深圳市越疆科技有限公司

Dobot M1

Demo说明（LabVIEW）

Demo说明

文档版本：V2.2

发布日期：2021-1-7



**版权所有** **© 越疆科技有限公司2017。 保留一切权利。**

非经本公司书面许可，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。

**免责申明**

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：深圳市南山区桃源街道福光社区留仙大道3370号

南山智园崇文园区2号楼1003

网址：<http://cn.dobot.cc/>

前 言

**修订记录**

| 时间 | 修订记录 |
| --- | --- |
| 2021/1/7 | 第一次发布 |
| 2021/1/25 | 新增17版本的说明 |
| 2021/1/26 | 新增常见错误说明 |
| 2021/2/22 | 新增部分API说明 |

**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. 基于LabVIEW的Demo 1](#_Toc27938)

[1.1 环境搭建 1](#_Toc10735)

[1.2 LabVIEW Demo说明 4](#_Toc21438)

[1.2.1 工程说明 4](#_Toc1502)

[1.2.2 程序框图说明 5](#_Toc12358)

[1.3 常见错误说明 7](#_Toc16485)

[1.3.1 TCP连接错误 7](#_Toc11509)

[1.3.2 安装报错 8](#_Toc28587)

[1.4 部分API使用说明 8](#_Toc12486)

[1.4.1 Get PTPJumpParams 9](#_Toc10858)

[1.4.2 设置和获取关节点位参数（Set/Get PTPJointParams） 9](#_Toc15504)

[1.4.3 设置和获取坐标轴点位参数（Set/Get PTPCoordinateParams） 9](#_Toc10010)

[1.4.4 设置和获取关节坐标系点动参数（Set/Get JOGJointParams） 10](#_Toc27131)

[1.4.5 设置和获取笛卡尔坐标系点动参数（Set/Get JOGCoordinateParams） 10](#_Toc10632)

[1.4.6 设置手臂方向（SetArmOrientation） 11](#_Toc25411)

[1.4.7 获取实时位姿（GetPose） 11](#_Toc23068)

[1.4.8 校准机械臂（ResetPose） 11](#_Toc8671)

[1.4.9 获取报警状态（GetAlarmsState） 12](#_Toc15453)

[1.4.10 设置/获取用户坐标系（Set/Get UserCoordinate） 12](#_Toc29569)

[1.4.11 设置/获取工具坐标系（Set/Get ToolCoordinate） 12](#_Toc26220)

# 基于LabVIEW的Demo

本文档旨在协助用户熟悉Dobot M1常用API和快速搭建开发环境。

## 环境搭建

该Demo基于LabVIEW实现，所以需安装LabVIEW。

本文以Windows 10操作系统为例进行安装配置说明，请用户根据实际情况替换。

**操作步骤**

安装LabVIEW，以“LabVIEW 2018”为例，安装方法不做详细说明。

安装LabVIEW插件 vipm-17.0.2007-windows-setup.exe，安装方法不做详细说明。

双击 。安装包。此安装包是基于LabVIEW环境下调用Dobot API所封装成的 SDK. 通过VI Package Manager可以进行安装。（labview17对应的版本是）

