# uArm Python SDK API

**SwiftAPI类属性**

|  |  |  |
| --- | --- | --- |
| **属性** | **功能** | **备注** |
| connected | 连接状态 |  |
| port | 端口 |  |
| baudrate | 波特率 |  |
| power\_status | 电源状态 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**SwiftAPI类接口**

**注意：**

1. **为避免接口参数顺序变更，调用接口时尽量使用关键字参数**
2. **以下接口文档描述的参数顺序不一定是自然顺序，请用关键字传参**
3. **调用接口时如果指定了回调，请注意回调不能阻塞，否则有可能会阻塞某些功能**
4. **返回值需要wait参数为True时才有效，否则返回值就是回调的参数**
5. **回调参数就是返回的结果，返回的结果可能是命令的执行结果或者超时**

|  |  |  |
| --- | --- | --- |
| **API接口名称** | **功能及参数** | **备注** |
| SwiftAPI(…) | 功能：实例初始化  参数：  port/dev\_port: 端口，默认是None，会自动选用第一串口，dev\_port参数是为了兼容pyuf，优先于port参数  baudrate/baud: 波特率，默认是115200，baud参数是为了兼容pyuf，优先于baudrate参数  filters: 串口过滤，字典，如{‘hwid’: ‘USB VID:PID=2341:0042’}  cmd\_pend\_size: 命令缓存最大个数，默认为5  callback\_thread\_pool\_size: 处理回调用到的线程池大小，默认为0，即不适用线程处理回调  do\_not\_open: 是否不自动连接，默认是False，即在实例初始化时自动尝试连接一遍，如果设为True，则需要另行调用connect接口 | 1. 当不指定串口连接时，自动连接的串口不一定是串口号最小的 2. 一个程序创建多个实例来连接多个串口时，如果都不指定串口名字，会自动连接不同的串口 |
| connect(…) | 功能：连接串口，实例在初始化时已经连接(除非指定了do\_not\_open参数)  参数：  port: 端口，默认是None，会连接实例初始化时设定的端口  baudrate: 波特率，默认是None，会使用实例初始化设定的波特率  timeout: 串口的读超时时间，默认是None，会使用实例初始化设定的timeout |  |
| disconnect(…) | 功能：断开连接  参数：  is\_clean: 是否清除线程池，默认是True |  |
| waiting\_ready(…) | 功能：等待机械臂准备好  参数：  timeout：等待的超时时间，默认是3s |  |
| send\_cmd\_sync(…) | 功能：发送同步命令，即发送命令并等到执行结果返回或超时  参数：  msg: 命令，字符串，默认为None  timeout: 超时时间，默认为None，使用实例默认的5s  返回值：   1. 执行结果 2. ‘TIMEOUT’ |  |
| send\_cmd\_async(…) | 功能：发送异步命令，即发送命令但不等待执行结果返回，但可以指定回调  参数：  msg: 命令，字符串  timeout: 超时时间，默认为None，使用实例默认的5s  callback: 回调函数 |  |
| get\_device\_info(…) | 功能：获取设备信息  参数：  timeout: 超时时间，默认为None, 使用10s  返回值：{  “device\_type”: “”,  “hardware\_version”: “”,  “firmware\_version”: “”,  “api\_version”: “”,  “device\_unique”: “”  } |  |
| reset(…) | 功能：复位  参数：  speed: 速度，默认为None，使用最后一次运动指定的速度或者初始的1000  wait: 是否等待，默认为True  timeout: 超时时间，默认为None，使用实例默认的 |  |
| get\_mode(…) | 功能：获取模式  参数：  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的  callback：回调函数  返回值：wait为True有效   1. mode, 0表示常规模式，1表示激光模式，2表示3D打印模式，3表示画笔模式 2. ‘TIMEOUT’ | 仅仅支持SwiftPro |
| set\_mode(…) | 功能：设置模式  参数：  mode：要设置的模式（0, 1, 2, 3），默认0  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的  callback：回调函数  返回值：   1. mode, 0表示常规模式，1表示激光模式，2表示3D打印模式，3表示画笔模式 2. ‘TIMEOUT’ | 仅仅支持SwiftPro |
| get\_position(…) | 功能：获取当前坐标  参数：  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的  callback：回调函数  返回值：   1. [x, y, z] 2. ‘TIMEOUT’ |  |
| set\_position(...) | 功能：设置移动坐标  参数：  x：X坐标，默认None，使用最后一次移动的X值或初始的150  y：Y坐标，默认None，使用最后一次移动的Y值或初始的0  z：Z坐标，默认None，使用最后一次移动的Z值或初始的150  speed：速度，默认None，使用最后一次移动的speed值或初始的1000  relative：是否是相对移动，默认False  wait：是否等待结果返回，默认False  timeout: 超时时间，默认为10  callback：回调函数  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| get\_polar(…) | 功能：获取当前极坐标  参数：  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的  callback：回调函数  返回值：   1. [stretch, rotation, height] 2. ‘TIMEOUT’ |  |
| set\_polar(…) | 功能：设置移动极坐标  参数：  stretch/s：长度，默认为None，s参数是为了兼容pyuf，单位mm  rotation/r：旋转角度，默认为None，r参数是为了兼容pyuf，单位degree，范围0-180  height/h：高度，默认为None，h参数是为了兼容pyuf，单位mm  speed：速度，默认None，使用最后一次移动的speed值或初始的1000  relative：是否是相对移动，默认False  wait：是否等待结果返回，默认False  timeout: 超时时间，默认为10  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| get\_servo\_angle(…) | 功能：获取电机角度  参数：  servo\_id：电机号(0: BOTTOM, 1: LEFT, 2: RIGHT, 3: HAND)，默认为None，返回所有的  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. 如果指定了电机号并且存在则返回电机角度，否则返回电机角度列表 2. ‘TIMEOUT’ |  |
