

Dobot M1 Script

接口说明

文档版本: V1.4.3

发布日期: 2019-10-30

版权所有 © 越疆科技有限公司2019。 保留一切权利。

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

免责声明

在法律允许的最大范围内，本手册所描述的产品（含其硬件、软件、固件等）均“按照现状”提供，可能存在瑕疵、错误或故障，越疆不提供任何形式的明示或默示保证，包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证；亦不对使用本手册或使用本公司产品导致的任何特殊、附带、偶然或间接的损害进行赔偿。

在使用本产品前详细阅读本使用手册及网上发布的相关技术文档并了解相关信息，确保在充分了解机器人及其相关知识的前提下使用机械臂。越疆建议您在专业人员的指导下使用本手册。该手册所包含的所有安全方面的信息都不得视为Dobot的保证，即便遵循本手册及相关说明，使用过程中造成的危害或损失依然有可能发生。

本产品的使用者有责任确保遵循相关国家的切实可行的法律法规，确保在越疆机械臂的使用中不存在任何重大危险。

越疆科技有限公司

地址：深圳市南山区留仙大道3370号南山智园崇文区2号楼9-10楼

网址：<http://cn.dobot.cc/>

前言

目的

本文档旨在对 Dobot Script 接口进行详细说明，并给出基于 Dobot Script 接口开发应用程序的一般流程。

读者对象

本手册适用于：

- 客户工程师
- 安装调试工程师
- 技术支持工程师

修订记录

时间	修订记录
2019/10/30	第一次发布

符号约定

在本手册中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害
 警告	表示有中度或低度潜在危害，如果不能避免，可能导致人员轻微伤害、机械臂毁坏等情况
 注意	表示有潜在风险，如果忽视这些文本，可能导致机械臂损坏、数据丢失或不可预知的结果
 说明	表示是正文的附加信息，是对正文的强调和补充

目 录

1. 简介.....	1
2. Script 接口说明.....	2
2.1 Dobot 指令简介	2
2.2 PTP 功能.....	2
2.2.1 设置 JUMP 模式下抬升高度和最大抬升高度.....	5
2.2.2 同步模式下设置 JUMP 模式下抬升高度和最大抬升高度.....	5
2.2.3 获取 JUMP 模式下抬升高度和最大抬升高度.....	6
2.2.4 设置 PTP 运动时速度百分比和加速度百分比	6
2.2.5 同步模式下设置 PTP 运动时速度百分比和加速度百分比.....	6
2.2.6 获取 PTP 运动的速度百分比和加速度百分比	7
2.2.7 执行 PTP 指令	7
2.2.8 同步模式下执行 PTP 指令	8
2.2.9 执行 ARC 指令.....	8
2.2.10 同步模式下执行 ARC 指令	9
2.2.11 执行 CIRCLE 指令	9
2.2.12 同步模式下执行 CIRCLE 指令	10
2.2.13 设置机械臂方向	10
2.2.14 同步模式下设置机械臂方向	10
2.2.15 获取机械臂实时位姿	11
2.3 WAIT 功能	11
2.3.1 执行时间等待指令	11
2.3.2 同步模式下执行时间等待指令	11
2.3.3 延时指令	12
2.4 触发功能	12
2.4.1 执行触发指令	12
2.4.2 同步模式下执行触发指令	13
2.5 I/O 功能.....	13
2.5.1 设置数字输出电平	13
2.5.2 同步模式下设置数字输出电平	14
2.5.3 读取数字输出电平	14
2.5.4 读取数字输入电平	14
2.5.5 读取连续数字输入地址的电平	15
2.5.6 读取模拟输入值	15
2.6 其他	15
2.6.1 加载 Playback 数据	15
2.6.2 获取 Playback 数据	15
2.6.1 设置码垛名称	16
2.6.2 获取准备点数据	16
2.6.3 获取过渡点数据	17
2.6.4 获取码垛点数据	17

1. 简介

用户可通过脚本控制机械臂的运行，Dobot M1提供丰富的Script接口，如速度/加速度设置、运动模式设置以及I/O配置等，采用Python脚本语言开发，可供用户二次开发时调用。本文档详细描述Script接口。