点击“Install”按钮后开始进行安装，在安装过程中会自动打开LabVIEW软件，安装结束前不要关闭LabVIEW软件。如下图1.1所示。

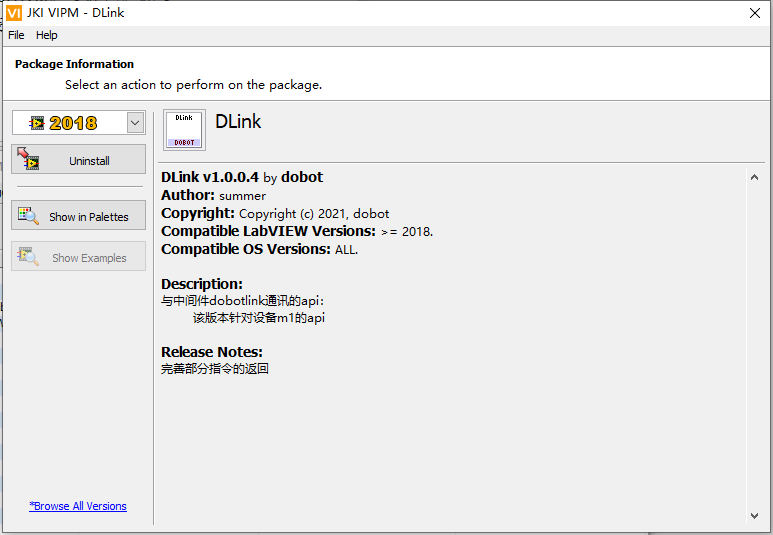


图 1.1 安装 SDK

安装成功后，可以在主页面选择“Installed”模式可以看到显示已经安装了Dlink安装包。如下图 1.2 所示。

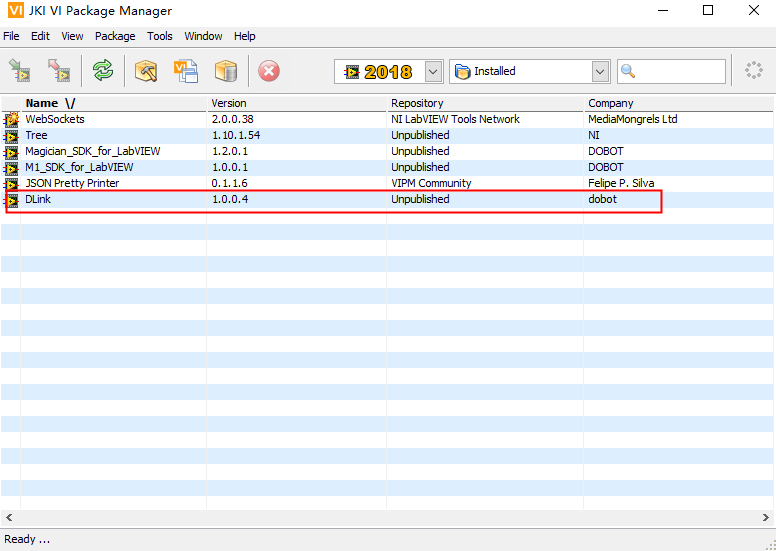
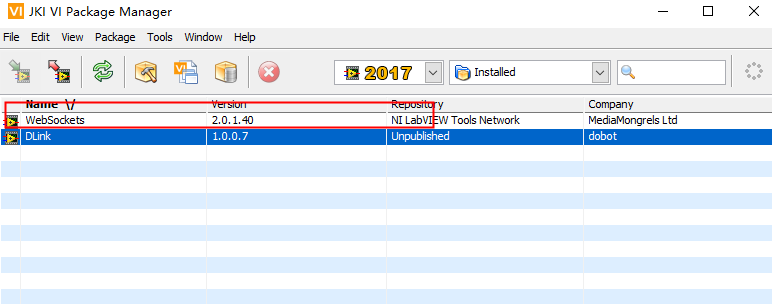


图 1.2 安装sdk成功界面

点击链接（vipm://mediamongrels\_ltd\_lib\_websockets\_api）出现如下图所示框。安装我们所需要的websocket插件。



安装成功后的如图所示。



启动LabVIEW并导入LabVIEW Demo，如图 1.3所示。

其中，“template.vi”为用户提供如何调用dobotlink的api的模板。“alarm.vi”为调用报警指令的前面板，“arc.vi”为调用圆弧插补指令的前面板，“coordinate.vi”为调用坐标系变换指令的前面板，“cp.vi”为调用连续运动轨迹指令的前面板，“device.vi”为调用设备信息指令的前面板，“io.vi”为调用端口输入输出指令的前面板，“jog.vi”为调用点动功能的前面板，“ptp.vi”为调用PTP指令的前面板。

