首先安装ROS2

# 1、ROS2安装（换源后装的快点 不然太慢了 这东西还是看运气）

[ROS2安装方法 - ROS2入门教程 (guyuehome.com)](https://book.guyuehome.com/ROS2/1.%E7%B3%BB%E7%BB%9F%E6%9E%B6%E6%9E%84/1.3_ROS2%E5%AE%89%E8%A3%85%E6%96%B9%E6%B3%95/)

如果以下命令

sudo curl -sSL [https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg](https://raw.githubusercontent.com/ros/rosdistro/master/ros.key%20-o%20/usr/share/keyrings/ros-archive-keyring.gpg)  
出错的话

[彻底解决【“curl: (7) Failed to connect to raw.githubusercontent.com port 443: Connection refused”】错误\_curl: (7) failed to connect to nodejs.org port 443-CSDN博客](https://blog.csdn.net/donaldsy/article/details/107482368)

Gitclone老是失败

[解决Linux系统git clone失败或超时问题-CSDN博客](https://blog.csdn.net/weixin_42771853/article/details/133135301" \l ":~:text=%E6%96%87%E7%AB%A0%E6%B5%8F%E8%A7%88%E9%98%85%E8%AF%BB1k%E6%AC%A1%E3%80%82%20%E9%A6%96%E5%85%88%E4%BD%BF%E7%94%A8%20sudo%20vim%20%2Fetc%2Fhosts%20%E8%BF%9B%E5%85%A5hosts%EF%BC%8C%E6%AD%A4%E6%97%B6%E6%98%AF%E6%9F%A5%E7%9C%8B%E6%A8%A1%E5%BC%8F%E3%80%82,%E4%BD%BF%E7%94%A8git%20clone%E5%B8%B8%E5%B8%B8%E4%B8%8D%E6%88%90%E5%8A%9F%EF%BC%8C%E4%BB%A5%E4%B8%8B%E6%98%AF%E8%A7%A3%E5%86%B3%E5%8A%9E%E6%B3%95%EF%BC%8C%E4%BA%B2%E6%B5%8B%E6%9C%89%E6%95%88%E3%80%82%20%E6%8C%89%E4%B8%8B%20i%20%E8%BF%9B%E5%85%A5%E7%BC%96%E8%BE%91%E6%A8%A1%E5%BC%8F%EF%BC%8C%E6%AD%A4%E6%97%B6%E9%9C%80%E8%A6%81%E6%8F%92%E5%85%A5%E4%B8%A4%E4%B8%AAIP%E5%9C%B0%E5%9D%80%E3%80%82%202.%20%E9%87%8D%E5%90%AF%E7%BD%91%E7%BB%9C%E5%92%8C%E6%9C%8D%E5%8A%A1%E5%99%A8%E3%80%82)**（这个测试过后有用！）用这个就行了 非常有用！！！**

法二：看运气 一般般有用

sudo vim /etc/hosts

添加

192.30.253.113 github.com

192.30.252.131 github.com

185.31.16.185 github.global.ssl.fastly.net

74.125.237.1 dl-ssl.google.com

173.194.127.200 groups.google.com

74.125.128.95 ajax.googleapis.com

# 2、安装pangolin

https://github.com/stevenlovegrove/Pangolin

安装pangolin git clone --recursive https://github.com/stevenlovegrove/Pangolin.git

计算机生成了可选文字:
cdPangolin-e.5/
mkdirbuild&&cdbuild
sudomakeinstall

计算机生成了可选文字:
4然0就完成了，可以下面代码进行验证
1Pangolin/bui1d/examp1es/He110Pang01in
2．/He110Pangolin

