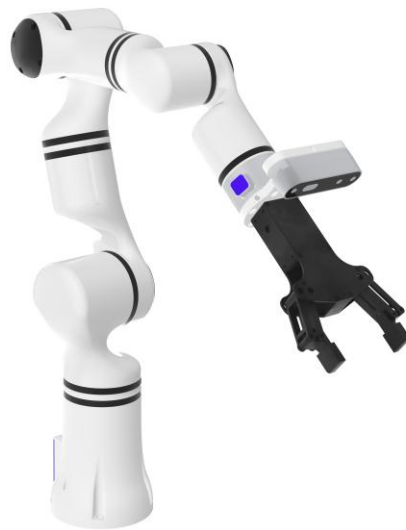


RM 机械臂与 Realsense D435 手眼标定教程



睿尔曼智能科技（北京）有限公司

文件修订记录:

版本号	时间	备注
V1.0	2021-11-14	拟制

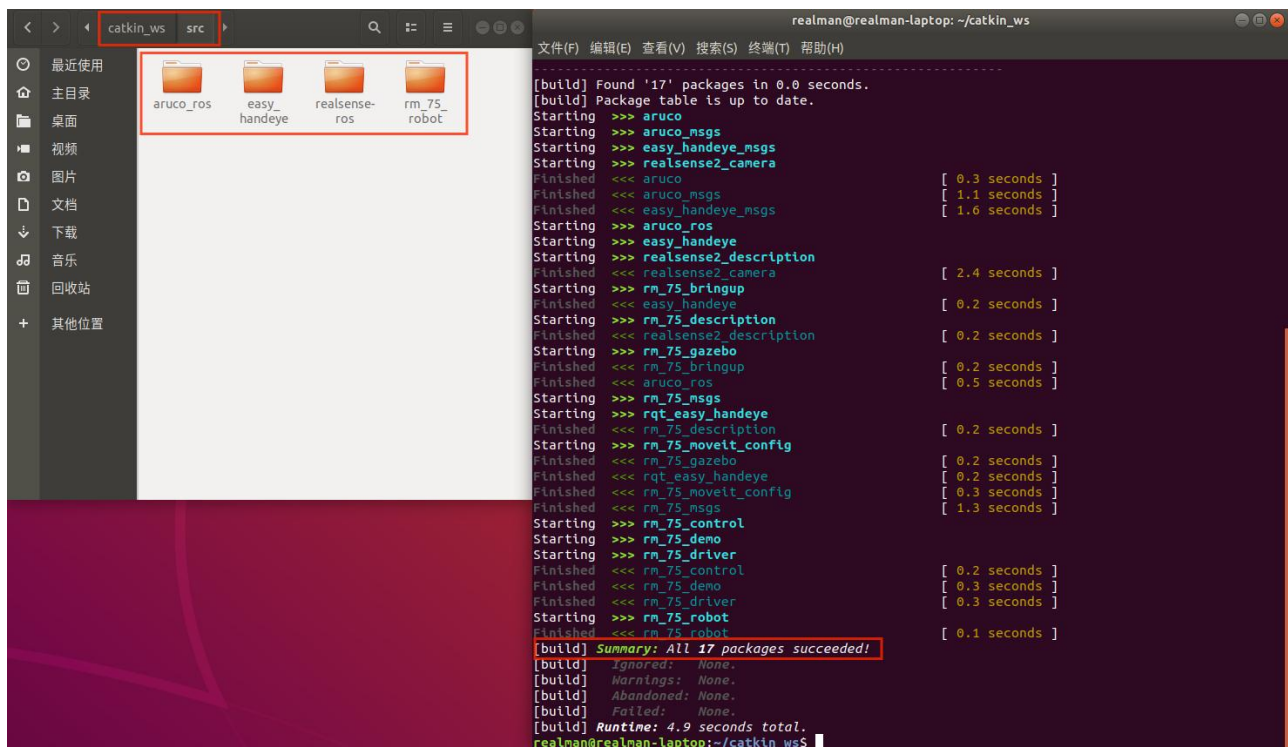
目录

1. 环境要求.....	4
2. 概述.....	4
3. 开始前准备.....	5
4. aruco_ros 配置.....	8
5. easy_handeye 配置.....	9
6. 启动相关 launch 文件开始标定.....	11

1. 环境要求

本教程主要介绍 RM 机械臂与 Realsense D435 相机手眼标定的配置及方法，由于不同处理器架构在系统环境及相关功能包的安装配置上存在差异，所以在此不做相关介绍，如果使用我司提供的设备，出厂默认会将系统环境及相关功能包安装配置好或者提供配套的相关系统环境搭建的教程，在此默认系统环境满足以下要求：

- 系统：Ubuntu 18.04.6
- ROS: melodic
- OpenCV 库：OpenCV 3.2.0
- Realsense D435: librealsense sdk (2.50.0)、realsense-ros 功能包 (2.3.2)
- Marker 标记识别：Aruco 功能包
- 手眼标定：easy_handeye 功能包
- Moveit!
- RM 机械臂 ROS 功能包（根据不同型号机械臂提供配套的 ROS 包，使用方法基本相同，提供有配套的使用说明文档，这里以 RM75-B 机械臂配套的 ROS 包即 rm_75_robot 为例）
- Catkin-tools 工具包



```
[build] Found '17' packages in 0.0 seconds.
[build] Package table is up to date.
Starting >>> aruco
Starting >>> aruco_msgs
Starting >>> easy_handeye_msgs
Starting >>> realsense2_camera
Finished <<< aruco [ 0.3 seconds ]
Finished <<< aruco_msgs [ 1.1 seconds ]
Finished <<< easy_handeye_msgs [ 1.6 seconds ]
Starting >>> aruco_ros
Starting >>> easy_handeye
Starting >>> realsense2_description
Finished <<< realsense2_camera [ 2.4 seconds ]
Starting >>> rm_75_bringup
Finished <<< easy_handeye [ 0.2 seconds ]
Starting >>> rm_75_description
Finished <<< realsense2_description [ 0.2 seconds ]
Starting >>> rm_75_gazebo
Finished <<< rm_75_bringup [ 0.2 seconds ]
Finished <<< aruco_ros [ 0.5 seconds ]
Starting >>> rm_75_msgs
Starting >>> rqt_easy_handeye
Finished <<< rm_75_description [ 0.2 seconds ]
Starting >>> rm_75_moveit_config
Finished <<< rm_75_gazebo [ 0.2 seconds ]
Finished <<< rqt_easy_handeye [ 0.2 seconds ]
Finished <<< rm_75_moveit_config [ 0.3 seconds ]
Finished <<< rm_75_msgs [ 1.3 seconds ]
Starting >>> rm_75_control
Starting >>> rm_75_deno
Starting >>> rm_75_driver
Finished <<< rm_75_control [ 0.2 seconds ]
Finished <<< rm_75_deno [ 0.3 seconds ]
Finished <<< rm_75_driver [ 0.3 seconds ]
Starting >>> rm_75_robot
Finished <<< rm_75_robot [ 0.1 seconds ]
[build] Summary: All 17 packages succeeded!
[build] Ignored: None.
[build] Warnings: None.
[build] Abandoned: None.
[build] Failed: None.
[build] Runtime: 4.9 seconds total.
realman@realman-laptop:~/catkin_ws$
```

2. 概述

手眼标定原理参考文章：[3D 视觉之手眼标定 \(qq.com\)](http://qq.com)

手眼标定分两种情况：

- Eye-to-hand 眼在手外：标定的是相机坐标系相对于机器人基座坐标系的位姿

- Eye-in-hand 眼在手上：标定的是相机坐标系相对于机器人工具坐标系的位姿

本教程主要介绍 RealSense D435 相机安装在 RM 机械臂上即眼在手上的标定方法。

3. 开始前准备

3.1 将相机安装固定到机械臂末端

将相机通过转接件固定到机械臂末端，如图所示（这里未安装末端工具，用户如需要可以一并装上）：



3.2 接线

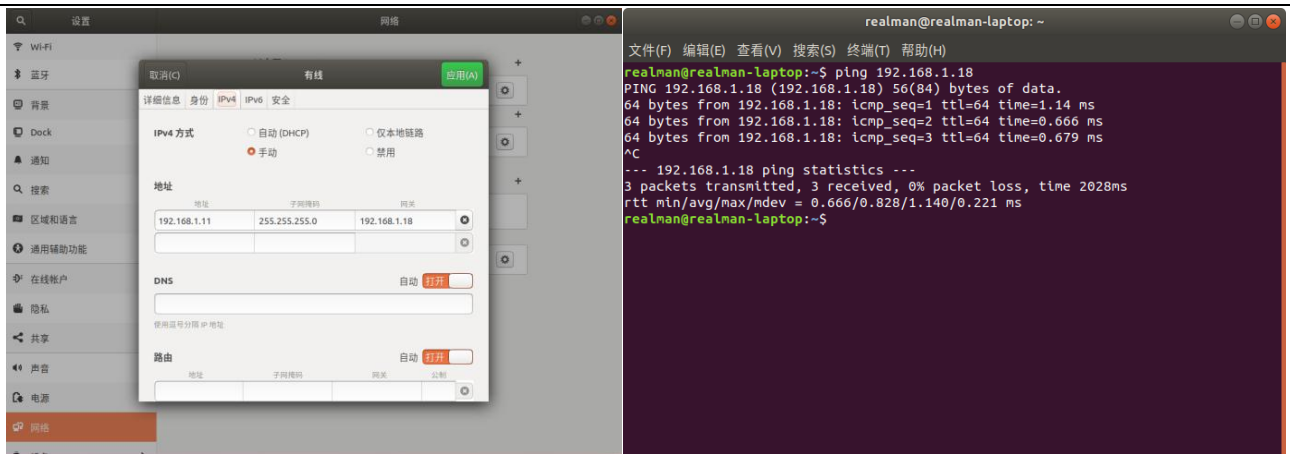
- 机械臂接上电源
- 机械臂外侧网口通过网线连接到主机或交换机上（注意：机械臂默认固定 IP 为 192.168.1.18，所以要确保主机与机械臂在同一局域网内能够连通）
- 相机通过 Type-C 数据线将相机与主机 USB3.0 接口相连