2. Script 接口说明

2.1 Dobot 指令简介

控制器支持两种类型的指令：立即指令与队列指令。

- 立即指令：Dobot控制器在收到指令后立即处理该指令，而不管当前控制器是否在还在处理其他指令。
- 队列指令：Dobot控制器在收到指令后会该指令放入控制器内部的指令队列中，Dobot控制器将顺序执行指令。

关于Dobot指令更具体的内容，请参见《Dobot通信协议》。

说明

Script接口涉及到同步模式和非同步模式，详细说明如下：

- 同步模式即上位机下发同步指令后，会等待Dobot控制器执行完当前同步指令后，上位机再下发下一条指令。
- 非同步模式即下发队列指令或立即指令。上位机下发指令后，不会等待Dobot控制器执行完当前指令，立即下发下一条指令。

2.2 PTP 功能

PTP点位模式即实现点到点运动，Dobot M1的点位模式包括MOVJ、MOVL以及JUMP三种运动模式。不同的运动模式，示教后存点回放的运动轨迹不同。

- MOVJ：关节运动，由A点运动到B点，各个关节从A点对应的关节角运行至B点对应的关节角。关节运动过程中，各个关节轴的运行时间需一致，且同时到达终点，如图 2.1所示。

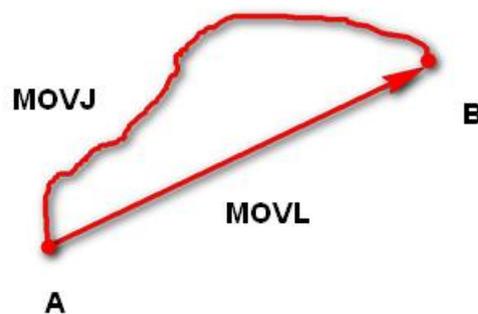


图 2.1 MOVL 和 MOVJ 运动模式

- MOVL：直线运动，A点到B点的路径为直线，如图 2.1所示。
- JUMP：门型轨迹，A点到B点以MOVJ运动模式移动，如图 2.2所示。
 1. 以MOVJ运动模式上升到一定高度（Height）。
 2. 以MOVJ运动模式过渡到最大抬升高度（Limit）。
 3. 以MOVJ运动模式平移到B点上方的高度处。
 4. 以MOVJ运动模式过渡到B点高度加上Height后的高度处。

5. 以MOVJ运动模式下降到B点所在位置。

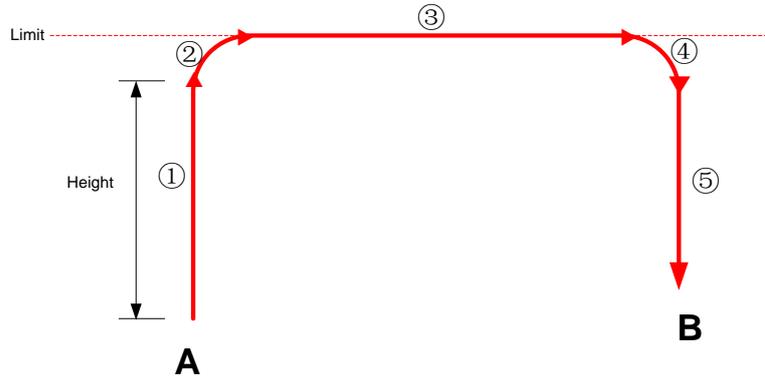
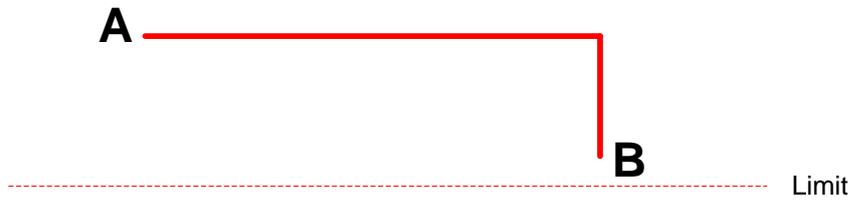


