

## 4. 机器人常用设置

### 4.1. 设置全局速度

原型	<code>SetSpeed(vel)</code>
描述	设置全局速度
必选参数	<ul style="list-style-type: none"><li><code>vel</code>: 速度百分比, 范围[0~100]</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

#### 4.1.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 error = robot.SetSpeed(20)
5 print("设置全局速度错误码:", error)
```

### 4.2. 设置系统变量值

原型	<code>SetSysVarValue(id,value)</code>
描述	设置系统变量
必选参数	<ul style="list-style-type: none"><li><code>id</code>: 变量编号, 范围[1~20];</li><li><code>value</code>: 变量值</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

#### 4.2.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 for i in range(1,21):
5     error = robot.SetSysVarValue(i,10)
6 robot.WaitMs(1000)
7 for i in range(1,21):
8     sys_var = robot.GetSysVarValue(i)
9     print("系统变量编号:", i, "值", sys_var)
```

### 4.3. 设置工具参考点-六点法

原型	SetToolPoint(point_num)
描述	设置工具参考点-六点法
必选参数	point_num：点编号,范围[1~6]
默认参数	无
返回值	错误码 成功-0 失败- errcode

#### 4.3.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接，连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  t_coord = [1.0,2.0,3.0,4.0,5.0,6.0]
6  for i in range(1,7):
7      robot.DragTeachSwitch(1)#切入拖动示教模式
8      time.sleep(5)
9      error = robot.SetToolPoint(i) #实际应当控制机器人按照要求移动到合适位置后再发送指令
10     print("六点法设置工具坐标系，记录点",i,"错误码",error)
11     robot.DragTeachSwitch(0)
12     time.sleep(1)
13     error = robot.ComputeTool()
14     print("六点法设置工具坐标系错误码",error)
```

### 4.4. 计算工具坐标系-六点法

原型	ComputeTool()
描述	计算工具坐标系-六点法（设置完六个工具参考点后再进行计算）
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none"><li>错误码 成功-0 失败- errcode</li><li>返回值（调用成功返回）tcp_pose [x,y,z,rx,ry,rz] 工具坐标系</li></ul>

### 4.5. 设置工具参考点-四点法

原型	SetTcp4RefPoint(point_num)
描述	设置工具参考点-四点法
必选参数	point_num：点编号,范围[1~4]
默认参数	无
返回值	<ul style="list-style-type: none"><li>错误码 成功-0 失败- errcode</li><li>返回值（调用成功返回）tcp_pose [x,y,z,rx,ry,rz] 工具坐标系</li></ul>

4.5.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接，连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  t_coord = [1.0,2.0,3.0,4.0,5.0,6.0]
6  for i in range(1,5):
7      robot.DragTeachSwitch(1)#切入拖动示教模式
8      time.sleep(5)
9      error = robot.SetTcp4RefPoint(i) #应当控制机器人按照要求移动到合适位置后再发送指令
10     print("四点法设置工具坐标系，记录点",i,"错误码",error)
11     robot.DragTeachSwitch(0)
12     time.sleep(1)
13 error,t_coord= robot.ComputeTcp4()
14 print("四点法设置工具坐标系错误码",error,"工具TCP",t_coord)
```

4.6. 计算工具坐标系-四点法

原型	ComputeTcp4()
描述	计算工具坐标系-四点法（设置完四个工具参考点后再进行计算）
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none"><li>错误码 成功-0 失败- errcode</li><li>返回值（调用成功返回） tcp_pose [x,y,z,rx,ry,rz] 工具坐标系</li></ul>

4.7. 设置工具坐标系

原型	SetToolCoord(id,t_coord,type,install)
描述	设置工具坐标系
必选参数	<ul style="list-style-type: none"><li>id :坐标系编号，范围[0~14]；</li><li>t_coord :工具中心点相对末端法兰中心位姿，单位[mm][°]；</li><li>type :0-工具坐标系，1-传感器坐标系；</li><li>install :安装位置，0-机器人末端，1-机器人外部</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

4.7.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接，连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  t_coord = [1.0,2.0,3.0,4.0,5.0,6.0]
6  error = robot.SetToolCoord(10,t_coord,0,0)
7  print("设置工具坐标系错误码",error)
```

## 4.8. 设置工具坐标系列表

原型	<code>SetToolList(id,t_coord ,type,install)</code>
描述	设置工具坐标系列表
必选参数	<ul style="list-style-type: none"><li><code>id</code>:坐标系编号, 范围[0~14];</li><li><code>t_coord</code>: [x,y,z,rx,ry,rz] 工具中心点相对末端法兰中心位姿, 单位[mm][°];</li><li><code>type</code>: 0-工具坐标系, 1-传感器坐标系;</li><li><code>install</code>: 安装位置, 0-机器人末端, 1-机器人外部</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

### 4.8.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 t_coord = [1.0,2.0,3.0,4.0,5.0,6.0]
6 error = robot.SetToolList(10,t_coord,0,0)
7 print("设置工具坐标系列表错误码",error)
```

