# uArm Python SDK API

**SwiftAPI class property**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Function** | **Notes** |
| connected | Connect Status |  |
| port | Port |  |
| baudrate | Baud Rate |  |
| power\_status | Power Status | Firmware 4.0 or later are not supported yet. |
| device\_type | Device Type | Initial value: None.  Not available before get device info. |
| hardware\_version | Hardware Version | Initial value: None.  Not available before get device info. |
| firmware\_version | Firmware Version | Initial value: None.  Not available before get device info. |
|  |  |  |
|  |  |  |

**SwiftAPI class interface**

**Note：**

1. **Please use keywords as Parameter to call functions to avoid changing the order of interface Parameter.**
2. **Parameter order in this document isn’t necessary nature order, please use keywords to send parameter.**
3. **If callback was pointed while calling function, please note that callback should not be blocked, or it may block some Functions.**
4. **Return will be valid when wait parameter is True, or Return will be callback parameter.**
5. **Callback Parameter is return value, return value could be “executive result” or “TIMEOUT”.**
6. **uArm-Python SDK is partly compatible with pyuf, if you want to run your pyuf based script on this SDK, please change the import part of your script.**
7. **If you want to run your firmware 3.2.0 (or earlier) based script on firmware 4.0 or later with this SDK, please reset the speed by “set\_speed\_factor”, for instance, set\_speed\_factor(0.0005).**

|  |  |  |
| --- | --- | --- |
| **API** | **Function and Parameter** | **Notes:** |
| SwiftAPI(…) | Function：example initialization  Parameter：  **port/dev\_port**: port, default value: None, choose the first port by automatically, dev\_portParameter is design for compatible with pyuf, take precedence over than portParameter  **baudrate/baud**: baud rate, default value: 115200, baudParameter is design for compatible with pyuf, take precedence than baudrate Parameter  **filters**: COM port filter, dictionary, such as {‘hwid’: ‘USB VID:PID=2341:0042’}, default value: None without filter  **cmd\_pend\_size**: maximum of command catches, default value:2  **cmd\_timeout**: default value command timeout duration, default value:2  **callback\_thread\_pool\_size**: the size of the thread pool used to handle callback. default value:0, means give up using thread pool. If value is 1, means use asyncio to handle the callback, if value more than 2, use the thread pool to handle the callback.  **do\_not\_open**: do/don’t connect automatically, default value: False, try to connect automatically while initializing the example, if set True, you have to call connect function separately.  **enable\_handle\_thread**: do/don’t use thread to handle result. True/False, default is True.  **enable\_write\_thread**: do/don’t use thread to handle sending. True/False, default is False.  **enable\_handle\_report\_thread**: do/don’t use thread to handle the reporting, True/False, default is False. | 1. If you do not assign a special COM port, the COM port automatically connected may not be the port has minimum port ID. 2. If you build multiple example to connect multiple COM port in one process without assigning special port for every example, it will automatically connect different COM port. |
| connect(…) | Function：connect the COM port, already done on the example initialization (unless assigned do\_not\_open Parameter)  Fuction: Connect the port,  Parameter：  **port**: port, default value: None, will connect the COM port set by the example initialization.  **baudrate**: baud rate, default value: None, using the baud rate set on example initialization. timeout: COM port read timeout duration, default value: None, use the timeout set on example initialization. |  |
| disconnect(…) | Function：disconnect the uArm  Parameter：  **is\_clean**: clean the threads pool, default value: True |  |
| waiting\_ready(…) | Function：wait until the uArm is ready  Parameter：  **timeout**： timeout duration, default value: 5s |  |
| send\_cmd\_sync(…) | Function：send synchronized command, send command until return execute result or timeout.  Parameter：  msg: command, string, default value: None  timeout: timeout duration, default value: None, using the example default value.  no\_cnt: do not add cnt prefix or not, default is False  Return：   1. Execute result. 2. ‘TIMEOUT’ | If you want to send the msg directly(do not add the cnt prefix), you can set the no\_cnt is True. |
| send\_cmd\_async(…) | function：send asynchronous command. Send command without waiting for return, but you could assign call back.  Parameter：  **msg**: command, string  **timeout**: timeout duration, default: None, example default value.  **callback**: callback function |  |
| get\_power\_status(…) | function：get the status of the power supply Parameter：  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use example default value.  **callback**：callback function  **Return**：   1. True/False |  |