图 2.2 JUMP 运动模式

JUMP运动模式下，如果起始点或结束点高度大于等于最大抬升高度，或者起始点抬升到一定高度后，大于等于最大抬升高度，其运动轨迹与图 2.2有所不同。假设A为起始点，B为结束点，Limit为最大抬升高度，Height为抬升高度。

- A点、B点高度均大于Limit，且A点高度大于B点高度。



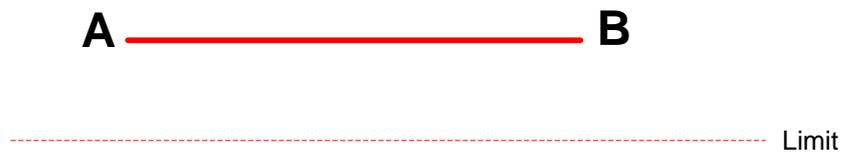
- A点、B点高度均大于Limit，且A点高度小于B点高度。



- A点高度大于Limit，B点高度小于Limit。



- A点、B点高度相同，且大于Limit。



- A点高度小于Limit，B点高度大于Limit。



- A点、B点高度与Limit相同。



- A点、B点高度均小于Limit，A点高度+Height和B点+Height大于Limit。



注意

使用脚本编程时，如果在笛卡尔坐标系使用对应的运动指令，则需在每一条运动指令前加上方向指令，表示机械臂的运动方向。

2.2.1 设置 JUMP 模式下抬升高度和最大抬升高度

表 2.1 设置 JUMP 模式下抬升高度和最大抬升高度

原型	dType.SetPTPJumpParams(api, jumpHeight, zLimit, isQueued)
功能	设置 JUMP 模式下抬升高度和最大抬升高度
参数	api: Dobot 库对象，不能修改 jumpHeight: 抬升高度 zLimit: 最大抬升高度 isQueued: 是否将该指令指定为队列命令。取值范围：0，立即指令；1，队列指令
返回	如果设置为立即指令，则返回 0 如果设置为队列指令，则返回该指令在队列的索引号
示例	dType.SetPTPJumpParams(api, 50, 100, 0)

2.2.2 同步模式下设置 JUMP 模式下抬升高度和最大抬升高度

表 2.2 同步模式下设置 JUMP 模式下抬升高度和最大抬升高度

原型	dType.SetPTPJumpParamsSync(api, jumpHeight, zLimit)
功能	同步模式下设置 JUMP 模式下抬升高度和最大抬升高度
参数	api: Dobot 库对象，不能修改 jumpHeight: 抬升高度 zLimit: 最大抬升高度
返回	无
示例	无

2.2.3 获取 JUMP 模式下抬升高度和最大抬升高度

表 2.3 设置 JUMP 模式下抬升高度和最大抬升高度

原型	dType.GetPTPJumParams(api)
功能	获取 JUMP 模式下抬升高度和最大抬升高度
参数	api: Dobot 库对象, 不能修改
返回	jumpHieght (List[0]): 抬升高度 zLimit (List[1]): 最大抬升高度
示例	无

2.2.4 设置 PTP 运动时速度百分比和加速度百分比

表 2.4 设置 PTP 运动时速度百分比和加速度百分比

原型	dType.SetPTPCommonParams(api, velocityRatio, accelerationRatio, isQueued)
功能	设置 PTP 运动时速度百分比和加速度百分比
参数	api: Dobot 库对象, 不能修改 velocityRatio: PTP 模式速度百分比, 关节坐标轴和笛卡尔坐标轴共用 accelerationRatio: PTP 模式加速度百分比, 关节坐标轴和笛卡尔坐标轴共用 isQueued: 是否将该指令指定为队列命令。取值范围: 0, 立即指令; 1, 队列指令
返回	如果设置为立即指令, 则返回 0 如果设置为队列指令, 则返回该指令在队列的索引号
示例	dType.SetPTPCommonParams(api, 50, 50, 1)