## 4.9. 设置外部工具参考点-三点法

原型	<code>SetExTCPPoint(point_num)</code>
描述	设置外部工具参考点-三点法
必选参数	<code>point_num</code> : 点编号,范围[1~3]
默认参数	无
返回值	错误码 成功-0 失败- errcode

### 4.9.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 etcp = [1.0,2.0,3.0,4.0,5.0,6.0]
6 etool = [21.0,22.0,23.0,24.0,25.0,26.0]
7 for i in range(1,4):
8     error = robot.SetExTCPPoint(i) #应当控制机器人按照要求移动到合适位置后再发送指令
9     print("三点法设置外部工具坐标系, 记录点",i,"错误码",error)
10    time.sleep(1)
11 error,etcp = robot.ComputeExTCF()
12 print("三点法设置外部工具坐标系错误码",error,"外部工具TCP",etcp)
13 error = robot.SetExToolCoord(10,etcp,etool)
14 print("设置外部工具坐标系错误码",error)
15 error = robot.SetExToolList(10,etcp,etool)
16 print("设置外部工具坐标系列表错误码",error)
```

## 4.10. 计算外部工具坐标系-三点法

原型	<code>ComputeExTCF (point_num)</code>
描述	计算外部工具坐标系-三点法（设置完三个参考点后再进行计算）
必选参数	<code>point_num</code> ：点编号,范围[1~3]
默认参数	无
返回值	<ul style="list-style-type: none"><li>错误码 成功-0 失败- errcode</li><li>etcp [x,y,z,rx,ry,rz] 外部工具坐标系</li></ul>

## 4.11. 设置外部工具坐标系

原型	<code>SetExToolCoord(id,etcp ,etool)</code>
描述	设置外部工具坐标系
必选参数	<ul style="list-style-type: none"><li><code>id</code>:坐标系编号，范围[0~14];</li><li><code>etcp</code>:外部工具坐标系，单位[mm][°];</li><li><code>etool</code>:末端工具坐标系，单位[mm][°];</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

### 4.11.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接，连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  etcp = [1.0,2.0,3.0,4.0,5.0,6.0]
6  etool = [21.0,22.0,23.0,24.0,25.0,26.0]
7  error = robot.SetExToolCoord(10,etcp,etool)
8  print("设置外部工具坐标系错误码",error)
```

## 4.12. 设置外部工具坐标列表

原型	<code>SetExToolList(id,etcp ,etool)</code>
描述	设置外部工具坐标列表
必选参数	<ul style="list-style-type: none"><li><code>id</code>:坐标系编号，范围[0~14];</li><li><code>etcp</code>:外部工具坐标系，单位[mm][°];</li><li><code>etool</code>:末端工具坐标系，单位[mm][°];</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

4.12.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接，连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 etcp = [1.0,2.0,3.0,4.0,5.0,6.0]
6 etool = [21.0,22.0,23.0,24.0,25.0,26.0]
7 error = robot.SetExToolList(10,etcp,etool)
8 print("设置外部工具坐标列表错误码",error)
```

4.13. 设置工件参考点-三点法

原型	SetWObjCoordPoint(point_num)
描述	设置工件参考点-三点法
必选参数	point_num :点编号,范围[1~3]
默认参数	无
返回值	错误码 成功-0 失败- errcode

4.13.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接，连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 w_coord = [11.0,12.0,13.0,14.0,15.0,16.0]
6 robot.SetToolList(0,[0,0,0,0,0,0],0,0)#设置参考点前应当将工具和工件号坐标系切换至0
7 robot.SetWObjList(0,[0,0,0,0,0,0])
8 for i in range(1,4):
9     error = robot.SetWObjCoordPoint(i) #实际应当控制机器人按照要求移动到合适位置后再发送指令
10    print("三点法设置工件坐标系，记录点",i,"错误码",error)
11    time.sleep(1)
12 error, w_coord = robot.ComputeWObjCoord(0)
13 print("三点法计算工件坐标系错误码",error,"工件坐标系", w_coord)
```