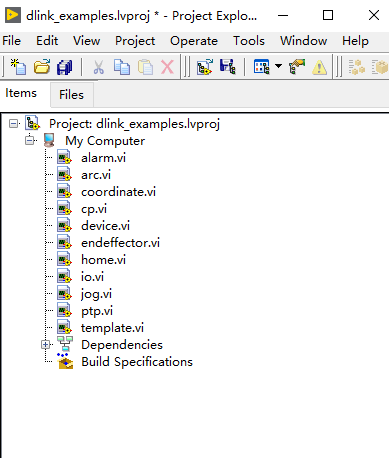


图 1.3 打开LabVIEW Demo工程

在LabVIEW Demo工程中打开“device.vi”。

显示“device.vi”前面板，“Ctrl+E”打开程序框图，如图1.4所示。

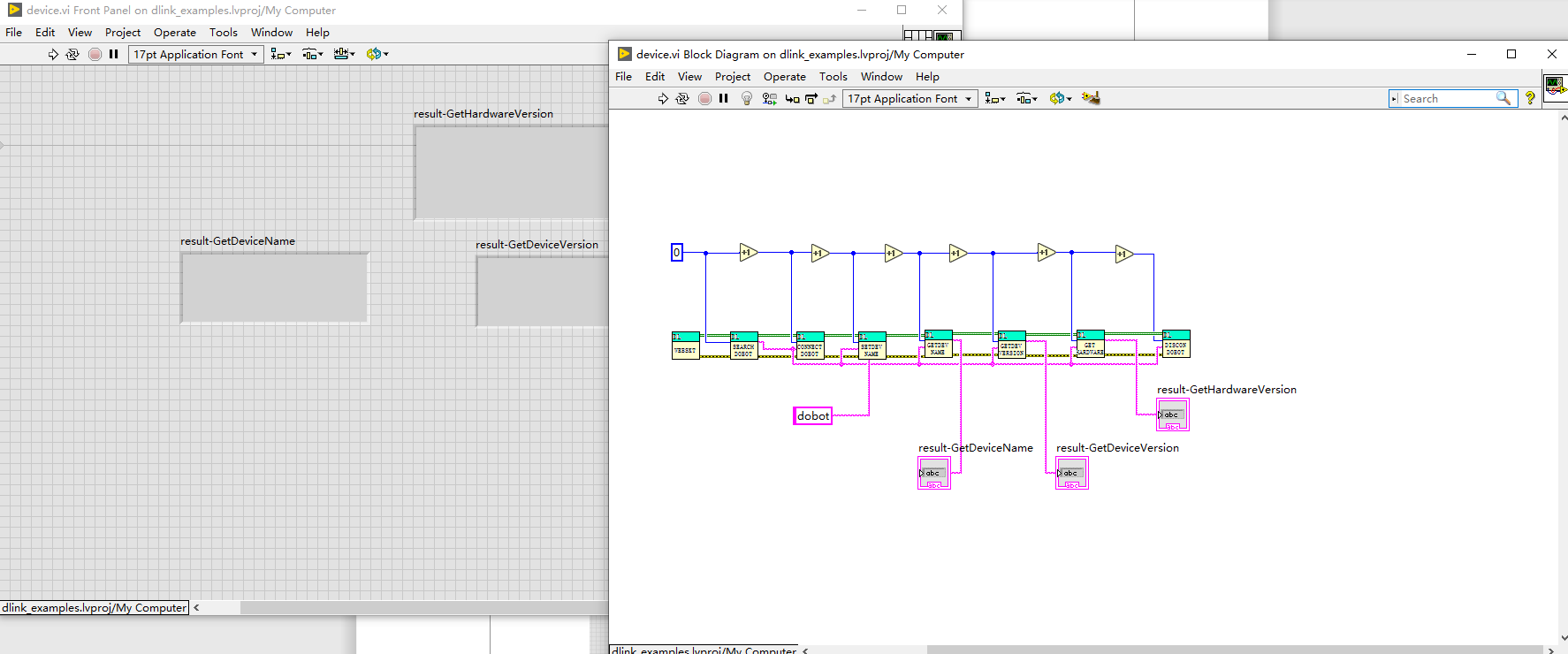


