

6. 机器人状态查询

6.1. 获取机器人安装角度

原型	<code>GetRobotInstallAngle()</code>
描述	获取机器人安装角度
参数	无
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） [yangle,zangle] yangle-倾斜角,zangle-旋

6.1.1. 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 ret = robot.GetRobotInstallAngle()
5 print("获取机器人安装角度", ret)
```

6.2. 获取系统变量值

原型	<code>GetSysVarValue(id)</code>
描述	获取系统变量值
必选参数	<ul style="list-style-type: none"><code>id</code>: 系统变量编号，范围[1~20]
默认参数	无
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） var_value: 系统变量值

6.2.1. 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 for i in range(1,21):
6     error = robot.GetSysVarValue(i)
7     print("系统变量编号:",i,"值", error)
```

6.3. 获取当前关节位置(角度)

原型	<code>GetActualJointPosDegree(flag = 1)</code>
描述	获取关节当前位置(角度)
必选参数	<ul style="list-style-type: none"><code>flag</code>: 0-阻塞, 1-非阻塞
默认参数	无
返回值	错误码 成功-0 失败- errcode - 返回值 (调用成功返回) joint_pos=[j1,j2,j3,j4,j5,j6]

6.3.1. 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 ret = robot.GetActualJointPosDegree()
5 print("获取当前关节位置 (角度)", ret)
```

6.4. 获取当前关节位置(弧度)

原型	<code>GetActualJointPosRadian(flag = 1)</code>
描述	获取关节当前位置(弧度)
必选参数	无
默认参数	<ul style="list-style-type: none"><code>flag</code>: 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值 (调用成功返回) joint_pos=[j1,j2,j3,j4,j5,j6]

6.4.1. 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 ret = robot.GetActualJointPosRadian()
5 print("获取当前关节位置 (弧度)", ret)
```

6.5. 获取关节反馈速度-deg/s

原型	<code>GetActualJointSpeedsDegree (flag = 1)</code>
描述	获取关节反馈速度-deg/s
必选参数	无
默认参数	<ul style="list-style-type: none"><code>flag</code>: 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值 (调用成功返回) speed=[j1,j2,j3,j4,j5,j6]

6.5.1. 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  ret = robot.GetActualJointSpeedsDegree()
5  print("获取关节反馈速度-deg/s", ret)
```

6.6. 获取TCP指令合速度

GetTargetTCPCompositeSpeed (flag = 1)

获取TCP指令合速度

...

flag : 0-阻塞, 1-非阻塞 默认1

错误码 成功-0 失败- errcode - 返回值（调用成功返回） [tcp_speed,ori_speed] tcp_speed 线性合速度 ori_sp

6.6.1. 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  ret = robot.GetTargetTCPCompositeSpeed()
5  print("获取TCP指令合速度", ret)
```

6.7. 获取TCP反馈合速度

GetActualTCPCompositeSpeed (flag = 1)

获取TCP反馈合速度

...

flag : 0-阻塞, 1-非阻塞 默认1

错误码 成功-0 失败- errcode - 返回值（调用成功返回） [tcp_speed,ori_speed] tcp_speed 线性合速度 ori_sp

6.7.1. 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  ret = robot.GetActualTCPCompositeSpeed()
5  print("获取TCP反馈合速度", ret)
```

6.8. 获取TCP指令速度

原型	<code>GetTargetTCPSpeed (flag = 1)</code>
描述	获取TCP指令速度
必选参数	无
默认参数	<ul style="list-style-type: none"><code>flag</code>: 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） speed [x,y,z,rx,ry,rz]

6.8.1. 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 ret = robot.GetTargetTCPSpeed()
5 print("获取TCP指令速度", ret)
```

6.9. 获取TCP反馈速度

原型	<code>GetActualTCPSpeed (flag = 1)</code>
描述	获取TCP反馈速度
必选参数	无
默认参数	<ul style="list-style-type: none"><code>flag</code>: 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） speed [x,y,z,rx,ry,rz]