| get\_device\_info(…) | function：get device information  Parameter：  timeout: timeout duration, default value: None, use 10s  Return:{  “device\_type”: “SwiftPro”,  “hardware\_version”: “3.2.0”,  “firmware\_version”: “3.3.0”,  “api\_version”: “3.2.0”,  “device\_unique”: “xxxxxxxxxx”  } |  |
| set\_speed\_factor(…) | function：set speed factor（design for firmware 4.0 or later to compatible the speed of the firmware 4.0 or later）, all the input speed will multiply by this factor.  Parameter：  **factor**: default value:1, same with the original speed value. |  |
| reset(…) | Function：reset  Parameter：  **speed**: default value: None, use the speed of last movement or initial value  **wait**: do/don’t wait, default value: True  **timeout**: timeout duration, default value: None, example default value. |  |
| get\_mode(…) | Function：get the mode of the uArm  Parameter：  **wait**：wait for return(true/false), default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function  Return：take effect while wait is True   1. mode, 0 normal mode; 1 Laser mode; 2 3D printing mode;3 drawing mode | uArm Swift Pro only |
| set\_mode(…) | Function：set mode  Parameter：  **mode**：set mode of uArm（0, 1, 2, 3） default value:0  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value.  **callback**：callback function  Return：   1. mode, 0 normal mode; 1 Laser mode; 2 3D printing mode;3 drawing mode | Only support uArm Swift Pro |
| get\_position(…) | Function: get current coordinate.  Parameter：  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value.  **callback**：callback function  Return：   1. [x, y, z] 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| set\_position(...) | Function：set target position for uArm  Parameter：  **x**：X axis value, default value: None, use the X value of last movement or initial value 150.  **y**：Y axis value, default value: None, use the Y value of last movement or initial value 0.  **z**：Z axis value, default value: None, use the Z value of last movement or initial value 150.  **speed**：speed value, default value: None, use the speed value of last movement for default speed 1000.  **relative**：is/isn’t relative movement, default value: False  **wait**：do/don’t wait for return, default value: False  **timeout**: timeout duration, default value:10s  **callback**：callback function  **cmd**: use ‘G0’ or ’G1’, default value:’G0’  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| get\_polar(…) | Function：get current polar coordinate.  Parameter：  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function  Return：   1. [stretch, rotation, height] 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| set\_polar(…) | Function：set target polar coordinate  Parameter：  **stretch**/s：stretch length, default value: None, sParameteris design for compatible with pyuf, unit：mm  **rotation**/r：rotation anger, default value: None, rParameteris design for compatible with pyuf, unit：degree, 0-180°  **height**/h：hight, default value: None, hParameter is design for compatible with pyuf, unit: mm  **speed**：speed, default value: None, use the speed of last movement or initial value 1000  **relative**：relative movement or not, default value: False  **wait**：do/don’t wait for return, default value: False  **timeout**: timeout duration, default value:10s  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| get\_servo\_angle(…) | Function：get the angle of the motor  Parameter：  **servo\_id**：motor ID (0: BOTTOM, 1: LEFT, 2: RIGHT), default value: None, return all motor angle value.  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. If motor ID was assigned, it will return angle of the motor, else return motor angle list. 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| set\_servo\_angle(…) | Function：set angle of the motor Parameter：  **servo\_id**：motor ID (0: BOTTOM, 1: LEFT, 2: RIGHT, 3: HAND), default value:0  **angle**：angle of the motor, default value:90  **speed**: speed, default value: None, use the speed of last movement or initial value 1000  **wait**：do/don’t wait for return, default value: False  **timeout**: timeout duration, default value:10s  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| set\_wrist(…) | Function：set the angle of the 4th motor Parameter：  **angle**：angle of motor, default value:90  **speed**: speed value, default value: None, use the speed of last movement or initial value 1000  **wait**：do/don’t wait for return, default value: False  **timeout**: timeout duration, default value:10s  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| get\_servo\_attach(…) | Function：check whether the motor is attached Parameter：  **servo\_id**：motor ID (0: BOTTOM, 1: LEFT, 2: RIGHT, 3: HAND), default value:0  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. True/False 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| set\_servo\_attach(…) | Function：attach the motor  Parameter：  **servo\_id**：motor ID, default value: None, means all motors. **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| set\_servo\_detach(…) | Function：detach the motor.  Parameter：  **servo\_id**：motor ID, default value: None, all motors  **wait**：do/don’t wait for return, default value: True  timeout: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| set\_buzzer(…) | Function：control the buzzer.  