**gtsam你得在orb和rtab之前装和编译好**

# 3、更换GTSAM版本

（非常关键 镜像源好像没ab92779b25b04b..版本 。最好或者说必须在linux里成功gitclone下来才能切换到ab92779b25b04b..版本（没试过host复制过来行不行）这个还可以探索）

sudo apt remove ros-humble-gtsam

sudo cp -rf /usr/include/eigen3/Eigen /usr/include

sudo cp -rf /usr/include/eigen3 /usr/local/include

sudo cp -rf /usr/include/Eigen /usr/local/include

git clone https://github.com/borglab/gtsam.git

cd gtsam

git checkout ab92779b25b04b376fbbd1846bbbd21904c50e7a

cmakelists文件显示该版本为4.1.0

mkdir build

cd build

计算机生成了可选文字:
t河题也是因为踬事不配，忘了跟你说了，因为懌的配指明了不甲TBS所I瑟关跹它就可以，我
是直在CMa迮Lists．以0吧一行注跹了
include(cmake/HandleTBB.cmake)

cmake GTSAM\_WITH\_TBB=OFF -DGTSAM\_Bcmake -DGTSAM\_BUILD\_WITH\_MARCH\_NATIVE=OFF -DUILD\_EXAMPLES\_ ALWAYS=OFF DGTSAM\_BUILD\_UNSTABLE=OFF -DGTSAM\_USE\_SYSTEM\_EIGEN=ON ..

上面的好像语法错误，要用下面这个

cmake -DGTSAM\_WITH\_TBB=OFF -DGTSAM\_BUILD\_WITH\_MARCH\_NATIVE=OFF -DGTSAM\_BUILD\_EXAMPLES\_ALWAYS=OFF -DGTSAM\_BUILD\_UNSTABLE=OFF -DGTSAM\_USE\_SYSTEM\_EIGEN=ON ..

sudo make install -j4

# 4、安装配置ORB\_SLAM3 v0.4-beta版本

cd ~

git clone <https://github.com/UZ-SLAMLab/ORB_SLAM3.git> ORB\_SLAM3\_v0.4

镜像源（https://gitee.com/zjyfzu/ORB\_SLAM3.git）需要切换好像

git checkout v0.4-beta

cd ORB\_SLAM3\_v0.4\_RGBL/

git checkout v0.4-beta

下载补丁

wget <https://gist.githubusercontent.com/matlabbe/f5cb281304a1305b2824a6ce19792e13/raw/f9faa15c5d35084d123639578ac4ce2ca88bf006/orbslam3_v4_rtabmap_fix.patch>

补丁下载不了的话，window直接打开链接，右键页面另存为后缀为.patch的文件，再复制给linux

git apply orbslam3\_v4\_rtabmap\_fix.patch

chmod +x build.sh

将build.sh文件的make -j 全部改为 make

./build.sh

sudo gedit ~/.bashrc

export ORB\_SLAM\_ROOT\_DIR=~/ORB\_SLAM3\_v0.4

source ~/.bashrc

# 5、安装rtabmap，不能安装在ROS工作空间下

cd ~

git clone <https://github.com/introlab/rtabmap.git> rtabmap

镜像源https://gitee.com/xhb2016888/rtabmap.git

cd rtabmap

git checkout 0.21.1-humble

cd build

cmake -DWITH\_G2O=OFF -DWITH\_ORB\_SLAM=ON ..

计算机生成了可选文字:
ywx@ywx：-Itestl/src/rtabmap
0
With
ZED
With
ZEDOC
With
RealSense
WithRealSense22，54·1
WithmyntEyeS
With
DepthAI
Odomet「yApp「oaches：
With
loamvelodyne
With
floam
With
Itbfovis
With
Iibvts02
With
dVOCore
With
okvis
Withmsckfvio
WithVINS．Fusion
With
OpenVINS
With
0R8SLAM3
NO(ZEDsdkand/orCudanotfound)
NO(ZEDOpenCapturenotfound）
NO(Iibreatsensenotfound）
YES〈License：Apache．2）
NO
NO
NO
NO
NO
NO
NO
NO
NO
NO
NO
(mynteyeSSdknotfound)
(WITH
DEPTHAI=OFF)
(WITH
LOA卜I—OFF）
(WITH
FLOAMOFF)
(WITH
FOVIS=OFF)
(WITH
VIS02=OFF)
(WITH
DVO=OFF）
(WITH
OKVIS=OFF)
(WITHMSCKF_VIO=OFF)
(HITH
VINS—OFF）
(HITH
OPENVINS=OFF)
Showalloptionswith．
Configu「ingdone
Generatingdone
Buildfileshavebeen
ywx@ywx
written
YES（License：GPLv3）
-LA]g「epWITH
to：/home/ywx/testl/src/rtabmap