图1.4 device的前面板和程序框图

连接机械臂，单击运行即可。

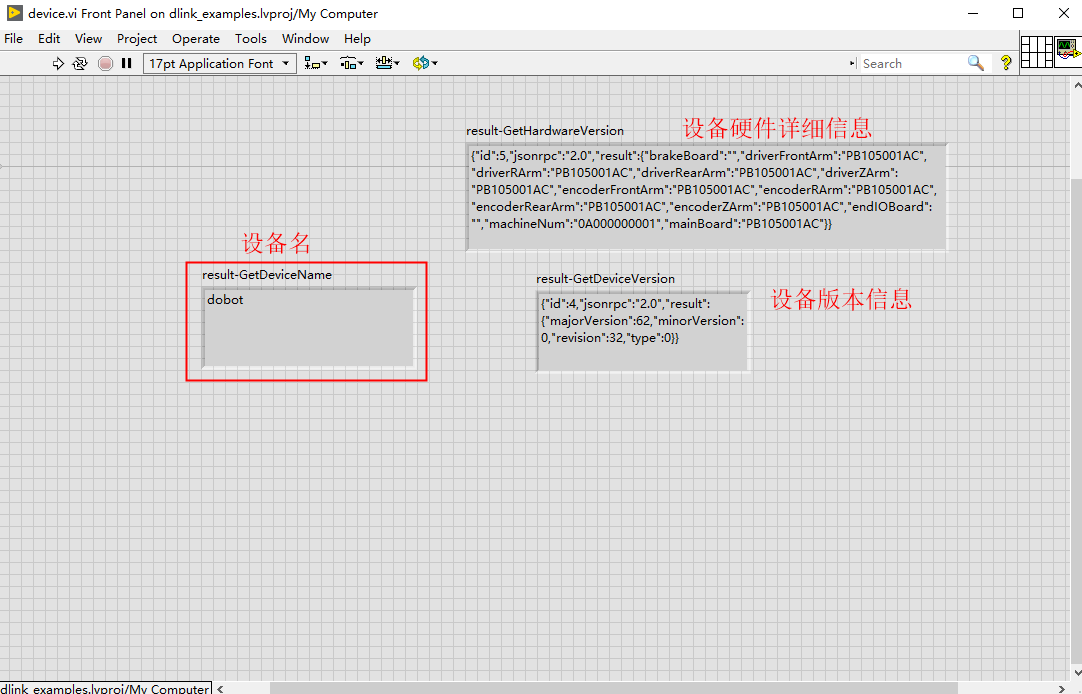


图 1.5 device.vi运行结果

## LabVIEW Demo说明

### 工程说明

用户打开“template.vi”，前面板和程序框图如下图1.6所示。