3.3 测试

主机与机械臂都上电启动后，在主机打开终端，执行以下命令测试主机与机械臂是否连通：

```
ping 192.168.1.18
```

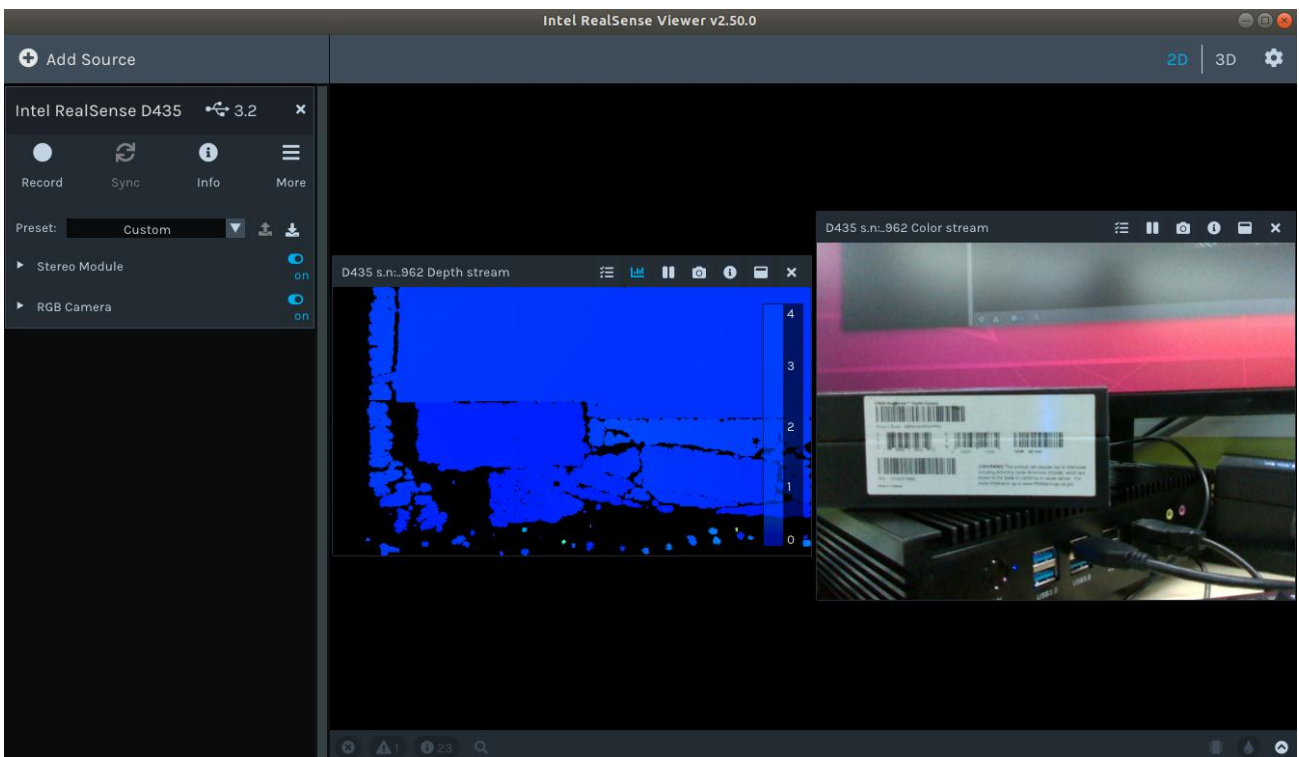
RM 机械臂与 RealSense D435 手眼标定教程



在终端中执行如下命令打开可视化测试界面(若未检测到相机尝试重新连接 USB):

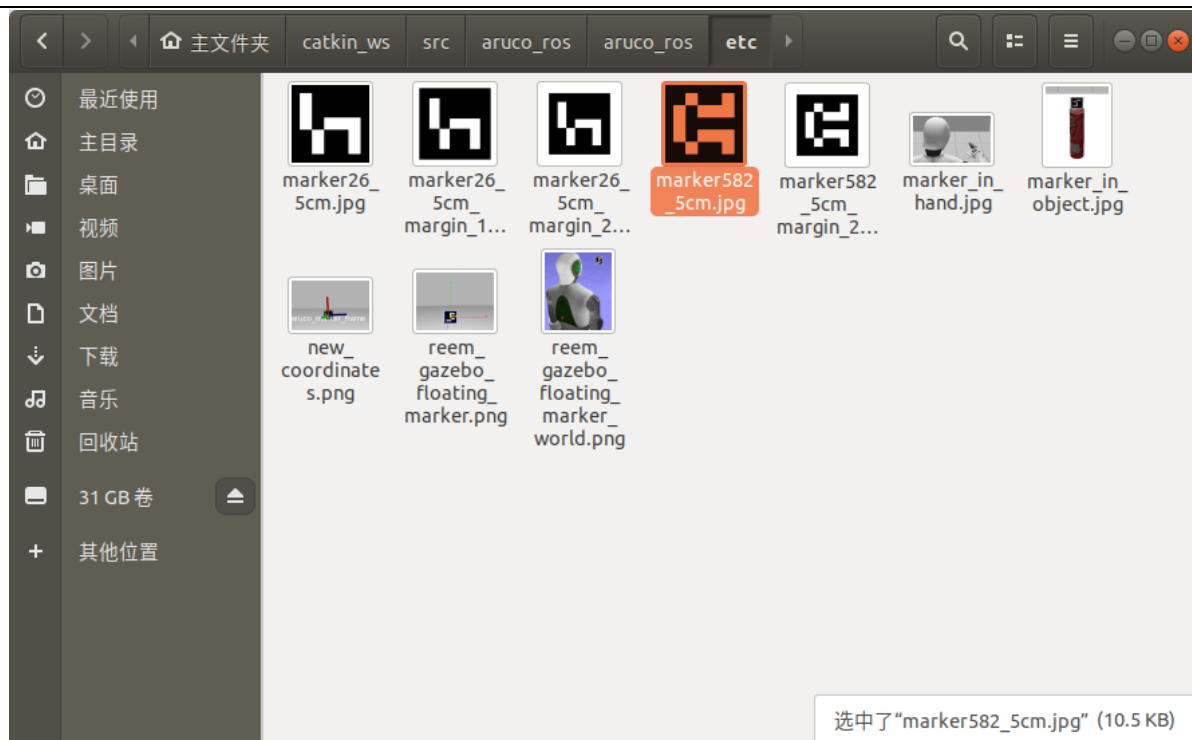
```
realsense-viewer
```

相机连接成功后，将左侧【Stereo Module】和【RGB Camera】选择“on”状态，右侧会显示深度图像和 RGB 图像画面。鼠标移动到深度图像位置左下方画面会显示实时测量的深度距离，如图说明相机连接正常：



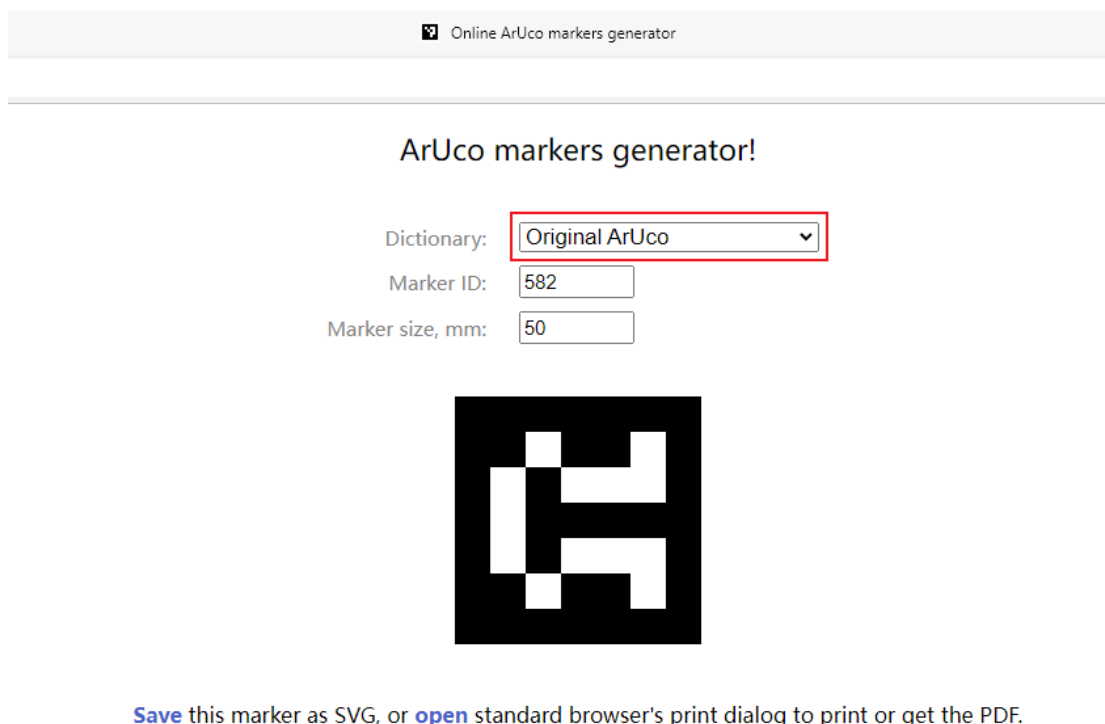
3.4 打印 Marker 标签

打印标定需要使用到的 Marker 标签，可以在 aruco_ros 包中找到提供好的 marker 标签进行打印，如图：



也可以从下面的网站下载 marker 标签并打印出来:

