goods-396.html>) ×1
- Veyron Servo Driver 24-Channel ×1
- 5V舵机供电电源 ×1
- 9V逻辑供电电源 ×1
- TowerPro SG90舵机 (<http://www.dfrobot.com.cn/goods-214.html>) ×1

测试代码：

```
void setup() {
```

```
Serial.begin(115200);//Set the baudrate to 115200 A:1 B:1
delay(100);//wait for baudrate setting finished
}
void loop() {
  Serial.print("#5 P750");// Channel 5 will move to 750us within 500ms
  delay(5);                //wait for first comand transmission done, if you send
  // a long command, you'd better extend it
  Serial.print("\r");      // send Carriage Return <CR>
  delay(1000);             //wait for servo go to the set position
  Serial.print("#5 P2200");// Channel 5 will move to 2200us within 500ms
  delay(5);
  Serial.print("\r");
  delay(1000);
}
```


Arduino&XBee与Veyron_Servo_Driver_24-Channel无线通讯实例



使用XBee无线模块和Veyron_Servo_Driver_24-Channel的XBee无线模块通讯，实现舵机动作控制。

准备材料

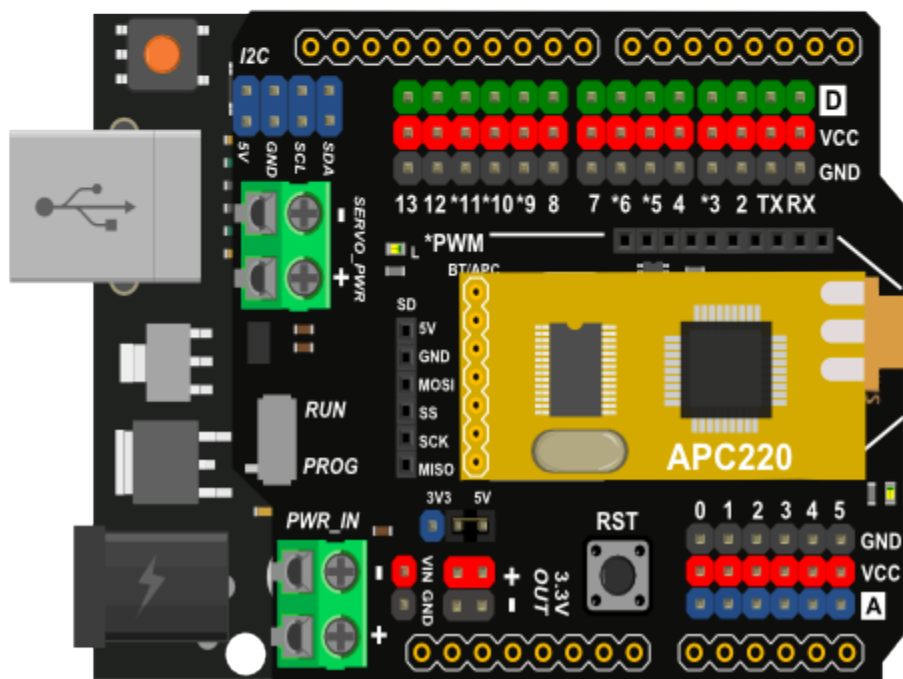
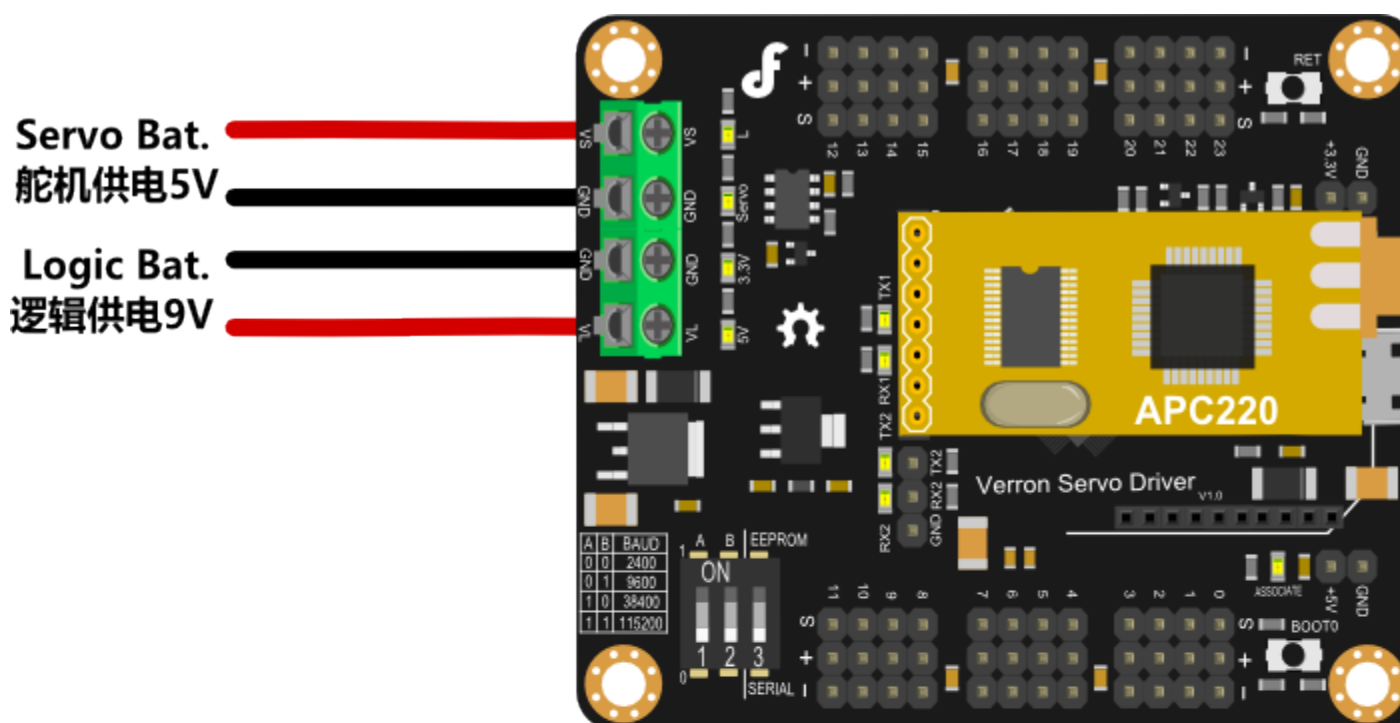
- DFRduino UNO R3 (<http://www.dfrobot.com.cn/goods-521.html>) ×1
- IO 传感器扩展板 V7 (<http://www.dfrobot.com.cn/goods-791.html>) ×1
- XBee 1mW Zigbee 无线数传模块 (<http://www.dfrobot.com.cn/goods-241.html>) ×2
- USB电缆 (<http://www.dfrobot.com.cn/goods-396.html>) ×1
- Veyron Servo Driver 24-Channel ×1