2.2.5 同步模式下设置 PTP 运动时速度百分比和加速度百分比

表 2.5 同步模式下设置 PTP 运动时速度百分比和加速度百分比

原型	dType.SetPTPCommonParamsSync(api, velocityRatio, accelerationRatio)
功能	同步模式下设置 PTP 运动时速度百分比和加速度百分比
参数	api: Dobot 库对象, 不能修改 velocityRatio: PTP 模式速度百分比, 关节坐标轴和笛卡尔坐标轴共用 accelerationRatio: PTP 模式加速度百分比, 关节坐标轴和笛卡尔坐标轴共用
返回	无
示例	dType.SetPTPCommonParamsSync(api, 30, 30)

2.2.6 获取 PTP 运动的速度百分比和加速度百分比

表 2.6 设置 PTP 运动的速度百分比和加速度百分比

原型	dType.GetPTPCCommonParams(api)
功能	获取 PTP 运动的速度百分比和加速度百分比
参数	api: Dobot 库对象, 不能修改
返回	velocityRatio (List[0]): PTP 模式速度百分比, 关节坐标轴和笛卡尔坐标轴共用 accelerationRatio (List[1]): PTP 模式加速度百分比, 关节坐标轴和笛卡尔坐标轴共用
示例	无

2.2.7 执行 PTP 指令

表 2.7 PTP 运动的速度百分比和加速度百分比

原型	dType.SetPTPCmd(api, ptpMode, x, y, z, rHead, isqueued)
功能	执行 PTP 指令。设置 PTP 相关参数后, 调用此函数可使机械臂运动至设置的目标点
参数	<p>api: Dobot 库对象, 不能修改</p> <p>ptpMode: PTP 模式, 取值范围: 0~9</p> <ul style="list-style-type: none"> 0: JUMP 模式, 平移时运动模式为 MOVJ, (x,y,z,rHead) 为笛卡尔坐标系下的目标点坐标 1: MOVJ 模式, (x,y,z,rHead) 为笛卡尔坐标系下的目标点坐标 2: MOVL 模式, (x,y,z,rHead) 为笛卡尔坐标系下的目标点坐标 3: JUMP 模式, (x,y,z,rHead) 为关节坐标系下的目标点坐标 4: MOVJ 模式, (x,y,z,rHead) 为关节坐标系下的目标点坐标 5: MOVL 模式, (x,y,z,rHead) 为关节坐标系下的目标点坐标 6: MOVJ 模式, (x,y,z,rHead) 为关节坐标系下的坐标增量 7: MOVL 模式, (x,y,z,rHead) 为笛卡尔坐标系下的坐标增量 8: MOVJ 模式, (x,y,z,rHead) 为笛卡尔坐标系下的坐标增量 9: JUMP 模式, 平移时运动模式为 MOVL, (x,y,z,rHead) 为笛卡尔坐标系下的目标点坐标 <p>x, y, z, rHead: 坐标参数, 可为笛卡尔坐标、关节坐标、笛卡尔坐标增量或关节坐标增量</p> <p>isQueued: 是否将该指令指定为队列命令。取值范围: 0, 立即指令; 1, 队列指令</p>
返回	<p>如果设置为立即指令, 则返回 0</p> <p>如果设置为队列指令, 则返回该指令在队列的索引号</p>
示例	<p>机械臂在 MOVJ 模式下运行至 (244, -136, 80,0)</p> <p>dType.SetPTPCCommonParams(api, 50, 50, 1)</p>

```
dType.SetPTPCmd(api, 1, 244, -136, 80,0, 1)
```