[Online ArUco markers generator \(chev.me\)](http://OnlineArUcoMarkersGenerator.com)



注意:

- Dictionary 一定要选 Original ArUco
- Marker ID 和 Marker size 自选, 在 launch 文件中做相应的修改(本教程演示使用 Marker ID: 582, Marker size: 50mm)
- 打印时, 要选择原始大小, 否则要测量一下打印出来的真实大小

4. aruco_ros 配置

4.1 简介

aruco 是一种类似二维码的定位标记辅助工具，通过在环境中部署 Markers，可以辅助机器人进行定位，弥补单一传感器的缺陷，纠正误差，本教程使用的手眼标定 easy_handeye 功能包需要借助这个工具进行手眼标定。

4.2 配置 aruco_ros 的 launch 文件

在 aruco_ros 功能包的 launch 目录下拷贝一份原有的 single.launch 文件或新建一个 launch 文件，命名为 single_realsense.launch，执行以下命令（本教程默认用户 ROS 的工作空间为 cakin_ws，用户根据实际的环境进入相应路径进行操作）：

```
cd ~/catkin_ws/src/aruco_ros/aruco_ros/launch
touch single_realsense.launch
gedit single_realsense.launch
```

根据以下内容进行修改或者覆盖为以下内容后保存：

```
<launch>

  <arg name="markerId"          default="582"/>
  <arg name="markerSize"        default="0.05"/>    <!-- in m -->
  <arg name="eye"                default="left"/>
  <arg name="marker_frame"       default="aruco_marker_frame"/>
  <arg name="ref_frame"          default="camera_color_frame"/>    <!-- leave empty and the pose will be
published wrt param parent_name -->
  <arg name="corner_refinement" default="LINES" />    <!-- NONE, HARRIS, LINES, SUBPIX -->

  <node pkg="aruco_ros" type="single" name="aruco_single">
    <remap from="/camera_info" to="/camera/color/camera_info" />
    <remap from="/image" to="/camera/color/image_raw" />
    <param name="image_is_rectified" value="True"/>
    <param name="marker_size"        value="$(arg markerSize)"/>
    <param name="marker_id"          value="$(arg markerId)"/>
    <param name="reference_frame"     value="$(arg ref_frame)"/>    <!-- frame in which the marker
pose will be refered -->
    <param name="camera_frame"       value="camera_color_frame"/>
    <param name="marker_frame"       value="$(arg marker_frame)" />
    <param name="corner_refinement"  value="$(arg corner_refinement)" />
  </node>

</launch>
```



```

<launch>
  <arg name="markerId" default="582"/>
  <arg name="markerSize" default="0.05"/> <!-- in m -->
  <arg name="eye" default="left"/>
  <arg name="marker_frame" default="aruco_marker_frame"/>
  <arg name="ref_frame" default="camera_color_frame"/> <!-- Leave empty and the pose will be published wrt param parent_name -->
  <arg name="corner_refinement" default="LINES"/> <!-- NONE, HARRIS, LINES, SUBPIX -->

  <node pkg="aruco_ros" type="single" name="aruco_single">
    <remap from="/camera_info" to="/camera_color/camera_info" />
    <remap from="/image" to="/camera_color/image_raw" />
    <param name="image_is_rectified" value="True"/>
    <param name="marker_size" value="$(arg markerSize)"/>
    <param name="marker_id" value="$(arg markerId)"/>
    <param name="reference_frame" value="$(arg ref_frame)"/> <!-- frame in which the marker pose will be referred -->
    <param name="camera_frame" value="camera_color_frame"/>
    <param name="marker_frame" value="$(arg marker_frame)"/>
    <param name="corner_refinement" value="$(arg corner_refinement)"/>
  </node>
</launch>

```

- **markerId**: 使用的 Marker 标签的 ID，这里在准备时打印的 Marker 标签 ID 为 582
- **markerSize**: Marker 标签的实际大小，单位为 m，这里在准备时打印的 Marker 标签实际大小为 5 厘米即 0.05m
- **ref_frame**: 参考坐标系名称，这里选择 Realsense 的 camera_color_frame 作为参考坐标系
- **<remap from="/camera_info" to="/camera_color/camera_info" />**: 将/camera_info 重映射为对应 Realsense 实际发布的相应的 Topic 即/camera_color/camera_info
- **<remap from="/image" to="/camera_color/image_raw" />**: 将/image 重映射为对应 Realsense 实际发布的相应的 Topic 即/camera_color/image_raw
- **camera_frame**: 相机坐标系，修改为实际的相机坐标系 camera_color_frame

5. easy_handeye 配置

5.1 简介

easy_handeye 是用于手眼标定的功能包之一，借助它可以实现眼在手上和眼在手外的手眼标定，在这里我们介绍使用 easy_handeye 完成眼在手上的手眼标定。

5.2 配置 easy_handeye 标定的 launch 文件

在 easy_handeye 功能包的 launch 目录下新建一个 launch 文件，命名为 eye_in_hand_calibrate.launch，执行以下命令（本教程默认用户 ROS 的工作空间为 cakin_ws，用户根据实际的环境进入相应路径进行操作）：

```

cd ~/catkin_ws/src/easy_handeye/easy_handeye/launch
touch eye_in_hand_calibrate.launch
gedit eye_in_hand_calibrate.launch

```

编辑输入以下内容并保存：

```

<?xml version="1.0" ?>
<launch>
  <!-- 生成标定文件的名称 -->
  <arg name="namespace_prefix" default="rm_rs_d435" />

```

RM 机械臂与 Realsense D435 手眼标定教程

```
<!-- RM 机械臂 MoveIt!配置的 move_group 为 arm，所以修改为 arm -->
<arg name="move_group" default="arm" />

<!-- start easy_handeye -->
<include file="$(find easy_handeye)/launch/calibrate.launch" >
  <arg name="namespace_prefix" value="$(arg namespace_prefix)" />
  <arg name="move_group" value="$(arg move_group)" />
  <!-- 这里使用眼在手上的方式进行标定，所以此处改成 true -->
  <arg name="eye_on_hand" value="true" />

  <!--tracking_base_frame 为 realsense 的相机坐标系-->
  <arg name="tracking_base_frame" value="camera_color_frame" />
  <!--tracking_marker_frame 对应 aruco_ros 包中 single_realsense.launch 中的 marker_frame 的值-->
  <arg name="tracking_marker_frame" value="aruco_marker_frame" />
  <!--robot_base_frame 为机器人基座坐标系-->
  <arg name="robot_base_frame" value="base_link" />
  <!--robot_effector_frame 为工具坐标系，如夹爪，吸盘等，但实际的 rm 机器人模型未添加夹爪，
  所以这里设置为末端关节 Link7-->
  <arg name="robot_effector_frame" value="Link7" />

  <arg name="freehand_robot_movement" value="false" />
  <arg name="robot_velocity_scaling" value="0.5" />
  <arg name="robot_acceleration_scaling" value="0.2" />
</include>
</launch>
```

注意: `<arg name="robot_effector_frame" value="Link7" />`这里是以 RM75 即 7 轴机械臂为例所以为 Link7，如果使用的是 RM65 即 6 轴机械臂，则应该修改为 Lin6

5.3 配置发布 TF 的 launch 文件

手眼标定完成后需要根据标定的文件发布 TF 坐标转换才能够将相机识别到物体的坐标转换到相应机械臂的坐标系上，修改 easy_handeye 功能包的 launch 目录下的 publish.launch 文件，执行以下命令（本教程默认用户 ROS 的工作空间为 cakin_ws，用户根据实际的环境进入相应路径进行操作）：

```
cd ~/catkin_ws/src/easy_handeye/easy_handeye/launch
gedit publish.launch
```

