



# 福昕PDF编辑器

• 永久 • 轻巧 • 自由

升级会员

批量购买



**永久使用**

无限制使用次数



**极速轻巧**

超低资源占用，告别卡顿慢



**自由编辑**

享受Word一样的编辑自由



扫一扫，关注公众号

# 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：	1. 当不指定串口连接时，自动连接

	<p><b>port/dev_port:</b> 端口，默认是 <code>None</code>，会自动选用第一串口，<b>dev_port</b> 参数是为了兼容 <code>pyuf</code>，优先于 <b>port</b> 参数</p> <p><b>baudrate/ baud:</b> 波特率，默认是 <code>115200</code>，<b>baud</b> 参数是为了兼容 <code>pyuf</code>，优先于 <b>baudrate</b> 参数</p> <p><b>filters:</b> 串口过滤，字典，如 <code>{'hwid': 'USB VID:PID=2341:0042'}</code></p> <p><b>cmd_pend_size:</b> 命令缓存最大个数，默认为 <code>5</code></p> <p><b>callback_thread_pool_size:</b> 处理回调用到的线程池大小，默认为 <code>0</code>，即不适用线程处理回调</p> <p><b>do_not_open:</b> 是否不自动连接，默认是 <code>False</code>，即在实例初始化时自动尝试连接一遍，如果设为 <code>True</code>，则需要另行调用 <code>connect</code> 接口</p>	<p>的串口不一定是串口号最小的</p> <p>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</p>
<code>connect(...)</code>	<p>功能：连接串口，实例在初始化时已经连接(除非指定了 <b>do_not_open</b> 参数)</p> <p>参数：</p> <p><b>port:</b> 端口，默认是 <code>None</code>，会连接实例初始化时设定的端口</p> <p><b>baudrate:</b> 波特率，默认是 <code>None</code>，会使用实例初始化设定的波特率</p> <p><b>timeout:</b> 串口的读超时时间，默认是 <code>None</code>，会使用实例初始化设定的 <b>timeout</b></p>	
<code>disconnect(...)</code>	<p>功能：断开连接</p> <p>参数：</p> <p><b>is_clean:</b> 是否清除线程池，默认是 <code>True</code></p>	
<code>waiting_ready(...)</code>	<p>功能：等待机械臂准备好</p> <p>参数：</p> <p><b>timeout:</b> 等待的超时时间，默认是 <code>2s</code></p>	
<code>send_cmd_sync(...)</code>	<p>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</p> <p>参数：</p> <p><b>msg:</b> 命令，字符串，默认为 <code>None</code></p> <p><b>timeout:</b> 超时时间，默认为 <code>None</code>，使用实例默认的 <code>2s</code></p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 执行结果</li> <li>2. <code>'TIMEOUT'</code></li> </ol>	
<code>send_cmd_async(...)</code>	<p>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</p> <p>参数：</p> <p><b>msg:</b> 命令，字符串</p>	

	<p><b>timeout:</b> 超时时间，默认为 <b>None</b>，使用实例默认的 2s</p> <p><b>callback:</b> 回调函数</p>	
get_power_status	<p>功能：获取电源状态</p> <p>参数：</p> <p><b>wait:</b> 是否等待结果返回，默认 <b>True</b></p> <p><b>timeout:</b> 超时时间，默认为 <b>None</b>，使用实例默认的</p> <p><b>callback:</b> 回调函数</p> <p>返回值：</p> <p>1. <b>True/False</b></p>	
get_device_info(...)	<p>功能：获取设备信息</p> <p>参数：</p> <p><b>timeout:</b> 超时时间，默认为 <b>None</b>，使用 10s</p> <p>返回值：{</p> <pre> "device_type": "", "hardware_version": "", "firmware_version": "", "api_version": "", "device_unique": "" </pre> <p>}</p>	
reset(...)	<p>功能：复位</p> <p>参数：</p> <p><b>speed:</b> 速度，默认为 <b>None</b>，使用最后一次运动指定的速度或者初始的 1000</p> <p><b>wait:</b> 是否等待，默认为 <b>True</b></p> <p><b>timeout:</b> 超时时间，默认为 <b>None</b>，使用实例默认的</p>	
get_mode(...)	<p>功能：获取模式</p> <p>参数：</p> <p><b>wait:</b> 是否等待结果返回，默认 <b>True</b></p> <p><b>timeout:</b> 超时时间，默认为 <b>None</b>，使用实例默认的</p> <p><b>callback:</b> 回调函数</p> <p>返回值：<b>wait</b> 为 <b>True</b> 有效</p> <p>1. <b>mode</b>, 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</p> <p>2. <b>'TIMEOUT'</b></p>	仅仅支持 <b>SwiftPro</b>
set_mode(...)	<p>功能：设置模式</p> <p>参数：</p> <p><b>mode:</b> 要设置的模式（0, 1, 2, 3），默认 0</p> <p><b>wait:</b> 是否等待结果返回，默认 <b>True</b></p> <p><b>timeout:</b> 超时时间，默认为 <b>None</b>，使用实</p>	仅仅支持 <b>SwiftPro</b>