make -j4（不需要install）

计算机生成了可选文字:
94％]
[94％]
[94％]
[94％]
94％]
[94％]
[94％]
[94％]
94％]
[95％]
[95％]
95％]
[96％]
[96％]
[96％]
96％]
[97％]
[97％]
[97％]
98％]
[98％]
[98％]
[9]
[98％]
[98％]
[98到
98％]
[99％]
[99％]
[99到
99％]
[16謂]
[1％]
YWX@YWX：-/testl/src/rtabmap/build
Builttargetrgbdcamera
BuildingCXX0习e匚ttools/Ca1tbratton/CMakeFttes/catibratton.dtr,/matn.cpp.O
LinkingCXXexecutable一/．．/bin/rtabmap-epipotar_geometry
LinkingCXXexecutable。．/．．/btn/rtabmap-odometryViewer
Builttargeteptpolar_geometry
Building（××objecttoots/matcher/CMakeFites/matcher
Builtta「getodomet「yviewe「
BuildingCXX0e匚texamples/BONMapping/CMakeFtIes/bowmapping.di「/main.cpp.o
LinkingCXXexecutable．．/．./btn/rtabmap-dataRecorder
LinkingCXXexecutable一/．．'bin/rtabmap-calibration
Builtta「getdataReco「de「
Genc「moc_mapBLJiIde「.cpp
BuildingCXXobjectexamples/RCBDMapptng/CMakeFtles/rgbd_mapptng、「/matn.cpp.o
Builttargetcalibration
Generatingmoc_l'iap8utLderÅi_fi.cpp
Gene「moc_mapBuiIdc「·Cpp
Building（××objectexamples/WtftMapptng/CMakeFttes/wtft_mapptng、dir/matn.cpp.o
LinkingCXXexecutable
．/··/bin/rtabmap-bow—mapping
Builtta「getbo城napping
Gene「otingmoc_MopBuLIde「.cpp
Building（××objectexamples/NoEventsExampte/C卜lakeFttes/noEventsExamp1e.dtr/matn.cpp.o
LinkingCXXexecutable一/．./bin/rtabmap-matcher
Builtta「getmatche「
Building〔XXobjectexamples/NoEventsExample/CMakeFiIes/noEventsExampIe.dir/moc_MapButIde「，匚pp0
Building（××object。Cpp·0
BuildingCXXobjectexamples/RGBDMapptng/CMakeFtles/「gbd_mapping.di.「/moc_MapButlde「·cpp@0
BuildingCXXobjectexamples/WifiMapping/CMakeFiles/wtfi_mapping.di「/moc_MapBuiIde「Nifi.cpp.o
LinkingCXXexecutable
．／．。[bin/rtabmap
BuilttargetnoEventsExampIe
LinkingCXXexecutable
Builttargetrgbdmapping
LinkingCXXexecutable
Builtta「getwifi_mapping
-noEventsExampLe
，寻·，/btn/rtabmap-rgbd_mapping
·．/·。-wift_mapptng
ywx@ywx．。/et1/「匚「tabma卩'