编辑修改对应的内容并保存：

```
<?xml version="1.0"?>
<launch>
  <!--修改 eye_on_hand 参数默认为 true -->
  <arg name="eye_on_hand" doc="eye-on-hand instead of eye-on-base" default="true" />
  <!-- 修改 namespace_prefix 参数，与眼在手上标定 launch 文件[eye_in_hand_calibrate.launch]中的
  “namespace_prefix”一致，这样才能找到标定好的 YAML 文件-->
  <arg name="namespace_prefix" default="rm_rs_d435" />
  <arg if="$(arg eye_on_hand)" name="namespace" value="$(arg namespace_prefix)_eye_on_hand" />
```

```
<arg unless="$(arg eye_on_hand)" name="namespace" value="$(arg namespace_prefix)_eye_on_base" />

<!--it is possible to override the link names saved in the yaml file in case of name clashes, for example-->
<arg if="$(arg eye_on_hand)" name="robot_effector_frame" default="" />
<arg unless="$(arg eye_on_hand)" name="robot_base_frame" default="" />
<arg name="tracking_base_frame" default="" />

<arg name="inverse" default="false" />
<arg name="calibration_file" default="" />

<!--publish hand-eye calibration-->
<group ns="$(arg namespace)">
  <param name="eye_on_hand" value="$(arg eye_on_hand)" />
  <param unless="$(arg eye_on_hand)" name="robot_base_frame" value="$(arg robot_base_frame)" />
  <param if="$(arg eye_on_hand)" name="robot_effector_frame" value="$(arg robot_effector_frame)" />
  <param name="tracking_base_frame" value="$(arg tracking_base_frame)" />
  <param name="inverse" value="$(arg inverse)" />
  <param name="calibration_file" value="$(arg calibration_file)" />
  <node name="$(anon handeye_publisher)" pkg="easy_handeye" type="publish.py" output="screen"/>
</group>
</launch>
```

```
<?xml version="1.0"?>
<launch>
  <!--修改eye_on_hand参数默认为true -->
  <arg name="eye_on_hand" doc="eye-on-hand instead of eye-on-base" default="true" />
  <!--修改namespace_prefix参数, 与眼在手上标定launch文件[eye_in_hand_calibrate.launch]中的"namespace_prefix"一致, 这样才能找到标定好的YAML文件-->
  <arg name="namespace_prefix" default="rm_rs_d435" />
  <arg if="$(arg eye_on_hand)" name="namespace" value="$(arg namespace_prefix)_eye_on_hand" />
  <arg unless="$(arg eye_on_hand)" name="namespace" value="$(arg namespace_prefix)_eye_on_base" />

  <!--it is possible to override the link names saved in the yaml file in case of name clashes, for example-->
  <arg if="$(arg eye_on_hand)" name="robot_effector_frame" default="" />
  <arg unless="$(arg eye_on_hand)" name="robot_base_frame" default="" />
  <arg name="tracking_base_frame" default="" />

  <arg name="inverse" default="false" />
  <arg name="calibration_file" default="" />

  <!--publish hand-eye calibration-->
  <group ns="$(arg namespace)">
    <param name="eye_on_hand" value="$(arg eye_on_hand)" />
    <param unless="$(arg eye_on_hand)" name="robot_base_frame" value="$(arg robot_base_frame)" />
    <param if="$(arg eye_on_hand)" name="robot_effector_frame" value="$(arg robot_effector_frame)" />
    <param name="tracking_base_frame" value="$(arg tracking_base_frame)" />
    <param name="inverse" value="$(arg inverse)" />
    <param name="calibration_file" value="$(arg calibration_file)" />
    <node name="$(anon handeye_publisher)" pkg="easy_handeye" type="publish.py" output="screen"/>
  </group>
</launch>
```

6. 启动相关 launch 文件开始标定

6.1 启动 Realsense 节点

首先确认相关 ROS 包都已编译，执行如下命令确保能够编译成功：

```
cd ~/catkin_ws
catkin build
```

```
realman@realman-laptop: ~/catkin_ws
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
Finished <<< rm_75_description [ 0.2 seconds ]
Finished <<< rm_75_gazebo [ 0.2 seconds ]
Starting >>> rm_75_moveit_config
Finished <<< rm_75_moveit_config [ 0.2 seconds ]
Starting >>> rqt_easy_handeye
Finished <<< rm_75_msgs [ 2.2 seconds ]
Starting >>> rm_75_control
Finished <<< rm_75_control [ 0.3 seconds ]
Starting >>> rm_75_demo
Finished <<< rqt_easy_handeye [ 5.3 seconds ]
Starting >>> rm_75_driver
Finished <<< rm_75_driver [ 0.3 seconds ]
Finished <<< aruco_ros [ 11.8 seconds ]
Finished <<< rm_75_demo [ 18.1 seconds ]
Starting >>> rm_75_robot
Finished <<< rm_75_robot [ 2.1 seconds ]

[build] Summary: All 17 packages succeeded!
[build] Ignored: None.
[build] Warnings: None.
[build] Abandoned: None.
[build] Failed: None.
[build] Runtime: 26.1 seconds total.
[build] Note: Workspace packages have changed, please re-source setup files to use them.
realman@realman-laptop:~/catkin_ws$ source devel/setup.bash
```

确认相机通过 Type-C 数据线将相机与主机 USB3.0 接口相连，执行如下命令启动 Realsense 节点：

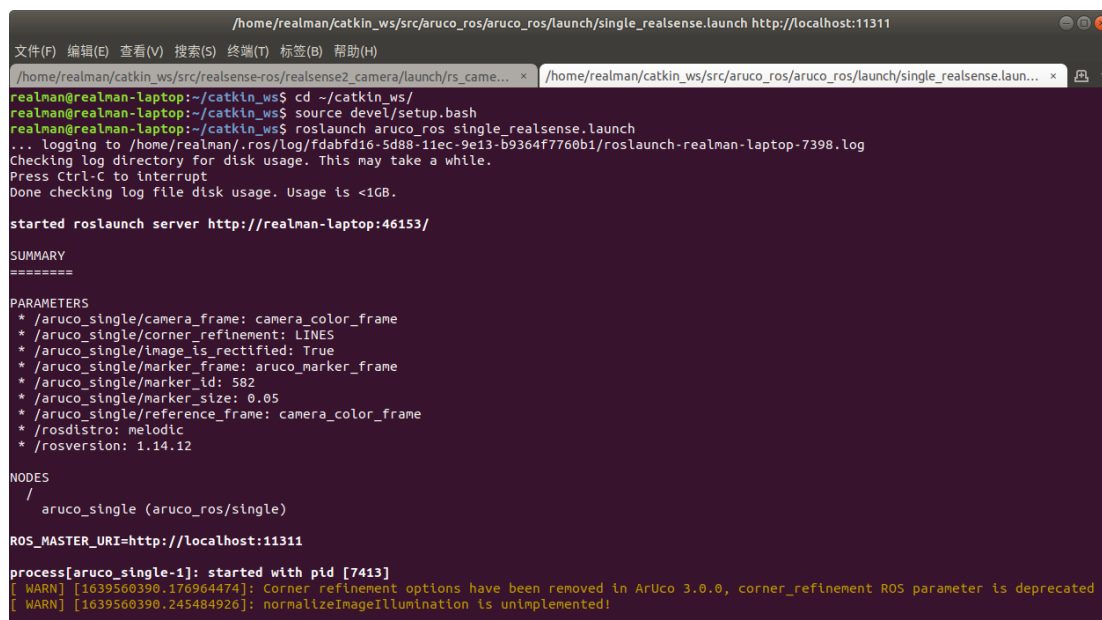
```
cd ~/catkin_ws
source devel/setup.bash
roslaunch realsense2_camera rs_camera.launch
```

```
/home/realman/catkin_ws/src/realsense-ros/realsense2_camera/launch/rs_camera.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[INFO] [1639471966.582111583]: Built with LibRealSense v2.50.0
[INFO] [1639471966.582154314]: Running with LibRealSense v2.50.0
[INFO] [1639471966.608428472]:
[INFO] [1639471966.614174509]: Device with serial number 137322078962 was found.
[INFO] [1639471966.614247944]: Device with physical ID /sys/devices/pci0000:00/0000:00:14.0/usb3/3-1/3-1:1.0/video4linux/vide
bo0 was found.
[INFO] [1639471966.614293765]: Device with name Intel RealSense D435 was found.
[INFO] [1639471966.614935268]: Device with port number 3-1 was found.
[INFO] [1639471966.614991985]: Device USB type: 3.2
[INFO] [1639471966.618542300]: getParameters...
[INFO] [1639471966.672613411]: SetupDevice...
[INFO] [1639471966.672663406]: JSON file is not provided
[INFO] [1639471966.672699541]: ROS Node Namespace: camera
[INFO] [1639471966.672738873]: Device Name: Intel RealSense D435
[INFO] [1639471966.672779843]: Device Serial No: 137322078962
[INFO] [1639471966.672797659]: Device physical port: /sys/devices/pci0000:00/0000:00:14.0/usb3/3-1/3-1:1.0/video4linux/video
[INFO] [1639471966.672826760]: Device FW version: 05.13.00.50
[INFO] [1639471966.672855684]: Device Product ID: 0x0B07
[INFO] [1639471966.672880771]: Enable PointCloud: Off
[INFO] [1639471966.672902753]: Align Depth: Off
[INFO] [1639471966.672941290]: Sync Mode: Off
[INFO] [1639471966.673001089]: Device Sensors:
[INFO] [1639471966.675687660]: Stereo Module was found.
[INFO] [1639471966.681077192]: RGB Camera was found.
[INFO] [1639471966.681142596]: (Confidence, 0) sensor isn't supported by current device! -- Skipping...
[INFO] [1639471966.681200985]: num_filters: 0
[INFO] [1639471966.681241109]: Setting Dynamic reconfig parameters.
[INFO] [1639471966.732640846]: Done Setting Dynamic reconfig parameters.
[INFO] [1639471966.735070792]: depth stream is enabled - width: 848, height: 480, fps: 30, Format: Z16
[INFO] [1639471966.735891283]: color stream is enabled - width: 640, height: 480, fps: 30, Format: RGB8
[INFO] [1639471966.735960930]: setupPublishers...
[INFO] [1639471966.739830103]: Expected frequency for depth = 30.00000
[INFO] [1639471966.786686451]: Expected frequency for color = 30.00000
[INFO] [1639471966.808855024]: setupStreams...
[INFO] [1639471966.936163645]: SELECTED BASE:Depth, 0
[INFO] [1639471966.954314706]: RealSense Node Is Up!
[WARN] [1639471967.007367169]:
```

