

上肢控制接口upper_controller.proto使用文档


1.依赖相关

选择操作系统框架，将grpc相关依赖放入

/usr/local/bin
/usr/local/include
/usr/local/lib



grpc_redist_x86_64.tar.gz
42.26MB





grpc_redist_aarch64.tar.gz
41.53MB




2.CMakeList相关

在CMakeList.txt中直接引入库


代码块

```
1 target_link_libraries(${PROJECT_NAME}
2     grpc++ grpc++_reflection grpc gpr protobuf absl_log_internal_check_op
3     absl_log_internal_message absl_log_internal_nullguard absl_cord
4     absl_cordz_info absl_synchronization absl_cordz_functions
5 )
```

3.upper_controller.proto接口说明





upper_controller.proto
1.41KB





 upper_controller.grpc.pb.h



 upper_controller.pb.h



upper_controller.pb.cc



upper_controller.grpc.pb.cc

这个proto文件定义了一个名为UpperController的gRPC服务，主要用于控制机器人上半身的运动和相关配置。

消息类型定义

位置相关消息：

EffectorPosition：包含左右两侧的float数组，表示末端执行器位置，不定长，可存入夹爪/灵巧手数据(单位映射值)

代码块

```
1 message EffectorPosition {
2     repeated float left = 1;
3     repeated float right = 2;
4 }
5
```

EndPose：包含左右两侧的float数组，表示末端位姿，顺序为xyzrpy，若可获取臂型角则存入第七维，否则将自动补齐默认值（单位rad+m）

代码块

```
1 message EndPose {
2     repeated float left = 1;
3     repeated float right = 2;
4 }
```

ArmPosition：包含左右两侧的float数组，表示双臂关节角数据，可変长，一般存入7维，可兼容5DoF机械臂（单位rad）

```

1 message ArmPosition {
2     repeated float left = 1;
3     repeated float right = 2;
4 }

```

NeckPose/WaistPose：分别表示颈部和腰部的姿态，可变长（单位rad）

代码块

```

1 message NeckPose {
2     repeated float neck = 1;
3 }
4
5 message WaistPose {
6     repeated float waist = 1;
7 }

```

复合消息：

EndPayload：组合末端位姿和末端执行器数据

ArmPayload：组合双臂关节角数据和末端执行器数据

代码块

```

1 message EndPayload {
2     EndPose end = 1;
3     EffectorPosition effector = 2;
4 }
5
6 message ArmPayload {
7     ArmPosition arm = 1;
8     EffectorPosition effector = 2;
9 }

```

响应消息：

Response：包含操作是否成功的布尔值和消息字符串

Config：包含多个配置参数（负责人、过滤级别、各部位模式等），其中延续ros2控制模式输入，在希望单轴控制时，请将不需要的模式置0！

代码块

```

1 message Config {

```

```

2      int32 incharge = 1; //inCharge=1;           //当loco、mani同时运行, mani是否介入控制
3      int32 filter_level = 2; //滤波等级0~5, 0最大滤波, >=5不滤波
4      int32 arm_mode = 3; //0无, 1回正, 2下肢命令透传, 3关节轴控, 4笛卡尔身体系
5      int32 digit_mode = 4; //0无, 1回正, 2下肢命令透传, 3关节轴控, 4伸直
6      int32 neck_mode = 5; //0无, 1回正, 2下肢命令透传, 3关节轴控, 4导航随动, 5看左手, 6
    看右手, 7看双手中间
7      int32 waist_mode = 6; //0无, 1回正, 2下肢命令透传, 3关节轴控, 4姿态控制
8  }

```

服务定义

UpperController服务提供了10个RPC方法, 可分为几类:

末端控制: sendEndAction/recvEndState

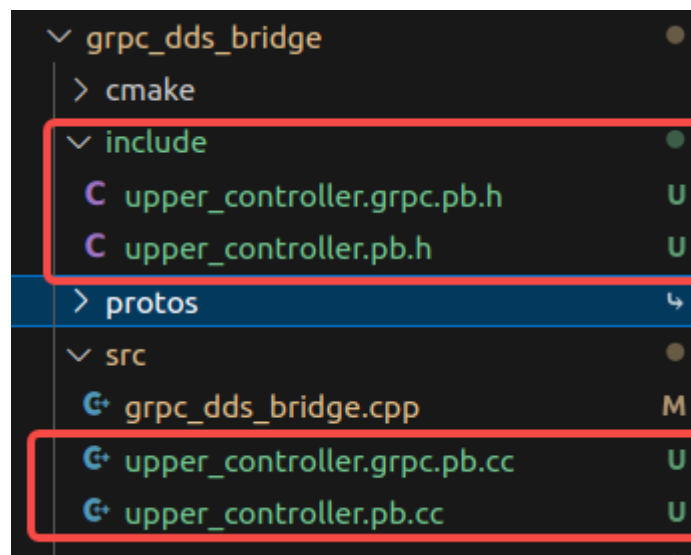
手臂控制: sendArmAction/recvArmState

配置管理: setConfig/getConfig

颈部控制: setNeckPose/getNeckPose

腰部控制: setWaistPose/getWaistPose

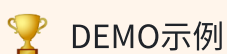
4.客户端说明



1. 发送频率

使用UnaryGrpc数据发送端为客户端, 发送频率尽量保证在100hz以上, 保证机器人的平滑运动

2. 客户端创建方式



DEMO示例

http://192.168.112.189/HumanoidShanghai/perception_grpc/src/branch/main/src/grpc

代码块

```
1 auto channel = grpc::CreateChannel(  
2     "localhost:50051",  
3     grpc::InsecureChannelCredentials()  
4 );  
5 auto stub = UpperController::NewStub(channel);
```

请注意，区别与ros2/dds通信，现所有控制接口合并，机器人内服务端有且只能接入1个客户端，客户端代码编写时，请注意例如在PDA在接入手动单轴模式时才可创建客户端，摇操设备在设备接入机器人内部程序时才可创建客户端，**任何设备断开即客户端需断开，不可长期占用。**

5.测试