2.2.8 同步模式下执行 PTP 指令

表 2.8 同步模式下执行 PTP 指令

原型	dType.SetPTPCmdSync(api, ptpMode, x, y, z, rHead)
功能	同步模式下执行 PTP 指令。设置 PTP 相关参数后，调用此函数可使机械臂运动至设置的目标点
参数	<p>api: Dobot 库对象，不能修改</p> <p>ptpMode: PTP 模式，取值范围：0~9</p> <p>0: JUMP 模式，平移时运动模式为 MOVJ，(x,y,z,rHead) 为笛卡尔坐标系下的目标点坐标</p> <p>1: MOVJ 模式，(x,y,z,rHead) 为笛卡尔坐标系下的目标点坐标</p> <p>2: MOVL 模式，(x,y,z,rHead) 为笛卡尔坐标系下的目标点坐标</p> <p>3: JUMP 模式，(x,y,z,rHead) 为关节坐标系下的目标点坐标</p> <p>4: MOVJ 模式，(x,y,z,rHead) 为关节坐标系下的目标点坐标</p> <p>5: MOVL 模式，(x,y,z,rHead) 为关节坐标系下的目标点坐标</p> <p>6: MOVJ 模式，(x,y,z,rHead) 为关节坐标系下的坐标增量</p> <p>7: MOVL 模式，(x,y,z,rHead) 为笛卡尔坐标系下的坐标增量</p> <p>8: MOVJ 模式，(x,y,z,rHead) 为笛卡尔坐标系下的坐标增量</p> <p>9: JUMP 模式，平移时运动模式为 MOVL，(x,y,z,rHead) 为笛卡尔坐标系下的目标点坐标</p> <p>x, y, z, rHead: 坐标参数，可为笛卡尔坐标、关节坐标、笛卡尔坐标增量或关节坐标增量</p>
返回	无
示例	无

2.2.9 执行 ARC 指令

表 2.9 执行圆弧插补功能接口说明

原型	dType.SetARCCmd(api, cirPoint, toPoint, isQueued)
功能	<p>执行 ARC 指令。设置 ARC 运动的速度和加速度后，调用该函数可使机械臂按 ARC 模式运动至目标点</p> <p>该运动模式需结合其他模式一起使用，构成圆弧上三点</p>
参数	<p>api: Dobot 库对象，不能修改</p> <p>cirPoint: 圆弧中间点，需设置为笛卡尔坐标</p> <p>toPoint: 圆弧目标点，需设置为笛卡尔坐标</p>

	isQueued: 是否将该指令指定为队列命令。取值范围: 0, 立即指令; 1, 队列指令
返回	如果设置为立即指令, 则返回 0 如果设置为队列指令, 则返回该指令在队列的索引号
示例	dType.SetARCCmd(api, [62,265,120,50], [-58,266,120,76], 1)

2.2.10 同步模式下执行 ARC 指令

表 2.10 执同步模式下执行圆弧插补功能接口说明

原型	dType.SetARCCmdSync(api, cirPoint, toPoint)
功能	同步模式下执行 ARC 指令。设置 ARC 运动的速度和加速度后, 调用该函数可使机械臂按 ARC 模式运动至目标点 该运动模式需结合其他模式一起使用, 构成圆弧上三点
参数	api: Dobot 库对象, 不能修改 cirPoint: 圆弧中间点, 需设置为笛卡尔坐标 toPoint: 圆弧目标点, 需设置为笛卡尔坐标
返回	无
示例	无

2.2.11 执行 CIRCLE 指令

圆形模式与圆弧模式相似, 示教后存点回放的运动轨迹为整圆。使用圆形模式时, 也需结合其他运动模式确认圆形上的三点。

表 2.11 执行 CIRCLE 指令

原型	dType.SetCircleCmd(api, cirPoint, toPoint, count, isQueued)
功能	执行 CIRCLE 指令。设置整圆运动的速度和加速度后, 调用该函数可使机械臂按 CIRCLE 模式运动至目标点 该运动模式需结合其他模式一起使用, 构成圆上三点
参数	api: Dobot 库对象, 不能修改 cirPoint: 圆弧中间点, 需设置为笛卡尔坐标 toPoint: 圆弧目标点, 需设置为笛卡尔坐标 count: 整圆个数 isQueued: 是否将该指令指定为队列命令。取值范围: 0, 立即指令; 1, 队列指令
返回	如果设置为立即指令, 则返回 0 如果设置为队列指令, 则返回该指令在队列的索引号
示例	dType.SetCircleCmd (api, [62,265,120,50], [-58,266,120,76], 1, 1)