	<p>例默认的</p> <p>callback: 回调函数</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>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</li> <li>2. 'TIMEOUT'</li> </ol>	
get_position(...)	<p>功能: 获取当前坐标</p> <p>参数:</p> <p>wait: 是否等待结果返回, 默认 True</p> <p>timeout: 超时时间, 默认为 None, 使用实例默认的</p> <p>callback: 回调函数</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. [x, y, z]</li> <li>2. 'TIMEOUT'</li> </ol>	
set_position(...)	<p>功能: 设置移动坐标</p> <p>参数:</p> <p>x: X 坐标, 默认 None, 使用最后一次移动的 X 值或初始的 150</p> <p>y: Y 坐标, 默认 None, 使用最后一次移动的 Y 值或初始的 0</p> <p>z: Z 坐标, 默认 None, 使用最后一次移动的 Z 值或初始的 150</p> <p>speed: 速度, 默认 None, 使用最后一次移动的 speed 值或初始的 1000</p> <p>relative: 是否是相对移动, 默认 False</p> <p>wait: 是否等待结果返回, 默认 False</p> <p>timeout: 超时时间, 默认为 10s</p> <p>callback: 回调函数</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
get_polar(...)	<p>功能: 获取当前极坐标</p> <p>参数:</p> <p>wait: 是否等待结果返回, 默认 True</p> <p>timeout: 超时时间, 默认为 None, 使用实例默认的</p> <p>callback: 回调函数</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. [stretch, rotation, height]</li> <li>2. 'TIMEOUT'</li> </ol>	
set_polar(...)	<p>功能: 设置移动极坐标</p>	

	<p>参数:</p> <p>stretch/s: 长度, 默认为 None, s 参数是为了兼容 pyuf, 单位 mm</p> <p>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</p> <p>height/h: 高度, 默认为 None, h 参数是为了兼容 pyuf, 单位 mm</p> <p>speed: 速度, 默认 None, 使用最后一次移动的 speed 值或初始的 1000</p> <p>relative: 是否是相对移动, 默认 False</p> <p>wait: 是否等待结果返回, 默认 False</p> <p>timeout: 超时时间, 默认为 10s</p> <p>callback: 回调函数, 默认为 None</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
get_servo_angle(...)	<p>功能: 获取电机角度</p> <p>参数:</p> <p>servo_id: 电机号 (0: BOTTOM, 1: LEFT, 2: RIGHT, 3: HAND), 默认为 None, 返回所有的</p> <p>wait: 是否等待结果返回, 默认 True</p> <p>timeout: 超时时间, 默认为 None, 使用实例默认的 2s</p> <p>callback: 回调函数, 默认为 None</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>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</li> <li>2. 'TIMEOUT'</li> </ol>	
set_servo_angle(...)	<p>功能: 设置电机角度</p> <p>参数:</p> <p>servo_id: 电机号, 默认 0</p> <p>angle: 电机角度, 默认 90</p> <p>speed: 速度, 默认 None, 使用最后一次移动的 speed 值或初始的 1000</p> <p>wait: 是否等待结果返回, 默认 False</p> <p>timeout: 超时时间, 默认为 10s</p> <p>callback: 回调函数, 默认为 None</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
set_wrist(...)	<p>功能: 设置末端电机角度</p> <p>参数:</p>	

	<p>angle: 电机角度, 默认 90</p> <p>speed: 速度, 默认 None, 使用最后一次移动的 speed 值或初始的 1000</p> <p>wait: 是否等待结果返回, 默认 False</p> <p>timeout: 超时时间, 默认为 10s</p> <p>callback: 回调函数, 默认为 None</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
get_servo_attach(...)	<p>功能: 获取电机是否锁住</p> <p>参数:</p> <p>servo_id: 电机号, 默认 0</p> <p>wait: 是否等待结果返回, 默认 True</p> <p>timeout: 超时时间, 默认为 None, 使用实例默认的 2s</p> <p>callback: 回调函数, 默认为 None</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. True/False</li> <li>2. 'TIMEOUT'</li> </ol>	
set_servo_attach(...)	<p>功能: 锁定电机</p> <p>参数:</p> <p>servo_id: 电机号, 默认 None, 即全部电机</p> <p>wait: 是否等待结果返回, 默认 True</p> <p>timeout: 超时时间, 默认为 None, 使用实例默认的 2s</p> <p>callback: 回调函数, 默认为 None</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
set_servo_detach(...)	<p>功能: 解锁电机</p> <p>参数:</p> <p>servo_id: 电机号, 默认 None, 即全部电机</p> <p>wait: 是否等待结果返回, 默认 True</p> <p>timeout: 超时时间, 默认为 None, 使用实例默认的 2s</p> <p>callback: 回调函数, 默认为 None</p> <p>返回值:</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
set_buzzer(...)	<p>功能: 蜂鸣器控制</p> <p>参数:</p>	