4.14. 计算工件坐标系-三点法

原型	ComputeWObjCoord()
描述	计算工件坐标系-三点法（三个参考点设置完后再进行计算;
必选参数	method 计算方式 :0: 原点-x轴-z轴 1: 原点-x轴-xy平面
默认参数	无
返回值	<ul style="list-style-type: none"><li>错误码 成功-0 失败- errcode</li><li>返回值（调用成功返回） wobj_pose [x,y,z,rx,ry,rz] 工件坐标系</li></ul>

4.15. 设置工件坐标系

原型	SetWObjCoord(id,w_coord)
----	--------------------------

描述	设置工件坐标系
必选参数	<ul style="list-style-type: none"><li><code>id</code>:坐标系编号, 范围[0~14];</li><li><code>w_coord</code>:坐标系相对位姿, 单位[mm][°];</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

### 4.15.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 w_coord = [11.0,12.0,13.0,14.0,15.0,16.0]
6 error = robot.SetWObjCoord(11,w_coord)
7 print("设置工件坐标系错误码",error)
```

## 4.16. 设置工件坐标系列表

原型	<code>SetWObjList(id,w_coord)</code>
描述	设置工件坐标系列表
必选参数	<ul style="list-style-type: none"><li><code>id</code>:坐标系编号, 范围[0~14];</li><li><code>w_coord</code>:坐标系相对位姿, 单位[mm][°];</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

### 4.16.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 w_coord = [11.0,12.0,13.0,14.0,15.0,16.0]
6 error = robot.SetWObjList(11,w_coord)
7 print("设置工件坐标系列表错误码",error)
```

## 4.17. 设置末端负载重量

原型	<code>SetLoadWeight(weight)</code>
描述	设置末端负载重量,错误负载重量设置可能会导致拖动模式下机器人失控
必选参数	<ul style="list-style-type: none"><li><code>weight</code>:单位[kg]</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

4.17.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接，连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 error = robot.SetLoadWeight(0)#! !! 负载重量设置应于实际相符(错误负载重量设置可能会导致拖动模式下机器人失控)
```

4.18. 设置机器人安装方式-固定安装

原型	SetRobotInstallPos(method)
描述	设置机器人安装方式-固定安装,错误安装方式设置会导致拖动模式下机器人失控
必选参数	<ul style="list-style-type: none"><li>method :0-平装，1-侧装，2-挂装</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

4.18.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接，连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 error = robot.SetRobotInstallPos(0) #! !! 安装方式设置应与实际一致 0-正装，1-侧装，2-倒装（错误安装方式设置会导致拖动模式下机器人失控）
5 print("设置机器人安装方式错误码",error)
```

4.19. 设置机器人安装角度-自由安装

原型	SetRobotInstallAngle(yangle,zangle)
描述	设置机器人安装角度-自由安装,错误安装角度设置会导致拖动模式下机器人失控
必选参数	<ul style="list-style-type: none"><li>yangle : 倾斜角</li><li>zangle : 旋转角</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

4.19.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接，连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 error = robot.SetRobotInstallAngle(0.0,0.0) #! !! 安装角度设置应与实际一致（错误安装角度设置会导致拖动模式下机器人失控）
5 print("设置机器人安装角度错误码",error)
```



## 4.20. 设置末端负载质心坐标

原型	<code>SetLoadCoord(x,y,z)</code>
描述	设置末端负载质心坐标,错误负载质心设置可能会导致拖动模式下机器人失控
必选参数	<ul style="list-style-type: none"><li><code>x</code>, <code>y</code>, <code>z</code>: 质心坐标, 单位[mm]</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

### 4.20.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 error = robot.SetLoadCoord(3.0,4.0,5.0) #!!! 负载质心设置应于实际相符(错误负载质心设置可能会导致拖动模式下机器人失控)
5 print("设置负载质心错误码",error)
```

## 4.21. 等待指定时间

原型	<code>WaitMs(t_ms)</code>
描述	等待指定时间
必选参数	<ul style="list-style-type: none"><li><code>t_ms</code>:单位[ms]</li></ul>
默认参数	无
返回值	错误码 成功-0 失败- errcode

### 4.21.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 error = robot.WaitMs(1000)
5 print("等待指定时间错误码",error)
```

## 4.22. 设置机器人加速度

在 python 版本加入: SDK-v2.0.4

原型	<code>Set0accScale(acc)</code>
描述	设置机器人加速度
必选参数	<ul style="list-style-type: none"><li><code>acc</code>:机器人加速度百分比</li></ul>
默认参数	无

返回值	错误码 成功-0 失败- errcode
-----	----------------------

### 4.22.1. 代码示例

```
1  from fairino import Robot
2  # 与机器人控制器建立连接，连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  robot.SetOaccScale (20)
```