## 1. RM-65B ROS包架构简析

rm\_65\_description 描述RM-65机械臂物理结构的URDF和XACRO文件以及模型、配置文件等  
rm\_bringup 启动ROS节点的launch文件存放处  
rm\_driver 控制RM-65机械臂的ROS驱动节点，包括和机械臂建立以太网Socket通讯、更新机械臂全局状态等  
rm\_msgs RM-65包使用的ROS控制、状态消息定义  
rm\_control RM-65的高级运动控制器，将接收到的moveit机械臂轨迹作三次样条插值，按默认50HZ频率发送给rm\_driver  
rm\_gazebo RM-65机械臂在Gazebo仿真环境下的模拟所用到的参数和文件配置  
rm\_65\_moveit\_config 为RM-65机械臂提供的ROS MoveIt！运动规划框架的配置文件及简单的演示demo  
rm\_65\_demo RM-65机器人部分demo的存放位置，包括Moveit的编程案例，如场景规划、避障、pick and place等  
rm\_install(TODO) RM-65机器人的环境配置脚本

## 2. 编译与安装及过程问题

### 1. 系统环境要求

经过睿尔曼测试的操作系统版本： Ubuntu 20.04

经过睿尔曼测试的ROS版本： ROS1 Noetic

未经过测试的操作系统版本与ROS版本，用户根据需求，可自行尝试可行性

### 2.ROS1 Noetic安装

在机械臂ROS包的rm\_install 文件夹下打开终端，执行以下命令自动安装ROS1 Noetic并配置Ubuntu环境：

sudo bash ros1\_noetic\_install.sh # Install ROS1 Noetic

#!/bin/bash  
# Version: 1.4  
# Date: 2023-06-19  
# Author: Herman Ye @Realman Robotics  
#  
# Warning: This script is ONLY for ROS1 Noetic in ubuntu 20.04  
# set -x  
set -e  
# UBUNTU CONFIGURATION BEGINS HERE  
# Check if script is run as root (sudo)  
if [ "$(id -u)" != "0" ]; then  
    echo "This script must be run with sudo privileges. for example: sudo bash ros1\_noetic\_install.sh"  
    read -p "Press any key to exit..."  
    exit 1  
fi  
# Get script directory  
SCRIPT\_DIR=$(dirname "$0")  
# Get the username of the non-root user  
USERNAME=$SUDO\_USER  
echo "Current user is: $USERNAME"  
# Save logs to files  
LOG\_FILE="${SCRIPT\_DIR}/ros1\_noetic\_install.log"  
ERR\_FILE="${SCRIPT\_DIR}/ros1\_noetic\_install.err"  
rm -f ${LOG\_FILE}  
rm -f ${ERR\_FILE}  
# Redirect output to console and log files  
exec 1> >(tee -a ${LOG\_FILE} )  
exec 2> >(tee -a ${ERR\_FILE} >&2)  
# Output log info to console  
echo "ROS1 Noetic installation started!"   
echo "Installation logs will be saved to ${LOG\_FILE}"  
echo "Installation errors will be saved to ${ERR\_FILE}"  
# No Password sudo config  
sudo sed -i 's/^%sudo.\*/%sudo ALL=(ALL) NOPASSWD:ALL/g' /etc/sudoers  
# Get architecture of the system  
if [ $(uname -m) = "x86\_64" ]; then  
  MIRROR="https://mirrors.tuna.tsinghua.edu.cn/ubuntu/"  
else  
  MIRROR="https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/"  
fi  
echo "Current system architecture is: $(uname -m)"  
echo "Current mirror is: $MIRROR"  
# Backup original software sources  
sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup  
# Clear original software sources  
sudo echo "" > /etc/apt/sources.list  
# Replace software sources  
echo "deb $MIRROR focal main restricted universe multiverse" >> /etc/apt/sources.list  
echo "deb $MIRROR focal-updates main restricted universe multiverse" >> /etc/apt/sources.list  
echo "deb $MIRROR focal-backports main restricted universe multiverse" >> /etc/apt/sources.list  
if [ $(uname -m) = "x86\_64" ]; then  
  echo "deb http://security.ubuntu.com/ubuntu/ focal-security main restricted universe multiverse" >> /etc/apt/sources.list  
else  
  echo "deb http://ports.ubuntu.com/ubuntu-ports/ focal-security main restricted universe multiverse" >> /etc/apt/sources.list  
fi  
# System update  
sudo apt update  
sudo apt upgrade -y  
