

# API 结构

我们将自始至终使用以下格式的命令对领航者进行控制

```
{'o': 0, 'v': 0, 'c': 0, 'd': 0, 'r': 0, 'a': 0}
```

指令以 JSON 格式构成，一共有 o,v,c,d,r,a 这 6 个键和与之对应的 6 个值数据构成。下表记录了对于这几个键的详细描述。

JSON 键	范围	一般值	说明
o	[0,正无穷]	0  1	大于 1 表示激活车辆并初始化陀螺仪数据。小车电机转动运行过程中值必须为 0
v	[0,150]	[0,100]	小车电机运行的相对速率，可用于控制车速
c	[0,正无穷]	[0,359]	小车在平面直角坐标系中平移运动的方向。
d	[0,正无穷]	0  1	小车自转，转弯方向，大于 0 时表示顺时针旋转，等于 0 表示逆时针旋转
r	[0,正无穷]	[0,正无穷]	小车自转速率（r 毫秒每转）
a	[0,正无穷]	[0,正无穷]	小车自转固定度数

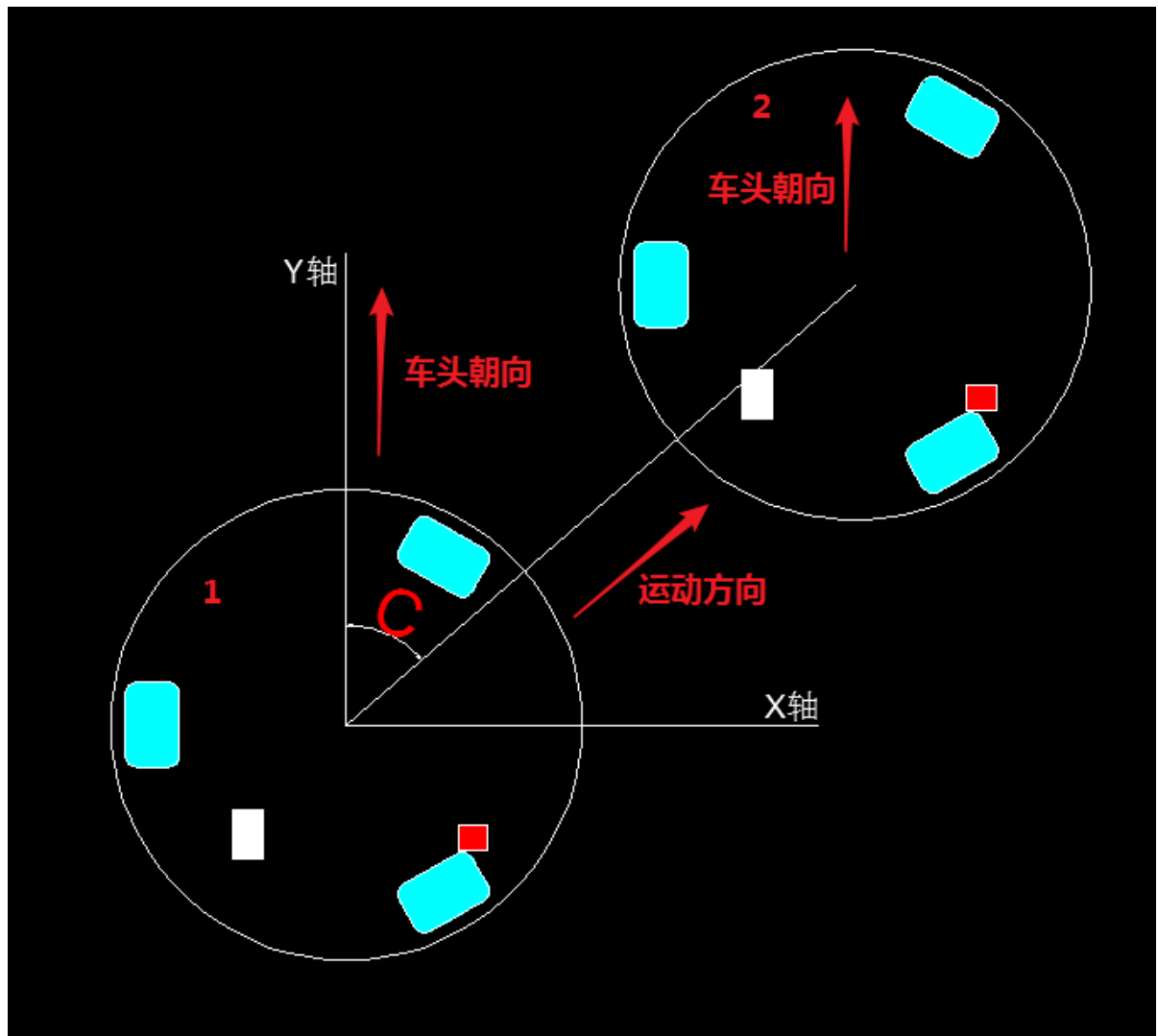
我们可以用以下两个命令来控制车，体验一下实际的作用。

```
#速率 20，向正前方向行驶
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 20, 'c': 0, 'd': 0, 'r': 0, 'a': 0})

#速率 50，向正前方向行驶
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 50, 'c': 0, 'd': 0, 'r': 0, 'a': 0})
```