	<p>frequency/freq：频率，默认 None，为 1000HZ，freq 参数是为了兼容 pyuf</p> <p>duration: time：时间，默认 None，为 2s，time 参数是为了兼容 pyuf</p> <p>wait：是否等待结果返回，默认 False</p> <p>timeout: 超时时间，默认为 None，使用实例默认的 2s</p> <p>callback: 回调函数，默认为 None</p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
set_pump(...)	<p>功能：吸头控制</p> <p>参数：</p> <p>on：是否打开，True 打开，False 关闭，默认 False</p> <p>wait：是否等待结果返回，默认 True</p> <p>timeout: 超时时间，默认为 None，使用实例默认的 2s</p> <p>callback: 回调函数，默认为 None</p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
set_gripper(...)	<p>功能：夹子控制</p> <p>参数：</p> <p>catch: True 为合上，False 为打开，默认 False</p> <p>wait：是否等待结果返回，默认 True</p> <p>timeout: 超时时间，默认为 None，使用实例默认的 2s</p> <p>callback: 回调函数，默认为 None</p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> <li>3. Other, 参照开发手册里的报错表</li> </ol>	
get_analog(...)	<p>功能：获取指定引脚的模拟电平</p> <p>参数：</p> <p>pin：引脚，默认 0</p> <p>wait：是否等待结果返回，默认 True</p> <p>timeout: 超时时间，默认为 None，使用实例默认的 2s</p> <p>callback: 回调函数，默认为 None</p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 模拟电平</li> </ol>	



	2. 'TIMEOUT'	
get_digital()	<p>功能：获取指定引脚的数字电平</p> <p>参数：</p> <ul style="list-style-type: none"> <li>pin: 引脚，默认 0</li> <li>wait: 是否等待结果返回，默认 True</li> <li>timeout: 超时时间，默认为 None，使用实例默认的 2s</li> <li>callback: 回调函数，默认为 None</li> </ul> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 数字电平</li> <li>2. 'TIMEOUT'</li> </ol>	
get_limit_switch(...)	<p>功能：获取吸头是否限位</p> <p>参数：</p> <ul style="list-style-type: none"> <li>wait: 是否等待结果返回，默认 True</li> <li>timeout: 超时时间，默认为 None，使用实例默认的 2s</li> <li>callback: 回调函数，默认为 None</li> </ul> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. True/False</li> <li>2. 'TIMEOUT'</li> </ol>	
get_gripper_catch(...)	<p>功能：获取夹子状态</p> <p>参数：</p> <ul style="list-style-type: none"> <li>wait: 是否等待结果返回，默认 True</li> <li>timeout: 超时时间，默认为 None，使用实例默认的 2s</li> <li>callback: 回调函数，默认为 None</li> </ul> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 0: stop, 1: working, 3: catch thing</li> <li>2. 'TIMEOUT'</li> </ol>	
get_pump_status(...)	<p>功能：获取吸泵状态</p> <p>参数：</p> <ul style="list-style-type: none"> <li>wait: 是否等待结果返回，默认 True</li> <li>timeout: 超时时间，默认为 None，使用实例默认的 2s</li> <li>callback: 回调函数，默认为 None</li> </ul> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 0: stop, 1: working, 3: catch thing</li> </ol> <p>'TIMEOUT'</p>	
grove_init(...)	<p>功能：初始化 grove，在 Port {pin} 初始化模块 {grove_type}</p> <p>参数：</p> <ul style="list-style-type: none"> <li>pin: pin/port，默认为 None，必须传</li> <li>grove_type: 模块，必须传</li> <li>value: 值，根据不同模块决定是否传</li> </ul>	M2305 P{} N{} V{}