# Install pip  
sudo apt install python3-dev -y  
sudo apt install pip -y # If you haven't already installed pip  
# Install gnome-terminal  
sudo apt install gnome-terminal -y # If you haven't already installed gnome-terminal  
# Set default pip source  
pip config set global.index-url http://pypi.tuna.tsinghua.edu.cn/simple  
pip config set global.trusted-host pypi.tuna.tsinghua.edu.cn  
# ROS1 NOETIC INSTALLATION BEGINS HERE  
# Configure your Ubuntu repositories  
sudo add-apt-repository restricted  
sudo add-apt-repository universe  
sudo add-apt-repository multiverse  
# Setup your sources.list  
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'  
# Add the ROS key  
ros\_key="${SCRIPT\_DIR}/ros.key"  
rm -f "${ros\_key}"  
wget http://packages.ros.org/ros.key  
sudo apt-key add ros.key  
# Install catkin\_tools  
pip install -U catkin\_tools  
# Install dependencies  
sudo apt install libvtk7-jni libvtk7-java libvtk7-dev libvtk7-qt-dev libpcl-dev -y  
# Update the system packages index to the latest version  
sudo apt update  
# Install Curl  
sudo apt install curl -y # If you haven't already installed curl  
# Install ROS1 Noetic  
sudo apt install ros-noetic-desktop-full -y  
# Install dependencies  
sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential -y  
# Environment setup  
if ! grep -q "source /opt/ros/noetic/setup.bash" /home/$USERNAME/.bashrc; then  
    echo "# ROS1 Noetic Environment Setting" | sudo tee -a /home/$USERNAME/.bashrc  
    echo "source /opt/ros/noetic/setup.bash" | sudo tee -a /home/$USERNAME/.bashrc  
    echo "ROS1 Noetic environment setup added to /home/$USERNAME/.bashrc"  
else  
    echo "ROS1 Noetic environment is already set in /home/$USERNAME/.bashrc"  
fi  
source /home/$USERNAME/.bashrc  
# Initialize rosdepc by fishros under BSD License  
# https://pypi.org/project/rosdepc/#files  
sudo pip install rosdep  
sudo pip install rosdepc  
# sudo pip install -i https://pypi.tuna.tsinghua.edu.cn/simple -U rosdep  
# Init & update rosdep   
sudo rosdepc init > /dev/null  
#sudo rosdep fix-permissions  
# su -l $USERNAME -c 'rosdepc update' > /dev/null  
echo "rosdepc init completed!"  
# System update again  
sudo apt update  
sudo apt dist-upgrade -y  
# Verifying ROS1 installation  
clear  
# Define the variables to be printed  
TEXT1="ROS1 Noetic installation completed!"  
TEXT2="Please open a new terminal and run roscore to verify the installation:"  
TEXT3="roscore"  
# Define the colors  
RED='\033[0;31m'  
BLUE='\033[0;34m'  
GREEN='\033[1;32m'  
NC='\033[0m'  
# Calculate the center of the terminal window  
TERMINAL\_WIDTH=$(tput cols)  
TEXT1\_PADDING=$((($TERMINAL\_WIDTH-${#TEXT1})/2))  
TEXT2\_PADDING=$((($TERMINAL\_WIDTH-${#TEXT2})/2))  
TEXT3\_PADDING=$((($TERMINAL\_WIDTH-${#TEXT3})/2))  
# Print the text in the center of the screen in the desired colors  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo -e "${GREEN}$(printf '%\*s' $TEXT1\_PADDING)${TEXT1} ${NC}"  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT2} ${NC}"  
echo -e "${RED}$(printf '%\*s' $TEXT3\_PADDING)${TEXT3} ${NC}"  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""  
echo ""