6.9.1. 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 ret = robot.GetActualTCPSpeed()
5 print("获取TCP反馈速度", ret)
```

6.10. 获取当前工具位姿

原型	<code>GetActualTCPPose (flag = 1)</code>
描述	获取当前工具位姿
必选参数	无
默认参数	<ul style="list-style-type: none"><code>flag</code>: 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） tcp_pose=[x,y,z,rx,ry,rz]

6.10.1. 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  ret = robot.GetActualTCPPose()
5  print("获取当前工具位姿", ret)
```

6.11. 获取当前工具坐标系编号

原型	<code>GetActualTCPNum (flag = 1)</code>
描述	获取当前工具坐标系编号
必选参数	无
默认参数	<ul style="list-style-type: none"><code>flag</code> : 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） tool_id:工具坐标系编号

6.11.1. 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  ret = robot.GetActualTCPNum()
5  print("获取当前工具坐标系编号", ret)
```

6.12. 获取当前工件坐标系编号

原型	<code>GetActualWObjNum (flag = 1)</code>
描述	获取当前工件坐标系编号
必选参数	无
默认参数	<code>flag</code> : 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） wobj_id:工件坐标系编号

6.12.1. 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  ret = robot.GetActualWObjNum()
5  print("获取当前工件坐标系编号", ret)
```

6.13. 获取当前末端法兰位姿

原型	<code>GetActualToolFlangePose (flag = 1)</code>
----	---

描述	获取当前末端法兰位姿
必选参数	无
默认参数	<ul style="list-style-type: none"> <code>flag</code>: 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） flange_pose=[x,y,z,rx,ry,rz]

6.13.1. 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  ret = robot.GetActualToolFlangePose()
5  print("获取当前末端法兰位姿", ret)

```

6.14. 逆运动学求解

原型	<code>GetInverseKin(type,desc_pos,config)</code>
描述	逆运动学，笛卡尔位姿求解关节位置
必选参数	<ul style="list-style-type: none"> <code>type</code>:0-绝对位姿(基坐标系), 1-相对位姿（基坐标系）, 2-相对位姿（工具坐标系） <code>desc_pose</code>: [x,y,z,rx,ry,rz],工具位姿，单位[mm][°]
默认参数	<ul style="list-style-type: none"> <code>config</code>:关节配置, [-1]-参考当前关节位置求解, [0~7]-依据关节配置求解 默认-1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） joint_pos=[j1,j2,j3,j4,j5,j6]

6.14.1. 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  J1=[95.442,-101.149,-98.699,-68.347,90.580,-47.174]
5  P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
6  ret = robot.GetInverseKin(0,P1,config=-1)
7  print("逆运动学，笛卡尔位姿求解关节位置", ret)

```

6.15. 逆运动学求解-指定参考位置

原型	<code>GetInverseKinRef(type,desc_pos, joint_pos_ref)</code>
描述	逆运动学，工具位姿求解关节位置，参考指定关节位置求解
必选参数	<ul style="list-style-type: none"> <code>type</code>:0-绝对位姿(基坐标系), 1-相对位姿（基坐标系）, 2-相对位姿（工具坐标系） <code>desc_pos</code>: [x,y,z,rx,ry,rz]工具位姿，单位[mm][°] <code>joint_pos_ref</code>: [j1,j2,j3,j4,j5,j6], 关节参考位置，单位[°]
默认参数	无
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） joint_pos=[j1,j2,j3,j4,j5,j6]

6.15.1. 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 J1=[95.442,-101.149,-98.699,-68.347,90.580,-47.174]
5 P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
6 ret = robot.GetInverseKinRef(0,P1,J1)
7 print("逆运动学，工具位姿求解关节位置，参考指定关节位置求解", ret)
```

6.16. 逆运动学求解-是否有解

原型	<code>GetInverseKinHasSolution(type,desc_pos, joint_pos_ref)</code>
描述	逆运动学，工具位姿求解关节位置 是否有解
必选参数	<ul style="list-style-type: none"><code>type</code>:0-绝对位姿(基坐标系), 1-相对位姿（基坐标系），2-相对位姿（工具坐标系）<code>desc_pos</code>: [x,y,z,rx,ry,rz]工具位姿，单位[mm][°]<code>joint_pos_ref</code>: [j1,j2,j3,j4,j5,j6]，关节参考位置，单位[°]
默认参数	无
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） result: “True” -有解， “False” -无解

6.16.1. 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 J1=[95.442,-101.149,-98.699,-68.347,90.580,-47.174]
5 P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
6 ret = robot.GetInverseKinHasSolution(0,P1,J1)
7 print("逆运动学，工具位姿求解关节位置是否有解", ret)
```