- 5V舵机供电电源 ×1
- 9V逻辑供电电源 ×1
- TowerPro SG90舵机 (<http://www.dfrobot.com.cn/goods-214.html>) ×1

测试代码：

```
void setup() {  
  Serial.begin(115200); //Set the baudrate to 115200 A:1 B:1  
  delay(100); //wait for baudrate setting finished  
}  
void loop() {  
  Serial.print("#5 P750"); // Channel 5 will move to 750us within 500ms  
  delay(5); //wait for first comand transmission done, if you send  
  // a long command, you'd better extend it  
  Serial.print("\r"); // send Carriage Return <CR>  
  delay(1000); //wait for servo go to the set position  
  Serial.print("#5 P2200"); // Channel 5 will move to 2200us within 500ms  
  delay(5);  
  Serial.print("\r");  
  delay(1000);  
}
```

Arduino&APC220与Veyron_Servo_Driver_24-Channel无线通讯实例



使用APC220无线模块和Veyron_Servo_Driver_24-Channel的APC220无线模块通讯，实现舵机动作控制。

准备材料

- DFRduino UNO R3 (<http://www.dfrobot.com.cn/goods-521.html>) ×1
- IO 传感器扩展板 V7 (<http://www.dfrobot.com.cn/goods-791.html>) ×1
- APC220无线数传 (<http://www.dfrobot.com.cn/goods-235.html>) ×1
- USB电缆 (<http://www.dfrobot.com.cn/goods-396.html>) ×1

- Veyron Servo Driver 24-Channel ×1
- 5V舵机供电电源 ×1
- 9V逻辑供电电源 ×1
- TowerPro SG90舵机 (<http://www.dfrobot.com.cn/goods-214.html>) ×1