OptimizerG2O.cpp中77行三个头文件改成orbslam3中的绝对路径（如果报错需要改的话）

# 6、安装rtabmap\_ros

cd ~/test1/src

git clone <https://github.com/introlab/rtabmap_ros.git> rtabmap\_ros

镜像源https://gitee.com/congcongyanyi/rtabmap\_ros.git

cd rtabmap\_ros/

git checkout 0.21.1-humble

回到test1/目录下

rosdep update && rosdep install --from-paths src --ignore-src -r -y

这一步出错 连接不上的话（看运气）

[rosdep安装与使用\_rosdep install --from-paths src --ignore-src --ros-CSDN博客](https://blog.csdn.net/weixin_45378779/article/details/103617471)

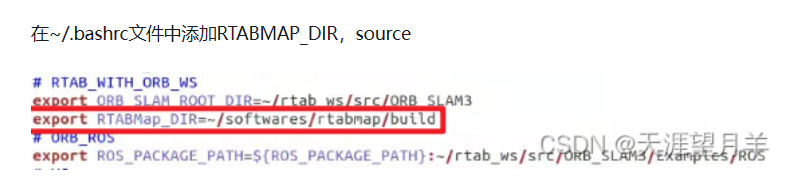
sudo apt remove ros-humble-rtabmap

export MAKEFLAGS="-j2" # Can be ignored if you have a lot of RAM (>16GB)

colcon build --symlink-install --parallel-workers 2 --cmake-args -DCMAKE\_BUILD\_TYPE=Release

**爆红色错 检查不到路径 就重新启动个终端（先加这句）**

export RTABMap\_DIR=~/rtabmap/build **在~/.bashrc添加rtabmap的路径**



可能要自己安装下colcon

sudo apt install python3-colcon-common-extensions

检查是否安装完成

Colcon –help

ros2 run rtabmap\_slam rtabmap --verison检查是否成功接入orbslam3

上面那个输出不了跟下面那个一样用下面那个指令

ros2 run rtabmap\_odom rgbd\_odometry --version

计算机生成了可选文字:
YWX@YWX:～
一5Source．
ywx@ywx：
5「0s2「un「tabmap_slam「tabmap
ywx@ywx·
;[INFO][1695788352．699119179]
0·00e000
[INFO][1695788352．70e065053]
30．000000
[INFO][1695788352．7ee112131]
.[INFO][1695788352．70e159528]
false
[INFO][1695788352．706192552]
。[INFO][1695788352．7ee219282]
t「t_Je
.[INFO][1695788352．70e244924]
false
[INFO][1695788352．73e271157]
minneighbors
2
,[INFO][1695788352．7ee348273]
16
rtabmap(maps)：
rtabmap(maps)：
rtabmap(maps)：
rtabmap(maps)：
rtabmap(maps)：
rtabmap(maps):
rtabmap(maps)：
「tabmap(maps)：
rtabmap(maps)：
Q
map_fttter「adius
napfilterangle
map_cteanup
map_always_update
mapempty_rayt「acing
cloudoutput_voxettzed
cloudsubt「actfilte「ing
CLOUdsubtractfiltering
octomap_tree_depth
[INFO]
[INFO]
[INFO]
[INFO]
[INFO]
[INFO]
[INFO]
[INFO]
[INFO]
[INFO]
[1695788352·72213e584]
[1695788352．722196719]
[1695788352“722206214]
[1695788352。722234S29]
[1695788352．722259933}
[1695788352．722291513]
[1695788352．722322173]
[1695788352·722331857]
[1695788352“722343953]
[16957g8352722355743]
[rtabmap]：
[rtabmap]，
[rtabmap]：
[「tabmap]：
[「tabmap]·
[rtabmap]:
[rtabmap]：
[rtabmap]：
[rtabmap]：
[rtabmap]：
[rtabmap]：
[rtabmap]，
[rtabmap]：
[rtabmap]：
[rtabmap]：
[「tab卩旧习：
[rtabmap]：
[「tabmap]，
[rtabmap]，
rtabmap•
rtabmap.
「tabmap.
rtabmap.
rtabmap.
「tabmap.
「tabmap•
rtabmap.
「tabmap.
rtabmap:
id
mapframeid
baselink
map
logto「osoutlevel
tntttalpose
useactionfor_goat
false
tfdelay
．050300
tftole「ance
0·100000
0d05en50「一Syn（
false
false
gen_scan
gendepth
false
RTAB-Map:
PCL：
WithVTK：
OpenCV：
With
With
With
With
With
With
With
With
OpenCVxfeatures2d：
OpenCVnonfree：
ORBOcTree：
SuperpointTo「ch：
Python3:
FastCV：
OpenGV:
Madgwtck：
1．12。1
false
false
false
false
false
false
t「