Parameter：  **frequency/freq**：frequency, default value: None, 1000HZ, frequency Parameter is design for compatible with pyuf.  **duration**: time：time, default value: None, 为2s, time Parameter is design for compatible with pyuf  **wait**：do/don’t wait for return, default value: False  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| set\_pump(…) | Function：control the suction cup  Parameter：  **on**：do/don’t open, True: open, False: close, default value: False  **wait**：do/don’t wait for return, default value: True  **check**: check the pump status and wait or not if wait is True, default is False  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| set\_gripper(…) | Function：control the gripper  Parameter：  **catch**：True, catch, False, release, default value: False  **wait**：do/don’t wait for return, default value: True  **check**: check the catch status and wait or not if wait is True, default is False  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of “uArm Swift Pro developer guide” |  |
| get\_analog(…) | Function：get assigned pin analog level Parameter：  **pin**：pin, default value:0  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. analog level 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| get\_digital() | Function：get assigned pin’s digital level Parameter：  **pin**：pin, default value:0  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. Digital level 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| get\_limit\_switch(…) | Function：get the status of the limit switch Parameter：  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：call back function, default value: None  Return：   1. True/False 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| get\_gripper\_catch(…) | Function：get the status of the gripper  Parameter：  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. 0: stop, 1: working, 3: catch thing 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| get\_pump\_status(…) | Function：get the status of the pump  Parameter：  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. 0: stop, 1: working, 3: catch thing 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| grove\_init(…) | Function：initialize Grove, initialize {Grove\_type} on Port ｛pin｝  Parameter：  **pin**：pin/port, default value: None, must send.  **Grove\_type**：module, must send  **value**：value, do/don’t send depends on different module.  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” | M2305 P{} N{} V{} |
| grove\_control(…) | Function：Grove control  Parameter：  **pin**：pin/port, default value: None, must send.  **value**：value  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| register\_grove\_callback(…) | Function：Call back function of register Grove Parameter：  **pin**：pin/port, default value: None, must send  **callback**：callback function, default value: None |  |
| release\_grove\_callback(…) | Function：clear Grove events, callback function  Parameter：  **pin**：pin/port, default value: None, assign the callback for the pin to clear指定要清除的pin的回调  **callback**： registered Grove event callback function of assigned pin, default value: None, erase all callback of registered Grove event of assigned pin. |  |
| set\_report\_grove(…) | Function：Enable Grove auto feedback  Parameter：  **pin**：pin/port, default value: None, must send  **interval**：report interval, default value:0.5, 0 means stop report.  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| register\_power\_callback(…) | Function：register power event callback function  Parameter：  **callback**：callback function, default value: None |  |
| release\_power\_callback(…) | **Function**：erase power events callback function  Parameter：  **callback**：callback function of previous registered power event. default value: None, erase all callback function of registered power events. |  |
| register\_report\_position\_callback(…) | Function：register position report event callback function  Parameter：  **callback**：callback function, default value: None |  |
| release\_report\_position\_callback(…) | Function：erase position report event callback function  Parameter：  **callback**：callback function of previous registered position report event, default value: None, erase all callback of registered position report event. |  |
| set\_report\_position(…) | Function：set position report.  Parameter：  **Interval**：report interval, default value:1s, 0 means stop reporting.  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| register\_key0\_callback(..) | Function：callback function of register button 0（BUTTON\_MENU）event  Parameter：  **callback**：callback function, default value: None |  |
| release\_key0\_callback(..) | Function：erase the callback function of button 0（BUTTON\_MENU） event  Parameter：  **callback**：callback function of previous registered button 0（BUTTON\_MENU）event, default value: None, erase all callback function of button 0（BUTTON\_MENU） |  |
| register\_key1\_callback(..) | Function：callback function of register button 1（BUTTON\_PLAY）  Parameter：  **callback**：callback function, default value: None |  |