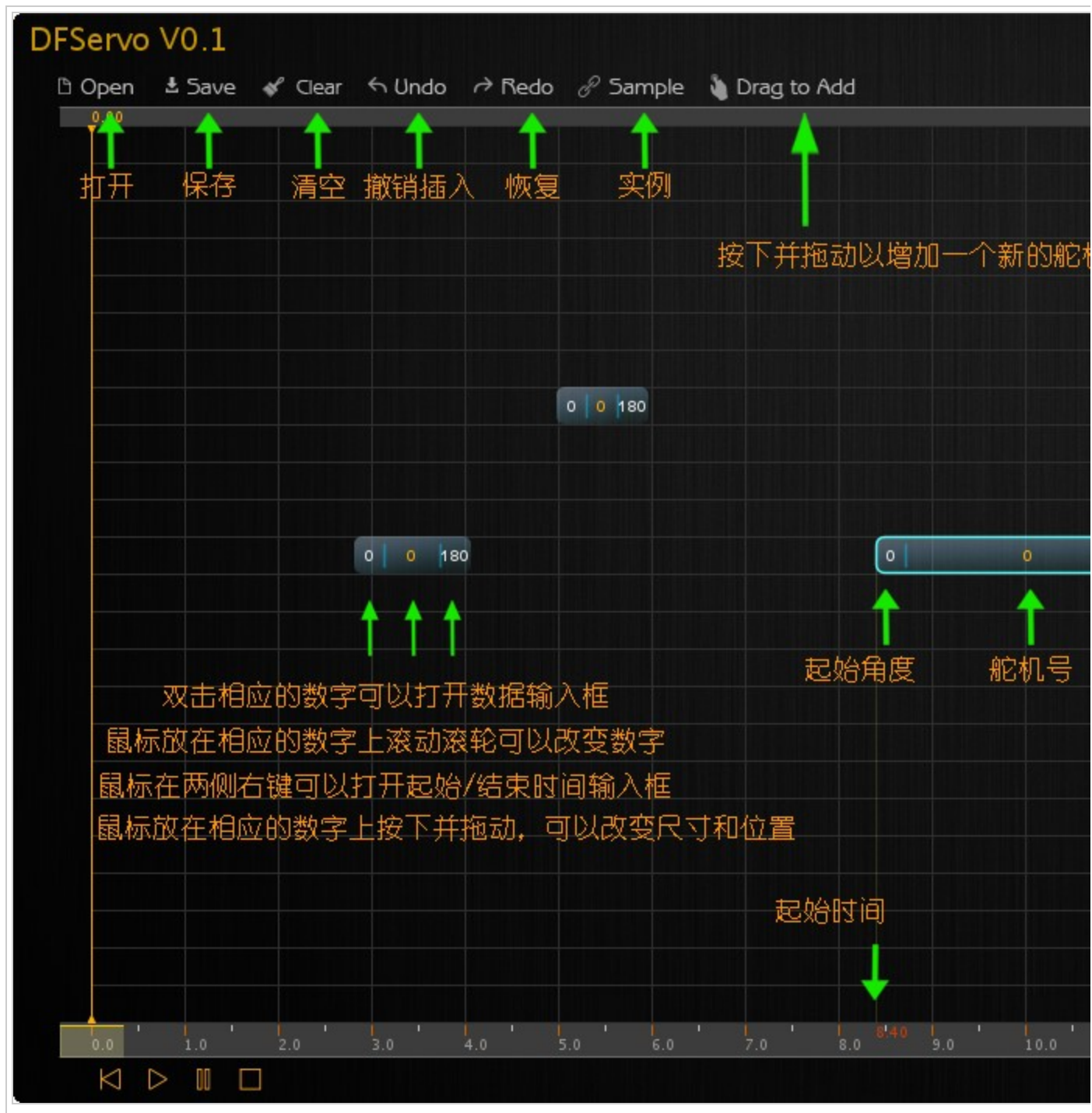
测试代码：

```
void setup() {  
  Serial.begin(115200); //Set the baudrate to 115200 A:1 B:1  
  delay(100); //wait for baudrate setting finished  
}  
void loop() {  
  Serial.print("#5 P750"); // Channel 5 will move to 750us within 500ms  
  delay(5); //wait for first comand transmission done, if you send  
  // a long command, you'd better extend it  
  Serial.print("\r"); // send Carriage Return <CR>  
  delay(1000); //wait for servo go to the set position  
  Serial.print("#5 P2200"); // Channel 5 will move to 2200us within 500ms  
  delay(5);  
  Serial.print("\r");  
  delay(1000);  
}
```

