

# 系统安装

## 1. 下载官方镜像源: [Jetson Download Center | NVIDIA Developer](#)

Jetson Nano Developer Kit SD Card Image

4.6 2021/08/04

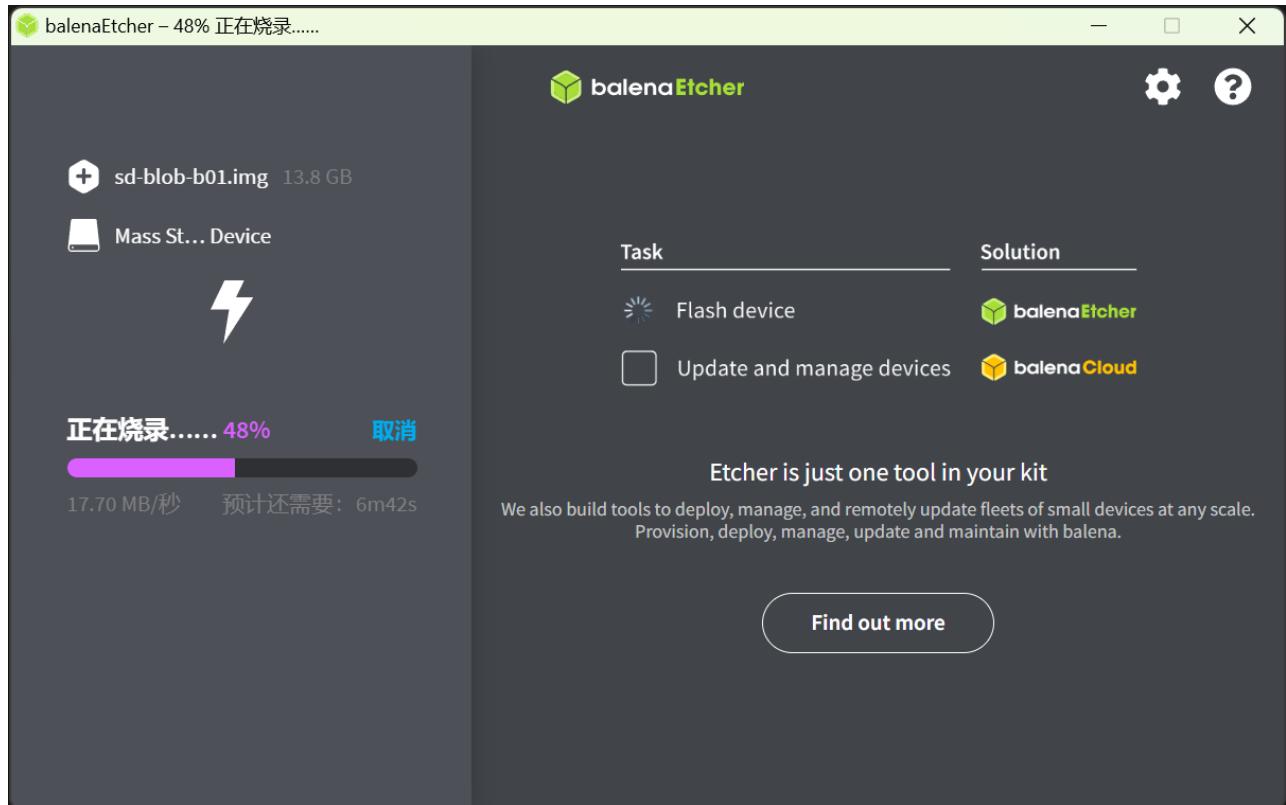
This SD card image works for all Jetson Nano Developer Kits (both 945-13450-0000-100 and the older 945-13450-0000-000) and is built with JetPack 4.6. Download and write it to your microSD card and use it to boot the developer kit.

md5sum: 7c853a93e4c94400978aaa7a7d27b7ea

[More Information >](#)

DOWNLOADS  
[Jetson Nano Developer Kit SD Card Image](#)

## 2. 使用烧录软件烧录系统



## 3. 安装镜像源

# ROS安装

## 1. 使用小鱼进行快速换源、安装ros-melodic

```
1 wget http://fishros.com/install -O fishros && . fishros
```

# Realsense SDK安装

理论上可以使用apt安装，但由于apt安装版本与ros-realsense包存在兼容问题，无法获得imu流，所以采用源码编译的方式获得最新的SDK版本

## 一、安装依赖项

### 1. 更新

```
1 sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade
```

### 2. 安装构建librealsense二进制文件所需的核心包和受影响的内核模块

```
1 sudo apt-get install libssl-dev libusb-1.0-0-dev libudev-dev pkg-config libgtk-3-dev
```