## 3. MoveIt！及ROS-Control安装

ROS1 Noetic安装完成后，在机械臂ROS包的rm\_install 文件夹下打开终端，执行以下命令自动安装配置MoveIt！运动规划框架和ROS-Control控制器：

sudo bash moveit1\_install.sh # Install MoveIt! and ROS-Control

#!/bin/bash  
# Version: 1.3  
# Date: 2023-07-19  
# Author: Herman Ye @Realman Robotics  
#  
# Warning: This script assumes that the ubuntu20.04 system and ROS1 Noetic have been installed correctly  
# If not, please execute ros1\_noetic\_install.sh first.  
#  
# set -x  
set -e  
# Get script directory  
SCRIPT\_DIR=$(dirname "$0")  
# Get the username of the non-root user  
USERNAME=$SUDO\_USER  
echo "Current user is: $USERNAME"  
# Check if script is run as root (sudo)  
if [ "$(id -u)" != "0" ]; then  
    echo "This script must be run with sudo privileges. for example: sudo bash moveit1\_install.sh"  
    read -p "Press any key to exit..."  
    exit 1  
fi  
echo "sudo privileges check passed"  
# Check if script is run in ubuntu20.04  
if [ "$(lsb\_release -sc)" != "focal" ]; then  
    echo "This script must be run in ubuntu20.04"  
    read -p "Press any key to exit..."  
    exit 1  
fi  
echo "ubuntu20.04 check passed"  
# Check if script is run in ROS1 Noetic  
if [[ "$(sudo -u $USERNAME dpkg -l ros-noetic-desktop-full)" == \*ii\* ]]; then  
    echo "ROS1 Noetic check passed"  
else  
    echo "This script must be run with ROS1 Noetic-desktop-full"  
    read -p "Press any key to exit..."  
    exit 1  
fi  
# Save logs to files  
LOG\_FILE="${SCRIPT\_DIR}/moveit1\_install.log"  
ERR\_FILE="${SCRIPT\_DIR}/moveit1\_install.err"  
rm -f ${LOG\_FILE}  
rm -f ${ERR\_FILE}  
# Redirect output to console and log files  
exec 1> >(tee -a ${LOG\_FILE} )  
exec 2> >(tee -a ${ERR\_FILE} >&2)  
# Add GitHub520 Host to host for GitHub access in China  
# https://github.com/521xueweihan/GitHub520  
sudo apt install curl -y  
sudo sed -i "/# GitHub520 Host Start/Q" /etc/hosts && curl https://raw.hellogithub.com/hosts >> /etc/hosts  
echo "GitHub520 Host added to host file"  
# sudo sed -i 's/#DNS=/DNS=114.114.114.114/' /etc/systemd/resolved.conf  
# echo "DNS server changed to 114.114.114.114"  
sudo systemctl restart systemd-resolved.service  
echo "Refreshed network settings, sleep 5 seconds"  
sleep 5  
# Install catkin the ROS build system  
sudo apt install ros-noetic-catkin python3-catkin-tools python3-osrf-pycommon -y  
# Install wstool  
sudo apt install python3-wstool -y  
# Install moveit  
sudo apt-get install ros-noetic-moveit -y  
sudo apt-get install ros-noetic-moveit-visual-tools -y  
# Warning: Installing all subpackages of moveit may cause dependency conflicts, please do so with caution.  
# sudo apt install ros-noetic-moveit-\* -y  
# Install ros\_control  
sudo apt-get install ros-noetic-ros-control ros-noetic-ros-controllers -y  
sudo apt-get install ros-noetic-controller-interface ros-noetic-controller-manager-msgs ros-noetic-controller-manager  
# Create A Catkin Workspace and Download MoveIt Source  
sudo rm -rf /home/$USERNAME/ws\_moveit  
mkdir -p /home/$USERNAME/ws\_moveit/src  
cd /home/$USERNAME/ws\_moveit/src  
clear  
sleep 1  
# Define the variables to be printed  
TEXT0=""  
TEXT1="Moveit installation completed!"  
TEXT2="Please open a new terminal and run roslaunch to verify the installation:"  
TEXT3="roslaunch panda\_moveit\_config demo.launch rviz\_tutorial:=true"  
TEXT4="1. Click 'Add' in the left panel, and add the following items:"  
TEXT5="2. Add 'RobotModel', 'MotionPlanning' to the left panel"  
TEXT6="3. Try to drag the end effector to see if the robot arm moves"  
TEXT7="4. Click 'Plan & Execute' to see the robot arm move"  
TEXT8="5. If you see the robot arm move, the installation is successful"  
# Define the colors  
RED='\033[0;31m'  
BLUE='\033[0;34m'  
GREEN='\033[1;32m'  
NC='\033[0m'  
# Calculate the center of the terminal window  
TERMINAL\_WIDTH=$(tput cols)  
TEXT1\_PADDING=$((($TERMINAL\_WIDTH-${#TEXT1})/2))  