6.17. 正运动学求解

原型	<code>GetForwardKin(joint_pos)</code>
描述	正运动学，关节位置求解工具位姿
必选参数	<ul style="list-style-type: none"><code>joint_pos</code>: [j1,j2,j3,j4,j5,j6]:关节位置，单位[°]
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode返回值（调用成功返回） desc_pos=[x,y,z,rx,ry,rz]

6.17.1. 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  J1=[95.442,-101.149,-98.699,-68.347,90.580,-47.174]
5  ret = robot.GetForwardKin(J1)
6  print("正运动学，关节位置求解工具位姿", ret)
```

6.18. 获取当前关节转矩

原型	GetJointTorques (flag = 1)
描述	获取当前关节转矩
必选参数	无
默认参数	flag：0-阻塞，1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回）torques=[j1,j2,j3,j4,j5,j6]

6.18.1. 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  ret = robot.GetJointTorques()
5  print("获取当前关节转矩", ret)
```

6.19. 获取当前负载的重量

原型	GetTargetPayload (flag = 1)
描述	获取当前负载的质量
必选参数	无
默认参数	flag：0-阻塞，1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回）weight 单位 [kg]

6.19.1. 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  ret = robot.GetTargetPayload(0)
5  print("获取当前负载的质量", ret)
```

6.20. 获取当前负载的质心

原型	GetTargetPayloadCog (flag = 1)
----	--------------------------------

描述	获取当前负载的质心
必选参数	无
默认参数	<code>flag</code> : 0-阻塞, 1-非阻塞 默认1
返回值	错误码 成功-0 失败- errcode - 返回值 (调用成功返回) tcp_offset=[x,y,z,rx,ry,rz]: 相对位姿, 单位 [r

6.20.1. 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 ret = robot.GetTargetPayloadCog(0)
5 print("获取当前负载的质心", ret)
```

6.21. 获取当前工具坐标系

	<code>GetTCPOffset (flag = 1)</code>
	获取当前工具坐标系
数	无
数	<code>flag</code> : 0-阻塞, 1-非阻塞 默认1
直	错误码 成功-0 失败- errcode - 返回值 (调用成功返回) tcp_offset=[x,y,z,rx,ry,rz]: 相对位姿, 单位 [r

6.21.1. 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 ret = robot.GetTCPOffset()
5 print("获取当前工具坐标系", ret)
```

6.22. 获取当前工件坐标系

	<code>GetWObjOffset (flag = 1)</code>
	获取当前工件坐标系
数	无
数	<code>flag</code> : 0-阻塞, 1-非阻塞 默认1
直	错误码 成功-0 失败- errcode - 返回值 (调用成功返回) wobj_offset=[x,y,z,rx,ry,rz]: 相对位姿, 单位 [r

6.22.1. 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 ret = robot.GetWObjOffset()
5 print("获取当前工件坐标系", ret)
```

6.23. 获取关节软限位角度

1

返回值 (调用成功返回) [j1min,j1max,j2min,j2max,j3min,j3max, j4min,j4max,j5min, j5max, j6min,j6max].

6.23.1. 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 ret = robot.GetJointSoftLimitDeg()
5 print("获取关节软限位角度", ret)
```

6.24. 获取系统时间

原型	GetSystemClock()
描述	获取系统时间
必选参数	无
默认参数	无
返回值	- 错误码 成功-0 失败- errcode - 返回值（调用成功返回） t_ms: 单位 [ms]

6.24.1. 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 ret = robot.GetSystemClock()
5 print("获取系统时间", ret)
```

6.25. 获取机器人当前关节配置

原型	GetRobotCurJointsConfig()
----	---------------------------

描述	获取机器人当前关节配置
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） config: 范围 [0~7]

