

3. 机器人IO

3.1. 设置控制箱数字量输出

原型	<code>SetD0(id, status, smooth=0, block=0)</code>
描述	设置控制箱数字量输出
必选参数	<ul style="list-style-type: none"><code>id</code>:io编号, 范围[0~15];<code>status</code>:0-关, 1-开;
默认参数	<ul style="list-style-type: none"><code>smooth</code>:0-不平滑, 1-平滑 默认0;<code>block</code>:0-阻塞, 1-非阻塞 默认0
返回值	错误码 成功-0 失败- errcode

3.1.1. 代码示例

```
1  from fairino import Robot
2  import Robot
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  # 测试控制箱D0
6  for i in range(0,16):
7      error = robot.SetD0(i,1)      #打开控制箱D0
8      time.sleep(1)
9  for i in range(0,16):
10     robot.SetD0(i,0)      #关闭控制箱D0
```

3.2. 设置工具数字量输出

原型	<code>SetToolD0 (id, status, smooth=0, block=0)</code>
描述	设置工具数字量输出
必选参数	<ul style="list-style-type: none"><code>id</code>:io编号, 范围[0~1];<code>status</code>:0-关, 1-开;
默认参数	<ul style="list-style-type: none"><code>smooth</code>:0-不平滑, 1-平滑;<code>block</code>:0-阻塞, 1-非阻塞。



返回值

错误码 成功-0 失败- errcode

3.2.1. 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 # 测试工具D0
5 error_tooldo = 0
6 for i in range(0,2):
7     error = robot.SetToolD0(i,1)    #打开工具D0
8     robot.WaitMs(1000)
9     for i in range(0,2):
10        error = robot.SetToolD0(i,0)    #关闭工具D0
```

3.3. 设置控制箱模拟量输出

原型	<code>SetA0(id,value,block=0)</code>
描述	设置控制箱模拟量输出
必选参数	<ul style="list-style-type: none"><code>id</code>:io编号, 范围[0~1];<code>value</code>:电流或电压值百分比, 范围[0~100%]对应电流值[0~20mA]或电压[0~10V];
默认参数	<ul style="list-style-type: none"><code>block</code>: [0]-阻塞, [1]-非阻塞 默认0
返回值	错误码 成功-0 失败- errcode

3.3.1. 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 # 测试控制箱A0
5 error = robot.SetA0(0,100.0)
6 print("设置A00错误码:", error)
7 error = robot.SetA0(1,100.0)
8 print("设置A01错误码:", error)
```

3.4. 设置工具模拟量输出

原型	<code>SetToolA0(id,value,block=0)</code>
描述	设置工具模拟量输出
必选参数	<ul style="list-style-type: none"><code>id</code>:io编号, 范围[0];<code>value</code>:电流或电压值百分比, 范围[0~100%]对应电流值[0~20mA]或电压[0~10V];



latest



默认参数	<ul style="list-style-type: none"><code>block</code>: [0]-阻塞, [1]-非阻塞 默认0
返回值	错误码 成功-0 失败- <code>errcode</code>

3.4.1. 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 # 测试末端A0
5 error = robot.SetToolA0(0,100.0)
6 print("设置ToolA0错误码:", error)
7 Robot.WaitMs(1000)
8 error = robot.SetToolA0(0,0.0)
9 print("设置ToolA0错误码:", error)
```

3.5. 获取控制箱数字量输入

原型	<code>GetDI(id, block=0)</code>
描述	获取控制箱数字量输入
必选参数	<ul style="list-style-type: none"><code>id</code>: io编号, 范围[0~15];
默认参数	<ul style="list-style-type: none"><code>block</code>: 0-阻塞, 1-非阻塞 默认0
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- <code>errcode</code><code>di</code>: 0-低电平, 1-高电平

3.5.1. 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 error = robot.GetDI(0,0)
5 print("获取DI0",error)
```

3.6. 获取工具数字量输入

原型	<code>GetToolDI(id, block=0)</code>
描述	获取工具数字量输入
必选参数	<ul style="list-style-type: none"><code>id</code>: io编号, 范围[0~1];
默认参数	<ul style="list-style-type: none"><code>block</code>: 0-阻塞, 1-非阻塞 默认0