6.2 启动 aruco 节点识别 Marker 标签

打开一个新的终端，执行以下命令启动 single_realsense.launch:

```
cd ~/catkin_ws
source dev/setup.bash
roslaunch aruco_ros single_realsense.launch
```



```
/home/realman/catkin_ws/src/aruco_ros/aruco_ros/launch/single_realsense.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
/home/realman/catkin_ws/src/realsense-ros/realsense2_camera/launch/rs_came... x /home/realman/catkin_ws/src/aruco_ros/aruco_ros/launch/single_realsense.laun... x
realman@realman-laptop:~/catkin_ws$ cd ~/catkin_ws/
realman@realman-laptop:~/catkin_ws$ source devel/setup.bash
realman@realman-laptop:~/catkin_ws$ roslaunch aruco_ros single_realsense.launch
... logging to /home/realman/.ros/log/fdabfd16-5d88-11ec-9e13-b9364f7760b1/roslaunch-realman-laptop-7398.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://realman-laptop:46153/

SUMMARY
=====
PARAMETERS
 * /aruco_single/camera_frame: camera_color_frame
 * /aruco_single/corner_refinement: LINES
 * /aruco_single/image_is_rectified: True
 * /aruco_single/marker_frame: aruco_marker_frame
 * /aruco_single/marker_id: 582
 * /aruco_single/marker_size: 0.05
 * /aruco_single/reference_frame: camera_color_frame
 * /rostdistro: melodic
 * /rosversion: 1.14.12

NODES
 /
   aruco_single (aruco_ros/single)

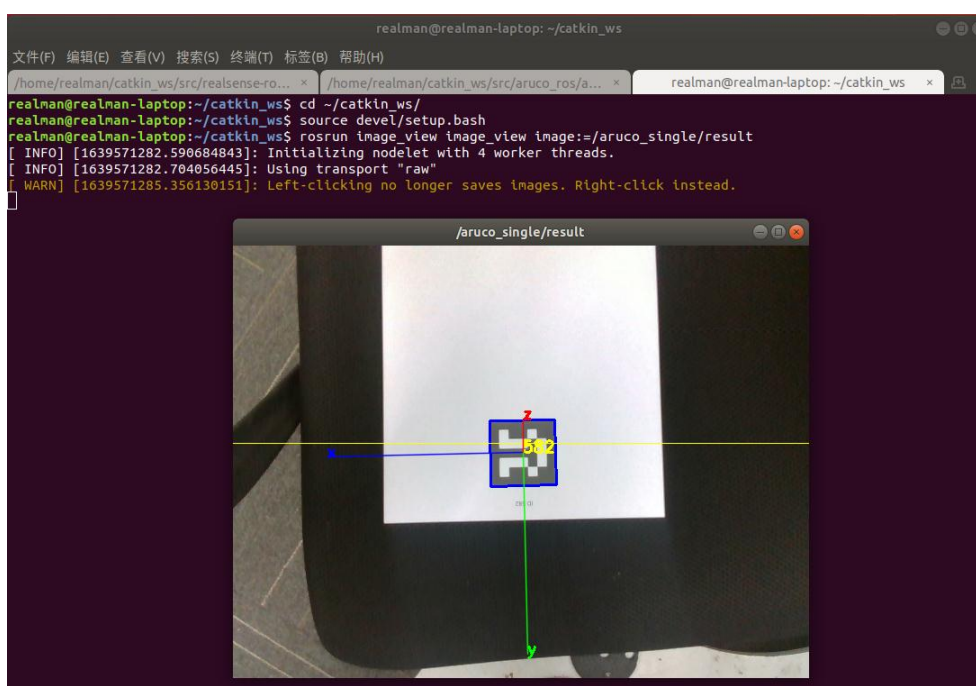
ROS_MASTER_URI=http://localhost:11311

process[aruco_single-1]: started with pid [7413]
[ WARN] [1639560390.176964474]: Corner refinement options have been removed in ArUco 3.0.0, corner_refinement ROS parameter is deprecated
[ WARN] [1639560390.245484926]: normalizeImageIllumination is unimplemented!
```

6.3 启动 image_view 节点显示图像

打开一个新的终端，执行以下命令启动 image_view 订阅/aruco_single/result 显示图像:

```
cd ~/catkin_ws
source dev/setup.bash
roslaunch image_view image_view image:=/aruco_single/result
```



6.4 启动 RM 机械臂的 control 和 driver 节点

打开一个新的终端，执行以下命令启动 control 节点：

```
cd ~/catkin_ws
source dev/setup.bash
roslaunch rm_75_control rm_75_control.launch
```

注意：本教程以 RM75 机械臂为例，如果使用的是 6 轴即 RM65 机械臂，则启动 control 的命令为：roslaunch rm_control rm_control.launch

```
/home/realman/catkin_ws/src/rm_75_robot/rm_75_control/launch/rm_75_control.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
/home/realman/catkin_ws/src... x /home/realman/catkin_ws/src... x realman@realman-laptop: ~/c... x /home/realman/catkin_ws/src... x
realman@realman-laptop:~/catkin_ws$ cd ~/catkin_ws/
realman@realman-laptop:~/catkin_ws$ source devel/setup.bash
realman@realman-laptop:~/catkin_ws$ roslaunch rm_75_control rm_75_control.launch
... logging to /home/realman/.ros/log/4124c866-5da2-11ec-9e13-b9364f7760b1/roslaunch-realman-laptop-10011.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://realman-laptop:38743/

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.12

NODES
/
  rm_75_control (rm_75_control/rm_control)

ROS_MASTER_URI=http://localhost:11311

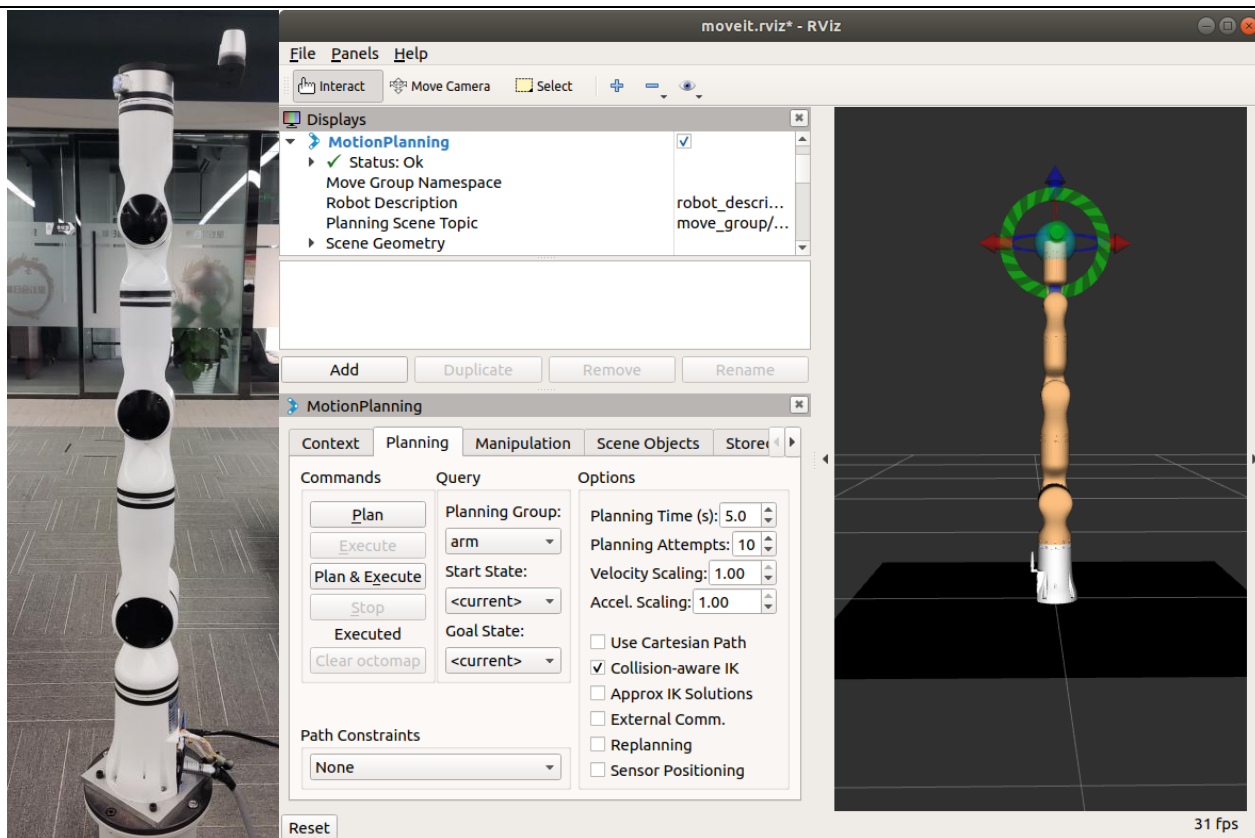
process[rm_75_control-1]: started with pid [10026]
```