2.2.12 同步模式下执行 CIRCLE 指令

表 2.12 同步模式下执行 CIRCLE 指令

原型	dType.SetCircleCmdSync(api, cirPoint, toPoint, count)
功能	同步模式下执行 CIRCLE 指令。设置整圆运动的速度和加速度后，调用该函数可使机械臂按 CIRCLE 模式运动至目标点 该运动模式需结合其他模式一起使用，构成圆上三点
参数	api: Dobot 库对象，不能修改 cirPoint: 圆弧中间点，需设置为笛卡尔坐标 toPoint: 圆弧目标点，需设置为笛卡尔坐标 count: 整圆个数
返回	无
示例	无

2.2.13 设置机械臂方向

表 2.13 设置机械臂方向

原型	dType.SetArmOrientation(api, armOrientation, isQueued)
描述	设置机械臂方向
参数	api: Dobot 库对象，不能修改 armOrientation: 机械臂方向。0: 左手方向；1: 右手方向 isQueued: 是否将该指令指定为队列命令。取值范围：0，立即指令；1，队列指令
返回	如果设置为立即指令，则返回 0 如果设置为队列指令，则返回该指令在队列的索引号
示例	dType.SetArmOrientation(api, 0, 1) dType.SetPTPCmdSync(api, 1, 300, 0, 100, 0, 1)

2.2.14 同步模式下设置机械臂方向

表 2.14 同步模式下设置机械臂方向

原型	dType.SetArmOrientationSync(api, armOrientation)
描述	同步模式下设置机械臂方向
参数	api: Dobot 库对象，不能修改 armOrientation: 机械臂方向。0: 左手方向；1: 右手方向
返回	无

示例	无
----	---

2.2.15 获取机械臂实时位姿

表 2.15 获取实时位姿

原型	dType.GetPose(api)
功能	获取机械臂实时位姿
参数	api: Dobot 库对象, 不能修改
返回	x (List[0]): 笛卡尔坐标系下 X 轴坐标 y (List[1]): 笛卡尔坐标系下 Y 轴坐标 z (List[2]): 笛卡尔坐标系下 Z 轴坐标 r (List[3]): 笛卡尔坐标系下 R 轴坐标 J1 (List[4]): 关节坐标系下 J1 轴坐标 J2 (List[5]): 关节坐标系下 J2 轴坐标 J3 (List[6]): 关节坐标系下 J3 轴坐标 J4 (List[7]): 关节坐标系下 J4 轴坐标
示例	无

2.3 WAIT 功能

2.3.1 执行时间等待指令

表 2.16 执行时间等待功能接口说明

原型	dType.SetWAITCmd(api, waitTime, isQueued)
功能	执行时间等待指令。如果需设置前个指令运行后的暂停时间, 可调用此函数 该指令只能作为队列指令, “isQueued” 必须设置为 “1”。将该命令设置为立即指令可能会导致正在执行的 WAIT 队列指令的暂停时间变化
参数	api: Dobot 库对象, 不能修改 waitTime: 等待时间。单位: ms isQueued: 是否将该指令指定为队列命令。取值范围: 0, 立即指令; 1, 队列指令
返回	如果设置为立即指令, 则返回 0 如果设置为队列指令, 则返回该指令在队列的索引号
示例	dType.SetWAITCmd(api, 1000, 1)

2.3.2 同步模式下执行时间等待指令

表 2.17 同步模式下执行时间等待功能接口说明

原型	dType.SetWAITCmdSync(api, waitTime)
功能	同步模式下执行时间等待指令
参数	api: Dobot 库对象，不能修改 waitTime: 等待时间。单位: ms
返回	无
示例	无