TEXT2\_PADDING=$((($TERMINAL\_WIDTH-${#TEXT2})/2))  
TEXT3\_PADDING=$((($TERMINAL\_WIDTH-${#TEXT3})/2))  
# Finished  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
echo -e "${GREEN}$(printf '%\*s' $TEXT1\_PADDING)${TEXT1} ${NC}"  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
read -rp "Do you want to download the tutorial code? (y/n) " confirm  
  if [[ "$confirm" =~ ^[Yy]$ ]]; then  
    # Download Example Code(already in the moveit.rosinstall)  
    # cd /home/$USERNAME/ws\_moveit/src   
    clear  
    echo "Connecting to GitHub, please wait..."  
    echo "If the download stuck here for a long time"  
    echo "please check your network connection and rerun this script"  
    git clone https://github.com/ros-planning/moveit\_tutorials.git -b master  
    git clone https://github.com/ros-planning/panda\_moveit\_config.git -b noetic-devel  
    # git clone https://github.com/ros-controls/ros\_control.git -b noetic-devel  
    # Clone MoveIt packages from source  
    # git clone https://github.com/ros-planning/moveit\_msgs.git  
    # git clone https://github.com/ros-planning/moveit\_resources.git  
    # git clone https://github.com/ros-planning/geometric\_shapes.git --branch noetic-devel  
    # git clone https://github.com/ros-planning/srdfdom.git --branch noetic-devel  
    # git clone https://github.com/ros-planning/moveit.git  
    # git clone https://github.com/PickNikRobotics/rviz\_visual\_tools.git  
    # git clone https://github.com/ros-planning/moveit\_visual\_tools.git  
    # git clone https://github.com/ros-planning/moveit\_tutorials.git  
    # git clone https://github.com/ros-planning/panda\_moveit\_config.git --branch noetic-devel  
        # Rosdepc install  
    cd /home/$USERNAME/ws\_moveit/src  
    rosdepc install -y --from-paths . --ignore-src --rosdistro noetic > /dev/null  
    echo "Rosdep install finished"  
    # Build the Workspace  
    cd /home/$USERNAME/ws\_moveit  
    catkin config --extend /opt/ros/noetic --cmake-args -DCMAKE\_BUILD\_TYPE=Release  
    catkin init  
    catkin build  
    # Environment setup  
    if ! grep -q "/home/$USERNAME/ws\_moveit/devel/setup.bash" /home/$USERNAME/.bashrc; then  
        echo "# ws\_moveit Environment Setting" | sudo tee -a /home/$USERNAME/.bashrc  
        echo "source /home/$USERNAME/ws\_moveit/devel/setup.bash" >> /home/$USERNAME/.bashrc  
        echo "ws\_moveit environment setup added to /home/$USERNAME/.bashrc"  
    else  
        echo "ws\_moveit environment is already set in /home/$USERNAME/.bashrc"  
    fi  
    source /home/$USERNAME/.bashrc  
    # Verifying Moveit1 installation  
    clear  
    # Print the text in the center of the screen in the desired colors  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT1\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT1\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
    echo -e "${GREEN}$(printf '%\*s' $TEXT1\_PADDING)${TEXT1} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT2} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT1\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
    echo -e "${RED}$(printf '%\*s' $TEXT3\_PADDING)${TEXT3} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT1\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT4} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT5} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT6} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT7} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT8} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT1\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT1\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT2\_PADDING)${TEXT0} ${NC}"  
    echo -e "${NC}$(printf '%\*s' $TEXT1\_PADDING)${TEXT0} ${NC}"  
  else  
    read -p "Ok. The installation is complete. Press any key to exit..."  
    exit 0  
  fi