| release\_key1\_callback(..) | Function：erase callback function of button 1（BUTTON\_PLAY）event  Parameter：  **callback**：callback function of previous registered button 1（BUTTON\_PLAY）event, default value: None, erase all callback function of button 1（BUTTON\_PLAY） |  |
| set\_report\_keys(…) | Function：set keys repot Parameter：  **on/is\_on**: True/False, True turn on report, False stop report, default value: True, is\_on Parameter is design for compatible with pyuf  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” |  |
| register\_limit\_switch\_callback(…) | Function： register the callback function of limit switch.  Parameter：  **callback**：callback function, default value: None |  |
| release\_limit\_switch\_callback(…) | Function：erase the callback function of limit switch Parameter：  **callback**：callback function of previous registered limited switch event, default value: None, erase all callback function of registered limited switch event. |  |
| get\_is\_moving(…) | Function：check if the uArm is moving  Parameter：  **wait**：do/don’t wait feedback, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. True/False |  |
| flush\_cmd(…) | Function：wait all command catch return or time out Parameter：  **timeout**: duration of timeout; default value: None, no time out  **wait\_stop**: do/don’t wait the uArm finish the movement or timeout, default value: False  Return：   1. ‘OK’ 2. ‘TIMEOUT’ |  |
| set\_fans(…) | Function：control the fan（only available on 3D printing mode, it will automatically switch to 3D printing mode. ）  Parameter：  **on**：True/False, default value: False, turn off the fan.  **wait**：do/don’t wait for return, default value: True  **timeout**: timeout duration, default value: None, use default value  **callback**：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” | 1. Only support uArm SwiftPrro 2. Currently do not support firmware 4.0 or later. |
| set\_temperature(…) | Function：set target temperature（Only available on 3D printing function, uArm will switch to 3D printing mode automatically）  Parameter：  Temperature：Target temperature, default value:0  **Block**: default value: False, if set True, will report temperature automatically and will block the system from responding other commands until it reach the target temperature.  **wait**：do/don’t wait for return, default value: True, result here is result of execution instead of reach the target temperature.  **timeout**: timeout duration, default value: None, use default value  **callback**：call back function, default value: None  **Return**：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” | 1. Only support uArm SwiftPrro 2. Currently do not support firmware 4.0 or later. |
| get\_temperature(…) | Function：get temperature  Return：  {  “current\_temperature”: 0.  “target\_temperature”: 0,  } | 1. Only support uArm SwiftPrro 2. Currently do not support firmware 4.0 or later. |
| set\_3d\_feeding(…) | Function：set 3D printing feed motor. （Only available under 3D printing mode, please switch the work mode manually. Temperature should be more than 170℃, recommend temperature:200℃. Parameter：  **distance**：set the length of the PLA feed by feed motor, default value:0, distance>0: load PLA；distance<0, unload PLA.  **speed**：speed of feed motor, default value:100  relative：relative/absolute length of the PLA, default value: True. If set True, you have to calculate the relative length of PLA.  **x**：moving on X axis while loading PLA, default value: None, do not move on X axis.  **y**：moving on Y axis while loading PLA, default value: None, do not move on y axis.  **z**：moving on Z axis while loading PLA, default value: None, do not move on Z axis.  **wait**：wait for feedback, default value: True  **timeout**: timeout duration, default value:30  **callback**：callback function, default value: None  **Return**：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” | 1. Only support uArm SwiftPrro   2. Currently do not support firmware 4.0 or later. |
| set\_acceleration2(…) | Function：set acceleration（normally do not set that parameter）  Parameter：  printing\_moves：Printing moves, default value: None  retract\_moves：Retract only (no X, Y, Z) moves, default value: True  travel\_moves：Travel (no printing) moves, default value: None  min\_feedrate：Min Feed Rate (units/s), default value: None  min\_travel\_feedrate：Min Travel Feed Rate (units/s), default value: None  min\_segment\_time：Min Segment Time (us), default value: None  max\_xy\_jerk：Max XY Jerk (units/sec^2), default value: None  max\_z\_jerk：Max Z Jerk (units/sec^2), default value: None  min\_e\_jerk：Max E Jerk (unit/sec^2), default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” | Not support firmware 4.0 or later. |
| set\_acceleration(…) | Function：set the acceleration  Parameter：  acc：acceleration, default value:1.3  wait：do/don’t wait for feedback, default value: True  timeout: timeout duration, default value: None, use default value  callback：callback function, default value: None  Return：   1. ‘OK’ 2. ‘TIMEOUT’ 3. Other, refer to the Error of ”uArm Swift Pro Developer Guide” | Only support firmware 4.0 or later. |