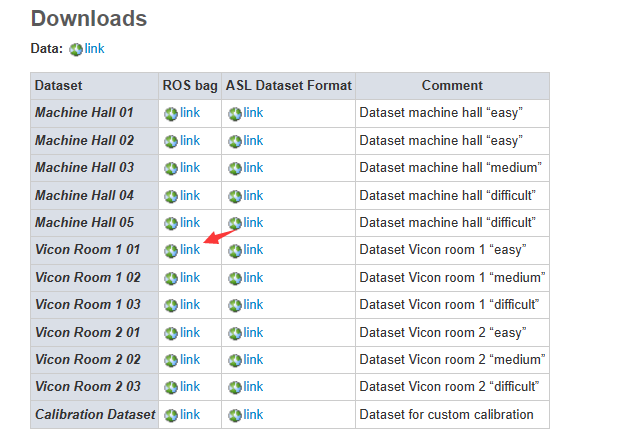
计算机生成了可选文字:
Q
With
Python3:
With
FastCV：
With
OpenGV：
With
Madgwtck：
With
PDAL：
With
TORO：
With
g20：
With
GTSAM：
With
Vertigo:
With
CVSBA：
With
Ceres：
With
OpenN12：
With
With
F「eenect2·
With
K4W2：
With
K4A：
With
DC1394：
With
FlyCapture2：
With
ZED：
With
ZEDOpenCaptu「e:
With
RealSense:
With
RealSenseSLAM：
With
RealSense2：
With
MYNTEYES：
With
DepthAI：
With
tibpotntmatcher：
With
CCCoreLtb：
With
octonap：
With
Cpu-tsdf:
With
openchisel。
WithAliceVision：
With
LOA卜1：
With
FLOAM：
With
FOVIS：
With
vt502：
With
0：
With
ORBSLAM3：
With
OKVIS：
With
MSCKFVIO：
With
VINS-Fuston:
With
OpenVINS：
Y'以@尹以5
YWX@YWX:～
false
false
false
false
false
false
false
trtJe
false
false
false
t「刂e
false
false
false
false
false
true
false
false
false
false
false
false
false
false
false
false
false
false
false
false
false

# 7、数据集测试

**一、euroc\_datasets.launch.py 测试**

[kmavvisualinertialdatasets – ASL Datasets (ethz.ch)](https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets)

下载



**前置安装 不知道哪些是必备的 但是都装了**

sudo apt install python3-rosbag

sudo apt-get install ros-humble-image-proc

sudo apt-get install ros-humble-imu-complementary-filter

sudo apt install python3-pip

sudo apt update

sudo apt install ros-humble-rosbag2\*

**第一步**

euroc\_datasets.launch.py 在

/home/tlc/test1/install/rtabmap\_examples/share/rtabmap\_examples/launch

底下 所以用下面的命令

执行

ros2 launch rtabmap\_examples euroc\_datasets.launch.py args:="Odom/Strategy 5 OdomORBSLAM/VocPath /home/rosadmin/test1/src/ORB\_SLAM3/Vocabulary/ORBvoc.txt" MH\_seq:=true raw\_images\_for\_odom:=true