再打开一个新的终端，执行以下命令启动 driver 节点：

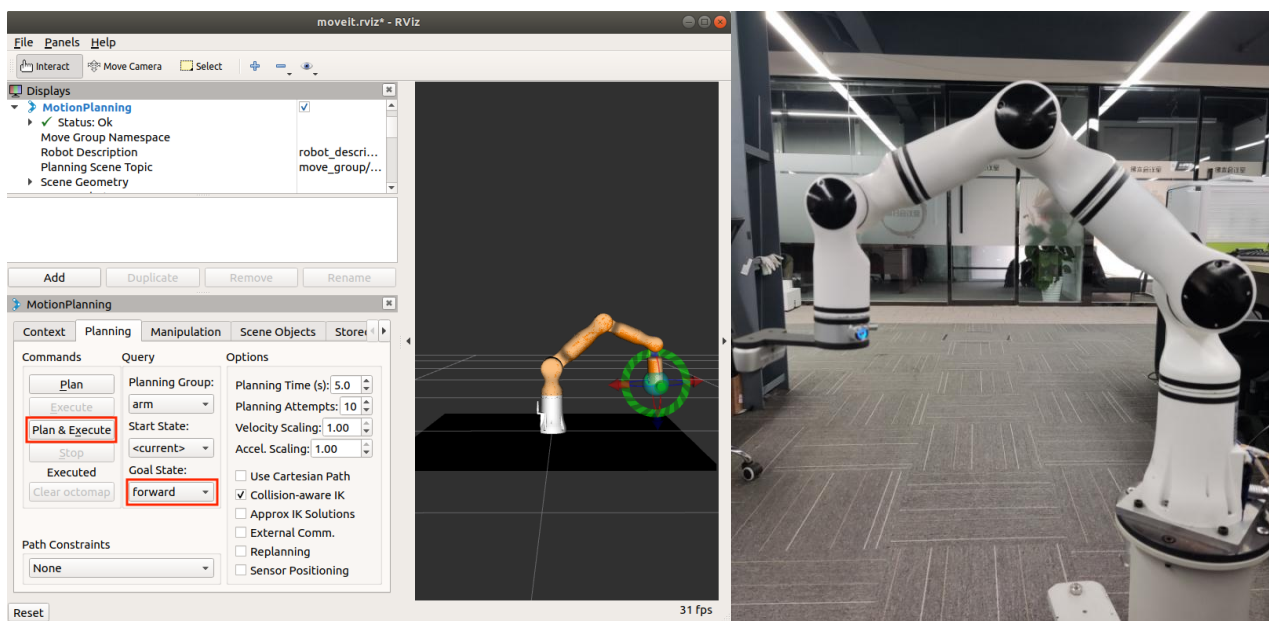
```
cd ~/catkin_ws
source dev/setup.bash
roslaunch rm_75_bringup rm_robot.launch
```

注意：本教程以 RM75 机械臂为例，如果使用的是 6 轴即 RM65 机械臂，则启动 driver 的命令为：roslaunch rm_bringup rm_robot.launch

运行成功后在 rviz 中可以看到机器人模型与真实机械臂的状态保持一致，如图所示：

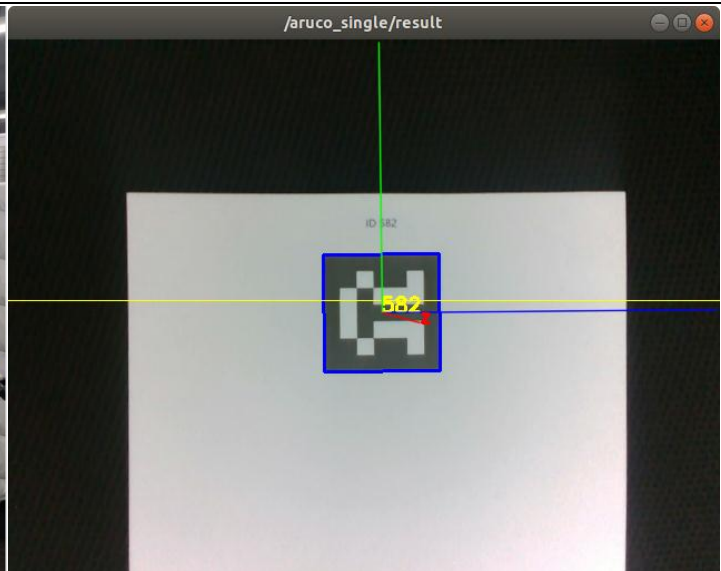
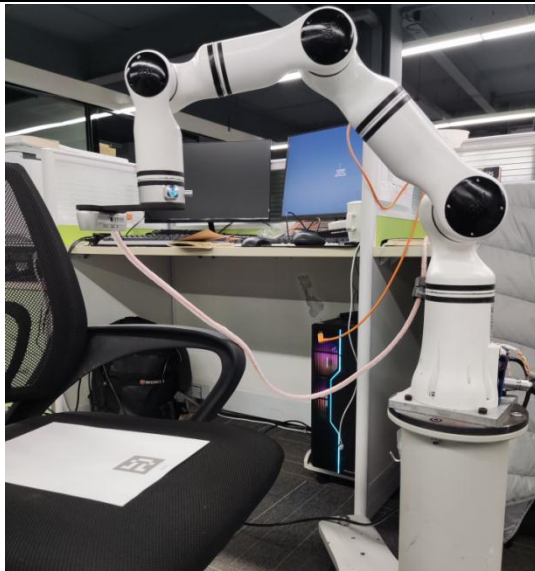


在 rviz 的 MotionPlanning 下的 Planning, Goal State 选择 forward 然后点击 Plan & Execute, 可以看到真实机械臂会按照 rviz 中 MoveIt! 规划的路径运动, 如图所示:



6.5 启动 easy_handeye 手眼标定节点

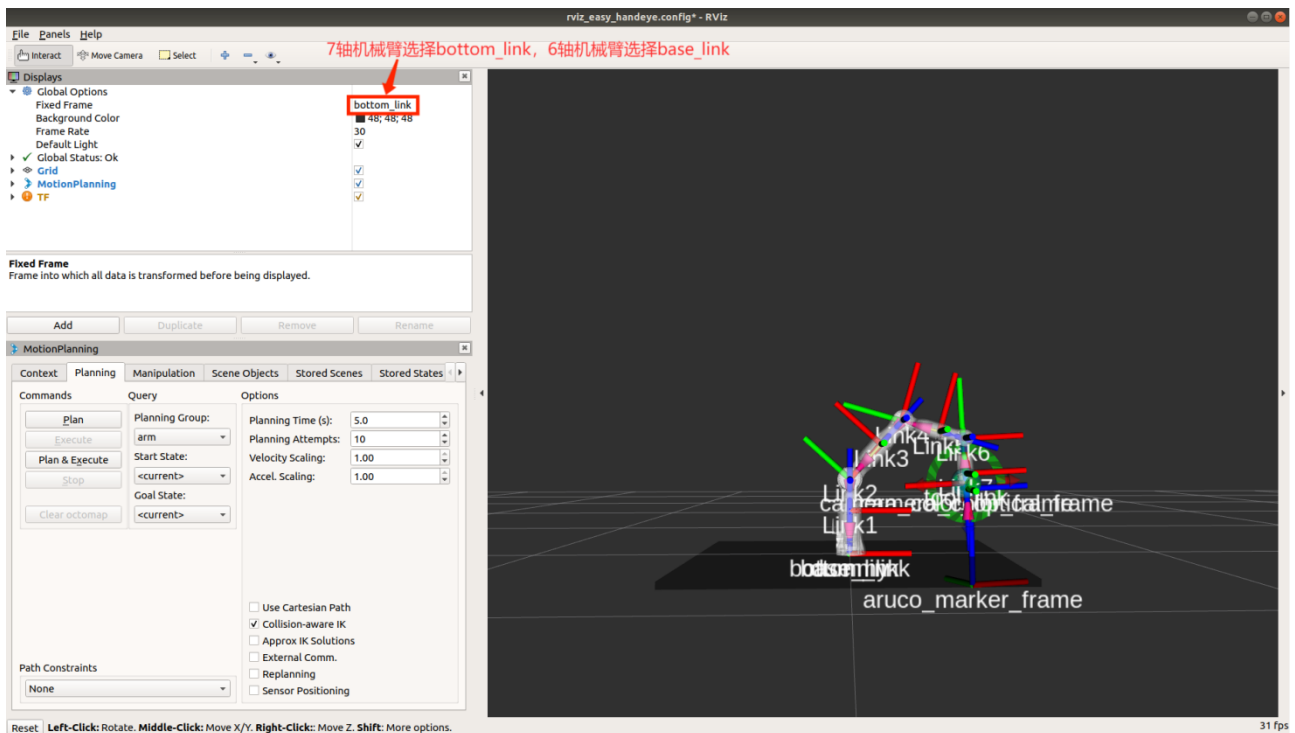
在启动标定程序前, 建议先在 rviz 中通过 Moveit 或手动调节机械臂到一个合适的姿态, 然后将 aruco 二维码移动至相机视野中心处附近, 如图所示 (建议参考图中的姿态, 避免在标定时有不可达的规划):



打开一个新的终端，执行以下命令启动手眼标定节点：

```
cd ~/catkin_ws
source dev/setup.bash
roslaunch easy_handeye eye_in_hand_calibrate.launch
```