Veyron_Servo_Driver_24-Channel 软件使用说明

1.快速参考图

先发一张图，所谓一图胜千言。。。



2.软件界面说明

Open可以打开一个已保存的文件（该软件是用xml格式存储数据的）

Save可以将制作好的程序保存到电脑

Clear将会清空时间轴，也就是删除时间轴上所有的舵机

Undo从当前时间轴上删除最后添加的舵机

Redo恢复最后删除的舵机到当前时间轴

Sample同时添加24路舵机,每行一个,以方便用户调试，这将会清除时间轴上已添加的舵机

Drag to add添加一个新的舵机，注意，它的操作方法比较特殊，当光标移到它上面时，按住鼠标左键不要放开，然后拖动鼠标到下面的时间轴上，在合适的位置再放开鼠标，一个新的舵机就添加完成了，你可以在接下来的步骤中修改它的参数

Play Mode播放模式选择，有三种：

1.play only once 仅播放一次 2.back and forth 往复播放 3.loop playback 循环播放

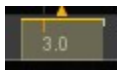
Connect打开串口，就是与舵机控制器连接

COM29串口选择，要选择你的舵机控制器对应的串口，如果没有发现串口，则会显示No Port

57600波特率选择，当前默认值就是57600，一般不需要改动



设置时间轴总时间



在时间轴游标底部的拖动手柄，点击可拖动，从而改变当前的时间



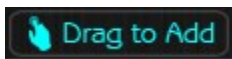
播放控制区，分别是：回到起始位置，播放，暂停，停止

基本操作步骤：

1.首先下载安装程序



2.打开桌面上的程序



3.点击并拖到时间轴区域的合适位置，这就增加了一个新的舵机：

4.鼠标分别指向舵机的左右两边和中间部分按下并拖动，可以分别改变舵机的起始时间和结束时间

5.鼠标分别指向舵机的中间部分按下并拖动，可以改变舵机的位置

6.鼠标分别指向舵机的左右两边并右击，可以精确输入舵机的起始时间和结束时间

7.鼠标分别指向舵机的左右两边并双击，可以分别输入舵机的起始角度和结束角度

8.鼠标指向舵机中间部分右击或双击，可以改变舵机号

9.鼠标在时间轴空白处点击并左右拖动，可以改变时间轴当前视图位置

以上就是基本的操作方法，调试完成后就可以保存到电脑以便下次使用