**第二步**

sudo pip install rosbags

rosbags-convert V1\_01\_easy.bag

cd V1\_01\_easy

执行

ros2 bag play V1\_01\_easy.db3 --clock

参考资料：

# Example to run euroc datasets:

# $ sudo pip install rosbags # See https://docs.openvins.com/dev-ros1-to-ros2.html

# $ rosbags-convert V1\_01\_easy.bag

# $ rosbags-convert MH\_01\_easy.bag

#

# $ ros2 launch rtabmap\_examples euroc\_datasets.launch.py gt:=true

# $ cd V1\_01\_easy

# $ ros2 bag play V1\_01\_easy.db3 --clock

#

# $ ros2 launch rtabmap\_examples euroc\_datasets.launch.py gt:=false

# $ cd MH\_01\_easy

# $ ros2 bag play MH\_01\_easy.db3 --clock

sudo apt install python3-rosbag

sudo apt-get install ros-humble-image-proc

sudo apt-get install ros-humble-imu-complementary-filter

sudo apt install python3-pip

sudo apt update

sudo apt install ros-humble-rosbag2\*

二、rgbdslam\_datasets.launch.py 测试（没实现）

# Example to run rgbd datasets:

# [ROS1] Prepare ROS1 rosbag for conversion to ROS2

#wget http://vision.in.tum.de/rgbd/dataset/freiburg3/rgbd\_dataset\_freiburg3\_long\_office\_household.bag

# $ rosbag decompress rgbd\_dataset\_freiburg3\_long\_office\_household.bag

#wget https://raw.githubusercontent.com/srv/srv\_tools/kinetic/bag\_tools/scripts/change\_frame\_id.py

# Edit change\_frame\_id.py, remove/comment lines beginning with "PKG" and "import roslib", change line "Exception, e" to "Exception"

# $ roscore

#python3 change\_frame\_id.py -o rgbd\_dataset\_freiburg3\_long\_office\_household\_frameid\_fixed.bag -i rgbd\_dataset\_freiburg3\_long\_office\_household.bag -f openni\_rgb\_optical\_frame -t /camera/rgb/image\_color

# [ROS2]

# sudo pip install rosbags # See https://docs.openvins.com/dev-ros1-to-ros2.html

#rosbags-convert rgbd\_dataset\_freiburg3\_long\_office\_household\_frameid\_fixed.bag

# ros2 launch rtabmap\_examples rgbdslam\_datasets.launch.py

#cd rgbd\_dataset\_freiburg3\_long\_office\_household\_frameid\_fixed

#ros2 bag play rgbd\_dataset\_freiburg3\_long\_office\_household\_frameid\_fixed.db3 –clock

# 8、构建无人机

参考

[NovoG93/sjtu\_drone：ROS/ROS 2 gazebo四轴飞行器模拟器。 (github.com)](https://github.com/NovoG93/sjtu_drone)

**要求**

此软件包已使用 ROS 2 Humble 版本 （Ubuntu 22.04） 和 Gazebo 11 进行了测试。

**下载和构建**

cd ~/git && git clone git@github.com:NovoG93/sjtu\_drone.git -b ros2

**(下不下来就win下下来再复制过去)**

cd ~/test1/src && ln -s ~/git/sjtu\_drone

cd .. && rosdep install -r -y --from-paths src --ignore-src --rosdistro $ROS\_DISTRO && colcon build --packages-select-regex sjtu\*

**启动**

**ROS 2 Source Installation**

1. Start gazebo, spawn drone, open teleop in xterm window, and open rviz:  
   ros2 launch sjtu\_drone\_bringup sjtu\_drone\_bringup.launch.py
2. Takeoff:  
   ros2 topic pub /drone/takeoff std\_msgs/msg/Empty {} --once
3. Move drone: (use teleop window)
4. Land:  
   ros2 topic pub /drone/land std\_msgs/msg/Empty {} --once

使用gazebo仿真时遇到的问题

**仿真时严重卡顿问题:**

尝试下打开gazebo在Camera栏中先选择Orthographic，此时会发现拖拉栅格十分丝滑，仿真也变得丝滑了很多，再根据需要考虑是否需要再切换回Perspective。