6.25.1. 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 ret = robot.GetRobotCurJointsConfig()
5 print("获取机器人当前关节配置", ret)
```

6.26. 获取默认速度

原型	<code>GetDefaultTransVel()</code>
描述	获取默认速度
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） vel: 单位 [mm/s]

6.26.1. 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 ret = robot.GetDefaultTransVel()
5 print("获取默认速度", ret)
```

6.27. 查询机器人运动是否完成

原型	<code>GetRobotMotionDone()</code>
描述	查询机器人运动是否完成
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） state:0-未完成，1-完成

6.27.1. 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 ret = robot.GetRobotMotionDone()
5 print("查询机器人运动是否完成", ret)
```

6.28. 查询机器人错误码

<code>GetRobotErrorCode()</code>
查询机器人错误码
无
无
错误码 成功-0 失败- errcode - 返回值（调用成功返回） [maincode subcode] maincode 主错误码 subcode 子错误码

6.28.1. 代码示例

<pre>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 ret = robot.GetRobotErrorCode() 5 print("查询机器人错误码", ret)</pre>

6.29. 查询机器人示教管理点位数据

<code>tRobotTeachingPoint(name)</code>
查询机器人示教管理点位数据
name: 点名
错误码 成功-0 失败- errcode - 返回值（调用成功返回） 点位数据[x,y,z,rx,ry,rz,j1,j2,j3,j4,j5,j6,tool, wobj,speed, ...]

6.29.1. 代码示例

<pre>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 ret = robot.GetRobotTeachingPoint("11") 5 print("查询机器人示教管理点位数据错误码", ret)</pre>

6.30. 获取SSH公钥

原型	<code>GetSSHKeygen()</code>
描述	获取SSH公钥
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode - 返回值（调用成功返回） keygen 公钥

6.30.1. 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 ret = robot.GetSSHKeygen() #获取SSH
5 print("获取SSH", ret)
```

6.31. 计算指定路径下文件的MD5值

	<code>ComputeFileMD5(file_path)</code>
	计算指定路径下文件的MD5值
收	<ul style="list-style-type: none"><code>file_path</code>: 文件路径包含文件名，默认Traj文件夹路径为:/fruser/traj/,如/fruser/traj/trajHelix_air
收	无
错误码	成功-0 失败- errcode - 返回值（调用成功返回） keygen 公钥

6.31.1. 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 ret = robot.ComputeFileMD5("/fruser/201.lua") #计算指定路径下文件的MD5值
5 print("计算指定路径下文件的MD5值", ret)
```

6.32. 获取机器人版本信息

在 python 版本加入: SDK-v2.0.1

原型	<code>GetSoftwareVersion()</code>
描述	获取机器人版本信息
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcodeReturn: (if success) robotModel, webVersion, controllerVersion

6.32.1. 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
5 ret = robot.GetSoftwareVersion()
6 print("GetSoftwareVersion(): ", ret)
```

6.33. 获取机器人硬件版本信息

在 python 版本加入: SDK-v2.0.1

息

[- errcode
s) ctrlBoxBoardVersion, driver1Version, driver2Version , driver3Version, driver4Version, driver5Version

6.33.1. 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  ret = robot.GetSlaveHardVersion()
5  print("GetSlaveHardVersion(): ", ret)
```

6.34. 获取机器人固件版本信息

在 python 版本加入: SDK-v2.0.1

息

[- errcode
s) ctrlBoxBoardVersion, driver1Version, driver2Version , driver3Version, driver4Version, driver5Version

6.34.1. 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  ret = robot.GetSlaveFirmVersion()
5  print("GetSlaveFirmVersion(): ", ret)
```

6.35. 获取DH补偿参数

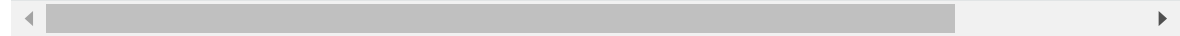
在 python 版本加入: SDK-v2.0.1

Compensation()

DH补偿参数

错误码 成功-0 失败- errcode

return: (if success) dhCompensation 机器人DH参数补偿值(mm) [cmpstD1,cmpstA2,cmpstA3,cmpstD4,c



6.35.1. 代码示例

```
1  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.GetDHCompensation()
5  print(error)
```