成功运行后，会同时打开三个界面：



界面 1

rqt_easy_handeye__Hand-eye Calibration - rqt

Form

Info

Name: /rm_rs_d435_eye_on_hand/

Type: eye on hand

Tracking Base Frame: camera_color_frame

Tracking Marker Frame: aruco_marker_frame

Robot Base Frame: base_link

Robot Effector Frame: Link7

Calibration algorithm: OpenCV/Tsai-Lenz

Actions

Take Sample

Remove Sample

Compute

Save

Samples

Result

界面 2

rqt_easy_handeye__Hand-eye Calibration automatic movement - rqt

Local Mover

0/0

No plan yet

Hello

Check starting pose

Next Pose

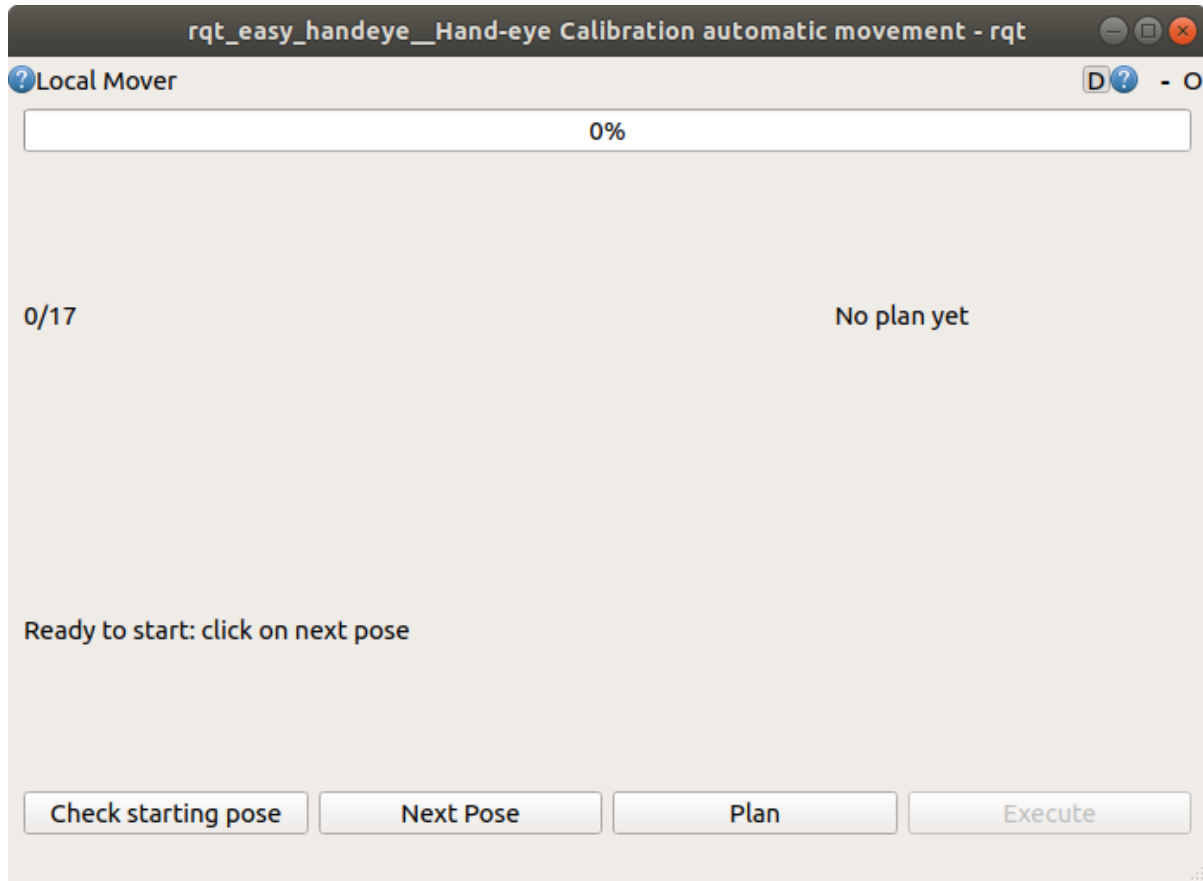
Plan

Execute

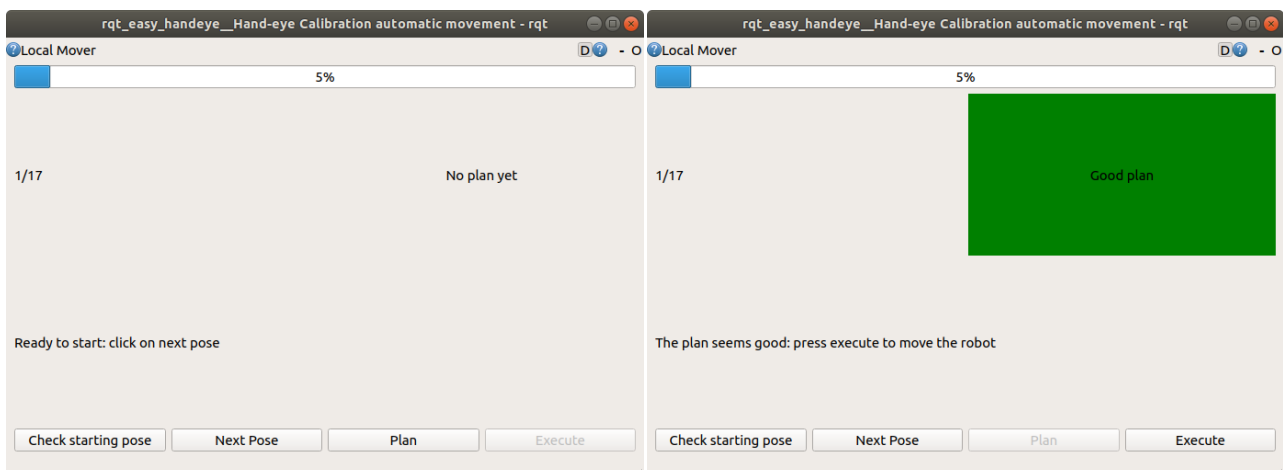
界面 3

标定步骤：

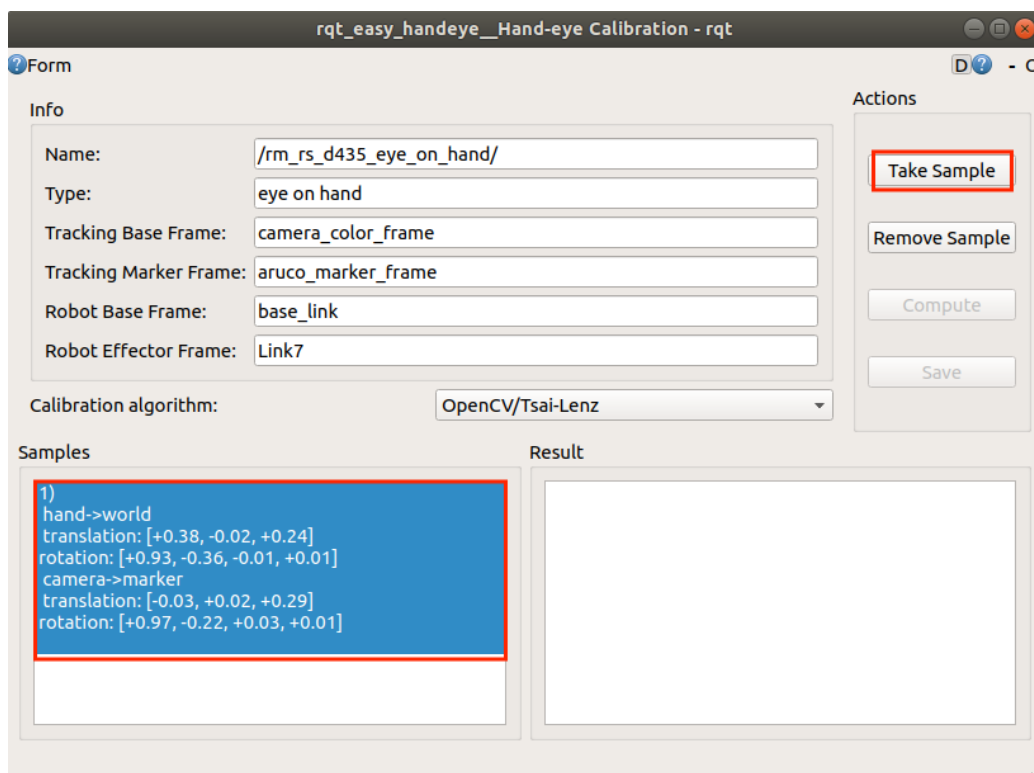
- 1) 在界面 3 中，点击 Check starting pose，稍微等待，若检查成功，界面会出现：0/17，Ready to start，如图所示（如果检测失败或者没有出现 0/17，则说明当前机械臂的姿态不合适，可能有规划不可达，需要关闭标定节点然后调节机械臂到一个合适姿态并将 aruco 二维码移动至相机视野中心处附近再重启启动标定节点继续标定）：