返回值	错误码 成功-0 失败- errcode - <code>di</code> : 0-低电平, 1-高电平
-----	---

3.6.1. 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 tool_di = robot.GetToolDI(1,0)
5 print("获取ToolDI",tool_di)
```

3.7. 等待控制箱数字量输入

原型	<code>WaitDI(id,status,maxtime,opt)</code>
描述	等待控制箱数字量输入
选参数	<ul style="list-style-type: none"><code>id</code>:io编号, 范围[0~15];<code>status</code>:0-关, 1-开;<code>maxtime</code>:最大等待时间, 单位[ms];<code>opt</code>:超时后策略, 0-程序停止并提示超时, 1-忽略超时提示程序继续执行, 2-一直等待
默认参数	无
返回值	错误码 成功-0 失败- errcode

3.7.1. 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 max_waittime = 2000
5 #等待控制箱DI
6 error = robot.WaitDI(0,1,max_waittime,0)
7 print("WaitDI错误码",error)
```

3.8. 等待控制箱多路数字量输入

原型	<code>WaitMultiDI(mode,id,status,maxtime,opt)</code>
描述	等待控制箱多路数字量输入
选参数	<ul style="list-style-type: none"><code>mode</code>: [0]-多路与, [1]-多路或;<code>id</code>:io编号, bit0~bit7对应DI0~DI7, bit8~bit15对应CI0~CI7;<code>status</code>:bit0~bit7对应DI0~DI7状态, bit8~bit15对应CI0~CI7状态位的<code>maxtime</code>:最大等待时间, 单位[ms];<code>opt</code>:超时后策略, 0-程序停止并提示超时, 1-忽略超时提示程序继续执行, 2-一直等待

人参数	无
返回值	错误码 成功-0 失败- errcode

3.8.1. 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 max_waittime = 2000
5 #等待控制箱多路DI
6 error = robot.WaitMultiDI(1,3,1,max_waittime,0)
7 print("WaitMultiDI错误码",error)
```

3.9. 等待工具数字量输入

原型	<code>WaitToolDI(id,status,maxtime,opt)</code>
描述	等待末端数字量输入
选参数	<ul style="list-style-type: none"><code>id</code>:io编号, 范围[0~1];<code>status</code>:0-关, 1-开;<code>maxtime</code>:最大等待时间, 单位[ms];<code>opt</code>:超时后策略, 0-程序停止并提示超时, 1-忽略超时提示程序继续执行, 2-一直等待
默认参数	无
返回值	错误码 成功-0 失败- errcode

3.9.1. 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 max_waittime = 2000
5 #等待工具DI
6 error = robot.WaitToolDI(1,1,max_waittime,0)
7 print("WaitToolDI错误码",error)
```

3.10. 获取控制箱模拟量输入

原型	<code>GetAI(id, block = 0)</code>
描述	获取控制箱模拟量输入
选参数	<ul style="list-style-type: none"><code>id</code>:io编号, 范围[0~1];
默认参数	<ul style="list-style-type: none"><code>block</code>:0-阻塞, 1-非阻塞 默认0

返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>value</code>: 输入电流或电压值百分比，范围 [0~100] 对应电流值 [0~20mA] 或电压 [0~10V]
-----	--

3.10.1. 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 error = robot.GetAI(0)
5 print("获取AI0",error)
```

3.11. 获取工具模拟量输入

原型	<code>GetToolAI (id, block = 0)</code>
描述	获取末端模拟量输入
可选参数	<ul style="list-style-type: none"><code>id</code>:io编号，范围[0];
默认参数	<ul style="list-style-type: none"><code>block</code>:0-阻塞，1-非阻塞 默认0
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>value</code>: 输入电流或电压值百分比，范围 [0~100] 对应电流值 [0~20mA] 或电压 [0~10V]

3.11.1. 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 error = robot.GetToolAI(0)
5 print("获取ToolAI0",error)
```

3.12. 等待控制箱模拟量输入

原型	<code>WaitAI(id,sign,value,maxtime,opt)</code>
描述	等待控制箱模拟量输入
可选参数	<ul style="list-style-type: none"><code>id</code>:io编号，范围[0~1];<code>sign</code>:0-大于，1-小于<code>value</code>:输入电流或电压值百分比，范围[0~100]对应电流值[0~20mA]或电压[0~10V];<code>maxtime</code>:最大等待时间，单位[ms];<code>opt</code>:超时后策略，0-程序停止并提示超时，1-忽略超时提示程序继续
默认参数	无