## 横行霸道的特殊 API

不同于普通小车，领航者可以支持横向运动，就像螃蟹爬那样，利用设置键 `c` 的值就可以控制小车平移运动的角度。所谓平移运动就是保持车头的方向不变，车体向任意一个方向发生平移。如下图所示，领航者从 1 移动到 2 的过程就属于平移。领航者的运动方向是与 Y 轴在顺时针方向的夹角。



可以用以下几个指令来控制领航者，体验一下 `c` 的实际用途。

*#速率 20，向 90 度方向行驶*

```
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 20, 'c': 90, 'd': 0, 'r': 0, 'a': 0})
```

*#速率 20，向 180 度方向行驶（后退）*

```
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 20, 'c': 180, 'd': 0, 'r': 0, 'a': 0})
```

## 定速旋转

领航者能以固定的速度进行自转，这里用到了  $d$  和  $r$  两个参数，其中  $d$  表示自转的方向， $d$  的值为 0 表示小车将逆时针旋转， $d$  的值不为 0 时表示小车将顺时针旋转。 $r$  的值代表旋转一圈所需的毫秒数，这个值越小理论上转得越快，但是由于小车所在的物理世界局限性，使得这个值不能低于 1200，也就是这个车最快是 1.2 秒自转一圈。

可以用以下几个指令来控制领航者，体验一下  $d, r$  的实际用途。

```
#以 10 秒钟一圈的速率朝逆时针方向一直旋转
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 0, 'c': 0, 'd': 0, 'r': 10000, 'a': 0})
#以 10 秒钟一圈的速率朝顺时针方向一直旋转
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 0, 'c': 0, 'd': 1, 'r': 10000, 'a': 0})
#以 3 秒钟一圈的速率朝顺时针方向一直旋转
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 0, 'c': 0, 'd': 1, 'r': 3000, 'a': 0})
```

## 旋转固定角度

很多时候我们希望领航者能只旋转一个恰到好处的固定角度,这时就需要用到  $a$ ， $a$  表示自转的角度，取值范围从 1 到正无穷都可以取，需要配合定速旋转中的  $r$  和  $d$  参数一起使用。

可以用以下几个指令来控制领航者，体验一下  $a$  的实际用途。

```
#以 3 秒钟一圈的速率朝顺时针方向 旋转 90 度停下
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 0, 'c': 0, 'd': 1, 'r': 3000, 'a': 90})
#以 3 秒钟一圈的速率朝顺时针方向 旋转 720 度停下
bwtricar.v2.sendCommandDirectly({'o': 0, 'v': 0, 'c': 0, 'd': 1, 'r': 3000, 'a': 720})
```

## 更多简单接口示例

# 顺时针方向以每 10000 毫秒每转的速率旋转 45 度

```
commandJson = {"o":0,"v":0,"c":0,"d":1,"r":10000,"a":45}
```

# 逆时针方向以每 10000 毫秒每转的速率旋转 720 度

```
commandJson = {"o":0,"v":0,"c":0,"d":0,"r":10000,"a":720}
```

# 以 100 的速率在平面直角坐标系中向 45 度方向保持车头朝向不变平移前进

```
commandJson = {"o":0,"v":10,"c":45,"d":0,"r":0,"a":0}
```

# 逆时针方向以每 10000 毫秒每转的速率持续旋转

```
commandJson = {"o":0,"v":0,"c":0,"d":0,"r":10000,"a":0}
```

# 顺时针方向以每 10000 毫秒每转的速率持续旋转

```
commandJson = {"o":0,"v":0,"c":0,"d":1,"r":10000,"a":0}
```

# 顺时针方向以每 10000 毫秒每转的速率持续旋转与以 50 的速率在平面直角坐标系中向 0 度方向保持车#头朝向不变平移前进的叠加态（一般用于车辆右转弯）

```
commandJson = {"o":0,"v":50,"c":0,"d":1,"r":10000,"a":0}
```

# 逆时针方向以每 10000 毫秒每转的速率持续旋转与以 50 的速率在平面直角坐标系中向 0 度方向保持车#头朝向不变平移前进的叠加态（一般用于车辆左转弯）

```
commandJson = {"o":0,"v":50,"c":0,"d":0,"r":10000,"a":0}
```