### 3. 安装构建工具

```
1 sudo apt-get install git wget cmake build-essential
```

### 4. 准备Linux后端和开发环境（需拔下摄像头）

```
1 sudo apt-get install libglfw3-dev libgl1-mesa-dev libglu1-mesa-dev at
```

## 二、安装librealsense2

### 1. 下载librealsense2的最新稳定版本

```
1 git clone https://github.com/IntelRealSense/librealsense.git
```

### 2. 从librealsense2根目录运行权限脚本

```
1 cd librealsense
2 ./scripts/setup_udev_rules.sh
3
4 # 可以通过运行以下命令删除权限
5 ./scripts/setup_udev_rules.sh --uninstall
```

### 3. 为以下对象构建和应用修补的内核模块：

- Ubuntu 20/22 (focal/jammy) , 带 LTS 内核 5.13、5.15

```
1 ./scripts/patch-realsense-ubuntu-lts-hwe.sh
```

- 带 LTS 内核的 Ubuntu 18/20 (< 5.13)

```
1 ./scripts/patch-realsense-ubuntu-lts.sh
```

- Jetson NANO 基于4.9的自定义内核

```
1 ./scripts/patch-realsense-ubuntu-L4T.sh
```

### 4. 通过检查生成的日志以及检查内核日志中的最新条目来检查修补的模块安装

```
1 sudo dmesg | tail -n 50
```

```
2 应该出现与uvcvideo有关的条目
```

运行

```
[17131.047232] usbcore: deregistering interface driver uvcvideo
[17131.129422] uvcvideo: loading out-of-tree module taints kernel.
[17131.138168] usbcore: registered new interface driver uvcvideo
[17131.138172] USB Video Class driver (1.1.1)
```

uvcvideo 驱动程序经历了注销、加载和注册的过程，并且现在已在系统中激活，版本为 1.1.1。

## 三、构建realsense2 SDK

### 1. 创建build目录

```
1 mkdir build && cd build
```

### 2. 运行cmake configure

```
1 # 默认构建 librealsense2 以及演示和教程
2 cmake ../
3 # 构建系统用于优化模式，通常用于发布版本。在优化模式下，编译器会尝试提高程序的运行效率，可能会禁用某些调试信息。
4 cmake ../ -DCMAKE_BUILD_TYPE=Release
5 # 同时构建librealsense2库、演示程序和教程。
6 cmake ../ -DBUILD_EXAMPLES=true
7 # 对于没有 OpenGL 或 X11 的系统，仅构建文本示例
8 cmake ../ -DBUILD_EXAMPLES=true -DBUILD_GRAPHICAL_EXAMPLES=false
```

### 3. 重新编译并安装 librealsense2 二进制文件

```
1 sudo make uninstall && make clean && make && sudo make install
```

## Realsense\_ROS安装

### 1. 创建工作空间

```
1 mkdir -p ~/catkin_ws/src
2 cd ~/catkin_ws/src
```

### 2. git clone 下载ros-realsense包和ddynamic\_reconfigure包（如果网络问题无法gitclone，可以到github上下载安装包再解压，注意ros版本melodic对应的是ros1，功能包名为realsense-ros-ros1-legacy）

```
1 git clone https://github.com/IntelRealSense/realsense-ros.git
2 git clone https://github.com/pal-robotics/ddynamic_reconfigure.git
```

### 3. 版本检查

```
1 cd realsense-ros/
2 git tag | sort -V | grep -P "^\d+\.\d+\.?(\d+)?\d?" | tail -1
```

### 4. 编译软件包

```
1 cd ~/catkin_ws/src  
2 catkin_init_workspace  
3 cd ..  
4 catkin_make clean  
5 catkin_make -DCATKIN_ENABLE_TESTING=False -DCMAKE_BUILD_TYPE=Release  
6 catkin_make install
```

catkin\_make clean后报错，表明cv\_bridge需要的/usr/include/opencv目录没有找到

```
CMake Error at /opt/ros/melodic/share/cv_bridge/cmake/cv_bridgeConfig.cmake:113 (message):  
Project 'cv_bridge' specifies '/usr/include/opencv' as an include dir,  
which is not found. It does neither exist as an absolute directory nor in  
'$prefix'//usr/include/opencv'. Check the issue tracker  
'https://github.com/ros-perception/vision_opencv/issues' and consider  
creating a ticket if the problem has not been reported yet.  
Call Stack (most recent call first):  
/opt/ros/melodic/share/catkin/cmake/catkinConfig.cmake:76 (find_package)  
realsense-ros/realsense2_camera/CMakeLists.txt:11 (find_package)  
  
-- Configuring incomplete, errors occurred!  
See also "/home/z/catkin_ws/build/CMakeFiles/CMakeOutput.log".  
See also "/home/z/catkin_ws/build/CMakeFiles/CMakeError.log".  
Invoking "cmake" failed
```