| set\_servo\_angle(…) | 功能：设置电机角度  参数：  servo\_id：电机号，默认0  angle：电机角度，默认90  speed: 速度，默认None，使用最后一次移动的speed值或初始的1000  wait：是否等待结果返回，默认False  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| set\_wrist(…) | 功能：设置末端电机角度  参数：  angle：电机角度，默认90  speed: 速度，默认None，使用最后一次移动的speed值或初始的1000  wait：是否等待结果返回，默认False  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| get\_servo\_attach(…) | 功能：获取电机是否锁住  参数：  servo\_id：电机号，默认0  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. True/False 2. ‘TIMEOUT’ |  |
| set\_servo\_attach(…) | 功能：锁定电机  参数：  servo\_id：电机号，默认None，即全部电机  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| set\_servo\_detach(…) | 功能：解锁电机  参数：  servo\_id：电机号，默认None，即全部电机  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| set\_buzzer(…) | 功能：蜂鸣器控制  参数：  frequency/freq：频率，默认None，为1000HZ，freq参数是为了兼容pyuf  duration: time：时间，默认None，为2s，time参数是为了兼容pyuf  wait：是否等待结果返回，默认False  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| set\_pump(…) | 功能：吸头控制  参数：  on：是否打开，True打开，False关闭，默认False  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| set\_gripper(…) | 功能：夹子控制  参数：  catch：True为合上，False为打开，默认False  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, 参照开发手册里的报错表 |  |
| get\_analog(…) | 功能：获取指定引脚的模拟电平  参数：  pin：引脚，默认0  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. 模拟电平 2. ‘TIMEOUT’ |  |
| get\_digital() | 功能：获取指定引脚的数字电平  参数：  pin：引脚，默认0  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. 数字电平 2. ‘TIMEOUT’ |  |
| get\_limit\_switch(…) | 功能：获取吸头是否限位  参数：  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. True/False 2. ‘TIMEOUT’ |  |
| get\_gripper\_catch(…) | 功能：获取夹子状态  参数：  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. 0: stop, 1: working, 3: catch thing 2. ‘TIMEOUT’ |  |
| get\_pump\_status(…) | 功能：获取吸泵状态  参数：  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. 0: stop, 1: working, 3: catch thing   ‘TIMEOUT’ |  |
| grove\_init(…) | 功能：初始化grove，在Port ｛pin｝初始化模块 {grove\_type}  参数：  pin：pin/port，默认为None， 必须传  grove\_type：模块，必须传  value：值，根据不同模块决定是否传  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ | M2305 P{} N{} V{} |
| grove\_control(…) | 功能：grove控制  参数：  pin：pin/port，默认None，必须传  value：值  wait：是否等待结果返回，默认True  timeout: 超时时间，默认为None，使用实例默认的5s  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ |  |
| register\_grove\_callback(…) | 功能：注册grove事件的回调函数  参数：  pin：pin/port，默认None，必须传  grove\_type：模块，默认None，必须传  callback：回调函数，默认为None |  |
| set\_report\_grove(…) | 功能：开启grove自动上报  参数：  pin：pin/port，默认None，必须传  interval：上报间隔，默认0.5，0表示停止上报  callback：回调函数，默认为None  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ |  |
| register\_power\_callback(…) | 功能：注册电源事件回调函数  参数：  callback：回调函数，默认为None |  |
| register\_report\_position\_callback(…) | 功能：注册位置上报事件回调函数  参数：  callback：回调函数，默认为None |  |
| set\_report\_position(…) | 功能：设置位置上报  参数：  Interval：上报间隔，默认1s，0表示停止上报  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ | 仅仅支持SwiftPro |
| register\_key0\_callback(..) | 功能：注册按键0（BUTTON\_MENU）事件回调函数  参数：  callback：回调函数，默认为None |  |
| register\_key1\_callback(..) | 功能：注册按键1（BUTTON\_PLAY）事件回调函数  参数：  callback：回调函数，默认为None |  |
| set\_report\_keys(…) | 功能：设置按键上报  参数：  on/is\_on: True/False, True为开启上报，False为停止上报，默认True，is\_on参数是为了兼容pyuf  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ | 仅仅支持SwiftPro |
| register\_limit\_switch\_callback(…) | 功能：z注册吸头限位事件回调函数  参数：  callback：回调函数，默认为None |  |
| flush\_cmd(…) | 功能：等待命令缓存中的所有命令返回或者超时  参数：  timeout: 等待的超时时间，默认为None，即不超时  返回值：   1. ‘OK’ 2. ‘TIMEOUT’ |  |
|  |  |  |
|  |  |  |
|  |  |  |