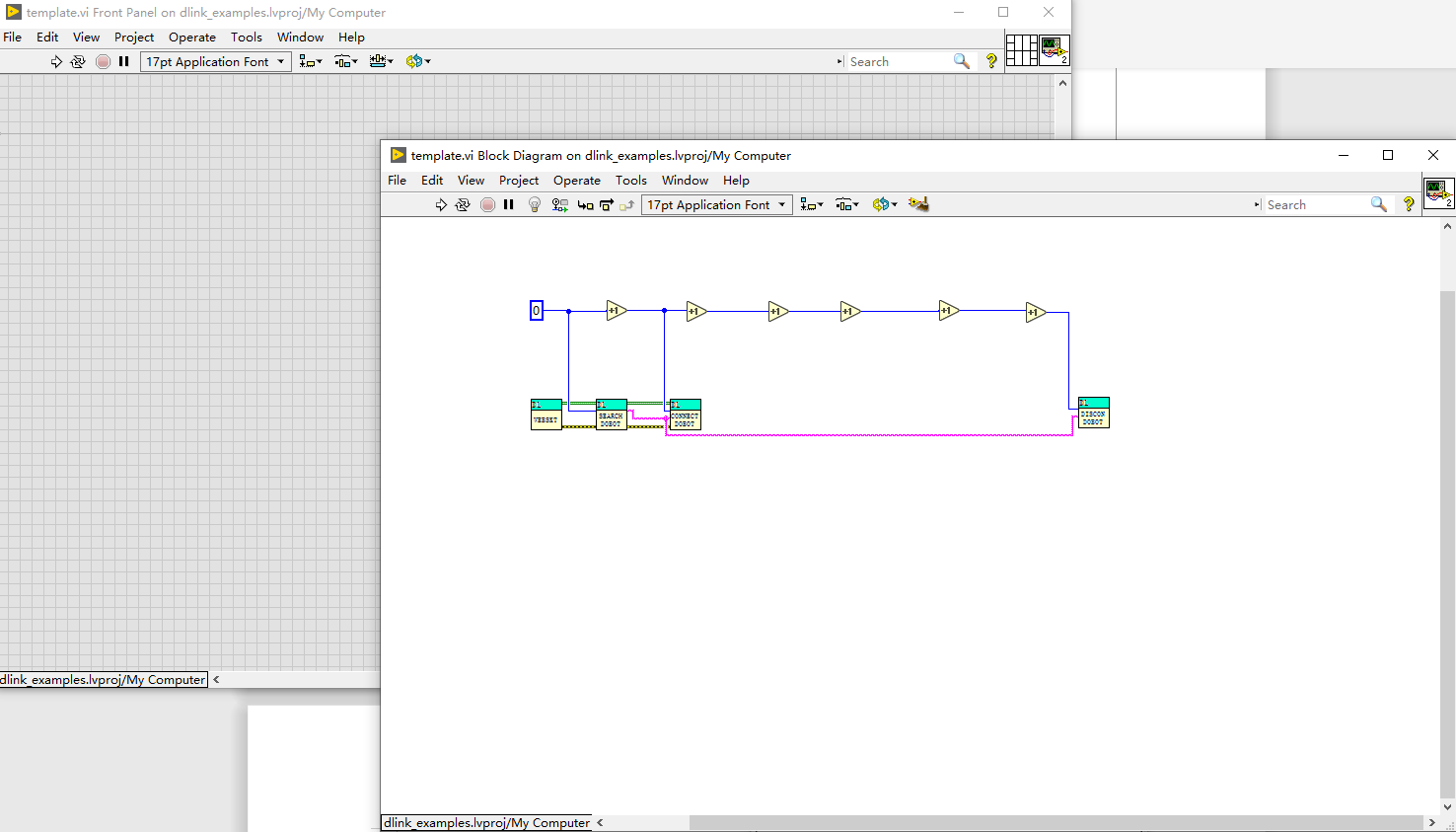


图 1.6 template前面板和程序框图示意图

用户可在程序框图界面右键选择“用户库 > DLink”调用DobotLink API，如图 1.7所示。详细的API说明请参考《dobot-m1-api》。

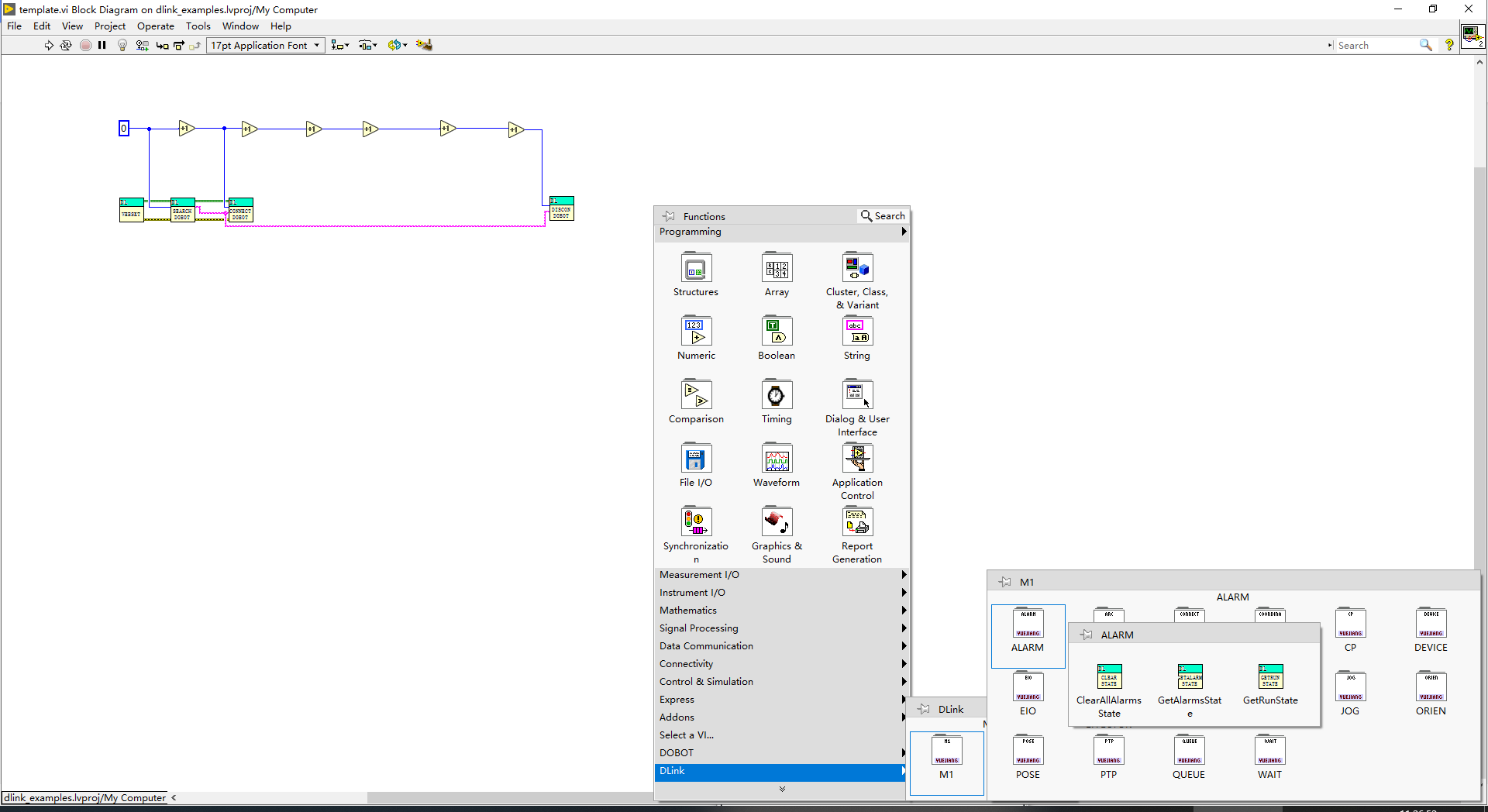
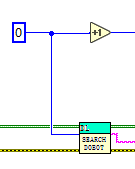


图 1.7 调用API

### 程序框图说明

本示例基于与dobotlink的websocket通讯，所以需调用websocket的api。设备指令基本连接端口有以下（id为每一个指令的专属ID，不可重复，websocket输入输出，error输入输出，结果输出）。