a. 检查opencv是否安装

```
1 pkg-config --modversion opencv4
```

出现版本号代表已安装

```
z@z:~/catkin_ws$ pkg-config --modversion opencv4  
4.1.1
```

若没有安装可以用以下命令安装OpenCV 4

```
1 sudo apt-get install libopencv-dev
```

b. 检查OpenCV的安装路径

```
1 # 搜索系统文件查找包含opencv2的目录  
2 find / -name "opencv2" -type d 2>/dev/null
```

```
z@z:~/catkin_ws$ find / -name "opencv2" -type d 2>/dev/null  
/usr/include/opencv4/opencv2
```

### c. 先将更改报错的cmake文件路径

```
1 cd /opt/ros/melodic/share/cv_bridge/cmake/  
2 sudo vim cv_bridgeConfig.cmake
```

```
# flag project as catkin-based to distinguish if a find_package()-ed project is a catkin project  
set(cv_bridge_FOUND_CATKIN_PROJECT TRUE)  
  
if(NOT "include;/usr/include;/usr/includeopencv2" STREQUAL "")  
    set(cv_bridge_INCLUDE_DIRS "")  
    set(_include_dirs "include;/usr/include;/usr/includeopencv2")  
    if(NOT "https://github.com/ros-perception/vision_opencv/issues" STREQUAL "")  
        set(_report "Check the issue tracker 'https://github.com/ros-perception/vision\_opencv/issues' and consider creating a ticket if the problem has not been reported yet.")  
    elseif(NOT "http://www.ros.org/wiki/cv_bridge" STREQUAL "")  
        set(_report "Check the website 'http://www.ros.org/wiki/cv\_bridge' for information and consider reporting the problem.")  
    else()  
        set(_report "Report the problem to the maintainer 'Vincent Rabaud <vincent.rabaud@gmail.com>' and request to fix the problem.")  
    endif()
```

### d. 再将opencv2目录移动到/usr/include/目录下，以解决所有文件依赖问题

```
1 # 因为已经查过，ros-realsense包中依赖的opencv路径都是/usr/include/opencv2  
2 sudo mv /usr/include/opencv4/opencv2 /usr/include/
```

## 5. 添加环境变量

```
1 echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc  
2 source ~/.bashrc
```

## 6. 运行launch文件测试节点

```
1 roslaunch realsense2_camera rs_camera  
2 # 新开另一个终端  
3 rostopic list
```

```
#@zj:~$ rostopic list
/camera/color/camera_info
/camera/color/image_raw
/camera/color/image_raw/compressed
/camera/color/image_raw/compressed/parameter_descriptions
/camera/color/image_raw/compressed/parameter_updates
/camera/color/image_raw/compressedDepth
/camera/color/image_raw/compressedDepth/parameter_descriptions
/camera/color/image_raw/compressedDepth/parameter_updates
/camera/color/image_raw/theora
/camera/color/image_raw/theora/parameter_descriptions
/camera/color/image_raw/theora/parameter_updates
/camera/color/metadata
/camera/depth/camera_info
/camera/depth/image_rect_raw
/camera/depth/image_rect_raw/compressed
/camera/depth/image_rect_raw/compressed/parameter_descriptions
/camera/depth/image_rect_raw/compressed/parameter_updates
/camera/depth/image_rect_raw/compressedDepth
/camera/depth/image_rect_raw/compressedDepth/parameter_descriptions
/camera/depth/image_rect_raw/compressedDepth/parameter_updates
/camera/depth/image_rect_raw/theora
/camera/depth/image_rect_raw/theora/parameter_descriptions
/camera/depth/image_rect_raw/theora/parameter_updates
/camera/depth/metadata
/camera/extrinsics/depth_to_color
/camera/motion_module/parameter_descriptions
/camera/motion_module/parameter_updates
/camera/realsense2_camera_manager/bond
/camera/rgb_camera/auto_exposure_roi/parameter_descriptions
/camera/rgb_camera/auto_exposure_roi/parameter_updates
/camera/rgb_camera/parameter_descriptions
/camera/rgb_camera/parameter_updates
/camera/stereo_module/auto_exposure_roi/parameter_descriptions
/camera/stereo_module/auto_exposure_roi/parameter_updates
/camera/stereo_module/parameter_descriptions
/camera/stereo_module/parameter_updates
/diagnostics
/rosout
/rosout_agg
/tf
/tf_static
```