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

3.12.1. 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 max_waittime = 2000
5 #等待控制箱AI
6 error = robot.WaitAI(0,0,50,max_waittime,1) #忽略超时提示程序继续执行
7 print("WaitAI错误码",error)
```

3.13. 等待工具模拟量输入

原型	WaitToolAI(id,sign,value,maxtime,opt)
描述	等待末端模拟量输入
必选参数	<ul style="list-style-type: none"><code>id</code>:io编号, 范围[0];<code>sign</code>:0-大于, 1-小于<code>value</code>:输入电流或电压值百分比, 范围[0~100]对应电流值[0~20mA]或电压[0~10V];<code>maxtime</code>:最大等待时间, 单位[ms];<code>opt</code>:超时后策略, 0-程序停止并提示超时, 1-忽略超时提示程序继续执行, 2-一直等待
默认参数	无
返回值	错误码 成功-0 失败- errcode

3.13.1. 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 max_waittime = 2000
5 #等待工具AI
6 error = robot.WaitToolAI(0,0,50,max_waittime,0)
7 print("WaitToolAI错误码",error)
```

3.14. 设置控制箱DO停止/暂停后输出是否复位

在 python 版本加入: SDK-v2.0.5

原型	SetOutputResetCtlBoxDO(resetFlag)
描述	设置控制箱DO停止/暂停后输出是否复位
必选参数	<ul style="list-style-type: none"><code>resetFlag</code>: 0-不复位; 1-复位

默认参数	无
返回值	错误码 成功-0 失败- errcode



latest



3.14.1. 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
5  robot = Robot.RPC('192.168.58.2')
6  time.sleep(5)
7  error = robot.SetD0(1,1)
8  print("SetD0 1 return:",error)
9
10 error = robot.SetD0(3,1)
11 print("SetD0 3 return:",error)
12
13 error = robot.SetToolD0(1,1)
14 print("SetToolD0 return:",error)
15
16 error = robot.SetA0(0,25)
17 print("SetA0 0 return:",error)
18
19 error = robot.SetA0(1,87)
20 print("SetA0 1 return:",error)
21
22 error = robot.SetToolA0(0,54)
23 print("SetToolA0 return:",error)
24
25 error = robot.SetOutputResetCtlBoxD0(1)
26 print("SetOutputResetCtlBoxD0 return:",error)
27
28 error = robot.SetOutputResetCtlBoxA0(1)
29 print("SetOutputResetCtlBoxA0 return:",error)
30
31 error = robot.SetOutputResetAxleD0(1)
32 print("SetOutputResetCtlBoxD0 return:",error)
33
34 error = robot.SetOutputResetAxleA0(1)
35 print("SetOutputResetCtlBoxA0 return:",error)
36
37 error = robot.ProgramRun()
38 print("ProgramRun return:",error)
39 time.sleep(3)
40 error = robot.ProgramStop()
41 print("ProgramPause return:",error)
42
43 time.sleep(5)
44
45 error = robot.SetD0(1,1)
46 print("SetD0 1 return:",error)
47
48 error = robot.SetD0(3,1)
49 print("SetD0 3 return:",error)
50
51 error = robot.SetToolD0(1,1)
52 print("SetToolD0 return:",error)
53
54 error = robot.SetA0(0,25)
55 print("SetA0 0 return:",error)
56
57 error = robot.SetA0(1,87)
58 print("SetA0 1 return:",error)
59
60 error = robot.SetToolA0(0,54)
```



latest