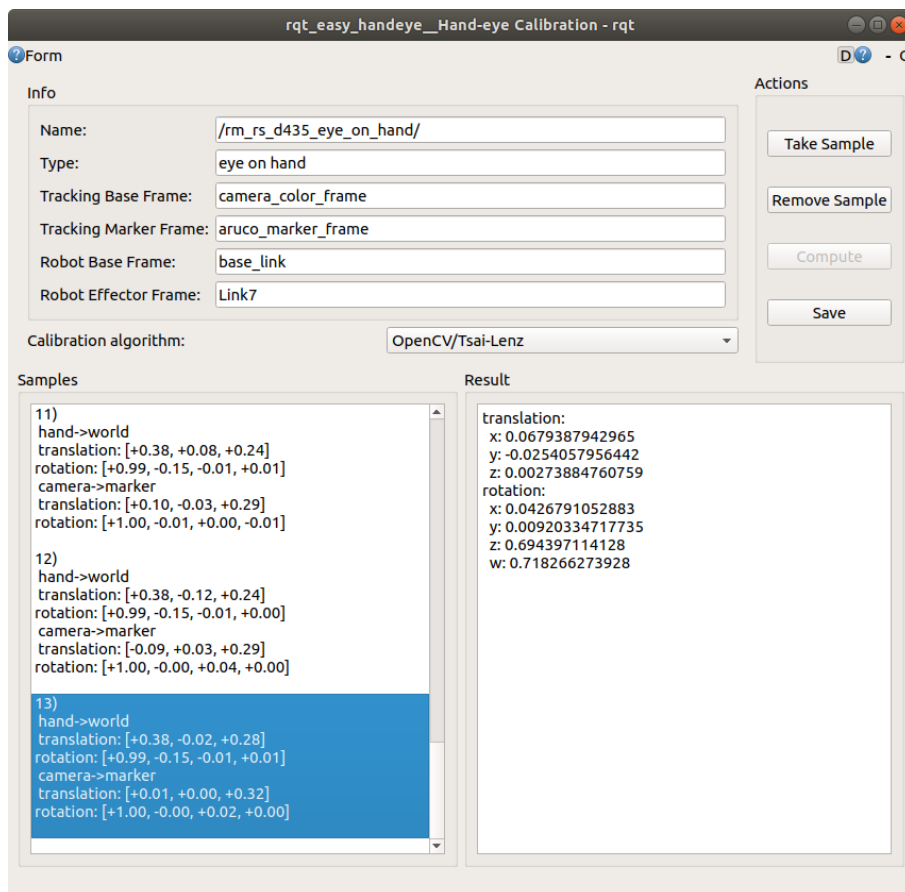
- 2) 界面 3 中依次点击 Next Pose, Plan, Execute，机械臂会移动至新的位置，若二维码完全的在相机视野范围内，且能检测成功，则进行下一步，若二维码不在相机视野范围内或部分在视野外，检测不到，则再次依次点击 Next Pose, Plan, Execute，直到二维码完全在相机视野范围内，且能检测成功，再进行下一步（一般前 4 次会依据 aruco 二维码位置机械臂移动到上下左右位置，二维码都不完全在相机视野范围内，从第五次开始检测成功），如图所示：



- 3) 界面 2 中点击 Take Sample，若 Samples 对话框中出现有效信息，说明第一个点标定成功



- 4) 重复执行步骤 2 和步骤 3，直至 17 个点全部标定完毕
- 5) 界面 2 中点击 Compute，则 Result 对话框中会出现结果



6) 界面 2 中 Save, 会将结果保存为一个 YAML 文件, 路径为 `~/ros/easy_handeye`

```
realman@realman-laptop: ~/ros/easy_handeye
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
realman@realman-laptop:~$ cd ~/ros/easy_handeye/
realman@realman-laptop:~/ros/easy_handeye$ ls
rm_rs_d435_eye_on_hand.yaml
realman@realman-laptop:~/ros/easy_handeye$ cat rm_rs_d435_eye_on_hand.yaml
parameters:
  eye_on_hand: true
  freehand_robot_movement: false
  move_group: arm
  move_group_namespace: /
  namespace: /rm_rs_d435_eye_on_hand/
  robot_base_frame: base_link
  robot_effector_frame: Link7
  tracking_base_frame: camera_color_frame
  tracking_marker_frame: aruco_marker_frame
transformation:
  qw: 0.7182662739282054
  qx: 0.042679105288303695
  qy: 0.009203347177354382
  qz: 0.6943971141283752
  x: 0.06793879429646016
  y: -0.02540579564417963
  z: 0.0027388476075899167
realman@realman-laptop:~/ros/easy_handeye$
```

标定完成后在每个终端按 Ctrl+C 即可结束运行节点。

6.6 简单测试

按照 6.1 至 6.4 启动 Realsense、aruco、image_view、rm_control、rm_driver 相关节点, 不需要再启动 easy_handeye 手眼标定节点, 然后再打开一个新的终端, 执行以下命令发布手眼标定后的 TF:

```
roslaunch easy_handeye publish.launch
```

```
/home/realman/catkin_ws/src/easy_handeye/easy_handeye/launch/publish.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
/home/realman/... x /home/realman/... x realman@real... x /home/realman/... x /home/realman/... x /home/realman/... x
realman@realman-laptop:~/catkin_ws$ roslaunch easy_handeye publish.launch
... logging to /home/realman/.ros/log/b9a2478a-5da9-11ec-9e13-b9364f7760b1/roslaunch-realman-laptop-14505.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://realman-laptop:41179/

SUMMARY
=====
PARAMETERS
* /rm_rs_d435_eye_on_hand/calibration_file:
* /rm_rs_d435_eye_on_hand/eye_on_hand: True
* /rm_rs_d435_eye_on_hand/inverse: False
* /rm_rs_d435_eye_on_hand/robot_effector_frame:
* /rm_rs_d435_eye_on_hand/tracking_base_frame:
* /rostdistro: melodic
* /rosversion: 1.14.12

NODES
  /rm_rs_d435_eye_on_hand/
    handeye_publisher_realman_laptop_14505_503373985637918066 (easy_handeye/publish.py)

ROS_MASTER_URI=http://localhost:11311

process[rm_rs_d435_eye_on_hand/handeye_publisher_realman_laptop_14505_503373985637918066-1]: started with pid [14520]
[INFO] [1639576159.067217]: Loading the calibration from file: /home/realman/.ros/easy_handeye/rm_rs_d435_eye_on_hand.yaml
[INFO] [1639576159.084798]: loading calibration parameters into namespace /rm_rs_d435_eye_on_hand/
[INFO] [1639576159.087175]: Storing calibration /rm_rs_d435_eye_on_hand/ into the parameters server
[INFO] [1639576159.095454]: Storing parameters for calibration /rm_rs_d435_eye_on_hand/ into the parameters server
```

执行以下命令可以查看相机与机械臂基座标系的 TF 转换：

```
roslaunch tf_echo /base_link /camera_color_frame
```

```

realman@realman-laptop: ~/catkin_ws
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
/home/real... x /home/real... x realman@re... x /home/real... x /home/real... x /home/real... x realman@re... x
realman@realman-laptop:~/catkin_ws$ source devel/setup.bash
realman@realman-laptop:~/catkin_ws$ roslaunch tf_echo
build/ devel/ logs/ src/
realman@realman-laptop:~/catkin_ws$ roslaunch tf_echo /base_link /camera_color_frame
At time 1639576250.710
- Translation: [0.448, -0.011, 0.272]
- Rotation: in Quaternion [-0.604, 0.797, -0.012, 0.030]
            in RPY (radian) [-3.087, 0.033, -1.844]
            in RPY (degree) [-176.853, 1.888, -105.650]
At time 1639576251.110
- Translation: [0.448, -0.011, 0.272]
- Rotation: in Quaternion [-0.604, 0.797, -0.012, 0.030]
            in RPY (radian) [-3.087, 0.033, -1.844]
            in RPY (degree) [-176.853, 1.888, -105.650]
At time 1639576251.910
- Translation: [0.448, -0.011, 0.272]
- Rotation: in Quaternion [-0.604, 0.797, -0.012, 0.030]
            in RPY (radian) [-3.087, 0.033, -1.844]
            in RPY (degree) [-176.853, 1.888, -105.650]
At time 1639576253.110
- Translation: [0.448, -0.011, 0.272]
- Rotation: in Quaternion [-0.604, 0.797, -0.012, 0.030]
            in RPY (radian) [-3.087, 0.033, -1.844]
            in RPY (degree) [-176.853, 1.888, -105.650]
^Crealman@realman-laptop:~/catkin_ws$

```

执行以下命令如果检测到 marker 标签，会显示 marker 在相机坐标系下的位姿：

```
rostopic echo /aruco_single/pose
```

```

realman@realman-laptop: ~/catkin_ws
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
/home/rea... x /home/rea... x realman@... x /home/rea... x /home/rea... x /home/rea... x realman@... x realman@... x
stamp:
  secs: 1639576317
  nsecs: 593172550
frame_id: "camera_color_frame"
pose:
  position:
    x: 0.00583826657385
    y: 0.00115117337555
    z: 0.323438853025
  orientation:
    x: 0.999895356589
    y: -0.00408941997197
    z: 0.0136437827652
    w: -0.00252976440596
---
header:
  seq: 53002
  stamp:
    secs: 1639576317
    nsecs: 626530647
  frame_id: "camera_color_frame"
pose:
  position:
    x: 0.00584393972531
    y: 0.00113184936345
    z: 0.323426783085
  orientation:
    x: 0.999877054903
    y: -0.00429952109633
    z: 0.0150793679583
    w: 4.31141540931e-05
---
^Crealman@realman-laptop:~/catkin_ws$

```

标定过程可能会受相关硬件及环境等条件因素的影响，所以标定结果可能存在一定的误差，用户标定完成后可以通过获取物体在相机坐标系下的位姿，然后通过 TF 转换到机械臂下的位姿进行测试，存在误差可以后续在程序进行补偿计算。