发现没有imu节点，因为在launch配置中imu默认不启动

### a. 修改launch文件

```
1 cd ~/catkin_ws/src/realsense-ros-ros1-legacy/realsense2_camera/launch
2 vim rs_camera
3
4 # 将<arg name="unite_imu_method"      default="" />修改为
5 <arg name="unite_imu_method"      default="linear_interpolation" />
6 # 将<arg name="enable_gyro"        default="false" />修改为
7 <arg name="enable_gyro"        default="true" />
8 # 将<arg name="enable_accel"      default="false" />修改为
9 <arg name="enable_accel"      default="true" />
```

```
<arg name="enable_gyro"          default="true" />
<arg name="enable_accel"         default="true" />
```

```
<arg name="unite_imu_method"     default="linear_interpolation" />
```

### b. source配置文件

```
1 source ~/.bashrc
```

### c. 重新运行launch文件

```

1 rosrun realsense2_camera rs_camera.launch
2 # 新开另一个终端
3 rostopic list

```

```

/camera/color/image_rect_color/info
/camera/color/image_rect_color
/camera/color/image_rect_color/compressed
/camera/color/image_rect_color/compressed/parameter_descriptions
/camera/color/image_rect_color/compressed/parameter_updates
/camera/color/image_rect_color/compressedDepth
/camera/color/image_rect_color/compressedDepth/parameter_descriptions
/camera/color/image_rect_color/compressedDepth/parameter_updates
/camera/color/image_rect_color/theory
/camera/color/image_rect_color/parameter_descriptions
/camera/color/image_rect_color/parameter_updates
/camera/color/metaData
/camera/color/rgb
/camera/color/rgb/info
/camera/color/rgb/image_rect_color/compressed
/camera/color/rgb/image_rect_color/compressed/parameter_descriptions
/camera/color/rgb/image_rect_color/compressed/parameter_updates
/camera/color/rgb/image_rect_color/compressedDepth
/camera/color/rgb/image_rect_color/compressedDepth/parameter_descriptions
/camera/color/rgb/image_rect_color/compressedDepth/parameter_updates
/camera/color/rgb/theory
/camera/color/rgb/theory/parameter_descriptions
/camera/color/rgb/theory/parameter_updates
/camera/color/theory
/camera/color/theory/parameter_descriptions
/camera/color/theory/parameter_updates
/camera/color/theory/metaData
/camera/extrinsic/depth_to_color
/camera/extrinsic/depth_to_color/info
/camera/imu
/camera/imu/parameter_descriptions
/camera/imu/parameter_updates
/camera/realsense2_camera_manager/bond
/camera/realsense2_camera_manager/parameter_descriptions
/camera/realsense2_camera_manager/parameter_updates
/camera/realsense2_camera_manager/metaData
/camera/stereo_module/auto_exposure_roi/parameter_descriptions
/camera/stereo_module/auto_exposure_roi/parameter_updates
/camera/stereo_module/metaData
/camera/stereo_module/parameter_descriptions
/camera/stereo_module/parameter_updates
/rosout
/rosout_agg
/rf_static

```

发现 /camera/imu 节点

#### d. 检查imu话题是否正常

```

1 # 若在安装realsenseSDK和realsense-ROS包时使用apt安装，可能导致不兼容，无法获得imu流
2 rostopic echo /camera/imu

```

```

seq: 2529
stamp:
  secs: 1731425815
  nsecs: 372588158
frame_id: "camera_imu_optical_frame"
orientation:
  x: 0.0
  y: 0.0
  z: 0.0
  w: 0.0
orientation_covariance: [-1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: -0.0029305132
  y: 0.00283146861345
  z: -0.00681956493477
angular_velocity_covariance: [0.01, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.01]
linear_acceleration:
  x: -0.39738972502
  y: -0.191900004544
  z: 0.00279345008479
linear_acceleration_covariance: [0.01, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.01]
<...
header:
  seq: 2524
  stamp:
    secs: 1731425815
    nsecs: 377599855
frame_id: "camera_imu_optical_frame"
orientation:
  x: 0.0
  y: 0.0
  z: 0.0
  w: 0.0
orientation_covariance: [-1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: 6.92788125889e-06
  y: -0.00157713479931
  z: -0.001330622353
angular_velocity_covariance: [0.01, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.01]
linear_acceleration:
  x: 0.403546267276
  y: 0.001330622353
  z: 0.00279355898875
linear_acceleration_covariance: [0.01, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.01]

