

Xvisio\_ROS\_Wrapper\_Guide文档

*v1.0/ 2022.02*

***Xvisio Confidential***

**Copyright and Proprietary Information Notice**

Copyright © 2018 Xvisio Ltd. All right reserved. This document contains confidential and proprietary information that is the property of Xvisio Ltd. All other product or company names may be trademarks of their respective owners.

Xvisio Ltd.

Room 408, building 1, No.288, Tong Xie Road, Changning District, Shanghai

[http://www.Xvisiotech.com](http://www.xvisiotech.com/)

目录

[1. 概况 4](#_Toc96682706)

[2. Xvisio ROS wrapper接口介绍 4](#_Toc96682707)

[2.1 class RosWrapper接口 4](#_Toc96682710)

[2.2 class RosDevice接口 5](#_Toc96682711)

# 概况

本文档是Xvisio ROS wrapper的接口说明文档，主要是帮助开发者熟悉Xvisio ROS wrapper接口，并能够快速开发相关应用

# Xvisio ROS wrapper接口介绍



## class RosWrapper接口

* + 1. **获取设备接口**

接口功能：获取当前设备。

接口声明： bool cbGetDevices(xv\_sdk::GetDevices::Request &req, xv\_sdk::GetDevices::Response &res);

请求参数： 略

应答参数： res 返回device对象。

返回值：成功：true，失败：false

* + 1. **发布新设备接口**

接口功能：发布新设备广播。

接口声明：void RosWrapper::publishNewDevice(const std::string &rosNamespace)

请求参数：const std::string &rosNamespace 广播消息数据。

应答参数： 无

返回值：无

**2.1.3设备监测接口**

接口功能：监测设备列表。

接口声明： void watchDevices();

请求参数： 无

应答参数： 无

返回值：无

## class RosDevice接口

**2.2.1初始化接口**

接口功能：初始化设备。

接口声明： void init();

请求参数： 无

应答参数： 无

返回值：无

**2.2.2开启slam接口**

接口功能：开启slam功能。

接口声明：bool cbSlam\_start(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” Slam started”，失败：”failed”

返回值：成功：true，失败：false

**2.2.3关闭slam接口**

接口功能：关闭slam功能。

接口声明：bool cbSlam\_stop(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” Slam stopped”，失败：”failed”

返回值：成功：true，失败：false

**2.2.4获取预测时间的6dof位置接口**

接口功能：获取预测时间的6dof位置信息。

接口声明：bool cbSlam\_getPose(xv\_sdk::GetPose::Request &req, xv\_sdk::GetPose::Response &res);

请求参数： req.prediction 预测时间，单位为秒

应答参数： res.pose 预测时间的6dof位置信息

返回值：成功：true，失败：false

**2.2.5获取当前时间的6dof位置接口**

接口功能：5获取当前时间的6dof位置信息。

接口声明：bool cbSlam\_getPoseAt(xv\_sdk::GetPoseAt::Request &req, xv\_sdk::GetPoseAt::Response &res);

请求参数：req.timestamp 传入时间戳，单位为秒

应答参数： res.pose 传入时间戳对应的6dof位置信息

返回值：成功：true，失败：false

**2.2.6打开imu接口**

接口功能：打开imu功能。

接口声明：bool cbImuSensor\_startOri(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” Orientation started”，失败：”failed”

返回值：成功：true，失败：false

**2.2.7关闭imu接口**

接口功能：关闭imu功能。

接口声明：bool cbImuSensor\_stopOri(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” Orientation stopped”，失败：”failed”

返回值：成功：true，失败：false

**2.2.8获取预测时间的imu数据接口**

接口功能：获取预测时间的imu数据。

接口声明：bool cbImuSensor\_getOri(xv\_sdk::GetOrientation::Request &req, xv\_sdk::GetOrientation::Response &res);

请求参数： req.prediction 预测时间，单位为秒

应答参数： res. orientation预测时间的imu数据信息

返回值：成功：true，失败：false

**2.2.9获取当前时间的imu数据接口**

接口功能：获取当前时间的imu数据。

接口声明：bool cbImuSensor\_getOriAt(xv\_sdk::GetOrientationAt::Request &req, xv\_sdk::GetOrientationAt::Response &res);

请求参数：req.timestamp 传入时间戳，单位为秒

应答参数： res. orientation传入时间戳对应的imu数据信息

返回值：成功：true，失败：false

**2.2.10打开rgb相机接口**

接口功能：打开rgb相机功能。

接口声明：bool cbColorCamera\_start(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” Color camera started”，失败：”failed”

返回值：成功：true，失败：false

**2.2.11关闭rgb相机接口**

接口功能：关闭rgb相机功能。

接口声明：bool cbColorCamera\_stop(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” Color camera stopped”，失败：”failed”

返回值：成功：true，失败：false

**2.2.12打开tof相机接口**

接口功能：打开tof相机功能。

接口声明：bool cbTofCamera\_start(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” ToF camera started”，失败：”failed”

返回值：成功：true，失败：false

**2.2.13关闭tof相机接口**

接口功能：关闭tof相机功能。

接口声明：bool cbTofCamera\_stop(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” ToF camera stopped”，失败：”failed”

返回值：成功：true，失败：false

**2.2.14打开rgbd相机接口**

接口功能：打开rgbd相机功能。

接口声明：bool cbRgbd\_start(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);;

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” ToF camera started”，失败：”failed”

返回值：成功：true，失败：false

**2.2.15关闭rgbd相机接口**

接口功能：关闭rgbd相机功能。

接口声明：bool cbRgbd\_stop(std\_srvs::Trigger::Request &req, std\_srvs::Trigger::Response &res);

请求参数：略

应答参数： res.success：成功：true，失败：false

res.messag：成功：” ToF camera stopped”，失败：”failed”

返回值：成功：true，失败：false

**2.2.16获取ROS命名空间接口**

接口功能：获取ROS命名空间。

接口声明：std::string getRosNamespace() const;

返回值：ROS命名空间字符串。

**2.2.17获取帧ID接口**

接口功能：获取帧ID。

接口声明：std::string getFrameId(const std::string& defaultId) const;

输入参数：默认ID

返回值：帧ID字符串。