3.一个简单的例程

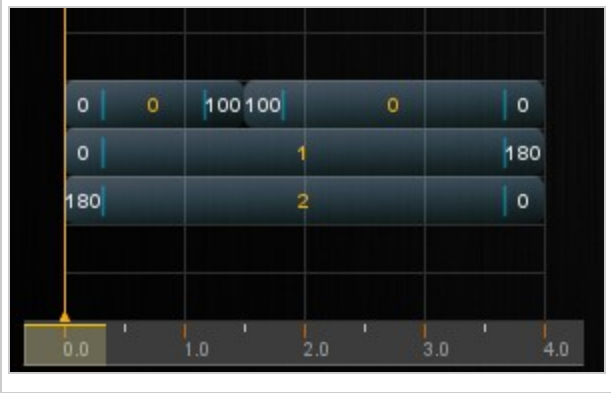
这个例程的解释：

时间轴总时长是4秒；

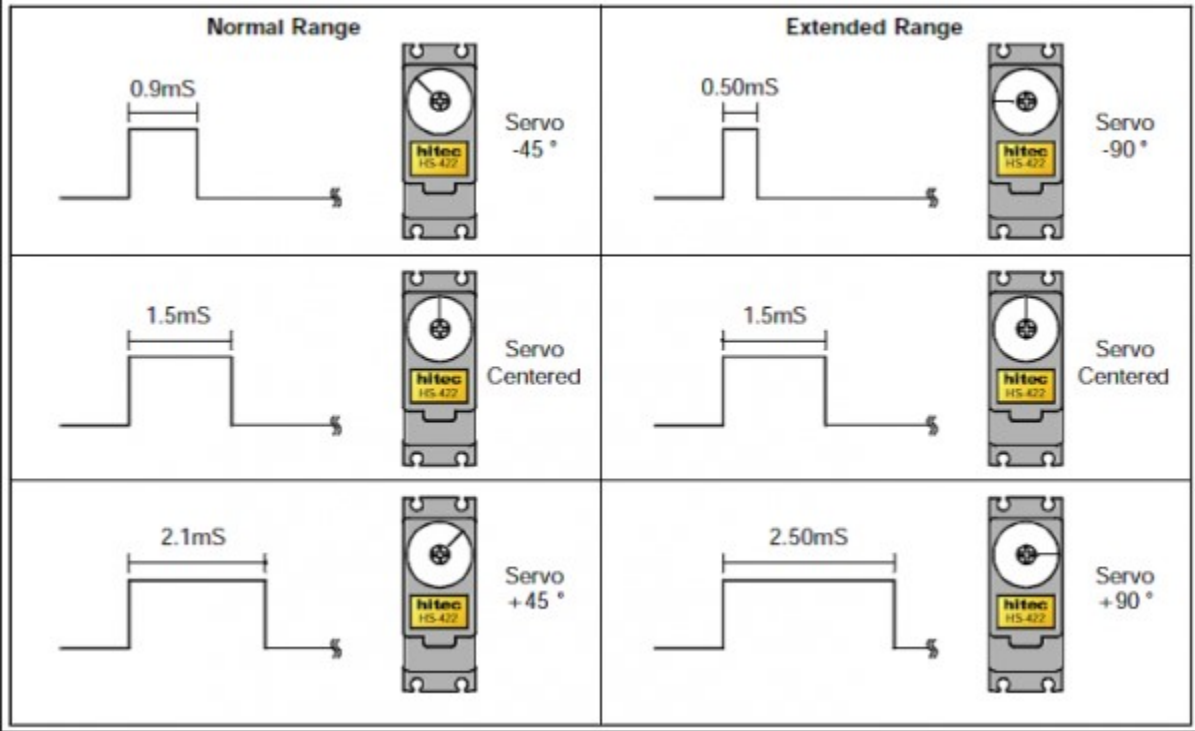
在第0至1.5秒，舵机0 从0度转到100度，在第1.5至4秒，舵机0从100度转到0度；

在第0至4秒，舵机1从0度转到180度；

在第0至4秒，舵机2从180度转到0度。



舵机角度与PWM信号的关系



控制器命令集

1. 单个舵机和舵机群运动命令

```
# <ch> P <pw> S <spd>... # <ch> P <pw> S <spd> T <time><cr>
```

- <ch> : 舵机通道号, 0 – 23。
- <pw> : 脉冲宽度, 单位微秒 (us), 500 – 2500。
- <spd> : 单通道的运动速度, 单位us/秒。(可选)
- <time> : 所有通道的速度, 单位毫秒 (ms), 最大65535。(可选)
- <cr> : 结束回车符, ASCII码中的13。(必选)
- <esc> : 取消当前的命令, ASCII码中的27。

单个舵机运动实例：

```
#5 P1600 S750 <cr>
```


通道5将以750us/秒的速度移动到1600us位置。为了更好的理解速度这个概念，举个例子，如上图所示的舵机与脉冲信号（PWM）的关系，当舵机从-90度到0度时，脉冲宽度为1ms时间即1000us，也就是说1000us脉冲宽度舵机就会转90度，那么100us/秒的速度就表示舵机花10秒的时间就可以转到90度，2000us/秒的速度就表示舵机花0.5秒的时间就可以转到90度。公式：运行时间（秒）=脉冲宽度（us）/速度（us/秒）。

```
#5 P1600 T1000 <cr>
```

通道5将在1秒内从任何位置移动到1600us位置。

```
#5 P1600 #10 P750 T2500 <cr>
```

通道5移动到1600us位置，通道10移动到750us的位置，2个都同时在2500ms内完成，这个命令能协调多个舵机的速度，即使2个舵机的初始位置相差很远，都可以使他们同时开始转动并同时停止到指定位置上。这条命令非常适合人形双足机器人多舵机同时运动，可自动协调所有舵机的速度，完成复杂步态的同步。