2.3.3 延时指令

表 2.18 延时指令

原型	dType.dSleep(ms)
功能	延时指令（上位机下发）
参数	ms: 延时时间单位: ms
返回	无
示例	无

2.4 触发功能

2.4.1 执行触发指令

表 2.19 执行触发指令

原型	dType.SetTRIGCmd(api, address, mode, condition, threshold, isQueued)
功能	执行触发功能 该指令只能作为队列指令，“isQueued”必须设置为“1”。将该命令设置为立即指令可能会导致正在执行的 TRIG 队列指令的触发条件变化。
参数	api: Dobot 库对象，不能修改 address: I/O地址。若mode设置为0，取值范围: 1~24。若mode设置为1，取值范围: 1~6 mode: 触发模式。0: 电平触发。1: A/D 触发 condition: 触发条件 电平触发: 0, 等于。1, 不等于 A/D 触发: 0, 小于。1, 小于等于。2, 大于等于。3, 大于 threshold: 触发阈值: 电平触发, 0 或 1。A/D 触发: 0~4095 isQueued: 是否将该指令指定为队列命令。取值范围: 0, 立即指令; 1, 队列指令
返回	如果设置为立即指令，则返回 0 如果设置为队列指令，则返回该指令在队列的索引号

示例	无
----	---

2.4.2 同步模式下执行触发指令

表 2.20 同步模式下执行触发指令

原型	dType.SetTRIGCmdSync(api, address, mode, condition, threshold)
功能	同步模式下执行触发功能
参数	api: Dobot 库对象, 不能修改 address: I/O地址。若mode设置为0, 取值范围: 1~24。若mode设置为1, 取值范围: 1~6 mode: 触发模式。0: 电平触发。1: A/D 触发 condition: 触发条件 电平触发: 0, 等于。1, 不等于 A/D 触发: 0, 小于。1, 小于等于 2, 大于等于。3, 大于 threshold: 触发阈值: 电平触发, 0 或 1。A/D 触发: 0~4095
返回	无
示例	无

2.5 I/O 功能

在 Dobot 控制器中, 所有的扩展 I/O 都是统一编址的。根据现有情况, I/O 的功能包括以下内容:

- 高低电平输出功能。
- 读取输入高低电平功能。
- 读取输入模数转换值功能。

I/O 详细说明, 请参见《Dobot M1 用户手册》。

2.5.1 设置数字输出电平

表 2.21 设置数字输出电平

原型	dType.SetIODO(api, address, level, isQueued)
功能	设置数字输出电平
参数	api: Dobot 库对象, 不能修改 address: 数字输出 I/O 地址。取值范围: 1~22 level: 输出电平。0: 低电平。1: 高电平 isQueued: 是否将该指令指定为队列命令。取值范围: 0, 立即指令; 1, 队列指令

返回	如果设置为立即指令，则返回 0 如果设置为队列指令，则返回该指令在队列的索引号
示例	dType.SetIODO(api, 1, 0, 0)

2.5.2 同步模式下设置数字输出电平

表 2.22 同步模式下设置数字输出输出电平

原型	dType.SetIODOSync(api, address, level)
功能	同步模式下设置数字输出电平
参数	api: Dobot 库对象，不能修改 address: 数字输出 I/O 地址。取值范围：1~22 level: 输出电平。0: 低电平。1: 高电平
返回	无
示例	dType.SetIODOSync(api, 1, 1)

2.5.3 读取数字输出电平

表 2.23 读取数字输出电平

原型	dType.GetIODO(api, addr)
功能	读取数字输出电平
参数	api: Dobot 库对象，不能修改 address: 数字输出 I/O 地址。取值范围：1~22
返回	对应数字输出电平值。0: 低电平；1: 高电平
示例	dType.SetIODOSync(api, 1, 1) level = dType.GetIODO(api, 1)

2.5.4 读取数字输入电平

表 2.24 读取数字输入电平

原型	dType.GetIODI(api, addr)
功能	读取数字输入电平
参数	api: Dobot 库对象，不能修改 address: 数字输入 I/O 地址。取值范围：1~24
返回	对应数字输入电平值。0: 低电平；1: 高电平
示例	无

2.5.5 读取连续数字输入地址的电平

表 2.25 读取连续数字输入地址的电平

原型	dType.GetIODIs(api, startAddr, endAddr)
功能	读取连续数字输入地址的电平
参数	api: Dobot 库对象, 不能修改 startAddr: 数字输入起始地址。取值范围: 1~24 endAddr: 数字输入结束地址。取值范围: 1~24
返回	对应连续数字输入地址的电平值
示例	无