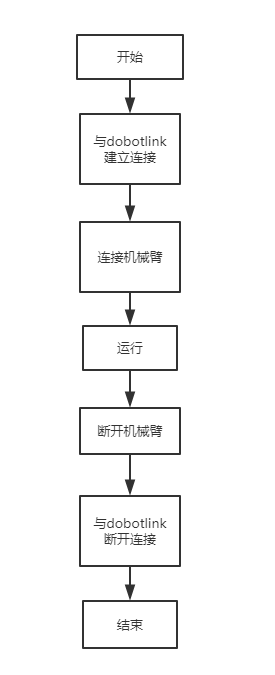


图 1.8 Demo实现流程

Demo各程序框图说明如下所示。

* + - * 1. 与dobotlink建立连接。

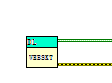


图 1.9 与dobotlink建立连接

* + - * 1. 连接机械臂。

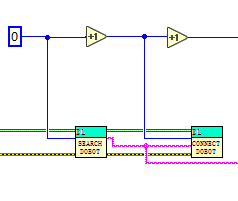


图 1.10 连接机械臂

* + - * 1. 断开机械臂。

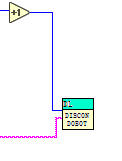


图 1.11 通过指令队列执行PTP指令

* + - * 1. 与dobotlink断开连接（程序运行结束自动断开与dobotlink的连接，所以此处可不连）。



图 1.12 断开连接

## 常见错误说明

### TCP连接错误

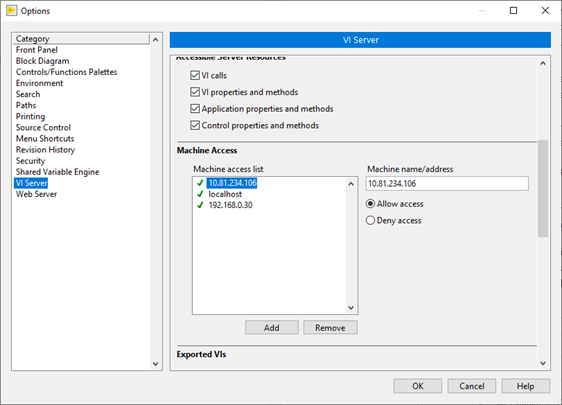
#### lALPBFDk9hkfzzDNAh7NBTk_1337_542

解决：请确认dobotlink是否正常开启，开启成功标志：右下角显示图标

### 安装报错

#### 安装报错，以下解决方案作为参考：

LabVIEW -> Options -> VI server -> Machine Access -> add -> "localhost"



设置后请尝试再次安装。

## 部分API使用说明

所有API说明请查看文档dobot-m1-api.pdf,现将部分与该文档有出入的API说明.

### Get PTPJumpParams

该api输出多了一个isUsingZLimit，一般情况下该参数默认为true，用户可不作考虑。

### 

### 设置和获取关节点位参数（Set/Get PTPJointParams）

这两条命令用于设置和获取再现速度参数，包括关节再现速度和加速度，此命令设置的速度相关参数仅适用于再现运动，对于JOG功能无效。

J1-J4的速度最大值分别是：180、180、1000、1000。

J1-J4的加速度最大值分别是：20000、20000、80000、50000。

#### 设置关节坐标系点位参数（SetPTPJointParams），用于控制再现运动的速度实现运动的快慢控制。

表 1 设置关节点位参数

|  |  |
| --- | --- |
| 原型 | **SetPTPJointParams** |
| 功能 | 设置关节坐标系点位参数 |
| 参数 | velocity: array(float):分别对应设置J1-J4的速度设置  acceleration: array(float):分别对应设置J1-J4的加速度设置 |
| 返回 | 指令是否设置成功。返回true或者false |

表 2 获取关节点位参数

|  |  |
| --- | --- |
| 原型 | **GetPTPJointParams** |
| 功能 | 获取关节坐标系点位参数 |
| 参数 | 无 |
| 返回 | velocity: array(float):分别对应设置J1-J4的速度设置  acceleration: array(float):分别对应设置J1-J4的加速度设置 |

### 

### 设置和获取坐标轴点位参数（Set/Get PTPCoordinateParams）

表 3 设置笛卡尔坐标系点位参数

|  |  |
| --- | --- |
| 原型 | **SetPTPCoordinateParams** |
| 功能 | 设置笛卡尔坐标系点位参数 |
| 参数 | xyzVelocity：float //PTP模式下xyz 3轴坐标轴速度  rVelocity：float //PTP模式下末端速度  xyzAcceleration：float //PTP模式下xyz 3轴坐标轴加速度  rAccleration：float //PTP模式下末端加速度 |
| 返回 | 指令是否设置成功。返回true或者false |

表 4 获取笛卡尔坐标系点位参数

|  |  |
| --- | --- |
| 原型 | **GetPTPCoordinateParams** |
| 功能 | 获取笛卡尔坐标系点位参数 |
| 参数 | 无 |
| 返回 | xyzVelocity：float //PTP模式下xyz 3轴坐标轴速度  rVelocity：float //PTP模式下末端速度  xyzAcceleration：float //PTP模式下xyz 3轴坐标轴加速度  rAccleration：float //PTP模式下末端加速度 |

### 设置和获取关节坐标系点动参数（Set/Get JOGJointParams）

表 5 设置关节点位参数

|  |  |
| --- | --- |
| 原型 | **SetJOGJointParams** |
| 功能 | 设置关节坐标系点位参数 |
| 参数 | velocity: array(float) //4轴关节速度  acceleration: array(float) //4轴关节加速度 |
| 返回 | 指令是否设置成功。返回true或者false |