你可以使用该命令进行速度和时间组合，组合必须根据下面的规则：

- A．所有通道的开始和结束将同时完成。
- B．如果某个通道指定了速度，那么它将不会快于指定速度（可以根据需要调节移动速度）。
- C．如果某个通道指定了时间，那么它将在指定的时间移动到指定位置（可以根据需要调节移动时间）。

```
#5 P1600 #17 P750 S500 #2 P2250 T2000 <cr>
```

通道5移动到1600us位置，通道17移动到750us的位置，通道2机移动到2250us的位置，整个动作需要2000us，但是通道17的舵机不会按500us/秒的速度运行，这个需要取决于通道17的初始位置。假设通道17的初始位置在2000us，它被指定移动1250us，超过500us/秒的限制，那么他将至少花2500ms完成动作，再假设通道17初始位置在1000us，只需要它移动250us，那么在500us/秒以内，那么他将花2000ms完成动作。

注意：第一条定位命令不能包含速度和时间，格式为“# <ch> P <pw>”，因为在初始状态下，控制器不知道舵机当前位置在什么地方，所以必须这样做。

2．脉冲偏移：

```
# <ch>P0 <offset value> ... # <ch> P0 <offset value> <cr>
```

<ch>：舵机通道号，0－23。

<offset value>：100 到 -100us。

<cr>：结束回车符，ASCII码中的13。

对某个通道的脉冲宽度进行校正，这样人形机器人就可以不用通过机械硬件来进行位置校正。

3．数字输出：


```
# <ch><lvl> ... # <ch> <lvl><cr>
```

<ch> : 舵机通道号, 0 – 23。

<lvl> : 通道输出逻辑电平, 高‘H’或低‘L’。

<cr> : 结束回车符, ASCII码中的13。

该通道将在接受到回车指令20ms内输出电平。

数字输出实例：

```
#3H #4L <cr>
```

该命令使通道3输出高电平（+3.3V），通道4输出低电平（0V）。

4. 字节输出：

```
# <bank> : <value><cr>
```

<bank> : 0 = 通道0-7, 1 = 通道8-15, 2 = 通道16-23。

<value> : 十进制输出 (0-255), Bit0=LSB。

该命令允许一次性写入8位二进制, 并将同时更新bank里的所有通道, 更新将在接受到回车符号后20ms内完成

字节输出实例：

```
#1:123 <cr>
```

该命令使bank 1输出十进制123, 123 (十进制) = 01111011 (二进制), bank 1为通道8-15, 那么bank 1中通道8和13为0, 其他通道为1。

5. 查询运动状态：

```
Q <cr>
```

如果舵机正在转动, 返回值为“+”, 如果移动到指定位置, 返回“.”。

这条命令的返回值可能延迟50us至5ms。

6. 查询脉冲宽度：

```
QP <ch><cr>
```

<ch> : 舵机通道号, 0 – 23。

返回值为一个字节 (二进制), 表示舵机当前的脉冲宽度, 分辨率10us, 比如脉冲宽度是1500us, 那么返回为150 (二进制)。

该命令可查询多个舵机的脉冲宽度，每个舵机一个字节，返回值将延迟50us至5ms，典型为100us。

疑难解答

问：在用arduino控制舵机的实验中，我按照图示的接线，外部电源供电也没问题，但舵机一直不转，请问怎么解决？

答：您好！是由于波特率设置的问题，请检查板子上的波特率设置开关以及控制方式的选择开关。如果您代码中用的是115200，请将开关置于：A1 B1的位置，另外一个开关置于SERIAL一边。另外，开关存在接触不良的情况，请将开关先拨到0，0的位置，再稍微用点力拨到另一端。即可解决！

相关文档

- Veyron Windows驱动和DFServo软件下载 (<https://github.com/DFRobot/Visual-Servo-Controller>)

➔ 购买Veyron Servo Driver 24-Channel (SKU:DRI0029) (<http://www.dfrobot.com.cn/goods-976.html>)

来自“[http://wiki.dfrobot.com.cn/index.php?title=\(SKU:DRI0029\)Veyron_Servo_Driver_24-Channel&oldid=96098](http://wiki.dfrobot.com.cn/index.php?title=(SKU:DRI0029)Veyron_Servo_Driver_24-Channel&oldid=96098)”

-
- 本页面最后修改于2016年8月25日 (星期四) 14:39。
 - 此页面已被浏览过14,527次。