2.5.6 读取模拟输入值

表 2.26 读取模拟输入

原型	dType.GetIOADC(api, addr)
功能	读取模拟输入值
参数	api: Dobot 库对象, 不能修改 address: 模拟输入地址。取值范围: 1~6
返回	返回模拟输入值。取值范围: 0~4095
示例	无

2.6 其他

2.6.1 加载 Playback 数据

表 2.27 加载 Playback 数据

原型	playbackData.load(fileName)
功能	加载 Playback 脚本中的数据 Playback 脚本必须保存在“安装目录\M1Studio\config\pbstore”路径下, 否则会造成加载 Playback 数据失败
参数	filename: Playback 脚本名称, 格式: 字符串
返回	无
示例	请参见程序 2.1

2.6.2 获取 Playback 数据

表 2.28 加载 Playback 数据

原型	playbackData.get(rowName)
功能	根据 Playback 脚本中具体行名称获取对应点位数据 该指令需与 playbackData.load(fileName)一起使用
参数	rowName: Playback 数据中对应行名称, 格式: 字符串
返回	对应行中的点位数据, 由 x, y, z,r 组成
示例	请参见程序 2.1

程序 2.1 加载 Playback 数据示例

```
#加载 playback 文件 文件必须存储在 config/pbstore 路径下 只填写文件名即可
playbackData.load("playbackDemo")
#获取某一行的点位坐标 该行的名称 (name) 执行 ptp 命令
for i in range(0,5):
pose = playbackData.get("first")
dType.SetPTPCmdSync(api, 1, pose[0], pose[1], pose[2], pose[3], 1)
pose = playbackData.get("second")
dType.SetPTPCmdSync(api, 1, pose[0], pose[1], pose[2], pose[3], 1)
```

2.6.1 设置码垛名称

表 2.29 设置码垛名称

原型	matrixPallet.setMatrixPallet(name)
功能	设置码垛名称 码垛名称需和“Script”页面创建码垛时名称一样
参数	name: 码垛名称。格式: 字符串
返回	无
示例	详细请参见程序 2.2

2.6.2 获取准备点数据

表 2.30 获取准备点数据

原型	matrixPallet.getReadyPoint()
功能	获取码垛准备点数据 调用此接口前需先创建码垛
参数	无
返回	码垛准备点数据

示例	详细请参见程序 2.2
----	-------------

2.6.3 获取过渡点数据

表 2.31 获取过渡点数据

原型	matrixPallet.getTransPoint()
功能	获取过渡点数据 调用此接口前需先创建码垛
参数	无
返回	码垛过渡点数据
示例	详细请参见程序 2.2

2.6.4 获取码垛点数据

表 2.32 获取码垛点数据

原型	matrixPallet.getPalletPoint()
功能	获取码垛点数据 调用此接口前需先创建码垛
参数	无
返回	码垛点数据
示例	详细请参见程序 2.2



注意

如果在脱机条件下执行码垛程序，需确保创建码垛时Dobot M1与电脑通过网线连接。如果通过串口连接，则无法在脱机条件下执行码垛程序。

程序 2.2 码垛示例

```
#set matrix pallet config
matrixPallet.setMatrixPallet("matrixTest")
#get matrix pallet ready point
readyPoint = matrixPallet.getReadyPoint()
#get matrix transition ready point
transPoint = matrixPallet.getTransPoint()
#get matrix pallet point list
palletPoint = matrixPallet.getPalletPoint()

for point in palletPoint:
```

```
#go to ready point
print("ready Point:", readyPoint)
dType.SetPTPCmd(api, 1, readyPoint[0], readyPoint[1], readyPoint[2], readyPoint[3], isQueued=1)
#go to transition point
print("trans Point:", transPoint)
dType.SetPTPCmd(api, 1, transPoint[0], transPoint[1], transPoint[2], transPoint[3], isQueued=1)
#go to pallet point
print("pallet point:", point)
dType.SetPTPCmd(api, 1, point[0], point[1], point[2], point[3], isQueued=1)
```