表 6 获取关节点位参数

|  |  |
| --- | --- |
| 原型 | **GetJOGJointParams** |
| 功能 | 获取关节坐标系点位参数 |
| 参数 | 无 |
| 返回 | velocity: array(float) //4轴关节速度  acceleration: array(float) //4轴关节加速度 |

### 设置和获取笛卡尔坐标系点动参数（Set/Get JOGCoordinateParams）

表 7 设置坐标轴点位参数

|  |  |
| --- | --- |
| 原型 | **SetJOGCoordinateParams** |
| 功能 | 设置坐标轴点位参数 |
| 参数 | velocity: array(float) //该指令设置的是笛卡尔坐标系的参数，即X、Y、Z、R轴的速度  acceleration: array(float) //该指令设置的是笛卡尔坐标系的参数，即X、Y、Z、R轴的加速度 |
| 返回 | 指令是否设置成功。返回true或者false |

表 8 获取坐标轴点位参数

|  |  |
| --- | --- |
| 原型 | **GetJOGCoordinateParams** |
| 功能 | 获取坐标轴点位参数 |
| 参数 | 无 |
| 返回 | velocity: array(float) //该指令设置的是笛卡尔坐标系的参数，即X、Y、Z、R轴的速度  acceleration: array(float) //该指令设置的是笛卡尔坐标系的参数，即X、Y、Z、R轴的加速度 |

### 设置手臂方向（SetArmOrientation）

表 9 设置手臂方向

|  |  |
| --- | --- |
| 原型 | **SetArmOrientation** |
| 功能 | 设置手臂方向 |
| 参数 | orientation: int (0:lefty 1:righty) |
| 返回 | 指令是否设置成功。返回true或者false |

### 获取实时位姿（GetPose）

表 10 获取实时位姿

|  |  |
| --- | --- |
| 原型 | **GetPose** |
| 功能 | 获取实时位姿 |
| 参数 | 无 |
| 返回 | xyzr：array(float) //机械臂坐标系x、y、z、r  jointAngle：array(float) //机械臂关节轴(J1、J2、J3、J4)角度 |

### 校准机械臂（ResetPose）

表 11 校准机械臂

|  |  |
| --- | --- |
| 原型 | **ResetPose** |
| 功能 | 校准机械臂 |
| 参数 | frontAngle1: float // frontAngle1和frontAngle2分别为机械臂以左右手到达同一点时小臂的角度  frontAngle2: float |
| 返回 | 指令是否设置成功。返回true或者false |

### 获取报警状态（GetAlarmsState）

表 12 获取报警状态

|  |  |
| --- | --- |
| 原型 | **GetAlarmsState** |
| 功能 | 获取报警状态 |
| 参数 | 无 |
| 返回 | state: array(int) //数组state中的每一个字节可以标识8个报警项的报警状态，且MSB在高位，LSB在低位。 |

### 设置/获取用户坐标系（Set/Get UserCoordinate）

表 13 设置用户坐标系

|  |  |
| --- | --- |
| 原型 | **SetUserCoordinate** |
| 功能 | 设置用户坐标系 |
| 参数 | xyzr: array(float) |
| 返回 | 指令是否设置成功。返回true或者false |

表 14 获取用户坐标系

|  |  |
| --- | --- |
| 原型 | **GetUserCoordinate** |
| 功能 | 获取用户坐标系 |
| 参数 | 无 |
| 返回 | xyzr: array(float) |

### 设置/获取工具坐标系（Set/Get ToolCoordinate）

表 15 设置工具坐标系

|  |  |
| --- | --- |
| 原型 | **SetToolCoordinate** |
| 功能 | 设置工具坐标系 |
| 参数 | xyzr: array(float) |
| 返回 | 指令是否设置成功。返回true或者false |

表 16 获取工具坐标系

|  |  |
| --- | --- |
| 原型 | **GetToolCoordinate** |
| 功能 | 获取工具坐标系 |
| 参数 | 无 |
| 返回 | xyzr: array(float) |