	<p><b>wait:</b> 是否等待结果返回，默认 True</p> <p><b>timeout:</b> 超时时间，默认为 None，使用实例默认的 2s</p> <p><b>callback:</b> 回调函数，默认为 None</p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> </ol>	
grove_control(...)	<p>功能：grove 控制</p> <p>参数：</p> <p><b>pin:</b> pin/port，默认 None，必须传</p> <p><b>value:</b> 值</p> <p><b>wait:</b> 是否等待结果返回，默认 True</p> <p><b>timeout:</b> 超时时间，默认为 None，使用实例默认的 2s</p> <p><b>callback:</b> 回调函数，默认为 None</p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> </ol>	
register_grove_callback(...)	<p>功能：注册 grove 事件的回调函数</p> <p>参数：</p> <p><b>pin:</b> pin/port，默认 None，必须传</p> <p><b>grove_type:</b> 模块，默认 None，必须传</p> <p><b>callback:</b> 回调函数，默认为 None</p>	
set_report_grove(...)	<p>功能：开启 grove 自动上报</p> <p>参数：</p> <p><b>pin:</b> pin/port，默认 None，必须传</p> <p><b>interval:</b> 上报间隔，默认 0.5，0 表示停止上报</p> <p><b>callback:</b> 回调函数，默认为 None</p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> </ol>	
register_power_callback(...)	<p>功能：注册电源事件回调函数</p> <p>参数：</p> <p><b>callback:</b> 回调函数，默认为 None</p>	
register_report_position_callback(...)	<p>功能：注册位置上报事件回调函数</p> <p>参数：</p> <p><b>callback:</b> 回调函数，默认为 None</p>	
set_report_position(...)	<p>功能：设置位置上报</p> <p>参数：</p> <p><b>Interval:</b> 上报间隔，默认 1s，0 表示停止上报</p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> </ol>	仅仅支持 SwiftPro

	2. 'TIMEOUT'	
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'	
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'	
set_fans(...)	功能：风扇控制（仅在 3D 打印模式有效，此接口会自动切换到 3D 打印模式） 参数： on: True/False，默认 False，关闭风扇 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 2s callback: 回调函数，默认为 None 返回值： 1. 'OK' 2. 'TIMEOUT'	
set_temperature(...)	功能：设置温度（仅在 3D 打印模式有效，此接口会自动切换到 3D 打印模式） 参数： Temperature: 目标温度，默认 0 Block: 默认 False，如果设为 True，会自动	

	<p>上报温度，但会阻塞整个系统，无法响应其它命令知道达到目标温度以后才响应</p> <p><b>wait:</b> 是否等待结果返回，默认 <b>True</b>，这里的结果只是命令的执行结果，而不是达到目标温度的结果</p> <p><b>timeout:</b> 超时时间，默认为 <b>None</b>，使用实例默认的 2s</p> <p><b>callback:</b> 回调函数，默认为 <b>None</b></p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> </ol>	
get_temperature(...)	<p>功能：获取温度</p> <p>返回值：</p> <pre>{     "current_temperature": 0.     "target_temperature": 0, }</pre>	
set_3d_feeding(...)	<p>功能：设置进料系统的进料（此接口需要在 3D 打印模式下使用，请自行切换模式，另外需要温度在 170 摄氏度才有效，如果要进行 3D 打印，一般来说，目标温度需要 200 度左右）</p> <p>参数：</p> <p><b>distance:</b> 进料距离，默认 0，大于 0 表示进料，小于 0 表示退料</p> <p><b>speed:</b> 速度，默认 100</p> <p><b>relative:</b> 是否是相对距离，默认 <b>True</b>，如果设为 <b>True</b>，则需要自行累加进料距离</p> <p><b>x:</b> 在进料过程要移动的 X 坐标，默认 <b>None</b>，即不移动 X 坐标</p> <p><b>y:</b> 在进料过程要移动的 Y 坐标，默认 <b>None</b>，即不移动 Y 坐标</p> <p><b>z:</b> 在进料过程要移动的 Z 坐标，默认 <b>None</b>，即不移动 Z 坐标</p> <p><b>wait:</b> 是否等待结果返回，默认 <b>True</b></p> <p><b>timeout:</b> 超时时间，默认为 30</p> <p><b>callback:</b> 回调函数，默认为 <b>None</b></p> <p>返回值：</p> <ol style="list-style-type: none"> <li>1. 'OK'</li> <li>2. 'TIMEOUT'</li> </ol>	
set_acceleration(...)	<p>功能：加速度设置（一般不要设置）</p> <p>参数：</p> <p><b>printing_moves:</b> Printing moves，默认 <b>None</b></p> <p><b>retract_moves :</b> Retract only (no X, Y, Z)</p>	

	<p>moves, 默认 True</p> <p>travel_moves: Travel (non printing) moves, 默认为 None</p> <p>min_feedrate: Min Feed Rate (units/s), 默认为 None</p> <p>min_travel_feedrate: Min Travel Feed Rate (units/s), 默认为 None</p> <p>min_segment_time: Min Segment Time (us), 默认为 None</p> <p>max_xy_jerk: Max XY Jerk (units/sec^2), 默认为 None</p> <p>max_z_jerk: Max Z Jerk (units/sec^2), 默认为 None</p> <p>min_e_jerk: Max E Jerk (unit/sec^2), 默认为 None</p> <p>返回值:</p> <p>1. 'OK'</p>	