```

## 夹爪设计

见文件handle\_with\_nano.stl

## 程序编写

#### 1. 新建工作空间

```
1 mkdir -p ~/z_ws/src
2 cd ~/z_ws/src
3 catkin_init_workspace
4 catkin_create_pkg gripper rospy std_msgs    # rospy std_msgs是所需依赖
5 cd ~/z_ws/
6 catkin_make
7 # 使用python开发ros只需要一次catkin_make，目的是将pkg加入到ROS的pkg列表，ROS能够找到
8 echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
9 source ~/.bashrc
```

## 2. 编写代码gripper.py(控制夹爪开合), record\_bag.py(记录数据)

```
1 cd ~/z_ws/src/gripper
2 mkdir scripts && cd scripts
3 vim gripper.py
4 vim record_bag.py
5 # 为能够使用rosrun启动增加运行权限
6 sudo chmod +x gripper.py record_bag.py
```

见文件

gripper.py

record\_bag.py

## 3. 测试

```
1 roscore
2 # 新开终端
3 roslaunch realsense2_camera rs_camera.launch
4 # 新开终端
5 rosrun gripper gripper.py
6 # 新开终端
7 rostopic echo /servoping
```

```
z@z:~$ rosrun gripper gripper.py
[WARN] [1731428450.093082]: servoping is successfully initialized
```

```
---
layout:
  dim: []
  data_offset: 0
data: [1.0, 1.0, 5.900000000000001]
---
layout:
  dim: []
  data_offset: 0
data: [1.0, 1.0, 5.900000000000001]
---
layout:
  dim: []
  data_offset: 0
data: [1.0, 1.0, 5.900000000000001]
---
layout:
  dim: []
  data_offset: 0
data: [1.0, 1.0, 5.900000000000001]
---
layout:
  dim: []
  data_offset: 0
data: [1.0, 1.0, 5.900000000000001]
```

```
1 # 新开终端
2 rosrun gripper record_bag.py
3 #按按钮测试
```

```
z@z:~/z_ws/src/gripper/scripts$ rosrun gripper record_bag.py
/usr/lib/python2.7/dist-packages/Jetson/GPIO/gpio.py:386: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings
  RuntimeWarning)
[WARN] [1731435701.817659]: okk, start to record!!!! Press the button again to stop.
[WARN] [1731435704.356691]: stop to record!!!!
[WARN] [1731435706.933073]: okk, start to record!!!! Press the button again to stop.
[WARN] [1731435712.657004]: stop to record!!!!
```

#### 4. 确认无误后将这两个节点写进rs\_camera.launch (启动更快)

```
1 cd ~/catkin_ws/src/realsense-ros-ros1-legacy/realsense2_camera/launch
2 vim rs_camera.launch
3
4 # 在launch中添加以下内容
5 <node pkg="gripper" type="gripper.py" name="servoping" output="screen"/>
6 <node pkg="gripper" type="record_bag.py" name="record_bag" output="screen"/>
```

```
<node pkg="gripper" type="gripper.py" name="servoping" output="screen"/>
<node pkg="gripper" type="record_bag.py" name="record_bag" output="screen"/>
```

#### 5. 运行测试

```
1 roslaunch realsense2_camera rs_camera.launch
```

# 设置开机自动启动launch

```
1 cd /etc/systemd/system  
2 sudo vim launch.service  
3 # 按照如下编辑  
4  
5 #设置服务文件的权限  
6 sudo chmod 644 /etc/systemd/system/launch.service  
7 #启动服务  
8 sudo systemctl start launch.service  
9 #设置服务开机自启动  
10 sudo systemctl enable launch.service
```

```
[Unit]  
Description=My Launch File  
  
[Service]  
ExecStart=/usr/bin/roslaunch realsense2_camera rs_camera.launch  
Restart=always  
User=z  
Environment="DISPLAY=:0"  
Environment="ROS_MASTER_URI=http://localhost:11311"  
  
[Install]  
WantedBy=multi-user.target
```

```
z@z:/etc/systemd/system$ sudo chmod 644 /etc/systemd/system/launch.service  
z@z:/etc/systemd/system$ sudo systemctl start launch.service  
z@z:/etc/systemd/system$ sudo systemctl enable launch.service  
Created symlink /etc/systemd/system/multi-user.target.wants/launch.service → /etc/systemd/system/launch.service.
```