```

61 print("SetToolAO return:",error)
62 error = robot.SetOutputResetCtlBoxD0(0)
63 print("SetOutputResetCtlBoxD0 return:",error)
64
65 error = robot.SetOutputResetCtlBoxA0(0)
66 print("SetOutputResetCtlBoxA0 return:",error)
67
68 error = robot.SetOutputResetAxleD0(0)
69 print("SetOutputResetCtlBoxD0 return:",error)
70
71 error = robot.SetOutputResetAxleA0(0)
72 print("SetOutputResetCtlBoxA0 return:",error)
73
74 error = robot.ProgramRun()
75 print("ProgramRun return:",error)
76 time.sleep(3)
77 error = robot.ProgramStop()
78 print("ProgramPause return:",error)

```

3.15. 设置控制箱AO停止/暂停后输出是否复位

在 python 版本加入: SDK-v2.0.5

原型	<code>SetOutputResetCtlBoxD0(resetFlag)</code>
描述	设置控制箱AO停止/暂停后输出是否复位
必选参数	<ul style="list-style-type: none"> <code>resetFlag</code>: 0-不复位; 1-复位
默认参数	无
返回值	错误码 成功-0 失败- errcode

3.16. 设置末端工具DO停止/暂停后输出是否复位

在 python 版本加入: SDK-v2.0.5

原型	<code>SetOutputResetAxleD0(resetFlag)</code>
描述	设置末端工具DO停止/暂停后输出是否复位
必选参数	<ul style="list-style-type: none"> <code>resetFlag</code>: 0-不复位; 1-复位
默认参数	无
返回值	错误码 成功-0 失败- errcode

3.17. 设置末端工具AO停止/暂停后输出是否复位

在 python 版本加入: SDK-v2.0.5

原型	<code>SetOutputResetAxleA0(resetFlag)</code>
描述	设置末端工具AO停止/暂停后输出是否复位
必选参数	<ul style="list-style-type: none"><code>resetFlag</code> : 0-不复位; 1-复位
默认参数	无
返回值	错误码 成功-0 失败- errcode

3.18. 设置扩展DO停止/暂停后输出是否复位

在 *python* 版本加入: SDK-v2.0.5

原型	<code>SetOutputResetExtDO (resetFlag)</code>
描述	设置扩展DO停止/暂停后输出是否复位
必选参数	<ul style="list-style-type: none"><code>resetFlag</code> : 0-不复位; 1-复位
默认参数	无
返回值	错误码 成功-0 失败- errcode

3.18.1. 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
5  robot = Robot.RPC('192.168.58.2')
6
7  error = robot.SetAuxD0(1,True,False,False)
8  print("SetAuxD0 1 return:",error)
9
10 error = robot.SetAuxD0(3,True,False,False)
11 print("SetAuxD0 3 return:",error)
12
13 error = robot.SetAuxA0(0,10,False)
14 print("SetAuxA0 0 return:",error)
15
16 error = robot.SetAuxA0(1,87,False)
17 print("SetAuxA0 1 return:",error)
18
19 error = robot.SetOutputResetExtD0(1)
20 print("SetOutputResetExtD0 return:",error)
21
22 error = robot.SetOutputResetExtA0(1)
23 print("SetOutputResetExtA0 return:",error)
24
25 error = robot.ProgramRun()
26 print("ProgramRun return:",error)
27 time.sleep(3)
28 error = robot.ProgramStop()
29 print("ProgramPause return:",error)
30
31 time.sleep(3)
32 error = robot.SetAuxD0(1,True,False,False)
33 print("SetAuxD0 1 return:",error)
34
35 error = robot.SetAuxD0(3,True,False,False)
36 print("SetAuxD0 3 return:",error)
37
38 error = robot.SetAuxA0(0,10,False)
39 print("SetAuxA0 0 return:",error)
40
41 error = robot.SetAuxA0(1,87,False)
42 print("SetAuxA0 1 return:",error)
43
44 error = robot.SetOutputResetExtD0(0)
45 print("SetOutputResetExtD0 return:",error)
46
47 error = robot.SetOutputResetExtA0(0)
48 print("SetOutputResetExtA0 return:",error)
49
50 error = robot.ProgramRun()
51 print("ProgramRun return:",error)
52 time.sleep(3)
53 error = robot.ProgramStop()
54 print("ProgramPause return:",error)
```



3.19. 设置扩展AO停止/暂停后输出是否复位

在 *python* 版本加入: SDK-v2.0.5

原型	<code>SetOutputResetExtAO (resetFlag)</code>
描述	设置扩展AO停止/暂停后输出是否复位
必选参数	<ul style="list-style-type: none"><code>resetFlag</code>: 0-不复位; 1-复位
默认参数	无
返回值	错误码 成功-0 失败- <code>errcode</code>