## 4. ROS包安装与编译

### 4.1 创建ROS工作空间

mkdir -p ~/realman\_ws/src # Create a ROS workspace  
cd ~/realman\_ws/ # Enter the workspace  
catkin init # Initialize the workspace

### 4.2 获取ROS包

cd ~/realman\_ws/src/ # Enter the workspace

将睿尔曼智能提供的机械臂ROS源码包拷贝到 realman\_ws工作空间的 src 目录下

**方法1**

如果你将睿尔曼智能提供的机械臂ROS源码包放在了Home目录下，可以使用以下命令将其拷贝到realman\_ws工作空间的 src 目录下

cp -r ~/rm\_65\_robot ~/realman\_ws/src/ # Copy the ROS package to the workspace

**方法2**

使用图形化操作界面，手动将文件夹拷贝到realman\_ws工作空间的 src 目录下

### 4.3 安装依赖

cd ~/realman\_ws # Enter the workspace

提示：

如果你在中国大陆地区，可能会遇到rosdep update命令无法正常执行的情况，这是由于rosdep的源服务器在国外，国内网络无法正常访问导致的。

rosdep update # Update ROS dependencies

此时，你可以使用国内个人开发者制作的rosdepc update命令来代替rosdep update，具体操作如下：

rosdepc update # Update ROS dependencies in China

在完成了rosdep update或者rosdepc update后，安装ROS依赖：

rosdep install -y --from-paths . --ignore-src --rosdistro noetic -r # Install ROS dependencies

提示：

如果你使用其他版本的ROS，需要将此处的noetic更改为你所使用的ROS版本号

### 4.4 编译

cd   
catkin build

## 3. 在RViz中显示模型

cd ~/  
source devel/setup.bash  
roslaunch rm\_65\_description display.launch

    
    
    name="robot\_state\_publisher"  
    pkg="robot\_state\_publisher"  
    type="robot\_state\_publisher" />  
    
    
    name="rviz"  
    pkg="rviz"  
    type="rviz"  
    args="-d $(find rm\_65\_6f\_description)/config/urdf.rviz" />

这行定义了一个名为robot\_description的参数，它的值是通过执行command属性里指定的命令得到的。这个命令的作用是用xacro来解析一个xacro文件（rm\_65\_6f.urdf.xacro），这个文件是用来描述机器人模型的。

$(find xacro)/xacro：用ROS1的launch的find命令，来查找xacro包的路径。然后拼接上/xacro，得到了xacro解析器的完整路径

$(find rm\_65\_6f\_description)/urdf/rm\_65\_6f.urdf.xacro：这部分命令使用了同样的find命令，查找rm\_65\_6f\_description包的路径。然后拼接上/urdf/rm\_65\_6f.urdf.xacro，得到了一个xacro文件的完整路径。这个xacro文件包含了描述机器人模型的数据。

通过xacro 从.xacro文件转为.urdf文件方便robot\_state\_publihser使用。

这行启动了一个名为robot\_state\_publisher的节点，这个节点来自robot\_state\_publisher包，其类型也是robot\_state\_publisher。该节点的任务是读取robot\_description参数，并发布机器人的状态信息。

这行启动了一个名为joint\_state\_publisher\_gui的节点，这个节点来自joint\_state\_publisher\_gui包。此节点的功能是提供一个界面供用户手动控制机器人的关节状态，并且它的输出会显示在屏幕上。

这行启动了一个名为rviz的节点，这个节点来自rviz包。rviz是ROS的一个3D可视化工具，用来展示机器人的状态和环境。args="-d $(find rm\_65\_6f\_description)/config/urdf.rviz"表示启动rviz时，使用rm\_65\_6f\_description包中的urdf.rviz配置文件。-d 是一个命令行选项，用于指定一个RViz配置文件。

这个launch文件的功能是启动一些节点，加载机器人模型，并提供了一个界面供用户操作和观察机器人的状态。